Glottissynchrone harmonische Filterung stimmhafter Sprache für die automatische Spracherkennung

Diplomarbeit

Vorgelegt von Henning Schal
am 30. September 2009

Erstgutachter: Prof. Dr. Sven Behnke
Zweitgutachter: Prof. Dr. Michael Clausen

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik VI
Autonome Intelligente Systeme

Inhaltsverzeichnis

Kapitel 1.

Einleitung

Als Medium für die zwischenmenschliche Kommunikation hat sich die gesprochene Sprache durchgesetzt. Gegenüber anderen Arten der Kommunikation weist diese einige Vorteile auf. Informationen und Anweisungen können von Menschen zuverlässig und schnell in Laute umgesetzt werden. In der Nähe befindliche Zuhörer können diese Laute verstehen und umsetzen. Das gelingt Menschen auch bei starkem Lärm, beispielsweise an einer vielbefahrenen Straße. Ein weiterer Vorteil gesprochener Sprache ist es, dass Sprecher und Zuhörer ihre visuelle Aufmerksamkeit während der verbalen Kommunikation unabhängig vom Gegenüber auf andere Dinge konzentrieren können. So können sie beispielsweise beim Autofahren mithilfe einer Freisprechanlage telefonieren. Auch die Bewegungsfreiheit wird beim Sprechen nicht eingeschränkt. Diese Vorteile der Lautsprache motivieren dazu, es auch künstlichen Systemen zu ermöglichen, gesprochene Sprache robust zu erkennen.

1.1. Problemstellung

Das Ziel dieser Arbeit ist es, Sprache automatisch und unter schweren Bedingungen zu erkennen. Dazu sollen die Eigenschaften stimmhafter Sprache ausgenutzt werden. Ein großer Teil gesprochener Sprache besteht aus dieser stimmhaften Sprache, die durch ein quasi-periodisches Quellsignal angeregt wird. In der Idealvorstellung ist das Quellsignal eine Folge von periodisch auftretenden Impulsen zu sogenannten Verschlussmomenten.

Es existieren bereits Verfahren, welche die Verschlussmomente und die Grundfrequenz dieses Quellsignals in Sprachsignalen schätzen. Eines dieser Verfahren benutzt eine harmonische Filterung, welche die Eigenschaften stimmhafter Sprache im Frequenzraum auszunutzt, um die Grundfrequenz robust zu schätzen.

In dieser Arbeit werden weitere Möglichkeiten untersucht, diese harmonische Filterung für die robuste automatische Spracherkennung zu verwenden.

Dazu soll untersucht werden, ob die harmonische Filterung verwendet werden kann, um nicht nur die Grundfrequenz zu schätzen, sondern auch Verschlussmomente robust zu finden. Dann soll darauf aufbauend ein iteratives Verfahren entworfen werden, dass mit begrenzter Vorausschau die Verschlussmomente und die Grundfrequenz verfolgt. Um die Verschlussmomente zu verfolgen, soll glottissynchron vorgegangen werden. Das heißt, dass die Untersuchung des Sprachsignals synchron mit den Verschlussmomenten geschieht. Zusätzlich sollen Untersuchungen angestellt werden, wie die glottissynchrone Verarbeitung für die robuste automatische Spracherken-

nung verwendet werden kann. Außerdem soll untersucht werden, ob die harmonische Filterung dazu verwendet werden kann, bei stimmhafter Sprache Merkmale für die Spracherkennung aus dem Phasengang zu extrahieren.

Es existiert ein Verfahren zur Merkmalsextraktion, das mithilfe einer Grundfrequenzschätzung robust Sprache erkennen kann. Das in dieser Arbeit zu entwickelnde Verfahren zur Grundfrequenzschätzung, soll mit diesem existierenden Verfahren zu einem Frontend kombiniert werden, mit dem robust Sprache erkannt werden kann.

1.2. Aufbau der Arbeit

In Kapitel 2 wird ein Überblick über verschiedene Bereiche gegeben, die zum Verständnis der Arbeit notwendig sind. Es handelt sich dabei um Grundlagen der digitalen Signalverarbeitung, der Phonetik und der automatischen Spracherkennung. Danach werden in Kapitel 3 einige verwandte Arbeiten zur Schätzung der Grundfrequenz, der Verschlussmomente und zur Rauschunterdrückung vorgestellt. Kapitel 4 ist der Hauptteil der vorliegenden Arbeit. Darin wird ein iteratives Verfahren entworfen, mit dem die Eigenschaften des Quellsignals robust und mit begrenzter Vorausschau verfolgt werden sollen. Es werden Techniken zur Schätzung der Verschlussmomente und der Grundfrequenz an das iterative Verfahren angepasst und integriert. Schließlich werden einige Überlegungen angestellt, wie die glottissynchrone Verarbeitung für die robuste automatische Spracherkennung benutzt werden kann. Details zur Realisierung werden in Kapitel 5 erläutert. Genauigkeit und Robustheit des entwickelten Algorithmus werden in Kapitel 6 evaluiert. Verschiedene Eigenschaften des Algorithmus werden in diesem Kapitel auch visuell evaluiert. Eine Zusammenfassung und Diskussion in Kapitel 7 schließt die Arbeit ab.

Kapitel 2.

Grundlagen

Das Grundlagenkapitel legt die Basis für die folgenden Kapitel. Die digitale Signalverarbeitung, die Phonetik und automatische Spracherkennung werden vorgestellt. Auf weiterführende Literatur wird verwiesen.

2.1. Digitale Signalverarbeitung

Häufig wird Audiosignalverarbeitung in zeit- und wertediskreten Systemen durchgeführt. Auch im Rahmen der Diplomarbeit ist das der Fall. Daher müssen kontinuierliche Vorgänge auf diskrete Systeme abgebildet werden. Einige Grundlagen zur digitalen Signalverarbeitung werden daher in diesem Abschnitt vorgestellt. Die Darstellungen basieren im Wesentlichen auf zwei Quellen [CM03, OS75].

2.1.1. Signale, Abtastung und Signalräume

Signale sind Gegenstand der Signalverarbeitung. Sie können wie folgt definiert werden.

Definition 2.1.1 (Signal) *Ein Signal ist eine Abbildung:*

$$f : D^n \to \mathbb{C}^m \text{ für } m, n \in \mathbb{N}^+ \text{ und } D \in \{\mathbb{R}, \mathbb{Z}\}$$

Für $D = \mathbb{R}$ heißt ein Signal kontinuierlich. Für $D = \mathbb{Z}$ heißt ein Signal diskret. Ist der Definitionsbereich D eindimensional, also $n = 1$, spricht man auch von zeitdiskreten und zeitkontinuierlichen Signalen, da Signale häufig durch Zeit parametrisiert sind.

Die folgenden Ausführungen beziehen sich auf den eindimensionalen Fall $m = n = 1$. Ein Beispiel für ein zeitdiskretes Signal ist der Einheitsimpuls $\delta : \mathbb{Z} \to \mathbb{C}$.

$$\delta(n) = \begin{cases} 1 + 0i \text{ für } n = 0, \\ 0 + 0i \text{ für } n \neq 0 \end{cases} \tag{2.1}$$

Digitale Signale sind nicht nur zeitdiskret, sondern zusätzlich auch wertdiskret. Ein zeitkontinuierliches Signal kann durch *Abtastung* in ein zeitdiskretes Signal umgewandelt werden. Zu diskreten Abtastpunkten $Z \subseteq \mathbb{R}$ wird der Wert des zeitkontinuierlichen Signals übernommen. Bei der *periodischen Abtastung* wird das zeitkontinuierliche Signal $x_c \in \mathbb{C}^{\mathbb{R}}$ äquidistant abgetastet, um das zeitdiskrete Signal $x_d \in \mathbb{C}^{\mathbb{Z}}$

zu erhalten.

$$x_d(n) = x_c(n \cdot T), -\infty < n < \infty, T \in \mathbb{R} \tag{2.2}$$

T heißt *Abtastperiode*. Die *Abtastrate* ist 1/T.

Um die Menge der Signale

$$\mathbb{C}^D := \{x | x : D \to \mathbb{C}\}, \text{D } \in \{\mathbb{R}, \mathbb{Z}\}. \tag{2.3}$$

handhabbarer zu machen, werden *Signalräume* definiert. Signalräume sind Vektorräume über dem Körper $\mathbb{C}$, denen eine Menge von Signalen zugrundeliegt.

Wichtige Signalräume sind die Lebesgue-Räume $l^p(\mathbb{Z})$, $1 \leq p \leq \infty$. Sie enthalten alle zeitdiskreten Signale $x \in \mathbb{C}^{\mathbb{Z}}$, die

$$\sum_{n \in \mathbb{Z}} ||x(n)||^p < \infty \text{ für } 1 \leq p \leq \infty \tag{2.4}$$

erfüllen. Auf den Lebesgue-Räumen, kann eine Norm für $x \in l^p(\mathbb{Z})$ definiert werden.

$$||x||_p := (\sum_{n \in \mathbb{Z}} |x(n)|^p)^{1/p} \text{ ,für } 1 \leq p < \infty \tag{2.5}$$

$$||x||_\infty := sup|x(n)| : n \in \mathbb{Z} \text{ ,für } p = \infty$$

$||x||_2^2$ wird Energie eines Signals $x \in l^2(\mathbb{Z})$ genannt. Der Lebesgue-Raum $L^2(\mathbb{R})$ ist der zeitkontinuierliche Gegenpart zum $l^2(\mathbb{Z})$ und enthält alle quadratisch integrierbaren zeitkontinuierlichen Signale $S \subseteq \mathbb{C}^D$.

2.1.2. Periodische Signale, Fourier-Transformation und das Abtasttheorem

Definition 2.1.2 (zeitkontinuierliches periodisches Signal) *Ein Signal $f \in \mathbb{C}^{\mathbb{R}}$ ist periodisch mit Periode $T \in \mathbb{R}$(auch T-periodisch genannt), genau dann wenn*

$$f(t) = f(t + T).$$

Jedes periodische Signal $f \in \mathbb{C}^{\mathbb{R}}$ mit Periode T ist vollständig bekannt, wenn es für das Intervall $[0, T]$ bekannt ist. f kann durch eine lineare Abbildung in ein 1-periodisches Signal überführt werden. Daher kann man sich bei der Betrachtung periodischer Funktionen in vielen Fällen auf die Betrachtung 1-periodischer Funktionen beschränken und die Ergebnisse verallgemeinern. Der $L^2([0,1])$ ist ein Vektorraum und enthält alle 1-periodischen Signale, die über dem Intervall $[0,1]$ quadratisch integrierbar sind.

Wie jeder Vektorraum, hat der $L^2([0,1])$ mindestens eine Basis. Eine wichtige Basis des $L^2([0,1])$ ist diese Menge von Signalen:

$$\{e_k := e^{2\pi ikt} | k \in \mathbb{Z}\} \subset L^2([0,1]) \text{ für } t \in [0,1]. \tag{2.6}$$

Jedes Signal $f \in L^2([0,1])$ kann daher als Linearkombination dieser Basisvektoren als sogenannte *Fourier-Reihe* dargestellt werden.

$$f(t) = \sum_{k=-\infty}^{\infty} F(k) e^{2\pi ikt}, \text{ für } t \in [0,1] \tag{2.7}$$

Die Koeffizienten

$$F(k) = \int_0^1 f(t)e^{-2\pi ikt}dt, \text{ für } k \in \mathbb{Z} \tag{2.8}$$

werden *Fourier-Koeffizienten* genannt. Das Signal f im *Zeitbereich* kann also durch das Signal $F(k) \in l^2(\mathbb{Z})$ im sogenannten *Frequenzbereich* dargestellt werden. $|F(k)| \in \mathbb{R}$ ist die *Magnitude* des k-ten Fourier-Koeffizienten, $\angle F(k) \in [-\pi, \pi]$ ist die *Phase* des k-ten Fourier-Koeffizienten.

Die *diskrete Fourier-Transformation (DFT)* kann die Fourier-Koeffizienten $F(k)$ einer 1-periodischen zeitkontinuierlichen Funktion $f \in L^2([0,1])$ annähern. Dazu wird das Integral aus 2.8 durch eine Riemann-Summe ersetzt. Für einen Vektor $v \in \mathbb{C}^N$ ($N \in \mathbb{N}^+$), ist dessen diskrete Fourier-Transformation (DFT) gegeben durch:

$$V(k) := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} v(j)e^{-2\pi ijk/N}, k = 0, 1, ..., N-1 \tag{2.9}$$

$V(k)$ ist N-periodisch, es genügt daher, die Koeffizienten für $k = 0, ..., N-1$ zu berechnen. Dies kann durch die *schnelle Fourier-Transformation (FFT)* in Zeit $O(Nlog(N))$ geschehen. Können die Fourier-Koeffizienten nicht analytisch berechnet werden, ist die FFT eine Möglichkeit, die Koeffizienten effizient zu approximieren. Die *zeitdiskrete Fourier-Transformation* für $x \in l^2(\mathbb{Z})$

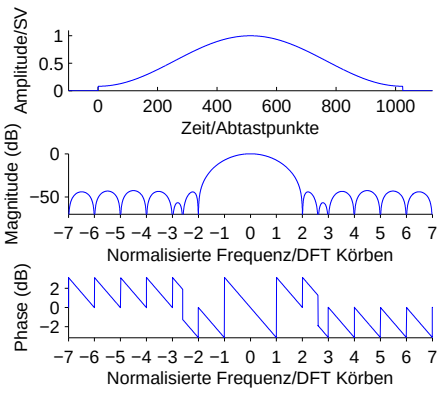
$$X(\omega) := \sum_{k=-\infty}^{\infty} x(k)e^{-2\pi ik\omega}, \text{ für } \omega \in [0,1] \tag{2.10}$$

wird für zeitdiskrete Signale mit endlichem Träger durch die DFT angenähert, indem Periodizität angenommen wird. $X(\omega)$ ist 1-periodisch. Die zeitdiskrete Fourier-Transformation wiederum nähert die *zeit-kontinuierliche Fourier-Transformation* für $f \in L^2(\mathbb{R})$

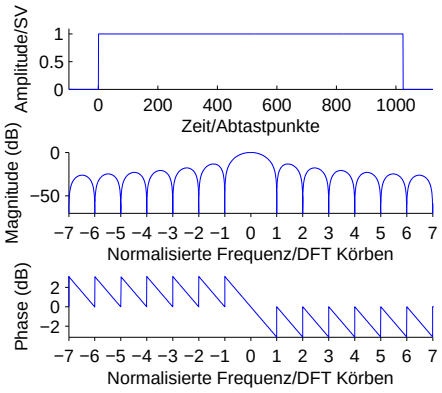
$$F(\omega) := \int_{k=-\infty}^{\infty} f(t)e^{-2\pi i\omega t}dt, \text{ für } \omega \in \mathbb{R} \tag{2.11}$$

durch eine Riemann-Summe an. Der Übergang von zeitkontinuierlichen Signalen zu zeitdiskreten Signalen durch Abtastung ist im Allgemeinen verlustreich. Die Darstellung eines zeitkontinuierlichen Signals im Frequenzbereich kann genutzt werden, um die Abtastrate zu bestimmen, die für einen verlustlosen Übergang mindestens notwendig ist. Ist das zeitkontinuierliche Signal $f \in L^2(\mathbb{R})$ *bandbegrenzt*, das heißt $F(\omega) = 0$ für $|\omega| > \Omega > 0$, dann kann f aus dem mit Periode T abgetasteten Signal x rekonstruiert werden, falls die Periode T bekannt ist und $T \leq \frac{1}{2\Omega}$. Dies ergibt sich aus dem *Abtasttheorem von Shannon*. Die Abtastrate 2Ω wird *Nyquist-Rate* genannt. Wird die Abtastrate zu gering gewählt, tritt *aliasing* auf: Bei der zeitdiskreten Fourier-Transformation werden hohe Frequenzanteile des zeitkontinuierlichen Signals in tiefere Frequenzbereiche verschoben.

Die DFT leidet nicht nur unter aliasing, sondern zusätlich unter dem *Leakage-Effekt*, der sich aus der begrenzten Frequenzauflösung der DFT ergibt. Wird ein bandbegrenztes zeitkontinuierliches Signal mit der Nyquist-Rate f_s periodisch abgetastet, und ein Ausschnitt von N Abtastpunkten aus diesem Signal $x \in l^2(\mathbb{R})$ mit der



(a) Hamming Fenster



(b) Rechteckiges Fenster

Abbildung 2.1.: Fensterfunktionen. Oben findet sich die Darstellung der Fenster im Zeitbereich, unten die Darstellung des Amplitudengangs und des Phasengangs im (kontinuierlichen) Frequenzbereich.

DFT analysiert, so entsprechen die Koeffizienten $V(k)$ des *Korbes* k den Frequenzen

$$f_{analysis}(k, N) = \frac{kf_s}{N} \ , \ k \in \{0, 1, ..., \lfloor N/2 \rfloor\}. \tag{2.12}$$

War das abgetastete Signal $x \in l^2(\mathbb{R})$ nicht $\frac{N}{f_s}$-periodisch, so enthält x auch Frequenzen außer der *Grundfrequenz* $f_{analysis}(1)$ und den *Harmonischen* $f_{analysis}(k)$, $k \in \{1, ..., \lfloor N/2 \rfloor\}$. Diese Frequenzanteile verschmieren dann in verschiedene Frequenzkörbe/Koeffizienten der DFT. Das Ausschneiden des endlichen Signals kann man sich als Multiplikation von x mit einer sogenannten *Fensterfunktion* vorstellen: $x'(n) = x(n) \cdot w(n)$. w hat oft einen endlichen Träger. In Abbildung 2.1 sind verschiedene Fensterfunktionen im Zeit- und Frequenzbereich dargestellt. Ein Hamming-Fenster $w \in l^2(\mathbb{Z})$ der Länge $N + 1 \in \mathbb{N}$ ist

$$w(n) = \begin{cases} 0.54 - 0.46 \cdot cos(2\pi \frac{n-1}{N}) \text{ für } 1 \leq n \leq N + 1 \\ 0 \ , \ sonst \end{cases} \tag{2.13}$$

Ein Rechteckfenster-Fenster $w \in l^2(\mathbb{Z})$ der Länge $N \in \mathbb{N}$ ist

$$w(n) = \begin{cases} 1 \text{ für } 1 \leq n \leq N \\ 0 \ , \ sonst \end{cases} \tag{2.14}$$

Im Frequenzbereich erkennt man, dass für $\frac{N}{f_s}$-periodische Signale bei einer rechteckigen Fensterfunktion kein Leakage auftritt (weil Multiplikation im Zeitbereich einer Konvolution im Frequenzbereich entspricht).

2.1.3. Systeme

Ein *System* T ist eine Abbildung zwischen zwei Signalräumen. Im Folgenden werden nur Abbildungen zwischen Räumen zeitdiskreter Signale betrachtet. Ein Beispiel für

ein System ist die Konvolution.

Definition 2.1.3 (Konvolution) *Seien x, y:* $\mathbb{Z} \to \mathbb{C}$ *Signale. Dann ist die zeitdiskrete Konvolution von x und y definiert als*

$$(x * y)(n) := \sum_{k \in \mathbb{Z}} x(k)y(n-k), \text{ für } n \in \mathbb{Z}$$

Systeme können anhand ihrer Eigenschaften klassifiziert werden. Ein Hilfsmittel dazu ist die sogenannte *Impulsantwort* $h := T[\delta]$ eines Systems T. Eine wichtige Klasse von Systemen sind die sogenannten *linearen zeitinvarianten Systeme (LZI-Systeme)* [CM03]. Ein kontinuierliches LZI-System $T : l^p(\mathbb{Z}) \to l^q(\mathbb{Z})$ bildet konvergierende Folgen des Eingabesignals auf ebenfalls konvergierende Folgen des Ausgangssignals ab. T kann dann vollständig durch seine Impulsantwort beschrieben werden.

$$T[x] = x * h \text{ für alle } x \in l^p(\mathbb{Z}) \tag{2.15}$$

Ein LZI-System $T : l^p(\mathbb{Z}) \to l^q(\mathbb{Z})$ wird *stabil* genannt, wenn es kontinuierlich ist und seine Impulsantwort absolut summierbar ist. T ist *BIBO-stabil*, wenn es stabil ist und $p = \infty$. T ist kausal, wenn es kontinuierlich ist und für seine Impulsantwort h gilt, dass $h(n) = 0$ für $n < 0$. Der Frequenzgang eines BIBO-stabilen LZI-Systems mit Impulsantwort $h \in l^1(\mathbb{Z})$ ist gegeben durch die zeitdiskrete Fouriertransformierte von h.

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-2\pi ik\omega}, \text{ für } \omega \in [0, 1] \tag{2.16}$$

Der Amplitudengang von T ist $|H(\omega)| \in \mathbb{R}$, der Phasengang von T ist $\angle(H(\omega)) \in [-\pi, \pi]$. Der kontinuierliche Phasengang $arg[H(\omega)] \in \mathbb{R}$ ist eine kontinuierliche Funktion in ω. Der Phasengang wird kontinuierlich gemacht, indem die Beziehung $e^{i\angle H(\omega)} = e^{i(\angle H(\omega)+k2\pi)}$ für $k \in \mathbb{Z}$ ausgenutzt wird. Die Gruppenlaufzeit von T ist definiert als

$$\tau(\omega) = -\frac{d}{d\omega} arg[H(\omega)]. \tag{2.17}$$

Der Frequenzgang und die Gruppenlaufzeit des Systems T können genutzt werden, um das Verhalten von T zu analysieren (da Konvolution im Zeitbereich Multiplikation im Frequenzbereich entspricht [OS75]).

2.1.4. Übertragungsfunktionen und Minimalphasensystem

Die *z-Transformation* für ein Signal $x : \mathbb{Z} \to \mathbb{C}$ ist durch

$$X(z) := \sum_{n \in \mathbb{Z}} x(n)z^{-n}, \text{ für } z \in \mathbb{C} \tag{2.18}$$

gegeben. Diese Summe konvergiert im Allgemeinen nicht für alle $z \in \mathbb{C}$ und ist dann nicht definiert. Die z-Transformation kann verwendet werden, um Systeme zu

charakterisieren. Ist T ein BIBO-stabiles LZI-System mit Impulsantwort $h = T[\delta] \in l^1(\mathbb{Z})$, so wird die z-Transformation

$$H(z) := \sum_{n\in\mathbb{Z}} h(n)z^{-n}, \text{ für } z \in \mathbb{C} \tag{2.19}$$

Übertragungsfunktion von T genannt. Anhand der Eigenschaften der Übertagungsfunktion lässt sich zum Beispiel ebenfalls die Kausalität eines Systems definieren [OS75]. Ist die Übertragungsfunktion *rational*, d.h.

$$H(z) = (\frac{b_0}{a_0})\frac{\prod_{k=1}^{M}(1-c_k z^{-1})}{\prod_{k=1}^{N}(1-d_k z^{-1})},\ N,\ M \in \mathbb{N};\ b_0, a_0, c_k, d_k \in \mathbb{C}, \tag{2.20}$$

so existiert auch das *inverse System* mit der Übertragungsfunktion $H_i(z) = \frac{1}{H(z)}$ [OS75].

Definition 2.1.4 (Minimalphasensystem) *Ein System T mit rationaler Übertragungsfunktion H heißt Minimalphasensystem, wenn T sowie das durch $H_i := \frac{1}{H(z)}$ definierte System stabil und kausal sind.*

Ein Minimalphasensystem zeichnet sich unter anderem dadurch aus, dass der Phasengang aus dem Amplitudengang rekonstruiert werden kann. Umgekehrt kann aus dem Phasengang der Amplitudengang bis auf Skalierung wiederhergestellt werden [OS75]. Der durchschnittliche Phasengang und die durchschnittliche Gruppenlaufzeit eines Minimalphasensystems betragen Null [YS95]. Außerdem verzögert ein Minimalphasensystem die in seiner Impulsantwort enthaltene Energie minimal: Sind zwei Impulsantworten $h_{min}, h \in l^1(\mathbb{Z})$ gegeben, deren Amplitudengang $|H(\omega)|$ identisch ist, so zeichnet sich die Impulsantwort des Minimalphasensystems h_{min} durch

$$\sum_{m=0}^{n} |h[m]|^2 \leq \sum_{m=0}^{n} |h_{min}[m]|^2 \text{ für alle } n \in \mathbb{N}^+ \tag{2.21}$$

aus. Die Impulsantwort eines Minimalphasensystems wird auch *Minimalphasensignal* genannt.

2.1.5. Gefensterte Fourier-Transformation

Definition 2.1.5 (Gefensterte Fourier-Transformation) *Die zeitdiskrete gefensterte Fourier-Transformation oder Kurzzeit-Fourier-Transformation eines Signals $x \in \mathbb{C}^{\mathbb{Z}}$ und eines Fensters $w \in \mathbb{C}^{\mathbb{Z}}$ ist definiert als*

$$V(m,\lambda) = \sum_{n=-\infty}^{\infty} x(m+n)w(n)e^{-i2\pi\lambda n}\ ,\ \lambda \in [0,1],\ n \in \mathbb{Z}.$$

Hat das Fenster w einen endlichen Träger, also beispielsweise $w(n) = 0$ für $0 \leq n < L \in \mathbb{N}^+$, dann kann die zeitdiskrete gefensterte Fourier-Transformation X_m für das Signal $x_m(n) := \sum_{m=-\infty}^{\infty} x(n+m)w(m)$ mittels der DFT angenähert werden [OS75]:

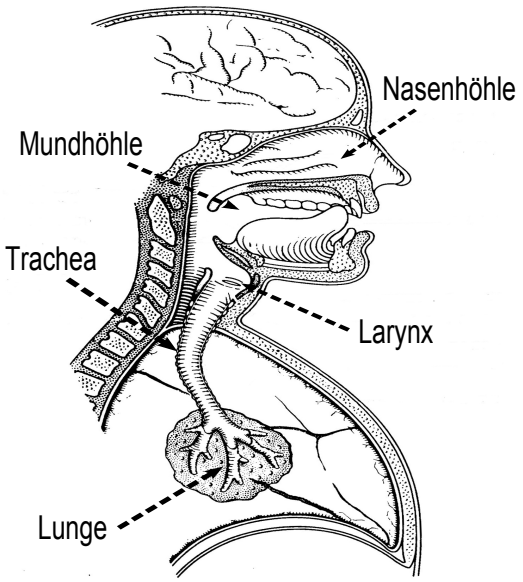


Abbildung 2.2.: Sprechapparat des Menschen. (Quelle: [Fla65], bearbeitet)

$$X_m(k) = \sum_{n=0}^{L-1} x(m+n)w(n)e^{-i2\pi(k/L)n} \text{ , } 0 \leq k \leq L-1,\ n \in \mathbb{Z}. \tag{2.22}$$

Die zeitdiskrete gefensterte Fourier-Transformation hängt von der Zeit ab und kann somit Veränderungen der Frequenzzusammensetzung der gefensterten Signale x_m über die Zeit erfassen. Das ist sinnvoll, wenn das Signal x nicht stationär ist, wie beispielsweise Sprachsignale.

2.2. Phonetik

Um Spracherkennung und Sprachlauterzeugung mit künstlichen Systemen durchzuführen, ist es hilfreich, den Sprechapparat und das auditive System des Menschen zu verstehen. Daher wird in diesem Abschnitt der Sprechapparat und seine Funktionsweise vorgestellt. Anschließend wird auf das auditive System eingegangen, das Schall verarbeitet.

Die Phonetik beschäftigt sich mit der Erzeugung und Wahrnehmung von Sprachlauten. Da in dieser Arbeit nur ein Überblick über Teile der Phonetik gegeben werden kann, verweise ich hier auf einführende und weiterführende Literatur [PN91, Ste98, PM95, Fla65, SKD97].

2.2.1. Sprachlauterzeugung

Die Sprachproduktion ist ein komplexer Vorgang, der seinen Anfang in einer Äußerungsabsicht nimmt. Soll diese Äußerung durch Laute vorgenommen werden, beginnt die *Sprachlauterzeugung*. Die Gesamtheit der menschlichen Organe, die an

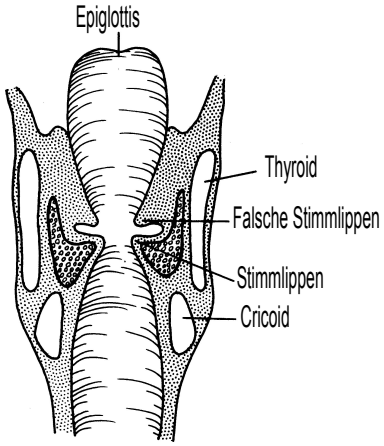


Abbildung 2.3.: Frontalansicht des Kehlkopfes. (Quelle: [SKD97], bearbeitet)

der Produktion von Lautsprache beteiligt sind, wird als *Sprechapparat* bezeichnet [PK08]. In Abbildung 2.2 ist der Sprechapparat schematisch dargestellt. Die Respiration (Atmung), die Phonation (Stimmgebung) und die Artikulation sind die funktionalen Einheiten des Sprechapparates. Im Folgenden werden diese Einheiten näher beschrieben.

2.2.1.1. Respiration

Die *Respiration* ensteht im Wesentlichen durch Volumenänderungen der Lunge durch Muskelkraft. Beim Ausatmen wird das Lungenvolumen verringert. Daraufhin erhöht sich der Luftdruck in der Lunge und ein Luftstrom wird aus der Lunge über die *Trachea (Luftröhre)* und weitere Stationen bis zu Mund- und Nasenöffnungen gepresst. Sprachlaute werden in der Regel beim Ausatmen erzeugt. Dazu wird der Fluss der ausströmenden Luft durch Verengungen behindert, wodurch Luftdruckschwankungen entstehen, welche sich als Schall ausbreiten. Ohne einen Luftstrom kann der Sprechapparat keine Laute generieren, weshalb die Respiration ein wichtiger Teil der Sprachlauterzeugung ist.

2.2.1.2. Phonation

Die *Phonation (Stimmgebung)* geschieht in der *Larynx (Kehlkopf)*, die in Abbildung 2.3 dargestellt ist.

Phonation ist nicht einheitlich definiert. Hier wird Phonation als durch Atmung angeregte Schallerzeugung im Kehlkopf definiert [Fla65]. Bei diesem Prozess spielen die *Stimmlippen* eine herausragende Rolle, die in Abbildung 2.4(a) gezeigt werden und sich im Kehlkpopf befinden. Die Ritze zwischen den Stimmlippen wird *Glottis (Stimmritze)* genannt. Die Stimmlippen können durch verschiedene Muskeln verschlossen, geöffnet, gestreckt oder entspannt werden. Alle Organe des Sprechapparates, die unterhalb der Glottis liegen, bezeichnet man als *subglottales System*. Das

(a) Aufsicht der Stimmlippen. (Quelle: [SKD97])

(b) Vibration der Stimmlippen. (Quelle: [Fla65])

Abbildung 2.4.: Stimmlippen. Links ist die Anatomie der Stimmlippen skizziert. Rechts sieht man die Öffnungsgrade der Glottis während eines Phonationszyklus.

supraglottale System umfasst alle Organe des Sprechapparates, die oberhalb der Glottis angesiedelt sind. Während der Phonation werden die Stimmlippen durch Muskelkraft aneinander angenähert und angespannt. Die Atmung versetzt die Stimmlippen dann in Schwingung.

Die Schwingung der Stimmlippen durchläuft verschiedene Phasen, die in Abbildung 2.4 aufgezeigt werden. Als Ausgangspunkt liegen dort die Stimmlippen aneinander, die Glottis ist verschlossen. Es ist kein Druckausgleich zwischen subglottalem und supraglottalem System möglich. Beim Ausatmen erhöht sich der Druck im subglottalen System, wodurch die Stimmlippen schließlich auseinander gedrückt werden und Luft zu strömen beginnt. Die Stimmlippen öffnen sich immer weiter, der lokale Druck zwischen den Stimmlippen wird aufgrund des Bernoulli-Effektes geringer [PM95]. Es ensteht eine Sogwirkung, aufgrund derer die Stimmlippen sich wieder schließen. Der Moment, in dem die Stimmlippen sich vollständig verschließen, der Lufstrum also unterbrochen wird, heißt *Verschlussmoment* der Glottis oder *GCI (Glottal Closure Instant)*. Der Vorgang, der von einem Verschlussmomenent zum Nächsten führt, ist der sogenannte *Phonationszyklus*. Wie der Name schon andeutet, wiederholt sich der Phonationszyklus periodisch, da mit dem Verschlussmoment die Ausgangslage des Zyklus wiederhergestellt ist. Der Phonationszyklus lässt sich wie oben beschrieben in drei Abschnitte einteilen: Die *geschlossene Phase*, die *Öffnungsphase* und die *Schließphase*. Die *Grundfrequenz* oder *Fundamentalfrequenz* in Hertz (Hz) ist die Anzahl der Phonationszyklen pro Sekunde. Bei Frauen liegt die Grundfrequenz im Mittel bei etwa 230 Hz, bei Männern liegt sie bei 120 Hz [PM95]. Dieser Unterschied ist hauptsächlich durch unterschiedliche Massen und Längen der Stimmlippen bedingt. Individuen können die Grundfrequenz durch Streckung der Stimmlippen variieren. Dadurch werden die Stimmlippen angespannt und die Masse der Stimmlippen pro Länge nimmt ab. Beides erhöht die Grundfrequenz [SKD97]. Bei konstanter Ausatmung und Stimmlippenstreckung bleibt auch die Grundfrequenz konstant. Je stärker die Stimmlippen gestreckt werden, desto höher ist die Grundfrequenz. Das periodische Öffnen und Schließen der Glottis erzeugt Schall. In Abbildung 2.5 ist die Öffnungsfläche der Glottis über die Zeit dargestellt.

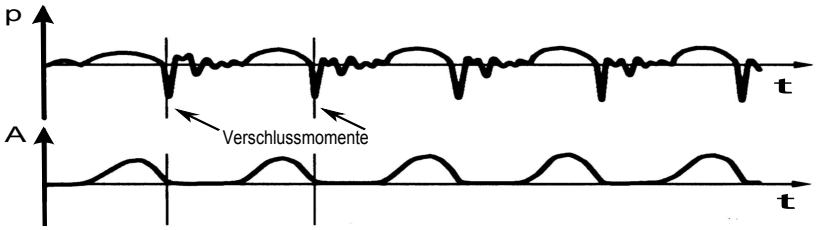


Abbildung 2.5.: Schallerzeugung. Unten ist die Öffnungsfläche der Glottis während der Phonation dargestellt. Oben sieht man parallel dazu den Schalldruck oberhalb der Glottis. Es fällt die starke Schwankung zum Verschlussmoment auf. (Quelle: [PM95])

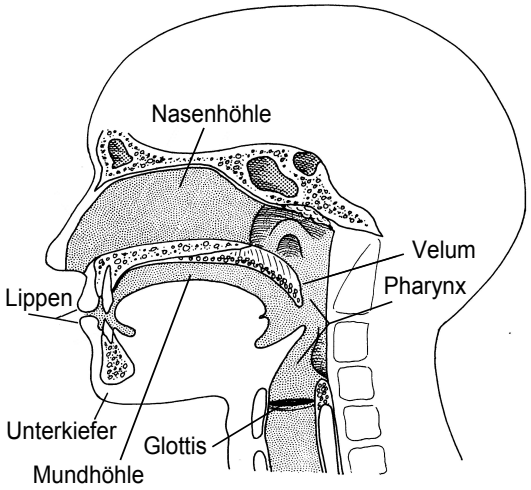


Abbildung 2.6.: Die Artikulatoren und Höhlen des Vokaltraktes. (Quelle: [SKD97], bearbeitet)

Zum Vergleich ist dazu der Schalldruck direkt oberhalb der Glottis aufgetragen. Im Moment des Glottisverschlusses kommt es zu einem negativen Luftdruckimpuls, der sich als Schallwelle ausbreitet [PM95].

2.2.1.3. Artikulation

Der Begriff *Artikulation* wird in verschiedenen Zusammenhängen verwendet. Hier definiere ich Artikulation als die Tätigkeit des Sprechapparates oberhalb des Kehlkopfes bei der Lautbildung [PN91]. Im Wesentlichen besteht diese Tätigkeit darin, die *Stellung des Vokaltraktes* durch sogenannte Artikulatoren zu verändern. Der *Vokaltrakt* wird durch die Hohlräume zwischen den Stimmlippen und der Mund- und Nasenöffnung gebildet. In Abbildung 2.6 sind der Vokaltrakt und seine Hohlräume abgebildet: *Der Pharynx (Rachen)*, die *Mundhöhle* sowie die *Nasenhöhle*. Die Form

dieser Höhlen kann durch *Artikulatoren* verändert werden. Solche Artikulatoren sind beispielsweise die Zunge, das Velum (Gaumensegel) und die Mandibula (Unterkiefer), siehe Abbildung 2.6. Beim Sprechen werden die Artikulatoren ständig bewegt, weshalb sich auch die Stellung des Vokaltraktes ständig ändert. Der Vokaltrakt weist Resonanzeigenschaften auf. Je nach Stellung des Vokaltraktes variieren die Resonanzfrequenzen des Vokaltraktes. Wird der Vokaltrakt durch Schall angeregt, so unterdrückt er gewisse Frequenzkomponenten, andere lässt er passieren.

Bei der Erzeugung von Sprachlauten wird der Vokaltrakt durch verschiedene Quellen angeregt. Zum einen ist das der Schall, der durch die Phonation im Kehlkopf entsteht. Zum anderen ist das Schall, der durch Verengungen im Vokaltrakt und dadurch verursachte Verwirbelungen des Luftstroms entsteht. Der nicht durch Phonation angeregte Teil der Sprache wird *stimmlose Sprache* genannt. Durch Phonation angeregte Sprache wird *stimmhafte Sprache* genannt.

2.2.1.4. Quelle-Filter-Modell

Häufig wird die Sprachlauterzeugung des Menschen mit dem *Quelle-Filter-Modell* von Fant beschrieben [Fan70]. Ein *Quellsignal* oder *Anregungssignal* $s \in l^2(\mathbb{Z})$ regt ein System an, dass durch ein kontinuierliches LZI-System T mit Impulsantwort $h \in l^2(\mathbb{Z})$ gegeben ist. Für stimmlose Sprache wird das Quellsignal durch weißes Rauschen mit einem flachen Amplitudengang modelliert. Für stimmhafte Sprache wird das Quellsignal aus einem Modell der Glottis abgeleitet. Das Filter T beschreibt die Eigenschaften des Vokaltraktes sowie die Schallabstrahlung am Mund. Das Filter T ändert sich mit der Zeit, da die Vokaltraktstellung beim Sprechen variiert. In dem Modell wird angenommen, dass Quelle und Filter unabhängig voneinander sind. Die Nasenhöhlen werden bei der Modellierung des Vokaltraktes nicht berücksichtigt.

2.2.2. Sprachlautwahrnehmung

Schall verarbeitet der Mensch im *auditiven System (Hörorgan)*. Das auditive System besteht aus dem *peripheren Hörorgan (Ohr)* und dem *zentralen Hörorgan* (zum Hören erforderlichen Teile des Zentralnervensystems) [PN91]. Im Folgenden wird zunächst das Ohr und seine Funktionen näher beschrieben, im Anschluss werden einige Aspekte des zentralen Hörorgans erläutert.

