



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Dirk Jagdmann

Analyse und Resynthese des Gitarrenklangs

*Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science*

Dirk Jagdmann

Thema der Diplomarbeit

Analyse und Resynthese des Gitarrenklangs

Stichworte

Gitarre, Klang, Analyse, Synthese, DFT, DCT, Regression

Kurzzusammenfassung

Der Mensch kann unterschiedliche Musikinstrumente problemlos anhand ihres Klangs identifizieren. Auch unterschiedliche Modelle des gleichen Instrumenttyps lassen sich von einem geübten Hörer unterscheiden. Dennoch ist die genaue Vorgang, welcher den Klangeindruck des Ohres in Gehirn analysiert und zuordnet, bislang nicht bekannt.

Diese Diplomarbeit beschäftigt sich mit der Analyse und optionalen Resynthese von Gitarrenklängen. Zunächst wird diskutiert wie der Klang durch eine schwingende Saite entsteht. Die digitale Aufnahme einer realen Gitarre wird dann von Algorithmen in eine Reihe von Parametern transformiert. Diese Parameter stellen das Charakteristische des Gitarrenklangs und des aufgenommenen Gitarrentyps und -modells dar und sollten zwecks handhabbarkeit aus möglichst wenigen Zahlen bestehen.

Zur Kontrolle das die ermittelten Parameter auch tatsächlich die spezifischen Besonderheiten der Aufnahme widerspiegeln wird parallel zur Analyse ein Resyntheseverfahren entwickelt, mit dem sich Klänge reproduzieren lassen. Durch eine akustische Kontrolle (vom Menschen) läßt sich prüfen ob der reproduzierte Klang eine Ähnlichkeit zur originalen Aufnahme besitzt, was anzeigt, dass die Parameter den Klang treffend beschreiben.

Dirk Jagdmann

Title of the paper

Analysis and Resynthesis of Guitarsounds

Keywords

Guitar, Sound, Analysis, Synthesis, DFT, DCT, Regression

Abstract

Distinct musical instruments can be easily classified by a human listener. Even different models of the same instrument type can be distinguished by the trained listener. But the exact operations by which the human brain analyses and recognizes sound are unknown of today.

This diploma thesis develops algorithms to analyse and resynthesize guitar sounds. At first the sound creation of the guitar by means of a swingung string are discussed. Then an algorithm is developed to transform a digital record of guitar sound into a set of parameters. These parameters ideally should represent the *characteristic nature* of the sound and should consist of as few numbers possible.

To prove that the calculated parameters indeed contain relevant information to describe the original sound, a resynthesis algorithm is developed which will produce a new digital sound from the calculated parameters. If the synthesized sound resembles the original sound the analysis parameters contain the relevant information.

Inhaltsverzeichnis

Zusammenfassung	2
Konventionen, CD, Webseite, Danksagung	5
1 Problemstellung	6
1.1 Ähnliche Arbeiten	9
1.2 Zusammenfassung	10
2 Analyse	11
2.1 Konstruktion eines Klangs aus Obertönen	11
2.2 Was ist in einer Aufnahme enthalten?	12
2.3 Detaillierte Analyse der Attackphase	15
2.4 Detaillierte Analyse der Sustainphase	16
2.5 Resynthese	20
2.6 Zusammenfassung	20
3 Design	22
3.1 Extraktion von Attack- und Sustainphase	22
3.2 Bestimmung von Grundfrequenz und Obertönen	23
3.2.1 Parameter der DFT	24
3.2.2 Bestimmung der Grundfrequenz	28
3.3 Bestimmung der Obertöne	29
3.4 Berechnung von Frequenz-, Amplituden- und Phasenmatrix	31
3.5 Erste Version der Synthese	33
3.6 Optimierung der Analyse	36
3.6.1 Phasenmatrix	36
3.6.2 Frequenzmatrix	37
3.6.3 Amplitudenmatrix	39
3.7 Variable Syntheselänge	42
3.8 Arrangement der Synthesedaten	43
3.9 Analyse der Attackphase	44
3.10 Zusammenfassung	45
4 Realisierung	46
4.1 Verwendete Hard- und Software	46
4.2 Besonderheiten bei der Umsetzung des Designs	46
4.3 Besonderheiten bei einzelnen Algorithmen	48
4.4 Aufbau des Source Codes	48

Inhaltsverzeichnis

5	Ausblick	50
	Literaturverzeichnis	53
A	Autokorrelation	55
A.1	Berechnung der Genauigkeiten von 3.2.2	56
B	Berechnung neuer Zeilen mittels exponentieller Regression	57
C	Glossar	59
D	Trivia	63
E	History	64

Konventionen

Zahlen werden in der englischen Schreibweise dargestellt mit einem Punkt als Dezimaltrenner. Enthält die Zahl einen Dezimalpunkt, dann ist die Zahl auf 2 bis 4 Stellen gerundet, da sie i. d. R. das Ergebnis einer numerischen Berechnung ist. Solche Ergebnisse enthalten die für 32bit IEEE floating point Zahlen typischen Fehler.

Der Begriff Kosinus wird in Rest dieser Arbeit mit C als Cosinus benutzt, da dadurch die Beziehung zur auch im deutschen gebräuchlichen Abkürzung DCT besser hergestellt wird.

Inhalt der CD

Die Begleit CD ist im Format CD-Extra erstellt worden. Die ersten 13 Tracks enthalten Beispielklänge, welche im Text jeweils mit einem CD Icon im Rand erwähnt werden. Die Nummer des Icons entspricht der Nummer des Tracks auf CD. Die CD enthält weiterhin einen Daten-Track, auf welchem sich der Linux Source Code befindet. Zusätzlich sind die Sources dieses Textes im \LaTeX -Format ebenfalls enthalten.

Webseite

Unter <http://cubic.org/~doj/diplom> kann der Sourcecode dieser Arbeit bezogen werden, die Beispielklänge, sowie zukünftige und korrigierte Versionen dieses Textes.

Danksagung

Ich möchte mich bei Prof. Wolfgang Fohl für die Betreuung dieser Diplomarbeit bedanken. Weiterhin beim Peer-Review-Team Björn Buske, Lukas Grunwald und Nils Pipenbrinck für hilfreiche Anmerkungen bei der Erstellung des Textes bedanken. Zum Schluss bei Michael Stickel für Hilfe beim Druck der Pflichtexemplare.

1 Problemstellung

Von den fünf Sinnen des Menschen¹ ist der Hörsinn einer der leistungsfähigsten und braucht den Vergleich mit anderen Lebewesen aus dem Tierreich nicht zu scheuen. Während unser Sehsinn nur durchschnittlich entwickelt ist und von vielen Tieren weit übertroffen wird, man denke an den Adlerblick, und auch der Geruchs- und Geschmacksinn nicht über das Mittelmaß hinauskommen, gehört der Hörsinn zu unseren Stärken. Auch wenn beim *modernen Durchschnittsmenschen* der Sehsinn stark gegenüber dem Hören dominiert und deswegen der Hörsinn oftmals schlecht trainiert ist, hat der Mensch doch ein großes Potential *in seinen Ohren*. Bei Berufen wie Musikern oder Akustikern ist die Fähigkeit unterschiedlichste Töne und Geräusche genau zu identifizieren sehr gut ausgebildet. Für blinde Menschen sind die Ohren ein sehr guter Ersatz für die Sehkraft und viele Dinge, die ein nicht Blinder einfach *sieht* kann ein Blinder genauso zuverlässig hören.

Die Leistung des menschlichen Hörsinns kann man durch folgende Größen beschreiben. Es können Frequenzen von ≈ 16 Hz bis ≈ 20000 Hz erkannt werden. Dies entspricht einem Frequenzbereich der 4 Zehnerpotenzen umfasst. Unser Gehör kann in diesem Bereich auch kleine Unterschiede in der Frequenz von zwei Tönen (weniger als 1 Hz) erkennen. Des weiteren kann das Gehör auch eine große Bandbreite von Lautstärken unterscheiden. Die untere Grenze, ab welcher ein Ton nicht mehr wahrgenommen werden kann, ist Frequenzabhängig und ergibt bei einem normalhörigen Menschen eine parabelähnliche Kurve. Die obere Kurve *Schmerzgrenze* zeigt, ab welcher Amplitude das Schmerzempfinden einsetzt. Die Punkt an welchem sich beide Kurven treffen geben die höchste und niedrigste wahrnehmbare Frequenz an.² Die *Abbildung 1.1* zeigt in welchem frequenzabhängigem Lautstärkebereich das Gehör Schwingungen wahrnimmt.³ Die Fähigkeit eine Schallschwingung als Musik oder Sprache zu erkennen ist jedoch in einem deutlich kleineren Bereich möglich, als das generelle Erkennen eines akustischen Signals.

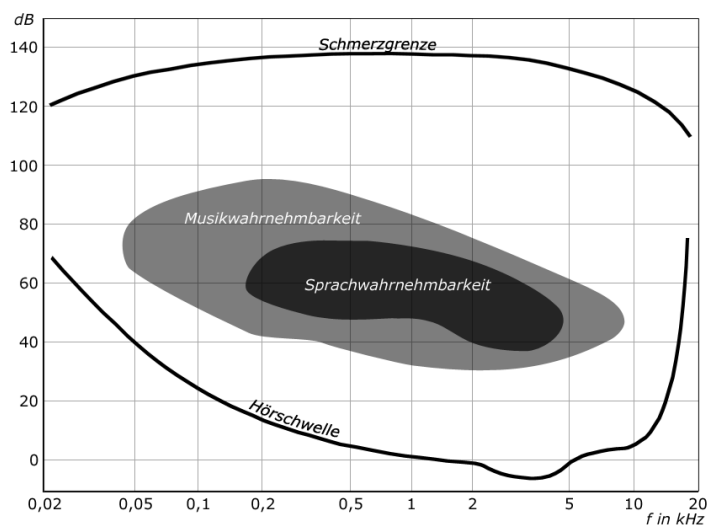
Während die im letzten Absatz beschriebenen Möglichkeiten Merkmale unseres Ohres sind, die im wesentlichen durch den Aufbau des menschlichen Ohres begrenzt sind, ist die akustische Wahrnehmung eines Klangereignisses im Gehirn noch deutlich leistungsfähiger. Obwohl z.B. eine Trompete, eine Geige und ein Klavier einen deutlich unterschiedlichen Klang haben, sich also ihr Frequenzspektrum unterscheidet, fällt es dem Menschen überhaupt nicht schwer zu sagen, ob diese drei Instrumente den gleichen Ton bzw. Frequenz spielen oder ob sie gleich laut spielen. Und auch wenn alle drei zusammen spielen fällt es uns nicht schwer die einzelnen Töne der Instrumente heraus zu hören.

¹Sehen, Hören, Riechen, Schmecken, Fühlen

²Eine ausführliche Beschreibung der Hörleistung findet sich in [Vog99].

³Die Skala auf der Ordinate ist in dB SPL angegeben, welche ein absolutes Maß für die Lautstärke eines Tons ist. Näheres dazu in [Fou04].

1 Problemstellung



Diese Abbildung stammt aus [Wikipedia:Hörschwelle](#)

Abbildung 1.1: Hörfläche des (gesunden) Menschen

Einem geübten Hörer ist es ein leichtes, jedes Instrument (bzw. Instrumentengruppe) in einem großen Orchester präzise zu hören.

Wie das Gehirn den vom Ohr wahrgenommenen Klang quasi in seine Einzelteile zerlegen kann, ist weiterhin nicht genau geklärt und Gegenstand aktiver Forschung. Für eine Maschine bzw. den Computer ist solch eine Aufgabe eine große Herausforderung. Mit einem Teilaspekt der maschinellen Wahrnehmung von Instrumenten beschäftigt sich diese Arbeit.

Ziel ist es, eine Methode zu entwickeln, welche den Klang einer Gitarre beschreibt bzw. charakterisiert. Die Gitarre ist insofern ein interessantes Instrument, da es sie in einer Vielzahl von unterschiedlichen Typen gibt, welche sich stark im Klang unterscheiden aber dennoch alle wiederum *typisch* nach Gitarre klingen. Man unterteilt die Bauformen zunächst in drei Typen: akustische, semi-akustische und elektrische Gitarren.

Akustische Gitarren sind die bekannteste Form. Charakteristisch ist der große, hohle Korpus, welcher durch seine Bauform die Schwingungen der Saiten verstärkt und an die Luft abgibt. In der Regel sind akustische Gitarren aus Holz gebaut, es gibt jedoch auch Modelle aus Metall. Die Saiten können aus einem beliebigen Material gefertigt sein.

Elektrische Gitarren haben i. d. R. nur ein Holzbrett als Korpus und geben deshalb die Schwingungen der Saiten nur schlecht an die Luft ab. Sie sind deutlich leiser als akustische Gitarren. Sie besitzen jedoch sog. Pickups, welche die Schwingungen der Saiten in einen schwachen Wechselstrom umsetzen, der über einen Verstärker wiedergegeben werden kann.

Semiakustische Gitarren sind eine Mischung aus den beiden zuvor beschriebenen Typen. Sie haben einen hohlen Korpus, welcher jedoch nicht so tief ist wie bei einer akustischen Gitarre. In [Abbildung 1.2](#) sind deutlich die beiden f -förmigen Schalllöcher zu erkennen,

1 Problemstellung



Abbildung 1.2: Beispiele für unterschiedliche Gitarrentypen (QUELLE [Wik])

ähnlich denen einer Violine. Sie besitzen ebenfalls Pickups. Normalerweise wird eine solche Gitarre auch mit Hilfe der Pickups als elektrische Gitarre genutzt. Durch den schwingenden Korpus besitzt so eine Gitarre jedoch einen anderen Klang als eine reine akustische Gitarre.

Von allen drei Typen gibt es nun eine große Auswahl an unterschiedlichen Modellen, die sich in ihrer Größe und den zum Bau verwendeten Materialien unterscheiden. Alle diese Faktoren haben einen Einfluss auf den Klang. Einen nicht minder großen Anteil am Klang hat auch der Spieler. Vergleicht man zwei Spieler und zwei Gitarren des gleichen Typs, lässt dabei jeden Spieler auf jeder Gitarre spielen, so ist der klangliche Unterschied zwischen beiden Spielern i. d. R. größer als der Unterschied zwischen den Gitarren. Der Mensch kann mit ein wenig Übung sehr zuverlässig die unterschiedlichen Typen oder auch Modelle und Spieler unterscheiden. Ein zu entwickelndes Computersystem soll nun ähnlich dem Menschen die unterschiedlichen Gitarren und Spieler unterscheiden können.

Der Computer soll also in die Lage versetzt werden, durch die Analyse einer Aufnahme Spieler und Gitarre zu bestimmen. Eine gleichwertige Fragestellung wäre, festzustellen ob zwei Aufnahmen von demselben Spieler/Gitarre stammen. Dies ist die *Analyse des Gitarrenklangs*.

Da Computer letztendlich nur Zahlen verarbeiten, ist das Ergebnis so einer Analyse zunächst eine Reihe von Zahlen, die wiederum vom Menschen interpretiert werden müssen. Um zu untersuchen, ob diese Zahlen auch tatsächlich das wesentliche Merkmal des Klangs beschreiben, sollte sich aus dem Ergebnis ein neuer Gitarrenklang errechnen lassen. Wenn der berechnete Klang dem ursprünglichen Klang möglichst ähnlich klingt, hat man eine Bestätigung, dass diese Zahlen tatsächlich ein für den Klang wesentliches Merkmal beinhalten. Dies ist die *Resynthese des Gitarrenklangs*.

Interessant ist die Frage, wie sich die Ergebnisse bei unterschiedlichen Gitarren und Spielern unterscheiden. Weiterhin ist es interessant wie sich der resynthetisierte Klang verändert, wenn man die Zahlen verändert, und ob z.B. beim Mittelwert von zwei unterschiedlichen Zahlenreihen zweier unterschiedlicher Gitarren der synthetisierte Klang wie eine Mischung aus beiden Gitarren klingt. Um die Zahlen weiterhin handhabbar zu machen, ist es wünschenswert, mit möglichst wenigen Zahlen zu arbeiten, welche trotzdem eine große Variation der Synthese erlauben.

1.1 Ähnliche Arbeiten

Der Großteil der Untersuchungen zum Klang der Gitarre versucht auf Basis eines physikalischen Modells einen Klang zu synthetisieren, der nicht unbedingt dem einer realen Gitarre entsprechen muß, sondern für den Hörer wie eine ideale oder durchschnittliche Gitarre klingen soll. Dazu wird die Gitarre i.d.R. als Kombination von drei schwingenden Systemen betrachtet: Saite, Korpus und die vom Korpus eingeschlossene Luft. Für die Saite wird ein Modell einer schwingenden Saite entwickelt. Beim Korpus sind vor allem die Schwingungen der Decke und des Bodens ausschlaggebend, welche als schwingende Membran angesehen werden. Die Schwingung der Luft im Hohlraum des Korpus wird durch die Schwingung eines *Helmholtz-Resonator*⁴ beschrieben. Die physikalischen Grundlagen des Helmholtz-Resonators beschreibt [FR98].

Die Saitenschwingung kann natürlich durch numerische Integration der Wellengleichung 2.2 berechnet werden. Das gleiche Ergebnis läßt sich aber auch mittels des *Karplus-Strong* Algorithmus ([KS83]) oder seiner Weiterentwicklung, dem *Digital Waveguide* Algorithmus ([Smi92]), erzielen.

Die Algorithmen zur physikalischen Simulation der Gitarre werden seit Anfang der 90'er Jahre hauptsächlich an der Technischen Universität Helsinki unter Matti Karjalainen und Vesa Välimäki weiterentwickelt. [KVT98] und [KVJ93] beschreiben, wie man Saite, Korpus und Hohlraum zu einem Gesamtmodell kombinieren kann, um Klänge zu synthetisieren.

Ein Algorithmus, welcher den Gitarrenklang durch physikalische Simulation berechnet und dabei das Ergebnis möglichst wie eine reale Gitarre klingt, ist jedoch noch nicht entwickelt worden. Die durch physikalische Simulation gewonnenen Klänge klingen zwar denen einer echten Gitarre je nach Aufwand erstaunlich ähnlich, es ist jedoch nicht möglich den Klang einer bestimmten Gitarre gezielt zu reproduzieren.

[KR02] diskutieren die Analyse von Klängen mittels DFT und die Rekonstruktion von Audio Daten durch die Analyseergebnisse, wobei Algorithmen zur Rekonstruktion vorgestellt werden, die Defekte in der Aufnahme (Lücken oder Kratzer bei Aufnahme von Schallplatte) beheben. Einen ähnlichen Ansatz wie ich benutzen [BR02] um Klänge mittels exponentiell gedämpften Sinusoiden zu synthetisieren.

⁴Ein Helmholtz-Resonator ist ein Körper, welcher einer verallgemeinerten Flasche entspricht. Er besitzt einen großen Hohlkörper an dessen Öffnung ein hohler Zylinder sitzt. Durch Variation der Länge und des Durchmessers des Zylinders, sowie des Volumens des Hohlkörpers läßt sich die Resonanzfrequenz des Helmholtz-Resonators verändern.

1 Problemstellung

Speziell mit der Analyse und Resynthese von akustischen Gitarren beschäftigt sich [Tol98]. [Ero01] beschreibt eine Methode, unterschiedliche Musikinstrumente zu erkennen, indem Aufnahmen analysiert werden. In [Jen99] wird ein Verfahren entwickelt um Klänge zu analysieren und klassifizieren.

Einen ähnlichen Ansatz wie diese Arbeit beschreibt [EVKL00] bei dem mittels Analyse einer Aufnahme die Parameter für eine physikalische Simulation einer klassischen Gitarre ermittelt werden sollen.

1.2 Zusammenfassung

Für den Menschen ist es leicht, unterschiedliche Musikinstrumente anhand ihres Klangs zu unterscheiden. Mehr noch, auch unterschiedliche Instrumente der selben Art können oftmals anhand ihres Klangs unterschieden werden. Auch wenn es leicht ist, den Unterschied zwischen verschiedenen Instrumenten zu bestimmen, bzw. zu sagen, dass zwei unterschiedliche Klänge von unterschiedlichen Instrumenten produziert wurden, so ist es doch nicht einfach, einen einzelnen Klang genau zu beschreiben. Oder anders ausgedrückt, zu beschreiben, was das Spezifische des Klangs z.B. einer Gitarre ist und wie sich der Klang der Gitarre von dem eines Klaviers unterscheidet.

Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, mit dem die Parameter eines Klangs beschrieben werden können. Diese Parameter werden computergestützt parametrisiert. Um die Plausibilität der Parameter zu prüfen wird weiterhin ein Verfahren zur Synthese entwickelt, welche mit den ermittelten Parametern neue, dem Original möglichst ähnliche Klänge erzeugen kann.

2 Analyse

Um ein Verfahren zur Analyse von Gitarrenklängen zu entwickeln, muss zunächst untersucht werden, woraus ein typischer Gitarrenklang besteht. Diese Untersuchung ergibt dann Anhaltspunkte, welche Eigenschaften im Klang vorhanden sind und wie diese Eigenschaften untersucht werden können. In diesem Kapitel wird untersucht, aus welchen Bestandteilen der Klang aufgebaut ist, und mit welchen mathematischen Methoden diese Bestandteile untersucht werden können. Es ergeben sich eine Menge von Parametern, welche die einzelnen Bestandteile beschreiben. Durch Elimination von nicht relevanten Parametern wird die Ergebnismenge schrittweise immer weiter verkleinert, bis der minimale Satz von Parametern gefunden ist, welcher den Klang beschreibt.

2.1 Konstruktion eines Klangs aus Obertönen

Der Klang einer Gitarre wird durch die Schwingung der Saite und zu einem kleineren Teil, durch die Schwingung des Gitarrenkörpers erzeugt. Da beide Schwingungen räumlich nah beieinander liegen kann der Mensch bei einem normalen Abstand diese nicht voneinander unterscheiden, sondern nimmt beide gleichzeitig und zusammen als ein Klangereignis wahr. Man kann beide Schwingungen also auch als *eine* (kompliziertere) Schwingung des *Systems* Gitarre auffassen. Mathematisch wird eine solche Schwingung durch eine periodische Funktion beschrieben. Die genaue Gestalt dieser Funktion ist zunächst nicht bekannt, jedoch einige ihrer Eigenschaften, welche es ermöglichen die Schwingung auf eine andere Art zu beschreiben, welche für eine Analyse besser geeignet ist.

