Universität Karlsruhe (TH), Fakultät für Informatik
Institut für Technische Informatik
Lehrstuhl für Intelligente Sensor-Aktor-Systeme (ISAS)
Prof. Dr.-Ing. Uwe D. Hanebeck

Diplomarbeit

Trennung von Musikaufnahmen mit impliziter Transkription der getrennten Spuren

Tobias Gehrig

13. Oktober 2008

Referent: Prof. Dr.-Ing. Uwe D. Hanebeck
Korreferent: Prof. Dr.-Ing. Heinz Wörn

Betreuer: Dipl.-Ing. Frederik Beutler

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Diplomarbeit selbstständig angefertigt zu haben. Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis aufgeführt.

Karlsruhe, 13. Oktober 2008

Tobias Gehrig

Inhaltsverzeichnis

Abbildungsverzeichnis **V**

Tabellenverzeichnis **IX**

Notation **XI**

1 Einleitung **1**

1.1 Problembeschreibung 1

1.2 Stand der Forschung 2

1.3 Beitrag dieser Arbeit 4

2 Klangmodelle **6**

2.1 Sinusoidal-Modell 10

2.1.1 Harmonische Schwingung 10

2.1.2 Quasi-harmonische Schwingung 10

2.1.3 Zeitvariante Amplitude 11

2.2 ARMA Modell 12

2.2.1 AR-Modell für Sinusschwingung 12

2.2.2 AR-Modell für summierte Sinusschwingungen 13

2.3 Physikalisches Modell 14

2.3.1 Kommutierte Synthese 15

2.3.2 Karplus-Strong Modell 15

2.3.3 Erweitertes Karplus-Strong Modell 17

2.4 Stereo Modellierung 28

3 Klanganalyse **30**

3.1 Onset Detektion . . . 30

3.1.1 Schätzung des Transientenanfangs . . . 31

3.1.2 Schätzung des Transientenendes . . . 32

3.2 Kurzzeit-Fourier-Transformation (STFT) . . . 32

3.2.1 Fensterfunktionen . . . 33

3.3 Peak Detektion . . . 33

3.3.1 Peak Interpolation . . . 34

3.3.2 Spektralmaskierung . . . 35

3.4 Grundtonschätzung . . . 36

3.4.1 Differenz der Frequenzdifferenzen . . . 37

3.4.2 Nichtlineare Regression mittels Inharmonizitätskurve . . . 37

3.5 Teiltonverfolgung . . . 38

3.5.1 Energy Decay Relief (EDR) . . . 39

3.6 Schleifenfilter Identifikation . . . 40

3.6.1 Saitendämpfungsfilter Identifikation . . . 40

3.6.2 Saitensteifheitsfilter Identifikation . . . 44

3.6.3 Saitenstimmungsfilter Identifikation . . . 45

3.7 Rekonstruktion der Aggregatanregung . . . 45

4 Algorithmen **47**

4.1 Kalman Filter . . . 47

4.1.1 Prädiktion . . . 47

4.1.2 Korrektur . . . 48

4.1.3 Randbedingungen für Zustände . . . 48

4.2 Unscented Kalman Filter . . . 49

4.2.1 Unscented Transformation . . . 49

4.2.2 Unscented Transformation für bedingt lineare Modelle . . . 50

4.2.3 UKF Algorithmus . . . 51

4.3 Interacting Multiple Model Algorithmus . . . 52

4.3.1 Interaktion . . . 52

4.3.2 Filterung 53
4.3.3 Kombination 54

5 Experimente **55**
5.1 Trennung zweier Sinusschwingungen 55
5.2 Trennung zweier Karplus-Strong-Schwingungen 59
5.3 Überlegungen zu Optimierungsmöglichkeiten 64
5.3.1 Komplexität 64
5.3.2 Robustheit der Modellschätzung 65

6 Fazit und Ausblick **66**

7 Danksagung **68**

Literaturverzeichnis **69**

Abbildungsverzeichnis

2.1 Die ersten vier Moden einer idealen Saite. 7

2.2 Zeitvariante Spektralhülle eines Klavierklangs. 8

2.3 Zeitvariante Spektralhülle eines Trompetenklangs. 8

2.4 Exemplarische Darstellung einer ADSR Hüllkurve mit den vier Phasen: Attack, Decay, Sustain und Release. 9

2.5 Hüllkurve eines Klaviertons. 9

2.6 Hüllkurve eines Trompetentons. 10

2.7 Pol-Nullstellendiagramm für eine Sinusschwingung. 13

2.8 Strukuturdiagramm eines LTI-Systems einer gedämpften Sinusschwingung. . . . 14

2.9 Schematische Darstellung der Synthese eines Saiteninstruments. 15

2.10 Schematische Darstellung der kommutierten Synthese eines Saiteninstruments mit Aggregatanregung. 15

2.11 Schematische Darstellung der kommutierten Synthese eines Saiteninstruments. . 15

2.12 Strukturdiagramm für das Karplus-Strong Modell. 16

2.13 Amplitudenantwort, Phasenlaufzeit und Impulsantwort einer linearen Interpolation. 18

2.14 Strukturdiagramm für das erweiterte Karplus-Strong Modell. 18

2.15 Amplitudenantwort und Impulsantwort eines Zupfrichtungsfilters. 19

2.16 Amplitudenantwort und Impulsantwort eines Zupfpositionsfilters. 20

2.17 Amplitudenantwort und Impulsantwort eines FIR-Filters als Saitendämpfungsfilter. 22

2.18 Amplitudenantwort, Phasenantwort und Impulsantwort eines Einpolfilters als Saitendämpfungsfilter. 23

2.19 Phasenlaufzeit und Impulsantwort einer Kaskade von $M = 4$ identischen Allpassfiltern zweiter Ordnung als Saitensteifheitsfilter. 24

2.20 Phasenlaufzeit und Impulsantwort einer Kaskade von $M = 8$ identischen Allpassfiltern erster Ordnung als Saitensteifheitsfilter. 25

2.21 Phasenlaufzeit und Impulsantwort eines Allpassfilters als Saitenstimmungsfilter. 27

3.1 Strukturdiagramm des verwendeten Klanganalyseverfahrens. 30

3.2 Einschwinghüllkurve für einen Gitarrensaite. 31

3.3 Onsetfunktion mit gewähltem Schwellwert. 32

3.4 Spektrogramm einer Aufnahme einer Gitarrensaite. 33

3.5 Fensterfunktionen: Hamming-Fenster und Gauss-Fenster 34

3.6 Parabolische Interpolation eines spektralen Maximums mittels dreier Stützstellen aus dem Spektrum. 35

3.7 Spektralmaskierung des Spektrums in Anlehnung an den Maskierungseffekt beim menschlichen Gehör. 36

3.8 Inharmonizitätskurve für die gemessenen Teiltonfrequenzen mit der harmonischen Frequenz als Vergleich. 38

3.9 Aus der Teiltonverfolgung resultierende Frequenzverläufe. 39

3.10 Aus der Teiltonverfolgung resultierende Amplitudenverläufe. 40

3.11 Energy Decay Relief (EDR) einer Aufnahme einer Gitarrensaite. 41

3.12 Aus der Teiltonverfolgung basierend auf dem Energy Decay Relief resultierende Frequenzverläufe. 42

3.13 Aus der Teiltonverfolgung basierend auf dem Energy Decay Relief resultierende Amplitudenverläufe. 42

3.14 Lineare Regression zur Approximation der Hüllkurve des ersten Teiltons durch eine Gerade. 43

3.15 Amplitudenantwort des Saitendämpfungsfilters und gemessene Dämpfungsfaktoren. 44

3.16 Phasenlaufzeit des Saitensteifheitsfilters verglichen mit der optimalen Phasenlaufzeit. 45

3.17 Geschätzte Aggregatanregung zusammen mit dem Originalsignal. 46

4.1 Strukturdiagramm für den Interacting Multiple Model Algorithmus. 53

5.1 Originalsignal der ersten Sinusschwingung. 56

5.2 Originalsignal der zweiten Sinusschwingung. 56

5.3 Mixtur zweier Sinusschwingungen mit additivem Rauschen. 57
5.4 Signalschätzung für den ersten Sinuston. 57
5.5 Signalschätzung für den zweiten Sinuston. 58
5.6 Summensignalschätzung für die Trennung zweier Sinusschwingungen. 58
5.7 Modellwahrscheinlichkeiten für die Trennung zweier Sinusschwingungen. 59
5.8 Originalsignal der ersten Karplus-Strong-Schwingung. 60
5.9 Originalsignal der zweiten Karplus-Strong-Schwingung. 60
5.10 Mixtur zweier Karplus-Strong-Schwingungen mit additivem Rauschen. 61
5.11 Signalschätzung für die erste Karplus-Strong-Schwingung. 61
5.12 Differenz zwischen Originalsignal und Schätzung für die erste Karplus-Strong-Schwingung. 62
5.13 Signalschätzung für die zweite Karplus-Strong-Schwingung. 62
5.14 Differenz zwischen Originalsignal und Schätzung für die zweite Karplus-Strong-Schwingung. 62
5.15 Summensignalschätzung für die Trennung zweier Karplus-Strong-Schwingungen. 63
5.16 Modellwahrscheinlichkeiten für die Trennung zweier Karplus-Strong-Schwingungen. 63

Tabellenverzeichnis

2.1 Filterdesignkonstanten für die Schätzung des Saitensteifheitsfilters. 25

Notation

Konventionen

x	Skalar
$\boldsymbol{x}$	Zufallsvariablen werden durch klein- und fettgeschriebene Buchstaben dargestellt. Realisierungen einer Zufallsvariablen erhalten reguläre Kleinbuchstaben, z.B. x.
$\hat{x}$	Mittelwert einer Zufallsvariablen $\boldsymbol{x}$.
$x(k)$	Skalar zum Zeitpunkt k.
$\underline{x}$	Spaltenvektor
$\underline{\boldsymbol{x}}$	Zufallsvektor
$\underline{x}(k)$	Vektor zum Zeitpunkt k.
$\mathbf{A}$	Matrizen werden groß und fett geschrieben.
$\mathbb{N}$	Menge der natürlichen Zahlen.
$\mathbb{R}$	Menge der reellen Zahlen.
$\sim$	Verteilungsoperator. Z.B. bedeutet $\boldsymbol{x} \sim \mathcal{U}$, dass $\boldsymbol{x}$ gemäß der Verteilung $\mathcal{U}$ verteilt ist.
■	Ende eines Beispiels oder einer Beschreibung zur Referenzanwendung.
□	Ende eines Beweises.

KAPITEL 1

Einleitung

Musikaufnahmen sind in der Regel auf CD im Stereoformat vorhanden. In manchen Situationen mag es allerdings wünschenswert sein, aus der Stereoaufnahme die einzelnen Instrumente zu extrahieren. Dies ist z.B. der Fall, wenn man daraus ein Remix erstellen möchte, aber die original Mehrspuraufnahmen nicht vorhanden oder unzugänglich sind. Aber auch ein Musiker kann sich vor diesem Problem sehen, wenn er beispielsweise gerne einen neuen Song einstudieren möchte oder einfach einen Part des Songs durch sein Spiel ersetzen möchte. Hätte er nun die ursprünglichen Tonspuren, könnte er einfach die Tonspur, die er einstudieren möchte, separat anhören bzw. die ausblenden, welche er durch sein Spiel ersetzen möchte. Ein weiterer Fall findet sich im Gebiet des Music-Information-Retrieval, wo z.B. durch query by humming oder query by playing, in einer Datenbank nach Songs gesucht werden soll. Diese Datenbank muss aber zuvor erst mit Daten versorgt werden, welche man z.B. durch automatische Transkription von CD-Aufnahmen erhält.

1.1 Problembeschreibung

Es stellt sich also nun das Problem aus Stereoaufnahmen bzw. Monoaufnahmen die ursprünglichen Tonspuren zu extrahieren.

Dieses Problem ist unterbestimmt, da in der Regel mehr als zwei Instrumente zusammen spielen werden. Da für *Beamforming* oder *Independent Component Analysis* aber die Voraussetzung besteht, dass mindestens soviele Signalmixturen wie darin enthaltene Quellensignale vorhanden sind, sind diese Verfahren in diesem Falle nicht anwendbar.

Ein weiteres Problem besteht in der Signalstruktur. Aufgrund der harmonischen Struktur sowohl der Klänge als auch der Musik überlagern sich Obertöne verschiedener Grundtöne. Dadurch lassen sich die Signale nicht mit binären Frequenzmaskierungsalgorithmen trennen.

Auch die zeitliche Korrelation zwischen einzelnen Events durch das rhythmische Zusammenspiel der Musiker stellt ein Problem für Algorithmen wie z.B. die *Independent Subspace Analysis* dar, da hierdurch die Unabhängigkeitsannahme verletzt wird.

1.2 Stand der Forschung

Im folgenden Abschnitt soll nun ein kurzer Überblick über bereits getätigte Anstrengungen in der Forschung gegeben werden, dieses Problem zu lösen.

Der größte Teil existierender Verfahren zur Quellentrennung zielt auf die Trennung von Sprache ab [43, 71]. Manche nutzen auch explizit Eigenschaften von Sprachsignalen aus [71], was sie nur bedingt für Musik nutzbar macht. Andere wiederum machen möglichst wenige Annahmen über die zu trennenden Quellen, was sie vielseitig einsetzbar macht. Es gibt allerdings auch einige Verfahren, die speziell für die Trennung von musikalischen Signalen konzipiert sind.

Exisitierende Verfahren zur Quellentrennung können allgemein nach [35, §1.3] grob in drei Kategorien unterteilt werden. Dies sind einmal *Computational Auditory Scene Analysis* (CASA), bei der das Signal auf bekannte Merkmale untersucht wird, dann *Blind Source Separation* (BSS) bzw. Independent Component Analysis (ICA), bei der außer der Unabhängigkeit der Quellen in der Regel keine weiteren Annahmen über die Quellen gemacht werden, und *Generalized Nonlinear Processing* (GNLP), das einschränkendere Annahmen über die Quellen macht als BSS Algorithmen und somit eine Untergruppe derer ist, allerdings nicht so starke Einschränkungen, wie dies bei CASA der Fall ist.

Computational Auditory Scene Analysis (CASA)

Merkmale die bei CASA beispielsweise genutzt werden sind die harmonische Struktur der musikalischen Töne, Tonhöhen, Tonanschläge am Tonanfang, instrumentspezifisches bandbeschränktes Rauschen oder Klangfarbenhüllkurven [35, §1.3.1]. Für diese Merkmale sind in der Regel generative Modelle vorhanden, wie z.B. *Sinusoidal Modelle*, unter Umständen auch kombiniert mit Modellen für zusätzliches Rauschen [49] und/oder Transienten [61].

Ein Verfahren aus diesem Bereich wurde z.B. in [15] vorgestellt, was mittels eines *Partikelfilters* mehrere zeitvariante (quasi-)harmonische Komponenten, jeweils bestehend aus Grundton und zugehörigen Obertönen, im *Spektrogramm* verfolgt und somit trennt. Dabei wird sowohl die Anzahl der harmonischen Komponenten im Signal wie auch deren Anfangs- und Endzeit und die Frequenz mit Amplitude jedes *Teiltones* geschätzt.

In [13] wurde ein *Markov chain Monte Carlo* (MCMC) Verfahren vorgestellt, das auf vorsegmentierten Daten mittels einer Gabor Repräsentation der Signale die Trennung mehrerer musikalischer Töne vornimmt.

Ein weiteres Frequenzbereichsverfahren dieser Kategorie wurde in [41] präsentiert, das mittels mehrerer Agenten, die jeweils auf dem Residuum arbeiten, versucht die Signale zu trennen. Es werden drei Agententypen verwendet: Ein Ereigniserkenner, ein Verfolgergenerator und mehrere Verfolger. Der Ereigniserkenner ist dafür zuständig, das Residuum zu berechnen, indem er die Prädiktionen der einzelnen Verfolger vom Eingangssignal subtrahiert. Der Verfolgergenerator

beobachtet, das daraus resultierende Residuum, um eventuell einen neuen Verfolger zu initiieren. Die Verfolger haben jeweils die Aufgabe, harmonische Signale mit einer, vom Verfolger vorgegebenen, Frequenz zu extrahieren. Das Eingangssignal für diese Verfolger ergibt sich als die Summe aus dem vom Ereigniserkenner berechneten Residuum und der eigenen Prädiktion. Außerdem existiert ein Verfolger für Geräusche. Dieser verfolgt die Spektralintensität der nicht-harmonischen Komponenten im Signal und adaptiert diese Intensität in Phasen, in denen keine harmonischen Quellen vorhanden sind. Dieses Verfahren wurde allerdings nur auf Sprache evaluiert.

Der Vorteil der Verfahren aus dieser Kategorie ist, dass sie auf einkanalige Signalmixturen anwendbar sind. Außerdem können die getrennten Signale anhand der Modellparameter leicht nachbearbeitet werden, um z.B. *Pitchshifting* oder *Timestretching* durchzuführen. Allerdings haben sie den Nachteil in der Regel sehr rechenaufwändig zu sein und nur für spezialisierte Problemstellungen geeignet zu sein. Außerdem werden unter Umständen sehr viele Trainingsdaten bzw. Segmentierungsdaten benötigt, die nicht unbedingt immer vorhanden sind.

Blind Source Separation (BSS)

Ein großer Vertreter der BSS Algorithmen ist die Independent Component Analysis. Eine ausführliche Diskussion darüber findet sich in [21]. Dabei wird angenommen, dass die Quellen *statistisch unabhängig* sind und die Mixturen $\underline{x}(t)$ im einfachsten Fall eine Linearkombination der Quellen $\underline{s}(t)$ darstellen

$$\underline{x}(t) = \mathbf{A}\underline{s}(t). \tag{1.1}$$

Dies wird ausgenutzt, um die *Mischmatrix* $\mathbf{A}$ bzw. deren Inverse zu schätzen. Allerdings gilt die Bedingung, dass mindestens so viele Signalmixturen wie Quellen vorhanden sind, was dazu führt, dass bei Stereoaufnahmen maximal zwei Signale voneinander getrennt werden können. Ist diese Bedingung nicht erfüllt, so spricht von einem *unterdeterminierten Problem*. Außerdem gibt es hierbei das Skalierungs- und Permutationsproblem, d.h. die Signale können bis auf einen konstanten Faktor bestimmt werden und die Reihenfolge der getrennten Komponenten kann nicht festgestellt werden.

Eine Abwandlung von der ICA stellt die *Independent Subspace Analysis* (ISA) dar [9]. Diese arbeitet auf der *Kurzzeit-Fourier-Transformation* einer Monoaufnahme und ermöglicht so die Trennung im unterdeterminierten Fall. Dazu wird die ICA über einem Block von mehreren aufeinanderfolgenden Frequenzscheiben ausgeführt. Dadurch erhält man eine Darstellung der Frequenzscheiben als eine gewichtete Summe von *Spektralbasen*. Diese werden dann unter Verwendung eines sogenannten *Ixegramms* zu *Mehrkomponentenunterräume* (engl. Multicomponent Subspaces) gruppiert, um als Basis für die einzelnen Quellen zu fungieren. Dabei handelt es sich bei dem Ixegramm um eine Matrix, bestehend aus paarweisen Ähnlichkeitsmaßen über die Menge der unabhängigen Komponenten mittels einer Approximation der symmetrischen

Kullback-Leibler-Divergenz. Zur Rekonstruktion der so getrennten Quellen werden die Mehrkomponentenunterräume rückprojeziert und mittels einer inversen Fourier Transformation und *Overlap-Add* wieder in den Zeitbereich transformiert.

Generalized Nonlinear Processing (GNLP)

Algorithmen aus der Kategorie des Generalized Nonlinear Processings verwenden z.B. Zeitfrequenzmasken, um die Mehrdeutigkeiten im unterdeterminierten Fall zu lösen. Dazu gehört beispielsweise die *Degenerate Unmixing Estimation Technique* (DUET) [71], welche die Tatsache ausnutzt, dass Sprachsignale in der Regel sich in der Zeitfrequenzdarstellung nicht bzw. kaum überlappen. Zur Trennung werden deshalb binäre Zeitfrequenzmasken verwendet. Dazu wird zuerst ein Histogramm über das Verhältnis zwischen der Zeitfrequenzdarstellung des linken und rechten Kanals erstellt, mittels dem die räumliche Information der einzelnen Quellen, beschrieben durch den Dämpfungsfaktor und die Verzögerung zwischen linkem und rechtem Kanal, ermittelt wird. Mit Hilfe dieser Parameter können dann die binären Frequenzmasken ermittelt werden.

In [46] wurde auf Basis der DUET ein Geräuschunterdrückungsverfahren names *Musical Interference Unmixing Estimation Technique* (MINUET) vorgestellt, was einstreuende Musik aus einem Sprachsignal entfernt.

Da die Annahme, dass sich die Quellen in der Zeitfrequenzdarstellung nicht überlappen, bei Musikaufnahmen, aufgrund der gespielten Harmonien und der harmonischen Obertonverteilung der Klänge, nicht erfüllt wird, wurde in [34] eine Erweiterung der DUET präsentiert, die es ermöglicht, pro Zeitfrequenzband zwei aktive Signale zu trennen, statt bei DUET einem. Dies wird über einen Bayes'schen Ansatz erreicht. In [35, §4] wurden zu dem mehrere Möglichkeiten der Phasenrekonstruktion und deren psychoakkustische Wirkung aufgezeigt.

Diese Verfahren haben allerdings das Problem, dass sie Quellen mit identischen Mixturparametern nicht trennen können. Außerdem verursachen die Frequenzmasken, vor allem binäre Frequenzmasken, Aliasingeffekte, was dazu führt, dass sich das Ergebnis unter Umständen anhört wie eine stark komprimierte MP3.

1.3 Beitrag dieser Arbeit

In dieser Arbeit soll diese Trennung modellbasiert mittels probabilistischer Zustandsschätzer im Zeitbereich umgesetzt werden. Dabei wird für jeden zu trennenden Ton ein separates Modell verwendet. Somit enthalten die getrennten Signale im Idealfall nur Vorkommnisse von dem Ton, für den das jeweilige Modell "zuständig" ist. Dies ermöglicht auch die direkte Weiterverwendung für eine Transkription der Musikaufnahme.

Dafür werden in Kap. 2 zuerst einige betrachtete Modelle beschrieben, mit Hilfe derer man Klänge beschreiben kann, nämlich das Sinusoidal-Modell in unterschiedlichen Formen und physikalisch motivierte Modelle, wie das Karplus-Strong Modell und seine Erweiterungen. Danach wird in Kap. 3 im Detail ein Verfahren der Klanganalyse zur Schätzung der nötigen Modellparameter beschrieben. Darauf folgt eine Betrachtung der verwendeten Algorithmen in Kap. 4. Aufbauend auf dem Kalman Filter, der für lineare Schätzprobleme Verwendung findet, wird ein lineares Regressionsfilter beschrieben, nämlich das Unscented Kalman Filter, was für nichtlineare Schätzprobleme genutzt wird. Zum Ende dieses Kapitels wird der Interacting Multiple Model Algorithmus vorgestellt, der zur Zustandsschätzung mittels mehrerer vorgegebener Modelle dient. Abgeschlossen wird die Arbeit mit der Präsentation einiger durchgeführter Experimente und angedachter Optimierungsmöglichkeiten in Kap. 5, gefolgt von einem Fazit und einem Ausblick auf noch zu betrachtende Probleme in Kap. 6.

KAPITEL 2

Klangmodelle

Bevor beschrieben wird, wie man Klänge modellieren kann, soll allerdings erst einmal die Frage beantwortet werden, was man überhaupt unter einem Klang versteht.

Ein musikalischer Ton ist in der Regel aus einem *Grundton* und mehreren *Obertönen* zusammengesetzt, wobei der Grundton auch als erster *Teilton* bezeichnet wird und der erste Oberton entsprechend als zweiter Teilton usw. . Wenn der Ton von einem harmonischen Instrument erzeugt wird, so sind die Frequenzen f_p der Obertöne ganzzahlige Vielfache der Grundfrequenz f_0

$$f_p = p \cdot f_0, \quad p \in \mathbb{N}. \tag{2.1}$$

Dies sind gerade die Frequenzen der sogenannten *Moden*, auch *Schwingungsmoden*, des entsprechenden Instruments, die aus den stationären Eigenschaften stehender Wellen des Schwingungssystems folgen [19, §10.1] [68]. In Abb. 2.1 sind die ersten vier Moden einer idealen Saite veranschaulicht, die als vollkommen gleichförmig und elastisch angenommen wird.

Reale Saiten besitzen jedoch eine gewisse *Saitensteifheit*. Dies führt dazu, dass die Saite nicht nur durch die äußere Anregung in Schwingung versetzt wird, sondern zusätzlich auch durch die, von der Saitensteifheit verursachte, Rückstellkraft eine Beschleunigung erfährt [19, §10.1]. Als Resultat davon werden die Obertöne leicht nach oben verstimmt.

Je mehr die Obertöne von ihrer idealen harmonischen Frequenz abweichen, desto mehr nimmt der Mensch diesen Klang als Geräusch wahr, da er keinen Grundton mehr identifizieren kann [33, §2.1.2.2].

Die Klangfarbe eines Klanges wird dabei unter anderem durch die *Spektralhülle* und die *Hüllkurve* über die Zeit beschrieben [22, §2.3]. Die Spektralhülle wurde in [22, §6.3] als die maximale Amplitude der einzelnen Teiltöne definiert. In [32] wurde zur Identifikation von Instrumenten die zeitvariante Spektralhülle, also die Hüllkurven für die einzelnen Teiltöne, verwendet. Eine solche zeitvariante Spektralhülle ist in Abb. 2.2 auf Seite 8 und Abb. 2.3 auf Seite 8 exemplarisch für einen Klavierklang bzw. einen Trompetenklang abgebildet. Die Hüllkurve gibt den Pegelverlauf des Klangs im Zeitbereich an und kann grob in vier Phasen unterteilt werden [22, §5.1] [62]: *Anstiegsphase* (engl. Attack), *Abklingphase* (engl. Decay), *Haltephase* (engl. Sustain) und *Freigabephase* (engl. Release). Eine solche Hüllkurve wird auch *ADSR-Hüllkurve* genannt und ist examplarisch in Abb. 2.4 auf Seite 9 veranschaulicht. Dabei beschreibt die Anstiegs-

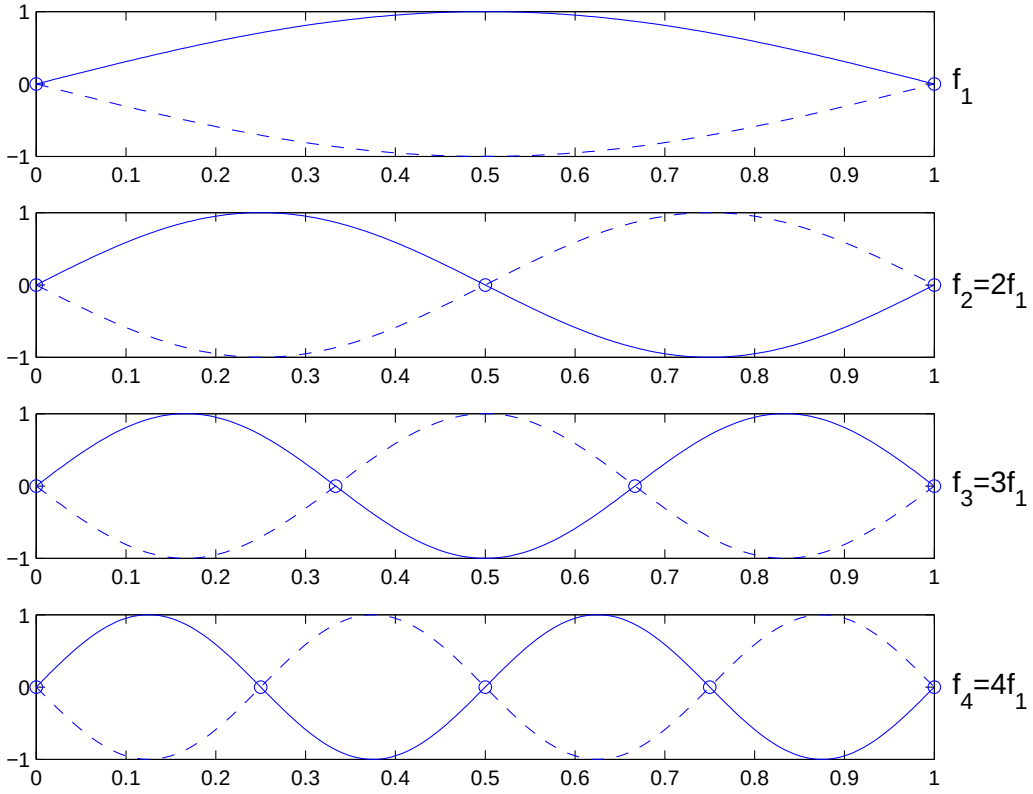

Abbildung 2.1: Die ersten vier Moden einer idealen Saite. Gezeigt sind jeweils die maximalen Auslenkungen. Die mit o markierten Stellen zeigen die Knoten der Moden, an denen sich die Moden nicht bewegen.

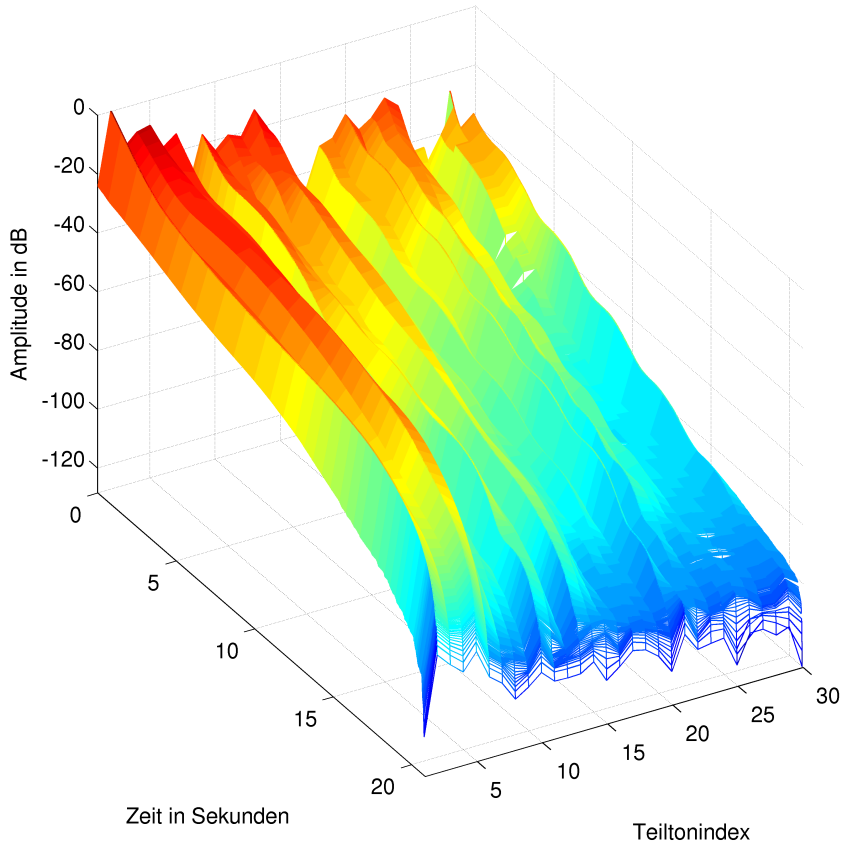

Abbildung 2.2: Zeitvariante Spektralhülle eines Klavierklangs.

