

Haiko Nahm

Untersuchung von Möglichkeiten der  
Wavelet-Transformation bei der Analyse und  
Synthese von Gitarrenklängen

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

Betreuender Prüfer : Prof. Dr.Ing. Wolfgang Fohl  
Zweitgutachter : Prof. Dr.Ing. Jürgen Missun

Abgegeben am 28. Februar 2011

**Haiko Nahm**

**Thema der Diplomarbeit**

Untersuchung von Möglichkeiten der Wavelet-Transformation bei der Analyse und Synthese von Gitarrenklängen

**Stichworte**

Klang-Modellierung, tonale Modellierung, Transienten-Modellierung, Anschlaggeräusche, Wavelet-Transformation, Wavelet-Pakete, Gitarre, Audiokompression, Instrumenten-Synthese

**Kurzzusammenfassung**

In dieser Arbeit wird ein Verfahren zur Analyse und Synthese von Gitarrenklängen vorgestellt. Das Signal wird dabei separat mit seinem tonalen und transienten Anteil modelliert. Der tonale Anteil wird mit einem sinusoidalen Modelling verarbeitet, der transiente Anteil mit einem Modelling, welches die Wavelet-Transformation nutzt. Es wird auf die Theorie der Wavelet-Transformation, sowie ihre Möglichkeiten in der Klang-Modellierung eingegangen. Anschliessend wird das Verfahren in einem Hörtest beurteilt.

**Haiko Nahm**

**Title of the paper**

Research of possibilities of the wavelet-transform for the analysis and synthesis of guitar sounds

**Keywords**

Sound Modeling, Tonal Modeling, Transient Modeling, Attack Transients, Wavelet Transform, Wavelet Packages, Guitars, Audio Compression, Instruments Synthesis

**Abstract**

The purpose of this work is to present a method for the analysis and synthesis of guitar sounds. In this method the signal is seperatly mold in its tonal and transient component. The tonal part is proessed with sinusoidal modelling, the transient part with a modeling using the wavelet transformation. The theory of the wavelet transformation is described, so is its possibilities with sound modeling. This work concludes with a listening test to evaluate the results.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>5</b>
<b>Abbildungsverzeichnis</b>	<b>6</b>
<b>Struktogrammverzeichnis</b>	<b>8</b>
<b>1. Einführung</b>	<b>10</b>
1.1. Problemstellung und Motivation . . . . .	10
1.2. Eigenschaften der Konzertgitarre . . . . .	12
1.3. Implementierung des Verfahrens . . . . .	13
<b>2. Grundlagen der Wavelet-Transformation</b>	<b>15</b>
2.1. Die Wavelet-Analyse . . . . .	15
2.2. Die kontinuierliche Wavelet-Transformation . . . . .	16
2.3. Die diskrete Wavelet-Transformation . . . . .	18
2.4. Anwendung der Wavelet-Transformation . . . . .	26
<b>3. Wavelets in der Klang-Modellierung</b>	<b>28</b>
3.1. Wavelets in der Tonalen-Modellierung . . . . .	28
3.2. Wavelets in der Transienten-Modellierung . . . . .	32
3.2.1. Sequentielle Extraktion der Transienten . . . . .	32
3.2.2. Simultane Extraktion der Transienten . . . . .	38
<b>4. Das entwickelte Analyse- und Synthese-Verfahren</b>	<b>43</b>
4.1. Trennung von tonalen und transienten Anteilen des Signals . . . . .	43
4.2. Analyse und Synthese des tonalen Anteils . . . . .	46
4.2.1. Analyse des tonalen Anteils . . . . .	47
4.2.2. Synthese des tonalen Anteils . . . . .	51
4.2.3. Beurteilung der Analyse und Synthese des tonalen Signalanteils . . . . .	52
4.3. Analyse und Synthese des transienten Anteils . . . . .	54
4.3.1. Analyse des transienten Signalanteils mit schneller Wavelet-Transformation . . . . .	55
4.3.2. Synthese des transienten Anteils mit schneller Wavelet-Transformation . . . . .	72
4.3.3. Analyse und Synthese des transienten Anteils unter Verwendung von Wavelet-Paketen . . . . .	78

---

4.3.4. Bewertung des entwickelten Verfahrens zur Transienten-Modellierung	84
<b>5. Hörtest zur Evaluation der Transienten-Synthese</b>	<b>87</b>
5.1. Testdurchführung . . . . .	87
5.2. Testergebnis und Bewertung . . . . .	88
<b>6. Fazit und Ausblick</b>	<b>90</b>
6.1. Erreichte Ziele . . . . .	90
6.2. Ausblick . . . . .	91
<b>Literaturverzeichnis</b>	<b>92</b>
<b>A. Inhalt der Daten-CD</b>	<b>95</b>
<b>B. Testbogen zum Hörtest</b>	<b>96</b>

# Tabellenverzeichnis

3.1. Vergleich Qualität Wavelet-Transformation und Komprimierung von tonalem Signal . . . . .	29
4.1. Vergleich Komprimierungsraten bei verschiedenen Wavelet-Basen . . . . .	60
4.2. Verteilung Wavelet-Koeffizienten . . . . .	61
4.3. Kompressionsraten Wavelet-Paket-Transformation . . . . .	81
5.1. Zuordnung der Testfiles zur Gitarrenklang-Datenbank . . . . .	88
5.2. Testergebnisse des Hörtest Teil 1 . . . . .	88
5.3. Testergebnisse des Hörtest Teil 2 . . . . .	89

# Abbildungsverzeichnis

1.1. Matlab-Dateien zum Analyse-Verfahren . . . . .	13
1.2. Matlab-Dateien zur Separation des nontonalen Anteils . . . . .	14
1.3. Matlab-Dateien zum Synthese-Verfahren . . . . .	14
2.1. Daubechies-8 Wavelet . . . . .	16
2.2. Haar-Wavelet . . . . .	16
2.3. Beispiel der $(a, s)$ -Ebene einer Wavelet-Transformation . . . . .	18
2.4. 2-Kanal-PR-Filterbank (Bild aus (Bae05)) . . . . .	19
2.5. Wavelet-Tree . . . . .	19
2.6. Wavelet-Transformation (Bild aus (Bae05)) . . . . .	23
2.7. Wavelet-Rekonstruktion (Bild aus (Bae05)) . . . . .	23
2.8. Wavelet-Pakete . . . . .	25
2.9. Haar-Wavelet-Pakete der 3. Baum-Stufe (Bild aus (Bae05)) . . . . .	25
3.1. Vergleich Spektrum modelliertes, tonales Signal STFT und DWT . . . . .	30
3.2. Wavelet-Koeffizienten bei Sinus-Signalen verschiedener Frequenzen . . . . .	31
3.3. Blockschaltbild der Extraktion des transienten Signalanteils . . . . .	33
3.4. Nontonaler Signalanteil, extrahierte Transiente und Residue . . . . .	34
3.5. Ausgangssignal, extrahierte Transiente und Residue . . . . .	37
3.6. Rampen-Signal und Wavelet-Transformation . . . . .	39
3.7. Verlauf der Maxima der Wavelet-Koeffizienten bei Rampen-Signal . . . . .	40
3.8. Nontonaler Signalanteil, extrahierte Transiente und Residue (aus (Dau01)) . . . . .	41
4.1. Gesamtsignal und extrahierter nontonaler Anteil . . . . .	45
4.2. Blockschaltbild Analyse tonaler Signalanteil . . . . .	47
4.3. Phasenverläufe der Partialtöne (aus (vC09)) . . . . .	53
4.4. Blockschaltbild Transienten-Modellierung . . . . .	55
4.5. Blockschaltbild Hüllkurven-Ermittlung . . . . .	56
4.6. Vergleich der ermittelten Hüllkurven . . . . .	56
4.7. Fehlerhafte Ermittlung vom Beginn der Attack-Phase . . . . .	57
4.8. APR-Hüllkurve . . . . .	59
4.9. Ermittelte Hüllkurve . . . . .	59
4.10. Verteilung der Wavelet-Koeffizienten über die Skalen der Wavelet-Transformation . . . . .	60

---

4.11. Approximations-Koeffizienten . . . . .	62
4.12. Wavelet-Koeffizienten der Skala 5 . . . . .	63
4.13. Wavelet-Koeffizienten der Skala 6 . . . . .	63
4.14. Wavelet-Koeffizienten der Skala 7 . . . . .	63
4.15. Wavelet-Koeffizienten der Skala 8 . . . . .	63
4.16. Anzahl Koeffizienten bei verschiedenen Saiten . . . . .	64
4.17. Blockschaltbild Transienten-Rekonstruktion . . . . .	72
4.18. Transiente und APR-Hüllkurven für zwei verschiedene Gitarrenklänge . . . . .	73
4.19. Vergleich Synthetische Transienten mit und ohne APR-Hüllkurven-Verfahren . . . . .	74
4.20. Spektrum mit schneller FWT resynthetisierte Transiente . . . . .	84
4.21. Spektrum mit Wavelet-Paketen resynthetisierte Transiente . . . . .	84
4.22. Synthetische Transiente Saite 1 nach neuem Verfahren . . . . .	86
4.23. Synthetische Transiente Saite 1 nach vonColer Verfahren . . . . .	86
4.24. Synthetische Transiente Saite 6 nach neuem Verfahren . . . . .	86
4.25. Synthetische Transiente Saite 6 nach vonColer Verfahren . . . . .	86

# Struktogrammverzeichnis

3.1. Programmablauf zu <i>extractfromnonton_transient.m</i> . . . . .	35
3.2. Variablen zu <i>extractfromnonton_transient.m</i> . . . . .	36
4.1. Programmablauf zu <i>process.m</i> . . . . .	46
4.2. Variablen zu <i>process.m</i> . . . . .	46
4.3. Programmablauf zu <i>extract_features_tonal.m</i> . . . . .	48
4.4. Variablen zu <i>extract_features_tonal.m</i> . . . . .	48
4.5. Die Struktur <i>features</i> . . . . .	49
4.6. Programmablauf zu <i>getTemporalCourseHarmonics.m</i> . . . . .	50
4.7. Variablen zu <i>getTemporalCourseHarmonics.m</i> . . . . .	50
4.8. Programmablauf zu <i>synthesizer.m</i> . . . . .	51
4.9. Programmablauf zu <i>synthesize_frame.m</i> . . . . .	52
4.10. Variablen zu <i>synthesize_frame.m</i> . . . . .	52
4.11. Die Struktur <i>trans_feat</i> . . . . .	66
4.12. Programmablauf zu <i>analysis_features_trans.m</i> . . . . .	66
4.13. Variablen zu <i>analysis_features_trans.m</i> . . . . .	67
4.14. Programmablauf zu <i>envelope_calc.m</i> . . . . .	68
4.15. Variablen zu <i>envelope_calc.m</i> . . . . .	69
4.16. Programmablauf zu <i>get_envelope.m</i> . . . . .	69
4.17. Variablen zu <i>get_envelope.m</i> . . . . .	69
4.18. Programmablauf zu <i>get_wavelet_koeff.m</i> . . . . .	70
4.19. Variablen zu <i>get_wavelet_koeff.m</i> . . . . .	70
4.20. Programmablauf zu <i>entrophie_codierung.m</i> . . . . .	71
4.21. Variablen zu <i>entrophie_codierung.m</i> . . . . .	71
4.22. Programmablauf zu <i>resynth_trans.m</i> . . . . .	75
4.23. Variablen zu <i>resynth_trans.m</i> . . . . .	75
4.24. Programmablauf zu <i>entrophie_decodierung.m</i> . . . . .	76
4.25. Variablen zu <i>entrophie_decodierung.m</i> . . . . .	76
4.26. Programmablauf zu <i>create_transient.m</i> . . . . .	76
4.27. Variablen zu <i>create_transient.m</i> . . . . .	77
4.28. Programmablauf zu <i>analysis_features_trans_pack.m</i> . . . . .	78
4.29. Variablen zu <i>analysis_features_trans_pack.m</i> . . . . .	79
4.30. Die Struktur <i>trans_feat_pack</i> . . . . .	79
4.31. Programmablauf zu <i>get_wavelet_koeff_pack.m</i> . . . . .	80



---

4.32. Variablen zu <i>get_wavelet_koeff_pack.m</i> . . . . .	80
4.33. Programmablauf zu <i>resynth_trans_pack.m</i> . . . . .	82
4.34. Variablen zu <i>resynth_trans_pack.m</i> . . . . .	82
4.35. Programmablauf zu <i>create_transient_pack.m</i> . . . . .	82
4.36. Variablen zu <i>create_transient_pack.m</i> . . . . .	83

# 1. Einführung

## 1.1. Problemstellung und Motivation

Die modellhafte Analyse und Synthese von Klangereignissen ist für unterschiedliche Anwendungen von Interesse. Sie dient unter anderem der Kompression von Audiomaterial, also zur Datenreduktion, der synthetischen Klangerzeugung oder diversen Analysen vorhandenen Materials, wie z.B. dem Auffinden von Notenanfängen im Zeitbereich eines Musiksignals oder der Bestimmung der Notenwerten. In einer Analyse wird das Ausgangsmaterial in einen Parametersatz überführt. In bestimmten Modellen kann je nach Anwendung in einer Synthese aus diesen Parametern das Signal wieder rekonstruiert werden. Die hinter der Analyse und Synthese stehenden Verfahren sowie die Art der Parameter unterscheiden sich von Modell zu Modell sowie in Bezug auf die gewünschte Anwendung.

Das in dieser Arbeit vorgestellte Verfahren fundiert auf wahrnehmungsbasierten Modellen. Diese Modelle beschreiben Klangereignisse, indem sie diese in akustische Sinneinheiten unterteilt. Ein Audiosignal wird dabei als Überlagerung mehrerer Komponenten-Signale verstanden, wobei letztere in diesem Fall in Kategorien eingeteilt werden, die sich an der menschlichen, akustischen Wahrnehmung orientieren. Komponenten sind dabei z.B. tonale und nicht-tonale Signalanteile, (quasi)stationäre Anteile und stochastische Klangereignisse. Ziel der Modellierung ist es, das Ausgangssignal in diese Komponenten aufzuteilen, um diese dann getrennt voneinander modellieren zu können.

Das hier behandelte Modell unterteilt ein Audiosignal in drei Komponenten, den tonalen, (quasi)stationären Anteil, also den reinen Klang, den transienten Anteil, also das Anschlaggeräusch, sowie der Residue, einen stochastischem Restanteil, der überwiegend als Noise wahrgenommen wird:

$$x(t) = x_{ton}(t) + x_{tran}(t) + x_{res}(t). \quad (1.1)$$

Angewendet wird das Verfahren auf Audiosignale in Form von einzelnen Tönen, im Speziellen Aufzeichnungen von einzeln angeschlagenen Gitarrensaiten. Ziel der Analyse und Synthese ist es, das Ausgangsmaterial, vorliegend in digitalen Aufzeichnungen der analogen Ausgangssignale, nach dem beschriebenen Modell in einen Parametersatz der verschiedenen Komponenten zu zerlegen und aus diesem in einer Synthese das Ausgangssignal

möglichst wahrheitsgetreu wieder zu rekonstruieren, so dass es nach einer Digital/Analog Umwandlung für das menschlichen Gehör sich möglichst gering von ausgehenden, analogem Signal unterscheidet. Am genauesten würde dieses durch ein reines Sample-Verfahren entstehen, bei dem das digitalisierte Signal wieder direkt in ein analoges umgewandelt wird. Das hat jedoch zwei entscheidende Nachteile. Der so als reines Digitalsignal vorliegende Datensatz bietet wenige Möglichkeiten zur Anschauung und Manipulation, Eigenschaften des Tones wie Notenhöhe, Anschlagstärke oder musikalische Betonung sind darin nicht direkt ersichtlich und veränderbar. Zudem benötigt ein solches Verfahren ein Maximum an Speicherplatz für den Datensatz zur Klang-Resynthese. Daher ist es ein Ziel des Verfahrens, in der Analyse das Audiosignal in ein Modell nach 1.1 zu überführen, das entsprechende Parameter zur Verfügung stellt, aus denen sich dann das Ausgangssignal möglichst gering verlustbehaftet resynthetisieren lässt.

Ausgangspunkt der Überlegungen dieser Arbeit ist das im Rahmen einer Diplomarbeit entwickelte Verfahren aus (vC09). In dem dort vorgestellten Verfahren zur Analyse und Synthese von Gitarrenklängen werden tonaler und transientser Signalanteil voneinander getrennt modelliert, die Residue wird vernachlässigt. Der tonale Signalanteil wird mit einem spektralen Modell modelliert, im Ausgangssignal werden über eine Kurzzeit-Fourier-Transformation (STFT) die Grundtöne sowie zwanzig zugehörige Partialtöne bestimmt und in ein sinusoidales Modell überführt, aus dessen resultierenden Parametersatz der tonale Signalanteil wieder resynthetisiert wird.

Der transiente Signalanteil, im Sinne eines Gitarrentones definiert als das Anschlaggeräusch der Gitarre, wird ebenfalls über ein sinusoidales Verfahren modelliert, einer einfachen Fourier-Transformation, wobei das nontonale Signal (zuvor mittels einer Eliminierung des tonalen Anteils im Spektrum des Ausgangssignals bestimmt) über eine Filterbank in zwanzig Bänder unterteilt wird. Da der reinen Fourier-Transformation die Zeitauflösung fehlt, also die Information über die Veränderung des Spektrums über die Zeit, wird in der Analyse im Zeitbereich zu jedem der zwanzig Bänder eine APR-Hüllkurve aus der Signalhüllkurve bestimmt und neben den Sinusoiden im Parametersatz zum transienten Signalanteil hinterlegt (Bandweise Hüllkurven und Wellenformen), aus dem in der Synthese die Transiente rekonstruiert werden kann.

Der durch Addition der resynthetisierten tonalen und transienten Signalanteile hergestellte, synthetische Gitarrenklang hat dabei mehrere Schwachstellen. Neben einem etwas künstlichen Klangcharakter, der auf die Modellierung des tonalen Signalanteils zurückzuführen ist, wirken die Anschlaggeräusche relativ dumpf, teilweise wenig scharf und sind zeitlich zum tonalen Anteil leicht versetzt.

Motivation dieser Arbeit ist es, diese Punkte zu verbessern, mit Hauptaugenmerk auf dem transienten Signalanteil. Wie aus der Literatur herausgeht ((DT02) oder (Nsa07)), sind sinusoidale Modelle für die Modellierung transienter Signalanteile schlecht geeignet, da diese schnellen Wechsels im Zeitbereich unterworfen sind, wofür sinusoidale Transformationen weniger geeignet sind, zudem hat das in (vC09) entworfene Verfahren den Nachteil, dass die

resultierenden Transienten in ihrer Klangcharakteristik stark von der Genauigkeit der ermittelten Hüllkurven abhängig sind, da diese, über rein stationäre Signale „gestülpt“ werden.

Für eine Modellierung stochastischer Signale mit schnellen Veränderungen im Zeitbereich findet man in der Literatur daher häufig neben der Anwendung einer STFT die Anwendung der Wavelet-Transformationen. Diese Familie der linearen Zeit-Frequenz-Transformationen hat den Vorteil, in ihrer Frequenz- und Zeitauflösung variabel zu sein und ist sehr gut für die Darstellung schneller Wechsel im Zeitbereich geeignet, auch können sie diese meist schon mit einer verhältnismäßig geringen Anzahl von Koeffizienten darstellen.

In diesem Projekt werden also, nach einer Einführung in die Thematik und Theorie der Wavelet-Transformation, die Möglichkeiten einer Anwendung dieser auf ein Modell zur Analyse und Synthese von Klangereignissen nach der Formel 1.1 erforscht und in einer Übersicht dargestellt, um dann im Hauptteil ein neues Verfahren vorzustellen, das mit Hilfe der Wavelet-Transformation eine Verbesserung der Transienten-Modellierung zum Ziel hat.

## 1.2. Eigenschaften der Konzertgitarre

Das hier entwickelte Verfahren zur Analyse und Synthese von Klangereignissen wird auf Aufzeichnungen eingespielter Gitarrenklänge einer Konzertgitarre angewendet. Eine Konzertgitarre, zugehörig zu der Instrumentengruppe der Saiteninstrumente, besitzt sechs Saiten aus Nylonseide - drei Diskant-Saiten, die mit massivem Nylon umspannen sind, sowie drei Bass-Saiten, die mit Kupfer oder Silberdraht umspannen sind. Der Ton wird durch Zupfen oder Schlagen der Saiten erzeugt, wobei diese in Schwingung geraten. Verstärkt wird der Ton durch den Korpus der Gitarre, wobei dieser eine Vielzahl von Resonanzfrequenzen besitzt, die durch die Saite angeregt werden. Die Resonanzfrequenz des Hohlraumes liegt bei ca.  $f = 100\text{Hz}$ , die Resonanzfrequenzen der Decke und des Bodens bei etwas unter  $f = 200\text{Hz}$ , bei  $f = 300 - 400\text{Hz}$  sowie  $f = 550\text{Hz}$ . Die Resonanzfrequenz des Halses liegt unter  $f = 100\text{Hz}$ .

In dieser Arbeit werden ausschließlich Aufnahmen gezupfter Töne verwendet, eingespielt auf einer Gitarre vom Typ Hense, mit sonorer Spielweise. Verarbeitet werden pro Saite vier verschiedene Aufzeichnungen. Die entsprechenden Soundfiles stammen aus einer Datenbank (digitalisiert mit einer Samplerate von 44.1kHz und einer Quantisierung von 24Bit), die im Zuge des Projekts einer Instrumentenerkennung mit Support Vector Machines an der HAW Hamburg ((Dos08)) erstellt wurde.

Im Folgenden tauchen Bezeichnungen für bestimmte Aufzeichnungen von Gitarrenklängen in der Form *hoffmannL hense sonor S6-VI Bb3-4* auf. HoffmannL Hense beschreibt die Gitarre, sonor die Spielweise, S6-VI die Saite, Bb3-4 die Aufzeichnung.

### 1.3. Implementierung des Verfahrens

Das in dieser Arbeit vorgestellte Verfahren sowie weitere Implementierungen in vorangestellten Kapiteln wurden in Matlab (R2008a, TheMathworks,Natick,MA,USA) programmiert. Das Verfahren besteht aus drei Hauptroutinen:

- Separation nontonaler Signalanteil von Gesamtsignal
- Separate Analyse transienter sowie tonaler Signalanteil
- Separate Synthese transienter sowie tonaler Signalanteil

Das Verfahren wird in den entsprechenden Funktionen in Matlab durchgeführt, die in Abbildung 1.1, Abbildung 1.2 und Abbildung 1.3 dargestellt sind. Eine detaillierte Beschreibung der Funktionen erfolgt in dem Kapitel zu dem entwickelten Verfahren.

Die Analyse sowie Synthese erfolgen in parallelen Abläufen zum tonalen und transienten Signalanteil. Die Routine für den tonalen Signalanteil wurde aus früheren Arbeiten an der HAW Hamburg übernommen, ebenso die Separation des nontonalen Signalanteils des Gesamtsignals. Das Modell für den transienten Signalanteil wurde in dieser Arbeit implementiert.

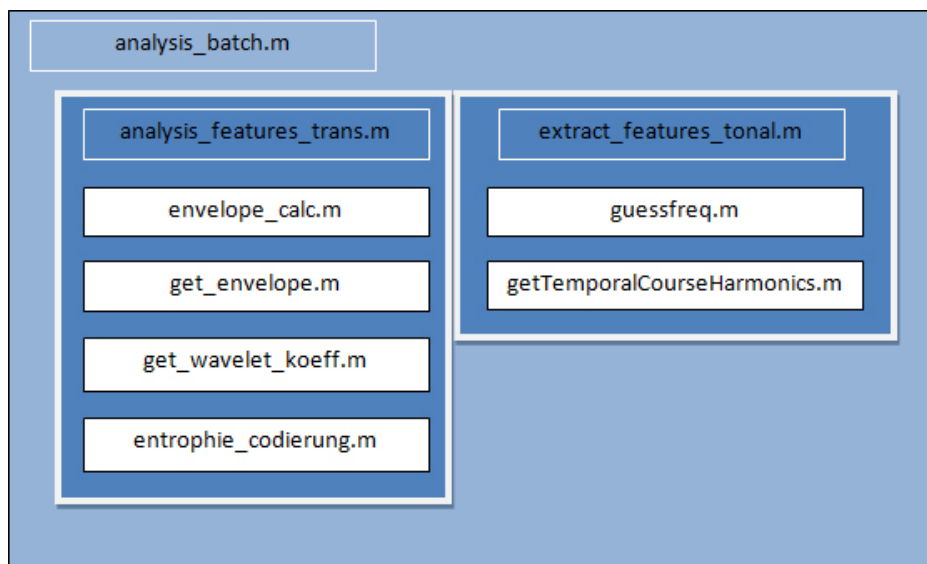


Abbildung 1.1.: Matlab-Dateien zum Analyse-Verfahren

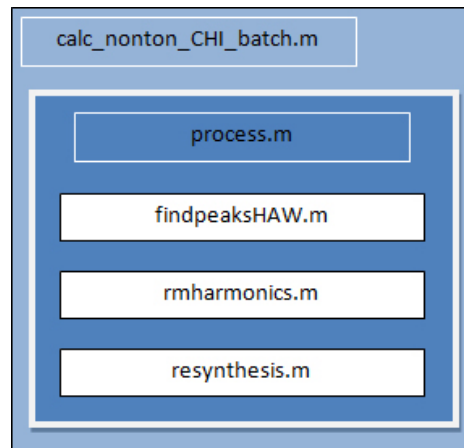


Abbildung 1.2.: Matlab-Dateien zur Separation des nontonalen Anteils

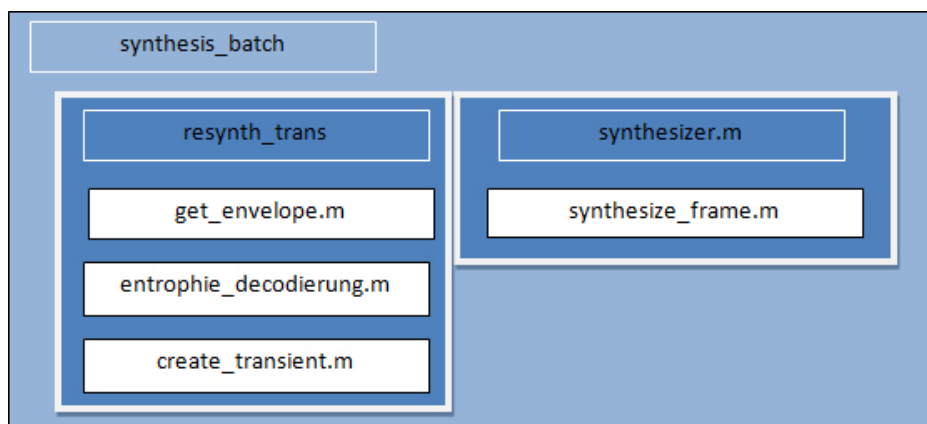


Abbildung 1.3.: Matlab-Dateien zum Synthese-Verfahren

## 2. Grundlagen der Wavelet-Transformation

Die Wavelet-Theorie hat ihren Ursprung in der Signaltheorie. Die Wavelet-Transformation umschreibt ein Verfahren zur Analyse und Synthese von Funktionen und erlaubt es, diese als Summe von groben Näherungen und immer feineren Details darzustellen. Gegenüber der Fourier-Transformation hat sie den Vorteil, Signale sowohl im Zeit- als auch im Frequenzbereich darstellen zu können. Darüber hinaus ist sie dabei nicht wie die Kurzzeit-Fourier-Transformation an eine feste Zeit-/Frequenz-Auflösung gebunden und dadurch wesentlich flexibler, kann also somit z.B. in Bereichen, in denen es nötig ist, schnelle Wechsel im Zeitbereich besser erfassen, was in Hinsicht auf die in dieser Arbeit behandelten Transienten von großem Vorteil ist. Die Bezeichnung „Wavelet“ kommt daher, dass bildlich gesprochen, ein Signal über die Zeit mit einer verschobenen sowie gestauchten/gestreckten „kleinen Welle“ abgetastet wird.

Innerhalb dieses Kapitels wird in Abschnitt 2.1 auf die Wavelet-Analyse als Ganzes eingegangen, in Abschnitt 2.2 auf die kontinuierliche Wavelet-Transformation, in Abschnitt 2.3 auf die diskrete Wavelet-Transformation, im abschließendem Abschnitt 2.4 auf verschiedene Anwendungsmöglichkeiten.

Der hauptsächliche Teil der folgenden Ausführungen bezieht sich auf Erkenntnisse aus dem Lehrbuch zur Wavelet-Transformation (Bae05). Wegweisend für die Entwicklung der Wavelet-Transformation, insbesondere in der diskreten Form, ist die Arbeit der Mathematikerin Ingrid Daubechies (Dau92).

### 2.1. Die Wavelet-Analyse

In der Wavelet-Analyse wird ein Signal im Gegensatz zur STFT (Kurzzeit-Fourier-Transformation) nicht in eine Zeit/Frequenz-Ebene transformiert, sondern in eine Zeit/Skalen-Ebene (Skalen und Frequenzen haben einen direkten Zusammenhang, wie im nächsten Abschnitt dargelegt wird). Hierbei ist es ebenso möglich längere Zeitintervalle mit hoher Frequenzauflösung bei niedrigen Frequenzen zuzulassen, als auch kürzere Zeitintervalle mit höherer Frequenzauflösung bei hohen Frequenzen.

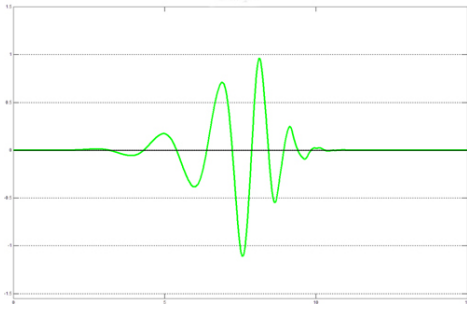


Abbildung 2.1.: Daubechies-8 Wavelet

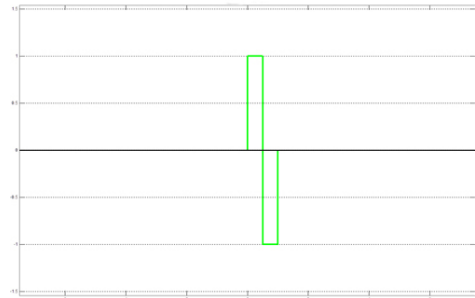


Abbildung 2.2.: Haar-Wavelet

Die in diesem Zusammenhang zunächst auftauchende Frage dürfte sein, was genau ein Wavelet eigentlich ist. Ein Wavelet ist eine endliche Wellenform, dessen Integral stets gleich Null ist. In Abbildung 2.1 ist ein Wavelet zu sehen, genau gesagt ein Daubechies-8 Wavelet, in Abbildung 2.2 ein Haar-Wavelet, ein noch etwas größeres Wavelets, sozusagen die Mutter aller Wavelets.

Man kann an dieser Stelle erkennen, dass durch die Endlichkeit und Irregularität der Form eines Wavelets (im Gegensatz zu einem Sinusoid) die daraus resultierende Analyse komplexer in ihrer Anschauung, Handhabung und Interpretation ist. Dennoch kann man sich die Wavelet-Analyse schon in weniger komplexen Gebilden zu Nutzen machen. Gerade in der Form der diskreten Wavelet-Transformation (DWT) ist eine höhere Anschauung gegeben, siehe dazu Abschnitt 2.3.

