Masterarbeit

Erstellung einer Sprachdatenbank sowie eines Programms zu deren Analyse im Kontext einer Sprachsynthese mit spektralen Modellen

zur Erlangung des akademischen Grades
Master of Science

vorgelegt dem
Fachbereich Mathematik, Naturwissenschaften und Informatik
der Technischen Hochschule Mittelhessen

Tobias Platen
im August 2014

Referent: Prof. Dr. Erdmuthe Meyer zu Bexten
Korreferent: Prof. Dr. Keywan Sohrabi

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.
Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Inhaltsverzeichnis

Downloads: Die im Rahmen dieser Arbeit erstellten Sprachdatenbank und dem Quellcode in C++ zur Analyse von Sprachdaten kann unter http://platen-software.de/voice_synthesis heruntergeladen werden.

Zusammenfassung

Die menschliche Stimme vollständig zu modellieren und zu imitieren ist eine nur teilweise gelöste Herausforderung. In den letzten Jahren gab es dazu fachübergreifende Forschungen im medizinischen, nachrichtentechnischen und musikalischen Bereich, um die Funktion der Stimme zu verstehen. Sprachsynthese und musikalische Sound-Synthese wurden traditionell als getrennt behandelt, heute fasst man beides zusammen.

Das Ziel dieser Arbeit ist die Entwicklung eines universellen Stimmensynthesizers auf Basis einer Sprachdatenbank, welcher Sprache, aber auch Gesang, synthetisieren kann. Um Mehrsprachigkeit und einen natürlichen Klang zu erreichen, ist es notwendig, dass möglichst viele Schritte in der Erstellung der Sprachdatenbank automatisiert werden können. Insbesondere soll die Erstellung einer Sprachdatenbank ohne medizinische Geräte, wie z.B. einem Laryngographen möglich sein.

Nach einer Einführung in die Sprachsynthese werden moderne Algorithmen zur spektralen Analyse und Nachbearbeitung von Sprache vorgestellt. Da es sich bei Sprachsignalen um harmonische Signale handelt, wird ein Verfahren zur Bestimmung der harmonischen Frequenzen und Phasen behandelt.

Zur Darstellung von harmonischen Signalen gibt es zwei Interpretationen: die Modellierung von Frequenzkomponenten und die Modellierung von Stimmenpulsen. Letztere führt zu bekannten Algorithmen wie TD-PSOLA und Voice-Pulse-Modelling (VPM). Moderne spektrale Modelle wie z.B. EpR basieren auf beiden Darstellungen eines harmonischen Signals. Ein weiteres Merkmal dieser Modelle ist die Trennung des Anregungssignals vom Beitrag des Vokaltraktes.

Zum Schluss wird noch die Anwendbarkeit dieser Modelle für konkatenative Sprachsynthese behandelt. Dies beinhaltet auch das Sampling der Stimme und die teilweise automatisierte Erstellung einer Sprachdatenbank. Die dafür benötigte, in C++ geschriebene Software ist ebenfalls Teil dieser Masterarbeit.

Notationen

Variablen

h Harmonische

t, x, y kontinuierliche Variablen im Zeitbereich

f kontinuierliche Variable im Frequenzbereich

φ kontinuierlicher Phasenwert

ω Kreisfrequenz

`freqPerBin` frequency per bin = `fs`/`fftSize`

`expct` erwarteter Phasenunterschied = $2\pi \cdot int($`fftSize`/`osamp`$)/$`fftSize`

`osamp` Überabtastung

`fftSize` Größe der schnellen Fouriertransformation (ist immer eine Zweierpotenz)

n diskrete Variable im Zeitbereich

k diskrete Variable im Frequenzbereich

f_s Samplerate in Hz

f_0 fundamentale Frequenz

Abkürzungen

FT Fourier Transform

FFT Fast Fourier Transform

STFT Short-Time Fourier Transform

MFCC Mel Frequency Cepstral Coefficients

MFPA Maximally Flat Phase Alignment

EpR Excitation plus Resonances

NBVPM Narrow-Band Voice Pulse Modeling

WBVPM Wide-Band Voice Pulse Modeling

1 Einführung

1.1 Motivation

Seit dem Erfolg des rein digitalen Popstars Hatsune Miku [14] steigt das Interesse an der Synthese der menschlichen Stimme. Für die Transformation von Gesang entwickelte Algorithmen lassen sich häufig auch für die Synthese von besonders natürlich klingender Sprache einsetzen, weil die zugrunde liegenden Modelle sehr ähnlich sind. Moderne Algorithmen wie z.B. STRAIGHT ermöglichen so die Konvertierung von Sprache nach Gesang, welcher verglichen mit anderen Synthesizern sehr natürlich klingt. [2]

Viele ältere Sprachausgaben verwenden einfache spektrale Modelle, wie z.B. die Formantsynthese, da man hier mit relativ kleinen Sprachdatenbanken verständliche Sprache produzieren kann. Dabei werden nur die zur Verständlichkeit benötigten Informationen gespeichert, sprecherspezifische Merkmale fehlen. Dies hat zur Folge, dass die Sprache recht unnatürlich klingt. Bei moderneren Sprachausgaben wird immer häufiger auf Sampling zurückgegriffen, da man damit potentiell einen natürlicheren Klang erreichen kann. Da heutige Rechner über mehr Arbeitsspeicher und Festplattenplatz verfügen, kann man auch größere Sprachdatenbanken verwenden.

Häufig wird das Sampling mit spektralen Modellen kombiniert, da dies eine Reihe von Vorteilen wie z.B. mehr Flexibilität hat. Für die Synthese der menschlichen Stimme wird dabei in der Regel auf einen Corpus zurückgegriffen. Dieser besteht häufig aus Diphones, welche auch die Koartikulationseffekte enthalten. Bei der Formantsynthese wird die Koartikulation dagegen häufig vernachlässigt, was zu einem unnatürlichen Klang führt.

Da die menschliche Stimme ein kontinuierliches Anregungssignal hat, ist das Sampling hier schwieriger als z.B. bei einem Tasteninstrument. Um das Anregungssignal zu extrahieren, wurden in den letzten Jahren spezielle Algorithmen wie z.B PLATINUM [22] entwickelt. Damit ist ein - im Vergleich zu einem Formantsynthesizer oder Vocoder - natürlicher Klang möglich. PLATINUM bietet, abgesehen vom natürlichen Klang, Vorteile gegenüber STRAIGHT und dem bekannten TD-PSOLA Algorithmus.

Freie Sprachausgaben wie z.B. Festival und eSpeak klingen weniger natürlich als dies mit modernen Algorithmen möglich ist. Beide Sprachausgaben sind durch Erweiterungen in der Lage, Gesang zu synthetisieren. Zusammen mit GNU Lilypond ist ein barrierefreier Notensatz [28] möglich. Weitere freie Sprachausgaben, welche für die Musikproduktion entwickelt wurden, sind v.Connect-STAND [6] und Sinsy [5]. Beide sind jedoch auf die japanische Sprache beschränkt, welche mit etwa 50 möglichen offenen Silben und

dem Endkonsonanten [[N]] nur sehr wenige Phonemkombinationen enthält. Damit ist es Hobbysängern möglich, die eigene Stimme zu samplen und für die Musikproduktion zu benutzen. Teilweise wird auch ein recht natürlicher Klang erreicht.

Die Erstellung einer Sprachdatenbank für die deutsche Sprache ist um einiges schwieriger, da es in der deutschen Sprache etwa um die 1800 Diphones gibt, für Englisch sind es etwa 1200 Diphones . Bisher gibt es keine deutsche Sprachdatenbank für die konkatenative Festival-Sprachsynthese. Daher soll die Erstellung einer solchen Sprachdatenbank im Rahmen dieser Arbeit durchgeführt werden. Nur auf diese Weise lässt sich die Sprachqualität von eSpeak verbessern. Nach demselben Prinzip können dann für andere Sprachen Sprachdatenbanken erstellt werden.

1.2 Ziele

Ziel dieser Arbeit ist es, moderne Algorithmen zur Sprachsynthese zu evaluieren und damit den Sprachsynthesizer eSpeak [1], welcher von den beiden Screenreadern NVDA und Orca verwendet wird, zu verbessern. Dabei soll die Sprachsynthese möglichst natürlich und realistisch klingen. Dies ist eine sehr schwierige Herausforderung, da viele Probleme nur teilweise oder nicht zufriedenstellend gelöst sind.

Für den Sprachsynthesizer der neuen Generation soll konkatenative Synthese verwendet werden, da dies zu einem potentiell besseren Klang führt. Dies impliziert, dass Samples einer Sprachaufnahme transformiert werden können, ohne dass sich die Qualität verschlechtert und die Übergänge der Konkatenation weich sind. Dafür sind Modelle zur Beschreibung der Sprachdaten notwendig. Insbesondere spektrale Modelle bieten interessante Vorteile gegenüber produktiven Modellen, welche für die Sprachsynthese nutzbar gemacht werden sollen.

Gegenstand dieser Masterarbeit ist ein Modell, das das Anregungssignal vom Beitrag des Vokaltrakts trennt. Dadurch erhält man mehr Flexibilität und kann Optimierungen durchführen, die den Rechenaufwand bei der Synthese senken. Für einen natürlichen Klang ist auch ein geeignetes Phasenmodell notwendig, welches ebenfalls in dieser Arbeit behandelt wird.

Die zeitliche Auflösung von Algorithmen im Zeitbereich wie TD-PSOLA (Time-Domain Pitch Synchronous OverlapAdd) soll mit den Vorteilen eines spektralen Modells kombiniert werden, während die Nachteile der Time-Domain Algorithmen vermieden werden. Damit kann man die Position eines einzelnen Pulses und dessen harmonische Amplituden unabhängig voneinander bearbeiten. Einzelne Transformationen sind nun unabhängig voneinander durchführbar, ohne dass sich die Qualität dabei verschlechtert.

Da konkatenative Sprachsynthese mit menschlichen Sprachaufnahmen arbeitet, ist es von entscheidender Bedeutung, wie diese Sprachaufnahmen erstellt werden und welche Segmente konkateniert werden. In der Praxis hat sich dafür die Diphone Synthese be-

währt. In der Regel wird dafür ein künstlicher Speech Corpus verwendet, welcher im Gegensatz zur natürlichen Sprache möglichst wenig Redundanzen enthält. Das Erstellen der Datenbank ist sehr aufwändig und erfordert viele manuelle Schritte. Um für möglichst viele Sprachen Stimmdatenbanken erstellen zu können ist es wichtig, möglichst viele Schritte automatisieren zu können. Die Gesamtlänge der Sprachaufnahmen soll dabei möglichst gering gehalten werden, um den Sprecher zu entlasten. Gleichzeitig muss jedoch die Menge der Aufnahmen alle Phoneme einer Sprache abdecken.

Die Komprimierung der Sprachdaten spielt ebenfalls eine wichtige Rolle. Daher sollen im Laufe dieser Arbeit auch Algorithmen zur Sprachdatenkomprimierung behandelt werden. Insbesondere auf mobilen Geräten ist dies sehr wichtig, da hier weniger Ressourcen gegenüber einem Desktop-PC zur Verfügung stehen. Dabei muss berücksichtigt werden, dass die Dekomprimierung die CPU nicht zu stark beansprucht.

Es soll gezeigt werden, dass konkatenative Synthese gegenüber der vorher verwendeten Formantsynthese eine Verbesserung der Klangqualität bringt. Hörtests, aber auch spektrale Ansichten, sind dabei unerlässlich. Aus der spektralen Ansicht kann man z.B. ablesen, warum eSpeak so unnatürlich klingt. Durch kontinuierliche Hörtests lassen sich auch Rückschlüsse auf Modelle ziehen, die so kontinuierlich verbessert werden können.

Zuletzt soll noch der Beitrag von Masanori Morise gewürdigt werden. Dessen Software WORLD und die dazugehörige Dokumentation sind unter freien Lizenzen verfügbar, was deren Verwendung sowohl für akademische Projekte als auch für den alltäglichen Produktiveinsatz ohne Einschränkungen nutzbar macht. Im Laufe dieser Arbeit entstandene Software und auch die Speech-Corpera werden ebenfalls unter freien Lizenzen veröffentlicht, damit sie für Anwender zugänglich sind und zur weiteren Forschung zur Verfügung stehen.

1.3 Historische Sprachsynthesen

Die älteste Apparatur zur Sprachsynthese ist Wolfgang von Kempelens Sprechmaschine aus dem Jahr 1769. Mit dem Aufkommen der Elektrotechnik Anfang des zwanzigsten Jahrhunderts wurde dann der Vocoder und der Voder erfunden. Der Vocoder diente ursprünglich der Verschlüsselung von Sprache, mit dem Voder lässt sich Sprache synthetisieren. Beide Geräte wurden von Homer Dudley um etwa 1930 erfunden. Sowohl die Sprechmaschine als auch der Voder benötigen einen menschlichen Operator, es ist damit also kein automatisches Text-To-Speech möglich. In den 1960ern machten es weitere Fortschritte in der Nachrichtentechnik möglich, mit digitalen Bausteinen Sprache zu synthetisieren.

1.3.1 Die Sprechmaschine

Die Sprechmaschine ist eine mechanische Nachbildung der menschlichen Sprechorgane. Ein Blasebalg erzeugt einen Luftstrom, welcher durch ein Rohrblatt in einzelne Pulse zerhackt wird. Durch den Bernoulli-Effekt entsteht ein Unterdruck, welcher das Rohrblatt gegen die Halterung drückt. Durch die Federwirkung öffnet sich das Rohrblatt erneut. Wenn sich genügend Luftdruck aufgebaut hat, schließt sich die Glottis erneut und der Vorgang beginnt von vorne. Der Luftstrom, der aus der Glottis kommt, wird als Voice-Source bezeichnet und ist reich an Obertönen. Der Vokaltrakt bestehend aus Mund und Nase wird durch einen Gummitrichter und zwei Nasenröhren nachgebildet. Durch Verformung des Vokaltraktes und Zuhalten von Mund oder Nase lassen sich unterschiedliche Laute artikulieren. Einige Laute lassen sich nicht artikulieren, da eine Zunge fehlt. Der Vokaltrakt wirkt als Filter und verstärkt einige Frequenzen (Formanten), während andere Frequenzen durch die Nase abgeschwächt werden (Antiformanten). Das Vokaltrakt-Filter wird daher auch als Formantfilter bezeichnet. Moderne digitale Sprachsynthesizer arbeiten auch nach diesem Source-Filter Modell. Kempelen erkannte damals schon die Bedeutung der Koartikulation bei seiner Spechmaschine, während Kratzensteins Verfahren, basierend auf mit speziellen Resonatoren versehene durchschlagende Zungenpfeifen, nur 5 Monophthonge erzeugen kann. [8]

1.3.2 Der Vocoder und der Voder

Der Vocoder ist eine elektronische Vorrichtung, welche ein Sprachsignal in einzelne überlappende Bänder teilt. Das Signal wird durch eine Filterbank aus Bandpassfiltern aufgespalten und durch Envelope-Follower wird die Amplitude des Signals ermittelt. Durch die Vertauschung der Bänder lässt sich ein Sprachsignal verschlüsseln. Auf der anderen Seite wird durch die Verwendung der Umkehrfunktion die ursprüngliche Hüllkurve wiederhergestellt. Als Carrier kann weißes Rauschen oder ein beliebiges harmonisches oder polyphones Signal verwendet werden. Eine Weiterentwicklung des Vocoders ist der Voder. Hier wird die Hüllkurve von Hand eingestellt. Das Carrier oder Anregungssignal ist entweder ein Rauschen (für stimmlose Laute) oder ein periodisches Signal, das von einem Oszillator erzeugt wird. Die Frequenz des Oszillators bestimmt die Tonhöhe. Mit dem Voder ließ sich erstmals verständliche Sprache synthetisieren, hier fehlen keine Laute. Moderne computerimplementierte Sprachsynthesizer basieren auf der diskreten Variante des Vocoders. Da deren Sprachqualität oft schlecht ist, werden im Laufe dieser Arbeit Wege zur Verbesserung aufgezeigt.

1.3.3 Linear Predictive Coding

Bei LPC wird die Impulsantwort des Vokaltrakts durch ein All-Pole Filter approximiert. Man erhält sowohl die fundamentale Frequenz als auch die spektrale Hüllkurve aus den Werten der Autokorrelation. Ein Gerät, welches nach diesem Verfahren arbeitet, ist das

Speak & Spell von Texas Instruments. Dabei wird eine sehr kleine Sprachdatenbank in einem ROM verwendet. Aufgrund eines stark vereinfachten Modells klingt das Gerät sehr unnatürlich, die Sprache ist aber verständlich.

1.4 Moderne Algorithmen zur Sprachsynthese

Zur Sprachsynthese wurden in den letzten Jahrzehnten verschiedene Algorithmen und Modelle entwickelt. Bei den Algorithmen wird zwischen Time-Domain und Spectral-Domain unterschieden, wobei es auch Algorithmen gibt, welche die Vorteile von Time-Domain Algorithmen mit denen von spektralen Modellen kombinieren. Daneben gibt es noch physikalische Modelle, welche auch heute noch von artikulatorischen Sprachsynthesizern verwendet werden. Spektrale Modelle sind für diese Arbeit besonders wichtig, da sie ähnlich funktionieren wie das menschliche Gehör, welches Tonhöhe und Klangfarbe getrennt wahrnehmen kann. Die Tonhöhe wird durch die fundamentale Frequenz f_0 bestimmt, während die Klangfarbe durch die vom Resonanzkörper verstärkten Frequenzen (Formanten) bestimmt wird. Bei menschlicher Sprache sind nur die ersten zwei Formanten für die Verständlichkeit notwendig, die weiteren Formanten bestimmen den charakteristischen Klang der Stimme.

1.4.1 Formantsynthese

Formantsynthese ist ein Verfahren, das auf einem vereinfachten spektralen Modell basiert, mit dem sich Sprache synthetisieren lässt. Für jeden Vokal und für stimmhafte Konsonaten werden die Frequenzen und Bandbreiten der Formanten aus einer Tabelle entnommen. Dann wird ein harmonisches Anregungssignal entweder durch eine Filterbank oder eine Kaskade gefiltert, so dass die Formanten im Ausgangssignal hörbar sind. Die Formanttabelle ist im Vergleich zu Sprachdatenbanken anderer Sprachsynthesizer sehr klein, daher wird Formantsynthese oft auf weniger leistungsfähigen Computern eingesetzt. Stimmlose Laute können durch moduliertes Rauschen sehr einfach erzeugt werden. Formantsynthese basiert auf der vereinfachenden Annahme, dass ein Anregungssignal durch ein lineares Filter gefiltert wird. In der Realität gibt es jedoch eine Kopplung zwischen der Quelle und dem Vokaltrakt , die Filterkette ist also nicht linear. Die Koartikulation wird auch sehr häufig vernachlässigt. Ein häufiges Referenzmodell, welches auch in eSpeak verwendet wird, ist der Klatt-Synthesizer [19]. Der Vorteil der Formantsynthese gegenüber der konkatenativen Synthese ist die Flexibilität.

1.4.2 Konkatenative Synthese

Bei konkatenativer Synthese handelt es sich um ein auf Sprachaufnahmen basierendes Samplingverfahren. Vorhande Sprachaufnahmen werden aus einer Datenbank entnommen und aneinander angehängt. Dies hat den Vorteil, dass man einen sehr natürlichen Klang erhält. Nachteilig ist der größere Platzbedarf und der erhöhte Aufwand zum Erstellen einer solchen Datenbank. Aus diesem Grund wurden Algorithmen zur Sprachdatenkomprimierung und Werkzeuge zum halbautomatischen Erstellen der Sprachdatenbanken entwickelt. Die meisten modernen Sprachsynthesizer arbeiten nach diesem Prinzip. Konkatenative Synthese setzt voraus, dass man ein Sprachsignal in Tonhöhe, Länge und Klangfarbe verändern kann, ohne dass sich die Qualität merklich ändert. Dies wird auch als Prosody-Matching bezeichnet. Konkatenative Synthese ist sowohl im Zeitbereich als auch im Frequenzbereich möglich. Ein Beispiel für einen konkatenativen Sprachsynthesizer, welcher die Sprachdaten mittels Residual-LPC komprimiert und für das Prosody-Matching aufbereitet, ist das Festival Speech Synthesis System. Auch eSpeak kann zur Ansteuerung eines externen konkatenativen Sprachsynthesizers verwendet werden.

