

Technische Universität Berlin

Repräsentation und Modifikation von spektralen
Hüllkurven natürlicher Klänge auf der Basis von
Formantmodellen

Magisterarbeit im Studiengang Kommunikationswissenschaft
vorgelegt von:

Robin Schmidt (Matr.-Nr. 194594)

Erstgutachter: Prof. Dr. Stefan Weinzierl

Zweitgutachter: Prof. Dr. Axel Röbel

Berlin, den 07.11.2006

INHALTSVERZEICHNIS

1. Spektrale Hüllkurven	2
1.1 Das Problem	2
1.1.1 Quelle-Filter Modellierung	3
1.1.2 Pre-Emphase Filterung	4
1.2 Autoregressive Modellierung des Filters	5
1.2.1 Schätzung des Filters über Lineare Prädiktion	6
1.2.2 Schätzung des Filters über Discrete All-Pole Modeling	9
1.3 Cepstrum-basierte Schätzung der Hüllkurve	10
1.3.1 Schätzung der Hüllkurve über Cepstral Smoothing	10
1.3.2 Schätzung der Hüllkurve mit dem True Envelope Algorithmus	13
1.4 Umrechnung der Koeffizienten und TE-LPC	20
1.5 Anwendungen und Grenzen	21
2. Formantmodelle	24
2.1 Serielle Allpolfilter	24
2.1.1 Das Modell	24
2.1.2 Polpaare und Formanten	29
2.1.3 Frequenzselektive Lineare Prädiktion und Unterabtastung	31
2.1.4 Transformation der Formanten	33
2.1.5 Implementation und Praxistest	35
2.1.6 Vor- und Nachteile des Modells	38
2.2 Parallele Allpolstruktur	40
2.2.1 Das Modell	40

2.2.2	Zusammenhang zu Formantwellenformen (FOF's)	43
2.2.3	Transformation der Formanten	46
2.2.4	Implementation und Praxistest	48
2.2.5	Vor- und Nachteile des Modells	53
2.3	Serielle Parametrische Equalizer	54
2.3.1	Das Modell	54
2.3.2	Die Kostenfunktion und ihr Gradient	59
2.3.3	Die frequenzabhängige Gewichtung des Fehlers	60
2.3.4	Initialisierung der Equalizer-Parameter	62
2.3.5	Transformation der Formanten	63
2.3.6	Implementation und Praxistest	64
2.3.7	Vor- und Nachteile des Modells	69
2.4	Anwendungen und Ausblick	69
<i>Anhang</i>		71
<i>A. Gradientenbasierte Minimierung einer Kostenfunktion</i>		72
<i>B. Herleitungen</i>		82
B.1	Impulsantwort eines Zweipol-Filters mit einer Nullstelle	82
B.2	Betragsfrequenzgang einer Biquad-Reihenschaltung	84
B.3	Ableitungen der Kostenfunktion nach den Biquad-Koeffizienten	86
B.4	Ableitungen der Biquad-Koeffizienten nach den Equalizer-Parametern	89
B.5	Ableitungen der Kostenfunktion nach den Equalizer-Parametern	90
<i>C. MatLab-Implementation</i>		92

Im diesem Text wird die folgende Notation benutzt:

Variable Bedeutung

A	Amplitude einer Formantwellenform (FOF)
$A(z)$	Transferfunktion eines Prediction-Error Filters
bw_{Hz}	Bandbreite eines Filters in Hz
bw_{oct}	Bandbreite eines Filters in Oktaven
$C[r]$	Wert eines diskreten Cepstrums bei der quefreny r
c	eine Konstante, meist vom Benutzer wählbar
$D[r]$	Wert eines Differenz-Cepstrums im True-Envelope Algorithmus
\mathbf{d}	Suchrichtung in einem Optimierungsverfahren
E	Fehler- bzw. Kostenfunktion, an einigen Stellen auch Energie
G	globaler Gain-Faktor für eine Filterstruktur
\mathbf{g}	Gradient einer Funktion
g	Gain-Faktor eines Filters oder Gain-Parameter einer Equalizer-Stufe
g_{dB}	Gain-Parameter eines Peak-Filters in dB
\mathbf{H}	Hesse-Matrix einer Funktion
$H(z)$	Transferfunktion eines Filters (allgemein)
$H_s(z)$	Transferfunktion des s -ten Teilfilters in einer Zusammenschaltung von S gleichartigen Filtern (seriell oder parallel)
P	Ordnung eines AR-Modells oder Radius eines Polstellenpaares
P_c	Ordnung eines Cepstrums
R	Radius einer komplexen Zahl r , die als Zähler in einem Partialbruch steht
S	Anzahl von einzelnen Filterstufen in einer Zusammenschaltung von Teilfiltern (seriell oder parallel)
$w(\Omega_k)$	Gewichtung für den Fehler bei einer normierten Kreisfrequenz Ω_k
$X[k]$	Wert eines diskreten Spektrums am Bin mit dem Index k
α	Dämpfungs-konstante einer Formantwellenform (FOF) oder Lernschrittweite in einem Optimierungsverfahren
β	Gewichtung für die alte Suchrichtung beim (Scaled) Conjugate Gradient Verfahren
γ	Bandbreitenparameter einer Equalizer-Stufe
λ	Lernschrittweite im True Envelope Algorithmus oder Skalierungsparameter beim Scaled Conjugate Gradient Verfahren.
$\boldsymbol{\theta}$	ein Parametervektor
Ω	normierte Kreisfrequenz (allgemein)
Ω_c	normierte Mittenfrequenz einer Equalizer-Stufe
$\omega, \varphi, \phi, \psi$	werden als (Phasen)Winkel für komplexe Zahlen in Polardarstellung benutzt, ω gelegentlich auch allgemein für Kreisfrequenzen

ZUSAMMENFASSUNG

Gegenstand dieser Arbeit ist die Modellierung von Klängen mit Hilfe von ausgewählten digitalen Filterstrukturen. Dabei ist es das Ziel, eine intuitiv verständliche Parametrisierung der spektralen Hüllkurve aus dem Signal zu gewinnen, d.h. aus den charakteristischen Eigenschaften der spektralen Hüllkurve soll ein Satz von Parametern errechnet werden, der dann dem Benutzer präsentiert und zur freien Editierung zu Verfügung gestellt werden kann. Die einzelnen Parameter sollen dabei jeweils eine intuitive Interpretation zulassen, so dass ein Benutzer des jeweiligen Signalmodells in der Lage ist, gezielt musikalisch sinnvolle Transformationen der Parameter durchzuführen. Ein besonderer Schwerpunkt liegt dabei auf der Modellierung der Formanten des Signals, da diese perzeptuell besonders wichtig sind.

Im ersten Kapitel werden verschiedene Techniken zur Schätzung der spektralen Hüllkurve betrachtet, im zweiten Kapitel werden dann 3 verschiedene Filterstrukturen betrachtet, deren Frequenzgang als Modell dieser spektralen Hüllkurve dienen soll. Dabei soll untersucht und verglichen werden, inwieweit die Parameter der jeweiligen Filterstruktur geeignet sind, um Formanten zu beschreiben, welche Möglichkeiten zur Transformation der spektralen Hüllkurve diese Parameter ermöglichen und welche Artefakte von solchen Transformationen zu erwarten sind. Die Filterstrukturen und Algorithmen zur Parameterbestimmung wurden in der Sprache MatLab implementiert und anhand eines Beispielsignals getestet. Die in der Arbeit dargestellten Ergebnisse können anhand dieser Implementation nachvollzogen werden (siehe dazu Anhang C) und es können die entsprechenden Algorithmen auf eigene Signale angewandt werden.

1. SPEKTRALE HÜLLKURVEN

1.1 Das Problem

Viele Spektren von natürlichen Klängen enthalten zweierlei Arten von spektraler Information die einander überlagern. Einerseits sieht man im Spektrum von z.B. harmonischen Klängen die Spektrallinien der einzelnen auftretenden Frequenzen. Zwischen diesen diskreten Frequenzen ist das Amplitudenspektrum im Prinzip gleich Null (das ist natürlich eine stark idealisierte Vorstellung). Andererseits hat jede solche Spektrallinie eine bestimmte Amplitude - verbindet man die spektralen Peaks auf geeignete Weise miteinander, so erhält man eine spektrale Hüllkurve, die die grundsätzliche Form des Spektrums beschreibt ohne jedoch einzelne Frequenzen aufzulösen. Am Beispiel des Spektrums einer männlichen Stimme, die ein langgezogenes 'a' spricht, wird dies deutlich (siehe Abb. 1.1).

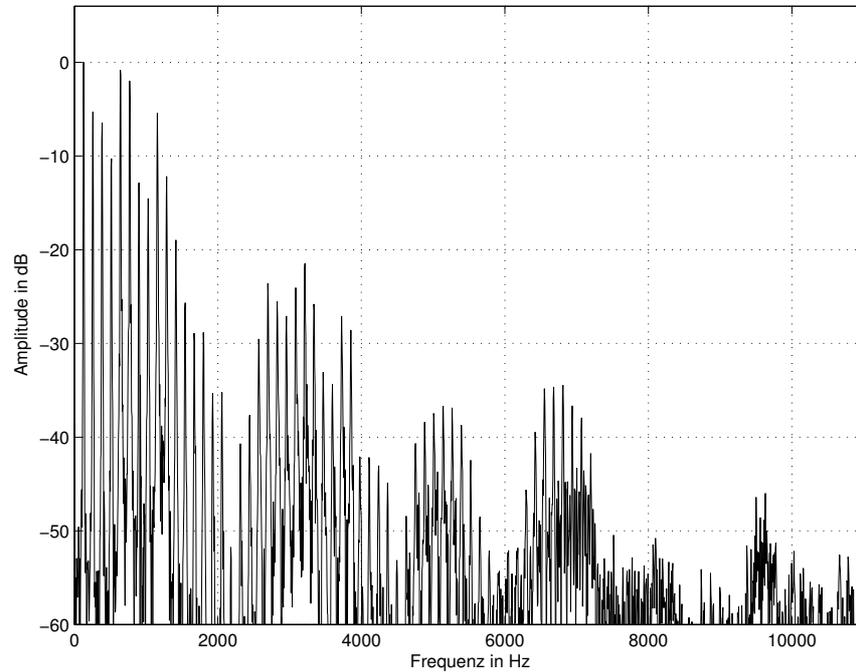


Fig. 1.1: FFT-Spektrum eines männlichen 'Ahh'

Dieses Signal wird im weiteren Verlauf der Arbeit immer wieder als Beispielsignal herangezogen werden. Die Sample-Rate des Signals beträgt 44.1 kHz, so dass man das Spektrum eigentlich bis 22.05 kHz darstellen könnte. Zur besseren Erkennbarkeit jedoch, wird

nur der Bereich bis zu 11.025 kHz dargestellt. Wir sehen einerseits die (mehr oder weniger) diskreten Spektrallinien bei ganzzahligen Vielfachen der Grundfrequenz (die bei ca. 130 Hz liegt), andererseits sehen wir, dass z.B. die Frequenzbereiche um 700 Hz und 1200 Hz mehr Energie enthalten als umliegende Frequenzbereiche. Solche Frequenzbereiche mit besonders viel Energie nennt man Formanten. Diese entstehen physikalisch häufig durch Resonatoren, deren Resonanzfrequenzen oft unabhängig von der Grundfrequenz des Klangs sind. Bei akustischen Saiteninstrumenten beispielsweise werden diese Resonanzfrequenzen vor allem durch Form und Größe des Korpus bestimmt - und diese Dinge sind schließlich völlig unabhängig davon, welche Note gerade gespielt wird. Bei der menschlichen Stimme sind die Resonatoren durch die Geometrie der Hohlräume im Mund- und Rachenbereich bestimmt, während das Anregungssignal bei stimmhaften Lauten durch das periodische Öffnen und Schließen der Glottis entsteht. Hier kann es zwar durchaus zu gewissen Wechselwirkungen zwischen Grundfrequenz und Formantbereichen kommen, näherungsweise kann man die Trennung jedoch auch hier vollziehen. Bei der menschlichen Stimme sind die Lagen der Formantbereiche ein wichtiges Kriterium zur Unterscheidung von Stimmen und auch zur Unterscheidung verschiedener Laute. Für viele interessante Signaltransformationen ist es wünschenswert, die spektrale Hüllkurve von den (durch die Periodizität des Signals gegebenen) spektralen Peaks zu trennen, denn wenn man diese beiden Informationen getrennt zu Verfügung hat, kann man sie auch getrennt voneinander modifizieren. Anwendungsbeispiele sind Pitch-Shifting unter Erhaltung der spektralen Hüllkurve und Verschiebung bzw. Skalierung der spektralen Hüllkurve unter Erhaltung der Tonhöhe.

1.1.1 Quelle-Filter Modellierung

Eine Möglichkeit zur Schätzung der spektralen Hüllkurve besteht darin, sich das Signal $x[n]$ als Faltung zwischen einem weißen Anregungssignal $e[n]$ und einem (Synthese-)Filter mit der Impulsantwort $h[n]$ vorzustellen:

$$x[n] = e[n] * h[n]$$

und das Filter in geeigneter Weise zu modellieren. Ein solches Filter lässt sich im zeitdiskreten Bereich im allgemeinen Fall durch eine lineare Differenzgleichung der Form:

$$x[n] = \sum_{p=0}^P b_p e[n-p] - \sum_{q=1}^Q a_q x[n-q]$$

realisieren. Für unser Signal $x[n]$ wird also angenommen, dass es sich als Linearkombination des Anregungssignals $e[n]$, vergangenen Werten des Anregungssignals $e[n-p]$, $p = 1, \dots, P$ und vergangenen Werten des Signals selbst $x[n-q]$, $q = 1, \dots, Q$ beschreiben lässt. Ein solches allgemeines Filter ist als Blockschaltbild in Abbildung 1.2 dargestellt.

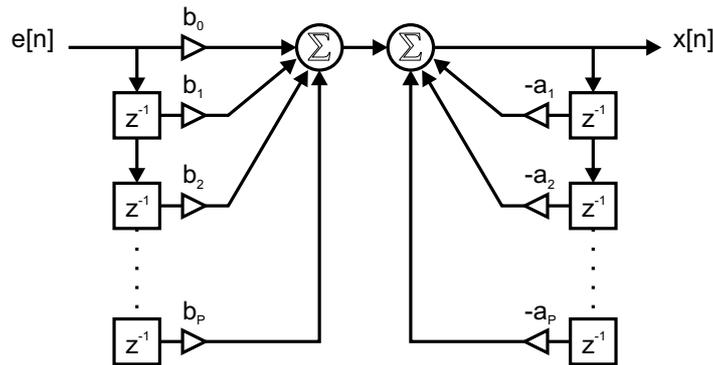


Fig. 1.2: allgemeines Filter der Ordnung P in Direktform

Die Annahme, dass das Anregungssignal weiß sein soll, bedeutet, dass alle Frequenzen die gleiche Amplitude haben und das Spektrum somit einfach eine Konstante ist. Streng genommen ist dies nur bei weißem Rauschen der Fall. Da wir es aber in vielen Fällen mit periodischen Signalen zu tun haben, wollen wir den Begriff 'weiß' in dieser Arbeit etwas allgemeiner fassen und verlangen, dass alle im Signal vorkommenden Frequenzen die gleiche Amplitude haben sollen (und nicht generell alle Frequenzen). Bei periodischen Signalen hat das Spektrum dann die Form eines Delta-Kamms, das zugehörige Zeitsignal wäre in diesem Fall ebenfalls ein Delta- bzw. Impulskamm. Wenn wir das Synthesefilter in geeigneter Weise modellieren, dann können wir die Übertragungsfunktion dieses Modells als Schätzung für die gesuchte spektrale Hüllkurve ansehen. Im Folgenden soll die Transferfunktion unseres Synthesefilters mit $H(z)$ und das Modell (d.h. die Schätzung) des Synthesefilters mit $\hat{H}(z)$ bezeichnet werden. Wie wir dieses Filter modellieren bzw. schätzen können, ist Gegenstand der Abschnitte 1.2.1 und 1.2.2.

1.1.2 Pre-Emphasis Filterung

Wie bereits erwähnt, wird bei der Quelle-Filter-Modellierung von einem weißen Anregungssignal ausgegangen. In der Realität ist es jedoch in vielen Fällen so, dass auch das Anregungssignal schon eine gewisse Tiefpass-Charakteristik hat. In der spektralen Hüllkurve des Signals zeigt sich dies oft als eine Art linearer Abwärtstrend (als grobe Näherung). Bei der menschlichen Stimme beispielsweise weist das glottale Anregungssignal einen Abfall von ca. $-12\text{dB}/\text{Okt}$ auf, die Abstrahlung des Schalls vom Mund erfolgt jedoch mit einer Hochpass-Charakteristik mit ca. $+6\text{dB}/\text{Okt}$, so dass also insgesamt ein $-6\text{dB}/\text{Okt}$ Abfall übrig bleibt, der nichts mit irgendwelchen Formant-Frequenzen zu tun hat. Im Hinblick auf die Modellierung von Formanten erscheint es daher sinnvoll, diese Tiefpass-Charakteristik zuerst einmal herauszunehmen. Dies wird durch ein so genanntes Pre-Emphasis Filter erreicht. In der Praxis verwendet man hier oft ein einfaches nichtrekursives Filter erster Ordnung der Form:

$$y[n] = x[n] + b_1 x[n-1]$$

wobei der Koeffizient b_1 nahe -1 gewählt wird - bei $b_1 = -1$ handelt es sich bei dem Pre-Emphasis Filter um ein einfaches Differenzen-Filter. Für die spätere Resynthese sollte

das Pre-Emphase Filter invertierbar sein, daher sollte der Koeffizient vom Betrag her etwas kleiner als 1 (Größenordnung: -0.99) gewählt werden (bei $b_0 = -1$ haben wir eine Nullstelle auf dem Einheitskreis, das inverse Filter müsste somit eine Polstelle auf dem Einheitskreis haben und wäre damit nur noch grenzstabil).

Optimale Pre-Emphase Da die Wahl irgendeines bestimmten Wertes für b_1 jedoch ziemlich willkürlich erscheint und die bestmögliche Wahl von Signal zu Signal unterschiedlich ist, bietet es sich an, auch dieses b_1 aus dem Signal selbst zu gewinnen. Dies kann sehr einfach realisiert werden, indem man einen linearen Prädiktor (siehe Abschnitt 1.2.1) erster Ordnung bestimmt. Dieser ergibt sich einfach als r_1/r_0 wobei r_1 und r_0 die Autokorrelationskoeffizienten für $lag = 0$ und $lag = 1$ sind. Das entsprechende Prediction-Error Filter wird dann als Pre-Emphase Filter benutzt. Dieser Ansatz der optimalen Pre-Emphase wird auch innerhalb dieser Arbeit verfolgt. Messungen an realen Signalen lieferten Koeffizienten im Bereich zwischen ca. -0.9 und -0.999 . Dies ist ein ziemlich großer Bereich - der Abstand vom Einheitskreis überspannt zwei Zehnerpotenzen. Die Wirkung der Pre-Emphase Filterung auf das Signalspektrum wird deutlich beim Vergleich von Abb. 1.3 und Abb. 1.1.

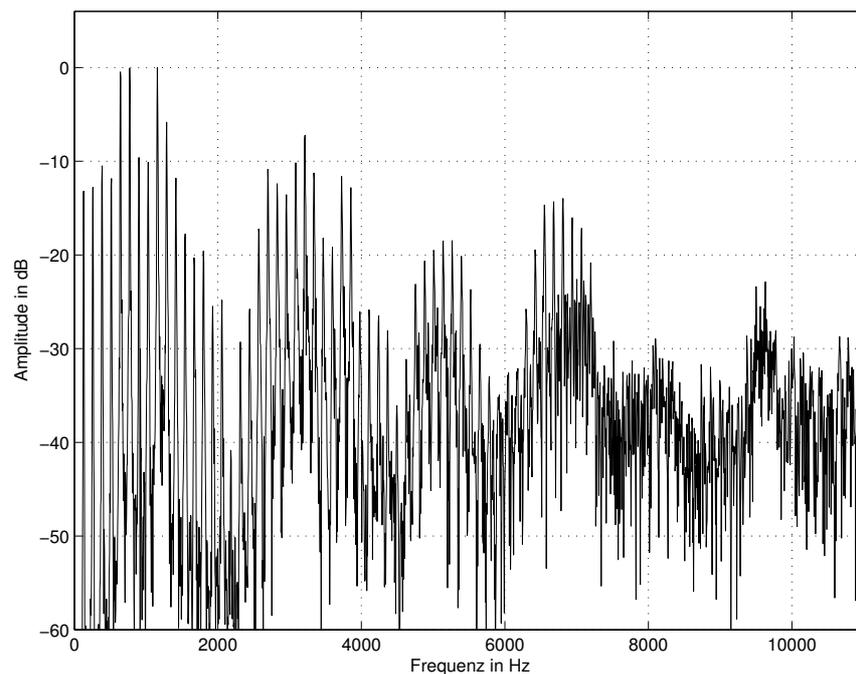


Fig. 1.3: FFT-Spektrum eines männlichen 'Ah' mit optimaler Pre-Emphase

1.2 Autoregressive Modellierung des Filters

Bei der autoregressiven Modellierung des Synthesefilters macht man die vereinfachende Annahme, dass dieses Filter ausschließlich Polstellen (und keine Nullstellen) besitzt - man spricht daher von einem Allpolfilter. Da Polstellen mit spektralen Peaks einhergehen, ist

dies für die Modellierung von Formanten sicherlich ein plausibles Modell. Ein solches Allpolfilter kann Zeitbereich durch ein rein rekursives IIR-Filter realisiert werden, ein Filter also, dass nur das aktuelle Eingangs-Sample und vergangene Ausgangs-Samples zur Berechnung des aktuellen Ausgangs-Samples verwendet. Die Feedforward-Koeffizienten des Filters sind also alle (bis auf b_0) gleich Null. Man modelliert das Signal $x[n]$ also durch:

$$x[n] = e[n] - \sum_{p=1}^P a_p x[n-p]$$

wobei hier $b_0 = 1$ angenommen wurde - dies ist keine wirkliche Einschränkung, da der b_0 -Koeffizient bei Allpol-Filtern einfach als Verstärkungsfaktor am Eingang des Filters angesehen werden kann, und somit genauso gut als Lautstärke des Anregungssignals interpretiert werden kann. Das Blockschaltbild zu dieser vereinfachten Differenzgleichung ist in Abbildung 1.4 gegeben.

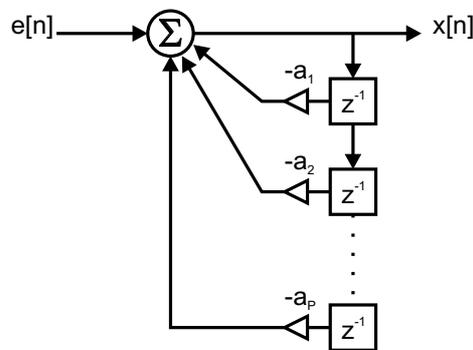


Fig. 1.4: Allpol-Filter der Ordnung P in Direktform

Die Aufgabe besteht nun darin, die Filterkoeffizienten a_p aus dem Signal $x[n]$ in der Art und Weise zu schätzen, dass die Übertragungsfunktion des Allpolfilters möglichst gut mit dem Spektrum des Signals übereinstimmt. Die etwas vage Ausdrucksweise 'möglichst gut' muss dazu natürlich erst einmal mathematisch gefasst werden. In der Regel stellt man dazu zuerst eine geeignete Kostenfunktion auf und minimiert diese dann bezüglich der gesuchten Parameter. Zwei Verfahren zur Schätzung der Filterkoeffizienten werden im Folgenden kurz dargestellt.

1.2.1 Schätzung des Filters über Lineare Prädiktion

Wie die Bezeichnung 'Lineare Prädiktion' bereits andeutet, spielt bei diesem Verfahren eine Vorhersage des Signals eine Rolle. Dabei ergibt sich das vorhergesagte Signal (welches mit $\hat{x}[n]$ bezeichnet werden soll) als Linearkombination aus vergangenen Werten des Signals selbst. Die Vorhersagevorschrift lautet demnach:

$$\hat{x}[n] = \sum_{p=1}^P w_p x[n-p]$$

Der vorhergesagte Wert ist also einfach eine gewichtete Summe von vergangenen Signalwerten mit w_p als Gewichtungsfaktor für den p Samples zurückliegenden Signalwert. Diese Vorhersagevorschrift stellt übrigens ein FIR-Filter mit den Koeffizienten $0, w_1, \dots, w_P$ dar. Dieses Filter wird als Prädiktorfilter oder einfach als Prädiktor bezeichnet und die Transferfunktion dieses Filters wird üblicherweise mit $P(z)$ bezeichnet. Die Differenz zwischen tatsächlichem Wert $x[n]$ und vorhergesagtem Wert $\hat{x}[n]$ ergibt den Vorhersagefehler:

$$\begin{aligned} e[n] &= x[n] - \hat{x}[n] \\ &= x[n] - \sum_{p=1}^P w_p x[n-p] \end{aligned} \quad (1.1)$$

Dass der Vorhersagefehler ebenso wie das Anregungssignal im Quelle-Filter Modell als $e[n]$ bezeichnet wird, ist kein Zufall - bei der linearen Prädiktion ist nämlich das Vorhersagefehler-Signal tatsächlich die Schätzung des Anregungssignals. Man kann das e also sowohl als 'error' als auch als 'excitation' lesen - je nachdem ob man den Aspekt der Signalvorhersage oder der Quelle-Filter-Modellierung in den Vordergrund stellen will. Ein Blick auf Gleichung 1.1 zeigt, dass die Vorschrift zur Berechnung des Vorhersagefehlers auch wieder nichts anderes als ein FIR-Filter ist - man bezeichnet dieses Filter in der Literatur auch als Prediction-Error-Filter, die Transferfunktion wird üblicherweise mit $A(z)$ bezeichnet. Bei der linearen Prädiktion minimiert man den mittleren quadratischen Vorhersagefehler, die Kostenfunktion lautet demnach:

$$\begin{aligned} E_{LP} &= \frac{1}{N} \sum_{n=1}^N (e[n])^2 \\ &= \frac{1}{N} \sum_{n=1}^N (x[n] - \hat{x}[n])^2 \end{aligned} \quad (1.2)$$

wobei N die Anzahl der Fehler-Samples ist, über die gemittelt wird. Der Index LP soll angeben, dass dies die Kostenfunktion bei der Linearen Prädiktion ist. Die Minimierung dieser Funktion bezüglich der Gewichte w_p (Ableiten nach den w_p , Nullsetzen der partiellen Ableitungen) führt zu einem linearen Gleichungssystem, der so genannten Wiener-Hopf Gleichung. Im Falle der linearen Prädiktion (diese Gleichung wird meist in einem etwas allgemeineren Zusammenhang hergeleitet) steht mit der Levinson/Durbin-Rekursion ein effizienter Algorithmus zur Lösung dieses Gleichungssystems zu Verfügung - für eine Herleitung siehe [6]. Man kann zeigen, dass die Minimierung des mittleren quadratischen Vorhersagefehlers tatsächlich dazu führt, dass die Prädiktor-Gewichte w_p die Werte der $-a_p$ in einem solchen Allpolmodell annehmen (für $N \rightarrow \infty$ und unter der Voraussetzung, dass das Anregungssignal weißes Rauschen ist und die Ordnung des Modells größer oder gleich der Ordnung des Synthesefilters ist). Wenn die w_p identisch mit den $-a_p$ sind, dann rekonstruiert das Prediction-Error Filter das Anregungssignal $e[n]$ im zugrundeliegenden Allpol-Modell. Ein Blick auf Abbildung 1.5 verdeutlicht dies: im Prediction-Error Filter $A(z)$ werden genau dieselben Werten subtrahiert, die zuvor im Synthesefilter $H(z)$ addiert wurden.

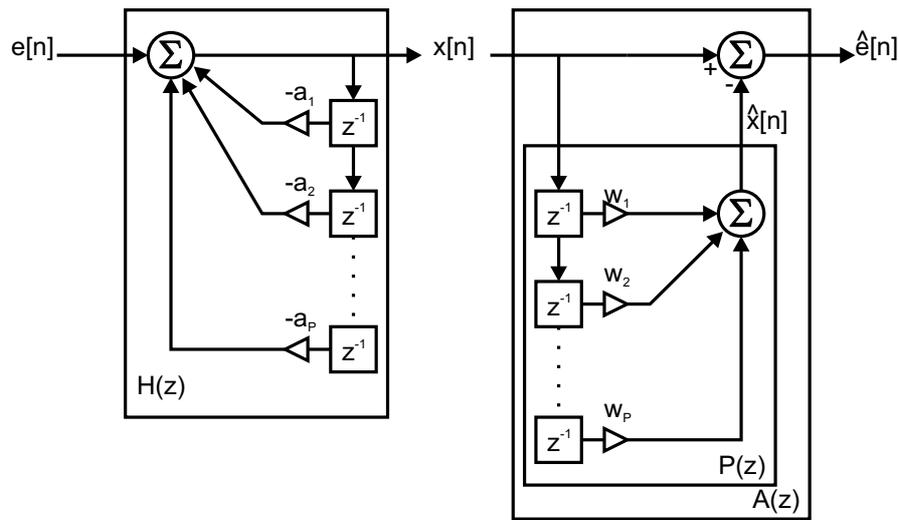


Fig. 1.5: Synthese-Allpol-Filter, Prädiktor und Prediction-Error-Filter

Durch Minimierung der Varianz des Vorhersagefehlers kann also ein Filter gefunden werden, dass aus dem beobachteten Signal $x[n]$ wieder das (als weiß angenommene) Anregungssignal $e[n]$ rekonstruiert. Dieses ist ein reines Nullstellenfilter und das inverse Filter zu dem Synthese-Allpolfilter, welches das Signal erzeugt hat. Die Transferfunktion dieses Prediction-Error Filters ist gegeben mit:

$$A(z) = 1 - \sum_{p=1}^P w_p z^{-p}$$

Wenn wir dieses Prediction-Error-Filter nun wiederum invertieren, erhalten wir demnach eine Schätzung für das Synthese-Allpolfilter aus unserem Signalmodell. Dieses ist gegeben durch die Transferfunktion:

$$\begin{aligned} \hat{H}(z) &= \frac{1}{A(z)} \\ &= \frac{1}{1 - \sum_{p=1}^P w_p z^{-p}} \end{aligned}$$

Oft wird in diese Transferfunktion auch noch ein globaler Gain-Faktor G aufgenommen, der sich aus der Varianz des Anregungssignals ergibt, so dass die Transferfunktion die Form:

$$\hat{H}(z) = \frac{G}{1 - \sum_{p=1}^P w_p z^{-p}} \quad (1.3)$$

annimmt. Der Frequenzgang dieses Allpolfilters wird als Schätzung der spektralen Hüllkurve benutzt. In Abbildung 1.6 ist eine typische spektrale Hüllkurve abgebildet, die auf diese Weise geschätzt wurde.

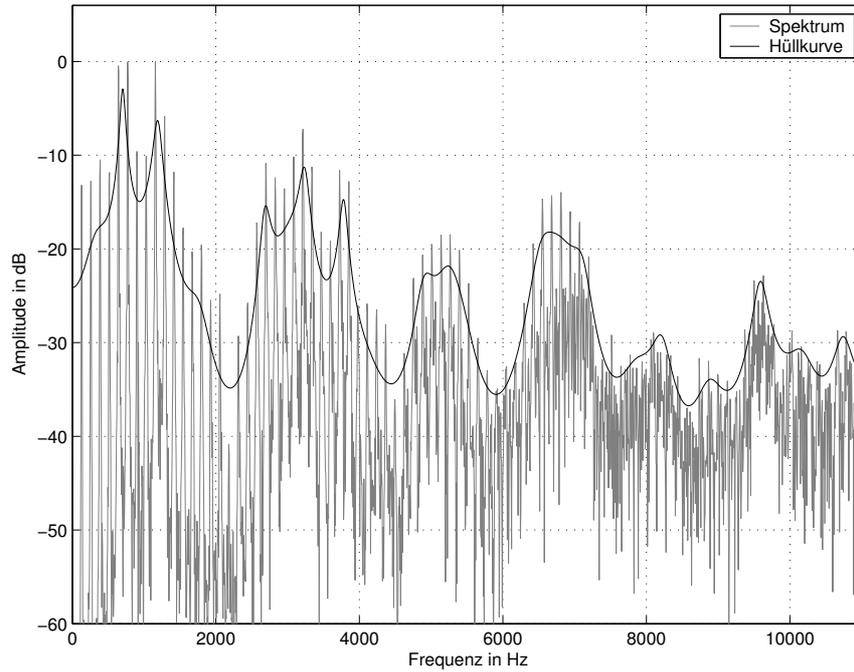


Fig. 1.6: FFT-Spektrum und spektrale Hüllkurve (geschätzt über Lineare Prädiktion, Ordnung: $P = 100$)

1.2.2 Schätzung des Filters über Discrete All-Pole Modeling

Die Schätzung eines AR-Modells über lineare Prädiktion basiert auf der Annahme eines weißen Rauschens als Anregungssignal. Bei Signalen mit periodischer Anregung liefert sie keine guten Ergebnisse - insbesondere wenn die Grundfrequenz des Anregungssignals sehr hoch liegt. In diesem Fall ist die spektrale Hüllkurve nur an den Frequenzen der einzelnen Harmonischen beobachtbar. Man kann zeigen (siehe [7]), dass die Kostenfunktion (1.2) im Frequenzbereich folgende Form annimmt:

$$E_{LP} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{P(\Omega)}{\hat{P}(\Omega)} d\Omega \quad (1.4)$$

bzw. bei diskreten Spektren:

$$E_{LP} = \frac{1}{K} \sum_{k=0}^{K-1} \frac{P(\Omega_k)}{\hat{P}(\Omega_k)} \quad (1.5)$$

wobei $P(\Omega)$ und $\hat{P}(\Omega)$ die Leistungsdichtespektren (die quadrierten Betragsfrequenzgänge) des Eingangssignals und des Modellfilters sind. Diese Kostenfunktion bedeutet, dass Lineare Prädiktion versucht, das gemittelte Verhältnis zwischen diesen beiden Leistungsdichtespektren zu minimieren (und diese Mittelung erfolgt über den gesamten Frequenzbereich bzw. über alle Bins). Bei Linienspektren, wie sie bei periodischen Anregungssignalen auftreten, ist $P(\Omega)$ natürlich fast überall nahe 0 und hat nur an bestimmten Frequenzstützstellen (den ganzzahligen Vielfachen der Grundfrequenz) relevante Werte. Dies führt bei Linearer Prädiktion zu dem Effekt, dass die Peaks des Modellfilters tendenziell in Richtung der Harmonischen verschoben werden. Beim Discrete Allpole Modelling

Verfahren (im Folgenden DAP genannt), wird als Kostenfunktion eine diskrete Version des Itakura-Saito Fehlers benutzt, der gegeben ist mit:

$$E_{IS} = \frac{1}{M} \sum_{m=1}^M \left(\frac{P(\Omega_m)}{\hat{P}(\Omega_m)} - \ln \frac{P(\Omega_m)}{\hat{P}(\Omega_m)} - 1 \right) \quad (1.6)$$

wobei die Mittelung hier nicht über alle Bins ausgeführt wird, sondern nur über diejenigen, an denen wir auch tatsächlich eine spektrale Information haben - also diejenigen Bins, die bei den Harmonischen des Eingangssignals liegen - daher wurde hier auch ein anderer Index für die Bins gewählt. Die Bezeichnung Discrete Allpole Modeling kommt daher, dass dies bei periodischen Eingangssignalen (idealisiert) diskrete Spektrallinien sind. Für kleine Werte von E_{IS} entspricht diese Kostenfunktion ungefähr dem mittleren quadratischen Abstand zwischen den logarithmierten Spektren. Da diese Kostenfunktion nur an bestimmten Frequenzstützstellen eine gute Anpassung fordert, wird sie dasjenige Allpol-Filter bevorzugen, welches an eben diesen Stützstellen am besten mit dem Signalspektrum übereinstimmt - und dabei ist der Kostenfunktion egal, wie das Spektrum des Modellfilters anderswo aussieht. Insbesondere wird nicht versucht, das Allpol-Spektrum auch auf die Bereiche zwischen den Harmonischen anzupassen, in denen wir ja sowieso keine Information haben. Die Peaks des Modellspektrums können dabei durchaus woanders liegen, als die (harmonischen) Peaks des Eingangssignals. Bei Harmonischen Signalen ist DAP in jedem Fall der Linearen Prädiktion vorzuziehen. Für mehr Details zum Verfahren siehe [4].

1.3 Cepstrum-basierte Schätzung der Hüllkurve

1.3.1 Schätzung der Hüllkurve über Cepstral Smoothing

Auch beim Cepstral Smoothing Verfahren stellt man sich das Signal als das Ergebnis einer Filterung eines Anregungssignals vor. Eine solche Filterung lässt sich im Zeitbereich als Faltung zwischen dem Anregungssignal und der Impulsantwort des Filters beschreiben. Im Frequenzbereich entspricht dies der Multiplikation des Spektrums des Anregungssignals mit der Übertragungsfunktion des Filters. Wir haben es demnach im Frequenzbereich mit einer multiplikativen Verknüpfung zweier Spektren zu tun. Um diese Verknüpfung rückgängig zu machen, wird sie zunächst in eine additive Verknüpfung umgewandelt, indem man das logarithmierte Spektrum betrachtet (denn es gilt: $\ln(a \cdot b) = \ln(a) + \ln(b)$). Wir wollen des weiteren annehmen, dass die 'schnellen' Veränderungen im Spektrum (d.h. die Amplitude ändert sich stark zwischen zwei nahe benachbarten Frequenzen) vom Anregungssignal verursacht wurden, und die 'langsamen' Veränderungen (d.h. der grundsätzliche Verlauf) durch das Filter verursacht wurden. Nun ist eine additive Überlagerung von schnell und langsam veränderlichen Anteilen etwas, das aus dem Zeitbereich wohlbekannt ist. Im Zeitbereich benutzen wir Tiefpass-Filter um die langsam veränderlichen Teile herauszufiltern und Hochpass-Filter um die schnell veränderlichen Anteile herauszufiltern - genau dasselbe können wir jedoch auch mit einem Spektrum tun. Wir könnten z.B. einfach eine Mittelung zwischen benachbarten Bins (im Beispiel sind dies 5 Stück)

vornehmen:

$$\tilde{X}[k] = \frac{1}{5} (X[k-2] + X[k-1] + X[k] + X[k+1] + X[k+2])$$

und erkennen, dass dies nichts anderes als ein (nicht kausales) moving-average (MA) Filter auf dem Spektrum darstellt. Das Ergebnis $\tilde{X}[k]$ wäre damit eine geglättete Version des ursprünglichen Spektrums $\tilde{X}[k]$. Nun kann man Filter einerseits direkt im Zeitbereich implementieren, andererseits aber auch als Multiplikation von Spektren im Frequenzbereich realisieren. Und etwas ganz ähnliches geschieht beim Cepstral Smoothing: Zuerst wird das logarithmierte Amplitudenspektrum (invers) Fourier-transformiert, dies ergibt das so genannte (reelle) Cepstrum:

$$C[r] = \sum_{k=0}^{K-1} \ln |X[k]| e^{j \frac{2\pi k r}{K}}$$

Das Cepstrum ist deshalb reell, weil das Betragsspektrum $|X(k)|$ eine gerade Symmetrie aufweist, dadurch werden die Imaginärteile der Fourier-Transformierten zu Null (hätten wir nicht das logarithmierte Betragsspektrum, sondern das logarithmierte komplexe Spektrum transformiert, hätten wir das komplexe Cepstrum erhalten - dies ist aber nur zur Information und im Folgenden nicht weiter relevant). Der Begriff 'Cepstrum' ist ein Kunstwort und ergibt sich durch Umkehrung der ersten Silbe im Wort 'Spectrum'. Dieses Wortspiel spiegelt die Tatsache wieder, dass es sich beim Cepstrum um die inverse Fourier-Transformierte eines (logarithmierten) Spektrums handelt (welches somit formal wieder im Zeitbereich angesiedelt ist). Die einzelnen Stützstellen werden als 'quefrequencies' bezeichnet. Wenn wir nun nur die frühen 'quefrequencies' beibehalten (die den langsamen Veränderungen im log-Spektrum entsprechen) und die späten 'quefrequencies' auf Null setzen, das Ganze dann zurücktransformieren und den Logarithmus durch die Exponentialfunktion rückgängig machen, erhalten wir also ein geglättetes Spektrum. In Abbildung 1.7 ist wieder das Spektrum des 'Ahh' zusammen mit einer cepstral geglätteten Version dieses Spektrums dargestellt.