## 2.2. Die kontinuierliche Wavelet-Transformation

Die kontinuierliche Wavelet-Transformation lässt sich aus der Kurzzeit-Fourier-Transformation herleiten. Diese lässt sich in der Praxis als gefensterte Fourier-Transformation betrachten. Um neben der Frequenzlokalisierung bei nicht-stationären Signalen auch eine gute Zeitauflösung zu erreichen, wird eine Fensterfunktion  $g(t)$  gewählt und anstelle von  $f$  wird ein mit  $g(t)$  um  $a$  verschobenes, gefenstertes  $f$  Fourier-transformiert, wobei der Verschiebungsparameter  $a$  ebenfalls variabel sein kann. Die kontinuierlich transformierte ist also gegeben durch:

$$f^g(a, \xi) = \int_{-\infty}^{\infty} f(t)g(t - a)e^{-j2\pi\xi t} dt \quad (2.1)$$

Da dieser Transformation eine nach unten beschränkte Zeit-Auflösung inne wohnt ist die



Überlegung, die Fensterfunktion  $g$  noch mit einem Faktor  $s > 0$  zu strecken bzw. zu stauchen. Man definiert eine mit dem Faktor  $s$  gestreckte und in  $t$ -Richtung gestreckte und dann um  $a$  verschobene Version von  $g$ :

$$g_{s,a}(t) = s^{-\frac{1}{2}} g\left(\frac{t-a}{s}\right) \quad (2.2)$$

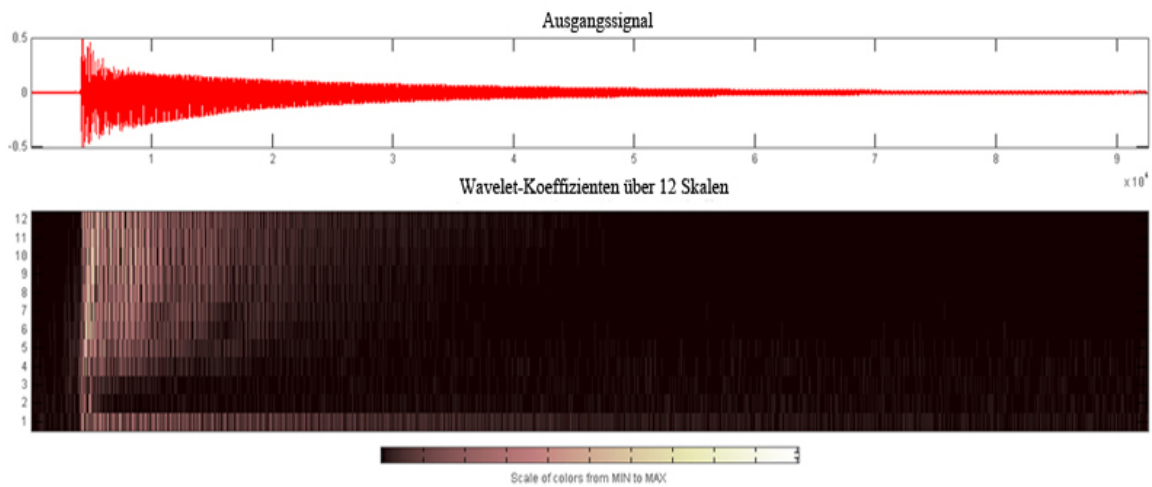
Der zusätzliche Streckungsfaktor  $s^{-\frac{1}{2}}$  dient Normierungszwecken.

Man hat in der Transformation nun also die drei Parameter  $a$ ,  $\xi$  und  $s$ . Die Idee der Wavelet-Transformation besteht nun darin, den Frequenzparameter  $\xi$  (gegeben in der Formel (2.1) durch  $e^{-j2\pi\xi}$ ) überflüssig werden zu lassen, indem  $g$  selbst schon eine Art Lokalisation im Frequenzbereich aufweist. Man wählt für  $g$  also nun eine kurze Welle, also ein Wavelet. Wird bei diesem Wavelet nun also der Skalierungsfaktor  $s$  verändert, es also gestaucht oder gestreckt, ändert sich dadurch nicht nur die Breite dieser Fensterfunktion, sondern auch die Frequenz der Welle. Wir ersetzen also in (2.1) den Faktor  $g(t-a)e^{-j2\pi\xi}$  durch  $s^{-\frac{1}{2}}g\left(\frac{t-a}{s}\right)$ , und die ursprüngliche Fensterfunktion  $g$  durch das für Wavelets übliche Zeichen  $\psi$ . Die kontinuierliche Wavelet-Transformation resultiert dann also in eine Funktion  $f$ , bezogen auf ein bestimmtes Wavelet  $\psi$ , mit den Variablen  $s$  und  $a$ :

$$f^\psi(a, s) = \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-a}{s}\right) dt = \int_{-\infty}^{\infty} f(t) \psi_{s,a}(t) dt \quad (2.3)$$

Anschaulich wird also ein Wavelet zu Beginn eines Signals mit dem Signalabschnitt gleicher Länge verglichen und ein Koeffizient  $C$  bestimmt, der aussagt, wie stark dieser Signalabschnitt mit dem Wavelet korreliert. Im nächsten Schritt wird das Wavelet am Signal um  $t = a$  verschoben und erneut ein Faktor-Koeffizient bestimmt. Dieser Schritt wird wiederholt, bis das Ende des Signals erreicht ist. Nun wird das Wavelet um den Faktor  $s$  gestreckt und die vorherigen Schritte wiederholt. Das Ergebnis ist eine  $(a, s)$ -Ebene mit verschiedenen Koeffizienten  $C$ , den sogenannte Wavelet-Koeffizienten. Grafisch lässt sich diese Ebene darstellen, indem man von kleinen  $C$  zu großen  $C$  dunklere bis hellere Farbtöne wählt. In Abbildung 2.3 sehen wir ein über 12 Skalen mit linearem Faktor und mit einem Daubechies-8 Wavelet (siehe Abbildung 2.1) abgetastetes Signal, in diesem Fall ein Gitarrenton, mit dem in dieser Arbeit gearbeitet wird.

Man kann sehen, dass das Daubechies-8 Wavelet mit dem Signal an den hellen Stellen in der gegebenen Skalierung am stärksten korreliert. Der Zusammenhang vom Skalierungsfaktor und Frequenz lässt sich grob wie folgt beschreiben:

Abbildung 2.3.: Beispiel der  $(a, s)$ -Ebene einer Wavelet-Transformation

niedriger Skalierungsfaktor  $a \implies$  komprimiertes Wavelet  $\implies$  hohe Frequenz  $f$   
 hoher Skalierungsfaktor  $a \implies$  gestrecktes Wavelet  $\implies$  niedrige Frequenz  $f$

Mann kann also sagen, dass bei den niedrigen Skalierungsfaktoren große Wavelet-Koeffizienten auf hohe Frequenzanteile schließen lassen, und umgekehrt bei den hohen Skalierungsfaktoren auf niedrige Frequenzanteile. Was genau die Koeffizienten über die Funktion aussagen, hängt stark von dem für die Transformation verwendeten Wavelet ab - die Möglichkeiten hier sind, wenn man nur die vorgestellten Daubechies- und Haar-Wavelets vergleicht, zu denen sich noch eine große Menge weiterer Vertreter gesellen, sehr breit gefächert.

### 2.3. Die diskrete Wavelet-Transformation

Die diskrete Wavelet-Transformation beruht auch auf dem Wunsch, eine Funktion  $f$  als Linearkombination von Grundfunktionen  $\psi_k$  darzustellen:

$$f = \sum_k c_k \psi_k \quad (2.4)$$

Am besten nähert man sich der diskreten Wavelet-Transformation vom Verfahren des Subband Coding aus. Ein diskretes Signal kann als Abtastung eines kontinuierlichen, bandbe-

grenzten Signals betrachtet werden, bei einer Abtastfrequenz von 1 mit den Frequenzen  $|\xi| \leq \frac{1}{2}$ . Wenn dieses Signal nun in die Analysebank einer PR-Filterbank nach Abbildung 2.4 mit entsprechenden Filtergängen eingespeist wird, entstehen zwei Teile - ein hochfrequentes Signal  $D(s)$  sowie ein niederfrequentes Signal  $A(s)$ .  $D(s)$  enthält im wesentlichen Anteile mit Frequenzen zwischen  $|\xi| = \frac{1}{4}$  und  $|\xi| = \frac{1}{2}$ ,  $A(s)$  Anteile mit  $|\xi| \leq \frac{1}{4}$ .

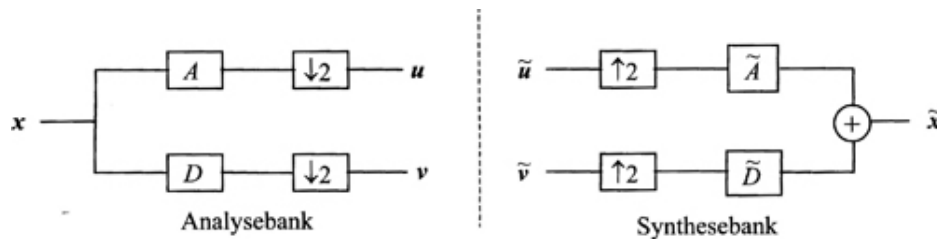


Abbildung 2.4.: 2-Kanal-PR-Filterbank (Bild aus (Bae05))

Durch das den Filtern folgende Downsampling zu  $u_1 = (\downarrow 2)A(s)$  und  $v_1 = (\downarrow 2)D(s)$  werden die Frequenzen in  $A(s)$  wieder auf den Bereich  $|\xi| \leq \frac{1}{2}$  normalisiert. Man wendet also im nächsten Schritt die PR-Filterbank auf  $u_1$  an. Daraus resultiert nun ein Signal  $v_2 = (\downarrow 2)D(u_1)$ , welches (nach Rekonstruktion) Frequenzanteile von  $s$  zwischen  $|\xi| = \frac{1}{8}$  und  $|\xi| = \frac{1}{4}$  enthält, sowie ein Signal  $u_2 = (\downarrow 2)A(u_1)$ , welches Frequenzanteile  $|\xi| \leq \frac{1}{8}$  darstellt. Dieses Verfahren wird in  $k$  Schritten wiederholt, und wir erhalten Signale  $v_1, v_2, v_3, v_4, v_5, \dots, v_k, u_k$ , wobei  $v_i$  immer Anteile vom Ursprungssignal  $s$  im Frequenzband  $\frac{1}{2^{i+1}} \leq |\xi| \leq \frac{1}{2^i}$  darstellt. Dieses Verfahren bezeichnet man als *Subband Coding*. Man kann sich die Approximation des Ursprungssignals in eine Zerlegung in einen Faktoren-Baum, im Zusammenhang mit der Wavelet-Transformation Wavelet-Tree genannt, nach Abbildung 2.5 vorstellen.

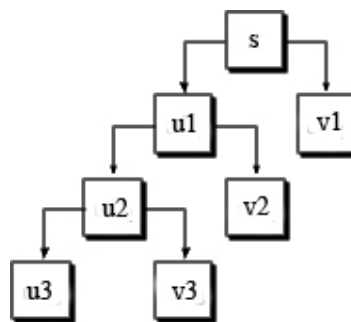


Abbildung 2.5.: Wavelet-Tree

Neben jener des Subband-Coding ist die Theorie der orthogonalen Multiskalen-Analyse (MSA) für die diskrete Wavelet-Transformation von zentraler Bedeutung. Durch Verwendung

einer Skalierungsfunktion  $\varphi$ , die man sich als kurzen, mehrheitlich positiven Impuls vorstellen kann, kann eine Funktion an beliebiger Stelle in beliebiger Auflösung betrachtet werden. Nach der Wavelet-Theorie (siehe (Bae05)) arbeiten wir in einem Funktionenraum  $L^2(\mathbb{R})$ , vergleichbar einem Vektorraum, der durch die Wavelet-Orthonormalbasen aufgestellt wird. Der Wunsch nach orthogonalen Systemen ergibt sich u.a. aus der daraus folgenden leichten Bestimmung der Koeffizienten in (2.4).

Um nun eine  $L^2$  Funktion  $f$  in einer Skala  $2^m$  darzustellen, soll sie möglichst genau als Linearkombination einer mit  $2^n$  gestreckten und  $n2^m$  verschobenen Version von  $\varphi$  geschrieben werden:

$$f \approx \sum_{m \in \mathbb{Z}} u_{m,n} \varphi_{m,n} \quad \text{möglichst genau bezüglich der } L^2\text{-Norm,} \quad (2.5)$$

wobei gilt

$$\varphi_{m,n}(t) = s^{-\frac{m}{2}} \varphi(2^{-m}t - n). \quad (2.6)$$

An die Skalierungsfunktion  $\varphi$  lassen sich nun drei Forderungen knüpfen. Zunächst stellen wir die „Orthonormalitätsbedingung“

$$\langle \varphi_{0,n}, \varphi \rangle = \delta_{0,n} \quad \text{für } n \in \mathbb{Z}. \quad (2.7)$$

$\delta_{0,n}$  ist das sogenannte Kronecker-Symbol (= 1 für  $n = k$ , sonst immer 0). Diese Bedingung führt dazu, dass die  $\varphi_{m,n}$  ( $n \in \mathbb{Z}$ ) bei fester Skala  $m$  eine orthonormierte Familie bilden, was eine besonders einfache Bestimmung der Koeffizienten  $u_{m,n}$  zur Folge hat, für die beste  $L^2$ -Approximation einer Funktion  $f$  also gilt:

$$u_{m,n} = \langle \varphi_{m,n}, f \rangle = \int_{-\infty}^{\infty} \varphi_{m,n}(t) f(t) dt. \quad (2.8)$$

Die zweite Forderung ist die der „2-Skalenrelation“. Jede Skalierungsfunktion  $\varphi_{m,n}$  ( $n \in \mathbb{Z}$ ) erzeugt einen Funktionenraum  $V_m$ , auch bezeichnet man die beste Approximation von  $f$  in einer Skala  $2^m$  mit  $A_m f$ . Es wird nun die Forderung aufgestellt, dass eine Approximation  $A_m f$ , die in einer Skala  $2^m$  exakt dargestellt werden kann, auch in einer feineren Skala  $2^p$  darstellbar ist, d.h. für  $p < m$  soll  $V_m$  in jedem  $V_p$  erhalten sein. Dazu wird verlangt, dass es Zahlen  $h_k$  ( $k \in \mathbb{Z}$ ) gibt, für die gilt:

$$\begin{aligned}\varphi &= \sum_{k \in \mathbb{Z}} h_k \varphi_{-1,k}, \quad \text{das heißt} \\ \varphi(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2t - k)\end{aligned}\tag{2.9}$$

Hinzu kommt als dritte Forderung noch die „Mittlungseigenschaft“, d.h.  $\varphi$  muss integrierbar sein und

$$\int_{-\infty}^{\infty} \varphi(t) dt = 1.\tag{2.10}$$

Unter der Voraussetzung von (2.8), (2.9) und (2.10) ist also eine orthogonale Multiskalen-Analyse gegeben, wenn eine Funktion  $\varphi$  sowie Zahlen  $h_k (k \in \mathbb{Z})$  vorliegen, die diese erfüllen. Aus ihnen ergeben sich zwei direkte weitere Eigenschaften, die eine direkte Brücke zur Praxis, der orthogonalen PR-Filterbank, schaffen:

$$\sum_{k \in \mathbb{Z}} h_k = \sqrt{2}\tag{2.11}$$

$$\sum_{k \in \mathbb{Z}} h_k h_{k+2n} = \delta_{0,n} \quad \text{für alle } n\tag{2.12}$$

Aus (2.12) ergibt sich nun eine Parallelität zu den Eigenschaften einer orthogonalen PR-Filterbank (nach Abbildung 2.4), deren Eigenschaften und daraus folgende Algorithmen zur Bestimmung der vier Filter (aus dem zu definierendem Filter  $\tilde{A}$ ) sich nach Kapitel 3.3 aus (Bae05) zu

$$A(z) = \tilde{A}(z^{-1})\tag{2.13}$$

sowie

$$\tilde{D}(z) = -z^{-l} \tilde{A}(-z^{-1}) \quad , \quad D(z) = (-z)^l \tilde{A}(-z)\tag{2.14}$$

ergeben. Es folgt nun, dass ein Digitalfilter  $H$  mit einer Impulsantwort  $h = (h_k)_{k \in \mathbb{Z}}$  als Tiefpassfilter einer solchen orthogonalen PR-Filterbank qualifiziert ist.

Nun bilden die Funktionen  $\varphi_{m,n}$  aus zwei Gründen noch keine für die Wavelet-Transformation benötigte Orthonormalbasis des gesamten Funktionenraumes  $L^2(\mathbb{R})$ . Funktionen verschiedener Skalen  $2^m$  brauchen nicht orthogonal zueinander sein und

Funktionen gröberer Skalen lassen sich wegen (2.9) als Linearkombinationen der Funktionen feinerer Skalen schreiben. Die  $\varphi_{m,n}$  einer festen Skala  $2^m$  ergeben jedoch eine Orthonormalbasis von  $V_m$ , und  $V_m$  ist, wie bereits besprochen, in  $V_{m-1}$  enthalten. Ziel ist es also  $\varphi_{m,n}$  ( $n \in \mathbb{Z}$ ) zu einer Orthonormalbasis von  $V_{m-1}$  zu ergänzen. Aus dieser Überlegung ergeben sich nun eine Funktion  $\psi$  und ein Faktor  $g_k$ , beide in Abhängigkeit zu der bisherigen Skalierungsfunktion  $\varphi$  und dem Faktor  $h_k$  wie folgt:

$$\begin{aligned} \psi &= \sum_{k \in \mathbb{Z}} g_k \varphi_{-1,k} \quad \text{mit} \quad g_k = (-1)^k h_{l-k}, \quad \text{also} \\ \psi(t) &= \sqrt{2} \sum_{k \in \mathbb{Z}} (-1)^k h_{l-k} \varphi(2t - k) \end{aligned} \quad (2.15)$$

Für jede  $L^2$ -Funktion  $f$  gilt also nun

$$f = \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} v_{m,n} \psi_{m,n} \quad (2.16)$$

Aus der orthogonalen MSA und dem Subband Coding lässt sich nun direkt die schnelle Wavelet-Transformation abbilden. Aus den Relationen

$$\varphi_{m+1,n} = \sum_{k \in \mathbb{Z}} h_{k-2n} \varphi_{m,k} \quad \text{ergibt sich aus (2.9)} \quad (2.17)$$

und

$$\psi_{m+1,n} = \sum_{k \in \mathbb{Z}} g_{k-2n} \varphi_{m,k} \quad \text{ergibt sich aus (2.15)} \quad (2.18)$$

ergeben sich mittels Skalarproduktbildung mit  $f$  die Beziehungen

$$u_{m+1,n} = \sum_{k \in \mathbb{Z}} h_{k-2n} u_{m,k} \quad , \quad v_{m+1,n} = \sum_{k \in \mathbb{Z}} g_{k-2n} u_{m,k} \quad (2.19)$$

Mit diesen Formeln lassen sich nun alle Wavelet-Koeffizienten nach dem Schema in Abbildung 2.6 rekursiv errechnen.

$\mathbf{u}_m$  bedeutet hier  $(u_{m,n})_{n \in \mathbb{Z}}$  etc. und  $H'$  sowie  $G'$  bezeichnet die Operationen

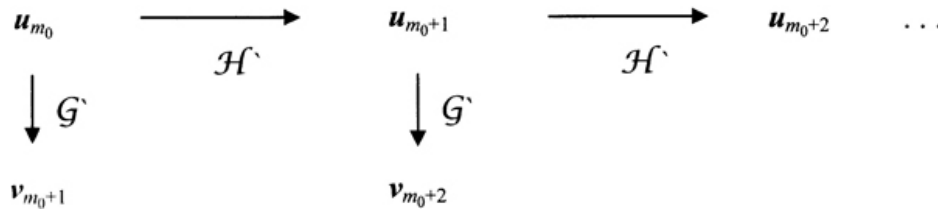


Abbildung 2.6.: Wavelet-Transformation (Bild aus (Bae05))

$$H'(x)_n = \sum_{k \in \mathbb{Z}} h_{k-2n} x_k \quad , \quad G'(x)_n = \sum_{k \in \mathbb{Z}} g_{k-2n} x_k. \quad (2.20)$$

Die Rekonstruktion erfolgt nach dem Schema in Abbildung 2.7. die zwei Operationen  $H$  und  $G$  sind definiert als

$$H(x)_n = \sum_{k \in \mathbb{Z}} h_{n-2k} x_k \quad , \quad G'(x)_n = \sum_{k \in \mathbb{Z}} g_{n-2k} x_k. \quad (2.21)$$

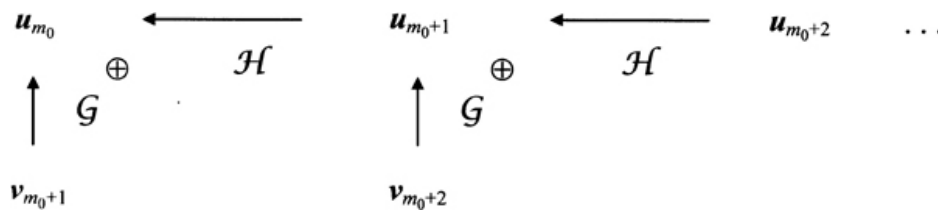


Abbildung 2.7.: Wavelet-Rekonstruktion (Bild aus (Bae05))

Die schnelle Wavelet-Transformation lässt sich nun direkt in die digitale Signalbearbeitung übertragen, und zwar zu einer assoziierten Filterbank. Die erste Operation in (2.20) bedeutet, dass die Folge  $x$  zunächst mit der Folge  $(h'_k)_{k \in \mathbb{Z}}$  gefaltet und anschließend dezimiert (=Downsampling) wird. (2.21) bedeutet eine Faltung mit  $(h_k)_{k \in \mathbb{Z}}$  und anschließender Spreizung (=Upsampling). Das führt direkt zu einer orthogonalen PR-Filterbank wie in Abbildung 2.4 mit der auf (2.20) sowie (2.21) bezogenen Gleichsetzung von  $A = H'$ ,  $\tilde{A} = H$ ,  $D = G'$  und  $\tilde{D} = G$ .

Es wurde also nun ein Schema gefunden, unter dem sich unter Wahl einer geeigneten Skalierungsfunktion  $\varphi$  eine einer orthogonalen MSA assoziierten Filterbank erstellen lässt, mit deren Hilfe sich die Wavelet-Koeffizienten nach dem Muster des Subband Coding erstellen lassen und aus denen umgekehrt wieder das Ursprungssignal rekonstruiert werden kann. Außen vor wird in dieser Ausführung die biorthogonale Multiskalen-Analyse gelassen, da diese im späteren Verlauf der Arbeit keine Anwendung findet. Sie liefert die Möglichkeit, symmetrische Filterbanken zuzulassen, zum Beispiel die biorthogonalen Spline-Filter. Mehr zu diesem Thema findet man in (Bae05).

Die eigentliche Arbeit der diskreten Wavelet-Transformation besteht (mit den hier vorgeestellten Werkzeugen) in der Konstruktion geeigneter Wavelet-Filter. Hier wird nur kurz auf die schon eingangs erwähnten und in dieser Arbeit verwendeten Daubechies-Wavelets von Daubechies eingegangen, welche ein Verfahren zur direkten Entwicklung von dazugehörigen Daubechies-Filtern vorgestellt hat. Im orthogonalen Fall, also wenn  $H = \tilde{H}$  gilt, ist ein FIR-Filter  $H$  ein Daubechies-Filter, wenn das Produktfilter  $M(z) = H'(z)H(z) = H(z^{-1})H(z)$  die Eigenschaft  $M(z) + M(-z) = 2$  besitzt und  $H$  bei  $z = -1$  eine Nullstelle möglichst hoher Ordnung  $p$  hat. Durch das von Daubechies entwickelte mathematische Verfahren lassen sich nun nach einem festen Schema Daubechies-Filter für verschiedene Ordnungen  $p$  konstruieren (in Abbildung 2.1 sehen wir ein Daubechies-Wavelet mit  $p = 8$ ).

Erwähnenswert sind an dieser Stelle, da später in der Arbeit auch angewendet, die sogenannten Wavelet-Pakete. Wie zuvor beschrieben wurde wird bei der schnellen Wavelet-Transformation bzw. dem Subband-Coding die PR-Filterbank immer rekursiv auf den Tiefpasskanal angewendet, anschaulich in Abbildung 2.5 dargestellt. Nun ist es möglich, die Filterbank auch auf den Hochpasskanal anzuwenden, es entstehen mit jeder Skala  $k$  also  $2^k$  neue Koeffizienten-Folgen. Die Frage ist nun, mit welchen Basisfunktionen diese neuen Koeffizienten zu multiplizieren sind, und was für Eigenschaften diese haben.

Zwecks Untersuchung der Wavelet-Pakete wird die besprochene orthogonale Multiskalen-Analyse zu Grunde gelegt, mit der feinsten Skala  $2^0 = 1$  und Skalierungsfunktion bzw. Wavelet

$$\varphi_{1,0} = \sum_k h_k \varphi_{0,k} \quad \psi_{1,0} = \sum_k g_k \psi_{0,k} \quad (2.22)$$

mit Filterkoeffizienten  $h_k$  und  $g_k$  der assoziierten Filterbank.

Ganzzahlig verschobene Kopien  $\varphi_{0,n}$  von  $\varphi$  bilden im Funktionenraum  $V_0$  eine Orthonormalbasis, gleichzeitig ist darin eine weitere Orthonormalbasis enthalten, zusammengesetzt aus den Basisfunktionen  $\varphi_{1,n}$  von  $V_1$  und den entsprechend verschobenen Wavelets  $\psi_{1,n}$ . Diese können auch als Orthonormalbasis eines Unterraumes  $W_1$  von  $V_0$  aufgefasst werden.  $V_0$  wird also in die Unterräume  $V_1$  und  $W_1$  zerlegt, man schreibt auch  $V_0 = V_1 \oplus W_1$ . Das wiederholte Einspeisen beider Ausgabekanäle der Analyse-Filterbank führt also zu einer Aufteilung der Funktionenräume nach Abbildung 2.8.

Dabei gilt  $V_0 = W_{[0,0]}$ , und für eine Skala  $2^m$

$$V_0 = W_{[m,0]} \oplus W_{[m,1]} \oplus W_{[m,2]} \oplus \dots \oplus W_{[m,2^m-1]}. \quad (2.23)$$

Jeder der Räume  $W_{[m,i]}$  hat eine Orthonormalbasis, die aus den um  $n2^m$  ( $n \in \mathbb{Z}$ ) verschobenen Kopien  $\psi_{[m,i],n}$  einer Funktion  $\psi_{[m,i]}$  besteht. Diese Funktion nennt man Wavelet-Pakete, man erhält sie nach (angefangen mit  $\psi_{[0,0]} = \varphi$ )



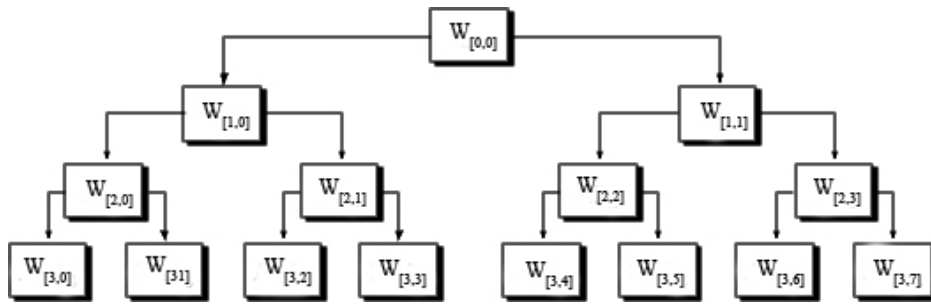


Abbildung 2.8.: Wavelet-Pakete

$$\Psi_{[m+1,2i]} = \sum_k h_k \Psi_{[m,i],k} \quad , \quad \Psi_{[m+1,2i+1]} = \sum_k g_k \Psi_{[m,i],k}. \quad (2.24)$$

Für  $m \geq 1$  gilt  $W_{[m,0]} = V_m$ ,  $W_{[m,1]} = W_m$  und  $\psi_{[m,0]} = \varphi_{m,0}$ ,  $\psi_{[m,1]} = \psi_{m,0}$ . Zwei Pakete mit gleichem Index  $i$  sind also gestreckte Versionen voneinander. Der gesamte unter  $W_{[1,0]}$  befindliche Teilbaum kann also als gestrecktes Abbild des gesamten Baumes aufgefasst werden.

In Abbildung 2.9 sehen wir die acht Wavelet-Pakete auf der dritten Stufe des Baumes, ausgehend von einer Haarschen Basis. Die ersten vier sind also gestreckte Versionen der zweiten Stufe, die ersten beiden der ersten Stufe. Man sieht, dass mit diesen Paketen eine bessere Frequenzlokalisierung als bei einer einfachen, schnellen Wavelet-Transformation möglich ist, auch fällt aber auf, dass die Nummerierung im binären Baum nicht der von aufsteigenden Frequenzen entspricht.

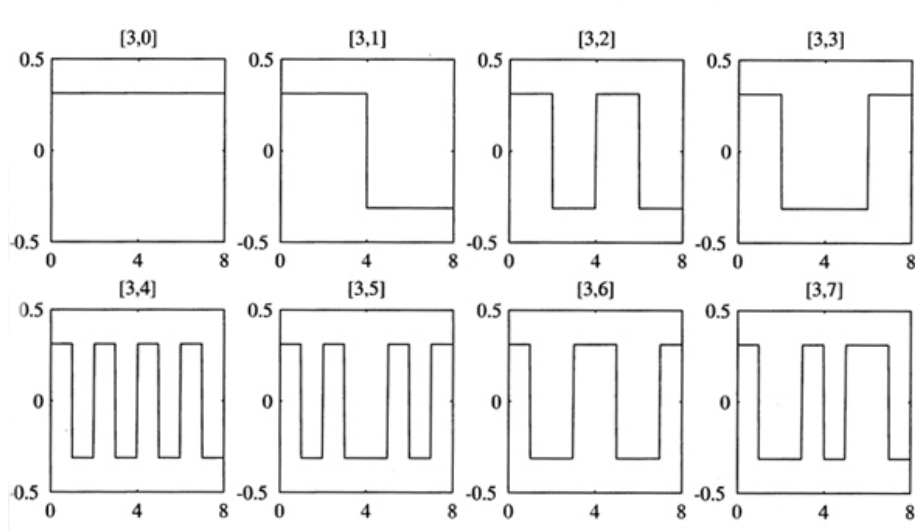


Abbildung 2.9.: Haar-Wavelet-Pakete der 3. Baum-Stufe (Bild aus (Bae05))

Möchte man nun ein Signal in Wavelet-Pakete zerlegen, muss entschieden werden, ob bei jedem entstehenden neuen Knoten, also Unterraum, eine weitere Aufteilung durch die Anwendung der Analyse-Filterbank erfolgen soll, denn der Baum muss nicht zwingend überall in die gleiche Tiefe verfolgt werden. Die in Wavelet-Pakete zerlegten Signale lassen sich stets mit den Enden einer jeden möglichen Verzweigung rekonstruieren, an Abbildung 2.8 angelehnt ließe sich das Signal also auch mittels der Koeffizientenfolgen  $(x_{[m,i]}$  bezüglich der Basisfunktion von  $W_{[m,i]}$ )  $x_{[2,0]}, x_{[2,1]}, x_{[2,3]}, x_{[3,4]}$  sowie  $x_{[3,5]}$  rekonstruieren.

Für die Suche nach einer optimalen orthonormalen Basis von  $W_{[m,i]}$  lässt sich ein Such-Algorithmus entwickeln, anhand einer adaptiven Zerlegung der zu zerlegenden Funktion  $f$ . Anhand dieses Algorithmus kann man eine optimale Baumstruktur für ein gewünschtes Signal ermitteln (mehr zum Algorithmus selbst in (Bae05)). Bei einem Chirp-Signal führt das z.B. zu einer Struktur mit möglichst vielen Ästen in beide Richtungen (hohe Frequenzauflösung), bei einer Funktion mit großen Sprüngen zu einem einfachen Ast ähnlich der Wavelet-Transformation (hohe Zeitauflösung). Ein großer Vorteil der Wavelet-Pakete besteht also in der Möglichkeit, je nach den vom Ursprungssignal geforderten Vorgaben, ein genaueres Mittel zwischen Frequenz- und Zeitauflösung zu finden. Ein Nachteil ist die Codierung der ermittelten Daten - neben den Koeffizienten muss auch die Information über die zugehörige Baumstruktur übermittelt werden, bei der einfachen Wavelet-Transformation reicht die Information der Rekursionstiefe.

## 2.4. Anwendung der Wavelet-Transformation

Die Möglichkeiten zur Anwendung der in diesem Kapitel grob vorgestellten Wavelet-Theorie und -Transformation sind breit gefächert. Dabei liegt ihre Stärke weniger wie bei der Fourier-Transformation in einer anschaulichen Darstellung von Funktionen (vor allem stationärer Signale), da die Zeit/Skalen-Ebene zwar wie erwähnt die Konzentration hoher bzw. niedriger Frequenzen für bestimmte Zeitwerte aufzeigt, aber das Signal nicht übersichtlich in seine Frequenzanteile zerlegt.

