

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

ELEKTROTECHNIK

VERTIEFUNGSFACH DIGITALE SIGNALVERARBEITUNG

Schlussbericht der Diplomarbeit

Mehrkanalige Raumakustik-Inversion

Andy Rohr

12. Dezember 2001

Inhaltsverzeichnis

Aufgabenstellung	iii
Abstract	v
1 Projektplanung	1
1.1 Struktur	1
1.2 Zeitaufwand	1
1.3 Projektplan	2
2 Theorie	3
2.1 Allgemeine Einführung	3
2.2 Einkanalige Inversion	4
2.2.1 Berechnung der inversen Impulsantwort	5
2.3 Mehrkanalige Inversion	5
2.3.1 Das multiple-input/output theorem (MINT)	5
2.4 Zweikanalige Inversion nach MINT	6
2.4.1 Berechnung der inversen Impulsantwort	7
2.5 Schlussfolgerungen	7
3 Entwicklung und Resultate	9
3.1 Bestimmung der Raumimpulsantworten	9
3.1.1 Messverfahren	10
3.1.2 Auswertung	10
3.2 Simulation in MATLAB	11
3.2.1 Rahmenbedingungen	11
3.2.2 Einkanalige Inversion	14
3.2.3 Zweikanalige Inversion	21
3.2.4 Vergleich und Diskussion	26
3.3 DSP-Implementation	28
3.3.1 FIR-Filter-Implementation	28
3.3.2 Vorbereitungen und Download der Filterkoeffizienten	29
3.3.3 Systemaufbau	30
3.4 Messung am Echtzeitsystem	30
3.4.1 Einkanalige Inversion	31
3.4.2 Zweikanalige Inversion	32
3.4.3 Vergleich und Diskussion	34
3.5 Resultate und Schlussfolgerungen	35
3.6 Schlusswort	36
3.6.1 Ziel erreicht?	36
3.6.2 Persönliche Bemerkungen	36

A Listings	37
A.1 MATLAB Programme	37
A.1.1 Einkanalige Inversion	37
A.1.2 Zweikanalige Inversion	40
A.2 DSP-Programm	44
B Messequipment	49
Literaturverzeichnis	51
Abbildungsverzeichnis	53

Aufgabenstellung

Aufgabe

Die theoretischen und praktischen Aspekte der Raumakustik-Entzerrungsproblematik sind zu studieren. Dabei soll sowohl der einkanalige als auch der zweikanalige Fall betrachtet werden. Im Auge zu behalten ist dabei eine Echtzeitrealisierung mit Hilfe des Labor-DSP-Systems zusammen mit einer MATLAB-Berechnung der FIR-Entzerrerkoeffizienten. Schliesslich ist das System in dieser Form zu realisieren, wobei der Raum aufgrund gemessener Übertragungsfunktionen in einem zweiten DSP zu simulieren ist. Damit der Echtzeit-Rechenaufwand im Rahmen bleibt, ist ein möglichst kleiner Raum mit kurzen Stossantworten zu verwenden (vgl. [4]). Das System soll einen Mess/Rechen-Mode zur Bestimmung der Entzerrerkoeffizienten und einen anschließenden Echtzeit-Kompensations-Mode umfassen. Die Resultate des ein- und des zweikanaligen Falles sind zu vergleichen.

Literatur

- (1) Miyoshi M. und Kaneda Y., Inverse Filtering of Room Acoustics. IEEE Trans. Acoust., Speech, Signal Processing, Vol. 36, No. 2, Febr. 1988, 145-152.
- (2) Nelson P.A., Orduna-Bustamante F. und Hamada H., Inverse Filter Design and Equalization Zones in Multichannel Sound Reproduction. IEEE Trans. Speech and Audio Processing, Vol. 3, May 1995, 185-192.
- (3) Fast Transversal Filters for Deconvolution in Multichannel Sound Reproduction. IEEE Trans. Speech and Audio Processing, Vol. 9, No. 4, May 2001, 429-440.
- (4) Bisatz G.A. und Gisi M., Inverse Raumakustik-Filterung mit DSP. Diplomarbeit Nr. D94-06 am Labor für Digitale Signalverarbeitung der HSR, 1994.

Bericht, Dokumentation

Über die Arbeit ist ein Bericht zu verfassen. Alle verwendeten Quellen sind im Literaturverzeichnis des Berichts anzugeben (Hinweise im Text). Der Bericht ist doppelter Ausführung (Papier, gebunden) abzugeben; ein Exemplar verbleibt am Labor für digitale Signalverarbeitung der HSR, das Doppel erhalten die Verfasser nach dem Diplomabschluss zurück. Die erstellten Programme und die Dateien des Berichtes werden vom Laborassistenten entgegengenommen. Ferner ist eine Zusammenfassung der Arbeit (max. eine Seite) dem Abteilungssekretariat zukommen zu lassen (als WORD- und pdf-File an sekretariat.e@hsr.ch).

Termine, Bedingungen

Gemäss Vorgaben und Terminplan des Vorstandes der Abteilung für Elektrotechnik.

Ausgabe der Aufgabenstellung, Beginn der Arbeit: Dienstag, 23.10.2001

Arbeitsplatz: DS-Labor 6.003 (zuständig: Laborassistent)

Abgabe des Berichts und der Zusammenfassung: Mittwoch, 12.12.2001, 12.00 Uhr

Erstellung Projekt-Poster: 13.12.2001

Interne Präsentation: 13./14.12.2001

Ausstellung der Arbeiten: 14.12.2001, nachmittags

Abbau der Arbeiten: 17.12. – 19.12.2001

Mündliche Prüfungen über die Arbeiten: 14.01. – 19.01.2002

Assistenz

A. Rüegg, HSR-Assistent (Arbeitsplatz: 6.003)

P. Roffler, Labor-Assistent (Arbeitsplatz: 6.005)

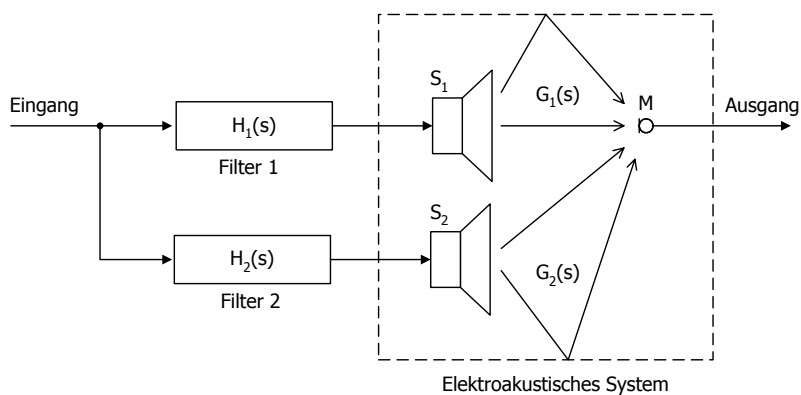
Rapperswil, 22.10.2001

Prof. Dr. A. Schüeli

Abstract

Von einem Lautsprecher abgestrahlte Audiosignale werden in einem Abhörraum, linear verzerrt durch mehrfache Reflexionen an den Wänden, der Decke, dem Fussboden an Möbeln etc. Das Schalldrucksignal am Ort eines Hörers besteht aus einer Vielzahl überlagerter Teilsignale unterschiedlicher Dämpfungen und Laufzeiten; das Signal am Hörort verzerrt. Von der Übertragungsstrecke vom Lautsprechereingang zum Ort des Hörers kann durch Verwendung eines Mikrofons am Hörort eine verzerrende Übertragungsfunktion, die Raumübertragungsfunktion, gemessen werden. Diese Verzerrung kann theoretisch dadurch egalisiert werden, dass in den elektrischen Pfad vor dem Lautsprecher ein Entzerrer eingeschaltet wird, dessen Übertragungsfunktion der inversen Raumübertragungsfunktion entspricht. Die Bestimmung und Realisierung der inversen Raumübertragungsfunktion ist jedoch in der Praxis eine nichttriviale Angelegenheit, ist doch diese Übertragungsfunktion im allgemeinen nichtminimalphasig. Es drängen sich FIR-Approximationen auf, deren Berechnung und Implementierung einen grossen Aufwand erfordern. Es ist einleuchtend, und Untersuchungen bestätigen dies, dass eine Raumakustik-Entzerrung mit Hilfe von zwei (oder mehreren) Entzerrer-Lautsprecher-Pfaden («Kanälen») unproblematischer ist und bessere Resultate zeigt.

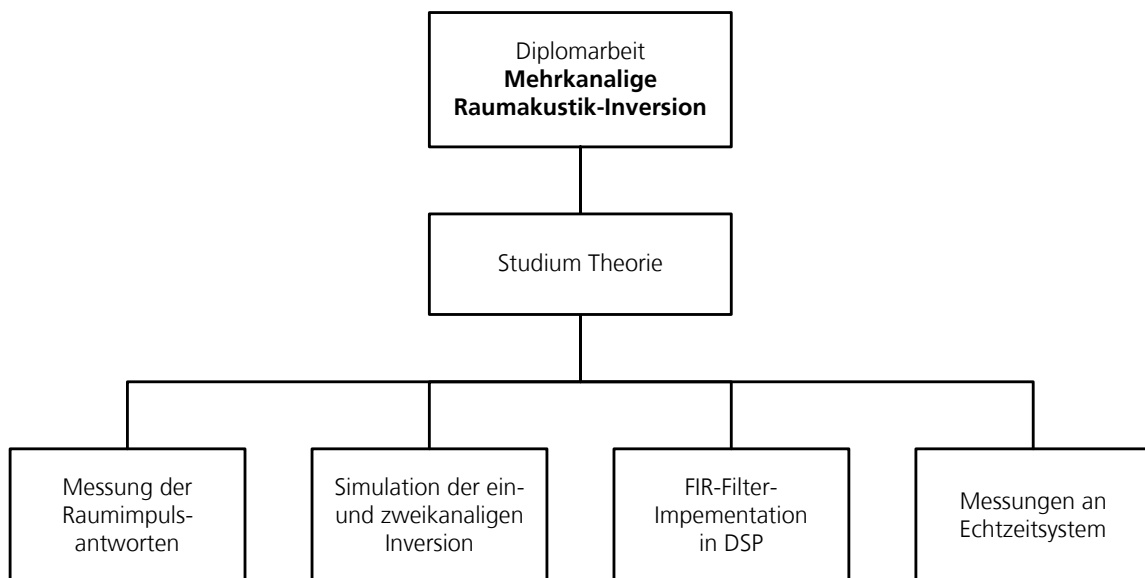
Im Rahmen dieser Diplomarbeit wurden die theoretischen und praktischen Aspekte der Raumakustik-Entzerrungsproblematik studiert und dabei sowohl der einkanalige als auch der zweikanalige Fall betrachtet und in MATLAB simuliert. Dazu mussten erst Raumimpulsantworten eines realen Testraumes gemessen werden, die Grundlage dieser Simulation bilden. Im Auge behalten wurde dabei eine Echtzeitrealisierung mit Hilfe des Labor-DSP-Systems zusammen mit einer MATLAB-Berechnung der FIR-Entzerrerkoeffizienten. Schliesslich wurde das System in dieser Form realisiert und die Wirkung der inversen Filter anhand von Messungen in diesem Testraum studiert. Die Messresultate des ein- und des zweikanaligen Falles wurden untereinander und mit der Simulation verglichen.



1 Projektplanung

1.1 Struktur

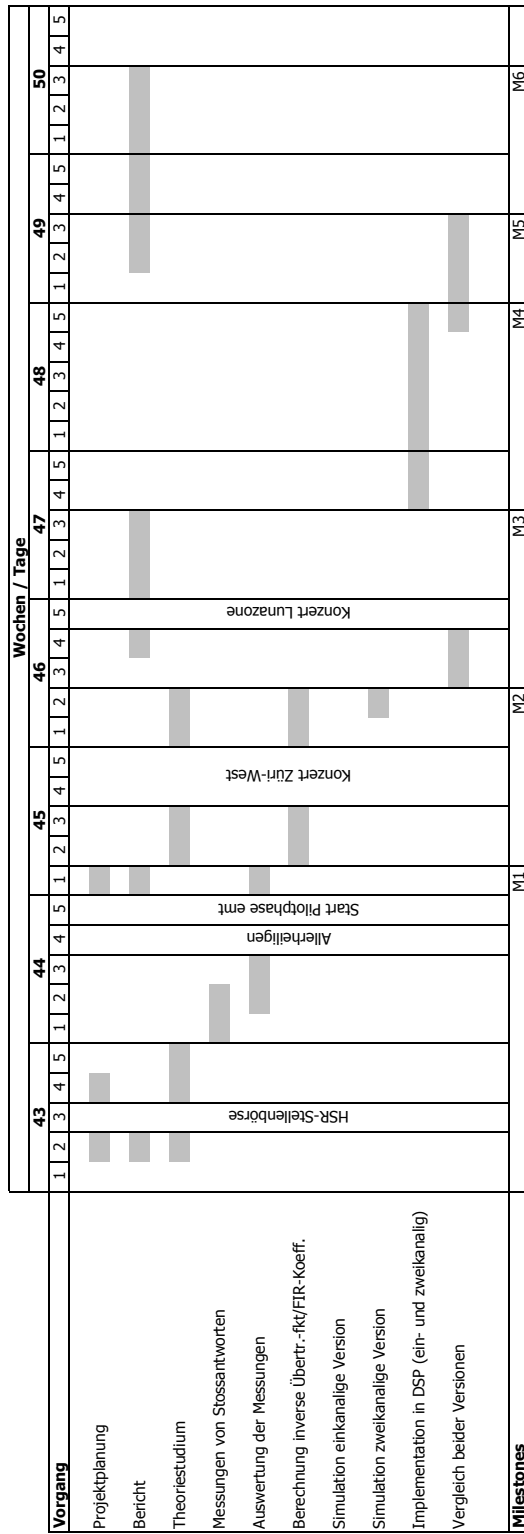
Diese Studienarbeit lässt sich in folgende Themenbereiche aufteilen. Gleichzeitig stimmt die Anordnung auch mit der Bearbeitungsreihenfolge zusammen.