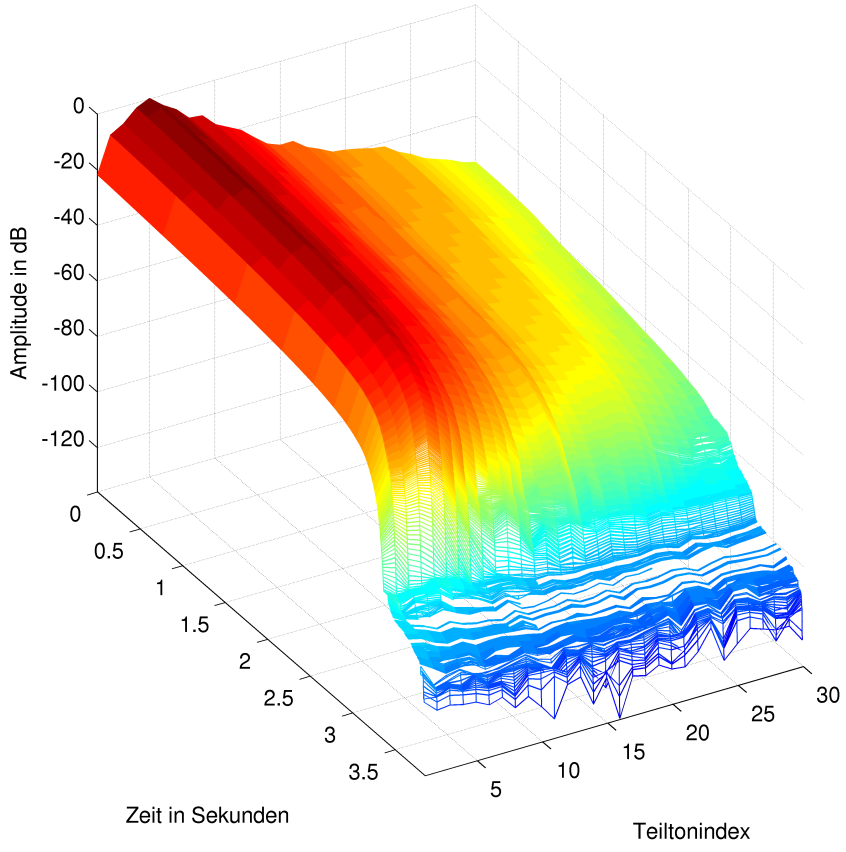

Abbildung 2.3: Zeitvariante Spektralhülle eines Trompetenklangs.

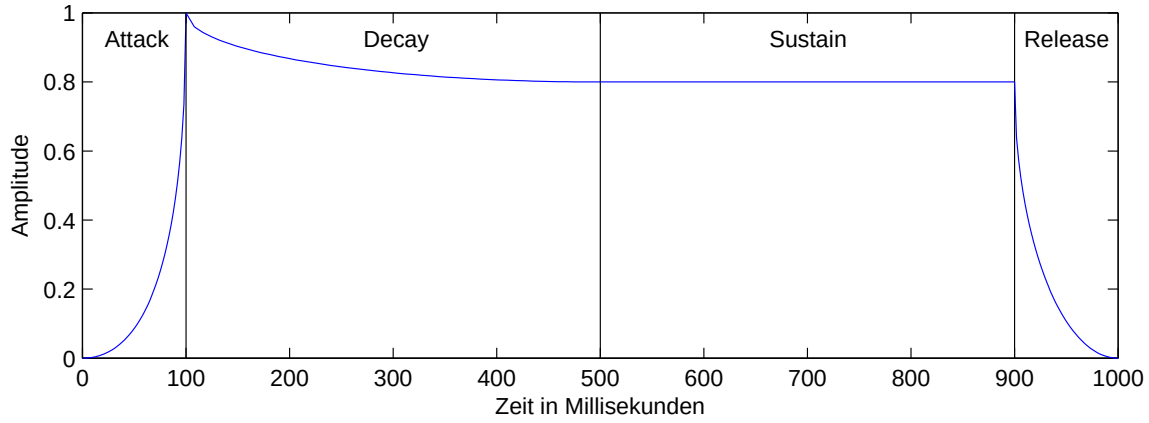

Abbildung 2.4: Exemplarische Darstellung einer ADSR Hüllkurve mit den vier Phasen: Attack, Decay, Sustain und Release.

phase den Zeitraum, bis die Schwingung ihren maximalen Pegel erreicht hat. Darüber hinaus ist die Anstiegsphase und der damit verbundene *Transient* auch entscheidend für die Klangfarbenunterscheidung [33, §2.1.4] [17, §3.1.2]. Direkt danach fällt der Pegel in der Abklingsphase wieder ab und wird in der Haltephase auf einem konstanten *Haltepegel* gehalten. Die Freigabephase beschreibt schließlich den Zeitraum, in dem der Ton noch weiter nachklingt, obwohl er nicht mehr aktiv ist. Dies entspricht z.B. im Falle einer Gitarre dem Nachklingen des Tones im Korpus, nachdem die Saite abgedämpft wurde und somit nicht mehr schwingt.

In der Realität schließen sich die Abkling- und Haltephase oft gegenseitig aus. So hat beispielsweise ein Saiteninstrument, das nur einmal angeschlagen oder gezupft wird, wie z.B. ein Klavier, in der Regel keine Haltephase, da die Saitenschwingung kontinuierlich schwächer wird (siehe Abb. 2.5). Dagegen haben Instrumente mit fortlaufender Anregung, wie z.B. eine Trompete, im

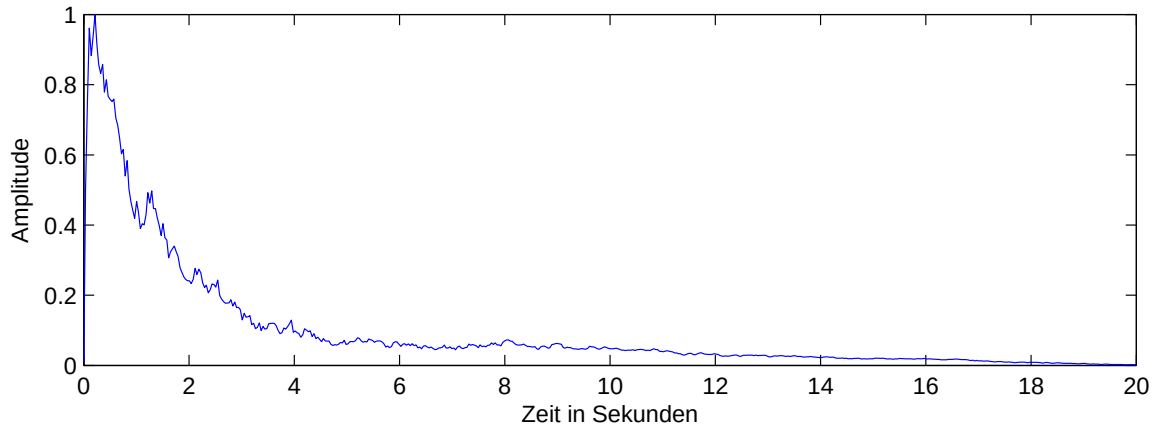

Abbildung 2.5: Hüllkurve eines Klaviertons.

Allgemeinen keine Abklingphase, da dem System kontinuierlich Energie zugeführt wird (siehe Abb. 2.6 auf der folgenden Seite).

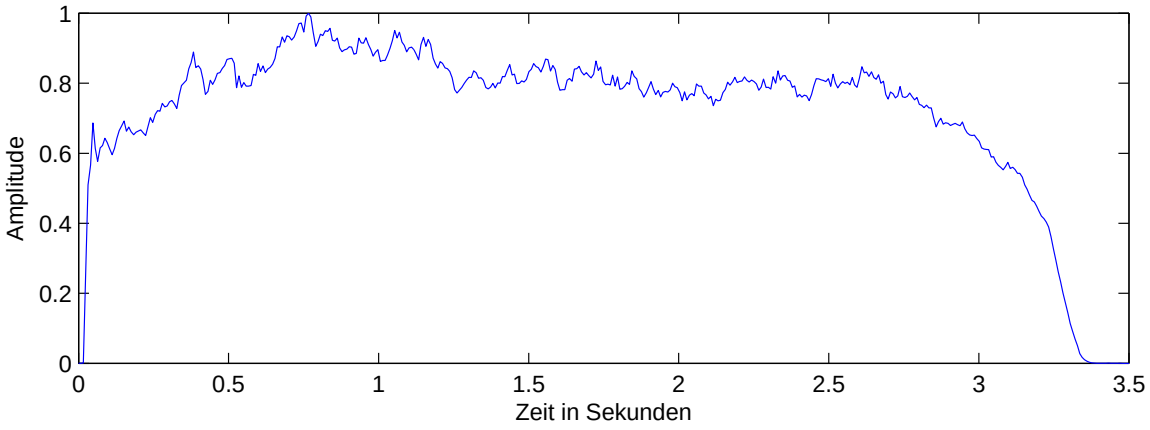

Abbildung 2.6: Hüllkurve eines Trompetentons.

In den folgenden Abschnitten sollen nun verschiedene Modelle betrachtet werden, die zum Modellieren von Klängen verwendet werden können.

2.1 Sinusoidal-Modell

Ein sehr allgemeines und oft in der Literatur verwendetes Modell, ist das *Sinusoidal-Modell* [22, 49, 61, 15, 14, 10, 12, 13, 42]. Hierbei wird der Klang als eine Summe von P gewichteten Sinusschwingungen dargestellt, was einer *additiven Synthese* entspricht.

Dies kann allgemein beschrieben werden als

$$y(t) = \sum_{p=1}^{P} a_p(t) \cdot \sin\left(2\pi \int_{\tau=0}^{t} f_p(\tau)\delta\tau + \phi_{0,p}\right) \tag{2.2}$$

wobei $\phi_{0,p}$ der Phasenoffset und $a_p(t)$ und $f_p(t)$ respektive die vom Zeitpunkt t abhängige Momentanamplitude und -frequenz der p-ten Sinusschwingung sind.

2.1.1 Harmonische Schwingung

Handelt es sich bei dem zu modellierenden Klang um eine *harmonische Schwingung*, so lassen sich die einzelnen Frequenzen der *Obertöne* relativ zu einem *Grundton* f_0 als ganzzahlige Vielfache dessen ausdrücken

$$f_p = p \cdot f_0, \quad p \in \mathbb{N}, \tag{2.3}$$

wobei f_p die Frequenz des p-ten *Teiltons* bzw. der $(p-1)$-ten Obertons ist.

2.1.2 Quasi-harmonische Schwingung

Nicht alle Musikinstrumente besitzen reine harmonische Obertöne. Bei Saiteninstrumenten wie z.B. dem Klavier sind die Obertöne aufgrund der Saitensteifheit leicht nach oben verstimmt.

Der 40te Teilton kann sogar zwei Ganztöne höher als der harmonische Oberton sein [22, §3.3.3]. Schwingungen, deren Obertöne nicht exakt ganzzahlige Vielfache des Grundtons sind, nennt man *quasi-harmonische Schwingungen*. Diese Eigenschaft lässt sich durch den *Inharmonizitätskoeffizienten* β ausdrücken [13]

$$f_p = p \cdot f_0 \sqrt{\frac{1 + p^2\beta}{1 + \beta}}. \tag{2.4}$$

Eine alternative Definition ist in [22, §8.2] und [54][1] gegeben

$$f_p = p \cdot f_0 \sqrt{1 + p^2\beta}, \tag{2.5}$$

wobei bei dieser Definition f_1 nicht exakt dem Grundton f_0 entspricht.

Werte für β wurden unter Verwendung von (2.5) in [22, §8.2] exemplarisch für Violine, Klavier, Trompete und Flöte zu $1{,}2 \cdot 10^{-6}$, $3{,}4 \cdot 10^{-4}$, $3{,}6 \cdot 10^{-5}$ und $6{,}6 \cdot 10^{-4}$ bestimmt, wobei die Werte für die Flöte aufgrund zu starken Rauschens als fehlerhaft angegeben sind.

Die Inharmonizität bei Saiteninstrumenten [44, §2]

$$\beta = \frac{\pi^3 Q d^4}{64 l^2 T} \tag{2.6}$$

ist abhängig von dem *Elastizitätsmodul* (Young's Modul) Q, dem Durchmesser d, der Länge l und der Spannung T der Saite.

Der Schwellwert für gerade noch wahrnehmbare Inharmonizität wurde in [54][2] mit

$$\beta_{thresh} = e^{2{,}54 \log(f_0) - 24{,}6} \tag{2.7}$$

angegeben.

2.1.3 Zeitvariante Amplitude

Das Sinusoidal-Modell in (2.2) hat das Problem, dass tiefe Frequenzen, die über die Sinusfunktion modelliert werden sollte, ebenso über die zeitvarianten Amplituden höherer Frequenzen modelliert werden können. Abhilfe dafür schafft die Projektion der Amplitude auf eine kleine Menge von I Basisfunktionen ϕ_i [12, §2.1]

$$a_p(t) = \sum_{i=0}^{I} a_{p,i} \phi_i(t). \tag{2.8}$$

Dabei ist ϕ_i eine zeitverschobene Version einer nicht oszillierenden Funktion $\phi(\tau)$ wie z.B. einem Spline oder ein Fensterfunktion wie dem Hammingfenster oder Gaussfenster

$$\phi_i(t) = \phi(t - i\Delta_t) \tag{2.9}$$

mit der Schrittweite $\Delta_t = \frac{N-1}{I}$ und der Signallänge N.

1 http://ccrma.stanford.edu/~jos/pasp/Stiff_String_Dispersion_Filter_Design.html
2 http://ccrma.stanford.edu/~jos/pasp/Stiff_String_Dispersion_Filter_Design.html

2.2 ARMA Modell

Ein autoregressive moving-average Modell (ARMA) lässt sich durch seine Übertragungsfunktion $H(z)$ wie folgt beschreiben

$$H(z) = \frac{Z(z)}{N(z)} = \frac{\sum_{\mu=0}^{m} b_\mu z^\mu}{\sum_{\nu=0}^{n} c_\nu z^\nu} \tag{2.10}$$

Nimmt man ein kausales Modell an, so muss $m \leq n$ gelten. Im Folgenden wird ohne Einschränkung der Allgemeingültigkeit $m = n$ angenommen. Außerdem setzen wir $c_n = 1$.

Im Zeitbereich lässt sich ein kausales ARMA-Modell durch die Zustandsgleichung mit dem Anfangszustand $\underline{x}(k_0)$ darstellen [48]

$$\underline{x}(k+1) = \mathbf{A}(k)\underline{x}(k) + \mathbf{B}(k)\underline{v}(k), \quad k \geq k_0 \tag{2.11}$$

$$\underline{y}(k) = \mathbf{C}(k)\underline{x}(k) + \mathbf{D}(k)\underline{v}(k), \quad k \geq k_0, \tag{2.12}$$

wobei $\underline{x}(k)$ und $\underline{x}(k+1)$ die n-dimensionale Zustandsvektoren zum aktuellen und darauf folgenden Zeitpunkt sind. $\underline{y}(k)$ ist der r-dimensionale Ausgabevektor und $\underline{v}(k)$ der l-dimensionale Eingangsvektor. Die Matrizen $\mathbf{A}(k)$, $\mathbf{B}(k)$, $\mathbf{C}(k)$ und $\mathbf{D}(k)$ haben die Dimensionen $n \times n$, $n \times l$, $r \times n$ und $r \times l$.

Für die Umsetzung gibt es vier verschiedene kanonische Strukturen. Die Belegung obiger Matrizen sei hier exemplarisch für die *erste kanonische Struktur* mit $l = r = 1$ gezeigt

$$\mathbf{A} = \begin{pmatrix} -c_{n-1} & 1 & 0 & \cdots & 0 \\ -c_{n-2} & 0 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -c_1 & 0 & \cdots & 0 & 1 \\ -c_0 & 0 & \cdots & \cdots & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{n-1} & - & b_n c_{n-1} \\ b_{n-2} & - & b_n c_{n-2} \\ & \vdots & \\ b_1 & - & b_n c_1 \\ b_0 & - & b_n c_0 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}, \quad \mathbf{D} = b_n. \tag{2.13}$$

2.2.1 AR-Modell für Sinusschwingung

Nimmt man den einfachsten Fall an, dass ein Klang allein aus einer Sinusschwingung einer festen Frequenz f und dem Dämpfungsfaktor ρ besteht, so kann dies mittels eines AR-Modells zweiten Grades beschrieben werden. Damit das System eine Sinusschwingung mit der Frequenz f modelliert, muss die Übertragungsfunktion ein Maximum bei der Frequenz f haben. Da es sich hierbei um ein reellwertiges System handelt, gilt $H(z^*) = H^*(z)$. Somit wird das Maximum durch zwei Polstellen, die zueinander konjugiert komplex sind, beschrieben

$$z_{\infty 1} = z_{\infty 2}^* = \rho e^{i\psi} = \rho e^{i\omega T}, \quad \omega = 2\pi f, \quad T = \frac{1}{f_s}, \tag{2.14}$$

wobei ω die Kreisfrequenz und f_s die Abtastrate angibt. Dies ist in einem Pol-Nullstellendiagramm in Abb. 2.7 veranschaulicht.

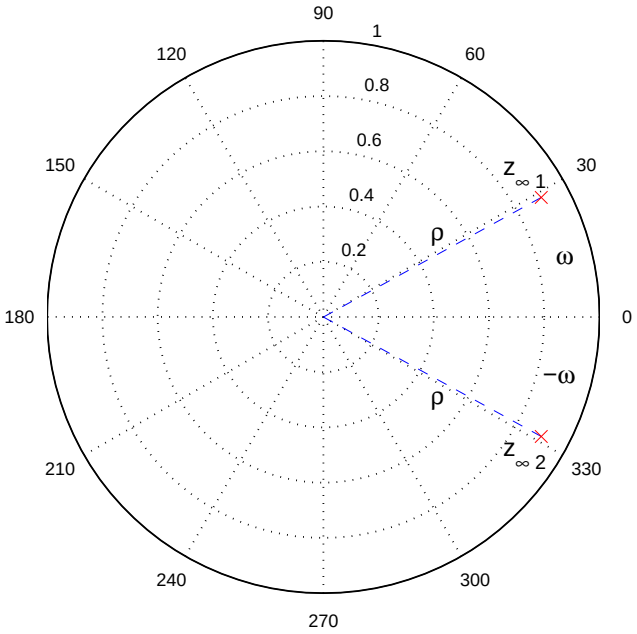

Abbildung 2.7: Pol-Nullstellendiagramm für eine Sinusschwingung mit der Frequenz $f = 3520$ Hz und der Dämpfung $\rho = 0{,}9$ bei einer Abtastrate $f_s = 44100$ Hz.

Für ein AR-Modell zweiten Grades ist der Zusammenhang zwischen den Systemparametern c_ν und den Polstellen $z_{\infty i}$ durch folgende zwei Gleichungen gegeben

$$c_0 = z_{\infty 1} z_{\infty 2} \tag{2.15}$$

$$c_1 = -(z_{\infty 1} + z_{\infty 2}). \tag{2.16}$$

Somit ergeben sich die Systemparameter für unseren Fall zu

$$c_0 = \rho^2, \quad c_1 = -2 \operatorname{Re} z_{\infty 1} = -2\rho \cos \psi. \tag{2.17}$$

Für eine auf die Anfangsamplitude 1 normierte Sinusschwingung setzen wir

$$b_0 = 0, \quad b_1 = \rho \sin \psi, \quad b_2 = 0. \tag{2.18}$$

Somit wird das System in der ersten kanonischen Struktur durch die folgende Matrizen beschrieben

$$\mathbf{A} = \begin{pmatrix} 2\rho\cos\psi & 1 \\ -\rho^2 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \rho \sin \psi \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \mathbf{D} = 0. \tag{2.19}$$

Dieses Modell ist in Abb. 2.8 auf der folgenden Seite grafisch veranschaulicht.

2.2.2 AR-Modell für summierte Sinusschwingungen

Mittels des Modells aus dem vorigen Abschnitt kann nun auf einfachste Weise ein Modell zusammengestellt werden, was aus mehreren Sinusschwingungen besteht, ähnlich wie in [10, §5.2.1].

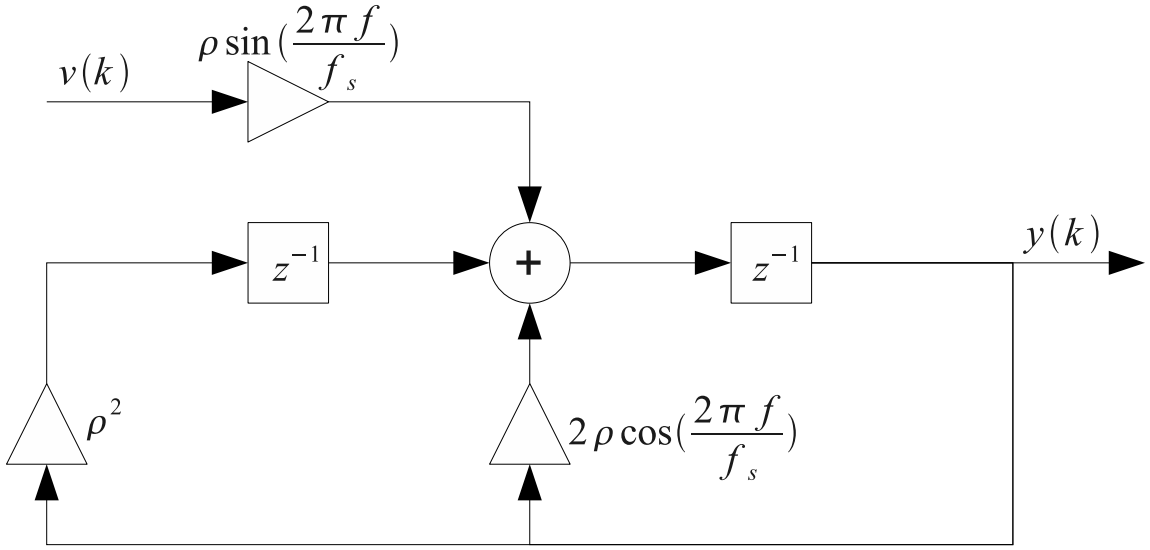

Abbildung 2.8: Strukuturdiagramm eines LTI-Systems einer gedämpften Sinusschwingung.

Dafür setzt man die Transitionsmatrizen $\mathbf{A}_p$ der einzelnen Sinusmodelle zu einer Blockdiagonalmatrix $\mathbf{A}$ zusammen. Ebenso werden die Matrizen $\mathbf{B}_p$ und $\mathbf{C}_p$ jeweils aneinander gesetzt, sowie die einzelnen Zustandsvektoren $\underline{x}_p(k)$

$$\underline{x}(k) = \begin{pmatrix} \underline{x}_1(k) \\ \vdots \\ \underline{x}_p(k) \\ \vdots \\ \underline{x}_P(k) \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{A}_p & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{A}_P \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_p \\ \vdots \\ \mathbf{B}_P \end{pmatrix} \mathbf{C} = \begin{pmatrix} \mathbf{C}_1^T \\ \vdots \\ \mathbf{C}_p^T \\ \vdots \\ \mathbf{C}_P^T \end{pmatrix}^T . \quad (2.20)$$

Dabei bleibt $\mathbf{D} = 0$.

Wie man leicht sehen kann ist der Ausgabevektor $\underline{y}(k)$ nun die Summe der Ausgabevektoren $\underline{y}_p(k)$ der einzelnen Sinusmodelle

$$\underline{y}(k) = \mathbf{C}\underline{x}(k) = \sum_{p=1}^{P} \mathbf{C}_p \underline{x}_p(k) = \sum_{p=1}^{P} \underline{y}_p(k). \quad (2.21)$$

Dies entspricht der *vierten kanonischen Struktur.*

Im weiteren Verlauf werden wir dieses Modell allerdings nicht weiter betrachten, da für einen authentischen Klang zusätzlich noch der Einschwingvorgang modelliert werden müsste, sowie für jede Sinusschwingung separat ein Dämpfungsfaktor zu bestimmen wäre.

2.3 Physikalisches Modell

Eine weitere Möglichkeit eine Klang zu modellieren, ergibt sich beim *Physical Modelling.* Hier werden die physikalischen Eigenschaften eines Instruments modelliert. Beispielsweise für eine

einfache Gitarrensaite kann diese Modellierung grob in drei Blöcke unterteilt werden: Die Anregung durch ein Plektrum oder Finger, die Saite selbst und der Gitarrenkorpus, der den Klang dann abstrahlt (siehe Abb. 2.9).

Abbildung 2.9: Schematische Darstellung der Synthese eines Saiteninstruments.

2.3.1 Kommutierte Synthese

Diese Kette lässt sich vereinfachen, wenn man das Modell für Saite $S(z)$ und Korpus $B(z)$ jeweils durch ein *linear, zeitinvariantes (LTI) System* approximiert. In diesem Fall können die beiden Teilsysteme für Saite und Korpus aufgrund der Kommutativität der Faltung vertauscht werden. Somit kann man die Anregung zusammen mit dem Korpus als Aggregatanregung $A(z)$ modellieren, indem man das Anregungssignal $e(k)$ mit der Impulsantwort des Korpus' $b(k)$ faltet [54][3] (siehe Abb. 2.10). Dies nennt man auch *kommutierte Synthese*.

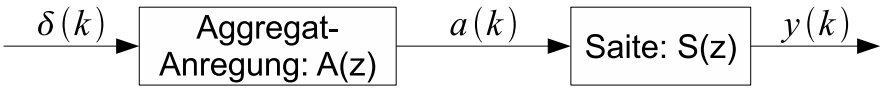

Abbildung 2.10: Schematische Darstellung der kommutierten Synthese eines Saiteninstruments mit Aggregatanregung.

Wenn man weiter auch $E(z)$ mit einem LTI System approximiert, dann kann man auch Anregung und Korpus miteinander vertauschen. Somit kann man als Eingang für das Anregungssystem statt eines Einheitsimpulses die Impulsantwort des Korpus' $b(k)$ verwenden [25, §2.3] (siehe Abb. 2.11).

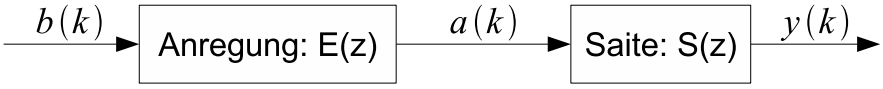

Abbildung 2.11: Schematische Darstellung der kommutierten Synthese eines Saiteninstruments.

2.3.2 Karplus-Strong Modell

Eines der einfachsten Modelle für eine ideale Saite mit einfacher Saitendämpfung ist das *Karplus-Strong-Modell* [25, §2.2] [54][4] (siehe Abb. 2.12 auf der folgenden Seite). Dies besteht aus einer

3 http://ccrma.stanford.edu/~jos/pasp/Commuted_Synthesis_Strings.html
4 http://ccrma.stanford.edu/~jos/pasp/Karplus_Strong_Algorithm.html

Schleife bestehend aus einem *Verzögerungskette* (engl. Delayline) der Länge L und einem *Schleifenfilter* (engl. Loopfilter) $H_{loop}(z) = 0{,}5 + 0{,}5z^{-1}$, der als Mittelwert der letzten beiden Sample in der Verzögerungskette realisiert ist. Die Übertragungsfunktion dieses Saitenmodells ist somit [29, §2] [26]

$$S(z) = \frac{1}{1 - z^{-L} H_{loop}(z)}. \tag{2.22}$$

Angeregt wird das System durch einen *Noiseburst* $\nu(k)$, also einer kurzen Rauschsequenz. Die Länge L der Verzögerungskette wird in Abhängigkeit der Grundfrequenz f_0 gewählt. Da das Schleifenfilter eine *Phasenlaufzeit* eines halben Samples hinzufügt, ergibt sich der Zusammenhang zwischen Länge L der Verzögerungskette und der Grundfrequenz f_0 etwa als [55][5]

$$f_0 \approx \frac{f_s}{L + \frac{1}{2}} \tag{2.23}$$

mit der Abtastrate f_s. Das Schleifenfilter sorgt dafür, dass die Schwingung frequenzabhängig gedämpft wird.

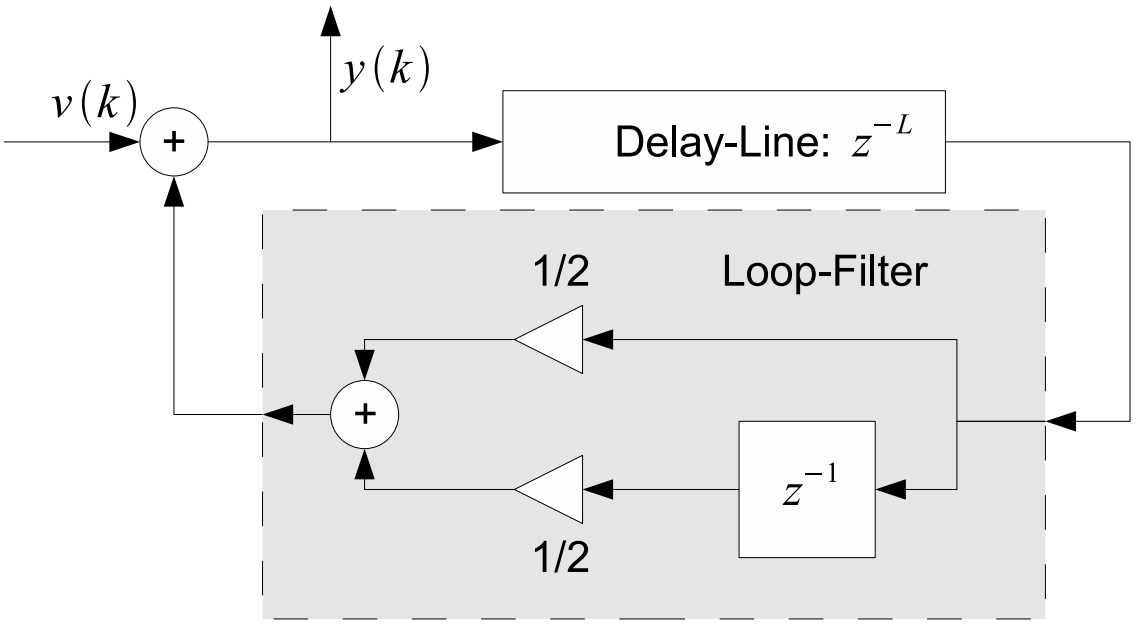

Abbildung 2.12: Strukturdiagramm für das Karplus-Strong Modell.

Dieses Modell kann als Spezialfall des *Digitalen Waveguide Modells* hergeleitet werden [54][6] und entspricht diesem, wenn zusätzlich noch ein Kammfilter für die Zupfposition davor geschaltet wird, um das Spektrum des Noisebursts zu färben [25, §2.1].

Ein ähnliches Modell auf Basis einer *Filterstruktur mit unendlicher Impulsantwort* (engl. Infinite Impulse Response, IIR) wurde in [59] vorgestellt.

5 http://ccrma.stanford.edu/realsimple/faust_strings/Karplus_Strong_Digitar_Algorithm.html
6 http://ccrma.stanford.edu/~jos/pasp/Karplus_Strong_Algorithms.html

Zustandsdarstellung

Das Karplus-Strong Modell kann nun mit den folgenden Parametern in der Zustandsdarstellung formuliert werden

$$\mathbf{A} = \begin{pmatrix} \mathbf{0}^{1\times(L-1)} & 0{,}5 & 0{,}5 \\ \mathbf{I}^{L\times L} & & \mathbf{0}^{L\times 1} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 \\ \mathbf{0}^{L\times 1} \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & \mathbf{0}^{1\times L} \end{pmatrix}, \quad \mathbf{D} = 0. \tag{2.24}$$

Fractional Delay

Im allgemeinen genügt eine ganzzahlige Länge der Verzögerungskette nicht, um eine gewünschte Grundfrequenz korrekt zu modellieren. Vor allem bei einer niedrigen Abtastrate bzw. einer hohen Grundfrequenz, kann sich so der Ton verstimmt anhören. Abhilfe dafür schafft entweder ein Anpassen der Abtastrate, was allerdings nicht praktibel ist und bei parallel zu synthetisierenden Tönen gar nicht funktioniert, oder eine *Fractional Delay Approximation*.

Hierfür wird der Abgreifpunkt am Ende der Verzögerungskette interpoliert [54][7], um Zwischenwerte zu erhalten, so dass man die eigentlich gewünschte Periodenlänge erhält. Die einfachste Methode, dies umzusetzen, ist eine *Lagrange Interpolation* ersten Grades, was einer *linearen Interpolation* entspricht [54][8]. Die Übertragungsfunktion dafür lässt sich wie folgt schreiben [53, §2.5]

$$H_\eta(z) = (1-\eta) + \eta z^{-1}, \tag{2.25}$$

wobei $\eta \in [0,1]$ der gewünschte Bruchteil eines Samples ist. Die Amplitudenantwort und Phasenlaufzeit sowie die Impulsantwort dieser Interpolation sind in Abb. 2.13 auf der nächsten Seite exemplarisch für $\eta = 0{,}5$ abgebildet.