Eine primäre und auch in dieser Arbeit Beachtung findende kommende Anwendung der Wavelet-Transformation ist die Datenkompression. Sie wird vor allem in der Bildbearbeitung benutzt, kann aber auch wie hier auf Audio-Signale angewendet werden. Nach der Zerlegung eines Signals mittels der Wavelet-Transformation in seine Wavelet-Koeffizienten diverser Skalen und die Approximations-Koeffizienten der höchsten Skala (nach Abbildung 2.5) kann das Ursprungssignal mittels dieser wieder vollständig rekonstruiert werden. Nun haben diese Koeffizienten immer genau die gleiche Datenmenge wie das Ausgangssignal, jedoch ist es nun möglich, diese Koeffizienten zu quantisieren, also auf vorgegebene Werte zu runden. Die einfachste Vorgehensweise ist dabei das Setzen eines bestimmten Thresholds - jegliche Koeffizienten-Werte kleiner dieses gewählten Wertes werden gleich Null gesetzt.

Für eine reale Reduktion der Datenmenge müssen die quantisierten Koeffizienten noch einer Codierung, einer sogenannten Entropie-Codierung, unterworfen werden. Mehr dazu in der Anwendung einer solchen unter Kapitel 4.3.1.

Das nun aus diesen quantisierten Koeffizienten rekonstruierte Signal führt zu einem Verlust von Informationen, wie ja auch ein mittels Fourier-Transformation approximiertes und mit einer abgebrochenen Fourier-Reihe rekonstruiertes Signal verlustbehaftet ist. Es ist also nun verlangt, Transformation und Quantisierung so zu wählen, dass möglichst wenig Informationsverlust bei möglichst hoher Kompressionsrate resultiert. Es entstehen drei Baustellen:

1. Verwendung einer einfachen Transformation oder von von Wavelet-Paketen, und natürlich die Wahl einer geeigneten Wellenform für das Wavelet, dem die Basis zu Grunde liegt.

2. Die Quantisierung, also ob eine uniforme skalare Quantisierung gewählt wird (unabhängiges Runden jedes Koeffizienten an einen bestimmten Wert bzw. Threshold), oder eine vektorielle Quantisierung, bei dem die Koeffizienten abhängig von ihren nächstliegenden Koeffizienten gerundet werden. Man spricht auch von einem harten oder weichen Threshold. Zusätzlich können auch verschieden große Quantisierungsschritte gewählt werden, und die Koeffizienten verschiedener Skalen mit unterschiedlichen Thresholds belegt werden, ähnlich z.B. höherer Quantisierungsschritte bei z.B. höheren Frequenzen bei der Quantisierung fourier-transformierter Signale.

3. Die Entropie-Codierung, dessen Ziel es ist, die quantifizierten Koeffizienten möglichst effektiv binär zu codieren. Hierzu findet sich in der Literatur eine Fülle von Möglichkeiten, eine einfache Methode findet unter Kapitel 4.3 Anwendung.

Weitere Anwendungsmöglichkeiten der Wavelet-Transformation sind das Denoising, also dem Unterdrücken von Rauschen, ein wichtiger Zweig, der jedoch in dieser Arbeit keine Rolle spielt. Bekannt und weit erforscht sind auch Möglichkeiten der Kanten-Erkennung, der Bereich der Randwertprobleme, und auch die Lokalisation von Singularitäten, auf die unter Kapitel 3 noch eingegangen wird. Allgemein ist die Wavelet-Transformation als im Vergleich junges Gebiet noch ein sehr aktiv beforschtes, und in den letzten Jahrzehnten haben sich eine Fülle von Möglichkeiten in ihrer Anwendung in den Verschiedensten Bereichen der Mathematik- und Ingenieurwissenschaften aufgetan.

## 3. Wavelets in der Klang-Modellierung

Fortschreitend mit der mathematischen Entwicklung der Wavelet-Theorie findet diese auch vermehrt ihren Platz in der digitalen Verarbeitung von Audio-Signalen. Die Anwendungsmöglichkeiten umfassen dabei ein weites Spektrum, in dem vor allem die besonderen Eigenschaften der Wavelet-Transformation in Bezug auf Zeit-/Frequenzauflösung, Komprimierung, De-Noising, und ihrem Verhalten bei außergewöhnlichen Ereignissen innerhalb eines Signals (z.B. Sprünge) ausgenutzt werden. Mit Hilfe der Wavelet-Transformation werden sowohl reine Audio-Coder verwirklicht ((Nav10), (VR06) oder (vL03)), oder die Erkennung bestimmter Musikinstrumente ((Nis07)), als auch, wie für das Thema dieser Arbeit relevant, Modellierungs-Verfahren von Audio-Signalen im allgemeinen, insbesondere dabei von transienten Signalanteilen.

Im Bezug auf die für die Klang-Analyse und Synthese oft angewendete Aufteilung eines Audiosignals, in tonalem-, transienten- und ggf. Noise-Anteil (Residue), worauf bereits in der Einleitung eingegangen wurde, spielt die Wavelet-Transformation insbesondere bei dem transienten Teil aufgrund ihrer Eigenschaften eine wichtige Rolle.

### 3.1. Wavelets in der Tonalen-Modellierung

Wie in den meisten Arbeiten zu diesem Thema ausgeführt (z.B. (DT02) oder (Nsa07)) ist die Anwendbarkeit der Wavelet-Transformation in Bezug auf tonale, also quasi-stationäre Signale, von geringerer Bedeutung. Hier hat sich die Anwendung von Approximationen mit trigonometrischen Basen wie der Fourier-Transformation, der Kurzzeit-Fourier-Transformation oder der Cosinus-Transformation bewährt, und ist in Bezug auf die Eigenschaften kontinuierlicher Signale dafür wesentlich besser geeignet.

Tonale Signalanteile haben die Eigenschaft, keine plötzlichen Veränderungen in kurzen Zeitabschnitten zu beinhalten, die Frequenzauflösung der Transformation ist also von wesentlich entscheidenderer Bedeutung, die Zeitauflösung der DTFT ist hier also zumeist ausreichend. Der Vorteil der Wavelet-Transformation gegenüber der Kurzzeit-Fourier-Transformation, über die Signallänge nicht an eine feste Zeit-/Frequenzauflösung gebunden zu sein (siehe Kapitel 2), kommt bei stationären bzw. quasi-stationären Signalen logischerweise nicht sonderlich zu tragen. Auch ist die Anschaulichkeit der Fourier-Transformation weitaus umfangreicher - anhand der Sinusoide kann genau abgelesen bzw. bei einer freien Modellierung auch bestimmt

Threshold	Komprimierungsrate [%]	Klang-Qualität
0,1	91,28	starke Artefakte, klares „Schnarren“
0,05	91,28	mittlere Artefakte, mittleres „Schnarren“
0,01	91,28	mittlere Artefakte, mittleres „Schnarren“
0,005	89,17	kleinere Artefakte, kleines „Schnarren“
0,001	83,72	keine Qualitäts-Minderung hörbar

Tabelle 3.1.: Vergleich Qualität Wavelet-Transformation und Komprimierung von tonalem Signal

werden, wo sich Signalanteile im Spektrum befinden, während die Wavelet-Transformation mit ihrer Zeit-Skalen-Auflösung und ihren Koeffizienten lediglich auf bestimmte Signalanteile in bestimmten Frequenzregionen schließen lässt. Wavelet-Pakete mit ihrer genaueren Frequenzauflösung können hier Abhilfe schaffen, nur ist hier der zu betreibende Aufwand und die Komplexität der Darstellung erheblich höher. Gerade bei der Modellierung von tonalen Musiksignalen, also Tönen, ist eine direkte Anschaulichkeit des genauen Frequenzverhaltens der Grundschwingung sowie der Obertöne gewünscht und notwendig, wenn diese z.B. im weiterführender Bearbeitung im Sinne der Tonleiter verändert werden sollen.

Bei komplexeren Modellierungs-Verfahren, wenn z.B. der tonale Anteil direkt aus einem Signal gewonnen werden soll, und die Qualität einer einfachen STFT nicht ausreicht, wird dabei oft auf eine mehr oder minder stark modifizierte diskrete Cosinus-Transformation zurückgegriffen ((DT02), (VT97)).

Anhand einfacher Versuchsdurchführungen lassen sich ein paar dieser Sachverhalt verdeutlichen. Der synthetisierte, tonale Anteil des Gitarrenklanges *hoffmannL hense sonor S1-VI Bb5-1* (Erklärung der Referenz siehe Kapitel 1) wird einer schnellen, diskreten Wavelet-Transformation unterzogen, die resultierenden Koeffizienten einem harten Threshold  $t_h$  unterzogen, und das daraus rekonstruierte Signal untersucht. Es soll zunächst ermittelt werden, bei welchen Komprimierungsraten der Wavelet-Koeffizienten es noch möglich ist, dieses rein tonale Signal ohne hörbare Störungen wie Artefakte und ähnliche Klangverfälschungen zu rekonstruieren. Als Wavelet-Basis für die Transformation wurde ein Daubechies-8 Wavelet gewählt, das über eine Sinus-ähnliche Form verfügt (siehe Abbildung 2.1). Die Wavelet-Transformation wurde bis zu einer Skalentiefe von  $k = 10$  durchgeführt. Eine Beschreibung der Höreindrücke sind in der Tabelle 5.3 zu finden.

In Abbildung 3.1 ist das Spektrum des mittels STFT und das mittels diskreter Wavelet-Transformation (bei einem Threshold von  $t_h = 0,1$ ) modellierten Signalanteils dargestellt. Deutlich zu erkennen sind die spektralen Anteile, die durch die DWT zwischen dem Grundton und den Obertönen entstehen und das Signal mit hörbaren Artefakten stören.

Zum Vergleich: das synthetische, tonale Signal wurde in (vC09) mit einer STFT modelliert, hierzu wurden bei einem Signal von  $N_{samples} = 92611$  Samples eine Datenmenge von

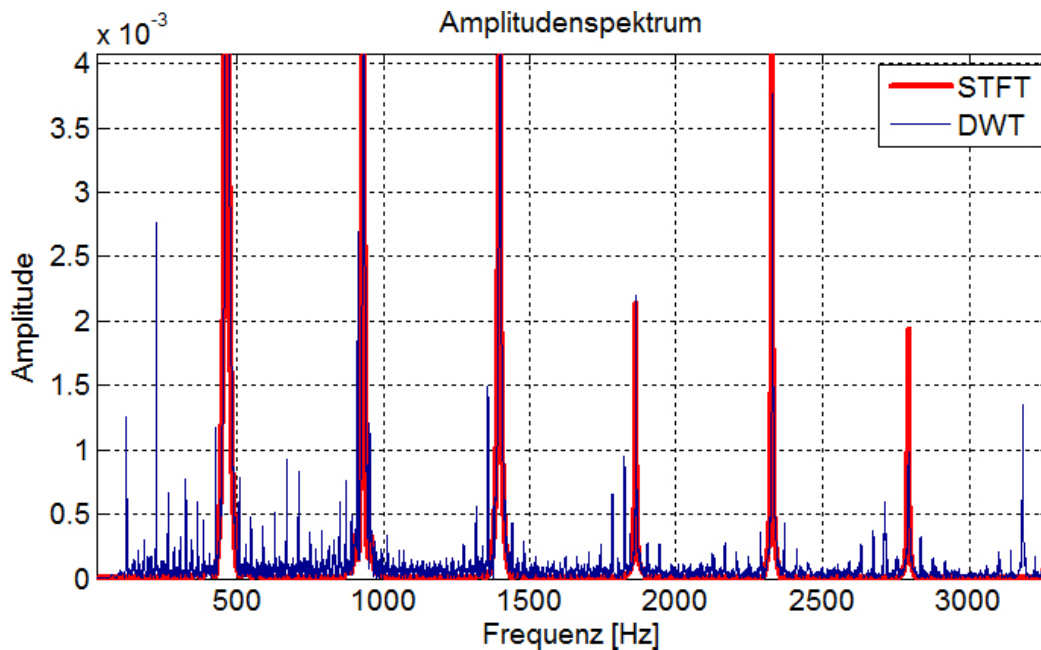


Abbildung 3.1.: Vergleich Spektrum modelliertes, tonales Signal STFT und DWT

$N_{data} \approx 3538$  zur Synthese benötigt, was einer Komprimierungsrate von  $Kr_{stft} \approx 96,18\%$  entspricht. Die angewandte, schnelle Wavelet-Transformation benötigt also für eine vergleichbare Klangqualität eine weitaus höhere Datenmenge, für eine Synthese eines tonalen Signalanteils der Gitarrenklänge.

Noch drastischer macht sich das wenig aussagekräftige Verhalten der Wavelet-Transformation in Bezug auf stationäre Signale anhand eines simplen Beispiels deutlich. In Abbildung 3.2 sind drei Verläufe von Wavelet-Koeffizienten nach einer Transformation einfacher Sinus-Signale mit den Frequenzen  $1\text{Hz}$ ,  $10\text{Hz}$  und  $100\text{Hz}$  abgebildet (Wavelet-Basis: Daubechies-8, Skalentiefe: 10, die X-Achse markiert in unterschiedlichen Intervallen ab Null die oberste Approximationsstufe, sowie anschließend die Skalen in absteigender Reihenfolge). Es ist zu erkennen, dass höhere Frequenzen sich auch auf die Koeffizienten auswirken, es kommt auch hier in den unterschiedlichen Skalen vermehrt zu höheren Schwingungen, und die Anzahl signifikanter Koeffizienten scheint mit höheren Frequenzen zu steigen. Auch können wir uns die Formen anhand des zugrunde liegenden Subband-Codings (vergleiche Kapitel 2) erklären (die tiefsten Frequenzanteile liegen jeweils in den Approximation-Koeffizienten, siehe Verteilung bei  $1\text{Hz}$  in Abbildung 3.2) - ein anschauliches Muster, aus dem sich eine praktische Anwendung in Bezug auf die Modellierung einfacher, stationärer Signale ergibt lässt sich jedoch nicht ableiten.

Anhand dieser Beispiele wird deutlich, warum die Wavelet-Transformation bei der Analyse und Synthese von (quasi)stationären Signalen wenig Bedeutung hat, sondern in ihrer Anwendung erst bei den stochastischen Signalen, die schnellen, plötzlichen Veränderungen im

Zeitbereich unterworfen sind, zur Entfaltung kommt, wie es z.B. bei den Transienten der Fall ist.

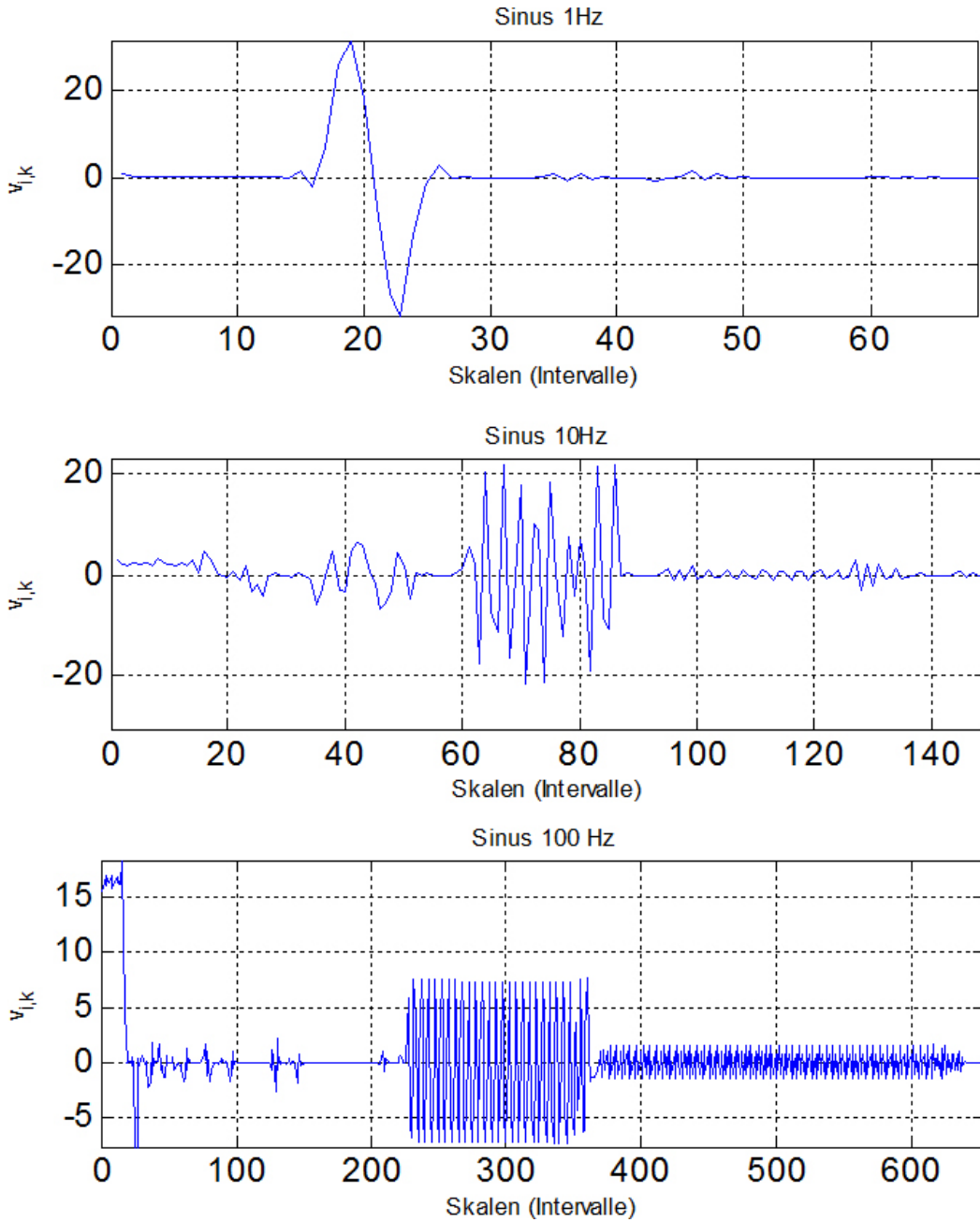


Abbildung 3.2.: Wavelet-Koeffizienten bei Sinus-Signalen verschiedener Frequenzen

## 3.2. Wavelets in der Transienten-Modellierung

Die Modellierung von transienten Signalanteilen, insbesondere bei der Anwendung von Wavelets, ist eng mit ihrer Extraktion vom ursprünglichen Audiosignal verbunden. In der Drei-Komponenten-Zerlegung, in (Dau05) als eine von drei Methoden zur Transienten-Extraktion vorgestellt, lassen sich grob zwei verschiedene Vorgehensweisen unterscheiden, die sequentielle Extraktion der drei Komponenten, sowie die simultane Extraktion.

Die Wavelet-Transformation findet vor allem auch in der Bild-Komprimierung Anwendung. Bilder haben oft eine Struktur, die aus sich nur gemächlich verändernden Regionen besteht, mit plötzlichen, schnellen Veränderungen zwischen diesen Regionen. Die Wavelet-Transformation eignet sich hervorragend zur Darstellung solcher schnellen Wechsel im Signal (also genau der Charakteristik von Transienten), die signifikanten Wavelet-Koeffizienten ballen sich in solchen Fällen auf engem Raum, eine Eigenschaft, die auch in diversen Audio-Codern ausgenutzt wird ((Mal93), (TP00)).

### 3.2.1. Sequentielle Extraktion der Transienten

Das in (DT02) vorgeschlagene Verfahren extrahiert zunächst den tonalen Anteil mittels einer diskreten Cosinus-Transformation (DCT) mit anschließendem Koeffizienten-Thresholding, und bezieht aus dem übrig bleibenden, nontonalen Signalanteil den transienten Part dann mittels einer Wavelet-Transformation. Diese Transformation kann innerhalb der Transform-Coding-Strategien als eine Veränderung der Basis des Signals angesehen werden (im Bezug auf die Wavelet-Transformation also durch eine Transformation über eine bestimmte Wavelet-Basis), mit dem Ziel, im Sinne des zu extrahierenden Signalanteils die überflüssigen Teile auszufaktorisieren, also die Anzahl der signifikanten Koeffizienten zu reduzieren. Die Wahl der Transformation muss dabei dem gewünschten Resultat angepasst werden. In der Literatur finden sich viele Anwendungen solcher angepassten Transformationen, siehe (NSJ84) oder (PV01). Bei der Wahl der Basis kann man sich dabei auf eine Auswahl geeignet für eine bestimmte Klasse von Signalen beschränken (z.B. transiente Signalanteile), oder aber auch die Basis gänzlich dem Ausgangssignal anpassen, hierzu gibt es z.B. eine Abhandlung über die sogenannten Best-Basis-Strategien in (Wic94).

Die Extraktion des transienten Signalanteils aus dem nontonalen Signal in (Dau05) erfolgt vereinfacht beschrieben durch die Wavelet-Transformation des nontonalen Signalanteils mit einer geeigneten Wavelet-Basis (in diesem Fall eine Basis mit guter Zeitauflösung, bzw. Filter mit kurzen Impulsantworten in der zugehörigen Filterbank), und einem anschließendem Thresholding der resultierenden Wavelet-Koeffizienten. Die inverse Wavelet-Transformation dieser signifikanten Wavelet-Koeffizienten wiederum soll den transienten Signalanteil ergeben. Zusätzlich wird vorgeschlagen, die Wavelet-Transformation innerhalb von kurzen Zeitfenstern durchzuführen, und für jedes Zeitfenster die Koeffizienten einen an die Verteilung



signifikanter Koeffizienten innerhalb eines grösseren Fensters angepassten Threshold zu unterziehen.

Da wir mittels dem angewendeten Verfahren aus (FPE<sup>+</sup>06) bereits über den nontonalen Signalanteil  $x_{nonton}(t)$  der Gitarrentöne verfügen, wurde dieser vergleichsweise unkompliziert zu implementierende Ansatz einem Anwendungs-Versuch unterzogen, angewendet auf den Gitarrenklang *hoffmannL hense sonor S1-VI Bb5-1*. Die Funktion *extractfrom\_nonton\_transient.m* führt mit dem eingehenden, nontonalen Signalanteil eine Wavelet-Transformation durch, behält die signifikanten Koeffizienten, und rekonstruiert aus diesen wiederum den vermeintlichen transienten Signalanteil (siehe Blockschaltbild 3.3).

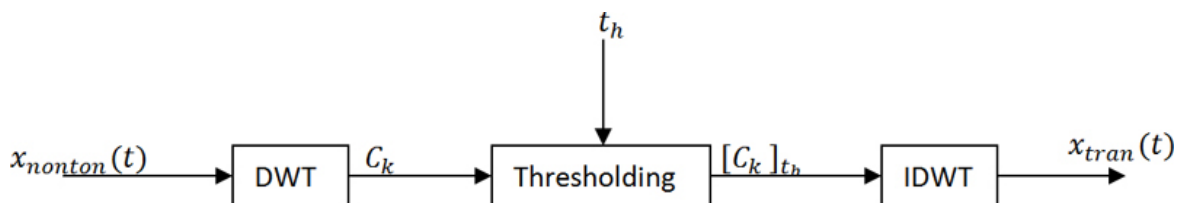


Abbildung 3.3.: Blockschaltbild der Extraktion des transienten Signalanteils

Es wird bei der Transformation auf eine Zeitfensterung und angepasstem Threshold verzichtet. In Bezug auf die Basis wird ein Wavelet mit hoher Frequenz gewählt, da diese die Bedingung der scharfen Zeitauflösung erfüllen (siehe (Dau92)). Es bietet sich ein Daubechies-Wavelet hoher Ordnung an, es wird die Daubechies-10 Wavelet Basis angewendet, mit einer Skalen-Tiefe von  $k = 10$  bei der Transformation. Es wird ein harter Threshold von  $t_h = 0.1$  gewählt, das führt zu einer Anzahl von ca. 100 „signifikanten“ Koeffizienten, angeglichen an die Anzahl verwendeter Koeffizienten in (Dau05). In Abbildung 3.4 sind die drei Signale  $x_{nonton}(t)$ ,  $x_{tran}(t)$  sowie  $x_{res}(t)$  abgebildet.

Deutlich erkennbar ist, dass die „lauten“ Anteile, also die hohen Amplituden, sich in der extrahierten Transienten befinden. Mehr Aufschluss bringt der Hörtest - das resultierende transiente Signal ist ein dumpfes Klopfgeräusch, während in der Residue hörbar das per Definition zu vermutende Rauschen sowie Anteile, die sich sehr nach der höherfrequenten Resonanz des Gitarrenkorpus anhören, zu finden sind. Letzterer resultiert aus dem aus (FPE<sup>+</sup>06) übernommenen Verfahren zur Extraktion des nontonalen Signalanteils - in (vC09) wurde bereits beobachtet, dass sich in der Residue teils ausgeprägte Maxima bei den Resonanzfrequenzen des Gitarrenkorpus befinden, welcher durch die Saiten-Schwingung und dem Saiten-Anschlag angeregt wird.

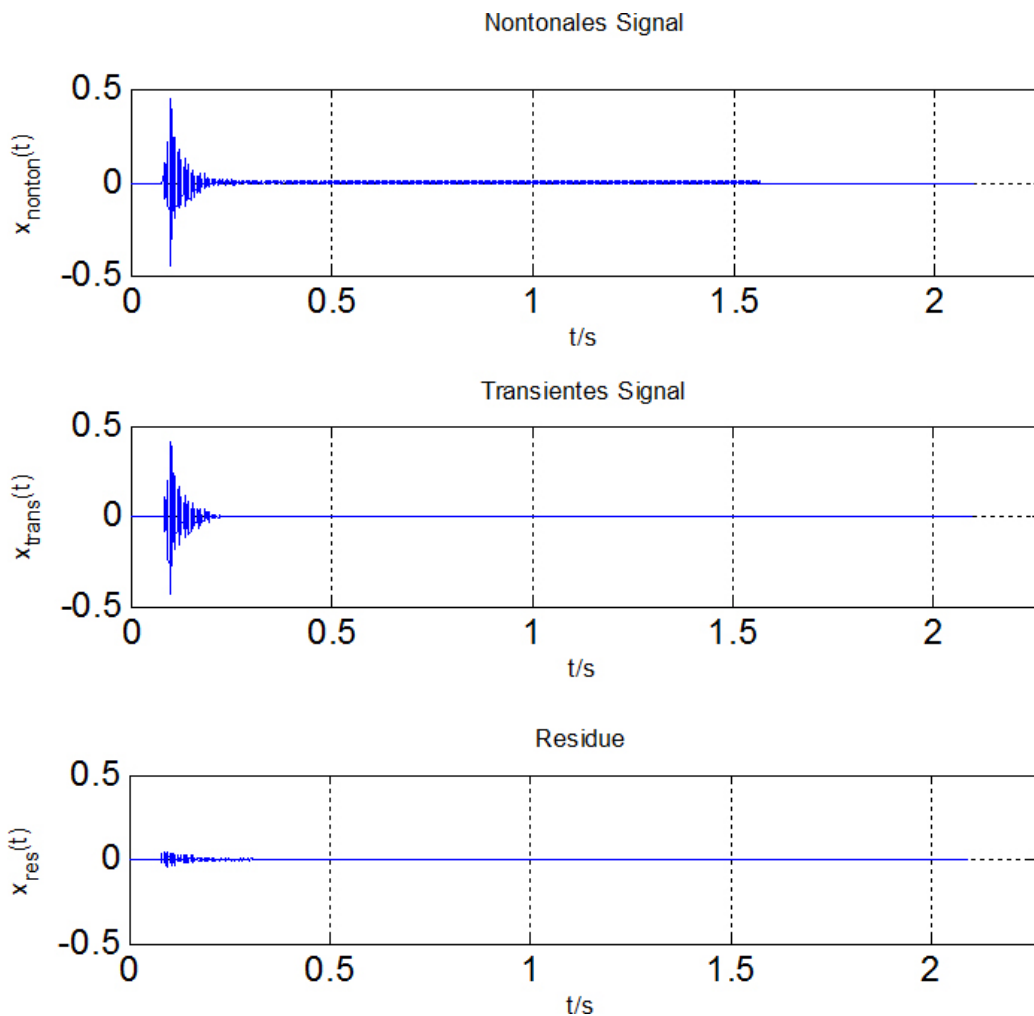


Abbildung 3.4.: Nontonaler Signalanteil, extrahierte Transiente und Residuum

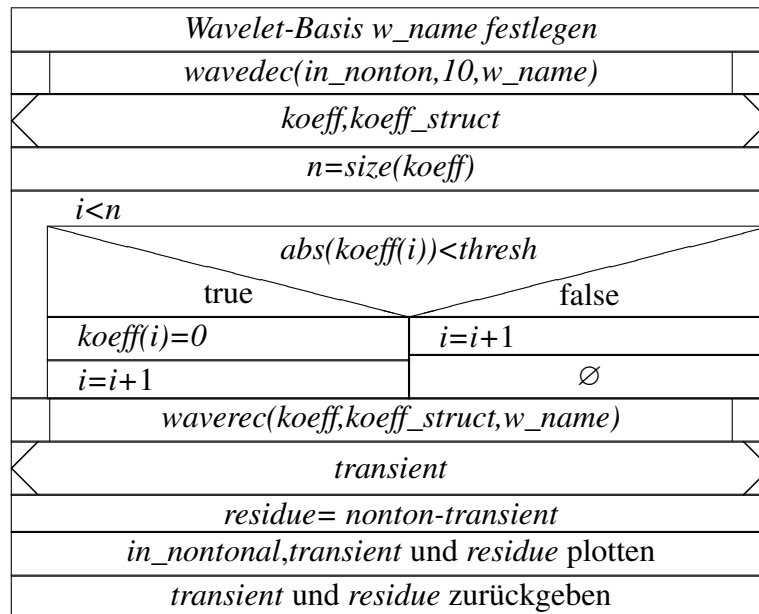
Bei einer Veränderung des Threshold auf 0.01 bleibt auch die hochfrequente Korpusresonanz zu größten Teilen in  $x_{tran}(t)$  enthalten, jedoch sind in diesem Fall mit 603 ungefähr sechs mal so viele Koeffizienten zur Rekonstruktion nötig. Die Möglichkeit, hier noch im nontonalen Signalanteil diese höherfrequente Korpusresonanz herauszufiltern, könnte interessante Möglichkeiten zu weiterführende Untersuchungen bieten. In erster Linie bestätigt wird hier das erwartete Verhalten der Wavelet-Transformation: dieser eher tonale Signalanteil entfällt, je niedriger der Threshold für die signifikanten Wavelet-Koeffizienten gesetzt wird.

### Programmablauf

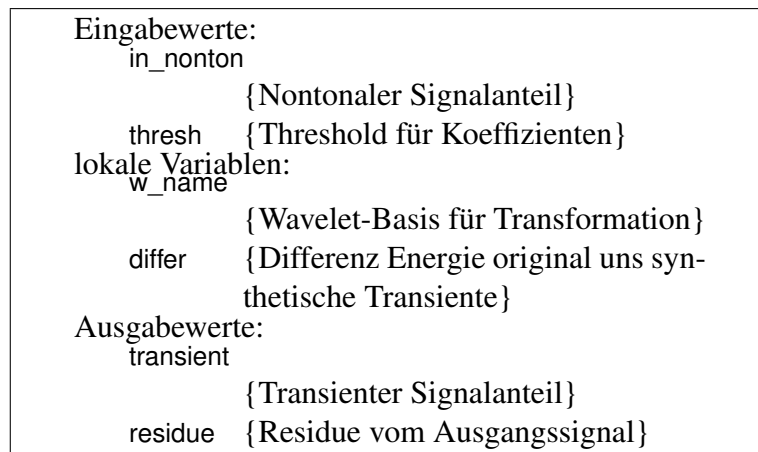
- An die Funktion `extractfromnonton_transient.m` (Der Programmablauf der Funktion ist im Struktogramm 3.1 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm

3.2) werden der nontonale Signalanteil  $in\_nonton$  sowie der gewünschte Threshold  $thresh$  übergeben.

- Die Funktion führt dann mit der Matlab-Funktion *wavedec* eine Wavelet-Transformation des nontonalen Signalanteils mit der lokal zu deklarierenden Wavelet-Basis  $w\_name$  durch, mit der Skalen-Tiefe  $k = 10$ . Anschliessend werden die resultierenden Wavelet-Koeffizienten kleiner dem Threshold  $thresh$  auf Null gesetzt.
- Anschliessend wird aus den Koeffizienten mit der Matlab-Funktion *waverec* das rekonstruierte Signal erstellt und dieses als transienter Signalanteil *transient* deklariert. Dieser wird dann vom nontonalen Signalanteil subtrahiert und das Ergebnis als Residue *residue* deklariert.
- Abschließend werden nontonaler, transienter Signalanteil und die Residue geplottet, und letztere beiden Signale aus der Funktion zurückgegeben.



