



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterthesis

Patrick Nolte

Latenzoptimierte Tonhöhenverschiebung
polyphoner Gitarrensingale

Patrick Nolte

Latenzoptimierte Tonhöhenverschiebung
polyphoner Gitarrensingale

Masterthesis eingereicht im Rahmen der Masterprüfung
im Studiengang Informations- und Kommunikationstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrich Sauvagerd
Zweitgutachter: Prof. Dr. Lutz Leutelt

Abgegeben am 13. April 2021

Patrick Nolte

Thema der Masterthesis

Latenzoptimierte Tonhöhenverschiebung polyphoner Gitarrensingale

Stichworte

Pitch Shifting, Frequency Scaling, polyphone Gitarrensingale, Latenz, OLA, SOLA, WSOLA, Roller, Phase Vocoder, Constant-Q Transformation

Kurzzusammenfassung

Diese Thesis beschäftigt sich mit der latenzoptimierten Tonhöhenverschiebung polyphoner Gitarrensingale für die Realisierung eines Echtzeitsystems. Dazu werden verschiedene klassische sowie neuartige Algorithmen im Zeit- und Frequenzbereich auf ihre Eignung hin untersucht. Es wird diskutiert, inwieweit sich die Optimierung der Latenz auf die Klangqualität solcher Algorithmen auswirkt.

Title of the paper

Latency Optimized Pitch Shifting of Polyphonic Guitar Signals

Keywords

Pitch Shifting, Frequency Scaling, polyphonic Guitar Signals, Latency, OLA, WSOLA, Roller, Phase Vocoder, Constant-Q Transform

Abstract

This thesis deals with the latency optimized pitch shifting of polyphonic guitar signals for the realization of a real-time system. For this purpose, various classical as well as novel algorithms in the time and frequency domain are examined concerning their suitability. It is discussed to what extent the optimization of the latency affects the sound quality of such algorithms.

Danksagung

An dieser Stelle möchte ich mein Wort an all diejenigen richten, die mich bei der Erstellung dieser Arbeit begleitet und unterstützt haben.

Bei meinem Erstprüfer Herrn Prof. Sauvagerd bedanke ich mich für die Betreuung meiner Arbeit, den wertvollen Anregungen sowie dafür, dass er jederzeit ein offenes Ohr für mich hatte. Bei Herrn Prof. Leutelt bedanke ich mich für die Zweitprüfung dieser Arbeit.

Ich bedanke mich bei Oliver für das Korrekturlesen und den musikalischen Ausgleich während der Erstellung dieser Arbeit.

Mein besonderer Dank geht an meine Familie und meine Freundin Laura, die mich während des Studiums sowie bei der Erstellung dieser Arbeit mit vielen Tipps und aufmunternden Worten unterstützt haben.

Hamburg, April 2021

Inhaltsverzeichnis

Tabellenverzeichnis	7
Abbildungsverzeichnis	8
Abkürzungsverzeichnis	10
Symbolverzeichnis	12
1 Einführung	14
1.1 Motivation	14
1.2 Ziele dieser Arbeit	16
2 Grundlagen	18
2.1 Analyse von Gitarrensignalen	18
2.2 Kurzzeit-Spektralanalyse	22
2.2.1 Diskrete Fourier-Transformation	22
2.2.2 Unschärfe-Prinzip und Zero Padding	25
2.2.3 Fensterung und Leck-Effekt	28
2.2.4 Short-Time Fourier Transformation und Overlap-Add	31
2.3 Constant-Q Transformation mithilfe nichtstationärer Gabor-Systeme	37
2.3.1 Gabor-Systeme	38
2.3.2 Von der festen zur variablen Zeit-Frequenzauflösung	39
2.3.3 Constant-Q Systeme	42
2.3.4 Effiziente Implementierung	43
2.3.5 Realisierung als Echtzeitsystem	44
3 Pitch Shifting	46
3.1 Definition und Zusammenhang mit Time Scaling	46
3.2 Herausforderungen und typische Artefakte	51
4 Algorithmen im Zeitbereich	61
4.1 Overlap-Add (OLA)	63
4.2 Synchronous Overlap-Add (SOLA)	65

4.3	Waveform Similarity Overlap-Add (WSOLA)	66
4.4	Roller Algorithmus	69
4.4.1	Frequenzverschiebung mittels Single Sideband (SSB) Modulation	70
4.4.2	Filterbank-Design	77
5	Algorithmen im Frequenzbereich	84
5.1	Phase Vocoder auf Basis der Short-Time Fourier Transformation . . .	85
5.1.1	Standard Phase Vocoder	85
5.1.2	Verbesserter Phase Vocoder	90
5.2	Phase Vocoder auf Basis der Constant-Q Transformation	93
5.2.1	Wahl eines Zeit-Frequenz-Gitters	93
5.2.2	Darstellung der Systemkoeffizienten	94
5.2.3	Pitch Shifting mithilfe der CQT	95
5.2.4	Pitch Shifting mithilfe der sliCQ	97
6	Ergebnisse und Auswertung	98
6.1	Testsignale	99
6.2	Parameterwahl	101
6.3	Klangqualität	102
6.3.1	Hörttest	102
6.3.2	Objektive Audiomerkmale	105
6.4	Latenz	109
7	Schlussbemerkung	116
7.1	Fazit	116
7.2	Ausblick	118
	Literaturverzeichnis	119
A	Vergleich der Algorithmen anhand objektiver Audiomerkmale	124
A.1	Zero-Crossing-Rate	125
A.2	Spectral Centroid	127
A.3	Spectral Entropy	129
A.4	Spectral Spread	131
B	Beigefügte CD	133

Tabellenverzeichnis

3.1	Musikalische Noten mit den entsprechenden Frequenzen	49
6.1	Übersicht der gewählten Parameter der untersuchten Algorithmen	101

Abbildungsverzeichnis

1.1	Beispielhafter Signalfluss für Komponenten mit digitaler Signalverarbeitung	15
2.1	Anschlag der tiefen E-Saite	19
2.2	Anschlag der hohen E-Saite im 22. Bund	20
2.3	Unschärfe-Prinzip anhand von zeitbegrenzten Gauß-Fenstern	25
2.4	Unterschiedliche Signaldauer einer Sinusschwingung mit 100 Hz	27
2.5	Entstehung des Leck-Effekts	29
2.6	Vergleich von verschiedenen Fensterfunktionen	30
2.7	Ablauf der STFT	32
2.8	Auswirkung von verschiedenen Fensterlängen bei der STFT	35
2.9	Zeit-Frequenz Ebene bei der STFT	36
2.10	Änderung der Zeit-Frequenz-Auflösung über die Zeit	40
2.11	Änderung der Zeit-Frequenz-Auflösung über die Frequenz	41
2.12	Zeit-Frequenz-Gitter eines Constant-Q Systems	43
3.1	Doppelte Abspielgeschwindigkeit einer Sinusschwingung	47
3.2	Pitch Shifting als Kombination aus Time Scaling und Resampling	48
3.3	Beispiel für Pitch Shifting um zwei Halbtöne nach oben	50
3.4	Erhaltung der Formanten durch Anpassung der spektralen Hüllkurve	54
3.5	Pitch Shifting einer weiblichen Stimme	57
3.6	Pitch Shifting einer akustischen Gitarre	58
3.7	Pitch Shifting einer E-Gitarre nach oben	59
3.8	Pitch Shifting einer E-Gitarre nach unten	60
4.1	Ablauf bei Time Scaling Algorithmen mit Overlap-Add	61
4.2	Overlap-Add (OLA)	63
4.3	Synchronous Overlap-Add (SOLA)	65
4.4	Waveform Similarity Overlap-Add (WSOLA)	67
4.5	Roller Algorithmus	69
4.6	Signalflussdiagramm der Double Sideband Modulation	70
4.7	Spektrale Wirkung einer Double Sideband Modulation	71
4.8	Signalflussdiagramm der Single Sideband Modulation	71
4.9	Reelles Signalflussdiagramm einer Single Sideband Modulation	72

4.10	Spektrale Wirkung einer Single Sideband Modulation	73
4.11	SSB Modulation mithilfe eines FIR Hilbert-Filters	74
4.12	SSB Modulation durch spektrale Manipulation	75
4.13	IIR Brückenfilter aus Biquad-Allpass-Sektionen	76
4.14	SSB Modulation mithilfe eines IIR Allpass-Brückenfilters	77
4.15	FIR Filterbank: Amplitudengänge aller Bandpassfilter	80
4.16	IIR Filterbank: Amplitudengänge aller Bandpassfilter	82
4.17	Realisierung als Multiraten-System	82
5.1	Phase Vocoder: Nutzung des Phasenspektrums	86
5.2	Time Scaling: Erhalt der horizontalen Phasenkohärenz	88
5.3	Pitch Shifting: Erhalt der horizontalen Phasenkohärenz	89
5.4	Geeignete Zeit-Frequenz-Gitter für Pitch Shifting	94
6.1	Vier Testsignale mit unterschiedlicher Komplexität	99
6.2	WSOLA Algorithmus: Betrachtung der maximalen Latenz	110
6.3	IIR Filterbank: Gruppenlaufzeiten der einzelnen Bandpässe	112
6.4	Gruppenlaufzeit des IIR Brückenfilters ($s = 0$)	113
6.5	Anregung des Systems mit einem Dirac-Impuls ($s = -1$)	113
6.6	PV-STFT Algorithmus: Betrachtung der Latenz	115
A.1	Zero-Crossing-Rate - Testsignal „Zweiklang“	125
A.2	Zero-Crossing-Rate - Testsignal „Dreiklang“	125
A.3	Zero-Crossing-Rate - Testsignal „Vierklang“	126
A.4	Zero-Crossing-Rate - Testsignal „Palm Mute“	126
A.5	Spectral Centroid - Testsignal „Zweiklang“	127
A.6	Spectral Centroid - Testsignal „Dreiklang“	127
A.7	Spectral Centroid - Testsignal „Vierklang“	128
A.8	Spectral Centroid - Testsignal „Palm Mute“	128
A.9	Spectral Entropy - Testsignal „Zweiklang“	129
A.10	Spectral Entropy - Testsignal „Dreiklang“	129
A.11	Spectral Entropy - Testsignal „Vierklang“	130
A.12	Spectral Entropy - Testsignal „Palm Mute“	130
A.13	Spectral Spread - Testsignal „Zweiklang“	131
A.14	Spectral Spread - Testsignal „Dreiklang“	131
A.15	Spectral Spread - Testsignal „Vierklang“	132
A.16	Spectral Spread - Testsignal „Palm Mute“	132

Abkürzungsverzeichnis

COLA	Constant Overlap-Add
CQT	Constant-Q Transformation
DAW	Digital Audio Workstation
DFT	Discrete Fourier-Transformation
DSB	Double Sideband
DTFT	Discrete-Time Fourier-Transformation
FFT	Fast Fourier-Transformation
FIR	Finite Impulse Response
FRFR	Flat Range Flat Response
FS	Frequency Shifter
IDFT	Inverse Discrete Fourier-Transformation
IFFT	Inverse Fast Fourier-Transformation
IIR	Infinite Impulse Response
ISTFT	Inverse Short-Time Fourier-Transformation
OLA	Overlap-Add
PV-STFT	Phase Vocoder auf Basis der STFT
sliCQ	Sliced Constant-Q Transformation
SOLA	Synchronous Overlap-Add
SSB	Single Sideband

STFT	Short-Time Fourier-Transformation
WOLA	Weighted Overlap-Add
WSOLA	Waveform Similarity Overlap-Add
ZCR	Zero-Crossing-Rate bzw. Nulldurchgangsrate

Symbolverzeichnis

α	Skalierungsfaktor beim Pitch Shifting / Time Scaling
Δf	Frequenzauflösung
Γ	Spektrale Hüllkurve
Ω	Normierte Kreisfrequenz
$\Phi(\mathbf{k})$	Phasenspektrum
τ	Latenz
$\tau_g(\mathbf{k})$	Gruppenlaufzeit
$\mathbf{A}(\mathbf{k})$	Amplitudenspektrum
$\mathbf{A}_{dB}(\mathbf{k})$	Amplitudenspektrum in dB
B	Bandbreite
b	b -tes Frequenzband einer Filterbank
$c_{m,k}$	Systemkoeffizienten eines Gabor-Systems
$F(\mathbf{k})$	Frequenz des k -ten DFT Koeffizienten
$F_m^{IF}(\mathbf{k})$	Momentanfrequenz des k -ten DFT Koeffizienten
F_s	Abtastfrequenz
g_k	Analyse-Filter der Constant-Q Transformation
$g_{m,k}$	Zeit-Frequenz Atome eines Gabor-Systems
$h(\mathbf{n})$	Synthese-Fenster

H_A	Analyse-Schrittweite
H_S	Synthese-Schrittweite
k	Frequenzstützstellen bzw. diskrete Frequenzvariable
L	Signallänge
m	Analyse-Zeitpunkte
N	Blocklänge oder Filterordnung
n	Zeitstützstellen bzw. diskrete Zeitvariable
Q	Gütefaktor
s	Anzahl von Halbtönen
T_s	Abtastzeit
$w(n)$	Fensterfunktion bzw. Analyse-Fenster
$X_m(k)$	Kurzzeit-Spektrum des m -ten Blocks von $x(n)$

1 Einführung

1.1 Motivation

Der Effekt der Tonhöhenverschiebung ist uns tatsächlich geläufiger als zunächst gedacht. Beispielsweise führt das langsamere Abspielen einer Schallplatte zu einer niedrigeren Tonhöhe. Damit einhergehend wird allerdings auch die Abspieldauer der Schallplatte verlängert. Die zeitliche Dauer und Tonhöhe sind somit aneinander gekoppelt. Ein bekanntes Beispiel sind die hohen Stimmen aus der Disney Produktion *Chip 'n' Dale*, die erst durch das deutlich schnellere Abspielen des Tonbandes ihren typischen Charakter erhalten.

Die Veränderung der Tonhöhe ohne Beeinflussung der zeitlichen Dauer wird Tonhöhenverschiebung oder auch Pitch Shifting genannt. Beide Begriffe werden in dieser Arbeit synonym verwendet. Dieser Effekt stellt eine wesentlich größere Herausforderung dar. Es existiert eine Vielzahl von Algorithmen, um Audiosignale unter Beibehaltung des originalen Charakters mit möglichst hoher Klangqualität in der Tonhöhe zu verschieben. Diese Algorithmen setzen nahezu durchgehend auf digitale Signalverarbeitung und verfolgen unterschiedliche Ansätze. Eine hohe Klangqualität ist jedoch mit einer aufwendigen Signalverarbeitung und hohen Latenz verbunden. Mit *élastique-Pro* des Herstellers *zplane* ist ein kommerzieller Algorithmus erhältlich, der heutzutage in den meisten *Digital Audio Workstations (DAW)* integriert ist und bei vielen Studioproduktionen zur nachträglichen Tonhöhenverschiebung sowie Klangformung dient. *élastiquePro* erreicht eine hohe Klangqualität, besitzt allerdings auch eine hohe Latenz von ca. 150 ms bei 48 kHz [Zpl20]. In dieser Anwendung spielt die Latenz jedoch keine kritische Rolle, da das Pitch Shifting erst nachträglich und nicht während des Spielens angewendet wird.

Eine beliebte Anwendung ist die Tonhöhenverschiebung von Gitarrensignalen in Echtzeit. Der Pitch Shifter wird zwischen E-Gitarre und Verstärker geschaltet und ermöglicht dem Nutzer während des Spielens die Tonhöhenverschiebung nach oben oder unten. Ein bekannter Pitch Shifter in Form eines Pedals ist das *Whammy*TM des amerikanischen Herstellers *DigiTech*[®], welches in den letzten Jahrzehnten in zahlreichen Musikstücken verwendet wurde. In digitalen Gitarrenverstärkern, wie dem *Kemper Profiler*TM,

ist ein Pitch Shifter bereits integriert. Da der Gitarrist direkt mit dem Effekt interagiert, ist eine niedrige Latenz ein entscheidender Faktor für ein gutes Spielgefühl. Eine zu hohe Latenz sorgt für ein irritierendes Gitarrenspiel und verhindert im schlimmsten Fall die Synchronisation zu den Mitmusikern. Ob die Latenz wahrgenommen wird, hängt neben der absoluten Zahl auch von der Struktur des Signals ab. Enthält das Signal starke Transienten¹, wird die Latenz früher als bei rein harmonischen Klängen wahrgenommen. Als Richtwert für die Latenz wird in der Literatur eine Grenze von ca. 10 ms genannt [Gör11, S. 251-252].

Die digitale Signalverarbeitung ist in der heutigen Veranstaltungstechnik stark verbreitet. Analoge Mischpulte und Gitarrenverstärker werden zunehmend durch digitale Pendanten verdrängt, die durch die digitale Signalverarbeitung Latenzen aufweisen. Im Signalfluss des Gitarristen können sich so mehrere digitale Komponenten befinden. Die einzelnen Latenzen sind zwar vernachlässigbar klein, allerdings summieren sich diese immer weiter auf. Ein möglicher Signalfluss mit praxisnahen Latenzen für Funkstrecke, Gitarrenverstärker und Mischpult ist in der folgenden Abbildung dargestellt:

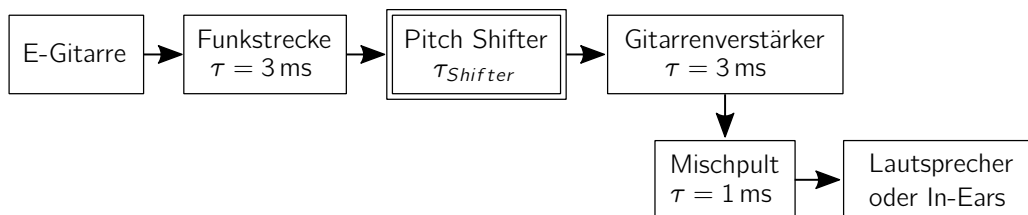


Abbildung 1.1: Beispielhafter Signalfluss für Komponenten mit digitaler Signalverarbeitung und den einzelnen Systemlatenzen τ

Hört der Gitarrist sein verstärktes Gitarrensignal über eine klassische Gitarrenbox oder einen *Flat Range Flat Response (FRFR)* Lautsprecher, kommen durch die Schallgeschwindigkeit von ca. 343 m/s zusätzlich ca. 2.9 ms pro Meter Abstand zum Lautsprecher hinzu. Bei der Verwendung von In-Ear Monitoring mit analoger Funktechnik entsteht dagegen keine nennenswerte Latenz. Der Pitch Shifter des Kemper ProfilersTM besitzt nach eigenen Messungen² eine dynamische Latenz von knapp 7 ms bis weit über 20 ms.

¹Transienten sind kurze, perkussive Signalabschnitte mit hohem Pegel am Anfang eines Schallereignisses. Bei einer Gitarre treten diese beim starken Anschlag einer Saite auf.

²Die Messungen wurden mithilfe eines Audio Interfaces und der DAW *Studio One*[®] 4 des Herstellers *PreSonus Audio Electronics, Inc.* durchgeführt. Als Messsignale wurden Gitarrensignale und Sinussignale verschiedener Frequenzen verwendet.

Diese Latenz verhindert zwar nicht die Synchronisation mit den anderen Musikern, allerdings leidet, insbesondere bei schnellem Gitarrenspiel, die direkte Interaktion mit dem Instrument.

1.2 Ziele dieser Arbeit

Aus der zuvor genannten Motivation heraus, soll in dieser Arbeit der Fokus in erster Linie auf eine latenzoptimierte Tonhöhenverschiebung gerichtet werden, um die Gesamtlatenz des in Abbildung 1.1 aufgezeigten Systems zu minimieren. Dazu werden verschiedene Algorithmen im Zeit- und Frequenzbereich auf ihre Eignung untersucht und jeweilige Vor- und Nachteile aufgezeigt. Verschiedene Algorithmen werden mithilfe der Software MATLAB[®] des Herstellers *The MathWorks Inc.* implementiert und untersucht.

Da das Leistungspotential eines Pitch Shifting Algorithmus von der Struktur des Eingangssignals abhängt, werden als Eingangssignale lediglich Gitarrensignale betrachtet. Des Weiteren ist der Pitch Shifter grundsätzlich das erste Glied nach der Gitarre (bzw. dem Funksystem) und befindet sich somit in der Signalkette vor anderen Gitarreneffekten und dem Gitarrenverstärker.

Der Pitch Shifter soll nicht zur Klangformung eingesetzt werden, sondern den Gitarrensound in möglichst hoher Qualität in der Tonhöhe verschieben. Die Erwartungen an die Klangqualität eines Pitch Shifters sind dementsprechend hoch. Beim Pitch Shifting können jedoch verschiedene Artefakte auftreten. Eine Übersicht über typische Artefakte wird in Abschnitt 3.2 gegeben. Die Reduzierung der Latenz geht zwangsläufig mit einer Verschlechterung der Klangqualität einher, sodass ein Trade-off vorliegt. Es wird untersucht, inwieweit Pitch Shifting Algorithmen in der Latenz optimiert werden können, bis die Klangqualität unzureichend ist.

Bei Gitarrensignalen kommt dem Pitch Shifting nach unten eine größere Bedeutung zu, da in der Praxis für das Pitch Shifting nach oben für gewöhnlich auf einen Kapodaster zurückgegriffen wird. Dieser verkürzt die schwingende Länge der Gitarrensaiten und erhöht die Tonhöhe auf natürlichem Wege. In Kapitel 6 werden die Algorithmen dennoch in Halbtonschritten bis ± 1 Oktave miteinander verglichen.

Gliederung

In Kapitel 2 werden wichtige Grundlagen für die Tonhöhenverschiebung und die Algorithmen im Frequenzbereich beschrieben. Da die Pitch Shifting Algorithmen nur anhand von Gitarrensignalen untersucht und angepasst werden, wird in Abschnitt 2.1 zunächst eine Analyse von Gitarrensignalen durchgeführt. Insbesondere werden die spektralen Grenzen einer E-Gitarre in Standard-Stimmung ermittelt. In Abschnitt 2.2 wird die Kurzzeit-Spektralanalyse anhand der STFT eingeführt, die die Grundlage für Pitch Shifting mithilfe des Phase Vocoders in Abschnitt 5.1 bildet. Mit der Constant-Q Transformation wird in Abschnitt 2.3 zudem eine weitere Zeit-Frequenz Darstellung eingeführt, die durch die Verwendung nichtstationärer Gabor-Systeme und der dadurch sichergestellten perfekten Rekonstruktion in den letzten Jahren deutlich an Bedeutung gewonnen hat. Die Transformation bietet insbesondere für musikalische Signale interessante Eigenschaften und bildet in Kombination mit dem Phase Vocoder in Abschnitt 5.2 einen weiteren Ansatz im Frequenzbereich.

Kapitel 3 beschäftigt sich mit der Definition des Pitch Shiftings und dem Zusammenhang zum Time Scaling, bei dem die Länge des Signals geändert wird, ohne die Tonhöhe zu beeinflussen. Darüber hinaus werden die beim Pitch Shifting typischerweise anzutreffenden Artefakte und Herausforderungen, wie die Verarbeitung stark polyphoner Signale oder den Erhalt der Formanten, diskutiert.

In dieser Arbeit werden sechs verschiedene Algorithmen untersucht, die die Tonhöhenverschiebung im Zeitbereich (Kapitel 4) oder Frequenzbereich (Kapitel 5) berechnen. In Kapitel 4 wird, neben drei klassischen (auf Overlap-Add basierten) Verfahren, der Roller Algorithmus vorgestellt, der die Zerlegung des Eingangssignals mithilfe einer Filterbank durchführt.

Ein weiterer und verbreiteter Ansatz ist die Berechnung der Tonhöhenverschiebung mithilfe des Phase Vocoders, der eine Transformation des Eingangssignals in den Frequenzbereich voraussetzt. Die Funktionsweise des Phase Vocoders wird in Kapitel 5 anhand der im Grundlagen-Kapitel eingeführten Zeit-Frequenz Darstellungen erläutert.

In Kapitel 4 und 5 wird überwiegend die Funktionsweise und Konstruktion der Algorithmen erläutert sowie jeweils eine kurze Einschätzung der Algorithmen in Bezug auf die Tonhöhenverschiebung von Gitarrensignalen gegeben. Eine detaillierte Auswertung und der Vergleich der Algorithmen ist abschließend in Kapitel 6 anhand geeigneter Testsignale zu finden. Neben der Beurteilung der Klangqualität wird vor allem die Frage geklärt, ob die eingangs angesetzte Latenz von 10 ms erreicht werden kann.

2 Grundlagen

In Abschnitt 2.1 wird eine Analyse von Gitarrensingen durchgeföhrt, um insbesondere die spektralen Grenzen und das Obertonverhalten zu bestimmen. Die gesammelten Erkenntnisse werden später in das Design und die Optimierung der Algorithmen mit einbezogen.

Abschnitt 2.2 befasst sich mit der diskreten Fourier-Analyse und der Short-Time Fourier-Transformation. Dieser Abschnitt bildet die Grundlage für das Pitch Shifting mithilfe des Phase Vocoders in Abschnitt 5.1.

Eine alternative Zeit-Frequenz Darstellung ist die Constant-Q Transformation, die in Abschnitt 2.3 vorgestellt wird. Diese Transformation eignet sich insbesondere für musikalische Signale und wird in Abschnitt 5.2 ebenfalls für das Pitch Shifting mithilfe des Phase Vocoders verwendet.

2.1 Analyse von Gitarrensingen

In diesem Abschnitt werden die Struktur und die Eigenschaften von Gitarrensingen untersucht, um die Wahl und Optimierung eines Algorithmus gezielt auf diese Art von Signalen anzupassen. Im Folgenden wird insbesondere die spektrale Bandbreite untersucht.

Da der Pitch Shifter das erste Glied nach der E-Gitarre ist, werden direkt die Ausgangssignale der Tonabnehmer (also ohne Einfluss eines Gitarrenverstärkers) untersucht. Die Signale werden mithilfe des Audiointerfaces *Focusrite Scarlett 6i6 2nd Gen* mit einer Abtastrate von $F_s = 48\text{ kHz}$ aufgenommen. Da eine E-Gitarre mit passiver Elektronik einen verhältnismäßig hohen Ausgangswiderstand besitzt, wird am Audiointerface der Eingang von „Line“ auf „Instrument“ umgeschaltet, um den Eingangswiderstand zu erhöhen.

Spektrale Bandbreite

Der menschliche Hörbereich umfasst den Frequenzbereich von ca. 16 Hz bis 20 kHz [Gör11, S. 28], wobei die höchste Frequenz mit steigendem Alter allmählich abnimmt. Ein naheliegender Ansatz ist es, diesen gesamten Hörbereich in der Frequenz zu skalieren. Aus Sicht der Signalverarbeitung ist es jedoch nicht immer sinnvoll den gesamten Frequenzbereich zu bearbeiten, insbesondere wenn das Eingangssignal nur eine kleinere Frequenzbandbreite belegt. Die höchste im Signal vorkommende Frequenz wirkt sich direkt auf die notwendige Abtastrate aus, wobei eine zu hoch gewählte Abtastfrequenz zu einem unnötig erhöhten Rechenaufwand führt. Dem Gegenüber steht die niedrigste im Signal vorkommende Frequenz, welche die minimale Frequenzauflösung vorgibt.

Um die spektrale Bandbreite einer E-Gitarre zu analysieren, wird jeweils einmal der niedrigste und höchste Ton angeschlagen. In der gesamten Arbeit wird eine E-Gitarre mit sechs Saiten in Standardstimmung (E, A, D, G, H, E / Kammerton $a^1 = 440$ Hz) und 22 Bünden betrachtet. Dadurch ist der niedrigste Ton die Note E2 ($f_{E2} = 82.41$ Hz) beim Anschlag der tiefen E-Saite und der höchste Ton die Note D6 ($f_{D6} = 1.174$ kHz) beim Anschlag im 22. Bund der hohen E-Saite. In Abbildung 2.1 ist zunächst das Spektrogramm beim Anschlag der tiefen E-Saite dargestellt. Die Saite wurde mit einem Plektrum angeschlagen. Dies führt zu einem Anschlag mit ausgeprägten Transienten.

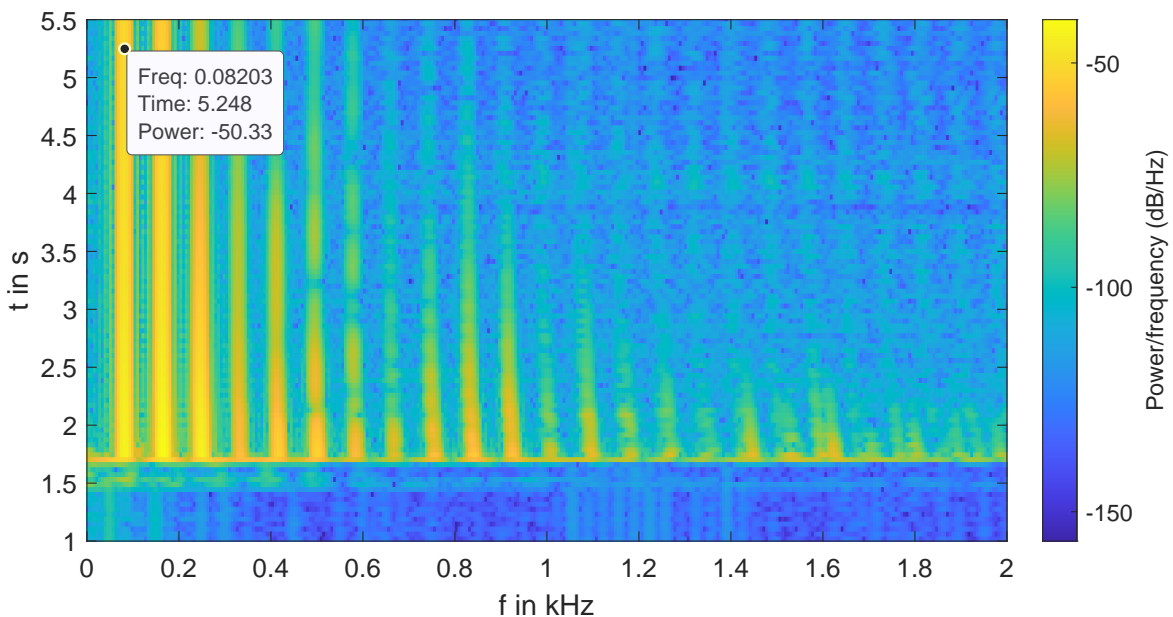


Abbildung 2.1: Spektrogramm: Anschlag der tiefen E-Saite mit $f_{E2} = 82.41$ Hz bei $F_s = 48$ kHz

Die tiefste Frequenz entspricht der zu erwartenden Note E2 bei ca. 82.41 Hz. Weitere Harmonische entstehen bei $k \cdot f_{E2}$, die sogar über 1 kHz schwach ausgeprägt sind. Eine Subharmonische bei beispielsweise 41.2 Hz entsteht dagegen nicht. Weiterhin fällt auf, dass in dem Moment des Anschlags bei ca. 1.75 s die Transienten in Form eines breit angeregten Spektrums erkennbar sind. Für diesen kurzen Moment ähnelt dies einem weißen Spektrum, wie es für impulsive Signale zu erwarten ist, wobei die Energie zu den hohen Frequenzen stetig abnimmt. Frequenzen unterhalb von f_{E2} werden daher ebenfalls angeregt. Im Allgemeinen lässt sich das Spektrogramm zeitlich in einen perkussiven und harmonischen Teil unterteilen.

Die Abbildung 2.2 zeigt das Spektrogramm beim Anschlag der hohen E-Saite im 22. Bund.

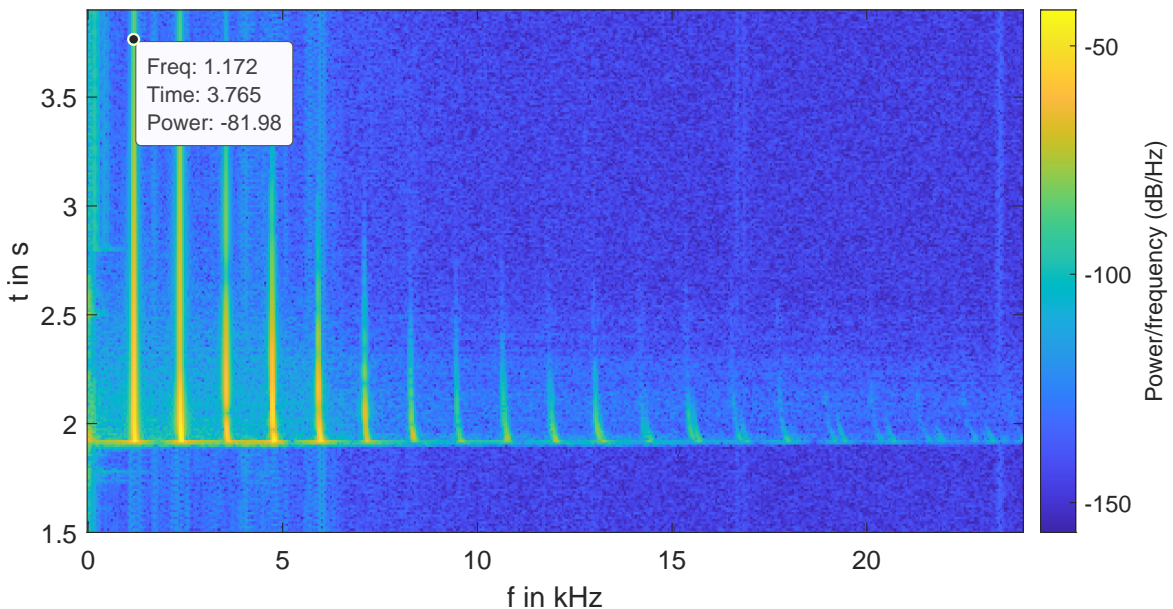


Abbildung 2.2: Spektrogramm: Anschlag der hohen E-Saite im 22. Bund mit $f_{D6} = 1.174$ kHz bei $F_s = 48$ kHz

Die zuvor beobachteten Erkenntnisse lassen sich analog übertragen. Ebenfalls entstehen Harmonische, die bis ca. 15 kHz nennenswerte Energie enthalten. In dem Moment des Anschlags wird das Spektrum breit angeregt.

Aus den Beobachtungen lassen sich zwei mögliche Vereinfachungen zur Reduzierung des Rechenaufwands ableiten. Da im harmonischen Fall keine Frequenzanteile unterhalb von f_{E2} vorhanden sind, ließe sich die Frequenzskalierung erst ab 82.41 Hz durchführen. Diese Vereinfachung trifft zwar für den perkussiven Teil nicht vollständig zu, stellt allerdings einen sinnvollen Kompromiss aus Klangqualität und Rechenaufwand dar. An

dieser Stelle sei angemerkt, dass diese Vereinfachung nicht bei jedem Algorithmus angewendet werden kann, da dies abhängig von der grundlegenden Struktur des Algorithmus ist.

Da in Gitarrensingen oberhalb von 15 kHz keine nennenswerte Energie vorhanden ist, wäre zudem eine Reduzierung der Abtastfrequenz auf $F_s = 30$ kHz möglich. Für eine weitere Senkung des Rechenaufwands auf Kosten der Klangqualität wäre darüber hinaus eine Abtastfrequenz von 20 kHz denkbar, sofern ein entsprechendes Anti-Aliasing Filter vorgeschaltet ist. Die Sinnhaftigkeit der niedrigeren Abtastrate ist letztlich abhängig von der Rechenintensität des verwendeten Algorithmus und der zur Verfügung stehenden Rechenleistung der Hardware.

2.2 Kurzzeit-Spektralanalyse

Dieser Abschnitt befasst sich mit der diskreten Fourier-Analyse und der Definition der Kurzzeit-Spektralanalyse mithilfe der Short-Time Fourier-Transformation. Die hier gesammelten Erkenntnisse bilden insbesondere die Grundlage für das Pitch Shifting mithilfe des Phase Vocoders in Abschnitt 5.1.

2.2.1 Diskrete Fourier-Transformation

Dieser Unterabschnitt basiert auf den Erkenntnissen von [Wer19; Zöl11; Rau16; KS10; KS17].

Gegeben sei ein diskretes Signal $x(n)$, welches durch eine ideale Abtastung eines kontinuierlichen Signals $x(t)$ mit einem periodischen Dirac-Impulskamm mit der Abtastzeit $T_s = 1$ beschrieben werden kann:

$$x(n) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad \text{mit } n \in \mathbb{Z} \quad (2.1)$$

Zu diesem unendlich langen diskreten Signal lässt sich die *Discrete-Time Fourier-Transformation (DTFT)* berechnen, die ein kontinuierliches und mit der Abtastfrequenz periodisches Spektrum ergibt:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j\omega n} \quad \text{mit } \omega \in \mathbb{R} \quad (2.2)$$

Da auf rechnergestützten Systemen die Berechnung des Frequenzspektrums eines unendlich langen Signals praktisch nicht möglich ist, wird das diskrete Signal auf eine Blocklänge N begrenzt:

$$x(n) = x(t) \cdot \sum_{n=0}^{N-1} \delta(t - nT_s) \quad \text{mit } n \in \mathbb{Z} \quad (2.3)$$

Die Begrenzung der Signallänge führt bei der Berechnung des Frequenzspektrums zu einer Abtastung des durch Gleichung 2.2 gegebenen Spektrums an den Stellen $\omega = \frac{2\pi k F_s}{N}$. Die DTFT (kontinuierliches Spektrum) wird dadurch in ein diskretes Spektrum

überführt. Die Berechnung des Frequenzspektrums eines endlichen, diskreten Signals wird durch die *Discrete Fourier-Transformation (DFT)* beschrieben:

$$X(k) = X(e^{j\omega})|_{\omega=\frac{2\pi k}{N}} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N} \quad \text{mit } k = 0, 1, \dots, N-1 \quad (2.4)$$

Die *Inverse Discrete Fourier-Transformation (IDFT)* ist wiederum durch

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cdot e^{j2\pi kn/N} \quad \text{mit } n = 0, 1, \dots, N-1 \quad (2.5)$$

gegeben. Das Spektrum $X(k)$ besteht somit aus N Frequenzstützstellen, die im Abstand der Frequenzauflösung äquidistant auf der Frequenzachse verteilt sind. Die periodische Wiederholung des Spektrums mit der Abtastfrequenz ist identisch mit der DTFT. Die Frequenzauflösung ist nur von der Abtastfrequenz und der Blocklänge abhängig:

$$\Delta f = \frac{F_s}{N} \quad (2.6)$$

Aus Gleichung 2.2 und Gleichung 2.4 geht zudem hervor, dass die DTFT und DFT an den Stellen $f = kF_s/N$ identisch sind. Eine Erhöhung der Blocklänge gegen unendlich überführt die DFT in eine DTFT.

Im Zusammenhang mit der DFT und der Frequenzauflösung ergeben sich zwei Schlussfolgerungen:

- Eine Erhöhung der Blocklänge N verbessert die Frequenzauflösung.
- Eine Erhöhung der Abtastfrequenz F_s bei gleicher Blocklänge N verschlechtert die Frequenzauflösung bei gleichzeitig höherem Rechenaufwand.

Im Folgenden werden alle Betrachtung zur diskreten Fourier-Analyse anhand der DFT bzw. der *Fast Fourier-Transformation (FFT)*, die lediglich eine effiziente Implementierung der DFT darstellt, durchgeführt.

Amplitudenspektrum, Phasenspektrum und Gruppenlaufzeit

Ausgehend von der Definition der DTFT wird zunächst die normierte Kreisfrequenz Ω definiert:

$$\Omega = \frac{\omega}{F_s} \quad (2.7)$$

Das, durch die DTFT berechnete, komplexe Frequenzspektrum setzt sich aus dem Amplituden- und Phasenspektrum zusammen:

$$X(e^{j\Omega}) = |X(e^{j\Omega})| \cdot e^{j\Phi(\Omega)} \quad (2.8)$$

Das Amplitudenspektrum ergibt sich aus der einfachen Betragsbildung des Frequenzspektrums:

$$A(\Omega) = \frac{1}{N} |X(e^{j\Omega})| = \frac{1}{N} \sqrt{[\Re \{X(e^{j\Omega})\}]^2 + [\Im \{X(e^{j\Omega})\}]^2} \quad (2.9)$$

Für eine detailliertere Darstellung bei gleichzeitig sehr großen und kleinen Amplitudenwerten bietet sich zudem eine logarithmische Darstellung in dB an:

$$A_{dB}(\Omega) = 20 \log_{10} A(\Omega) \text{ dB} \quad (2.10)$$

Das Phasenspektrum $\Phi(\Omega)$ berechnet sich durch:

$$\Phi(\Omega) = \arctan \frac{\Im \{X(e^{j\Omega})\}}{\Re \{X(e^{j\Omega})\}} \quad (2.11)$$

Die negative Ableitung des Phasenspektrums führt wiederum zur Gruppenlaufzeit $\tau_g(\Omega)$:

$$\tau_g(\Omega) = -\frac{d}{d\Omega} \Phi(\Omega) \quad (2.12)$$

Das diskrete Amplitudenspektrum $A(k)$ und Phasenspektrum $\Phi(k)$ sowie die Gruppenlaufzeit $\tau_g(k)$ ergeben sich (wie in Gleichung 2.4) durch eine Abtastung an den Stellen $\omega = \frac{2\pi k F_s}{N}$.

2.2.2 Unschärfe-Prinzip und Zero Padding

Dieser Unterabschnitt basiert auf den Erkenntnissen von [Kar17; KS17].

Das Unschärfe-Prinzip stellt eine wichtige Grenze dar und beschreibt im Zusammenhang mit der Fourier-Transformation, dass ein Signal nicht gleichzeitig beliebig genau in Zeit und Frequenz aufgelöst werden kann. Zeitauflösung und Frequenzauflösung verhalten sich grundsätzlich komplementär. In der Literatur ist dieses Naturgesetz auch unter der *Küpfmüllerschen Unbestimmtheitsrelation* anzutreffen und stellt eine auf die Nachrichtentechnik abgewandelte Interpretation der *Heisenbergschen Unschärferelation* dar. Die Zeitdauer Δt und Bandbreite Δf eines Signals stehen im folgenden Zusammenhang:

$$\Delta t \cdot \Delta f \geq K \quad (2.13)$$

In Abbildung 2.3 ist dieses Unschärfe-Prinzip anhand von zeitbegrenzten Gauß-Fenstern mit verschiedener Zeitdauer visualisiert. An dieser Stelle sei angemerkt, dass die Konstante K von der Definition der Bandbreite und Zeitdauer abhängt. Werden beispielsweise die Zeitdauer und Bandbreite so definiert, dass der jeweilige Funktionswert auf 50 % gesunken ist, ergibt das Produkt aus Δt und Δf ca. den Wert 1 [Kar17, S. 68].

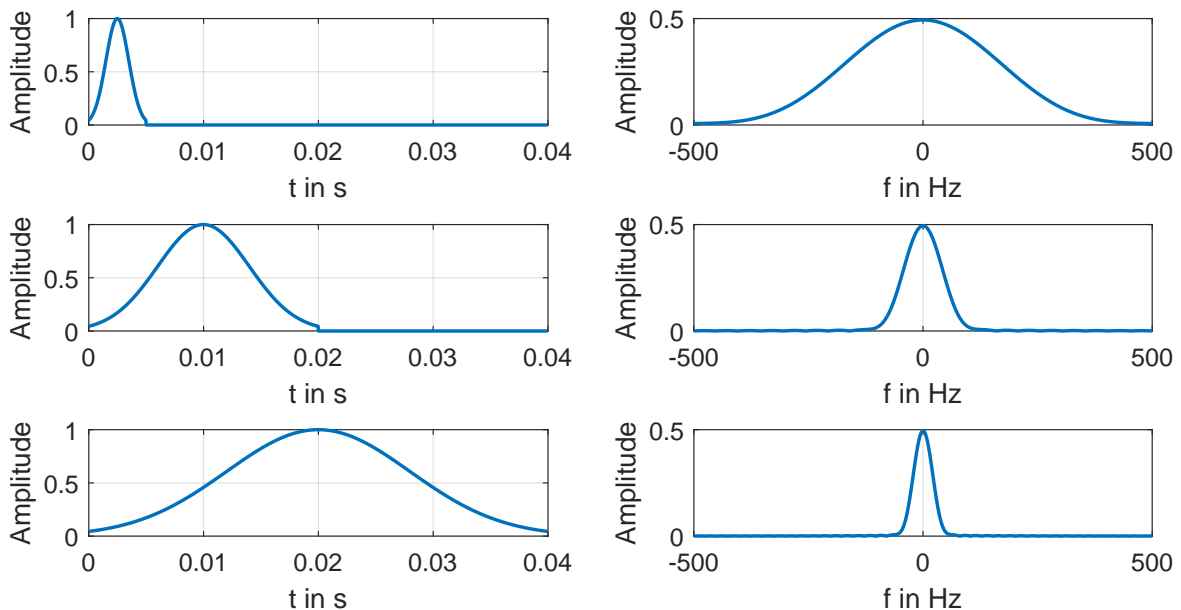


Abbildung 2.3: Unschärfe-Prinzip anhand von zeitbegrenzten Gauß-Fenstern verschiedener Breite (für eine bessere grafische Darstellung sind die jeweiligen Spektren mit Zero Padding interpoliert)

Dieser Zusammenhang ist in Abbildung 2.3 deutlich zu erkennen. Umso kleiner die Zeitdauer des Fensters, desto größer ist zwangsläufig die Bandbreite. Umgekehrt führt eine Erhöhung der Zeitdauer zu einer kleineren Bandbreite, also zu einer besseren spektralen Auflösung.

Aus dem Unschärfe-Prinzip geht also hervor, dass die Frequenzauflösung durch eine größere Zeitdauer (größere Blocklänge) verbessert werden kann. Eine häufig anzutreffende Methode ist das Anhängen von Nullen an das Signal (englisch *Zero Padding*), welche in Abbildung 2.4 anhand einer Sinusschwingung mit 100 Hz und variierender Signaldauer demonstriert ist. Alle Signale werden durch Zero Padding auf die gleiche Länge von 400 ms gebracht. Die DFT wird dadurch über eine größere Blocklänge berechnet. Es ist allerdings zu beachten, dass das Anhängen von Nullen dem Signal keine weiteren Informationen hinzufügt. Das Spektrum wird zwar nach Gleichung 2.6 durch mehr Frequenzstützstellen feiner aufgelöst (bzw. interpoliert), die Bandbreite Δf wird dadurch allerdings nicht beeinflusst. Zero Padding kann als Erhöhung der Messdauer interpretiert werden. Die Bandbreite Δf hängt aber lediglich von der Signaldauer Δt und nicht von der Messdauer ab.