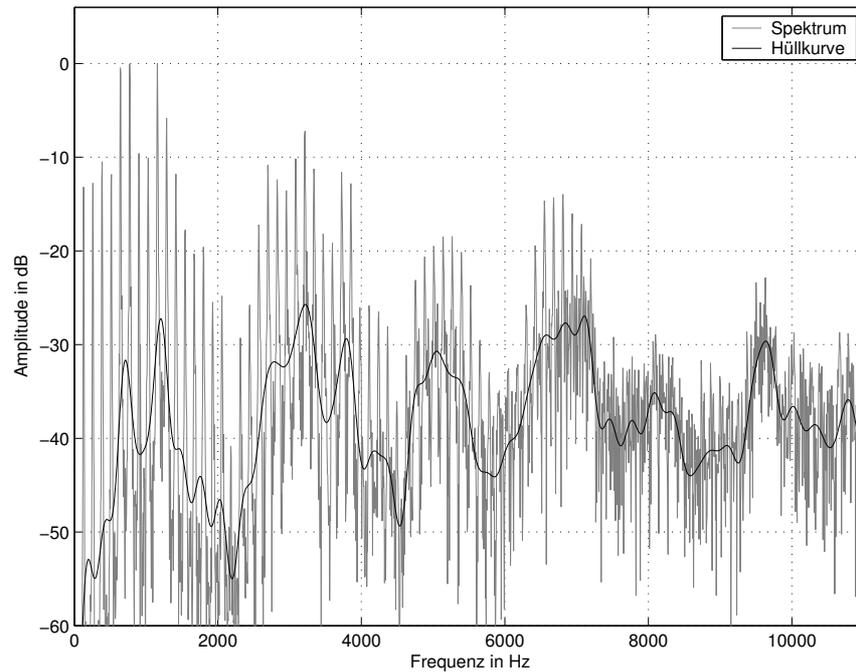


Fig. 1.7: FFT-Spektrum und spektrale Hüllkurve (geschätzt über Cepstral Smoothing, Ordnung: $P_c = 172$)

Die Abbildung zeigt deutlich ein mögliches Problem bei diesem Ansatz: das geglättete Spektrum folgt nicht den Peaks im Frequenzgang, sondern dem Mittelwert. Wie bei der Allpol-Modellierung auch, muß beim Cepstral Smoothing eine Entscheidung getroffen werden, wie viele Cepstral-Koeffizienten wir benutzen wollen - d.h. welche höchste 'quefrequency' wollen wir zulassen. Dies wird auch als die Ordnung des Cepstrums bezeichnet. Bei cepstral basierten Hüllkurven-Schätzern können wir diese Entscheidung recht gut automatisieren, wenn wir die Grundfrequenz des Signals kennen. In diesem Fall können wir nämlich einen minimalen Frequenzabstand angeben, bei dem wir im geglätteten Spektrum noch Details auflösen wollen. Wenn wir wissen, dass wir die einzelnen Spektrallinien, die an den ganzzahligen Vielfachen der Grundfrequenz auftreten, nicht einzeln auflösen wollen, erhalten wir eine maximal sinnvolle Ordnung P_c für unser Cepstrum (siehe [14]):

$$P_c \leq \frac{f_s}{2f_0}$$

wobei f_0 die Grundfrequenz des Signals und f_s die Sample-Rate sind. Wenn wir P_c kleiner als diesen Wert wählen, wird unsere spektrale Hüllkurve glatter, wenn wir P_c größer wählen, kommen wir langsam in den Bereich, wo einzelne Spektrallinien aufgelöst werden. Für die hier dargestellten Plots wurde diese optimale Ordnung gewählt. Die Grundfrequenz des Signals wurde dabei anhand der Autokorrelationsfunktion des Signals geschätzt (siehe [11]).

1.3.2 Schätzung der Hüllkurve mit dem True Envelope Algorithmus

Es wurde bereits erwähnt, dass Cepstral Smoothing zu einer geschätzten Hüllkurve führt, die dem Mittelwert und nicht den Peaks folgt. Beim True Envelope Algorithmus wird dieses Problem in einer iterativen Weise gelöst. Man beginnt damit, zwei Vektoren $A[k], V[k]$ mit dem log-Spektrum und $-\infty$ zu initialisieren. In einer Schleife wird dann jeweils $A[k]$ elementweise mit dem Maximum von $A[k]$ und $V[k]$ überschrieben - in der ersten Iteration ist dieses Maximum an allen Bins gleich $A[k]$, da ja $V[k]$ noch $-\infty$ ist. Jetzt wird $A[k]$ cepstral geglättet, und diese geglättete Version in $V[k]$ geschrieben. Wieder vergleichen wir $V[k]$ elementweise mit $A[k]$ und behalten jeweils den größeren Wert. Man iteriert diese Schritte so lange, bis die maximale Differenz zwischen dem originalen log-Spektrum und dem geglätteten log-Spektrum einen gewissen Schwellwert in dB unterschreitet (in [14] wurde dieser Schwellwert z.B. als $\Delta = 2dB$ gewählt). Eine Variante des Algorithmus ist in Algorithmus 1 als Pseudo-Code angegeben. Eine 'Variante' ist es deshalb, weil hier nicht der natürliche Logarithmus des Spektrums genommen wird, sondern dB -Werte. Dies macht konzeptionell keinen Unterschied, erleichtert aber den Vergleich der maximalen Differenz mit dem Schwellwert Δ .

Algorithmus 1 True Envelope Algorithmus (ursprüngliche Form)

```

 $P_c \leftarrow \lfloor f_s / (2f_0) \rfloor$ 
 $\Delta \leftarrow 2$ 
 $X_{dB}[k] \leftarrow 20 \log_{10}(X[k]) \quad k = 0, \dots, K - 1$ 
 $A[k] \leftarrow X_{dB}[k] \quad k = 0, \dots, K - 1$ 
 $V[k] \leftarrow -\infty \quad k = 0, \dots, K - 1$ 
converged  $\leftarrow$  false
while not converged do
    1.) nimm elementweise das Maximum von  $A[k], V[k]$ :
     $A[k] \leftarrow \max(A[k], V[k]) \quad k = 0, \dots, K - 1$ 
    2.) transformiere in den Cepstral-Bereich:
     $C' \leftarrow iFFT(A)$ 
    3.) Fenster das aktuelle Cepstrum  $C'$  mit Rechteckfenster:
     $C[k] \leftarrow C'[k] \quad k = 0, \dots, P_c$  and  $k = K - P_c, \dots, K - 1$ 
     $C[k] \leftarrow 0 \quad k = P_c + 1, \dots, K - P_c - 1$ 
    4.) transformiere das gefensterete Cepstrum  $C$  zurück in den log-Spektralbereich:
     $V \leftarrow FFT(C)$ 
    5.) überprüfe Konvergenzkriterium:
    if  $\max(X_{dB} - V) \leq \Delta$  then
        converged  $\leftarrow$  true
    end if
end while
return  $10^{V/20}$ 

```

Eine paar Anmerkungen zu einigen technischen Details in Algorithmus 1: Die Variablen A, V, C sind jeweils Vektoren (eigentlich sogar komplexe Vektoren, der Imaginärteil ist je-

doch Null). Die Schreibweise $10^{V/20}$ meint einen Vektor, dessen Element k sich zu $10^{V[k]/20}$ ergibt - die Operation ist also elementweise auszuführen. In der Anweisung, die bestimmte Cepstral-Koeffizienten aus C'_i in C_i kopiert und andere auf Null setzt (Schritt 3), ist die inherente Symmetrie der diskreten Fourier Transformation (DFT) berücksichtigt, wobei davon ausgegangen wird, dass die Array-Indizes von $0, \dots, K-1$ laufen (was auch in vielen Programmiersprachen der Fall ist, in MatLab jedoch laufen sie von $1, \dots, K$, so dass die Anweisung in MatLab entsprechend angepasst werden muss. In allen Zeilen, wo ein $k = 0, \dots, K-1$ auftaucht, wird man in den meisten Programmiersprachen for-Schleifen verwenden - in MatLab jedoch können solche Anweisungen oft durch Vektor-Operationen ausgedrückt werden.

Verbesserungen des Algorithmus

Fensterfunktionen Der angegebene Pseudocode stellt die ursprüngliche Form des True-Envelope Algorithmus dar. Obwohl der Algorithmus in dieser Form schon recht brauchbare Ergebnisse liefert, gibt es dennoch folgendes Problem: Das Nullsetzen der Cepstral-Koeffizienten oberhalb von P_c ist äquivalent zu einer Rechteckfensterung des Cepstrums mit der Fenstermitte bei $r = 0$. Durch diese Rechteck-Fensterung des Cepstrums entsteht ein 'Überschwingen' der resultierenden spektralen Hüllkurve an Stellen, bei denen sich das Spektrum sehr schnell ändert. Dieses Überschwingen ist in der Signalverarbeitung als Gibbs'sches Phänomen bekannt, und kann in den Griff bekommen werden, indem man andere Fenster als das Rechteckfenster verwendet. In [14] wurde beispielsweise ein Hamming-Fenster benutzt, dessen Breite so angepasst wurde, dass die Hauptkeule ungefähr dieselbe Breite hat wie die des entsprechenden Rechteckfensters - dazu muss als Breite des Hamming-Fensters ca. $1.66 \cdot P_c$ gewählt werden (das Rechteckfenster ist im Cepstral-Bereich genau P_c breit).

Wenn man jetzt aber einfach in Algorithmus 1 die Multiplikation mit dem Rechteckfenster (Schritt 3) durch eine Multiplikation mit einem Hamming-Fenster ersetzt, handelt man sich folgendes neue Problem ein: Aufgrund dessen, dass das Hamming-Fenster (wie auch die meisten anderen gängigen Fensterfunktionen) im Durchlassbereich nicht identisch 1 ist, werden bei iterierter Anwendung des Fensters auch diejenigen Cepstral-Koeffizienten signifikant gedämpft, die wir eigentlich behalten wollen - d.h. wir bekommen eine unerwünschte Dämpfung innerhalb des (cepstralen) Durchlassbereiches, welche zu einer systematisch zu starken Glättung der resultierenden spektralen Hüllkurve führt. Deutlich wird dies wenn man die Berechnung des gefensternten Cepstrums C_i als Update- bzw. Lernregel von Iteration $i-1$ zu Iteration i der Form $C_i = f(C_{i-1}, \dots)$ auffasst. Im ursprünglichen Algorithmus hätte diese Lernregel dann die Form:

$$\begin{aligned} C_i[r] &= W[r]C'_i[r] \\ &= W[r](C_{i-1}[r] + C'_i[r] - C_{i-1}[r]) \\ &= W[r]C_{i-1}[r] + W[r](C'_i[r] - C_{i-1}[r]) \\ &= W[r]C_{i-1}[r] + W[r]D_i[r] \end{aligned}$$

wobei mit $D_i[r]$ eine Bezeichnung für die Differenz $C'_i[r] - C_{i-1}[r]$ eingeführt wurde und $W[r]$ das cepstrale Fenster bezeichnen soll. Der erste Summand bedeutet, dass die Beiträge

aus früheren Iterationen zu den 'hochfrequenten' (aber immer noch innerhalb des Fensters liegenden) Cepstral-Koeffizienten tendenziell im Laufe der Iterationen abklingen, da bei diesen Frequenzen $W[r] < 1$ gilt. Beim Hamming Fenster beispielsweise sind alle $W[r] < 1$ außer für $r = 0$, und je kleiner der Wert $W[r]$ ist, desto schneller klingen die Beiträge aus früheren Iterationen zu $C_i[r]$ ab. Beim Rechteckfenster gibt es dieses Problem nicht, da hier alle Werte $W[r]$, die innerhalb des Fensters liegen, gleich eins sind. Die Lösung dieses Problems liegt nun darin, die Fensterfunktion nur jeweils auf die Änderung des gefensterten Cepstrums C_i anzuwenden. Die Update-Regel wird dazu in folgender Weise modifiziert:

$$C_i[r] = C_{i-1}[r] + W[r]D_i[r] \quad (1.7)$$

mit dieser modifizierten Lernregel ist nun die problemlose Benutzung beliebiger Fensterfunktionen möglich, ohne dass wir eine systematisch zu starke Glättung der spektralen Hüllkurve befürchten müssen. Für die Initialisierung von C , d.h. für C_{-1} wählen wir dabei den Nullvektor. Im speziellen Falle des Rechteckfensters reduziert sich diese Regel zu der ursprünglichen Form.

Optimale Lernschrittweite Wenn wir nun schon an dem Punkt sind, dass wir den Update-Schritt in Form einer Lernregel $C_i[r] = C_{i-1}[r] + \Delta_i[r]$ formulieren können, ist es eine nahe liegende Fragestellung, ob es nicht eventuell möglich wäre, die Konvergenz des Algorithmus durch eine geeignet gewählte Lernschrittweite λ zu beschleunigen. Die Lernregel bekommt damit folgende Form:

$$C_i[r] = C_{i-1}[r] + \lambda_i \cdot W[r]D_i[r]$$

wobei wir jetzt also in jedem Lernschritt i einen geeigneten Wert für die Lernrate λ_i suchen (in Gleichung 1.7 wäre sie gleich 1). Man kann zeigen, dass für den speziellen Fall eines Impulskamm-artigen Spektrums eine Konvergenz in einem Schritt erzielt wird, wenn man die Lernrate λ_i folgendermaßen wählt:

$$\lambda_i = \frac{E(D_i)}{E_{in}(D_i)} \quad (1.8)$$

wobei mit $E(D_i)$ die Gesamtenergie des Differenz-Cepstrums D_i gemeint ist:

$$E(D_i) = \sum_{k=0}^{K-1} D_i^2[k] \quad (1.9)$$

und mit $E_{in}(D_i)$ derjenige Anteil der cepstralen Energie innerhalb des Durchlassbereiches unseres cepstralen Fensters ('in-band-energy') gemeint ist:

$$E_{in}(D_i) = \sum_{k=0}^{P_c} D_i^2[k] + \sum_{k=K-P_c}^{K-1} D_i^2[k] \quad (1.10)$$

Die Aufspaltung in zwei Teilsummen kommt hierbei durch die Berücksichtigung der DFT-Symmetrie zustande - unser cepstrales Fenster ist schließlich symmetrisch um $r = 0$ und die negativen Frequenzen ($r < 0$) werden im DFT-Array auf den Bereich $K - 1, K -$

$2 \dots, K/2+1$ abgebildet. Da bei beliebigen nicht-rechteckigen Fensterfunktionen die Grenze zwischen 'im Fenster' und 'außerhalb des Fensters' nicht so scharf abgegrenzt ist, kann in diesem Fall auch eine gewichtete Summe sinnvoll sein:

$$E_{in}(D_i) = \sum_{k=0}^{K-1} W^2[k] D_i^2[k] \quad (1.11)$$

bei der ein Cepstral-Anteil danach gewichtet wird, wie weit innerhalb des Fensters er liegt. Beim Rechteckfenster reduziert sich dies auch wieder zu der zuvor angegebenen Formel. Die so errechnete Lernschrittweite ist in jedem Fall ≥ 1 , da ja $E_{in}(D_i) \leq E(D_i)$ gelten muss (denn E_{in} summiert ja weniger Terme auf) - demzufolge sind die Lernschritte jetzt größer als vorher. Da wir es aber in der Praxis nicht mit idealen Impulskamm-Spektren zu tun haben, kann es Sinn machen, λ kleiner als oben genannten Wert zu wählen um zu große Schritte zu vermeiden - diese könnten unter Umständen nämlich dazu führen, dass im letzten Schritt die Hüllkurve zu weit nach oben rutscht. In [13] wurde die Wurzel aus obigem Wert verwendet: $\lambda = \sqrt{E(D_i)/E_{in}(D_i)}$. Zwischen diesen beiden Varianten liegt ein ganzes Kontinuum von Möglichkeiten - allgemein könnte man $\lambda = (E(D_i)/E_{in}(D_i))^c$ mit einer Konstanten $0.5 \leq c \leq 1$ wählen. Man kann den Exponenten natürlich auch noch kleiner machen, bis man schließlich wieder bei $\lambda = 1$ für $c = 0$ ankommt. Während dieser Arbeit hat sich empirisch ein Wert von $c = 0.8$ als recht praxistauglich herausgestellt. Die Einführung von Fensterfunktionen und einer optimierten Lernschrittweite führt zu Algorithmus 2.

Algorithmus 2 True Envelope Algorithmus (verbesserte Form)

```

 $P_c \leftarrow \lfloor f_s / (2f_0) \rfloor$ 
 $\Delta \leftarrow 2$ 
 $X_{dB}[k] \leftarrow 20 \log_{10}(X[k]) \quad k = 0, \dots, K - 1$ 
 $A[k] \leftarrow X_{dB}[k] \quad k = 0, \dots, K - 1$ 
 $V[k] \leftarrow -\infty \quad k = 0, \dots, K - 1$ 
 $C[k] \leftarrow 0 \quad k = 0, \dots, K - 1$ 
 $c \leftarrow$  irgendwo zwischen 0.5 ... 1.0, 0.8 ist gut in der Praxis
converged  $\leftarrow$  false
while not converged do

    nimm elementweise das Maximum von  $A[k], V[k]$ :
     $A[k] \leftarrow \max(A[k], V[k]) \quad k = 0, \dots, K - 1$ 

    transformiere in den Cepstral-Bereich:
     $C' \leftarrow iFFT(A)$ 

    Berechne die Differenz zum Cepstrum  $C$  aus der letzten Iteration:
     $D[k] = C'[k] - C[k] \quad k = 0, \dots, K - 1$ 

    berechne die 'in-band-energy' und Gesamtenergie dieses Differenz-Cepstrums:
     $E = \sum_{k=0}^{K-1} D^2[k]$ 
     $E_{in} = \sum_{k=0}^{K-1} W^2[k] \cdot D^2[k]$ 

    berechne die optimale Lernschrittweite:
     $\lambda = (E/E_{in})^c$ 

    führe den Lernschritt für  $C$  aus:
     $C[k] \leftarrow C[k] + \lambda(W[k] \cdot D[k]) \quad k = 0, \dots, K - 1$ 

    transformiere das neue Cepstrum  $C$  zurück in den log-Spektralbereich:
     $V \leftarrow FFT(C)$ 

    überprüfe Konvergenzkriterium:
    if  $\max(X_{dB} - V) \leq \Delta$  then
        converged  $\leftarrow$  true
    end if

end while
return  $10^{V/20}$ 

```

In der MatLab-Funktion `spectralEnvelopeViaTE` (siehe C) ist der Algorithmus in dieser Form implementiert. Diese Implementation wurde benutzt um die spektrale Hüllkurve unseres Beispielsignals, das 'Ahh', zu schätzen. Die Ergebnisse sind in den Abbildungen 1.8 und 1.9 zu sehen, wobei zum Vergleich einmal des Rechteck- und einmal das Hamming-Fenster zum Einsatz kam.

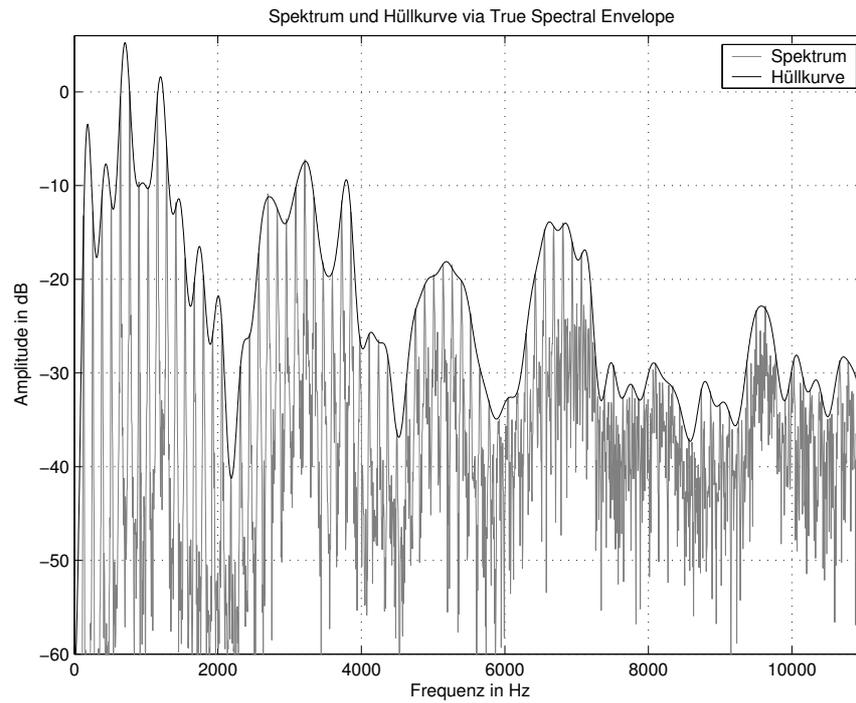


Fig. 1.8: FFT-Spektrum und spektrale Hüllkurve (geschätzt mit dem True Envelope Algorithmus mit Rechteckfenster, Ordnung: $P_c = 172$)

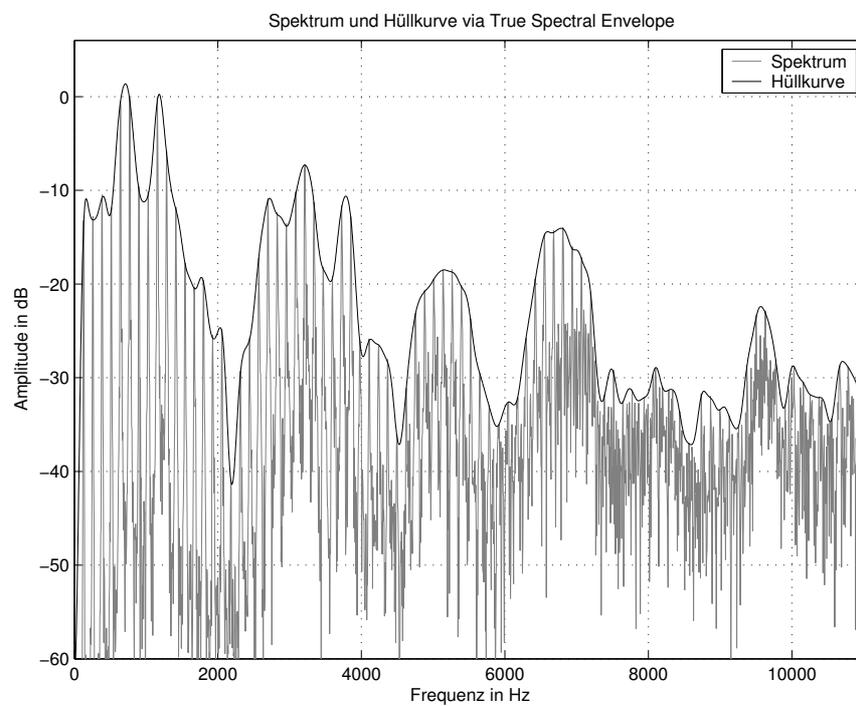


Fig. 1.9: FFT-Spektrum und spektrale Hüllkurve (geschätzt mit dem True Envelope Algorithmus mit Hamming-Fenster, Ordnung: $P_c = 172$)

Wie man sieht, liefert der True Envelope Algorithmus Hüllkurven, die in der Tat sehr gut mit dem übereinstimmen, was man als Mensch intuitiv als spektrale Hüllkurve einzeichnen würde. Der Unterschied zwischen Rechteck- und Hamming-Fenster mag auf den ersten Blick nicht allzu dramatisch aussehen, wenn man jedoch bedenkt, dass wir es hier mit einer Darstellung im dB -Bereich zu tun haben und dann z.B. das erste Maximum der Hüllkurve vergleicht, sieht man, dass dieses beim Rechteckfenster um ca. 6 dB über das eigentliche spektrale Peak hinauschießt - d.h. in einer linearen Darstellung würde die geschätzte Hüllkurve an dieser Stelle zweimal so groß sein wie eigentlich erwünscht. Beim Hamming-Fenster ist dieses Überschwingen erheblich reduziert. Die Verwendung von Fensterfunktionen stellt somit offensichtlich eine erhebliche Verbesserung des Algorithmus dar. Die Anpassung der Lernschrittweite auf der anderen Seite, bewirkt eine erhebliche Beschleunigung der Konvergenz des Algorithmus. Wenn man bedenkt, dass in jeder Iteration eine FFT und eine $iFFT$ ausgeführt werden müssen, ist dies natürlich auch ein wichtiger Punkt.

Weitere Effizienzsteigerungen sind möglich, wenn man in Betracht zieht, dass das Eingangsspektrum relevante Information eigentlich nur bei den spektralen Peaks enthält und somit sehr viel für unsere Zwecke irrelevante Information enthält. Durch geeignete Unterabtastung des Spektrums (siehe [14]) kann der Rechenaufwand weiter reduziert werden. Da in dieser Arbeit jedoch keine Echtzeit-Implementierung realisiert werden soll, soll dies hier nicht weiter ausgeführt werden.

1.4 Umrechnung der Koeffizienten und TE-LPC

Obwohl die Cepstrum-basierte und die autoregressive Schätzung der spektralen Hüllkurve auf den ersten Blick nichts miteinander zu tun zu haben scheinen, ist es möglich, die Cepstral-Koeffizienten in AR-Koeffizienten umzurechnen und umgekehrt. Dazu werden die log-Spektren aus beiden Verfahren gleichgesetzt:

$$\ln \left(\frac{G}{1 - \sum_{k=1}^P w_k e^{-j\Omega k}} \right) = \sum_{k=0}^{\infty} C[k] e^{-j\Omega k}$$

Einige trickreiche Umformungen führen dann zu folgenden Rekursionsformeln für die Berechnung der Cepstral-Koeffizienten aus den AR-Koeffizienten:

$$\begin{aligned} C[0] &= \ln(G) \\ C[1] &= w_1 \\ C[n] &= \sum_{k=1}^{n-1} \left(1 - \frac{k}{n} \right) w_k C[n-k] + w_n \quad \text{mit } w_k = 0 \text{ für alle } n > P \end{aligned} \quad (1.12)$$

bzw. für die Berechnung der AR-Koeffizienten aus den Cepstral-Koeffizienten:

$$\boxed{\begin{aligned} G &= e^{C[0]} \\ w_1 &= C[1] \\ w_n &= - \sum_{k=1}^{n-1} \left(1 - \frac{k}{n} \right) w_k C[n-k] + c_n \end{aligned}} \quad (1.13)$$

Mit Hilfe von Gl. 1.13 könnte man also z.B. mit dem True-Envelope Algorithmus ein autoregressives Signalmodell zu bestimmen. In der Praxis stellt sich aber heraus, dass diese Rekursionen nur in einer Richtung gut funktionieren, nämlich zur Umrechnung von AR-Koeffizienten in Cepstral-Koeffizienten. Die Herleitung der Rekursionen geht nämlich jeweils von unendlichen Modellordnungen aus, was in der Praxis natürlich nicht gegeben ist. Für genauere Betrachtungen siehe [11].

Eine andere Möglichkeit, aus dem True-Envelope die Parameter für eine autoregressive Signalmodell zu bestimmen, ist in [12] beschrieben. Hier wird aus dem gefundenen True-Envelope zunächst die dazugehörige Autokorrelationsfunktion bestimmt. Autokorrelationsfunktionen sind im allgemeinen die inverse Fourier-Transformation des Betragsquadrats-Spektrums - wenn wir den True-Envelope also auf einer logarithmischen Amplituden-Skala vorliegen haben, müssen wir zunächst in die lineare Darstellung umrechnen (via Exponentialfunktion) und quadrieren. Jetzt wird eine *iFFT* angewandt, um die Autokorrelationsfunktion zu bestimmen und die so gefundenen Autokorrelationskoeffizienten werden dann als Eingabe für die Levinson/Durbin Rekursion benutzt (anstatt der Autokorrelationskoeffizienten des Signals selbst wie bei der konventionellen linearen Prädiktion). Die so bestimmten AR-Koeffizienten definieren ein autoregressives Filter, dessen Frequenzgang wesentlich besser mit unserer spektralen Hüllkurve übereinstimmt als das über konventionelle Lineare Prädiktion bestimmte Filter. Daher werden die in den folgenden Kapiteln dieser Arbeit verwendeten AR-Modelle vorzugsweise über diese so genannte TE-LPC Methode bestimmt werden.

1.5 Anwendungen und Grenzen

Anwendungen

Nachdem nun also verschiedene Verfahren zur Schätzung der spektralen Hüllkurve kurz vorgestellt wurden, sollen nun einige Anwendungsmöglichkeiten betrachtet werden.

Wenn wir die spektrale Hüllkurve eines Signals kennen, können wir diese aus dem Spektrum herausrechnen, indem wir jedes FFT-Bin des Spektrums durch den Wert der spektralen Hüllkurve an diesem Bin dividieren. Das resultierende Spektrum stellt eine geweißte Version des Originalspektrums dar. Nun können wir die spektrale Hüllkurve in beliebiger Weise transformieren und danach die transformierte Hüllkurve wieder mit dem geweißten Spektrum multiplizieren. Dies kann z.B. genutzt werden für:

Formant-Scaling Eine einfache Transformation der spektralen Hüllkurve besteht darin, sie entlang der Frequenzachse strecken oder zu stauchen - dies führt dazu, dass die Mittenfrequenzen von sämtlichen Formanten und Antiformanten um denselben Faktor auf der Frequenzachse nach oben oder unten verschoben werden. Realisiert werden kann dies durch eine Abbildung der Form:

$$Y(\omega) = X(\omega/c)$$

wobei $X(\omega)$ die originale spektrale Hüllkurve, $Y(\omega)$ die transformierte spektrale Hüllkurve und c der Streckungs- bzw. Skalierungsfaktor ist. Um die Sache zu vereinfachen, wurde hier von einer kontinuierlichen Funktion $X(\omega)$ ausgegangen - obige Gleichung soll nur die Idee vermitteln. In der Praxis haben wir es jedoch mit diskreten FFT-Bins zu tun, so dass zur Berechnung eines diskreten $Y[k]$ eine geeignete Interpolation von $X[k]$ erfolgen muss. Des weiteren müssten wir im Falle von $c < 1$ (Stauchung) bei den hohen $Y[k]$ die spektrale Hüllkurve $X[k]$ bei Frequenzen auslesen, die größer als die halbe Sample-Rate sind. Da diese Frequenzen jedoch nicht wirklich existieren, würde man hier einfach die spektrale Hüllkurve Null setzen. Außerdem muss natürlich die Symmetrie der DFT beachtet werden. Diese Details sollen hier jedoch nicht weiter ausgeführt werden. Bei einer solchen Streckung/Stauchung bleiben die Frequenzverhältnisse zwischen den einzelnen Formant- und Antiformant-Mittenfrequenzen erhalten. Physikalisch entspräche dies einer Verkleinerung oder Vergrößerung des Resonanzkörpers, je nachdem, ob der Skalierungsfaktor c größer oder kleiner als 1 ist. Damit könnte man Effekte wie z.B. eine 'Mickey-Mouse-Stimme' ohne Tonhöhenverschiebung realisieren.

Formant-Shifting Beim Formant-Shifting wird die spektrale Hüllkurve entlang der Frequenzachse verschoben:

$$Y(\omega) = X(\omega - c)$$

Dadurch würden sich auch Frequenzverhältnisse zwischen den Formant-Mittenfrequenzen ändern - wenn man sich einen physikalischen Resonanzkörper vorstellt, würde dieser dadurch (zusätzlich zu einer insgesamt Vergrößerung oder Verkleinerung) verformt werden.

Formant-Warping Schließlich könnte man auch beliebige nichtlineare Funktionen auf die Hüllkurve anwenden. Dies ist der allgemeinste Fall:

$$Y(\omega) = X(f(\omega))$$

wobei die Funktion $f(\omega)$ die so genannte Warping-Map ist. Diese Funktion sollte eindeutig invertierbar sein, um eine eins-zu-eins Abbildung zwischen der originalen und der neuen Frequenzachse zu gewährleisten.

Pitch-Shifting mit Formanterhaltung Beim Pitch-Shifting wird die Tonhöhe eines Audio-signals verändert. Dabei ändert sich nicht nur die Grundfrequenz, sondern mit der Grundfrequenz werden auch sämtliche Formantfrequenzen verschoben. Für natürlich klingendes Pitch-Shifting ist dies unerwünscht - beim Pitch-Shift einer Stimme nach oben würde es z.B. einen Mickey-Mouse-Effekt geben. Daher ist es sinnvoll, zuerst die spektrale Hüllkurve aus dem Signal herauszurechnen, dann den Pitch-Shifting Algorithmus auszuführen und hinterher die originale spektrale Hüllkurve wieder darüberzulegen. In gewisser Weise ist dies Umkehrung des Formant-Scalings: mit Formant-Scaling kann eine Mickey-Mouse-Stimme ohne Tonhöhenverschiebung erzeugt werden, während hier eine Tonhöhenverschiebung ohne Mickey-Mouse-Effekt erzielt wird.

Vocoding Als Vocoder bezeichnet man im Allgemeinen einen Effekt, bei dem die spektrale Hüllkurve eines so genannten Modulatorsignals auf ein Trägersignal übertragen wird. Typischerweise wird als Trägersignal gerne ein Flächenklang und als Modulatorsignal eine Stimme benutzt. Beim klassischen Kanal-Vocoder kommen zur Analyse der spektralen Hüllkurve des Modulators sowie zur Synthese des Vocoder-Signals Filterbänke zum Einsatz. Es bietet sich jedoch natürlich auch an, die in 1.2 und 1.3 beschriebenen Methoden für diesen Effekt zu nutzen. Zum Beispiel kann man zuerst jeweils von Modulator- und Trägersignal die spektralen Hüllkurven bestimmen, dann aus dem Trägersignal dessen spektrale Hüllkurve herausrechnen (d.h. das Trägerspektrum durch seine spektrale Hüllkurve dividieren) und schließlich die spektrale Hüllkurve des Modulatorsignals mit dem geweißten Trägerspektrum multiplizieren (das Ganze natürlich blockweise mit Overlap/Add). Ein solcher Vocoder wurde in dem m-File `trueEnvelopeVocoder.m` implementiert - bei der typischen Verwendung dieses Vocoders mit Flächenklang als Träger und Stimme als Modulator bietet diese Implementation eine exzellente Sprachverständlichkeit.

Für ausführlichere Betrachtungen zu Transformationen der spektralen Hüllkurve siehe [20].

Grenzen

Alle oben beschriebenen Effekte benutzen bzw. transformieren die spektrale Hüllkurve als Ganzes - d.h. die Transformationen sind im Wesentlichen globaler Natur. In einigen Fällen ist es jedoch wünschenswert, bestimmte Aspekte der spektralen Hüllkurve getrennt voneinander editieren zu können. Beispielsweise könnte man das erste Peak in der spektralen Hüllkurven nach unten, das zweite jedoch nach oben verschieben wollen, um z.B.

den Vokal 'a' in den Vokal 'e' zu transformieren. Dazu benötigen wir eine Repräsentation der spektralen Hüllkurve, in der wir bestimmte, intuitiv verständliche Parameter zur Verfügung haben, die wir gezielt verändern können - wie z.B. Formant-Mittenfrequenzen und -Bandbreiten. Wie wir aus der rohen spektralen Hüllkurve solche Parameter gewinnen können, und welche zusätzlichen Transformationsmöglichkeiten uns diese Parameter geben, ist Gegenstand des nächsten Kapitels, welches den Hauptteil dieser Arbeit darstellt.

2. FORMANTMODELLE

2.1 Serielle Allpolfilter

2.1.1 Das Modell

In diesem Abschnitt soll eine serielle Verschaltung von Allpolfiltern zweiter Ordnung als Modell für unser Synthesefilter dienen. Das zu dieser Filterstruktur gehörende Blockschaltbild ist in Abb. 2.1 dargestellt.

