Speech Generation and Modification in Concatenative Speech Synthesis

Ingmund Bjørkan

A Dissertation Submitted in Partial Fulfillment
of the Requirements for the Degree of

Philosophiae Doctor

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

2010

NTNU
Norwegian University of Science and Technology

Thesis for the degree philosophiae doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

ISBN 978-82-471-2094-1 (printed version)
ISBN 978-82-471-2096-5 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU,2010:68

Abstract

The work presented in this thesis is related to speech generation and speech modification in unit selection synthesis. A major problem in unit selection synthesis systems is the large variability in the synthetic speech quality due to audible discontinuities. Although using an exhaustive search in a large speech unit database, audible discontinuities occasionally occur due to concatenating speech units from different speech contexts which do not fit together acoustically. The main focus in this thesis has been to alleviate the problem of audible discontinuities at concatenation points.

The thesis consists of a theory part and three experiment chapters. The first experiment chapter concerns the selection of speech units from the speech unit database, in order to better avoid audible discontinuities at concatenation points. A listening test on detecting discontinuities in vowel joins is presented. A comparison of different objective spectral distance measures was then performed, using the ratings from the listening test as a reference. In addition to classic spectral distance measures, a correlation based distance measure was tested in this experiment. This distance measure was found to be very correlated to pitch mismatches, and not so promising for detecting spectral mismatches. The distance measures' correlation to human ratings were however relatively low in this test. In addition, such a test would be influenced by the specific test design and the synthesis system. Hence, too certain conclusions can not be drawn from this experiment. Join cost function design based on perceptual experiments is then discussed, and a probabilistic join cost model is proposed.

Another approach to alleviate the problem of audible discontinuities is to apply modification of the speech signal by the use of signal processing. The strategy is to apply a speech model, and then smooth estimated speech parameter trajectories across the concatenation points. Finally, synthetic speech can be reconstructed by the speech model. In this thesis, the use of modification by a harmonic speech model has been tested for smoothing of pitch discontinuities and spectral mismatches.

The use of speech modification gives an additional need for robust speech analysis, which is the topic of the second experiment chapter. Specifically, a robust speech processing algorithm was developed for the estimation of parameters for a pitch synchronous harmonic speech model. The processing algorithm can however be applied for any type of pitch synchronous speech processing. The algorithm is based on a pitch synchronous frame based approach, using *zero phase instants* as analysis instants, where the zero phase instants are defined as the discrete time instants nearest to zero phase of the first harmonic component. The robustness of the pitch estimation was experienced to be essential for the performance of this algorithm. An approach for self-validation of the pitch estimate at run time of the algorithm was therefore added. This approach was based on using the distance between estimated zero phase instants to validate the regular frame based pitch estimate at each frame. This approach was experienced to detect possible pitch and voicing estimation errors, and regions of turbulent or irregular speech where no pitch was well defined from a signal processing view.

Three different pitch estimators were tested and compared for the use in this speech processing algorithm. A pitch estimator based on maximizing the harmonic-to-noise ratio (HNR), a pitch estimator based on maximizing the signal-to-noise ratio (SNR) of a harmonic model, and a pitch estimator based on the ESPRIT algorithm. In a comparison of these pitch estimators, the HNR-based pitch estimator was found as the most appropriate. This was due to that the HNR-based pitch estimator did not depend on any prior coarse pitch estimate, and it was also effective, simple, and produced relatively smooth estimated pitch contours. The estimator was made robust to pitch halving errors by introducing a modified version of the HNR estimator, referred to as the *average-HNR*, which corresponds to a penalizing of low pitch period estimates. In a comparison of the pitch estimators to a reference obtained from manually checked pitch marks, it was found that the average-HNR pitch estimator was practically unbiased to this reference. It was also found that this pitch estimator was a good approximation to the computationally more complex SNR-based pitch estimator.

The work on speech modification, presented in the third experiment chapter, has been focused on the use of different variants of a frame based pitch synchronous harmonic sine wave model. Special for the algorithms described in this thesis, is the use of the zero phase instants as analysis instants, leading to a strictly pitch synchronous approach. When evaluating a speech modification method, two subjects are of special interest: The ability of the method to remove audible discontinuities, and the resulting overall speech quality or timbre.

For a test of the overall speech quality or timbre of the modified speech, a listening test was conducted, comparing pitch modified speech by the pitch synchronous harmonic model to pitch modified speech by the classic TD-PSOLA approach for one female voice and one male voice. The two approaches gave modified speech of relatively similar quality. The pitch synchronous harmonic model approach was preferred for the female voice. A problem with some noise, or hoarseness, in the synthetic speech was however encountered when excessively lowering the pitch. This was probably the reason for that the TD-PSOLA approach was preferred when lowering the pitch for the male voice.

For the smoothing of audible discontinuities in unit selection synthesized speech, it was experienced that pitch discontinuities could be successfully smoothed (removed) in many cases, while the methods tested for smoothing of audible discontinuities due to spectral mismatches were not very successful. For the smoothing of pitch, an approach referred to as *global smoothing* is proposed. This approach was found to avoid the problem of short duration speech units having too few speech frames for proper smoothing.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* (PhD) at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU). My supervisor has been Torbjørn Svendsen at the Department of Electronics and Telecommunications, NTNU.

The work has been carried out in the period from August 2003 to October 2009, including the equivalent of half a year of full time courses, as well as teaching assistant duties for the department. The PhD work was funded by a scholarship from the Research Council of Norway through the FONEMA project, which is part of the language technology programme KUNSTI.

Acknowledgements

First and foremost, I would like to thank my supervisor Torbjørn Svendsen for all his help during the work of this thesis. Through valuable comments, corrections, discussions, review, and advices, he has contributed significantly to the understanding and progress during the work on this thesis. I would also like to thank all the coworkers on the Fonema project for interesting discussions and collaboration. In particular, I would like to thank Ingunn Amdal and Dyre Meen. Ingunn for her help with the experiment on applying the work on speech modification to the Festival unit selection system and for helpful comments and corrections to this thesis. I will thank Dyre for his contribution to the work with the ESPRIT algorithm for voicing and pitch estimation, by both sharing ideas and some Python code. Dyre is also acknowledged for his work on the speech synthesis processor, which made it possible to have a complete speech synthesis system working in the Python programming language. I am in addition grateful to Snorre Farner for numerous interesting discussions on speech synthesis and speech modification. In particular, I will thank for his contribution to the perceptual experiment presented in Chapter 7, and for the discussions on speech modification.

I also wish to thank all my friends and colleagues at the speech processing group of NTNU for a tremendously nice working environment and for memorable social events. I am also very grateful to everyone that have spared their time for attending the listening tests in this thesis and made these tests possible.

Finally, I will thank my parents for all their support, and my dear Hilde for her love, support, and encouragement. Together with our little boy Johan they have provided essential encouragement for this project.

Trondheim, October 2009
Ingmund Bjørkan

Contents

Abbreviations

AR	Autoregressive
ARMA	Autoregressive moving average
dB	Decibel
DFT	Discrete Fourier transform
Eq	Equation
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
GCI	Glottal closure instants
HMM	Hidden Markov model
HNM	Harmonic plus noise model
HNR	Harmonic to noise ratio
Hz	Hertz
IDFT	Inverse discrete Fourier transform
LF-model	Liljencrants-Fant model
LP	Linear prediction
MA	Moving average
Mfcc	Mel-frequency cepstral coefficients
MSE	Mean square error
NP	Negative peak
NTNU	Norwegian University of Science and Technology
OLA	Overlap and add
PSOLA	Pitch Synchronous Overlap and Add
ROC	Receiver Operating Characteristic
SEEVOC	Spectral Envelope Estimation Vocoder
SNR	Signal to noise ratio

STFT	Short time Fourier transform
TD	Time domain
TOBI	Tones and break indices
WLS	Weighted least squares
ZP	Zero phase

List of Symbols

A_k	Complex amplitude of the k′th sinusoidal component	37
$\underline{\mathbf{a_i}}$	Vector of estimated harmonic amplitudes for i′th frame	38
a_k	Amplitude of the k′th harmonic component	36
$a_{lp}(k)$	Linear prediction coefficients	12
C	Total cost function	23
C^c	Join cost function	23
C^t	Target cost function	23
$c[n]$	Real cepstrum	20
$\underline{\mathbf{c}}$	Vector of cepstral coefficients	53
D_*	Distance obtained by some specified distance measure *	27
d	General distance measure	26
$E(z)$	Z-transform of excitation signal	12
$e[n]$	Excitation signal	11
$\underline{\mathbf{e_k}}$	The k′th eigenvector	70
FM	Maximum voiced frequency	35
f_0	Fundamental frequency (pitch) in Hz	35
$G(z)$	Glottal filter	10
$H_s(z)$	System filter	11
$h[n]$	Harmonic signal	36
N_w	Speech frame size	13
$n_a(i)$	Discrete time analysis instant of the i′th speech frame	16
n_s	Synthesis time instant (discrete)	47
n_0	Pitch pulse onset time (discrete)	51
O_q	Opening quotient	15
P_*	Power of signal *	19
$P()$	Probability of	82
$R(z)$	Radiation filter	10
R_s	Correlation matrix	12
$r[n]$	Noise component(or alternatively stochastic component)	35
$S(n,\omega)$	Short Time Fourier Transform	20

$S(z)$	Z-transform of speech signal	12
$s[n]$	Speech signal	11
$\hat{s}[n]$	reconstructd speech signal	19
$s_i[n]$	The i'th speech frame	16
T_0	Pitch period	63
T_*	Pitch period estimate by a specific approach, *	128
t_c	Glottal closure time instant	14
t_i	The i'th target value	23
t_{zp}	Continuous time zero phase instant	100
u_i	The i'th speech unit	23
$V(z)$	Vocal tract filter	10
$w[n]$	Weighting window	37
w_j^c	Concatenation cost function weights	24
w_j^t	Target cost function weights	24
$w_a^i[n]$	Analysis weighting window for i'th speech frame	16
$w_s^i[n]$	Synthesis weighting window for i'th speech frame	17
$\eta[n]$	White noise	70
λ_k	The k'th eigenvalue	70
μ	Mean value	92
σ	Variance	92
σ_e^2	Prediction error variance	12
$\phi_e(\omega)$	The phase spectrum of the excitation signal	50
$\underline{\boldsymbol{\phi_i}}$	Vector of estimated harmonic phases for i'th frame	38
ϕ_k	Phase offset of the k'th harmonic component	36
$\phi_s(\omega)$	The system filter phase response	51
ω_k	The k'th harmonic frequency in radians per second	35
ω_l	The l'th sine wave frequency in radians per second	33
ω_0	Fundamental frequency (pitch) in radians per second	51

Chapter 1

Introduction

Text to speech synthesis is used in many applications. For example, it is used as an assistive technology for people with various disabilities. For people with vision impairment, a text to speech synthesizer can be used to read email and web pages and other electronic text material. A synthesizer can also aid people with severe speech impairment by a voice-output communication aid, which is a device that can produce synthetic speech. A speech synthesizer is also commonly used by people with dyslexia to read or to check self-written text by listening. Another application is human machine speech communication when a large vocabulary is needed or desirable. One example is telephone based news services, where a server converts news in some electronic text format into synthesized speech.

1.1 Speech synthesis

A speech synthesis system can basically be divided into two parts, a front end that analyzes text, and a back end that generates the speech waveform based on information from the front end. In principle, the front end should extract all necessary information that the back end needs to produce intelligible and natural sounding speech. In Figure 1.1, a block diagram for a standard text to speech synthesis is shown.

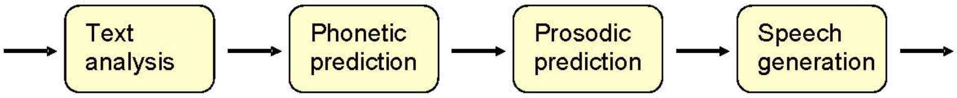


FIGURE 1.1: Text-to-speech synthesis block diagram

The first three blocks belong to the front end or preprocessor of the speech synthesizer. The first block consists of text analysis. The main tasks in this block are to divide the text into a document structure (sentences, sections, chapters etc.), normalize the text (for example dealing with abbreviations and numbers), and to do linguistic and semantic analysis.

The phonetic prediction block produces a string of sound symbols, e. g. *phonemes*, representing the phonetic content (what is to be said). Phonemes are the elementary sounds of a language, normally represented in a phonetic alphabet, e. g. SAMPA [1]. An example of an elementary sound from the English language is the phoneme /@U/ in the word boat. The phonetic string is normally obtained from the labeled text by using a phonetic lexicon, which maps written words (orthographic words) into a string of phonemes. A set of pronunciation rules are used for handling words which are not in the lexicon.

The prosodic prediction block predicts a prosodic realization ("melody" of speech) of the synthetic utterance, normally including predicted values for duration, pitch and power/loudness for each unit.

The task for the last block, speech generation, is then to use this information and produce intelligible and natural sounding speech. The focus in this thesis has been on challenges and problems related to the last block, speech generation.

1.2 Speech synthesis systems

Speech synthesis systems can basically be divided into two main classes, model based synthesis and concatenative synthesis. Model based synthesis depends on acoustical models in order to produce parametric driven speech, while concatenative synthesis concatenates segments of recorded speech. Model based synthesis can be highly intelligible, but due to the difficult and complex task of obtaining good enough speech models, the synthesized speech has so far a degraded speech quality to some extent. One example of a model based approach is HMM synthesis [2], which has a growing interest in speech synthesis research.

Concatenative synthesis can be very natural in the sense of having a speech quality close to human speech, but it may suffer from audible discontinuities at concatenation points. There are three variants of concatenative synthesis: domain specific synthesis, diphone synthesis and unit selection synthesis.

Domain specific synthesis normally concatenates words or phrases of speech, and can be used when the output of the synthesis system is limited to a small domain of utterances. The quality of domain specific synthesis can be very high, as it is possible to ensure that all the units needed for high quality synthetic speech is present in the database.

For general speech synthesis with a large vocabulary, the speech units in the synthesis system would have to be much smaller, like for example phonemes or diphones, where a diphone is a speech unit lasting from the middle of one phoneme to the middle of the next. The use of small speech units is necessary in order to be able to synthesize all possible phonetic and prosodic variation in the language with a limited database size.

Diphone synthesis speech databases consist of only one unit of each diphone occurring in the language, and the speech is commonly read with a flat pitch. During synthesis, pitch and duration modification are used to obtain a desired prosody. This method is advantageous with respect to database size and prosodic flexibility, but has some degraded naturalness due to the monotonic recording of the speech and the use of prosodic modification. The use of modification in diphone synthesis would generally be a trade-off between poor prosody and possible distortion due to modification.

Unit selection synthesis is the most popular variant of concatenative synthesis, and was first proposed by Nakajama and Hamada in 1988 [3]. Since then it has been further developed, and it is today considered as the *state of the art* in text-to speech synthesis. The work in this thesis has been focused on the speech generation module in concatenative synthesis, and in particular unit selection synthesis.

1.2.1 Unit selection synthesis

Unit selection synthesis systems use a large database of recorded speech, commonly 1-10 hours of speech. An important part of the construction of a unit selection is system is therefore database design, including manuscript design, choice of speaker, and recording procedure. When the database is recorded, the data has to be segmented and analyzed according to the needs of the preprocessor and the speech generation module. Unit selection hence consists of an analysis and a synthesis part, which is illustrated in figure 1.2.

Accurate segmentation of the speech is very important for high synthesis quality, and is therefore normally performed manually or by a manual inspection of automatic segmentation. For large volumes of speech, this is a time consuming task. The cost of generating new voices for unit selection synthesis is therefore very high. Unit selection systems can be designed to use different kinds of speech units. Typical units used in a unit se-

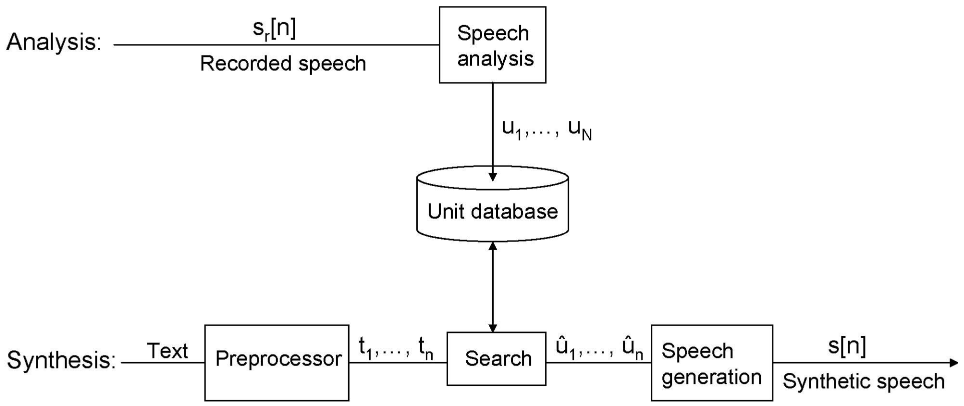


FIGURE 1.2: An example of a typical unit selection synthesis system. The recorded speech material is first analyzed in an analysis step, where the speech sentences are segmented into speech units and different speech features are extracted and saved in a speech database. During synthesis, the front end generates a set of target values for some selected features, represented by $t_1 \ldots t_n$, which is used in a search for a best possible unit sequence $\hat{u}_1 \ldots \hat{u}_n$. The speech generation block finally generates the synthesized speech from the selected unit sequence.

lection system can be phonemes, diphones, demiphones (half phonemes), demisyllables (half syllables) or possibly a combination of different types of units.

When the speech database is segmented into speech units, a further analysis of the speech data is performed. The purpose of this analysis is to extract information needed by the synthesizer in the search for an optimal unit sequence, and to extract information that can be used by the speech generation module to produce a natural and smooth speech output. Some examples of information that typically are extracted are duration (given from segmentation), voicing, fundamental frequency (pitch) for voiced speech units, energy, spectral characteristics, and the speech units' context in the sentence. Examples of such context information are phonetic context like the left and right neighboring phoneme of the speech unit, syllable context, word context, and prosodic context like toneme and stress.

The preprocessor in Figure 1.2 consists of the three first blocks in Figure 1.1, and its job is to generate a set of target values $t_{i..n}$, where the target values are predicted values for the properties of each unit that is to be synthesized. The most important property included in the target t_i, is the phonetic identity of the speech unit. Other properties in the target t_i could be phonetic and prosodic context, and possibly predicted duration, voicing, power and pitch.

During synthesis, a search for the best unit sequence $\hat{u}_i \ldots \hat{u}_n$ in the speech database is performed by minimizing a cost function, where the total cost consists of a target cost function and a concatenative cost or join cost function. The purpose of the target cost function is to minimize the difference between the target values and the selected units, while the join cost function measures how well two units can be concatenated. The selected units are finally concatenated by the speech generation module to produce synthetic speech. A more detailed description of the use of cost functions in unit selection synthesis will be given in Chapter 3.

1.3 Challenges in unit selection synthesis

A problem with unit selection synthesis systems is the large variability in quality, varying from almost perfect speech to very poor quality speech with many disturbing discontinuities. Audible discontinuities can occur when the chosen speech units are taken from different speech contexts that do not fit acoustically. Expanding the speech database can reduce this problem to some extent by increasing the probability for finding speech units that can be concatenated without causing audible discontinuities. However, this problem seems to be difficult to solve only by database expansion. One reason for this is that speech consists of a large number of rare events [4], and it would be practically impossible to have database coverage of all units that are needed.

Two approaches that can be applied to reduce audible discontinuities at unit boundaries are:

1. Improve the selection of speech units from the speech database, in order to find speech units that can be concatenated without causing audible discontinuities.

2. Apply signal processing to modify the selected speech units, in order to remove audible discontinuities. This can for example be performed by smoothing acoustic parameter trajectories across concatenation points.

For both these approaches, a better understanding of the factors that cause perceptual discontinuities is needed. For the first approach, it is important to find which features that correlate with human perception of discontinuities, in order to improve the join cost function. For the second approach, modification can be applied to those features that are known to correlate well with the human perception of discontinuities. If modification of speech units can be applied to smooth some types of bad joins, a smaller speech database could possibly be applied, leading to a lower cost of building new voices and a synthesis system with less requirements with respect to memory and possibly computational power.

A problem with using signal processing to modify speech segment is that the modification could lead to degraded speech quality or unnatural speech. Hence, the use of modification would be a trade-off between possibly degraded naturalness and audible discontinuities. A challenge is therefore also to improve the signal processing methods to produce a minimum of distortion. Modification of speech require either explicitly or implicitly the use of a speech model. This parameterization or speech modeling can be performed in the analysis stage in Figure 1.2. During synthesis, the synthesizer has to decide to what degree modification shall be applied, then modify those parameters that need to be modified, and finally synthesize the speech from the modified speech parameterization. Possible improvements can hence be made both to the modification algorithms and to the parameter estimation in the analysis stage. Central to this field of research is hence the topic of speech modeling.

Another application of speech modification is the possibility for controlling the speech prosody. Natural speech can have large variations in rhythm, stress and melody. A speech synthesizer based solely on unit selection would however be limited by the prosodic variation occurring in the speech database inventory. Hence, a successful use of modification to adjust the prosody could possibly give a better prosody control and ease the requirements on prosodic variation in the database.

Although the two strategies enumerated in this section may seem unrelated at first sight, they are highly related. For example, if a feature of speech, e. g. pitch, can be effectively smoothed across unit boundaries, the unit selection system would not have to be so selective with respect to this feature in the selection of units, and the system could put more weight to other features like for example phonetic context and spectral characteristics. This relation could also be turned the other way round. If a better selection of units is performed, the need for modification of units would be reduced, leading to a smaller risk for distortion due to modification of the speech signal.

1.4 Outline of the thesis

This thesis is divided into two parts, one part consisting of theory, Chapter 2-6, and one part consisting mainly of experiments, Chapter 7-9. However, theory related to a specific experiment is in some cases presented in relation to the experiment.

The first theory chapter concerns speech modeling, which is the basis for all remaining chapters. Speech modeling is needed in concatenative synthesis for several reasons. For example, a parameterization of the speech is needed for calculating cost functions for the selection of speech units. In addition, speech modeling is essential for the task of speech modification.

In the second theory chapter, unit selection synthesis will be described in more detail. Theory on spectral distance measures related to the first experiment chapter on join cost function design will also be described in this chapter.

Sinusoidal modeling, and then in particular a harmonic sine wave model, will be described in some detail in the third theory chapter. The harmonic model is central in this thesis, and has been applied in both the second and the third experiment chapter.

In the fourth theory chapter, Chapter 5, theory related to speech modification will be presented. The main focus of this chapter is speech modification by a harmonic model, as variants of the harmonic model have been applied in the experiments on speech modification in Chapter 9.

The fifth theory chapter, Chapter 6, will concern pitch and voicing estimation, which are important parts of the speech-processing algorithm described in the second experiment chapter. All of the approaches for speech synthesis in this thesis do however depend on pitch and voicing estimation in some way. For voiced speech, knowledge of the pitch can give an improved estimation of speech model parameters through the possibility of applying pitch synchronous analysis.

The first experiment chapter, Chapter 7, concerns the design of the join cost function in unit selection synthesis. A listening test on the perception of discontinuities in vowel joins is presented, and the results from this listening test are applied as a reference for comparing several spectral distance measures. Join cost function design based on perceptual experiments is also discussed.

Both the design of join cost functions and the application of speech modification require speech modeling and acoustic parameter extraction. This analysis step is the topic of the second experiment chapter, Chapter 8, which concerns robust pitch synchronous speech processing and pitch estimation. Robust estimation of parameters in the preprocessing step is very important for the quality of the modified speech. For example, gross errors in the pitch or the voicing estimate could lead to severely distorted modified speech.

In the third experiment chapter, Chapter 9, the topic is the use of speech modification by variants of a harmonic model. Special for the modification approaches described in this thesis is the use of the strictly pitch synchronous processing algorithm, described in Chapter 8, for extracting the speech model parameters. Both pitch and duration modification and smoothing of speech model parameters across concatenation points are discussed.

Chapter 2

Speech modeling

Two different approaches to speech modeling are the speech production approach and the sinusoidal modeling approach. The speech production approach tries to model the underlying physical process, while the sinusoidal approach in principle can approximate the signal without any assumption of the underlying physical model by decomposing the observed signal into a sum of sine wave components.

In this chapter, the speech production approach to speech modeling will be described. The model of speech production is naturally central to speech synthesis and it also affects sinusoidal modeling techniques when it comes to speech analysis and speech modification. A selection of some standard methods for speech analysis and speech representation is included in this chapter. Sinusoidal modeling will be described in its own chapter (Chapter 4).

Speech modeling is a large field, and is well documented in literature. This chapter will thus be brief, and concentrate mostly on the topics related to the experiments in this thesis. The theory is mostly derived from Huang et al. [5], Quatieri [6] and Rabiner and Schafer [7].

2.1 The source-filter model of speech

Speech production is a physical process, which starts when air from the lungs is passed through the vocal cords in the larynx. The resulting wave, referred to as the glottal wave, then enters the vocal tract. The vocal tract consists of the oral cavity from the larynx to the lips and the nasal passage that is coupled to the oral tract. The glottal wave propagates through the time varying vocal tract, including energy loss due to heat conduction and viscous friction at the vocal tract walls. In addition, there is nasal and glottal coupling and radiation at the lips [6].

A detailed model for the physical process is difficult to obtain, but simplified models can provide a good approximation in practice. A widely used model for speech production is presented in Figure 2.1 [7].

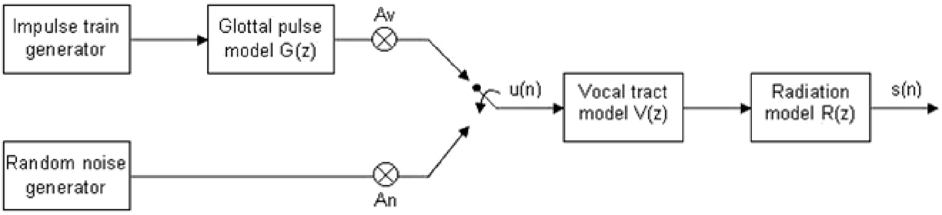


FIGURE 2.1: Speech production model

For voiced sounds the excitation is modeled as an impulse train with gain A_v . This model can be related to the (quasi) periodic mechanism of the vocal cords. When the vocal cords are closed, the air pressure builds up until the vocal cords open. When the vocal cords open the air flows into the vocal tract while the pressure drops until the vocal cords again close. Typical example of voiced sounds are vowels (e. g. /A/,/e/), nasals (e. g. /m/,/n/), and semi vowels as laterals and glides (e. g. /l/,/j/). For unvoiced sounds the vocal cords are open and the sound is created by friction of moving air against a constriction. Unvoiced sounds are generally characterized by a turbulent waveform, where the excitation can be modeled as white noise with gain A_n . Examples of unvoiced sounds are for example fricatives (e. g. /f/,/s/) and plosives (e. g. /p/,/t/,/k/). A variant of the speech production model is to allow a mixed excitation, where the excitation is a sum of both a voiced excitation and an unvoiced excitation. A mixed excitation is for example necessary to model voiced fricatives appropriately.

The glottal filter, $G(z)$, is due to that the vocal cords constrict the path from the lungs to the vocal tract. Glottal flow modeling will be described further in Section 2.1.3. The vocal tract filter, $V(z)$, models the propagation of the wave through the cavity of the vocal tract, and is the filter that give most sounds their phoneme identity. A model for the vocal tract will be briefly described in the next section. Finally, the wave is passed through the radiation filter, $R(z)$. The radiation filter describes the radiation at the lips and has been found to approximate the derivative of the volume velocity, particularly at low frequencies. If the model is assumed to be exactly a differentiator, it will introduce a high pass effect of about 6 dB/octave. The radiation does however not quite give the 6 dB/octave roll-off effect [6].

2.1.1 The lossless tube model of the vocal tract

The vocal tract filter is generally time varying, as the speaker constantly changes the vocal tract as a part of articulation. However, a widely used assumption is to assume that the vocal tract is slowly varying, so that the vocal tract characteristics can be assumed wide-sense stationary [5] during short intervals of time. This assumption is reasonable if the spectral characteristics for each phoneme are relatively slowly varying, and the analyzed time interval is short relative to the duration of the phoneme.

The frequency response due to the wave propagation through the vocal tract cavity will for a time instant t be dependent on the cross section area of the cavity $A(t)$. The lossless-tube-model [6] approximates the area, $A(t)$, as a concatenation of a set of uniform lossless tubes, where the term "lossless" refers to the assumption of no losses due to thermal and viscous effects. That is, the effect of propagation is assumed to only be influenced by the geometry of the cross section area. Assuming the area does not change over time, the solution to the differential wave equations yields an all pole model [5], where each junction of two tubes results in a reflection or equivalently a one-pole digital filter. With N tube-sections of uniform length, the lossless tube model has N/2 complex poles corresponding to the resonance frequencies of the filter, which is commonly referred to as the formants of a voiced speech sound. The position, relative energy and bandwidth of the formants are important perceptual cues for the recognition of vowels in the auditory system.

2.1.2 Linear prediction

Motivated by the lossless tube model described in the previous section, a widely used model in speech modeling is the all pole model, which is also known as autoregressive (AR) modeling [8] or linear prediction (LP) [9].

A common approach is to model the cascade of filters in Figure 2.1 by one single all pole filter [10], referred to as the system filter $H_s(z)$.

$$H_s(z) = G(z) \cdot V(z) \cdot R(z) \tag{2.1}$$

The speech signal, $s[n]$, can then be expressed as the convolution of the excitation signal, $e[n]$, and the system filter response, $h_s[n]$. When assuming an all pole model with p poles, the Z-transform [5] of the filter can be expressed as:

$$H_s(z) = \frac{S(z)}{E(z)} = \frac{1}{1 - \sum\limits_{k=1}^{p} a_{lp}(k) z^{-k}} = \prod_{k=1}^{p} \frac{z^p}{(z - \alpha_k)} \tag{2.2}$$

where $E(z)$ and $S(z)$ are the Z-transforms of the excitation and the signal respectively, $a_{lp}(k)$ are the filter coefficients of the all pole filter, and α_k are the poles of $H_s(z)$. With real filter coefficients $a_{lp}(k)$, the poles will either be real or occur in complex conjugated pairs, and it can hence be seen that there will be at most $p/2$ formants or resonances, corresponding to the pairs of complex conjugate poles in Eq. 2.2. Taking the inverse Z-transform of Eq. 2.2 results in

$$s[n] = \sum_{k=1}^{p} a_{lp}(k) s[n-k] + e[n], \tag{2.3}$$

which shows that a modeled speech sample is a weighted sum of the p previous samples, where p is the order of the model, and the excitation signal $e[n]$ can be interpreted as the model error.

It can be shown that the solution that minimizes the mean square error is given by the Yule Walker equations [8], also known as the normal equations:

$$\sum_{k=1}^{p} a_{lp}(k) R_s(i,k) = -R_s(i,0), \quad i = 1, \ldots, p, \tag{2.4}$$

where R_s is the correlation matrix estimated from the signal $s[n]$, defined as

$$R_s(i,k) = E\{s[i] \cdot s[k]\}, \tag{2.5}$$

where $E\{\}$ denotes the expectation operator [8]. The minimum variance of the model error, σ_e^2 , is referred to as the prediction error variance, and is given by:

$$\sigma_e^2 = R_s(0,0) + \sum_{k=1}^{p} a_{lp}(k) R_s(0,k) \tag{2.6}$$

The correlation matrix, R_s, is generally not known, and has to be estimated from the speech data. The estimation is performed by windowing the speech, typically using windows with duration of about 5 ms-30 ms, assuming the speech signal is wide sense stationary within the duration of the window.

Two methods for estimating the correlation matrix is referred to as the autocorrelation method and the covariance method. Assuming that the speech is a real random process, the correlation matrix in the autocorrelation method is estimated as

$$\widehat{R}_s(i,k) = \widehat{R}_s(|i-k|) = \frac{1}{N_s} \sum_{n=0}^{N_s-1-|i-k|} s[n]s[n+|i-k|], i,k \leq p, \tag{2.7}$$

where N_w is the number of data samples.The correlation matrix based on the autocorrelation method is always positive definite, hence it always gives a stable solution. The correlation matrix is also Toeplitz, and can be efficiently inverted by the Levinson-Durbin recursive algorithm [8]. However, in the autocorrelation method the speech samples are assumed to be zero outside the analysis window, which leads to an unbiased estimate of the correlation matrix [7] since the estimator averages in zeros from outside the window [7]. This problem is most prominent when using short duration windows, and it can be shown that the estimate is asymptotically unbiased as $N_s \rightarrow \infty$. The covariance method forms an unbiased estimate of the correlation function:

$$\widehat{R}_s(i,k) = \frac{1}{N_s} \sum_{n=0}^{N_s-1} s[n-i]s[n-k], i,k \leq p. \tag{2.8}$$

However, applying the covariance method for the computation of a correlation matrix [8], does not yield a Toeplitz matrix, and the correlation matrix may not be positive semidefinite. Hence, using a correlation matrix based on the covariance method may lead to an unstable AR-filter. The covariance method is therefore not so commonly used, but it may be the most appropriate method if the number of data samples is small.

2.1.3 Glottal flow modeling

Some problems arise with the use of an all pole model to model the cascade of filters in Eq. 2.1. One problem is that the cascade of filters, $H_s(z)$, may contain not only poles, but also zeros. A high order all-pole (AR-filter) can approximate a low order all-zero filter (MA-filter) if the model order p is high enough, however, at the cost of loosing some of the physical interpretation of the filter characteristics. Another problem is that in autoregressive modeling the model error, $e[n]$, is assumed to be a white noise process, while the voiced excitation in Figure 2.1 is modeled as an impulse train.

For modification of speech it could be advantageous to have separate models for the source and the filter if a modification of only the voice source is desired. An example of modification of the voice source is for example pitch modification, where the vocal tract filter normally is kept constant.

Some approaches therefore apply a variant of the source-filter model where the glottal source is separated from the vocal tract filter. A commonly used model is to use the derivative of the glottal flow as the source to the vocal tract filter [11–13].

$$s(t) = \frac{d}{dt}u(t) * v(t), \tag{2.9}$$

where $\frac{d}{dt}u(t)$ is the glottal flow derivative, and $v(t)$ is the vocal tract filter. This model can be motivated by exchanging the position of the radiation filter and the vocal tract filter in Figure 2.1. If the radiation filter is modeled as a differentiator, the exchange of filter order can be performed due to the linearity of the differentiator and the convolution operator. If using the time domain representation of the source filter model in Figure 2.1, the voiced speech can be expressed as

$$s(t) \approx \frac{d}{dt}[u(t) * v(t)] = [\frac{d}{dt}u(t)] * v(t), \tag{2.10}$$

Common parametric models for the glottal flow derivative, $\frac{d}{dt}u(t)$, are the Rosenberg model [15], the KLGOTT88 model [16], and the Liljencrants-Fant model (LF model) [17]. A typical glottal flow derivative for one glottal cycle is shown in Figure 2.2, where the glottal cycle is generated by the LF-model using typical parameters.

The physical mechanism of the vocal cords gives rise to different phases in the glottal pitch cycle, T_0. In the closed phase, T_c, the vocal cords are closed. In the open phase, T_e, the vocal cords open and air flow through the glottis, where the glottis is the opening between the vocal cords. Finally, the return phase, T_a, is the time interval from the negative peak of the glottal wave derivative to the time of complete glottal closure. It should be noticed that in Figure 2.2 the glottal closure instant is defined as the negative peak of the glottal flow derivative, corresponding to the time instant t_c. This choice is consistent with other glottal flow models as the RB model and the KLGOTT88 model. A reason for using the negative peak of the glottal flow derivative as an estimate of the "closure instant", is that this time instant might be more well defined than the instant of complete closure. In some cases, the vocal cords might not even close completely. Throughout this thesis, the negative peak of the glottal flow derivative will be referred to as the glottal closure instant. For a slightly simpler LF-model, the return phase can be included into the closed phase. Then the waveform can be effectively modeled by considering only two phases [13]. The parametric

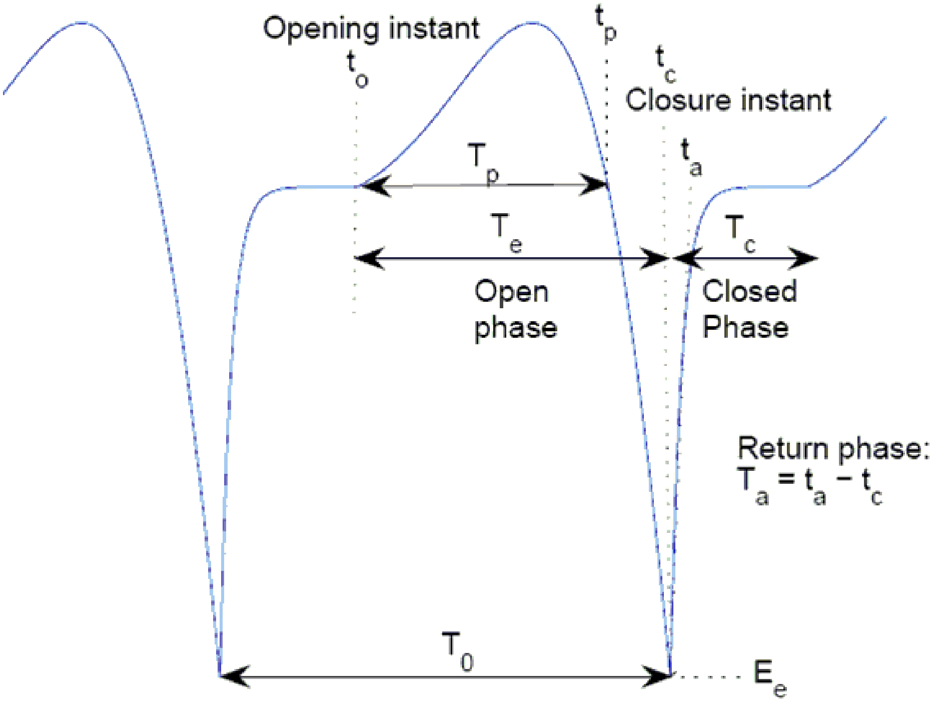


FIGURE 2.2: A typical example of a glottal flow cycle, generated by the LF-model [13]

LF model can then be expressed

$$v_{lf}(t) = \begin{cases} E_0 e^{\gamma_o(t-t_o)} sin(\omega(t-t_o)), & t_o \leq t < t_c \\ -E_1[e^{-\gamma_c(t-t_c)} - e^{\gamma_c T_c}]. & t_c \leq t < t_c + T_c \end{cases} \tag{2.11}$$

where $v_{lf}(t)$ is one cycle of the glottal flow derivative estimate, t_o is the start of the open phase, and t_c is the start of the closed phase. Three of the parameters, E_0, γ_o and ω, describe the shape of glottal flow during the open phase, while γ_c and E_1 describe the glottal flow in the return phase. Due to the required continuity of the source waveform, both E_0 and E_1 can be expressed by the value of the negative peak, E_e, at time $t = t_c$. In Vincent et al. [13] the space of shape parameters for the glottal wave derivative was chosen to be the opening quotient $O_q = \frac{T_e}{T_0}$, the return phase quotient, $O_a = \frac{T_a}{T_c}$, and the asymmetry coefficient $\alpha_m = \frac{T_p}{T_e}$.

A problem with the source-filter model in Eq. 2.9 is that it introduces the difficult problem of separating the glottal derivative and the vocal tract filter, which can be considered a blind deconvolution problem. The problem is to obtain an estimate of the vocal tract filter coefficients that has not been influenced by the source. Another problem with this model is nonlinear effects due to interaction between the source and vocal tract filter [6]. A common approach to the estimation problem is to try to eliminate most of the effect of the source, in order to obtain a better estimate of the vocal tract filter. This can for example be performed by inverse filtering [14; 12] or by a closed phase analysis [11]. Vincent et al. [13] propose a multivariable search algorithm minimizing a least square error criterion [13], while Bozkurt et al. propose a decomposition algorithm based on an all zero representation of the Z-transform of the signal [18], exploiting the phase properties of the source and the vocal tract filter.

2.2 Analysis of speech

When analyzing a speech utterance, a sequence of short speech segments is obtained by applying a sliding time window with a given window duration and time shift. The windowed speech segments, referred to as speech frames, typically have a duration of 5-30 ms, and for voiced speech the speech frames typically contain only a couple of pitch periods. The time shift between adjacent speech frame centers will be referred to as the frameshift. The speech frames can be defined to be non-overlapping or to have a certain degree of overlap, where an overlap-factor of two (window duration twice as large as the frameshift) is a widely used choice in many applications. A speech frame can be expressed as

$$s_i[n] = s[m] \cdot w_a^i[m - n_a(i)], \tag{2.12}$$

where $s_i[n]$ is the $i'th$ speech frame obtained from the speech signal $s[m]$ by using a window $w_a^i[m - n_a(i)]$. The window w_a^i, referred to as the analysis window, is nonzero only in a short time interval around the origin, where the origin for the $i'th$ speech frame is at the analysis instant $n_a(i)$. The analysis window is normally symmetric around the origin. For some applications, it is practical to let the analysis instant coincide with a sample instant, which requires an odd number of nonzero samples in the window function. The non-zero part of the speech frame $s_i[n]$ can then be expressed as

$$s_i[n] = s[m] \cdot w_a^i[m - n_a(i)], \; n_a(i) - N_i \leq m \leq n_a(i) + N_i, \tag{2.13}$$

where the center of the speech frame is defined to be at the origin ($n = 0$), corresponding to the analysis instant $m = n_a(i)$. $N(i)$ defines the frame size $(2N_i + 1)$ given by $n = -N(i) \ldots, N(i)$. The frame size can be chosen to be fixed, or it can be chosen to be dependent on i. For example, a frame size dependent on the pitch estimate is commonly applied. If the analysis window is a rectangular window, the speech frame will simply represent a piece of the speech signal centered at the analysis instant.

2.3 Overlap and add

Mainly, two different approaches for reconstruction or synthesis of a speech signal is reported in the literature. Either a sample by sample reconstruction from a parametric model of the speech, or a frame based approach where each speech frame is reconstructed from a speech model before the speech frames are combined back into one speech signal by using an overlap and add procedure [19], OLA. In this thesis, the main focus will be on the frame based approach.

Assuming we have estimated the parameters of a speech model for each speech frame $s_i[n]$ of a speech segment, each frame can be reconstructed from the speech model to yield a set of reconstructed or resynthesized speech frames $\hat{s}_i[n]$. Speech models that can be applied are for example variants of the source filter model or variants of the sinusoidal model, which will be described in Chapter 4.

The synthesized speech frames are only dependent on the analysis window functions, $w_a^i[n]$, by the parameter estimation process. However, since the speech frames can be overlapping, synthesis weighting functions have to applied for the reconstruction of the speech signal from the speech frames. The reconstructed signal, $\hat{s}[m]$, can then be expressed as

$$\hat{s}[m] = \frac{1}{\sum\limits_i w_s^i[m - n_a(i)]} \sum_i \hat{s}_i[m - n_a(i)] \cdot w_s^i[m - n_a(i)], \tag{2.14}$$

where w_s^i is the $i'th$ weighting window, which is defined to be non-zero in the interval $n = -N(i), \ldots, N(i)$. The synthesis weighting windows should be chosen so that the synthesis weighting windows fulfill:

$$\sum_i w_s^i[m - n_a(i)] = 1 \tag{2.15}$$

For example, if the speech frames are spaced with a constant frame shift N, a Hanning window or a triangular window of length $2N + 1$, corresponding to an overlap factor of 2, would fulfill Eq. 2.15, which is shown in Figure

2.3. The OLA approach will then give exact reconstruction of the original speech signal if the resynthesized speech frames are equal to the original speech frames.

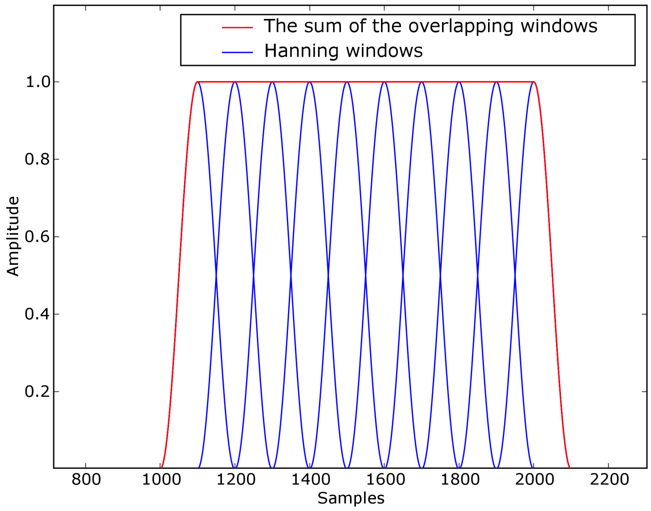


Figure 2.3: The figure shows Hanning windows spaced with a constant frame shift, $N = 100$, and each with the same window length $N_w = 201$ (in samples). The sum of the weighting windows is also shown, showing that the window requirement in Eq. 2.15 is fulfilled.

2.4 Analysis by synthesis

One approach to the estimation of parameters for a speech model is in the literature referred to as *analysis by synthesis* [5]. This approach chooses the model parameters that minimize the mean square error between the original signal and the reconstructed signal, also referred to as a mean square error criterion (MSE). The MSE criterion can be applied to each speech frame separately, or to the final synthesized signal after the overlap and add procedure, which in general can lead to slightly different results. The MSE for the i'th speech frame is defined as

$$\epsilon = \frac{1}{N_w(i)} \sum_n |s_i[n] - \hat{s}_i[n]|^2, \tag{2.16}$$

where $s_i[n]$ is the speech frame, assuming a rectangular analysis window, $\hat{s}_i[n]$ is the resynthesized signal from the speech model, and $N_w(i)$ is the frame size. A common measure for the fit of the speech model is the signal-to-noise ratio, SNR, where the noise in this case refers to the reconstruction error, $r[n] = s[n] - \hat{s}[n]$. The SNR is defined as the power of the signal divided by the power of the noise, commonly represented in the unit decibel (dB) corresponding to 10 times the base 10 logarithm.

$$SNR = 10 \cdot log_{10} \frac{P_s}{P_r}, \tag{2.17}$$

where P_r is the power of the reconstruction error, and P_s is the power of the original signal.

It should be noted that the SNR is not necessarily a good measure for the quality of resynthesized speech. For example, if an original speech signal is flipped upside down by multiplying all samples with -1, the SNR for the modified signal will be terrible, although the modified signal can not be distinguished from the original by listening. However, in general there will be some relation between the SNR and the audible quality, which can be motivated by an interpretation of the reconstruction error as additive noise to the signal.

2.5 Fourier analysis

Central to many approaches to sinusoidal analysis and speech processing is the Fourier transform. For a discrete time sequence, the Fourier transform can be approximated by the discrete Fourier transform, DFT. For a finite sequence $x[n]$, the DFT is defined as

$$X(k) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \; k = 0, \ldots, N-1, \tag{2.18}$$

where $X(k)$ is the DFT sampled at the (angular) frequencies $2\pi k/N$, n are the time samples, and N is the number of frequency bins in the DFT. The sequence $x[n]$ can then be reconstructed exactly by the inverse discrete Fourier transform, IDFT [21].