2.2.2.1. Das Ohr

Das Ohr besteht aus *Außenohr, Mittelohr und Innenohr*, siehe Abbildung 2.7. Eine Hauptfunktion des Außen- und Mittelohres ist es, Schall auf das Innenohr zu übertragen und zu verstärken. Im Innenohr wird der Schall dann analysiert. Vom Außenohr gelangt der Schall über den *Gehörgang* zum *Trommelfell*. Das Trommelfell wird in Schwingung versetzt. Diese Schwingung wird im Mittelohr über *Hammer, Amboß und Steigbügel* auf das *ovale Fenster* übertragen. Das ovale Fenster bildet den Übergang zum Innenohr. Im Innenohr befindet sich die *Cochlea (Gehörschnecke)*, die die Form einer Schnecke mit zweieinhalb Windungen hat. In Abbildung 2.8 findet sich ein Querschnitt durch die Cochlea. Der *Ductus Cochlearis (Endolymphschlauch)* in

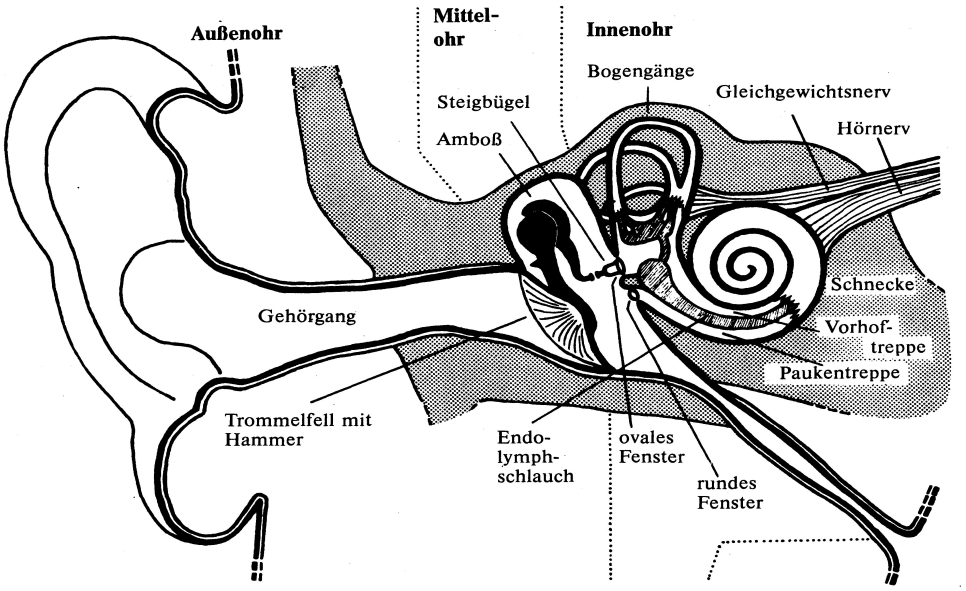


Abbildung 2.7.: Das periphere Hörorgan (das Ohr). (Quelle: [PN91])

der Mitte der Cochlea trennt die *Scala vestibuli (Vorhoftreppe)* von der *Scala tympani (Paukentreppe)*. Der Endolymphschlauch wird auf der Seite der Vorhoftreppe durch die *Reissnersche Membran* begrenzt. Auf der anderen Seite wird er von der *Basilarmembran* begrenzt. Auf der Basilarmembran liegt das *Cortische Organ*. Es besteht aus sogenannten *inneren* und *äußeren Haarzellen*, sowie aus der *Tektorialmembran*, die über den Haarzellen liegt. Die Bewegungen des *Steigbügels* werden über das *ovale Fenster* auf die Cochlea übertragen. Laut der *Wanderwellentheorie* von Békésy wird dadurch die Basilarmembran in Schwingung versetzt [PM95]. Die sogenannten *Wanderwellen* laufen entlang der Basilarmembran und erreichen ihre maximale Amplitude an gewissen Orten der Basilarmembran und laufen dann schnell aus. Die Orte, an denen die Wanderwellen ihre maximale Amplitude erreichen, sind abhängig von der Frequenz der Steigbügelbewegungen, siehe Abbildung 2.9. Laufen die Wanderwellen entlang der Basilarmembran, wird die Basilarmembran relativ zur Tektorialmembran bewegt [Fla65]. Die Haarzellen, die zwischen Tektorialmembran und Basilarmembran liegen, werden eingedrückt. An den Orten maximaler Amplitude der Wanderwellen, werden die Haarzellen am stärksten eingedrückt. Die Haarzellen werden durch *Nervenfasern* innerviert. Das Abknicken der Haarzellen regt das Feuern neuronaler Signale an. Die Häufigkeit, die Orte und die Zeitpunkte zu denen die Neurone feuern sind im Wesentlichen die Informationen, die im Cortischen Organ aus dem Schall gewonnen werden. Jede Haarzelle hat eine *charakteristische Frequenz*, bei der sie am Stärksten auf Anregungssignale reagiert. Diese charakteristische Frequenz ergibt sich aus dem Ort, an dem sie auf der Basilarmembran liegt. Schall wird daher entlang der Basilarmembran in seine Frequenzanteile zerlegt. Benachbarte Orte auf der Basilarmembran entsprechen dabei

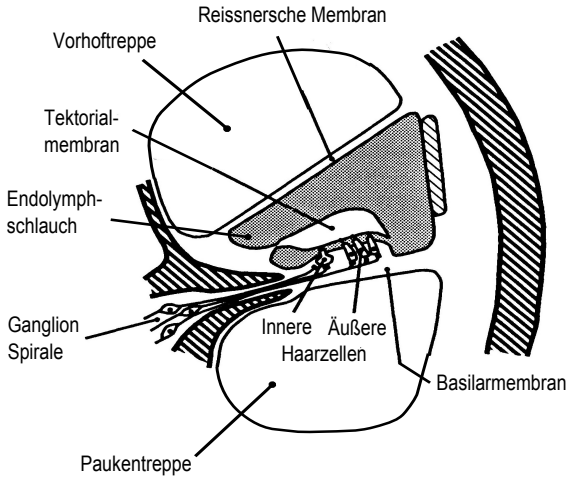


Abbildung 2.8.: Querschnitt durch die Cochlea. (Quelle: [PM95], bearbeitet)

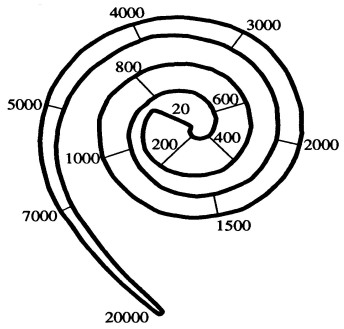


Abbildung 2.9.: Abbildung der Frequenzen (in Hz) entlang der Basilarmembran. (Quelle: [PM95])

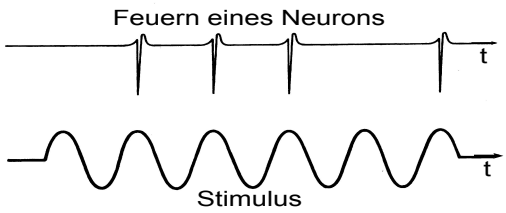


Abbildung 2.10.: Phase-locking [SKD97]. Eine Sinusschwingung regt die Cochlea an (unten). Ein Neuron feuert synchron zu der Sinusschwingung (oben). Kann ein Neuron nicht im richtigen Moment feuern, wird eine Periode übersprungen. (Quelle: [SKD97])

benachbarten Frequenzen. Diese Nachbarschaftsbeziehung wird auch als *Tonotopie* bezeichnet [PM95]. Die Häufigkeit mit der ein Neuron feuert korreliert mit der Amplitude des Schalls und mit der Nähe der Anregungsfrequenz zu der charakteristischen Frequenz der entsprechenden Haarzelle[Gre95]. Ein Neuron feuert besonders häufig, wenn ein Signal seiner charakteristischen Frequenz gerade einsetzt. Bleibt die Amplitude dann konstant, geht die Häufigkeit der Entladungen wieder zurück und bleibt dann auf einem gewissen Niveau. Dieser Vorgang wird *Adaption* genannt [Ste98]. Die Zeitpunkte, zu denen ein Neuron feuert, korellieren mit der Wellenform des Schalls [Gre95]. Dieses Phänomen wird *phase-locking* genannt (siehe Abbildung 2.10) [Gre95]. Neurone können nicht häufiger als einmal pro Millisekunde feuern [SKD97]. Um dennoch phase-locking für Frequenzen größer als 1 Khz zu erreichen, feuern Neurone synchron zu vielfachen der Anregungsperiode [SKD97]. Robustes phase-locking ist so bis zu Frequenzen von 2,5 kHz möglich [Gre95].

2.2.2.2. Die Hörbahnen

Die *Hörbahnen* sind der zentralnervöse Teil des auditorischen Systems. Die Hörbahnen führen über verschiedene Stationen von der *Medulla Oblongata* über das *Mittelhirn* bis zum *auditiven Cortex*. Die einzelnen Verarbeitungsschritte in der Hörbahn sind noch nicht vollständig verstanden [SKD97]. An dieser Stelle gehe ich nur auf einige Aspekte der Hörbahnen ein. Es gibt zum einen *afferente Hörbahnen*, die von der Cochlea zum auditiven Cortex führen. Andererseits gibt es *efferente Hörbahnen* die in umgekehrter Richtung verlaufen. Beispielsweise existieren efferente Verbindungen zu den Haarzellen [SKD97]. Durch diese kann neuronale Aktivität gehemmt werden [SKD97]. Es wird vermutet, dass hierdurch gezielt eine höhere Frequenzauflösung erreicht werden kann und Störsignale ausgeblendet werden können [Ste98]. Die Tonopie auf der Basilarmembran wird durch die Projektionen der afferenten Hörbahnen beibehalten [SKD97]. Auch in höheren Verarbeitungsschichten sind benachbarte Frequenzen räumlich benachbart.

2.2.2.3. Psychoakustik

Regt Schall in Form einer Sinusschwingung mit einer gewissen Frequenz und Amplitude das auditive System an, so kann man den Schalldruckpegel dieses akustischen

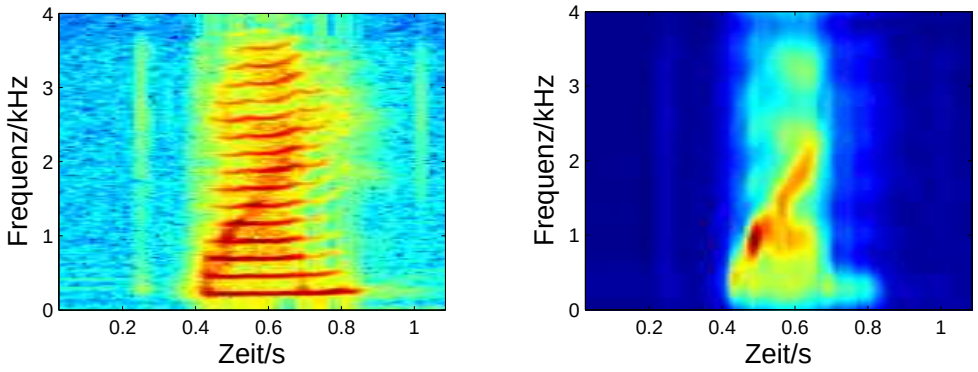
Signals messen. Die korrespondierende psychoakustische Größe zur physikalischen Größe des Schalldruckpegels ist die *Lautstärke*. Änderungen des Schalldruckpegels nimmt der Mensch als Änderung der Lautstärke wahr. Die korrespondierende psychoakustische Größe zur Frequenz ist die *Tonhöhe*. Änderungen der Frequenz nimmt der Mensch als Variation der Tonhöhe wahr. In der Psychoakustik wird unter anderem empirisch untersucht, wie Schalldruckpegel und Frequenz in Zusammenhang mit der Wahrnehmung von Lautstärke und Tonhöhe stehen. Es hat sich gezeigt, dass ein Anstieg des Schalldruckpegels um 10 dB in etwa eine Verdopplung der wahrgenommenen Lautstärke zur Folge hat [Ste98]. Eine Tonhöhe nimmt der Mensch bei periodischen Signalen wahr. Bei Grundfrequenzen bis zu 500 Hz besteht ein linearer Zusammenhang zwischen Grundfrequenz und empfundener Tonhöhe. Bei höheren Frequenzen besteht ein logarithmischer Zusammenhang zwischen Grundfrequenz und Tonhöhe [Ste98].

2.3. Automatische Spracherkennung

Die Aufgabe der *automatischen Spracherkennung* ist es, die korrekte textuelle Darstellung des Gesprochenen zu rekonstruieren [ST95]. Im engeren Sinn hat automatische Spracherkennung also zunächst wenig mit dem Verstehen von Sprache zu tun. Durch die Berücksichtung von Kontextinformationen zur Spracherkennung wird diese Abgrenzung jedoch verwaschen.

Ein üblicher Weg zur automatischen Spracherkennung ist eine *wahrscheinlichkeitsorientierte* Spracherkennung [ST95]. Ein Sprachsignal wird in einem *Frontend* in eine Folge von *Merkmalsvektoren* umgewandelt. Diese Folge wird dann in einem *Backend* mit Mitteln der Stochastik analysiert, um aus der Folge von Merkmalsvektoren die wahrscheinlichste *Wortfolge* zu schätzen, die zu den beobachteten Merkmalsvektoren führt [ST95]. Dazu werden häufig sogenannte *Hidden Markov Modelle* trainiert und eingesetzt [ST95]. Oft verwendete Merkmalsvektoren sind die sogenannten *Mel-Frequenz-Cepstrum-Koeffizienten (MFCC)*. Dabei wird ein Signal in den Frequenzbereich transformiert. Der Phasengang wird ignoriert, der Amplitudengang wird an die Tonhöhenempfindung des Menschen angepasst. Dazu kann beispielsweise eine Mel-Frequenz Filterbank verwendet werden. Im Anschluss wird eine *Lautheitstransformation* durchgeführt, um das Spektrum an das Lautstärkeempfinden des Menschen anzupassen. Dies kann annäherungsweise durch den natürlichen Logarithmus geschehen. Dieses reelle Spektrum wird dann in den Zeitbereich transformiert. Das entspricht einer Frequenzanalyse des reellen Spektrums, daher wird diese Darstellung auch reelles *Cepstrum* genannt, in Anlehnung an den Begriff Spektrum. Die Frequenzen werden im Cepstrum als *Quefrenzen* bezeichnet. Niedrige Frequenzen des Cepstrums entsprechen den langsamen Änderungen der Filtereigenschaften des Vokaltraktes in Frequenzrichtung (siehe Abbildung 2.11(b)). In den höheren Quefrenzen befindet sich ein Maximum, dass die harmonische Struktur des Quellsignals beschreibt. Die niedrigen Quefrenzen des Cepstrums werden als Merkmale verwendet, da die Filtereigenschaften die Stellung des Vokaltraktes beschreiben.

In der Praxis wird ein zeitdiskretes reelles Sprachsignal durch Abtasten der gefensterten Fourier-Transformation in *Frames* unterteilt (siehe Abschnitt 2.1.5). Im



(a) Amplitudengang (b) Liftering

Abbildung 2.11.: Gefensterte Fouriertransformation und Liftering. In a) ist der Amplitudengang einer gefensterten Fourier-Transformation eines Sprachsignals dargestellt. In b) ist ein gefilterter (liftering) Amplitudengang zu sehen, in dem der (hochfrequente) Einfluss der Quelle entfernt wurde. Von rot über gelb nach blau nimmt die Magnitude ab.

Standard ES 201 108 der European Telecommunications Standards Institute (ETSI) wird als Fensterfunktion ein Hammingfenster mit einem Träger von 25 ms gewählt, mit einem Abstand von 10 ms zwischen zwei Frames [Eur03a]. Bei einem Sprachsignal mit einer Abtastrate von 8 kHz entspricht das einer *Fensterlänge* von 200 Abtastpunkten und einem *Frameabstand* von 80 Abtastpunkten. Die DFTen der Frames nähern dann die Frequenzbereichdarstellung von Ausschnitten des Signals an, auf denen die MFCC berechnet werden: Nach dem Anwenden der Mel-Frequenz-Filterbank und der Lautheitstransformation kann das Cepstrum mittels der inversen DFT oder der inversen Diskreten Kosinustransformation berechnet werden (weil das reelle Cepstrum berechnet wird [ST95]). Als Merkmalsvektor für einen Frame werden die ersten Koeffizienten des Cepstrums gewählt, beispielsweise die ersten dreizehn Koeffizienten. Oft wird noch ein weiterer Koeffizient hinzugefügt, der die Energie erfasst, die in einem Frame vorhanden ist. Zusammenhänge zwischen aufeinanderfolgenden Frames werden häufig durch *Delta-* und *Delta-Delta* Koeffizienten berücksichtigt [ST95].

Kapitel 3.

Verwandte Arbeiten

In diesem Kapitel werden Arbeiten vorgestellt, die der vorliegendenden Arbeit thematisch verwandt sind. Darunter sind auch solche, deren Ergebnisse und Verfahren in der Diplomarbeit verwendet werden. Die verwandten Arbeiten sind in drei Gruppen eingeteilt. In Abschnitt 3.1 werden Verfahren vorgestellt, die Grundfrequenz und Verschlussmomente bei stimmhafter Sprache erkennen. In Abschnitt 3.2 werden verwandte Arbeiten zur robusten automatischen Spracherkennung vorgestellt. Diese Verfahren nutzen teilweise die Eigenschaften des Quellsignals, um robuste Merkmalsvektoren zu extrahieren. Schließlich werden in Abschnitt 3.3 Verfahren vorgestellt, die aus dem Phasengang Merkmale zur Spracherkennung extrahiert.

3.1. Das Quellsignal

Es existieren verschiedene Ansätze, die das Quellsignal stimmhafter Sprache für einen einzigen Phonationszyklus modellieren. Ein gängiger Ansatz ist das LF-Modell von Liljencrants und Fant [FLL85]. In diesem Abschnitt werden im Gegensatz dazu Verfahren vorgestellt, welche die Frequenz und die Lage der Phonationszyklen im Sprachsignal schätzen. Das heißt, es wird versucht, die Grundfrequenz und die Verschlussmomente zu finden (siehe Abschnitt 2.2.1.2). Als Bedingung für die Existenz der Grundfrequenz muss eine Entscheidung getroffen werden, ob ein Sprachsegment stimmhaft (*voiced*) oder nicht stimmhaft (*unvoiced*) ist.

3.1.1. Grundfrequenzschätzung

Die Verfahren zur Schätzung der Grundfrequenz f_0 eines Sprachsignals können in zwei Gruppen unterteilt werden. Die erste Gruppe nutzt die Periodizität des Sprachsignals im Zeitbereich aus, um die Grundfrequenz zu schätzen. Die zweite Gruppe verwendet die harmonische Struktur des Sprachsignals im Frequenzbereich. Generell ist die Grundfrequenz nur dort definiert, wo Sprache stimmhaft ist. Auch dann ist das Sprachsignal nicht perfekt periodisch. Kein Phonationszyklus gleicht einem anderen und auch die Länge von aufeinander folgenden Phonationszyklen ist nie identisch. Allerdings bestehen große Ähnlichkeiten. So ändert sich die Grundfrequenz innerhalb von 100ms selten um mehr als eine Oktave [DO03]. Im Allgemeinen ändert sich die Grundfrequenz bei gesprochener Sprache nur sehr langsam [DO03]. Über kurze Zeitintervalle kann stimmhafte Sprache annäherungsweise periodisch fortgesetzt werden. Um die Grundfrequenz zu einem gewissen Zeitpunkt festzustellen, wird dementsprechend häufig ein Ausschnitt des Signals um diesen Zeitpunkt herum untersucht, der

einige wenige Phonationszyklen enthält. Bei der Grundfrequenzschätzung können *Oktavenfehler* auftreten. Als Grundfrequenz wird dann ein Teiler oder ein Vielfaches der tatsächlichen Frequenz ausgegeben. Im Folgenden werden einige Methoden zur Grundfrequenzschätzung vorgestellt.

3.1.1.1. Zeitbereich-Methoden

Methoden, die im Zeitbereich für ein zeitdiskretes Sprachsignal verwendet werden, basieren häufig auf Abwandlungen der *Autokorrelations-Methode* von Fujisaki [Fuj60]. Zum Zeitpunkt $m \in \mathbb{Z}$ wird die Grundfrequenz bestimmt, indem die Kurzzeit-Autokorrelation berechnet wird. Besteht eine starke Korrelation zwischen dem Signal mit einer um T verschobenen Version seiner selbst, so wird vermutet, dass die Grundfrequenz $\frac{1}{T}$ beträgt. Maxima der Kurzzeit-Autokorrelations-Funktion sind daher Kandidaten für die Grundfrequenz.

Eine weitere Methode im Zeitbereich ist die *Average Magnitude Difference Function* (AMDF) von Ross et al. [RSC+74]. Hier wird vermutet, dass die Grundfrequenz $\frac{1}{T}$ beträgt, wenn die absolute durchschnittliche Differenz zwischen dem Signal und einer um T verschobenen Version seiner selbst gering ist. Der Unterschied zur Autokorrelations-Methode besteht hauptsächlich darin, wie die verschobenen Signale miteinander verglichen werden.

Ob ein Bereich der Sprache stimmhaft ist, kann mit beiden Methoden durch die Bewertungsfunktion ermittelt werden: Besteht große Ähnlichkeit mit dem verschobenen Signal, ist der Ausschnitt wahrscheinlich stimmhaft. Oktavenfehler tauchen bei beiden Methoden auf, denn auch bei Verschiebungen um Vielfache und Teiler der Grundfrequenz herrscht Ähnlichkeit.

Allgemein hängt die Genauigkeit der Grundfrequenzschätzung im Zeitbereich von der Abtastrate ab, was nachteilig ist. Durch Interpolation wird dieses Problem beispielsweise vom *Super Resolution Pitch Determination* Algorithmus von Medan et al. angegangen [MYC91]. Die Ähnlichkeit wird bei diesem Ansatz mittels einer Kreuzkorrelationsfunktion zwischen verschobenen, ausgeschnittenen Versionen des Sprachsignals bewertet (siehe Abbildung 3.1). Die Verschiebung und die Länge der Fenster hängt jeweils von der zu bewertenden Grundfrequenz ab. Die Bewertungsfunktion wird normalisiert, damit die unterschiedlichen Längen der Fenster und die unterschiedliche Verteilung der Energie im Signal ausgeglichen werden.

3.1.1.2. Frequenzbereich-Methoden

Wie bei der Merkmalsextraktion aus Sprachsignalen wird das Signal häufig in Frames mit konstantem Frameabstand zerlegt (siehe Abschnitt 2.1). Die Fensterlänge wird meist so gewählt, dass eine kleine Anzahl von Phonationszyklen hineinpasst. Je länger die Fenster sind, desto genauer ist die Frequenzauflösung. Andererseits können Grundfrequenzänderungen über die Zeit bei längeren Fenstern schwerer verfolgt werden. Ein genereller Nachteil der Frequenzbereich-Methoden liegt darin, dass für jeden Frame eine DFT berechnet werden muss.

Die *Cepstrummethode* von Noll wird den Frequenzbereich-Methoden zugeordnet [Nol67]. Es wird ausgenutzt, dass im Cepstrum die harmonische Struktur stimm-

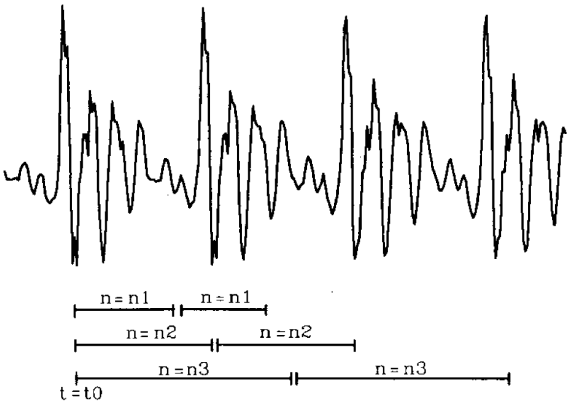


Abbildung 3.1.: Super Resolution Pitch Determination Algorithmus [MYC91]. Ein Ausschnitt stimmhafter Sprache wird analysiert. Die normalisierte Kreuzkorrelation wird jeweils zwischen zwei Ausschnitten gleicher Länge berrechnet. Die Grundfrequenz liegt in diesem Beispiel bei einer Fensterlänge und einer Verschiebung von n=n2 Abtastpunkten. Dies ist der erste Schritt im Super Resolution Pitch Determination Algorithmus. Mittels Interpolation wird die Genauigkeit dann noch erhöht. (Quelle: [MYC91], bearbeitet)

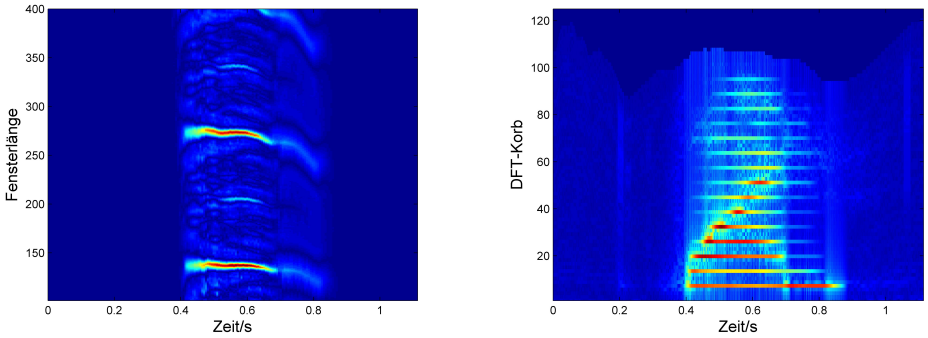
hafter Sprache als ein Peak bei entsprechender Quefrenz auftaucht (siehe Abschnitt 2.1).

Die folgenden Arbeiten zur Grundfrequenzschätzung sind für diese Arbeit besonders wichtig, weil grundlegende Prinzipien auch in der vorliegenden Arbeit verwendet werden. Es handelt sich um eine *pitch-scaled (grundfrequenz-skalierte)* Methode zur Grundfrequenzbestimmung von Muta et al. [MBFS88]. Der Begriff grundfrequenzskaliert bezieht sich darauf, dass die Länge (der Träger) der Fensterfunktion an ein Vielfaches eines Phonationszyklus angepasst wird. Bei der *pitch-synchronous (glottissynchronen)* Analyse ist die Fensterfunktion zusätzlich mit der Lage des Phonationszyklus synchron [JM08], das ist hier nicht Fall. Die grundfrequenz-skalierte und glottissynchrone Analyse weicht von der gefensterten Fourier-Transformation in der Weise ab, dass die Fensterlängen bzw. die Frame-Abstände variabel sind.

Muta et al. schlagen vor, für jeden Frame zunächst mit einer iterativen Autokorellations-Methode eine initiale Fensterlänge zu schätzen, die vier Phonationszyklen enthält (siehe Zeitbereich-Methoden). Ist diese Initialisierung gefunden, wird die Grundfrequenzschätzung verfeinert, indem der Leakage-Effekt ausgenutzt wird. Als Fensterfunktion wird ein *Hanning-Fenster* benutzt. Ist die Fensterlänge l tatsächlich entsprechend der Grundfrequenz gewählt, sind die *harmonischen Frequenzen* (vergleiche Abschnitt 2.1):

$$f_{analysis}(4 \cdot j, l), j \in \{1, ..., H_l := \lfloor \frac{1}{4} \lceil l/2 \rceil \rfloor - 1\}. \tag{3.1}$$

Das Signal ist nicht nur l-periodisch, sondern auch $\frac{l}{4}$-periodisch. Im Idealfall liegt die gesamte Energie (die *harmonische Energie*) in den harmonischen Frequenzen. Die



(a) Harmonische Energie

(b) Grundfrequenz-skalierte gefensterte Fourier-Transformation

Abbildung 3.2.: Grundfrequenz-skalierte Verarbeitung nach Behnke [Beh03]. Links ist eine harmonische Oberfläche zu sehen: Jedem Zeitpunkt und jeder Fensterlänge bzw. Grundfrequenz ist eine harmonische Energie zugeordnet. Die Grundfrequenz liegt hier bei einer Fensterlänge von ungefähr einhundertfünfzig Abtastpunkten. Energie fällt zusätzlich in großem Maß auf benachbarte Oktaven. Rechts ist der Amplitudengang einer grundfrequenz-skalierten gefensterten Fourier-Transformation mit rechteckiger Fensterfunktion dargestellt. Es fällt die Konzentration der Energie in den harmonischen Körben auf, die durch die grundfrequenz-skalierte Verarbeitung und die verwendete Fensterfunktion bedingt ist. Von rot über gelb nach blau nimmt die Magnitude ab.

harmonischen Frequenzen werden bei einer DFT von den *harmonischen Körben*

$$4 \cdot k, k \in \{1, ..., H_l\} \tag{3.2}$$

angenähert (vergleiche Abschnitt 2.1).

Durch den Leakage-Effekt landet nicht die gesamte harmonische Energie in den harmonischen Körben, sondern es es landet auch harmonische Energie in den Körben $4 \cdot k - 1$ und $4 \cdot k + 1$, $k \in \{1, ..., H_l\}$. Wieviel Energie neben den harmonischen Körben landet, kann anhand der Energie in den harmonischen Körben und dem Wissen über das Spektral-Leakage berechnet werden (siehe Abbildung 2.1). Weicht die tatsächliche Energie in den nicht-harmonischen Körben von der berechneten Energie ab, so ist dies ein Hinweis darauf, dass die Fensterlänge nicht exakt vier Phonationszyklen fasst. Die Differenz zwischen der berechneten und der gemessenen Energie in den nicht-harmonischen Körben wird mittels Newton-Verfahren minimiert. In jedem Iterationsschritt muss hierzu eine DFT der neu geschätzten Länge berechnet werden.

Jackson und Shadle nutzten den Algorithmus von Muta et al. für ein *pitch-scaled harmonic filter* (PSHF), das ein Sprachsignal in einen harmonischen und einen nicht-harmonischen Anteil zerlegt [JS01]. Ein *harmonischer Filter* ist ein Filter, der die harmonische Struktur eines Signals für eine bestimmte Grundfrequenz f_0 passieren lässt, und andere Frequenzanteile dämpft. Jackson und Shadle berechneten für jeden Frame mit der oben beschriebenen Methode die Grundfrequenz. Die Energie in den harmonischen und nicht-harmonischen Körben verwendeten sie jeweils als Schätzung

für den harmonischen und nicht-harmonischen Anteil der Sprache. Da auch in den harmonischen Körben Energie liegt, die nicht zum Sprachsignal gehört, interpolierten Jackson und Shadle diese Werte anhand der umliegenden nicht-harmonischen Körbe.

Behnke erweiterte beide Methoden, um sie robuster gegenüber Rauschen zu machen [Beh03]. Er verwendete kein Hanning-Fenster, sondern ein rechteckiges Fenster. Dies hat zur Folge, dass die harmonische Energie bei entsprechender Fensterlänge ausschließlich in die harmonischen Körbe $4 \cdot k$, $k \in \{1, ..., H_l\}$ fällt. Die Energie in den nicht-harmonischen Körben wurde ähnlich zu dem Ansatz von Jackson und Shadle als lokale Schätzung des Rauschens für die benachbarten harmonischen Körbe aufgefasst. Als Erweiterung zu Muta et al. wurde das geschätzte Rauschen in den harmonischen Körben abgezogen. Die Energie der bereinigten harmonischen Körbe wird dann verwendet, um die Fensterlänge l zu bewerten (siehe Abbildung 3.2). Ist viel Energie vorhanden, ist vermutlich die Grundfrequenz gefunden. Zu bedenken ist bei diesem Ansatz, dass auch der nicht-stimmhafte Anteil der Sprache von den harmonischen Koeffizienten abgezogen wird und daher als Rauschen aufgefasst wird. Ist die Grundfrequenzschätzung nicht korrekt, wird auch ein Teil der stimmhaften Sprache abgezogen. Um Oktavenfehler zu vermeiden, schlagen Roa, Bennewitz und Behnke Oktavenfilter vor [RBB07].

3.1.1.3. Grundfrequenz-Verfolgung

Die oben vorgestellten Verfahren zur Grundfrequenzschätzung liefern lokal (für einen Frame) eine Aussage darüber, wie wahrscheinlich verschiedene Grundfrequenzen sind. Die Maxima können von Frame zu Frame zwischen weit enfernten Grundfrequenzen springen, beispielsweise durch Oktavenfehler. Häufig wird dynamische Programmierung oder ein Median-Filter angewandt, um diese Sprünge zu vermeiden [GR00]. Dabei fließt zumeist die Annahme ein, dass die Grundfrequenz sich von Frame zu Frame nur wenig ändert. Behnke verwendete dynamische Programmierung, um einen Grundfrequenzpfad zu finden. Dazu wird die gesamte harmonische Oberfläche für ein Sprachsignal berechnet (siehe Abbildung 3.2) [Beh03]. Der RAPT Algorithmus von Talkin verwendet ebenfalls dynamische Programmierung in Verbindung mit einer Grundfrequenzschätzung anhand einer modifizierten Korrelationsmethode (NCCF) [Tal95, GR00]. Der Super Resolution Pitch Determination Algorithmus ist ein Echtzeit-Verfahren und grenzt den Suchraum für Grundfrequenzen ein, so dass sich aufeinanderfolgende Grundfrequenzschätzungen nur wenig unterscheiden können [MYC91]. Das hat den Nachteil, dass die richtige Grundfrequenz gegebenenfalls nicht im Suchraum enthalten ist.

3.1.2. Verschlussmomentdetektion

Verfahren, die den Verschlussmoment detektieren, können ebenfalls in Zeit- und Frequenzbereich-Methoden unterteilt werden. Im Zeitbereich wird die Energiekonzentration beim Verschlussmoment ausgenutzt. Im Frequenzbereich weist der Phasengang auf Verschlussmomente hin. Verfahren, die Verschlussmomente detektieren, schätzen gleichzeitig die Grundfrequenz: Die Grundfrequenz $1/T$ entspricht der Dauer T zwischen zwei Verschlussmomenten (siehe Abschnitt 2.2.1.2). Manchmal führen

die Methoden zunächst eine inverse Filterung des Sprachsignals auf Grundlage des Quelle-Filter-Modells durch. Dazu wird *LPC (Linear Predictive Coding)* verwendet: Das Sprachsignal wird in ein Restsignal und in Koeffizienten aufgeteilt. Das Restsignal entspricht dem Anregungssignal, die Koeffizienten beschreiben die Filtereigenschaften [Wie66]. Dies hat den Vorteil, dass im Restsignal der Einfluss des Filters entfernt ist. Allerdings ist LPC nur eine näherungsweise Lösung für die inverse Filterung. Häufig gehen die Verfahren zur Detektion der Verschlussmomente Abtastpunkt für Abtastpunkt vor: Zum Zeitpunkt des Verschlussmoments weisen die Verfahren eine Besonderheit auf, die auf den Verschlussmoment schließen lässt. Verfahren im Zeitbereich unterteilen das Sprachsignal nicht notwendigerweise in Frames. Es muss kein Fenster mit ausreichendem Support gewählt werden, damit das Spektrum in ausreichender Auflösung geschätzt werden kann. Die Fensterlänge für Frequenzbereich-Methoden erstreckt sich in der Regel über mehrere Phonationszyklen, was die Schätzung für eine Periode ungenauer macht.

3.1.2.1. Zeitbereich-Methode

Eine Methode von Whitman und Etter im Zeitbereich beruht auf dem sogenannten *Teager Energie Operator (TEO)* [WE94]. Dieser ist für ein reelles zeitdiskretes Signal $x \in l^2(\mathbb{Z})$ ein Maß der Energie:

$$E_{Teager}(n) = x(n)^2 - x(n-1) \cdot x(n+1),\ x \in \mathbb{Z} \tag{3.3}$$

Das Maß verwendet nur je zwei benachbarte Werte und ist somit lokal. Maxima dieser Energiefunktion sind Kandidaten für Verschlussmomente. Der ETSI Standard ES 202 050 verwendet den geglätteten TEO zur Lokalisierung von Verschlussmomenten [Eur07]. Da der TEO Operator sehr lokal ist, weist er auch Maxima auf, die keinen Verschlussmomenten entsprechen. Umso stärker ein Maximum ausgeprägt ist, desto wahrscheinlicher ist ein Verschlussmoment gefunden.

3.1.2.2. Frequenzbereich-Methoden

Yegnanarayana und Smits entwickelten eine Methode zur Verschlussmomentdetektion, die im Frequenzbereich arbeitet [YS95, YM99]. Die Methode arbeitet auf dem Restsignal des LPC (siehe oben). Das Restsignal wird in Frames aufgeteilt. Jeder Frame enthält wenige (1-2) Phonationszyklen und der Frameabstand beträgt einen Abtastpunkt. Für jeden Frame wird mittels einer DFT die Darstellung im Frequenzbereich geschätzt. Die Gruppenlaufzeit wird mit Gleichung 3.8 berechnet. Ausreisser in der Gruppenlaufzeit werden mit einem Median Filter entfernt. Es wird dann die durchschnittliche Gruppenlaufzeit berechnet. Schließlich wird jedem Frame in einer *Phasensteigungs-Funktion* diese durchschnittliche Gruppenlaufzeit zugeordnet. Die positiven Nulldurchgänge dieser Funktion sind Kandidaten für Verschlussmomente [YM99]. Es wird hierbei angenommen, dass ein Frame des Restsignals ein Minimalphasensignal ist, wenn der Verschlussmoment der erste Abtastpunkt eines Frames ist.

Brookes, Naylor und Gudnason schlagen einen Energie-gewichteten Durchschnitt der Gruppenlaufzeit vor, um Ausreißer bei der Bestimmung der durchschnittlichen

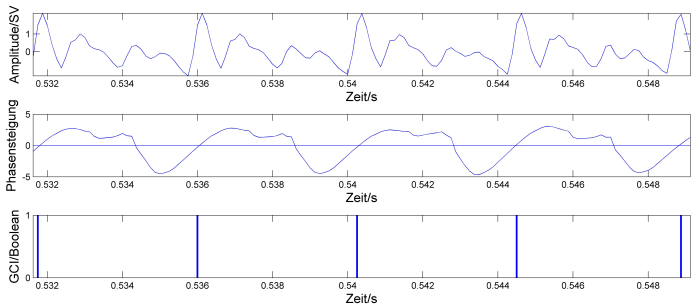


Abbildung 3.3.: Verschlussmomentdetektion mit dem DYPSA Algorithmus [NKGB07]. Ein stimmhafter Ausschnitt aus einem Sprachsignal ist oben abgebildet. In der Mitte findet sich die geglättete Phasensteigungsfunktion. Die positiven Nulldurchgänge markieren Kandidaten für Verschlussmomente. In diesem Fall wählte DYPSA alle diese Kandidaten als Verschlussmomente (unten).

Gruppenlaufzeit zu dämpfen [BNG06]. Die vorgeschlagene Durchschnittsbildung entspricht der Berechnung des *Center of Gravity* im Zeitbereich [BNG06].

3.1.2.3. Verschlussmoment-Verfolgung

Die oben beschriebenen Verfahren liefern jeweils Kandidaten für Verschlussmomente (*GCI-Kandidaten*). Kounoudes, Naylor und Brookes entwickelten auf Basis der Gruppenlaufzeit-Methode von Yegnanarayana und Smits den DYPSA Algorithmus, der GCI-Kandidaten Kosten zuordnet und mittels dynamischer Programmierung GCI-Kandidaten auswählt [KNB02]. In neuerer Zeit wurde DYPSA von Naylor, Kounoudes, Gudnason und Brookes erweitert, indem die energie-gewichtete Gruppenlaufzeit verwendet wird [NKGB07] (siehe Abbildung 3.3). Die Kostenfunktion für die dynamische Programmierung berücksichtigt unter anderem, dass

- Abstände zwischen benachbarten Verschlussmomenten ähnlich sind
- die Energie im Zeitbereich zum Verschlussmoment groß ist
- Ähnlichkeit zwischen zwei aufeinanderfolgenden Phonationszyklen herrscht

Die GCI-Kandidaten werden mittels der energie-gewichteten Gruppenlaufzeit-Methode bestimmt, dabei werden Fenster verwendet, die höchstens einen Phonationszyklus enthalten. Die Kostenfunktion berücksichtigt die oben genannten Punkte.

Auch die Methode von Whitman und Etter (siehe oben) beinhaltet einen Algorithmus, der GCI-Kandidaten auswählt und dann im Wesentlichen eine Maximumsuche durchführt [WE94]. Ein Algorithmus von Abu-Shikhah und Deriche berücksichtigt zusätzlich, dass Verschlussmomente einen gewissen Minimal- und Maximalabstand zueinander haben [ASD99]. In leicht abgewandelter Form wird der Algorithmus von Abu-Shikhah und Deriche im ETSI Standard ES 202 050 verwendet [Eur07].