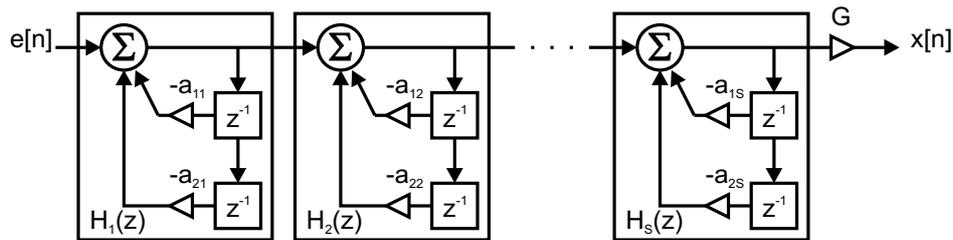


Fig. 2.1: Serielle Verschaltung von S Allpolfiltern 2.Ordnung

Die Differenzgleichung einer einzelnen Filterstufe mit dem Index s lautet:

$$y_s[n] = y_{s-1}[n] - a_{1s}y_s[n-1] - a_{2s}y_s[n-2] \quad (2.1)$$

wobei $y_{s-1}[n]$ das Ausgangssignal derjenigen Stufe ist, die der Stufe s in der Reihe vorge-schaltet ist. Bei der ersten Filterstufe ist dieses Signal durch das Eingangssignal des Filters zu ersetzen. Die Transferfunktion einer einzelnen Filterstufe mit dem Index s gegeben ist durch:

$$H_s(z) = \frac{1}{1 + a_{1s}z^{-1} + a_{2s}z^{-2}} \quad (2.2)$$

Die Transferfunktion einer Reihenschaltung S solcher Filterstufen ergibt sich als Multiplikation der Transferfunktionen der einzelnen Stufen, außerdem wird ein globaler Gain-Faktor G eingeführt:

$$\begin{aligned} H(z) &= G \prod_{s=1}^S H_s(z) \\ &= G \prod_{s=1}^S \frac{1}{1 + a_{1s}z^{-1} + a_{2s}z^{-2}} \end{aligned} \quad (2.3)$$

Für die Berechnung der Filterkoeffizienten a_{1s}, a_{2s} der einzelnen Stufen gehen wir von einem Allpolfilter in Direktform aus - so wie es z.B. durch Lineare Prädiktion, Discrete Allpole Modeling oder TE-LPC geliefert wird. Im weiteren Verlauf dieser Arbeit werden wir solche Allpol-Filter grundsätzlich mit der TE-LPC Methode schätzen, da die Frequenzgänge der resultierenden Filter bei dieser Methode am besten mit der spektralen Hüllkurve übereinstimmen. Dieses Allpolfilter müssen wir dann in eine serielle Verschaltung von Filtern zweiter Ordnung zerlegen. Wie später noch genauer ausgeführt werden wird, müssen dazu zunächst die Nullstellen des Nennerpolynoms der Transferfunktion des Allpolfilters gefunden werden (das sind die Polstellen des Filters). Da diese Polstellen entweder reell sind, oder als konjugiert komplexe Paare auftreten, können wir je zwei solcher konjugiert komplexer Pole zu einem quadratischen Polynom mit reellen Koeffizienten zusammenfassen - und dies sind genau die gesuchten Koeffizienten für eine Filterstufe. Einzelne reelle Polstellen werden in dieser Filterstruktur als Spezialfall mit erfasst, denn sie können durch ein Filter realisiert werden, bei dem der Koeffizient a_2 zu Null wird. Jede dieser Zweipol-Filterstufen hat demnach zwei Freiheitsgrade a_1, a_2 (oder auch nur einen, wenn $a_2 = 0$ ist). Es jedoch intuitiv besser verständlich, wenn man nicht die Koeffizienten selbst als einstellbare Parameter betrachtet, sondern den Betrag und Winkel des jeweiligen Polstellenpaares, das durch das Filter realisiert wird. Dieses Polstellenpaar wollen wir mit den komplexen Zahlen $re^{j\varphi}$ und $re^{-j\varphi}$ bezeichnen. Der Winkel φ des Polpaares entspricht dann nämlich der Resonanzfrequenz des Filters (als normierte Kreisfrequenz), d.h. die zugehörige Resonanzfrequenz in Hz ist:

$$f_{res} = \frac{\varphi}{2\pi} f_s \quad (2.4)$$

Der Radius r andererseits bestimmt, wie stark die Resonanz ausgeprägt ist. Der Zusammenhang zwischen Radius r und Winkel φ einerseits und den Filterkoeffizienten a_1 und a_2 andererseits ist wie folgt:

$$\begin{aligned} a_1 &= -2r \cos \varphi \\ a_2 &= r^2 \end{aligned} \quad (2.5)$$

Hierbei ist anzumerken, dass die Resonanzfrequenz φ und diejenige Frequenz, bei der im Frequenzgang des Filters ein Peak auftritt, im allgemeinen nicht identisch sind, da die beiden Pole auch noch miteinander wechselwirken (siehe [19]). Jeder der beiden Pole sitzt nämlich, bildlich gesprochen, auf der Flanke des jeweils anderen. Der Zusammenhang zwischen Resonanzfrequenz φ des Filters und der Peak-Frequenz (die hier mit ψ bezeichnet werden soll) ist wie folgt:

$$\cos \psi = \frac{1 + r^2}{2r} \cos \varphi \quad (2.6)$$

Für den Fall, dass $r \approx 1$ ist jedoch die Vorstellung, dass bei $\Omega = \varphi$ ein Peak im Frequenzgang auftritt, eine recht gute Näherung. In [18] findet man für diesen Fall auch eine Näherungsformel für die Bandbreite bw_{Hz} des Peaks (in Hz):

$$bw_{Hz} = -\frac{\ln r}{\pi} f_s \quad (2.7)$$

Da in dieser Arbeit ohnehin Filter betrachtet werden, bei denen viele (und nicht nur zwei) Polstellen miteinander wechselwirken, wird die Gleichung 2.6 zur Umrechnung der

Resonanzfrequenz φ in die Peak-Frequenz ψ jedoch hier nicht zur Anwendung kommen - wir benutzen die Resonanzfrequenz φ direkt. In den Abbildungen 2.2, 2.3 und 2.4 sind beispielhaft einige Betragsfrequenzgänge solcher Zweipol-Filter mit verschiedenen Winkeln und verschiedenen Radien der Polstellen dargestellt. Im ersten Plot ist der Winkel $\varphi = \pi/4$, dies entspricht bei einer Sample-Rate von 44.1 kHz einer Frequenz von 5512.5 Hz. Wir werden die zu modellierenden Formanten in der Regel im Frequenzbereich unterhalb von 5 kHz suchen - wie wir sehen, zeigen die Zweipol-Frequenzgänge in diesem Bereich (von φ) eine Art Tiefpass-Charakter. Wenn wir also beispielsweise einen Formanten durch ein solches Zweipol-Filter modellieren wollen, wird dies zu beachten sein. Wir werden jedoch gleich sehen, dass eine einfache 'Ein Formant entspricht einem Polpaar'-Zuordnung ohnehin nicht so ohne weiteres möglich ist.

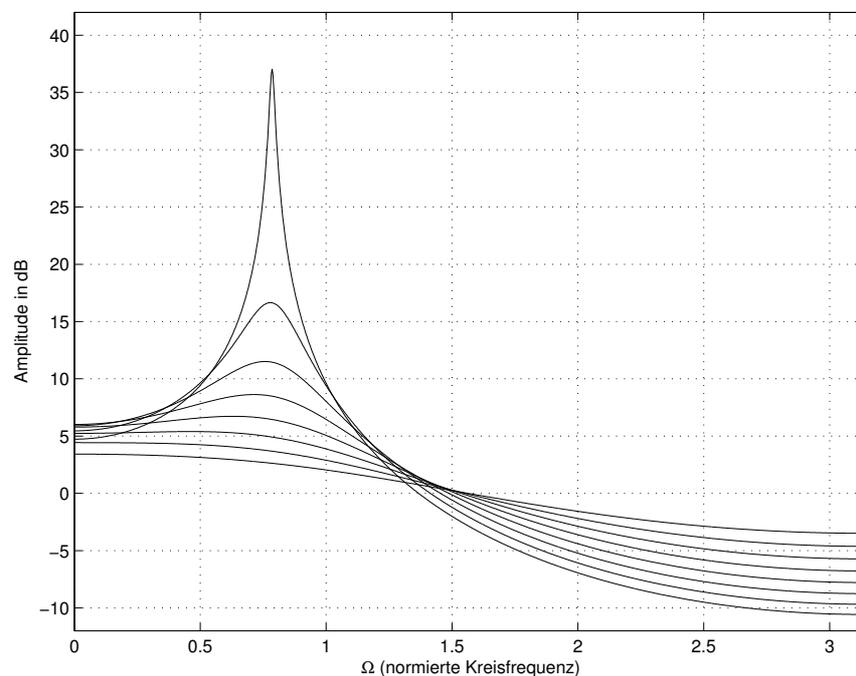


Fig. 2.2: Familie von Zweipol-Übertragungsfunktionen mit $\varphi = \frac{\pi}{4}$ und $r = 0.29, 0.39, 0.49, \dots, 0.99$

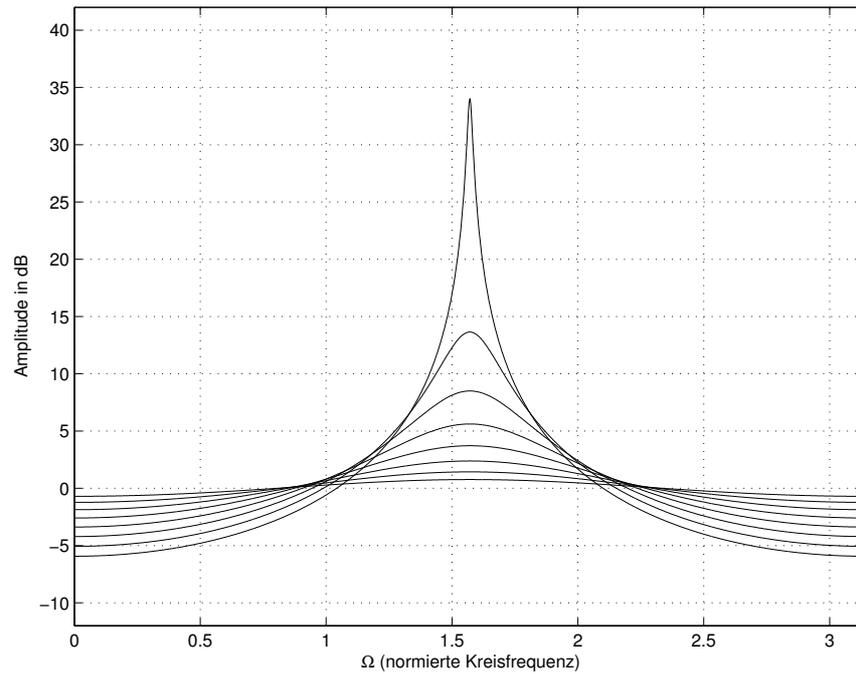


Fig. 2.3: Familie von Zweipol-Übertragungsfunktionen mit $\varphi = \frac{\pi}{2}$ und $r = 0.29, 0.39, 0.49, \dots, 0.99$

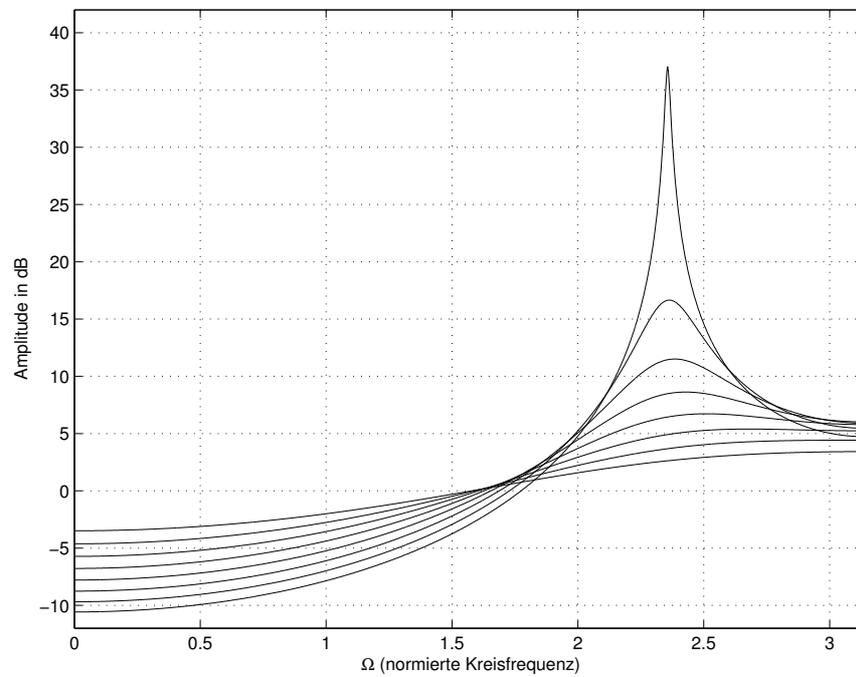


Fig. 2.4: Familie von Zweipol-Übertragungsfunktionen mit $\varphi = \frac{3\pi}{4}$ und $r = 0.29, 0.39, 0.49, \dots, 0.99$

Die Koeffizienten unseres Modellfilters liegen zunächst einmal in Direktform vor, d.h. wir haben die Rekursionskoeffizienten a_p und den Gain-Faktor G für ein Filter, das durch die Differenzgleichung

$$y[n] = Gx[n] - \sum_{p=1}^P a_p y[n-p]$$

bzw. durch die Transferfunktion:

$$H(z) = \frac{G}{1 + \sum_{p=1}^P a_p z^{-p}}$$

beschrieben ist. Was wir für eine intuitive Transformation von Formanten jedoch benötigen, sind die Resonanzfrequenzen und Bandbreiten der einzelnen Polpaare. Um diese zu finden, benötigen wir eine Zerlegung des Nennerpolynoms der Transferfunktion in seine einzelnen Linearfaktoren. Wir suchen demzufolge eine Zerlegung der Transferfunktion der Art:

$$\begin{aligned} H(z) &= G \frac{1}{1 + \sum_{p=1}^P a_p z^{-p}} \\ &= G \frac{1}{\prod_{p=1}^P (z - z_p)} \end{aligned}$$

wobei die z_p die komplexen Nullstellen des Polynoms $1 + \sum_{p=1}^P a_p z^{-p}$ sind. Um eventuell auftretender Verwirrung bezüglich der Notation zuvorzukommen: mit P ist die Ordnung des AR-Modells bezeichnet, und p wird als Index für die einzelnen Filterkoeffizienten benutzt. Der Buchstabe S hingegen bezeichnet die Anzahl der Zweipol-Stufen in unserem Modell, s wird als Index für eine solche Zweipol-Stufe benutzt, so dass $S = P/2$. Da die Polynom-Koeffizienten a_p reelle Zahlen sind, treten komplexe Nullstellen jeweils als konjugiert komplexe Paare auf, zusätzlich können einzelne reelle Nullstellen auftreten. In MatLab steht zur Berechnung von Nullstellen eines Polynoms aus dessen Koeffizienten die `roots`-Funktion zur Verfügung. Nachdem die Nullstellen des Nenners, und damit die Polstellen unseres Allpolmodells in der z -Ebene gefunden wurden, können wir jeweils zwei konjugiert komplexe Polstellen $r_s e^{j\varphi_s}$ und $r_s e^{-j\varphi_s}$ zu einem quadratischen Faktor der Form aus Gl. 2.2 mit reellen Koeffizienten a_{1s} und a_{2s} zusammensetzen. Dieser quadratische Term in der Transferfunktion kann dann durch eine Zweipol-Filterstufe realisiert werden, wobei die Resonanzfrequenz der Zweipol-Stufe gegeben ist mit φ_s (als normierte Kreisfrequenz), und die Bandbreite (in Hz) sich näherungsweise über Gleichung 2.7 aus dem Radius r_s bestimmen lässt. Eine Reihenschaltung solcher Stufen, wie sie in Abb. 2.1 dargestellt ist, führt dann wieder zu unserer Gesamt-Transferfunktion. Nun ist es natürlich möglich die Phasenwinkel φ_s (d.h. die Resonanzfrequenzen) und Radien r_s (d.h. die Bandbreiten) frei zu verändern. Das hier betrachtete Modell bietet uns also pro Filterstufe zwei unabhängig voneinander einstellbare Parameter - erstens die Resonanzfrequenz, die wir über den Winkel des Polpaares kontrollieren können und zweitens den Radius des Polpaares, der die Bandbreite und Intensität des spektralen Peaks bestimmt. In diesem Modell sind also Bandbreite und Intensität aneinander gekoppelt.

2.1.2 Polpaare und Formanten

Es wurde bereits angedeutet, dass es im allgemeinen leider nicht möglich ist, jeden Formanten mit einem bestimmten Polpaar zu identifizieren. In Abb. 2.5 ist das Spektrum und der Frequenzgang eines Allpolfilters, das über TE-LPC geschätzt wurde, dargestellt.

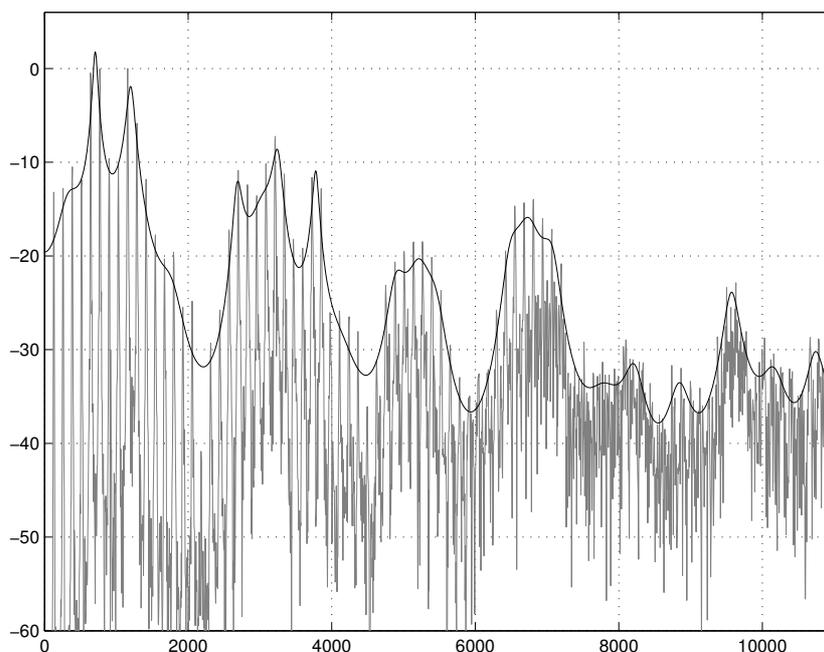


Fig. 2.5: FFT-Spektrum des Signals und Frequenzgang des Modellfilters (geschätzt über TE-LPC, LPC-Ordnung: $P = 100$, cepstrale Ordnung $P_c = 172$)

Der Plot ähnelt sehr stark dem Plot in Abb. 1.6, es gibt jedoch einige Abweichungen aufgrund des verwendeten Schätzverfahrens (in Abb. 1.6 war dies die konventionelle LPC-Analyse, hier jedoch das TE-LPC Verfahren). Aufgrund der geringen Grundfrequenz des Signals (ca. 130 Hz) treten die Vorteile des TE-LPC Verfahrens hier jedoch nicht so deutlich hervor. Es wurde wieder eine Ordnung von 100 für das Modellfilter gewählt - d.h. wir würden eine Reihenschaltung von 50 Zweipol-Filtern benötigen, um die spektrale Hüllkurve in dieser Genauigkeit darzustellen. Ein Blick auf die Verteilung der Polstellen des Modellfilters in der z -Ebene (Abb. 2.6) zeigt, dass es nicht möglich ist, einzelne Polpaare mit einzelnen Formanten zu identifizieren. Vielmehr ist es also offenbar so, dass immer mehrere Polpaare zu einem Formanten beitragen. Wir sehen eine ziemlich große Anzahl von Polen, die sich entlang des Einheitskreises verteilen und kaum eine intuitive Interpretation zulassen.

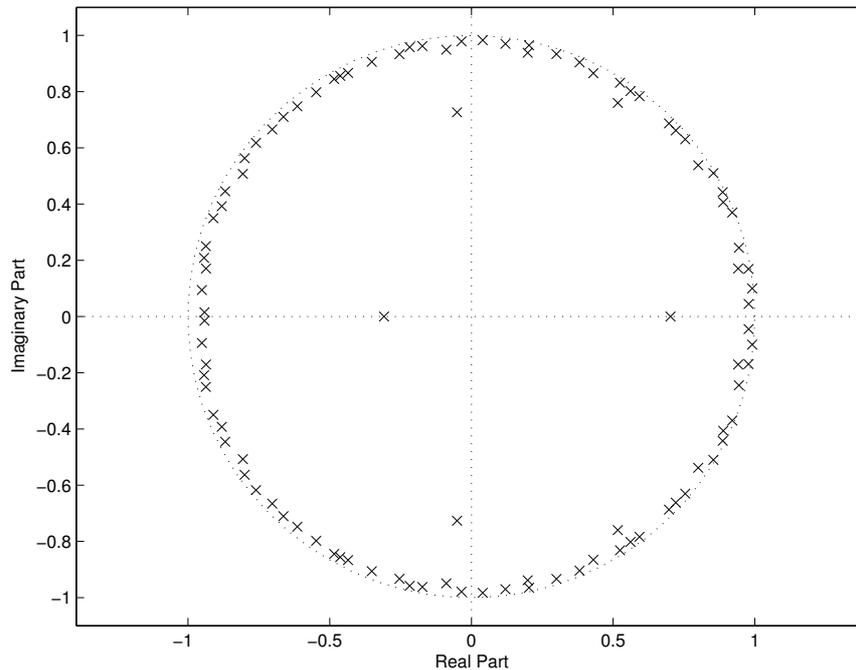


Fig. 2.6: Polstellen des Filters mit dem Frequenzgang aus Abb. 2.5

Nun ist es aber erklärtes Ziel dieser Arbeit, griffige Parameter zur Verfügung zu stellen, mit denen einzelne Formanten identifiziert und verändert werden können. Beispielsweise würde man als Musiker oder Sound-Designer gerne eine überschaubare Anzahl an Formant-Mittelfrequenzen und -Bandbreiten aus dem Signal gewinnen und diese verändern wollen. Wenn wir nun z.B. 5 Formanten mit jeweils einem Polpaar modellieren wollten, könnten wir die Ordnung für unser Modellfilter auf 10 reduzieren und damit das Modell zwingen, die spektrale Hüllkurve mit nur 5 Polpaaren zu approximieren. Dies würde dann zu einem Modell der spektralen Hüllkurve führen, welches mit 5 Zweipol-Filtern realisierbar wäre - jedes davon mit einer Resonanzfrequenz φ und einem Polradius r , welcher die Bandbreite bestimmt. Mit diesen Parametern könnte man dann intuitiv arbeiten. Unglücklicherweise würde der Frequenzgang dieses 10-Pol Modellfilters (bei einer Sample-Rate von $f_s = 44.1$ kHz) nicht mehr viel mit den gesuchten Formanten zu tun haben, da das Filter immer versucht, im gesamten Frequenzbereich eine gute Übereinstimmung zur spektralen Hüllkurve zu erreichen. In Abb. 2.7 ist dieses Problem deutlich zu erkennen. Es sei noch einmal daran erinnert, dass das Spektrum eigentlich bis 22.05 kHz reicht, hier jedoch zur besseren Erkennbarkeit nur der Bereich bis 11.025 kHz dargestellt ist - das Modellfilter jedoch versucht natürlich auch noch im hier nicht dargestellten Frequenzbereich zwischen 11.025 kHz bis 22.05 kHz eine Anpassung zu erreichen.

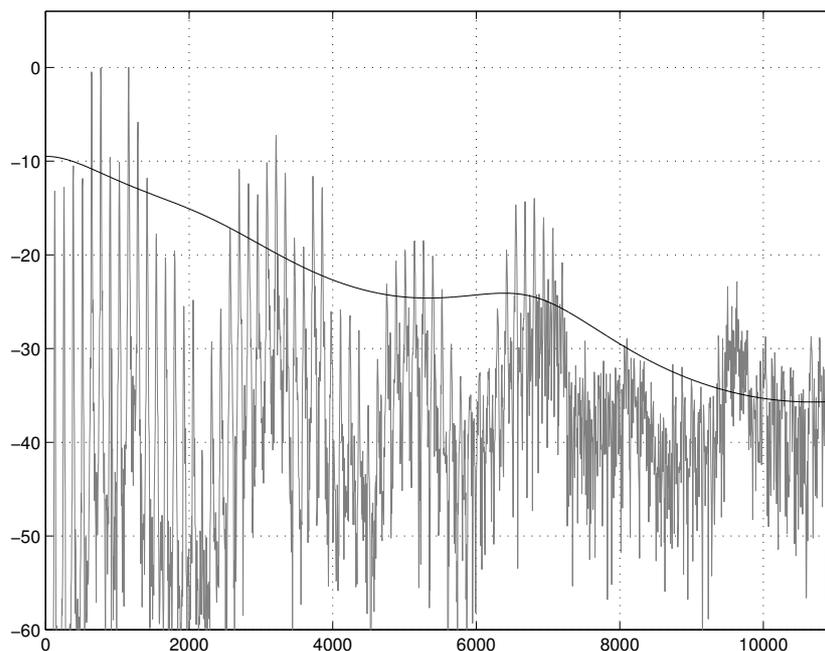


Fig. 2.7: FFT-Spektrum und spektrale Hüllkurve (geschätzt über TE-LPC, LPC-Ordnung: $P = 10$, cepstrale Ordnung $P_c = 172$)

2.1.3 Frequenzselektive Lineare Prädiktion und Unterabtastung

Das Problem ist also, dass wir zu viele Pole benötigen, um eine vernünftige Anpassung des Modellfilters an die spektrale Hüllkurve zu erreichen, und durch die vielen Pole (bzw. Polpaare) dann aber keinen intuitiv verständlichen Zugriff mehr auf die Modell-Parameter haben. Als Faustregel gilt, dass man für eine akzeptable Anpassung des Modellfilters an die spektrale Hüllkurve mindestens (von der Größenordnung her) so viele Pole benötigt, wie die Sample-Rate in kHz ausgedrückt hoch ist - bei einer Sample-Rate von 44.1 kHz also um die 44 Pole (d.h. 22 Polpaare). In obiger Betrachtung wurde diese Zahl zwar um mehr als den Faktor 2 übertroffen, dennoch wären auch 22 Polpaare zu viel, um eine eindeutige Korrespondenz zwischen einzelnen Formanten und einzelnen Polpaaren herzustellen.

Bei einer geringen Modell-Ordnung (wie z.B. $P = 10$) auf der anderen Seite, wurde die mangelnde Übereinstimmung zwischen spektraler Hüllkurve des Signals und Frequenzgang des Modellfilters im interessierenden Frequenzbereich als Problem identifiziert. Und genau hier setzt die Idee der frequenzselektiven linearen Prädiktion an. Anstatt eine Anpassung des Modellfilters im gesamten Frequenzbereich zu fordern, wird hier eine Anpassung nur in einem interessierenden Frequenzbereich $\Omega_u < \Omega < \Omega_o$ gefordert. Dabei sind Ω_u und Ω_o die untere und obere normierte Kreisfrequenz, bei denen eine gute Anpassung gefordert wird. Das allgemeine Verfahren zur Bestimmung des Modellfilters ist in [7] und [8] beschrieben. Innerhalb dieser Arbeit soll es genügen es einen einfachen Spezialfall zu betrachten, nämlich den, dass die untere Grenzfrequenz $\Omega_u = 0$ ist - das bedeutet der interessierende Frequenzbereich wird nur nach oben, nicht aber nach unten beschnitten. In diesem Fall können wir das Signal zunächst tiefpassfiltern und dann unterabtasten,

um dann auf dem unterabgetasteten Signal dieselben Verfahren anzuwenden wie bisher. Wenn uns beispielsweise eine Anpassung des Frequenzgangs vor allem im Bereich bis zu 5 kHz interessiert, dann bietet es sich an, das Signal bei 5 kHz steilflankig zu filtern, dann auf 10 kHz Sample-Rate unterabzutasten und auf diesem unterabgetasteten Signal ein Modellfilter zu bestimmen - und hier können wir jetzt tatsächlich mit einer Ordnung von $P = 10$ ein akzeptables Modellfilter bestimmen. Da es für die Unterabtastung jedoch generell vorteilhaft ist, wenn die neue Sample-Rate sich aus der alten durch Teilung mit einer ganzen Zahl ergibt, wurde als neue Sample-Rate hier 11.025 kHz gewählt. Mit dieser Sample-Rate liegt die Nyquist-Frequenz bei 5.5125 kHz. In den Abbildungen 2.8 und 2.9 sind der Frequenzgang und die Polverteilung eines 10-Pol Filters dargestellt, welches via TE-LPC auf dem 4-fach unterabgetasteten Signal geschätzt wurde.

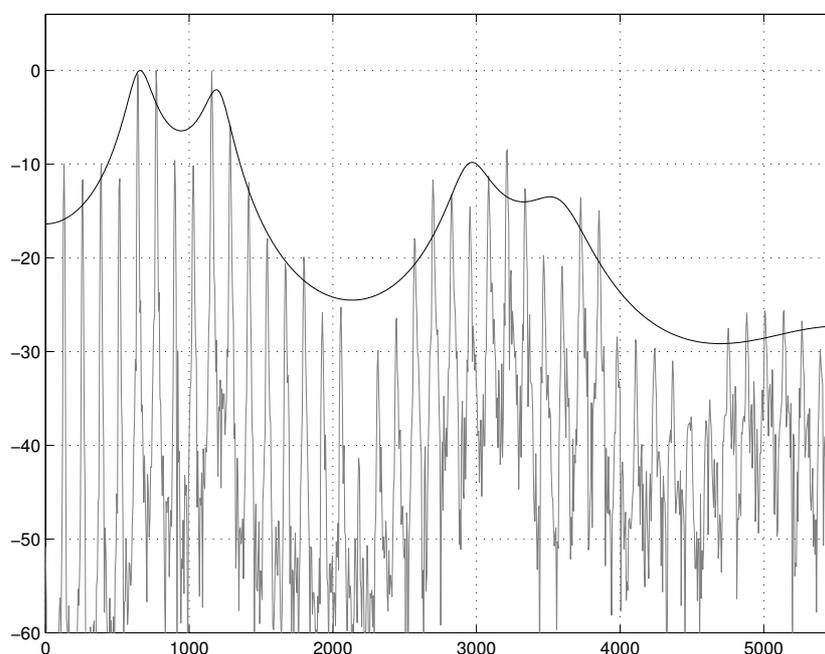


Fig. 2.8: FFT-Spektrum des unterabgetasteten Signals und Frequenzgang des 10-Pol-Modellfilters (geschätzt über TE-LPC)

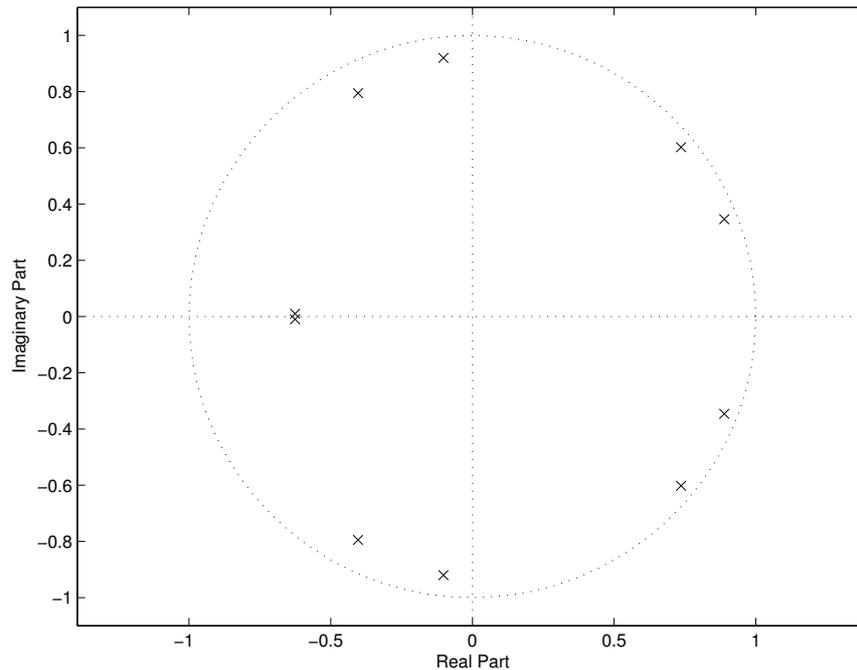


Fig. 2.9: Polstellen des Filters mit dem Frequenzgang aus Abb. 2.8

Wir sehen einerseits, dass die spektrale Hüllkurve durch den Frequenzgang des Modellfilters einigermaßen zufriedenstellend nachgebildet wird, und andererseits, dass die Verteilung der Polstellen tatsächlich eine Interpretation der Polpaare als Formantfilter zulässt.

2.1.4 Transformation der Formanten

In diesem Abschnitt soll untersucht werden, welche Transformationsmöglichkeiten sich aus diesem Modell ergeben. Als ersten Schritt einer Formant-Transformation müssen wir zunächst die vorhandenen Formanten aus dem Signal entfernen. Dies geschieht, indem wir das gefundene Modellfilter invertieren und das Signal mit diesem inversen Filter filtern - in der LPC-Terminologie heißt das, wir wenden das Prediction-Error-Filter auf das Signal an. Wir erhalten dadurch eine geweißte Version des Signals. Wenn wir auf dieses geweißte Signal wieder das Modellfilter anwenden, erhalten wir das Originalsignal zurück (das Modellfilter ist ja das inverse Prediction-Error-Filter). Aber vorher wollen wir natürlich das Modellfilter so modifizieren, dass bei der Anwendung des modifizierten Modellfilters bestimmte Formant-Parameter verändert werden. Ein Blick auf die möglichen Frequenzgänge einer einzelnen Zweipol-Stufe (Abbildungen 2.2, 2.3 und 2.4) zeigt, dass die Veränderung der Parameter einer Filterstufe sich immer im gesamten Frequenzbereich auswirken wird. Die dargestellten Frequenzgänge sind mit logarithmischer Amplitude aufgetragen (in dB). Um den (logarithmierten) Frequenzgang einer Reihenschaltung solcher Filterstufen zu erhalten, müsste man daher S solcher Kurven miteinander addieren (der Logarithmus transformiert die Multiplikation der Frequenzgänge in ein Addition). Es sitzt demzufolge jeder Gipfel - bildlich gesprochen - auf den Flanken aller anderen Gipfel. Und da diese Flanken sich über den gesamten Frequenzbereich erstrecken ohne wirklich flach

zu werden (außer bei 0 Hz und bei $f_s/2$), bewirkt eine Verschiebung oder Bandbreitenänderung eines solchen Peaks immer eine Veränderung des gesamten Frequenzgangs.

In der Praxis ist es nicht zwingend erforderlich, das modifizierte Modellfilter tatsächlich in der Form einer Reihenschaltung von Zweipol-Resonatoren zu implementieren. Alternativ dazu kann auch aus den neuen Polstellen wieder die Direktform errechnet werden. Dabei muss natürlich darauf geachtet werden, die konjugiert komplexen Polstellen immer nur paarweise zu modifizieren. In MatLab beispielsweise erfolgt die Umrechnung von Polstellen in Direktform-Koeffizienten über die `poly`-Funktion.

Die einzelnen Schritte, um eine Formant-Transformation durchzuführen sind hier noch einmal zusammengefasst:

- Das Signal wird zunächst um den Faktor 4 unterabgetastet (natürlich mit entsprechendem Anti-Aliasing-Tiefpass).
- Auf dem unterabgetasteten Signal wird das optimale Pre-Emphase-Filter bestimmt und auf das Signal angewandt.
- Mittels TE-LPC wird ein AR-Modell für das resultierende Signal bestimmt.
- Das (unterabgetastete, pre-emphase-gefilterte) Signal wird mit dem inversen Modell-Filter geweißt.
- Die Polstellen des AR-Modells werden bestimmt und in Formant-Mittenfrequenzen und -Bandbreiten umgerechnet (Gleichungen 2.4 und 2.7).
- Die gewünschten Transformationen werden auf die Formanten angewandt, und aus den neuen Formantfrequenzen und -Bandbreiten werden die neuen Polstellen bestimmt.
- Die neuen Polstellen werden paarweise zu quadratischen Termen mit reellen Koeffizienten zusammengefasst. Diese Koeffizienten können für ein Synthesefilter mit der Struktur in Abb. 2.1 benutzt werden. Alternativ dazu können aus den neuen Polstellen auch wieder Koeffizienten für ein Synthesefilter in Direktform berechnet werden.
- Das Synthesefilter wird auf das geweißte Signal angewandt.
- Das inverse Pre-Emphase Filter wird auf das Signal angewandt.

Um wieder ein Signal der ursprünglichen Sample-Rate zu erhalten, kann das transformierte Signal jetzt wieder auf die ursprüngliche Sample-Rate interpoliert werden. Dieses interpolierte Signal ist natürlich nach wie vor bandbegrenzt auf den Bereich bis zur alten (unterabgetasteten) Nyquist-Frequenz. Um wieder ein Signal der vollen Bandbreite zu erhalten, ist es möglich, die Signalanteile des originalen (nicht transformierten) Signals im verlorengegangenen Frequenzbereich wieder hinzuzufügen. Dazu muss nach dem Anti-Alias-Filter für die Unterabtastung ein Residualsignal bestimmt werden, indem der tiefpassgefilterte Anteil von Originalsignal abgezogen wird - es bleiben die Signalanteile oberhalb der unterabgetasteten Nyquist-Frequenz übrig. Wir nehmen hier an, dass das Anti-Alias-Filter ein Nullphasenfilter war, so dass die Subtraktion im Zeitbereich tatsächlich

zum gewünschten Ergebnis führt. Für Nicht-Echtzeit Verarbeitung stellt diese Forderung kein Problem dar, denn hier können Nullphasenfilter entweder durch nichtkausale FIR-Filter oder bidirektionale IIR-Filter realisiert werden. Nach der Interpolation des transformierten Signals auf die ursprüngliche Sample-Rate kann dieses Residualsignal wieder dazuaddiert werden. Das Ergebnis ist ein natürlicher klingendes transformiertes Signal.

Wenn nur einzelne Formanten transformiert werden sollen, ist es übrigens nicht nötig, das Signal komplett zu weißen. Es genügt in diesem Falle, wenn man nur dasjenige Teilfilter invertiert, das mit dem zu transformierenden Formanten assoziiert ist. Man kann sich dies durch folgende Betrachtung recht einfach klarmachen: Wir bezeichnen wie bisher mit $H(z)$ die Transferfunktion unseres Modellfilters, mit $H_s(z)$ die Transferfunktion des s -ten Teilfilters in der Reihenschaltung, mit $H'_s(z)$ das modifizierte s -te Teilfilter und schließlich mit $H'(z)$ das modifizierte gesamte Filter. Die Transferfunktion zur Formanttransformation nennen wir $T(z)$ - sie setzt sich aus der inversen Transferfunktion des Modellfilters (whitening) und der neuen Transferfunktion mit den modifizierten Formanten in folgender Weise zusammen:

$$T(z) = \frac{H'(z)}{H(z)} = \frac{\prod_{s=1}^S H'_s(z)}{\prod_{s=1}^S H_s(z)}$$

Alle Faktoren, die im oberen und unteren Produkt identisch sind, kürzen sich einfach heraus. Wenn also nur der durch Stufe s modellierte Formant verändert werden soll, genügt es $T(z) = H'_s(z)/H_s(z)$ auf das Signal anzuwenden. Das Filter $1/H_s(z)$ kann als eine Art partielles Whitening-Filter aufgefasst werden, welches nur denjenigen Anteil von $H(z)$ aus dem Signal heraus nimmt, der durch $H_s(z)$ verursacht wurde.