Struktogramm 3.1: Programmablauf zu *extractfromnonton\_transient.m*

Struktogramm 3.2: Variablen zu *extracfromnonton\_transient.m*

Es scheint also mittels der Wavelet-Transformation schon in einfach zu implementierender Form möglich zu sein, den transienten Signalanteil in einer akzeptablen Qualität aus dem nontonalen Signalanteil zu extrahieren, und somit auch die Residue zu bilden. Mittels feinerer Implementierung mit Fensterung und an Verteilungswahrscheinlichkeiten angepassten Threshold ließe sich die Qualität noch erhöhen, dennoch hat dieses Verfahren die Schwäche, von der Genauigkeit der vorhergehenden Extraktion des nontonalen Anteils durch Eliminierung des tonalem Signalanteils abhängig zu sein. Das sich durch eine einfache Wavelet-Transformation mit folgendem Thresholding der Wavelet-Koeffizienten allein die tonalen und transienten Signalanteile aus dem Ursprungssignal nicht trennen lassen ((*Dau01*)), lässt sich schnell erkennen, wenn man das hier entwickelte Programm anstatt auf den nontonalen Signal-Anteil auf das Ursprungssignal anwendet - hier sind nach der Rekonstruktion immer noch deutliche Anteile der tonalen Komponente vorhanden, siehe auch Abbildung 3.5.

Eine andere Herangehensweise ist die Anwendung einer Transformation, bei der die Auflösung lokal angepasst wird, man spricht von der *adaptive-time-frequency-analysis*. Ein einfaches Beispiel hierfür sind die in Kapitel 2 bereits vorgestellten Wavelet-Pakete, ein Beispiel für einen Algorithmus zur Findung der besten Basis für solche Transformationen mit angepasster Auflösung ist der „best orthogonal basis blgorithmus“ ((*RC92*)).

Ein Beispiel für die Anwendung von Wavelet-Paketen findet sich in (*KH96*). Hier wird ebenfalls zunächst mit einem sinusoidalen Modelling der tonale Signalanteil extrahiert, und der resultierende nontonale Signalanteil einer Wavelet-Transformation unterworfen. Dabei kommt eine Transformation mit Wavelet-Paketen zur Anwendung, da es möglich ist, dessen Frequenzbänder in Bezug auf die kritische Bänder-Struktur des menschlichen Hörsystems anzupassen, wie es auch bei geläufigen Audio-Codern wie MP3 in Bezug auf andere Transformations-Verfahren ausgenutzt wird. Hier wird der nontonale Anteil insbesondere in den höheren Frequenzen in Informationen über „Kanten“ und „Hintergrund-Rauschen“ aufgeteilt. Die Detektion der Kanten wird unter Zuhilfenahme einer „statistical-region-matching“

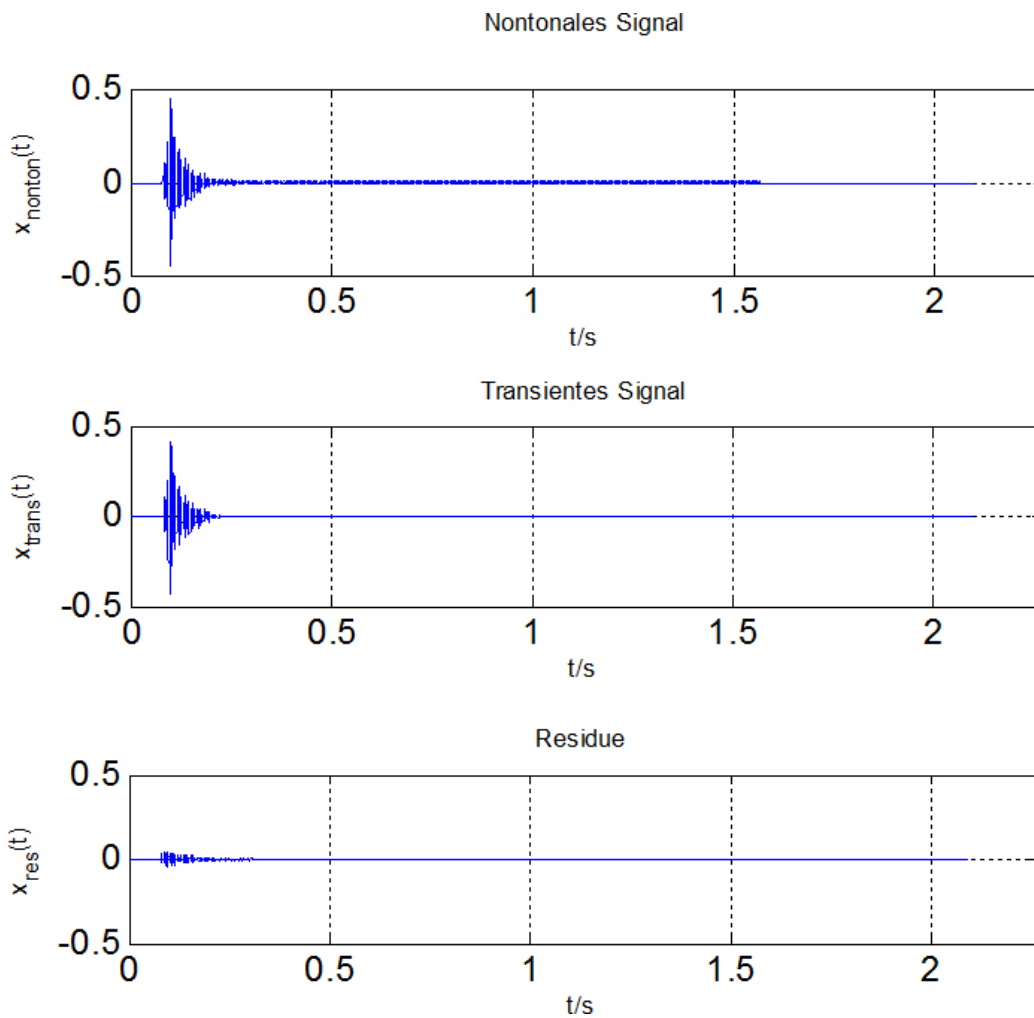


Abbildung 3.5.: Ausgangssignal, extrahierte Transiente und Residue

Methode durchgeführt, wie sie in (Har80) beschrieben ist. Die Detektion an sich erfolgt mit einer dyadischen Wavelet-Zerlegung des nontonalen Signals, jedes Band der dyadischen Wavelet-Transformation (Wavelet-Pakete) wird durch einen „Edge-Detektor“ geleitet. Weiterhin werden „Edges“ beseitigt, die zu nahe aneinander liegen, um vom menschlichen Gehör separat wahrgenommen zu werden. Die resultierenden „Edges“ werden dann als Teil des transienten Signalanteils wahrgenommen und von dem nontonalem Signal subtrahiert, um die Residue (=Rauschen) zu erhalten.

### 3.2.2. Simultane Extraktion der Transienten

Um den transienten Signalanteil direkt aus dem Ausgangssignal gewinnen zu können, werden in (Dau05) ein paar Möglichkeiten zur simultanen Extraktion der Signalkomponenten angesprochen.

Eine Technik zur simultanen Extraktion der Signalkomponenten ist die Anwendung der *Sparse-Overcomplete-Methode*. Das Ziel dieser ist es, ein Signal  $x$  als lineare Kombination aus festgelegten elementaren Wellen, sogenannten „atoms“ darzustellen:

$$x = \sum_k \alpha_k \varphi_k \quad , \quad (3.1)$$

wobei  $\alpha_k$  Skalare und  $\varphi_k$  die Atoms sind, die aus einem Dictionary  $D$  bezogen werden. Man sagt, das Dictionary  $D$  ist *overcomplete*, wenn es den gesamten Raum überspannt und mehr Elemente hat als die Dimension  $N$  des Raumes. Man sucht nun eine möglichst kurze (sparse) Dekomposition, so dass eine kleine Anzahl  $K \ll N$  Atoms ausreicht um eine gute Approximation des Signales zu erreichen:

$$x \approx \sum_{j=1}^K \alpha_{kj} \varphi_{kj} \quad . \quad (3.2)$$

Wenn man nun das Dictionary  $D$  aus Atoms zweier Klassen anlegt, so dass die Atoms  $S$  den tonalen Anteil des Signals repräsentieren, und die Atoms  $T$  den transienten Anteil, und  $D = S \cup T$ , führt eine „sparse-approximation“ des Signals zu einer natürlichen Teilung in tonalen und transienten Anteil. Die hierfür geeigneten Atoms sind schon aus vorhergehenden Betrachtungen bekannt, z.B. modifizierte DCT-Atoms für den tonalen Part, und Wavelet-Atoms für den transienten Part. Beide Anteile können jeweils durch eine Rekonstruktion unter Verwendung nur der Atoms in  $S$  bzw. in  $T$  hergestellt werden. Mit Blick auf Kapitel 2 lassen sich diese Atoms, also elementare Wellen, als die gestreckten, gestauchten Wavelets der kontinuierlichen Wavelet-Transformation verstehen.

Das Verfahren der *General-Overcomplete-Dictionaries* ist ein mathematisch relativ komplexes, vor allem das Finden einer guten *sparse-approximation* ist nicht trivial, (G.94) zeigt, dass es sich um ein NP-Schweres Problem handelt. Zu dieser Thematik, in Bezug auf digitale Signalbearbeitung (und auch konkret zu dem hier behandelten Anwendung der Klang-Modellierung und der damit verbundenen Extraktion der tonalen und transienten Signalanteile) gibt es inzwischen viele Techniken, eine davon die *Matching-Pursuit-Methode* nach (Mal93).

Die *Matching-Pursuit-Methode* lässt sich als Iterationsverfahren verstehen, das zu jeder Wiederholung ein „bestes“ Atom  $\varphi_{k_0}$  auswählt, also jenes, das am stärksten mit dem Signal

$K_0 = \operatorname{argmax}_k |\langle x, \phi_k \rangle|$  korreliert. Das entsprechend gewichtete Atom wird dann vom Signal subtrahiert und der Prozess wird wiederholt, bis ein bestimmtes Anhalt-Kriterium erfüllt ist.

Für die Anwendung bei der Separation von Audiosignalen in tonalen, transienten und residue Anteil wurde dieses Verfahren in (Dau06) zur sogenannten *Molecular Matching Pursuit*-Methode erweitert, welches bei jeder Wiederholung eine ganze Gruppe benachbarter Atoms (genannt *Molecule*) auswählt. Weitere Beispiel für die Anwendung der *Matching-Pursuit-Methode* in Verbund mit den angesprochenen *Dictionaries* für die Extraktion der Signalanteile ist in (DD01) oder in (VC04) zu finden.

Eine sehr aktuelle Arbeit ((BMV10)) extrahiert den transienten Anteil von Audio-Signalen, indem sie als Überlagerung passender Wavelet-Atoms modelliert werden, und nutzt dabei das vorhersehbare und besondere Verhalten der Transienten in der Zeit-Skalen Ebene aus. Ausgenutzt wird hierbei das Vermögen der Wavelet-Transformation, Singularitäten in einem Signal zu Charakterisieren, wie es z.B. der Onset eines Tones, zugleich Beginn einer Transiente, ist. In der Literatur der Wavelet-Theorie ist (MLH92) hierzu eine wegweisende Arbeit. Im speziellen wird ein Modell für die Verteilung der Energie der Singularitäten über die Zeit-Skalen-Ebene entwickelt, im Sinne einer Untersuchung ihrer Beziehungen zu Transienten in Audio-Signalen. Das Verhalten der Energie der Transienten über die Skalen wird für ihre Bestimmung ausgenutzt.

Eine grundlegende Erkenntnis aus (MLH92) über das Verhalten der Wavelet-Atoms bzw. der Koeffizienten Wavelet-Transformierter Signale bezüglich Singularitäten in Bezug auf Transiente lässt sich in Abbildung 3.6 erkennen.

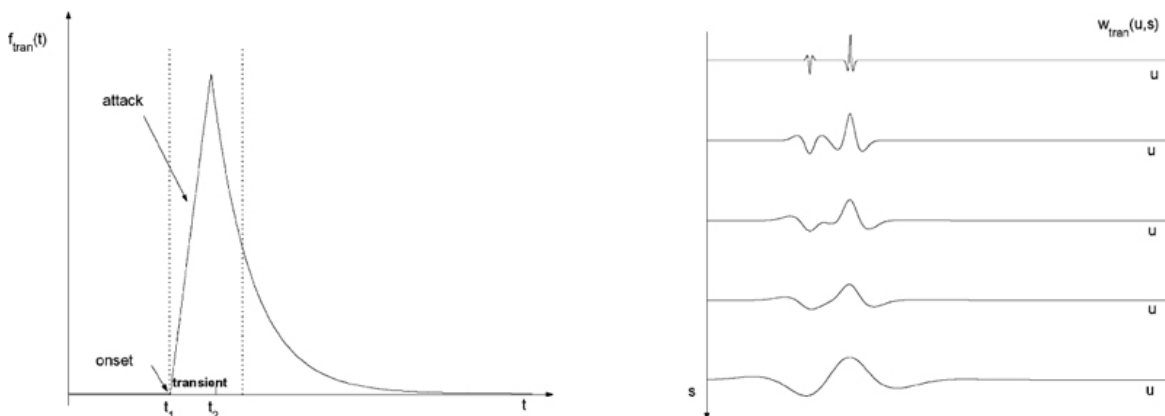


Abbildung 3.6.: Rampen-Signal und Wavelet-Transformation

Die Form der Rampe beinhalten zwei solcher einzigartigen Punkte, den Onset und das

Ende der Attack-Phase, bei denen es im Zeitbereich zu einer abrupten Veränderung im Signalverlauf kommt. Bei einer Wavelet-Transformation mittels einer Wavelet-Basis mit genügende verschwindenden Momenten (siehe dazu Kapitel 2) erzeugen diese Punkte signifikante Beiträge zur Energie der Wavelet-Transformation, während die abklingende Form kaum dazu beiträgt - man erhält also grosse Koeffizienten an dem Onset und dem Ende der Attack-Phase, wie anhand des Verlaufs der Koeffizienten in den verschiedenen Skalen in Abbildung 3.6 rechts zu erkennen ist.

In (BMV10) wird weiterhin ausgenutzt, dass die Mittelpunkte dieser Atoms an den signifikanten Stellen (oder eben die dazugehörigen Koeffizienten), bezogen auf Abbildung 3.7 links die Punkte bei  $t_1$  und  $t_2$ , in der Zeit-Skalen-Ebene  $(u, s)$  kontinuierlichen Kurven folgen, mit abstoßendem Verhalten, abhängig von der Geschwindigkeit des Abfalls bzw. Anstiegs des Signals (siehe Abbildung 3.7).

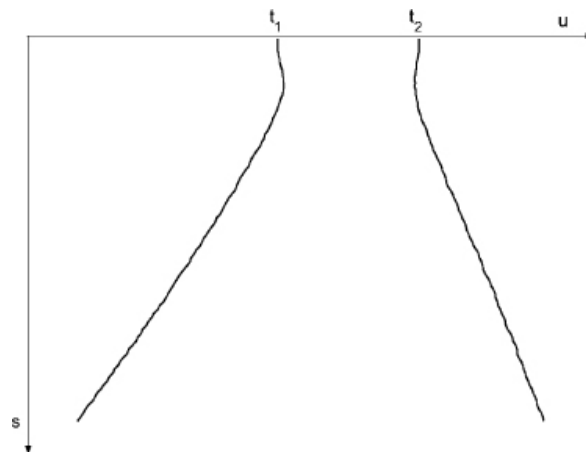


Abbildung 3.7.: Verlauf der Maxima der Wavelet-Koeffizienten bei Rampen-Signal

Dieses Verhalten kann nun mit einem entwickeltem präzisiertem *Evolution-Law* modelliert werden und so die für die Transienten signifikanten Atoms bzw. Koeffizienten ermittelt werden, mit deren Hilfe der transiente Signalanteil dann modelliert werden kann.

Ein ähnlicher Ansatz wird in (Dau01) untersucht. Auch hier wird ausgenutzt, dass sich transiente Signalanteile nicht nur in Form einiger signifikanter Koeffizienten (insbesondere in den untereren Skalen) ausdrücken, sondern diese auch in der Zeit-Skalen-Ebene strukturiert sind - es ist also sehr wahrscheinlich, dass signifikante Koeffizienten von zusätzlichen, ebenfalls signifikanten Koeffizienten in benachbarten Skalen begleitet werden.

Ein Ergebnis dieses Verfahrens ist in Abbildung 3.8 zu finden, angewandt jedoch nur auf ein



Signal, dessen tonaler Anteil bereits extrahiert wurde. Deutlich erkennbar ist die Baumstruktur über die Skalen der zur Rekonstruktion ausgewählten Koeffizienten.

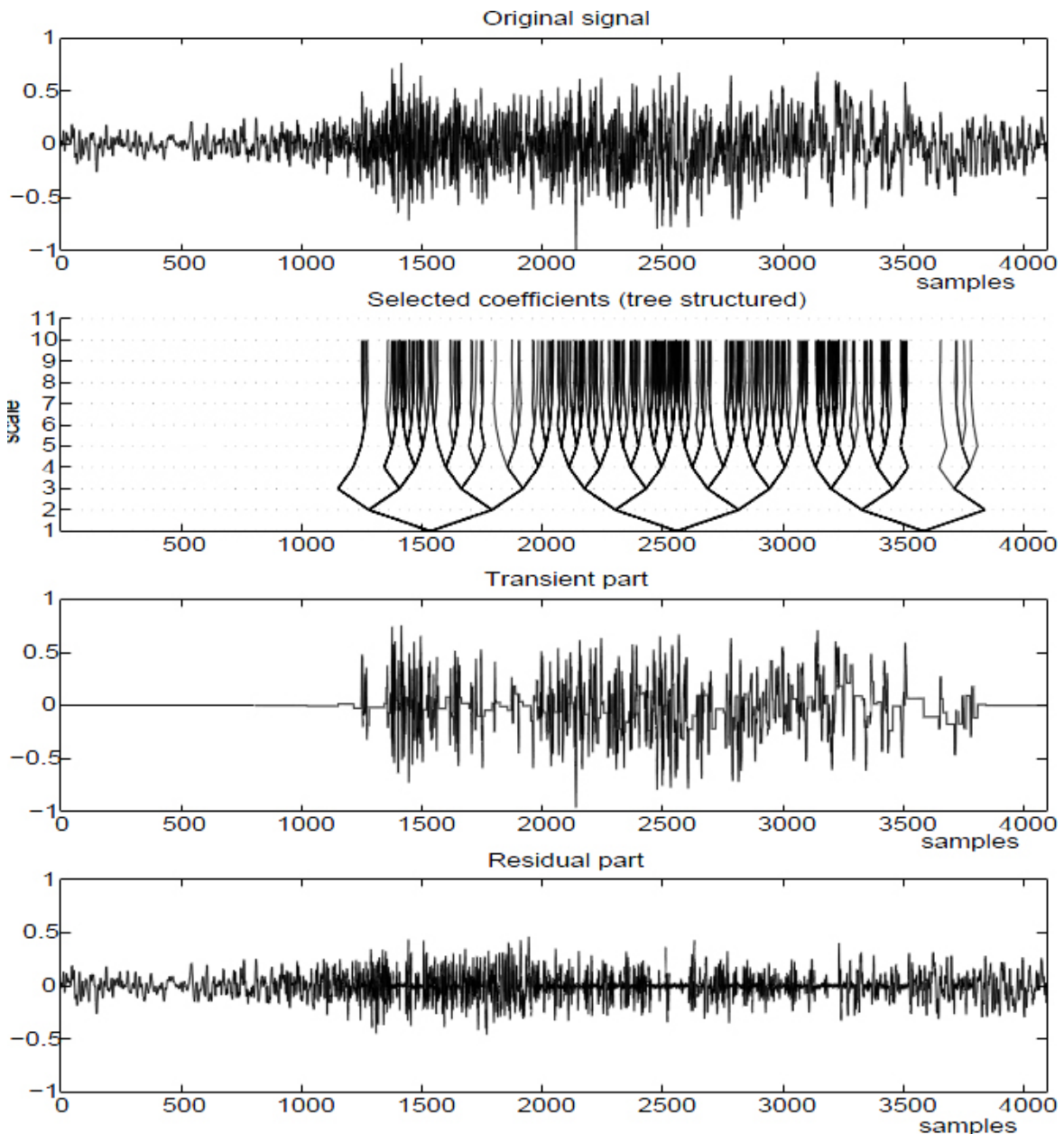


Abbildung 3.8.: Nontonaler Signalanteil, extrahierte Transiente und Residue (aus (Dau01))

Der Grundsatz dieser Vorgehensweisen ist das zusammenhängende Verhalten der Wavelet-Koeffizienten bzw. der Wavelet-Atoms über die Zeit-Skalen-Ebene. Die Schwierigkeit besteht darin, geeignete Kriterien zu finden, welche Koeffizienten als signifikant anzusehen sind, und wie genau diese innerhalb der Zeit-Skalen-Ebene zusammenhängen, um letztendlich für die

Rekonstruktion nur die für den transienten Anteil relevanten Koeffizienten zu verwenden - hierzu sind erweiternde, mathematische Methoden anzuwenden, wie eben z.B. die *Matching Pursuit Methode*.

## 4. Das entwickelte Analyse- und Synthese-Verfahren

Wie einleitend beschrieben, soll das hier entwickelte Analyse- und Synthese-Verfahren auf dem Modell einer Unterteilung eines akustischen Signals in die Komponenten tonaler Signalanteil, transienter Signalanteil und Residue erfolgen:

$$x(t) = x_{ton}(t) + x_{tran}(t) + x_{res}(t). \quad (4.1)$$

Dabei soll der tonale Signalanteil, wie schon bei (vC09) unter Anwendung einer Kurzzeit-Fourier-Transformation, modelliert werden. Der transiente Anteil wird nach Vorbild der unter Kapitel 3.2 vorgestellten Möglichkeiten unter Zuhilfenahme der Wavelet-Transformation modelliert, wobei mit Blick auf die damit verbundenen Modifikations-Eigenschaften weiterhin auch mit einer ADR-Hüllkurve gearbeitet wird. Die dafür notwendige Extraktion des nontonalen Signalanteils aus dem Gesamtsignal erfolgt wie in (vC09) durch eine Eliminierung der tonalen Anteile. Die Residue wird bei der Synthese vernachlässigt, so dass der synthetisierte Gitarrenton aus tonalem und transientem Signalanteil besteht:

$$x_{syn}(t) = x_{ton,syn}(t) + x_{tran,syn}(t). \quad (4.2)$$

### 4.1. Trennung von tonalen und transienten Anteilen des Signals

Für eine Modellierung der Audio-Signale, also in diesem Fall der Gitarrentöne, ist es nötig, diese nach dem angestrebtem Modell in ihre tonalen Anteile, transienten Anteile und die Residue zu zerlegen. Der tonale Anteil wird direkt in dem Verfahren zu seiner Analyse (siehe folgendes Unterkapitel 4.1) von Gesamtsignal extrahiert. Für die Modellierung des transienten Signalanteil es ist nötig, diesen vom nontonalen Signalanteil zu extrahieren.

Zu diesem Zweck wird ein Verfahren aus (FPE<sup>+</sup>06) angewendet, in dem aus dem Gesamtsignal der tonale Anteil entfernt wird, was zu einem nontonalen Signalanteil führt, bestehend aus transientem Anteil und der Residue:

$$x(t) - x_{ton}(t) = x_{nonton}(t) = x_{tran}(t) + x_{res}(t) \quad (4.3)$$

Der nontonale Signalanteil enthält dabei alle Signalanteile, die nicht zu den harmonischen Partialtönen des Klangs gehören. Seine Trennung vom tonalen Signalanteil erfolgt im Frequenzbereich. Das Signal  $x(t)$  wird zunächst mittels der Fourier-Transformation in den Frequenzbereich überführt, es besteht dann aus

$$X(f) = X_{ton}(f) + X_{nonton}(f). \quad (4.4)$$

Um aus diesem Spektrum nun  $X_{nonton}(f)$  extrahieren zu können, wird das Phasenspektrum  $\varphi(X(f))$  sowie das Amplitudenspektrum  $|X(f)|$  errechnet und die Grundfrequenz  $f_0$  des Klangs geschätzt. Im Anschluss wird dann das Amplitudenspektrum nach Peaks im Bereich einer gewissen Toleranz der zu erwartenden Partialtöne der Grundfrequenz gesucht. Die Toleranz  $\delta_{max}$  wird bestimmt als (übernommen aus (vC09))

$$\delta_{max} = f_0 \cdot 0,2 \cdot \frac{L_f}{f_s}, \quad (4.5)$$

wobei  $f_s$  die Samplingfrequenz und  $L_f$  die Länge des Analyse-Fensters ist. Die gefundenen Peaks innerhalb dieser Toleranzbereiche werden dann eliminiert, durch lineare Interpolation werden die entstehenden Lücken geschlossen. Das so entstehende, nontonale Amplitudenspektrum  $X_{nonton}(f)$  wird dann wieder in den Zeitbereich übertragen:

$$x_{nonton} = IFFT(|X_{nonton}| \cdot \varphi(X(f))) \quad (4.6)$$

In Abbildung 4.1 sind das Gesamtsignal und der so extrahierte nontonale Anteil des Gitarrenklanges *hoffmannL\_hense\_sonor\_S1-VI\_Bb5-1* dargestellt. Deutlich erkennbar ist die Begrenzung der Amplitude des nontonalen Anteils, was ungefähr der Charakteristik eines Anschlaggeräusches entspricht.

Um nun weiterführend den transienten Signalanteil  $x_{tran}(x)$  von diesem so ermittelten nontonalen Signalanteil zu extrahieren, also von der Residue  $x_{res}$  zu trennen, wird in (vC09) mittels einer Hüllkurve eine Trennung im Zeitbereich vorgenommen, also der Signalanteil ab einem Zeitpunkt *attS* bis zu einem Zeitpunkt *attE* von  $x_{nonton}(x)$  als transienter Anteil definiert. Diese rein im Zeitbereich stattfindende Trennung hat zur Folge, dass die Anteile der Residue  $x_{res}$  innerhalb dieses Zeitbereichs ebenfalls zum transienten Anteil hinzu definiert werden, das der Residue innewohnende Rauschen wird aber an dieser Stelle vom Anschlaggeräusch mehr oder minder geschluckt und erscheint nicht hörbar.

Wie unter Kapitel 3.2 gezeigt wurde, kann durch die Wavelet-Transformation und ein geeignetes Thresholding der transiente Anteil etwas sauberer von der Residue getrennt werden,

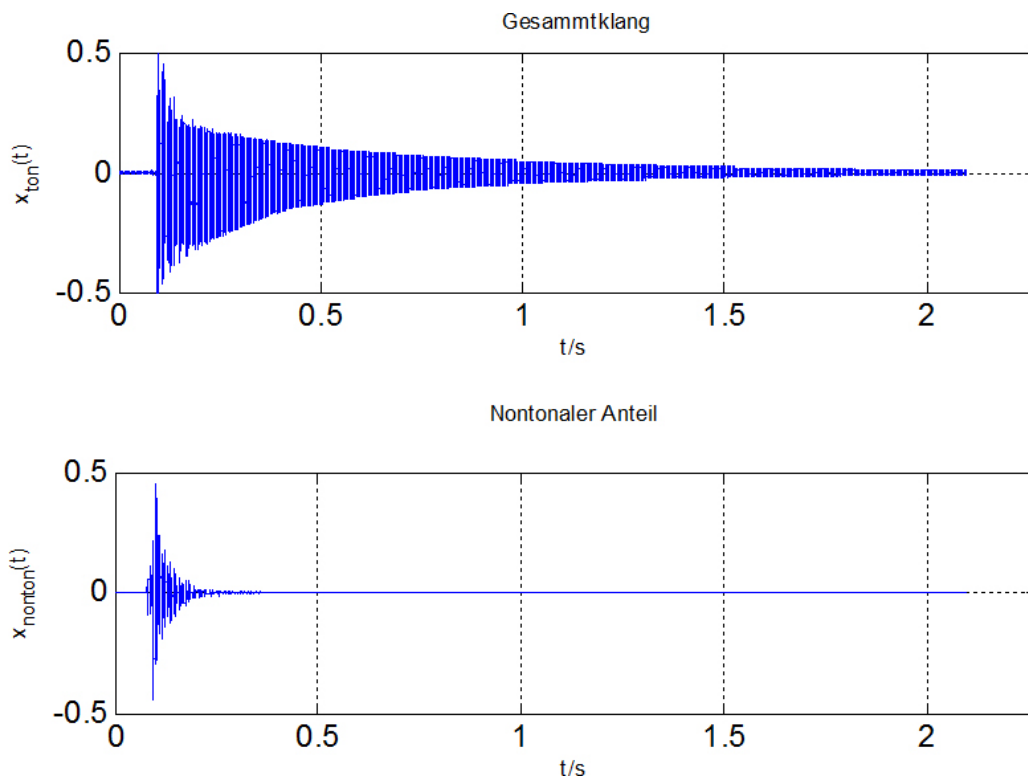


Abbildung 4.1.: Gesamtsignal und extrahierter nontotaler Anteil

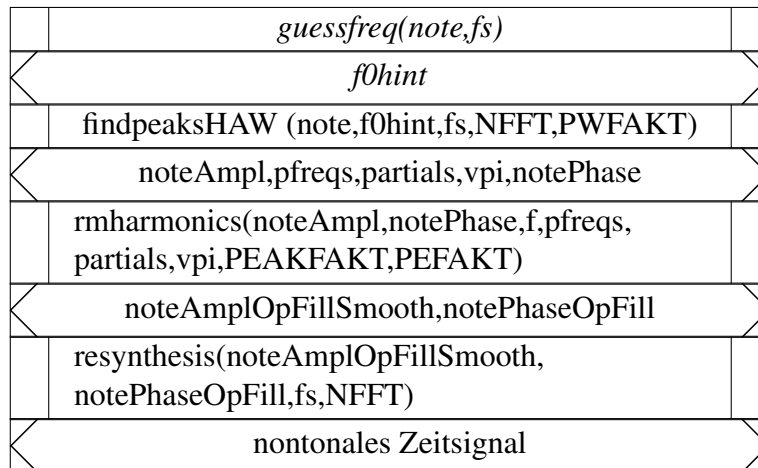
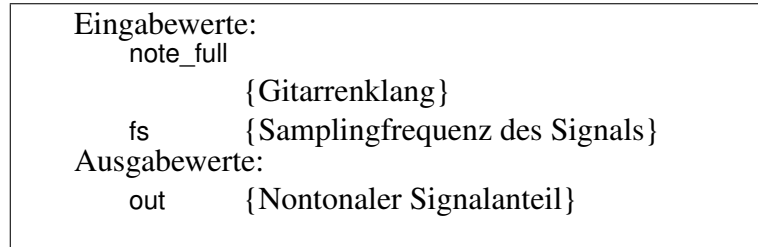
auf die Beschaffenheit der resultierenden Transienten und Residue wurde an der Stelle bereits eingegangen. Im folgenden Verfahren zur Modellierung der Transienten (Kapitel 4.1) kommt eine Kombination beider Ansätze zum Einsatz.

### Programmablauf der Extraktion des nontotalen Anteils

Der nontonale Anteil wird mit der Matlab-Funktion *process.m* extrahiert. Der Programmablauf ist im Struktogramm 4.1 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.2. Diese Funktion sowie weitere darin aufgerufene stammen aus vorherigen Projekten an der HAW Hamburg und sind auf der Daten-CD in einem entsprechend separaten Order *HAW* zu finden und fanden überwiegend in (Voi07) und (Dos08) Verwendung.

- Zunächst wird die Grundfrequenz  $f_0$  mittels der Matlab-Funktion *guessfreq.m* geschätzt. Dies geschieht anhand einer Autokorrelation, die Beschreibung der Funktion befindet sich in (Voi07).
- Die Funktion *findpeaksHAW.m* ermittelt die Peaks der Partialtöne innerhalb des Amplitudenspektrums  $|X(f)|$  des Gesamtsignals (Dos08).

- Mit der Funktion *rmharmonics.m* ((Dos08)) werden in dem Spektrum die Peaks der Partialtöne entfernt und die entstehenden Lücken mit einer linearen Interpolation geschlossen. Es resultiert das nontonale Amplitudenspektrum  $|X_{nonton}(f)|$ .
- Mit der Funktion *resynthesis* wird dann schließlich aus  $|X_{nonton}(f)|$  mittels der inversen Fouriertransformation das nontonale Zeitsignal  $x_{nonton}(t)$  erstellt.