3.2. Rauschunterdrückung

Liegt ein Satz in textueller Darstellung vor, geht das in Abschnitt 2.3 beschriebene Verfahren zur Extraktion von Merkmalsvektoren davon aus, dass sich die Merkmalsvektoren für verschiedene ausgesprochene Versionen dieses Satzes ähneln. Auch dann ähneln sich die Vektoren, wenn das Sprachsignal mit variierender Grundfrequenz, von verschiedenen Sprechern, in unterschiedlicher Lautstärke oder mit anderen Variationen vorliegt. Eine gewisse Variabilität der Folgen von Merkmalsvektoren lässt sich durch das Verwenden von vielen Beispielen beim Trainieren der Hidden Markov Modelle erfassen (siehe Abschnitt 2.3). Dennoch ist es vorteilhaft, wenn sich Folgen von Merkmalsvektoren, denen die gleiche textuelle Darstellung zugrunde liegt, möglichst ähnlich sind. Daher werden Verfahren zur *Normalisierung* verwendet, um *robuste automatische Spracherkennung* zu erreichen [BDMD+06].

Verfahren, die automatische Spracherkennungssysteme robust gegenüber *Rauschen* und *Nachhall* machen, sind der Inhalt dieses Abschnitts.

Rauschen wird als additive Überlagerung des Sprachsignals durch Störsignale angesehen. Nachhall wird oft durch ein kontinuierliches LZI-System T modelliert. Ein durch Rauschen und Nachhall *lärmbehaftetes* Signal $y \in l^2(\mathbb{Z})$ ergibt sich annäherungsweise als

$$y = x * r + l, \text{ für } x, r, l \in l^2(\mathbb{Z}). \tag{3.4}$$

x entspricht dem Sprachsignal, so wie es die Lippen des Sprechers verlässt, r ist die Impulsantwort des Systems T und l ist ein Störsignal. Zusätzlich ist zu berücksichtigen, dass sich das System T im Allgemeinen mit der Zeit ändert, wenn auch langsam. Für das Rauschen l wird unterschieden ob dieses *stationär* ist oder nicht. Häufig wird angenommen, dass l stationär ist.

Eine Möglichkeit zur robusten Spracherkennung gegenüber Lärm ist es, entsprechend lärmbehaftete Trainingsbeispiele zu verwenden (*Multi-Condition Training*) [PgHG00]. Andere Methoden versuchen, Rauschen und Nachhall aus dem Sprachsignal zu entfernen, so dass die resultierenden Merkmalsvektoren denen für ein Sprachsignal ohne Rauschen und ohne Nachhall möglichst ähnlich sind. Der Erfolg der letztgenannten Methoden kann zum einen durch Hörversuche evaluiert werden, zum anderen durch die Erkennungsleistung von automatischen Spracherkennern [DO03].

Spectral Subtraction ist eine der am weitesten verbreiteten Methoden zur Rauschunterdrückung, die ursprünglich Boll vorschlug [Bol79]. Die Grundidee besteht darin, in der Frequenzbereichsdarstellung additives Rauschen zu entfernen:

$$X(\omega) = Y(\omega) - R(\omega) \tag{3.5}$$

$X, Y, R \in L^2([0;1])$ sind die Fourier-Transformierten des Sprachsignals x, des lärmbehafteten Signals y und des Rauschens r. Dabei kann y als gegeben vorausgesetzt werden, weil es das beobachtete Signal darstellt. In der Praxis wird das Signal y oft durch Fensterung in Frames unterteilt. Für ein Frame w wird das Spektrum Y_w mit einer DFT geschätzt. Die Herausforderung besteht darin, eine Schätzung für das Rauschen $\hat{R}_w$ für diesen Frame zu finden. Häufig wird angenommen, dass das Rauschen stationär ist und daher das Spektrum des Rauschens in Frames erhalten werden kann, in denen keine Sprache vorhanden ist [LO79]. Der Erwartungswert des

Rauschens in Frames ohne Sprache kann als Rauschschätzung verwendet werden. Der Phasengang wird ignoriert, da sein Erwartungswert nach Annahme null ist:

$$\hat{X}_w(k) = (|Y_w(k)| - |\hat{R}_w(k)|)e^{\angle Y_w(k)},\ \hat{X}_w, Y_w, \hat{R}_w \in l^2(\mathbb{Z}) \tag{3.6}$$

Hierdurch wird der Phasengang des lärmbehafteten Signals übernommen. Ein generelles Problem bei Methoden des Spectral Subtraction, die sich nur auf die Amplitude beziehen, ist die Tatsache, dass Auslöschungen nicht berücksichtigt werden [LL08]: Zeigen Nutzsignal und Rauschen in entgegengesetzte Richtungen, wird das Nutzsignal ausgelöscht, beim Spectral Subtraction wird dann zusätzlich der Erwartungswert des Rauschens abgezogen. Ein Ansatz, um dem zu begegnen, ist das *Magnitude Averaging* von Boll [Bol79]: Die Magnitude des Signals wird zeitlich geglättet, weshalb bei der Subtraktion des erwarteten Rauschens ein geringerer Fehler auftritt.

Da $|Y_w(k)| - |\hat{R}_w(k)|$ auch negative Werte annehmen kann, wird dieser Term oft nach unten begrenzt.

$$|\hat{X}_w(k)| = \max(|Y_w(k)| - |\hat{R}_w(k)|, \gamma|\hat{R}_w(k)|),\ \gamma \geq 0 \in \mathbb{R} \tag{3.7}$$

Die Beschränkung der Amplitude auf Werte größer Null (*flooring*) hat den sogenannten *Musical Noise* zur Folge, weil an einigen isolierten Stellen Energie übrig bleibt und kein Rauschteppich auf allen Frequenzen. *Nonlinear Spectral Subtraction (NSS)* ist eine der Methoden, die zur Unterdrückung des Musical Noise eingesetzt wird [LB92]: Bei niedrigem Signal-Rausch-Verhältnis in einem Frequenzbereich wird ein größerer Anteil des geschätzten Rauschens abgezogen, so dass im Idealfall nur noch dort Energie übrig bleibt, wo die Energie des Sprachsignals überwiegt.

Zur Rauschunterdrückung kann auch ein *Wiener-Filter* verwendet werden [Wie66]. Es wird ein Filter geschätzt, der ein lärmbehaftetes Signal bereinigt. Ein linearer Wiener-Filter ist optimal bezüglich der durchschnittlichen quadratischen Abweichung zwischen klarem Signal und gefiltertem Signal. Um ein Wiener-Filter zu berechnen, ist wie beim Spectral Subtraction eine Schätzung des Rauschens erforderlich. Bei der framebasierten Verarbeitung von Signalen wird in der Regel ein adaptiver Wiener-Filter verwendet, der für jeden Frame auf Basis einer adaptiven Rauschschätzung entworfen wird. Auch der Wiener Filter leidet unter Musical Noise: Der Wiener-Filter kann als Spezialfall des Spectral Subtraction aufgefasst werden kann und es wird wie beim Spectral Subtraction eine künstliche untere Grenze für das gefilterte Signal verwendet. Der *Two-Stage Mel-Warped Wiener-Filter* von Agarwal und Cheng wendet den Wiener-Filter zweimal an [AAC99]. In der ersten Stufe wird der Wiener-Filter anhand der Rauschschätzung für stationäres Rauschen entworfen. Im gefilterteten Signal findet sich nach Annahme noch weißes Rauschen. In der zweiten Stufe wird anhand der Eigenschaften von weißem Rauschen ein Wiener-Filter entworfen, welcher das verbliebene Rauschen entfernen soll. Dies ist ein Ansatz, um auch nicht-stationäres Rauschen zu entfernen.

Ein direkter Ansatz, um nicht-stationäres Rauschen zu entfernen, ist das *Harmonic Tunneling* von Ealey, Kelleher und Pearce [EKP01]. Sie nutzen die harmonische Struktur stimmhafter Sprache aus: Im Spektrum eines Sprachsignals findet sich

zwischen den Harmonischen im Idealfall keine Energie. Die Energie zwischen den Harmonischen kann daher als Rauschschätzung verwendet werden, wenn davon ausgegangen wird, dass die Energie des Rauschens sich für benachbarte Frequenzen nur wenig unterscheidet.

Die folgenden Ansätze unterdrücken Lärm, indem sie nur die Anteile des Signals passieren lassen, die dem Sprecher zuzuordnen sind.

Der *RASTA-Filter* von Hermansky und Morgan ist im Wesentlichen ein Bandpass-Filter auf dem Spektrogram des Sprachsignals [HM94]. Der Vokaltrakt bewegt sich mit einer gewissen Geschwindigkeit. Effekte im Spektrogram, welche sich schneller oder langsamer ändern als dies bei Sprache der Fall ist, werden durch den RASTA-Filter entfernt. Hierdurch kann zum Beispiel Nachhall entfernt werden.

Ein weiterer Ansatz zum Hervorheben des Sprachsignals ist das *Adaptive Comb Filtering* von Frazier [Fra75]. Mittels eines Kammfilters, der an die Tonhöhe angepasst ist, werden bei stimmhafter Sprache die harmonischen Komponenten des lärmbehafteten Signals ausgewählt.

Behnke verwendete grundfrequenzskalierte Verarbeitung, um die harmonische Struktur besser aufzulösen und ließ nur eine gefilterte harmonische Struktur passieren [Beh]. Er kombinierte einen harmonischen Verarbeitungsweg für stimmhafte Sprache mit einem nicht-harmonischen Verarbeitungsweg zu einem *Harmonic Frontend*. Der nicht-harmonische Verarbeitungsweg verwendet Spectral Subtraction und einen Bandpassfilter ähnlich dem RASTA-Filter [Beh].

Bei stimmhafter Sprache ist die Energie in einigen Bereichen stärker konzentriert als in anderen. Dies nutzen Macho und Cheng beim *SNR-dependent waveform processing (SWP)* aus [MC01]. Das lärmbehaftete Signal wird mit einem Fenster gewichtet, so dass die energiereichen Anteile stimmhafter Sprache betont werden. SWP wird vom ETSI Standard ES 202 050 angewendet [Eur07]. SWP wird dort eingesetzt, nachdem ein *Two-Stage Mel-Warped Wiener-Filter* Rauschen entfernt hat, und somit die Schätzung der Verschlussmomente robuster wird.

3.3. Gruppenlaufzeit-Merkmal

Die aus dem Phasengang gewonnene Gruppenlaufzeit kann benutzt werden, um Formanten aus einem Sprachsignal zu extrahieren. Murthy, Murthy und Yegnanarayana schlagen hierzu vor, die Gruppenlaufzeit mittels einer DFT zu schätzen [MMY89]. Diese Berechnung ist fehleranfällig, da der geschätzte Phasengang kontinuierlich gemacht werden muss (*phase-unwrapping*, vgl. Abschnitt 2.1.3). Um phase-unwrapping zu vermeiden, berechnen sie die Gruppenlaufzeit direkt:

$$\tau(k) = \frac{X_R(k) \cdot Y_R(k) + X_I(k) \cdot Y_I(k)}{|X(k)|^2}, k = 0, ..., N-1 \tag{3.8}$$

Es ist $x \in \mathbb{R}^N$ ein Ausschnitt eines Sprachsignals, $y(n) = n \cdot x(n) \in \mathbb{R}^N$. X und Y sind die DFTen von x und y. R und I stehen für den Realteil und den Imaginärteil von X(k). Da der Teiler in Gleichung 3.8 Null werden kann, legen Murthy, Murthy und Yegnanarayana einen Minimalwert für den Teiler fest. An dieser Stelle

fliesst daher Wissen über den Amplitudengang ein. Donglai und Paliwal schlagen ein Produktspektrum vor [DP04]:

$$Q(k) = X_R(k) \cdot Y_R(k) + X_I(k) \cdot Y_I(k),\ k = 0, ..., N-1 \tag{3.9}$$

3.4. Zusammenfassung und Einordnung

In diesem Kapitel wurden Verfahren vorgestellt, um

- die Grundfrequenz zu schätzen
- Verschlussmomente zu finden
- robuste Merkmalsvektoren zu extrahieren
- den Phasengang zur Merkmalsextraktion zu verwenden.

Das Ziel der Arbeit ist es, ein robustes Frontend zu entwickeln. Dazu werden verschiedene der vorgestellten Techniken übernommen, angepasst und kombiniert. Im Wesentlichen werden dabei die Eigenschaften stimmhafter Sprache genutzt, um den Sprecher mithilfe von harmonischer Filterung vom Hintergrund zu trennen. Die Grundfrequenz und die Verschlussmomente werden dabei mit begrenzter Vorausschau geschätzt.

Um das zu erreichen, wird die Methode von Roa, Bennewitz und Behnke zur robusten Grundfrequenzerkennung verwendet und in ein iteratives Verfahren mit begrenzter Vorausschau integriert. Ähnlich wie beim Super Resolution Pitch Determination Algorithmus wird der Suchraum, in dem die Grundfrequenz vermutet wird, eingeschränkt, sobald stimmhafte Sprache erkannt wird. Der Suchraum wird nur sehr langsam eingeschränkt, um die Grundfrequenz auch bei Rauschen nicht zu verfehlen. Um Laufzeit einzusparen, wird die harmonische Oberfläche nicht komplett berechnet, sondern sie wird so berechnet, dass mit wenig Aufwand eine Schätzung werden abgegeben kann, die in Richtung der Grundfrequenz weist. In späteren Iterationen kann die Oberfläche dann für einen kleineren Teil der harmonischen Oberfläche genauer berechnet werden.

Die Methode von Yegnanarayana zur Verschlussmomenterkennung mithilfe der Gruppenlaufzeit wird abgeändert, so dass die Verschlussmomente ebenfalls mit begrenzter Vorausschau bestimmt werden können. Es wird versucht, die Methode so abzuändern, dass sie robuster wird. Ebenso soll nicht für alle Abtastpunkte die Gruppenlaufzeit berechnet werden. Dazu wird ein Indikator benutzt, der auf Verschlussmomente verweist, so dass die Analysefenster in dem iterativen Verfahren immer genauer um die Verschlussmomente zentriert werden. Dazu werden die Eigenschaften der Phase für periodische Signale in den harmonischen Körben bei grundfrequenzskalierter Verarbeitung ausgenutzt.

Zur Merkmalsextraktion wird das Verfahren von Behnke's Harmonic Frontend übernommen.

Kapitel 4.

Entwurf eines robusten Frontends

Das Ziel in diesem Kapitel ist es, ein robustes Frontend zu entwickeln, das eine Folge von Merkmalsvektoren aus verrauschten Sprachsignalen ableitet.

Der wesentliche Aspekt des Algorithmus ist es, dass die Eigenschaften des Quellsignals genutzt werden, um den Sprecher vom Hintergrund zu trennen. Dazu wird ein Algorithmus entwickelt, der mithilfe von harmonischer Filterung Grundfrequenz und Verschlussmomente des Quellsignals in einem verrauschten Sprachsignal schätzt. Dabei darf zu jedem Zeitpunkt nur begrenzt weit in die Zukunft geschaut werden. Um das zu erreichen, werden Verfahren für die Verschlussmoment- und Grundfrequenzverfolgung abgewandelt, die dynamische Programmierung verwenden und das Sprachsignal vollständig kennen [Beh, NKGB07]. Für die Merkmalsextraktion werden dann die Eigenschaften des Quellsignals ausgenutzt, um robuste Merkmalsvektoren zu berechnen. Dabei wird keine begrenzte Vorausschau gefordert. Es wird die Merkmalsextraktion des Harmonic Frontends von Behnke verwendet (siehe Abschnitt 3.2) [Beh].

In Abschnitt 4.1 werden die Beschränkungen, denen der Algorithmus unterworfen ist, genauer beschrieben. Im Anschluss wird in Abschnitt 4.2 ein Gerüst für einen iterativen Algorithmus vorgestellt, welches diese Beschränkungen beachtet und eine robuste Schätzung der Verschlussmomente und der Grundfrequenz ermöglichen soll. Daraufhin werden im Abschnitt 4.3 einige nachfolgend benötigte Definitionen gegeben. In Abschnitt 4.4 wird das grundsätzliche Vorgehen bei der Verarbeitung des Phasengangs dargestellt. Anschließend werden in den Abschnitten 4.5 und 4.6 Techniken zur Schätzung der Grundfrequenz und der Verschlussmomente in das Gerüst zur iterativen Verarbeitung integriert und angepasst. Abschließend wird in Abschnitt 4.7 ein Experiment durchgeführt, dass die Ähnlichkeit von Amplitudengang und Phasengang bei stimmhafter Sprache zeigen soll. Ebenso wird dort erläutert, wie die Merkmalsextraktion mithilfe des Harmonic Frontends von Behnke durchgeführt wird.

4.1. Anforderungen

Der Algorithmus arbeitet auf einem zeitdiskreten Signal $x \in l^2(\mathbb{Z})$. x wurde mit einer Abtastrate $f_s > 0$ von einem zeitkontinuierlichen Signal $f \in L^2(\mathbb{R})$ periodisch abgetastet. Seine Arbeit nimmt der Algorithmus bei einem Abtastwert $x(t_i)$, $t_i \in \mathbb{Z}$ zum Zeitpunkt $t_i \cdot \frac{1}{f_s}$ auf.

Für einen Abtastpunkt $t_{i+1} \in \mathbb{Z}$, $t_i + V_{it} \geq t_{i+1} > t_i$, $V_{it} \in \mathbb{N}^+$ muss der Algo-

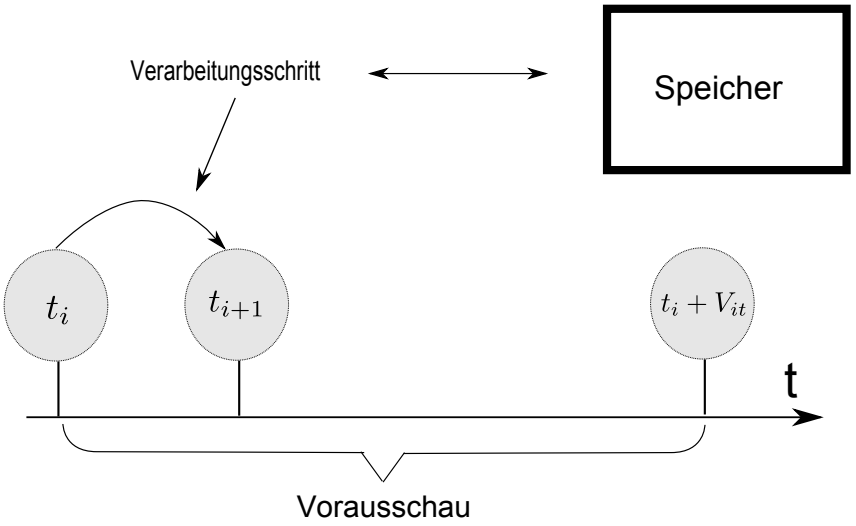


Abbildung 4.1.: Anforderungen an den Algorithmus. Im Verarbeitungsschritt $i+1$ werden die Schätzungen über das Quellsignal für alle Abtastpunkte $t_i < t' \leq t_{i+1}$ festgelegt. Zwischen zwei Verarbeitungsschritten liegt ein gewisser Abstand: $t_i < t_{i+1}$. Für einen Verarbeitungsschritt ist beschränkt viel Speicher, beschränkt viel Zeit und eine begrenzte Vorausschau vorhanden.

rithmus für die Abtastpunkte $t' \in \mathbb{Z}$, $t_i < t' \leq t_{i+1}$ die Schätzung des Quellsignals abgeschlossen haben: Erst wird entschieden, ob t' stimmhaft ist, oder nicht. Wenn t' stimmhaft ist, wird t' eine Grundfrequenz zugeordnet und es wird entschieden, ob t' ein Verschlussmoment ist, oder nicht.

Sind diese Zuordnungen geschehen, rückt der Algorithmus zum Abtastpunkt t_{i+1} vor. Der *Verarbeitungsschritt* $i+1$ ist dann beendet. Dabei durfte der Algorithmus maximal V_{it} Abtastpunkte nach vorne rücken. Das soll dem Algorithmus eine variable Schrittweite für die glottissynchrone Verarbeitung ermöglichen.

Ein Verarbeitungsschritt muss gewisse Bedingungen erfüllen. Der Algorithmus hat eine begrenzte *Vorausschau* (siehe Abbildung 4.1). Um die Entscheidungen im Verarbeitungsschritt $i+1$ zu treffen, dürfen lediglich die Abtastwerte für die Abtastpunkte $t_i, t_i + 1, ..., t_i + V_{it}$ gelesen werden. Es steht Speicher zur Verfügung, auf den lesend und schreibend zugegriffen werden darf. Dieser Speicher hat eine begrenzte Größe. Es können zum Beispiel begrenzt viele Abtastwerte aus der Vergangenheit gespeichert werden. In einem Verarbeitungsschritt dürfen nur konstant viele Operationen durchgeführt werden.

Durch diese Anforderungen an den Algorithmus soll gewährleistet werden, dass der Algorithmus auf entsprechend schneller Hardware und effizienter Implementierung echtzeitfähig ist.

4.2. Struktur

Der Algorithmus soll die beschriebenen Anforderungen erfüllen. Gleichzeitig soll iterativ vorgegangen werden: Die Vorausschau wird verwendet, um Zwischenergebnisse zu verbessern. Da die Vorausschau und die verfügbaren Operationen begrenzt sind, muss die Iteration nach einer konstanten Anzahl von Schritten abbrechen.

4.2.1. Motivation

Es wird versucht, verschiedene Schätzungen iterativ zu verbessern:

- Stimmhaft/nicht stimmhaft?
- Grundfrequenz
- Rauschen
- Verschlussmomente

Das soll helfen, die Laufzeit zu verkürzen: Um grobe Schätzungen zu berechnen, muss weniger gerechnet werden. Ebenso ermöglicht es die iterative Verarbeitung, aufbauend auf Zwischenschätzungen die Verarbeitung anzupassen. Bei der iterativen Verarbeitung mit begrenzter Vorausschau müssen daher nicht sofort Entscheidungen festgesetzt werden, wenn ein Signalausschnitt zum ersten Mal gesehen wird, sondern es können vorsichtige Schätzungen abgegeben werden, die nur leicht in eine Richtung weisen. Nachfolgenden Iterationsschritten wird dadurch im besten Fall eine Information übergeben, auf deren Grundlage, eine bessere Entscheidung getroffen werden kann.

4.2.2. Iterative Verarbeitung

Die Grundstruktur, welche die iterative Verarbeitung ermöglichen soll, ist in Abbildung 4.2 dargestellt. Im *Trichterschritt* $i+1$ werden die $B > 1 \in \mathbb{N}$ *Iterationsketten* $i+1$, ..., $i+1+B-1$ erweitert. (i,b), $b \in \{1,...,B\}$ ist der b. *Iterationspunkt* der Iterationskette i. Den Iterationspunkten können Schätzungen über Größen zugewiesen werden. Beispielsweise beträgt die Grundfrequenzschätzung für die Iterationskette i beim *Iterationsanfang* $(i,1)$ 250 Hz. Diese Schätzung wird dann entlang der Iterationskette verbessert, bis beim *Iterationsende* (i,B) als *Endschätzung* eine Grundfrequenz von 254 Hz festgelegt wird. Die Schätzungen des Iterationspunktes (i,b) gelten für den Abtastpunkt $t_{i,b} \in \mathbb{Z}$. Der Abtastpunkt $t_{i,B}$ des Iterationsendes wird auch als t_i bezeichnet. Damit die Iterationsketten einander nicht überholen können oder auseinander driften, werden einige Bedingungen gestellt. Für $i \in \mathbb{Z}$ und $b \in \{1,...,B\}$ gilt

- $t_{i+1,b} > t_{i,b}$
- $t_{i+1,1} \leq t_{i,1} + T_{max_1}$, $T_{max_1} \in \mathbb{N}$
- $|t_{i,b} - t_{i,1}| \leq T_{max_b}$, $T_{max_b} \in \mathbb{N}$

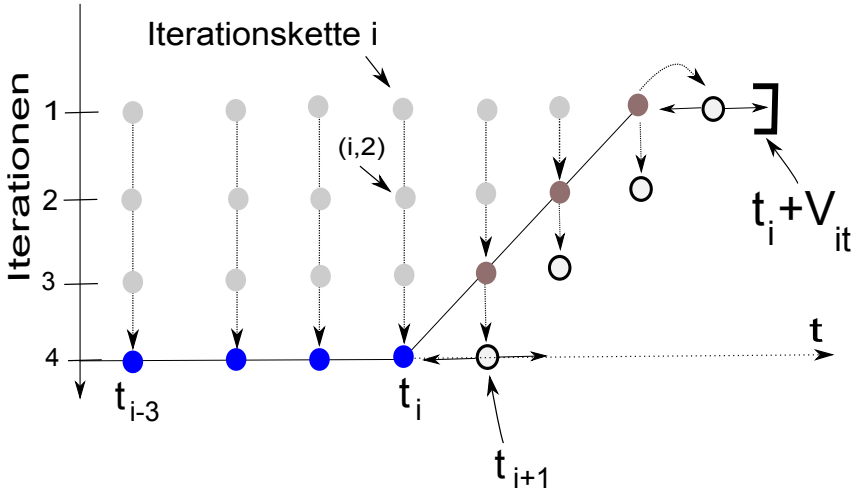


Abbildung 4.2.: Struktur der iterativen Verarbeitung. Eine *Iterationskette* i schätzt verschiedene Werte (wie beispielsweise die Grundfrequenz) für Abtastpunkte $t_{i,b}$. Die *Iterationskette* i besteht aus mehreren *Iterationspunkten*. Vom ersten bis zum letzten Iterationspunkt werden die Schätzungen einer Iterationskette verfeinert. Im Bild wurde gerade die Iterationskette i *abgeschlossen*. Im anstehenden *Trichterschritt* $i+1$ sind die Iterationsketten $i+1$ bis $i+1+B-1$ *aktiv* und werden jeweils um einen Iterationspunkt erweitert. Ist die Zahl der Iterationen B (hier 4), so wird die Iterationskette $i+1$ abgeschlossen und die Iterationskette $i+1+B-1$ wird *angelegt*. Die dunkelgrauen Punkte bilden gemeinsam mit den blauen Punkten eine Nachbarschaft, welche genutzt werden kann, um zeitlich benachbarte Schätzungen für die Erweiterung einer Iterationskette zu benutzen. In diesem Beispiel ist $t_{i,b} = t_{i,1}$. Bei der glottissynchronen Verarbeitung können die Abtastpunkte $t_{i,b}$ einer Iterationskette dagegen variieren, damit sich die Iterationsketten mit den Verschlussmomenten synchronisieren können (siehe Abschnitt 4.6).

Der Trichterschritt $i+1 \in \mathbb{Z}$ benötigt dann eine maximale Vorausschau von V_{it} Abtastpunkten:

$$\begin{aligned} t_{i+B,B} - t_{i,B} &\leq t_{i+B,1} + T_{max_b} - t_{i,B} \\ &\leq t_{i,1} + B \cdot T_{max_1} + T_{max_b} - t_{i,B} \\ &\leq t_{i,B} + B \cdot T_{max_1} + 2 \cdot T_{max_b} - t_{i,B} \\ &\leq B \cdot T_{max_1} + 2 \cdot T_{max_b} \\ &= V_{it} - V_p,\, V_p \in \mathbb{N}^+ \end{aligned} \tag{4.1}$$

V_p entspricht der Vorausschau, die benötigt wird, um den Iterationspunkt $(i+B, B)$ zu berechnen.

Ein Trichterschritt ist ein Verarbeitungsschritt: Im Trichterschritt $i+1$ werden die Endschätzungen der Trichterkette $i+1$ mit einer Vorausschau von V_{it} festgelegt. Dazu dürfen beschränkt viele Operationen und beschränkter Speicher verwendet werden. Die Abtastpunkte $t_i < t' \leq t_{i+1}$ können durch Stetigkeitsannahmen berücksichtigt werden.

Die Bezeichnung *Iterationsebene* wird im Folgenden verwendet, um Vorgänge zuzuordnen, die einen festen Iterationsschritt $b \in \{1, ..., B\}$, aber verschiedene Iterationsketten betreffen.

4.3. Definitionen

An dieser Stelle werden Definitionen zusammengefasst, die in der weiteren Arbeit verwendet werden.

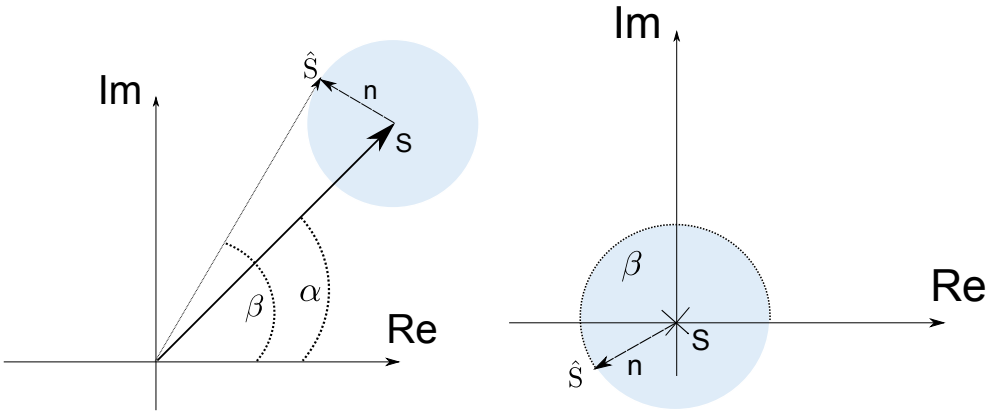
Im Folgenden wird von einem *Sprachsignal* $x \in l^2(\mathbb{Z})$ ausgegangen, dass von einem bandbegrenzten Signal $f \in L^2(\mathbb{R})$ mit einer Abtastrate von $f_s = 8000$ periodisch abgetastet wurde. f_s ist dabei größer als die Nyquist-Rate. Ein *Frame* oder *Ausschnitt* ist ein Vektor $x_{l,m} \in \mathbb{R}^l$, der aus dem Sprachsignal x mit einem Fenster $w \in l^2(\mathbb{Z})$ ausgeschnitten wird, um den Abtastpunkt $m \in \mathbb{Z}$ zentriert ist und $l \in L := \{L_{min} = 100, 101, ..., L_{max} = 400\}$ Abtastpunkte enthält.

$$x_{l,m}(n) = w(n) \cdot x(m + n - \lfloor \frac{l}{2} \rfloor),\, n \in \{1, 2, ..., l\},\, w \in l^2(\mathbb{Z}). \tag{4.2}$$

Die *Fensterlänge* l heißt zum Abtastpunkt m *grundfrequenz-skaliert*, falls der Frame $x_{l,m}$ genau vier Phonationszyklen gleicher Länge enthält. Einer Fensterlänge $l \in L$ wird eine Grundfrequenz $f_0 \in \mathbb{Q}$ zugeordnet.

$$f_0(l) = ((l \cdot \frac{1}{f_s})/4)^{-1}. \tag{4.3}$$

Gleichung 4.3 ergibt sich, weil die Fensterlängen l als grundfrequenz-skaliert angenommen werden, so dass in ein Fenster jeweils vier Phonationszyklen hineinpassen. Die Menge der Fensterlängen wird auf L beschränkt, weil das Grundfrequenzen zwischen 80 Hz und 323 Hz entspricht und übliche Grundfrequenzen abdeckt (siehe Abschnitt 2.2.1.2). Die gesamte Anzahl der betrachteten Fensterlängen beträgt $L_g = L_{max} - L_{min} + 1 = 301$.



(a) Hoher Absolutwert (b) Absolutwert von Null

Abbildung 4.3.: Links sieht man einen Vektor S mit hohem Absolutwert. Auf diesen wird ein Fehler n mit konstantem Absolutwert und beliebigem Winkel addiert. Der maximale Fehler des Winkels $\angle\hat{S}$ beträgt $\beta - \alpha$. Rechts hat S einen Absolutwert von Null. Der resultierende Vektor $\hat{S}$ übernimmt den Winkel des Fehlers n.

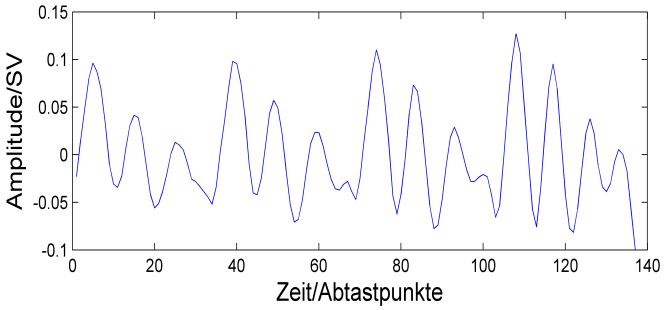
Ein Frame $x_{l,m}$ heißt *glottissynchron* mit *Lage* $p \in [-\pi; \pi]$, falls l für m grundfrequenzskaliert ist und der Abtastpunkt $(m - \frac{p}{\pi}\frac{l}{8})\frac{1}{fs} \in \mathbb{R}$ ein Verschlussmoment ist.

$X_{l,m} \in \mathbb{C}^l$ ist die DFT eines Frames $x_{l,m}$. Der j-te *harmonische Koeffizient* des j-ten *harmonischen Korbes* ist eines Frames $x_{l,m}$ ist (vergleiche Abschnitt 3.1.1.2):

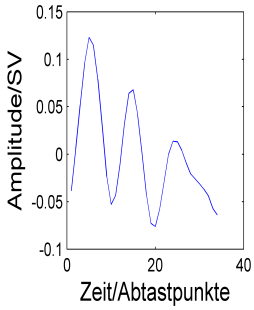
$$X(4 \cdot j), j \in \{1, 2, ..., H_l := \lfloor \frac{1}{4} \lceil l/2 \rceil \rfloor - 1\}. \tag{4.4}$$

4.4. Phase

Einer der Gedanken der Arbeit ist es, den Phasengang zu verwenden, und nicht zu verwerfen, wie das häufig getan wird. In Abbildung 4.3 wird gezeigt, wie sich der Winkel einer komplexen Zahl $S \in \mathbb{C}$ verhält, wenn sie von einem Fehler $n \in \mathbb{C}$ additiv überlagert wird. Schätzt man das Spektrum eines grundfrequenz-skalierten Frames mit einer DFT und verwendet eine rechteckige Fensterfunktion, so erhält man in nicht-harmonischen Körben eine komplexe Zahl, die dem Hintergrund zugeordnet ist (siehe Abschnitt 3.1.1.2). Der Absolutbetrag dieser Zahl ist bei einem klaren Sprachsignal im Idealfall Null, der Winkel/die Phase ist jedoch beliebig. Auch bei einem moderat lärmbehafteten Signal bleibt der Betrag der nicht-harmonischen Koeffizienten deutlich niedriger als bei den harmonischen Koeffizienten. Der Winkel der nicht-harmonischen Koeffizienten korelliert nicht mit dem Sprachsignal und kann als zufällig verteilt angenommen werden. Soll beispielsweise die Differenz der Winkel eines harmonischen und eines nicht-harmonischen Koeffizienten berechnet werden, so enthält das Ergebnis wenig Einfluss des Sprechers. Da der Winkel der nicht-harmonischen komplexen Zahl zufällig ist, ist auch das Ergebnis der Berechnung zufällig.



(a) Vier Phonationszyklen



(b) Harmonisches Spektrum im Zeitbereich

Abbildung 4.4.: Harmonisches Spektrum. Links sind vier Phonationszyklen stimmhafter Sprache aufgetragen. Rechts sieht man das in den Zeitbereich transformierte harmonische Spektrum, welches einen Phonationszyklus beschreibt, beginnend mit dem Verschlussmoment.

Möchte man die Phase nur dort verarbeiten, wo der Einfluss des Sprechers überwiegt, kann man sich auf die harmonischen Koeffizienten konzentrieren. Gemeinsam mit der DC-Komponente $X_{l,m}(0)$ können die harmonischen Koeffizienten eines grundfrequenz-skalierten Frames $x_{l,m}$ als die untere Hälfte des Spektrums eines einzigen Phonationszyklusses aufgefasst werden (siehe Abbildung 4.4). Bei diesem *harmonischen Spektrum* überwiegt im Idealfall der Einfluss des Sprechers für alle Koeffizienten. Wählt man die Fensterlänge von vorneherein so, dass nur ein Phonationszyklus hineinpasst, ergeben sich einige Nachteile. Es gibt keine nicht-harmonischen Körbe, in denen Rauschen geschätzt werden kann. Das Rauschen fällt vollständig auf die harmonischen Körbe. Die Grundfrequenz kann weniger genau angepasst werden, weil ein um einen Abtastpunkt längeres Fenster einer Verlängerung des vier Zyklen fassenden Fensters um vier Abtastpunkte entspricht. Andererseits können mit einem längeren Fenster schlechter zeitliche Veränderungen der Grundfrequenz erfasst werden. In der vorliegenden Arbeit wird ausschließlich die Phase der harmonischen Koeffizienten verwendet. Ein einzelner Phonationszyklus stimmhafter Sprache ist näherungsweise ein Minimalphasensignal, wenn der Ausschnitt beim Verschlussmoment beginnt (siehe Abschnitt 2.1.4) [YS95].

4.5. Grundfrequenz

Nachdem in den letzten Abschnitten die Struktur des Algorithmus und das grundsätzliche Vorgehen bei der Verarbeitung der Phaseninformation beschrieben wurde, werden in den folgenden Abschnitten die einzelnen Bestandteile des Algorithmus genauer erläutert.

Die Grundfrequenzverfolgung basiert auf dem Verfahren von Behnke und dem Verfahren von Roa, Bennewitz und Behnke (siehe Abschnitt 3.1.1.2) [Beh03, RBB07]. Um die iterative Verarbeitung mit begrenzter Vorausschau zu ermöglichen, werden Anpassungen vorgenommen. In Abbildung 4.5 ist ein Überblick über die iterative

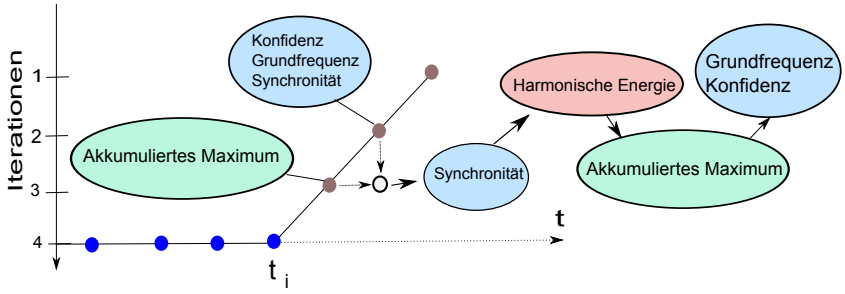


Abbildung 4.5.: Überblick über die iterative Grundfrequenzschätzung. Für die Iterationskette $i+2$ wird der dritte Iterationspunkt berechnet.

Verarbeitung gegeben. Nachdem ein Iterationspunkt in einem Trichterschritt bearbeitet wurde, liegen für ihn Werte vor:

- Grundfrequenzschätzung
- Synchronitätsgrad
- Konfidenz
- Akkumuliertes Maximum

Der *Synchronitätsgrad* korrespondiert mit der Unsicherheit der Grundfrequenzschätzung. Je sicherer die Grundfrequenzschätzung ist, desto höher der Synchronitätsgrad und es werden nur noch Grundfrequenzen mit einer kleinen Abweichung von der Grundfrequenzschätzung berücksichtigt, ähnlich wie das bei dem Super Resolution Pitch Determination Algorithmus der Fall ist (siehe Abschnitt 3.1.1.3). Der Synchronitätsgrad wird anhand der *Konfidenz* angepasst, die ein lokales Maß über die Sicherheit einer Schätzung darstellt: Konfidente Messungen führen dazu, dass der Synchronitätsgrad erhöht wird.

Auf jeder Iterationsebene wird ein *akkumuliertes Maximum* verwaltet, das für jede Grundfrequenz den Nutzen eines gemäß Nutzenfunktion optimalen Pfades enthält, der zu dieser Grundfrequenz führt.