2.1.5 Implementation und Praxistest

In diesem Abschnitt wollen wir nun also die Theorie in die Praxis umsetzen, und auf unser Beispielsignal bestimmte Formant-Transformationen anwenden. Die zu den hier dargestellten Experimenten gehörenden MatLab-Implementationen sind im Anhang zu finden. Zunächst soll die Mittenfrequenz des zweiten Formanten (also der Formant im Bereich von 1.2 kHz) um den Faktor 1.5 nach oben verschoben werden, alle anderen Formanten sollen da bleiben, wo sie sind. Das Ergebnis dieser Transformation ist in Abb. 2.10 dargestellt. In Abb. 2.11 sind der Frequenzgang des originalen und des modifizierten Modellfilters gegenübergestellt (wobei die Kurven jeweils so normiert wurden, dass das Maximum bei 0 dB liegt).

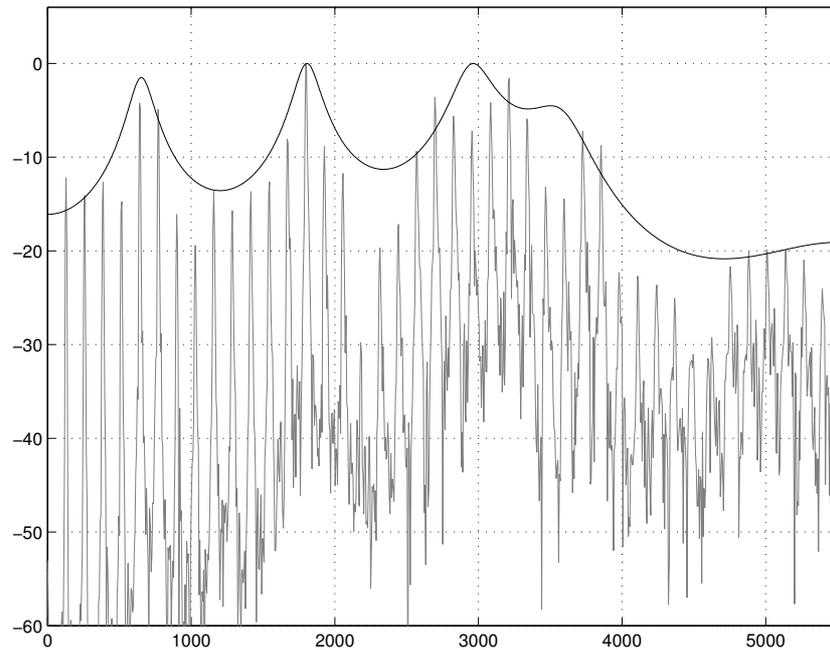


Fig. 2.10: FFT-Spektrum des transformierten Signals und Frequenzgang des modifizierten 10-Pol-Modellfilters (2. Formantfrequenz um Faktor 1.5 erhöht)

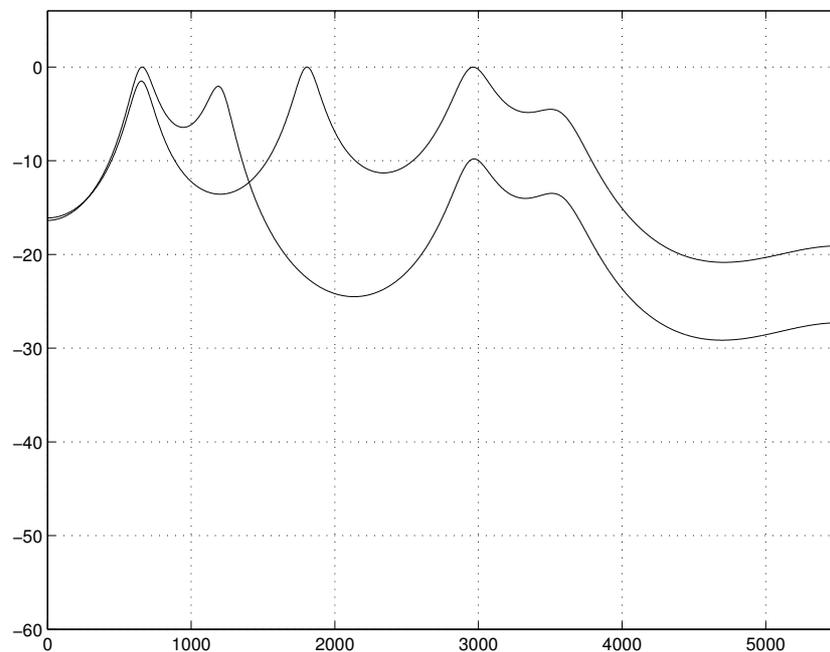


Fig. 2.11: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formantfrequenz um Faktor 1.5 erhöht)

Deutlich zu sehen ist die globale Auswirkung dieser Formantverschiebung. Durch die Verschiebung der Mittenfrequenz nach rechts erfährt der gesamte Frequenzbereich rechts des

Formanten eine relative Anhebung und der Bereich links davon eine (nicht ganz so deutliche) relative Absenkung im Vergleich zum Originalsignal.

Als zweites Experiment soll die Bandbreite des zweiten Formanten schmaler gemacht werden. Es wurde ein Skalierungsfaktor von 0.25 gewählt. Die Ergebnisse sind in den Abbildungen 2.12 und 2.13 dargestellt.

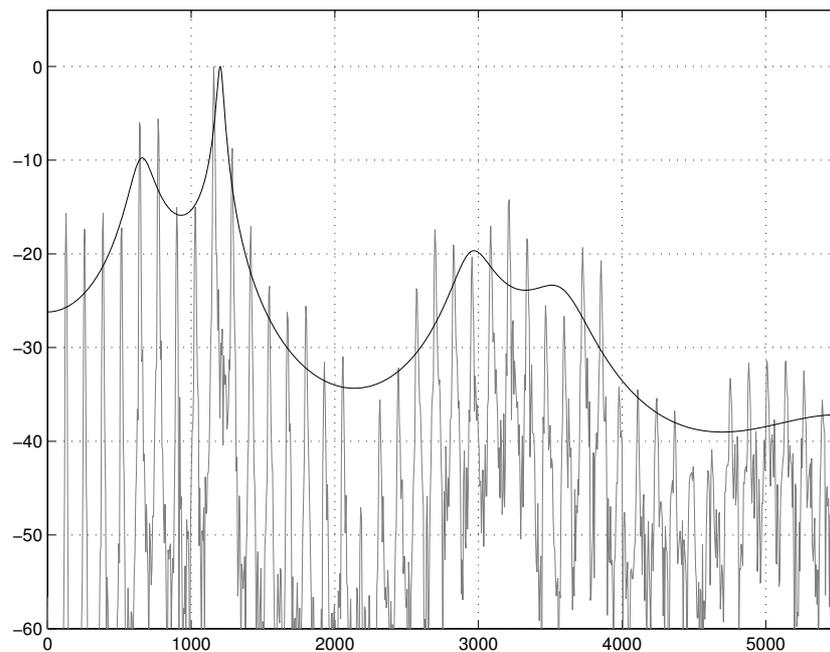


Fig. 2.12: FFT-Spektrum des transformierten Signals und Frequenzgang des modifizierten 10-Pol-Modellfilters (2. Formant-Bandbreite um Faktor 0.25 gesenkt)

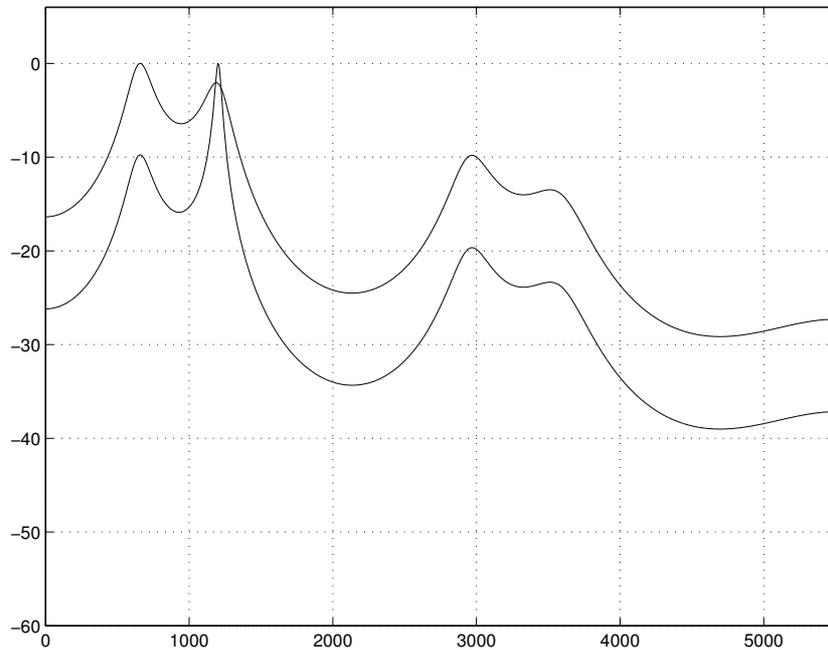


Fig. 2.13: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formant-Bandbreite um Faktor 0.25 gesenkt)

Hier wird die Kopplung von Formant-Bandbreite und -Intensität deutlich sichtbar - das zweite Peak im modifizierten Modellfrequenzgang ist jetzt sowohl schmaler als zuvor (was ja auch beabsichtigt war) als auch höher als zuvor (dies der unvermeidliche Nebeneffekt). Nun ja, welcher Effekt von den beiden der Haupt- und welcher der Nebeneffekt ist, liegt natürlich im Auge des Betrachters bzw. ist abhängig von der jeweiligen Zielsetzung.

2.1.6 Vor- und Nachteile des Modells

Vorteile

- Die Modellparameter sind relativ leicht zu bestimmen (über AR-Modellierung und Faktorisierung des AR-Modells)
- Bei der menschlichen Stimme ist das Modell physikalisch plausibel, da eine Reihenschaltung von Resonatoren akustisch einer Röhre mit wechselnden Rohrdurchmessern entspricht.
- Bei Transformationen, die nur einzelne Filterstufen betreffen, ist kein komplettes Whitening des Signals nötig - es genügt hier, die entsprechenden Teilfilter zu invertieren.

Nachteile

- Formant-Intensität und -Bandbreite sind aneinander gekoppelt (beides wird durch den Polradius r gleichzeitig bestimmt).
- Jede einzelne Filterstufe beeinflusst den Frequenzgang immer global, d.h. wenn beispielsweise die Resonanzfrequenz eines Zweipol-Resonators verschoben wird, hat dies Auswirkungen auf den gesamten Frequenzgang und nicht nur auf die Umgebung der Resonanzfrequenz.
- Um intuitive Transformationen durchführen zu können, muss die Modellordnung gering gewählt werden. Um bei geringer Modellordnung jedoch noch eine gute Anpassung des Filters an die tatsächlichen Formanten hinzubekommen, muss im allgemeinen bei niedrigen Sample-Raten gearbeitet werden.

2.2 Parallele Allpolstruktur

2.2.1 Das Modell

In diesem Abschnitt betrachten wir eine Parallelschaltung von Filtern zweiter Ordnung als Modell für unser Synthesefilter. Wie wir noch sehen werden nimmt unser Modell die in Abb. 2.14 dargestellte Form an.

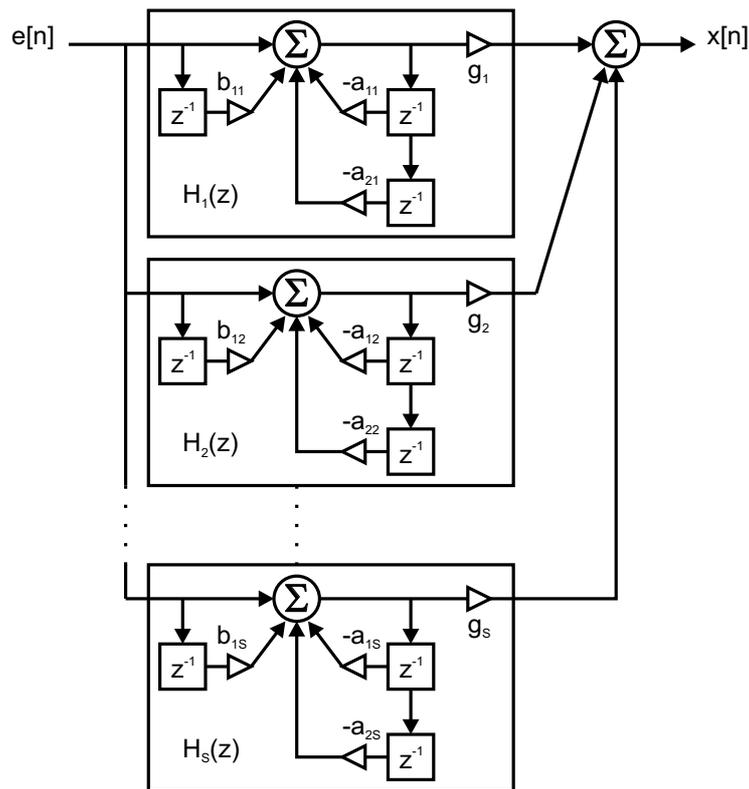


Fig. 2.14: Parallele Verschaltung von S Filtern 2. Ordnung zu einer Allpolstruktur

Wieder gehen wir von einem Allpolmodell in Direktform aus, welches wir über TE-LPC bestimmen. Bei einer parallelen Verschaltung von Filtern ergibt sich die gesamte Transferfunktion als die Summe der einzelnen (komplexen) Teil-Transferfunktionen. Wir suchen zunächst eine Zerlegung unseres Modellfilters der Art:

$$\begin{aligned} H(z) &= \frac{G}{1 + \sum_{p=1}^P a_p z^{-p}} \\ &= \sum_{s=1}^S \frac{Num_s}{Den_s} \end{aligned}$$

wobei mit Num_s und Den_s Zähler und Nenner der Filterstufe mit dem Index s bezeichnet werden sollen. Für diese einzelnen Zähler und Nenner wollen wir fordern, dass sie höchstens zweiter Ordnung sein sollen. Um die Koeffizienten für diese Parallelstruktur

zu finden, muss die Transferfunktion in Partialbrüche höchstens zweiter Ordnung zerlegt werden. Es zeigt sich, dass bei einer solchen Zerlegung quadratische Terme in z nur in den Nennern entstehen, für die Zähler ergeben sich lineare Terme in z - dies bedeutet, dass wir pro Teilfilter zwei konjugiert komplexe Polstellen und eine reelle Nullstelle erhalten. Eine solche Partialbruchzerlegung ist immer dann problemlos möglich, wenn die Ordnung des Nenners größer als die Ordnung des Zählers ist, und die Polstellen paarweise verschieden sind (keine mehrfachen Polstellen). Da wir es ja mit einem Allpolmodell zu tun haben, ist die erste Bedingung in jedem Fall erfüllt, denn die Ordnung des Zählers ist hier ja Null. Dass die zweite Bedingung erfüllt ist, wollen wir ebenfalls annehmen, da es extrem unwahrscheinlich ist, dass unser Modellfilter mehrere Polstellen auf exakt dem gleichen Punkt platziert. Wenn die Ordnung des Zählers größer als die des Nenners ist, kommt ein zusätzlicher paralleler FIR-Teil hinzu, bei mehrfachen Polstellen kommen parallele Teilfilter höherer Ordnung hinzu. Die Partialbruchzerlegung, so wie sie z.B. in der Funktion `residuez` aus der Signal-Processing Toolbox von MatLab implementiert ist, liefert die komplexen Zähler r_i und komplexen Polstellen p_i von Partialbrüchen in einer Form, so dass die Gesamt-Transferfunktion durch folgende Summe zusammengesetzt ist:

$$H(z) = \sum_{s=1}^S \frac{r_s}{1 - p_s z^{-1}}$$

Da die einzelnen Zähler r_s und Polstellen p_s komplex sind, ist auch das Ausgangssignal eines solchen Teilfilters komplex. Da unser Modellfilter aber reelle Koeffizienten hat, existiert zu jedem Partialbruch mit den Werten (r_s, p_s) ein zweiter Partialbruch mit den jeweils konjugiert komplexen Werten (r_s^*, p_s^*) . Da ja alle Teilfilter parallel zueinander sind, können wir die Transferfunktionen dieser beiden Einpol-Teilfilter addieren:

$$H_s(z) = \frac{r_s}{1 - p_s z^{-1}} + \frac{r_s^*}{1 - p_s^* z^{-1}}$$

und diese Summe umformen zu einer Transferfunktion zweiter Ordnung der Form:

$$H_s(z) = \frac{b'_{0s} + b'_{1s} z^{-1}}{1 + a_{1s} z^{-1} + a_{2s} z^{-2}} \quad (2.8)$$

mit den reellen Koeffizienten:

$$\begin{aligned} b'_{0s} &= 2\operatorname{Re}\{r_s\} \\ b'_{1s} &= -2\operatorname{Re}\{r_s p_s^*\} \\ a_{1s} &= -2\operatorname{Re}\{p_s\} \\ a_{2s} &= |p_s|^2 \end{aligned}$$

Die Summe von S solcher Transferfunktionen ergibt dann wieder die gesamte Transferfunktion. Im Gegensatz zu der seriellen Filterstruktur aus Filtern mit Transferfunktionen der Form 2.2 haben die Teilfilter in diesem Modell zwei Freiheitsgrade mehr, nämlich die Koeffizienten b'_0 und b'_1 . Diese beiden zusätzlichen Parameter bestimmen einerseits die Ausgangslautstärke des Teilfilters, denn wenn man beide einfach mit einem Faktor c skaliert, wird die Ausgangsamplitude um denselben Faktor c lauter. Andererseits bewirkt die zusätzliche Nullstelle (bei $z = -b'_1/b'_0$) eine zusätzliche Hochpass- bzw. Tiefpasswirkung

und eine Veränderung des Phasenfrequenzganges. Die b -Koeffizienten wurden deshalb mit Strichen versehen, da es sinnvoller ist, die Transferfunktion in folgender Form zu schreiben:

$$H_s(z) = g_s \frac{1 + b_{1s}z^{-1}}{1 + a_{1s}z^{-1} + a_{2s}z^{-2}} \quad (2.9)$$

mit

$$\begin{aligned} g_s &= b'_{0s} \\ b_{1s} &= b'_{1s}/b'_{0s} \end{aligned}$$

Es wurde hier einfach b'_{0s} im Zähler ausgeklammert, in g_s umbenannt und vor den Bruch geschrieben. Damit können wir nämlich die jeweiligen g_s direkt als Gain-Faktoren der einzelnen Teilfilter interpretieren, während der b_{1s} -Koeffizient die Position der Nullstelle bestimmt (sie liegt genau bei $z_0 = -b_{1s}$). Das Blockschaltbild unserer Filterstruktur nimmt daher die in Abb. 2.14 dargestellte Form an. Um sich die Wirkung der Nullstelle in den einzelnen Teilfiltern vorstellen zu können, wurde in Abb. 2.15 eine Familie von Amplitudenfrequenzgängen eines Filters mit einer einzelnen reellen Nullstelle dargestellt.

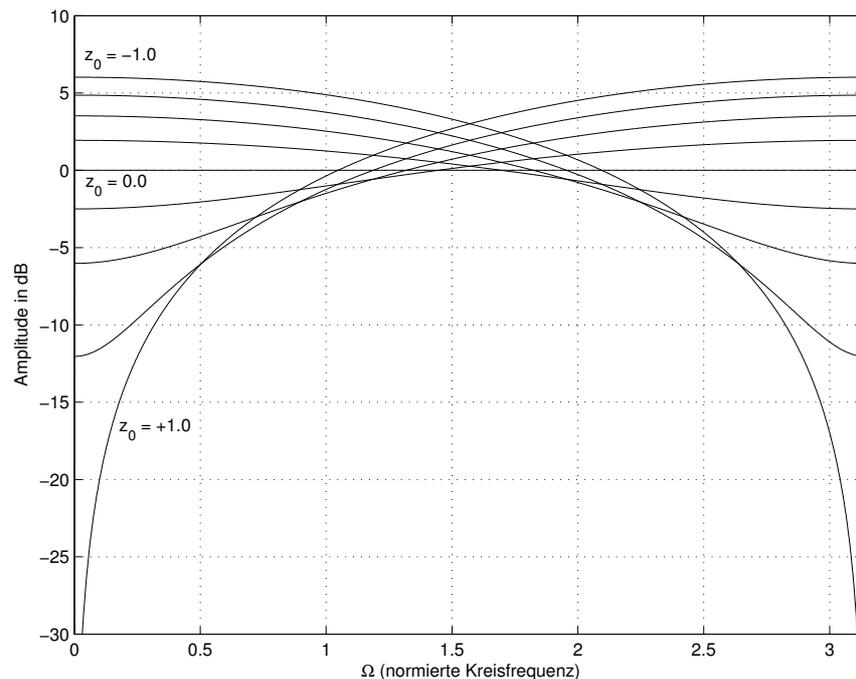


Fig. 2.15: Familie von Amplitudenfrequenzgängen einer Nullstelle $z_0 = -1.0, -0.75, -0.5, \dots, +1.0$

Da man sich das Zweipol-Filter mit Nullstelle als Reihenschaltung eines Zweipol-Filters und eines solchen Nullstellenfilters vorstellen kann, ergibt sich der gesamte Amplitudenfrequenzgang eines Zweipol-Ein-Nullstellen Filters als Multiplikation des Amplitudenfrequenzgangs eines Zweipol-Filters (so wie sie im vorherigen Abschnitt betrachtet wurden) mit dem Amplitudenfrequenzgang des Nullstellenfilters. Da die Plots hier jedoch die Amplitude logarithmisch auftragen, müsste man die Kurven addieren. Wenn die Nullstelle

genau bei $z_0 = 0$ liegt, verhält sie sich neutral, d.h. der Frequenzgang ist identisch Eins. Wandert die Nullstelle nach rechts in Richtung $z_0 = 1$, ist im Amplitudenfrequenzgang eine Hochpasswirkung zu beobachten. Dies ist einleuchtend, da der Punkt $z = 1$ dem Gleichspannungsanteil entspricht. Wandert die Nullstelle dagegen nach links in Richtung $z_0 = -1$, gibt es eine Tiefpasswirkung ($z = -1$ entspricht ja der Nyquist-Frequenz). Im Gegensatz zu den Zweipol-Filtern aus dem vorherigen Abschnitt, bekommen wir demzufolge noch eine Hoch- bzw. Tiefpasswirkung hinzu. Eine weitere Auswirkung der Nullstelle ist eine Veränderung des Phasenfrequenzgangs. Da die Ausgangssignale der Teilfilter ja addiert werden, sind die Phasenbeziehungen zwischen den Teilsignalen natürlich auch von großer Bedeutung. In Abb. 2.2.1 ist daher eine Familie von Phasenfrequenzgängen von Zweipol-Filtern mit Nullstelle dargestellt. Es ist eventuell im Plot nicht zweifelsfrei zu erkennen, aber die Phasengänge liegen übereinander ohne sich bei der Resonanzfrequenz zu überschneiden.

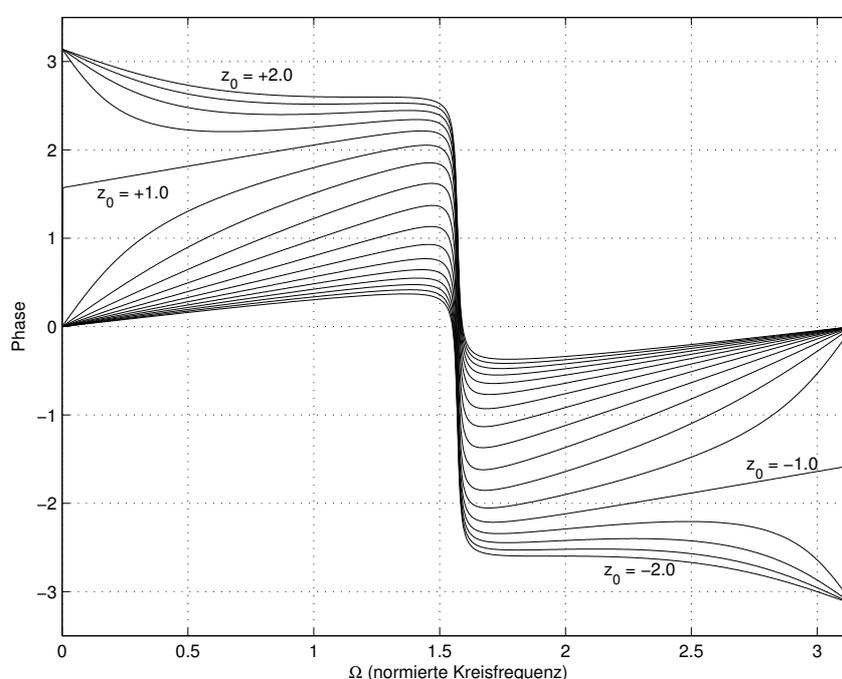


Fig. 2.16: Familie von Zweipol-Phasenfrequenzgängen mit einer Nullstelle z_0 , $\varphi = \frac{\pi}{2}$, $r = 0.99$, $z_0 = -2.0, -1.75, -1.5, \dots, +2.0$

2.2.2 Zusammenhang zu Formantwellenformen (FOF's)

Die einzelnen parallelen Teilfilter haben also jeweils eine Transferfunktion der Form:

$$\begin{aligned} H(z) &= g \frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \\ &= g \left(\frac{r}{1 - p z^{-1}} + \frac{r^*}{1 - p^* z^{-1}} \right) \end{aligned}$$

wobei sich in der zweiten Zeile die Zähler ($r = Re^{j\varphi}$, $r^* = Re^{-j\varphi}$) und Polstellen ($p = Pe^{j\omega}$, $p^* = Pe^{-j\omega}$) durch eine Partialbruchzerlegung des Bruchs (ohne den Gain-Faktor)

ergeben. Auf den Index s für das Teilfilter wird in diesem Abschnitt verzichtet, da wir hier sowieso nur ein einzelnes Teilfilter betrachten. Wenn wir als Eingangssignal für unsere Parallelstruktur einen Einheitsimpuls annehmen, dann ergibt sich das zugehörige Ausgangssignal als Summe der Impulsantworten der einzelnen Teilfilter. Die Impulsantwort eines solchen Teilfilters ist gegeben mit (siehe Anhang B.1):

$$h[n] = 2gR \cdot P^n \cdot \cos(\omega n + \varphi)$$

Diese Gleichung können wir als eine (abgetastete) exponentiell abklingende Sinusschwingung mit der normierten Kreisfrequenz ω auffassen, die im allgemeinen die Form:

$$x[n] = A \cdot e^{-\alpha n} \cdot \sin(\omega n + \phi)$$

hat. Um diesen allgemeinen Ausdruck mit unserer Impulsantwort in Übereinstimmung zu bringen, stellen wir zunächst fest, dass $\cos(\varphi) = \sin(\varphi + \pi/2)$ gilt. Wenn wir in der allgemeinen Form also:

$$\phi = \varphi + \frac{\pi}{2} \quad (2.10)$$

einsetzen, stimmt schon mal der Schwingungsterm in beiden Formen überein (bezüglich der Startphase, bezüglich der Frequenz sowieso). Die Gesamtamplitude A im allgemeinen Ausdruck können wir mit dem Faktor $2gR$ identifizieren:

$$A = 2gR \quad (2.11)$$

Den exponentiellen Abklingterm $e^{-\alpha n}$ schließlich können wir mit dem Term P^n in Übereinstimmung bringen, indem wir die Regel $P^n = e^{n \ln(P)}$ anwenden und

$$\alpha = -\ln(P) \quad (2.12)$$

setzen. Der Dämpfungsfaktor α ist übrigens direkt proportional zur Bandbreite des Filters. In Gleichung 2.7 wurde als Näherung für die Bandbreite eines Zweipol-Filters mit dem Polpaar-Radius P die Näherung:

$$bw_{Hz} = -\frac{\ln(P)}{\pi} f_s \quad (2.13)$$

angegeben. Wenn wir diese Formel nach $-\ln(P)$ umstellen, und das Ergebnis in die Gleichung für α einsetzen, erhalten wir:

$$bw_{Hz} = \frac{\alpha f_s}{\pi} \quad (2.14)$$

Wir erhalten also als Impulsantwort eine exponentiell abklingende Sinusschwingung:

$$\boxed{h[n] = A \cdot e^{-\alpha n} \cdot \sin(\omega n + \phi)} \quad (2.15)$$

deren Parameter A, α, ω, ϕ aus den Parametern der Transferfunktion in Partialbruchdarstellung g, R, φ, P, ω berechnet werden können. Die Parameter R, φ, P, ω selbst müssen unter Umständen erstmal durch die Partialbruchzerlegung aus den Filterkoeffizienten b_1, a_1, a_2 berechnet werden. Dabei hängen der Betrag P und Winkel ω des Polpaars

(p, p^*) mit den Koeffizienten a_1, a_2 im Nenner über Gleichung 2.5 zusammen, die hier nach den Parametern aufgelöst wird:

$$\begin{aligned} P &= \sqrt{a_2} \\ \omega &= \arccos\left(-\frac{a_1}{2P}\right) \end{aligned} \quad (2.16)$$

Für Betrag R und Winkel φ des Zählerpaares ergibt sich:

$$\begin{aligned} R &= \sqrt{\Re^2\{r\} + \Im^2\{r\}} \\ \varphi &= \arctan\left(\frac{\Im\{r\}}{\Re\{r\}}\right) \end{aligned}$$

wobei mit $\Re\{r\}$ und $\Im\{r\}$ Real-, bzw. Imaginärteil der komplexen Zahl r gemeint sind. Diese ergeben sich zu (siehe Anhang B.1):

$$\begin{aligned} \Re\{r\} &= \frac{1}{2} \\ \Im\{r\} &= -\frac{\sin(2\omega) + 2(b_1/P)\sin(\omega)}{2(1 - \cos(2\omega))} \end{aligned}$$

In Abb. 2.17 ist ein Beispiel einer solchen Impulsantwort dargestellt.

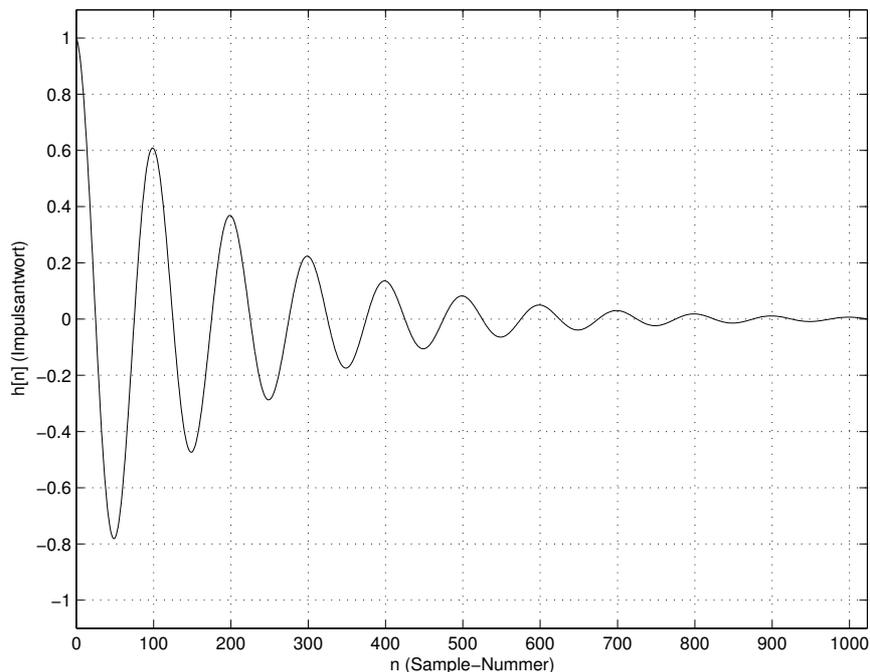


Fig. 2.17: Impulsantwort eines Teilfilters bzw. einer Formantwellenform (FOF) mit $A = 1$, $\alpha = 0.005$, $\omega = \pi/50$, $\phi = \pi/2$

Wenn ein Filter mit solch einer Impulsantwort durch einen Impulskamm als Eingangssignal angeregt wird, entsteht am Ausgang des Filters eine Wellenform, die sich durch Überlagerung der Antworten auf die einzelnen Impulse ergibt. Diese Wellenform hat dann

im Spektrum einen Formanten bei der Resonanzfrequenz ω des Filters und wird aus diesem Grunde als Formantwellenform bezeichnet. Im Französischen heißen diese Formantwellenformen 'Forme d'Onde Formantique', abgekürzt FOF, was sich als gängige Bezeichnung eingebürgert hat. Für Näheres zum Thema FOF's siehe [15] und [16].

2.2.3 Transformation der Formanten

Für die Transformation der Formanten wird zunächst wieder, genau wie beim seriellen Modell, für das unterabgetastete Signal ein AR-Modell bestimmt (über TE-LPC), und danach mit Hilfe des inversen Modellfilters eine geweißte Version des Signals generiert. Jetzt wird die Transferfunktion des Modellfilters jedoch nicht faktorisiert, sondern stattdessen eine Partialbruchzerlegung durchgeführt (mit Hilfe der `residuez`-Funktion) - diese führt zu einer Repräsentation des Modellfilters als parallele Verschaltung von Einpol-Filtern, die im allgemeinen komplexe Koeffizienten haben. Da unser Ausgangspunkt jedoch eine Transferfunktion mit reellen Koeffizienten war, finden wir zu jedem dieser komplexwertigen Einpol-Filter ein konjugiert komplexes Gegenstück. Wenn wir nun die Transferfunktionen der Einpol-Filter mit den jeweils konjugiert komplexen Polen addieren, erhalten wir Zweipol-Filter mit einer Nullstelle und reellen Koeffizienten (b'_0, b'_1, a_1, a_2) . Wenn wir im Nenner b'_0 ausklammern und in g umbenennen, erhalten wir eine Darstellung, bei der b_0 auf 1 normalisiert ist, und in der g direkt als Gain-Faktor interpretiert werden kann. Unsere Freiheitsgrade sind nun also g, b_1, a_1 und a_2 . Die intuitive Interpretation von g wurde gerade erläutert, die Koeffizienten a_1 und a_2 können über (Gl. B.3) und (2.13) in eine Resonanzfrequenz und -bandbreite umgerechnet werden. Der Parameter b_1 bestimmt die Position der reellen Nullstelle und damit einen allgemeinen Tiefpass- oder Hochpasscharakter und gleichzeitig beeinflusst er den Phasenfrequenzgang - so ganz intuitiv verständlich ist er daher noch nicht. Alternativ dazu können wir die Parameter g, b_1, a_1, a_2 der einzelnen Teilfilter auch in die entsprechenden FOF-Parameter A, α, ω, ϕ umrechnen, die Transformationen in dieser Repräsentation durchführen und aus den modifizierten FOF-Parametern dann die neuen Filter-Parameter berechnen. Aufgrund der unmittelbaren Interpretierbarkeit aller 4 FOF-Parameter wurde dieser Weg auch in den nachfolgend dargestellten Experimenten gewählt. Da die erforderlichen Gleichungen recht unübersichtlich verstreut und teilweise nicht nach den gesuchten Variablen umgestellt sind, sind die Umrechnungen noch einmal in algorithmischer Schreibweise in Algorithmus 3 und 4 angegeben.

Algorithmus 3 Umrechnung von Filter-Koeffizienten in FOF-Parameterinput: g, b_1, a_1, a_2 output: A, α, ω, ϕ

$$P \leftarrow \sqrt{a_2}$$

$$\alpha \leftarrow -\ln(P)$$

$$\omega \leftarrow \arccos\left(\frac{-a_1}{2P}\right)$$

$$\Re\{r\} \leftarrow \frac{1}{2}$$

$$\Im\{r\} \leftarrow -\frac{\sin(2\omega) + 2(b_1/P)\sin(\omega)}{2(1 - \cos(2\omega))}$$

$$R \leftarrow \sqrt{\Re^2\{r\} + \Im^2\{r\}} = \sqrt{\frac{1}{4} + \Im^2\{r\}}$$

$$\varphi \leftarrow \arctan\left(\frac{\Im\{r\}}{\Re\{r\}}\right) = \arctan(2 \cdot \Im\{r\})$$

$$\phi \leftarrow \varphi + \frac{\pi}{2}$$

$$A \leftarrow 2gR$$

Algorithmus 4 Umrechnung von FOF-Parametern in Filter-Koeffizienteninput: A, α, ω, ϕ output: g, b_1, a_1, a_2

$$\varphi \leftarrow \phi - \frac{\pi}{2}$$

$$P \leftarrow e^{-\alpha}$$

$$\Re\{r\} \leftarrow \frac{1}{2}$$

$$\Im\{r\} \leftarrow \Re\{r\} \cdot \tan(\varphi) = \frac{1}{2} \cdot \tan(\varphi)$$

$$R \leftarrow \sqrt{\Re^2\{r\} + \Im^2\{r\}} = \sqrt{\frac{1}{4} + \Im^2\{r\}}$$

$$a_2 \leftarrow P^2$$

$$a_1 \leftarrow -2P \cos(\omega)$$

$$g \leftarrow \frac{A}{2R}$$

$$b_1 \leftarrow -\frac{P}{2} \frac{2(1 - \cos(2\omega))\Im\{r\} + \sin(2\omega)}{\sin(\omega)}$$

Die Zuweisung $\Re\{r\} \leftarrow \frac{1}{2}$ kann man sich eigentlich sparen und in den Anweisungen, die diese Größe benötigen, direkt die Konstante $\frac{1}{2}$ benutzen. Sie wurde der Vollständigkeit halber aber dennoch eingebaut. Diese Umrechnungen sind auf diese Weise in den Funktionen `filterCoeffsToFofParams` und `fofParamsToFilterCoeffs` implementiert.

Ein 'partielles Whitening' des Signals, so wie es beim seriellen Allpolmodell beschrieben wurde ist hier übrigens nicht möglich. Dies wird klar, wenn man sich wieder die Transferfunktion zur Transformation $T(z)$ ansieht. Diese nimmt hier folgende Form an:

$$T(z) = \frac{H'(z)}{H(z)} = \frac{\sum_{s=1}^S H'_s(z)}{\sum_{s=1}^S H_s(z)}$$

Selbst wenn viele der H_s mit ihren korrespondierenden H'_s identisch sind, ist hier kein Kürzen möglich. Für Transformationen mit diesem Modell muss daher immer zuerst ein komplettes Whitening mit dem Prediction-Error Filter ausgeführt, und bei der Resynthese die komplette Filterbank implementiert werden.