Struktogramm 4.1: Programmablauf zu *process.m*Struktogramm 4.2: Variablen zu *process.m*

## 4.2. Analyse und Synthese des tonalen Anteils

Da die Anwendung der Wavelet-Transformation in Bezug auf die Modellierung tonaler Signalanteile sich als nicht praktikabel herausgestellt hat (siehe Kapitel 3.1) sondern die Verwendung einer auf sinusoidalen Basisfunktionen basierende Transformation hier angebracht ist wird auch in diesem Verfahren dazu eine Modellierung mittels einer Kurzzeit-Fourier-Transformation (STFT) implementiert. Zu diesem Zweck wird das Verfahren aus (vC09) übernommen.

### 4.2.1. Analyse des tonalen Anteils

Um in einem sinusoidalem Verfahren auch den zeitlichen Verlauf der Partialtöne zu berücksichtigen sollen die lokalen Maxima im Amplitudenspektrum in der Zeit-Frequenz-Ebene (TF-Ebene) verfolgt werden. Zu diesem Zweck wird aus dem Ausgangssignal  $x(t)$  die Grundfrequenz  $f_0$  geschätzt (wie schon bei dem Vorgehen zur Extraktion des nontonalen Signalanteils), anschließend  $x(t)$  mittels einer STFT in die TF-Ebene transformiert. Hierbei entsteht eine Matrix  $X(n, m)$ , mit  $n$  als Zeitindex und  $m$  als Frequenzindex. Anschliessend werden in jeder Zeile  $X(n)$  der Matrix die Maxima im Bereich eines Frequenz-Bins der Partialtöne zur ermittelten Grundfrequenz  $f_0$  gesucht und als Amplitude für den zugehörigen Partialton innerhalb der Zeile gespeichert. Das Ergebnis ist eine reelwertige Matrix  $TCH(n, k)$  mit  $n$  als Zeitindex und  $k$  als Index des Partialtons. (Blockschaltbild siehe Abbildung 4.2).

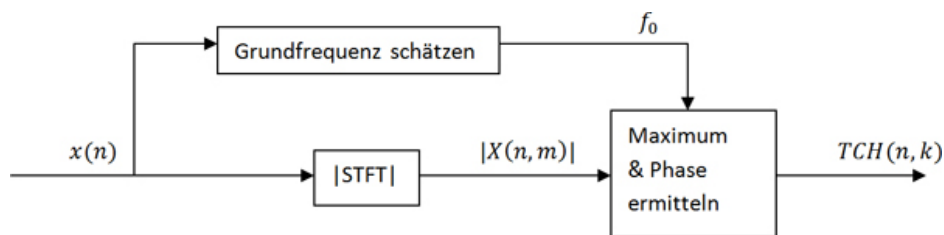


Abbildung 4.2.: Blockschaltbild Analyse tonaler Signalanteil

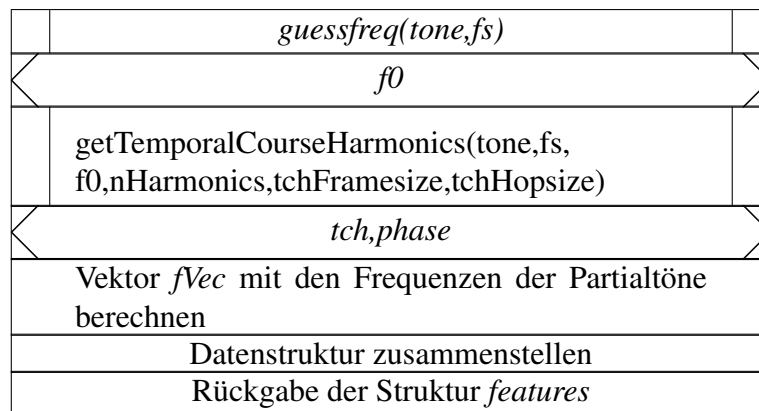
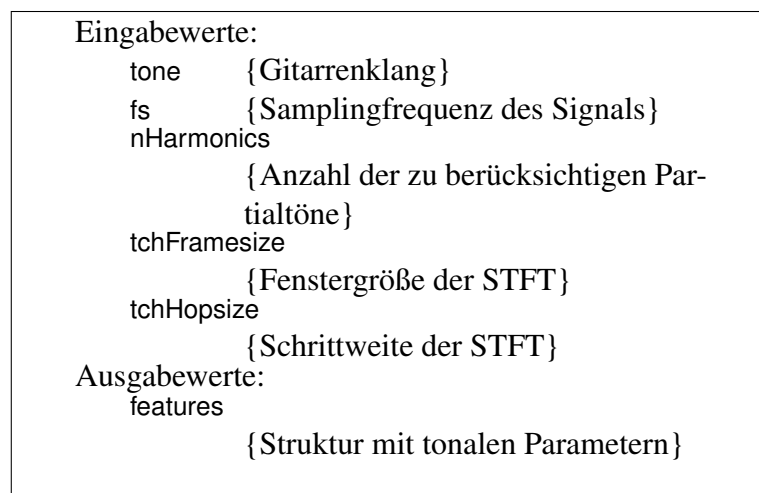
Entscheidende Parameter sind dabei die Framesize  $L_f$ , also die Größe des Zeit-Fensters (in diesem Verfahren wird ein Blackman-Fenster eingesetzt), und die Hopsizes  $L_h$ , also die Größe der Intervallschritte, in denen das Zeit-Fenster verschoben wird, der STFT, die sich sowohl auf Zeit- als auf Frequenzauflösung auswirken. Bezogen auf das Verfahren bedeutet die Hopsizes, dass für jeden Partialton Stützpunkte mit einem Abstand von  $L_h$  erzeugt werden.

#### Programmablauf der Analyse des tonalen Signalanteils

Der tonale Signalanteil wird mit der Matlab-Funktion `extract_features_tonal.m` analysiert. Der Programmablauf ist im Struktogramm 4.3 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.4. In Struktogramm 4.5 ist die Parameter-Struktur `features` beschrieben. Die Funktion stammt aus (vC09), weitere Unterfunktionen aus früheren Projekten an der HAW-Hamburg.

- Die Grundfrequenz  $f_0$  wird mit der Matlab-Funktion `guessfreq.m` (Dos08) ermittelt.

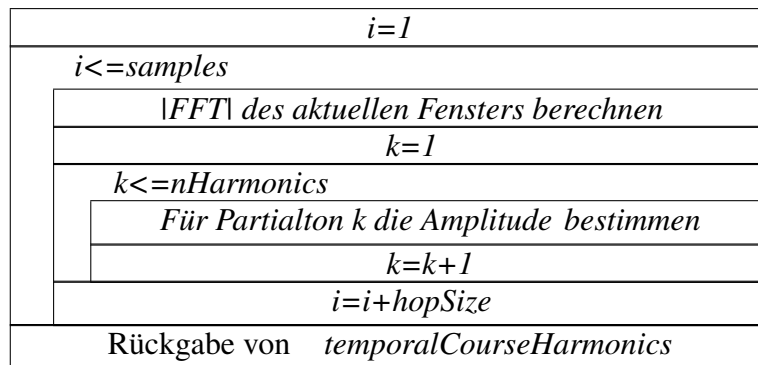
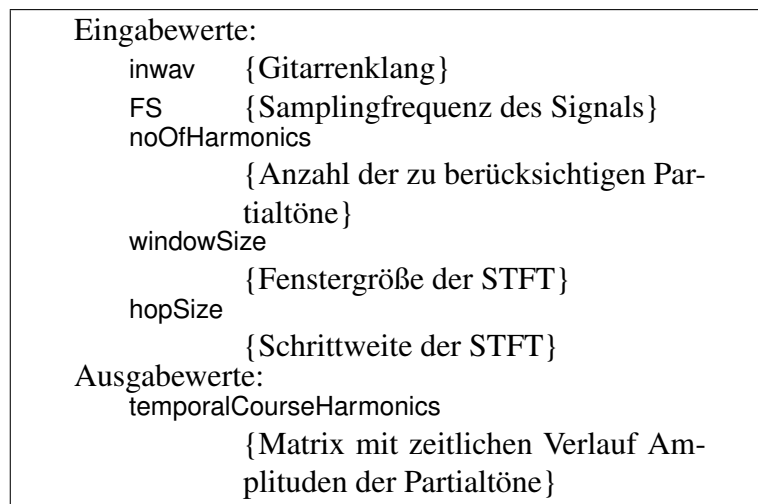
- Zur Vermeidung fehlerhafter Grundfrequenzen wird das ermittelte  $f_0$  der dichtesten aller möglichen Frequenzen des Notensystems *notes* angepasst (dies setzt eine gleichschwebend temperierte Gitarre voraus).
- In der Funktion *getTemporalCourseHarmonics.m* (aus (Dos08), angepasst in (vC09)) werden mittels einer STFT die zeitlichen Verläufe der Partialtöne (Amplitude) ermittelt. Der Programmablauf ist im Struktogramm 4.6 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.7.

Struktogramm 4.3: Programmablauf zu *extract\_features\_tonal.m*Struktogramm 4.4: Variablen zu *extract\_features\_tonal.m*



Elemente von <i>features</i>	
samplerate	{ Samplefrequenz }
tchFramesize	{ Fenstergröße der STFT }
tchHopsize	{ Schrittweite der STFT }
freqcomplete	{ Grundfrequenz des Klangs }
expex	{ Frequenzen der Partialtöne }
amp	{ Matrix mit Amplituden der Partialtöne }

Struktogramm 4.5: Die Struktur *features*

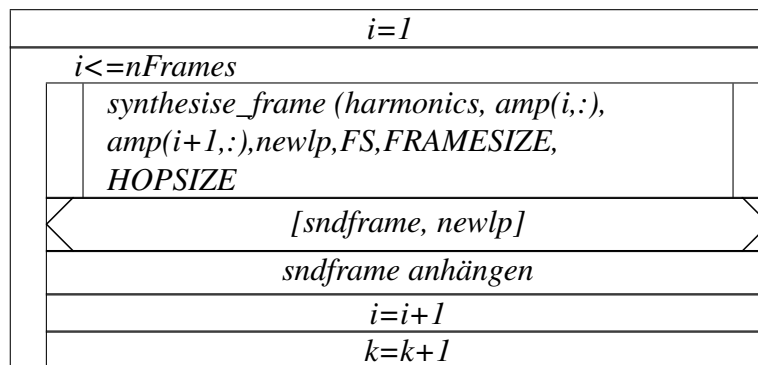
Struktogramm 4.6: Programmablauf zu *getTemporalCourseHarmonics.m*Struktogramm 4.7: Variablen zu *getTemporalCourseHarmonics.m*

### 4.2.2. Synthese des tonalen Anteils

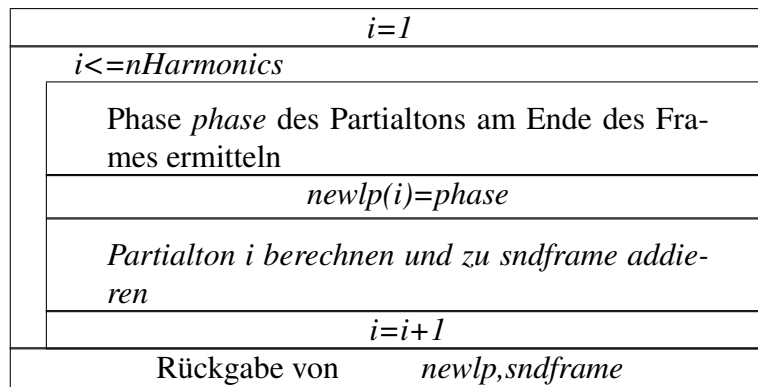
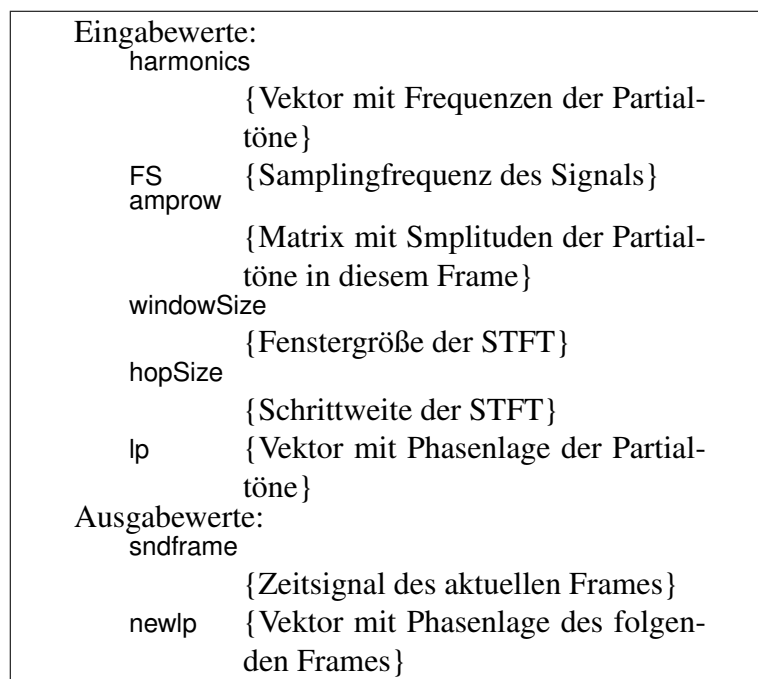
Auch die Synthese erfolgt nach der gleichen Methode wie in (vC09). Es ist möglich, die in der Matrix  $TCH(n, k)$  gespeicherten Partialtöne sowohl in Frequenzbereich (IFFT-Synthese) als auch im Zeitbereich zu resynthetisieren, hier wird letztere Methode angewendet. Zu diesem Zweck kommen  $N_{part}$  Sinusgeneratoren mit den Frequenzen der jeweiligen Partialtöne zum Einsatz. Die Amplituden werden durch eine Hüllkurve bestimmt, für die jeder Frame einen Wert darstellt, die dann linear interpoliert werden.

#### Programmablauf der Synthese des tonalen Signalanteils

Der tonale Signalanteil wird mit der Matlab-Funktion *synthesizer.m* synthetisiert. Der Programmablauf ist im Struktogramm 4.8 dargestellt. In der Funktion wird für jede Zeile der Matrix  $TCH(n, k)$  in einer Schleife die Funktion *synthesise\_frame.m* (Programmablauf in Struktogramm 4.9, Ein- und Ausgabewerte in Struktogramm 4.10) aufgerufen und die Zeitsignale erzeugt, welche dann zum synthetischen, tonalem Signal  $y_{ton}$  zusammengefügt werden.



Struktogramm 4.8: Programmablauf zu *synthesizer.m*

Struktogramm 4.9: Programmablauf zu *synthesize\_frame.m*Struktogramm 4.10: Variablen zu *synthesize\_frame.m*

### 4.2.3. Beurteilung der Analyse und Synthese des tonalen Signalanteils

Eine Beurteilung dieses einfachen Verfahrens des Sinusoidal-Modellings wurde bereits in (vC09) vorgenommen. Das Fazit ist, dass der Klang der Gitarre so nicht naturgetreu synthetisiert werden kann, es entsteht ein in gewisser Art „künstlicher“ und hohler Klang. Ne-

ben dem Fehlen von Anschlaggeräuschen, die in der folgenden Synthese des transienten Signalanteils beigefügt werden sollen, wird dafür eine Schwäche dieser tonalen Synthese genannt. In dem Verfahren werden die Phasenverschiebungen der Partialtöne vernachlässigt und der tonale Signalanteil mit einfachen, linearen Phasengängen synthetisiert. Nach (And04) trägt diese jedoch zur Klangfarbe bei, in (Tac76) konnte nachgewiesen werden, dass für eine Frequenz unter 1kHz der Partialtöne sich die Phasenlagen auf die Klangfarbe auswirkt. In Abbildung 4.3 sind die Phasenverläufe der ersten fünf Partialtöne einer Gitarrensaiten (Hense-sonor-S1-Bb5) zu sehen - klar ist zu erkennen, dass sie sich untereinander mit der Zeit ändern. Bei einer Resynthese der Gitarrentöne wäre es anhand einer Berücksichtigung der Phasenverschiebung der Partialtöne möglich, diese zu verbessern, indem man nach Analyse des Ausgangssignals bei der STFT innerhalb eines FFT-Bins nur diejenigen Frequenzen für die Synthese auswählt, bei der die Phasendifferenz zwischen Anfang und Ende des gewählten FFT-Fensters stimmt, wenn bei der FFT die Fenstergröße kein ganzzahliges Vielfaches der Periodendauer ist. Auf diesem Wege ließe sich also die Genauigkeit der Frequenzen der Partialtöne weiter verbessern.

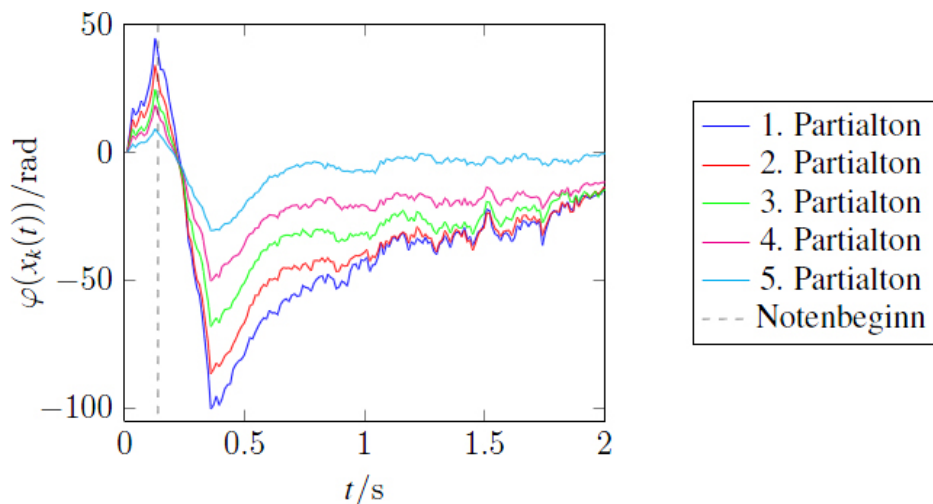


Abbildung 4.3.: Phasenverläufe der Partialtöne (aus (vC09))

Direkte Auswirkung auf die Klang-Qualität der Synthese hat die Anzahl der verwendeten Partialtöne, in (vC09) konnte jedoch ab einer Anzahl von 20 Partialtönen kein Qualitätsunterschied mehr wahrgenommen werden. Diese wandern ab bestimmten hohen Frequenzen auch aus dem Bereich des hörbaren Spektrums (ca.  $22050\text{Hz}$ ) und können sich entsprechend nicht mehr auswirken.

Auch wirken sich die Parameter Framesize und Hopsizel auf die Qualität aus. Ein großes Analyse-Fenster gewährleistet eine hohe Frequenzauflösung, verringert jedoch die Zeitaufklärung. Hier muss ein Kompromiss geschlossen werden, eine Verfeinerung ist möglich durch die Anwendung einer Hopsizel kleiner der Framesize (hier verwendet: Framesize  $L_f = 2048$  Samples und Hopsizel  $L_h = 512$  Samples), so dass die Analyse-

Fenster der STFT sich überlappen. Das Problem der STFT bleibt jedoch ihre Unflexibilität - das Analyse-Fenster und somit die Zeit- und Frequenzauflösung bleiben über die gesamte Analyse konstant. Abhilfe kann hier mit der diskreten Cosinus-Transformation (DCT) geschaffen werden, die, wie in Kapitel 3.1 angesprochen, daher auch in vielen Arbeiten zur tonalen Synthese Anwendung findet.

### 4.3. Analyse und Synthese des transienten Anteils

Der transiente Teil des Gitarrenklangs wird mittels einer Wavelet-Transformation modelliert. Gewählt wurde dieses Verfahren mit der Motivation, aufgrund der Besonderheiten der Wavelet-Transformation in Bezug auf Vorteile bei nicht-stationären Signalen mit schnellen Veränderungen im Zeitbereich (der transiente Part eines Signals ist ein solches) gegenüber einer sinusoiden Modellierung, worauf bereits in den Kapiteln 2 und 3.2 eingegangen wurde. In der Literatur finden wir Ausarbeitungen zu möglichen Ansätzen, insbesondere z.B. bei (VC04) und (VC07). Dort wird ein Modell für eine Modellierung von Transienten mittels Matching Pursuits mit einem Wavelet Dictionary vorgestellt, zur Anwendung kommen hier Wavelet-Pakete. Auch liefern weitere der unter 3.2 vorgestellten Verfahren zur Transienten-Extraktion automatisch eine Möglichkeit zur Modellierung eben dieser mit.

Das hier angewandte Verfahren hat die in (vC09) entwickelte Variante zur Vorlage, kombiniert aber anstelle einer sinusoiden Modellierung eine Modellierung durch eine Wavelet-Transformation nach Vorbild der unter 3.2 durchgeführten Implementierung mit parametrischen Hüllkurven für jeden einzelnen Gitarrenklang. Die Wavelet-Transformation birgt in diesem Zusammenhang mehrere Vorteile. Zunächst kann mit ihr das nicht-stationäre, transiente Signal bei gleicher bis besserer Komprimierungsrate, bei dabei gleich bleibender bis besserer Qualität, komprimiert werden. Die der Transformation in Wavelet- und Aproximations-Koeffizienten inne wohnende Darstellung in der Zeit-Skalen-Ebene (siehe dazu Kapitel 2) sorgt zudem für eine Frequenz- sowie auch Zeitauflösung, die der im sinusoiden Modellierung angewandten Kurzzeit-Fourier-Transformation überlegen ist. Auch entfällt bei diesem Verfahren die der Transformation vorausgehende Aufteilung des transienten Signals in mehrere Frequenzbänder, was zugleich zu einer Reduzierung der benötigten Hüllkurven führt - hier ist im Gegensatz zum vorgestellten Verfahren in (vC09) nur noch eine Hüllkurve pro Gitarrenton nötig.

Die Hüllkurven für das Verfahren werden mit kleinen Modifikation für höhere Genauigkeit und Verbesserung in der Zeitauflösung nach dem Schema aus (vC09) erstellt. Für die Wavelet-Transformation werden zwei unterschiedliche Verfahren getestet - die gewöhnliche schnelle Wavelet-Transformation sowie Wavelet-Pakete - und die Ergebnisse verglichen sowie unter 5 in ihrer Qualität evaluiert. Als Wavelet-Basen werden einfache und weithin bekannte ausgewählt, ihr Ergebnis durch Anhören der Ergebnisse und Betrachtung der Signalformen sowie Signalspektren verglichen und für dieses Verfahren eine uniforme Anwendung

des Daubechies-Wavelets der Ordnung 8 (siehe Abbildung 2.1) für sowohl die Wavelet-Transformation als auch die Wavelet-Pakete ausgewählt.

### 4.3.1. Analyse des transienten Signalanteils mit schneller Wavelet-Transformation

Ziel ist es, aus dem unter 4.1 extrahierten, nontonalen Signalanteils des Gitarrentones mit Hilfe einer schnellen Wavelet-Transformation und einer parametrischen Hüllkurve den transienten Anteil zu extrahieren und zu modellieren, nach dem Schema des Blockschaltbildes in Abbildung 4.4.

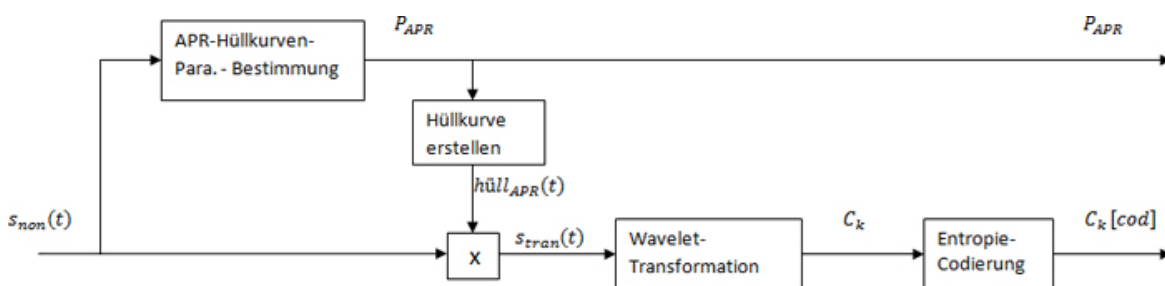


Abbildung 4.4.: Blockschaltbild Transienten-Modellierung

Zunächst wird eine aus der klassischen AHDSR-Hüllkurve abgeleitete APR-Hüllkurve des nontonalen Signals ermittelt. Diese besteht aus fünf Segmenten - einem Einschwingen, einer Peak-Dauer, und drei Abkangsegmenten (nach Abbildung 5.8 aus (vC09)). Hierzu müssen zunächst die Hüllkurven der einzelnen Gitarrentöne erstellt werden. Das unter (vC09) entwickelte Verfahren eines einfachen Glättungstiefpasses weist mehrere Schwächen auf, die Amplitudenwerte sind zu gering, die Hüllkurven relativ ungenau und es kommt aufgrund der Filter-Laufzeit zu einem Laufzeitfehler, der im Endergebnis der Resynthese ein verzögertes Einsetzen des Anschlaggeräusches zur Folge hat.

Zum Ziele einer präziseren Detektion wird die Hüllkurve in diesem Verfahren mittels der Hilbert-Transformation erstellt, was zunächst ein sehr genaues Ergebnis zur Folge hat. Aufgrund der hohen Welligkeit dieser Hüllkurve wird sie anschließend mit einem Tiefpass geglättet, um eine korrekte Berechnung der Hüllkurven-Parameter zu garantieren. In Abbildung 4.5 ist ein Blockschaltbild des Vorgangs zu sehen.

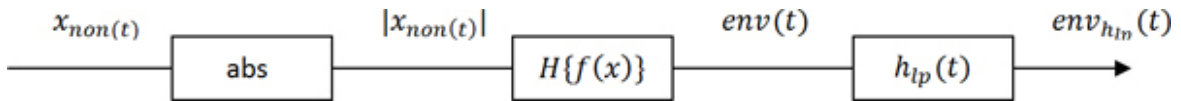


Abbildung 4.5.: Blockschaltbild Hüllkurven-Ermittlung

Zum Vergleich sind in Abbildung 4.6 die mittels Hilbert-Transformation ermittelte Hüllkurve, die gleiche nach dem per Tiefpass geglättete und die mit dem Verfahren aus (vC09) ermittelte Hüllkurve abgebildet, jeweils in einem Plot mit dem nontonalen Ursprungssignal.

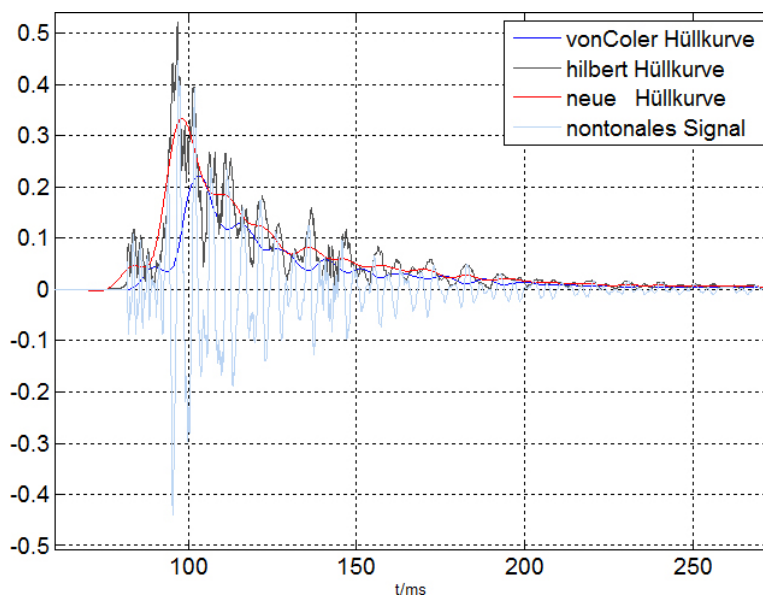


Abbildung 4.6.: Vergleich der ermittelten Hüllkurven

Deutlich zu sehen ist eine ca.  $12\text{ms}$  lange Verzögerung der alten Hüllkurve im Anstiegsverlauf des Anschlaggeräusch, die neu ermittelte Hüllkurve kann diesem wesentlich präziser folgen. Hierbei könnte es sich um eine der Ursachen der leichten Verzögerung bei Einsetzen des Anschlaggeräusches gegenüber den tonalen Signalanteils bei (vC09) handeln, die damit behoben wäre, indem für den Glättungstiefpass die Matlab-Funktion `filtfilt.m` verwendet wird, welche das Signal einmal von vorne und einmal von hinten gesehen filtert und die Resultate mittelt.

Im nächsten Schritt werden anhand der ermittelten Hüllkurve die Hüllkurven-Parameter nach dem Schema der APR-Hüllkurve bestimmt. In der vonColer Version findet sich dabei eine Schwäche bei der Bestimmung des Anfangspunktes der Attack-Phase (*attS*). Dieser wird, wie dort nachzulesen ist, an dem Schnittpunkt der Hüllkurve und einer Graden durch die



Punkte *attE* (Ende der Attack-Phase) *attS* (der Punkt auf Der Hüllkurve auf halben Weg zwischen *attE* und der Abszisse, linksseitig von *attE*) gesetzt. Dieses Vorgehen setzt aber nun voraus, dass der Anstieg der Hüllkurve von ihrem Startpunkt aus zum Maximum, was ca. als Attack-Phase der APR-Hüllkurve interpretiert wird, mit einer nahezu linearen Steigung verläuft. Wie man an der Hüllkurve des nontonalen Teils des Gitarrentons *hoffmannL hense sonor S6-VI Bb3-4* jedoch sehen kann, ist das so nicht immer gegeben. In Abbildung 4.7 ist dies deutlich zu sehen, ebenso, wo der Startpunkt *attS* nach dem alten Verfahren fälschlicher Weise nun bestimmt werden würde. Es ist deutlich ersichtlich, dass eine Folge auch hier ein späteres Einsetzen des Anschlaggeräusches sein kann.

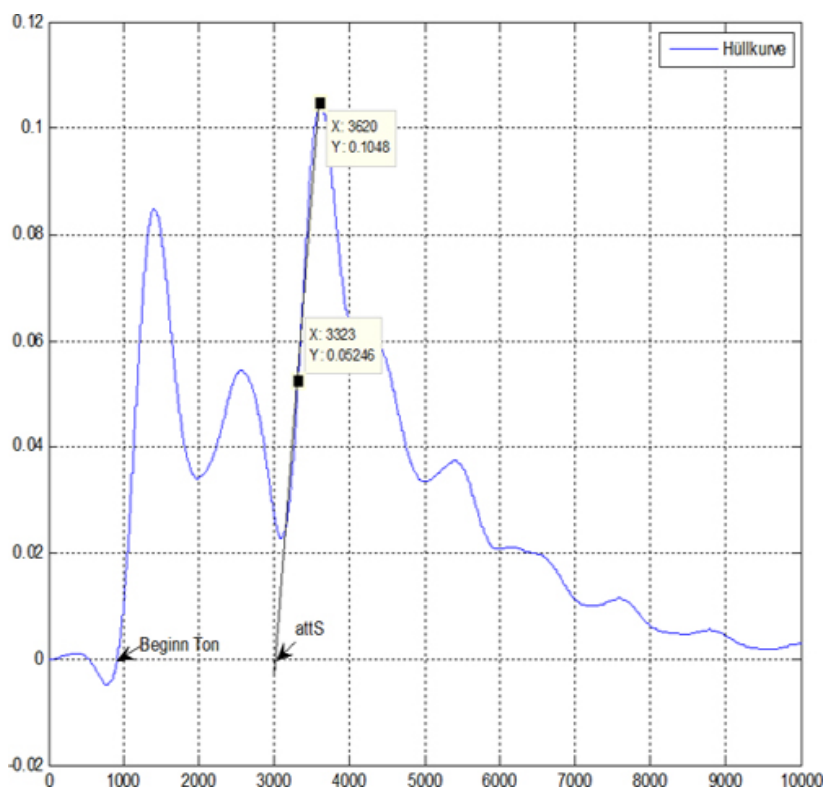


Abbildung 4.7.: Fehlerhafte Ermittlung vom Beginn der Attack-Phase

Der Grund für die hier zu sehende Unstetigkeit zu Beginn des nontonalen Signalanteils könnte an dem unter (vC09) angesprochenen Problem der Trennung von tonalem und nontonalen Anteil der Signale liegen, in manchen Fällen könnten wichtige Teile der Transienten im Frequenzspektrum genau dort liegen, wo dieses beschnitten wird, um die dort ebenfalls lokalisierten tonalen Anteile zu eliminieren. Eine andere Begründung liegt in der Natur des Anschlagvorgangs, aus dem sich die in der Hüllkurve in Abbildung 4.7 deutlich zu sehenden

zwei Maxima mit zwischenzeitlichem Abklingen ergeben könnten: wie im Kapitel 3.2 bereits beschrieben wurde, ist im transienten Signalanteil neben dem Anschlaggeräusch der Saite auch eine Resonanz des Gitarrenkorpus auf den Anschlag enthalten. Je nach Dicke der Saite könnten sich diese Momente unterschiedlich ausprägen, bei unterschiedlicher Dicke der Saiten könnten beide Ereignisse des Anschlag zeitlich weiter auseinander rücken, was der Vergleich der Abbildung 4.6 (dünne Saite) und 4.7 (dicke Saite) vermuten lassen könnte. Die bei einigen Gitarrenklängen auftauchende Charakteristik der Einhüllenden ähnlich Abbildung 4.7 lässt in ihrem Ausmaß die Vermutung zu, dass es sich hier nicht mehr um die in (vC09) beschriebene, durch unsauberes Spielen hervorgerufene, Pre-Attack handelt.