1.2 Zeitaufwand

Rubrik	Zeitaufwand [h]
Theoriestudium	17.6
Messungen	18.7
Simulation MATLAB	47.1
Implementation DSP	12.7
Projektplanung	2.0
Besprechungen	3.3
Dokumentation	87.3
Overhead	6.8
Total	195.5

1.3 Projektplan



Milestones:

- M1 Stossantworten von Raum aufgenommen und ausgewertet
- M2 Simulation einkanalige und zweikanalige Simulation aufgebaut (MATLAB)
- M3 Dokumentation der Theorie und der Simulation abgeschlossen
- M4 Implementation beider Versionen in DSP abgeschlossen
- M5 Vergleich beider Versionen in Simulation und Implementation in DSP abgeschlossen
- M6 Abgabe Diplomarbeit

2 Theorie

2.1 Allgemeine Einführung

Akustische Signale, die sich in einem Raum ausbreiten, werden durch die Reflexionen an den Wänden und Gegenständen im Raum linear verzerrt. Diese Verzerrung können beispielsweise die Sprachverständlichkeit erheblich verringern. Auch bei der akustischen Reproduktion von Musik in einem Raum sind sie unerwünscht, da sie das Klangbild «verfälschen». Um diese unerwünschten Verzerrungen rückgängig zu machen, müsste ein Filter vorgeschaltet werden, das das Inverse der Übertragungsfunktion des Raumes realisiert. Man betrachte das elektroakustische System dargestellt in Abbildung 2.1. Die Übertragungsfunktion zwischen dem Lautsprecher S und dem Mikrophon M sei $G(s)$. Das Filter mit der Übertragungsfunktion $H(s)$ wurde dem elektroakustischen System vorgeschaltet. Dieses soll so dimensioniert werden, dass beim Mikrophon M eine Gesamtübertragungsfunktion von $H(s) \cdot G(s) = 1$ herrscht. Folglich ist $H(s)$ bestimmt durch:

$$H(s) = \frac{1}{G(s)} \quad (2.1)$$

Da aber die Übertragungsfunktion einer Raumakustik im Allgemeinen nichtminimalphasig ist, stellt diese Inversion ein nichttriviales Problem dar. Ein solches inverses Filter wird instabil, wenn $G(s)$ nichtminimalphasig ist [4]. Die Nullstellen ausserhalb des Einheitskreises in der z-Ebene werden zu Polen, ein solches System ist instabil.

Es existieren einige «konventionelle» Methoden zur inversen Filterung, dargestellt in [1] und [2]. Alle diese Methoden sind nicht in der Lage eine exakte Inversion zu realisieren. Vielmehr versuchen diese Methoden den mittleren quadratischen Fehler zu minimieren («least squares error» (LSE)-Kriterium). Diese Methoden benutzen nur einen Übertragungskanal im Raum. Ist die Übertragungsfunktion dieses Pfades nichtminimalphasig, dann ist eine exakte Inversion nicht möglich (siehe [4]). Andere Methoden, basierend auf dem in [1] vorgestellten **multiple-input/output inversion theorem** (MINT), ermöglichen hingegen theoretisch eine *exakte* inverse Filterung einer akustischen Impulsantwort in einem Raum. Dies wird möglich, indem mehrere Übertragungspfade in demselben Raum benutzt werden. Es werden allgemein mehrere Lautsprecher und mehrere

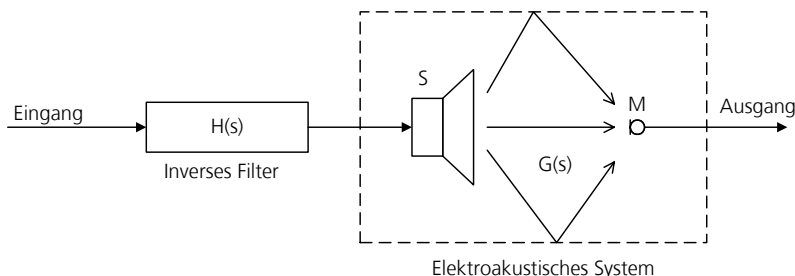


Abbildung 2.1: Einkanaliges elektroakustisches System mit vorgeschaltetem inversem Filter. Der Pfad zwischen dem Lautsprecher S und dem Mikrophon M hat die Übertragungsfunktion $G(s)$.

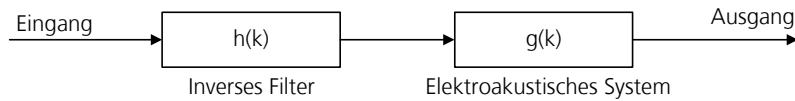


Abbildung 2.2: Einkanalige Inversion.

Mikrophone eingesetzt. Ein solches System wird ein multiple-input/output-system genannt. Bei den Ansätzen gemeinsam ist, dass sie den Raum als ein linear finite impulse response (FIR) System approximieren. Folglich können auch die inversen Filter in FIR-Struktur realisiert werden, so dass Stabilität garantiert ist.

2.2 Einkanalige Inversion

Wie bereits erwähnt benutzt diese Methode nur einen Übertragungskanal (single-input/single-output-system, vgl. Abbildung 2.2). Daher ist es im Allgemeinen nicht möglich eine exakte Inversion zu erreichen, ist die Übertragungsfunktion des Raumes doch allgemein nichtminimalphasig. Die Inversion wird gemäss dem LSE-Kriterium angenähert. Die diskrete, nichtminimalphasige Raumimpulsantwort des verwendeten Pfades sein $g(k)$, $h(k)$ die diskrete Impulsantwort des inversen FIR-Filters und $d(k)$ die gewünschte Gesamtimpulsantwort. Um eine Approximation an die exakte Inversion zu erhalten, ist die Forderung der Gesamtimpulsantwort ein diskreter Dirac-Stoss. Es gilt folglich folgende Beziehung:

$$d(k) = g(k) * h(k) \quad \text{wobei} \quad d(k) = \begin{cases} 1 & \text{wenn } k = 0 \\ 0 & \text{wenn } k > 0 \end{cases} \quad (2.2)$$

Diese diskrete Faltung kann auch in Matrixschreibweise dargestellt werden:

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} g(0) & & & 0 \\ g(1) & g(0) & & \\ \vdots & g(1) & \ddots & \\ g(m) & \vdots & & g(0) \\ & g(m) & & g(1) \\ & & \ddots & \vdots \\ 0 & & & g(m) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ \vdots \\ \vdots \\ h(m) \end{bmatrix} \quad (2.3)$$

oder in Kurzform:

$$D = GH \quad (2.4)$$

- D : Vektor der gewünschten Gesamtimpulsantwort. Länge: $2m + 1$
- H : Vektor der Impulsantwort des inversen Filters. Länge: $m + 1$
- G : Toeplitz-Matrix, aufgebaut aus dem Vektor der Raumimpulsantwort. Grösse: $(2m + 1) \times (m + 1)$

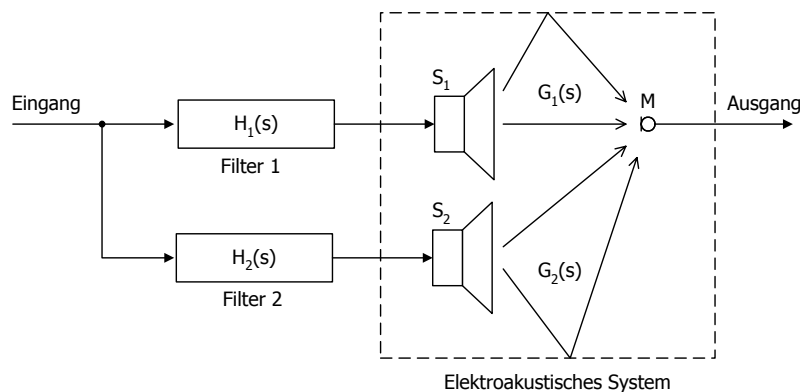


Abbildung 2.3: Zweikanaliges elektroakustisches System mit vorgeschalteten Filtern (Pfad 1 und 2)

2.2.1 Berechnung der inversen Impulsantwort

Um nun den Vektor $h(k)$, also die Koeffizienten des inversen FIR-Filters, zu erhalten, muss das Gleichungssystem (2.3) nach H aufgelöst werden. Dieses Gleichungssystem ist aber überdeterminiert, da $2m + 1 > m + 1$. Die Matrix G müsste quadratisch sein, damit eine Lösung der Form $H = G^{-1}D$ existieren würde¹. In der konventionellen LSE-Methode [1] wird H daher wie folgt berechnet:

$$H = (G^T G)^{-1} G^T D \quad (2.5)$$

G^T ist dabei die transponierte Matrix von G . Es ist also nicht möglich, das exakte Inverse der Raumimpulsantwort zu erhalten. In [1] wurde gezeigt, dass auch eine beliebige Verlängerung der Filterlänge nicht eine beliebig genaue Approximation bringt.

2.3 Mehrkanalige Inversion

In einigen Übertragungssystemen ist es möglich anstatt nur einem Übertragungskanal mehrere zu verwenden. So auch in akustischen Systemen. Anstatt nur einem Lautsprecher könnten mehrere werden (vgl. Abbildung 2.3), allgemein könnten auch mehrere Mikrophone Anwendung finden. Es entsteht ein multiple-input/output System (MIMO).

2.3.1 Das multiple-input/output theorem (MINT)

Eine *exakte* Inversion kann gemäss dem multiple-input/output theorem (MINT) [1] realisiert werden, wenn mehrere Übertragungspfade im Raum genutzt werden. Voraussetzung dafür ist aber, dass die Übertragungsfunktionen der verschiedenen Pfade keine gemeinsame Nullstelle besitzen. In [1] wurde gezeigt, dass es möglich ist schon mit einem zweikanaliges System mit zwei Lautsprechern und einem Mikrophon eine exakte Inversion zu erreichen. Auch diese Diplomarbeit beschränkte sich auf ein solches zweikanaliges System.

¹Ein lineares Gleichungssystem ist dann lösbar, wenn es gleichviele von einander linear unabhängige Gleichungen wie Unbekannte besitzt. Dann ist die Matrix G quadratisch und invertierbar.

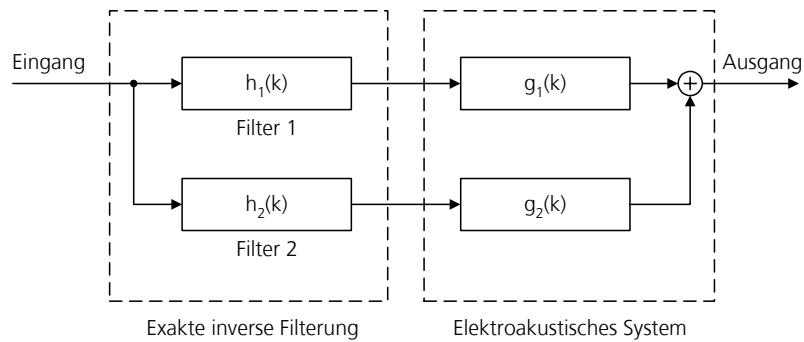


Abbildung 2.4: Zweikanalige Inversion basierend auf den multiple-input/output theorem (MINT).

2.4 Zweikanalige Inversion nach MINT

Man betrachte das System in Abbildung 2.4. Dieses enthält nun zwei Raumimpulsantworten $g_1(k)$ und $g_2(k)$, dementsprechend auch zwei Filter mit den Impulsantworten $h_1(k)$ und $h_2(k)$. Am Ort des Mikrophons werden die Signale beider Pfade überlagert. An diesem Ort soll das Signal völlig unverzerrt ankommen, also, abgesehen von der Laufzeitverzögerung, dem Eingangssignal entsprechen. Wenn die Filter eine exakte Inversion realisieren sollen, muss gelten:

$$d(k) = g_1(k) * h_1(k) + g_2(k) * h_2(k) \quad \text{wobei} \quad d(k) = \begin{cases} 1 & \text{wenn } k = 0 \\ 0 & \text{wenn } k > 0 \end{cases} \quad (2.6)$$

Wiederum können diese diskreten Faltungen und die Superposition in Matrixschreibweise dargestellt werden:

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} g_1(0) & & 0 & g_2(0) & & 0 \\ g_1(1) & g_1(0) & & g_2(1) & g_2(0) & \\ \vdots & g_1(1) & \ddots & \vdots & g_2(1) & \ddots & g_2(0) \\ g_1(m) & \vdots & g_1(1) & g_2(m) & \vdots & g_2(1) & \\ & g_1(m) & \vdots & g_2(m) & \vdots & & \\ 0 & & \ddots & g_1(m) & 0 & \ddots & g_2(m) \end{bmatrix} \begin{bmatrix} h_1(0) \\ \vdots \\ h_1(m-1) \\ h_2(0) \\ \vdots \\ h_2(m-1) \end{bmatrix} \quad (2.7)$$

oder in Kurzform:

$$D = G_1(z)H_1(z) + G_2(z)H_2(z) = [G_1 \ G_2] \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = GH \quad (2.8)$$

- D : Vektor der gewünschten Gesamtimpulsantwort. Länge: $2m$
- H_1 : Vektor der Impulsantwort des inversen Filters Pfad 1. Länge: m
- H_2 : Vektor der Impulsantwort des inversen Filters Pfad 2. Länge: m
- G_1 : Toeplitz-Matrix, aufgebaut aus dem Vektor der Raumimpulsantwort Pfad 1. Die letzte Spalte und die letzte Zeile wurden weggelassen. Grösse: $2m \times m$
- G_2 : Toeplitz-Matrix, aufgebaut aus dem Vektor der Raumimpulsantwort Pfad 2. Die letzte Spalte und die letzte Zeile wurden weggelassen. Grösse: $2m \times m$
- H : Vektor zusammengesetzt aus H_1 und H_2 . Länge: $2m$
- G : Matrix zusammengesetzt aus G_1 und G_2 . Grösse: $2m \times 2m$

2.4.1 Berechnung der inversen Impulsantwort

Die Matrizen G_1 und G_2 sind Toeplitz-Matrizen zusammengesetzt aus dem Vektor der Raumimpulsantwort g_1 bzw. g_2 wie bei der einkanaligen Version, aber mit dem Unterschied, dass hier die letzte Spalte und die letzte Zeile weggelassen wurden. So wird erreicht, dass diese Matrizen die Grösse $2m \times m$ bekommen. Der Grund hierfür ist folgender: Damit das Gleichungssystem (2.8) nach H aufgelöst (G invertierbar) werden kann, muss G quadratisch sein. Dann existiert eine eindeutige Lösung:

$$\begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = [G_1 \ G_2]^{-1} D \quad (2.9)$$

Doch was bedeutet es, wenn die letzte Spalte und Zeile einfach weggelassen werden? Betrachtet man die Matrizenmultiplikation in (2.8) genauer, stellt man fest, dass ein Weglassen der letzten Spalte in Matrix G_1 bzw. G_2 nur das letzte Element des Vektors H_1 bzw. H_2 betrifft. Im Lösungsvektor D fehlen die Summanden $g_1(i)h(m)$ bzw. $g_2(i)h(m)$. Anders ausgedrückt: Bei der Lösung (2.9) wird auf den letzten Wert des Impulsantwort h_1 bzw. h_2 «verzichtet»; dieser kommt im Gleichungssystem (2.8) gar nicht mehr vor. Folglich sind die Impulsantworten der Filter h_1 und h_2 um einen Wert kürzer als die Raumimpulsantworten g_1 und g_2 .

2.5 Schlussfolgerungen

Beide Verfahren liefern also eine Lösung des Inversionsproblems, wobei das zweikanalige eine rechnerisch exakte Lösung liefert. Das einkanalige Verfahren bietet zwar prinzipbedingt keine exakte Lösung, jedoch eine mit minimiertem quadratischen Fehler (LSE). Beiden Verfahren ist gemeinsam, dass sie sehr rechenaufwändig sind. So muss in (2.5) eine Matrix der Grösse $2m + 1 \times m + 1$ und in (2.9) eine Matrix der Grösse $2m \times 2m$ invertiert werden. Wobei $m + 1$ die Länge der berücksichtigten Raumimpulsantwort ist. In [2] wurde auf Algorithmen wie der Levinson-Durbin-Algorithmus hingewiesen, der in angepasster Form zur Lösung dieser speziellen Gleichungssystemen mit Toeplitz-Block-Matrizen eingesetzt werden kann. Damit können diese effizient gelöst werden. Natürlich liefert die Gaussche Variante die Lösung, welche im folgenden auch verwendet wurde.