Kompensation der Bandbreitenänderung im Gain-Faktor

Auch in der Repräsentation über die FOF-Parameter gibt es immer noch eine Kopplung zwischen der Dämpfung α (bzw. der dazu proportionalen Bandbreite bw_{Hz}) und der Intensität des Formanten. Wenn man beispielsweise ein α_s verringert, dann vergrößert sich dadurch auch die relative Intensität des Formanten, d.h. das Peak im Frequenzgang ragt höher heraus. Dies ist intuitiv einleuchtend, wenn man sich klarmacht, dass die Impulsantwort ja langsamer abklingt, wenn α klein ist und somit auch mehr Energie enthält. Eine Möglichkeit zur Kompensation dieses Effekts, die sich als praktisch sinnvoll erwiesen hat, liegt darin, die Fläche unter der Abklingfunktion $e^{-\alpha t}$ (hier als kontinuierliche Funktion betrachtet) konstant zu halten, indem die Abklingfunktion mit einem zusätzlichen Faktor skaliert wird. Dieser Kompensationsfaktor kann dann im Gain-Faktor g bzw. in der FOF-Amplitude A absorbiert werden. Die Fläche E unter der Abklingfunktion ist gegeben mit:

$$E = \int_0^{\infty} e^{-\alpha t} dt = \lim_{c \rightarrow \infty} \int_0^c e^{-\alpha t} dt = \lim_{c \rightarrow \infty} \left. \frac{1}{-\alpha} e^{-\alpha t} \right|_0^c = \lim_{c \rightarrow \infty} \left(\frac{1}{-\alpha} e^{-\alpha c} - \frac{1}{-\alpha} e^{-\alpha \cdot 0} \right)$$

Diese Fläche ist allerdings nicht identisch mit der Energie der Impulsantwort, der Buchstabe E soll hier nur andeuten, dass diese Fläche jedoch als 'ein Maß' für die Energie benutzt wird (aber dies soll hier nicht im mathematisch strengen, maßtheoretischen Sinn verstanden werden). Da wir von stabilen Filtern und damit von $\alpha > 0$ ausgehen können, existiert das Integral, denn der Grenzwert wird in diesem Fall zu Null. Damit:

$$E = 0 - \frac{1}{-\alpha} e^{-\alpha \cdot 0} = \frac{1}{\alpha}$$

Die Fläche unter der Abklingfunktion ist also umgekehrt proportional zur Dämpfung α . Außerdem ist sie natürlich direkt proportional zu einer eventuell vorhandenen multiplikativen Konstante A , d.h.:

$$E = \int_0^{\infty} A e^{-\alpha t} dt = \frac{A}{\alpha}$$

Wenn wir nun bei einer Veränderung von α fordern wollen, dass diese Fläche konstant bleiben soll (d.h. $E_{new} = E_{old}$) dann bedeutet dies:

$$\frac{A_{new}}{\alpha_{new}} = \frac{A_{old}}{\alpha_{old}}$$

und somit:

$$A_{new} = A_{old} \frac{\alpha_{new}}{\alpha_{old}}$$

Wir skalieren also die FOF-Amplitude einfach mit dem Faktor $\alpha_{new}/\alpha_{old}$.

2.2.4 Implementation und Praxistest

Wir wollen in diesem Abschnitt die gleichen Transformationen vornehmen, wie im Praxistest des seriellen Allpolmodells und die Ergebnisse miteinander vergleichen. Wir gehen

hier von der Repräsentation der einzelnen Teilfilter in ihrer äquivalenten FOF-Repräsentation aus, d.h. zur Formant-Transformation verändern wir die FOF-Parameter A , α , ω , ϕ . Die hier dargestellten Ergebnisse können anhand des MatLab-Skripts `trafoParallel.m` nachvollzogen werden. Als erstes soll wieder die Mittenfrequenz des zweiten Formanten (also ω_2) um den Faktor 1.5 erhöht werden. Das Ergebnis ist in Abb. 2.18 dargestellt.

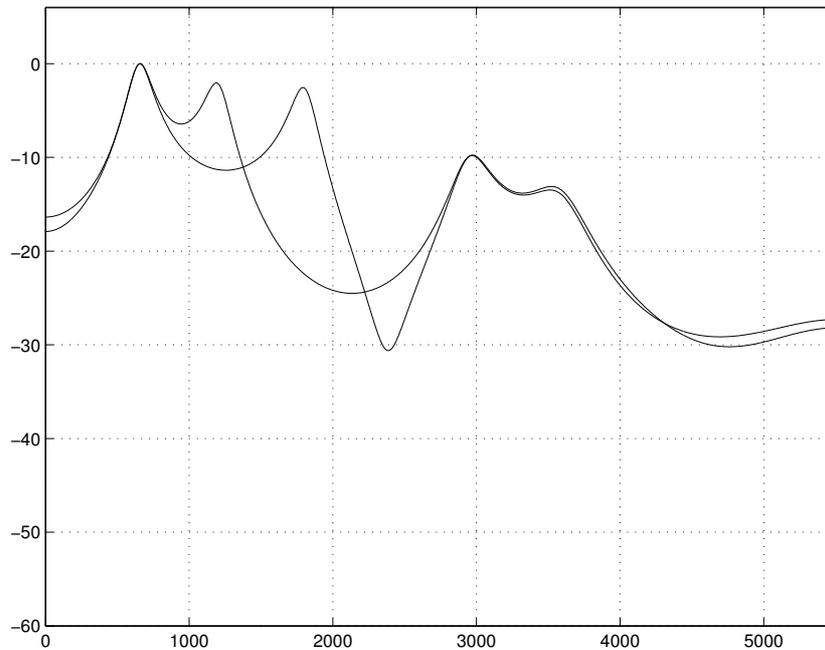


Fig. 2.18: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formantfrequenz um Faktor 1.5 erhöht)

Im Gegensatz zur gleichen Transformation mit dem seriellen Allpolmodell sehen wir, dass die generelle Anhebung der höheren Frequenzen hier ausbleibt. Beim seriellen Allpolmodell war diese ja deshalb zu beobachten, weil die Verschiebung der Mittenfrequenz eines Teilfilters hier immer global wirkte. Weiterhin ist zu beobachten, dass das relative Minimum zwischen 2. und 3. Formanten nicht nur enger, sondern auch tiefer wird. Hier sehen wir den Effekt der Interferenz zwischen den Ausgangssignalen der parallelen Teilfilter. Bei bestimmten Frequenzen überlagern sich die Teilsignale nämlich gegenphasig, so dass bei diesen Frequenzen eine Abschwächung durch destruktive Interferenz auftritt. Wie stark dieser Effekt ausgeprägt ist, hängt natürlich von den jeweiligen Phasen- und Amplitudenverhältnissen der Teilfilter im Übergangsbereich ab. Beim vorliegenden Signal beispielsweise würde eine Anhebung der Frequenz des zweiten Formanten um den Faktor 1.93 zu einer regelrechten Kerbe im Frequenzgang führen. Dies ist in Abb. 2.19 deutlich zu sehen.

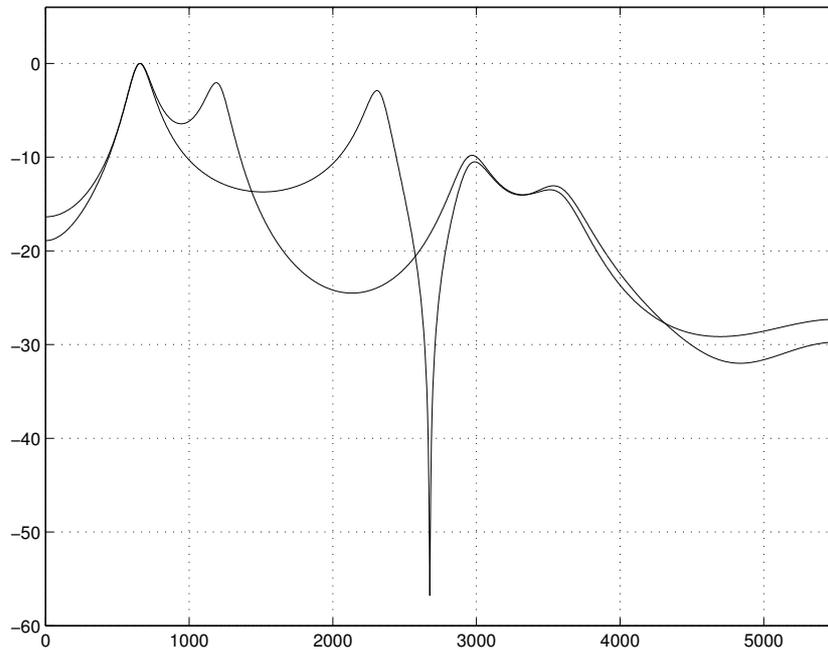


Fig. 2.19: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formantfrequenz um Faktor 1.93 erhöht) - man beachte den Einbruch aufgrund von Interferenz zwischen den Teilsignalen

Es ist natürlich prinzipiell möglich, solche Effekte über die Startphasen ϕ der einzelnen Formant-Filter zu kontrollieren. Dies wird in [16] als 'dephasing' bezeichnet. Um ein solches dephasing allerdings wirklich gezielt durchführen zu können, muss natürlich erst einmal mathematisch klar definiert werden, was Ziel des dephasings sein soll - eine vage Formulierung, wie z.B. 'das relative Minimum soll ungefähr so tief bleiben, wie es ist', reicht hier nicht aus. Dann müssten die Phasen und Amplituden aller Teilfilter in Betracht gezogen und aufeinander abgestimmt werden. Insgesamt erscheint also eine gezielte Einstellung der Phasen zur Vermeidung des Einbruchs als nicht praktikabel.

Als nächstes soll die Bandbreite (bzw. die dazu proportionale Dämpfung α) des zweiten Formanten wieder um den Faktor 0.25 gesenkt werden.

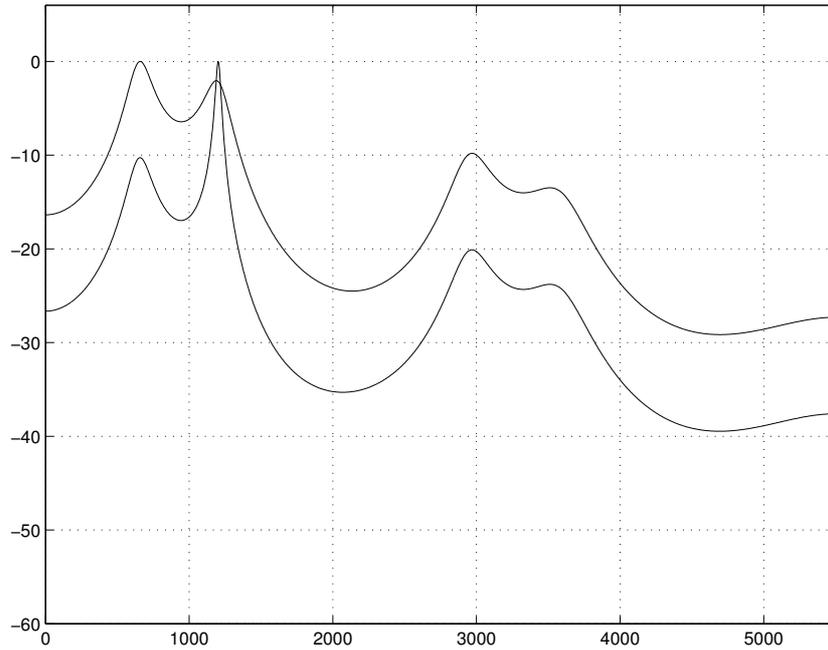


Fig. 2.20: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formant-Bandbreite um Faktor 0.25 gesenkt)

Wie in Abb. 2.20 zu sehen, ähnelt das Ergebnis sehr stark dem aus dem seriellen Allpolmodell. Hier wurde allerdings tatsächlich nur α_2 skaliert, ohne dabei die höhere Energie des Formanten aufgrund des geringeren Polabstands zum Einheitskreis über den Gain-Faktor g_2 zu kompensieren. Dies wurde in Abb. 2.21 getan - der Gain-Faktor wurde hier mit $\alpha_{new}/\alpha_{old}$ skaliert. Die Höhe des Peaks bleibt hier ungefähr erhalten, jedoch sind wieder Seiteneffekte auf die umliegenden Frequenzbereiche aufgrund der neuen Amplitudenverhältnisse der interferierenden Signale zu beobachten.

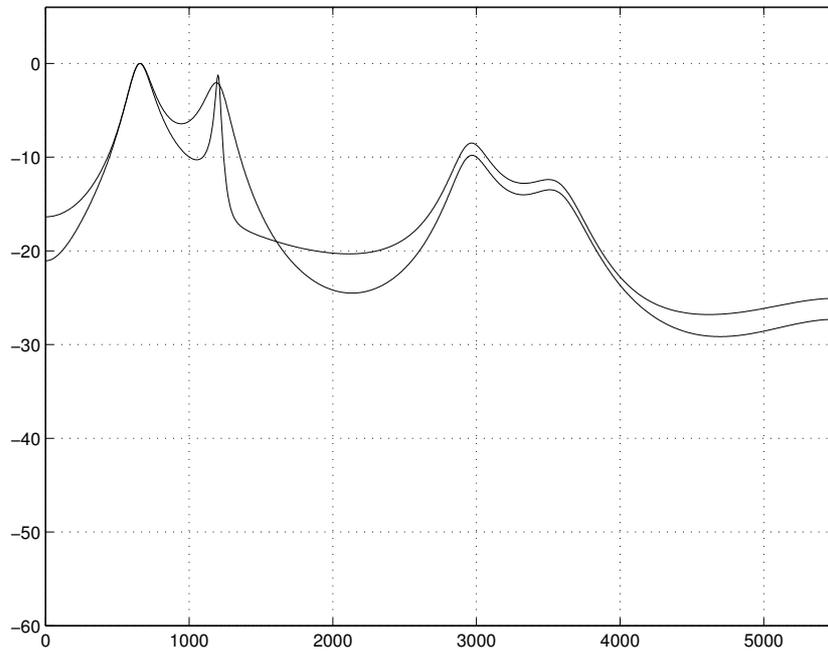


Fig. 2.21: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formant-Bandbreite um Faktor 0.25 gesenkt mit Kompensation im Gain-Faktor)

Schließlich ist es mit diesem Modell natürlich auch zusätzlich möglich, die Formant-Intensitäten direkt über die Gain-Faktoren zu verändern. In Abb. 2.22 ist die Wirkung der Reduzierung des Gain-Faktors für Formant 2 um den Faktor 0.25 zu sehen.

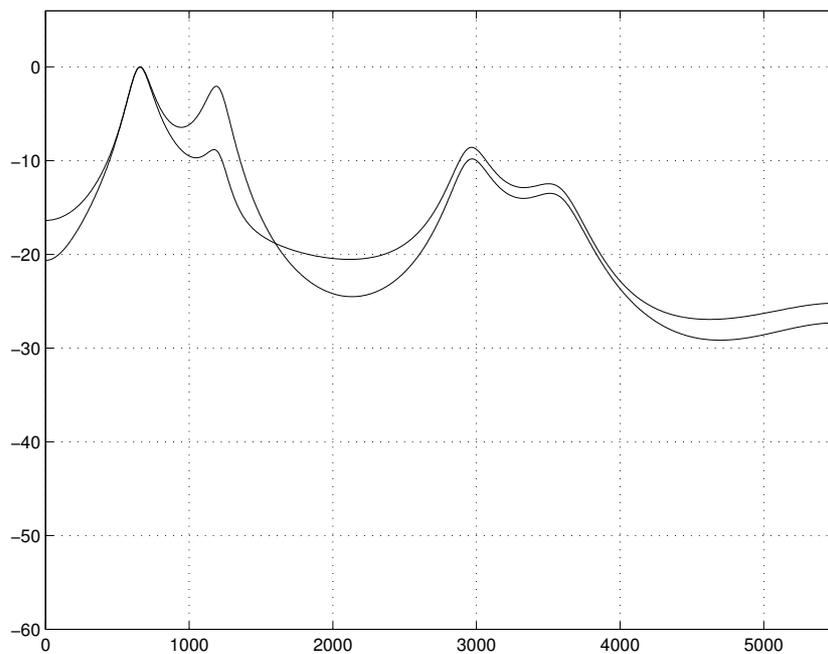


Fig. 2.22: Frequenzgänge des originalen und des modifizierten 10-Pol-Modellfilters (2. Formant-Intensität um Faktor 0.25 gesenkt)

Auch das Verändern der Formant-Bandbreiten und -Intensitäten kann zu unerwünschten Nebeneffekten, wie zum Beispiel Einbrüchen im Frequenzgang, aufgrund der neuen Phasen- und Amplitudenverhältnisse zwischen den Teilsignalen führen.

2.2.5 Vor- und Nachteile des Modells

Vorteile

- Die Modellparameter sind auch hier noch relativ leicht zu bestimmen (über AR-Modellierung und Partialbruchzerlegung des AR-Modells)
- Formant-Frequenzen, -Bandbreiten und -Intensitäten sind unabhängig voneinander regelbar.

Nachteile

- Bei Transformationen treten schwer planbare Seiteneffekte auf, aufgrund der Interferenz zwischen den Ausgangssignalen der parallelen Filter.
- Da auch dieses Modell auf der Zerlegung einer Allpol-Transferfunktion beruht, gelten bezüglich der Sample-Rate dieselben Einschränkungen wie beim seriellen Allpol-Modell

2.3 Serielle Parametrische Equalizer

2.3.1 Das Modell

Für die Modellierung von Formantspektren wird in diesem Abschnitt eine Reihenschaltung von parametrischen Equalizern untersucht. Diese einzelnen parametrischen Equalizerstufen werden auch hier durch Filter zweiter Ordnung realisiert, allerdings enthalten diese Filter zweiter Ordnung jetzt jeweils ein Polstellenpaar und ein Nullstellenpaar (sind also 'echte' Biquads). Als Blockdiagramm sieht die Struktur unseres Modellfilters folgendermaßen aus:

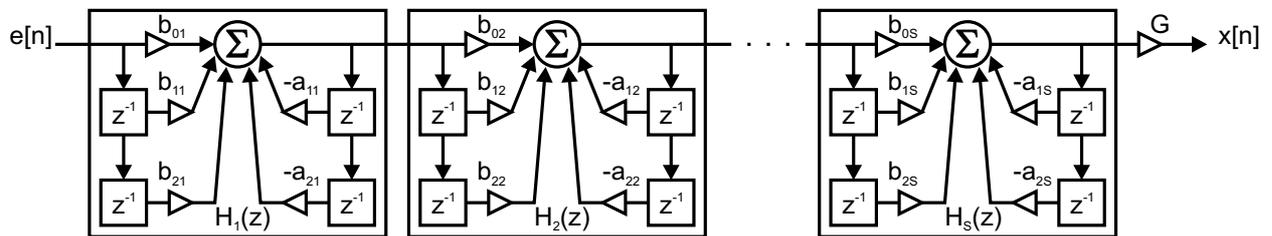


Fig. 2.23: Serielle Verschaltung von S Filtern 2.Ordnung (Biquads)

wobei die Transferfunktion einer einzelnen Biquad-Stufe mit dem Index s gegeben ist durch:

$$H_s(z) = \frac{b_{0s} + b_{1s}z^{-1} + b_{2s}z^{-2}}{a_{0s} + a_{1s}z^{-1} + a_{2s}z^{-2}} \quad (2.17)$$

mit $a_{0s} = 1 \forall s$. Die Transferfunktion einer Reihenschaltung S solcher Equalizer ergibt sich als Multiplikation der Transferfunktionen der einzelnen Stufen, außerdem wird ein globaler Gain-Faktor G eingeführt:

$$\begin{aligned} H(z) &= G \prod_{s=1}^S H_s(z) \\ &= G \prod_{s=1}^S \frac{b_{0s} + b_{1s}z^{-1} + b_{2s}z^{-2}}{a_{0s} + a_{1s}z^{-1} + a_{2s}z^{-2}} \end{aligned} \quad (2.18)$$

wobei der Index s über die einzelnen Equalizerstufen läuft.

Als Grundlage für das Design der einzelnen Biquad-Stufen werden die Gleichungen für die Berechnung der Filterkoeffizienten von Robert Bristow Johnson [3] benutzt. Nach Robert

Bristow Johnson ergeben sich die Koeffizienten für einen parametrischen Equalizer zu:

$$\begin{array}{l}
 b_0 = \frac{1 + \gamma\sqrt{g}}{1 + \gamma/\sqrt{g}} \\
 b_1 = \frac{-2 \cos(\Omega_c)}{1 + \gamma/\sqrt{g}} \\
 b_2 = \frac{1 - \gamma\sqrt{g}}{1 + \gamma/\sqrt{g}} \\
 a_0 = 1 \\
 a_1 = \frac{-2 \cos(\Omega_c)}{1 + \gamma/\sqrt{g}} \\
 a_2 = \frac{1 - \gamma/\sqrt{g}}{1 + \gamma/\sqrt{g}}
 \end{array} \quad (2.19)$$

wobei, wie man sieht, $b_1 = a_1$ gilt. Dabei ist Ω_c die normierte Mittenfrequenz des Equalizers. Diese ergibt sich aus der Mittenfrequenz f_c (in Hz) und der Sample-Rate f_s (in Hz) wie folgt:

$$\Omega_c = 2\pi \frac{f_c}{f_s} \quad (2.20)$$

Der Parameter g ist der Verstärkungsfaktor - und zwar nicht in dB, sondern als Faktor mit der die Amplitude eines Sinus mit einer Frequenz von Ω_c verstärkt wird. Oft ist es jedoch intuitiver, den Verstärkungsfaktor in dB anzugeben - hierfür werden wir die Bezeichnung g_{dB} benutzen. Für die Umrechnung gilt:

$$g = 10^{g_{dB}/20} \quad (2.21)$$

Der Parameter γ ist der Bandbreitenparameter, der sich wie folgt aus der (in Oktaven gegebenen) Bandbreite bw_{oct} berechnet:

$$\gamma = \sinh \left(\frac{\ln(2)}{2} bw_{oct} \frac{\Omega_c}{\sin(\Omega_c)} \right) \sin(\Omega_c) \quad (2.22)$$

Aus der Transferfunktion $H(z)$ (Gl. 2.18) ergibt sich der Frequenzgang des Equalizers durch Auswertung der Transferfunktion entlang des Einheitskreises. Dies geschieht durch die Substitution $z \leftarrow e^{j\Omega}$. Also:

$$H(e^{j\Omega}) = G \prod_{s=1}^S \frac{b_{0s} + b_{1s}e^{-j\Omega} + b_{2s}e^{-2j\Omega}}{a_{0s} + a_{1s}e^{-j\Omega} + a_{2s}e^{-2j\Omega}} \quad (2.23)$$

In den Abbildungen 2.24, 2.25 und 2.26 sind beispielhaft einige Frequenzgänge solcher parametrischer Equalizer abgebildet.

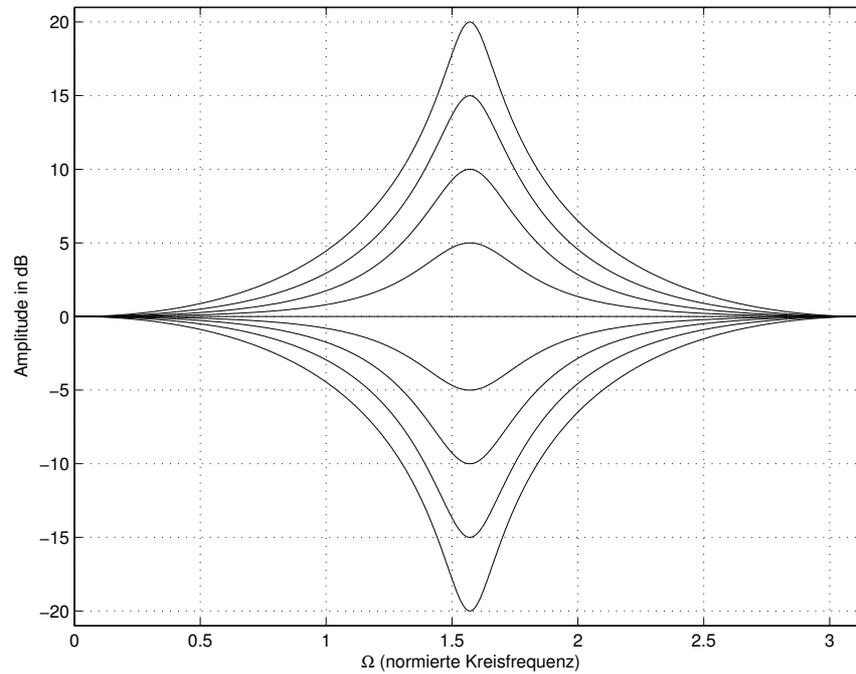


Fig. 2.24: Familie von Equalizer-Übertragungsfunktionen mit $\Omega_c = \pi/2$, $bw_{oct} = 0.5$ und verschiedenen Gain-Werten $g_{dB} = -20, -15, -10, \dots, +20$

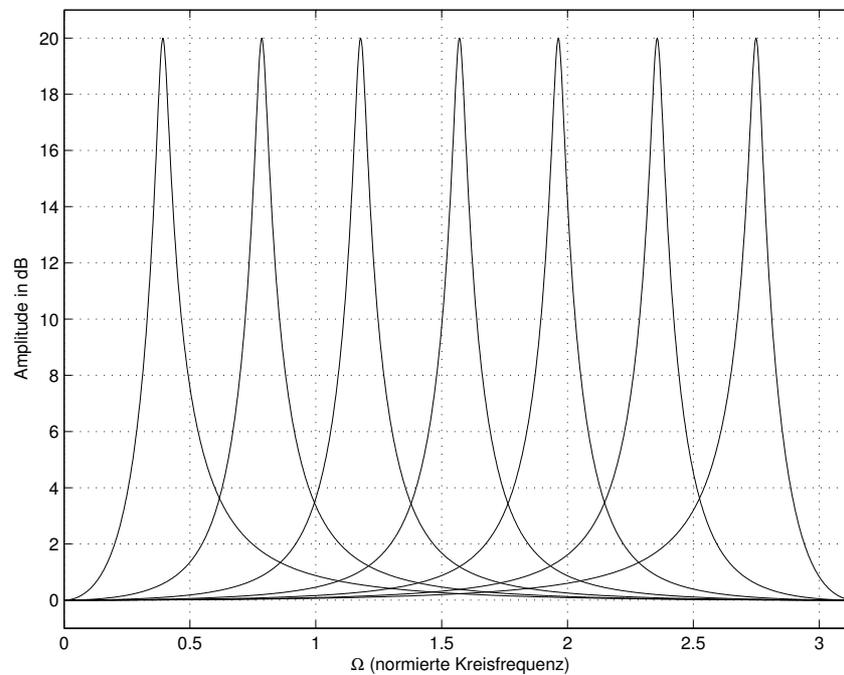


Fig. 2.25: Familie von Equalizer-Übertragungsfunktionen mit $g_{dB} = 20$, $\gamma \approx 0.684$ (das entspricht $bw_{oct} = 0.5$ bei $\Omega_c = \pi/2$) und verschiedenen Mittenfrequenzen $\Omega_c = \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \dots, \pi - \frac{\pi}{8}$

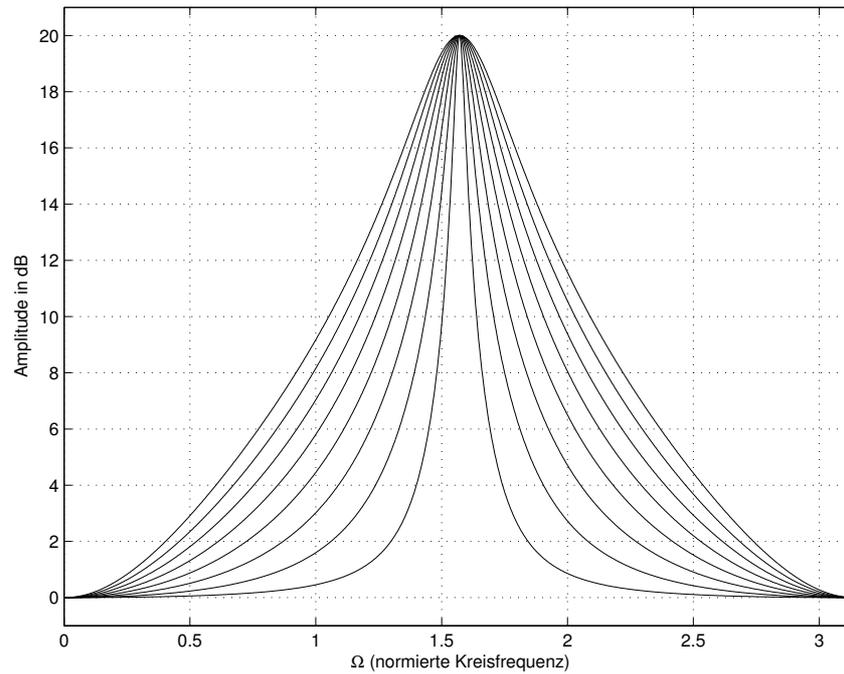


Fig. 2.26: Familie von Equalizer-Übertragungsfunktionen mit $g_{dB} = 20$, $\Omega_c = \pi/2$ und verschiedenen Bandbreiten $bw_{oct} = \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \dots, 1$

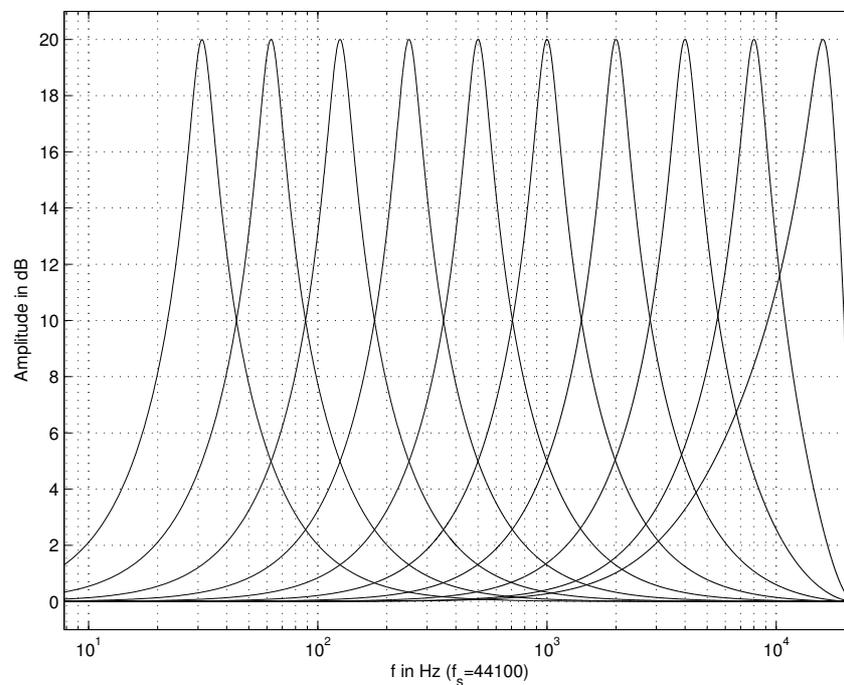


Fig. 2.27: Familie von Equalizer-Übertragungsfunktionen mit $g_{dB} = 20$, $bw_{oct} = 1.0$ und verschiedenen Mittenfrequenzen $f_c = 31.25Hz, 62.5Hz, 125Hz, \dots, 16kHz$ bei einer Sample-Rate von $f_s = 44.1kHz$, logarithmisch skalierte Frequenzachse

Die Übertragungsfunktionen der einzelnen Equalizerstufen sind also Funktionen, die eine Glockenform haben, solche Filter bezeichnet man daher auch als Glocken- bzw. im englischen als Bell- oder Peak-Filter. Die Asymmetrie, die in Abb. 2.25 bei den Frequenzgängen mit besonders kleiner und besonders großer Mittenfrequenz zu sehen ist, ist durch die Art der Achsenskalierung begründet. Wenn man die Frequenzachse logarithmisch skaliert, die Mittenfrequenzen exponentiell variiert (z.B. jeweils eine Oktave Abstand zwischen zwei Bändern läßt) und die Bandbreite bw_{oct} anstatt γ konstant hält (γ ist eine nichtlineare Funktion von bw_{oct} und Ω_c), dann verschwindet diese Asymmetrie mehr oder weniger. Dies ist in Abbildung 2.27 zu sehen. Eine gewisse Rest-Asymmetrie in der Übertragungsfunktion bleibt bei sehr hohen Mittenfrequenzen dennoch erhalten. Dies liegt an den Frequenzverzerrungen bei der bilinearen Transformation der analogen Prototyp-Filter zu digitalen Filtern (siehe [3]).

Bemerkung: Die Multiplikation der einzelnen Frequenzgänge aufgrund der Reihenschaltung ist äquivalent zu einer Addition der einzelnen log-Frequenzgänge. Das bedeutet, dass wir in diesem Modell unsere logarithmierte spektrale Hüllkurve durch eine Superposition von Glockenkurven approximieren. Hier besteht eine gewisse Analogie zu Gaussian Mixture Models, bei denen eine Funktion $p(x)$ (meist ist dies eine Wahrscheinlichkeitsdichte, daher das p) als Superposition von Gauss-Glocken modelliert wird:

$$\begin{aligned}\hat{p}(x) &= \sum_{i=1}^N w_i \mathcal{N}(\mu_i, \sigma_i) \\ &= \sum_{i=1}^N w_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right)\end{aligned}$$

wobei $\hat{p}(x)$ das Modell für $p(x)$ sein soll. Die Gewichtungsfaktoren w_i für die einzelnen Gauss-Glocken entsprechen unseren g -Parametern, die Varianzen σ_i entsprechen den γ -Parametern, und die Erwartungswerte μ_i entsprechen unseren Mittenfrequenzen Ω_c . Dies nur so als kleine Randbemerkung, für mehr Details zu dem Thema siehe [1].

Antiformanten

Eine Besonderheit dieses speziellen Formantmodells ist die, dass hier ohne großen zusätzlichen Aufwand auch Antiformanten dargestellt werden können. Antiformanten sind Frequenzbereiche in denen weniger Energie vorhanden ist als in umliegenden Frequenzbereichen - also die Täler der spektralen Hüllkurve. Bei der Stimme beispielsweise entstehen solche Antiformanten bei Nasalen und Nasalvokalen durch Zusammenwirken von Mund- und Nasenhöhle. Das einzige, was wir tun müssen, um in diesem Modell die Antiformanten mit zu modellieren ist, Werte kleiner als eins für die gain-Faktoren g_s zuzulassen (bzw. Werte kleiner Null für die entsprechenden dB -Werte).

2.3.2 Die Kostenfunktion und ihr Gradient

Der Parametervektor

Wir haben also S Filterstufen, von denen jede einzelne über einen Verstärkungsparameter (g bzw. g_{dB}), einen Mittenfrequenzparameter (Ω_c bzw. f_c) und einen Bandbreitenparameter (γ bzw. bw_{oct}) verfügt, wobei sich die verschiedenen Darstellungsformen der Parameter ineinander umrechnen lassen. Zusätzlich haben wir noch den globalen Verstärkungsfaktor G (bzw. G_{dB} , wenn man diesen in dB ausdrücken will). Insgesamt haben wir also $3S + 1$ einzustellende Parameter. Oft ist es übersichtlicher, diese Parameter in einem Vektor zusammenzufassen. Wir definieren also den Parametervektor unserer Filterstruktur als:

$$\boldsymbol{\theta} := (G \ g_1 \dots g_S \ \Omega_{c1} \dots \Omega_{cS} \ \gamma_1 \dots \gamma_S)^T \quad (2.24)$$

Die Kostenfunktion

Die spektrale Hüllkurve des Eingangssignals, dessen Formanten und/oder Antiformanten mit Hilfe einer solchen Equalizer-Kette nachgebildet werden soll, sei mit $X(e^{j\Omega})$ bezeichnet. Als Kostenfunktion bietet sich eine gewichtete mittlere quadratische Abweichung zwischen der logarithmierten quadrierten spektralen Hüllkurve des Eingangssignals und dem logarithmierten Leistungsdichtespektrum des Equalizers an:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} w(\Omega) (\ln |X(e^{j\Omega})|^2 - \ln |H(e^{j\Omega})|^2)^2 d\Omega$$

Hierbei ist $w(\Omega)$ eine Gewichtungsfunktion die es ermöglicht, in bestimmten Frequenzbereichen eine bessere Anpassung des Equalizers an das Spektrum zu bewirken, wobei in anderen Frequenzbereichen unter Umständen eine Verschlechterung der Anpassung in Kauf genommen wird. Die oben genannte Kostenfunktion ist jedoch in erster Linie von theoretischem Wert, da hier angenommen wird, dass unsere zu approximierende Funktion $\ln |X(e^{j\Omega})|$ als kontinuierliche Funktion in Form eines mathematischen Ausdrucks vorliegt (und dass das Integral gelöst werden kann). In der Praxis haben wir es jedoch mit diskreten Spektren zu tun - d.h. unsere spektrale Hüllkurve ist nur an bestimmten Stützstellen Ω_k mit $k = 0, \dots, K - 1$ definiert, wobei K die Anzahl dieser Stützstellen (Bins) ist. Daher geht das Integral über in eine Summe:

$$E = \frac{1}{K} \sum_{k=0}^{K-1} w(\Omega_k) (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2$$

Die Kostenfunktion so zu definieren macht mathematisch Sinn, ist aber aus programmier-technischer Sicht Verschwendung, da aufgrund der DFT-Symmetrie die Werte oberhalb von $K/2$ redundant sind, hier aber zweimal aufsummiert werden. Es reicht völlig aus, die Summe nur bis $(K/2)$ laufen zu lassen, und das Ergebnis mit 2 zu multiplizieren. Das Ergebnis ist nicht exakt identisch aufgrund der Sonderrollen der Bins 0 (DC) und $K/2$ (Nyquist-Frequenz) - bei denen würde man den Fehler jetzt doppelt gewichten. Aber das sind wirklich Feinheiten, die in der Praxis kaum relevant sein dürften, nötigenfalls aber

in der Gewichtungsfunktion kompensiert werden können. Wir definieren daher unsere Kostenfunktion als:

$$E = \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2 \quad (2.25)$$

Diese Kostenfunktion sollte bezüglich der Parameter $\Omega_{cs}, g_s, \gamma_s$ mit $s = 1, \dots, S$ und G minimiert werden. Da man die Reihenfolge der S Equalizerstufen beliebig vertauschen (permutieren) kann, ohne die Gesamt-Transferfunktion zu ändern, ergeben sich schon mal $S!$ äquivalente Lösungen. Wir lösen dies auf, indem wir festlegen, dass die erste Stufe für den untersten Formanten, die zweite für den nächsten usw. zuständig ist. Die Equalizerstufen sind also aufsteigend nach ihrer Mittenfrequenz sortiert. Falls Antiformanten explizit modelliert werden sollen, folgen danach die entsprechenden Equalizerstufen (ebenfalls aufsteigend nach der Mittenfrequenz sortiert).