Daraus ergibt sich die erweiterte Übertragungsfunktion für das Saitenmodell [60, §1]

$$S(z) = \frac{1}{1 - z^{-L} H_\eta(z) H_{loop}(z)} \tag{2.26}$$

und die modifizierten Parameter für die Zustandsdarstellung

$$\mathbf{A} = \begin{pmatrix} \mathbf{0}^{1\times(L-1)} & 0{,}5(1-\eta) & 0{,}5\eta & 0{,}5 \\ \mathbf{I}^{L\times L} & & \mathbf{0}^{L\times 2} & \\ \mathbf{0}^{1\times(L-1)} & 1-\eta & \eta & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 \\ \mathbf{0}^{(L+1)\times 1} \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 1 & \mathbf{0}^{1\times(L+1)} \end{pmatrix}. \tag{2.27}$$

2.3.3 Erweitertes Karplus-Strong Modell

Das *erweiterte Karplus-Strong Modell* erweitert, wie der Name schon sagt, das Karplus-Strong Modell um mehrere Elemente, um das Modell vielfältiger verwenden zu können [54][9]. Die Struktur dieser Verallgemeinerung ist in Abb. 2.14 auf der folgenden Seite gezeigt. Hier wird nun

7 http://ccrma.stanford.edu/~jos/pasp/Delay_Line_Interpolation.html
8 http://ccrma.stanford.edu/~jos/pasp/Linear_Interpolation.html
9 http://ccrma.stanford.edu/~jos/pasp/Extended_Karplus_Strong_Algorithm.html

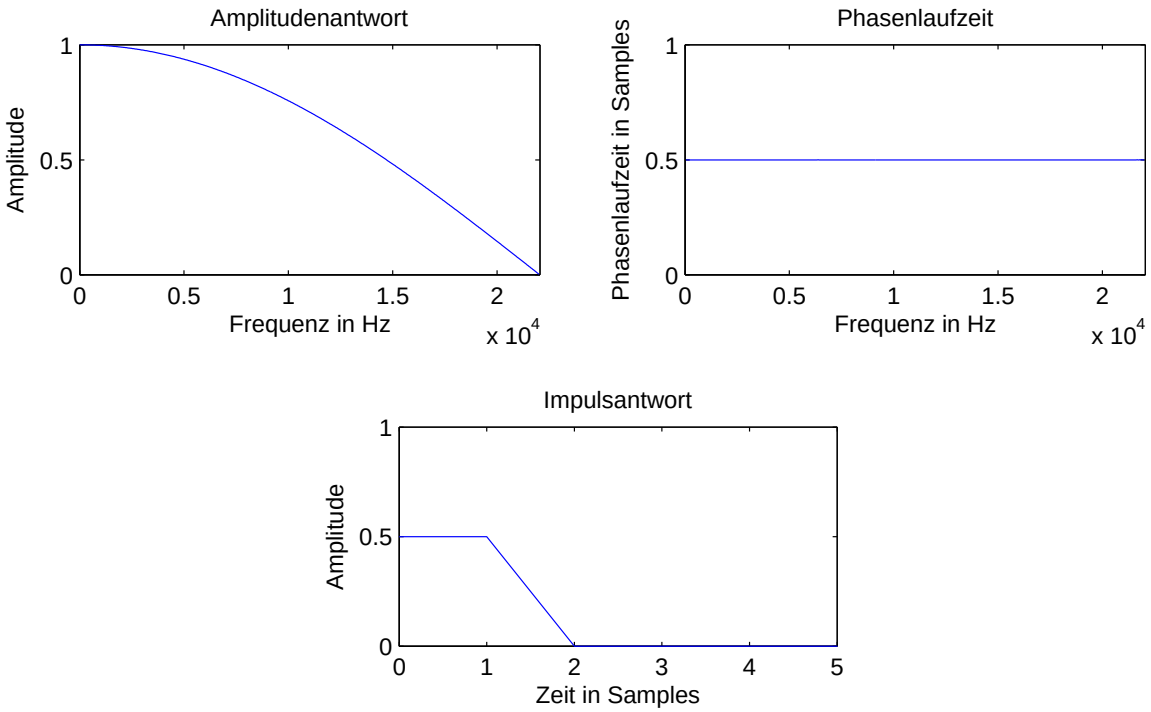

Abbildung 2.13: Amplitudenantwort, Phasenantwort und Impulsantwort einer linearen Interpolation für $\eta = 0{,}5$.

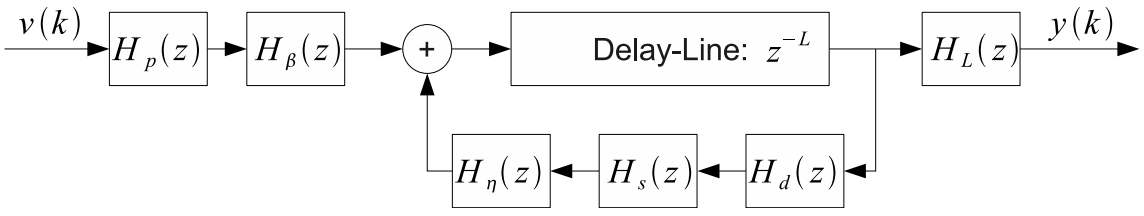

Abbildung 2.14: Strukturdiagramm für das erweiterte Karplus-Strong Modell.

zusätzlich für die Zupfrichtung ein Tiefpassfilter $H_p(z)$ und für die Zupfposition ein Kammfilter $H_\beta(z)$ verwendet. Desweiteren ist das Schleifenfilter nun in ein Saitendämpfungsfilter $H_d(z)$, ein Allpassfilter $H_s(z)$ für die Saitensteifheit und ein weiteres Allpassfilter erster Ordnung $H_\eta(z)$ für die Saitenstimmung zerlegt. Am Ausgang wird dann noch ein Tiefpassfilter $H_L(z)$ für die Dynamik angehängt.

Da alle Filter in der Rückkopplungsschleife eine gewisse Verzögerung des Signals zur Folge haben, muss die Länge der Verzögerungskette entsprechend diesen Bedingungen angepasst sein, so dass die Gesamtverzögerung der Komponenten in der Rückkopplungsschleife der gewünschten Periodenlänge entspricht.

Zupfrichtungsfilter

Für das *Zupfrichtungsfilter* wird ein *einpoliges Tiefpassfilter* (engl. One-Pole Low-Pass Filter) verwendet, der den Gleichstromanteil ungedämpft durchlässt [55][10]

$$H_p(z) = \frac{1-p}{1-pz^{-1}}, \tag{2.28}$$

wobei $p \in [0,1)$ die unterschiedlichen Winkel, mit der eine Saite bei der Auf- bzw. Abwärtsbewegung des Plektrums angeschlagen wird, simulieren soll, da sich abhängig von diesem Winkel unter anderem die Anschlagstärke ändert. Somit werden also zwei verschiedene Werte für p verwendet, entsprechend der Zupfrichtung. In [54][11] wird zum Beispiel $p_1 = 0$ für eine Richtung und $0 < p_2 < 1$ für die andere Richtung vorgeschlagen. In Abb. 2.15 ist exemplarisch für $p = 0{,}5$ die Amplitudenantwort und Impulsantwort des Filters gezeigt.

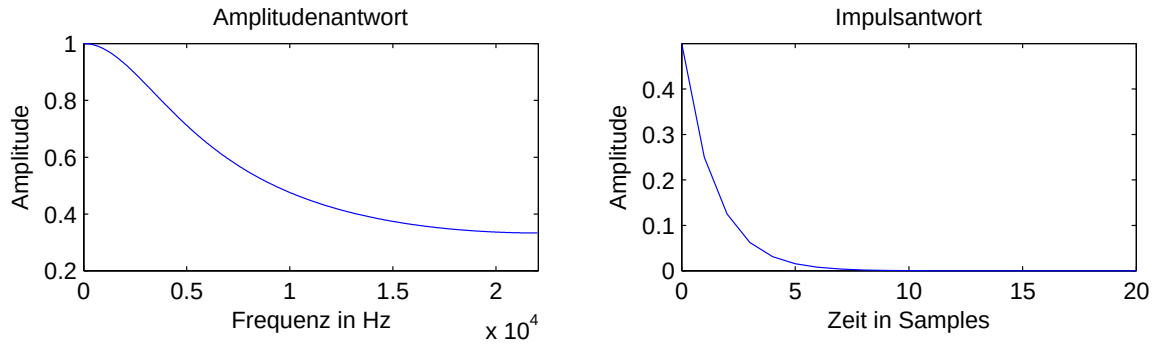

Abbildung 2.15: Amplitudenantwort und Impulsantwort eines Zupfrichtungsfilters für $p = 0{,}5$.

10 http://ccrma.stanford.edu/realsimple/faust_strings/Pick_Direction_Lowpass_Filter.html
11 http://ccrma.stanford.edu/~jos/pasp/Extended_Karplus_Strong_Algorithm.html

Zupfpositionsfilter

Für die Modellierung der Zupfposition wird ein Kammfilter verwendet [55][12] [53, §2.1]. Dieser modelliert die Tatsache, dass einige Teiltöne abhängig von der Zupfposition weniger bis gar nicht angeregt werden.

Satz 2.1 *Alle natürlichen Moden, die einen Knoten an der Zupfposition haben, werden durch das Zupfen nicht angeregt, da die Anregung einer Mode proportional zu ihrer Bewegungsfreiheit am Anregungspunkt ist [19, §10.2].*

Die Übertragungsfunktion dieses Filters lautet

$$H_\beta(z) = 1 - z^{-\lfloor \beta P \rfloor} \tag{2.29}$$

wobei P die Periodenlänge in Samples und $\beta \in (0,1)$ die normalisierte Zupfposition zwischen Steg ($\beta = 0$) und Sattel ($\beta = 1$) angibt [64]. Die Amplitudenantwort und Impulsantwort eines Zupfpositionsfilters für $\beta = 0{,}3$ und $P = 100{,}2273$ ($f_0 = 440$ Hz) ist exemplarisch in Abb. 2.16 abgebildet.

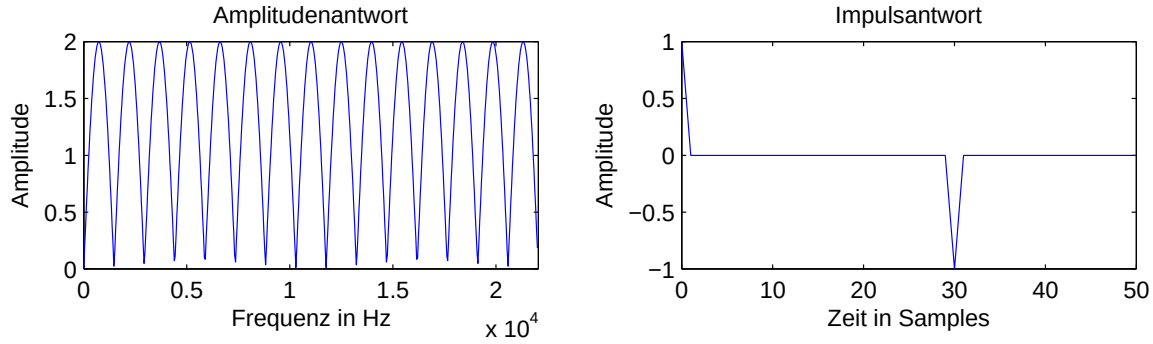

Abbildung 2.16: Amplitudenantwort und Impulsantwort eines Zupfpositionsfilters mit der normalisierten Zupfposition $\beta = 0{,}3$ für die Grundfrequenz $f_0 = 440$ Hz und die Abtastrate $f_s = 44100$ Hz.

Saitendämpfungsfilter

Ein ideales Saitendämpfungsfilter ist ein nullphasiges oder linearphasiges [54][13] Tiefpassfilter. Er hat also einen frequenzabhängigen Dämpfungsfaktor $|H_d(z)| \leq 1$, der für eine exponentiell gedämpfte Schwinung sorgt, und eine konstante Verzögerung für alle Frequenzen.

Symmetrisches Filter mit endlicher Impulsantwort Ein symmetrisches *Filter mit endlicher Impulsantwort* (engl. Finite Impulse Response Filter, FIR) ist für den Saitendämpfungsfilter die erste Wahl, da damit ein stabiles, linearphasiges Filter garantiert ist.

12 http://ccrma.stanford.edu/realsimple/faust_strings/Pick_Position_Comb_Filter.html
13 http://ccrma.stanford.edu/~jos/pasp/Damping_Filter_Design.html

Die Übertragungsfunktion eines allgemeinen FIR-Filters M-ter Ordnung der Länge $M+1$ ist gegeben durch [57][14]

$$H(z) = \sum_{n=0}^{M} b_n z^{-n}. \tag{2.30}$$

Da hier aber ein symmetrisches Filter verwendet wird, reduziert sich der Freiheitsgrad der Filterparameter b_n auf die Hälfte, da $b_n = b_{M-n}$.

Die Amplitudenantwort ergibt sich somit für ein FIR-Filter mit geradzahliger Ordnung M zu [54][15]

$$\left|H_d\left(e^{i\omega T}\right)\right| = h_0 + 2\sum_{n=1}^{M/2} h_n \cos(\omega n T), \tag{2.31}$$

wobei $h_n = b_{M/2+n}$.

Das einfachste nicht-degenerierte Schleifenfilter ergibt sich somit als FIR-Filter der Länge 3 und hat die Amplitudenantwort [54][16]

$$\left|H_d\left(e^{i\omega T}\right)\right| = h_0 + 2h_1 \cos(\omega T) \tag{2.32}$$

und die Übertragungsfunktion [53, §2.3]

$$H_d(z) = h_1 + h_0 z^{-1} + h_1 z^{-2} = z^{-1}(h_0 + h_1(z + z^{-1})). \tag{2.33}$$

Dieses Filter fügt eine frequenzunabhängige Phasenlaufzeit von einem Sample

$$\Delta_d = 1 \tag{2.34}$$

in die Schleife ein, was in der Verzögerungskette entsprechend kompensiert werden muss.

Da die Dämpfung mit zunehmender Frequenz zunimmt ergibt sich die Beschränkung der Parameter h_n zu

$$h_0 \geq 2h_1 > 0. \tag{2.35}$$

Die Amplitudenantwort beschreibt dann einen "angehobenen Kosinus".

Eine intuitivere Beschreibung der Filterparameter erhält man, indem man sie durch die Parameter Klanghelligkeit (engl. brightness) B und Haltedauer (engl. sustain) S beschreibt [54][17]

$$\begin{aligned} \rho &= e^{-6{,}91\frac{\lambda}{S}} \\ h_0 &= \rho\frac{1+B}{2} \\ h_1 &= \rho\frac{1-B}{4}. \end{aligned} \tag{2.36}$$

Dabei gibt $\lambda = \frac{1}{f_0}$ die Periodenlänge und S die Dauer, bis das Signal bei maximaler Klanghelligkeit auf -60 dB abgedämpft ist, in Sekunden an. $B \in [0,1]$ gibt das Verhältnis an, inwiefern

14 http://ccrma.stanford.edu/~jos/filters/FIR_Transfer_Function.html
15 http://ccrma.stanford.edu/~jos/pasp/Frequency_Dependent_Damping.html
16 http://ccrma.stanford.edu/~jos/pasp/Length_FIR_Loop_Filter.html
17 http://ccrma.stanford.edu/~jos/pasp/Length_FIR_Loop_Filter_I.html

hohe Frequenzen schneller abgedämpft werden als tiefe Frequenzen, und ρ die frequenzunabhängige Gesamtdämpfung. In Abb. 2.17 ist die Amplitudenantwort und Impulsantwort eines solchen Saitendämpfungsfilters exemplarisch mit den Parametern $B = 0{,}3$ und $S = 4$ s für die Grundfrequenz $f_0 = 440$ Hz dargestellt.

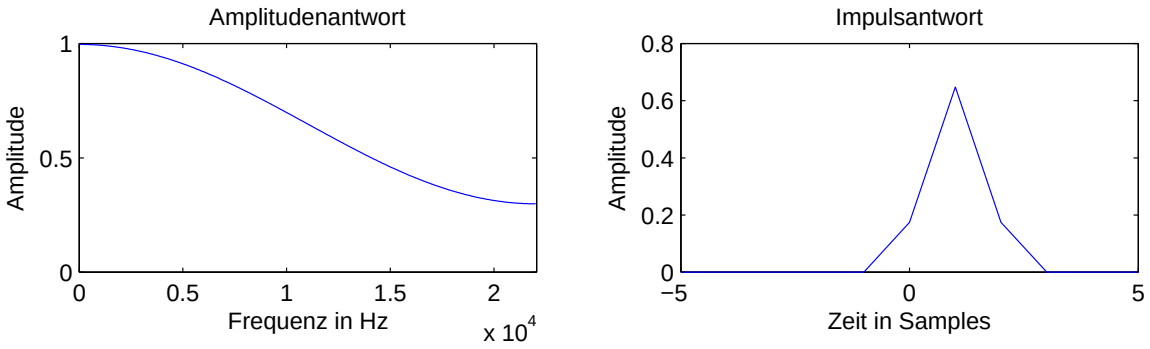

Abbildung 2.17: Amplitudenantwort und Impulsantwort eines FIR-Filters als Saitendämpfungsfilter mit der Klanghelligkeit $B = 0{,}3$ und der Haltedauer $S = 4$ s für die Grundfrequenz $f_0 = 440$ Hz und die Abtastrate $f_s = 44100$ Hz.

Einpolfilter Eine Alternative für die Wahl des Saitendämpfungsfilter ist z.B. ein Einpolfilter [60, 16, 4, 3]. Dieses wird durch die Übertragungsfunktion

$$H_d(z) = \frac{\rho(1+a)}{1+az^{-1}} \tag{2.37}$$

beschrieben, wobei ρ auch hier die Gesamtdämpfung angibt und $a < 0$ einem Cutoff Parameter entspricht. Allerdings ist hierbei darauf zu achten, dass das Filter *minimalphasig* gewählt wird und die Abweichung von der linearen Phase im Allpassfilter des Saitensteifheitsfilters wieder ausgeglichen wird, da es ansonsten zu ungewollten Verstimmungen der Obertöne kommt. In Abb. 2.18 sind sowohl die Amplitudenantwort und Phasenantwort als auch die Impulsantwort eines Einpolfilters als Saitendämpfungsfilter dargestellt. Dabei wurde dieselbe Gesamtdämpfung $\rho = 0{,}9961$ verwendet, wie sie für das Beispiel des FIR-Filters ermittelt wurde, und der Cutoff Parameter auf $a = -0{,}5$ gesetzt.

Die Verallgemeinerung des Einpolfilters, nämlich das *Allpolfilter*, wurde in [25, §3.3] betrachtet. Es wurde aber eher ein IIR-Filter erster bzw. zweiter Ordnung empfohlen. Allerdings gelten auch für diese Filter die gleichen Bedingungen wie für das Einpolfilter.

Saitensteifheitsfilter

Das Saitensteifheitsfilter simuliert das Verhalten einer Saite abhängig von ihrer Steifheit, durch welche die Frequenzen der Teiltöne leicht nach oben verstimmt werden. Ein ideales Saitensteifheitsfilter ist ein Allpass Filter [54][18]. Es verändert also nicht die Amplitude, sondern verzögert

18 http://ccrma.stanford.edu/~jos/pasp/Dispersion_Filter_Design.html

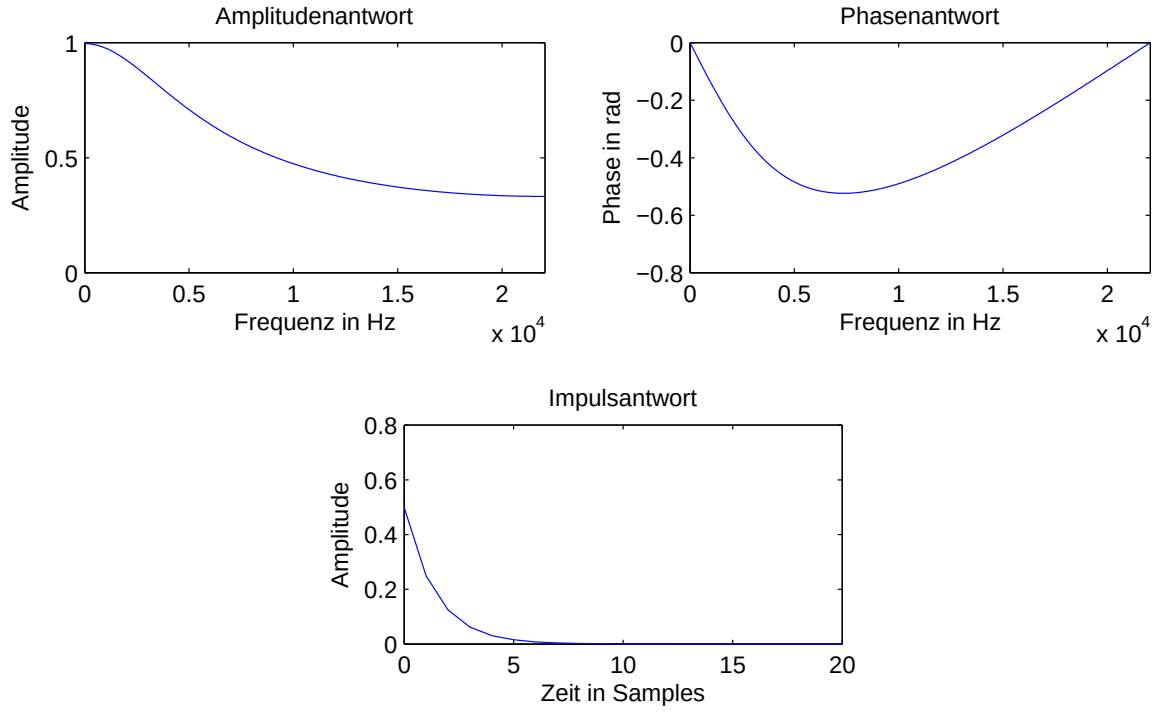

Abbildung 2.18: Amplitudenantwort, Phasenantwort und Impulsantwort eines Einpolfilters als Saitendämpfungsfilter mit dem Cutoff Parameter $a = -0{,}5$ und der Gesamtdämpfung $\rho = 0{,}9961$ für die Grundfrequenz $f_0 = 440$ Hz und die Abtastrate $f_s = 44100$ Hz.

nur frequenzabhängig das Signal entsprechend einer vorgegebenen Inharmonizität. Dafür ist in der Regel ein Filter höherer Ordnung nötig.

Kaskade von M identischen Allpassfiltern zweiter Ordnung In [45] wurde dafür eine Kaskade von M identischen Allpassfiltern zweiter Ordnung verwendet

$$H_s(z) = \left(\frac{a_2 + a_1 z^{-1} + z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \right)^M . \tag{2.38}$$

Die Parameter a_k werden nach dem *Thiran Allpassfilter Verfahren* wie folgt bestimmt [27] [45, §II.A]

$$a_k = (-1)^k \binom{N}{k} \prod_{n=0}^{N} \frac{D - N + n}{D - N + k + n}, \quad k = 1,2,\ldots,N, \tag{2.39}$$

wobei in diesem Fall der Grad der einzelnen Allpassfilter mit $N = 2$ gegeben ist, somit gilt konkret [27]

$$a_1 = -2\frac{D-2}{D+1} \tag{2.40}$$

$$a_2 = \frac{(D-1)(D-2)}{(D+1)(D+2)}. \tag{2.41}$$

Der zugehörige Parameter D gibt die Phasenlaufzeit für den Gleichstromanteil an [45, §II.D.2] und wird abhängig von der Grundfrequenz f_0, der Inharmonizität β und der gewünschten

Kaskadenordnung M gewählt

$$D(I_{key}, \beta) = e^{C_d(\beta) - I_{key} k_d(\beta)}, \tag{2.42}$$

wobei

$$I_{key}(f_0) = \log_{\sqrt[12]{2}} \frac{f_0 \sqrt[12]{2}}{27{,}5} \tag{2.43}$$

$$k_d(\beta) = e^{k_1 (\ln \beta)^2 + k_2 \ln \beta + k_3} \tag{2.44}$$

$$C_d(\beta) = e^{C_1 \ln \beta + C_2}. \tag{2.45}$$

Die in diesen Gleichungen verwendeten Konstanten k_1, k_2, k_3, C_1 und C_2 sind abhängig von der Kaskadenordnung M und sind in [45, §II.C] exemplarisch für $M = 1$ und $M = 4$ geschätzt worden.

Je nach Grundtonhöhe wurden für tiefere Frequenzen eine Kaskade von $M = 4$ und für höhere Frequenzen ein einzelnes Filter zweiter Ordnung verwendet ($M = 1$), um den, von der Grundfrequenz abhängigen, Inharmonizitäten gerecht zu werden.

In Abb. 2.19 ist sowohl die Phasenlaufzeit für diesen Filter, zusammen mit der optimalen, als auch die Impulsantwort exemplarisch für $M = 4$, $\beta = 10^{-4}$ und $f_0 = 440$ Hz dargestellt.

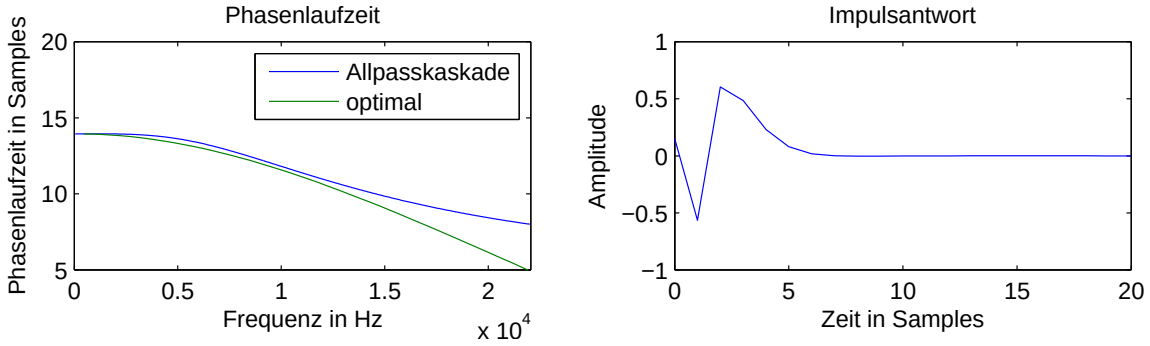

Abbildung 2.19: Phasenlaufzeit und Impulsantwort einer Kaskade von $M = 4$ identischen Allpassfiltern zweiter Ordnung als Saitensteifheitsfilter für die Inharmonizität $\beta = 10^{-4}$, die Grundfrequenz $f_0 = 440$ Hz und die Abtastrate $f_s = 44100$ Hz. Für die Phasenlaufzeit ist zusätzlich zu der des Filters noch die optimale Phasenlaufzeit eingezeichnet.

Kaskade von M identischen Allpassfiltern erster Ordnung In [44] wurde diese Struktur auf die Verwendung von Filter erster Ordnung erweitert

$$H_s(z) = \left(\frac{a_1 + z^{-1}}{1 + a_1 z^{-1}} \right)^M. \tag{2.46}$$

Für das Thiran Allpassfilter erster Ordnung ergibt sich der Parameter a_1 zu [44, §4]

$$a_1 = \frac{1 - D}{D + 1}. \tag{2.47}$$

Parameter	Wert
k_1	$-0{,}00179$
k_2	$-0{,}0233$
k_3	$-2{,}93$
m_1	$0{,}0126$
m_2	$0{,}0606$
m_3	$-0{,}00825$
m_4	$1{,}97$

Tabelle 2.1: Filterdesignkonstanten für die Schätzung des Saitensteifheitsfilters.

Auch in diesem Fall wird der Parameter D mittels (2.42)-(2.44) und

$$C_d(\beta, M) = e^{(m_1 \ln M + m_2)\ln\beta + m_3 \ln M + m_4} \tag{2.48}$$

berechnet, wobei hier die verwendeten Konstanten k_1, k_2, k_3, m_1, m_2, m_3 und m_4 unabhängig von M, wie in Tabelle 2.1 aufgelistet, gewählt werden [44, §3].

In Abb. 2.20 ist sowohl die Phasenlaufzeit für diesen Filter, zusammen mit der optimalen, als auch die Impulsantwort exemplarisch für $M = 8$, $\beta = 10^{-4}$ und $f_0 = 440$ Hz dargestellt.

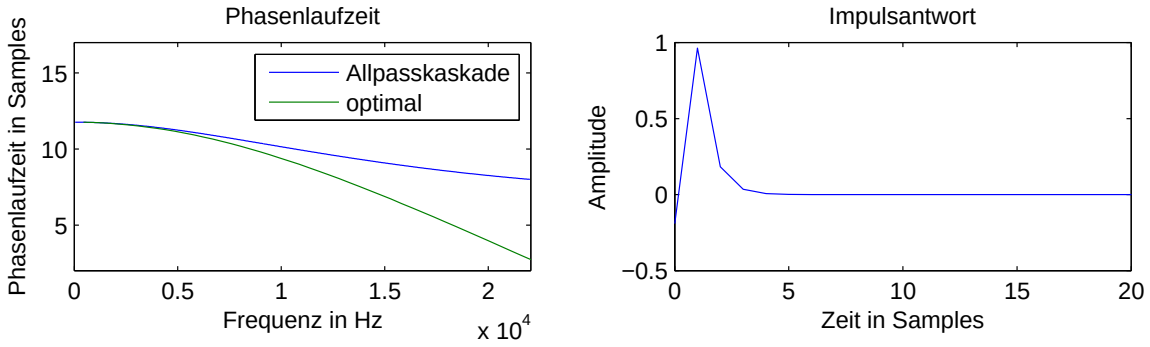

Abbildung 2.20: Phasenlaufzeit und Impulsantwort einer Kaskade von $M = 8$ identischen Allpassfiltern erster Ordnung als Saitensteifheitsfilter für die Inharmonizität $\beta = 10^{-4}$, die Grundfrequenz $f_0 = 440$ Hz und die Abtastrate $f_s = 44100$ Hz. Für die Phasenlaufzeit ist zusätzlich zu der des Filters noch die optimale Phasenlaufzeit eingezeichnet.

Beide Verfahren bieten somit eine einfache Methode, um mittels geschlossener Formeln ein Filter zu entwickeln. Die Phasenlaufzeit, welche durch diese beiden Varianten in die Rückkopplungsschleife eingefügt wird, kann für Kaskaden niedriger Ordnung etwa durch

$$\tilde{\Delta}_s = D \cdot M \tag{2.49}$$

approximiert werden [45, §II.A] [44, §4]. Für diese Approximation wird angenommen, dass für ein einzelnes Allpassfilter die Phasenlaufzeit der Grundfrequenz etwa der Phasenlaufzeit D für den Gleichstromanteil entspricht, da das Thiran Allpassfilter die Eigenschaft hat, an der Nullfrequenz eine maximal flache *Gruppenlaufzeit* zu haben. Solange dieser Approximationsfehler

kleiner ist als die *gerade noch wahrnehmbare Frequenzdifferenz* von 0,5% der gewünschsten Frequenz, ist der Fehler akzeptabel. Für höhere Ordnungen genügt allerdings diese Approximation nicht mehr, da in diesem Fall der Schätzfehler M mal so groß ist wie für ein einziges Allpassfilter [44, §4]. In diesem Fall muss die Phasenlaufzeit entsprechend ihrer Definition

$$\tau(\omega) \triangleq -\frac{\Theta(\omega)}{\omega}, \quad \omega = 2\pi f_0 \tag{2.50}$$

berechnet werden, wobei $\Theta(\omega) = \angle H_s\left(e^{i\omega T}\right)$ die Phasenantwort des Saitensteifheitsfilters und ω die Kreisfrequenz angibt. $\tau(\omega)$ gibt die Phasenlaufzeit in Sekunden an. Die gesuchte Phasenlaufzeit in Samples ergibt sich somit zu

$$\Delta_s = f_s \tau(\omega) \tag{2.51}$$

für die Abtastrate f_s.

Allpassfilter M-ter Ordnung In [4] wurde ein Verfahren von Rocchesso und Scalcon, basierend auf [28] präsentiert, um das Saitensteifheitsfilter mittels der *Methode der kleinsten Quadrate* für eine gegebene Phasenantwort zu schätzen. Das resultierende Allpassfilter hat, im Gegensatz zu den ersten beiden Verfahren, M verschiedene Koeffizienten. Dadurch lässt sich die Phasenantwort genauer modellieren.