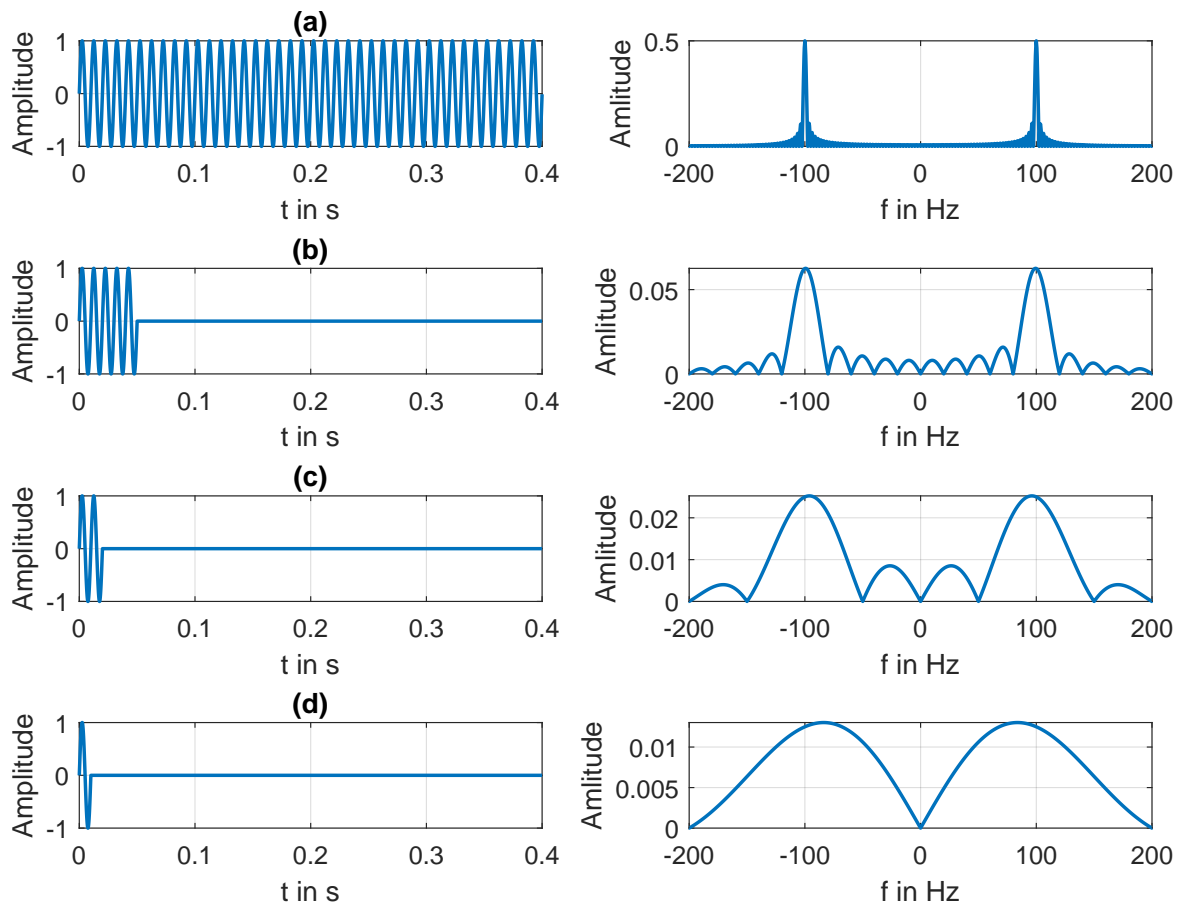


Abbildung 2.4: Unterschiedliche Signaldauer einer Sinusschwingung mit 100 Hz (für eine bessere grafische Darstellung sind die jeweiligen Spektren mit Zero Padding interpoliert)

In Abbildung 2.4 ist außerdem zu erkennen, dass die Spektren durch die Verkürzung der Signaldauer von (a) nach (d) zunehmend unsymmetrischer werden und sich der maximale Amplitudenwert Richtung 0 Hz bewegt. Die Ursache hierfür liegt in dem Symmetrie-Prinzip, welches beschreibt, dass ein reelles Eingangssignal immer zu einem achsensymmetrischen Amplitudenspektrum führt. In allen Fällen ragt allerdings der negative und positive Frequenzbereich in den jeweils anderen Bereich hinein, sodass die Verkürzung der Signaldauer zu immer stärkeren Überlagerung im Frequenzbereich führt [Kar17, S. 127 ff.]. In (d) führt dies fälschlicherweise zu einer Frequenz von ca. 90 Hz.

Eine Erhöhung der Signaldauer führt bei der DFT zu einer kleineren Bandbreite der einzelnen Frequenzstützstellen. Gleichzeitig nimmt jedoch die zeitliche Auflösung ab.

Bei nicht-stationären Signalen (bezogen auf die Blocklänge N) ist im Spektrum nicht zu erkennen zu welchem Zeitpunkt die einzelnen Frequenzanteile aufgetreten sind.

All diese, durch das Unschärfe-Prinzip gegebenen, Einschränkungen führen bei dem Ziel der Latenzoptimierung zu erheblichen Herausforderungen. Bei allen auf dem Frequenzbereich basierenden Ansätzen muss das Signal blockweise verarbeitet werden. Dies beeinflusst zwangsläufig die Systemlatenz. Eine möglichst kleine Blockgröße ist wünschenswert, um eine niedrige Latenz zu erreichen. Gleichzeitig wird jedoch die Bandbreite der Frequenzstützstellen erhöht und so eine genaue Frequenzbestimmung erschwert. Die hier gesammelten Erkenntnisse werden in Abschnitt 2.3 und Kapitel 5 fortwährend eine Rolle spielen und an entsprechender Stelle nochmals aufgegriffen.

2.2.3 Fensterung und Leck-Effekt

Dieser Unterabschnitt basiert auf den Erkenntnissen von [Kar17; Wer19; HRS02].

Die DFT stellt eine blockorientierte Operation dar, die das Frequenzspektrum zu einem endlichen Zeitsignal der Länge N berechnet. Eine Sinusschwingung ist allerdings ein periodisches und unendlich langes Signal. Bei einer Fourier-Analyse muss dieses Signal zwangsläufig zeitlich auf die gewünschte Länge N begrenzt werden. Dies wird als Fensterung (englisch *Windowing*) bezeichnet. Das Ausschneiden des gewünschten Signalabschnitts kann mit unterschiedlichen Fensterfunktionen durchgeführt werden, wobei das einfache Ausschneiden des Signalabschnitts einem Rechteck-Fenster entspricht. Die einzelnen Abtastwerte (englisch *Samples*) werden somit nicht gewichtet, sodass jeder einzelne Amplitudenwert unverändert bleibt.

Dieser Vorgang ist in Abbildung 2.5 in (a) dargestellt. Das Rechteck-Fenster mit einer Breite von 200 ms wurde so gewählt, dass durch die Fensterung von der Sinusschwingung ein Vielfaches der Periodenlänge erfasst wird (hier exakt 20 Perioden). Die DFT berechnet nun das Spektrum dieser gedanklich periodisch fortgesetzten Zeitfenster-Sequenz. Durch die Erfassung ganzer Perioden wird das Signal stetig fortgesetzt. Das Amplitudenspektrum zeigt dadurch exakt zwei Peaks bei -100 Hz und 100 Hz. Alle anderen Frequenzstützstellen sind gleich Null.

In (b) wird nun ausgehend von dem gleichen Signal lediglich die Länge des Rechteck-Fensters auf ca. 194.8 ms verkürzt, wodurch eine gebrochene Anzahl Perioden erfasst wird. Die periodische Fortsetzung dieser Zeitfenster-Sequenz führt nun zu einer raschen Steigungsänderung, die im ursprünglichen Signal nicht enthalten ist und mit weiteren Frequenzanteilen einhergeht. Das im entsprechenden Amplitudenspektrum ersichtliche „Auslaufen“ wird Leck-Effekt (englisch *Leakage*) genannt und ist unerwünscht.

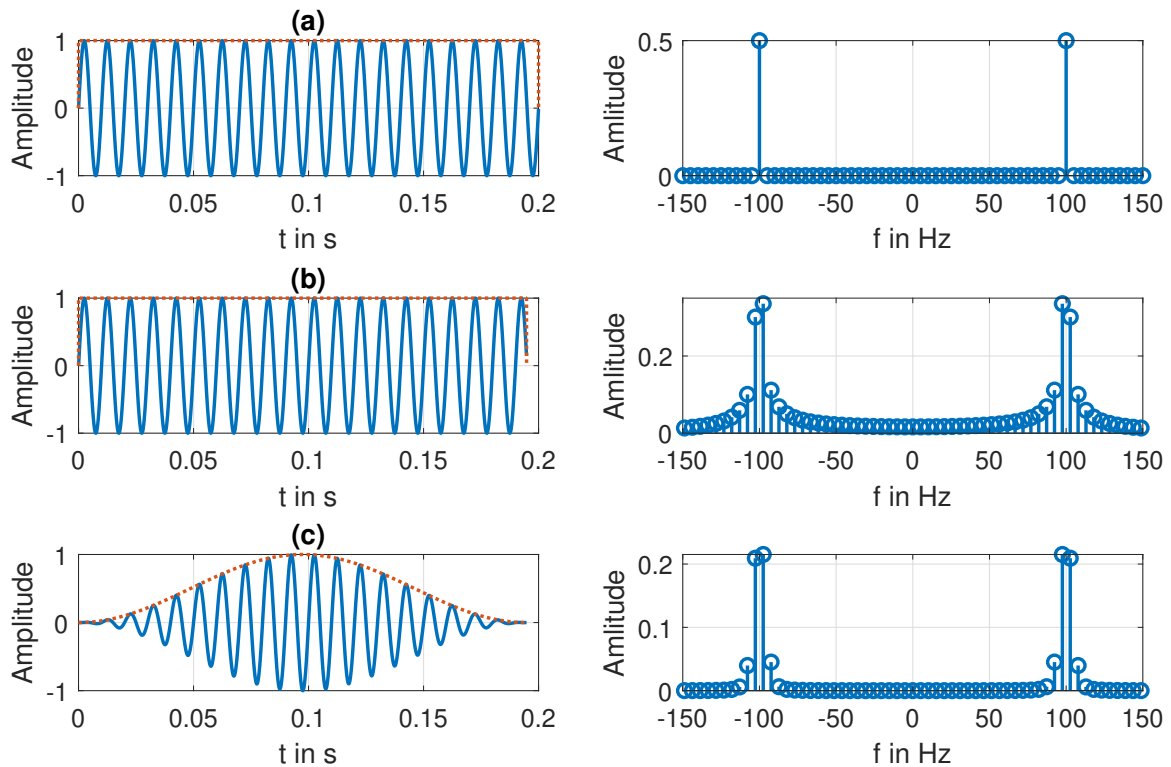


Abbildung 2.5: Entstehung des Leck-Effekts und Auswirkung einer gewichtenden Fensterfunktion (von-Hann-Fenster)

Der Leck-Effekt kann gänzlich vermieden werden, wenn durch das Fenster von einer periodischen Schwingung exakt ein Vielfaches der Periodenlänge erfasst wird. In der Praxis ist das allerdings nicht umzusetzen, da dies Vorkenntnisse über die spektrale Zusammensetzung des vorliegenden Signals erfordert, die letztlich erst durch die Fourier-Analyse bekannt wird. Anfang und Ende der Zeitfenster-Sequenz können von einer fehlerhaften periodischen Fortsetzung betroffen sein, daher werden diese Bereiche durch speziell konstruierte Fensterfunktionen $w(n)$ gewichtet:

$$x_w(n) = x(n) \cdot w(n) \quad (2.14)$$

Dieser Vorgang ist in Abbildung 2.5 in (c) anhand eines Von-Hann-Fensters (bzw. Hanning-Fensters) dargestellt. Durch die Gewichtung werden die Samples am Anfang und Ende langsam gegen Null geführt. Eventuell fehlerhafte periodische Fortsetzungen fallen so deutlich weniger ins Gewicht. Im Amplitudenspektrum zeigt sich dies anhand eines geringeren Leck-Effekts.

In der Literatur ist eine Vielzahl von unterschiedlichen Fensterfunktionen zu finden. Ein

Vergleich im Zeit- und Frequenzbereich von einer kleinen Auswahl an typischerweise verwendeten Fensterfunktionen ist in der folgenden Abbildung 2.6 dargestellt.

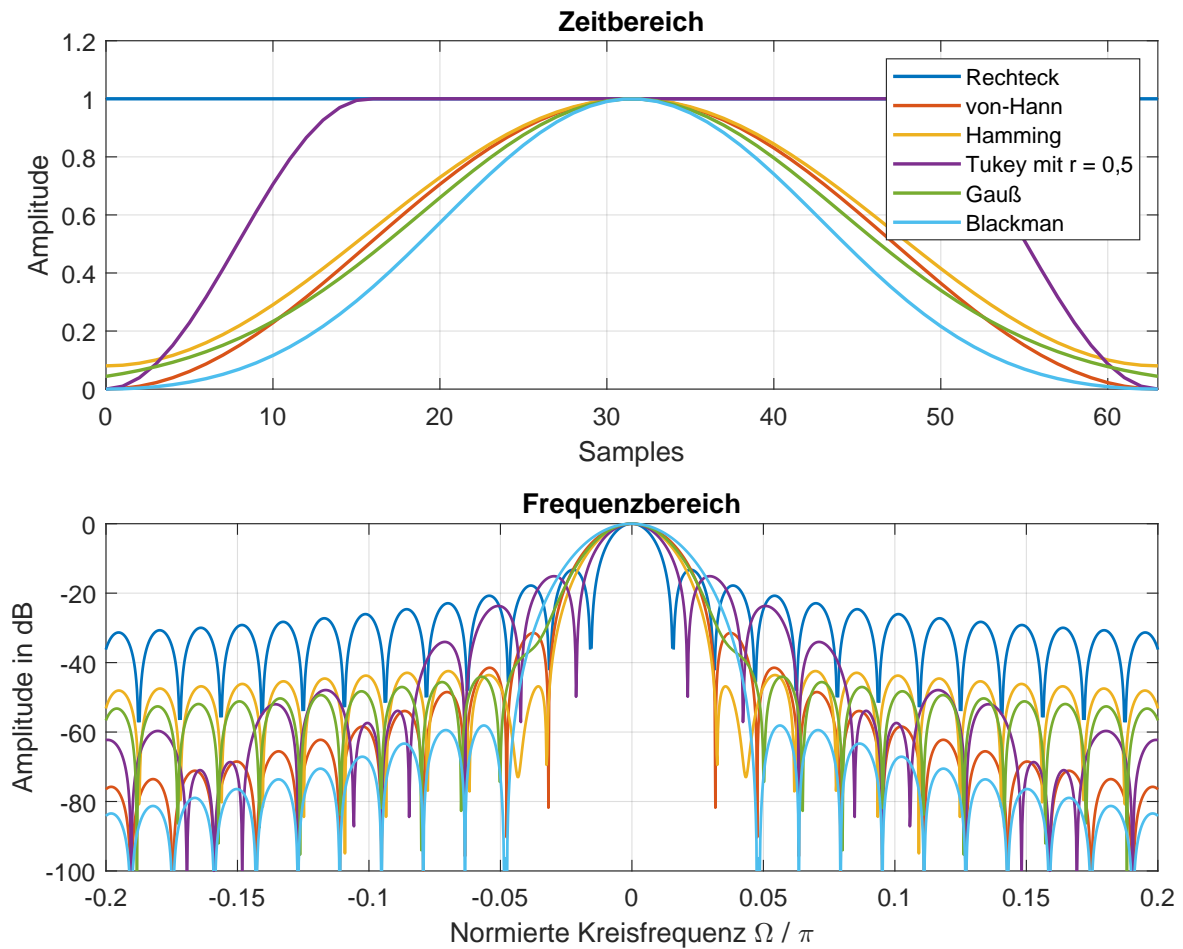


Abbildung 2.6: Vergleich von verschiedenen Fensterfunktionen (für eine bessere grafische Darstellung sind die jeweiligen Spektren mit Zero Padding interpoliert)

Alle speziellen Fensterfunktionen haben gemeinsam, dass sie das Signal im Zeitbereich allmählich gegen Null führen. Im Frequenzbereich wird zur Einteilung der Fensterfunktionen hauptsächlich die Breite der sogenannten Hauptkeule (englisch *Main Lobe*) und die Nebenkeulen-Dämpfung (englisch *Side Lobe Attenuation*) betrachtet, wobei nicht beide Parameter gleichzeitig optimiert werden können. Das Rechteck-Fenster hat beispielsweise die schmalste Hauptkeule, aber auch gleichzeitig die niedrigste Nebenkeulen-Dämpfung. Das Blackman-Fenster zeichnet sich dagegen durch die hier stärkste Nebenkeulen-Dämpfung aus, wobei als Kompromiss die breiteste Hauptkeule in Kauf genommen werden muss. Eine Besonderheit zeigt das Hamming-Fenster, welches die erste Nebenkeule besonders stark dämpft. Eine Verlängerung des Fensters

führt nach dem Unschärfe-Prinzip zu einer schmaleren Hauptkeule bzw. einer kleineren Bandbreite.

Ein detaillierter Vergleich von verschiedensten Fensterfunktionen ist in [HRS02] zu finden.

2.2.4 Short-Time Fourier Transformation und Overlap-Add

Dieser Unterabschnitt basiert auf den Erkenntnissen von [Kar17; KS17; Wer19; HRS02; Set07; Smi11].

Periodische Signale besitzen ein Linienspektrum, nichtperiodische Signale (wie z. B. Rauschen) hingegen ein breites und kontinuierliches Spektrum. Gitarrensingale können als fastperiodische Signale angesehen werden, die sich über einen gewissen Zeitraum wiederholen. Bei fastperiodischen Signalen ist es allerdings unsinnig eine spektrale Analyse über das gesamte Signal durchzuführen, da sich die spektrale Verteilung über die Zeit gesehen ändert (ausklingende und wechselnde Akkorde etc.).

Die diskrete *Short-Time Fourier-Transformation (STFT)* leitet sich aus der DFT (bzw. deren effizienten Implementierung FFT) ab und bildet ein wichtiges Werkzeug für die Zeit-Frequenz-Analyse. Die STFT ermöglicht es, ein Signal abschnittsweise zu analysieren, indem (im Falle einer Offline-Anwendung) einzelne Blöcke aus dem Signal ausgeschnitten und jeweils einer spektralen Analyse unterzogen werden. In dem in dieser Arbeit verfolgten Echtzeit-System werden die Blöcke hingegen sukzessiv aus den eingehenden Samples des Analog-Digital-Wandlers gebildet und entsprechend verarbeitet.

In Abbildung 2.7 ist der gesamte Ablauf, bestehend aus der Transformation in den Frequenzbereich und Rücktransformation in den Zeitbereich, dargestellt. Das zu verarbeitende Signal wird mithilfe einer Fensterfunktion der Länge N in sich überlappende Blöcke zerlegt. Die Vorgehensweise verringert den Leck-Effekt und verhindert gleichzeitig einen Informationsverlust. Der optimale Überlappungsgrad ist allerdings für jede Fensterfunktion unterschiedlich. Ziel ist es, dass die Aufsummierung aller Fenster entlang der Zeitachse möglichst eine Konstante ergibt und das Signal nicht in der Amplitude verändert wird. Für das hier dargestellte Von-Hann-Fenster ergibt sich beispielsweise ein optimaler Überlappungsgrad von 50 % [HRS02, S. 31]. Jeder gefensterte Block kann nun nacheinander mittels FFT in den Frequenzbereich überführt werden. Der gesamte Ablauf bis hin zum Frequenzbereich wird auch Analyse genannt.

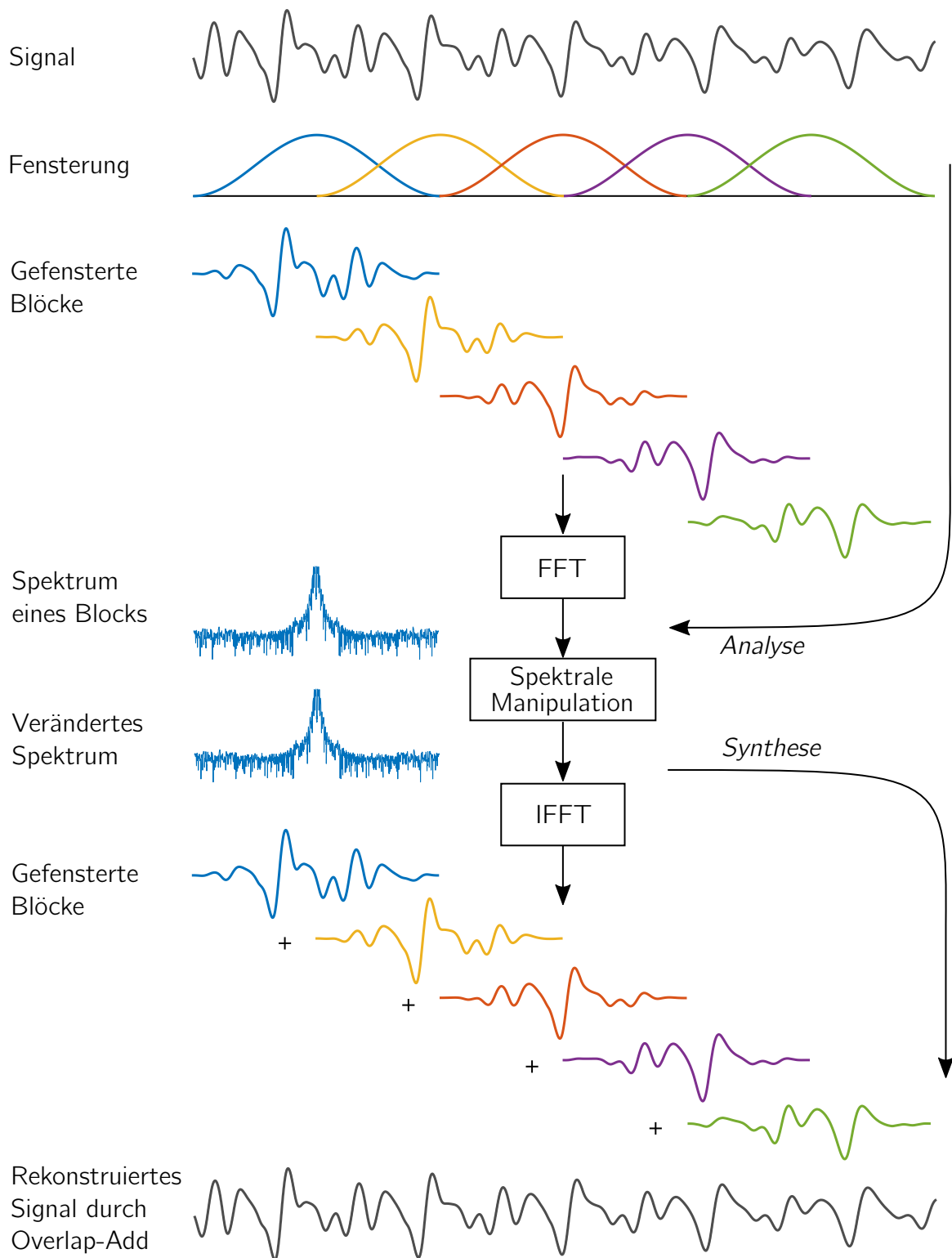


Abbildung 2.7: Ablauf der STFT (nach [Set07, S. 117])

Der Block „Spektrale Manipulation“ ist in Kapitel 5 von zentraler Bedeutung, da dort mittels Phase Vocoder die einzelnen Spektren in der Tonhöhe verschoben werden. An dieser Stelle wird aber erst einmal von keiner spektralen Manipulation ausgegangen, so dass die *Inverse Fast Fourier-Transformation (IFFT)* der einzelnen Spektren zu den exakt gleichen gefensterten Blöcken führt. Die anschließende Aufsummierung aller Blöcke entlang der Zeitachse wird *Overlap-Add (OLA)* genannt und ergibt durch die geschickte Kombination von Fensterfunktion und Überlappungsgrad exakt das ursprüngliche Signal. Der Prozess der Rücktransformation in den Zeitbereich bis zur Zusammensetzung der einzelnen Blöcke wird als Synthese bezeichnet.

Die STFT, also der Analyse-Teil der Abbildung 2.7, setzt sich somit folgendermaßen zusammen:

$$X_m(k) = \sum_{n=0}^{N-1} x(n + mH_A)w(n) \cdot e^{-j2\pi kn/N} \quad \text{mit } k = 0, 1, \dots, N-1 \quad (2.15)$$

$X_m(k)$ beschreibt dabei das komplexe Frequenzspektrum des m -ten Blocks um den Zeitpunkt mH_A , wobei $m \in \mathbb{N}_0$ ist. H_A steht für die Analyse-Schrittweite und beschreibt den Überlappungsgrad der aufeinanderfolgenden Fenster in Samples. Die für ein Von-Hann-Fenster optimale Analyse-Schrittweite beträgt beispielsweise $H_A = N/2$.

Die entsprechende *Inverse Short-Time Fourier-Transformation (ISTFT)* führt wiederum zu den einzelnen Blöcken m der Länge N .

$$x_m(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_m(k) \cdot e^{j2\pi kn/N} \quad \text{mit } n = 0, 1, \dots, N-1 \quad (2.16)$$

Das OLA Verfahren setzt das rekonstruierte Signal anschließend wieder zusammen:

$$x(n) = \sum_{m=0}^M x_m(n) \quad (2.17)$$

Weighted Overlap-Add (WOLA) und Constant Overlap-Add (COLA)

In dem zuvor betrachteten Fall wird das berechnete Spektrum nicht modifiziert, sodass die Rücktransformation wieder zu den gefensterten Blöcken $x_m(n)$ führt. Diese einfache Aufsummierung nach dem OLA Verfahren führt (bei korrekter Wahl der Fensterfunktion und des Überlappungsgrads) zu einer perfekten Rekonstruktion des Signals. Bei einer nichtlinearen Modifikation des Spektrums, wozu Pitch Shifting und Time Scaling zählen, bietet sich ein weiteres Synthese-Fenster nach der Rücktransformation an, um

Artefakte an den Randbereichen der einzelnen Blöcke zu kontrollieren. Diese zusätzliche Gewichtung der rücktransformierten Blöcke wird *Weighted Overlap-Add (WOLA)* genannt und führt zu einer besseren Synthese des Ausgangssignals [Smi11]. Das OLA Verfahren (siehe Gleichung 2.17) und WOLA Verfahren unterscheiden sich somit nur durch ein zusätzliches Synthese-Fenster $h(n)$:

$$x(n) = \sum_{m=0}^M x_m(n)h(n) \quad (2.18)$$

Die *Constant Overlap-Add (COLA)* Bedingung beschreibt, dass sich die zeitlich aufeinanderfolgenden Fenster konstant aufsummieren, sodass eine perfekte Rekonstruktion gewährleistet ist. Sie lässt sich sowohl für das OLA als auch das WOLA Verfahren formulieren. Beim OLA Verfahren gilt für das Analyse-Fenster $w(n)$:

$$\sum_m w(n - mH_A) = \text{konstant} \quad (2.19)$$

Für WOLA gilt wiederum für die Kombination aus Analyse- und Synthese-Fenster der folgende Zusammenhang. Die Synthese-Schrittweite H_S entspricht der Analyse-Schrittweite H_A .

$$\sum_m w(n - mH_A)h(n - mH_S) = \text{konstant} \quad (2.20)$$

Bei WOLA wird häufig sowohl in der Analyse als auch in der Synthese dasselbe Fenster verwendet ($h(n) = w(n)$), sodass die folgende Bedingung erfüllt sein muss:

$$\sum_m w^2(n - mH_A) = \text{konstant} \quad (2.21)$$

Eine gängige Methode zur Konstruktion eines Fensters für WOLA besteht darin die Quadratwurzel aus bereits bekannten OLA Fenstern zu ziehen und den Überlappungsgrad zu übernehmen. Alternativ kann auch die Fensterfunktion beibehalten und der Überlappungsgrad angepasst werden. Im Falle eines Von-Hann-Fensters kann Gleichung 2.21 beispielsweise durch einen Überlappungsgrad von 75 % oder der Quadratwurzel des Fensters und einem Überlappungsgrad von 50 % erfüllt werden.

Die STFT wird in Abschnitt 5.1 immer mit dem WOLA Verfahren verwendet.

Spektrogramm und Unschärfe-Prinzip

Eine beliebte Darstellungsform der STFT ist das Spektrogramm, wie es bereits in Abschnitt 2.1 zur Darstellung der Gitarrensingale verwendet wurde. Ein Spektrogramm stellt die Amplitudenspektren von $X_m(k)$ als zwei-dimensionale Abbildung dar, wobei die x- und y-Achse als Frequenz- und Zeitachse verwendet und die Amplitudenwerte über eine Farbcodierung mit meist logarithmischer Skala dargestellt werden. In der folgenden Abbildung 2.8 sind vier Spektrogramme mit unterschiedlicher Fensterlänge bzw. Blocklänge N dargestellt, wobei sich das analysierte Signal nicht unterscheidet.

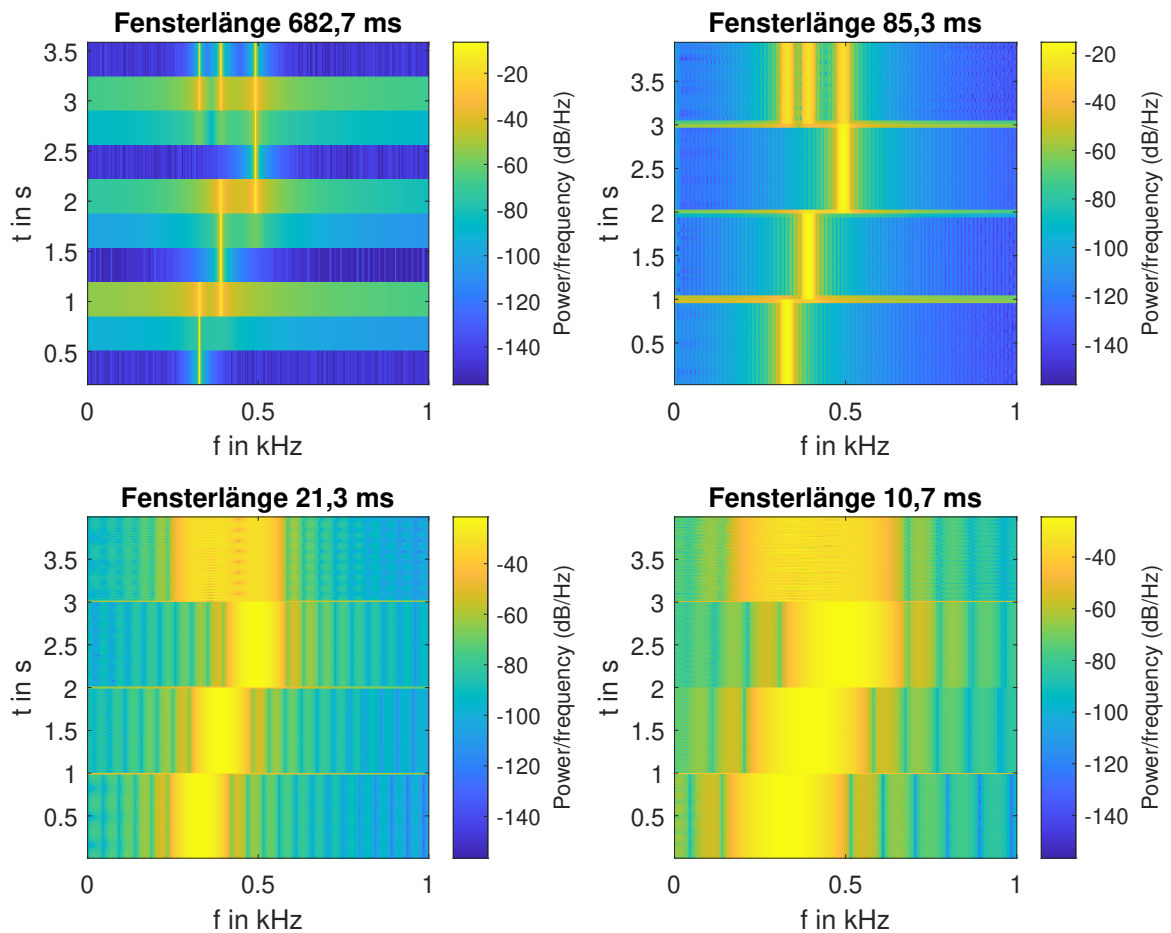


Abbildung 2.8: Auswirkung von verschiedenen Fensterlängen auf die Zeit-Frequenz Ebene bei der STFT

Das analysierte Signal besteht aus einzelnen Sinusschwingungen mit den Frequenzen $f_{E4} = 329.64$ Hz, $f_{G4} = 392$ Hz sowie $f_{H4} = 493.88$ Hz, die in den ersten drei Sekunden nacheinander erklingen. Erst ab der dritten Sekunde erklingen alle Sinusschwingungen

gleichzeitig als Dreiklang (E-Moll Akkord). Anhand der unterschiedlichen Fensterlängen wird an dieser Stelle das Unschärfe-Prinzip nochmal direkt sichtbar. Bei einer Fensterlänge von 682.7 ms (entspricht einer Blocklänge von $N = 32768$) ist die Frequenzauflösung sehr hoch, sodass die einzelnen Sinusschwingungen ohne Probleme voneinander zu unterscheiden sind. Allerdings leidet nach dem Unschärfe-Prinzip (siehe Gleichung 2.13) die Zeitauflösung, sodass der Anschein erweckt wird, dass die Sinusschwingungen um die Zeitpunkte 1 s und 2 s gleichzeitig erklingen. Bei einer Fensterlänge von 10.7 ms (entspricht einer Blocklänge von $N = 512$) sind beim Dreiklang die einzelnen Frequenzen aufgrund der geringen Frequenzauflösung nicht voneinander unterscheidbar und verschmieren ineinander. Allerdings sind nun die zeitlichen Umschaltpunkte gut lokalisierbar. Ein geeigneter Kompromiss aus Zeit- und Frequenzauflösung ist an dieser Stelle das Spektrogramm mit einer Fensterlänge von 85.3 ms ($N = 4096$).

Die STFT spannt eine Zeit-Frequenz Ebene auf, die von der gewählten Blocklänge abhängig ist. Die nachfolgende Abbildung 2.9 visualisiert diesen Zusammenhang anhand von zwei verschiedenen Blocklängen.

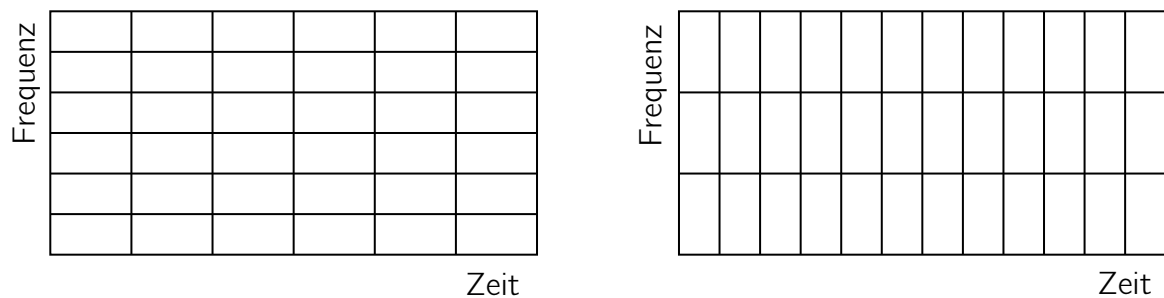


Abbildung 2.9: Zeit-Frequenz Ebene bei der STFT

Da die STFT auf der DFT basiert, ist das Zeit-Frequenz Raster immer äquidistant verteilt. Größere Blocklängen im Zeitbereich führen zu kleineren Abständen zwischen den Frequenzstützstellen (links). Eine Verkleinerung der Blocklänge verringert hingegen die Frequenzstützstellen (rechts). Der Flächeninhalt der einzelnen Kästchen bleibt hingegen konstant.

Diese äquidistante bzw. feste Zeit-/Frequenzauflösung kann jedoch durch den Einsatz von sogenannten nichtstationären Gabor Systemen adaptiv verändert werden, welche im folgenden Abschnitt 2.3 hergeleitet wird. Dabei ist zu beachten, dass das Unschärfe-Prinzip nicht übergangen werden kann.

2.3 Constant-Q Transformation mithilfe nichtstationärer Gabor-Systeme

Bei der DFT bzw. STFT sind die Frequenzstützstellen äquidistant auf der Frequenzachse verteilt. Die musikalischen Noten sind allerdings logarithmisch verteilt (siehe Tabelle 3.1), sodass im tiefen Frequenzbereich zu wenige und im hohen Frequenzbereich zu viele Frequenzstützstellen vorhanden sind. Wünschenswert wäre folglich eine logarithmische Verteilung der Frequenzstützstellen.

Eine optimale Abbildung der logarithmischen Verteilung ist durch eine *Constant-Q Transformation (CQT)* gegeben, die 1991 von *Brown* in [Bro91] vorgestellt wurde. Der Name leitet sich von einer konstanten Güte Q (also dem konstanten Verhältnis zwischen Mittenfrequenz und Bandbreite) der einzelnen Bandpassfilter einer Filterbank her. Durch diesen Aufbau wird einerseits die Frequenzauflösung im tiefen Frequenzbereich verbessert, andererseits steigt die Zeitauflösung zu den hohen Frequenzen. Letzteres führt zu einer besseren Auflösung von Transienten. Problematisch an der Realisierung der CQT von *Brown* ist allerdings, dass die Transformation nicht invertierbar ist und die Vorteile nur bei einer spektralen Analyse zu tragen kommen.

Eine CQT mit perfekter Rekonstruktion kann mithilfe von nichtstationären Gabor-Systemen aufgebaut werden, die auf der Frametheorie basieren und die Idee von klassischen Gabor-Systemen weiterführen. Im vergangenen Jahrzehnt wurden dazu zahlreiche Veröffentlichungen publiziert, auf denen ebenfalls die Erläuterungen der folgenden Unterabschnitte basieren:

- Theorie klassischer Gabor-Systeme (*Holighaus* [Hol10])
- Theorie und Implementierung nichtstationärer Gabor-Frames (*Balasz et al.* [Bal+11])
- Konstruktion einer CQT (*Dörfler et al.* [Dör+11])
- Theorie und Implementierung nichtstationärer Gabor-Frames (*Holighaus* [Hol13])
- Konstruktion einer echtzeitfähigen CQT (*Holighaus et al.* [Hol+13])

Zunächst werden in Unterabschnitt 2.3.1 klassische Gabor-Systeme in Bezug auf die Frametheorie erläutert, die (wie die STFT) nur eine feste Zeit-Frequenzauflösung ermöglichen. Der Übergang zu den nichtstationären Gabor-Systemen ist anschließend in Unterabschnitt 2.3.2 dargestellt. Mit solchen Systemen ist eine variable Zeit-/Frequenzauflösung möglich. Mithilfe von frequenzzeitig nichtstationären Gabor-Systemen, die auch als unregelmäßige Filterbank interpretiert werden können, kann die Bandbreite der Filter und die frequenzzeitige Abtastdichte prinzipiell beliebig variiert

werden. In Unterabschnitt 2.3.3 werden die hier betrachteten Constant-Q Systeme als ein mögliches frequenzzeitiges System vorgestellt. Eine effiziente Implementierung ist zudem mithilfe der FFT bzw. IFFT möglich, welche in Unterabschnitt 2.3.4 diskutiert wird. Den Abschluss bildet das Kapitel mit der Realisierung nichtstationärer Systeme als Echtzeitsystem (Unterabschnitt 2.3.5).

An dieser Stelle sei erwähnt, dass die folgenden Unterabschnitte als kurze Einleitung in die grundlegende Idee und die Begrifflichkeiten nichtstationärer Gabor-Systeme zu verstehen sind. Es wird vordergründig die Analyse mithilfe nichtstationärer Gabor-Systemen beschrieben. Eine detaillierte Erläuterung der dahinterliegenden Theorie würde den Rahmen dieser Arbeit übersteigen, daher wird für eine umfassende Beschreibung und Synthese mithilfe entsprechender Dual-Frames auf die oben genannte Literatur verwiesen.

2.3.1 Gabor-Systeme

Dennis Gábor stellte 1946 ein Verfahren vor, dass die Zerlegung eines Signals in elementare Signale beschreibt:

$$x = \sum_{m,k} c_{m,k} g_{m,k} \quad (2.22)$$

In der Frametheorie wird die Signalfolge $g_{m,k}$ als Frame bezeichnet, wenn mithilfe der Systemkoeffizienten $c_{m,k}$ eine perfekte Rekonstruktion des Signals x möglich ist. $g_{m,k}$ werden auch als Zeit-Frequenz-Atome bezeichnet, wobei jedes Atom eine gewisse Lage und Konzentration in Zeit und Frequenz hat. Die Frametheorie beschreibt unter welchen Bedingungen eine stabile und perfekte Rekonstruktion möglich ist.

Gabor-Systeme sind ein Teilgebiet der Gabor-Analyse. Sie sind im diskreten Fall im Hilbertraum $\mathcal{H} = \mathbb{C}^L$ definiert, wobei L der Signallänge entspricht. Die Zeit-Frequenz-Atome klassischer Gabor-Systeme (auch kurz Gabor-Atome) werden mithilfe von Translationen und Modulationen einer einzigen Fensterfunktion g gebildet, die zu einer festen Zeit-Frequenzauflösung führen. Translation \mathbf{T} und Modulation \mathbf{M} sind an dieser Stelle folgendermaßen definiert:

$$\mathbf{T}_m x(n) = x(n - m) \quad (2.23)$$

$$\mathbf{M}_k x(n) = x(n) \cdot e^{j2\pi kn/L} \quad (2.24)$$

Ein diskretes Gabor-System $\mathcal{G}(g, a, b)$ ist als

$$\mathcal{G}(g, a, b) = \{g_{m,k} = \mathbf{M}_{bk} \mathbf{T}_{am} g \text{ mit } k = 0, \dots, K - 1; m = 0, \dots, M - 1\} \quad (2.25)$$

definiert, wobei $M = \frac{L}{a}$ und $K = \frac{L}{b}$ entspricht. Die Gabor-Transformation zur Berechnung der Systemkoeffizienten besteht dann aus dem inneren Produkt zwischen dem Signal x und den Gabor-Atomen $g_{m,k}$:

$$c_{m,k} = \langle x, g_{m,k} \rangle = \sum_{n=0}^{L-1} x(n) \overline{g(n - ma)} \cdot e^{-j2\pi bk n/L} \quad (2.26)$$

Allgemein entspricht die Gabor-Transformation einer Auswertung bzw. Abtastung der STFT auf einer diskreten Teilmenge der Zeit-Frequenz-Ebene, die durch die Gitterparameter a und b beeinflusst wird. Allgemein gilt: Je kleiner das Produkt ab , desto redundanter ist das System. Die in Unterabschnitt 2.2.4 vorgestellte STFT führt durch die halbe Überlappung der Analyse-Blöcke bereits intuitiv zu einer geringeren Redundanz.

Gabor-Frames haben die Eigenschaft, dass die Existenz eines entsprechenden Dual-Frames $\mathcal{G}(\tilde{g}, a, b)$ mit derselben Struktur garantiert ist. Mit dessen Hilfe kann das ursprüngliche Signal perfekt rekonstruiert werden. Für weitere Details der Konstruktion und Berechnung von Gabor-Frames und den entsprechenden Dual-Frames wird auf die eingangs erwähnte Literatur verwiesen.

2.3.2 Von der festen zur variablen Zeit-Frequenzauflösung

Die Zeit-Frequenz-Atome werden bei klassischen Gabor-Systemen mithilfe von Translationen und Modulationen einer einzigen Fensterfunktion aufgebaut, wodurch (wie bei der STFT) eine feste Zeit-Frequenzauflösung erreicht wird. Demgegenüber stehen die *nichtstationären Gabor-Systeme*, die aus einer endlichen Menge von Fensterfunktionen und deren Modulationen aufgebaut sind. Durch die Flexibilität der Analyse-Fenster und der Abtastzeitpunkte ist eine variable Zeit-Frequenzauflösung möglich.

Änderung der Auflösung über die Zeit

Die variable Auflösung wird erreicht, indem an verschiedenen Zeitpunkten unterschiedliche adaptive Fenster zugelassen werden. Die Zeit-Frequenz-Atome eines solchen nicht-

stationären Gabor-Systems $\mathcal{G}(g_m, b_m)$ entstehen dann durch gewöhnliche Modulationen:

$$g_{m,k}(n) = \mathbf{M}_{kb_m} g_m(n) = g_m(n) e^{j2\pi kb_m n/L} \quad (2.27)$$

Jedes Fenster g_m liegt zentriert um die Zeitpunkte a_m und kann beliebig variiert werden, wodurch eine unregelmäßige Abtastung entlang der Zeitachse möglich wird. Pro Zeitpunkt wird die Frequenzachse jedoch regelmäßig abgetastet (äquidistante Verteilung). Die folgende Abbildung 2.10 zeigt beispielhaft ein mögliches Zeit-Frequenz-Gitter.

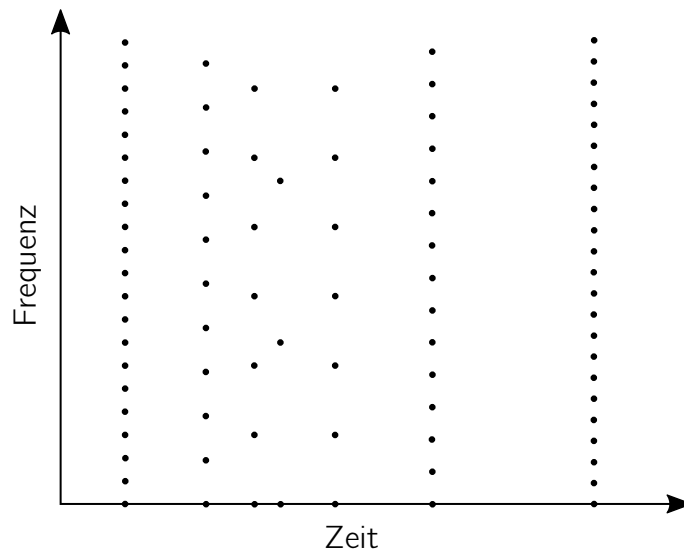


Abbildung 2.10: Änderung der Zeit-Frequenz-Auflösung über die Zeit (nach [Hol13, S. 68])

Die Systemkoeffizienten sind wieder durch das innere Produkt aus Eingangssignal und Zeit-Frequenz-Atome gegeben:

$$c_{m,k} = \langle x, g_{m,k} \rangle = \langle x, \mathbf{M}_{kb_m} g_m \rangle \quad (2.28)$$

Änderung der Auflösung über die Frequenz

Nichtstationäre Gabor-Systeme können ebenfalls so angewendet werden, dass die Zeit-Frequenzauflösung über die Frequenz verändert wird. Aus algorithmischer Sicht wird das nichtstationäre Gabor-System auf die Fouriertransformierte des Eingangssignals angewendet, um frequenzseitig die Bandbreite und Abtastdichte zu steuern. Die Filter g_k werden dafür direkt im Frequenzbereich als Bandpassfilter um die Mittenfrequenzen

ω_k entworfen. Die Zeit-Frequenz-Atome des frequenzseitigen nichtstationären Gabor-Systems $\mathcal{G}(g_k, a_k)$ ergeben sich dann durch Modulation der Filter bzw. Translation der entsprechenden Impulsantworten:

$$g_{m,k}(n) = \mathcal{F}^{-1}(\mathbf{M}_{-ma_k}g_k)(n) = \mathbf{T}_{ma_k}\mathcal{F}^{-1}(g_k)(n) \quad (2.29)$$

Durch diesen Aufbau kann die Frequenzachse unregelmäßig abgetastet werden. Die Abtastung entlang der Zeitachse kann durch den Parameter a_k je Frequenzkanal unterschiedlich gewählt werden. Bezogen auf einen Frequenzkanal ist die zeitliche Abtastung allerdings regelmäßig. a_k kann als individuelle Analyse-Schrittweite des k -ten Frequenzkanals angesehen werden und muss die folgende Bedingung erfüllen:

$$a_k \leq \frac{L}{L_k} \quad (2.30)$$

L_k bezeichnet die Länge des k -ten Filters und $m = 0, \dots, L/a_k - 1$. Die größtmögliche Wahl der Analyse-Schrittweite a_k für jeden Frequenzkanal führt dann zu einem System mit minimaler Redundanz. Die Gesamtanzahl der Analyse-Punkte pro Frequenzkanal ergibt sich zudem durch $N_k = L/a_k$. Die Abbildung 2.11 zeigt ein mögliches Zeit-Frequenz-Gitter für ein solches System.