Der Gradient der Kostenfunktion

Für die Optimierung einer Funktion ist es im allgemeinen nötig, deren partielle Ableitungen bezüglich der zu optimierenden Parameter zu kennen. Da wir die Kostenfunktion 2.25 bezüglich des Gain-Faktors G , der Mittenfrequenzen Ω_{cs} , Bandbreiten γ_s und Verstärkungsfaktoren g_s der einzelnen Equalizerstufen (die alle in $|H(e^{j\Omega_k})|$ stecken) optimieren wollen, benötigen wir also Ausdrücke für die Ableitungen der Kostenfunktion nach eben diesen Parametern. Wenn man die einzelnen partiellen Ableitungen zu einem Vektor zusammenfasst, so wird dieser Vektor als Gradient der Funktion E bezeichnet. Unser Gradient hat demzufolge die Form:

$$\nabla E = \left(\frac{\partial E}{\partial G} \quad \frac{\partial E}{\partial g_1} \quad \dots \quad \frac{\partial E}{\partial g_S} \quad \frac{\partial E}{\partial \Omega_{c1}} \quad \dots \quad \frac{\partial E}{\partial \Omega_{cS}} \quad \frac{\partial E}{\partial \gamma_1} \quad \dots \quad \frac{\partial E}{\partial \gamma_S} \right)^T \quad (2.26)$$

Die Herleitung entsprechender Ausdrücke für die benötigten partiellen Ableitungen ist in Anhang B.5 ausgeführt. In Anhang A werden einige Verfahren zur gradientenbasierten Minimierung der Kostenfunktion vorgestellt.

2.3.3 Die frequenzabhängige Gewichtung des Fehlers

In der Definition der Kostenfunktion wurde eine frequenzabhängige Gewichtung des Fehlers $w(\Omega_k)$ eingeführt. Dies geschah mit dem Hintergedanken, dass es sinnvoll sein könnte, in bestimmten Frequenzbereichen eine bessere Anpassung des Modellspektrums an die spektrale Hüllkurve zu verlangen als in anderen. In diesem Abschnitt sollen nun verschiedene solcher Funktionen betrachtet werden, und untersucht werden, welche Funktionen als besonders geeignet erscheinen. Die einfachste Variante wäre eine einheitliche Gewichtung des Fehlers mit dem Faktor 1:

$$w(\Omega_k) = 1 \quad \forall \Omega_k$$

dies entspräche dem Fall, in dem wir diese Gewichtung gar nicht erst eingeführt hätten. Nun erscheint es wenig sinnvoll, von unserer Equalizer-Kette zu erwarten, im gesamten

Spektrum eine gute Anpassung hinzubekommen. Da wir ja die Formanten mit den einzelnen Bändern modellieren wollen, erscheint es eher sinnvoll, eine gute Anpassung nur in denjenigen Frequenzbereichen zu verlangen, in denen auch tatsächlich Formanten vorhanden sind. Bei der Synthese werden sich die Equalizer in allen anderen Frequenzbereichen neutral verhalten (der Frequenzgang eines Equalizers ist ja ungefähr eins außerhalb des Wirkungsbereiches). Daher wählen wir eine bandpass-artige Gewichtung:

$$w(\Omega_k) = \begin{cases} 1 & \text{für } \Omega_u < \Omega_k < \Omega_o \\ 0 & \text{sonst} \end{cases}$$

wobei Ω_u und Ω_o die untere und obere Grenzfrequenz sind, bei der uns die Abweichungen zwischen spektraler Hüllkurve des Signals und Frequenzgang der Equalizer-Kette interessieren. Unterhalb der unteren Grenzfrequenz und oberhalb der oberen Grenzfrequenz werden sich die Equalizer, wie gesagt, dann bei der Synthese ungefähr neutral verhalten - ganz im Gegensatz zu dem Zweipol-Resonator, der ja ober- oder unterhalb seiner Resonanzfrequenz eine abfallende Flanke aufweist. Mit einer solchen Gewichtung werden innerhalb eines bestimmten Frequenzfensters alle Abweichungen gleichmäßig gewichtet. Wir wollen jedoch in erster Linie die Bereiche in der Nähe von Formanten gut approximieren - wenn die Equalizer-Kette die Täler zwischen den Formanten nicht ganz so gut nachbildet, ist dies meist weniger störend (wie störend dies tatsächlich ist, hängt davon ab, wo man neue Formanten setzen will und welches Whitening-Filter man benutzt, siehe 2.3.5). Daher bietet es sich an, Frequenzbereiche um Formanten herum stärker zu gewichten als andere. Eine einfache Möglichkeit, dies zu realisieren besteht darin, den Fehler einfach mit der spektralen Hüllkurve des Signals zu gewichten:

$$w(\Omega_k) = \begin{cases} |X(\Omega_k)| & \text{für } \Omega_u < \Omega_k < \Omega_o \\ 0 & \text{sonst} \end{cases}$$

denn so wird die Gewichtungsfunktion genau dort groß, wo auch besonders viel Energie im Eingangssignal vorhanden ist. Man könnte jetzt auch noch einen Exponenten für die spektrale Hüllkurve einführen, so dass:

$$\boxed{w(\Omega_k) = \begin{cases} |X(\Omega_k)|^c & \text{für } \Omega_u < \Omega_k < \Omega_o \\ 0 & \text{sonst} \end{cases}} \quad (2.27)$$

dies führt bei $0 < c < 1$ dazu, dass die spektrale Hüllkurve weniger Einfluss auf die Gewichtung hat (für $c \rightarrow 0$ geht ja auch $w(\Omega_k) \rightarrow 1$). Für $c > 1$ würde sich die spektrale Hüllkurve dagegen noch stärker auf die Gewichtung auswirken. In der Praxis hat sich jedoch eine Gewichtung mit $c = 1$ als recht erfolgreich herausgestellt, daher wurde dies auch so implementiert. Übrigens ermöglicht die bandpass-artige Gewichtung des Fehlers in der Praxis auch eine effizientere Berechnung des Gradienten der Kostenfunktion, da die Summe jetzt nur noch über denjenigen Bereich auszuführen ist, in dem $w(\Omega_k) \neq 0$ ist. Das bedeutet, wir berechnen jetzt nur noch:

$$\frac{\partial E}{\partial \theta} = \frac{\partial}{\partial \theta} \frac{2}{K} \sum_{k=k_u}^{k_o} w(\Omega_k) (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2$$

für die partielle Ableitung der Kostenfunktion nach einem beliebigen Parameter θ . Dabei entsprechen die Bin-Indizes k_u und k_o denjenigen Bins, die zu Ω_u und Ω_o gehören. Für die Ω_u und Ω_o können wir beliebige Werte festlegen, die Bin-Frequenzen Ω_k können aber natürlich nur diskrete Werte annehmen. Daher legen wir fest, dass für k_u dasjenige Bin gewählt wird, für das Ω_k gerade kleiner als Ω_u wird, und für k_o dasjenige Bin gewählt wird, für das Ω_k gerade größer als Ω_o wird. Wenn wir z.B. Formanten nur zwischen 200 Hz und 5 kHz suchen und die Nyquist-Frequenz bei 22050 Hz liegt, dann können wir die Summe auf ungefähr ein Viertel ihrer ursprünglichen Indexmenge reduzieren.

2.3.4 Initialisierung der Equalizer-Parameter

Die Minimierung der Kostenfunktion (2.25) stellt ein nichtlineares Optimierungsproblem dar, welches hier durch einen iterativen Algorithmus (den SCG-Algorithmus) gelöst wird. Da bei nichtlinearen Optimierungsproblemen die Kostenfunktion im allgemeinen nicht nur ein (globales) Minimum, sondern zahlreiche lokale Minima besitzen kann, ist es entscheidend, an welchem Punkt im Parameterraum wir mit dem Abstiegsverfahren beginnen. Der Startpunkt entscheidet nämlich darüber, in welches lokale Minimum der Parametervektor hineinlaufen wird. Bei unserem Problem bedeutet dies konkret, dass wir Startwerte für den globalen Gain-Faktor G , die Gain-Faktoren der einzelnen Equalizer-Stufen g_s , $s = 1, \dots, S$, die (normierten) Mittenfrequenzen Ω_{cs} , $s = 1, \dots, S$ und die Bandbreiten-Parameter γ_s , $s = 1, \dots, S$ festlegen müssen.

Für den globalen Gain-Faktor G bietet es sich an, eine gemittelte Amplitude des interessierenden Frequenzbereichs $\Omega_u < \Omega < \Omega_o$ zu wählen, so dass die Übertragungsfunktion im Falle von $g_s = 1$, $s = 1, \dots, S$ einfach eine horizontale Linie bei eben diesem Mittelwert ergibt. Es wurde vorsichtigerweise von 'einer' statt von 'der' gemittelten Amplitude gesprochen, da uns verschiedene Möglichkeiten der Mittelung zur Verfügung stehen - nämlich das arithmetische und das geometrische Mittel. Das geometrische Mittel kann als arithmetische Mittelung der dB-Amplituden angesehen werden und erscheint hier recht sinnvoll und hat sich auch in der Praxis als vorteilhaft erwiesen. Wir initialisieren also G folgendermaßen:

$$\begin{aligned} G &= \left(\prod_{k=k_u}^{k_o} |X(\Omega_k)| \right)^{\frac{1}{k_o - k_u + 1}} \\ &= \exp \left(\frac{1}{k_o - k_u + 1} \sum_{k=k_u}^{k_o} \ln |X(\Omega_k)| \right) \end{aligned} \quad (2.28)$$

wobei die zweite Form für die Implementierung gewählt wurde, da es bei der ersten Form numerische Probleme gab.

Für die Initialisierung der S Equalizer-Mittenfrequenzen Ω_{cs} wählen wir die S höchsten Peaks der spektralen Hüllkurve innerhalb des Frequenzbereichs $\Omega_u < \Omega < \Omega_o$. Dazu werden zunächst diejenigen Frequenzen gesucht, bei denen überhaupt ein (lokales) Maximum vorliegt. Ein Maximum bedeutet, dass der Amplitudenwert bei diesem Bin größer als der des linken und des rechten Nachbar-Bins sein muss. Aus diesen Maxima werden nun mit Hilfe der MatLab-Funktion `max` die S höchsten herausgesucht. Da diese Funktion immer

nur ein Maximum pro Aufruf liefert, wird der entsprechende Amplitudenwert des Maximums auf Null gesetzt, nachdem es gefunden wurde (damit beim nächsten Aufruf nicht wieder das gleiche Maximum gefunden wird).

Nachdem wir die Mittenfrequenzen Ω_{cs} auf diese Weise gefunden haben, erscheint es sinnvoll, als Gain-Wert g_s der Equalizer-Stufe s jeweils denjenigen Wert zu nehmen, den die spektrale Hüllkurve dort (bei Ω_{cs}) hat. Allerdings interessiert uns hier ja nicht der absolute Wert, sondern vielmehr interessiert uns, wie weit dieser Wert oberhalb des Mittelwertes liegt. Daher teilen wir den Absolutwert der spektralen Hüllkurve noch durch den globalen Gain-Faktor G :

$$g_s = \frac{1}{G} |X(\Omega_{cs})|$$

Die Bandbreitenparameter γ wurden einfach mit einer kleinen Konstante $\gamma = 0.005$ initialisiert. Dies führt zu einer recht schmalbandigen Initialisierung, die schon mal an ziemlich genau den richtigen Frequenzen mit sinnvollen Gain-Werten beginnt. Im Laufe der Optimierung breiten sich die Glockenkurven der Equalizer dann aus, bis sie den Raum unter der spektralen Hüllkurve gut ausfüllen. Dabei werden aber natürlich die anderen Parameter auch noch nachjustiert, so dass am Ende der Optimierung insgesamt eine gute Übereinstimmung zwischen spektraler Hüllkurve und Equalizer-Übertragungsfunktion erreicht wird.

Falls Antiformanten explizit modelliert werden sollen, werden die entsprechenden Start-Parameter für die Antiformant-Stufen in ganz analoger Weise bestimmt.

2.3.5 Transformation der Formanten

Nachdem wir den Frequenzgang unserer Equalizer-Kette nun also an die spektrale Hüllkurve unseres Signals angepasst haben, ist es relativ leicht, einzelne spektrale Peaks zu modifizieren. Zunächst benötigen wir dazu wieder ein geweißtes Signal - also eine Version des Eingangssignals in dem die vorhandenen spektralen Peaks geglättet wurden. Dieses geweißte Signal können wir dann mit einer Equalizer-Kette filtern, deren Parameter wir durch unsere Optimierung gewonnen haben. Damit prägen wir dem geweißten Signal die Formanten auf. Vorher steht es uns natürlich frei, die einzelnen Equalizer-Parameter beliebig zu verändern. Normalerweise ist es wünschenswert, bei unveränderter Einstellung der Equalizer-Parameter, eine möglichst exakte Resynthese des Eingangssignals zu erhalten. Exakte Resynthese ist immer dann möglich, wenn als Whitening-Filter das inverse Modellfilter benutzt wird. Man kann zeigen, dass eine Equalizerstufe mit Gain g und eine Stufe mit dem dazu reziproken Gain-Wert $1/g$ invers zueinander sind. Dies ist keine Selbstverständlichkeit, wird aber durch die Form der hier gewählten Filterdesign-Gleichungen gewährleistet. Davon kann man sich anhand des MatLab-Skripts `proofEqualizerInversion.m` überzeugen, welches mit Hilfe der Symbolic-Math Toolbox zeigt, dass das Produkt der Transferfunktionen zweier Equalizerstufen mit reziproken Gain-Werten zu Eins wird.

Wenn auf eine exakte Resynthese bei unveränderter Parametereinstellung verzichtet werden kann, dann können als Whitening-Filter auch andere Filter verwendet werden. Eine nahe liegende Wahl könnte beispielsweise das Prediction-Error-Filter sein, das bei der

Linearen Prädiktion auftritt (bzw. TE-LPC). Der Vorteil dieser Vorgehensweise ist der, dass man als Ausgangssignal für die Resynthese in den meisten Fällen ein 'weißeres' Signal erhält, als es die inverse Equalizer-Kette generieren könnte. Dies ist besonders dann interessant, wenn man die neuen Formanten in Bereiche verschieben möchte, in denen vorher ein tiefes Tal in der spektralen Hüllkurve vorhanden war, welches die inverse Equalizer-Kette nicht ausreichend glätten konnte. Dieser Fall tritt z.B. dann auf, wenn im Signal Antiformanten vorhanden waren, diese aber von der Equalizer-Kette nicht erfasst wurden.

Da sich auch dieses Modell, genau wie Zweipol-Kette, aus einer Reihenschaltung von Teilfiltern ergibt, ist es bei Transformationen einzelner Teilfilter nicht nötig, ein echtes Whitening durchzuführen. Es genügt, das zu transformierende Teilfilter zu invertieren.

2.3.6 Implementation und Praxistest

In diesem Abschnitt soll die Anpassung von Equalizer-Transferfunktionen an spektrale Hüllkurven von realen Signalen in der Praxis getestet werden. Dazu wurde das MatLab-Skript `trafoEqualizers.m` implementiert, anhand dessen die hier dargestellten Versuche nachvollzogen werden können. Wie in den vorherigen Abschnitten auch, soll uns als Beispielsignal das 'Ahh' dienen. Die spektrale Hüllkurve selbst wird dabei mit dem True-Envelope-Algorithmus (mit Hamming-Fenster) bestimmt, während zur Optimierung der Equalizer-Parameter der Scaled Conjugate Gradient Algorithmus zum Einsatz kommt - dies sind momentan für ihren jeweiligen Zweck die leistungsfähigsten Algorithmen, die uns (zumindest innerhalb dieser Arbeit) zur Verfügung stehen. In Abb. 2.28 ist noch einmal der Ausschnitt bis 5.5125 kHz aus dem Spektrum des Testsignals und der zugehörigen spektralen Hüllkurve dargestellt.

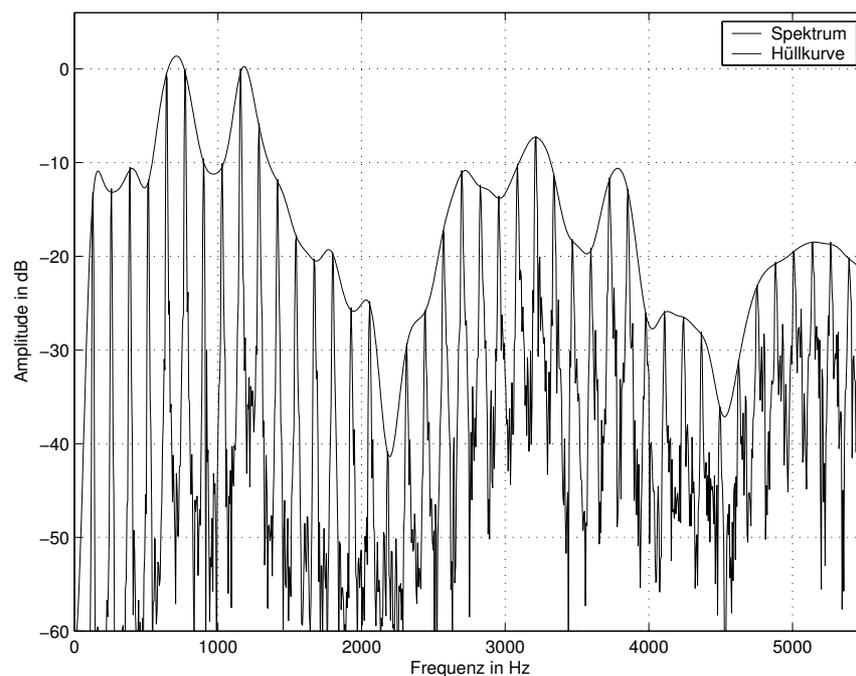


Fig. 2.28: Ausschnitt von 0 bis 5.5125 kHz aus dem Spektrum der spektralen Hüllkurve

In Abb. 2.29 ist das Ergebnis der Anpassung des Frequenzgangs einer Equalizerkette mit 5 Stufen auf diese spektrale Hüllkurve dargestellt. Als untere und obere Grenze für die bandpass-artige Fehlergewichtung wurden hier 600 und 4000 Hz gewählt, der Exponent c für die Fehlergewichtung wurde auf 1 gesetzt. Während der Optimierung versucht jede Equalizerstufe die Form der spektralen Hüllkurve in einer gewissen Umgebung um die Equalizer-Mittenfrequenz möglichst gut nachzubilden, wobei jedoch auch die Effekte der 'benachbarten' Stufen berücksichtigt werden. Da die Bänder hier auf die Maxima der spektralen Hüllkurve im relevanten Frequenzbereich initialisiert wurden (siehe gepunktete Linie), modelliert jedes Band nach der Optimierung einen Formanten. Der Einbruch/Antiformant bei ca. 2.2 kHz wird dabei natürlich nicht berücksichtigt. Hätte man den Exponent c in der Fehlergewichtung geringer, gewählt (z.B. $c = 0$ entsprechend einer gleichförmigen Fehlergewichtung), wären Abweichungen im Bereich des Antiformanten etwas stärker ins Gewicht gefallen und die Equalizer-Kette hätte versucht, über den globalen Gain-Faktor G die Kurve insgesamt tiefer zu legen und dies über höhere g_s in den einzelnen Stufen wieder auszugleichen. Dies wäre dann aber auf Kosten einer schlechteren Anpassung an den Formanten gegangen.

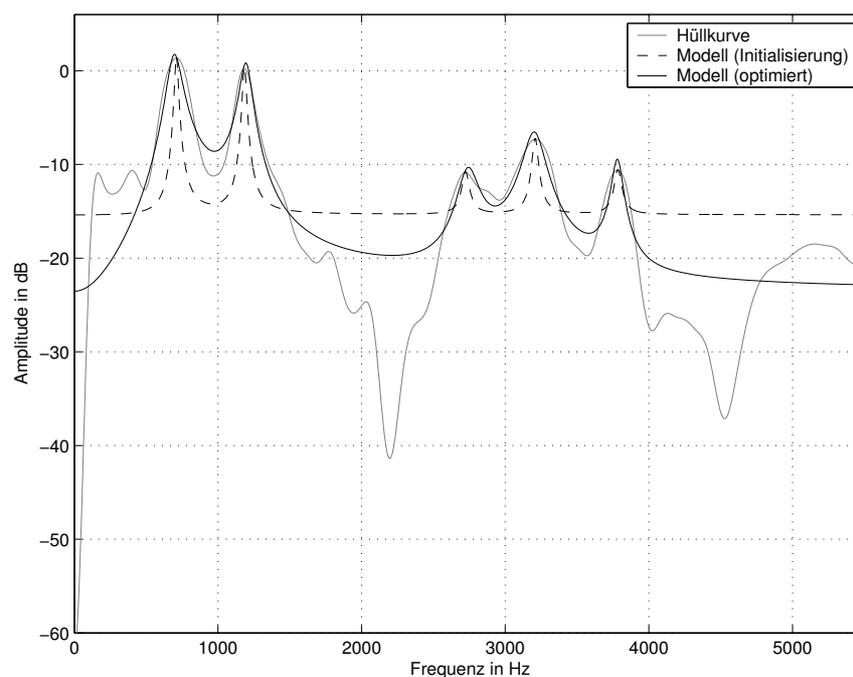


Fig. 2.29: Spektrale Hüllkurve und Frequenzgang einer Kette von 5 Equalizerstufen (5 Formanten) die auf diese Hüllkurve angepasst wurde

Eine bessere Möglichkeit um die Anpassung der Equalizer-Kurve auch in den Antiformant-Bereichen zu verbessern, liegt darin, diese Bereiche explizit zu modellieren, indem bestimmte Equalizerstufen auf die Minima der spektralen Hüllkurve initialisiert werden. In Abb. 2.29 wurden 6 Equalizerstufen benutzt wobei eine auf das Minimum der spektralen Hüllkurve initialisiert wurde. Wie im Ergebnis der Optimierung zu sehen ist, passt sich der Einbruch, der durch diese Stufe erzeugt wird, an die Form des Antiformanten an.

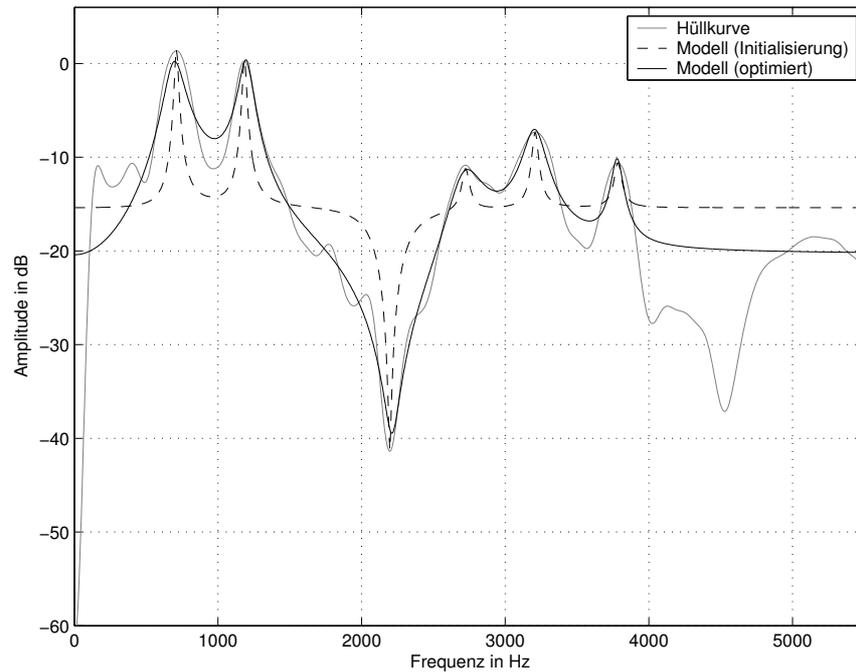


Fig. 2.30: Spektrale Hüllkurve und Frequenzgang einer Kette von 6 Equalizerstufen (5 Formanten, 1 Antiformant) die auf diese Hüllkurve angepasst wurde

Mit diesem Modell mit 5 Formant-Bändern und einem Antiformant-Band wollen wir nun unsere Transformationen durchführen. Es werden wieder dieselben 3 Transformationen als Beispiel herangezogen wie schon in den vorherigen Abschnitten, also die Verschiebung der zweiten Formantfrequenz um den Faktor 1.5, die Verringerung der zweiten Formant-Bandbreite um den Faktor 0.25 und die Verringerung der zweiten Formant-Intensität um den Faktor 0.25. Die Ergebnisse sind in den Abbildungen 2.32, 2.33 und 2.34 dargestellt. In Abb. 2.31 ist zu Vergleichszwecken noch einmal das Spektrum und der Modellfrequenzgang abgebildet.

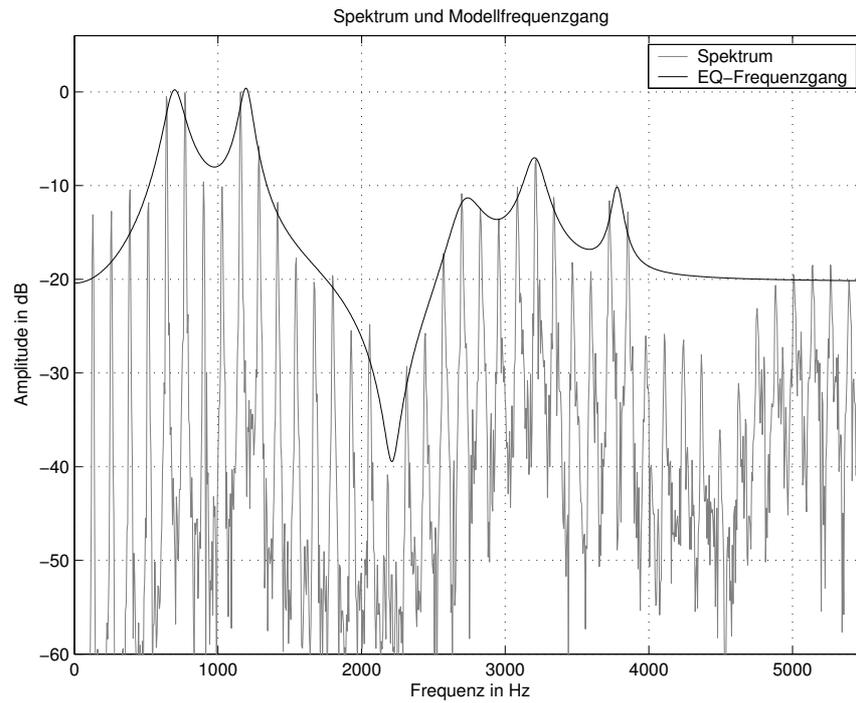


Fig. 2.31: Signalspektrum und Frequenzgang einer angepassten Kette von 6 Equalizerstufen

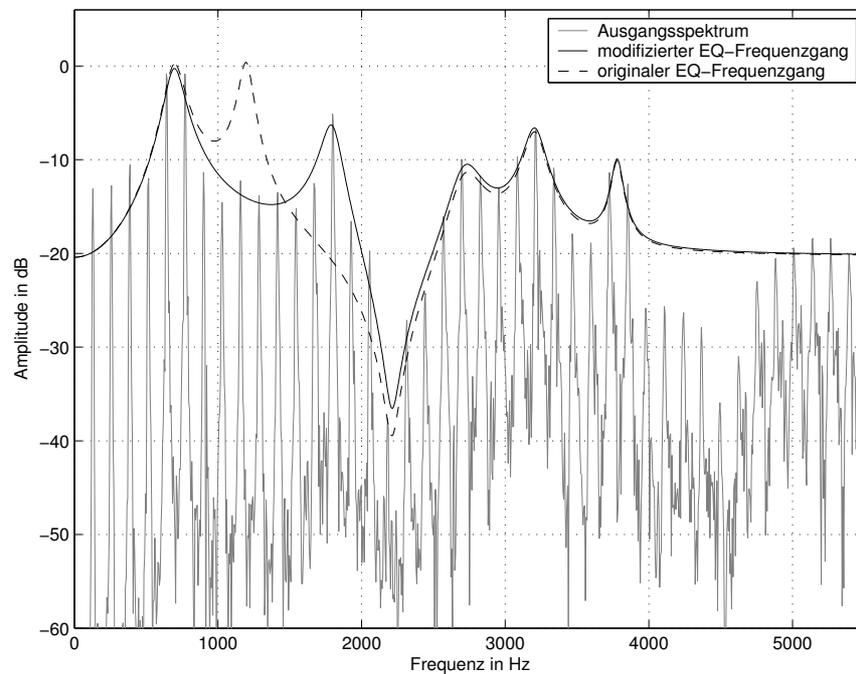


Fig. 2.32: Ausgangssignalspektrum und Frequenzgang des originalen und modifizierten Modell-Equalizers (2. Formantfrequenz um Faktor 1.5 erhöht)

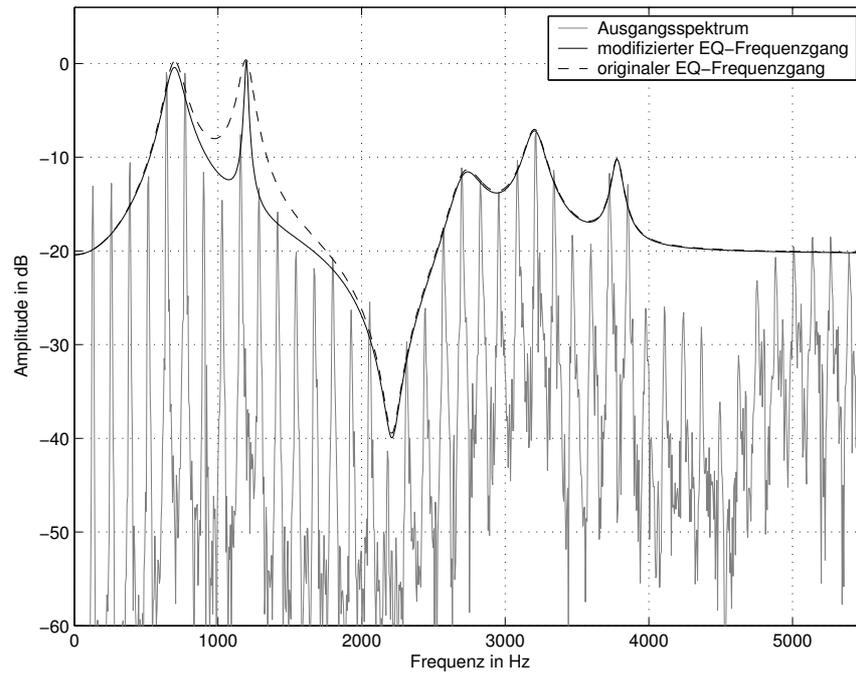


Fig. 2.33: Ausgangssignalspektrum und Frequenzgang des originalen und modifizierten Modell-Equalizers (2. Formant-Bandbreite um Faktor 0.25 gesenkt)

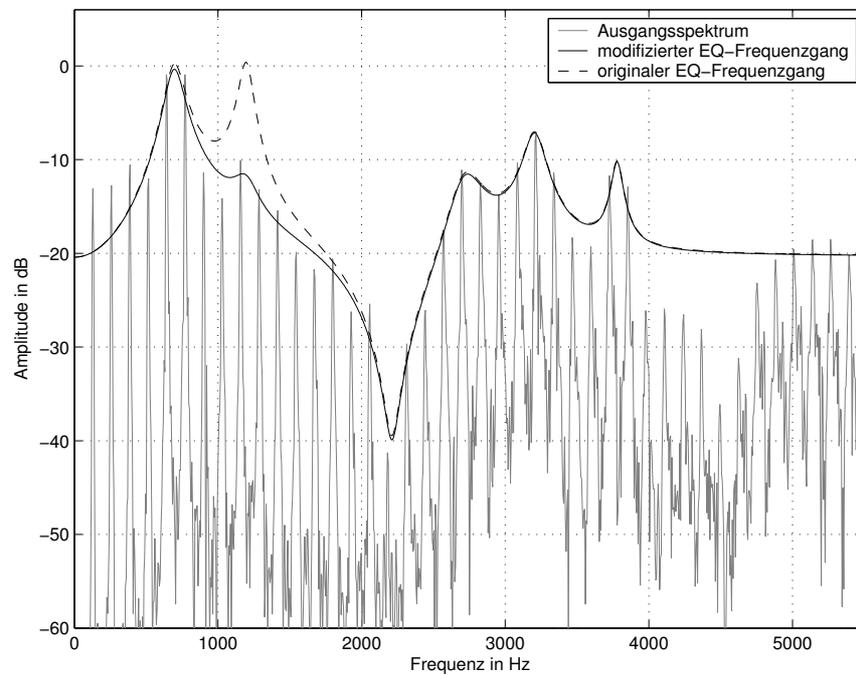


Fig. 2.34: Ausgangssignalspektrum und Frequenzgang des originalen und modifizierten Modell-Equalizers (2. Formant-Intensität um Faktor 0.25 gesenkt)

Die Vergleiche zwischen dem originalen Modellfrequenzgang und den jeweiligen transfor-

mierten Frequenzgängen zeigen, dass die Modellierung von Formanten durch Equalizer-Bänder zu Transformationsmöglichkeiten führt, die der intuitiven Erwartung recht gut entsprechen. Wie man sieht, wird bei Transformationen das Signalspektrum nur im Wirkungsbereich des modifizierten EQ-Bandes verändert - in den Bereichen außerhalb dieses Wirkungsbereichs bleibt es weitestgehend so wie es ist. Unerwünschte und schwer planbare Seiteneffekte bei der Veränderung einzelner Formant-Parameter treten hier also nicht auf.

2.3.7 Vor- und Nachteile des Modells

Vorteile

- Intuitiv einleuchtende Parametrisierung der Formanten.
- Formant-Frequenzen, -Bandbreiten und -Intensitäten unabhängig voneinander regelbar.
- Keine schwer vorhersagbaren Wechselwirkungen zwischen den Formanten (wie z.B. Interferenz zwischen Teilsignalen wie im parallelen Modell oder globale Veränderung der spektralen Hüllkurve wie beim seriellen Allpol-Modell).
- Antiformanten können mit denselben Mitteln ebenfalls modelliert werden.
- Flexible Gewichtung des Fehlers möglich.
- Bei Transformationen einzelner Teilfilter kein Whitening nötig.

Nachteile

- Aufwändige iterative Optimierung nötig.

2.4 Anwendungen und Ausblick

Anwendungen

Analyse von Formant-Parametern Die Extraktion der einzelnen Formant-Parameter aus der spektralen Hüllkurve kann als eine Art Feature-Extraktion angesehen werden. Solche Features spielen eine wichtige Rolle bei Klassifikationsproblemen. Beispielsweise könnten die Mittenfrequenzen der ersten 3 Formanten als Eingabevektor für einen Klassifikator zur Vokalerkennung dienen. Außerdem können Formanttabellen erstellt werden, in denen die typischen Formantfrequenzen verschiedener Vokale tabelliert sind, so wie dies z.B. im Anhang von [2] geschehen ist. Auch könnten Formant-Tabellen bestimmter Sprecher/Sänger oder auch von Instrumenten erstellt werden.

Synthese und Transformation anhand von analysierten Formant-Parametern Die reine Extraktion von bestimmten Parametern aus einem Signal ist für sich genommen natürlich keine besonders kreative Anwendung. Die Kenntnis solcher Parameter ermöglicht aber auch die gezielte Synthese von Klängen mit bestimmten gewünschten Eigenschaften. Beispielsweise kann es bei der subtraktiven Synthese nützlich sein, die Formant-Parameter von bestimmten akustischen Instrumenten zu kennen um deren Klang nachzuahmen. Transformationen von Klängen sind auch möglich. Wenn man beispielsweise die Formant-Parameter des Vokals 'a' und des Vokals 'e' kennt und ein 'a' als Eingangssignal hat, dann kann man es in ein 'e' transformieren indem man zuerst die 'a'-Formanten herausfiltert ('whitening') und anschließend die 'e'-Formanten aufprägt. Auch die Transformation der Stimme von einem Sprecher/Sänger zu einem anderen wäre so möglich - wobei dazu wohl noch etwas mehr gehört als nur die Anpassung der Formanten. Diese wäre jedoch sicherlich ein sehr wichtiger Teil eines solchen Voice-Transformers.

Freie Veränderung von Formanten Diese Anwendung ist genau das, was in den jeweiligen Praxistests der einzelnen Formantmodelle demonstriert wurde. Nachdem die Formant-Parameter extrahiert und ein entsprechend geweißtes Signal generiert wurde, können die Formanten anhand ihrer Parameter frei verändert werden und dann wieder auf das Signal aufgeprägt werden. Diese Anwendung ist sicherlich aus kreativer Sicht die interessanteste.

Ausblick

In der vorliegenden Arbeit wurden bestimmte Dinge, wie z.B. die Modellordnung und der interessierende Frequenzbereich von außen vorgegeben. Für bestimmte Anwendungen könnte es sinnvoll sein, auch diese Dinge automatisch zu bestimmen. So wäre es z.B. denkbar, die obere Grenzfrequenz des interessierenden Frequenzbereichs anhand eines spektralen Rolloff-Points festzulegen. Die Anzahl der zu modellierenden Formanten (und evtl. Antiformanten) könnte anhand von Schwellwerten für die relativen Amplituden der Maxima/Minima in der spektralen Hüllkurve automatisch bestimmt werden.

ANHANG

A. GRADIENTENBASIERTE MINIMIERUNG EINER KOSTENFUNKTION

In diesem Abschnitt sollen iterative Verfahren zur Minimierung einer Kostenfunktion vorgestellt werden. Solche Verfahren werden innerhalb dieser Arbeit benötigt, um den Frequenzgang der Equalizerkette an die spektrale Hüllkurve des Signals anzupassen. Allgemein werden solche Optimierungsverfahren immer dann benötigt, wenn man das Minimum (oder Maximum) einer Funktion nicht analytisch bestimmen kann, so wie es bei unserer Kostenfunktion (2.25) der Fall ist.

Optimierungsverfahren erster Ordnung

Gradientenabstieg Die einfachste Möglichkeit eine differenzierbare Funktion zu minimieren, besteht darin, mit irgendeinem Anfangswert (bzw. Parametervektor $\boldsymbol{\theta}_0$) zu starten, von dort aus die Richtung des steilsten Abstiegs zu berechnen (dies ist der negative Gradient) und einen kleinen Schritt in diese Richtung zu machen. Dies wird so lange wiederholt, bis man an einem Minimum angekommen ist (man merkt dies daran, dass der Gradient Null wird). Die Update-Regel beim Gradientenabstieg lautet demnach:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta \cdot \nabla E(\boldsymbol{\theta}_i)$$

wobei η eine kleine Konstante ist, die für das jeweilige Problem passend gewählt werden muss, $\boldsymbol{\theta}_i$ ist der Parametervektor, an dem wir gerade sind und $\boldsymbol{\theta}_{i+1}$ ist der neue (verbesserte) Parametervektor, zu dem wir hingehen. Es ist anschaulich, sich eine Art Gebirgslandschaft vorzustellen, in der wir ein (möglichst tiefes) Tal suchen. Es handelt sich demnach um ein iteratives Verfahren, d.h. man startet mit irgendeinem Initialisierungsvektor $\boldsymbol{\theta}_0$ und verbessert diesen Schritt für Schritt - der Index i steht hier für die aktuelle Iteration.