3 Entwicklung und Resultate

Die im vorhergehenden Kapitel vorgestellten in der Literatur bekannten Methoden zur Raumakustik-Inversion sollen nun umgesetzt werden. Dabei sind zwei Stufen vorgesehen: Eine Simulation in MATLAB und nachfolgend eine Echtzeit-Realisierung der Inversions-Filter mit einem DSP. Um jedoch die inversen Filter berechnen zu können, müssen natürlich erst die Raumimpulsantworten bekannt sein. Daher wurden in einem ersten Schritt Impulsantworten von zwei verschiedenen Pfaden in einem Raum aufgenommen. Im nächsten Schritt wurden die inversen Filter berechnet, die Wirkung dieser simuliert und die beiden Verfahren miteinander verglichen. Schliesslich wurden mit einem DSP die inversen Filter als Echtzeit-System realisiert und die Wirkung des Systems mit den inversen Filtern gemessen.

3.1 Bestimmung der Raumimpulsantworten

Als erstes mussten Raumimpulsantworten gemessen werden, die ein Übertragungspfad in einem Raum charakterisieren. Im Hinblick auf die vorhandene Rechenkapazität des PCs und des DSPs wurde schnell klar, dass ein Raum mit einer möglichst kurzen Impulsantwort verwendet werden muss, damit der signifikante Teil mit einer möglichst geringen Anzahl Samples erfasst und in die Rechnung einbezogen werden kann. Der Testraum soll also möglichst klein sein und eine hohe Dämpfung aufweisen. Da keiner der Räume an der HSR beide Kriterien erfüllte, musste auf die Privatwohnung des Autors ausgewichen werden. Dort diente das Schlafzimmer/Büro als Messraum.

Die Messung erfolgte mittels der «Deconvolution»-Methode. Weisses Rauschen wird auf ein im zu messenden Raum befindlichen Lautsprecher gegeben. Dieses Signal wird von einem Mikrophon an einer anderen Stelle im Raum wieder aufgenommen. Ein PC mit einer Soundkarte nimmt beide Signale auf und legt die Daten in einem Microsoft-PCM-Wavefile ab. Diese können nun mit

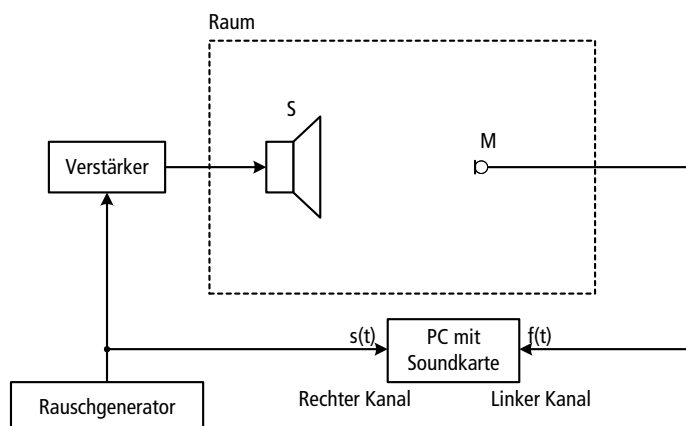


Abbildung 3.1: Anordnung zur Messung der Impulsantworten verschiedener Pfade in einem Raum.

MATLAB weiterverarbeitet werden. Abbildung 3.1 zeigt die Messanordnung.

3.1.1 Messverfahren

Die Messung wurde mit oben beschriebener Messanordnung durchgeführt. Um Impulsantworten von zwei verschiedenen Pfaden zu erhalten, wurden zwei Lautsprecher an verschiedenen Positionen im Raum plazierte. Dabei wurde darauf geachtet, dass die beiden Position nicht eine symmetrische Anordnung ergaben. Beide Pfade wurden ausgemessen.

Nachdem der Raum ca. fünf Sekunden mit dem Rauschsignal beschallt worden ist, wurde die Aufzeichnung der Rauschsignale gestartet. Es wurden jeweils ca. 12 Sekunden aufgezeichnet. Die Samplingrate betrug 44.1 kHz. Die aufgezeichneten Wave-Files¹ sind folgende:

f-leftpath-44k-01.wav	Mikrophonsignal des Pfades linker Lautsprecher - Mikrophon
s-leftpath-44k-01.wav	Originales Rauschsignal
f-rightpath-44k-01.wav	Mikrophonsignal des Pfades rechter Lautsprecher - Mikrophon
s-rightpath-44k-01.wav	Originales Rauschsignal

Das originale Rauschsignal $s(t)$ wird direkt von Rauschgenerator in den rechten Kanal der Soundkarte eingespeist. Das Mikrophonsignal $f(t)$ ist das von Raum verzerrte Rauschsignal und wird in den linken Kanal der Soundkarte eingespeist. So ist es möglich die beiden Rauschsignale mit ihrer korrekten Phasenbeziehung zueinander aufzunehmen, so dass die Laufzeit des Schalls zwischen Lautsprecher S und Mikrophon M in der Impulsantwort herausgelesen werden kann.

3.1.2 Auswertung

Die aufgenommenen Rauschsignale wurden in der Folge ausgewertet. Interessant ist, dass aus diesen die Raumimpulsantwort recht genau bestimmt werden kann.

Die Bestimmung der Raumimpulsantworten erfolgt über die «Deconvolution»-Methode. Folgende Beziehung bildet die Grundlage dieser Messmethode:

$$H(\omega) = \frac{\Phi_{sf}(\omega)}{\Phi_s(\omega)} \quad (3.1)$$

wobei:

$$\Phi_s(\omega) = \lim_{T \rightarrow \infty} \frac{1}{T} |S_T(\omega)|^2 \quad \text{Leistungsdichtespektrum von } s(t) \quad (3.2)$$

$$\Phi_{sf}(\omega) = \lim_{T \rightarrow \infty} \frac{1}{T} S_T^*(\omega) F_T(\omega) \quad \text{Kreuzleistungsdichtespektrum von } s(t) \text{ und } f(t) \quad (3.3)$$

$$S_T(\omega) = \int_0^T s(t) e^{-j\omega t} dt \quad \text{Fouriertransformation von } s(t) \quad (3.4)$$

$$F_T(\omega) = \int_0^T f(t) e^{-j\omega t} dt \quad \text{Fouriertransformation von } f(t) \quad (3.5)$$

Mit der Beziehung (3.1) lässt sich die Übertragungsfunktion $H(\omega)$ berechnen. Eine Fourier-Rücktransformation liefert schliesslich $h(t)$, die Impulsantwort. Diese Berechnungen wurden mit MATLAB² ausgeführt.

¹FILES: ROOT/meas/systemid/

²FILES: ROOT/sim/singlechannel/decon.m und ROOT/sim/multichannel/decon.m

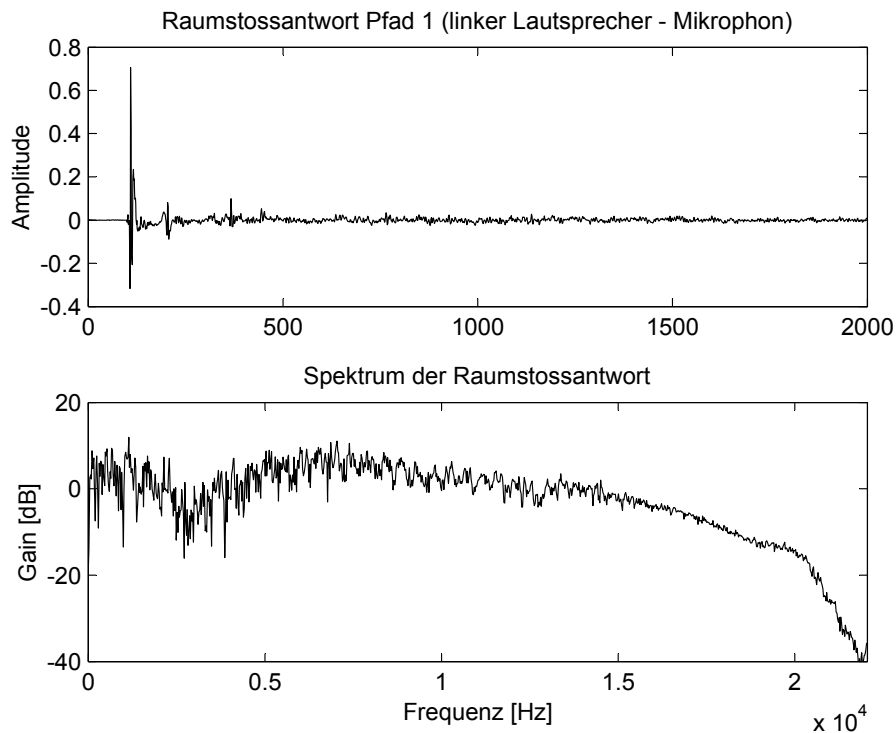


Abbildung 3.2: Auswertungsergebnisse: Raumimpulsantwort und zugehöriges Leistungsspektrum des ersten Pfades

Die Ergebnisse der Berechnung sind in Abbildung 3.2 und 3.3 dargestellt. Es kann festgestellt werden, dass 2000 Samples die Raumimpulsantwort genügend genau beschreiben. Nach dieser Zeit (45 ms) sind die Amplituden schon fast Null. Das Leistungsspektrum zeigt bei sehr hohen Frequenzen einen deutlichen Abfall, der auf die beschränkte Übertragungsbandbreite des Messsystems zurückzuführen ist (Anti-Aliasing-Filter der Soundkarte, Frequenzgang der Lautsprecher, des Mikrophons und der Verstärker, Kabel usw.). In der gemessenen Übertragungsfunktion sind also auch die Einflüsse dieses Messequipments³ enthalten. Es wird jedoch angenommen, dass dieses in einem Nutzband von 100 Hz bis 10 kHz jedoch einen annähernd linearen Frequenzgang besitzt.

3.2 Simulation in MATLAB

Die Filterkoeffizienten der einkanaligen wie auch die zweikanaligen Inversion werden in MATLAB berechnet. Dazu werden die vorgestellten mathematischen Methoden verwendet. Nach einer Berechnung der Filter sollen die Wirkung dieser simuliert, und ein Vergleich der Resultate des einkanaligen mit denen des zweikanaligen Falles gezogen werden.

3.2.1 Rahmenbedingungen

Die beschränkt verfügbaren Ressourcen an Rechenleistung und Speicherkapazität setzen gewisse Rahmenbedingungen an die Simulation und die DSP-Implementation. Auch die Bandbreiten der

³Eine Liste des Messequipments ist im Anhang B zu finden.

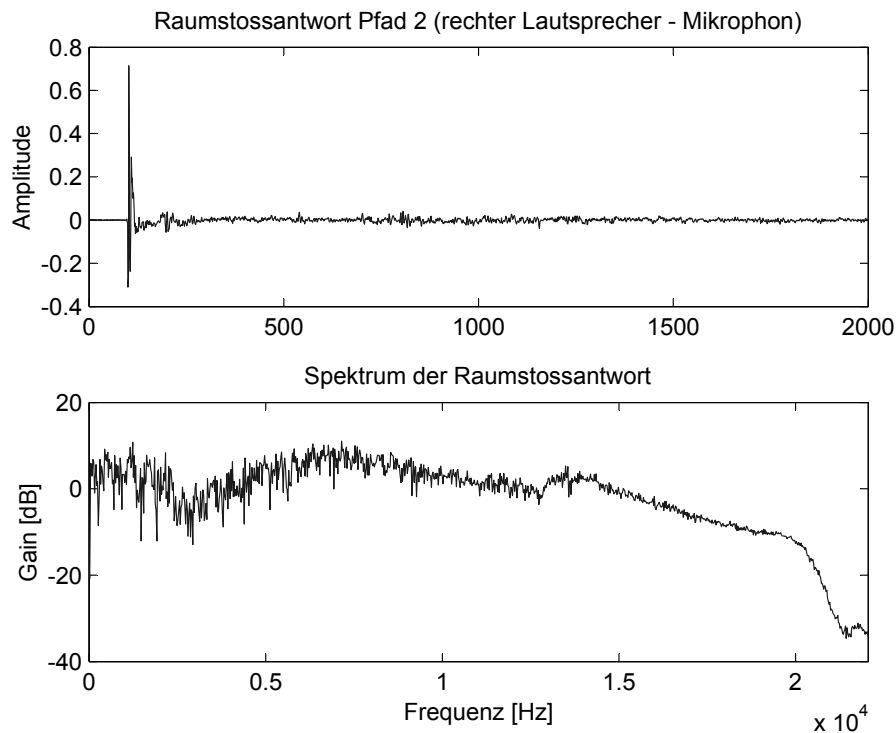


Abbildung 3.3: Auswertungsergebnisse: Raumimpulsantwort und zugehöriges Leistungsspektrum des zweiten Pfades

realen Systeme erfordern gewisse Einschränkungen.

Beschränkung der Länge der Raumimpulsantwort

Obwohl für Gleichungssysteme mit Toeplitz-Block-Matrizen optimierte Lösungsverfahren existieren wie der Levinson-Durbin-Algorithmus [2], wurde darauf der Einfachheit halber verzichtet ein solches zu verwenden. Zur Matrizen-Inversion in MATLAB wurde die Standardfunktion `inv()` verwendet. Auf dem verwendeten PC mit Intel PII 300 MHz und 256 MB RAM unter MATLAB 6 dauerte die Inversion der Matrix G aus (2.8) ca. 15 bis 20 Minuten, wenn 2000 Samples der Raumimpulsantwort miteinbezogen wurden; ein gerade noch erträgliches Mass. Folglich konnten nicht mehr als 2000 Samples der Raumimpulsantwort verwendet werden, was sich aber in den Versuchen mit dem stark gedämpften Testraum als genügend erwies, da sie nach 2000 Samples weitgehend abgeklungen ist.