Jede periodische Funktion kann durch eine Kombination anderer periodischer Funktionen dargestellt werden. Kombination bedeutet hier eine mathematische Operation wie Addition, Multiplikation usw. Eventuell benötigt man zur Darstellung unendlich viele Glieder bei der Kombination. Der Vorteil ist aber, dass man eine komplizierte, d. h. mathematisch schwer zu beschreibende Funktion, durch eine Reihe bzw. Kombination von einfacheren Funktionen ersetzen kann. Der einfachste und universellste Fall ist es die Funktion als eine Addition von Sinusoiden¹ unterschiedlicher Frequenz f und Amplitude a darzustellen.

$$s(t) = a_1 \cos_1(f_1) + a_2 \cos_2(f_2) + a_3 \cos_3(f_3) + \dots \quad (2.1)$$

Die Darstellung der Funktion $s(t)$ durch Sinusoiden ist dann optimal, wenn die Anzahl der Sinusfunktionen minimal ist. Unter der Voraussetzung, dass die Funktion $s(t)$ periodisch ist, sind die Frequenzen f_1, f_2, \dots proportional zueinander, genauer sie sind

¹Sinus- oder Cosinusfunktionen. Da Sinus- und Cosinusfunktionen einander sehr ähnlich sind, sind sie für diese Anwendungen i. d. R. austauschbar. Sie werden daher gemeinsam als Sinusoiden bezeichnet.

alle Vielfache der kleinsten Frequenz. Diese kleinste Frequenz wird als Grundfrequenz bezeichnet, die restlichen Frequenzen bzw. die Sinusschwingung mit der entsprechenden Frequenz als Oberton. Alle Sinusschwingungen zusammen, inkl. der Grundschwingung, werden im Verlauf als Obertöne bezeichnet.

Ist die Funktion $s(t)$ an jeder Stelle beliebig oft differenzierbar, d. h. sie enthält keinen Knick oder Sprung, dann kann die Funktion durch endlich viele Obertöne dargestellt werden. Einen Knick im Schwingungsverlauf kann man durch beliebig viele Obertöne beliebig genau nähern. Da Computer jedoch nur mit diskreten, also endlich genauen, Werten rechnen können, kann eine unendlich differenzierbare Funktion im Computer nicht dargestellt werden. Die Funktion $s(t)$ kann also in der diskreten Darstellung exakt durch endlich viele Obertöne beschrieben werden, so dass sich dieses Verfahren in der Praxis benutzen lässt.

2.2 Was ist in einer Aufnahme enthalten?

Um Gitarrenklänge zu analysieren, muss zunächst einmal eine geeignete Aufnahme einer Gitarre vorliegen. Bei dieser Aufnahme dürfen nicht zwei oder mehr Töne gleichzeitig klingen. Des weiteren sollte jeder einzelne Ton eine längere Zeit klingen, damit genug *Klangmaterial* für die Analyse zur Verfügung steht. Am besten ist es, wenn einfach die gesamte Tondauer vom Anschlag an aufgenommen wird. Da die unterschiedlichen Tonhöhen auf den unterschiedlichen Saiten des Instrumentes sich wiederum ebenfalls unterscheiden, ist es angebracht, von jeder spielbaren Position auf dem Griffbrett mindestens eine Aufnahme zu haben, damit der gesamte Klangumfang des Instrumentes erfasst werden kann.

Diese Anforderungen werden von einem Musikstück kaum erfüllt. Die benötigten Aufnahmen müssen also für jedes zu untersuchende Instrument angefertigt werden.² Spieltechniken wie Vibrato oder Glissando machen den Klang zwar für Menschen interessanter, erschweren jedoch die Analyse nur unnötig und sollten deshalb bei der Aufnahme vermieden werden. Bei der Synthese können diese Techniken einfach aus dem grundlegenden Klang erzeugt werden. Ziel der Analyse ist es, eine Beschreibung für ein Instrument zu finden, welche möglichst unabhängig vom Spieler und seiner Technik ist. Für die weitere Betrachtung in diesem Kapitel gehen wir davon aus, dass die einzelne Aufnahme jeweils einen Ton enthält.

Abbildung 2.1 zeigt den Ton F (174 Hz) einer akustischen Gitarre. Wie bei allen Saiteninstrumenten, die nicht mit einem Bogen gespielt werden, ist die maximale Lautstärke bereits kurz nach dem Anschlag erreicht und nimmt danach gleichmäßig wieder ab. Dieses Verhalten ist typisch für einen akustischen Vorgang, bei dem nur einmal am Anfang Energie aufgewendet wird, um das System (die Saite) zu erregen. Dabei wird die Saite an einem Punkt durch den Finger der Spielers aus ihrer Ruheposition gebracht. Wird die Saite losgelassen, so schwingt sie wieder in die Ruheposition zurück, kann bei Erreichen derselben jedoch aufgrund der Massenträgheit nicht abrupt stehen bleiben. Es bildet sich eine Schwingung um die Ruheposition herum aus. Schwingt das System, dann wird

²Die für diese Arbeit benutzte Aufnahmetechnik wird im Kapitel 4 vorgestellt.

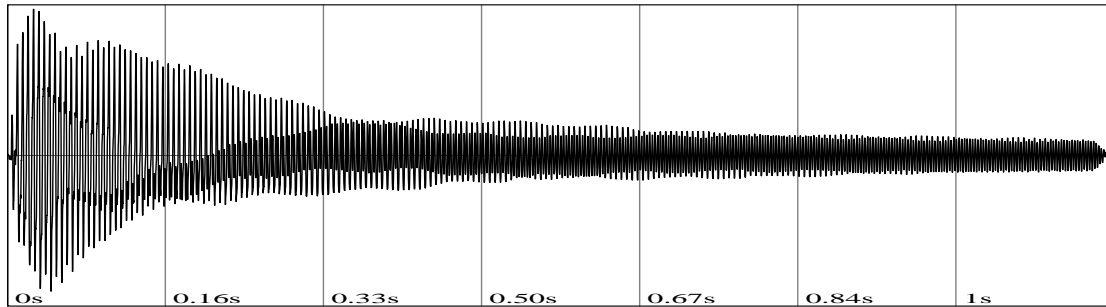


Abbildung 2.1: Gitarrenton

ihm kontinuierlich Energie entzogen (durch Schallabstrahlung und interne Reibungsverluste), so dass die Lautstärke nach Erreichen des Maximums stetig abnimmt. Spielt man ein Saiteninstrument mit einem Bogen, so wird der Energieverlust des schwingenden Systems durch das Streichen des Bogens wieder ausgeglichen. Dadurch ist mit solchen Instrumenten ein beliebig langer Ton, der sogar seine Lautstärke im Verlauf beliebig ändern kann, möglich.³

Betrachtet man den Anfang des Tons⁴ in [Abbildung 2.1](#), so fällt auf, dass die maximale Amplitude der Schwingung erst einige Perioden nach Beginn des Klangs erreicht wird. Die Annahme des letzten Absatzes, dass direkt nach dem Anschlag durch den Finger, an dem die maximale Auslenkung der Saite erreicht ist, auch die Lautstärke maximal wäre, ist also nicht zutreffend. Vielmehr benötigt die Saite eine gewisse Zeit um das Lautstärke-Maximum zu erreichen. Man teilt den Verlauf des Gitarrentons deshalb in zwei *Phasen*.⁵ Die erste Phase vom Anschlag bis zum Lautstärkemaximum wird *Attackphase* genannt, der Rest des Tons ist die *Sustainphase*.⁶

Nachdem sich die Saite eingeschwungen hat, beginnt kurz nach dem Lautstärkemaximum die Sustainphase. Im Gegensatz zum Attack ändert sich der Klangeindruck nun langsamer, wobei der Klang mit abnehmender Lautstärke auch dumpfer klingt. Ein dumpfer Klang bedeutet i. d. R., dass hauptsächlich tiefe Frequenzen vorhanden sind, bzw. die tiefen Frequenzen deutlich größere Amplituden haben als die hohen oder mittleren. Mit zunehmender Dauer des Klangs verringert sich die Amplitude der hohen Frequenzen, wobei die sich die Amplitude umso schneller verringert, je höher die Frequenz ist. Dies liegt an dem Energieverlust der Saite, welcher mit zunehmender Frequenz proportional zunimmt.

Die [Abbildungen 2.2](#) und [2.3](#) zeigen die Sonogramme des tiefsten und höchsten spielbaren Tons einer E-Gitarre. Beim tiefen E (83 Hz) haben die höchsten Obertöne eine

³Gleiches gilt auch für Blasinstrumente, bei denen durch die sich bewegende Luft der Energieverlust ausgeglichen wird. Da die im Blasinstrument schwingende Luft im Vergleich zur Saite eine verschwindend geringe träge Masse besitzt, erlischt der Ton auch fast gleichzeitig mit dem Ende des Blasens.

⁴im linken Teil des Bildes

⁵Diese Phasen sind nicht zu verwechseln mit der Phase einer Sinusfunktion.

⁶Die Namen sind dem ADSR Modell entnommen, welches den Lautstärkeverlauf eines Tons anhand von vier Phasen beschreibt: Attack, Decay, Sustain, Release. Das ADSR Modell wurde 1965 von Vladimir Ussachevsky zur Steuerung der Moog Synthesizer erfunden. [[Leo03](#)] [[MC00](#)]

2 Analyse

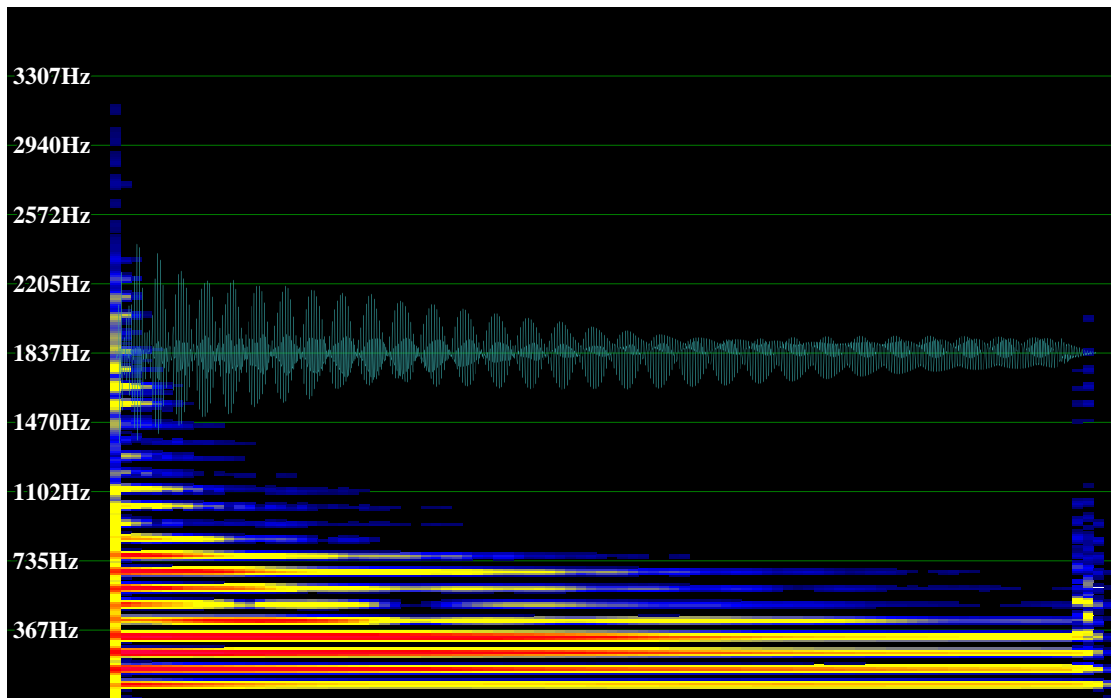


Abbildung 2.2: Sonogramm des tiefen E 83 Hz

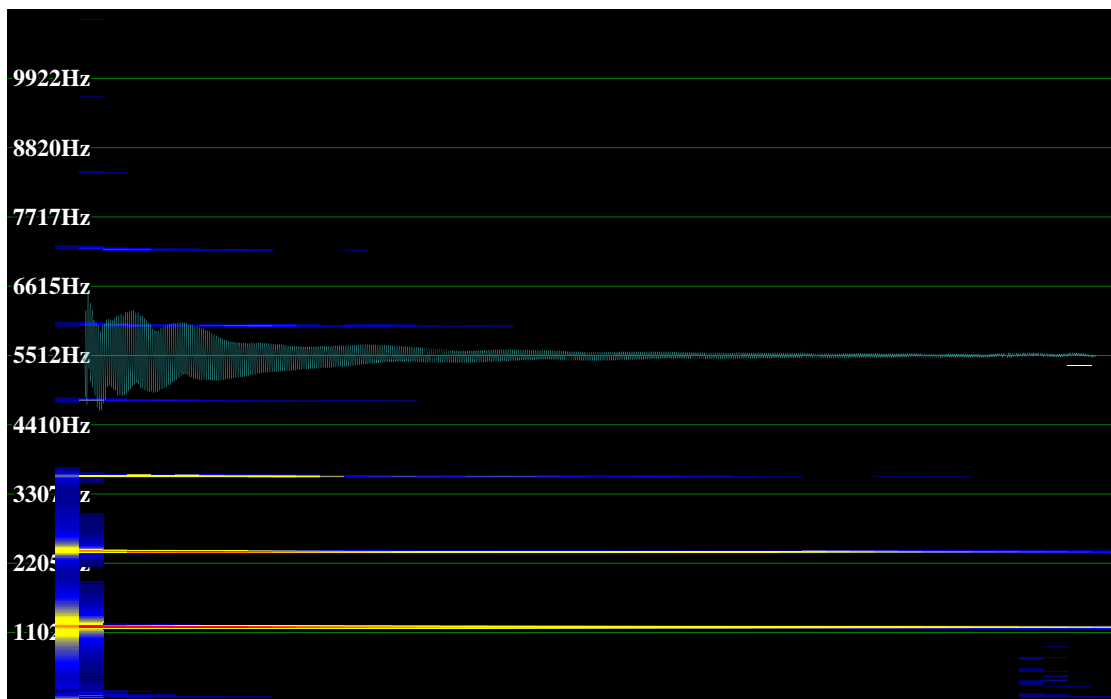


Abbildung 2.3: Sonogramm des hohen D 1176 Hz

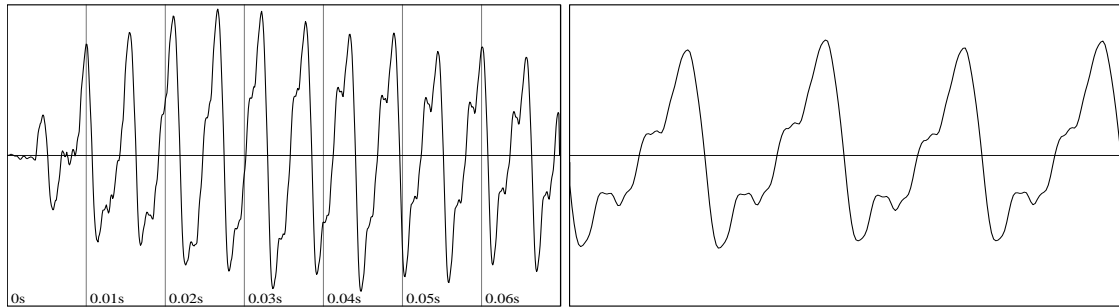


Abbildung 2.4: links: Attackphase, rechts: 4 Perioden in der Sustainphase

Frequenz von ≈ 3200 Hz. Dies ist nur während der Attackphase der Fall. Mit Erreichen der Sustainphase gibt es 20 Obertöne bis ≈ 1800 Hz. Bei der Hälfte des Tons sind nur noch 8 Obertöne vorhanden bis ≈ 700 Hz. Diese bleiben bis zum Ende des Tons auch erhalten. Man sieht, dass Frequenzen über 1000 Hz sehr schnell verschwinden. Entsprechend hat das hohe D mit einer Grundfrequenz von 1176 Hz auch eine kurze Dauer, da die Obertöne bei einer so hohen Grundfrequenz sehr schnell absorbiert werden. Am Anfang der Sustainphase sind hier lediglich 6 Obertöne vorhanden bis ≈ 7000 Hz. Nach Ablauf der halben Sustainphase sind noch drei meßbar bis ≈ 3500 Hz.

Die Sonogramme zeigen, dass die Attackphasen der Klänge aus vielen Frequenzen bestehen, die nicht unbedingt den (reinen) Obertönen der Grundfrequenz entsprechen. Die Attackphase ist jedoch nur kurz und entspricht bei diesen Klängen $\approx 0,1$ s.⁷ Die Sustainphase besteht hauptsächlich aus Frequenzen, die den reinen Obertönen entsprechen. Die hohen Frequenzen in den Sonogrammen am Ende der Klänge resultieren aus dem Fadeout am Ende des Samples, welches nachträglich eingefügt wurde, um die Samplelänge zu verkürzen. Durch dieses Fadeout werden die hohen Frequenzen erzeugt, welche bei einem natürlichen Ausklang nicht auftreten würden.

Die Sustainphase kann folglich durch die Amplituden der Obertöne im zeitlichen Verlauf des Klangs beschrieben werden. Frequenzen, die nicht einem Oberton entsprechen, kommen in der Sustainphase nicht vor. Für die Attackphase ist eine Beschreibung nur mittels Obertönen nicht ausreichend.

2.3 Detaillierte Analyse der Attackphase

In *Abbildung 2.4* links sieht man den Anfang eines Klangs. Mit der sechsten Periode ist bereits die maximale Lautstärke erreicht, ab diesem Zeitpunkt beginnt also die Sustainphase da mit dem Erreichen des Maximums die Saite vollkommen eingeschwungen ist. Die Attackphase umfasst in diesem Fall die ersten fünf Perioden. Über dem gesamten Tonumfang der Gitarre variiert die Länge der Attackphase zwischen 2 und 6 Perioden.

Durch den Anschlag der Saite wird diese aus ihrer (nicht schwingenden) Ruheposition

⁷Die Attackphase entspricht hier zwei Spalten im Sonogramm. Eine Spalte entspricht 4096 Samples, bei 44100 Hz Samplerate also $\approx 0,1$ s.

2 Analyse

gebracht und dabei in eine Dreiecksform verbogen. Verläßt der Finger bzw. das Plektrum dann die Saite, beginnt die Schwingung, indem die Saite wieder in Richtung ihrer Ruheposition zurückschwingt. Die Bewegung solch einer transversalen Schwingung wird durch die allgemeine Wellengleichung

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} \quad (2.2)$$

beschrieben, welche eine allgemeine Lösung der Form

$$y = f_1(ct - x) + f_2(ct + x) \quad (2.3)$$

hat. c ist eine Konstante, die aus der Dichte und Spannung der Saite gebildet wird.⁸

Formel 2.3 zeigt, dass die Schwingung der Saite durch zwei Schwingungen bzw. Wellen beschrieben werden kann, die in entgegengesetzter Richtung auf der Saite *laufen*. An den Saitenenden werden die Wellen reflektiert, dabei jedoch die Frequenzanteile der Welle, die nicht einem Schwingungsmodus der Saite bzw. einem Oberton entsprechen stark abgedämpft. Die Energie dieser gedämpften Frequenzen wird auf Frequenzen übertragen, die Obertönen entsprechen. Die Amplitude der Obertonfrequenzen nimmt also zu und damit auch die Gesamtamplitude. Nach ein paar Reflektionen bzw. Perioden enthalten die beiden Wellen (fast) nur noch Frequenzen, die Obertönen entsprechen, wodurch die gesamte Energie des Anschlags in Obertonfrequenzen umgesetzt wurde. Die Schwingung der Saite hat sich nun etabliert und die Amplituden nehmen aufgrund der allgemeinen Dämpfung des Schwingungssystems Gitarre wieder ab.⁹

2.4 Detaillierte Analyse der Sustainphase

Abbildung 2.4 rechts zeigt einen kurzen Ausschnitt aus der Sustainphase. Es sind zwei Perioden der Grundfrequenz abgebildet. Obwohl sich der Klang während der Sustainphase stetig ändert, sich also die Amplituden der Obertöne stetig ändern, sind diese Veränderungen für kurze Zeitabstände jedoch sehr gering. In der Abbildung kann man beispielsweise keinen Unterschied zwischen der linken und rechten Periode erkennen. Für einen kurzen Zeitabschnitt kann man also die Schwingung in der Sustainphase als eine periodische Funktion betrachten.

Die spektrale Zusammensetzung einer periodischen Funktion kann durch die Fourier-Transformation ermittelt werden. Da eine digitale Aufnahme eine Menge von diskreten Messwerten ist und keine stetige (mathematische) Funktion, benötigt man zur Spektralanalyse eine spezielle Variation, die *Diskrete Fourier-Transformation* (DFT).¹⁰

Der *Frame* ist die Menge von Werten, welche transformiert wird. Da dies oftmals nur ein kleiner Teil der gesamten zu untersuchenden Wertemenge ist, schaut man sozusagen durch ein Fenster von jeweils 4096 Samples auf den gesamten Klang. Das Ergebnis der

⁸Eine Herleitung dieser Formeln findet sich in [FR98] und [Coo02].

⁹Die im letzten Kapitel erwähnten Modelle für schwingende Saiten *Karplus-Strong* und *Digital Waveguide* basieren auf dem Modell, die Schwingung der Saite durch zwei Wellen zu beschreiben.

¹⁰Näheres zur DFT in [Smi03] und `RealSoundSynthesis`.

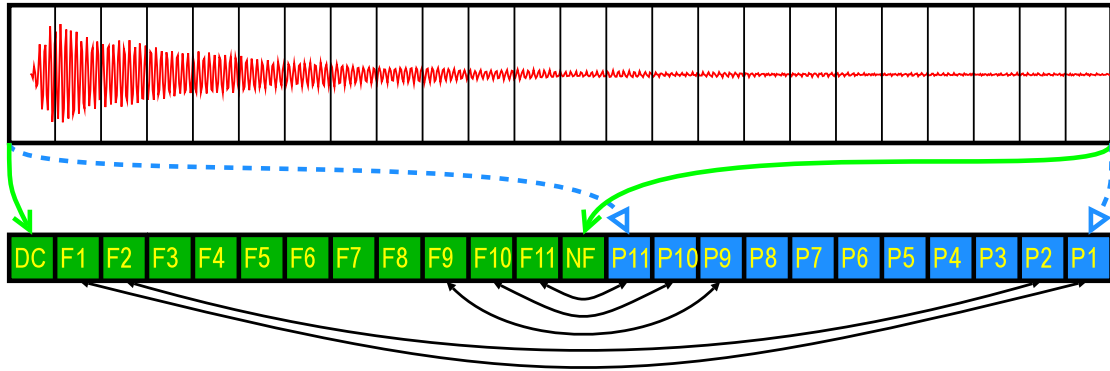


Abbildung 2.5: Darstellung der Arbeitsweise der DFT

DFT, wenn man sie als spektrale Untersuchung interpretiert, ist eine halb so große Menge von Werten, *Bin* genannt. Ein Bin beschreibt das Verhalten eines (kleinen) Abschnitts im Frequenzspektrum, bzw. die in diesem Abschnitt auftretenden Frequenzen. Ein Bin wird durch zwei Werte der Ergebnismenge der Transformation beschrieben. Um die Amplitude und Phase der jeweiligen Frequenz zu ermitteln muss das Ergebnis der DFT noch einmal umgerechnet werden. Die DFT berechnet nämlich nur die Amplitude je einer Sinus- und Cosinusfunktion der dem Bin entsprechenden Frequenz. Die Amplituden werden als realer und imaginärer Teil einer komplexen Zahl aufgefasst, wobei die realen Teile alle in der linken Hälfte des Ergebnisframes abgelegt werden, die komplexen Teile in der rechten. Aus beiden Teilen können dann Amplitude und Phase einer Sinusfunktion durch folgende Formeln berechnet werden.

$$A[x] = \sqrt{\operatorname{Re}[x]^2 + \operatorname{Im}[\operatorname{Framesize} - x]^2} \quad (2.4)$$

$$P[x] = \arctan\left(\frac{\operatorname{Im}[\operatorname{Framesize} - x]}{\operatorname{Re}[x]}\right) \quad (2.5)$$

Nach der Umrechnung erhält man einen Datensatz gemäß [Abbildung 2.5](#).