Abstiegsverfahren allgemein Der oben beschriebene Gradientenabstiegs-Algorithmus ist der mit Abstand einfachste Vertreter einer ganzen Klasse von Optimierungsverfahren, die alle versuchen aus einem anfänglichen Parametervektor $\boldsymbol{\theta}_0$ durch schrittweise Verbesserung ein Minimum der Kostenfunktion zu finden. Allgemein benötigt ein solches Optimierungsverfahren in jeder Iteration erstens eine Richtung \mathbf{d}_i , in die der Parametervektor $\boldsymbol{\theta}$ bewegt wird und zweitens eine Schrittweite α_i , die sagt, wie weit in diese Richtung gegangen werden soll. Die verallgemeinerte Update-Regel lautet demnach:

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha_i \cdot \mathbf{d}_i$$

wobei beim Gradientenabstieg $\mathbf{d}_i = -\nabla E(\boldsymbol{\theta}_i)$ und $\alpha_i = \eta$ gilt. Es gibt zahlreiche heuristische Verbesserungsvorschläge für solche Gradientenabstiegsverfahren, die zu schnellerer Konvergenz führen (z.B. Einführung von Momentum-Termen, Adaption der Lernrate, etc.). Eine Übersicht über solche Verfahren findet sich in [5] - dort werden diese Verfahren zwar im Zusammenhang mit Neuronalen Netzen und dem Backpropagation-Algorithmus betrachtet, da der Backpropagation-Algorithmus aber nichts anderes als ein Algorithmus zur Gradientenberechnung ist, können viele Dinge auch bei anderen Optimierungsproblemen angewandt werden.

Es gibt jedoch noch deutlich bessere Möglichkeiten (im Sinne von schnellerer Konvergenz des Verfahrens), die Richtung \mathbf{d}_i und Schrittweite α_i zu wählen. Dies sind die so genannten Verfahren zweiter Ordnung und sollen im Folgenden betrachtet werden.

Minimierung quadratischer Funktionen

Quadratische Form Zur Motivation von Optimierungsverfahren zweiter Ordnung wollen wir zunächst annehmen, dass unsere Kostenfunktion eine quadratische Funktion des Parametervektors dargestellt. Wir nennen diese quadratische Kostenfunktion $E_q(\boldsymbol{\theta})$, um sie von der tatsächlichen Kostenfunktion $E(\boldsymbol{\theta})$ zu unterscheiden. Die allgemeine Form einer solchen quadratischen Funktion sieht wie folgt aus:

$$E_q(\boldsymbol{\theta}) = c - \mathbf{b}^T \boldsymbol{\theta} + \frac{1}{2} \boldsymbol{\theta}^T \mathbf{A} \boldsymbol{\theta} \quad (\text{A.1})$$

wobei der Skalar c , der Vektor \mathbf{b} und die Matrix \mathbf{A} als gegeben angenommen werden. Für die Matrix \mathbf{A} wollen wir zusätzlich die Annahme machen, dass sie positiv definit ist. In diesem Fall beschreibt die Gleichung einen nach oben offenen Hyperparaboloiden (anschaulich stelle man sich einfach einen nach oben offenen Paraboloiden über einer (w_1, w_2) -Ebene vor). Um das Minimum dieser Funktion bezüglich des Parametervektors $\boldsymbol{\theta}$ zu finden, leiten wir die Funktion nach $\boldsymbol{\theta}$ ab:

$$\mathbf{g}(\boldsymbol{\theta}) := \nabla E|_{\boldsymbol{\theta}} = \frac{\partial E_q(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\mathbf{b} + \mathbf{A} \boldsymbol{\theta} \quad (\text{A.2})$$

wobei $\mathbf{g}(\boldsymbol{\theta})$ als der lokale Gradient an der Stelle $\boldsymbol{\theta}$ definiert wurde. Wenn wir diese Ableitung Null setzen, kommen wir zu dem linearen Gleichungssystem:

$$\mathbf{A} \boldsymbol{\theta} = \mathbf{b} \quad (\text{A.3})$$

dessen Lösungsvektor $\boldsymbol{\theta}_{opt}$ unser gesuchtes Optimum ist. Im Falle einer positiv definiten Matrix \mathbf{A} ist dieses Optimum ein Minimum (und nicht etwa ein Maximum oder Sattelpunkt). Es besteht demnach eine Äquivalenz zwischen der Optimierung einer quadratischen Funktion und der Lösung eines linearen Gleichungssystems. Im Prinzip ist ein solches lineares Gleichungssystem mit $\boldsymbol{\theta}_{opt} = \mathbf{A}^{-1} \mathbf{b}$ ja analytisch lösbar, jedoch kann die Matrixinversion bei großen Matrizen extrem aufwändig werden (der Aufwand skaliert mit $\mathcal{O}(N^3)$). Die zweite Ableitung der Funktion bezüglich des Parametervektors $\boldsymbol{\theta}$ ist die so genannte Hesse-Matrix \mathbf{H} und in Falle der quadratischen Funktion einfach gegeben durch:

$$\mathbf{H}(\boldsymbol{\theta}) := \nabla \nabla E|_{\boldsymbol{\theta}} = \mathbf{A} \quad (\text{A.4})$$

ist hier also eine konstante Matrix bezüglich $\boldsymbol{\theta}$. Im Folgenden sollen effiziente Verfahren betrachtet werden, um eine solche quadratische Form zu minimieren. In der Tat wurden die im Folgenden vorgestellten Optimierungsverfahren zweiter Ordnung ursprünglich als iterative Lösungsmethoden für lineare Gleichungssysteme entwickelt.

Optimierungsverfahren zweiter Ordnung

Quadratische Approximation der Kostenfunktion Nun ist die Annahme, dass unsere Kostenfunktion einfach eine quadratische Funktion der Parameter ist, natürlich in den meisten Fällen falsch. Die folgenden Betrachtungen sind jedoch trotzdem nützlich, weil wir in der Nähe unseres Arbeitspunktes $\boldsymbol{\theta}_i$ eine quadratische Taylor-Entwicklung unserer Kostenfunktion ansetzen können, d.h. die Basis aller Optimierungsverfahren zweiter Ordnung ist eine quadratische Approximation der Kostenfunktion um unseren aktuellen Arbeitspunkt $\boldsymbol{\theta}_i$ der Form:

$$E_q(\boldsymbol{\theta}) = E(\boldsymbol{\theta}_i) + (\boldsymbol{\theta} - \boldsymbol{\theta}_i)^T \mathbf{g}_i + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_i)^T \mathbf{H}_i(\boldsymbol{\theta} - \boldsymbol{\theta}_i) \quad (\text{A.5})$$

mit \mathbf{g}_i als dem lokalen Gradienten der Kostenfunktion an der Stelle $\boldsymbol{\theta}_i$:

$$\mathbf{g}_i := \nabla E|_{\boldsymbol{\theta}_i} \quad (\text{A.6})$$

und \mathbf{H}_i als der lokalen Hesse-Matrix an der Stelle $\boldsymbol{\theta}_i$ mit den Elementen:

$$(\mathbf{H}_i)_{kj} := \left. \frac{\partial^2 E}{\partial \theta_k \partial \theta_j} \right|_{\boldsymbol{\theta}_i} \quad (\text{A.7})$$

Um eventuell auftretender Verwirrung vorzubeugen: Bei quadratischen Funktionen ist die Hesse-Matrix unabhängig vom aktuellen Parametervektor $\boldsymbol{\theta}_i$ und damit auch unabhängig von der Iteration i , bei allgemeinen Funktionen ist die Hesse-Matrix jedoch eine lokale Größe. Je nachdem, was wir gerade betrachten, wird der Index i mitgeschrieben oder auch nicht.

Line Search und Optimierung der Lernrate Bei den Optimierungsverfahren erster Ordnung wird die Lernschrittweite α_i entweder als Konstante festgelegt oder in irgendeiner heuristischen Art und Weise während des Lernprozesses verändert. Beim Line-Search Verfahren wird für jeden Lernschritt eine optimale Lernrate bestimmt, in dem Sinne, dass entlang einer zuvor festgelegten Suchrichtung \mathbf{d}_i ein (lokales) Minimum erreicht wird. Es wird deshalb in jeder Iteration die optimale Schrittweite bestimmt, gemäß:

$$\alpha_i = \underset{\alpha}{\operatorname{argmin}} E(\boldsymbol{\theta}_i + \alpha \mathbf{d}_i)$$

Diese Gleichung ist erstmal als eine Forderung an unser α_i zu verstehen und macht noch keine Aussage darüber, wie wir dieses α_i bestimmen könnten. Eine einfache Idee wäre es, einfach solange mit einer bestimmten Schrittweite in Richtung \mathbf{d}_i zu gehen, wie es bergab geht. Dies ist jedoch ziemlich aufwändig, da ja nach jedem Schritt der neue Wert der Kostenfunktion berechnet werden müsste. Etwas ausgereifere Verfahren suchen zuerst drei

Punkte entlang der Suchrichtung, legen durch diese eine Parabel (dies ist eindimensionale Funktion, da sie nur von α abhängt) und bestimmen das Minimum dieser Parabel. Dieser neue Punkt wird zusammen mit zwei von den alten für eine nächste Parabel-Approximation benutzt. Dann wird das Minimum dieser neuen Parabel bestimmt, usw. - solche Verfahren bezeichnet man im allgemeinen als Line-Search Verfahren, da sie ein Minimum entlang einer zuvor festgelegten Linie suchen. Wenn wir allerdings von einer quadratischen Kostenfunktion ausgehen, dann können wir die optimale Schrittweite auch explizit ausrechnen. Wenn wir nämlich in unserer quadratischen Approximation A.5 einfach $\boldsymbol{\theta} = \boldsymbol{\theta}_i + \alpha \mathbf{d}_i$ einsetzen ergibt sich:

$$\begin{aligned} E_q(\boldsymbol{\theta}_i + \alpha \mathbf{d}_i) &= E(\boldsymbol{\theta}_i) + (\boldsymbol{\theta}_i + \alpha \mathbf{d}_i - \boldsymbol{\theta}_i)^T \mathbf{g}_i + \frac{1}{2} (\boldsymbol{\theta}_i + \alpha \mathbf{d}_i - \boldsymbol{\theta}_i)^T \mathbf{H}_i (\boldsymbol{\theta}_i + \alpha \mathbf{d}_i - \boldsymbol{\theta}_i) \\ &= E(\boldsymbol{\theta}_i) + (\alpha \mathbf{d}_i)^T \mathbf{g}_i + \frac{1}{2} (\alpha \mathbf{d}_i)^T \mathbf{H}_i (\alpha \mathbf{d}_i) \end{aligned}$$

diesen Ausdruck können wir nach α ableiten:

$$\frac{\partial}{\partial \alpha} E_q(\boldsymbol{\theta}_i + \alpha \mathbf{d}_i) = \mathbf{d}_i^T \mathbf{g}_i + \alpha \mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i$$

Nullsetzen und Umstellen nach α ergibt unsere gesuchte optimale Schrittweite α_i :

$$\boxed{\alpha_i = -\frac{\mathbf{d}_i^T \mathbf{g}_i}{\mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i}} \quad (\text{A.8})$$

Die Bedingung dafür, dass wir tatsächlich auf ein Minimum zusteuern (und nicht etwa auf ein Maximum oder einen Sattelpunkt) ist die, dass die Hesse-Matrix \mathbf{H}_i positiv definit ist. Später werden wir ein Verfahren betrachten in welchem die Hesse-Matrix modifiziert wird, um auch dann abwärts zu gehen, wenn die Hesse-Matrix nicht positiv definit ist, so dass dieses Verfahren insgesamt robuster ist.

Conjugate Direction Verfahren Wenn wir mit N die Dimensionalität unseres Parameterraums bezeichnen, dann kann man für quadratische Funktionen mit positiv definiten Hesse-Matrix zeigen (siehe [17] oder [1]), dass man mit einer derart gewählten Lernschrittweite nach spätestens N Schritten im globalen Minimum ankommt, wenn die Suchrichtungen bei den einzelnen Lernschritten folgende Bedingung erfüllen:

$$\mathbf{d}_i^T \mathbf{H} \mathbf{d}_j = 0 \quad \text{für } i \neq j \quad (\text{A.9})$$

Man spricht in diesem Falle davon, dass die Richtungen \mathbf{d}_i $i = 1, \dots, N$ zueinander konjugiert sind bezüglich der Matrix \mathbf{H} . Man beachte, dass sich diese Bedingung bei $\mathbf{H} = \mathbf{I}$ mit der Einheitsmatrix \mathbf{I} auf die normale Definition von Orthogonalität von Vektoren reduziert. Daher spricht man auch gelegentlich von Orthogonalität bezüglich der Matrix \mathbf{H} . Wenn es also gelingt einen Satz von N Suchrichtungen zu erzeugen, die obige Bedingung erfüllen, dann kann eine quadratische (Kosten-)Funktion in N Schritten minimiert werden.

Conjugate Gradient Verfahren Die Frage ist nun, wie man einen Satz von Suchrichtungen finden kann, die obiger Bedingung genügen. Die Lösung liegt darin, als erste Suchrichtung den negativen Gradienten zu wählen: $\mathbf{d}_1 = -\mathbf{g}_1$, und für alle weiteren Suchrichtungen eine Linearkombination aus der vorherigen Suchrichtung und dem aktuellen negativen Gradienten zu wählen:

$$\boxed{\mathbf{d}_{i+1} = -\mathbf{g}_{i+1} + \beta_i \mathbf{d}_i} \quad (\text{A.10})$$

wobei die Koeffizienten β_i durch Einsetzen der Bedingung A.9 gefunden werden können. Dies führt zu:

$$\beta_i = \frac{\mathbf{g}_{i+1}^T \mathbf{H} \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{H} \mathbf{d}_i}$$

Dies garantiert erstmal, dass zwei aufeinanderfolgende Richtungen konjugiert zueinander sind. Man kann jedoch zeigen, dass diese Eigenschaft auch für alle paarweise verschiedenen Suchrichtungen gilt. Durch verschiedene Umformungen kann man noch weitere äquivalente Formeln für β_i herleiten. Es gilt:

$$\boxed{\begin{aligned} \beta_i &= \frac{\mathbf{g}_{i+1}^T \mathbf{H} \mathbf{d}_i}{\mathbf{d}_i^T \mathbf{H} \mathbf{d}_i} && (\text{originale Form}) \\ &= \frac{\mathbf{g}_{i+1}^T (\mathbf{g}_{i+1} - \mathbf{g}_i)}{\mathbf{d}_i^T (\mathbf{g}_{i+1} - \mathbf{g}_i)} && (\text{Hestenes-Stiefel Form}) \\ &= \frac{\mathbf{g}_{i+1}^T (\mathbf{g}_{i+1} - \mathbf{g}_i)}{\mathbf{g}_i^T \mathbf{g}_i} && (\text{Polak-Ribière Form}) \\ &= \frac{\mathbf{g}_{i+1}^T \mathbf{g}_{i+1}}{\mathbf{g}_i^T \mathbf{g}_i} && (\text{Fletcher-Reeves Form}) \end{aligned}} \quad (\text{A.11})$$

Obwohl all diese Formeln für β im Falle einer quadratischen Kostenfunktion äquivalent sind, macht es bei der Anwendung auf nichtquadratische Funktionen sehr wohl einen Unterschied (hinsichtlich der Konvergenzeigenschaften), welche Form man benutzt. In der Literatur wird meist der Polak-Ribière Form die beste Konvergenz zugeschrieben. Daher werden wir auch diese verwenden. Bei der Gleichung zur Berechnung von α_i und auch in der originalen Form der Gleichung für β_i tritt das Matrix-Vektor-Produkt $\mathbf{d}_i^T \mathbf{H} \mathbf{d}_i$ auf - wir haben jedoch noch nichts dazu gesagt, wie wir dieses berechnen können. Klar, im Falle einer bekannten quadratischen Funktion der Form (A.1) mit bekannter Matrix $\mathbf{H} = \mathbf{A}$ können wir es einfach und direkt berechnen. Aber wir wollen ja letztendlich darauf hinaus, eine unbekannte, nichtquadratische Kostenfunktion zu minimieren (die wir in der Umgebung unseres Arbeitspunktes durch eine quadratische Taylor-Entwicklung approximieren). Die Bezeichnung 'unbekannt' soll in diesem Zusammenhang so verstanden werden, dass wir uns die Funktion einfach als Black-Box vorstellen können, in die wir beliebige Eingabevektoren hineingeben können und als Ausgabe den dazugehörigen Funktionswert und Gradienten erhalten - in unserem Falle ist die Black-Box natürlich eher eine 'White-Box', da uns ja mit Gleichung (2.25) die Funktion bekannt ist. Grundsätzlich ist es natürlich für viele Funktionen möglich, für die Hesse-Matrix \mathbf{H}_i an der Stelle $\boldsymbol{\theta}_i$ explizite Gleichungen anzugeben. Meistens wird dies jedoch sehr aufwändig sein. Zum Glück gibt es eine relativ einfache Möglichkeit, das Produkt aus Hesse-Matrix und einem beliebigen Vektor zu approximieren. Sei \mathbf{v} ein beliebiger Vektor. Dann gilt:

$$\mathbf{v}^T \mathbf{H} \equiv \mathbf{v}^T \nabla(\nabla E)$$

Dies können wir an der Stelle $\boldsymbol{\theta}$ durch endliche Differenzen approximieren (siehe [1]):

$$\mathbf{v}^T \nabla(\nabla E) = \frac{\nabla E(\boldsymbol{\theta} + \epsilon \mathbf{v}) - \nabla E(\boldsymbol{\theta})}{\epsilon} + \mathcal{O}(\epsilon)$$

wobei ϵ eine kleine Konstante ist. Für eine höhere Genauigkeit der Approximation können wir auch eine zweiseitige Differenz nehmen:

$$\mathbf{v}^T \nabla(\nabla E) = \frac{\nabla E(\boldsymbol{\theta} + \epsilon \mathbf{v}) - \nabla E(\boldsymbol{\theta} - \epsilon \mathbf{v})}{2\epsilon} + \mathcal{O}(\epsilon^2)$$

Dies würde allerdings eine weitere Gradientenberechnung (an der Stelle $\boldsymbol{\theta} - \epsilon \mathbf{v}$) erfordern. Da wir es aber ohnehin nicht wirklich mit einer quadratischen Funktion zu tun haben und daher sowieso alles nur näherungsweise gilt, erscheint der erhöhte Aufwand der zusätzlichen Gradientenberechnung nicht gerechtfertigt. In der Praxis sollte ϵ umgekehrt proportional zur Länge des Vektors \mathbf{v} gewählt werden, also $\epsilon \propto 1/\|\mathbf{v}\|$). Deshalb werden wir in der Implementation die Näherung:

$$\boxed{\mathbf{d}_i^T \mathbf{H}_i \approx \frac{\mathbf{g}(\boldsymbol{\theta}_i + \epsilon' \mathbf{d}_i) - \mathbf{g}(\boldsymbol{\theta}_i)}{\epsilon'}} \quad (\text{A.12})$$

mit $\epsilon' = \epsilon/\|\mathbf{d}\|$ benutzen (zur Erinnerung $\mathbf{g}(\boldsymbol{\theta}_i) \equiv \nabla E(\boldsymbol{\theta}_i) \equiv \nabla E|_{\boldsymbol{\theta}_i}$ per Definition). Die Gleichungen (A.12) (A.8), (A.10) und (A.11) lassen sich zu dem Conjugate Gradient Algorithmus (Algorithmus 5) zusammenfassen. Bei der Anwendung dieses Algorithmus auf nichtquadratische Funktionen ist man natürlich nach N Schritten im allgemeinen noch nicht in einem Minimum angekommen. Es ist gängige Praxis, die Suchrichtung nach jeweils N Lernschritten wieder mit dem negativen Gradienten zu initialisieren und das Verfahren zu wiederholen. Einen kleinen Haken hat die Sache jedoch. Für den Fall, dass die (lokale) Hesse-Matrix nämlich nicht positiv definit ist, ist nicht garantiert, dass der Algorithmus auch tatsächlich eine Richtung auswählt in der es 'bergab' geht. Daher ist es in der Praxis im allgemeinen nötig, die optimale Lernschrittweite α über Line-Search Verfahren zu bestimmen anstatt über Gleichung (A.8), wie dies im hier abgedruckten Falle gemacht wird. Oder man benutzt das...

Scaled Conjugate Gradient (SCG) Verfahren Beim Scaled Conjugate Gradient Verfahren wird mit einer modifizierten Version der Hesse-Matrix gearbeitet, bei der ein Vielfaches der Einheitsmatrix addiert wird, so dass die resultierende Matrix (die wir \mathbf{H}' nennen wollen) positiv definit wird. Wir setzen also:

$$\mathbf{H}' := \mathbf{H} + \lambda \mathbf{I}$$

Diese Matrix kann man immer positiv definit machen, einfach indem man λ hinreichend groß wählt. Damit ergibt sich für die Berechnung der Schrittweite zu:

$$\begin{aligned} \alpha_i &= -\frac{\mathbf{d}_i^T \mathbf{g}_i}{\mathbf{d}_i^T \mathbf{H}'_i \mathbf{d}_i} \\ &= -\frac{\mathbf{d}_i^T \mathbf{g}_i}{\mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i + \lambda_i \|\mathbf{d}_i\|^2} \end{aligned}$$

Algorithmus 5 Optimierung via Conjugate Gradient Verfahren

 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$ $\mathbf{g} \leftarrow \nabla E(\boldsymbol{\theta})$ $\mathbf{d} \leftarrow -\mathbf{g}$ $\epsilon \leftarrow$ kleine positive Zahlconverged \leftarrow **false****while** not converged **do** approximiere Matrix-Vektor Produkt $\mathbf{d}^T \mathbf{H}$ (A.12): $\epsilon' \leftarrow \epsilon / \|\mathbf{d}\|$ $\mathbf{d}^T \mathbf{H} \leftarrow \frac{\mathbf{g}(\boldsymbol{\theta} + \epsilon' \mathbf{d}) - \mathbf{g}}{\epsilon'}$

berechne Lernschrittweite (A.8):

 $\alpha \leftarrow -\frac{\mathbf{d}^T \mathbf{g}}{\mathbf{d}^T \mathbf{H} \mathbf{d}}$

ACHTUNG: diese Art der Berechnung der Schrittweite kann bei nichtquadratischen Funktionen zu einer Erhöhung der Kosten führen - für robustere Optimierung werden hier normalerweise Line-Search Verfahren angewandt.

führe den Lernschritt aus:

 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \mathbf{d}$ berechne den Gradienten an der neuen Position (merke dabei den alten für die Berechnung von β): $\mathbf{g}_{old} \leftarrow \mathbf{g}$ $\mathbf{g} \leftarrow \nabla E|_{\boldsymbol{\theta}}$ berechne β nach der Polak-Ribière Form (A.11): $\beta \leftarrow \frac{\mathbf{g}^T (\mathbf{g} - \mathbf{g}_{old})}{\mathbf{g}_{old}^T \mathbf{g}_{old}}$

berechne die neue Suchrichtung für die nächste Iteration (A.10):

 $\mathbf{d}_{old} \leftarrow \mathbf{d}$ $\mathbf{d} \leftarrow -\mathbf{g} + \beta \mathbf{d}_{old}$

überprüfe ein Konvergenzkriterium:

 converged $\leftarrow (\mathbf{g} \approx \mathbf{0})$ **end while**

Dies stellt uns vor das Problem, dass jetzt (in jedem Lernschritt) ein passender Wert für λ_i gewählt werden muss. Dazu definieren wir den Nenner in obiger Gleichung als δ_i :

$$\boxed{\delta_i = \mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i + \lambda_i \|\mathbf{d}_i\|^2} \quad (\text{A.13})$$

so dass:

$$\boxed{\alpha_i = -\frac{\mathbf{d}_i^T \mathbf{g}_i}{\delta_i}} \quad (\text{A.14})$$

Bei einer positiv definiten Matrix \mathbf{H}' ist auch dieser Nenner δ_i positiv. Dies legt nahe, λ_i so zu wählen, dass $\delta_i > 0$ gilt. Wenn wir also beispielsweise bei Iteration i feststellen, dass $\delta_i < 0$ ist, erhöhen wir den Wert von λ_i , diesen erhöhten Wert bezeichnen wir mit $\bar{\lambda}_i$. Der dazugehörige erhöhte Wert von δ_i sei mit $\bar{\delta}_i$ bezeichnet und es gilt:

$$\bar{\delta}_i = \delta_i + (\bar{\lambda}_i - \lambda_i) \|\mathbf{d}_i\|^2$$

In [9] wird $\bar{\lambda}_i$ derart gewählt, dass, für den Fall dass $\delta_i < 0$ ist, für den erhöhten Wert $\bar{\delta}_i$ gilt:

$$\bar{\delta}_i = -\mathbf{d}_i^T \mathbf{H}_i \mathbf{d}_i$$

Dies führt zu folgender Gleichung für $\bar{\lambda}_i$:

$$\boxed{\bar{\lambda}_i = 2 \left(\lambda_i - \frac{\delta_i}{\|\mathbf{d}_i\|^2} \right) \quad \text{für } \delta_i < 0} \quad (\text{A.15})$$

Damit haben wir nun also eine Regel um λ_i zu erhöhen, so dass \mathbf{H}' in jedem Fall positiv definit ist. Was also noch fehlt, ist eine Regel, wann und wie wir λ wieder absenken können. Im Scaled Conjugate Gradient Algorithmus hat der Skalierungs-Parameter λ zwei Funktionen: einerseits soll er sicherstellen, dass \mathbf{H}' in jedem Schritt positiv definit ist, andererseits soll er Schrittweite α dann reduzieren, wenn die quadratische Approximation der Kostenfunktion schlecht ist - und da λ im Nenner für die Berechnung von α vorkommt, tut er dies, wenn er einen großen Wert annimmt. Dort wo die quadratische Approximation gut passt, sollte α jedoch seinen ursprünglichen Wert nach Gleichung (A.8) annehmen - dies ist für kleine λ der Fall. Daher benötigen wir zunächst ein Maß für die Güte unserer quadratischen Approximation. Dieses wollen wir (zur Iteration i) mit Δ_i bezeichnen. Beim SCG-Algorithmus wird der Quotient aus tatsächlicher Fehler-Verringerung und (durch die Approximation) vorhergesagter Fehler-Verringerung als Vergleichsparameter benutzt:

$$\begin{aligned} \Delta_i &= \frac{\Delta E}{\Delta E_q} \\ &= \frac{E(\boldsymbol{\theta}_i) - E(\boldsymbol{\theta}_i + \alpha_i \mathbf{d}_i)}{E(\boldsymbol{\theta}_i) - E_q(\boldsymbol{\theta}_i + \alpha_i \mathbf{d}_i)} \end{aligned}$$

wobei E_q wieder die quadratische Approximation (A.5) sein soll. Einsetzen führt zu:

$$\boxed{\Delta_i = -\frac{2[E(\boldsymbol{\theta}_i) - E(\boldsymbol{\theta}_i + \alpha_i \mathbf{d}_i)]}{\alpha_i \mathbf{d}_i^T \mathbf{g}_i}} \quad (\text{A.16})$$

Dieser Vergleichsparameter ist nahe 1, wenn die quadratische Approximation und die tatsächliche Kostenfunktion sich ungefähr gleich verhalten. In diesem Fall kann λ reduziert

werden. Anderenfalls sollte die Schrittweite α verkleinert, und damit λ also erhöht werden. In [9] wird folgende Regel vorgeschlagen:

$$\boxed{\begin{array}{ll} \text{wenn } \Delta_i > 0.75 & \text{dann } \lambda_{i+1} = \lambda_i/2 \\ \text{wenn } \Delta_i < 0.25 & \text{dann } \lambda_{i+1} = 4\lambda_i \end{array}} \quad (\text{A.17})$$

Wenn $\Delta_i < 0$ wird, dann würde der Lernschritt zu einer Erhöhung der Kostenfunktion führen - dies ist natürlich nicht erwünscht. Daher wird in diesem Fall kein Lernschritt ausgeführt, sondern stattdessen erstmal λ nach (A.17) erhöht, eine entsprechende neue Lernschrittweite α berechnet und der Ausdruck für Δ neu ausgewertet. Dies wird so lange fortgesetzt, bis Δ positiv wird - erst dann wird tatsächlich ein Lernschritt ausgeführt. Wir haben nun also Regeln, um den Skalierungsparameter λ während der Optimierung anzupassen. In der Praxis sollte man den Bereich für λ nach oben und unten begrenzen, da es sonst bei endlicher Zahlendarstellung irgendwann dazu kommen kann, dass λ exakt Null wird oder den ∞ -Wert annimmt. Von diesen Werten würde λ dann nicht mehr entkommen können, da der Computer ja generell $c \cdot 0 = 0$ und $\infty/c = \infty$ für eine beliebige Konstante $c > 0$ berechnet. Als Startwert wird in der Regel eine kleine Konstante $\lambda_1 > 0$ gewählt. In Algorithmus 6 ist der SCG-Algorithmus in Pseudo-Code dargestellt. Die zugehörigen Initialisierungen sind aus Gründen besserer Text-Formatierung aus dem Algorithmus selbst herausgenommen und folgen direkt hier:

```

 $\theta$   $\leftarrow$   $\theta_0$ 
 $\epsilon$   $\leftarrow$  kleine positive Zahl
 $\lambda$   $\leftarrow$  kleine positive Zahl
 $\mathbf{g}$   $\leftarrow$   $\nabla E(\boldsymbol{\theta})$ 
 $\mathbf{g}_{old}$   $\leftarrow$   $\mathbf{g}$ 
 $\mathbf{d}$   $\leftarrow$   $-\mathbf{g}$ 
 $\mathbf{d}_{old}$   $\leftarrow$   $\mathbf{d}$ 
converged  $\leftarrow$  false

```

Algorithmus 6 Optimierung via Scaled Conjugate Gradient Verfahren**while** not converged **do**approximiere Matrix-Vektor Produkt $\mathbf{d}^T \mathbf{H}$ (A.12):

$$\epsilon' \leftarrow \epsilon / \|\mathbf{d}\|$$

$$\mathbf{d}^T \mathbf{H} \leftarrow \frac{\mathbf{g}(\boldsymbol{\theta} + \epsilon' \mathbf{d}) - \mathbf{g}}{\epsilon'}$$

berechne den Nenner δ für die Lernschrittweite (A.13):

$$\delta = \mathbf{d}^T \mathbf{H} \mathbf{d} + \lambda \|\mathbf{d}\|^2$$

prüfe, ob δ positiv (entsprechend einer positiv definiten Matrix \mathbf{H}'), wenn nicht, erhöhe λ (A.15) und berechne δ neu:**if** $\delta < 0$ **then**

$$\lambda \leftarrow 2\left(\lambda - \frac{\delta}{\|\mathbf{d}\|^2}\right)$$

$$\delta = \mathbf{d}^T \mathbf{H} \mathbf{d} + \lambda \|\mathbf{d}\|^2$$

end if

berechne Lernschrittweite (A.14):

$$\alpha = -\frac{\mathbf{d}^T \mathbf{g}}{\delta}$$

berechne den Vergleichsparameter Δ (A.16):

$$\Delta = -\frac{2[E(\boldsymbol{\theta}) - E(\boldsymbol{\theta} + \alpha \mathbf{d})]}{\alpha \mathbf{d}^T \mathbf{g}}$$

Wenn der Lernschritt zu einer Verringerung der Kostenfunktion führt, führe ihn aus. Wenn er aber eine Vergrößerung der Kostenfunktion bewirkt (dies ist der Fall für $\Delta < 0$), erhöhe λ und springe aus der aktuellen Iteration der Schleife heraus ohne Lernschritt:**if** $\Delta > 0$ **then**

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \alpha \mathbf{d}$$

erhöhe oder senke λ (Eq. A.17) entsprechend der Güte der quadratischen Approximation:**if** $\Delta > 0.75$ **then**

$$\lambda \leftarrow \lambda/2$$

else if $\Delta < 0.25$ **then**

$$\lambda \leftarrow 4\lambda$$

end if**else**

$$\lambda \leftarrow 4\lambda$$

continue**end if**Es wurde ein erfolgreicher Lernschritt ausgeführt. Berechne Gradienten an der neuen Position (merke dabei den alten Gradienten für die Berechnung von β):

$$\mathbf{g}_{old} \leftarrow \mathbf{g}$$

$$\mathbf{g} \leftarrow \nabla E|_{\boldsymbol{\theta}}$$

berechne β nach der Polak-Ribière Form (A.11):

$$\beta \leftarrow \frac{\mathbf{g}^T (\mathbf{g} - \mathbf{g}_{old})}{\mathbf{g}_{old}^T \mathbf{g}_{old}}$$

Berechne Suchrichtung für die nächste Iteration (A.10):

$$\mathbf{d}_{old} \leftarrow \mathbf{d}$$

$$\mathbf{d} \leftarrow -\mathbf{g} + \beta \mathbf{d}_{old}$$

prüfe ein Konvergenzkriterium:

$$\text{converged} \leftarrow (\mathbf{g} \approx \mathbf{0})$$

end while

B. HERLEITUNGEN

B.1 Impulsantwort eines Zweipol-Filters mit einer Nullstelle

Wir betrachten eine Transferfunktion zweiter Ordnung der Art:

$$H(z) = g \underbrace{\frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}}_{:=H'(z)} \quad (\text{B.1})$$

so wie sie von den Teilfiltern in einer parallelen Struktur realisiert wird. Da dies später benötigt wird, wurde mit $H'(z)$ ein Bezeichner für die Transferfunktion ohne Gain-Faktor g eingeführt. Ziel ist es, eine geschlossene Formel für die Impulsantwort eines solchen Filters anzugeben. Dazu zerlegen wir den Bruch in der Transferfunktion zunächst in zwei Partialbrüche, so dass:

$$H(z) = g \left(\frac{r_1}{1 - p_1 z^{-1}} + \frac{r_2}{1 - p_2 z^{-1}} \right)$$

Wir wollen hier davon ausgehen, dass der Nenner von $H'(z)$ konjugiert komplexe Nullstellen hat. In diesem Fall nimmt die Partialbruchzerlegung folgende Form an (siehe [18]):

$$H(z) = g \left(\frac{r}{1 - p z^{-1}} + \frac{r^*}{1 - p^* z^{-1}} \right)$$

Aus einer Tabelle für z -Transformationen (z.B. aus [10]) entnehmen wir das Transformationspaar:

$$a^n u[n] \longleftrightarrow \frac{1}{1 - a z^{-1}}$$

wobei die Folge $a^n u[n]$ die inverse z -Transformierte der Transferfunktion auf der rechten Seite darstellt und damit die zu dieser Transferfunktion gehörige Impulsantwort. Mit $u[n]$ soll hier die Einheitssprung-Funktion gemeint sein, die definiert ist als:

$$u[n] = \begin{cases} 1 & \text{für } n \geq 0 \\ 0 & \text{für } n < 0 \end{cases}$$

Da wir wissen, dass wir es mit kausalen Filtern zu tun haben und somit generell $h[n] = 0$ für $n < 0$ gilt, wollen wir auf den Faktor $u[n]$ zur Vereinfachung der Notation verzichten. Es soll uns also folgendes Transformationspaar reichen:

$$a^n \longleftrightarrow \frac{1}{1 - a z^{-1}} \quad n \geq 0$$

Aufgrund der Linearität der z -Transformation ergibt eine Addition von zwei Partialbrüchen dieser Form im Zeitbereich eine Addition der beiden zugehörigen Impulsantworten. Des Weiteren bewirken die Faktoren r bzw. r^* im Zähler jeweils eine Multiplikation der Impulsantworten mit dem gleichen Faktor. Der globale Vorfaktor g resultiert ebenso in einem Vorfaktor g in der Impulsantwort. Die inverse z -Transformierte und damit die gesuchte Impulsantwort unserer Transferfunktion lautet somit:

$$h[n] = g(rp^n + r^*(p^*)^n)$$

Sei $r = Re^{j\varphi}$ und $p = Pe^{j\omega}$, dann ist:

$$\begin{aligned} h[n] &= g(Re^{j\varphi}(Pe^{j\omega})^n + Re^{-j\varphi}(Pe^{-j\omega})^n) \\ &= gR \cdot P^n \cdot (e^{j\varphi}(e^{j\omega})^n + e^{-j\varphi}(e^{-j\omega})^n) \\ &= gR \cdot P^n \cdot \underbrace{(e^{j(\varphi+\omega n)} + e^{-j(\varphi+\omega n)})}_{2 \cos(\varphi+\omega n)} \end{aligned}$$

was uns zu der endgültigen Form der Impulsantwort führt:

$$\boxed{h[n] = 2gR \cdot P^n \cdot \cos(\omega n + \varphi)} \quad (\text{B.2})$$

Dabei hängen der Betrag P und Winkel ω des Polpaares (p, p^*) mit den Koeffizienten a_1, a_2 im Nenner über Gleichung 2.5 zusammen, die hier nach den Parametern P, ω aufgelöst wird:

$$\begin{aligned} P &= \sqrt{a_2} \\ \omega &= \arccos\left(-\frac{a_1}{2\sqrt{a_2}}\right) \end{aligned}$$

Der Betrag R und Winkel φ des Zählerpaares (r, r^*) bzw. die komplexe Zahl r selbst kann durch Auswerten der Gleichung:

$$r = (1 - pz^{-1})H'(z)\Big|_{z=p}$$

gefunden werden, wobei $H'(z)$, wie oben eingeführt, die Transferfunktion ohne den Gain-Faktor g darstellt. Dabei ist anzumerken, dass der Ausdruck $(1 - pz^{-1})$ an der Stelle $z = p$ zu Null wird. Dass der Zähler r dennoch nicht identisch Null ist, liegt daran, dass in $H'(z)$ an der Stelle $z = p$ eine Polstelle auftritt, denn es ergibt sich durch Faktorisierung des Nenners:

$$\begin{aligned} r &= (1 - pz^{-1})H'(z)\Big|_{z=p} \\ &= (1 - pz^{-1})\frac{1 + b_1z^{-1}}{1 + a_1z^{-1} + a_2z^{-2}}\Big|_{z=p} \\ &= (1 - pz^{-1})\frac{1 + b_1z^{-1}}{(1 - pz^{-1})(1 - p^*z^{-1})}\Big|_{z=p} \\ &= \frac{1 + b_1z^{-1}}{1 - p^*z^{-1}}\Big|_{z=p} \end{aligned}$$

Nun wird $z = p$ eingesetzt:

$$r = \frac{1 + b_1 p^{-1}}{1 - p^* p^{-1}}$$

mit $p^{-1} = (P e^{j\omega})^{-1} = P^{-1} e^{-j\omega}$ und $p^* = (P e^{j\omega})^* = P e^{-j\omega}$:

$$\begin{aligned} r &= \frac{1 + b_1 P^{-1} e^{-j\omega}}{1 - P e^{-j\omega} P^{-1} e^{-j\omega}} \\ &= \frac{1 + (b_1/P) e^{-j\omega}}{1 - e^{-j\omega} e^{-j\omega}} \\ &= \frac{1 + (b_1/P) e^{-j\omega}}{1 - e^{-2j\omega}} \end{aligned}$$

Durch Erweitern mit $1 - e^{2j\omega}$ und einige weitere Umformungen gelangt man zu einer Form in der Real- und Imaginärteil voneinander getrennt sind:

$$\boxed{r = \frac{1}{2} - j \frac{\sin(2\omega) + 2(b_1/P) \sin(\omega)}{2(1 - \cos(2\omega))}} \quad (\text{B.3})$$

Betrag und Phase können nun wie gewohnt extrahiert werden.

