Singing Voice Vibrato: Measurement and Modification

Diploma Thesis

Peter Sciri

Supervisors: DI Dr. Alois Sontacchi
o.Univ.-Prof. Mag. DI Dr. Robert Höldrich

Graz, June 14, 2011

kunst uni graz

TU Graz
Graz University of Technology

institut für elektronische musik und akustik

The intention of this thesis is to present a concept to describe inherent attributes of the human singing voice in a qualitative manner. Many aspects of the singing voice are unique to it such as its wide tonal range, the rich repertoire of timbral colorizations or the possibilities regarding dynamics and phrasing while singing. The entity of all those attributes yields an instrument with very special properties. As one of those properties, vibrato can be singled out being one of the most artistic and virtuoso features of the singing voice. Therefore, this issue will be discussed prominently in the course of this thesis.

Objective of this work will be to present a framework of measures to describe the singing voice on a physical basis considering the related anatomy and mechanisms involved in voice production. Based on state-of-the-art digital modelling techniques for the processes in the vocal apparatus the respective model parameters are determined from microphone recordings. The involved measurement procedures include pitch, amplitude and formant tracking. Special attention is paid to the estimation of the glottal source signal by the development of a new constrained closed phase glottal inverse filtering algorithm. The presented novelties in this work include the combination of multiple stages to determine the instant of glottal closure from microphone signals as well as cycle prototyping to ensure coherence of analysis.

Exploiting the knowledge derived from a small scale empirical study on vocal vibrato, an algorithm for actively influencing vibrato in singing voice signals will be presented. A set of predefined vibrato patterns has been recorded by a classically educated baritone singer under studio conditions. These recordings are used to investigate the relationship between voice model parameters and the occurrence of vibrato.

As one of the possible practical scenarios, vibrato cancellation will be discussed in detail. An assessment of conceptual and computational possibilities and limitations will be given. Furthermore, the usage of a linear time-variant Lattice filter as well as the implementation of an asymmetric pitch-synchronous overlap-and-add technique to perform synthesis will be presented. As one result, the complete reduction of a semitone vibrato to a static pitch is shown.

Das Ziel dieser Diplomarbeit besteht darin, Konzepte zur qualitativen Beschreibung inhärenter Attribute der menschlichen Singstimme vorzustellen. Diese besitzt viele einzigartige Eigenschaften, die sie aus der Gruppe der Musikinstrumente hervorheben. Ihr großer Tonumfang, das umfangreiche Repertoire an klangfarblichen Variationsmöglichkeiten sowie ihre Vielseitigkeit in Phrasierung und Dynamik machen sie zu einem ganz besonderen Mittel künstlerischen Ausdrucks. Als eine dieser Eigenschaften kann das Vibrato hervorgehoben werden, welches zu den künstlerisch und technisch virtuosesten Aspekten der Singstimme zählt und daher in dieser Arbeit vordergründig behandelt wird.

Ziel ist die Ableitung eines Satzes von Maßen zur Beschreibung der Singstimme auf physikalischer Basis anhand der beteiligten anatomischen Vorgänge im Stimmapparat. Basierend auf digitalen state-of-the-art Modellierungsverfahren zur Beschreibung dieser Abläufe werden die entsprechenden Modellparameter aus Mikrofonaufnahmen ermittelt. Im Speziellen werden Tonhöhe, Amplitude sowie Lagen und Amplituden der Formanten als Funktion der Zeit bestimmt. Hohe Aufmerksamkeit erhält außerdem die Berechnung des *glottal source* Signals durch die Entwicklung eines neuartigen *constrained closed phase inverse filtering* Algorithmus. Die in dieser Arbeit vorgestellten Neuerung umfassen unter Anderem eine angepasste Methode zur exakten Bestimmung des Schließzeitpunktes der Glottis aus Mikrofonaufnahmen wie auch *cycle prototyping*, welches die Kohärenz der Analyseumgebung ermöglicht.

Auf die Erkenntnisse aus einer kleinen empirischen Studie über Gesangsvibrato zurückgreifend wird ein Ansatz zur aktiven Beeinflussung von Vibrato in Stimmsignalen vorgestellt. Dafür wurde eine Satz aus vordefinierten Vibrato Abfolgen von einem Bariton Sänger klassischer Ausbildung unter Studiobedingungen aufgenommen. Diese Daten werden verwendet um die Zusammenhänge zwischen den Sprachmodellparametern und dem Vorhandensein von Vibrato zu untersuchen und die resultierenden Erkenntnisse im Anschluss diskutiert.

Als eine der praktischen Anwendungsszenarien wird *vibrato cancellation* im Detail vorgestellt wobei eine Gegenüberstellung von konzeptionellen und rechnerischen Möglichkeiten und Einschränkungen durchgeführt wird. Zusätzlich wird spezielles Augenmerk auf die verwendeten Synthesetechniken gelegt. Die Implementierungen eines linearen, zeitvarianten (LTV) Lattice filters sowie des Konzepts von *asymmetric pitch-synchronous overlap-and-add* werden vorgestellt. Als anschauliches Ergebnis kann die Reduktion eines Halbton Vibratos zu einer statischen Tonhöhe gezeigt werden.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

date

signature

Contents

Introduction **8**

1 Fundamentals Of Voice Production **13**

1.1 Introduction 14
1.2 Components Of The Voice Production Mechanism 14
1.2.1 The Larynx 15
1.2.2 The Vocal Tract 17
1.3 Source-Filter Model Of Speech 17
1.3.1 Properties Of The Glottal Source 19
1.3.2 Properties Of The Vocal Tract 21
1.4 Lossless Tube Model 23
1.5 Speech vs. Singing 24
1.6 What is Vibrato? 24

2 Determination Of Voice Model Parameters **26**

2.1 Introduction 27
2.2 F0 Tracking 28
2.2.1 Overview 28
2.2.2 The YIN Algorithm 29
2.2.3 Results 35
2.2.4 Removing Octave Errors 36
2.3 Amplitude Tracking 41
2.4 Formant Tracking 42
2.5 Time-Aligned Representation 45
2.6 The Cycle-Aligned Representation 46
2.7 Estimation Of The Open Quotient 48
2.8 Empirical Data 49
2.8.1 Test Specifications 49
2.8.2 Recording Environment And Equipment 50

2.8.3 Testing Individuals 51
2.8.4 Results and Interpretation 51

3 Time-Spectral Processing Techniques 55
3.1 Introduction 56
3.2 Overlap-And-Add 58
3.2.1 Possibilities of OLA 59
3.3 Glottal Inverse Filtering 60
3.3.1 Motivation 61
3.4 Linear Predictive Coding 63
3.4.1 Basic Principles 63
3.4.2 Solving the LPC Equations 66
3.4.3 The Lattice Formulation: An Alternative Representation 69
3.5 Synthesis Techniques 72
3.5.1 Pitch-Synchronous Overlap-And-Add (PSOLA) 72
3.5.2 Asymmetric Pitch-Synchronous Overlap-And-Add (A-PSOLA) . . . 73
3.5.3 Implementation of an LTV Lattice filter 76
3.5.4 Effects of Parameter Interpolation 80
3.5.5 Performance Issues 81

4 Constrained Closed Phase Covariance Glottal Inverse Filtering 86
4.1 Introduction 87
4.2 Identifying the Instants of Significant Excitation in Voiced Speech 88
4.2.1 Basic Idea 90
4.2.2 Energy Weighting 92
4.2.3 Robustness Against Additive Noise 93
4.2.4 Robustness Against Echo and Reverberation 93
4.2.5 Effects of the Window Length 95
4.2.6 Removing False Positives from the Candidate Set 96
4.2.7 Results 102
4.3 Cycle Prototyping 103
4.3.1 The Problem 104
4.3.2 The Idea 106
4.3.3 Limitations 107
4.3.4 Results 108
4.4 Constraining Root Positions 108

5 Modification Of Vibrato **111**

5.1 Introduction 112

5.1.1 Generating Artificial Vibrato 112

5.1.2 Cancelling Existing Vibrato 114

5.2 General Algorithmic Thoughts 117

5.3 Synthesis Benchmarks 118

5.3.1 Example: Vibrato Cancellation 119

5.3.2 Results 120

5.3.3 Performance Issues 121

6 Conclusion **125**

6.1 Summary 126

6.2 Outlook 127

A Empirical Results **128**

Introduction

The human voice is an outstanding instrument of communication. Besides its evolutionarily derived function of exchanging necessary information with other individuals it also transports knowledge, emotion and arts.

Singing can be regarded as one of the most basic forms of musical expression in human history. Obviously, there is no need for an instrument - all you need to do is raise your voice. Nevertheless, in classical, western music but also in many other cultures and musical backgrounds, singing is equally seen as one of the highest forms of art.

In this context the vocal vibrato can be singled out as one of the most prominent and virtuoso characteristics of well educated singing. Its mastery requires many years of voice training but still remains a complex task.

In the course of this work we want to present an overview on the mechanisms and procedures involved in singing voice production as well as an algorithm that is capable of measuring and actively influencing the occurrence of vibrato in audio recordings on basis of the physical model of voice production.

In chapter 1 we will describe the *physical components* that contribute to voice production and their influence on the singing voice. Subsequently, we will discuss *two mathematical modelling techniques* that are commonly used to simulate the voice production apparatus in digital speech signal processing. Concluding this chapter, we will evaluate if these models are equally valid for singing voice and which of the model parameters are useful to describe vocal vibrato.

Chapter 2 focusses on a set of *measurement techniques* that we have used to determine this set of parameters form audio recordings in our algorithm. Here we pay special attention to the accurate estimation of *pitch, amplitude* and the *formant data.* Additionally, we will introduce two methods to *intuitively illustrate the processes in the vocal apparatus.* Subsequently, we will evaluate the presented techniques on a *small scale empirical study* we have carried out in the course of this work (the audio data is available online at `http://sciri.at/files/DA/audio`). The resulting data is also used to illustrate the observations and conclusions we have drawn on the evolution of the model parameters during vibrato.

After having presented a way to determine the model parameters from real world audio signals we will discuss a set of methods to respectively modify these parameters in chapter 3. We present the concept of *formant conserving pitch shifting* and its extension to *glottal inverse filtering.* This will allow for determining the actual glottal excitation signal as it occurs in the vocal apparatus and applying any changes on the very physical basis. We will discuss the determination of the vocal tract parameters by *linear prediction* and subsequently two *time-spectral synthesis techniques* that are founded on this model.

The main challenge in inverse filtering is the exact estimation of the influence of the vocal tract resonator. We will therefore describe the multilayered *constrained close phase covariance glottal inverse filtering* algorithm in chapter 4. It allows for highly accurate estimates of the vocal tract influence and thus enables us to remove it from the speech signal. We will extensively discuss the crucial aspect of *determining the instant of glottal closure* and introduce set of post processing techniques to refine the prior calculated data.

Chapter 5 combines all of the earlier described techniques to establish a basis of *modifying vibrato* on a physically and musically meaningful basis. We present a set of abstract control parameters and discuss the two possible scenarios of *vibrato synthesis* and *vibrato cancellation.* Latter will be presented more intensively in a practical example and the resulting problems and possibilities of implementation will be discussed at the end of this chapter.

Acknowledgements

I have spent a lot of time thinking and elaborating a list of people I want to thank at this point - people who have been in close relation to the development of this work but also people who have gone with me for much longer periods of my life. My family and friends, whose love, affection and patience have made me the person I am.

Quickly, the problem arose in which *order* these people are to be mentioned? Isn't there always some inherent ranking in putting subjects or objects in an order? So what could an engineer do to circumvent this delicate matter?
... *randomize!*

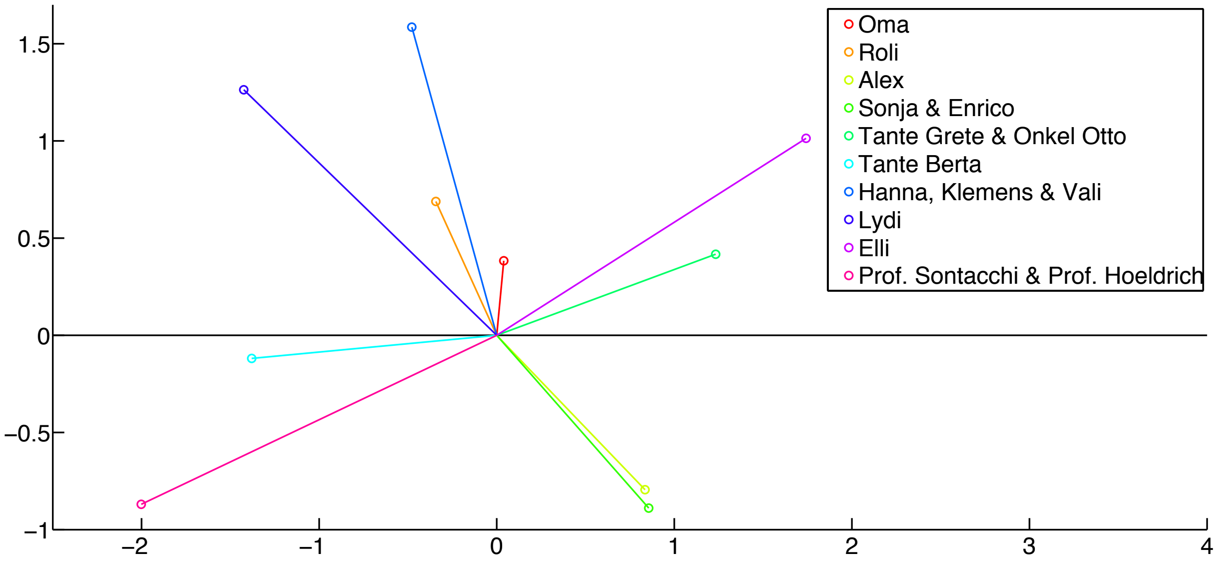


Figure 0.1: *Acknowledgements sorting randomizer*; the order of the list is derived from the distance of the respective acknowledgement atom to the origin

The "*acknowledgements sorting randomizer*" uses two gaussian random value generators to define the x and y coordinates of a set of data points on a 2d plane. The resulting distance to the origin determines the order of acknowledgement as depicted in figure 0.1.

Danke, liebe Oma, dass du immer an mich geglaubt hast! Du fehlst mir sehr!

Ein großer danke geht an Roland, der für mich die Aufnahmen der kleinen Studie durchgeführt hat und an Georg Smola für seine gesangliche Spende.

Ebenso großer Dank gebührt Alex, der mich in den letzten Jahren sowohl am Fels als auch am Computer durch seinen eigenen Ehrgeiz über meine Grenzen hinausgehen hat lassen.

Ein unglaublich großes Dankeschön richte ich an meine wunderbaren Eltern Sonja und Enrico. Ohne eure liebevolle Unterstützung und Fürsorge wäre ich nie an diesem Punkt angelangt.

Danke auch an meine Tanten Berta und Grete und an Onkel Otto. Eine Familie wie euch kann man sich nur wünschen.

Ebenso danke ich meiner "Ersatzfamilie" in unserer gemeinsamen Wohnung, bestehend aus Hanna, Klemens und Vali. Dementsprechend natürlich auch Christina und Christof.

Ein liebevolles Danke schicke ich meiner Schwester Lydi. Wenn du lachst geht mir das Herz auf!

Das Wort Dank allein kann wohl kaum beschreiben, was meiner Elli hier gebührt. Du bist der Wind in meinem Segel, der schützende Hafen in der Nacht, der starke Ast, der mich trägt. Meine Gefährtin. Ich liebe dich!

Einen großen Dank richte ich auch an meine Betreuer Prof. Alois Sontacchi und Prof. Robert Höldrich. Ohne Ihre aufmerksame Hilfe, Unterstützung und Geduld wäre diese Arbeit nicht möglich gewesen.

Nicht zu allerletzt danke ich auch meinem treuen MacBook, denn die eigentliche Arbeit hat ja es für mich getan.

1 Fundamentals Of Voice Production

1.1 Introduction

Semantic speech is one of the most apparent differences between humans and other socially living animals. The capability of interaction beyond the basis of non-verbal, hence mimic or gestural communication is often regarded as one of the “boosters” of human evolution. The cerebral development of the human species is therefore strongly related to the evolution of speech.

In this chapter we want to present the physical components that take part in speech production, as well as two examples of models that have been derived from the physiological properties of the voice production mechanism and have evolved to the two most commonly used models of speech.

Later we will briefly compare the processes of speaking and singing in order to establish the mathematical basis of the subsequent analysis and synthesis procedures.

At the end of this chapter we want to describe the actual process of vibrato generation and by which parameters the occurrence of vibrato can be classified.

1.2 Components Of The Voice Production Mechanism

The human apparatus of voice production is commonly subdivided into two major parts:

1. the *larynx*, containing the vocal folds that form the *glottis* as well as a vast number of muscles and ligaments to control positioning of the laryngal cartilages and thus the tension of the vocal folds and

2. the *vocal tract* that consists of *pharynx*, *mouth* and the *nasal cavities*.

An additional factor that intensively contributes to voice production are of course the *lungs*. They are responsible for the flow of air that passes first the glottis and afterwards the vocal tract. A schematic illustration of the entire vocal system is provided in figure 1.1.

The production of a sound can be generalized as sequence of processes:

1. air enters the lungs due to the under-pressure provoked by the release of muscular force during an intake

2. the air leaves the lungs trough the *trachea*

3. it passes the *larynx* and the comprising vocal folds that cause *phonation*

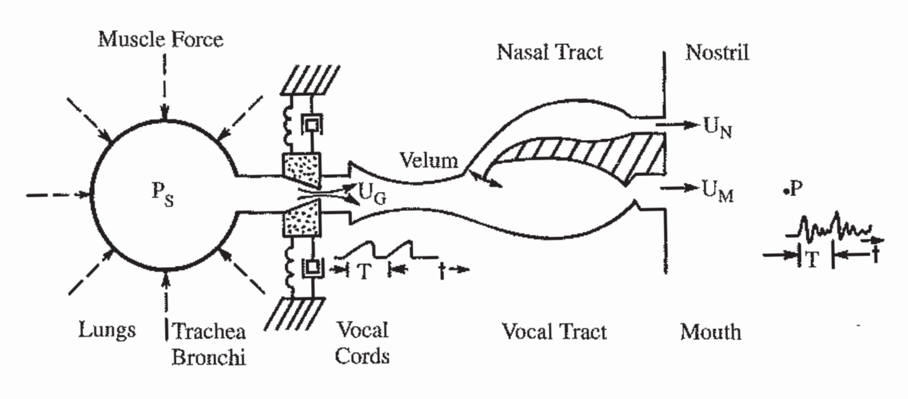


Figure 1.1: Schematized diagram of the vocal apparatus ([1] p.88)

4. the resulting airflow is *frequency shaped* by the following pharynx, mouth and nasal cavities.

1.2.1 The Larynx

The larynx is source of *phonation* and is located as the topmost part of the trachea which transports the air that is pressed outwards by the lungs.

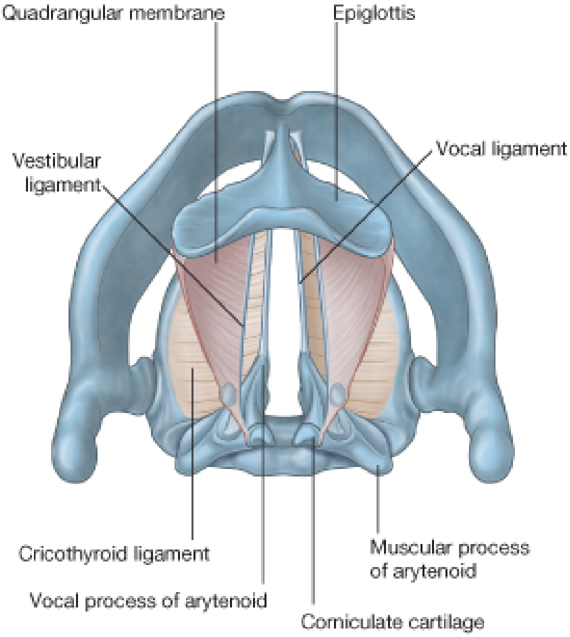


Figure 1.2: Anatomical illustration of the larynx ([2] p.954)

As illustrated in figure 1.2 it is constructed of a large number of cartilages i.e. the thyroid cartilage, the cricoid cartilage and the arytenoid cartilage, muscles and ligaments. On the inside it contains the *vocal folds* that are built by various muscles and ligaments (for more detail please refer to e.g. [3] or clinical literature like [2]).

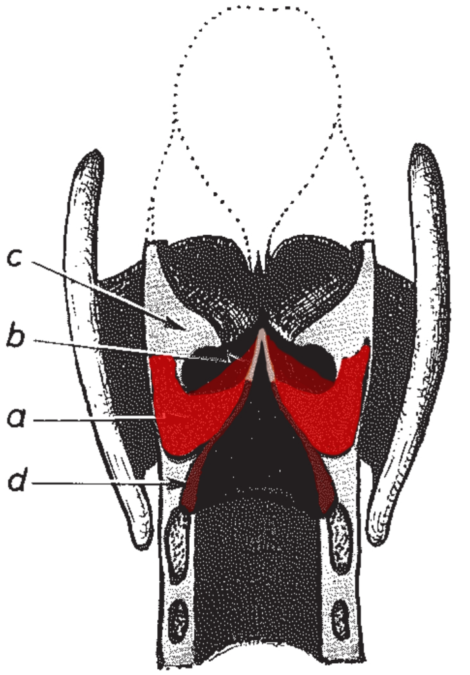


Figure 1.3: Frontal section of the larynx; (a) vocal folds (*labium vocale*); (b) vocal cord (*ligamenta vocalia*); (c) vestibular fold (*plicae vestibulares*); (d) conus elasticus ([3] p.34)

When air starts to flow through the vocal folds the *Bernoulli effect* of the air flow induces a force that causes the vocal folds to approach each other. In the moment of contact - the instant of glottal closure - the air flow stops abruptly and the force is released so the vocal folds can return to their initial position likewise allowing the air flow to start again.

This process is repeated periodically and the rate of closure can be compared to the swinging of a chord and is therefore dependent on the *tension* of the chord. Thus, we can assume that the *glottal source signal* is

1. *quasi periodic* and can be described as a set of harmonic partials and

2. the fundamental frequency is a function of vocal fold tension.

This results in a *voiced sound* with harmonic partials as it occurs in vowels or voiced fricatives (for more information on the regulation of vocal fold movement please refer to e.g. [4]).

In the case of phonation of an *unvoiced sound* on the other hand, the vocal folds *do not close*. Hence, the resulting *turbulent* air flow does not expose periodicity and thus there will not be any harmonic partials as in unvoiced fricatives and plosives. The processes in the larynx are also responsible for the perceptive attributes *pressedness* or *breathiness* of voice.

1.2.2 The Vocal Tract

The vocal tract consists of *pharynx*, *mouth* and the *nasal cavities* and is responsible for *articulation*.

In contrast to the larynx, no active sound generation takes place here. The effect of the vocal tract is commonly described by means of a *resonator*. The resonator consists of 3-4 major resonance peaks that are critically influenced by the position of jaw, tongue, velum and the lips. Furthermore, the so-called *formant frequencies* are the major distinctive feature in articulation, allowing us to discriminate the different phonemes.

In contrast to the oral cavities, the acoustically parallel nasal cavities form *anti resonances*. Modelling the resulting *pole-zero-model* of the vocal tract is a yet unsolved issue as the contemporary estimation of poles and zeros of the transfer function leads to numerical problems [5].

As, on the other hand, in history *all-zero* models have been successfully used to describe the vocal tract resonator we have decided to remain with this simplification in our work.

1.3 Source-Filter Model Of Speech

From the previous sections we know that the speech signal consists of two major components: the *glottal source* signal $g[n]$, which is responsible for the harmonic properties of the human voice and the *vocal tract resonances* $H(z)$ that introduce spectral shaping to the glottal source and are crucial to the formation of the phonems and thus for speech intelligibility.

As coupling between these two components is very weak interaction between phonation and articulation becomes negligible. Thus, we can assume the two parts to be *independent* [1]. This, as a consequence, allows for linearization and separation of excitation and

[1] note that this *partially valid* simplification has been dealt with in literature as *subglottal coupling* and there also exist models of speech production that take into account a certain amount of dependency

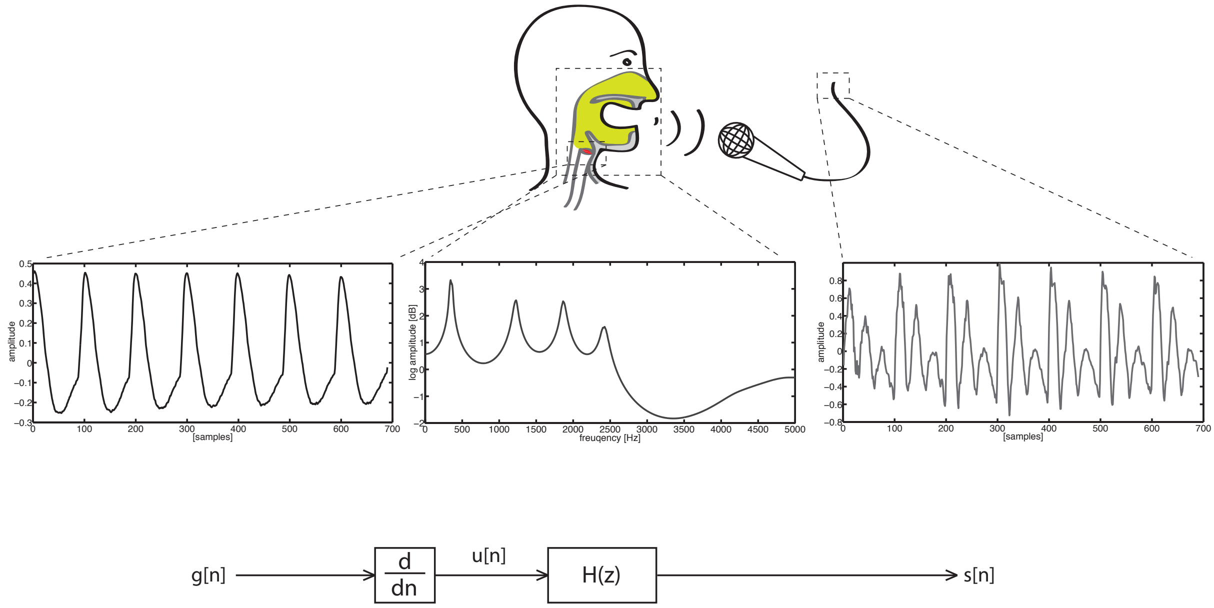


Figure 1.4: The source-filter model of speech; the lungs press air through the vocal folds and according to their tension a periodic opening and closing $g[n]$ sets in (the left graph displays the volume velocity waveform); radiation at the vocal folds $\frac{d}{dn}g[n]$, henceforth $u[n]$, excites the vocal tract system $H(z)$ as displayed in center graph (the lip radiation can be approximation by deriving the volume velocity); the filtered vocal signal $s[n]$ leaves the mouth (right graph)

transmission system [1].

Due to the lip radiation effect (see [1], chapter 5) that can be modelled as the *first derivative* of the glottal source signal the actual excitation of the vocal tract can be denoted as $u[n] = \frac{d}{dn}g[n]$.

As the vocal tract can be regarded as a *linear time-variant filter* $H(z)$ that is BIBO stable and has a system response $h[n]$ we can assume that the resulting speech signal $s[n]$ can be written as the convolution of source and filter like

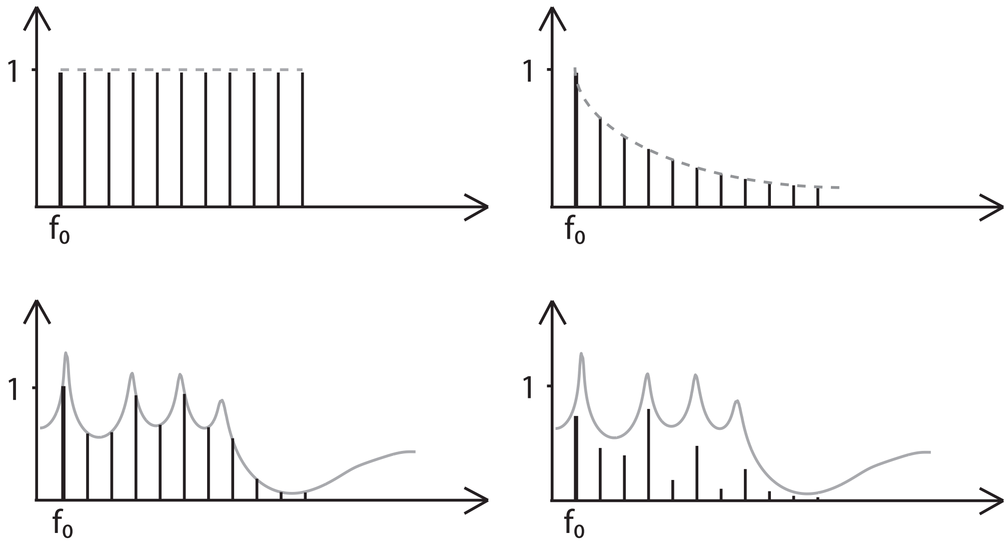
$$s[n] = u[n] * h[n] \tag{1.1}$$

or in the z-domain

$$S(z) = U(z)H(z) \tag{1.2}$$

where $U(z)$ denotes the z-transform of the glottal source derivative. Obviously, we are only able to measure the speech signal and hence the combined transfer function of glottal source and vocal tract - clearly there is an infinite set of possible combinations. Setting

of GS and VTR (especially during the open phase)



(a) Glottal source is uniform

(b) Glottal source is not uniform

Figure 1.5: Glottal source spectral properties (above) and the effect of the vocal tract (solid grey line) on the speech signal (below); note that in the non-uniform case there is no immediately obvious relation between partial amplitudes of the speech signal and the vocal tract response

the goal to estimate only one of the two parameters leads us to the problem of *blind deconvolution*. To facilitate this problem we can exploit some special properties of the glottal source signal as described in the following section.

Figure 1.4 gives us an illustration how the path of a voiced speech signal looks like.

1.3.1 Properties Of The Glottal Source

Due to the quasi periodic oscillation of the vocal folds the glottal source signal (GS) can be regarded as a harmonic signal of M partials. If $g[n]$ was white or rather *uniform* ($G(z) = 1$) we could assign the entire spectral envelope we measure in the speech signal to the vocal tract response (VTR). Unfortunately, this is not the case for most of the time but there is an approximation that we will exploit later in chapter 4.

To illustrate the problem we can decompose the the spectral envelope of the glottal source into

$$G(z) = W(z)\Phi(z) \tag{1.3}$$

where $\Phi(z)$ is the z-transform of a *perfectly harmonic* and *uniform partial gain* signal as e.g. illustrated in figure 1.5(a). The term $W(z)$ denotes a weighting term that is *approximately* exponential as depicted as the grey dashed line in figure 1.5(b).

The overall spectral envelope thus is a combination of the vocal tract response as well as the amplitude distribution of the partials of the glottal source like described in eq. 1.3. For the glottal source Sundberg [4] points out that the decrease of amplitude is about 12dB per partial. Though we could make use of this logarithmic relation for our computation we have no reliable information on the actual partial distribution.

We will see that there is a special property of the glottal source that enables us to circumvent this problem and compute a very accurate estimate of the VTR. So let let us take a closer look at the vocal source signal.

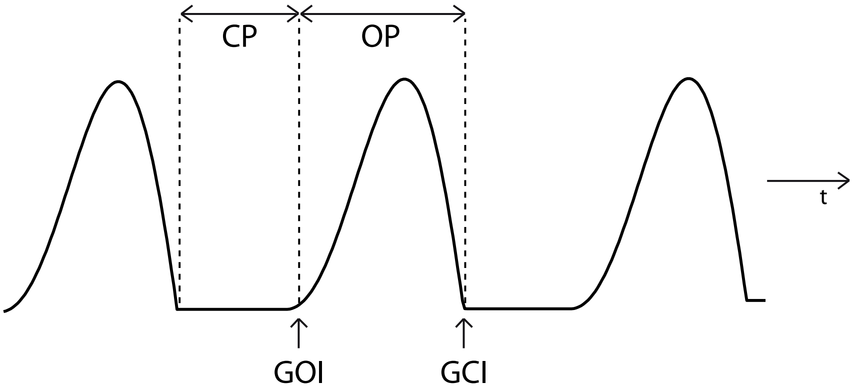


Figure 1.6: Volume velocity waveform; in the *closed phase (CP)* the vocal folds are closed and no air flows; at the *instant of glottal opening (GOI)* air flow sets in and ends at the *instant of glottal closure (GCI)*; this time segment is also called the *open phase (OP)*

In general, the glottal source signal is constructed as illustrated in figure 1.6: there exists an *open phase* (OP) where air flows through the vocal folds. It starts with the *instant of glottal opening* (GOI) and ends with the *instant of glottal closure* (GCI).

The following *closed phase* (CP) on the other hand can defined as the time interval where *no airflow takes place.*

After radiation at the vocal folds the actual waveform that excites the vocal tract looks like displayed in figure 1.7. Here the CP is preceded by very impulsive event that coincides with the instant of glossal closure. This impulse is also referred to as the *major excitation* of the vocal tract resonator.

The following closed phase is of particular interest for us because

1. the impulsive excitation at the GCI can be assumed to be *approximately white*,

2. *no further excitation* takes place while the vocal folds are closed and

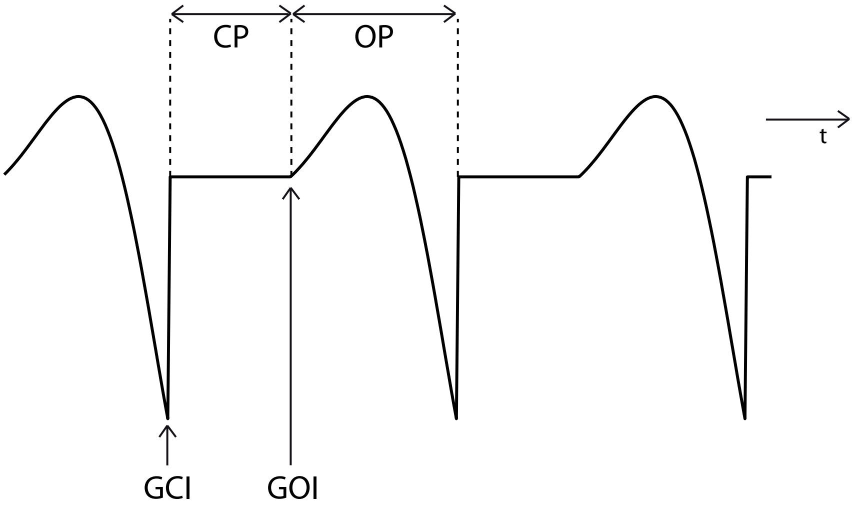


Figure 1.7: Derivative of the volume velocity waveform; after radiation at the vocal folds this is the signal that excites the vocal tract; again *closed phase (CP)* and *open phase (OP)* are defined by the *instants of glottal opening (GOI) and closure (GCI)*; note that in the derivative the GCI corresponds to a well defined peak

3. little or no *subglottal coupling* [6] occurs.

If we consider these assumptions, we can modify our model of voice production so that can identify the system response in a most appropriate manner by restraining the analysis frame to the time interval of glottal closure. This allows for measuring only the impulse response of the vocal tract driven by the major excitation free of any additional energy introduced to the system. Hence, we can assume $U(z) = 1$ for the duration of the closed phase and as a consequence the measured spectral envelope of speech corresponds directly to the VTR ($S_{CP}(z) = H(z)$).

This special approach of decomposing the speech signal into filter and source and will be described detailled later in chapter 4.

1.3.2 Properties Of The Vocal Tract

Apart from the two classes of voiced and unvoiced sounds, the glottal source signal varies primarily in fundamental frequency. Spectral variations are rarely observed but can occur e.g. when the open quotient changes [4], [7]. Thus, the principal task of the vocal tract for performing *articulation* is to shape the existing spectral envelope.

On the basis of acoustic sound propagation in the superglottal cavities a set of resonance emerge. More precisely, there are two to three major resonance peaks or *formants* that allow the human auditory system to discriminate the different phonemes. In table 1.1 an overview of formant locations of the german vowels is given.

german vowel	F_1	F_2
U	320Hz	800Hz
O	500Hz	1000Hz
A	1000Hz	1400Hz
E	500Hz	2300Hz
I	320Hz	3200Hz

Table 1.1: Table of german vowels and the approximate center frequencies of the first two formants

Besides articulation, the spectral envelope of speech also critically influences perception of emotion. In [8] e.g. enjoyment, fear and anger are related to an increase of formant gain in the frequency rage of $0 - 3,7kHz$.

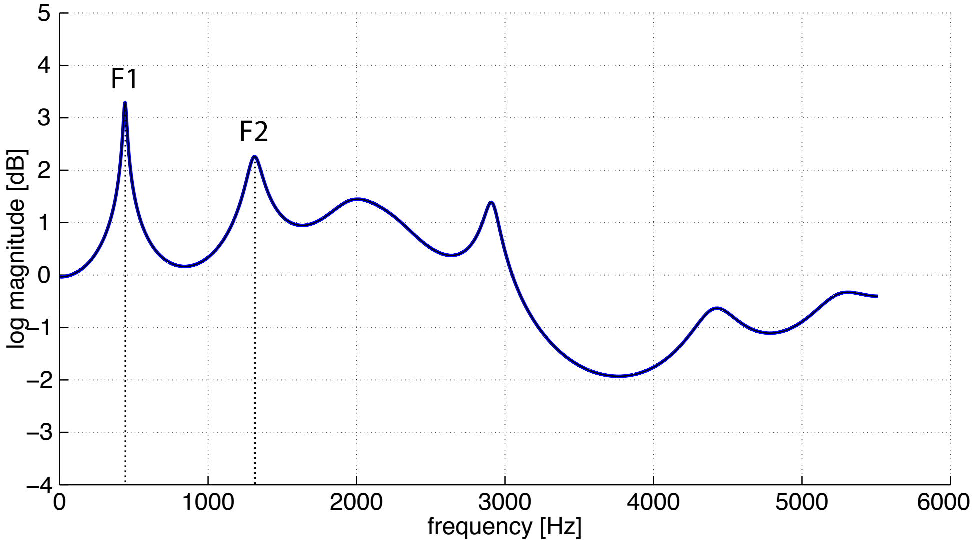


Figure 1.8: An example of the vocal tract transfer function of the sung vowel [ə]. The first two formant resonances are located at $F1 \approx$ 350Hz and $F2 \approx$ 1200Hz

Concerning the possibilities of modelling the VTR by means of a digital filter, the most common solution is using an *IIR all-pole* representation: the poles are positioned according to the formant frequencies and by using an appropriate amount p of complex-conjugate poles also the quality and gain of the formants can be modelled. This way we can write a representation of the vocal tract transfer function $H(z)$ as

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{1.4}$$

where a_k denotes the k-th coefficient of an order p IIR filter.

1.4 Lossless Tube Model

A widely used alternative to modelling the vocal tract response by means of an all-pole direct form filter has been derived from the acoustic point of view on sound propagation in the human vocal tract [1]. On this behalf, the entire cavities above the glottis can be modelled by a set of *lossless tubes* with different diameters and lengths but constant cross-sectional areas. Such a lossless tube model is depicted in figure 1.9.

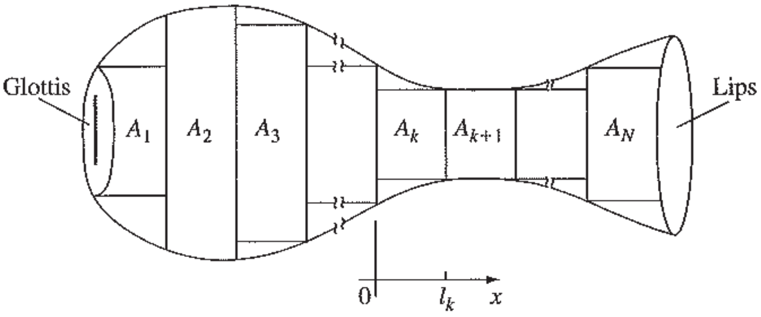


Figure 1.9: Concatenation of N lossless acoustic tubes ([1] p.214)

Basis of this model is the acoustic transfer function of a single uniform lossless tube

$$V_a(s) = \frac{1}{\cosh\left(sl/c\right)} = \frac{2e^{-sl/c}}{1 + e^{-s2l/c}} \tag{1.5}$$

where s denotes the Laplace transform of the complex angular frequency $j\omega$, l the length of a lossless tube and c the sound propagation speed.

According to eq. 1.5 the resonances of such a tube can be rewritten as

$$s_n = \pm j \left[\frac{(2n+1)\pi c}{2l}\right], \quad n = 0, \pm 1, \pm 2, \ldots \tag{1.6}$$

where the integer n denotes the n-th pole.

By applying appropriate boundary conditions for the glottis and the lips ([1] p. 217ff) and concatenating N lossless tube segments we can accurately model the vocal tract.

The resulting system transfer function is not characterized by an impulse response as for classic direct form filters but by a set of *reflection coefficients*. This also leads to an argument why lossless tube models have become a popular modelling technique: it is founded in the close relation to the *lattice formulation* of digital filters [1] which allows for performing continuous time modifications of signals while conserving filter stability under all circumstances.

In a later section we will present the resulting implementation of a *linear time-variant Lattice filter* to model the vocal tract filter.

1.5 Speech vs. Singing

Basically, singing involves the same procedures of voice production as speech does. The primary difference between speech and singing lies within the majority of voiced sounds. Though, of course, sung phrases still consist of words that are built by phonemes of all kinds, the perceived *pitch contour* that we regard as musical phrase relies on vowels.

Additionally, the occurrence of long, sustained tones while singing e.g. simply a long note or also while performing a *coloratura* can only by accomplished with voiced sounds.

Thus, as a substantial simplification, we will focus our investigations in the course of this work on the group of *voiced* sounds.