Bandpass als geforderte Übertragungsfunktion

Die Gleichung (2.2) beschreibt, dass die Faltung der Raumimpulsantwort mit der Filterimpulsantwort, also die Gesamtimpulsantwort, einen Dirac-Stoß ergeben soll. Diese Forderung ist aber nur für theoretische Untersuchungen sinnvoll. Ein reales System kann diese Forderung ohnehin nicht erfüllen, da die Bandbreite nach oben wie nach unten, nicht zuletzt durch eine endliche Samplingfrequenz des AD-Wandlers vor dem DSP bedingt, begrenzt ist. Daher wurde im Hinblick auf die DSP-Implementation eine andere Forderung gewählt: Eine Bandpass-Impulsantwort. Genausogut könnte man eine andere Impulsantwort fordern, die dann nach dem LSE-Kriterium approximiert

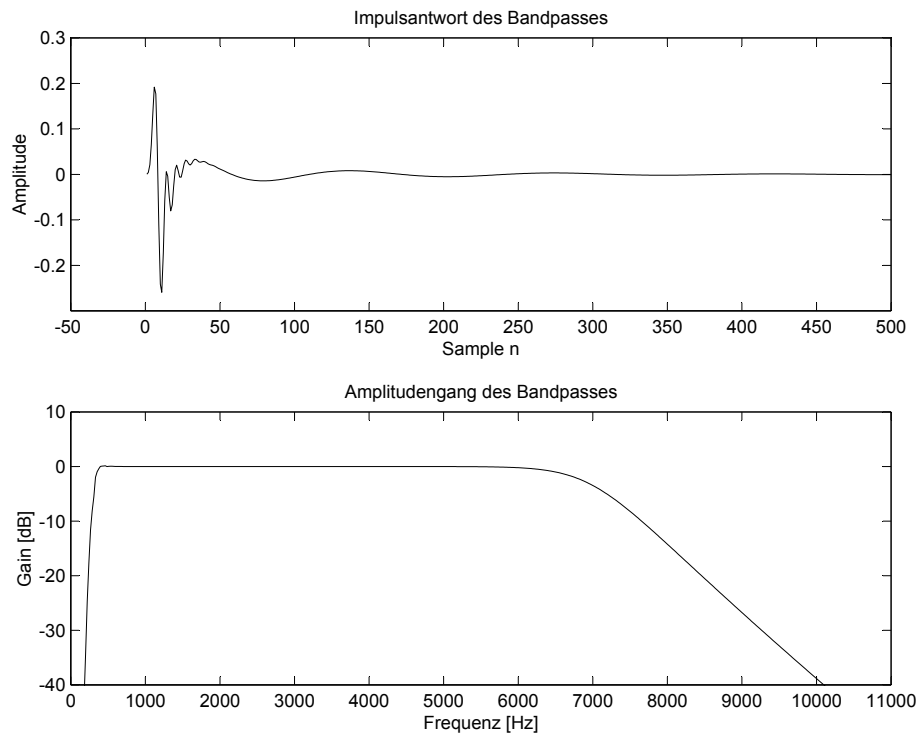


Abbildung 3.4: Erste 500 Samples der Impulsantwort und Leistungsspektrum des zu approximierenden Bandpasses (Butterworth Ordnung 8, Grenzfrequenzen: 300 Hz und 7000 Hz)

(einkanalige Inversion) oder exakt gelöst (zweikanalige Inversion) würde.

Für die Simulation wie auch für die Implementation wurde daher als Forderung die Impulsantwort eines Butterworth-Bandpasses gewählt (siehe Abbildung 3.4). Dieser ist achter Ordnung und besitzt die Grenzfrequenzen 300 Hz und 7000 Hz. Zudem muss die geforderte Bandpassimpulsantwort um mindestens die Laufzeit des Schalles von Lautsprecher zu Mikrofon (direkter Pfad) verzögert werden, um nicht ein nichtkausales Filter zu fordern.

Downsampling der Filterkoeffizienten

Im Hinblick auf die DSP-Implementation der Filter muss beachtet werden, dass die Filterkoeffizienten dezimiert werden können. Dies, da die Samplingfrequenz des AD-Wandlers lediglich auf 22050 Hz eingestellt werden kann. Der Grund hierfür ist die beschränkte Rechenleistung des DSP. Damit möglichst lange Filter realisiert werden können, muss die Samplingrate möglichst klein gewählt werden. Gleichzeitig wäre ein grosse Bandbreite wünschenswert, was eine hohe Samplingrate bedingen würde. Der getroffene Kompromiss liegt bei 22050 Hz. Bei dieser Frequenz kann mit dem verwendeten DSP ein FIR-Filter mit 800 Koeffizienten realisiert werden. Somit kann das Filter 36.3 ms, also 80% der berücksichtigten Raumimpulsantwort abdecken, während es bei 44100 Hz Samplingrate nur gerade 9.1 ms, also nur 20% der berücksichtigten Raumimpulsantwort ist. Dazwischen liegt der Faktor 4!

Die Simulation jedoch wird mit Daten durchgeführt, die mit 44.1 kHz gesampelt wurden. Entsprechend sind auch die Filterkoeffizienten für einen 44.1 kHz-Betrieb berechnet. Diese müssen also vor dem Download downsampled werden. Damit dabei kein Aliasing entsteht, darf der Amplitudengang der Filters bei der halben neuen Samplingfrequenz – also bei 11025 Hz – keine si-

gnifikante Amplitude mehr haben. Um dies zu erreichen muss die obere Grenzfrequenz des Soll-Bandpasses (siehe Abbildung 3.4) tief genug und die Ordnung hoch genug gewählt werden. Diese Parameter wurden so gewählt, dass die Amplitude des Filter-Amplitudenspektrums bei 11025 Hz weniger als -40 dB beträgt.

3.2.2 Einkanalige Inversion

Zuerst wurde ein System gemäss Abbildung 2.1 betrachtet. In diesem existiert nur ein elektroakustischer Übertragungspfad. Der Theorie nach kann mit einem solchen einkanaligen System keine exakte Inversion erreicht werden, sondern lediglich eine Approximation nach dem least squared error-Kriterium (LSE). Diese Lösung liefert die Gleichung (2.5), die nun in MATLAB umgesetzt wurde.

In der Folge wurde die Inversion separat für beide Pfade mit unterschiedlichen Filterlängen simuliert. Einmal mit der vollen Filterlänge – dabei entspricht die Filterlänge der Länge der in der Rechnung berücksichtigten Raumimpulsantwort, also 2000 Samples – und einmal mit den 800 ersten Koeffizienten des Filters. So kann ein Vergleich zwischen der maximal möglichen Inversion und der Inversion, wie sie später vom Echtzeitsystem ausgeführt wird, gezogen werden.

Simulation mit voller Filterlänge

Basis der einkanaligen Inversion stellt natürlich wieder der Aufbau dar, wie er für die Messung der Raumimpulsantworten des Pfades 1 (linker Lautsprecher - Mikrophon) und des Pfades 2 (rechter Lautsprecher - Mikrophon) verwendet wurde. Für beide Pfade wurde je ein entsprechendes inverses Filter berechnet. Beide Fälle wurden einzeln simuliert und auf Unterschiede untersucht. Die Resultate sind in Abbildung 3.5 und 3.6 dargestellt.

Es fällt auf, dass sich der ist-Amplitudengang recht gut an den soll-Amplitudengang «anschmiegt» – wie auch die Gesamtimpulsantwort (Faltung der Raumimpulsantwort mit der Filterimpulsantwort) praktisch keine signifikanten Reflexionen mehr zeigt – jedoch mit einigen stärkeren lokalen Abweichungen. Besonders bei der Simulation des Pfades 1 sind die Abweichungen stellenweise recht gross. Der Fehler zwischen dem ist-Amplitudengang und dem soll-Amplitudengang (Betrag der Differenz der Amplitudengänge in Leistungs-dB) ist für beide Pfade in Abbildung 3.7 und 3.8 dargestellt. Dabei ist auffallend, dass der Fehler bei tiefen Frequenzen ca. -25 dB (Pfad 1) bis -30 dB (Pfad 2) und bei hohen Frequenzen ca. -60 dB (Pfad 2) bis -65 dB (Pfad 1) beträgt.

Zudem wurde der Fehler im Zeitbereich (Differenz zwischen soll-Impulsantwort und ist-Impulsantwort) als mean squared error (MSE) berechnet⁴, der ja durch die LSE-Methode minimiert worden ist.

Pfad	MSE
Pfad 1:	$5.861 \cdot 10^{-6}$
Pfad 2:	$1.794 \cdot 10^{-6}$

Auch der MSE zeigt für Pfad 1 ein grösserer Fehler als für Pfad 2. Beide liegen jedoch in der selben Grössenordnung. Insgesamt wird Pfad 2 deutlich besser entzerrt als Pfad 1. Offenbar ist die Übertragungsfunktion des Pfades 2 fast minimalphasig, während Pfad 1 eine grössere Abweichung zu einem Phasenminimum-System hat und folglich schlechter invertiert werden kann.

Simulation mit reduzierter Filterlänge

Die Simulation wurde wiederholt wie im vorhergehenden Abschnitt jetzt jedoch mit reduzierter Filterlänge; es werden nur die ersten 1600 Samples der berechneten Filterimpulsantwort verwendet. Dieser Versuch soll simulieren, wie sich das Echtzeitsystem verhält, wo nur eine beschränkte

⁴File: ROOT/sim/singlechannel/calcmse.m

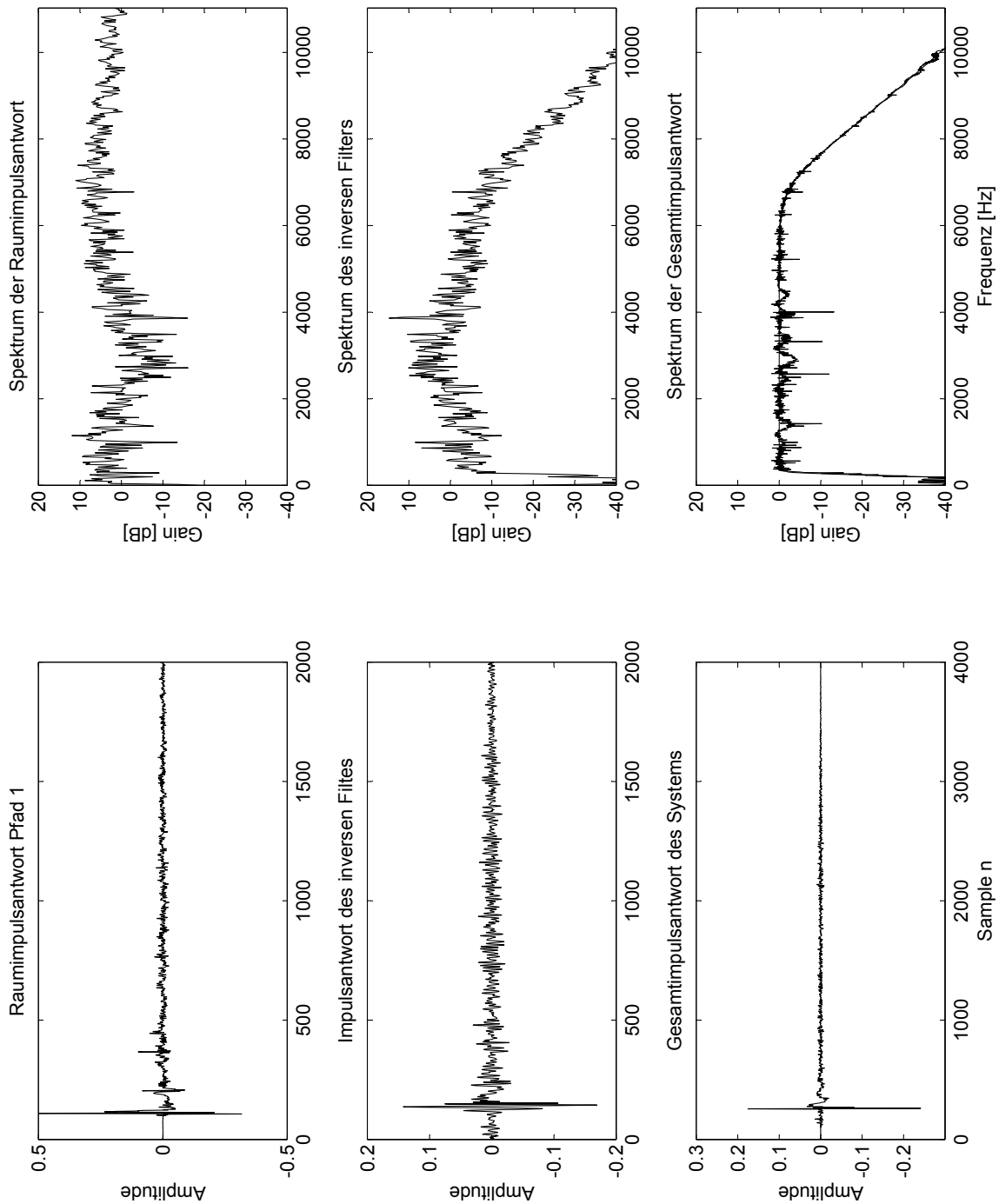


Abbildung 3.5: Simulationsergebnis einkanalige Inversion Pfad 1: Raumimpulsantwort, Impulsantwort des inversen Filters, Gesamtimpulsantwort (Faltung der zwei vorhergehenden) und zugehörige Leistungsspektren. Ins Spektrum der Gesamtimpulsantwort wurde zudem das Soll-Amplitudenspektrum eingezeichnet. Die Filterlänge beträgt 2000 Samples.

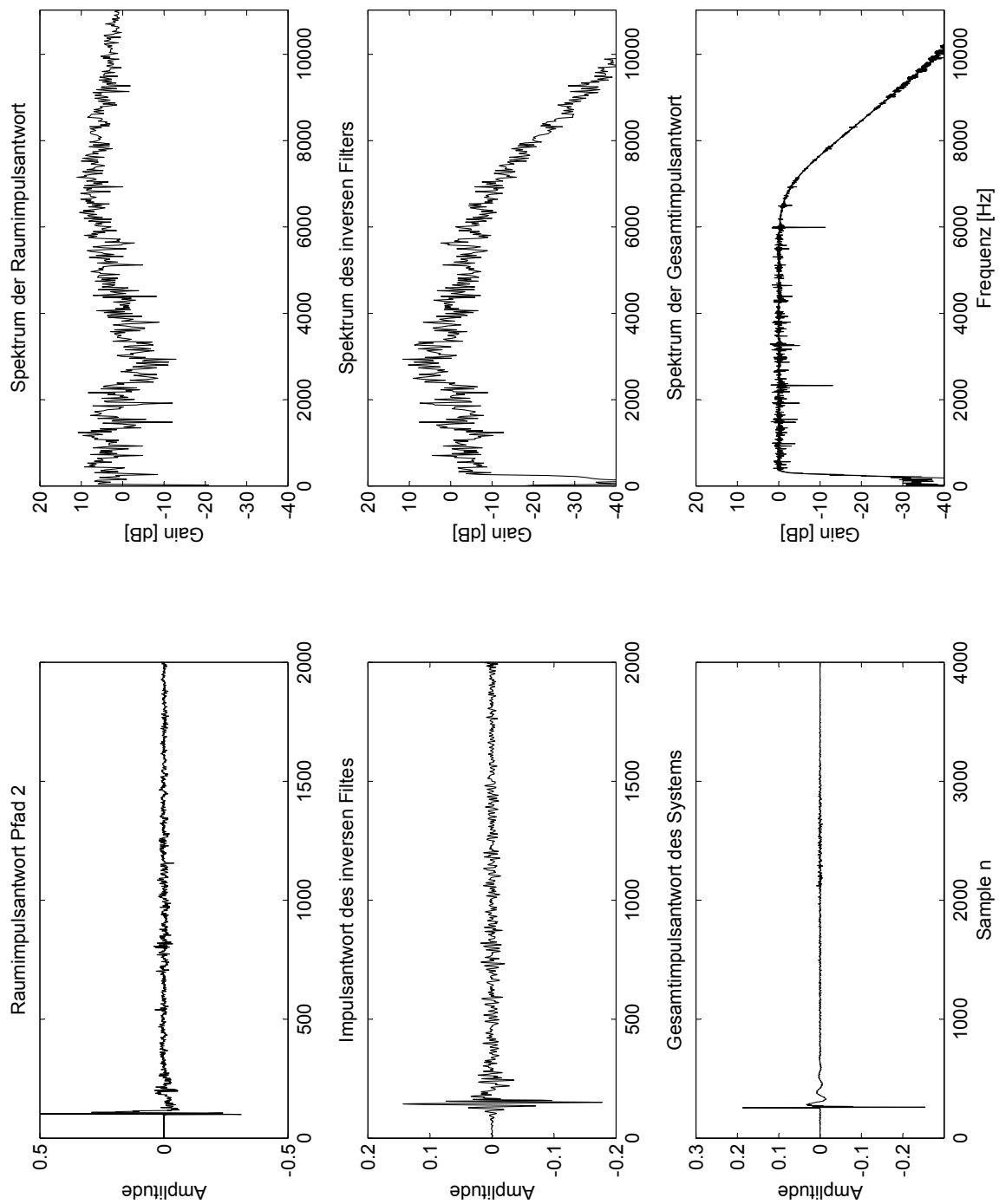


Abbildung 3.6: Simulationsergebnis einkanalige Inversion Pfad 2: Raumimpulsantwort, Impulsantwort des inversen Filters, Gesamtimpulsantwort (Faltung der zwei vorhergehenden) und zugehörige Leistungsspektren. Ins Spektrum der Gesamtimpulsantwort wurde zudem das soll-Amplitudenspektrum eingezeichnet, das hier jedoch praktisch nicht sichtbar ist, da es vom ist-Amplitudenspektrum fast vollständig verdeckt wird. Die Filterlänge beträgt 2000 Samples.