2 Spektrale Modelle zur Sprachsynthese

2.1 Faltung, Fouriertransformation und Vocoder

Zur Sprachsynthese werden oft spektrale Modelle verwendet, da sich so die einzelnen Frequenzen unabhängig voneinander bearbeiten lassen. Zur Überführung eines diskreten Signals, welches z.B. von einem Mikrofon aufgenommen und digitalisiert wurde, wird die schnelle Fouriertransformation (FFT) verwendet. Dabei erhält man zwei Spektren: das Amplitudenspektrum und das Phasenspektrum. Die Fouriertransformation kann als Filterbank interpretiert werden, wobei die Anzahl der Bins der Anzahl der Bänder entspricht. Durch die Envelope-Follower des Vocoders (siehe Abb. 2.1) wird die Phaseninformation aus dem Eingangssignal entfernt. Im Amplitudenspektrum ist alle für die Verständlichkeit benötigte Information enthalten, das Phasenspektrum braucht aus diesem Grund nicht übertragen werden. Moduliert man ein weißes Rauschen mit den Amplituden der Formantfrequenzen , so erhält man ein verständliches Sprachsignal, welches jedoch noch das Rauschen enthält und daher unnatürlich klingt. Für einen natürlichen Klang ist daher auch die Phase des Anregungssignals von Bedeutung. Ein Phase-Vocoder [9] ist eine digitale Implementierung eines spektralen Modells, welches auf dem Vocoder basiert.

Eine andere Interpretation der Fouriertransformation ist die spektrale Faltung. Mit dieser Interpretation lässt sich auch das Anregungssignal (Carrier) modellieren, welches für einen natürlichen Klang von Bedeutung ist. Die Faltung ist im Frequenzbereich viel schneller zu berechnen, da dies der Multiplikation eines komplexen Vektors entspricht. Im Zeitbereich sind deutlich mehr Multiplikationen und Additionen zu berechnen, dies entspricht dann einer gewichteten Addition. Die dafür verwendete imaginäre Exponentialfunktion ist eine Eigenfunktion der Faltung [16, Seite 229]. Die Fouriertransformation, bei der es sich um eine Integraltransformation handelt, wird folgendermaßen dargestellt:

Faltung: $g(y) = \int_{-\infty}^{+\infty} f(x)h(y-x)dx$

Fouriertransformation: $f(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x)e^{-i\omega x}dx$

Dabei ist ω die Kreisfrequenz, x und y sind Variablen im Zeitbereich. Um die Faltung im Frequenzbereich zu berechnen wird $h(y-x)$ durch die komplexe Exponentialfunktion ersetzt und ein konstanter Faktor eingefügt. [20]

Bei der diskreten Fouriertransformation wird das unendliche Integral duch eine endliche Summe ersetzt, so dass die Berechnung praktisch durchführbar wird. Dabei erhält man die Faltung des Signals X mit der Transformation des Analysefensters W [26]. Um die Faltung schnell zu berechnen, stellt SciPy die Funktion `fftconvolve` bereit. Bei der diskreten Fouriertransformation wird der Term $\frac{1}{\sqrt{2\pi}}$ durch die Anzahl der Bins ersetzt.

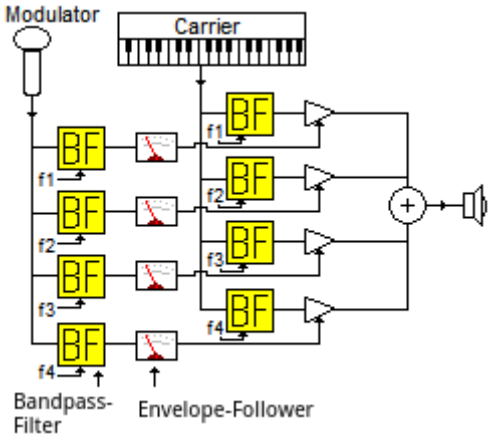

Abbildung 2.1: Darstellung eines aus einer analogen Filterbank bestehenden Vocoders. Das Ausgangssignal ist die Faltung von Modulator und Carrier. Ein Vocoder führt die Faltung bandweise durch und addiert die einzelnen Bänder. Der Vocoder wird häufig als Effektgerät in der Musik verwendet, damit lässt sich z.B. polyphoner Gesang erzeugen. [7](Audio [1])

2.2 Phase Vocoder

Mit dem Phase Vocoder [9] kann man die Tonhöhe und die Länge eines Signals unabhängig voneinander verändern. Da sich die Zusammensetzung eines Signals über die Zeit ändert, wird jeweils nur ein kurzes Zeitfenster mit der diskreten Fouriertransformation analysiert. Innerhalb eines kurzen Zeitfensters treten dagegen fast keine Änderungen im Frequenzbereich auf, das Signal ist hier quasi-stationär. Jedes Signal kann als eine Summe von Sinusfunktionen, deren Parameter sich mit der Zeit langsam ändern, dargestellt werden. Verschiebt man die Frequenzen, so kann man die Tonhöhe verändern, ohne dass sich die Dauer des Signals verändert. Die Frequenzauflösung ist abhängig von der Größe des Zeitfensters. Bei einem kleinen Zeitfenster ist die Frequenzauflösung entsprechend klein, man erhält jedoch eine größere zeitliche Auflösung. Umgekehrt erhält man bei einer größeren Frequenzauflösung eine kleinere zeitliche Auflösung. Alle messbaren Frequenzen sind ein Vielfaches der Inversen der Länge des zur Analyse genutzten

Fensters. Dies hat zur Folge, dass sich eine in der Realität vorkommende Frequenz auf mehrere benachbarte Bins verteilt. Dieses künstliche Frequenzraster ist also eine Folge der diskreten Darstellung und führt zum Leck-Effekt.

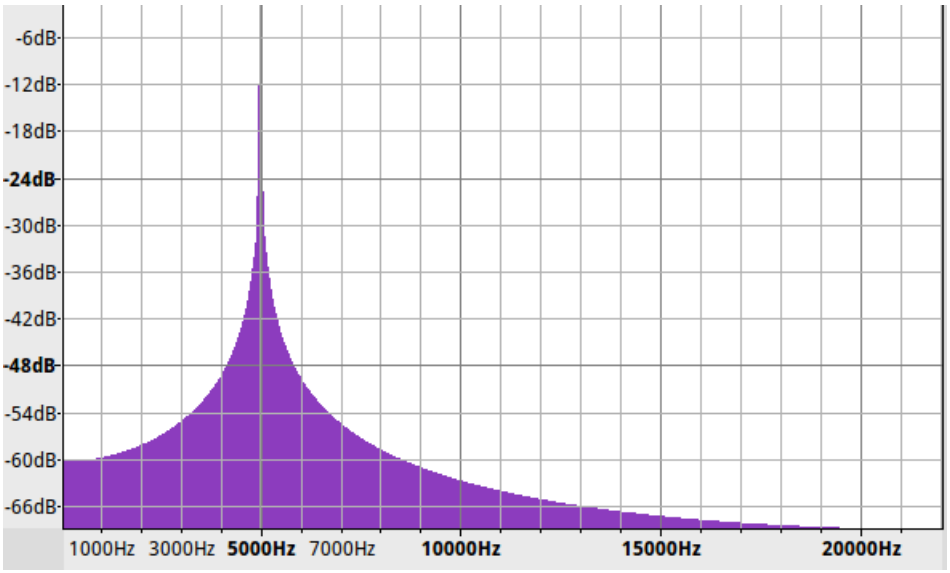

Abbildung 2.2: Leck-Effekt unter Verwendung eines Rechteck-Fensters

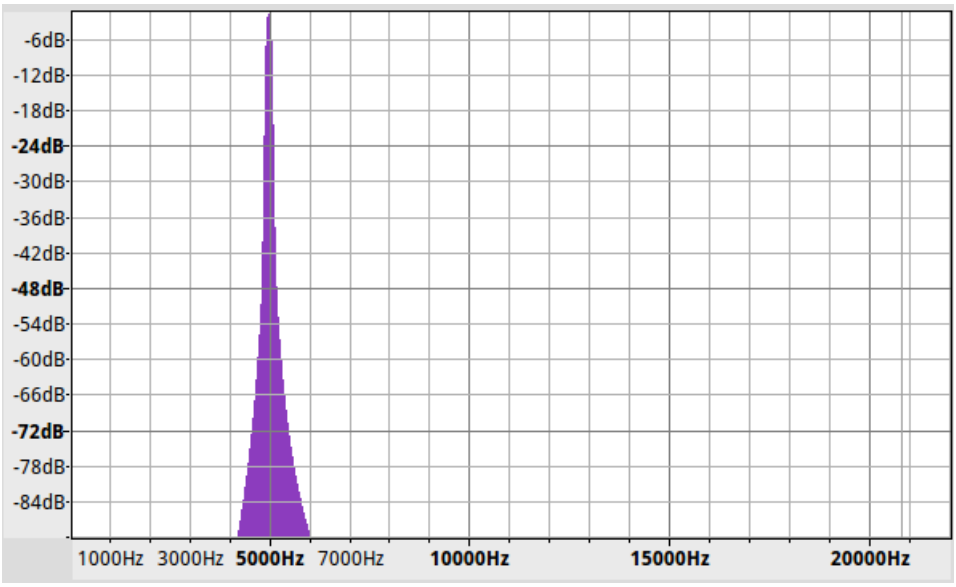

Abbildung 2.3: Leck-Effekt unter Verwendung eines Hanning-Fensters

Durch die Auswahl einer Fensterfunktion lassen sich die Unregelmäßigkeiten am Rand des Fensters verringern, die Frequenzen in den Sidelobes werden abgeschwächt und die Genauigkeit der Messungen wird verbessert. Eine häufig eingesetzte Fensterfunktion ist das Hanning-Fenster (auch als Von-Hann-Fenster bekannt). Die Fensterfunktion gewichtet die einzelnen Samples mit unterschiedlichen Werten. Im Zeitbereich werden Unre-

gelmäßigkeiten an den Rändern des Fensters weichgezeichnet, das gefensterte Signal hat dann an beiden Rändern den Wert 0.

Jeder Kanal der DFT wird durch Magnitude, Bin-Frequenz und Phase eindeutig beschrieben. Verändert man dessen Frequenz, so muss auch dessen Phase angepasst werden, damit es nicht zu Phasenauslöschungen (auch als Reverbance bezeichnet) kommt. Wenn die Frequenz eines Sinusoids nicht mit einer der messbaren Frequenzen identisch ist, dies ist in der Praxis niemals der Fall, so führt dies dazu, dass die gemessene Phase in jedem Frame einen anderen Wert hat. Jeder Frame hat dann einen eigenen Phase-Offset, welcher sich mit einem Phasenakkumulator berechnen lässt, wenn die Frequenz konstant ist. Verändert man die Frequenz eines Bins, so muss auch dessen Phase-Offset verändert werden. Damit verschieben sich auch die Phasen der folgenden Frames um denselben Betrag. Für das menschliche Ohr ist nur dieser Phase-Offset zwischen zwei Frames hörbar, jedoch nicht der Absolutwert der Phase im ersten Frame. Der zur Resynthese verwendete Phasenakkumulator kann daher mit Zufallswerten initialisiert werden. Die wahre Frequenz eines Sinusoids lässt sich nur mit dem Phase-Offset und dessen Bin-Frequency berechnen. Der Phase-Offset gibt die Abweichung von der festen Bin-Frequency an. Für die Resynthese wird dann der Phase-Offset aus der wahren Frequenz und der Bin-Frequency mittels eines Phasenakkumulators berechnet. Damit kann jeder Kanal als ein Frequenzband dargestellt werden, dessen Mittenfrequenz genau die Bin-Frequency ist. Da der Phase-Offset auf das Intervall $\pm\pi$ beschränkt ist, ist auch die Bandbreite eines Kanals beschränkt.

Die Fensterung führt jedoch zu einem weiteren Problem: Damit der Phase-Offset eindeutig interpretiert werden kann, muss mit überlappenden Frames gearbeitet werden. In der Regel wird mit einem Overlap-Factor von 4 oder größer gearbeitet, zwei benachbarte Frames überlappen sich daher um mindestens 75%. Bei kleineren Overlapfaktoren moduliert die Fensterfunktion das zu analysierende Signal, dies ist in (Audio [2]) deutlich hörbar. Die Auswirkungen des Overlap Factors auf die Bandbreite eines Kanals lassen sich einfach berechnen: $\texttt{bw} = \texttt{osamp} \cdot \texttt{freqPerBin}$. Da `osamp` ganzzahlig ist, ist auch die Bandbreite eines Kanals ein ganzzahliges Vielfaches der Raster-Frequenz. Durch einen höheren Overlap steigt auch die Bandbreite der Kanäle und der Hauptkeule, während die zeitliche Auflösung sinkt. [9]

Ein kleinerer Overlap führt dazu, dass die wahren Frequenzen nicht mehr richtig erkannt werden, da dann auch die Frequenzbänder nicht mehr überlappen und Frequenzen dazwischen mehrdeutig zugeordnet werden. In der Resynthese erhält man dann nahe beieinander liegende Frequenzkomponenten. Dies führt zu einer Schwebung und zu dem metallischen Klang eines Vocoders. Durch einen höheren Overlap (auch als Oversampling bezeichnet) kann das Problem gelöst werden, dabei steigt jedoch auch der Aufwand zur Berechnung. Diese Lösung wird auch als Phase-Locking bezeichnet. Dabei werden die Phasen und wahren Frequenzen der benachbarten Kanäle einander angeglichen, so dass keine Phasenauslöschungen mehr entstehen können. Für die Weiterverarbeitung sind nur die True-Frequency und die Magnitude von Bedeutung, die Phaseninformation braucht nicht übertragen zu werden.

Algorithmus 2.1 Berechnung der wahren Frequenz (aus [9])

```
/* compute magnitude and phase */
magn = 2.*sqrt(real*real + imag*imag);
phase = atan2(imag,real);
/* compute phase difference */
tmp = phase - gLastPhase[k];
gLastPhase[k] = phase;
/* subtract expected phase difference */
tmp -= (double)k*expct;
/* map delta phase into +/- Pi interval */
qpd = tmp/M_PI;
if (qpd >= 0) qpd += qpd&1;
else qpd -= qpd&1;
tmp -= M_PI*(double)qpd;
/* get deviation from bin frequency from the +/- Pi interval */
tmp = osamp*tmp/(2.*M_PI);
/* compute the k-th partials' true frequency */
freq = (double)k*freqPerBin + tmp*freqPerBin;
```

Da die Hauptkeule der Fensterfunktion mehrere Bins breit ist, verteilt sich die Frequenz eines Sinusoids auf diese Bins. Der Sinusoid liegt immer genau in dem Bin, dessen Amplitude den höchsten Wert hat. Durch das Overlapping wird für alle Bins der Hauptkeule dieselbe wahre Frequenz berechnet, ohne Overlapping erhält man unterschiedliche Frequenzen. Die Amplituden der Bins können dann einfach addiert werden, da die Resynthese-Phasen einander angeglichen sind. Der minimal benötigte Overlap hängt auch von der Fensterfunktion und der Breite von deren Hauptkeule ab.

Die Veränderung der Tonhöhe ist durch Verschieben der berechneten Sinuskomponenten einfach möglich. Dabei verschieben sich jedoch auch die Formantfrequenzen, so dass die Klangfarbe ebenfalls verändert wird. Für Sprachsynthese ist es aber wichtig, Algorithmen zu haben, mit denen sich die Klangfarbe und die Tonhöhe unabhängig voneinander bearbeiten lassen. Um die ursprüngliche Klangfarbe wiederherzustellen, kann man vor der Verschiebung die spektrale Hüllkurve berechnen, z.B. mit dem Cepstrum [27], und nachher diese auf das verschobene Signal anwenden. Dieses Verfahren wird z.B. zur Korrektur der Tonhöhen von Gesang eingesetzt. Dabei kann sich die Klangqualität erheblich verschlechtern, da eine Änderung der Klangfarbe durch das verschmierte Spektrum erschwert wird. Ursprünglich wurde der Phase-Vocoder für polyphone Signale entwickelt, daher ist es nicht möglich, die harmonische Zusammensetzung eines Sprachsignals auszunutzen. Auch Frequenzen in der Nähe der Frequenzgrenzen können zu Problemen führen, da die gemessenen Frequenzen dann negativ oder größer als die halbe Samplerate werden. Diese Frequenzen sind jedoch nicht hörbar und werden daher bei der Resynthese einfach ignoriert.

Bin	Bin Frequency	True Frequency	Magnitude
0	0.000000	-43.066406	0.889218
1	43.066406	48.431618	1.937133
2	86.132812	48.588169	0.989389
3	129.199219	150.502655	1.960648
4	172.265625	138.397659	1.672815
5	215.332031	243.256882	1.372568
6	258.398438	251.353271	4.090764
7	301.464844	271.203217	3.631035
8	344.531250	351.055450	13.560002
9	387.597656	353.969910	178.309799
10	430.664062	440.019348	505.830505
11	473.730469	440.039246	345.499756
12	516.796875	527.211304	27.783810
13	559.863281	528.404602	6.682389
14	602.929688	621.315674	4.086450
15	645.996094	637.450378	1.803178
16	689.062500	667.163330	1.599435
17	732.128906	729.331116	2.250777
18	775.195312	738.707031	1.018117
19	818.261719	832.406982	1.892177
20	861.328125	828.987244	1.234164
21	904.394531	929.738708	1.495662
22	947.460938	928.690430	1.599610
23	990.527344	1029.575439	1.301059
24	1033.593750	1024.933960	2.076987
25	1076.660156	1037.259766	1.125592
26	1119.726562	1126.565063	3.281369
27	1162.792969	1190.755249	1.104974
28	1205.859375	1226.516846	5.881757
29	1248.925781	1234.865601	17.282423
30	1291.992188	1320.089600	131.076431
31	1335.058594	1320.066772	160.679260
32	1378.125000	1406.612061	45.038883
33	1421.191406	1414.413696	6.842116
34	1464.257812	1499.679565	3.471327
35	1507.324219	1517.409302	2.194740
36	1550.390625	1512.641724	2.151571
37	1593.457031	1610.941895	1.895323
38	1636.523438	1615.018921	1.693528
39	1679.589844	1713.117798	1.377131
40	1722.656250	1710.437378	1.720258

Tabelle 2.1: Frequenzen mit und ohne Phase-Locking eines Rechtecksignals, berechnet mit Algorithmus 2.1

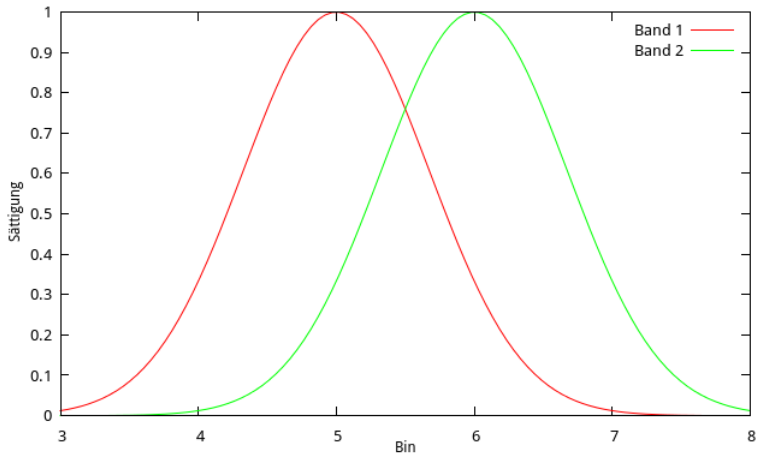

Abbildung 2.4: Zwei überlappende Frequenzbänder als Glockenkurven dargestellt

Bei Sprache können bei tiefen Frequenzen Artefakte auftreten. Höhere Frequenzen sind weniger problematisch, da die Hüllkurve mit der Frequenz fällt. Kennt man die True-Frequencies und Magnituden, so kann man damit die harmonischen Frequenzen berechnen und harmonische Anteile von Inharmonischen trennen. In Tabelle 2.1 sind zwei harmonische Peaks erkennbar, diese liegen bei 440 Hz und bei 1320 Hz.

2.3 Spectral Model Synthesis

Sprache besteht aus stimmhaften und stimmlosen Segmenten. Stimmlose Segmente lassen sich einfach als moduliertes Rauschen modellieren, daher kann deren Phase bei der Analyse ignoriert werden. Bei der Resynthese wird die Phase mit Zufallswerten gefüllt. Ein stimmhaftes Segment ist durch das Vorhandensein einer fundamentalen Frequenz f_0 bestimmt. Diese wird als Tonhöhe (Pitch) wahrgenommen. Vielfache der fundamentalen Frequenz werden als Harmonische oder Obertöne bezeichnet. Neben den harmonischen Frequenzen enthält ein Sprachsignal noch einen inharmonischen Rest, welcher wie stimmlose Signale beschrieben wird. Die Ursache dafür sind durch die Atmung verursachte Turbulenzen.

2.3.1 Harmonic Trajectories

Grundgedanke von Spectral Model Synthesis ist die Trennung der harmonischen Komponente vom Rauschen. Dadurch lassen sich die Klangfarbe und Tonhöhe unabhängig voneinander bearbeiten. (siehe Abb. 2.5) Die Tonhöhe lässt sich durch Verschieben der harmonischen Frequenzen verändern. Damit sich die Klangfarbe nicht ändert, werden die Formantamplituden zwischen den ursprünglichen Harmonischen neu interpoliert.