1.6 What is Vibrato?

One of the most prominent differences between speech and singing is the *vocal vibrato*. According to Sundberg's definition of vibrato [9] it can be described as a "*regular fluctuation in pitch, timbre and/or loudness*". Thus, it can be characterized by following parameters as also illustrated in figure 1.10

1. rate,
2. extent,
3. waveform and
4. regularity.

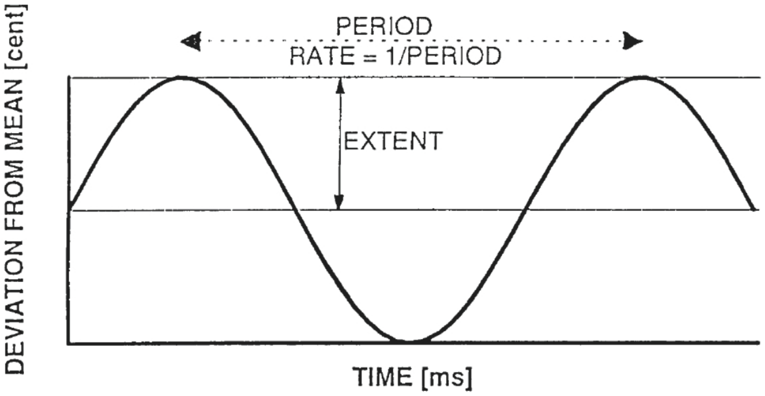


Figure 1.10: Physical parameters to describe vibrato [9]

Sundberg regards the variation of pitch to be the predominant characteristic of vocal vibrato. Thus, the vibrato *rate* describes the modulation frequency by which the (imagined) center pitch is modulated. For classically educated singers it usually lies within a range of $5 - 8Hz$ according to [9].

The corresponding *extent* describes the amount of frequency modulation. Typical values here are in the range of roughly ±1 semitone pitch deviation or approx. 6% of the center frequency. An interesting observation Sundberg describes [9] that singers tend to *increase* the vibrato extent when increasing loudness e.g. in the case of a *crescendo.*

Waveform and *regularity* reflect the temporal form of modulation and may be useful to classify different types of vibrato [10].

The above definitions can also be applied in the same manner to the modulation of the *amplitude* of a vocal signal.

2 Determination Of Voice Model Parameters

2.1 Introduction

If we consider the model of speech production presented in chapter 1 we can derive a set of fundamental model parameters.

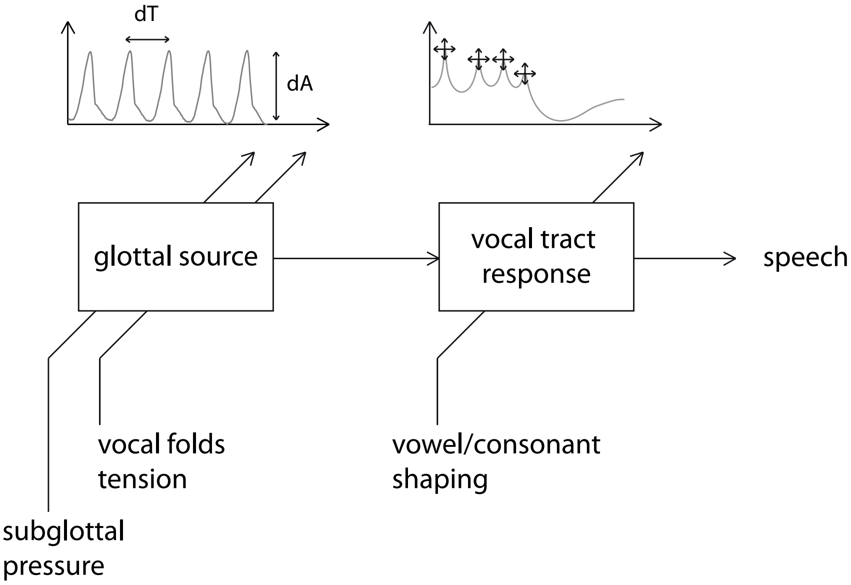


Figure 2.1: Parameters of the voice production model; temporal and spectral shape of the glottal source are primarily defined by subglottal air pressure from the lungs and the vocal folds tension; vowel/consonant shaping results in a set of formant resonances with corresponding center frequencies and gains

Regardless of the actual modelling technique we can simplify the model as illustrated in figure 2.1 into

1. *source parameters*:
   a) fundamental frequency
   b) amplitude
   c) open quotient
2. *vocal tract parameters*: formant frequencies and gains

In this chapter we will present and discuss a set of techniques to extract these parameters from real world audio signals and two useful ways to illustrate the measured data. In the end of this chapter we will also introduce the results of a small empirical study regarding the evolution of the voice production parameters in the course of vibrato generation.

2.2 F0 Tracking

Measuring pitch perception is commonly achieved by measuring the fundamental frequency or first partial tone of an instrument. There do exist exceptions like the tonal components of bell sounds that do not directly correspond to a fundamental frequency but those exceptions shall not be dealt with in this work.

The definition of the fundamental frequency f_0 as the inverse period $(1/T_0)$ of a *perfectly* periodic signal $x[n]$ yields some problems: on the first hand, a perfectly periodic signal - given that it even exists - does not carry any information i.e. modulation. Therefore, in practical cases it is not interesting.

Consequently, regarding *not perfectly periodic* signals, the definition of T_0 as the "*smallest positive member of the infinite set of time shifts that leave the signal invariant*" [11] cannot entirely hold. Obviously, voice and instrument sounds do not belong the group of perfectly periodic signals but the can be assumed to be *short-time stationary* for at least a few periods.

Therefore, block signal processing techniques can be utilized the explore the evolution of a possible fundamental frequency.

2.2.1 Overview

Generally, two main concepts are described in literature:

- spectrum based methods and
- time domain based methods.

Frequency domain methods usually use some kind of STFT sprectrogram and subsequent peak-picking. These methods imply a strong formation of the f_0 in contrast to its partials and - maybe more critically - to the noise and interference levels. Spectral domain methods are easy to compute by using an intelligent set of FFT parameters but come with certain inaccuracy introduced by the STFT spectral and temporal resolution (the usual trade-off between high resolution in time or frequency).

Time domain methods on the other hand have been developed mainly on basis of autocorrelating the signal $x[n]$ and determining the first maximum within a certain search range. This maximum then corresponds to the fundamental period T_0 and thus the inverse f_0. The complexity of finding this maximum increases among other things with the amount and amplitude of partials. The autocorrelation representation becomes ambiguos and causes common errors like mixing up octaves. This is primarily caused by a strong first harmonic but also strong resonances in the lower formant regions might interfere with the alorithm.

One disadvantage of temporal methods comes from the high amount of computational load of the autocorrelation function (ACF). Though there is an alternative formulation by deriving the ACF from the power spectral density (PSD) that can be computed more cheaply by means of the FFT but this formulation is not suitable for all approaches as the ACF decreases with τ due to finite window length. This effect will be discussed later in section 2.2.2.

Other approaches include cepstral and linear predictive analysis (long term prediction). An overview of current f_0 tracking algorithms is given in [11], [12], [13] and [14]. As a result of this comparison we decided to choose the *YIN algorithm* as proposed in [11] for this work as it allows for high temporal accuracy and frequency resolution, low latencies (for a hypothetical real-time implementation) and only very low error rates concerning e.g. octave mismatches.

2.2.2 The YIN Algorithm

As mentioned before the YIN algorithm performs processing in the temporal domain. A rough schematic of the procedure is provided in figure 2.2.

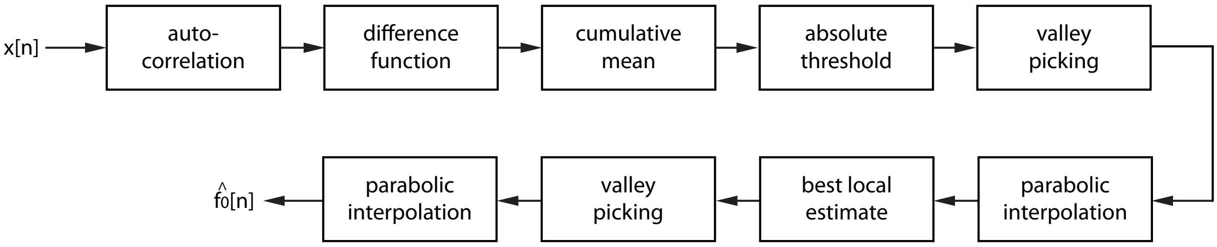


Figure 2.2: Flow chart of the YIN pitch estimator

Stage I: Autocorrelation

The signal $x[n]$ is divided into M blocks of window length W with a hop size H. For every block the autocorrelation up to a certain lag of τ is computed by

$$r_m(\tau) = R_{xx}(m, \tau) = \sum_{j=1+mH}^{1+mH+W} x_j x_{j+\tau} \tag{2.1}$$

where $r_m(\tau)$ denotes the m-th row of the autocorrelation matrix R_{xx} and the block index m is defined as $m = 1, 2, 3, \ldots, M$.

This formulation stands in contrast to the more commonly used definition of

$$r'_m(\tau) = \sum_{j=1+mH}^{1+mH+W-\tau} x_j x_{j+\tau} \tag{2.2}$$

which exposes a reduced integration/summation window length while increasing the value of τ (note that the upper summation boundary in eq. 2.2 decreases with τ). This stems from the finite window lengths used in block signal analysis and zero padding outside the defined segment. As a consequence it leads to a *decreasing* envelope of the ACF over τ and hence favors certain period/frequency ranges. This discrepancy is illustrated in figure 2.3.

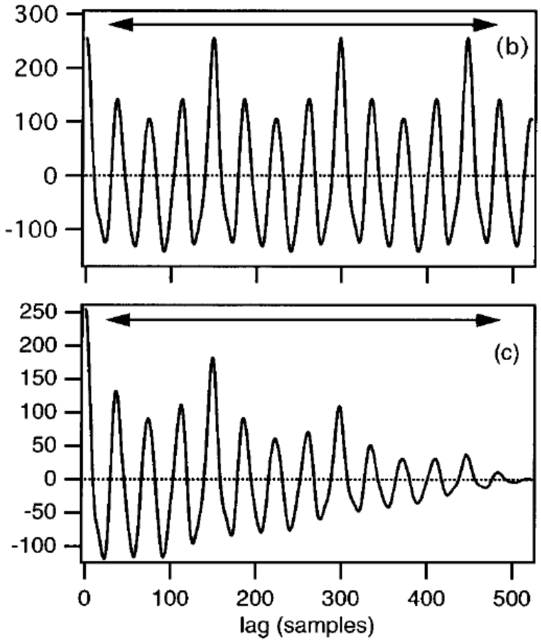


Figure 2.3: Comparison of two definitions of the autocorrelation function: the upper figure shows the ACF according to eq. 2.1; the bottom figure shows the ACF according to 2.2 (taken from [11])

To avoid this favouring the authors suggest the usage of formulation 2.1. Obviously, we still have to use finite length windows for our analysis but we can simulate infinitely long windows by intelligently defining calculation limits: the *static window* has to have at least τ more samples than the *sliding window*. In this way, we never have to multiply by zero-padded values. Of course this comes with the drawback larger overall window lengths.

In the case of a samplewise hop $H = 1$ of the analysis window as it has been employed in this work we have derived an efficient matrix formulation. The signal $x[n]$ is processed into a block matrix $\mathbf{X}$ of the dimension $N \times \tau$:

$$\mathbf{X} = \begin{bmatrix} x[1] & x[2] & \dots & x[\tau-1] & x[\tau] \\ x[2] & x[3] & \dots & x[\tau] & x[\tau+1] \\ x[3] & x[4] & \dots & x[\tau+1] & x[\tau+2] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x[W] & x[W+1] & \dots & x[W+\tau-1] & x[W+\tau] \\ x[W+1] & x[W+2] & \dots & x[W+\tau] & x[W+\tau+1] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x[N-\tau] & x[N-\tau+1] & \dots & x[N-1] & x[N] \end{bmatrix}_{N \times \tau} \tag{2.3}$$

Then the multiplication of a signal excerpt $\mathbf{x}_k$ of length W

$$\mathbf{x}_k = \begin{bmatrix} x[k] \\ x[k+1] \\ \vdots \\ x[W] \end{bmatrix}_{W \times 1} \tag{2.4}$$

where k denotes the hop iteration step is multiplied with the corresponding matrix excerpt $\mathbf{X}_k$ with the dimension $W \times \tau$

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{X}_{k,1} & \mathbf{X}_{k,2} & \dots & \mathbf{X}_{k,\tau} \\ \mathbf{X}_{k+1,1} & \mathbf{X}_{k+1,2} & \dots & \mathbf{X}_{k+1,\tau} \\ \dots & \dots & \ddots & \dots \\ \mathbf{X}_{k+W,1} & \mathbf{X}_{k+W,2} & \dots & \mathbf{X}_{k+W,\tau} \end{bmatrix}_{W \times \tau} \tag{2.5}$$

leads to the k-th line of the autocorrelation matrix R_{xx}

$$\mathbf{r}_{xx_k} = \underbrace{\mathbf{x}_k^{\mathsf{T}}}_{1 \times W} \underbrace{\cdot \mathbf{X}_k}_{W \times \tau} \tag{2.6}$$

which is equivalent to

$$r_k(\tau) = \Big[x[k]x[k] \quad x[k]x[k+1] \quad x[k]x[k+2] \quad \dots \quad x[k]x[k+\tau]\Big]_{1\times\tau} \tag{2.7}$$

Hence, R_{xx} can be computed by performing this operation for all values of $k = 1, \dots, N - W$. Shortening the analysis to $N - W$ samples is due to the ambiguities that occur when performing autocorrelation over a zero-padded signal segment. We avoid this by not performing analysis on samples outside the signal length.

In contrast to other definitions of the autocorrelation that compute the fundamental period from samples *in the past* we decided to use samples *from the future*. This is due to the fact, that later determination of the *instants of glottal closure (GCI)* (see section 4.2) relies on information concerning the *next* instant rather that the last.

Stage II: Difference function

The assumption of a quasi-periodic signal allows for further processing. Modelling the signal as a shift-invariant function

$$x[n] - x[n+T] = 0 \quad , \quad \forall n \tag{2.8}$$

which also holds after summation of squares differences

$$\sum_{j=n+1}^{n+W} (x[j] - x[j+T])^2 = 0 \tag{2.9}$$

has the consequence that a possible period can be determined by finding a zero in the difference function $d_n(\tau)$ defined as

$$d_n(\tau) = \sum_{j=1}^{W} (x[j] - x[j+\tau])^2 \tag{2.10}$$

Expanding the binomial term leads to

$$d_n(\tau) = r_n(0) + r_{n+\tau}(0) - 2r_n(\tau) \tag{2.11}$$

or in matrix form

$$D(n,\tau) = R_{xx}(n,0) + R_{xx}(n+\tau,0) - 2R_{xx}(n,\tau) \tag{2.12}$$

This formulation allows for following interpretations: the first two terms correspond to energy terms. $r_n(0)$ is not dependent on τ and hence has no significant influence on the difference function except for a linear bias. The third term $-2r_n(\tau)$ is dependent on τ and equals two times the inverse ACF but no change can be expected except that former local maxima are transformed into local minima. The actual improvement stems from the middle term $r_{n+\tau}(0)$ which also varies with τ and abolishes the direct relation between maxima of $r_n(\tau)$ and minima in $d_n(\tau)$. According to the authors error rates could be reduced to less than 20% of the initial values.

Stage III: Cumulative mean normalized difference function

The difference function as described in the previous section is prone to a major systematic problem: similar to the ACF it exposes a first "candidate" at zero-lag - here a perfect candidate match equals 0. Due to the imperfect periodicity of the signals we have to deal with T_0 candidates *close to zero* but not necessarily *equal to zero*. This yields a problem of global versus local minimum detection. Setting an absolute and static lower threshold is not satisfactory. Normalizing the difference function $d_n(\tau)$ to its cumulative mean circumvents this problem. The new function $d'_n(\tau)$ starts with 1 at zero lag remaining at relatively high values at low lags and drops below 1 where the difference function is below average.

Again in matrix formulation this operation can be written as

$$D'(n,\tau) = \begin{cases} 1, & \text{if } \tau = 0, \\ \dfrac{D(n,\tau)}{\dfrac{1}{\tau}\displaystyle\sum_{j=1}^{\tau} D(n,j)} & \text{otherwise} \end{cases} \tag{2.13}$$

One benefit that comes with this processing step is independency from the original signal amplitude. This shall be discussed further in section 2.2.3.

Stage IV: Absolute threshold

The absolute threshold is globally applied to the prior steps and is a first instance of restraining a search range. Only dips that fall below a certain threshold value θ remain as candidates. If no suitable dip is left the *global minimum* is chosen.

Obviously, the absolute threshold does not employ any "intelligence" but it allows for minimizing the systematic problem of the autocorrelation approach to pick dips at "too low" lags which normally result in octave errors.

So for every line d'_k of matrix D' we calculate following conditional relation

$$d'_{k,limit}(\tau) = \begin{cases} d'_k(\tau) & \text{if } d'_k(\tau) < \theta, \\ \min d'_k(\tau) & \text{if } \nexists\ d'_k(\tau) < \theta \ \forall\tau \ \wedge \ \tau = \operatorname{argmin} d'_k(\tau), \\ 1 & \text{otherwise} \end{cases} \tag{2.14}$$

This yields a matrix as displayed in figure 2.5(d). Additionally, a security measure has been introduced by the second condition in eq. 2.14: for the possible case of no value below the threshold θ the algorithm picks the *global minimum* of line d'_k.

Detecting the first minima by e.g. finding the zero-crossing in the first derivative of this matrix yields the candidate set Ξ.

Stage V: Parabolic interpolation

As for lower sample rates e.g. 8kHz time resolution becomes a critical issue for T_0 detection an interpolation step is desirable. YIN uses standard parabolic interpolation to achieve subsample precision. A prior detected candidate of set Ξ and its two neighbouring values of $d'(\tau)$ are used to perform interpolation. An illustration is given in figure 2.4.

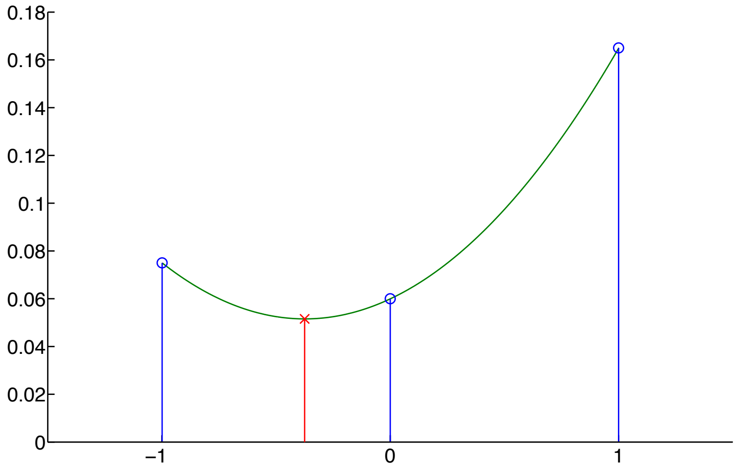


Figure 2.4: Parabolic interpolation: candidate dip at iteration step k and its preceding and following values (blue). The interpolated value (red) is derived from the parabolic interpolation curve (green)

Stage VI: Best local estimate

Up to this point the algorithm has not used any kind of probability measure on the candidates. Their computation involved only straight-forward differentiation, normalization and limitation. According to the authors the *best local estimate* criterion is similar to median smoothing or dynamic programming approaches [11]. The idea is based on following steps: For each iteration step k

1. search for $\min d'_\varepsilon(\Xi_\varepsilon)$ where the time interval ε is defined as $[k - T_{max}/2, k + T_{max}/2]$ and Ξ_ε is the estimated period at ε. This minimum is set as the

2. center of search range of e.g. ±20% of the original period on which the estimation algorithm is applied again.

This procedure is described by the authors as "shopping around the vicinity of each analysis point for a better estimate". Obviously, the interpolation step has to be performed again after this stage to regain subsample precision.

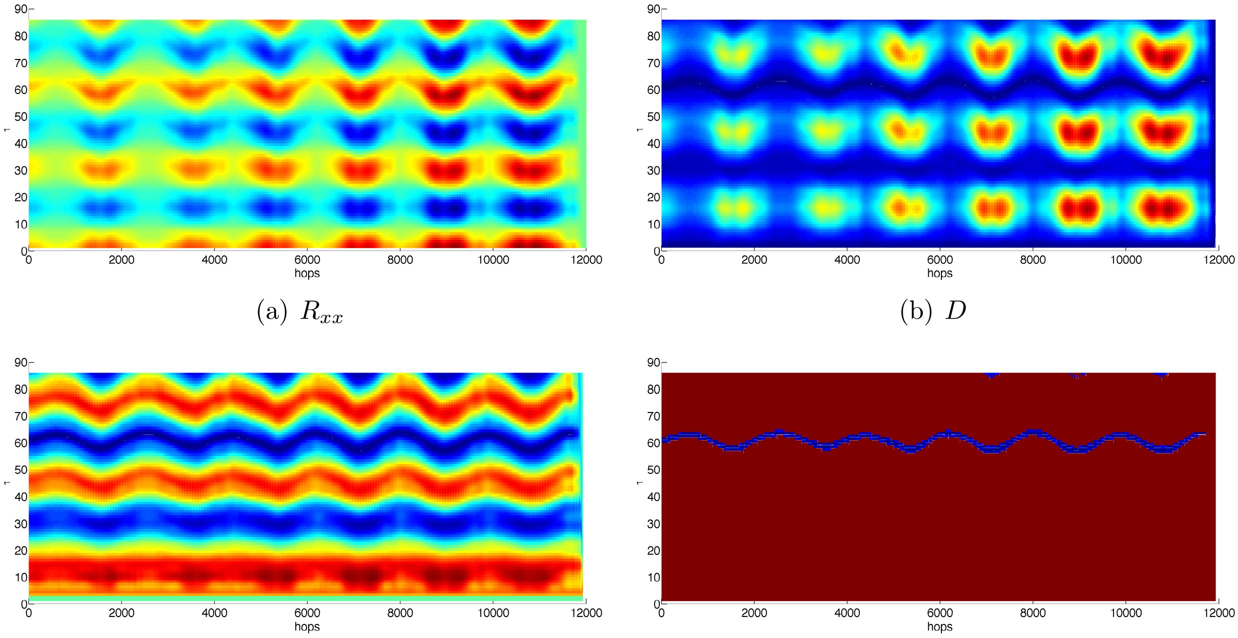
According to our opinion, this stage still has some potential of improvements. For instance situations can occur in which an octave error exposes the lowest dip although its surrounding dips form a distinct trajectory. As a consequence the algorithm would choose this wrong dip as search center and subsequently discard the prior truthfully detected estimates.

An idea to circumvent this problem would be taking the average or median minimum instead of the absolute minimum within ε as search center. This would ensure the center to truly lie in the vicinity of most estimates. Though we see the potential of improvements manipulation of the algorithm shall not be in the scope of this work an thus has not been implemented.

2.2.3 Results

In [14] a comparative study on the performance of different F_0-tracking algorithms has been carried out. It showed the very stable and relatively reliable results that could be achieve by YIN. In our work, this observation could be confirmed though, like any other approach, YIN did not perform perfectly for arbitrary signals.

Figure 2.5 shows the progress of stage I-IV where YIN processes a one second baritone recording (at 11025Hz sample rate). Length of the analysis window and maximum lag τ_{max} have been set to match a lower frequency boundary of 130Hz which seems reasonable concerning the tonal range of a baritone singer and window hop $h = 1$ is samplewise.



(a) R_{xx}

(b) D

(c) D'

(d) D'_{limit}

Figure 2.5: Stages I-IV of the YIN algorithm applied on a one second excerpt of a baritone recording; (a) autocorrelation matrix R_{xx}; (b) difference matrix D; (c) cumulative mean difference matrix D'; (d) limited matrix D'_{limit}

First figure 2.5(a) displays the autocorrelation matrix R_{xx} as computed with the parameters mentioned above. Very clearly four maxima can be seen of which only one may correspond to the actual T_0. The secondary maxima can be described as harmonic and subharmonic of the signal. By performing the *difference function stage* we obtain the difference matrix D as displayed in figure 2.5(b). It exposes a distinct clarification of the dips around a lag value of approx. 60. Still the matrix is prone to a certain dependency to signal amplitude which can be seen in the fluctuating values of maxima and minima. This can be equalised in the subsequent *cumulative mean stage* as displayed in figure 2.5(c). Afterwards the *absolute threshold* introduced in figure 2.5(d) suggests the temporal evolution of the T_0 quite explicitly.

After performing interpolation, best local estimate and interpolation again the final result of the analysis is displayed in figure 2.8.

2.2.4 Removing Octave Errors

Although the YIN algorithm makes use of a vast number of measures to avoid the occurrence of octave errors, there still exists a slight possibility that the resulting pitch contour *does* contain octave errors. We have developed a pretty straight forward approach that

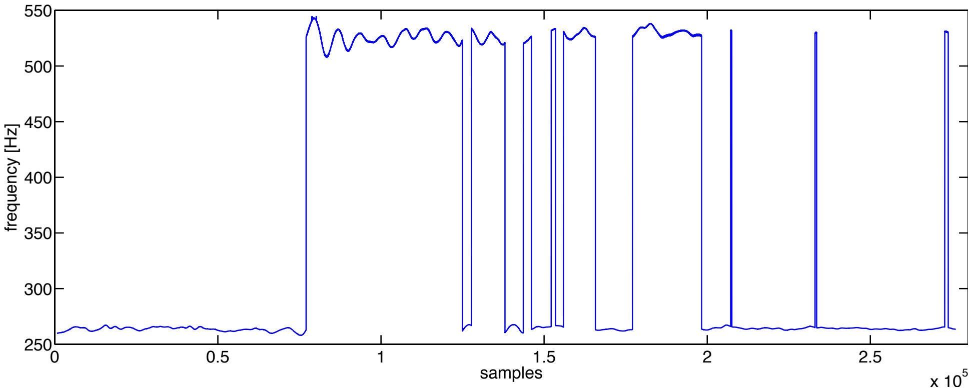


Figure 2.6: Pitch contour containing a large number of octave errors ($f_s = 44100Hz$)

1. automatically detects data segments that might correspond to octave errors and
2. uses this data to perform reassignment for the affected areas within the pitch contour

and leads to quite satisfactory results.

The detection of the defected data segments is done by finding “abnormal” jumps in pitch. Though music can contain pitch shifts of one ore more octaves this never happens in time intervals of e.g. one sample. Thus, computing the derivative of the pitch contour exposes the time instants in which the pitch jumps from the correct octave to the faulty one and back. An example is given in figure 2.6.

We can write the *detection function* $\delta[n]$ as

$$\delta[n] = \begin{cases} 1 & \text{where } \frac{d}{dn} f_0[n] > \gamma, \\ -1 & \text{where } \frac{d}{dn} f_0[n] < -\gamma, \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

The threshold parameter γ can be either a fixed value that defines a maximum frequency jump of e.g. $50Hz$ or more reasonably an adaptive threshold $\gamma[n]$ that is computed as a moving average (MA) by e.g.

$$\gamma[n] = \frac{1}{M} \sum_{m=0}^{M-1} f_0[n+m] \tag{2.16}$$

where M denotes the length of the MA filter. The resulting detection function for the given example might look as displayed in figure 2.7.

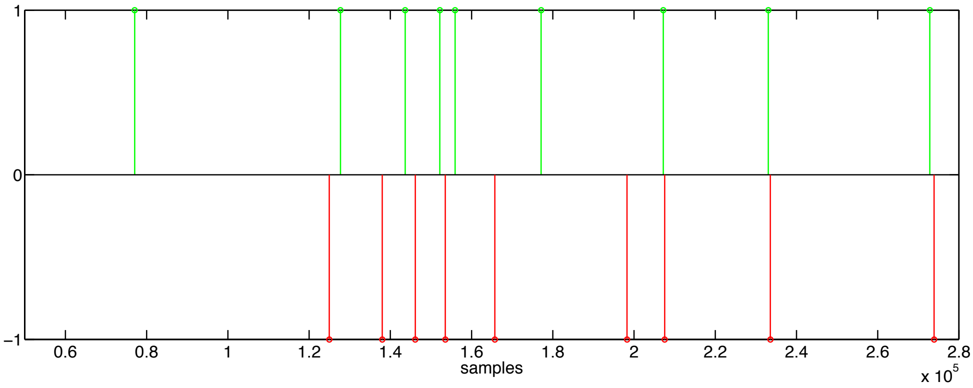


Figure 2.7: Detection function $\delta[n]$ for the occurrence of octave errors; the green lines indicate the beginning of defected data segments and the red lines the corresponding segment ends ($f_s = 44100 Hz$)

After the segment boundaries have been calculated, the further steps involve the reassignment of the affected data areas.

But before performing the reassignment steps we first have to decide by *which amount* the faulty pitch segments shall be shifted. Time domain pitch detection in harmonic signals is generally prone to result in a *too high* or in a *too low* octave. This depends on the spectral properties of the signal (i.e. the weighting of the partials) and also on the type of autocorrelation performed. In the case of the YIN algorithm our experience shows that defected pitch estimation tends to result in the *too high* octave rather than in the *too low*. This seems also comprehensible as the algorithm tries to find the *first* maximum (here of course the minimum) within the similarity measure (autocorrelation). Thus, it is prone to mistakenly choose the *double frequency*.

For this reason we have focussed our approach to find and reassign the *too high octave errors* and correct them. Im practice, we simply perform the reassignment by *deviding* the faulty pitch segments by the factor 2. A detailled illustration of the algorithm and the resulting corrected pitch contour is provided in figures 2.9(a)-(d)

Example

Figure 2.8 shows an example of a pitch contour computed by the described methods in this section. This data can be used to compute *rate*, *extent*, *waveform* and *regularity*

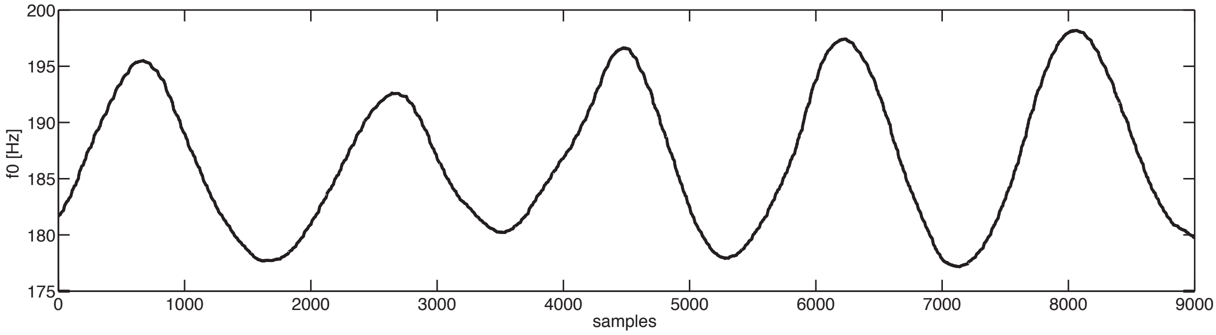
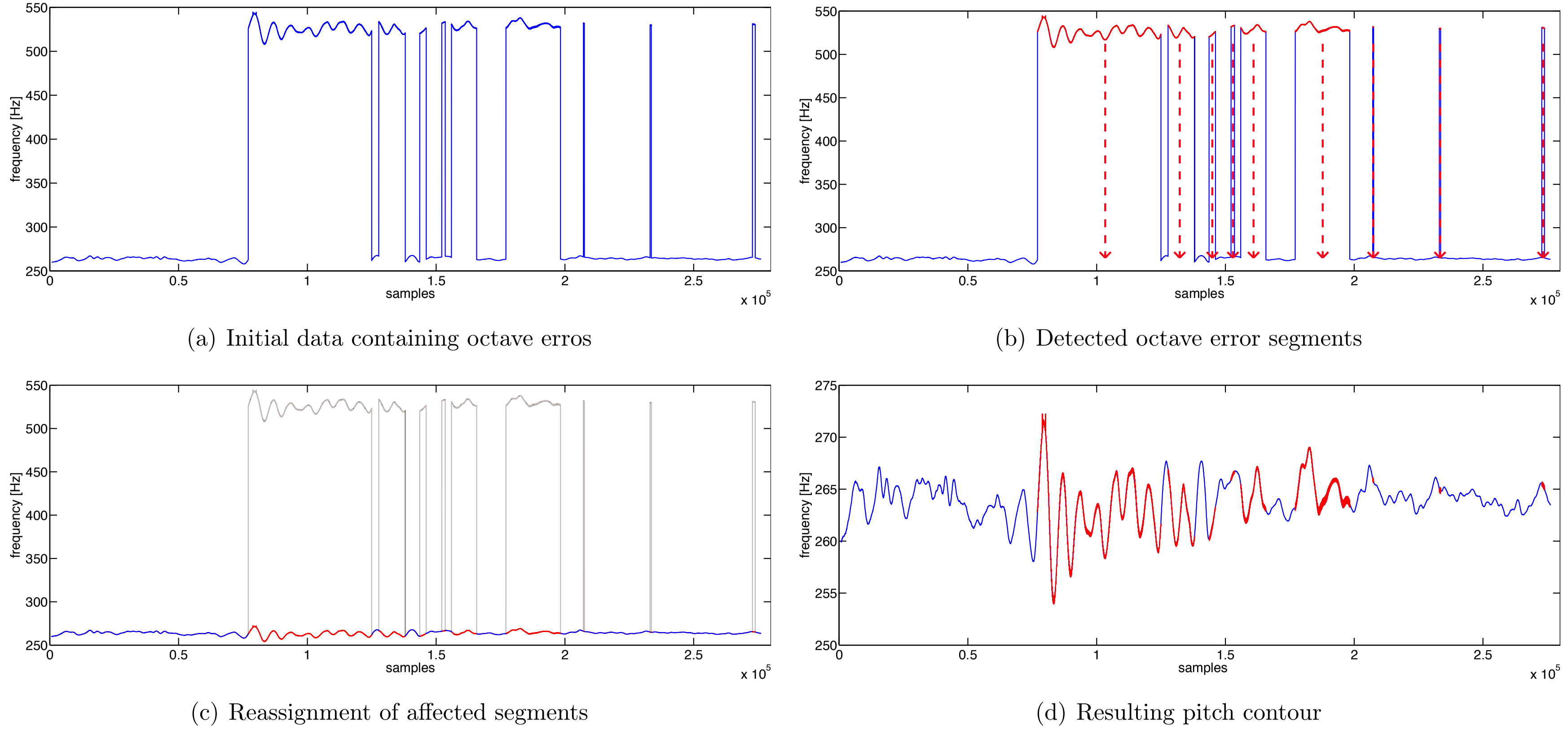


Figure 2.8: Pitch evolution of an approx. 1s baritone recording ($f_s = 11025Hz$) of a sustained vowel [ə] with well educated vibrato

of the singer's vibrato on the one hand as suggested by Sundberg in [9] and as a priori knowledge for the *inverse filtering* algorithm described later in section 3.3.



(a) Initial data containing octave erros

(b) Detected octave error segments

(c) Reassignment of affected segments

(d) Resulting pitch contour

Figure 2.9: Removal of octave errors in a pitch contour; (a) shows the original pitch contour containing a large number of octave errors (taken from the sample `a_hoch.wav`, $f_s = 44100Hz$); (b) shows the detected octave error segments that have to be reassigned as indicated by the red dashed arrowa; the reassignment operation is depicted in (c); (d) shows the resulting pitch contour with the reassigned areas marked as red line segments

2.3 Amplitude Tracking

Many works in literature that deal with singing voice vibrato tend to track the temporal evolution of the amplitudes of various partials of the audio signal. This yields additional information on the vocal tract response as described in [12] and [15]. Our approach performs decomposition into glottal source and vocal tract response later so for the moment it seems reasonably sufficient the get a global picture of the amplitude development. Similar to F_0 tracking, we want to get insight in *rate*, *extent*, *waveform* and *regularity* of the amplitude modulation. Sundberg [9] awards only very little perceptual influence to the amplitude modulation regarding the fluctuation of pitch as the primary component. Nevertheless we want to investigate the relation between frequency and amplitude modulation.

As a basis of analysis we picked a standard RMS value detector according to the formulation

$$a_{RMS}[n] = \sqrt{\frac{1}{\tau} \sum_{t=-\frac{\tau}{2}}^{\frac{\tau}{2}} x[n+\tau]^2} \tag{2.17}$$

where τ defines the “integration” time. The larger the value of τ becomes the smoother the amplitude trajectory becomes obviously with accepting the loss a certain amount of detailled information. Reasonable values have been found to lie around 25ms window length.

Example

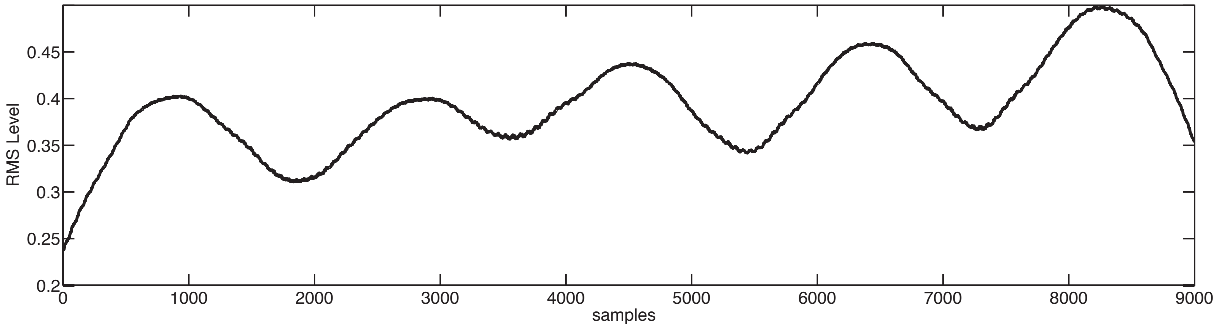


Figure 2.10: RMS level amplitude evolution of an approx. 1s baritone recording ($f_s = 11025Hz$) of a sustained vowel [ə] with well educated vibrato

2.4 Formant Tracking

According to Sundbergs definition of vibrato [9] not only fluctuation of pitch and loudness occur during vibrato but also a variation of timbre. Timbre is often referenced to as the *spectral properties* of a musical signal. In singing, the temporal evolution of timbre can be associated with the changes in the vocal tract transfer function. The major resonances, hence the *formants*, have significant influence on the production of voiced and unvoiced sounds. As pointed out by Arroabarren in [15] though, neither the spectral envelope of the GS nor the VTR itself change significantly during a vibrato cycle. Therefore, characterization of vibrato could be carried out without including the computation of the vocal tract response.

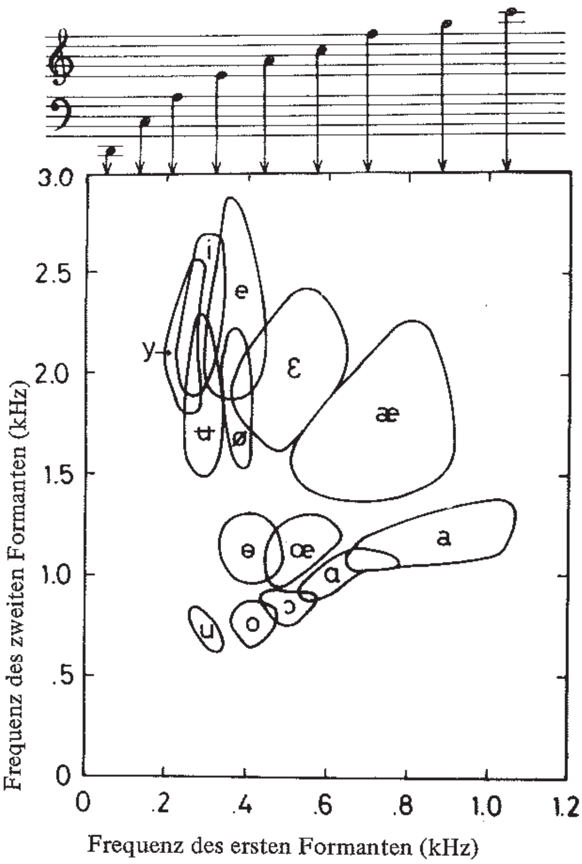


Figure 2.11: Relationship between the first two formant frequencies for a set of (swedish) vowels [4] p.39

Nevertheless, we decided employ formant tracking into our analysis system as we compute the specific resonator information in the course of glottal inverse filtering as described later in chapter 5.

Extraction of the vocal tract response coefficients is performed by *constrained closed phase covariance linear predictive coding*. This complex and multi-layered algorithm is described extensively in chapter 5 and shall not be discussed at the moment.

The resulting information consists of an all-zero FIR filter representation $A(z)$ of the *inverse* vocal tract transfer function. Thus, the inverse $\frac{1}{A(z)}$ (stability is granted by constraining root positions according to [16]) contains information on the *center frequency* and *magnitude* of the formant resonators.

This information is based on instantaneous measurements at the respective instants of glottal closure and thus introduces some kind of additional discretization to the temporal representation. But as we will point out later, due to the nature of the utilized measurement technique, the so computed estimates can also be regarded as valid also for the intermediate time intervals.