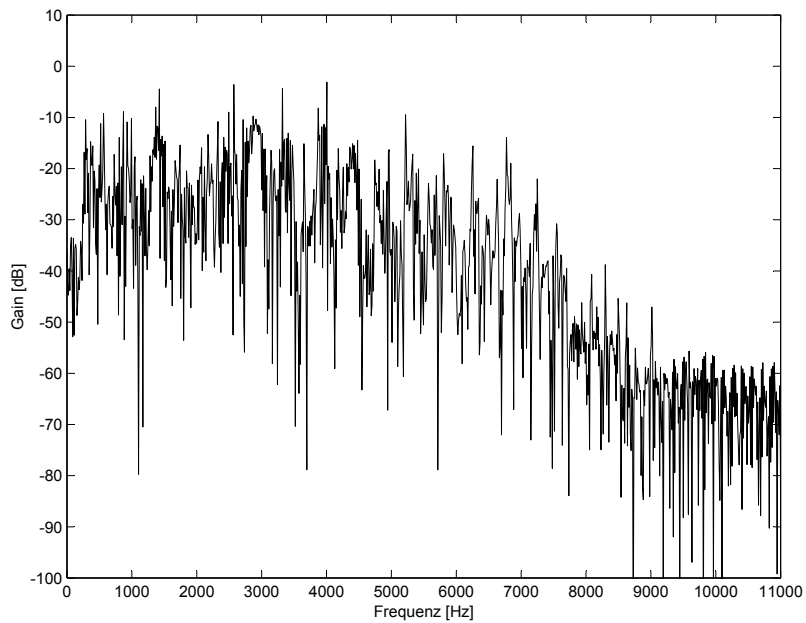


Abbildung 3.7: Pfad 1: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 2000 Samples.

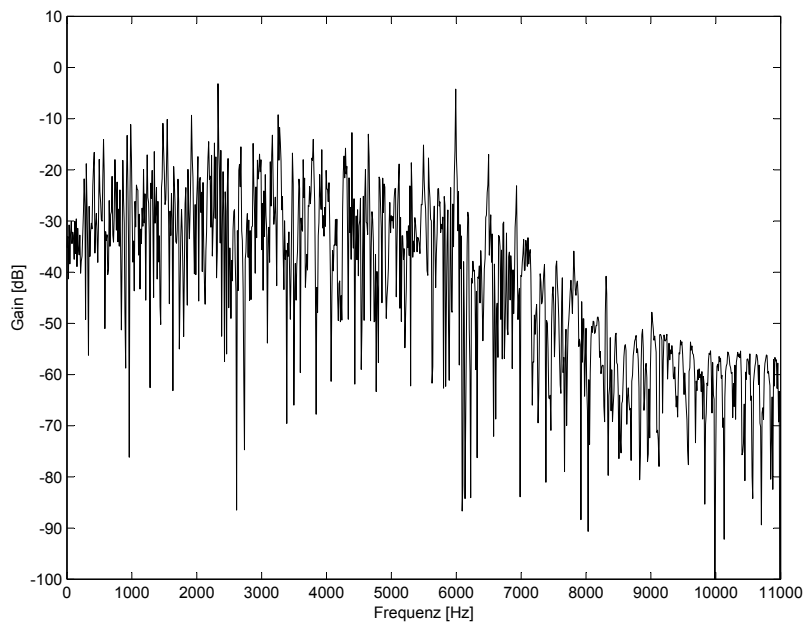


Abbildung 3.8: Pfad 2: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 2000 Samples.

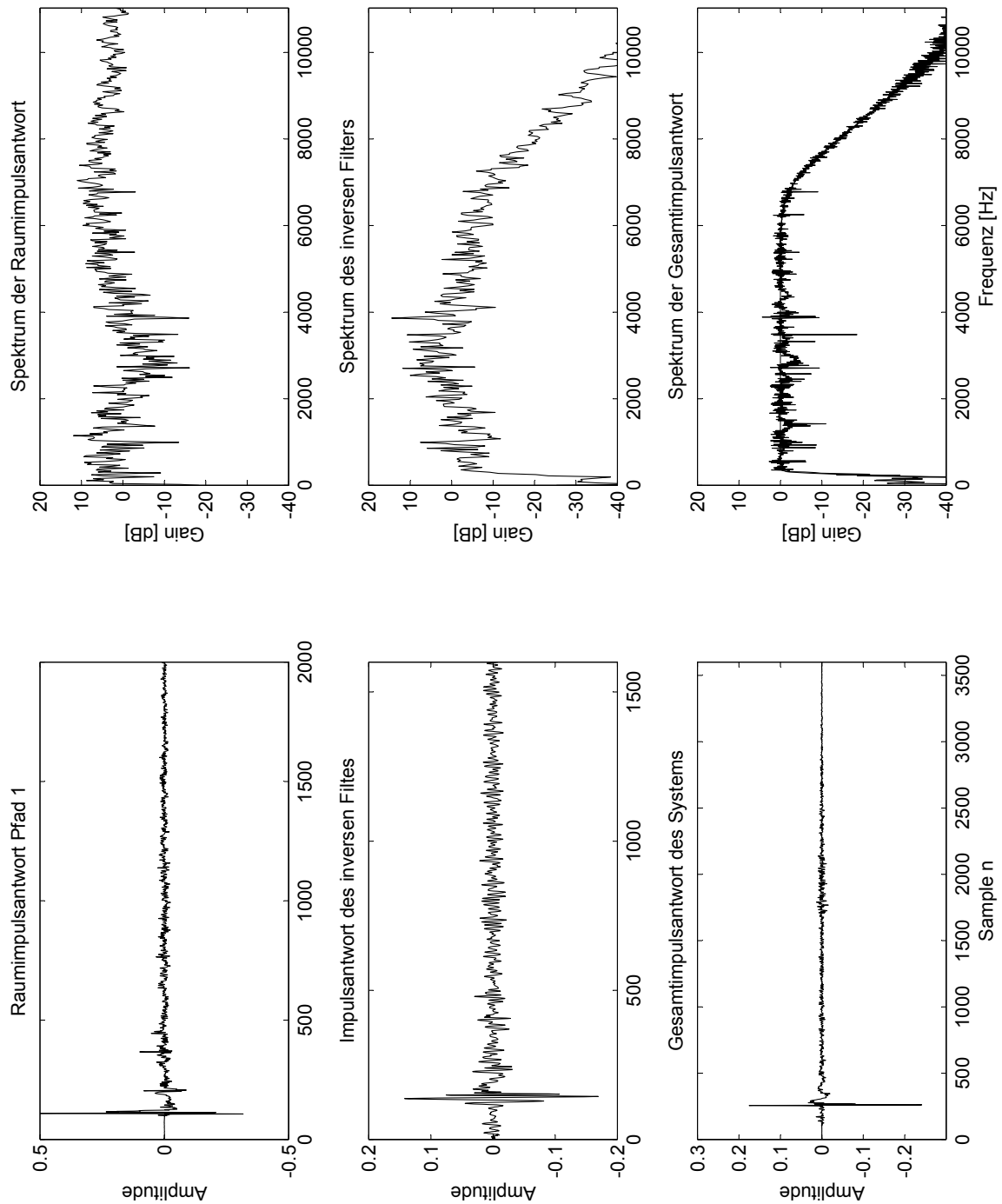


Abbildung 3.9: Simulationsergebnis einkanalige Inversion Pfad 1: Raumimpulsantwort, Impulsantwort des inversen Filters, Gesamtimpulsantwort (Faltung der zwei vorhergehenden) und zugehörige Leistungsspektren. Ins Spektrum der Gesamtimpulsantwort wurde zudem das Soll-Amplitudenspektrum eingezeichnet. Die Filterlänge beträgt 1600 Samples.

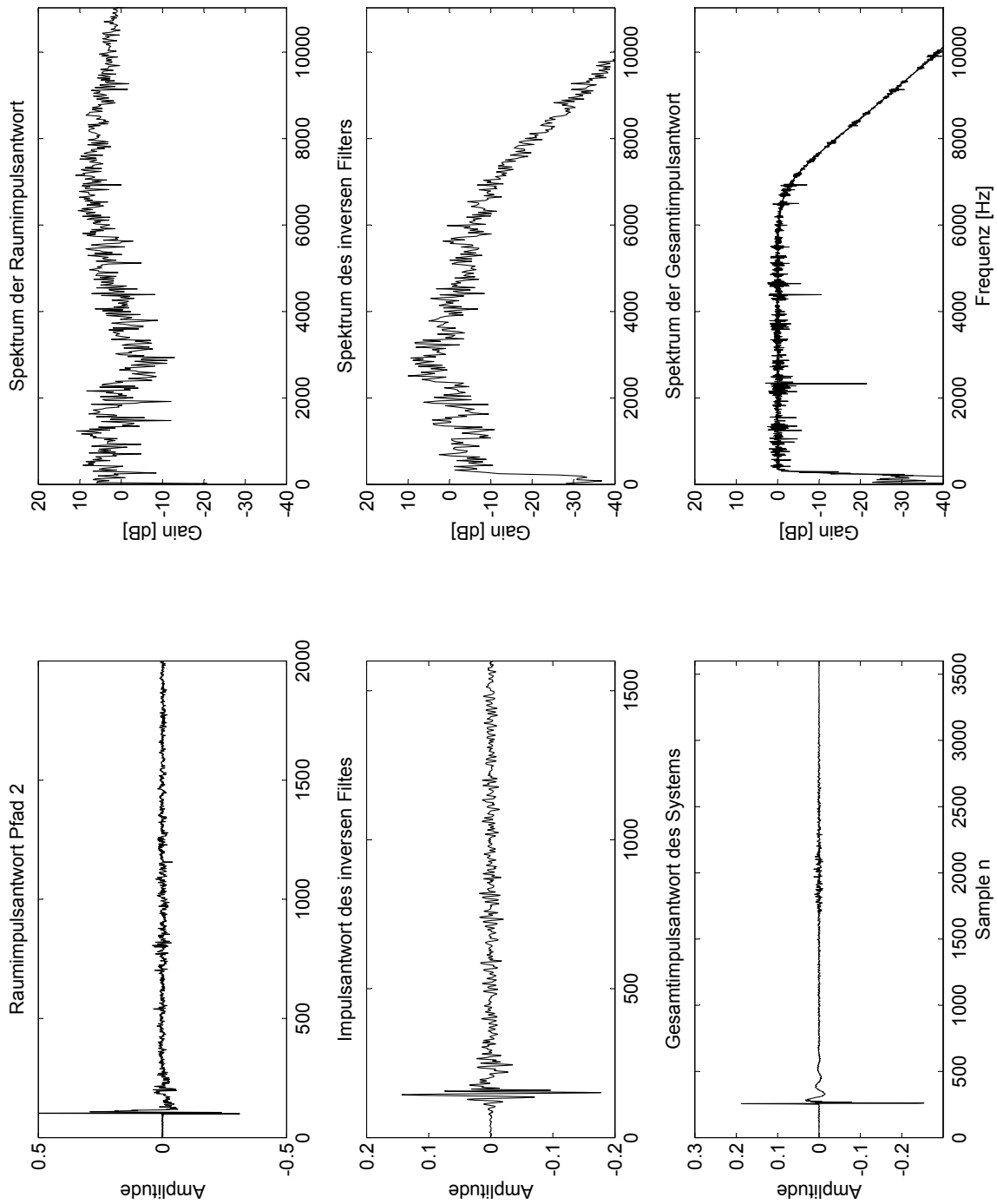


Abbildung 3.10: Simulationsergebnis einkanalige Inversion Pfad 2: Raumimpulsantwort, Impulsantwort des inversen Filters, Gesamtimpulsantwort (Faltung der zwei vorhergehenden) und zugehörige Leistungsspektren. Ins Spektrum der Gesamtimpulsantwort wurde zudem das soll-Amplitudenspektrum eingezeichnet, das hier jedoch praktisch nicht sichtbar ist, da es vom ist-Amplitudenspektrum fast vollständig verdeckt wird. Die Filterlänge beträgt 1600 Samples.

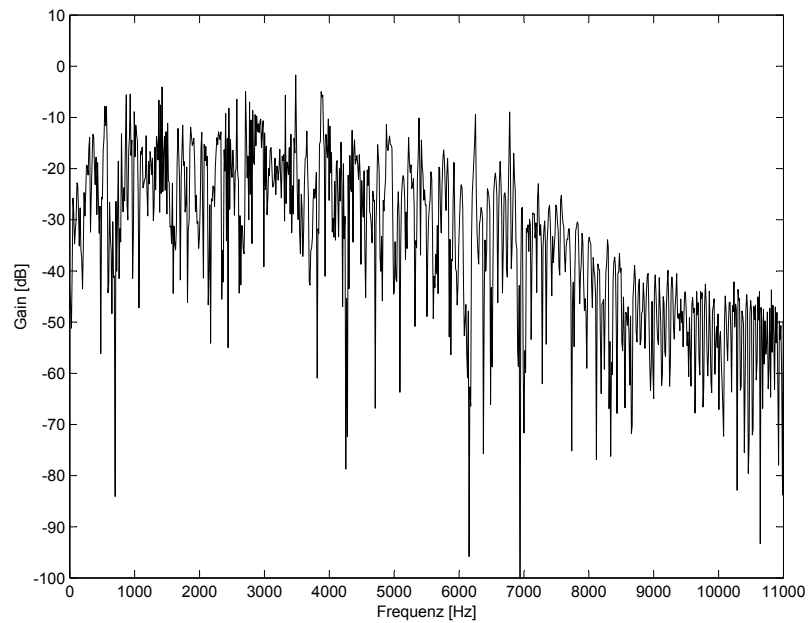


Abbildung 3.11: Einkanalige Inversion Pfad 1: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 1600 Samples.