Der erste (linke) Bin entspricht der Frequenz 0 Hz, bzw. dem Gleichanteil der Eingabe. Für eine Frequenz von 0 Hz ist eine Phase nicht sinnvoll, deshalb besitzt dieser Bin keinen Phasenwert. Für die Nyquist-Frequenz¹¹ kann ebenfalls keine Phase mehr dargestellt werden kann. Deshalb haben die Frequenzen zwei Werte mehr als die Phasen. Die Frequenzen von 0 Hz bis zur Nyquist Frequenz werden linear auf die Anzahl der Frequenzbins verteilt. Je größer der Frame, desto höher ist die Frequenzauflösung, bzw. desto genauer kann eine einzelne Frequenz bestimmt werden. Der numerische Wert eines Bins entspricht der Summe der Amplituden und Phasen aller dem Bin entsprechenden Frequenzen. [Abbildung 2.6](#) zeigt, wie das Spektrum eines Klangs (grüne Kurve) in 23 Bins (blaue Kurve) eingeteilt wird. Jeder Bin entspricht dem Mittelwert der Frequenzen des Spektrums innerhalb des Bins.

¹¹Die Nyquist-Frequenz ist $\frac{\text{Samplerate}}{2}$ Hz und legt die maximal darstellbare Frequenz in einem Sample fest.

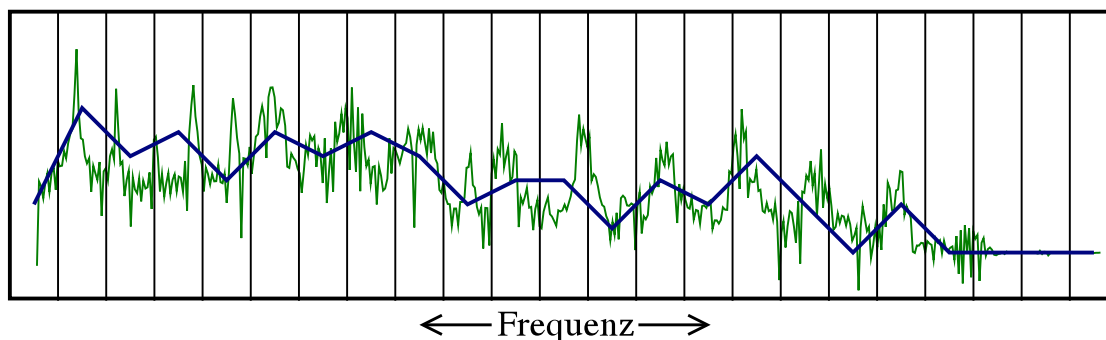


Abbildung 2.6: Frequenzauflösung der DFT, Größe der Bins

Um das Ergebnis der DFT als Spektraluntersuchung zu interpretieren, müssen die Werte eines Frames jedoch eine (annähernd) periodische Funktion darstellen. Entsprechen sie keiner periodischen Funktion, so ist das Ergebnis der DFT das Spektrum der gemittelten Perioden. Enthält ein Frame beispielsweise in der ersten Hälfte eine Sinusschwingung mit 500 Hz, Amplitude 1 und in der zweiten Hälfte eine Sinusschwingung mit 1000 Hz, Amplitude 1 so wird die DFT das Ergebnis von zwei Sinusschwingungen mit 500 und 1000 Hz und Amplitude 0.5 ergeben. Die Information, dass beide Schwingungen nicht gleichzeitig vorhanden waren ist durch die DFT verloren gegangen.

Um die Sustainphase mittels DFT zu analysieren, muss für die Framegröße ein Kompromiss aus Frequenz- und Zeitauflösung gefunden werden. Ein guter Erfahrungswert ist es, die Framegröße so zu wählen, dass sie ungefähr einer zehntel Sekunde entspricht. Innerhalb eines Zehntels ändert sich der Klang kaum. Bei einer Samplerate von 44100 Hz ergäbe dies eine Framegröße von 4410 Samples. Da die optimierte Version der DFT¹² am schnellsten bei 2er-Potenzen ausgeführt werden kann, wählt man meistens eine Framegröße von 4096 Samples. Ein Bin hat somit eine Frequenzauflösung von $\frac{44100/2}{4096/2} \approx 10$ Hz.¹³

Eine Sinusschwingung wird durch 3 Parameter exakt beschrieben: Frequenz, Amplitude und Phase. Genau diese drei Parameter werden durch die DFT berechnet. Da ein Klang durch die Amplituden seiner Obertöne genau beschrieben werden kann, hat man mit der DFT ein Verfahren um alle benötigten Informationen zur Analyse eines Klangs zu ermitteln.

Bei einer Framegröße von 4096 Samples ergibt die DFT $2048 + 2$ Werte von Amplituden der Frequenzen und $2048 - 2$ Werte von Phasen der Frequenzen. Der größte Teil dieser Werte ist jedoch irrelevant, da sie keinem Oberton entsprechen (und somit ein Amplitude von ≈ 0 haben). Es reicht, wenn nur die einem Oberton entsprechenden Werte weiter betrachtet werden. Bei der Untersuchung der Sonogramme habe ich festgestellt, dass die obertonreichen tiefen Klänge einer Gitarre in der Sustainphase maximal 20 Obertöne besitzen. Die DFT kann also auf drei Mengen bzw. Vektoren mit ungefähr 20 Tupeln reduziert werden. Die Vektoren enthalten die Frequenz, Amplitude und Phase

¹²Fast Fourier-Transformation (FFT)

¹³ $44100/2$ entspricht der höchsten darstellbaren Frequenz (Nyquist). $4096/2$ entspricht der Anzahl der Bins welche Frequenzen entsprechen.

2 Analyse

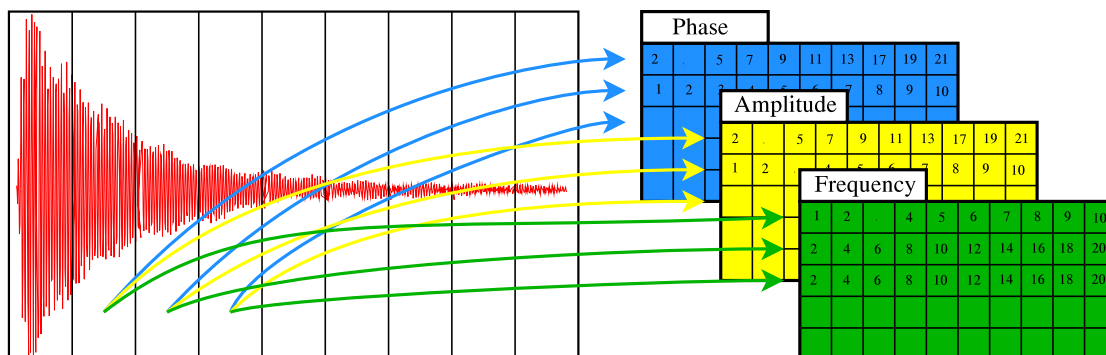


Abbildung 2.7: Konstruktion der Matrizen

aller Obertöne eines Klangs.

Das zu analysierende Sample ist in der Regel größer als 4096 Samples. Es wird also durch mehrere DFT Transformationen untersucht, jede im Abstand einer Framelänge. Man erhält also auch mehrere Vektoren für Frequenz, Amplitude und Phase. Diese Vektoren kann man nun in drei Matrizen als jeweils eine Zeile eintragen. Diese Matrizen enthalten dann jeweils 20 Spalten für jeden potentiell vorhandenen Oberton und $\text{Samplelänge}/4096$ Zeilen. In den drei Matrizen sind dann alle Parameter der Obertöne des Klangs enthalten. Unter der Annahme, dass der Klang des Instrumentes vollständig aus Obertönen konstruiert werden kann und dass für kurze Zeitabschnitte der Klang in der Sustainphase periodisch ist, repräsentieren diese Matrizen alle relevanten Informationen des Klangs. Im Vergleich zum vollständigen Sample ist bereits eine erhebliche Reduktion der Daten bzw. Parameter vorgenommen worden. Eine Samplelänge von 3 Sekunden und eine Samplerate von 44100 Hz ergibt eine Samplegröße von 132300 Samples. Dem gegenüber stehen 3 Matrizen mit 20 Spalten und 32 Zeilen = 1920 Werte. Dies entspricht einer Reduktion auf $\approx 1.5\%$.

1920 Werte zur Beschreibung eines Klangs sind zwar im Vergleich zum gesamten Sample schon eine gute Verbesserung, aber immer noch zu viele um damit praktisch zu arbeiten. Wollte man hiermit beispielsweise den Unterschied im Klang zwischen einer akustischen und elektrischen Gitarre erläutern, so müsste man eine Relation zwischen zwei Mengen von jeweils 1920 Werten herstellen. Die Werte müssen für eine sinnvolle Auswertung und Interpretation weiter komprimiert bzw. abstrahiert werden.

Wären die Werte in den Matrizen unkorreliert, so könnte man keine Beziehungen zwischen den Werten untereinander feststellen und hätte somit auch keinen Ansatz, die Anzahl der Werte zu verringern. Andererseits sind komplett unkorrelierte Werte zufällig verteilt und entsprechen dem Klang von Rauschen. Der Gitarrenklang hat jedoch nur wenig Ähnlichkeit mit Rauschen, er entspricht eher einer sich langsam ändernden periodischen Funktion. Wäre der Klang ausschließlich eine periodische Funktion, so wären alle Zeilen der Matrix identisch. Im nächsten Kapitel wird gezeigt, dass sich die Zeilen in den Matrizen der untersuchten Klänge ähneln und es dadurch einen Ansatz für eine weitere Reduktion der Ergebniswerte gibt.

Der Vergleich der Analyseergebnisse von zwei Aufnahmen unterschiedlicher Instrumente ergibt dann den Unterschied im Klang. Einige Werte der Ergebnismenge sollten bei allen Gitarren ungefähr gleich sein. Diese Werte repräsentieren dann den *typischen* Gitarrenklang. Werte, die bei unterschiedlichen Gitarren unterschiedliche Größen annehmen, beschreiben den spezifischen Klang des Instrumentes bzw. Spielers. Analysiert man die Aufnahme eines anderen Instrumentes und vergleicht das Ergebnis mit dem einer Gitarre, sollten sich auch einige oder alle Werte unterscheiden, die beim Vergleich zweier Gitarren ungefähr gleich sind.

Je weniger Werte das Ergebnis der Analyse produziert und je besser sich diese Werte entsprechend diesen Kriterien verhalten, desto besser ist die Analyseverfahren. Im Idealfall hätte man nur zwei Werte, einer dessen numerische Größe für den allgemeinen Klang einer Gitarre steht und ein zweiter Wert, welcher das Modell bzw. den Spieler beschreibt.

2.5 Resynthese

Die im letzten Abschnitt vorgestellte Methode, den Klang zu analysieren, zielt darauf, ab den Klang der Gitarre mit möglichst wenig Werten/Parametern zu beschreiben. Ob diese Parameter aber tatsächlich für den Klang relevante Informationen enthalten oder etwas irrelevantes oder gar rein zufällig sind, kann durch die Analyse allein nicht sichergestellt werden. Sollte die Analyse wirklich die für den Klang relevanten Parameter bestimmen, so muss es möglich sein, aus diesen Parametern wieder einen Klang zu berechnen. Wenn nun dieser Klang genauso klingt wie der ursprüngliche, dann sind die für den Klang relevanten Informationen durch die Analyse erhalten worden und die nicht relevanten Informationen entfernt. Durch den Vergleich des ursprünglichen und des analysiert-resynthetisierten Klangs kann also eine subjektive Bewertung der Güte der Analyseverfahren vorgenommen werden.

Die Resynthese besteht aus mehreren Schritten, die den Schritten der Analyse entsprechen, in der jeweils gegensätzlichen Richtung. Entsprechend der Natur einer mathematischen Transformation kann die DFT durch die Inverse Diskrete Fourier-Transformation (IDFT) exakt die ursprüngliche Datenmenge erzeugen. Die aus der DFT erzeugten drei Matrizen lassen sich ebenfalls wieder in die Ergebnismenge einer DFT überführen, auf welche dann die IDFT angewendet werden kann. Bei der zu entwickelnden weiteren Reduktion der Matrizen ist dann darauf zu achten, dass es immer eine Möglichkeit gibt aus dem Ergebnis wieder die ursprüngliche Matrix zumindest näherungsweise wiederherzustellen.

2.6 Zusammenfassung

Um eine rechnergestützte Analyse des Gitarrenklangs durchzuführen, muß zunächst eine digitale Aufnahme einer Gitarre angefertigt werden. Die Aufnahme besteht aus jeweils einem einzeln gespielten Ton von allen Positionen auf dem Griffbrett. Den Einzelton unterteilt man in zwei Phasen, welche unterschiedlich analysiert werden. Die erste und deutliche kürzere Phase ist die Attackphase, die zweite die Sustainphase.

2 Analyse

Die Attackphase kann bei manchen Gitarren vernachlässigt werden, da sie durch einen geschickten Start der Synthese sehr gut imitiert werden kann. Bei Gitarren, bei denen diese Methode kein befriedigendes Ergebnis liefert, muß für die Attackphase eine spezielle Analyse und Synthese verwendet werden.

Die Sustainphase kann durch die spektrale Analyse in eine Reihe von Obertönen zerlegt werden, welche mit relativ wenigen Zahlen bzw. Parametern beschrieben werden können. Diese Darstellung erlaubt es einen (fast) beliebig langen Klang zu synthetisieren.

3 Design

Im letzten Kapitel wurde ein Verfahren beschrieben, den Klang von Gitarren durch eine Menge von Zahlen zu beschreiben. In diesem Kapitel wird nun eine Möglichkeit beschrieben, dieses Verfahren umzusetzen.

Vorausgesetzt wird, dass eine digitalisierte Aufnahme einer Gitarre vorliegt in der jeweils nur ein Ton gespielt wird. Diese Aufnahme kann einen oder auch mehrere Töne hintereinander enthalten. Bei mehreren Tönen wird die Aufnahme in entsprechend viele Abschnitte (automatisch) unterteilt und jeder Abschnitt einzeln bearbeitet. Die digitalisierte Aufnahme liegt als eine 1 kanalige, PCM¹ kodierte Datei vor mit einer Samplerate von 44100 Hz. Diese Datei wird dann im Speicher als ein Array von 32bit floating point Zahlen abgelegt. Der Wertebereich der Samples reicht dabei von $-1 \dots +1$.

3.1 Extraktion von Attack- und Sustainphase

Der zu analysierende Abschnitt enthält die Aufnahme eines Tons. Normalerweise befindet sich am Anfang des Abschnitts ein kurzes Stück Stille, bzw. der Abschnitt beginnt nicht genau mit dem Anschlag der Saite. Theoretisch kann der Anschlag aber zu einem beliebigen Zeitpunkt innerhalb der Aufnahme erfolgen und dieser Zeitpunkt variiert bei unterschiedlichen Aufnahmen. Zunächst muß der Anschlag der Saite genau an den Anfang des Aufnahmeabschnitts verschoben werden, bzw. eventuell vorhandene Stille am Anfang des Abschnitts entfernt werden.

Dazu wird für den Abschnitt ein Dezibelprofil erstellt. Über- oder unterschreitet die Lautstärke gewisse Pegel stellt dies den Anfang und das Ende des Klangs dar. Der Pegel wird mittels der folgenden Formel ermittelt:²

$$\text{dB} = 20 \log_{10} (|s|) \quad (3.1)$$

s ist der Samplewert im Bereich $-1 \dots +1$. Ein Pegel von 0 dB entspricht der maximalen Lautstärke, die halbe Lautstärke hat einen Wert von ≈ 6 und absolute Stille entspricht $-\infty$ dB. Die sonst bei Ermittlung von dB Pegeln übliche Division durch den theoretisch maximalen Wert kann hier entfallen, da der maximale Wert 1 ist.

Für jeden Samplewert des Abschnitts kann man mit Formel 3.1 den entsprechenden Pegel berechnen. Danach sucht man nach dem ersten Sample, dessen Pegel größer als ein Schwellenwert ist. Der Schwellenwert ist so gewählt, dass er ungefähr der erwarteten Lautstärke eines Tons kurz nach dem Anschlag entspricht. Die Position dieses Samples

¹*Pulse Code Modulated*. Die digitale Aufnahme wird durch eine Abfolge von Integerworten der Größe 8, 16, 24 oder 32bit gespeichert. Alternativ wird auch häufig das 32bit IEEE float Format für ein Wort benutzt. Näheres hierzu in [Smi03] und [Coo02].

²Eine ausführliche Beschreibung der unterschiedlichen Dezibel Varianten enthalten [Fou04] und [Kat02].

gibt einen Anhaltspunkt für den Beginn des Tons. Danach werden die nachfolgenden Pegel untersucht, bis sie unter einen zweiten Schwellenwert fallen. Der zweite Schwellenwert entspricht der geringsten Lautstärke, bei welcher der Tons noch hörbar ist, bzw. noch einem sinnvollen Klang entspricht.

Die gerade vorgestellte Methode funktioniert so jedoch nicht, da sie jedes einzelne Sample mit den Schwellenwerten vergleicht. Insbesondere beim zweiten Schwellenwert wird die Methode zu früh abbrechen, da prinzipiell jede Periode im Klang einen Samplewert ≈ 0 enthalten kann und somit unter den zweiten Schwellenwert fallen würde. Ermittelt man die Lautstärke aber nicht für jedes Sample einzeln, sondern immer für einen Bereich von Samples, kann das Ende des Klangs sehr präzise ermittelt werden und es wird auch nicht versehentlich zu früh ermittelt. Der Pegel eines Bereichs entspricht dann dem Pegel des betragsmäßig größten Samples innerhalb des Bereichs.

Sind die Aufnahmen gut ausgesteuert, liegt also die maximale Lautstärke zwischen den numerischen Samplewerten 0.8 und 1, haben sich Schwellenwerte von -15 dB und -25 dB als praktisch erwiesen, den Anfang eines Tons sicher zu erfassen.

$$\text{dB} = 20 \log_{10} (\max(|s[i]| \dots |s[i+n]|)) \quad (3.2)$$

Der zu untersuchende Abschnitt des Samples wird also in Bereiche von n Samples unterteilt. Für jeden Bereich wird der Pegel gemäß [Formel 3.2](#) ermittelt. Wird der obere Schwellenwert überschritten, so befindet sich innerhalb dieses Bereichs der Anfang des Klangs.³ Alle Bereiche vor dem aktuellen Bereich können gelöscht werden. Beim Erreichen des unteren Schwellenwertes (nachdem der obere erreicht wurde) ist der Klang beendet. Alle nachfolgenden Bereiche können gelöscht werden. Wird der obere Schwellenwert im gesamten Abschnitt nicht erreicht, so enthält der Abschnitt keine für die Analyse verwendbaren Daten.

[Abbildung 3.1](#) zeigt den Lautstärkeverlauf eines Tons. Die Höhe der bunten Fläche unterhalb des roten Samples gibt die maximale Lautstärke innerhalb des Abschnittes an. Die im gesamten Klang maximale Lautstärke entspricht der vollen Höhe des Diagramms. Die grünen Flächen bedeuten, dass die Lautstärke in diesem Abschnitt über dem unteren Schwellenwert liegt und somit die Samplewerte gültig sind. Die Lautstärke in Bereichen von blauen Flächen ist zu gering und die entsprechenden Abschnitte werden abgeschnitten. Die Samplewerte des *grünen Bereichs* werden nun weiter analysiert.

3.2 Bestimmung von Grundfrequenz und Obertönen

Wie im vorherigen Kapitel erläutert, sollen aus dem Sample drei Matrizen mit den für den Klang relevanten Frequenzen, Amplituden und Phasen erzeugt werden. Die Berechnung erfolgt mittels der diskreten Fourier-Transformation für mehrere Frames fester

³Die Annahme, dass der Klang in diesem Bereich *wirklich* beginnt trifft zu, da ein Gitarrenklang eine sehr kurze Attackphase besitzt. Die maximale Lautstärke des Klangs wird also schon nach sehr kurzer Zeit erreicht. Meine Analyse der Zeiten vom Anschlag der Saite bis zum Erreichen des Schwellenwertes von -15 dB über alle Spielpositionen auf einer Gitarre hat ergeben, dass eine Bereichsgröße von 128 Samples den Anfang sicher erfasst.

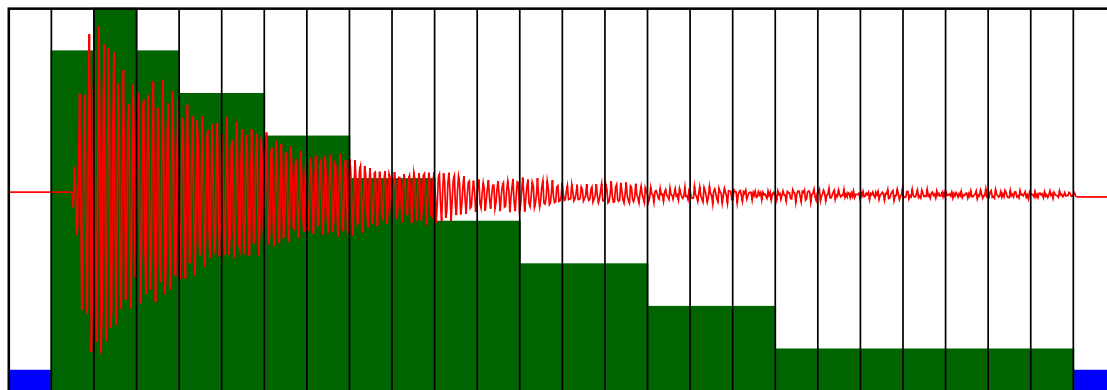


Abbildung 3.1: Abschnitte eines Samples und dessen Pegel

Größe. Dabei muss man zunächst eine sinnvolle Framegröße finden, die einen optimalen Kompromiss zwischen der zeitlichen Auflösung des Klangs und der spektralen Auflösung findet.

3.2.1 Parameter der DFT

Prinzipiell ändert sich der Klang des Gitarrentons stetig. Wie jedoch im vorherigen Kapitel diskutiert, kann man durchaus für einen kurzen Zeitabschnitt davon ausgehen, dass der Klang sich nicht verändert. Ein allgemeiner Erfahrungswert ist, dass sich in einem Zeitabstand von einer $1/10$ Sekunde der Klang während der Sustainphase kaum ändert. Das **KLANGBEISPIEL 5** zeigt jeweils drei Ausschnitte vom Anfang, Mitte, Ende eines Klangs aus drei unterschiedlichen Aufnahmen von jeweils $1/10$ Sekunde. Die einzelnen Zehntel werden jeweils 20 mal hintereinander gespielt, so dass sich eine Länge von 2 Sekunden pro Ausschnitt ergibt. Der erste Ausschnitt ist die zehntel Sekunde nach dem Erreichen der maximalen Lautstärke. Durch den in dieser Phase starken Lautstärkeabfall klingt dieser Ausschnitt nicht sehr gleichförmig. Die beiden nächsten Ausschnitte jedoch klingen fast wie ein konstanter und damit periodischer Klang.