Example

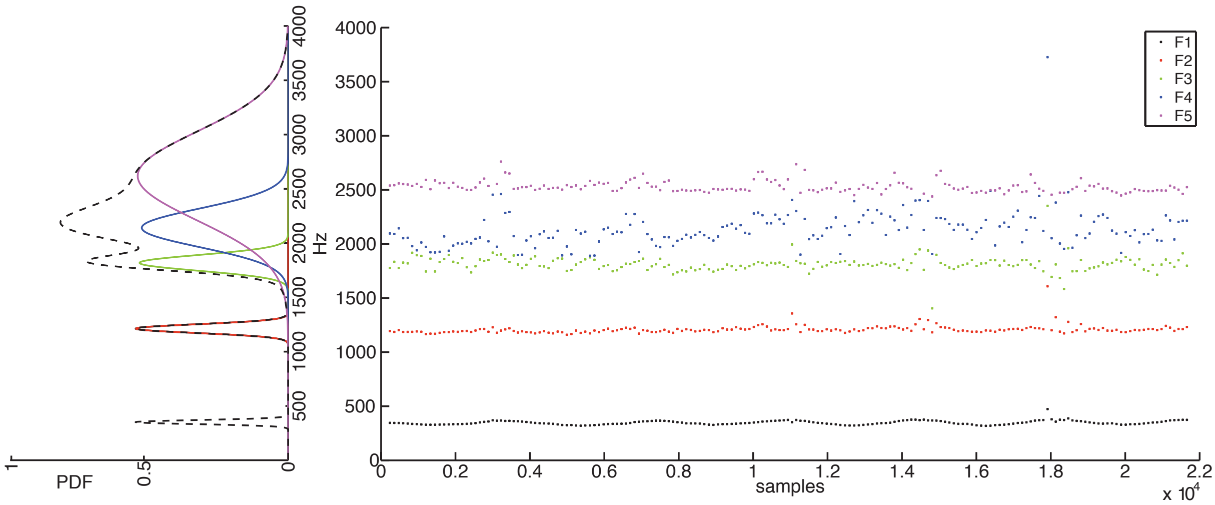


Figure 2.12: Example of tracking formants of an approx. $1s$ baritone recording ($f_s = 22050Hz$) of a sustained vowel [ə] with well educated vibrato; on the right side the temporal evolution ($f_s = 22050Hz$) and on the left side the statistic distribution of the contributing formants illustrated as mean and variance of each formant (solid lines) and the resulting gaussian mixture (dashed line); mean center frequencies [F1: $347.9Hz$, F2: $1209.0Hz$, F3: $1814.6Hz$, F4: $2137.5Hz$, F5: $2615.1Hz$], formant frequency variances [F1: $0.099Hz$, F2: $0.455Hz$, F3: $1.491Hz$, F4: $8.976Hz$, F5: $47.620Hz$]

As displayed in figure 2.12 we can illustrate the temporal evolution of the formant estimates together with their statistical distribution. In this case we have used the first two statistical moments mean and variance. Clearly visible, the variance of the first two formants (at approx. 350Hz and 1200Hz) is very small compared to the other three formant estimates. This fact, on the one hand, can interpreted as that the position of the formant does not significantly change over time - it is supported by the Arroabarren's judgement on the influence of the VTR to vibrato perception [15] - and can be very useful for classification of phonemes. The upper three formants on the other hand, expose much larger variances. This can be due to less distinct resonances but also allows for presuming that there are maybe only two formants with less resonance quality sharing a larger bandwidth.

2.5 Time-Aligned Representation

For further analysis it is crucial to have exact time alignment between the *temporal* estimations regarding the instantaneous frequency and the instantaneous amplitude as well as the *spectral* estimations of the vocal tract response.

As mentioned before, in the case of the VTR estimates we have significantly less temporal resolution. Due to the exactness of measurement and the short-time stationary nature of the vocal tract though we can assume that the VTR does not *significantly* change for the duration of a single glottal cycle.

Example

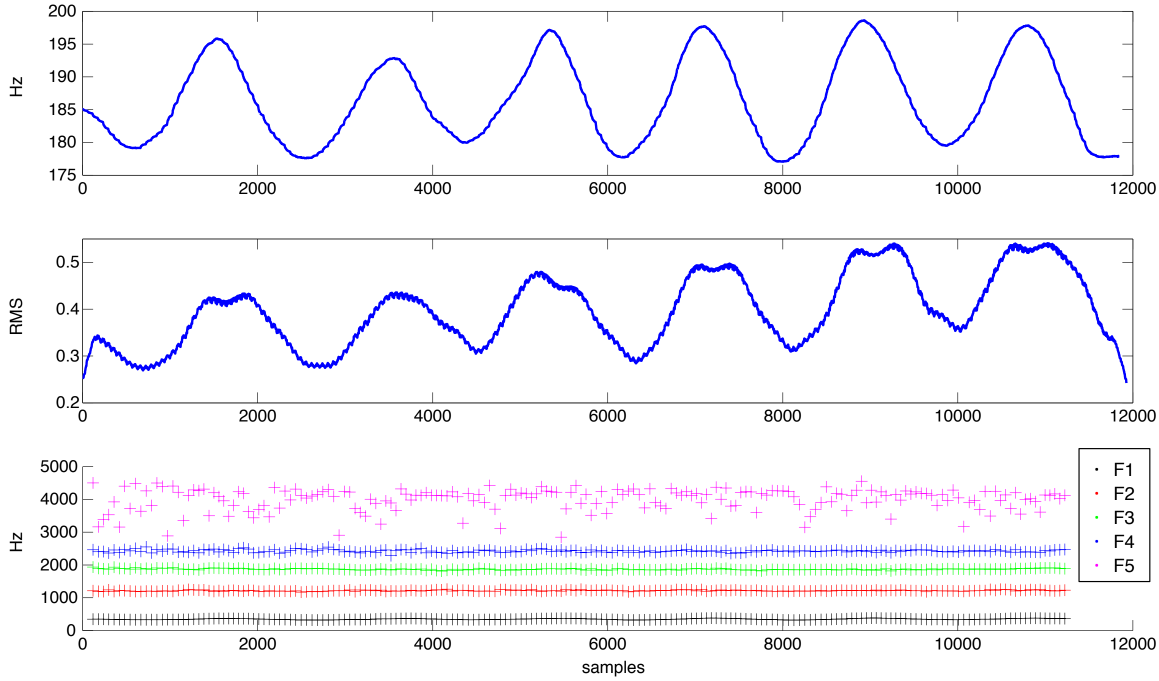


Figure 2.13: Example for a time-aligned representation of instantaneous frequency f_0 (top), instantaneous amplitude (middle) and the corresponding formant estimates of the vocal tract transfer function (bottom) of a 1s baritone recording ($f_s = 11025Hz$) with well educated vibrato.

Figure 2.13 shows an example of the relationship between instantaneous frequency, amplitude and formant positions in a time-aligned representation. One can see clearly that while pitch and loudness seem to vary in a correlated manner the positions of the formants do not change significantly nor do they expose any kind of correlation to the behaviour of pitch and amplitude.

2.6 The Cycle-Aligned Representation

The representation we also call "*glottPlot*" in the context of this thesis is an intuitive way to illustrated the processes inside the larynx that we have developed in the course of this work.

It allows for investigating the actual relations between open and closed phase of a glottal cycle in comparison to its predecessors and successors.

Idea of the cycle-aligned representation is as follows:

1. calculate the glottal source signal $u[n]$ by *constrained closed phase covariance inverse filtering* (chapter 4) and then
2. segment $u[n]$ into cycles according to the data from the detection of the *instant of glottal closure (GCI)* (section 4.2)
3. "rotate" every cycle and align them starting at the corresponding GCI.

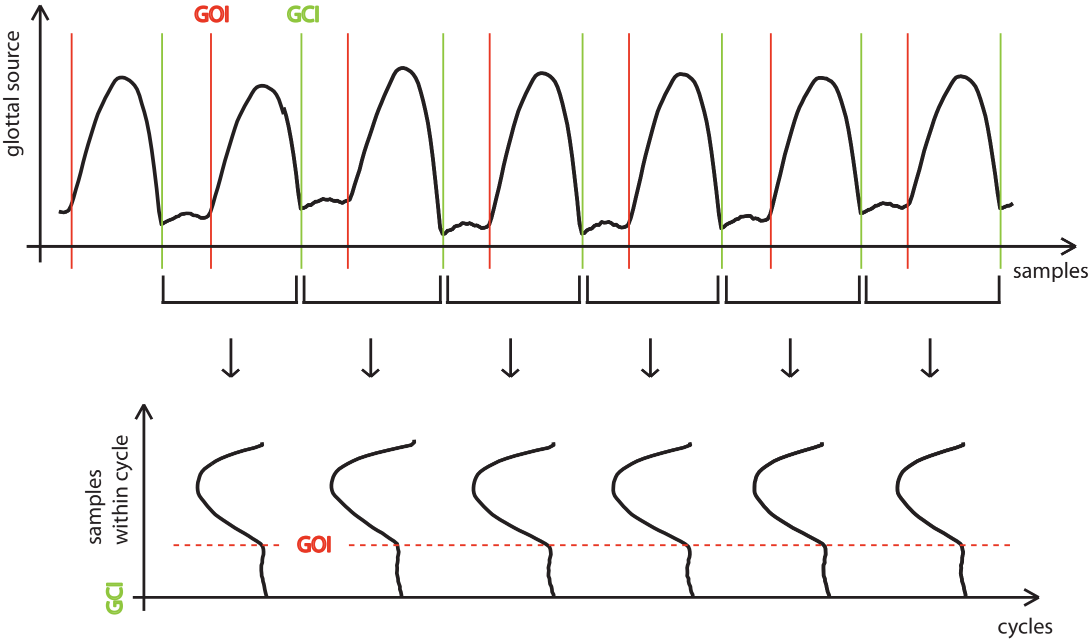


Figure 2.14: Construction of the cycle-aligned representation; the glottal source signal is segmented into single cycles that are aligned by the respective instant of glottal closeure

Naturally, the same technique can be used to illustrate the internal relations of the glottal source derivative or even the speech signal itself. The latter is also used to perform *cycle prototyping* as described in section 4.3.

Figure 2.14 illustrates the construction of the cycle-aligned representation and figure 2.15 a representative example.

Example

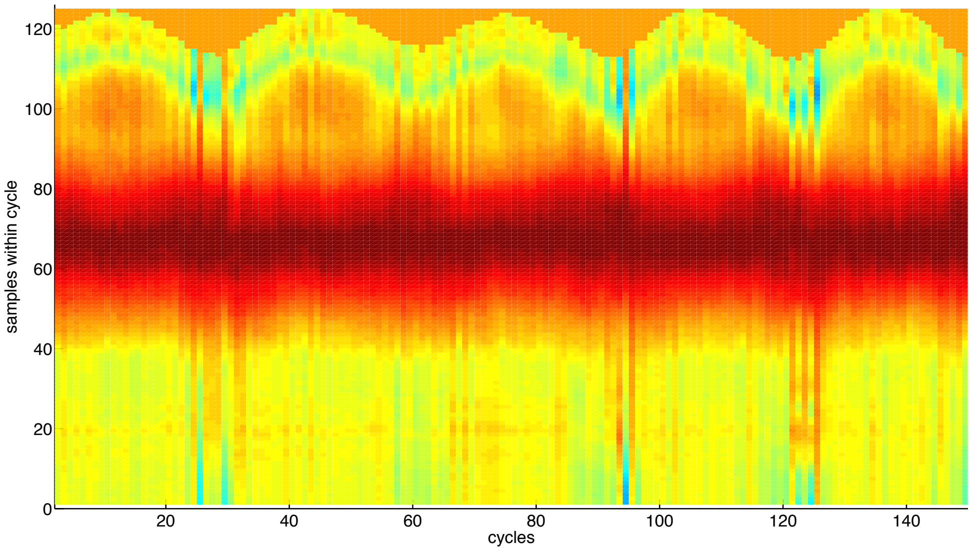


Figure 2.15: Example of a cycle-aligned representation for the volume velocity waveform of a $1s$ baritone recording of a sustained vowel [ə] with well educated vibrato ($f_s = 22050Hz$)

In the example provided in figure 2.15 one can see the variation of the pitch period that is indicated by the orange area at the top (unallocated matrix space).

Furthermore, an interesting observation can be made from this visualization: although the pitch period changes signifanctly over time (approx. ±8 samples) the duration of the closed phase (at the bottom up to approx. sample 40) *does not change*. Likewise, the first half of the open phase (approx. sample 40-80) still *does not show* any significant modulation related to the vibrato. Only the second half of the open phase - the "*release phase*" (approx. from sample 80) seems to be responsible for the variation of the overall cycle length.

We will intensify the analysis of this observation later when we will discuss the results of our small empirical study (section 2.8) and present the resulting assumptions for *modifying* *vibrato* in the respective chapter 5.

2.7 Estimation Of The Open Quotient

As discussed by several authors (e.g. [17], [18], [19], [20], [21]) the direct estimation of the *instant of glottal opening (GOI)* is very much more complicated that deriving the instant of glottal closure. Generally spoken, this is due to the fact the the process of opening corresponds to a fairly slow increase of air flow through the glottis compared to the abrupt stop of airflow at the instant of closure. Thus, recalling the temporal shape of the glottal source derivative (acoustic radiation, see figures 1.6 and 1.7) we often cannot determine a distinct onset.

$$oq = 1 - cq = \frac{T_o}{T_o + T_c} = \frac{T_o}{T} \tag{2.18}$$

Fortunately, closed phase inverse filtering algorithms have shown to be more tolerant to including data from the *following* open phase compared to the effects of including data from *before* the GCI.

We use this knowledge and retreat from trying to compute GOIs at highest exactitude but remain with an estimate of an *open quotient* instead.

The idea is to use the cycle aligned representation dscribed in section 2.6 and use the *graphic data* as a basis. Instead of detecting the GOI of every cycle, a more global relation can be used by detecting the *edge* that is visible between the zero-excitation of the closed phase and the beginning of the open phase. As suitable algorithm to perform this edge detection has been developed by Canny [22] in 1986.

Figure 2.16 shows the performance of the Canny algorithm. The black line indicates the detected egde. Note that at the end (and in many also at the beginning) there are some spurious deviations that have to be smoothed before further analysis.

In the course our analysis of vocal vibrato we have found the closed phase of a glottal cycle to remain approximately unchanged during vibrato. This is also clearly visible in figure 2.16. In our case we therefore use a median filter to compute an average value of the GOI instant within a series of glottal cycles (e.g. for a segment of 1s of speech). This averaged value can be used to describe as well the *instant open quotient* as the *variation of the open quotient* within the analyzed segment.

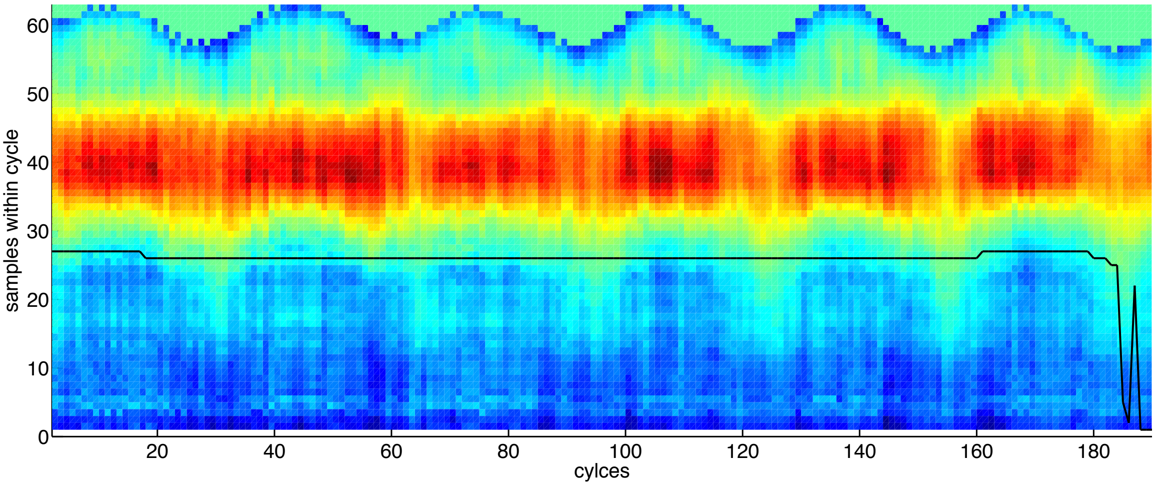


Figure 2.16: Cycle aligned representation and the detected GOI position for a 1s vibrato signal ($f_s = 11025Hz$) through edge detection by the Canny algorithm (solid black line)

2.8 Empirical Data

In the course of this work we have performed a small scale study on the temporal evolution of vibrato. The actual testing task and the environment, in which the test hast been carried out will be described in this section.

2.8.1 Test Specifications

The individual was asked to repeatedly sing the following pattern:

1. hold a pitch as static as possible for 2s,

2. apply vibrato to this pitch and hold for 2s,

3. release the vibrato and hold static pitch again (approx. 1s).

Additionally, the individual was asked to choose three different pitches according to the following rule:

1. p_H: a *high* pitch sung at approx. 80% of the chest voice ambitus,

2. p_M: a *medium* pitch sung at approx. 50% of the chest voice ambitus or pleasant middle range,

3. p_L: a *low* pitch sung at approx. 20% of the chest voice ambitus.

As a third parameter we decided to explore the effect of different vocal tract shapes on vibrato by instructing the individual to sing vibrato on

1. the vowel [a] and
2. the vowel [e].

as these vowels a generally regarded to be sung easily in every register.

2.8.2 Recording Environment And Equipment

All files of the testing corpus have been recorded in a professional studio environment.

Very low reverberation time and early reflection levels have been guaranteed by the acoustic measures available in the recording studio.

The signal chain used to record the test files involved a *Brauner Phantom C* microphone plugged into a *Symmetrix 528E*. A/D conversion was performed by a *Yamaha 01v96* digital mixing console at a sample rate of $44100Hz$ and a word length of 24bit. Note that the voice processor was used only to provide the necessary $48V$ phantom power and the microphone preamplifier - no spectral or dynamic modifications whatsoever have been performed by the Symmetrix and an absolutely clean signal has been sent to the A/D converter. The digital audio was recorded on a DAW (*Mac Pro* with *Logic 9*).

To ensure the validity of the amplitude measurements the individual was instructed to keep a constant distance of approximately $30cm$ between mouth and microphone throughout the entire testing procedure.

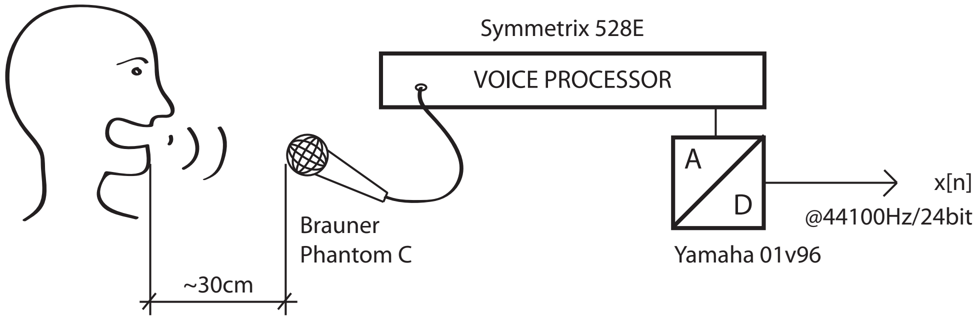


Figure 2.17: Recording setup of the test set (of course the Brauner Phantom C looks significantly different than in the illustration)

2.8.3 Testing Individuals

The test set was recorded by a professional singer that had gone through classical voice education on university level.

2.8.4 Results and Interpretation

For every test file we preformed the following analysis procedures:

1. pitch analysis,
2. amplitude analysis,
3. formant tracking,
4. glottal inverse filtering and
5. open quotient analysis.

Before the actual analysis was carried out the audio data was resampled to a rate of $f_s = 22050Hz$ or $f_s = 11025Hz$ in some cases. Motivation to do so stems from the enormous data load that occurs when performing *samplewise* pitch analysis with the YIN algorithm. Additionally, we can assume that the spectral components of voiced speech above $4 - 5kHz$ are negligible for our analysis tasks and excluding the frequency range above $5kHz$ (as the Nyquist frequency of $f_s = 11025kHz$) can also contribute to a better SNR.

Segmentation of the test files into segments of *static pitch* and segments of *vibrato* has been done by hand-labelling the transitions in the test files.

The resulting data is illustrated by means of the time-aligned and the cycle aligned representation in appendix A.

Time-Aligned Representation

From investigating the time-aligned representation we can draw the following conclusions:

1. *Using only the data of the pitch evolution we can hardly classify the occurrence of vibrato.*
   Comparing the pitch contour in the areas without vibrato with the areas where vibrato is sung we can hardly define where the transitions between the areas are located. The “static” pitch areas expose nearly as strong variations in fundamental frequency as the areas with vibrato do.

The measure that exposes the most significant change at the transitions is the *amplitude* contour. Unfortunately, in the course of this study, the individual was not introduced to keep also the loudness static throughout the recording. Thus, the amplitude increases when vibrato is applied and decreases again at the beginning of the static phase. Besides the increase the amplitude also exposes a *periodic* modulation as constituted by Sundberg.

Especially visible in the case of the `e_mittel.wav` test file we can observe quasi no difference of the pitch contour in or around the vibrato segment. The amplitude evolution instead clearly shows a quasi sinusoidal modulation and an increase in level. Similar behaviour can be seen in `e_tief.wav`, `a_mittel.wav` or `a_tief.wav`.

One little exception has been found in `a_mittle.wav` where at the beinning of the vibrato segment the formants $F3$, $F4$, and $F5$ drop about $200Hz$ and all formants remain unsteady during the vibrato phase. We are not sure if this stems from an actual variation of the VTR or from a measurement error.

Concluding we can say that for classifying vibrato, the modulation of the amplitude, hence the *tremolo*, seems to be an important and powerful feature. This stands in a little bit contrast to Sundberg's assumptions that the influence of the loudness variation is little compared to the perceived pitch variation.

2. *The vocal tract transfer function does not significantly change during vibrato*:
   None of the investigated cases exposed any kind of regular fluctuation of formant frequencies according to the existence and properties of vibrato.

   As already noted above there seems to be an exception but the remaining test have confirmed that there seems to be *no signifcant relation between the existence of vibrato and a variation of the vocal tract transfer function* as Arroabarren had already pointed out in [12].

Cycle-Aligned Representation

From investigating the cycle-aligned representation we can draw the following conclusions:

1. *The open quotient increases with amplitude and the occurrence of vibrato.*
   While the two phases of static pitch expose similar values of the open quotient, the areas of vibrato come with an increase of the open quotient. Unfortunately, we cannot directly relate this factor to the existence of vibrato as we also have to deal with an increase in loudness which can also be responsible for this observation.

Comparing e.g. `a_mittel.wav`, which shows very strong variations of the open quotient to `e_hoch.wav` with only small deviations we discover that also the respective amplitude contours show similar behaviour. Especially `a_mittle.wav` allows for imagining that there is a direct relation between the amplitude an the open quotient.

2. *The length of the closed phase and the first half of the open phase do not change during vibrato*:

   This very interesting observation is illustrated in figure 2.18. It is clearly visible that the amount of pitch period variation (topmost black, sinusoid type line) only occurs in the second half of the glottal open cycle - the "release" phase. The remaining closed phase and the first "attack" part of the open phase do not expose any significant variations in the course of vibrato.

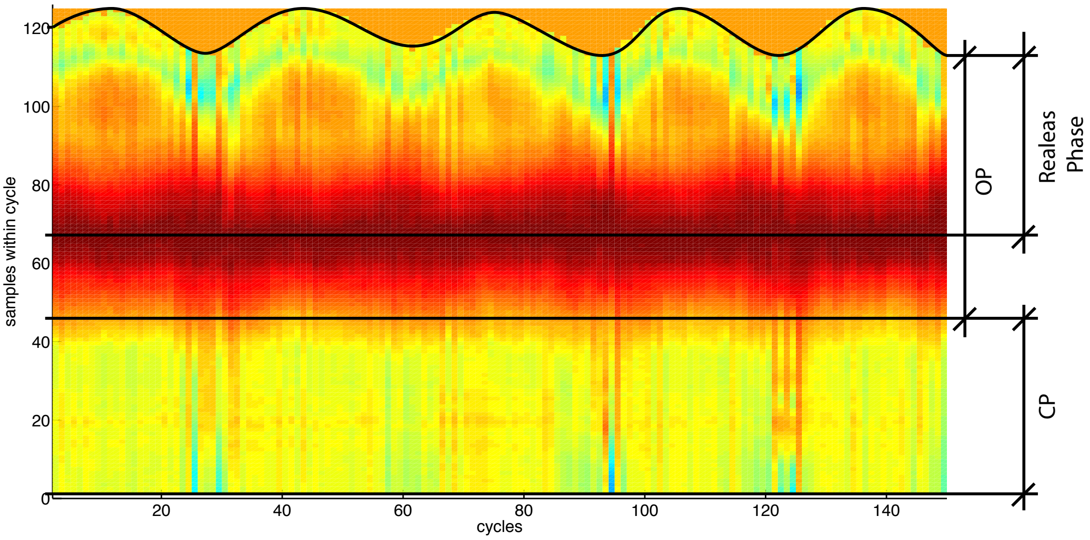


Figure 2.18: Illustration of the relation between duration of the open and the closed phase of a glottal cycle during vibrato; note that the CP and the first, attack-like part of the OP do not alter in length while vibrato. The "release phase" in contrast is responsible for the variation in pitch period; data taken from a 1s baritone recording ($f_s = 22050Hz$)

This knowledge will be exploited later in our approach of actively influencing vibrato by restraining our modifications to this specific time interval. This will enable us to vary vibrato on a physically meaningful basis as it also occurs in the larynx of a singer.

As closing note we want to mention that the test sequence we have developed turned out to be a quite tricky task for the test individual. Obviously, in "natural" singing it is very unusual to perform phrasing of this kind. For future studies it would be reasonable therefore the ensure the “naturalness” in order to establish an optimum environment for the individuals by elaborating the test task together with singers or trainers of voice. It also occurred to us that the measured vibrato could be characterized as “artificial” or “pressed”. This is probably due to the unfamiliar task rather than to the “quality” of the singer.

3 Time-Spectral Processing Techniques

3.1 Introduction

The first question that arises when thinking about modifying the temporal evolution of a sound is in which domain the changes should be applied. In the case of our problem the spectral behaviour of the signal is directly related to the time signal. More specific, the partials of the singing voice exhibit (quasi) the same variation in frequency and amplitude as the fundamental (for further information regarding the behaviour of partials in a singing voice vibrato and how to measure them, please refer to [15] or [23]).

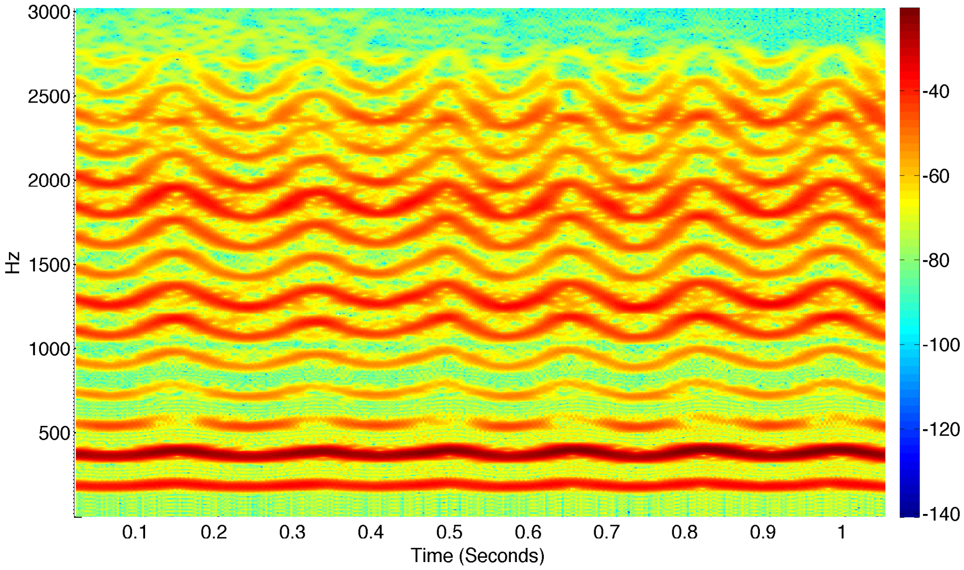


Figure 3.1: Spectrogram of a short segment of vibrato sung by a baritone

Taking a look at figure 3.1 one can see the spectrogram of a sustained vowel sung by a professional baritone. We can clearly identify the fundamental frequency that changes over time and the simultaneous variation of the first 13 partials. If we consider the voice signal $s[n]$ to be a weighted combination of M harmonics

$$s[n] = \sum_{m=1}^{M} c_m e^{j2\pi m f_0 n} \tag{3.1}$$

where the integer m denotes the m-th of M partials of the fundamental frequency f_0 and c_m corresponds to respective *complex* weighting factor, we can relate the variation of the m-th partial directly to the change of the fundamental as

$$\Delta f_m = m\Delta f_0 \tag{3.2}$$

The weighting factor w_m contains information on amplitude and phase relation of the partials

$$c_m = |c_m|e^{j\angle c_m} \tag{3.3}$$

This formulation yields two major consequences:

1. perfect periodicity and

2. infinity of the time series.

Both is *not* the case for speech but we can assume that human speech is *short-time stationary.* This allows us to use short-time analysis methods as described e.g. in [1] p. 257ff. Thereby, overlapping windows of a certain length N are regarded as segments of an infinite signal that has a fundamental period of the window length N.

We are therefore allowed to use the formulation in eq. 3.1 within *a single analysis frame* as a valid representation of the physical process.

Also, we can use this to approximate the discrete Fourier transform $S_r(k)$ of $s[n]$ at an arbitrary time step r with window length of N as

$$S_r(k) = \sum_{k=0}^{N-1} \sum_{m=1}^{M} 2\pi\delta(N - c_m m f_0) \tag{3.4}$$

To point our the independence of source and weighting we can separate the weighting factor from the harmonic signal like before in eq. 1.3 as

$$\begin{aligned} S_r(k) &= C_r(k)\Phi_r(k) && (3.5) \\ &= H_r(k)\underbrace{W_r(k)\Phi_r(k)}_{U_r(k)} && (3.6) \end{aligned}$$

where $\Phi(k)$ denotes the discrete Fourier transform of the harmonic signal with *uniform gain and phase* for all M partials

$$\Phi_r(k) = \sum_{k=0}^{N-1} \sum_{m=1}^{M} 2\pi\delta(N - m f_0) \tag{3.7}$$

and $W_r(k)$ equals the Fourier transform of the frequency dependent weighting factor of the glottal source $u[n]$ that has the respective Fourier transform $U_r(k)$ and decreases with approx. 12dB per partial [4]. The vocal tract transfer function is denoted as $H_r(k)$ here.

Note that at this point we have a complete formulation of the voice production model but, again, we have no certainty regarding the combination $H_r(k)W_r(k)$. By using the restriction of the glottal closed phaser to the analysis interval we can force the weighting term to become $W_r(k) = 1, \forall k$. This will allow for calculating an accurate estimate of the vocal tract filter and using this data to perform *glottal inverse filtering* (section 3.3) by using a time-variant linear filter technique.

3.2 Overlap-And-Add

One algorithm to perform these operational steps in the time domain is better known as *Overlap-And-Add* (OLA): a signal $x[n]$ gets segmented and windowed by r overlapping windows $w[n]$ of fixed length N. In the second step these windowed blocks are summed up to the synthesized signal $y[n]$. This, under certain circumstances, allows for *perfect reconstruction* of the original signal.

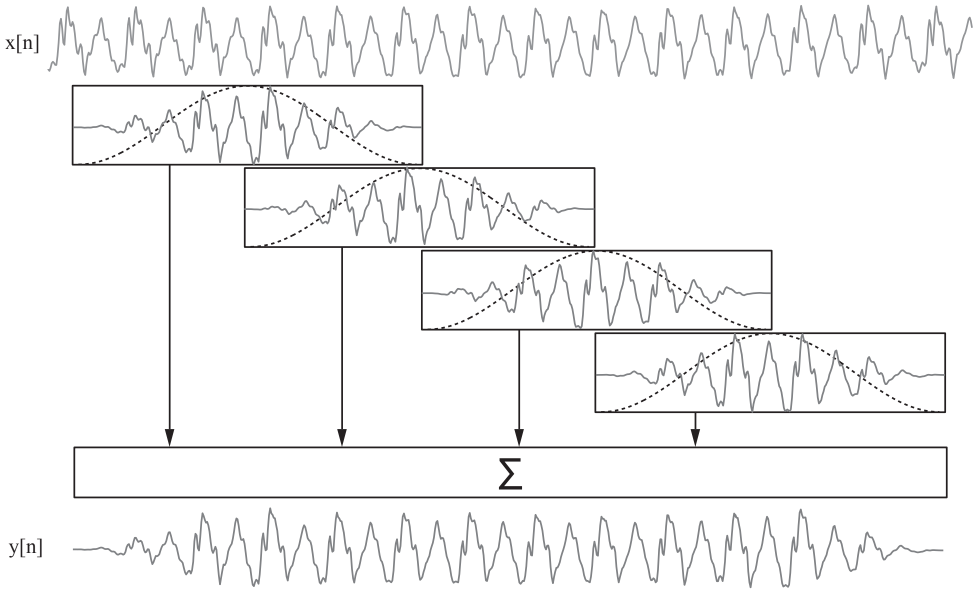


Figure 3.2: Rough schematic of the Overlap-and-Add method

In a mathematical formulation one would define

$$y[n] = \sum_{r=-\infty}^{\infty} y_r[n] \tag{3.8}$$

where r is an integer and $y_r[n]$ denotes a single causal window of length N defined as

$$y_r[n] = x[n]w[rN - n] \tag{3.9}$$

which leads us to

$$y[n] = \sum_{r=-\infty}^{\infty} y_r[n] = x[n] \left(\sum_{r=-\infty}^{\infty} w[rN - n] \right) = x[n]\tilde{w}[n] \tag{3.10}$$

As a direct consequence of eq. 3.10 we can derive the *condition for perfect reconstruction*:

$$\tilde{w}[n] = \sum_{r=-\infty}^{\infty} w[rN - n] = C \tag{3.11}$$

where C denotes the *reconstruction gain.*

Note that the reconstruction gain C is *not* a function of time n and thus constant for all values of n. There is only a limited amount of suitable window types for this task. We therefore have to choose the form of our window and the amount of overlap very carefully.

Using an energy conserving window like the *Hann* window and an overlap of 50% is one of the possible combinations that allow for perfect resynthesis.

3.2.1 Possibilities of OLA

This technique is one of the most basic block signal processing tools. In the case described before the original signal and the synthesized signal had to be the same $x[n] = y[n]$. The real powers of this method stem from an intermediate *processing step* as displayed in figure 3.3.

It allows for performing operations in the time domain *and* in the frequency domain simultaneously on signal excerpts and recombining those modified segments.

Note that the filters applied to every single frame are still *LTI* but the the slowly varying vocal tract transfer function allows for *changing* the filter response - hence the filter parameters - and the overlapped signal will still be phase coherent in most cases.

In our case we can use this step to remove the influence of the prior computed the time variant vocal tract response from the speech signal unveiling the glottal source signal as a consequence.

Regardless of the promising idea of OLA there are some serious problems when recombining *altered* signal segments lengths that basically stem from phase misalignments. Therefore, special restrictions have to be applied to gain more accurate results. These measures will be described in a later section.

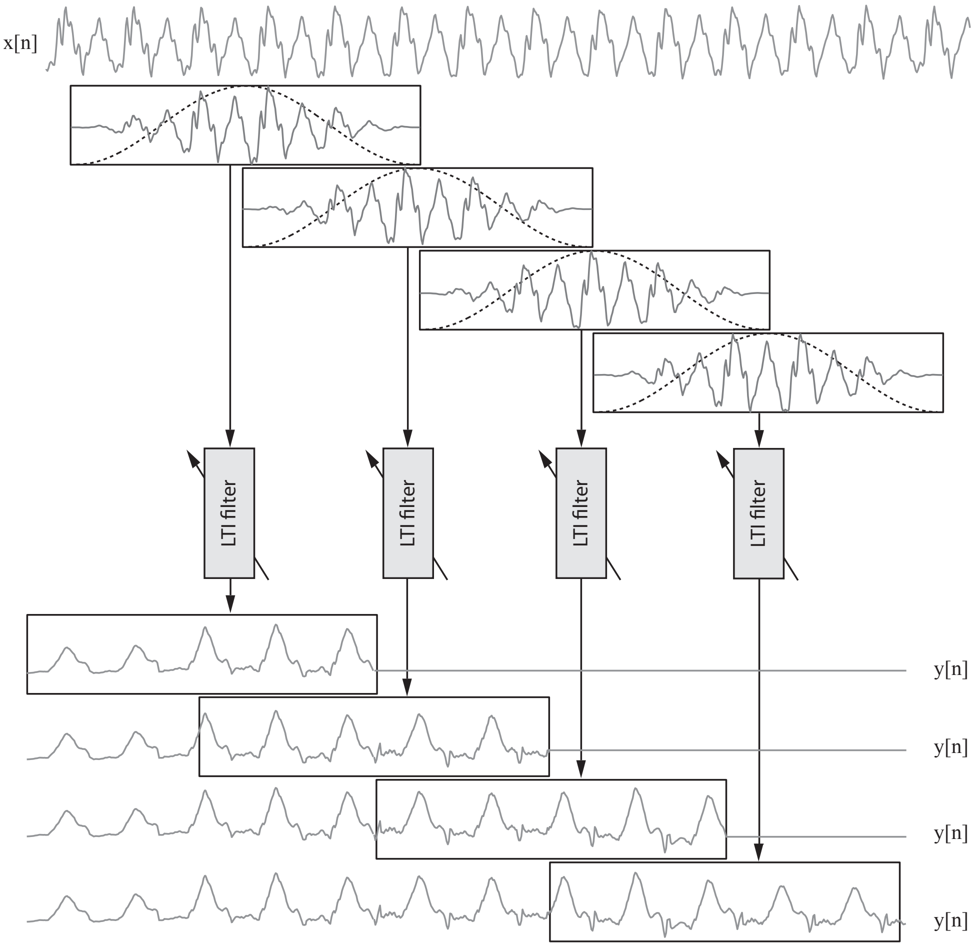


Figure 3.3: Rough schematic of the Overlap-and-Add method used for implementation of an LTV filter; the filter block (grey) changes only from block to block; for the duration of a single block it can be assumed to be linear time-*invariant* (LTI) though

3.3 Glottal Inverse Filtering

If we decided to use a straight-froward approach to alter the fundamental frequency in the time or frequency domain by e.g. resampling and not taking into account the special properties of the glottal source, we would also inherently change the overall transfer function $W(z)$ and as a consequence the position of the vocal tract resonances. This would lead to an unnatural sound or even al loss of comprehensibility. To avoid this, we have to use a method that conserves the vocal tract resonances while allowing for changes in the time domain.

To circumvent this problem, several well known algorithms have been proposed in history. Generally, *formant conserving pitch shifting* is an approach that takes care about the fact of "spectral consequences" of temporal modifications. Those algorithms try to compute an estimate of the vocal tract transfer function and conserve it while the modifications in the time domain are applied. This is commonly done by using some kind of auto-regressive modelling (AR) to perform blind deconvolution of the glottal source and the vocal tract response. One special case of AR is *linear predictive coding* which plays a major role in our algorithm (a detailled description follows in section 3.4).

3.3.1 Motivation

We want to take this a step further by maximizing the accuracy of the VTR estimate. This, in consequence, allows for more exactly deconvolving the the system response from the speech signal wich yields the glottal source signal as a result. By accurately estimating the glottal source as a consequence, we can comput achieve a maximum flexibility when applying our modifications. To briefly recap from chapter 2, figure 3.4 illustrates the parameters an actual singer modifies while singing a sustained vowel.

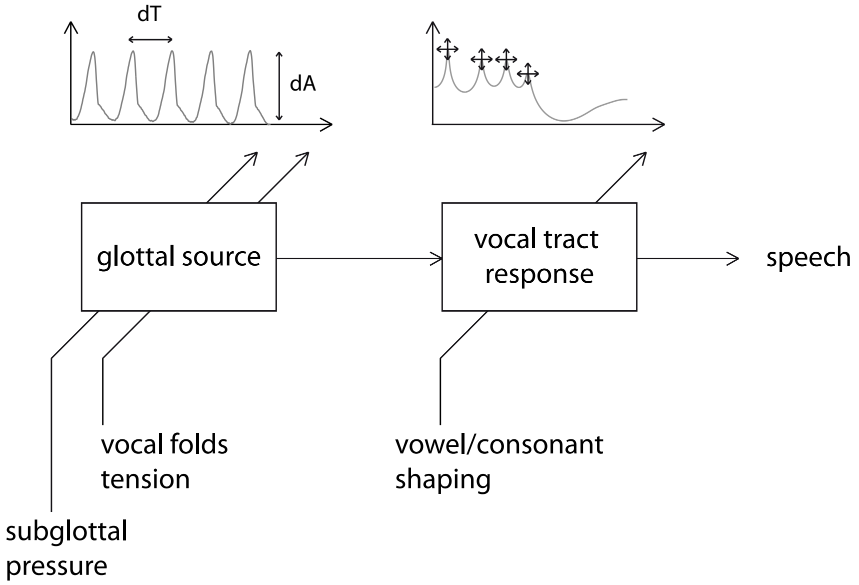


Figure 3.4: Parameters of the voice production model; temporal and spectral shape of the glottal source are primarily defined by subglottal air pressure from the lungs and the vocal folds tension; vowel/consonant shaping results in a set of formant resonances with corresponding center frequencies and gains

Though, of course, this is a quite complex task, the set of parameters can be reduced to the following basic controls:

1. *subglottal pressure*: is related to the amplitude of the glottal source
2. *tension of the vocal folds*: directly related to the pitch of voiced sounds and primary control parameter to discriminate voiced/unvoiced speech and normal/whispered speech
3. *vowel/consonant shaping*: defines the position of formants

By accurately estimating the glottal source, we inherently gain exact knowledge on the VTR and hence the capability of arbitrarily and actively changing every single parameter.

Obviously, exact inverse filtering is a very delicate task and various authors have proposed their methods to perform inverse filtering. Arroabarren et al. [24] have evaluated the performance of three notable algorithms namely

- Analysis By Synthesis [25],
- Glottal Spectrum-Based Inverse Filtering [26] and
- Closed Phase Covariance Inverse Filtering [27]

of which latter outperformed the others in many issues.