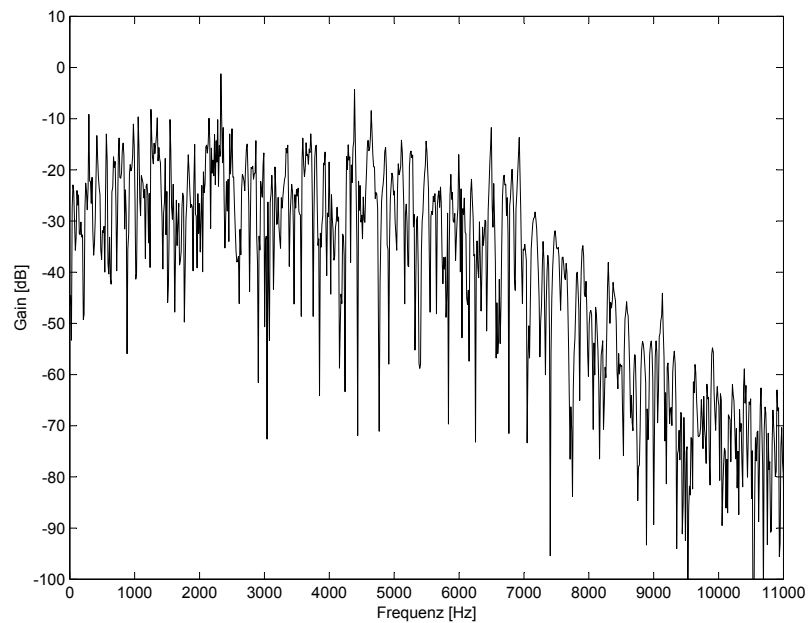


Abbildung 3.12: Einkanalige Inversion Pfad 2: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 1600 Samples.

Filterlänge eingesetzt werden kann. Versuche mit dem vorhandenen DSP-System haben ergeben, dass die maximale Filterlänge bei einer Samplingrate von 22050 Hz 800 Koeffizienten beträgt. Da die Simulation mit 44100 Hz-Daten läuft, können also 1600 Samples der Filterimpulsantwort verwendet werden. Die Koeffizienten werden ja vor dem Download noch um den Faktor zwei reduziert (Downsampling).

Wiederum wurden beide Pfade simuliert und die Ergebnisse in Abbildung 3.9 und 3.10 dargestellt. Die Übertragungsfunktionen zeigen sich jetzt viel «verrauschter» als bei der Simulation mit der vollen Filterlänge. Auch die Gesamtimpulsantworten zeigen ab dem Sample 1600 ein kleines «Rauschpaket». Dieses Phänomen ist auch einfach zu erklären: Stellt man sich die Faltung der Raumimpulsantwort mit der Filterimpulsantwort vor als ein Übereinanderschieben und Aufsummieren der Produkte der ersteren mit der umgedrehten letzteren, so wird klar, dass nach der Verschiebung 1600 – wo die Filterimpulsantwort dann zu Ende ist – die Faktoren fehlen, die die Werte der Raumimpulsantwort nach Sample 1600 kompensieren könnten.

Doch insgesamt wird auch mit dieser reduzierten Filterlänge ein gutes Resultat erzielt. Die Absenkung im Amplitudengang der Raumübertragungsfunktion bei 3 kHz wird bei der Simulation mit der vollen Filterlänge wie auch hier sehr gut ausgeglichen. Dass das Amplitudenspektrum der Gesamtübertragungsfunktion etwas «verrauscht» ist, kann mit der Fouriertheorie erklärt werden, indem man die Gesetzmässigkeit der Symmetrie der beiden Bereiche (Zeitbereich – Frequenzbereich) ausnützt. Man kann folgende Aussage machen: Signifikante Werte weit hinten in der Impulsantwort («hohe Frequenzen») bedeuten eine sich schnell ändernder Amplitudengang («hohe Frequenzanteile im Signal»)⁵. Daher erscheint der Amplitudengang mehr «verrauscht». Der Grundverlauf jedoch bleibt der selbe.

Der Fehler zwischen Soll- und Ist-Übertragungsfunktion, dargestellt in Abbildung 3.11 und 3.10 hat etwas zugenommen, bei Pfad 1 vor allem auch in den höheren Frequenzen, dort beträgt der Fehler jetzt –50 dB während er bei Pfad 2 mit –70 dB abgenommen hat. Bei den tiefen Frequenzen ist aber bei beiden Pfaden eine Zunahme zu vermerken: –20 dB bei Pfad 1 und –25 dB bei Pfad 2. Im Mittel ist der Fehler in den tieferen Frequenzen um ca. 5 dB (Faktor 3.2) grösser. Eine Zunahme zeigt auch der mean squared error (MSE):

Pfad	MSE	Zunahme
Pfad 1:	$8.239 \cdot 10^{-6}$	41%
Pfad 2:	$3.237 \cdot 10^{-6}$	80%

3.2.3 Zweikanalige Inversion

Werden zwei Übertragungspfade für die Inversion der Raumakustik verwendet, so sagt die Theorie ein signifikant besseres Resultat voraus. Diese Situation wurde natürlich auch simuliert. In der vorangegangenen Simulation wurde je ein Pfad für sich alleine verwendet und ein inverses Filter berechnet. Jetzt werden beide diese Pfade, also genau die gleiche Messanordnung wie für die einkanalige Inversion, für die zweikanalige Inversion verwendet.

Wiederum wurden zwei Simulationen durchgeführt. Eine mit einer Filterlänge von 1999 Samples⁶, und eine mit der Filterlänge von 1600 Samples.

Simulation mit voller Filterlänge

Es gilt nun das Gleichungssystem (2.8) zu lösen. Im Unterschied zum Gleichungssystem (2.5) der einkanaligen Version, ist dieses nicht überdeterminiert, sondern genau determiniert. Die Lösung

⁵Dies wird auch beim «zero padding» ausgenützt, aber mit umgekehrter Wirkung: Es bewirkt eine «Glättung» des Amplitudenganges.

⁶Bei der zweikanaligen Inversion sind, wie im Theorieteil beschrieben, die Filterimpulsantworten um einen Wert kürzer als die Raumimpulsantworten.

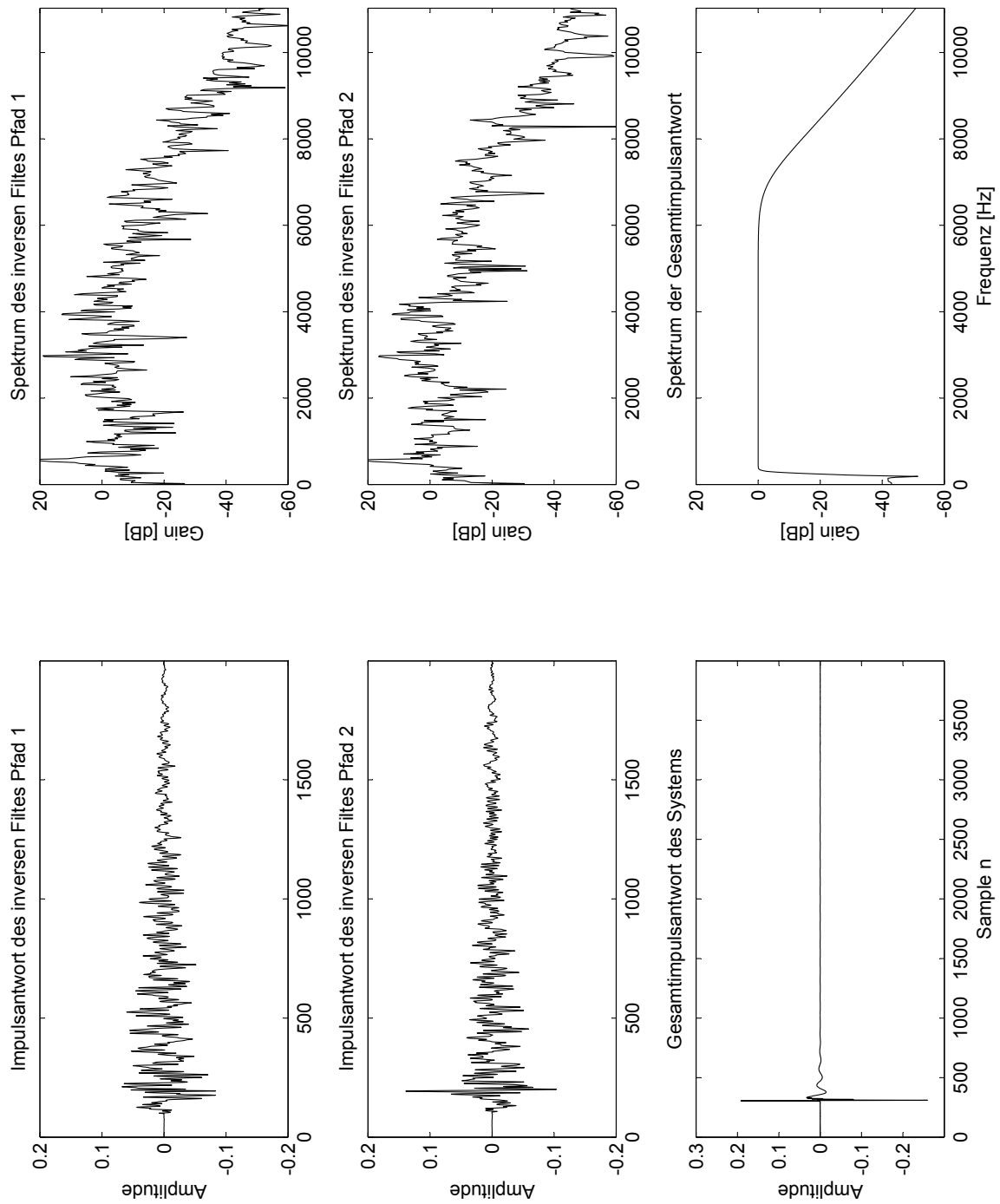


Abbildung 3.13: Simulationsergebnis zweikanalige Inversion: Impulsantwort der beiden inversen Filter, Gesamtimpulsantwort und zugehörige Leistungsspektren. Filterlänge: 1999 Samples.

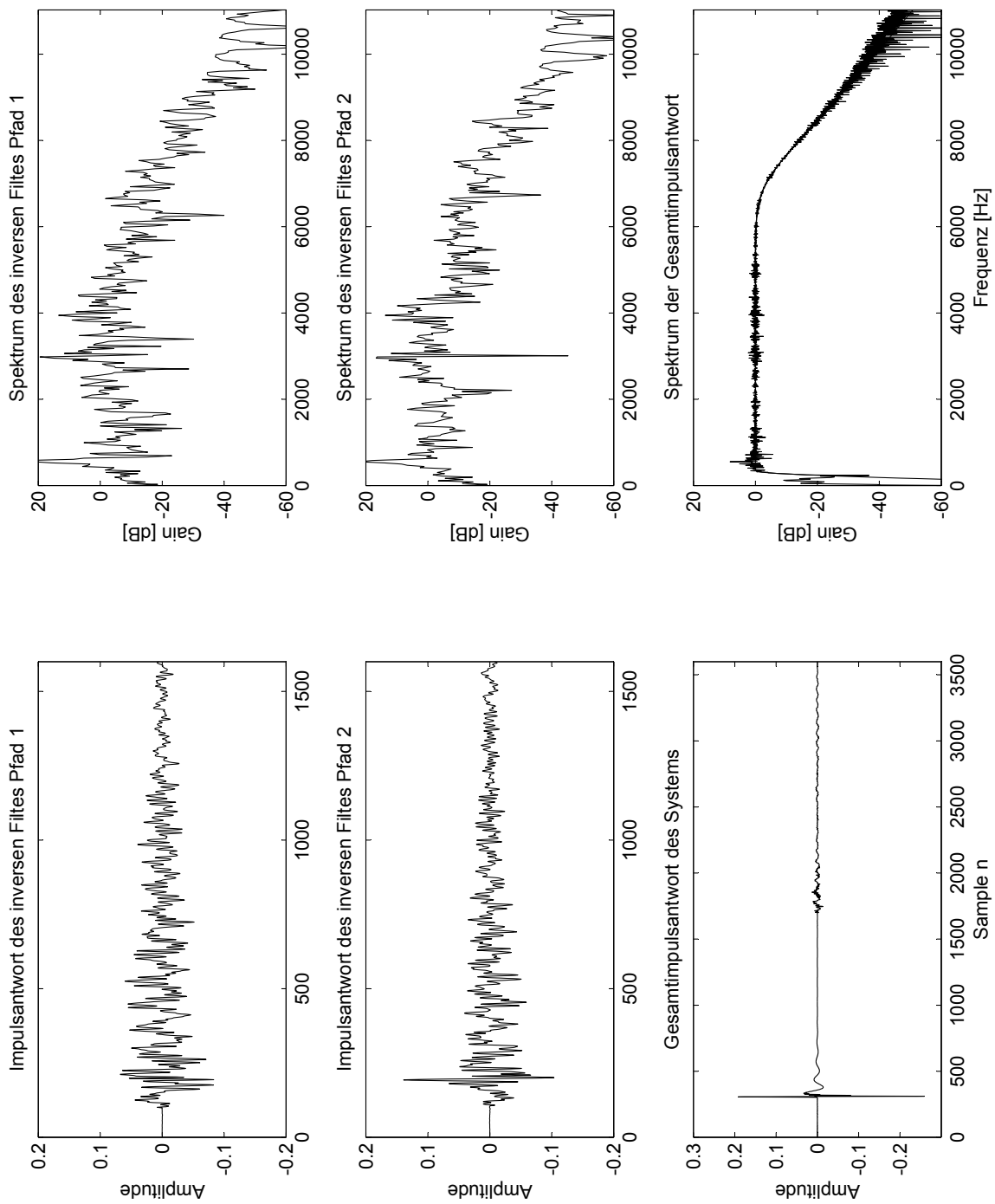


Abbildung 3.14: Simulationsergebnis zweikanalige Inversion: Impulsantwort der beiden inversen Filter, Gesamtimpulsantwort und zugehörige Leistungsspektren. Filterlänge: 1600 Samples.