Angesichts dieser Betrachtung erscheint die alte Bestimmung von  $attS$  als ungeeignet, in dem hier entwickelten Verfahren wurde daher die Hüllkurve vom ermittelten Maximum aus nach links, bis zu dem 0.05-fachem Wert des Maximum, abgefahren und dort  $attS$  gesetzt. Dies erscheint als geeigneter Kompromiss für die Bestimmung geeigneter Hüllkurven-Parameter in Anbetracht der vereinzelt Hüllkurven-Formen ähnlich Abbildung 4.7. Die Parameter der Hüllkurve ermitteln sich also nun wie folgt:

$$\begin{aligned}
 attS &= 0.05 * max(env) \\
 attE &= 0.9 * max(env) \\
 rel1 &= 0.9 * max(env) \\
 rel2 &= 0.5 * max(env) \\
 rel3 &= 0.2 * max(env) \\
 relE &= 0.05 * max(env)
 \end{aligned}
 \tag{4.7}$$

Wie im vonColer-Verfahren wird nun der Teil zwischen  $attS$  und  $relE$  des nontonalen Signals als das Anschlaggeräusch betrachtet. Das nontonale Signal  $s_{nonton}(t)$  wird mit der aus den ermittelten Parametern erstellten APR-Hüllkurve  $APR(t)$  multipliziert und das Resultat als transienter Signalanteil  $s_{tran}(t)$  definiert. In den Abbildungen 4.8 und 4.9 sind die ermittelte Hüllkurve sowie die aus den APR-Hüllkurven-Parametern erstellte APR-Hüllkurve für den Gitarrenton *hoffmannL hense sonor S1-VI Bb5-1* abgebildet.

Dieses Anschlaggeräusch wird nun mit der Wavelet-Transformation modelliert. Hierzu wird in einem ersten Verfahren eine einfache, schnelle Wavelet-Transformation des nontonalen Signalanteils durchgeführt, nach dem unter 2.3 vorgestelltem Verfahren, wie es in Abbildungen 2.5 und 2.6 dargestellt ist. Diese hat gegenüber der gefensterter Fourier-Transformation, also der Verwendung von Sinusoiden, den unter Kapitel 3 bereits angesprochenen Vorteil, nicht stationäre Signale in Bezug auf die Zeitauflösung, und insbesondere bei schnellen Veränderungen im Signal, gut darstellen zu können und dabei auch bei hoher Komprimierungsrate noch sehr befriedigende Ergebnisse erzielen zu können.

In einem ersten Schritt wird für das jeweilige Signal eine bestgeeignete Tiefe der Wavelet-Transformation bestimmt. Hierzu stellt Matlab die Funktion *wmaxlev.m* zur Verfügung, welche

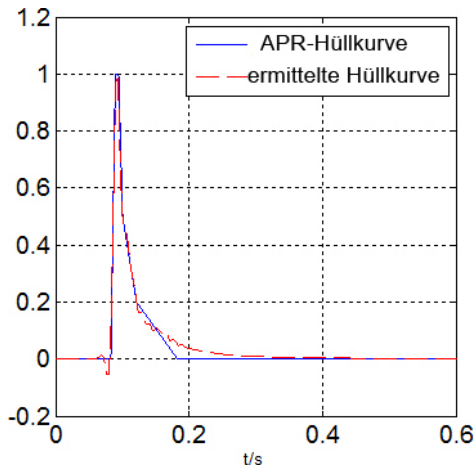


Abbildung 4.8.: APR-Hüllkurve

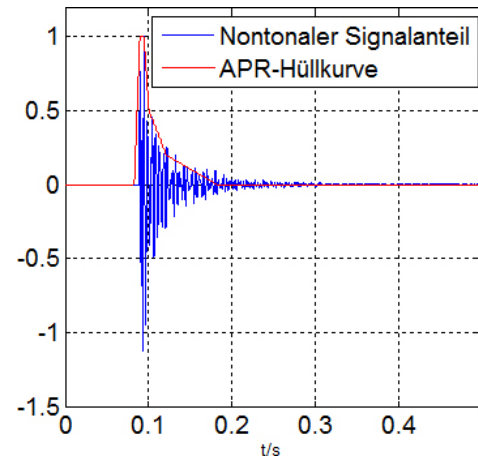


Abbildung 4.9.: Ermittelte Hüllkurve

eine maximal vernünftige Skalen-Tiefe für eine Wavelet-Transformation bestimmt. In diesem Fall wird die maximal vernünftige Skalen-Tiefe für jeden Gitarrenklang auf 12 bestimmt. Anschliessend wird das nontonale Signal Wavelet-transformiert. Neben der Tiefe ist hier die gewählte Wavelet-Basisfunktion von entscheidender Bedeutung. In Tabelle 4.1 sind exemplarisch für Transienten der Gitarrentöne *hoffmannL hense sonor* für die Gitarrenseite 1 sowie die Gitarrenseite 6 die Komprimierungsraten für verschiedene Wavelet-Basen bei gleichbleibenden Threshold abgebildet, in diesem Fall 0.01.

Aus diesen Ergebnissen lässt sich zunächst ablesen, dass bei den orthogonalen Wavelet-Basen Daubechies-Wavelet und Symlets-Wavelets mit steigender Anzahl verschwindender Momente (im folgenden als höhere Ordnung bezeichnet) (siehe Kapitel 2) zumeist auch die Kompressionsrate ansteigt, jedoch ab einer gewissen Grenze nicht mehr wesentlich. Die nicht-orthogonalen Biorthogonal-Wavelet-Basen zeigen hier leicht schlechtere Werte. Auch fällt eine allgemein bessere Kompressionsrate bei der tieferen Gitarrensaite auf - das hängt jedoch mit dem deutlich niedrigerem Energie-Level ihres transienten Signals und den damit verbundenen kleineren Wavelet-Koeffizienten zusammen. Insgesamt scheint sich die Wahl der Wavelet-Basis in ihrer Wellenform nicht wesentlich auf die Kompressionsrate auszuwirken.

Von entscheidenderer Bedeutung ist bei der Wahl der Wavelet-Basis jedoch der eigentliche Höreindruck, der sich nicht in Zahlen oder Graphen ausdrücken lässt. Hier lassen sich bei den gröberen, weniger welligen Wavelet-Basen niedrigerer Ordnung, wie db1, db4 oder sym4, klare Qualitätsunterschiede zu denen höherer Ordnung ausmachen, man hört vermehrt Artefakte und Störgeräusche heraus oder die Transienten klingen dumpfer und platter. Die rekonstruierten Signale der mittels Biorthogonaler-Wavelet-Basen transformierten Transienten klingen deutlich schlechter. Hingegen lässt sich zwischen der Transformation mit der Daubechies-8- und Daubechies-12 Basis kein Unterschied mehr hören, auch das Resultat

Wavelet-Basis	Gitarrensaite 1 [%]	Gitarrensaite 6 [%]
Haar(db1)	99,40	99,68
db4	99,73	99,83
db8	99,73	99,84
db12	99,73	99,83
sym4	99,71	99,84
sym8	99,74	99,84
bior2.6	99,68	99,81
bior4.4	99,73	99,85

Tabelle 4.1.: Vergleich Komprimierungsraten bei verschiedenen Wavelet-Basen

der sym8-Transformation unterscheidet sich nicht mehr hörbar davon. Da zwischen db8 und db12 kaum noch ein Unterschied bezüglich der Komprimierungsrate auszumachen ist, wird für das angewandte Verfahren, wie eingangs bereits erwähnt, die Daubechies-8 Wavelet-Basis als geeignete verwendet.

Mit der Matlabfunktion *wavedec.m* werden die Koeffizienten bestimmt. Die Verteilung der Wavelet-Koeffizienten für die Wavelet-Transformation (mit Wavelet-Basis Daubechies-8, Skalentiefe 12) der Transienten der ersten Gitarrensaite (*hoffmannL hense sonor S2-VI F5-7*) ist im Abbildung 4.10 dargestellt. In Tabelle 4.2 ist die Zuordnung der Koeffizienten auf die zwölf Skalen dargestellt ( $c_k$  = Koeffizienten der Skala  $k$ ,  $a$  = Approximation Stufe 12).

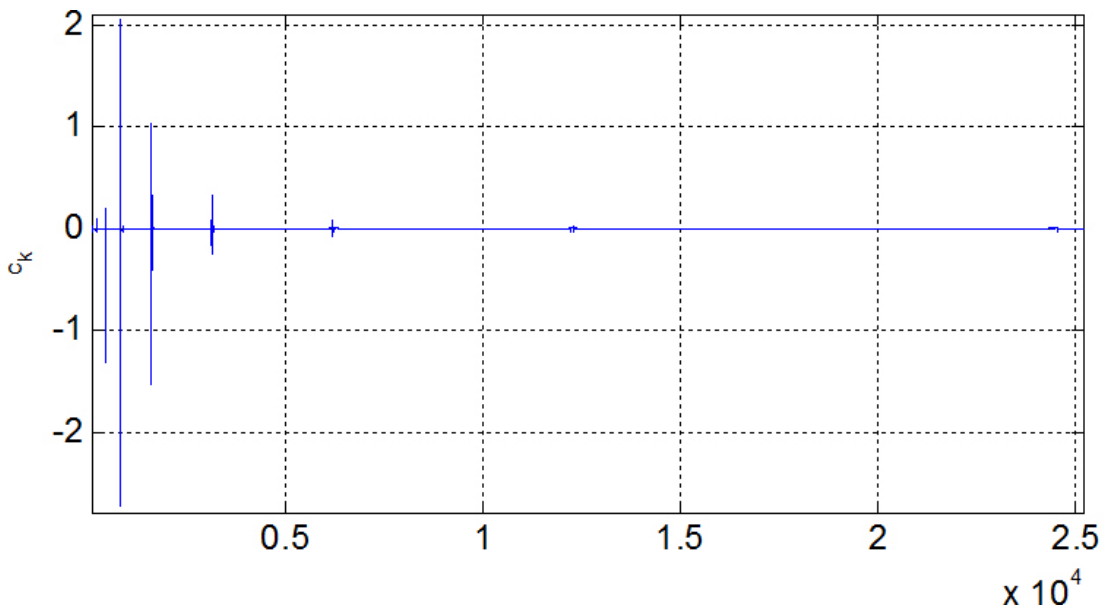


Abbildung 4.10.: Verteilung der Wavelet-Koeffizienten über die Skalen der Wavelet-Transformation

Koeffizienten	Zuordnung
$a$	1 – 37
$c_{12}$	38 – 74
$c_{11}$	75 – 134
$c_{10}$	135 – 239
$c_9$	240 – 434
$c_8$	435 – 810
$c_7$	811 – 1548
$c_6$	1549 – 3009
$c_5$	3010 – 5917
$c_4$	5918 – 11737
$c_3$	11738 – 23326
$c_2$	23327 – 46491
$c_1$	46492 – 92804

Tabelle 4.2.: Verteilung Wavelet-Koeffizienten

Zwei Eigenschaften fallen sofort auf - Die Koeffizienten konzentrieren sich auf ein paar wenige Regionen, und ab einer letzten Anhäufung im Bereich der zweiten Detail-Stufe (Skala 2) sind überhaupt keine Koeffizienten mehr vorhanden. Daraus lässt sich direkt auf den in Kapitel 2.3 hergestellten Bezug der Skalenregionen zur Frequenz, nach dem beim Subband-Coding gilt, dass jede Skalen-Stufe  $i$  genau Frequenzen im Bereich  $\frac{1}{2^{i+1}} \leq |\xi| \leq \frac{1}{2^i}$  enthält, schließen, dass die Detail-Stufen des transienten Signalanteils keine Frequenzen  $|\xi| > \frac{1}{2^2} = 11025Hz$  bei  $f_s = 44100Hz$  enthält, die Informationen über diese Frequenzen stehen alleine in  $a$ . Bei einem Zoom auf  $a$  (Abbildung 4.11) sehen wir hier lediglich einen Koeffizienten oberhalb von  $|0.005|$ , es kann also davon ausgegangen werden, dass die Transiente wenige bis keine wichtigen Anteile oberhalb von  $11025Hz$  enthält.

Die größten Wavelet-Koeffizienten konzentrieren sich auf vier Bereiche, dargestellt in Abbildung 4.12 bis 4.15. Diese vier Abschnitte liegen in den Skalierungsstufen 5, 6, 7 und 8. Auf Frequenzen bezogen lässt sich also daraus anhand des in Kapitel 2.3 hergestellten Bezugs der Skalenregionen zur Frequenz folgern, dass sich die hauptsächlichen Signalanteile der Transienten zwischen den Frequenzen  $86, 1328Hz$  und  $1378Hz$  befinden. Hier wird deutlich, dass die Wavelet-Transformation in Bezug auf Frequenz-Eigenschaften zunächst keine sonderlich detaillierten Informationen über ein Signal liefert.

Nun ist ein geeigneter harter Threshold  $t_h$  zu ermitteln, unter dem die Koeffizienten gelöscht, also gleich Null gesetzt werden. Auf der einen Seite ist eine möglichst hohe Komprimierungsrate erwünscht, auf der anderen Seite ist diese verlustbehaftet, und man möchte nach

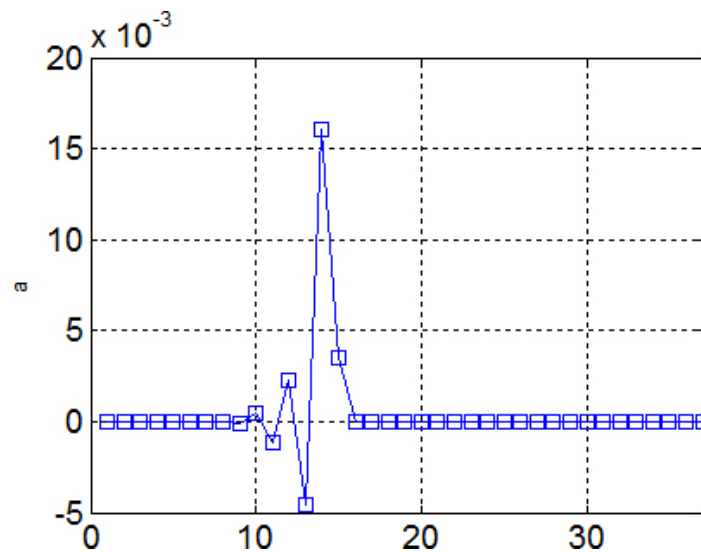


Abbildung 4.11.: Approximations-Koeffizienten

einer Rekonstruktion ein noch möglichst echt klingendes Signal erhalten. Bei der Komprimierungsrate wird sich an dem alten Verfahren mit Sinusoiden orientiert. Dort wird, unabhängig von den zur Synthese ebenfalls benötigten Hüllkurvenparametern, pro Gitarrenklang eine Anzahl von 330 Werten für die Sinusoide (110 Sinusoide, pro Sinusoid 3 Werte Amplitude, Frequenz und Phase) zur Transientenmodellierung benötigt. Bei der Wavelet-Transformation schwankt dieser Wert von Ton zu Ton, da die Anzahl der jeweils unter dem Threshold liegenden Koeffizienten von Signal zu Signal unterschiedlich ist. In Abbildung 4.16 sind für drei verschiedene Thresholds die jeweilig resultierende Menge von Koeffizienten ungleich Null dargestellt, für 24 Gitarrenklänge von Typ *hoffmannL hense sonor*, für die Transienten von jeweils vier Klangaufzeichnungen pro Saite.

Unmittelbar fällt auf, dass die Anzahl der jeweils übrig bleibenden Koeffizienten auch bei gleich bleibendem Threshold stark schwankt. Ursache dafür ist die unterschiedliche Signal-Charakteristik der jeweiligen Anschlaggeräusche, nur scheint es hier keine Linearität zu geben, die zum Beispiel aussagen würde, dass mit zunehmender Dicke der jeweiligen Seite mehr oder weniger Koeffizienten für eine bestimmte Rekonstruktions-Qualität benötigt werden. Zu sehen sind aber Regelmäßigkeiten zwischen verschiedenen Seiten, so produziert die Transiente der vierten Seite deutlich weniger Koeffizienten über dem Threshold als andere Seiten, die erste Seite hingegen offenbar deutlich mehr. Der Grund wird deutlich, wenn man sich die nontonalen Signale anhört, gerade bei z.B. der vierten Saite fallen diese sehr dumpf und flach aus, was seinen Ursprung in der in (vC09) schon beschriebenen

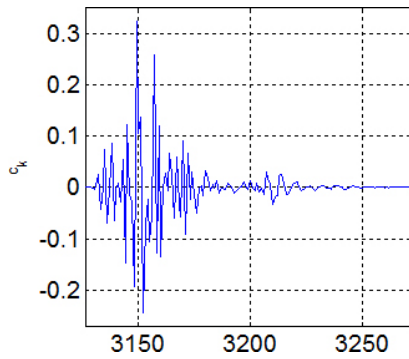


Abbildung 4.12.: Wavelet-Koeffizienten der Skala 5

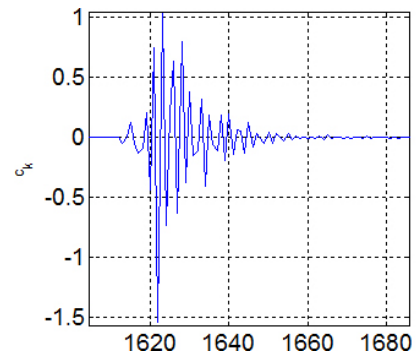


Abbildung 4.13.: Wavelet-Koeffizienten der Skala 6

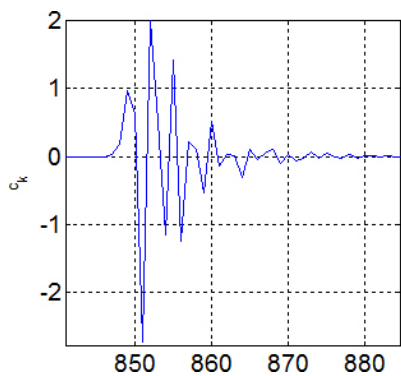


Abbildung 4.14.: Wavelet-Koeffizienten der Skala 7

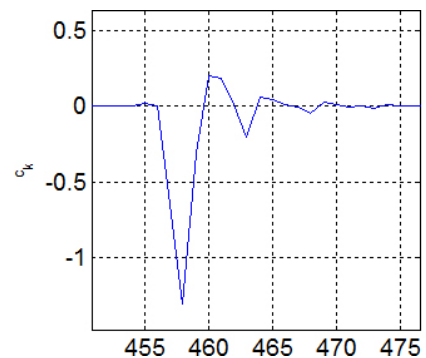


Abbildung 4.15.: Wavelet-Koeffizienten der Skala 8

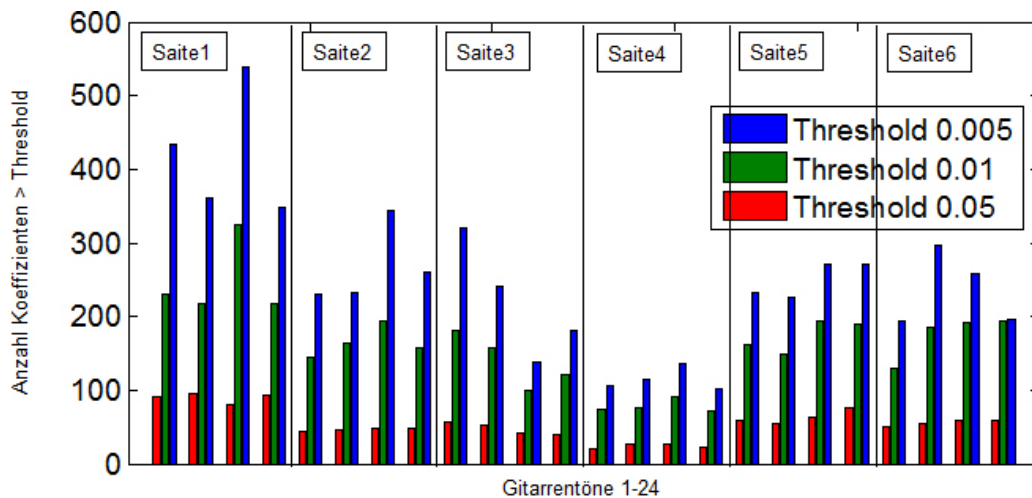


Abbildung 4.16.: Anzahl Koeffizienten bei verschiedenen Saiten

Problematik der Signaltrennung hat, offenbar entfernt das Verfahren neben dem tonalem Anteil hier auch Teile des transienten Signalanteils.

Ein Schema, das sich logisch auf bestimmte Eigenschaften der einzelnen Gitarrensaiten bezieht, lässt sich nicht erkennen. Die Schwankungen der Anzahl signifikanter Koeffizienten bei den vier verschiedenen Aufzeichnungen für einzelne Saiten lassen aber erkennen, dass die unterschiedliche Härte des Saitenanspiels, was sich auf die Schärfe der Transienten auswirkt, auch auf die Menge größerer Koeffizienten auswirkt. Eine weiterführende Untersuchung in diese Richtung könnte zu neuen Erkenntnissen bezüglich der Modellierungseigenschaften der Wavelet-Transformation bezüglich transienter Signale führen.

Bezüglich der Wahl eines geeigneten Thresholds ist 0.005 eine gute Wahl, hier bleiben, wie Abbildung 4.16 zu entnehmen, um die 300 Koeffizienten übrig, also eine Anzahl ähnlich der des von Coler-Verfahrens mit Sinusoiden.

Für eine effiziente digitale Speicherung der Wavelet-Koeffizienten müssen diese noch codiert werden. In diesem Fall wird eine einfache Entropie-Codierung benutzt, die in einer Wertefolge alle Koeffizienten ungleich Null speichert, und in einer weiteren Wertefolge die Anzahl jedem Koeffizienten vorstehender Nullen. Dieses Verfahren ist einfach zu implementieren, hat jedoch den Nachteil, doppelt so viele Variablen wie Koeffizienten ungleich Null zur Modellierung zu benötigen. Hier könnten geschicktere, binäre Verfahren zu einer effektiveren Codierung führen.

Für die Modellierung des transienten Teils werden in der Analyse aus dem nontonalen Signal also pro Gitarrenklang folgende Parameter bestimmt: Sechs APR-Hüllkurven-Parameter, sowie die mit einem Threshold von 0.005 komprimierten Wavelet-Koeffizienten. Da letztere in einer Folge von durch unregelmäßig mit vielen Nullen unterbrochenen Werten vorliegen, müssen diese noch einer Entropie-Codierung unterzogen werden.

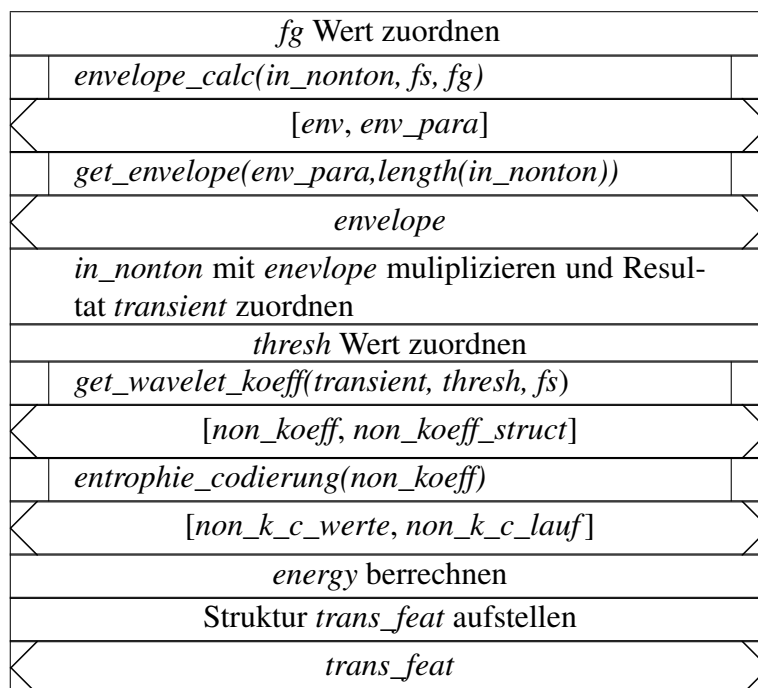


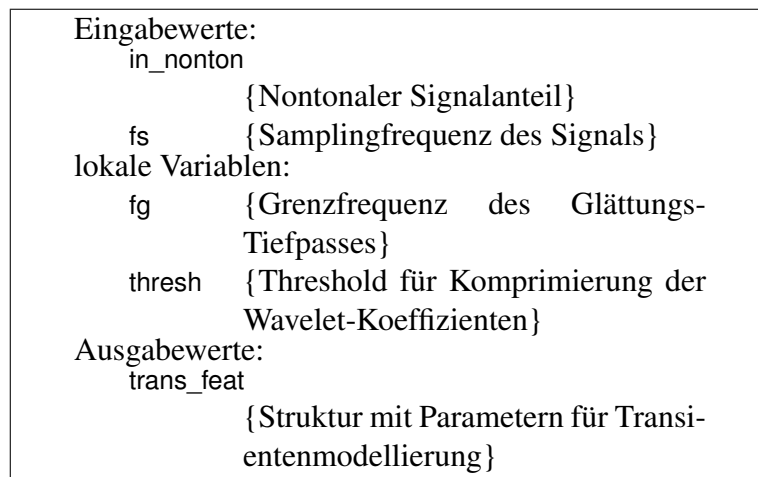
### Programmablauf der Analyse

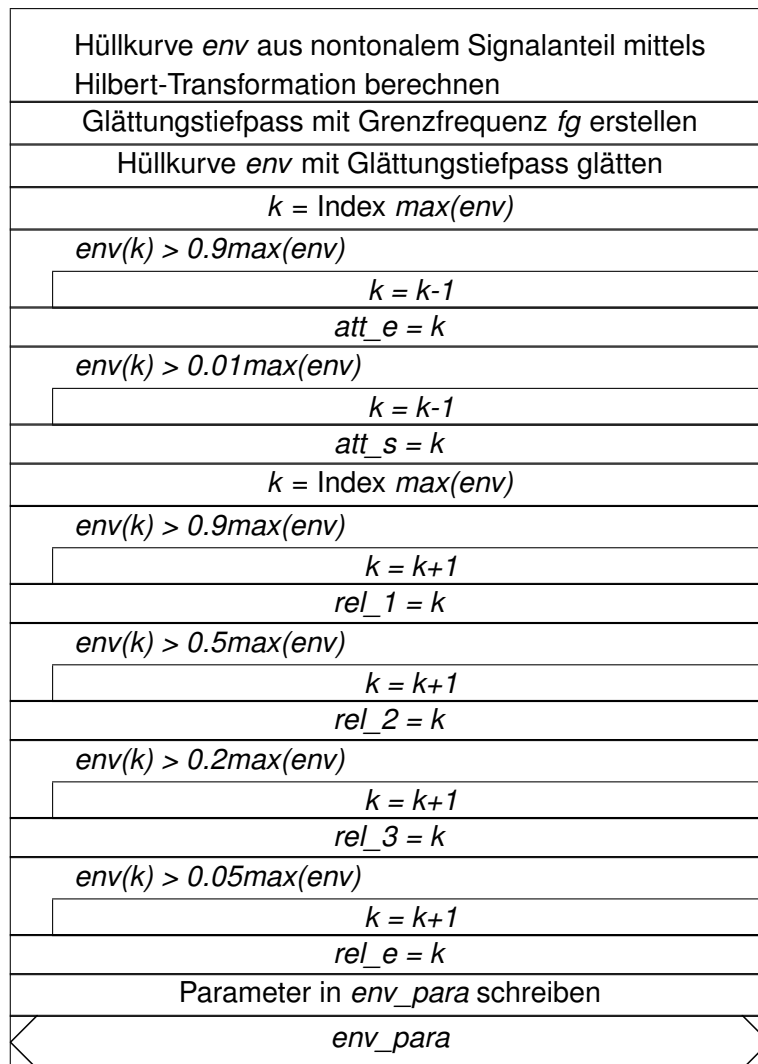
Der Programmablauf der Analyse der Transienten lässt sich folgendermaßen umschreiben:

- Mit der Funktion *analysis\_features\_trans.m* werden aus dem nontonalen Signal *in\_nonton* die Parameter für die Transienten-Anteile der Gitarrenklänge extrahiert und mittels der Struktur *trans\_feat* übergeben, die in Struktogramm 4.11 beschrieben ist. Der Programmablauf ist im Struktogramm 4.12 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.13.
- Zunächst wird für die Hüllkurven-Bestimmung eine Grenzfrequenz *fg* für den Glättungstiefpass bestimmt. Dann wird mit der Funktion *envelope\_calc* die Hüllkurve *env* bestimmt, sowie die sechs APR-Hüllkurven-Parameter *env\_para* anhand dieser Hüllkurve bestimmt. Zur Hüllkurven-Bestimmung wird das nontonale Signal zunächst Hilbert-Transformiert, die resultierende Hüllkurve anschließend mit einem Tiefpass der Grenzfrequenz *fg* geglättet. Die APR-Hüllkurven-Parameter werden anhand des Schemas aus 4.7 bestimmt. Der Programmablauf ist im Struktogramm 4.14 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.15.
- Im nächsten Schritt wird aus den APR-Hüllkurven-Parametern mit der Funktion *get\_envelope* die APR-Hüllkurve erstellt, und diese mit dem nontonalen Signal multipliziert, das Ergebnis ist der transiente Signalanteil *transient*. Der Programmablauf von *get\_envelope* ist im Struktogramm 4.16 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.17.
- Nachdem ein Threshold-Wert *thresh* für die Komprimierung der Wavelet-Koeffizienten festgelegt ist, werden mit der Funktion *get\_wavelet\_koeff* die Wavelet-Koeffizienten des transienten Signalanteils bestimmt. Innerhalb dieser Funktion wird zunächst mit der Matlab-Funktion *wmaxle* die optimale Tiefe der Wavelet-Transformation für die zuvor festgelegte Wavelet-Basis *w\_name* bestimmt. Im nächsten Schritt wird mit der Matlab-Funktion *wavedec* die Wavelet-Koeffizienten *koeff* und deren Struktur *koeff\_struct* der transienten Signals, in der bestimmten Tiefe, mit der gewählten Wavelet-Basis, bestimmt. Der Programmablauf von *get\_wavelet\_koeff* ist im Struktogramm 4.18 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.19.
- Im nächsten Schritt werden die Koeffizienten Codiert. Die Funktion *entrophie\_codierung* führt eine Entrophie-Codierung durch, und übergibt zwei Wertefolgen *non\_k\_c\_werte* und *non\_k\_c\_lauf*. Der Programmablauf von *get\_envelope* ist im Struktogramm 4.20 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.21.
- Im letzten Schritt wird die Energie *energy* der originalen Transienten bestimmt.