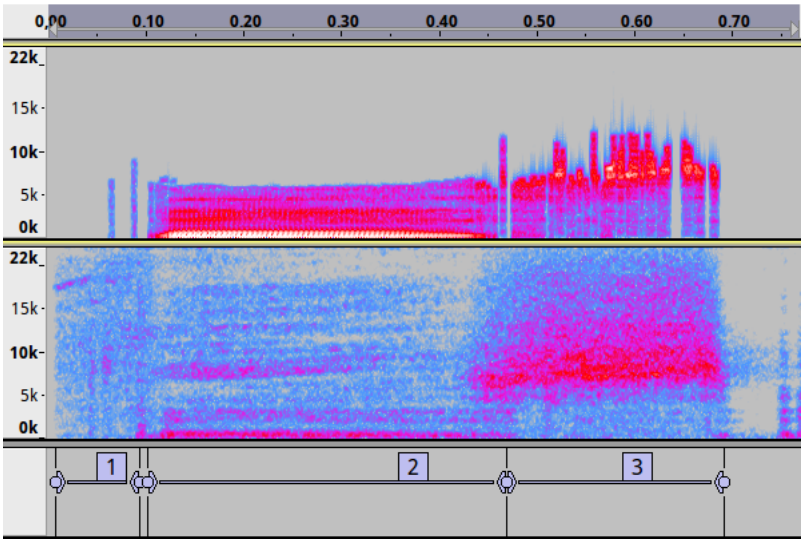

Abbildung 2.5: Sprachdatendarstellung mit SMS: Segment 1 ist ein stimmloser Konsonant, Segment 2 ein Vokal und Segment 3 ein stimmhafter Konsonant (mixed Sound). Grafik erstellt unter der Verwendung von [24]

Das Residual kann unabhängig bearbeitet werden. Die Klangfarbe des stimmhaften Teils wird ausschließlich durch die Amplituden der Harmonischen bestimmt, die Phase wird genauso behandelt wie beim Phase-Vocoder. Durch Interpolation der harmonischen Amplituden lässt sich die Tonhöhe unabhängig von der Klangfarbe verändern. Das Restsignal wird durch Subtraktion der Harmonischen vom ursprünglichen Signal ermittelt. In der Praxis beeinflusst auch das Restsignal die Klangfarbe, insbesondere bei stimmhaften Konsonanten wie z.B. [[R]] führt dies zu Unregelmäßigkeiten.
Die fundamentale Frequenz wird über die lokalen Maxima des Spektrums (spectral peaks) berechnet. Da sich die Zusammensetzung des Signals mit der Zeit ändert, ändern sich auch die Positionen der Peaks. Jeder der Peaks liegt im Idealfall genau auf einer harmonischen Komponente, so dass man von der Position der Peaks auf die fundamentale Frequenz schließen kann. Wurde die fundamentale Frequenz erkannt, so wird derselbe Frame noch einmal analysiert, diesmal mit einer Fenstergröße, die ein ganzzahliges Vielfaches der Periodenlänge ist. Dadurch wird das Spektrum weniger verschmiert und man erhält ein genaueres Ergebnis. Für den nächsten Frame wird dann die im letzten Frame berechnete Periodenlänge genutzt und erneut an die geringfügig veränderten Werte angepasst. Dabei wird versucht, gegenüber dem vorherigen Frame eine Fortsetzung der harmonischen Tracks zu finden. Dies gelingt jedoch nicht immer. Insbesondere ist die Referenzimplementierung [25] oft nicht in der Lage, die Harmonischen und die fundamentale Frequenz zu erkennen, auch bei längeren stimmhaften Signalen nicht.

Wenn sich die fundamentale Frequenz schnell ändert oder Störsignale als Harmonische interpretiert werden, so führt dies zu Unregelmäßigkeiten in der harmonischen Fortsetzung, die in der Resynthese auch zu hörbaren Artefakten führen. Auch Attacks und Releases führen zu solchen Problemen, da das Signal hier nicht mehr stationär ist.

T1	T2	T3	T4	T5
311.286469	311.912079	312.644867	312.833221	311.328186
623.394714	623.688232	622.755981	622.987976	620.909302
935.180054	934.625183	936.666016	934.341003	934.936157
1246.286743	1246.591431	1248.659058	1246.561279	1246.260986
1558.040771	1557.505371	1559.681396	1557.673706	1559.106079
1870.114624	1870.120361	1870.171631	1870.421143	1870.622314
2180.243896	2181.412109	2184.310059	2182.444580	2183.929199
2496.777344	2493.641846	2498.675049	2490.151611	2492.291504
2802.739014	2805.329590	2812.168701	2804.385986	2803.054443
3119.335205	3116.958008	3116.470215	3119.116455	3118.191162
3436.315674	3427.212646	3426.838623	3427.080078	3434.969727
3739.091797	3742.905029	3743.506348	3739.352295	3738.541504
4050.274170	4052.493164	4060.271484	4051.400635	4048.284668
4373.381836	4359.736816	4371.482910	4359.381836	4366.775879
4677.876465	4675.421387	4686.289062	4671.494141	4673.742188
4989.358398	4988.153809	4993.708008	4986.124023	4988.258789
5311.376465	5289.208008	5308.208984	5279.347656	5312.831543
5929.295898	5917.673340	5918.624023	5929.483887	5923.653320

Tabelle 2.2: Harmonic Trajectories an 5 unterschiedlichen benachbarten Zeitpunkten: Die Zeilen stellen die Fortsetzung über die Zeit dar, in den Spalten befinden sich die harmonischen Frequenzen zu einem Zeitpunkt in Hz. Die fundamentale Frequenz liegt bei ungefähr 312 Hz. Größere Abweichungen in den höheren Frequenzen deuten auf inharmonische Anteile hin. Werte berechnet unter Verwendung der Software aus [25]

Die genaue Frequenz eines Peaks wird genauso berechnet wie beim Phase-Vocoder, welcher ein stationäres Signal voraussetzt. Der Vorteil hier ist, dass nur an den harmonischen Kandidaten die wahre Frequenz berechnet werden muss. Neu auftretende Harmonische führen hier auch zu Problemen bei der Zuordnung der Phase, da jeder harmonische Track über seine Phase definiert ist. Genau wie beim Phase-Vocoder arbeitet auch SMS mit einer konstanten Framerate. Betrachtet man die Harmonischen als überlappende Frequenzbänder, so ist deren Auflösung größer als die Kanalauflösung beim Phase-Vocoder. Die Bandbreite ist dann ein Vielfaches der fundamentalen Frequenz. Wird keine fundamentale Frequenz erkannt, so werden allen harmonischen Amplituden Werte von Null zugewiesen, das ganze Signal wird dann wieder mit einer voreingestellten Fenstergröße statistisch analysiert.

Da die FFT mit einer Zweierpotenz an Samples arbeitet, wird das gefensterte Signal, dessen Länge von der Periodenlänge abhängt, von rechts mit Nullen aufgefüllt. Dadurch wird der Frequenzgang bis auf Rundungsfehler nicht beeinflusst, da ein Nullsignal keine Frequenzen enthält. Jedoch verschieben sich die messbaren Phasen. Durch die konstante

FFT-Größe ist die tiefstmögliche, analysierbare fundamentale Frequenz bestimmt. Bei einer Fenstergröße von 2048 und einer Samplerate von 44100 Hz liegt diese bei etwa 71 Hz. Dies ist in der Praxis jedoch kein Problem, da so tiefe Frequenzen auch in einer männlichen Stimme nicht vorkommen. Ein größeres Fenster würde die zeitliche Auflösung nur unnötig verringern und den Rechenaufwand erhöhen.

Unregelmäßigkeiten in der gemessenen Phase und damit in der wahren Frequenz treten vor allem in den höheren Obertönen auf, da sich Messfehler hier vervielfachen und das Signal/Noise-Ratio hier geringer ist. Durch Transformationen und Konkatenation kann der Fehler noch weiter verstärkt werden, so dass Artefakte zu hören sind. Aus diesem Grund wird ein Phasenmodell benötigt, mit dem die Phasen vorhergesagt werden können. Das Phasenspektrum hängt auch von der genauen Position des Analysefensters ab. Dies wird in der ursprünglichen Implementierung [25] aber nicht berücksichtigt, die rein harmonischen Phasen werden von Frame zu Frame bei der Resynthese einfach inkrementiert. Dadurch können weitere Artefakte auftreten, wenn Parameter verändert werden. Spectral Model Synthesis modelliert Frequenzkomponenten getrennt für den harmonischen und den inharmonischen Teil des Spektrums. Damit lassen sich diese unabhängig voneinander bearbeiten, man hat jedoch keine Kontrolle über die Positionen der Glottispulse und deren Form. Die Form hängt von den harmonischen Frequenzen sowie deren Amplituden und Phasen am Voice-Pulse-Onset ab.

Harmonische Frequenzen lassen sich entweder als Sinusoids oder spektrale Regionen darstellen. Daher gibt es zwei unterschiedliche Methoden zur Resynthese eines rein harmonischen Signals. Im ersten Fall wird mit einer Time-Domain-Oscillatorbank gearbeitet, welches jedes Sample als eine Summe von Sinusfunktionen darstellt. In diesem Fall benötigt man keine überlappenden Fenster. Für jede dieser Sinusfunktionen wird ein eigener Phasenakkumulator verwendet. Dieses Verfahren funktioniert auch dann noch, wenn das Signal nicht stationär ist, jedoch ist es bei einer großen Anzahl von Sinusoids nicht mehr effizient. So arbeitet z.B. der letzte Verarbeitungschritt von eSpeak nach diesem Verfahren. Um die Effizienz zu verbessern gibt es die IFFT-Methode, welche von (Depalle and Rodet 1990) und (Dutoit 1993) vorgeschlagen wurde [26, Seite 81]. Eine Implementierung des Verfahrens ist in der libsms [25, Datei src/synthesis.c] enthalten. Jedoch funktioniert diese in der Effizienz verbesserte Variante nur, wenn das Signal stationär ist.

Die andere Methode basiert auf dem Phase-Locked-Vocoder (siehe oben) und verwendet für jeden Bin einen eigenen Phasenakkumulator. Aus der True-Frequency lässt sich dann die Bin-Frequency und die Phasenabweichung berechnen.

2.3.2 Shape Invariance

Basierend auf dem vereinfachten Source-Filter Modell ist die Form der einzelnen Pulse im Zeitbereich unabhängig von der Tonhöhe. Ein Sprachsignal kann daher als sich überlappende Impulsantworten des Vokaltrakt-Filters beschrieben werden, wobei als Anregungssignal nur eine Impulsfolge ohne Bandbeschränkung mit der Frequenz f_0 dient. [26] Diese Darstellung wird als spektrale Faltung bezeichnet. Die Form des Zeitsignals in der Nähe des Voice-Pulse-Onset hängt dabei nur von der Impulsantwort des Vokaltrakt-Filters ab. Im Zeitbereich arbeitende Algorithmen, wie z.B. TD-PSOLA, erhalten die Shape Invariance in den meisten Fällen, da hier die Impulsantwort einfach unverändert kopiert wird, während bei Vocodern und SMS die Shape Invariance nicht gewährt ist. Das bedeutet auch, dass es keine Phasenkohärenz zwischen den einzelnen Frames gibt und Veränderungen einzelner Parameter zu Phasiness führen können. Damit die Phasenkohärenz am Voice-Pulse-Onset gewährleistet ist, gibt es Algorithmen, welche dessen zeitliche Position samplegenau berechnen. Dies setzt voraus, dass die Hüllkurve der fundamentalen Frequenz akkurat ermittelt wurde, da die Phase direkt von der Frequenz abhängig ist. Damit lässt sich für jedes Sample die fundamentale Phase ermitteln und das Analysefenster genau auf einem Voice-Pulse-Onset positionieren. Ist das Analysefenster zwischen zwei Voice-Pulse-Onsets, so führt dies zu hörbaren Phasenauslöschungen bei der Resynthese, da sich die Spektra zweier Pulse mit starker Amplitudenmodulation vermischen. Die Voice-Pulse-Onsets entsprechen den Glottal Closure Instants, also den Zeitpunkten, an denen sich die Glottis schließt.

Verschiebt man das Analysefenster, so verschiebt sich auch die Phasenhüllkurve. Der Phasenwert an Bin 0 ist genau dann minimal, wenn das Analysefenster genau auf einem Voice-Pulse-Onset liegt. Verschiebt man das Analysefenster nach rechts oder links, so steigt der Wert bis die halbe Periodenlänge erreicht ist, danach fällt der Wert wieder. Die gesamte Phasenhüllkurve ist im Idealfall flach und fällt unter den Formanten ab. Diesen Zustand bezeichnet man auch als Maximally Flat Phase Alignment (MFPA) [26]. Für ein rein harmonisches Signal lässt sich daher die Phase direkt aus der Amplitudenhüllkurve berechnen. Das Spektrum wird dann als minimalphasig bezeichnet. Im Idealfall ist der Phasenunterschied zwischen zwei identischen Pulsen immer Null. Dies macht sich auch der MPFA Algorithmus zu Nutze:

> „The MFPA algorithm attempts to find the time-shift Δt that minimizes the phase differences between harmonics, therefore obtaining a maximally flat phase alignment.“ [26, Seite 56]

Hat man die Position des ersten Pulses bestimmt, lassen sich damit auch die Positionen der weiteren Pulse einfach bestimmen. Da sich zu jedem Puls ein Maximum in der Wellenform findet, (siehe Abb. 2.6) lässt sich die Position des ersten Pulses einfach suchen, indem man innerhalb der ersten Periode des Signals nach dem Maximum sucht. Dann kann man mit MFPA die Positionen der weiteren Pulse markieren. Dies funktioniert auch dann, wenn sich die fundamentale Frequenz ständig ändert, da die fundamentale Phase samplegenau linear interpoliert wird. Damit lassen sich die Punkte berechnen, auf denen

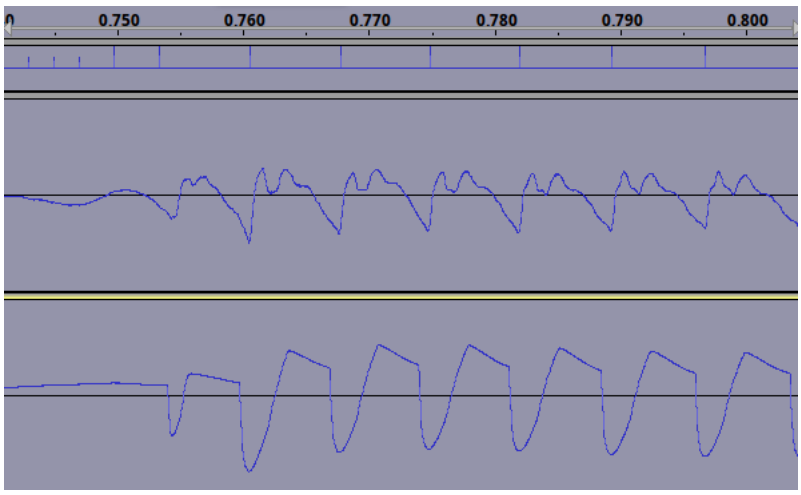

Abbildung 2.6: Sprachsignal und Laryngographsignal aus der CMU Arctic Sprachdatenbank [10]. Die untere Wellenform stellt das Laryngographsignal dar, darüber befindet sich das Sprachsignal und der kleine Track weiter oben mit den Pulsen stellt die Berechungsergebnisse des PLATINUM Algorithmus [23] dar. Kleinere Pulse stehen für stimmlose Segmente. Darüber befindet sich die Zeitachse. Der Phasenunterschied zwischen zwei Pulsen ist minimal.

das Analysefenster positioniert wird, und man bei Transformationen den bestmöglichen Klang erhält. Die Shape Invariance ist dadurch gewährleistet. Ursprünglich wurde die Minimierung der Phasenunterschiede für jede der harmonischen Phasen durchgeführt und der Track ausgewählt, bei dem der Fehler am geringsten ist.

Der PLATINUM Algorithmus [23] erhält ebenfalls die Shape Invariance, während dies bei den Vorgänger-Algorithmen STRAIGHT und TANDEM-STRAIGHT nicht der Fall ist, da hier ein künstliches Anregungssignal verwendet wird. Dieses besteht teilweise aus Rauschen und teilweise aus einem harmonischen Pulse-Train.

2.4 Voice Pulse Modelling

Ein stimmhaftes Sprachsignal kann als Sequenz von sich wiederholenden Impulsantworten eines Filters beschrieben werden. Jede Impulsantwort selbst lässt sich als ein Anregungssignal, gefiltert durch einen Vokaltrakt oder Formantfilter, beschreiben. Mit Voice Pulse Modelling ist es möglich, die Positionen der Glottispulse und deren Frequenzen unabhängig voneinander zu bearbeiten. Dies ist bei Verfahren, welche Frequenzkomponenten modellieren (Phase Vocoder und SMS) nicht möglich, da sich jede Frequenzkomponente auf mehrere Pulse verteilt. Voice Pulse Modelling kombiniert also die Vorteile von Time-Domain Algorithmen wie TD-PSOLA, mit denen von Spectral Domain Algorithmen wie Vocodern.

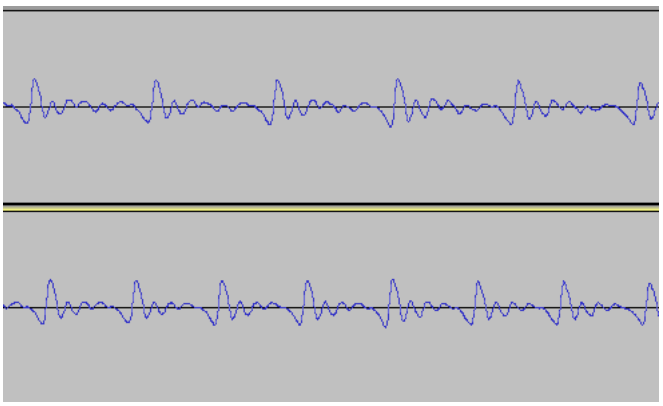

Abbildung 2.7: Veränderung der Tonhöhe eines Sprachsignals mit TD-PSOLA

2.4.1 TD-PSOLA

Mit dem TD-PSOLA Algorithmus [16, Kapitel 10] lässt sich die Tonhöhe eines Sprachsignals verändern, ohne dass dieses in den Frequenzbereich transformiert werden muss. Dabei wird der Abstand zwischen zwei benachbarten Pulsen verändert, ohne dass sich die spektrale Hüllkurve verändert. Auch Änderungen der Sprechgeschwindigkeit sind sehr einfach möglich, indem Pulse wiederholt oder ausgelassen werden. Dabei kommt es nicht zu modellbedingten Verschlechterungen der Qualität. TD-PSOLA ist so effizient, dass es in Echtzeit auf einem 386 oder einem ARM-System durchgeführt werden kann. Änderungen der Klangfarbe sind ebenfalls möglich, jedoch nur durch lineare Skalierung der Glottispulse. Die einzelnen Pulse werden pitch-synchron über eine Gewichtungsfunktion aus dem ursprünglichen Signal ausgeschnitten und dann durch Verschiebung und Addition zusammengefügt. (siehe Formel 2.4.1.a) Werden keine Veränderungen durchgeführt, so erhält man im Idealfall wieder das Ausgangssignal. Dies funktioniert nur, wenn das Signal rein periodisch und die Periodenlänge bekannt ist. Diese ist dann konstant und wird als t_0 bezeichnet.

Formel 2.4.1.a: $\tilde{s}(n) = \sum_{i=-\infty}^{+\infty} s(n)w(n - it_0)$ [16]

Wenn die Tonhöhe verändert wird, wird das Spektrum mit der veränderten fundamentalen Frequenz reharmonisiert, d.h. die Positionen der Harmonischen verändern sich, ohne dass sich die Hüllkurve verändert. (siehe Abb. 2.7).

Damit die Reharmonisierung gelingt, muss das Analysefenster schmal genug sein, so dass ein Dirac-Impulsspektrum modelliert wird. Ist das Analysefenster zu klein, so ist die Reharmonisierung grobkörnig. Ist es zu groß, so erreicht man keine Reharmonisierung, da dann Linien im Spektrum auftreten, weil das so modellierte Spektrum kein Puls mehr ist. [16] Die Form des Signals in der Nähe der Fenstermitte verändert sich dabei nicht, die Shape-Invariance bleibt daher gewahrt. Bei realen Sprachsignalen ist die Periodenlänge niemals konstant, da sich die Eigenschaften der Sprachorgane ständig ändern. Dadurch kann es zu hörbarer Phasiness kommen, wenn die Analysefenster auf unterschiedlichen

Positionen innerhalb der Wellenform zentriert sind. Daher ist es wichtig, die fundamentale Frequenz genau zu bestimmen, damit der MFPA Algorithmus die Positionen der Glottal Closure Instants (CGIs) berechnen kann. Bevor der MFPA Algorithmus erfunden wurde, hat man die CGIs von Hand markiert. Dies ist sehr fehleranfällig und es kam daher häufig zu Phase- und Pitch-Mismatches bei der Konkatenation. Bei der Erstellung einer Sprachdatenbank ist es daher wichtig, mit einer möglichst konstanten Tonhöhe zu sprechen um diese Fehlerquelle zu minimieren.