Naturally, also this approach is not perfect as is has strong limitations that come with higher pitched voices. As Arroabarren pointed out, the closed phase of a voiced segment tends to become shorter the higher the pitch becomes. This, as a consequence, leads to shortened analysis intervals and - in the worst case - no analysis frame at all.

Nevertheless, the remaining performance aspects motivated us to employ this method in our algorithm.

Another reason to use closed phase analysis was described by various authors: *subglottal coupling* [28], [6], [29] occurs primarily in the open phase where the *subglottal cavity* becomes an additional resonating volume and results in a variation of the *first* formant frequency. To minimize the influence of glottal coupling the analysis interval is restrained to the glottal closed phase. Though also in this phase there may occur a certain amount coupling through the glottal tissue, the influence can be regarded as significantly less [24].

3.4 Linear Predictive Coding

Linear predictive analysis has evolved to one of the most powerful and used techniques in speech processing and communication. It's capability to exactly estimate various parameters of the discrete-time model of speech such as pitch, formants, short-time spectra, etc. as well as regarding the possibilities of effective and computationally cheap speech coding for transmission have made LPC analysis probably the most widely spread method.

The basic concept of linear prediction is directly related to the source/filter model of speech as it allows for estimating a system response by linear combination of speech samples in the past. There have been elaborated various formulations of the linear prediction idea (many of them are equivalent or at least closely related to each other).

This section is intended to give a brief overview on the basic principles and performance of four different formulations or rather implementations of linear prediction. For further detail please refer to literature e.g. [1] p. 487ff.

3.4.1 Basic Principles

As we have pointed out in the past sections, the z-transform of the speech signal $s[n]$ is a combination of the VTR and the spectral shape of the glottal pulse train.

$$S(z) = U(z)H(z) \tag{3.12}$$

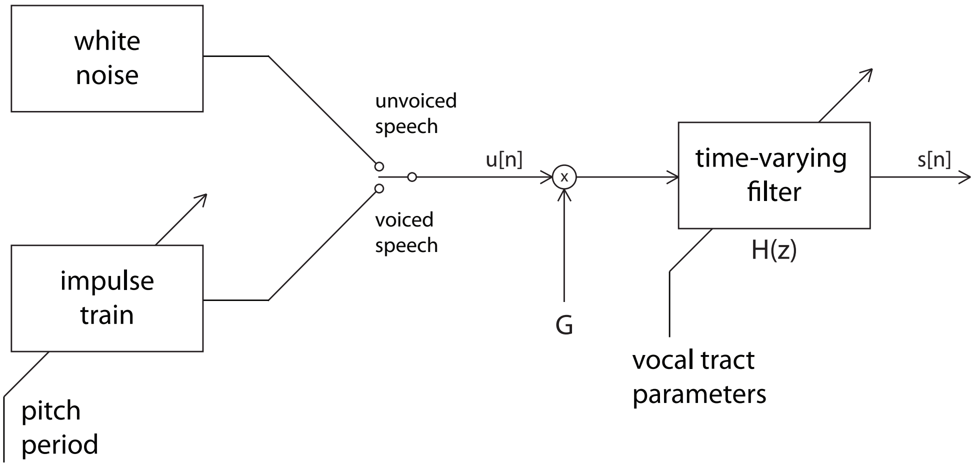


Figure 3.5: Simplified model of speech production

or by introducing a gain G as depicted in figure 3.5 we could define the VTR $H(z)$ as

$$H(z) = \frac{S(z)}{GU(z)} = \frac{S(z)}{E(z)} = \frac{1}{1 - \sum\limits_{k=1}^{p} a_k z^{-k}} \tag{3.13}$$

Later we will see that the influence of the glottal source can be minimized and approximated as $U(z) = 1$ by restraining the analysis frame to the closed phase as mentioned before in section 1.3.1. For now let us stick to this general formulation to derive the further steps.

As illustated in figure 3.5 the model makes use of the following parameters:

- Excitation parameters:
  - voiced/unvoiced classification
  - pitch period
  - gain parameter G
- Vocal tract parameters
  - coefficients of the order p all-pole filter

Classifying a sound in voiced or unvoiced is subject to numerous studies. For a good assessment please refer to e.g. [1], chapter 10. By focussing our work on singing voice vibrato we can assume the sound to be voiced and hence periodic. We therefore decided not to include voiced/unvoiced classification in this work.

As reviewed in chapter 1 the all-pole model of the vocal tract has certain limitations i.e. regarding nasalized sounds. The nasal cavities act as *anti*-resonances, hence, as *zeros* in a transfer function. Calculation of pole-zero transfer functions is a very much more complicated issue though. Works like Kang et al. [5] use an extension of the lossless tube model (the nasal cavities resemble an additional third branch) to simulate the entire vocal tract also for nasalized sounds. Direct estimation of the transfer function parameters is a non linear problem. An assessment on different modelling techniques has been carried out by Walker and Murphy in 2007 [30].

For an all-pole system as provided in figure 3.5 we can define the *wide-sense stationary* (WSS) speech signal $s[n]$ as a sum of p past samples weighted by the filter coefficients

$\{a_k\}$ of the transfer function $H(z)$ as a difference equation

$$s[n] = \underbrace{Gu[n]}_{\text{excitation}} + \underbrace{\sum_{k=1}^{p} a_k s[n-k]}_{\text{IIR system response}} \tag{3.14}$$

which can be seen as the inverse z-transform of

$$S(z) = GU(z)H(z) \tag{3.15}$$

A p-th order estimator of the speech signal $\tilde{s}[n]$ can be defined as

$$\tilde{s}[n] = \sum_{k=1}^{p} \alpha_k s[n-k] \tag{3.16}$$

where α_k denotes the k-th prediction coefficient and its system function can be written as

$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} \tag{3.17}$$

Now we can define the *prediction error* $e[n]$ as

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \tag{3.18}$$

which leads to

$$E(z) = S(z) - P(z) \tag{3.19}$$

If the speech signal $s[n]$ is valid according to definition 3.14 and the predictor has the optimum coefficients $\{\alpha_{k,opt}\} = \{a_k\}$ then $e[n]$ becomes

$$e[n] = s[n] - \tilde{s}[n] = \underbrace{Gu[n] + \sum_{k=1}^{p} a_k s[n-k]}_{s[n]} - \underbrace{\sum_{k=1}^{p} \alpha_{k,opt} s[n-k]}_{\tilde{s}[n]} = Gu[n] \tag{3.20}$$

Combining these relations as

$$\begin{aligned} E(z) &= \frac{S(z)}{H(z)} && (3.21)\\ \frac{E(z)}{S(z)} &= 1 - P(z) = \frac{1}{H(z)} && (3.22)\\ A(z) = \frac{E(z)}{S(z)} &= \frac{GU(z)}{S(z)} = \frac{1}{H(z)} && (3.23) \end{aligned}$$

defines the *prediction error filter* $A(z)$ as the *inverse filter* of the vocal tract response $H(z)$.

In other words, the result of optimal prediction is an all-zero FIR filter that is the inverse of the vocal tract all-pole IIR filter.

3.4.2 Solving the LPC Equations

To perform *optimal* prediction we need an optimization criterion. As in many other scenarios, we define a *minimum means squared error* (MMSE) as measure for optimum filter approximation.

$$\varepsilon = \mathsf{E}\left\{|e[n]|^2\right\} \tag{3.24}$$

As we are dealing with *ergodic processes* we can compute the expected value as

$$\varepsilon_r = \frac{1}{N} \sum_{n=r}^{N+r-1} |e[n]|^2 \tag{3.25}$$

of an N sample long window starting at the time index r. So if we substitute here with eq. 3.18 we get

$$\begin{aligned} \varepsilon_r &= \sum_m |s_r[m] - \tilde{s}_r[m]|^2 && (3.26)\\ &= \sum_m \left| s_r[n] - \sum_{k=1}^{p} \alpha_k s_r[m-k] \right|^2 && (3.27)\\ &= \sum_m \left[s_r^2[n] - 2\left(s_r[n] \sum_{k=1}^{p} \alpha_k s_r[m-k] \right) + \left(\sum_{k=1}^{p} \alpha_k s_r[m-k] \right)^2 \right] && (3.28) \end{aligned}$$

where $s_r[m]$ is a segment of speech of (for the moment) unspecified length.

To find the optimal coefficients $\{\alpha_{k,opt}\}$ we have to set

$$\frac{\partial \varepsilon_r}{\partial \alpha_i} = 0 \tag{3.29}$$

for $i = 1, 2, \ldots, p$. To simplify the derivative we can rewrite eq. 3.28 in matrix form as

$$\begin{aligned}
\varepsilon(\boldsymbol{\alpha}) &= \mathsf{E}\left\{(s[n] - \tilde{s}[n])^2\right\} = \mathsf{E}\left\{(s[n] - \boldsymbol{\alpha}^T \mathbf{s}[n])^2\right\} && (3.30)\\
&= \mathsf{E}\left\{s^2[n] - 2s[n]\boldsymbol{\alpha}^T \mathbf{s}[n] + \boldsymbol{\alpha}^T \mathbf{s}[n]\mathbf{s}^T[n]\boldsymbol{\alpha}\right\} && (3.31)\\
&= \mathsf{E}\left\{s^2[n]\right\} - 2\boldsymbol{\alpha}^T s[n]\mathbf{s}[n] + \boldsymbol{\alpha}^T R_{xx}\boldsymbol{\alpha} && (3.32)\\
\nabla_{\boldsymbol{\alpha}}\varepsilon(\boldsymbol{\alpha}) = 0 &= 0 - 2\mathbf{p}[n] + 2R_{xx}\boldsymbol{\alpha}_{opt} && (3.33)
\end{aligned}$$

and as a consequence the *normal equations* can be written in the form of a *Wiener-Hopf solution* as

$$\boldsymbol{\alpha}_{opt} = R_{xx}^{-1}\mathbf{p}[n] \tag{3.34}$$

where the crosscorrelation vector in this special case is defined as $\mathbf{p}[n] = s[n]\mathbf{s}[n]$ and therefore corresponds to the autocorrelation $\mathbf{r}[r]$. Hence, we write

$$\boldsymbol{\alpha}_{opt} = R_{xx}^{-1}\mathbf{r}[n] \tag{3.35}$$

In literature there are two commonly used variations of the formulation that differ only in the definition of the signal vector $\mathbf{s}_r[n]$. In the

Autocorrelation Method

we assume the signal $s[n]$ to be equal 0 outside a finit interval of the waveform segment of length $0 \leq m \leq L-1$. Thus, we can define

$$s_r[m] = s[m+r]w[m] \tag{3.36}$$

where $w[m]$ is a finit length window (e.g. Hamming, Hann or rectangular) that is zero outside the priorly mentioned interval. Applying these definitions to eq. 3.35 leads to an $p \times p$ positive-definite and symmetric *autocorrelation matrix* R_{xx} that exposes *Toeplitz structure*. Therefore, the equations can be efficiently solved by the *Levinson-Durbin recursion* [31].

In the

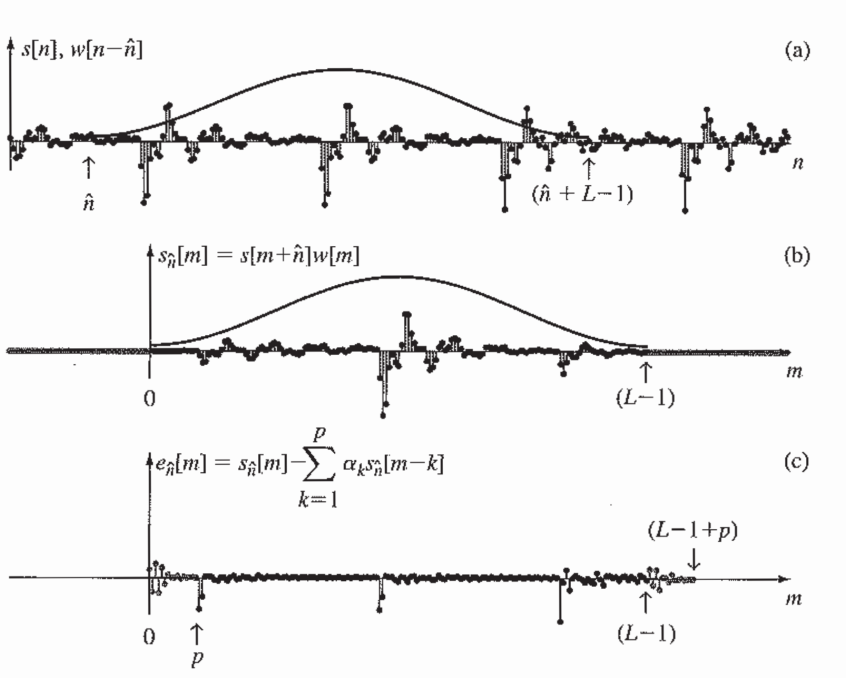


Figure 3.6: Windowing in the autocorrelation method (note that the index $\hat{n}$ corresponds to r in this work); (a) speech signal $s[n]$ and Hamming window positioned at time $n = \hat{n}$; (b) windowed segment $s_{\hat{n}}[m]$; (c) segment of prediction error signal, $e_{\hat{n}}[m]$, over the range $0 \leq m \leq L-1+p$ obtained using an optimum predictor with $p = 15$ ([1] p.494)

Covariance Method

the definition of the analysis interval is slightly different. To circumvent edge effects as described in [1] chapter 9 additional p (the order of the predictor) samples before the beginning of the analysis frame are required. Thus, the interval for the analysis frame $\mathbf{s}_r$ is defined as $-p \leq m \leq L-1$ where *no edge smoothing windowing* is necessary:

$$s_r[m] = s[m+r-p] \tag{3.37}$$

Again applying these definitions to eq. 3.35 the resulting matrix R_{xx} is no more an autocorrelation matrix but rather a *covariance matrix* Φ_{xx}. It is symmetric and positive-definite but *not* Toeplitz. Therefore, solving the set of equations by the Levinson recursion is not possible but instead the *Cholesky decomposition* may be used (see [1] chapter 9.5.1 for more detail).

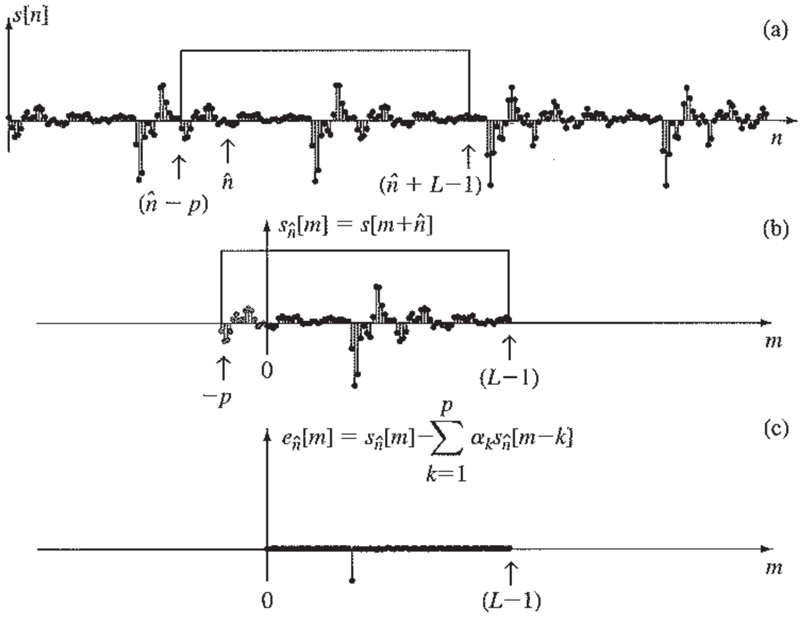


Figure 3.7: Windowing in the covariance method (note that the index $\hat{n}$ corresponds to r in this work); (a) speech signal $s[n]$ and rectangular windows positioned over the range $\hat{n} - p \leq n \leq \hat{n} + L - 1$; (b) windowed segment, $s_{\hat{n}}[m]$, defined over a range $-p \leq m \leq L - 1$; (c) segment of prediction error signal $e_{\hat{n}}[m]$, defined over the range $0 \leq m \leq L - 1$ using an optimum predictor with $p = 15$ ([1] p.498)

Summary

The *autocorrelation method* requires an L sample *windowed* waveform segment that is assumed to be zero outside the window. It can guarantee stability for all-pole filters [32] but the windowing introduces a decrease in spectral resolution. For finite length analysis frames even stability can not be granted in some cases [33].

The *covariance method* does not take any assumptions for the signal outside the analysis frame, hence no smoothing windowing is required, but it has to be provided p preceding samples in order to be consistent with the correlation boundaries. Stability also can not be assured by the covariance method but it comes with a higher spectral resolution.

3.4.3 The Lattice Formulation: An Alternative Representation

Itakura [34] in 1974 and independently Burg [35] in 1975 have developed an alternative formulation of the linear prediction problem. While the two methods described in section

3.4 use two separate steps for the computation of the correlation values and solving the resulting normal equations, the proposed *lattice method* is capable of performing both computations in one recursively operated step. The necessary derivations stem from the Levinson recursion in which at the i-th iteration a *forward prediction error* is computed as

$$e^{(i)}[m] = s[m] - \sum_{k=1}^{i} \alpha_k^{(i)} s[m-k] \tag{3.38}$$

In terms of a z-transform we can write

$$E^{(i)}(z) = A^{(i)}(z) S^{(i)}(z) = \left(1 - \sum_{k=1}^{i} \alpha_k^{(i)} z^{-k}\right) S(z) \tag{3.39}$$

By substituting the *update rules* of the Levinson-Durbin algorithm ([1] p.528) we get the z-transform of the *backward prediction error*

$$B^{(i)}(z) = z^{-i} A^{(i)}(z^{-1}) S(z) \tag{3.40}$$

and the inverse z-transform gives us

$$b^{(i)}[m] = s[m-i] - \sum_{k=1}^{i} \alpha_k^{(i)} s[m+k-i] \tag{3.41}$$

This in consequence allows for interpreting eq. 3.39 as combination of $E^{(i-1)}(z)$ and $B^{(i-1)}(z)$ as

$$E^{(i)}(z) = E^{(i-1)}(z) - k_i z^{-1} B^{(i-1)}(z) \tag{3.42}$$

where k_i is often referred to as *PARCOR* or *partial correlation* or *reflection coefficient* at the i-th iteration. It is a direct consequence of the formulation of the Levinson-Durbin algorithm.

The knowledge of these relations and the recursive nature of the Levinson-Durbin algorithm allow for constructing the two basic filter structures in in a recursive manner.

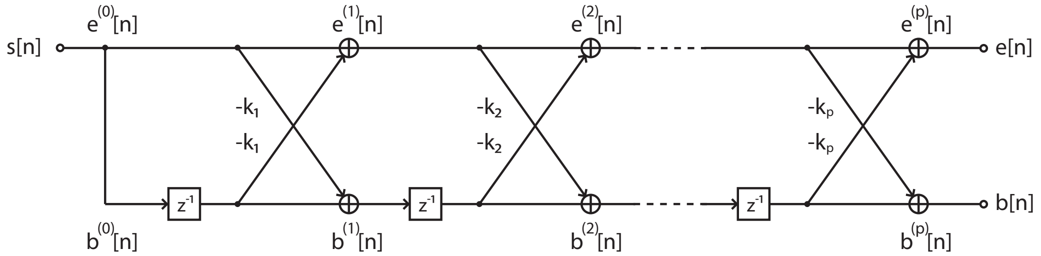


Figure 3.8: Signal flow graph of an FIR lattice filter to calculate the *forward* prediction error $e[n]$

FIR Lattice Network

$$\begin{array}{rcll} e^{(0)}[n] & = & b^{(0)}[n] = s[n] & 0 \le n \le L-1 \\ e^{(i)}[n] & = & e^{(i-1)}[n] - k_i b^{(i-1)}[n-1], & 1 \le i \le p, \\ & & & 0 \le n \le L-1+i, \\ b^{(i)}[n] & = & b^{(i-1)}[n-1] - k_i e^{(i-1)}[n], & 1 \le i \le p, \\ & & & 0 \le n \le L-1+i, \\ e[n] & = & e^{(p)}[n], & 0 \le n \le L-1+p \end{array} \qquad (3.43)$$

IIR Lattice Network

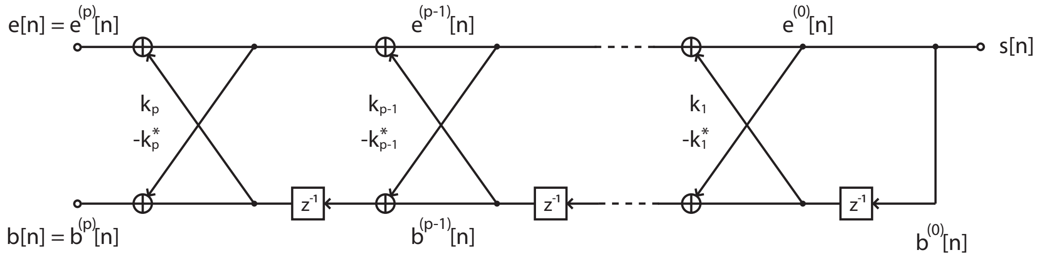


Figure 3.9: Signal flow graph of an IIR lattice filter to calculate the speech signal $s[n]$

$$\begin{array}{rcll} e^{(p)}[n] & = & e[n], & 0 \le n \le L-1+p, \\ e^{(i-1)}[n] & = & e^{(i)}[n] - k_i b^{(i-1)}[n-1] & i = p, p-1, \ldots 1, \\ & & & 0 \le n \le L-1+i-1, \\ b^{(i)}[n] & = & b^{(i-1)}[n-1] - k_i e^{(i-1)}[n], & i = p, p-1, \ldots 1, \\ & & & 0 \le n \le L-1+i, \\ s[n] & = & e^{(0)}[n] = b^{(0)}[n] & 0 \le n \le L-1 \end{array} \qquad (3.44)$$

Direct Computation of the k_i Parameters

The obvious and straight forward method of computing the parameters k_i is by employing the Levinson-Durbin algorithm or rather *PARCOR Lattice Algorithm*. An alternative formulation has been proposed by Burg an is known as the *Burg Method*.

While the PARCOR method minimizes only the forward prediction error, the Burg method minimizes the *sum* of forward and backward error.

A study on the performance of different methods to calculate the parameters has been published by Makhoul in 1977 [36] and subsequently [37] where he also proposed a novel *covariance lattice* method that we employed in our *LTV lattice filter* implementation (see section 3.5.3).

For a more detailled description of the lattice formulation and its computational aspects, powers, drawbacks and capabilities please refer to e.g. [1], [32] [36] or [38].

3.5 Synthesis Techniques

In the following section we are going to briefly present the two synthesis methods we use in this work to compile the glottal source waveform after inverse filtering as an intermediate step for modification and for resynthesizing the final signal containing the applied variations.

3.5.1 Pitch-Synchronous Overlap-And-Add (PSOLA)

The most common method in block signal processing is *OLA*. As pointed out prior and more intensively discussed by e.g. Bonada et al. [39] using arbitrary block sizes and positioning of the frames results in audible spectral and temporal distortions. More precisely, using a window size that involves *more than one* glottal excitation leads to “doubled phase alignments” and as a result to an “undesired roughness characteristic” [39].

Therefore, the idea of synchronizing OLA to the pitch, thus the name *Pitch-Synchronous OLA*, involves

1. restraining the analysis/synthesis interval to the length of a single glottal cycle and
2. centering the window around the glottal excitation.

The crux in performing PSOLA lies normally within point two: detecting the so-called *pitch marks* that indicate the instant of major excitation is not a trivial task. In our case though, we may use the instants of glottal closure (GCI) estimated for closed phase LPC as pitch marks as they correspond to the instant of major excitation.

3.5.2 Asymmetric Pitch-Synchronous Overlap-And-Add (A-PSOLA)

A small refinement to PSOLA we have come up with in the course of this work deals with the problem that arises when concerning speech signals with *altering* pitch as is the case for vibrato.

Amplitude conserving windows like the *Hann* window have to be used with a specific overlap. Otherwise, the total energy of the signal cannot be conserved. This would exactly be the case if we applied a fixed length Hann window to our pitch changing singing signal. Therefore, using an *adaptive window size* but still keeping the exact centering of the frame according to the pitch mark brought us to the *Hybrid Hann Window.*

Hybrid Hann Windows

Generally, the Hann window is defined as

$$w[m] = 0.5\left(1 - \cos\frac{2\pi m}{L}\right) \tag{3.45}$$

for $0 \leq m \leq L$ where L denotes the window length. The length of a windows has to be *odd* to ensure that the center value has the maximum weight of 1.

Now the idea is to combine two windows of different window length to a single window like illustrated in figure 3.10. Thereby, the total window length L remains unchanged and only position of center within the frame determines the window lengths L_1 and L_2 that are used to compute the left and right slope.

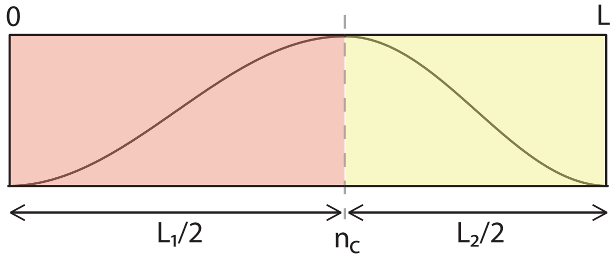


Figure 3.10: Shape of a hybrid Hann window $w[n]$; n_c indicates the center sample of the window ($w[n_c] = 1$)

The hybrid window can be denoted as

$$w_{hy}[m] = \begin{cases} 0.5\left(1 - \cos\frac{2\pi m}{L_1}\right), & 0 \leq m \leq n_c, \\ 0.5\left(1 - \cos\frac{2\pi m}{L_2}\right), & n_c < m \leq L \end{cases} \tag{3.46}$$

where the window lengths L_1 and L_2 are defined as

$$L_1 = 2n_c \tag{3.47}$$

$$L_2 = 2(L - n_c) - 1 \tag{3.48}$$

Letting a series of such windows overlap in a manner as displayed (a little exaggerated) in figure 3.11 allows for using (PS)OLA techniques for non uniform pitch signals. The only restriction for this approach is that every succeeding frame inherits its *left half* window length L_1 from its predecessor's *right half* window. Thus

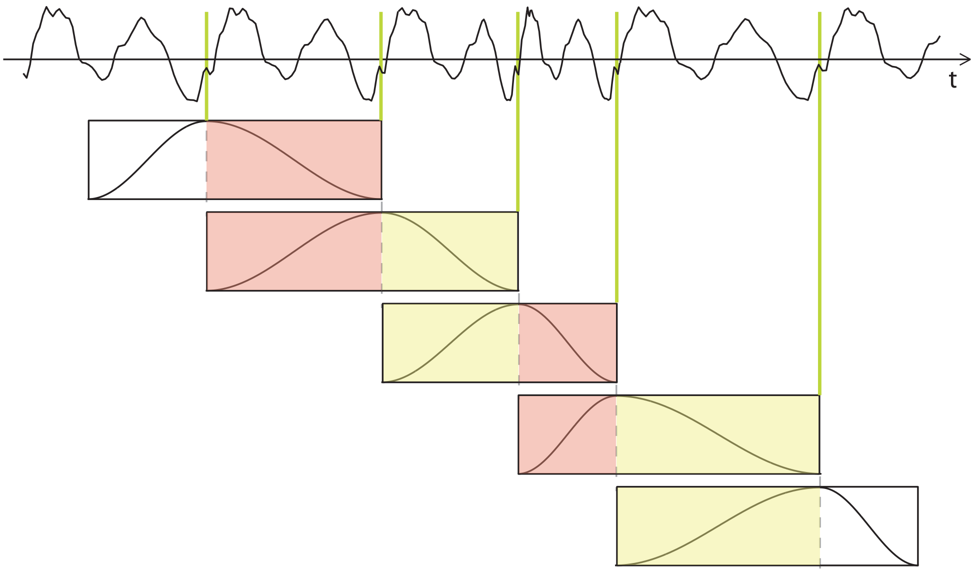


Figure 3.11: Overlapping of various non uniform length hybrid Hann windows

$$L_1^{(i)} = L_2^{(i-1)} \tag{3.49}$$

where the integer i indicates the i-th frame of analysis. By summing the window series itself we can proof that the amplitude is conserved for all windows except the very first and the very last: the gain can be assumed to be equal to one.

$$\sum_{i=1}^{J} w^{(i)}[n] = 1 \qquad \forall n \tag{3.50}$$

where J denotes the total number of analysis frames and every window $w^{(i)}[n]$ is defined for the interval $[m_0^{(i)}, m_0^{(i)} + L^{(i)}]$.

Influence of the Hybrid Window on the Magnitude Response

Obviously, manipulating the shape of the temporal evolution of a Hann window will come with a certain change in its magnitude response. To ensure, this change does not significantly influence the measurement we carried out a small scale simulation to illustrate the possible effects.

The simulation involved a series of different hybrid windows. All of them were of length $L = 301$ and the position of the center n_c was modified by ±100 samples. Figure 3.12 shows the log-magnitude response of the entire test set (the simulated sampling rate was chosen as $f_s = 11025Hz$). Though the Hann window itself exposes a relatively strong

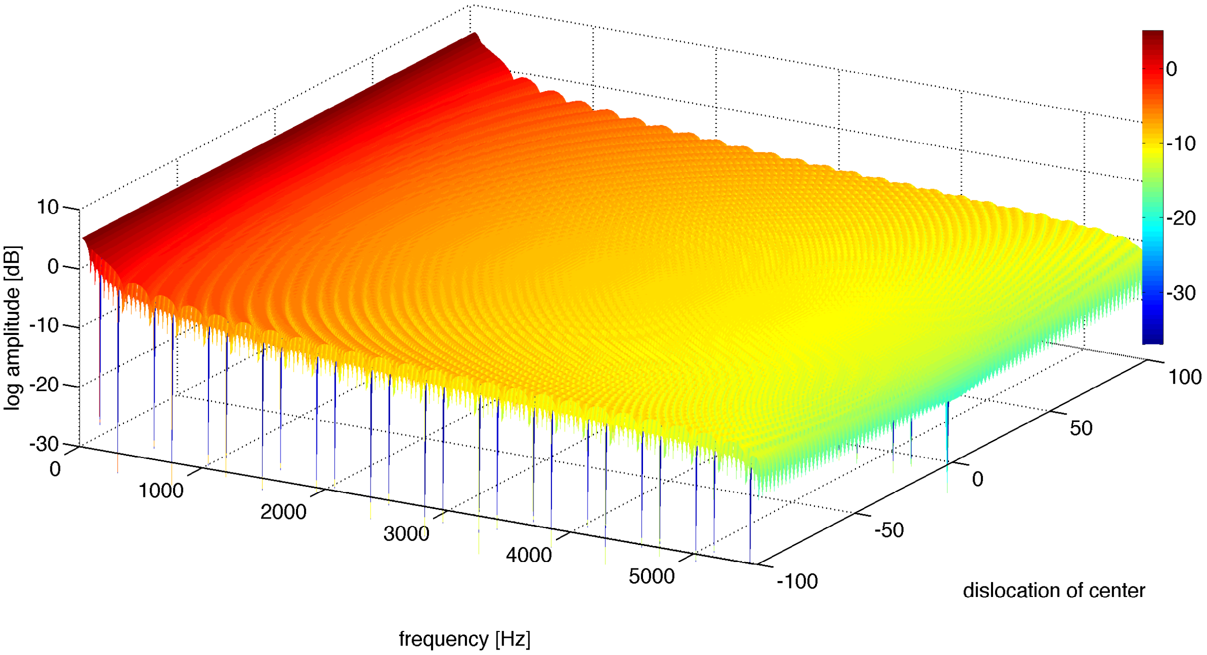


Figure 3.12: Log-magnitude response of a hybrid Hann window ($L = 301$, $f_s = 11025Hz$) with varying position of center ($y = 0$ window center lies at sample 150; $y = -50$ window center lies at sample 100, etc.)

non uniform frequency response, the influence of the asymmetry seems to be relatively small. There is a visible movement in the ripple but nevertheless the entire surface can be seen as approximately flat. We therefore assume that we do not introduce significant additional distortions to our analysis system by using hybrid windows.

3.5.3 Implementation of an LTV Lattice filter

An approach to circumvent the problems that arise with the use of PSOLA techniques for speech signal processing is replacing the classic FIR/IIR filter with its Lattice representation. The definition of the Lattice is directly related to the *lossless tube model of speech* as described in section 1.4. In contrast to direct form implementations, Lattice filters allow for interpolating their filter coefficients while still guaranteeing overall filter stability. Therefore a continuous variation (though of course in the discrete time domain) of the reflection coefficients k_i at every time step n can reasonably model the sound propagation in the vocal tract.

To implement such a *linear time variant* discrete filter we have to provide an environment where the reflection coefficients can change while the actual filter states are not influenced by the process of coefficient update. On behalf of this, we have to need to read out the current filter states before updating the coefficient set. Henceforth, we will speak of a *filter states memory*. How this is implemented in the case of an FIR and an IIR filter

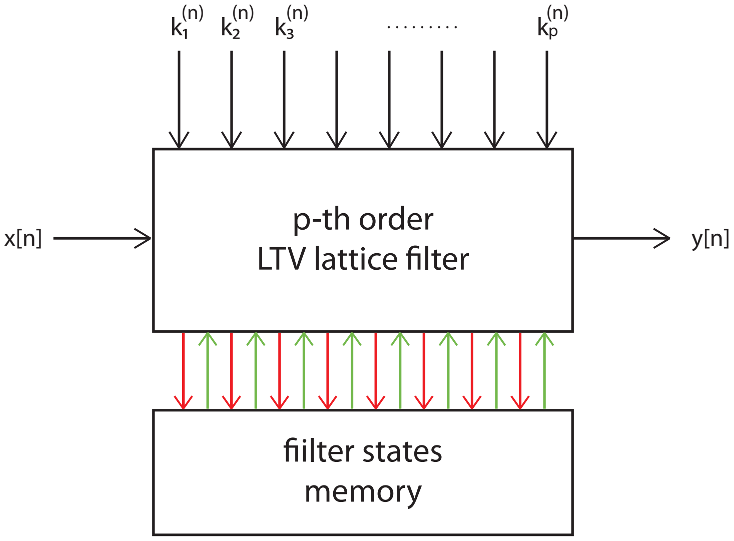


Figure 3.13: Flow graph of an LTV lattice filter; $k_i^{(n)}$ denotes the i-th filter coefficient at the corresponding time step n

is described in laster in this section.

Now let us first recall the relation between the reflection coefficient k_i of the i-th stage of a order p Lattice filter to the forward and the backward prediction error according the *Burg definition* ([1] eq. 9.128):

$$\hat{k}_i = \frac{2\sum_{m=0}^{L-1} e^{(i-1)}[m] b^{(i-1)}[m-1]}{\sum_{m=0}^{L-1} \left(e^{(i-1)}[m]\right)^2 + \sum_{m=0}^{L-1} \left(b^{(i-1)}[m-1]\right)^2} \tag{3.51}$$

All-Zero Filter (FIR)

Due to the recursive nature of the i filter stages we can separate the *forward* $e[n]$ and the *backward* $b[n]$ prediction error path.

$$e^{(i)}[n] = e^{(i-1)}[n] - k_i b^{(i-1)}[n-1] \tag{3.52}$$

$$b^{(i)}[n] = b^{(i-1)}[n-1] - k_i e^{(i-1)}[n] \tag{3.53}$$

The filter states memory is implemented as two p and $p+1$ respectively dimensional vectors that replace the unit delay elements in the circuit. The “now” vector $\boldsymbol{\nu}_n$ is related to the “then” vector $\boldsymbol{\theta}_n$ by

$$\boldsymbol{\theta}_n^{(i)} = \boldsymbol{\nu}_{n-1}^{(i)} \tag{3.54}$$

where the first value of $\boldsymbol{\theta}_n$ is updated by the current input sample

$$\boldsymbol{\theta}_n^{(1)} = s[n] \tag{3.55}$$

The update takes place after the recursive calculation according to equations 3.52 and 3.53 has finished with the last filter stage.

The filter takes the time series $s[n]$ as input and returns the forward error series $e[n]$ that corresponds to the filtered input.

The respective reflection coefficients of a time-step n are located in a coefficient matrix $\mathbf{K}$ that has to meet following restrictions:

1. the number of *rows* has to be equal the *number of input samples* N and
2. the number of *collumns* has to be equal the *filter order* p.

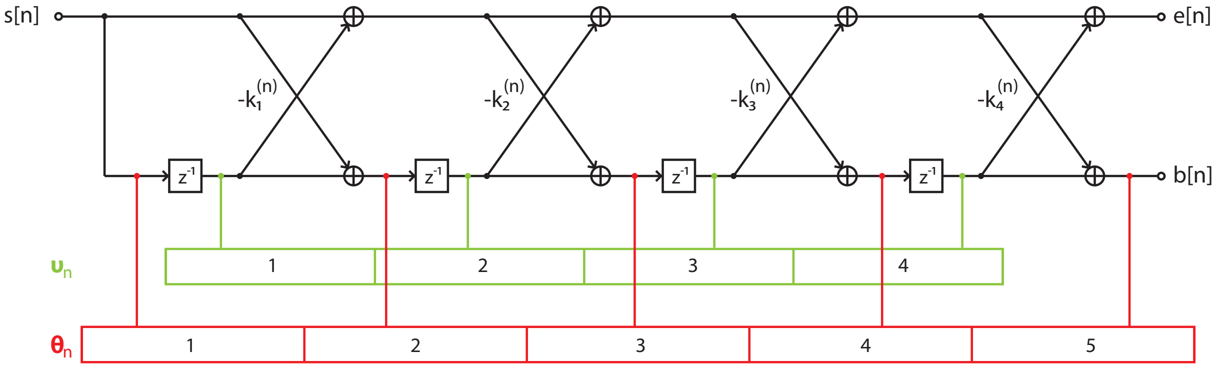


Figure 3.14: Signal path of a $p = 4$ order all-zero lattice filter with *filter state memory* ($\boldsymbol{\nu}_n$ and $\boldsymbol{\theta}_n$)

Thus, such a matrix would have the following form

$$\mathbf{K} = \begin{bmatrix} k_1^{(1)} & k_1^{(2)} & \dots & k_1^{(p)} \\ k_2^{(1)} & k_2^{(2)} & \dots & k_2^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ k_N^{(1)} & k_N^{(2)} & \dots & k_N^{(p)} \end{bmatrix}_{N \times p} \tag{3.56}$$

The algorithm can be written in pseudo-code form as

```
Algorithm 3.5.1: LATTICEFIR($K, x$)

 comment: LTV Lattice All-ZERO filter

 $order \leftarrow$ SIZEOF($K$)
 for $n \leftarrow 1$ to COLLUMNS($x$)
   do
     $k_n \leftarrow K(n, :)$
     $y[n] \leftarrow x[n]$
     for $i \leftarrow 1$ to $order$
       do
         $\boldsymbol{\theta}_n^{(i)} \leftarrow \boldsymbol{\nu}_n^{(i)} - y[n] \times k_n^{(i)}$
         $y[n] \leftarrow y[n] - \boldsymbol{\nu}_n^{(i)} \times$ CONJ($k_n^{(i)}$)
     $\boldsymbol{\theta}_n \leftarrow$ PREPEND($x[n]$)
     $\boldsymbol{\nu}_n \leftarrow \boldsymbol{\theta}_n$
     comment: time shift
 return ($y$)
```

All-Pole Filter (IIR)

The *all-pole* Lattice filter is calculated similar to the all-zero filter. It has the input $e[n]$ and the output $s[n]$. Otherwise than figure 3.15 might suggests, the iteration through the filter stages also takes place from *left to right* - thus from the highest order stage to the smallest order.

The corresponding difference equations 3.57 and 3.58 define the necessary computational steps per iteration.

$$e^{(i-1)}[n] = e^{(i)}[n] + k_i b^{(i-1)}[n-1] \tag{3.57}$$

$$b^{(i)}[n] = b^{(i-1)}[n-1] - k_i e^{(i-1)}[n] \tag{3.58}$$

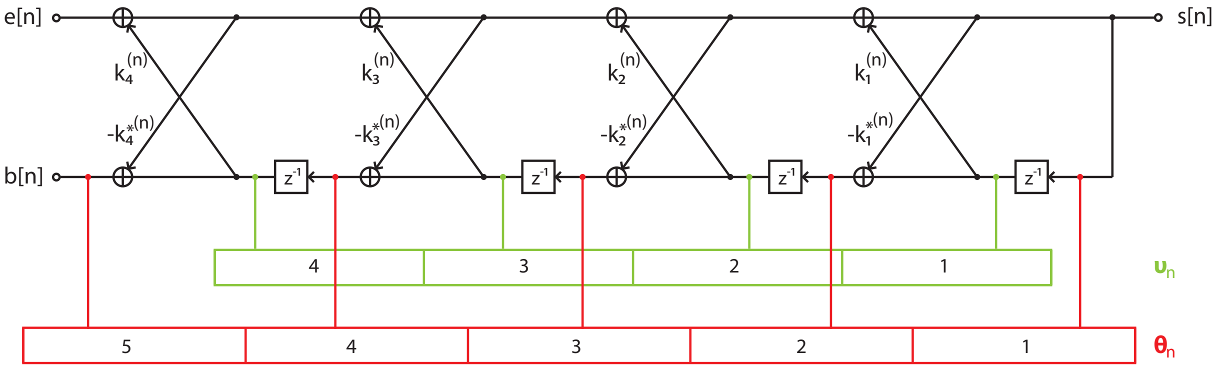


Figure 3.15: Signal path of a $p = 4$ order all-pole lattice filter with *filter state memory* ($\boldsymbol{\nu}_n$ and $\boldsymbol{\theta}_n$)

Again, we can write the algorithm in form of pseudo-code as

```
Algorithm 3.5.2: LATTICEIIR($K, x$)

comment: LTV Lattice All-POLE filter

$order \leftarrow$ COLLUMNS($K$)
for $n \leftarrow 1$ to SIZEOF($x$)
    do
        $k_n \leftarrow K(n, :)$
        $y[n] \leftarrow x[n]$
        for $i \leftarrow 0$ to $order - 1$
            do
                $y[n] \leftarrow y[n] + \boldsymbol{\nu}_n^{(order-i)} \times k_n^{(order-i)}$
                $\boldsymbol{\theta}_n^{(order+1-i)} \leftarrow \boldsymbol{\nu}_n^{(order-i)} - y[n] \times \mathrm{CONJ}(k_n^{(order-i)})$
        $\boldsymbol{\theta}_n^{(1)} \leftarrow y[n]$
        $\boldsymbol{\nu}_n \leftarrow \boldsymbol{\theta}_n$
        comment: time shift
return ($y$)
```

3.5.4 Effects of Parameter Interpolation

As we have introduced the temporal limit of the glottal closed phase to the LPC analysis of our speech signal motivated by the gain of a more exactly measuring vocal tract response, we have to perform some sort of extrapolation for the intermediate time interval.