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}, \; n = 0, \ldots, N-1. \tag{2.19}$$

A single Fourier transform can not capture the spectral content of a long speech signal $s[m]$, hence the speech signal is divided into speech frames.

The Fourier transform of a windowed speech segment is referred to as the short time Fourier transform, STFT [20]. Using the same notation as in the definition of the speech frames, the discrete-time STFT can be expressed as:

$$S(n,\omega) = \sum_{m=-\infty}^{\infty} s[m]w[m-n]e^{-j\omega m}, \tag{2.20}$$

where $s[m]$ is the speech signal and $w[m-n]$ is a window function which is non-zero only in a short interval with its center at $m = n$. The STFT is hence a function of both time and frequency.

Using the notation defined in Section 2.2, the discrete time STFT of the $i'th$ speech frame can be expressed by using $n = n_a(i)$ and $w[m-n] = w_a^i[m-n]$ in Eq. 2.13. The STFT can be estimated by the discrete Fourier transform, which leads to a transform that is discrete in both time and frequency

$$S(n,k) = S(n,\omega)|_{\omega=\frac{2\pi}{N}k} = \sum_{m=-\infty}^{\infty} s[m]w[m-n]e^{-j2\pi km/N},\ k = 0,\ldots,N-1 \tag{2.21}$$

2.6 Cepstral coefficients

The complex cepstrum $\hat{x}[n]$ of a sequence $x[n]$ is defined as the IDFT of the log spectrum.

$$\hat{x}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} ln(X(\omega))e^{j\omega n}d\omega, \tag{2.22}$$

where $\hat{x}[n]$ is the complex cepstrum, $X(\omega)$ is the complex spectrum of $x[n]$, and the complex logarithm of $X(\omega)$ is defined as

$$ln(X(\omega)) = ln|X(\omega)| + j\angle(X(\omega)) \tag{2.23}$$

If the real logarithm, defined as the logarithm of the magnitude of the spectrum ($ln|X(\omega)|$) is taken in Eq. 2.22 instead of the complex logarithm, the cepstrum (or real cepstrum), $c[n]$ is obtained. The real cepstrum is widely used in speech processing, and the coefficients $c[n]$ will be referred to as the *cepstral coefficients*. If the sequence $x[n]$ is real, both the complex and the real cepstrum are also real sequences. The difference is that in the real cepstrum the phase information in the complex spectrum is discarded.

Since the real cepstrum is the inverse transform of the real part of $ln(X(\omega))$, it can be shown that the real cepstrum is the even part of the complex cepstrum [5].

$$c[n] = \frac{\hat{x}[n] + \hat{x}[-n]}{2} \tag{2.24}$$

The complex cepstrum can hence be derived from the real cepstrum when the signal $x[n]$ is a minimum phase sequence [21]. This is due to that the complex cepstrum of a minimum phase sequence can be shown to be a right sided sequence ($\hat{x}[-n] = 0$) [5]. Hence, if a speech frame can be modeled by a minimum phase rational Z-transform, the real cepstrum can be applied to effectively represent both the magnitude and the phase of the speech spectrum.

Chapter 3

Unit selection synthesis

Unit selection synthesis is based on concatenating small units of speech selected from a large database with several candidates for each unit [22]. The search for an (optimal) unit sequence, $\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n$, is normally based on a combination of two cost functions. A target cost function and a join cost function [22]. The purpose of the target cost function is to measure how well a unit u_i matches prosodic and phonetic features of the target, t_i, where the target values t_i are predicted in the synthesizer's front end. The target cost can hence be expressed as $C^t(t_i, u_i)$, where a large cost corresponds to a poor match to the target. The objective of the join cost function is to measure how well two units can be concatenated, $C^c(u_{i-1}, u_i)$. The total cost for a unit sequence $u_1, \ldots, u_n$ can then be expressed as:

$$C(u_1, \ldots, u_n, t_1 \ldots t_n) = \alpha \sum_{i=1}^{n} C^t(t_i, u_i) + (1 - \alpha) \sum_{i=1}^{n+1} C^c(u_{i-1}, u_i), \qquad (3.1)$$

where α is a relative weight between the two cost functions, and u_0 and u_{n+1} are defined to be silence segments. The task is then to find the unit sequence that minimizes the total cost

$$\widehat{u}_{i \ldots n} = \arg \min_{u_1, \ldots, u_n} C(u_1, \ldots, u_n, t_1 \ldots t_n) \qquad (3.2)$$

Considering the speech units as nodes with a cost equal to the target cost, and the join cost as the cost of the path between two nodes, the search problem can be stated as a search for the optimal (lowest cost) path through a lattice [23], where the optimal unit sequence can be found by the Viterbi algorithm [24]. The target cost and the join cost will be described in more detail below.

3.1 The join cost function

The join cost function normally consists of a weighted sum of distance measures applied to various features extracted from the speech segments around the concatenation boundary. Commonly used features are for example pitch, loudness and spectral content or acoustic similarity. It is important that the distance measures on these features correlate well with the human perception of audible discontinuities, and also that all relevant features are included in the join cost function.

A common approach is to use a linear cost function [23].

$$C^c(u_{i-1}, u_i) = \sum_j w_j^c \cdot d_j(u_{i-1}, u_i), \tag{3.3}$$

where the weights, w_j^c, are the relative importance of each distance measure d_j, applied to the j'th feature. Some of the questions to be answered for the design of join cost functions include what features to extract and how to extract them, and also what distance measures to apply. Finally, the distance measures have to be properly weighted in the join cost function in Eq. 3.3.

3.2 The target cost function

The target cost function can be defined in a similar way as the join cost function.

$$C^t(u_i, t_i) = \sum_j w_j^t \cdot d_j^t(t_i, u_i), \tag{3.4}$$

where the weights, w_j^t, are the relative importance of each distance measure d_j^t, applied to the j'th feature. Features that typically are included in the target cost function are phoneme identity, pitch, duration and power, left and right phoneme context, position in syllable and phrase, toneme and stress.

3.3 Training of the weights in the cost function

Two classical methods for training of the weights in the cost functions [22] are known as *Weight Space Search* and *Regression Training*. Both these approaches try to tune the weights in the cost function by minimizing the difference between the synthetic speech and the original speech in the speech database.

In the weight space search an original sentence from the speech database is synthesized without using any of the speech units in the original sentence, commonly referred to as copy synthesis. Then an objective distance function is applied to estimate the perceptual distance between the synthesized sentence and the original. In [22] a cepstral distance measure was applied for this objective distance function. This process can be repeated for a range of weight sets and multiple utterances. The best weight set is chosen as the one that minimizes the measured error over all utterances. The limitations with this approach is that it is very computationally demanding, and that its appropriateness depend on the quality of the objective distance function(s) that are applied.

In the regression training approach, the weights for the join and target cost function are trained separately. In this approach, the join cost weights are trained from perceptual experiments on speech unit joins or by hand tuning. The weights for the target cost function are then obtained by using an objective distance function and a multiple linear regression approach. For each unit in the database, the n nearest (for example $n = 20$) units with respect to the objective distance function are used in the regression, where the target weights for this type of unit are determined by minimizing the difference between the target cost function and the objective distance function. The quality of the output when using the two training methods is reported to be similar, but the regression method is more computationally effective [22].

3.4 Spectral distance measures

The join cost function consists of one or several distance measures that are intended to measure the spectral difference, or distance, between the speech segments at each side of a join.

The distance measures in the join cost function should account for the properties of the human auditory system, in order to make the distance measures more correlated with human perception [25; 26]. For example, the perception of pitch differences has been found to be approximately related to a logarithmic scale, while the perception of loudness can be approximated by a cubic root compression, and is also frequency dependent [25].

Normally, pitch and loudness difference are treated separately from spectral distance measures, although the difference in pitch and loudness also can be considered as spectral differences in strict sense. The notion of spectral distance measures will therefore in this thesis be restricted to distance measures that measure other spectral differences than the difference in pitch and loudness. That is, the distance between power normalized spectrum envelopes. Different spectral parameterizations and different weighting of frequency bands lead to different spectral distance measures, which would have different correlation to the human perception of spectral discontinuities.

Normally, a distance measure $d(x, y)$ is restricted to have some desirable properties [27].

1. Symmetry, $d(x, y) = d(y, x)$
2. Positive definiteness, $d(x, y) > 0, x \neq y$ and $d(x, y) = 0$ when $x = y$.

A third desirable property for a distance measure is that it fulfils the triangular inequality $d(x, y) \leq d(x, z) + d(z, y)$. However, the spectral distance measures described in this chapter have not been restricted to fulfill this property. It can also be discussed whether the property of symmetry is necessary. Although symmetry is desirable, the most important property for a spectral distance measure is a high correlation with human perception. Hence, the symmetry property has not been an absolute requirement either. In the next section, the spectral distance measures applied in the experiments in Section 7 will be described briefly.

3.4.1 Log spectral magnitude distance

One class of spectral distance measures is the class of log spectral magnitude distance measures [27]. Defining two spectra $P(\omega)$ and $Q(\omega)$, this class of distance measures consist of the set of L_p norms, defined by $(d_p)^p$, where

$$(d_p)^p = \frac{1}{2\pi} \int_{-\pi}^{\pi} |ln(P(\omega)) - ln(Q(\omega))|^p \, d\omega \tag{3.5}$$

For $p = 1$, the mean absolute log spectral measure is obtained, while $p = 2$ corresponds to the root mean square (rms) log spectral measure. For the limiting case as p approaches infinity the peak log spectral difference is obtained.

In [27], it is shown that the Euclidian distance of cepstral coefficients, calculated from AR-spectra[1] equals the rms log spectral distance measure,

$$(d_2)^2 = \sum_{k=-\infty}^{\infty} (c_P(k) - c_Q(k))^2 \tag{3.6}$$

[1] Spectra obtained using an AR model (all pole model) for the speech frames, where the parameters can be estimated by linear prediction (Section 2.1.2)

As Eq. 3.6 introduces an infinite sum, an approximation to the rms log spectral measure must be obtained by using a fixed number L of cepstral coefficients

$$D_{cep} = (c_P(0) - c_Q(0))^2 + 2\sum_{k=1}^{M}(c_P(k) - c_Q(k))^2, \tag{3.7}$$

where D_{cep} denote the cepstral distance measure. The number of cepstral coefficients defines the degree of smoothing of the estimated spectra P and Q. It can be shown that the cepstral measure approximates the rms log spectral measure from below, and asymptotically equals the rms log spectral measure when $M \to \infty$ [27].

Another distance measure related to this class of spectral distances is the symmetric likelihood ratio [27]. Given two AR-spectra $P(\omega)$ and $Q(\omega)$, the non-symmetric version of this distance measure can be expressed as:

$$\delta/\alpha - 1 = \int_{-\pi}^{\pi} \frac{|P(\omega) - Q(\omega)|^2}{Q(\omega)} d\omega \tag{3.8}$$

Using $P(\omega)$ as a reference instead of $Q(\omega)$ by switching P and Q in the above equation, will give slightly different distance, $\delta'/\alpha' - 1$, as long as P is different from Q. The symmetric likelihood ratio is therefore defined as the mean of these two non-symmetric distances.

$$D_{LR} = \frac{\delta/\alpha + \delta'/\alpha'}{2} - 1, \tag{3.9}$$

In [27] it is shown how this measure can be computed from the linear prediction coefficients related to P and Q, and the autocorrelation coefficients of the two speech segments related to P and Q. It is also shown that this measure approximates the rms log spectral measure from above.

3.4.2 Symmetrical Kullback Leibler Distance

A distance measure similar to the rms log spectral distance measure is the symmetric Kullback-Leibler distance [28]. Given two AR spectra $P(\omega)$ and $Q(\omega)$ the measure is defined as:

$$D_{skl} = \frac{1}{4\pi} \int_{0}^{2\pi} (P(\omega) - Q(\omega))\, ln \left(\frac{P(\omega)}{Q(\omega)}\right) d\omega \tag{3.10}$$

In [28] it is shown how this distance measure can be computed exactly from AR coefficients:

$$D_{SKL} = \sum_{l=1}^{p} \left(\rho'_{P,l} ln \frac{B(\alpha_l^{-1})}{A(\alpha_l^{-1})} + \rho'_{Q,l} ln \frac{A(\beta_l^{-1})}{B(\beta_l^{-1})} \right), \quad (3.11)$$

where A and B represent the estimated denominator polynomials of the all pole Z-transforms corresponding to P and Q respectively, α are the roots of $A(z)$, β are the roots of $B(z)$, and ρ'_P and ρ'_Q are normalized residues corresponding to a partial fraction decomposition of the Z-transforms. A detailed description of this approach is found in [28].

3.4.3 Mel frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCC) [5], is a parameterization of the speech spectrum widely used in speech recognition. The task of speech recognition and the task of detecting audible discontinuities are quite different problems. Both tasks do however depend on measuring the spectral content of speech. The use of an auditory based frequency scale like the Mel-scale or the Bark-scale [5], would yield a better correlation with human perception. The Mel-scale is approximately linear up to 1 kHz and logarithmic at higher frequencies,

$$Mel(f) = 2595 \cdot log_{10}(1 + \frac{f}{700}). \quad (3.12)$$

The Mel-frequency cepstral coefficients are obtained by first obtaining frequency band energies from a filter bank analysis of the estimated STFT by using uniform triangular filters equally spaced on the Mel-scale. Finally, a discrete cosine transform is applied to the logarithm of the filter bank energies. A widely used distance measure is then to use the Euclidian distance of the MFCC coefficients, resulting in a distance measure similar to Eq. 3.6. In a join cost function setting, it could be desired to treat energy as a feature on its own, which would make it intuitive to leave out the zero'th coefficient related to energy, in order to make the features in the join cost function more uncorrelated. The MFCC distance measure can in this case be calculated as

$$D_{mfcc} = \sqrt{\sum_{k=1}^{M} (\tilde{c}_P(k) - \tilde{c}_Q(k))^2}, \quad (3.13)$$

where M is the number of coefficients, and $\tilde{c}_P(k)$ and $\tilde{c}_Q(k)$ are the MFCC coefficients for the power spectra $P(\omega)$ and $Q(\omega)$ respectively.

3.4.4 Cross-correlation

The cross-correlation of two short time segments will have a high peak if the waveforms in the segments are similar. Hence, it can possibly be applied as a time domain measure of spectral distortion [29].

The cross-correlation is dependent on both the phase mismatch and the power mismatch. Hence, the segments should be power normalized and phase aligned in order to be a measure of spectral difference. The cross-correlation distance measure is therefore defined as the Euclidian distance between two power normalized and pitch synchronous speech frames.

$$D_{psc} = \sum_{n=1}^{N_w} (s_L[n] - s_R[n])^2 \tag{3.14}$$

where $s_L[n]$ and $s_R[n]$ are power normalized speech frames to the left and to the right of the concatenation point respectively. Both frames must have the same duration, represented by the number of samples, N_w. Writing out this expression show that this measure is equal to $2 \cdot (1 - S_{s_L s_R}(0))$, where $S_{s_L s_R}(0)$ is the cross-correlation of $s_L[n]$ and $s_R[n]$ at lag zero.

Chapter 4

Sinusoidal modeling

Several approaches have been proposed for speech modeling by sinusoidal models for speech synthesis. Examples are the phase vocoder (Flanagan and Golden (1966) [30]) , the harmonic model (Almeida and Silva (1984) [31]), general sinusoidal models (e.g. Hedelin (1981) [32], McAulay and Quatieri (1986) [33] and George and Smith (1997) [34]), and hybrid sinusoidal/stochastic models based on a stochastic/deterministic decomposition of the speech (e.g. Serra (1990) [35], Stylianou (1993) [36], and Griffin and Lim (1988) [37]).

The sine wave models have been successfully applied to low bit rate speech coding, and have also been shown to be promising for the application of speech modification. For example, the methods proposed by McAulay and Quatieri [38], George and Smith [34] and Stylianou [39], are reported to facilitate high quality speech modification.

The focus in this thesis is the harmonic model. The harmonic model is a sine wave model where the sine wave frequencies are the integer multiples of the fundamental frequency. As the harmonic model is a special case of a general sine wave model, the general sine wave model is described first in this chapter, before the harmonic model is described in more detail.

4.1 The general sinusoidal model

A decomposition of a signal into a series of sine wave components can be applied to any signal without any assumption of how the underlying signal was generated. Specifically, if the signal is exactly periodic the signal can be reconstructed by a sum of harmonic sine wave components, where the harmonic frequencies are defined as integer multiples of the fundamental frequency. In the case of a general signal, the fundamental frequency can be defined to be the lowest frequency of this harmonic series. Note that for speech signals the fundamental frequency represents a physical quantity, the frequency of glottal closure instants. The fundamental frequency is commonly referred to as the pitch of the speech. Therefore, pitch and fundamental frequency is used for the same term also in this thesis. A definition of pitch is presented in Section 6.1.

Voiced speech is known to be "almost" periodic or quasiperiodic, hence the spectrum will have peaks at or close to the harmonic frequencies. However, there are also aharmonic components in the speech, motivating the general sinusoidal model. For example, aharmonic components in the spectrum can be due to aspiration or fricative noise or simply due to a time varying fundamental frequency.

In the general sinusoidal model the speech is modeled by a sum of time varying sine wave components [1] [6], where most of the sine wave frequencies typically are close to the harmonic frequencies.

$$\hat{s}(t) = \sum_{l=1}^{L} a_l(t) \cdot \cos(\theta_l(t)), \tag{4.1}$$

where $\hat{s}(t)$ is the modeled speech signal, $a_l(t)$ are the sine wave amplitudes, $\theta_l(t)$ are time varying sine wave phases, and L is the number of sine wave components. The time varying phases $\theta_l(t)$ can be expressed as

$$\theta_l(t) = \int_0^t \omega_l(\tau) d\tau + \phi_l, \tag{4.2}$$

where $\omega_l(t)$ are time varying sine wave frequencies and ϕ_l are the phase offsets at time $t = 0$.

A frame-based algorithm has to be applied for estimating the parameters of the general sine wave model. The speech can then be reconstructed (synthesized) from the model parameters either by reconstructing each speech frame separately and using the OLA method described in Section 2.3, or by reconstructing the speech sample by sample from the discrete version of Eq. 4.1. For the latter approach, the model parameters have to be estimated for all samples n, which can be achieved by interpolating the sine wave parameters estimated for each frame in time. The sine wave amplitudes and frequencies can simply be linearly interpolated between the frame centers. The phases are more difficult to interpolate due to the non-linear relationship with the time varying frequency and because the phase estimates are

[1] It should be noted that the term sine wave component is used on a general basis to denote a sine wave with any phase, hence also cosines will be referred to as sine waves.

obtained modulo 2π and have to be unwrapped to yield smooth phase tracks. McAulay and Quatieri's approach to the phase interpolation problem is to define a phase interpolation function that is a cubic polynomial and to use continuity constraints at the frame borders in addition to a maximally smooth phase derivative constraint to solve for the unwrapping factor [33].

In a frame based synthesis approach, the parameters can be assumed constant within the speech frame, and then the sine wave model in Eq. 4.1, using discrete time, simplifies to

$$\hat{s}_i[n] = \sum_{l=1}^{L} a_l \cdot \cos(\omega_l \cdot n + \phi_l), \tag{4.3}$$

where $s_i[n]$ is a speech frame, a_l are the sine wave amplitudes, ω_l are the sine wave frequencies and ϕ_l are the phases. In the case of constant parameters within the speech frame the time variation is introduced by the OLA method, although a synthesis equation with time varying parameters could be applied also when using a frame based synthesis.

4.1.1 Estimation of sine wave parameters by peak picking

Different approaches can be applied for the estimation of sine wave frequencies, amplitudes and phases in Eq. 4.3. A widely used approach for determining a set of sine wave frequencies is to search for the most prominent peaks in the Short Time Fourier Transform, STFT. Two other approaches related to the estimation of sine wave parameters are the deterministic least squares approach and the ESPRIT algorithm. Deterministic least squares estimation can be applied to estimate sine wave amplitudes and phases when the sine frequencies are determined. This approach is described in relation to the estimation of amplitudes and phases for a harmonic model in Section 4.2.2. The ESPRIT algorithm, which is described in relation to pitch estimation in Chapter 6, is also an alternative for determining the sine wave frequencies.

When the speech is quasiperiodic, the magnitude of the STFT will have peaks close to the harmonic frequencies. An approach, proposed by McAulay and Quatieri [33], is therefore to apply a peak-picking algorithm of the STFT by applying a high resolution DFT [33]. The peaks are selected by detecting all values in the STFT that are greater than its two nearest neighbors, and also above a specified threshold computed from the value of the maximum peak. The amplitudes and phases are estimated by evaluating the estimated STFT at the chosen frequencies. Since the STFT of a rectangular window has a poor side lobe structure, a normalized Hamming window is used, however, at the expense of broadening the main lobes. The duration of the analysis frame is therefore made at least 2.5 times the average pitch period, in order to maintain frequency resolution. After the peak picking procedure, a frame-to-frame peak matching is performed. In order to account for rapid changes of the spectral peaks, the concept of "birth" and "death" of sine components is introduced, and a rule-based algorithm tries to match neighboring sinusoidal components across frames to obtain continuous frequency tracks along the time axis.

4.1.2 Decomposition into a stochastic and deterministic part

A possible weakness with a sinusoidal model is that a voiced speech frame in general consists of both (quasi)periodic components and stochastic (noise) components. The sinusoidal model can however implicitly model the stochastic component in a speech frame by a superposition of high frequency sinusoidal components [6]. This modeling of the high frequency content is adequate for reconstructing speech without audible distortion if enough sine wave components are used. This can also be motivated by the fact that a DFT can be used to model any signal exactly if the number of frequency bins in the DFT, or alternatively the number of sine waves, is equal to (or greater than) the length of the window.

A problem with the sinusoidal model is however that the stochastic component will be mixed with the deterministic component in a complex way. Applying pitch and time scale modification to the sine-based models can lead to stretching or compression of the stochastic component, which can result in an unwanted tonality [6]. Many approaches therefore try to model the speech as a sum of a periodic deterministic component and a non-periodic stochastic component, in order to obtain a better model of the stochastic component. In [35] and [40] the signal is divided into a deterministic and a stochastic part over the whole band, while for the multiband excitation vocoder [37], the signal is divided into a sum of outputs of several frequency bands, where a voicing decision is made for each frequency band. In the harmonic plus noise model [41], the two components are assumed to fall in separate time varying frequency bands.

Typically, the stochastic part is estimated by a white noise input filtered by a linear time varying filter, which will be described in relation to the harmonic plus noise model in the next section. A limitation of this representation is that not all aharmonic sounds are appropriately modeled by such a model. For example, a sharp attack at a vowel onset or a plosive may be better represented by sum of short duration coherent sine waves, or alternatively the input of an impulse driven system [6]. Another problem can be to make the stochastic and deterministic parts integrate in such a way that the signal is perceived as only one signal.

4.2 The harmonic plus noise model

The harmonic model is a special case of a sinusoidal model, where the sine wave frequencies by definition are the harmonic frequencies. The harmonic model has both coding and computational advantages, as it is only necessary to estimate one sine wave frequency, the fundamental frequency. For a frame-based approach, the fundamental frequency, f_0, is normally assumed to be constant within the speech frame, resulting in a specific case of Eq. 4.3 where the sine wave frequencies $f_l = f_k = k \cdot f_0 = 2\pi k\omega_0$. The heuristic frame to frame peak matching is then unnecessary, as each harmonic of the fundamental can be interpreted as a frequency track.

The harmonic plus noise model (HNM) is based on a decomposition of the speech signal $s[n]$

$$s[n] = h[n] + r[n], \tag{4.4}$$

where $h[n]$ is a harmonic component and $r[n]$ is a stochastic (noise) component. The decomposition is performed for each speech frame by dividing the speech spectrum into two separate frequency bands (high and low frequencies) by estimating a time varying split frequency referred to as the maximum voiced frequency, FM.

The parameters of the harmonic plus noise model are as in the sinusoidal model estimated by using a frame-based speech-processing algorithm. In the algorithm used by Stylianou [36], the duration of the speech frames and the frame shift were set to be pitch adaptive, i.e. the frame shift and the frame size depend on an initial pitch estimate. The duration of the overlapping speech frames were set to be two pitch periods with a frame shift of one period. This means that the estimation algorithm is pitch synchronous and that a relatively high time resolution is achieved, at least for high pitched speakers. Short duration analysis windows will reduce the averaging due to aharmonic components in the speech signal, and will hence improve the signal-to-noise ratio in the modeling of each speech frame.

In the next section, the topic will be the estimation and synthesis of the harmonic part, while the estimation and synthesis of the noise part will be discussed in Section 4.2.3. It should be noticed that the harmonic part equals the harmonic model when the whole frequency band is modeled by sinusoidal components, $FM = Fs/2$, and $r[n] = 0$. As a specific case of a sinusoidal model, the harmonic model is an alternative for modeling the speech frames alone as discussed in Section 4.1.2.

4.2.1 The harmonic model

Using the same speech frame notation as in Section 2.4, the analysis instant, $n_a(i)$, of the i'th speech frame is defined to be at the center of the speech frame and at a whole sample. This would require an odd frame size. The frame size, N_w, should be pitch adaptive for optimal estimation, and if a frame size duration of approximately two periods is desired, a practical choice is to use $N_w = 2N(i) + 1$, where $N(i)$ is the estimated pitch period of the $i'th$ frame rounded to a whole number of samples.

$$N(i) = [Fs/f_0(i)], \tag{4.5}$$

where [ ] denotes a rounding operation to the nearest integer. Omitting frame notation (dependency on i) for simplicity, the harmonic part, $h[n]$, can be written as

$$h[n] = \sum_{k=0}^{K} a_k \cdot \cos(k(2\pi f_0)(n - n_a) + \phi_k), \quad n_a - N \leq n \leq n_a + N, \tag{4.6}$$

where a_k are the harmonic amplitudes, ϕ_k are the harmonic phases, f_0 is the fundamental frequency, n_a is the speech frame center, N defines the frame size, and K is the number of harmonic components. The highest harmonic frequency, $K \cdot f_0$, is defined to be below the maximum voiced frequency FM.

$$K(f_0) = \lfloor FM/f_0 \rfloor, \tag{4.7}$$

where $\lfloor\ \rfloor$ denote a rounding operation to the nearest integer being smaller than the argument.

Notice that if a frame based approach is applied, the parameters of the harmonic model which in general are functions of time, have to be estimated only for each analysis instant $n_a(i)$. The inclusion of the constant offset term (k=0) in Eq. 4.6 is more a matter of implementation, and this term could be omitted without any significant changes in the parameter estimates of the harmonic model if the speech signal has zero mean.

If the noise part is set to zero ($r(n) = 0$) and $FM = Fs/2$, the classical full band harmonic model is obtained. Then the number of harmonic components K for the i'th speech frame depends on the sample frequency and the fundamental frequency f_0. Equation 4.6 will be referred to as the harmonic model equation or simply as the harmonic model.

4.2.2 Estimation of harmonic amplitudes and phases

Two approaches for the estimation of harmonic amplitudes and phases for a voiced speech frame will be described. The first approach applies a weighted least squares criterion in the time domain [36; 42]. The second approach applies the DFT, as in the estimation of parameters described for the general sinusoidal model.

4.2.2.1 Weighted least squares estimation

The weighted least squares estimation, referred to as WLS-estimation, is based on minimizing the weighted least square reconstruction error of the harmonic model. The WLS error function is defined as

$$\epsilon = \sum_{n_a-N}^{n_a+N} w^2[n](s_i[n] - h_i[n])^2, \tag{4.8}$$

where $w[n]$ is a weighting window, typically chosen to be a Hamming window [5], $s_i[n]$ is the original speech frame, and $h_i[n]$ is the reconstructed speech frame by the harmonic model. If the weighting window $w[n]$ is chosen to be a Hamming window, the least square error criterion will lead to a better fit of the signal at the center of the speech frame.

Given an estimated pitch $\hat{f}_0(i)$, the amplitudes and phases in the harmonic model can be estimated by solving the over-determined system of 2N+1 equations and $2 \cdot K + 1$ unknowns by deterministic least squares estimation. For convenience, we write the harmonic model equation using complex exponentials, and omit frame notation (dependency on i).

$$h[n] = \sum_{k=-K}^{K} A_k \cdot e^{j(\cdot 2\pi f_0)k(n-n_a)}, n_a - N \le n \le n_a + N, \tag{4.9}$$

where $A_{-k} = A_k$ are complex amplitudes. Switching to matrix notation, the harmonic part, $\underline{\mathbf{h}} = h[n]$, can be written as

$$\underline{\mathbf{h}} = \mathbf{B}\underline{\mathbf{x}}, \tag{4.10}$$

where $\underline{\mathbf{x}} = [A_{-K}, A_{-K-1}, \ldots, A_K]$ and $\mathbf{B}$ is a $(2N+1) \times (2K+1)$ Toeplitz matrix defined as

$$\mathbf{B} = \begin{bmatrix} \underline{\mathbf{b}}_{-K} & \vdots & \underline{\mathbf{b}}_{-K+1} & \vdots & \vdots & \ldots & \vdots & \underline{\mathbf{b}}_K \end{bmatrix} \tag{4.11}$$

where $\underline{\mathbf{b}}_k$ is a $(2N+1) \times 1$ vector defined as

$$\underline{\mathbf{b}}_k^T = \begin{bmatrix} e^{j2\pi k f_0(-N)} & e^{j2\pi k f_0(-N+1)} & e^{j2\pi k f_0(-N+2)} & \ldots & e^{j2\pi k f_0(N)} \end{bmatrix} \tag{4.12}$$

The weighted least squares criterion in Eq. 4.8 can then be expressed in matrix notation as

$$\min_{\underline{\mathbf{x}}} ||\mathbf{W}(\underline{\mathbf{s}} - \mathbf{B}\underline{\mathbf{x}})||^2, \tag{4.13}$$

where $\underline{\mathbf{s}}$ is the speech frame $s_i[n]$ and $\mathbf{W}$ is a $(2N+1) \times (2N+1)$ diagonal matrix with the weight vector (e.g. Hamming window) $\underline{\mathbf{w}}$ on the diagonal.

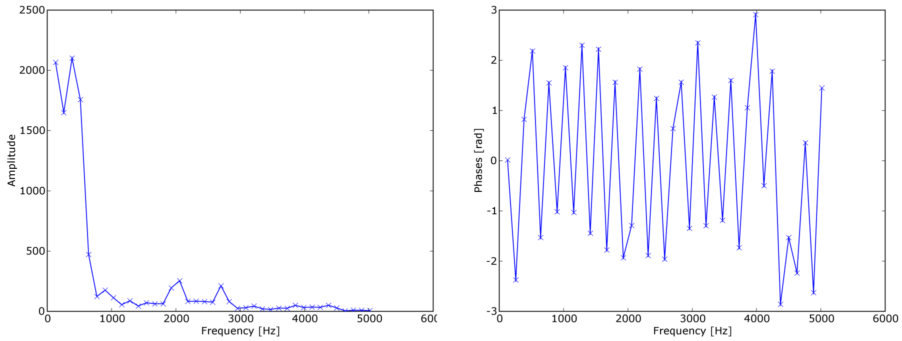
$$diag(\mathbf{W}) = \left[w[-N] \ \vdots \ w[n+1] \ \vdots \ \ldots \ w[N] \right] \tag{4.14}$$

The least squares solution [5] also known as the *pseudoinverse* is then

$$\underline{\mathbf{x}} = (\mathbf{B^T W^T W B})^{-1} \mathbf{B^T W^T W} \underline{\mathbf{s}} \tag{4.15}$$

The real amplitudes and phases in Eq. 4.6 can be obtained from the complex amplitudes by $\underline{\mathbf{a}} = 2 \cdot |\underline{\mathbf{x}}|$, and $\underline{\boldsymbol{\phi}} = \angle \underline{\mathbf{x}}$.

Harmonic amplitudes and phases estimated from a speech frame from a male /e:/-sound using the WLS-estimation are shown in Figure 4.1. Formants (resonance frequencies) in the amplitude spectrum can be observed at about 500 Hz, 2000 Hz and 2700 Hz.



(a) Amplitudes

(b) Phases

FIGURE 4.1: Estimated amplitudes (to the left) and phases (to the right) for a speech frame from an /e:/ sound of a male speaker using the WLS-estimation with a Hamming weighting window. A full band harmonic model on 16 kHz sampled speech was applied in this example, although only estimated amplitudes and phases below 5000 Hz are displayed in the figure.

The weighting window in the WLS-criterion [43] is important for a high quality of the resynthesized speech by a harmonic model. A weighting window like for example the Hamming window would lead to less error at the center of the frame and larger error at the frame edges, as shown in Figure 4.2.

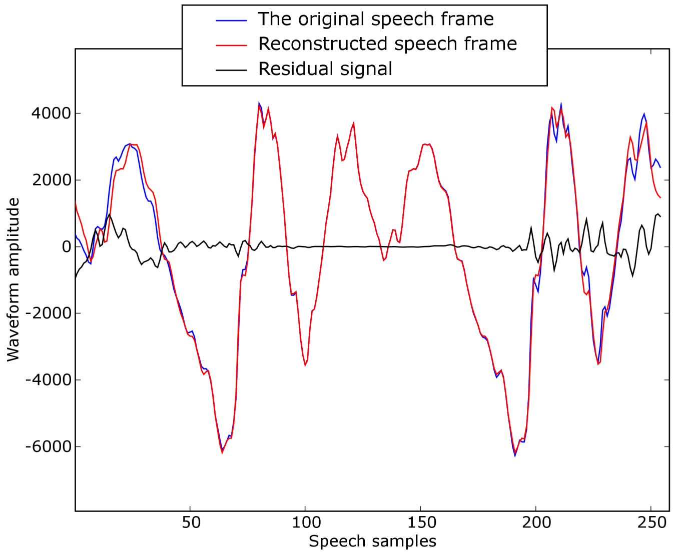


FIGURE 4.2: A speech frame from an /e:/ sound of a male speaker, the reconstructed speech frame from the harmonic model using WLS parameter estimation, and the residual error (difference) signal.

The residual signal will have larger error where the adjacent speech frames overlap, while it is almost zero at the speech frame centers. An example of a reconstructed voiced segment is shown in Figure 4.3.

The total SNR for an OLA resynthesized segment would generally be significantly lower when using the WLS-estimation with a Hamming weighting window due to the small error at each speech frame center where the degree of overlapping is low. When a Hamming weighting window is applied in the WLS-estimation, there is no audible distortion in the reconstructed speech signal. A rectangular weighting window (no weighting) generally yields a slightly noisy reconstruction of the speech.

When a Hamming weighting window (or a similar window) is applied in the estimation and pitch synchronous analysis is assumed, the WLS estimation can be interpreted as going towards a one period analysis with smoothing constraints to the adjacent pitch cycles. A motivation for this interpretation is that an one period analysis (speech frames with one period duration) yields a perfect reconstruction of the whole speech signal as an IDFT would give a perfect reconstruction of each frame. However,

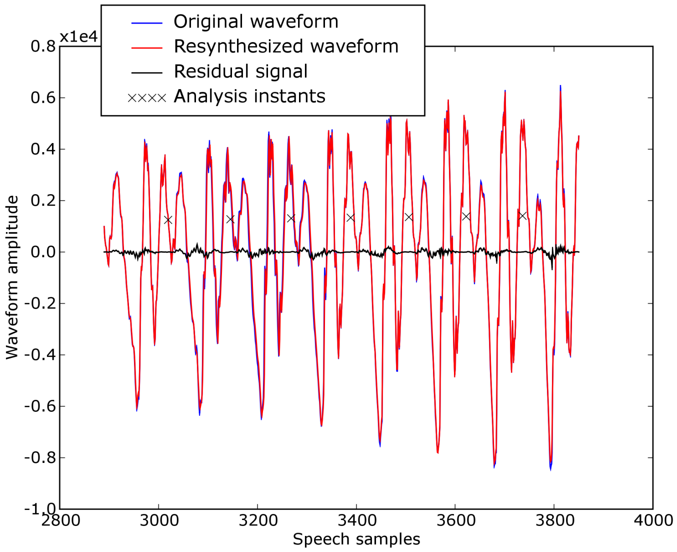


FIGURE 4.3: A speech segment from an /e:/ sound of a male speaker and a reconstructed speech segment reconstructed by the OLA method and the harmonic model using WLS parameter estimation. The residual signal (error/difference) is also plotted and the overall SNR for this segment was calculated to be 31.88 dB. The analysis instants (frame centers) are marked with 'x'.

it would then be difficult to concatenate modified speech frames without getting audible discontinuities at speech frame boundaries. When using speech frames with a duration of two periods, a smooth concatenation of modified speech frames can be achieved by the overlap and add method.

4.2.2.2 Using the DFT to estimate the harmonics

The short time Fourier transform of a harmonic signal $h[n]$ using a window function $w_a[n]$, can be expressed as

$$S_h(\omega) = \sum_{k=-K}^{K} |A_k| e^{j\phi_k} W_a(\omega - k\omega_0), \tag{4.16}$$

where $W(\omega - k\omega_0)$ is the Fourier Transform of the window function, ω_0 is the fundamental frequency, and S_h is the Fourier transform of the harmonic signal. The mean square error between the speech signal and the harmonic signal can be expressed in the frequency domain as

$$E(\omega) = |S(\omega) - S_h(\omega)|^2, \tag{4.17}$$

where $E(\omega)$ is the square frequency domain error between the signal and the harmonic signal. If the main lobes do not overlap, the amplitudes and phases that minimize Eq. 4.17 is

$$\phi_k = \angle S(k\omega_0), \tag{4.18}$$

and

$$|A_k| = \frac{|S(k\omega_0)|}{W(0)}. \tag{4.19}$$

The amplitudes and phases can hence be estimated by using the discrete Fourier transform, defined in Eq. 2.18.

If a rectangular analysis window of size $2N + 1$ is applied, the Fourier Transform of the window is

$$W_a(\omega) = \frac{sin((2N+1)\omega/2)}{sin(\omega/2)}, \tag{4.20}$$

where $W_a(\omega)$ will be zero when ω equals multiples of $2\pi/(2N+1)$. The frequency difference between two harmonic components in radians is $2\pi/T_0$, where T_0 is the period. The main lobes will hence not overlap if the rectangular window is at least two pitch periods [5]. This can be seen by setting $N = T_0$ in Eq. 4.20 and observing that

$$\frac{2\pi}{2T_0+1} < \frac{\pi}{T_0} \tag{4.21}$$

Although the main lobes do not overlap when the analysis window is at least two periods it should be noticed that the DFT method suffers from spectral leakage due to the window. Longer duration windows and the use of Hamming or Hanning windows in the estimation could reduce this problem. For example, in the estimation described for the general sinusoidal model the speech frames were obtained by using a Hamming window with 2,5 period duration. If a higher time resolution is desired, e.g. analysis window duration of two periods, WLS estimation would be a more appropriate choice for the estimation of parameters for the harmonic model [44].

4.2.3 Synthesis by a harmonic plus noise model

In the harmonic plus noise model, the harmonic and the noise parts are synthesized separately, and finally added together. The harmonic part of the speech can be synthesized either sample by sample or frame by frame using the harmonic model equation similarly as in the general sinusoidal model. In the frame-based approach the OLA method, described in Section 2.3, is finally applied to produce the synthetic speech. The OLA method is often preferred as it is less computationally intensive and avoids the explicit need for parameter interpolation across speech frames.

4.2.3.1 Estimation and synthesis of the noise part

When the harmonic part of a speech frame is estimated, the noise part in the HNM model is estimated by modeling the residual signal [36], $r_i[n] = s_i[n] - h_i[n]$. This can be interpreted as an analysis by synthesis approach, since the noise component is modeled as a fit to the residual signal.

One approach to the estimation of parameters for the noise component is to assume the noise part can be modeled as filtered white noise as in the speech production model. For example, if an AR model is assumed for the vocal tract filter, linear prediction can be applied to estimate the parameters of the model. However, if applying WLS-estimation for the estimation of the harmonic part, the residual signal will have almost no error at the center of the frame and higher error at the edges of each frame, see Figure 4.2, and hence the low frequency components in the residual signal would bias the estimated AR-model. When using WLS estimation, it is therefore better to estimate the parameters of the AR-model from the original speech signal [36]. A high pass filter with cut off frequency equal to the maximum voiced frequency $FM(i)$ is applied in a postfiltering step to reduce low frequency noise components [36].

When using a harmonic plus noise model it is very important that the noise part is synchronized with the harmonic part. If not, the noise part may not perceptually integrate with the harmonic part, and the result may be perceived as noisy speech. To alleviate this problem, a time domain energy envelope function is applied to the synthetic noise signal to provide modulated filtered white noise. In [36] it is proposed to use a fixed triangular like time domain envelope, although the time domain envelope in general was found to be speaker dependent [36].

The noise component for a speech frame, $r_i[n]$ can then be expressed as

$$r_i[n] = \gamma_i[n] \cdot (v_i[n] * \eta_i[n] * m_{hp}[n]), \tag{4.22}$$

where $v_i[n]$ is the estimated all-pole vocal tract filter impulse response, $\eta_i[n]$ is unit variance white noise, $m_{hp}[n]$ denotes the high pass filter, and $\gamma_i[n]$ is a time domain energy envelope function. Another approach to make the noise part integrate better with the harmonic part is to apply a time varying noise power [45]. A modified time domain energy envelope function, $\tilde{\gamma}_i[n]$, can then be expressed as $\tilde{\gamma}_i[n] = \sigma_\eta(i) \cdot \gamma_i[n]$, where $\sigma_\eta(i)$ denotes the time varying noise power.

4.2.4 Removing phase mismatches

Representing speech units with a set of pitch synchronous frames is particularly convenient when it comes to concatenation of sounds. As the sound units can be represented as a set of frames, the overlap-add procedure will ensure a smooth boundary, where the signal at the segment border will be a weighted average of the adjacent frames. To avoid phase distortion at the boundary, the two adjacent speech frames should be phase aligned (maximally in phase). The phase mismatch can be estimated from the cross-correlation function of the adjacent frames, or the frames can be aligned using a pitch-marking algorithm [46–49]. However, when a harmonic model is applied, the estimated phases of the harmonic model can be applied to solve this problem. In [50], a post-processing of all the voiced speech frames is proposed. The idea is to linearly propagate the phases of the harmonic components such that the phase of the first harmonic component in all speech frames is zero. As the distance to propagate is the same for all the sine wave components, the phase change of the $k'th$ harmonic with frequency $k\omega_0$ is k times the phase change of the first harmonic component:

$$\tilde{\phi}(k\omega_0) = \phi(k\omega_0) - k\phi(\omega_0), \tag{4.23}$$

where $\tilde{\phi}_k = \tilde{\phi}(k\omega_0)$ are the new propagated phases.

This method is reported to completely remove the phase mismatch problems in the AT&T's Text-to-Speech system [50], successfully applied to concatenation of diphones for various voices.

Chapter 5

Speech modification and smoothing

The notion of *speech modification* will in this thesis refer to the use of signal processing to change one or several acoustic parameters to some predecided desired target. Considering modification of one feature at a time, the task is to modify this feature while keeping other features of the speech signal as unmodified as possible. For example, in pitch modification the goal is to modify the pitch while keeping other characteristics of the speech like duration and spectral content unmodified. The notion of *smoothing* will refer to the use of modification to smooth speech model parameter trajectories across speech unit boundaries in order to remove audible discontinuities at unit boundaries.

Different types of modification can be performed on the speech signal. The main types are prosodic modification, voice source modification and vocal tract filter modification. Prosodic modification refers to all types of modification that changes the prosody of the speech signal, including pitch and duration modification. Voice source modification refers to modification of the speaker characteristics or the voice quality, while vocal tract filter modification refers to the modification of the vocal tract filter, for example by modifying formant positions, energies or bandwidths in order to obtain smoother parameter trajectories across speech unit boundaries.

In this chapter, the main focus will be on pitch and duration modification using a frame based approach. First, the classic TD-PSOLA algorithm is described. Then modification employing a frame based sine wave model is described, including spectral envelope estimation and phase modeling based on the source filter model of speech production. It should be noted that the harmonic model used in the experiments in Chapter 9 is a special case of this frame based sine wave model. Finally, modification using source filter deconvolution and smoothing of speech across unit boundaries are briefly discussed.

5.1 Pitch Synchronous Overlap and Add

A classic approach for pitch and/or duration modification is known as time domain pitch synchronous overlap and add, TD-PSOLA [51]. The TD-PSOLA algorithm is based upon representing the speech signal as a set of speech frames, where the speech signal can be reconstructed by the overlap and add method, as described in Section 2.3. In addition, the speech frames are required to be pitch synchronous, meaning that the frame shift is one pitch period in voiced speech regions. For a perfect reconstruction of the speech signal, the synthesis weighting windows have to sum to one for all samples as described for the OLA method (Eq. 2.14). This can approximately be achieved if the speech frames span two pitch periods and Hanning or triangular[1] synthesis windows are applied.

$$\hat{s}[n] = \sum_{i=-\infty}^{\infty} w_s^i[n - n_a(i)] \cdot s_i[n - n_a(i)], \tag{5.1}$$

where $\hat{s}[n]$ is the reconstructed signal, $w_s^i[n]$ is the symmetric synthesis weighting window of the $i'th$ speech frame, and $s_i[n - n_a(i)]$ is the speech frame centered at the analysis instants $n_a(i)$.

The TD-PSOLA algorithm relies on the knowledge of the speech signal's pitch, voicing, and also a set of pitch marks which defines the analysis instants, $n_a(i)$. These parameters can be estimated in an off-line speech processing step in order to minimize the processing at run time.

In case of pitch modification, a desired pitch contour $T_s(t)$ can be expressed relative to the original pitch contour

$$T_s(t) = \beta(t) \cdot T_a(t), \tag{5.2}$$

where $\beta(t)$ is a time varying pitch scaling factor, and $T_a(t)$ is the estimated original pitch period contour.

Duration modification can be viewed as a compression or an expansion of the time axis, which in general also leads to a new synthesis pitch

[1] An exact reconstruction would in principle require a constant pitch or more elaborate window functions.

contour that is a compressed or expanded version of the analysis pitch contour[2].

$$T_s(D(t)) = T_a(t), \tag{5.3}$$

where $D(t)$ represents a mapping of the time axis to a warped time axis corresponding to the desired time scale modification. For most practical cases a constant duration modification factor can be applied, $D(t) = \alpha t$. $\alpha = 2$ would for example correspond to a duration doubling.

In order to modify the pitch and/or duration, the analysis instants can be replaced with new synthesis instants $n_s(j)$, where the distances between the new synthesis instants correspond to the new synthesis pitch period contour T_s. The calculation of these new synthesis instants will be described further in the next section. If the duration is to be kept constant when the pitch is modified, the number of new synthesis instants will in general be different from the number of analysis instants. The synthesis speech frames corresponding to the new synthesis instants can be selected by using the original speech frames that are nearest to the new synthesis instants on the time scaled axis. The modified speech signal $\tilde{s}[n]$ is finally synthesized by OLA.

$$\tilde{s}[n] = \sum_{j=-\infty}^{\infty} \tilde{w}_s^j[n - n_s(j)] s_j[n - n_s(j)], \tag{5.4}$$

where $\tilde{s}[n]$ is the modified signal, $n_s(j)$ are the new synthesis instants, $s_j[n - n_s(j)]$ are the synthesis speech frames, and $\tilde{w}_s^j[n - n_s(j)]$ are the synthesis weighting windows.