2.4.2 MBR-PSOLA

Da TD-PSOLA ein rein periodisches Signal erwartet und reale Sprachsignale niemals rein periodisch sind, verwendet man Spectral Model Synthesis (auch als Hybrid Harmonic / Stochastic Synthesis [16, Kapitel 9], kurz H/S bekannt), um das Signal mit konstanter fundamentaler Frequenz zu resynthetisieren. Dadurch lassen sich Pitch und Phase-Mismatches vollständig vermeiden, da dann die harmonischen Frequenzen und Phasen benachbarter PSOLA-Frames identisch sind. Die Interpolation der Impulsantworten kann dadurch einfach als Time-Domain-Linear-Smoothing durchgeführt werden, so dass keine hörbaren Envelope-Mismatches mehr auftreten. Diese Reharmonisierung mit der durchschnittlichen fundamentalen Frequenz der Sprachdatenbank, muss nur einmal bei der Erstellung der Sprachdatenbank durchgeführt werden. Auf diese Weise lässt sich ein sehr effizienter Sprachsynthesizer bauen, welcher auch sehr natürlich klingt.

Bei der Resynthese wird dann TD-PSOLA verwendet. Die Positionen der Glottispulse lassen sich dann sehr einfach berechnen, da sich diese an einem Vielfachen der konstanten Periodenlänge befinden. Da bei der Analyse die fundamentale Frequenz erkannt wird, ist hier keine manuelle Erstellung der Pitchmarks mehr notwendig. Das Time-Domain Linear Smoothing verhält sich ähnlich wie die Interpolation der harmonischen Frequenzen zwischen zwei Frames, ist aber schneller zu berechnen, da hier keine Transformation in den Zeitbereich mehr durchgeführt werden muss. Gleichzeitig sind durch die Resynthese mit einer konstanten fundamentalen Frequenz die Probleme der Phasenkonkatenation weitgehend gelöst, so dass in den phonetischen Übergängen keine Artefakte zu hören sind. Durch eine Kodierung mit DPCM lassen sich die Sprachdatenbanken noch besser komprimieren, ohne dass das Echtzeitverhalten negativ beinflusst wird.

2.4.3 NBVPM

Narrow Band Voice Pulse Modelling [26] setzt voraus, dass die spektrale Hüllkurve einer Impulsantwort in der Bandbreite beschränkt ist und sich diese nur langsam über die Frequenz ändert. Da das Anregungssignal langsam mit der Frequenz fällt, fällt auch die Hüllkurve. (siehe Abb. 2.8). Samplet man ein Signal über mehrere Perioden mit einer Frequenzauflösung, die größer ist als die Bandbreite der Impulsantwort, so erhält man eine schmalbandige Darstellung, die genau eine Impulsantwort beschreibt. Wenn die Un-

terschiede und der Abstand zwischen benachbarten Pulsen minimal sind, so lässt sich das harmonische Spektrum einfach berechnen. Das stationäre Signal verhält sich dann wie eine periodische Wiederholung eines Pulses. Da immer mehrere Perioden zusammengefasst analysiert werden (siehe Abb. 2.9), funktioniert dieses Verfahren nur, wenn die Signalzusammensetzung über einen längeren Analysezeitraum stationär ist und sich die fundamentale Frequenz nicht ändert.

Die harmonischen Frequenzen und Phasen werden dann durch Interpolation der Spektralen Peaks gewonnen. Durch Berechnung der inversen Fouriertransformation kann dann ein Zeitsignal gewonnen werden, welches mit Algorithmen wie TD-PSOLA transformiert werden kann. Damit lässt sich dann z.B. die Tonhöhe verändern, indem der Abstand zwischen den Pulsen verändert wird. [26, Seite 98] Bildet man die Differenz zwischen dem ursprünglichen Signal und der NBVPM-Resynthese so lässt sich ein Residual berechnen, welches unabhängig von der harmonischen Komponente transformiert werden kann. Dadurch lässt sich die Flexibilität von spektralen Modellen mit der für Time-Domain Algorithmen typischen Fähigkeit, einzelne Pulse unabhängig voneinander zu bearbeiten, kombinieren. NBVPM ähnelt daher Verfahren wie MBR-PSOLA und SMS, da in beiden Verfahren die Reharmonisierung durch eine Trennung von harmonischen und inharmonischen Komponenten durchgeführt wird.

Für die Analyse wird ein Fenster verwendet, welches drei Perioden lang ist, da hier die harmonischen Peaks recht gut zu erkennen sind. Bei längeren Fenstern werden Transienten stärker verschmiert, so dass die Verständlichkeit beeinträchtigt wird. Da die fundamentale Frequenz hier über die Peaks bestimmt wird, darf die Fensterfunktion nicht zu kurz sein. Der Algorithmus arbeitet mit einer konstanten Hop-size, so dass sich Phasendifferenzen und die wahren Frequenzen wie beim Phase-Vocoder berechnen lassen. Die Phasen der harmonischen Frequenzen werden mit dem MFPA Algorithmus [26] so modifiziert, dass keine Phasiness auftritt, wenn Parameter verändert werden. Die Länge der resynthetisierten Pulse ist dabei unabhängig von der fundamentalen Frequenz und hängt nur von den Eigenschaften des Vokaltraktes ab. Da der Algorithmus unter schmalbandigen Bedingungen arbeitet, ist die zeitliche Auflösung auf die Hop-Size beschränkt. Dies führt dazu, dass Transienten verschmiert werden und die Sprache in diesem Fall unnatürlich klingt. Auch bei stimmhaften Lauten macht sich der Nachteil der schmalbandigen Analyse bemerkbar, da ein reales Sprachsignal niemals stationär ist. Die ursprüngliche Annahme, dass die benachbarten Pulse sehr ähnliche Formen haben, ist insbesondere in phonetischen Übergängen nicht mehr wahr. Da NBVPM ein periodisches Signal voraussetzt, eignet sich das Verfahren nur für stimmhafte Segmente. Für stimmlose Segmente kann der Phase-Vocoder eingesetzt werden. Insbesondere führen Transienten zwischen stimmhaften und stimmlosen Segmenten hier zu hörbaren Artefakten. Aus diesem Grund ist ein Transientenmodell erforderlich, mit dem sich die Natürlichkeit von harten Konsonanten erhalten lässt. Außerdem können kleinere Analysefehler manchmal störende Nebengeräusche verursachen. Ein weiterer Nachteil ist die schlechte zeitliche Auflösung gegenüber reinen Time-Domain Algorithmen wie TD-PSOLA.

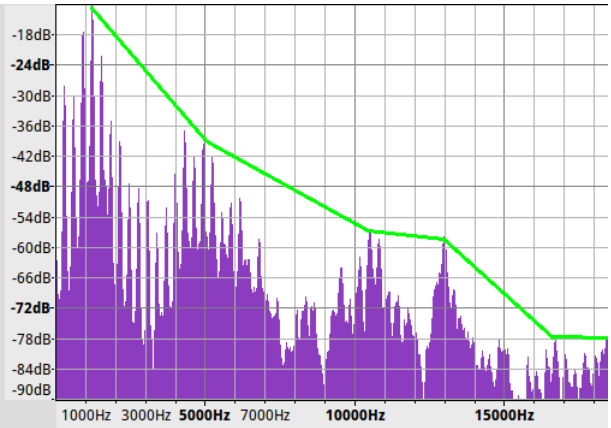

Abbildung 2.8: Fallende Hüllkurve eines schmalbandigen Signals

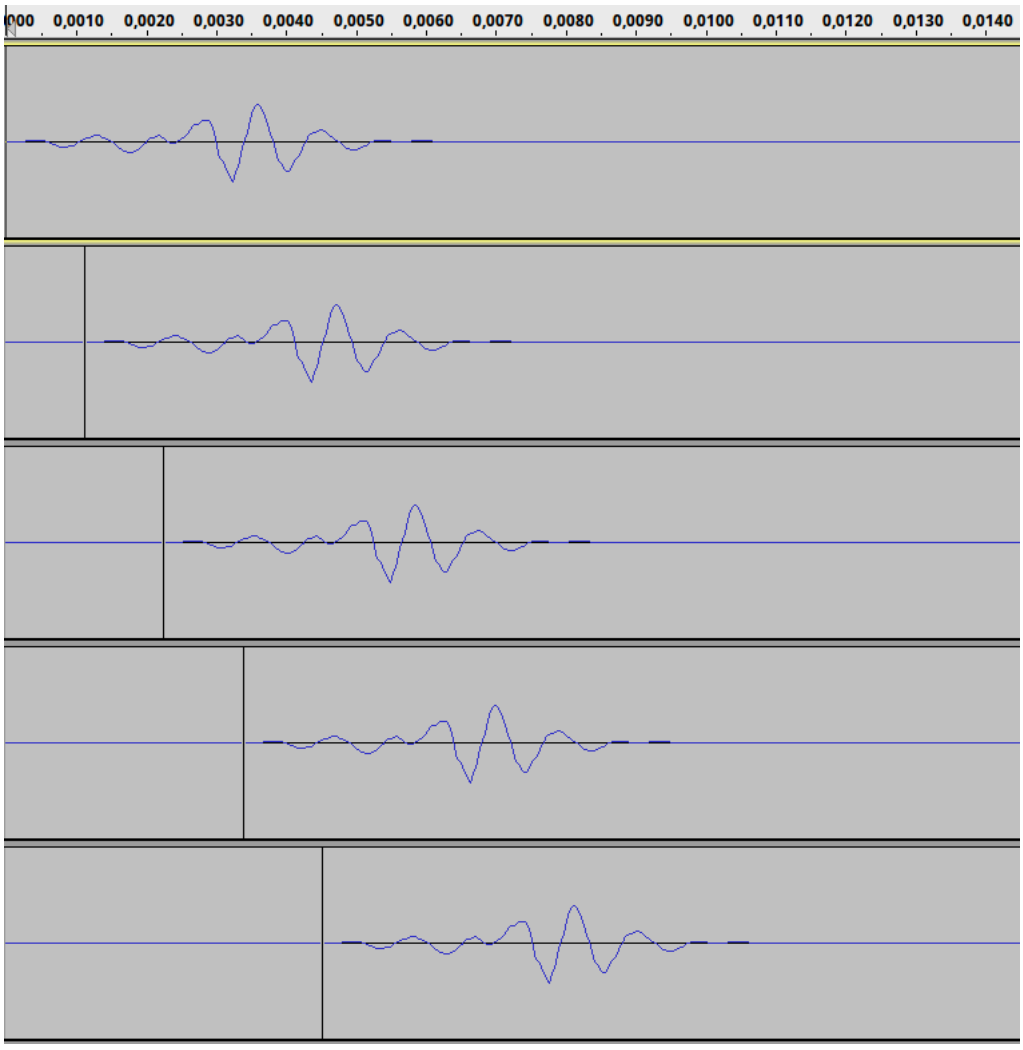

Abbildung 2.9: Mehrere identische überlappende Stimmenpulse in gleichen Abständen. Berechnet man das Spektrum der Summe dieser Pulse, so erhält man eine ungefähre Darstellung des harmonischen Spektrums, welches Peaks an den harmonischen Frequenzen enthält. In diesem Beispiel beträgt der Abstand zwischen zwei Pulsen 50 Samples.

2.4.4 WBVPM

Mit Wide Band Voice Pulse Modelling lässt sich die zeitliche Auflösung verbessern. Dies ist möglich, indem die harmonische Struktur eines Sprachsignals so breitbandig wie nur möglich analysiert wird. Die Impulsantwort wird hier nicht als Impulsantwort des Vokaltrakt-Filters gefaltet mit dem Anregungssignal verstanden, sondern als der zeitliche Bereich zwischen zwei Voice-Pulse-Onsets. Die schmalbandigen Impulsantworten können länger oder kürzer sein als die Periodenlänge eines Sprachsignals. Sie enthalten nur den harmonischen Teil des Sprachsignals. WBVPM dagegen modelliert harmonische Komponenten und ein Rauschen, ohne dass z.B. über Harmonic Trajectories eine Trennung vorgenommen werden muss.

WBVPM setzt voraus, dass die fundamentale Frequenz und damit die Periodenlänge vor der Analyse mit einem geeigneten Algorithmus (z.B. yinfft [12] oder dio [23]) ermittelt wurde, und diese keine Sprünge enthält. Damit lassen sich die harmonischen Frequenzen und deren Phasen berechnen. Ein stationäres periodisches Signal $s(n)$ mit einer Samplerate von FS lässt sich einfach als eine Summe von Sinusfunktionen ohne eine stochastische Komponente darstellen: [26, Seite 105]

$$s(n) = \sum_{k=1}^{0.5T} a_k \cos(2\pi \frac{f_k}{f_s} n + \theta_k)$$

Die Fouriertransformation eines Signals unter Verwendung einer Fensterfunktion ist als Faltung des Signals mit der Fensterfunktion definiert. Bei WBVPM wird ein Rechteckfenster verwendet, welches genau eine Periodenlänge des Signals lang ist. Dadurch sind alle messbaren Frequenzen Vielfache der fundamentalen Frequenz, da die Fouriertransformation des Rechteckfensters zwischen den Harmonischen den Wert 0 hat (siehe Abb. 2.10) Diese Bedingungen bezeichnet man als maximal breitbandig, da das Rechteckfenster unendlich viele Harmonische, also keine Bandbeschränkung, hat. Die gemessenen Amplituden entsprechen bei einem rein harmonischen Signal dann den harmonischen Amplituden. Reale Signale enthalten jedoch noch ein Rauschen, welches ebenfalls im breitbandigen Spektrum auftritt. Daher ist auch dieses an den Harmonischen und in den Frequenzen dazwischen messbar. Die DTFT eines normalisierten Rechteckfensters wird folgendermaßen dargestellt, f_k steht dabei für die harmonischen Frequenzen:

$$x(n) = s(n) w_R(n)$$

$$X(f) = \sum_k W_R(f_k - f) S(f_k)$$

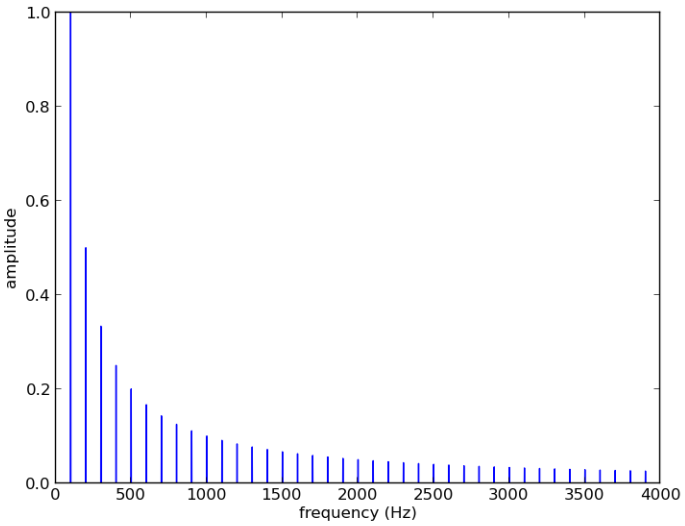

Abbildung 2.10: Abtastung eines Signals nur an den harmonischen Frequenzen: Wenn die Länge des Rechteckfensters ein ganzzahliges Vielfaches der Periodenlänge ist, so sind die messbaren Frequenzen ganzzahlige Vielfache der fundamentalen Frequenz. Bei der kürzestmöglichen Fensterlänge ist die Analyse maximal breitbandig.

Da die DFT mit einer ganzzahligen Anzahl Samples arbeitet und die Periodenlänge in der Realität jedoch niemals ganzzahlig ist, muss diese auf eine ganze Zahl gerundet werden. Damit verschieben sich die messbaren Frequenzen und deren Phasen geringfügig gegenüber den wahren Harmonischen. In der Praxis wird jedoch immer die FFT verwendet, da die DTFT ineffizient zu berechnen ist. Die FFT erwartet eine Zweierpotenz an Samples. Das Signal wird dann von rechts mit Nullen aufgefüllt, wodurch die messbaren Frequenzen nicht mehr mit den Harmonischen übereinstimmen und auch die messbaren Phasen verändert werden. Die Amplituden und Phasen der harmonischen Frequenzen lassen sich dann nur über Interpolation berechnen, ein direktes Ablesen ist durch das Frequenzraster nicht mehr möglich. Harmonische Peaks sind im breitbandigen Spektrum weniger ausgeprägt, trotzdem lassen sich die harmonischen Amplituden ablesen, da f_0 bekannt ist. Alternativ kann man das Signal mehrfach wiederholen und danach mit der Fouriertransformation analysieren.

Da es in WBVMP keine Trennung zwischen Rauschen und den Harmonischen gibt, lässt sich so eine Verbesserung der Klangqualität erreichen. Vergleicht man das ursprüngliche Signal mit der WBVPM-Resynthese, so erhält man, verglichen mit NBVPM, einen geringeren Fehler, d.h. die Energie des Residuals ist geringer. Sowohl Phasen als auch Amplituden stimmen, bis auf Rundungsfehler und Analysefehler, überein. Die Klangfarbe ist dann direkt über das Spektrum bestimmt und lässt sich durch Skalierung des Spektrums verändern. Gleichzeitig lässt sich die Tonhöhe unabhängig verändern, indem

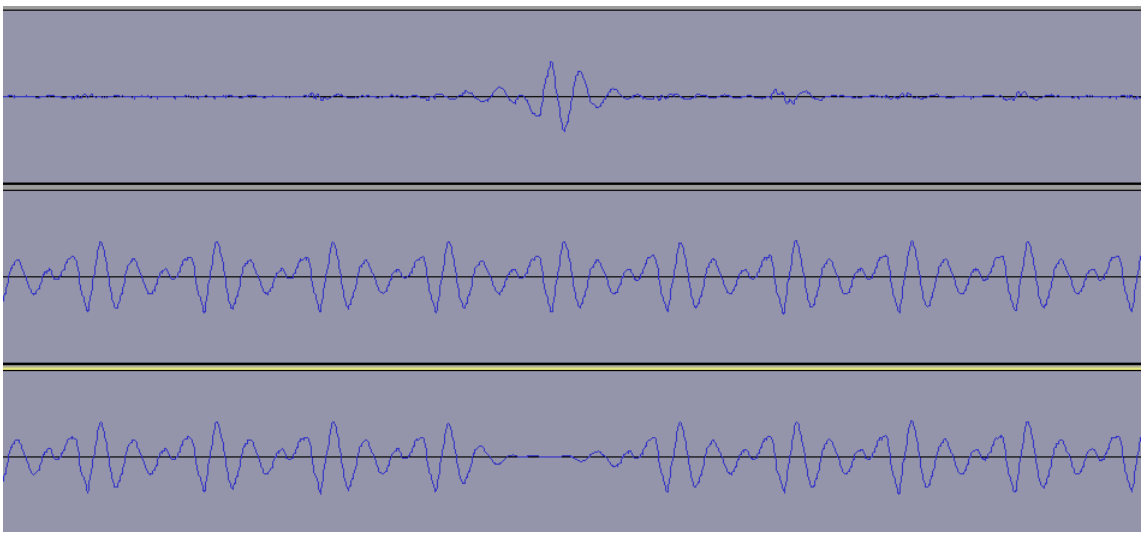

Abbildung 2.11: Kopiersynthese mit WBVPM: Die Energie des Residuals ist sehr gering. Die erste Wellenform ist das Residual, darunter befindet sich das ursprüngliche Signal und dessen Kopie. Ein Puls fehlt in der Kopie, daher erscheint dieser im Residual.

man die Abstände zwischen den Glottispulsen verändert. Daher verhält sich WBVPM ähnlich wie TD-PSOLA, wenn das Spektrum unverändert übernommen wird.

Für den schmalbandigen Ansatz (SMS und NBVPM) ist die wichtigste Fehlerursache das nichtstationäre Signal. Größere Abweichungen in der fundamentalen Frequenz verursachen daher auch größere Fehler. Dadurch, dass bei WBVPM nur ein oder zwei Perioden des Signals analysiert werden, wird erzwungen, dass das Signal über diesen kurzen Zeitraum stationär ist. Daher lässt sich diese Art von Fehlern durch WBVPM verringern. (siehe Abb.2.11) Im Gegensatz dazu sind die Fehler des breitbandigen Ansatzes (z.B. Phase-Vocoder) durch die schlechte Frequenzauflösung verursacht, da hier die Hauptkeule des Analysefensters mehrere Harmonische breit ist und diese dann nicht mehr unterscheidbar sind. Bei WBVPM ist die mögliche Frequenzauflösung maximal, da das Analysefenster genau eine Periodenlänge umfasst. Die Hauptfehlerquelle bei WBVPM ist daher die Ungenauigkeit in der Messung der fundamentalen Frequenz, da die Fensterlänge und Position davon abhängen. Daher ist es wichtig, dass die fundamentale Frequenz möglichst akkurat ermittelt wurde. Ist die ermittelte Frequenz ungenau oder falsch, so ist in stimmhaften Segmenten Phasiness zu hören, da die vorhergesagten Voice-Pulse-Onsets nicht mit den tatsächlichen übereinstimmen. Zusätzlich ist das Spektrum dann verschmiert, da eine nicht ganzzahlige Anzahl Perioden analysiert wird. Dadurch entsteht eine Abweichung zwischen den gemessenen harmonischen Frequenzen und den realen Harmonischen, welche mit höheren Frequenzen auch deutlich mehr ins Gewicht fällt (siehe Abb. 2.12).