Einer zehntel Sekunde entsprechen 4410 Samples. Der nächstgelegene Wert, der einer 2er Potenz entspricht ist 4096.⁴ Die Frequenzauflösung der diskreten Fourier-Transformation mit diesen Parametern entspricht folglich $44100/4096 \approx 10.8$ Hz. Ca. 10 Hertz Auflösung erscheinen zunächst nicht besonders präzise. So beträgt der Frequenzunterschied zwischen den tiefem E (82.4) und F (87.3) nur 4.9 Hz. Erst ab der zweiten Oktave bei F (174.6) und Fis (185.0) wird der Unterschied zwischen den Grundfrequenzen größer als 10 Hz. Die tiefsten 13 Töne der Gitarre können also mit diesen DFT Parametern nicht eindeutig bestimmt werden. *Zero Padding* ist eine Methode, die Frequenzauflösung der DFT zu verbessern. Dabei wird das zu untersuchende Sample vergrößert indem entsprechend viele Nullen angehängt werden. Durch das vergrößerte Sample verkleinert sich entsprechend der Frequenzumfang des einzelnen Bins, da sich die Samplerate nicht

⁴2er Potenzen haben bei der Verwendung einer FFT implementierungstechnische Vorteile und resultieren in einer verbesserten Performance der FFT.

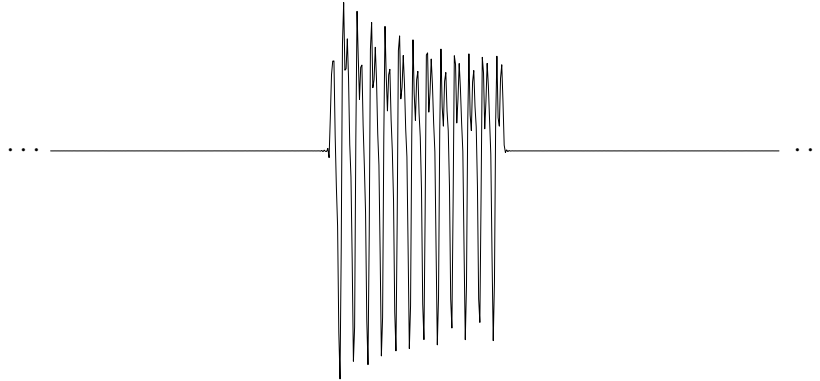


Abbildung 3.2: Sichtweise der DFT

verändert hat.⁵ Im weiteren Verlauf dieses Kapitels wird jedoch gezeigt, dass die ursprüngliche Frequenzauflösung von ≈ 10 Hz ausreicht, um die Klänge zu analysieren.

Die DFT ist die Variante der Fourier-Transformation für eine endlich große Menge von diskreten Werten. Genau wie die Fourier-Transformation für kontinuierliche Funktionen nimmt die DFT aber implizit an, dass die zugrundeliegende Menge unendlich groß ist. Alle nicht in der Menge enthaltenen Werte bekommen implizit den Wert 0. In unserem Fall hat der von der DFT transformierte Frame einen Gültigkeitsbereich von 0 bis 4095. Alle übrigen ganzen Zahlen von $-\infty$ bis $+\infty$ werden implizit auf 0 gesetzt. Nehmen wir also aus dem zu untersuchenden Sample einen Frame aus der Mitte heraus, so *sieht* die DFT diese Daten entsprechend [Abbildung 3.2](#).

Mathematisch ausgedrückt untersucht die DFT das Ergebnis der Multiplikation von zwei Funktionen. Die erste Funktion ist das Sample mit einem endlich großen Wertebereich, die zweite ist ein *Rechteckfenster* entsprechend Formel [3.3](#) mit einem unendlich großen Wertebereich. Das Ergebnis ist eine Funktion, die von $-\infty$ bis $+\infty$ definiert ist.

$$f(x) = \begin{cases} 1 & : 0 \leq x < n \\ 0 & : \text{sonst} \end{cases} \quad (3.3)$$

Bei einer Framegröße von 4096 ist $n = 4096$.

Entsprechend dem Faltungstheorem entspricht eine Multiplikation im Zeitbereich einer Faltung im Frequenzbereich. Wird also das Sample vor der Diskreten Fourier-Transformation mit der Rechteckfunktion multipliziert (was sich durch die Benutzung des Frames nicht verhindern lässt), dann ist das Ergebnis entsprechend verfälscht. Genauer gesagt sind die Amplituden der Frequenzen mit einer von der Rechteckfunktion abhängigen Funktion überlagert. Für eine genaue Frequenzanalyse ist dieses Ergebnis nicht mehr verwendbar.

⁵Eine genaue Beschreibung des Zero Paddings findet sich in [\[Coo02\]](#).

3 Design

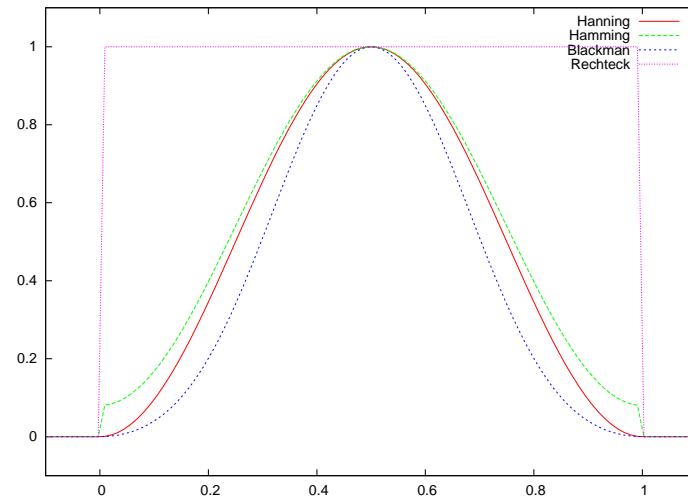


Abbildung 3.3: Fensterfunktionen

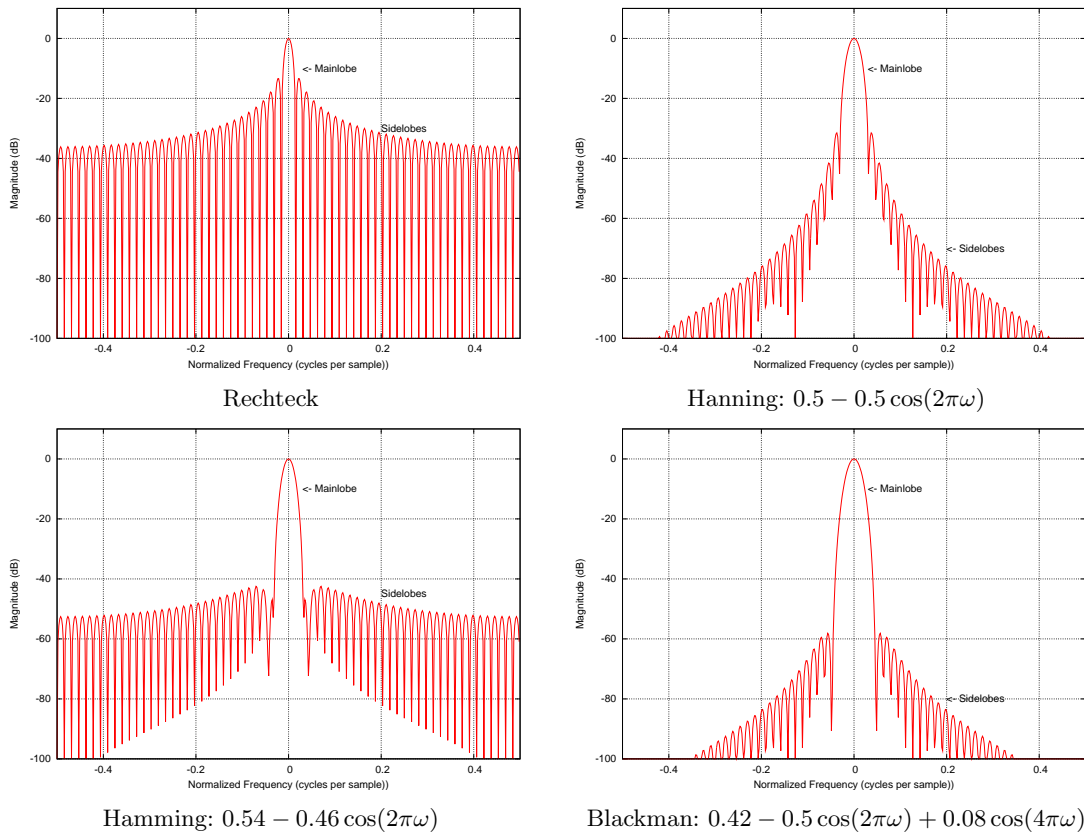


Abbildung 3.4: Spektrum von Fensterfunktionen

$$\delta(x) = \begin{cases} 1 & : x = 0 \\ 0 & : x \neq 0 \end{cases} \quad (3.4)$$

Das neutrale Element der Faltung ist die Delta Funktion (Formel 3.4). Das Frequenzspektrum der Fensterfunktion sollte möglichst große Ähnlichkeit mit der Delta Funktion haben, damit die Faltung mit der Fensterfunktion das Spektrum des untersuchten Klangs möglichst wenig ändert. Man sieht in [Abbildung 3.4](#), dass das Rechteckfenster die schlechteste Fensterfunktion zur Analyse ist. Die Sidelobes sind wesentlich breiter und haben vor allem einen größeren numerischen Wert. Dies resultiert in einer stärkeren Tiefpaßfilterung des Spektrums. Eine im Klang vorhandene Frequenz wird also nicht mehr durch einen spitzen Peak im Spektrogramm angezeigt, sondern beeinflusst auch die Peaks benachbarten Frequenzen, bzw. erhöht ihren numerischen Wert. Dies wird als *Spectral Leakage* bezeichnet, da eine Frequenz quasi auf ihre Nachbarn *leckt*. Das Rechteckfenster besitzt allerdings das schmalste Mainlobe mit einer Breite die der Frequenzauflösung eines Bins entspricht und kann deshalb eine Frequenz am präzisesten bestimmen. Aufgrund der geringen Auflösung der DFT bei der Analyse bringt diese Präzision in unserem Fall jedoch keine Vorteile.

Andere Fensterfunktionen bieten hier ein besseres Spektrum. Dabei unterscheiden sie sich in der Breite der Lobes und in der Amplituden der Sidelobes. Den besten Kompromiss bezüglich der Sidelobes für den Fall, dass das gesamte Spektrum untersucht werden soll, bietet das *Blackman-Fenster* entsprechend [Formel 3.5](#). Ist man nur an einem Teil des Spektrums interessiert, so kann man frequenzoptimierte Fenster entwickeln, welche für einen Teil des Spektrums noch bessere Resultate erzeugen. Da jedoch in unserem Fall unklar ist, welche Frequenzen für die Analyse von Interesse sind, muss das gesamte Spektrum untersucht werden. Eine ausführliche Diskussion über Fensterfunktionen findet sich in [\[Smi03\]](#).

$$\text{blackman}(x) = 0.54 - 0.46 \cos(2\pi x) + 0.08 \cos(4\pi x) \quad , 0 \leq x \leq 1 \quad (3.5)$$

[Abbildung 3.5](#) zeigt das Ergebnis wenn ein Frame des (gesamten) Samples mit einem Blackman Fenster multipliziert wird. Man erkennt, dass “Ereignisse” am Anfang und Ende des Frames nur stark abgeschwächt analysiert werden können, da sie durch das Fenster stark gedämpft werden (Lautstärke nahe 0). Ereignisse in der Mitte des Frames werden dagegen fast unverändert an die DFT übergeben. Wenn man nun die Ereignisse am Anfang und Ende des Frames untersuchen möchte, so muss man eine zweite Analyse machen, mit einem Frame, der um eine halbe Framelänge versetzt ist. In diesem zweiten Schritt ist dann der Anfang bzw. das Ende in der Mitte des neuen Frames und kann analysiert werden. Dieses Verfahren wird als *Sliding Window* bezeichnet, da das Fenster nicht immer um die volle Fensterbreite über dem zu untersuchenden Klang verschoben wird, sondern um einen beliebigen Teil. Es *gleitet* quasi über den Klang.

Zur Untersuchung der Sustainphase ist der Gebrauch des Sliding Window nicht nötig, denn die Schwingungen innerhalb eines Frames sind periodisch. Ein Frame von 4096 Samples bzw. $\frac{1}{10}$ Sekunde enthält jedoch mindestens 8 Perioden (beim tiefen E mit

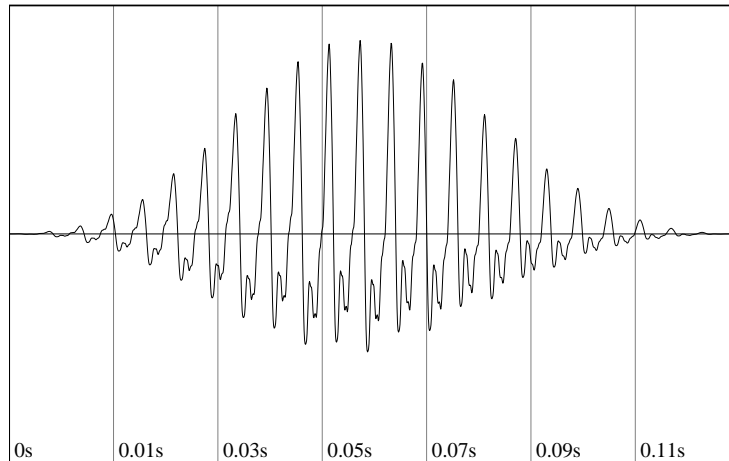


Abbildung 3.5: Blackman Fenster auf ein Sampleframe angewendet

≈ 80 Hz), so dass in der für die Untersuchung relevanten Framemitte noch genügend Perioden zur Analyse vorhanden sind.

3.2.2 Bestimmung der Grundfrequenz

Der wichtigste beschreibende Parameter des gesamten Klangs ist die Grundfrequenz. Sie legt die wahrgenommene Tonhöhe, und damit auch den Namen des Tons (c, fis usw.), und das Verhältnis der Obertöne fest. Je genauer die Grundfrequenz bestimmt wird, desto genauer sind auch die davon abhängigen oder abgeleiteten Analyseergebnisse.

Mittels der DFT kann die Grundfrequenz ermittelt werden. Das Ergebnis einer DFT ist jedoch nur sinnvoll, wenn die Samples innerhalb des untersuchten Frames einen periodischen Klang enthalten und ferner mindestens eine Periode im Frame enthalten ist. Über den gesamten Klangverlauf kann man jedoch nicht von einer periodischen Funktion sprechen. Bei dem Arbeiten mit Frames, in denen der Klang als periodisch betrachtet werden kann, ist die Frequenzauflösung nicht sehr hoch; die Framegröße muß hinreichend klein sein, damit der Klang innerhalb eines Frames ungefähr periodisch ist. Eine genaue Grundfrequenzbestimmung, mit einer Auflösung von ungefähr einem Herz, ist deshalb mit der DFT nicht möglich.

Eine weitere Möglichkeit, die Grundfrequenz zu bestimmen, ist mittels Autokorrelation.⁶ Dabei muß die obere Grenze sinnvoll bestimmt werden, so dass wirkungsvoll verhindert werden kann, dass dieselbe Periode mit sich selbst korreliert. Bei Benutzung einer unteren Grenze kann die Anzahl der Rechenschritte reduziert werden. Beide Grenzen lassen sich bei der Analyse von Gitarrenklängen einfach bestimmen, durch das tiefe E mit ≈ 82 Hz und das hohe e mit ≈ 1320 Hz. Setzt man die untere Grenze auf 80 Hz und die obere Grenze auf 1350 Hz, so lassen sich alle Tonhöhen der Gitarre sicher mittels Autokorrelation bestimmen.

⁶Eine ausführliche Beschreibung dieses Verfahrens bietet [Anhang A](#).

Das Ergebnis des Autokorrelationsalgorithmus aus [Anhang A](#) ist die Anzahl von Samples, bei denen die Autokorrelation den größten oder niedrigsten Wert berechnet hat.⁷ Das Sample besitzt also in diesem Abstand die größte Ähnlichkeit mit sich selbst. Aus dieser Zahl lässt sich mit [Formel 3.6](#) die diesem Abstand entsprechende Frequenz berechnen. Diese Frequenz ist die Grundfrequenz des Klangs.

$$f = \frac{\text{Samplerate}}{\text{Autocorrelate}(M)} \quad (3.6)$$

Die Autokorrelation ermittelt die Grundfrequenz mit der maximal möglichen Genauigkeit. Diese wird durch die Samplerate vorgegeben und ist weiterhin frequenzabhängig. Tiefe Frequenzen können wesentlich genauer bestimmt werden als hohe, da für die tiefen Frequenzen *mehr Samples zur Verfügung stehen*. Für den tiefsten Ton mit 82 Hz entspricht die Genauigkeit ≈ 0.15 Hz (1.8‰, 0.73 Cent)⁸, beim höchsten Ton mit 1320 Hz sind es ≈ 39.3 Hz (3%, 31 Cent).⁹

3.3 Bestimmung der Obertöne

Neben der Grundfrequenz sind die Frequenzen und Amplituden der Obertöne wichtige Parameter des Klangs. Mittels Autokorrelation lassen sich jedoch Obertöne nicht bestimmen. Die Obertöne müssen mit der DFT ermittelt werden.

Um einen Anhaltspunkt zu erhalten, welche Obertöne im gesamten Verlauf des Klangs vorkommen, kann man eine DFT über den gesamten Klang berechnen. Da diese DFT mit einer großen Anzahl von Samples arbeitet, z. B. bei einer Klanglänge von 2.5 Sekunden ≈ 100000 Samples, ist die Frequenzauflösung hier sehr hoch. Dies führt dazu, dass nicht nur der einer Frequenz entsprechende Bin eine hohe Amplitude enthält¹⁰ sondern auch die direkt benachbarten. Der Plot einer solchen DFT Analyse zeigt keine spitzen Ausschläge (Peaks) bei einzelnen Bins, sondern eher recht runde *Berge* die mehrere Bins umfassen. Die Breite der Berge wird auch durch leichte Unterschiede in den Frequenzen der Obertöne vergrößert. Der einzelne Oberton ändert im Verlauf des Klangs neben seiner Amplitude auch im geringen Maße seine Frequenz. Da das Ergebnis der DFT immer das Mittel über die gesamte Analysedauer ist, führt eine Veränderung der Obertonfrequenz zu einem breiteren Berg.

Obertöne haben jedoch im idealen Fall ein ganzzahliges Vielfaches der Frequenz des Grundtons. Der Grundton wurde bereits mit der bestmöglichen Genauigkeit bestimmt. Es können also ausgehend vom Grundton die Frequenzen der idealen Obertöne berechnet werden, siehe [Formel 3.7](#).

$$\text{Oberton}_i = (i + 1) \cdot f \quad , 0 \leq i < \infty \quad (3.7)$$

⁷ Je nachdem ob die Autokorrelation durch Multiplikation oder Differenz berechnet wird, gibt der größte oder niedrigste Wert die größte Korrelation an.

⁸ Cent ist ein hunderstel Halbton. Die Berechnung der Genauigkeit erfolgt mit $\text{Cent} = (f \cdot \sqrt[12]{2} - f) \cdot \frac{100}{\text{Abweichung in Hz}}$

⁹ Herleitung der beiden Werte in [Anhang A.1](#).

¹⁰ ebenfalls die entsprechenden Phasenbins

3 Design

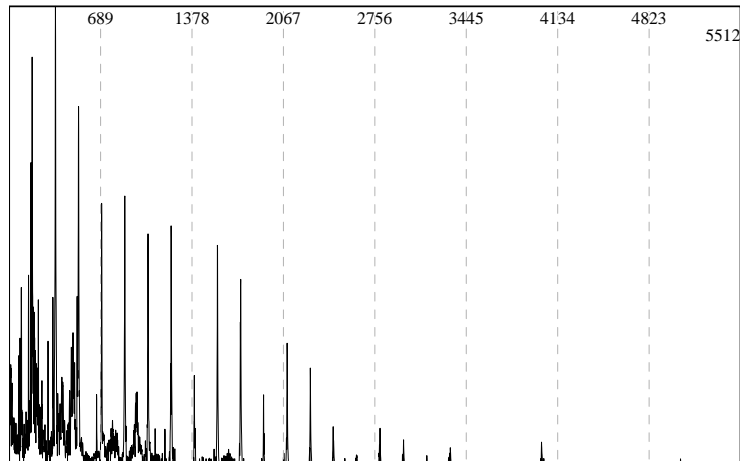


Abbildung 3.6: Amplituden der Frequenzen eines 2 sec Klangs

f ist die Frequenz des Grundtons. Der 0. Oberton entspricht dem Grundton. Da aufgrund des *Sampling Theorems* ein digitales Sample keine Frequenzen größer als die Nyquist-Frequenz enthalten kann¹¹, brauchen nur Obertöne bis zur Nyquist-Frequenz berücksichtigt werden und nicht potentiell unendlich viele. Für den folgenden Text gilt, dass die Menge aller Obertöne immer auch den Grundton beinhaltet. Man spricht im englischen dann auch von *Harmonics*.

Mittels [Formel 3.8](#) kann der Bin ermittelt werden, der die Frequenz f abdeckt.

$$\text{Bin} = \frac{f}{\text{Samplerate}/2} \cdot \frac{\text{Framegröße}}{2} + 0.5 \quad (3.8)$$

Setzt man in diese Formel die Frequenzen der erwarteten (idealen) Obertöne ein, so bekommt man die Nummer des Bins, der dieser Frequenz entspricht. Genau dieser Bin, oder einer der benachbarten Bins (in einem nicht zu großen Abstand) sollte ein lokales Maximum enthalten. Der Bin mit dem lokalen Maximum zeigt also die Frequenz des i -ten Obertons an. Beträgt der Suchbereich um den idealen Oberton 10% der Frequenz des Obertons, so wird das lokale Maximum bzw. der reale Oberton sicher gefunden.

Mit diesem Verfahren lassen sich die realen Frequenzen aller Obertöne für den Gesamtklang ermitteln. Wir erhalten damit einen Anhaltspunkt, welche Obertöne bei den folgenden Analyseschritten zu erwarten sind. Die Phase und Amplituden der Obertöne des gesamten Klangs sind uninteressant, da sie so gut wie keine Aussagekraft besitzen. Besonders die Amplitude nimmt im Klangverlauf bis auf 0 ab, so dass der gemittelte Wert keine brauchbare Information enthält.

¹¹Das Sampling Theorem wird in [\[Smi03\]](#) näher erläutert.

3.4 Berechnung von Frequenz-, Amplituden- und Phasenmatrix

Im letzten Kapitel wurde beschrieben, dass aus dem gesamten Klang drei Matrizen berechnet werden, die die Parameter der ersten 20 Obertöne jeweils im Abstand eines Frames von 4096 Samples enthalten. Der gesamte Klang wird also in Frames unterteilt. Für jeden Frame wird zunächst die Grundfrequenz mittels Autokorrelation bestimmt. Weicht diese Grundfrequenz zu stark von der Grundfrequenz des gesamten Klangs ab (Unterschied $> 5\%$), so enthält dieser Frame nicht brauchbare Daten, die jedoch im Mittel über den Gesamtklang verloren gingen. In den meisten Fällen ist dann dieser Frame und damit der gesamte Klang nicht uniform genug und würde keine brauchbaren Analyseergebnisse produzieren, so dass dieser Klang verworfen wird.