Kaskade von mehreren unterschiedlichen komplexwertigen Allpassfiltern erster Ordnung Eine weitere Alternative, die es ermöglicht eine Kaskade von mehreren unterschiedlichen komplexwertigen Allpassfiltern erster Ordnung für eine gegebene Phasenantwort zu schätzen, wurde in [1] vorgestellt. Dafür wird die Phasenantwort in mehrere gleichgroße Bänder unterteilt und für jedes ein Allpassfilter erster Ordnung ermittelt. Diese Methode hat, laut den Autoren, vor allem auch den Vorteil, unabhängig von der Filterordnung numerisch robust zu sein, wohingegen andere Verfahren meist nur Filter niedriger Ordnung robust schätzen können.

Saitenstimmungsfilter

Für das Saitenstimmungsfilter kann prinzipiell eine Lagrange Interpolation, wie sie in Abschnitt 2.3.2 beschrieben wurde, verwendet werden. Allerdings hat die Lagrange Interpolation den Nachteil, dass sie in der Feedbackschleife des Saitenmodells zu Verstärkungsverzerrungen führt, wie in Abb. 2.13 auf Seite 18 zu sehen ist. Dies wiegt umso schwerer, wenn die Saitenschwingung kaum gedämpft ist, wie dies z.B. bei Stahlsaiten der Fall ist. In diesem Fall wird der Roll-Off der Amplitudenantwort des Interpolators nicht genügend maskiert, was zu hörbaren Verzerrungen führt [54][19].

19 http://ccrma.stanford.edu/~jos/pasp/Linear_Interpolation.html

Allpass Interpolation Eine Alternative, die zu keinen solchen Amplitudenverzerrungen führt ist die *Allpass Interpolation* [54][20]. Dieser Interpolator wird durch die Differenzengleichung

$$\hat{x}(k-\Delta_\eta) \triangleq y(k) = \eta \cdot [x(k) - y(k-1)] + x(k-1) \tag{2.52}$$

und die Übertragungsfunktion

$$H_\eta(z) = \frac{\eta + z^{-1}}{1 + \eta z^{-1}} \tag{2.53}$$

beschrieben. Dabei wird der Parameter η abhängig von der angestrebten Verzögerung Δ_η wie folgt gewählt

$$\eta \approx \frac{1-\Delta_\eta}{1+\Delta_\eta}. \tag{2.54}$$

Die Phasenlaufzeit und Impulsantwort einer Allpass Interpolation für $\Delta_\eta = 0{,}5$ ist exemplarisch in Abb. 2.21 dargestellt.

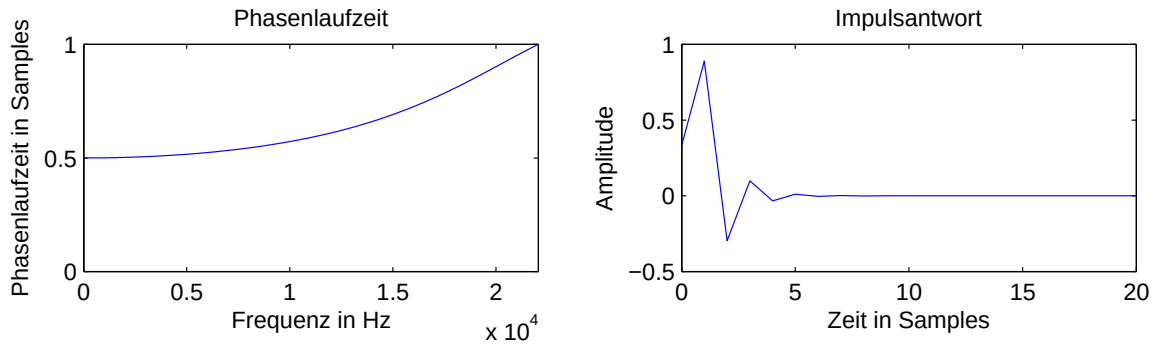

Abbildung 2.21: Phasenlaufzeit und Impulsantwort eines Allpassfilters als Saitenstimungsfilter für eine Verzögerung von $\Delta_\eta = 0{,}5$ Samples.

Die Verzögerung Δ_η hängt von der gewünschten Periodenlänge $\frac{f_s}{f_0}$, der Länge L der Verzögerungskette und den, durch die anderen Filter in der Rückkopplungsschleife verursachten, Verzögerungen Δ_d und Δ_s ab

$$\Delta_\eta = \frac{f_s}{f_0} - L - \Delta_d - \Delta_s, \tag{2.55}$$

wobei die Länge L der Verzögerungskette definiert ist als

$$L = \left\lfloor \frac{f_s}{f_0} - \Delta_d - \Delta_s \right\rfloor. \tag{2.56}$$

Allerdings hat die Allpass Interpolation den Nachteil, dass sie eine kleinen Verzögerungsfehler verursacht [54][21] und es nicht erlaubt, beliebig schnell sich verändernde Verzögerungen zu erlauben, da, aufgrund der Rekursivität des Interpolators, dieser eine gewisse Latenz hat, bevor er den stabilen Zustand erreicht. Wenn die Verzögerung im Bereich eines Samples liegt, so kann dieses Problem vernachlässigt werden [54][22].

20 http://ccrma.stanford.edu/~jos/pasp/First_Order_Allpass_Interpolation.html
21 http://ccrma.stanford.edu/~jos/pasp/Linear_Interpolation.html
22 http://ccrma.stanford.edu/~jos/pasp/First_Order_Allpass_Interpolation.html

Dynamikfilter

Wenn eine Saite stärker angeschlagen wird, so steigt auch das Spektralzentrum an und der Ton erklingt somit heller. Dies wird im erweiterten Karplus-Strong Modell, ähnlich wie beim Zupfrichtungsfilter, durch ein einpoliges Tiefpassfilter mit ungedämpftem Gleichstromanteil erreicht [55][23] [53, §2.4]

$$H_L(z) = \frac{1 - R_L}{1 - R_L z^{-1}}. \tag{2.57}$$

Dabei ist $R_L \in [0, 1)$ ein von der Grundfrequenz f_0 und dem gewünschten Dynamikpegel L abhängiger Parameter. Der Dynamikpegel wird als Bandbreite zwischen 0 und der halben Abtastrate, der *Nyquistfrequenz*, $\frac{f_s}{2}$ definiert und führt somit bei kleinen Werten zu einem dumpfen Ton und bei großen zu hellen Tönen.

Für eine feste Grundfrequenz kann R_L allein abhängig von der Bandbreite L wie folgt gewählt werden [52, §3.11.2]

$$R_L \triangleq e^{-\frac{\pi L}{f_s}}. \tag{2.58}$$

Damit diese Charakteristik allerdings für verschiedene Tonhöhen gleich ausfällt, muss $R_L(f_0)$ abhängig von der jeweiligen Grundfrequenz so gewählt werden, dass die Amplitudenantwort des Filters für jede Grundefrequenz gleich ist. Dies erreicht man mit folgender Wahl für $R_L(f_0)$ [52, §3.11.2]

$$R_L(f_0) = \frac{1 - G_L^2(f_m)\cos\left(\frac{2\pi f_0}{f_s}\right)}{1 - G_L^2(f_m)} \pm 2G_L(f_m)\sin\left(\frac{\pi f_0}{f_s}\right)\frac{\sqrt{1 - G_L^2(f_m)\cos^2\left(\frac{\pi f_0}{f_s}\right)}}{1 - G_L^2(f_m)}, \tag{2.59}$$

wobei der Wert verwendet wird, für den $R_L(f_0) < 1$ ist. Die Referenzamplitudenantwort

$$G_L(f_m) = \left|H_L\left(e^{i\frac{2\pi f_m}{f_s}}\right)\right| \tag{2.60}$$

ist für eine Referenzfrequenz

$$f_m = \sqrt{f_u f_l} \tag{2.61}$$

definiert, die sich als logarithmisches Mittel zwischen der Frequenz des tiefsten zu erwartenden Tones f_l und des höchsten zu erwartenden Tones $f_u < \frac{f_s}{2}$ ergibt.

2.4 Stereo Modellierung

Bei professionellen CD-Aufnahmen wird in der Regel darauf geachtet, dass eine *Monokompatibilität* [69] eingehalten wird, so dass auch ein akzeptabler Hörgenuss z.B. über einfache Küchenradios gegeben ist. Um dies zu gewährleisten, wird in der Regel zur Erstellung des Stereobilds eines Klanges einfaches *Panning*, d.h. unterschiedliche Lautstärke für linken und rechten Kanal, verwendet. Da bei Zeitverzögerungen zwischen linken und rechtem Kanal bei der Monowiedergabe unschöne Kammfiltereffekte entstehen können.

23 http://ccrma.stanford.edu/realsimple/faust_strings/Dynamic_Level_Lowpass_Filter.html

Diese Tatsache kann bei der Trennung ausgenutzt werden, indem das Panning z.b. wie in [35] über das *Prinzip der konstanten Leistung* (engl. constant power law) modelliert

$$s_{L,i}(t) = \cos(\theta_i) \cdot y_i(t) \tag{2.62}$$
$$s_{R,i}(t) = \sin(\theta_i) \cdot y_i(t), \tag{2.63}$$

wobei $\theta_i \in [0, \frac{\pi}{2}]$ der *Panningparameter* der i-ten Spur ist.

Die bisherigen Zustandsmodelle können nun durch einfaches Ersetzen der Messmatrix $\mathbf{C}$ zur Modellierung von Stereoaufnahmen erweitert werden

$$\mathbf{C}_{stereo,i} = \begin{pmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{pmatrix} \cdot \mathbf{C}_i. \tag{2.64}$$

KAPITEL 3

Klanganalyse

Für die Klanganalyse wurde in dieser Arbeit ein größtenteils auf [22] basierendes Verfahren verwendet, wie es in Abb. 3.1 dargestellt ist. Aus den daraus gewonnenen Daten sollen dann

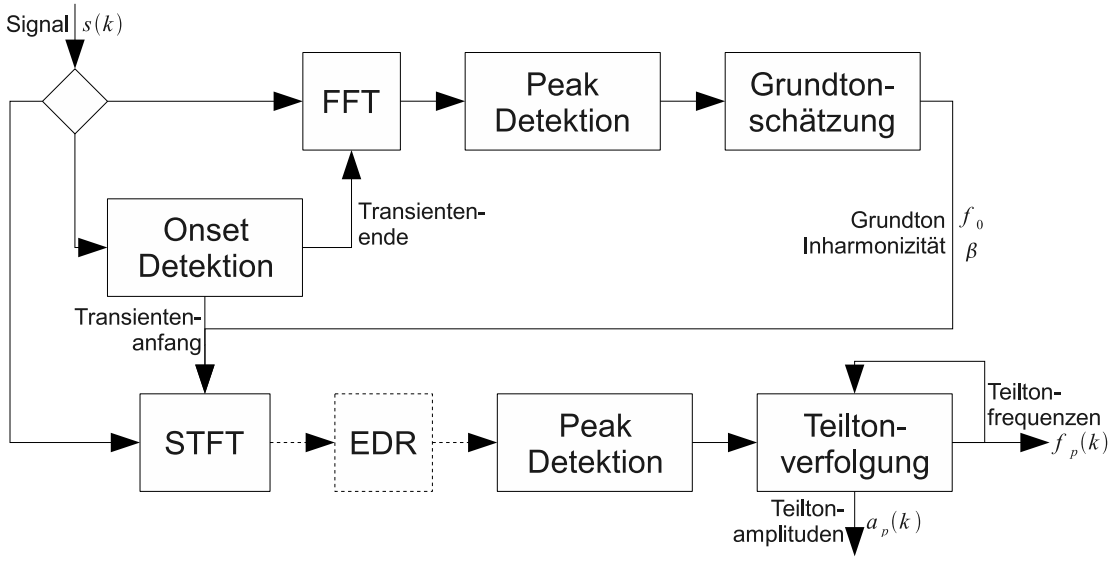

Abbildung 3.1: Strukturdiagramm des verwendeten Klanganalyseverfahrens.

die Parameter der Schleifenfilter eines erweiterten Karplus-Strong Modells und daraus wiederum die Aggregatanregung geschätzt werden. Dies soll im Detail in den folgenden Abschnitten vorgestellt und anhand einer Aufnahme einer Gitarrensaite veranschaulicht werden. Dazu wurde aus einem Soundfont einer Ovation Akkustikgitarre von Robert C. Allen [2] das Sample der offenen A-Saite extrahiert. Dabei handelt es sich um eine einkanalige Aufnahme mit der Abtastrate $f_s = 44100$ Hz.

3.1 Onset Detektion

Vor der eigentlichen Klanganalyse wurde zuerst der Onset bestimmt, also der Zeitpunkt, an dem der Ton anfängt zu erklingen. Dafür wurde das von Paul Masri [33, §5.2.2] beschriebene

Verfahren der *Einschwinghüllkurve* (engl. Attack Envelope) verwendet. Dabei handelt es sich um ein Zeitbereichsverfahren, das mittels eines *Peakfollowers* die Hüllkurve des Signals berechnet. Zusätzlich zum Onset kann mit dieser Methode auch das Ende des Transienten für angeschlagene bzw. gezupfte Instrumente bestimmt werden.

3.1.1 Schätzung des Transientenanfangs

Das Signal $s(k)$ wird in nicht überlappende Blöcke von L Samples zerlegt, innerhalb derer der Maximalwert des Betrags bestimmt wird

$$y(l) = \max_{i=0}^{L-1}(|s(l \cdot L + i)|). \tag{3.1}$$

Das Resultat ist die Sampleblocksequenz $y(l)$. Die Blockgröße wird in der Regel bei einer Abtastrate von 44100 Hz auf 50 Samples gesetzt, da dies einen guten Kompromiss zwischen Rechenzeit und Genauigkeit darstellt.

Die Einschwinghüllkurve wird nun als das Maximum zwischen dem aktuellen Sampleblockwert $y(l)$ und der letzten Peakfollowerausgabe $P(l-1)$ multipliziert mit einem Dämpfungsfaktor ρ_{DECAY} berechnet

$$P(l) = \max(y(l), P(l-1) \cdot \rho_{DECAY}). \tag{3.2}$$

In Abb. 3.2 ist dies veranschaulicht.

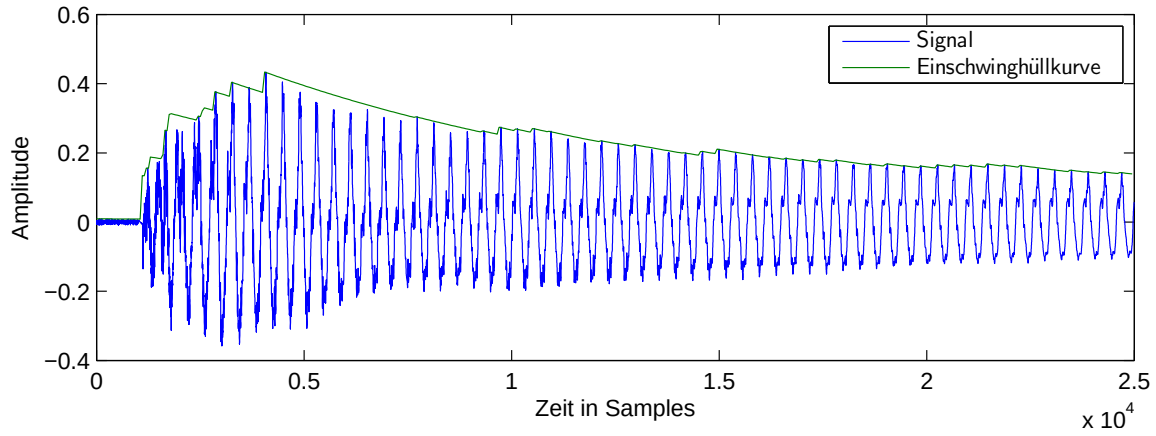

Abbildung 3.2: Einschwinghüllkurve für einen Gitarrensaite.

Zur Bestimmung des Onsets vergleicht man das Verhältnis zwischen der aktuellen und vorherigen Peakfollowerausgabe $P(l)$ und $P(l-1)$ mit einem gegebenen Schwellwert $T_{D,AE}$ (siehe Abb. 3.3 auf der nächsten Seite)

$$\frac{P(l)}{P(l-1)} > T_{D,AE}. \tag{3.3}$$

Um Fehldetektionen am Anfang eines Signales zu vermeiden, wird $P(0)$ mit dem Maximum aus einer Folge von aufeinanderfolgenden Blöcken am Anfang des Signals initialisiert. Laut [33, §5.2.2.1] ist eine Anzahl von 10 Blöcken hierfür eine gute Wahl.

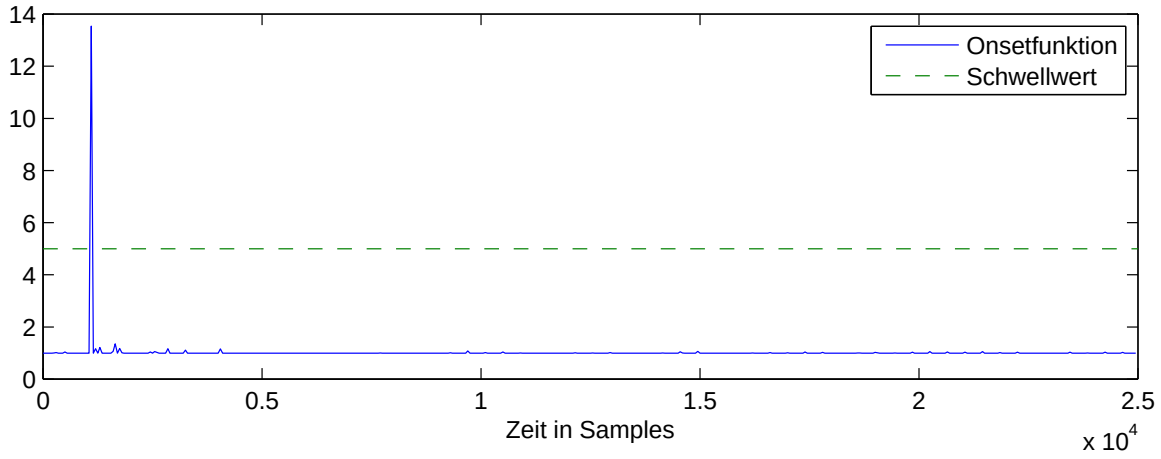

Abbildung 3.3: Onsetfunktion mit gewähltem Schwellwert.

3.1.2 Schätzung des Transientenendes

Für die Erkennung des Transientenendes wird der Zeitpunkt bestimmt, zu dem die Hüllkurve unter einen vordefinierten Bruchteil des Maximums nach dem Onset gefallen ist bzw. eine maximale Dauer überschritten wurde. Dafür wird das Maximum nach dem Onset als

$$P_{SUSTAIN} = \max(P_{SUSTAIN}, y(l)) \tag{3.4}$$

bestimmt, wobei $P_{SUSTAIN}$ mit dem Hüllkurvenwert $P(l_{ONSET})$ zum Zeitpunkt l_{ONSET} des Onsets initialisiert wird. w Das Ende des Transienten ergibt sich dann durch

$$\frac{P(l)}{P_{SUSTAIN}} < T_{S,AE} \quad \text{oder} \quad l \geq l_{ONSET} + T_{MAX}, \tag{3.5}$$

wobei $T_{S,AE}$ der Schwellwert ist, unter den das Verhältnis fallen muss, bzw. T_{MAX} die Maximaldauer für einen Transienten in Sampleblöcken, damit der Transient als beendet gilt.

3.2 Kurzzeit-Fourier-Transformation (STFT)

Da nun bekannt ist, an welcher Stelle im Signal der Klang tatsächlich anfängt, kann nun eine Zeitfrequenzanalyse durchgeführt werden, die mit dem Transientenanfang beginnt. Dazu wird in der Regel die Kurzzeit-Fourier-Transformation (engl. Short Time Fourier Transformation - STFT) verwendet. Dabei handelt es sich um eine Fouriertransformation, die auf, mit einer Fensterfunktion $w(k)$ multiplizierten, Blöcke des Signals $s(k)$ angewendet wird. Dies wird durch die folgende Gleichung ausgedrückt [56][1]

$$S(l,n) = \sum_{k=-N/2}^{N/2-1} s(l \cdot L_{STEP} + k) \cdot w(k) e^{-i\frac{2\pi n(l \cdot L_{STEP}+k)}{N}}. \tag{3.6}$$

1 http://ccrma.stanford.edu/~jos/sasp/Practical_Computation_STFT.html

Dabei ist n der Index des zu analysierenden Frequenzbins mit der Frequenz $f = \frac{f_s n}{N}$ im l-ten Block, mit der Abtastrate f_s und der Fenster- bzw. Blockgröße N. L_{STEP} gibt die Schrittweite an, um die das Fenster jeweils verschoben wird.

Das Betragsspektrum $|S(l,n)|$ über die Zeit wird auch Spektrogramm genannt und ist in Abb. 3.4 exemplarisch für die Aufnahme einer Gitarrensaite dargestellt.

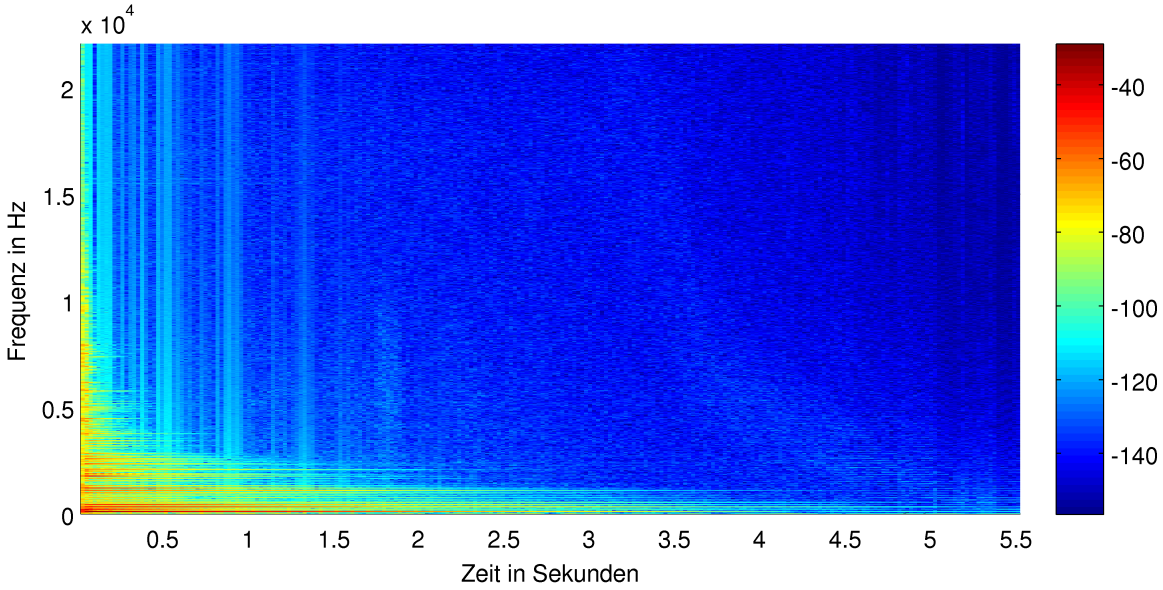

Abbildung 3.4: Spektrogramm einer Aufnahme einer Gitarrensaite.

3.2.1 Fensterfunktionen

Für die Fensterfunktion kann zum Beispiel ein Hamming-Fenster [22, §3.2.1] [63] [70]

$$w_{HAMMING}(k) = 0{,}54 + 0{,}46 \cdot \cos\left(\frac{2\pi k}{N-1}\right) \tag{3.7}$$

oder ein Gauss-Fenster

$$w_{GAUSS}(k) = e^{-\frac{k^2}{2\sigma^2}} \tag{3.8}$$

für

$$-\frac{N}{2} \leq k \leq \frac{N}{2} - 1. \tag{3.9}$$

verwendet werden, wie sie in Abb. 3.5 auf der nächsten Seite gezeigt sind.

3.3 Peak Detektion

Im nächsten Schritt wird im Betragsspektrum $|S(l,n)|$ nach mehreren Maxima gesucht. Dafür wird das Frequenzspektrum, nachdem ein Maximum gefunden wurde, unterhalb des Maximums, solange die Steigung positiv ist, und oberhalb des Maximums, solange die Steigung negativ ist,

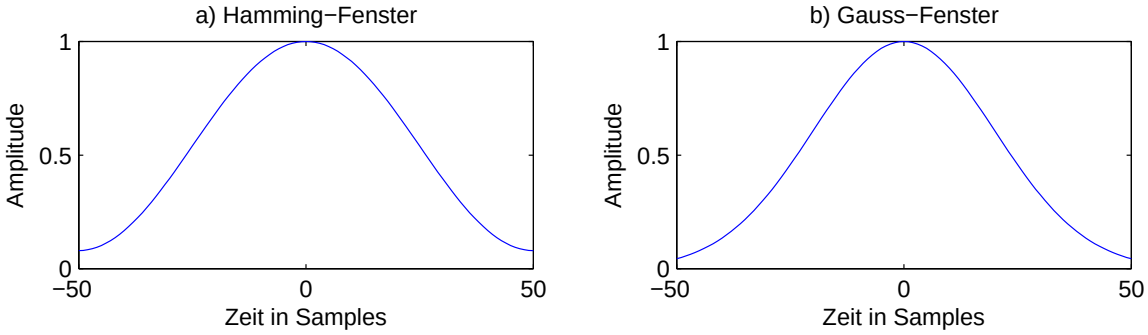

Abbildung 3.5: Fensterfunktionen für $N = 100$: a) Hamming-Fenster und b) Gauss-Fenster mit $\sigma = \frac{N}{5}$.

auf Null gesetzt [22, §3.2.1]. Wenn die maximale Anzahl an Peaks gefunden wurde bzw. die Amplitude eines Peaks unter ein vorgegebenes Verhältnis zur maximalen Amplitude fällt, so endet die Suche nach Maxima.

3.3.1 Peak Interpolation

Die Frequenzauflösung so gefundener Maxima ist abhängig von der gewählten Blockgröße der STFT. Um eine genauere Schätzung der zugehörigen Frequenzen und Amplituden zu erhalten, kann eine Interpolation verwendet werden. Hierfür eignet sich beispielsweise die *quadratische* bzw. *parabolische Interpolation* [56][2]. Diese Interpolation gilt als exakt, wenn ein Gaussfenster verwendet wird, da die Transformationsamplitude eines solchen auf einer Dezibelskala eine Parabel ist. Die Tatsache, dass in der Praxis das Gaussfenster abgeschnitten wird, führt zu keiner großen Abweichung von einer Parabel.

Wenn nun ein Maximum im n_{max}-ten Frequenzbin gefunden wurde, so werden für die Interpolation zusätzlich die beiden umliegenden logarithmierten Amplituden mitverwendet

$$\alpha = \log(|S(l, n_{max} - 1)|), \quad \beta = \log(|S(l, n_{max})|), \quad \gamma = \log(|S(l, n_{max} + 1)|). \tag{3.10}$$

Die interpolierte Position des Maximums ergibt sich dann aus dem Korrekturterm

$$\delta = \frac{1}{2} \frac{\alpha - \gamma}{\alpha - 2\beta + \gamma}. \tag{3.11}$$

Dieser Korrekturterm liegt im Bereich $[-\frac{1}{2}, \frac{1}{2}]$.

Aus der daraus resultierenden absoluten Position $n_{max} + \delta$ folgt die interpolierte Frequenzschätzung

$$f_{max} = \frac{f_s(n_{max} + \delta)}{N} \tag{3.12}$$

mit der zugehörigen logarithmierten Amplitude

$$\log(a_{max}) = \beta - \frac{1}{4}(\alpha - \gamma)\delta \tag{3.13}$$

2 http://ccrma.stanford.edu/~jos/sasp/Quadratic_Interpolation_Spectral_Peaks.html

bzw. der Amplitude

$$a_{max} = e^{\beta - \frac{1}{4}(\alpha - \gamma)\delta}. \tag{3.14}$$

Dieses Interpolationsverfahren ist in Abb. 3.6 veranschaulicht.

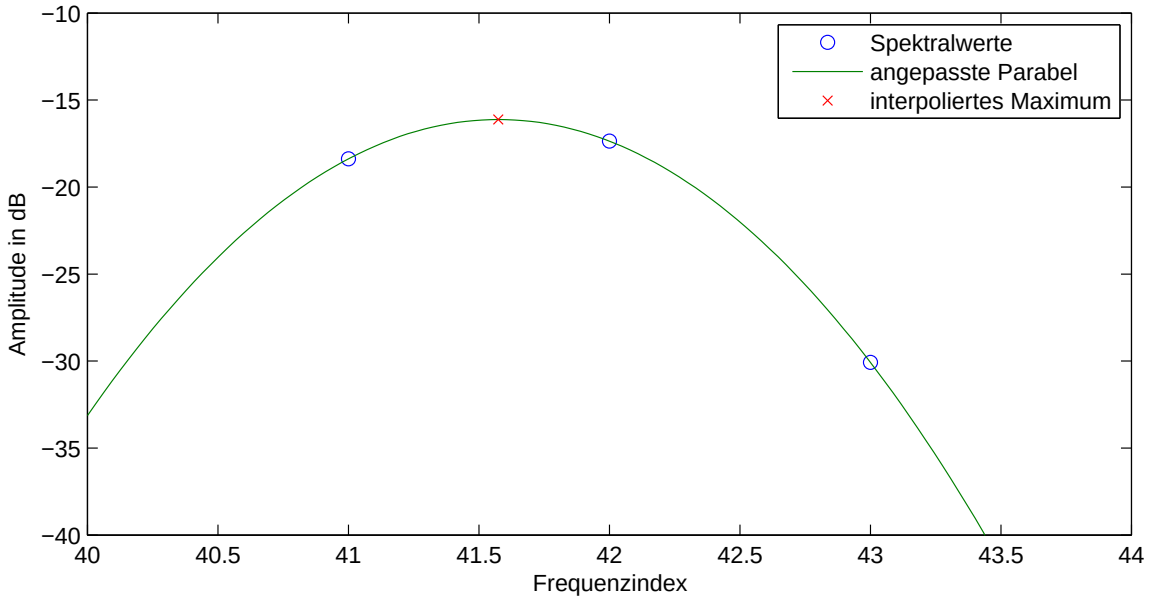

Abbildung 3.6: Parabolische Interpolation eines spektralen Maximums mittels dreier Stützstellen aus dem Spektrum.

3.3.2 Spektralmaskierung

Um zu vermeiden, dass falsche Teiltöne erkannt werden, wird eine Maskierungsmethode verwendet, welche den *Maskierungseffekt* [67] im menschlichen Gehör nachahmt [22, §3.2.2], bei dem schwache Frequenzspitzen neben lauten Frequenzspitzen maskiert und somit nicht wahrgenommen werden. Dazu wird über das Amplitudenspektrum ein Schwellwertfunktion berechnet, die zu schwache Teiltöne neben lauten Teiltönen ausblenden soll. Diese Funktion $w_{mask}(n)$ ergibt sich als 0,9 mal das Maximum innerhalb eines Blocks der Größe $2 \cdot w_{sz}$

$$w_{mask}(n) = 0{,}9 \cdot \max_{i=-w_{sz}}^{w_{sz}} (|S(l, n+i)|). \tag{3.15}$$

Es ist sinnvoll w_{sz} abhängig von der zu erwartenden Grundfrequenz zu wählen. Vor allem sollte w_{sz} kleiner als der Frequenzbandindex n_0 der Grundfrequenz $\tilde{f}_0$ sein, da ansonsten essentielle Peaks verloren gehen.

Hier wurde für eine erste Grundfrequenzhypothese eine Peak Detektion für einige wenige Maxima (z.B. 10) ohne Spektralmaskierung durchgeführt, gefolgt von der Grundtonschätzung wie sie im nächsten Abschnitt beschriebenen wird. Schließlich wurde erneut eine Peak Detektion

unter Verwendung der Spektralmaskierung mit $w_{sz} = \lfloor \frac{n_0}{3} \rfloor$ und den sonst gewünschten Parametern durchgeführt. Das Ergebnis der so gefundenen Maxima zusammen mit der verwendeten Spektralmaske ist in Abb. 3.7 veranschaulicht.