5.1.1 Calculation of synthesis instants

The remaining task is to calculate the new synthesis instants, $n_s(j)$, that correspond to the desired pitch and duration contour. The distance between adjacent synthesis instants can then be expressed as

$$n_s[j+1] - n_s[j] = \frac{1}{n_s[j+1] - n_s[j]} \int\limits_{n_s[j]}^{n_s[j+1]} T_s(t)dt, \tag{5.5}$$

where the distance between new synthesis instants is calculated as the average pitch period between the new synthesis instants.

[2] Normally the warping of the time axis introduce only small changes in the pitch contour, and if the pitch is constant the pitch will remain the same

If the pitch contour $T_s(t)$ is linear in between two synthesis instants, Eq. 5.5 simplifies to:

$$n_s[j+1] - n_s[j] = \frac{T_s(n_s[j])}{1 - b/2}, \tag{5.6}$$

where b is the slope of the linear pitch contour. By applying Eq. 5.5 or Eq. 5.6, the synthesis time instants can be calculated recursively. For example, when $n_s[0]$ is defined, Eq. 5.6 can be used to calculate $n_s[j]$ for increasing j. Commonly, a piecewise linear pitch contour is assumed. Equation 5.6 will then be a good approximation if the slope b is slowly varying or small, which normally is a reasonable assumption in pitch period contours of speech. Alternatively, a second order equation can be solved to obtain the new synthesis time instants [5].

For example, if a constant pitch segment is assumed for simplicity, a halving of the duration would lead to new synthesis instants that correspond to every second analysis instant on the time scaled (compressed) time axis. The modified signal is then obtained by the OLA method using every second speech frame from the original signal with the same spacing as in the orignal sentence, leading to a signal of half the duration.

5.1.2 Properties of TD-PSOLA

TD-PSOLA is known to lead to some distortion of the modified speech [51]. In the frequency domain, this distortion corresponds to a multiplication of the spectrum by the Fourier transform of the analysis window. When using speech frames of two periods, corresponding to wide band TD-PSOLA, this leads to a broadening of formants [51].

It is also important that the analysis instants in the TD-PSOLA approach are close to the main excitation instants in the speech. If the position of the analysis instants deviate from the excitation instants with more than 30% (relative to the pitch period), the result is a very hoarse speech quality [51]. From a practical point of view, it is important to center the analysis window around the main excitations in the speech signal in order to avoid echoes, or reverberation, of the excitations in the modified speech.

Another weakness with TD-PSOLA, is that the approach can lead to buzzyness when segments are stretched [5] This is due to inducing a periodicity at high frequencies by repeating similar speech frames, and is in particular a problem for voiced fricatives.

5.2 Pitch and duration modification using a sinusoidal model

An approach that alleviates the effect of the analysis window is known as frequency domain PSOLA[51]. In this approach, a speech model is applied to synthesize modified speech frames based on the estimated spectral characteristics of each frame. Different speech models can be applied to model the speech frames. Speech models that can be applied are for example the source filter model, e.g. LPC-PSOLA [5; 52], or a sinusoidal model or a hybrid sinusoidal/source-filter model. In the experiments in this thesis, a sinusoidal or more specifically a harmonic model has been applied. Hence, the main focus in this chapter will be on modification by a harmonic model. A source filter model and a sinusoidal/harmonic model can however be combined [6], which will be described further in the next section.

A challenge is to preserve the spectral content of the original speech during pitch modification. For example, if a harmonic model is applied, a naive approach could be to synthesize speech frames simply by using a modified pitch $\tilde{f}_0$ in the harmonic model equation. This would however lead to a warped frequency spectrum resulting in a distorted speech quality. To keep the spectral content unmodified, the amplitude and phase spectrum have to be resampled at the new harmonic frequencies, $\tilde{f}_k$.

$$\tilde{f}_k = k \cdot \tilde{f}_0, \quad k = 1, \ldots, \tilde{K}, \tag{5.7}$$

Assuming a perfectly periodic impulse train source, the spectrum is in theory discrete, with non-zero values only for the harmonic frequencies. Hence, strictly we do not know the frequency response in between the harmonics. However, in Section 2.1 the model of speech production motivates the use of an AR-model for the vocal tract filter, which would yield a smooth underlying spectral envelope. The assumption of a smooth underlying spectral envelope implies that a spectral magnitude envelope and a spectral phase envelope can be obtained by interpolating the original amplitude spectrum and phase spectrum respectively [3]. The interpolation of the sine wave phases is however not as straightforward as the interpolation of the sine wave amplitudes. This is because the phases vary rapidly with time. For example, the phase of the k'th harmonic will vary within the range $[0, k \cdot (2\pi)]$ during a pitch period (assuming constant pitch). The estimated phases for the speech frame will however be obtained in the interval $[0, 2\pi]$. Hence, the estimated phases have to be properly unwrapped before the interpolation can be performed. In the next section, a phase model

[3] Alternatively, the complex spectrum can be interpolated.

based on the source filter model of speech production will be described, before amplitude and phase spectrum interpolation are discussed further.

5.3 A sinusoidal representation of the source filter model

The source filter model of speech production can be used in combination with the sine wave model [6; 33], in order to model the source and the vocal tract filter separately. For pitch modification, a separation of source and filter is desirable as only the source is to be modified while the vocal tract filter is to be kept constant.

The source filter model, described in Section 2.1.2, can in the case of a time varying system filter be expressed as [6]

$$s(t) = \int_{0}^{t} h_s(t, t-\tau) e(\tau) d\tau, \tag{5.8}$$

where $h_s(t, t-\tau)$ is a time varying system filter and $e(t)$ is the excitation. In general $e(t)$ can be chosen to represent an arbitrary source with time varying parameters as in the general sine wave model [33].

$$e(t) = \sum_{k=0}^{L} a_e(t, \omega_k(t)) \cdot cos(\theta_e(t, \omega_k(t))), \tag{5.9}$$

where $a_e(t, \omega_k)$ are the time varying excitation signal amplitudes and $\theta_e(t, \omega_k)$ are the time varying phases. However, for the harmonic sine wave model applied in this thesis, the quasiperiodic nature of voiced speech is modeled by an overlap and add of piecewise periodic segments. Hence, only a periodic excitation will be considered here. The periodic excitation can be expressed by the sine wave model with constant parameters

$$e(t) = \sum_{k=0}^{K} a_e(\omega_k) cos(\int_{t_0}^{t} \omega_k d\tau + \phi_e(\omega_k)), \tag{5.10}$$

where $e(t)$ is the periodic excitation, implying constant amplitudes $a_e(\omega_k)$ and constant frequencies ω_k. The phase of the $k'th$ sine wave can be expressed as a phase offset $\phi_e(\omega_k)$ and a time varying term depending on the sine wave frequency ω_k. When a periodic excitation is assumed, the frequencies ω_k are constant, and the integral in Eq. 5.10 yields a phase model varying linearly with time.

$$\theta_e(t, \omega_k) = (t - t_0)\omega_k + \phi_e(\omega_k), \tag{5.11}$$

where $\phi_e(\omega_k)$ are the fixed phase offsets. If the phase offsets $\phi_e(\omega_k)$ are zero for all k, the excitation will be a peaked pulse like waveform with a peak at t_0. The time instant where all the sine waves are in phase (or maximally in phase) is defined as the pitch pulse onset time [6]. For a periodic excitation with $\phi_e(k) = 0$ for all k, the sine wave components will be in phase for every time instant

$$t_0 + i \cdot T \tag{5.12}$$

where T is the excitation period, and i is an integer [4].

If the parameters of the excitation are constant over the duration of the impulse response, the sine wave model can be expressed as the convolution of the excitation and the system filter, corresponding to a multiplication of the complex sinusoidals in the frequency domain.

$$s(t) = \sum_{k=0}^{K} a_e(\omega_k) \cdot |H_s(t,\omega_k)| \cdot cos((t-t_0)\omega_k + \phi_e(\omega_k) + \phi_s(t,\omega_k)), \tag{5.13}$$

where $|H_s(t,\omega_k)|$ is the magnitude response of the system filter and $\phi_s(t,\omega_k)$ is the phase response. When assuming a time invariant system filter

$$H_s(t,\omega_k) = H_s(\omega_k) = a_s(\omega_k) \cdot e^{j\phi_s(\omega_k)}, \tag{5.14}$$

it is seen that the amplitudes and phases can be identified as

$$a_{\omega_k} = a_e(\omega_k) \cdot a_s(\omega_k) \tag{5.15}$$

$$\theta_{\omega_k}(t) = (t-t_0)\omega_k + \phi_e(\omega_k) + \phi_s(\omega_k), \tag{5.16}$$

where $\theta_{\omega_k}(t)$ are sine wave phases, $\phi_s(\omega_k)$ are the system filter phases, $\phi_e(\omega_k)$ are the excitation phases and $(t-t_0)\omega_k$ is a time varying phase term, also referred to as linear propagation. Similarly, a_{ω_k} are the estimated amplitudes, $a_s(\omega_k)$ are the system filter amplitudes and $a_e(\omega_k)$ are the excitation amplitudes.

If considering the amplitudes and phases of a harmonic model in discrete time, the amplitudes and phases can similarly be expressed as

$$a_k = a_e(k\omega_0) \cdot a_s(k\omega_0) \tag{5.17}$$

$$\phi_k = (n_a - n_0)k\omega_0 + \phi_e(k\omega_0) + \phi_s(k\omega_0), \tag{5.18}$$

where n_0 is defined to be the discrete time instant corresponding to the pitch pulse onset time t_0. ω_0 is the fundamental frequency, a_k and ϕ_k are

[4] If a pitch synchronous approach with a frame shift of one period is applied, there will be the same number of pitch pulse onset times as speech frames.

the harmonic amplitudes and phases estimated at the analysis instant n_a, $\phi_s(k\omega_0)$ are the system filter phases, $\phi_e(k\omega_0)$ are the excitation phases and $(n_a - n_0)\omega_k$ are the time varying phases resulting from the propagation of the pitch pulse onset time to the analysis instant.

From the model in Eq. 5.17 it can be seen that the sine wave amplitudes in this model can be interpolated and resampled without any separation of source and filter. For the interpolation of phases, the term $(n_a - n_0)\omega_k$ in Eq. 5.18 has to be considered, introducing the phase unwrapping problem or equivalently the problem of estimating the pitch pulse onset time.

5.4 Spectral Envelope Estimation

The interpolation of the sine wave amplitudes is performed by estimating a spectral envelope. The term *spectral envelope* refers to a smooth curve or envelope, approximating the spectral magnitude response. The spectral envelope can be estimated by the use of a speech model, like an all pole model or a more general pole zero model. Another approach is to fit a smooth function that goes through the peaks of the STFT. One example of the latter approach is the SEEVOC algorithm [53]. The SEEVOC algorithm uses a peak picking algorithm to identify the spectral peaks in the STFT, and interpolates between the peaks in the estimated STFT to obtain an initial estimate of the spectral envelope. Then an iteration scheme is applied to improve the estimate of the spectral envelope in the valleys of the STFT spectrum. Another promising approach, STRAIGHT [110], applies pitch adaptive spectral analysis to estimate a smooth surface in time and frequency, applying interpolation in both time and frequency. A third approach, related to a harmonic sine wave model, is based on fitting a smooth function to the estimated harmonic amplitudes with a set of cepstral coefficients. These cepstral coefficients are referred to as the discrete cepstrum coefficients [55], and will be described in more detail in the next section.

5.4.1 Discrete cepstrum coefficients

Given a set of K sine wave amplitudes a_k, corresponding to the harmonic frequencies f_k, the discrete cepstrum coefficients [54] are obtained by minimizing the squared error between an estimated spectral envelope $|S(f_k)|$ and the observed harmonic amplitudes a_k in the log domain. Omitting frame notation the error function is defined as

$$\varepsilon = \sum_{k=1}^{K} (ln(a_k) - ln|S(f_k)|)^2 = ||ln(\underline{\mathbf{a}}) - ln(\underline{\mathbf{S}})||^2, \tag{5.19}$$

where

$$ln(\underline{\mathbf{a}}) = [ln(a_1), ln(a_2), \ldots, ln(a_K)]^T$$
$$ln(\underline{\mathbf{S}}) = [ln|S(f_1)|, ln|S(f_2)|, \ldots, ln|S(f_K)|]^T$$

$\underline{\mathbf{a}}$ is the vector of observed harmonic amplitudes, and $\underline{\mathbf{S}}$ is a vector consisting of the estimated spectral envelope evaluated at the harmonic frequencies.

From the definition of the cepstrum, see Section 2.6, an estimated log spectral envelope can be represented as the magnitude of the Fourier transform of the complex cepstrum or equivalently as the cosine transform of the (real) cepstral coefficients.

$$ln|S(f_k)| = c_0 + 2\sum_{m=1}^{p} c_m \cdot cos(2\pi f_k m), \tag{5.20}$$

where c_m are the cepstral coefficients, and $p+1$ is the number of cepstral coefficients in the cepstrum vector $\underline{\mathbf{c}} = [c_0, c_1, \ldots, c_p]$.

Using matrix notation the log spectrum can be expressed as

$$ln(\underline{\mathbf{S}}) = \mathbf{M}\underline{\mathbf{c}}, \tag{5.21}$$

where the matrix $\mathbf{M}$ is defined as:

$$\mathbf{M} = \begin{bmatrix} 1 & 2cos(2\pi f_1) & 2cos(2\pi 2 f_1) & \ldots & 2cos(2\pi p f_1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2cos(2\pi f_K) & 2cos(2\pi 2 f_K) & \ldots & 2cos(2\pi p f_K) \end{bmatrix} \tag{5.22}$$

To obtain well behaved solutions, a smoothness constraint $\lambda \cdot \mathbf{R}(S(f))$ is used to penalize rapid variations in the spectral envelope [55].

$$\varepsilon_r = ||ln(\underline{\mathbf{a}}) - ln(\underline{\mathbf{S}})||^2 + \lambda\mathbf{R}(S(f)), \tag{5.23}$$

λ denotes the degree of regularization, and should be increased as the cepstral order p approaches K. The regularization term, $\lambda \cdot \mathbf{R}$, is optimally chosen to be a matrix with diagonal elements $8\pi^2[0, 1^2, 2^2, \ldots, p^2]$ [55].

The cepstral vector $\underline{\mathbf{c}}$ that minimizes the error in Eq. 5.23 is obtained by solving the linear system of equations. The solution, also referred to as the pseudoinverse, is

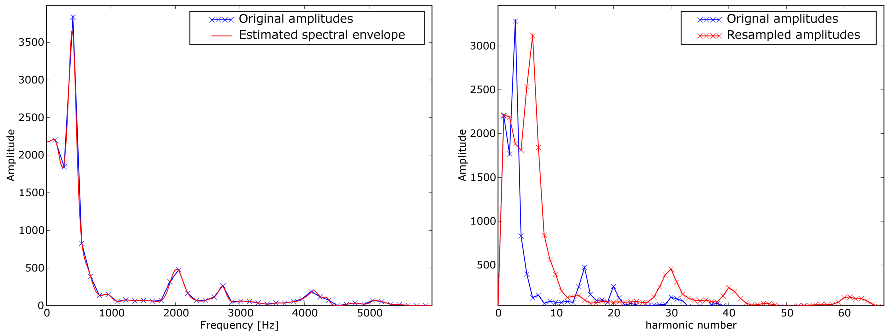
$$\underline{\mathbf{c}} = [\mathbf{M}^T\mathbf{M} + \lambda\mathbf{R}]^{-1}\mathbf{M}^T ln(\underline{\mathbf{a}}), \tag{5.24}$$

5.4.2 Interpolation of the amplitude spectrum

If the frequencies f_k in the M-matrix in Eq. 5.22 are the harmonic frequencies, and $\underline{\mathbf{a}}$ is the set of estimated amplitudes corresponding to the harmonic frequencies, a mapping $\underline{\mathbf{c}} = \Psi_1(\underline{\mathbf{a}}, f_0)$, can be defined by Eq. 5.24. Similarly a mapping $\underline{\tilde{\mathbf{a}}} = \Psi_2(\underline{\mathbf{c}}, \tilde{f}_0)$, can be defined by Eq. 5.21 using the harmonic frequencies of $\tilde{f}_0$ in the $\mathbf{M}$ matrix. These two mappings define an interpolation and a resampling of the harmonic amplitude spectrum to new harmonic amplitudes $\underline{\tilde{\mathbf{a}}}$ at the fundamental frequency $\tilde{f}_0$, preserving the estimated spectral envelope. The number of cepstral coefficients defines the amount of smoothing of the original estimated amplitude spectrum. A number of 32 cepstral coefficients was found in [54] to be a good heuristic threshold for avoiding distortion due to too much smoothing of the spectral envelope. However, in general the number of cepstral coefficients required for distortionless reconstruction of the speech will depend on the complexity or smoothness of the original harmonic amplitude spectrum for each frame and also the bandwidth. As a simplified rule, the number of cepstral coefficients required for distortionless reconstruction will increase with the number of harmonic amplitudes to be fitted, which depend on pitch and bandwidth. An example of a spectral envelope estimated by the discrete cepstrum approach is shown in Figure 5.1(a). In Figure 5.1(b), the original amplitude spectrum is compared to the amplitude spectrum obtained when resampling the spectral envelope from the spectral envelope for a halving of the pitch. Note that the harmonic amplitudes are plotted as a function of the harmonic number, k, in order to illustrate both the preservation of the spectrum shape and the change in amplitude for each harmonic sine wave track.

5.5 Phase modeling

Interpolation of the phase spectrum is a more difficult problem due to the need for phase unwrapping, as discussed in Section 5.2. The importance of the interpolation (resampling) of the phase spectrum is however not so clear as for the magnitude spectrum. In some approaches, designed to obtain a low bit rate, the phase information has been omitted [56; 57]. This has been motivated by the fact that intelligible speech can be obtained even without the phase information, although with a slightly reduced speech quality. Omitting phase information in this sense does not mean that a random phase is used, as it is important that the phases of the sine waves of adjacent speech frames are continuous in time, in order to avoid abrupt phase changes that can cause audible phase discontinuities, at least in the low



(a) Spectral envelope

(b) Amplitude spectrum

FIGURE 5.1: Estimated spectral envelope by the discrete cepstrum approach (to the left), and resampled amplitudes of the harmonic frequencies for a halving of the pitch to the right.

frequency voiced bands. Thus, omitting phase information in this sense means to use a model of the phase. Examples of such phase models are the zero phase model, the minimum phase model and the maximum phase model [6]. All models have been reported to yield reconstructed speech of high intelligibility, but with some loss of naturalness [6].

The zero phase model simply defines all phases to be zero at the pitch pulse onset times, or alternatively at the analysis instants when a pitch synchronous approach is assumed. A high peak is hence generated in the waveform at the pitch pulse onset times. An example of a speech segment synthesized by the zero phase model is shown in Figure 5.2, where the speech is synthesized by a pitch synchronous harmonic model and the OLA method. For comparison, the same speech segment as shown in Figure 4.2 at page 39 was used for this example.

The minimum phase model employs a model of the system filter as a stable causal system. The minimum phase model and the maximum phase model are also attractive for coding purposes, since both the phase and magnitude spectrum then can be represented by the cepstral coefficients, defined in Section 2.6. The phases can then be calculated as

$$\phi_s(\omega) = -2 \sum_{m=1}^{\infty} c_m sin(m\omega), \tag{5.25}$$

where $\phi_s(\omega)$ are the system filter phase and c_m are the cepstral coefficients describing the system filter spectral envelope. The sum in Eq. 5.25 must in a practical case be approximated with a finite number of cepstral coefficients.

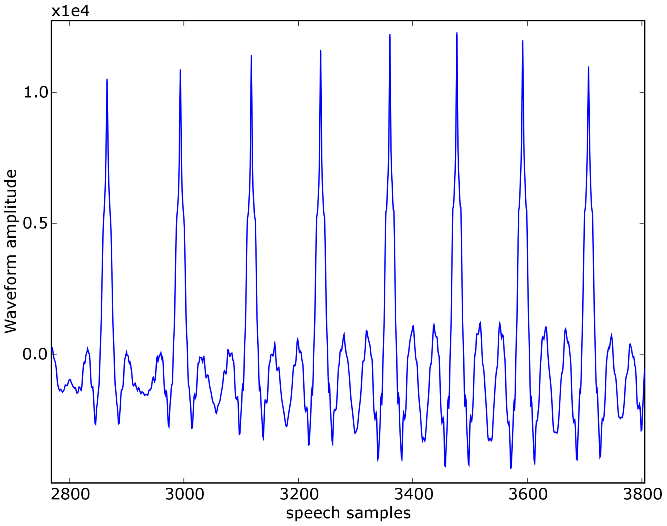


FIGURE 5.2: A reconstructed speech frame from an /e:/ sound of a male speaker using a pitch synchronous harmonic model and OLA. The harmonic amplitudes where obtained by WLS-estimation, while the phases for each frame were set to zero.

The appropriateness of using a minimum phase assumption depends on the system filter properties. A problem is that the system filter also includes the glottal source. The glottal source is more appropriately modeled as two poles outside the unit circle or as a set of zeros both inside and outside the unit circle [6]. Thus, the system filter is more appropriately modeled with a mixed phase filter. In addition, the nasal cavity can introduce zeros both inside and outside the unit circle, where the zeros outside the unit circle also contribute to a mixed phase representation. It can be shown that the minimum phase model corresponds to flipping the poles and zeros that are outside the unit circle to its reciprocal location inside the unit circle. For the glottal source this relates to modifying the source to have a sharp attack and a slow decay instead of a slow opening and an abrupt decay, and hence the energy is compressed towards the origin [6]. An example showing a speech segment synthesized by the harmonic model using a minimum phase model (Eq. 5.25) and the OLA method is shown in Figure 5.3. For comparison, the same example segment as shown for the zero phase model is used.

The loss of naturalness when applying a zero phase, minimum phase

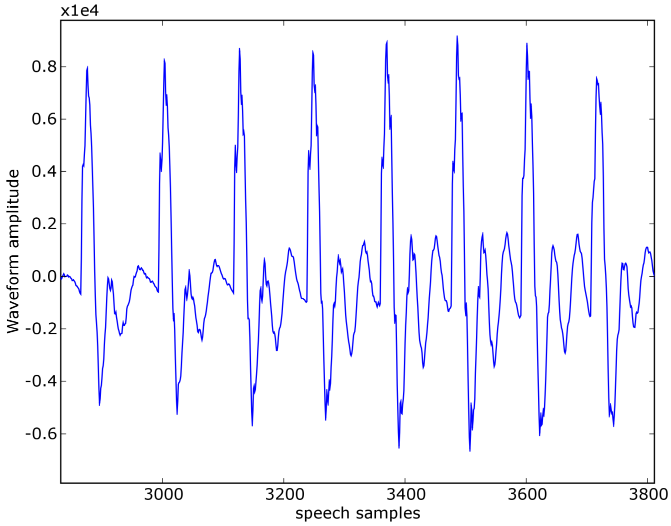


FIGURE 5.3: A synthesized speech frame from an /e:/ sound of a male speaker using a pitch synchronous harmonic model and OLA. The harmonic amplitudes were obtained by WLS-estimation, while the harmonic phases were obtained from the discrete cepstral coefficients using the minimum phase assumption.

or maximum phase model, indicates that the phases are important to obtain high quality modification of speech. At least, it seems important to maintain the relations between the sine wave phases, in order to preserve the natural time domain shape of the speech. In auditory modeling theory, this can be explained by a phasic/tonic view of auditory neural processing described in [6]. This implies that the auditory system is sensitive to both the frequency domain envelope and the time domain envelope of the waveform. It is thus important to maintain the phase relation between the sine waves, so that the time domain envelope of the waveform also is preserved [58]. However, when performing time and pitch scale modification it is impossible to preserve both the time domain envelope and the frequency domain envelope exactly, motivating the use of error minimization approaches [59].

5.5.1 Phase interpolation

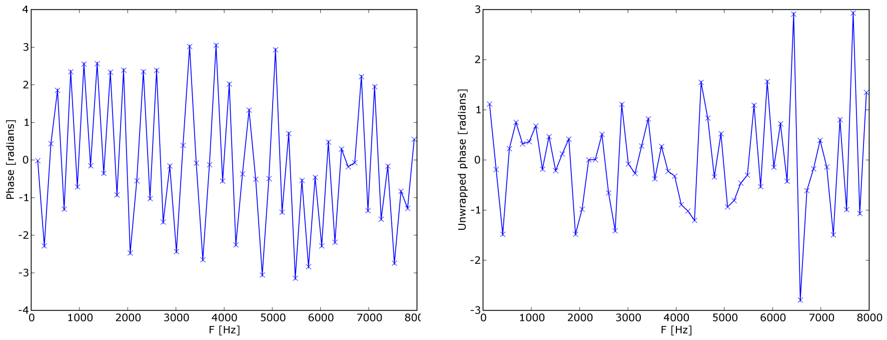
In order to better preserve the time domain shape of the speech signal when performing pitch modification, the phase spectrum of the original signal can also be interpolated and resampled in the frequency domain. The sine wave phase model for a speech frame based on constant sine wave frequencies, derived in Section 5.3, is

$$\phi_k = (n_a - n_0)\omega_k + \phi_e(\omega_k) + \phi_s(\omega_k), \tag{5.26}$$

where ϕ_k are the sine wave phases, $\phi_e(\omega_k)$ are the excitation phases, and $\phi_s(\omega_k)$ are the system filter phases. n_a is the analysis instant, and n_0 is the pitch pulse onset time. The excitation phases $\phi_e(\omega_k)$ will be defined to be zero in the following discussion, which corresponds to an excitation signal where all sine wave phases are exactly in phase at the pitch pulse onset times. Alternatively, the excitation phases can be interpreted as integrated into the system filter. Further, the analysis instant, n_a, is defined to be the origin of the speech frame ($n = 0$) as in the definition of the speech frame in Eq. 2.12.

An approach to the phase interpolation problem is to unwrap the phases by removing the dominating linear term in Eq. 5.26. This corresponds to estimating the pitch pulse onset time n_0 [60], and then propagating the phases back to the pitch pulse onset time by linear wave propagation to obtain an estimate of $\phi_s(\omega_k)$. The assumption is that the system filter response $\phi_s(\omega_k)$ is a more smooth function, better suited for frequency domain interpolation. An example of unwrapped phases compared to the originally estimated phases is shown in Figure 5.4.

The pitch pulse onset time [60] is also in literature referred to as a significant excitation instant, where the main excitation instant in voiced speech is known to be at or just preceding the glottal closure instant due to an abrupt closure of the vocal folds [61]. The estimation of the onset time can be performed by an MSE approach [60], or by a fixed point analysis [62], or by estimating the slope of the phase spectrum as in the approach of [61], where the excitation instants are estimated using the negative derivative of the phase spectrum, known as the group delay function. The estimation of the excitation instants is reported to be difficult, due to that there may be several excitation instants within a frame. In addition, if the voicing is weak it may even be difficult to define the instant of excitation at all. In addition to the major excitation, other excitations could be estimated at the start of the open phase, or anytime due to noise bursts in the speech [61]. For example, the release burst of a plosive will also give rise to an excitation instant. In synthesized speech, the estimation errors in the onset time



(a) Original phases

(b) Unwrapped phases

FIGURE 5.4: Estimated phases for a speech frame from an /e:/ sound of a male speaker, estimated phases to the left, and unwrapped phases to the right.

can result in adjacent frames having irregular excitation instants, leading to discontinuities in the phase of adjacent frames and a variable pitch (jitter), leading to the perceptual effect similar to a relaxed or more hoarse voicing. A method to avoid this effect is to constrain the excitation instants to be regularly spaced related to the estimated pitch period [33].

5.6 Modification of speech based on source filter separation

In the separation of the source and the vocal tract filter for the sinusoidal model described in this chapter, the glottal filter was included in the system filter. However, a similar separation could be performed regarding the glottal flow derivative as the excitation signal.

The source filter model based on source filter deconvolution, described in Section 2.1.3, gives more freedom in how to modify the glottal source. For example, the shape parameters of the glottal source model, e.g. a LF-model, can be kept constant during modification, leading to a possibly modified spectral envelope of the glottal filter, while the vocal tract filter is preserved. A motivation for more freedom in how to modify the glottal source is to get more control of the voice source [63], which might also be needed for pitch modification to be consistent with speech production. For example, the amplitude of the first harmonic component is known to be correlated to the open quotient O_q, defined in Section 2.1.3, of the voice source [16]. More control of the voice source could make it possible to do more complex voice source modifications. The parameterization of the voice source could also be applied in spectral distance measures, in order to detect audible discontinuities due to voice source characteristics in unit selection synthesis. A challenge in the source filter deconvolution approach is however to do a complete separation of the source and the filter.

5.7 Smoothing techniques in concatenative synthesis

Many approaches have been proposed for minimizing or smoothing spectral mismatch at concatenation borders. One approach for minimizing the spectral mismatch is to search for the optimal point of concatenation between two speech units [64]. Another approach is to add additional context sensitive units (normally diphones) to the database inventory, applying spectral clustering to avoid a too large number of units [65]. Another strategy is to reduce spectral mismatches by smoothing the spectral content across speech unit boundaries [39; 66–68]. The methods for spectral smoothing can be used in combination with the methods for minimizing spectral mismatch. In the next section, the focus will be on a smoothing approach based on a harmonic model.

5.7.1 Pitch and amplitude smoothing for a harmonic model

Assuming the possibility for phase mismatches are removed in the speech analysis process, e.g. by using a pitch synchronous approach, only the amplitudes and the pitch have to be considered for smoothing at unit boundaries. In [39] a linear interpolation around the concatenation point t_i is proposed. The difference in pitch and amplitude at each side of the concatenation border is measured, and this difference is then propagated to the left and to the right of the concatenation point.

$$
\begin{aligned}
\Delta\omega_0 &= (\omega_0^R - \omega_0^L)/2 \\
\tilde{\omega}_0^l &= \omega_0^l + \Delta\omega_0 \frac{l}{L}, \quad l = L, L-1, \ldots, 1 \\
\tilde{\omega}_0^r &= \omega_0^r - \Delta\omega_0 \frac{r}{R}, \quad r = R, R-1, \ldots, 1
\end{aligned}
$$

ω_0^L is the pitch of the speech frame immediately to the left of the concatenation border. The pitch of the next speech frame to the left is denoted

ω_0^{L-1} and so forth. Similarly, ω_0^R is the pitch of the speech frame immediately to the right of the concatenation border, ω_0^{R-1} is the pitch of the next speech frame to the right, and so forth. L and R are the number of frames available for smoothing to the left and right respectively. Using similar notation for the smoothing of amplitudes, the smoothed amplitudes of the k'th harmonic can be expressed as:

$$\begin{aligned}
\Delta a_k &= (a_k^R - a_k^L)/2 \\
\tilde{a}_k^l &= a_k^l + \Delta a_k \frac{l}{L}, \quad l = L, L-1, \ldots, 1 \\
\tilde{a}_k^r &= a_k^r - \Delta a_k \frac{r}{R}, \quad r = R, R-1, \ldots, 1
\end{aligned}$$

It is reported in [42] that the amplitude smoothing makes formant discontinuities less perceptible, but that if formant frequencies are very different at each side of a boundary the problem is not completely solved. Drawbacks with this modification scheme is that it is dependent on the number of speech frames available for smoothing at each side of the boundary, and that it does not account for the derivative of the parameter trajectories at each side of the concatenation boundary.

Chapter 6

Pitch and voicing estimation

This theory chapter will concentrate on pitch estimators based on sinusoidal speech models, which will be further discussed in Chapter 9 in relation to a pitch synchronous speech-processing algorithm. The requirements for the pitch estimation in this pitch synchronous processing algorithm are mainly to provide robust and unbiased estimates when using relatively short duration analysis windows of the speech. For this task, pitch estimators based on sinusoidal models are well suited, which will be motivated in this chapter. A broader view of pitch estimation methods can be found in [69–72]. The theory of this chapter is mostly derived from Hess [69], Quatieri [6] and Therrien [8].

6.1 Definition of pitch

Originally, the term pitch was used to describe the perceived tone height of a periodic input signal, while the term fundamental frequency was intended to be used in the signal processing or speech production view of periodicity. However, the term pitch has been widely used in the literature for different definitions of periodicity, and will also in this thesis be used in a wide sense, synonymous to the term fundamental frequency. In a speech production view, the pitch period (T_0) can be defined as (Hess [69], 1983):

Definition 1
T_0 is defined as the elapsed time between two successive laryngeal pulses. Measurement start at a well specified point within the glottal cycle, preferably at the point of glottal closure, or if the glottis does not close completely, at the point where the glottal area reaches its minimum.

This definition motivates the notion of a pitch period: starting with a glottal closure instant and lasting until the glottis closes again. In a signal processing view, the pitch period can be defined as ([69]):

Definition 2
T_0 is defined as the average length of several periods, i.e, as the average elapsed time between a small number of successive laryngeal cycles. How the averaging is performed and how many parameters are involved are matters of the individual method

The second definition is the standard definition for any method that applies short term stationary analysis, as for example the cepstrum method [73] (Noll,1967), and the autocorrelation method [74] (Rabiner, 1977).

The term voicing is more difficult to define, as the excitation of speech can be mixed, with both voiced and unvoiced components. Different thresholds can however be defined to provide a binary voicing decision, where the thresholds can be set to optimize performance in the specific application.

6.2 Pitch halving and doubling

One fundamental problem in pitch estimation is to avoid pitch halving and doubling errors. This problem is due to that several possible fundamental frequencies can be equally valid from a signal processing point of view. For example, assume a perfectly periodic signal with fundamental period T_0. Then the signal is also perfectly periodic with period $n \cdot T_0$, where n is a positive integer, and hence there is an ambiguity in the solution. In Figure 6.1 this ambiguity is illustrated.

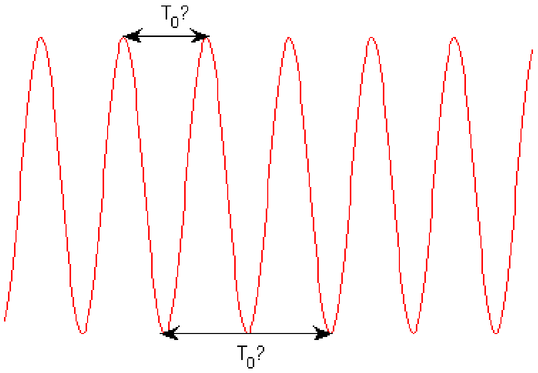


FIGURE 6.1: The pitch halving/doubling ambiguity.

6.3 Model based pitch estimation

In the classic autocorrelation based pitch estimator [74], the pitch estimate is obtained from the short time autocorrelation function [6] by using an estimate of the pitch period as the time lag to the maximum peak of the autocorrelation function. The height of the maximum peak can be used as a measure of voicing. However, without modifications, this approach is known to have some weaknesses due to effects from windowing the speech signal. A rectangular window of the speech signal will correspond to assuming that the speech signal is zero outside the window. The short time autocorrelation of the windowed sequence will then have a triangular envelope that leads to a biased pitch estimate [71]. Using a longer time window could improve the bias problem, but at the cost of poorer time resolution, due to an increased averaging of the pitch when the signal is not perfectly periodic. In addition, a longer analysis window would in general lead to loss of stationarity, making the pitch estimation task more difficult. An approach to avoid the bias problem without using a longer analysis window is to apply a speech model. With the use of a speech model, the short time characteristics of the speech signal can be extrapolated to infinite time by assuming a periodic signal. Then a long duration analysis window can be applied to the extrapolated signal, avoiding the bias problem without loss of stationarity.

6.4 Pitch estimation based on a sine wave model

Different criteria have been used to derive pitch estimators based on sinusoidal models. One approach is to use the criterion of self-similarity, which also can be used to derive the autocorrelation pitch estimator [6]. The error function used in this approach is

$$E(T) = \sum_{n=-\infty}^{\infty} (s_i[n] - s_i[n+T])^2, \tag{6.1}$$

where $s_i[n]$ is a speech frame and T denotes a candidate pitch period estimate. Minimizing $E(T)$ with respect to T (for T larger than a lower bound), results in the autocorrelation pitch period estimator

$$\hat{T}_0 = \arg\max_T \left(\sum_{-\infty}^{\infty} s_i[n] \cdot s_i[n+T] \right) \tag{6.2}$$

Then consider a general sinusoidal model for $s_i[n]$

$$\hat{s}_i[n] = \sum_{l=1}^{L} a_l e^{j\omega_l n + \phi_l}, \tag{6.3}$$

where the amplitudes a_l, the frequencies ω_l, and the phases ϕ_l are estimated from the short time segment $s_i[n]$. If inserting this model in Eq. 6.1 and using the extrapolation principle, it can be shown that the resulting pitch estimator results in a comb filter approach [6]. This approach corresponds to running the waveform through a filter with peaks at multiples of a hypothesized fundamental frequency, and then selecting the estimated pitch as the candidate pitch that gives the largest energy of the filter output.

The comb filter approach is however ambiguous to integer multiples of the pitch period. Hence, the approach suffers from possible pitch halving errors [6]. Note that it does not suffer from pitch doubling errors. This can be seen by assuming a perfectly periodic signal, which results in a perfectly discrete harmonic spectrum, and observing that the power of the filter output is unchanged when applying a comb filter with the double resolution corresponding to $f_0/2$. If applying a comb filter with half the number of peaks, corresponding to $2 \cdot f_0$, the filter output power will be halved.

Pitch estimators based on the self similarity criterion hence may suffer from pitch halving errors. This class of pitch estimators includes the autocorrelation method, standard frequency domain comb filter methods and homomorphic methods (like the cepstrum method [73]). Modifications to these pitch estimators can however be made to alleviate the problem of pitch halving errors. For example, in a modified autocorrelation method [106], a weight is tuned to penalize the selection of low pitch periods.

6.4.1 Pitch estimation based on a harmonic model

Another criterion that can be used to derive a pitch estimator based on a sinusoidal model is to select the pitch that minimizes the mean squared error between the harmonic model and the speech waveform [6]. The harmonic model for a speech frame yields a perfectly periodic signal, hence the approach can be interpreted as a search for the fundamental frequency of a periodic signal that is most similar to the speech waveform in the analysis frame. The approach can be expressed as minimizing the error function

$$E(f_0, \underline{\boldsymbol{a}}, \underline{\boldsymbol{\phi}}) = \frac{1}{N_w} |s_i[n] - \hat{s}_i[n; f_0, \underline{\boldsymbol{a}}, \underline{\boldsymbol{\phi}}]|^2, \tag{6.4}$$

where $s_i[n]$ is the original speech frame of size N_w, $\hat{s}_i[n]$ is the fitted harmonic model, depending on the candidate pitch f_0, the estimated harmonic amplitudes, $\underline{\mathbf{a}}$, and the estimated harmonic phases $\underline{\boldsymbol{\phi}}$. Assuming the harmonic amplitude and phases are estimated from the STFT as described in Section 4.2.2.2, it can be shown that the resulting pitch estimator can be expressed as

$$\widehat{\omega}_0 = \arg\max_{\omega_0} \sum_{k=1}^{K(\omega_0)} |S(k\omega_0)|^2, \tag{6.5}$$

where $\widehat{\omega}_0 = 2\pi \hat{f}_0$ is the estimated pitch, $S(k\omega_0)$ is the STFT evaluated at the candidate harmonics, and $K(\omega_0)$ is the number of harmonic components. However, this estimator also acts as comb filter. Hence, without modifications this approach also suffers from possible pitch halving errors.

In the Multiband excitation (MBE) vocoder [37] a pitch estimator based on a frequency domain criterion similar to the error criterion in Eq. 6.4 is applied.

$$\varepsilon(\omega) = \frac{1}{2\pi} \int\limits_{-\pi}^{\pi} \left(|S_i(\omega)| - |\hat{S}_i(\omega)|\right)^2 d\omega, \tag{6.6}$$

where $S_i(\omega)$ is the spectrum of the speech frame, and $\hat{S}_i(\omega)$ is the spectrum of the estimated speech model. In contrast to the error function in Eq. 6.4, this error function applies only the magnitude spectrum. Hence, phase information is ignored. The error function in Eq. 6.6 is minimized by a search algorithm over the parameter space of the speech model, where the spectral parameters are estimated for each candidate pitch. To improve the computational efficiency of the search, the search is first performed by evaluating Eq. 6.5 on a coarse[1] grid using a FFT. Then $\varepsilon(\omega)$ is evaluated using a finer grid around the initial minimum. Once a pitch period that minimizes ε is found, the errors at submultiples of the pitch are also evaluated, and the smallest pitch period leading to a comparable error is chosen. Hence, the problem of pitch halving errors is alleviated.

6.4.2 Pitch estimation using a general sine wave model

The pitch halving ambiguity can also be avoided when some a priori knowledge of the vocal tract spectral envelope is known [6; 75]. The approach is based on using a general sine wave model for the speech waveform $s_i[n]$ in Eq. 6.4 and an extrapolation of the sine wave representation to a larger time interval. Quatieri suggests to estimate the a priori spectral envelope by the SEEVOC [53] method [6], see Section 5.4. A problem is however that the

[1] In practice the coarse grid consisted of integer pitch periods

SEEVOC algorithm also to some extent depends on a coarse pitch estimate [76].

Without going into the details of the derivation of this estimator, which is described in detail in [6; 75], the result can be expressed as minimizing the mean squared error $E(\omega_0)$, or equivalently as maximizing a function $\zeta(\omega_0)$.

$$E(\omega_0) = P_s - 2\zeta(\omega_0), \tag{6.7}$$

where $E(\omega_0)$ is the mean square error, P_s is the power of the observed signal, and $\zeta(\omega_0)$ can be expressed as [6].

$$\zeta(\omega_0) = \sum_{k=1}^{K(\omega_0)} \bar{a}(k\omega_0) \left[\sum_{l=1}^{K} a_l W(\omega_l - k\omega_0) - \frac{1}{2}\bar{a}(k\omega_0) \right], \tag{6.8}$$

where ω_l and a_l respectively are the frequencies and amplitudes of the sine wave representation, $W(x) = |\frac{sin(N_w \cdot x/2)}{N_w sin(x/2)}|$, N_w is the framesize, and $\bar{a}(\omega)$ is the a priori spectral envelope.

Assuming the input speech is periodic, $W(\omega_l - k\omega_0)$ is zero at submultiples of the true pitch ω^*, and due to that the spectral envelope is always non-zero, it can be seen that the function ζ fulfils

$$\zeta\left(\frac{\omega^*}{m}\right) < \zeta(\omega^*), m = 2, 3, \ldots \tag{6.9}$$

which shows that this estimator avoids the pitch halving ambiguity [6].

This is easier to see when setting the prior spectral envelope $\bar{a}(k\omega_0)$ to be constant,

$$\zeta(\omega_0) = \sum_{l=1}^{K} a_l \left[\sum_{k=1}^{K(\omega_0)} \bar{a} W(\omega_l - k\omega_0) \right] - \frac{1}{2} \sum_{k=1}^{K(\omega_0)} \bar{a}^2 \tag{6.10}$$

The first term is a correlation like term, similar to a comb filter in the frequency domain, and the second term is a negative compensation or penalizing factor for low frequency candidates.

Using the optimal ω_0 and Eq. 6.7, the SNR can be expressed as:

$$SNR = \frac{P_s}{E} = \frac{P_s}{P_s - 2\zeta(\omega_0)} \tag{6.11}$$

Hence also maximization of the SNR of a harmonic model leads to an unambiguous pitch estimate under the assumption of an a priori spectral envelope.

The SNR can also be used for voicing decisions: If the SNR is large the harmonic fit is good, which indicates that the input speech is probably voiced. In [75] the SNR is related to the probability of voicing using a heuristic mapping from the SNR to an approximate probability of voicing.

6.5 Pitch and voicing estimation based on HNR

Another pitch estimation approach related to a harmonic model is to select the candidate pitch that maximizes the harmonic to noise power ratio, HNR. The HNR approach is based on dividing the speech into a deterministic component (harmonic component) and a stochastic component (noise component). Several approaches can be applied to estimate the stochastic component. One approach is to set the stochastic component to equal the residual of a harmonic model or a harmonic+noise model [36]. This would however lead to a stochastic component dominated by high frequencies. Considering the speech production model in Section 2.1, the stochastic component can be modeled as filtered white noise, and hence the stochastic component should be wide band. This has motivated a separation of the speech into a deterministic component and a wide band noise component [77; 78]. Considering the i'th speech frame, the decomposition can be expressed as

$$s_i[n] = (\tilde{h}_i[n] + \tilde{r}_i[n]) \circledast v_i[n], \tag{6.12}$$

where $\tilde{h}_i[n]$ and $\tilde{r}_i[n]$ are the harmonic and noise component of the excitation signal respectively, $v_i[n]$ is the impulse response of the vocal tract filter, and $\circledast$ denote circular convolution. The approach for separating the harmonic and the stochastic component is based on a spectrum peak picking approach, using a frequency analysis of the inverse filtered speech signal and an iterative algorithm, described in detail in [77; 78]. Once the estimation of the harmonic and noise component is performed, the harmonic to noise ratio is calculated as

$$HNR_e = \sum_{\omega} \frac{|\tilde{H}(\omega)|^2}{|\tilde{R}(\omega)|^2}, \tag{6.13}$$

where $\tilde{H}(\omega)$ is the spectrum of the harmonic part and $\tilde{R}(\omega)$ is the noise spectrum. The summation over ω refers to the sum of the discrete number of frequency bins used in the DFT. The e subscript notation in HNR_e is used to indicate that the measure is an estimate of the HNR of the excitation signal. In the experiments in Chapter 8, the HNR will be defined as the power ratio of the harmonic and the aharmonic part of the original speech signal.

Pitch estimation based on the HNR can be performed by searching for the fundamental frequency candidate that maximizes the HNR, similar to the MBE-approach described in Section 6.4.1. In [77], it is proposed to first use a simple pitch estimator to narrow down the search area to 5 pitch candidates and their neighborhoods. Then the iterative decomposition algorithm is applied to calculate the HNR_e for all candidate frequencies. This approach was found to be better at discriminating against both pitch doubling and halving than the sinusoidal based pitch estimator described in Section 6.4.2.

The HNR can also be used for voicing estimation, as it is an estimate of the power ratio of the voiced deterministic component to the unvoiced noise component. In [79] this voicing estimator outperformed other voicing measures such as zero crossing rate and a voicing measure based on the gain and the first coefficient from LP analysis. A computational disadvantage with the HNR voicing measure is however that the HNR is dependent on a fundamental frequency estimate, and hence a search for an optimal fundamental frequency has to be performed for each speech frame, even for unvoiced speech frames.