In Abbildung 4.5 werden beispielsweise die Werte für den Iterationspunkt $(i+2,3)$ berechnet. Anhand des Synchronitätsgrades und der Konfidenz des Iterationspunktes $(i+2,2)$ wird der Synchronitätsgrad des Iterationspunktes $(i+2,3)$ bestimmt. Der neue Synchronitätsgrad gemeinsam mit der Grundfrequenz des Punktes $(i+2,2)$ definieren ein Intervall, in dem die Grundfrequenz vermutet wird. Für einige Grundfrequenzen aus diesem Intervall wird dann eine *harmonische Energie* berechnet, die im Idealfall bei der richtigen Grundfrequenz maximal ist. Daraufhin kann das akkumulierte Maximum der Ebene 2 aktualisiert werden. Die Grundfrequenz, deren Pfad den größten Nutzen hat, wird als Grundfrequenzschätzung des Iterationspunktes $(i+2,3)$ gewählt. Abschließend wird die Konfidenz der Messung bestimmt, indem die harmonische Energie bei der Grundfrequenzschätzung mit einem *Energielevel* verglichen wird.

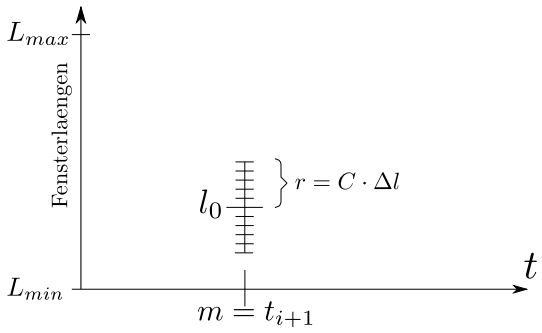


Abbildung 4.6.: Die Spalte E_m wird abgetastet. Für die Iterationskette $i+1$ wird die harmonische Energie für verschiedene Fensterlängen berechnet. Die aktuelle Grundfrequenzschätzung beträgt l_0, der Synchronitätsradius beträgt beispielsweise 100. Daraus ergibt sich eine Abtastdistanz $\Delta l = \frac{100}{C} = \frac{100}{5} = 20$. Die harmonische Oberfläche $E(100, 20)$, wird zum Abtastpunkt m für $l = l_0 + k \cdot \Delta l$ und $k \in \{-C, ..., C\}$ berechnet.

In den folgenden Abschnitten wird zunächst erläutert, wie die harmonische Energie für unterschiedliche Synchronitätsgrade berechnet wird. Im Anschluß werden Details der iterativen Verarbeitung erläutert.

4.5.1. Vorüberlegungen

Ein Wert $E_{l,m} \in \mathbb{R}_{\geq 0}$ ordnet einem Frame $x_{l,m}$ eine harmonische Energie zu (vergleiche Abschnitt 3.1.1.2). In der *Spalte* $E_m := (E_{L_{min},m}, ..., E_{L_{max},m}) \in \mathbb{R}^{|L|}$ ist im Idealfall der Wert $E_{l,m}$ für eine grundfrequenz-skalierte Fensterlänge l maximal. Es ist aufwändig, die *harmonische Oberfläche* $E \in \mathbb{R}^{|L| \times \mathbb{Z}}$ des Sprachsignals x zu berechnen, weil für jeden Wert $E_{l,m}$ die DFT $X_{l,m}$ benötigt wird (siehe Abschnitt 4.5.2). In dem iterativen Verfahren, das die Grundfrequenz schätzt, wird daher für jeden Iterationspunkt die harmonische Energie für einen festen Abtastpunkt m und eine kleine Menge von Fensterlängen der Spalte E_m geschätzt.

Dazu wird zum einen ein Intervall von Fensterlängen festgelegt, in dem die Grundfrequenz vermutet wird. Zum Anderen wird die harmonische Energie nur für wenige Fensterlängen berechnet, die dieses Intervall abdecken. Das Intervall wird durch eine *zentrale Fensterlänge* $l_0 \in L$ und einen *Synchronitätsradius* $r \in [0; \frac{L_g - 1}{2} = 150 =: r_{max}]$ festgelegt. Das Intervall ergibt sich als $I = [l_0 - r; l_0 + r]$.

Der Synchronitätsradius $r \in [0; \frac{L_g - 1}{2} = 150 =: r_{max}]$ wird abhängig von einem *Synchronitätsgrad* $s \in [0; 1]$ bestimmt.

$$r = (1 - s) r_{max}. \tag{4.5}$$

Je größer der Synchronitätsgrad ist, desto kleiner wird der Synchronitätsradius und die Grundfrequenz wird nur noch in einem kleinen Intervall von Fensterlängen vermutet. Um dieses Intervall abzudecken, stehen $2 \cdot C + 1$, $C \in \{5, ..., 14\}$ Abtastpunkte zur Verfügung. Die *Abtastdistanz* $\Delta l \in \{1, 2, ..., \Delta l_{max} = \lceil \frac{r_{max}}{2} \rceil\}$ ergibt sich aus

dem Synchronitätsradius r und der Konstanten C (siehe Abbildung 4.6).

$$\Delta l = \max\{\lceil \frac{r}{C} \rceil, 1\}. \tag{4.6}$$

Für eine zentrale Fensterlänge $l_0 \in L$, eine Konstante $C \in \{5, ..., 14\}$, und eine Abtastdistanz $\Delta l \in \{1, 2, ..., \Delta l_{max}\}$ wird eine Menge von Fensterlängen definiert.

$$F(l_0, C, \Delta l) := \{l_0 + \Delta l \cdot k \in L | k \in \{-C, ..., 0, ...C\} \subseteq \mathbb{Z}\}. \tag{4.7}$$

Hat man eine zentrale Fensterlänge l_0, eine Konstante C und einen Synchronitätsradius r gegeben, kann eine Menge von Fensterlängen berechnet werden. Wird beispielsweise $C = 5$, $l_0 = 250$ und $r = 150$ gewählt, ergibt sich Δl als 30. Dann ergibt sich $F(250, 5, 30)$ als Menge von Fensterlängen. Dann ist $250 + 30 \cdot 5 = 400$ die größte dieser Fensterlängen und $250 - 30 \cdot 5 = 100$ ist die kleinste Fensterlänge. Bei der Berechnung der Spalte E_m für diese Fensterlängen wird dann der übliche Grundfrequenzbereich abgedeckt und es wird jede 30-te Fensterlänge berechnet.

Um Abtastfehler und Oktavenfehler zu minimieren, wird eine Funktion gesucht, welche die harmonische Oberfläche E abhängig vom Synchronitätsradius r und der Abtastdistanz Δl berechnet. Die Funktion

$$E_{l,m} : S \times D \to \mathbb{R} \ , \ S = [0; r_{max}], \ D = \{1, 2, ..., \Delta l_{max}\} \tag{4.8}$$

wird gesucht. Diese Funktion soll berücksichtigen, dass für einen großen Synchronitätsradius ein großer Grundfrequenzbereich abgedeckt wird und daher Oktavenfehler auftreten können. Außerdem soll beachtet werden, dass für große Werte von Δl mit wenigen Abtastpunkten ein großer Grundfrequenzbereich untersucht werden muss, und daher Abtastprobleme auftreten.

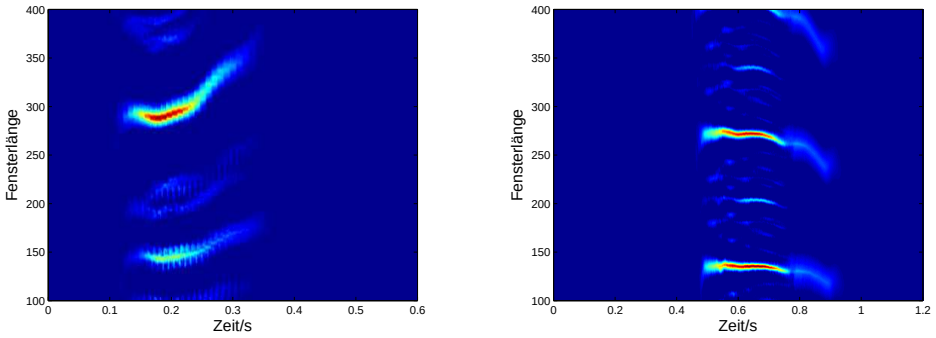
Für eine Konstante C hängt die Berechnung der harmonischen Oberfläche $E(r, \Delta l)$ nur von dem Synchronitätsradius r ab. Andererseits soll der Paramater C in Testläufen variiert werden, um die Auswirkungen für verschieden große Mengen von Abtastpunkten zu testen. Zur Vereinfachung werden Oktavenfilter verwendet, die im Idealfall unabhängig vom Synchronitätsgrad eingeschaltet bleiben können. Dann kann unter der Voraussetzung von eingeschalteten Oktavenfiltern untersucht werden, wie die harmonische Oberfläche bei variierender Abtastdistanz berechnet werden kann. Zusammenfassend wird die Funktion

$$E^O_{l,m} : D \to \mathbb{R}, \ D = \{1, 2, ..., \Delta l_{max}\} \tag{4.9}$$

gesucht.

4.5.2. Oberflächendesign

Zunächst wird erläutert, wie die harmonische Oberfläche $E(1)$ berechnet wird. Die Berechnung geschieht analog zu dem Vorgehen von Behnke [Beh03]. Im Anschluss werden Oktavenfilter zur Berechnung von $E^O(1)$ vorgestellt. Diese basieren im Wesentlichen auf der Arbeit von Roa, Bennewitz und Behnke [RBB07]. Schließlich wird ein Verfahren beschrieben, wie $E^O(\Delta l_{max})$ berechnet wird. $E^O(\Delta l)$ wird dann durch Interpolation zwischen den Extremfällen $E^O(\Delta l_{max})$ und $E^O(1)$ berücksichtigt.



(a) „Three “ (b) „One “

Abbildung 4.7.: Die harmonische Oberfläche $E(1)$ wurde für zwei Sprachsignale berechnet. Links liegt die Grundfrequenz bei einer Fensterlänge von ca. 300 Abtastpunkten. Rechts liegt sie bei ca. 140 Abtastpunkten. Die roten Bereiche zeigen Stellen an, denen viel harmonische Energie zugeordnet wird. Von gelben bis zu dunkelblauen Bereichen wird immer weniger harmonische Energie gefunden.

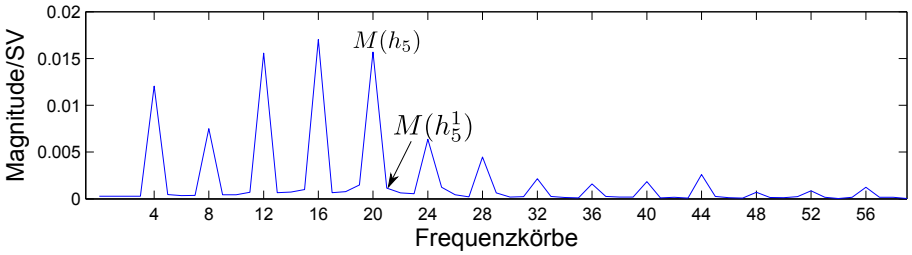


Abbildung 4.8.: Grundfrequenz-skaliertes Spektrum. Ist die Fensterlänge des Frames $x_{l,m}$ grundfrequenz-skaliert, so fällt die harmonische Energie auf die harmonischen Körbe h_j. In den benachbarten Körben h_j^l können stimmlose Sprache und Rauschen geschätzt werden.

4.5.2.1. Oberfläche $E(1)$

Für die Berechnung von $E(1)$ wird zum Ausschneiden der Frames $x_{l,m}$ ein Rechteckfenster gewählt. Für die DFT $X_{l,m}$ wird die vorhandene harmonische Energie geschätzt:

Von der Magnitude $M_{l,m}(h_j) := |X_{l,m}(4 \cdot j)|$ des j-ten harmonischen Korbes h_j, $j \in \{1, ..., H_l := \lfloor \frac{1}{4} \lceil l/2 \rceil \rfloor - 1\}$ wird zunächst das lokal geschätzte Rauschen in einer Form des Spectral Subtraction gewichtet abgezogen [Beh]:

$$\hat{M}_{l,m}(h_j) = M_{l,m}(h_j) - R_{l,m}(h_j). \tag{4.10}$$

Mit der Definition $M_{l,m}(h_j^k) := |X_{l,m}(4 \cdot j + k)|$, $k \in \{-3, ..., 0, ..., 3\}$ kann die Nachbarschaft der harmonischen Körbe angesprochen werden. Da ein rechteckiges Fenster verwendet wird, kann das Rauschen für einen harmonischen Korb h_j lokal in den be-

nachbarten nicht-harmonischen Körben geschätzt werden [Beh]:

$$\begin{aligned} R_{l,m}(h_j) = \frac{1}{44} \cdot (&15(M_{l,m}(h_j^1) + M_{l,m}(h_j^{-1})) \\ &+6(M_{l,m}(h_j^2) + M_{l,m}(h_j^{-2})) \\ &+1(M_{l,m}(h_j^3) + M_{l,m}(h_j^{-3}))). \end{aligned} \tag{4.11}$$

Die harmonische Energie für einen Frame $x_{l,m}$ ist

$$E_{l,m}(1) = max\{0, \frac{1}{l} \sum_{i \in H_l} \hat{M}_{l,m}(h_j)\}. \tag{4.12}$$

Die Normalisierung mit $\frac{1}{l}$ wird vorgenommen, um systematische Energieunterschiede auszugleichen, die durch verschiedene Fensterlängen bedingt sind (ähnlich dem Super Resolution Pitch Determination Algorithmus, siehe Abschnitt 3.1.1.1). In Abbildung 4.7 ist die Oberfläche $E(1)$ für zwei Sprachsignale dargestellt. Harmonische Energie tritt nicht nur bei der Fensterlänge auf, die der Grundfrequenz entspricht. Auch benachbarte Fensterlängen enthalten Energie. Ausgeprägte Maxima finden sich zumeist bei den benachbarten Oktaven, weshalb ein Oktavenfilter eingesetzt wird.

4.5.2.2. Oberfläche $E^O(1)$

Der *Vielfachfilter* versucht, Energie bei Fensterlängen zu entfernen, die acht anstatt vier Phonationszyklen enthalten und deshalb im Idealfall nur in jedem achten Frequenzkorb Energie enthalten [RBB07]:

$$V_{l,m}(h_j) = \begin{cases} min\{\hat{M}_{l,m}(h_j), \hat{M}_{l,m}(h_{j-1}) + \hat{M}_{l,m}(h_{j+1})\} \text{ für } j = 2, ..., H_l - 1 \\ \hat{M}_{l,m}(h_j) \text{ sonst.} \end{cases} \tag{4.13}$$

Der *Teilerfilter* soll zusätzlich Energie bei Fensterlängen entfernen, die zwei Phonationszyklen enthalten. Dann findet sich in jedem zweiten Frequenzkorb harmonische Energie. Der Teilerfilter wird abweichend von einem vorgeschlagenen Teilerfilter implementiert:

$$T_{l,m}(h_j) = \begin{cases} 0 \text{ für } V_{l,m}(h_j) < 0.5(M_{l,m}(h_j^2) + M_{l,m}(h_j^{-2})) \text{ und } j = 2, ..., H_l - 1 \\ 0 \text{ für } \hat{M}_{l,m}(h_j) < 0.5(M_{l,m}(h_j^2) + M_{l,m}(h_j^{-2})) \text{ und } j \in \{1, H_l\} \\ V_{l,m}(h_j) \text{ sonst.} \end{cases} \tag{4.14}$$

Im Gegensatz dazu schlagen Roa, Bennewitz und Behnke vor, von $E_{l,m}$ einen Teil der Energie abzuziehen, die bei Vielfachen der Fensterlänge l gefunden wird [RBB07]. In dieser Arbeit ist das nicht ohne weiteres möglich, weil nicht die gesamte harmonische Oberfläche E berechnet wird.

Der Vielfachfilter kann *ausgeschaltet* werden, indem der Teilerfilter die Eingabe des Vielfachfilters erhält.

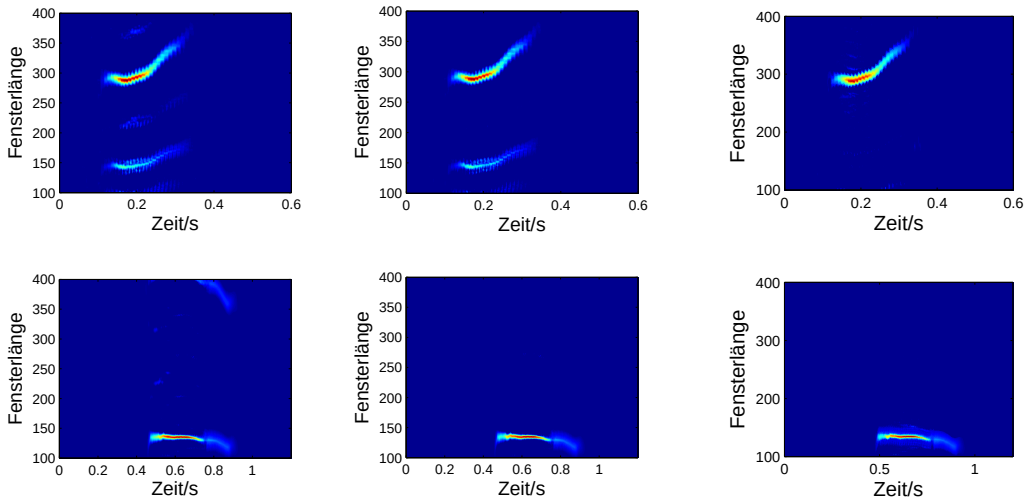


Abbildung 4.9.: Oktavenfilter. Oben kann die Wirkung der Oktavenfilter für einen Sprecher mit tiefer Grundfrequenz beobachtet werden. Unten finden sich gefilterte harmonische Oberflächen für eine Sprecherin mit hoher Grundfrequenz. Die ungefilterten Oberflächen können in Abbildung 4.7 betrachtet werden. Links sieht man einen Vielfachfilter, welcher nur die harmonischen Körbe $h_2, h_4, ...$ filtert. Das reicht aus, um das Maximum bei der halben Grundfrequenz zu filtern (links unten). In der Mitte wurden, wie im Text beschrieben, (fast) alle harmonischen Körbe gefiltert. Das entfernt auch Maxima, die anderen Verhältnissen entsprechen. Rechts sieht man, wie zusätzlich der Teilerfilter eingeschaltet wurde. Bei diesem muss auch der erste harmonische Korb gefiltert werden: Wenn das Analysefenster nur zwei Phonationszyklen enthält, landet die Grundfrequenz in h_1^{-2} und die zweite Harmonische in h_1.

Die oktavengefilterte harmonische Oberfläche $E^O(1)$ ergibt sich zu

$$E^O_{l,m}(1) = \frac{1}{l} \sum_{j=1}^{H_l} T_{l,m}(h_j). \tag{4.15}$$

In Abbildung 4.9 kann die Wirkung der Oktavenfilter beobachtet werden. Beim Vergleich der Abbildungen 4.7 und 4.9 fällt auf, dass nicht nur Energie verschwindet, sondern auch Energie auftaucht: Rechts unten in Abbildung 4.9 erscheint um die grundfrequenz-skalierte Fensterlänge harmonische Energie. Das liegt am Teilerfilter: Nachdem der Teilerfilter angewendet wurde, ist die Energie in allen harmonischen Körben größer Null. Auslöschungen, die bei der Summenbildung in Gleichung 4.12 möglich sind, werden verhindert. Das ist auch ein Nachteil, weil die Genauigkeit beeinträchtigt wird. Als weiterer Nachteil ist zu nennen, dass Unstetigkeiten in der harmonischen Oberfläche auftreten und Harmonische gelöscht werden, die nicht weit aus dem Rauschen ragen, was ebenfalls die Genauigkeit beeinträchtigt.

Neben der Anwendung der Oktavenfilter hilft zudem die Zentrierung der Frames $x_{l,m}$ um den Abtastpunkt m bei der Vermeidung von Oktavenfehlern. In dieser Arbeit ist es besonders wichtig, dass Frames $x_{l,m}$ zu einem bestimmten Abtastpunkt m, aber für verschiedene Fensterlängen den Beginn stimmhafter Sprache gleichzeitig „sehen“, weil nur eine begrenzte Vorausschau möglich ist. Schauen größere Fensterlängen viel weiter in die Zukunft als kleinere Fensterlängen, so werden Teiler der Grundfrequenz fälschlicherweise als Grundfrequenz aufgefasst. Andererseits wird die Energie bei größeren Fensterlängen durch einen größeren Faktor geteilt (siehe Gleichung 4.15). Schauen kleine Fensterlängen genauso weit in die Zukunft wie große Fensterlängen, so werden Vielfache als Grundfrequenz erkannt, weil die kleineren Fensterlängen das einsetzende Sprachsignal früher vollständig enthalten. Die Frames $x_{l,m}$ um den Abtastpunkt m zu zentrieren ist also ein Kompromiss.

Bemerkungen

Bei klarer Sprache sprechen die vorgestellten Oktavenfilter im Idealfall für eine grundfrequenz-skalierte Fensterlänge nicht an, weil dann in die Körbe h_j^2 und h_j^{-2} keine Energie durch Leakage fliesst. Die Oktavenfilter sind generell kritisch zu betrachten, weil sie Unstetigkeiten in die harmonische Oberfläche einführen. Experimentiert wurde auch mit einem alternativen Teilerfilter, der beim Schätzen des Rauschens die Körbe h_j^{-2} und h_j^2 stärker gewichtet:

$$T'_{l,m}(h_j) = \begin{cases} V_{l,m}(h_j) - \alpha(M_{l,m}(h_j^2) + M_{l,m}(h_j^{-2})),\ \alpha \in [0;1] \text{ und } j = 2, ..., H_l - 1 \\ \hat{M}_{l,m}(h_j) - \alpha(M_{l,m}(h_j^2) + M_{l,m}(h_j^{-2})),\ \alpha \in [0;1] \text{ und } j \in \{1, H_l\}. \end{cases} \tag{4.16}$$

Dieser lieferte bei Testläufen schlechtere Erkennungsraten auf der Aurora-2 Datenbank, ist aber dennoch eine Alternative zu dem oben beschriebenen Teilerfilter, da er zwar die Oberfläche schärfer macht, aber keine Unstetigkeiten in dieser verursacht (siehe auch Kapitel 6).

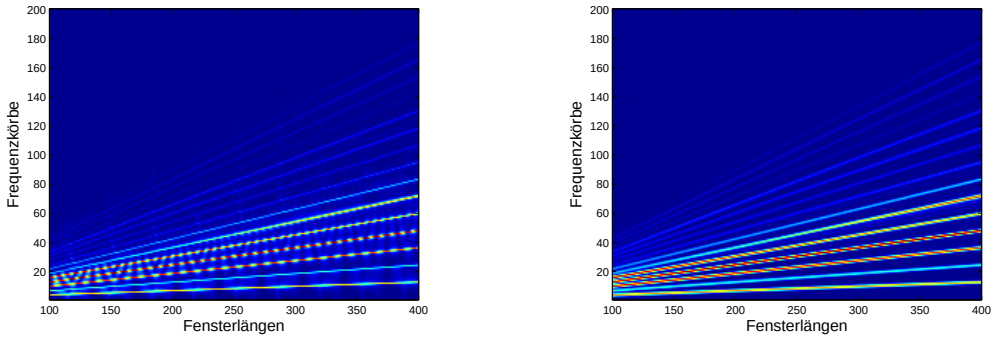


Abbildung 4.10.: Hohe Frequenzen in der harmonischen Oberfläche. Links sieht man die Magnitude der DFTen $X_{l,m}$ ($m \in \mathbb{Z}$ fest, $l \in L$), die mit einem rechteckigen Fenster ausgeschnitten wurden. Rechts wurden die Frames mit einem Hamming-Fenster ausgeschnitten.

4.5.2.3. Oberfläche $E^O(\Delta l_{max})$

In diesem Abschnitt ist es das Ziel, Abtastfehler zu minimieren, wenn eine Spalte E_m mit wenigen Abtastpunkten untersucht wird. Es wird also die harmonische Oberfläche $E^O(\Delta l_{max})$ gesucht. Da es das Ziel ist, nicht die gesamte Spalte E_m zu berechnen, kann keine Tiefpass-Filterung im Nachhinein durchgeführt werden. Stattdessen werden grundsätzliche Überlegungen angestellt, welche Eigenschaften die harmonische Oberläche hat und wie die Berechnung der harmonischen Oberfläche verändert werden kann, um hochfrequente Anteile der Spalten E_m bereits bei der Berechnung der Werte $E^O_{l,m}(\Delta l_{max})$ zu dämpfen. Es zeigt sich dann, dass es ein gegenläufiges Ziel ist, den hochfrequenten Anteil der Spalten E_m zu reduzieren, und gleichzeitig bei starkem Rauschen die Grundfrequenz zu finden. Deshalb wird versucht, einen Kompromiss zu finden.

Eigenschaften der harmonischen Oberfläche

In Abbildung 4.10 ist die Magnitude der DFTen $X_{l,m}$ für verschiedene Fensterlängen zu einem Abtastpunkt m dargestellt. Aus diesen DFTen wird die Spalte E_m berechnet. Zur Vereinfachung werden die Eigenschaften der Spalte E_m in der Nähe der grundfrequenz-skalierten Fensterlänge betrachtet. Andere Maxima werden im Idealfall von den Oktavenfiltern unterdrückt.

In Abbildung 4.11 wird der erste harmonische Korb der DFTen $X_{l,m}$ einer Spalte E_m betrachtet. Ist die Fensterlänge $l = 150$ für m grundfrequenz-skaliert, so findet sich im ersten harmonischen Korb die gesamte Energie der Grundfrequenz. Es geht keine Energie durch Leakage verloren.

Im Weiteren wird überlegt, wie sich die Energie im ersten harmonischen Korb bei Abweichungen von der grundfrequenz-skalierten Fensterlänge verhält.

In Abbildung 4.11 ist dargestellt, wie die harmonische Energie im ersten harmonischen Korb bei einer Fensterlänge von 165 Abtastpunkten ermittelt werden kann,

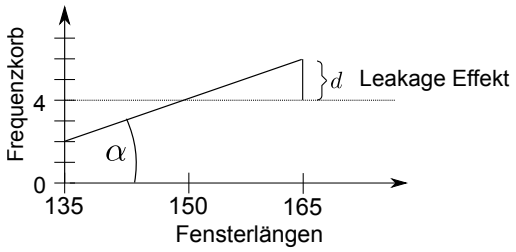


Abbildung 4.11.: Leakage-Effekt und Einfluss der Frequenz.

wenn $l = 150$ die grundfrequenz-skalierte Fensterlänge ist. Die Energie der Grundfrequenz wird durch Leakage auf verschiedene Frequenzkörbe verteilt. Anhand der Frequenzantwort der verwendeten Fensterfunktion kann ermittelt werden, wieviel Energie der Grundfrequenz in den ersten harmonischen Korb leckt (siehe Abbildung 2.1). Geht man davon aus, dass übliche Fensterfunktionen wie ein Hamming-Fenster oder ein Rechteckfenster verwendet werden, so landet die geleckte Energie zum großen Teil innerhalb des Hauptmaximums der Frequenzantwort der Fensterfunktion und die Nebenmaxima sind stark gedämpft (siehe Abbildung 2.1). Daher kann annäherungsweise davon ausgegangen werden, dass in der Spalte E_m in der Nähe der grundfrequenz-skalierten Fensterlänge $l = 150$ der Einfluss der Grundfrequenz im ersten harmonischen Korb überwiegt.

Allgemein hängt es von der Frequenz ab, wie schnell die Energie von einem Frequenzkorb in den Nächsten wandert (vergleiche Gleichung 2.12):

$$\text{Frequenzkorb}(f, l) = \frac{f}{f_s} l \ , l \in L, 0 \leq f \leq \frac{f_s}{2}. \tag{4.17}$$

Energie in höheren Frequenzen wandert daher schneller von einem Frequenzkorb in den Anderen (vergleiche dazu Abbildung 4.10).

Daraus folgt, dass die Distanz d in Abbildung 4.11 bei Abweichungen von der grundfrequenz-skalierten Fensterlänge für hohe Grundfrequenzen schnell groß wird. Dann landet schon bei kleinen Abweichungen von der grundfrequenz-skalierten Fensterlänge nur noch wenig Energie durch Leakage im ersten harmonischen Korb. Allgemein gilt das nicht nur für die Grundfrequenz, sondern für alle Harmonischen: Für Harmonische mit hoher Frequenz leckt schon bei kleinen Abweichungen von der grundfrequenz-skalierten Fensterlänge nur noch wenig Energie in den zugehörigen harmonischen Korb.

In einer Spalte E_m tragen also hohe Harmonische im Signal zum hochfrequenten Anteil in der Nähe der grundfrequenz-skalierten Fensterlänge bei, wenn eine übliche Fensterfunktion verwendet wird. Die Breite des Maximums bei der grundfrequenz-skalierten Fensterlänge kann daher durch eine Dämpung von Harmonischen und die Wahl der Fensterfunktion beeinflusst werden.

Es ist noch nicht berücksichtigt, dass sich bei variierender Fensterlänge l auch das analysierte Signal ändert und dass in den harmonischen Körben auch andere Energie außer der Energie der Harmonischen landet. Ebensowenig ist berücksichtigt, wie sich die Oktavenfilter, das Spectral Subtraction und Rauschen auf die harmonische

Oberfläche auswirken.

Generell flacht die Energie durch das Spectral Subtraction bei Abweichungen von der grundfrequenz-skalierten Fensterlänge stärker ab. Ist die Fensterlänge l nicht grundfrequenz-skaliert, landet harmonische Energie durch Leakage auch neben den harmonischen Körben. Diese Energie wird beim Spectral Subtraction abgezogen. Der Vielfachfilter und der Teilerfilter sind problematisch, weil sie schwer kontrollierbare Unstetigkeiten in der Oberfläche verursachen. Andererseits erreichen die Oktavenfilter im Idealfall, dass Fensterlängen abseits der grundfrequenz-skalierten Fensterlänge gedämpft oder ausgelöscht werden.

Der Teilerfilter hat zusätzlich die Eigenschaft, dass er bei Abweichungen von der grundfrequenz-skalierten Fensterlänge l für hohe Harmonische schneller anspricht. Diese Harmonischen erhalten dann durch den Teilerfilter ein Gewicht von Null. Durch Auslöschungen aufgrund des Spectral Subtraction ist es sonst möglich, dass hohe Harmonische mit negativen Werten die tiefen Harmonischen in Gleichung 4.12 unterdrücken (siehe Abbildung 4.7). Rauschen führt in der Spalte E_m zu Energie bei Fensterlängen abseits der Grundfrequenz oder zu Auslöschungen von Harmonischen bei einer grundfrequenz-skalierten Fensterlänge.

Harmonische

In Abbildung 4.12 sind harmonische Oberflächen zu sehen, bei denen nur der erste Summand aus Gleichung 4.12 verwendet wurde. Zusätzlich sind beide Oktavenfilter eingeschaltet. Vergleicht man die Oberflächen mit denen in Abbildung 4.9, so fällt auf, dass die harmonische Energie um die Grundfrequenz weniger stark abfällt. Für den Sprecher mit tiefer Grundfrequenz verschwindet die Energie auf der harmonischen Oberfläche fast vollständig. Das liegt daran, dass die in der Arbeit verwendeten Sprachsignale aus der Aurora-2 Datenbank stammen [PgHG00]. Die Sprachsignale werden dort jeweils mit dem Filter G.712 vorverarbeitet, welcher eine übliche Frequenzcharakteristik aus der Telekommunikation aufweist [It96]. Der Filter dämpft Frequenzen unterhalb von $300\,Hz$. In Abbildung 4.12 ist unten dargestellt, welche Auswirkungen das auf die Energie in den harmonischen Körben hat. Die durchschnittliche Energie im ersten harmonischen Korb liegt bei Männern deutlich unterhalb der Energie, die im zweiten harmonischen Korb zu finden ist. Bei Frauen verhält sich dies ähnlich. Der Unterschied fällt weniger drastisch aus, weil Frauen generell eine höhere Grundfrequenz aufweisen. Die gedämpfte Energie bei niedrigen Frequenzen erklärt, warum der Teilerfilter für die Grundfrequenz weniger zuverlässig arbeitet als für höhere Harmonische. Die Annahmen über die Energieverteilung stimmen dort nicht mehr. Der harmonische Korb h_1^{-2} enthält für eine Fensterlänge, die halb so groß wie die grundfrequenz-skalierte Fensterlänge ist, die Grundfrequenz. Dort findet sich für Sprecher mit tiefer Grundfrequenz kaum Energie. Der Teilerfilter nimmt im Gegensatz dazu an, dass bei einem Oktavenfehler harmonische Energie in h_1^{-2} fällt.

In Abbildung 4.13 ist ein Beispiel einer Auslöschung abgebildet. Werden bei der Berechnung der harmonischen Oberfläche mehrere Harmonische aggregiert, so sinkt die Wahrscheinlichkeit, dass die Energie in allen harmonischen Körben bei der

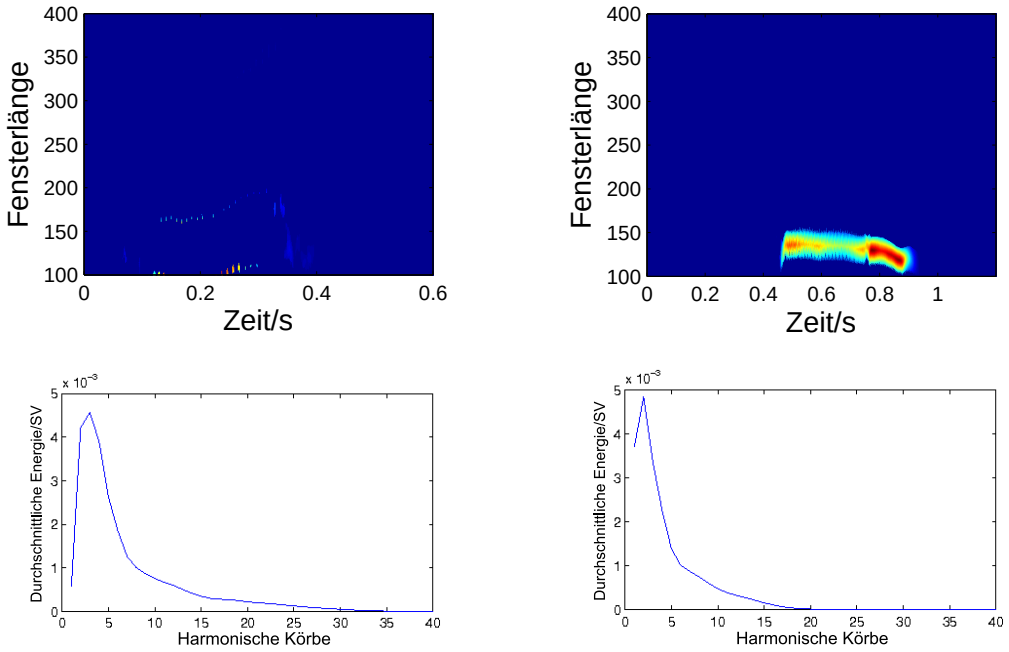


Abbildung 4.12.: Einfluss der Harmonischen. Oben sind jeweils harmonische Oberflächen zu sehen. Nur der erste harmonische Korb aus Gleichung 4.15 wurde berücksichtigt und die Oktavenfilter sind eingeschaltet. Links wurde ein Sprecher mit tiefer Grundfrequenz, rechts eine Sprecherin mit hoher Grundfrequenz analysiert. Unten ist die Verteilung der Energie für Männer (links) und Frauen (rechts) für die harmonischen Körbe dargestellt. Jeweils wurde der Mittelwert für 500 Sprachdateien aus der Aurora-2 Datenbank ([PgHG00]) berechnet. Die harmonische Oberfläche links zeigt, dass der Teilerfilter nicht die gesamte Energie der zweiten Harmonischen löschen konnte, die sich bei einer Fensterlänge von circa 150 Abtastpunkten im ersten harmonischen Korb befindet.

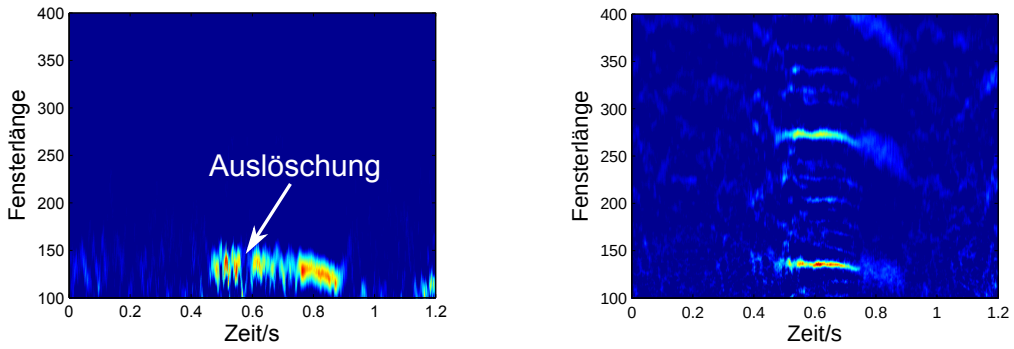


Abbildung 4.13.: Auslöschungen. Links ist eine harmonische Oberfläche dargestellt, welche nur den ersten harmonischen Korb verwendet. Rechts wurden bei der Berechnung der Oberfläche die ersten zehn Summanden berücksichtigt. Die Oberflächen wurden jeweils ohne Oktavenfilter bei einem Signal-Rausch-Verhältnis von $0\,dB$ berechnet.

grundfrequenz-skalierten Fensterlänge von Rauschen überlagert wird. Ebenso sinkt die Wahrscheinlichkeit, dass ein durch Rauschen verursachtes Maximum in der Spalte E_m größer als das Maximum bei der grundfrequenz-skalierten Fensterlänge ist.

Kompromiss

Nachdem einige Überlegungen über die Eigenschaften der harmonischen Oberfläche angestellt wurden, wird jetzt versucht, einen Kompromiss zu finden, um eine Abtastung mit niedriger Abtastrate zu ermöglichen und die Grundfrequenz dennoch bei Rauschen und für Sprecher mit tiefer Grundfrequenz zu erkennen.

Bei der Festlegung der Funktion $E_{l,m}(\Delta l_{max})$ ist es nicht das Ziel, die Spalten E_m zu rekonstruieren. Aufgrund der Unstetigkeit der Oktavenfilter kann das nicht erreicht werden. Das Maximum bei der grundfrequenz-skalierten Fensterlänge soll sich auch bei niedriger Abtastrate von unerwünschten Maxima abheben.