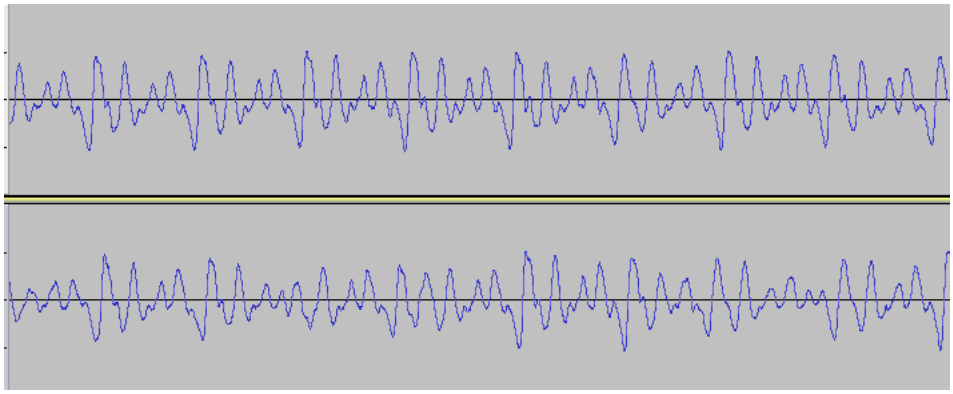

Abbildung 2.12: Phasiness durch geringfügig veränderte f_0. Oben ist das Ursprungssignal dargestellt, darunter das Ergebnis der Kopiersynthese mit verfälschter f_0

Bei den niedrigen Frequenzen kann dieser Effekt noch durch die Interpolation ausgeglichen werden, bei den höheren jedoch nicht. Auch Sprünge in der fundamentalen Frequenz führen zu solchen Fehlern.

Im vereinfachten Modell der harmonischen Phasenhüllkurve am Voice-Pulse-Onset ist diese flach und fällt linear unter jedem Formanten ab. Da der Vokaltrakt ständig in Bewegung ist, verschieben sich auch die Formantfrequenzen und damit der Phase-Shift. Dadurch ändert sich auch die Frequenz der Harmonischen geringfügig, auch unter der Annahme, dass die fundamentale Frequenz konstant ist. Daher ist ein Sprachsignal in Wirklichkeit inharmonisch. Die Abweichung vom idealen harmonischen Spektrum hängt vor allem von der Bandbreite des Formants und dessen Bewegungsgeschwindigkeit ab. Insbesondere in phonetischen Übergängen sind die Abweichungen sehr groß, so dass hier bei WBVPM Fehler auftreten. In der Praxis ist der Phase-Shift jedoch nicht linear, so dass sich die Abweichungen nicht einfach berechnen lassen und jede Frequenzkomponente eine eigene Abweichung hat. (siehe Abb. 2.13) Aus diesem Grund wird ein Phasenmodell benötigt, welches diesen nichtlinearen Phase-Shift beschreibt.

Da Hintergrundrauschen sich nicht auf die fundamentale Frequenz auswirkt, ist WBVPM gegenüber diesem robust. Bei NBVPM und SMS dagegen kann sich dieses negativ auf die Harmonic Trajectories auswirken, so dass dann Artefakte zu hören sind.

Mit WBVPM lassen sich ohne große Änderungen am Algorithmus auch stimmlose Segmente bearbeiten, bei WORLD wird dasselbe Verfahren angewendet. Das gemessene Signal enthält dann nur Rauschen und keine harmonischen Anteile:

> „Unvoiced signals can be processed as if they were voiced by assigning an arbitrary fundamental frequency. “ [26, Seite 142]

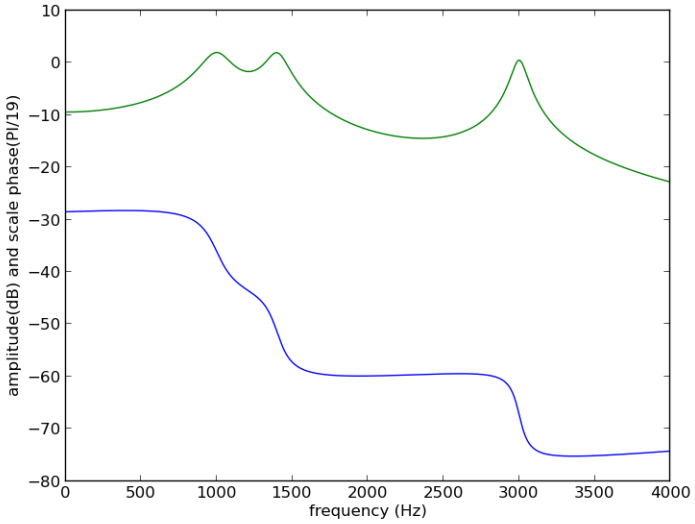

Abbildung 2.13: Vereinfachtes Phasenmodell am Voice-Pulse-Onset

Insgesamt erhält man mit WBVPM bessere Ergebnisse als mit anderen spektralen Verfahren, gleichzeitig lassen sich aber auch die Vorteile von Time-Domain Algorithmen nutzen. So ist die zeitliche Auflösung gegenüber NBVPM und SMS verbessert. Da das Signal nicht in zwei Teile zerlegt wird, hat man hier etwas weniger Flexibilität gegenüber NBVPM:

> „WBVPM is able to model voice pulses in frequency domain with a set of sinusoids, which represent both harmonic and noisy components. It provides an independent control of each single pulse, thus allowing pulse sequence transformations with ease. This ability is typical of time-domain methods, but complex to achieve in frequency domain, since it implies dealing with complex subharmonics patterns. “ [26, Seite 142]

2.5 STRAIGHT und TANDEM-STRAIGHT

Die Algorithmen STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHT spectrum) [18] und dessen Nachfolger TANDEM-STRAIGHT [22, 23] zerlegen ein Sprachsignal in eine periodische und eine aperiodische Komponente. Ähnlich wie bei WBVPM muss die fundamentale Frequenz vorher bestimmt worden sein, um die spektrale Hüllkurve und die Gewichte zu bestimmen. Für den periodischen Teil des Sprachsignals wird dabei ein periodischer Pulse-Train mit der Länge T0 verwendet, während für den aperiodischen Teil weißes Rauschen verwendet wird. Beide Komponenten werden über die Gewichte miteinander vermischt, so dass man ein teilweise periodisches Anregungssignal erhält.

Im ersten Schritt wird die spektrale Hüllkurve aus dem Powerspektrum berechnet. Dafür wird über eine geeignete Fensterfunktion ein kurzes Zeitfenster ausgeschnitten. Dabei spielt es keine Rolle, ob das Fenster auf einem CGI zentriert ist oder nicht, da die Phaseninformation verworfen wird. Dann wird das Signal in den Frequenzbereich transformiert und das so gewonnene Powerspektrum anschließend logarithmiert. Dieses Powerspektrum enthält jedoch noch Interferenzen, welche durch die Periodizität des Anregungssignals verursacht werden. Als Fensterlänge dient ein ganzzahliges Vielfaches der Periodenlänge. Da die Länge der Fensterfunktion bekannt ist, ist es möglich diese zu normalisieren, so dass die Fläche unter der Fensterfunktion den Wert „1“ hat. Als Fensterfunktion sind z.B. Hanning, Blackman, Nuttall oder Kaiser geeignet. Für die Weiterverarbeitungen müssen die temporalen Schwankungen entfernt werden und das Spektrum muss geglättet werden, damit man eine Hüllkurve erhält. Um die Schwankungen zu entfernen, wird ein Fenster der Länge 3 Perioden verwendet. Längere Fensterlängen führen zu einer erhöhten Fehleranfälligkeit.

Im nächsten Schritt müssen durch die Periodizität verursachte Interferenzen entfernt werden. Das temporal stabile Spektrum enthält noch Peaks, da hier eine harmonische Struktur zugrunde liegt. Diese Peaks sind auch eine Folge des Leck-Effekts, welcher unvermeidbar auftritt. Daher muss das Spektrum weichgezeichnet werden. Dies kann z.B. mit einem Rechteckfenster der Länge f_0 im Frequenzbereich geschehen. Eine andere Möglichkeit ist die zusätzliche Verwendung von Cepstral-Liftering .

Nachdem die Hüllkurve bestimmt wurde, wird noch die Periodizität des Signals bestimmt. Die Periodizität ist hier im Gegensatz zum Powerspektrum nicht mehr erkennbar und beide Spektren sind multiplikativ verknüpft. Daher lässt sich die Periodizität am Quotienten beider Spektra ermitteln. In der Nähe der fundamentalen Frequenz und der harmonischen Frequenzen befinden sich Peaks. Wenn ein Signal jedoch teilweise inharmonisch ist, so weichen die gemessenen harmonischen Frequenzen von den berechneten Peaks ab. Anhand dieser Abweichung werden die Gewichte berechnet, mit denen periodische und aperiodische Impulsantworten gemischt werden. Dabei bleibt die Robustheit gegenüber Background-Noise erhalten, gleichzeitig ist die zeitliche Auflösung gegenüber SMS und NBVPM verbessert.

2.6 Sprachsynthese mit WORLD

WORLD [23] ist ein Framework zur Analyse und Synthese von Sprache, welches auf dem Vocoder und dem Source Filter Modell basiert. Damit lässt sich Sprache synthetisieren, die genauso natürlich klingt wie die ursprünglichen Sprachaufnahmen. Das Framework zerlegt Sprache in 3 Komponenten: f_0, spektrale Hüllkurve und Anregungssignal. Diese Parameter lassen sich unabhängig voneinander modifizieren und zur Resynthese benutzen. WORLD ist freie Software unter der BSD-Lizenz und kann daher als Phonemerzeuger für eSpeak verwendet werden.

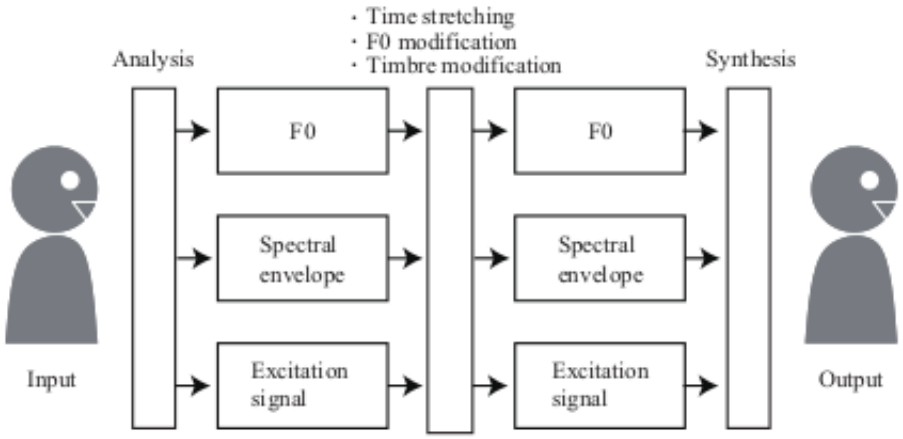

Abbildung 2.14: Überblick über das WORLD-Framework; aus [23]

2.6.1 Bestimmung der fundamentalen Frequenz mit DIO

Distributed Inline-filter Operation (DIO) ist ein Verfahren, mit dem sich die Periodenlänge effizient und möglichst genau berechnen lässt. Deren Kehrwert ist die fundamentale Frequenz f_0. Ungenaue oder falsche Werte für f_0 führen zu hörbaren Fehlern. Bei einem einfachen Sinusoid kann man die Frequenz durch Messen der Abstände zwischen den Nulldurchgängen errechnen. Harmonische Signale können weitere Nulldurchgänge haben, welche das Messergebnis verfälschen. Filtert man alle Obertöne oberhalb der fundamentalen Frequenz heraus, so ist nur die fundamentale Frequenz messbar. Dabei wird ein Nuttall-Fenster als Tiefpassfilter eingesetzt. Da dessen benötigte Cutoff-Frequenz unbekannt ist, werden verschiedene Frequenzkandidaten getestet und anhand der Periodizität bewertet. Der Kandidat mit der größten Periodizität wird als fundamentale Frequenz angenommen. Als Maß für die Periodizität dient die Standardabweichung. DIO ist schneller als andere Algorithmen, da hier keine Frame-By-Frame Analyse durchgeführt wird, sondern die Faltung nur einmal für jede Cutoff-Frequenz mit der Fouriertransformation durchgeführt wird.

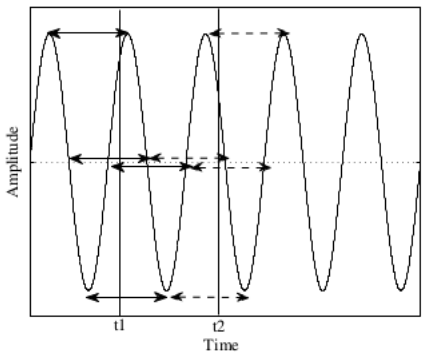

Abbildung 2.15: Bestimmung der fundamentalen Frequenz mit DIO. Das Signal wurde zuvor tiefpassgefiltert und enthält daher keine Frequenzen oberhalb von f_0; aus [23]

Bei längeren Signalen ist DIO jedoch ungeeignet, hier kann man das Cepstrum oder die Autokorrelation verwenden. Bei beiden Transformationen erhält man wieder ein Zeitsignal, an welchem die harmonische Struktur sichtbar ist. Ein erster Peak ist bei der Position t_0, daher lässt sich damit auf die fundamentale Frequenz f_0 schließen.

2.6.2 Bestimmung der spektralen Hüllkurve mit STAR

Da stimmhafte Sprache eine f_0 hat, ist die Wellenform periodisch. Wie bei SMS wird über ein Fenster, welches 3 Perioden lang ist, ein Teil des Signals ausgeschnitten. Über die STFT lässt sich ein Amplitudenspektrum bestimmen, welches jedoch Peaks an den harmonischen Frequenzen enthält. Anschließend wird erst der Logarithmus, danach das Integral berechnet. Dadurch erhält man eine fallende Kurve, welche zur Berechnung einer weichen Hüllkurve benutzt werden kann, welche keine Peaks mehr enthält.

Diese Kurve wird einmal nach links und einmal nach rechts verschoben, und zwar genau um die fundamentale Frequenz. Dann berechnet man die Differenz aus der linken und der rechten Kurve und teilt diese durch f_0. Durch die Subtraktion gelangt man vom Integral zurück in die ursprüngliche Darstellung. Danach wird die Logarithmierung umgekehrt, da spätere Verarbeitungsprozesse lineare Werte erwarten. Auf diese Weise lässt sich die Hüllkurve akkurat berechnen, während bei LPC die Nullstelle in der Transferfunktion nicht erkannt wird.

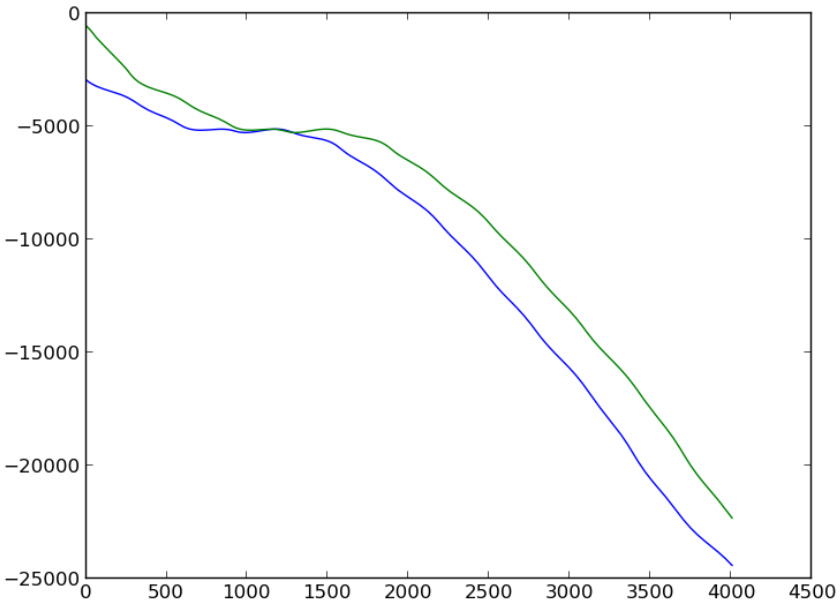

Abbildung 2.16: High Levels und Low Levels als verschobene Integrale des Spektrograms.

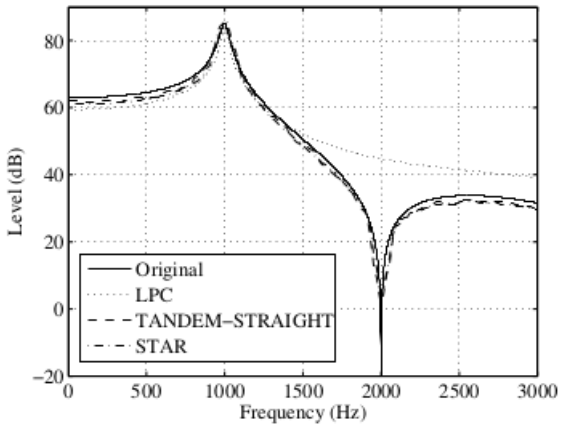

Abbildung 2.17: Bestimmung der spektralen Hüllkurve mit TANDEM-STRAIGHT, STAR und LPC. Diese besteht hier aus einer Polstelle (dem Maximum links) und einer Nullstelle (dem Minimum rechts) in der Transferfunktion. Die Differenz aus beiden Graphen geteilt durch f_0 ist die spektrale Hüllkurve in dB; aus [23]

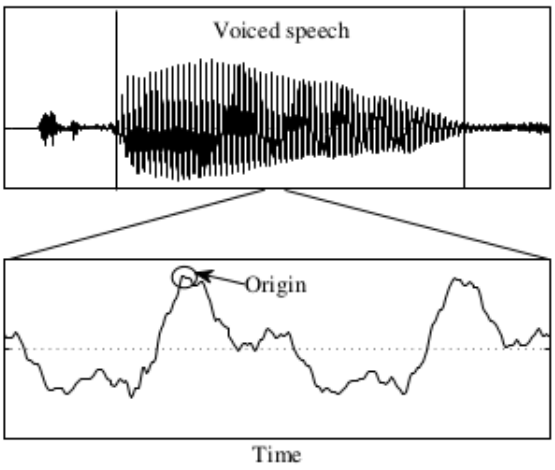

Abbildung 2.18: Bestimmung des ersten Voice-Pulse-Onsets mit dem PLATINUM-Algorithmus; aus [23]

2.6.3 Extraktion des Anregungssignals mit PLATINUM

PLATINUM berechnet das Anregungssignal aus der gefensterten Wellenform sowie der spektralen Hüllkurve und der fundamentalen Frequenz. In einem typischen Vocoder wird der Dirac-Impuls als Anregungssignal und das minimalphasige Spektrum als Impulsantwort des Vokaltraktes verwendet. Beim STRAIGHT-Vocoder dagegen wird das Anregungssignal als Gemisch aus einem periodischen Pulse-Train und weißem Rauschen modelliert. Dabei werden beide Komponenten getrennt berechnet und für den periodischen Teil wird das minimalphasige Spektrum verwendet. Durch die Gewichtung lässt sich die Sprachqualität verbessern, jedoch erreicht man nicht die von TD-PSOLA bekannte Natürlichkeit, da hier die Phaseninformation verloren geht. Für einen guten Klang ist es erforderlich, die ursprüngliche Phase des Anregunssignals zu ermitteln. Dafür wird das Spektrum Y(ω) über eine Fensterfunktion ermittelt, welche genau zwei Perioden lang ist und auf einem Voice-Pulse-Onset zentriert (siehe Abb. 2.18) ist. Zur Berechnung von deren Positionen wird das Integral der fundamentalen Frequenz und die Position des ersten Pulses innerhalb eines stimmhaften Segments verwendet. Für den ersten Puls wird zusätzlich nach einem lokalen Maximum gesucht.

Berechnung der fundamentalen Phase: $\varphi_0(x) = \int_{-\infty}^{+\infty} f_0(x)$

Berechnung des Anregungssignals: [23] $X(\omega) = \frac{Y(\omega)}{H(\omega)}$

Dann teilt man das Spektrum Y(ω) durch das minimalphasige Spektrum der zuvor berechneten STAR-Hüllkurve H(ω) und erhält so das Anregungssignal X(ω). Diese Art der Berechnung bezeichnet man als inverses Filtern. Unter der Voraussetzung, dass Y(ω)

gemischtphasig ist, ist X(ω) maximalphasig. Im Idealfall sind hier die Phasenunterschiede zwischen zwei benachbarten Pulsen minimal, solange der aperiodische Anteil am Sprachsignal nicht zu groß ist. Das Spektrum von X(ω) ist sehr flach, wenn die spektrale Hüllkurve korrekt ermittelt wurde.