Im zweiten Schritt wird der Frame mit einem Blackman Fenster multipliziert und dann diskret Fourier transformiert. Da die Frequenzauflösung eines Frames nicht besonders hoch ist¹², ist es ein großer Vorteil, dass die Frequenzen der erwarteten Obertöne bereits ermittelt wurden, so dass in diesem Analyseschritt nur noch die Amplituden der Obertöne ermittelt werden müssen. Es reicht aus, die maximale Amplitude der Frequenz eines Obertons aus drei Bins zu ermitteln, dem Bin, welcher der Frequenz des erwarteten Obertons entspricht und seinen beiden direkten Nachbarn. Wäre die Suche *breiter*, so besteht die Gefahr, die Amplitude einer nicht erwünschten Nebenfrequenz zu messen und diese fälschlich als Frequenz bzw. Maximum des Obertons zu interpretieren. Durch die Suche nach dem Maximum in der unmittelbaren Nähe zu bereits bestimmten Obertönen wird die Wahrscheinlichkeit eines falschen Fundes minimiert.¹³

Die Nummer bzw. der Index des Bins gibt die Frequenz des Obertons an, entsprechend Formel 3.9.

$$f = \frac{\text{Bin\#} - 0.5}{\text{Framegröße}/2} \cdot \frac{\text{Samplerate}}{2} \quad (3.9)$$

$$A[x] = \sqrt{\text{Re}[x]^2 + \text{Im}[\text{Framegröße} - x]^2} \quad (3.10)$$

$$P[x] = \arctan\left(\frac{\text{Im}[\text{Framegröße} - x]}{\text{Re}[x]}\right) \quad (3.11)$$

Formel 3.10 zeigt wie das Ergebnis einer DFT in Amplitude und Phase einer Sinusfunktion transformiert werden kann. Dabei gilt $0 < x < \text{Framegröße}$.¹⁴

Für jeden Frame ergeben sich somit drei Vektoren je 20 Zahlen. Die jeweils i -ten Zahlen jedes Vektors bilden ein *3-Tupel*, mit dem sich die Obertonschwingung beschreiben lässt.

Diese Vektoren werden nun als Zeilenvektoren in die 3 Matrizen eingetragen. Die oberste Zeile entspricht dem ersten Frame, die folgenden Frames werden jeweils als neue Zeilen an die Matrix angehängt. Wie im vorigen Kapitel erläutert, wurden somit die Daten des Gesamtklangs auf 1.5% der ursprünglichen Anzahl reduziert.

¹² ≈ 10 Hz. In vorherigen Kapitel genauer erläutert.

¹³Der Algorithmus wird in 4.4 genauer erläutert.

¹⁴Für Index 0 und $\text{Framegröße}/2$ liefert die DFT keine Phase, hier ist $A[0] = \text{Re}[0]$. Näheres dazu in 2.4.

3 Design

506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2594.75	3133.08	3649.88	4166.67	4705
506.03	1044.36	1561.16	2077.95	2594.75	3133.08	3649.88	4166.67	4705
506.03	1044.36	1561.16	2077.95	2594.75	3133.08	3649.88	4166.67	4705
506.03	1044.36	1561.16	2077.95	2594.75	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705
506.03	1044.36	1561.16	2077.95	2616.28	3133.08	3649.88	4188.21	4705

Abbildung 3.7: Frequenzmatrix eines c5

53.8443	65.9115	32.1688	21.5412	20.4876	10.8822	2.8964	0.6466	0.3059	0.1944
52.9287	59.6621	28.6668	18.7019	18.5585	8.6261	2.8145	0.6938	0.1665	0.2755
50.1032	53.6063	22.0757	15.3144	15.6269	6.8131	2.8167	0.4690	0.1075	0.2244
47.4572	48.4006	15.2476	12.0662	12.8844	5.0415	3.0055	0.3583	0.1495	0.1347
44.9455	43.6292	9.4212	9.9607	10.5373	3.6589	2.9619	0.3592	0.1578	0.1113
42.4908	39.2642	4.9268	9.5516	8.6129	2.6495	2.6508	0.3603	0.1186	0.1479
40.2277	35.3355	2.0882	10.2387	7.0780	2.0522	2.3466	0.2886	0.0637	0.1746
38.1067	31.7398	1.2276	10.8561	5.7447	1.6478	2.1412	0.2366	0.0136	0.1617
36.1289	28.4074	1.8988	11.3722	4.6692	1.2312	1.9523	0.2081	0.0322	0.1668
34.2217	25.4917	3.0032	11.7808	3.8545	1.0437	1.7392	0.1849	0.0538	0.1621

Abbildung 3.8: Amplitudenmatrix eines c5

-0.068	-1.889	-1.185	-1.718	2.468	-2.346	-0.734	2.622	-1.849	1.266
1.733	1.726	-2.313	-0.995	2.081	2.096	-0.419	2.042	2.330	2.820
-2.844	-1.112	2.576	-0.780	-1.794	-0.096	-0.434	-2.032	0.109	2.464
-1.115	2.390	1.256	-0.520	0.844	-2.031	-0.265	0.709	-1.715	2.283
0.650	-0.289	0.083	-0.126	-2.549	2.597	0.111	0.568	2.585	-3.068
2.455	-2.873	-0.953	0.480	-2.532	1.137	0.731	-2.465	0.960	-2.150
-1.995	0.893	-2.053	1.376	0.769	-0.177	1.537	1.109	-0.676	-0.988
-0.141	-1.576	2.696	2.498	-2.064	-1.38	2.548	-1.331	-2.181	0.638
1.704	2.231	1.455	-2.623	1.416	-2.711	-2.679	2.349	0.455	2.182
-2.742	-0.261	0.640	-1.467	-1.411	1.965	-1.639	-0.494	-0.798	-2.674

Abbildung 3.9: Phasenmatrix eines c5



Die Abbildungen 3.7, 3.8 und 3.9 zeigen die ersten 10 Zeilen und linken 10 Spalten eines C (506 Hz) gespielt auf einer elektrischen Gitarre. Die Frequenzen in der Frequenzmatrix 3.7 entsprechen der Frequenz in der Mitte des Bins, in dem ein lokales Maximum erkannt wurde. Man kann erkennen, dass sich die Frequenzen von Zeile zu Zeile kaum ändern, was bedeutet, dass sich die Frequenzen der Obertöne in einem Maße verändern, welches geringer ist als die Frequenzauflösung eines Bins. Die Amplituden der Obertöne nehmen erwartungsgemäß mit der Zeit ab, was durch die kleineren Zahlen in den unteren Zeilen der Amplitudenmatrix 3.8 bestätigt wird. Die Amplitude der ersten Obertöne nimmt wie erwartet mit zunehmender Frequenz ab, so dass die linken Spalten höhere Werte haben als die rechten. Die Phasen der Obertöne sind eher zufällig zwischen $-\pi$ und $+\pi$ verteilt.

3.5 Erste Version der Synthese

Die drei Matrizen sind das Ergebnis des ersten Schritts der Analyse. Da die Anzahl der Daten erheblich reduziert wurde, könnte man auch von einer verlustbehafteten Kompression¹⁵ der Ausgangsdaten sprechen. Das bedeutet andererseits, dass aus den Matrizen nicht mehr auf triviale Weise auf die Ausgangsdaten geschlossen werden kann. Es kann zu diesem Zeitpunkt eine erste Version der Resynthese formuliert werden, die dem ersten Schritt der Analyse entspricht.

Der Ausgangspunkt für die Resynthese sind die Zeilen der Matrizen. Jede Zeile entspricht einem Frame; jeder Frame kann deshalb unabhängig von anderen Frames synthetisiert werden. Dabei werden pro Frame 4096 Samples erzeugt, so dass das Ergebnis der Resynthese ein großes Sample von der auf ein Vielfaches von 4096 aufgerundeten Länge des ursprünglichen Samples ist. Allerdings sind in den Matrizen keine Informationen über die Framegröße und Samplerate enthalten, so dass es auch möglich ist, die Resynthese mit einer anderen Samplerate durchzuführen. Eine Änderung der Samplerate ändert jedoch nicht unbedingt die Länge des Klangs. Bei einer Änderung der Framegröße kann jedoch der synthetisierte Klang eine neue Länge erhalten. Bei einer zu großen Änderung der Länge geht jedoch der Charakter des Gitarrenklangs verloren.

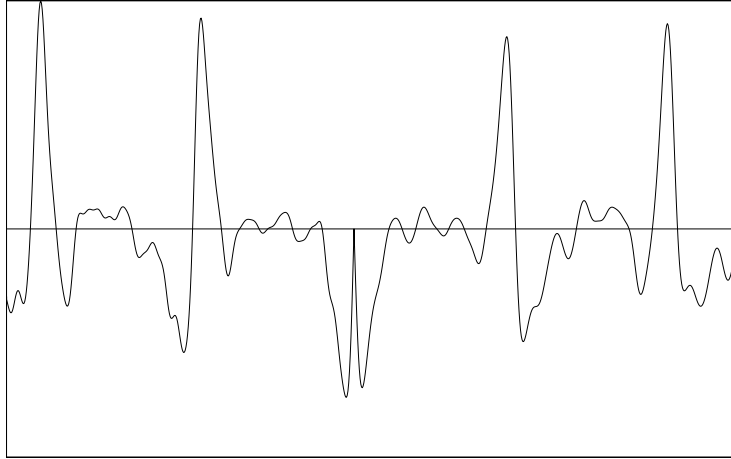
Die naheliegendste Möglichkeit, einen Frame zu synthetisieren, ist die Verwendung der Inversen Diskreten Fourier-Transformation. Dazu muss aus den 3 Vektoren für Frequenz, Amplitude und Phase des Frames das Eingabearray der IDFT konstruiert werden. Bei einer Framegröße von 4096 Samples müssen aus $3 \cdot 20$ Werten 4096 generiert werden. Die sinnvollste Möglichkeit ist es, alle nicht direkt bestimmbar Werte auf 0 zu setzen. Das Ergebnis dieser Synthese ist jedoch nicht befriedigend. Jeweils beim Übergang zwischen zwei Frames hört man deutlich ein Klicken. Da alle 0.1 Sekunden ein Frame beginnt ist der synthetisierte Klang von einem unangenehmen Klicken durchzogen.

KLANGBEISPIEL 7

Dieses ist andererseits zu erwarten. Denn die Synthese der einzelnen Frames erfolgt vollkommen unabhängig voneinander. Jeder Frame besteht aus der Summe von 20 Frequenzen mit unterschiedlichen Frequenzen, Amplituden und Phasen. Während des Fra-



¹⁵Im Sinne von Datenreduktion



In der Mitte der Abbildung beginnt ein neuer Frame, bei dem alle Obertöne einen Phasenwert von 0 haben. Der Samplewert ist deshalb ebenfalls 0 und erzeugt eine nicht stetige Stelle im Sample.

Abbildung 3.10: Ausschnitt eines Klangs ohne Phasensynchronisation

mes sind die entsprechenden Parameter konstant. Im nächsten Frame können sie jedoch unterschiedlich sein. Der Übergang zwischen den Frames erfolgt vollkommen unabhängig bzw. abrupt. Die Sinusfunktionen sind folglich an den Framegrenzen nicht stetig, was sich in einen Klicken bemerkbar macht. Um das Klicken zu vermeiden, müssen die Sinusfunktionen im gesamten Verlauf des Samples stetig sein. Dies ist mit der IDFT jedoch nicht zu erreichen.

Die Stetigkeit der Sinusfunktionen kann jedoch erreicht werden, indem der Frame sampleweise direkt berechnet wird. Der Frame wird mittels der additiven Synthese von Sinusfunktionen erzeugt. Die Stetigkeit zwischen zwei Frames wird erreicht, indem die drei Parameter jeder Sinusfunktion über den Frameverlauf interpoliert werden, so dass sie mit dem letzten Sample des Frames bereits den Parametern des nächsten Frames entsprechen. Die Parameter verändern sich also kontinuierlich. Da auch im ursprünglichen Klang die Parameter sich in Wirklichkeit permanent verändern, fällt diese fortwährende Veränderung der Parameter nicht negativ auf.

Abbildung 3.11 zeigt den C++ Sourcecode dieser Synthese. Die Hilfsfunktion *interpolate* berechnet eine lineare Interpolation zwischen den Werten *from* und *to*. Ist $v = 0$ so entspricht das Ergebnis *from*, für $v = 1$ ist das Ergebnis *to*. Mit dieser Hilfsfunktion werden die Parameter Frequenz, Amplitude und Phase für jeden Oberton im Verlauf des Frames interpoliert. Beim ersten Sample des Frames entspricht das Ergebnis der Interpolation genau den Werten der diesem Frame entsprechenden Vektoren¹⁶. Beim letzten Sample des Frames ist das Ergebnis der Interpolation der $\frac{n-1}{n}te$ ¹⁷ Teil der Parameter des nächsten Frames.

Die Funktion *synthframe1* berechnet die Samples eines Frames. Die Parameter *freq*,

¹⁶Zeilen der Matrizen

¹⁷ $n :=$ Framegröße

3 Design

```
1 double interpolate(double from, double to, double v)
  {
3   return (1.0-v)*from + v*to;
  }
5 const int FRAMESIZE=4096, SAMPLERATE=44100;
  typedef std::vector<double> v_t;
7 v_t lastfreq, lastamp, lastphase;
  v_t synthframe1(const v_t freq, const v_t amp, const v_t phase)
9 {
  v_t frame(FRAMESIZE);
11 for(unsigned i=0; i!=freq.size(); ++i)
    for(int f=0; f!=FRAMESIZE; ++f)
13     {
15         double f_=double(f)/double(FRAMESIZE);
        frame[f] += sin( 2.0 * M_PI *
17             double(f)/SAMPLERATE * interpolate(lastfreq[i],freq[i],f_)
                + interpolate(lastphase[i], phase[i], f_)
19             ) * interpolate(lastamp[i], amp[i], f_);
    }
  lastfreq=freq; lastamp=amp; lastphase=phase;
21 return frame;
  }
```

Abbildung 3.11: Synthese eines Frames mittels Frequenz, Amplitude, Phase

amp, *phase* sind Vektoren mit den Werten der Obertöne für diesen Frame. Sie entsprechen jeweils einer Zeile aus den Matrizen. Die äußere Schleife (Index i) berechnet jeden der 20 Obertöne. Die innere Schleife (Index f) berechnet den Samplewert des Frames an Stelle f . Man erkennt, dass die Parameter für die Sinusfunktion im Schleifenkörper bei jedem Sample aus dem Wert des letzten Frames und des aktuellen Frames interpoliert werden. Die Werte des letzten Frames werden in den globalen Variablen *lastfreq*, *lastamp*, *lastphase* gespeichert, welche jeweils am Ende der Funktion *synthframe1* auf die aktuellen Werte gesetzt werden. Beim ersten Frame werden *lastfreq*, *lastamp*, *lastphase* auf die Werte des ersten Frames gesetzt.¹⁸ Hier findet also effektiv keine Interpolation statt.

KLANGBEISPIEL 8 bestätigt, dass diese Kombination aus Analyse und Synthese ein gutes Ergebnis erzeugt. Es werden 5 verschiedene Gitarrentöne unterschiedlicher Tonhöhe jeweils im Original und als Resynthese gespielt. Die synthetisierten Klänge sind den originalen zwar ähnlich, klingen jedoch auch alle mehr oder weniger *schief*. Der Grund liegt in der Frequenzauflösung der DFT, wodurch die Frequenz der Obertöne nicht exakt, sondern nur mit einer Genauigkeit von ≈ 10 Hz bestimmt werden kann. Alle synthetisierten Klänge enthalten also *verstimmte* Obertöne. Dieses Manko wird jedoch in den folgenden Abschnitten behoben. Die synthetisierten Klänge enthalten ferner nur die Sustainphase.



3.6 Optimierung der Analyse

3.6.1 Phasenmatrix

Das menschliche Gehör kann zwei (gleichartige, periodische) Funktionen nur anhand von Frequenz und Amplitude unterscheiden. Sind Frequenz und Amplitude gleich klingen die Funktionen auch gleich.¹⁹ Ferner enthalten die Phasenwerte bei der Analyse mittels DFT keine musikalisch relevanten Informationen, wenn das analysierte Signal periodisch ist. Bei aperiodischen Signalen wird jedoch genau die aperiodizität durch die Ausprägung der Phasenwerte erzeugt.

Das bedeutet, dass die Information der Phasenmatrix für den wahrgenommenen Klang vollkommen unerheblich ist. Es ist kein Unterschied zwischen zwei Klängen feststellbar, wenn bei einem die Phasen entsprechend dem Ergebnis der Analyse synthetisiert werden, bei dem zweiten jedoch alle Phasen mit dem Wert 0 starten. Die Phasenmatrix 3.9 aus Abschnitt 3.4 bestätigt diese Vermutung. Die Phase verändert sich von Frame zu Frame und auch von Oberton zu Oberton zufällig und sprunghaft. Hätte dies eine Auswirkung auf die Wahrnehmung des Klangs, so sollte sich der Charakter des Klangs ebenfalls sprunghaft ändern. Dies kann aber nicht beobachtet²⁰ werden. Eine Veränderung der Phase erzeugt also keinen Unterschied in der Wahrnehmung.

Die Analyse kann folglich optimiert werden, indem sämtliche Werte der Phase einfach verworfen werden. Es müssen nur noch zwei Matrizen gespeichert bzw. konstruiert wer-

¹⁸im Pseudocode nicht gezeigt

¹⁹Nur unter der Voraussetzung, dass beide Töne nicht gleichzeitig gespielt werden, ansonsten gibt es Phasenauslöschungen, welche man sehr deutlich wahrnehmen kann.

²⁰oder besser ausgedrückt: gehört

```

v_t synthframe1(const v_t freq, const v_t amp)
2 {
    v_t frame(FRAMESIZE);
4   for(unsigned i=0; i!=freq.size(); ++i)
    {
6       double f_;
#define P 2.0 * M_PI * interpolate(lastfreq[i],freq[i],f_) \
8         * double(f)/SAMPLERATE + lastphase[i]
        for(int f=0; f!=FRAMESIZE; ++f)
10        {
            f_=double(f)/double(FRAMESIZE);
12            frame[f] += sin(P) * interpolate(lastamp[i], amp[i], f_);
        }
14        f_=1.0;
        lastphase[i]=std::fmod(P, 2.0*M_PI);
16    }
    lastfreq=freq; lastamp=amp;
18    return frame;
}

```

Abbildung 3.12: Synthese eines Frames mittels Frequenz und Amplitude

den. Bei der Resynthese wird beim ersten Frame jeder Oberton bei Phase 0 gestartet. Wegen der Stetigkeit der Sinusfunktionen muss aber ab dem zweiten Frame in Phase gearbeitet werden. In [Abbildung 3.12](#) ist der neue C++ Sourcecode zu sehen.²¹

3.6.2 Frequenzmatrix

Bei einem Vergleich der Zeilen in der Frequenzmatrix fällt auf, dass sich die Werte in den einzelnen Spalten kaum unterscheiden. Dies ist auch nicht weiter verwunderlich, denn die Frequenzauflösung innerhalb eines Frames beträgt ≈ 10 Hz und die Frequenzen der Obertöne verändern sich nur selten um mehr als 10 Hz, so dass sie von einem anderen Bin erfasst werden und damit unterschiedliche Werte in den Zeilen produzieren. In einem ersten Schritt zur Optimierung der Frequenzmatrix können alle Zeilen durch eine einzige ersetzt werden. Diese Zeile könnte aus dem arithmetischem Mittel der Zeilen gebildet werden. Diese Zeile bzw. Vektor wird dann für jeden Frame verwendet. Das [KLANGBEISPIEL 9](#) zeigt, dass sich der Klangeindruck beim Vergleich der Resynthese mit einer kompletten Frequenzmatrix und einer einzelnen Zeile kaum ändert.

Der verbleibende Frequenzvektor gibt also die durchschnittlichen Frequenzen der 20 Obertöne an. Da er aus den gemittelten Zeilen der Matrix erzeugt wurde, entspricht er dem Durchschnitt über den gesamten Klang. Es entspricht also den mittels der ersten DFT über das gesamte Sample ([Abschnitt 3.3](#)) ermittelten Obertönen, nur mit geringerer Frequenzauflösung. Das letzte Klangbeispiel hat jedoch gezeigt, dass die grobe Frequenzauflösung der frameweisen DFT vollkommen ausreicht, den originalen Klang

²¹Es wird nur die neue Synthesefunktion gezeigt, der Rest bleibt gleich wie bei 3.11.



```

1 v_t synthframe2(const double basefreq, const v_t amp)
  {
3   v_t freq;

5   for(unsigned i=1; i<=20; ++i)
       freq.push_back(basefreq*i);
7
   return synthframe1(freq, amp);
9 }

```

Abbildung 3.13: Berechnung der idealen Obertöne für die Resynthese

überzeugend zu resynthetisieren. Diese Erkenntnis eröffnet die Möglichkeit für eine weitere Optimierung. Ausgehend von der Grundfrequenz kann man gemäß [Formel 3.7](#) die idealen Obertöne berechnen. Diese weichen zwar von den tatsächlichen Obertönen ab, die Abweichung ist jedoch i. d. R. kleiner als die Frequenzauflösung der frameweisen DFT. Der Vektor der gemittelten Obertöne entspricht also dem Vektor der ersten 20 idealen Obertöne, wenn man beide mit der Frequenzauflösung der frameweisen DFT diskretisiert. Die Frequenzen der Obertöne können also nur aus der Grundfrequenz berechnet werden. Das [KLANGBEISPIEL 10](#) zeigt den Vergleich zwischen der Synthese mit voller Frequenzmatrix und berechneten Obertönen.



Werden die berechneten, und damit idealen bzw. theoretischen, Obertöne benutzt, so klingen die synthetisierten Klänge nicht mehr verstimmt oder *schief*. Bei der Benutzung der Obertöne der Frequenzmatrix unterliegt die Frequenz des einzelnen Obertons der Auflösung der DFT und weicht somit um bis zu 10 Hz von der realen Frequenz ab. Werden die Frequenzen der Obertöne jedoch berechnet, so erhalten wir einen reinen und unverstimmten Klang.

[Abbildung 3.13](#) zeigt den nächsten Schritt in der Resynthese. Die Frequenzmatrix wird nicht mehr benötigt. Die Frequenzen der Obertöne werden aus der Grundfrequenz des zu synthetisierenden Tons berechnet (Zeile 6). Damit wird für jeden Frame eine der Frequenzmatrix entsprechende Zeile bzw. Vektor erstellt, mit dem die Funktion *synthframe1* aufgerufen werden kann um die Samples zu berechnen.

Nach der durchgeführten Analyse mittels DFT kann die Frequenzmatrix komplett verworfen werden. Nur die Grundfrequenz des Tons muss gespeichert werden. Dies ist ein großer Vorteil zur Ermittlung der Klangparameter. Denn neben der weiteren Reduktion von zu speichernden Daten, die jetzt nur noch eine Matrix mit 20 Spalten und eine Grundfrequenz umfassen, ist das Analyseergebnis zumindest teilweise frequenzunabhängig geworden.