Wie in Abbildung 4.12 zu sehen ist, werden hohe Harmonische durch die gegebene Energieverteilung stärker gedämpft. Das wird durch eine einfache abfallende Funktion verstärkt, die den harmonischen Korb h_j gewichtet:

$$g(j) = max\{0, 1 - \frac{1}{u} \cdot (j-1)\},\ u \in \mathbb{R}_{>0}. \tag{4.18}$$

Harmonische, deren Gewicht gleich Null ist, werden völlig unterdrückt. Für Gewichte kleiner 1 werden die Harmonischen in der Aggregation gedämpft. Der Teilerfilter aus Gleichung 4.14 trägt ebenfalls dazu bei, dass hohe Harmonische weniger Gewicht erhalten, weil sie niedrige Harmonische nicht unterdrücken können. Neben der stärkeren Gewichtung von niedrigen Harmonischen, wird für die Berechnung von $E^O(\Delta l_{max})$ kein Rechteckfenster verwendet, sondern ein Hamming-Fenster. Wie in Abbildung 2.1 dargestellt ist, hat das Hamming-Fenster ein breiteres Hauptmaximum und konzentriert dort mehr Energie, weshalb die Energie in der Nähe der

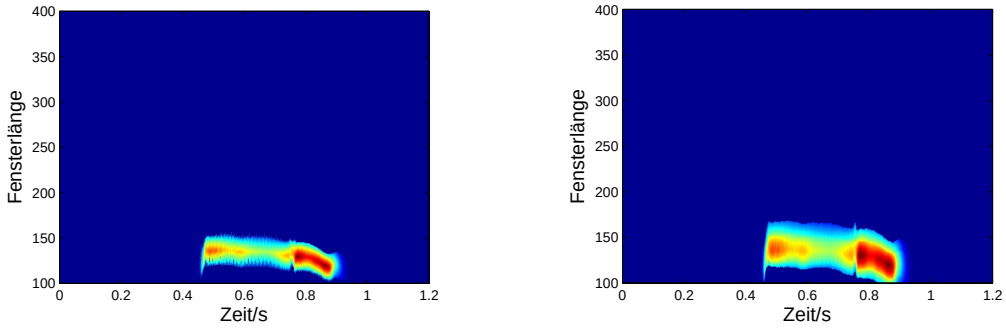


Abbildung 4.14.: Fensterfunktion. Für die Berechnung der linken Oberfläche wurde als Fensterfunktion eine Rechteckfunktion verwendet. Auf der rechten Seite wurde ein Hamming-Fenster benutzt. In beiden Fällen wurde jeweils nur der erste harmonische Korb berücksichtigt und die Oktavenfilter sind eingeschaltet.

grundfrequenz-skalierten Fensterlänge schwächer abflacht. In Abbildung 4.14 wurde rechts die harmonische Oberfläche berechnet, wobei ein Hamming-Fenster zum Ausschneiden der Frames verwendet wurde. Das Rauschen wird nicht in allen nichtharmonischen Körben geschätzt. In Abbildung 2.1 ist zu erkennen, dass Rauschen für einen Frame $x_{l,m}$, der mit einem Hamming-Fenster ausgeschnitten wurde, in den Körben h_j^{-2} und h_j^2 geschätzt werden kann, weil dort für einen grundfrequenz-skalierte Fensterlänge keine harmonische Energie durch Leakage landet:

$$R^h_{l,m}(h_j) = \frac{1}{2} \cdot ((M^h_{l,m}(h_j^2) + M^h_{l,m}(h_j^{-2})). \tag{4.19}$$

Die gefilterten harmonischen Körbe $\hat{M}^h_{l,m}(h_j)$ ergeben sich analog zu Gleichung 4.10:

$$\hat{M}^h_{l,m}(h_j) = M^h_{l,m}(h_j) - R^h_{l,m}(h_j). \tag{4.20}$$

Die Oktavenfilter arbeiten ausschließlich auf den harmonischen Körben h_j und den mittleren Körben h_j^{-2} und h_j^2. Aus diesem Grund können die Oktavenfilter $T^h_{l,m}(h_j)$ und $V^h_{l,m}(h_j)$ bei der Verwendung eines Hamming-Fensters analog zu den Gleichungen 4.13 und 4.14 definiert werden. Es ergibt sich die harmonische Oberfläche $E^O(\Delta l_{max})$ (siehe Abbildung 4.15):

$$E^O_{l,m}(\Delta l_{max}) = \frac{1}{\sum_{i=1}^{l} w(n)} \sum_{j=1}^{H_l} g(j) T^h_{l,m}(h_j). \tag{4.21}$$

Die Normalisierung wird analog zur Berechnung von $E(1)$ als Summe der Werte der Fensterfunktion vorgenommen (siehe Gleichung 4.15). Neben der Normalisierung der Energie für verschiedene Fensterlängen wird zusätzlich zu erreichen versucht, dass die Energieunterschiede zwischen den harmonischen Oberflächen $E^O_{l,m}(\Delta l_{max})$ und $E^O_{l,m}(1)$ klein bleiben. Systematische Energieunterschiede zwischen den Berechnungsmethoden ergeben sich durch das unterschiedliche Leakage der verwendeten

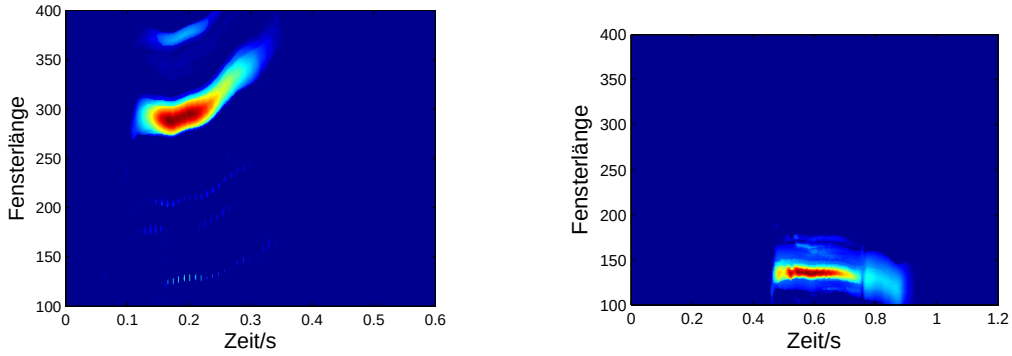


Abbildung 4.15.: $E^O(\Delta l_{max})$. Für zwei Sprachsignale sind links und rechts harmonische Oberflächen $E^O(\Delta l_{max})$ zu sehen. Für tiefe Grundfrequenzen flachen die Maxima schwächer ab. Links fehlt die großflächige hellblaue Umrandung, die rechts zu beobachten ist, weil links die Grundfrequenz im Signal stark gedämpft ist. Der Parameter u wurde als 9 gewählt.

Fensterfunktionen, die Gewichtung der Harmonischen und durch die unterschiedliche Form des Spectral Subtraction. Ist die Fensterlänge l grundfrequenz-skaliert, wird bei klarer Sprache im Idealfall in beiden Fällen keine Energie durch Spectral Subtraction abgezogen (siehe Abbildung 2.1). Für die Berechnung von $E^O_{l,m}(\Delta l_{max})$ wird ein Hamming-Fenster wie in Abschnitt 5.2 verwendet. Deshalb ist für $l \in L$

$$\sum_{n=1}^{l} w(n) = 0.54 \cdot l - 0.46 \sum_{n=0}^{l-1} cos(2\pi \frac{n}{l}) = 0.54 \cdot l. \tag{4.22}$$

Aus diesem Grund ist die Energie für grundfrequenz-skalierte Fensterlängen l in den harmonischen Körben bei Verwendung eines Hamming-Fensters oder eines Rechteck-Fensters wegen der Normalisierung gleich, weil die nicht-harmonischen Körbe bei Verwendung eines Rechteckfensters im Idealfall keine Energie enthalten (siehe Gleichung 5.1). Die Gewichtung der harmonischen Körbe führt dann trotzdem zu Energieunterschieden.

Abgesehen von der Gewichtung der harmonischen Körbe, ist die harmonische Energie für eine grundfrequenz-skalierte Fensterlänge l daher für die harmonischen Oberflächen $E(1)$, $E^O(1)$ und $E^O(\Delta l_{max})$ im Idealfall gleich groß.

4.5.2.4. Oberfläche $E^O(\Delta l)$

Es stehen jetzt Berechnungsmethoden für die Oberflächen $E^O(\Delta l_{max})$ und $E^O(1)$ zur Verfügung. Es wurde versucht, $E^O(\Delta l_{max})$ so zu berechnen, dass das Maximum der Spalten $E^O_m(\Delta l_{max})$ mit möglichst wenigen Abtastpunkten erkannt werden kann. Die Oberfläche $E^O(1)$ wurde so berechnet, dass das Maximum möglichst genau bestimmt werden kann (dazu können bei kleinem Synchronitätsradius gegebenfalls auch die Oktavenfilter ausgeschaltet werden). Dabei wurde zusätzlich zu berücksichtigen

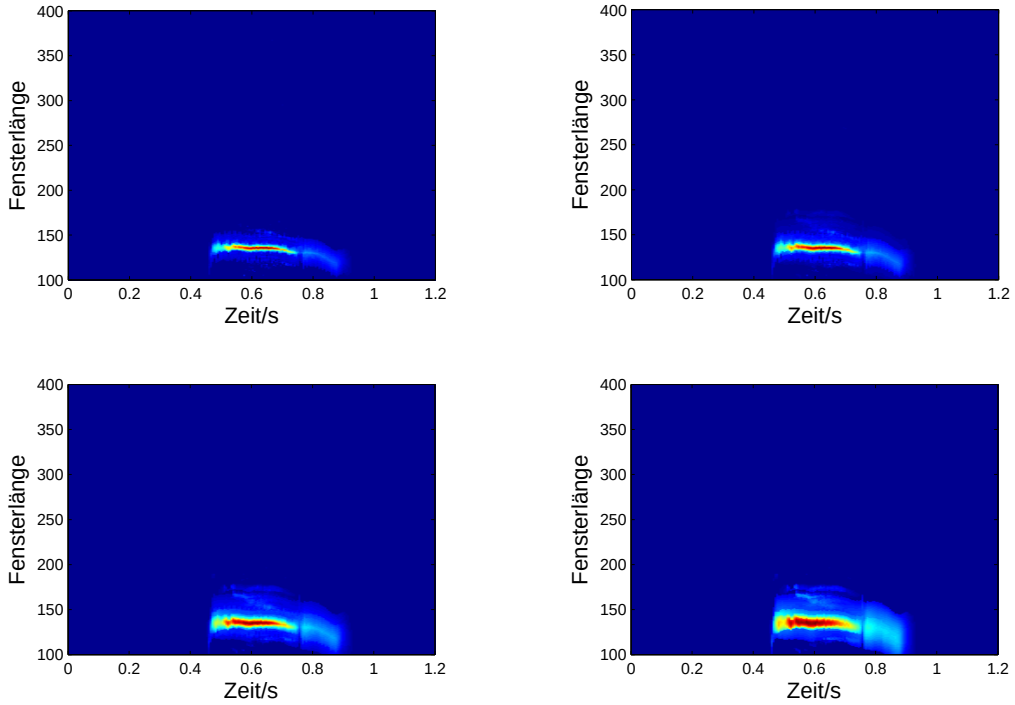


Abbildung 4.16.: $E^O(\Delta l)$. Für verschiedene Werte Δl sind harmonische Oberflächen dargestellt. Von links nach rechts und von oben nach unten steigt Δl vom Minimum zum Maximum an. Der Parameter u wurde als $\frac{1}{9}$ gewählt.

versucht, dass die Energieunterschiede $|E^O_{l,m}(1) - E^O_{l,m}(\Delta l_{max})|$ für grundfrequenzskalierte Fensterlängen klein bleiben sollen.

Um die Oberflächen für $\Delta l \in \{1, ..., \Delta l_{max}\}$ zu erhalten, wird linear interpoliert (siehe Abbildung 4.16):

$$E^O_{l,m}(\Delta l) = \sum_{j=1}^{H_l} g(j,a) \cdot (a \cdot \frac{1}{\sum_{j=1}^{l} w(n)} T^h_{l,m}(h_j) + (1-a) \cdot \frac{1}{l} T_{l,m}(h_j)) \ , \tag{4.23}$$

$$\text{mit, } a = \frac{\Delta l - 1}{max\{1, \Delta l_{max} - 1\}} \text{ und} \tag{4.24}$$

$$g(j,a) = max\{0, 1 - \frac{a}{u} \cdot (j-1)\}, \ u \in \mathbb{R}_{>0} \tag{4.25}$$

Bemerkungen

Dass die Interpolation linear durchgeführt wird, ist eine Vereinfachung. Ist Δl_{max} klein genug (im Extremfall 1), so wird es für die Zwischenschritte Δl sowie für Δl_{max} keine Abtastprobleme geben. Ist Δl_{max} groß, und die Abtastrate reicht aus, um Abtastfehler zu vermeiden, so ist es möglich, dass für Zwischenschritte Δl Abtastfehler

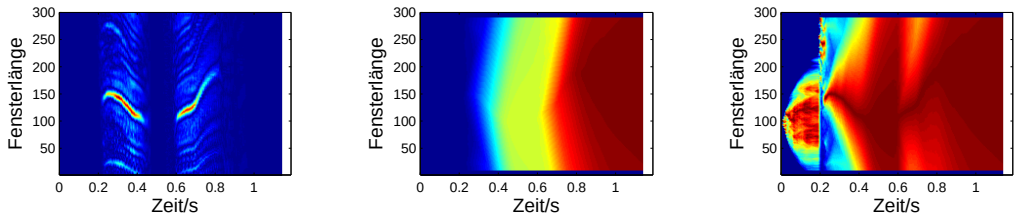


Abbildung 4.17.: Grundfrequenzverfolgung. Links ist eine harmonische Oberfläche E zu sehen, wie Behnke sie verwendete [Beh03]. In der Mitte sind die akkumulierten Maxima A_i zu sehen, welche direkt aus der Oberflaeche E abgeleitet werden. Rechts wurden die Spalten A_i zur Visualisierung auf Werte zwischen 0 und 1 skaliert. In roten Bereichen wurde viel harmonische Energie gesammelt. Über gelb nach blau wurde immer weniger harmonische Energie gesammelt.

auftreten. Das nicht ausreichende Wissen über die richtige Art der Interpolation kann durch einen kleinen Wert von Δl_{max} ausgeglichen werden. Alternativ kann versucht werden, die Zuordnung Δl nach a nicht linear vorzunehmen, sondern von Hand oder mit anderen Verfahren zu optimieren.

Abhängig von Δl_{max} kann der Parameter u eingestellt werden. Das Hamming-Fenster wird bei der Berechnung von $E^O(\Delta l_{max})$ grundsätzlich verwendet (außer für $\Delta l_{max} = 1$). Dabei wird davon ausgegangen, dass Δl_{max} so groß ist, dass mindestens die Verwendung des Hamming-Fensters für die Vermeidung von Abtastfehlern notwendig ist. Für die Berechnung von $E^O_{l,m}(\Delta l_{max})$ wird das Hamming-Fenster vorgezogen, weil dafür keine Harmonischen bei der Berechnung der Oberfläche unterdrückt werden, die bei der Aggregation helfen, aus dem Rauschen zu ragen.

4.5.3. Iterative Verarbeitung

Das Ziel der iterativen Verarbeitung ist es, die Grundfrequenzschätzung in jedem Iterationsschritt zu verbessern. Dazu wird die Grundfrequenz in jeder Iterationsebene verfolgt. Dabei erhält die Iterationsebene $b \in \{1, ..., B\}$ Informationen darüber, wo die Iterationsebene $b-1$ die Grundfrequenz vermutet und wie sicher diese Schätzung ist. In der Iterationsebene b kann die Grundfrequenzschätzung mithilfe dieser Informationen dann verfeinert werden. Die Grundfrequenzverfolgung für jede einzelne Iterationsebene ist an das Vorgehen von Behnke angelehnt [Beh, Beh03]. Dieses Vorgehen wird daher kurz vorgestellt.

Abtastpunkten $t_i \in \mathbb{Z}$, $i \in \mathbb{Z}$ soll jeweils eine Grundfrequenz zugeordnet werden. Von der harmonischen Oberfläche E des Sprachsignals x werden die *akkumulierten Maxima* $A_i \in \mathbb{R}^{|L|}$ abgeleitet (siehe Abbildung 4.17). Um die Spalte $A_i \in \mathbb{R}^{|L|}$ zu berechnen, werden die Spalten A_{i-1} und E_{t_i} benötigt[1].

Für die Berechnung des akkumulierten Maximums A_i wird eine *Nutzenfunktion* verwendet [RBB07]. Als Nutzen wird berücksichtigt, dass ein großer Wert E_{l,t_i} ein

[1] Es gilt $t_{i+1} = t_i + 40$. Die harmonische Oberfläche wird nicht für alle Abtastpunkte berechnet, sondern alle 5 ms wird eine Spalte benötigt.

Hinweis auf die Grundfrequenz ist. Bei der Berechnung der Nutzenfunktion fließt ein, dass sich die Grundfrequenz nur langsam ändert.

Der Eintrag $A_{l,i}$ enthält den Nutzen des Grundfrequenzpfades, welcher bezüglich der Nutzenfunktion optimal ist und zum *Knoten* (l, i) führt[2]. Als Grundfrequenzpfad für einen Abtastpunkt $t_{i_{max}} \in \mathbb{Z}$ wird der Pfad gewählt, welcher zu einem maximalen Wert innerhalb der Spalte $A_{i_{max}}$ führt.

Die Konfidenz für Abtastpunkte $t_i \in \mathbb{Z}$ wird berechnet, indem die harmonische Energie auf dem Grundfrequenzpfad für den Abtastpunkt $t_i \in \mathbb{Z}$ mit einem Energielevel $e_i \in R$ verglichen wird.

4.5.3.1. Erweiterung der Iterationskette i

In diesem Abschnitt liegt die Situation aus Abbildung 4.5 vor. Für die b-te Iterationsebene soll der Iterationspunkt (i, b), $b \in \{2, ..., B\}$ berechnet werden.

Vom Iterationspunkt $(i-1, b)$ ist das akkumulierte Maximum $A^b_{i-1} \in \mathbb{R}^{|L|}$ der Iterationsebene b bekannt. Durch den Iterationspunkt $(i, b-1)$ sind ein Synchronitätsgrad $s \in [0; 1]$, eine Grundfrequenzschätzung $l \in L$ sowie eine Konfidenz $k \in [0; 1]$ gegeben.

Synchronitätsgrad

Abhängig von s und k wird ein Synchronitätsgrad für den Iterationspunkt (i, b) ermittelt.

$$s(i,b) = \begin{cases} s \text{ für } k < 1 \\ min\{s + c, 1\} \text{ für } k = 1 \end{cases} \tag{4.26}$$

Das bewirkt, dass für konfidente Messungen in der Iterationsebene $b-1$ der Synchronitätsgrad in der Iterationsebene b um eine Konstante $c \in [0; 1]$ erhöht wird. Der Parameter c kann zum Beispiel so gewählt werden, dass $c \cdot (B-1) = 1$ ist. Beträgt der Synchronitätsgrad im ersten Iterationsschritt Null, kann der Algorithmus sich dann bis zum letzten Iterationsschritt vollständig synchronisieren. Der Synchronitätsgrad $s(i, b)$ und eine Grundfrequenzschätzung l sind jetzt gegeben. Daher kann die Abtastdistanz Δl berechnet werden und es ergibt sich eine Menge von Fensterlängen $F \subseteq L$ (siehe Gleichung 4.7).

Harmonische Oberfläche

Für diese Menge von Fensterlängen wird dann die Spalte $E^b_{t_i}(\Delta l)$ berechnet. Mittels einer Nearest-Neighbor-Interpolation werden die Werte der Spalte $E^b_{t_i}(\Delta l)$ für die fehlenden Fensterlängen $l' \in L \setminus F$ festgelegt. In Abschnitt 4.5.2 wurde versucht, die Fehler, die durch diese Interpolation entstehen, zu minimieren. Es zeigt sich in Kapitel 6, dass eine Nearest-Neighbor Interpolation leicht bessere Ergebnisse erzielt, als

[2] Stellt man sich die Matrix A als Graphen mit gerichteten Kanten zwischen den Spalten A_i und A_{i+1} vor, ist der verwendete Algorithmus zum Berechnen des größten Nutzen von Knoten (l, i) ähnlich zum Dijkstra Algorithmus.

eine alternativ verwendete lineare Interpolation. Die Interpolation für Zwischenwerte stellt eine Art der Normalisierung dar: Die Spalten der harmonischen Oberfläche E^b werden für variierende Mengen von Fensterlängen berechnet. Die Spalten werden dann durch die Interpolation vergleichbar gemacht.

Akkumuliertes Maximum und Grundfrequenzschätzung

Nach der vorgenommen Normalisierung wird das akkumulierte Maximum A_i^b dann analog zum Vorgehen von Behnke berechnet [Beh]. Für Fensterlängen $l \in L \setminus F$, die nur einen nächsten Nachbar in F haben, werden die Einträge des akkumulierten Maximums auf $\min\{A_i^b\}$ gesetzt. Das soll bewirken, dass A_i^b nicht für Fensterlängen maximal werden kann, für welche die harmonische Oberfläche nicht berechnet wurde, aber noch akkumulierte harmonische Energie aus der Vergangenheit vorhanden ist. Pfaden, die außerhalb der berechneten harmonischen Oberfläche liegen, wird daher ein niedriger Nutzen zugeordnet.

Jetzt wird für den Iterationspunkt (i, b) eine Grundfrequenzschätzung $l(i, b) \in L$ gewählt. In dieser Arbeit sind weder A noch E eines Sprachsignals vollständig bekannt, bevor eine Entscheidung über den Grundfrequenzpfad getroffen werden muss: Zum Abtastpunkt t_i wird die Spalte E_{t_i} nicht für alle Fensterlängen berechnet, außerdem sind aufgrund der begrenzten Vorausschau nur wenige Spalten $E_{t'_{i'}}$ für $i' > i$ bekannt.

Zum Abtastpunkt t_i wird die Fensterlänge $l(i, b)$ als Schätzung der Grundfrequenz ausgewählt, deren Nutzen bezüglich der Nutzenfunktion maximal ist. Innerhalb der Spalte A_i ist $A_{l(i,b),i}$ also maximal.

Es wird in Kauf genommen, dass die Grundfrequenzschätzung von Abtastpunkt t_{i-1} zu Abtastpunkt t_i stärker springen kann, als das die Nutzenfunktion für Pfade zulässt. Vor einsetzender Sprache ist die Grundfrequenz nicht definiert, und während stimmhafter Sprache, kann der Synchronitätsradius große Sprünge vermeiden. Die Verwendung des Pfades mit der meisten gesammelten harmonischen Energie ist bei großem Synchronitätsradius in frühen Iterationsschritten hilfreich, damit Sprünge durch Ausreißer in der harmonischen Oberfläche vermieden werden. Diese Ausreißer können zum Beispiel durch Rauschen oder ein Versagen der Oktavenfilter hervorgerufen werden.

Informationen aus der Zukunft über die harmonische Oberfläche für die Berechnung des Iterationspunktes (i, b) werden an dieser Stelle ignoriert (siehe Kapitel 7). Diese könnten gegebenenfalls helfen, zum Beginn eines Sprachsignals Sprünge besser zu vermeiden und den Grundfrequenzpfad zu glätten.

Konfidenz

Schließlich wird die Konfidenz $k(i, b)$ ähnlich zum Vorgehen von Behnke berechnet [Beh]. Das Energielevel $e(i, b)$ wird als

$$e(i, b) = 0.99 \cdot e(i - 1, b) + 0.01 \cdot max\{E_{t_i}^b\} \tag{4.27}$$

berechnet. Das Energielevel passt sich durch den rekursiven Filter langsam an die gemessenen Maximalwerte an. Das bewirkt eine gewisse Invarianz gegenüber der Variabilität von Rausch- und Signalstärke. Ebenso passt sich das Energielevel zu einem gewissen Grad an systematische Energieunterschiede der harmonischen Oberflächen $E(\Delta l)$ an. Dennoch ist es wünschenswert, dass die Berechnungsvorschriften $E(\Delta l)$ normalisiert werden, so dass $|E_{l,m}(\Delta l) - E(_{l,m}\Delta l')|$ klein ist (siehe Abschnitt 4.5.2). Die Konfidenz ergibt sich durch Vergleich der Energie auf dem Grundfrequenzpfad mit dem oben berechneten Energielevel [Beh]:

$$k(i,b) = min\{1, \frac{E^b_{l(i,b),t_i}}{e(i,b)}\}. \tag{4.28}$$

Die Entscheidung, ob ein Abtastpunkt t_i stimmhaft ist, wird anhand der Konfidenz $k(i,B)$ der B-ten (abschließenden) Iterationsebene getroffen: Erst werden Ausreißer mit einem Median-Filter entfernt. Abtastpunkte, zu denen die gefilterte Konfidenz eins beträgt, werden als stimmhaft gewertet. Um den Median-Filter zu berechnen, werden die aktuellsten Konfidenzen der benachbarten Iterationsketten verwendet, damit keine zusätzliche Vorausschau benötigt wird. Um den Support für den Median-Filter zu gewährleisten, muss die Anzahl der Iterationsebenen (B) größer als 3 sein.

Zusammenfassend zeigt die Iterationsebene $b-1$ der Iterationsebene b, wo die Grundfrequenz vermutet wird, und wie sicher die Schätzung ist. Beruhend auf diesen Informationen wird dann die Spalte $E^b_{t_i}$ erneut untersucht und eine neue Schätzung wird abgegeben.

Bemerkungen

Durch die Verwendung des rekursiven Filters bei der Berechnung des Energielevels tritt eine Phasenverschiebung auf. Generell ist der Energielevel zu Beginn eines stimmhaften Abschnitts niedrig und bleibt nach einem stimmhaften Abschnitt zunächst hoch. Dieses Verhalten ist erwünscht, da nur dem lautesten Sprecher zugehört werden soll. Nachteilig ist das, wenn ein Sprecher leiser wird, weil der Energielevel dann gegebenenfalls zu hoch ist.

Es wurde für alle Iterationsebenen das gleiche Verhalten gewählt. Das heißt, dass die Verarbeitung nur abhängig vom Synchronitätsradius und nicht abhängig von der Iterationsebene ist. Durch die Propagierung des Synchronitätsgrades durch die Iterationsebenen, ist es dennoch so, dass die Iterationsebene b höchstens den Synchronitätsgrad der Ebene $b+1$ aufweist. Dass der Synchronitätsgrad nicht unabhängig von der Konfidenz erhöht wird, soll bewirken, dass keine falschen Entscheidungen getroffen werden müssen, wenn keine stimmhafte Sprache vorliegt. Dafür ist es allerdings notwendig, dass alle Iterationsebenen mit variierendem Synchronitätsgrad arbeiten. Wie sich in Kapitel 6 zeigt, ist diese Wahl mit Nachteilen verbunden, weil daher in allen Iterationsebenen die Oktavenfilter verwendet werden, die schwer kontrollierbar sind.

In dieser Arbeit wurde der Vielfachfilter für einen Synchronitätsradius kleiner 100 ausgeschaltet, der Teilerfilter blieb immer angeschaltet. Es muss dabei abgewogen

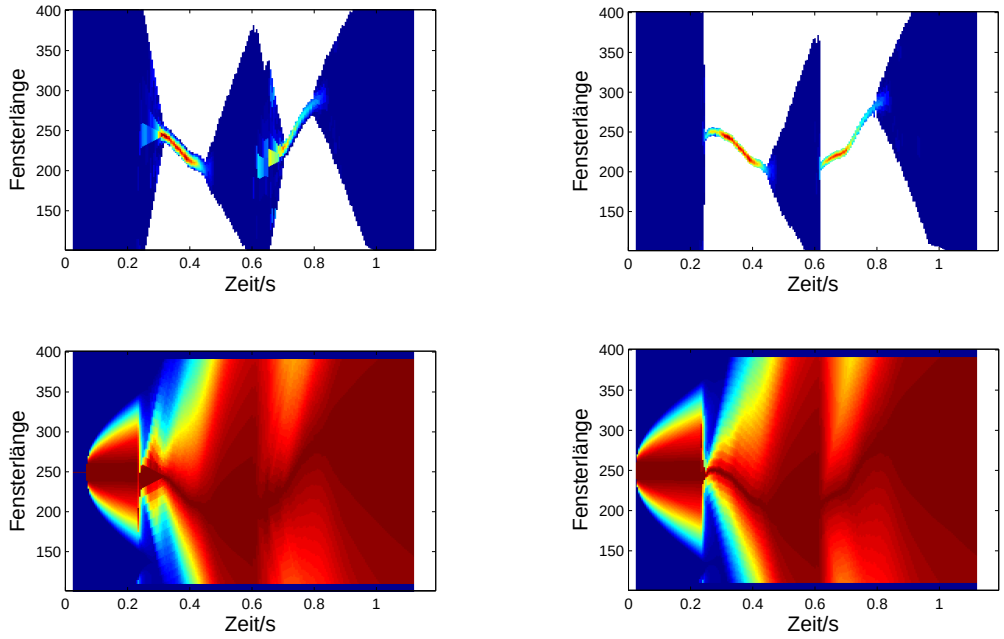


Abbildung 4.18.: Iterationsebenen. Links sind E^1 und A^1 der 1-ten Iterationsebene abgebildet. Rechts sind E^{10} und A^{10} des Iterationsendes zu sehen. Die weißen Bereiche der harmonischen Oberflächen E wurden nicht berechnet. Für alle Spalten E_m wurde die harmonische Energie für die gleiche Anzahl von Fensterlängen ausgerechnet.

werden, dass die Energie der harmonischen Oberfläche dabei variiert, aber andererseits die Genauigkeit der Berechnungen erhöht werden kann und weniger stimmhafte Sprache verfehlt wird, die fälschlicherweise gelöscht wird. Generell kann auch damit experimentiert werden, die Oktavenfilter schrittweise auszuschalten. Das wurde in der vorliegenden Arbeit nicht näher untersucht. Eine beispielhafte Erläuterung des Verhaltens des Algorithmus für klare und mit Rauschen behaftete Sprache findet sich in Abschnitt 6.5.

4.5.3.2. Iterationsanfang

Für den Iterationsanfang $(i, 1)$ der i-ten Iterationskette werden Synchronitätsgrad, Grundfrequenzschätzung und Konfidenz vom Iterationspunkt $(i-1, 1)$ übernommen. Der Synchronitätsgrad wird dann ähnlich wie oben angepasst:

$$s(i,1) = \begin{cases} max\{0, s - c_1\} \text{ für } k < 1 \text{ und } c_1 \in [0;1] \\ min\{s + c_2, 1 - c_3\} \text{ für } k = 1 \text{ und } c_2, c_3 \in [0;1] \end{cases} \tag{4.29}$$

Bei der Wahl von c_2 ist zu berücksichtigen, dass ein großer Wert zu einer Festlegung in der ersten Iterationsebene auf eine Grundfrequenz nach wenigen konfidenten Messungen führt. Wird c_2 klein gewählt, führt dies dazu, dass Δl lange groß bleibt und

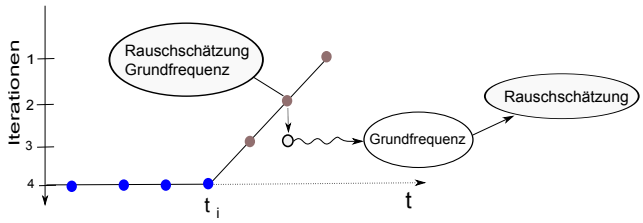


Abbildung 4.19.: Schrittweise Filterung. Überblick über die Schrittweise Filterung.

daher die Vorteile eines hohen Synchronitätsgrades nicht genutzt werden. c_1 steuert, wie stark der Synchronitätsradius steigt, wenn keine konfidente Messung vorliegt. Dieser Wert wird im Idealfall so gewählt, dass der Synchronitätsradius schnell genug steigt, um eine neu einsetzende Äußerung einzufangen und gleichzeitig so langsam steigt, dass keine Oktavenfehler auftreten. Der Synchronitätsgrad wird beim Iterationsanfang durch c_3 nach oben begrenzt, weil das Signal dort zum ersten Mal gesehen wird, die Grundfrequenz sich in der Regel ändert, und daher keine Sicherheit über die Grundfrequenz bestehen kann.

Beim Iterationsanfang springt die zentrale Fensterlänge nur dann auf das Maximum der Spalte A_i, wenn eine konfidente Messung vorliegt. Ansonsten geht in die Berechnung der zentralen Fensterlänge gewichtet auch die bisherige zentrale Fensterlänge $l(i-1,1)$ und die Fensterlänge 250 in die Berechnung der neuen zentralen Fensterlänge ein. Zum einen soll das bewirken, dass die zentrale Fensterlänge sich nur langsam von der Oktave fortbewegen kann, wo zuletzt stimmhafte Sprache vermutet wurde. Zum anderen wird die zentrale Fensterlänge in Richtung der Fensterlänge von 250 gezogen, damit eventuelle Oktavenfehler überwunden werden können. Bei einem Synchronitätsgrad von Null werden dann die Fensterlängen 100 bis 400 abgedeckt, wenn die zentrale Fensterlänge nach 250 gezogen wurde (siehe Abschnitt 4.5.1).

4.5.4. Schrittweise Filterung

In diesem Abschnitt wird versucht, Lärmschätzung und Grundfrequenzschätzung iterativ zu verbessern. Dazu werden Schätzungen des Rauschspektrums und der Grundfrequenz aufeinander aufbauend entlang der Iterationsketten verbessert (siehe Abbildungen 4.19 und 4.20). Das Signal soll also schrittweise immer stärker gefiltert werden. Zum Iterationsbeginn ist wenig über das Rauschen bekannt, weil dort das Signal noch nicht gesehen wurde. Dort wird stationäres Rauschen angenommen, welches als gemittelte Rauschschätzung früherer Iterationsketten berechnet berechnet wird. Die Rauschschätzung wird dann zum Iterationsbeginn bei der Berechnung der harmonischen Oberfläche zu einem kleinen Teil von den Spektren $X_{l,m}$ abgezogen. Dabei besteht die Hoffnung, dass die Grundfrequenz besser erkannt werden kann, wenn die Spektren schon leicht gefiltert wurden. Da die Rauschschätzung noch ungenau ist, wird nur ein kleiner Teil der Rauschschätzung abgezogen.

Der Ansatz ist ähnlich zum Vorgehen im ETSI Standard ES 202 050, bei dem zunächst ein Two-Stage Wiener-Filter angewandt wird, bevor versucht wird, Ver-

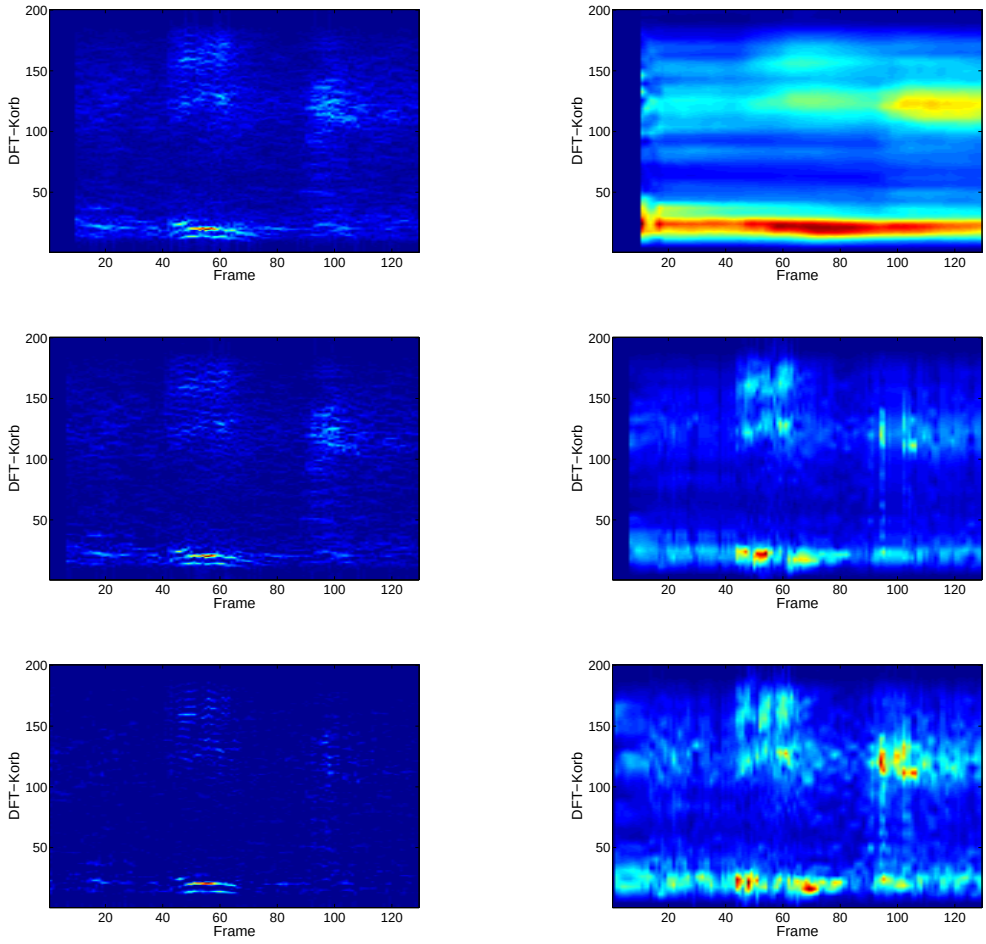


Abbildung 4.20.: Rausch- und Signalschätzungen. Links ist eine STFT eines verrauschten Sprachsignals zu sehen, das von oben nach unten immer stärker gefiltert wurde. Rechts sind die zugehörigen Rauschschätzungen zu sehen, anhand derer die DFTen $X_{l,m}$ für die Grundfrequenzschätzung gefiltert wurden. Oben rechts wurde stationäres Rauschen geschätzt. Die Magnitude wird von rot über gelb nach blau geringer.

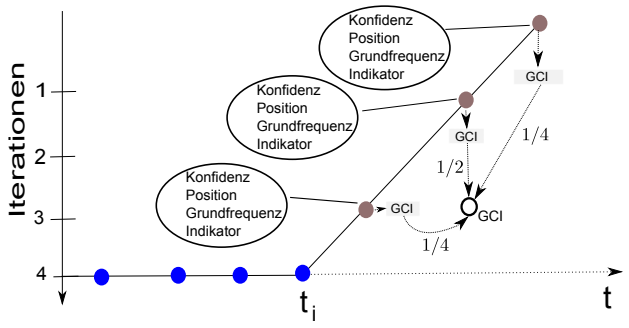


Abbildung 4.21.: Verschlussmomente. Überblick über die iterative Schätzung der Verschlussmomente.

schlussmomente zu finden (siehe Abschnitt 3.2). Bei stimmhafter Sprache kann dann in späteren Iterationsebenen ein größerer Teil der Rauschschätzung abgezogen werden, weil die Grundfrequenzschätzung genauer wird, und daher nicht-stationäres Rauschen besser geschätzt werden kann. Das Rauschen wird wie beim Harmonic Tunneling zwischen den Harmonischen geschätzt (siehe Abschnitt 3.2). Problematisch bei diesem Ansatz ist, dass die Rauschschätzung für die Frames $X_{l,m}$ auf verschiedene Fensterlängen interpoliert werden muss. Ebenso kann das Spektrum des Rauschens nicht in allen Körben geschätzt werden, was Abtastprobleme mit sich bringt.

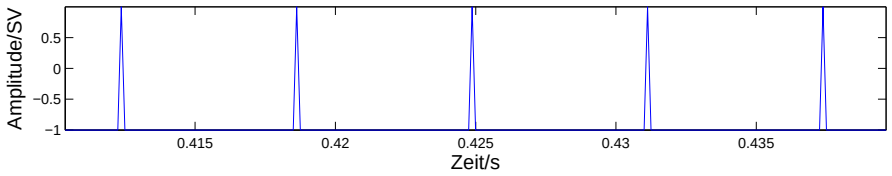
4.6. Verschlussmomente

In diesem Abschnitt wird erläutert, wie die Verschlussmomente geschätzt werden. Dazu wird erst ein Indikator vorgestellt, der auf Verschlussmomente zeigt. Im Anschluss wird die iterative Schätzung der Verschlussmomente dargestellt und die glottissynchrone Verarbeitung beschrieben, bei der die Frames $x_{l,m}$ glottissynchron plaziert werden.

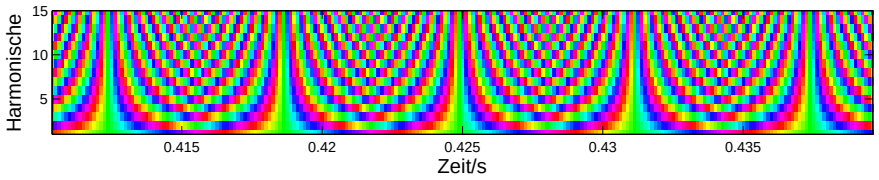
In Abbildung 4.21 ist im Überblick dargestellt, wie die iterative Schätzung der Verschlussmomente vonstatten geht. Um den Verschlussmoment für den Iterationspunkt (i, b) zu bestimmen, werden Verschlussmomentschätzungen der Iterationspunkte $(i-1, b)$, $(i, b-1)$ und $(i+1, b-2)$ gewichtet zusammengeführt. Ein Indikator im Zusammenhang mit einer Position und einer Grundfrequenzschätzung ergeben die vorläufige Verschlussmomentschätzung eines Iterationspunktes.