2.7 Sprachdatenkompression

Da die Analyse mit WORLD sehr große Datenmengen (etwa 12 MB pro Sekunde an Rohdaten) liefert, ist eine passende Kompression sinnvoll. Eine unkomprimierte WORLD Sprachdatenbank, bestehend aus etwa 5 Minuten Sprache, wäre dann etwa 3,6 GB groß. Dies ist zu viel für den praktischen Einsatz. Da die Rohdaten sehr viel Redundanz enthalten, lassen sie sich auch gut komprimieren. Die komprimierte Sprachdatenbank „Namine Ritsu Connect“ [4] ist etwa 400 MB groß. Teilweise lassen sich Sprachdatenbanken ohne hörbare Qualitätsverluste auch noch viel kleiner komprimieren, da insbesondere auch das Anregungssignal sehr viel Redundanzen enthält.

2.7.1 MFCC

Amplitudenspektra wie z.B. das STAR-Spektrum lassen sich sehr gut komprimieren, indem man Mel Frequency Cepstral Coefficients (MFCC) berechnet. Diese lassen sich auch als unterscheidbares Merkmal zur Spracherkennung nutzen, welche z.B. sinnvoll ist um eine Sprachdatenbank zu labeln. Auch die japanische Sprachausgabe HTS, welche z.B. von NVDA genutzt wird, verwendet MFCCs. Nachteilig bei MFCCs ist, dass die Phaseninformation verloren geht.

Zur Berechnung wird das Spektrum logarithmiert und in die Mel-Skala transformiert. Danach wird das transformierte Spektrum über die Fouriertransformation in den Zeitbereich transformiert. Das Spektrum liegt dann als eine Reihe von Koeffizienten vor, von denen nur die ersten gespeichert werden müssen. Durch die Entfernung weiterer Koeffizienten wird das Spektrum geglättet, dies wird auch als Liftering bezeichnet. Durch die Glättung des Spektrums lassen sich die lokalen Maxima und Minima einfacher berechnen, da deren maximale Anzahl von der Anzahl der Koeffizienten abhängt. 32 Koeffizienten sind in dem meisten Fällen ausreichend um den groben Verlauf des Spektrums darzustellen.

2.7.2 Vorbis

Vorbis ist ein universeller und verlustbehafteter, von der Xiph.Org Foundation entwickelter Audiocodec, welcher sowohl Sprache als auch Musik und andere Audiodaten komprimieren kann. Da das mit PLATINUM ermittelte Restsignal phasenbehaftet ist, kann es nicht mit MFCCs komprimiert werden. Aus diesem Grund wird das Restsignal

in den Zeitbereich transformiert und mit Vorbis komprimiert. Alternativ kann man auch die Faltung aus dem Vokaltrakt-Filter und dem Anregungsignal bilden und diese mit Vorbis komprimieren. Dadurch erreicht man noch kleinere Sprachdatenbanken, jedoch kann dies bei der Resynthese zu Phase-Mismatches führen, da die Phasen des Residuals nicht einander angeglichen sind. Für stimmlose Segmente ist kein Phase-Mismatch möglich, daher kann man hier die Impulsantwort des Sprachsignals mit Vorbis komprimieren, ohne dass es zu Artefakten kommt. Um Phase-Mismatches zu verhindern kann man allen Frames dieselbe konstante Phase geben und diese dann einzeln mit Vorbis komprimieren. Alternativ kann man auch ein resynthetisiertes harmonisches Signal mit konstanter Phase und f_0 komprimieren. Dieses kann dann mit TD-PSOLA bearbeitet werden, ohne dass Phase-Mismatches auftreten.

2.8 Spektrale Stimmenmodelle

Der Vokaltrakt ist ständig in Bewegung. Dadurch ändern sich auch die Formantfrequenzen und die Phasenantwort des Vokaltrakt-Filters. Selbst unter der Annahme, dass f_0 konstant ist, ist die Stimme inharmonisch, da die Phasenverschiebung zwischen zwei Frames Änderungen in den harmonischen Frequenzen verursacht. In der Wirklichkeit ist die f_0 jedoch niemals konstant, so dass dadurch weitere inharmonische Anteile entstehen. Viele Sprachsynthese-Algorithmen setzen aber ein harmonisches (und daher auch periodisches) Signal voraus, da sich nur bei harmonischen Signalen die Phasen einfach vorhersagen lassen können. Lässt man inharmonische Anteile weg, so erhält man ein Sprachsignal, welches dem ursprünglichen ähnlich klingt, in einigen Fällen sind jedoch größere Unterschiede hörbar. Bei einigen Sprechern ist der inharmonische (oder aperiodische) Teil stärker ausgeprägt als bei anderen. Ein Beispiel für einen Sänger, bei dem dies der Fall ist, ist Louis Armstrong [26, Seite 168].

2.8.1 Excitation plus Resonances (EpR)

EpR ist ein Modell, welches das Anregungssignal und die Resonanzen des Vokaltrakts getrennt modelliert. Zusätzlich gibt es noch ein Residual, welches den Unterschied zwischen dem berechneten EpR-Modell und dem urprünglichen harmonischen Spektrum beschreibt. Das Modell besteht aus 3 Filtern in einer Kaskade: Das erste Filter modelliert das Anregungssignal, das zweite modelliert den Vokaltrakt und das dritte addiert die Unterschiede in Dezibel, so dass man das ursprüngliche Signal erhält, wenn keine Transformationen vorgenommen werden. Damit lässt sich die Magnitudenhüllkurve eines Sprachsignals beschreiben. Zwischen zwei harmonischen Frequenzen wird das harmonische Spektrum interpoliert. Die Phase wird durch EpR nicht modelliert, da EpR ein reines Amplitudenmodell ist.

EpR Source Filter

Die Voice-Source wird als eine exponentiell fallende Kurve im Frequenzbereich und einer Resonanz modelliert. Diese Resonanz wird auch als glottaler Formant [15] oder Source-Resonanz [26] bezeichnet. Die fallende Kurve ist durch eine Verstärkung (Gain) und zwei Brightness Werte (Slope und Slopedepth) definiert und wird folgendermaßen dargestellt:[26]

Frequenzgang der Source-Kurve:$Source_{db}(f) = Gain_{db} + SlopeDepth_{db}(e^{Slope \cdot f} - 1)$

Durch lineare Regression der harmonischen Frequenzen können die Werte berechnet werden. Die Source-Resonanz modelliert das Spektrum im tiefen Frequenzbereich unterhalb des ersten Formants. Im Gegensatz zu den Vokaltrakt-Resonanzen verursacht diese keine Phasenverschiebung, da diese Teil des Anregungssignals ist und nicht die Resonanz einer Röhre modelliert.

EpR Vocal Tract Filter

Das Vokaltrakt-Filter wird als eine Menge von Resonanzen $R_0 \dots R_N$ und einem Residual modelliert. Jede Resonanz wird als ein symmetrischer Bandpassfilter zweiter Ordnung modelliert, auf dieselbe Weise wie auch die Source-Resonanz modelliert wird. Zusätzlich wird im vereinfachten Modell noch ein linearer Phase-Shift hinzugefügt. Die Beiträge der einzelnen Resonanzen werden als Linearwerte addiert, danach erfolgt eine Umwandlung in Dezibel. Zum Schluss wird noch die residual Envelope addiert, um Fehler des Modells auszugleichen. [26, Seite 126] Wenn keine Transformationen vorgenommen werden, so erhält man mit diesem Modell wieder das ursprüngliche harmonische Spektrum. Der Vorteil gegenüber der Verwendung einer Hüllkurve ist, dass man Resonanzen unabhängig voneinander verschieben kann. Dies ist insbesondere bei der Konkatenation ein Vorteil, da man hier die natürlichen Formantbewegungen erhalten kann, ohne dass sich die Hüllkurve abflacht. Für EpR werden zwei Instanzen des Modells verwendet: eines modelliert das harmonische Spektrum, das andere modelliert das SMS-Residual, welches die inharmonischen Frequenzen enthält. Das Gesamtverhalten des Systems wird folgendermaßen modelliert: [26]

$$EpR_{db}(f) = Source_{db}(f) + 20log_{10} \sum_{i=0}^{M} R_i(f) + S_{Residual_{dB}}(f)$$

2.8.2 Minimal- und maximalphasiges Spektrum

Klassische Vocoder (LPC und FFT basiert) verwenden ausschließlich das minimalphasige Spektrum. Neuere Forschungen führen zu einer Mixed-Phase-Darstellung des Spektrums unter der Annahme, dass ein Anregungssignal mit einem Vokaltrakt-Filter gefaltet wird.

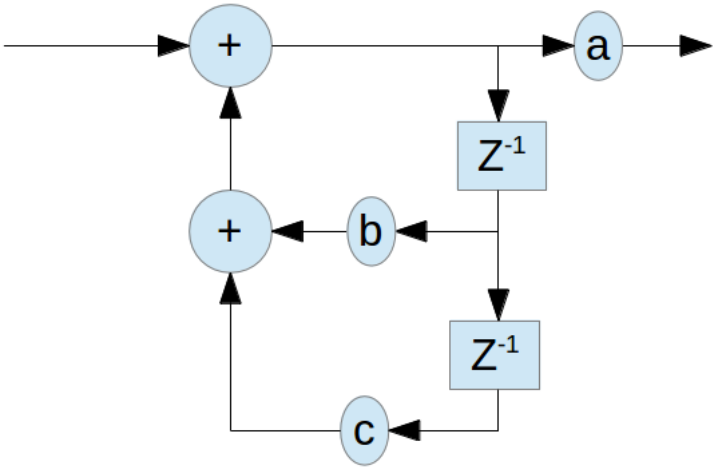

Abbildung 2.19: Formantfilter zweiter Ordnung zur Sprachsynthese. Dieser Filtertyp wird vom Klatt-Synthesizer und dem EpR-Modell verwendet. Da es sich um ein IIR-Filter handelt, lassen sich Phase und Amplitude nicht getrennt modellieren. Daher ist das Filter minimalphasig.

Dabei wird das Vokaltrakt-Filter durch ein minimalphasiges Spektrum dargestellt, während das Anregungssignal maximalphasig ist. Traditionell wird das Vokaltrakt-Filter als IIR-Filter mit einer kleinen Anzahl von Koeffizienten modelliert (siehe Abb. 2.19). Die Filterkoeffizienten lassen sich aus der Autokorrelation ermitteln. Dieses Verfahren wird auch als Linear Predictive Coding (LPC) bezeichnet.

Eine andere in WORLD verwendete Möglichkeit das minimalphasige Spektrum zu beschreiben, ist die Anwendung des Cepstrums. In beiden Fällen ist es nicht möglich, Frequenzgang und Phasengang getrennt zu modellieren, da Imaginärteil und Realteil über die Hilberttransformation [27] zusammenhängen. Als Anregungssignal dient häufig ein Pulse-Train für periodische Signale und weißes Rauschen für nicht periodische Signale. Ein auf diese Weise erzeugtes künstliches Anregungssignal enthält jedoch nicht die Phase des ursprünglichen Signals, da die periodische Impulsantwort als minimalphasiges Spektrum dargestellt wird und Sprachsignale in der Realität jedoch gemischtphasig sind. Um das maximalphasige Spektrum von dem bereits bekannten minimalphasigen Spektrum zu trennen, verwendet man inverses Filtern. Damit lässt sich das Anregungssignal extrahieren, ohne dass die Phase verfälscht wird.

Da das Vokaltrakt-Filter minimalphasig ist, lässt sich dieses über eine reell-wertige Folge beschreiben. Dieses ist dann kausal und stabil und alle Pole und Nullstellen liegen innerhalb des Einheitskreises. [27] Da das maximalphasige Spektrum durch inverses Filtern gewonnen wird, befinden sich die Pole und Nullstellen des Anregungssignals außerhalb oder auf dem Einheitskreis. Das Anregungssignal selbst besteht wiederum aus einem kausalen und einem antikausalen Teil. Der CGI ist die Stelle, wo sich die Verbindung zwischen beiden Teilen befindet. Wenn eine antikausale Komponente vorhanden ist, so

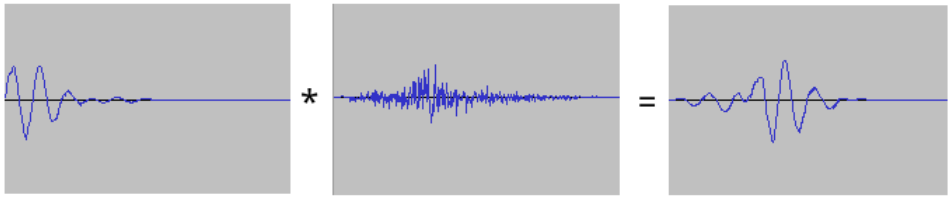

Abbildung 2.20: Darstellung eines Sprachsignals als Faltung eines minimalphasigen Vokaltrakt-Filters mit einem maximalphasigen Anregungssignal: $H(x) * X(x) = Y(X)$. Der Stern $*$ steht für die Faltungsoperation und nicht für die Multiplikation.

ist ein Signal maximalphasig oder gemischtphasig.[15, Seite 37] Der Teil links vom CGI hat die Form einer Resonanz-Impulsantwort, wobei diese auf dem CGI anfängt und von dort rückwärts läuft. Eine solche Impulsantwort ist nicht mit einem digitalen Filter realisierbar, da die Ausgabe des Filters vor dem Triggerpuls am Eingang erscheint. Der Teil rechts vom CGI ist kausal und enthält keine Resonanz. Beide Teile lassen sich durch ein Tiefpassfilter modellieren.

Die Verwendung des gemischtphasigen Modells ist gleichbedeutend mit der Annahme, dass es zwei unterschiedliche Typen von Resonanzen gibt: mehrere Vokaltrakt-Resonanzen und eine Source-Resonanz, welche auch als glottaler Formant bezeichnet wird. Der glottale Formant ist dabei antikausal, während alle andere Resonanzen kausal sind. Nur die kausalen Resonanzen verursachen im EpR-Modell einen Phase-Shift. [26]

Das maximalphasige Anregungssignal wird immer als FIR-Filter modelliert, während man für das minimalphasige Vokaltrakt-Filter z.B. ein MLSA-Filter verwenden kann. Alternativ kann man auch die komplexen Spektren multiplizieren, um die Faltung zu erhalten. Der Vorteil von FIR-Filtern ist, dass sich hier Phasengang und Frequenzgang unabhängig voneinander modellieren lassen. Nachteilig ist die höhere Anzahl der Koeffizienten gegenüber einem IIR-Filter. In der z-Transformation hat ein FIR-Filter nur Nullstellen und keine Pole und bei IIR-Filtern ist dies umgekehrt. Modelliert man den Voice-Pulse als FIR-Filter, so lassen sich dessen Nullstellen in zwei Hälften aufteilen: Für den minimalphasigen und kausalen Teil liegen die Nullstellen innerhalb des Einheitskreises und für die antikausale Allpasskomponente auf oder außerhalb des Einheitskreises. Dieses Trennverfahren wird auch als „zeros of the z-transform (ZZT)“ [15] bezeichnet. Da der Grad des Polynoms, dessen Nullstellen berechnet werden sollen, mit der Samplerate steigt, funktioniert dieses Verfahren nur bei niedrigen Samplerates. Stimmen mit einer höheren Samplerate müssen auf 16 kHz heruntergerechnet werden. Verwendet man dagegen die Fouriertransformation, so kann man auch mit höheren Sampleraten arbeiten. Das in WORLD verwendete Verfahren (siehe Abb. 2.20) ist gegenüber ZZT deutlich robuster, da hier keine Nullstellen berechnet werden müssen und die Faltung im Frequenzbereich durchgeführt wird und nicht in der z-Transformation. Außerdem ist ZZT dafür bekannt, ein Rauschen in dem abgetrennten Anregungssignal zu produzieren,

insbesondere in den höheren Frequenzen. WORLD dagegen produziert kein hörbares Rauschen, da die Amplituden des künstlich hinzugefügten Rauschens sehr gering sind. Da WORLD mit dem Cepstrum arbeitet und der Logarithmus für 0 nicht definiert ist, wird ein weißes Rauschen mit sehr geringer Amplitude auf das Eingangssignal addiert, wodurch das Spektrum nur Werte größer Null enthält. Zusätzlich enthält auch das Eingangssignal noch ein unvermeidbares Hintergrundrauschen. Aus diesem Grund ist das Rauschen immer im maximalphasigen Teil sichtbar.

2.8.3 Das Hüllkurven-Phasenmodell

Bonada schlägt in [26, Seite 164] ein Phasenmodell vor, welches hier als Hüllkurven-Phasenmodell bezeichnet wird, um es von anderen Phasenmodellen zu unterscheiden. In diesem Modell wird die Phase nicht in zwei Bestandteile zerlegt, sondern ausschließlich aus der spektralen Hüllkurve generiert. Dies hat den Vorteil, dass man die Phaseninformation nicht in der Sprachdatenbank speichern muss und so eine kleinere Datenbanken enthält. Außerdem wird dadurch die Konkatenation vereinfacht, da dann keine Phasenkonkatenation mehr durchgeführt werden muss. Die Konkatenation und auch die Kodierung der Phasen ist besonders fehleranfällig, da kleine Fehler hier zu hörbaren Phasenauslöschungen führen. [16, Seite 238] Durch Verwendung dieses vereinfachten Modells können keine Phase-Mismatches mehr entstehen, da es hier nur eine Phase und nicht zwei unterschiedliche Phasen gibt. Wie bei der Zerlegung in ein minimal- und maximalphasiges Spektrum wird die Phase hier immer am Voice-Pulse-Onset gemessen. Das Ziel ist hier nicht, die Phase mit möglichst wenig Fehlern vorherzusagen, sondern einfach eine gegenüber der urprünglichen ähnlich klingenden Phasenhüllkurve zu generieren. Die Phasenhüllkurve wird nicht aus dem Resonanzmodell gewonnen, da dieses nicht sehr robust ist. Insbesondere würden hier neu auftretende oder verschwindende Resonanzen zu Phase-Mismatches führen. Daher verwendet man die gesamte breitbandige spektrale Hüllkurve, welche auch das EpR-Residual enthält, wenn EpR verwendet wird. Ursprünglich wurde die Phasenhüllkurve aus der skalierten und verschobenen Timbre-Hüllkurve gewonnen. Da dieses Verfahren nicht bei sehr vielen Stimmen funktioniert, wurde ein anderes Verfahren entwickelt, bei dem die Phase aus der Ableitung der logarithmierten Amplitude gewonnen wird. Skaliert man diese Hüllkurve, so erhält man eine gute Approximation der Phase für die niedrigen Frequenzen. Laut [26] funktioniert dieses Verfahren bei einer großen Anzahl von Stimmen. Auf diese Weise lässt sich ein minimalphasiges Spektrum berechnen, ohne dass dafür das Cepstrum verwendet werden muss. Das Phasenmodell wird nur für den harmonischen Teil des Spektrums verwendet, da die Phasen des inharmonischen Spektrums Zufallswerte sind. Dabei ist $\hat{\phi}_h$ der Phasenunterschied zwischen zwei Frequenzen und α ein konstanter Faktor. [26, Seite 165]

$$\hat{\phi}_h = \alpha 20 log_{10}(\frac{a_{h+1}}{a_h})$$

Die Auswahl des Faktors α ist dabei von entscheidender Bedeutung, da bei einem schlecht

gewählten Faktor Phasiness zu hören sein kann. Experimente der Music Technology Group (MTG) der Universität Pompeu Fabra, Barcelona haben ergeben, dass $\pi/19$ ein guter Wert ist [26, Seite 165] und sich das ursprüngliche Phasenspektrum nur geringfügig von dem Modell unterscheidet, zumindest für die niedrigen Frequenzen. Bei höheren Frequenzen kann vorhandenes Rauschen zu Fehlern führen, insbesondere auch bei WBVPM.