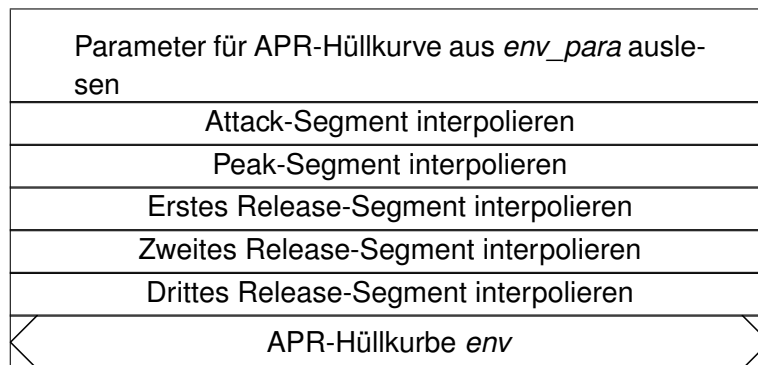
Elemente von <i>trans_feat</i>	
<i>env_para</i>	{Stützpunkte der APR-Hüllkurve}
<i>env</i>	{Hüllkurve Nontonaler Signalanteil}
<i>k_werte</i>	{Werte codierte Wavelet-Koeffizienten}
<i>k_lauf</i>	{Laufängen codierte Wavelet-Koeffizienten}
<i>length_koeff</i>	{Ursprüngliche Länge Wavelet-Koeffizienten-Vektor}
<i>fs</i>	{Samplingfrequenz transienter Signalanteil}
<i>energy</i>	{Energie original Transiente}

Struktogramm 4.11: Die Struktur *trans\_feat*Struktogramm 4.12: Programmablauf zu *analysis\_features\_trans.m*

Struktogramm 4.13: Variablen zu *analysis\_features\_trans.m*

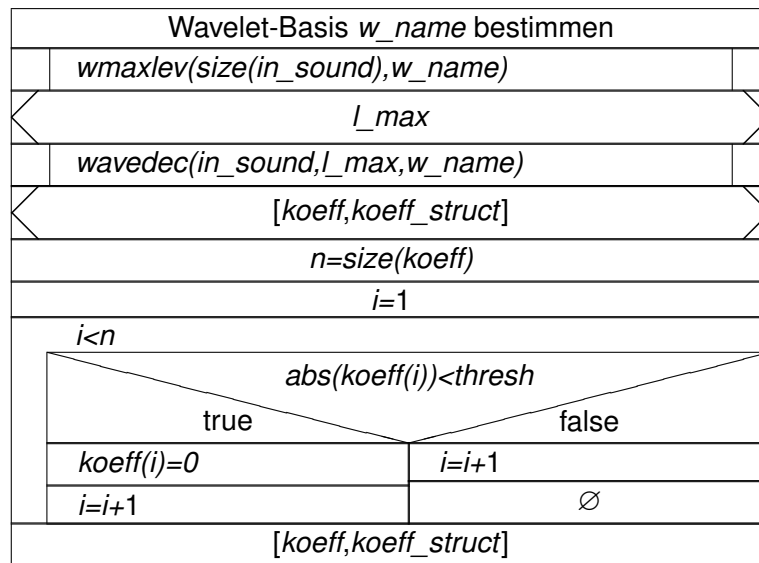
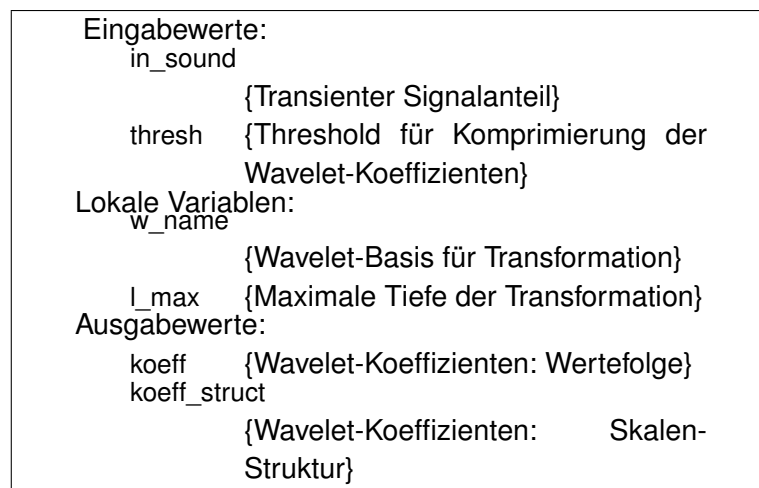
Struktogramm 4.14: Programmablauf zu *envelope\_calc.m*

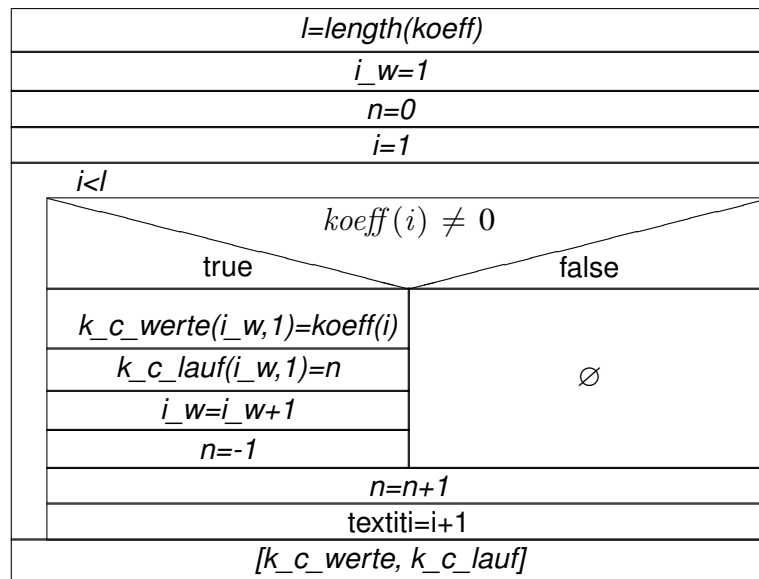
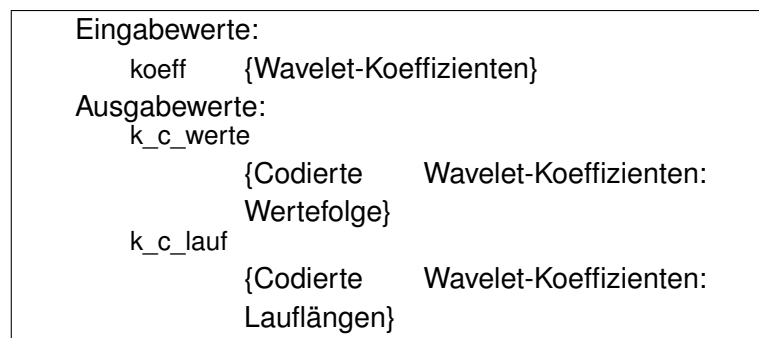
Eingabewerte:	
in_nonton	{Nontonaler Signalanteil}
fs	{Samplingfrequenz des Signals}
fg	{Grenzfrequenz für Glättungstiefpass}
Ausgabewerte:	
env	{Hüllkurve vom nontonalen Signalanteil}
env_para	{Parameter APR-Hüllkurve}

Struktogramm 4.15: Variablen zu *envelope\_calc.m*Struktogramm 4.16: Programmablauf zu *get\_envelope.m*

Eingabewerte:	
env_parameter	{Parameter für APR-Hüllkurve}
len	{Länge der Hüllkurve}
Ausgabewerte:	
env	{APR-Hüllkurve}

Struktogramm 4.17: Variablen zu *get\_envelope.m*

Struktogramm 4.18: Programmablauf zu *get\_wavelet\_koeff.m*Struktogramm 4.19: Variablen zu *get\_wavelet\_koeff.m*

Struktogramm 4.20: Programmablauf zu *entrophie\_codierung.m*Struktogramm 4.21: Variablen zu *entrophie\_codierung.m*

### 4.3.2. Synthese des transienten Anteils mit schneller Wavelet-Transformation

Die Synthese der Transienten der Gitarrenklänge erfolgt nun mittels der im vorherigen Abschnitt ermittelten Parameter. Sie erfolgt nach dem Schema in Blockschaltbild 4.17.

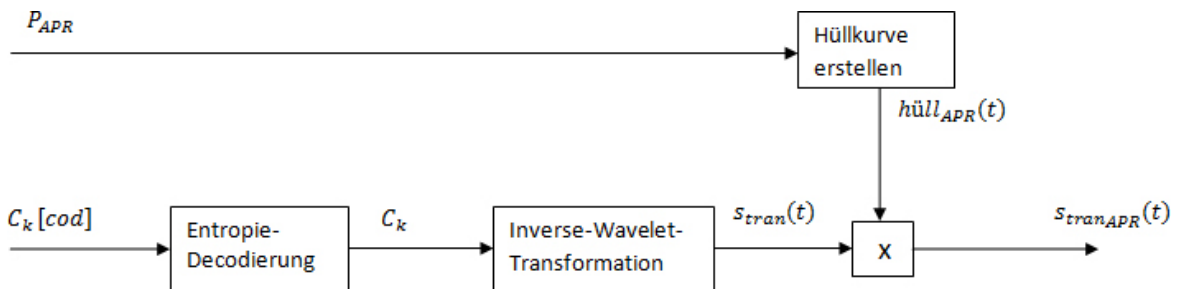


Abbildung 4.17.: Blockschaltbild Transienten-Rekonstruktion

Die Multiplikation der mittels inverser Wavelet-Transformation wiederhergestellten Transienten mit der aus den Hüllkurven-Parametern rekonstruierten APR-Hüllkurve ist dabei Optional und für die Synthese an sich nicht zwingend notwendig, bietet jedoch in der Modellierung bestimmte Modifikationsmöglichkeiten - an dieser Stelle könnten die verschiedenen Eigenschaften des Anschlaggeräusches wie Dauer oder Schärfe der Attack-Phase oder der Abklingphase direkt verändert werden. An dieser Stelle tritt so ein gravierender Vorteil der Transienten-Modellierung mittels Wavelet-Transformation zu Tage - da in ihr im Gegensatz zu den Sinusoiden neben der Frequenzinformation auch die Zeitinformation des Signals enthalten ist, können die Transienten ohne eine Multiplikation mit einer Hüllkurve rekonstruiert und somit resynthetisiert werden. Zudem sorgt das Thresholding der Wavelet-Koeffizienten, wie in Kapitel 3.2 gezeigt wurde, automatisch zu einer Eliminierung der Residue des Ausgangssignal.

Zunächst müssen die codierten Wavelet-Koeffizienten  $C_k[cod]$  decodiert werden, hierzu wird ein der Entropie-Codierung aus Kapitel 2.3 inverses Verfahren angewandt. Im Anschluss daran werden die Transienten  $s_{trans}(t)$  aus den Wavelet-Koeffizienten  $C_k$  mit einer inversen, schnellen Wavelet-Transformation rekonstruiert, Matlab stellt hierzu die Funktion *waverec.m* zur Verfügung, die diese mit einer gewählten Wavelet-Basis rekonstruiert, hier ist die gleiche Basis wie bei der Analyse zu wählen.

Anschließend werden aus den APR-Hüllkurven-Parametern  $P_{APR}$  die APR-Hüllkurve  $hüll_{APR}(t)$  erstellt, und diese mit den rekonstruierten Transienten multipliziert. Dieser Schritt wäre rein formal nicht mehr nötig, aufgrund der Eigenschaften der Wavelet-Transformation ist die Zeitinformation, wie bereits erwähnt, in den Transienten bereits enthalten. Die Abbildung 4.18, zwei rekonstruierte Transiente  $s_{trans}(t)$  mit der jeweils zugehörigen APR-Hüllkurve  $hüll_{APR}(t)$  (für die Gitarrenklänge *hoffmannL hense sonor S1-VI Bb5-1* und *hoffmannL hense sonor S2-VI F5-1*), veranschaulicht diesen Sachverhalt.



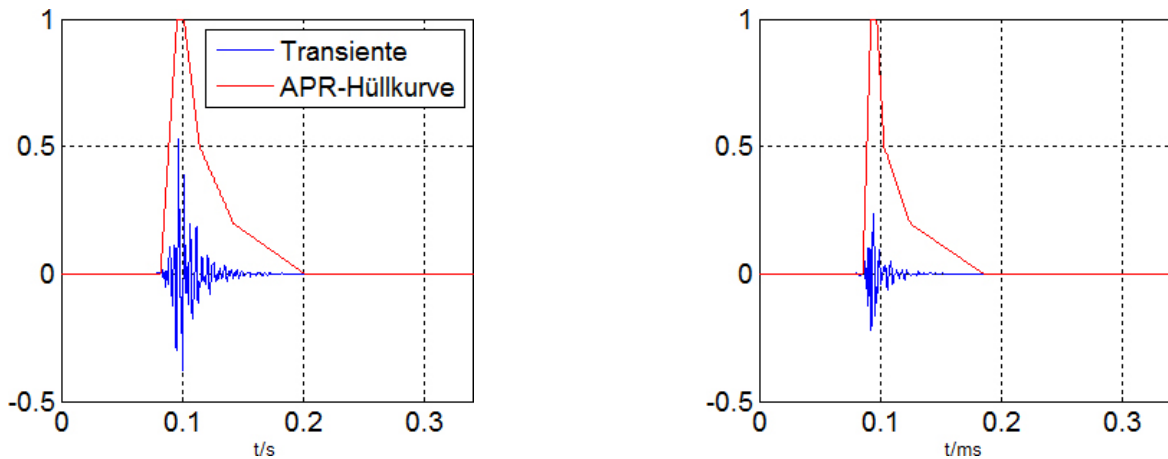


Abbildung 4.18.: Transiente und APR-Hüllkurven für zwei verschiedene Gitarrenklänge

Dieser Schritt wird hier lediglich zur Offenhaltung bereits angesprochener Modifikationsmöglichkeiten beibehalten. Es kommt dabei hinzu, dass die Gewichtung der rekonstruierten Transienten mit der APR-Hüllkurve noch zu einer Verfälschung dieser führt, da diese in ihren Amplitudenwerten über den Zeitverlauf ja bereits korrekt gewichtet ist. Ein Vergleich zweier resultierender Transienten, einmal synthetisiert mit anschließender Gewichtung durch die APR-Hüllkurve, einmal ohne (Abbildung 4.19), zeigt hier jedoch aber kaum nennenswerte Unterschiede im Verlauf, was mit der im Anschluss an die Rekonstruktion noch durchgeführten Angleichung an das Energie-Level des originalen, transienten Signalanteils zusammenhängt. An dieser Stelle gibt es also eine Wahlmöglichkeit zwischen einer möglichst genauen Rekonstruktion des transienten Signalanteils und der Möglichkeit, mittels Veränderung der APR-Hüllkurven-Parameter das Anschlaggeräusch zu modifizieren.

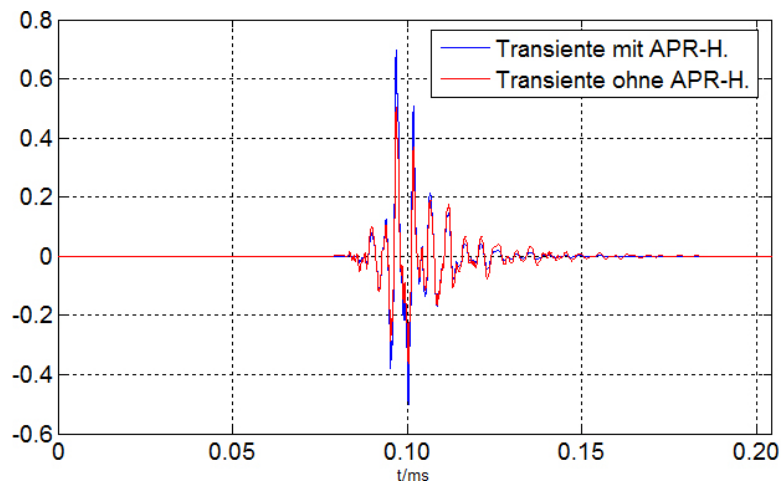


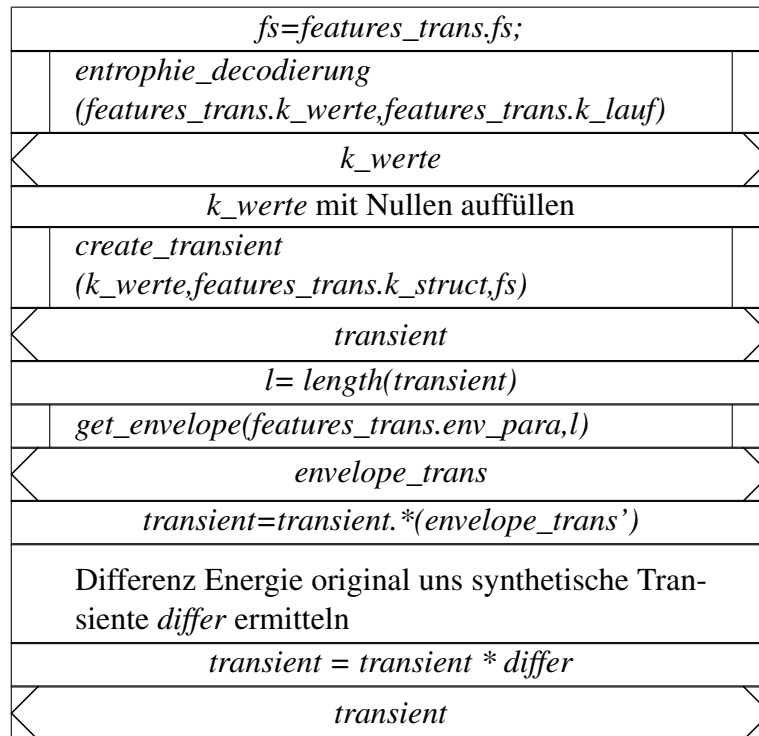
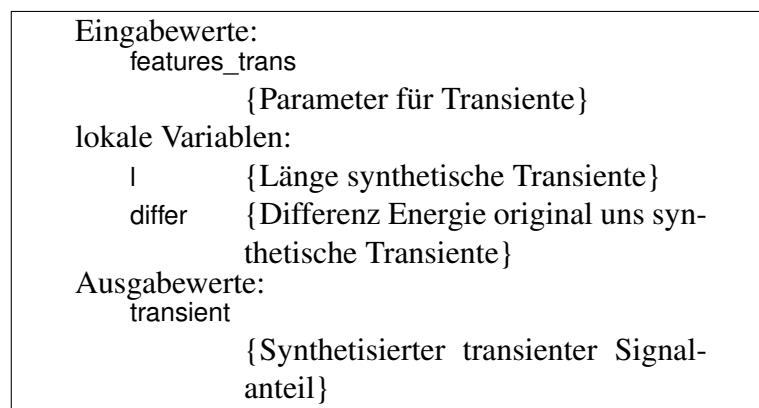
Abbildung 4.19.: Vergleich Synthetische Transienten mit und ohne APR-Hüllkurven-Verfahren

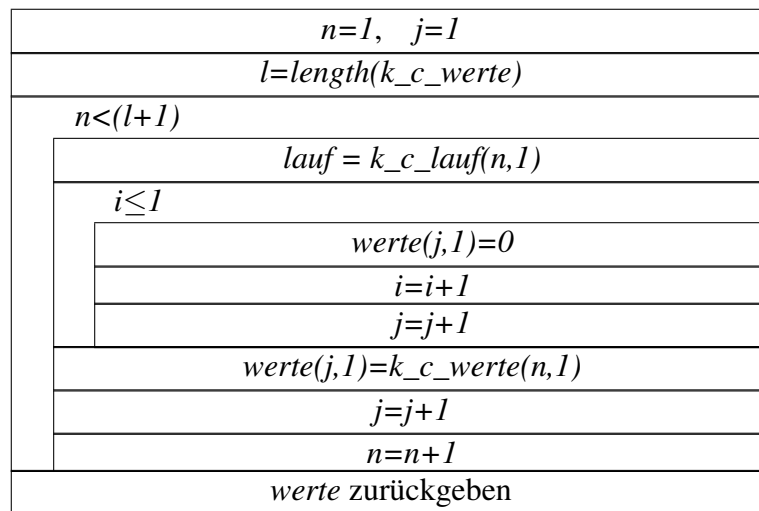
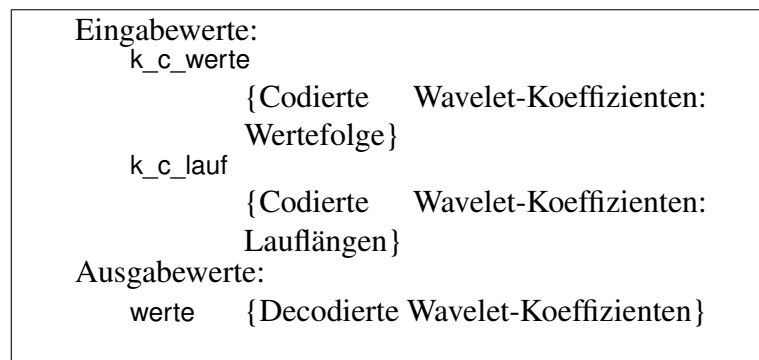
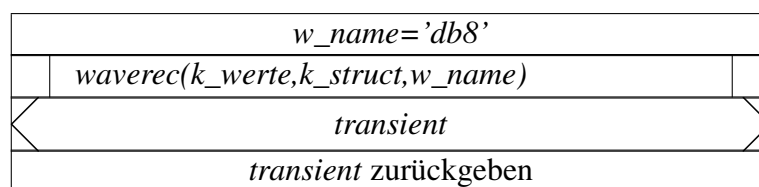
### Programmablauf der Synthese

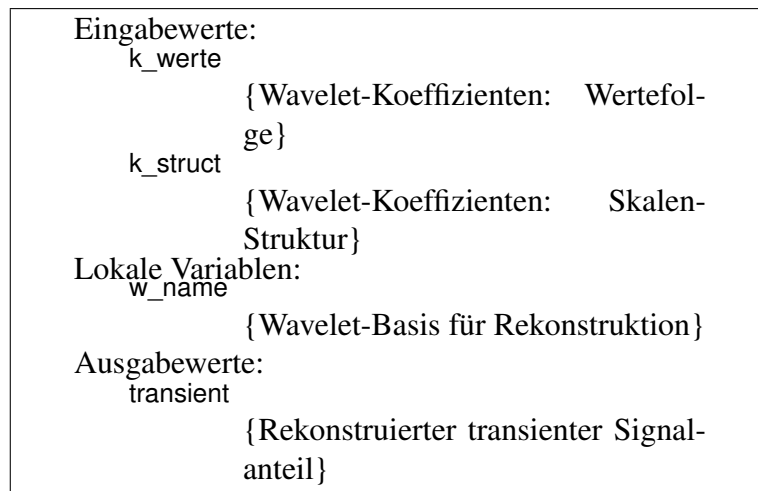
Der Programmablauf der Synthese der Transienten lässt sich folgendermaßen umschreiben:

- Mit der Funktion *resynth\_trans.m* werden aus den in der Analyse ermittelten Parametern *features\_trans* für die Transienten-Modellierung die Transienten der Gitarrentöne resynthetisiert und als *transient* zurückgegeben. Der Programmablauf ist im Struktogramm 4.22 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.23.
- Zunächst wird aus den Parametern *features\_trans* die Samplefrequenz *fs* ausgelesen. Dann werden aus den Parametern die Entropie-Codierten Wavelet-Koeffizienten an die Funktion *entrophie\_decodierung* übergeben, die diese decodiert. Der Programmablauf der Funktion *entrophie\_decodierung* ist im Struktogramm 4.24 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.25.
- Im nächsten Schritt werden an die decodierten Wavelet-Koeffizienten *k\_werte* noch die fehlenden Nullen angehängt. Dann wird aus ihnen in der Funktion *create\_transient* mit der inversen Wavelet-Transformation der transiente Signalanteil *transient* resynthetisiert. Der Programmablauf der Funktion *create\_transient* ist im Struktogramm 4.26 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.27.
- Für die APR-Hüllkurven-Ermittlung wird die Länge *l* der synthetischen Transienten ermittelt. Im nächsten Schritt wird mir der bereits im vorhergegangenen Unterkapitel beschriebenen Funktion *get\_envelope* die APR-Hüllkurve *envelope\_trans* erstellt, und diese mit der synthetischen Transienten multipliziert.

- Abschließend wird die Differenz *differ* der Energie der originalen Transienten und der synthetischen Transienten ermittelt, und letztere entsprechend angepasst. Der synthetisch erzeugte, transiente Signalanteil wird übergeben.

Struktogramm 4.22: Programmablauf zu *resynth\_trans.m*Struktogramm 4.23: Variablen zu *resynth\_trans.m*

Struktogramm 4.24: Programmablauf zu *entrophie\_decodierung.m*Struktogramm 4.25: Variablen zu *entrophie\_decodierung.m*Struktogramm 4.26: Programmablauf zu *create\_transient.m*

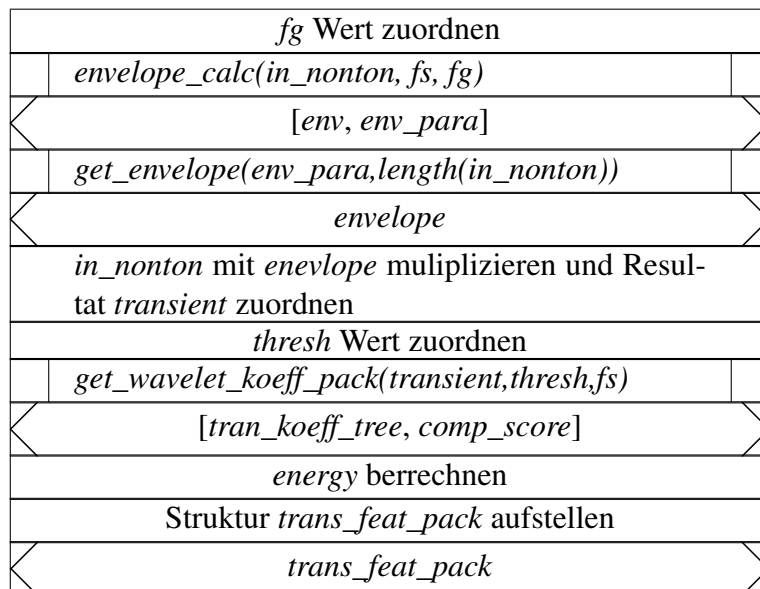
Struktogramm 4.27: Variablen zu *create\_transient.m*

### 4.3.3. Analyse und Synthese des transienten Anteils unter Verwendung von Wavelet-Paketen

In den beiden vorherigen Abschnitten vorgestelltem Verfahren zur Analyse und Synthese des transienten Signalanteils wird die schnelle Wavelet-Transformation angewandt.

Alternativ dazu wurde das Verfahren mit den unter Kapitel 2 vorgestellten Wavelet-Pakete durchgeführt. Auch hierzu stellt Matlab einige Funktionen zur Verfügung, mit Hilfe dieser wurde das Analyse- und Synthese Verfahren entsprechend modifiziert.

In der Analyse wird *analysis\_features\_trans.m* durch *analysis\_features\_trans\_pack.m* ersetzt. Der Programmablauf ist im Struktogramm 4.28 dargestellt, die Eingabe- sowie Ausgabewerte in Struktogramm 4.29. Die alternative Parameter-Struktur *trans\_feat\_pack* ist im Struktogramm 4.30 dargestellt. Sie unterscheidet sich in sofern, als dass in ihrem Verlauf die Transiente einer Transformation in Koeffizienten einer Wavelet Paket Struktur erfolgt, mit der gleichen Wavelet Basis, Daubechies-8. diese Transformation wird in der Funktion *get\_wavelet\_koeff\_pack.m*, die die Funktion *get\_wavelet\_koeff.m* ersetzt, durchgeführt (Programmablauf im Struktogramm 4.31, Eingabe- sowie Ausgabewerte in Struktogramm 4.32). Die Matlab Funktion *wpdencmp* ermittelt für eine gegebene Waveletbasis *w\_name*, einer gegebenen Skalen-Tiefe (wir wählen aus Performance-Gründen eine Tiefe von zehn, höhere Werte führen zu erheblichen Rechenzeiten) und einem festgelegtem Threshold (auch hier wählen wir einen Wert von 0.005) die Wavelet Paket Koeffizienten, wobei die Baumstruktur automatisch dem zu transformierendem Signal angepasst wird, nach einem Entropie-Kriterium nach (RC92).



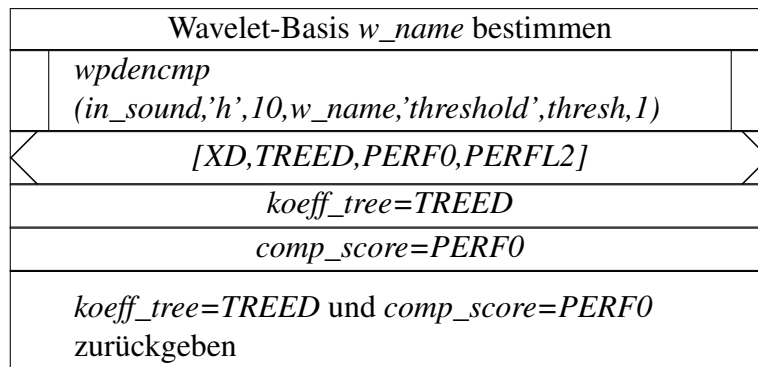
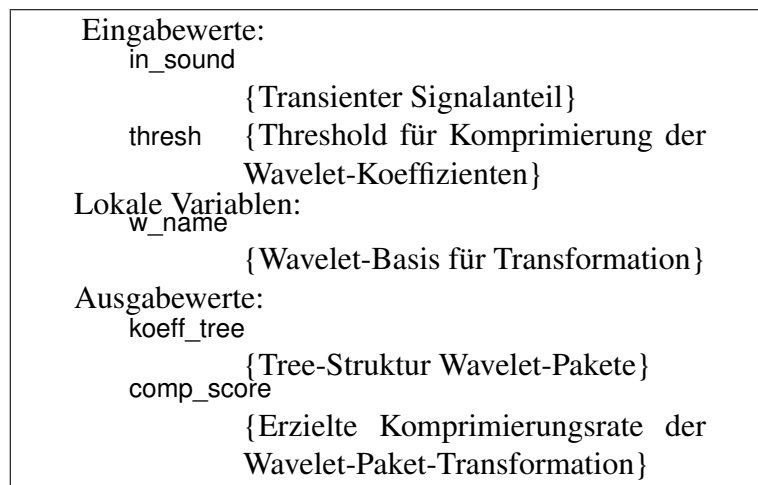
Struktogramm 4.28: Programmablauf zu *analysis\_features\_trans\_pack.m*

Eingabewerte:	
in_nonton	{Nontonaler Signalanteil}
fs	{Samplingfrequenz des Signals}
lokale Variablen:	
fg	{Grenzfrequenz des Glättungs-Tiefpasses}
thresh	{Threshold für Komprimierung der Wavelet-Koeffizienten}
Ausgabewerte:	
trans_feat	{Struktur mit Parametern für Transientenmodellierung}

Struktogramm 4.29: Variablen zu *analysis\_features\_trans\_pack.m*

Elemente von <i>trans_feat</i>	
env_para	{Stützpunkte der APR-Hüllkurve}
env	{Hüllkurve nontonaler Signalanteil}
tree	{Tree-Struktur der Wavelet Pakete}
score	{Erzielte Komprimierungsrate der Wavelet-Paket-Transformation}
energy	{Energie original Transiente}
fs	{Samplingfrequenz transienter Signalanteil}

Struktogramm 4.30: Die Struktur *trans\_feat\_pack*

Struktogramm 4.31: Programmablauf zu *get\_wavelet\_koeff\_pack.m*Struktogramm 4.32: Variablen zu *get\_wavelet\_koeff\_pack.m*



Saite 1 [%]	Saite 2 [%]	Saite 3 [%]	Saite 4 [%]	Saite 5 [%]	Saite 6 [%]
99,70	99,69	99,78	99,24	99,80	99,75

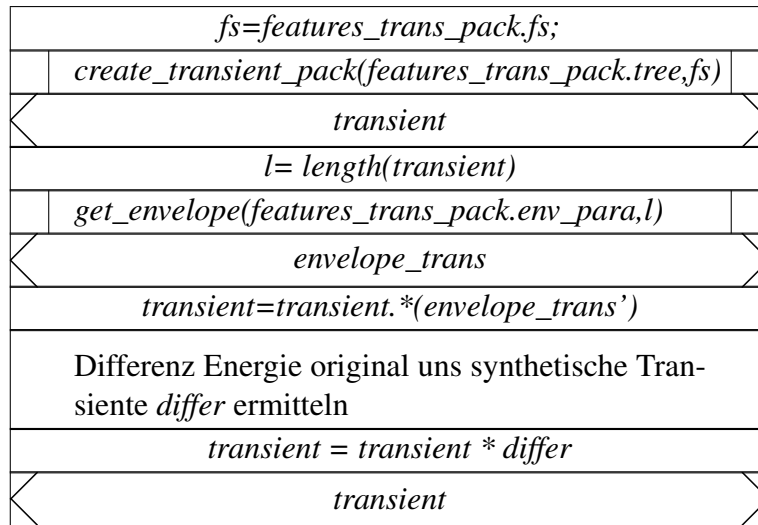
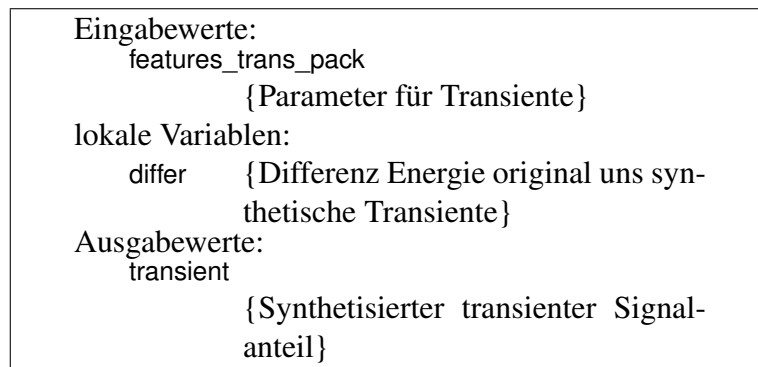
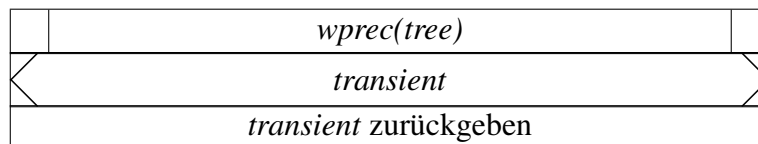
Tabelle 4.3.: Kompressionsraten Wavelet-Paket-Transformation

Matlab speichert die Baum-Struktur sowie die Wavelet-Koeffizienten dieser Paket-Transformation in einer Tree-Struktur ab, bei der es nicht möglich ist, mit geringem Aufwand die Koeffizienten für eine Codierung zu extrahieren. Daher wird in diesem Verfahren die gesamte Tree-Struktur zur Resynthese des transienten Signalanteils gespeichert und verwertet. Die Funktion *wpdencmp* übergibt jedoch zusätzlich einen Wert *PERF0*, der die Kompressionsrate in Prozent angibt. Dieser wird zusammen mit der Tree-Struktur in der Parameterstruktur *trans\_feat* gespeichert.