6.6 Pitch and voicing estimation using ESPRIT

In spectral estimation theory, the subspace methods, including the ESPRIT algorithm, constitute a well known class of spectral estimators, designed to estimate complex sinusoids in noise [8]. The subspace methods assume the observed sequence is of the form:

$$s[n] = \sum_{l=1}^{M} A_l e^{j2\pi f_l n} + \eta[n], \tag{6.14}$$

where A_l is a complex amplitude, $|A_l|e^{j\phi_l}$, and $\eta[n]$ is the noise. The subspace methods are based on eigenvector analysis of the correlation matrix, which can be interpreted as dividing the signal into a signal space and a noise space. If the noise is white, the correlation matrix R_s can be expressed as

$$R_s = \sum_{l=1}^{M} P_l \underline{\boldsymbol{x}}_l \underline{\boldsymbol{x}}_l^{*T} + \sigma_0^2 \boldsymbol{I}, \tag{6.15}$$

where $P_l = E\{|A_l|^2\}$ is the power of the l'th sinusoid, $\underline{\boldsymbol{x}}_l = e^{j\omega_l n}$, $n = 0, \ldots, N-1$, where N is the frame length, and σ_0^2 is the variance of the noise. Solving the eigenvalue problem:

$$R_s \underline{\boldsymbol{e}}_k = \lambda_k \underline{\boldsymbol{e}}_k, \tag{6.16}$$

will give M signal eigenvectors, $\underline{\boldsymbol{e}}_1 \ldots \underline{\boldsymbol{e}}_M$, and $Ns - M$ noise eigenvectors, $\underline{\boldsymbol{e}}_{M+1} ... \underline{\boldsymbol{e}}_{Ns}$, where Ns is the size of the correlation matrix. The signal eigenvectors will correspond to large eigenvalues, $\lambda_1 \ldots \lambda_M$, depending on the power of the sinusoids, while the noise eigenvectors will ideally be orthogonal to all $\underline{\mathbf{x}}_l$, and hence correspond to small eigenvalues, $\lambda_{M+1}...\lambda_{Ns}$, with magnitude σ_0^2. Similar expressions can be obtained for sinusoids in colored noise if applying a whitening transform [8].

6.6.1 Estimation of the sine wave frequencies using ESPRIT

ESPRIT, Estimation of signal parameters via rotational techniques, exploits an invariance principle that naturally exists for time series [80]. The first step in the ESPRIT algorithm is to estimate the correlation matrix, R_s. The correlation matrix should be estimated with a method that corresponds to the covariance method (or the modified covariance method [8]), defined in Section 2.1.2. The next step is to estimate the number of complex sinusoids. Looking at the eigenvalues of the correlation matrix as in the previous section, it is seen that the eigenvectors corresponding to the noise ideally should have small eigenvalues σ_0^2, while the sinusoidal components have larger eigenvalues. However, in practice there is not necessarily a clear threshold that divides the weakest sinusoidal components and the noise components. Solutions to this problem could be to use some form of the Akaike information criterion or the minimum description length, developed for the case of sinusoids in noise [81], or to use some simple thresholding.

When the number of complex sinusoids, M, is estimated, the ESPRIT algorithm can be applied to estimate the frequencies $f_1, f_2, \cdots, f_M$. Note that this vector of estimated frequencies will consist of positive and negative frequencies occurring in pairs. Hence there will only be $M/2$ positive frequencies, $f_1, f_2, \cdots, f_{M/2}$, which will be referred to as the ESPRIT frequency vector. The total least squares version (TLS) of the ESPRIT algorithm can be summarized as [8]:

1. Define the $N+1$-dimensional random vector $\underline{\mathbf{s}}$ pertaining to $N+1$ consecutive data samples $s[0], s[1], \ldots, s[N]$ and estimate the correlation matrix $\hat{\mathbf{R}}_\mathbf{s}$ from the data samples

2. Compute the generalized eigenvectors and eigenvalues of $\hat{\mathbf{R}}_s$.
   $\hat{\mathbf{R}}_s \underline{\mathbf{e}}_k = \lambda_k \underline{\mathbf{e}}_k, \qquad k = 1, \ldots, N+1$

3. Estimate the number of signals M.

4. Generate a basis spanning the signal subspace and partition it as

$$\bar{\mathbf{B}} = \begin{bmatrix} \vdots & & \vdots \\ \underline{\mathbf{e}}_1 & \cdots & \underline{\mathbf{e}}_M \\ \vdots & & \vdots \end{bmatrix} = \begin{bmatrix} & \mathbf{B} & \\ & & \\ x & \cdots & x \end{bmatrix} = \begin{bmatrix} x & \cdots & x \\ & & \\ & \mathbf{B}' & \end{bmatrix}$$

where the $x \cdots x$ denote a row that is not of direct concern.

5. Compute the matrix $\mathbf{V}$ of right singular vectors using singular value decomposition of $\begin{bmatrix} \mathbf{B} & \mathbf{B}' \end{bmatrix}$ and partition $\mathbf{V}$ into four $(M \times M)$ submatrices $\begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}$

6. Compute the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_M$ of the matrix $\mathbf{\Psi}_{TLS} = -\mathbf{V}_{12}\mathbf{V}_{22}^{-1}$.

7. Find the desired frequencies from $\omega_k = \angle\lambda_k, k = 1, 2, \ldots, M$

6.6.2 Estimation of the ESPRIT amplitudes and phases

When the sine wave frequencies are estimated, the complex amplitudes can be estimated in the same way as described for the harmonic model using least squares estimation or weighted least squares estimation, see Section 4.2.2.1. The ESPRIT frequencies will for a voiced frame be close to the most prominent harmonic frequencies of f_0, with almost the same amplitudes as in the harmonic model, see Figure 6.2. However, the ESPRIT algorithm can sometimes provide frequencies in between harmonics, or two frequencies that are very closely spaced. In the case of two sine waves with almost equal frequency, the estimation of the amplitudes will normally produce one high energy sine wave and one low energy sine wave. Hence, the low energy sine wave could be considered more or less as a "spurious" component in this case.

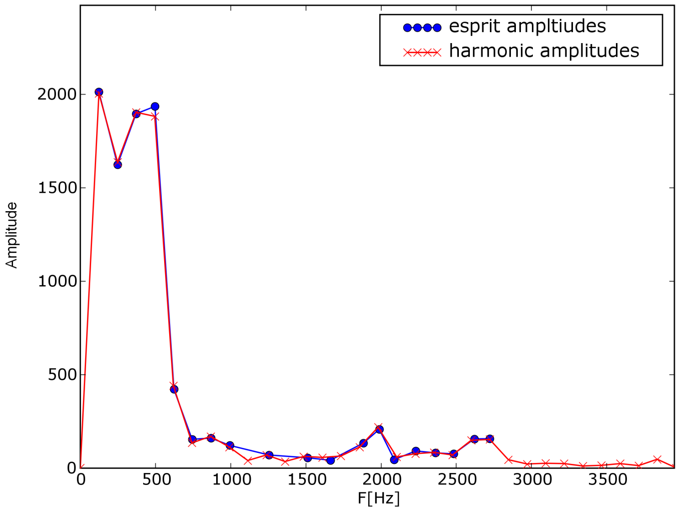


FIGURE 6.2: Example of estimated ESPRIT amplitudes compared to the estimated harmonic amplitudes (WLS-estimation). The number of complex exponentials M in the estimation was chosen to be 38 corresponding to 19 real sine waves. The example speech frame was from an /e:/-sound from a male speaker.

6.6.3 Pitch and voicing estimation

Two reasonable options for pitch estimation by the ESPRIT algorithm are to choose the lowest frequency from the ESPRIT frequency vector, or to use a weighting of all the frequencies in the frequency vector. The latter approach can be motivated by Figure 6.2, where the frequencies obtained by the ESPRIT algorithm are close to the harmonic frequencies. In [82] an iterative least squares approach is proposed for pitch estimation, where the error function is the difference between the magnitude weighted ESPRIT frequency vector and the harmonic frequencies corresponding to the candidate pitch. A similar approach is proposed in [83], where the L first frequencies in the ESPRIT frequency vector is assumed to be harmonically related, where L is a chosen threshold for including only relatively low frequency components. However, no magnitude weighting was applied in this approach.

In principle, the ESPRIT pitch estimation does not suffer from pitch halving errors. However, the estimation would depend on a robust estimation of the correlation matrix, which would require some a priori knowledge of the pitch. The appropriateness of the ESPRIT pitch estimation would also rely on the validity of the assumption that the ESPRIT frequencies are harmonically related.

Voicing estimation can be performed by measuring to what extent the ESPRIT frequencies are harmonically related. More precisely, the variance of the interpeak distance in the frequency vector is proposed as a measure of voicing [82]. A theoretical advantage with the ESPRIT approach is that the subspace methods also model additive noise. Hence, it should theoretically yield a robust approach for voicing estimation with respect to additive noise.

Chapter 7

Building join cost functions

The topic in this chapter is the design of join cost functions based on perceptual experiments. How well a join cost function works will depend on how well the distance measures in the join cost function discriminate good and bad joins, and on how many relevant features that are included in the cost function and how these features are combined and weighted.

In the first section of this chapter, a listening test on the detection of discontinuities in two Norwegian vowels is described. The listening test was conducted to obtain a reference for comparing different objective spectral distance measures, which will be presented in the second section. In the third section, a probabilistic join cost model is proposed, and two different strategies for building a join cost function based on perceptual experiments are presented.

7.1 Listening test design

The perceptual experiment [84] was conducted on listeners' detection of audible discontinuities in vowel joins generated by concatenative synthesis, using a Norwegian female speaker. Joins in two Norwegian long vowels A and e, SAMPA */A:/* and */e:/*, were tested. The listening test was conducted as a binary forced choice experiment, where listeners had to make a forced binary decision whether a join was discontinuous or not. 20 adult volunteer listeners, most of them employees at the Signal Processing Group at NTNU, participated in the test.

7.1.1 Test stimuli generation

The stimuli in the experiment was generated from the speech database Prosdata [85], which is a collection of 502 Norwegian sentences read by a female speaker. This speech database consists of manually segmented phonemes, syllables and words, in addition to estimates of the pitch and root mean square energy values. A simple automatic demiphone (half phone) segmentation was performed by dividing each phoneme into two equal length demiphones. The speech synthesizer used to generate the stimuli was implemented in Matlab [86], using a Mysql database [87] for storing speech unit data.

The stimuli in this experiment consisted of one vowel join inside a test word. The test word was in the middle of a relatively short test sentence. Test sentences were selected among the original sentences in the speech database. That is, a search among the original sentences in the database was applied to find suitable test sentences containing suitable test words. If a test sentence was very long, only a part of the sentence, containing the desired vowel, was extracted. Then, in order to make a speech unit join at the middle of the vowel, the demisyllable starting at the middle of the vowel was replaced by another instance of this demisyllable from the database, as illustrated in Figure 7.1.

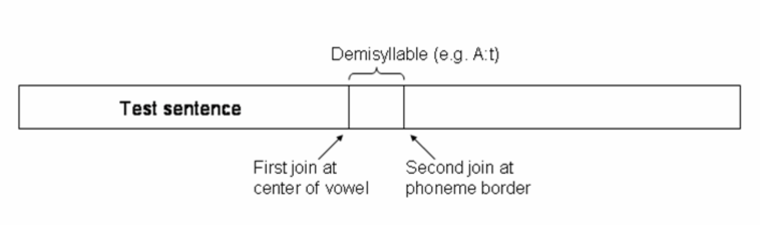


FIGURE 7.1: Illustration of a test stimulus, where a target demisyllable in the sentence has been replaced with another instance of the demisyllable from the speech database.

This gave two joins: The first join at the middle of the vowel, and a second (unwanted) join at the phoneme boundary between the syllable containing the vowel and the next syllable. A bad second join could possibly influence the listeners' choice, and was hence a noise source in this experiment. This noise source was attempted minimized by restricting the (unwanted) second join to have an unvoiced consonant on at least one side of the join. The unwanted second join was also the reason for using demisyllables as speech units, as the second unwanted join then will be further away from the vowel join. Then, it could possibly be easier for the listeners to separate a bad vowel join from a possible bad second join. However, a disadvantage with the choice of using demisyllables was that fewer speech units were available for possible replacement.

Speech unit candidates for replacing the demisyllable were selected from the database by using the synthesizer's target cost function. In this speech unit search the prosodic parameters of the original sentence were used as target values, and the speech units with a target score higher than a chosen threshold were chosen as test stimuli candidates. The features included in this target cost function were the pitch difference at the beginning and at the end of the unit, duration, root mean square energy, left and right phonetic context, syllable context, and word context. The weights of this target cost function had been manually tuned by evaluating the quality of copy synthesis[1].

7.1.1.1 Waveform generation

The test stimuli were finally generated by concatenating the speech waveforms of the speech units in the sentence. In order to avoid discontinuities at the join due to phase mismatches, the synthesizer used cross-correlation of two single period speech frames at each side of the join to estimate the phase difference. This phase mismatch estimate was used to adjust the boundaries of the speech units. For voiced sounds, the synthesizer also used an overlap-add of two periods across the join to avoid clicks due to waveform discontinuity. In order to avoid possible discontinuities due to root mean square energy (rms) difference at the join, the waveform of the inserted speech unit candidate was scaled so that the rms energy in the vowel parts at each side of the join were equal.

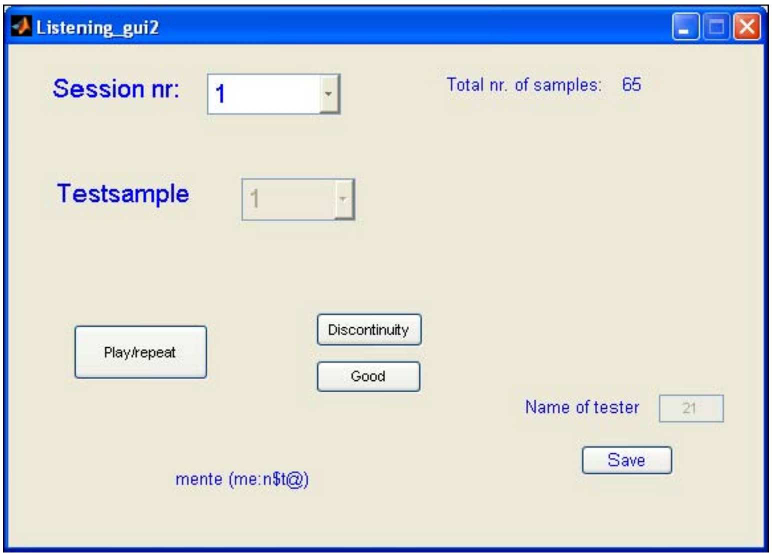
7.1.2 Test Procedure

The test consisted of 210 stimuli with a join in the vowel /*e:*/ and 248 stimuli with a join in the vowel /*A:*/. The test was designed for 20 listeners. The listeners were split in 4 groups with 5 listeners in each group. The test was split into two sessions for all listeners, one session with /*e:*/ stimuli and one session with /*A:*/ stimuli. All listeners in the same group listened

[1] The term copy synthesis refers to the synthesis of a sentence that exists in the speech database from which the synthesizer is built. The prosodic parameters of the speech units in the original sentence are used as target values. The units from the original sentence are excluded from the speech unit search. The quality of the synthesized sentence can then be compared to the quality of the original sentence.

to the same block of stimuli, but with a randomization of the order of the stimuli within each session. The ten first stimuli of the /e:/ sessions constituted a familiarization phase, where examples of two good joins, three bad joins, and five practice stimuli were presented to the listener. Two or three original sentences were also added to each session, making a total of 65 stimuli in each session.

The listening test was presented using a graphical interface where the listeners could press a button to play or repeat a stimulus. The listeners could repeat a stimulus as many times as they wanted, before they had to make a forced binary decision whether the join contained an audible discontinuity or not. All participants in the test were given an instruction before the test started. Listeners were instructed to focus their listening on the word containing the join, and ignore possible audible discontinuities elsewhere in the sentence. For each stimulus the test-word containing the join was shown in a graphical interface with its SAMPA transcription in parenthesis, so that the listeners could know where in the sentence to concentrate their listening. The graphical user interface for the test is shown in Figure 7.2. The test was performed on the same computer with the same headphones by all listeners. The test took normally about 20 minutes for completion.



Session nr: 1

Total nr. of samples: 65

Testsample 1

Name of tester 21

mente (me:n$t@)

FIGURE 7.2: The graphical user interface for the listening test.

7.2 Comparing spectral distance measures

In this experiment, a set of spectral distance measures are compared for the detection of audible discontinuities in vowel joins [84](2005). The main purpose of the experiment was to contribute to the problem of choosing spectral distance measure(s) for the join cost function, and to test the performance of a correlation based distance measure. As in [88] and [91] the problem was approached as a binary detection problem using ROC curves [96] to compare the distance measures.

The comparison is based on using the ratings from the listening test described in the previous section as a reference. The study follows earlier studies of comparing different spectral distance measures [88–93]. Macon and Wouters found that Euclidian distance on LPC-based cepstral coefficients was best [92]. Klabbers and Veldhuis found that the Kullback Leibler distance on LPC power spectra was the best predictor [88]. Stylianou and Syrdal found that the Kullback Leibler distance on DFT based power spectra and Euclidian distance on Mel frequency cepstral coefficients were promising detectors [91], while Donovan suggested to use Mahalanobis distance between cepstral parameters employing decision trees [93]. After this experiment was conducted, Pantazis and Stylianou have proposed to use an AM-FM parameterization of the speech [94].

Due to the large variability of the results in previously reported experiments, some of the most promising distance measures from the previous tests were compared in this test. In addition, a distortion measure based on cross-correlation, reported to improve the join cost function in [29], was included. The pitch difference at the concatenation point was also analyzed due to that some stimuli in the listening test had relatively high difference in the fundamental frequency (f_0) at the concatenation point. This was due to that there sometimes were few good candidates in the speech database with respect to this feature.

7.2.1 Distance measures

In order to calculate spectral distances, a spectral parameterization has to be obtained from each of the speech units adjacent to the join. In this experiment, a rectangular analysis window with its center at the concatenation point and duration of two periods was applied for the estimation of the spectrum at each side of the join. Hence, for the spectral analysis, the speech units were "expanded" with one period of speech data obtained from the speech units' original context.

The distance measures tested in this experiment were:

1. D_{skl}: Symmetrical Kullback Leibler distance evaluated on AR power spectra using 16 AR coefficients. The distance measure was calculated using Eq. 3.10 in Section 3.4.2. A more detailed description of the calculation of this distance is described in [28].

2. D_{cep}: Euclidian distance of a series of cepstral coefficients evaluated from AR power spectra using 16 AR coefficients. The distance measure was calculated as in Eq.3.6 in Section 3.4.1, except that the zero'th cepstral coefficient related to energy was omitted.

3. D_{LR}: Likelihood ratio. This measure was calculated as the mean of the non-symmetrical likelihood ratios calculated from AR spectra, as in Eq. 3.9 in Section 3.4.1.

4. D_{mfcc}: Euclidian distance between Mel frequency cepstral coefficients, calculated by Eq. 3.13 in Section 3.4.3 using 13 Mfcc coefficients. Note that the zero'th Mfcc coefficient representing energy was not used.

5. D_{mpsc}: Modified pitch synchronous cross-correlation. A modified version of the cross-correlation measure described in Section 3.4.4. The modification of the measure is described in more detail in the next section.

6. D_{F0}: The logarithm of the absolute value of the F0 difference between the estimated pitch at each side of the join.

7.2.2 Modified pitch synchronous cross-correlation

The purpose of the modification of the cross-correlation measure, described in Section 3.4.4, was to make the distance measure more uncorrelated to the f_0 feature. It was also a method to avoid the problem of calculating the cross-correlation measure, defined in Section 3.4.4, for speech frames of different length.

The D_{mpsc} distance measure was calculated using exactly one power normalized period from each side of the concatenation point: the last period of the unit to left of the concatenation point, $x_L[n]$, and the first period of the unit to the right of the concatenation point, $x_R[n]$. If the length of $x_L[n]$ and $x_R[n]$ are equal, the distance measure is identical to the measure proposed in [29] defined as the Euclidian distance between two pitch synchronous power normalized signals $x_L[n]$ and $x_R[n]$. If $x_L[n]$ and $x_R[n]$ were of different length, a modification was performed by either stretching or compressing $x_R[n]$ to the length of $x_L[n]$, followed by an interpolation to maintain the same sample instants. A piecewise cubic spline interpolation was performed to calculate the new samples. Denoting the interpolated version of $x_R[n]$ as $x'_R[n]$, the D_{mpsc} distance measure was calculated as

$$D_{mpsc} = \frac{1}{N}\sum_{n=1}^{N}(x_L[n] - x'_R[n])^2, \tag{7.1}$$

An example of a stretched and resampled waveform $x'_R[n]$ is shown in Figure 7.3

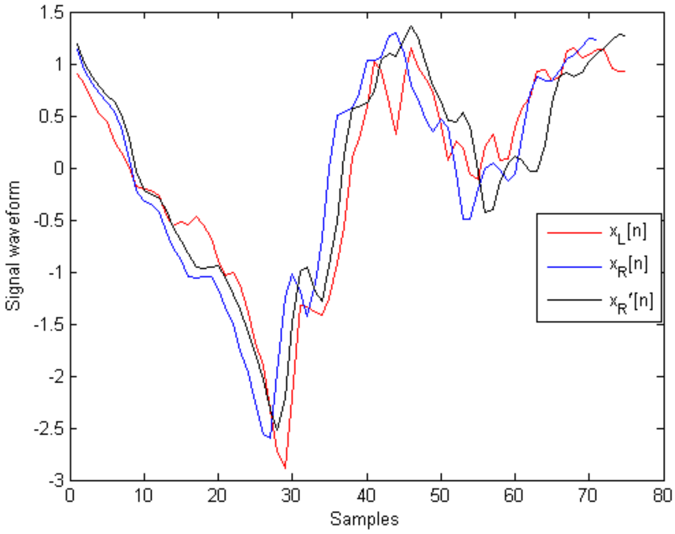


FIGURE 7.3: An example of a stretched and resampled waveform, $x'_R[n]$, obtained by stretching the waveform $x_R[n]$ to the length of $x_L[n]$.

Looking at vowel joins in natural sentences, this modification of the distance measure reduced the mean of this distance measure at natural joins by 68%. It should be noted that time domain stretching or compression leads to a warping of the frequency spectrum.

7.2.3 Results

The ten stimuli of the familiarization phase and the original sentences were all excluded when analyzing the results. All the original sentences were correctly labeled by all listeners as being without a discontinuity. For the other stimuli, the consistency between listeners was on average 72.7% for /e:/ and 68.6% for /A:/, where the consistency was defined as the percentage of listeners having chosen the same answer for a given stimuli. This indicates that the test was a difficult task, although many of the inconsistencies could be explained by different critical levels among listeners. For example, one listener could detect 20 discontinuities, while another listener could detect 40 on the same test stimuli, leading to a lower correlation between the listeners without that there necessarily is an inconsistency in the results.

Letting class Ω_0 represent the stimuli rated as including a perceived discontinuity, and letting class Ω_1 represent the stimuli rated as good, we have a two-class classification problem with several features, represented by the distance measures. Due to that five different listeners had evaluated each stimulus, two options were possible for the classification: Either to use an average of the listeners ratings for each stimulus, leading to a fuzzy classification [96], or to use a majority vote. The two approaches showed to yield similar results, hence only the results using average listener ratings will be presented here. The average rating of a stimulus can be interpreted as the estimated probability for a listener detecting an audible discontinuity for the given stimuli.

Although distance measures generally do not provide Gaussian distributed measurements, ROC curves [96] are informative describing the distance measures' ability to separate the two classes. Letting x denote an observed distance, and letting x^* be a specified distance threshold, the ROC curves plot the hit rate on the y-axis and the false alarm rate on the x-axis for different thresholds x^*. The hit rate, $P(hit)$, is defined as the probability of successfully detecting a discontinuity

$$P(hit) = P(x > x^* | \Omega_0), \tag{7.2}$$

where x is the observed distance, x^* is a chosen threshold, and Ω_0 is the class of perceived discontinuities. The false alarm rate, $P(false\,alarm)$, is defined as the probability of rejecting a good candidate,

$$P(false\,alarm) = P(x > x^* | \Omega_1), \tag{7.3}$$

where the class Ω_1 represents the good joins.

The calculation of $P(hit)$ and $P(false\,alarm)$ for a given distance measure was performed by choosing 100 thresholds x^* evenly spaced between the smallest and the largest observed value for the given distance measure. For each threshold the number of correctly classified stimuli (hits) and the number of misses (false alarms) were counted.

The ROC-curve for the D_{LR} distance measure will not be shown below, as the ROC curve for D_{LR} approximately followed the same ROC-curve as the ROC curve for D_{cep}. The D_{LR} distance measure was found to have a measured correlation with D_{cep} of 0.97. The ROC curves are shown in Figure 7.4 for /e:/, and in Figure 7.5 for /A:/.

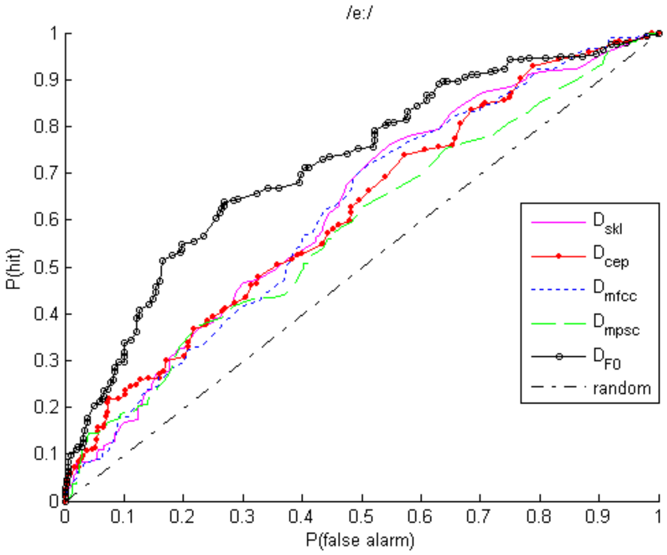


FIGURE 7.4: ROC curves for the vowel /e:/ . $P(hit)$, defined in Eq. 7.2, is plotted on the y-axis, and the false alarm rate $P(false\ alarm)$, defined in Eq. 7.3, is plotted on the x-axis.

Although the whole ROC curve is of interest, the hit rates at high false-alarm rates are especially interesting as they correspond to small distances x^* in the definition of $P(hit)$. A high false alarm-rate corresponds to the case of choosing the assumed best unit out of many good candidates, which is the desired situation in a unit selection point of view. From the ROC curves we see that D_{F0} was the best predictor of discontinuities in this specific test. We also see that the detection by D_{F0} was lower for /A:/ than for /e:/. However, we can not necessarily state that pitch differences are less important for the vowel /*A:*/. The detection rate will obviously also depend on the number of stimuli that actually cause an audible discontinuity due to pitch difference in the specific set of stimuli. For the vowel /A:/ there were more speech unit candidates to choose from in the database when generating the stimuli, leading to fewer stimuli with a high pitch difference at the join. This reflects that a feature's discriminability of the two classes (good or bad joins) will be correlated with the number of audible discontinuities that actually are due to this feature. Hence, the results of such tests would be highly dependent of the synthesis system and the

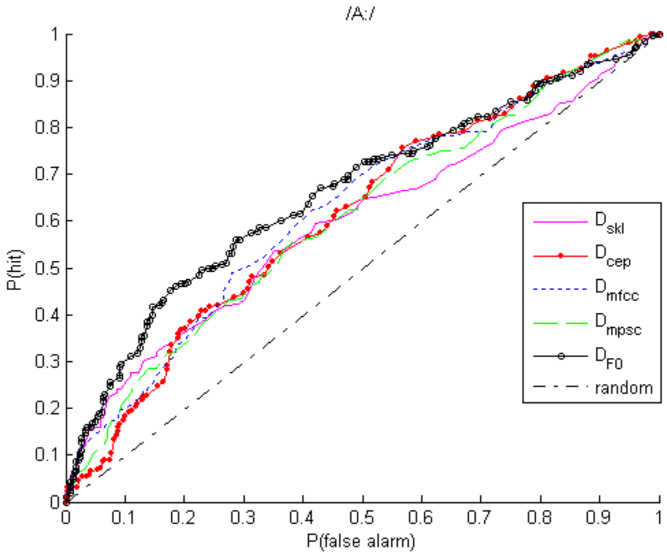


FIGURE 7.5: ROC curves for the vowel /A:/ .

test design.

7.2.4 Removing stimuli with high pitch differences

In order to reduce the influence of the stimuli with high pitch difference at the join, the stimuli with the highest log f_0 differences were removed from the test data. In practice, this was performed by removing the stimuli with a pitch difference higher than a given threshold defined by the criterion

$$P(\Omega_0 | D_{F0}) > 0.5 \tag{7.4}$$

That is, all stimuli with a probability of an audible discontinuity due to the D_{F0} distance measure higher than 0.5 were removed from the data set. The spectral distance measures were then analyzed on the remaining data. The removal criterion excluded data where the log f_0 difference was greater than 0.067 for */e:/* and less than 0.062 for */A:/*. This implied that 36.7% of the */e:/* stimuli and 30.2% of the */A:/* stimuli were removed from the original data. The resulting ROC curves are shown in Figure 7.6 for /e:/, and in Figure 7.7 for /A:/.

An interesting observation is that when high pitch differences were removed, see Figure 7.6 and Figure 7.7, the detection rate at high false-alarm

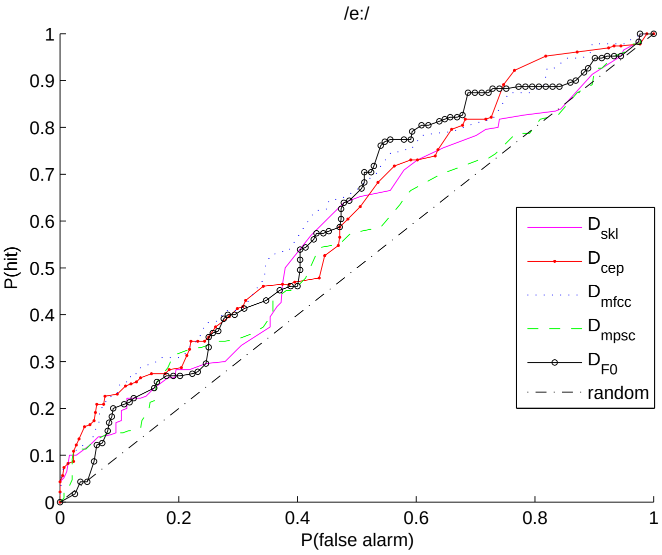


FIGURE 7.6: ROC curves after data with high F0 differences were removed for the vowel /e:/ .

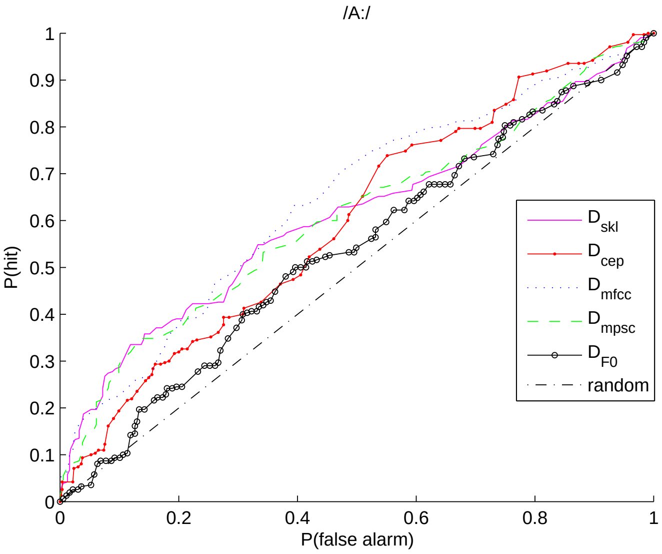


FIGURE 7.7: ROC curves after data with high F0 differences were removed for the vowel /A:/ .

rates was constant or increased for D_{cep} and D_{mfcc} for both */e:/* and */A:/*, while the hit rates of D_{skl} and D_{mpsc} were reduced. This indicates that D_{cep} and D_{mfcc} may be more orthogonal to D_{F0} than the other distance measures. However, more data would be needed to confirm such a hypothesis. When high pitch differences were removed, D_{mfcc} was the best performing distance measure, while D_{cep} was equally good or better for high false alarm rates.

7.2.5 Summary

In this experiment, five different spectral distance measures and f_0 difference was tested as detectors of human perceived discontinuities in two Norwegian vowels. The results were analyzed in two steps, where test stimuli with high pitch differences at the join were removed before analyzing the spectral distance measures.

Overall D_{F0} was the best detector in this specific test, showing that these types of experiments also reflect the frequencies of the different types of discontinuities occurring in the test stimuli, which could be one explanation for the large variability of results in such tests between different test systems and test stimuli. This suggests that it would be a good idea to establish carefully designed and shared test databases in this research field. Still, it would be important to be aware that the results still would reflect the specific test design and synthesis system in use.

Of the spectral distance measures, D_{mfcc} and D_{cep} (or D_{LR}) were the most promising distance measures, somewhat better than D_{skl}. The modified cross-correlation measure (D_{mpsc}) was not very promising in this test, and without the modification it showed to be highly correlated with pitch mismatches.

The detection of spectral discontinuities was quite low in this test. Hence, more experiments should be considered to draw conclusions on the performance of the spectral distance measures in this test.

7.3 Join cost function design

The topic in this section is the design of a join cost function for unit selection synthesis based on perceptual experiments. When using binary perceptual experiments to train the join cost function, the join cost function can be identified as the discriminant function [96] of a two-class pattern recognition problem. Similarly, in [91] a linear regression approach was applied, and in [94] a Fischer linear discriminant function was applied.

The approach to join cost function design proposed in this section is based on defining the join cost function as the probability of a listener perceiving a bad join, which leads to a nonlinear join cost function. The design of the join cost function can then be related to classical pattern recognition techniques as for example logistic regression [105] or neural networks [96].

Two different strategies for join cost function design are proposed in this section.

1. The first strategy is based on actively using the specific synthesizer for which the join cost function is to be built[2]. The idea is to use the target cost function of the specific speech synthesizer to generate stimuli for the perceptual experiment that typically would occur in the speech synthesis system. Then a join cost function could be fitted to the specific synthesizer in use. The problem of selecting features for the join cost function can then be solved by using stepwise linear regression [98] or stepwise logistic regression [99] on a set of candidate features.

2. The second strategy is to build a cost function from a set of chosen uncorrelated features. Perceptual experiments should then ideally be conducted on each feature in isolation. Then the probability for an audible discontinuity given each specific feature can be estimated. This could lead to a join cost function that is more general in the sense that it could be applied for different synthesizers and voices.

The outline of this section will be to first describe the proposed probabilistic join cost function, related to the second strategy. Then the probabilistic join cost function approach is compared to a classic linear join cost function, using the data from the listening test to estimate the join cost functions.

It should be noted that the detection of discontinuities with the spectral distance measures in the listening test was quite low. Hence, the test data are not the best for performing a comparison of these two approaches. The

[2] As in the listening test described in this chapter.

test data should also be split into a training set and an evaluation set for such a comparison. The purpose of the comparison presented in this section is hence only to present how the probabilistic approach can be applied for join cost function design.

7.3.1 A probabilistic model for the join cost function

The probabilistic model is based on defining the join cost as the probability of a listener perceiving an audible discontinuity for a given join. The model is hence based on that a join is only good if it is free of any audible discontinuity, which means that all types of audible discontinuities are considered to give the same amount of distortion. If this property is considered unfair, an additional weighting could be applied. However, the goal in speech synthesis should be to produce speech without any audible discontinuities.

In general, the probability of a perceived discontinuity at a concatenation point is a function of the windowed speech segments at each side of the concatenation point, referred to as s_L and s_R, assuming the windows have sufficiently long duration. The probability of an audible discontinuity for a specific join can then be expressed as the conditional probability $P(D|s_L, s_R)$, where D denotes an audible discontinuity.

However, in order to obtain an estimate of this probability, a set of distance measures, $d_1(s_L, s_R), \ldots, d_n(s_L, s_R)$, has to be applied:

$$P(D|s_L, s_R) \approx P(D|d_1, d_2, \ldots, d_n). \tag{7.5}$$

It is hence important that the distance measures have high correlation with human perception to make this approximation as correct as possible, as discussed in the previous section.

If it is assumed that statistically independent distance measures are used in the join cost function, Eq. 7.5 can be simplified to:

$$P(D|d_1, \ldots, d_n) = P(D|d_1) \cdot P(D|d_2) \cdots P(D|d_n) = \prod_{i=1}^{n} P(D|d_i) \tag{7.6}$$

This means in practice that the distance measures in the probabilistic join cost function should be as uncorrelated as possible in this approach. If the assumption of statistical independence is not valid, a join cost function with interaction terms would in general be needed.

Defining the probability of a good join $P(G) = 1 - P(D)$, gives the expression $P(G|d_1, d_2, \ldots, d_n) = 1 - P(D|d_1, d_2, \ldots, d_n)$, which by the assumption of statistical independence can be expressed as:

$$P(G|d_1, d_2, \ldots, d_n) = \prod_{i=1}^{n} P(G|d_i), \tag{7.7}$$

which leads to the join cost function:

$$C(\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n) = 1 - P(G|d_1, d_2, \ldots, d_n) = 1 - \prod_{i=1}^{n} \tilde{d}_i, \tag{7.8}$$

where $\tilde{d}_i = P(G|d_i)$. It is seen that in this model there are no explicit weights, but instead the difficult problem of estimating the conditional probabilities $P(G|d_i)$ is introduced.

7.3.2 Methods

Two methods for join cost function design are compared in this section.

1. Stepwise linear regression
2. The probabilistic approach, using logistic regression to estimate the posteriori probabilities

The first method is based on using a classic linear join cost function using stepwise linear regression to estimate the weights of the join cost functions, while the second method is based on using the proposed probabilistic join cost model.

As the second approach is based on using uncorrelated features in the join cost function, the join cost function in this comparison were defined to consist of only two features, one spectral distance measure, chosen to be the D_{mfcc} distance measure, and the difference in fundamental frequency, D_{F0}. In order to describe the two methods, some theory on stepwise linear regression and logistic regression will be presented in this subsection.

7.3.2.1 Method 1: Stepwise linear regression

The weights and which distance measures to include in the join cost functions can be determined by stepwise linear regression. This method assumes a linear model of the input parameters (linear cost function) and enters features one by one as long as new features significantly reduce the error variance [98]. A hypothesis test [98] is used to decide if a new feature should be entered into the model. The insertion of features into the model is continued until a feature fails to induce a significant increase in the explained regression.

The criterion used in stepwise linear regression is to minimize the sum of squared error as in standard linear regression [98],

$$\epsilon = \sum_{j=1}^{N} e_j^2 = \sum_{j=1}^{N} (y_j - \widehat{y}_j)^2, \tag{7.9}$$

where ϵ is the sum of squared error, e_j is the j'th residual, y_j is the average rating of the j'th stimuli in the listening test, and $\widehat{y}_j$ is the estimated join cost for the j'th stimuli.

$$\widehat{y}_j = \widehat{C}^c_{lin} = \sum_i w_i^c \cdot d_i(j), \tag{7.10}$$

where the weights w_i are the estimated regression coefficients, and $d_i(j)$ is the i'th distance measure calculated for the j'th stimuli.

A weakness with applying linear regression to binary data, is that the assumption for the hypothesis testing relies on normally distributed residuals e_j. In this experiment this problem was possibly somewhat reduced, due to that five listeners were used to evaluate each stimuli, leading to that the average ratings were quantized to six levels, $0, 0.2, 0.4, 0.6, 0.8, 1$.

7.3.2.2 Logistic regression

The estimation of the posterior probability $P(G|d_i \ldots d_n)$, can be identified as a classical problem in binary pattern recognition problem, and is a well known research field, for example related to speech recognition and neural networks. This section will therefore only give a brief introduction to logistic regression [105], which is a widely used approach to this kind of problem. In logistic regression [105], the posterior probability is modeled by a logistic function, which would give a cost function expressed as:

$$\widehat{C}^c(d_i \ldots d_n) = \widehat{P}(D|d_i \ldots d_n) = l(\xi) = \frac{e^{\xi}}{1 + e^{\xi}}, \tag{7.11}$$

where $l(\xi)$ is the logistic function, and ξ is a linear discriminant function.

$$\xi = \tilde{w}_0 + \sum_i \tilde{w}_i d_i, \tag{7.12}$$

where $\tilde{w}_i$ are the weights of the discriminant function. The logistic function can be interpreted as a squashing function, squashing the cost function to the interval [0,1], as shown in Figure 7.8.

As in linear regression, the criterion in the logistic regression is to minimize the squared error between the independent variable and the listeners' ratings of the stimuli. Hence, the approach can be interpreted as a linear

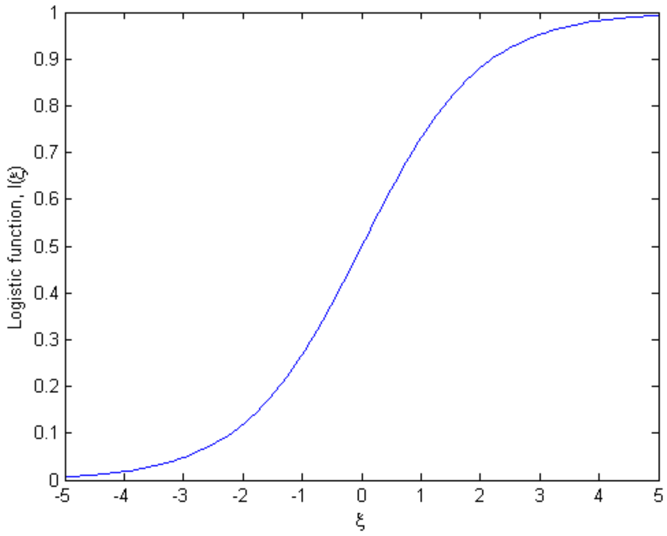


FIGURE 7.8: The logistic function.

regression approach using the natural logarithm of the log ”odds” as the independent variable:

$$\xi = ln\left(\frac{P(D|d_i \ldots d_n)}{P(G|d_i \ldots d_n)}\right), \tag{7.13}$$

which can be shown by inserting Eq. 7.13 into Eq. 7.11. As in stepwise linear regression, hypothesis testing can be applied for determining whether a feature should enter the join cost function or not. The strategy of using the speech synthesizer to generate relevant stimuli for the perceptual experiment should be applied when using this approach, as different sets of test stimuli cold give very different results. For the case of logistic regression, a likelihood ratio test [99] can be applied to select the distance measures.

7.3.2.3 Method 2: The probabilistic approach

In the listening test in this chapter, discontinuities due to both pitch and spectral discontinuities were present in the stimuli. The estimation of the conditional probabilities $P(G|d_i)$ in Eq. 7.8 is hence difficult. For a demonstration of the probabilistic approach, the posterior probabilities for the pitch difference, $P(G|D_{F0})$, and the spectral distance measure, $P(G|D_{mfcc})$, were estimated by the use of logistic regression. It should be noted that logistic regression was applied to estimate the conditional probabilities separately in this approach, and not to estimate the join probability as described above.

Before the logistic regression was performed, a linear transformation of the distance measures were performed

$$\bar{d}_i = \frac{d_i - \mu_i}{\sigma_i}, \tag{7.14}$$

where μ_i was the estimated mean, and σ_i was the estimated standard deviation for the i'th distance measure d_i. The posterior probability for each feature was then estimated by logistic regression, using Eq. 7.11 with $\xi = w_0^i + w_1^i \cdot \bar{d}_i$. In this case, two features, D_{F0} and D_{mfcc}, were used.

Stimuli that had a large probability for a pitch discontinuity were removed from the data set before the logistic regression was applied to the MFCC feature. Similarly, the stimuli with a high probability for a discontinuity due to the MFCC feature was removed before logistic regression was applied to the f_0 feature.

The probabilistic join cost function was finally calculated by Eq. 7.8

$$\widehat{C}^c_{prob}(D_{F0}, D_{mfcc}) = 1 - \widehat{P}(G|D_{F0}) \cdot \widehat{P}(G|D_{mfcc}) \tag{7.15}$$

7.3.3 Results

When applying stepwise linear regression on all the data from the perceptual experiment, D_{F0} and D_{cep} were selected for */e:/* when applying a significance level of 0.05. However, D_{mfcc} could be used with almost the same performance. For */A:/* D_{F0} and D_{mfcc} were selected at significance level 0.05. This was the reason for selecting the D_{mfcc} and the D_{F0} feature for the join cost function in this comparison.

The estimates of the posterior probabilities $\hat{P}(G|D_{F0})$ and $\hat{P}(G|D_{mfcc})$, estimated from the /e:/-stimuli, are shown in Figure 7.9.

The ROC-curves for the estimated join cost function are shown in Figure 7.10 for /e:/ and in Figure 7.11 for /A:/.

From the ROC curves, it is seen that the linear and the probabilistic join cost functions were about equally good for fitting the observed data, which would be more or less expected due to the dominance of discontinuities due to pitch differences, and also due to that the difference between a linear and a nonlinear discriminant function would probably be small when using only two features.

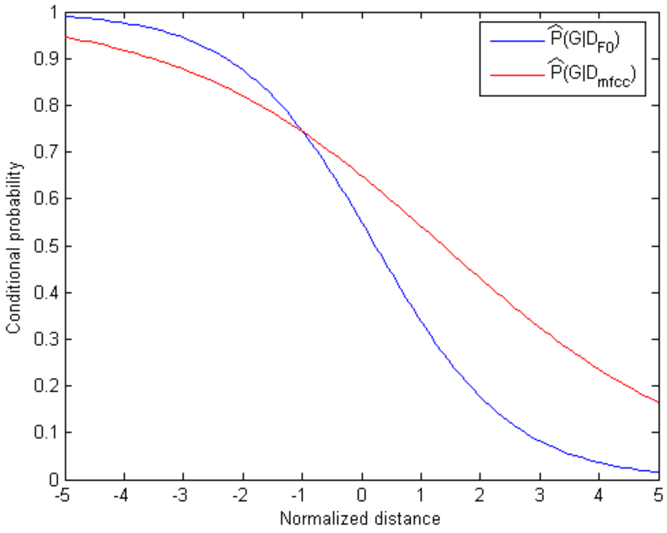


FIGURE 7.9: The estimates of the posterior probabilities $\hat{P}(G|D_{F0})$ and $\hat{P}(G|D_{mfcc})$.