Benachbarte Töne, z. B. F und Fis, klingen bekanntlich besonders ähnlich. Die Ähnlichkeit bezieht sich sowohl auf die Tonhöhe, also den numerischen Wert der Grundfrequenz, als auch auf den *Klang*, also die Ausprägung der Obertöne. Man erwartet, dass sich die Obertöne von F und Fis im Grunde ähnlich verhalten, nur jeweils um den Unterschied in der Grundfrequenz von F und Fis in ihrer Tonhöhe verschoben. Dies kann nun experimentell überprüft werden, indem die Resynthese mit der selben Amplitudenmatrix und den

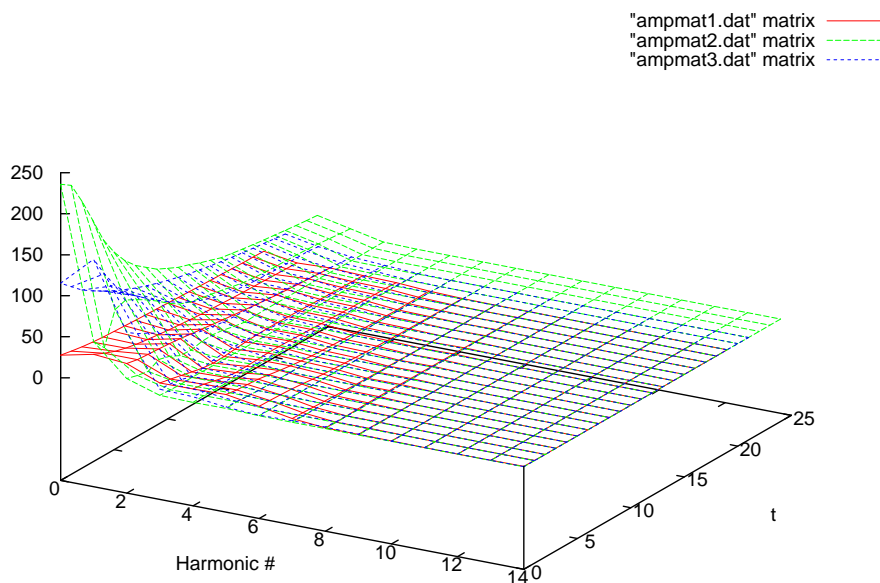


Abbildung 3.14: Amplituden Matrizen von drei Klängen

Grundfrequenzen von F und Fis mit der Resynthese der unterschiedlichen Amplitudenmatrizen verglichen wird (und auch mit den Originalaufnahmen). **KLANGBEISPIEL 11**

Leere Saiten²² sollten jedoch gesondert behandelt werden, also nicht mit den benachbarten Tönen zusammengefasst werden, da sie ein deutlich obertonreicheres Spektrum besitzen als die gegriffenen Töne. Die Saite kann an den starren Enden von Brücke und Hals deutlich besser bzw. freier schwingen (leere Saite) als auf dem Bundstab (gegriffene Saite). Der Finger hinter dem Bundstab liegt nicht perfekt unbeweglich auf dem Griffbrett, so dass die Saite auf dem Bundstab nicht perfekt starr aufliegt. Somit wirkt der Finger effektiv wie ein Tiefpaßfilter und gegriffene Saiten enthalten weniger intensive hohe Frequenzen als die leere Saite.

3.6.3 Amplitudenmatrix

Von den ursprünglich drei Matrizen konnte die Phasenmatrix komplett verworfen werden, die Frequenzmatrix wurde durch einen einzigen Wert ersetzt. Es ist offensichtlich, dass die Amplitudenmatrix nicht so stark reduziert bzw. optimiert werden kann, denn ansonsten bliebe nur ein einziger Wert übrig, die Grundfrequenz, mit dem sich nur eine einzige Sinusfunktion synthetisieren ließe.

²²Als leere Saite bezeichnet man die 6 Töne, welche beim Spielen einer Saite erzeugt werden ohne auf den Griffbrett zu greifen.



3 Design

Eine Betrachtung von verschiedenen Amplitudenmatrizen gibt jedoch einen Anhaltspunkt, wie diese weiter optimiert werden kann. In [Abbildung 3.14](#) sind drei typische Matrizen abgebildet. Man erkennt, dass die Werte in den Matrizen sich keinesfalls unkorreliert verhalten, sondern untereinander große Ähnlichkeit haben. Die maximalen Werte finden sich bei allen Matrizen in derselben Ecke. Ausgehend von dieser Ecke fallen die Werte rasch in beiden Dimensionen auf 0 ab. Man kann annehmen, dass sich ausgehend von der Ecke mit dem Wertemaximum die Werte mit einer von dem Maximum abhängigen Funktion in beiden Dimensionen nähern lassen. Diese Funktion ist zunächst unbekannt.

Sollten in der Matrix Werte innerhalb der Zeilen und Spalten miteinander korreliert sein, so kann man die der Korrelation zugrundeliegende Gesetzmäßigkeit oftmals nicht direkt erkennen. Es existieren viele Algorithmen bzw. Transformationen, welche eine andere Wertedarstellung der Matrix erzeugen bei der die Korrelation deutlicher erkennbar ist. Hierzu zählen neben der Diskreten Fourier-Transformation auch die Wavelet-Transformation und die *Diskrete Cosinus-Transformation* (DCT).

Die Werte der diskret cosinus transformierten Matrix stellen die Amplituden von Cosinusfunktionen unterschiedlicher Frequenz dar, jeweils in Spalten- und Zeilenrichtung. Von der linken oberen Ecke aus betrachtet besteht die transformierte Matrix aus einer Reihe von Cosinusfunktionen steigender Frequenz und variabler Amplitude. In der linken oberen Ecke befinden sich Cosinusfunktionen mit der kleinsten Frequenz in der rechten unteren Ecke die Funktionen mit der größten Frequenz. Ähnlich wie bei der Fourier-Transformation kann man das Ergebnis der DCT als ein Frequenzspektrum mit ausschließlich Cosinusfunktionen betrachten.²³

Je mehr die Werte innerhalb der Matrix korreliert sind, desto geringer ist der Anteil an hohen Frequenzen in der transformierten Matrix. Das typische Ergebnis der DCT ist, dass der numerische Wert von Zellen nahe der linken oberen Ecke²⁴ deutlich größer als 1 ist. Je mehr man sich der rechten unteren Ecke²⁵ nähert desto mehr streben die Werte der Zellen gegen 0. Je stärker die Werte der Ausgangsmatrix korreliert sind, desto mehr Zellen der transformierten Matrix enthalten Werte ≈ 0 .

$$c(i) = \begin{cases} 1 & : i = 0 \\ 2 & : \text{sonst} \end{cases}$$

$$dct[x, y] = \sqrt{\frac{c(x) \cdot c(y)}{X \cdot Y}} \sum_{n=0}^Y \sum_{m=0}^X mat[m, n] \cdot \cos\left(\pi(2m+1)\frac{x}{2X}\right) \cdot \cos\left(\pi(2n+1)\frac{y}{2Y}\right) \quad (3.12)$$

$$idct[x, y] = \sum_{n=0}^Y \sum_{m=0}^X mat[m, n] \cdot \sqrt{\frac{c(m) \cdot c(n)}{X \cdot Y}} \cdot \cos\left(\pi(2m+1)\frac{x}{2X}\right) \cdot \cos\left(\pi(2n+1)\frac{y}{2Y}\right) \quad (3.13)$$

²³Einzelheiten zur DCT finden sich in [Smi03] und [PM:].

²⁴tiefe Frequenzen

²⁵hohe Frequenzen

3 Design

1.6816	2.0620	1.4125	0.8334	0.5212	0.4144	0.3276	0.2092	0.0672	-0.0716
0.6421	0.7493	0.4368	0.1729	0.0618	0.0577	0.0256	-0.0645	-0.1792	-0.2418
0.2307	0.2545	0.1052	-0.0347	-0.1045	-0.1062	-0.0972	-0.0966	-0.0959	-0.0635
0.1318	0.1469	0.0663	-0.0122	-0.0556	-0.0536	-0.0360	-0.0196	-0.0126	-0.0065
0.0363	0.0376	0.0134	-0.0084	-0.0174	-0.0106	-0.0051	-0.0044	-0.0094	-0.0130
0.0293	0.0309	0.0133	-0.0003	-0.0044	-0.0010	-0.0003	-0.0020	-0.0071	-0.0114
0.0076	0.0071	0.0026	0.0001	-0.0002	-0.0004	-0.0028	-0.0056	-0.0083	-0.0093
0.0094	0.0104	0.0050	0.0012	0.0018	0.0017	-0.0014	-0.0043	-0.0064	-0.0064
0.0019	0.0017	-0.0004	-0.0027	-0.0005	0.0016	-0.0002	-0.0021	-0.0050	-0.0057
0.0035	0.0041	0.0016	-0.0013	-0.0007	0.0003	0.0001	0.0	-0.0022	-0.0029

Abbildung 3.15: Diskret Cosinus Transformierte Amplitudenmatrix

Die Formeln 3.12 und 3.13 zeigen die für die Analyse und Resynthese benutzte Variante der 2-dimensionalen DCT und IDCT²⁶. X und Y sind die Anzahl der Spalten und Zeilen der Matrix mat . x und y sind der Laufindex mit der jede Zelle der Matrix einmal berechnet wird. Durch die Verwendung der Hilfsfunktion c und des Skalierungsfaktors $\sqrt{\frac{c(x) \cdot c(y)}{X \cdot Y}}$ ergibt sich die Eigenschaft:

$$idct(dct(mat)) = mat$$

Abbildung 3.15 zeigt die diskret cosinus transformierte Matrix von der Amplitudenmatrix 3.8. Man erkennt, dass bei der transformierten Matrix der Großteil der unteren Zeilen Werte nahe 0 enthalten. Dies entspricht auch den Erwartungen. Die unteren Zeilen der Amplitudenmatrix enthalten die Amplituden der hinteren Frames des Klangs. Da die Amplituden der einzelnen Obertöne abnehmen, sind diese Werte i. d. R. kleiner als die Werte von früheren Frames bzw. oberen Zeilen. Die Amplituden der einzelnen Obertöne nehmen bekanntlich exponentiell ab, folglich sind die logarithmierten Differenzen zwischen den Zeilen ähnlich groß. Dies bedeutet aber, dass sich die gesuchte Funktion, welche das Verhalten der Obertöne bzw. der Amplituden in den Zeilen beschreibt, recht gleichförmig verhält; oder anders ausgedrückt als Summe von Cosinusfunktionen mit wenig hohen Frequenzen dargestellt werden kann. Da bei der DCT die hohen Frequenzen durch Werte unten und rechts dargestellt werden, sind die unteren Zeilen erwartungsgemäß ≈ 0 , denn wir erwarten keine hohen Frequenzen in der Darstellung der unbekanntenen Funktion.

Die DCT transformierte Amplitudenmatrix kann folglich auf die obersten zwei oder drei Zeilen reduziert werden. Bei der Resynthese werden die fehlenden Zeilen durch Nullvektoren ersetzt und die benötigte Amplitudenmatrix mit der IDCT berechnet. Das **KLANGBEISPIEL 12** zeigt den Vergleich einer auf 2 und drei Zeilen reduzierten Amplitudenmatrix jeweils mit dem Original.

Die Transformation mittels der DCT bringt nochmals eine erhebliche Reduktion der Daten bzw. den Klang beschreibenden Parameter. Für einen Klang sind nun nur noch



²⁶Inverse Diskrete Cosinus-Transformation

2 oder 3 Zeilen à 20 Spalten und eine Grundfrequenz erforderlich. Die ursprünglichen 100000 Samples werden nun durch 41–61 Werte dargestellt.

3.7 Variable Syntheselänge

Die bislang vorgestellten Algorithmen zur Resynthese haben zwar die Möglichkeit geboten, die Grundfrequenz und das Spektrum des Klangs beliebig zu bestimmen, jedoch war die Länge des synthetisierten Klangs immer abhängig von den Größen der benutzten Matrizen. Möchte man die Resynthese nicht nur zur Kontrolle der Analyse- oder Resyntheseverfahren verwenden, sondern damit einen Synthesizer erstellen, mit dem man tatsächlich (musikalisch) spielen kann, so muss dieser in der Lage sein Klänge beliebiger Länge zu erzeugen.

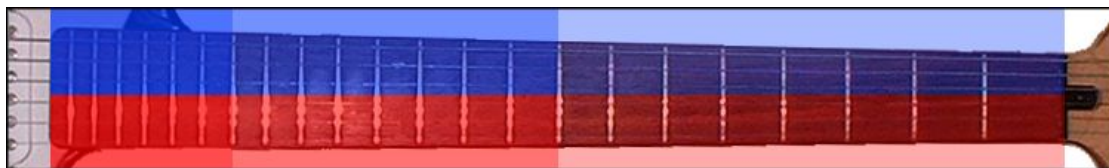
Durch das bereits beschriebene Verfahren der Synthese haben alle synthetisierten Klänge die gleiche Länge. Diese ergibt sich aus der Größe der quadratischen Matrix, welche für die IDCT verwendet wird und der Framegröße zur additiven Synthese der Sinusfunktionen. Möchte man einen kürzeren Ton synthetisieren, so ist dies relativ einfach möglich. Man kann einfach den synthetisierten Klang mit einer linearen Funktion multiplizieren, die beim Zeitpunkt 0 den Wert 1 hat und beim Zeitpunkt des Endes des Tons den Wert 0. Dies ist nur möglich, wenn die gewünschte Länge bereits am Anfang der Synthese feststeht. Ist dies nicht der Fall, z. B. bei einem interaktiven Spiel auf einen Keyboard, so kann zunächst der Klang ganz normal synthetisiert werden und beim Ende wird dann relativ schnell ausgeblendet. Dies ergibt ein sehr wirklichkeitsgetreues Verhalten, denn bei einer echten Gitarre schwingt die Saite nach dem vorzeitigen Abdämpfen ebenfalls noch für eine kurze Zeit nach.

Um den synthetisierten Klang länger klingen zu lassen als durch die Dimension der IDCT ermöglicht wird, müssen an die berechnete Amplitudenmatrix zusätzliche Zeilen angefügt werden. Mit diesen Zeilen können dann weitere Frames berechnet werden. Die zusätzlichen Zeilen sollten möglichst gut dem *Charakter* der vorherigen Zeilen entsprechen. Betrachtet man die Amplitudenmatrix spaltenweise, dann entsprechen die Werte in den Spalten von oben nach unten ungefähr einer exponentiellen Funktion.²⁷ Diese Werte lassen sich durch eine exponentielle Regression sehr gut fortsetzen, da die Werte gegen 0 streben.²⁸ Die Amplitudenmatrix lässt sich also spaltenweise durch die Regression beliebig *verlängern* wodurch sich beliebig lange Klänge synthetisieren lassen. Da die *Spaltenfunktionen* jedoch gegen 0 streben sind diese Klänge nicht beliebig lange hörbar. Der maximale noch hörbare synthetisierte Klang ist i. d. R. nicht wesentlich länger als der gleichartige gespielte Klang. [Abbildung 5.1](#) zeigt unter anderem die Amplituden der ersten 8 Obertöne eines Klangs.

Im [Anhang B](#) wird der Algorithmus zur Berechnung neuer Matrixzeilen genauer erläutert.

²⁷wurde genauer im vorigen Abschnitt erläutert.

²⁸Interpolationsmethoden sind hierfür nicht geeignet, da sie Funktionen liefern, die optimiert sind, Funktionswerte zwischen zwei gegebenen Werten zu berechnen. In diesem Fall sind jedoch Werte außerhalb der vorgegebenen Wertemenge zu berechnen, was Funktionen die mittels Regression ermittelt wurden besser bewerkstelligen.



Die Saitengruppe der tiefen drei Saiten ist blau, die hohe Saitengruppe rot. Jeweils 8 Halbtöne werden in einer Bundgruppe zusammengefasst, die durch unterschiedlich transparente bzw. helle Blau- und Rot-Töne dargestellt sind. Die letzte Bundgruppe umfasst nur 6 Halbtöne, da dieses Griffbrett nur 22 Bünde hat. Durch die Interpolation können aber auch die *fehlenden* zwei Halbtöne pro Saite synthetisiert werden.

Abbildung 3.16: Einteilung des Griffbretts in 6 Gruppen

3.8 Arrangement der Synthesedaten

Im Abschnitt 3.6.2 wurde gezeigt, dass sich die für die Resynthese relevanten Daten zwischen zwei benachbarten Halbtönen hauptsächlich im Wert für die Grundfrequenz unterscheiden. Dies bedeutet, dass man durchaus zwei benachbarte Halbtöne mit dem gleichen Parametersatz synthetisieren kann, und nur die Grundfrequenz des Tons entsprechend wählen muss. Dies gilt auch für Tonabstände von mehr als einem Halbton, wobei mit wachsendem Tonabstand auch der Klang unterschiedlicher wird.

Beim Design eines Syntheseprogramms, welches den gesamten Tonumfang einer Gitarre bereitstellen soll, muss man sich überlegen, wie die $6 \times 24 = 144$ Positionen auf dem Griffbrett bzw. spielbaren Töne durch eine möglichst kleine Anzahl von Parametersätzen erzeugt werden können. Dazu bietet es sich zunächst einmal an, das Griffbrett in eine Anzahl von Zonen zu unterteilen. Die tiefen drei Saiten²⁹ sind i. d. R. umwickelte Stahlsaiten. Die hohen drei Saiten bestehen bei einer akustischen Konzertgitarre aus Nylon und bei akustischen Westerngitarren und elektrischen Gitarren aus einfachem Stahldraht. Jeweils die hohen und tiefen 3 Saiten wären geeignet, zusammengefasst als eine *Saitengruppe* für die Synthese benutzt zu werden. Für eine detailliertere Synthese kann auch jede Saite einzeln oder eine andere Kombination von *Saitengruppen* benutzt werden. In horizontaler Richtung sollte das Griffbrett in mindestens drei Abschnitte unterteilt werden, d.h. jeweils 8 Halbtöne werden durch einen Parametersatz synthetisiert. Für eine verbesserte Synthese können auch 4, 5 oder 6 Abschnitte gewählt werden. Diese Abschnitte werden dann als *Bundgruppen* bezeichnet.³⁰

Im einfachsten Fall ergeben sich somit $2 \times 3 = 6$ Parametersätze für die Synthese einer Gitarre. Der einzelne Parametersatz wird dabei aus den von der Aufnahme analysierten Halbtönen der jeweiligen Gruppe gewonnen. Es bietet sich an, einfach die einzelnen Analyseergebnisse zu mitteln. Ob dabei das arithmetische Mittel oder z. B. der Median bessere Ergebnisse bringt, kann nicht allgemein gesagt werden, dies muss für jede Gitarre durch Ausprobieren und Überprüfen des Syntheseergebnisses einzeln entschieden werden.

Soll nun ein bestimmter Ton synthetisiert werden, so muss zunächst die Position des

²⁹E, A und D bei der normalen Stimmung

³⁰Saiten- und Bundgruppen sind von mir für diese Arbeit eingeführte Begriffe.

Tons auf dem Griffbrett bestimmt werden. Ein Ton, also ein Klang mit einer gegebenen Grundfrequenz, kann auf bis zu 6 Positionen auf dem Griffbrett gespielt werden, abhängig von der Anzahl der Bünde prinzipiell auf jeder Saite ein Mal. Für die Griffbrettposition wird dann die entsprechende Saiten- und Bundgruppe ermittelt. Mit dem Parametersatz für die Saiten- und Bundgruppe kann dann mit der Grundfrequenz des Tons der Klang synthetisiert werden.

Um den Effekt abzuschwächen, dass zwei benachbarte Halbtöne bzw. Griffpositionen, die in zwei unterschiedlichen Gruppen liegen, sich stärker unterscheiden als zwei Halbtöne aus derselben Gruppe, können die Parametersätze zwischen zwei Gruppen interpoliert werden. Dann werden aus 6 Parametersätzen wieder 144 Sätze interpoliert, für jede Griffbrettposition einer.

Natürlich kann prinzipiell jede Grundfrequenz für die Synthese verwendet werden, also auch Frequenzen die unterhalb des tiefen E (82 Hz) oder des hohen e (1320 Hz) liegen. Auch Frequenzen zwischen zwei Halbtönen sind möglich, Viertel- oder Dritteltöne. Damit läßt sich der Synthesizer im Gegensatz zur echten Gitarre auf jede beliebige Stimmung einstellen und ist nicht auf die wohltemperierte zwölftönige Stimmung fixiert.

3.9 Analyse der Attackphase

Die Attackphase wurde bislang vollkommen ignoriert. Weder bei der Analyse, noch bei der Resynthese wurde die Attackphase gesondert berücksichtigt, obwohl im letzten Kapitel vermutet wurde, dass sich die Attackphase grundlegend von der Sustainphase unterscheidet. Dies stimmt natürlich auch, aber wie so oft bei der Analyse von Signalen ist die Interpretation auf viele Weisen möglich.

Die Sonogramme in [Abschnitt 2.2](#) zeigten, dass die Attackphase im Gegensatz zur Sustainphase deutlich mehr hohe Frequenzen enthält. Die zu klärende Frage ist nun, woher diese Frequenzen stammen. Die Vermutung im vorherigen Kapitel war, dass beim Anschlag nicht harmonische Frequenzen auf der Saite auftreten, welche jedoch nach kurzer Zeit von den harmonischen Schwingungen der Obertöne abgelöst werden.

Ein ähnliches Bild ergibt sich jedoch, wenn mehrere Sinusfunktionen zum gleichen Zeitpunkt mit Phase 0 starten. In diesem Fall ergibt sich eine Flanke im Signal, die sich bei der DFT als entsprechend hochfrequente und unharmonische Frequenzen zeigt.³¹ Es kann somit durch eine Analyse mit DFT nicht genau ermittelt werden, wie die hochfrequenten Anteile der Attackphase entstehen.

Beim Vergleich von synthetisierten Tönen und Original zeigt sich jedoch, dass die Synthese nicht nur die Sustainphase gut nachbilden kann, sondern auch den Anfang des Tons überzeugend modelliert. Allerdings ist der Anfang qualitativ nicht ganz so gut wie die Sustainphase. Es gibt also neben den durch den Start der Obertöne produzierten hohen Frequenzen noch weitere Ursachen.

³¹Die steilste Flanke hat die (periodische) Rechteckfunktion, welche der Summe von allen Obertönen mit gleicher Amplitude entspricht.

3.10 Zusammenfassung

In diesem Kapitel wurde die im letzten Kapitel entwickelte Vorgehensweise zur Analyse und Synthese von Gitarrenklängen dargestellt. Zunächst wurde eine Methode entwickelt, mit der ein Klang in Attack- und Sustainphase zerlegt werden kann, damit beide Phasen getrennt untersucht werden. Danach wurde gezeigt, wie die Sustainphase mittels der Diskreten Fourier-Transformation auf die spektrale Verteilung der Obertöne untersucht werden kann. Die Anzahl der für die Synthese verwendeten Obertöne wurde danach durch die Verwendung der Diskreten Cosinus Transformation weiter reduziert.