In the case of synthesis using *A-PSOLA* we inherently perform this step by "cross-fading" windows that have been filtered with different sets of coefficients.

In the case of a Lattice filter on the other hand, we do have to perform this *interpolation* procedure. We have evaluated three kinds of interpolation implementations. Namely

1. keeping the coefficients *static in the OP* and *interpolating in the CP*,

2. interpolating the *entire cycle* between two measurements or

3. keeping the coefficients *static in the CP* and *interpolating in the OP*.

All of which expose similar behaviour as illustrated in figure 3.16. In the areas of linear interpolation the formant frequencies move in a somewhat parabolic manner. Though stability is still granted the influence on the time signal is significant and audible. We have evaluated the three interpolation approaches also in combination with Hann interpolation

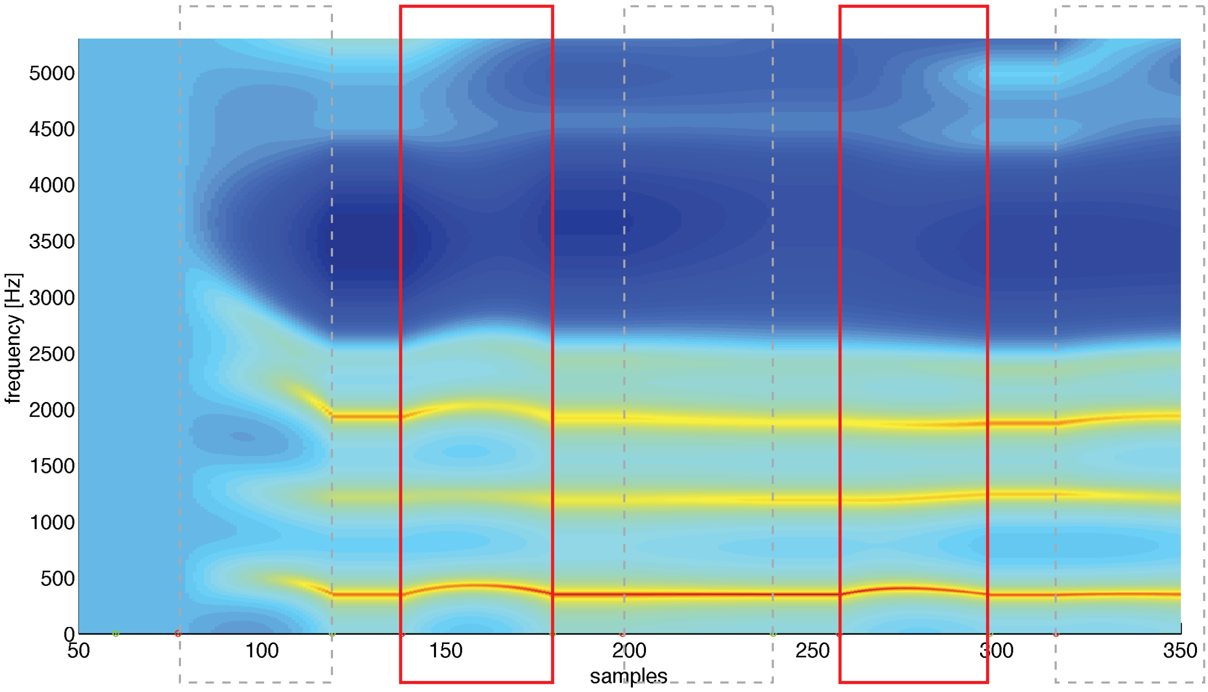


Figure 3.16: Frequency response of the Lattice filter coefficients for 300 samples of a time series; the boxes indicate the areas where the coefficients are linearly interpolated (in the open phase); the additionally *red* colored boxes mark the areas where parameter interpolation has introduced significant distortion on the frequency response.

or additional median smoothing. None of these measures showed significantly improved results.

In order to stick to our earlier agreement of deconvolving the speech signal in the most accurate manner, we decided to employ the third option (keeping the coefficients *static in the CP* and *interpolating in the OP*) because the actual measurement takes place during the CP. Hence, it seems reasonable to us to use the corresponding coefficient set at least for the duration of the closed phase.

3.5.5 Performance Issues

In the courese of this work we have also performed a small comparative study on the different presented linear prediction techniques as well as a straight-forward evaluation of the performance of A-PSOLA and LTV Lattice filtering for synthesis.

Performance of the Different LPC Approaches

To compare the performance of the available linear prediction approaches we have developed the following scenario:

1. we compose an artificial glottal source signal $g[n]$ according to the *Liljencrants-Fant (LF) model* of the glottal source derivative waveform [40], [29]. Additionally to the shape parameters [40] of the model we pass the *period* and the *open quotient* as parameters.

2. we filter $g[n]$ with a vocal tract response $H(z)$ (averaged from a real measurement)

3. we perform closed phase LPC with the following methods:

    a) autocorrelation,

    b) covariance,

    c) Lattice Burg,

    d) Lattice covariance

4. we perform inverse filtering and

5. compare the shape of the estimated GSD waveform $\hat{g}[n]$ as well as the

6. exactness of the spectral estimation of the VTR $\hat{H}(z)$

The results of this small scale study are illustrated in figures 3.17 and 3.18.

In the sense of spectral estimation all approaches except the autocorellation approach seem to perform quite well. The spectral shape of the vocal tract response $H(z)$ i.e. the location of the formant frequencies has been recognized by most the algorithms. Outstanding performance was achieved by the covariance method of classical linear prediction. The graphs (blue and black) overlay nearly perfectly in figure 3.17.

Concerning the temporal domain, a similar behaviour could be ovserved. This seems reasonable due to the duality of spectral and temporal domain. Again, the graph of the original signal and the inverse filtered estimate $\hat{g}[n]$ computed by the classic covariance method are quasi identical.

As a conclusion of this assessment we can say that the *covariance method* of the classical LPC approach outperformed all of its competitors. For this reason we decided to use this approach for our further computations regarding closed phase glottal inverse filtering.

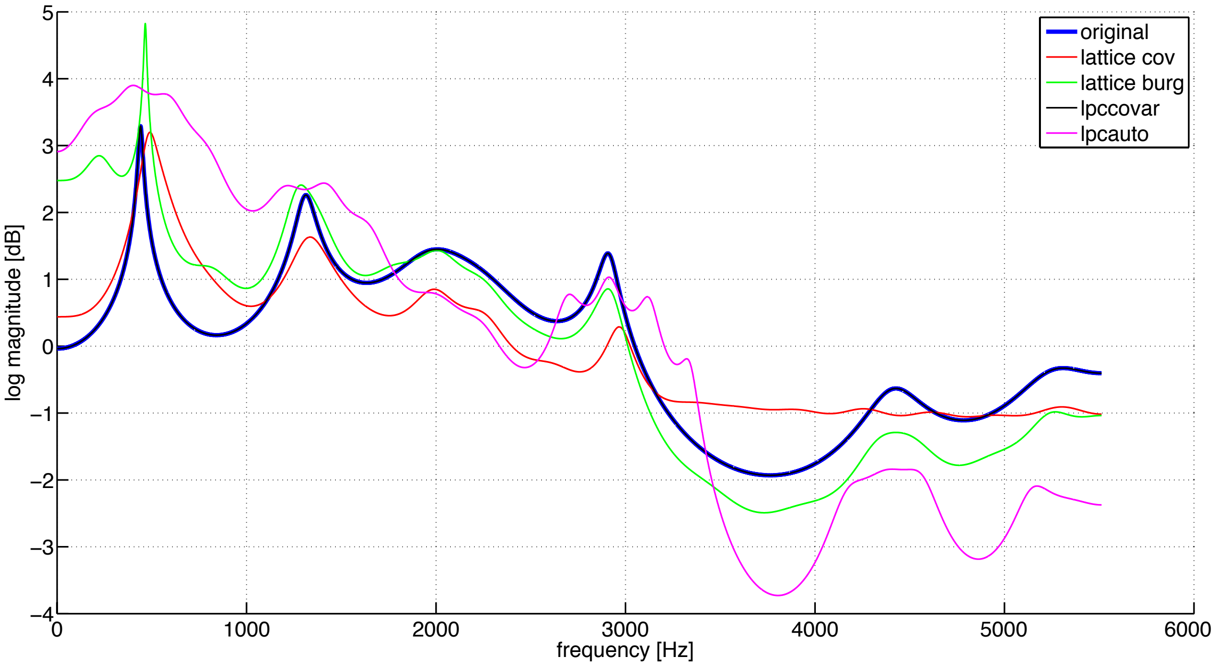


Figure 3.17: Comparison of the spectral fitting properties four LPC approaches; length of CP analysis frame $N = 45$, LPC order $p = 36$; note that the blue (original) and the black (lpccovar) graph overlay nearly perfectly

Decomposing/Recomposing

To get an idea of the distortions introduced by the entire inverse filtering and resynthesising algorithm we performed a "piece of wire" simulation. On this behalf, we perform every step of the algorithm as displayed in figure 3.19 with the special case of *not applying any modifications*.

As a consequence and for an ideally working decomposition and composition algorithm, we would expect $\tilde{s}[n] = s[n]$. Additionally, by using the GCIs as pitchmarks we can ensure that the overall temporal relation of the single signal frames is conserved during the entire algorithm. Thus, we have decided to perform evaluation by comparing the actual *time domain signals*.

The simulations have been performed with an artificial speech signal, generated according to the steps described in the last section.

A statistical description of the resulting error signal $e[n] = s[n] - \tilde{s}[n]$ is described in table 3.1 whereas a graphic illustration is given in figure 3.20.

We have run the simulations for both synthesis techniques described before (A-PSOLA section 3.5.2 and LTV Lattice section 3.5.2). While we have encountered relatively high

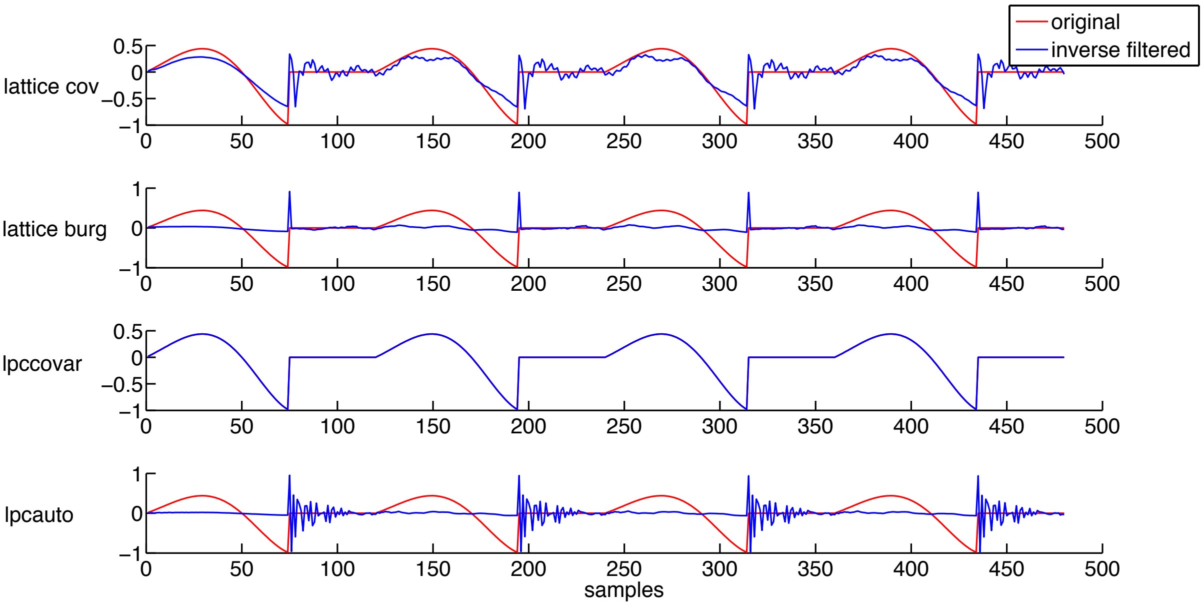


Figure 3.18: Comparison of the performance of four LPC techniques in the temporal domain by means of inverse filtering; note that in subfigure 3 the red and the blue graph coincide ($f_s = 11025Hz$)

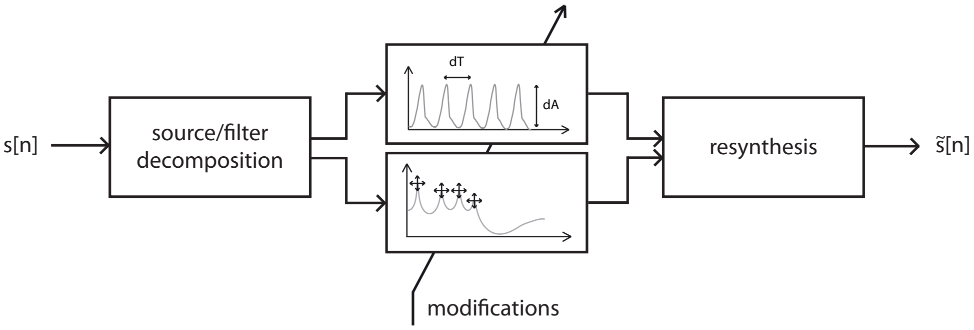


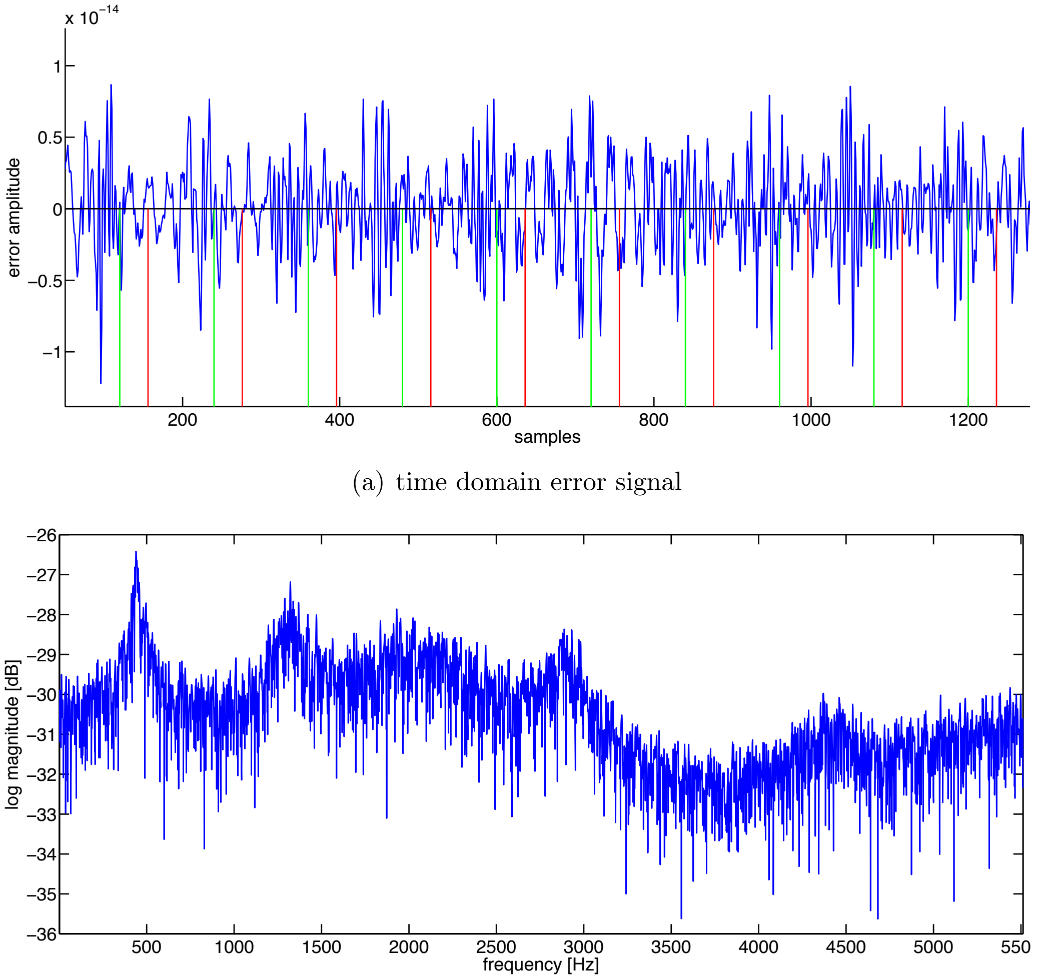
Figure 3.19: Generalized flowchart of the entire algorithm

error variances for the case of A-PSOLA (which we could not resolve up to this point) we have also shown that the LTV Lattice technique seems to be perfectly suitable for decomposing an recomposing the signal.

The simulations have given an average SNR of 32.21dB which we didn't expect in the first place. In terms of audibility the distortions introduced by the LTV Lattice method are therefore clearly negligible.

	mean μ	variance σ^2
A-PSOLA	0.0029	0.0310
LTV Lattice	-0.9724e-17	1.8940e-29

Table 3.1: Mean and variance of the error signal after decomposing and recomposing the speech signal without changes applied ($e[n] = s[n] - \tilde{s}[n]$)



(a) time domain error signal

(b) error spectrum

Figure 3.20: Illustration of the (a) time domain error signal and the corresponding error spectrum using the LTV Lattice method (green marks indicate GCIs, red marks indicate GOIs); (b) depicts the spectral properties of the error signal; note that it generally has very low amplitudes and that it is not completely white but rather exposes a VTR like distribution

4 Constrained Closed Phase Covariance Glottal Inverse Filtering

4.1 Introduction

The algorithm we want to present in the following sections as illustrated in the flow chart consists of multiple layers and iterations. To ease the description we can superficially divide it into the following main parts:

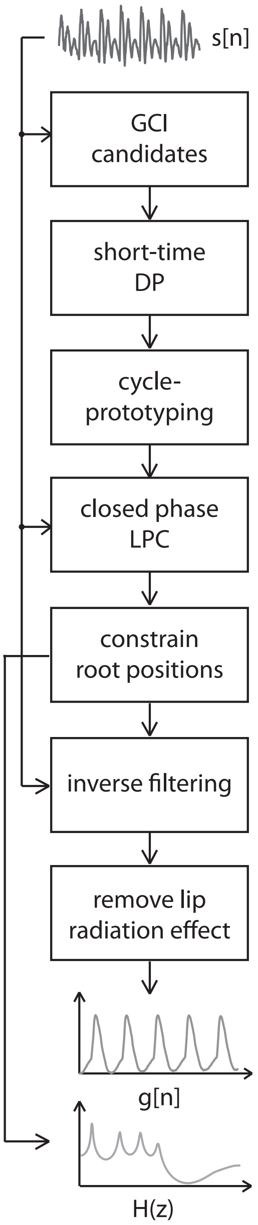


Figure 4.1: Flowchart of the constrained closed phase covariance glottal inverse filtering algorithm used to perform deconvolution of the glottal source signal $g[n]$ and the vocal tract response $H(z)$

1. the *instants of glottal closure* are estimated from the speech waveform (section 4.2)
2. the resulting candidate set is processed to eliminate false positives (section 4.2.6)
3. calculation of a *cylce prototype* to refine the alignment of the GCIs (section 4.3)
4. performance of *covariance LPC* on the estimated closed phase of a glottal cycle to unveil the vocal tract response (section 3.4)
5. constraining the root positions to guarantee invertibility (section 4.4)
6. deconvolution of glottal source $g]n|$ and vocal tract transfer function $H(z)$
7. removal of the lip radiation effect by integration

Some of these steps also refer to the priorly computed pitch and amplitude contours.

After the execution of these steps, an accurate estimate of the glottal source i.e. the volume velocity waveform as well as an exact estimation of the vocal tract transfer function are available for further usage.

4.2 Identifying the Instants of Significant Excitation in Voiced Speech

Primary constraint but also most crucial aspect of closed phase inverse filtering is accurate detection of the *instant of glottal closure (GCI)*, hence, the instant of significant excitation. The GCI determines the beginning of a relatively small analysis frame and allows for a set of simplifications regarding the spectral influence of the glottal source signal to the measured speech signal (see section 1.3.1). Unfortunately, determination of this instants from speech has shown to be a tricky task.

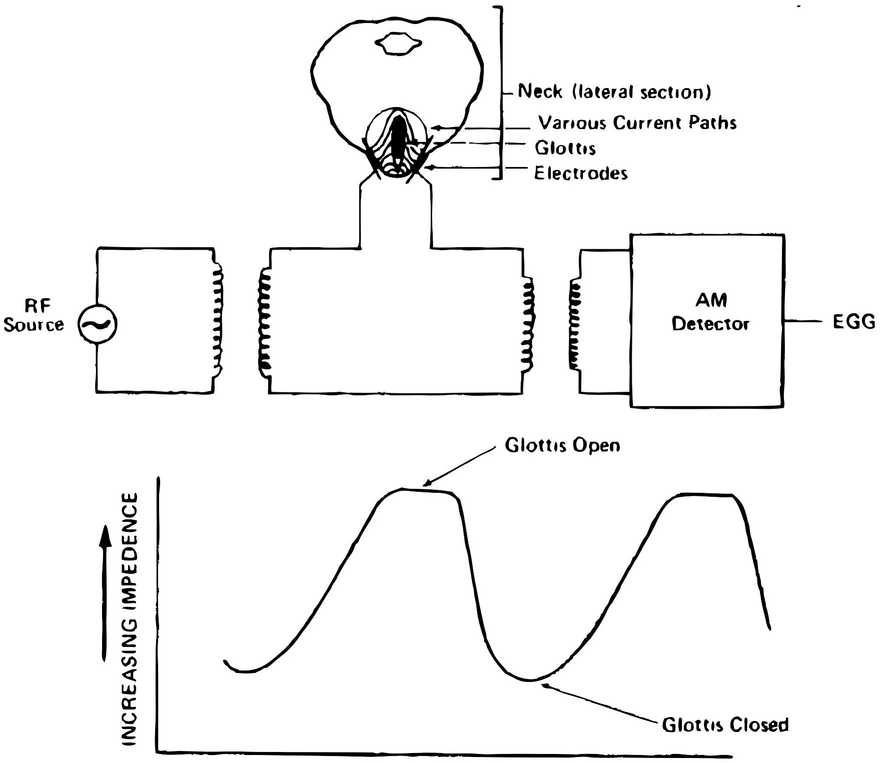


Figure 4.2: Schematic view on an EGG instrument and waveform [17], p. 808

In literature many works use *electroglottogram (EGG)* signals [17] as a reference: those signals are generated by measuring the temporal evolution of electrical impedance through the larynx. The measurements are carried out with a special device containing two electrodes attached to a person's thyroid cartilage providing a continuos measure of the extent of glottal closure i.e. the proximity of the vocal folds. Other techniques using e.g. laryngoscopic devices can provide even exacter measures of the glottal aperture but, due to their invasive nature, they are very uncomfortable for the subject and might also disturb the voice utterance.

These signals have been used to *define* the instant of glottal closure and are widely used in *two-channel* applications, in which an EGG signal is recorded contemporaneously to the speech signal. Although estimation of GCIs from EGG signals can be quite reliably solved by algorithms like the HQTx algorithm [41], or more recently by the SIGMA algorithm [42], in an everyday scenario this information is not available so obviously, to detect the instances of glottal closure from a speech signal we cannot refer to the source signal directly.

Knowledge of the exact position in time of the instant of glottal closure, on the other hand, is the most crucial prerequisite to closed phase analysis. Even small errors in positioning the analysis frame can cause strongly corrupted results. Many different approaches have been presented in literature exploiting different properties of the voice production apparatus.

Naylor et al. have presented the DYPSA algorithm [18] for the detection of GCIs without the usage of an EGG signal in 2007 which also contained a wide range comparison of different methods.

One of the first works in literature dealing with this problem was by Strube [43] in 1974 proposing the usage of the autocovariance matrix of the speech signal. Subsequently refined by Wong [27] in 1979, the authors tended to use an LPC residual signal instead of the speech signal directly. Alternatively, some authors tried to determine the instants of glottal closure by searching for short-time energy peaks within the speech signal e.g. in [44] where the Frobenius norm of the signal matrix is used identify those peaks. In [19] the authors suggest the usage of lossless tube models and LPC analysis to estimate the the waveform of acoustic input power at the glottis which can also be used to determine GCIs.

The most promising approach dealing with the issue of GCIs was proposed by [45] in 1995 and later refined in [46] and [47]. The novel approach utilized a group-delay function to determine the times of major excitation, hence, lottal closure from an LPC residual. In [20] the authors published a large scale study on the robustness of group-delay based methods for identifying GCIs from speech signals. They concluded their study by proposing a novel measure derived from the group-delay function that shall be briefly described in this section together with its underlying ideas.

4.2.1 Basic Idea

The usage of the group-delay measure has emerged with the necessity of identifying the position of a peak within an analysis time frame.

This need further stems from the the assumption that the major excitation of the vocal tract system takes place at this instant of glottal closure [45]. Although there are minor exceptions to this rule regarding weak voice or at the instant of release of a stop burst, where the major excitation does not necessarily correspond to an instant of glottal closure, all those instants are further called *significant instants*.

For a given input signal $s[n]$ a window of N samples is applied at the time r

$$x_r[n] = w[n]s[n+r] \tag{4.1}$$

for $n = 0, ..., N-1$

The N-point discrete-time Fourier transform of this signal can be denoted as

$$X_r(k) = \sum_{n=0}^{N-1} x_r[n]e^{-2j\pi nk/N} \tag{4.2}$$

and correspondingly derive the group-delay as given by [46] as

$$\tau_r(k) = \frac{-d\arg(X_r)}{dk} \tag{4.3}$$

As a first approximation, apart from the issue of significance of an instant, the excitation signal is assumed to be minimum phase within one glottal cycle.

Although the minimum phase property of the glottal volume velocity waveform can not be exploited by our algorithm, simply because we lack the knowledge of this signal, the minimum phase property remains valid under certain conditions also for the speech wave form. The speech signal is the result of the minimum phase glottal excitation signal convolved with the also minimum phase impulse response of the vocal tract including the nasal tract. Due to the quasiperiodic impulse triggered overlapping of those impulse responses, the speech signal loses its minimum phase property. This can be avoided by examining only a short excerpt of time wherein the property still holds. This, on the other hand, causes the problem of a finit length data window to feed the analysis.

According to [48] (chap. 5) the average group-delay of a minimum phase system equals zero. In the case of an impulsive event that excites the system, the average group-delay corresponds to the location of the excitation within the analysis frame [45].

In the original paper [45] from 1995 the authors suggested to compute this measure

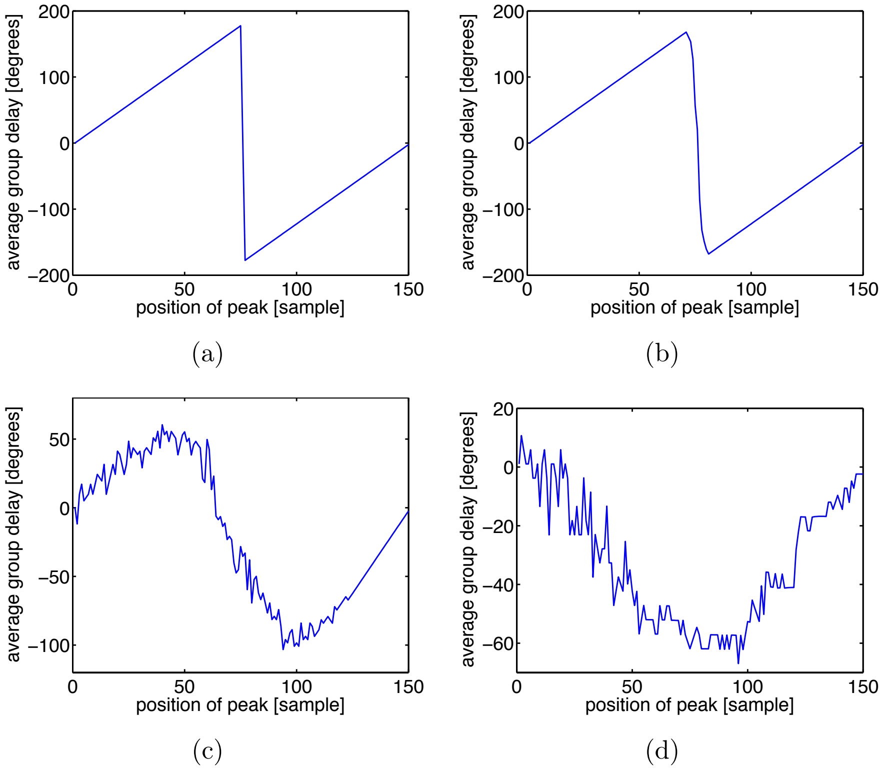


Figure 4.3: Relation of impulsiv event and average group delay of a minimum phase system; a dirac impulse is moved from sample 1 to 150 through a vector of length $L = 150$ and the corresponding average group delay is computed under different conditions: (a) no noise; (b) -40dBFS additive white gaussian noise (AWGN); (c) -20dBFS AWGN; (d) 0dBFS AWGN; note that the negative going zero crossing indicating that the inpuls is located at the center of the window remains clearly visibly also under relatively noisy conditions

by fitting a straight line to the unwrapped phase spectrum of an analysis frame first. Subsequently the slope of this line resembles the position of a peak within the frame i.e. an average slope of zero implies that the excitation is located at the *center* of the analysis frame. Hence, considering the sequence of average phase slopes of a moving analysis window as a series of time, a negative-going zero crossing in this *phase slope function* would indicate the occurrence of an impulse at the center of the corresponding analysis frame.

However, this method lacked robustness to noise and reverberation. In the following publication in 1999 [47] the authors used an analytic approach to compute the group-delay function:

$$\tau_r(\omega) = \frac{X_R(\omega)Y_R(\omega) + X_I(\omega)Y_I(\omega)}{X_R^2(\omega) + X_I^2(\omega)} \tag{4.4}$$

where $X_r(\omega) = X_R(\omega) + jX_I(\omega)$ and $Y_r(\omega) = Y_R(\omega) + jY_I(\omega)$ both are the Fourier transforms of a windowed LPC residual $x[n]$ and $y[n] = nx[n]$. For the sake of readability the index $_r$ has been omitted in equation 4.4 as it is obvious that the real and imaginary parts of X and Y refer to the corresponding frame at time r.

This term was also derived by Brookes et al. in [20] as

$$\tau_r(k) = \Re\left\{\frac{Y_r(k)}{X_r(k)}\right\} \tag{4.5}$$

The authors criticized this approach though because of its computational inefficiency (there have to be computed two Fourier transforms at each iteration step) as well as the missing lower bound for $X_r(k)$ which may lead into a dominance of the denominator in the following averaging step:

$$d_{AV}[r] = \frac{1}{N}\sum_{k=0}^{N-1}\tau_r(k) = \frac{1}{N}\sum_{k=0}^{N-1}\frac{Y_r(k)}{X_r(k)} \tag{4.6}$$

To circumvent this problem the authors suggested an alternative formulation of the problem.

4.2.2 Energy Weighting

They proposed introducing a lower boundary by weighting each term by $|X_r(k)|^2$. This corresponds to the energy content in every frequency bin k. Hence, they chose *Energy-Weighted Group Delay* as an appropriated denomination. This measure is defined by

$$d_{EW}[r] = \frac{\sum\limits_{k=0}^{N-1}|X_r(k)|^2\tau_r(k)}{\sum\limits_{k=0}^{N-1}|X_r(k)|^2} \tag{4.7}$$

After performing a set of simplifications and substitutions (for more detailed information please refer to [20] p. 458) the expression can be rewritten as

$$d_{EW}[r] = \frac{\sum\limits_{n=0}^{N-1} n x^2[n]}{\sum\limits_{n=0}^{N-1} x^2[n]} \tag{4.8}$$

This formulation can be seen as a *center of energy* within an analisys frame $x_r[n]$. The measure also allows for an efficient time-domain implementation, it is bounded and lies in a range from 0 to $N-1$ as long as $x_r[n]$ does not equal zero for the entire length N of an analysis frame.

4.2.3 Robustness Against Additive Noise

The sensitivity of the group-delay measure to additive white Gaussian noise (AWGN) has been studied as well in the original publications [45] and [47] as in the quantitative assessment [20]. In [47] the authors followed an analytical approach to show the independency of the performance of their algorithm from AWGN whereas the latter also introduced a qualitative comparison of the different measures.

Figure 4.4 shows the results taken from [20] comparing the sensitivity of four different methods to signal corruption with AWGN at a range of -30 to +30dB. d_{AV} corresponds to the original measure proposed by Smits et al. while d_{EW} resembles the energy-weigthed group-delay. The other two methods shall not be discussed further in this work (also because of their quite obvious sensitivity to noise).

Remarkably, the d_{EW} measure seems to suffer only very little under noisy conditions - likewise the original d_{AV} measure. What is observable as well is the fact that with decreasing SNR the spread and bias in all four measures increase while a clear tendency of the the median value versus 50 which corresponds to the center of the window in this case.

4.2.4 Robustness Against Echo and Reverberation

Again Murthy et al. followed an analytic approach to deal with this problem [47]. They performed an analytical simulation of a *mild echo or reverberant condition* [47] and concluded that their this would probably not hold for real situations - especially due to the nonstationary nature of speech.

Brookes et al. however focussed only on the case of two impulses falling in one analysis frame, which is often the case when the window length is ill-dimensioned in relation to

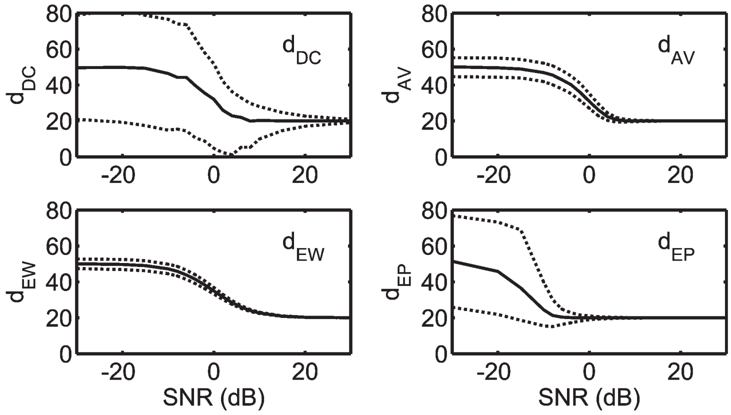


Figure 4.4: Variation of d_{DC}, d_{AV}, d_{EW} and d_{EP} as the signal-to-noise ratio (SNR) varies from -30 to +30dB for an input consisting of a single impulse at $n_0 = 20$ with additive white Gaussian noise in a window length of $N = 101$. For each measure, the graph shows the median value of d_* and the upper and lower quartiles. [20]

the instantaneous fundamental frequency of speech. They evaluated the performance of the four known methods with the following scenario: For an input series

$$x[n] = (1-a)\delta[n] + a\delta[n-n_0] \tag{4.9}$$

containing two imuplses at $n = \{0, n_o\}$ and with a scaling factor $0 < a < 1$.

For the d_{AV} and the d_{EW} measure they analytically derived the following results:

$$d_{AV} = \frac{n_0}{1 - b^{N/\gcd(n_0,N)}} \tag{4.10}$$

$$d_{EW} = \frac{n_0}{1 + b^2} \tag{4.11}$$

where the authors picked $b = 1 - a^{-1}$ for conveniency reasons. The operator $\gcd(\bullet, \bullet)$ denotes the greatest common divisor.

Graphically the comparison is depicted in figure 4.5 and shows the behaviour of the four analysed methods while changing the relative amplitude a from 0 to 1 i.e. shifting the weight from one impulse to the other.

By changing the relative amplitude a of the two peaks in the test signal the methods show different behaviours. Most obvious behaviour shwows the original d_{AV} by choosing simply the larger peak. This is due to the dependency of the exponent in equation 4.10 on the $\gcd(n_0, N)$ operation and causes the very rapid transition between the decisions.

The d_{EW} measure on the other hand shows a relatively smooth transition. The S-shape

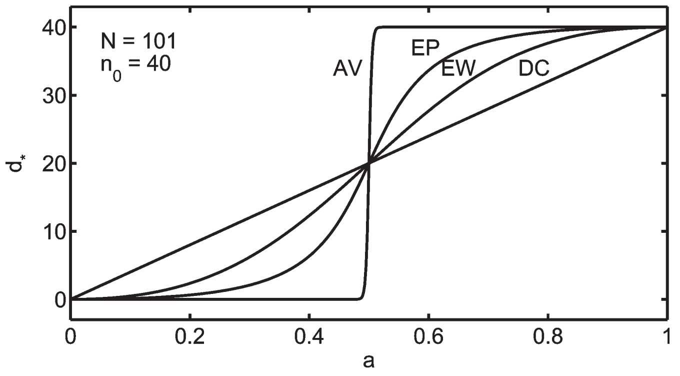


Figure 4.5: Values of d_{DC}, d_{AV}, d_{EW} and d_{EP} for a signal containing impulses at samples 0 and 40 of amplitude $1-a$ and a respectively. The window length is 101 and a varies between 0 and 1. [20]

of the curve stems from the definition of the d_{EW} measure as a center of gravity rather than a fixed threshold as it is the case with the d_{AV} measure.

The authors conclude that the steepness of the curve as it is the case for d_{AV} could be achieved by increasing the exponent of the d_{EW} as well but it has not necessarily been shown to be useful i.e. improve the performance in a real scenario.

4.2.5 Effects of the Window Length

The considerations in the last section consequently lead to the question "How does the window length affect the results?". Naturally, if two impulses fall within a single analysis frame, the algorithm has to decide, which of both is to be treated as a significant instant. If, on the other hand, the window size is chosen in such a way there is *only one* impulse within the window, this problem could be omitted entirely.

Brookes et al. have discussed this case as well and their thoughts are illustrated in figure 4.6.

Quite obviously the case in which the window length matches the predominant period of the impulse chain returns the best results (as also noted in [18]). In this illustration the d_{EP} measure has been used but the authors lined out similar performance of the d_{EP} and d_{EW} measures. Therefore, figure 4.6 can be considered representative also for the d_{EW} measure.

In this work we try to exploit this property of the group-delay measure by using a variable window length computed from the instantaneous frequency estimation by the YIN algorithm. By doing this we can ensure that the window length always lies within

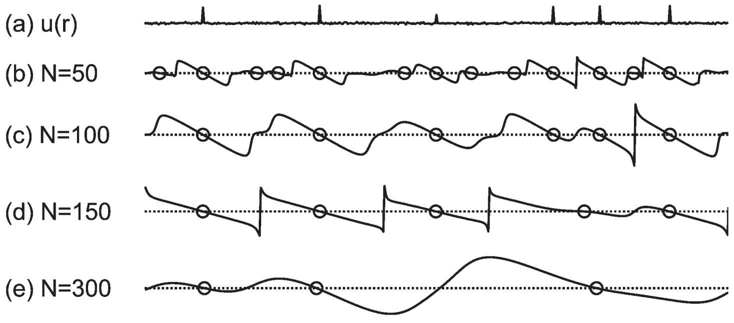


Figure 4.6: (a) Impulse train with a dominant period of 100 samples and an SNR of 10dB. (b)-(e) the waveform of d'_{EP} for different window lengths, N. The circles mark the negative-going zero crossings (NZCs). [20]

reasonable values and the probability of an occurrence of multiple excitations within an analysis frame is reduced. The task of determining and removing spurious impulses, as they might for instance occur by ill-dimensioned analysis frames as displayed in figure 4.6, is done by post-processing stages that will be described in to subsequent sections.

4.2.6 Removing False Positives from the Candidate Set

Using the group-delay measure with a samplewise sliding window allows us to generate a set of candidates that should contain *at least the really occurring* GCIs. This can be assured by using a window length that is picked dynamically corresponding to the instantaneous frequency: at no time two GCIs can fall within a single analysis window which could lead into a missed candidate (i.e. if one of both is very much larger that the other).

Hence, any spurious GCIs appearing in the candidate set can be denoted as *false positives.*

A candidate set at this state could for instance look like depicted in figure 4.7: it depends very much on the order of LPC filtering. In the course of this work relatively low LPC orders of e.g. five ($f_s = 11025Hz$) have shown to provide a very satisfying estimate.

Now every candidate has to be evaluated if its occurrence is justifiable or spurious. This is done by computing a multi-dimensional cost-vector $\mathbf{c}(k)$ for each candidate k. The dimensions of this cost-vector are defined by a set of considerations that must hold for a *true* GCI.