4.6.1. Indikator

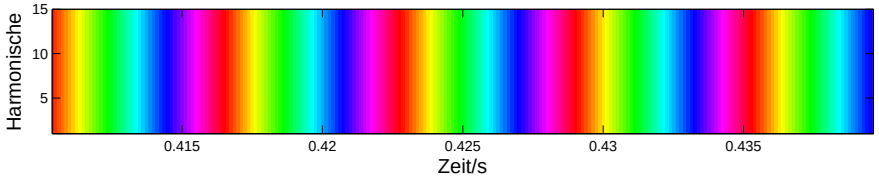
In diesem Abschnitt wird die Berechnung des Indikators beschrieben, der auf Verschlussmomente verweist. Dazu wird anhand des idealisierten Beispiels in Abbildung 4.22 erläutert, wie sich die Phase in den harmonischen Körben für grundfrequenzskalierte Frames eines periodischen Signals verhält.



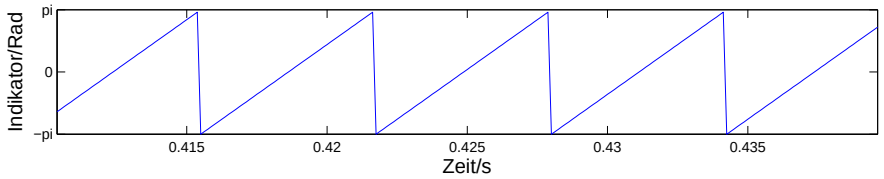
(a) Signal



(b) Phase



(c) Phasendifferenzen



(d) Indikator

Abbildung 4.22.: Idealisierte Betrachtung. In Abbildung 4.22(a) ist ein idealisiertes Anregungssignal dargestellt. In Abbildung 4.22(b) ist die Phase in den harmonischen Körben von grundfrequenz-skalierten DFTen $X_{l,m}$ zu sehen. In Abbildung 4.22(c) sind die Phasendifferenzen aus Gleichung 4.31 dargestellt. Die Phasen werden als Farbton im HSV-Farbraum kodiert. Grün entspricht einer Phase von 0, rot entspricht $\frac{-\pi}{2}$, blau entspricht $\frac{\pi}{2}$ und lila entspricht $\pm pi$. Unten ist der Indikator I aufgetragen.

Idealisierte Betrachtung

Zunächst wird für einen Frame $x_{l,m}$ die *korrigierte Phase* des j-ten harmonischen Korbes

$$p_{l,m}(j) := \angle(X_{l,m}(4 \cdot j) \cdot e^{i \cdot \frac{2\pi}{l} 4 \cdot j \cdot \lfloor \frac{l}{2} \rfloor}) \in [-\pi; \pi],\ j \in \{1, 2, ..., H_l\} \tag{4.30}$$

definiert. Die Korrektur wird vorgenommen, weil die Frames $x_{l,m}$ für gerade und ungerade Fensterlängen unterschiedlich um m liegen (siehe Gleichung 4.2). Die *Phasendifferenzen* für den j-ten harmonischen Korb des Frames $x_{l,m}$ werden als

$$pd_{l,m}(j) := p_{l,m}(j+1) - p_{l,m}(j)\ ,\ j \in \{1, 2, ..., H_l - 1\} \tag{4.31}$$

definiert[3].

In Abbildung 4.22(a) ist ein Sprachsignal x zu sehen, das ein idealisiertes Quellsignal darstellt, welches lediglich einen Impuls zum Verschlussmoment enthält. Eine Fensterlänge $l' \in L$ mit $P = \frac{l}{4}$ ist in diesem Beispiel für alle Abtastpunkte m grundfrequenz-skaliert. Ein idealisierter Phonationszyklus dauert also $\frac{P}{f_s}$ Sekunden.

In Abbildung 4.22(b) ist die Phase der grundfrequenz-skalierten DFTen $X_{l',m}$ zu sehen. Weil das Signal x periodisch ist, lässt sich das Verhalten der Phase in den harmonischen Körben durch Regeln der DFT erklären. Die Phase im j-ten harmonischen Korb eines um $s \in \mathbb{Z}$ Abtastpunkte verschobenen Frames ergibt sich als

$$p_{l',m+s}(j) = p_{l',m}(j) + \frac{2\pi}{l'} 4 \cdot j \cdot s. \tag{4.32}$$

In Abbildung 4.22(c) ist zu sehen, wie sich die Phasendifferenzen verhalten:

$$pd_{l',m+s}(j) = (p_{l',m}(j+1) - p_{l',m}(j)) + \frac{2\pi s}{P}\ ,\ j \in \{1, 2, ..., H_{l'} - 1\} \tag{4.33}$$

Die Phasendifferenzen rotieren in höheren harmonischen Körben nicht schneller, sondern sind $\frac{P}{f_s}$-periodisch.

Es ist für $m, s \in \mathbb{Z}$:

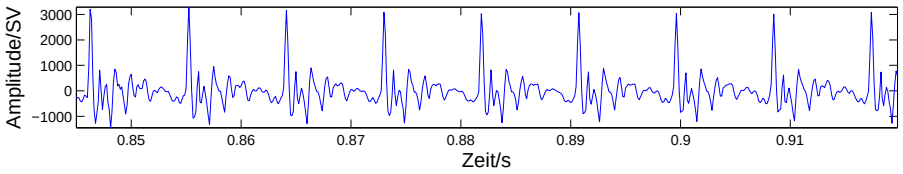
$$I'_{l'}(m+s) := \angle(\sum_{j=1}^{H_{l'}-1} e^{i \cdot pd_{l',m+s}(j)}) \tag{4.34}$$

$$= \angle(\sum_{j=1}^{H_{l'}-1} e^{i \cdot (p_{l',m}(j+1) - p_{l',m}(j) + \frac{2\pi s}{P})} \tag{4.35}$$

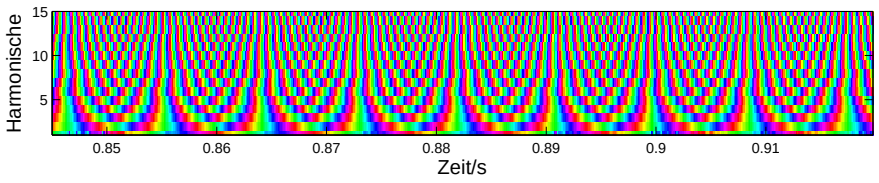
$$= \angle(e^{-i \cdot \frac{2\pi s}{P}} \sum_{j=1}^{H_{l'}-1} e^{i \cdot pd_{l',m}(j)}) \tag{4.36}$$

$$= I'_{l'}(m) + \frac{2\pi s}{P} \tag{4.37}$$

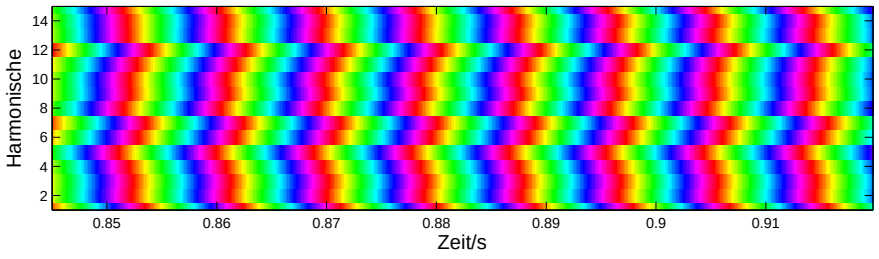
[3] Alle Winkel/Phasen liegen im Intervall $[-\pi; \pi]$. Alle Werte, die aus Operationen mit zwei Winkeln resultieren, werden wieder in dieses Intervall gelegt: $\phi = mod(\phi' + \pi, 2 \cdot \pi) - \pi$.



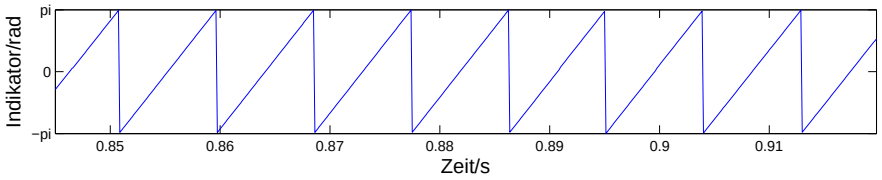
(a) Signal



(b) Phase



(c) Phasendifferenzen



(d) Indikator

Abbildung 4.23.: Betrachtung für ein Sprachsignal.

Die Steigung des *Indikators* $I'_{l'}$ beträgt $\frac{2\pi}{P}$. Ist beispielsweise bekannt, dass $I'_{l'}(m) = -\frac{\pi}{2}$ ist, so ist $I'_{l'}(m + \frac{1}{4}P) = 0$. Sind ein Abtastpunkt m, eine Zykluslänge P, sowie ein Wert $I'_{l'}(m)$ gegeben, kann für Abtastpunkte m' der Wert des Indikators bestimmt werden.

Für einen *Synchronisationswert* $p \in [-\pi; \pi]$ *zeigt* der Indikator daher auf Abtastpunkte m, zu denen der Indikator den Wert p annimmt.

Der Indikator zeigt im Allgemeinen auch zwischen Abtastpunkte.

Allgemeiner Fall

In Abbildung 4.23 sind die Berechnungen für ein Sprachsignal aus der Aurora-2 Datenbank dargestellt [PgHG00]. Die Berechnung des Indikators I' wurde hierzu

abgeändert. Der Indikator für einen Frame $x_{l,m}$ ist

$$I_l(m) := \angle(\sum_{j=1}^{H_l-1} md_{l,m}(j)e^{i \cdot pd_{l,m}(j)}) \ , \ j \in \{1, 2, ..., H_{l'} - 1\}. \tag{4.38}$$

Die Phasendifferenzen werden gemeinsam mit einer Magnitude $md_{l,m}(j) \in \mathbb{R}$ als komplexe Zahlen aufgefasst. Die Gewichtung mit der Magnitude $md_{l,m}(j) \in \mathbb{R}$ soll die Phasendifferenzen gewichten, für die ein kleiner Fehler vermutet wird.

Fehler in den Phasendifferenzen treten durch Überlagerung mit Rauschen und für Fensterlängen l auf, die nicht grundfrequenz-skaliert sind:

- Wie in Abschnitt 4.4 dargestellt wurde, sind die Fehler durch Rauschen größer, wenn die Magnitude des Rauschens verhältnismäßig groß ist.
- Ähnlich wie die Magnitude der harmonischen Körbe in Abschnitt 4.5.2 bei Abweichungen von einer grundfrequenz-skalierten Fensterlänge schwächer wird, treten in den Phasendifferenzen bei Abweichungen von der grundfrequenz-skalierten Fensterlänge Fehler auf (siehe Abbildung 2.1). Diese Abweichungen sind für niedrige Harmonische bei gleicher Abweichung von der grundfrequenz-skalierten Fensterlänge geringer.

Um beide Einflüsse zu berücksichtigen, werden die Magnituden $md_{l,m}(j)$ für eine Abtastdistanz Δl als Mittelwert der Summanden j und $j + 1$ aus Gleichung 4.23 gewählt:

- In der Phasendifferenz $pd_{l,m}(j)$ sind die Phasen der harmonischen Körbe j und $j+$ enthalten.
- Die Energie in hohen Harmonischen aus Gleichung 4.23 ist generell niedrig, weil keine Präemphase angewandt wird. Die Energie wird bei einem großem Synchronitätsradius für hohe Harmonische zusätzlich gedämpft, weil dann größere Abweichungen von der grundfrequenz-skalierten Fensterlänge auftreten.
- $md_{l,m}(j)$ ist aufgrund des Spectral Subtraction klein, wenn Rauschen vorliegt.

Ein großes Gewicht erhalten automatisch die Harmonischen, welche in den Bereich von Formanten fallen. Da die Formanten ihren Ursprung in der Filterung durch den Vokaltrakt und nicht im Quellsignal haben, wirkt sich dies auf den Indikator aus, weil die Formanten im Phasengang kodiert sind (siehe Abschnitt 3.3 und Kapitel 6).

Es wird angenommen, dass der Indikator I zum Verschlussmoment Null beträgt. In Kapitel 6 wird gezeigt, dass diese Annahme für stimmhafte annäherungsweise zutrifft.

Auch DYPSA verwendet einen energie-gewichteten Group Delay und weicht von der Annahme ab, dass ein Minimalphasensignal vorliegt. Dort wird angenommen, dass ein Verschlussmoment ein Center of Gravity in einem Signalausschnitt ist, der weniger als einen Phonationszyklus erfasst (siehe Abschnitt 3.1.2.3). Abgrenzend werden hier rauschgefilterte Gewichte verwendet, und nur die harmonischen Körbe berücksichtigt.

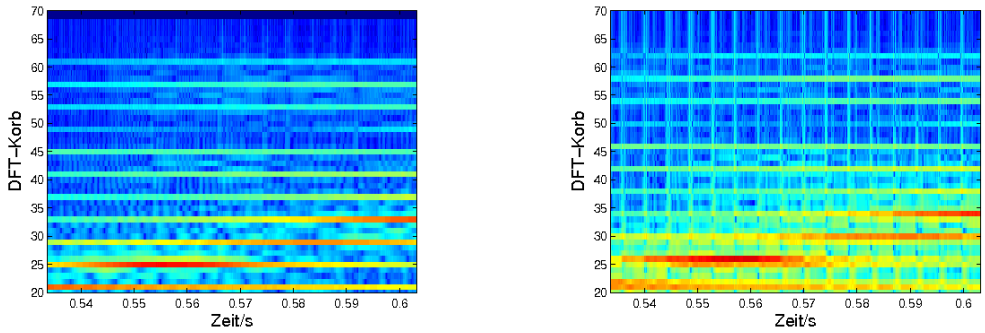


Abbildung 4.24.: Glottissynchrone Verarbeitung. Links ist die Magnitude von grundfrequenz-skalierten DFTen $X_{l,m}$ zu sehen. Rechts wurden Fensterlängen verwendet, die von der grundfrequenz-skalierten Fensterlänge abweichen. An den vertikalen Linien sind die Verschlussmomente zu erkennen.

Bemerkungen

Unabhängig davon, bei welchem Synchronisationswert der Verschlussmoment vermutet wird, ist der Indikator I_l periodisch mit der Dauer des Phonationszyklus, wenn ein periodisches Signal untersucht wird und l für m grundfrequenz-skaliert ist.

Die Gewichtung mit der oktavengefilterten Magnitude aus Gleichung 4.23 verhindert, dass die Winkel der komplexen Zahlen in Gleichung 4.38 sich um 180 Grad drehen, weil die Magnitude in einem harmonischen Korb nach Anwendung des Teilerfilters nicht kleiner als Null sein kann. Ebenso werden nur Harmonische berücksichtigt, die weit aus dem Rauschen herausragen. Das sind gegebenfalls nur die Formanten, was die Berechnung ungenauer aber robuster macht. Der Vielfachfilter wird für einen kleinen Synchronitätsradius ausgeschaltet, weshalb dieser die Berechnung dann nicht mehr beeinflusst. Die Wahl der Gewichte der Phasendifferenzen ist in gewisser Weise willkürlich. Es können andere Möglichkeiten untersucht werden. Damit trotz der willkürlichen Wahl der Gewichte der Indikator zum Verschlussmoment Null beträgt, müssen im Idealfall alle Phasendifferenzen zum Verschlussmoment Null betragen.

4.6.2. Glottissynchrone Verarbeitung

In dieser Arbeit wird neben der grundfrequenz-skalierten Verarbeitung auch glottissynchrone Verarbeitung durchgeführt. Das heißt, dass die Analysefenster beispielsweise immer zum Verschlussmoment angelegt werden (Synchronisationswert gleich Null) oder genau zwischen zwei Verschlussmomenten (Synchronisationswert gleich π). In Abbildung 4.24 kann beobachtet werden, wie sich die Magnitude in den Körben einer grundfrequenz-skalierten DFT verhält. Ist das Signal periodisch und die Fensterlängen sind grundfrequenz-skaliert, so bleibt die Magnitude in den harmonischen Körben über die Zeit konstant. Sind die Fensterlängen nicht genau an die Grundfrequenz angepasst, treten insbesondere zum Verschlussmoment Unstetigkeiten auf, weil dort besonders große Schwankungen in der Amplitude des Sprachsignals auftre-

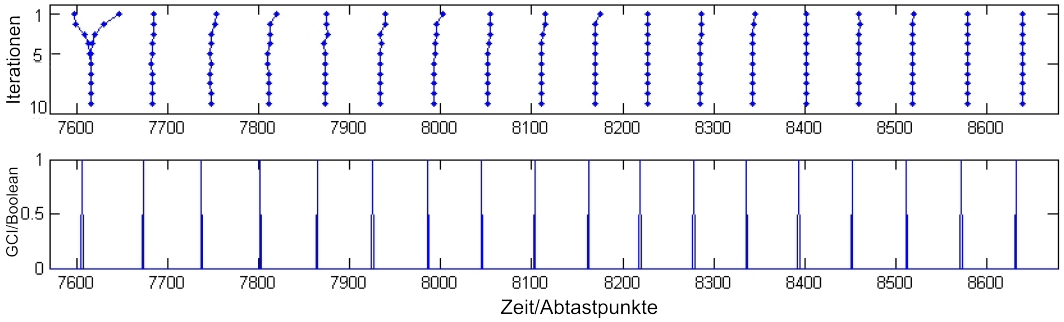


Abbildung 4.25.: Iterationsketten für die Verschlussmomente. Oben wurden Iterationspunkte einer Iterationskette mit Linien verbunden. Die Iterationspunkte sind entsprechend ihrer Verschlussmomentschätzung platziert. Unten sind zum Vergleich die Verschlussmomente zu sehen, wie DYPSA sie festlegte.

ten und so größere Sprungstellen an den Rändern entstehen. Dies tritt verstärkt auf, wenn rechteckige Fensterfunktionen verwendet werden, wie das in Abbildung 4.24 der Fall ist. Daher wird in dieser Arbeit versucht, die Analysefenster glottissynchron zu platzieren, wobei die Analysefenster mittig zwischen zwei Verschlussmomente gelegt werden, damit die Verschlussmomente nicht am Rand der Analysefenster liegen. Um die Verschlussmomente zu erkennen, ist das glottissynchrone Vorgehen sinnvoll, weil Fehler der Grundfrequenzschätzung sich in Voraussagefehlern des Indikators niederschlagen. Ein weiteres Ziel ist es, die Eigenschaften der Phase in den harmonischen Körben glottissynchroner Frames bei stimmhafter Sprache zu untersuchen.

4.6.3. Iterative Verarbeitung

Im Folgenden wird erläutert, wie Iterationsketten für die Verschlussmomentschätzung erweitert werden.

4.6.3.1. Erweiterung der Iterationskette i

In Abbildung 4.25 werden Iterationsketten für die Verschlussmomentschätzung visualisiert. Wie in Abbildung 4.21 dargestellt wurde, ist jedem Iterationspunkt (i, b) eine Schätzung über die Grundfrequenz $l(i, b) \in L$ und eine Position $t_{i,b} \in \mathbb{Z}$ zugeordnet. Die Position $t_{i,b}$ ist der Abtastpunkt, an welchem die Frames $x_{l,t_{i,b}}$ des Iterationspunktes (i, b) zentriert werden. Der oben beschriebene Indikator verweist dann im Zusammenhang mit der Grundfrequenzschätzung auf Zeitpunkte, zu denen der Indikator beispielsweise den Synchronisationswert $p = 0$ oder $p = \frac{\pi}{2}$ vermutet.

Für einen Abtastpunkt $t \in \mathbb{Z}$ gibt $v_{i,b}(t)$ die Distanz zur nächsten Verschlussmomentschätzung des Iterationspunktes (i, b) an:

$$v_{i,b}(t) := \min_{k \in \mathbb{Z}} \{t - [t_{i,b} - 0.5 \cdot \frac{(I_{l(i,b)}(t_{i,b}) + p)}{\pi} \frac{l(i,b)}{4} + k \cdot l(i,b)]\} \tag{4.39}$$

Um die Position $t_{i,b} \in \mathbb{Z}$ für den Iterationspunkt (i, b) zu berechnen, werden die

Verschlussmomentschätzungen der Iterationspunkte $(i-1,b)$, $(i,b-1)$ und $(i+1,b-2)$, die der Position $t_{i,b-1}$ am Nächsten liegen, zusammengeführt.

$$t'_{i,b} = t'_{i,b-1} + \frac{1}{2}v_{i,b-1}(t_{i,b-1}) + \frac{1}{4}[v_{i-1,b}(t_{i,b-1}) + v_{i+1,b-2}(t_{i,b-1})] \in \mathbb{R} \tag{4.40}$$

Es werden die direkt benachbarten Iterationsketten als Nachbarschaft berücksichtigt, weil stimmhafte Sprache über kurze Zeiträume näherungsweise periodisch ist. Die Position $t'_{i,b-1}$ geht in die Berechnung ein, damit die Schätzungen früherer Iterationsschritte Gewicht bekommen. In höheren Iterationsebenen erhalten daher auch weiter entfernte Iterationsketten Gewicht. In Abbildung 4.25 fällt auf, dass bei dem verwendeten Verfahren zwei Iterationsketten auf denselben Verschlussmoment hinauslaufen können.

Um eine höhere Robustheit zu erreichen, werden die Schätzungen $v_{i,b}$ mit der Konfidenz des Iterationspunktes (i,b) gewichtet. Gab es keine konfidenten Messungen, wird die Position $t_{i,b}$ aus dem Iterationsschritt $(i,b-1)$ übernommen.

Die Position $t_{i,b} \in \mathbb{Z}$ ergibt sich durch Rundung von $t'_{i,b} \in \mathbb{R}$ zur nächsten ganzen Zahl.

Um den Anforderungen an einen Trichterschritt nachzukommen, werden überholende Iterationsketten zurückgesetzt (siehe Abschnitt 4.2.2): $t_{i,b} = max\{t_{i,b}, t_{i-1,b} + 1\}$. Iterationspunkte einer Iterationskette dürfen nicht zu weit auseinander liegen:

$$t_{i,b} = \begin{cases} max\{t_{i,b}, t_{i,1} - 200\}, \text{ für } t_{i,b} \leq t_{i,1} \\ min\{t_{i,b}, t_{i,1} + 200\}, \text{ für } t_{i,b} > t_{i,1} \end{cases} \tag{4.41}$$

Bemerkungen

Die variablen Frameabstände können bei der Grundfrequenzverfolgung in der Kostenfunktion, dem Energielevel-Update und dem Update der Parameter c_2 und c_3 berücksichtigt werden (siehe Abschnitt 4.5).

4.6.3.2. Iterationsanfang

Die Berechnung der Position $t_{i,1}$ erfolgt mithilfe der Verschlussmomentschätzungen der Iterationspunkte $(i-1,1)$ und $(i-2,1)$ die der Position $\hat{t}(i,1) = t_{i-1,1} + \frac{l(i-1,1)}{4} \in \mathbb{R}$ am Nächsten liegen.

Für den Iterationsanfang werden Schätzungen zurückgesetzt, die zu weit in die Zukunft reichen:

$$t_{i+1,1} = min\{t_{i+1,1}, t_{i,1} + 100\} \tag{4.42}$$

Das geschieht, damit die Anforderungen an den Algorithmus bezüglich der Vorausschau eingehalten werden (siehe Abschnitt 4.2.2).

Für $B = 10$ ergibt sich eine maximale Vorausschau von $V_{it} = 10 \cdot 100 + 2 \cdot 200 + \lceil L_{max} \rceil = 1600$ (siehe Gleichung 4.1). Das entspricht einer Vorausschau von 200 ms. Die maximale Schrittweite von 100 Abtastpunkten beim Iterationsanfang entspricht der (angenommenen) maximalen Länge eines Phonationszyklus.

4.6.4. Experiment

Durch die Synchronisation mit den Verschlussmomenten ist die Phase in den harmonischen Körben benachbarter glottissynchroner Frames annäherungsweise gleich. Das liegt darin begründet, dass sich benachbarte glottissynchrone Frames für ein periodisches Signal gleichen.

In Abbildung 4.26(a) kann dies beobachtet werden. Ähnlich wie beim Magnitude Averaging besteht die Hoffnung, dass die Stetigkeit der Phase genutzt werden kann, um Spectral Subtraction für stimmhafte Sprache auch auf die Phase zu erweitern, und nicht nur die Magnitude zu verwenden.

In Abbildung 4.26(b) wurden die komplexen Zahlen in den harmonischen Körben geglättet. Dabei wurde ausgenutzt, dass die komplexen Zahlen in benachbarten glottissynchronen Frames im Idealfall die gleiche Magnitude und die gleiche Phase aufweisen. Die komplexen Zahlen wurden gewichtet aufaddiert.

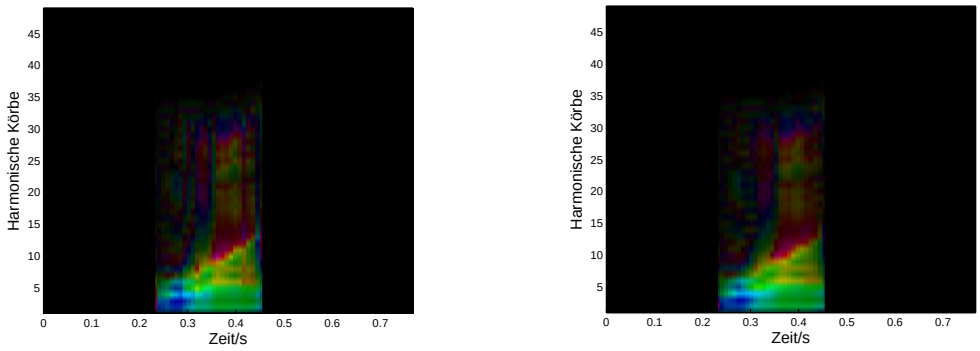
Glottissynchrone Verarbeitung kann auch erreicht werden, indem die Abstände zwischen Frames an die geschätzte Grundfrequenz angepasst werden. Dann ist die Phase benachbarter Frames ebenfalls gleich. Allerdings ist dann die Lage der Frames unbekannt, was eine weitere Information darstellt. Für das idealisierte Anregungssignal in Abbildung 4.22 ist beispielsweise die Phase in allen harmonischen Körben bekannt, wenn die die Lage eines Frames bekannt ist. Im Idealfall ist also nicht nur die Stetigkeit der Phase zwischen benachbarten glottissynchronen Frames gegeben, sondern auch der Winkel der harmonischen Koeffizienten ist gegeben. Das ist eine zusätzliche Information. Es kann versucht werden, diese beim Spectral Subtraction für die robuste Spracherkennung zu berücksichtigen. Wie in Abbildung 4.23(b) zu sehen ist, trifft die Idealvorstellung für stimmhafte Sprache nicht exakt zu, weil sich zwei Phonationszyklen stimmhafter Sprache nie gleichen.

Bemerkungen

Ähnlich wie bei der Aggregation der harmonischen Körbe für die Berechnung der harmonischen Oberfläche die Information über die Stetigkeit der Magnitude benachbarter Frames in gewisser Weise verloren geht, werden bei der Aggregation der Phasendifferenzen zur Bildung des Indikators Informationen über die Stetigkeit der Phase vernachlässigt. Es kann untersucht werden, ob die Stetigkeit ausgenutzt werden kann, indem für die Berechnung des Indikators die komplexen Zahlen in den harmonischen Körben mithilfe der benachbarten Frames wie in Abbildung 4.26(b) geglättet werden.

4.7. Merkmalsextraktion

Für die Merkmalsextraktion wird Behnkes Harmonic Frontend übernommen [Beh]. Um Rauschen aus einem Sprachsignal zu entfernen, kombiniert Behnke zwei Verarbeitungswege: Ein harmonischer Verarbeitungsweg filtert das Spektrum eines Frames für stimmhafte Sprache (siehe Abschnitt 3.2). Dazu wird die Schätzung der Grundfrequenz benötigt. Der nicht-harmonische Verarbeitungsweg filtert das Spektrum



(a) Glottissynchrone Phase (b) Glottissynchrone geglätte Phase

Abbildung 4.26.: Glottissynchrone Phase. Links wurden die komplexen Zahlen in den harmonischen Körben glottissynchroner Frames abgebildet. Die Phase wird durch Farbe kodiert, die Magnitude wird durch Helligkeit kodiert. Rechts wurde eine Glättung der komplexen Zahlen vorgenommen.

für den stimmlosen Teil des Signals. Zu jedem Abtastpunkt werden die gefilterten Spektren beider Verarbeitungswege berechnet. Beide Spektren werden kombiniert, so dass bei stimmhafter Sprache der harmonische Verarbeitungsweg mehr Gewicht erhält.

Aus dem kombinierten, gefilterten Spektrum werden dann Merkmalsvektoren mit einem üblichen Verfahren abgeleitet. Es werden MFCC's verwendet.

Der nicht-harmonische Verarbeitungsweg verwendet einen Bandpass-Filter (ähnlich dem RASTA-Filter) und Spectral Subtraction. Der Bandpass Filter kann mit einer Vorausschau von 150 ms berechnet werden. Für $B = 10$ Iterationsebenen wird dann eine Vorausschau von

$$V_g = V_{it} + V_{filt} = 1600 + 1200 = 2800. \tag{4.43}$$

Abtastpunkten benötigt. Das entspricht 350 ms.

Für das Spectral Subtraction wird der Erwartungswert des Rauschens über das gesamte Signal gemittelt. Diese Rauschschätzung kann beispielsweise durch eine adaptive Rauschschätzung ersetzt werden, ähnlich wie sie im ETSI Standard ES 202 050 V1.1.5 vorgeschlagen wird [Eur07].

Im Anschluss wird eine einfache Energienormalisierung vorgenommen, die Wissen über das gesamte Sprachsignal voraussetzt. Auch hier existieren Ansätze, eine adaptive Normalisierung vorzunehmen [JJPM08].

4.7.1. Glottissynchrone Verarbeitung

Bei der glottissynchronen Verarbeitung muss berücksichtigt werden, dass die Frameabstände variabel sind: Die Frames, welche in der letzten Iterationssebene analysiert werden, sind synchron zu den Verschlussmomenten. Das hat zur Folge, dass der Bandpass-Filter auf einer verzerrten Eingabe arbeitet. Einige Verschlussmomente,

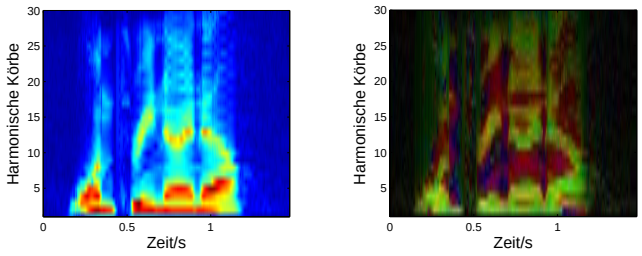


Abbildung 4.27.: Phasendifferenzen. Links ist die Magnitude in den harmonischen Körben glottissynchoner DFTen $X_{l,m}$ dargestellt. Rechts werden komplexe Zahlen in den harmonischen Körben visualisiert: Die Phasendifferenzen bilden den Winkel der komplexen Zahl und werden durch den Farbton im HSV-Farbraum kodiert. Die Magnitude wird durch Helligkeit kodiert.

besonders zu Beginn stimmhafter Sprache, werden von zwei Iterationsketten angesteuert. Außerdem werden für Äußerungen von Sprechern mit tiefer Grundfrequenz weniger Frames pro Sekunde analysiert als für Personen mit hoher Grundfrequenz, obwohl die Grundfrequenz in der Regel nicht mit der Geschwindigkeit der Vokaltraktbewegung korreliert ist. Um diese Probleme zu umgehen, werden die glottissynchronen Spektren interpoliert, so dass Spektren mit festen Zeitabständen erhalten werden[4].

4.7.2. Experiment

In diesem Abschnitt wird ein Experiment vorgestellt, das versucht, die Verwandschaft von Magnitudenspektrum und Phasenspektrum für das harmonische Spektrum stimmhafter Sprache zu zeigen. Dazu werden glottissynchrone Frames x_{l^*,m^*} verwendet[5]. In Abbildung 4.27 ist links die Magnitude der harmonischen Körbe der DFTen X_{l^*,m^*} dargestellt. Die Darstellung in der Mitte ergibt sich aus

$$C_{l^*,m^*}(j) := \hat{M}_{l^*,m^*}(h_j)e^{i \cdot pd_{l^*,m^*}(j)+pd_{l^*,m^*}(j+1)}, j \in \{2, ..., H_{l^*} - 1\}. \tag{4.44}$$

Es werden dazu die gefilterten Magnituden $\hat{M}$ aus Gleichung 4.10 und die Phasendiferenzen aus Gleichung 4.31 verwendet. In Abbildung 4.27 kann die Ähnlichkeit der Darstellung aus Gleichung 4.44 und des Magnitudenspektrums beobachtet werden. Die Stetigkeit von Magnitude und Phase in Zeit- und Frequenzrichtung ist durch die glottissynchrone Verarbeitung gegeben. Die Darstellung kann allerdings nicht direkt für die Spracherkennung verwendet werden, weil keine Backends existieren, die komplexe Zahlen als Eingabe akzeptieren. Die Phase direkt für die Spracherkennung zu verwenden kann versucht werden, aber phase-unwrapping ist dann ein Problem. Daher kann auch überlegt werden, die Verfahren aus Abschnitt 3.3 zur Verwendung

[4] Auch der Mensch synchronisiert sich aufgrund des Phase-Locking mit dem Anregungssignal, dennoch bleibt dort die Information über die Zeit erhalten.

[5] Glottissynchrone Frames können auch aus grundfrequenz-skalierten Frames durch Rotation im Frequenzraum berechnet werden.

des Phasengangs für die Merkmalsextraktion auf das harmonische Spektrum der glottissynchronen Frames anzuwenden.

Kapitel 5.

Realisierung

In diesem Kapitel werden Details zur Realisierung des Algorithmus erläutert. Das Programm MATLAB[1] wurde zur Realisierung und Entwicklung der Algorithmen verwendet.

5.1. Grafische Oberfläche

Eine grafische Oberfläche wurde programmiert, um das Verhalten des Algorithmus zu visualisieren (siehe Abbildung 5.1).

5.2. Laufzeit

In Kapitel 4 wurde ein Algorithmus entworfen, welcher nach oben beschränkte Zeit für jeden Trichterschritt benötigt. In jedem Trichterschritt wird für die B Iterationsebenen jeweils ein Iterationspunkt berechnet. Für einen Iterationspunkt besteht der wesentliche Aufwand darin, die harmonische Oberfläche auszurechnen. Für jeden Iterationspunkt werden $C \cdot 2 + 1$ Punkte der Oberfläche berechnet (siehe Abschnitt 4.5.1). Um den Wert $E^O_{l,m}(1)$ zu berechnen, wird der Frame $x_{l,m}$ mit einem rechteckigen Fenster ausgeschnitten und seine DFT $X_{l,m}$ mit Hilfe einer FFT bestimmt. Die DFT $X^h_{l,m}$ des Frames $x^h_{l,m}$, welcher mit einem Hamming-Fenster ausgeschnitten wurde, wird benötigt, um $E^O_{l,m}(\Delta l_{max})$ zu berechnen. $X^h_{l,m}$ kann mithilfe von $X_{l,m}$ im Frequenzraum berechnet werden [Lyo04]:

$$X^h_{l,m}(k) = 0.54X_{l,m}(k) + 0.46 \cdot 0.5 \cdot (X_{l,m}(k-1) + X_{l,m}(k+1)) \tag{5.1}$$

Das entspricht der Verwendung der ersten l Einträge eines Hamming-Fensters der Länge $l + 1$ als Fensterfunktion (siehe Gleichung 2.13).

Die Berechnung einer zweiten FFT kann daher vermieden werden. Von den l Einträgen der DFT $X^h_{l,m}$ müssen für die Berechnungen des Algorithmus die $l/4$ Einträge der Körbe h_j, h_j^2 und h_j^{-2} ermittelt werden.

Um die Laufzeit der Implementierung zu ermitteln, wird die Zeit gemessen, die für einen Trichterschritt benötigt wird. In jedem Trichterschritt ist die Anzahl der Operationen durch eine Konstante nach oben begrenzt. Die Anzahl der Operationen pro Trichterschritt variiert jedoch, weil die Länge der Analysefenster von der

[1] http://www.mathworks.de/

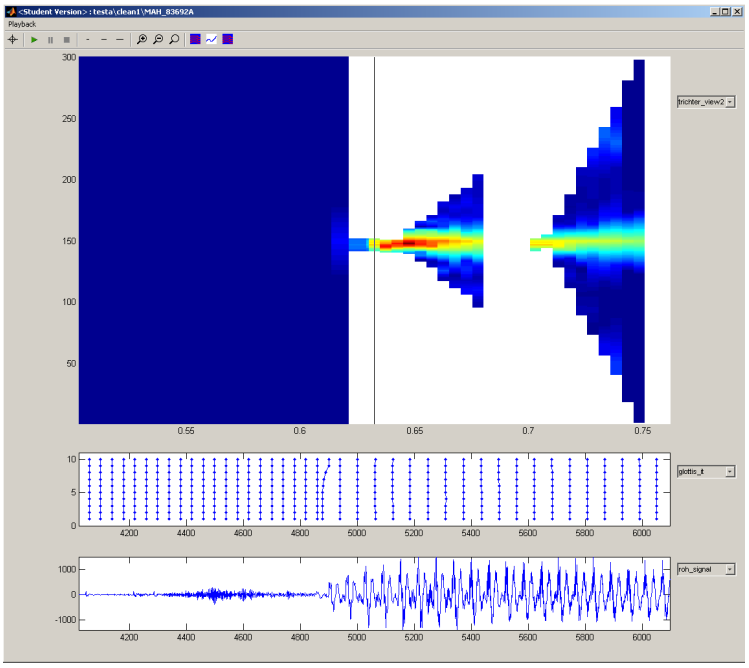


Abbildung 5.1.: Grafische Oberfläche. Unten können verschiedene Kurven ausgewählt und dargestellt werden. Oben befindet sich ein bildartiger Bereich, in dem beispielsweise ein Spektrogramm dargestellt werden kann. In diesem Beispiel ist dort die harmonische Oberfläche für verschiedene Iterationspunkte dargestellt. Der aktuelle Zeitpunkt wird durch die vertikale Linie markiert. Links davon ist die harmonische Oberfläche dargestellt, welche auf der abschließenden Iterationsebene berechnet wurde. Rechts der Linie ist das Signal noch nicht vollständig bekannt und die Iterationsketten sind noch nicht abgeschlossen. Zwei trichterartige Strukturen sind zu sehen. Der rechte Trichter zeigt die harmonische Oberfläche der verschiedenen Iterationsebenen für den aktuellen Zeitpunkt. Der linke Trichter zeigt die harmonische Oberfläche der höchsten Iterationsebenen der aktiven Iterationsketten. Außerhalb der Vorausschau wurden noch keine Iterationsketten begonnen.

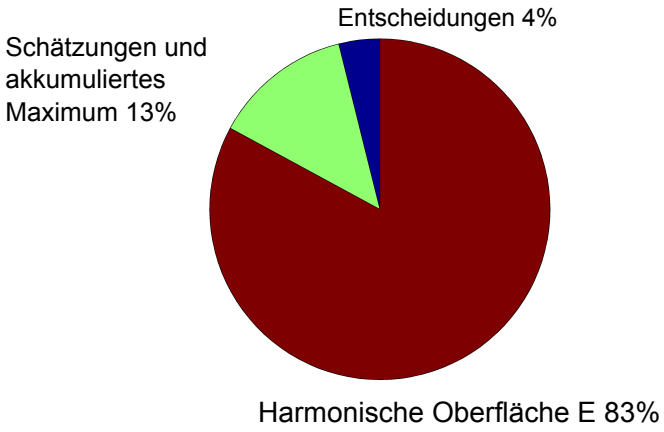


Abbildung 5.2.: Laufzeit für einen Trichterschritt. Der Großteil der Laufzeit wird für die Berechnung der harmonischen Oberfläche benötigt. Ein weiterer signifikanter Anteil fällt auf die Berechnung des akkumulierten Maximums und der Schätzungen für die Iterationspunkte. Die Entscheidungen, die im letzten Iterationsschritt gefällt werden, benötigen 4% der Laufzeit. Die Gesamtlaufzeit für einen Trichterschritt beträgt 29 ms.