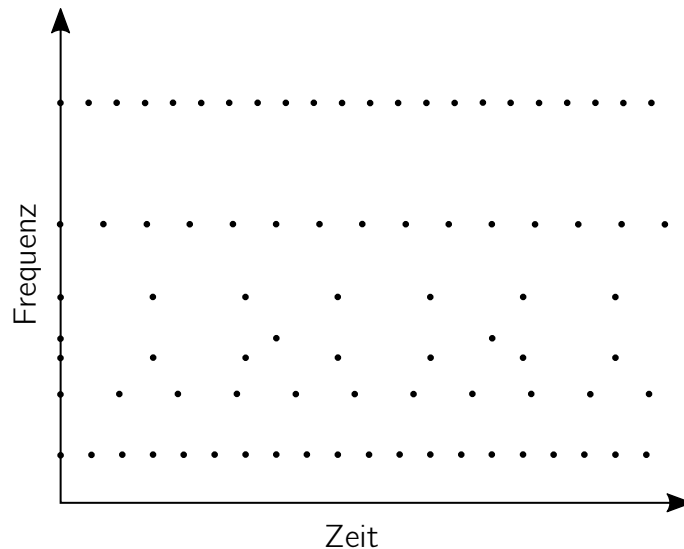


Abbildung 2.11: Änderung der Zeit-Frequenz-Auflösung über die Frequenz (nach [Hol13, S. 70])

Die Berechnung der Systemkoeffizienten ergibt sich folglich durch:

$$c_{m,k} = \langle x, g_{m,k} \rangle = \langle x, \mathcal{F}^{-1}(\mathbf{M}_{-ma_k} g_k) \rangle \quad (2.31)$$

2.3.3 Constant-Q Systeme

Mithilfe von nichtstationären Gabor-Systemen sind durch die Flexibilität der Analyse-Fenster Transformationen mit unterschiedlichsten Zeit-Frequenz-Gittern möglich. Die CQT kann mit einem frequenzseitig nichtstationären Gabor-System realisiert werden und eignet sich besonders für musikalische Signale, da die Verteilung der Frequenzstützstellen der logarithmischen Verteilung der musikalischen Noten entspricht.

Die Mittenfrequenzen f_k der Analyse-Filter g_k ergeben sich durch die Vorgabe der kleinsten zu analysierenden Frequenz f_{min} und der Anzahl von Filtern pro Oktave N_B :

$$f_k = f_{min} \cdot 2^{\frac{k-1}{N_B}} \quad \text{mit } k = 1, 2, \dots, K \quad (2.32)$$

K berechnet sich für einen vorgegebenen Frequenzbereich f_{min} bis $f_{max} < F_s/2$ durch:

$$K = \text{floor} \left[N_B \cdot \log_2 \left(\frac{f_{max}}{f_{min}} \right) \right] + 1 \quad (2.33)$$

Um eine perfekte Rekonstruktion zu gewährleisten, wird der verbleibende Frequenzbereich unterhalb von f_{min} und oberhalb von f_{max} bis hin zur Nyquist-Frequenz mit jeweils einem weiteren Filter erfasst. Insgesamt werden aus Symmetriegründen $2K + 2$ Filter benötigt:

$$f_k = \begin{cases} 0 & \text{für } k = 0 \\ f_{min} \cdot 2^{\frac{k-1}{N_B}} & \text{für } k = 1, \dots, K \\ F_s/2 & \text{für } k = K + 1 \\ F_s - f_{2K+2-k} & \text{für } k = K + 2, \dots, 2K + 1 \end{cases} \quad (2.34)$$

Die Bandbreite B_k eines Filters g_k wird mit $B_k = f_{k+1} - f_{k-1}$ festgelegt, sodass sich eine konstante Güte von $Q = f_k/B_k$ für $k = 1, \dots, K$ ergibt. Die Bandbreiten der Filter

können folgendermaßen zusammengefasst werden:

$$B_k = \begin{cases} 2f_{min} & \text{für } k = 0 \\ f_k/Q & \text{für } k = 1, \dots, K \\ F_s - 2f_K & \text{für } k = K + 1 \\ f_{2K+2-k}/Q & \text{für } k = K + 2, \dots, 2K + 1 \end{cases} \quad (2.35)$$

Die Bandbreite B_k (in Hz) kann über $L_k = B_k L / F_s$ in die Filterlänge (in Samples) umgerechnet werden. Die Filter werden direkt im Frequenzbereich mit klassischen Fensterfunktionen wie z. B. dem Von-Hann-Fenster mit der Länge L_k entworfen. Die folgende Abbildung 2.12 zeigt ein beispielhaftes Zeit-Frequenz-Gitter mit minimaler Redundanz eines solchen Constant-Q Systems.

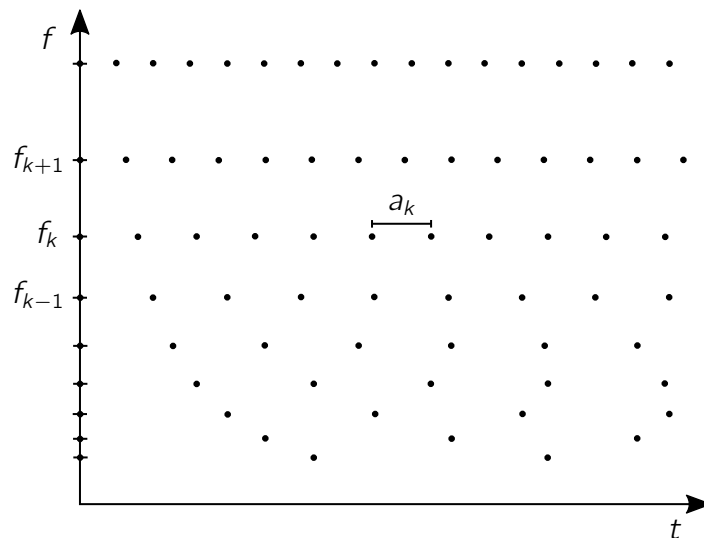


Abbildung 2.12: Zeit-Frequenz-Gitter eines Constant-Q Systems (nach [SKS13, S. 567])

2.3.4 Effiziente Implementierung

Nichtstationäre Gabor-Systeme können mithilfe der FFT sowie IFFT effizient implementiert werden, welches im Folgenden anhand der frequenzseitig nichtstationären Gabor-Systeme aufgezeigt wird.

Wie zuvor beschrieben, werden die Systemkoeffizienten durch das innere Produkt aus Eingangssignal x und Zeit-Frequenzatomen $g_{m,k}$ berechnet. Nach der parsevalschen

Formel ist dies mit dem inneren Produkt der Fourier-Transformierten gleichzusetzen:

$$c_{m,k} = \langle x, g_{m,k} \rangle = \langle \mathcal{F}x, \mathcal{F}g_{m,k} \rangle \quad (2.36)$$

Das Einsetzen der spezifischen Zeit-Frequenz-Atome für frequenzzeitige Systeme führt dann zu dem folgenden Ausdruck:

$$\begin{aligned} c_{m,k} &= \langle \mathcal{F}x, \mathcal{F}g_{m,k} \rangle = \langle \mathcal{F}x, \mathbf{M}_{-ma_k} g_k \rangle & (2.37) \\ &= \sum_n^{L-1} \mathcal{F}x(n) \cdot \overline{\mathbf{M}_{-ma_k} g_k} \\ &= \sum_n^{L-1} \mathcal{F}x(n) \cdot \overline{g_k}(n) \cdot e^{j2\pi nma_k/L} \\ &\stackrel{\wedge}{=} \text{IFFT}_{N_k} [\text{FFT}_L(x) \overline{g_k}] \end{aligned}$$

Während vom Eingangssignal die gesamte FFT der Länge L berechnet werden muss, wird pro Frequenzkanal lediglich eine IFFT der Länge N_k berechnet. Die Zeit-Frequenz-Ebene wird insgesamt auf einer diskreten Teilmenge ausgewertet, die durch a_k bzw. N_k vorgegeben wird. Dies kann auch als Unterabtastung des Ausgangs $c_{m,k}$ einer Filterbank interpretiert werden, bei der die jeweils neue Abtastfrequenz $F_s^k = F_s/a_k$ beträgt. Bei jedem Frequenzkanal werden die DFT Koeffizienten nach der Anwendung des Filters g_k um f_k nach unten verschoben, sodass diese zentriert um die Null im Frequenzintervall $(-F_s^k/2, F_s^k/2]$ liegen. Alle DFT Koeffizienten außerhalb von $(-F_s^k/2, F_s^k/2]$ werden verworfen, weshalb an dieser Stelle von Unterabtastung im Frequenzbereich gesprochen wird. Die endgültigen Systemkoeffizienten werden durch eine IFFT der verbleibenden DFT Koeffizienten berechnet. Die Systemkoeffizienten $c_{m,k}$ entsprechen einer unregelmäßigen Matrix mit $2K + 2$ Spalten, wobei jede Spalte $N_k = L/a_k$ Einträge hat:

$$c \in \mathbb{C}^{L/a_k \times 2K+2}$$

2.3.5 Realisierung als Echtzeitsystem

Im vorherigen Kapitel wurde aufgezeigt, dass eine FFT über das gesamte Eingangssignal berechnet werden muss, um die Systemkoeffizienten zu erhalten. Die Realisierung als Echtzeitsystem ist somit in dieser Form nicht möglich.

In Unterabschnitt 2.2.4 wurde die Kurzzeit-Spektralanalyse (STFT) durch die Zerlegung des Eingangssignals in überlappende Blöcke eingeführt. Dieses Prinzip ist auch

bei nichtstationären Gabor-Systemen möglich und wurde von *Holighaus et al.* als *Sliced Constant-Q Transformation (sliCQ)* in [Hol+13] vorgestellt. Da allerdings die Filter g_k einen kompakten Träger im Frequenzbereich aufweisen, sind die entsprechenden Impulsantworten unendlich lang (*Infinite Impulse Response (IIR)*). Die Autoren empfehlen daher die Zerlegung des Signals mithilfe von Fenstern, die durch beidseitiges Zero-Padding auf die doppelte Länge erweitert sind. Zudem ist für die Konstruktion der Filter g_k die Verwendung einer Fensterfunktion mit hoher Nebenkeulen-Dämpfung zu empfehlen (z. B. Blackman). Dieses Vorgehen reduziert ebenfalls zeitliches Aliasing, welches nach einer spektralen Manipulation auftritt.

Die Anwendung der CQT auf kürzere Blöcke hat erheblichen Einfluss auf die gewünschte Platzierung der Frequenzstützstellen, die (als Folge aus dem Unschärfe-Prinzip) nicht zwangsläufig sichergestellt sein muss. Die daraus entstehenden Konsequenzen stehen in Konflikt mit der anvisierten Latenzoptimierung und werden in Unterabschnitt 5.2.4 anhand des Pitch Shifting Algorithmus diskutiert.

3 Pitch Shifting

In diesem Kapitel wird eine Einführung in den Pitch Shifting Effekt gegeben. In Abschnitt 3.1 wird dazu zunächst die Definition von Pitch Shifting und der Zusammenhang mit Time Scaling erläutert. Aufgrund der Komplexität des Pitch Shifting Effekts können in dem frequenzskalierten Signal unterschiedlichste Artefakte auftreten. Typische Artefakte werden in Abschnitt 3.2 aufgezeigt.

Das grundlegende Ziel von Pitch Shifting ist, dass das transponierte Signal so klingt, als ob es direkt in der anvisierten Tonhöhe aufgenommen worden wäre. Die Klangfarbe (englisch *Timbre*) soll also möglichst erhalten bleiben. In Abschnitt 3.2 wird die Definition von Formanten und die Wirkung des Pitch Shifting Effekts auf die Lage der Formanten bzw. auf die Form der spektralen Hüllkurve diskutiert.

In diesem Kapitel wird zunächst die grundlegende Wirkungsweise eines Pitch Shifters diskutiert. Eine detaillierte Übersicht und Erläuterung verschiedener Algorithmen wird, in Zeit- und Frequenzbereich unterteilt, mit Kapitel 4 und Kapitel 5 gegeben.

3.1 Definition und Zusammenhang mit Time Scaling

Pitch Shifting (zu deutsch Tonhöhenverschiebung oder Tonhöhenänderung) ist ein Audioeffekt, der die Tonhöhe eines Audiosignals verändert, ohne dabei die zeitliche Dauer des Signals zu beeinflussen. Ein weiterer, weniger häufig verwendeter Begriff ist zudem Frequency Scaling.

Demgegenüber steht das sogenannte Time Scaling (auch Time Stretching genannt), welches die Länge bzw. die Geschwindigkeit eines Audiosignals ohne Beeinflussung der Tonhöhe verändert. So lassen sich Signale im Nachhinein zeitlich strecken oder stauchen. Time Scaling stellt damit das genaue Gegenteil zum Pitch Shifting dar, wobei beide Effekte gleichermaßen schwierig in hoher Klangqualität zu berechnen sind. Im Gegensatz zum Pitch Shifting kann Time Scaling aufgrund der Veränderung der Signallänge nicht in Echtzeitanwendungen verwendet werden [DM16].

Eine Schallplatte kann mit einfachen Mitteln schneller abgespielt werden, wobei neben der kürzeren Abspieldauer gleichzeitig die Tonhöhe erhöht wird. Die Tonhöhe und die

zeitliche Dauer lassen sich somit nicht unabhängig voneinander einstellen. Der gleiche Effekt lässt sich erzeugen, wenn ein bereits abgetastetes Signal mit einer höheren Abtastrate abgespielt wird. Mithilfe einer Abtastratenkonvertierung (Resampling) kann zudem die originale Abtastrate beibehalten werden, indem das bereits abgetastete Signal zunächst durch die Abtastratenkonvertierung auf eine niedrigere Abtastrate umgesetzt wird. Das anschließende Abspielen des dezimierten Signals mit der originalen (höheren) Abtastrate führt ebenfalls zu einer höheren Tonhöhe und verkürzten Abspieldauer [DM16, S. 21]. Dieser Zusammenhang ist in der folgenden Abbildung 3.1 anhand einer einfachen Sinusschwingung dargestellt.

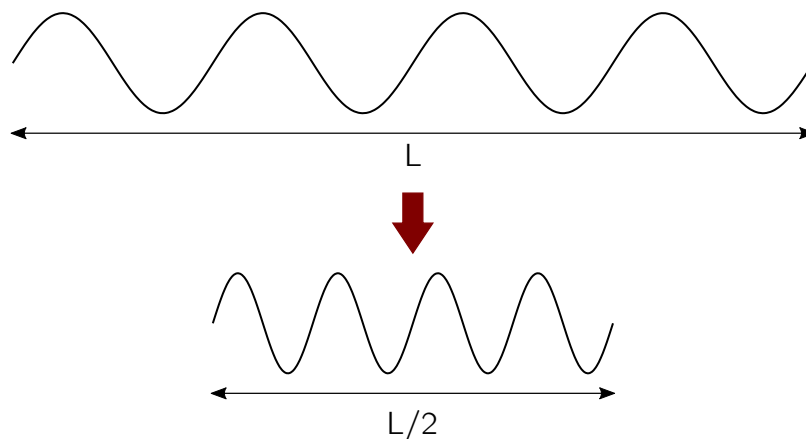


Abbildung 3.1: Doppelte Abspielgeschwindigkeit einer Sinusschwingung führt zu einer Erhöhung der Frequenz und Verkürzung der Signaldauer

Die voneinander unabhängige Manipulation von Tonhöhe bzw. zeitlicher Dauer ist hingegen deutlicher schwieriger zu realisieren und wird stark erforscht. In der Literatur wird als Maß für die zeitliche Veränderung bzw. die Änderung der Tonhöhe der Faktor α eingeführt. Bei $\alpha > 1$ wird das Signal gestreckt bzw. in der Tonhöhe erhöht, $\alpha < 1$ führt hingegen zu einer Stauchung des Signals bzw. Verringerung der Tonhöhe [DM16].

Pitch Shifting als Kombination aus Time Scaling und Resampling

Pitch Shifting lässt sich durch eine Kombination von Time Scaling und Resampling realisieren. So kann beispielsweise die Tonhöhe um den Faktor α erhöht werden, indem das Signal zunächst durch einen Time Scaling Algorithmus um den gleichen Faktor α zeitlich gestreckt wird. Die Tonhöhe ist gleich, das Signal aber länger. Durch Resampling wird das Signal wieder auf die ursprüngliche Länge gebracht, die Tonhöhe aber

gleichzeitig erhöht, da das Signal mit der ursprünglichen Abtastfrequenz abgespielt wird [DM16, S. 21]. Dieser Ablauf ist in Abbildung 3.2 visualisiert.

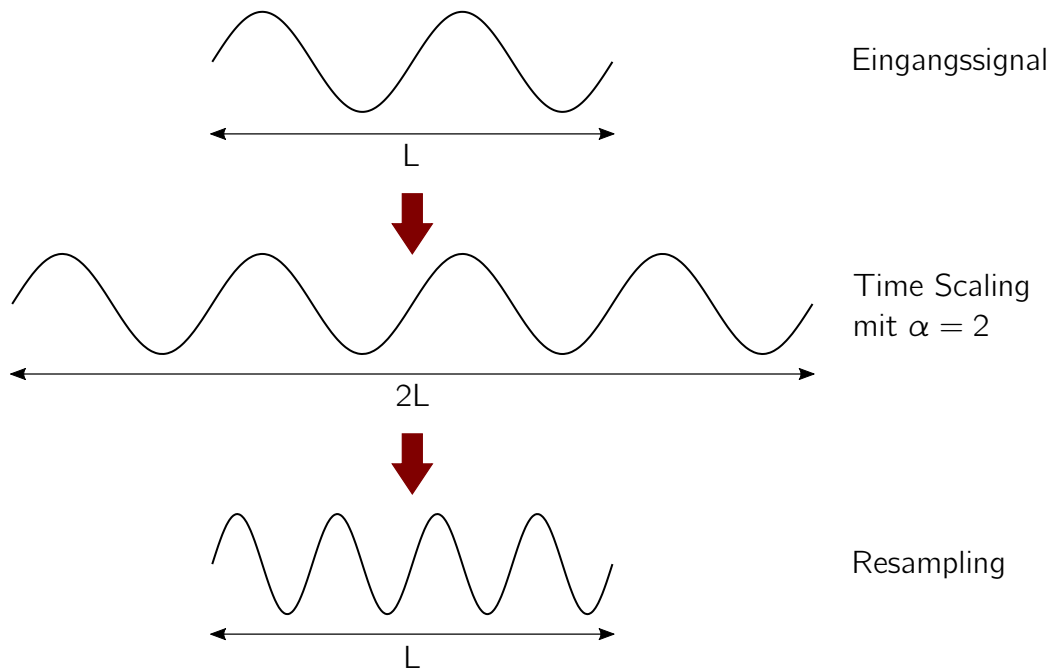


Abbildung 3.2: Pitch Shifting als Kombination aus Time Scaling und Resampling

Wirkung eines Pitch Shifters auf das Spektrum

Eine direkte Berechnung des Pitch Shifting Effekts ohne den Umweg über Time Scaling ist ebenso möglich. Für ein besseres Verständnis der Wirkung eines Pitch Shifters auf das Spektrum eines Signals, ist die Betrachtung der spektralen Verteilung musikalischer Noten hilfreich. Die einzelnen Noten sind nicht linear sondern logarithmisch über die Frequenzachse verteilt. Eine Oktave entspricht der Verdoppelung (bzw. Halbierung) der Frequenz, wobei eine Oktave wiederum aus 12 Halbtönen besteht. Ein Halbton wird in 100 Cent unterteilt [Gör11]. Die folgende Tabelle 3.1 zeigt beispielhaft die entsprechenden Frequenzen für drei Oktaven ab der tiefsten Note der E-Gitarre (E2).

Tabelle 3.1: Musikalische Noten mit den entsprechenden Frequenzen

Note	Frequenz in Hz	Note	Frequenz in Hz	Note	Frequenz in Hz
E2	82,41	E3	164,81	E4	329,63
F2	87,31	F3	174,61	F4	349,23
F#2	92,50	F#3	185,00	F#4	370,00
G2	98,00	G3	196,00	G4	392,00
G#2	103,83	G#3	207,65	G#4	415,30
A2	110,00	A3	220,00	A4	440,00
B2	116,54	B3	233,08	B4	466,16
H2	123,47	H3	246,94	H4	493,88
C3	130,81	C4	261,63	C4	523,25
C#3	138,59	C#4	277,18	C#5	554,37
D3	146,83	D4	293,66	D5	587,33
D#3	155,56	D#4	311,13	D#5	622,25

Um ein Signal in der Tonhöhe zu verändern, müssen, aufgrund der logarithmischen Verteilung der Noten, die einzelnen Frequenzanteile f_i mit dem Faktor α multipliziert werden. Dies entspricht einer Skalierung des Amplitudenspektrums.

$$f_i^y = f_i^x \cdot \alpha \quad \text{mit } \alpha \in \mathbb{R}^+ \quad (3.1)$$

Aus diesem Grund wird Pitch Shifting auch Frequency Scaling genannt. Der Faktor α kann grundsätzlich den in Gleichung 3.1 genannten Zahlenbereich annehmen. Im musikalischen Kontext macht es allerdings überwiegend Sinn nur eine Abstufung in Halbtonschritten vorzunehmen. Die Berechnungsvorschrift von α für die gewünschte Verschiebung um s Halbtöne ergibt sich direkt aus der logarithmischen Verteilung der Noten:

$$\alpha = 2^{(s/12)} \quad \text{mit } s \in \mathbb{Z} \quad (3.2)$$

Für eine Verschiebung der Tonhöhe um beispielsweise zwei Halbtöne nach oben bzw. unten ergibt sich für α :

$$\alpha_{s=2} = 2^{(2/12)} = 1,122 \quad (3.3)$$

$$\alpha_{s=-2} = 2^{(-2/12)} = 0,891 \quad (3.4)$$

In Abbildung 3.3 ist beispielhaft das Pitch Shifting um zwei Halbtöne nach oben dargestellt. Das Amplitudenspektrum wird so skaliert, dass der G-Dur Akkord im Eingangssignal als A-Dur Akkord am Ausgang erscheint.

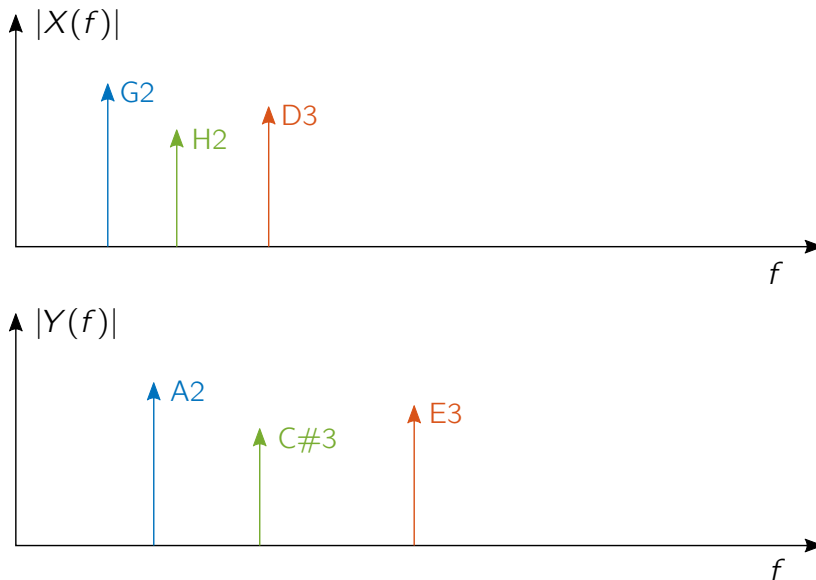


Abbildung 3.3: Beispiel für Pitch Shifting um zwei Halbtöne nach oben (ohne Betrachtung von Obertönen)

Abgrenzung zum Frequency Shifting

Die Begriffe Pitch Shifting bzw. Frequency Scaling sind an dieser Stelle klar vom Frequency Shifting abzugrenzen. Beim Frequency Shifting werden die einzelnen Frequenzanteile f_i um eine feste Frequenz f_{shift} verschoben, sodass folgender Zusammenhang gilt:

$$f_i^y = f_i^x + f_{shift} \quad \text{mit } f_{shift} \in \mathbb{R} \quad (3.5)$$

Durch die Frequenzverschiebung werden allerdings sämtliche harmonische Zusammenhänge zerstört. Das musikalische Eingangssignal wird dadurch stark verfremdet.

3.2 Herausforderungen und typische Artefakte

Die Berechnung des Pitch Shifting Effekts ist sowohl im Zeit- als auch Frequenzbereich möglich, wobei jeder Bereich gewisse Vor- und Nachteile mit sich bringt. Um eine hohe Klangqualität zu erreichen, werden aufwendige Algorithmen benötigt, die teils mit einem hohen Rechenaufwand einhergehen. Wie gut die Klangqualität eines Pitch Shifting Algorithmus ist, hängt stark von der Struktur des Eingangssignals ab. So erzielen Algorithmen im Zeitbereich tendenziell bessere Ergebnisse bei rein perkussiven Klängen, haben aber demgegenüber eher Probleme bei harmonischen Klängen. Algorithmen im Frequenzbereich erzielen durch die spektrale Analyse bessere Ergebnisse hinsichtlich harmonischen Klängen, allerdings sind perkussive bzw. transientenreiche Klänge problematisch [DM16].

Eine weitere Einteilung ist zudem die Unterscheidung in monophone und polyphone Signale. Als monophon oder einstimmig wird ein Signal bezeichnet, wenn zu jedem Zeitpunkt nur ein einziger Ton (bestehend aus Grundton und Obertönen) erzeugt wird. Beispiele hierfür sind die menschliche Stimme oder auch Blech- und Holzblasinstrumente. Demgegenüber stehen polyphone oder mehrstimmige Signale, die mindestens zwei Töne zu einem Zeitpunkt enthalten. Die Spektren solcher Signale sind komplexer und die Berechnung der Tonhöhenverschiebung daher deutlich aufwendiger und schwieriger. Eine Gitarre mit sechs Saiten kann maximal sechs Töne gleichzeitig erzeugen. Ein solches Signal ist als stark polyphon anzusehen.

An dieser Stelle sei hervorzuheben, dass es im mathematischen Sinne keinen perfekten Pitch Shifting Algorithmus gibt. Die Ursache hierfür liegt letztlich in der *Küpfmüllerschen Unbestimmtheitsrelation* (siehe Unterabschnitt 2.2.2), die beschreibt, dass ein Signal nicht gleichzeitig exakt in Zeit und Frequenz lokalisierbar ist. Durch diese physikalische Grenze erzeugt letztlich jeder Algorithmus Artefakte. Die Entwicklung und Optimierung eines Pitch Shifting Algorithmus wird daher zu vielen Teilen ebenfalls nach hörtechnischen Gesichtspunkten durchgeführt, da der wahrgenommene Höreindruck zählt und psychoakustische Aspekte gezielt ausgenutzt werden.

Im Zusammenhang mit Pitch Shifting werden in zahlreichen Forschungsarbeiten unterschiedliche Artefakte beschrieben, die häufig charakteristisch für den Zeit- oder Frequenzbereich sind [DM16; LD97; JAS08]. Typische Artefakte im Zeitbereich sind beispielsweise:

- Duplizierung oder Überspringen von Transienten
- Unstetigkeiten bzw. Phasensprünge
- Verstimmung (englisch *Detuning*)
- Effekte ähnlich einer periodischen Frequenzmodulation (*Warbling*)
- Effekte ähnlich einer Amplitudenmodulation (*Tremolo*-Effekt)

Folgende Artefakte sind hingegen charakteristisch für den Frequenzbereich:

- Transienten-Verschmierung (englisch *Transient Smearing*)
- Verminderung der Direktheit bzw. Präsenz (*Phasiness / Reverberation*)
- Effekte ähnlich einer Amplitudenmodulation (*Tremolo*-Effekt)

In den Kapiteln 4 und 5 werden die in dieser Arbeit untersuchten Algorithmen in Zeit- und Frequenzbereich unterteilt und vorgestellt. Die Ursachen und Auswirkungen der oben genannten Artefakte werden an entsprechender Stelle anhand des jeweiligen Algorithmus angeführt und erläutert.

Klangfarbe und Formanten

Zwei Personen, die den gleichen Ton singen, klingen trotzdem unterschiedlich. Durch Mundhöhle, Nasennebenhöhlen oder auch Rachen werden Resonanzräume gebildet, die aufgrund der individuellen anatomischen Form und Größe bei jedem Menschen unterschiedlich sind. Die von den Resonanzfrequenzen umgebenen Frequenzbereiche werden Formanten genannt und sind im Wesentlichen unabhängig von dem gesungenen Grundton. In diesen Frequenzbereichen findet eine (relativ gesehen) hohe Verstärkung statt. Mithilfe von Sprach- bzw. Gesangstechniken kann allerdings die Lage und Höhe der Formanten durch gezielte Veränderung der Resonanzräume, unabhängig vom Grundton, verändert werden. Diese Techniken werden beispielsweise bei Stimmenimitationen angewandt. Neben dem Grundton, der die wahrgenommene Tonhöhe vorgibt, werden ebenfalls weitere Harmonische erzeugt, die durch die Resonanzräume in ihrer Amplitude beeinflusst werden. Die daraus resultierende spektrale Zusammensetzung führt zu einer für jeden Menschen individuellen Klangfarbe (englisch *Timbre*) [Wei08].

Formanten lassen sich ebenfalls bei Instrumenten definieren, weshalb sich jedes Gitarrenmodell durch eine individuelle Klangfarbe auszeichnet. Die spektrale Lage und Höhe der Formanten wird bei einer akustischen Gitarre beispielsweise durch den geometrischen Aufbau (Korpus, Hals etc.), aber auch durch die Position der gegriffenen Note auf dem Griffbrett, beeinflusst. Zum Beispiel kann die Note D3 an drei verschiedenen

Positionen auf dem Griffbrett erzeugt werden - als leere D-Saite, im fünften Bund der A-Saite oder im zehnten Bund der tiefen E-Saite. Alle drei Positionen unterscheiden sich klanglich, wobei die letzte Position am wärmsten klingt. Die Formanten bzw. die Obertonstruktur (die durch die Formanten beeinflusst wird) sind im Allgemeinen nicht die einzigen Attribute, die die Klangfarbe einer Stimme oder eines Instruments definieren. Das Gehör wertet neben der spektralen Zusammensetzung beispielsweise ebenfalls die zeitlich Einhüllende aus [Wei08, S. 65-73]. Im Zusammenhang mit Pitch Shifting wird jedoch überwiegend und (ohne weitere Vorkehrungen) unvermeidbar die Lage der Formanten verändert und so die Klangfarbe verfremdet. Diese Herausforderung wird in der Literatur insbesondere bei der Tonhöhenverschiebung von Stimmen in zahlreichen Veröffentlichungen untersucht, wie z. B. in [Bri95; Roy19; Zöl11].

Beim Pitch Shifting wird daher häufig zwischen Algorithmen mit Erhalt der Formanten (englisch *Formant Preservation*) und ohne Erhalt der Formanten (englisch *Non Formant Preservation*) unterschieden. In Abbildung 3.4 werden beide Arten anhand einer gesungenen Note C4 visualisiert, die um vier Halbtöne nach oben zur Note E4 verschoben wird. Durch die Tonhöhenverschiebung wird, wie in Abschnitt 3.1 erklärt, das gesamte Amplitudenspektrum skaliert. Im Falle von harmonischen Signalen werden zwar alle Harmonischen in der Frequenz korrekt skaliert, allerdings werden die Amplituden der einzelnen Harmonischen einfach beibehalten und so ebenfalls die Lage der Formanten skaliert (b). Wird über das Spektrum eine Hüllkurve (englisch *Spectral Envelope*) gelegt, wird diese Hüllkurve beim Pitch Shifting mitskaliert. Die Klangfarbe ist allerdings von der Lage der Formanten abhängig. Durch die nun beispielsweise höhere Lage der Formanten entsteht der bei Stimmen bekannte „Mickey Mouse“ Effekt - die originale Klangfarbe geht verloren und die Stimme klingt künstlich und piepsig. Auf der anderen Seite klingt die Stimme beim Pitch Shifting nach unten unnatürlich warm und wuchtig. Für harmonische Signale ist es daher wichtig die originale Hüllkurve auch nach dem Pitch Shifting beizubehalten, wodurch ebenfalls die Lage und Höhe der Formanten, und damit die Klangfarbe, erhalten bleibt (c). Nach [DM16, S. 22-23] muss dazu die Hüllkurve $\Gamma_m(k)$ des originalen Spektrums $X_m(k)$ sowie die Hüllkurve $\Gamma_m^{shift}(k)$ des in der Tonhöhe verschobenen Spektrums $X_m^{shift}(k)$ bestimmt werden. Das Amplitudenspektrum des gewünschten Signals $x^{Mod}(n)$ ergibt sich aus dem Verhältnis beider Hüllkurven:

$$|X_m^{Mod}(k)| = |X_m^{Shift}(k)| \frac{\Gamma_m(k)}{\Gamma_m^{shift}(k)} \quad (3.6)$$

Das Signal $x^{Mod}(n)$ enthält nun den Frequenzinhalt von $X_m^{Shift}(k)$, besitzt jedoch die spektrale Hüllkurve $\Gamma_m(k)$ von $X_m(k)$.

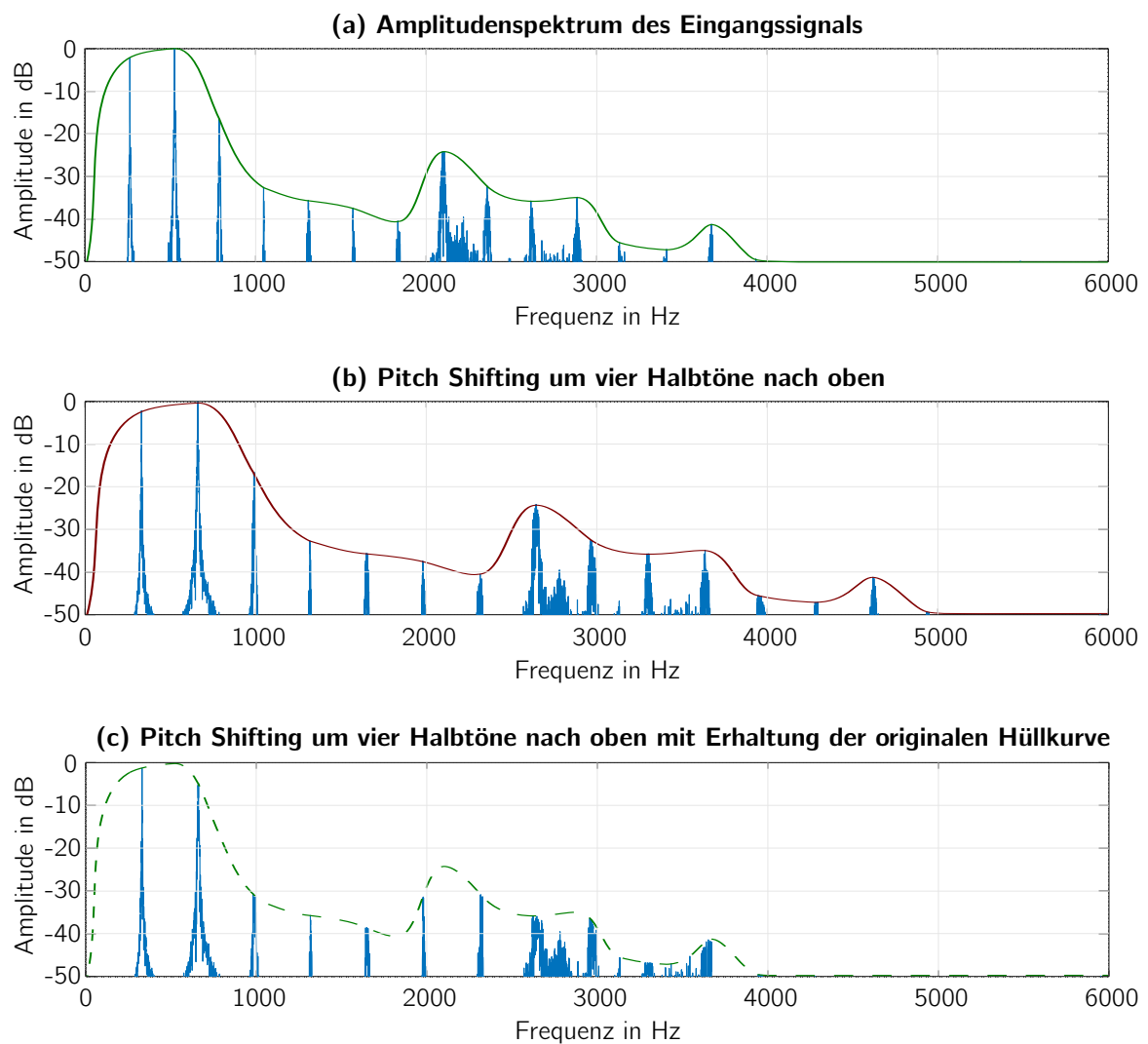


Abbildung 3.4: Erhaltung der Formanten durch Anpassung der spektralen Hüllkurve

Die Berechnung der originalen Hüllkurve und Anpassung der Hüllkurve des frequenzskalierten Signals kann somit unabhängig und nach dem eigentlichen Pitch Shifting durchgeführt werden. Die Herausforderung besteht hauptsächlich in der robusten Berechnung der Hüllkurve eines Signals. In [Zöl11, S. 280] werden drei Möglichkeiten zur Berechnung der spektralen Hüllkurve vorgeschlagen:

- Channel Vocoder: Bestimmung der Amplituden einzelner Frequenzbänder
- Linear Prediction Coding (LPC): Design eines rekursiven Filters (nur Pole), welches den spektralen Inhalt repräsentiert
- Cepstrum: Berechnung des gleitenden Mittelwerts mittels Tiefpassfilter im Cepstrum-Bereich

Mit [GT09] und [RR05] sind zwei Veröffentlichungen zu finden, in denen die Anwendung der Cepstrum- bzw. LPC-Technik in Bezug auf Pitch Shifting vorgestellt wird. Ein Vergleich verschiedener Ansätze ist darüber hinaus beispielsweise in [Zöl11] und [Roy19] zu finden.

Dynamische Formanten

In den folgenden Abbildungen 3.5 bis 3.8 werden unterschiedliche Signale (weibliche Stimme, akustische Gitarre und E-Gitarre) einer Tonhöhenverschiebung ohne und mit Erhaltung der spektralen Hüllkurve bzw. Formanten unterzogen. Für einen Vergleich der frequenzskalierten Signale im Bezug auf eine originalgetreue Klangfarbe werden im Vorwege ebenfalls Signale in der jeweils anvisierten Tonhöhe aufgenommen. Diese Signale sind in den jeweils ersten beiden Spektrogrammen dargestellt. Als Pitch Shifting Algorithmus wurde an dieser Stelle auf die Implementierung des Phase Vocoder auf Basis der STFT von *Driedger und Müller* [DM18] zurückgegriffen (detaillierte Beschreibung des Phase Vocoder in Abschnitt 5.1). Die aufgenommenen und berechneten Signale liegen als Hörbeispiele auf der beigefügten CD (Anhang B) vor.

In Abbildung 3.5 ist die Aufnahme einer weiblichen Stimme dargestellt, die in (a) die Note C4 und in (b) die Note E4 singt. In (c) ist nun das Spektrogramm des frequenzskalierten Signals aufgezeigt. Der Vergleich zu (b) zeigt, dass die einzelnen Frequenzanteile korrekt in der Frequenz zu der Note E4 skaliert wurden. Wie bereits in Abbildung 3.4 erläutert, wird allerdings ebenfalls die spektrale Hüllkurve von (a) skaliert, sodass die Formanten im Bereich 2.1 kHz – 2.6 kHz nun im Bereich 2.6 kHz – 3.3 kHz liegen. Der hörtechnische Eindruck zeigt den bereits erwähnten „Mickey Mouse“ Effekt. Die Anpassung der spektralen Hüllkurve nach Gleichung 3.6 zeigt in (d) eine deutliche Verbesserung der Lage der oberen Formanten. In (b) ist allerdings deutlich zu sehen, dass die ursprünglich vorhandene 3. Harmonische bei E4 nur noch schwach vorhanden ist. Dies lässt sich durch eine unbewusste Veränderung des Vokaltrakt (und damit der Resonanzräume) der Sängerin erklären, um die Note E4 singen zu können. Da allerdings für die Anpassung der spektralen Hüllkurve in (d) die Hüllkurve von (a) als Grundlage verwendet wird, erscheint die 3. Harmonische mit derselben Amplitude. Dieses Beispiel

zeigt, dass Formanten nur teilweise statisch sind. Sich je nach Grundton dynamisch verändernde Formanten lassen sich nach dem Ansatz in Gleichung 3.6 nicht korrigieren, da dies Vorkenntnisse über die Stimme bzw. das Instrument erfordert.

Das Beispiel einer akustischen Gitarre in Abbildung 3.6 zeigt dieselbe Problematik. Durch die Konstruktion der Gitarre zeigt sich in (a) bei der Note G2 ein spezieller zeitlicher Verlauf. In dem Moment des Anschlags wird eine vergleichsweise hohe Amplitude beim Grundton und der 2. Harmonischen erzeugt, die jedoch sehr schnell wieder abfällt. Die 3. und 4. Harmonische klingen dagegen lange aus. Der Anschlag der Note D3 im zehnten Bund der tiefen E-Saite in (b) zeigt dagegen ein komplett anderes Verhalten, da hier Grundton und 2. Harmonische am längsten ausklingen. Das frequenzskalierte Signal in (d) ist daher weit davon entfernt die originale Klangfarbe der Gitarre zu erhalten. Hier wäre eine weitaus komplexere Anpassung von spektraler Hüllkurve und zeitlicher Einhüllenden notwendig, die ebenfalls Vorkenntnisse über die verwendete Gitarre erfordern würde.

In Abbildung 3.7 und 3.8 wird zudem anhand einer E-Gitarre deutlich, dass die Position der gegriffenen Note auf dem Griffbrett starke Auswirkungen auf das Obertonspektrum hat. Umso höher die Position auf dem Griffbrett ist, desto wärmer (obertonärmer) ist der entstehende Ton. Wird nun die Note G2 in der Tonhöhe nach D3 verschoben (Abbildung 3.7), ist das Obertonspektrum auch nach der Anpassung der spektralen Hüllkurve zu stark ausgeprägt, sodass der Ton heller als in (b) klingt. Beim Pitch Shifting nach unten von D3 nach G2 (Abbildung 3.8) fehlt hingegen in (d) das ausgeprägte Obertonspektrum aus (b), sodass der Ton deutlich zu warm klingt.

Insgesamt lässt sich feststellen, dass die Anpassung der spektralen Hüllkurve nach Gleichung 3.6 zwar deutliche Verbesserungen in Bezug auf den Erhalt der Klangfarbe ermöglicht, allerdings bei stark dynamischem Verhalten der Formanten und zeitlicher Einhüllenden scheitert.

Aufgrund des Ziels der Latenzoptimierung wird der Fokus auf den reinen Pitch Shifting Effekt und den damit verbundenen Artefakten aus Abschnitt 3.2 gelegt. Die optimale Korrektur der spektralen Hüllkurve kann separat und unabhängig vom Pitch Shifting berechnet werden und ist nicht Teil dieser Arbeit.