On the first hand, one can assume that the speech signal excerpt around a rightfully detected candidate shows a large value of correlation with the corresponding excerpt of the

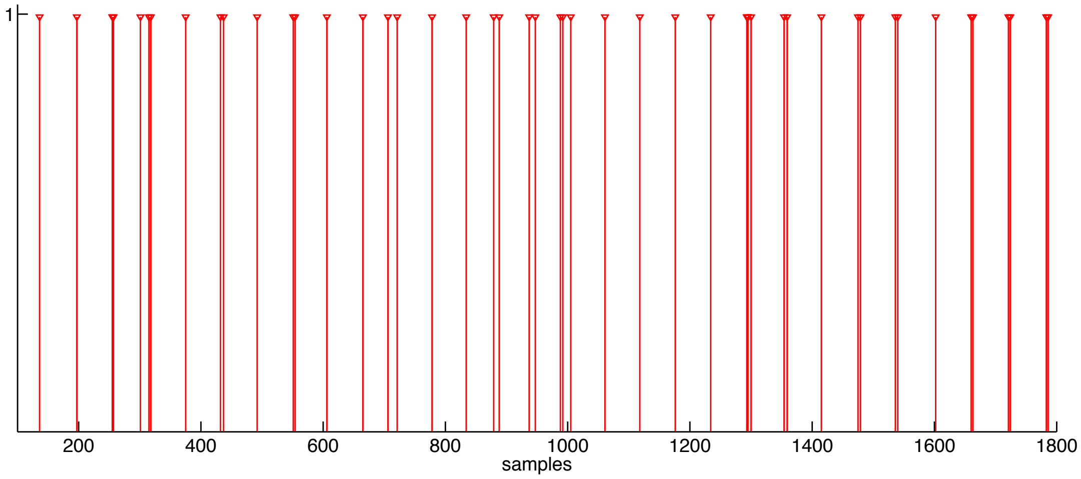


Figure 4.7: Initial GCI candidate set containing a vast number of false positives; LPC order $p = 10$; computed from a 1s baritone recording ($f_s = 11025Hz$)

preceding (and rightfully detected) candidate. This leads to the so-called *speech waveform similarity cost* as also supposed in [18]. In contrast to Naylor et al. the cost is computed for three different combinations of candidates which allows to recognize similarities on a wider temporal range.

The costs are computed as

$$c_{WS_1}(k) = 1 - \frac{\gamma_{k-1,k}}{\sqrt{\gamma_{k-1,k-1}\gamma_{k,k}}} \tag{4.12}$$

$$c_{WS_2}(k) = 1 - \frac{\gamma_{k-2,k}}{\sqrt{\gamma_{k-2,k-2}\gamma_{k,k}}} \tag{4.13}$$

$$c_{WS_3}(k) = 1 - \frac{\gamma_{k-3,k}}{\sqrt{\gamma_{k-3,k-3}\gamma_{k,k}}} \tag{4.14}$$

where $\gamma_{k-1,k}$ denotes the cross-covariance of the two speech waveform frames of the corresponding analysis frames for the candidates k and $k-1$. The terms of the denominator like $\gamma_{k-1,k-1}$ and $\gamma_{k,k}$ denote the auto-covariance of the corresponding speech signals - thus the signal power.

The cost itself is derived from the *normalized cross-correlation estimator* and has a value between 0 and 1 where 0 resembles full correlation (hence no cost) and 1 implicites full decorrelation.

The second, and maybe the more severe criterion for detecting true GCIs is the constraint that there can not occur any severe variations in pitch within the time interval of a single glottal cycle. This is also a reason for the short-term stationarity of speech that claims that both, the excitation signal and the vocal tract response, vary only very slowly in time (compared to our analysis times) and can therefore be assumed as stationary for a single analysis window. This leads us to the following *Pitch Deviation Cost* as again suggested in [18]:

$$c_P(k) = 1 - e^{-|\psi(\delta-1)|^2} \tag{4.15}$$

where the pitch deviation δ is defined as

$$\delta = \frac{\min\{n_k - n_{k-1}, n_{k-1} - n_{k-2}\}}{\max\{n_k - n_{k-1}, n_{k-1} - n_{k-2}\}} \tag{4.16}$$

The variable n resembles the sample where the analysis frames of candidates indexed as k are centered.

The cost allows for small penalties for small pitch deviations up to 25% whereas larger deviations are penalized more severely according to the exponential relationship. In [18] Naylor et al. have suggested a value of $\psi = 3.3$ to achieve this behaviour. The cost has also a value range of 0 to 1 where again 0 means a well located candidate according to the preceding candidates.

One big disadvantage of this measure has been revealed when e.g. a spurious candidate is followed by two rightful candidates. In this case, due to the min/max formulation in equation 4.15 the latter would be penalized with a large cost and probably dismissed from the candidate set. Therefore an alternative formulation has been found that we call the T_0 *Deviation Cost*.

As we have knowledge on the instantaneous frequency at every analysis frame, we can use this information to omit the min/max formulation of equation 4.15 and write

$$c_{T_{0_1}}(k) = \frac{|n_k - n_{k-1}|}{T_{0_{k-1}}} \tag{4.17}$$

$$c_{T_{0_2}}(k) = \frac{|n_k - n_{k-2}|}{2T_{0_{k-2}}} \tag{4.18}$$

$$c_{T_{0_3}}(k) = \frac{|n_k - n_{k-3}|}{3T_{0_{k-3}}} \tag{4.19}$$

where n again denotes the center of the analysis frame of candidate k and T_0 the corresponding instantaneous frequency estimate computed by the YIN algorithm.

To establish an upper boundary the following constraint is applied to each cost:

$$c_{T_{0_j}} = \begin{cases} c_{T_{0_j}} & \text{if } c_{T_{0_j}} < 1, \\ 1 & \text{else} \end{cases} \tag{4.20}$$

where the index j denotes the level of time shift for the comparison. Now the measure is bounded beween 0 and 1 where again 0 resembles a maximum likelihood that the candidate k is a valid one whereas 1 suggests a spurious detection.

The final cost-vector $\mathbf{c}(k)$ can now be defined as

$$\mathbf{c}(k) = \begin{bmatrix} c_{WS_1}(k) \\ c_{WS_2}(k) \\ c_{WS_3}(k) \\ c_P(k) \\ c_{T_{0_1}}(k) \\ c_{T_{0_2}}(k) \\ c_{T_{0_3}}(k) \end{bmatrix} \tag{4.21}$$

The DYPSA algorithm [18] uses also further measures like the *Normalized Energy Cost* or the *projected candidate cost* and the *ideal phase-slope function deviation cost.* In this work this cost-measures have not been employed as the usage of the seven-dimensional cost-vector $\mathbf{c}(k)$ has exposed very satisfying results.

Having defined multiple dimensions of our cost-vector, each of which lying in a range from 0 to1, the overall cost is computed by it's geometrical distance from the origin. In contrast to the DYPSA algorithm, where the overall cost is defined as

$$c(k) = \boldsymbol{\lambda}^T \mathbf{c}(k) \tag{4.22}$$

where $\boldsymbol{\lambda}$ is a weighting vector for the costs and therefore the overall-cost $c(k)$ is a weighted sum of the dimensions of $\mathbf{c}(k)$.

In our approach this value is computed as the *Euclidian Distance* of $\mathbf{c}(k)$ from the origin. Thus, the overall cost of candidate k is defined as

$$c(k) = \sqrt{\sum_{j=1}^{J} \mathbf{c}_j(k)^2} \tag{4.23}$$

where J denotes the number dimensions of $\mathbf{c}(k)$ or more intuitively

$$c(k) = |\mathbf{c}(k)| = \sqrt{\mathbf{c}(k)^T \mathbf{c}(k)} \tag{4.24}$$

This measure is positive and bounded by the number of dimensions of the cost-vector as

$$\lim_{c_j \to \max c_j \forall j} c(k) = \sqrt{\sum_{j=1}^{J} (\max c_j(k))^2} = \sqrt{J} \tag{4.25}$$

provided that $0 \leq \mathbf{c}_j(k) \leq 1 \; \forall j$ which holds in our case.

As a consequence of the existence of an upper and a lower boundary the usage of a fixed threshold for further analysis is facilitated.

Using this measure directly as a statical quantifier of a single candidate however is not very sensible. The cost value is always derived from the candidate k itself and its predecessors $k-1$, $k-2$ and $k-3$. This means that the likelihood of candidate k depends crucially on the prior estimated candidates. Consequently, if, and only if, the three preceding candidates have been detected rightfully, the candidate k can have a minimum cost $c(k)$.

While these thoughts have been dealt with also in the DYPSA algorithm, where the authors decided to use the capabilities of *dynamic programming (DP)* to overcome this problem, we tried to avoid this approach because of one major disadvantage: dynamic programming relies on the *Viterbi algorithm* [49] which tires to find a forward trellis by determining the cost minimum at the end of a time series and performing the trace back. As a direct consequence this leads to lack of real-time capability.

Like in the case of DYPSA, the DP analysis can also be performed on subsets of the time series of candidates but it comes with some kind of uncertainty regarding the boundaries between the subsets. Accounting for the continuity of the entire time series, the separation into multiple subsets leads also into the situation that every subset is treated individually with the same initial probability distribution. In other words, even if an optimum solution for a single subset can be found, it can not guarantee consistency for the entire data set.

Even though there are publications on a *short-time Viterbi algorithm* like described e.g. in [50], we tired to avoid a Markovian approach at all and find a more straight-forward solution.

In our approach we evaluated the cost of every candidate k by using various combinations of the preceding candidates so as to find one combination that leads to a minimum cost $c(k)$ which then determines the combination of candidates. See an example in figure 4.8 where various combinations of candidates lead to different costs of candidate k.

Mathematically the minimum solution can be derived from the general formulation of the cost

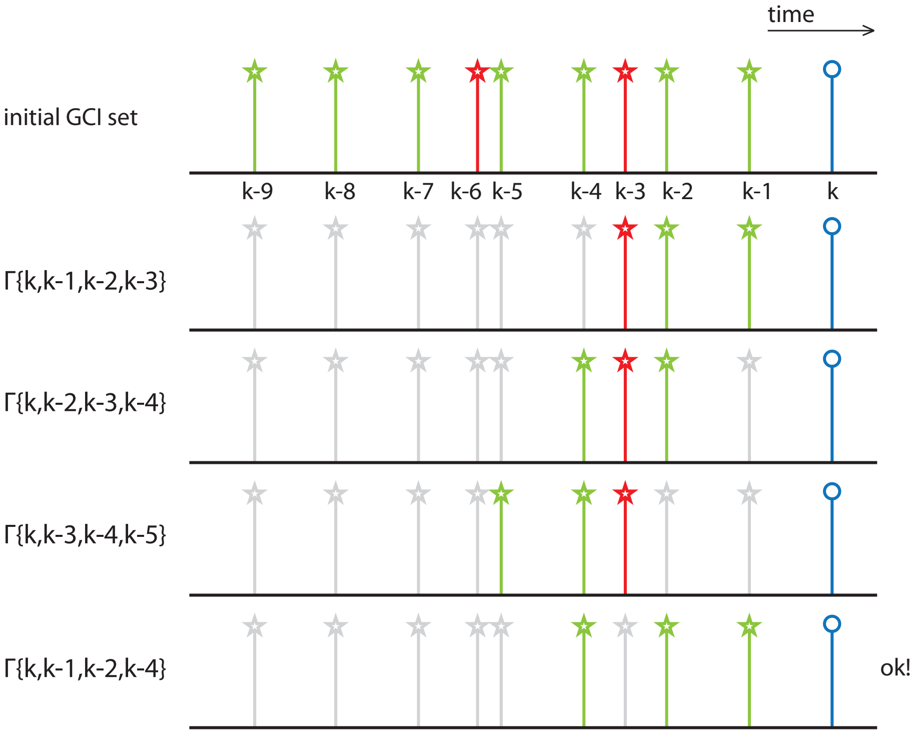


Figure 4.8: Various combinations of candidates and the resulting cost for candidate k (blue circle); excerpt of the calculation of the dynamic cost matrix $\mathbf{C}(k)$ according to eq. 4.27; initial set consists of true positives (green asterisk) and false positives (red asterisk); last example combination would have minimum cost value in $\mathbf{C}(k)$

$$c(k) = \Gamma\left\{\Omega\right\} = \Gamma\left\{k, k_1, k_2, k_3\right\} \tag{4.26}$$

where Γ denotes the operator of cost calculation and the parameters k, k_1, k_2 and k_3 correspond to the respective candidates of subset Ω. Those candidates vary according to following formulation

$$\mathbf{C}(k) = \begin{bmatrix} \Gamma\left\{k, k-1, k-2, k-3\right\} & \Gamma\left\{k, k, -1k-3, k-4\right\} & \Gamma\left\{k, k-1, k-2, k-4\right\} \\ \Gamma\left\{k, k-2, k-3, k-4\right\} & \Gamma\left\{k, k-2, k-4, k-5\right\} & \Gamma\left\{k, k-2, k-3, k-5\right\} \\ \Gamma\left\{k, k-3, k-4, k-5\right\} & \Gamma\left\{k, k-3, k-5, k-6\right\} & \Gamma\left\{k, k-3, k-4, k-6\right\} \end{bmatrix} \tag{4.27}$$

where $\mathbf{C}(k)$ denotes the dynamic cost matrix.

To find and optimum solution for the current candidate k within the candidate set Ω simply the minimum element of $\mathbf{C}(k)$ has to be found and the candidate set has to be rearranged according to combination that has lead to the minimum

$$\Omega_{opt} = \operatorname{argmin} \mathbf{C}(k) \tag{4.28}$$

where Ω_{opt} denotes the optimum solution for the affected subset.

In some case, though, there cannot be found any reasonable combination for candidate k resulting into a relatively high cost. Thus, it can be assumed that candidates with e.g. a cost $c(k) > 0.5$ will never take part in a sensible combination within the temporal evolution and are therefore dismissed from the data set. Note that this is the only possibility where a candidate k can be dismissed *directly* - in all other cases the candidate k only determines which combination of preceding candidates in Ω remains.

Now the two major differences of our approach compared to DYPSA can be summarized as

1. *recursive nature of cost calculation*: candidates are not dismissed directly because of their high cost but a subset of candidates is picked that leads to a minimum cost for the actual candidate

2. *quasi-real-time capability* (at least theoretically) due to its iterative calculation omitting optimization by back-tracing: only a small set of candidates is necessary to compute the local minima (example: 7 candidates used for optimization; $F_0 = 200$Hz; resulting delay (without computational load): 35ms)

4.2.7 Results

The performance of the algorithm has not yet been evaluated on a large scale test but first experiments have shown quite satisfying results. See figure 4.9 for an example of the entire algorithm performing.

The example in figure 4.9 is taken from a recording of a Baritone singer singing the vowel “e” on a sustained note at a pitch of approx. 186Hz. The recording was sampled at 44100Hz but has benn downsampled to 11025Hz for the sake of computational efficiency.

The sample has *not* been recorded in an anechoic chamber - hence is corrupted by reverberation and additive interferences. Yet, the algorithm achieves to compute a remarkably “fluid” oscillation of the group-delay measure which is also due to the LPC order of 5 which (as pior mentioned) has exposed the most convenient results for this sample rate.

To demonstrate the power of the optimization step see figure 4.10 where a vast amount of false positives is effectively reduced to a reasonable number.

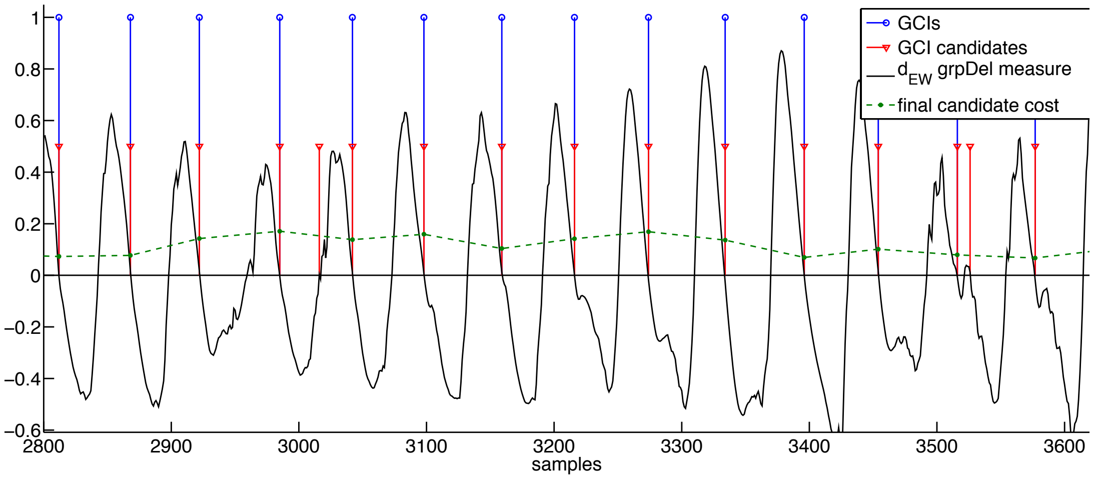


Figure 4.9: Example of entire algorithm: negative-going zero crossings of group-delay measure (black) indicate GCI candidates (red); note the spurious overdetections at ca. samples 3000 and 3530; resulting candidates (blue) after postprocessing with corresponding final cost (green/dashed); computed from a 1s baritone recording ($f_s = 11025Hz$)

In contrast to the example in figure 4.9, this example has been computed with an LPC order of 10 wich dramatically increased the number of initial false positives. This effect seems reasonable if you consider that higher orders of LPC can model the vocal tract response more precisely. Therefore, the residual signal will not expose an as distinct peak at the instant of glottal closure as the lower order residual might expose. Though there are lots of false positives, the optimization algorithm successfully finds the most harmonic solution for the entire time series.

Tests also have shown that further increasing of the LPC order leads not only into higher computational load but also into a certain performance limit of our algorithm. With an order of 15 or above the results start to become corrupted e.g. certain GCIs are missing. Therefore a certain amount of “imperfection” in modelling the vocal tract response by means of a relatively low order LPC analysis has lead to the most convincing results.

4.3 Cycle Prototyping

The detection of the instant of glottal closure returns a vector containing the calculated and validated candidates as well as a vector containing the corresponding instants of glottal opening. Note, that for the moment the estimation of the GOIs relies only on an

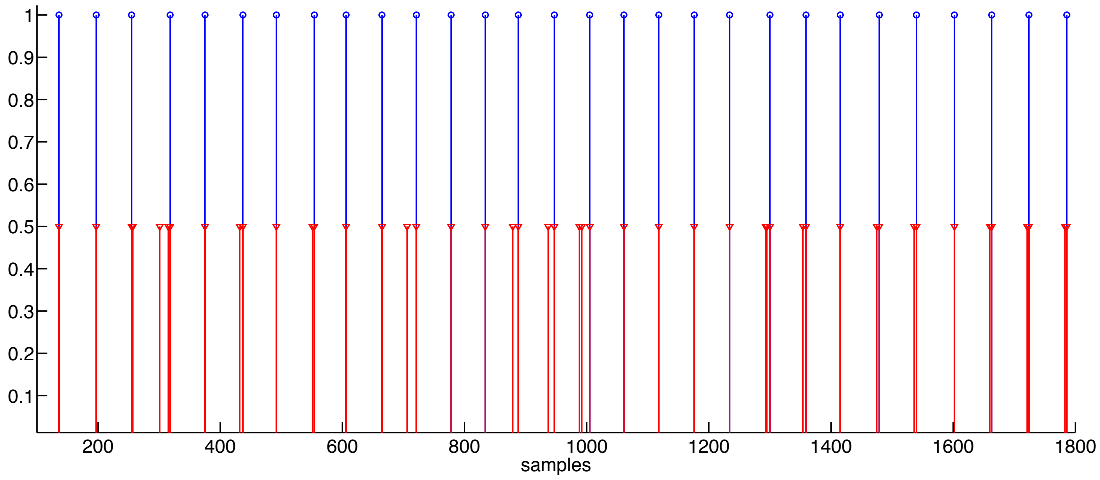


Figure 4.10: Example of 10th order LPC leading into vast overdetection of candidates (red). After optimization a set of GCIs (blue) remains that exposes high periodicity; computed from a 1s baritone recording ($f_s = 11025Hz$)

a priori fixed assumption of the *open quotient* oq:

$$oq = \frac{T_{open}}{T_{open} + T_{closed}} \tag{4.29}$$

The exact estimation of the instant of opening is very difficult as the rise in energy in the excitation signal is hardly recognizable by the method described in the last section. As pointed out in [18] and [45] the exact positioning of the GOI is not crucial for inverse filtering. Therefore, we keep this simplification for the subsequent steps and update the open quotient after having performed inverse filtering.

4.3.1 The Problem

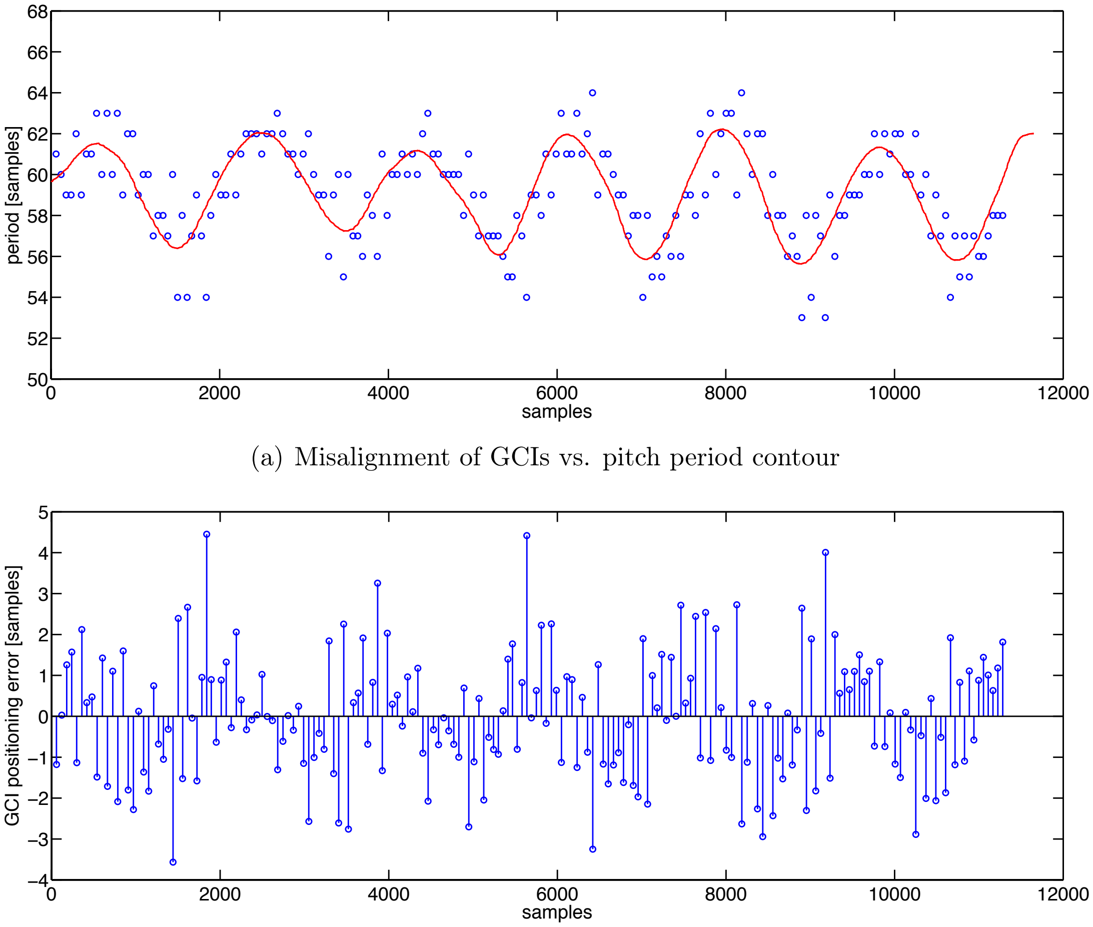
The sequence of GCIs can be regarded as marks that indicate the periodicity of a signal. In literature they are sometimes referenced as *pitch-marks*. Now this sequence has to be related to the pitch of the signal i.e. taking the derivative of the GCI sequence we would expect approximately the shape of the pitch or rather the *pitch period contour* T_0:

$$\frac{d}{dk}GCI(k) \approx T_0(GCI(k)) \tag{4.30}$$

The variable k denotes the k-th instant.

Of course both sequences will not be absolutely equal due to the quantisation introduced by the sampling rate but rounding the corresponding values of T_0 would have the same effect.

Our first idea was to simply modify the GCI positions so they would fit the pitch period contour as displayed in figure 4.11.



(a) Misalignment of GCIs vs. pitch period contour

(b) Positioning error in samples; $\mu = -0.0270$, $\sigma^2 = 2.2854$

Figure 4.11: Set of GCIs surrounding a pitch contour computed from a 1s baritone recording ($f_s = 11025Hz$); in (a) the GCIs (blue circles) should lie as closes as possible to the pitch contour (solid red line); (b) shows the respective deviation from the pitch in samples

By using brute-force in repositioning GCIs we could achieve quite promising results. The one major problem of this approach was that by building the derivative of the GCI series all following values were dependent on the *first* value. Thus, all measures applied on the derivative of the GCI sequence were inherently prone to the mispositioning of the very first GCI. For this reason we decided to search for another solution of this problem.

4.3.2 The Idea

On property of the GCI misalignment we discovered is its *zero-mean* nature as indicated in figure 4.11(b). This, consequently, suggests that the misalignment could be reduced by some kind of *averaging*.

Using the glottPlot (see section 2.6 on how a glottPlot is constructed) as a utility to give a measure of the alignment of multiple glottal cycles allowed for a very interesting observation. As depicted in figure 4.13(a), we used the initial set GCIs to visualize the *coherence*. Clearly recognizable, the glottPlot shows a high amount of *blur* or *fuzziness*. This is due to the fact, that the exact temporal positioning of the GCI as the *beginning* of a single cycle was not coherent for multiple cycles.

Nevertheless, the basic relations between the cycles, hence the shape of the waveform, seemed to be *approximately consistent*. Thus, the idea we have come up with is that by *averaging the cycles* we could compute a *cycle prototype* that reflects rough shape of a single glottal cycle.

Thus, we can define the cycle prototype $c_{proto}[m]$ as

$$c_{proto}[m] = \frac{1}{K} \sum_{k=1}^{K} x[GCI(k) + m] \tag{4.31}$$

where K denotes the number of involved cycles and and the integer m is defined in the interval $0 \leq m \leq \max\{T_0\}$.

Naturally, this simplification can only be valid if the set of observations is consistent. Large changes in fundamental frequency or vocal tract response might distract the algorithm. Therefore, the limitation to a relatively static pitch area should be introduced. This is still the case for a sustained vowel, even with a strong vibrato sung. An example for such a prototyp is provided in figure 4.12.

Now the prototype can be regarded as the *approximate shape* of a glottal cycle for a certain time period. Thus, singling out one glottal cycle any divergence of from the prototype would indicate that the GCI estimate was not consistent. Therefore, by comparing the shape of every single cycle to the prototype we can measure the deviation.

Mathematically, this is done by means of the *normalized crosscorrelation* as denote in eq. 4.32

$$\text{ncf}(k) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{(x[n] - \bar{x}_N[n])(y[n-k] - \bar{y}_N[n])}{\sigma_x \sigma_y} \tag{4.32}$$

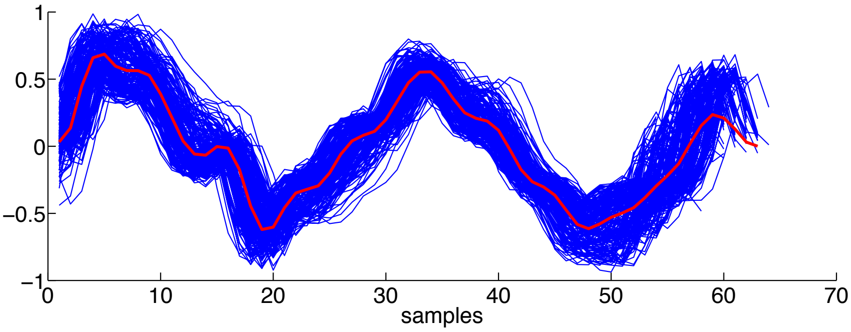
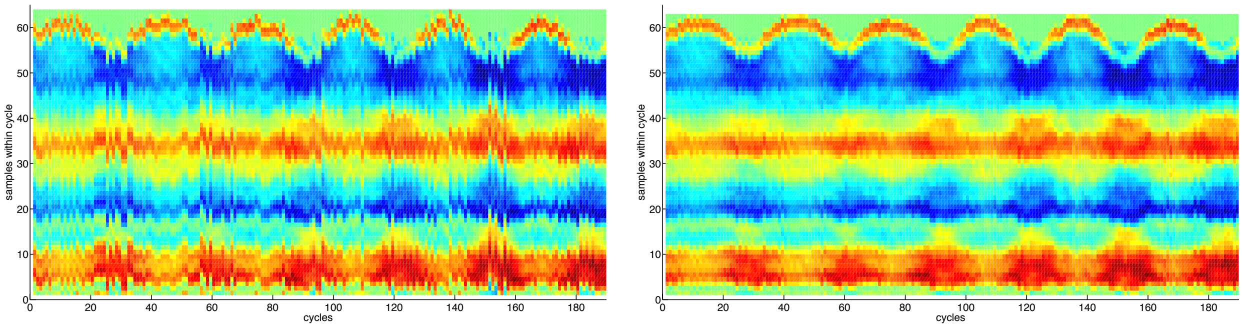


Figure 4.12: Cycle prototype; the blue lines correspond to the overlaid plotting of the involved glottal cycle; the red line resembles the prototype determined by averaging ($f_s = 11025Hz$)

and by finding the peak value within this function we can determine the amount of deviation from the prototype - hence, a temporal *offset* in samples.

Using this offset and *subtracting it from the corresponding GCI* provides the alignement we have desired (according to the zero-mean assumption from before).

A comparison of *before* and *after* cycle prototyping is illustrated in figures 4.13(a) and (b) as well as in figure 4.14 where the resulting GCI positioning is depicted.



(a) Uncorrected

(b) Corrected

Figure 4.13: GlottPlots of the original and the corrected set of GCIs based on the raw speech audio data; note that in the right figure the “blur” has been significantly reduced

4.3.3 Limitations

As mentioned before, we have to ensure that the prototype is built from a consistent set of observations. No dramatic increase or decrease in pitch, nor severe changes in the vocal tract filter may occur as this would corrupt the prototype.

Additionally, we can not judge the *absolute* positioning of the GCI set. By prototyping and realigning we can only ensure that position of the GCI is consistent *within* a glottal cycle. No assumptions may be made accounting for the validity of the actual GCI. However, we do know that the a misplacement of the a single GCI will be the *same for all* GCIs involved. Therefore, we could use this step to "tweak" inexact estimations in retrospect by introducing an additional linear offset.

4.3.4 Results

As depicted in figure 4.14 we have achieved a significantly better alignment to the pitch contour. Of course, the discretization due to the temporal resolution of the sample rate can not be affected by this method. The remaining positioning error remains at values of max. one LSB.

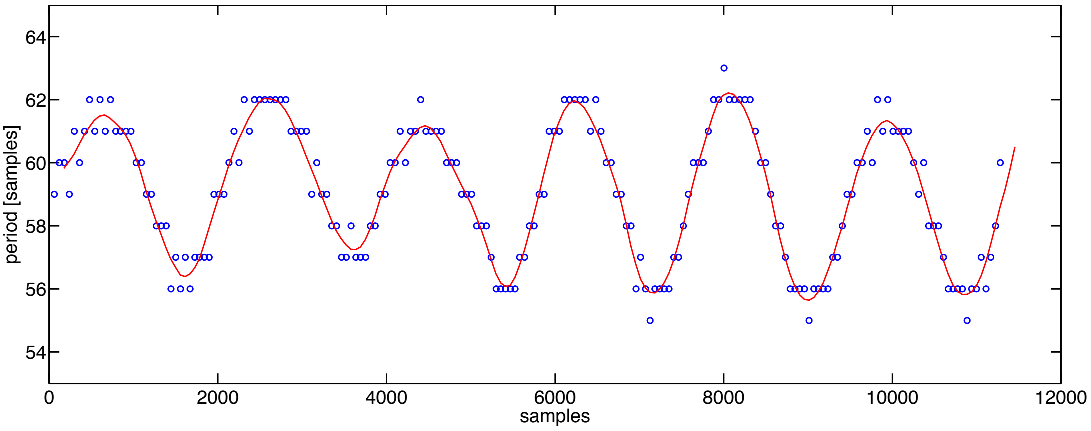


Figure 4.14: Resulting GCI positioning after cycle prototyping ($f_s = 11025Hz$)

4.4 Constraining Root Positions

At this moment we finally have available a trustworthy set of glottal closure instants and can use one of the prior described approaches of linear prediction to compute the spectral envelope of the vocal tract transfer function.

Alku and Magi [16] have proposed a modified approach to linear prediction in which they propose a method to inherently constrain the DC influence of the VTR estimate. This is a measure to circumvent problems that arise with misalignment of GCIs for inverse filtering.

Additionally, they performed an evaluation of the *minimum-phase property* of every VTR estimate $\hat{H}_r(z)$. This is due to the fact that we need to guarantee *invertibility* of the filter. As described in detail in section 3.4 the linear prediction error filter $\hat{A}(z)$ actually equals the inverse of the vocal tract filter, hence

$$\hat{A}(z) = \frac{1}{\hat{H}(z)} \tag{4.33}$$

if we consider the excitation signal to be zero within the analysis frame.

As the prediction error filter is FIR the stability issue caused by roots located on or outside the unit circle is not of immediate importance. Nevertheless, as the vocal tract definitely *has* a stable and minimum phase transfer function and we therefore want to constrain our measurements to match systems of this kind.

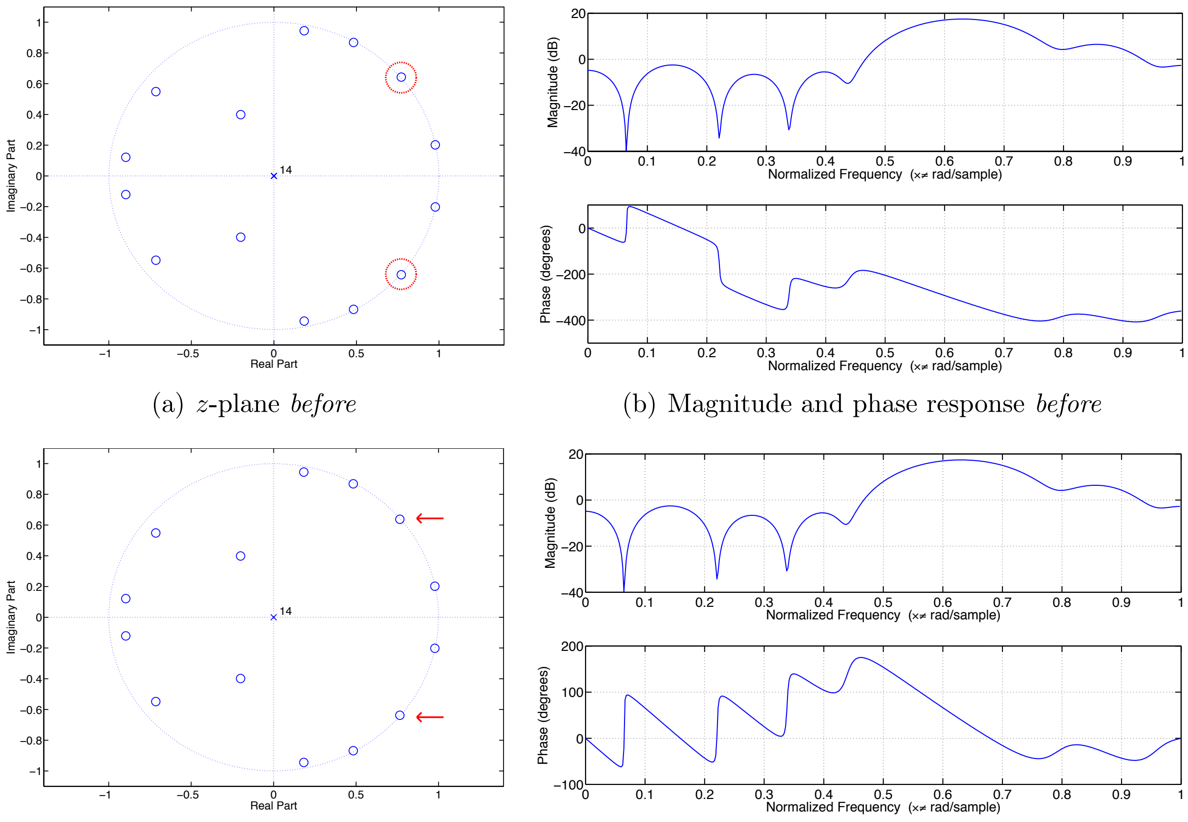
Root constraining involves two steps:

1. the roots of the estimated VTR $\hat{H}(z)$ are solved and any root $r_k(z)$ that has a radius $|r_k(z)| > 1$ it is replaced by its inverse conjugate $1/r_k^*(z)$. Graphically spoken it is *mirrored inside* the unit circle.

2. any *positive real* roots are removed directly as they correspond to a DC component of the filter that should not be a part of the system function. Another motivation for removing positive real roots stems from the fact that there is a possibility of the differentiator (that has a positive real root) of the lip radiation effect being included in the VTR. This would not be directly problematic if we knew that in *every* VTR estimate there is the differentiator included. As we primarily try to compute *only* the effects of the vocal tract and not the effects of sound propagation in the larynx we decided to prevent any VTR from including the lip radiation effect by removing positive real roots.

Figure 4.4 illustrates the effect of root constraining. In 4.15(a)-(b) you see a VTR estimate that is composed by conjugate complex roots (which matches the model properties of the vocal tract model). Two of these roots are located slightly outside the unit circle ($r = 1.043$). Though the magnitude response lies in a reasonable range of values, the phase response shows clearly *non minimum phase* properties with a total phase shift of approximately 400 degrees and an average group delay $\langle d\varphi/df \rangle = -0.0123$.

In figures 4.15(c)-(d) the roots have been mirrored inside the unit circle by the proposed relation ($r = 0.9982$). Comparing the two magnitude responses no variations can be seen but the effect on the phase response is significant. Firstly, the maximum phase shift is



(a) z-plane *before*

(b) Magnitude and phase response *before*

(c) z-plane *after*

(d) Magnitude and phase response *after*

Figure 4.15: Comparison of z-plane roots illustration and magnitude/phase response of the inverse VTR estimate before and after constraint is applied; (a) shows two roots outside the unit circle ($r = 1.043$, $\varphi = \pm 0.2039\pi$) and the corresponding magnitude/freuqency response in (b); (c) and (d) show the corresponding illustrations of the corrected system function ($r = 0.9982$): note that the magnitude response has not changed but the phase response shows minimum phase property and a very low average group-delay ($\langle d\varphi/df \rangle = 2.5500e-05$).

reduced to approx. 180 degrees and the average group delay of the system becomes quasi zero ($\langle d\varphi/df \rangle = 2.5500e - 05$).

5 Modification Of Vibrato

5.1 Introduction

Modifying singing voice vibrato may not seem a very complex task on the first hand. There is a wide range of pitch modification algorithms already available of which many also take into account the source-filter nature of human voice production.

Nevertheless, to the best of our knowledge, none of the existing algorithms tries to perform the desired modifications on a basis that corresponds to the physical basis of vibrato production.

Aim of this work is using the existing models of speech production to gain access to the fundamental parameters of voice production which are equally valid for singing and *actively* take control over those parameters in the way an actual singer would do.

In general, there are two possible scenarios we want to investigate in this work:

- *create vibrato* where there is none

  or conversely

- *remove vibrato* so only a static pitch remains.

Both tasks presume a priori knowledge of the original pitch contour and then perform the desired modifications. The actual algorithm that enables us to perform these modifications will be described the later sections of this chapter.

Before we come to this point we want to consider what the two tasks of vibrato modification may look like in detail and what will be necessary to execute them.

The motivation to actively influence vibrato occurrence lies in the field of studio and post-production where the singer's vocal tracks have already been recorded and no further changes in the musical material can be performed by the artist himself or herself.

5.1.1 Generating Artificial Vibrato

Now let us consider following situation: a singer of classical voice has recorded an aria during a past studio session and while editing and mixing the studio engineer decides that a certain phrase could deserve some more vibrato. Of course changing the musical expression of a singer is a very delicate matter and many musicians might refuse such a step in the first place. Nevertheless, we want to provide an outlook what *at least could* be achieved by means of signal processing.

Now this singer has recorded an arpeggio of an A7 chord as illustrated in figure 5.1 which contains a very long sustained note on the septime that exposes a very static pitch.

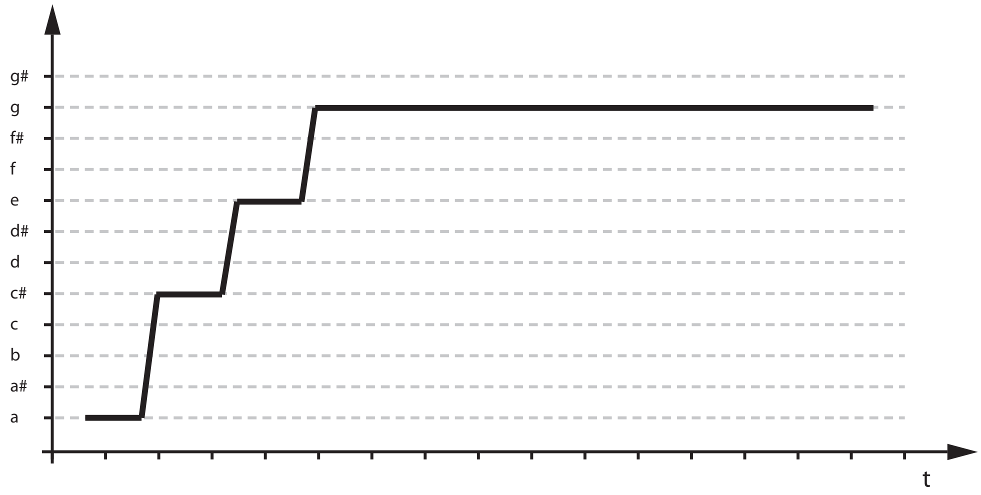


Figure 5.1: Pitch contour of an A7 chord sung as arpeggio

Unsatisfied by the musical expression of the singer, the engineer decides to add a quantum of vibrato to this specific note which might result in a pitch contour as depicted in figure 5.2.