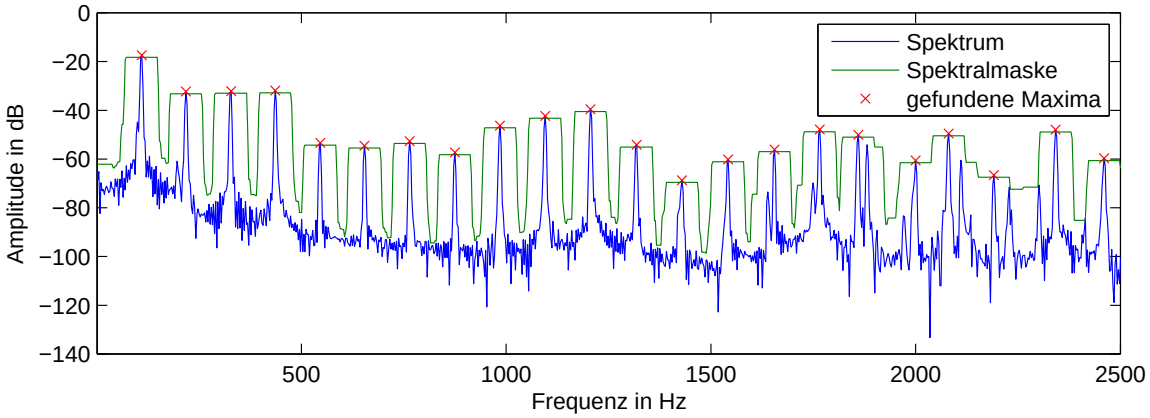

Abbildung 3.7: Spektralmaskierung des Spektrums in Anlehnung an den Maskierungseffekt beim menschlichen Gehör.

3.4 Grundtonschätzung

Nun werden alle Peaks über einem gegebenen Schwellwert herangezogen, um den Grundton des Klanges zu schätzen. Dafür werden zuerst die Differenzen aufeinanderfolgender Frequenzpeaks berechnet [22, §3.3.1]

$$\begin{aligned} \Delta f_1 &= f_1 - 0 \\ \Delta f_2 &= f_2 - f_1 \\ &\dots \\ \Delta f_i &= f_i - f_{i-1} \\ &\dots \\ \Delta f_N &= f_N - f_{N-1} \\ \Delta f &= \{\Delta f_1, \Delta f_2, \dots, \Delta f_N\}. \end{aligned} \tag{3.16}$$

Im Gegensatz zu dem in [22, §3.3.1] präsentierten Verfahren wird nun der Median aus der so gewonnenen Menge von Frequenzdifferenzen als Schätzung für die Grundfrequenz verwendet

$$\tilde{f}_0 = \text{median}(\Delta f). \tag{3.17}$$

Um etwas bessere Amplitudenschätzungen zu erhalten, wird auf Basis der in (3.17) gefundenen Grundfrequenz erneut eine Fourier Transformation durchgeführt, deren Fenstergröße $N' = 4 \cdot \frac{f_s}{\tilde{f}_0}$ diesmal auf das vierfache der Periodenlänge der geschätzten Grundfrequenz $\tilde{f}_0$ gesetzt wird, und die Grundtonschätzung wiederholt.

3.4.1 Differenz der Frequenzdifferenzen

Eine Grundtonschätzung, welche die Inharmonizität berücksichtigt, erhält man, wenn man die Differenzen aufeinanderfolgender Frequenzdifferenzen

$$\Delta\Delta f_i = \Delta f_i - \Delta f_{i-1} \tag{3.18}$$

gemittelt über L aufeinanderfolgende Obertöne von den Frequenzdifferenzen Δf subtrahiert [22, §3.3.1]

$$\Delta f_i' = \Delta f_i - \frac{1}{L}\sum_{j=1}^{L} \Delta\Delta f_{i-j}. \tag{3.19}$$

3.4.2 Nichtlineare Regression mittels Inharmonizitätskurve

Eine weitere Verbessung der Grundtonschätzung erhält man, wenn man auf die gemessenen Frequenzen $\hat{f}_p$ der Teiltöne eine nichtlineare Regression unter Verwedung der Inharmonizitätskurve aus (2.4) bzw. (2.5) anwendet [22, §3.3.3]. Dazu wird die Methode der kleinsten Quadrate verwendet

$$\min_{f_0,\beta} \sum_{p=1}^{P} (\hat{f}_p - f_p(f_0,\beta))^2. \tag{3.20}$$

Dabei ist $f_p(f_0,\beta)$ die zu erwartende Frequenz des p-ten Teiltons entsprechend (2.4) bzw. (2.5) abhängig vom Grundton f_0 und der Inharmonizitätskoeffizienten β. Frequenzen von Teiltönen, die nicht in ermittelt werden konnten, werden in der Berechnung von (3.20) einfach ausgelassen.

Um den Fehler in den wichtigen tieffrequenten Teiltönen zu reduzieren, kann die Differenz in (3.20) jeweils mit $\frac{1}{p}$ gewichtet werden

$$\min_{f_0,\beta} \sum_{p=1}^{P} \left(\frac{f_p(f_0,\beta) - \hat{f}_p}{p} \right)^2. \tag{3.21}$$

Hierdurch wird ein größerer Fehler für hochfrequente Teiltöne toleriert, da die Optimierung nicht mehr für die tatsächlichen Frequenzabweichungen durchgeführt wird, sondern für die Frequenzabweichungen der lokalen Grundfrequenzen der Teiltöne. Dies wird ersichtlich, wenn für $f_p(f_0,\beta)$ z.B. die Definition aus (2.5) eingesetzt und $\hat{f}_p$ entsprechend (2.3) durch

$$\hat{f}_p = p \cdot \hat{f}_{p,0} \tag{3.22}$$

substituiert wird, wobei $\hat{f}_{p,0}$ die, auf Basis der Messung $\hat{f}_p$ geschätzte, lokale Grundfrequenz für den p-ten Teilton ist. Nachdem p gekürzt wird, ergibt sich somit

$$\min_{f_0,\beta} \sum_{p=1}^{P} \left(f_0\sqrt{1+p^2\beta} - \hat{f}_{p,0} \right)^2. \tag{3.23}$$

Auf die, für den Gitarrenklang, direkt nach dem Transienten gefundenen Teiltonfrequenzen wurde diese nichtlineare Regression durchgeführt. Das Ergebnis ist in Abb. 3.8 auf der folgenden Seite abgebildet. Zum Vergleich ist zudem die harmonische Frequenz für die entsprechenden Teiltonindizes dargestellt.

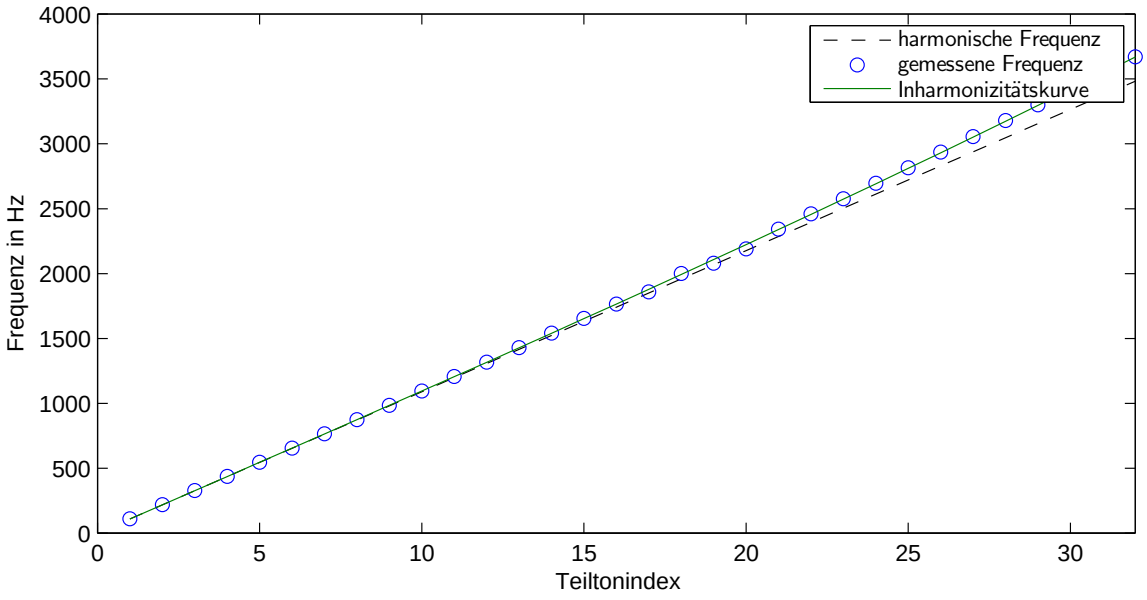

Abbildung 3.8: Inharmonizitätskurve für die gemessenen Teiltonfrequenzen mit der harmonischen Frequenz als Vergleich. Der zugehörige Inharmonizitätskoeffient und die Grundfrequenz sind $\beta = 1{,}077 \cdot 10^{-4}$ respektive $f_0 = 108{,}8434$ Hz.

3.5 Teiltonverfolgung

Für das Verfolgen der Amplituden und Frequenzen der einzelnen Teiltöne, wurde ein relativ einfaches Verfahren, ähnlich dem aus [58, §6], verwendet, so wie es bereits in Abb. 3.1 auf Seite 30 gezeigt wurde. Dazu wurde zuerst eine Peak Detektion mit Grundtonschätzung auf dem Segment vorgenommen, das auf den Transienten folgt. Dafür wurde die ermittelte Position aus (3.5) verwendet. Auf die so geschätzten Frequenzen der Teiltöne wurde die Kurve aus (2.4) angepasst, um die Parameter f_0 und β zu ermitteln.

Mit Hilfe dieser Parameter können "Leitfrequenzen" ermittelt werden, die dazu dienen sollen, Peaks in den einzelnen gemessenen Spektren miteinander über die Zeit hinweg zu verbinden. Dazu wird jeweils im Bereich um eine Leitfrequenz g_p nach einem passenden Peak gesucht. Dieser Bereich wird durch die Abweichung

$$d_p = \delta(g_p - g_{p-1}) \tag{3.24}$$

von der Leitfrequenz definiert, dabei ist $\delta \in (0, 0{,}5]$ der erlaubte Abweichungsfaktor. Sollten mehrere Peaks in diesen Bereich fallen, so wird der verwendet, welcher der Leitfrequenz am nächsten ist. Sollte kein bereits gefundener Peak in diesem Bereich gefunden werden, so wird nach dem, der Leitfrequenz in diesem Bereich am nächsten liegendem, Maximum gesucht. Sollte auch dies scheitern, so wird der zur Leitfrequenz gehörende Amplitudenwert verwendet.

Die Leitfrequenzen werden nach der Verarbeitung eines Spektrums mit den neu ermittelten Frequenzen f_p aktualisiert

$$g'_p = \alpha \cdot f_p + (1 - \alpha) \cdot g_p. \tag{3.25}$$

Die auf diese Weise für den Beispielklang ermittelten Frequenzverläufe der ersten 20 Teiltöne und die Amplitudenverläufe, der Übersichtlichkeit halber nur für die ersten 10 Teiltöne, sind in Abb. 3.9 respektive Abb. 3.10 auf der nächsten Seite veranschaulicht.

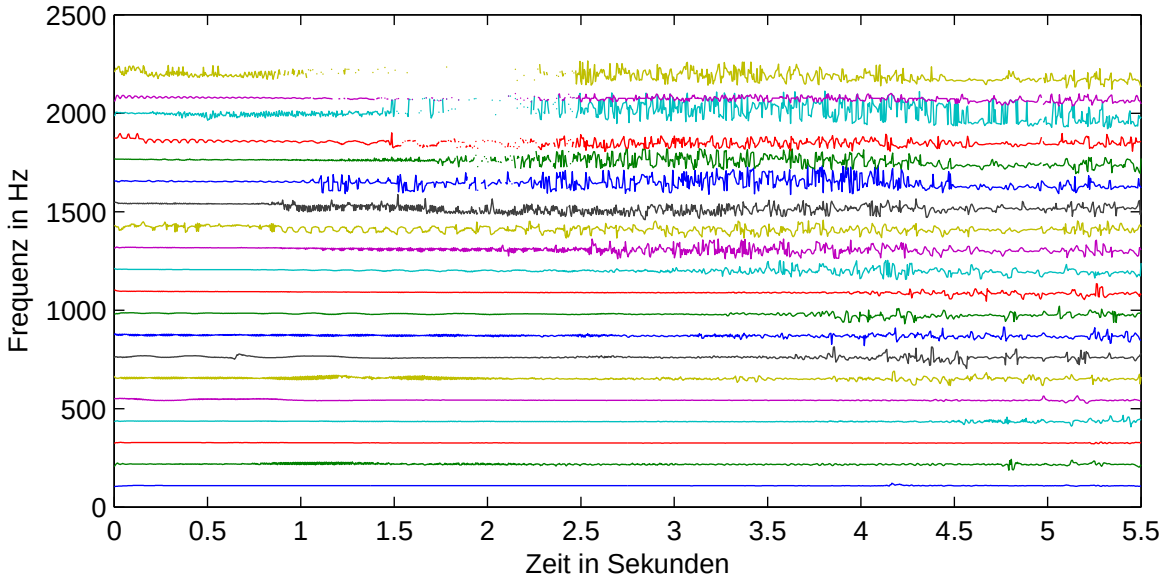

Abbildung 3.9: Frequenzverläufe der ersten 20 Teiltöne, resultierend aus der Teiltonverfolgung basierend auf dem Spektrogramm. Man erkennt, dass für die höheren Teiltöne um 2 Sekunden keine Messdaten vorliegen, da hier die Amplituden der Teiltöne unterhalb des Schwellwerts liegen. Außerdem werden die Messungen mit zunehmender Zeit verrauschter, da sich die Amplituden immer mehr der Rauschgrenze nähern.

Eine Alternative zu dem hier vorgestellten, etwas simplen Verfahren, könnte z.B. die Verwendung eines *Joint Probabilistic Data Association Filters* (JPDAF) [5, §9.3] sein, mit Hilfe dem es möglich ist, die Werte der Leitfrequenzen durch die einzelnen gefundenen Peaks probabilistisch zu aktualisieren oder damit direkt die Amplitudenwerte der Hüllkurven zu schätzen. Diese Alternative wurde hier allerdings aufgrund der damit verbunden Komplexität nicht in Betracht gezogen, da dies auch nicht das Hauptaugenmerk dieser Arbeit sein soll.

3.5.1 Energy Decay Relief (EDR)

Eine wesentlich robustere Schätzung der Hüllkurven der einzelnen Teiltöne erhält man, wenn man das verwendete Spektrogramm durch das *Energy Decay Relief* (EDR) ersetzt. Dabei handelt es sich um eine Zeit-Frequenzverteilung, die für ein Frequenzband n zu einem Zeitpunkt k die Gesamtsignalenergie angibt, die noch im System vorhanden ist

$$\mathrm{EDR}(k,n) = \sum_{l=k}^{L} |S(l,n)|^2. \tag{3.26}$$

Das EDR für den hier untersuchten Gitarrenklang ist in Abb. 3.11 auf Seite 41 abgebildet.

Eine Teiltonverfolgung basierend auf dem EDR statt des Spektrogramms ist in Abb. 3.12 auf Seite 42 und Abb. 3.13 auf Seite 42 durch die Frequenz- bzw. Amplitudenverläufe der Teiltöne,

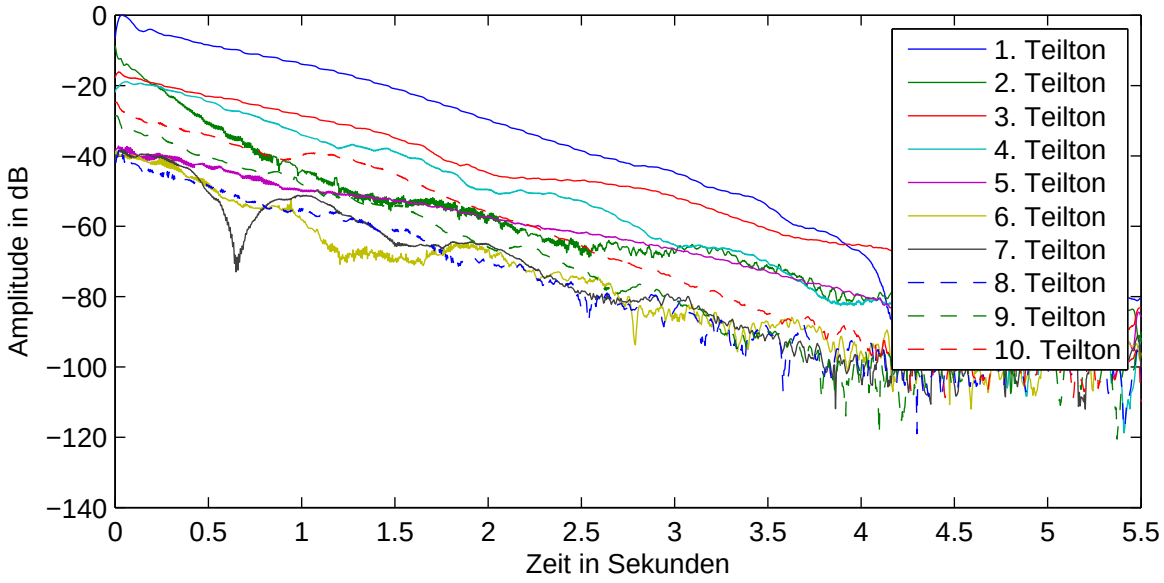

Abbildung 3.10: Amplitudenverläufe der ersten 10 Teiltöne, resultierend aus der Teiltonverfolgung basierend auf dem Spektrogramm. Man sieht, dass die Amplituden der Teiltöne auf der logarithmischen Skala etwa linear abfallen, mit Ausnahme der Teiltöne, die aufgrund von Schwebungen leicht pulsiernde Amplituden besitzen.

die schon in Abb. 3.9 auf der vorigen Seite bzw. Abb. 3.10 zur Veranschaulichung gewählt wurden.

Es ist sofort ersichtlich, dass die Frequenz- und Amplitudenverläufe, die unter Verwendung des EDR ermittelt wurden, weniger verauscht und somit robustere Schätzungen sind, als die auf Basis des Spektrogramms ermittelten Verläufe.

3.6 Schleifenfilter Identifikation

Im Folgenden sollen nun die Parameter für die Schleifenfilter des erweiterten Karplus-Strong Modells aus Abschnitt 2.3.3 identifiziert werden.

3.6.1 Saitendämpfungsfilter Identifikation

Mit Hilfe der Hüllkurven der einzelnen Teiltöne kann nun ein passendes Saitendämpfungsfilter entwickelt werden.

Schätzung der Dämpfungsfaktoren

Zuerst müssen die frequenzabhängigen Dämpfungsfaktoren geschätzt werden. Da die Hüllkurve der Teiltöne einer angeschlagenen Saite exponentiell abklingt, kann diese auf eine Dezibelskala durch eine Gerade approximiert werden [54][3]. Dazu wird eine lineare Regression auf dem Teil

3 http://ccrma.stanford.edu/~jos/pasp/General_Loop_Filter_Design.html

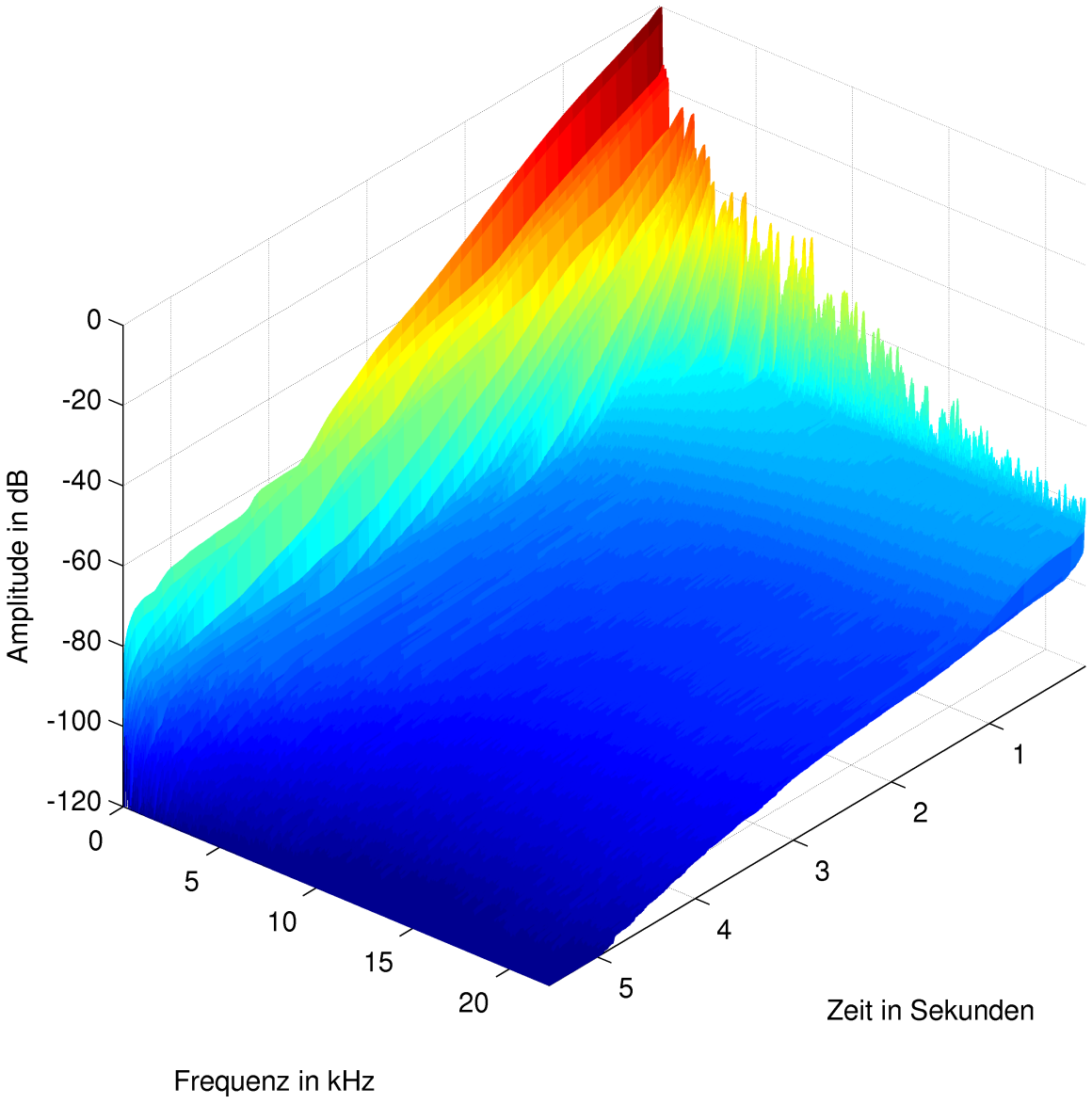

Abbildung 3.11: Energy Decay Relief (EDR) einer Aufnahme einer Gitarrensaite.

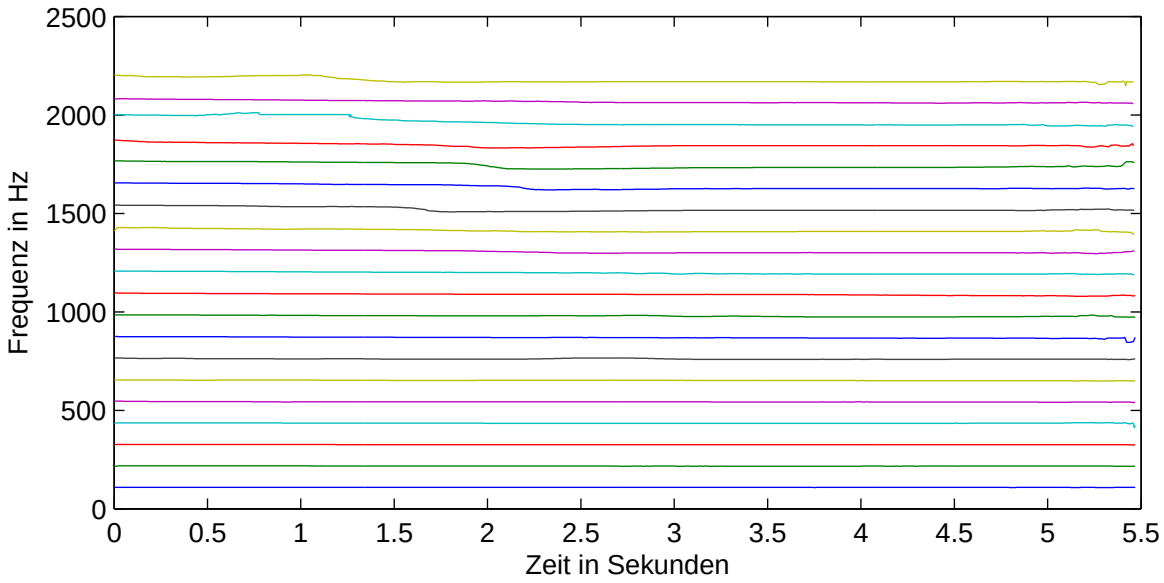

Abbildung 3.12: Frequenzverläufe der ersten 20 Teiltöne, resultierend aus der Teiltonverfolgung basierend auf dem Energy Decay Relief.

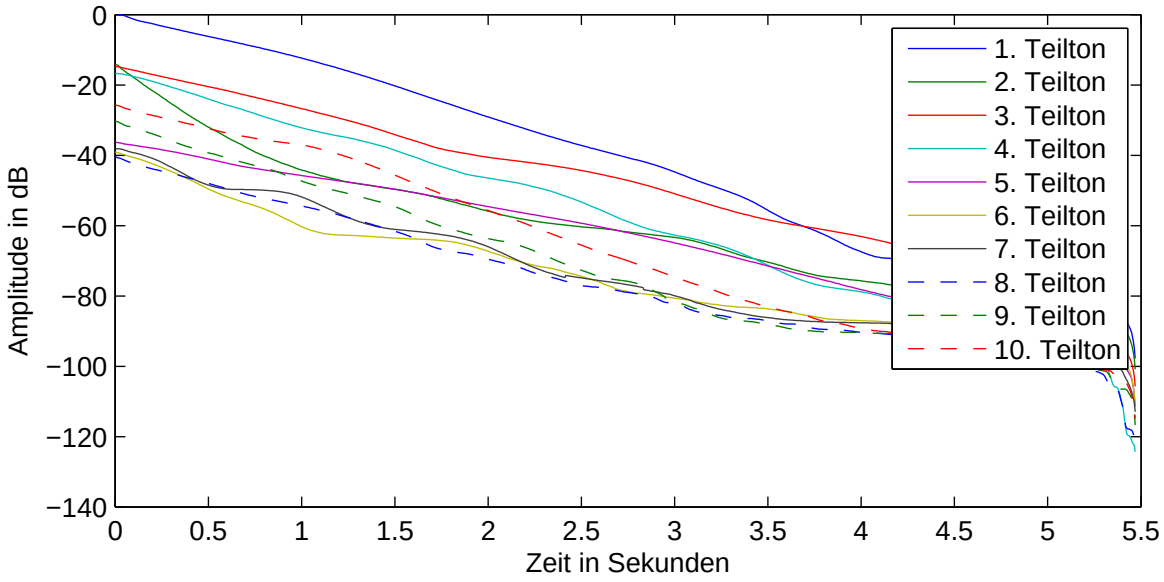

Abbildung 3.13: Amplitudenverläufe der ersten 10 Teiltöne, resultierend aus der Teiltonverfolgung basierend auf dem Energy Decay Relief.

der Hüllkurve der einzelnen Teiltöne verwendet, der direkt nach dem Ende des Transienten anfängt und, bis die Amplitude um etwa -20 dB abgefallen ist, andauert. Dies ist etwa der Punkt, an dem die zweite Abklingphase der Saite anfängt. Die Gleichung für die anzupassende Gerade mit der Steigung β_p und dem Aufpunkt α_p lautet

$$y(x) \triangleq \beta_p x + \alpha_p. \tag{3.27}$$

Die resultierende Approximation ist in Abb. 3.14 exemplarisch für die Hüllkurve des ersten Teiltons dargestellt.

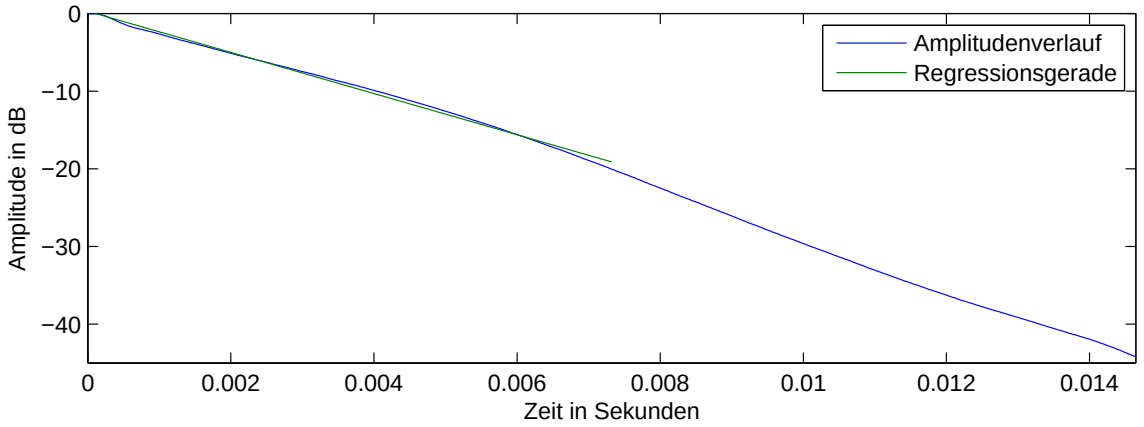

Abbildung 3.14: Lineare Regression zur Approximation der Hüllkurve des ersten Teiltons durch eine Gerade.

Die Steigung β_p der approximierten Geraden kann dann verwendet werden, um den Dämpfungsfaktor g_p für den p-ten Teilton zu schätzen. Dieser ergibt sich für die STFT-Schrittweite L_{STEP} und die Periodenlänge $\lambda = \frac{f_s}{f_0}$ der Grundfrequenz f_0 in Samples bei einer Abtastrate f_s zu [25, §3.2]

$$g_p = 10^{\frac{\beta_p \lambda}{20 L_{STEP}}}. \tag{3.28}$$

Diese Dämpfungsfaktoren bestimmen das Amplitudenspektrum des zu ermittelnden Saitendämpfungsfilters $H_d(z)$ an den Frequenzen f_p der zugehörigen Teiltöne und sind für das Beispiel der Gitarrensaite in Abb. 3.15 auf der folgenden Seite abgebildet. Es wurden hier nur die Messungen für die ersten 20 Teiltöne herangezogen, da die übrigen Messungen sehr fehleranfällig waren. Dies genügt auch für die Schätzung, da zum einen, wie aus Abb. 3.4 auf Seite 33 ersichtlich ist, höhere Frequenzen relativ schnell gedämpft werden und zum anderen der im Folgenden gewählte Filtertyp ohnehin nicht genügend Freiheitsgrade hat, um Details im hochfrequenten Bereich der Amplitudenantwort zu modellieren.

Schätzung der Filterparameter

Als Filtertyp für das Saitendämpfungsfilter soll hier ein symmetrischer FIR-Filter zweiter Ordnung mit der Parametrisierung aus (2.36) verwendet werden, wie es in (2.32) und (2.33) definiert wurde. Die Parameter B und S für dieses Filter werden durch die Methode der kleinsten

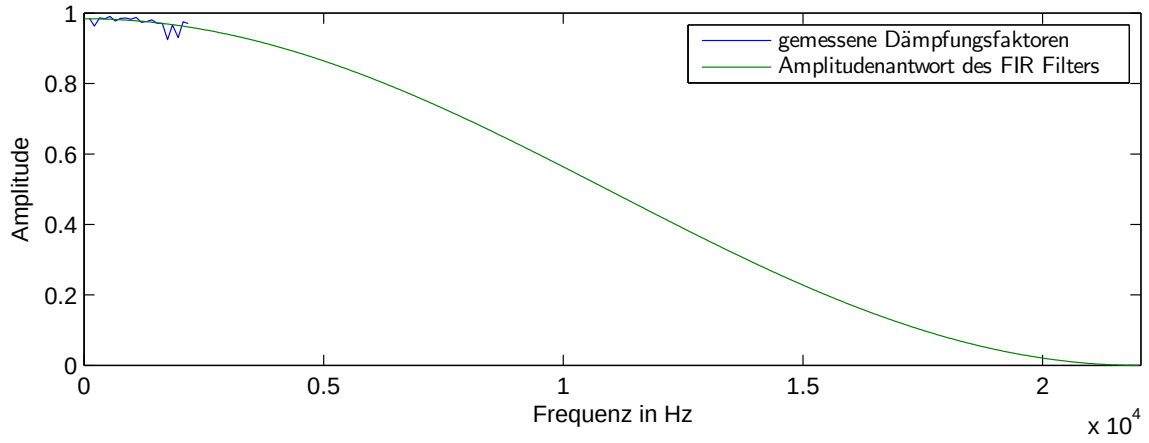

Abbildung 3.15: Gemessene Dämpfungsfaktoren für die ersten 20 Teiltöne und Amplitudenantwort des geschätzten Saitendämpfungsfilters für die Klanghelligkeit $B = 0$ und Haltedauer $S = 3{,}9716$ s.