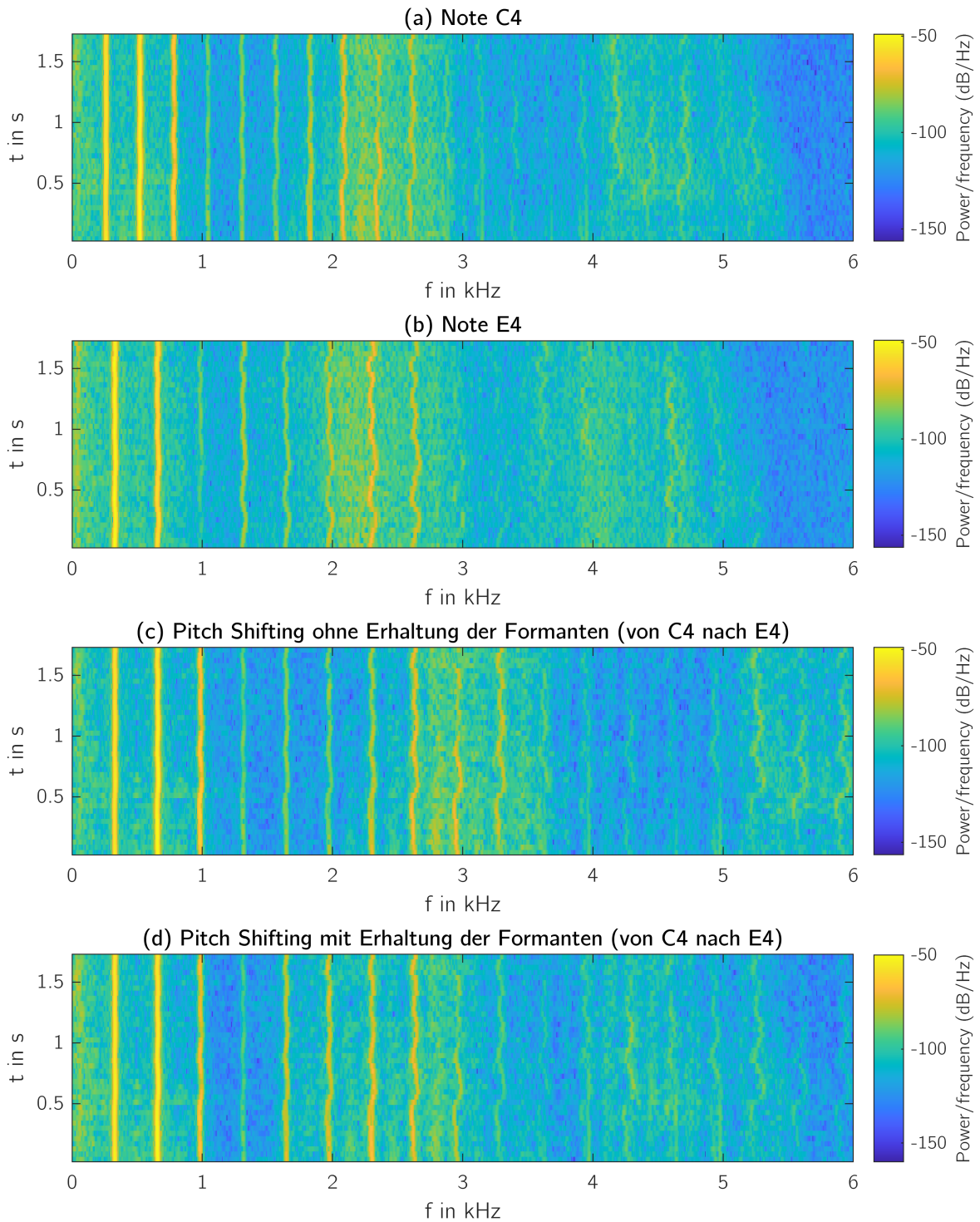


Abbildung 3.5: Pitch Shifting einer weiblichen Stimme um vier Halbtöne nach oben

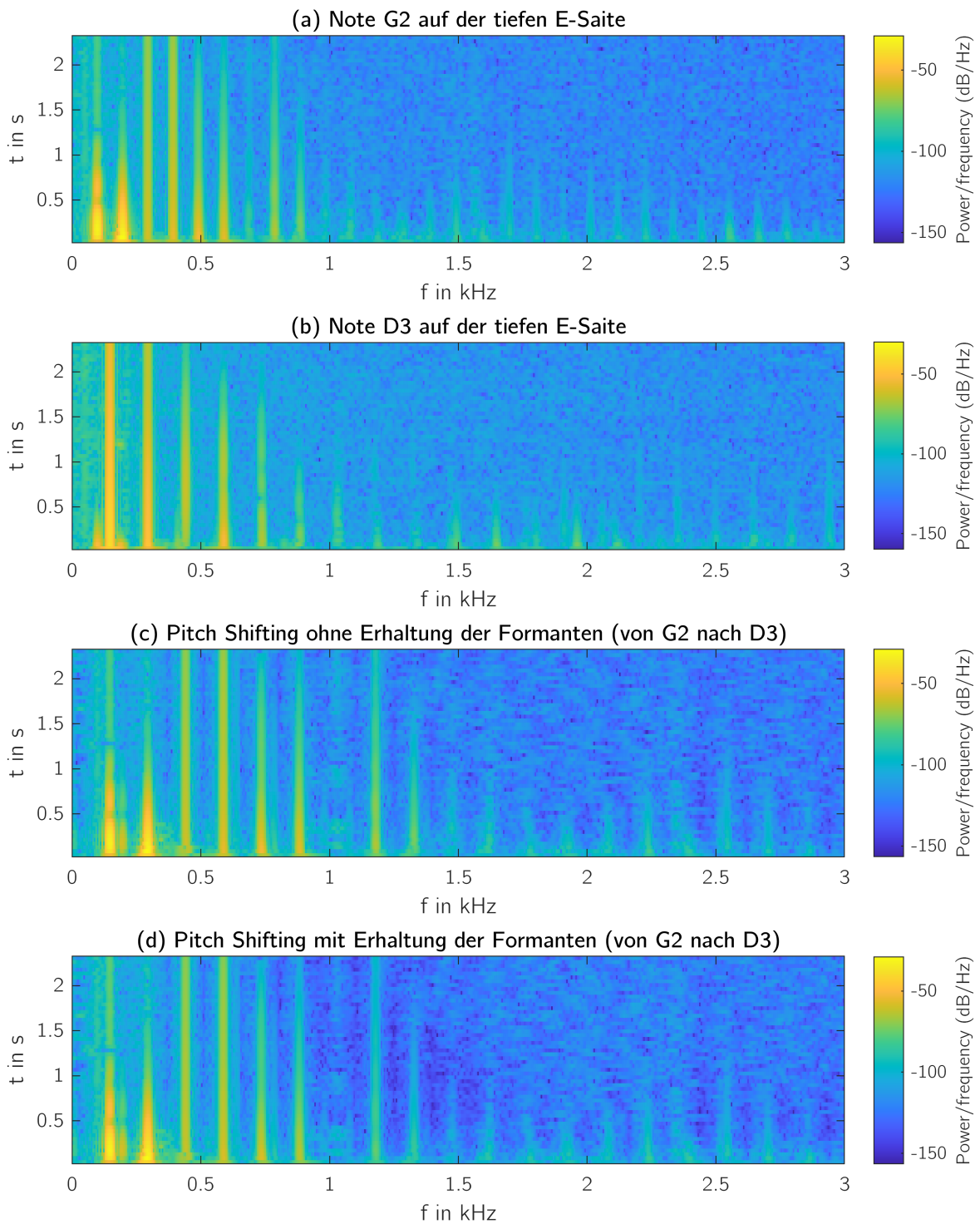


Abbildung 3.6: Pitch Shifting einer akustischen Gitarre um sieben Halbtöne nach oben

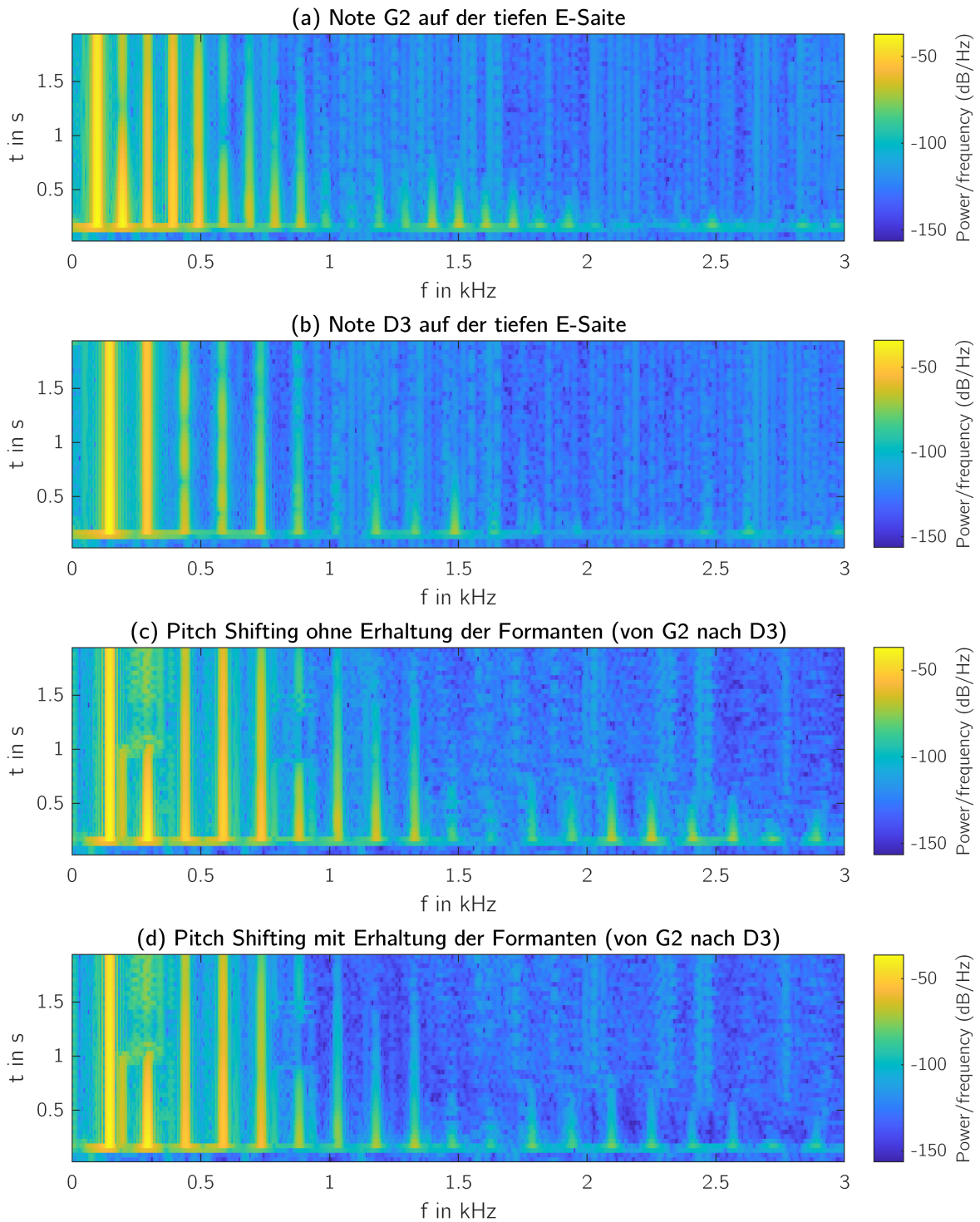


Abbildung 3.7: Pitch Shifting einer E-Gitarre um sieben Halbtöne nach oben

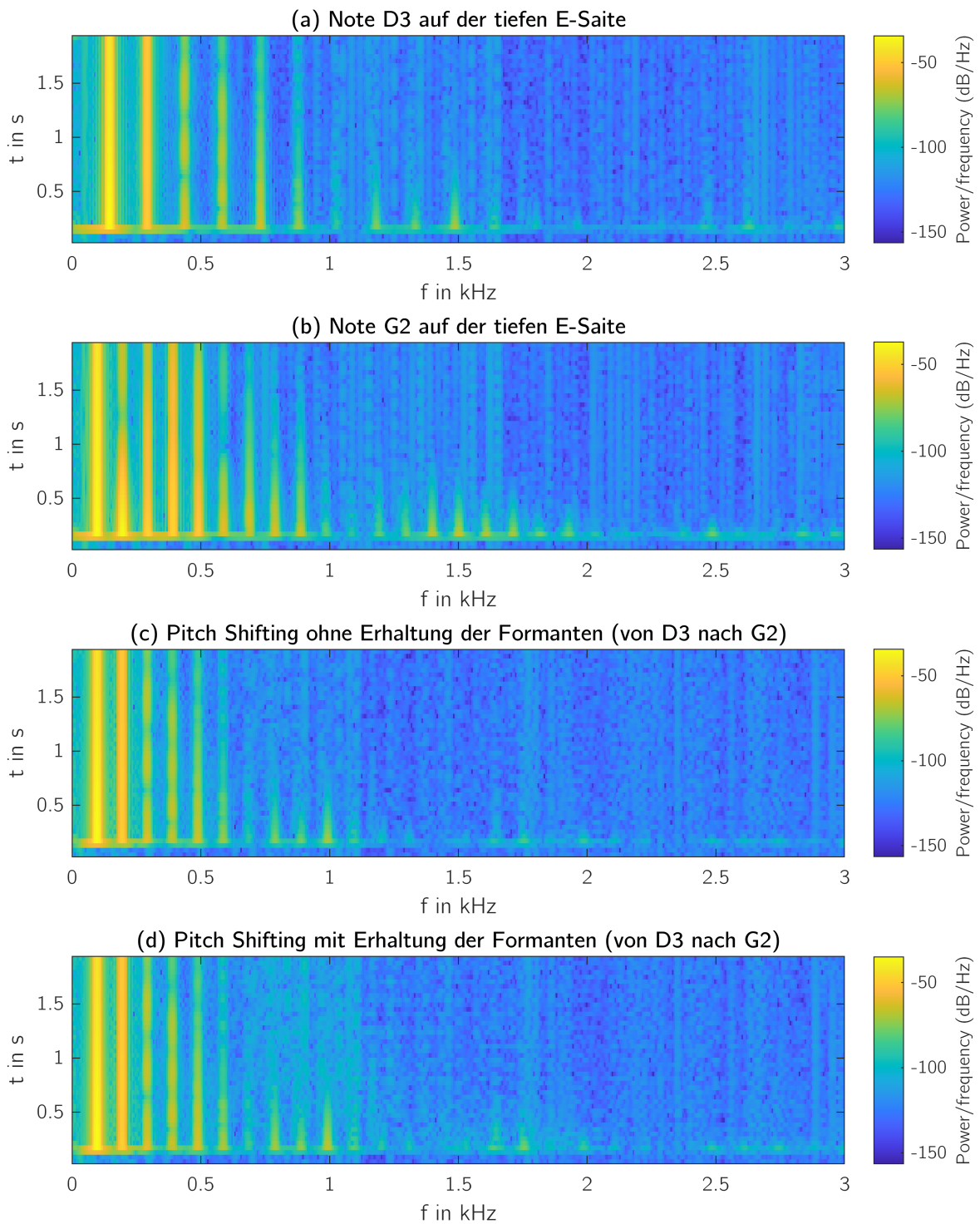


Abbildung 3.8: Pitch Shifting einer E-Gitarre um sieben Halbtöne nach unten

4 Algorithmen im Zeitbereich

In der Vergangenheit wurde eine Vielzahl an unterschiedlichen Algorithmen publiziert, die das Pitch Shifting bzw. Time Scaling von Signalen im Zeitbereich ermöglichen. Algorithmen im Zeitbereich basieren überwiegend auf der Overlap-Add Methode (beschrieben anhand der STFT in Unterabschnitt 2.2.4) und arbeiten dadurch grundsätzlich blockbasiert. Die Algorithmen unterscheiden sich vor allem darin wie die Signale für eine optimale Überlappung analysiert werden, damit bei der neuen Zusammensetzung des Signals beispielsweise möglichst keine Unstetigkeiten auftreten.

Algorithmen basierend auf der Overlap-Add Methode führen durch die Neuordnung des Signals zumeist nur Time Scaling nach dem Streckungsfaktor α durch. Erst die Kombination mit Resampling führt zum gewünschten Pitch Shifting (siehe Abbildung 3.2) [DM16, S. 21]. Der Streckungsfaktor wird an dieser Stelle als Verhältnis von Synthese-Schrittweite H_S zu Analyse-Schrittweite H_A definiert:

$$\alpha = \frac{H_S}{H_A} \quad (4.1)$$

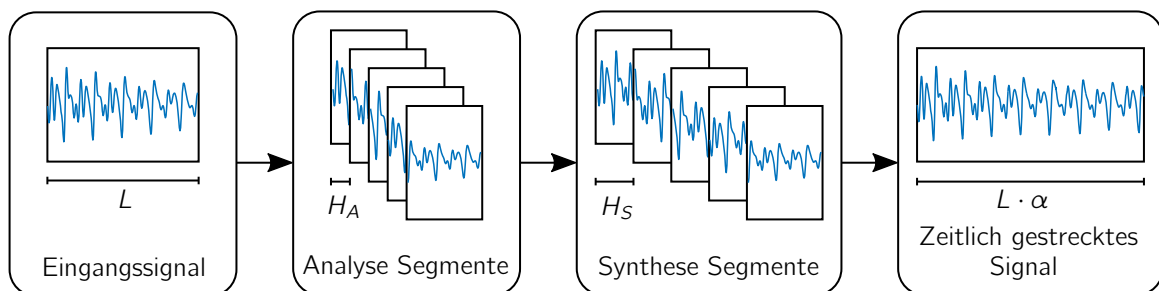


Abbildung 4.1: Grundsätzlicher Ablauf bei Time Scaling Algorithmen, die auf Overlap-Add Verfahren basieren (nach [DM16])

Um die Algorithmen besser miteinander vergleichen zu können, wird in allen Abbildungen ein Time Scaling bzw. Pitch Shifting mit $\alpha = 2$ durchgeführt. Dies entspricht einer Streckung des Signals bzw. Erhöhung der Tonhöhe.

An dieser Stelle sei erwähnt, dass es zahlreiche Algorithmen gibt, die auf einer Overlap-Add Technik basieren und sich teilweise nur in Details unterscheiden. Zu erwähnen ist beispielsweise das *Synchronous Overlap-Add Fixed Synthesis (SOLAFS)* Verfahren [HM91], welches im Gegensatz zum *Synchronous Overlap-Add (SOLA)* Verfahren die Synthese-Schrittweite festsetzt und damit einen ähnlichen Ansatz wie *Waveform Similarity Overlap-Add (WSOLA)* verfolgt. Beim *Pitch Synchronous Overlap-Add (PSOLA)* Verfahren wird im Signal zunächst nach Pitch Perioden gesucht, nach denen das Signal in entsprechende Blöcke zerlegt wird. Diese Blöcke werden anschließend in der Synthese gestaucht oder gestreckt wieder zusammengefügt. Das Pitch Shifting kann bei diesem Verfahren ohne das ansonsten notwendige Resampling erreicht werden [CM90], [McA13], [Zöl11, S. 205-209]. Ein weiterer Ansatz ist zudem *Global and Local Search Time Scale Modification (GLS-TSM)* [YP96].

Aufgrund der großen Anzahl an Variationen mit ähnlichem Ansatz werden in dieser Arbeit nur das OLA, SOLA und WSOLA Verfahren untersucht. Ein Vergleich von unterschiedlichen (auf Overlap-Add basierten) Ansätzen ist in [Dor05] und [DLC06] zu finden.

In Abschnitt 4.4 wird abschließend der sogenannte *Roller*-Algorithmus vorgestellt, der sich durch seinen samplebasierten Ansatz von allen anderen untersuchten Algorithmen unterscheidet.

In diesem Kapitel werden die Funktionsweise, Vor- und Nachteile sowie mögliche Artefakte der untersuchten Algorithmen erläutert. Eine detaillierte Bewertung und Gegenüberstellung aller Algorithmen im Zeit- und Frequenzbereich ist in Kapitel 6 zu finden.

4.1 Overlap-Add (OLA)

Das OLA Verfahren stellt die einfachste Möglichkeit dar, um ein Signal zeitlich zu strecken oder zu stauchen, wodurch es gleichzeitig das Verfahren mit dem geringsten Rechenaufwand ist. Das Verfahren wird beispielsweise von *Driedger und Müller* in [DM16] bzw. von *Driedger* in [Dri11] sowie von *Royer* in [Roy19] vorgestellt. Eine Visualisierung des Ablaufs des OLA Verfahrens ist der folgenden Abbildung 4.2 dargestellt.

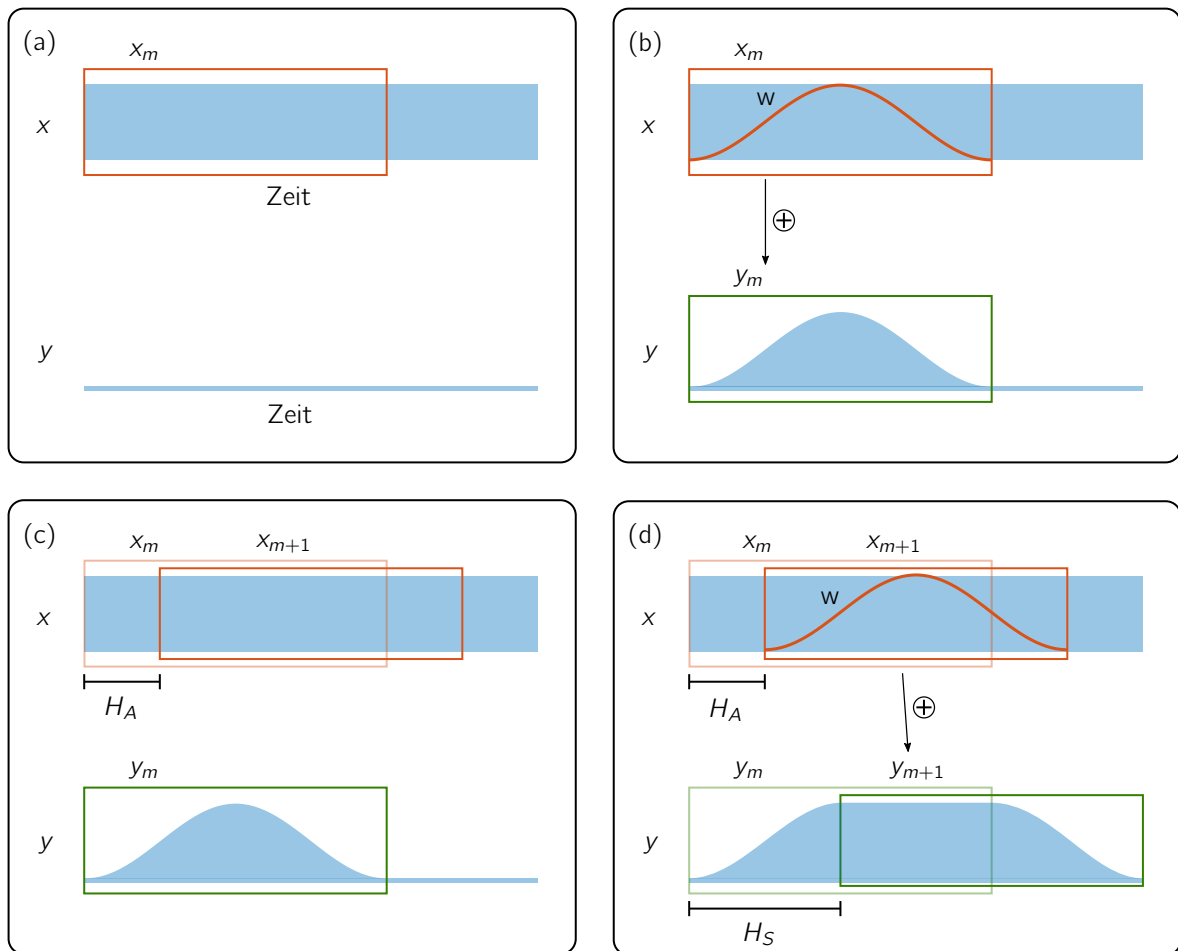


Abbildung 4.2: Time-Scaling mithilfe von OLA (nach [DM16, S. 5])

Beginnend bei (a) wird zunächst ein Block x_m der Länge N aus dem Signal herausgeschnitten und in (b) nach einer Fensterung (z. B. Von-Hann-Fenster) dem Ausgangssignal hinzuaddiert. In (c) wird nun der nächste Block x_{m+1} im Abstand der Analyse-Schrittweite H_A dem Eingangssignal entnommen. Nach einer Fensterung in (d) wird

dieser Block im Abstand der Synthese-Schrittweite H_S auf das Ausgangssignal überlappend hinzuaddiert.

Allgemein werden die ausgeschnittenen Segmente immer von festen Positionen im Eingangssignal (zeitliches Raster in Abstand H_A) an feste Positionen im Ausgangssignal (zeitliches Raster in Abstand H_S) kopiert. Die Struktur des Eingangssignals hat also keinen Einfluss auf das Verfahren. Meist wird H_S festgesetzt, um nach der COLA Bedingung ein konstantes Aufaddieren der Fenster zu garantieren und den Effekt einer Amplitudenmodulation zu verhindern. Der Streckungsfaktor α ergibt dann die notwendige Analyse-Schrittweite H_A .

Da durch die feste Analyse- und Synthese-Schrittweite die Struktur des Eingangssignals keinen Einfluss auf die Neuzusammensetzung des Signals hat, werden lokale periodische Strukturen nicht erhalten. Harmonische Eingangssignale werden deshalb stark verfremdet. Zudem treten durch die festen Parameter H_A und H_S starke Phasensprünge auf, da die Übergangsbereiche beim Zusammenfügen nicht passen (Warbling-Effekt). Folglich kommt dieser Algorithmus nicht in Frage, da die harmonische und polyphone Struktur von Gitarrensingen zu starken Artefakten führt.

Bei rein perkussiven Eingangssignalen (wie z. B. Schlagzeug) wird allerdings eine recht hohe Klangqualität erreicht, da solche Klänge kaum lokal periodische Strukturen aufweisen. Es ist allerdings darauf zu achten, dass die Blocklänge klein gehalten wird (ca. 10 ms), um das Duplizieren von Transienten zu reduzieren. Aufgrund dieser positiven Eigenschaft wurde von *Driedger* und *Müller* ein Verfahren vorgestellt, das ein Signal zunächst in einen harmonischen und einen perkussiven Teil zerlegt. Die perkussiven Anteile werden mittels OLA bearbeitet – die harmonischen Anteile hingegen mit einem Algorithmus, der gute Ergebnisse bei diesem Signaltyp erzielt, aber bei Transienten eventuell mit Verschmierungen reagiert [DM16; DM15; DM14a]. Bei Gitarrensingen liegt allerdings das Problem hauptsächlich in der latenzoptimierten Verarbeitung der harmonischen Strukturen, sodass der Ansatz der Harmonisch-Perkussiv-Zerlegung in dieser Arbeit nicht weiter verfolgt wird.

4.2 Synchronous Overlap-Add (SOLA)

Der Ablauf von OLA zeigt, dass die Neuzusammensetzung des Ausgangssignals in Abhängigkeit des Eingangssignals erfolgen muss, um eine möglichst gute Synchronisation der einzelnen Blöcke in den Überlappungsbereichen zu gewährleisten. Dazu wurde 1985 von *Roucos* und *Wilgus* in [RW85] das SOLA Verfahren vorgestellt, bei dem der Parameter H_S in der Synthese-Stufe flexibel gehalten wird, um den möglichst optimalen Überlappungspunkt zu treffen. Dadurch sollen, die im Signal enthaltenen, periodischen Strukturen möglichst erhalten bleiben. Die folgende Abbildung 4.3 zeigt den vom OLA Verfahren abgeleiteten und verfeinerten Ablauf. Weitere Beschreibungen des Algorithmus sind zudem in [Zöl11, S. 191] und [Mah01, S. 51] zu finden.

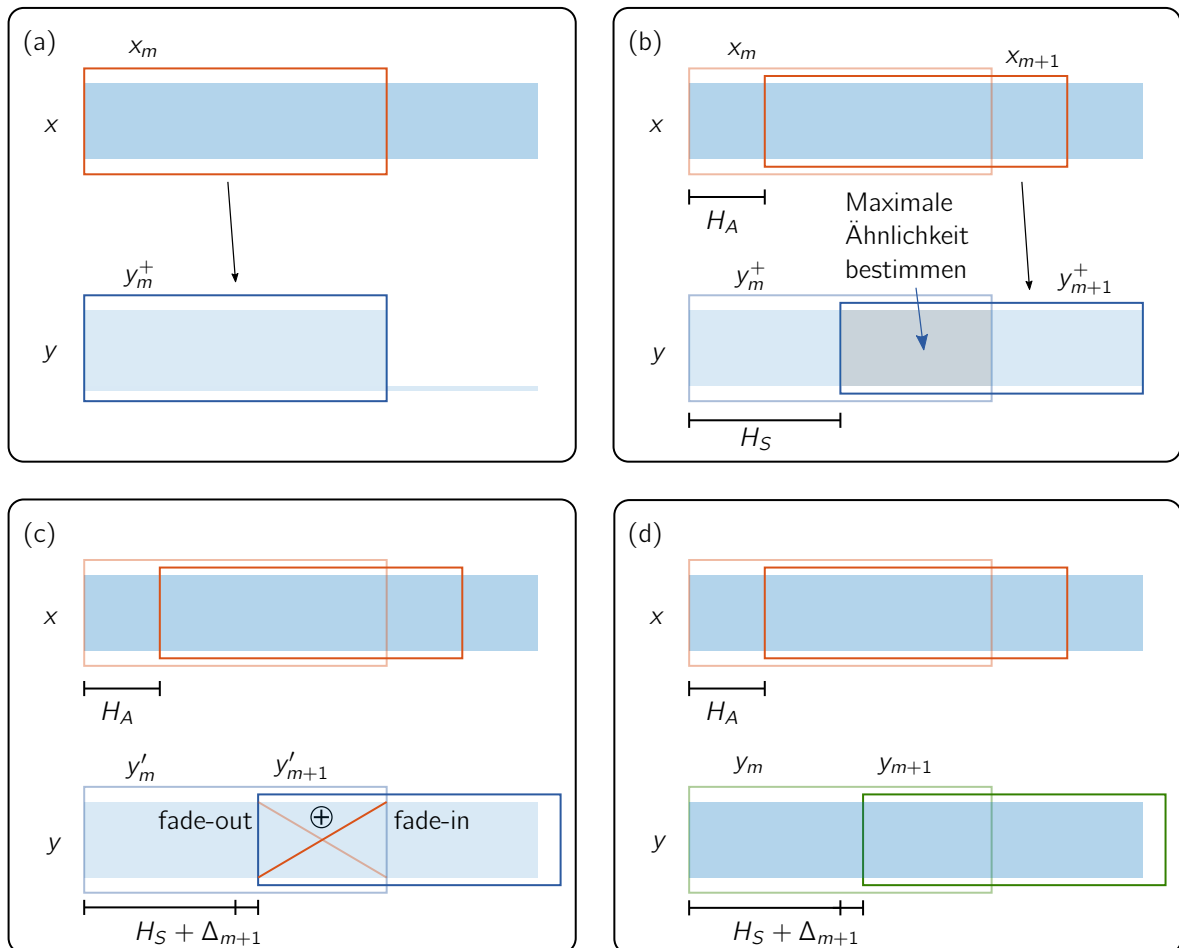


Abbildung 4.3: Time-Scaling mithilfe von SOLA

In (a) wird zunächst ein Block x_m aus dem Eingangssignal entnommen. Wie bereits beim OLA Verfahren wird in (b) der nächste Block x_{m+1} im Abstand H_A gebildet und im Abstand H_S überlappend angeordnet, aber noch nicht addiert. Um die periodischen Strukturen zu erhalten und Phasensprünge zu vermeiden, wird nun im grau gefärbten Überlappungsbereich von y_m^+ und y_{m+1}^+ die maximale Ähnlichkeit bestimmt. Dies wird typischerweise über die Berechnung der Kreuzkorrelation durchgeführt. In (c) zeigt sich nun, dass der optimale Überlappungspunkt um Δ_{m+1} verschoben liegt. Das endgültige Ausgangssignal ergibt sich durch die Aufsummierung von y'_m und y'_{m+1} , wobei im Überlappungsbereich nach dem Prinzip einer Fensterfunktion die Blöcke zunächst gegen null gewichtet werden. Die in (c) eingezeichnete Gewichtung führt so zu einer konstanten Aufsummierung im Überlappungsbereich.

Da der Korrelationsansatz in der Synthese-Stufe angewendet wird, wird an dieser Stelle von *Output Similarity Synchronization* gesprochen. Dies hat den Nachteil, dass der aus der Kreuzkorrelation bestimmte Parameter Δ_{m+1} direkten Einfluss auf die Signallänge von $y(n)$ hat. In dem oberen Beispiel ist das Ausgangssignal daher um Δ_{m+1} länger als gewünscht. Durch das ohnehin notwendige Resampling kann zwar die Signallänge wieder kompensiert werden, allerdings wird in diesem Fall nicht die korrekte Tonhöhe erreicht. Aus diesem Grund eignet sich das SOLA Verfahren grundsätzlich nicht für das in dieser Arbeit anvisierte Echtzeitsystem und wird daher nicht weiter verfolgt.

4.3 Waveform Similarity Overlap-Add (WSOLA)

Der WSOLA Ansatz stellt im Grunde eine Variation von SOLA dar und wurde von *Verhelst* und *Roelands* im Jahr 1993 in [VR93] vorgestellt. Weitere Untersuchungen sind in [Dri11, S. 21], [DM16], [RB19] sowie [Mah01, S. 55] zu finden.

Beim WSOLA-Ansatz wird bei der Positionierung der Blöcke ebenfalls eine zeitliche Toleranz zugelassen, um aufeinanderfolgende Blöcke möglichst optimal überlappen zu lassen. Ziel ist es, eine maximale Ähnlichkeit zwischen der synthetisierten und originalen Wellenform zu erzielen. Im Gegensatz zum SOLA Verfahren wird die Synthese-Schrittweite H_S festgesetzt und stattdessen die Analyse-Schrittweite H_A nach dem gewünschten Streckungsfaktor variiert. Mit einem Von-Hann-Fenster und einem Überlappungsgrad von 50 % kann so beispielsweise die COLA-Bedingung erfüllt werden. Die zeitliche Toleranz Δ_{max} wird in der Analyse-Stufe gewährt und ermöglicht eine flexiblere und bessere Positionierung der Analyse-Fenster. Die folgende Abbildung 4.4 veranschaulicht den Ablauf des Algorithmus.

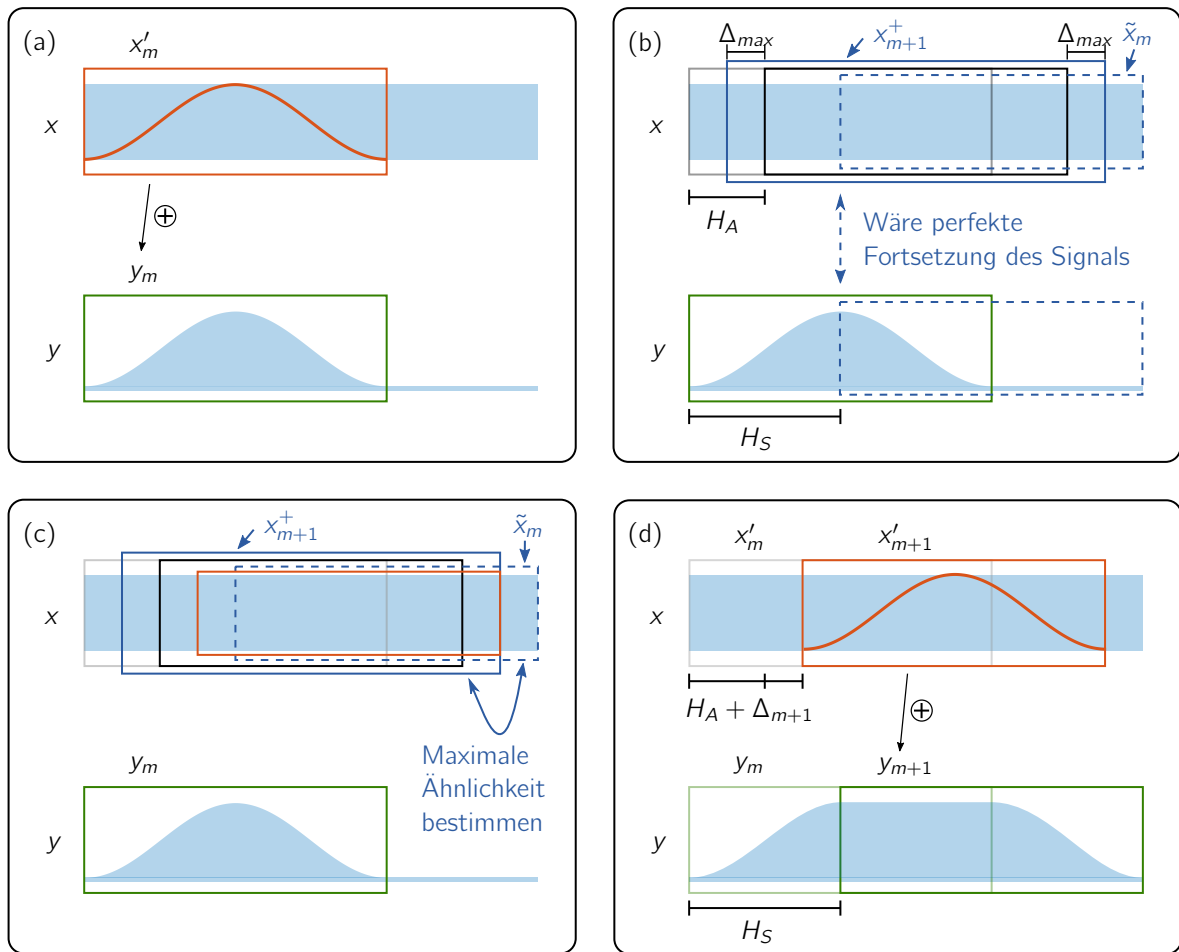


Abbildung 4.4: Time-Scaling mithilfe von WSOLA (nach [DM16, S. 7])

In (a) wird zunächst ein Block x'_m aus dem Eingangssignal entnommen, gefenstert und zum Ausgangssignal hinzuaddiert. x'_m bezeichnet dabei einen Block, dessen optimale zeitliche Position bereits bestimmt wurde. In (b) wird nun im Abstand H_A der nächste Block der Länge N anvisiert, wobei links und rechts zusätzlich eine Toleranz von Δ_{max} zugelassen wird. Dieser um $2\Delta_{max}$ größere Auswahlbereich wird hier als x_{m+1}^+ bezeichnet. Zusätzlich wird im Eingangssignal im Abstand H_S ein Block \tilde{x} der Länge N gebildet, der aus Sicht des bisher synthetisierten Signals eine perfekte Fortsetzung des Signals bedeuten würde. Darauf aufbauend werden in (c) diese beiden Blöcke mit einander verglichen und deren maximale Ähnlichkeit bestimmt. Dies wird typischerweise über eine Kreuzkorrelation durchgeführt. Das Ergebnis dieser Kreuzkorrelation führt dann zu dem um Δ_{m+1} verschobenen und rot eingezeichneten Block x'_{m+1} , der anschließend nach dem Overlap-Add Prinzip im Abstand H_S auf das Ausgangssignal addiert wird.

Im Gegensatz zum SOLA Verfahren wird bei WSOLA die Korrelation am Eingangssignal

durchgeführt, weshalb dieser Ansatz als *Input Similarity Synchronization* eingeordnet wird. Die Synchronisation zeigt sich durch die Ausrichtung der periodischen Strukturen im Überlappungsbereich, sodass Phasensprünge vermieden werden. WSOLA kann zudem als Echtzeitsystem realisiert werden, da die Signallänge des synthetisierten Signals durch die feste Synthese-Schrittweite exakt dem Streckungsfaktor folgt.

Ein bekanntes Problem ist allerdings, dass die korrekte Darstellung von Transienten nicht immer sichergestellt ist. Bei allen auf Overlap-Add basierten Ansätze kann es bei der Streckung des Signals ($\alpha > 1$) zur Duplizierung oder bei der Stauchung des Signals ($\alpha < 1$) zum Überspringen von Transienten kommen. Abhilfe kann eine vorangestellte Analyse zur Lokalisierung aller Transienten schaffen, wie sie beispielsweise in [GL08] vorgeschlagen wird. Mithilfe eines Transienten-Detektors werden zunächst die zeitlichen Positionen der Transienten erkannt. Befindet sich im nachfolgenden WSOLA-Ablauf ein Analyse-Fenster in der unmittelbaren Umgebung einer Transiente, wird H_A gleich H_S gesetzt, sodass dieser Bereich unverändert zum Ausgangssignal kopiert wird.

Darüber hinaus wird beim WSOLA Verfahren hauptsächlich die stärkste periodische Struktur durch den Vergleich der Wellenformen berücksichtigt und das Δ_{m+1} entsprechend gewählt. Bei stark polyphonen Signalen kann es trotzdem zum Warbling-Effekt kommen, da es bei den weniger dominanten Frequenzanteilen zu Phasensprünge kommt. In Kapitel 6 wird dieses Problem anhand von unterschiedlich komplexen Testsignalen aufgezeigt.

4.4 Roller Algorithmus

Der Roller Algorithmus wurde 2008 von *Juillerat et al.* in [JAS08] vorgestellt und arbeitet ebenfalls komplett im Zeitbereich. Im Gegensatz zu den bisher vorgestellten Algorithmen wird kein Overlap-Ansatz verwendet, sondern auf IIR-Filter, Oszillatoren und Ringmodulatoren zurückgegriffen. Der Roller Algorithmus stellt in dieser Arbeit den einzigen samplebasierten Ansatz dar. Alle anderen Algorithmen arbeiten dagegen blockbasiert.

Das Eingangssignal wird zunächst über eine große IIR Filterbank in B Frequenzbänder zerlegt. Jedes einzelne Frequenzband wird anschließend mithilfe eines *Frequency Shifters (FS)* um eine jeweils individuelle Frequenz f_b^{shift} verschoben. Wie bereits mit Gleichung 3.5 erläutert, sind Frequency Scaling (= Pitch Shifting) und Frequency Shifting zwei unterschiedliche Operationen. Die grundlegende Idee besteht darin, dass durch die unterschiedlichen Frequenzverschiebungen der Frequenzbänder global eine Tonhöhenverschiebung approximiert wird. Das Ausgangssignal ergibt sich dann durch eine Aufsummierung aller in der Frequenz verschobenen Frequenzbänder. Die folgende Abbildung 4.5 visualisiert diesen Ablauf in Form eines Blockschaltendiagramms.

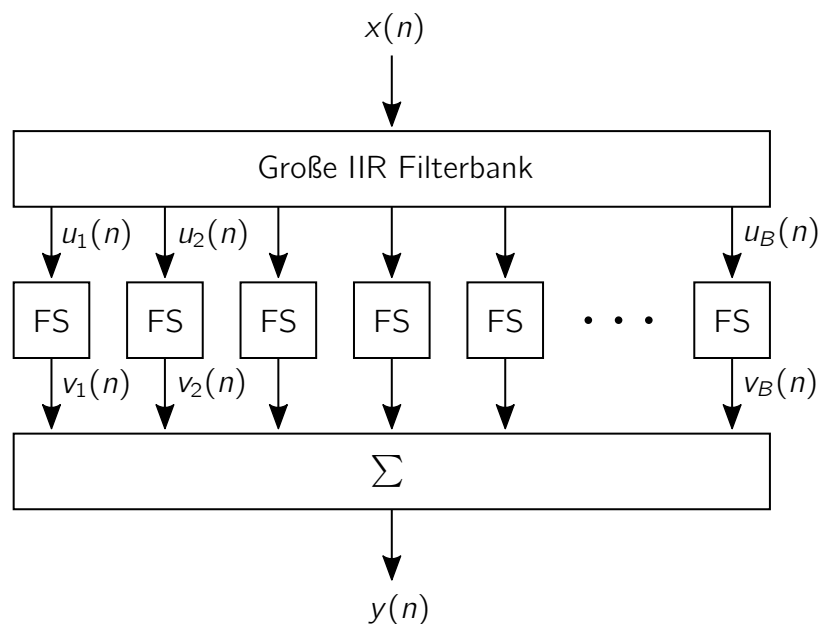


Abbildung 4.5: Roller Algorithmus (nach [JAS08])
FS bezeichnet einen Frequency Shifter

Um Pitch Shifting mit vielen einzelnen Frequenzverschiebungen zu approximieren, muss jedes einzelne Frequenzband in der Frequenz skaliert werden. Für jedes Frequenzband mit der Mittenfrequenz f_b^c ergibt sich nach dem gewünschten Streckungsfaktor α die notwendige Frequenzverschiebung f_b^{shift} :

$$f_b^{shift} = (f_b^c \cdot \alpha) - f_b^c \quad (4.2)$$

Die Frequenzverschiebung f_b^{shift} ist also für $\alpha < 1$ negativ und für $\alpha > 1$ positiv.

4.4.1 Frequenzverschiebung mittels Single Sideband (SSB) Modulation

Dieser Unterabschnitt basiert auf den Erkenntnissen von [KS12b; War98; Mar99; JAS08].

Zunächst wird eine *Double Sideband (DSB)* Modulation, die eine Variante einer Amplitudenmodulation mit unterdrücktem Träger darstellt, betrachtet. Mathematisch wird die DSB Modulation durch eine einfache Multiplikation eines reellen Oszillator-Signals, dem Trägersignal, mit dem zu modulierenden Signal ausgedrückt, welches in der folgenden Abbildung 4.6 als Signalfussdiagramm dargestellt ist.

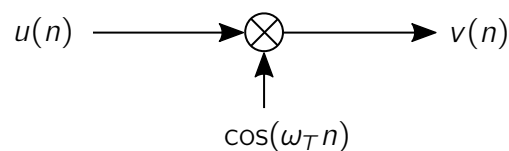


Abbildung 4.6: DSB Modulation durch Multiplikation eines reellen Signals $u(n)$ mit einem reellen Oszillator-Signal (nach [KS12a])

Die Abbildung 4.7 zeigt die spektrale Wirkung dieser Modulation. Ein reelles Signal $u(n)$ besitzt immer ein symmetrisches Amplitudenspektrum, welches in ein oberes und unteres Seitenband eingeteilt werden kann. Beide Seitenbänder enthalten aber letztlich dieselbe Information. Die Multiplikation mit dem reellen Oszillator-Signal mit der Trägerfrequenz ω_T führt dazu, dass sowohl das obere als auch untere Seitenband um die Trägerfrequenz liegt. Aus Sicht der Nachrichtentechnik bedeutet die in beiden Seitenbändern redundante Information bei gegebener Sendeleistung eine schlechtere Reichweite und ein schlechteres Signal-Rausch-Verhältnis. Die hier gewünschte Frequenzverschiebung lässt sich so nicht realisieren, da am Ausgang die Summen- und Differenzfrequenzen aus der/den Eingangsfrequenz(en) und der Trägerfrequenz entstehen.

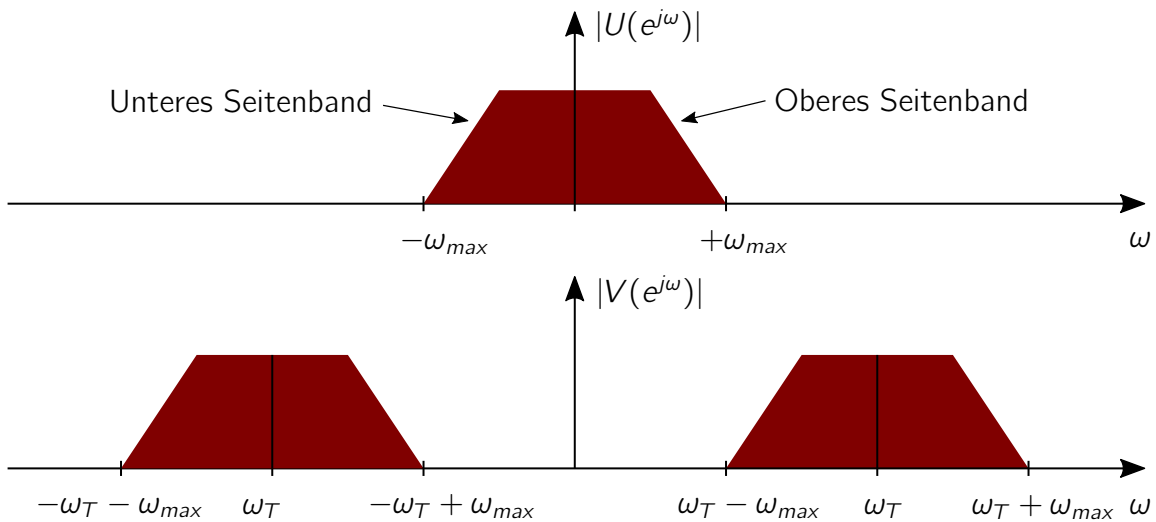


Abbildung 4.7: Spektrale Wirkung einer DSB Modulation (nach [War98])

Bei der *Single Sideband (SSB)* Modulation (auch I/Q Modulation oder zu deutsch Einseitenbandmodulation genannt) wird hingegen das aus $u(n)$ gebildete analytische Signal $\underline{u}(n)$ mit einem komplexen Oszillator-Signal multipliziert. Dies ist in der nachstehenden Abbildung 4.8 als Signalflussdiagramm dargestellt.

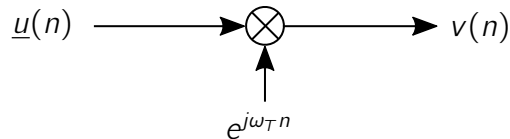


Abbildung 4.8: SSB Modulation durch Multiplikation eines analytischen Signals $\underline{u}(n)$ mit einem komplexen Oszillator-Signal (nach [KS12a])

Das analytische Signal $\underline{u}(n)$ enthält lediglich ein Seitenband und wird als komplexe Zahl mit I- (In-Phase) und Q-Pfad (Quadratur) dargestellt:

$$\underline{u}(n) = u_I(n) + ju_Q(n) \quad (4.3)$$

Ein solches Signal kann beispielsweise durch die Phasenmethode mithilfe eines Hilbert-Transformators erzeugt werden. Im Q-Pfad befindet sich dabei ein Hilbert-Filter, welches dem Eingangssignal eine 90° Phasenverschiebung hinzufügt. Aus einem Kosinus wird so beispielsweise ein Sinus.