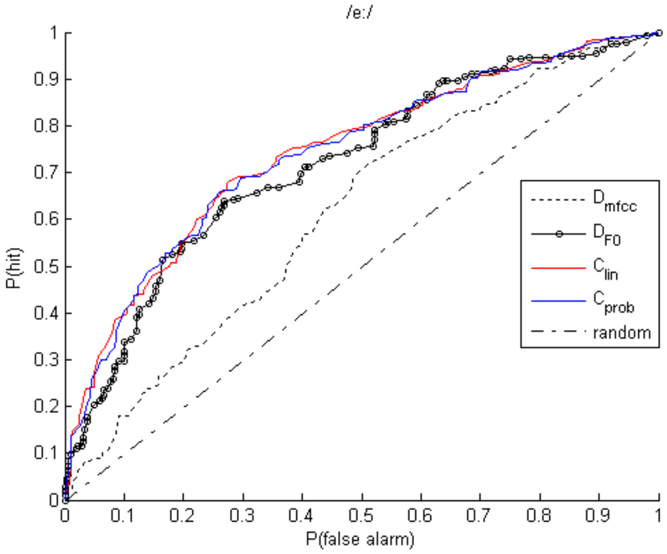


FIGURE 7.10: ROC-curves for the linear join cost function and the probabilistic approach for /e:/. The ROC curve for D_{F0} and D_{mfcc} are also added for comparison.. $P(hit)$ is defined as in Eq. 7.2, and the false alarm rate $P(false\ alarm)$ is defined as in Eq. 7.3.

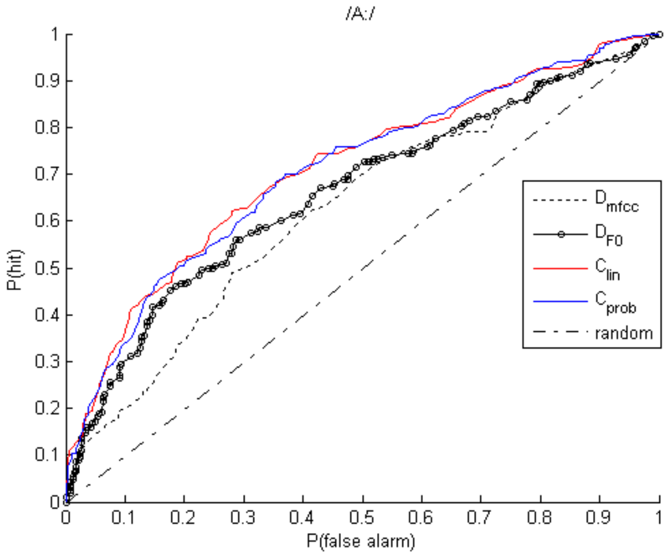


FIGURE 7.11: ROC-curves for the linear join cost function and the probabilistic cost function for /A:/. The ROC curve for D_{F0} and D_{mfcc} are also added for comparison.

7.3.4 Discussion

In the listening test in this experiment and in similar perceptual experiments, the correlation between objective distance measures and the human detection of discontinuities has been reported to be relatively low. One reason could be that not all relevant features are included in the cost function. The use of several uncorrelated features could possibly improve the detection. Examples of possible features could be features on voice quality[63], spectral tilt, formant positions, formant bandwidths and formant energies.

A motivation for using a nonlinear probabilistic approach for the join cost function design is that the linear regression approach is known to have some theoretical weaknesses for modeling probabilities. Specifically, due to the strictly linear model, the probability (or join cost) will turn greater than one for large distances, while for small distances it could possibly turn negative. If many features are included in the join cost function, it will hence be difficult to obtain a fair score for the probability of an audible discontinuity due to "noise" from irrelevant features. If using many features in the cost function, the need for a nonlinear cost function would increase. A fair score for the probability of an audible discontinuity is also needed for a fair integration with the target cost function in a unit selection synthesis system. A probabilistic join cost function has hence some nice theoretical properties, as it goes to unity for large distances and towards zero for small distances.

A possible weakness with the strategy of using the synthesizers target cost function to generate listening test stimuli, is that the resulting join cost function will be too dependent of the specific synthesis system, and hence have poor generalization properties. The use of a probabilistic approach and isolated perceptual experiments on each feature could perhaps lead to a cost function that could be applied for different voices and sound types. Possible weaknesses with the probabilistic approach are that it depends on the assumption of uncorrelated features and the difficult task of estimating the probability of an audible discontinuity.

7.3.5 Summary

In this section, a probabilistic approach for join cost function design is proposed. The approach is based on designing perceptual experiments on isolated features to estimate the probability of an audible discontinuity given the specific feature. Another proposed approach is based on using the specific synthesizer's target cost function for generating the stimuli for the perceptual experiments. Then either a linear or a nonlinear join cost function can be trained by using stepwise linear regression or stepwise logistic regression, using standard hypothesis testing to select relevant features for the join cost function. If many features are included in the join cost function, a nonlinear join cost function would probably be the better choice, as it theoretically can balance the different features more flexibly than a linear cost function.

Chapter 8

Robust pitch synchronous speech analysis

In this chapter, topics related to the analysis step for unit selection synthesis is presented. Specifically, a robust pitch synchronous speech-processing algorithm is presented. The preprocessing algorithm presented in this chapter was originally intended for the estimation of parameters for a frame-based pitch synchronous unit selection system applying speech modification by a harmonic model. However, the algorithm can in principle be used for any kind of pitch synchronous speech processing. Special for this algorithm is the use of the *zero phase instants*, defined as the instants of zero phase for the first harmonic component, as the analysis instants. The algorithm will therefore be referred to as the zero phase algorithm (ZP-algorithm).

The outline of this chapter is to first present the ZP-algorithm. Then, the selection of a pitch estimator for the ZP-algorithm is discussed, comparing three different pitch estimators with focus on robustness to pitch halving errors. In a final section, the pitch estimators are compared to a reference pitch estimate, where the reference pitch estimate was obtained by a manual inspection of an automatic labeling of pitch marks.

8.1 A robust pitch synchronous speech-processing algorithm

The starting point for the work with the ZP-algorithm was the approach proposed by Stylianou [42] for estimating the parameters of the harmonic model by WLS estimation. The use of the WLS parameter estimation, described in Section 4.2.2.1, requires a pitch adaptive processing, meaning that the frameshift and the analysis window duration depend on the pitch estimate. Specifically, the duration of voiced speech frames is defined to be two (estimated) pitch periods and the frameshift in voiced regions is defined to be one pitch period. The pitch adaptive processing concept introduces an additional challenge with respect to the robustness of the pitch estimation. A robust estimation of the pitch is especially important when speech modification is used in the synthesis system, because gross pitch errors can cause severe distortion in pitch-modified speech. Two new concepts have been introduced in the ZP-algorithm.

- The use of zero phase instants as analysis instants for pitch synchronous speech processing.
- The use of a self-validation procedure to validate the reliability of the pitch estimation at run time of the algorithm.

In addition, the algorithm was extended to estimate glottal closure instants and the negative peaks of the speech signal. A motivation for these three subjects will be described respectively.

A fundamental issue when concatenating speech units in unit selection synthesis is to avoid phase mismatch at speech unit boundaries. For example, in [50] a post-processing step was proposed to avoid the phase mismatch problem. In this post-processing step, the estimated harmonic phases of each voiced speech frame were propagated by linear wave propagation so that the phase of the first harmonic component of each frame became zero, as described in 4.2.4. This post-processing step can be avoided if a strictly pitch synchronous analysis is applied. In the ZP-algorithm, the analysis instants are therefore moved to the sample nearest to the position of zero phase for the first harmonic component. These time instants will be referred to as the discrete zero phase instants. The post-processing step proposed in [50] is then no longer needed. The zero phase instants also define a set of pitch synchronous pitch marks for the voiced speech, which can be applied directly for speech modification. A pitch synchronous approach could lead to less variation in the parameter estimates due to minimizing the effect of the position of the analysis instant relative to the pitch cycle. This could lead to better conditions for interpolation and smoothing of parameters along the time axis.

The use of a prior pitch estimate can be used to increase the robustness of the pitch estimation, but only if the prior pitch estimate is reliable. Note that a prior pitch estimate is implicitly used in many processing algorithms, as the duration of the analysis window is normally set on the basis of the pitch estimate of the preceding speech frame. The use of the pitch estimate of the previous frame as a prior estimate could be error prone as a poor initial pitch estimate could affect the estimation of the next frame, and in worst case lead to a pitch estimation "deadlock". A typical example of a "deadlock" is when a high pitch is estimated in a turbulent region of the speech signal, which for example can occur at unvoiced to voiced boundaries. A pitch adaptive algorithm blindly trusting the prior could then be stuck at estimating a high pitch, due to that the next frame size is set too short for the pitch estimator to estimate the correct pitch. A problem in pitch adaptive processing is hence events of irregular voiced speech, like for example creaky voice, where the speech signal is voiced, but where the pitch is not well defined from a signal processing view. Due to this problem, a self-validation of the pitch estimate was implemented. The idea is that the algorithm continuously should validate its current pitch estimate during the run of the algorithm. If a successful validation can be performed, the algorithm can both increase the robustness of the pitch estimate by applying the assumption of a smooth pitch curve in regular regions of the speech signal, and also avoid gross pitch errors and pitch estimation deadlocks when passing through irregular or turbulent regions of the speech signal.

The estimation of zero phase instants was also used as an intermediate step to obtain an estimate of both glottal closure instants and the negative peaks of the speech signal. The glottal closure instants of the speech signal were needed in order to apply the TD-PSOLA speech modification approach, which relies on using analysis instants in the neighborhood of the glottal closure instants [51]. The estimated negative peaks of the speech signal were also applied as pitch marks in the evaluation of the pitch estimators, which will be described in the last section of this chapter.

8.1.1 The ZP-algorithm

A flow chart of the preprocessing algorithm is shown in Figure 8.1. A minus superscript is used to denote variables that are intermediate or prior estimates. A short summary of the algorithm is given below.

The ZP-algorithm processes the speech sentence by sentence. Each sentence is processed frame by frame in a simple left to right manner. The frameshift in the algorithm is defined to be one pitch period in the voiced regions, while a fixed frameshift is used in the unvoiced regions. The choice of frameshift in the unvoiced regions should be based on the average pitch of the speaker in order to obtain smooth transitions from the unvoiced regions to the voiced regions.

The first module in the ZP-algorithm is voicing estimation. This module will be described further in Section 8.1.1.2. Next, if the analysis frame is estimated as voiced, a robust pitch estimator is applied to obtain an initial estimate of the pitch. A pitch estimator optimizing the SNR of a harmonic model was initially used in the ZP-algorithm. However, a HNR-based estimator, modified to avoid pitch halving errors, was found to be more effective. The choice of pitch estimator will be discussed further in Section 8.2.

Next, a self-validation of the pitch estimate is performed, referred to as the regular/irregular decision in Figure 8.1. The pitch validation procedure is described in more detail in a separate flow chart when this module is described in Section 8.1.1.3. Next, the analysis instant for the speech frame is moved to the zero phase instant. This procedure is described in detail in the next section. After the zero phase instant estimation, the pitch of the new analysis speech frame is estimated, and the speech frame is resized according to the refined pitch estimate. The speech frame size was set to be $2 \cdot N_0 + 1$, where N_0 was the estimate of the pitch period rounded to the nearest whole sample. Finally, voiced speech processing is performed, including estimation of parameters for a harmonic model.

As a final step in the loop, the initial analysis instant for the next speech frame is set. It should be noticed that the results from the pitch validation was applied also at this step. That is, the initial position of the next analysis instants was set by using a frameshift of one estimated pitch period, using the last reliable pitch period estimate available.

8.1.1.1 Centering the analysis frame at the zero phase instant

The continuous time zero phase instants, $t_{zp}(i)$, are defined as the time instants where the first harmonic component has zero phase. The discrete zero phase instants, which are used as analysis instants, $n_a(i)$, are defined as the continuous zero phase instants quantized to the nearest sample instants. That is, given an initial speech frame, the analysis instant is moved from the initial position to the sample nearest to the estimated continuous time zero phase instant.

Given an initial speech frame, the first step is to estimate the harmonic amplitudes and phases. In principle, only the phase of the first harmonic component is needed for moving the analysis instant to the zero phase instant. However, in this algorithm the parameters for all the harmonic components were estimated all at once by the WLS estimation (see Section 4.2.2.1). Omitting frame notation for simplicity, and representing time in

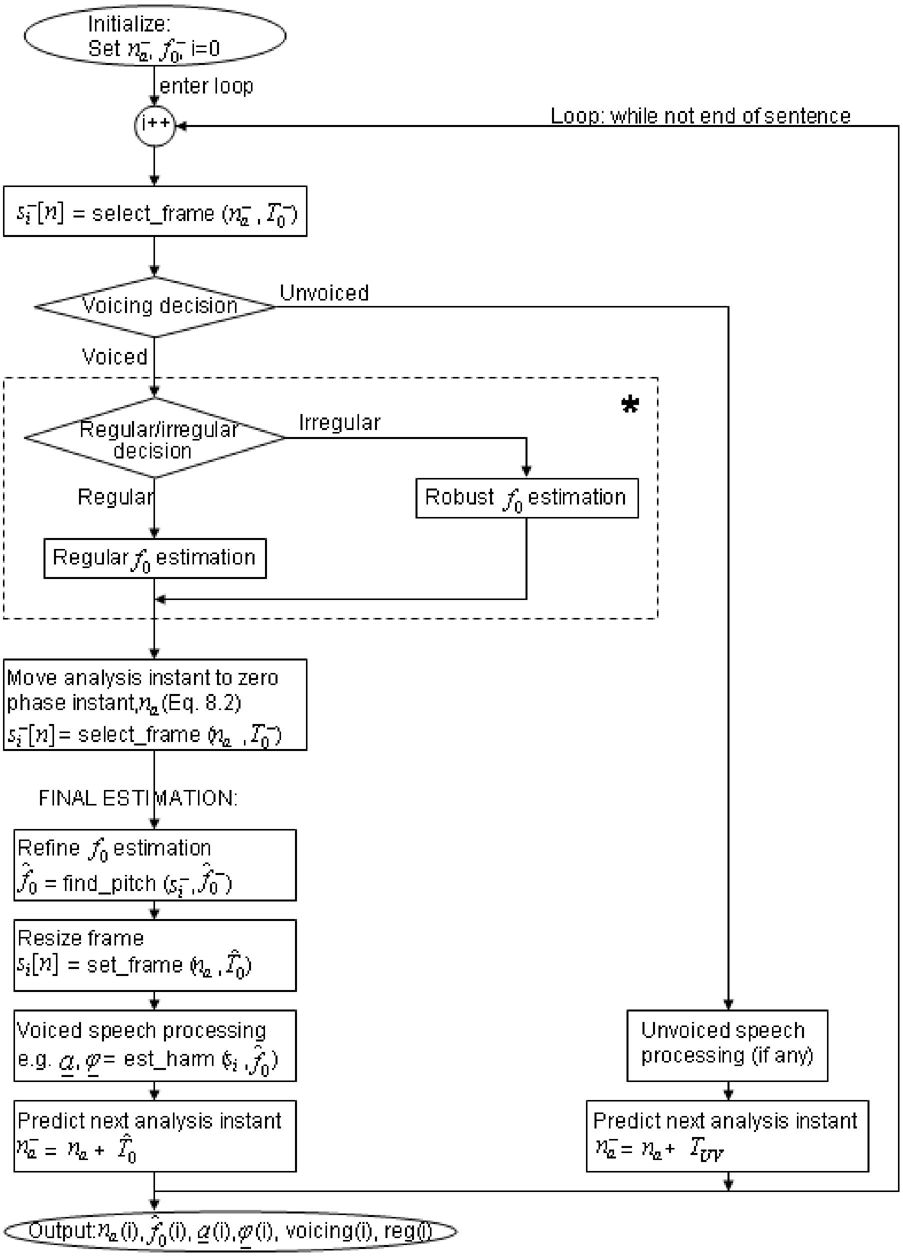

Initialize:
Set n_a^-, f_0^-, i=0
enter loop
i++
Loop: while not end of sentence
$s_i^-[n]$ = select_frame (n_a^-, T_0^-)
Voicing decision
Unvoiced
Voiced
*
Regular/irregular decision
Irregular
Regular
Robust f_0 estimation
Regular f_0 estimation
Move analysis instant to zero phase instant, n_a (Eq. 8.2)
$s_i^-[n]$ = select_frame (n_a, T_0^-)
FINAL ESTIMATION:
Refine f_0 estimation
$\hat{f}_0$ = find_pitch (s_i^-, $\hat{f}_0^-$)
Resize frame
$s_i[n]$ = set_frame (n_a, $\hat{T}_0$)
Voiced speech processing
e.g. $\underline{a}$, $\underline{\varphi}$ = est_harm (s_i, $\hat{f}_0$)
Predict next analysis instant
n_a^- = n_a + $\hat{T}_0$
Unvoiced speech processing (if any)
Predict next analysis instant
n_a^- = n_a + T_{UV}
Output: n_a(i), $\hat{f}_0$(i), $\underline{a}$(i), $\underline{\varphi}$(i), voicing(i), reg(i)

FIGURE 8.1: Flowchart of the ZP-algorithm. n_a denotes the analysis instant, $\hat{f}_0 = 1/\hat{T}_0$ denotes the pitch estimate, $s_i[n]$ denotes the i'th speech frame, $\underline{\boldsymbol{a}}$ and $\underline{\boldsymbol{\phi}}$ denote the harmonic amplitudes and phases respectively. A minus superscript is used to denote initial or temporary estimates. The additional output variables voicing and reg, refer to the binary voicing decision and the binary regular/irregular decision.

the unit samples, the continuous time zero phase instant was estimated as

$$t_{zp} = n_a^- - \frac{\phi_1^- \cdot Fs}{2\pi \hat{f}_0^-}, \tag{8.1}$$

where n_a^- is the initial analysis instant (center of speech frame) in whole samples, $\hat{f}_0^-$ is the estimated pitch, ϕ_1^- is the estimated phase of the first harmonic component when using n_a^- as the analysis instant, and F_s is the sample frequency.

In general the problem of finding the discrete zero phase instant for a speech frame can be solved iteratively. First, the analysis instant is moved from the initial position to the sample instant closest to the estimated continuous time zero phase instant.

$$n_a = [t_{zp}], \tag{8.2}$$

where n_a is the new analysis instant, t_{zp} is defined in Eq. 8.1, and [] denotes a rounding operation to the nearest whole sample. When a new analysis instant is obtained, a new speech frame can be defined using n_a as the frame center. Then a new estimate of the pitch, the first harmonic phase and the zero phase instant can be obtained. This process can be repeated until the position of the discrete zero phase instant (analysis instant) has converged. However, in most cases, this iterative procedure converges in one step. In a few cases, when the analysis frame is irregular, the zero phase instant might not converge within the frame at all. However, in these cases the speech is of a turbulent nature, and centering the analysis window around the high energy peaks of the waveform could be advantageous from an *analysis by synthesis* (See Section 2.4) point of view.

A robust method which ensures a close enough sampling of the analysis instants also in irregular areas of the speech is to do exactly one iteration step, and to limit the shift from the initial position to be maximum half a period. An example of estimated zero phase instants is shown in Figure 8.2.

8.1.1.2 Estimation of voicing

The first task of the speech-processing algorithm is the estimation of voicing, see Figure 8.1. For applying the algorithm easily to new voices, the voicing estimator should ideally work well with a minimum need for manual tuning of the estimator. The voicing measure has however not been the main focus in the work on this algorithm. An approach dependent on some manual tuning was therefore used in the ZP-algorithm. The approach was

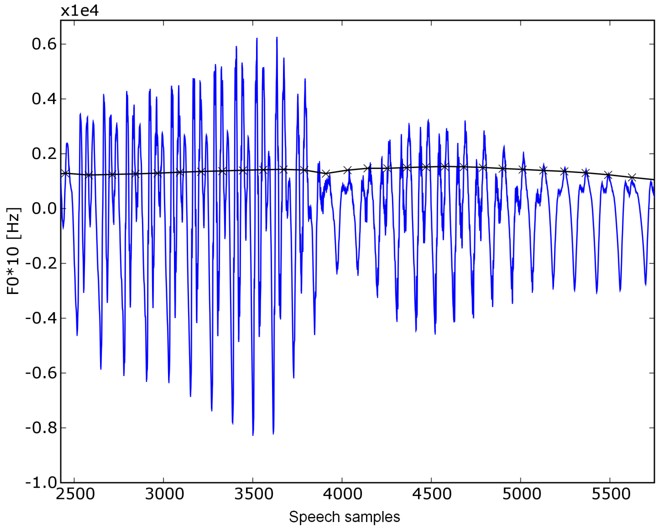


FIGURE 8.2: An example of the tracking of zero phase instants from a speech segment from a male speaker. The 'x'-marks show the estimated zero phase instants.

based on a manually tuned threshold for the largest eigenvalue of the estimated speech frame correlation matrix, which can be related to a measure of spectral flatness [9]. The largest eigenvalue will be small if no strong complex sines are present in the speech frame, as in the unvoiced frames, while for voiced frames the largest eigenvalue would be large, due to one or more strong complex sines.

It was observed that this voicing measure sometimes had problems with classifying both high energy plosive bursts and low energy voiced frames correctly. This problem could be reduced by combining the eigenvalue-based voicing measure with the HNR-based voicing measure [79] or by using the normalized correlation coefficient of unit delay (lag 1) [100]. Another method to improve the voicing estimate was to use the observation that the speech had a rapid increase in the largest eigenvalue at unvoiced to voiced boundaries, while at the end of voiced segments, the largest eigenvalue decreased slowly. This observation motivated to use a higher threshold of the largest eigenvalue for entering the voiced state than leaving the voiced state. This approach has a smoothing effect on the voicing estimate, as the algorithm tries to stay in its current voicing state, and only change its state when enough "evidence" is available. This approach led to less jumps back and forth in the voicing state at transition regions between voiced and unvoiced speech.

Errors in the voicing detection were mainly observed to be in the transition regions between unvoiced and voiced speech. For a better time resolution of the voicing measure, and hence a more accurate detection of voicing in the transition regions, detection of glottal closure instants by phase analysis [62; 61], would probably be required.

8.1.1.3 Detection of irregular speech

The detection of irregular frames, or alternatively the validation of the pitch, was based on that the estimated pitch and the distance between zero phase instants should be consistent. The distance between zero phase instants was defined as

$$T_{zp}(i) = t_{zp}(i) - t_{zp}(i-1), \tag{8.3}$$

where $t_{zp}(i)$ is the continuous time zero phase instant of the i'th frame, defined by Eq. 8.1.

Speech frames were labeled as regular if the distance to the previous zero phase instant could be predicted from the estimated pitch with a given accuracy.

$$|T_{zp}(i) - Fs/\hat{f}_0^{-}(i)| < T_{LIMIT}, \tag{8.4}$$

where $T_{zp}(i)$ is the distance between zero phase instants, Fs is the sample frequency, and T_{LIMIT} is a defined threshold. The optimal threshold would depend on the pitch of the voice in use. For example, for a female voice this threshold was manually tuned to be about 3 samples, while for a male voice with a lower average pitch the threshold was tuned to be almost 10 samples. To avoid a manual tuning of this threshold for different speakers, a possible method could be to let this threshold be a chosen percentage of the average pitch period for the speaker.

As a convention, the first voiced frame of a voiced segment was always set to be irregular. The purpose of this convention was to improve the robustness of the pitch estimation at unvoiced to voiced boundaries.

For improved robustness of the regular/irregular decision, it was made somewhat harder for the algorithm to step into the regular mode than to stay in the regular mode. This was implemented by also checking for pitch consistency for the speech frame succeeding the first possible regular frame.

$$|T_{zp}(i+1) - Fs/\hat{f}_0^{-}(i+1)| < T_{LIMIT} \tag{8.5}$$

This implied that the calculations that normally should have taken place in the next step of the loop had to be calculated. It should be noted that these calculations were only applied for the purpose of this "forward check", and new calculations were made in the next step of the loop. A flow chart describing the pitch validation process is presented in Figure 8.3.

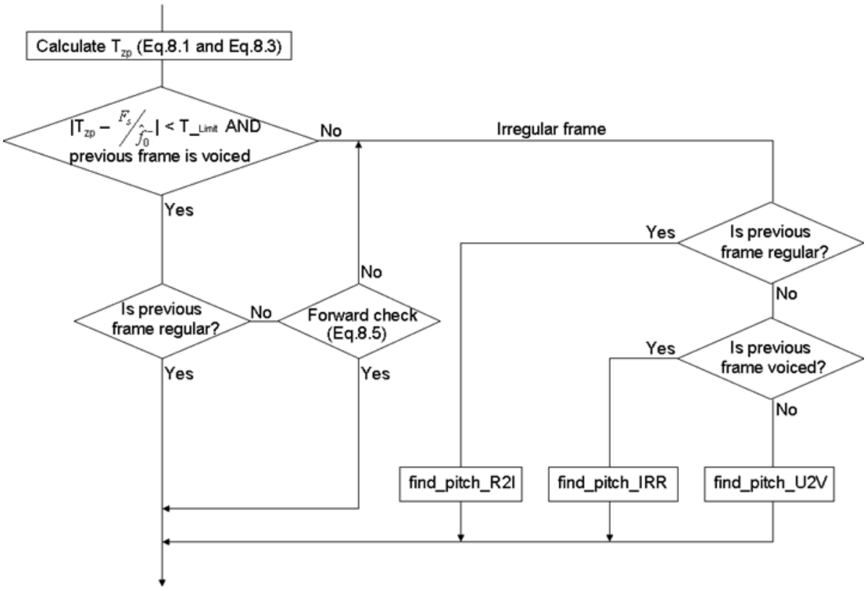

Calculate T_{zp} (Eq.8.1 and Eq.8.3)
$|T_{zp} - F_s/\hat{f}_0| < T\_{Limit}$ AND previous frame is voiced
No
Irregular frame
Yes
Yes
Is previous frame regular?
No
No
Is previous frame regular?
No
Forward check (Eq.8.5)
Yes
Is previous frame voiced?
Yes
Yes
No
find_pitch_R2I
find_pitch_IRR
find_pitch_U2V

FIGURE 8.3: Flowchart of the regular/irregular decision. This flow chart corresponds to the regular/irregular decision, marked by an asterisk(*) in the flow chart of the whole speech-processing algorithm in Figure 8.1.

Three different methods for robust pitch estimation were defined: find-pitch-U2V at unvoiced to voiced boundaries, find-pitch-IRR in irregular regions and find-pitch-R2I at regular to irregular boundaries. Below, these pitch estimation methods will be described.

find-pitch-U2V

At unvoiced to voiced boundaries we have no prior pitch estimate other than some speaker dependent limits obtained from an expected or estimated fundamental frequency range. A long duration window was therefore used to obtain an estimate of the average pitch. The duration of this analysis frame should be long enough for a robust estimation of the smallest possible pitch (longest period) in the predicted pitch range of the speaker. The analysis frame was extended only into the voiced speech region by keeping the first sample of the speech frame fixed.

When a long duration analysis window is applied, it is important that the pitch estimator is unambiguous to pitch halving and doubling. The robust average-HNR method, which will be described in Section 8.2.3.4, was used for this estimation. It should be noted that the long duration analysis window only was applied for the robust pitch estimation, while a window of two periods based on the robust pitch estimate was used in the final voiced speech processing block (See Figure 8.1).

find-pitch-IRR

When the algorithm is in an irregular region of the speech signal, the situation is the same as at an unvoiced to voiced boundary, as a reliable prior pitch estimate does not exist. Hence, a long duration window and a robust pitch estimator were also applied in this case. This situation is however a bit different from the unvoiced to voiced boundary case, as an irregular frame could also occur at the end of a voiced segment. The duration of the analysis window was therefore heuristically set to be four periods, based on the last reliable pitch estimate available. The starting point of the extended analysis speech frame was also in this case fixed. In irregular regions, the pitch might not be well defined. However, in normal cases the sinusoidal modeling procedure will be able to model these waveform segments anyhow. Pitch modification of long duration irregular segments would however be risky, and could lead to severely distorted speech. An example of creaky voice in the middle of a segment is shown in Figure 8.4

find-pitch-R2I

Another cause for pitch estimation errors was experienced to be rapid changes in the vocal tract filter, which for example may occur at phoneme boundaries. The waveform shape could then change rapidly, leading to possible gross pitch estimation errors. This is typically a problem when the duration of the analysis window is short, as the pitch estimator does not "see" the whole picture. This problem is related to the vocal tract interaction problem or formant interaction problem [6]. Reported approaches for reducing the effect of the vocal tract filter on the pitch estimation are for example to apply lowpass filtering [71] or amplitude compression [6].

In the areas of regular speech, it is reasonable to assume a relatively smooth pitch curve as a function of time. This can be tested by checking the

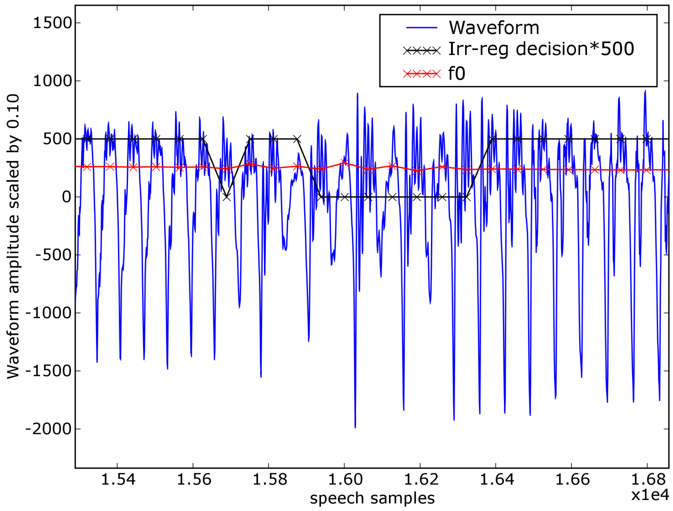


FIGURE 8.4: An example of an irregular speech region (in this case creaky voice) in the middle of a vowel segment from a female voice. The estimated zero phase instants are marked as ’x’. The binary irregular/regular descision is represented as a vector of either 0 (irregular) or 1 (regular), scaled up for this plot.

rate of change in the pitch contour relative to a threshold. In principle, this test could have been performed for all frames where the previous frame was regular. However, it was assumed that the pitch validation approach would detect a gross pitch error by proposing an "irregular" frame, and this test was hence only performed at regular to irregular boundaries. If an unlikely change in the pitch contour slope was found, two methods were applied to check the estimate. The first method was to estimate an average pitch estimate using a long duration window. The second method was to apply an inverse filtering approach and then estimate the pitch based on the inverse filtered speech. An inverse filtering approach could possibly give an improvement if a rapidly changing vocal tract filter is the problem, while the average pitch estimate obtained from a longer duration window could help for a rapidly changing vocal tract filter and could also give better robustness in case of short duration segments of noise or irregular speech. The pitch estimate that ensured the best continuity of the pitch curve was chosen in these cases. It should be emphasized that this technique should only be applied when the previous frame is estimated as a regular speech frame, otherwise the pitch estimation algorithm could get into a pitch estimation deadlock. An example of the vocal tract filtering problem is shown in Figure 8.5.

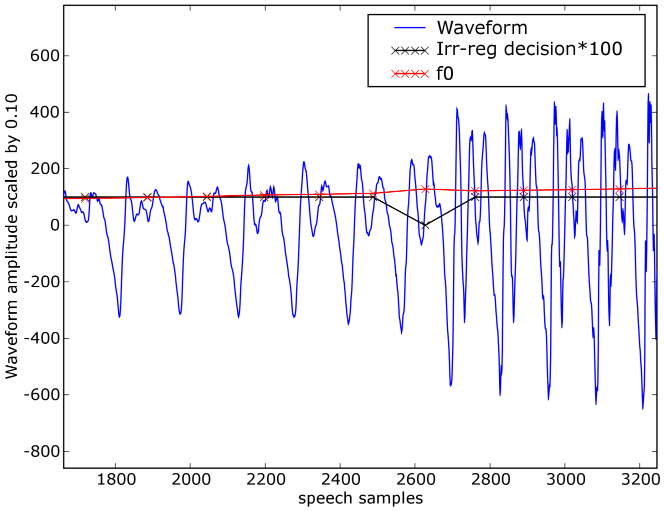


FIGURE 8.5: An example of the vocal tract interaction problem for the male voice. Due to the rapid change in the speech waveform, the regular pitch estimate fails and the processing algorithm goes into irregular mode. At the first irregular frame, the algorithm applies the pitch estimator defined for the regular to irregular boundary. At the next frame, it applies the pitch estimator defined for irregular regions, before it returns to regular mode and applies the regular pitch estimator.

8.1.1.4 Estimation of glottal closure instants

The ZP-algorithm was also used to estimate the glottal closure instants and the negative peaks of the speech signal.

The estimation of glottal closure instants was performed by detecting the negative peaks of the inverse filtered speech frames. The inverse filtering approach was similar to iterative adaptive inverse filtering (IAIF) [14]. The inverse filtering approach is described in detail in Section 9.2.1.2. When analyzing a typical glottal flow derivative waveform, see Figure 2.2, it was seen that the zero phase instant of a typical glottal flow derivative signal was approximately at the positive waveform peak, about midway between the glottal closure instants, but skewed a little to the right. For speech signals there would be some variation in the location of the zero phase instants relative to the glottal closure instants due to the influence of the vocal tract filter. However, in most cases the zero phase instant would be located approximately midway between two glottal closure instants. Assuming a coarse pitch estimate is known, the glottal closure instant preceding a zero phase instant can be found by searching for the minimum of the inverse filtered speech signal in a range around the position where the phase of the first harmonic is $-\pi$ (half a period). A Blackman-Tukey filter was applied to avoid high frequency noise in the inverse filtered signal, as proposed in [12]. The search range was set as wide as one estimated period, in order to make the estimation independent of the position of the zero phase instant. An example of an estimated glottal closure instant is shown in Figure 8.6. The appropriateness of this approach would rely on a robust coarse pitch estimate and on the appropriateness of the inverse filtering approach.

The negative peaks of the inverse filtered signal are correlated with the negative peaks of the speech signal, as seen in Figure 8.6. The negative peaks of the speech signal were therefore estimated by searching for the most negative peaks of the speech signal in a narrow interval around the estimated glottal closure instants. The range of the interval was chosen heuristically to be 20% of the estimated pitch period. In Figure 8.7, an example of the result of this approach is shown for a voiced segment of the speech signal from a male speaker.

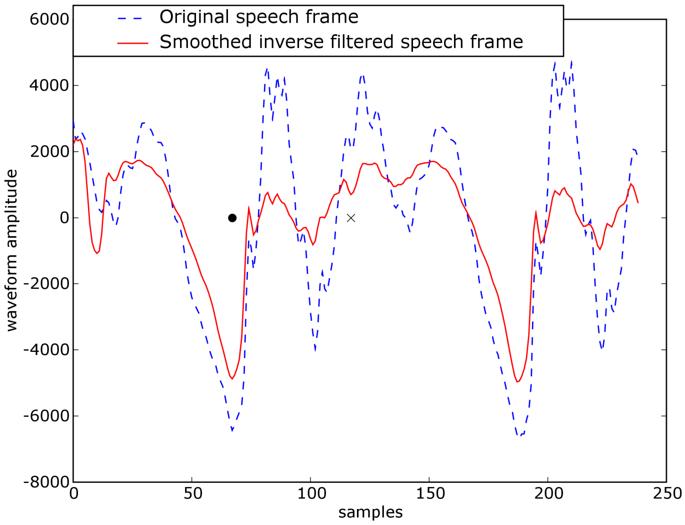


FIGURE 8.6: An illustration of the approach used for estimation of glottal closure instants. The search region corresponds to the part of the smoothed inverse filtered speech frame preceding the zero phase instant. The discrete zero phase instant is marked by an 'x', while the (discrete) negative peak of the smoothed inverse filtered signal is marked by a filled circle. A speech frame from an /e:/ sound from a male voice was applied for this example.

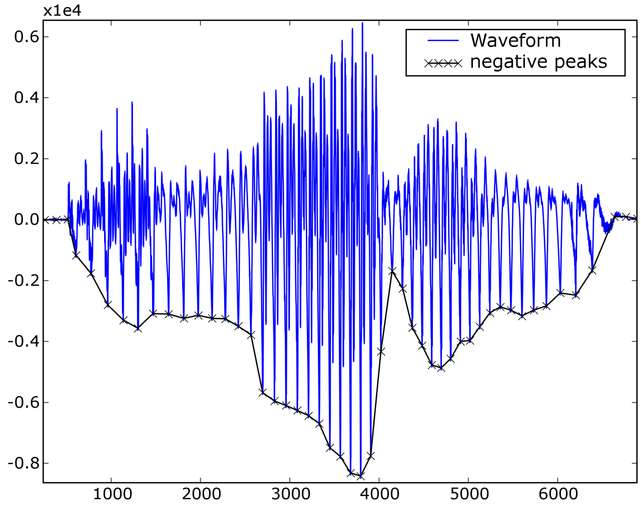


FIGURE 8.7: An example of the estimation of the most negative peaks in the waveform for a voiced region from a male voice.

8.1.2 Evaluation of the ZP-algorithm

An important part of the algorithm is robust pitch estimation. Two approaches has been applied to evaluate the algorithm with respect to possible gross pitch estimation errors. The first approach was to analyze the regions marked as irregular by the pitch validation method. The second method was to listen to pitch modified speech, as proposed in [6]. A harmonic model was used for pitch modification of the speech, which is further described in the experiment chapter on speech modification, Chapter 9. If a gross pitch estimation error occurs over several frames, it will lead to severe audible distortion in the modified speech. Note that it would in general not suffice to listen at resynthesized (reconstructed) speech, due to that the sinusoidal modeling procedure can give high quality resynthesized speech even when the pitch is poorly estimated.

In a comparison of pitch estimators, described in Section 8.3, the pitch period estimate obtained from the distance between zero phase instants is compared to the pitch period estimate obtained from the manually inspected negative peaks of the speech waveform. This is also an evaluation of the use of zero phase instants as pitch marks relative to the use of the negative peaks of the speech signal as pitch marks.

8.1.2.1 The applied speech databases

The ZP-algorithm was tested on four speech databases, consisting of two male and two female voices. Two of the speech databases, one male and one female voice, were recorded as a part of the speech synthesis project FONEMA [102]. These speech databases were recorded as two of several small speech databases in a study intended for the selection of voices for the recording of a larger speech database for unit selection synthesis. The male voice is referred to as the *t15* speech database and the female voice is referred to as the *t16* speech database. These two databases were recorded from the same text manuscript, containing 519 sentences with 25527 phones. The databases were phonetically labeled, or segmented, by an automatic phone label alignment approach [103]. The recording procedures for these databases were based on the experience with a previously obtained (larger) speech database recording [104] in the FONEMA project. Speech was recorded in a studio at 48 kHz, and then downsampled to 16 kHz. An "expressive" speaking style was desired in these recordings.

The two other speech databases were the speech database *Prosdata* [85], described in Chapter 7, and a male diphone voice [101] consisting of 1480 unique diphones, read with a relatively flat pitch. These two databases were phonetically labeled by a manual phonetic segmentation of the speech.

8.1.3 Results

The regions labeled as irregular by the pitch validation method were experienced to be either (correctly estimated) irregular regions, e. g. turbulent regions of the voiced speech, or errors in the pitch or voicing estimation. Hence, manual inspection of the estimated irregular regions was important for identifying problem cases in the pitch estimation.

Gross pitch errors led to severely distorted modified speech, which was the main motivation for the work on the pitch validation module and robust pitch estimation. Pitch modified speech synthesized from the parameters obtained in the ZP-algorithm was experienced to avoid severely distorted speech. The general quality of the pitch modified speech will be further discussed in Chapter 9.

The results of the manual inspection of pitch marks will be presented in more detail in relation to the comparison of pitch estimators in Section 8.3. The result of the manual inspection of the estimated pitch marks was however quite promising, with 0.001 % gross errors for the male voice and 0.006 % gross errors for the female voice. This means that in the regular regions of the speech signal, the negative peaks of the speech signal could be detected quite robustly. If the pitch estimate in the irregular regions (unfairly) were counted as pitch estimation errors, the errors would be 1.5 % and 1.6 % respectively. Some factors that could affect these results are however that the manual thresholds on voicing and pitch validation were mainly tuned on these sentences, and also that more sentences and voices should have been tested. However, it should be noted that the amount of creaky voice or other types of irregular speech would be very speaker dependent. Hence, it would in general be impossible to do a fair comparison of different speech-processing algorithms for different speech databases, except perhaps if also irregular regions in the speech signal are detected. For a fair comparison to other algorithms, a shared manually labeled speech database would be needed.

8.1.4 Discussion

An alternative to using zero phase instants as analysis instants in the pitch synchronous speech-processing algorithm could be to use the estimated glottal closure instants. These instants could be estimated directly and replace the function of the zero phase instants in this algorithm, or they can be estimated as in this section by using the zero phase instants as an intermediate step.

If voicing errors and pitch estimation errors are rare, the regular/irregular detection can possibly be used as an automatic detector of creaky voice or other irregular speech. For example, if several consecutive frames are estimated as irregular, the algorithm could mark these regions as irregular. For robustness, the detected irregular regions could be manually inspected.

The use of pitch marks as analysis instants in the voiced regions and fixed analysis instants in the unvoiced regions may give a border mismatch problem at unvoiced to voiced boundaries. This is due to that the unvoiced frame shift might not fit with the first zero phase instant of the voiced segment. No audible distortion was however experienced from possible border mismatch. One explanation could be that the high energy events are well modeled, and that speech often is of a relatively turbulent nature at these boundaries anyway. Moving the analysis instant of the first voiced speech frame to the nearest zero phase instant is a consistent way of splitting unvoiced and voiced segments for a concatenative synthesizer.

8.1.5 Summary

In this section, a pitch synchronous speech-processing algorithm for the estimation of parameters for a harmonic model has been presented. The speech-processing algorithm depends on estimating zero phase instants in voiced speech regions, which are defined as the instants of zero phase for the first harmonic component in a harmonic model. These time instants can be used as pitch marks/analysis instants for speech synthesis and modification. They can also serve as an intermediate step for the estimation of the glottal closure instants or the negative peaks of the speech signal.

A self-validation method for the pitch estimate was proposed as one method for avoiding gross pitch errors related to pitch adaptive processing algorithms. The method validates the reliability of the pitch estimate at run time by comparing the frame-based estimate to the estimated distance between zero phase instants. With this pitch validation approach, the algorithm could both exploit the property of a relatively smooth pitch curve in regular regions of the speech signal, and at the same time avoid using the pitch estimate of the previous frame as a prior estimate in irregular/turbulent regions of the speech signal. If pitch and voicing errors are rare, the pitch validation method can be applied as a detector of irregular speech regions, possibly aided by manual inspection of the detected regions for robustness.

8.2 Robust pitch estimation

The topic in this section is robust frame-based pitch estimation for the ZP-algorithm. The main task of the robust pitch estimator is to estimate a coarse pitch, as another pitch estimator can refine this estimate in the final estimation step in the ZP-algorithm, see Figure 8.1. Robustness to pitch halving and doubling, described in Section 6.2, is an important property for this pitch estimator. Secondly, the pitch estimate should not be biased due to using a short duration speech frame. Pitch estimators based on sinusoidal models can fulfill both these criteria. In addition, when using a sinusoidal or a harmonic model for reconstruction of the voiced speech, a sinusoidal pitch estimator can be designed to minimize the squared error to the original waveform, being optimal from an analysis by synthesis point of view. Therefore, the focus has been on sinusoidal pitch estimators in this thesis.

Three pitch estimators have been tested for the ZP-algorithm: an SNR-based estimator, an HNR-based estimator, and an ESPRIT-based estimator. In this section these pitch estimators are described, and robustness to gross pitch errors are discussed. Specially, the HNR estimator is described in detail in this section, as it is very simple and can be made robust to pitch halving errors. In addition, effects from the analysis window size and different variants of the HNR estimator are discussed.

Several approaches have been proposed in the literature to resolve the pitch halving ambiguity [75; 77; 106]. The reason for presenting another estimator robust to pitch halving is its simplicity, and that it does not rely on any prior coarse pitch estimate. In addition, the estimator can be interpreted as an approximation to the SNR-based estimator, being optimal from an analysis-by-synthesis point of view. It should be noted that the focus in this section is on robustness to gross pitch errors. In the next section, a comparison of the different pitch estimators with respect to accuracy and bias is presented.

The outline of the section is to first present the SNR-based estimator. Then the HNR-based estimator is presented, and the method for alleviating pitch halving errors is described. In addition, window effects and variants of the estimator is discussed. Finally, an ESPRIT-based pitch estimator is described. Background topics for this section are presented in Chapter 6.

8.2.1 Pitch estimation based on optimization of SNR for a harmonic model

The first pitch estimator tested for the ZP-algorithm was a pitch estimator based on maximizing the SNR for a harmonic speech model. That is, minimizing the difference between the original signal and the signal reconstructed by a harmonic model.

$$\epsilon = \frac{1}{N_w} \sum_{n=1}^{N_w} |s_i[n] - \hat{s}_i[n]|^2, \tag{8.6}$$

where ϵ is the mean square reconstruction error, $s_i[n]$ is the original speech frame of length N_w, and $\hat{s}_i[n]$ is the estimated signal by the harmonic model. The inverse SNR can from Eq. 8.6 be obtained by dividing by the power of the original signal. Hence, minimizing ϵ is equivalent to maximizing the SNR.

The SNR can then be calculated for all possible candidate $\hat{f}_0$ in the pitch range of the speaker, using the harmonic model equation (Eq. 4.6) to calculate $\hat{s}_i[n]$. This implies that the harmonic amplitudes and phases have to be estimated for every possible $\hat{f}_0$ in the search. The maximum of SNR as a function of $\hat{f}_0$ then provides an optimal pitch estimate for the speech frame in an analysis by synthesis point of view.

This SNR-based estimator is conceptually the same as the estimator proposed by Griffin[37], described in Section 6.4.1. However, a difference is that a time domain measure of the reconstruction error was applied in the variant in this thesis. This implies that also the estimated phases of the harmonic components contribute to the error.

In the variant of this pitch estimator used in this thesis, the amplitudes and phases of the harmonic model were estimated by the WLS-estimation, described in Section 4.2.2.1. Due to the computational complexity of the WLS-estimation, the search was very computationally demanding. It was hence not feasible to do a full search over the whole f_0-range of the speaker. Instead, a smaller search range in the neighborhood of a prior coarse pitch estimate was applied. This could be performed due to that the function $\text{SNR}^{-1}(f)$ showed to have a convex shape for regular voiced frames, at least in a certain range around the minimum value. A search for the optimal $\hat{f}_0$ was therefore performed by first evaluating $\text{SNR}^{-1}(f)$ on a sparse grid around an initial estimate f_1. Then a high resolution grid around the first minimum value was used to obtain a high resolution minimum of $\text{SNR}^{-1}(f)$.An example of $\text{SNR}^{-1}(\hat{f}_0)$ evaluated for every integer pitch candidate is shown in Figure 8.8.