Quadrate geschätzt, indem folgende Summe minimiert wird [60, §2.1]

$$\min_{S,B} \sum_{p=1}^{P} w_p (|H_d(e^{i\frac{2\pi f_p}{f_s}})| - g_p)^2. \tag{3.29}$$

Der Gewichtungsfaktor w_p lässt sich aus dem Fehler, der sich durch die lineare Regression ergibt, wie folgt berechnen

$$w_p = \frac{1}{N_p} \sum_{k=0}^{N_p-1} (A_p(k) - (\beta_p k + \alpha_p))^2, \tag{3.30}$$

wobei N_p die Länge des untersuchten Abschnitts der logarithmierten Hüllkurve $A_p(k)$ ist und α_p und β_p die Parameter der linearen Regression sind.

Die Amplitudenantwort des so geschätzten Saitendämpfungsfilters ist in Abb. 3.15 gezeigt. Hierfür wurde der Mittelwert der gemessenen Teiltonfrequenzen über die Zeit verwendet, um darauf eine nichtlineare Regression der Inharmonitätskurve durchzuführen.

3.6.2 Saitensteifheitsfilter Identifikation

Der Einfachheit halber soll das Saitensteifheitsfilter hier anhand der aus (3.21) erhaltenen Grundfrequenz f_0 und Inharmonizität β nach dem in [44] beschriebenen Verfahren ermittelt werden.

Das Saitensteifheitsfilter hat bei diesem Verfahren, wie schon in Abschnitt 2.3.3 erwähnt, die Struktur einer Kaskade von M Allpassfiltern erster Ordnung, dessen Übertragungsfunktion in (2.46) und deren Parametrisierung in (2.47), (2.42)-(2.44) und (2.48) angegeben ist.

Die Phasenlaufzeit des resultierenden Saitensteifheitsfilters ist zusammen mit der optimalen Phasenlaufzeit für die gegebene Grundfrequenz f_0 und Inharmonizität β, definiert durch [45,

§II.A]

$$P_p = \frac{f_s}{f_0\sqrt{1+\beta p^2}} \tag{3.31}$$

in Abb. 3.16 abgebildet. Die relativ großen Abweichungen der Phasenlaufzeit im hochfrequenten Bereich schlagen nicht so sehr ins Gewicht, da wie auch schon für das Saitendämpfungsfilter bemerkt wurde, der hochfrequente Anteil relativ schnell gedämpft wird, wodurch sich die dort gemachten Fehler nicht so sehr auf das Resultat auswirken.

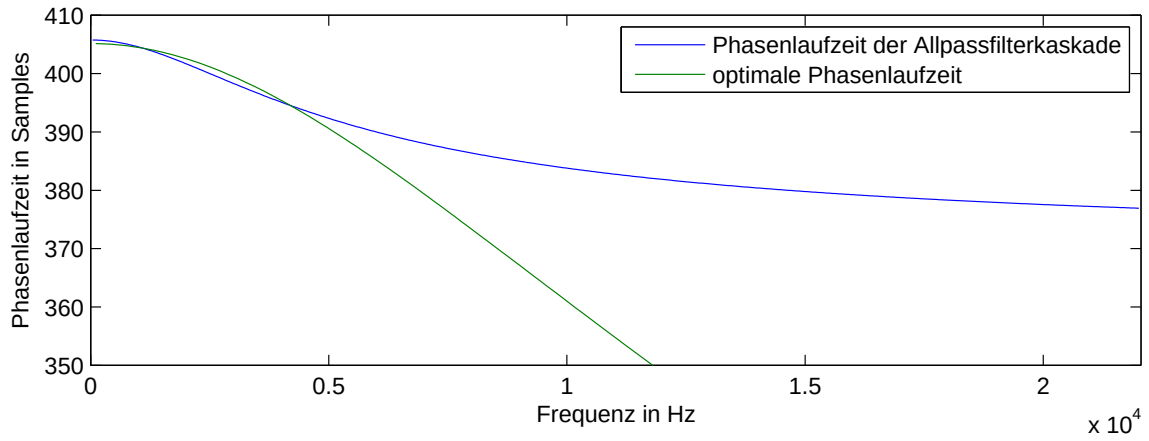

Abbildung 3.16: Phasenlaufzeit des Saitensteifheitsfilters unter Verwendung einer Kaskade von $M = 8$ Thiran Allpassfiltern erster Ordnung und optimale Phasenlaufzeit für $f_0 = 108{,}8495$ Hz und $\beta = 3{,}874 \cdot 10^{-5}$.

3.6.3 Saitenstimmungsfilter Identifikation

Aus der Phasenlaufzeit (2.34) des Saitendämpfungsfilters und der Phasenlaufzeit (2.49) bzw. (2.51) des Saitensteifheitsfilters für die Grundfrequenz kann nun der Parameter für das Saitenstimmungsfilter aus (2.53) mittels (2.54) und (2.55) berechnet werden.

Die Länge L der Verzögerungskette folgt dann direkt aus (2.56).

3.7 Rekonstruktion der Aggregatanregung

Nun, da die Verzögerungskettenlänge L, das Saitendämpfungsfilter $H_d(z)$, das Saitensteifheitsfilter $H_s(z)$ und das Saitenstimmungsfilter $H_\eta(z)$ bekannt sind, kann unter Verwendung der kommutierten Synthese aus Abschnitt 2.3.1, die Aggregatanregung rekonstruiert werden.

Eine einfache Möglichkeit, dies zu erreichen, ist die *inverse Filterung* [25, §3.1] [29, §2]. Dazu wird das aufgenommene Signal mit einem inversen Filter $S^{-1}(z)$ gefiltert. Dieses ergibt sich als Inverse der Übertragungsfunktion des Saitenmodells $S(z)$

$$S^{-1}(z) = \frac{1}{S(z)} = 1 - z^{-L}H_{loop}(z), \tag{3.32}$$

wobei

$$H_{loop}(z) = H_d(z)H_s(z)H_\eta(z). \tag{3.33}$$

Die resultierende Aggregatanregung ist in Abb. 3.17 zusammen mit dem Originalsignal abgebildet. Man kann leicht sehen, dass die Aggregatanregung wesentlich kürzer als der eigentliche Klang ist. Für die praktische Synthese wird deshalb die Aggregatanregung gekürzt und als *Wavetable* abgespeichert. Alternativ kann diese auch durch ein ARMA Modell modelliert werden.

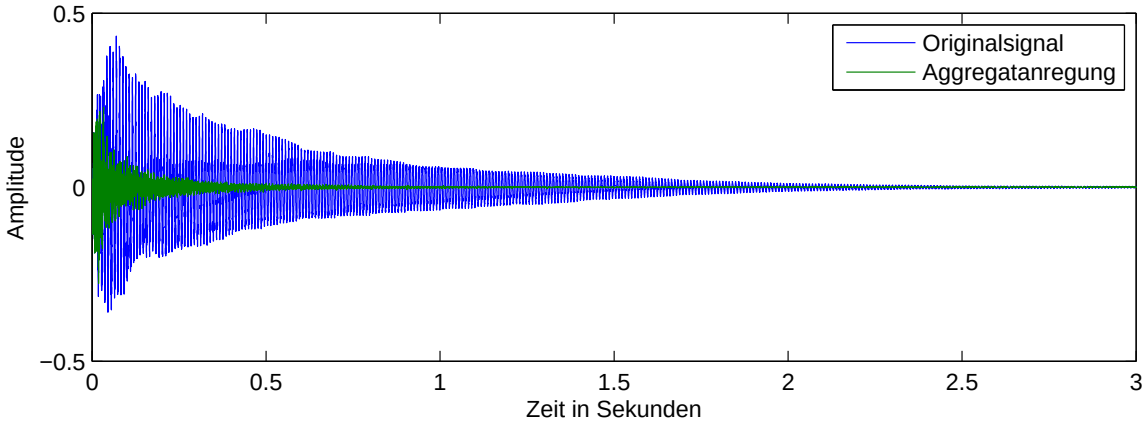

Abbildung 3.17: Geschätzte Aggregatanregung zusammen mit dem Originalsignal.

Statt der hier beschriebenen inversen Filterung wurde in [60, §2.2] sinusoidale Modellierung zur Rekonstruktion verwendet, dadurch lassen sich auch Frequenzvariationen im Originalsignal modellieren, was bei der inversen Filterung nicht möglich ist, da hier Zeitinvarianz angenommen wird.

Extraktion der Korpusmoden

Um die Aggregatanregung noch weiter zu kürzen, kann diese noch weiter aufgeteilt werden. Dazu werden z.B. die *Korpusmoden*, also die Resonanzen des Korpus', extrahiert, die am wenigsten gedämpft sind, da diese die Aggregatanregung unnötig lang machen [54][4] [60]. Darauf wird hier allerdings nicht weiter eingegangen.

4 http://ccrma.stanford.edu/~jos/pasp/Mode_Extraction_Techniques.html

KAPITEL 4

Algorithmen

Im folgenden Kapitel sollen Algorithmen betrachtet werden, die zur modellbasierten Trennung von Signalen herangezogen werden.

4.1 Kalman Filter

Handelt es sich bei einem Modell um ein lineares System mit gauss'scher Wahrscheinlichkeitsdichte, so ist der *Kalman Filter* ein optimaler rekursiver Zustandsschätzer dafür [6, §4.2]. Er erfreut sich vor allem auch auf dem Gebiet des *Trackings* und der Positionsbestimmung verschiedenster Objekte an Beliebtheit [65].

Dem Kalman Filter unterliegt ein Dynamik- und Messmodell ähnlich dem aus (2.11) und (2.12)

$$\underline{x}(k) = \mathbf{A}(k)\underline{x}(k-1) + \underline{v}(k), \quad k \geq k_0 \tag{4.1}$$

$$\underline{y}(k) = \mathbf{C}(k)\underline{x}(k) + \underline{w}(k), \quad k \geq k_0, \tag{4.2}$$

wobei $\underline{v}(k)$ und $\underline{w}(k)$ jeweils additive mittelwertfreie gauß'sche Rauschprozesse mit den Kovarianzmatrizen $\mathbf{Q}(k) = \mathrm{Cov}(\underline{v}(k))$ und $\mathbf{R}(k) = \mathrm{Cov}(\underline{w}(k))$ sind und $\mathbf{A}(k)$ und $\mathbf{C}(k)$ respektive die *Zustandsübergangs-* und *Messmatrizen* sind.

Die Zustandsschätzung selbst erfolgt dann rekursiv über eine *Prädiktor-Korrektor-Struktur*. Es wird also zuerst eine Vorhersage (*Prädiktion*) über den Zustand und die Ausgabe zum nächsten Zeitpunkt gemacht und dann anhand der Abweichung zur tatsächlichen Messung eine Korrektur des aktuellen Zustands vorgenommen.

4.1.1 Prädiktion

Es wird also zunächst der Zustand $\underline{\boldsymbol{x}}(k)$ zum Zeitpunkt k basierend auf den Messungen $\mathcal{Y}_{k-1} = y(i)_{i=k_0}^{k-1}$ bis zum Zeitpunkt $k-1$ bzw. der letzten Zustandsschätzung $\underline{\hat{x}}(k-1|\mathcal{Y}_{k-1})$ prädiziert. Dafür wird der Erwartungswert von (4.1) mittels der Zustandsübergangsmatrix $\mathbf{A}(k)$ berechnet [5, §2.4]

$$\underline{\hat{x}}(k|\mathcal{Y}_{k-1}) = \mathbf{A}(k)\underline{\hat{x}}(k-1|\mathcal{Y}_{k-1}). \tag{4.3}$$

Der erwartete Fehler der Prädiktion wird durch die prädzierte Zustandsfehlerkovarianzmatrix

$$\mathbf{P}(k|\mathcal{Y}_{k-1}) = \mathbf{A}(k)\mathbf{P}(k-1|\mathcal{Y}_{k-1})\mathbf{A}(k)^T + \mathbf{Q}(k) \tag{4.4}$$

mit der *Modellfehlerkovarianz* $\mathbf{Q}(k)$ beschrieben. Die prädizierte Messung erfolgt dann als Erwartungswert von (4.2) für den prädizierten Zustand $\underline{\hat{x}}(k|\mathcal{Y}_{k-1})$

$$\underline{\hat{y}}(k|\mathcal{Y}_{k-1}) = \mathbf{C}(k)\underline{\hat{x}}(k|\mathcal{Y}_{k-1}). \tag{4.5}$$

Für den weiteren Verlauf soll die Schreibweise der Prädiktionen etwas vereinfacht werden

$$\begin{aligned}
\underline{\hat{x}}(k|k-1) &= \underline{\hat{x}}(k|\mathcal{Y}_{k-1}) && (4.6)\\
\underline{\hat{x}}(k) &= \underline{\hat{x}}(k|\mathcal{Y}_{k}) && (4.7)\\
\mathbf{P}(k|k-1) &= \mathbf{P}(k|\mathcal{Y}_{k-1}) && (4.8)\\
\mathbf{P}(k) &= \mathbf{P}(k|\mathcal{Y}_{k}) && (4.9)\\
\underline{\hat{y}}(k|k-1) &= \underline{\hat{y}}(k|\mathcal{Y}_{k-1}). && (4.10)
\end{aligned}$$

4.1.2 Korrektur

Für die Korrektur wird nun der *Innovationsvektor*

$$\underline{r}(k) = \underline{y}(k) - \underline{\hat{y}}(k|k-1) \tag{4.11}$$

als Differenz zwischen tatsächlicher Messung $\underline{y}(k)$ und deren Prädiktion berechnet. Der Fehler dieses *Residuums* drückt wiederum die *Innovationskovarianzmatrix*

$$\mathbf{S}(k) = \mathbf{C}(k)\mathbf{P}(k|k-1)\mathbf{C}(k)^T + \mathbf{R}(k) \tag{4.12}$$

mit der *Messfehlerkovarianz* $\mathbf{R}(k)$ aus.

Mit dem *Kalman Gain* wird nun die Gewichtung der Messung gegenüber der Prädiktion bestimmt

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{C}(k)^T\mathbf{S}^{-1}. \tag{4.13}$$

Dadurch ergibt sich die neue a-posteriori Schätzung des Zustands als

$$\underline{\hat{x}}(k) = \underline{\hat{x}}(k|k-1) + \mathbf{K}(k)\underline{r}(k) \tag{4.14}$$

und der Fehler beschrieben durch die Zustandsfehlerkovarianzmatrix zu

$$\mathbf{P}(k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{C}(k))\mathbf{P}(k|k-1). \tag{4.15}$$

4.1.3 Randbedingungen für Zustände

Sollen bei der Schätzung der Zustände Randbedingungen eingehalten werden, so kann der Kalman Filter entsprechend für Gleichheitbedingungen [50] bzw. Ungleichheitsbedingungen [51] erweitert werden.

4.2 Unscented Kalman Filter

Will man die Modellparameter eines linearen Modells oder einen Teil davon zusätzlich zu dem Zustand des Modells schätzen, um ein Finetuning zu erhalten, wird das Modell nicht-linear

$$\underline{y}(k) = g(\underline{x}(k)). \tag{4.16}$$

Hierfür eignet sich der *Unscented Kalman Filter* (UKF) [23], der eine Linearisierung der Transformationsfunktion $g(\underline{x}(k))$ mittels mehrerer *Sigmapunkte* durchführt, auf die jeweils die Transformationsfunktion angewandt wird. Diese Linearisierung nennt sich *Unscented Transformation*.

Der UKF hat gegenüber dem *Extended Kalman Filter* (EKF) [5, §3.3], bei dem die Linearisierung über eine Taylor-Approximation erster Ordnung vorgenommen wird, den Vorteil, dass keine Jakobimatrix gebildet werden muss. Beim UKF wird im Gegensatz dazu auch nicht die nicht-lineare Funktion an sich approximiert, sondern die Verteilung einer N-dimensionalen Zufallsvariable $\underline{\boldsymbol{x}}(k)$.

Im Folgenden wird der Zeitindex k der Übersichtlichkeit halber weggelassen, wo er selbsterklärend ist.

4.2.1 Unscented Transformation

Die Unscented Transformation (UT) approximiert also eine nicht-lineare Funktion, indem zuerst $2 \cdot N + 1$ Sigmapunkte $\mathcal{X}_i$ mit zugehörigen Gewichten w_i gewählt werden. Diese werden so gewählt, dass sie als Mittelwert und Kovarianzmatrix $\underline{\hat{x}}$ und $\mathbf{P}$ haben. Dies entspricht oberflächlich einer *Monte Carlo Methode*. Allerdings werden bei der UT die Samples nicht zufällig, sondern deterministisch gewählt [23, §3]

$$\begin{aligned}
\mathcal{X}_0 &= \underline{\hat{x}} \\
w_0 &= \frac{\kappa}{N+\kappa} \\
\mathcal{X}_i &= \underline{\hat{x}} + \left(\sqrt{(N+\kappa)\mathbf{P}}\right)_i \\
w_i &= \frac{1}{2(N+\kappa)} \\
\mathcal{X}_{i+N} &= \underline{\hat{x}} - \left(\sqrt{(N+\kappa)\mathbf{P}}\right)_i \\
w_{i+N} &= w_i,
\end{aligned} \tag{4.17}$$

wobei $\kappa \in \mathbb{R}$ mit $N + \kappa \neq 0$. $\left(\sqrt{(N+\kappa)\mathbf{P}}\right)_i$ ist die i-te Spalte der Matrixquadratwurzel $\mathbf{A}$ von $(N+\kappa)\mathbf{P} = \mathbf{A}\mathbf{A}^T$. Eine effiziente und stabile Methode, die Matrixquadratwurzel zu berechnen, ist z.B. die Choleskyzerlegung.

Aus den resultierenden Sigmapunkte werden die transformierten Sigmapunkte

$$\underline{\mathcal{Y}}_i = g(\underline{\mathcal{X}}_i) \tag{4.18}$$

berechnet. Die gewichtete Summe der transfomierten Sigmapunkte ergibt dann den Mittelwert des transformierten Zustands

$$\underline{\hat{y}} = \sum_{i=0}^{2N} w_i \underline{\mathcal{Y}}_i. \tag{4.19}$$

Die dazugehörige Fehlerkovarianzmatrix $\mathbf{P}_y$ und Kreuzkovarianzmatrix $\mathbf{P}_{xy}$ ist

$$\mathbf{P}_y = \sum_{i=0}^{2N} w_i (\underline{\mathcal{Y}}_i - \underline{\hat{y}})(\underline{\mathcal{Y}}_i - \underline{\hat{y}})^T \tag{4.20}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2N} w_i (\underline{\mathcal{X}}_i - \underline{\hat{x}})(\underline{\mathcal{Y}}_i - \underline{\hat{y}})^T. \tag{4.21}$$

Die UT hat allerdings bei nicht-linearen Modellen hoher Dimension den Nachteil, sehr rechenaufwendig zu sein, auf Grund der Anzahl an Sigmapunkten, die für die Approximation nötig sind und für die jeweils die Transformationsfunktion berechnet werden muss.

4.2.2 Unscented Transformation für bedingt lineare Modelle

In unserem Fall handelt es sich aber nicht um ein rein nicht-lineares Modell, sondern um ein bedingt lineares Modell, da es, wenn man die Modellparameter festhält, wieder ein lineares System ist. Die Transformationsfunktion lässt sich somit in einen nicht-linearen und einen bedingt linearen Anteil trennen

$$g(\underline{x}) = g(\underline{x}_A) + \mathbf{G}(\underline{x}_A)\underline{x}_B, \tag{4.22}$$

wobei $\mathbf{G}(\underline{x}_A)$ die bedingte Transformationsmatrix ist und $\underline{x}_A$ und $\underline{x}_B$ respektive der nicht-lineare und lineare Anteil des Zustands $\underline{x} = [\underline{x}_A^T, \underline{x}_B^T]^T$ sind. Die Kovarianzmatrix $\mathbf{P}$ des Zustands kann ebenso partitioniert werden

$$\mathbf{P} = \begin{pmatrix} \mathrm{Cov}(\underline{\boldsymbol{x}}_A|\underline{\boldsymbol{x}}_A) & \mathrm{Cov}(\underline{\boldsymbol{x}}_A|\underline{\boldsymbol{x}}_B) \\ \mathrm{Cov}(\underline{\boldsymbol{x}}_B|\underline{\boldsymbol{x}}_A) & \mathrm{Cov}(\underline{\boldsymbol{x}}_B|\underline{\boldsymbol{x}}_B) \end{pmatrix} = \begin{pmatrix} \mathbf{P}_A & \mathbf{P}_{AB} \\ \mathbf{P}_{AB}^T & \mathbf{P}_B \end{pmatrix}. \tag{4.23}$$

Für solche bedingt linearen Modelle, kann die Anzahl der nötigen Sigmapunkte auf $2 \cdot N_A + 1$ reduziert werden [37], wobei N_A die Anzahl der nicht-linearen Dimensionen ist, indem man die im Folgenden beschriebene Unscented Transformation für bedingt lineare Modelle verwendet.

Hierfür wählt man zuerst die Sigmapunkte $\underline{\mathcal{X}}_{i,A}$ und deren zugehörige Gewichte w_i für den nicht-linearen Anteil des Zustandsvektors analog der Sigmapunktewahl bei der Standard UT (4.17) für die Kovarianzmatrix $\mathbf{P}_A$.

Daraufhin werden die Sigmapunkte $\underline{\mathcal{X}}_{i,B}$ und die bedingte Kovarianzmatrix $\mathbf{\Gamma}$ des bedingt linearen Anteils berechnet

$$\underline{\mathcal{X}}_{i,B} = \mathrm{E}(\underline{\boldsymbol{x}}_B|\underline{\mathcal{X}}_{i,A}) = \underline{x}_B + \mathbf{P}_{AB}^T \mathbf{P}_A^{-1}(\underline{\mathcal{X}}_{i,A} - \underline{x}_A) \tag{4.24}$$

$$\mathbf{\Gamma} = \mathrm{Cov}(\underline{\boldsymbol{x}}_B|\underline{\boldsymbol{x}}_A) = \mathbf{P}_B - \mathbf{P}_{AB}^T \mathbf{P}_A^{-1} \mathbf{P}_{AB}. \tag{4.25}$$

Daraus ergeben sich die Sigmapunkte für den gesamten Zustandsraum zu

$$\underline{\mathcal{X}}_i = \begin{pmatrix} \underline{\mathcal{X}}_{i,A} \\ \underline{\mathcal{X}}_{i,B} \end{pmatrix}. \tag{4.26}$$

Aus diesen Sigmapunkten werden dann die transformierten Sigmapunkte nach (4.18) berechnet.

Für die weitere Verarbeitung im UKF wird nun der Mittelwert $\underline{y}$ des transformierten Zustandes, dessen Kovarianzmatrix $\mathbf{P}_y$ und Kreuzkovarianzmatrix $\mathbf{P}_{xy}$ approximiert

$$\hat{\underline{y}} = \sum_{i=0}^{2N_A} w_i \underline{\mathcal{Y}}_i \tag{4.27}$$

$$\mathbf{P}_y = \sum_{i=0}^{2N_A} w_i [(\underline{\mathcal{Y}}_i - \hat{\underline{y}})(\underline{\mathcal{Y}}_i - \hat{\underline{y}})^T + \mathbf{G}(\underline{\mathcal{X}}_{i,A}) \mathbf{\Gamma} \mathbf{G}(\underline{\mathcal{X}}_{i,A})^T] \tag{4.28}$$

$$\mathbf{P}_{xy} = \sum_{i=0}^{2N_A} w_i \left[(\underline{\mathcal{X}}_i - \hat{\underline{x}})(\underline{\mathcal{Y}}_i - \hat{\underline{y}})^T + \begin{pmatrix} \mathbf{0} \\ \mathbf{\Gamma} \mathbf{G}(\underline{\mathcal{X}}_{i,A})^T \end{pmatrix} \right]. \tag{4.29}$$

4.2.3 UKF Algorithmus

Um nun die Zustandsschätzung für nicht-lineare Transformationsfunktionen wie beim Kalman Filter durchzuführen, wird die Unscented Transformation jeweils für die zu approximierende Transformationsfunktion auf deren Argument und zugehörige Kovarianzmatrix angewendet [24, §3.2].

Nicht-lineares Systemmodell mit additivem Rauschen

Ist das Systemmodell nicht-linear, so wird die Unscented Transformation für das Systemmodell unter Verwendung von $\underline{x}(k-1|k-1)$ und $\mathbf{P}(k-1|k-1)$ durchgeführt.

Der resultierende Mittelwert des transformierten Zustands aus (4.19) bzw. (4.27) ersetzt den prädizierten Zustand $\hat{\underline{x}}(k|k-1)$ aus (4.3) und aus der resultierenden Kovarianzmatrix von (4.20) bzw. (4.28) $\mathbf{P}_{UT}(k|k-1) \triangleq \mathbf{P}_y$ erhält man die prädizierte Zustandsfehlerkovarianzmatrix

$$\mathbf{P}(k|k-1) = \mathbf{P}_{UT}(k|k-1) + \mathbf{Q}(k). \tag{4.30}$$

Nicht-lineares Messmodell mit additivem Rauschen

Bei einem nicht-linearen Messmodell wird entsprechend die Unscented Transformation für das Messmodell unter Verwendung von $\underline{x}(k|k-1)$ und $\mathbf{P}(k|k-1)$ durchgeführt.

Der resultierende Mittelwert des transformierten Zustands aus (4.19) bzw. (4.27) ersetzt nun die prädizierte Messung $\hat{\underline{y}}(k|k-1)$ aus (4.5) und aus der resultierenden Kovarianzmatrix von (4.20) bzw. (4.28) $\mathbf{S}_{UT}(k|k-1) \triangleq \mathbf{P}_y$ erhält man die Innovationskovarianzmatrix

$$\mathbf{S}(k) = \mathbf{S}_{UT}(k|k-1) + \mathbf{R}(k). \tag{4.31}$$

Die Kreuzkovarianzmatrix $\mathbf{P}_{xy}$ aus (4.21) bzw. (4.29) wird zur Berechnung des Kalman Gains

$$\mathbf{K}(k) = \mathbf{P}_{xy}(k)\mathbf{S}(k)^{-1} \tag{4.32}$$

verwendet. Zur Berechnung der a-posteriori Zustandsfehlerkovarianzmatrix (4.15) wird mangels $\mathbf{C}(k)$ eine alternative Formel verwendet [66]

$$\mathbf{P}(k) = \mathbf{P}(k|k-1) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}(k)^T. \tag{4.33}$$

Alle anderen Berechnungen erfolgen analog wie beim Kalman Filter.

4.3 Interacting Multiple Model Algorithmus

Da das Ziel dieser Arbeit die Trennung von Musikaufnahmen ist, sollen nun, aus der Signalmixtur, die vorliegt, die einzelnen Signale rekonstruiert werden. Allerdings handelt es sich dabei um nicht-stationäre Signale, d.h. in diesem Fall, dass sie entweder vorhanden sind oder nicht.

Da dieser aktuelle Zustand nicht bekannt ist, ist es ein intuitiver Ansatz, mehrere Modelle parallel darauf zu testen, ob sie die aktuelle Signalmixtur beschreiben.

Dafür eignet sich der *Interacting Multiple Model Algorithmus* (IMM) [7, 8, 30, 36]. Ein Stukturdiagramm dieses Algorithmus' ist in Abb. 4.1 abgebildet. Hier wird der aktuelle Zustand $\underline{\hat{x}}_j(k)$ und die zugehörige Modellwahrscheinlichkeit μ_j für M parallel laufende Modelle geschätzt. Vor jeder Iteration wird ein *Interaktionsschritt* durchgeführt, bei dem, anhand der aktuellen Modellwahrscheinlichkeiten und einer gegebenen Modellübergangswahrscheinlichkeit p_{ij}, eine *Mischwahrscheinlichkeit* $\mu_{i|j}$ berechnet wird. Diese dient als Gewichtungsfaktor zur Berechnung neuer Mischzustände $\underline{x}_{0j}(k)$, die den einzelnen Schätzern der Modelle als Initialzustand für die nächste Iteration dienen. Abgeschlossen wird eine Iteration mit einem *Kombinationsschritt*, der einen, entsprechend der Modellwahrscheinlichkeiten, gewichteten Zustand $\underline{\hat{x}}(k)$ berechnet, der das Signal zum Zeitpunkt k am besten beschreibt.

Im Folgenden soll dieser Algorithmus nun im Detail beschrieben werden [36, §3.1].

4.3.1 Interaktion

Im Interaktionsschritt werden zunächst die Mischwahrscheinlichkeiten

$$\mu_{i|j}(k-1|k) = \frac{p_{ij}\mu_i(k-1)}{\mu_j(k|k-1)} \tag{4.34}$$

auf Basis der prädizierten Modellwahrscheinlichkeiten [39, §IV.C]

$$\mu_j(k|k-1) = \sum_i p_{ij}\mu_i(k-1) \tag{4.35}$$

und der Modellübergangswahrscheinlichkeiten

$$p_{ij} = P(m_j(k)|m_i(k-1)) \tag{4.36}$$

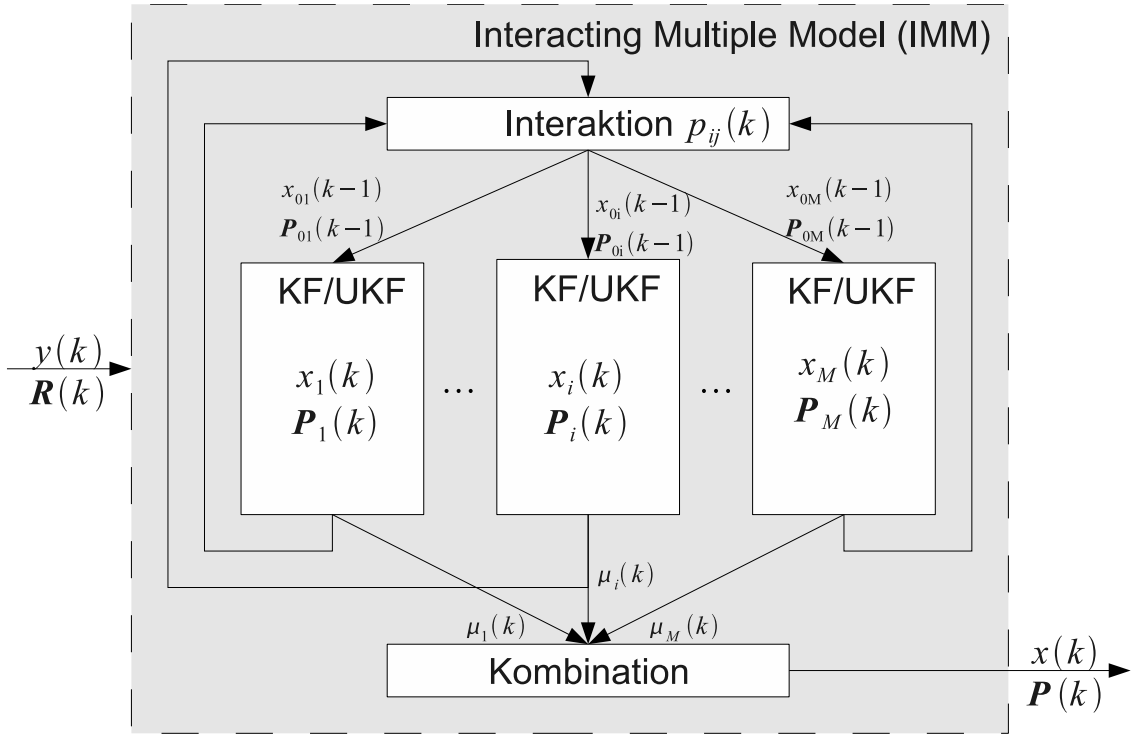

Abbildung 4.1: Strukturdiagramm für den Interacting Multiple Model Algorithmus.

von Modell i, zum Zeitpunkt $k-1$, zu Modell j, zum Zeitpunkt k, zu wechseln, für alle $i, j = 1, \ldots, M$ berechnet.