Jedem Schritt in der Analyse, der letztendlich eine Datenreduktion der originalen Aufnahme ist, entspricht ein Schritt in der Synthese des Klangs, welcher der Datenreduktion entgegensteht.

4 Realisierung

In diesem Kapitel werden die Besonderheiten vorgestellt, welche sich bei der Umsetzung der in den letzten beiden Kapiteln entwickelten Ideen ergeben. Des weiteren stelle ich meine Arbeitsumgebung vor, mit der ich die Aufnahmen erstellt habe. Als letztes wird der grobe Aufbau des für diese Arbeit entwickelten Source Codes vorgestellt, so dass der Interessierte die relevanten Stellen schnell im Code auffinden kann.

4.1 Verwendete Hard- und Software

Die Software für Analyse und Synthese wurde in C++ auf einem [32-bit x86 GNU/Linux](#) System entwickelt. Zur grafischen Ausgabe von Spektrogrammen und Oszilloskopen wird [QT](#) benutzt. Die Aufnahme und Ausgabe von Sound erfolgt mittels [Jack](#). Der Synthesizer wird mittels [MIDI](#) durch die [Alsa Lib](#) gesteuert.

Die Aufnahmen für die Klangbeispiele wurden mit einer [M-Audio Delta 1010](#) gemacht. Diese Karte kann bis zu 24 bit und 96 KHz verarbeiten, die Aufnahmen wurden in diesem Fall in 24 bit und 44.1 KHz gemacht. Die 24 bit Aufnahmen wurden immer im 32 bit float .wav Format gespeichert.

Die Signale der elektrischen Gitarren wurden von einem [Line6 POD Pro](#) auf *Line Level* konvertiert und dann mit einem analogen Eingang der Delta 1010 aufgenommen. Der digitale Ausgang des POD Pro kann dazu leider nicht genutzt werden, da der POD nicht so eingestellt werden kann, dass das Eingangssignal unverändert auf dem digitalen Ausgang anliegt. Dies ist erst mit dem Nachfolgemodell POD Pro XT möglich, welches mir nicht zur Verfügung stand.

4.2 Besonderheiten bei der Umsetzung des Designs

Das im letzten Kapitel entwickelte Design einer Software bzw. der benötigten Algorithmen lässt sich ohne größere Schwierigkeiten in C++ umsetzen. Dabei habe ich zwei getrennte Programme erstellt, eines zur Analyse und eines zur Synthese.

Mit dem Analyseprogramm kann man in Echtzeit das Spektrogramm und die Oszilloskopdarstellung eines Audio Signals betrachten. Dies dient u. a. zur Überprüfung, ob das Setup für eine Aufnahme korrekt ist und somit brauchbare Audiodaten am Eingang der Soundkarte anliegen. Hier ist vor allem die korrekte Aussteuerung, also die Einstellung der Lautstärke des Eingangssignals wichtig, damit bei der Zerlegung der Aufnahme in Einzeltöne die dB Grenzen¹ erreicht werden. Es kann dann eine beliebig lange Aufnahme gemacht werden, wobei ein Monokanal aufgenommen wird. Die Aufnahme kann dann zur

¹siehe 3.1

4 Realisierung

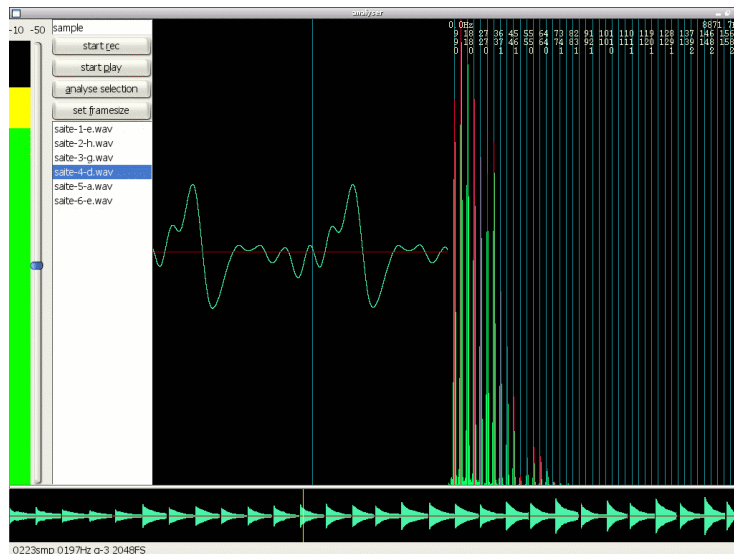


Abbildung 4.1: Screenshot des Analyseprogramms

Kontrolle abgehört und dauerhaft gespeichert werden. Ist man mit der Aufnahme zufrieden, kann diese analysiert werden. Dabei wird für jeden erfolgreich separierten Einzelton eine Datei gespeichert, welche das Analyseergebnis enthält. Diese Datei ist eine ASCII Datei im [Matlab](#) Format.²

Die Matlabdateien können dann vom Syntheseprogramm direkt abgespielt werden. Dabei wird der Klang zunächst mit der gleichen Grundfrequenz wie der Originalklang abgespielt. Dies dient zur manuellen Kontrolle, ob bei der Analyse des Klangs auch ein plausibles Ergebnis erreicht wurde. Denn trotz der mehrstufigen Prüfung während der Analyse, ob der sich der Originalklang zur Analyse eignet, sind einige Analyseergebnisse nicht brauchbar. Welche Eigenschaften des Samples das Analyseergebnis unbrauchbar machen ist noch nicht geklärt.

Brauchbare Matlabdateien müssen dann manuell in die Saiten- und Bundgruppen eingeteilt werden. Die Dateien der Gruppen werden dann in einem zweiten Analyseschritt vom Analyseprogramm gemittelt und das Ergebnis in einer neuen Matlabdatei gespeichert. Diese neue Datei enthält dann den Parametersatz für eine Gruppe.

Werden die Gruppdateien in das Syntheseprogramm geladen, so startet das Programm den interaktiven Modus. Nun können über MIDI die Klänge synthetisiert werden und direkt auf der Soundkarte ausgegeben werden. Damit ist dann ein echtes *Spielen* des Synthesizers möglich.

²Bei der Entwicklung der Algorithmen war es nützlich, die Ergebnisse meiner Programme in Matlab bzw. [GNU Octave](#) verwenden zu können. Deshalb habe ich einige einfache Routinen geschrieben, welche ASCII Dateien mit der Matlab Syntax für Variablen, Vektoren und Matrizen lesen und schreiben können.

4.3 Besonderheiten bei einzelnen Algorithmen

Bei der für die Synthese verwendeten Inversen Diskreten Cosinus Transformation werden oftmals negative Zahlen erzeugt. Dies sind Rechenfehler, die entstehen, da große Teile der DCT transformierten Matrix durch 0 ersetzt werden, womit keine exakte Rücktransformation mehr möglich ist. Da es nicht sinnvoll ist eine negative Intensität bzw. Amplitude einer Frequenz anzugeben, werden negative Matrixzellen auf 0 gesetzt.

Die exponentielle Regression zur Berechnung neuer Matrixzeilen hat Schwierigkeiten, wenn Werte innerhalb einer Spalte 0 sind, denn die Funktion $c \cdot e^x$ kann nur 0 werden bei $c = 0$. In diesem Fall ist das Ergebnis jedoch gleich der konstanten Funktion $f(x) = 0$. Um bei der Regression dennoch eine brauchbare exponentielle Funktion zu erhalten, ist es sinnvoll Zellen mit 0 durch eine kleine positive Zahl zu ersetzen, beispielsweise die kleinste Zahl innerhalb der Spalte.

Aber auch wenn alle Zellen der Spalte positive Werte größer 0 sind, kann das Ergebnis der Regression unbrauchbar sein, da durch die Berechnung mit floating point Zahlen die (interne) Darstellungsgenauigkeit der FPU unterschritten wurde. In diesem Fall ist das Ergebnis die float Zahl NaN.³ Dies bedeutet in diesem Fall, dass das Ergebnis eine so kleine Zahl ist, dass sie nicht mehr als float dargestellt werden kann. Die Ergebnisse der Regression sind also vor der weiteren Verwendung auf NaN zu überprüfen und in diesem Fall die entsprechende Variable auf 0 zu setzen.

4.4 Aufbau des Source Codes

Ein Sample wird von den Funktionen in `analyse.cpp` analysiert und das Ergebnis für jeden gefundenen Ton in einer Matlabdatei gespeichert.

analyse() separiert ein (großes) Sample in Einzeltöne mittels der dB Schwellenwerte. Jeder Einzelton wird mit `analyseSingle()` analysiert.

analyseSingle() analysiert einen Klang, als Ergebnis wird eine Matlabdatei geschrieben, welche u. a. Grundfrequenz des Klangs und die diskret cosinus transformierte Amplitudenmatrix enthält. Dazu wird zunächst die Grundfrequenz des Klangs mittels Autokorrelation bestimmt. Aus der Grundfrequenz werden die idealen Obertöne durch `expected()` berechnet und die realen Obertöne (gemittelt für den gesamten Klang) mit `findharmonics()` bestimmt. Danach wird die Amplitudenmatrix der Sustainphase erzeugt und die transformierte Matrix gespeichert.

expected() Diese Funktion berechnet die Frequenzen der idealen Obertöne einer Grundfrequenz bis die Nyquist-Frequenz erreicht ist. Rückgabewert ist ein array mit den Frequenzen der Obertöne.

fft_freq2bin() berechnet die durchschnittliche Frequenz, die einem Bin entspricht. Siehe Formel 3.8.

³Not a Number

fft_bin2freq() berechnet die Nummer eines Bins, welche der Frequenz entspricht. Siehe Formel 3.9.

findharmonics() sucht Obertöne innerhalb eines Periodogramms. Rückgabewert ist ein array mit den Bin-Nummern, welche die maximale Amplitude in der Umgebung der zu suchenden Obertöne hat.

Liegt bereits eine Aufnahme vor, so kann diese mit dem Programm **extract** analysiert werden. Dieses ist lediglich eine triviale `main()` Funktion, welche `analyse()` für jede Aufnahme aufruft.

Die Funktionen für die Resynthese befinden sich in `synthesizer.cpp`. Das Programm **synthesizer** erlaubt die Synthese eines Klangs aus einer Matlabdatei, welcher als Sample gespeichert wird. Alternativ kann das Programm als interaktiver MIDI Synthesizer benutzt werden.

linearinterpolation() berechnet einen linear interpolierten Zwischenwert. Rückgabewert ist der interpolierte Wert. Siehe [Abbildung 3.11: interpolate\(\)](#).

renderframe() berechnet einen Syntheseframe. Die erforderlichen Parameter werden linear interpoliert. Siehe [Abbildung 3.11: synthframe1\(\)](#).

calharmonics() berechnet die ersten n Obertöne für eine Grundfrequenz. Rückgabewert ist ein array mit den Frequenzen der Obertöne.

synthesize() synthetisiert einen Klang. Dazu werden die Parameter des Klangs von der Matlabdatei gelesen und die neue Amplitudenmatrix mittels IDCT generiert. Um die Resynthese eines längeren Klangs zu ermöglichen werden die Koeffizienten der exponentiellen Regression bestimmt. Danach wird der Klang mit der gewünschten Länge berechnet und als Sample gespeichert.

interactive() implementiert den interaktiven MIDI Synthesizer. Dieser wartet auf Note-On und -Off MIDI Ereignisse und ruft `synthesize()` auf um die gewünschte Tonhöhe und Länge zu synthetisieren.

Die restlichen Sourcecode Dateien implementieren das **analyse** Programm, mit dem sich Klänge aufnehmen lassen und die Sampledaten in Echtzeit in der Oszilloskop- und Spektrumansicht betrachten lassen (siehe [Abbildung 4.1](#)).

5 Ausblick

Mit dem im vorherigen Kapitel vorgestellten Verfahren lässt sich die Sustainphase eines Gitarrenklangs gut resynthetisieren. Die Interpretation der zur Synthese notwendigen Parameter, hinsichtlich ihrer Charakteristik den Klang zu beschreiben, ist jedoch schwierig, da durch die DCT die Parameter in einem nicht intuitiven mathematischem Raum liegen. Die numerischen Werte der transformierten Matrix sind nämlich nicht mehr voneinander unabhängig. Ändert man den Wert einer Zelle, so wirkt sich diese Änderung bei der IDCT auf alle Zellen der transformierten Matrix aus. Es lassen sich bei der Untersuchung von mehreren diskret cosinus transformierten Amplitudenmatrizen unterschiedlicher Gitarren kaum Gemeinsamkeiten oder Unterschiede zwischen den Matrizen erkennen. Die Vorgestellte Methode ist somit brauchbar um von einer Aufnahme einer Gitarre die Parameter für einen *Synthesizer* zu ermitteln, lässt jedoch keine Vergleichenden Aussagen zu.

Die DCT ist eine Transformation um eine Menge von Werten beliebiger Dimension durch eine Menge von Cosinusfunktionen darzustellen, ähnlich wie die DFT die Werte durch Sinus- und Cosinusfunktionen bzw. Sinusoiden + Phase darstellt. Eine Verallgemeinerung dieses Ansatzes, bei dem beliebige periodische Funktionen zur Darstellung verwendet werden ist die *Wavelet-Transformation*. Kennt man die den Werten der Matrix zugrundeliegende Funktion, so kann man mittels der Wavelet-Transformation und der Wahl eines geeigneten *Wavelets* eine Transformation berechnen in der nur wenige Werte ungleich 0 sind. Die Interpretation der Wavelet-Transformation ist jedoch ähnlich wie bei der DCT schwierig.¹

Abbildung 5.1 zeigt wie gut die Amplituden der Obertöne eines typischen Klangs durch die DCT+IDCT rekonstruiert werden. Zusätzlich werden die exponentiellen Regressionen der rekonstruierten und originalen Obertonamplituden dargestellt. Man sieht z. B. beim dritten Oberton, dass der rekonstruierte Oberton nur wenig Gemeinsamkeit mit dem Original besitzt. Die Regression der originalen Amplituden stellen in allen Fällen die beste Approximation dar. Ein anderer Ansatz wäre folglich, die Amplituden der Obertöne durch die Parameter einer exponentiellen Regression zu beschreiben. Dies würde zwei Parameter pro Oberton ergeben, genauso viele wie bei der Analyse mit DCT. Der Vorteil ist jedoch, dass die Parameter der einzelnen Obertöne unabhängig voneinander sind. Die Parameter der Regression lassen sich in einem weiteren Schritt auf mögliche Korrelationen bzw. Gemeinsamkeiten untersuchen. Gibt es Gemeinsamkeiten so lässt sich eventuell in einem weiteren Schritt die Anzahl der Parameter nochmals reduzieren oder es lassen sich durch den Vergleich unterschiedlicher Klänge Aussagen über die Charakteristik unterschiedlicher Gitarren machen.

¹Die DCT kann man als einen Spezialfall der allgemeineren Wavelet-Transformation betrachten.

5 Ausblick

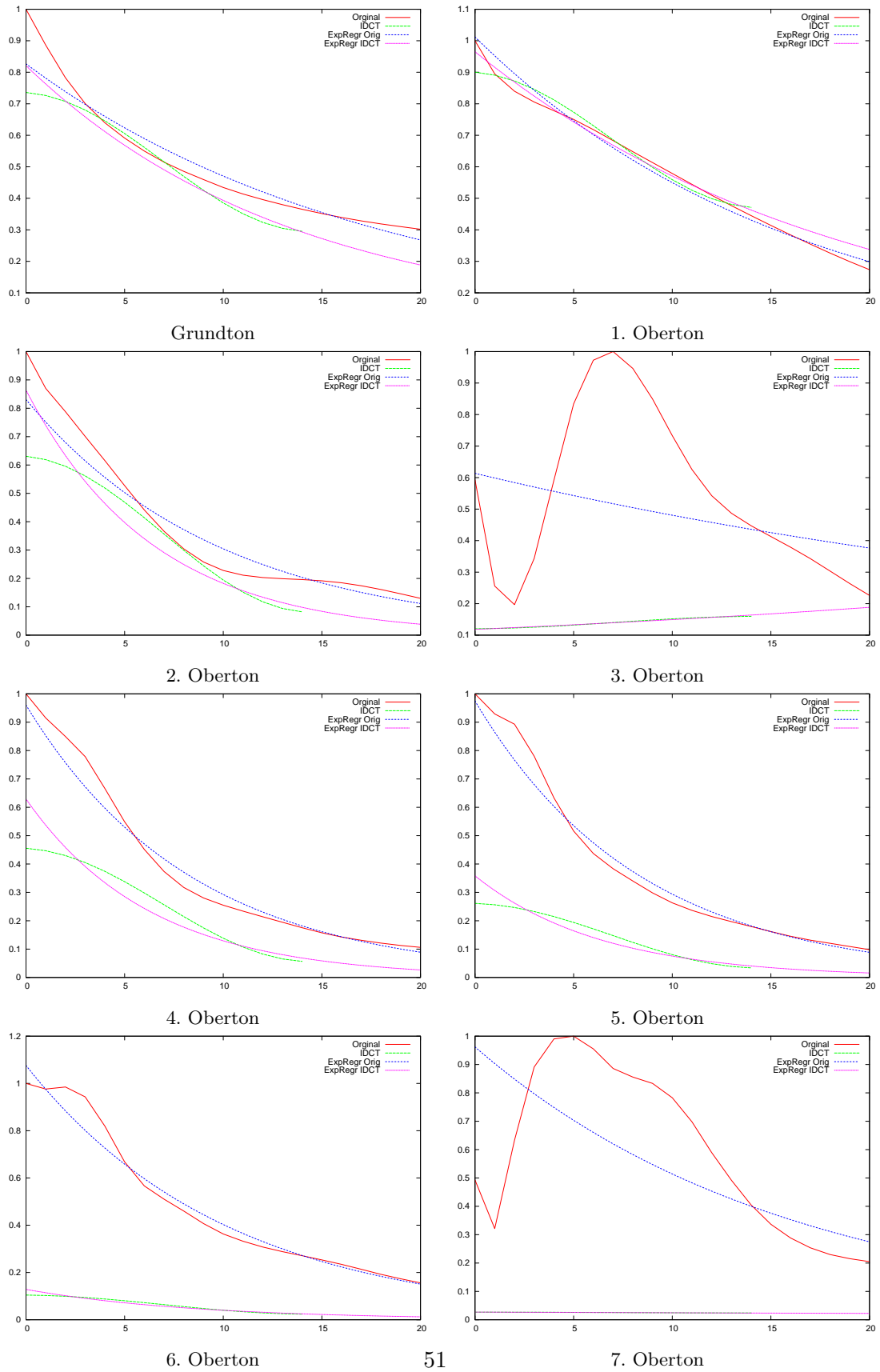


Abbildung 5.1: Amplituden der ersten 8 Obertöne

5 Ausblick

Eine gesonderte Analyse und Resynthese der Attackphase wurde bislang nicht vorgenommen. Trotzdem erzeugt die Resynthese ein brauchbares Ergebnis, wenn die ursprüngliche Attackphase keinen ausgeprägten inharmonischen Frequenzanteil besitzt, z. B. bei einer elektrischen Solid-Body-Gitarre. Für andere Gitarrentypen, insbesondere akustische, sollte die Attackphase gezielt analysiert werden und durch eine eigene Resynthese reproduziert werden.

Die Attackphase hat nur eine sehr kurze Länge von 3 bis 6 Perioden innerhalb derer sich das Spektrum stark verändert. Zur Analyse der Attackphase ist eine genaue spektrale Untersuchung erforderlich. Dazu sollten zunächst die Perioden der Attackphase genau erfasst werden. Dann kann jede Periode einzeln durch eine DFT untersucht werden. Zur Verbesserung der Frequenzauflösung kann das Sample einer Periode durch Zero-Padding vor der DFT erweitert werden.

Literaturverzeichnis

- [BR02] BOYER, Rémy ; ROSIER, Julie: Iterative method for harmonic and exponentially damped sinusoidal Models. In: *Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02)* 1.1
- [Coo02] COOK, Perry R.: *Real Sound Synthesis for Interactive Applications*. AK Peters <http://www.cs.princeton.edu/~prc/AKPetersBook.htm>. – ISBN 1-56881-168-3 8, 1, 5
- [Ero01] ERONEN, Antti: *Automatic Musical Instrument Recognition*, Tampere University of Technology, Masters Thesis, 2001. http://www.cs.tut.fi/sgn/arg/music/eronen/eronen_msc.pdf. – Online-Ressource 1.1
- [EVKL00] ERKUT, Cumhur ; VÄLIMÄKI, Vesa ; KARJALAINEN, Matti ; LAURSON, Mikael: Extractino of Physical and Expressive Parameters for model-based Sound Synthesis of the Classical Guitar. In: *AES108th Convention, Paris* 1.1
- [Fou04] FOUAD, Hesham: Understanding the Decibel. Version:2004. <http://audioanecdotes.com>. In: *Audio Anecdotes* Bd. 1. AK Peters, Ltd, Kapitel 2. – ISBN 1-56881-104-7 3, 2
- [FR98] FLETCHER, Neville H. ; ROSSING, Thomas D.: *The Physics of Musical Instruments*. 2nd ed. Springer, 1998. – ISBN 0-387-98374-0 1.1, 8
- [GB04] GREENEBAUM, Ken ; BARZEL, Ronen: *Audio Anecdotes*. Bd. 1. AK Peters <http://audioanecdotes.com>. – ISBN 1-56881-104-7 D
- [Hot94] HOTOP, Hans-Jürgen: *Numerische Methoden*. 2. veränderte Auflage. Wißner <http://users.etech.haw-hamburg.de/users/hotop/veroffen.htm>. – ISBN 3-928898-62-0 B
- [Jen99] JENSEN, Kristoffer: *Timbre Models of Musical Sounds*, University of Copenhagen, Masters Thesis, 1999. <http://www.diku.dk/musinf/krist/TMoMS.pdf>. – Online-Ressource 1.1
- [Kat02] *Kapitel 5*. In: KATZ, Bob: *Mastering Audio*. Focal Press. – ISBN 0-240-80545-3, <http://www.digido.com> 2
- [Knu99] KNUTH, Donald E.: *Digital Typography*. Bd. 78. Stanford, California: Center for the Study of Language and Information : CSLI Lecture Notes, 1999. – ISBN 1-57586-010-4 D

Literaturverzeichnis

- [KR02] KAUPPINEN, Ismo ; ROTH, Kari: Audio Signal Extrapolation - Theory and Applications. In: *Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02)* 1.1
- [KS83] KARPLUS, Kevin ; STRONG, Alex: Digital Synthesis of Plucked String and Drum Timbres. In: *Computer Music Journal* 7 (1983), Nr. 2, S. 43–44 1.1
- [KVJ93] KARJALAINEN, Matti ; VÄLIMÄKI, Vesa ; JÁNOSY, Zoltán: Towards High-Quality Sound Synthesis of the Guitar and String Instruments. In: *International Computer Music Conference, Tokyo* 1.1
- [KVT98] KARJALAINEN, Matti ; VÄLIMÄKI, Vesa ; TOLONEN, Tero: Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond. In: *Computer Music Journal* 22 (1998), Nr. 3, 17-32. <http://www.acoustics.hut.fi/~vpv/publications/cmj98.pdf> 1.1
- [Leo03] LEON, Richard: Dr. Robert & his Modular Moogs, 1964-1981. In: *Sound on Sound* (2003), 10. <http://www.soundonsound.com/sos/oct03/articles/moogretro.htm> 6
- [MC00] MANSUETI, Maurizio ; CIRILLO, Luca: An Interview with Bob Moog. In: *Il Giaguaro "In Lounge" Magazin* 2 (2000). <http://www.thereminvox.com/article/articleview/154> 6
- [PM:] *Discrete Cosine Transform.* <http://planetmath.org/encyclopedia/DiscreteCosineTransform.html> 23
- [Smi92] SMITH, Julius O.: Physical Modelling using Digital Waveguides. In: *Computer Music Journal* 16 (1992), Nr. 4, 74-91. <http://ccrma.stanford.edu/~jos/pmudw/pmudw.pdf> 1.1
- [Smi03] SMITH, Steven W.: *Digital Signal Processing.* Newnes <http://www.dspguide.com>. – ISBN 0-750674-44-X 10, 1, 3.2.1, 11, 23
- [Tol98] TOLONEN, Tero: *Model-Based Analysis and Resynthesis of Acoustic Guitar Tones*, Helsinki University of Technology, Masters Thesis, 1998. <http://www.acoustics.hut.fi/~ttolonen/Thesis.Tol.ps.gz>. – Online-Ressource 1.1
- [Vog99] VOGEL, Helmut: *Gerthsen Physik*. 20. veränderte Auflage. Springer, 1999. – ISBN 3-540-65479-8 2
- [Wei] WEISSTEIN, Eric W.: *Autocorrelation.* <http://mathworld.wolfram.com/Autocorrelation.html> 1
- [Wik] *Abbildungen von Gitarren.* <http://de.wikipedia.org/wiki/Bild:GitarreSchema.jpg> <http://de.wikipedia.org/wiki/Bild:Gitarre.jpg> http://de.wikipedia.org/wiki/Bild:Fender_Stratocaster.jpg, 1.2

A Autokorrelation

Autokorrelation ist ein mathematischer Algorithmus, mit dem untersucht werden kann, wie stark eine Menge sich selbst ähnelt. Das Ergebnis der Autokorrelation ist eine Menge gleicher Größe wie die untersuchte Menge. Man untersucht, wie stark die Menge sich selbst ähnelt, indem die Menge mit einer rotierten bzw. verschobenen Version ihrer selbst multipliziert wird. Je größer das Ergebnis desto größer ist die Ähnlichkeit bei der entsprechenden Rotation / Verschiebung.

$$a[i] = \sum_{j=0}^n m[j] \cdot m[(j+i) \bmod n] \quad , 0 \leq i < n \quad (\text{A.1})$$

Es kann bewiesen werden, dass der maximale Wert der Ergebnismenge a den Index 0 hat.¹ Dies ist auch zu erwarten, denn der Index 0 bedeutet, dass die Menge *nicht* verschoben wurde. Die Menge besitzt mit der identischen Menge die größte Ähnlichkeit. Der Index des zweitgrößten Wertes zeigt dann, bei welcher Verschiebung die nächst kleinere Ähnlichkeit erreicht wird.