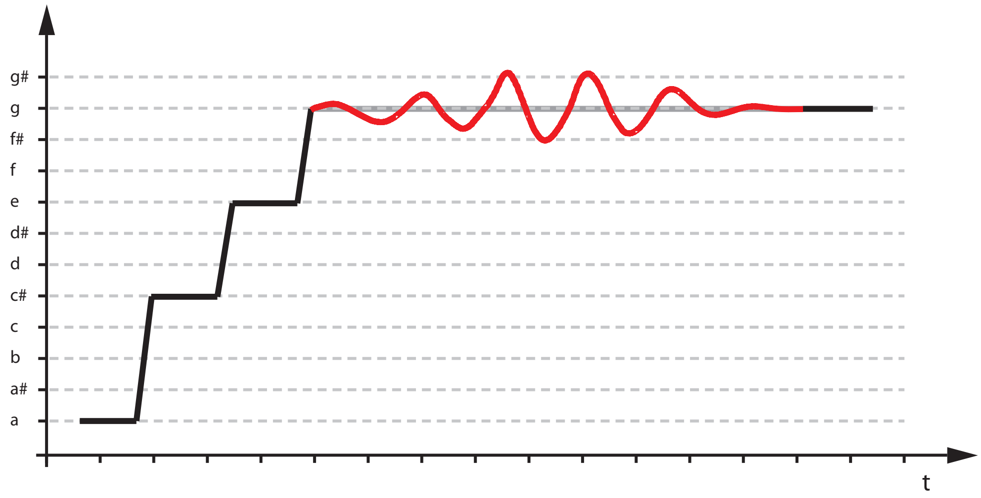


Figure 5.2: Pitch contour of an A7 chord sung as arpeggio with artificial vibrato on the sustained note

In this case, the temporal evolution of the vibrato extent is a function of time whereas the rate and the waveform can remain approximately unchanged. This leads us to a set of desired *controll parameters.*

Control Parameters

1. *Vibrato Extent* v_e: seems reasonable as primary control parameter as this is what a singer adjusts in the first place i.e. he decided whether to sing vibrato or not. A suitable interval for this parameter could be e.g.

   $$-\delta_{st}(1) < v_e < \delta_{st}(1) \tag{5.1}$$

   where $\delta_{st}(x)$ denotes a pitch of x *semitones* away from the original pitch f_0 defined as

   $$\delta_{st}(x) = f_0 2^{(12+x)/12} \tag{5.2}$$

   The vibrato extent should be zero by default and only $\neq 0$ if the user desires active modification.

2. *Vibrato Rate* v_r: can remain relatively static as the variety across different singers is not wide spread. Most singers have a vibrato rate between 6-8Hz. Nevertheless, the user should be given access to this parameter

3. *Vibrato Waveform and Regularity* are the two less interesting parameters, as they do not change significantly - or at least a well defined singing vibrato should not expose severe changes in regularity and its waveform. Therefore those two can be fixed sinusoidals or definable through templates.

5.1.2 Cancelling Existing Vibrato

In contrast to creating artificial vibrato one might also have the desire to *remove* vibrato from a recording. For example the pitch contour displayed in figure 5.3 has been taken from a baritone recording of the aria “Mein teurer Heiland, lass Dich fragen” of J.S. Bach’s *St. John’s Passion.* The professional singer has made extensive use of his vibrato to enrich the german word “*freut*” [frəit] sung over six notes of a major scale (note that the original recording was not in A major but the contour has been adapted to fit this scale). Figure 5.3 and 5.4 only display the voiced sounds of this word: the voiced [r] at the beginning is followed by an [ə], an [i]. Neither the fricative [f] nor the plosive [t] do appear.

Taking a look at figure 5.3 makes clear that the actual pitch that has been sung is only very hardly recognizable - at least from the pitch representation alone. In figure 5.4 we tried to find an appropriate alignment to illustrate the actual pitch as is would have been produced by e.g. a virtual string instrument driven by MIDI input.

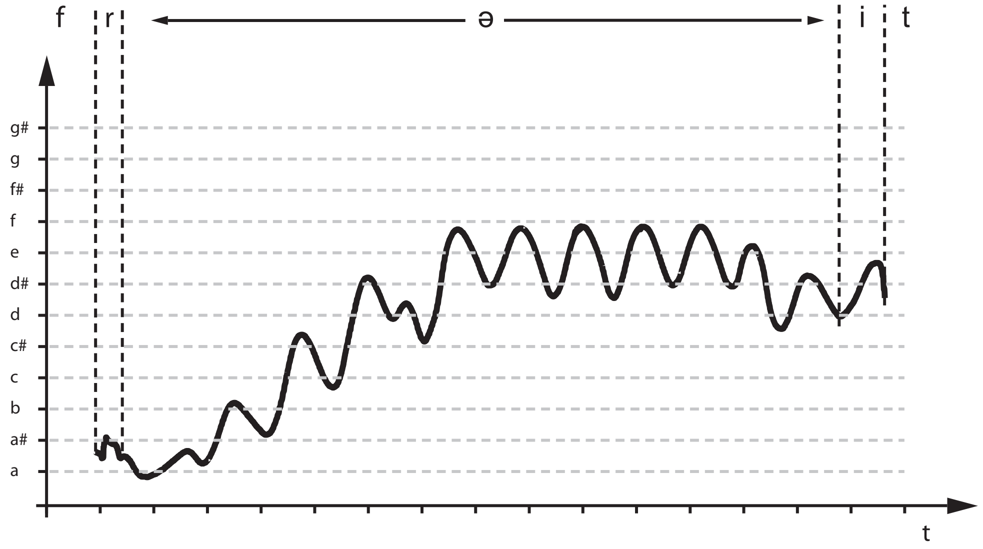


Figure 5.3: Pitch contour of the german word *"freut"* sung over six tones of a major scale and the alignment of the phonemes

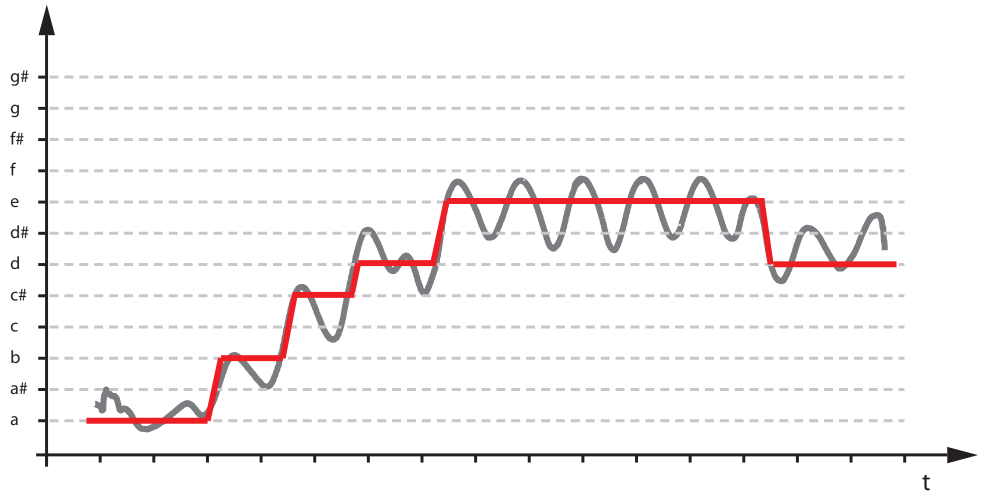


Figure 5.4: Pitch contour of the german word *"freut"* sung over six tones of a major scale and the theoretical pitch (red)

Aim of the algorithm shall be a method that allows for reducing the vibrato for a certain segment. Obviously, this is the more complex task compared to creating vibrato. The algorithm has to estimate a *target pitch* that is not available from either a user input or a score.

The task of pitch tracking though - or rather *pitch matching/transcription* to the tempered scale including rhythmic information - is a very complex task and has been subject to a wide range of scientific research. Thus, we decided not to include an entire algorithm of this kind in this work but instead focussed our interest on finite areas of long sustained notes, where the target pitch $f_{0,target}$ can be assumed to be *quasi stationary* and can be defined as the *mean pitch* in the time interval $[N_0, N_1]$ of the sustained note.

$$f_{0,target} = f_{0,mean} = \frac{1}{N_1 - N_0} \sum_{n=N_0}^{N_1} f_0[n] \tag{5.3}$$

Hence, the deviation of the instantaneous pitch $f_0[n]$ from the target pitch $f_{0,target}$ in the defined time interval $[T_0, T_1]$ leads to the correction term $\delta_{f_0}[n]$ defined as

$$\delta_{f_0}[n] = f_{0,target} - f_0[n] \tag{5.4}$$

Defining the *"vibrato cancellation extent"* v_{ce} as

$$0 < v_{ce} < 1 \tag{5.5}$$

and making it a function of time n, we can use it as a linear weight to the correction term δ to compute the new pitch $f_{0,new}$ as

$$f_{0,new}[n] = f_0[n] + \underbrace{v_{ce}[n]\delta_{f_0}[n]}_{\text{compensation term}} \tag{5.6}$$

where $v_{ce} = 0$ leads to a compensation term of 0 and therefore introduced no change in pitch.

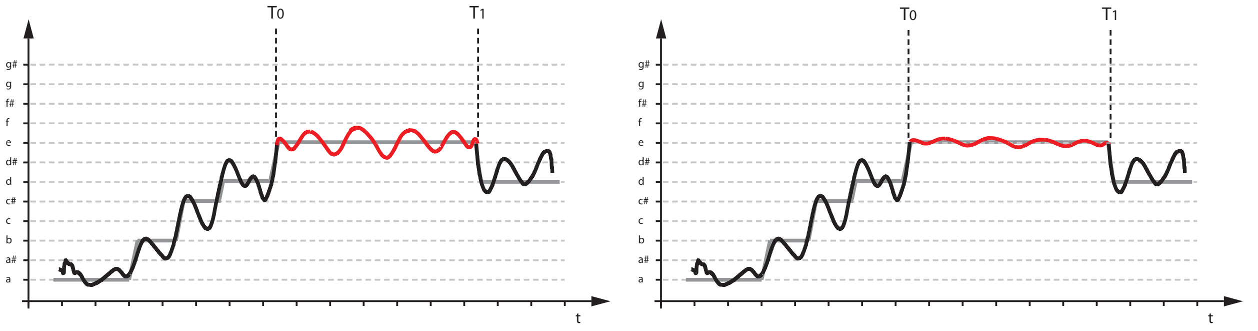
If v_{ce} has a value of 1 instead, we can assume that

$$f_{0,new}[n] = f_0[n] + \delta_{f_0}[n] = f_0[n] + f_{0,target} - f_0[n] = f_{0,target} \tag{5.7}$$

and only the static pitch of $f_{0,target}$ remains.

Two illustrations of weak and strong vibrato cancellation are provided in figures 5.5(a) and (b) where (a) shows the effect of moderate vibrato cancellation or rather vibrato *reduction* whereas (b) shows the result of nearly complete vibrato *equalization.*

The next sections will present in detail the algorithm we have come up with that allows for executing the proposed modifications to singing voice vibrato.



(a) Moderate vibrato reduction

(b) Severe vibrato cancellation (vibrato *equalization*)

Figure 5.5: Vibrato Cancellation

5.2 General Algorithmic Thoughts

As described in the previous sections 2.6 and 2.8.4 we have discovered that the actual pitch period variation seems to be produced in the end of the glottal open phase. Using this knowledge we have developed an algorithm that enables us to perform modifications focussed on this time interval.

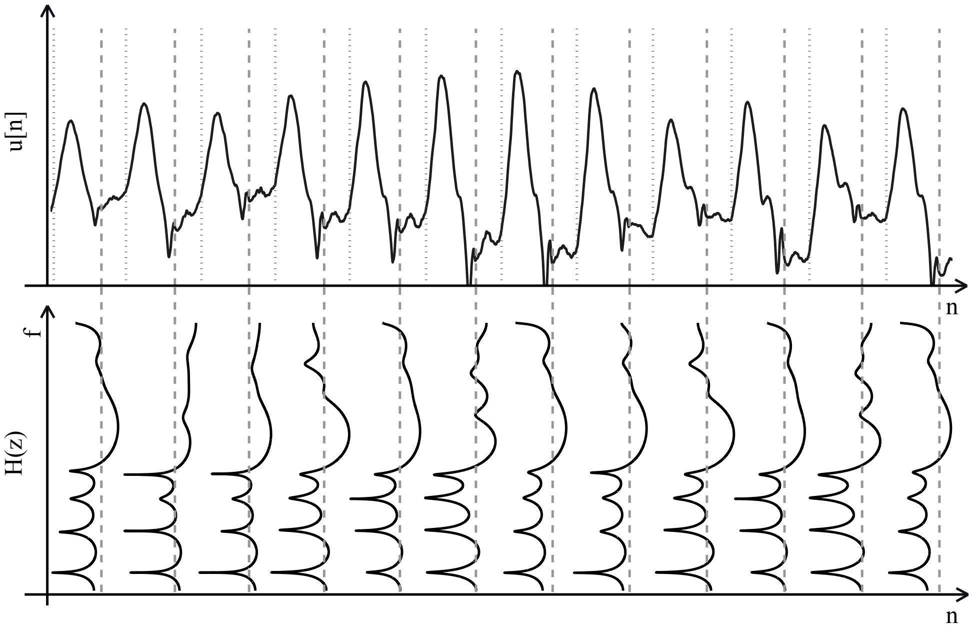


Figure 5.6: Pitch aligned representation of glottal source signal $u[n]$ (upper graph) and the vocal tract response $H(z)$ (lower graph)

In the first place we have to make sure to compute an accurate estimate of the glottal volume velocity waveform that corresponds to the air flow through the vocal folds. This is implemented in the *constrained close phase covariance glottal inverse filtering* algorithm presented in chapter 4. The resulting pitch synchronous representation of glottal source signal and vocal tract response can be imagined as illustrated in figure 5.6.

Now we can use the glottal source signal and the cycle-aligned representation as described in section 2.7 to compute the respective beginnings of the open, and consequently of the release phase. More important that the absolute positions of the instants within a single cycle is the *overall consistency* over a larger number of glottal cycles.

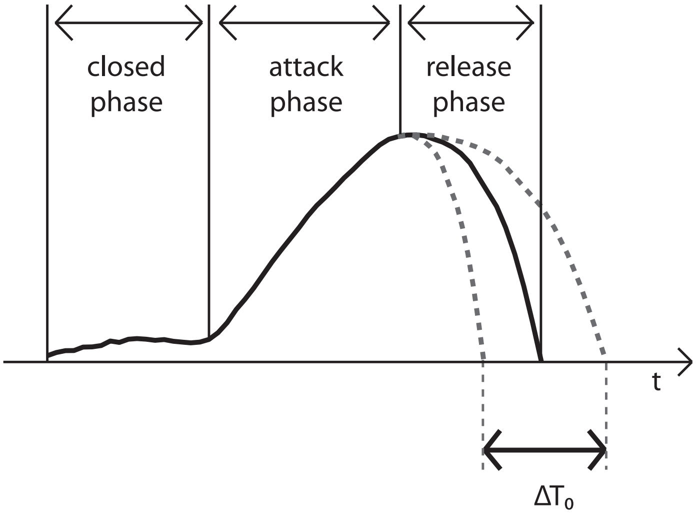


Figure 5.7: Illustration of one glottal source cycle and the time interval which is affected by the occurrence of vibrato indicated by ΔT_0. This can be described as a variation of the release phase and also corresponds to the vibrato extent. Cancelling or generating vibrato will be done by modifying this segment of cycle data

Figure 5.7 illustrates the idea of applying the desired modifications to a glottal source signal. The grey dashed lines indicate the approximate shape of a glottal cycle at the minimum and maximum pitch period occurring in a vibrato cycle. The solid black line resembles the possible *average* cycle form which is the starting point in the case of vibrato generation and, on the other hand, the desired result of vibrato cancellation.

5.3 Synthesis Benchmarks

In the course of this work we tried to investigate the possibilities of our algorithm regarding creating or cancelling vibrato. As the basis for both task lies in the same time-aligned representation of glottal source an vocal tract response we decided to focus on the *cancellation* of vibrato as it seemed the more challenging task.

In this section we want to present the possibilities and problems of cancelling vibrato with a practical implementation.

5.3.1 Example: Vibrato Cancellation

To perform vibrato cancellation we have to have a priori knowledge of the target pitch period. Naturally, this can be a function of time as well. In the case of our study this has been computed as the *average pitch* which seems reasonable concerning that vibrato is commonly regarded to be a regular, sinusoidal variation *around* a center pitch.

The algorithm involves the following steps:

1. segment glottal source signal into cycles according to the GCIs
2. upsample by integer factor (e.g. 4 in our case)
3. determine the beginning of the release phase
4. perform time variation by fractional resampling
5. downsample by the same integer factor
6. concatenate the sequential cycles to the modified glottal source

A schematic illustration of these steps is given in figure 5.8.

After the modified glottal source signal has been composed, the vocal tract transfer function is reapplied pitch synchronously by either the A-PSOLA technique described in section 3.5.2 or the time variant Lattice method presented in section 3.5.3.

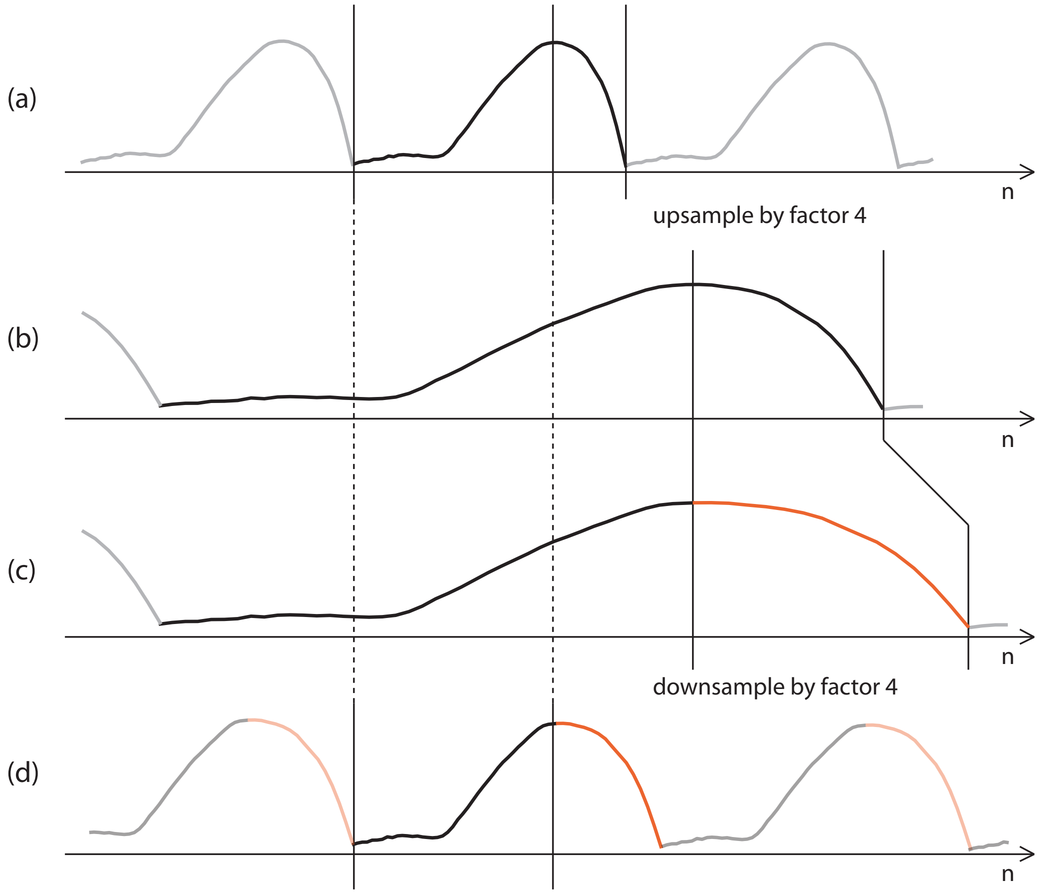


Figure 5.8: Schematic illustration of the algorithm to remove or generate vibrato; (a) the glottal source $g[n]$ is segmented into cycles; (b) after upsampling the beginning of the release phase is detected; (c) the release phase is modified according to the target pitch $f_{0,target}[n]$; (d) after downsampling the modified glottal source $\hat{g}[n]$ is concatenated

5.3.2 Results

We can modify the pitch in a manner that quasi only a static pitch remains. The statistic momentums to show this fact are given in table 5.1 whereas a graphic illustration of the original and the resulting pitch evolution is given in figure 5.9.

Figures 5.10 and 5.11 illustrate two methods of calculating the modified glottal source signal and the spectral effects on the resynthsized speech signal. More precisely, we want to illustrate the effect of constraining the modification window to the release phase of the glottal cycles. These effects can be seen on the one hand in the cycle-aligned representation and on the other hand, maybe yet more interestingly, also in the spectrograms of the synthesized speech signals. It has been shown that the spectral properties of the original

	mean μ	variance σ^2
original	$186.51Hz$	39.6270
modified	$186.42Hz$	0.1523

Table 5.1: Mean and variance of original and modified pitch of a 1s baritone recording

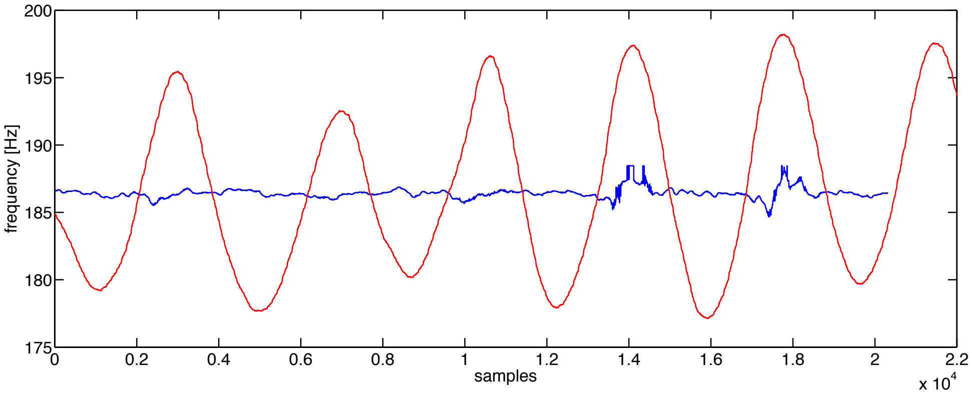


Figure 5.9: Pitch contour of original 1s baritone recording with strong variations in pitch (solid red line) and the achieved vibrato reduction (solid blue line); $f_s = 22050Hz$

signal could be conserved more precisely, especially regarding higher frequency bands around $6000Hz$.

5.3.3 Performance Issues

As already visible in figure 5.9 and even more clearly notable in figures 5.10 and 5.11 we encounter a number of undesired deviations from the ideally flat pitch contour. These artefacts are primarily related to

1. an invalid estimation of the VTR and the subsequent invalid glottal source signal or

2. problems that occur while performing inverse filtering or during resynthesis.

The exactitude of the VTR measurement is directly related to the quality of the recording.

For instance it has occurred to us that recordings that have *not* been recorded under studio conditions in a quasi anechoic chamber showed less appropriate results. This is probably due to the disproportional length of the room impulse response that definitely does not lie within the length of a glottal closed phase. Hence, the estimate of the VTR was corrupted by the room impulse response and the computed glottal source showed relatively strong fluctuations that could not be argued by subglottal coupling or similar laryngal effects.

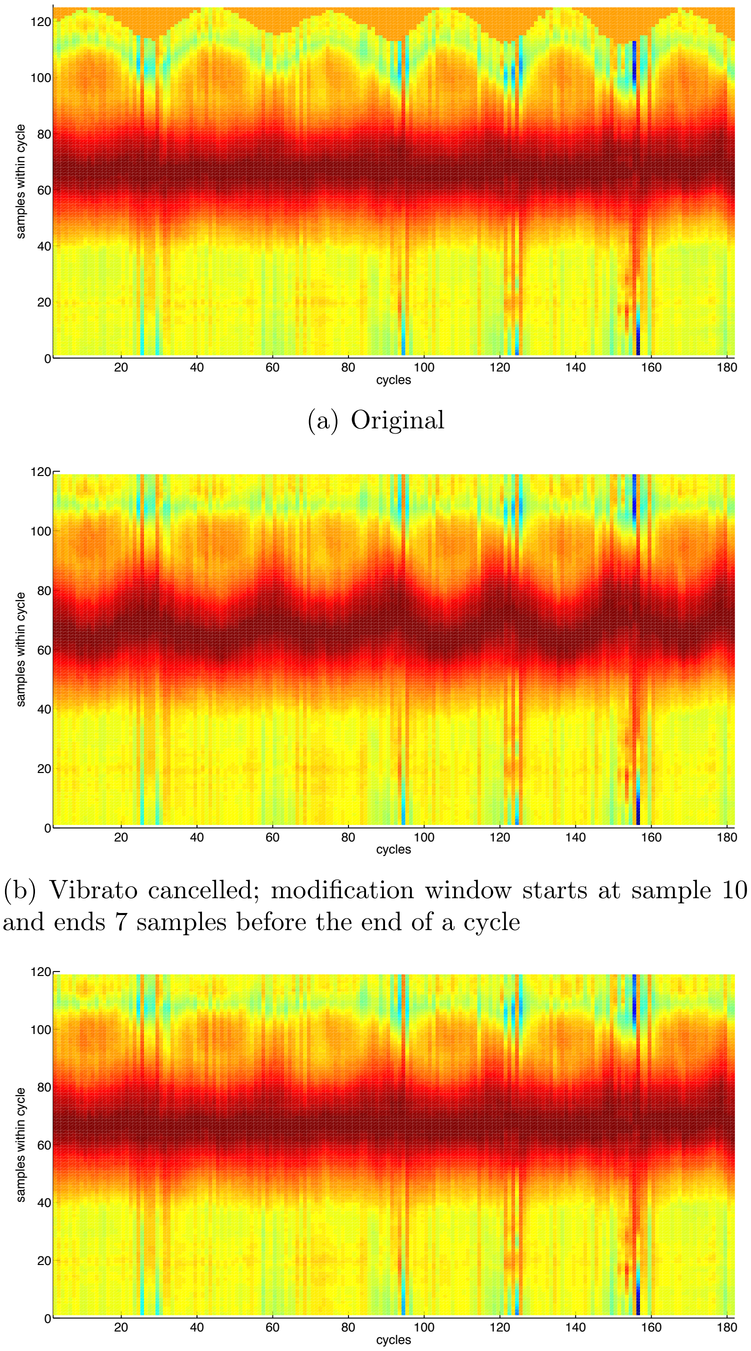
A-PSOLA

Although we tried to avoid inconsistencies by using hybrid Hann windows, we still fight with phase misalignments. Unfortunately, we were not yet able to resolve these problems in the course of this work.

LTV Lattice

The Lattice method yielded very promising results in the first place and as it is related so closely to the lossless tube model of speech it seemed a very suitable filtering technique for our task. Also guaranteed stability through the definition of the reflection coefficients seemed very promising.

Nevertheless, the sever effects of *coefficient interpolation* as discussed in section 3.5.4 corrupt the filter output intensively.

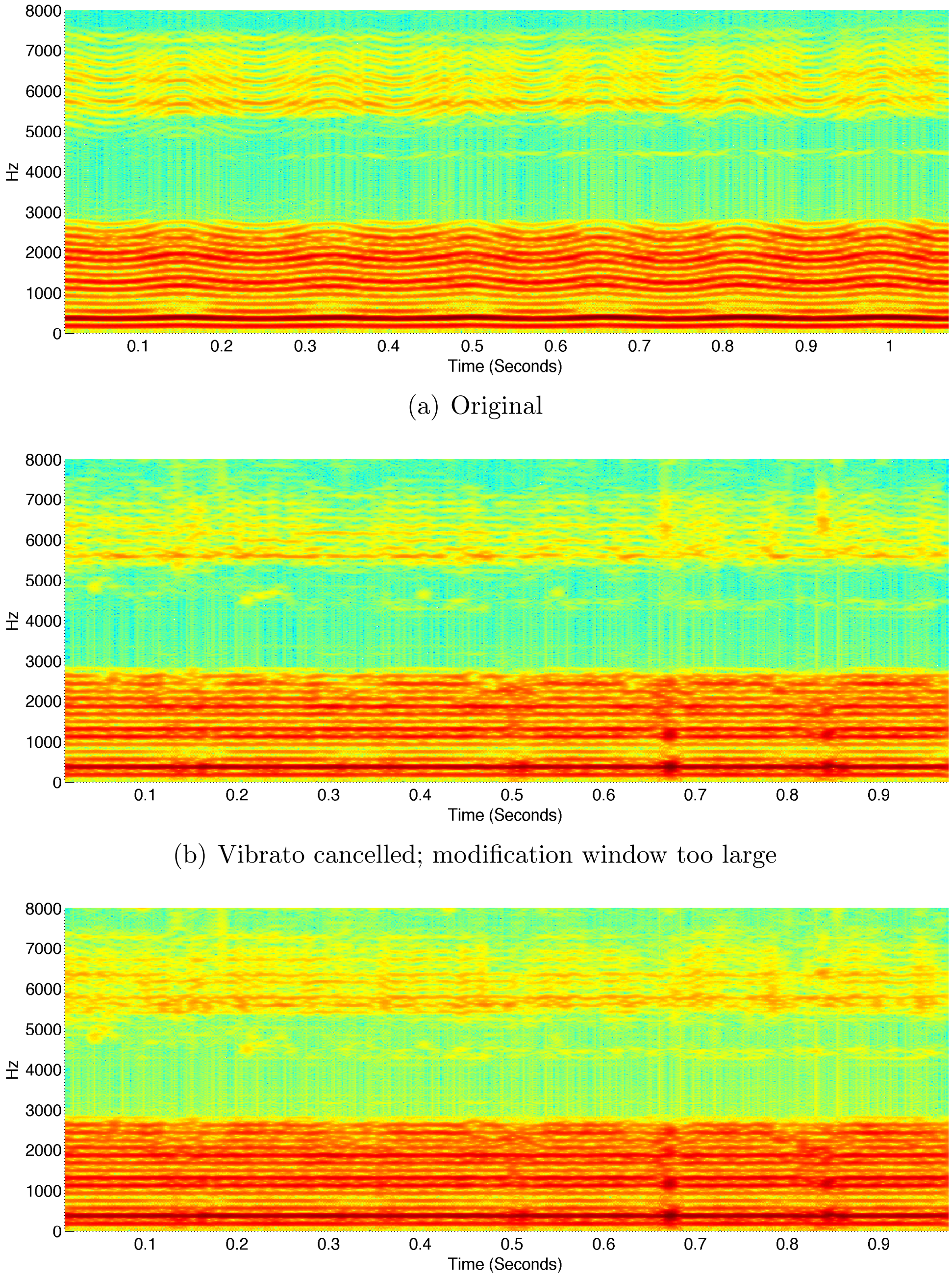


(a) Original

(b) Vibrato cancelled; modification window starts at sample 10 and ends 7 samples before the end of a cycle

(c) Vibrato cancelled; modification window starts at sample 72 and ends 7 samples before the end of a cycle

Figure 5.10: Comparison of two different window lengths of the modification interval. (a) shows the original glottal source signal of a 1s baritone recording; (b) shows modification regardless of the facts presented in the sections before; (c) modifications have been applied only to the release phase; note that the "shape of the red bar" indicating the attack phase of the glottal open cycle does not significantly change in contrast to (b)



(a) Original

(b) Vibrato cancelled; modification window too large

(c) Vibrato cancelled; modification window involves only the release phase

Figure 5.11: Spectral effect of the two cancellation windows described in figure 5.10(b) and (b); note that in the spectrogram of the original sample (a) you can see two relatively small formants around $6000 Hz$. If you compare (b) and (c) you can see that in the case of the specific modification of the release phase (c) the formant information is conserved much better than in the other case (b)

6 Conclusion

6.1 Summary

Focus of this thesis is the discussion of the contributing aspects of singing voice vibrato as well as procedures and methods to measure and modify these components.

In the first two chapters of this work we discuss the fundamental physical principles of voice production and sound propagation as well as two widely used models to describe these processes. Subsequently, having defined the descriptive parameters of the voice production models, we present the basic perceptual components of singing voice vibrato.

In chapter 2 we present a set of procedures to derive i.e. measure these parameters from real world audio signals. More precisely, we intensively discuss the measurement of the perceived pitch of a singing voice by presenting the YIN algorithm [11] while paying special attention on implementation issues as well as on an approach to resolve the octave error problem. We also present a straight-forward method to measure the effective amplitude of a signal and have elaborated an algorithm that allows for measuring the vocal tract transfer function and determining position and gain of the formants.

Additionally, we develop an intuitive way to visualize the internal relation of multiple glottal cycles by introducing the *cylce-aligned representation* as well as the relation between pitch, amplitude and the formant locations in the *time-aligned representation*. From these representations we derive a method to perform evaluation of the open quotient of the glottal source signal.

In a small empirical study we compare the temporal evolution of pitch, amplitude and formant frequencies as well as the evolution of the open quotient for a set of testfiles containing segments of static pitch and vibrato segments. The results and interpretation are discussed section 2.8.4.

In chapter 3 we present the mathematical basis of glottal inverse filtering that has been used to gain insight on both, spectral properties of the vocal tract and the processes inside the phonatory tract. In this context we pay special attention to two fundamentally different techniques of implementing linear time-variant filters: *asymmetric pitch-synchronous overlap-and-add (A-PSOLA)* and the *LTV Lattice filter.* For both techniques we discuss advantages and drawbacks considering performance issues.

Chapter 4 describes in detail the algorithm to perform blind deconvolution of the glottal source and the vocal tract response. The *constrained closed phase covariance glottal inverse filtering* approach allows for very accurate estimates of GS and VTR which is the necessary basis of any modification steps invoked later. Here we stress out the importance of exactly determining the instants of glottal closure and present an approach that theoretically allows for their quasi real time computation. Additionally, we introduce a novel technique to cross-check the validity of a GCI by ensuring maximum coherence of multiple neighbouring glottal cycles (*cycle prototyping).* A further post-processing step is introduced by *constraining root positions* which ensures the minimum-phase property and invertibility of the vocal tract transfer function.

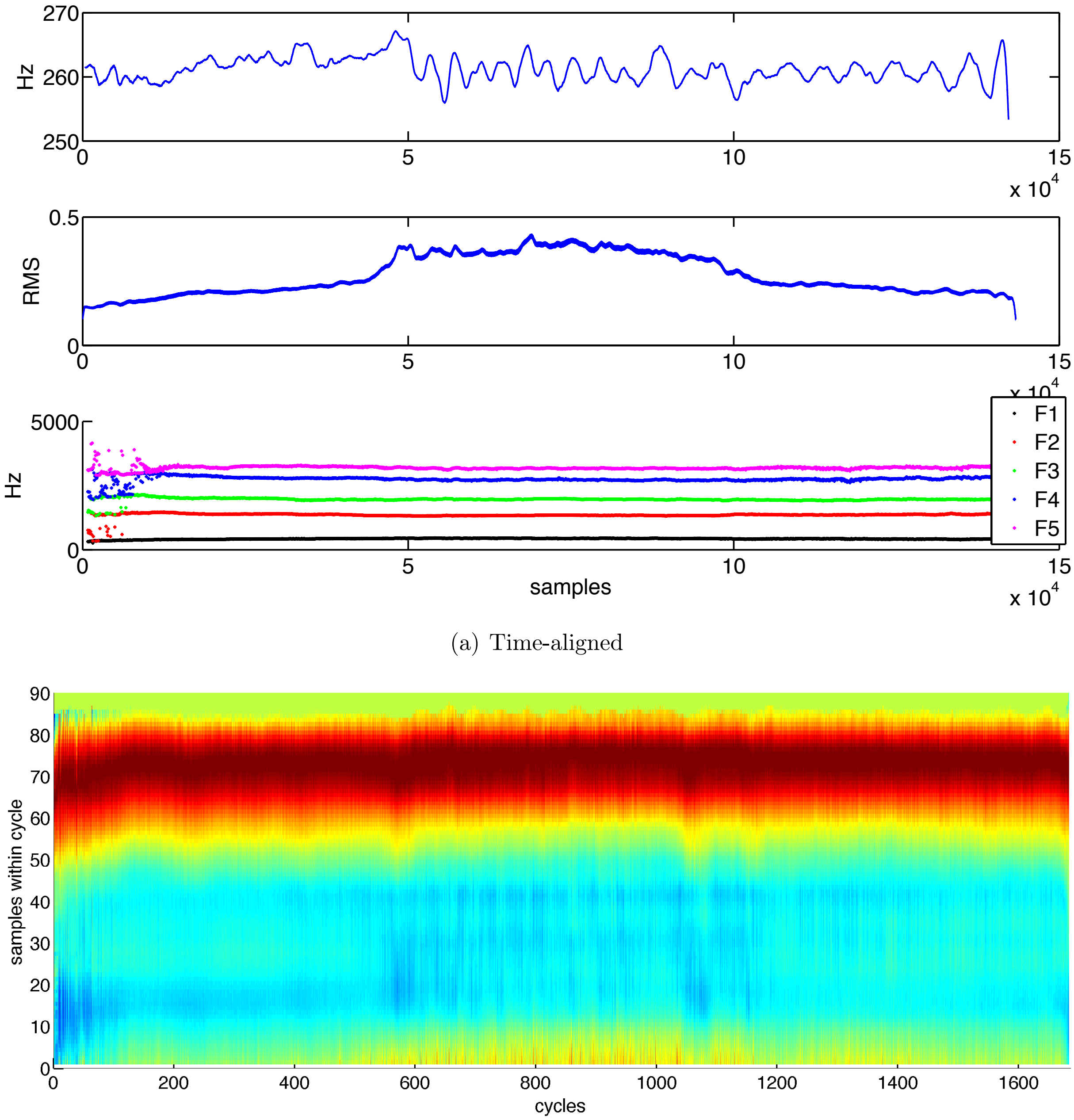
As the last part of this work we discuss the possibilities and requirements of modifying vocal vibrato in chapter 5. First we give a general overview on creating and cancelling vibrato by defining a set of control parameters. During research we have drawn a set of interesting observations and conclusions concerning the relation between glottal open and closed phase in segments of vocal vibrato that we also discuss in this chapter. Using the example of vibrato cancellation we exploit this new knowledge and present a novel approach to alter vibrato on a physically meaningful basis. Later we also discuss the computational issues and problems that we have encountered in the course of this work as well as the synthesis results we were able to achieve.

6.2 Outlook

Although high amount of attention has been paid to the accurate implementation of the synthesis techniques there is still a large potential of possible improvements. On the one hand, the A-PSOLA approach showed very promising results in the test benchmarks but in a real world scenario there is still a large quantity of sparse errors and glitches that are most probably related to low-frequency phase misalignments between overlapping synthesis windows. Investigating the phase evolution at these transient errors seems promising to understand the reason for these glitches.

The LTV Lattice filter on the other hand allows for absolutely perfect reconstruction of the original signal as long as no changes are applied. If modifications are applied though, the effects of coefficient interpolation as described in section 3.5.4 severely corrupt the synthesized signal. These effects are definitely worth further investigations and maybe there can be developed an interpolation algorithm that allows for conserving the formant frequency positions throughout the interpolation process.