Für eine Implementierung auf Hardware bietet sich die reellwertige Darstellung der SSB Modulation an, die nach der eulerschen Formel hergeleitet wird. H_H bezeichnet dabei

die Übertragungsfunktion des Hilbert-Filters:

$$v(n) = \Re\left\{ \underline{u}(n) \cdot e^{j\omega_T n} \right\} \quad (4.4)$$

$$\begin{aligned} &= \Re\left\{ [u(n) + jH_H\{u(n)\}] \cdot e^{j\omega_T n} \right\} \\ &= \Re\left\{ [u(n) + jH_H\{u(n)\}] \cdot [\cos(\omega_T n) + j \sin(\omega_T n)] \right\} \\ &= u(n) \cdot \cos(\omega_T n) - H_H\{u(n)\} \cdot \sin(\omega_T n) \end{aligned} \quad (4.5)$$

Die folgende Abbildung 4.9 stellt Gleichung 4.5 als Signalflussdiagramm dar:

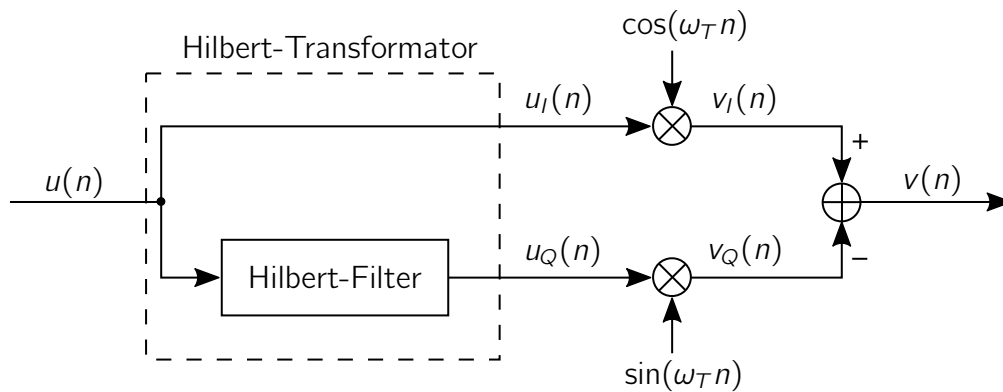


Abbildung 4.9: Reelles Signalflussdiagramm einer SSB Modulation (nach [War98])

Ist das Eingangssignal beispielsweise $u(n) = \cos(\omega_1 n)$ ergibt sich am Ausgang ein um ω_T verschobener Kosinus:

$$\begin{aligned} v_I(n) - v_Q(n) &= \left[\cos(\omega_T n) \cdot \cos(\omega_1 n) \right] - \left[\sin(\omega_T n) \cdot H_H\{\cos(\omega_1 n)\} \right] \\ &= \left[\cos(\omega_T n) \cdot \cos(\omega_1 n) \right] - \left[\sin(\omega_T n) \cdot \sin(\omega_1 n) \right] \\ &= \left[\frac{1}{2} \cos((\omega_T - \omega_1)n) + \frac{1}{2} \cos((\omega_T + \omega_1)n) \right] \\ &\quad - \left[\frac{1}{2} \cos((\omega_T - \omega_1)n) - \frac{1}{2} \cos((\omega_T + \omega_1)n) \right] \\ &= \cos((\omega_T + \omega_1)n) \end{aligned} \quad (4.6)$$

Besteht das Eingangssignal aus mehreren Sinusoide mit verschiedenen Frequenzen, entstehen durch die SSB Modulation keine Intermodulationsprodukte. Jeder Sinusoid wird auf der Frequenzachse um ω_T verschoben. Das Vorzeichen von $v_Q(n)$ oder alternativ

das Vorzeichen von ω_T entscheidet letztlich über die Richtung der Frequenzverschiebung. Die folgende Abbildung 4.10 fasst die spektrale Wirkung der SSB Modulation nochmal zusammen.

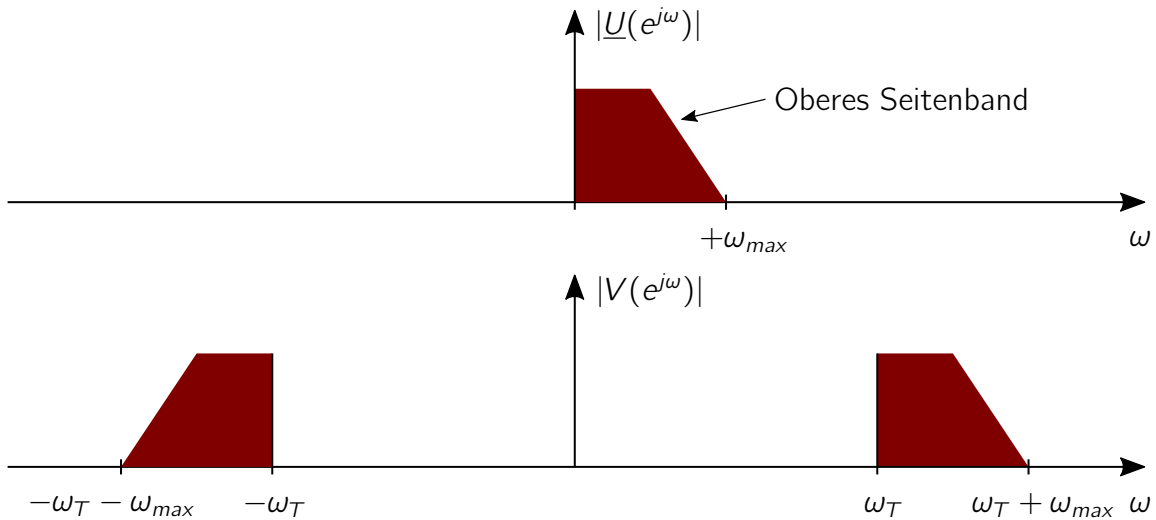


Abbildung 4.10: Spektrale Wirkung einer SSB Modulation (nach [War98])

Realisierung des Hilbert-Transformators

Für die praktische Realisierung des Hilbert-Transformators existieren verschiedene Ansätze, von denen die folgenden drei Möglichkeiten auf ihre Eignung für den Roller Algorithmus untersucht werden:

- FIR Hilbert-Filter
- FFT bzw. STFT und negative Frequenzen auf null setzen
- IIR Brückenfilter mit Allpass-Biquad Sektionen

Eine häufig verwendete Methode ist die Approximation eines idealen Hilbert-Transformators mit einem *Finite Impulse Response (FIR)* Hilbert-Filter der Ordnung N . Da ein (linearphasiges) FIR Filter dem Signal eine Verzögerung von $N/2$ hinzufügt, muss der I-Pfad ebenfalls um $N/2$ verzögert werden, da ansonsten nicht die korrekten Samples aufsummiert werden. Die folgende Abbildung 4.11 zeigt die Realisierung eines solchen Hilbert-Transformators mit einem FIR Hilbert-Filter im Q-Pfad als Teil der SSB Modulation.

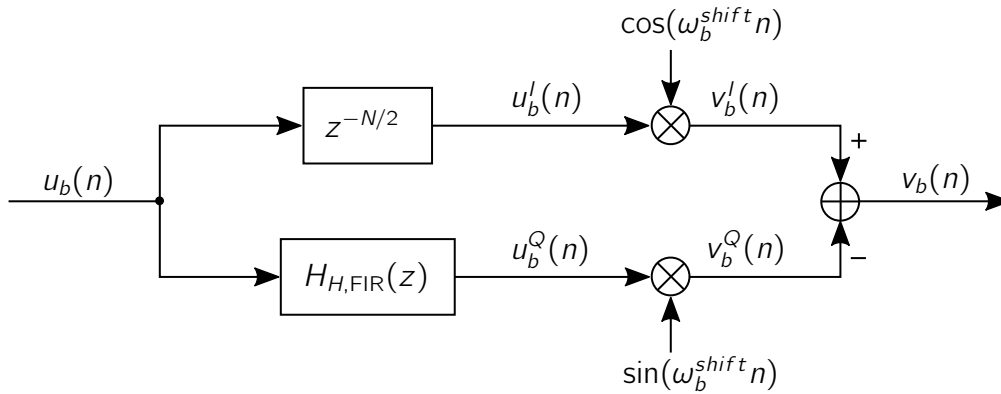


Abbildung 4.11: SSB Modulation mithilfe eines FIR Hilbert-Filters der Ordnung N

Der Entwurf des FIR Hilbert-Filters wird mithilfe von MATLAB[®] und dem *Parks-McClellan* Algorithmus nach den folgenden Spezifikationen durchgeführt:

- Eckfrequenz 80 Hz bzw. Übergangsbereich 160 Hz
- Sperrdämpfung 60 dB

Die notwendige Eckfrequenz ergibt sich aus den Untersuchungen der spektralen Bandbreite einer E-Gitarre in Abschnitt 2.1 durch die tiefe E-Saite ($f_{E2} = 82.41$ Hz). Die vorgegebene Spezifikation führt zu einem FIR Filter mit der Filterordnung $N = 978$. Da dieses Filter in jedem Frequenzband benötigt wird, ergibt sich dadurch ein massiver Rechenaufwand, der eine Implementierung in Echtzeit erschwert. Darüber hinaus führt die Verzögerung des Filters von $N/2$ bei einer Abtastrate von 48 kHz bereits zu einer Latenz von ca. 10.2 ms.

Eine weitere Möglichkeit besteht darin das analytische Signal im Frequenzbereich durch das Nullsetzen der negativen Frequenzkomponenten zu approximieren [Mar99]. $U_b(k)$ bezeichnet dabei die Fourier-Transformierte des b -ten Frequenzbandes und $\underline{U}_b(k)$ die Fourier-Transformierte des berechneten analytischen Signals:

$$\underline{U}_b(k) = \begin{cases} U_b(0) & \text{für } k = 0 \\ 2U_b(k) & \text{für } 1 \leq k \leq \frac{N}{2} - 1 \\ U_b(\frac{N}{2}) & \text{für } k = \frac{N}{2} \\ 0 & \text{für } \frac{N}{2} + 1 \leq k \leq N - 1 \end{cases} \quad (4.7)$$

Da das berechnete Spektrum periodisch mit der Abtastfrequenz ist, wird hier nur der Bereich von $[0, F_s)$ betrachtet. Durch die Periodizität wiederholt sich das manipulierte Spektrum ebenfalls im negativen Frequenzbereich, sodass dieser nicht vollständig verschwindet. Die wiederholenden Spektren weisen jedoch für sich genommen die nach

Gleichung 4.7 vorgenommene Manipulation auf. Im Rahmen eines Echtzeitsystems wird die Transformation in den Frequenzbereich und zurück in den Zeitbereich durch die STFT bzw. ISTFT durchgeführt. Die folgende Abbildung 4.12 zeigt die Realisierung in Verbindung mit der SSB Modulation.

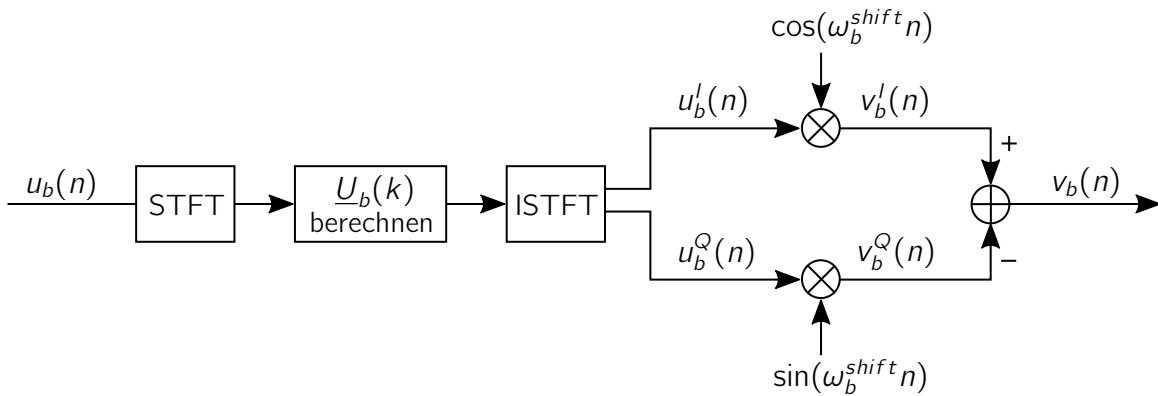


Abbildung 4.12: SSB Modulation durch spektrale Manipulation nach Gleichung 4.7

Entscheidend ist an dieser Stelle die Blocklänge N der STFT sowie die verwendete Fensterfunktion, um den Leck-Effekt zu kontrollieren. Dieser wirkt sich insbesondere auf die Unterdrückung der negativen Frequenzanteile im tieffrequenten Bereich aus. Ist die Blocklänge zu klein, werden die negativen Frequenzanteile im Übergangsbereich um 0 Hz, aufgrund der breiten Hauptkeule, schlecht unterdrückt. Jede Abweichung vom idealen Hilbert-Transformator führt dazu, dass die SSB Modulation zunehmend in eine gewöhnliche DSB Modulation übergeht. Dadurch werden dem Signal $v_b(n)$ falsche Frequenzanteile hinzugefügt, welche die Approximation des Pitch Shiftings, durch die Zerstörung der harmonischen Zusammenhänge, verschlechtern. Mit dem Ansatz des Nullsetzens der negativen Frequenzen kann eine geringere Blocklänge und damit eine geringe Latenz realisiert werden. Allerdings werden für jeden Zeitabschnitt m insgesamt b FFTs benötigt. Dies stellt ebenfalls einen großen Rechenaufwand dar.

Eine weitere Möglichkeit ist die Approximation des Hilbert-Transformators über ein IIR Filter, wie sie auch von *Juillerat et al.* in [JAS08] und *Wardle* in [War98] vorgeschlagen wird. In diesem Fall wird ein Brückenfilter aus Allpässen verwendet, die in Form von Biquad-Sektionen implementiert werden. Dieser Aufbau ist in Abbildung 4.13 aufgezeigt.

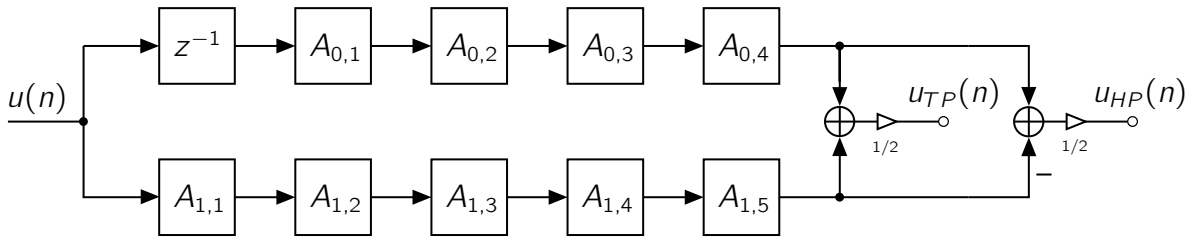


Abbildung 4.13: IIR Brückenfilter aus Biquad-Allpass-Sektionen

Jedes Allpassfilter besitzt die folgende (allgemeine) Übertragungsfunktion:

$$H(z) = \frac{b_0 + z^{-2}}{1 + b_0 z^{-2}} \quad (4.8)$$

Das Programm zum Entwurf eines solchen Brückenfilters wurde von Herrn Prof. Sauvagerd zur Verfügung gestellt. Über die unterschiedliche Zusammenschaltung der Ausgänge wird zunächst ein Tiefpass- bzw. Hochpassfilter realisiert, dessen Übergangsbereich wie bei einem FIR Halbbandfilter bei $F_s/4$ liegt. Die folgende Transformation führt dann eine Verschiebung um $F_s/4$ auf der Frequenzachse durch, um den gewünschten Hilbert-Transformator aus dem vorherigen Tiefpassfilter-Ausgang zu erhalten:

$$z \rightarrow jz \quad (4.9)$$

Die Filterspezifikationen sind grundsätzlich mit den Spezifikationen des oben entworfenen FIR Filters identisch. Da durch das Programm aber zunächst ein Tiefpassfilter um $F_s/4$ entworfen wird, wird als Eckfrequenz dementsprechend $F_s/4 + 80 \text{ Hz} = 12.08 \text{ kHz}$ (bei $F_s = 48 \text{ kHz}$) anvisiert. Der Entwurf des IIR Filters zeigt, dass dieselben Spezifikationen bereits mit einer Filterordnung von 19 erfüllt werden. Durch den zusätzlich einfachen Aufbau der Übertragungsfunktion eines einzelnen Allpasses, ergibt sich (insbesondere bezogen auf die starke Filtereigenschaft des Systems) ein akzeptabler Rechenaufwand, sodass das Filter für den Einsatz im Roller Algorithmus geeignet ist. Die folgende Abbildung 4.14 zeigt abschließend noch einmal den IIR Hilbert-Transformator in Verbindung mit der SSB Modulation.

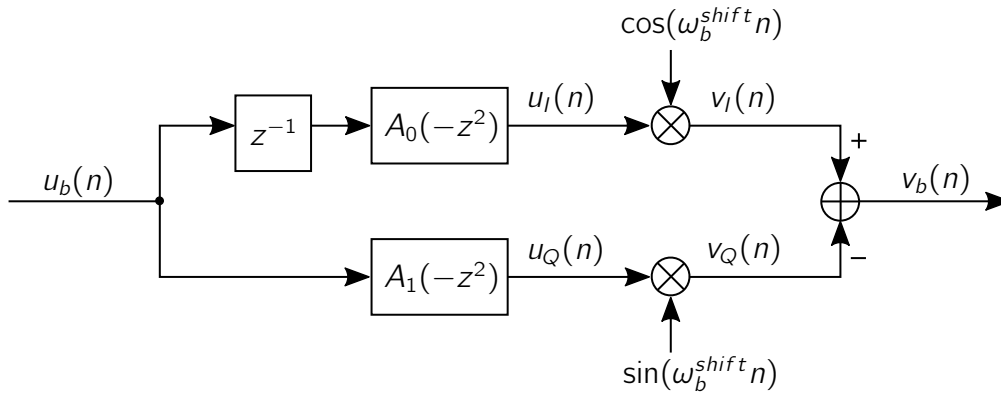


Abbildung 4.14: SSB Modulation mithilfe eines IIR Allpass-Brückenfilters

4.4.2 Filterbank-Design

Die Filterbank wird mit Bandpassfiltern realisiert, die das Eingangssignal in einzelne Frequenzbänder b zerlegen. Folgende Gedanken spielen bei dem Design der Filterbank eine Rolle:

- Platzierung der Bandpässe auf der Frequenzachse
- Größe der Filterbank
- Filtergrad und Übergangsbereiche von benachbarten Bandpässen

Da die musikalischen Noten logarithmisch über die Frequenzachse verteilt sind, bietet sich ebenfalls eine logarithmische Verteilung der Bandpässe an, die mindestens alle Noten abdecken. Die Mittenfrequenzen der Bandpässe ergeben sich durch:

$$f_b^c = f_{min} \cdot 2^{b/N_B} \quad \text{mit } b = 0, 1, \dots, B-1 \quad (4.10)$$

N_B bezeichnet die Anzahl an Frequenzbänder pro Oktave und sollte mindestens $N_B = 12$ betragen, da so auf exakt jede Note ein Bandpass fällt. Die Anzahl an Bandpassfilter für den Frequenzbereich f_{min} bis f_{max} berechnet sich durch:

$$B = \text{floor} \left[N_B \cdot \log_2 \left(\frac{f_{max}}{f_{min}} \right) \right] + 1 \quad (4.11)$$

Für den Frequenzbereich $f_{E2} = 82.41$ Hz bis 20 kHz werden bei 12 Filtern pro Oktave insgesamt 96 Bandpässe benötigt.

Die Größe der Filterbank (Anzahl an Frequenzbänder) ist ebenfalls ein wichtiger Parameter. Das Pitch Shifting wird über mehrfaches Frequency Shifting approximiert. Daher gilt, je größer die Filterbank, desto akkurater wird das Pitch Shifting. Gleichzeitig geht jedoch die Größe der Filterbank direkt mit Rechenaufwand einher, sodass ein sinnvoller Kompromiss gefunden werden muss. Vorteilhaft ist, dass durch die SSB Modulation keine Intermodulationsprodukte entstehen, wenn mehrere Frequenzanteile (z. B. Sinusoide) durch einen Bandpass erfasst werden. Jeder Sinusoid wird um die gleiche Frequenz f_b^{shift} verschoben. Problematisch sind Sinusoide, die nicht exakt die Mittenfrequenz des Frequenzbands treffen, da die Frequenzverschiebung nicht vom Signal, sondern lediglich von der Mittenfrequenz abhängt. Ein praktisches Beispiel wäre hier das Hochziehen einer Saite (auch Bending genannt), womit letztlich jede denkbare Frequenz zwischen f_b^c und f_{b+1}^c erzeugt werden kann. Daraus folgt, dass die Frequenzverschiebung für diesen Ton etwas zu gering ist, was zu einem leicht verstimmtten Klang führt (Detuning). Laut *Juillerat et al.* wird bei einer Filterbankgröße von ca. $B = 100$ eine eher niedrige und bei ca. $B = 200$ gute Qualität erreicht.

An die Bandpassfilter wird im tieffrequenten Bereich eine hohe Anforderung gestellt, da durch die logarithmische Verteilung der Noten die Mittenfrequenzen der Frequenzbänder dort sehr nah aneinander liegen. Es muss Sorge getragen werden, dass ein zum Frequenzband f_b^c zugehöriger Sinusoid nicht zusätzlich durch ein Nachbarfrequenzband erfasst wird, da ansonsten der Sinusoid mehrfach (wenn auch mit geringerer Amplitude) in der Frequenz verschoben wird. Die daraus resultierende hohe Flankensteilheit führt schließlich zu Filtern mit einer hohen Ordnung. Um die Filterordnung und damit den Rechenaufwand klein zu halten, schlägt *Juillerat et al.* die Verwendung von IIR Filtern vor. Allerdings können mit IIR Filtern im tieffrequenten Bereich nicht die gewünschten Spezifikationen erfüllt werden. Ein schmaler Durchlassbereich führt bei IIR Filtern in diesem Frequenzbereich zu einer hohen Gruppenlaufzeit und damit zu einer ebenfalls hohen Latenz. Eine hohe Flankensteilheit steigert die Gruppenlaufzeit darüber hinaus zusätzlich.

All diese Einschränkungen führen dazu, dass bei der Filterbank nicht eine perfekte Rekonstruktion angestrebt wird, sondern stattdessen psychoakustische Aspekte gezielt ausgenutzt werden, um die Latenz zu optimieren. Um die Gruppenlaufzeit im tieffrequenten Bereich niedrig zu halten, sollte die Breite des Durchlassbereichs eines Bandpasses laut *Juillerat et al.* mindestens 50 Hz betragen. Dieser Kompromiss führt im tieffrequenten Bereich zwangsläufig zu starken Überlappungen und einem verstimmtten Klang. In den höheren Frequenzbereichen werden zudem gezielt Lücken zwischen benachbarten Basspässen zugelassen, um die Flankensteilheit und damit die Filterordnung niedrig zu halten. *Juillerat et al.* schlagen Butterworth Bandpassfilter vierter Ordnung mit einem Übergang bei -12 dB vor. Dies führt zu einem Informationsverlust von ca. 35 %.

Aufbau einer FIR Filterbank

Um den Aufbau und die Funktionsweise des Roller Algorithmus zu testen, wird zunächst auf linearphasige FIR Filter mit hoher Ordnung zurückgegriffen. Das Ziel ist zunächst nicht die Latenz- sondern die Klangoptimierung, um die Grenzen des Algorithmus zu erfassen. Die FIR Filterbank dient später als klangliche Referenz.

Die Bandpässe werden mit den folgenden Spezifikationen entworfen:

- linke Sperrdämpfung $A_b^{stop1} = 40$ dB
- rechte Sperrdämpfung $A_b^{stop2} = 40$ dB
- linke Sperrfrequenz $f_b^{stop1} = f_b^c \cdot 2^{\frac{-1}{N_B}}$
- rechte Sperrfrequenz $f_b^{stop2} = f_b^c \cdot 2^{\frac{1}{N_B}}$
- linke Eckfrequenz $f_b^{pass1} = f_b^c - (f_b^c - f_b^{stop1}) / 100$
- rechte Eckfrequenz $f_b^{pass2} = f_b^c - (f_b^{stop2} - f_b^c) / 100$

Die Sperrdämpfung wird somit bei der Mittenfrequenz des jeweiligen Nachbarfrequenzbandes erreicht. Der Durchlassbereich wird durch die Wahl der Eckfrequenzen schmal gewählt, um die Filterordnungen der Bandpässe gering zu halten. Der Entwurf wird in MATLAB[®] mittels Fenstermethode (Kaiser-Fenster) realisiert, da diese Entwurfsmethode zuverlässig und verhältnismäßig schnell ist. Der Entwurf mithilfe des *Least-Square* oder *Parks-McClellan* Verfahrens eignet sich hier nicht, da die Filteranforderungen für die Algorithmen zu hoch sind. Die folgende Abbildung 4.15 zeigt die entworfene Filterbank mit insgesamt 96 Bandpässen ($N_B = 12$). Der Übergangsbereich zwischen benachbarten Frequenzbändern liegt bei ca. -6 dB.

Aufgrund der logarithmischen Verteilung ist die benötigte Filterordnung für das tiefste Frequenzband am höchsten und für das höchste Frequenzband am niedrigsten. Das Signal wird dadurch in jedem Frequenzband unterschiedlich verzögert, sodass die Signalverzögerung in jedem Frequenzband vor der Aufsummierung der einzelnen Signale v_b durch entsprechende Verzögerungsglieder angeglichen wird.

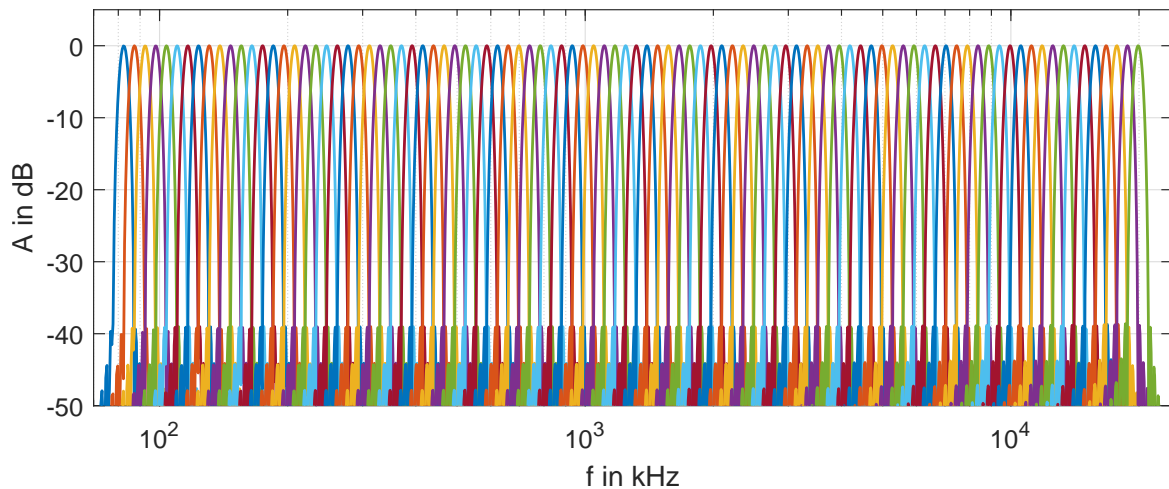


Abbildung 4.15: FIR Filterbank: Amplitudengänge aller Bandpassfilter

Das Design der Filterbank führt, durch die hohe Filterordnung des tiefsten Bandpassfilters, zu einer hohen Latenz. Die erzielte Klangqualität ist jedoch recht hoch. Zum einen sorgt die hohe Flankensteilheit der einzelnen Filter dafür, dass keine Verstimmungen auftreten, zum anderen klingen auch stark polyphone Signale (z. B. Vierklänge) sauber aus. Bei den zuvor betrachteten, auf Overlap-Add basierten, Verfahren treten dabei Phasensprünge auf, da die periodischen Strukturen nicht mehr korrekt ausgerichtet werden. Einzig bei Signalabschnitten mit starken Transienten zeigen sich Artefakte in Form eines verwaschenen Anschlags, dessen Ursache nicht abschließend geklärt werden kann. Da die Filterbank ohne Frequency Shifting jedoch gute Ergebnisse liefert und hörtechnisch kein Unterschied zum Eingangssignal besteht, liegt die Vermutung nahe, dass die vertikale Phasenkohärenz durch das Frequency Shifting verloren geht.

Aufbau einer IIR Filterbank

Aufgrund von nichtlinearen Phasengängen ist der Aufbau einer IIR Filterbank deutlich schwieriger zu realisieren. Darüber hinaus steht nun die Latenzoptimierung im Mittelpunkt, sodass IIR Bandpässe mit möglichst niedriger Ordnung zu bevorzugen sind. Wie bereits durch *Juillerat et al.* beschrieben, führen die verschiedenen Einschränkungen letztlich zu einem Kompromiss aus Latenz und Klangqualität. Um die Filterbank psychoakustisch zu optimieren, werden die Parameter zum großen Teil nach dem hörtechnischen Eindruck eingestellt. Folgende Parameter führen zu einem guten Kompromiss bei transientenreichen und stark polyphonen Signalen:

- Filtertyp: Butterworth
- Filterordnung $N = 8$
- Linke Eckfrequenz $f_{3dB,1} = f_b^c - 10 \text{ Hz} - 0,4 (f_b^c - f_b^c \cdot 2^{-1/N_B})$
- Rechte Eckfrequenz $f_{3dB,2} = f_b^c + 10 \text{ Hz} + 0,4 (f_b^c \cdot 2^{1/N_B} - f_b^c)$

Die Bandpässe werden mit Butterworth-Filtern realisiert. Diese haben zwar bei gleicher Filterordnung eine niedrigere Flankensteilheit als z. B. ein elliptisches Filter, der Phasengang ist jedoch vergleichsweise linearer. Das führt bei der abschließenden Aufsummierung der Signale v_b zu geringeren Auslöschungen.

Eine hohe Filterordnung ist aufgrund des Rechenaufwands und der steigenden Gruppenlaufzeit nicht erwünscht. Des Weiteren steigt mit der Filterordnung ebenfalls die Nichtlinearität der Phasengänge, die bei der abschließenden Aufsummierung zu stärkeren Auslöschungen führt. Die Filterordnung kann somit ohne weitere Maßnahmen, wie die Linearisierung der Phasengänge mit zusätzlichen Allpässen, nicht beliebig gesteigert werden. Eine Filterordnung von $N = 8$ führt im Gegensatz zu den von *Juillerat et al.* verwendeten Butterworth-Filtern vierter Ordnung zu hörbar geringerem Detuning. Im Vergleich zur FIR Filterbank ist allerdings die hohe Flankensteilheit aufgrund der niedrigen Filterordnung nicht realisierbar.

Im tieffrequenten Bereich kommt es daher zwangsläufig zu Überlappungen, die lediglich durch die Breite des Durchlassbereichs reduziert werden können. Problematisch ist allerdings, dass ein schmalerer Durchlassbereich zu einer höheren Gruppenlaufzeit führt. Ein sinnvoller Kompromiss zwischen Gruppenlaufzeit und Detuning bei gegebener Filterordnung ist eine Bandbreite von mindestens 20 Hz.

Hinzu kommt ein prozentualer Anteil des Abstandes zum Nachbar-Frequenzband, damit in den höheren Frequenzlagen nicht zu große Lücken zwischen den Frequenzbändern entstehen. Größere Lücken reduzieren zwar das Detuning, allerdings kann das Gitarrensignal durch den entstehenden Informationsverlust verfremdet werden. Der hörtechnische Eindruck zeigt, dass der Klang unter Informationsverlust deutlicher leidet, daher werden die Eckfrequenzen so gewählt, dass der Informationsverlust niedrig gehalten wird. Der in [JAS08] realisierte Übergangsbereich bei -12 dB (Informationsverlust ca. 35 %) kann hier nicht nachvollzogen werden, da das Ausgangssignal so stark an Direktheit verliert. Die folgende Abbildung 4.16 zeigt die Amplitudengänge aller entworfenen Bandpässe.

Die Realisierung mit der IIR Filterbank zeigt, insbesondere im Verhältnis zur niedrigen Filterordnung, ebenfalls gute Ergebnisse. Stark polyphone Signale klingen sauber aus und Transienten bleiben weitestgehend erhalten. Allerdings fällt auf, dass das Ausgangssignal weniger kraftvoll bzw. hohler klingt. Dies lässt sich durch die nichtlinearen

Phasengänge der Bandpässe erklären, die bei der abschließenden Aufsummierung zu Phasenauslöschungen führen.

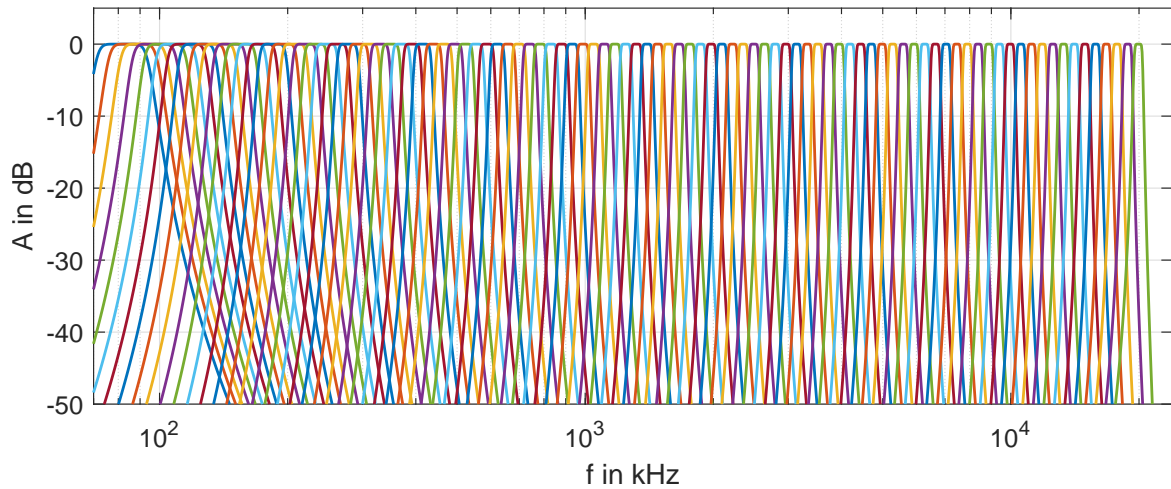


Abbildung 4.16: IIR Filterbank: Amplitudengänge aller Bandpassfilter

Realisierung als Multiraten-System

In der Praxis bietet sich die Realisierung als Multiraten-System an, um den Rechenaufwand trotz hoher Filteranforderungen zu senken. Die folgende Abbildung 4.17 zeigt einen möglichen Aufbau eines solchen Systems als Blockschaltdiagramm.

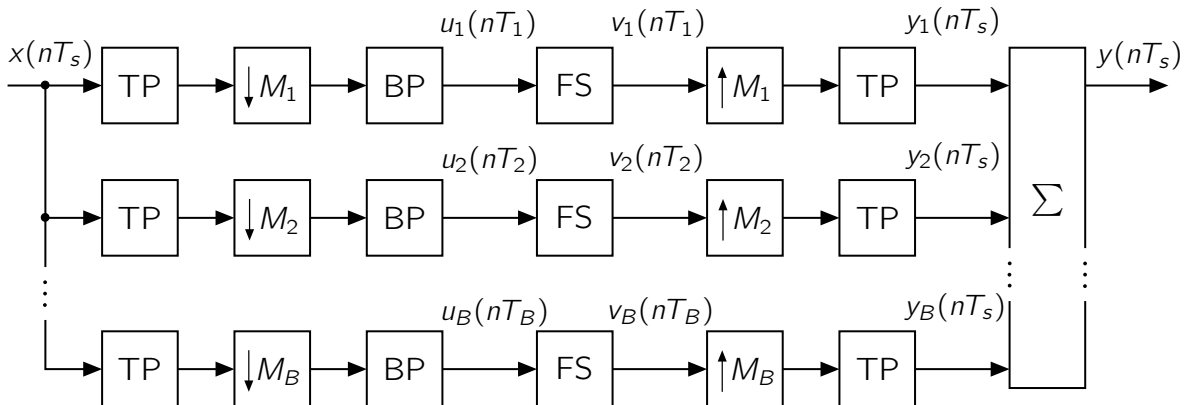


Abbildung 4.17: Realisierung als Multiraten-System

Ziel ist es, dass jeder Bandpass (BP) auf einer möglichst niedrigen Abtastfrequenz arbeitet. Dazu wird das Eingangssignal nach der Anwendung eines passenden Dezimationsfilters (TP) um den Faktor M_b dezimiert. Es muss allerdings beachtet werden,

dass das Abtasttheorem nicht durch das Pitch Shifting verletzt wird. Währenddessen Pitch Shifting nach unten unproblematisch ist, können beim Pitch Shifting nach oben die Frequenzanteile über die Nyquist-Frequenz verschoben werden, sodass Aliasing entstehen würde. Der Dezimationsfaktor muss dementsprechend kleiner gewählt werden. Nach dem Frequency Shifting werden die einzelnen Frequenzbänder mithilfe des gleichen Faktors M_b und eines Interpolationsfilters (TP) interpoliert und so auf die ursprüngliche Abtastrate gebracht.

Bei hohen Dezimations- und Interpolationsfaktoren ist zudem die Zerlegung in mehrere Dezimations- bzw. Interpolationsstufen sinnvoll, da so jeweils geringere Anforderungen an die Dezimations- bzw. Interpolationsfilter gestellt werden. In Summe benötigt eine solche Kaskade eine geringere Filterordnung. Durch die Zerlegung in eine Polyphasenstruktur kann darüber hinaus zusätzlich Rechenaufwand eingespart werden, da so die Dezimations- und Interpolationsfilter auf der niedrigeren Abtastrate laufen [KS16; KS12b].

Die Umsetzung eines Multiraten-Systems bietet sich vor allem für die Realisierung einer FIR Filterbank an. Durch die logarithmische Verteilung der Noten steigt die benötigte Filterordnung in Richtung tiefer Frequenzen an. Eine individuelle Dezimierung der Frequenzbänder mit M_b führt dazu, dass alle Bandpässe mit einer ähnlich niedrigen Filterordnung auskommen, ohne auf eine hohe Flankensteilheit zu verzichten.

Die gleichen Vorteile ergeben sich grundsätzlich auch bei der Realisierung einer IIR Filterbank. Da allerdings die Phasengänge von IIR Bandpässen immer nichtlinear sind, kommt es, wie bereits beschrieben, bei der einfachen Aufsummierung zu Phasenauslöschungen.

5 Algorithmen im Frequenzbereich

WSOLA ist in der Lage dominante periodische Strukturen zu erhalten, bei stark polyphonen Signalen scheitert das Verfahren jedoch. Algorithmen im Frequenzbereich basieren häufig auf dem Phase Vocoder und ermöglichen es alle periodischen Strukturen des Signals zu erhalten, wodurch bei polyphonen Signalen eine höhere Klangqualität erreicht werden kann. Der Phase Vocoder stellt dabei keine eigene Transformation dar, sondern baut auf bestehende Zeit-Frequenz-Analysewerkzeuge auf. Überwiegend wird die STFT verwendet, da sie effizient zu implementieren ist. In Abschnitt 5.1 wird daher der Phase Vocoder zunächst anhand des Transformationsergebnisses der STFT erläutert.

Bei der STFT führt die äquidistante Verteilung der Frequenzstützstellen allerdings dazu, dass (aufgrund der logarithmischen Verteilung der musikalischen Noten) im tieffrequenten Bereich zu wenige und im hohen Frequenzbereich zu viele Frequenzstützstellen platziert sind. Wie zuvor beim Roller Algorithmus, ist eine logarithmische Verteilung der Frequenzstützstellen wünschenswert. Diesen Punkt greift die CQT auf, die die Frequenzstützstellen mit konstanter Güte Q auf der Frequenzachse verteilt. Dieser Ansatz wird in Abschnitt 5.2 diskutiert.

5.1 Phase Vocoder auf Basis der Short-Time Fourier Transformation

5.1.1 Standard Phase Vocoder

Wie bereits in Unterabschnitt 2.2.2 aufgezeigt, hängt bei der DFT bzw. STFT die Frequenzauflösung von der Blocklänge N ab. Um beispielsweise eine Frequenzauflösung von 5 Hz zu erreichen, muss bei einer Abtastfrequenz von 48 kHz eine Blocklänge von $N = 9600$ verwendet werden. Dies führt allerdings auch zu einer Latenz von 200 ms. Die niedrigsten Noten E2 und F2 liegen auf der Frequenzachse nur 4.9 Hz auseinander (siehe Tabelle 3.1). Eine Verringerung der Blocklänge zugunsten der Latenz führt jedoch zu einer noch größeren Frequenzauflösung und damit sehr ungenauen Frequenzbestimmung dieser Frequenzanteile.

Bestimmung von Momentanfrequenzen

Eine Verbesserung der Frequenzbestimmung ist mithilfe des Phase Vocoders möglich. Dieser wurde 1966 von *Flanagan* und *Golden* in [FG66] vorgestellt und zunächst mithilfe einer Filterbank realisiert. Im Jahre 1986 stellte *Dolson* in [Dol86] eine effiziente Implementierung des Phase Vocoders mithilfe der FFT vor. Weitere Erläuterung sind in [Zöl11; Dri11; DM16; Ond18; Roy19; Set07; RB19] zu finden. Grundlegend wird davon ausgegangen, dass das Eingangssignal aus einer gewichteten Summe von Sinusoide besteht. Die Frequenzbestimmung kann verbessert werden, indem der Informationsgehalt der Phasenspektren ausgenutzt wird. Der Phase Vocoder vergleicht die Phasenwinkel einzelner Frequenzanteile bzw. Sinusoide von aufeinander folgenden Blöcken, um aus deren Abweichung die wahre Frequenz genauer zu bestimmen. In der Literatur wird dies auch als Bestimmung der Momentanfrequenz (englisch *Instantaneous Frequency*) bezeichnet. Dies stellt die Analyse-Stufe des Phase Vocoders dar. Das Pitch Shifting bzw. Time Scaling wird erst nach der Analyse-Stufe berechnet.

Im Folgenden wird die Funktionsweise der Analyse-Stufe des Phase Vocoders anhand eines Sinus-Signals mit einer Frequenz von 220 Hz erläutert. Die STFT verwendet ein Von-Hann-Fenster und eine Blocklänge von $N = 2048$ mit einem Überlappungsgrad von 50 %. Die Abtastfrequenz beträgt 48 kHz. Diese Parameter führen zu einer Frequenzauflösung von:

$$\Delta f = \frac{F_s}{N} = \frac{48 \text{ kHz}}{2048} = 23.438 \text{ Hz} \quad (5.1)$$

Die folgende Abbildung 5.1 zeigt das Amplituden- sowie Phasenspektrum von zwei aufeinander folgenden Blöcken m und $m + 1$.

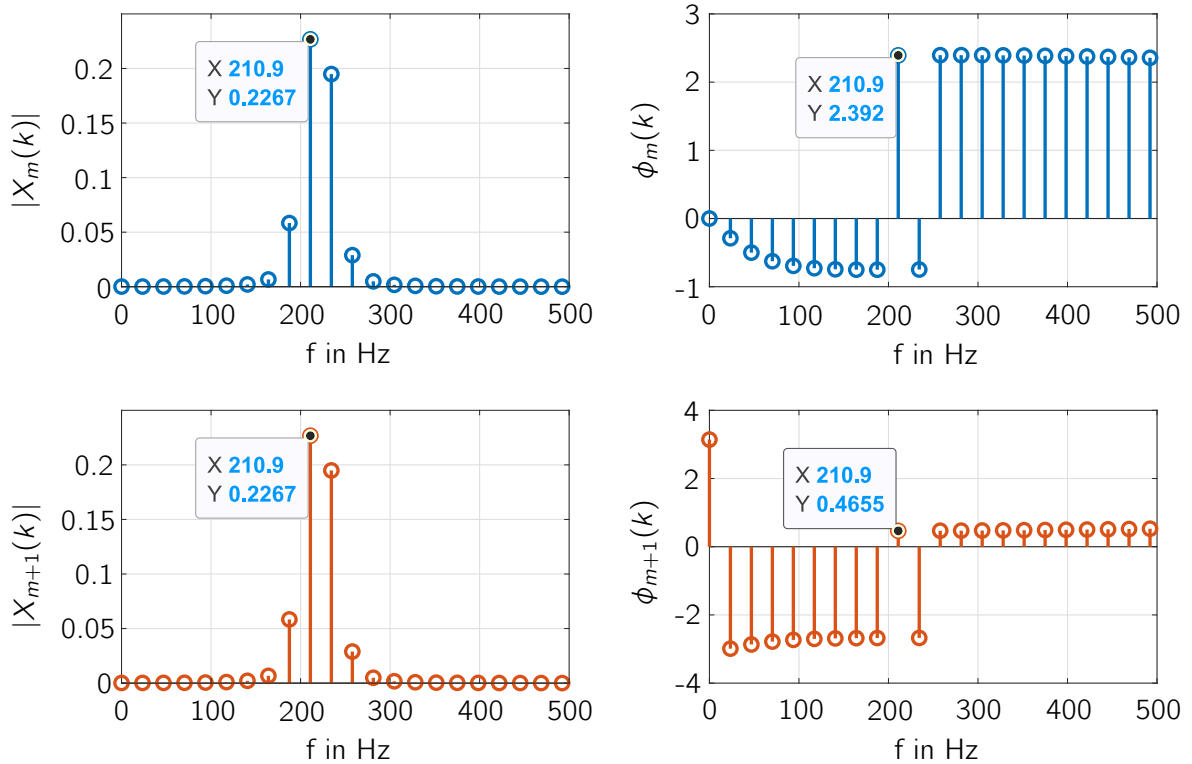


Abbildung 5.1: Phase Vocoder: Nutzung des Phasenspektrums zur Verbesserung der Frequenzbestimmung

Von dem Sinus-Signal werden keine ganzen Perioden erfasst, daher wird die Energie auf mehrere Frequenzstützstellen verteilt (Leck-Effekt). In den Amplitudenspektren liegt das Maximum bzw. der Peak in beiden Fällen bei 210.9 Hz. Die Abweichung von der wahren Frequenz beträgt damit 9.1 Hz. Der Phase Vocoder nutzt nun die Phasenspektren, um den Phasenfehler und damit die Momentanfrequenz zu berechnen. Mithilfe des Phasenwinkels des entsprechenden DFT Koeffizients kann die Phase nach dem Zeitabschnitt $\Delta t = T_s \cdot H_A = T_s \cdot N/2$ (50% Überlappung) vorhergesagt werden. $F(k) = \omega_k = \frac{2\pi k F_s}{N}$ bezeichnet dabei die Frequenz der k -ten Frequenzstützstelle.

$$\begin{aligned} \phi_m^{Pred}(k) &= \phi_m(k) + \omega_k \Delta t \\ &= 2.392 \text{ rad} + 2\pi \cdot 210.938 \text{ Hz} \cdot T_s \cdot H_A \\ &= 30.666 \text{ rad} \end{aligned} \quad (5.2)$$

Mithilfe dieses aufsummierten Phasenwinkels (auch *Unwrapped Phase* genannt) und

dem genauen Phasenwinkel vom nächsten Block $m + 1$ kann nun der Phasenfehler $\phi_m^{Err}(k)$ berechnet werden. Die Anweisung *wrap* bezeichnet dabei das Beschränken des Phasenwinkels auf den Wertebereich $(-\pi, \pi]$ (auch *Phase Wrapping* genannt).

$$\begin{aligned}\phi_m^{Err}(k) &= \text{wrap}(\phi_{m+1}(k) - \phi_m^{Pred}(k)) \\ &= \text{wrap}(0.466 \text{ rad} - 30.666 \text{ rad}) \\ &= 1.215 \text{ rad}\end{aligned}\tag{5.3}$$

Das Phase Wrapping ist nur fehlerfrei, wenn der Phasenfehler höchstens eine halbe Periode beträgt. Der Phase Vocoder erzeugt daher allgemein bessere Ergebnisse, wenn die grobe Frequenzbestimmung durch die STFT bereits sehr nah an der wahren Frequenz liegt und die Analyse-Schrittweite H_A klein ist.

Die Momentanfrequenz $F_m^{IF}(k)$ für den k -ten DFT Koeffizienten ergibt sich abschließend durch:

$$\begin{aligned}F_m^{IF}(k) &= \omega_k + \frac{\phi_m^{Err}(k)}{\Delta t} \\ &= 2\pi \cdot 210.938 \text{ Hz} + \frac{1.215 \text{ rad}}{T_s \cdot H_A} \\ &= 2\pi \cdot 220.003 \text{ Hz}\end{aligned}\tag{5.4}$$

Dieses Verfahren wird für alle Peaks wiederholt und funktioniert, solange die Peaks gut von einander getrennt sind [DM16]. Die Blocklänge der STFT kann somit (aufgrund des Unschärfe-Prinzips) nicht beliebig klein gewählt werden, da die Peaks ansonsten ineinander verschwimmen (vergleiche Abbildung 2.8).