Daraus lässt sich nun der Mittelwert und die Kovarianzmatrix für den Fall bestimmen, dass zum Zeitpunkt k das Modell j aktiv ist, gegeben der Messungen $\mathcal{Y}_{k-1}$ bis zum Zeitpunkt $k-1$

$$\underline{\hat{x}}_{0j}(k-1|k-1) = \sum_{i=1}^{M} \underline{\hat{x}}_i(k-1|k-1)\mu_{i|j}(k-1|k) \tag{4.37}$$

$$\begin{aligned}\mathbf{P}_{0j}(k-1|k-1) = \sum_{i=1}^{M} \{&\mathbf{P}_i(k-1|k-1) + [\underline{\hat{x}}_i(k-1|k-1) - \underline{\hat{x}}_{0j}(k-1|k-1)] \\ &\cdot [\underline{\hat{x}}_i(k-1|k-1) - \underline{\hat{x}}_{0j}(k-1|k-1)]^T\}\mu_{i|j}(k-1|k).\end{aligned} \tag{4.38}$$

4.3.2 Filterung

Im Filterungsschritt wird jetzt für jedes der M Modelle jeweils eine Prädiktion und Korruktur mittels eines Kalman Filters oder Unscented Kalman Filters oder ähnlichem durchgeführt. Dabei ersetzen die aus dem Interaktionsschritt gewonnenen Mittelwerte $\underline{\hat{x}}_{0j}(k-1|k-1)$ und Kovarianzmatrizen $\mathbf{P}_{0j}(k-1|k-1)$ die sonst in der Prädiktion verwendeten, aus der letzten Schätzung resultierenden, Mittelwerte und Kovarianzmatrizen.

Aus dem Residuum $\underline{r}_j(k)$ und der Innovationskovarianzmatrix $\mathbf{S}_j(k)$ ergibt sich die Likelihood-Funktion

$$\Lambda_j(k) = N(\underline{r}_j(k); 0, \mathbf{S}_j(k)). \tag{4.39}$$

Die neue Modellwahrscheinlichkeit

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k) \sum_i p_{ij}\mu_i(k-1) = \frac{1}{c}\Lambda_j(k)\mu_j(k|k-1) \tag{4.40}$$

für das j-te Modell zum Zeitpunkt k wird nun mittels dieser Likelihoodfunktion und einem Normalisierungsfaktor

$$c = \sum_i \Lambda_j(k)\mu_j(k|k-1) \tag{4.41}$$

berechnet.

4.3.3 Kombination

Für eine Gesamtschätzung des Zustands werden, analog zu (4.37), die modell-spezifischen Zustandsschätzungen $\underline{\hat{x}}_j(k|k)$ jeweils mit den Modellwahrscheinlichkeiten $\mu_j(k)$ gewichtet und aufsummiert

$$\underline{\hat{x}}(k|k) = \sum_j \underline{\hat{x}}_j(k|k)\mu_j(k). \tag{4.42}$$

Die dazugehörige kombinierte Zustandsfehlerkovarianzmatrix wird analog zu (4.38) berechnet

$$\mathbf{P}(k|k) = \sum_j \{\mathbf{P}_j(k|k) + [\underline{\hat{x}}_j(k|k) - \underline{\hat{x}}(k|k)][\underline{\hat{x}}_j(k|k) - \underline{\hat{x}}(k|k)]^T\}\mu_j(k). \tag{4.43}$$

KAPITEL 5

Experimente

Im Folgenden werden nun durchgeführte Experimente beschrieben und analysiert, sowie Überlegungen zu Optimierungsmöglichkeiten angestellt.

5.1 Trennung zweier Sinusschwingungen

Als einen ersten Versuch soll eine Mixtur aus zwei nicht-stationären Sinusschwingungen getrennt werden. Dazu wird der Interacting Multiple Model Algorithmus aus Abschnitt 4.3 verwendet.

Da es sich bei den Sinusschwingungen um nicht-stationäre Signale handelt, benötigen wir für jede der zu trennenden Schwingungen ein Modell für Stille und eines für die Sinusschwingung an sich. Für beide dieser benötigten Modelle wird das in (2.19) definierte Zustandsraummodell verwendet. Allerdings wird das Messmodell für Stille durch

$$\mathbf{C}_0 = \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{5.1}$$

ersetzt. Die Zustandsfehlerkovarianzmatrix wurde für beide Modelle mit $\mathbf{P}_0 = \mathbf{P} = \mathbf{I}^{2\times 2}$ initialisiert. Die Modellfehlerkovarianzmatrix wurde für Stille mit $\mathbf{Q}_0 = \mathbf{B}\mathbf{B}^T$ und für Aktivität mit $\mathbf{Q} = (10^{-12})^2\mathbf{B}\mathbf{B}^T$ gewählt und die Messfehlerkovarianz auf $\mathbf{R} = 0{,}01^2$ festgelegt. Der Dämpfungsfaktor wurde für beide Schwingungen auf $\rho = 1$ und die Frequenzen auf $f_1 = 27{,}5 \cdot 2^{\frac{49-1}{12}} = 440$ Hz und $f_2 = 27{,}5 \cdot 2^{\frac{46-1}{12}} \approx 369{,}9944$ Hz gesetzt. Das entspricht dem Ton a' und einem f#'. Dabei wurde eine Abtastrate von $f_s = 44100$ Hz gewählt. Die Anfangsmodellwahrscheinlichkeiten für Stille und Aktivität wurden jeweils mit $\mu_j = 0{,}5$ festgesetzt. Die Übergangswahrscheinlichkeit im selben Zustand zu bleiben wurde mit $p_{ii} = 0{,}98$ und die für einen Zustandswechsel mit $p_{ij} = 0{,}02$ definiert.

Mit diesen Parametern wurden Signale der beiden Schwingungen von jeweils zehn Periodenlängen generiert. Dafür wurde der Zustandsvektor anfangs mit $\mathbf{B}$ initialisiert. Die Signalmixtur wurde so erstellt, dass zuerst das erste Signal, dann das zweite Signal und schließlich beide zusammen auftreten, wobei hier das zweite Signal um die zweifache Periodenlänge der ersten Schwingung verzögert wurde. Dabei wurde am Anfang und am Ende jeweils eine Pause von 10 Samples und zwischen den einzelnen Sektionen Pausen von jeweils 100 Samples eingefügt.

Anschliessend wurde auf das resultierende Signal weißes Rauschen entsprechend der Messfehlerkovarianzmatrix addiert. Die Einzelsignale und die daraus resultierende Mixtur sind in Abb. 5.1 und Abb. 5.2 respektive Abb. 5.3 zu sehen.

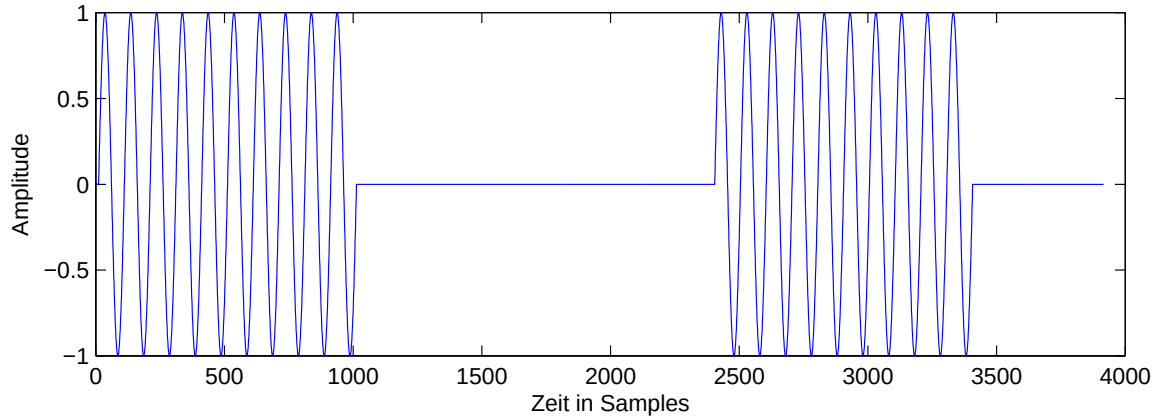

Abbildung 5.1: Originalsignal der ersten Sinusschwingung.

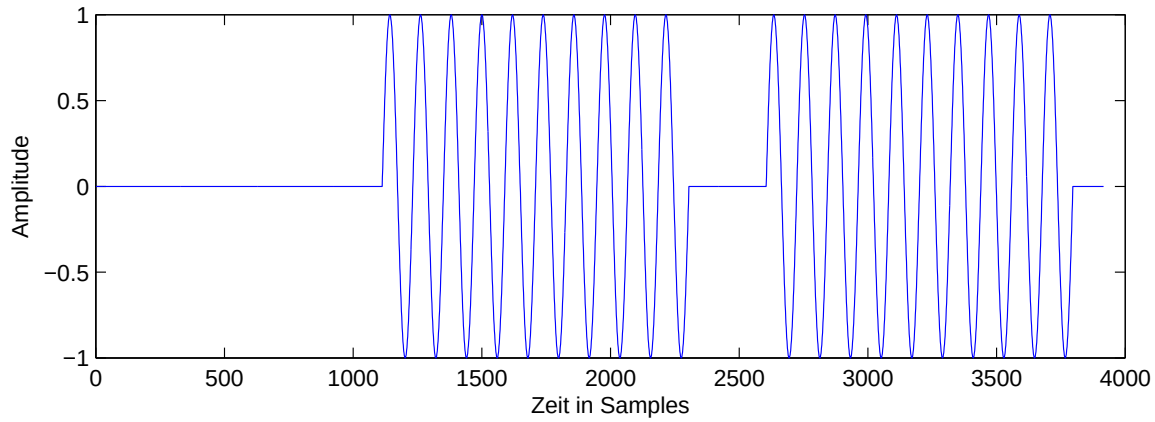

Abbildung 5.2: Originalsignal der zweiten Sinusschwingung.

Zur Trennung der Mixturen werden nun aus den beiden Modellen mit ihren Untermodellen alle Kombinationen gebildet. Es wird also jedes Untermodell des ersten Schwingungsmodells mit jedem Untermodell des zweiten Schwingungsmodells kombiniert. Diese Kombinationen werden analog zu (2.20) aus den Einzelmodellen zusammengesetzt. Daraus resultiert also die Partitionierung

$$\underline{x}_l(k) = \begin{pmatrix} \underline{x}_{1,i}(k) \\ \underline{x}_{2,j}(k) \end{pmatrix} \quad , \quad \mathbf{P}_l(k) = \begin{pmatrix} \mathbf{P}_{1,i}(k) & \mathbf{P}_{(1,i),(2,i)}(k) \\ \mathbf{P}_{(1,i),(2,i)}(k)^T & \mathbf{P}_{2,j}(k) \end{pmatrix}$$

$$\mathbf{A}_l = \begin{pmatrix} \mathbf{A}_{1,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{2,j} \end{pmatrix} \quad , \quad \mathbf{Q}_l = \begin{pmatrix} \mathbf{Q}_{1,i} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{2,j} \end{pmatrix}, \quad \mathbf{C}_l = \begin{pmatrix} \mathbf{C}_{1,i}^T \\ \mathbf{C}_{2,j}^T \end{pmatrix}^T . \tag{5.2}$$

Hierbei zeigt der Index i bzw. j den Index des jeweiligen Untermodells des ersten bzw. zweiten Schwingungsmodells an. Daraus ergeben sich in diesem Fall $2 \cdot 2 = 4$ Kombinationen. Für den

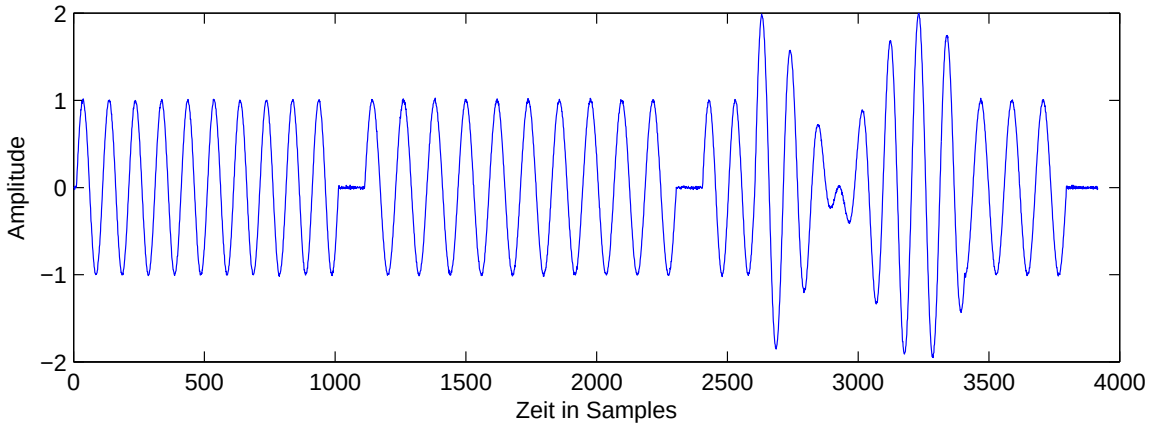

Abbildung 5.3: Mixtur zweier Sinusschwingungen mit additivem Rauschen.

allgemeinen Fall ergibt sich die Anzahl der Kombinationen zu

$$M = \prod_{i=1}^{N} M_i, \tag{5.3}$$

wobei M_i die Anzahl der Untermodelle des i-ten Schwingungsmodells angibt.

Für das IMM wurde auf eine Matlab Toolbox names EKF/UKF [47] zurückgegriffen und nach Bedarf erweitert. Zudem wurde vor jedem Iterationsschritt die Kovarianzmatrix des Modells für die Stille beider Töne auf die Identitätsmatrix gesetzt, um das Wachsen der Kovarianzmatrix für diesen Fall einzudämmen, da sonst das System instabil wird.

Die geschätzten Einzelsignale der beiden Töne sind in Abb. 5.4 bzw. Abb. 5.5 auf der folgenden Seite und die resultierende Summe der Schätzungen in Abb. 5.6 auf der nächsten Seite abgebildet.

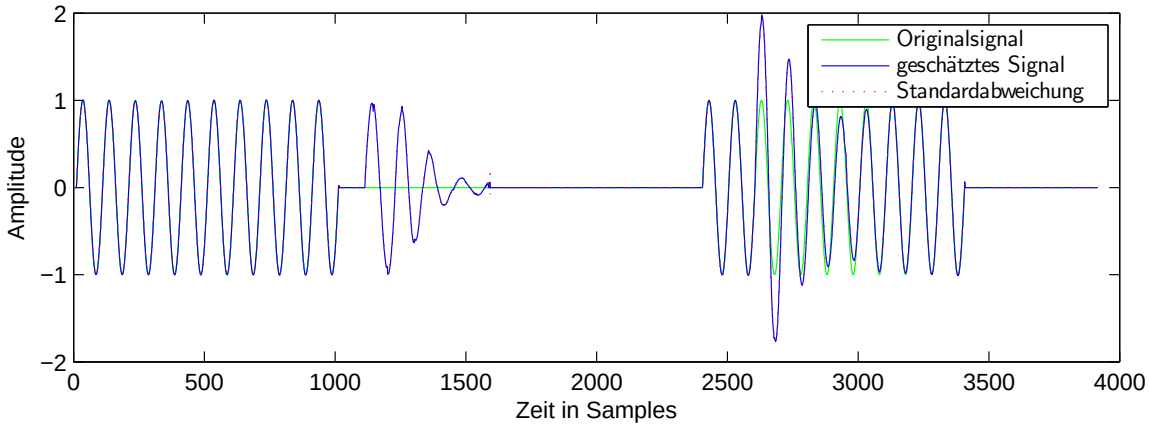

Abbildung 5.4: Signalschätzung für den ersten Sinuston.

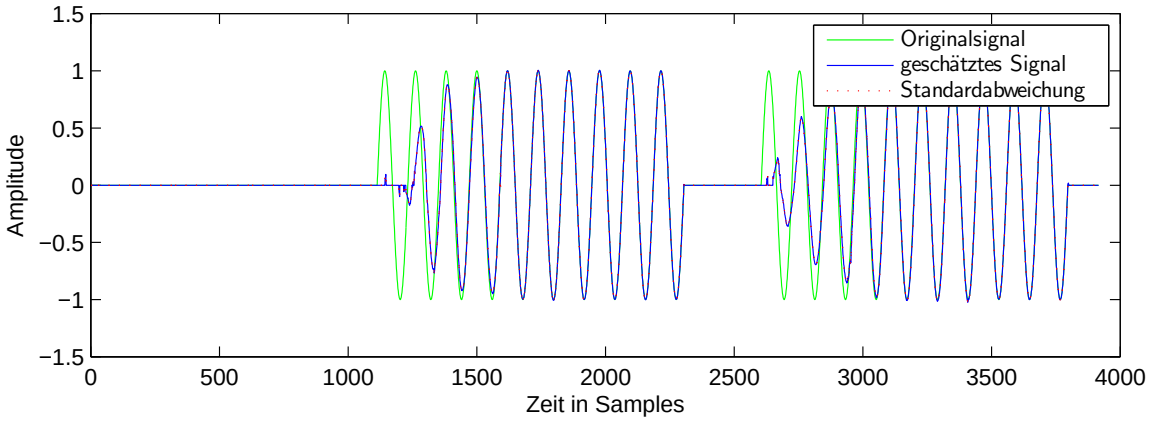

Abbildung 5.5: Signalschätzung für den zweiten Sinuston.

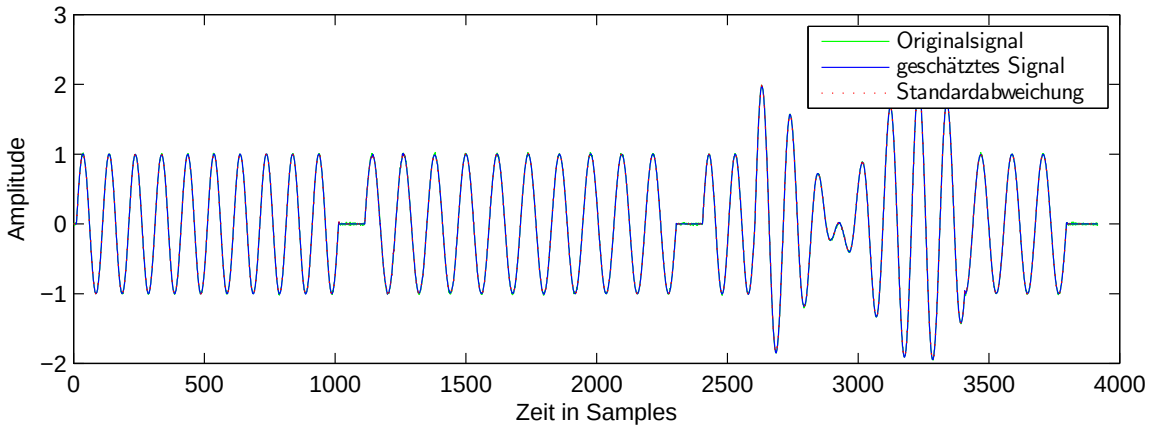

Abbildung 5.6: Summensignalschätzung für die Trennung zweier Sinusschwingungen.

Man kann erkennen, dass die Summe korrekt geschätzt wurde, wohingegen die Einzelsignale Fehler in der Schätzung aufweisen. Dies liegt an den teilweise falsch geschätzten Modellwahrscheinlichkeiten über die Zeit (siehe Abb. 5.7).

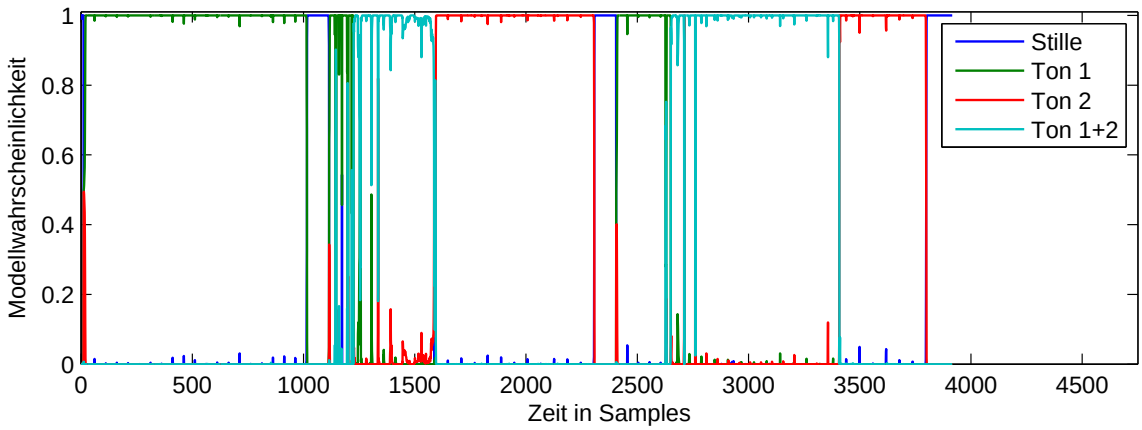

Abbildung 5.7: Modellwahrscheinlichkeiten für die Trennung zweier Sinusschwingungen.

Man sieht, dass die Wahrscheinlichkeiten relativ gut mit den tatsächlich zu den entsprechenden Zeiten gewählten Modellkombinationen übereinstimmen. Nur am Anfang einer neuen Kombination braucht das System etwas Zeit, bis das richtige Modell gefunden wird. Dabei sticht vor allem auch der Anfang der zweiten Sinusschwingung heraus. Die Ursache liegt hier wohl darin, dass sich einerseits die Schwingungen relativ ähnlich sind, da ihre Frequenzen nicht so weit auseinander liegen, und andererseits die Zustandsfehlerkovarianz für die erste Schwingung noch kleiner ist als die der zweiten Schwingung, da die Kovarianz der zweiten Schwingung durch die lange Stille angestiegen ist. Als Folge dessen wird die erste Schwingung präferiert wird. Es dauert etwa eine Periodelänge bis erkannt wird, dass der zweite Ton an dieser Stelle im Signal vorhanden ist. Daraufhin wird das Modell für die Aktivität beider Töne gewählt, da dieser Übergang wahrscheinlicher ist, als das beide Töne ihren Zustand gleichzeitig wechseln. In diesem Zustand bleibt das System bis die Schwingung des ersten Tones genügend gedämpft wurde, um sagen zu können, dass nur der zweite Ton aktiv ist, da beide Töne in der Summe das Eingangssignal korrekt modellieren. Dies folgt wohl aus der Fehlschätzung zum Anfang des zweiten Tones.

5.2 Trennung zweier Karplus-Strong-Schwingungen

Ein zweiter Versuch soll nun mit dem im Abschnitt 2.3.2 vorgestellten Karplus-Strong Modell unternommen werden. Hierfür wird das Zustandsraummodell aus (2.27) und die im vorangegangenen Abschnitt 5.1 gegebenen Parameter verwendet.

Bei der Erstellung der Mixtur wurde im Gegensatz zu den Sinusschwingungen der Zustandsvektor zufällig mit der Normalverteilung, gegeben durch die Zustandsfehlerkovarianzmatrix, mittelwertsfrei initialisiert. Dies entspricht dem Noiseburst, der als Eingangssignal für das Karplus-Strong-Modell verwendet wird. Die Einzelsignale und die daraus resultierende Mixtur sind in Abb. 5.8 und Abb. 5.9 respektive Abb. 5.10 zu sehen.

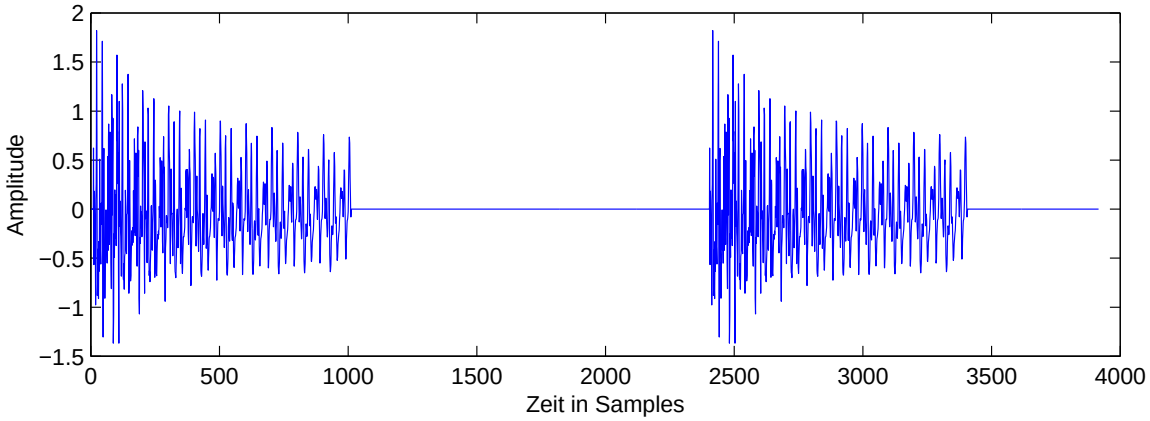

Abbildung 5.8: Originalsignal der ersten Karplus-Strong-Schwingung.

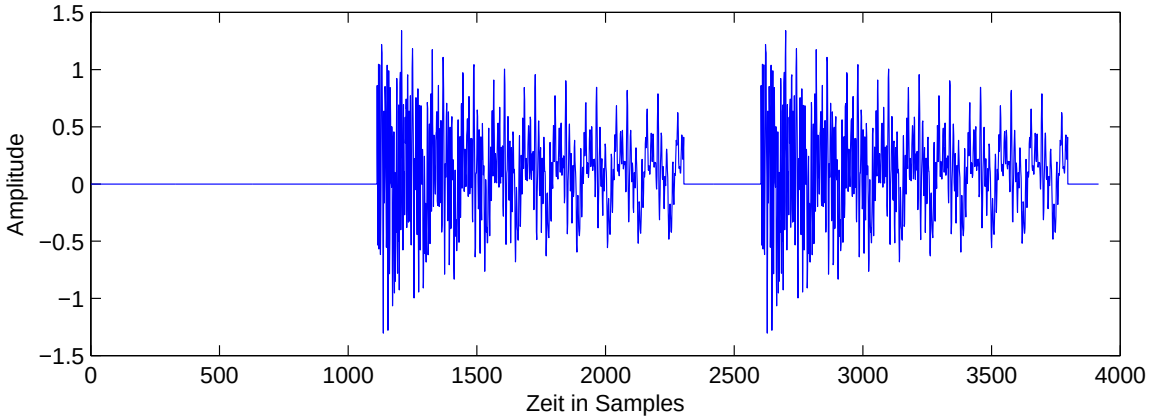

Abbildung 5.9: Originalsignal der zweiten Karplus-Strong-Schwingung.

Dadurch, dass die Anregung der Modelle als gauß'sches Rauschen modelliert wird, darf dieser Bereich in der Mixtur nicht aus Anregungsschwingungen mehrerer Quellen bestehen, da dies sonst nicht mehr trennbar ist. Außerdem ist in diesem Zusammenhang aufgrund der Parameterwahl vor jedem Ton eine Pause von mindestens einer Periodenlänge des Tons nötig, damit die Zustandsfehlerkovarianz sich wieder auf einen Wert einpendelt, um den neuen Noiseburst übernehmen zu können. Abhilfe dazu würde eine Modellierung der Anregungsschwingung schaffen, was allerdings bei diesem Beispiel aufgrund der zufälligen Wahl der Erregung nicht möglich ist.

In diesem Experiment wurde nicht vor jedem Iterationsschritt die Kovarianzmatrix des Modells für die Stille beider Töne auf die Identitätsmatrix gesetzt, so wie dies bei die Trennung der

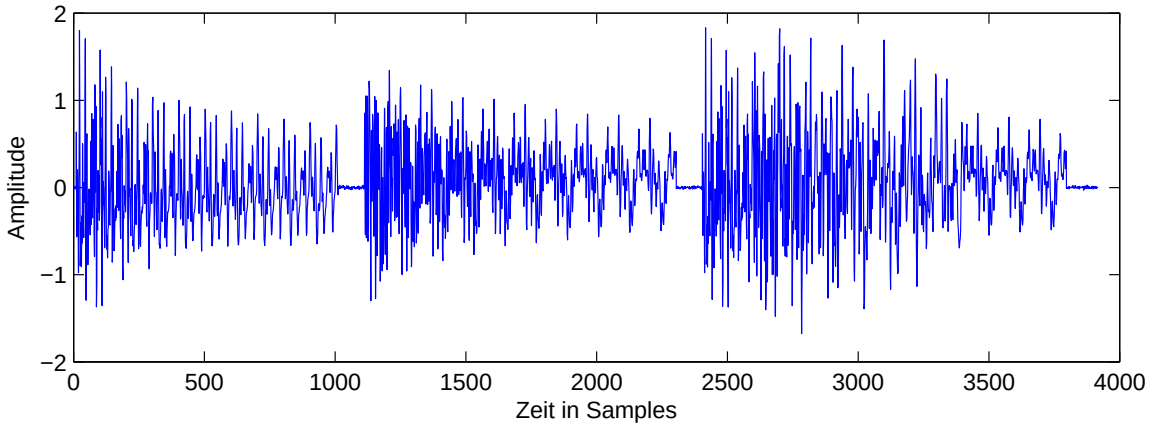

Abbildung 5.10: Mixtur zweier Karplus-Strong-Schwingungen mit additivem Rauschen.

Sinusschwingungen getan wurde, da dies bei den Karplus-Strong-Schwingungen zu schlechteren Ergebnissen führte.

Die geschätzten Einzelsignale der beiden Töne sind in Abb. 5.11 bzw. Abb. 5.13 auf der nächsten Seite und die resultierende Summe der Schätzungen in Abb. 5.15 auf Seite 63 abgebildet. Um den Unterschied zwischen den geschätzten Einzelsignalen und dem dazugehörigen Originalsignal besser zu erkennen ist zudem das Residuum für beide Signale in Abb. 5.12 auf der nächsten Seite bzw. Abb. 5.14 auf der folgenden Seite dargestellt.

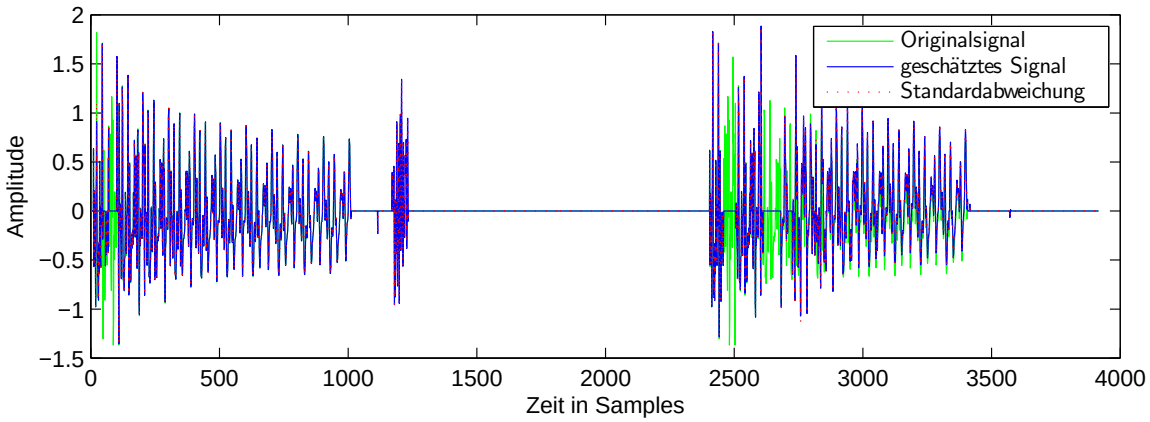

Abbildung 5.11: Signalschätzung für die erste Karplus-Strong-Schwingung.