B.2 Betragsfrequenzgang einer Biquad-Reihenschaltung

In diesem Abschnitt soll ein allgemeiner Ausdruck für den (quadrierten) Betragsfrequenzgang einer einzelnen Biquad-Filterstufe und einer Reihenschaltung von S solcher Stufen hergeleitet werden, da dies für spätere Betrachtungen benötigt wird. Zunächst betrachten wir das Betragsquadrat $|H(e^{j\Omega})|^2$ des Biquads. Hierfür gilt:

$$\begin{aligned} |H(e^{j\Omega})|^2 &= H(e^{j\Omega}) H^*(e^{j\Omega}) \\ &= H(e^{j\Omega}) H(e^{-j\Omega}) \\ &= \frac{b_0 + b_1 e^{-j\Omega} + b_2 e^{-2j\Omega}}{a_0 + a_1 e^{-j\Omega} + a_2 e^{-2j\Omega}} \cdot \frac{b_0 + b_1 e^{j\Omega} + b_2 e^{2j\Omega}}{a_0 + a_1 e^{j\Omega} + a_2 e^{2j\Omega}} \\ &=: \frac{Num}{Den} \end{aligned}$$

wobei die Abkürzungen *Num* für den Zähler (englisch: Numerator) und *Den* für den Nenner (englisch: Denominator) eingeführt wurden mit dem Hintergedanken, das ganze Ausmultiplizieren nur für den Zähler *Num* durchzuführen - für den Nenner *Den* sieht die

Rechnung dann ganz genauso aus. Es gilt also:

$$\begin{aligned}
Num &= (b_0 + b_1 e^{-j\Omega} + b_2 e^{-2j\Omega}) \cdot (b_0 + b_1 e^{j\Omega} + b_2 e^{2j\Omega}) \\
&= b_0^2 + b_0 b_1 e^{j\Omega} + b_0 b_2 e^{2j\Omega} \\
&\quad + b_1 e^{-j\Omega} b_0 + b_1 e^{-j\Omega} b_1 e^{j\Omega} + b_1 e^{-j\Omega} b_2 e^{2j\Omega} \\
&\quad + b_2 e^{-2j\Omega} b_0 + b_2 e^{-2j\Omega} b_1 e^{j\Omega} + b_2 e^{-2j\Omega} b_2 e^{2j\Omega} \\
&= b_0^2 + b_0 b_1 e^{j\Omega} + b_0 b_2 e^{2j\Omega} \\
&\quad + b_0 b_1 e^{-j\Omega} + b_1^2 \underbrace{e^{-j\Omega} e^{j\Omega}}_{e^{j\Omega-j\Omega}=1} + b_1 b_2 \underbrace{e^{-j\Omega} e^{2j\Omega}}_{e^{j\Omega}} \\
&\quad + b_0 b_2 e^{-2j\Omega} + b_1 b_2 \underbrace{e^{-2j\Omega} e^{j\Omega}}_{e^{-j\Omega}} + b_2^2 \underbrace{e^{-2j\Omega} e^{2j\Omega}}_1
\end{aligned}$$

Vereinfachen und Zusammenfassen von Termen mit gleichen Koeffizienten:

$$\begin{aligned}
Num &= b_0^2 + b_0 b_1 e^{j\Omega} + b_0 b_2 e^{2j\Omega} + b_0 b_1 e^{-j\Omega} + b_1^2 + b_1 b_2 e^{j\Omega} + b_0 b_2 e^{-2j\Omega} + b_1 b_2 e^{-j\Omega} + b_2^2 \\
&= b_0^2 + b_1^2 + b_2^2 + b_0 b_1 e^{j\Omega} + b_0 b_1 e^{-j\Omega} + b_0 b_2 e^{2j\Omega} + b_0 b_2 e^{-2j\Omega} + b_1 b_2 e^{j\Omega} + b_1 b_2 e^{-j\Omega}
\end{aligned}$$

Ausklammern der Koeffizienten wenn möglich und Anwendung von $\cos(\varphi) = \frac{e^{j\varphi} + e^{-j\varphi}}{2}$:

$$\begin{aligned}
Num &= b_0^2 + b_1^2 + b_2^2 + b_0 b_1 \underbrace{(e^{j\Omega} + e^{-j\Omega})}_{2 \cos(\Omega)} + b_0 b_2 \underbrace{(e^{2j\Omega} + e^{-2j\Omega})}_{2 \cos(2\Omega)} + b_1 b_2 \underbrace{(e^{j\Omega} + e^{-j\Omega})}_{2 \cos(\Omega)} \\
&= b_0^2 + b_1^2 + b_2^2 + 2 \cos(\Omega) b_0 b_1 + 2 \cos(2\Omega) b_0 b_2 + 2 \cos(\Omega) b_1 b_2 \\
&= b_0^2 + b_1^2 + b_2^2 + 2 \cos(\Omega) (b_0 b_1 + b_1 b_2) + 2 \cos(2\Omega) b_0 b_2
\end{aligned}$$

Für den Nenner *Den* sieht die Rechnung, wie gesagt, genauso aus. Es ergibt sich also:

$$Den = a_0^2 + a_1^2 + a_2^2 + 2 \cos(\Omega) (a_0 a_1 + a_1 a_2) + 2 \cos(2\Omega) a_0 a_2$$

Für unseren quadrierten Betragsfrequenzgang ergibt sich damit also insgesamt:

$$|H(e^{j\Omega})|^2 = \frac{b_0^2 + b_1^2 + b_2^2 + 2 \cos(\Omega) (b_0 b_1 + b_1 b_2) + 2 \cos(2\Omega) b_0 b_2}{a_0^2 + a_1^2 + a_2^2 + 2 \cos(\Omega) (a_0 a_1 + a_1 a_2) + 2 \cos(2\Omega) a_0 a_2} \quad (\text{B.4})$$

und somit für den Betragsfrequenzgang:

$$|H(e^{j\Omega})| = \sqrt{\frac{b_0^2 + b_1^2 + b_2^2 + 2 \cos(\Omega) (b_0 b_1 + b_1 b_2) + 2 \cos(2\Omega) b_0 b_2}{a_0^2 + a_1^2 + a_2^2 + 2 \cos(\Omega) (a_0 a_1 + a_1 a_2) + 2 \cos(2\Omega) a_0 a_2}} \quad (\text{B.5})$$

Für *S* in Reihe geschaltete Biquads ergibt sich der Betragsfrequenzgang als Produkt über Betragsfrequenzgänge der einzelnen Biquad-Stufen:

$$|H(e^{j\Omega})| = \prod_{s=1}^S |H_s(e^{j\Omega})|$$

bzw. mit zusätzlichem globalen Gain-Faktor *G*:

$$|H(e^{j\Omega})| = G \prod_{s=1}^S |H_s(e^{j\Omega})| \quad (\text{B.6})$$

und schließlich als quadrierter Betragsfrequenzgang einer solchen Reihenschaltung inklusive Gain:

$$\boxed{|H(e^{j\Omega})|^2 = G^2 \prod_{s=1}^S |H_s(e^{j\Omega})|^2} \quad (\text{B.7})$$

B.3 Ableitungen der Kostenfunktion nach den Biquad-Koeffizienten

In diesem Anhang sollen Ausdrücke für die Ableitung einer Kostenfunktion der Art:

$$E = \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2$$

bezüglich der Filterkoeffizienten hergeleitet werden. Dabei sollen $|X(e^{j\Omega_k})|^2$ ein beliebiges, vorgegebenes Betragsquadratsspektrum, $|H(e^{j\Omega_k})|^2$ der (Betragsquadrats)Frequenzgang einer Reihenschaltung von S Biquad-Stufen mit globalem Gain Faktor G , und $w(\Omega_k)$ eine frequenzabhängige Gewichtung des Fehlers sein. Die Filterkoeffizienten und G stecken also alle in $|H(e^{j\Omega_k})|^2$. Die Minimierung einer solchen Kostenfunktion bedeutet eine optimale Anpassung des logarithmierten Frequenzgangs der Biquad-Kette $|H(e^{j\Omega_k})|^2$ an den vorgegebenen logarithmierten Frequenzgang $|X(e^{j\Omega_k})|^2$ im Sinne der kleinsten Fehlerquadrate.

Betrachten wir zunächst die Ableitung dieser Kostenfunktion nach dem Koeffizienten b_0 der Filterstufe i , den wir mit b_{0i} bezeichnen wollen. Wir berechnen also:

$$\frac{\partial E}{\partial b_{0i}} = \frac{\partial}{\partial b_{0i}} \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2$$

Aufgrund der Linearität der Ableitung, kann die Ableitung in die Summe hineingezogen werden, außerdem ist die Gewichtungsfunktion $w(\Omega_k)$ unabhängig von b_{0i} und kann deswegen als konstanter Faktor vor die Ableitung gezogen werden:

$$\frac{\partial E}{\partial b_{0i}} = \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) \frac{\partial}{\partial b_{0i}} (\ln |X(e^{j\Omega_k})|^2 - \ln |H(e^{j\Omega_k})|^2)^2$$

Zur Vereinfachung der Schreibweise werden im weiteren Verlauf an einigen Stellen die Amplitudenfrequenzgänge von Eingangssignal und Equalizerkette anstatt mit $|X(e^{j\Omega_k})|$ bzw. $|H(e^{j\Omega_k})|$ abkürzend mit $|X|$ bzw. $|H|$ bezeichnet - wir wissen schließlich dass es sich hier um frequenzabhängige Größen handelt. Zur Ableitung des quadratischen Terms wird zuerst die Kettenregel angewandt:

$$\frac{\partial E}{\partial b_{0i}} = \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot 2(\ln |X|^2 - \ln |H|^2) \cdot \frac{\partial}{\partial b_{0i}} (\ln |X|^2 - \ln |H|^2)$$

Da unser vorgegebenes Spektrum $|X|$ nicht von b_{0i} (oder irgendwelchen anderen Parametern, die wir kontrollieren können) abhängt, wird die entsprechende Ableitung Null:

$$\frac{\partial E}{\partial b_{0i}} = \frac{2}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot 2(\ln |X|^2 - \ln |H|^2) \cdot \frac{\partial}{\partial b_{0i}} (-\ln |H|^2)$$

Durch Vertauschung der Terme in der ersten Klammer eliminieren wir das Minuszeichen vor $\ln |H|^2$, außerdem wird der konstante Faktor 2 aus der Summe herausgezogen:

$$\boxed{\frac{\partial E}{\partial b_{0i}} = \frac{4}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot (\ln |H|^2 - \ln |X|^2) \cdot \frac{\partial}{\partial b_{0i}} (\ln |H|^2)} \quad (\text{B.8})$$

Nun müssen wir also den logarithmierten Betragsquadratsfrequenzgang unserer Equalizerkette $\ln |H|$ nach b_{0i} ableiten. Dieser Betragsfrequenzgang ist gegeben mit Gleichung B.7, somit gilt:

$$\frac{\partial}{\partial b_{0i}} \ln |H|^2 = \frac{\partial}{\partial b_{0i}} \ln \left(G^2 \prod_{s=1}^S |H_s|^2 \right)$$

Eine angenehme Eigenschaft der Logarithmusfunktion ist es, dass sie Produkte in Summen verwandelt, denn es gilt: $\ln(a \cdot b) = \ln(a) + \ln(b)$, oder allgemeiner: $\ln \prod(\cdot) = \sum \ln(\cdot)$, wobei der Punkt in den Klammern als Platzhalter für irgendwelche Terme verstanden werden soll. Damit haben wir:

$$\frac{\partial}{\partial b_{0i}} \ln |H|^2 = \frac{\partial}{\partial b_{0i}} (\ln(G^2) + \sum_{s=1}^S \ln |H_s|^2) \quad (\text{B.9})$$

wobei G ja konstant bezüglich b_{0i} ist. Daher wird die entsprechende Ableitung zu Null, des weiteren können wir die Ableitung in die Summe hineinziehen:

$$\frac{\partial}{\partial b_{0i}} \ln |H|^2 = \sum_{s=1}^S \frac{\partial}{\partial b_{0i}} \ln |H_s|^2$$

An dieser Stelle ist anzumerken, dass die Ableitung von $\ln |H_s|^2$ nach b_{0i} nur für diejenige Biquad-Stufe p mit dem Index i einen Wert ungleich Null ergibt - alle anderen Übertragungsfunktionen hängen ja nicht von b_{0i} ab und sind somit als Konstanten bezüglich b_{0i} anzusehen - die entsprechenden Ableitungen werden also zu Null. Daher bleibt von der Summe nur der Term für $s = i$ übrig:

$$\frac{\partial}{\partial b_{0i}} \ln |H|^2 = \frac{\partial}{\partial b_{0i}} \ln |H_i|^2 \quad (\text{B.10})$$

Hier ist etwas Bemerkenswertes passiert: die Ableitung des logarithmierten Betragsfrequenzgangs der Biquad-Kette nach b_{0i} ist somit einfach gleich der Ableitung des logarithmierten Betragsfrequenzgangs der Biquad-Stufe mit dem Index i . Bezüglich der weiteren Herleitung entkoppeln die einzelnen Biquad-Stufen demnach. Die Kettenregel liefert:

$$\frac{\partial}{\partial b_{0i}} \ln |H_i|^2 = \frac{1}{|H_i|^2} \frac{\partial}{\partial b_{0i}} |H_i|^2 \quad (\text{B.11})$$

Ein Ausdruck für den Betragsquadratsfrequenzgang einer einzelnen Biquad-Stufe wurde zuvor mit Gleichung B.4 bestimmt. Wir setzen dieses Ergebnis nun also ein:

$$\frac{\partial}{\partial b_{0i}} |H_i|^2 = \frac{\partial}{\partial b_{0i}} \frac{b_{0i}^2 + b_{1i}^2 + b_{2i}^2 + 2 \cos(\Omega)(b_{0i}b_{1i} + b_{1i}b_{2i}) + 2 \cos(2\Omega)b_{0i}b_{2i}}{a_{0i}^2 + a_{1i}^2 + a_{2i}^2 + 2 \cos(\Omega)(a_{0i}a_{1i} + a_{1i}a_{2i}) + 2 \cos(2\Omega)a_{0i}a_{2i}}$$

die Ableitung ergibt:

$$\frac{\partial}{\partial b_{0i}} |H_i|^2 = \frac{2b_{0i} + 2 \cos(\Omega)b_{1i} + 2 \cos(2\Omega)b_{2i}}{Den_i}$$

wobei mit $Den_i := a_{0i}^2 + a_{1i}^2 + a_{2i}^2 + 2 \cos(\Omega)(a_{0i}a_{1i} + a_{1i}a_{2i}) + 2 \cos(2\Omega)a_{0i}a_{2i}$ wie auch schon in Anhang B.2 der Nenner des Bruchs gemeint sein soll. Hätten wir von Anfang an nicht nach b_{0i} sondern nach einem anderen Koeffizienten abgeleitet, wäre bei der Herleitung im Prinzip alles gleich gewesen, nur hier im letzten Schritt (bei der innersten Ableitung) ergeben sich die Unterschiede. In Gl. B.12 sind die Ableitungen von $|H_i|^2$ nach allen 3 b -Koeffizienten angegeben.

$$\boxed{\begin{aligned} \frac{\partial}{\partial b_{0i}} |H_i|^2 &= \frac{2b_{0i} + 2 \cos(\Omega)b_{1i} + 2 \cos(2\Omega)b_{2i}}{Den_i} \\ \frac{\partial}{\partial b_{1i}} |H_i|^2 &= \frac{2b_{1i} + 2 \cos(\Omega)(b_{0i} + b_{2i})}{Den_i} \\ \frac{\partial}{\partial b_{2i}} |H_i|^2 &= \frac{2b_{2i} + 2 \cos(\Omega)b_{1i} + 2 \cos(2\Omega)b_{0i}}{Den_i} \end{aligned}} \quad (B.12)$$

Für die Ableitungen nach den a -Koeffizienten, stellen wir fest, dass diese nur im Nenner vorkommen - der Zähler ($=: Num_i$) ist also konstant bezüglich der a -Koeffizienten. Wir benötigen daher ein letztes Mal die Kettenregel und den Zusammenhang $\frac{d}{dx} \frac{c}{x} = -\frac{c}{x^2}$, $c = const.$ um folgende Ableitungen zu erhalten:

$$\boxed{\begin{aligned} \frac{\partial}{\partial a_{0i}} |H_i|^2 &= -\frac{Num_i}{Den_i^2} (2a_{0i} + 2 \cos(\Omega)a_{1i} + 2 \cos(2\Omega)a_{2i}) \\ \frac{\partial}{\partial a_{1i}} |H_i|^2 &= -\frac{Num_i}{Den_i^2} (2a_{1i} + 2 \cos(\Omega)(a_{0i} + a_{2i})) \\ \frac{\partial}{\partial a_{2i}} |H_i|^2 &= -\frac{Num_i}{Den_i^2} (2a_{2i} + 2 \cos(\Omega)a_{1i} + 2 \cos(2\Omega)a_{0i}) \end{aligned}} \quad (B.13)$$

Durch entsprechende Rücksubstitution dieser Ergebnisse in (B.8) können wir nun also auch die Ableitungen der Kostenfunktion selbst nach den entsprechenden Koeffizienten berechnen. Für b_{0i} ist dies im Folgenden angegeben, wobei der Ausgangspunkt Gl. B.8 ist, für die anderen Koeffizienten ist die Vorgehensweise analog.

$$\begin{aligned} \frac{\partial E}{\partial b_{0i}} &= \frac{4}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot (\ln |H|^2 - \ln |X|^2) \cdot \frac{\partial}{\partial b_{0i}} (\ln |H|^2) \\ &= \frac{4}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot (\ln |H|^2 - \ln |X|^2) \cdot \frac{1}{|H_i|^2} \frac{\partial}{\partial b_{0i}} |H_i|^2 \\ &= \frac{4}{K} \sum_{k=0}^{K/2} w(\Omega_k) \cdot (\ln |H|^2 - \ln |X|^2) \cdot \frac{1}{|H_i|^2} \frac{2b_{0i} + 2 \cos(\Omega)b_{1i} + 2 \cos(2\Omega)b_{2i}}{Den_i} \end{aligned}$$

mit

$$Den_i = a_{0i}^2 + a_{1i}^2 + a_{2i}^2 + 2 \cos(\Omega)(a_{0i}a_{1i} + a_{1i}a_{2i}) + 2 \cos(2\Omega)a_{0i}a_{2i}$$

B.4 Ableitungen der Biquad-Koeffizienten nach den Equalizer-Parametern

Wir werden in B.5 die Ableitungen der Biquad-Koeffizienten nach den Equalizer-Parametern benötigen. Daher sollen hier Ausdrücke dafür angegeben werden (der Begriff 'herleiten' wäre hier wohl etwas übertrieben). Aus den Filterdesign-Gleichungen (Gleichung 2.19) für die Koeffizienten ergibt sich für den Verstärkungsparameter g :

$$\begin{aligned}\frac{\partial}{\partial g} b_0 &= \frac{\partial}{\partial g} \frac{1 + \gamma\sqrt{g}}{1 + \gamma/\sqrt{g}} \\ &= \frac{(\frac{\partial}{\partial g}(1 + \gamma g^{\frac{1}{2}}))(1 + \gamma g^{-\frac{1}{2}}) - (\frac{\partial}{\partial g}(1 + \gamma g^{-\frac{1}{2}}))(1 + \gamma g^{\frac{1}{2}})}{(1 + \gamma/\sqrt{g})^2} \\ &= \frac{(\frac{1}{2}\gamma g^{-\frac{1}{2}})(1 + \gamma g^{-\frac{1}{2}}) - (-\frac{1}{2}\gamma g^{-\frac{3}{2}})(1 + \gamma g^{\frac{1}{2}})}{(1 + \gamma/\sqrt{g})^2}\end{aligned}$$

Der Zähler wird ausmultipliziert und noch etwas vereinfacht:

$$\begin{aligned}\frac{\partial}{\partial g} b_0 &= \frac{\frac{1}{2}\gamma g^{-\frac{1}{2}} + \frac{1}{2}\gamma g^{-\frac{1}{2}}\gamma g^{-\frac{1}{2}} + \frac{1}{2}\gamma g^{-\frac{3}{2}} + \frac{1}{2}\gamma g^{-\frac{3}{2}}\gamma g^{\frac{1}{2}}}{(1 + \gamma/\sqrt{g})^2} \\ &= \frac{\frac{1}{2}\gamma g^{-\frac{1}{2}} + \frac{1}{2}\gamma^2 g^{-1} + \frac{1}{2}\gamma g^{-\frac{3}{2}} + \frac{1}{2}\gamma^2 g^{-1}}{(1 + \gamma/\sqrt{g})^2}\end{aligned}$$

weitere Vereinfachungen führen zu der Form, wie sie in B.14 zusammen mit den Ableitungen der anderen Koeffizienten nach dem Parameter g angegeben ist.

$$\boxed{\begin{aligned}\frac{\partial b_0}{\partial g} &= \frac{\frac{1}{2}\frac{\gamma}{\sqrt{g}} + \frac{\gamma^2}{g} + \frac{1}{2}\frac{\gamma}{\sqrt{g^3}}}{(1 + \gamma/\sqrt{g})^2}; & \frac{\partial b_1}{\partial g} &= \frac{-\frac{\gamma}{\sqrt{g^3}} \cos(\Omega_c)}{(1 + \gamma/\sqrt{g})^2}; & \frac{\partial b_2}{\partial g} &= \frac{-\frac{1}{2}\frac{\gamma}{\sqrt{g}} - \frac{\gamma^2}{g} + \frac{1}{2}\frac{\gamma}{\sqrt{g^3}}}{(1 + \gamma/\sqrt{g})^2} \\ \frac{\partial a_0}{\partial g} &= \frac{\partial}{\partial g} 1 = 0; & \frac{\partial a_1}{\partial g} &= \frac{\partial b_1}{\partial g}; & \frac{\partial a_2}{\partial g} &= \frac{\frac{\gamma}{\sqrt{g^3}}}{(1 + \gamma/\sqrt{g})^2}\end{aligned}} \quad (\text{B.14})$$

Für die partiellen Ableitungen der Filterkoeffizienten nach dem Mittenfrequenzparameter Ω_c ergibt sich:

$$\boxed{\begin{aligned}\frac{\partial b_0}{\partial \Omega_c} &= 0; & \frac{\partial b_1}{\partial \Omega_c} &= \frac{2 \sin(\Omega_c)}{1 + \gamma/\sqrt{g}}; & \frac{\partial b_2}{\partial \Omega_c} &= 0 \\ \frac{\partial a_0}{\partial \Omega_c} &= 0; & \frac{\partial a_1}{\partial \Omega_c} &= \frac{\partial b_1}{\partial \Omega_c}; & \frac{\partial a_2}{\partial \Omega_c} &= 0\end{aligned}} \quad (\text{B.15})$$

und schließlich für die partiellen Ableitungen der Filterkoeffizienten nach dem Bandbreitenparameter γ :

$$\boxed{\begin{aligned}\frac{\partial b_0}{\partial \gamma} &= \frac{\sqrt{g} - \frac{1}{\sqrt{g}}}{(1 + \gamma/\sqrt{g})^2}; & \frac{\partial b_1}{\partial \gamma} &= \frac{\frac{2}{\sqrt{g}} \cos(\Omega_c)}{(1 + \gamma/\sqrt{g})^2}; & \frac{\partial b_2}{\partial \gamma} &= \frac{-\sqrt{g} - \frac{1}{\sqrt{g}}}{(1 + \gamma/\sqrt{g})^2} \\ \frac{\partial a_0}{\partial \gamma} &= 0; & \frac{\partial a_1}{\partial \gamma} &= \frac{\partial b_1}{\partial \gamma}; & \frac{\partial a_2}{\partial \gamma} &= \frac{-\frac{2}{\sqrt{g}}}{(1 + \gamma/\sqrt{g})^2}\end{aligned}} \quad (\text{B.16})$$

B.5 Ableitungen der Kostenfunktion nach den Equalizer-Parametern

Wenn wir sowohl die Ableitungen der Kostenfunktion nach den Biquad-Koeffizienten als auch die Ableitungen der Biquad-Koeffizienten nach den Equalizer-Parametern kennen, können wir daraus nach der verallgemeinerten Kettenregel die Ableitungen der Kostenfunktion nach den Equalizer-Parametern berechnen. Die verallgemeinerte Kettenregel lautet:

sei

$$y = f(x_1, x_2, \dots, x_n)$$

eine Funktion der Variablen x_1, \dots, x_n wobei

$$\begin{aligned} x_1 &= g_1(t_1, t_2, \dots, t_m) \\ x_2 &= g_2(t_1, t_2, \dots, t_m) \\ &\vdots \\ x_n &= g_n(t_1, t_2, \dots, t_m) \end{aligned}$$

Funktionen von t_1, \dots, t_m sind. Dann ist die Ableitung von f bezüglich einem der t_i gegeben mit:

$$\frac{\partial f}{\partial t_i} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t_i} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t_i} + \dots + \frac{\partial f}{\partial x_n} \frac{\partial x_n}{\partial t_i}$$

In unserem Falle entspricht f der Kostenfunktion E , die x_i entsprechen den Biquad-Koeffizienten $b_{0i}, b_{1i}, b_{2i}, a_{0i}, a_{1i}, a_{2i}$ und die t_i entsprechen den Equalizer-Parametern $g_i, \Omega_{ci}, \gamma_i$. Es gilt demzufolge:

$$\boxed{\begin{aligned} \frac{\partial E}{\partial g_i} &= \frac{\partial E}{\partial b_{0i}} \frac{\partial b_{0i}}{\partial g_i} + \frac{\partial E}{\partial b_{1i}} \frac{\partial b_{1i}}{\partial g_i} + \frac{\partial E}{\partial b_{2i}} \frac{\partial b_{2i}}{\partial g_i} + \frac{\partial E}{\partial a_{0i}} \frac{\partial a_{0i}}{\partial g_i} + \frac{\partial E}{\partial a_{1i}} \frac{\partial a_{1i}}{\partial g_i} + \frac{\partial E}{\partial a_{2i}} \frac{\partial a_{2i}}{\partial g_i} \\ \frac{\partial E}{\partial \Omega_{ci}} &= \frac{\partial E}{\partial b_{0i}} \frac{\partial b_{0i}}{\partial \Omega_{ci}} + \frac{\partial E}{\partial b_{1i}} \frac{\partial b_{1i}}{\partial \Omega_{ci}} + \frac{\partial E}{\partial b_{2i}} \frac{\partial b_{2i}}{\partial \Omega_{ci}} + \frac{\partial E}{\partial a_{0i}} \frac{\partial a_{0i}}{\partial \Omega_{ci}} + \frac{\partial E}{\partial a_{1i}} \frac{\partial a_{1i}}{\partial \Omega_{ci}} + \frac{\partial E}{\partial a_{2i}} \frac{\partial a_{2i}}{\partial \Omega_{ci}} \\ \frac{\partial E}{\partial \gamma_i} &= \frac{\partial E}{\partial b_{0i}} \frac{\partial b_{0i}}{\partial \gamma_i} + \frac{\partial E}{\partial b_{1i}} \frac{\partial b_{1i}}{\partial \gamma_i} + \frac{\partial E}{\partial b_{2i}} \frac{\partial b_{2i}}{\partial \gamma_i} + \frac{\partial E}{\partial a_{0i}} \frac{\partial a_{0i}}{\partial \gamma_i} + \frac{\partial E}{\partial a_{1i}} \frac{\partial a_{1i}}{\partial \gamma_i} + \frac{\partial E}{\partial a_{2i}} \frac{\partial a_{2i}}{\partial \gamma_i} \end{aligned}} \quad (\text{B.17})$$

wobei jedoch jeweils der 4. Summand verschwindet, da ja $a_0 = \text{const.} = 1$ ist. Der Vollständigkeit halber, wurde der Term jedoch mitgeschrieben.

Jetzt fehlt uns nur noch die Ableitung der Kostenfunktion nach dem globalen Gain-Faktor G . Die ersten Herleitungsschritte dafür sind identisch mit denen zur Berechnung der Ableitungen nach den Biquad-Koeffizienten bis zu Gleichung (B.9), die hier nochmal in entsprechend angepasster Form angegeben ist:

$$\frac{\partial}{\partial G} \ln |H|^2 = \frac{\partial}{\partial G} (\ln(G^2) + \sum_{s=1}^s \ln |H_s|^2)$$

An dieser Stelle wäre die Herleitung dann so gut wie fertig gewesen und es ergibt sich:

$$\frac{\partial}{\partial G} \ln |H|^2 = \frac{\partial}{\partial G} \ln(G^2) = \frac{1}{G^2} \cdot \frac{\partial}{\partial G} G^2 = \frac{1}{G^2} \cdot 2G$$

und somit:

$$\boxed{\frac{\partial}{\partial G} \ln |H|^2 = \frac{2}{G}} \quad (\text{B.18})$$

C. MATLAB-IMPLEMENTATION

Die in dieser Arbeit vorgestellten Verfahren und Algorithmen wurden natürlich zu Testzwecken auch implementiert. Zur Implementation wurde die Sprache MatLab gewählt. Gründe für diese Wahl sind die weite Verbreitung im wissenschaftlich/technischen Bereich, die einfachen und komfortablen Möglichkeiten, Ergebnisse zu visualisieren, die mathematisch orientierte und verständliche Syntax der Sprache und schließlich die direkte Verfügbarkeit von Standardalgorithmen dank der umfangreichen Funktionsbibliothek (z.B. *FFT*, Levinson/Durbin Rekursion, Faktorisierung von Polynomen, etc.). Die geringere Effizienz von MatLab-Programmen gegenüber kompilierten Sprachen wie C/C++ ist im Rahmen einer solchen Arbeit akzeptabel, da es ja in erster Linie um 'proof of concept'-Implementationen geht - wie es zu neudeutsch so schön heißt. Bei der Benutzung der *m*-Files ist zu beachten, dass einige von ihnen auch Funktionen aus der Signal-Processing Toolbox benötigen. Da es insgesamt ziemlich viele, zum Teil auch recht lange *m*-Files geworden sind, werden in diesem Anhang nur die relevanten Funktionen und Skripte aufgelistet und deren Anwendungszweck kurz erklärt. Es wird jedoch nicht der komplette Code abgedruckt. Stattdessen sind die entsprechenden *m*-Files in Form eines .zip-Archives von <http://www.kgw.tu-berlin.de/forschung/magisterarbeiten/> herunterladbar - von dieser Adresse aus, muss man sich noch etwas durchklicken. Die *m*-Files sind meist ausführlich in englischer Sprache kommentiert und deren Benutzung wird erklärt, so dass hier lediglich eine Auflistung (in alphabetischer Reihenfolge) mit kurzer Funktionsbeschreibung ausreichen soll.

`biquadBank.m` implementiert eine Parallelschaltung von Biquad-Filtern.

`calculateErrorFunction.m` berechnet den Wert der Kostenfunktion $E(\boldsymbol{\theta})$ für die Equalizer-Kette (Gl. 2.25) an einer bestimmten Stelle $\boldsymbol{\theta}$.

`calculateErrorWeighting.m` berechnet einen Vektor mit Fehlergewichten $w(\Omega_k)$ für die Kostenfunktion anhand bestimmter Vorgaben (Gl. 2.27).

`calculateGradient.m` berechnet den Gradienten (Gl. 2.26) der Kostenfunktion an einer bestimmten Stelle $\boldsymbol{\theta}$.

`equalizeSignal.m` filtert ein Signal mit einer Reihenschaltung von parametrischen Equalizern.

`estimateEqParametersAhh.m` Skript zur Bestimmung von Equalizer-Parametern für das Beispielsignal. Das Skript speichert die gefundenen Parameter am Ende in `.mat`-Dateien ab. Diese `.mat`-Dateien werden von dem Skript `trafoEqualizer.m` benötigt um die entsprechenden Transformationen durchzuführen. Für das Beispielsignal sind die entspre-

chenden `.mat`-files aber schon im Archiv enthalten, so das der Benutzer die Optimierung nicht mehr selbst durchführen muss.

`estimateFundamental.m` schätzt die Grundfrequenz eines Signals.

`filterCoeffsToFofParams.m` Umrechnung von Filter-Koeffizienten (g, b_1, a_1, a_2) eines Zweipol-Filters mit Nullstelle in FOF-Parameter $(A, \alpha, \omega, \phi)$ (Algorithmus 3).

`findInitialEqualizerParameters.m` Funktion zur Initialisierung der Equalizer-Parameter.

`fofParamsToFilterCoeffs.m` Umrechnung von FOF-Parametern $(A, \alpha, \omega, \phi)$ in Koeffizienten (g, b_1, a_1, a_2) für ein Zweipol-Filter mit Nullstelle (Algorithmus 4).

`generateEqualizerCurve.m` berechnet den Amplitudenfrequenzgang einer Equalizerkette.

`optimizeEQParametersViaSCG.m` optimiert die Parameter einer Equalizerkette mit Hilfe des Scaled Conjugate Gradient (SCG) Verfahrens.

`proofEqualizerInversion.m` benutzt die Symbolic-Math Toolbox, um zu zeigen dass die Transferfunktionen zweier Biquad-Equalizer mit reziproken Gain-Faktoren invers zueinander sind.

`spectralEnvelopeViaTE.m` schätzt die spektrale Hüllkurve eines Signals mit Hilfe des True Envelope (TE) Algorithmus.

`telpc.m` Funktion zur Schätzung von AR-Modell-Parametern nach der TE-LPC Methode.

`trafoEqualizer.m`, `trafoParallel.m`, `trafoSerial.m` sind Skripte mit denen die verschiedenen Formant-Transformationen ausprobiert werden können. Mit diesen Skripten wurden auch die Plots in den jeweiligen Praxistests erstellt.

`transformFormantsViaEqualizer.m` Transformiert die Formanten eines Signals anhand einer auf das Signal angepassten Equalizerkette (deren Parameter zuvor geschätzt wurden) und einer zweiten Equalizerkette (deren Parameter die Ziel-Formanten darstellen).

`trueEnvelopeScale.m` implementiert eine Skalierung (Stauchung/Streckung) der spektralen Hüllkurve mit Hilfe des True Envelope Algorithmus (Formant-Scaling).

`trueEnvelopeVocoder.m` implementiert einen Vocoder auf Basis des True Envelope Algorithmus.

`updateParametersViaSCG.m` führt einen einzelnen Lernschritt im Scaled Conjugate Gradient (SCG) Algorithmus aus. Sollte innerhalb einer Art `'while(not converged)'`-Schleife verwendet werden.

Des weiteren sind sämtliche `.m`-files enthalten, mit denen die Plots in den verschiedenen Kapiteln erstellt worden sind.

LITERATURVERZEICHNIS

- [1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [2] Richard Boulanger (Editor). *The CSound Book*. MIT Press, 2000.
- [3] Robert Bristow-Johnson. The equivalence of various methods of computing biquad coefficients for audio parametric equalizers. <http://www.musicdsp.org/files/EQ-Coefficients.pdf>.
- [4] Amro El-Jaroudi and John Makhoul. Discrete all-pole modeling. In *IEEE Transactions On Signal Processing*, volume 39, No.2, February 1991.
- [5] Jondarr Gibb. Back propagation family album. <http://citeseer.comp.nus.edu.sg/gibb96back.html>, 1996.
- [6] Simon Haykin. *Adaptive Filter Theory (4th Edition)*. Prentice-Hall, 2002.
- [7] John Makhoul. Linear prediction: A tutorial review. In *Proceedings of the IEEE*, volume 63, No.4, April 1975.
- [8] John Makhoul. Spectral linear prediction: Properties and applications. In *IEEE Transactions On Acoustics, Speech and Signal Processing*, volume ASSP-23, No.3, June 1975.
- [9] Martin F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. PB-339 Preprint.
- [10] Oppenheim and Schaffer. *Discrete Time Signal Processing (2nd Edition)*. Prentice Hall, 1999.
- [11] Axel Röbel. *Analysis, modeling and transformation of audio signals - lecture material*. http://recherche.ircam.fr/equipes/analyse-synthese/roebel/amt_lecture.html.
- [12] Axel Röbel and Xavier Rodet. Improving lpc spectral envelope extraction of voiced speech by true-envelope estimation. <http://mediatheque.ircam.fr/articles/textes/Villavicencio06a/>.
- [13] Axel Röbel and Xavier Rodet. Real time signal transposition with envelope preservation in the phase vocoder. http://recherche.ircam.fr/equipes/analyse-synthese/roebel/paper/trueenv_icmc2005.pdf.

-
- [14] Axel Röbel and Xavier Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Proc. of the 8th Int. Conf. on Digital Audio Effects (DAFx'05)*, pp. 30-35, 2005.
- [15] Xavier Rodet. Time-domain formant-wave-function synthesis. *Computer Music Journal*, 1984.
- [16] Xavier Rodet, Yves Potard, and Jean-Baptiste Barrière. The chant project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 1984.
- [17] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. <http://www.cs.cmu.edu/afs/cs/project/quake/public/papers/painless-conjugate-gradient.ps>.
- [18] Julius O. Smith. *Introduction to Digital Filters with Audio Applications*. <http://ccrma.stanford.edu/~jos/GlobalJOSIndex.html>.
- [19] Ken Steiglitz. *A Digital Signal Processing Primer, with Applications to Digital Audio and Computer Music*. Addison-Wesley, 1996.
- [20] Udo Zölzer (Editor). *DAFX - Digital Audio Effects*. John Wiley & Sons LTD, 2002.