Time Scaling und Pitch Shifting

Wie bei den Algorithmen im Zeitbereich wird Time Scaling durch die Neupositionierung der Blöcke auf der Zeitachse erreicht, indem die Synthese-Schrittweite unterschiedlich zur Analyse-Schrittweite gewählt wird ($H_A \neq H_S$). Die bloße Neupositionierung entspricht allerdings dem aus Abschnitt 4.1 bekannten OLA Verfahren und führt zu Phasensprüngen. Dies wird auch als Verlust der horizontalen Phasenkohärenz bezeichnet. Um die Phasenkohärenz zu erhalten, ist die Anpassung des Phasenspektrums notwendig, die mithilfe des Phase Vocoder durchgeführt werden kann. Dieser Zusammenhang ist in der folgenden Abbildung 5.2 anhand einer einzelnen Sinusschwingung visualisiert. Das Phasenspektrum wird so verändert, dass die in rot eingezeichnete Sinusschwingung nach der Neupositionierung nahtlos an die blaue Sinusschwingung anknüpft.

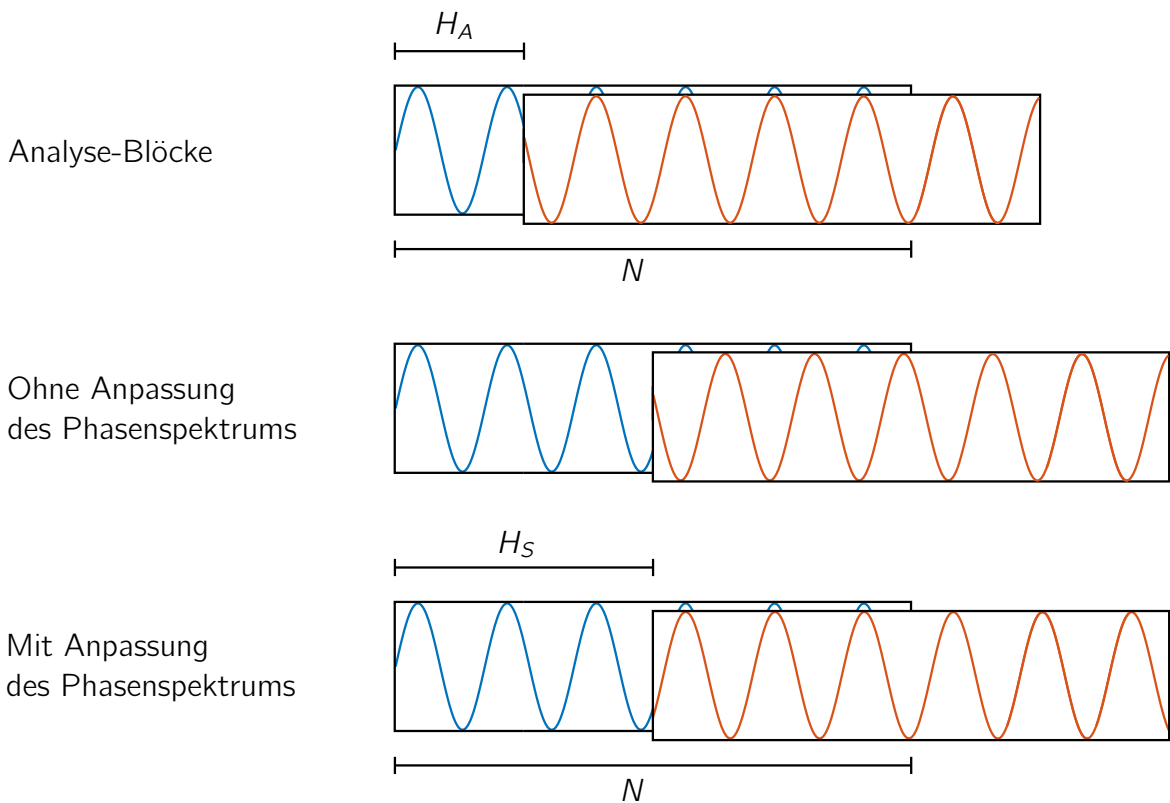


Abbildung 5.2: Time Scaling: Erhalt der horizontalen Phasenkohärenz durch Anpassung des Phasenspektrums

Mithilfe der STFT und dem Phase Vocoder kann die Anpassung des Phasenspektrums so erfolgen, dass die Phasenwinkel aller im Signal enthaltenen Frequenzanteile korrigiert werden. Selbst bei stark polyphonen Signalen wird so die horizontale Phasenkohärenz sichergestellt.

Wie bei den Algorithmen im Zeitbereich kann das Pitch Shifting durch die Kombination aus Time Scaling und Resampling erreicht werden. Die direkte Berechnung des Pitch Shiftings im Frequenzbereich ist jedoch ebenso denkbar (siehe Abbildung 5.3). Statt der Neupositionierung der Blöcke wird nun die Frequenz der Sinusschwingung im Frequenzbereich verändert ($H_A = H_S$). Ohne Anpassung des Phasenspektrums kommt es allerdings in gleicher Weise zu Phasensprüngen, welche durch den Phase Vocoder behoben wird. In der Praxis wird bei der Verwendung der STFT jedoch die Kombination aus Time Scaling und Resampling eingesetzt, da durch die äquidistante Verteilung der Frequenzstützstellen die Frequenzskalierung nur durch eine aufwendige Interpolation der DFT Koeffizienten möglich ist. Aus diesem Grund wird im Folgenden nur die Anpassung des Phasenspektrums für das Time Scaling erläutert. Eine direkte Berech-

nung des Pitch Shiftings ist hingegen mithilfe der CQT möglich und wird im nächsten Abschnitt 5.2 aufgezeigt.

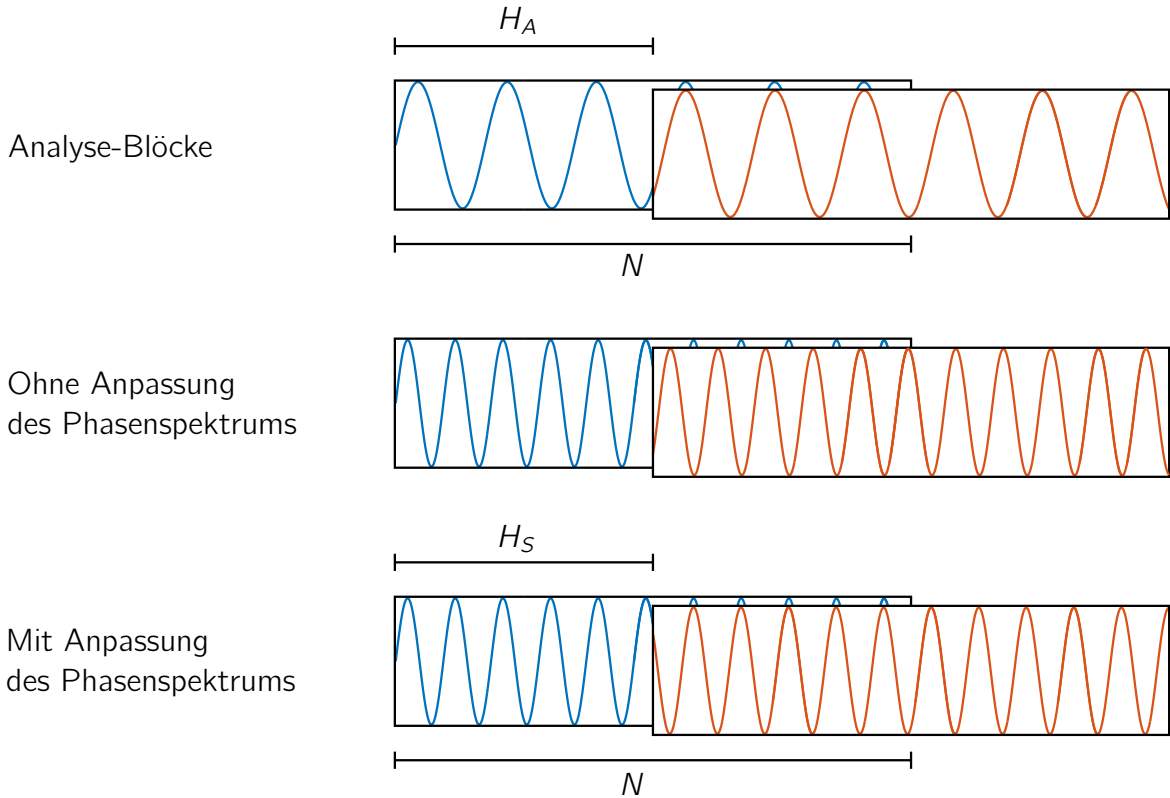


Abbildung 5.3: Pitch Shifting: Erhalt der horizontalen Phasenkohärenz durch Anpassung des Phasenspektrums

Für das Time Scaling wird nun ein modifiziertes Spektrum $X_m^{Mod}(k)$ gesucht, welches die horizontale Phasenkohärenz aller Frequenzanteile sicherstellt:

$$X_m^{Mod}(k) = |X_m(k)| \cdot e^{2\pi\phi_m^{Mod}(k)} \quad (5.5)$$

Die Phasenspektren werden iterativ berechnet. Dies wird in der Literatur auch als *Phase Propagation* bezeichnet. Im Folgenden wird davon ausgegangen, dass das Phasenspektrum des $(m - 1)$ -ten Blocks bereits korrigiert wurde. Anhand dieses modifizierten Phasenspektrums und der zuvor berechneten Momentanfrequenzen können nun die zu erwartenden Phasenwinkel nach dem Zeitabschnitt $\Delta t = H_S T_s$ bestimmt werden:

$$\phi_m^{Mod}(k) = \phi_{m-1}^{Mod}(k) + F_{m-1}^{IF}(k) \cdot H_S T_s \quad (5.6)$$

Für den ersten Block $m = 0$ wird zudem das originale Phasenspektrum übernommen:

$$\phi_0^{Mod}(k) = \phi_0(k) \quad (5.7)$$

Ist die Berechnung der Momentanfrequenzen exakt, werden ebenfalls die Phasenspektren korrekt berechnet, sodass die horizontale Phasenkohärenz von Block zu Block gegeben ist [DM16].

5.1.2 Verbesserter Phase Vocoder

Die horizontale Phasenkohärenz ist durch die im vorherigen Abschnitt beschriebenen Maßnahmen auch für stark polyphone Signale sichergestellt. Problematisch ist allerdings die Phasenkohärenz zwischen den DFT Koeffizienten innerhalb eines Blocks, welche auch als vertikale Phasenkohärenz bezeichnet wird und durch die Korrektur der Phasenspektren zerstört wird. Infolgedessen werden insbesondere Transienten stark abgeschwächt, sodass das Signal weniger direkt bzw. verhallt klingt. In der Literatur sind diese typischen Artefakte des Phase Vocoder auch unter *Phasiness*, *Reverberant* und *Loss of Presence* zu finden [Dri11; DM14a; LD97].

Eine häufig verwendete Methode zur Erhaltung der vertikalen Phasenkohärenz ist das *Identity Phase Locking*, welches 1999 von *Laroche* und *Dolson* in [LD99] vorgestellt wurde und eine Modifikation des Standard Phase Vocoder darstellt. Weitere Erläuterungen und Einsatzgebiete sind zudem in [LD00] dargestellt. Laut *Laroche* und *Dolson* beeinflusst ein einzelner Sinusoid (Peak im Spektrum) innerhalb eines Blocks direkt mehrere umliegende DFT Koeffizienten. Die Phasenwinkel der Koeffizienten innerhalb dieses Einflussbereichs dürfen daher nicht individuell angepasst werden, sondern müssen gemeinsam aktualisiert werden.

Im ersten Schritt wird im Amplitudenspektrum nach lokalen Maxima gesucht, da diese (ausgehend von der eingangs beschriebenen Modellbildung) einzelne im Signal enthaltene Sinusoide beschreiben. Eine mögliche Vorgehensweise ist der Vergleich der Amplitudenwerte innerhalb eines Blocks. Ist z.B. die Amplitude eines Koeffizienten größer als seine vier Nachbarn, wird dieser als Peak eingestuft. Dies stellt zwar eine recht primitive, aber dafür recheneffiziente Lösung dar. Die gefundenen Peaks unterteilen die Frequenzachse nun in „Einflussbereiche“, die um die Peaks herum verteilt sind. Ein Vorteil des Identity Phase Lockings besteht darin, dass die Berechnung der Momentanfrequenz nach Gleichung 5.4 und die Berechnung der Phase Propagation nach Gleichung 5.6 nur für die gefundenen Peaks durchgeführt wird. Die Phasen der anderen Koeffizienten werden nach dem neuen Phasenwinkel des entsprechenden Peaks (je nach Einflussbereich) angepasst.

Ziel des Verfahrens ist es, dass die Phasenunterschiede in dem unmodifizierten Analyse-Block (Ergebnis der STFT) zwischen dem Peak und den umliegenden Koeffizienten in der Synthese nachgebildet werden. k_P bezeichnet im Folgenden den Index eines gefundenen Peaks:

$$\phi_m^{Mod}(k) = \phi_m^{Mod}(k_P) - \phi_m(k_P) + \phi_m(k) \quad (5.8)$$

Durch diese Anpassung bleiben die ursprünglichen Phasenverhältnisse erhalten, wodurch die vertikale Phasenkohärenz deutlich verbessert wird. Der Rechenaufwand lässt sich zudem weiter minimieren, wenn zunächst nur der Phasenunterschied bzw. Rotationswinkel für jeden Peak berechnet wird:

$$\Phi_P = \phi_m^{Mod}(k_P) - \phi_m(k_P) \quad (5.9)$$

Aus dem Phasenunterschied wird anschließend für jeden Einflussbereich ein individueller Phasor gebildet:

$$Z_P = e^{j\Phi_P} \quad (5.10)$$

Der Phasor wird auf jeden Koeffizienten des entsprechenden Einflussbereichs angewendet. Der Ablauf des verbesserten Phase Vocoders nach dem Identity Phase Locking lässt sich schließlich folgendermaßen zusammenfassen [LD99]:

- 1) Peak-Suche in $|X_{m-1}(k)|$
- 2) Berechnung der Momentanfrequenz nach Gleichung 5.4 und des neuen Phasenwinkels nach Gleichung 5.6
- 3) Berechnung des Rotationswinkels Φ_P und Phasors Z_P
- 4) Anwendung des Phasors auf alle Koeffizienten im Einflussbereich inkl. des Peak-Koeffizienten: $X_m^{Mod}(k) = Z_P \cdot X_m(k)$
- 5) Wiederholung der vorherigen Schritte für alle anderen Peaks
- 6) Nächsten Block $X_{m+1}(k)$ verarbeiten

Zur Verbesserung der vertikalen Phasenkohärenz existieren zahlreiche weitere Varianten wie z. B. das *Scaled Phase Locking*, welches in der Frequenz variierende Sinusoide bei der Berechnung der Phase Propagation berücksichtigt [LD99]. Untersuchungen von *Laroche* und *Dolson* zeigen in Hörtests gegenüber dem Identity Phase Locking weitere klangliche Verbesserungen.

Des Weiteren kann es bei der Berechnungen der Momentanfrequenzen aufgrund des Phase Wrappings zu Fehlern kommen, die sich in den nächsten Blöcken fortsetzen und über die Zeit aufaddieren. Dieser Punkt wird von *Moinet* und *Dutoit* [MD11] mit dem *PVSOLA* (*Phase Vocoder with Synchronized Overlap-Add*) Verfahren aufgegriffen. Die Phasen werden regelmäßig zurückgesetzt, indem der originale Analyse-Block unmodifiziert in der Synthese verwendet wird. Die optimale Position des Blocks im Ausgangssignal wird dabei ähnlich wie bei SOLA bzw. WSOLA mittels Kreuzkorrelation berechnet.

Der PVSOLA Ansatz wurde zudem durch *Kraft et al.* in [Kra+12] aufgegriffen und leicht modifiziert. Statt den Analyse-Block unverändert in der Synthese zu verwenden, werden zumindest die Koeffizienten der gefundenen Peaks mithilfe des Phase Vocoders angepasst. Die Phasenwinkel der jeweils zwei nächsten Nachbar-Koeffizienten werden aus dem Eingangssignal übernommen und somit zurückgesetzt. Darüber hinaus wird das Scaled Phase Locking integriert.

In dem *PhaVoRIT* (*Phase Vocoder for Real-Time Interactive Time-Stretching*) Verfahren von *Karrer et al.* werden die Phasen hingegen während stillen Abschnitten zurückgesetzt, damit mögliche Artefakte in dem Moment möglichst nicht hörbar sind [KLB06].

Eine weitere Methode wurde außerdem von *Průša* und *Holighaus* in [PH17a] präsentiert, in der die Berechnung der Phasenwinkel mithilfe von Gradientenbestimmungen durchgeführt wird und so den Erhalt der Phasenkohärenz weiter optimiert wird. Vorteilhaft ist zudem, dass bei dem Verfahren keine Peak-Suche oder Transienten-Detektion benötigt wird. Eine Implementierung des Algorithmus ist unter [PH17b] zu finden.

Da das Identity Phase Locking bei Gitarrensingen bereits eine ausreichend hohe Klangqualität erreicht und das Ziel dieser Arbeit die Latenzoptimierung ist, wird in Kapitel 6 der Phase Vocoder mit dieser Methode verwendet.

5.2 Phase Vocoder auf Basis der Constant-Q Transformation

Wie im vorherigen Abschnitt erläutert, wird bei der STFT das Pitch Shifting über Time Scaling und entsprechendes Resampling berechnet, da direktes Pitch Shifting nur mit aufwendiger Interpolation möglich ist. Die äquidistante Frequenzauflösung ist für musikalische Signale nicht ideal. Bei der in Abschnitt 2.3 vorgestellten Constant-Q Transformation können die Frequenzstützstellen hingegen exakt auf die musikalischen Noten verteilt werden, sodass in den tiefen Frequenzen eine hohe Frequenzauflösung und in den hohen Frequenzen eine hohe Zeitauflösung erreicht wird. Die logarithmische Verteilung führt außerdem dazu, dass die Frequenzskalierung in eine einfache Verschiebung der Frequenzstützstellen übergeht. Das Pitch Shifting ist daher verhältnismäßig einfach direkt im Frequenzbereich möglich. Vorgestellt wurde diese Methode von *Schörkhuber et al.* in [SKS13] und [Sch+14], auf der dieser Abschnitt ebenfalls basiert. In [Sch+13] ist zudem eine MATLAB[®] Implementierung zum Pitch Shifting mithilfe der CQT bereitgestellt. Mit [HDL15] ist ein weiterer Artikel zu finden, der auf die beiden Veröffentlichungen aufbaut und den Ansatz auf die sliCQ überträgt. Die sliCQ wird in [Dör+13] ebenfalls als MATLAB[®] Implementierung bereitgestellt.

Die CQT ermöglicht auch speziellere Modifikationen, wie zum Beispiel das Ausschneiden einzelner Noten mittels Maskierung. Des Weiteren ist der Rechenaufwand durch das direkte Pitch Shifting unabhängig vom Skalierungsfaktor α . Die CQT muss allerdings bestimmte Voraussetzungen beim Zeit-Frequenz-Gitter und bei der Darstellung der Systemkoeffizienten erfüllen, um das Pitch Shifting berechnen zu können. Bevor in Unterabschnitt 5.2.3 die Unterschiede zur STFT erläutert werden, werden in Unterabschnitt 5.2.1 und 5.2.2 zunächst die Voraussetzungen an die CQT diskutiert. In Unterabschnitt 5.2.4 wird erläutert, welche Folgen eine echtzeitfähige Implementierung (sliCQ) auf die Zeit-Frequenz Ebene hat.

5.2.1 Wahl eines Zeit-Frequenz-Gitters

Die Möglichkeit des Pitch Shiftings direkt im Frequenzbereich hängt nicht nur von der Verteilung der Frequenzstützstellen, sondern auch von der Platzierung der Abtastzeitpunkte entlang der Zeitachse ab. Das Zeit-Frequenz-Gitter mit minimaler Redundanz aus Abbildung 2.12 ist für Pitch Shifting nicht geeignet, da die Koeffizienten nicht in der Frequenz verschoben werden können, ohne ihre zeitliche Position zu verändern. Benötigt wird eine Darstellung in der die Koeffizienten (zeitlich) vertikal ausgerichtet sind. In der folgenden Abbildung 5.4 sind zwei mögliche Darstellungen aufgezeigt, die für das Pitch Shifting geeignet sind.

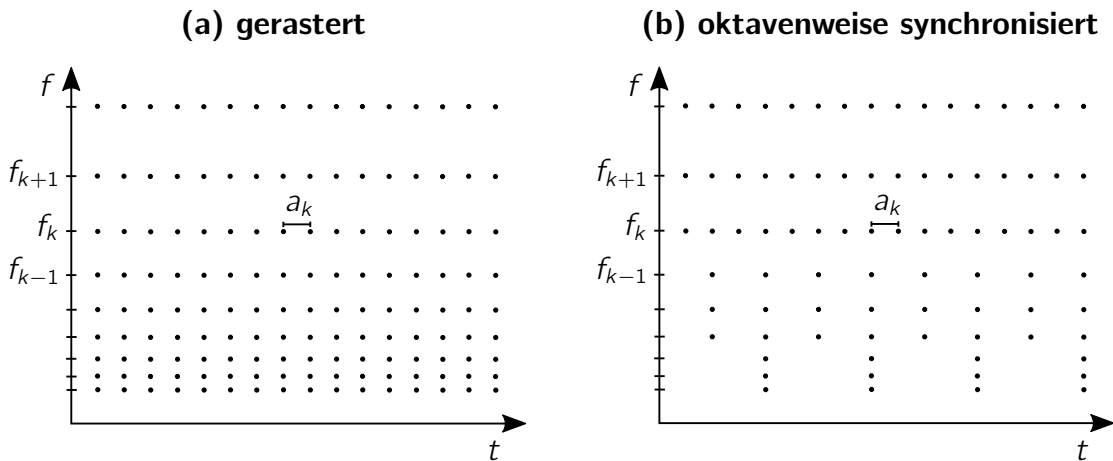


Abbildung 5.4: Geeignete Zeit-Frequenz-Gitter für Pitch Shifting (nach [SKS13, S. 567])

Die gerasterte Darstellung in (a) orientiert sich an der kleinsten Analyse-Schrittweite der höchsten Frequenzstützstelle und übernimmt diese für alle Frequenzkanäle. Diese Darstellung ermöglicht sowohl das Pitch Shifting nach oben mit $\alpha > 1$, als auch nach unten mit $\alpha < 1$, besitzt allerdings eine deutlich höhere Redundanz und damit einen höheren Rechenaufwand. Alternativ kann die Darstellung oktavenweise synchronisiert werden (b), indem die Analyse-Schrittweiten an die der höchsten Frequenz der jeweiligen Oktave angepasst werden. Die Redundanz wird leicht erhöht, allerdings eignet sich die Darstellung nicht für das Pitch Shifting nach oben, da an den Oktav-Übergängen Abtastpunkte fehlen. *Schörkhuber et al.* schlägt eine Verdopplung der Abtastpunkte (Halbierung der Analyse-Schrittweiten) bis auf in der höchsten Oktave vor, um das Pitch Shifting bis eine Oktave nach oben zu ermöglichen. Grundsätzlich ist für $\alpha < 1$ zu beachten, dass die CQT Frequenzstützstellen im Frequenzbereich kleiner f_{E2} besitzt, um die Verschiebung der Koeffizienten nach unten zu ermöglichen. Soll das Gitarrensiegel beispielsweise um eine Oktave nach unten verschoben werden ($\alpha = 0,5$), muss das Zeit-Frequenz-Gitter ebenfalls um eine Oktave nach unten bis $f_{E1} = 41.2$ Hz erweitert sein.

5.2.2 Darstellung der Systemkoeffizienten

Um den Rechenaufwand bei der Berechnung der CQT zu minimieren, werden die Systemkoeffizienten auf der Zeit-Frequenz-Ebene nur im Abstand a_k berechnet. Wie in Unterabschnitt 2.3.4 beschrieben, entspricht dies einer Unterabtastung im Frequenzbereich (Reduzierung von DFT Koeffizienten), die allerdings nicht mit einer Unterabtastung im Zeitbereich identisch ist. Das Amplitudenspektrum ist zwar identisch, allerdings unterscheidet sich das berechnete Phasenspektrum, das in der Interpretation

weniger intuitiv ist. Da auch beim Pitch Shifting auf Basis der CQT das Phasenspektrum mithilfe des Phase Vocoders angepasst werden muss, um die Phasenkohärenz zu erhalten, ist eine übliche Darstellung des Phasenspektrums wünschenswert.

Nach *Schörkhuber et al.* dürfen die verbleibenden DFT Koeffizienten nicht (wie in Unterabschnitt 2.3.4 beschrieben) zentriert um die Null liegen, sondern müssen mit der folgenden Mapping-Funktion in das Frequenzintervall $(-F_s^k/2, F_s^k/2]$ verschoben werden:

$$M(f, F_s^k) = f - \text{floor}\left(\frac{f}{F_s^k}\right) F_s^k \quad (5.11)$$

Dies entspricht einer zyklischen Verschiebung um $M(f_k, F_s^k)$. Durch diese angepasste Berechnung der Systemkoeffizienten ist das Phasenspektrum im klassischen Sinne interpretierbar.

5.2.3 Pitch Shifting mithilfe der CQT

Anpassung der Amplitudenspektren

Durch die logarithmische Verteilung der Frequenzstützstellen geht die für das Pitch Shifting notwendige Frequenzskalierung in eine einfache Verschiebung der Systemkoeffizienten entlang der Frequenzachse über. Die notwendige Verschiebung r hängt neben dem Skalierungsfaktor α ebenfalls von der Anzahl der Frequenzstützstellen pro Oktave N_B ab:

$$r = N_B \log_2(\alpha) \quad (5.12)$$

Werden für den Skalierungsfaktor, wie in Gleichung 3.2, nur Halbtonschritte zugelassen, ist der Verschiebungsfaktor r immer ganzzahlig. Für $N_B > 12$ ist zudem eine Tonhöhenverschiebung kleiner als ein Halbtonschritt möglich. Bei $N_B = 48$ ist beispielsweise eine Tonhöhenverschiebung um 25 cent möglich, wobei r mit ± 1 ebenfalls ganzzahlig ist. Eine Interpolation des Amplitudenspektrums entfällt daher ebenso.

Anpassung der Phasenspektren

Durch die Verschiebung der Koeffizienten entlang der Frequenzachse ist das Amplitudenspektrum bereits korrekt modifiziert. Wie in Abbildung 5.3 erläutert, geht durch die Veränderung der Tonhöhe allerdings die horizontale Phasenkohärenz verloren, die

ähnlich wie in Abschnitt 5.1 mithilfe des Phase Vocoders korrigiert werden muss. Um ebenfalls die vertikale Phasenkohärenz möglichst zu erhalten, wird der Phase Vocoder mit Identity Phase Locking verwendet.

Im Folgenden wird davon ausgegangen, dass die Koeffizienten bereits entlang der Frequenzachse um r Stützstellen verschoben wurden. Der Ablauf entspricht daher grundlegend dem aus Unterabschnitt 5.1.2, es gilt nun allerdings $H_A = H_S = a_k$.

Der neue Phasenwinkel des Peaks k_P berechnet sich daher durch:

$$\phi_m^{Mod}(k_P) = \phi_{m-1}^{Mod}(k_P) + F_{m-1}^{IF}(k_P) \cdot a_k T_s \quad (5.13)$$

Übertragen auf die in Abschnitt 2.3 definierte CQT ergibt sich der folgende Ablauf:

- 1) Peak-Suche in $|c_{m-1,k}|$
- 2) Berechnung der Momentanfrequenz $F_{m-1}^{IF}(k_P)$ und des neuen Phasenwinkels $\phi_m^{Mod}(k_P)$
- 3) Berechnung des Rotationswinkels Φ_P und Phasors Z_P
- 4) Anwendung des Phasors auf alle Koeffizienten im Einflussbereich inkl. des Peak-Koeffizienten: $c_{m,k}^{Mod} = Z_P \cdot c_{m,k}$
- 5) Wiederholung der vorherigen Schritte für alle anderen Peaks
- 6) Nächsten Zeitpunkt $c_{m+1,k}$ verarbeiten

Dadurch, dass die Frequenzstützstellen der CQT auf die musikalischen Noten verteilt sind, ist die Momentanfrequenz des gefundenen Peaks $F_{m-1}^{IF}(k_P) \approx F(k_P)$. Um den Rechenaufwand zu reduzieren, kann daher optional auf die Berechnung der Momentanfrequenzen verzichtet werden und der neue Phasenwinkel des Peaks k_P durch

$$\phi_m^{Mod}(k_P) = \phi_{m-1}^{Mod}(k_P) + F(k_P) \cdot a_k T_s \quad (5.14)$$

berechnet werden. Laut *Schörkhuber et al.* können durch diese Approximation zwar leichte Frequenz- und Amplitudenmodulationen entstehen, allerdings sind diese laut Hörtests in der Praxis kaum wahrnehmbar. Da die Methode durch die entfallende Bestimmung der Momentanfrequenzen zusätzlich robuster ist, wird die Phasenkorrektur nach Gleichung 5.14 durchgeführt.

5.2.4 Pitch Shifting mithilfe der sliCQ

Der im vorherigen Unterabschnitt beschriebene Ablauf basiert auf der normalen CQT und ist damit nicht echtzeitfähig. Die Methodik kann grundsätzlich auch auf die sliCQ übertragen werden, allerdings führt dies im Rahmen der Latenzoptimierung zu erheblichen Problemen.

Durch die geringe Bandbreite der Filter g_k im tiefen Frequenzbereich sind die entsprechenden Analyse-Fenster $\mathcal{F}^{-1}g_k$, bei Betrachtung der nennenswerten Energie, sehr lang. Für die normale CQT, die eine FFT über das gesamte Signal durchführt, stellt dies kein Problem dar. Die Realisierung als Echtzeitsystem (sliCQ) funktioniert hingegen nur, wenn für die Zerlegung des Eingangssignals eine große Blocklänge (und damit hohe Frequenzauflösung) gewählt wird oder die Bandbreiten der Filter entsprechend erhöht werden. Bei dem gewünschten Constant-Q System mit 12 Stützstellen pro Oktave sind die Filterbandbreiten B_k (siehe Gleichung 2.35) derart niedrig, dass die Zerlegung des Eingangssignals mit Blocklängen $> 10\,000$ Samples bei $F_s = 48\text{ kHz}$ (entspricht $\tau > 208\text{ ms}$) durchgeführt werden muss. Ist die Blocklänge zu klein, können die Filter nicht mehr an die gewünschten Frequenzpunkte platziert werden. Ein solches System kann die vorgegebene Spezifikation nicht erfüllen, sodass die gesamten Vorteile einer Constant-Q Darstellung verfallen.

Für das Pitch Shifting bedeutet das, dass die einfache Verschiebung der Koeffizienten um r Stützstellen nicht mehr möglich ist. Daran ändern auch Systeme mit variabler Güte nichts (wie sie in [Sch+14], [HDL15] sowie [Hol13] vorgestellt werden), die die Güte zu den tiefen Frequenzen hin langsam verringern, um die Länge der Analyse-Fenster zu verkürzen. Durch die höhere Bandbreite der Filter verringert sich im tiefen Frequenzbereich zwangsläufig die Anzahl an Frequenzstützstellen, sodass nicht alle Halbtonschritte abdeckt werden und die einfache Verschiebung entlang der Frequenzachse nicht mehr möglich ist.

Insgesamt lässt sich festhalten, dass die Vorteile gegenüber der STFT durch die angestrebte Latenzoptimierung verfallen. Aus diesem Grund wird die Tonhöhenverschiebung mithilfe der sliCQ in Kapitel 6 nicht weiter verfolgt.

6 Ergebnisse und Auswertung

In diesem Kapitel werden die einzelnen Algorithmen hinsichtlich ihrer Klangqualität und Latenz beurteilt und miteinander verglichen. Die Einschätzung der Klangqualität wird, neben einer hörtechnischen Beurteilung des Autors, mithilfe objektiver Audiomerkmale durchgeführt. In Abschnitt 6.4 wird die erreichbare Latenz und Effizienz der Algorithmen beurteilt.

In diesem Kapitel werden die folgenden Algorithmen ausgewertet:

- OLA
- WSOLA
- Roller
- *Phase Vocoder auf Basis der STFT (PV-STFT)*
- *élastiquePro* des Herstellers *zplane*

Das SOLA Verfahren und Pitch Shifting mithilfe der *slicQ* werden hier, aus den in den entsprechenden Abschnitten 4.2 bzw. 5.2.4 genannten Gründen, nicht betrachtet. An dieser Stelle ist noch einmal hervorzuheben, dass der Phase Vocoder auf Basis der CQT nicht echtzeitfähig ist und sich daher nicht für das anvisierte latenzoptimierte Echtzeitsystem eignet.

Als klangliche Referenz wird der kommerzielle Algorithmus *élastiquePro* in die Vergleiche miteinbezogen. Dieser ist beispielsweise in der DAW-Software Studio One[®] 4 des Herstellers *PreSonus Audio Electronics, Inc.* integriert. Die in der Tonhöhe verschobenen Testsignale werden direkt in Studio One[®] erstellt und für den Vergleich mit den untersuchten Algorithmen als Wave-Datei exportiert.

MATLAB[®] Implementierungen

Die genannten Algorithmen werden mithilfe von entsprechenden MATLAB[®] Implementierungen untersucht. Für das OLA und WSOLA Verfahren sowie den Phase Vocoder

auf Basis der STFT (PV-STFT) wird auf die Implementierung von *Driedger et al.* aus der *TSM Toolbox* zurückgegriffen, [DM18; DM14b].

Der Roller-Algorithmus steht durch eine eigene Implementierung zur Verfügung. Das MATLAB[®] Projekt ist auf der beiliegenden CD zu finden.

6.1 Testsignale

Die Algorithmen werden mithilfe von vier Testsignalen mit unterschiedlicher Komplexität untersucht. Bei allen vier Signalen handelt es sich um E-Gitarrenschnitte, die (wie in Abschnitt 2.1 beschrieben) mithilfe des Audiointerfaces *Focusrite Scarlett 6i6 2nd Gen* aufgenommen wurden. Die folgende Abbildung 6.1 zeigt die zeitlichen Verläufe der vier Testsignale.

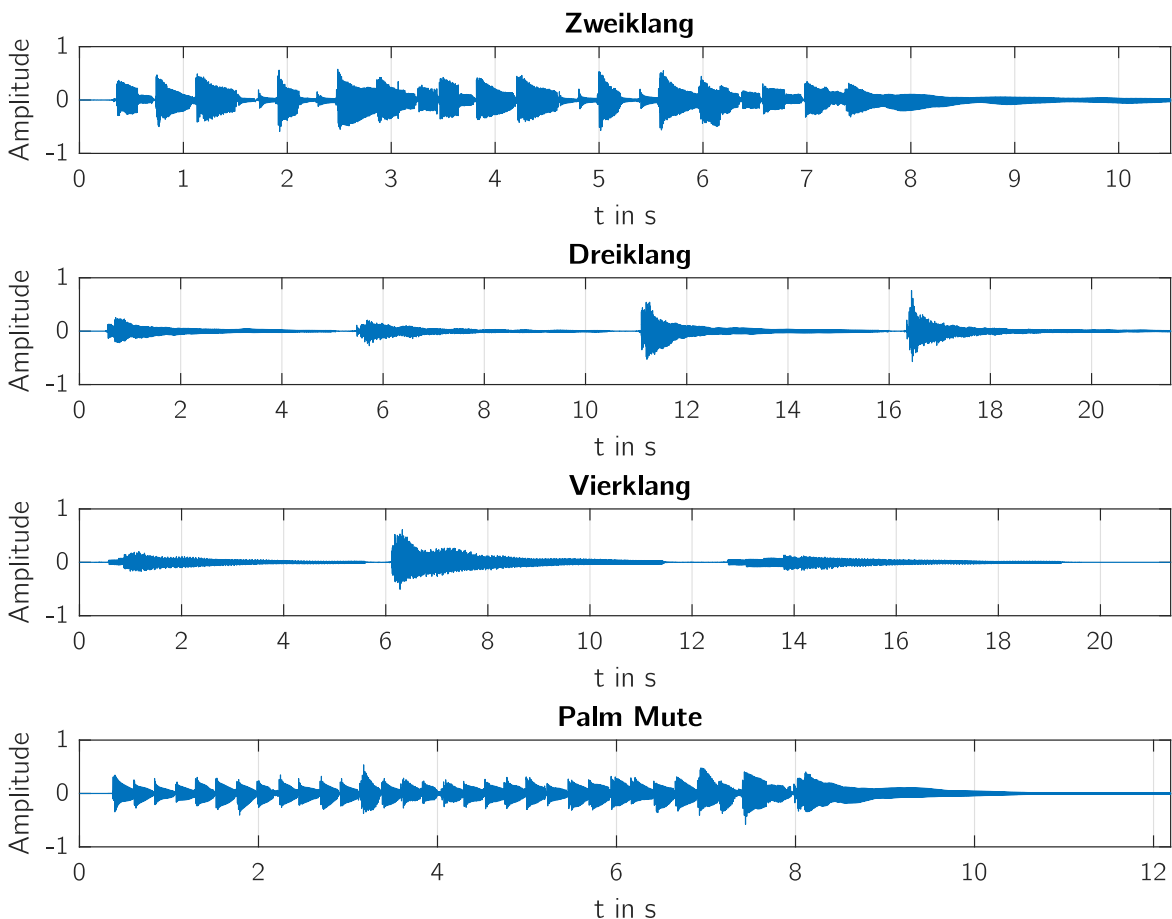


Abbildung 6.1: Vier Testsignale mit unterschiedlicher Komplexität

Die Testsignale sind so gewählt, dass an die Algorithmen unterschiedliche Herausforderungen gestellt werden, um die klanglichen Artefakte zu verdeutlichen. Die Signale unterscheiden sich insbesondere in Dynamik bzw. Anschlagstärke sowie der Anzahl von verschiedenen (gleichzeitig erklingenden) Tönen. Alle Signale wurden mit einem Plektrum eingespielt. Im Folgenden werden die Testsignale kurz beschrieben und charakterisiert:

Zweiklang

- Teilweise monophon bis schwach polyphon
- Starker Anschlag (hohe Transienten)
- Drei monophone Töne, gefolgt von schwach polyphonen Zweiklängen

Dreiklang

- Polyphon
- Offene G-Dur und A-Moll Dreiklänge
- Dynamikunterschiede durch zwei leichte Anschläge, gefolgt von zwei starken Anschlägen

Vierklang

- Stark polyphon
- E-Moll Septakkord über alle sechs Saiten (sechs Teiltöne)
- Dynamikunterschiede durch leichten und starken Anschlag (erster und zweiter Signalabschnitt)
- Übergang von monophon zu stark polyphon (dritter Signalabschnitt): Töne des Septakkords werden nacheinander angeschlagen und klingen ineinander aus

Palm Mute

- Schwach polyphon
- E-, D- und C-Powerakkord (Grundton, Quinte, Oktave)
- Gedämpfter Anschlag (Handgelenk auf Steg)
- Mittlere Anschlagstärke












Die Testsignale sind auf der beigefügten CD dokumentiert (siehe Anhang B).

6.2 Parameterwahl

Die Parameterwahl der Algorithmen entscheidet maßgeblich über die jeweils maximal erreichbare Klangqualität und Latenz. Wichtige Parameter sind beim OLA, WSOLA und PV-STFT Algorithmus die Blocklänge N und Synthese-Schrittweite H_S . Bei WSOLA kommt zudem der Toleranz-Parameter Δ_{max} hinzu. Wie beim Roller Algorithmus ist die passende Wahl der Parameter nur schwer analytisch zu bestimmen. Insbesondere im Bezug auf die gewünschte Latenzoptimierung ist es sinnvoll psychoakustische Aspekte auszunutzen, weshalb die Parameterwahl hinsichtlich des hörtechnischen Eindrucks optimiert wird. Beim OLA, WSOLA und PV-STFT Algorithmus werden jeweils drei unterschiedliche Varianten betrachtet, um einerseits die maximale Klangqualität und andererseits die niedrigste Latenz zu untersuchen. Der Roller Algorithmus wird mit den in Unterabschnitt 4.4.2 bestimmten Parametern verwendet.

Die folgende Tabelle 6.1 fasst die Parameterwahl zusammen. Die in der Tabelle dargestellten Farb- und Strichtypen dienen zur Unterscheidung der einzelnen Varianten in späteren Abbildungen.

Tabelle 6.1: Übersicht der gewählten Parameter der untersuchten Algorithmen

Algorithmus	Variante	Farbe	Parameter
OLA	1		$N = 128$ $H_S = 64$
	2		$N = 256$ $H_S = 128$
	3		$N = 512$ $H_S = 256$
WSOLA	1		$N = 512$ $H_S = 256$ $\Delta_{max} = 256$
	2		$N = 600$ $H_S = 300$ $\Delta_{max} = 300$
	3		$N = 1024$ $H_S = 512$ $\Delta_{max} = 512$
Roller	1		IIR Filterbank, siehe Unterabschnitt 4.4.2
PV-STFT	1		$N = 1024$ $H_S = 256$
	2		$N = 2048$ $H_S = 512$
	3		$N = 4096$ $H_S = 1024$
élastiquePro	1		nicht bekannt

6.3 Klangqualität

In diesem Abschnitt werden die Algorithmen hinsichtlich ihrer Klangqualität bewertet. Nach einer Beschreibung des hörtechnischen Eindrucks in Unterabschnitt 6.3.1 werden die Algorithmen in Unterabschnitt 6.3.2 anhand objektiver Beurteilungskriterien miteinander verglichen und beurteilt. Das Pitch Shifting wird bei allen Algorithmen in Halbtonschritten bis ± 1 Oktave betrachtet.

Die durch die Algorithmen berechneten Ausgangssignale liegen als Hörbeispiele auf der beigefügten CD (Anhang B) vor.

6.3.1 Hörtest

In diesem Unterabschnitt wird die erreichbare Klangqualität von jedem Algorithmus anhand des hörtechnischen Eindrucks beschrieben. Dabei wird insbesondere die jeweils maximal erreichbare Klangqualität und das Verhalten bei Reduzierung der Blocklänge diskutiert.

OLA

Beim OLA Verfahren ist die Neuordnung der Blöcke unabhängig vom Signal. Dieser einfache Ansatz führt zu erheblichen Problemen, da so die periodischen Strukturen der Gitarrensignale nicht erhalten bleiben. Selbst bei kleinen Skalierungsfaktoren und dem teilweise monophonen Testsignal *Zweiklang* führt dies zu Phasensprüngen, die das Signal verzerren. Kleinere oder größere Blocklängen (Variante 1 - 3) können die Phasensprünge nicht verhindern. Die Artefakte werden umso stärker, je größer die Tonhöhenverschiebung gewählt wird. Der Klang kann als unangenehm und metallisch beschrieben werden.

Das OLA Verfahren eignet sich nicht für die Tonhöhenverschiebung von Gitarrensignalen.

WSOLA

Das WSOLA Verfahren zeigt durch die variable Neuordnung der Blöcke deutlich bessere Ergebnisse. Bei dem weniger stark polyphonen Testsignal *Zweiklang* zeigen die zweite und dritte Variante über den gesamten Bereich keine hörbaren Artefakte und liegen im Vergleich zum kommerziellen Algorithmus *élastiquePro* gleich auf. Der

transientenreiche Klang bleibt erhalten. Bei der ersten Variante werden die monophonen Stellen ebenfalls gut wiedergegeben, allerdings entstehen bei den polyphonen Stellen starke Artefakte in Form von Phasensprüngen (ähnlich wie bei OLA).

Beim Testsignal *Palm Mute* treten bei Variante 2 beim D- und C-Powerakkord ebenfalls Phasensprünge auf. Der E-Powerakkord ist hingegen frei von Artefakten. Eine Erklärung hierfür ist, dass die Teiltöne des E-Akkords durch die höhere Tonhöhe spektral weiter auseinander liegen. Ein ähnliches Verhalten ist beim Testsignal *Dreiklang* zu hören. Während bei der ersten und zweiten Variante durchweg Phasensprünge entstehen, wird bei der Variante 3 der G-Dur Akkord gut in der Tonhöhe verschoben. Beim A-Moll Akkord treten hingegen wieder Phasensprünge auf.

Dass stark polyphone Signale beim WSOLA Algorithmus zu Problemen führen, zeigt sich abschließend noch einmal beim Testsignal *Vierklang*. Der leichte und starke Anschlag des gesamten Septakkords führt auch bei Variante 3 mit hoher Blocklänge zu starken Artefakten. Im dritten Signalabschnitt zeigt sich, dass die ersten drei Töne ohne Artefakte verschoben werden. Ab dem vierten Teilton das Signal allerdings zu komplex, sodass erneut Phasensprünge auftreten.

Insgesamt lässt sich festhalten, dass der WSOLA Algorithmus bei monophonen und schwach polyphonen Signalen (Zweiklänge und Powerakkorde) bei ausreichend hoher Blocklänge (Variante 3) gute Ergebnisse erzielt. Für stark polyphone Signale (Dreiklänge, Vierklänge etc.) ist das Verfahren jedoch nicht geeignet, da die periodischen Strukturen zerstört werden.

Roller

Beim Roller Algorithmus zeigt sich, dass selbst stark polyphone Signale wie das Testsignal *Vierklang* ohne Phasensprünge ausklingen. Dies stellt eine deutliche Verbesserung gegenüber anderen Algorithmen im Zeitbereich dar. Demgegenüber stehen jedoch erhebliche Phasenprobleme, die sich durch einen weniger direkten und dunkleren Klang äußern. Dieser Effekt fällt insbesondere bei dem transientenreichen Signal *Zweiklang* auf, welches selbst bei einer geringen Tonhöhenverschiebung verschmiert.

Des Weiteren sind in allen Signalen Amplitudenmodulationen zu hören, die durch die geringe Flankensteilheit der IIR Filter begründet sind. Frequenzanteile, die eigentlich nur durch das b -te Frequenzband erfasst werden dürfen, werden ebenfalls von den Nachbar-Frequenzbändern ($b - 1$) und ($b + 1$) erfasst. Die Frequency Shifter dieser Bänder führen jedoch eine leicht kleinere bzw. größere Frequenzverschiebung durch, weshalb die gleichen Frequenzanteile mehrfach unterschiedlich verschoben werden. Im Spektrum

des Ausgangssignals liegen diese anschließend mit hoher Amplitude nah beieinander. Dies führt zu einer hörbaren Amplitudenmodulation.

Der Roller Algorithmus erreicht bei einer moderaten Tonhöhenverschiebung im Bereich ± 4 Halbtönen eine akzeptable Klangqualität mit den zuvor genannten Artefakten. Der hörtechnische Eindruck zeigt, dass die Amplitudenmodulationen umso prägnanter werden, je größer die Tonhöhenverschiebungen nach oben gewählt wird. Die Frequenzanteile der Nachbar-Frequenzbänder ($b - 1 + s$) und ($b + 1 + s$) liegen nach dem Pitch Shifting durch die logarithmische Verteilung der Frequenzbänder weiter auseinander als bei kleinen Tonhöhenverschiebungen. Die Frequenz der Amplitudenmodulation wird daher mitskaliert und fällt durch die höhere Frequenz unangenehmer auf.