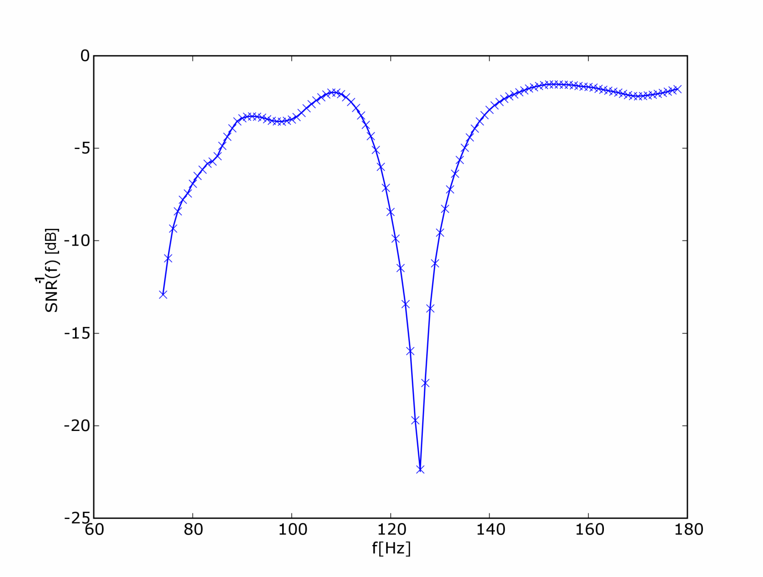


FIGURE 8.8: Inverse SNR [dB] for the harmonic model as a function of $\hat{f}_0$ [Hz]. Calculated for a speech frame from an /e:/-sound from a male speaker

Inside a voiced segment, the prior pitch estimate f_1 was set to the pitch estimate of the previous frame. However, this approach relies on a robust initial estimate f_1. This motivates the importance of an effective robust unambiguous pitch estimator at least for providing a robust initial estimate.

8.2.2 The f0/2-spectrum

For the example in Figure 8.8, it is seen that the function $\mathrm{SNR}^{-1}(f)$ has a well defined local minimum, which is easily found if the prior value is not too far from the minimum. However, decreasing the value of $\hat{f}_0$ will increase the number of parameters in the harmonic model as the harmonic frequencies then will be more closely spaced. This explains why small values of $\hat{f}_0$ in addition to the correct $\hat{f}_0$ would give high SNR's. Hence, the pitch halving ambiguity is not automatically resolved. Intuitively, the frequency $\hat{f}_0 = f_0/2$ would be the next minimum of $\mathrm{SNR}^{-1}(\hat{f}_0)$. Theoretically, the harmonic model could then contain the same harmonic components as a harmonic model using f_0 as the basis frequency, and in addition contain equally many sine waves to model the aperiodic content of the frame. Hence, a harmonic model using $\hat{f}_0 = f_0/2$ should give a higher SNR than a harmonic model using $\hat{f}_0 = f_0$. If we assume that the duration of an analysis frame is exactly two pitch periods, the harmonic model using $\hat{f}_0 = f_0/2$ as the basis frequency in the harmonic model would have the same frequency spacing as a $2N$-point DFT, where N is the length of one period, and $2N$ is the length of the speech frame. Then the speech signal can be perfectly reconstructed, yielding an infinite SNR for $f_0/2$.

This reflects that the harmonic model is capable of exact reconstruction of only one period of the speech, unless the speech is perfectly periodic. When the number of parameters are doubled, as when using $\hat{f}_0/2$ as the basis frequency, the harmonic model is capable of exact reconstruction of two periods.

The harmonic amplitude spectrum when using $\hat{f}_0/2$ as the basis frequency in the harmonic model equation, referred to as the $\hat{f}_0/2$-spectrum, is shown in Figure 8.9.

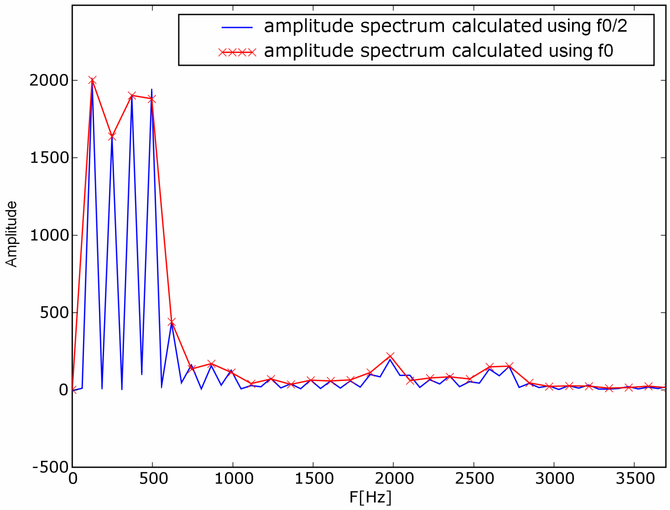


FIGURE 8.9: Harmonic amplitude spectra using respectively $\hat{f}_0/2$ and $\hat{f}_0$ as the basis frequency in the harmonic model

This spectrum will for a quasiperiodic signal (as speech) have high power for every second component, being estimates of the harmonic components, and low power for the odd (aperiodic) components. Hence, the ratio of the even amplitudes to the odd amplitudes of the amplitude spectrum can be used as an indicator for possible pitch halving.

Because the reconstruction by a harmonic model is perfect when using a model consisting of all the sine wave components in the $\hat{f}_0/2$-spectrum, and the harmonic (even) components approximately correspond to the power of the reconstructed signal by the harmonic model, the HNR would also be an approximation of the SNR for a harmonic model.

8.2.3 A pitch estimator based on the HNR

As motivated above, the ratio of the power of the even amplitudes to the odd amplitudes of the $f_0/2$-spectrum can be defined as an estimate of the harmonic-to-noise ratio:

$$HNR(f) = \frac{\sum\limits_{i=1}^{L/2} a_{2i}^2(f/2)}{\sum\limits_{i=1}^{L/2} a_{2i-1}^2(f/2)}, \tag{8.7}$$

where $a_i(f/2), i = 1 \ldots L$ are the estimated amplitudes of the $f/2$-spectrum for a given frequency f. For $f = f_0$, this ratio equals the power of the harmonic content divided by the power of the aperiodic content. This estimate of the HNR provides an estimate with low computational complexity related to the iterative algorithms proposed in [77; 78]. The use of the $f_0/2$-spectrum has the nice property that no peak picking procedure is needed, due to that every second component in the $f_0/2$-spectrum would correspond to the "peaks" in the spectrum when $f = f_0$. The HNR-based pitch estimate is then found by searching for the frequency that maximizes HNR(f).

$$\hat{f}_0 = \arg\max_f HNR(f) \tag{8.8}$$

The amplitudes of the $f_0/2$-spectrum can be estimated either by the WLS-method or by an 2N-point DFT, where N is the candidate pitch period length. The WLS-estimation is computationally expensive due to the matrix inversion, and it also runs into problems with a singular matrix when the length of the speech frame is shorter than the number of parameters [36]. A DFT-based estimation of the $f_0/2$-spectrum is both faster and robust, and would therefore be a better choice when the $f_0/2$-spectrum is applied in a search for the fundamental frequency.

When using a DFT to estimate the $f_0/2$-spectrum, the function HNR(f) can be estimated by varying the frequency sampling N in the DFT to obtain HNR(f)=HNR($2Fs/N$), where Fs is the sampling frequency. This provides a simple implementation of the pitch estimator as a search for possible fundamental periods, needing one DFT for each candidate pitch, where the pitch estimate finally is found by an argmax operation.

Given an input search range $[f0_{min}, f0_{max}]$ and an input speech frame, referred to as $x[n]$ instead of $s_i[n]$ for this example, the python code (close to pseudo code) for the algorithm can be written:

```
N_vector=range(2*int(Fs/(f0_max)),2*int(Fs/f0_min)+1)

for k in range(length(N_vector)):
    X=fft(x[n],N_vector[k])
    a=(2/N_vector[k])*abs(X[0:(N_vector[k]/2)+1])
    max_coeff=length(X)/2
    HNR[k]=sum(a[2:max_coeff:2]**2)/(sum(a[1:max_coeff:2]**2))
end

index=argmax(HNR)
f0=2*Fs/N_vector[index]
```

Fs is the sample frequency, *a* is the vector of estimated amplitudes, and the HNR is calculated by Eq. 8.7. The function *int(x)* returns the largest integer value less than or equal to x, the function *abs(x)* takes the absolute value, and the function *fft(s,N)* calculates the DFT of the signal s using N frequency bins.

Note that when using a DFT, the HNR can only be obtained for integer values of N, yielding a resolution in the period of half a sample. However, higher resolution can for example be obtained by using the WLS-estimation, or alternatively by interpolation of the function $HNR(f)$ around the maximum.

8.2.3.1 Resolving the pitch halving ambiguity

The unmodified HNR pitch estimator still suffers from the pitch halving ambiguity as it can be interpreted as a classic comb filter approach (Section 6.4). To alleviate this, we propose to use the average harmonic to noise ratio, referred to as the average-HNR estimator.

$$HNR_{avg}(f) = \frac{1}{L/2} HNR(f), \tag{8.9}$$

where $L/2$ is the number of harmonic components contributing to the HNR. This modification can be motivated by the shape of the $f_0/2$-spectrum. For the true $\hat{f}_0$, and only for the true $\hat{f}_0$, the $f_0/2$-spectrum would have exactly one noise component for each harmonic component. Hence, the average HNR would be largest for the true $\hat{f}_0$. For example, if a period twice as long as the true period N were given as the input to the estimation of HNR in Eq. 8.7, a 4N point DFT would be evaluated. Then there will be three noise components for each harmonic component instead of only one. Hence, the average ratio of an assumed harmonic component to the neighboring component would decrease, while a measure of the total harmonic to noise ratio would be more or less unchanged.

Assuming a perfectly periodic signal, a halving of the pitch would double the number of harmonic components without modifying the HNR. The average-HNR estimator hence penalizes pitch halving with a factor of about 0.5. Taking the logarithm on both sides of Eq. 8.9 yields

$$log(HNR_{avg}) = log(HNR(f)) - log(\frac{L}{2}), \tag{8.10}$$

which show that the averaging leads to a term penalizing low pitch candidates. Hence, the estimator has a similar shape as the analytic result obtained by Quatieri, described in Section 6.4.2. An example of the HNR estimator applied to a synthetic signal is shown in Figure 8.10. The unmodified HNR estimator generates a peak of about the same height for every halving of the pitch (doubling of the period), while the average-HNR estimator penalizes the long periods and avoid possible pitch halving.

The purpose of the penalizing term is to resolve the pitch halving ambiguity robustly without biasing the estimate. The effect of bias for this estimator will be small, as the function HNR(f) normally has a high sharp peak around f_0. This possible bias of the HNR estimator is further discussed in the comparison of pitch estimators presented in Section 8.3.

8.2.3.2 Effect of window size

When using a DFT based approach, it is known that a window shorter than two periods would smear the spectrum[5]. Hence, if the window is too short, a low HNR would be estimated even for the true pitch. This fact implies that if the window size is set to two periods of a reliable initial pitch estimate, the regular HNR estimator would not be vulnerable to neither pitch halving nor doubling of the pitch. Instead, it would depend on the initial pitch estimate to be reliable. An example of this effect is shown in Figure 8.11, where the peak for the double period (halving of pitch) vanishes when a two period window size is applied. The window size hence

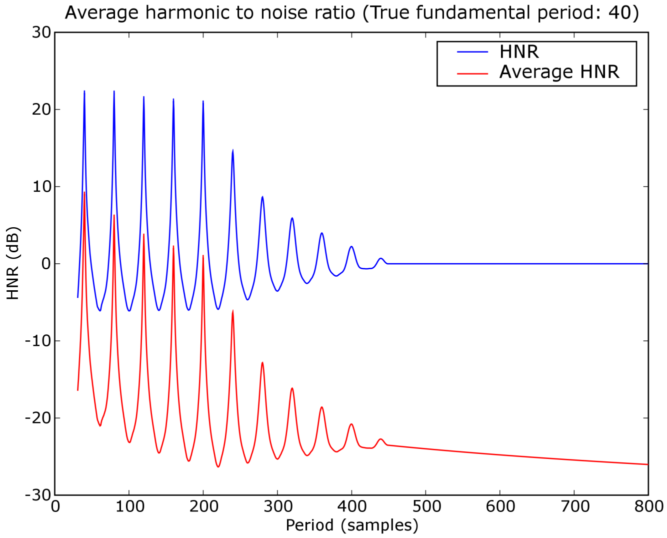


FIGURE 8.10: HNR for a synthetic signal consisting of three perfectly periodic harmonically related sinusoids in white noise.

defines a lower bound on the f_0-estimate that can be obtained with this method.

The effect of the window size was tested by applying the pitch estimator to a synthetic signal with known fundamental frequency, using different analysis window sizes in the estimation. The synthetic signals in this test were constructed to be perfectly periodic, consisting of N harmonic sine waves, where the amplitudes and phases of the harmonic sine waves were set according to spectrum estimates obtained from a male vowel. White noise was added to the signal to avoid an infinite HNR.

As expected, the pitch estimate could be biased when the window length was significantly shorter than two pitch periods due to increased smearing of the spectrum. A slightly too small window was however experienced to still give an estimate relatively close to the true value. In the ZP-algorithm, the pitch estimation method is to be applied either to a relatively long duration analysis window to obtain a coarse pitch estimate, or to shorter windows of about two periods when the prior pitch estimate is trusted.

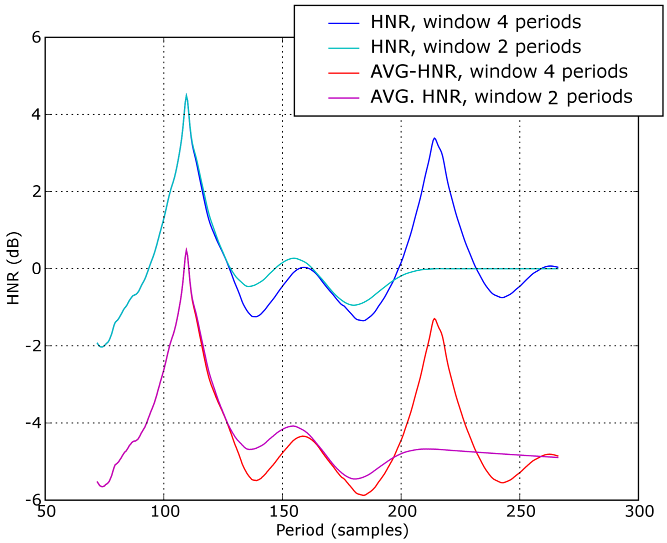


FIGURE 8.11: HNR and average HNR for the same speech segment using two different sized windows with the same center. The speech frame was from the middle of an /i:/-sound from a male speaker with an estimated pitch of 146Hz (Period of 109.6 samples).

8.2.3.3 Sensitivity to the noise energy estimate

In testing on synthetic signals, one weakness with the HNR-based pitch estimator was discovered when it was applied to extremely periodic (harmonic) signals. When the noise energy is extremely low, the estimator is very sensitive to small changes in the noise power due to that the HNR-estimator depends on a ratio.

The same example as used for motivating the use of the average HNR (Section 8.2.3.1), can be used to illustrate this problem: When evaluating the HNR in Eq. 8.7 for (the true) $f_0/2$, there will be three noise components for every harmonic component instead of one noise component for every harmonic component. Hence, one third of the noise amplitudes would be "erroneously" counted as harmonic amplitudes. The total harmonic power would hardly be changed as the true harmonic components would dominate, but a problem could be that the noise power is very low relative to the harmonic power. Hence, a small change in the noise power could provide a large change in the power ratio, which could exceed a factor of two under some conditions, where frame size was experienced to be a condition. This effect probably occurs rarely for real speech, as the signal needs to have very low noise power, being equivalent to that the speech signal must be extremely harmonic/periodic. However, for the female voice a pitch halving error was seen also when using the average-HNR estimator. This suggests considering variants of the estimator that could alleviate this problem.

8.2.3.4 Variants of the estimator

As mentioned in the previous section, a small weakness with the average HNR estimate in Eq. 8.9 is that all even amplitudes contribute to the harmonic energy, even when some harmonics really should contribute to the noise energy. One way to avoid this is do a voiced/unvoiced decision for each harmonic component in the $f0/2$-spectrum, and only let the harmonic components classified as voiced contribute to the harmonic energy. This was implemented in a variant of the estimator by checking the level of each harmonic amplitude in the $f_0/2$-spectrum to the level of the adjacent (noise) amplitudes. Specifically, a harmonic component was classified as voiced if the harmonic amplitude was (significantly) greater than the mean of the adjacent noise components,

$$a_{2i} \geq 1.1 \cdot \left(\frac{a_{2i-1} + a_{2i+1}}{2} \right), \tag{8.11}$$

where a_i is the $i'th$ amplitude of the f0/2-spectrum, as defined as in Eq. 8.7, and the factor 1.1 was a manually tuned threshold. The modified average HNR-measure was then calculated similarly as the average HNR, except that only the harmonic components classified as voiced contributed to the harmonic energy, while all the remaining components contributed to the noise energy.

This approach was experienced to avoid errors in extremely period synthetic signals, and also for an observed case of pitch halving error for the female voice. This variant is however only a conjectured improvement of the HNR-based estimator, as the number of observed problem cases was too small to say if this variant really gave an improvement in practice. This estimator was applied when a robust initial average pitch estimate from a relatively long duration analysis window was needed in the ZP-algorithm.

Other variants of the estimator which probably could avoid the problem of the estimator being sensitive to the noise estimate, is to use a difference measure instead of a ratio, or to maximize the average harmonic to signal ratio, HSR, calculated as the power of the even amplitudes divided on the power of all the amplitudes. A comparison of different variants of the HNR estimator is described in Section 8.3.3.2.

8.2.4 An ESPRIT-based pitch estimator

The third pitch estimator tested for the ZP-algorithm was an ESPRIT-based pitch estimator. The ESPRIT algorithm, described in Section 6.6.1, has in the literature been proposed for pitch estimation in [82; 83]. The approach used in this thesis was however implemented independently of the algorithms described in [82; 83], and the approach described here is therefore slightly different. Due to these implementational differences, the approach used in this thesis is described and discussed here.

The ESPRIT-based pitch estimator would avoid pitch doubling and halving errors if the estimated ESPRIT frequencies approximately correspond to the harmonic frequencies. However, the estimation of the correlation matrix would be influenced by the speech frame size. Hence, the approach depends on a robust initial pitch estimate. In the approach in this thesis, a $(N \times N)$ correlation matrix was estimated from a speech frame of size $2N+1$, where N was the estimated period. The next step was to estimate the number of sine wave frequencies, which was performed by applying a threshold to the eigenvalues of the estimated correlation matrix. Knowing that the noise eigenvalues ideally should be equal to the noise energy σ_0^2, a threshold on the eigenvalues of the correlation matrix was used to estimate the number of complex sines M. When modeling a regular voiced speech frame with a harmonic model, the SNR was experienced to be higher than 20 dB in normal cases. The ESPRIT estimation should provide about the same SNR for a general sine wave model when using an appropriate number of sine wave components. Hence, this heuristic SNR-level was applied to define an eigenvalue threshold:

$$\sigma_T^2 = \frac{P_s}{10^{SNR/10}} \tag{8.12}$$

σ_T is the eigenvalue threshold, P_s is the power of the speech frame, and SNR is the defined SNR level in dB (chosen to 20dB). M was then estimated as the number of eigenvalues larger than the threshold σ_T, using a lower bound of 6 complex sinusoidal components. Next, the ESPRIT frequencies, $f_1 \dots f_M$, were estimated as described in Section 6.6.1.

Two reasonable alternatives for obtaining a pitch estimate is either to choose the lowest frequency from the ESPRIT frequency vector, f_1, or to use a weighting of the (positive) frequencies in the frequency vector, assuming that the frequency vector contains only harmonic frequencies. In experiments with a harmonic speech model, the pitch estimation based on a weighted average of the ESPRIT frequencies was found to be better from an analysis by synthesis point of view. That is, the pitch estimate based on a weighted average led to a higher SNR of the reconstructed speech by a harmonic model. Hence, an approach using a weighted average of the positive frequencies, $f_1 \ldots f_{M/2}$, was chosen.

Two problems with using a weighted average of the estimated ESPRIT frequencies are the possible occurrence of spurious (noise) frequencies in the ESPRIT frequency vector, and that not all harmonic frequencies would in general be present in the ESPRIT frequency vector. The spurious sine wave components would however have very low power. Hence, to avoid influence from possible spurious frequencies, the positive ESPRIT frequencies were weighted by their respective estimated sine wave amplitudes, $a_1 \ldots a_{M/2}$, where the amplitudes were estimated using WLS-estimation.

The assumption for pitch estimation by a weighted average is that the ESPRIT frequency vector corresponds to the harmonic frequencies. However, it was experienced that in many cases one or several harmonics were not present in the ESPRIT frequency vector. Hence, to get an unbiased estimate of the pitch, it should also be estimated which harmonic components that are present in the ESPRIT frequency vector. In order to achieve this, an initial or a priori estimate of the pitch was applied. In practice, the lowest ESPRIT frequency, f_1, in the ESPRIT frequency vector was applied as the initial estimate of $\hat{f}_0$. The weighted average was then calculated as:

$$\hat{f}_0 = \frac{\sum\limits_{i=1}^{M/2} a_i^2 \cdot \frac{f_i}{\left[\frac{f_i}{f_1}\right]}}{\sum\limits_{i=1}^{M/2} a_i^2}, \tag{8.13}$$

where f_i are the estimated ESPRIT frequencies, a_i are the estimated amplitudes, [] denotes rounding to the nearest integer, and f_1 is the prior (initial) estimate. That is, each ESPRIT frequency in the weighted sum was divided by its estimated harmonic number to provide an estimate of the pitch. An iteration scheme, setting $f_1 = \hat{f}_0$ and using Eq. 8.13 iteratively was not applied in this thesis, but could possibly be an improvement of this approach when the estimated f_0 is very different from the prior estimate f_1.

The main difference to the approaches described in [82; 83] is that in this approach it is not assumed that all the harmonic frequencies are present in the ESPRIT frequency vector. This is possibly an improvement of the estimator if the prior estimate f_1 is reliable, as the ESPRIT frequency vector in general would not contain all harmonics.

8.2.5 Results

Three pitch estimators were tested for the ZP-algorithm. An SNR-based estimator, an HNR-based estimator, and an ESPRIT-based estimator. If gross pitch errors were avoided, all these three pitch estimators were experienced to be accurate enough for providing transparent resynthesized speech by a harmonic model, and to provide pitch modified speech of relatively high quality. However, the SNR-based estimator was very computationally demanding. Hence, a search in the full pitch range of the speaker was not found feasible in practice. For both the SNR-based approach and the ESPRIT-based approach, the use of prior pitch estimates was found to occationally introduce pitch estimation errors. The average-HNR estimator was found as the most appropriate estimator for the ZP-algorithm because it was not dependent on any prior pitch estimate and could avoid pitch halving errors. A further evaluation of the pitch estimators are described in the next section of this chapter.

8.2.6 Discussion

The SNR-based estimator is optimal from an analysis by synthesis point of view, but it was found too computationally demanding to be used in practice. However, with more processing power or a more effective implementation, the SNR-based estimator is an alternative. Another possibility is to first apply the HNR estimator, and then refine this estimate by the SNR-based estimator.

The HNR-based estimator performs a full search for the pitch in the possible f_0-range, hence it can be regarded as a computationally complex approach. However, it is still relatively effective due to the effective implementation of the DFT algorithm available for most programming languages. More advanced search algorithms can be implemented if a faster pitch estimator is desired, e.g. if the estimator is to be used for real time applications. If a reliable prior pitch is known, the search range of the estimator can be limited to the neighborhood of the prior estimate. However, such an approach would require the use of a pitch validation approach to avoid pitch estimation deadlocks.

The main difference of the average-HNR estimator compared to the method proposed by Quatieri [75] for resolving the pitch halving ambiguity, described in Section 6.2, is mainly that no spectral envelope estimate explicitly has to be estimated. Hence, the heuristic peak picking method for estimating the spectral envelope is avoided. The peak picking approach and the spectral envelope estimate used in [75] also depend on an initial coarse pitch estimate [76]. In [6] it is suggested to apply another pitch estimator in order to estimate a coarse pitch. However, this coarse pitch estimator would also need to be robust. The main difference from the estimator proposed in [77], described in Section 6.5, is that no elaborate iterative algorithm for estimating the harmonic to noise ratio of the glottal source is needed.

8.2.7 Summary

In this section, three pitch estimators have been described. These were an SNR-based estimator, an HNR-based estimator and an ESPRIT-based estimator. The pitch estimator based on the average HNR was found as the most appropriate for the ZP-algorithm. This was due to that it does not depend on any initial coarse pitch estimate, and that the estimator is robust to pitch halving and doubling errors. In addition, the HNR estimator can be interpreted as an approximation to the SNR-based estimator, which is optimal for the reconstruction by a harmonic model from an analysis by synthesis point of view. All three pitch estimators described in this section were accurate enough to provide transparent resynthesis and modified speech of relatively high quality if gross pitch errors were avoided.

A variant of the average-HNR estimator based on an unvoiced/voiced decision of every harmonic component, in addition to a variant of the estimator based on the harmonic-to-signal ratio, were in experiments on synthetic speech like signals found to be even more robust to pitch halving. The pitch estimators and the different variants of the HNR-based estimator are further compared in the next section.

8.3 Comparing pitch estimators

Ideally, an evaluation of different pitch estimation methods should be performed by applying the estimated pitch from a laryngograph signal as a reference, or by using a manually labeled reference [71]. Laryngograph signals were unavailable for the speech databases applied in this thesis. It was therefore decided to evaluate the pitch estimators by a comparison to automatically generated, but manually inspected, pitch marks.

A problem with hand labeled or manually inspected pitch estimates is however that these pitch estimates also are inaccurate. However, the pitch estimates obtained from hand labeled pitch marks should at least give a relatively unbiased reference of the pitch. This can be motivated by that there should exist exactly as many true glottal closure instants as labeled pitch marks.

In addition to a comparison to the pitch period estimated from the manually inspected pitch marks, the pitch estimators were also compared to each other in order to get additional information on the correlation of the estimators. The mean and the standard deviation was therefore calculated for the difference between all pairs of pitch estimators in this test.

8.3.1 The pitch estimators in this comparison

The reference method in this comparison is referred to as the negative peak (NP) method, T_{NP}. This method was defined as the distance between the pitch marks obtained by estimating the negative peaks of the speech signal. The negative peaks were first estimated by an automatic procedure, as described in Section 8.1.1.4, and then manually inspected in order to approximate a human labeling of pitch marks. The manual inspection procedure is described further in Section 8.3.2.1.

In total, 11 different methods in addition to the reference method were evaluated in this comparison. Six of these methods were different variants of the HNR-based method. All the pitch estimators were evaluated using the estimated pitch period of the speech signal, $\widehat{T_0} = F_s/\hat{f}_0$, where the sample frequency F_s of the speech was 16 kHz. The five main estimators in the comparison were:

1. T_{HNR} The average HNR method, applying an interpolation around the maximum HNR to avoid the quantization effect, calculated as described in Section 8.2.3.1.

2. T_{SNR} The SNR method, calculated as described in Section 8.2.1. The average HNR estimate from the speech processing algorithm was

used as an initial estimate, and a range of 10 samples to each side of the initial estimate was searched.

3. T_{ESP} The ESPRIT method, as described in Section 6.6.1

4. T_{ZP} The ZP method, calculated as the distance between zero phase instants, where the zero phase instants were represented in continuous time as in Eq. 8.1.

5. T_{GCI} The distance between glottal closure instants, where the glottal closure instants were estimated by the inverse filtering approach described in Section 8.1.1.4.

The pitch estimators in this test can be divided into two categories, frame-based estimators and pitch mark based estimators. The pitch mark based estimators, T_{NP},T_{GCI} and T_{ZP}, were all based on the distance between estimated time instants (pitch marks) in the speech signal. The sampling of these pitch mark based pitch curves did in general not coincide with the centers of the speech frames in the ZP-algorithm. Linear interpolation of the pitch mark based estimates were therefore applied to obtain estimates that were aligned in time with the frame-based methods.

8.3.1.1 The variants of the HNR estimator

The six variants of the HNR estimator were.

1. T_{FILTER} The same HNR method as used for T_{HNR}, but now applied to the inverse filtered signal, applying the inverse filtering method described in Section 9.2.1.2.

2. T_{NO-AVG} The HNR method, but without the low pitch penalty term, calculated as in Eq. 8.7

3. T_{AVG} The average HNR method, but quantized to half a sample as no interpolation around the maximum HNR was applied.

4. T_{ROBUST} The HNR-method, but applying a voiced/unvoiced decision for each harmonic component, in order to only include those harmonic components that are above the noise level to the calculation of harmonic energy, as described in Section 8.2.3.4

5. T_{DIFF} Similar to the HNR method, but applying the average difference of the harmonic energy and the noise energy instead of a ratio.

6. T_{HSR} Similar to the HNR method, only applying the harmonic to signal ratio instead of the harmonic to noise ratio.

8.3.2 Experimental procedure

The speech databases *t15* and *t16*, described in Section 8.1.2.1, were used in this experiment. Five sentences were analyzed for both the male and the female voice. In order to compare the pitch estimators, all the described pitch estimators were applied to exactly the same speech frames. That is, all the pitch estimates in this comparison were obtained in the final estimation step in the same run of the ZP-algorithm, see Figure 8.1. An independent initial coarse pitch estimate, estimated by the average HNR method, was applied for coarse pitch estimation in the ZP-algorithm. At the same run, the ZP-algorithm also estimated the negative peaks of the waveform, the glottal closure instants, and the zero phase instants. Hence, this approach provided paired observations for the pitch estimators containing no gross pitch estimation errors.

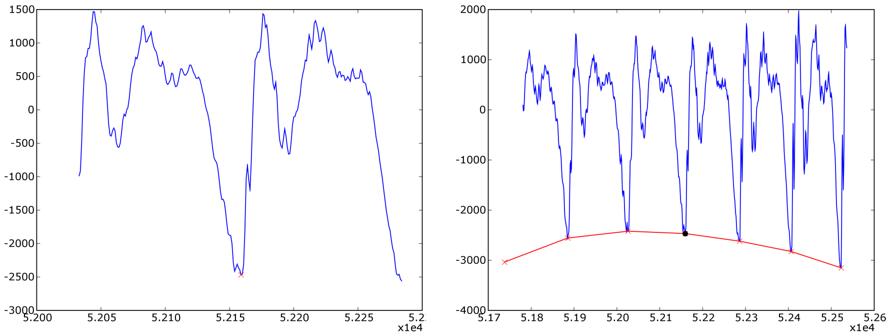
8.3.2.1 Manually inspected pitch marks

A manual check of the estimated negative peaks was conducted to ensure that there were no gross errors in the pitch reference based on these pitch marks. No pitch marks labeled as unaccepted in the manual check were used in the comparison of the pitch estimators.

A simple program was implemented for allowing an efficient and robust manual check of the negative peaks. Each proposed negative peak of the signal was displayed in two figures. One figure with high enough resolution to resolve the most negative sample, and one figure showing more of the context around the proposed negative peak. It was possible to zoom in both figures in case of doubt. An example of the view presented to the labeler is shown in Figure 8.12. For each proposed negative peak the labeler had to enter an integer number between 0 and 2. The number 0 in case of an error, the number 1 in case of accepting the proposed negative peak, and the number 2 in case the periodicity/negative peak was not well defined even from the view of the labeler, which could happen in some cases of irregular speech or very noisy voiced frames. 5 sentences of the female voice and 5 sentences of the male voice were labeled by the author. One sentence took on average about 25 minutes to label.

8.3.2.2 Analysis procedure

A pitch estimator can be characterized by its bias and standard deviation to the true estimate, but since the true pitch is not known, a reference pitch contour has to be used. Although T_{NP} was the manually checked reference, all pairs of pitch (period) estimators were compared with respect



(a) High resolution view

(b) Context view

FIGURE 8.12: Example of the view of the manual labeler (reduced in size). A figure with a resolution of two periods to the left, where the most negative sample is marked with a red cross. To the right a figure showing more of the context of the signal, where the negative peak to be labeled is marked with a black filled circle.

to bias and standard deviation. Given a pitch period contour $T_A[n], n = 0, 1, \ldots, N$, and a (reference) pitch period contour $T_B[n]$, the bias was calculated as:

$$\hat{\mu}_{AB} = \frac{1}{N} \sum_n (T_A[n] - T_B[n]), \tag{8.14}$$

and the standard deviation was calculated as

$$\hat{\sigma}_{AB} = \sqrt{\frac{1}{N-1} \sum_n (T_A[n] - T_B[n])^2} \tag{8.15}$$

To avoid interference of outliers, a maximum absolute difference threshold between the pitch period estimates and the reference method was set to 10 samples. Only the data below this threshold were contributing to the calculation of the mean and the standard deviation. A threshold of 10 samples corresponds to an error of approximately 10 percent of the average period. It should be noted that these outliers are not considered to be gross pitch period errors, just only abnormal large inaccuracies in the pitch estimators or/and in the reference. Typically, these large errors could occur at the beginning and at the end of the voiced segments, which should not be relevant for the measure of the general performance of the pitch estimators.

8.3.3 Results

Most of the negative peaks that were marked as "0" (error) or "2" (undefined) in the manual inspection were at the beginning or at the end of a voiced region. The results of the manual labeling are presented in Table 8.1.

Label	Male Voice	Female Voice
0	2 (0.001 %)	15 (0.006 %)
1	1519 (98.4 %)	2644 (97.8 %)
2	23 (0.015 %)	45 (0.016 %)

TABLE 8.1: Result of the manual check of the automatically detected negative peaks in the waveform. 0 represents errors, 1 represents accepted peaks, and 2 represents speech frames were the periodicity/negative peak was not well defined.

8.3.3.1 Analysis of bias and standard deviation

Bias and standard deviation were calculated as described Section in 8.3.2.2. Removing outliers resulted in that 0.7 % and 1.6 % of the data was removed for the male and the female voice respectively. All the methods had about the same number of outliers. The number of observations for each pitch period estimator before removing outliers were 2482 for the female voice and 1349 for the male voice. For the female voice the average period was 83.8 samples, and for the male voice it was 144.0 samples, using 16 kHz sampled speech signals. The mean and standard deviation for the difference between the pitch period estimators, with outliers removed, are presented in Table 8.2 and Table 8.3 respectively. For better readability, only the main methods are included in these tables, while the variants of the HNR estimator are discussed in a separate section.

From Table 8.2 it is seen that all the pitch estimators are close to unbiased with respect to the manually inspected pitch marks (T_{NP}). A bias of -0.10 and -0.14 samples (16 kHz speech) was measured for the HNR method relative to the NP method for the male and female voice respectively. Applying a hypothesis test with the hypothesis that the mean μ of the difference has a normal distribution $\mu \sim N(0, \frac{\sigma}{\sqrt{N}})$ show that this bias is significantly different from zero at least for the female voice. The probabilities for zero bias were estimated to be 0.085 and $5.9 \cdot 10^{-4}$ for the male and female voice respectively, using a two sided hypothesis test [98]. The

M/F	T_{HNR}	T_{SNR}	T_{ESP}	T_{ZP}	T_{GCI}	T_{NP}
T_{HNR}		-0.08/-0.22	-0.01/-0.23	0.05/-0.23	-0.13/-0.10	-0.10/-0.14
T_{SNR}			0.10/-0.02	0.16/-0.02	-0.02/0.10	0.00/0.06
T_{ESP}				0.11/0.01	-0.10/0.13	-0.07/0.09
T_{ZP}					-0.16/0.12	-0.13/0.10
T_{GCI}						0.01/-0.04

TABLE 8.2: Mean difference (bias) between the pitch period estimators for both a male and a female voice. The bias' are presented as samples of 16 kHz sampled speech. The first column consist of the methods corresponding to method A in Eq. 8.14, and the first row consist of the methods corresponding to method B in Eq. 8.14. The result for the male voice is displayed to the left of the forward slash and the results from the female voice to the right.

M/F	T_{HNR}	T_{SNR}	T_{ESP}	T_{ZP}	T_{GCI}	T_{NP}
T_{HNR}		0.78/0.72	1.57/1.14	2.26/1.39	2.67/2.27	2.13/2.03
T_{SNR}			1.35/1.07	2.24/1.30	2.63/2.18	2.06/1.98
T_{ESP}				1.99/1.28	2.61/2.25	2.14/2.06
T_{ZP}					2.85/2.25	2.47/2.10
T_{GCI}						2.37/2.32

TABLE 8.3: Standard deviation of the difference between paired observations from different pitch period estimators for both a male and a female voice. The standard deviations are presented as samples of 16 kHz sampled speech. The result for the male voice is displayed to the left of the forward slash and the results from the female voice to the right.

bias between the HNR method and the SNR method of -0.08 and -0.22 samples for the male and female voice is hence also significant, as the variance between these two methods are lower. Although the measured bias is statistically significant, a bias of about 0.10 samples is so small that it could be considered as practically unbiased.

If the estimators are unbiased, the better estimator would be the estimator with the least variance to the true estimate. The true underlying pitch period is unknown. However, when assuming that the true underlying pitch period curve is a relatively smooth curve in each voiced region, the pitch period estimators with the highest variance with respect to a smooth curve would also be suspected to have the largest variance with respect to the true estimate. During the procedure of manual pitch marking, it was experienced that the position of the most negative sample in the waveform could be very sensitive to noise and/or waveform shape, and hence errors of several samples could be expected to occur. Inspection of the pitch period curve obtained by the baseline NP-method, shown in Figure 8.13, show that this pitch period curve had large fluctuations relative to a smooth curve in some regions of the signal. Hence, the reference pitch estimate in this test was far from ideal. Probably, an initial low pass filtering of the speech signal [71] before the estimation of the negative peaks could have removed some of the noise in the reference estimate. Also the T_{GCI}-estimator was experienced to have large fluctuations in some regions of the signal. In Table 8.3 it is seen that these two estimators have the largest standard deviation relative to the other pitch period estimators.

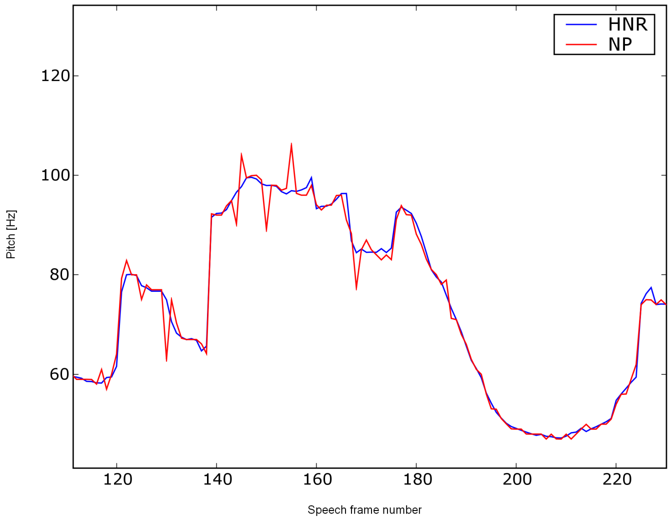


FIGURE 8.13: Example of a piece of the estimated pitch period curve for both the T_{HNR} and the T_{NP} method for the female voice. Note that the pitch period curves here are a concatenation of the pitch period curves for several voiced segments. Hence, the pitch period curves should only be piecewise smooth.

T_{HNR} should theoretically have high correlation with T_{SNR}, which is supported by the low standard deviation in Table 8.3. The difference between T_{HNR} and T_{SNR} is displayed in a histogram for the male voice in Figure 8.14, showing that in about 75% of the cases the absolute difference between these two methods were less than 0.25 samples. For a comparison, the histogram for the difference between T_{HNR} and T_{NP} is shown in Figure

8.15. Of the pitch period curves calculated from time instants related to the pitch cycle, the T_{ZP} method had the lowest standard deviation relative to the frame-based methods.

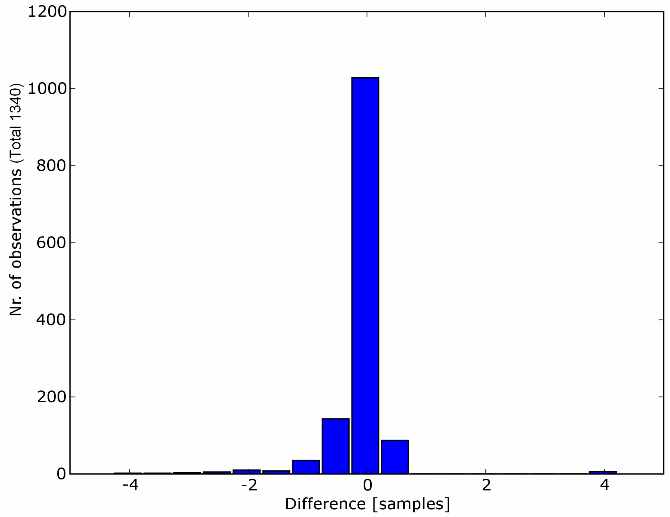


FIGURE 8.14: Histogram of the difference between the HNR method and the SNR method for the male voice. Bins are 0.5 samples wide.

8.3.3.2 Comparing the different variants of the HNR estimator

The variants of the HNR estimator performed relatively similar to the HNR method of the previous section (T_{HNR}), except for T_{DIFF}, which had a bias to the NP-method of -0.89 and -1.03 samples for the male and female voice respectively. This might be due to bias from the low pitch penalty term due to the averaging by the number of harmonic components, as the peak of the difference measure will be much more flat than if applying a ratio. T_{NO-AVG} had only a bias of 0.02 and 0.01 samples relative to T_{HNR}, indicating that the average term in the HNR method leads to insignificant bias. T_{AVG}, T_{NO-AVG}, T_{ROBUST} and T_{HSR} performed practically the same as T_{HNR}, with approximately zero bias and a standard deviation of about 0.50 and 0.40 samples to the T_{HNR} method for the male and female voice respectively. T_{FILTER} and T_{DIFF} had a somewhat larger deviation to both the HNR and the SNR method, with a standard deviation of approximately one sample for both voices. The standard deviation to the NP method was approximately 2 samples for all these methods.

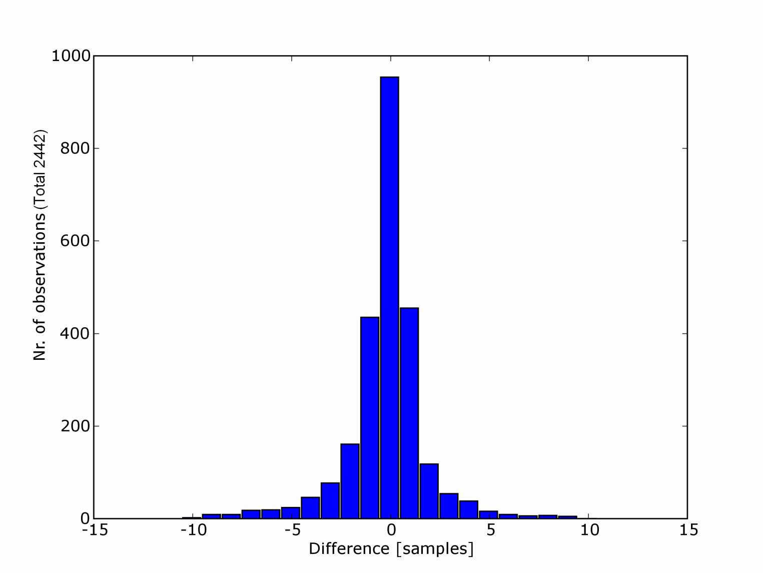


FIGURE 8.15: A histogram plot of the difference between the HNR and NP method for the female voice using all the data (outliers removed). Bins are 1 sample wide with a new bin starting at every half sample.

8.3.4 Summary

In this experiment, it was found that the HNR estimator (T_{HNR}) was practically unbiased to the reference obtained from manually inspected negative peaks of the speech signal, with a bias of about -0.1 samples. It was also seen that the estimate based on the HNR estimator was very close to the SNR-based method, with a standard deviation of only 0.70, and where 75% of the estimates had less absolute difference than 0.25 samples. When displaying the pitch curves, the sinusoidal estimators showed to follow a more smooth curve than the estimator based on the negative peaks in the speech signal, which supports the appropriateness of the sinusoidal estimators.

When comparing different variants of the HNR-based estimator, it was found that it was no significant bias between the HNR estimator with averaging (T_{AVG}) and the HNR-estimator without averaging (T_{HNR}). The T_{HSR} approach and the T_{ROBUST} approach were also found to be equally good as the T_{HNR} method.

Chapter 9

Experiments on speech modification

This chapter will concern experiments on modification of speech using different variants of a harmonic model. It should be noted that the processing algorithm described in Chapter 8 was applied in a preprocessing step to extract speech parameters. This implies that the modification algorithms described in this chapter are strictly pitch synchronous, using zero phase instants as analysis instants.

The outline of the chapter is to first describe the algorithms used for speech synthesis and modification of speech. In the second section, several variants of modification by a harmonic model are described, before a comparison of these variants based on listening to modified speech is presented. In the third section, one of these variants is compared to the widely used TD-PSOLA method in a listening test. In the fourth section, an experiment on applying modification and smoothing in unit selection synthesis is described, including a novel approach for the smoothing of pitch across speech unit boundaries.

9.1 Frame based pitch synchronous synthesis and modification

The first step towards applying modification in unit selection synthesis is speech analysis. The speech analysis and the speech synthesizer described in this thesis was implemented in the programming language Python [107], using a Mysql [87] database for storing speech unit information. Four voices were tested, two male and two female. The speech databases are described in Section 8.1.2.1.

In a preprocessing step, the phonemically segmented speech databases were analyzed sentence by sentence by the frame based pitch synchronous algorithm described in Chapter 8. In the analysis step, described in detail in Section 8.1, the parameters needed for reconstruction by a harmonic model were estimated. For all speech units, the model parameters of the speech frames belonging to a speech unit[1] were saved as attributes to that particular speech unit in the speech unit database. That is, for each speech unit a vector of analysis instants representing the speech frame centers of the speech unit, and a vector of voicing decisions were saved to the database. If a unit contained voiced frames, a matrix of harmonic amplitudes (number of voiced frames × number of harmonic components), a matrix of harmonic phases and a vector of pitch estimates were also saved to the database. In addition, filter coefficients for an AR filter of order 16 (representing the vocal tract filter) were saved for each voiced frame.

Modification of the speech was only performed on the voiced speech frames. The duration of unvoiced sounds changes less than the duration of voiced sounds when the rate of articulation is changed [6]. Thus, (relatively) natural pitch and duration modified speech could be obtained without modifying the duration of the unvoiced segments. For unvoiced speech regions the original speech frames (samples) were saved to the database, resulting in exact reconstruction of the unvoiced speech frames.