3 Erstellung einer deutschen Diphone-Sprachdatenbank

Bestandteil dieser Arbeit ist die Erstellung einer Sprachdatenbank für die deutsche Sprache. Diese besteht aus Samples eines Sprechers, welche mit den in Kapitel 2 behandelten Verfahren transformiert werden können. Dieser Sampling-Ansatz hat sich in der Praxis bewährt und wird z.B. vom Festival Speech Synthesis System [11] verwendet. Die in Kapitel 2 behandelten Verfahren sind jedoch neuer und bieten eine Reihe von Vorteilen gegenüber den von Festival verwendeten Algorithmen. Durch Sampling lässt sich ein Großteil der möglichen Klänge (Sonic Space) eines Musikinstruments oder Sprechers modellieren. Dabei können jedoch nur die Klänge reproduziert werden, die auch aufgenommen wurden. So wird auf Kosten der Flexibilität eine größere Natürlichkeit erreicht. Sprecherspezifische Merkmale werden erhalten, so ist z.B. bei der Festival-Stimme AWB ein schottischer Akzent zu hören. Eigentlich ist der Sonic Space ein unendlichdimensionaler Vektorraum. In der Praxis wird aber mit einer Vereinfachung gearbeitet, die den Sonic Space in endliche Subspaces aufteilt, die größtenteils unabhängig voneinander modelliert werden können. Bei Sprachsynthese ist vor allem die phonetische Achse von Bedeutung, bei Gesang kommen weitere Achsen hinzu. [26]

3.1 Hardware/Software Voraussetzungen

Da der Autor dieser Arbeit mit GNU/Linux arbeitet, wird dieses auch für die Erstellung einer Sprachdatenbank verwendet. Die meisten Werkzeuge sind jedoch portabel und laufen auch unter anderen Betriebssystemen. Einzig das Soundsystem ALSA ist nur zusammen mit einem Linux-Kernel nutzbar. Im Anhang finden sich Verweise auf die verwendete Software. So wurde z.B. die Software WORLD ursprünglich unter Windows entwickelt und dann auf freie Systeme portiert. Für die Erstellung der Sprachdatenbank kommen auch vermehrt Python-Skripte zum Einsatz, teilweise wird auch Numpy und Matplotlib genutzt.

Für die Aufnahmen wird ein hochwertiges Mikrofon benötigt. Für Sprach- und Gesangsaufnahmen sind Großmembranmikrofone sehr gut geeignet, da diese eine hohe Empfindlichkeit besitzen und gegenüber anderen Mikrofonen weniger Rauschen liefern. Am besten sind dabei Mikrofone mit USB-Anschluss, da diese fast immer class compliant sind. Man kann aber auch ein externes Interface verwenden, wenn dieses mit ALSA kompatibel ist.

Da die meisten Bearbeitungsschritte sehr viel CPU-Zeit benötigen, ist es wichtig ein aktuelles Rechnersystem mit möglichst schnellem Prozessor zu verwenden. Insbesondere erfordert die spektrale Analyse sehr viele Rechenschritte, welche annähernd in Echtzeit durchgeführt werden können, wenn das System schnell genug ist. Andere Prozesse können schon mal mehr als eine Stunde dauern. Es ist ebenfalls sehr wichtig, genügend freien Festplattenspeicher zu haben, da einige Prozesse sehr große temporäre Dateien anlegen.

3.2 Wiederverwendung von eSpeak

Die freie Sprachausgabe eSpeak [1] unterstützt bereits die deutsche Sprache, damit sind auch die Menge der Phoneme und die Ausspracheregeln bereits festgelegt. Die Qualität der synthetischen Sprache hängt von dem Sound-Rendering-Verfahren und der verwendeten Sprachdatenbank ab. Da eSpeak Formantsynthese und für alle Sprachen eine gemeinsame Formanttabelle verwendet, klingt die Stimme sehr robotisch. Die für den natürlichen Klang notwendige Koartikulation ist dabei nicht berücksichtigt.

Durch Verwendung von konkatenativer Synthese lässt sich dagegen ein sehr natürlicher Klang erreichen. Da eSpeak bereits über eine Infrastruktur zur Ansteuerung von konkatenativen Phonemerzeugern verfügt, muss nur das Sound-Rendering-Modul angepasst werden und eine Sprachdatenbank erstellt werden. Die anderen Komponenten können unverändert übernommen werden. Um eine Sprachdatenbank zu erstellen, sind folgende Schritte notwendig [11]:

- Auswahl der Phoneme
- Konstruktion der Diphones
- Synthese der Prompts
- Aufnahmen der Prompts
- Labeln der Phoneme und Diphones in den Aufnahmen
- Spektrale Analyse der Aufnahmen

3.3 Phoneme der deutschen Sprache

Für die Darstellung der Phoneme einer Sprache wird häufig die SAMPA-Notation verwendet. Dabei handelt es sich um eine Darstellung des Internationalen Phonetischen Alphabets nur mit ASCII-Zeichen. Die folgende Tabelle beinhaltet alle im Deutschen vorkommenden Phoneme in SAMPA-Notation:

Gruppe	Symbol
stimmlose Plosive	p t k
stimmhafte Plosive	b d g
stimmlose Affrikate	tS ts pf
stimmlose Frikative	f s S h
stimmhafte Frikative	v z Z x C
Nasale (immer stimmhaft)	m n N
Liquida	l R
Halbvokale	j
Stille	_
lange (betonte) Vokale	i: y: u: E: e: 2: o: a: aI OY aU
kurze (unbetonte) Vokale	I U Y E @ 9 O a 6

Tabelle 3.1: Phonemgruppen der deutschen Sprache, selbst erstellt auf Basis der Darstellung in [26, Seite 181]

Es gibt keinen einheitlichen Phoneset für die deutsche Sprache. Der glottale Stop [[?]] am Silbenanfang ist in dieser Tabelle absichtlich weggelassen worden, um die Sprachdatenbank nicht zu groß werden zu lassen. Stellt man Stille mit einer Länge weniger Millisekunden vor einen Vokal am Silbenanfang, so lässt sich die Funktion des glottalen Stops emulieren. In einigen Varianten des Deutschen, insbesondere in Bayern, wird zwischen zwei oder mehr Qualitäten des [[a]] unterschieden, im Standarddeutschen dagegen gibt es nur eines. Auch für die Liquida [[l]] und [[R]] gibt es verschiedene Aussprachen, die sich von Sprecher zu Sprecher unterscheiden. Im Deutschen wird auch zwischen stimmhaften und stimmlosen /s/ unterschieden, dies ist im Schriftbild jedoch nicht direkt zu erkennen. Für das stimmhafte /s/ wird die Notation [[z]] verwendet. Teilweise werden noch einige englische und französische Phoneme für deutsche Sprachausgaben verwendet, was für Lehnwörter von Bedeutung ist. Aus Gründen der Einfachheit werden hier aber nur die deutschen Phoneme verwendet. Einige Vokale können auch betont und unbetont, gespannt oder ungespannt vorkommen. Dafür benötigt man zwei unterschiedliche Schreibweisen. Der Doppelpunkt steht dabei immer für die lange Variante eines Vokals.

Da sehr viele Sprecher in Deutschland auch die englische Sprache beherrschen, kann man auch die englischen Phoneme beim Entwurf der Sprachdatenbank berücksichtigen und eine bilinguale Sprachdatenbank erstellen. Dafür sucht man nach Schnittmengen der Phonesets zweier Sprachen. Dabei ist zu berücksichtigen, dass einige Konsonanten im Deutschen anders klingen als im Englischen und es auch Unterschiede zwischen den verschiedene Varianten des Englischen gibt. Insbesondere weicht der Phoneset des amerikanischen Englischen von dem des britischen Englischen ab.

Bei einer lautgetreuen Sprache wie Deutsch kann man einfach alle Buchstaben und Digraphen des Alphabets nehmen und auf phonetische Symbole abbilden. Für die Vokale /ö/ und /ü/ werden hier die Phonetischen Symbole [[2:]] und [[9]] sowie [[Y]] und [[y:]] verwendet. Einige Vokale verschmelzen jedoch mit benachbarten Konsonanten zu einem

neuen Phonem oder Allophon. So ist die Ausprache von /er/ nicht immer einheitlich, häufig verschmelzen beide zu einem Schwa , welches als [[6]] dargestellt wird. Ein kurzes unbetontes /e/ wird ebenfalls als Schwa realisiert, dieses klingt dann jedoch anders und wird als [[@]] geschrieben. Für Sprachen, welche nicht lautgetreu sind, wird immer ein phonetisches Wörterbuch benötigt. Bei einigen Sprachsynthesizern (z.B. VOCALOID [26, Seite 225ff]) kann der Nutzer neue Wörter in das Wörterbuch eintragen. Häufig wird der regelbasierte Ansatz mit einem Wörterbuch kombiniert. Für die englische Sprache stellt die Carnegie Mellon University ein freies phonetisches Wörterbuch zur Verfügung [26, Seite 183].

3.4 Konstruktion der Diphone-Liste und Synthese der Prompts

Die Grundidee der Diphone-Synthese ist es, explizit alle Phonem-Phonem-Übergänge aufzulisten. Dabei wird vereinfachend angenommen, dass Koartikulationseffekte sich niemals über mehr als zwei Phoneme erstrecken. [11, Seite 101] Die dadurch mögliche Anzahl der Diphones ist immer die Quadratzahl der Phoneme einer Sprache. Aus den 43 Phonemen der deutschen Sprache erhält man so 1849 mögliche Diphones. In der Praxis reicht jedoch eine etwas geringere Anzahl aus, da nicht alle theoretisch möglichen Diphones in einer Sprache vorkommen. Hat man ausreichend lange Texte, so lässt sich nach einer Transkription in die phonetische Darstellung die Häufigkeit der einzelnen Diphones berechnen. Dabei wird man feststellen, dass einige Diphones mit einer Häufigkeit von 0.002% sehr selten sind und andere überhaupt nicht in dem Text vorkommen. Dies schließt jedoch nicht aus, dass diese seltenen Diphones in anderen Texten vorkommen. In natürlicher Sprache ist sehr viel Redundanz enthalten, daher konstruiert man einen synthetischen Corpus, welcher im Ideallfall jedes Diphone genau einmal enthält. Man kann auch ganze Sätze verwenden, wobei jeder Satz eine bestimmte Anzahl weiterer Diphones hinzufügt. Ein Beispiel für einen solchen Corpus ist die freie CMU-Arctic-Sprachdatenbank [10], welche aus Stimmen von unterschiedlichen Sprechern besteht. Ein Projekt, welches Sprachdaten sammelt, um damit eine Spracherkennung zu verbessern, ist Voxforge [21]. Im Gegensatz zur Sprachsynthese benötigt man zur Spracherkennung einen sehr großen Corpus, welcher sehr viele Redundanzen enthält. Damit lässt sich dann ein akustisches Modell erstellen, welches sich zur Spracherkennung aber auch zur Synthese nutzen lässt. Ein Beispiel für eine solche HMM-basierte Sprachausgabe ist HTS [17].

Um für die deutsche Sprache einen Corpus zu erstellen, wird ein Python-Skript verwendet, welches alle sinnvollen Diphone-Kombinationen auflistet. Jeder Prompt besteht aus ein oder zwei Diphones sowie einem Trägervokal, welcher als Koartikulationsperre dient und die Diphones nach links und rechts begrenzt. Um die Prompts zu synthetisieren wurde MBROLA mit der Stimme de2 verwendet. Dabei ist die fundamentale Frequenz auf einen konstanten Wert festgelegt und der Sprecher kann diese mit möglichst ähn-

Verzeichnis	Aufgabe
bin	ausführbare Skripte
down	Wellenform mit niedriger Samplingrate
etc	Recording-Skripte, Konfigurationsdateien
prompts	für die Aufnahme verwendete Prompts
spec	spektrale Darstellung der Recordings
tmp	temporäre Dateien
wav	Aufnahmen im Wave-Format

Tabelle 3.2: Verzeichnisstruktur der Sprachdatenbank, die im Rahmen dieser Arbeit erstellt wurde

lichen Eigenschaften nachsprechen. Auch die Sprechgeschwindigkeit und die Länge der Segmente ist festgelegt. Auf diese Weise erhält man möglichst konsistente Samples. Auch lassen sich so Fehler in der Aussprache reduzieren. Das Skript exportiert auch die Transkriptionen der Samples, welche später verwendet werden, um die Sprachdaten von Hand zu labeln. Damit kann sichergestellt werden, dass die phonetischen Symbole in der richtigen Reihenfolge den Segmenten zugeordnet werden können. Die Verzeichnisstruktur der Stimmdatenbanken wurde größtenteils vom Festival Speech Synthesis System übernommen, während für das Labeling das von VOCALOID und MBROLA verwendete SAMPA-Alphabet verwendet wird.

3.5 Aufnahme des Corpus

Nachdem der Corpus entworfen wurde und die Prompts gerendert wurden, kann mit der Aufnahme begonnen werden. Dabei ist es wichtig, mit einer möglichst reproduzierbaren Recording-Umgebung zu arbeiten, so dass die Aufnahme unterbrochen und später fortgesetzt werden kann. Der Abstand zwischen Sprecher und Mikrofon muss so gewählt werden, dass der Nahbesprechungseffekt [3] nicht auftritt und bei Explosivlauten keine Bursts sichtbar werden [15]. Teilweise lassen sich diese durch die Verwendung eines Pop-Killers vermeiden, es ist jedoch besser zusätzlich einen konstanten Abstand zwischen dem Sprecher und dem Mikrofon von etwa 80cm zu verwenden [15]. Die Verstärkung des Mikrofons wird zu Beginn ebenfalls auf einen konstanten Wert eingestellt, welcher zuvor experimentell ermittelt wurde. Die Lautstärke des Mikrofons wird so eingestellt, dass das Signal nicht übersteuert wird und das Hintergrundrauschen nicht hörbar ist. Preprocessing wie z.B. eine Rauschentfernung wird nicht durchgeführt, da dadurch auch das Nutzsignal degradiert wird.

Schon bei der Aufnahme soll möglichst viel automatisiert werden, da eine manuelle Qualitätskontrolle während der Aufnahme fehleranfällig oder gar unmöglich ist. Um eine möglichst konstante Tonhöhe zu erreichen, wird das Recording mittels eines Stimulus durchgeführt. Dabei wird ein Prompt vorgelesen, den der Sprecher dann nachspricht.

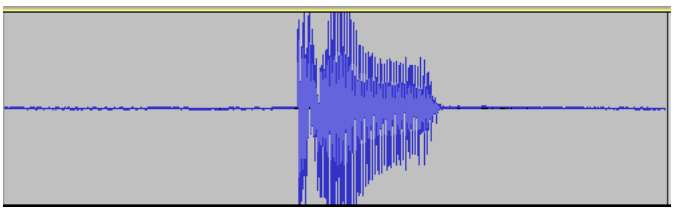

Abbildung 3.1: Hintergrundrauschen und Bursts bei einer fehlerhaften Aufnahme

Dann hört der Sprecher seine eigene Stimme, aber mit einer konstanten fundamentalen Frequenz. Dann wird eine weitere Aufnahme durchgeführt. Diese zwei Schritte wiederholen sich solange bis die Abweichung der Tonhöhe vom Sollwert ein Minimum unterschreitet. Danach wird der Prompt unter einer Nummer abgespeichert, welche laufend inkrementiert wird. Zum Schluss kann noch eine automatische Vorsegmentierung, z.B. mit dem Dynamic Time Warping Algorithmus durchgeführt werden. [26] Durch dieses als „Assisted-Recording“ [15, Seite 64] bekannte Verfahren lässt sich die statistische Verteilung der Tonhöhen in einer Sprachdatenbank verringern, so dass bei der Konkatenation weniger Artefakte auftreten. Ziel des Verfahrens ist es einen Speech-Corpus zu erstellen, welcher möglichst wenig Unterschiede in der Stimmenqualität enthält. Über ein Python-Skript lässt sich der gesamte Corpus, oder Teile davon, zur Qualitätskontrolle abspielen. Dabei wird immer erst der Prompt und dann das Recording abgespielt. Danach wird der Median-Wert der fundamentalen Frequenz angezeigt.

Um eine Sprachdatenbank zu erstellen ist es notwendig eine große Anzahl von Audio-Segmenten zu segmentieren, zu beschriften und zu analysieren [26, Seite 184]. Für die deutsche Sprache sind dies etwa 1400 Segmente. Daher ist es wichtig, möglichst viele Schritte mittels Skripten zu automatisieren.

Im ersten Schritt muss der Corpus in viele kleine Dateien zerlegt werden. Dies kann z.B. manuell geschehen, oder man kann längere Pausen als Worttrenner nutzen. Da dieses Verfahren sehr zeitaufwändig und fehleranfällig ist, ist es besser während der Aufnahme bereits eine Vorsegmentierung durchzuführen. Dann muss die Zuordnung von Segment zu Transkription nicht mehr von Hand durchgeführt werden. Im nächsten Schritt wird dann eine phonetische Segmentierung durchgeführt, entweder manuell oder mittels einer automatischen Spracherkennung. Für eine Spracherkennung werden akustische Modelle benötigt, ein Modell für die deutsche Sprache wird von Voxforge [21] bereitgestellt. Danach muss noch eine Diallophon-Segmentierung durchgeführt werden. Dabei sollten Grenzen möglichst auf stabile Frames gelegt werden, deren spektrale Hüllkurve sich nicht oder nur geringfügig ändert. Ein Maß für die Stabilität ist die normalisierte Summe von absoluten MFCC-Differenzen zwischen zwei benachbarten Frames.

Nachdem die Segmentierung abgeschlossen ist, werden die Samples spektral analysiert, damit spätere Transformationen effizient durchführbar sind. Dabei werden sehr große temporäre Dateien angelegt, da hier das Residual noch nicht komprimiert wird. Später wird das Residual dann mit Vorbis komprimiert. Im letzten Schritt wird noch eine manu-

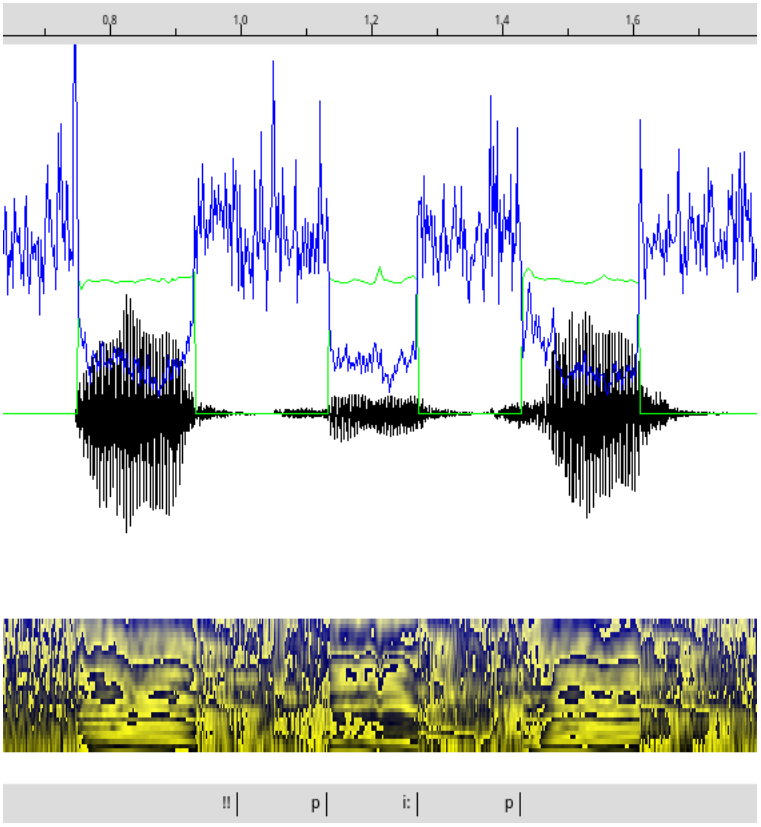

Abbildung 3.2: Darstellung der Bedienoberfläche der im Rahmen dieser Masterarbeit erstellten Software zur Segmentierung: die blaue Kurve stellt die Stabilität dar und die grüne die fundamentale Frequenz. Schwarz dargestellt ist die Wellenform des Sprachsignals und unten das Spektrogramm: die Intensität der Gelbtöne stellt die Sättigung bei den harmonischen Frequenzen dar. Weitere Erläuterungen siehe Text.

elle Kontrolle der Qualität der Samples durchgeführt, dabei werden fehlerhafte Samples markiert. Einige Fehler fallen jedoch nur auf, wenn man Transformationen vornimmt und Samples konkateniert. Die Visualisierung der konkatenierten Samples ermöglicht es, Fehler in den Samples oder den Konkatenationsalgorithmen festzustellen. Zusätzlich ist es möglich sich die Samples anzuhören, da nicht alle Sampleeigenschaften grafisch dargestellt werden können und nicht jede Art von hörbaren Fehlern in der Visualisierung festgestellt werden kann. Durch Verwendung von Skripten lassen sich viele dieser Schritte wiederholen und Tests können mit unterschiedlichen Algorithmen durchgeführt werden.

Damit bei der Konkatenation keine Amplituden-Mismatches auftreten, müssen die Samples bei der Erstellung der Sprachdatenbank in der Lautstärke angepasst werden. Für jedes Phonem müssen die Lautstärken der Grenzregionen vereinheitlicht werden, so dass dabei koartikulationsbedingte Lautstärkeunterschiede erhalten bleiben. Im ersten Schritt

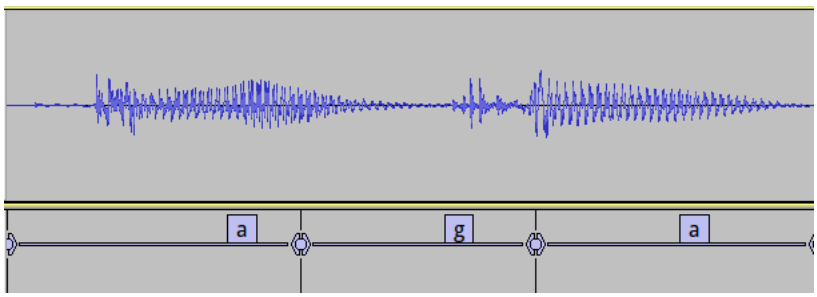

Abbildung 3.3: Durch Koartikulation verursachte Lautstärkeunterschiede. Einzelheiten siehe Text.

werden die durchschnittlichen Lautstärken für jedes Phonem ermittelt. Danach werden alle Phoneme im Grenzbereich auf die vorher bestimmte Lautstärke normalisiert. Aufnahmebedingte Pegelunterschiede können auf diese Weise jedoch nicht ausgeglichen werden, daher ist es wichtig bei der Aufnahme größere Schwankungen der Lautstärke zu vermeiden.