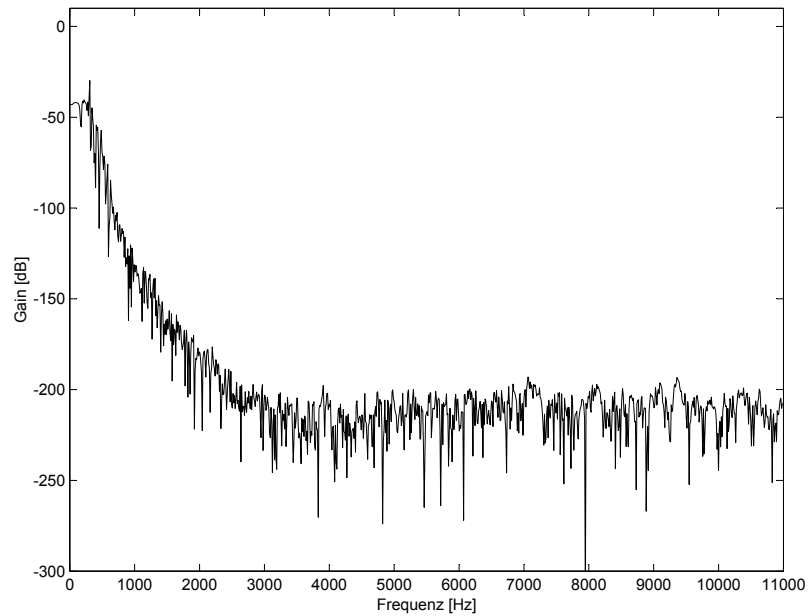


Abbildung 3.15: Zweikanalige Inversion: Fehler zwischen dem ist-Amplitudengang und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 1999 Samples.

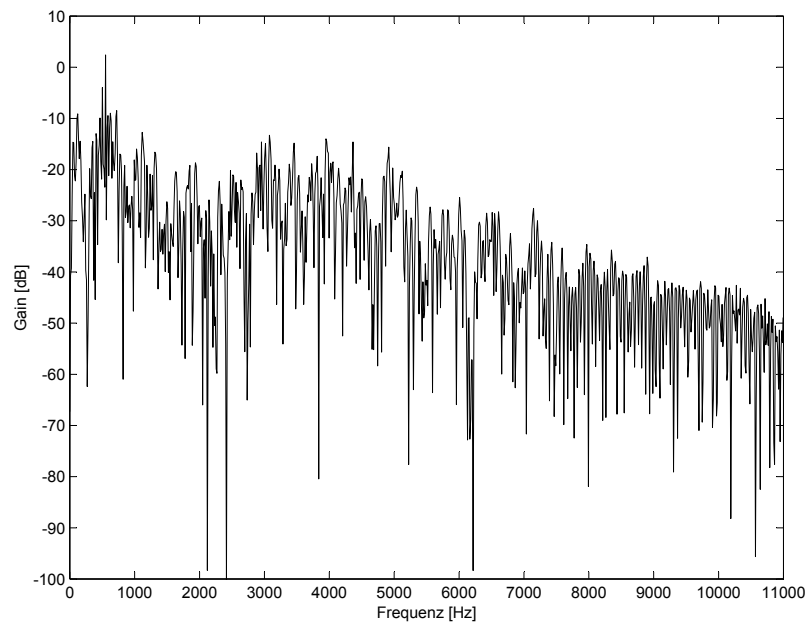


Abbildung 3.16: Zweikanalige Inversion: Fehler zwischen dem ist-Amplitudengang und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 1600 Samples.

decon.m	Raumimpulsantwort durch Deconvolution berechnen: <ol style="list-style-type: none"> 1. Wavefiles einlesen (originales Signal und Mikrophonsignal). 2. Beide Signale in den Frequenzbereich transformieren (FFT). 3. Leistungsdichtespektrum und Kreuzleistungsdichtespektrum nach der Periodogramm-Methode berechnen. 4. Übertragungsfunktion und Raumimpulsantwort berechnen. 5. Raumimpulsantwort kürzen (erste 2000 Samples Herausschneiden).
inverse.m	Matrizen-Inversion: <ol style="list-style-type: none"> 1. Toeplitz-Matrix G aufbauen. 2. $(G^T G)^{-1}$ berechnen (Matrizen-Inversion).
solvesys.m	Gleichungssystem lösen: <ol style="list-style-type: none"> 1. Gewünschte Bandpass-Übertragungsfunktion berechnen. 2. Lösungsvektor bestimmen (Laufzeitverzögerung einbauen). 3. Raumimpulsantwort berechnen.
convol.m	Gesamtübertragungsfunktion berechnen: <ol style="list-style-type: none"> 1. Raumimpulsantwort mit der Filterimpulsantwort falten. 2. Raumimpulsantwort, Filterimpulsantwort und Gesamtimpulsantwort und deren Spektren darstellen.

Tabelle 3.1: Die *m*-Files (MATLAB-Skripts), die die Berechnungen der einkanaligen Inversion ausführen. Sie liegen in *ROOT/sim/singlechannel/*.

liefert (2.9).

Nach der Berechnung in MATLAB wurden die Resultate wieder dargestellt. In Abbildung 3.13 sind, ähnlich wie in Abbildung 3.5 und 3.6, die Impulsantworten der beiden Filter und die Gesamtimpulsantwort des System zu finden, jedoch nicht die Raumimpulsantworten⁷. Wie erwartet wird eine perfekte Inversion erreicht. Die Gesamtimpulsantwort entspricht der Bandpass-Impulsantwort, der mean squared error (MSE) beträgt $3.9 \cdot 10^{-22}$, der nur noch durch die numerische Rechengenauigkeit zustande kommt. Praktisch gesehen ist er Null. Dies zeigt auch Abbildung 3.15. Der Fehler (soll- zu ist-Amplitudengang) fällt sehr schnell unter -100 dB.

Dieses Resultat (Abbildung 3.13) wurde durch gezieltes anpassen der «Vorlaufzeit» der geforderten Gesamtimpulsantwort erreicht, so dass die Rechnung es ein realistisches Resultat liefert, denn der Algorithmus findet praktisch immer eine Lösung, handelt es sich doch um ein normales lineares Gleichungssystem, das aufgelöst wird. So kann beispielsweise eine Impulsantwort gefordert werden, die eigentlich nichtkausale Filter bedingen würden (wenn die «Vorlaufzeit» zu kurz gewählt wird), und daher nicht realisierbar sind. Die Rechnung liefert aber trotzdem ein Resultat, da die gemessenen Werte am Anfang der Raumimpulsantwort nicht *genau* Null sind, was sie theo-

⁷Diese wurden aus Platzgründen weggelassen, sind sie doch in Abbildung 3.2 und 3.2 dargestellt.

retisch wären. Die berechneten Filterimpulsantworten enthalten dann aber unrealistisch grosse Werte, was einer Nichtrealisierbarkeit gleichkommt. Eine grosse «Vorlaufzeit» bringt aber immer eine realisierbare Lösung, denn das System bleibt kausal. Im Hinblick auf die DSP-Implementation musste aber darauf geachtet werden, dass die Amplitudenspektren der Filter bei 11025 Hz mindestens -40 dB Dämpfung haben. So musste ein wenig mit dieser «Vorlaufzeit» experimentiert werden, um eine machbare Lösung zu finden. Eine «Vorlaufzeit» von 300 Samples lieferte schliesslich eine gute Lösung.

Simulation mit reduzierter Filterlänge

Eine Wiederholung der Simulation mit einer Filterlänge von 1600 Samples bringt das Ergebnis dargestellt in Abbildung 3.13. Es fällt gleich auf, wie jetzt das Spektrum des Gesamtsystems etwas «verrauscht» ist. Auch die Gesamtimpulsantwort zeigt ab Sample 1600 wieder kleine «Rauschpakete»⁸. Diese bestehen aus vor allem aus einem tieffrequenten Anteil von ca. 560 Hz. Diese starke Frequenzkomponente ist auch in den Filterimpulsantworten wie auch in den Spektren dieser (ein Peak von $+20$ dB!) und im Spektrum der Gesamtimpulsantwort ersichtlich. Dieses Ergebnis hat die Rechnung gebracht. Es bleibt aber etwas unklar weshalb bei dieser Frequenz eine solche Überhöhung vorhanden ist. Es wird vermutet, dass die Konfiguration der Messanordnung am Ort des Mikrophons bei dieser Frequenz eine Auslöschung der Signale des direkten Pfades bewirkt. Interferenz – ein neues Problem, das bei der zweikanaligen Inversion auftreten kann.

Abbildung 3.16 zeigt wieder die Abweichung zwischen Soll- und Ist-Amplitudengang bei reduzierter Filterlänge. Der Fehler beträgt hohen Frequenzen ca. -50 dB und bei tieferen ca. -30 dB. Im Vergleich zu der einkanaligen Inversion mit reduzierter Filterlänge fällt auf, dass der mittlere Fehler bis 7 kHz ca. und 5 dB bis 10 dB geringer ist. Gleichzeitig ist er praktisch nie grösser als -15 dB. Der Amplitudengang des Gesamtsystems weist hier im Unterschied zur einkanaligen Inversion nie lokale «Löcher» auf. Die Berechnung des MSE liefert folgendes Resultat: $2.689 \cdot 10^{-6}$, also durchaus vergleichbar mit dem MSE des Pfades 2 bei der einkanaligen Inversion.

3.2.4 Vergleich und Diskussion

Es drängt sich nun ein Vergleich der zwei simulierten Inversions-Methoden auf, der die Erwartungen an die Resultate der Echtzeitimplementierung festlegt. Es folgt eine Zusammenfassung der bei den Simulationen erhaltenen Fakten und eine abschliessende Diskussion darüber.

Zusammenfassung der Fakten

	Einkanalige Inv.		Zweikanalige Inv.
	Pfad 1	Pfad 2	beide Pfade
MSE volle Filterlänge	$5.9 \cdot 10^{-6}$	$1.8 \cdot 10^{-6}$	0
MSE reduzierte Filterlänge	$8.2 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$	$2.7 \cdot 10^{-6}$
Fehler volle Filterlänge, tiefe Freq.	-25 dB	-30 dB	0
Fehler volle Filterlänge, hohen Freq.	-65 dB	-60 dB	0
Fehler reduz. Filterlänge, tiefe Freq.	-20 dB	-25 dB	-30 dB
Fehler reduz. Filterlänge, hohen Freq.	-50 dB	-70 dB	-50 dB

Diskussion

Zweikanalige Inversion

Die zweikanalige Inversion funktioniert perfekt, wenn die volle Filterlänge verwendet wird. Dies

⁸Dieses Phänomen wurde schon bei der einkanaligen Inversion beobachtet und erklärt.

decon.m	Raumimpulsantwort durch Deconvolution berechnen: <ol style="list-style-type: none"> 1. Wavefiles des Pfades 1 einlesen (originales Signal und Mikrophonsignal). 2. Beide Signale in den Frequenzbereich transformieren (FFT). 3. Leistungsdichtespektrum und Kreuzleistungsdichtespektrum nach der Periodogramm-Methode berechnen. 4. Übertragungsfunktion und Raumimpulsantwort berechnen. 5. Raumimpulsantwort kürzen (erste 2000 Samples Herausschneiden). 6. Wiederholung der obigen Schritte für Pfad 2.
inverse.m	Matrizen-Inversion: <ol style="list-style-type: none"> 1. Toeplitz-Block-Matrix $[G_1 \ G_2]$ aufbauen. 2. $[G_1 \ G_2]^{-1}$ berechnen (Matrizen-Inversion).
solvesys.m	Gleichungssystem lösen: <ol style="list-style-type: none"> 1. Gewünschte Bandpass-Übertragungsfunktion berechnen. 2. Lösungsvektor bestimmen (Laufzeitverzögerung einbauen). 3. Raumimpulsantwort berechnen.
convol.m	Gesamtübertragungsfunktion berechnen: <ol style="list-style-type: none"> 1. Raumimpulsantworten mit den Filterimpulsantworten falten. 2. Superposition der Faltungsprodukte berechnen (dies ist die Gesamtimpulsantwort). 3. Filterimpulsantworten und Gesamtimpulsantwort und deren Spektren darstellen.

Tabelle 3.2: Die m-Files (MATLAB-Scripts), die die Berechnungen der zweikanaligen Inversion ausführen. Sie liegen in `ROOT/sim/twochannel/`.

konnte von der Theorie her auch erwartet werden, da es die Lösung eines genau determinierten Gleichungssystems handelt. Besitzen die Übertragungsfunktionen der beiden Pfade keine gemeinsame Nullstelle, so ist das Gleichungssystem lösbar (die Matrix G ist invertierbar) und das Ergebnis ist eindeutig [1].

In einer Echtzeitimplementation kann aber nicht davon ausgegangen werden, dass Filter beliebiger Länge eingesetzt werden können. Die verfügbaren Ressourcen legen die Grenzen fest. Werden nun kürzere Filter verwendet, so arbeitet die zweikanalige Inversion auch nicht mehr perfekt. Die Simulationen haben gezeigt, dass dann die einkanalige Inversion der zweikanaligen recht nahe kommt. Das liegt aber auch daran, dass die einkanalige Inversion recht gut gelungen ist, was dadurch zu erklären ist, dass offenbar vor allem der Pfad 2 eine geringe Abweichung zu einem minimalphasigen System hat.

**Echtzeit-
implementation**

Einkanalige Inversion

Die einkanalige Inversion liefert auch ein recht gutes Ergebnis besonders Pfad 2. Ein Vergleich des Resultats mit der vollen Filterlänge mit dem das mit der reduzierten Filterlänge erreicht wurde, zeigt, dass die Zunahme des Fehlers recht gering ausfällt (ca. 5 dB). Bei der zweikanaligen Version ist der Fehler noch einmal um ca. 5 dB geringer. Auch ein Vergleich des MSE zeigt, dass die zweikanalige Inversion zwar besser ist, aber mit geringem Vorsprung.

Interferenz

Dies hat aber damit zu tun, dass die Impulsantworten der inversen Filter bei der zweikanaligen Variante eine starke tieffrequente Komponente bei ca. 560 Hz haben, die sich in der Gesamtimpulsantwort wiederfindet; daher wird auch der Fehler gross. Der Grund für diese starke Komponente könnte, wie bereits erwähnt, Interferenz sein. Man stelle sich das gleiche Experiment mit zwei Lautsprechern und einem Mikrophon in einem schalltoten Raum vor. Dort müssten Frequenzen existieren, die am Ort des Mikrophons zu 100% ausgelöscht werden, ein typisches Interferenzmuster würde im Amplitudengang auftreten. In einer abgeschwächten Form ist dies durchaus auch in einem echten Raum vorzustellen, wo der direkte Pfad Lautsprecher – Mikrophon dominant ist. Um diese Vermutung eingehend zu überprüfen fehlte leider die Zeit.