Da die untersuchten Samples nur Werte zwischen -1 und $+1$ annehmen, kann man anstatt der Multiplikation auch den Betrag der Differenz zweier Werte benutzen. Die Summanden von Formel A.1 sind dann $|m[j] \cdot m[(j+i) \bmod n]|$. Die größte Ähnlichkeit hat dann der jeweils kleinste Wert. Diese Variante hat bei der Verwendung von float Zahlen den Vorteil, dass die Ergebnisse i.d.R. genauer sind, denn bei der IEEE float Darstellung von Zahlen befinden sich zwischen 0 und 1 genauso viele Zahlen wie zwischen 1 und ∞ . Der Abstand zwischen zwei Zahlen nimmt jedoch mit der Größe des Betrags stark zu. Da prinzipiell nicht abgeschätzt werden kann, wie groß die Summen letztendlich werden, hat die Variante mit der Differenz Vorteile bezüglich der Genauigkeit.

Die Autokorrelation kann zur Frequenzbestimmung benutzt werden, wenn die Schwingung periodischen Charakter besitzt. Bei einer periodischen Schwingung ähneln sich die einzelnen Schwingungen, so dass das Ergebnis der Autokorrelation eine große Varianz besitzt. Es wird erwartet, dass in einem der Frequenz bzw. der Periodenlänge (in Samples) entsprechenden Abstand jeweils ein Minimum auftritt, da bei jeder Periodenlänge das Sample mit sich selbst die größte Ähnlichkeit besitzt. Da sich die Schwingung aber trotzdem permanent leicht ändert, sollte das Sample nach der ersten Periode die größte Ähnlichkeit und damit den kleinsten Wert haben. Die Verschiebung von 0 wird nun von vorneherein nicht beachtet, da diese kein Ergebnis darstellt. Da die Aufnahme aber mit einer recht hohen Samplerate von 44100 Hz erstellt wird, ist der Unterschied zwischen zwei aufeinanderfolgenden Samples i.d.R. sehr gering, denn der Gitarrenklang enthält kaum hochfrequente Anteile über 10 KHz. Die Wahrscheinlichkeit, dass eine

¹Den Beweis findet man in [Wei]

A Autokorrelation

große Ähnlichkeit bei einer nur sehr geringen Verschiebung festgestellt wird ist bei so hoher Samplerate hoch. Des weiteren wird mittels Autokorrelation nur die Grundfrequenz ermittelt, welche höchstens 1500 Hz erreicht. Die Anzahl der zu untersuchenden Verschiebungen bzw. die Größe der Ergebnismenge kann also erheblich reduziert werden.

Die Grundfrequenz der möglichen Töne einer Gitarre liegt zwischen 80 und 1500 Hz. Mittels [Formel A.2](#) lassen sich die beiden Indizes bestimmen, innerhalb deren die Autokorrelationsmenge bestimmt werden muss. Der Index des größten bzw. kleinsten Wertes dieser Menge ist dann die Verschiebung, die der Periodenlänge in Samples der Grundfrequenz entspricht.² Die dieser Länge entsprechende Frequenz lässt sich durch einfach Umformung von [Formel A.2](#) ermitteln.

$$\text{Index} = \text{int} \left(\frac{\text{Samplerate}}{\text{Frequenz}} \right) \quad (\text{A.2})$$

A.1 Berechnung der Genauigkeiten von 3.2.2

Eine Frequenz von 82 Hz entspricht bei einer Samplerate von 44100 Hz 538 Samples ($\frac{44100 \text{ Hz}}{82 \text{ Hz}} = 538$). Da die Samples immer den gleichen Abstand haben, also ganzzahlig sind, gibt es bei den Berechnungen immer einen kleinen Rundungsfehler. Die Frequenz, welche 538 Samples entspricht ist $\frac{44100}{538} = 81.97 \text{ Hz}$. Die nächsthöhere darstellbare Frequenz entspricht einer Periode von 537 Samples, also $\frac{44100}{537} = 82.12 \text{ Hz}$. Die Differenz aus diesen beiden Frequenzen gibt die Genauigkeit an, mit der die Autokorrelation unter diesen Rahmenbedingungen die Frequenz bestimmen kann: $82.12 - 81.97 = 0.15 \text{ Hz}$. Dies entspricht einer relativen Genauigkeit von $\frac{0.15 \text{ Hz}}{82 \text{ Hz}} \approx 1.80/100$.

Beim Ton e mit 1320 Hz ergeben sich folgende Werte: $\frac{44100}{1320} = 33$ Samples. $\frac{44100}{33} = 1336.36 \text{ Hz}$ und $\frac{44100}{32} = 1378.16 \text{ Hz}$. Die Genauigkeit entspricht also $1378.16 - 1336.36 = 41.8 \text{ Hz}$ bzw. $\frac{41.8 \text{ Hz}}{1320 \text{ Hz}} \approx 3.2 \%$.

²Genauer, die Länge der ersten Periode.

B Berechnung neuer Zeilen mittels exponentieller Regression

Neue Zeilen für die Amplitudenmatrix werden aus den vorhandenen Zeilen mittels einer exponentiellen Regression der Spalten berechnet. Dazu werden für jede Spalte die Parameter c und λ für eine Exponentialfunktion der Form

$$f(x) = c \cdot e^{\lambda \cdot i} \quad (\text{B.1})$$

berechnet. Dazu wird der natürliche Logarithmus aller Werte einer Spalte S in das Feld x übertragen:

$$x[i] = \ln(S[i]) \quad (\text{B.2})$$

Mit dem Feld x können dann c und λ wie folgt berechnet werden.

$$\lambda = \frac{1}{D} \cdot \left(n \cdot \sum_{i=1}^n i \cdot x[i] - \left(\sum_{i=1}^n i \right) \cdot \left(\sum_{i=1}^n x[i] \right) \right) \quad (\text{B.3})$$

$$b = \frac{1}{D} \cdot \left(\left(\sum_{i=1}^n i^2 \right) \cdot \left(\sum_{i=1}^n x[i] \right) - \left(\sum_{i=1}^n i \right) \cdot \left(\sum_{i=1}^n i \cdot x[i] \right) \right) \quad (\text{B.4})$$

$$D = n \cdot \sum_{i=1}^n i^2 - \left(\sum_{i=1}^n i \right)^2 \quad (\text{B.5})$$

$$c = e^b \quad (\text{B.6})$$

Nachdem für jede Spalte die Parameter c und λ berechnet wurden, können nun beliebig viele neue Zeilen berechnet werden, indem für jede Spalte in [Formel B.1](#) die Nummer der zu berechnenden Spalte in i eingesetzt wird.

Als Beispiel wird die 11. Zeile der Amplitudenmatrix [3.8](#) berechnet.¹ Für den Wert der ersten Spalte wird zunächst das Feld x konstruiert, dann c und λ .²

$$x = [53.8443, 52.9287, 50.1032, 47.4572, 44.9455, \\ 42.4908, 40.2277, 38.1067, 36.1289, 34.2217]$$

¹Im Synthesizer wird hier die durch die IDCT rekonstruierte Amplitudenmatrix verwendet.

²Diese beiden numerischen Werte wurden mit einem C-Programm berechnet und sind deshalb nur Näherungen, bzw. haben die für float Zahlen typischen Rundungsfehler.

B Berechnung neuer Zeilen mittels exponentieller Regression

$$\begin{aligned}\lambda &= -0.0525609 \\ c &= 55.168\end{aligned}$$

Durch einsetzen in [Formel B.1](#) mit $i = 11$ ergibt sich der gesuchte Wert der 11. Zeile. Der Term $11 - 1$ resultiert aus der Feldindizierung der Programmiersprache C, bei der das erste Element des Feldes den Index 0 hat.

$$55.168 \cdot e^{-0.0525609 \cdot (11-1)} \approx 32.615 \tag{B.7}$$

Die folgende Tabelle fasst die Ergebnisse der 11. Zeile zusammen.

	Spalte 1	Spalte 2	Spalte 3	Spalte 4	Spalte 5	Spalte 6	Spalte 7	Spalte 8	Spalte 9	Spalte 10
λ	-0.05256	-0.1055	-0.3721	-0.06769	-0.1921	-0.2718	-0.0568	-0.1440	-0.2443	-0.0347
c	55.168	66.2934	36.2771	17.1632	22.1849	11.0774	3.2183	0.6643	0.2593	0.1989
	32.6151	23.0741	0.8779	8.7223	3.2491	0.7314	1.8233	0.1574	0.0225	0.1406

Eine ausführlichere Einführung in Regressionstechniken findet sich in [\[Hot94\]](#).

C Glossar

ADSR Attack, Delay, Sustain, Release. Modell mit dem sich der Lautstärkeverlauf eines Tons beschreiben läßt.

Attackphase erste Phase des Gitarrenklangs. Siehe ADSR.

Autokorrelation Algorithmus mit dem die selbstähnlichkeit einer Menge bzw. Funktion bestimmt werden kann.

Bin Das Ergebnis der DFT ist eine Menge von Bins, welche jeweils einen Sinusoiden beschreiben.

Blackmanfenster Fensterfunktion zur Spektralanalyse mit DFT. Bestes *Alroundfenster*.

Boden (Gitarre) unteres Holzbrett des Korpus.

Bundstab Metallene Stäbchen auf dem Griffbrett, welche die Tonhöhe des gegriffenen Tons festlegen.

Decke (Gitarre) oberes Holzbrett des Korpus. Bei akustischen Gitarren enthält es i. d. R. ein Schalloch.

Dezibel Maß für die Lautstärke eines Klangs. Im allgemeinen jedoch ein Verhältnismaß für beliebige Größen. Für Anwendungen in der Akustik und Elektrotechnik gibt es viele standardisierte Dezibelvarianten.

Digital Waveguide Algorithmus Ein numerisch effizienter Algorithmus die numerische Integration der Wellengleichung zu berechnen.

Diskrete Fourier-Transformation Variante der Fourier-Transformation für diskrete Werte.

Fadeout Das Absinken der Lautstärke am Ende eines Klangs bis zum Erlischen der Schwingung.

Faltungstheorem $\mathcal{F}(f * g) = \sqrt{2\pi}(\mathcal{F}f) \cdot (\mathcal{F}g)$. Eine Multiplikation im Zeitbereich entspricht der Faltung im Frequenzbereich von zwei fourier-transformierten Funktionen. Der umgekehrte Fall gilt ebenfalls; eine Multiplikation im Frequenzbereich entspricht einer Faltung im Zeitbereich.

Fourier-Transformation Mathematische Transformation mit der eine Funktion vom Zeit- und den Frequenzbereich transformiert wird.

- FPU** Floating Point Unit. Prozessor der IEEE float Zahlen verarbeitet.
- Frame** Ausschnitt aus einem größeren Sample.
- Glissando** Kontinuierliche Veränderung der Frequenz eines Tons zur Frequenz eines anderen Tons.
- Grundfrequenz** Die tonhöhe bestimmende Frequenz eines Klangs, i.d.R. die kleinste Frequenz.
- Hals (Gitarre)** Holzteil zwischen Kopf und Korpus. Das Griffbrett ist auf dem Hals angebracht, sollte es aus einem anderen Holzteil bestehen; ansonsten ist das Griffbrett mit dem Hals identisch.
- Hammingfenster** Fensterfunktion zur Spektralanalyse mit DFT.
- Hanningfenster** Fensterfunktion zur Spektralanalyse mit DFT.
- IEEE** Institute of Electrical and Electronics Engineers
- Inverse Diskrete Fourier-Transformation** Variante der Inversen Fourier-Transformation für diskrete Werte.
- Inverse Fourier-Transformation** Transformiert eine Funktion vom Frequenz- in den Zeitbereich.
- Karplus-Strong Algorithmus** Eine Methode um die transversale Schwingung einer Saite numerisch effizient zu berechnen. Die Schwingung läßt sich jedoch noch effizienter mit dem Digital Waveguide Algorithmus berechnen. Beide Methoden sind äquivalent zur numerischen Integration der Wellengleichung.
- Kopf (Gitarre)** Im Kopf ist die (Stimmungs)Mechanik untergebracht. Er besteht i. d. R. zusammen mit dem Hals aus dem gleichen Stück Holz.
- Korpus** Der große (evtl. hohle) Körper der Gitarre.
- Lobe** Englischer Begriff für einen Berg in einem Periodogramm.
- longitudinale Schwingung** Schwingung einer Saite bei der die Bewegungsrichtung parallel zur Ausbreitungsrichtung verläuft. Bei Gitarren für den Klang von untergeordneter Bedeutung. Siehe transversale und torsionale Schwingung
- Mainlobe** Zentraler Berg im Periodogramm einer Fensterfunktion. Anhand des maximalen Wertes und der Breite lassen sich die Eigenschaften der Fensterfunktion charakterisieren (zusammen mit den Sidelobes).
- Mechanik (Gitarre)** Vorrichtung um die Saite zu spannen und stimmen.
- NaN** Not A Number. Spezialfall einer float Zahl, die keine gültige Zahl repräsentiert. Dies kann erreicht werden, wenn die Darstellungsgenauigkeit der float Zahl bei einer Berechnung unter- oder überschritten wurde.

- Nyquist-Frequenz** Die maximal darstellbare Frequenz in einem Sample. Abhängig von der Samplerate.
- Oberton** Stellt man einen Klang als Summe von Sinusoiden dar, heißen alle Sinusoiden mit Frequenzen größer der Grundfrequenz Oberton.
- PCM** Pulse Code Modulated. Direkte bzw. simpelste Möglichkeit Samples zu speichern.
- Periode** Ein Schwingungsdurchgang eines Klangs. Entspricht $\frac{1}{f}$ sec der Grundfrequenz.
- Periodogramm** Diagramm welches das Spektrum eines Klangs zeigt. Auch bekannt unter dem englischen Begriff *Power Spectrum*.
- Pickup** Bauteil einer elektrischen Gitarre mit dem schwingungen einer metallischen magnetisierbaren Saite in elektrischen Strom gewandelt werden.
- Plektrum** Kleines Plättchen mit welchem eine Saite angeschlagen wird. Normalerweise aus Kunststoff hergestellt.
- Rechteckfenster** Triviale Fensterfunktion mit i. d. R. schlechten Eigenschaften bei DFT.
- Regression** Mathematisches Verfahren um die Koeffizienten einer Funktion zu bestimmen, so dass die Funktion möglichst genau einer Menge von Meßwerten entspricht.
- Resynthese** siehe Synthese
- Ruheposition (der Saite)** Zustand bzw. Position der Saite im nicht angeregten, d.h. schwingungslosen Zustand.
- Sample** Englischer Begriff für Abtastwert. Bezeichnet der diskreten Wert einer Messung. In der Regel ist das Sample bezüglich Zeit und numerischem Wert diskretisiert.
- Samplerate** Englisch für Abtastfrequenz.
- Sattel (Gitarre)** Ende des Halses auf dem die Saite aufliegt.
- Schalloch** Loch in der Decke durch welches die Luft im Korpus zum Schwingen angeregt wird.
- Sidelobe** Nebenberge im Periodogramm. i. d. R. versucht man Sidelobes mit möglichst großer Dämpfung, also möglichst kleinen dB Wert zu generieren.
- Sinusoid** Eine Sinus- oder Cosinusfunktion mit nicht näher bestimmten bzw. benannten Faktoren für Amplitude, Frequenz und Phase.
- Sliding Window** Ein Verfahren einen Klang nicht ausschließlich im Frameabstand zu untersuchen. Der Frame wird dabei pro *Durchgang* eine beliebige Anzahl von Samples verschoben. Dabei muß das Ergebnis jedoch entsprechend der Verschiebung skaliert werden.

Sonogramm Eine Darstellung des Frequenzspektrums eines Klangs, bei dem die Amplituden der Frequenzen durch unterschiedliche Farben dargestellt werden. Prinzipiell identisch zum 3d-plot des Frequenzspektrums des Klangs, durch die Verwendung von Farben ist das Sonogramm jedoch eine reine 2d Darstellung.

Spectral Leakage Sinusoiden die im untersuchten Signal vorhanden sind, deren Frequenz aber zwischen zwei Referenzfrequenzen (der DFT) liegen, verteilen ihre Amplituden und Phasen auf die umliegenden (Referenz-)Sinusoiden. Dieser Effekt wird als Spectral Leakage bezeichnet. Das Ausmaß hängt von der verwendeten Fensterfunktion ab.

Spektralanalyse Analyse einer Funktion oder Wertemenge auf in der Menge vorkommenden bzw. in der Funktion enthaltenen Frequenzen. Implizit wird dabei angenommen, dass das untersuchte mathematische Objekt als Summe mehrerer Sinusoiden unterschiedlicher Frequenz dargestellt werden kann.

Steg (Gitarre) Befestigung der Saiten auf der Decke.

Sustainphase zweite Phase des Gitarrenklangs. Siehe ADSR.

Synthese Konstruktion bzw. Berechnung einer komplizierten Funktion durch mehrere einfache Funktionen. Bei Audio Anwendungen wird aber häufig jede Art von Algorithmus zur Klangerzeugung als Synthese(algorithmus) bezeichnet.

Tiefpaßfilter Ein Filter, der hohe Frequenzen dämpft bzw. tiefe Frequenzen verstärkt.

torsionale Schwingung Schwingung bei der die Saite in sich verdreht wird. Sie ist bei mit Bogen gespielten Saiteninstrumenten ein wesentliches Klangbestimmendes Merkmal. Siehe transversale und longitudinale Schwingung.

Transformation Ergebnismenge bzw.-funktion der Anwendung einer Funktion oder eines Algorithmus auf eine Ausgangsmenge bzw.- funktion.

transversale Schwingung Schwingung einer Saite bei der die Bewegungsrichtung senkrecht zur Ausbreitungsrichtung verläuft. Bei Gitarren die für den Klang wichtigste Schwingung der Saite. Siehe longitudinale und torsionale Schwingung

Vibrato Spieltechnik bei der die Frequenz eines Tons minimal moduliert wird. Dies läßt den Klang für den Zuhörer i. d. R. interessanter erscheinen.

Wellengleichung allgemeine Beschreibung der Ausbreitung von Wellen.

Zarge Seitliche Teile des Korpus zwischen Decke und Boden.

Zero Padding Verfahren die Frequenzauflösung der Spektraluntersuchung eines Samples mittels DFT zu verbessern.

D Trivia

Bei der Entwicklung der Software für Analyse und Resynthese und für das Erstellen der Printvorlage dieser Diplomarbeit wurde folgende (Hilfs)Software von mir entwickelt.

midimulticast Eine Software um MIDI Daten über UDP Multicast an andere Netzwerkrechner zu verschicken.

qtpod Eine Software um alle Funktionen des Line6 POD Pro per MIDI zu steuern.

sample2ps und verwandte Tools mit denen die Abbildungen von Audio Daten in dieser Arbeit erstellt wurden.

wavshaper ein Programm um eine (Bezier)Hüllkurve auf ein Sample anzuwenden. Damit wurde die cyanfarbene Gitarre auf dem Titelbild erstellt.

hawsty Ein \LaTeX Style mit dem die Titelseiten entsprechend den Layoutvorgaben der HAW-Hamburg erzeugt werden.

Für folgende Software wurden von mir Patches erstellt um Fehler zu beheben oder zusätzliche Funktionen zu implementieren: qjackctl, gnusound, lilypond, envy24control. Ich habe einen pascal2postscript compiler auf (modernes) Linux portiert, mit dem ich eine Routine für das sample2sonogram Programm compiliert habe. Der Fehler auf Seite 371 in [Knu99] wurde von mir entdeckt und brachte mir \$2.56 ein, sowie ein Fehler in [GB04] auf Seite 333.

Die von mir entwickelte Software besteht aus *approx XX* KB C++ Source Code, der Text dieser Arbeit aus $\approx XX$ KB \LaTeX Code. Zur Entwicklung und Verifizierung der Algorithmen wurden *XX* MB Samples von *X* unterschiedlichen Gitarren aufgenommen.

Die Entwicklung der Algorithmen und des Programmcodes fand von März 2005 bis Ende August 2005 statt, nachdem ich mich 15 Semester auf diese Aufgabe vorbereitet hatte.

E History

- V1.x**
- Kapitel *Trivia* hinzugefügt.
 - Layout besser and Web-PDF-Version angepasst

V1.0 22. August 2005 zur Prüfung eingereichte Version.