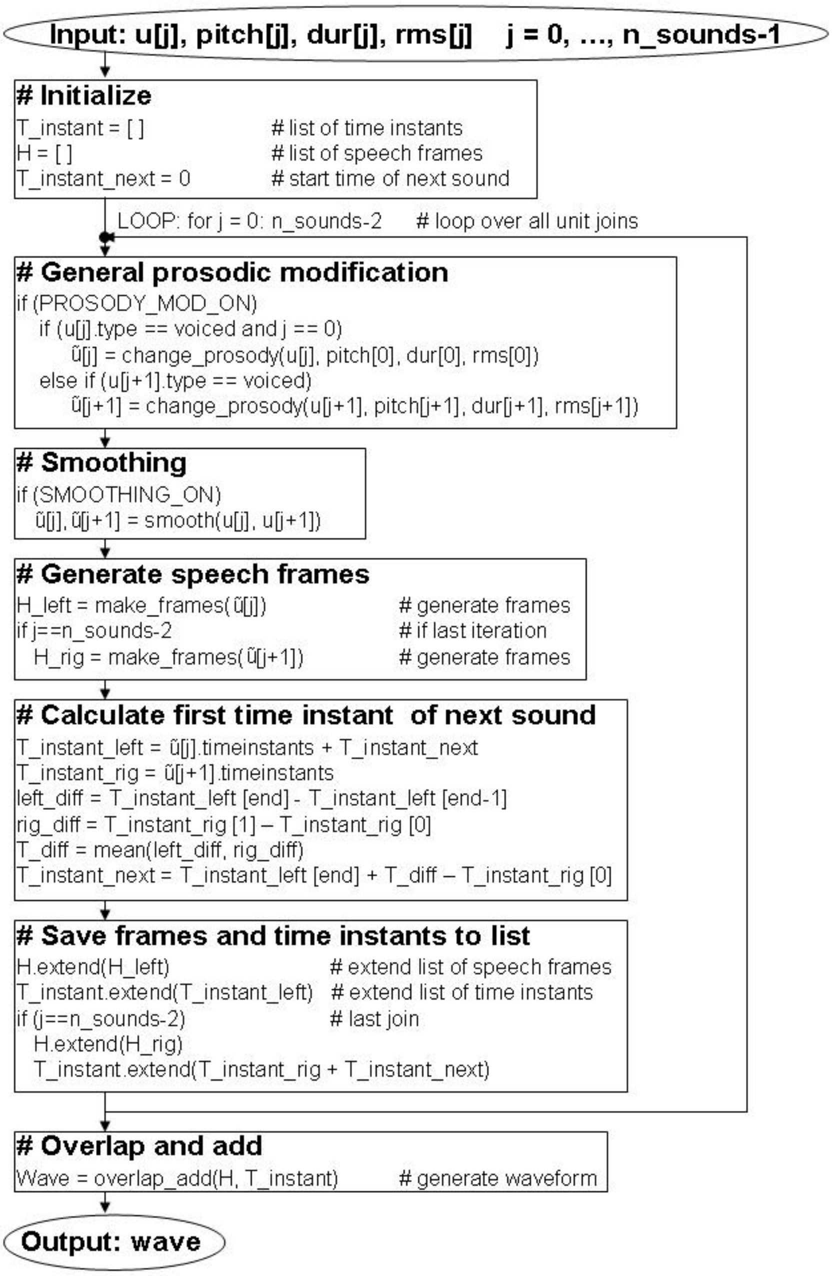
Two modification modules were implemented. One module for handling general prosodic modification, and one module for handling smoothing of parameter trajectories across speech unit boundaries. The general prosodic modification module consisted of pitch, duration and loudness modification, while the smoothing module also included the option of spectral modification by modifying the amplitudes and phases of the harmonic model.

9.1.1 The speech generation algorithm

A flow chart of the speech generation algorithm, including pseudo code, is shown in Figure 9.1. The input to the waveform generation algorithm is a list of speech units, $u_1, \ldots, u_n$, in addition to optional target contours for pitch, duration and root mean square energy. The algorithm processes the speech units as a loop over all speech unit joins, in order to perform smoothing across concatenation points.

The first step after the initialization, see Figure 9.1, is the (optional) general prosodic modification module. This module is used for prosodic modi-

[1] The speech units were defined by the phonetic segmentation of the database.



```
Input: u[j], pitch[j], dur[j], rms[j]    j = 0, ..., n_sounds-1

# Initialize
T_instant = [ ]                # list of time instants
H = [ ]                        # list of speech frames
T_instant_next = 0             # start time of next sound

LOOP: for j = 0: n_sounds-2     # loop over all unit joins

# General prosodic modification
if (PROSODY_MOD_ON)
   if (u[j].type == voiced and j == 0)
       ũ[j] = change_prosody(u[j], pitch[0], dur[0], rms[0])
   else if (u[j+1].type == voiced)
       ũ[j+1] = change_prosody(u[j+1], pitch[j+1], dur[j+1], rms[j+1])

# Smoothing
if (SMOOTHING_ON)
   ũ[j], ũ[j+1] = smooth(u[j], u[j+1])

# Generate speech frames
H_left = make_frames(ũ[j])                # generate frames
if j==n_sounds-2                          # if last iteration
  H_rig = make_frames(ũ[j+1])             # generate frames

# Calculate first time instant  of next sound
T_instant_left = ũ[j].timeinstants + T_instant_next
T_instant_rig = ũ[j+1].timeinstants
left_diff = T_instant_left [end] - T_instant_left [end-1]
rig_diff = T_instant_rig [1] – T_instant_rig [0]
T_diff = mean(left_diff, rig_diff)
T_instant_next = T_instant_left [end] + T_diff – T_instant_rig [0]

# Save frames and time instants to list
H.extend(H_left)                     # extend list of speech frames
T_instant.extend(T_instant_left)     # extend list of time instants
if (j==n_sounds-2)                   # last join
  H.extend(H_rig)
  T_instant.extend(T_instant_rig + T_instant_next)

# Overlap and add
Wave = overlap_add(H, T_instant)          # generate waveform

Output: wave
```

FIGURE 9.1: A flow chart of the waveform generation algorithm.

fication of the speech units based on the input target contours for pitch, duration and energy. This module was applied in the experiments described in Section 9.2 and Section 9.3.

The second step is the (optional) smoothing module, which performs smoothing of parameter trajectories across concatenation points. This module will be described in more detail in Section 9.4.

Next, as a third step in the loop, the waveforms of all the speech frames for the unit to the left of the current unit join are generated and saved to the list of frames. Note that the speech frames on the right side unit at this point still can be modified due to possible smoothing at the next join. The waveform generation of the voiced speech frames, denoted as the function **make_frames()** in Figure 9.1, was in general performed by using the harmonic model equation (Eq. 4.6). However, for the harmonic plus noise model a stochastic part was also synthesized, which is described in Section 9.2.1.3.

The fourth step of the algorithm is to estimate the duration from the last time instant of the (left side) speech unit to the first time instant of the next (right side) speech unit. This is necessary as the speech units can be selected from different speech contexts. An estimate of this duration is only known if two adjacent speech units are selected from the same context. Due to the pitch synchronous approach, this duration was estimated from the pitch contour at each side of the join. In practice, the duration was calculated as the mean of the difference between the two last pitch marks to the left of the join and the difference between the two first pitch marks on the right side of the join. Note that if the smoothing module is applied, (relatively) smooth parameter contours can be obtained with respect to harmonic amplitudes, harmonic phases and pitch across the concatenation points.

Finally, the overlap and add procedure, described in Section 2.3, was applied to generate the waveform of the synthesized sentence. Hanning windows with the same duration as the synthesized speech frames were applied as synthesis weighting windows in the OLA algorithm.

9.1.2 Prosodic modification

The purpose of the general prosodic modification module is to modify the prosody by modifying the pitch, duration, and the root mean square (rms) energy of a speech unit to some predecided target, where the target contours typically would be generated by the synthesizer's front end. The desired target contours were in this implementation defined to have one target value for each speech frame. Specifically, the target contour input was represented as a list of vectors with one vector for each speech unit.

Each vector contained one target value for each voiced speech frame in the corresponding speech unit. Then, a modification to any desired target contour with a resolution dependent on the number of voiced speech frames was possible. Modification of the pitch and duration by a constant scaling factor, e.g. 1.5, or by a fixed amount, e.g. 50 Hz for pitch modification, was also supported. This option was used when a uniform modification of a whole utterance was desired, using the same pitch scaling factor for all speech frames in the utterance.

Loudness modification was implemented as a scaling of the rms energy of each voiced frame with respect to a predecided target rms contour or an rms scale factor. The focus in the experiments has been on the more complex tasks of pitch and duration modification.

For each voiced speech unit, the pitch and duration modification in the general prosody modification module were divided into three steps.

1. Calculation of new time instants (synthesis instants).

2. Amplitude and phase spectrum interpolation

3. Duration modification

The two first steps handle pitch modification, while the third step handles duration modification. The separation of the pitch and duration modification step in the implementation made it easier to evaluate the effects of each step in isolation.

The first step in the general prosody modification module is to calculate new synthesis time instants corresponding to the new target pitch contour. In this step, only pitch modification was considered, temporarily disregarding the need for preserving the duration of the segment. That is, a one-to-one mapping from the original analysis instants to new synthesis instants was initially applied, leading to a modified speech unit with the same number of speech frames as the original speech unit. The synthesis instants were then calculated recursively from the modified pitch contour. The time instant calculation approach used in this thesis is described in more detail in its own subsection below.

The second step is to resample the spectrum at the new harmonic frequencies, in order to obtain modified harmonic amplitudes and phases preserving the original spectrum. The approach using discrete cepstral coefficients, described in Section 5.4.1, was applied for the interpolation of the amplitude spectrum. Several approaches were investigated for interpolation of the phase spectrum. These phase interpolation approaches are described further in the next section.

If the duration of a speech unit after the two first steps deviated from the desired target duration with at least one average pitch cycle, duration modification was performed. The duration was changed by modifying the number of synthesis time instants (or pitch cycles) to the integer number of synthesis time instants that made the new duration as close as possible to the desired target duration. In order to preserve the parameter contours in the time domain, an interpolation of all the parameter trajectories[2] was performed. Finally, new synthesis instants were recalculated from the resampled pitch contour.

9.1.2.1 Time instant calculation

In general, Eq. 5.5 (repeated below) should be fulfilled

$$n_s[i+1] - n_s[i] = \frac{1}{n_s[i+1] - n_s[i]} \int\limits_{n_s[i]}^{n_s[i+1]} T_s(t)dt, \tag{9.1}$$

where n_s are the new synthesis instants, and T_s is the new synthesis pitch period contour. However, when a strictly pitch synchronous preprocessing algorithm has been applied, two relatively independent pitch contour estimates exist. One pitch contour is estimated at each analysis instant by the frame based pitch estimator and another pitch contour estimate is obtained from the distance between pitch marks (analysis instants). The approach for time instant calculation in this thesis was based on preserving the original phase relation between the first harmonic components of adjacent overlapping speech frames at each zero phase instant for a constant pitch scale factor[3]. The pitch for each speech frame was therefore modified according to the target pitch, while the duration between speech frame centers was modified according to the average change in pitch, by averaging the pitch change for each pair of analysis instants.

By this approach, the duration between new synthesis instants was calculated mostly on the basis of the original duration between analysis instants. This approach could possibly be more robust to differences in the two pitch contour estimates. Eq. 9.1 would be fulfilled for a constant pitch contour and approximately fulfilled for a slowly varying pitch contour. However, it should be noted that in tests with time instant calculation

[2] This interpolation would at least include the pitch, the amplitudes of the harmonic components, and the phases of the harmonic components, needed for the harmonic model.

[3] For a time varying pitch scale factor, a speech frame model with a time varying pitch would in general be needed to fulfill this criterion.

based on (either) one of the available pitch contours, pitch modified speech of the same quality was obtained. Hence, the novel approach used in this thesis did subjectively not lead to any significant improvement. A pseudo code for the time instant calculation for a speech unit $u[j]$ is given below, serving as a description of the approach used in the experiments.

```
pitch=u[j].pitch                        # original pitch
T_instant=u[j].timeinstants             # original time instants for unit u[j]
phases=u[j]].phases                     # original matrix of phases

# Calculate the original continuous time zero phase instants (using Eq. 8.1):
T_instant_zp=calc_zp(T_instant,phases[1,:],pitch)

#Calculate the original duration between pitch marks
dur_orig=T_instant_zp[1:end]-T_instant_zp[0:end-1]

# Calculate the average pitch change
delta_f0=f0_new-pitch                   # f0_new represent the target pitch contour
delta_f0_m=(delta_f0[0:end-1]+delta_f0[1:end])/2

#Calculate the new average pitch
f0_new_m=(Fs/dur_orig)+delta_f0_m

# Define first time instant:
T_instant_mod[0]=T_instant[0]
T_instant_mod_zp[0]=T_instant_zp[0]
error_mod[0]=T_instant_mod_zp[0]-T_instant_mod[0]    # Round off error

# Calculate modified synthesis time instants recursively:
for i in range(1,n_frames):
   dur_new[i-1]=dur_orig[i-1]+Fs/f0_new_m[i-1]
   T_instant_mod_zp[i]=T_instant_mod_zp[i-1]+dur_new[i-1]
   T_instant_mod[i]=round(T_instant_mod_zp[i])
   error_mod[i]=T_instant_mod[i]-T_instant_mod_zp[i]
```

T_instant_zp is an array consisting of the original continuous time zero phase instants for a speech unit **u[j]**, and **dur_orig** is an array consisting of the durations between the continuous time zero phase instants. **f0_new** is an array representing the target pitch contour, containing the target pitch for each speech frame center. **T_instant** is an array of the discrete time zero phase instants (analysis instants). **T_instant_mod** is an array of the new discrete time synthesis instants, corresponding to $n_s[i]$ in Eq. 9.1. **T_instant_mod_zp** is an array of the new modified continuous time zero phase instants. Fs is the sampling frequency and **n_frames** denote the number of consecutive voiced speech frames in the speech unit. The round function rounds the continuous zero phase instants to the nearest integer. The round off errors, **error_mod**, were also calculated for possibly adjusting the harmonic phases by linear wave propagation in a final step. Note that only the duration between the synthesis time instants are of interest for the waveform generation algorithm in Figure 9.1. Hence, the first new time instant, **T_instant_mod[0]**, could in principle be set to any integer value.

9.2 Comparing variants of the harmonic model

The estimation and interpolation of the complex spectrum of a speech segment can be performed in several ways, which can lead to different quality and timbre of modified speech. Six variants of modification by a harmonic model are compared in this section. Five of the methods are mainly based on previously reported methods. However, the strictly pitch synchronous preprocessing step leads to some implementational differences. One of the variants can be considered novel. This approach is referred to as method 1d) in the description of the methods in the next section.

It should be noted that the comparison described in this section was a preliminary test for selecting one approach for a comparison to the classic TD-PSOLA approach in a formal listening test. The comparison of variants in this preliminary test was performed by the author by qualitative listening to pitch modified speech for one male voice and one female voice.

9.2.1 Methods

The different variants in this comparison are enumerated below.

1. A full band[4] harmonic model using different approaches for interpolation of the phase spectrum

    a) No interpolation of the harmonic phase spectrum.

    b) A minimum phase model for the harmonic phase spectrum.

    c) A group delay interpolation approach of the harmonic phase spectrum.

[4] The term full band refers to the use of the whole frequency band up to the Nyquist frequency, which in this experiment was the frequency band 0-8000 kHz.

   d) An interpolation based on a minimum phase model and an excitation phase.

2. A full band harmonic model using source filter separation of the speech based on inverse filtering.

3. A harmonic plus noise model using a fixed maximum voiced frequency.

9.2.1.1 Phase interpolation

One set of approaches, enumerated as method 1a-1d above, was a full band harmonic model with the use of various phase spectrum interpolation approaches. A linear phase model [6] (repeated below), described in Section 5.3, was the underlying phase model for the phase interpolation approaches.

$$\phi(\omega_k) = (n_a - n_0)\omega_k + \phi_e(\omega_k) + \phi_s(\omega_k),\ k = 1 \ldots K(\omega_0) \qquad (9.2)$$

where $\phi(\omega_k)$ are the estimated phases of the harmonic components, $\phi_e(\omega_k)$ are the excitation phases, $\phi_s(\omega_k)$ are the system filter phases, $\omega_k = k\omega_0$ are the harmonic frequencies, n_a is the analysis instant, and n_0 is the pitch pulse onset time (defined in Section 5.3).

Due to using a harmonic model, the harmonic number k is used as the argument, i.e. $\omega_k = k\omega_0$. That is, the notation for the phase of the harmonic components used throughout this chapter is

$$\phi_k = \phi(k) = (n_a - n_0)k\omega_0 + \phi_e(k) + \phi_s(k),\ k = 1 \ldots K(\omega_0), \qquad (9.3)$$

No interpolation (1a) The *no interpolation* approach was implemented by either truncating or zero padding the phase spectrum to fit the new modified number of harmonic components for each frame. This approach hence introduces a frequency warping of the phase spectrum. Notice that if this approach were applied to the harmonic amplitude spectrum, it would lead to severely distorted speech. This approach can be considered as a naive baseline method, which would shed some light on whether phase spectrum interpolation is necessary at all.

Minimum phase interpolation (1b) A minimum phase assumption has been tested for phase representation in low bit rate applications in general sinusoidal models [108] in relation to homomorphic deconvolution of speech [6]. In this approach a minimum phase assumption was tested for the phase representation for a harmonic model.

In the *minimum phase interpolation* approach, the phases of the excitation, ϕ_e, are assumed to be zero, and a causal minimum phase model is assumed for the system filter. The new resampled phases can then be obtained from the discrete cepstral coefficients representing the spectral envelope by Eq. 5.25 (repeated below).

$$\phi_s(\omega) = -2 \sum_{m=1}^{p} c_m sin(m\omega), \tag{9.4}$$

where ϕ_s are the system filter phases, c_m are the discrete cepstral coefficients, and p is the number of cepstral coefficients that is applied in the estimation of the spectral envelope. The estimated phases in this approach do hence depend on the estimated spectral envelope and the pitch, and was implemented as a mapping $\underline{\boldsymbol{\phi}} = \Psi_{\phi}(\underline{\mathbf{c}}, f_0)$, similar to the amplitude interpolation. If the phase model in Eq. 9.3 was to be followed strictly, the phases $\underline{\boldsymbol{\phi}}$ should also contain the propagation from the excitation instant to the analysis instant. However, in this approach this term was omitted. This can be interpreted as a linear wave propagation of the speech frames back to the excitation instant for all the voiced speech frames. The phases at the excitation instant, referred to as the excitation phases, of a speech frame can be estimated by linearly propagating the observed (estimated) phases back to the estimated exitation instant, similarly as in Eq. 4.23. In Figure 9.2, the minimum phase approximation for an example speech frame from a vowel segment is compared to the excitation phases, showing a relatively good fit. However, it should be noted that the approximation might not be equally good for all types of sounds, as discussed in 5.5.

Group delay interpolation (1c) The *group delay interpolation approach* is similar to a standard interpolation of the unwrapped phase spectrum, as discussed in Section 5.3. The unwrapping problem consists of removing the linear propagation term in the phase model in Eq. 5.18, and several methods to solve this problem are proposed in the literature [42; 60–62]. The method described here is similar to these methods, but is somewhat simpler due to exploiting the use of the zero phase instant preprocessing approach. Opposed to the approach reported in [42], we do not apply a prior estimate of the unwrapping factor based on the previous speech frame.

As the observed phase spectrum is a function sampled at the harmonics $\omega_k = k \cdot \omega_0, k = 1, \ldots, K$, the (negative) group delay spectrum at the origin, $(n_a = 0)$, can be estimated by differentiating Eq. 9.3 with respect to k.

$$\phi'(k) = -n_0\omega_0 + \phi_e'(k) + \phi_s'(k), \tag{9.5}$$

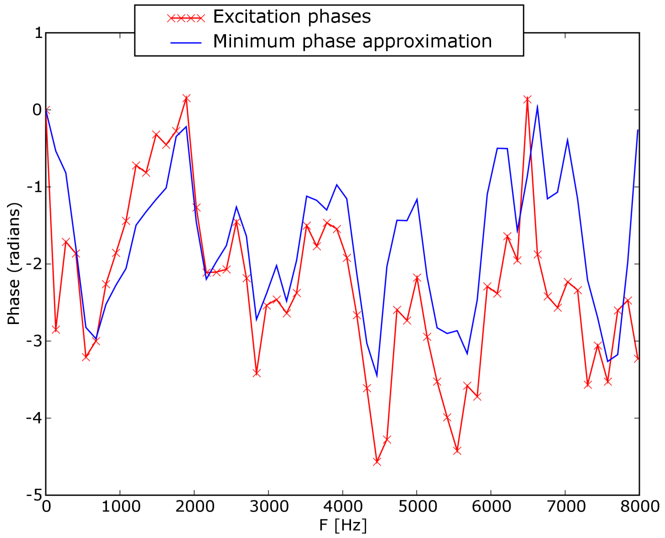


FIGURE 9.2: Excitation phases from an /e:/ sound of a male speaker, compared to the minimum phase approximation calculated from discrete cepstral coefficients. The excitation phases refer to the observed (estimated) phases of a harmonic model propagated back to an estimated excitation instant.

where the prime denotes the derivative with respect to k, approximated by the first order difference.

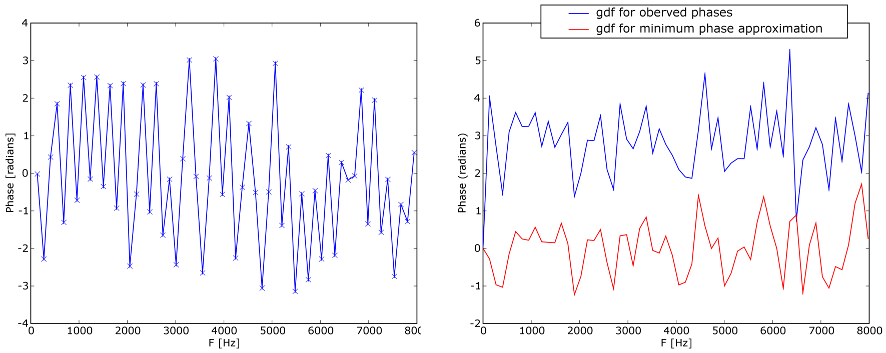
$$\phi'(k) = \phi(k) - \phi(k-1), \quad k > 0, \tag{9.6}$$

The constant phase offset due to the wave propagation can be recognized from Eq. 9.5 as

$$\Delta\phi = n_0 \cdot \omega_0, \tag{9.7}$$

where $\Delta\phi$ is the phase offset and n_0 is the pitch pulse onset time. When zero phase instants are applied as analysis instants, the center of the speech frames will be somewhere in between two glottal excitation instants. The time delay n_0 from the analysis instant to the excitation instant is hence always negative and experienced to be about half a period with some variations due to the different waveform shapes of different sounds. Assuming the zero phase instant is in between two excitation instants, the phase offset $\Delta\phi$ is restricted to be in the interval $[0, 2\pi]$, and will normally be close to π (half a period).

When the observed phases are defined to be represented in the interval $[-\pi, \pi]$, the first order difference is represented in the interval $[-2\pi, 2\pi]$. As $\Delta\phi$ was experienced to be approximately π, an unwrapping of the first order differences was performed by adding 2π to all negative phase differences. An example of a group delay spectrum for an /e:/-sound of a male speaker is shown in Figure 9.3(b), showing a relatively smooth phase spectrum at low frequencies. For comparison, the originally estimated phases for the harmonic components is shown in Figure 9.3(a). A comparison to the phases obtained by the minimum phase approximation, approach (1b), is also shown.



(a) Original phases

(b) group delay phases

FIGURE 9.3: Observed phases for a speech frame from an /e:/ sound of a male speaker, unwrapped phases to the left, and the estimated group delay function for the same frame to the right.

As the group delay phase spectrum is assumed to be relatively smooth, at least in the low frequency band, standard interpolation techniques can be applied to obtain the group delay spectrum at new sine wave frequencies, $\tilde{\phi}'(k)$. In the experiments in this chapter, the approach of using discrete cepstral coefficients was applied, similarly to the interpolation of the amplitude spectrum.

The new interpolated phase spectrum, $\tilde{\phi}(k)$, was then obtained recursively by setting $\tilde{\phi}(1) = \phi(1) \approx 0$ and using

$$\tilde{\phi}(k) = \tilde{\phi}(k-1) + \tilde{\phi}'(k), \qquad k > 1. \tag{9.8}$$

where $\tilde{\phi}(k)$ are the new interpolated phases and $\tilde{\phi}'(k)$ are the first order phase differences interpolated and resampled to the new harmonic components $(k\tilde{\omega}_0)$.

It should be noticed that the phase offset $\Delta\phi$ is included into the phase differences $\phi'(k)$, but as it is a constant, it will not be affected by the interpolation. This implies that a new modified zero phase instant will be located at the same relative position in the pitch cycle in the new modified speech frame as in the original speech frame. Another assumption is that the first order phase difference is within the range $[0, 2\pi]$, which is required for a correct unwrapping of the first order phase differences, see Figure 9.3(b). This is a reasonable assumption if $\Delta\phi$ is relatively close to π and the phase spectrum at the excitation instant has approximately zero mean and is sufficiently slowly varying. The assumption needs to be valid at least in the voiced frequency band. For unvoiced regions of the spectrum the phases are not important as random phases can be applied for the unvoiced components [6].

Minimum phase+excitation (1d) The last phase interpolation approach in this comparison is referred to as *minimum phase+excitation*. This approach was developed as an attempt to use the minimum phase assumption for interpolation (method 1b), but at the same time avoid the sharp excitations in the waveform resulting from the use of a minimum phase assumption only. It can hence be considered a partly novel approach.

In this approach the phase spectrum is separated into an excitation phase spectrum and a minimum phase system filter phase spectrum, assuming a convolutional source filter model. Assuming a minimum phase filter, the system filter phases were obtained from the estimated discrete cepstral coefficients using Eq. 9.4. Then, the excitation phases were calculated as

$$\phi_e(k) = \phi_k - \phi_s(k),\ k = 1, \ldots, K(\omega_0), \tag{9.9}$$

where ϕ_k are the estimated phases, $\phi_s(k)$ are the estimated system filter phases, and $\phi_e(k)$ are the resulting excitation phases sampled at the analysis instant. This approach can hence be interpreted as a source filter model, where the system filter is minimum phase and has an amplitude spectrum a_s (sampled at the harmonics).

During pitch modification, both ϕ_s and ϕ_e should be interpolated and resampled to the new harmonic frequencies in order to preserve the phase spectrum. However, in this approach, the excitation phases were kept unmodified using either a truncation or zero padding of the harmonic components as described in the "no interpolation approach"(1a). The system filter phases were resampled according to the new pitch by using the discrete cepstral coefficients as described in approach 1b. Then, the new harmonic phases at the analysis instant were obtained by inserting the interpolated system filter phases in Eq. 9.9.

A computational advantage with this approach is that the phase unwrapping step is avoided. That is, the phase term due to the propagation from the excitation instant to the analysis instant is preserved, corresponding to assuming that the position of the zero phase instant relative to the pitch cycle is preserved during pitch modification [5].

An example of the excitation phases, or residual phases, is shown in Figure 9.4. It should be noted that for this example the observed (estimated) phases were propagated back to the estimated excitation instant before calculating the residual phases. That is, the propagation term in Eq. 9.3 was estimated and removed.

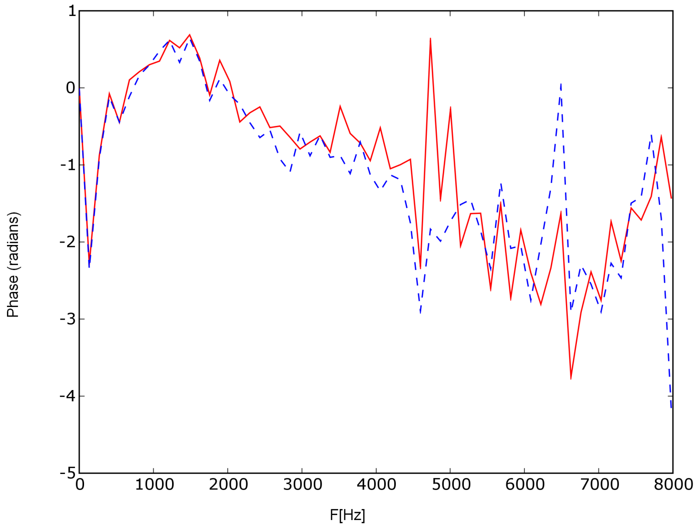


FIGURE 9.4: Residual phases, ϕ_e, for two succeeding speech frames from an /e:/ sound. The red curve is the residuals for the first frame, and the blue(dotted) curve is the residual phases for the second frame.

9.2.1.2 Source filter separation

The approach described in this section, enumerated as method 2, is based on a source filter separation of the harmonic model, as described in Section

[5] This would be a reasonable assumption if the waveform shape is preserved during modification.

5.3. That is, the speech was separated into an excitation, or source, signal, $e[n]$, and a vocal tract filter $v[n]$, where the source signal is an estimate of the glottal flow derivative. This approach has hence similarities to the approach applied in [13].

For the i'th speech frame, the speech model can be expressed as

$$h_i[n] = e_i[n] \circledast v_i[n], \tag{9.10}$$

where the excitation signal $e_i[n]$ is a harmonic representation of the glottal flow derivative estimate for the i'th frame, $v_i[n]$ is the estimated vocal tract filter of the i'th frame, and $\circledast$ denote circular convolution.

In the literature, several approaches are proposed for the estimation of the glottal flow derivative and the vocal tract filter, see Section 2.1.3. We applied a deconvolution approach similar to iterative adaptive inverse filtering (IAIF) [14].

The approach can be summarized as

1. A first order AR-filter was estimated from the original speech frame by linear prediction.

2. The original speech was filtered by the inverse of the filter obtained in the first step, corresponding to a preemphasis of the speech spectrum. This step is intended to remove most of the effect of the glottal source.

3. An AR-filter of order 16 was estimated from the preemphasized speech by linear prediction to obtain an estimate of the vocal tract filter

4. The glottal flow derivative was estimated by inverse filtering of the speech signal, using the estimated vocal tract filter

5. This procedure was repeated in a second iteration, using the obtained glottal flow derivative estimate instead of the original speech signal in step 1, in order to improve the estimate of the glottal source by using a higher order AR-filter.

All estimates were obtained pitch synchronously using the ZP-algorithm. That is, speech frames consisted of two pitch cycles and were centered at the zero phase instants. The autocorrelation method of linear prediction was applied.

The vocal tract filter estimation described above was included in the preprocessing algorithm described in Chapter 8. The filter coefficients and the residual gain of each frame were saved to the speech unit database along with the other parameters extracted in the preprocessing step.

The estimated vocal tract filter could then be expressed as

$$V_i(\omega) = \frac{\sigma_e(i)}{A_i(\omega)}, \tag{9.11}$$

where $V_i(\omega)$ is the frequency response of the vocal tract filter of the i'th frame, $A_i(\omega)$ is the denominator of the estimated all pole model, and $\sigma_e(i)$ is the estimated residual gain. The harmonic amplitudes, $a_s(k)$, and harmonic phases, $\phi_s(k)$, were then obtained by sampling the estimated vocal tract frequency response at the harmonic frequencies.

$$a_s(k) = |V(k\omega_0)| \tag{9.12}$$

$$\phi_s(k) = \angle\left(V(k\omega_0)\right), \tag{9.13}$$

The amplitudes and phases of the excitation were obtained assuming a convolutional model.

$$a_e(k) = a(k)/a_s(k)$$
$$\phi_e(k) = \phi(k) - \phi_s(k)$$

where a_e and ϕ_e are the excitation amplitudes and phases, and $a(k)$ and $\phi(k)$ are the originally estimated amplitudes and phases for the speech frame. In Figure 9.5, an example of the source filter separation of the amplitude spectrum is shown for a speech frame in the middle of an /e:/ sound for the male voice.

In Figure 9.6, an estimate of the glottal flow derivative, obtained by synthesizing only the source part of each frame, is compared to the original speech signal for a segment of the male speaker.

When applying pitch modification, the interpolation and resampling of the amplitudes and phases of the filter and the excitation can be performed separately, allowing different modification of the excitation and the vocal tract filter. Resampling of the vocal tract filter amplitudes and phases was performed by resampling the estimated vocal tract spectrum $V_i(\omega)$ at new harmonic frequencies $k \cdot \tilde{\omega}_0$ by Eq. 9.12 and Eq. 9.13. For the excitation signal, these strategies were tested:

- **2a** Interpolate and resample both the amplitude and the phase spectrum of the source.

- **2b** Keep the amplitudes and phases of the source constant, leading to a compressed or a stretched source signal.

- **2c** Interpolate the source amplitude spectrum, but keep the source phase spectrum unchanged.

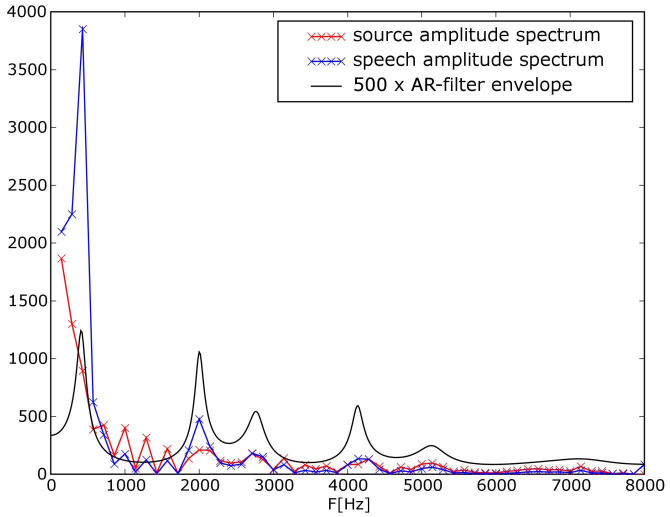


FIGURE 9.5: An example of the source filter separation of the amplitude spectrum for a speech frame from an /e:/ sound of the male speaker. The spectrum of the vocal tract filter was scaled up to the same energy level, as this filter was estimated from the preemphasized (high pass filtered) speech signal.

9.2.1.3 A harmonic plus noise model

A harmonic plus noise model, method 3, was implemented using a procedure similar to the procedure reported in [45]. The harmonic and noise part was synthesized separately and finally added to produce the synthesized speech. The harmonic part was calculated using a fixed maximum voiced frequency, Fm, of 5000 Hz.

The noise part was modeled as white noise modulated by a time varying gain factor and filtered by an all pole vocal tract filter.

$$r_i[n] = \sigma_{np}(i) \cdot \eta[n] * v_i[n], \tag{9.14}$$

where $r_i[n]$ is the stochastic (noise) part of the i'th speech frame, $\eta[n]$ is zero mean white noise with unit gain, $\sigma_{np}(i)$ is an estimated gain factor for the i'th frame, and $v_i[n]$ is the estimated all pole vocal tract filter of the i'th frame. It should be noted that the vocal tract filter was estimated by the same inverse filtering approach as described in Section 9.2.1.2. The stochastic part was hence different from the high frequency part suggested

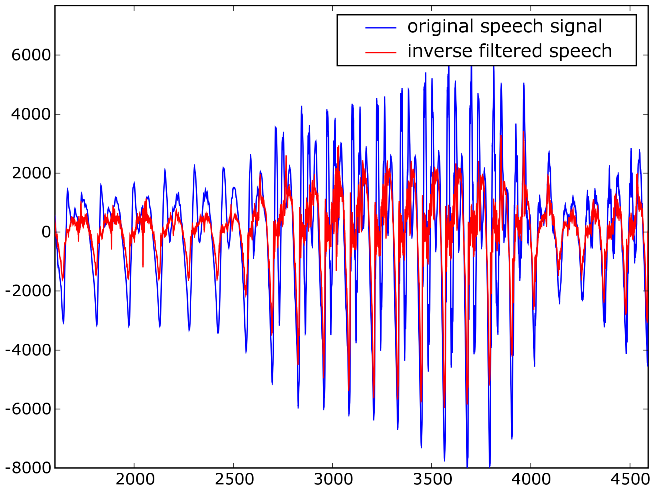


FIGURE 9.6: The estimated glottal flow derivative compared to the original signal for a speech segment from a male speaker.

in [36], as the stochastic part was synthesized as full band noise. No additional high pass filtering of the noise part was performed. The use of a time domain energy envelope was also omitted.

The gain factor σ_{np} for a voiced speech frame was estimated by setting the power of the noise component equal to the power of the high frequency part of the signal consisting of the harmonic components above the maximum voiced frequency (FM).

$$h_{FM}[n] = \sum_{k=K_{FM}(\omega_0)}^{K(\omega_0)} a_k cos(\omega_k n + \phi_k), \tag{9.15}$$

where $k = K_{FM}(\omega_0) \ldots K(\omega_0)$ represent the harmonic components above the maximum voiced frequency up to the Nyquist frequency. The gain factor σ_{np} was then calculated as

$$\sigma_{np}^2 = \frac{P_r}{\sigma_e^2} \tag{9.16}$$

where P_r is the power of the high frequency harmonic signal $h_{FM}[n]$, and σ_e is the estimated gain of the vocal tract filter. An example of the noise part and the harmonic part is shown for a short speech segment in Figure 9.7

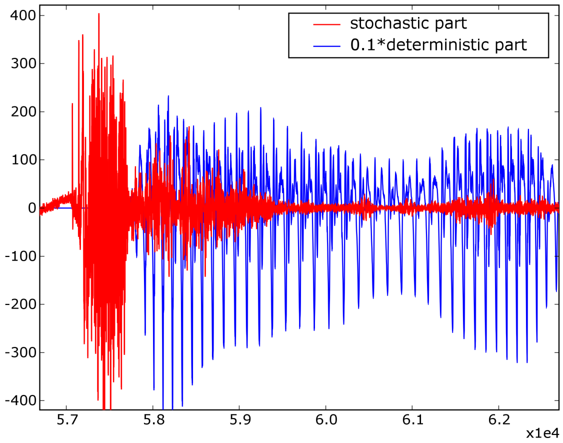


FIGURE 9.7: The noise part compared to the harmonic part for an example speech segment.

9.2.2 Test procedure

One male voice, *t15*, and one female voice, *t16*, described in Section 8.1.2.1, were used in this test. Pitch modification was performed using the general modification module in the waveform generation algorithm, described in the previous section. The duration of the sentences was preserved. 10 sentences were pitch modified using 4 different constant pitch scaling factors for both the male and the female voice. The pitch scaling factors were chosen to be 0.6, 0.8, 1.3 and 1.8. Hence, two factors were used for lowering the pitch contour of the utterance (0.6/0.8), and two factors were used for raising the pitch contour of the utterance (1.3/1.8). In total, 480 $(10(sentences) \times 4(scalingfactors) \times 6(variants) \times 2(voices))$ stimuli (modified sentences) were synthesized. It should be noted that for the source filter separation method (method 2), a separate informal qualitative evaluation of the three methods 2a, 2b and 2c was performed, and only 2a was used in this (also informal) comparison.

Finally, the stimuli were listened to by the author, using a pairwise comparison of variants for each voice and pitch scaling factor. Only a subjective qualitative evaluation of the stimuli was conducted. That is, the type of method was known when listening to a stimulus.

9.2.3 Results and discussion

The quality or timbre of the modified speech for a given method was very similar across different sentences. Hence, a number of 10 sentences were considered enough to describe the speech quality for a given method.

For moderate scaling factors, the quality of pitch modified speech was relatively high. However, for large scaling factors some distortion was introduced. When lowering the pitch, some noise similar to a slightly hoarse voice was introduced. This effect was in particular prominent for the male voice, although it could be perceived for both the male and the female voice. When raising the pitch at the 1.8 pitch scale factor, a weak tonal timbre was perceived for the male voice. This was however not perceived for the female voice.

The distortion was introduced in the pitch modification step. The duration modification step did not introduce distortion for moderate scale factors [6].

The speech quality was very similar for many of the variants, and hence the variants were difficult to rank. Four of the variants were considered equally good. The most promising variants were the variants 1c, 1d, 2a and 3. The variants 1c, 1d and 2a were practically impossible to distinguish, while variant 3 was considered to have a slightly different timbre of approximately the same quality.

Variant 1: The *no interpolation* approach, variant 1a, was only slightly worse than the approaches applying phase interpolation. However, for large pitch scale factors the approach introduced a slightly more noisy timbre of the speech.

The worst variant was the *minimum phase* approach, variant 1b, which led to a very buzzy speech quality. This was probably due to that the minimum phase approach led to sharp excitations in the generated waveform.

Variant 1c and 1d were considered to be of equal quality, and among the four best variants. For variant 1d, it should be noted that an interpolation of the residual (excitation) phases did not improve the speech quality.

Variant 2: For the source filter separation approach, it was found that interpolation of both the phases and the amplitudes of the excitation improved the resulting speech quality. If no phase interpolation was performed, variant 2c, the modified speech was slightly noisy for large pitch scale factors. If no interpolation of the excitation amplitudes was performed, variant 2b, distortion due to a slightly warped amplitude spectrum was perceived. However, it should be noted that only a relatively simple source

[6] Stretching of a segment with an extremely large duration scale factors could however lead to buzzyness, due to using almost identical speech frames successively.

filter deconvolution approach was applied in this thesis. For example, when synthesizing only the source signal, omitting the vocal tract filter, the source signal was perceived more or less as a buzz. However, the signal was still intelligible for a trained listener. Hence a complete separation of source and filter was not obtained. More elaborate methods for deconvolution [11–13], as described in Section 2.1.3, can possibly be applied to improve this method by a more accurate source filter deconvolution.

Variant 3: The harmonic plus noise model was only different from variant 1c, by that the high frequency part was synthesized as filtered white noise. The difference in speech quality was very small. However, the timbre of the speech could be described as slightly more aspirated. For low pitch scale factors, this effect could perhaps be considered to reduce naturalness. However, when raising the pitch of the male voice with a large pitch scaling factor, the weak tonality perceived for the male voice was removed. The stochastic part in this approach was perceptually very close to whispered speech.

A scaling of the stochastic component by a time domain energy envelope for each frame [36] gave subjectively no improvement in the speech quality for this approach. This could possibly be due to using the time varying noise power, corresponding to an amplitude modulation of the noise, which improved the integration with the harmonic part considerably.

9.2.4 Summary

Six variants of a pitch synchronous harmonic model were compared by the author by listening to pitch modified speech. Four of the variants that were tested were found to yield the same quality of pitch modified speech. This was variant 1c, applying phase interpolation on the group delay spectrum, variant 1d, applying a *minimum phase+excitation* phase interpolation approach, variant 2a, using a source filter deconvolution approach based on linear prediction and inverse filtering, and variant 3, a *harmonic plus noise model* with a fixed maximum voiced frequency of 5000 Hz and a full band stochastic component. The phase interpolation approaches gave a small improvement compared to using the original estimated phases directly without interpolation (1a), and gave a large improvement compared to using a minimum phase model for the phases. A possible advantage with the *minimum phase+excitation* approach is that phase unwrapping and the explicit phase interpolation step is avoided. However, instead it depends on a strictly pitch synchronous speech analysis step. The approach 1c was selected for a comparison to TD-PSOLA in a formal listening test, presented in the next section.

9.3 Listening test

The purpose of the listening test presented in this section was to evaluate the quality of pitch modification by the approach described in this thesis. Specifically, the variant using a full band harmonic model and the group delay interpolation method for the interpolation of phases, enumerated 1c in the previous section, was compared to the classic TD-PSOLA [51] approach.

The reason for using the TD-PSOLA approach as the baseline approach was that it is a classical and relatively well performing approach, despite having some weaknesses [51]. In previously reported tests, it has also been somewhat unclear whether the harmonic plus noise model outperforms the TD PSOLA method or not for moderate pitch scaling factors. For example, in [44] the difference in quality between the harmonic plus noise model and the TD-PSOLA method was reported to be insignificant, while in [109] a preference for modification by the harmonic plus noise model was found.

9.3.1 The algorithms used for pitch modification

A summary of the pitch modification algorithms compared in this test are:

- Method 1: **TD-PSOLA**.

  Modification by the TD-PSOLA method were performed as described in Section 5.1. The negative peaks of the speech signal were used as the analysis instants for the TD-PSOLA method, in order to avoid a hoarse speech quality, as discussed in Section 5.1.2. The negative peaks in the waveform were estimated in the preprocessing algorithm, as described in Section 8.1.1.4.

- Method 2: **Harmonic model**. Modification by a full band harmonic model.

  The modification of the voiced speech frames was performed as in the general prosody modification module, described in Section 9.1.2. The group delay phase interpolation method, enumerated as variant 1c in Section 9.2.1.1, was applied.

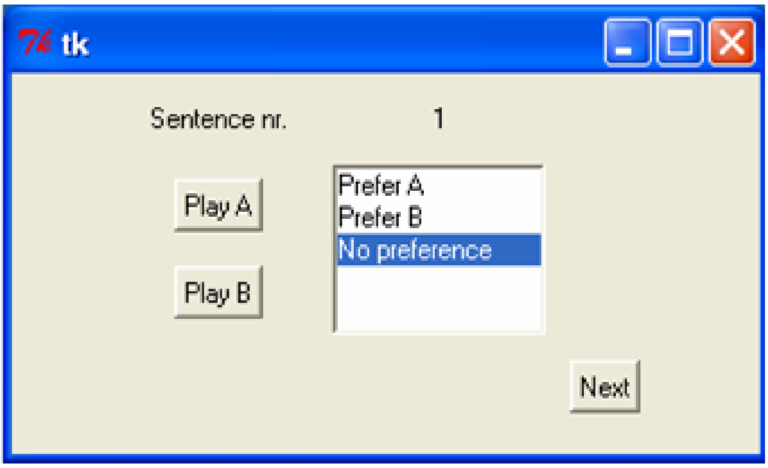
The modified sentences were synthesized by the waveform generation algorithm, see Figure 9.1. It should be noted that only the *general prosodic modification module*, was applied in this test. That is, the *smoothing module* was not applied in this test.

A TD-PSOLA specific modification module was implemented for modification by TD-PSOLA. For TD-PSOLA the synthesis speech frames were selected as the original speech frames being nearest to the new synthesis time instants on the warped time axis. Both methods used the same time instant calculation approach, described in Section 9.1.2.1.

9.3.2 Test procedure

One male voice and one female voice were used in this experiment. The speech databases were the *t15* and the *t16* speech database, described in Section 8.1.2.1. 8 original utterances from each of the speech databases, were pitch modified applying both a full band harmonic model and the TD-PSOLA method.

Four pitch scaling factors were tested for all test utterances: 0.6, 0.8, 1.4, and 1.8. The listening test hence consisted of $2(methods) \times 8(sentences) \times 4(scaling\ factors) \times 2(voices) = 2 \times 64$ stimuli. The test was performed as a paired comparison test. In a graphical user interface, the listeners could play the synthesized test utterance modified by the two modification algorithms. The two synthesized versions of the utterance (the stimuli) were presented as A and B, where it was random whether a stimulus was presented as A or B. The listener could play the stimuli as many times as desired, before choosing among: "prefer A", "prefer B" or "no preference". A figure showing the graphical interface is shown in Figure 9.8.



Sentence nr. 1

Prefer A
Prefer B
No preference

FIGURE 9.8: The graphical interface used in the listening test.

9.3.3 Results

Eight listeners conducted the test. The results are shown in Table 9.1, where the results are shown for each pitch scaling factor and for each voice.

Scale factor	Male Voice			Female Voice		
	Met. 1	**No pref.**	**Met. 2**	**Met. 1**	**No pref.**	**Met. 2**
0.6	58	3	3	2	9	53
0.8	37	20	7	5	18	41
1.3	13	40	11	6	30	28
1.8	14	28	22	11	25	28

TABLE 9.1: The listeners' choices summed over all listeners and sentences at each scaling factor for each of the two voices.