Grundfrequenzschätzung abhängt. Da der Fokus der Arbeit nicht auf der Echtzeitfähigket liegt, sondern darauf, dass der Algorithmus eine begrenzte Vorausschau hat und robust ist, wird daher an dieser Stelle lediglich versucht, einen Eindruck über die Größenordnung der Laufzeit eines Trichterschritts zu erlangen, und nicht exakte obere Schranken zu ermitteln. Um die Zeit zu messen, die ein Trichterschritt benötigt, wird ein Signal eines Sprechers mit tiefer Grundfrequenz analysiert. Zur Messung wurde ein Kern eines Intel Core Duo E8400 mit 3 GHz auf einem Windows-Betriebssystem mit 4 GB Arbeitsspeicher verwendet. Zur Evaluation der Laufzeit werden $B = 10$ Iterationsebenen verwendet. Die Anzahl der Abtastpunkte für die harmonische Oberfläche beträgt hier $2 \cdot C + 1 = 11$. Ein Trichterschritt dauerte dann im Mittel 29 ms. Tatsächlich müsste die Worst-Case Laufzeit ermittelt werden. Andererseits garantiert das Betriebssystem, das für die Messungen verwendet wurde, keine Antwortzeiten und ist daher für verlässliche Messungen nicht geeignet. Für einen Trichterschritt wird für die B Iterationsebenen jeweils ein Iterationspunkt berechnet. Für einen Iterationspunkt werden die harmonische Oberfläche, das akkumulierte Maximum und die Schätzungen über die verschiedenen Größen berechnet (siehe Abbildung 5.2). Für den B-ten Iterationspunkt werden einige zusätzliche Operationen ausgeführt, weil dort Entscheidungen getroffen werden.

Fraglich ist noch, wieviele Trichterschritte pro 10 ms durchgeführt werden. Im schlimmsten Fall führt ein Trichterschritt nur einen Abtastpunkt weiter (siehe Abschnitt 4.2.2). Das bedeutet, dass pro 10 ms im schlimmsten Fall $(10 \cdot 8 \cdot 29)$ ms gerechnet wird. Das entspricht einem Faktor von ca. 240, den die Implementierung zu langsam ist.

Bemerkungen

Um einen kleineren Faktor zu erreichen, gibt es verschiedene Möglichkeiten. Es kann beispielsweise eine Mindestschrittweite von bis zu 3.125 ms beim Iterationsanfang gefordert werden (das entspricht dem kürzesten angenommenen Phonationszyklus). Ein weiterer Ansatzpunkt ist eine Parallelisierung: In einem Trichterschritt können die B Iterationspunkte unabhängig voneinander berechnet werden. Innerhalb eines Iterationspunktes kann die harmonische Oberfläche für die $2 \cdot C + 1$ Abtastpunkte parallel berechnet werden. Insgesamt ist die Implementierung in MATLAB nicht laufzeitoptimiert und kann effizienter gestaltet werden.

5.3. Speicher

Der Algorithmus benötigt begrenzt viel Speicher. Bei der Implementierung des Algorithmus wurden dennoch die gesamten berechneten Oberflächen und die Werte aller Iterationsketten gespeichert, damit diese visualisiert werden können (siehe Abschnitt 5.1).

Kapitel 6.

Evaluation

In diesem Kapitel werden die Genauigkeit und Robustheit der entwickelten Algorithmen evaluiert. Dazu wird die Aurora-2 Datenbank verwendet [PgHG00]. Die Sprachsignale der Aurora-2 Datenbank liegen jeweils in klarer Sprache und für Signal-Rausch-Verhältnissen von $-5\,\mathrm{dB}$ bis $20\,\mathrm{dB}$ für mehrere Rauscharten vor. Die Sprachsignale enthalten Sequenzen von bis zu sieben Ziffern, die von US-Amerikanischen Erwachsenen gesprochen wurden. Die Abtastrate beträgt $8000\,\mathrm{Hz}$.

In Abschnitt 6.1 werden wichtige Paramter des Algorithmus zusammengefasst. Diese werden dann bei der Evaluation variiert. In den Abschnitten 6.2 und 6.3 wird die Schätzung der Grundfrequenz und der Verschlussmomente evaluiert. Da der Aurora-2 Datenbank keine Annotation des Quellsignals beiliegt, werden zur Evaluierung Referenz-Algorithmen herangezogen. Die Referenz-Algorithmen annotieren dazu ein klares Sprachsignal. Da auch die Referenz-Algorithmen nicht optimal sind, kann die Genauigkeit der Algorithmen nicht exakt bewertet werden. Der Fokus liegt auf dem Verhalten bei verrauschten Daten. Es wird getestet, wie stark die Leistung bei verrauschten Daten nachlässt. Die Eigenschaften der Algorithmen zur Schätzung des Quellsignals werden anhand von Subset 1 des Test Sets A der Aurora-2 Datenbank getestet. Die dort verwendete Rauschart heißt „Subway“.

In Abschnitt 6.4 wird evaluiert, welche Ergebnisse für die Aurora-2 Datenbank mit verschiedenen Parametern erreicht werden. Die Evaluation des robusten Frontends findet auf dem Test Set A und dem Test Set B der Aurora-2 Datenbank für Multi-Condition Training mit verschiedenen Rauscharten statt.

Schließlich werden in Abschnitt 6.5 Eigenschaften des Algorithmus visuell evaluiert.

6.1. Parameter

Es gibt viele Parameter, die im entwickelten Algorithmus verstellt werden können. Einige wichtige werden hier zusammengefasst.

Grundfrequenz

- Die Konstante $C \in \{5, ..., 14\}$ legt fest, wie viele Werte der harmonischen Oberfläche ein Iterationspunkt berechnet. Das sind $2 \cdot C + 1$ Werte (siehe Abschnitt 4.5.1).

- Die Konstante $B > 3$ legt die Anzahl der Iterationsebenen fest (siehe Abschnitt 4.2.2).
- Der Parameter $c \in [0;1]$ bestimmt, wie stark der Synchronitätsgrad einer Iterationsebene $b > 1$ bei konfidenten Messungen zunimmt (siehe Abschnitt 4.26).
- Der Parameter $c_2 \in [0;1]$ steuert, wie stark der Synchronitätsgrad in der ersten Iterationsebene bei konfidenten Messungen zunimmt (siehe Gleichung 4.29).
- Der Parameter $u > 0$ steuert, wie stark die hohen Harmonischen gedämpft werden (siehe Gleichung 4.18).

Als Anhaltspunkt für die Wahl der Parameter c und c_2 wird $k_S := \frac{1}{S-1}$ definiert. Beispielsweise steigt der Synchronitätsgrad für c=k_3 nach 2 konfidenten Messungen von Null auf eins (vergleiche Abschnitt 4.5.3.1).

Verschlussmomente

- Der Synchronisationswert p regelt, welche Lage die Frames bei der glottissynchronen Verarbeitung anstreben (siehe Abschnitt 4.6.1).

6.2. Grundfrequenz

Es gibt einen *Referenz-Algorithmus* und einen *Test-Algorithmus*, die ein Sprachsignal $x \in l^2(\mathbb{Z})$ annotieren. Abtastpunkten $m \in \{1, 2, ..., M\}$ wird zugeordnet, ob diese stimmhaft sind. Wenn m stimmhaft ist, wird m zudem eine Grundfrequenz zugeordnet (vergleiche Abschnitt 4.2.2). Die Abweichungen des Test-Algorithmus vom Referenz-Algorithmus werden dann durch Fehlerraten bewertet.

- Als *Unvoiced Error Rate* (UER) wird der Anteil der fälschlicherweise als stimmhaft erkannten Abtastpunkte bezeichnet.
- Als *Voiced Error Rate* (VER) wird der Anteil der fälschlicherweise als nicht stimmhaft klassifizierten Abtastpunkte bezeichnet.
- Als *Gross Pitch Error Rate* (GPER) wird der Anteil der korrekt als stimmhaft erkannten Abtastpunkte bezeichnet, bei denen die geschätzte Grundfrequenz des Test-Algorithmus um mehr als 20% von der Schätzung des Referenz-Algorithmus abweicht.
- Der *Root Mean Square Error* (RMSE) zwischen geschätzter Grundfrequenz von Test- und Referenz-Algorithmus in Hertz wird für Abtastpunkte berechnet, die der Test-Algorithmus korrekt als stimmhaft erkannte und bei denen kein Gross Pitch Error auftrat.

(a) Referenz-Algorithmus

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
CLEAN	0	0	0	0
20 dB	4,7	1,52	0,27	0,99
15 dB	6,32	1,82	0,22	1,15
10 dB	8,99	2,42	0,36	1,23
5 dB	13,97	2,61	1,31	1,54
0 dB	22,44	3,41	6,07	1,98
-5 dB	40,49	6,16	21,19	2,79

(b) Test-Algorithmus

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
CLEAN	12,96	1,52	1,24	1,77
20 dB	13,43	1,31	1,29	1,78
15 dB	14,08	1,18	1,26	1,84
10 dB	15,44	1,14	1,42	1,88
5 dB	18,62	1,20	1,84	2,16
0 dB	26,02	1,87	3,88	2,74
-5 dB	39,77	5,76	10,48	4,98

Tabelle 6.1.: Ergebnisse der Grundfrequenzschätzung für das Subset 1 von Test Set A der Aurora-2 Datenbank. Der Test-Algorithmus verwendet die Parameter $C = 9$, $B = 10$, $c = 1 \cdot k_{10}$, $c_2 = 0.6 \cdot k_{10}$, $u = 9$ und $p = \pi$.

Die Genauigkeit wird also nur für Abtastpunkte gemessen, zu denen der Test-Algorithmus korrekt stimmhaft ausgibt, denn nur dort wird die Genauigkeit von dem vorgestellten Algorithmus auch zur robusten Spracherkennung verwendet.

In Tabelle 6.1 sind Ergebnisse der Grundfrequenzerkennung zu sehen. Als Referenz-Algorithmus zur Evaluierung der Grundfrequenzschätzung und der Entscheidung über die Stimmhaftigkeit wird der Algorithmus von Behnke bei klarer Sprache verwendet (siehe Abschnitt 3.1.1.2) [Beh03]. Dieser verwendet keinen Oktavenfilter. Allerdings wird die harmonische Oberfläche vorher auf Grundfrequenzen untersucht, die über das gesamte Sprachsignal viel Energie enthalten. Diese Information wird Nutzenfunktion verwendet, was aufgrund der Eigenschaften der harmonischen Oberfläche $E(1)$ ebenfalls Oktavenfehler vermeidet.

Der in der vorliegenden Arbeit entwickelte Algorithmus wurde als Test-Algorithmus verwendet (siehe Tabelle 6.1 für die Parameterwahl).

Der Test-Algorithmus weist bei klarer Sprache eine hohe *Voiced Error Rate* auf und hat bezüglich des Referenz-Algorithmus einen RMSE von 1.77 Hz und eine Gross Pitch Error Rate von $1,24\,\%$. An dieser Stelle kann nicht exakt geklärt werden, welcher der Algorithmen zu welchem Abtastpunkt eine bessere Entscheidung trifft. Alle Fehler des Referenz-Algorithmus werden dem Test-Algorithmus zur Last gelegt.

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
Baseline	27,99	2,02	3,98	3,03
$c = 1 \cdot k_B$	29,81	1,91	3,13	3,06
$c = 2.5 \cdot k_B$	25,89	2,22	3,84	2,98
$c = 10 \cdot k_B$	24,53	2,97	4,53	3,59
$c_2 = 0.2 \cdot c_{10}$	31,47	1,5	3,04	3,04
$c_2 = 2 \cdot c_{10}$	25,15	3,39	11,52	3,77
Schrittweise	21,30	3,59	5,35	2,97
$B = 4$, $c = 1 \cdot c_4$	33,14	1,98	3,13	2,9
$B = 4$, $c = 2.5 \cdot c_4$	32,97	1,96	3,09	2,92
$B = 4$, $c = 4 \cdot c_4$	31,21	2,18	3,25	3,19
$B = 4$, $C = 9$, $c = 1 \cdot c_4$	27,99	1,91	3,46	2,62
Vielfachfilter	34,55	1,03	1,47	2,63
Teilerfilter	25,18	3,77	4,24	2,98
$\Delta l = 1$	35,94	2,36	5,48	3,76
$\Delta l = \Delta l_{max}$	26,51	3,22	4,17	3,97
$u = 6$	30,05	2,09	4,16	2,82
$u = 15$	26,83	1,93	3,72	2,89

Tabelle 6.2.: Einfluss der Parameter. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Der Baseline-Algorithmus verwendet die Parameter $C = 5$, $B = 10$, $c = 1.5 \cdot k_B$, $k_2 = 0.6 \cdot k_{10}$, $p = \pi$ und $u = 9$.

Die hohe Voiced Error Rate wird beim Test-Algorithmus in Kauf genommen, damit eine niedrige Unvoiced Error Rate erzielt werden kann. Das ist wichtig, weil bei Unvoiced Errors eine Synchronisation stattfindet, so dass ein tatsächlich einsetzendes Sprachsignal verpasst werden kann.

Im Vergleich zum Referenz-Algorithmus treten beim Test-Algorithmus bei schlechter werdendem Signal-Rausch-Verhältnis weniger Gross Pitch Errors auf. Das ist ein Hinweis darauf, dass die Oktavenfilter auch bei niedrigem Signal-Rausch-Verhältnis und bei begrenzter Vorausschau eine Synchronisation auf die richtige Oktave ermöglichen.

Die Voiced Error Rate nimmt beim Test-Algorithmus verhältnismäßig weniger stark zu, als beim Referenz-Algorithmus. Bei einem Signal-Rausch-Verhältnis von -5dB treten dann weniger Voiced Errors als beim Referenz-Algorithmus auf. Dabei leidet die Genauigkeit im Gegensatz zum Referenz-Algorithmus bei schlechter werdendem Signal-Rausch-Verhältnis stärker[1].

Im Folgenden wird gezeigt, wie einzelne Parameter das Verhalten des Algorithmus verändern. Dazu wurde ein Baseline-Algorithmus festgelegt (siehe Tabelle 6.2). Es wird dann beobachtet, wie sich das Verhalten des Algorithmus ändert, wenn jeweils

[1] Der Test-Algorithmus weist bei einer Signal-Rausch-Rate von 0 dB einen RMSE von 2.5 Hz im Vergleich zur eigenen Schätzung bei klarer Sprache auf.

ein Parameter variiert wird. Als Referenz-Algorithmus wird weiterhin der Algorithmus von Behnke verwendet (siehe oben).

Der Parameter c steuert, wie stark der Synchronitätsradius verringert wird, wenn konfidente Messungen vorliegen. Wird $c = 2.5 \cdot k_{10}$ groß gewählt, sinkt die Voiced Error Rate. Das deutet darauf hin, dass für große Werte für c die Wahrscheinlichkeit größer wird, dass der Algorithmus sich synchronisiert, die harmonische Oberfläche klarer sieht, und daher stimmhafte Sprache seltener verpasst. Andererseits steigt die Unvoiced Error Rate gegenüber kleinen Werten von c und die Genauigkeit wird geringer. Das ist ein Hinweis darauf, dass dann häufiger eine falsche Synchronisation stattfindet.

c_2 regelt, wie stark sich der Synchronitätsradius in der ersten Iterationsebene verändert. Für $c_2 = 2 \cdot c_{10}$ legt der Algorithmus sich sehr schnell auf eine Grundfrequenz fest. Es treten daher weniger Voiced Errors auf, aber die Genauigkeit lässt stark nach und es geschehen häufig Oktavenfehler. Hier kann auch beobachtet werden, dass die Oktavenfilter nur die Wahrscheinlichkeit verringern, dass Oktavenfehler auftreten. Legt man sich zu Iterationsbeginn nicht zu schnell auf eine Oktave fest, können Oktavenfehler bei begrenzter Vorausschau daher besser vermieden werden.

In der Zeile „Schrittweise“ wurde die schrittweise Filterung des Signals aus Abschnitt 4.5.4 evaluiert. Diese konnte im Rahmen der vorliegenden Arbeit nicht ausführlich genug untersucht und optimiert werden, als dass ein Gewinn entstanden wäre.

Die Variation der Anzahl der Iterationsebenen wurde in den nächsten Zeilen getestet. Es fällt auf, dass bei weniger Iterationsebenen und der gleichen Konstanten C ein größerer Voiced Error entsteht. Das lässt darauf schließen, dass es gelungen ist, die Abtastfehler soweit zu minimieren, dass die Iterationsebenen folgendenen Iterationsebenen einen Hinweis über die Grundfrequenz geben können. Bei stärkerem Zuziehen durch die Wahl eines großen Parameters c wird eine verhältnismäßig geringere Senkung des Voiced Errors erreicht.

Für $B = 4$ und $C = 9$ ist das Ergebnis besser als beim Baseline-Algorithmus. Zum Abtasten der Oberfläche verwendete eine Iterationskette $4 \cdot (2 \cdot 9 + 1) = 76$ Abtastpunkte. Das sind weniger als die $10 \cdot (2 \cdot 5 + 1) = 110$ Abtastpunkte, die der Baseline-Algorithmus verwendete. Das ist ein Hinweis darauf, dass bei der iterativen Verarbeitung kein wesentlicher Gewinn für die iterative Verbesserung von Zwischenergebnissen erzielt wurde. Zwei Iterationsebenen werden allerdings mindestens benötigt, damit in der ersten Iterationsebene nicht die gesamte Oberfläche berechnet werden muss und die glottissynchrone Verarbeitung ermöglicht wird, ohne die harmonische Oberfläche zu jedem Abtastpunkt zu berechnen.

In der Zeile „Vielfachfilter“ blieb der Vielfachfilter unabhängig vom Synchronitätsradius angeschaltet. Dadurch entstand ein größerer Voiced Error, weil mehr harmonische Energie gefiltert wurde. Gleichzeitig traten weniger Gross Pitch Errors auf und die Genauigkeit stieg. Das kann zum einen damit zusammen hängen, dass weniger stimmhafte Sprache erkannt wurde oder zum anderen damit, dass der Vielfachfilter auch bei kleinem Synchronitätsradius half, Nebenmaxima zu umgehen.

In der Zeile „Teilerfilter“ wurde der Teilerfilter bei einem Synchronitätsradius von kleiner als 100 ausgeschaltet. Der Teilerfilter entfernt zum einen Energie, zum ande-

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
Baseline	30,91	1,07	2,52	2,61
$\alpha = 0,0$	23,03	3,72	5,79	2,96
$u = 15$	35,04	1,15	2,21	2,58
$u = 4$	34,02	1,27	4,45	2,76
$\Delta l = 1$	72,99	0,83	2,48	2,9

Tabelle 6.3.: Alternativer Teilerfilter. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Der Baseline-Algorithmus verwendet die Parameter $C = 5$, $B = 10$, $c = 1.5 \cdot c_B$, $c_2 = 0.6 \cdot c_{10}$, $u = 9$, $p = \pi$ und $\alpha = 0.07$. Der Parameter α steuert, wie stark der alternative Teilerfilter T' eingeschaltet wird (siehe Gleichung 4.16).

ren verhindert er Auslöschungen. Der geringe Voiced Error deutet darauf hin, dass der Teilerfilter mehr Energie entfernt, als das Auslöschungen verhindert werden. Allerdings stieg der Gross Pitch Error, das kann zum einen damit zusammenhängen, dass der Voiced Error sank. Zum anderen ist es möglich, dass der Teilerfilter auch noch bei einem Synchronitätsradius von kleiner als 100 Gross Pitch Errors verhindert.

Zusammenfassend war es ein experimentell ermittelter Kompromiß, den Vielfachfilter für einen Synchronitätsradius kleiner als 100 auszuschalten und den Teilerfilter eingeschaltet zu lassen.

In der Zeile „$\Delta l = 1$“ der Tabelle 6.2 wurde unabhängig vom Synchronitätsradius die Oberfläche $E^O(1)$ berechnet. Dies verschlechtert das Ergebnis in allen Bereichen gegenüber dem Baseline-Algorithmus. Der Teilerfilter wurde allerdings weiter verwendet. Dadurch wird der Einfluss von hohen Harmonische häufig unterdrückt. Das kann daran beobachtet werden, dass eine starke Gewichtung hoher Harmonischer ($u = 15$) nicht zu einer höheren Voiced Error Rate führt. Andererseits kann gesehen werden, dass eine starke Gewichtung niedriger Harmonischer ($u = 6$) zu einer höheren Voiced Error Rate führt.

Um nachzuweisen, dass die Dämpfung von hohen Harmonischen ebenfalls bei der Minimierung von Abtastfehlern hilft, wurde der alternative Teilerfilter T' aus Gleichung 4.16 getestet (siehe Tabelle 6.3). Es fällt auf, dass ein sehr kleiner Faktor α genügt, um Oktavenfehler zu vermeiden - das Spectral Subtraction muss für die Körbe h_j^2 und h_j^{-2} nur wenig verstärkt werden. Der Teilerfilter T' bewirkt eine stärkere Gewichtung des Rauschens in den Körben h_j^2 und h_j^{-2}. Ebenso findet im ersten Iterationsschritt nicht sofort eine endgültige Synchronisation auf eine Oktave statt, weshalb die Wahrscheinlichkeit steigt, die richtige Oktave anzusteuern, falls Energie bei der Hälfte der grundfrequenz-skalierten Fensterlänge gedämpft wird.

Für den Teilerfilter T' kann ein Anstieg des Voiced Errors für $u = 15$ beobachtet werden (siehe Tabelle 6.3). Wird Δl unabhängig vom Synchronitätsradius auf eins gesetzt, steigt der Voiced Error stark. Das deutet darauf hin, dass es dann kaum noch gelingt, nachfolgenden Iterationsebenen einen Hinweis über die Grundfrequenz

zu gegeben.

6.3. Verschlussmomente

In diesem Abschnitt sind Referenz-Algorithmus und Test-Algorithmus wie in Abschnitt 6.2 definiert. Falls m stimmhaft ist, entscheiden die Algorithmen zusätzlich, ob m ein Verschlussmoment ist.

Als Referenz-Algorithmus wird DYPSA bei klarer Sprache verwendet [NKGB07]. Es wird die Funktion *dypsa* der VOICEBOX [Bro07] benutzt, die von Mike Brookes erstellt wurde, der an der Entwicklung von DYPSA beteiligt war.

Da DYPSA keine Entscheidung über die Stimmhaftigkeit trifft, werden Abtastpunkte als *Referenz-Verschlussmomente* ausgewählt, die DYPSA als Verschlussmomente angibt, und die der Test-Algorithmus als stimmhaft berichtet[2].

Als *Test-Verschlussmomente* werden Abtastpunkte bezeichnet, die der Test-Algorithmus als Verschlussmomente ausgibt und für welche der Referenz-Algorithmus stimmhaft ausgibt.

Jedem Referenz-Verschlussmoment werden die Verschlussmomente des Test-Algorithmus zugeordnet, die ihm näher als anderen Referenz-Verschlussmomenten sind.

- Als *Missed Error Rate* (MER) wird der Anteil der Referenz-Verschlussmomente bezeichnet, denen kein Verschlussmoment des Test-Algorithmus zugeordnet wurde.
- Als *Missed by Voiceless Error Rate* (MVR) wird der Anteil der Referenz-Verschlussmomente bezeichnet, denen kein Verschlussmoment des Test-Algorithmus zugeordnet wurde und die vom Test-Algorithmus als stimmlos klassifiziert wurden.
- Als *Multiple Error Rate* (MPER) wird die Anzahl der Test-Verschlussmomente bezeichnet, die durchschnittlich auf einen Referenz-Verschlussmoment fallen, dem mindestens ein Test-Verschlussmoment zugeordnet wurde.
- Als *Good Detection Rate* (GDR) wird der Anteil der Test-Verschlussmomente bezeichnet, die innerhalb von 0.25 ms eines Referenz-Verschlussmomentes liegen.
- Als *Fine Error Rate* (FER) wird der Anteil der Test-Verschlussmomente bezeichnet, die innerhalb von 1 ms zum nächsten Referenz-Verschlussmoment liegen.
- Als *Gross Error Rate* (GER) wird der Anteil der Test-Verschlussmomente bezeichnet, die weiter als 1 ms vom nächsten Referenz-Verschlussmoment entfernt sind.

[2] Der Vollständigkeit halber findet sich in Tabelle A.1 eine Auswertung, bei welcher DYPSA die Entscheidungen des Harmonic Frontends von Behnke bezüglich der Stimmhaftigkeit übernimmt.

(a) DYPSA

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	1	0	0	0	0	1	0
20 dB	91,8	4,37	3,83	2,26	4,25	1,0272	0,15
15 dB	86,08	7,27	6,65	3,33	5,52	1,0465	0,19
10 dB	77,44	11,48	11,08	5,19	7,64	1,0767	0,24
5 dB	63,45	16,39	20,16	9,21	12,08	1,1411	0,30
0 dB	46,38	20,41	33,21	17,44	20,87	1,2375	0,36
-5 dB	30,08	23,57	46,35	32,78	36,82	1,3371	0,45

(b) Ergebnis der Arbeit

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	57,56	38,73	3,71	0	2,52	1,0031	0,38
20 dB	57,45	38,60	3,94	2,26	4,64	1,0024	0,38
15 dB	57,28	38,44	4,28	3,34	5,66	1,0025	0,38
10 dB	57,15	38,03	4,81	5,22	7,50	1,0032	0,39
5 dB	56,80	37,01	6,19	9,31	11,66	1,0051	0,40
0 dB	54,87	36,40	8,73	17,62	20,15	1,0106	0,40
-5 dB	46,91	37,09	15,97	33,01	36,51	1,0332	0,44

Tabelle 6.4.: Ergebnisse der Verschlussmomentschätzung. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Verglichen wurde DYPSA als Referenz-Algorithmus mit dem in der vorliegenden Arbeit entwickelten Algorithmus als Test-Algorithmus. Der Test-Algorithmus verwendet als Parameter $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot k_{10}$, $p = 0.17 \cdot \pi$ und $u = 9$. Siehe Tabelle A für die Fehlerraten des Test-Algorithmus bezüglich Stimmhaftigkeit und Grundfrequenzerkennung.

- Der *Root Mean Square Error* (RMSE) zwischen Test-Verschlussmomenten und Referenz-Verschlussmomenten in ms wird für Abtastpunkte berechnet, die der Test-Algorithmus korrekt als stimmhaft erkannte und bei denen Test- und Referenz-Verschlussmoment nicht weiter als 1 ms auseinander liegen.

Ähnlich wie bei der Grundfrequenz-Evaluation wird die Genauigkeit für die Test-Verschlussmomente gemessen, die der Test-Algorithmus korrekt als stimmhaft erkennt.

In Tabelle 6.4 sind Ergebnisse der Verschlussmomentschätzung zu sehen. Bei klarer Sprache weicht der in dieser Arbeit entwickelte Algorithmus deutlich von DYPSA ab. Bei schlechter werdendem Signal-Rausch-Verhältnis lässt die Leistung des in der Arbeit entwickelten Algorithmus weniger nach als die Leistung von DYPSA. Die große Multiple Error Rate von DYPSA fällt auf. Diese ist dadurch be-

dingt, dass DYPSA viele GCI-Kandidaten generiert und von diesen per dynamischer Programmierung Verschlussmomente auswählt. Bei Rauschen werden sehr viele falsche GCI-Kandidaten generiert, so dass einige von diesen ausgewählt werden. Die GCI-Kandidaten, die DYPSA innerhalb von 1 ms Abweichung zum Referenz-Verschlussmoment findet, weisen auch bei Rauschen einen geringeren RMSE auf.

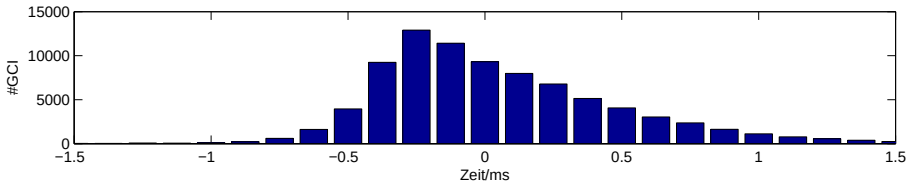
Die große Abweichung des Test-Algorithmus bei klarer Sprache ist zum Teil durch die Vorverarbeitung bedingt, die DYPSA vornimmt. Es wird eine Prä-Emphase hoher Frequenzen angewandt und zusätzlich wird das Restsignal des LPC verwendet (vergleiche Abschnitt 3.1.2). Dadurch wird der Einfluss der Formanten im besten Fall entfernt. Führt man die gleiche Vorverarbeitung auch bei dem in der Arbeit vorliegenden Algorithmus durch, so erreicht dieser bei klarer Sprache eine Good Detection Rate von 94.94%. Allerdings wurde bei der Grundfrequenzerkennung ein Voiced Error von 34 % begangen (siehe Tabelle A.5). Bei Rauschen ist die Vorverarbeitung allerdings schädlich, deshalb wird sie in der vorliegenden Arbeit nicht verwendet. Die restlichen Abweichungen können einerseits durch Fehler von DYPSA bedingt sein. Andererseits wird beim in der Arbeit entwickelten Algorithmus ein Fenster verwendet, das vier Phonationszyklen umfasst, und daher grundsätzlich weniger genau Verschlussmomente erfassen kann, als DYPSA, das ein Fenster von etwas weniger als einem Phonationszyklus verwendet. Ebenso hängt die Genauigkeit des entwickelten Verfahrens von der Güte der Grundfrequenzschätzung ab, weil der Indikator auf grundfrequenz-skalierte Fensterlängen angewiesen ist (siehe auch Tabelle A.3 für die Ergebnisse der Grundfrequenzerkennung bei den Parametern in Tabelle 6.4). Zusammenfassend lokalisiert DYPSA die Verschlussmomente genauer, weist aber bei Rauschen eine hohe Multiple Error Rate auf.

In Abbildung 6.1 ist die Verteilung der Test-Verschlussmomente bezüglich der Referenz-Verschlussmomente dargestellt. Eine Abweichung von 0 ms bedeutet, dass der Test-Verschlussmoment auf dem gleichen Abtastpunkt landete wie der Referenz-Verschlussmoment.

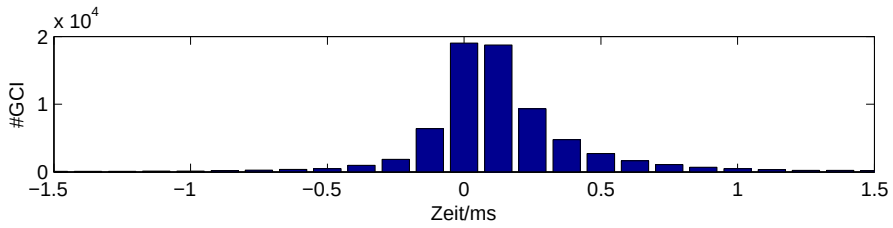
Ein Synchronisationswert von $p = 0.17 \cdot \pi$ wurde in Abbildung 6.1(a) verwendet. Es fällt auf, dass die Verschlussmomente systematisch verzögert gefunden werden - trotz des Synchronisationswertes von $p = 0.17 \cdot \pi$. Das lässt auf den Einfluss der Formanten schließen

Die Verteilung der Test-Verschlussmomente bei einem Signal-Rausch-Verhältnis von 0 dB ist in Abbildung 6.1(b) dargestellt. Als Referenz-Verschlussmomente wurden dort die Verschlussmomente verwendet, die der Test-Algorithmus bei klarer Sprache erkannte. Der Test-Algorithmus verwendete dazu einen Synchronisationswert von $p = \pi$.

Dieser Fall ist für die Experimente in den Abschnitten 4.6.4 und 4.7.2 relevant, wo es nicht notwendig ist, sich tatsächlich mit den den Verschlussmomenten zu synchronisieren. Dort ist es entscheidend, dass glottissynchrone Frames bei Rauschen und klarer Sprache einen annähernd gleiche Lage aufweisen. In Tabelle 6.5 ist zu sehen, wie die Erkennung bei schlechter werdendem Signal-Rausch-Verhältnis nachlässt. Bei einem Signal-Rausch-Verhältnis von 0 dB liegen 76.93 % der Test-Verschlussmomente innerhalb von 0.25 ms eines Referenz-Verschlussmomentes.



(a) $p = 0.17 \cdot \pi$, klare Sprache (Test Set A, Subset 1 der Aurora-2 Datenbank)



(b) $p = 0$, Rauschen (Test Set A, Subset 1 der Aurora-2 Datenbank bei 0 dB)

Abbildung 6.1.: Verteilung der Test-Verschlussmomente. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Oben ist der Referenz-Algorithmus DYPSA, unten ist der Test-Algorithmus gleichzeitig der Referenz-Algorithmus. Der Test-Algorithmus verwendete jeweils als Parameter $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot k_{10}$, und $u = 9$.

(a) Synchronisation

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	0	0	0	0	0	0	0
20 dB	98,91	0,1	0,2	2,33	2,53	1,0019	0.08
15 dB	97,87	1,8	0,3	3,43	3,65	1,0022	0.11
10 dB	95,07	4,28	0,7	5,24	5,47	1,0034	0,15
5 dB	89,28	9,56	1,2	9,07	9,41	1,0049	0,20
0 dB	76,93	19,08	4	17,56	18,34	1,0123	0,28
-5 dB	57,41	29,45	13,1	32,42	34,85	1,0358	0,38

Tabelle 6.5.: Ergebnisse der Synchronisation. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Als Referenz-Verschlussmomente werden die Test-Verschlussmomente bei klarer Sprache verwendet. Der Test-Algorithmus verwendet die Parameter $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot k_{10}$, $p = \cdot\pi$ und $u = 9$.

(a) Harmonic Frontend

SNR/dB	Subway	Babble	Car	Exhibition	Average
20 dB	98,53	98,28	98,48	98,36	98,41
15 dB	97,91	97,46	98,00	97,47	97,71
10 dB	96,68	95,62	96,60	95,68	96,15
5 dB	92,60	90,15	93,08	88,83	91,17
0 dB	79,46	72,76	83,45	72,60	77,07
-5 dB	44,64	36,28	48,20	36,36	41,37
Average	93,04	90,85	93,92	90,59	92,1

(b) Harmonic Frontend

SNR/dB	Restaurant	Street	Airport	Train Station	Average
20 dB	98,77	97,82	98,72	98,61	98,48
15 dB	98,04	97,25	97,94	97,93	97,78
10 dB	96,25	95,56	95,53	95,03	95,59
5 dB	89,28	90,72	88,94	88,65	89,40
0 dB	70,28	75,97	74,71	73,62	73,65
-5 dB	34,66	41,93	42,38	39,74	39,68
Average	90,52	91,46	91,17	90,77	90,98

Tabelle 6.6.: Aurora-2 Ergebnisse des Harmonic Frontend von Behnke [Beh]. Word Accuracy in Prozent für das Test Set A und B der Aurora-2 Datenbank bei Multi-Condition Training.

6.4. Merkmalsextraktion

Zur Evaluation des Frontends wird die Aurora-2 Datenbank herangezogen [PgHG00]. Die Spracherkennung wird von einem Backend übernommen. Die Routine zur Merkmalsextraktion wird aus dem Harmonic Frontend von Behnke übernommen. Diesem werden dazu Schätzungen über das Quellsignal übergeben. Gemessen wird die *Word Accuracy* in Prozent. Diese berücksichtigt, dass Worte nicht erkannt werden (*Deletion Error*), falsch erkannt werden (*Substitution Error* oder dass der Hintergrund als ein Wort erkannt wird (*Insertion Error*).

In der Tabelle 6.7 sind die Ergebnisse der vorliegenden Arbeit für die Aurora-2 Datenbank abgebildet. Verglichen wurde der in der vorliegenden Arbeit entwickelte Algorithmus mit dem Harmonic Frontend von Behnke (siehe Abschnitt 3.2) [Beh]. Die Ergebnisse des Harmonic Frontends sind in Tabelle 6.6 abgebildet.

Verglichen mit dem Harmonic Frontend von Behnke erreicht der Algorithmus eine schlechtere Word Accuracy. Besonders stark viel die Leistung bei Reden im Hintergrund ab. Das ist allgemein schweres Rauschen, gerade für einen Algorithmus, der auf der Erkennung der Grundfrequenz basiert. Eine Synchronisation mit Reden im Hintergrund wirkt sich negativ aus, wenn gerade stimmhafte Sprache einsetzt, die bei einer anderen Grundfrequenz zu finden ist. Bedingt durch die Synchroni-

(a) Ergebnis der Arbeit

SNR/dB	Subway	Babble	Car	Exhibition	Average
20	98.59	97,91	98,18	98,24	98,23
15	98.00	97,61	97,91	97,62	97,79
10	96.16	95,5	96,03	95,59	95,82
5	91.5	88,72	92,22	90,06	90,63
0	76,6	64,57	78,14	72,85	73,04
-5	38,26	22,73	34,87	34,87	31,67
Average	92,17	88,86	90,87	90,87	91,1

(b) Ergebnis der Arbeit

SNR/dB	Restaurant	Street	Airport	Train Station	Average
20	98,43	98,16	98,27	98,24	98,28
15	97,05	97,13	97,11	97,16	97,11
10	94,01	95,1	94,54	93,49	94,29
5	82,62	89,66	86,73	85,84	86,21
0	54,02	70,28	67,01	66,99	64,58
-5	31,75	16,23	26,9	32,89	26,94
Average	85,23	90,07	88,73	88,34	88,09

Tabelle 6.7.: Aurora-2 Ergebnisse der vorliegenden Arbeit. Word Accuracy in Prozent für das Test Set A und B der Aurora-2 Datenbank bei Multi-Condition Training. Als Parameter werden $C = 9$, $B = 10$, $c = 1 \cdot k_{10}$, $c_2 = 0.6 \cdot k_{10}$, $u = 9$ und $p = \pi$ verwendet.

	Subway	Babble	Car	Exhibition	Durchschnitt	Insgesamt
Baseline	76,6	64,57	78,14	72,85	73,04	91,1
Unsynchron	73,47	62,55	76,86	72,94	71,46	90,7
$p = 0$	74,95	60,67	76,95	72,63	71,3	90,59
Linear	76,11	63,21	77,63	73	72,49	90,93
Klare Sprache	82.65	85.94	87.35	82.23	84,54	94,04
$\alpha = 0.07$	74,55	59,64	77,99	73,46	71,41	90,55
$C = 5$	74,76	61,28	75,45	72,72	71,05	90,67

Tabelle 6.8.: Ergebnisse für unterschiedliche Parameter. Bei einem Signal-Rausch-Verhältnis von 0 dB ist die Word Accuracy in Prozent für das Test Set A der Aurora-2 Datenbank bei Multi-Condition Training dargestellt. In der Spalte „Insgesamt“ ist der Durchschnitt über die Signal-Rauschraten von 0 bis 20 dB zu sehen. Als Parameter des Baseline-Algorithmus wird $C = 9$, $B = 10$, $c = 1 \cdot k_{10}$, $c_2 = 0.6 \cdot k_{10}$, $u = 9$ und $p = \pi$ gewählt.

sation mit Sprechern im Hintergrund treten häufig Insertion Errors auf, bei denen fälschlicherweise Sprache erkannt wird, obwohl der Sprecher schweigt. Im Backend kann die sogenannte *Worteinfüge-Wahrscheinlichkeit* variiert werden, um die Wahrscheinlichkeit für das Einfügen von Wörtern einzustellen. Dadurch konnte die Word Accuracy verbessert werden, weil die schlechte Word Accuracy bei den Rauscharten „Babble“ und „Restaurant“ im Wesentlichen auf Insertion Errors zurückzuführen ist, die aufgrund der häufigen Synchronisation mit dem Hintergrund bei harmonischem Rauschen auftreten (siehe Tabelle A.2). Im Unterschied zum Harmonic Frontend benutzte der Test-Algorithmus begrenzte Vorausschau zur Bestimmung der Grundfrequenz und weniger Abtastpunkte zur Berechnung der harmonischen Oberfläche.