Da die auf diese Weise erstellte Sprachdatenbank als komplexe Verzeichnisstruktur vorliegt und sehr groß ist, kann diese nicht direkt an Endbenutzer ausgeliefert werden. Daher wird aus den Sprach- und Labeldaten eine einzige Datei erstellt, welche sehr viel kleiner ist als die temporären Dateien. f_0 und die spektrale Hüllkurve der benötigten Segmente können einfach übernommen werden, das Residual muss erst in den Zeitbereich transformiert werden. Im Zeitbereich lassen sich stille Bereiche des Signals wegschneiden, so dass die Datenbank weniger Redundanzen enthält. Danach wird das Residual mit Vorbis komprimiert, wodurch nicht hörbare Frequenzen entfernt werden. Dabei bleiben die Phasen der hörbaren Frequenzen erhalten, die Qualität der Segmente verschlechtert sich damit nicht. Zum Schluss werden die Labeldaten und die Sprachdaten in die Datei geschrieben, welche an den Benutzer ausgeliefert wird.

3.6 Sound Rendering

Das Sound-Rendering-Modul oder der Phonemerzeuger ist die Komponente, welche im letzten Verarbeitungsschritt den Klang erzeugt. Die Klangerzeugung wird durchgeführt, indem Samples aus einer Datenbank transformiert und konkateniert werden. Als Eingabe dient eine Low-Level-Transkription der Sprachdaten, welche für jedes Phonem die Länge und eine Pitch-Envelope für stimmhafte Laute enthält.

Die Art und Weise, wie die Transformationen durchgeführt werden, hängt von den verwendeten spektralen Modellen ab, die der Synthesizer verwendet. Viele der in Kapitel 2 vorgestellten spektralen Modelle ermöglichen Transformationen wie Time-Scaling oder eine Änderung der fundamentalen Frequenz und Lautstärke, ohne dass die Klangfarbe sich ändert. Gleichzeitig sind Änderungen der Klangfarbe unabhängig von den vorher ge-

```
#Aufruf
espeak -v mb-de2 "hallo welt" --pho
#Ausgabe
h       85
a       27      0 119 80 111 100 111
l       65
o:      35      0 111 80 107 100 107
v       65
E       51      0 102 80 75 100 75
l       65
t       101
_       301
_       1
```

Tabelle 3.3: Transkription mit eSpeak [1]

nannten Transformationen möglich. Die Konkatenation beinhaltet auch eine Änderung der Klangfarbe in den Randbereichen der Samples. Da der reale Sonic Space immer größer ist als die Abstraktion, welche zu Beginn dieses Kapitels eingeführt wurde [26, Seite 199], lassen sich Samples nicht direkt miteinander verbinden, ohne dass es zu hörbaren Artefakten kommt. So beschreibt die phonetische Achse die Klangfarbe nur sehr grob. Verbindet man Samples mit gleicher phonetischer Transkription, unterscheiden sich deren Formantfrequenzen. Auch bei der fundamentalen Frequenz kommt es zu einer fehlerhaften Fortsetzung. Ein Verfahren, welches diese Fehler korrigiert, wird als „correction spreading“ bezeichnet. Für jede Eigenschaft des Sprachsignals wird als erstes der Unterschied zwischen zwei Grenzframes berechnet. Danach wird dieser berechnete Wert gewichtet auf beide Seiten aufaddiert, wobei die Gewichte auf einer Seite positiv und auf der anderen Seite negativ sind. Dieses Verfahren lässt sich sowohl mit MFCCs als Timbre-Deskriptoren durchführen als auch mit dem EpR-Modell. Ein Vorteil des EpR-Modells ist es, dass man hier Formantfrequenzen unabhängig voneinander verändern kann. Wenn keine künstliche Hüllkurve für die fundamentale Frequenz verwendet wird, so lässt sich das Verfahren auch für die fundamentale Frequenz unverändert durchführen.

Damit bei der Konkatenation keine Phase-Mismatches auftreten, müssen auch die Phasen der harmonischen Frequenzen eine kontinuierliche Forsetzung ergeben, die nicht an den Segmentgrenzen unterbrochen werden. Im einfachsten Fall, wenn die Phase des Ausgangssignals nur von der fundamentalen Frequenz und der Klangfarbe abhängt, kann es daher keine Phase-Mismatches geben.

Zerlegt man die Phase in zwei Teile, so lässt sich ein Phase-Mismatch nur im minimalphasigen Teil des Spektrums vermeiden. Dieses beinhaltet auch alle durch Formanten bedingte Phase-Shifts. Das maximalphasige Anregungssignal ist dagegen schwieriger zu modellieren und Transformationen sind hier nur begrenzt durchführbar. Eine einfache Möglichkeit das Residual zu konkatenieren, ist ein Time-Domain-Linear-Smoothing. Das

Algorithmus 3.1 Die selbst erstellte Konkatenation mit dem "correction spreading"-Algorithmus in Anlehung an [26, Seite 200]

```
double diff = voice_feature[right_start] - voice_feature[left_end];

for(int i=left_start;i<left_end;i++) {
        double weight = (i-left_start)*1.0/(length);
        voice_feature[i] += 0.5*diff*weight;
}

for(int i=right_start;i<right_end;i++) {
        double weight = 1.0-((1+i-right_start)*1.0/(length));
        voice_feature[i] -= 0.5*diff*weight;
}
```

Rauschen auf den höheren Frequenzen tritt nur im maximalphasigen Spektrum auf, wo Phase-Mismatches kaum noch hörbar sind. Neben Phasenmodulationen enthält das Residual auch Amplitudenmodulationen, welche ebenfalls die Gesamtklangfarbe prägen. Daher kann es zu Artefakten kommen. wenn das ursprüngliche Sprachsignal starke Amplitudenmodulation enthält.

Verwendet man das EpR-Modell, so muss dessen Residual unabhängig konkateniert werden. Dabei wird die Differenz aus zwei benachbarten EpR-Instanzen berechnet, wobei vorher ein Mapping der Formantfrequenzen durchgeführt wurde. Dadurch unterscheiden sich beide Modelle nur in den Frequenzen zwischen den Formanten. Dieses Differenzsignal wird über ein Gewicht auf das bereits korrigierte harmonische Spektrum addiert. Dadurch verringern sich Fehler in der Fortsetzung der Wellenform. Eine andere Möglichkeit ist die Parametrierung der Wellenform des Anregungssignals, geeignete Modelle sind z.B. das Liljencrants-Fant-Modell (Fant 1986). [26, Seite 23]

4 Schlussfolgerungen

Diese Masterarbeit beschäftigt sich mit spektralen Modellen zur Sprachsynthese und darauf aufbauenden Samplingverfahren. Viele dieser Modelle basieren auf dem Vocoder, welcher ein Sprachsignal in überlappende Bänder zerlegt. Häufig wird dabei die Phase des Anregungssignals verworfen, was zu einer Verschlechterung des Klangs führt.

Eine andere Methode zur Sprachsynthese ist TD-PSOLA, mit der sich ein recht natürlicher Klang erreichen lässt. Zur konkatenativen Sprachsynthese ist diese Methode jedoch nicht flexibel genug. Ein neuartiger Ansatz, welcher Vorteile von Vocodern mit denen von TD-PSOLA kombiniert, ist der PLATINUM-Algorithmus, welcher in Kapitel 2 vorgestellt wurde.

Traditionelle spektrale Modelle zerlegen Sprachsignale in eine harmonische und eine inharmonische Komponente, wobei die Phase der inharmonischen Komponente verworfen wird. Neuere Algorithmen zerlegen ein Sprachsignal in ein minimalphasiges Spektrum, welches den Vokaltrakt modelliert, und ein Anregungssignal, welches immer maximalphasig ist. Beide Komponenten können getrennt voneinander verarbeitet werden. Zu ihrer Komprimierung werden unterschiedliche Algorithmen verwendet. Durch die Zerlegung in drei Komponenten, hat WORLD auch Vorteile gegenüber WBVPM und NBVPM. Einerseits ist die zeitliche Auflösung gegenüber NBVPM verbessert, andererseits hat man eine größere Flexibilität als bei WBVPM. eSpeak verwendet kein solches Modell, sondern greift für stimmhafte Segmente auf Formantsynthese zurück und verwendet Sprachaufnahmen nur für stimmlose Konsonanten. Durch die Verwendung eines gemischtphasigen Modells ist es daher möglich, Sprachaufnahmen auch für stimmhafte Segmente zu verwenden und die Nachteile der Formantsynthese zu vermeiden.

In Kapitel 3 wird die Erstellung einer Sampling-basierten, konkatenativen Sprachdatenbank, was Gegenstand dieser Masterarbeit war, beschrieben. Dabei wird auf ein spektrales Modell zurückgegriffen, welches es ermöglicht, einzelne Komponenten des Sprachsignals unabhängig voneinander zu transformieren. Auf diese Weise lässt sich ein sehr natürlich klingender Sprachsynthesizer bauen, welcher eine phonetische Low-Level-Transkription als Eingabe erhält. Im ersten Schritt muss die von eSpeak erstellte Transkription verarbeitet werden, der dafür benötigte Parser ist Teil der im Rahmen dieser Masterarbeit geschriebenen Software. Danach werden Sprachsegmente aus einer Sprachdatenbank entnommen und anschließend mit der „correction spreading“-Methode konkateniert. Ein erster Test zeigte, dass aufgrund dieser Methode keine Artefakte zu hören sind. Jedoch könnten bisher unentdeckte Fehler in der Sprachdatenbank oder in der Software zu hörbaren Fehlern führen. Die zeitlichen Vorgaben für die Erstellung ei-

ner Masterarbeit reichten für das hier behandelte Thema nicht aus, um alle notwendigen Tests durchzuführen. Hierzu müsste nun die Sprachdatenbank im produktiven Einsatz benutzt werden, in dem längere Textfolgen als Eingaben verwendet werden (im Rahmen dieser Arbeit wurden die Tests nur mit einzelnen Sätzen durchgeführt). So ist es möglich auch seltene Fehler zu finden. Danach ist es möglich für weitere Sprachen Sprachdatenbanken zu erstellen.

Das in Kapitel 2 vorgestellte EpR-Modell lässt sich ebenfalls verwenden, um realistische Formantbewegungen zu modellieren. Im Gegensatz zum All-Pole Filtermodell, welches von eSpeakEdit verwendet wird, lassen sich hier Amplituden, Bandbreiten und Frequenzen der Vokaltrakt-Resonanzen getrennt modellieren. Zusätzlich lassen sich auch Antiformaten modellieren. Daher ist es sinnvoll eSpeak um das EpR-Modell zu erweitern, da man mit EpR die Vorteile von Formantsynthese und konkatenativer Synthese nutzen kann. Zur Berechnung der harmonischen Phasen kann das Hüllkurven-Phasenmodell verwendet werden, so dass man einen kleineren Rechenaufwand gegenüber dem Cepstrum hat. Bisher wird diese Methode nicht in eSpeak verwendet. Um die Effizienz der Klangsynthese von eSpeak zu verbessern kann die Time-Domain Oszillatorbank, durch die in Kapitel 2 beschriebene IFFT-Methode ersetzt werden. Da eSpeak momentan nicht sehr modular ist, ist es sinnvoll bei der Implementierung dieser Verbesserungen auch ein Refactoring vorzunehmen.

HMM basierte Sprachausgaben haben den Vorteil, dass die Sprachdatenbanken sehr klein sind und die Sprache relativ natürlich gegenüber Formantsynthese klingt. Mit Hilfe einer Spracherkennung, welche dieselben statistischen Modelle verwendet, lässt sich der zeitliche Aufwand zur Erstellung einer Sprachdatenbank senken. Unter Umständen ließe sich auch die Anzahl der Fehler in der Sprachdatenbank vermindern. Da Voxforge sehr große Speech-Corpera zur Spracherkennung sammelt, lassen sich diese möglicherweise auch verwenden, um ein akustisches Modell für die Sprachsynthese zu erstellen.

Auch bei der Modellierung und Komprimierung des Anregungssignals ist noch sehr viel Spielraum für Verbesserungen. Dafür bieten sich insbesondere stochastische Modelle an, welche eine geringere Anzahl von Parametern verwenden, die unabhängig voneinander transformiert werden können. So lassen sich z.B. kausale und antikausale Komponenten unabhängig voneinander modellien. Eventuell kann durch ein verbessertes Modell auch auf die Vorbis-Komprimierung verzichtet werden, was insbesondere für Echtzeitsynthese von Bedeutung ist.

Glossar

All-Pole Filter Filter, dessen Transferfunktion nur Pole und keine Nullstellen enthält. Wird auch als IIR-Filter bezeichnet, da die Impulsantwort (theoretisch) unendlich lang ist. 10, 42

Allophon Variante eines Phonems. 49, 51

Cepstrum neue Transformation eines Singals in der Nachrichtentechnik, 1963 von Bogert, Healy und Tukey eingeführt: Anagramm von Spectrum. 17, 36, 42, 44

Diphon Kombination zweier benachbarter Phoneme. 8, 47, 49

Dirac-Impuls ein diskretes Signal, welches den Wert "1" an der Stelle "0" hat, ansonsten "0": wird zur Berechnung der Impulsantwort eines Systems genutzt, enthält alle Frequenzen. 25, 38

Envelope-Follower Hüllkurven-Demodulator zur Verfolgung der Lautstärke eines Signals über die Zeit. 10, 13

Formant bestimmte Frequenzen, welche durch den Vokaltrakt verstärkt werden. 13, 19, 23, 32, 54

Glottis die menschlichen Stimmlippen. 10, 23

Koartikulation Beinflussung der Artikulation eines jeden Sprachlauts durch die ihn umgebenden Sprachlaute. 10, 47, 49, 52

Laryngograph medizinisches Gerät zur Messung des Hautwiderstandes am Kehlkopf. 5, 24

Liftering Glättung des Spektrums unter Verwendung des Cepstrums. 34, 39

Mel-Skala eine logarithmische Skala für die Tonhöhe, bei niedrigen Frequenzen annähernd linear. 39

Phasiness Hörbarkeit von Fehlern in der Phase eines Signals. 23, 25, 27, 32, 45

Phonem Laut einer Sprache. 8, 9, 34, 47

Pulse-Train ein periodisches Signal, bestehend aus mehreren Dirac-Impulsen. 33, 38

Schwa Bezeichnung für den Zentralvokal, benannt nach einem hebräischen Buchstaben. 49

Speech Corpus eine oder mehrere Sprachaufnahmen, welche zur Sprachsynthese- oder Erkennung genutzt werden können. 9

Timbre-Hüllkurve auch als spektrale Hüllkurve bezeichnet, beschreibt die Klangfarbe eines Signals. 44

Vokaltrakt die Hohlräume von Mund und Nase. 10, 11, 27

Literaturverzeichnis

[1] Espeak. Available online at http://espeak.sourceforge.net/; visited on 2014-08-02.

[2] Interspeech 2007 - synthesis of singing challenge. Website. Available online at http://www.interspeech2007.org/Technical/synthesis_of_singing_challenge.php; visited on 2014-06-27.

[3] Nahbesprechungseffekt. Website. Available online at https://de.wikipedia.org/wiki/Nahbesprechungseffekt; visited on 2014-07-18.

[4] Namine ritsu connect. Website. Available online at http://www.canon-voice.com/connect.html; visited on 2014-06-27.

[5] Sinsy - a hmm-based singing voice synthesis system. Website. Available online at http://sinsy.sourceforge.net/; visited on 2014-06-27.

[6] v.connect-stand. Website. Available online at http://hal-the-cat.music.coocan.jp/ritsu_e.html; visited on 2014-06-27.

[7] Vocoder. Wikipedia. Available online at https://de.wikipedia.org/wiki/Vocoder; visited on 2014-06-10.

[8] Wolfgang von kempelen. Wikipedia. Available online at https://de.wikipedia.org/wiki/Wolfgang_von_Kempelen; visited on 2014-06-27.

[9] Stephan M. Bernsee. Pitch shifting using the fourier transform. Website, 1999. Available online at http://www.dspdimension.com/admin/pitch-shifting-using-the-ft/ visited on 2014-08-03.

[10] Alan W. Black. Cmu arctic. Website. Available online at http://www.festvox.org/cmu_arctic/ visited on 2014-08-03.

[11] Alan W. Black and Kevin A. Lenzo. *Building Synthetic Voices*. Available online at http://festvox.org/ visited on 2014-08-03.

[12] Paul M. Brossier. *Automatic Annotation of Musical Audio for Interactive Applications*. PhD thesis, Centre for Digital Music Queen Mary, University of London, 2006. Available online at http://aubio.org/phd/thesis/brossier06thesis.pdf visited on 2014-08-03.

[13] João Paulo Serrasqueiro Robalo Cabral. *HMM-based Speech Synthesis Using an Acoustic Glottal Source Model*. PhD thesis, The Centre for Speech Technology

Research Institute for Communicating and Collaborative Systems School of Informatics University of Edinburgh, 2011. Available online at https://www.era.lib.ed.ac.uk/bitstream/1842/4877/1/Cabral2011.pdf visited on 2014-08-03.

[14] Crypton Future Media, Inc. Hatsune miku. Available online at http://www.crypton.co.jp/download/pdf/info_miku_e.pdf; visited on 2014-04-01.

[15] Nicolas d'Alessandro. *Realtime and Accurate Musical Control of Expression in Voice Synthesis*. PhD thesis, University of Mons, Belgium, 2009. Available online at http://www.nicolasdalessandro.net/phd/phd-print.pdf; visited on 2014-08-03.

[16] Thierry Dutoit. *An introduction to text-to-speech synthesis*. Kluwer Academic Publishers, 1997.

[17] HTS Working Group. Hts: Hmm-based speech synthesis system. Website. Available online at http://hts.sp.nitech.ac.jp/ visited on 2014-07-18.

[18] Hideki Kawahara, Masanori Morise, and Ken-Ichi Sakakibara. Interference-free observation of temporal and spectral features in "shout" singing voices and their perceptual roles. In *Proceedings of the Stockholm Music Acoustics Conference 2013, SMAC 2013, Stockholm, Sweden*, 2013. Available online at http://iwk.mdw.ac.at/lit_db_iwk/download.php?id=18108 visited on 2014-04-01.

[19] Dennis H. Klatt. Software for a cascade/parallel formant synthesizer. Technical report, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1980.

[20] Jon Lederman. The fourier transform - diagonalizing the convolution operator. Available online at http://www.science20.com/jon_lederman/fourier_transform_diagonalizing_convolution_operator visited on 2014-05-13.

[21] Ken MacLean. Voxforge. Website. Available online at http://voxforge.org/; visited on 2014-07-18.

[22] Masanori Morise. Platinum: A method to extract excitation signals for voice synthesis system. Technical report, College of Information Science and Engineering, Ritsumeikan University,1–1–1 Nojihigashi, Kusatsu, 525–8577 Japan, 2012. Available online at https://www.jstage.jst.go.jp/article/ast/33/2/33_2_123/_pdf; visited on 2014-03-31.

[23] Masanori Morise. An attempt to develop a singing synthesizer by collaborative creation. In *Proceedings of the Stockholm Music Acoustics Conference 2013, SMAC 2013, Stockholm, Sweden*. Faculty of Engineering, University of Yamanashi, Japan, 2013. Available online at http://iwk.mdw.ac.at/lit_db_iwk/download.php?id=18114; visited on 2014-03-31.

[24] Music Technology Group. Spectral modelling synthesis tools for sound and music applications. Available online at https://github.com/MTG/sms-tools; visited on 2014-05-14.

[25] Music Technology Group. libsms library for spectral modeling synthesis. Website, 2009. Available online at http://mtg.upf.edu/static/libsms/; visited on 2014-03-31.

[26] Jordi Bonada Sanjaume. *Voice Processing and Synthesis by Performance Sampling and Spectral Models*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2008. Available online at http://mtg.upf.edu/static/media/PhD_jbonada.pdf; visited on 2014-03-31.

[27] Michael Tauch. Cepstrale techniken für die lautsprechermessung. Bachelorarbeit, Institut für Breitbandkommunikation der Technischen Universität Graz, 2009.

[28] Milan Zamazal. Singing computer. Website. Available online at http://devel.freebsoft.org/singing-computer; visited on 2014-06-27.