Insgesamt erreicht der Roller Algorithmus eine eher niedrige Klangqualität. Die Amplitudenmodulationen können höchstens im oberen Frequenzbereich reduziert werden, indem zwischen den Frequenzbändern gezielt Lücken eingeführt werden. Der dadurch entstehende Verlust an Signalenergie reduziert allerdings (wie in Unterabschnitt 4.4.2 beschrieben) die Klangqualität erheblich.

PV-STFT

Der PV-STFT Algorithmus ist in der Lage auch stark polyphone Signale wie das Testsignal *Vierklang* in der Tonhöhe zu verschieben. Es treten keine Phasensprünge auf, sodass die periodische Struktur des Eingangssignals erhalten bleibt. Dies gelingt allerdings nur, wenn die Blocklänge ausreichend groß gewählt wird. Während die dritte Variante alle Testsignale zuverlässig in der Tonhöhe verschiebt, kommt es bei der ersten Variante bei allen Testsignalen zu erheblichen Phasensprüngen. Die horizontale Phasenkohärenz kann durch die geringe Blocklänge und damit geringe Frequenzauflösung nicht sichergestellt werden. Die zweite Variante stellt die untere Grenze dar. Die horizontale Phasenkohärenz kann jedoch auch bei dem schwach polyphonen Testsignal *Zweiklang* nicht immer sichergestellt werden. Bei den stärker polyphonen Signalen *Dreiklang* und *Vierklang* nimmt die Klangqualität durch das dichtere Spektrum weiter ab.

Im Gegensatz zum WSOLA Algorithmus ist ein Verschmieren der Transienten wahrzunehmen, welches zu einem weniger direkten Klang führt. Dieser Artefakt wird beim Phase Vocoder in zahlreichen Veröffentlichungen beschrieben und ist durch den leichten Verlust der vertikalen Phasenkohärenz (trotz Identiy Phase Locking) begründet.

élastiquePro

Der kommerzielle Algorithmus *élastiquePro* vereint die gute Darstellung der Transienten des WSOLA Algorithmus mit dem Erhalt der horizontalen Phasenkohärenz bei stark polyphonen Signalen (ähnlich wie beim Phase Vocoder, Variante 3). Wie zu erwarten, bietet der Algorithmus in diesem Vergleich die höchste Klangqualität.

Auch wenn der Erhalt der Transienten und horizontalen Phasenkohärenz gegeben ist, klingen die in der Tonhöhe verschobenen Signale, insbesondere bei großen Tonhöhenverschiebungen, künstlich. Auf der CD (Anhang B) liegen hierzu beispielhaft zwei Audio-beispiele vor. Die Datei `palmMute_10semitone_elastiquePro.wav` enthält das durch *élastiquePro* um zehn Halbtöne nach oben verschobene Testsignal *Palm Mute*. Bei der zweiten Datei `palmMute_10Semitone_recorded.wav` wurde die Akkord-Abfolge direkt zehn Halbtöne höher eingespielt und aufgenommen. Der Höreindruck zeigt, dass die Tonhöhe zwar identisch ist, sich die Klangfarben jedoch erheblich voneinander unterscheiden. Dieses Beispiel verdeutlicht noch einmal die in Abschnitt 3.2 beschriebene Problematik. Die spektrale Hüllkurve bzw. Formanten verhalten sich über das gesamte Griffbrett stark dynamisch und werden auch durch den klanglich optimierten Algorithmus *élastiquePro* nicht berücksichtigt.

6.3.2 Objektive Audiomerkmale

In diesem Unterabschnitt werden die Algorithmen anhand der folgenden objektiven Beurteilungskriterien bzw. Audiomerkmale verglichen:

- Nulldurchgangsrate
- Spektraler Schwerpunkt
- Spektrale Entropie
- Spektrale Streuung

Es wird diskutiert, inwieweit die Klangqualität und die beim Pitch Shifting auftretenden Artefakte mithilfe von Audiomerkmale untersucht und beurteilt werden können. Die meisten der hier untersuchten Merkmale werden blockweise über das Signal berechnet, sodass der zeitliche Verlauf des Merkmals in Form eines Vektors dargestellt wird. Um die Komplexität zu reduzieren und den Vergleich zwischen den Algorithmen zu erleichtern, werden von den Merkmalen die arithmetischen Mittelwerte berechnet. Die grafische Darstellung der Ergebnisse ist in Anhang A zu finden. Die letzten vier Audiomerkmale

werden mithilfe der *Audio Toolbox* von MATLAB[®] berechnet. Für die FFT wird eine Blocklänge von $N = 2048$ und eine Analyse-Schrittweite von $H_A = 512$ gewählt.

Nulldurchgangsrate (Zero-Crossing-Rate (ZCR))

Die Nulldurchgangsrate (englisch *Zero-Crossing-Rate (ZCR)*) gibt die Frequenz an, mit der das Signal die x-Achse schneidet und kann folgendermaßen berechnet werden [RB19]:

$$\text{ZCR} = \left(\sum_{n=0}^L |\text{sign}[x(n)] - \text{sign}[x(n-1)]| \right) \cdot \frac{F_s}{4L} \quad (6.1)$$

Die Berechnung ist so ausgelegt, dass bei einem reinen Sinussignal die ZCR der Frequenz des Sinus entspricht.

Es ist zu erwarten, dass die ZCR beim Pitch Shifting nach unten kleiner und beim Pitch Shifting nach oben größer wird. In den Abbildungen A.1 bis A.4 ist zu erkennen, dass dies im Allgemeinen für jeden Algorithmus und jede Variante zutrifft. Größere Abweichungen zeigen sich beim OLA Algorithmus, die sich durch die feste Neuordnung der Blöcke begründen lassen. Eine absolute Aussage über die Klangqualität lässt sich aus der ZCR allerdings nicht ableiten. Dies ist insbesondere in Abbildung A.4 ersichtlich, da dort die Kurven von Roller und *élastiquePro* beinahe übereinander liegen. *élastiquePro* weißt jedoch (nach dem oberen Hörtest) eine deutlich besser Klangqualität auf.

Spektraler Schwerpunkt (Spectral Centroid)

Der Spectral Centroid beschreibt den spektralen Schwerpunkt und berechnet sich nach [Mat20] durch:

$$\mu_1 = \frac{\sum_{k=b_1}^{b_2} F(k) S_{XX}(k)}{\sum_{k=b_1}^{b_2} S_{XX}(k)} \quad (6.2)$$

$F(k)$ bezeichnet die Frequenz der k -ten Frequenzstützstelle. $S_{XX}(k)$ entspricht dem Leistungsdichtespektrum des Signals x . b_1 sowie b_2 sind die Indizes über den ausgewerteten Frequenzbereich (hier $[0, F_s/2]$). Es ist zu erwarten, dass der spektrale Schwerpunkt beim Pitch Shifting nach unten tiefer und beim Pitch Shifting nach oben höher

liegt. Zudem lässt sich aus dem Spectral Centroid ableiten wie hell der Klang eines Signals ist.

Die berechneten Mittelwerte sind in den Abbildungen A.5 bis A.8 gegenübergestellt. Wie beim ZCR sind beim OLA Algorithmus über alle Testsignale hinweg größere Abweichungen von den anderen Algorithmen zu beobachten. Der erwartete Verlauf tritt bei den verbleibenden Algorithmen ein. Der Roller Algorithmus liegt, abgesehen vom Testsignal *Vierklang*, unterhalb der anderen Kurven. Dies ist auf die Lücken zwischen den Frequenzbändern im oberen Frequenzbereich zurückzuführen (siehe Abbildung 4.16) und stimmt mit dem im Hörtest festgestellten dunkleren Klang überein. Einen Rückschluss auf Phasensprünge und auf die allgemeine Klangqualität erlaubt dieses Merkmal jedoch nicht.

Spektrale Entropie (Spectral Entropy)

Mithilfe der spektralen Entropie kann eine Aussage darüber getroffen werden, ob im Signal viele Peaks vorhanden sind. Sie berechnet sich nach [Mat20] durch:

$$\text{entropy} = \frac{- \sum_{k=b_1}^{b_2} S_{XX}(k) \cdot \log_2(S_{XX}(k))}{\log_2(b_2 - b_1)} \quad (6.3)$$

Die spektrale Entropie geht für ein reines Sinussignal gegen null. Bei einem Dirac-Impulskamm hingegen erreicht die Entropie bei jedem Impuls den maximalen Wert eins. Es ist zu erwarten, dass die Tonhöhe keinen Einfluss auf die spektrale Entropie hat. Vielmehr sind niedrigere Werte als Verlust der vertikalen Phasenkohärenz, also als Abschwächung der Transienten, zu interpretieren.

Die berechneten Mittelwerte sind in den Abbildungen A.9 bis A.12 aufgezeigt. Es fällt auf, dass die spektrale Entropie beinahe konstant bleibt. Auffällig ist, dass die Entropie bei der ersten Variante des PV-STFT bei einer starken Tonhöhenverschiebung sinkt. Als ein Maß für die im Signal auftretenden Phasensprünge kann dies allerdings nicht gewertet werden, da die erste Variante des WSOLA Algorithmus (die ähnliche Probleme aufweist), insbesondere beim Pitch Shifting nach unten, eine unauffällige Entropie zeigt.

Beim Roller Algorithmus ist deutlich erkennbar, dass die spektrale Entropie bei allen Testsignalen weit unterhalb der anderen Algorithmen liegt. Dies lässt einen Rückschluss auf den deutlich wahrnehmbaren, weniger direkteren Klang zu, der durch die Phasenprobleme und Lücken zwischen den oberen Frequenzbändern begründet ist.

Zusammenfassend lässt sich jedoch festhalten, dass eine absolute Aussage über die Klangqualität mithilfe dieses Merkmals ebenfalls nicht möglich ist.

Spektrale Streuung (Spectral Spread)

Die spektrale Streuung gibt die Konzentration des Spektrums um den spektralen Schwerpunkt an und berechnet sich nach [Mat20] durch:

$$\mu_2 = \sqrt{\frac{\sum_{k=b_1}^{b_2} (F(k) - \mu_1)^2 S_{XX}(k)}{\sum_{k=b_1}^{b_2} S_{XX}(k)}} \quad (6.4)$$

Es ist zu erwarten, dass die spektrale Streuung durch die Frequenzskalierung beim Pitch Shifting nach unten abnimmt und beim Pitch Shifting nach oben zunimmt, da die Frequenzanteile weiter zusammen- bzw. auseinanderrücken. Da Transienten das Spektrum breit anregen (vergleiche Abbildung 2.1) ist zudem davon auszugehen, dass beim Verlust der vertikalen Phasenkohärenz die spektrale Streuung sinkt.

Die berechneten Mittelwerte sind in den Abbildungen A.13 bis A.16 dargestellt. Der erwartete steigende Verlauf trifft grundsätzlich für alle Algorithmen zu. Beim OLA Algorithmus zeigen sich jedoch wieder größere Abweichungen von den übrigen Algorithmen, die durch die feste Neuordnung der Blöcke begründet sind. Der Roller Algorithmus liegt bei den Testsignalen *Zweiklang*, *Dreiklang* und *Palm Mute* unterhalb der anderen Algorithmen. Dies ist auf die Lücken zwischen den oberen Frequenzbändern zurückzuführen. Die Algorithmen WSOLA, PV-STFT und *élastiquePro* liegen nah beieinander. Eine Aussage über die Klangqualität und insbesondere den Erhalt der vertikalen Phasenkohärenz oder Beurteilung von möglichen Phasensprüngen gelingt mit diesem Audiomerkmals nicht.

6.4 Latenz

Im vorherigen Abschnitt wurden die Algorithmen hinsichtlich ihrer Klangqualität beurteilt. Insbesondere Unterabschnitt 6.3.1 zeigt, dass die Algorithmen nicht für jeden Signaltyp geeignet sind und die Blocklänge erheblichen Einfluss auf das Ergebnis der Tonhöhenverschiebung hat. In diesem Abschnitt wird die erreichbare Latenz in Verbindung mit den zuvor erzielten Erkenntnissen hinsichtlich der Klangqualität diskutiert. Im Folgenden sind die geeigneten Algorithmen und Varianten zusammengefasst:

- WSOLA, Variante 2: monophone Signale
- WSOLA, Variante 3: monophone und schwach polyphone Signale
- Roller: sämtliche Signaltypen im Bereich $s \pm 4$ Halbtöne
- PV-STFT, Variante 2: monophone und schwach polyphone Signale
- PV-STFT, Variante 3: stark polyphone Signale

Die Latenzen entstehen bei den Algorithmen insbesondere durch die blockbasierten Ansätze oder durch Filterungen des Signals. Weitere Latenzen können beispielsweise durch die Berechnung einer Kreuzkorrelation oder Fourier-Transformation entstehen. In Folgenden werden die zu erwartenden Latenzen für jeden Algorithmus analysiert und berechnet. Aufgrund der fehlenden Implementierung auf Hardware kann jedoch eine exakte Messung der Latenz nicht durchgeführt werden.

WSOLA

Die Latenz des WSOLA Algorithmus hängt durch den blockbasierten Ansatz hauptsächlich von der gewählten Blocklänge N ab. Für eine optimale Überlappung der Blöcke setzt das Verfahren zudem auf einen zeitlichen Toleranzbereich, in dem nach der maximalen Ähnlichkeit gesucht wird. Die Größe der zeitlichen Toleranz beeinflusst die resultierende Latenz ebenso. Dies wird anhand der folgenden Abbildung 6.2 und den extremen Tonhöhenverschiebungen von +1 Oktave ($\alpha = 2$) und -1 Oktave ($\alpha = 0.5$) verdeutlicht.

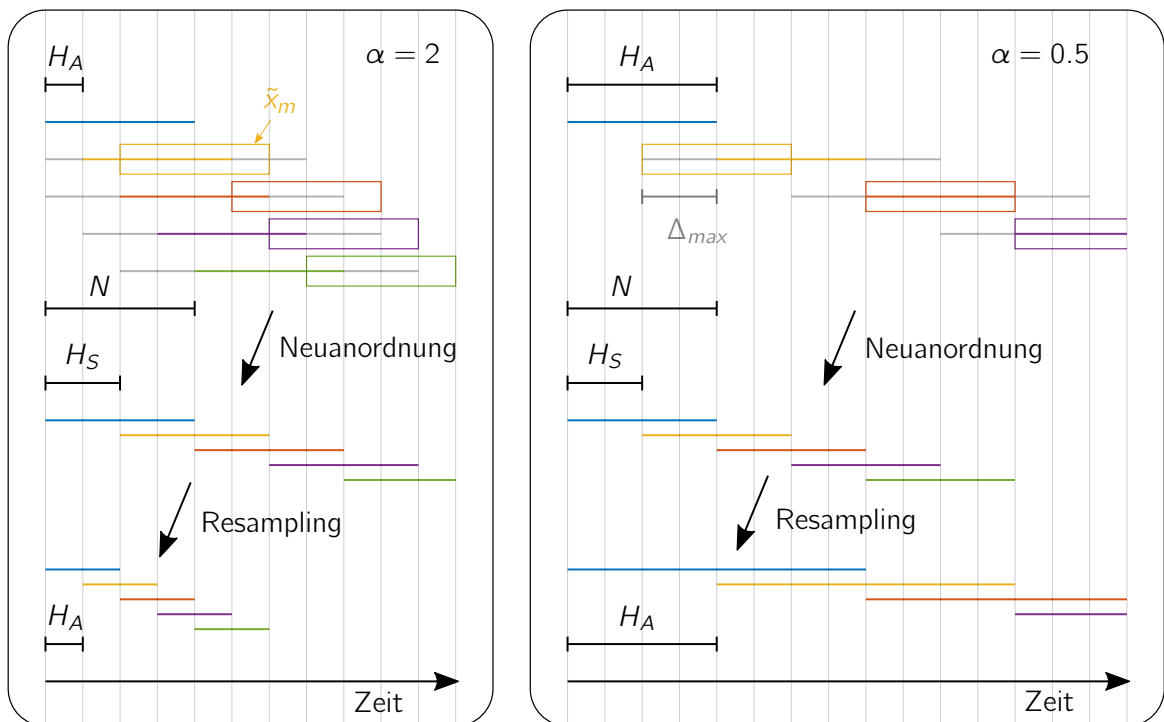


Abbildung 6.2: WSOLA Algorithmus: Betrachtung der maximalen Latenz bei unterschiedlicher Tonhöhenverschiebung

Es wird jeweils der grobe Ablauf des Algorithmus mit Zerlegung des Eingangssignals in Blöcke der Länge N im Abstand H_A , Neuanordnung der Blöcke im Abstand H_S und Resampling auf die ursprüngliche Signallänge betrachtet. Bei jedem Eingangsblock wird die zeitliche Toleranz in grauer Farbe angedeutet. Die farbigen Kästen entsprechen den Signalabschnitten, die eine perfekte Fortsetzung des bisher synthetisierten Signals bedeuten würden. Um die maximale Latenz zu untersuchen, wird immer der rechte Toleranzbereich voll ausgenutzt, sodass die farbigen Kästen maximal weit rechts liegen.

Bei $\alpha = 2$ zeigt sich, dass die farbigen Kästen über den zeitlichen Toleranzbereich ragen. Für die Bestimmung der maximalen Ähnlichkeit muss daher zusätzlich der vom Kasten eingeschlossene Signalabschnitt bekannt sein, der um $(H_S - H_A)$ weiter liegt. Für $\alpha > 1$ ergibt sich daher die folgende maximale Latenz:

$$\tau_{max}^{\alpha > 1} = \frac{N + \Delta_{max} + H_S - H_A}{F_s} \quad (6.5)$$

Diese maximale Latenz tritt allerdings nur auf, wenn bei der Bestimmung der maximalen Ähnlichkeit der rechte Toleranzbereich maximal ausnutzt wird. In allen anderen Fällen ist die Latenz kleiner, aber immer größer als $(N + \Delta_{max})$. Insgesamt ist die Latenz also dynamisch und bei $\alpha = 2$ maximal.

Bei $\alpha = 0.5$ zeigt sich, dass die farbigen Kästen nie über den rechten Toleranzbereich ragen können. Die Latenz ist statisch und ergibt sich durch:

$$\tau_{max}^{\alpha < 1} = \frac{N + \Delta_{max}}{F_s} \quad (6.6)$$

Für monophone Signale (Variante 2) ergeben sich daher die folgenden maximalen Latenzen:

$$\tau_{max}^{\alpha=2} = \frac{600 + 300 + 300 - 150}{48 \text{ kHz}} = 21.875 \text{ ms} \quad (6.7)$$

$$\tau_{max}^{\alpha < 1} = \frac{600 + 300}{48 \text{ kHz}} = 18.75 \text{ ms} \quad (6.8)$$

Für monophone und schwach polyphone Signale (Variante 3) sind die Latenzen entsprechend höher:

$$\tau_{max}^{\alpha=2} = \frac{1024 + 512 + 512 - 256}{48 \text{ kHz}} = 37.3 \text{ ms} \quad (6.9)$$

$$\tau_{max}^{\alpha < 1} = \frac{1024 + 512}{48 \text{ kHz}} = 32 \text{ ms} \quad (6.10)$$

Die hier berechneten Latenzen entstehen zwangsläufig durch den blockbasierten Ansatz und sind daher unabhängig von der gewählten Hardware. Weitere Latenzen entstehen durch die notwendigen Berechnungsschritte des Algorithmus. Insbesondere die Kreuzkorrelation und das Resampling zählen zu den rechenintensiven Operationen. Wie

schnell alle Berechnungsschritte durchgeführt werden, hängt stark von der verwendeten Hardware und Implementierung ab. Der ausschlaggebende Teil der Gesamtlatenz ist allerdings mit Gleichung 6.5 und 6.6 gegeben.

Roller

Der Roller Algorithmus unterscheidet sich durch seinen samplebasierten Ansatz vom WSOLA und PV-STFT Algorithmus. Latenzen entstehen insbesondere bei der Filterung durch die IIR Filterbank und bei den SSB Modulationen durch die IIR Brückenfilter. Durch die nichtlinearen Phasengänge der IIR Filter sind die Gruppenlaufzeiten allerdings frequenzabhängig. Die maximale Höhe der Gruppenlaufzeit hängt darüber hinaus von der Breite des Durchlassbereichs ab. Bei der Filterbank steigt die Breite des Durchlassbereichs zu den hohen Frequenzen an, weshalb die Gruppenlaufzeiten im tiefen Frequenzbereich am höchsten sind und in Richtung der hohen Frequenzen abfallen. Dies ist in der folgenden Abbildung 6.3 dargestellt:

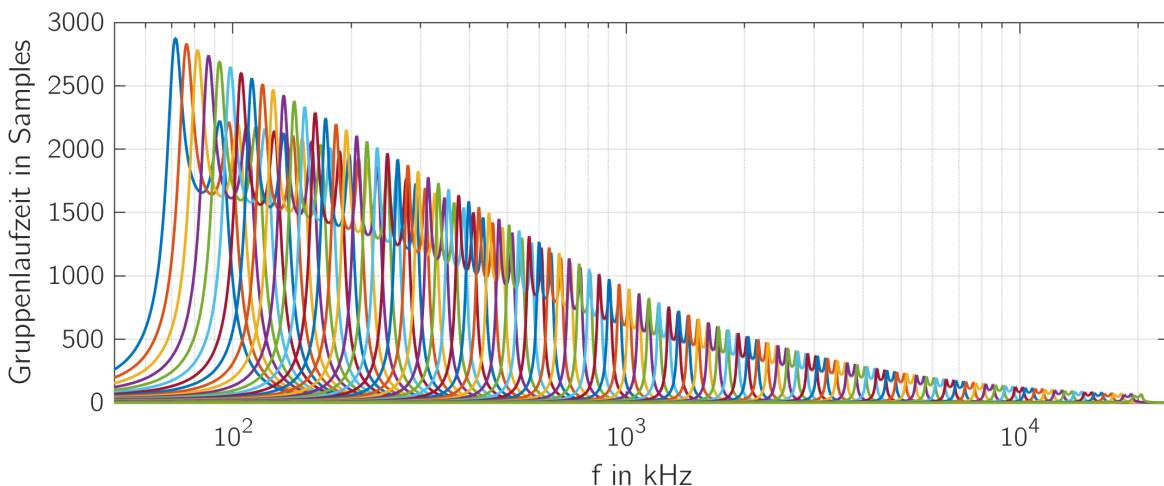


Abbildung 6.3: IIR Filterbank: Gruppenlaufzeiten der einzelnen Bandpässe

Hohe Frequenzen durchlaufen die IIR Filterbank daher deutlich schneller als tiefe Frequenzen. Durch das IIR Brückenfilter entstehen in den einzelnen Frequenzbändern weitere Latenzen. Die Abbildung 6.4 zeigt die Gruppenlaufzeit des Frequency Shifters ohne SSB Modulation ($s = 0$), da bei $s \neq 0$ ein nichtlinearer Prozess vorliegt.

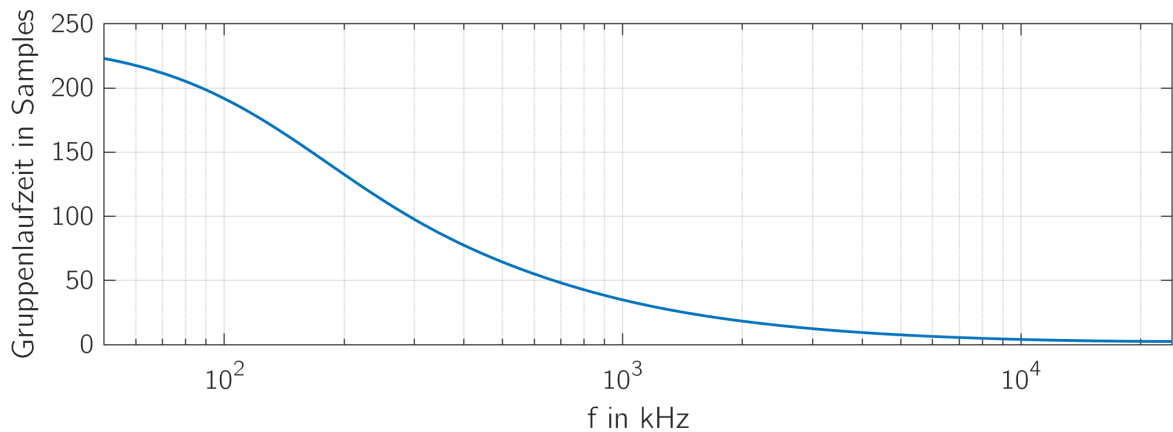


Abbildung 6.4: Gruppenlaufzeit des IIR Brückenfilters ($s = 0$)

Die Gruppenlaufzeit ist ebenfalls für die tiefen Frequenzen am höchsten. Um die Latenz des Gesamtsystems abzuschätzen, wird in Abbildung 6.5 ein Dirac-Impuls als Eingangssignal gewählt. Die Tonhöhenverschiebung wird in diesem Beispiel mit $s = -1$ durchgeführt. Die unteren beiden Teilabbildungen zeigen das Ausgangssignal und damit die Antwort auf den Impuls.

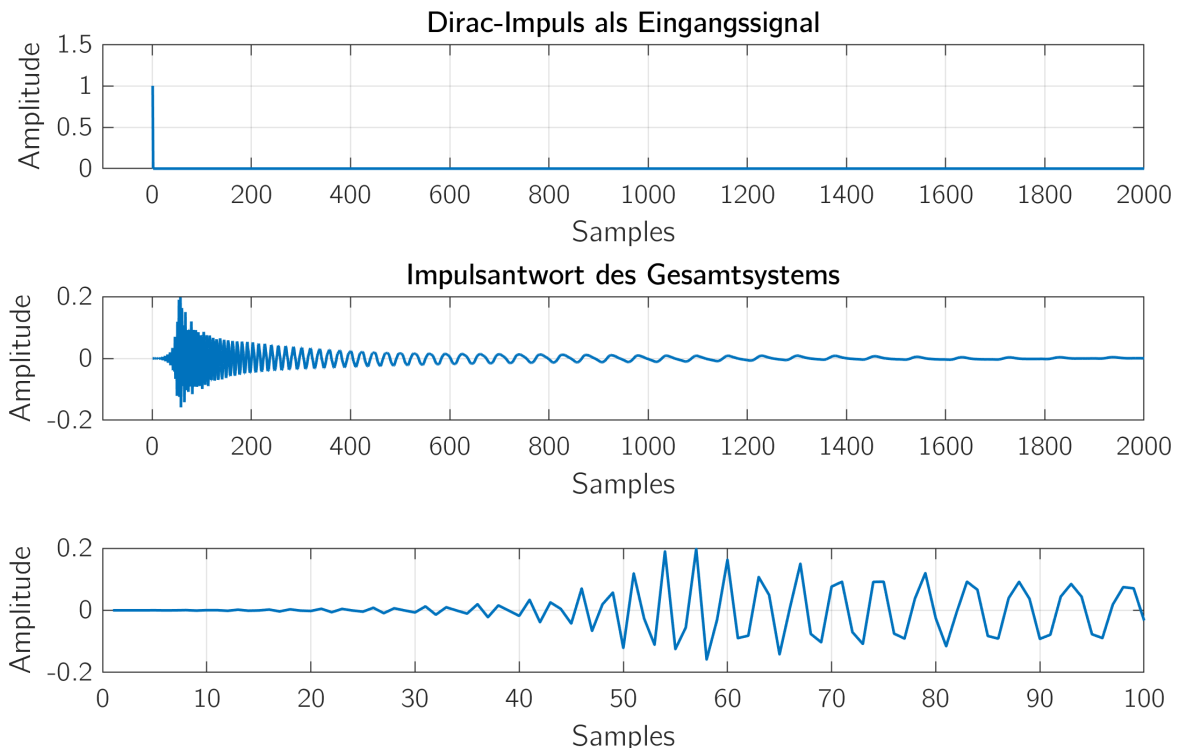


Abbildung 6.5: Anregung des Systems mit einem Dirac-Impuls ($s = -1$)

In der untersten Teilabbildung ist zu sehen, dass der Impuls um ca. 50 Samples verzögert am Ausgang erscheint. Der Impuls ist jedoch stark verschmiert und schwingt lange aus. Diese beiden Auswirkungen sind eine Folge aus den unterschiedlich hohen Gruppenlaufzeiten bzw. nichtlinearen Phasengängen. Dadurch, dass die Latenz für hohe Frequenzen am niedrigsten und für niedrige Frequenzen am höchsten ist, klingt der Dirac-Impuls ähnlich wie ein kurzer Sinus-Sweep mit fallender Frequenz. Dies ist ebenfalls die Ursache für den seltsamen Klang bei starken Transienten.

Es ergibt sich somit keine einheitliche Gesamtlatenz für den Roller Algorithmus. Die geringe Verzögerung von sehr hohen Frequenzen (> 10 kHz) führt zu der in Abbildung 6.5 dargestellten schnellen Reaktion auf den Dirac-Impuls. 50 Samples entsprechen bei einer Abtastfrequenz von 48 kHz einer Latenz von ca. 1 ms. Zu den tiefen Frequenzen steigt die Latenz allerdings durch die hohe Gruppenlaufzeit auf ca. 3000 Samples bzw. 62.5 ms an. Die Untersuchung der spektralen Bandbreite in Abschnitt 2.1 zeigt zudem, dass bei Gitarrensinalen die spektrale Energie hauptsächlich in den tieferen Frequenzbereichen verteilt ist. Der Grundton und die Obertöne der tiefen E-Saite (siehe Abbildung 2.1) liegen im besonders kritischen Frequenzbereich.

Die unterschiedlichen Gruppenlaufzeiten führen somit nicht nur zu einer deutlichen Verschlechterung der Klangqualität, sondern auch zu einer dynamischen Latenz, die von der gespielten Note abhängt. Es ist möglich, dass sich die geringere Gruppenlaufzeit im hohen Frequenzbereich positiv auf die wahrgenommene (psychoakustische) Latenz auswirkt.

PV-STFT

Die Latenz des PV-STFT Algorithmus hängt, durch die Transformation in den Frequenzbereich mithilfe der FFT, von der Blocklänge N ab. In der Abbildung 6.6 ist der grobe Ablauf des Algorithmus für die maximalen Skalierungsfaktoren $\alpha = 2$ und $\alpha = 0.5$ dargestellt. Das Prinzip ist ähnlich zum WSOLA Algorithmus, allerdings entfällt der zeitliche Toleranzbereich, da der Erhalt der horizontalen Phasenkohärenz im Frequenzbereich sichergestellt wird. H_S entspricht hier immer einem Viertel der Blocklänge.

Am Beispiel des roten Signalabschnitts wird deutlich, dass der Signalabschnitt des Ausgangssignals erst dann zur Verfügung steht, wenn der entsprechende Signalabschnitt der Länge N des Eingangssignals bekannt ist. Die Latenz ist somit unabhängig von α und berechnet sich durch:

$$\tau = \frac{N}{F_s} \tag{6.11}$$

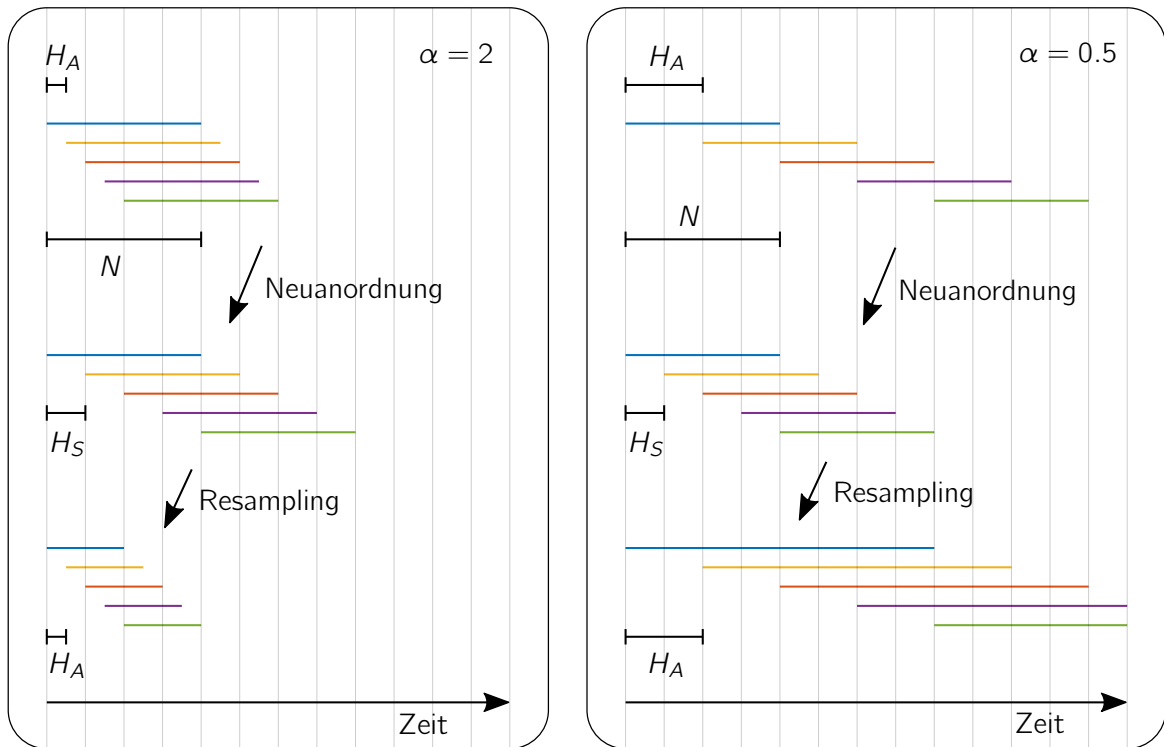


Abbildung 6.6: PV-STFT Algorithmus: Betrachtung der Latenz bei unterschiedlicher Tonhöhenverschiebung

Für monophone und schwach polyphone Signale (Variante 2) ergibt sich eine Latenz von:

$$\tau = \frac{2048}{48 \text{ kHz}} = 42.7 \text{ ms} \quad (6.12)$$

Für stark polyphone Signale (Variante 3) ist die Latenz dementsprechend mit

$$\tau = \frac{4096}{48 \text{ kHz}} = 85.3 \text{ ms} \quad (6.13)$$

doppelt so hoch. Ähnlich wie beim WSOLA Algorithmus sind diese Latenzen unabhängig von der gewählten Hardware. Weitere Latenzen entstehen durch die Berechnungsschritte des Algorithmus, wobei insbesondere die FFT und IFFT, die Anpassung des Phasenspektrums mithilfe des Phase Vocoders und das Resampling zu nennen sind. Ähnlich wie bei WSOLA gilt jedoch, dass der ausschlaggebende Teil der Gesamtlatenz durch Gleichung 6.11 gegeben ist.

7 Schlussbemerkung

7.1 Fazit

In dieser Thesis wurden die Funktionsweise und Klangqualität von verschiedenen klassischen sowie neuartigen Pitch Shifting Algorithmen im Bezug auf die polyphonen Signale einer E-Gitarre untersucht und gegenübergestellt. Die Tonhöhenverschiebung kann sowohl im Zeit- als auch Frequenzbereich berechnet werden, wobei in beiden Bereichen bereits zahlreiche Ansätze veröffentlicht wurden. In der Literatur wurde der Fokus jedoch überwiegend auf die Verbesserung der Klangqualität und weniger auf die Verbesserung der Latenz gelegt. Ein besonderes Augenmerk galt in dieser Arbeit daher der Optimierung der Latenz und den damit einhergehenden Auswirkungen auf die Klangqualität.

Die Untersuchungen zeigen, dass die erreichbare Klangqualität eines Pitch Shifting Algorithmus stark von der Struktur des Eingangssignals abhängt. Blockbasierte Algorithmen im Zeitbereich, wie das WSOLA Verfahren, erzielen bei monophonen und schwach polyphonen Signalen eine hohe Klangqualität mit einer guten Darstellung von Transienten. Bei stark polyphonen Signalen scheitert das Verfahren jedoch, da die periodischen Strukturen der Gitarrensiknale zerstört werden. Die Untersuchung der Latenz zeigt, dass die Blocklänge nur geringfügig verringert werden kann, da ansonsten keine optimalen Überlappungspunkte bei der Neuordnung der Blöcke existieren und Phasensprünge im Signal unvermeidbar sind. Eine Reduzierung auf monophone Signale führt zu einer minimalen Latenz von ca. 20 ms.

Für stark polyphone Signale eignet sich hingegen der Phase Vocoder, der eine Transformation des Signals in den Frequenzbereich voraussetzt und die Informationen der Phasenspektren zur Bestimmung der Momentanfrequenzen gewinnbringend einsetzt. Mit der CQT wurde eine alternative Zeit-Frequenz-Darstellung untersucht, die sich durch die logarithmische Verteilung der Frequenzstützstellen von der STFT (äquidistante Verteilung) unterscheidet und sich dadurch besonders für musikalische Signale eignet. Die Untersuchung der echtzeitfähigen Implementierung (sliCQ) zeigte jedoch, dass das Prinzip der logarithmischen Verteilung und der damit hohen Frequenzauflösung im tiefen Frequenzbereich, bei der in dieser Arbeit anvisierten Latenz, scheitert und die

Vorteile gegenüber der STFT verloren gehen. Das Unschärfe-Prinzip stellt eine physikalische Grenze dar, weshalb auch keine leistungsfähigere Hardware Abhilfe schafft. Durch die Reduzierung der Blocklänge hinsichtlich der anvisierten Latenz von 10 ms sind im Spektrum mehrere Peaks nicht mehr voneinander unterscheidbar, sodass die Korrektur der Phasenspektren mithilfe des Phase Vocoders misslingt. Bei schwach polyphonen Gitarrensingen kann eine Latenz von ca. 43 ms und bei stark polyphonen Signalen eine Latenz von ca. 85 ms erreicht werden. Beide Varianten eignen sich nicht für einen Pitch Shifter im Rahmen einer Live-Anwendung.

Mit dem Roller Algorithmus wurde ein weiteres Verfahren im Zeitbereich untersucht, das sich durch den samplebasierten Ansatz von den anderen Algorithmen unterscheidet. Während die Realisierung der SSB Modulation mithilfe eines IIR Brückenfilters gut umzusetzen ist, ist der Aufbau einer logarithmischen Filterbank mit erheblichen Einschränkungen verbunden. Die im tiefen Frequenzbereich notwendige, hohe Flankensteilheit der Bandpassfilter steht im Konflikt mit der Latenzoptimierung. FIR Filter zeigen durch die konstante Gruppenlaufzeit Vorteile im Bezug auf die Klangqualität, allerdings ist die niedrige Bandbreite und hohe Flankensteilheit im tiefen Frequenzbereich mit einer entsprechend hohen Filterordnung verbunden, die zwangsläufig zu einer Latenz von mehr als 200 ms führt. Mit IIR Filtern kann die Filterordnung und der Rechenaufwand zwar deutlich reduziert werden, allerdings führt die frequenzabhängige Gruppenlaufzeit zu einer deutlichen Verschlechterung der Klangqualität. Um die Gruppenlaufzeiten gering zu halten, müssen zudem Abstriche bei der Bandbreite und Flankensteilheit (niedrige Filterordnung) in Kauf genommen werden. Die unter diesen Aspekten konstruierte Filterbank führt zu einer weiteren Reduzierung der Klangqualität. Hohe Frequenzen werden durch die Filterbank zwar kaum verzögert (ca. 1 ms), gleichzeitig betragen jedoch die Gruppenlaufzeiten im tiefen Frequenzbereich weit über 50 ms. Eine weitere Reduzierung der Latenz führt zwangsläufig zu einem verstimmten Klang und noch stärker ausgeprägten Amplitudenmodulationen.

Bei der Beurteilung der Klangqualität zeigt sich, dass die Artefakte der Algorithmen nur schwierig mit objektiven Audiomeasurements darzustellen sind. Insbesondere der Verlust von horizontaler und vertikaler Phasenkohärenz lässt sich anhand der hier untersuchten Merkmale nicht nachweisen. Für die Bewertung von Pitch Shifting Algorithmen ist ein Hörtest unabdingbar.

Zusammenfassend komme ich zu dem Ergebnis, dass nicht jedes Verfahren für die stark polyphonen Signale einer E-Gitarre geeignet ist. Auch ohne Latenzbeschränkung sind die hier untersuchten, auf Overlap-Add basierten, Algorithmen im Zeitbereich nicht in der Lage die periodischen Strukturen stark polyphoner Signale zu erhalten. Andere Algorithmen, wie der Phase Vocoder oder Roller Algorithmus, können diese Signaltypen zwar ohne Phasensprünge in der Tonhöhe verschieben. Die Optimierung der Latenz hinsichtlich der Wahrnehmungsgrenze von 10 ms führt jedoch auf unterschiedliche Weise

zu klanglichen Artefakten, die den praktischen Nutzen eines solchen Pitch Shifters einschränken. Die in dieser Arbeit gesammelten Erkenntnisse verdeutlichen insbesondere den durch das Unschärfe-Prinzip vorgegebenen limitierenden Faktor bei der Signalverarbeitung.

7.2 Ausblick

Von den vorgestellten Algorithmen kann insbesondere der Roller Algorithmus weiter optimiert werden. Durch die Frequenzverschiebungen entstehen Phasenprobleme, die auch bei der Verwendung einer FIR Filterbank auftreten. Die Behebung dieser Ursache würde die Klangqualität besonders bei transientenreichen Signalen verbessern. Durch die Realisierung als Multiraten-System wäre zudem die Reduzierung der maximalen Gruppenlaufzeiten in Richtung tiefer Frequenzen denkbar, um diese über den gesamten Frequenzbereich konstanter zu halten. Dies würde neben der wahrgenommenen Gesamtlatenz ebenfalls die Klangqualität verbessern.

Pitch Synchronous Overlap-Add (PSOLA) stellt einen weiteren blockbasierten Algorithmus im Zeitbereich dar, bei dem im Signal nach Pitch Perioden gesucht wird. Die Zerlegung des Signals erfolgt anschließend nach diesen Pitch Markierungen. Interessant wäre hier ein direkter Vergleich mit dem WSOLA Verfahren, um die Grenzen von Overlap-Add basierten Algorithmen weiterführend zu analysieren.

Bei der Anwendung innerhalb einer DAW spielt die Latenz keine kritische Rolle, weshalb hier deutlich komplexere Berechnungen möglich sind. Die Untersuchungen in Abschnitt 3.2 zeigen, dass sich die spektrale Hüllkurve bzw. die Formanten dynamisch über das Gitarren-Griffbrett verändern. Dies wird selbst beim kommerziellen Algorithmus *élastiquePro* nicht beachtet. Das Miteinbeziehen der variierenden Hüllkurven würde die Klangqualität hinsichtlich der Natürlichkeit deutlich verbessern. Dies erfordert allerdings Vorkenntnisse über das individuelle Instrument. Denkbar wäre das Anlernen des Algorithmus mit Signalen über das gesamte Griffbrett, um das Verhalten bezüglich der variierenden spektralen Hüllkurven zu analysieren und abzuspeichern. Wird die Lage der gespielten Noten auf dem Griffbrett erkannt, kann die entsprechende Hüllkurve auf das in der Tonhöhe verschobene Signal angewendet werden. Dies würde bei starkem Pitch Shifting nach oben für einen wärmeren und beim Pitch Shifting nach unten für einen klareren Klang führen, sodass die Gitarrensignale deutlich natürlicher klingen würden.

Literaturverzeichnis

- [Bal+11] Balasz, Peter et al. "Theory, Implementation and Applications of Nonstationary Gabor Frames". In: (2011).
- [Bri95] Bristow-Johnson, Robert. "A Detailed Analysis of a Time-Domain Formant-Corrected Pitch-Shifting Algorithm". In: *Journal of the Audio Engineering Society* 43 (Mai 1995), S. 340–352.
- [Bro91] Brown, Judith. "Calculation of a constant Q spectral transform". In: *Journal of the Acoustical Society of America* (1991).
- [CM90] Charpentier, F. und Moulines, E. "Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis using Diphones". In: *Speech Communication* 9 (1990), S. 453–467.
- [DLC06] Dorran, David, Lawlor, Robert und Coyle, Eugene. "A comparison of time-domain time-scale modification algorithms". In: *Audio Engineering Society - 120th Convention Spring Preprints* 1 (Jan. 2006).
- [DM14a] Driedger, Jonathan und Müller, Meinard. "Improving Time-Scale Modification of Music Signals Using Harmonic-Percussive Separation". In: *IEEE Signal Processing Letters* 21.1 (Jan. 2014).
- [DM14b] Driedger, Jonathan und Müller, Meinard. "TSM Toolbox: MATLAB Implementations of Time-Scale Modification Algorithms". In: *Proceedings of the 17th International Conference on Digital Audio Effects*. 2014.
- [DM15] Driedger, Jonathan und Müller, Meinard. *Harmonisch-Perkussiv-Rest-Zerlegung von Musiksignalen*. Forschungsber. International Audio Laboratories Erlangen, Jan. 2015.
- [DM16] Driedger, Jonathan und Müller, Meinard. "A Review of Time-Scale Modification of Music Signals". In: *Applied Sciences* 6 (2016), S. 57. DOI: 10.3390/app6020057.
- [DM18] Driedger, Jonathan und Müller, Meinard. *TSM Toolbox*. International Audio Laboratories Erlangen. 11. Dez. 2018. URL: <https://www.audiolabs-erlangen.de/resources/MIR/TSMtoolbox/> (besucht am 17.12.2020).

- [Dol86] Dolson, Mark. "The Phase Vocoder: A Tutorial". In: *Computer Music Journal* 10.4 (1986), S. 14–27.
- [Dör+11] Dörfler, Monika et al. "Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames". In: (2011).
- [Dör+13] Dörfler, Monika et al. *NSGToolbox*. 30. Juli 2013. URL: <http://nsg.sourceforge.net/download.php>.
- [Dor05] Dorran, David. "Audio Time-Scale Modification". Diss. Dublin Institute of Technology, 2005.
- [Dri11] Driedger, Jonathan. "Time-Scale Modification Algorithms for Music Audio Signals". Masterarbeit. Saarland University, 3. Nov. 2011.
- [FG66] Flanagan, J. L. und Golden, R. M. "Phase Vocoder". In: *Bell System Technical Journal* 45 (1966), S. 1493–1509.
- [GL08] Grofit, Shahaf und Lavner, Yizhar. "Time-Scale Modification of Audio Signals Using Enhanced WSOLA With Management of Transients". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.1 (2008), S. 106–115.
- [Gör11] Görne, Thomas. *Tontechnik*. 3. Aufl. 2011. ISBN: 978-3-446-42395-4.
- [GT09] Gu, Hung-Yan und Tsai, Sung-Feng. "A Discrete-cepstrum Based Spectrum-envelope Estimation Scheme and Its Example Application of Voice Transformation". In: *Computational Linguistics and Chinese Language Processing* 14.4 (Dez. 2009), S. 363–382.
- [HDL15] Huang, Dong-Yan, Dong, Minghui und Li, Haizhou. "A Real-Time Variable-Q Non-Stationary Gabor Transform for Pitch Shifting". In: *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association*. 2015.
- [HM91] Hejna, Don und Musicus, Bruce R. "The SOLAFS Time-Scale Modification Algorithm". In: (1991).
- [Hol+13] Holighaus, Nicki et al. "A Framework for Invertible, Real-Time Constant-Q Transforms". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.4 (2013), S. 775–785.
- [Hol10] Holighaus, Nicki. "Zeit-Frequenz-Analyse mit Methoden der Gabor Analysis". Diplomarbeit. Justus-Liebig Universität Giessen, 2010.
- [Hol13] Holighaus, Nicki. "Theory and implementation of adaptive time-frequency transforms". Diss. Universität Wien, 2013.