In the analysis of results it is assumed that the choice of sentences and that the set of listeners did not affect the result of the test on average. Two factors that support these assumptions are that the listeners were very consistent in their ratings and that each modification method produced speech of very similar quality for different sentences.

The results in this test was analyzed as a two sided hypothesis test [98], using the null hypothesis that the two methods (method 1 and 2) produce modified speech of equal quality. The experiment is similar to a binomial experiment [98] with 64 trials. However, a "no preference" option was added as a third option in the test. The use of three options adds more information to the analysis than if a regular binomial experiment had been conducted, as the listener is not forced to select one of the methods if the methods are of the same quality.

If all the choices of the "no preference" option are randomly set to the options of preferring method 1 or 2, a binomial experiment is simulated (assuming $p = 0.5$ in the binomial experiment). If the choices of the "no preference option" are evenly distributed to method 1 and 2, the resulting probability distribution function would have the same mean, but a slightly smaller variance than a binomial distribution with $p = 0.5$. Hence, it was decided to divide the "no preference" choices evenly between method 1 and 2, and analyze the experiment as a binomial experiment with $n = 64$ trials. Due to overestimating the variance, the estimated p-values would be slightly too large. The binomial distribution was approximated with a normal distribution, which is a good approximation for large n [98]. Hence, a normal distribution, $N(\mu, \sigma)$, with mean μ and standard deviation σ was applied. A binomial experiment with $p = 0.5$ and $n = 64$ trials then results in a normal distribution approximation $N(np, \sqrt{np(1-p)}) = N(32, 4)$, which was the distribution used to obtain p-values for the listening test results. The estimated p-values are shown in Table 9.2 together with the preference in percentage for the harmonic model, method 2. The p-value is defined as the probability of observing the actual result of the test or a more extreme result when the null hypothesis is true.

Pitch scale factor	Male Voice		Female Voice	
	%	**p-value**	**%**	**p-value**
0.6	7.0	$5.99e-12$	89.8	$1.92e-10$
0.8	26.6	$1.81e-4$	78.1	$6.92e-6$
1.3	48.4	0.80	67.2	0.0059
1.8	56.3	0.31	63.3	0.033

TABLE 9.2: Method preference in percentage preference for modification with the harmonic model (Method 2), averaged over all listeners and sentences. The p-value presented in the table is an estimate of the probability of observing the actual result or a more extreme result, given that the methods are equal.

The p-values in Table 9.2 show that the harmonic model was preferred for the female voice, especially for lowering the pitch. For the male voice, there was no significant preference for any method when raising the pitch, while for lowering the pitch the TD-PSOLA method was preferred.

9.3.4 Discussion

After the test, the listeners were asked whether the task had been easy or not. A general opinion was that the stimuli were similar in many cases, but for some settings the stimuli had quite different speech timbre, though it was not always obvious which of the timbres that were the most natural. For example, when lowering of the pitch for the male voice, the TD-PSOLA method subjectively led to some distortion similar to a "dry" timbre of the speech. When using the harmonic model the speech timbre was subjectively slightly more natural. However, the harmonic model was not preferred in the listening test when lowering the pitch of the male voice. This was probably due to a slightly noisy realization of the synthetic speech for excessively low pitch scaling factors. This "noise" effect could to some extent also be perceived when lowering the pitch for the female voice.

Lowering the pitch of the female voice by the TD-PSOLA method subjectively led to significant distortion, which is reflected by the listening test results, and which explains the large difference between the results for the male and female voice at the 0.6 scaling factor.

For moderate scaling factors, the differences between the two approaches are very small, see Table 9.1. The pitch levels were not randomized, meaning that all stimuli at pitch scaling factor 0.6 were presented before the pitch scaling factor 0.8 and so forth. Listeners could then possibly easier recognize the two methods and choose to be consistent with their earlier choices for the same pitch level and the same speaker. For the pitch scaling factors of 0.8 and 1.3 the listeners were not consistent on which method they preferred, while for the 0.6 and the 1.8 pitch scaling factor the listeners were relatively consistent. This indicates that the difference between the two methods for the 0.8 and 1.3 pitch scaling factor was relatively small.

The different performance of the two modification methods with respect to the type of voice was quite distinct. TD-PSOLA introduces a broadening of formants [51], corresponding to a frequency domain multiplication of the spectrum by the Fourier transform of the analysis window. For a pitch synchronous approach, high-pitched voices lead to shorter duration analysis, and possibly a larger degree of distortion due to broader main lobes in the Fourier transform of the analysis window. It should be noticed, that in the comparison of HNM and TD-PSOLA reported in [109], where HNM was found to be the better approach, only female speakers were tested.

It is difficult to say why lowering the pitch by the harmonic model introduced some distortion. However, lowering the pitch gives an increased number of sine waves in the harmonic model at the same time as the duration of the synthesis speech frames increases. This may give problems with the assumption of stationarity and a relatively constant pitch over the duration of the speech frame. In addition, a one-to-many interpolation of sine wave parameters is needed. Inaccurate estimation and interpolation of phases and amplitudes will also probably lead to more noise in the modified speech. It may also be less correlation between frames at low pitch, which can lead to more distortion at the locations where speech frames overlap the most. The use of a harmonic plus noise model could not solve this problem, as the noise amplification was experienced to be due to the modification of the harmonic part. Possible improvements could be to apply some level of time domain smoothing of parameters in the estimation step [110], or to apply more elaborate speech models. For example, a time varying pitch and/or time varying amplitudes and phases (within the speech frame) could be taken into account [36; 49; 111].

9.3.5 Summary

In this listening test pitch modification by a pitch synchronous harmonic model was compared to pitch modification by the TD-PSOLA for one male and one female voice. For the female voice, the harmonic model was preferred. For the male voice, the methods were rated equal when raising the pitch, while the TD-PSOLA method was preferred when the pitch was lowered by a pitch scale factor of 0.6. The latter result is believed to be due to that a slightly noisy realization speech quality was introduced when the pitch was significantly lowered in the harmonic model approach. A topic for further work would be to find an approach that alleviates this effect.

9.4 Smoothing applied to unit selection synthesis

This section concerns experiments on smoothing of speech parameter trajectories across concatenation points in concatenative synthesis, in order to remove audible discontinuities in the synthesized speech. Two types of smoothing were tested:

- Smoothing of the pitch contour across concatenation points.
- Smoothing of spectral parameters across concatenation points. For example harmonic amplitude trajectories.

For smoothing of the pitch contour, two methods were tested. A baseline local smoothing approach [39], and one new *global smoothing* approach. For spectral smoothing, only the local smoothing approach was applied. The main purpose of the experiment was to see if audible discontinuities could be removed by smoothing parameter trajectories across concatenation points. It should be noted that only a qualitative informal evaluation of the methods, performed by the author, was applied.

9.4.1 Methods

Smoothed speech was generated by using the waveform generation algorithm, see Figure 9.1. The baseline local smoothing, referred to as the smoothing module in the waveform generation algorithm, was implemented as a propagation of the difference in pitch, harmonic amplitudes and phases at the concatenation point to the adjacent speech frames [39], as described in Section 5.7.1. The default was to use 5 voiced speech frames at either side of the join for this smoothing. However, for the smoothing to be correct, it had to be ensured that a voiced frame was not modified twice. For example, for a speech unit with less than 10 voiced frames and with a voiced unit on either side, only half the speech frames (rounded down) in the unit would be available for smoothing at either side of the unit. It was therefore decided that the local smoothing module only could affect speech frames between the concatenation point and the middle of a speech unit. When the number of available frames had been calculated, the amount of modification for each frame could be calculated. The modification of pitch was then performed the same way as described for the general prosodic modification module.

9.4.1.1 Global smoothing of pitch

In order to avoid the problem of having too few speech frames available for proper smoothing, a novel approach was implemented, referred to as *global smoothing*. The global smoothing approach is based on modifying the entire voiced segment. The assumption is that the pitch contour is smooth and relatively slowly varying for every voiced segment, meaning that large jumps in the pitch contour is assumed to only occur after unvoiced or silence segments. The global smoothing principle was only applied to the smoothing of pitch in this experiment.

Global smoothing of the pitch contour was implemented as a module before entering the loop in the waveform generation algorithm in Figure 9.1. The strategy is to estimate a smooth target pitch contour for each voiced segment in the sentence and at the same time minimize the amount of modification needed to obtain this smooth curve from the original pitch curve [7]. After a smooth target pitch curve is estimated, the general prosodic modification module in the waveform generation algorithm can be applied instead of the local smoothing module. The number of frames in the specific speech units within a voiced segment would then not influence the smoothing procedure.

In order to minimize the amount of modification, the smooth target pitch contour was calculated as the least squared error fit to the original pitch curve for each voiced segment. Several parametric functions could be applied to obtain this least squared error fit. In practice, a polynomial function was applied. When using a polynomial fit, it would be important to choose the order of the polynomial such that the natural variation of the pitch contour is preserved, but without using a too high order, which could lead to that discontinuities are fitted as well. In general, the order of the polynomial should depend on the desired number of valleys and

[7] The original pitch curve in this context refers to the pitch curve obtained when concatenating the selected speech units. The original pitch curve could hence contain discontinuities due to that the speech units are selected from different speech contexts.

peaks in the desired pitch contour. However, in this experiment a fifth order polynomial was applied to fit the original pitch curve for all the voiced segments in the sentence.

As a summary, the global smoothing module consisted of a first step identifying all the voiced regions of the sentence. In a second step the parameters of a fifth order polynomial function was estimated for each voiced region. Finally, the target pitch contour was obtained by sampling the estimated smooth pitch contour at the analysis instants. In Figure 9.9, an estimated smooth pitch contour is compared to the original pitch contour for one voiced segment.

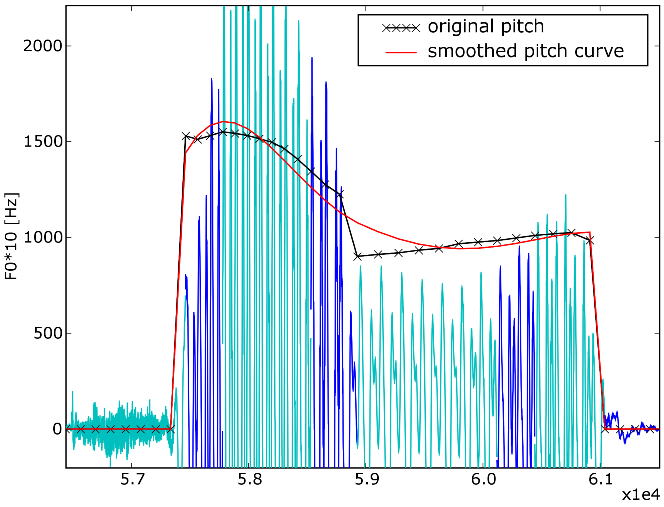


FIGURE 9.9: An example of pitch curve obtained by global smoothing of the pitch for the male voice. The change of color in the waveform corresponds to the unit boundaries.

9.4.2 Test procedure

One male and one female voice was tested for smoothing of acoustic parameter trajectories in unit selection synthesis. Ten sentences were analyzed for each voice. The speech databases *t15* and *t16*, described in Section 8.1.2.1, were used. These databases are relatively small. Hence, synthesized speech from these small databases contained several severe audible discontinuities for each sentence.

Synthetic speech was obtained by using two different unit selection synthesis systems. The Festival unit selection synthesis system [112], serving as a reference (baseline) system, and the python/mysql synthesis system, described in Section 9.1. In order to make the selection of speech units identical in both synthesis systems, it was chosen to use the Festival synthesis system to select the speech units. Information about the selected speech units, such as phoneme identity, phoneme context and segmentation boundaries were used to retrieve exactly the same speech units in the python/mysql synthesis system.

Smoothing were applied to the following parameter trajectories:

- The pitch contour, represented by the estimated pitch at each analysis instant.
- The amplitudes of the harmonic components, where the harmonic amplitudes were represented as a vector of amplitudes at each analysis instant.
- The phases of the harmonic components, where the phases were represented as a vector of phases at each analysis instant.
- The discrete cepstral coefficients, representing the spectral envelope at each analysis instant. Only the first 20 coefficients, representing the slowly varying part of the spectral envelope, were used in the smoothing. The amplitudes of the harmonic components were retrieved from the modified cepstral coefficients before synthesis.

The smoothed speech was finally evaluated by qualitative listening, performed by the author.

9.4.3 Results and discussion

The synthesized speech in the Festival system and the synthesized speech in the python/mysql system were not distinguishable when no modification was applied. The unmodified synthesized speech was however of a poor quality, with many discontinuities due to the use of relatively small speech databases. The task for the smoothing module was hence considered very difficult.

9.4.3.1 Pitch smoothing

Subjectively, the global smoothing approach removed audible pitch discontinuities more successfully than the local smoothing approach. However, the synthesized sentences did also contain many other types of discontinuities than pitch discontinuities, e.g. spectral mismatches. These spectral discontinuities outnumbered the number of pitch discontinuities. Hence, very careful listening was required to evaluate the effect of the pitch smoothing. The data was therefore considered to be too "noisy" for performing a formal listening test on pitch smoothing.

In Figure 9.9, an example of a segment from one of the synthesized sentences is shown. In this example the unit to the left of the pitch discontinuity (in blue color) consists of only three frames, and hence it has only one frame available for local smoothing. Hence, the local smoothing approach will obviously fail in this case. When using the global smoothing approach (the red contour), it is seen that a smooth pitch contour can be obtained.

The global smoothing approach depends on a small modification of the pitch of all the speech frames in the voiced segment. Subjectively, a small modification of the pitch could be performed without any audible distortion. However, the success of the global smoothing approach would obviously rely on the amount of modification needed for obtaining smooth pitch contours.

In the global smoothing approach described in this section, a fifth order polynomial was used to fit the original pitch curve. However, for very long voiced segments a fifth order polynomial could be insufficient to model all the natural variations of the pitch curve, hence a method to decide the optimal polynomial order could possibly improve the method. If for example the front end of the synthesis system generates a symbolic representation of the pitch, e.g. TOBI [113], a more appropriate polynomial order could possibly be estimated from the number of valleys and peaks in the predicted symbolic pitch curve.

9.4.3.2 Spectral smoothing

The smoothing of harmonic amplitudes and harmonic phases across the unit boundaries, described in Section 5.7.1, did subjectively not make much difference to the speech quality for the synthesized speech sentences in this test. Similarly as for the smoothing of pitch, the problem of too short speech units for proper smoothing could occur. For example, if two neighboring units are spectrally very different, it could be impossible to generate smooth speech without modifying the spectral content of one of the units entirely. Local smoothing of the cepstral coefficients did subjectively not lead to any improvements, except maybe for effectively smoothing the energy of the speech units across unit boundaries.

A theoretical weakness with using the amplitudes of the harmonic components for smoothing the spectrum is that the frequencies of the harmonic components vary with time. Hence, a proper smoothing of the estimated magnitude spectrum would only be performed when the pitch is constant. The pitch dependency could be avoided by a transformation to the discrete cepstral coefficients. However, a problem with using cepstral coefficients is that a relatively small change in the spectrum could result in a large change in the cepstral coefficients. Hence, it would probably be better to use other parameterizations of the spectrum. Applying smoothing and modification of parameters more directly related to the perception of speech would hence be a natural topic for further research. One approach could be to smooth line spectral frequencies [66; 68] across concatenation points, for example by using the source filter model variant of the harmonic model. Another approach could be to model the frequency spectrum at each frame by using classical speech parameters as formants, antiformants, spectral tilt, in addition to source signal parameters [16].

9.4.4 Summary

In this experiment, smoothing of parameter trajectories across unit concatenation points was evaluated. Subjectively, it was more difficult to remove audible discontinuities due to spectral mismatches than audible discontinuities due to discontinuities in the pitch contour.

Local smoothing of harmonic amplitude trajectories across concatenation points did not successfully remove audible discontinuities. More elaborate modeling of the spectral parameters, and possible global smoothing schemes based on models trained on real speech would be natural topics for further work. Audible distortion due to discontinuities in the pitch contour could however be removed successfully. A method, which was named *global smoothing* of pitch, did subjectively improve the smoothing of the pitch contour. The advantage of a *global smoothing* approach is that it avoids the problem of having only a limited number of frames available for smoothing.

Chapter 10

Concluding summary

The main focus in the work presented in this thesis, has been to avoid audible discontinuities at concatenation points in unit selection synthesis. The work has been performed on three different but related fields.

- Join cost function design, in order to improve the selection of units in unit selection synthesis.
- Pitch synchronous speech processing and pitch estimation.
- Modification of speech, in order to remove audible discontinuities at concatenation points or to obtain a target prosody.

10.1 Join cost function design

The work on join cost function design has been focused on the task of finding features and corresponding distance measures that correlate well with audible discontinuities, and on how to combine these distance measures into a join cost function.

A listening test on detecting audible discontinuities in vowel joins for two Norwegian vowels was conducted. The listeners' ratings were used as a reference to compare several objective spectral distance measures with respect to the correlation to human perception. However, due to that a small speech database was applied for generating the stimuli for this test, some of the stimuli contained a relatively high difference in the pitch at the concatenation point. The result was that the pitch difference was found as the best detector of audible discontinuities in this particular test. This shows that this type of experiments reflects the frequencies of the different types of discontinuities present in the test stimuli. Hence, the result could be very sensitive to the specific synthesis system and the specific test design. This could be one explanation for the large variability of the results in previously reported studies. For a better comparison of results across different studies, shared databases with carefully designed test stimuli should be established for this research field.

Of the spectral distance measures, the Euclidian distance on Mel frequency cepstral coefficients and the Euclidian distance on cepstral coefficients obtained from AR-spectra were found as the most promising. However, the detection of discontinuities due to spectral mismatches was relatively low in this listening test. Hence, more experiments should be conducted to draw more certain conclusions on the performance of these distance measures. A distance measure based on cross-correlation was also tested in this comparison, but it was not found very promising. It was also found that this measure was very correlated to pitch differences.

The work on join cost function design in this thesis has been based on defining the join cost as the probability for a listener detecting an audible discontinuity. This definition leads to the interpretation of the join cost function as the discriminant function of a two-class pattern recognition problem. Based on this definition, a probabilistic join cost model based on statistically independent features is proposed, leading to a nonlinear join cost function. The possible theoretical improvement with this join cost function is a better weighting of each feature when several features are used in the join cost function, and also that it could be easier to do a fair integration with the target cost function in a unit selection system. The possible improvement by using a probabilistic approach would in general depend on the robustness of the difficult estimation of the probability of an audible discontinuity. The data obtained by the listening test presented in this thesis was however not well suited for a test of this approach. Hence, more experiments should be conducted to see if the approach could give improvement in practice.

Two different strategies for the design of join cost functions are proposed in this thesis. One strategy is to use the target cost function of the speech synthesizer to generate the stimuli of the perceptual experiment, in order to obtain as relevant test stimuli as possible. Then a join cost function can be built by using a least squares fit to the listeners' ratings in the listening test. The problem of selecting features for the join cost function can be solved by using stepwise linear regression or stepwise logistic regression on a set of candidate features. A problem with this approach is that it may have poor generalization properties.

A second more general strategy is to first choose a set of relevant features, i.e. by using the strategy described above, and then perform perceptual experiments on each feature in isolation to obtain an estimate of the probability for an audible discontinuity given a specific feature. Then the probabilistic join cost function proposed in this thesis could be applied.

10.2 Pitch synchronous speech processing and pitch estimation

The work on speech-processing has been on developing a robust frame based pitch synchronous speech processing algorithm. The algorithm was designed for the estimation of parameters for a harmonic model, but can be applied to any kind of pitch synchronous speech processing. Special for this algorithm is the use of zero phase instants as analysis instants (speech frame centers) in voiced speech regions, where the zero phase instants are defined as the time instants of zero phase for the first harmonic component of a harmonic model. In addition, an approach for self-validation of the pitch estimate was applied. By the self-validation approach, the algorithm can avoid trusting the pitch estimate of the previous frame when passing through irregular or turbulent regions of the speech signal, and at the same time take advantage of the assumption of a relatively smooth pitch contour in regular regions of the speech signal. This pitch validation approach showed to increase the robustness of the pitch adaptive processing algorithm by reducing the occurrence of gross pitch estimation errors. In addition, this approach showed to detect regions of irregular speech, e.g. creaky voice. This approach could be used to automatically label speech units containing irregular regions in a unit selection speech synthesis system, possibly aided by manual inspection of the irregular regions.

An important part of the speech processing algorithm is robust pitch estimation. Three pitch estimators were tested for the use in this algorithm, an SNR-based pitch estimator, an HNR-based pitch estimator, and an ESPRIT-based pitch estimator. All of the three pitch estimators were found to be robust enough for producing transparent resynthesized speech, as well as producing relatively high quality pitch modified speech by the use of a harmonic model if gross pitch errors were avoided.

The HNR-based estimator was modified to resolve the problem of pitch halving errors by applying the average HNR. This pitch estimator was considered the most appropriate pitch estimator for the pitch synchronous speech processing algorithm due that it does not depend on any coarse initial pitch estimate, and also that it is very simple and relatively effective. This HNR-based estimator can be interpreted as an approximation to the SNR-based estimator, which is based on maximizing the signal-to-noise ratio for the reconstruction of the speech signal by a harmonic model.

In a comparison of the pitch estimators to a reference based on manually checked pitch marks, the estimator was found to be practically unbiased to the reference, and it was also found to produce smoother pitch contours than the reference pitch estimate. The pitch estimator was as expected found to be highly correlated to the computationally more complex SNR-based estimator. It was also verified that the use of the average HNR instead of the HNR gave insignificant bias.

When testing the HNR-based estimator on synthetic speech-like signals, a small weakness with the HNR-based estimator was found for extremely harmonic (periodic) speech frames, which could lead to a pitch halving error. This effect was due to that the estimator is formed as a ratio, which makes the HNR estimator very sensitive to the noise energy estimate. This weakness can however be avoided by using the harmonic-to-signal ratio or by using a variant of the HNR-based estimator based on a voiced/unvoiced decision of each harmonic component. In the comparison of pitch estimators, both these variants of the estimator were found to have the same performance as the HNR-based estimator.

10.3 Modification of speech

Different variants of a harmonic model have been tested for modification of speech in unit selection synthesis. The work on modification has been divided into two categories:

- General prosodic modification, consisting of modification of pitch and duration to given target contours.
- Smoothing of parameter trajectories across concatenation points, consisting of smoothing of both the pitch contour and the amplitude- and phase-trajectories of the harmonic components.

Special for the algorithms presented in this thesis are the use of a strictly pitch synchronous approach, using the zero phase instants, estimated in the preprocessing step, as analysis instants.

10.3.1 Prosodic modification

Several variants of a harmonic model were compared for the task of pitch and duration modification, before one of the most promising variants were compared to the classical TD-PSOLA method in a listening test for one male and one female voice.

The variants of modification by a harmonic model that were compared in this thesis were a harmonic+noise model, a source-filter deconvolution of the harmonic model parameters, and a full band harmonic model.

Several methods were tested for the interpolation of the phase spectrum of the harmonic model, including one novel phase spectrum interpolation approach, referred to as the *excitation+minimum phase* interpolation approach. Four of the variants of the harmonic model were subjectively evaluated to be equally good for the task of pitch modification. These variants were the harmonic+noise model, a harmonic model using a source-filter deconvolution approach, the full band harmonic model using a *group delay phase interpolation* approach, and a full band harmonic model using the *excitation+minimum phase* phase interpolation approach. A possible advantage with the *excitation+minimum phase* phase interpolation approach is that the phase unwrapping problem is avoided. However, instead this approach relies on a strictly pitch synchronous preprocessing step. In general, interpolation of the phase spectrum was found to slightly improve the quality of the pitch-modified speech.

The variant using a full band harmonic model and the *group delay phase interpolation* approach was selected for a comparison to the classical TD-PSOLA approach in an objective listening test. For the female voice, the modification by a harmonic model was preferred, especially for lowering the pitch. For the male voice, the methods were rated as equally good for raising the pitch, while the TD-PSOLA method was preferred for lowering the pitch. This was probably due to that the harmonic model introduced a noise effect similar to a slightly hoarse voice when excessively lowering the pitch.

10.3.2 Smoothing of parameter trajectories across concatenation points

Time domain smoothing of speech parameter trajectories across concatenation points, by the use of a harmonic speech model, was tested for one male and one female voice. Subjectively, it was more difficult to smooth audible discontinuities due to spectral mismatches than to smooth audible discontinuities due to pitch mismatches.

For the smoothing of the pitch contour, it was experienced that a local smoothing across concatenation points could be insufficient due to many short duration speech units, having too few speech frames available for proper smoothing. This was in particular a problem for the low-pitched male voice. To avoid this problem a new *global smoothing* approach was applied. This approach is based on fitting a smooth pitch contour to each voiced segment in the sentence. This approach was experienced to improve the smoothing of pitch discontinuities. However, it should be noted that the number of pitch discontinuities were relatively low relative to the number of discontinuities due to spectral mismatches in this experiment. Hence, more experiments should be conducted to verify this hypothesis.

10.4 Further work

The design of a join cost function that takes possible pitch modification of speech units into account could be a topic for further work. Synthetic design of stimuli for perceptual experiments on audible discontinuities is also an interesting topic. That is, modification of natural speech could be used to generate test stimuli for perceptual experiments on speech unit joins. Then the test stimuli in the perceptual experiment could be carefully designed, however, with the caveat that the modification methods could affect the result.

The pitch synchronous speech-processing algorithm could be applied to other related fields of speech technology. For example, to the task of automatic phonetic segmentation. Another topic would be to make a computationally effective version of the pitch synchronous speech processing algorithm for possibly applying it to real time applications like speech recognition. A nice property for real time applications is that the algorithm performs robust pitch synchronous analysis in a simple left to right manner. Another topic for further research could be to improve the relative simple approach for pitch validation, and to support the detection of irregular speech regions by more elaborate methods, e.g. based on spectral estimation.

A natural topic for further research would be to try to improve the pitch and duration modification. In particular, the effect of a slightly hoarse voice when excessively lowering the pitch should be avoided. Possibly, a more elaborate speech model could be needed to avoid this effect. Other topics could be to constrain the estimated pitch curve in the analysis stage to be strictly smooth with respect to the distance between estimated pitch epochs, and to look at effects of time domain smoothing of the spectrum in the estimation stage. It would also be interesting to apply more elaborate methods for source filter deconvolution. An improved source filter deconvolution approach could possibly lead to a modification approach that is more consistent with speech production, and could perhaps provide better quality of modified speech and more freedom in what types of modification that can be performed.

For the smoothing of spectral mismatches, it would be important to find spectral representations that better reflects the human perception of spectral mismatches. It would also be interesting to apply the global smoothing concept to the smoothing of spectral content. For example, if two speech units are acoustically very different, it could possibly be necessary to modify the spectral content of one of the units entirely. Models of speech unit transitions could be trained from natural speech, in order to help the synthesis system to generate natural speech parameter trajectories, similar to HMM-based synthesis.

Bibliography

[1] J. C. Wells et. al., "Standard Computer-Compatible Transcription", SAM-UCL-037, in ESPRIT PROJECT 2589 (SAM): Final Report;Year Three; 1.3.91-28.2.92, 1992

[2] K. Tokuda, H. Zen, A. W. Black, "An HMM-based speech synthesis system applied to English", *Proc. of 2002 IEEE SSW*, Sept. 2002.

[3] S. Nakajima and H. Hamada, "Automatic generation of synthesis units based on context oriented clustering", *Proc. ICASSP 1988*, pp. 659-662, (New York, USA), 1988

[4] J. P. H. van Santen. "Combinatorial issues in text-to-speech synthesis", *Proc. Eurospeech 1997*, vol 5., pp. 2511-2514, (Rhodes, Greece), 1997.

[5] X. Huang, A. Acero, H.-W. Hon, *Spoken Language Processing, A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2001.

[6] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and practice*, New Jersey: Prentice Hall, 2002.

[7] L. R. Rabiner, R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[8] C. W. Therrien, *Discrete Random Signals And Statistical Signal Processing* , Charles W. Therrien & Prentice Hall, 1992.

[9] J. Makhoul, "Linear prediction: A tutorial review", *Proceedings of the IEEE*, vol. 63, pp. 561-580, April 1975.

[10] B. S. Atal, S. L. Hanauer "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave", *Journal of the Acoustic Society of America*, vol. 50,pp. 637-655, August 1971.

[11] M. D. Plumpe, T. F. Quatieri, and D. A. Reynolds, "Modeling of the Glottal Flow Deriative Waveform with Application to Speaker Identification", *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 5,pp. 569-586, Sept. 1999.

[12] Q. Fu, P. Murphy, "Robust Glottal Source Estimation Based on Joint Source-Filter Model Optimization", *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 2,pp. 492-501, March. 2006.

[13] D. Vincent, O. Rosec, T. Chonavel, "Estimation of LF glottal source parameters based on an ARX model", *Interspeech*, pp. 333-336, 2005.

[14] P. Alku, "Glottal wave analysis with pitch syncronous iterative adaptive inverse filtering." *Speech Communication*, vol.11, pp.109-118, 1992.

[15] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels," *Journal of the Acoustic Society of America*, vol. 49., pp. 583-590, 1971.

[16] D. Klatt and L. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers", *J. Acoust. Soc. Am.*, 87(2):820-857, 1990.

[17] G. Fant, J. Liljencrants, and Q. Lin, "A four-parameter model of glottal flow", *Speech Transmission Lab. Quart. Prog. Status Rep.*, vol.4, pp.1-13, 1985.

[18] B. Bozkurt et. al., "Zeros of Z-transform (ZZT) representation with application to source-filter separation in speech.", *IEEE Signal Processing Letters* 12(4), pp. 344-347, 2005.

[19] R. E. Crochiere, "A Weighted Overlap-Add method of Short-Time Fourier Analysis/Synthesis", *IEEE Trans. Acoustics, Speech and Signal Processing,* vol ASSP-28, no. 1, pp. 99-102, Feb. 1980.

[20] M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time Fourier analysis", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. ASSP-28, pp. 55-69, Feb 1980.

[21] A. V. Oppenheim, R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[22] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", *Proc. ICASSP-96*, pp. 373-376, 1996.

[23] N. Campbell, A. Black, "Prosody and the Selection of Source Units for Concatenative Synthesis", in *Progress in speech synthesis*, eds J. van Santen, R Sproat, J Olive and J. Hirschberg, pp 279-282, Springer Verlag, 1997.

[24] A. J. Viterbi "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Trans. on Information Theory*, Vol. IT-13, pp. 260-269, 1967.

[25] H. Hermansky, "Perceptual linear predictive (PLP) analysis for speech", *Journal of the Acoustic Society of America*, pp. 1738-1752, 1990.

[26] H. Hermansky, N. Morgan, "RASTA processing of speech", *IEEE Transactions on Speech and Audio Processing,* special issue on Robust Speech

[27] A. H. Gray and J. D. Markel "Distance Measures for Speech Processing", *IEEE Trans. on ASSP*, pp. 380-391, Oct 1976.

[28] E. Klabbers and R. Veldhuis, "On the computation of Kullback Leibler measure for spectral distances", *IEEE Trans. SAP*, pp. 100-103, Jan. 2003.

[29] N. Nukaga, R. Kamoshida and K. Nagamatsu "Unit selection using pitch synchronous cross correlastion for japanese concatenative speech synthesis", *Proc. 5th ISCA Workshop on Speech synthesis*, Pittsburgh, 2004.

[30] J. L. Flanagan and R. M. Golden, "Phase vocoder", Bell System Technical Journal, vol. 45, no. 9, pp. 1493-1509, Nov. 1966

[31] L. B. Almeida and F. M. Silva, "Variable-Frequency Synthesis: An Improved Harmonic Coding Scheme", *Proc. IEEE Int. Conf. Acoustic, Speech and Signal Processing,* San Diego, CA, pp. 27.5.1-27.5.4,1984.

[32] P. Hedelin, "A Tone-Oriented Voice-Excited Vocoder", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing,* Atlanta, GA, pp. 205-208, 1981.

[33] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Transactions on Acoustics, Speech, Signal Processing,* vol. ASSP-34, pp. 744-754, Aug 1986.

[34] E. B. George, and M. J. T. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model", *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 5., pp. 389-406, Sep. 1997.

[35] X. Serra, J. O. Smith. "Spectral modeling synthesis: A Sound Analysis/Transformation/Synthesis System Based on a Deterministic Plus Stochastic Decompostition", *Computer Music Journal*, Vol.14, no. 4, pp. 12-24, Winter 1990.

[36] Y. Stylianou, "Modeling Speech Based on Harmonic Plus Noise Models", *Nonlinear Speech Modelling*, LNAI 3445, pp.244-260, Springer-Verlag, Berlin Heidelberg, 2005.

[37] D. W. Griffin, J. S. Lim "Multiband Excitation Vocoder", *IEEE Trans. Acoustics, Speech, and Signal Processing,* ASSP-36(2):236-243, Feb 1998.

[38] T. F. Quatieri and R. J. McAulay, "Speech transformations based on a sinusoidal representation", *IEEE Transactions Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no 6, pp. 1449-1464, Dec. 1986.

[39] Y. Stylianou, J. Laroche, E. Moulines, "High Quality Speech Modification based on a Harmonic + Noise Model" , *Proc. Eurospeech*, pp. 451-454, 1995

[40] B. Yegnanarayana, C d'Alessandro, V. Darsinos, "An iterative algorithm for decomposition of speech signals into periodic and aperiodic components", *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 1-11, 1998.

[41] Y. Stylianou, "Decomposition of Speech Signals into a Periodic and Non-periodic Part based on Sinusoidal Models", *Proc. ICSLP-96*, vol 2., pp. 1213-1216, Philadelphia

[42] Y. Stylianou, "Applying the Harmonic Plus Noise Model in Concatenative Speech Synthesis", *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No 1, pp. 21-29, Jan 2001.

[43] J. Laroche, Y. Stylianou, and E. Moulines, "HNM: A simple efficient harmonic + noise model for speech," *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, NY, October 1993.

[44] G. Bailly, "Accurate Estimation of Sinusoidal Parameters in a Harmonic + Noise model for speech synthesis", *Proc. EUROSPEECH '99*, pp. 1051-1054, Budapest, Hungary,September 1997.

[45] D. Vandromme, "Harmonic Plus Noise Model for Concatenative Speech Synthesis", *Diploma thesis, IDIAP, 2005*, IDIAP-RR 05-37, 2005.

[46] W. Mattheyses, W. Verhels, P. Verhoeve, "Robust pitch marking for prosodic modification of speech using TD-PSOLA", *Proc. of SPS-DARTS 2006*, pp. 43-46, 2006.

[47] C-Y. Lin, J-S. R. Jang, "A two-phase pitch marking method for TD-PSOLA synthesis", *Proc. Interspeech-2004*, pp. 1189-1192, 2004.

[48] T. V. Ananthapadmanabha, B. Yegnanarayana, "Epoch extraction of voiced speech", *IEEE Trans. ASSP-27,* Vol. 6, pp. 562-570, 1975.

[49] P. Zubrycki. A. Petrovsky, "Analysis/Synthesis Speech Model Based on the Pitch-Tracking Periodic-Aperiodic Decomposition", In *Information Processing and Security System,* K. Saeed, J. Pejas, Eds., Springer Verlag, 2005.

[50] Y. Stylianou, "Removing Phase Mismatches in Concatenative Speech Synthesis", *Proc. 3. ISCA Speech Synthesis Workshop*, pp. 267-272, Nov. 1998.

[51] E. Moulines and F. Charpentier , "Pitch-synchronous waveform processing techniques for text to speech synthesis using diphones", *Speech Communications*, Vol 9. , pp. 453-467, Dec. 1990.

[52] K. S. Rao, B. Yegnanarayana, "Prosodic manipulation using instants of significant excitation", *Proc. Int. conf. on Multimedian and Expo*,vol 1. , pp. 389-398, july 2003.

[53] D. Paul, "The Spectral Envelope Estimation Vocoder", *In IEEE Workshop on App. of sig. proc. to Audio and Acostics*, Mohonk, Oct. 1997.

[54] T. Eriksson, H. G. Kang, Y. Stylianou, "Quantization of the spectral envelope for sinusoidal coders", *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 37-40, 1998.

[55] O. Cappe, J. Laroche, E. Moulines, "Regularized Estimation of Cepstrum Envelope from Discrete Frequency Points", *IEEE Trans. Acoustics,Speech, and Signal Processing*, Vol. ASSP-39, no. 4, pp. 786-794. Aug. 1981.

[56] R. J. McAulay, T. F. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech model", *Proc. IEEE ICASSP-84*, Vol. 2, pp. 27.6.1-27.6.4, 1984.

[57] R. J. McAulay, T. F. Quatieri, "Sine-wave phase coding at low data rates", *Proc. IEEE ICASSP-91*, Vol. 1, pp. 577-580, 1991.

[58] R. J. McAulay, T. F. Quatieri, "Shape Invariant Time-Scale and Pitch Modification of Speech", *IEEE Trans. Speech and Audio Processing*, Vol. 40, no. 3, pp. 497-510 March. 1992.

[59] T. F. Quatieri, R. B. Dunn, T. E. Hanna, "Time-scale modification with temporal envelope invariance", *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, (New York), Oct. 1993.

[60] R. J. McAulay, T. F. Quatieri, "Phase modeling and its application to sinusoidal transform coding", *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 1713-1715, April 1986.

[61] R. Smits, B. Yegnanarayana, "Determinitation of Instants of Significant Excitation in Speech Using Group Delay Function", *IEEE Trans. Speech and Audio Processing*,Vol. 3, no. 5,pp- 235-333, Sept. 1995.

[62] H. Kawahara, Y. Atake, and P. Zolfaghari, "Accurate vocal event detection method based on a fixed-point analysis of mapping from time to weighted average group delay", *Proc. ICSLP 2000*, pp. 664-667, 2000.

[63] C. d'Alessandro, B. Doval, "Experiments in Voice Quality Modification of Natural Speech Signals: The Spectral Approach", *proc. SSW3-1998*, 277-282, 1998.

[64] A. Conkie, S. Isard, "Optimal Coupling of Diphones", *Progress in speech synthesis*, van Santen et al (eds.), Springer Verlag, 1997.

[65] E. Klabbers, R. Veldhuis, "On the reduction of concatenation artifacts in diphone synthesis". *Proc. ICSLP-98*, Sydney, Australia, Vol. 5, pp. 1983-1986, 1998.

[66] J. Wouters, M. Macon, "Control of spectral dynamics in concatenative speech synthesis", *IEEE Transactions on Speech and Audio Processing*, 9(1): 30-38, 2001.

[67] D. T. Chapell, J. H. L. Hansen, "A comparison of spectral smoothing methods for segment concatenation based speech synthesis", *Speech Communication*, Vol 36 (No 3/4), pp. 343-374, March 2002.

[68] M. Plumpe, A. Acero, H. Hon, X. Huang "HMM-based smoothing for concatentative speech synthesis", *Proc. ICSLP-98*, Sydney, Australia, Vol. 6, pp. 2751-2754.

[69] W. Hess, "Pitch and voicing determination", in *Advances in Speech Signal Processing* (S. Furui and M. M. Sondhi, eds.), New York: Marcel Dekker, 1991.

[70] D. Gerhard, "Pitch extraction and fundamental frequency: History and current techiques," *technical report*, Dept. of Computer Science, University of Regina, 2003.

[71] L. R. Rabiner et. al. , "A Comparative Performance Study on Several Pitch Detection Algorithms", *IEEE Trans. on ASSP.*, vol. 24, no. 5, pp.369-377, 1976.

[72] D. Talkin, "A robust algotihm for pitch trackin(RAPT)", *Speech coding and synthesis* (Elsvier, ed.), pp. 495-518, 1995.

[73] A. M. Noll, "Cepstrum pitch determination", *Acoustical Society of America*, vol. 41, pp. 293-309, Feb. 1967

[74] L. Rabiner, "On the use of autocorrelation analysis for pitch detection", *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-25, pp. 24-33, Feb. 1977

[75] R. J. McAulay and T. F. Quatieri, "Pitch estimation and voicing detection based on a sinusoidal speech model", *Proc. IEEE ICASSP-90*, vol. 1, pp. 249-252, 1990.

[76] W. Zhang, H. Kim, W. H. Holmes, "Investigation of the spectral envelope estimation vocoder and improved pitch estimation based on

the sinusoidal speech model", *In Proc. IEEE Int. Conf. on Information, Communications and Signal Processing*, Singapore, pp.513-516, Sept. 1997.

[77] R. Ahn and W. H. Holmes, "An improved harmonic-plus-noise decomposition method and its application to pitch determination", *IEEE Workshop on Speech Coding for Telecommunications Proceedings*, Pocono Manor, USA, pp. 41-42, Sep. 1997.

[78] C. d'Alessandro, B. Yegnanarayana and V. Darsinos, "Decomposition of speech signals into deterministic and stochastic components", *Proc. IEEE ICASSP-95*, pp.760-763, 1995.

[79] R. Ahn, W. H. Holmes, "Harmonic-plus-noise decomposition and its application in voiced/unvoiced classification", *Proc. IEEE TENCON 97*, vol.2 pp. 587-90, Brisbane, Australia 1997.

[80] R. Roy, T. Kailath, "ESPRIT-Estimation of Signal Parameters Via Rotational Invariance Techniques", *IEEE Transactions on Acoustics. Speech, and Signal Processing*, Vol. 37, no. 7, pp. 984-995, 1989.

[81] M. Wax, T. Kailath, "Detection of signals by information theoretic criteria", *IEEE Trans. ASSP*, Vol. 39, pp. 387-392, 1985.

[82] N. Malik, W. H. Holmes, "Pitch estimation and a measure of voicing from pseudo spectra.", *Proc. IEEE ICASSP-00*, pp. 1463-1466 Vol. 3

[83] L. Y. Ngan, H. C. So, P. C. Ching and S.W. Lee, "Joint Time Delay and Pitch Estimation for Speaker Localization", *Proc. Circuit and Systems*, ISCAS-03, Vol. 3, pp.722-725, May 2003.

[84] I. Bjørkan, S. Farner, T. Svendsen, "Comparing Spectral Distance Measures for Join Cost Optimization in Concatenative Speech Synthesis", *in Proc. Eurospeech-05*, Lisboa, 2005.

[85] J. E. Natvig, P. O. Heggtveit, *PROSDATA - A speech database for study of Norwegian prosody v2.0*, Telenor R&D, N 20/2000, Kjeller 2000.

[86] MATLAB version 7.0, The MathWorks, `http://www.mathworks.com`

[87] MYSQL version 4.0.17, `http://www.mysql.com`

[88] E. Klabbers and R. Veldhuis, "Reducing audible spectral discontinuities", *IEEE Trans. SAP*, pp. 39-51, Jan. 2001.

[89] J. Vepa, S. King, and P. Taylor, "Objective distance measures for spectral discontinuities in concatenative speech synthesis", in *ISCLP*, 2002.

[90] J. Vepa, S. King, and P. Taylor, "New objective distance measures for spectral discontinuities in concatenative speech synthesis", *Proc. IEEE 2002 Workshop on Speech Synthesis*, (Santa Monica,USA) September 2002.

[91] Y. Stylianou and A. K. Syrdal, "Perceptual and objective detection of discontinuities in concatenative speech synthesis", *Proc. ICASSP-01*, pp.837-840, Salt Lake City, 2001.

[92] J. Wouters, M. Macon, "Perceptual evaluation of distance measures for concatenative speech synthesis", *Proc. ICASSP-01*, 2001.

[93] R. E. Dononvan, "A new distance measure for costing spectral discontinuities in concatenative speech synthesis", *The 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001.

[94] Y. Pantazis, Y. Stylianou, E. Klabbers, "Discontinuity detection in concatenated speech synthesis based on nonlinear speech analysis", *Proc. INTERSPEECH-2005*, pp. 2817-2820, Lisbon, 2005.

[95] J. Vepa, S. King, "Join cost for unit selection synthesis", in *Text to Speech Synthis*, S. Naranyan, A. Alwan, Eds., Prentice Hall, 2004.

[96] R. Duda, R. E. Hart, D. G. Stork, *Pattern Classification*, John Wiley & Sons, 2001.

[97] N.-S. Kim, S.-S. Park "Discriminative Training for Concatenative Speech Synthesis", *IEEE Signal Processing Letters*, Vol 11 (No 1), pp. 40-43, Jan. 2004

[98] R. E. Wapole, R. H. Myers, S. L. Myers, *Probability and statistics.*, Prentice Hall, 1998.

[99] G. Casella, R. L. Berger, *Statistical Inference*, 2nd ed., Duxbury Press, 2001.

[100] B. Atal, L. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition", *IEEE Trans. on ASSP*, Vol. 24, No. 3, pp. 201-212, 1976.

[101] J. E. Natvig, P.O. Heggtveit, "En sanntids demonstrator for norsk tekst til talesyntese", *Telenor R&I, rapport TF R 15/93*, 1993.

[102] T. Svendesen et. al. "Fonema - Tools for Realistic Speech Synthesis in Norwegian" *Proc. Norsig 2005*, Stavanger, 2005.

[103] D. Meen, T. Svendsen, J. E. Natvig "Improving Phone Label Alignment by Utilizing Voicing Information" *Proc. SPECOM 2005*, pp. 683-686, Patras, Greece, 2005.

[104] I. Amdal "FonDat1: A Speech Synthesis Corpus for Norwegian" *Proc. LREC 2006*, Genoa, Italy, 2006.

[105] M. I. Jordan, "Why the logistic function? a tutorial discussion on probabilities and neural networks". Techincal Report 9503, MIT Computational Cognitive Science.

[106] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ration of a sampled sound", *IFA Proceedings,* 17, pp. 97-110, 1993.

[107] Python version 2.4, `http://www.python.org`

[108] T. F. Quatieri, "Minimum- and mixed-phase speech analysis/synthesis by adaptive homomorphic filtering", *IEEE Trans. Acoustics, Speech and Signal Processing,* Vol. ASSp-27, no. 4,pp. 328-335, Aug. 1979.

[109] Y. Stylianou, "Concatenative speech synthesis using a harmonic plus noise model", *Third ESCA Speech Synthesis Workshop*, pp. 261-266, Nov. 1998.

[110] H. Kawahara et. al., "Restructuring speech representations using a pitch-adaptive time frequency smoothing and a instantaneous frequency- based F0 extraction: Possible role of a repetitive structure in sound", *Speech Communication*, Vol. 27, pp. 187-207, 1999.

[111] Q. Li, L. Atlas, "Time-variant least squares harmonic modeling", *Proc. ICASSP-03*, Vol.2, pp. 41-44, 2003.

[112] A. Black, K. Lenzo, "Building Voices in the Festival Speech Synthesis System," unpublished document, Carnegie Mellon University, `http://www.festvox.org/bsv` .

[113] K. Silverman et. al., "ToBI: a standard for labeling English prosody", *In Proceedings of the Second International Conference on Spoken Language Processing*, 2: 867 - 70. Banff, Canada 1992.