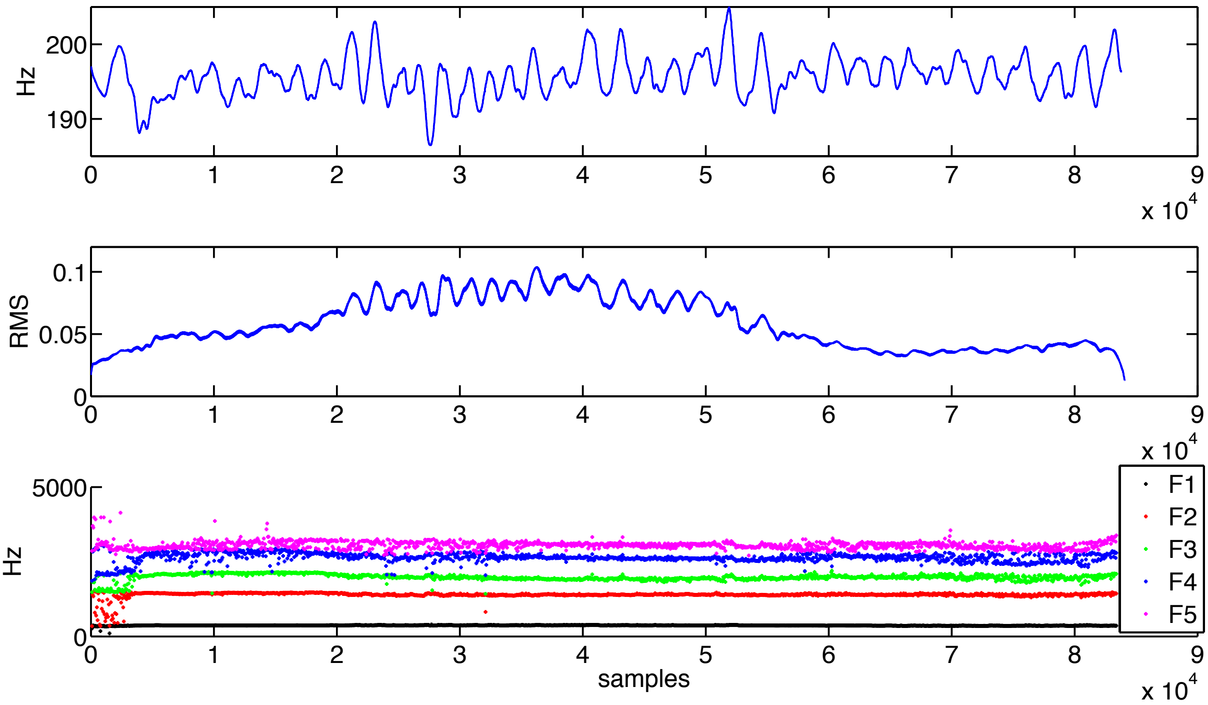
A Empirical Results



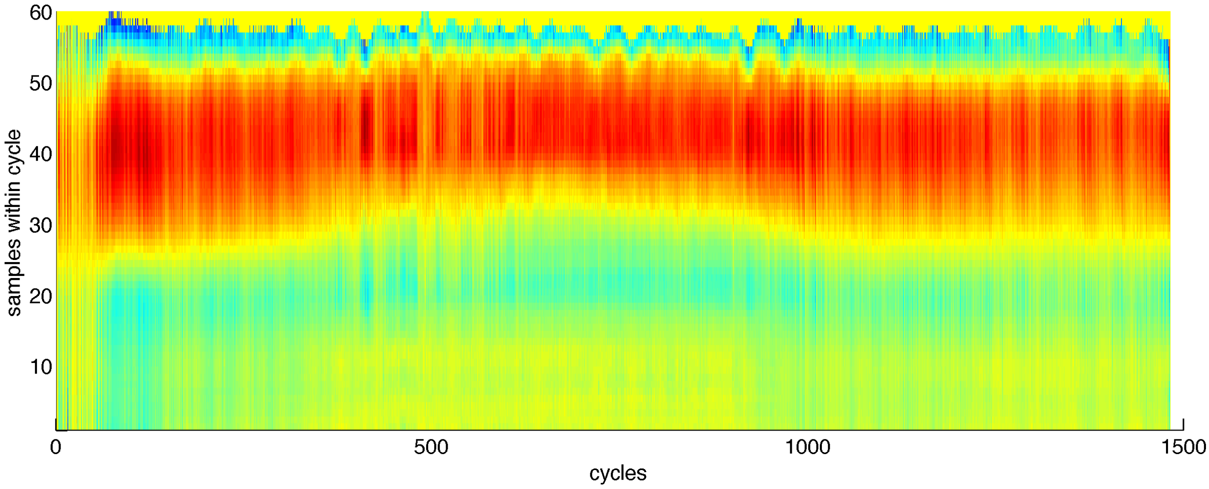
(a) Time-aligned

(b) GlottPlot

Figure A.1: Pitch, amplitude and formant tracking of sample `e_hoch.wav` ($f_s = 22050$) as well as corresponding glottPlot

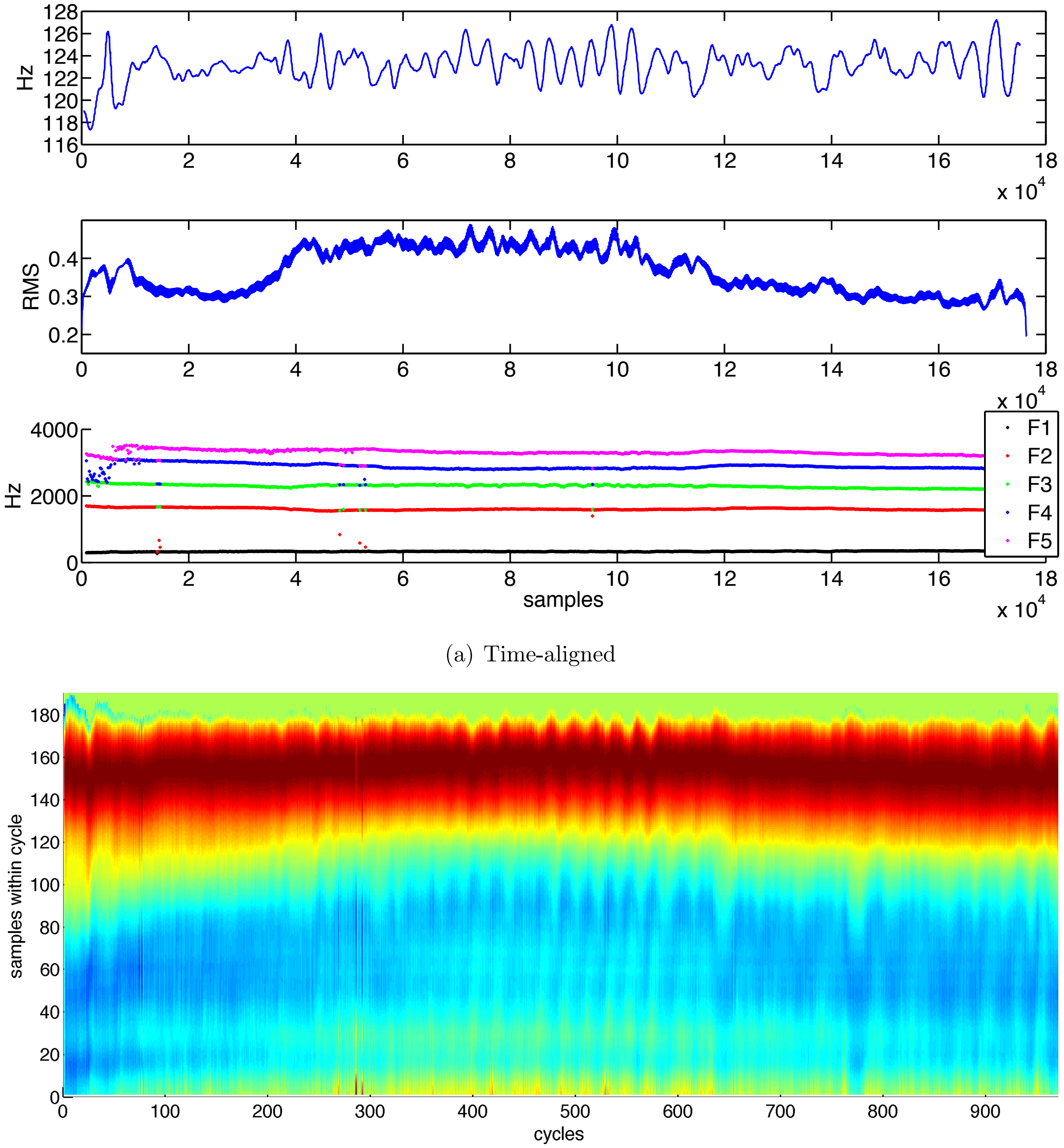


(a) Time-aligned



(b) GlottPlot

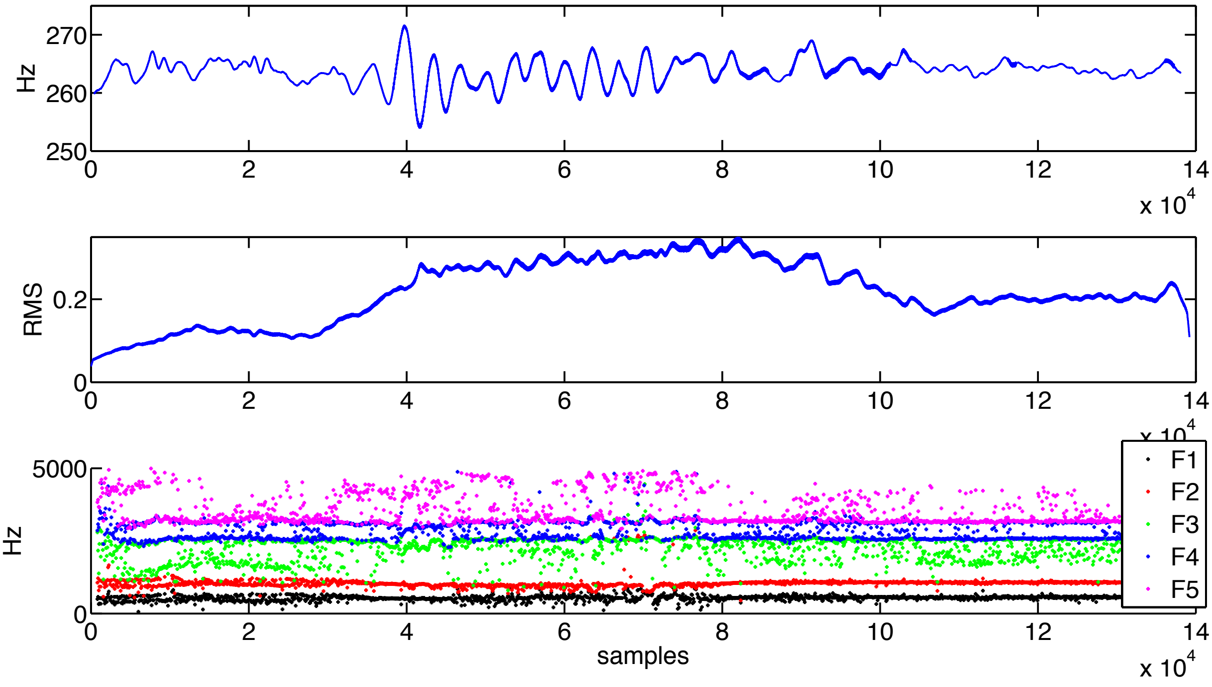
Figure A.2: Pitch, amplitude and formant tracking of sample `e_mittel.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot



(a) Time-aligned

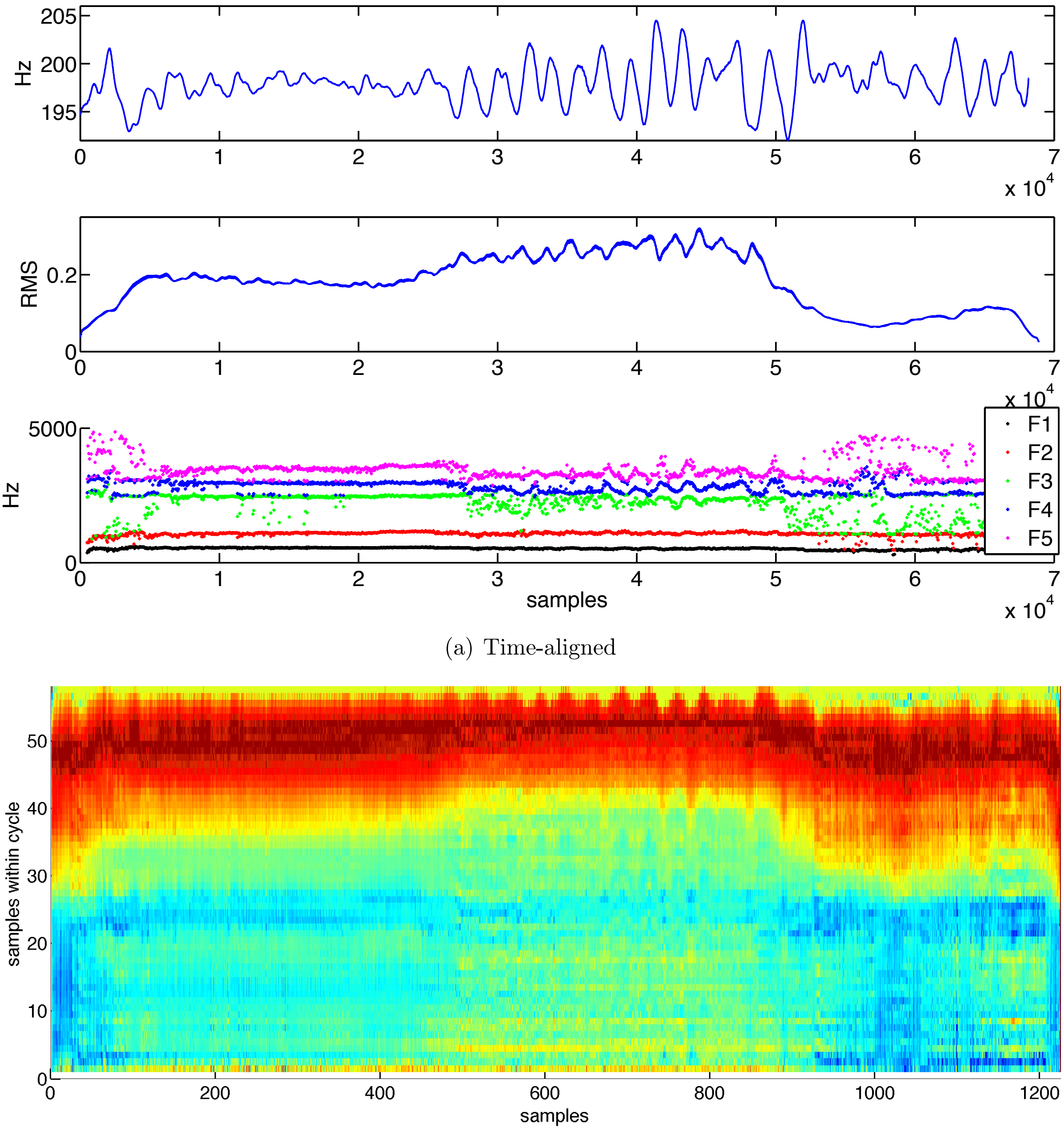
(b) GlottPlot

Figure A.3: Pitch, amplitude and formant tracking of sample `e_tief.wav` ($f_s = 22050$) as well as corresponding glottPlot



(a) Time-aligned

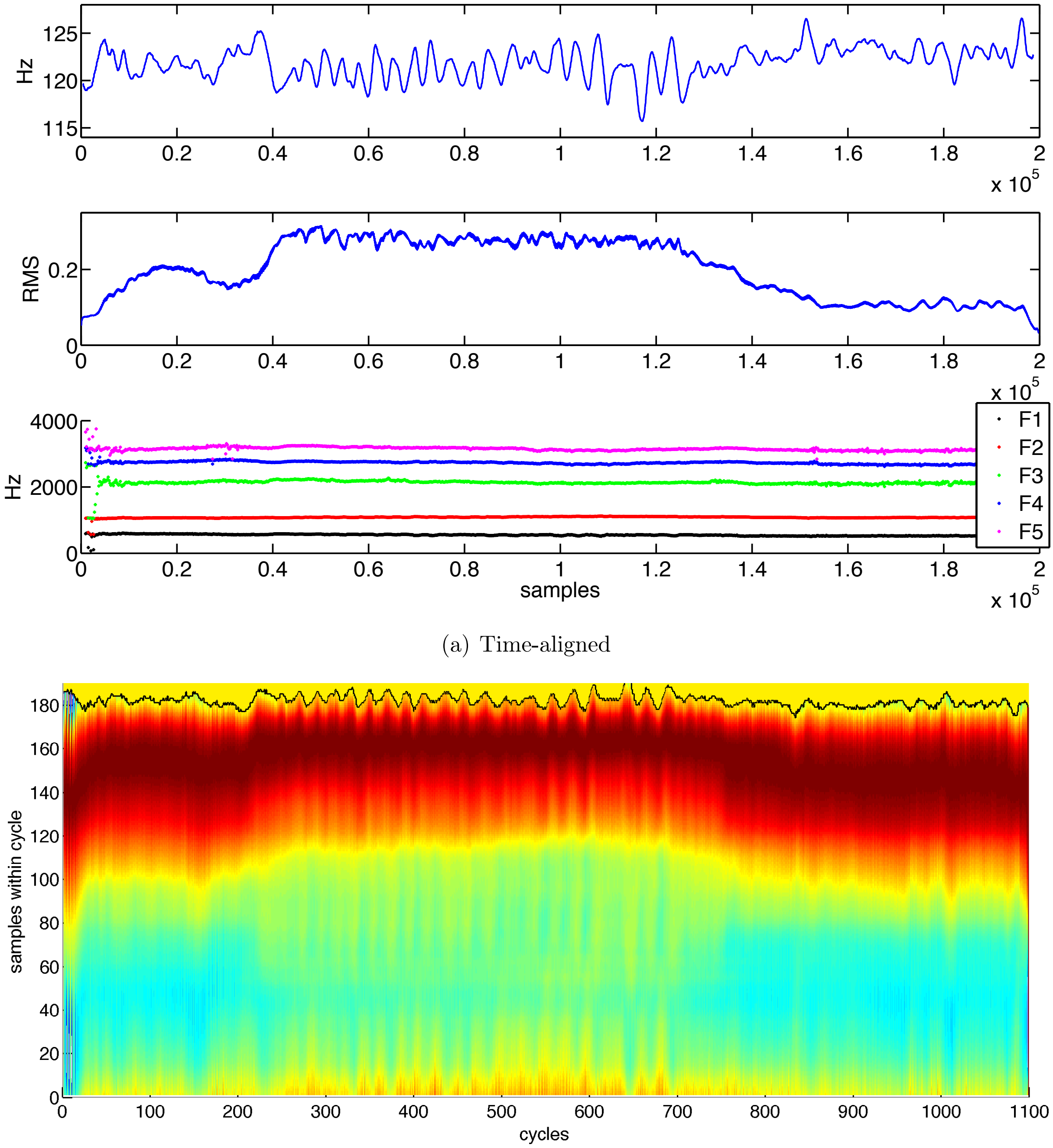
Figure A.4: Pitch, amplitude and formant tracking of sample `a_hoch.wav` ($f_s = 22050 Hz$). Unfortunately, due to the corrupted VTR estimates we could not compute a representative glottPlot of the glottal source signal.



(a) Time-aligned

(b) GlottPlot

Figure A.5: Pitch, amplitude and formant tracking of sample `a_mittel.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot



(a) Time-aligned

(b) GlottPlot

Figure A.6: Pitch, amplitude and formant tracking of sample `a_tief.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot

List of Figures

0.1 *Acknowledgements sorting randomizer*; the order of the list is derived from the distance of the respective acknowledgement atom to the origin 11

1.1 Schematized diagram of the vocal apparatus ([1] p.88) 15

1.2 Anatomical illustration of the larynx ([2] p.954) 15

1.3 Frontal section of the larynx; (a) vocal folds (*labium vocale*); (b) vocal cord (*ligamenta vocalia*); (c) vestibular fold (*plicae vestibulares*); (d) conus elasticus ([3] p.34) 16

1.4 The source-filter model of speech; the lungs press air through the vocal folds and according to their tension a periodic opening and closing $g[n]$ sets in (the left graph displays the volume velocity waveform); radiation at the vocal folds $\frac{d}{dn}g[n]$, henceforth $u[n]$, excites the vocal tract system $H(z)$ as displayed in center graph (the lip radiation can be approximation by deriving the volume velocity); the filtered vocal signal $s[n]$ leaves the mouth (right graph) 18

1.5 Glottal source spectral properties (above) and the effect of the vocal tract (solid grey line) on the speech signal (below); note that in the non-uniform case there is no immediately obvious relation between partial amplitudes of the speech signal and the vocal tract response 19

1.6 Volume velocity waveform; in the *closed phase (CP)* the vocal folds are closed and no air flows; at the *instant of glottal opening (GOI)* air flow sets in and ends at the *instant of glottal closure (GCI)*; this time segment is also called the *open phase (OP)* 20

1.7 Derivative of the volume velocity waveform; after radiation at the vocal folds this is the signal that excites the vocal tract; again *closed phase (CP)* and *open phase (OP)* are defined by the *instants of glottal opening (GOI) and closure (GCI)*; note that in the derivative the GCI corresponds to a well defined peak 21

1.8 An example of the vocal tract transfer function of the sung vowel [ə]. The first two formant resonances are located at $F1 \approx 350$Hz and $F2 \approx 1200$Hz 22

1.9 Concatenation of N lossless acoustic tubes ([1] p.214) 23

1.10 Physical parameters to describe vibrato [9] 25

2.1 Parameters of the voice production model; temporal and spectral shape of the glottal source are primarily defined by subglottal air pressure from the lungs and the vocal folds tension; vowel/consonant shaping results in a set of formant resonances with corresponding center frequencies and gains . . . 27

2.2 Flow chart of the YIN pitch estimator 29

2.3 Comparison of two definitions of the autocorrelation function: the upper figure shows the ACF according to eq. 2.1; the bottom figure shows the ACF according to 2.2 (taken from [11]) 30

2.4 Parabolic interpolation: candidate dip at iteration step k and its preceding and following values (blue). The interpolated value (red) is derived from the parabolic interpolation curve (green) 34

2.5 Stages I-IV of the YIN algorithm applied on a one second excerpt of a baritone recording; (a) autocorrelation matrix R_{xx}; (b) difference matrix D; (c) cumulative mean difference matrix D'; (d) limited matrix D'_{limit} . . 36

2.6 Pitch contour containing a large number of octave errors ($f_s = 44100Hz$) . 37

2.7 Detection function $\delta[n]$ for the occurrence of octave errors; the green lines indicate the beginning of defected data segments and the red lines the corresponding segment ends ($f_s = 44100Hz$) 38

2.8 Pitch evolution of an approx. 1s baritone recording ($f_s = 11025Hz$) of a sustained vowel [ə] with well educated vibrato 39

2.9 Removal of octave errors in a pitch contour; (a) shows the original pitch contour containing a large number of octave errors (taken from the sample `a_hoch.wav`, $f_s = 44100Hz$); (b) shows the detected octave error segments that have to be reassigned as indicated by the red dashed arrowa; the reassignment operation is depicted in (c); (d) shows the resulting pitch contour with the reassigned areas marked as red line segments 40

2.10 RMS level amplitude evolution of an approx. 1s baritone recording ($f_s = 11025Hz$) of a sustained vowel [ə] with well educated vibrato 41

2.11 Relationship between the first two formant frequencies for a set of (swedish) vowels [4] p.39 42

2.12 Example of tracking formants of an approx. $1s$ baritone recording ($f_s = 22050Hz$) of a sustained vowel [ə] with well educated vibrato; on the right side the temporal evolution ($f_s = 22050Hz$) and on the left side the statistic distribution of the contributing formants illustrated as mean and variance of each formant (solid lines) and the resulting gaussian mixture (dashed line); mean center frequencies [F1: $347.9Hz$, F2: $1209.0Hz$, F3: $1814.6Hz$, F4: $2137.5Hz$, F5: $2615.1Hz$], formant frequency variances [F1: $0.099Hz$, F2: $0.455Hz$, F3: $1.491Hz$, F4: $8.976Hz$, F5: $47.620Hz$] 43
2.13 Example for a time-aligned representation of instantaneous frequency f_0 (top), instantaneous amplitude (middle) and the corresponding formant estimates of the vocal tract transfer function (bottom) of a 1s baritone recording ($f_s = 11025Hz$) with well educated vibrato. 45
2.14 Construction of the cycle-aligned representation; the glottal source signal is segmented into single cycles that are aligned by the respective instant of glottal closeure 46
2.15 Example of a cycle-aligned representation for the volume velocity waveform of a $1s$ baritone recording of a sustained vowel [ə] with well educated vibrato ($f_s = 22050Hz$) 47
2.16 Cycle aligned representation and the detected GOI position for a 1s vibrato signal ($f_s = 11025Hz$) through edge detection by the Canny algorithm (solid black line) 49
2.17 Recording setup of the test set (of course the Brauner Phantom C looks significantly different than in the illustration) 50
2.18 Illustration of the relation between duration of the open and the closed phase of a glottal cycle during vibrato; note that the CP and the first, attack-like part of the OP do not alter in length while vibrato. The “release phase” in contrast is responsible for the variation in pitch period; data taken from a 1s baritone recording ($f_s = 22050Hz$) 53

3.1 Spectrogram of a short segment of vibrato sung by a baritone 56
3.2 Rough schematic of the Overlap-and-Add method 58
3.3 Rough schematic of the Overlap-and-Add method used for implementation of an LTV filter; the filter block (grey) changes only from block to block; for the duration of a single block it can be assumed to be linear time-*invariant* (LTI) though 60

3.4 Parameters of the voice production model; temporal and spectral shape of the glottal source are primarily defined by subglottal air pressure from the lungs and the vocal folds tension; vowel/consonant shaping results in a set of formant resonances with corresponding center frequencies and gains . . . 61

3.5 Simplified model of speech production 63

3.6 Windowing in the autocorrelation method (note that the index $\hat{n}$ corresponds to r in this work); (a) speech signal $s[n]$ and Hamming window positioned at time $n = \hat{n}$; (b) windowed segment $s_{\hat{n}}[m]$; (c) segment of prediction error signal, $e_{\hat{n}}[m]$, over the range $0 \leq m \leq L - 1 + p$ obtained using an optimum predictor with $p = 15$ ([1] p.494) 68

3.7 Windowing in the covariance method (note that the index $\hat{n}$ corresponds to r in this work); (a) speech signal $s[n]$ and rectangular windows positioned over the range $\hat{n} - p \leq n \leq \hat{n} + L - 1$; (b) windowed segment, $s_{\hat{n}}[m]$, defined over a range $-p \leq m \leq L - 1$; (c) segment of prediction error signal $e_{\hat{n}}[m]$, defined over the range $0 \leq m \leq L - 1$ using an optimum predictor with $p = 15$ ([1] p.498) 69

3.8 Signal flow graph of an FIR lattice filter to calculate the *forward* prediction error $e[n]$ 71

3.9 Signal flow graph of an IIR lattice filter to calculate the speech signal $s[n]$. 71

3.10 Shape of a hybrid Hann window $w[n]$; n_c indicates the center sample of the window ($w[n_c] = 1$) 73

3.11 Overlapping of various non uniform length hybrid Hann windows 74

3.12 Log-magnitude response of a hybrid Hann window ($L = 301$, $f_s = 11025Hz$) with varying position of center ($y = 0$ window center lies at sample 150; $y = -50$ window center lies at sample 100, etc.) 75

3.13 Flow graph of an LTV lattice filter; $k_i^{(n)}$ denotes the i-th filter coefficient at the corresponding time step n 76

3.14 Signal path of a $p = 4$ order all-zero lattice filter with *filter state memory* ($\boldsymbol{\nu}_n$ and $\boldsymbol{\theta}_n$) 78

3.15 Signal path of a $p = 4$ order all-pole lattice filter with *filter state memory* ($\boldsymbol{\nu}_n$ and $\boldsymbol{\theta}_n$) 79

3.16 Frequency response of the Lattice filter coefficients for 300 samples of a time series; the boxes indicate the areas where the coefficients are linearly interpolated (in the open phase); the additionally *red* colored boxes mark the areas where parameter interpolation has introduced significant distortion on the frequency response. 81

3.17 Comparison of the spectral fitting properties four LPC approaches; length of CP analysis frame $N = 45$, LPC order $p = 36$; note that the blue (original) and the black (lpccovar) graph overlay nearly perfectly 83

3.18 Comparison of the performance of four LPC techniques in the temporal domain by means of inverse filtering; note that in subfigure 3 the red and the blue graph coincide ($f_s = 11025Hz$) 84

3.19 Generalized flowchart of the entire algorithm 84

3.20 Illustration of the (a) time domain error signal and the corresponding error spectrum using the LTV Lattice method (green marks indicate GCIs, red marks indicate GOIs); (b) depicts the spectral properties of the error signal; note that it generally has very low amplitudes and that it is not completely white but rather exposes a VTR like distribution 85

4.1 Flowchart of the constrained closed phase covariance glottal inverse filtering algorithm used to perform deconvolution of the glottal source signal $g[n]$ and the vocal tract response $H(z)$ 87

4.2 Schematic view on an EGG instrument and waveform [17], p. 808 88

4.3 Relation of impulsiv event and average group delay of a minimum phase system; a dirac impulse is moved from sample 1 to 150 through a vector of length $L = 150$ and the corresponding average group delay is computed under different conditions: (a) no noise; (b) -40dBFS additive white gaussian noise (AWGN); (c) -20dBFS AWGN; (d) 0dBFS AWGN; note that the negative going zero crossing indicating that the inpuls is located at the center of the window remains clearly visibly also under relatively noisy conditions 91

4.4 Variation of d_{DC}, d_{AV}, d_{EW} and d_{EP} as the signal-to-noise ratio (SNR) varies from -30 to +30dB for an input consisting of a single impulse at $n_0 = 20$ with additive white Gaussian noise in a window length of $N = 101$. For each measure, the graph shows the median value of d_* and the upper and lower quartiles. [20] 94

4.5 Values of d_{DC}, d_{AV}, d_{EW} and d_{EP} for a signal containing impulses at samples 0 and 40 of amplitude $1 - a$ and a respectively. The window length is 101 and a varies between 0 and 1. [20] 95

4.6 (a) Impulse train with a dominant period of 100 samples and an SNR of 10dB. (b)-(e) the waveform of d'_{EP} for different window lengths, N. The circles mark the negative-going zero crossings (NZCs). [20] 96

4.7 Initial GCI candidate set containing a vast number of false positives; LPC order $p = 10$; computed from a 1s baritone recording ($f_s = 11025Hz$) . . . 97

4.8 Various combinations of candidates and the resulting cost for candidate k (blue circle); excerpt of the calculation of the dynamic cost matrix $\mathbf{C}(k)$ according to eq. 4.27; initial set consists of true positives (green asterisk) and false positives (red asterisk); last example combination would have minimum cost value in $\mathbf{C}(k)$ 101

4.9 Example of entire algorithm: negative-going zero crossings of group-delay measure (black) indicate GCI candidates (red); note the spurious overdetections at ca. samples 3000 and 3530; resulting candidates (blue) after post-processing with corresponding final cost (green/dashed); computed from a 1s baritone recording ($f_s = 11025Hz$) 103

4.10 Example of 10th order LPC leading into vast overdetection of candidates (red). After optimization a set of GCIs (blue) remains that exposes high periodicity; computed from a 1s baritone recording ($f_s = 11025Hz$) 104

4.11 Set of GCIs surrounding a pitch contour computed from a 1s baritone recording ($f_s = 11025Hz$); in (a) the GCIs (blue circles) should lie as closes as possible to the pitch contour (solid red line); (b) shows the respective deviation from the pitch in samples 105

4.12 Cycle prototype; the blue lines correspond to the overlaid plotting of the involved glottal cycle; the red line resembles the prototype determined by averaging ($f_s = 11025Hz$) 107

4.13 GlottPlots of the original and the corrected set of GCIs based on the raw speech audio data; note that in the right figure the “blur” has been significantly reduced 107

4.14 Resulting GCI positioning after cycle prototyping ($f_s = 11025Hz$) 108

4.15 Comparison of z-plane roots illustration and magnitude/phase response of the inverse VTR estimate before and after constraint is applied; (a) shows two roots outside the unit circle ($r = 1.043$, $\varphi = \pm 0.2039\pi$) and the corresponding magnitude/freuqency response in (b); (c) and (d) show the corresponding illustrations of the corrected system function ($r = 0.9982$): note that the magnitude response has not changed but the phase response shows minimum phase property and a very low average group-delay ($\langle d\varphi/df \rangle = 2.5500e - 05$). 110

5.1 Pitch contour of an A7 chord sung as arpeggio 113
5.2 Pitch contour of an A7 chord sung as arpeggio with artificial vibrato on the sustained note 113
5.3 Pitch contour of the german word *"freut"* sung over six tones of a major scale and the alignment of the phonemes 115
5.4 Pitch contour of the german word *"freut"* sung over six tones of a major scale and the theoretical pitch (red) 115
5.5 Vibrato Cancellation 117
5.6 Pitch aligned representation of glottal source signal $u[n]$ (upper graph) and the vocal tract response $H(z)$ (lower graph) 117
5.7 Illustration of one glottal source cycle and the time interval which is affected by the occurrence of vibrato indicated by ΔT_0. This can be described as a variation of the release phase and also corresponds to the vibrato extent. Cancelling or generating vibrato will be done by modifying this segment of cycle data 118
5.8 Schematic illustration of the algorithm to remove or generate vibrato; (a) the glottal source $g[n]$ is segmented into cycles; (b) after upsampling the beginning of the release phase is detected; (c) the release phase is modified according to the target pitch $f_{0,target}[n]$; (d) after downsampling the modified glottal source $\hat{g}[n]$ is concatenated 120
5.9 Pitch contour of original 1s baritone recording with strong variations in pitch (solid red line) and the achieved vibrato reduction (solid blue line); $f_s = 22050Hz$ 121

5.10 Comparison of two different window lengths of the modification interval. (a) shows the original glottal source signal of a 1s baritone recording; (b) shows modification regardless of the facts presented in the sections before; (c) modifications have been applied only to the release phase; note that the "shape of the red bar" indicating the attack phase of the glottal open cycle does not significantly change in contrast to (b) 123
5.11 Spectral effect of the two cancellation windows described in figure 5.10(b) and (b); note that in the spectrogram of the original sample (a) you can see two relatively small formants around $6000Hz$. If you compare (b) and (c) you can see that in the case of the specific modification of the release phase (c) the formant information is conserved much better than in the other case (b) 124

A.1 Pitch, amplitude and formant tracking of sample `e_hoch.wav` ($f_s = 22050$) as well as corresponding glottPlot 129
A.2 Pitch, amplitude and formant tracking of sample `e_mittel.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot 130
A.3 Pitch, amplitude and formant tracking of sample `e_tief.wav` ($f_s = 22050$) as well as corresponding glottPlot 131
A.4 Pitch, amplitude and formant tracking of sample `a_hoch.wav` ($f_s = 22050Hz$). Unfortunately, due to the corrupted VTR estimates we could not compute a representative glottPlot of the glottal source signal. 132
A.5 Pitch, amplitude and formant tracking of sample `a_mittel.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot 133
A.6 Pitch, amplitude and formant tracking of sample `a_tief.wav` ($f_s = 11025Hz$) as well as corresponding glottPlot 134

Bibliography

[1] Lawrence R. Rabiner and Ronald W. Schafer, *Theory and Application of Digitial Speech Processing*, Pearson Education, Inc., first 2011.

[2] R.L. Drake, W. Vogl, A.W.M. Mitchell, and H. Gray, *Gray's anatomy for students*, Number Teil 762 in Grays Anatomy for Students. Churchill Livingstone/Elsevier, 2010.

[3] Frederick Husler and Yvonne Rodd-Marling, *Singen - die physische Natur des Stimmorgans*, vol. 2, B. Schott's Söhne, Mainz, 1978.

[4] J Sundberg, *The science of the singing voice*, Northern Illinois University Press, DeKalb, IL, 1987.

[5] Moon Gi Kang and Byeong Gi Lee, "A generalized vocal tract model for pole-zero type linear prediction," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, apr 1988, pp. 687 –690 vol.1.

[6] Steven M. Lulich, Matias Zanartu, Daryush D. Mehta, and Robert E. Hillman, "Source-filter interaction in the opposite direction: Subglottal coupling and the influence of vocal fold mechanics on vowel spectra during the closed phase.," *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2638–2638, 2009.

[7] Nathalie Henrich, Christophe d'Alessandro, Boris Doval, and Michèle Castellengo, "Glottal open quotient in singing: Measurements and correlation with laryngeal mechanisms, vocal intensity, and fundamental frequency," *The Journal of the Acoustical Society of America*, vol. 117, no. 3, pp. 1417–1430, 2005.

[8] Michael Dickreiter, *Handbuch der Tonstudiotechnik*, vol. 6, Saur, 1997.

[9] Johan Sundberg, "Acoustic and psychoacoustic aspects of vocal vibrato," *Speech Transmission Laboratory. Quarterly Progress and Status Reports*, vol. 35, no. 2-3, pp. 045–068, 1994.

[10] B. Kostek and P. Zwan, “Automatic classification of singing voice quality,” in *Intelligent Systems Design and Applications, 2005. ISDA ’05. Proceedings. 5th International Conference on*, Sept. 2005, pp. 444–449.

[11] Alain de Cheveigne and Hideki Kawahara, “Yin, a fundamental frequency estimator for speech and music,” *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[12] I. Arroabarren, M. Zivanovic, X. Rodet, and A. Carlosena, “Instantaneous frequency and amplitude of vibrato in singing voice,” in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03). 2003 IEEE International Conference on*, April 2003, vol. 5, pp. V–537–40 vol.5.

[13] Stephen A. Zahorian and Hongbing Hu, “A spectral/temporal method for robust fundamental frequency tracking,” *The Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4559–4571, 2008.

[14] Adrian von dem Knesebeck and Udo Zölzer, “Comparison of pitch trackers for real-time guitar effects,” in *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, 2010.

[15] I. Arroabarren, X. Rodet, and A. Carlosena, “On the measurement of the instantaneous frequency and amplitude of partials in vocal vibrato,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1413–1421, July 2006.

[16] Paavo Alku, Carlo Magi, Santeri Yrttiaho, Tom Bäckström, and Brad Story, “Closed phase covariance analysis based on constrained linear prediction for glottal inverse filtering,” *The Journal of the Acoustical Society of America*, vol. 125, no. 5, pp. 3289–3305, 2009.

[17] Donald G. Childers and J. N. Larar, “Electroglottography for laryngeal function assessment and speech analysis,” *Biomedical Engineering, IEEE Transactions on*, vol. BME-31, no. 12, pp. 807 –817, dec. 1984.

[18] Patrick A. Naylor, Anastasis Kounoudes, Jon Gudnason, and Mike Brookes, “Estimation of glottal closure instants in voiced speech using the dypsa algorithm,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 34 –43, jan. 2007.

[19] D.M. Brookes and H.P. Loke, “Modelling energy flow in the vocal tract with applications to glottal closure and opening detection,” mar. 1999, vol. 1, pp. 213 –216 vol.1.

[20] M. Brookes, P.A. Naylor, and J. Gudnason, “A quantitative assessment of group delay methods for identifying glottal closures in voiced speech,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 456 – 466, mar. 2006.

[21] M.R.P. Thomas and P.A. Naylor, “The sigma algorithm: A glottal activity detector for electroglottographic signals,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1557 –1566, nov. 2009.

[22] John Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679 –698, nov. 1986.

[23] Marchand S. Rault J.-B. Lagrange, M., “Enhancing the tracking of partials for the sinusoidal modeling of polyphonic sounds,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1625 –1634, July 2007.

[24] I. Arroabarren and A. Carlosena, “Inverse filtering in singing voice: a critical analysis,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1422–1431, July 2006.

[25] H. Fujisaki and M. Ljungqvist, “Proposal and evaluation of models for the glottal source waveform,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP ’86.*, apr 1986, vol. 11, pp. 1605 – 1608.

[26] I. Arroabarren and A. Carlosena, “Glottal spectrum based inverse filtering,” in *Eighth European Conference on Speech Communication and Technology*, 2003.

[27] D. Wong, J. Markel, and Jr. Gray, A., “Least squares glottal inverse filtering from the acoustic speech waveform,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 4, pp. 350 – 355, aug 1979.

[28] TV Ananthapadmanabha and G. Fant, “Calculation of true glottal flow and its components,” *Speech Communication*, vol. 1, no. 3-4, pp. 167–184, 1982.

[29] M.D. Plumpe, T.F. Quatieri, and D.A. Reynolds, “Modeling of the glottal flow derivative waveform with application to speaker identification,” *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 5, pp. 569 –586, sep. 1999.

[30] Jacqueline Walker and Peter Murphy, “A review of glottal waveform analysis,” in *Progress in Nonlinear Speech Processing*, Yannis Stylianou, Marcos Faundez-Zanuy, and Anna Esposito, Eds., vol. 4391 of *Lecture Notes in Computer Science*, pp. 1–21. Springer Berlin / Heidelberg, 2007.

[31] N. Levinson, “The Wiener RMS (root mean square) error criterion in filter design and prediction,” *Selected papers of Norman Levinson*, p. 163, 1998.

[32] J. Makhoul, “Linear prediction: A tutorial review,” *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561 – 580, 1975.

[33] J. Markel and Jr. Gray, A., “Fixed-point truncation arithmetic implementation of a linear prediction autocorrelation vocoder,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 273 – 282, aug 1974.

[34] F. Itakura and S. Saito, “Digital filtering techniques for speech analysis and synthesis,” in *7th Int. Congr. Acoustics, Budapest*, 1971, number 25-C-1.

[35] John P. Burg, *Maximum Entropy Spectral Analysis*, Ph.D. thesis, Stanford Univ., Stanford, CA, May 1975.

[36] J. Makhoul, “Stable and efficient lattice methods for linear prediction,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 25, no. 5, pp. 423 – 428, Oct. 1977.

[37] J. Makhoul, “Correction to "stable and efficient lattice methods for linear prediction",” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 111, Feb. 1978.

[38] P.P. Vaidyanathan, *The Theory of Linear Prediction*, Synthesis lectures on signal processing. Morgan & Claypool, 2008.

[39] J. Bonada and X. Serra, “Synthesis of the singing voice by performance sampling and spectral models,” *Signal Processing Magazine, IEEE*, vol. 24, no. 2, pp. 67 –79, march 2007.

[40] G. Fant and Q. Lin, “Glottal source-vocal tract acoustic interaction,” *Speech Transmission Laboratory Quarterly Progress and Status Report*, vol. 1, pp. 13Â27, 1987.

[41] M. Huckvale, “Speech filing system: Tools for speech,” Online Available: http://www.phon.ucl.ac.uk/resource/sfs, University College London, 2004.

[42] Mark R. P. Thomas and Patrick A. Naylor, “The sigma algorithm for estimation of reference-quality glottal closure instants from electroglottograph signals,” *16th European Signal Processing Conference (EUSIPCO 2008)*, aug 2008.

[43] Hans Werner Strube, “Determination of the instant of glottal closure from the speech wave,” *The Journal of the Acoustical Society of America*, vol. 56, no. 5, pp. 1625–1629, 1974.

[44] Changxue Ma, Y. Kamp, and L.F. Willems, “A frobenius norm approach to glottal closure detection from the speech signal,” *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 2, pp. 258 –265, apr. 1994.

[45] R. Smits and B. Yegnanarayana, “Determination of instants of significant excitation in speech using group delay function,” *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 325 –333, sep 1995.

[46] B. Yegnanarayana and R.L.H.M. Smits, “A robust method for determining instants of major excitations in voiced speech,” may. 1995, vol. 1, pp. 776 –779 vol.1.

[47] P. Satyanarayana Murthy and B. Yegnanarayana, “Robustness of group-delay-based method for extraction of significant instants of excitation from speech signals,” *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 6, pp. 609 –619, nov. 1999.

[48] Enders A. Robinson, Tariq S. Durrani, and Lloyd G. Peardon, *Geophysical signal processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

[49] Andrew J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[50] J. Bloit and X. Rodet, “Short-time viterbi for online hmm decoding: Evaluation on a real-time phone recognition task,” mar. 2008, pp. 2121 –2124.