Im Vergleich erreichte ein ETSI Standard aus dem Jahr 2003 mit dem gleichen Backend, das in dieser Arbeit verwendet wurde, eine durchschnittliche Word Accuracy von 91.1% auf dem Test Set A und eine durchschnittliche Word Accuracy von $88,67\%$ auf dem Test Set B [Beh]. Im Gegensatz zur vorliegenden Arbeit wird im ETSI Standard auch die Merkmalsextraktion durchgeführt, ohne das gesamte Signal gesehen zu haben. Der ETSI Standard ES 202 050 aus dem Jahr 2003 wurde im Jahr 2006 von David Pearce auf die Aurora-2 Datenbank angewandt und erreichte eine durschnittliche Erkennungsrate von 92.29% [Eur03b, Aur]. Allerdings wurde dazu ein anderes Backend als in der vorliegenden Arbeit verwendet.

Zur Evaluation von verschiedenen Eigenschaften wurde der Algorithmus mit verschiedenen Einstellungen gestartet und auf dem Test Set A der Aurora-2 Datenbank getestet. In Tabelle 6.4 sind die Ergebnise dieser Testläufe in einer Tabelle zusammengefasst. Als Baseline-Algorithmus wurden die oben beschriebenen Einstellungen verwendet. In der Zeile „Unsynchron“ verwendete der Test-Algorithmus konstante Frameabstände. Das leicht schlechtere Ergebnis ist ein Hinweis darauf, dass die Erkennungsrate durch die Synchronisation und die dadurch notwendige Berücksichtigung von variablen Frameabständen zumindest nicht wesentlich schlechter wird. Für einen Synchronisationswert von $p = 0$ liegen die Ränder der Analysefenster auf den Verschlussmomenten. Das die Erkennungsrate bei einer derartigen Synchronisation nachlässt, ist ein Hinweis darauf, dass es bei der Verwendung von rechteckigen Fenstern sinnvoll ist, einen Synchronisationswert von $p = \pi$ zu wählen. In der Zeile „Linear“ wurde anstatt der sonst verwendeten Nearest-Neigbhor Interpolation eine Lineare Interpolation durchgeführt, um fehlende Einträge der harmonischen Oberfläche zu bestimmen (siehe Abschnitt 4.5.3.1). Das zeigte ein geringfügig schlechteres Ergebenis und spricht dafür, eine Nearest-Neighbor Interpolation zu verwenden. In dem Experiment „Klare Sprache“ wurde das Quellsignal bei klarer Sprache annotiert und diese Informationen wurden dann bei verrauschter Sprache zur Merkmalsextraktion verwendet. Bei einem Signal-Rausch-Verhältnis von 0 dB konnte so eine durchschnittliche Erkennungsrate von $84,54\%$ erreicht werden. Das zeigt das Potential, dass im Wissen über die Grundfrequenz liegt. Die Verwendung des alternativen Teilerfilters T' mit dem Parameter $\alpha = 0.07$ führte zu leicht schlechteren Ergebnissen. Es ist an dieser Stelle nicht klar, ob mit weiterer Optimierung eine ähnliche Leistung erreicht werden kann, wie mit dem in dieser Arbeit gewählten Teilerfilter. Für $C = 5$ ließ die Erkennungsrate insgesamt um $0,43\%$ nach, bei 0 dB allerdings um $1,99\%$. Hier lässt sich der notwendige Trade-Off zwischen der Minimierung von

Abtastfehlern und dem Aufwand bei der Berechnung der harmonischen Oberfläche erkennen.

6.5. Visuelle Evaluation

In diesem Abschnitt wird beispielhaft das Verhalten des Algorithmus bei klarer und verrauschter Sprache gezeigt.

6.5.1. Klares Sprachsignal

In Abbildung 6.2 sind harmonische Oberflächen für ein Sprachsignal abgebildet. Die harmonische Oberfläche, wie sie die erste Iterationsebene sieht, ist in Abbildung 6.2(a) für ein klares Sprachsignal dargestellt. Die weißen Stellen der harmonischen Oberfläche werden nicht berechnet, weil diese bereits in der ersten Iterationsebene ausgeschlossen werden.

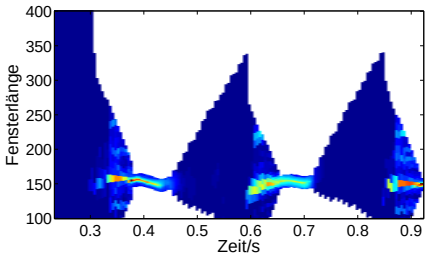
Bei einsetzender stimmhafter Sprache wird der Synchronitätsradius geringer, die harmonische Oberfläche kann dann genauer untersucht werden. Sobald ein gewisser Synchronitätsradius unterschritten ist, wird der Vielfachfilter ausgeschaltet. Das ist deutlich an dem hellblauen Bereich um die grundfrequenz-skalierte Fensterlänge herum zu erkennen, der auftaucht, wenn der Synchronitätsradius einen gewissen Wert unterschreitet. Dieser Sprung in der harmonischen Oberfläche zwischen benachbarten Spalten, wird in Kauf genommen, damit bei niedrigem Synchronitätsradius keine Energie aus den harmonischen Körben der grundfrequenz-skalierten Fensterlänge gelöscht wird und bei großem Synchronitätsradius Oktavenfehler vermieden werden. Der Teilerfilter bleibt unabhängig vom Synchronitätsradius eingeschaltet. Bei klarer Sprache ist die harmonische Oberfläche daher auch bei hohem Synchronitätsgrad weniger scharf, als das bei der Oberfläche $E(1)$ der Fall ist (vgl. Abbildung 4.7 und Abschnitt 4.5.2).

Abgesehen von dem starken Sprung beim Ausschalten des Vielfachfilters, kann beobachtet werden werden, dass die Energie von einer Spalte E_{t_i} zur Spalte $E_{t_{i+1}}$ bei variendem Synchronitätsradius nicht drastisch variiert.

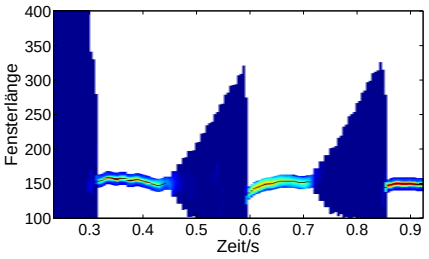
Wenn ein stimmhafter Ausschnitt im Sprachsignal beginnt, springt die zentrale Fensterlänge der ersten Iterationsebene auf die (im Idealfall) grundfrequenz-skalierte Fensterlänge. Gleichzeitig sinkt der Synchronitätsradius nur langsam, so dass die Wahl einer falschen Fensterlänge keine drastischen Auswirkungen hat. Liegt kein stimmhafter Ausschnitt vor, wird die zentrale Fensterlänge langsam in Richtung der Fensterlänge 250 gezogen, damit bei einem Synchronitätsgrad von 0 alle Fensterlängen abgedeckt werden.

In der letzten Iterationsebene wird die harmonische Oberfläche am genauesten berechnet (siehe Abbildung 6.2), weist aber auch die größten Sprünge bezüglich des Synchronitätsradius auf. Schwarz werden die Abtastpunkte und Fensterlängen markiert, bei denen der Algorithmus stimmhafte Sprache und die Grundfrequenz erkennt.

Als Schwäche fällt auf, dass auch in der letzten Iterationsebene der Synchronitätsgrad nicht überall eins beträgt, wo harmonische Energie auf der Oberfläche zu



(a) Erste Iterationsebene bei klarer Sprache.



(b) Letzte Iterationsebene bei klarer Sprache.

Abbildung 6.2.: Harmonische Obeflächen. Zu sehen sind harmonische Oberflächen, die von verschiedenen Iterationsebenen berechnet wurden.

beobachten ist. Es wird ein erhöhter Voiced Error in Kauf genommen (vgl. Tabelle 6.1). Eine weitere Schwäche ist es, dass die Konfidenz eins betragen kann, ohne dass der Synchronitätsgrad eins beträgt. Das hat zur Folge, dass die Grundfrequenz dann nur ungenau geschätzt werden kann.

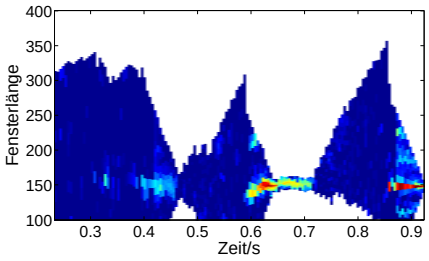
6.5.2. Verrauschte Sprachsignale

Bei verrauschten Sprachsignalen können die Schwächen des Algorithmus beobachtet werden. In Abbildung 6.3(a) kann beobachtet werden, dass es dem Algorithmus in der ersten Iterationsebene erst spät gelingt, sich mit dem ersten Segment stimmhafter Sprache zu synchronisieren, was durch Rauschen und Abtastprobleme verursacht wird.

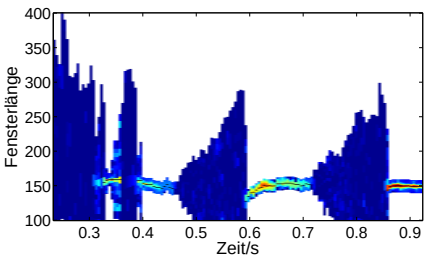
Bei einem Synchronitätsgrad von Eins fällt auf, dass in dem mittleren Segment stimmhafter Sprache Unstetigkeiten bei der grundfrequenz-skalierten Fensterlänge auftreten. Das liegt daran, dass der Teilerfilter auch bei einem Synchronitätsgrad von eins eingeschaltet bleibt. Dadurch werden nur Harmonische übrig gelassen, die weit aus dem Rauschen herausragen. Im Idealfall sind das bei einer grundfrequenz-skalierten Fensterlänge alle Harmonischen. Es fällt auf, dass die Harmonischen in benachbarten Spalten E_m bei unterschiedlich starken Abweichungen von der grundfrequenz-skalierten Fensterlänge abgeschnitten werden, was auf die unterschiedliche Stärke des Rauschens schließen lässt.

Die erste Iterationsebene ist in Abbildung 6.3(b) dargestellt. Dort wurde auch beim ersten Segment stimmhafter Sprache teilweise ein Synchronitätsgrad von eins erreicht. Wie in Abbildung 6.2(b) wurden Frames als stimmhaft markiert, die in der letzten Iterationsebene keinen Synchronitätsgrad von eins aufwiesen. Obwohl die Grundfrequenzschätzung dann ungenau ist, wurden die Frames korrekt als stimmhaft klassifiziert. In der Mitte des ersten Segments in Abbildung 6.3(b) ist zu sehen, dass der Algorithmus die Synchronität verliert. Das liegt im Wesentlichen daran, dass die erste Iterationsebene es nicht schaffte sich teilweise zu synchronisieren[3]. Daher müssen (fast) alle weiteren Iterationsschritte konfidente Messungen zeigen, damit ein

[3]Dieses Verhalten kann durch den Parameter c_2 gesteuert werden, siehe Abschnitt 6.2



(a) Erste Iterationsebene bei 0 dB.



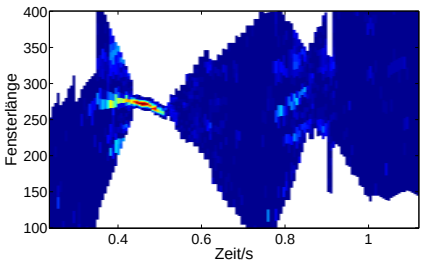
(b) Letzte Iterationsebene bei 0 dB.

Abbildung 6.3.: Harmonische Obeflächen. Bei einem Signal-Rausch-Verhältnis von 0 dB ist die harmonische Oberfläche der ersten Iterationsebene in Abbildung 6.3(a) und der letzten Iterationsebene in Abbildung 6.3(b) zu sehen. In Abbildung 6.2 finden sich die harmonischen Oberflächen, wie sie bei klarer Sprache gesehen wurden.

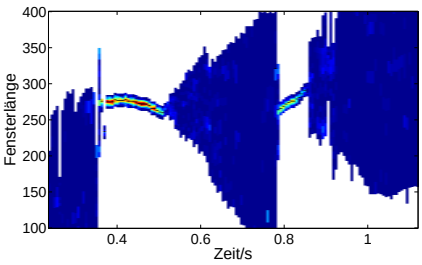
Synchronitätsgrad von eins erreicht werden kann[4]. Am Ende des ersten Segments kann beobachtet werden, dass der Algorithmus einige Frames als stimmhaft markierte, die bei klarer Sprache als nicht stimmhaft markiert wurden. Das kann zum einen mit starkem Rauschen zusammenhängen. Eine andere Ursache sind Energieunterschiede für unterschiedliche Synchronitätsradien, so dass der Energielevel nicht optimal kontrollierbar ist (siehe Gleichung 4.27).

In Abbildung 6.4(b) kann beobachtet werden, dass Sprünge in der letzten Iterationsebene von Frame zu Frame auftreten können. Das geschieht besonders zu Beginn eines Sprachsignals, wenn noch nicht genug harmonische Energie im akkumulierten Maximum bei der grundfrequenz-skalierten Fensterlänge gesammelt wurde. In der harmonischen Oberfläche der ersten Iterationsebene ist dort, wo die letzte Iterationsebene fälschlicherweise hinspringt, keine harmonische Energie zu sehen. Die Energie erscheint bei niedriger werdendem Synchronitätsradius, weil der Vielfachfilter ausgeschaltet wird. In Abbildung 6.4(b) traten in dem letzten Segment stimmhafter Sprache keine Sprünge in der letzten Iterationsebene auf, obwohl die erste Iterationsebene sich nicht vollständig synchronisierte.

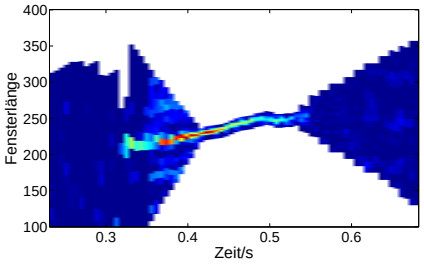
[4] Dieses Verhalten kann durch den Parameter c gesteuert werden, siehe Abschnitt 6.2



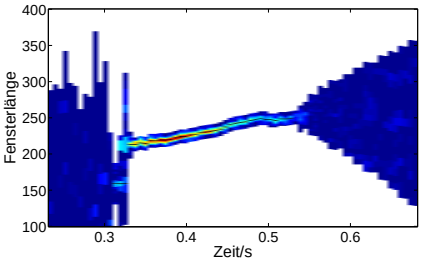
(a) Erste Iterationsebene



(b) Letzte Iterationsebene



(c) Erste Iterationsebene



(d) Letzte Iterationsebene

Abbildung 6.4.: Harmonische Obeflächen. Bei einem Signal-Rausch-Verhältnis von 0 dB ist die harmonische Oberfläche der ersten Iterationsebene und der letzten Iterationsebene für zwei verschiedene Sprachsignale dargestellt.

Kapitel 7.

Zusammenfassung

In der vorliegenden Arbeit wurde das Ziel verfolgt, Sprache automatisch und robust zu erkennen. Dazu wurden die Eigenschaften des Quellsignals stimmhafter Sprache ausgenutzt. Der Schwerpunkt der Arbeit lag auf der robusten Verfolgung von Grundfrequenz und Verschlussmomenten bei begrenzter Vorausschau.

Im Hauptteil wurde zunächst eine Struktur für eine iterative Verarbeitung mit begrenzter Vorausschau entworfen. Dabei gab es zwei Motivationen. Zum einen war die Motivation, durch die iterative Verarbeitung Entscheidungen nicht sofort treffen zu müssen. Stattdessen können zunächst Schätzungen abgegeben werden, die in die richtige Richtung weisen. Diese werden dann iterativ verbessert. Zum anderen sollte die iterative Verarbeitung helfen, die Laufzeit zu verkürzen, indem zunächst grobe Berechnungen mit wenig Aufwand durchgeführt werden, um den Suchraum einzugrenzen.

In die entworfene Struktur für die iterative Verarbeitung wurden Verfahren zur Verfolgung von Grundfrequenz und Verschlussmomenten integriert.

Um die Verfolgung der Grundfrequenz in die iterative Verarbeitung zu integrieren, wurden zunächst Methoden gesucht, um Abtastfehler bei der Berechnung einer harmonischen Oberfläche mit wenigen Abtastpunkten zu minimieren. Anschließend wurde erläutert, wie Iterationsketten für die Grundfrequenzverfolgung erweitert werden. Dabei wurde in frühen Iterationsebenen auf der harmonischen Oberfläche mit wenigen Abtastpunkten ein großer Grundfrequenzbereich abgedeckt. In späteren Iterationsebenen wurde die harmonische Oberfläche dort genauer untersucht, wo frühere Iterationsebenen die Grundfrequenz vermuteten. Ein Synchronitätsradius wurde verwendet, um den Suchraum für Grundfrequenzen langsam einzugrenzen, sobald stimmhafte Sprache erkannt wurde. Dieser wurde dabei nur verringert, wenn stimmhafte Sprache gefunden wurde, um keine falschen Entscheidungen treffen zu müssen. Aus diesem Grund wurde der Synchronitätsradius in allen Iterationsebenen variiert. Dann wurde ein iteratives Verfahren vorgestellt, mit dem Rauschen schrittweise aus einem Sprachsignal gefiltert wurde. Dazu wurden Rauschschätzung und Grundfrequenzschätzung aufeinander aufbauend entlang von Iterationsketten verbessert.

Zur Verfolgung von Verschlussmomenten wurde zunächst ein Indikator entwickelt, der im Zusammenhang mit einer Grundfrequenzschätzung auf Verschlussmomente zeigt. Dieser wurde dann in der iterativen Verarbeitung verwendet, um glottissynchrone Verarbeitung durchzuführen. Dabei wurden die Analysefenster zur Untersuchung des Sprachsignals in dem iterativen Verfahren jeweils dort platziert, wo der Indikator die Verschlussmomente vermutete. Das Verhalten der Phase in den harmonischen Körben glottissynchroner Frames wurde untersucht und mögliche An-

wendungen der glottissynchronen Verarbeitung für die robuste automatische Spracherkennung wurden diskutiert. Es wurde visuell die Ähnlichkeit zwischen Amplitudengang und einer entwickelten Darstellung des Phasengangs gezeigt. Schließlich wurden die glottissynchronen Frames in einem existierenden Frontend für die robuste automatische Spracherkennung verwendet.

In Kapitel 6 wurden die Eigenschaften des entwickelten Algorithmus evaluiert. Es wurde gezeigt, dass die Leistung des Algorithmus bei der Erkennung der Grundfrequenz nachlässt, wenn die harmonische Oberfläche mit wenigen Abtastpunkten untersucht wird, ohne die entworfenen Methoden für die Berechnung der harmonischen Oberfläche zu verwenden. Das ließ erkennen, dass die entworfenen Methoden erfolgreich bei der Minimierung von Abtastfehlern helfen.

Dann wurde bei der Evaluation festgestellt, dass ein angemessenes Variieren des Synchronitätsradius dabei hilft, Oktavenfehler zu vermeiden und keine stimmhafte Sprache zu verfehlen. Ein zu langsames Verringern des Synchronitätsradius resultierte darin, dass stimmhafte Sprache seltener erkannt wurde. Wurde der Synchronitätsradius zu schnell verringert, führte das zu einer vermehrten Synchronisation mit dem Hintergrund und zu Oktavenfehlern. Ebenso zeigte es sich, dass die Variation des Synchronitätsradius auf allen Iterationsebenen auch Probleme mit sich bringt, weil zum einen Normalisierungsschritte durchgeführt werden müssen und zum anderen Oktavenfilter bei großen Synchronitätsradien benötigt werden.

Es zeigte sich, dass die Verwendung der Iterationsebenen dabei hilft, weniger Punkte der harmonischen Oberfläche zu Berechnen. Zum einen müssen bei der Grundfrequenzerkennung nicht alle Punkte der harmonischen Oberfläche für einen Zeitpunkt berechnet werden. Zum anderen muss bei der glottissynchronen Verarbeitung die harmonische Oberfläche nicht für jeden Zeitpunkt berechnet werden. Es wurde festgestellt, dass die Möglichkeiten der entworfenen Struktur zur iterativen Verarbeitung für die Verbesserung von Zwischenergebnissen nicht ausreichend genutzt werden. Hier konnten zwei Problemquellen identifiziert werden. Erstens wurden bei der Grundfrequenzverfolgung zur Berechnung eines Iterationspunktes keine Informationen von Iterationsketten verwendet, die in der Zukunft liegen. Zweitens konnte die schrittweise Filterung des Signals entlang der Iterationsketten in der vorliegenden Arbeit nicht so weit untersucht und optimiert werden, dass ein Gewinn erzielt werden konnte. Im Vergleich zu einem Referenz-Algorithmus zur Grundfrequenzerkennung [Beh] war der entwickelte Algorithmus weniger robust. Abzuwägen ist dabei, dass der entwickelte Algorithmus im Unterschied zum Referenz-Algorithmus nur begrenzte Vorausschau hatte und die harmonische Oberfläche nicht für alle Fensterlängen berechnete.

Im nächsten Schritt wurde die Verfolgung der Verschlussmomente evaluiert. Es konnte gezeigt werden, dass eine robuste Verarbeitung der Phaseninformation in den harmonischen Körben von grundfrequenz-skalierten DFTen möglich ist. Im Gegensatz zu einem Referenz-Algorithmus (DYPSA) war die Erkennung der Verschlussmomente für klare Sprache ungenauer [NKGB07]. Bei verrauschter Sprache wurden dagegen weniger Zeitpunkte fälschlich als Verschlussmomente erkannt und die Genauigkeit war ähnlich hoch wie die des Referenz-Algorithmus. Als Ursache für die geringere Genauigkeit bei klarer Sprache konnten zwei Ursachen ausgemacht werden.

Zum einen führt DYPSA eine Vorverarbeitung des Signals durch. Wurde diese auch vom hier entwickelten Algorithmus durchgeführt, konnten bei klarer Sprache die Verschlussmomente genauer erkannt werden, allerdings immer noch mit Abweichungen von DYPSA. Bei Rauschen ließ die Erkennung unter Verwendung der Vorverarbeitung deutlich nach. Eine offene Frage ist, wie sich DYPSA bei Rauschen verhält, wenn die Vorverarbeitung nicht durchgeführt wird. Da für DYPSA hauptsächlich fälschlich erkannte Verschlussmomente ein Problem darstellen, stellt sich die Frage, ob diese ohne die Vorverarbeitung seltener auftreten würden. Zum anderen verwendet DYPSA ein Analysefenster, das weniger als einen Phonationszyklus abdeckt. Um eine robuste Erkennung der Verschlussmomente zu gewährleisten, wurde in der vorliegenden Arbeit ein grundfrequenz-skaliertes Fenster verwendet, dass vier Phonationszyklen abdeckt und daher lokale Ereignisse ungenauer erfassen kann. Als Vorteil gegenüber DYPSA ist zu nennen, dass der entwickelte Algorithmus Information über die Grundfrequenz verwendet. Ein Nachteil ist, dass nur begrenzte Vorausschau erlaubt ist, wohingegen DYPSA das Sprachsignal im vorhinein kennt und dynamische Programmierung verwendet.

Die glottissynchronen Frames, die der entwickelte Algorithmus als Ausgabe lieferte, wurden als Eingabe für ein existierendes Frontend zur Merkmalsextraktion verwendet (siehe Abschnitt 4.7). Die durchschnittliche Word Accuracy war vergleichbar mit einem Ergebnis des European Telecommunications Standards Institute (ETSI) aus dem Jahr 2003 [Beh]. Allerdings wurde in der vorliegenden Arbeit für die Merkmalsextraktion keine begrenzte Vorausschau verwendet. Verglichen mit der Word Accuracy des Harmonic Frontends von Behnke war die Erkennungsleistung geringer. Im Unterschied zum Harmonic Frontend wurde in der vorliegenden Arbeit begrenzte Vorausschau bei der Verfolgung der Grundfrequenz verwendet.

In einem Experiment konnte festgestellt werden, dass die Word Accuracy deutlich anstieg, wenn der Algorithmus die Grundfrequenz bei klarer Sprache schätzte und diese Information an das Frontend übergab. Dies lässt auf das Potential schließen, dass in dem Wissen über die Grundfrequenz zur robusten Spracherkennung liegt. Diese Erkenntnis motiviert die weitere Untersuchung der hier verfolgten Fragestellungen, um noch besser Sprache unter schweren Bedingungen zu erkennen. Ausgehend von den Ergebnissen der vorliegenden Arbeit kann weitere Forschung betrieben werden.

Es kann untersucht werden, wie die vorgeschlagene iterative Struktur weiter ausgenutzt werden kann. In der vorliegenden Arbeit wurde eine schrittweise Filterung durchgeführt, die noch nicht ausreichend untersucht werden konnte. Eine weitere offene Frage ist, wie eine Interaktion zwischen benachbarten Iterationsketten für die Grundfrequenzverfolgung erreicht werden kann.

Es wurde gezeigt, dass der Phasengang bei grundfrequenz-skalierter Vorgehensweise in den harmonischen Körben robust verarbeitet werden kann. Das kann in weiterführenden Arbeiten auf neue Anwendungen untersucht werden.

Eine weitere Möglichkeit besteht darin, das entwickelte Verfahren so zu ändern, dass abhängig vom Signal-Rausch-Verhältnis vorgegangen wird. Dies ist dadurch motiviert, dass sich gezeigt hat, dass bei klarer Sprache die Verschlussmomente mit Hilfe einer Vorverarbeitung deutlich genauer erkannt werden können. Bei klarer Sprache

können zudem die hohen Harmonischen stärker gewichtet werden, was gegebenfalls hilft, auch die Grundfrequenz genauer zu schätzen.

Anhang A.

Evaluation

(a) DYPSA

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	1	0	0	0	0	1	0
20 dB	89,80	5,02	5,17	4,73	6,99	1,0336	0,16
15 dB	84,43	7,80	7,76	6,25	8,77	1,0505	0,19
10 dB	76,31	11,81	11,88	8,84	11,54	1,0778	0,24
5 dB	63,33	16,57	20,10	13,60	16,70	1,1342	0,30
0 dB	46,91	20,31	32,78	21,73	25,29	1,2257	0,36
-5 dB	31,50	23,15	45,35	39,51	43,29	1,3213	0,44

(b) Ergebnis der Arbeit

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	57,81	38,29	3,9	13,74	15,93	1,0034	0,37
20 dB	57,66	38,51	3,83	14,07	16,24	1,0033	0,38
15 dB	57,55	38,28	4,16	14,81	16,92	1,0032	0,39
10 dB	57,36	37,90	4,73	16,15	18,26	1,0042	0,39
5 dB	56,86	36,96	6,17	19,3	21,50	1,0062	0,40
0 dB	54,44	36,32	9,25	26,12	28,58	1,0127	0,40
-5 dB	46,75	37,18	16,07	39,48	42,81	1,0342	0,44

Tabelle A.1.: Ergebnisse der Verschlussmomentschätzung. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. DYPSA in Kombination mit der Grundfrequenzerkennung des Harmonic Frontends von Behnke dienten als Referenz-Algorithmus [Beh, NKGB07]. Die Ergebnisse der Grundfrequenzerkennung finden sich in Tabelle 6.2. Der Test-Algorithmus verwendet als Parameter $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot k_{10}$, $p = 0.17 \cdot \pi$ und $u = 9$.

(a) Test Set A

SNR/dB	Subway	Babble	Car	Exhibition	Average
20 dB	98,71	98,1	98,12	98,06	98,25
15 dB	98,1	97,64	97,76	97,53	97,76
10 dB	96,04	95,77	96	95,46	95,82
5 dB	91,53	89,57	91,71	89,69	90,63
0 dB	77,1	67,35	77,3	72,11	73,47
-5 dB	39,7	25,42	30,45	36,89	32,12
Average	92,3	89,69	92,18	90,57	91,18

(b) Test Set B

SNR/dB	Restaurant	Street	Airport	Train Station	Average
20 dB	98,5	98,28	98,27	98,36	98,35
15 dB	97,36	97,19	97,44	97,28	97,32
10 dB	94,66	94,98	94,75	93,89	94,57
5 dB	84,83	89,84	87,35	86,52	87,14
0 dB	57,88	70,34	68,39	67,88	66,12
-5 dB	23,4	36,43	30,12	29,1	29,76
Average	86,65	90,13	89,24	88,79	88,7

Tabelle A.2.: Aurora-2 Ergebnisse. Wort-Akuratheit in Prozent für das Test Set A und B der Aurora-2 Datenbank bei Multi-Condition Training. Als Parameter wurden $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot c_B$ und $p = \pi$ und $u = 9$ verwendet. Es wurde eine geringere Insertion Probability gewählt ($p = -30$), um Insertion Errors durch Synchronisation mit dem Hintergrund auszugleichen.

(a) $p = 0.17\pi$

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
CLEAN	13,15	1,60	1,24	1,86
20 dB	13,41	1,49	1,29	1,88
15 dB	14,17	1,36	1,26	1,88
10 dB	15,55	1,26	1,4	1,97
5 dB	18,94	1,31	1,86	2,18
0 dB	26,10	1,98	3,29	2,7
-5 dB	40,27	5,42	8,93	4,72

Tabelle A.3.: Ergebnisse der Grundfrequenzschätzung. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank. Der Test-Algorithmus verwendet die Parameter $C = 9$, $B = 10$, $c = 1 \cdot k_{10}$, $c_2 = 0.6 \cdot k_{10}$, $u = 9$ und $p = 0.17\pi$.

(a) Vorverarbeitung Verschlussmomente

„Subway “	GDR (%)	FER (%)	GER (%)	MVR (%)	MR (%)	MPER (%)	RMSE (ms)
CLEAN	94,94	4,32	0,1	35,57	36,94	1,0034	0,16
0 dB	54,44	36,32	9,25	26,12	28,58	1,0127	0,40

Tabelle A.4.: Ergebnisse der Verschlussmomentschätzung mit Vorverarbeitung. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank für klare Sprache und bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. DYPSA in Kombination mit der Grundfrequenzerkennung des Harmonic Frontends von Behnke dienten als Referenz-Algorithmus [Beh, NKGB07]. Die Ergebnisse der Grundfrequenzerkennung finden sich in Tabelle A.5. Der Test-Algorithmus verwendet als Parameter $C = 9$, $B = 10$, $c = 1 \cdot c_B$, $c_2 = 0.6 \cdot k_{10}$, $p = 0 \cdot \pi$ und $u = 9$. Zusätzlich wurde die gleiche Vorverarbeitung wie bei DYPSA durchgeführt.

(a) Vorverarbeitung Grundfrequenz

„Subway“	VER(%)	UER(%)	GPER(%)	RMSE(Hz)
CLEAN	34,52	0,94	1,5	2,01
0 dB	29,84	9,68	10,92	5,09

Tabelle A.5.: Ergebnisse der Grundfrequenzschätzung mit Vorverarbeitung. Die Ergebnisse wurden für das Subset 1 des Test Sets A der Aurora-2 Datenbank für klare Sprache und bei einem Signal-Rausch-Verhältnis von 0 dB gemessen. Der Test-Algorithmus verwendet die Parameter $C = 9$, $B = 10$, $c = 1 \cdot k_{10}$, $c_2 = 0.6 \cdot k_{10}$, $u = 9$ und $p = 0\pi$. Zusätzlich wurde die gleiche Vorverarbeitung wie bei DYPSA durchgeführt.

Literaturverzeichnis

[AAC99] Anshu Agarwal, , Anshu Agarwal, and Yan Ming Cheng. Two-stage mel-warped wiener filter for robust speech recognition. In *Proc. ASRU*, pages 12–15, 1999.

[ASD99] Nazih Abu-Shikhah and Mohamed Deriche. A novel pitch estimation technique using the teager energy function. In *Proceedings of the Fifth International Symposium on Signal Processing and Its Applications*, volume 1, 1999.

[Aur] http://aurora.hsnr.de.

[BDMD+06] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi, and C. Wellekens. Impact of variabilities on speech recognition. In *SPECOM'2006*, pages 3–16, Saint-Petersburg, Russia, June 2006.

[Beh] Sven Behnke. Personal communications.

[Beh03] Sven Behnke. Report on the 2003 workshop on neuromorphic engineering p.37-43, 2003.

[BNG06] M. Brookes, P. A. Naylor, and J. Gudnason. A quantitative assessment of group delay methods for identifying glottal closures in voiced speech. *IEEE Transactions on Audio, Speech & Language Processing*, 14(2):456–466, 2006.

[Bol79] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, volume 29, 1979.

[Bro07] M. Brookes. Voicebox: Speech processing toolbox for matlab. http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html, 2007.

[CM03] Michael Clausen and Meinard Müller. Basic concepts of digital signal processing, 2003.

[DO03] Li Deng and Douglas O'Shaughnessy. *Speech processing: a dynamic and optimization-oriented approach.* Marcel Dekker Publishers, 2003.

[DP04] Zhu Donglai and K. K. Paliwal. Product of power spectrum and group delay function for speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP '04).*, 2004.

[EKP01] Douglas Ealey, Holly Kelleher, and David Pearce. Harmonic tunnelling: Tracking non-stationary noises during speech. *Eurospeech 2001*, 2001.

[Eur03a] European Telecommunications Standards Institute (ETSI). *Standard ES 201 108 V1.1.3 „Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms“*, 2003.

[Eur03b] European Telecommunications Standards Institute (ETSI). *Standard ES 202 050 V1.1.3 „Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms“*, 2003.

[Eur07] European Telecommunications Standards Institute (ETSI). *ETSI ES 202 050 V1.1.5 „Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms“*, 2007.

[Fan70] Gunnar Fant. *Acoustic Theory of Speech Production*. Walter de Gruyter, 1970.

[Fla65] J.L. Flanagan. *Speech analysis, synthesis and perception*. Springer, 1965.

[FLL85] G. Fant, J. Liljencrants, and Q. Lin. A four-parameter model of glottal flow. *STL-QPSR*, 20, 1985.

[Fra75] Ronald H. Frazier. Rasta processing of speech. Master's thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 1975.

[Fuj60] H. Fujisaki. Automatic extraction of fundamental period of speech by auto-correlation analysis and peak detection. *J. Acoust. Soc. Am. Volume*, 32, 1960.

[GR00] Randy Goldberg and Lance Riek. *A practical handbook of speech coders*. CRC Press, 2000.

[Gre95] Steven Greenberg. The ears have it: The auditory basis of speech perception. In *in the Proc. of the ICPhS*, pages 34–41, 1995.

[HM94] Hynek Hermansky and Nelson Morgan. Rasta processing of speech. *IEEE TRANSACTIONS ON SPECH AND AUDIO PROCESSING*, 2, 1994.

[It96] U. It. Itu recommendation g.712: Transmission performance characteristics of pulse code modulation channels. Recommendation, International Telecommunication Union, 1996.

[JJPM08] Niksa Jakovljevic, Marko Janev, Darko Pekar, and Dragisa Miskovic. Energy normalization in automatic speech recognition. In Petr Sojka, Ales Horák, Ivan Kopecek, and Karel Pala, editors, *TSD*, volume 5246 of *Lecture Notes in Computer Science*, pages 341–347. Springer, 2008.

[JM08] Daniel Jurafsky and James H. Martin. *Speech and Language Processing.* Prentice Hall, 2008.

[JS01] Philip J. B. Jackson and Christine H. Shadle. Pitch-scaled estimation of simultaneous voiced and turbulence-noise components in speech. *IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING*, 9, 2001.

[KNB02] Anastasis Kounoudes, Patrick A. Naylor, and Mike Brookes. The dypsa algorithm for estimation of glottal closure instants in voiced speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 2002.

[LB92] P. Lockwood and J. Boudy. Experiments with a nonlinear spectral subtractor (nss), hidden markov models and the projection, for robust speech recognition in cars. *Eurospeech 1991*, 11, 1992.

[LL08] Yang Lu and Philipos C. Loizou. A geometric approach to spectral subtraction. *Speech Communication*, 50(6):453–466, 2008.

[LO79] J.S. Lim and A.V. Oppenheim. Enhancement and bandwidth compression of noisy speech. In *Proceedings of the IEEE*, volume 67, 1979.

[Lyo04] Richard G. Lyons. *Understanding Digital Signal Processing.* Prentice-Hall, 2004.

[MBFS88] Hiroshi Muta, Thomas Baer, Hiroyuki Fukuda, and Shigeji Saito. A pitch-synchronous analysis of hoarseness in speech. *J. Acoust. Soc. Am.*, 33, 1988.

[MC01] D. Macho and Yan Ming Cheng. Snr-dependent waveform processing for improving the robustness of asr front-end. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 2001.

[MMY89] H. A. Murthy, K. V. Madhu Murthy, and B. Yegnanarayana. Formant extraction from phase using weighted group delay function. In *Electronics Letters*, volume 25, 1989.

[MYC91] Y. Medan, E. Yair, and D. Chazan. Super resolution pitch determination of speech signals. *IEEE Transactions on Signal Processing*, 39, 1991.

[NKGB07] Patrick A. Naylor, Anastasis Kounoudes, Jon Gudnason, and Mike Brookes. Estimation of glottal closure instants in voiced speech using the dypsa algorithm. *IEEE Transactions on Audio, Speech & Language Processing*, 15(1):34–43, 2007.

[Nol67] A.M. Noll. Cepstrum pitch determination. *J. Acoust. Soc. Am.*, 41, 1967.

[OS75] A.V. Oppenheim and R.W. Schaefer. *Digital Signal Processing.* Prentice-Hall, 1975.

[PgHG00] David Pearce, Hans günter Hirsch, and Ericsson Eurolab Deutschland Gmbh. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *in ISCA ITRW ASR2000*, pages 29–32, 2000.

[PK08] Beat Pfister and Tobias Kaufmann. *Sprachverarbeitung, Grundlagen und Methoden der Sprachsynthese und Spracherkennung.* Springer, 2008.

[PM95] Bernd Pompino-Marschall. *Einführung in die Phonetik.* de Gruyter, 1995.

[PN91] Magnus Petursson and Joachim Neppert. *Elementarbuch der Phonetik.* Helmut Buske Verlag Hamburg, 1991.

[RBB07] Sergio Roa, Maren Bennewitz, and Sven Behnke. Fundamental frequency estimation based on pitch-scaled harmonic filtering. In *IEEE conference Acoustics, Speech and Signal Processing*, volume 9, 2007.

[RSC+74] Myron J. Ross, Harry L. Schaffer, Andrew Cohen, Richard Freudberg, and Harold J. Manley. Average magnitude difference function pitch extractor. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 21, 1974.

[SKD97] J. Anthony Seikel, Douglas W. King, and David G. Drumright. *Anatomy and Physiology for speech, language and hearing.* Singular Publishing Group, 1997.

[ST95] E.G. Schukat-Talamazzini. *Automatische Spracherkennung.* Vieweg, 1995.

[Ste98] Kenneth N. Stevens. *Acoustic Phonetics.* The MIT Press, 1998.

[Tal95] D. Talkin. A robust algorithm for pitch tracking. 1995.

[WE94] Robert K. Whitman and Delores M. Etter. An investigation of estimating pitch periods using a non-linear differential operator. In *Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 2, 1994.

[Wie66] N. Wiener. *Extrapolation Interpolation and Smoothing of Stationary Time Series*. M.I.T: Press, 1966.

[YM99] B. Yegnanarayana and Hema A. Murthy. Robustness of group-delay-based method for extraction ofsignificant instants of excitation from speech signals. *IEEE Transactions on Speech and Audio Processing*, 199.

[YS95] B. Yegnanarayana and R.L.H.M. Smits. A robust method for determining instants of major excitations in speech signals. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 1995.

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum Unterschrift