- [HRS02] Heinzl, G., Rüdiger, A. und Schilling, R. *Spectrum and spectral density estimation by the Discrete Fouriertransform (DFT), including a comprehensive list of window functions and some new flat-top windows*. Techn. Ber. Max-Planck-Institut, 15. Feb. 2002.
- [JAS08] Juillerat, Nicolas, Arisona, Stefan Müller und Schubiger-Banz, Simon. "Low Latency Audio Pitch Shifting in the Time Domain". In: *International Conference on Audio, Language and Image Processing 2008*. Juli 2008, S. 29–35.
- [Kar17] Karrenberg, Ulrich. *Signale - Prozesse - Systeme. Eine multimediale und interaktive Einführung in die Signalverarbeitung*. 7. Aufl. Springer Vieweg, 2017. ISBN: 978-3-662-52658-3.
- [KLB06] Karrer, Thorsten, Lee, Eric und Borchers, Jan. *PhaVoRIT: A Phase Vocoder for Real-Time Interactive Time-Stretching*. Forschungsber. RWTH Aachen, 2006.
- [Kra+12] Kraft, Sebastian et al. "Improved PVSOLA Time-Stretching and Pitch-Shifting for Polyphonic Audio". In: *15th International Conference on Digital Audio Effects DAFx 2012 Proceedings*. 2012.
- [KS10] Kölzer, Hans und Sauvagerd, Ulrich. "Discrete-Time Signals. The Sampling process and spectrum of sampled signals". 22. Feb. 2010.
- [KS12a] Kölzer, Hans und Sauvagerd, Ulrich. "Hilbert filter". 6. Aug. 2012.
- [KS12b] Kölzer, Hans und Sauvagerd, Ulrich. "Up-sampling and Down-sampling of discrete signals using cascaded decimators/interpolators". 15. Nov. 2012.
- [KS16] Kölzer, Hans und Sauvagerd, Ulrich. "Efficient structures for Decimators and Interpolators". 11. Okt. 2016.
- [KS17] Kölzer, Hans und Sauvagerd, Ulrich. "Diskrete Fourier Transform (DFT)". 28. Feb. 2017.
- [LD00] Laroche, Jean und Dolson, Mark. "New Phase-Vocoder Techniques for Real-Time Pitch-Shifting, Chorusing, Harmonizing and other Exotic Audio Modifications". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (3. März 2000).
- [LD97] Laroche, J. und Dolson, M. "Phase-vocoder: about this phasiness business". In: *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*. 1997.
- [LD99] Laroche, J. und Dolson, M. "Improved phase vocoder time-scale modification of audio". In: *IEEE Transactions on Speech and Audio Processing* 7.3 (1999), S. 323–332.

- [Mah01] Mahfuz, Ejaz. "Packet Loss Concealment for Voice Transmission over IP Networks". Masterarbeit. McGill University Montreal, Kanada, 2001.
- [Mar99] Marple, L. "Computing the discrete-time analytic signal via FFT". In: *IEEE Transactions on Signal Processing* 47.9 (1999), S. 2600–2603.
- [Mat20] MathWorks. *Audio Toolbox: Spectral Descriptors (R2020a)*. 2020. URL: <https://www.mathworks.com/help/audio/ug/spectral-descriptors.html> (besucht am 25.03.2021).
- [McA13] McAulay, Robert John. "Sine-Wave based PSOLA Pitch Scaling with Real-Time Pitch Marking". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2013).
- [MD11] Moinet, Alexis und Dutoit, Thierry. "PVSOLA: A Phase Vocoder with Synchronized Overlap-Add". In: *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*. 2011.
- [Ond18] Onderka, Jan. "Pitch Shifting of Audio Signals in Real Time Using STFT on a Digital Signal Processor". Bachelorthesis. Faculty of Information Technology CTU in Prague, 15. Mai 2018.
- [PH17a] Průša, Z. und Holighaus, N. "Phase vocoder done right". In: *2017 25th European Signal Processing Conference (EUSIPCO)*. Sep. 2017.
- [PH17b] Průša, Zdeněk und Holighaus, Nicki. *Phase Vocoder Done Right - Demo*. 2017. URL: <https://l1tfat.github.io/notes/050/> (besucht am 25.02.2021).
- [Rau16] Rauscher-Scheibe, Annabella. "Signale und Systeme II". 2016.
- [RB19] Rai, Anil und Barkana, Buket D. "Analysis of three pitch-shifting algorithms for different musical instruments". In: *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. 2019.
- [Roy19] Royer, Théo. "Pitch-shifting algorithm design and applications in music". Masterarbeit. KTH Royal Institute of Technology in Stockholm, 2019.
- [RR05] Röbel, Axel und Rodet, Xavier. "Efficient Spectral Envelope Estimation and its application to pitch shifting and envelope preservation". In: *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx)*. Sep. 2005.
- [RW85] Roucos, Salim und Wilgus, Alexander M. "High Quality Time-Scale Modification for Speech". In: *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing* (1985).
- [Sch+13] Schörkhuber, Christian et al. *CQT Toolbox*. 2013. URL: <https://www.cs.tut.fi/sgn/arg/CQT/> (besucht am 25.02.2021).

- [Sch+14] Schörkhuber, Christian et al. "A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution". In: *AES 53rd International Conference on Semantic Audio*. 2014.
- [Set07] Sethares, William A. *Rhythm and Transforms*. 2007. ISBN: 978-1-84628-639-1.
- [SKS13] Schörkhuber, Christian, Klapuri, Anssi und Sontacchi, Alois. "Audio Pitch Shifting Using the Constant-Q Transform". In: *Journal of the Audio Engineering Society* 61.7 (2013), S. 562–572.
- [Smi11] Smith III, Julius O. *Spectral Audio Signal Processing*. 2011. URL: https://ccrma.stanford.edu/~jos/sasp/Weighted_Overlap_Add.html (besucht am 07.02.2021).
- [VR93] Verhelst, Werner und Roelands, Marc. "An Overlap-Add Technique based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Bd. 2. 1993.
- [War98] Wardle, Scott. "A Hilbert-Transformer Frequency Shifter for Audio". In: (1998).
- [Wei08] Weinzierl, Stefan. *Handbuch der Audiotechnik*. Springer-Verlag Berlin Heidelberg, 2008. ISBN: 978-3-540-34300-4.
- [Wer19] Werner, Martin. *Digitale Signalverarbeitung mit MATLAB. Grundkurs mit 16 ausführlichen Versuchen*. 6. Aufl. Springer Vieweg, 2019. ISBN: 978-3-658-18646-3.
- [YP96] Yim, S. und Pawate, B. I. "Computationally efficient algorithm for time scale modification (GLS-TSM)". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. 1996.
- [Zöl11] Zölzer, Udo. *DAFX: Digital Audio Effects*. 2. Aufl. John Wiley & Sons, 2011. 626 S. ISBN: 978-0-470-66599-2.
- [Zpl20] Zplane. *élastiquePro Real Time Pitch Shifting Plug-in*. 2020. URL: <https://products.zplane.de/elastique-pitch-2> (besucht am 11.11.2020).

A Vergleich der Algorithmen anhand objektiver Audiomerkmale

Auf den folgenden Seiten sind die Abbildungen zu dem Vergleich der Algorithmen anhand objektiver Audiomerkmale dargestellt. Die dargestellten Strich- und Farbtypen sowie die entsprechenden Parameter der Algorithmen können in Tabelle 6.1 nachgeschlagen werden.

A.1 Zero-Crossing-Rate

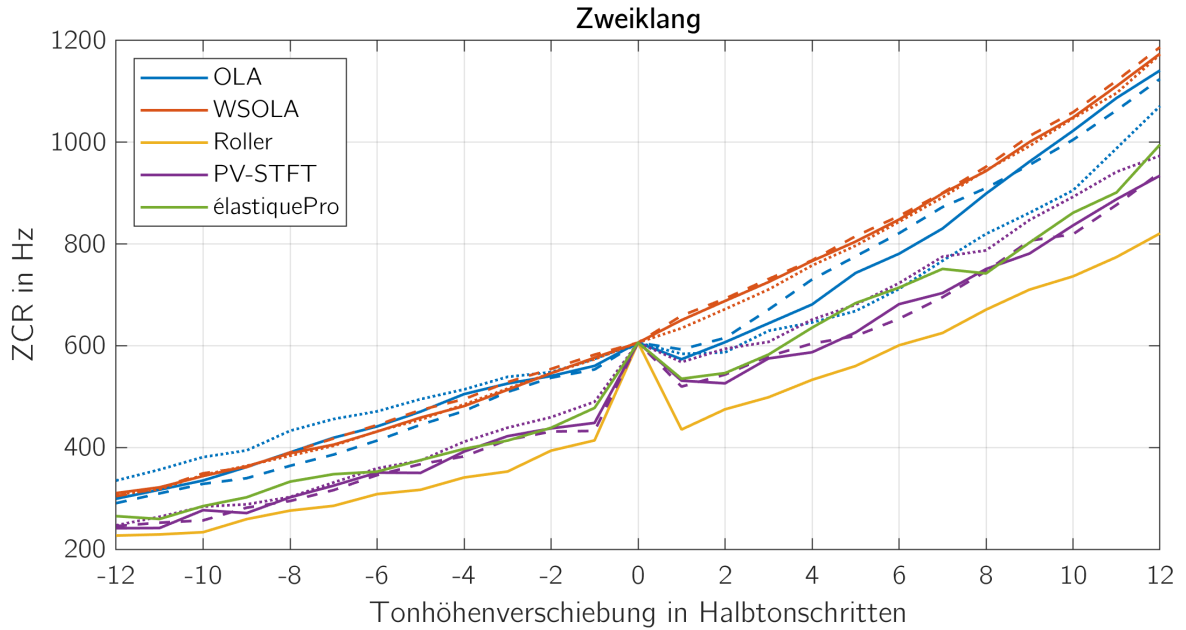


Abbildung A.1: Zero-Crossing-Rate - Testsignal „Zweiklang“

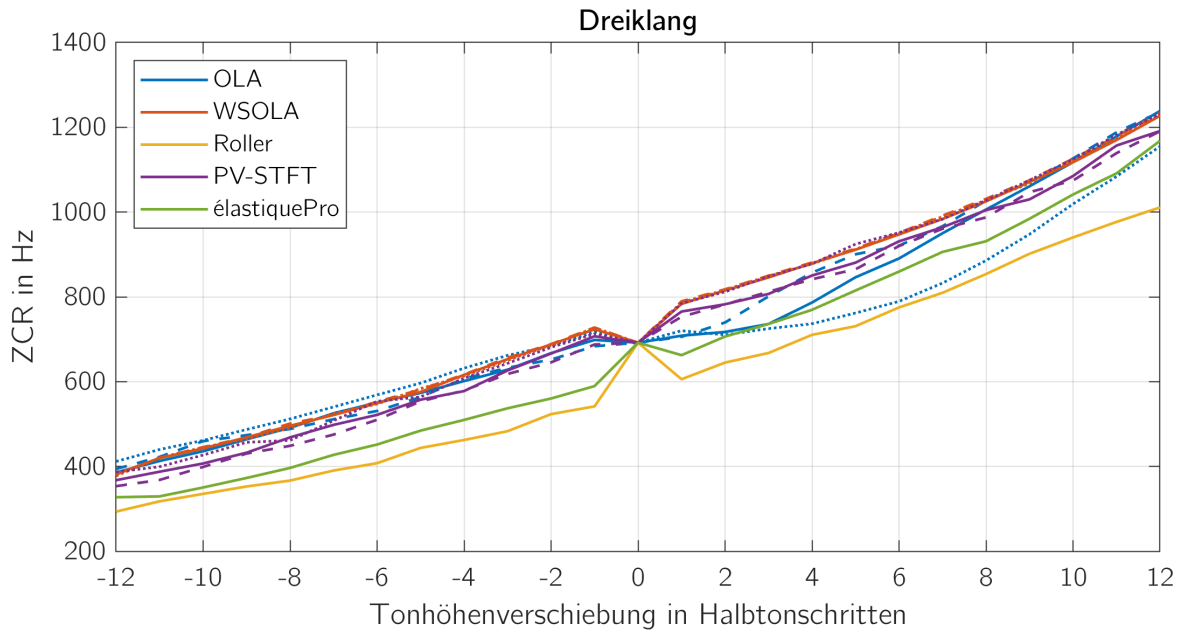


Abbildung A.2: Zero-Crossing-Rate - Testsignal „Dreiklang“

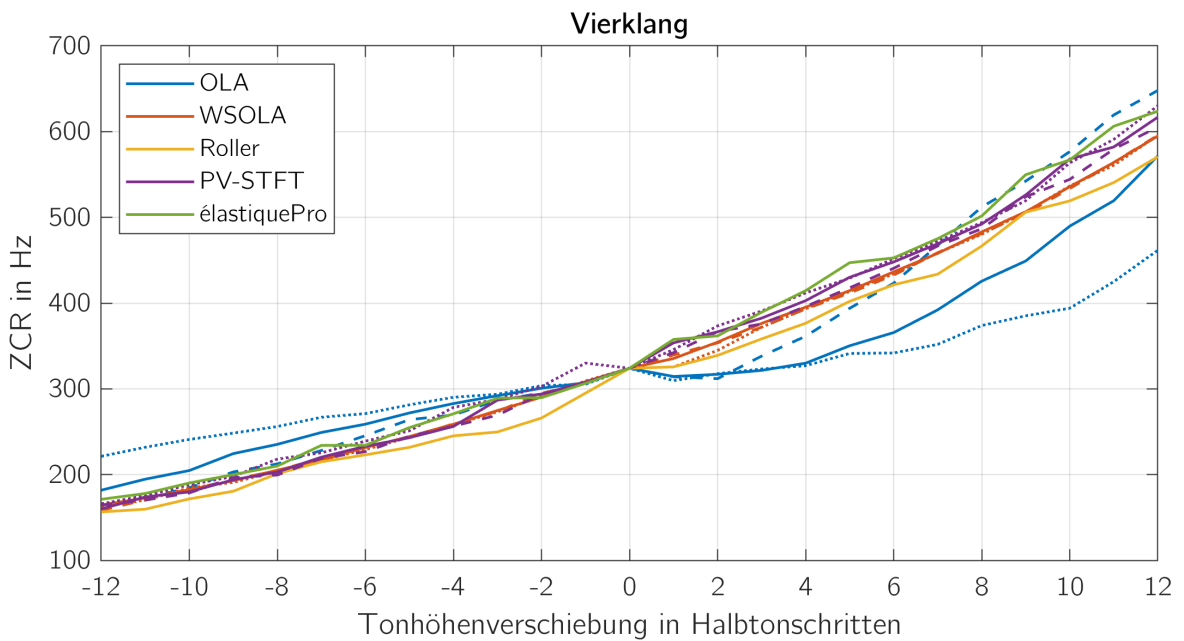


Abbildung A.3: Zero-Crossing-Rate - Testsignal „Vierklang“

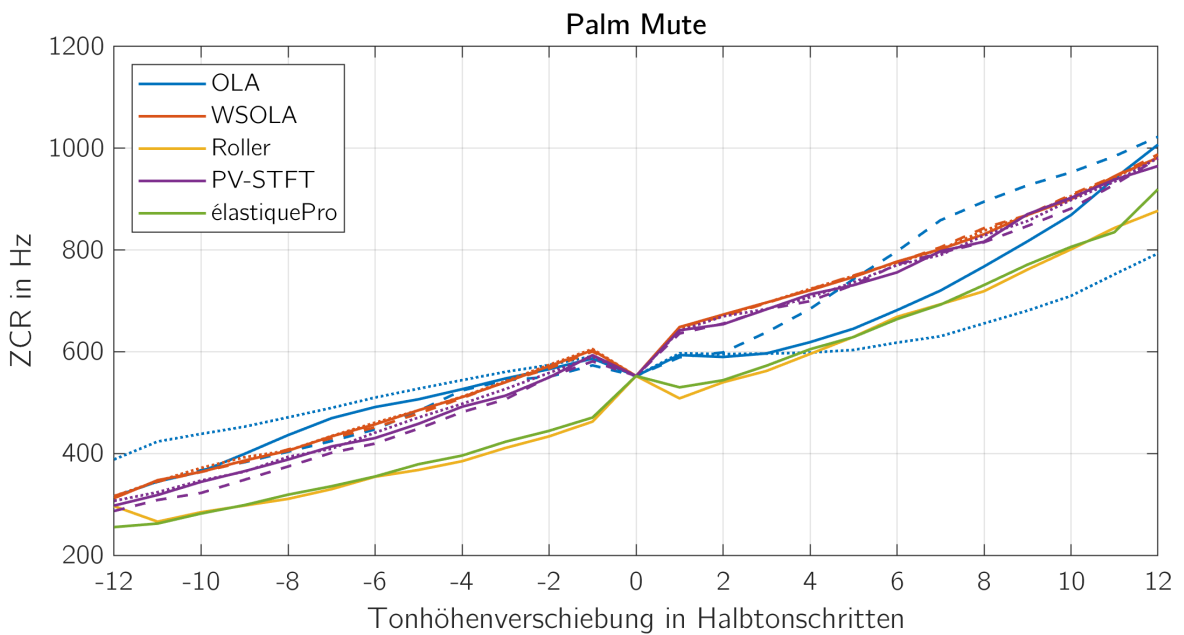


Abbildung A.4: Zero-Crossing-Rate - Testsignal „Palm Mute“

A.2 Spectral Centroid

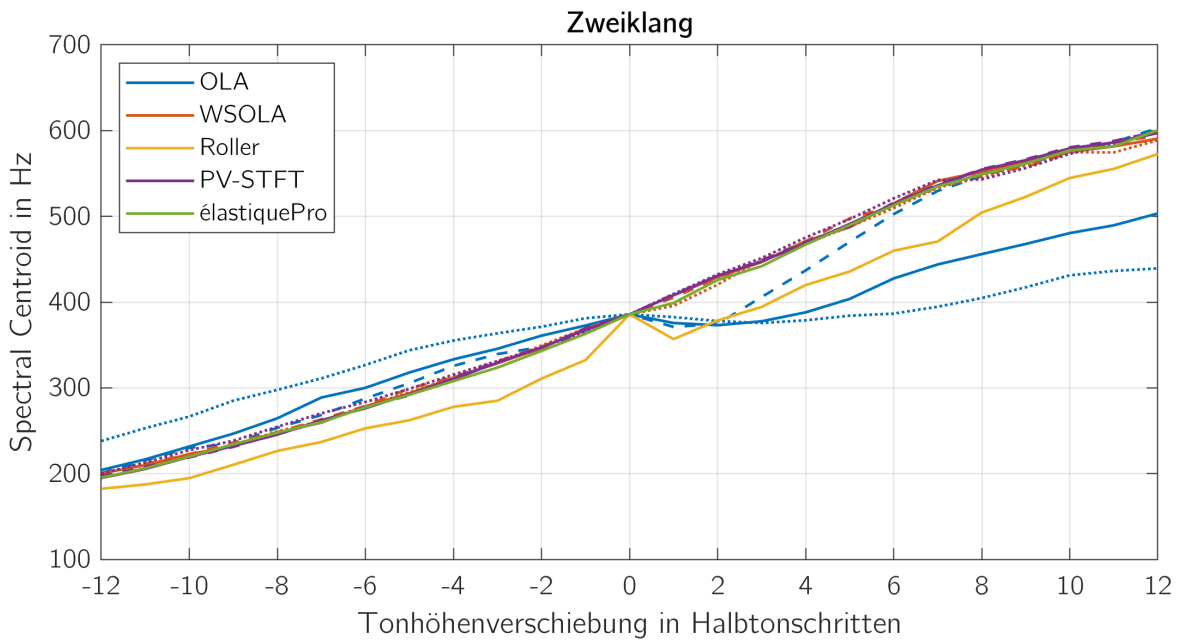


Abbildung A.5: Spectral Centroid - Testsignal „Zweiklang“

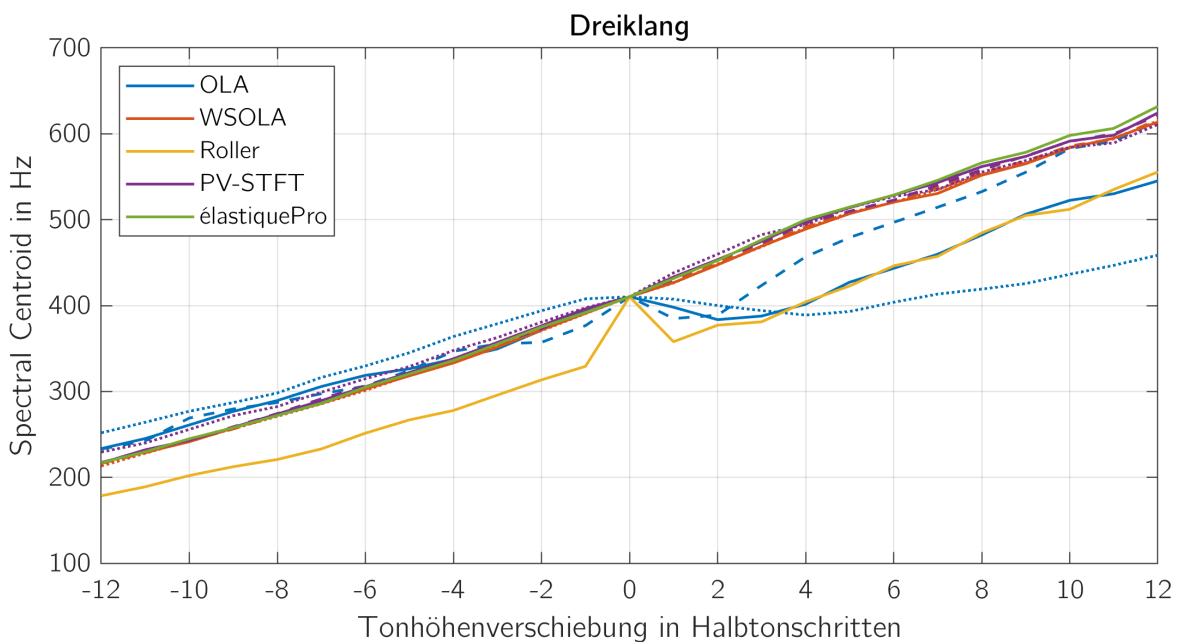


Abbildung A.6: Spectral Centroid - Testsignal „Dreiklang“

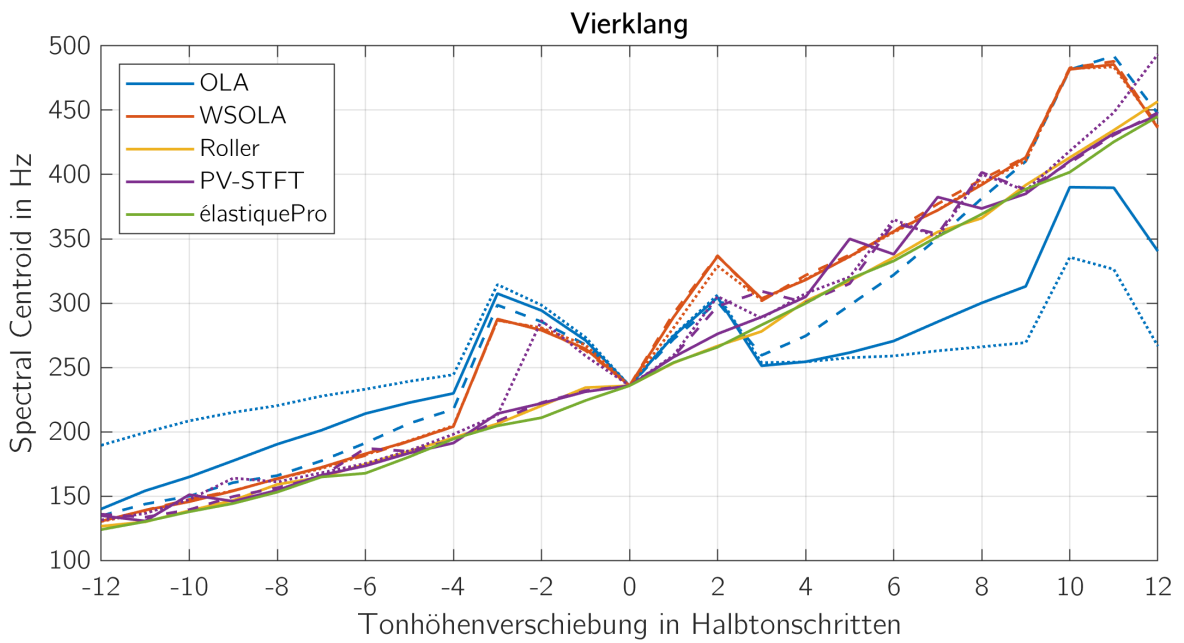


Abbildung A.7: Spectral Centroid - Testsignal „Vierklang“

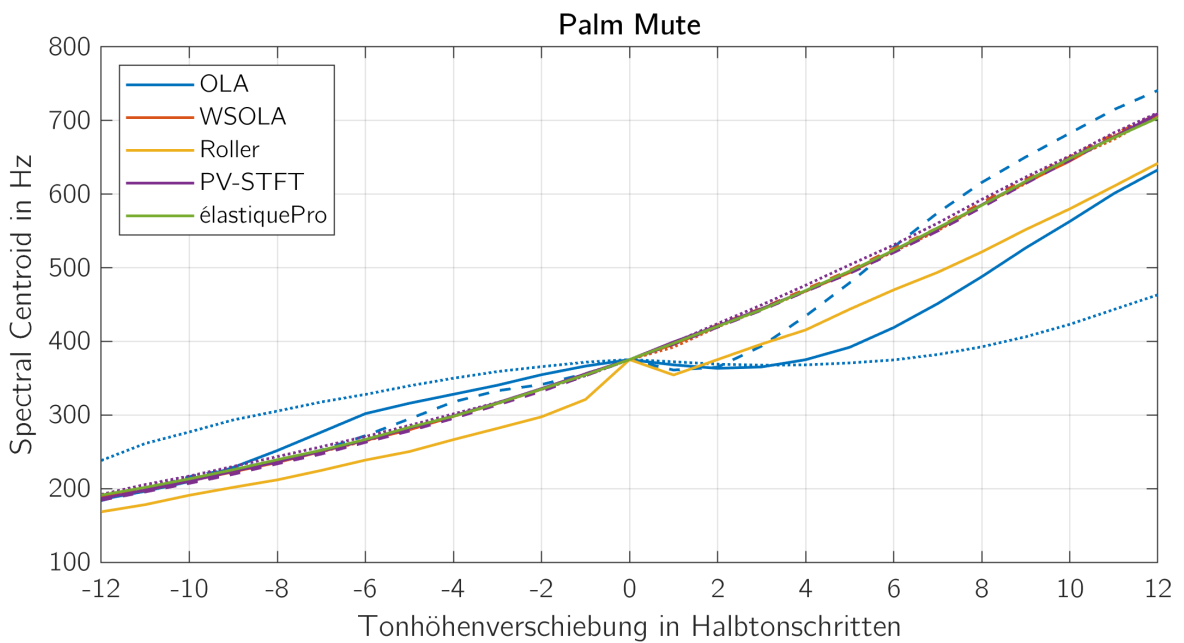


Abbildung A.8: Spectral Centroid - Testsignal „Palm Mute“

A.3 Spectral Entropy

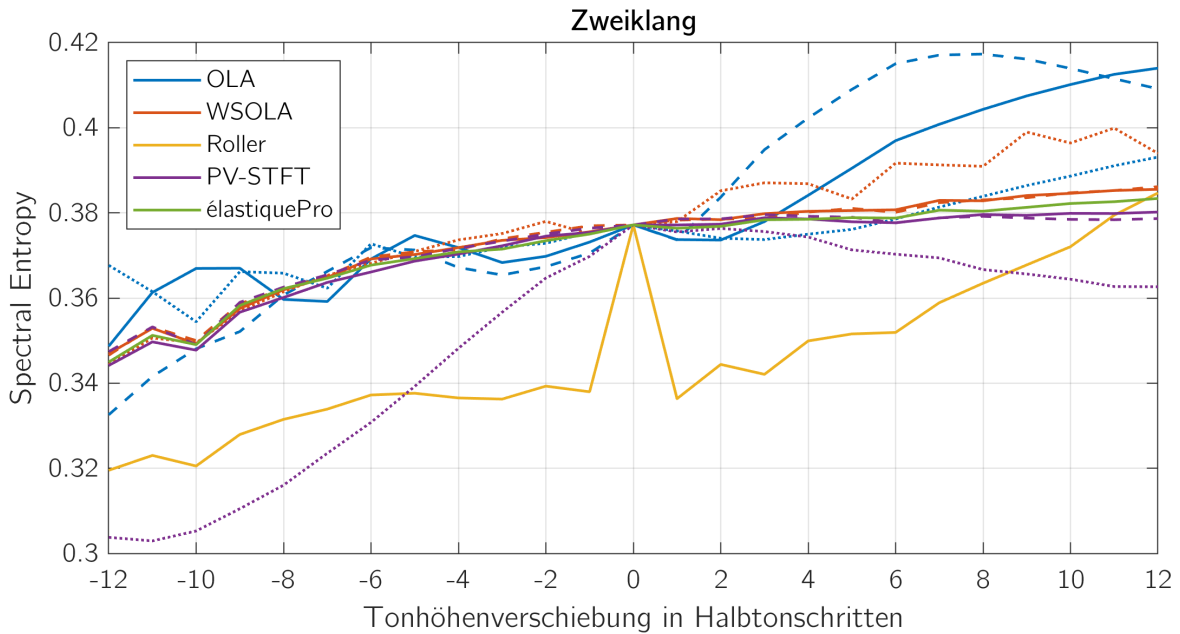


Abbildung A.9: Spectral Entropy - Testsignal „Zweiklang“

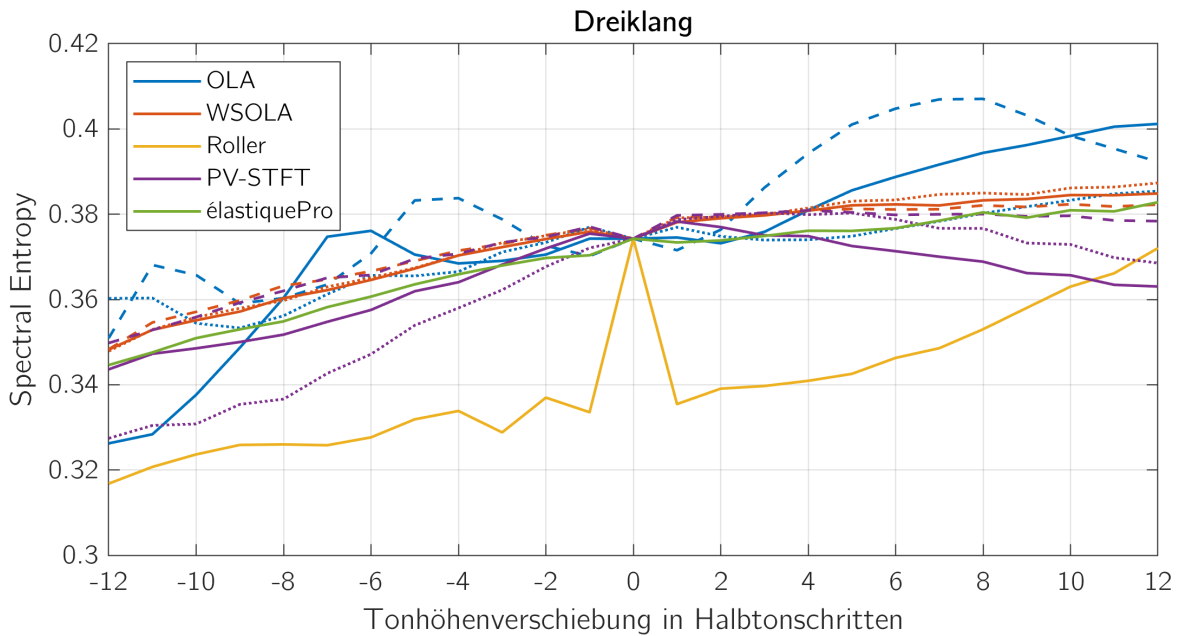


Abbildung A.10: Spectral Entropy - Testsignal „Dreiklang“

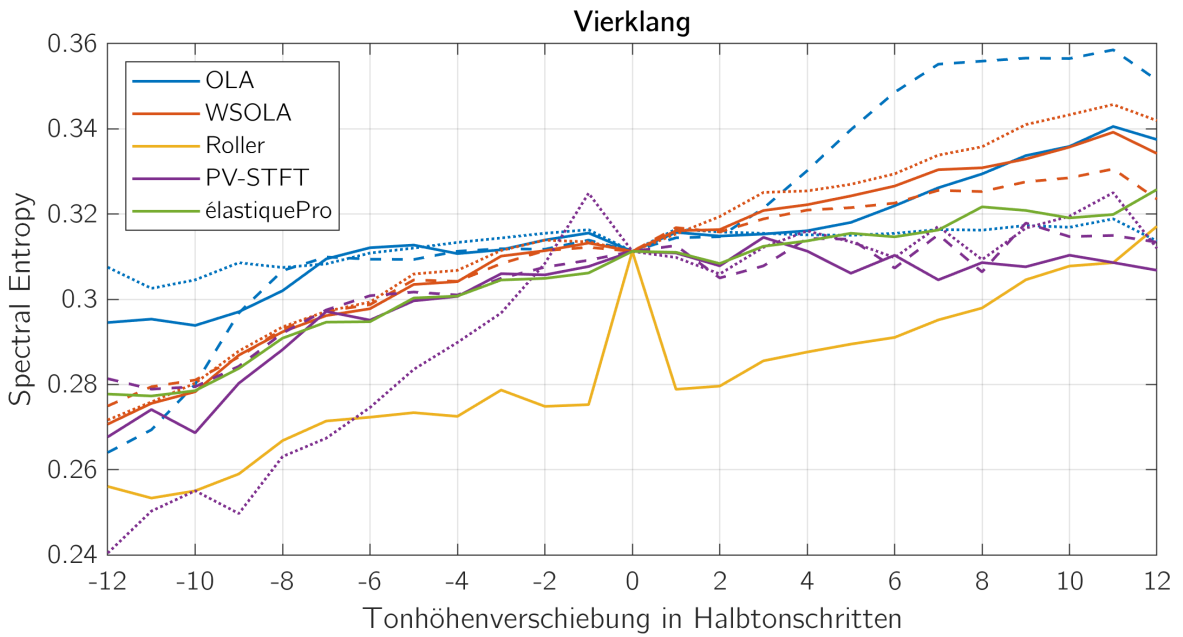


Abbildung A.11: Spectral Entropy - Testsignal „Vierklang“

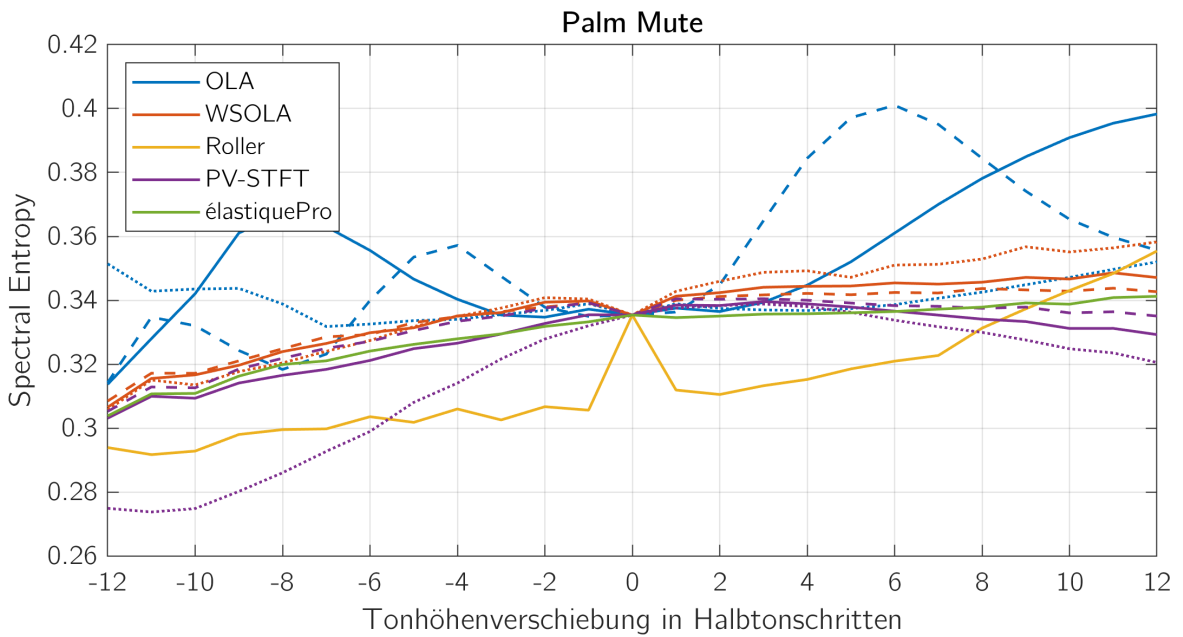


Abbildung A.12: Spectral Entropy - Testsignal „Palm Mute“

A.4 Spectral Spread

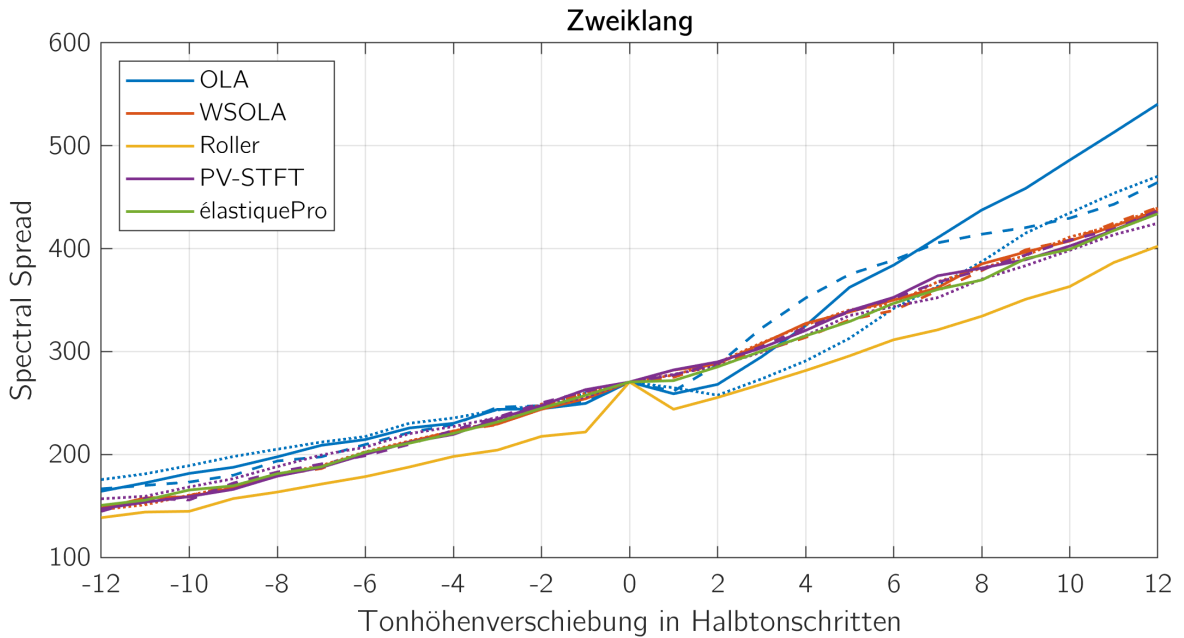


Abbildung A.13: Spectral Spread - Testsignal „Zweiklang“

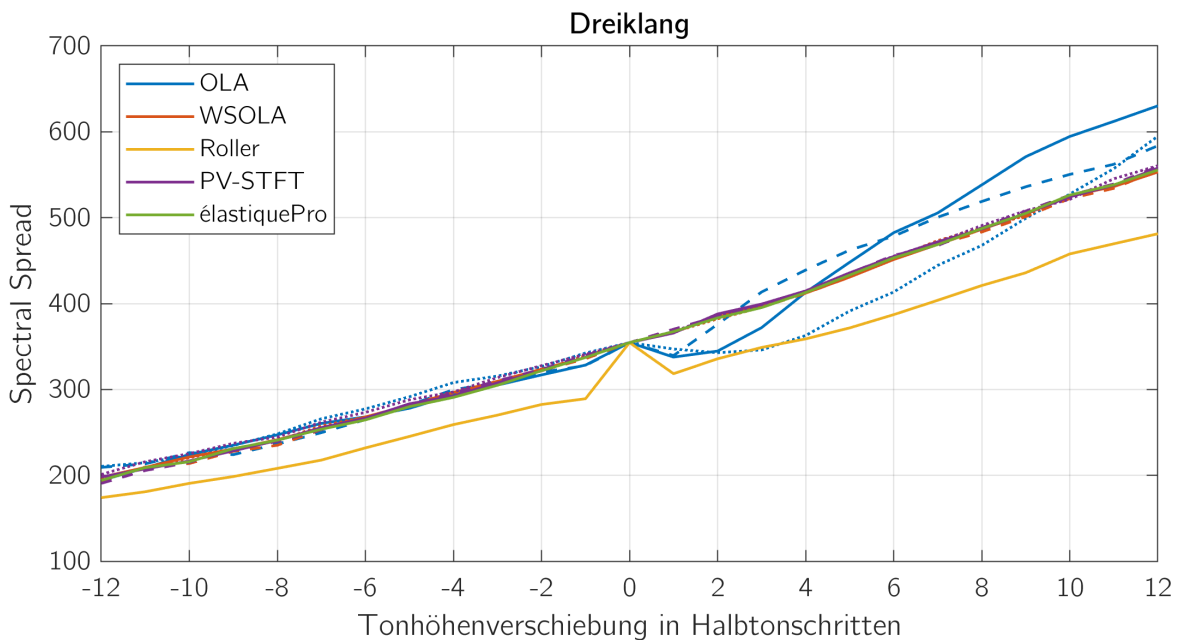


Abbildung A.14: Spectral Spread - Testsignal „Dreiklang“

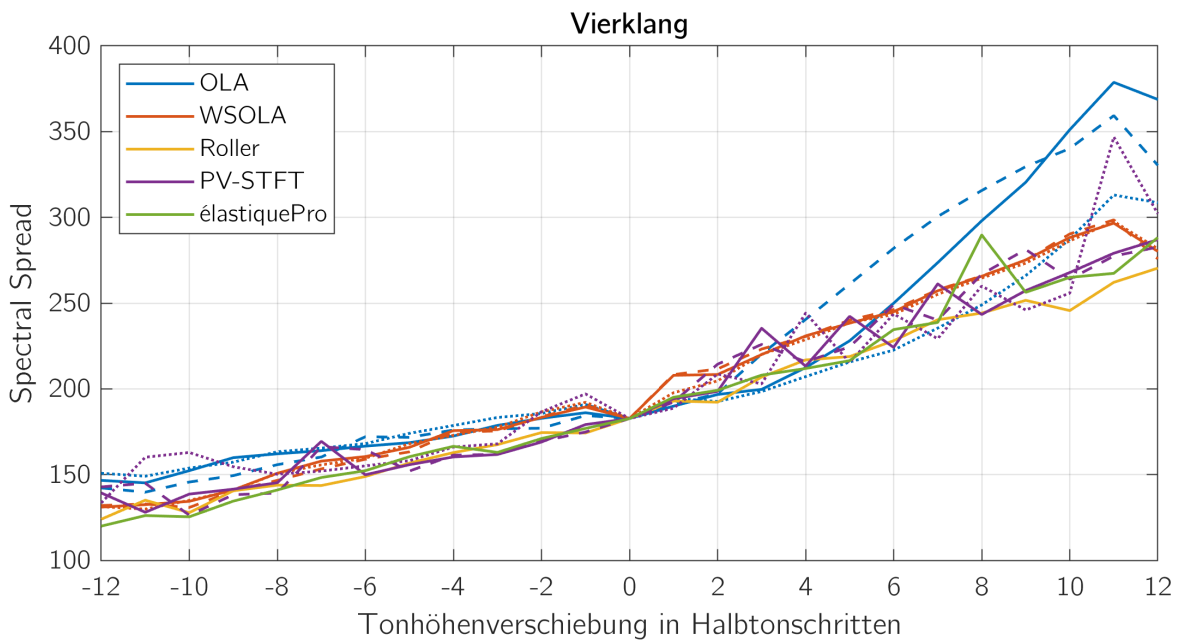


Abbildung A.15: Spectral Spread - Testsignal „Vierklang“

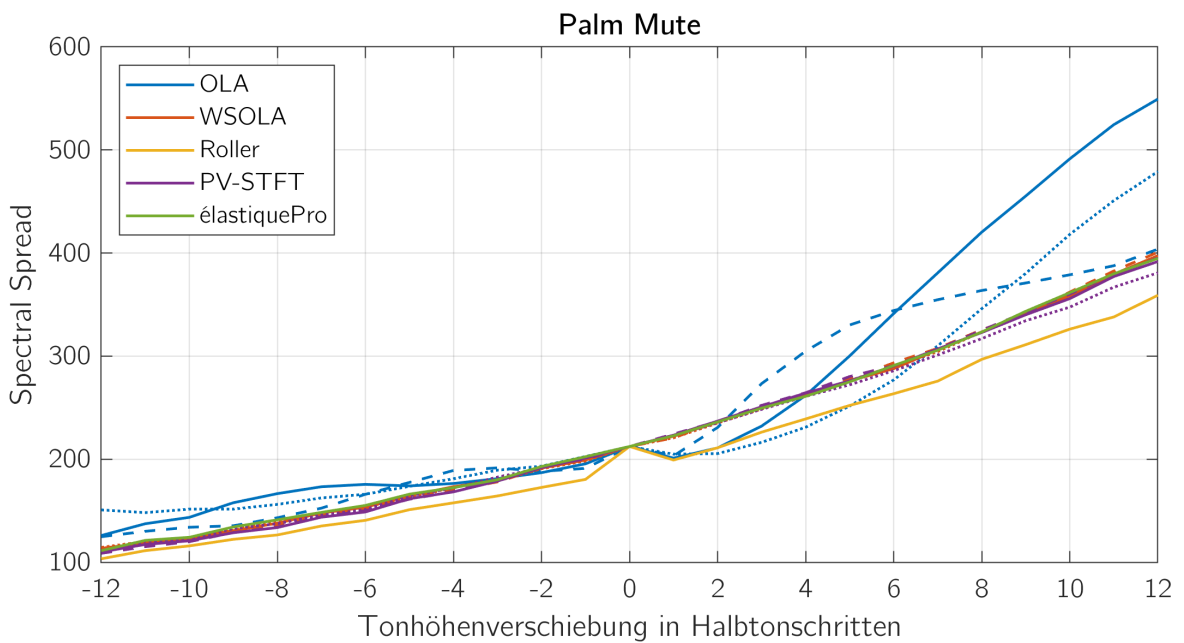


Abbildung A.16: Spectral Spread - Testsignal „Palm Mute“

B Beigefügte CD

Die beigefügte CD kann bei Erstprüfer Herrn Prof. Dr. Sauvagerd oder Zweitprüfer Herrn Prof. Dr. Leutelt eingesehen werden. Auf der CD befinden sich die folgenden Ordner und Dateien:

- *Masterthesis*
 - Masterthesis als PDF-Datei
- *Literatur*
 - Verwendete Literatur
- *MATLAB*
 - *Auswertung*
 - Berechnung und Vergleich der Audiomerkmale
 - *Roller Algorithmus*
 - Implementierung des Roller Algorithmus
 - *Externe Toolboxen*
 - Toolboxen zum OLA, WSOLA, PV-STFT und CQT Pitch Shifting Algorithmus sowie zur slicQ Transformation
- *Audiobeispiele*
 - *Dynamische Formanten*
 - Beispiele für dynamische Formanten
 - *Testsignale*
 - Zweiklang, Dreiklang, Vierklang und Palm Mute
 - *Pitch Shifting*
 - Ausgangssignale der Algorithmen

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 13. April 2021

Ort, Datum

Unterschrift