Man sieht hier deutlich, dass die Summe der Schätzungen recht gut der Originalmixtur entspricht, aber de Einzelschätzungen erneut jeweils immer am Anfang eines Tones Fehler beinhalten, was man vor allem im Residuum, also der Differenz zwischen dem jeweiligen Originalsignal und der Schätzung, gut sehen kann. Bei der Trennung des überlagerten Teils, in der dritten Sektion der Mixtur, ist zudem ein größerer Fehler erkennbar. Das rührt wohl daher, dass das Anregungsrauschen nicht genau genug geschätzt wurde, bevor das zweite Signal dazu kam. In diesem Experiment kommt ohnehin erschwerend hinzu, dass eben genau diese Rauschsequenz,

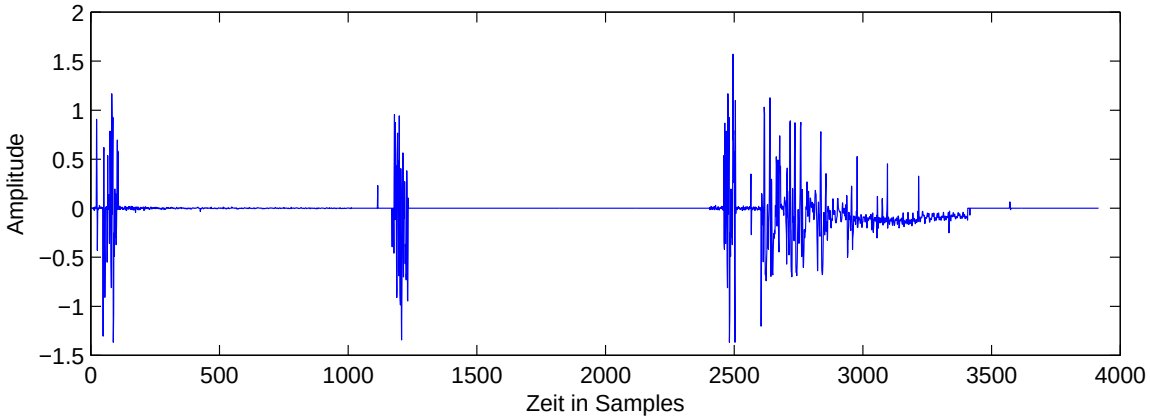

Abbildung 5.12: Differenz zwischen Originalsignal und Schätzung für die erste Karplus-Strong-Schwingung.

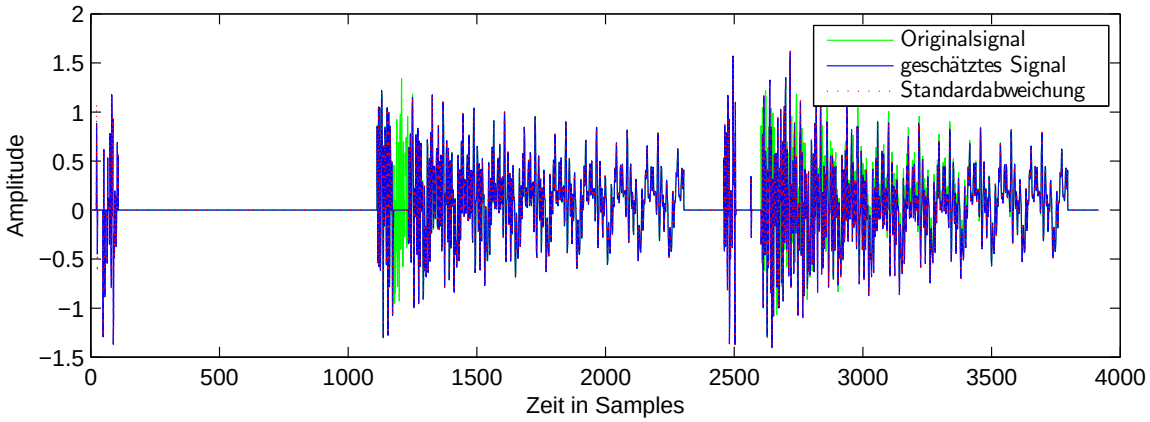

Abbildung 5.13: Signalschätzung für die zweite Karplus-Strong-Schwingung.

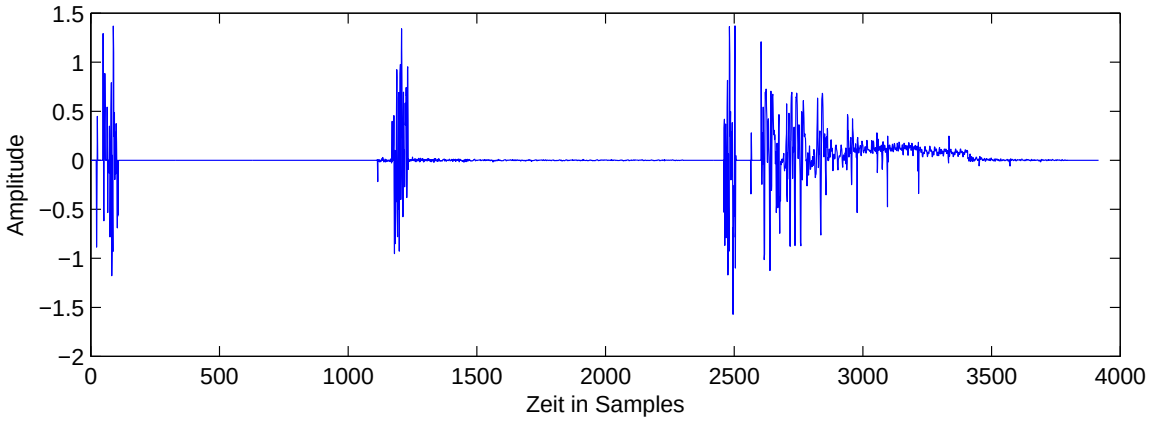

Abbildung 5.14: Differenz zwischen Originalsignal und Schätzung für die zweite Karplus-Strong-Schwingung.

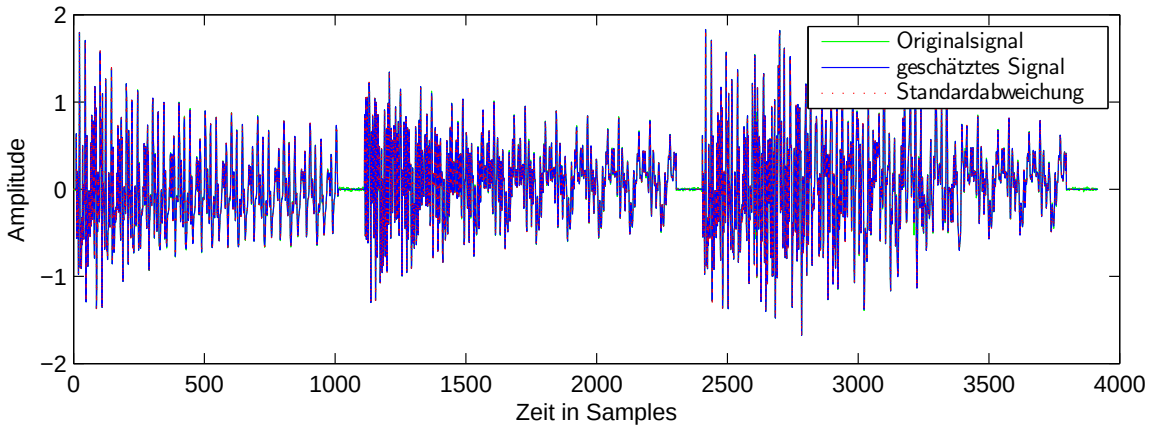

Abbildung 5.15: Summensignalschätzung für die Trennung zweier Karplus-Strong-Schwingungen.

die als Anregung dient, nicht deterministisch ist, wodurch in dieser Phase auch keine eindeutige Zuordnung zu einem bestimmten Ton stattfinden kann.

Andererseits hilft aber gerade diese unterschiedliche Anregung der beiden Töne in der Phase nach der Anregung dem Algorithmus, relativ schnell das korrekte Modell zu erkennen, was man an den Modellwahrscheinlichkeiten in Abb. 5.16 sehen kann.

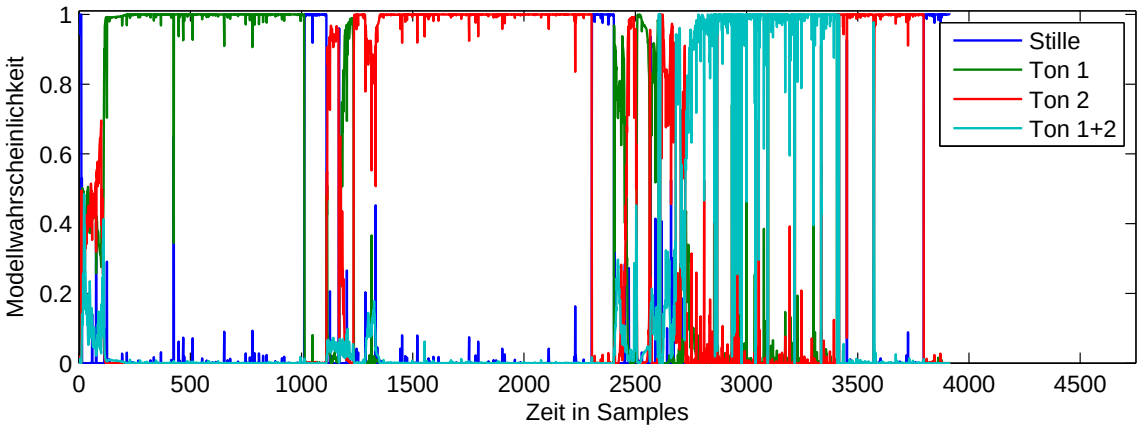

Abbildung 5.16: Modellwahrscheinlichkeiten für die Trennung zweier Karplus-Strong-Schwingungen.

Hier dauerte es wie bei den Sinusschwingungen auch etwa eine Periodenlänge bis eine definitive Aussage über das Modell getroffen werden konnte, jedoch wurde beim zweiten Ton dieses mal direkt nach der Anregung auch tatsächlich das korrekte Modell gewählt. Dafür traten hier verstärkt Probleme bei der Überlagerung in der dritten Sektion der Mixtur auf, da wie schon im Residuum zu sehen war, die Anregungssequenz nicht vollständig geschätzt werden konnte. Als Resultat ist die Wahrscheinlichkeit für das Modell, dass beide Töne aktiv sind, in dieser Passage hohen Schwankungen ausgesetzt.

5.3 Überlegungen zu Optimierungsmöglichkeiten

Bei den Experimenten sind einige Probleme sichtbar geworden, die einer Optimierung bedürfen. Dies ist zum einen die Komplexität durch die Kombinationen der Klangmodelle und zum anderen die korrekte Ermittlung des für die Trennung zu verwendenden Modells, also die Schätzung der Aktivität der einzelnen Quellen. Hier sollen nun mögliche Lösungen dafür betrachtet werden.

5.3.1 Komplexität

Alle Kombinationen der einzelnen Klangmodelle zu berechnen und zu verwenden ist für den praktischen Einsatz in der Regel nicht realisierbar, wenn es sich um eine größere Anzahl von Klangmodellen dreht. Deshalb sollen nun ein paar Überlegungen dazu angeführt werden, die Komplexität bzw. den Rechenaufwand zu reduzieren.

Modellmengen Adaption

Neben dem höheren Rechenaufwand kann es auch vorkommen, dass, aufgrund der großen Modellanzahl, viele der Modelle vom aktuellen Zustand des Systems sehr verschieden sind und so keinen Gewinn für die Zustandsschätzung bieten. Außerdem kann es zu unnötiger Konkurrenz zwischen den Modellen kommen, was zu Einbusen in den Schätzungen führen kann [36, §3.4]. Eine Lösung hierfür bietet die *Modellmengen Adaption* [36, §3.4] [31]. Hier werden die Zustände nicht immer für alle Modelle geschätzt, sondern nur für die wahrscheinlichsten.

Einzelmodelle

Vernachlässigt man die Korrelation zwischen einzelnen Klangmodellen, so kann man alternativ die bisher kombinatorischen Modelle so umformulieren, dass wieder die einzelnen Klangmodelle separat in IMMs verwendet werden. Dazu werden künstliche Messungen mit jeweils zugehörigen Messfehlerkovarianzen generiert, die sich wiederum als Kombination der tatsächlichen Messung und den Hypothesen der Einzelmodelle ergeben. Diese können dann dazu dienen, die aktuellen Zustände probabilistisch mit Algorithmen wie z.B. IMM-PDA [39], angepasst auf den Fall, dass die Messungen unterschiedliche Messfehlerkovarianzen haben, zu schätzen.

Dies ermöglicht es auch statt eines IMMs, was Stille und Aktivität modelliert einen integrierten Ansatz wie IMM-IPDA [39] zu verwenden, der die Existenzwahrscheinlichkeit mitschätzt und somit das IMM um das Stillemodell reduziert.

In diesem Zusammenhang wäre unter Umständen auch zu untersuchen, ob das Prinzip des *linearen Multitarget Trackings* [40, 38] bzw. ein Abwandlung davon hier angewendet werden kann, was die Komplexität des Problems erheblich reduzieren würde.

5.3.2 Robustheit der Modellschätzung

Um das, für die Trennung der Signale, korrekte Modell für einen bestimmten Zeitpunkt zu ermitteln, wurden bisher nur die Messungen bis zum Schätzzeitpunkt verwendet. Die Schätzung würde allerdings wesentlich robuster werden, wenn zudem auch zukünftige Messungen zur Schätzung verwendet würden.

Dies kann durch einen *Kalman-Smoother* erreicht werden. Dafür gibt es drei Varianten: Den *fixed-Interval Smoother*, den *fixed-Point Smoother* und den *fixed-Lag Smoother* [20, §4.3.3]. In [18] wurde beispielsweise der fixed-Lag Kalman-Smoother verwendet, um tieffrequente Störsignale aus Schallplattenaufnahmen zu entfernen. In [11] wurde ein fixed-Lag Smoother für den Interacting Multiple Model Algorithmus beschrieben.

KAPITEL 6

Fazit und Ausblick

In dieser Arbeit wurde ein Überblick über Modelle, wie das Sinusoidalmodell und das physikalisch motivierte Karplus-Strong Modell mit seinen Erweiterungen, gegeben, die für die Klangsynthese verwendet werden können. Außerdem wurde anhand eines Beispielklanges ein Analyseverfahren zur Schätzung der Modellparameter eines erweiterten Karplus-Strong Modells in Einzelschritten beschrieben. Des Weiteren wurde das Kalman Filter und das Unscented Kalman Filter, mit einer Erweiterung für bedingt lineare Modelle, sowie der Interacting Multiple Model Algorithmus beschrieben, um im Anschluss diese Algorithmen zusammen mit den vorgestellten Modellen zur Trennung von Signalmixturen im Zeitbereich zu verwenden. Dazu wurde das Verfahren auf einer Mixtur zweier Sinusschwingungen und einer Mixture zweier Karplus-Strong-Schwingungen getestet, die Ergebnisse veranschaulicht, sowie Optimierungsmöglichkeiten für die noch bestehenden Probleme präsentiert.

Das hier beschriebene Verfahren zur Trennung kommt ohne ein Fouriertransformation aus, da es komplett im Zeitbereich abläuft, desweiteren werden auch keine anderen Feature-Extraktionsverfahren wie Onset Detektion oder Grundtonschätzung zur Trennung verwendet, da dies über die unterschiedlichen, interagierenden Modelle und die Übergangswahrscheinlichkeiten geregelt wird.

Für die weitere Arbeit an dem vorgestellten Verfahren gilt es nun die in Abschnitt 5.3 beschriebenen Optimierungsmöglichkeiten genauer zu untersuchen, so z.B. die Reduktion der zu einem bestimmten Zeitpunkt in Betracht zu ziehenden Modelle auf die wahrscheinlichsten. Ebenso ist es sinnvoll, das vorgestellte Verfahren um einen fixed-Lag Smoother für den Interacting Multiple Modell Algorithmus zu erweitern. Außerdem gilt es noch den vorgestellten Algorithmus mittels Modellen, die durch das in Kap. 3 beschriebene Klanganalyseverfahren identifiziert wurden, auf echten Klängen zu evaluieren.

Desweiteren wäre es eine Untersuchung wert, weitere Informationen für die Trennung mit dem hier beschriebenen Verfahren zu berücksichtigen, wie z.B. die Tatsache, dass in westlicher Musik die Onsets der Töne normalerweise einem, durch das Tempo vorgegebenen, rhythmischen Raster genügen. Ebenso sind die Übergangswahrscheinlichkeiten zwischen den Tönen nicht

für alle Töne gleich, sondern durch die Tonart bestimmt, wodurch bestimmte Töne mit einer größeren Wahrscheinlichkeit auf gegebene Töne folgen. Außerdem ist durch die Harmonik auch die Wahrscheinlichkeit bestimmter Kombinationen von Tönen höher als die anderer Kombinationen.

Außerdem wurde in den Experimenten noch keine Stereoinformation ausgenutzt, die durchaus zu einer verbesserten Schätzung beitragen könnte. Man kann sich auch vorstellen, dass die aus dem Auftreten eines Klanges gewonnenen Informationen für die Schätzung bei einem erneuten Auftreten desselben Klanges ebenfalls hilfreich sein können.

KAPITEL 7

Danksagung

Ich bedanke mich bei Prof. Dr.-Ing. Uwe D. Hanebeck für die Möglichkeit dieses Thema zu bearbeiten, sowie für seine Ideen und Richtungsweisungen.

Desweiteren danke ich meinem Betreuer Dipl.-Ing. Frederik Beutler für seine Hilfestellungen, mich mit der Systemtheorie und Modellierung vertraut zu machen, sowie die zahlreichen Diskussionen über Möglichkeiten der Umsetzung dieser Arbeit und die Zeit, mir auch so manche Frage zu beantworten.

Auch Daniel Gärtner, Uwe Mayer, Sebastian Biemüller, Gerrit Holzbach und Guido Ciburski gilt mein Dank, die mir auch so manchen Ratschlag gegeben und Frage beantwortet haben.

Außerdem danke ich meiner Familie, vor allem meiner Frau und meinen Eltern für den Rückhalt in der Zeit der Bearbeitung dieser Arbeit und meinem Sohn für sein Lächeln, das er mir jeden Tag aufs neue schenkt.

Nicht zuletzt bedanke ich mich bei meinem Hauskreis und meinen Freunden für ihre Gebete, sowie meinem Hauskreisleiter Michael, der mir immer wieder ermutigende Worte zugesprochen hat, gerade auch wenn es nicht so gut lief.

Es gilt auch allen denen mein Dank, die mich bei dieser Arbeit in jeglicher Form unterstützt haben, sei es durch Gebet, Zeit oder Ermutigung, auch wenn sie hier nicht explizit erwähnt sind.

Schließlich danke ich meinem Herrn Jesus Christus für seine Führung und Gegenwart.

Literaturverzeichnis

[1] Abel, Jonathan S. und Julius Orion Smith III: *Robust Design of Very High-Order Allpass Dispersion Filters.* In: *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Seiten 13–18, Montreal, Quebec, Canada, September 18-20 2006.

[2] Allen, Robert C.: *Soundfont of an Ovation acoustic guitar.* `http://www.hammersound.com/cgi-bin/soundlink.pl?action=view_download_page;ID=592`, November 20 2000.

[3] Bank, Balász und Vesa Välimäki: *Robust loss filter design for digital waveguide synthesis of string tones.* IEEE Signal Processing Letters, 10(1):18–20, Januar 2003.

[4] Bank, Balázs, Federico Avanzini, Gianpaolo Borin, Giovanni De Poli, Federico Fontana und Davide Rocchesso: *Physically Informed Signal Processing Methods for Piano Sound Synthesis: A Research Overview.* EURASIP Journal on Applied Signal Processing, 2003(10):941, 2003.

[5] Bar-Shalom, Y. und T.E. Fortmann: *Tracking and Data Association.* Academic Press, New York, FL, 1988.

[6] Benning, Manjinder Singh: *Kalman Filtering for Computer Music Applications.* Diplomarbeit, University of Victoria, 2007.

[7] BLom, Henk A. P.: *An efficient filter for abruptly changing systems.* The 23rd IEEE Conference on Decision and Control, 23:656–658, Dezember 1984.

[8] Blom, Henk A.P. und Yaakov Bar-Shalom: *The interacting multiple model algorithm for systems with Markovian switching coefficients.* IEEE Transactions on Automatic Control, 33(8):780–783, 1988.

[9] Casey, Michael A.: *Separation of Mixed Audio Sources by Independent Subspace Analysis.* In: *Proceedings of the International Computer Music Conference*, Berlin, August 2000.

[10] Cemgil, Ali Taylan: *Bayesian Music Transcription.* Doktorarbeit, Radboud University Nijmegen, Nijmegen, the Netherlands, 2004.

[11] Chen, Bing und J. K. Tugnait: *An interacting multiple model fixed-lag smoothing algorithm for Markovian switching systems.* In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat No 98CH36171) CDC-98*, Band 1, Seiten 269–274, 1998.

[12] Davy, Manuel und Simon J. Godsill: *Bayesian Harmonic Models for musical pitch estimation and analysis.* Technischer Bericht CUED/F-INFENG/TR.431, Signal Processing Group, Cambridge University Engineering Department, Trumpington Street, Cambridgt CB2 1PZ, UK, November 13 2002.

[13] Davy, Manuel, Simon J. Godsill und Jérôme Idier: *Bayesian Analysis of Western Tonal Music.* Journal of the Acoustical Society of America (JASA), 119(4):2498–2517, April 2006.

[14] Ding, Yinong und Xiaoshu Qian: *Sinusoidal and Residual Decomposition and Residual Modeling of Musical Tones Using the QUASAR Signal Model.* In: *Proceedings of the ICMC 97*, Seiten 35–42. University of Michigan, Ann Arbor, 1997.

[15] Dubois, Corentin und Manuel Davy: *Joint Detection and Tracking of Time-Varying Harmonic Components: a Flexible Bayesian Approach.* IEEE transactions on Speech, Audio and Language Processing, 15(4):1283–1295, Mai 2007.

[16] Erkut, Cumhur: *Loop Filter Design Techniques For Virtual String Instruments.* In: *Proceedings of the International Symposium on Musical Acoustics (ISMA 2001)*, Band 1, Seiten 259–262, Perugia, Italy, September 10-14 2001.

[17] Gärtner, Daniel: *Der Mensch, Einschwingvorgänge und Neuronale Netze*, März 25 2007.

[18] Godsill, S.J. und C.H. Tan: *Removal of low frequency transient noise from old recordings using model-based signal separation techniques.* In: *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics ASPAA-97*, Seiten 4–7, 1997.

[19] Hall, Donald E.: *Musical Acoustics.* Brooks Cole, 3. Auflage, 2002.

[20] Hanebeck, Uwe D. und Oliver Schrempf: *Stochastische Informationsverarbeitung*, Wintersemenster 2007/2008. Vorlesungsskript.

[21] Hyvärinen, Aapo und Erkki Oja: *Independent component analysis: algorithms and applications.* Neural Networks, 13(4-5):411–430, 2000.

[22] Jensen, Kristoffer: *Timbre Models of Musical Sounds.* Doktorarbeit, Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 København Ø, 2000.

[23] Julier, Simon, Jeffrey Uhlmann und Hugh F. Durrant-Whyte: *A new method for the nonlinear transformation of means and covariances in filters and estimators.* IEEE Transactions on Automatic Control, 45(3):477, 2000.

[24] Julier, Simon J. und Jeffrey K. Uhlmann: *A New Extension of the Kalman Filter to Nonlinear Systems.* In: *SPIE AeroSense Symposium*, Orlando, FL, 1997.

[25] Karjalainen, Matti, Vesa Välimäki und Zoltán Jánosy: *Towards High-Quality Sound Synthesis of the Guitar and String Instruments.* International Computer Music Conference (ICMC93), September 10-15 1993.

[26] Karjalainen, Matti, Vesa Välimäki und Tero Telonen: *Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond.* Computer Music Journal, 22(3):17–32, 1998.

[27] Laakso, T.I., V. Valimaki, M. Karjalainen und U.K. Laine: *Splitting the unit delay [FIR/all pass filters design].* IEEE Signal Processing Magazine, 13(1):30–60, 1996.

[28] Lang, Markus und Timo I. Laakso: *Simple and robust method for the design of allpass filters using least-squares phase error criterion.* IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 41(1):40–48, Januar 1994.

[29] Laroche, J. und J.M. Jot: *Analysis/synthesis of quasi-harmonic sounds by use of the Karplus-Strong algorithm.* Le Journal de Physique IV, 2(c1):117–120, April 1992.

[30] Lerro, D. und Y. Bar-Shalom: *Interacting multiple model tracking with target amplitude feature.* IEEE Transactions on Aerospace and Electronic Systems, 29(2):494–509, 1993.

[31] Li, X. R. und Y. Bar-Shalom: *Mode-Set Adaptation In Multiple-Model Estimators For Hybrid Systems.* In: *Proc. 1992 American Control Conference*, Seiten 1794–1799, Chicago, IL, 1991.

[32] Martins, Luis Gustavo, Juan José Burred, George Tzanetakis und Mathieu Lagrange: *Polyphonic Instrument Recognition Using Spectral Clustering.* In: *Proceedings of the 8th International Conference on Music Information Retrieval*, Seiten 213–218, Vienna, Austria, September 23-27 2007. Österreichische Computer Gesellschaft.

[33] Masri, Paul: *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals.* Doktorarbeit, Faculty of Engineering, Department of Electrical and Electronic Engineering, University of Bristol, UK, Dezember 1997.

[34] Master, Aaron S.: *Bayesian two source modeling for separation of N sources from stereo signals.* Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, 4:iv–281 – iv–284 vol.4, Mai 17–21 2004.

[35] Master, Aaron S.: *Stereo Music Source Separation via Bayesian Modeling.* Doktorarbeit, Stanford University, Juni 2006.

[36] Mazor, E., A. Averbuch, Y. Bar-Shalom und J. Dayan: *Interacting Multiple Model Methods in Multitarget-Multisensor Tracking: Survey.* IEEE Trans. on Aerospace and Electronic Systems, 34(1):103–123, Januar 1998.

[37] Morelande, Mark R.: *An Unscented Transformation for Conditionally Linear Models.* In: *2007 IEEE International Conference on Acoustics Speech and Signal Processing - ICASSP 07*, Seiten III–1417 – III–1420, 2007.

[38] Musicki, Darko: *Limits of Linear Multitarget Tracking.* In: *2005 8th International Conference on Information Fusion*, Band 1, Seiten 205–210, Philadelphia, PA, Juli 25-28 2005.

[39] Musicki, Darko und Sofia Suvorova: *Tracking in Clutter using IMM-IPDA-Based Algorithms*. IEEE Transactions on Aerospace and Electronic Systems, 44(1):111–126, Januar 2008.

[40] Musicki, Darko, Sofia Suvorova und Subhash Challa: *Multi Target Tracking of Ground Targets in Clutter with LMIPDA-IMM*. In: *7th International Conference on Infomation Fusion*, Seiten 1104–1110, Stockholm, Sweden, Juni 2004.

[41] Nakatani, Tomohiro, Hiroshi G. Okuno und Takeshi Kawabata: *Residue-Driven Architecture for Computational Auditory Scene Analysis*. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Band 1, Seiten 165–174, Montreal, Quebec, Canada, August 20-25 1995.

[42] Parra, Lucas und Uday Jain: *Approximate Kalman filtering for the harmonic plus noise model*. In: *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat No 01TH8575) ASPAA-01*, Seiten 75–78, New Paltz, NY, USA, 2001.

[43] Pedersen, Michael Syskind, DeLiang Wang, Jan Larsen und Ulrik Kjems: *Two-Microphone Separation of Speech Mixtures*. IEEE Transactions on Neural Networks, 19(3):475–492, 2008.

[44] Rauhala, Jukka und Vesa Välimäki: *Dispersion Modeling In Waveguide Piano Synthesis Using Tunable Allpass Filters*. In: *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, Seiten 71–76, Montreal, Canada, September 18-20 2006.

[45] Rauhala, Jukka und Vesa Välimäki: *Tunable dispersion filter design for piano synthesis*. IEEE Signal Processing Letters, 13(5):253–256, 2006.

[46] Rickard, Scott, Conor Fearon, Radu Balan und Justinian Rosca: *MINUET: Musical Interference Unmixing Estimation Technique*. In: *Proceedings of the 39th Annual Conference on Conference on Information Sciences and Systems (CISS2004)*, Princeton, NJ, März 2004.

[47] Särkkä, Simo und Jouni Hartikainen: *EKF/UKF Toolbox for Matlab V1.2*. `http://www.lce.hut.fi/research/mm/ekfukf/`, September 4 2007.

[48] Schüssler, Hans Wilhelm: *Digitale Signalverarbeitung - Analyse diskreter Signale und Systeme*, Band 1. Springer-Verlag, 1988.

[49] Serra, Xavier: *Musical Sound Modeling with Sinusoids plus Noise*. Musical Signal Processing, 1997.

[50] Simon, Dan und T. Chia: *Kalman Filtering with State Equality Constraints*. IEEE Transactions on Aerospace and Electronic Systems, 39:128–136, Januar 2002.

[51] Simon, Dan und Donald L. Simon: *Kalman Filtering with Inequality Constraints for Turbofan Engine Health Estimation*. submitted for publication, 2002.

[52] Smith III, Julius Orion: *Techniques for Digital Filter Design and System Identification with Application to the Violin.* Doktorarbeit, Elec. Engineering Dept., Stanford University (CCRMA), Juni 1983.

[53] Smith III, Julius Orion: *Virtual electric guitars and effects using faust and octave.* In: *Proceedings of the 6th International Linux Audio Conference (LAC2008)*, 2008.

[54] Smith III, Julius Orion: *Physical Audio Signal Processing, May 2008 Edition.* http://ccrma.stanford.edu/~jos/pasp/, accessed 2008-08-30. online book.

[55] Smith III, Julius Orion: *Making Virtual Electric Guitars and Associated Effects Using Faust.* http://ccrma.stanford.edu/realsimple/faust_strings/, accessed 2008-09-13. online book.

[56] Smith III, Julius Orion: *Spectral Audio Signal Processing, October 2008 Draft.* http://ccrma.stanford.edu/~jos/sasp/, accessed 2008-09-29. online book.

[57] Smith III, Julius Orion: *Introduction to Digital Filters with Audio Applications.* http://ccrma.stanford.edu/~jos/filters/, accessed 2008-10-03. online book.

[58] Smith III, Julius Orion und Xavier Serra: *PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation.* In: *Proceedings of the International Computer Music Conference (ICMC-87)*, Tokyo, 1987.

[59] Su, Alvin W. Y. und Sheng-Fu Liang: *A new automatic IIR analysis/synthesis technique for plucked-string instruments.* IEEE Transactions on Speech and Audio Processing, 9(7):747–754, 2001.

[60] Tolonen, Tero und Vesa Välimäki: *Automated Parameter Extraction for Plucked String Synthesis.* In: *Proceedings of the 1997 International Symposium on Musical Acoustics (ISMA'97)*, Seiten 245–250, Edinburgh, Scotland, August 19-22 1997.

[61] Verma, Tony S., Scott N. Levine und Teresa H.Y. Meng: *Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals.* In: *Proceedings of the ICMC*, 1997.

[62] Wikipedia: *ADSR — Wikipedia, Die freie Enzyklopädie.* http://de.wikipedia.org/w/index.php?title=ADSR&oldid=52112497, 2008. [Online; Stand 28. Oktober 2008].

[63] Wikipedia: *Fensterfunktion — Wikipedia, Die freie Enzyklopädie.* http://de.wikipedia.org/w/index.php?title=Fensterfunktion&oldid=51581933, 2008. [Online; Stand 29. Oktober 2008].

[64] Wikipedia: *Gitarre — Wikipedia, Die freie Enzyklopädie.* http://de.wikipedia.org/w/index.php?title=Gitarre&oldid=51436896, 2008. [Online; Stand 7. Oktober 2008].

[65] Wikipedia: *Kalman-Filter — Wikipedia, Die freie Enzyklopädie.* http://de.wikipedia.org/w/index.php?title=Kalman-Filter&oldid=49984359, 2008. [Online; Stand 16. September 2008].

[66] Wikipedia: *Kalman filter — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Kalman_filter&oldid=238430838, 2008. [Online; accessed 16-September-2008].

[67] Wikipedia: *Maskierungseffekt — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Maskierungseffekt&oldid=51147024, 2008. [Online; Stand 12. Oktober 2008].

[68] Wikipedia: *Moden — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Moden&oldid=51540691, 2008. [Online; Stand 9. Oktober 2008].

[69] Wikipedia: *Monokompatibilität — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/w/index.php?title=Monokompatibilit%C3%A4t&oldid=42596711, 2008. [Online; Stand 14. September 2008].

[70] Wikipedia: *Window function — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Window_function&oldid=247207953, 2008. [Online; accessed 29-October-2008].

[71] Yilmaz, Özgür und Scott Rickard: *Blind Separation of Speech Mixtures via Time-Frequency Masking*. IEEE Transactions on Signal Processing, 52(7):1830–1847, 2004.