In Tabelle 4.3 sind die Kompressionsraten für die Transienten der sechs Gitarrenklänge (immer in ihrer ersten der vier vorhandenen Aufnahmen) angegeben. Um diese Werte mit denen in Tabelle 4.2 vergleichen zu können, wurde der Threshold auf 0.01 gesetzt.

Die Kompressionsraten sind denen der einfachen Wavelet-Transformation sehr ähnlich, gar in einem geringen Maße schlechter, die Anwendung von Wavelet-Paketen in dieser simplen Form birgt also in dieser Hinsicht keine direkten Vorteile. Dieses Ergebnis ist auch plausibel, haben die Wavelet-Pakete doch nach der in 2 besprochenen Theorie den Vorteil einer höheren Frequenzauflösung, was in Hinsicht auf die Daten-Komprimierung keine Vorteile hat. Inwieweit dieses Verfahren Vorteile auf die Qualität der Ergebnisse hat, lässt sich nur durch einen Hörtest feststellen, näheres dazu im Kapitel 4.3.4.

Für die Synthese der mittels Wavelet-Paketen modellierten transienten Signalanteile werden die entsprechenden Funktionen ebenfalls angepasst. Die Funktion *resynth\_trans\_pack.m* (Programmablauf im Struktogramm 4.33, Eingabe- sowie Ausgabewerte in Struktogramm 4.34) resynthetisiert den transienten Signalanteil nun ähnlich der Funktion *resynth\_trans.m*, mit dem Unterschied, dass die Transiente aus der Tree-Struktur, mit Hilfe der Funktion *create\_transient\_pack.m* (Programmablauf im Struktogramm 4.35, Eingabe- sowie Ausgabewerte in Struktogramm 4.36), synthetisiert. Hierzu wird die Matlab Funktion *wprec* verwendet, welche aus einer gegebenen Tree-Struktur das ursprüngliche Signal approximiert.

Struktogramm 4.33: Programmablauf zu *resynth\_trans\_pack.m*Struktogramm 4.34: Variablen zu *resynth\_trans\_pack.m*Struktogramm 4.35: Programmablauf zu *create\_transient\_pack.m*

Eingabewerte:	
tree	{Tree-Struktur der Wavelet-Pakete}
Ausgabewerte:	
transient	{Rekonstruierter transienter Signal- anteil}

Struktogramm 4.36: Variablen zu *create\_transient\_pack.m*

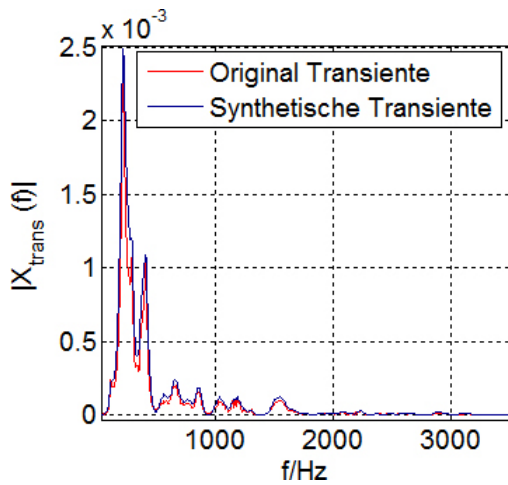


Abbildung 4.20.: Spektrum mit schneller FWT resynthetisierte Transiente

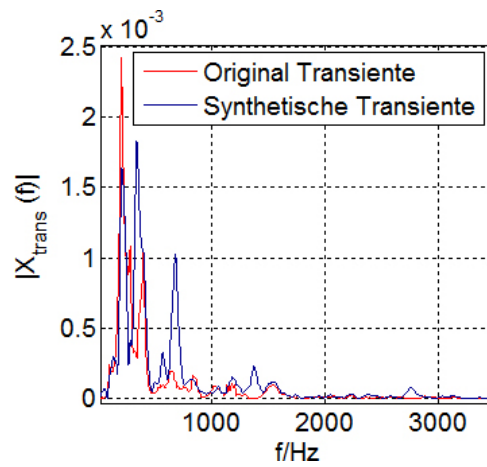


Abbildung 4.21.: Spektrum mit Wavelet-Paketen resynthetisierte Transiente

#### 4.3.4. Bewertung des entwickelten Verfahrens zur Transienten-Modellierung

In der qualitativen Beurteilung der synthetischen Transienten im Vergleich zwischen den Verfahren mit schneller Wavelet-Transformation und Wavelet-Paketen lässt sich bei gleicher Wavelet-Basis Daubechies-8 und gleichem Threshold 0.005 feststellen, dass die resultierenden synthetischen Anschlaggeräusche bei dem Verfahren mit Wavelet-Paketen, verglichen mit dem originalen, nontonalem Signal diesem in der Tonhöhe eher gleichen. Die der Wavelet-Paketen eigene bessere Frequenzauflösung macht sich hier also positiv bemerkbar. Hierzu werden in Abbildungen 4.20 und 4.21 die absoluten Frequenzgänge vom der originalen Transienten der ersten Gitarrensaite mit den beiden synthetisierten Transienten verglichen.

Deutlich ist zu erkennen, dass der Frequenzgang der mittels Wavelet-Paketen modellierten Transienten dem Frequenzspektrum der ursprünglichen Transienten deutlich gleicht, während jenes der mit einer einfacher Wavelet-Transformation modellierten abweicht. Es lässt sich daraus schließen, dass das Verfahren mit Wavelet Paketen im Bezug auf die Frequenzauflösung und damit auf die auditive Qualität einen Vorteil hat.

Im Bezug beider Verfahren auf jenes aus (vC09) fällt zunächst positiv die Beseitigung der dort auftauchenden, zeitlichen Verzögerung des Anschlaggeräusches auf, dieses setzt nun hörbar passend zum Klangbeginn ein. Ursache hierfür ist das verbesserte Verfahren zur Bestimmung der Hüllkurven und der veränderten Ermittlung des Anfangspunktes der APR-Hüllkurve *attS*.

Allgemein hat die Modellierung des transienten Signalanteils den Vorteil, mit dem Thresholding simultan den transienten Signalanteil von der Residue zu trennen (siehe dazu Kapitel 3.2). Auch klingen die Anschlaggeräusche echter und prägnanter. Es scheint also möglich, bei gleichen Komprimierungsraten eine bessere Qualität der Resynthese, verglichen mit einem sinusoidalen Verfahren, zu erreichen. Das hängt mit der unter Kapitel 3.2 besprochenen guten Darstellungsmöglichkeiten der Wavelet-Transformation von Signalen mit schnellen Veränderungen im Zeitbereich zusammen.

In den Abbildungen 4.22 und 4.23 sind die synthetischen, transienten Signalanteile aus dem Verfahren mit Sinusoiden aus (vC09) und aus dem Verfahren dieser Arbeit (mit einer Wavelet-Transformation) für den Gitarrenton *hoffmannL hense sonor S1-VI Bb5-1* abgebildet, in den Abbildungen 4.24 und 4.25 für den Gitarrenton *hoffmannL hense sonor S6-VI Bb3-1*).

Erkennbar sind mehrere Verbesserungen des Verfahrens. Zunächst setzten die neuen transienten Signalanteile ca. 10 – 20ms früher ein, womit die wegfallende Verzögerung des Anschlaggeräusches erklärt wäre. Auch erkennt man bei den neuen transienten Signalanteilen eine deutliche schärfere Steigung sowie einen schärferen Abfall der Energie, was den Vorteil der Wavelet-Transformation bei schnellen Veränderungen des Signalverlaufs über kurze Zeit unterlegt. Das Beispiel der Saite 6 zeigt zusätzlich noch offensichtliche Fehler im alten Verfahren, so entstanden hier Signalanteile vor dem Einsetzen des eigentlichen Anschlaggeräusches - das kommt im neuen Verfahren nun nicht mehr vor. Zuletzt ist das Energie-Niveau der neuen Transienten näher am dem der originalen Anschlaggeräusches, so dass diese im finalen, resynthetisierten Gitarrenklang prägnanter klingen.

Zu einem Vergleich der Modellierungs-Verfahren wurde unter Kapitel 5 zusätzlich ein Hörtest mit mehreren Testpersonen durchgeführt.

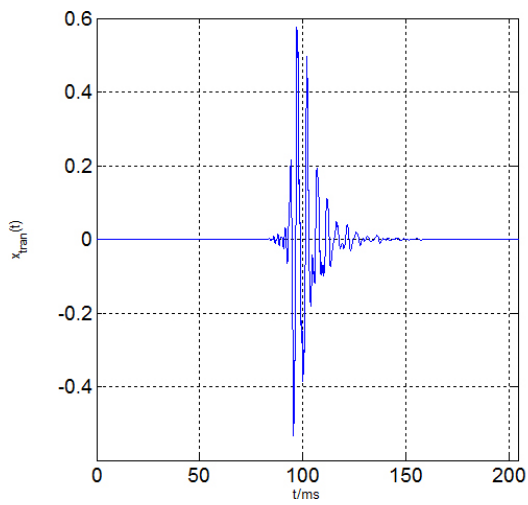


Abbildung 4.22.: Synthetische Transiente Saite 1 nach neuem Verfahren

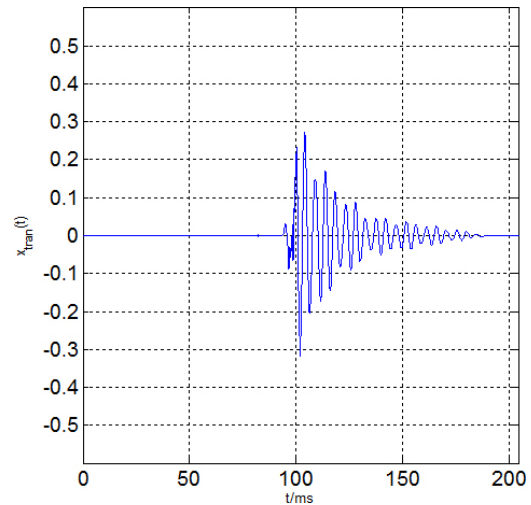


Abbildung 4.23.: Synthetische Transiente Saite 1 nach vonColer Verfahren

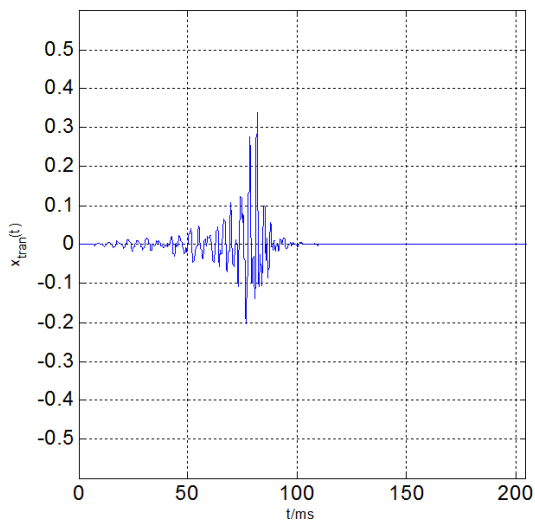


Abbildung 4.24.: Synthetische Transiente Saite 6 nach neuem Verfahren

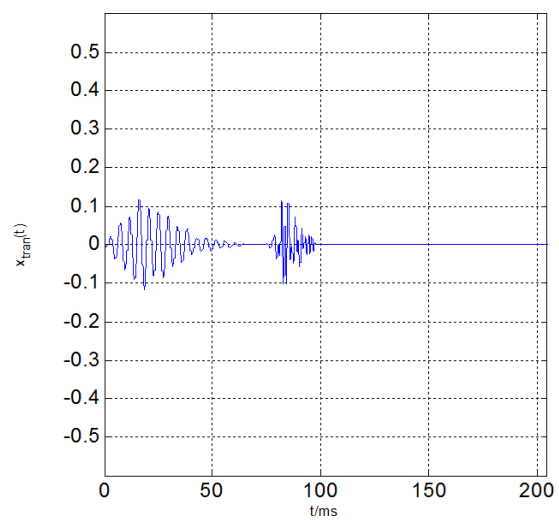


Abbildung 4.25.: Synthetische Transiente Saite 6 nach vonColer Verfahren

## 5. Hörtest zur Evaluation der Transienten-Synthese

Zur Evaluation der Ergebnisse des entwickelten Verfahrens zur Transienten-Synthese wurde ein Hörtest durchgeführt, um die Qualität des Verfahrens bewerten zu können, auch im Vergleich zum Verfahren aus (vC09). Der Test besteht aus zwei Teilen:

Im ersten Teil soll die Qualität der in dieser Arbeit resynthetisierten Transienten mit denen im Verfahren aus (vC09) verglichen werden. Zu diesem Zweck hören die Teilnehmer zu jeweils jeder Aufzeichnung eines Gitarrenklanges die originale Aufzeichnung und die jeweils resynthetisierten Gitarrenklänge aus beiden Verfahren, wobei ihnen dabei keine Information vorliegt, welcher synthetischer Klang aus welchem Verfahren stammt. Sie sollen für jeden Klang bestimmen, welcher der beiden synthetischen Klänge authentischer klingt.

In einem zweiten Teil soll aus dem hier entwickelten Verfahren die Qualität der Anschlaggeräusche, einmal mit schneller Wavelet-Transformation, einmal mit Wavelet-Paketen modelliert, verglichen werden. Zu diesem Zweck hören die Teilnehmer zu jeweils jedem Gitarrenklang eine resynthetisierte Version mit per schneller Wavelet-Transformation modellierten Transienten, und eine Version mit per Wavelet-Paketen modellierter Transienten.

### 5.1. Testdurchführung

Für den Hörtest wurden zehn Testpersonen mit den entsprechenden Sound-Dateien im .wav-Format und einem Fragebogen (siehe Anhang) ausgestattet. Ausgewählt wurden zu diesem Zweck folgende zehn Aufzeichnungen aus der in Kapitel 1 angesprochenen Datenbank der HAW-Hamburg:

Ausgesucht wurden sechs Testpersonen, die über Erfahrung mit dem Klang und dem Spiel von Konzertgitarren verfügen. Der Hörtest selber wurde von jeder Person privat unter den persönlich möglichst besten Testbedingungen durchgeführt. Die Soundfiles wurden mit einer Samplerate von  $44,1\text{kHz}$  und einer Quantisierung von  $24\text{bit}$  zur Verfügung gestellt.

Für den ersten Teil hat jede Testperson zehn Ordner mit jeweils der Original-Aufzeichnung und zwei unmarkierten, synthetischen Klängen (von Coler Verfahren und Verfahren aus dieser Arbeit) zur Verfügung gestellt bekommen, und sollte angeben, welcher synthetische

Aufzeichnung Nr.	Soundfile aus Datenbank
1	hoffmannL hense sonor S1-VI Bb5-2
2	hoffmannL hense sonor S1-VI Bb5-3
3	hoffmannL hense sonor S2-VI F5-2
4	hoffmannL hense sonor S3-VI Db5-2
5	hoffmannL hense sonor S3-VI Db5-3
6	hoffmannL hense sonor S4-VI Ab4-2
7	hoffmannL hense sonor S4-VI Ab4-3
8	hoffmannL hense sonor S5-VI Eb4-2
9	hoffmannL hense sonor S5-VI Eb4-3
10	hoffmannL hense sonor S6-VI Bb3-4

Tabelle 5.1.: Zuordnung der Testfiles zur Gitarrenklang-Datenbank

Aufzeichnung Nr.	vonColer-Verfahren besser [%]	Wavelet-Verfahren besser [%]
1	0	100
2	50	50
3	83	17
4	17	83
5	50	50
6	17	83
7	0	100
8	33	67
9	17	83
10	0	100

Tabelle 5.2.: Testergebnisse des Hörtest Teil 1

Klang authentischer klingt. Für den zweiten Teil hat jede Testperson zehn Ordner mit jeweils zwei unmarkierten, synthetischen Klängen (mit Schneller-Wavelet-Transformations(FWT)-Transienten und Wavelet-Paketen-Transienten) erhalten und sollte jeweils bestimmen, bei welchem Klang das Anschlaggeräusch authentischer wirkt.

## 5.2. Testergebnis und Bewertung

Im ersten Teil wurden die zehn verschiedenen synthetischen Gitarrenklänge prozentual je wie folgt als besser bewertet:

Deutlich erkennbar ist ein übermäßiges Empfinden, dass die synthetischen Klänge des in dieser Arbeit entwickelten Verfahrens mit der Transienten-Modellierung durch Wavelet-



Aufzeichnung Nr.	FWT-Transiente besser [%]	Wavelet-Pakete-Transiente besser [%]
1	50	50
2	33	67
3	50	50
4	50	50
5	33	67
6	50	50
7	33	67
8	50	50
9	33	67
10	33	67

Tabelle 5.3.: Testergebnisse des Hörtest Teil 2

Transformation authentischer klingen, lediglich bei dem Gitarrenklang 3 gibt es eine Abweichung zugunsten des vonColer-Verfahrens zu beobachten. Hierfür lässt sich jedoch keine deutlich erkennbare, objektive Begründung finden - dies könnte auf zufälliges, individuelles Hörempfinden zurückzuführen zu sein, eventuell würden sich bei einer größeren Testgruppe auch hier die Werte anpassen.

Gesondert fällt auf, das gerade bei den tieferen Saiten die Ergebnisse besonders deutlich ausfallen, gerade hier scheint das vonColer-Verfahren deutliche Schwächen aufzuweisen, die mit dem hier entwickelten Verfahren ausgebessert werden konnten.

Im zweiten Teil wurden die zehn verschiedenen synthetischen Gitarrenklänge, hinsichtlich des Anschlaggeräusches, prozentual je wie folgt als besser bewertet:

Dieses Ergebnis zeigt keine deutlichen Unterschiede der Qualität beider synthetischen Anschlaggeräusche auf. Zar kann man einen leichten Überschwung zugunsten der Wavelet-Pakete-Transienten erkennen, jedoch wurde im Verlauf des Hörtest von mehreren Versuchsteilnehmern angemerkt, dass sie kaum Unterschiede hören können und sich teilweise mit einer unbegründeten, zufälligen Entscheidung behelfen mussten.

# 6. Fazit und Ausblick

## 6.1. Erreichte Ziele

In dieser Arbeit wurde erfolgreich ein Analyse- und Synthese-Verfahren für Gitarrenklänge implementiert, welches das Ausgangssignal in einen tonalen und nontonalen Anteil zerteilt, den transienten Anteil aus dem nontonalen extrahiert und diese separat modelliert. Ausgangspunkt für dieses Vorhaben war das Verfahren aus (vC09), welches in dieser Arbeit verbessert werden sollte. Hauptaugenmerk lag dabei auf der Modellierung des transienten Anteils - in diesem Zusammenhang wurden insbesondere die Wavelet-Transformation und ihre Möglichkeiten untersucht.

Es ist im Rahmen dieser Arbeit gelungen, zunächst einen Überblick über die Theorie der Wavelet-Transformation zu schaffen, insbesondere hinsichtlich ihrer Möglichkeiten in Bezug auf Transienten-Modellierung. Es ist gelungen, ein Analyse- und Synthese Verfahren zu entwickeln, das den transienten Anteil des Gitarrenklangs mit Hilfe der Wavelet-Transformation vom nontonalen Anteil extrahiert und in einem Parameter-Satz modelliert. Beibehalten wurden dabei die Analyse und Parametrisierung der APR-Hüllkurve für die Transienten, welche ebenfalls verbessert werden konnte. Es konnten dabei zwei verschiedene Verfahren vorgestellt werden - eines nutzt eine einfache, diskrete Wavelet-Transformation eines Wavelet-Pakete zur Modellierung der transienten Anteile.

Mit diesem entwickelten Verfahren konnten Anschlaggeräusche modelliert werden, die prägnanter und schärfer klingen und so authentischer wirken als jene aus der bandweisen Hüllkurven und Wellenformen- Modellierung aus (vC09). Auch ist es im Zuge dessen gelungen, die Anschlaggeräusche im resynthetisierten Gesamtklang zeitlich richtig zu positionieren - sie wirken nicht mehr verspätet zum Beginn des tonalen Anteils. Auch ist es nun möglich, die Anschlaggeräusche im Parametersatz anhand der APR-Hüllkurve nun effektiver zu modifizieren, da diese nun für die gesamte Transiente vorliegen, da dank der Zeitauflösung der Wavelet-Transformation keine Unterteilung in Bänder mehr nötig ist.

Ein Hörtest konnte die verbesserte Qualität des Verfahrens untermauern. Auch zeigt sich dort in einer zweiten Untersuchung, dass die hier mit Wavelet-Paketen modellierte Transienten noch etwas authentischer wirken.

## 6.2. Ausblick

Dieses Verfahren bietet weiterhin weitreichende Möglichkeiten zur Verbesserung. Bereits angesprochen wurden Möglichkeiten zur Verbesserung der tonalen Modellierung, hier kann eine Berücksichtigung der Phasen für authentischeren Klang sorgen, ebenso die Entwicklung eines Modellierungs-Verfahrens unter Verwendung der diskreten Cosinus Transformation.

Das hier vorgestellte, neue Verfahren zur Transienten-Modellierung bietet ebenfalls weitreichende, weiterführende Möglichkeiten. So könnte gerade im Bereich der Wavelet-Pakete erforscht werden, wie es dort mit angepassten Baum-Strukturen möglich sein könnte, spezifische Modelle für Transienten, im speziellen Anschlaggeräuschen von Gitarren zu entwickeln. Hier könnte der Vorteil der Wavelet-Transformation, nämlich in Bezug auf das Verhältnis von Frequenz- und Zeitauflösung variabel zu sein, genauer auf die Anwendung angepasst werden. Auch lohnt sich eine Untersuchung der Auswirkung bestimmter Koeffizienten auf den Klang der Transienten, um hier im Parametersatz der Analyse mehr Möglichkeiten zur direkten Klanggestaltung zu erhalten.

Weiterhin liefert das in diesem Verfahren noch einfach gelöste Problem der Entropie-Codierung Möglichkeiten zur Verbesserung, hier könnten effektivere Mittel zur Codierung entwickelt werden.

Eine grosse Schwachstelle dieses Verfahrens ist nach wie vor die Separation von tonalem und nontonalem Anteil des Ausgangssignals. Hier werden in Kapitel 3 bereits ein paar Möglichkeiten angesprochen, wie es mit Hilfe der Wavelet-Transformation und weiterführender, mathematischer Werkzeuge möglich sein könnte, ein Verfahren zu implementieren, das die transienten Signalanteile direkt aus dem Ausgangssignal extrahiert und die Modellierung derer somit unabhängig von einer vorherigen, in diesem Fall groben Eliminierung des tonalen Signalanteils macht. Auch hier lohnen sich im Speziellen weiterführende Untersuchungen zu den Möglichkeiten angepasster Baum-Strukturen bei Wavelet-Paketen.

Darüber hinaus könnten weiterführende Überlegungen dahin gehen, das Verfahren zur Transienten-Modellierung dahin zu entwickeln, unabhängig vom Charakter des Klanges (z.B. Instrument) zu funktionieren oder noch weiterführend auch auf Klangfolgen, also z.B. Tonfolgen eines einzelnen oder mehreren Instrumenten, anwendbar zu sein.

# Literaturverzeichnis

- [And04] Andersen, Jensen Kristoffer, Tue H. *Importance and Representation of Phase in the Sinusoidal Model*. *JAES*, volume 52, 2004.
- [Bae05] Baeni, W. *Wavelets*, 2005. Einführung und Beschreibung Wavelet-Transformaton.
- [BMV10] Bruni, V., Marconi, S. and Vitulano, D. *Time-Scale Atoms Chains for Transients Detection in Audio Signals*. *Audio, Speech, and Language Processing, IEEE Transactions on*, volume 18(3):pages 420–433, 2010. ISSN 1558-7916. doi: 10.1109/TASL.2009.2032623.
- [Dau92] Daubechies, Ingrid. *Ten Lectures on Wavelets*, 1992. CBMS-NSF Regional Conference Series in Applied Mathematics.
- [Dau01] Daudet, Torresani, Molla. *Transient detection and encoding using wavelet coefficient trees*. *GRETSI - Actes de Colloques*, 2001.
- [Dau05] Daudet, L. *A review on techniques for the extraction of transients in musical signals*. *Comput. Music Modeling and Retrieval*, volume pp:pages 219–232, 2005.
- [Dau06] Daudet. *Sparse and structured decompositions of signals with the molecular matching pursuit*. *IEEE Transactions on Speech and Audio Processing*, 2006.
- [DD01] Duxbury, Chris and Davies, Sandler Mark, Mike. *Separation of transient information in musical audio using multiresolution analysis techniques*. *Proceedings of the COST G-6 Conference on Digital Audio Effects*, (5):pages 6–9, 2001.
- [Dos08] Dosenbach, Kerstin. *Klassifikation von Audiosignalen mit Support Vector Machines*. *DAFX*, 2008.
- [DT02] Daudet, L. and Torrésani, B. *Hybrid representations for audiophonic signal encoding*. *Signal Process.*, volume 82:pages 1595–1617, 2002.
- [FPE<sup>+</sup>06] Fragoulis, Dimitrios, Papaodysseus, Constantin et al.. *Automated Classification of Piano-Guitar Notes*. *IEEE Trans. on Audio, Speech, and Language Processing*, volume 14(3), 2006.
- [G.94] G., Davis. *Adaptive Nonlinear Approximations*, 1994.

- [Har80] Haralick, R. *Edge and Region Analysis for Digital Image Data. Computer Graphics and Image Processing*, volume 12:pages 60–73, 1980.
- [KH96] K.N. Hamdy, A.H. Tewfi, M. Ali. *Low bit rate high quality audio coding with combined harmonic and wavelet representations. Icassp: Acoustics, Speech, and Signal Processing*, volume 2:pages 1045–1048, 1996.
- [Mal93] Mallat, Zhang Z., S. *Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing*, volume 41:pages 3397–3415, 1993.
- [MLH92] Mallat, Stephane and Liang Hwang, Wen. *Singularity Detection And Processing With Wavelets. IEEE Transactions On Information Theory*, volume 38(2), 1992.
- [Nav10] Naveen, D. *Implementation of Psychoacoustic model in Audio Compression using Munich and Gammachirp Wavelets. International Journal of Engineering Science*, volume 2(5):pages 1066–1072, 2010.
- [Nis07] Nishan, Canagarajah C. *Musical instrument recognition with wavelet envelopes*, 2007.
- [Nsa07] Nsabimana, Zölzer U, F. *Transient Encoding Of audio Signals Using Dyadic Approximations. DAFx-07*, volume pp, 2007.
- [NSJ84] N. S. Jayant, P. Noll. *Digital coding of waveforms. Prentice Hall*, 1984.
- [PV01] P.P. Vaidyanathan, S. Akkrarakaran. *A review of the theory and applications of optimal sub-band and transform coders. Appl. Comp. Harm. Anal.*, volume 10:pages 254–289, 2001.
- [RC92] R.R. Coifman, M.V. Wickerhauser. *Entropybased algorithms for best basisselection. IEEE Trans. Information Theory*, volume 38:page 12411243, 1992.
- [Tac76] Tachi, Susumu. *Effect of Phase Relations on the Timbre of Harmonic Tones. Bulletin of the Mechanical Engineering Laboratory*, (25), 1976.
- [TP00] T. Painter, A. Spanias. *Perceptual coding of digital audio. IEEE*, volume 88:pages 457–515, 2000.
- [VC04] Vera-Candeas, Rosa-Zurera Martinez-Muñoz Lopez-Ferreras, Ruiz-Reyes. *Transient Modeling by Matching Pursuits With a Wavelet Dictionary for Parametric Audio Coding. Signal Processing*, volume 11, 2004.
- [VC07] Vera-Candeas, Ruiz-Reyes. *Optimizing a wavelet-based dictionary for transient modelling With application to parametric audio coding*, 2007.
- [vC09] von Coler, Hendrik. *Ein Verfahren zur Synthese von Gitarrenklängen mit separater Modellierung der tonalen und der transienten Anteile*, 2009.

- 
- [vL03] von Lee, Peter. *Wavelet Filter Banks in Perceptual Audio Coding*. Masterthesis, University of Waterloo, 2003.
- [Voi07] Voigt, Julia. *languntersuchung an Konzertgitarren: Korrelation zwischen physikalischen Signalmerkmalen und psychoakustischen Attributen*, 2007.
- [VR06] Vuletich, Juan M and Ruedin, Ana M C. *Wavelet Bases for Compact Music Representation*. *ReCALL*, 2006.
- [VT97] Verma T., Meng-T, Levine S. *Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals*. *Proc. of the International Computer Music Conference*, 1997.
- [Wic94] Wickerhauser, M. V. *Adapted wavelet analysis from theory to software*, 1994.

# A. Inhalt der Daten-CD

Die beigefügte Daten-CD hat folgenden Inhalt:

- Matlabfiles: Die Matlab-Dateien zur Implementierung dieses Verfahrens. Der Ordner *Verfahren1* beinhaltet die Files zum Verfahren mit Transienten Modellierung durch die einfache Wavelet-Transformation, der Ordner *Verfahren2* die Files zum Verfahren mit Transienten Modellierung durch Wavelet-Pakete. Im Ordner *Extract Transient* befindet sich die Matlab-Funktion *extracfromnonton\_transient* aus Kapitel 3.2.
- Latexfiles: Hier befindet sich der LaTeX-Code zur Ausarbeitung dieser Arbeit.
- Klangfiles: Hier befinden sich die Sound-Dateien und Parameter-Sätze der Analyse zu den entwickelten Verfahren, der Ordner *Klangfiles1* enthält die Dateien zum oben erwähnten *Verfahren1*, der Ordner *Klangfiles2* die Dateien zum *Verfahren2*.
- Sowie diese Ausarbeitung im pdf-Format.

## **B. Testbogen zum Hörtest**



# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §25(4) 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, 28. Februar 2011

Ort, Datum

Unterschrift