LSE für MINT

Die Aspekte der Echtzeitimplementation lassen den Wunsch nach einem LSE-Algorithmus für kürzere Filterlängen für die MINT-Methode aufkommen. Beim hier verwendeten Verfahren werden zuerst möglichst lange inverse Filter berechnet, indem möglichst viele Samples der Raumimpulsantwort verwendet werden; dann erst werden die Filterimpulsantworten gekürzt, beispielsweise auf 1600 Samples. Folglich weist die Gesamtimpulsantwort erst nach 1600 Samples einen «Fehler» auf. Der vorhergehende Teil (die ersten 1600 Samples) entspricht der optimalen Gesamtimpulsantwort. Die Methode entspricht also einem «zero forcing», der Fehler wird also gewissermassen nach hinten «aufgeschoben». Wünschenswert wäre eine LSE-Methode, die den Fehler gesamthaft minimiert. Um dieses Problem zu lösen, wurde in [2] von einem Kalman-Filter-Algorithmus ausgegangen, ein rekursiver Algorithmus zur Lösung des least-squares Problems. Aus Zeitgründen musste jedoch auf die Implementation eines solchen Algorithmus verzichtet werden.

Abschliessend kann festgestellt werden, dass ein Filter mit 1600 Koeffizienten eine gute Lösung des Inversions-Problems liefert, sowohl in der einkanaligen wie auch in der zweikanaligen Version.

3.3 DSP-Implementation

Nach den umfangreichen Simulationen galt es nun die Umsetzung in ein Echtzeitsystem vorzunehmen. Die FIR-Filter wurden mit einem DSP ASDP-21062 SHARC der Firma ANALOG DEVICES realisiert. Dieser ist Teil des Labor-DSP-Systems der HSR Rapperswil, ein PCI-Board für den PC.

3.3.1 FIR-Filter-Implementation

Die FIR-Filter wurden mittels der C-Bibliotheksfunktion⁹ `fir()` implementiert. Diese Funktion ist sehr gut optimiert. Eine Implementation in Assembler bringt keinen entscheidenden Vorteil. Dies haben Untersuchungen an einer Assembler-Implementation von Laborassistent A. Rüegg ergeben. Bei einer Samplingrate von 22050 Hz kann mit dem verwendeten Labor-DSP-System¹⁰ ein FIR-Filter mit 800 Koeffizienten implementiert werden. Das erstellte C-Programm `invfir.c`¹¹ wurde mit dem Compiler der Entwicklungsumgebung Release 3.3 von ANALOG DEVICES und mit Hilfe des and der HSR erstellen MATLAB-GUI `comptool` übersetzt.

⁹Aus der mit dem DSP gelieferten C-Library

¹⁰Details: siehe Anhang B «Messequipment»

¹¹File: ROOT/imp/invfir/invfir.c

Beschreibung des C-Programmes

Das Eingangssignal wird auf Kanal 1 eingespielen und Sample um Sample eingelesen. Jedes von D/A-Wandler ausgegebene Sample löst einen Interrupt aus. Zu diesem Zeitpunkt ist auch wieder ein Sample vom A/D-Wandler bereit, das in der dadurch aufgerufenen interrupt service routine verarbeitet wird. Die Zeile

```
sample_out = fir(sample_in, coeffs, state, FIR_LEN);
```

in `invfir.c` berechnet das resultierende Ausgangs-Sample `sample_out`, nachdem das Eingangssample `sample_in` in das FIR-Filter «hineingeschoben» wurde. Die Konstante `FIR_LEN` ist die Länge, `coeffs` die Koeffizienten und `state` der interne Buffer (die Speicherplätze) des Filters.

Implementation der zweikanaligen Inversion

Die zweikanalige Inversion benötigt grundsätzlich zwei FIR-Filter. Dadurch könnte hier die Filterlänge nur halb so lange gemacht werden, also nur 400 Koeffizienten pro Filter, da die Rechenkapazität des DSP aufgeteilt werden muss. Zudem würden zwei Anti-Aliasing-Filter und Anti-Interference-Filter benötigt. Dieses Manko kann umgangen werden, indem die Superposition der beiden Pfade nicht «in der Luft», also am Ort des Mikrophons, sondern erst später rechnerisch mit dem Computer vollzogen wird. Es spielt aus der systemtechnischen Sicht keine Rolle, wo die Superposition stattfindet. In der Simulation wird genau so verfahren. Erst wird die Faltung der Raumimpulsantwort des Pfades 1 mit seinem Filter berechnet, und dann entsprechend für den zweiten Pfad. In einem dritten Schritt wird erst die Gesamtimpulsantwort, die Summe der beiden Pfadimpulsantworten, berechnet.

Dieses Verfahren wurde so auch bei der Messung am Echtzeitsystem angewandt. Die Pfadimpulsantworten wurden separat gemessen und erst dann die Gesamtimpulsantwort durch Addition der beiden Pfadimpulsantworten berechnet. Folgende Vorteile können dadurch erzielt werden:

- Es wird nur ein Anti-Aliasing-Filter und Anti-Interference-Filter benötigt. Im DS-Labor ist ohnehin nur ein Gerät mit zwei einstellbaren Filtern vorhanden. Würden die zwei FIR-Filter gleichzeitig betrieben, müssten die beiden analogen Filter als AAF eingesetzt werden und auf AIF verzichtet werden, was systemtechnisch nicht korrekt wäre.
- Die Filterlänge kann bei 800 Koeffizienten belassen werden. Dadurch ist ein besserer Vergleich der einkanaligen mit der zweikanaligen Inversion zu erreichen.

3.3.2 Vorbereitungen und Download der Filterkoeffizienten

Wie bereits schon erwähnt, müssen die Filterkoeffizienten dezimiert werden. Genauer genommen wird einfach ein Downsampling um den Faktor 2 vorgenommen, d.h. jedes zweite Sample der berechneten Filterimpulsantwort wird einfach weggelassen. Mit der Wahl des Bandpasses als geforderte Gesamtübertragungsfunktion wurde sichergestellt, dass dabei kein Aliasing entsteht (siehe Abschnitt «Rahmenbedingungen»).

Die Filterkoeffizienten müssen gemäss dem C-Library Manual in umgekehrter Reihenfolge im C-Array `coeffs` abgelegt werden. Daher muss der entsprechende Vektor in MATLAB vor dem Download umgedreht werden. Danach werden die Koeffizienten noch um den Faktor 0.5 gewichtet. Damit wird ein Overflow des Zahlenbereiches und ein «Übersteuern» des D/A-Wandlers vermieden.

Schlussendlich können die Koeffizienten auf den DSP heruntergeladen werden. Die geschieht mit folgendem MATLAB-Befehl:

```
sharc('dl_flts', 'coeffs', h1dr);
```

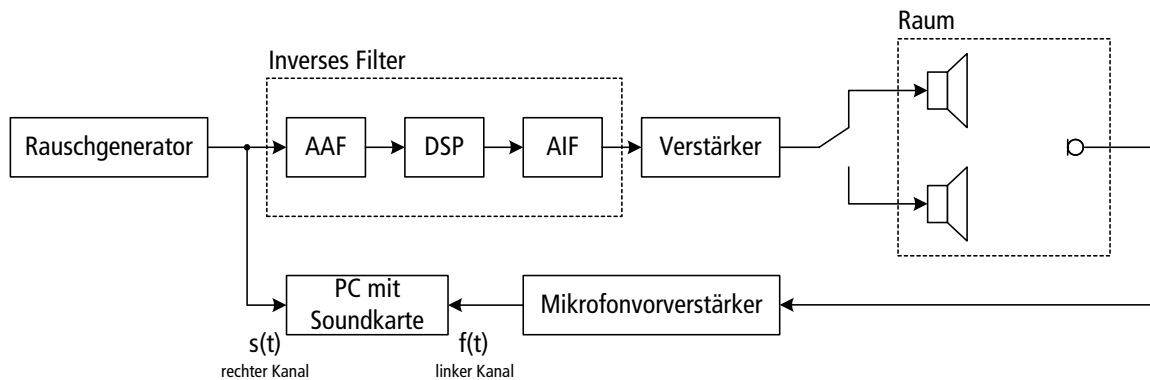


Abbildung 3.17: Aufbau des Echtzeitsystems. Dem DSP sind ein Anti-Aliasing- und ein Anti-Interference-Filter vor- bzw. nachgeschaltet. Der Raum wird wahlweise mit dem linken (Pfad 1) oder mit dem rechten Lautsprecher (Pfad 2) beschallt. Ein Mikrofon nimmt den Schall auf, das Signal wird vorverstärkt und in den linken Kanal der Soundkarte eingespielen. Das originale Rauschsignal wird zudem mit dem rechten Kanal verbunden.

Dabei ist `coeffs` die Name des C-Arrays und `h1dr` die Filterkoeffizienten. Der Datenaustausch zwischen MATLAB und DSP wurde über die an der HSR entwickelten Library `mat1ab.d11` (MEX-File) bewerkstelligt. Diese stellt solche MATLAB-Kommandos wie das obige zu Verfügung.

3.3.3 Systemaufbau

Die FIR-Filter wurden gemäss 2.3 den Lautsprechern vorgeschaltet. Der übrige Systemaufbau entspricht dem der Messungen der Raumimpulsantwort. Natürlich wurde die Anordnung der Lautsprecher und des Mikrophons so belassen, wie bei der Messung der Raumimpulsantwort verwendet wurde. Dem digitalen FIR-Filter, das in einem DSP realisiert wurde, sind ein Anti-Aliasing- und ein Anti-Interference-Filter vor- bzw. nachgeschaltet. Diese sind analoge Tiefpass-Filter 8. Ordnung mit den Grenzfrequenzen 7 kHz (AAF) und 9 kHz (AIF). Da es sich bei den zur Messung verwendeten Rauschsignalen und breitbandigen Signalen handelt, ein Filter zur Vermeidung von Aliasing nötig. Da auch nicht davon ausgegangen werden kann, dass das dem DSP nachfolgende System genügend Tiefpasscharakter besitzt (die halbe Samplingfrequenz beträgt 11025 Hz) um hohen spektralen Anteile, die durch die sample-and-hold Schaltung des D/A-Wandlers entstehen, genügend zu dämpfen, muss ein Anti-Interference-Filter eingesetzt werden.

3.4 Messung am Echtzeitsystem

An dem in obigen Abschnitt beschriebenen Echtzeitsystem wurde die Gesamtimpulsantwort wiederum nach dem gleichen Verfahren, das schon bei der Bestimmung der Raumimpulsantworten verwendet wurde, gemessen. Das Experiment umfasste drei Messungen:

1. Bestimmung der Gesamtimpulsantwort Pfad 1 mit entsprechendem inversen Filter.
2. Bestimmung der Gesamtimpulsantwort Pfad 2 mit entsprechendem inversen Filter.
3. Bestimmung der Gesamtimpulsantwort des zweikanaligen Systems mit inversen Filtern, bei der beide Pfade verwendet wurden.

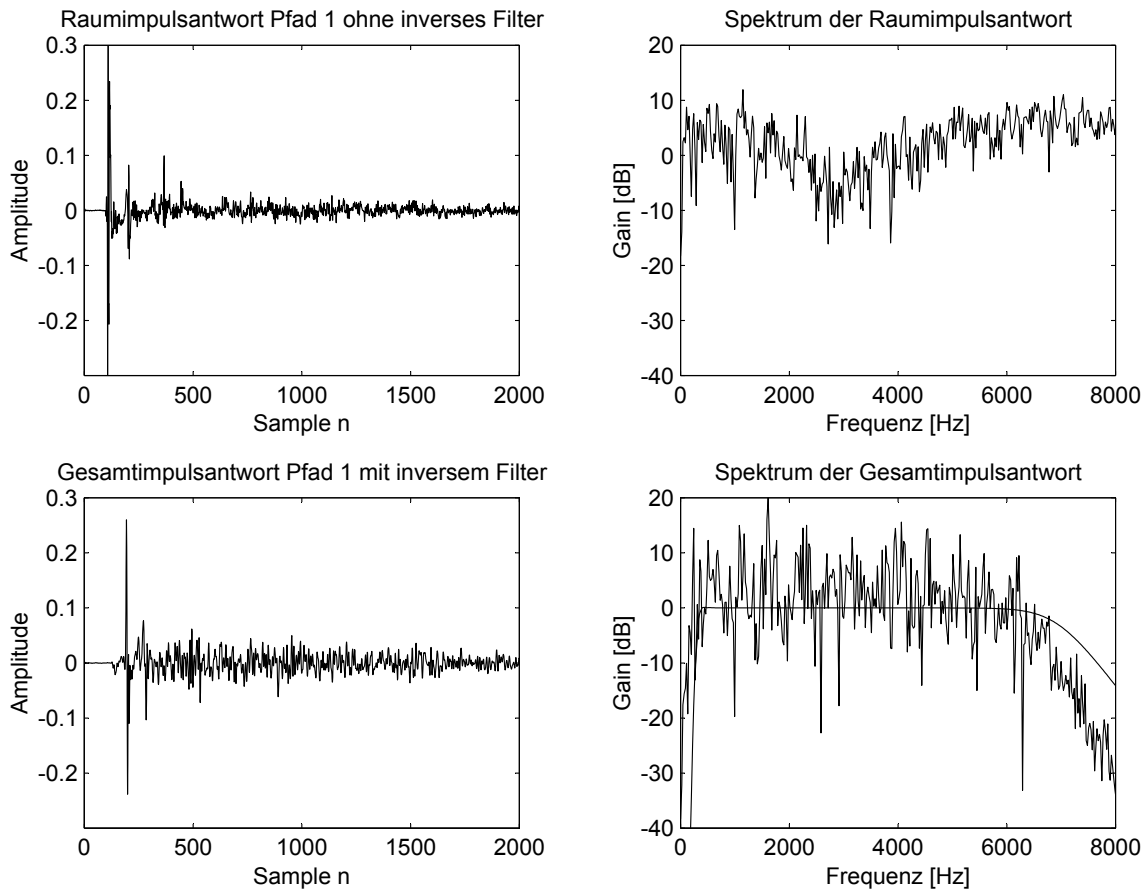


Abbildung 3.18: Messung des Pfades 1 mit inversem Filter.

Beim zweikanaligen System wurden die Pfade 1 und 2 separat gemessen. Die Superposition der bestimmten Pfadimpulsantworten erfolgte in MATLAB. Die Messdaten wurden in MATLAB ausgewertet und dargestellt¹².

3.4.1 Einkanalige Inversion

Abbildung 3.18 und 3.19 zeigen die Messresultate für Pfad 1 bzw. Pfad 2. Bei der Betrachtung der Ergebnisse bei beiden Pfaden fallen sofort vier Dinge auf:

1. Die Absenkung bei 3 kHz, die im Spektrum der Raumimpulsantwort vorhanden ist, ist nach der Entzerrung weitgehend ausgeglichen.
2. Die entzerrten Spektren sind sehr «unruhig» und wellig. Die Unterschiede betragen jetzt ca. 20 dB während es im Originalspektrum nur ca. 10 dB sind.
3. Das «Rauschen» in den Impulsantworten hat zugenommen, vor allem bei Pfad 1. Die originale Raumimpulsantwort wurde hier mit derselben Skalierung gezeichnet, wie die jetzt

¹²FILES: Einkanalige Inversion: R00T/meas/singlechannel/decon.m und R00T/meas/singlechannel/plotres.m, zweikanalige Inversion: R00T/meas/twochannel/decon.m und R00T/meas/twochannel/plotres.m

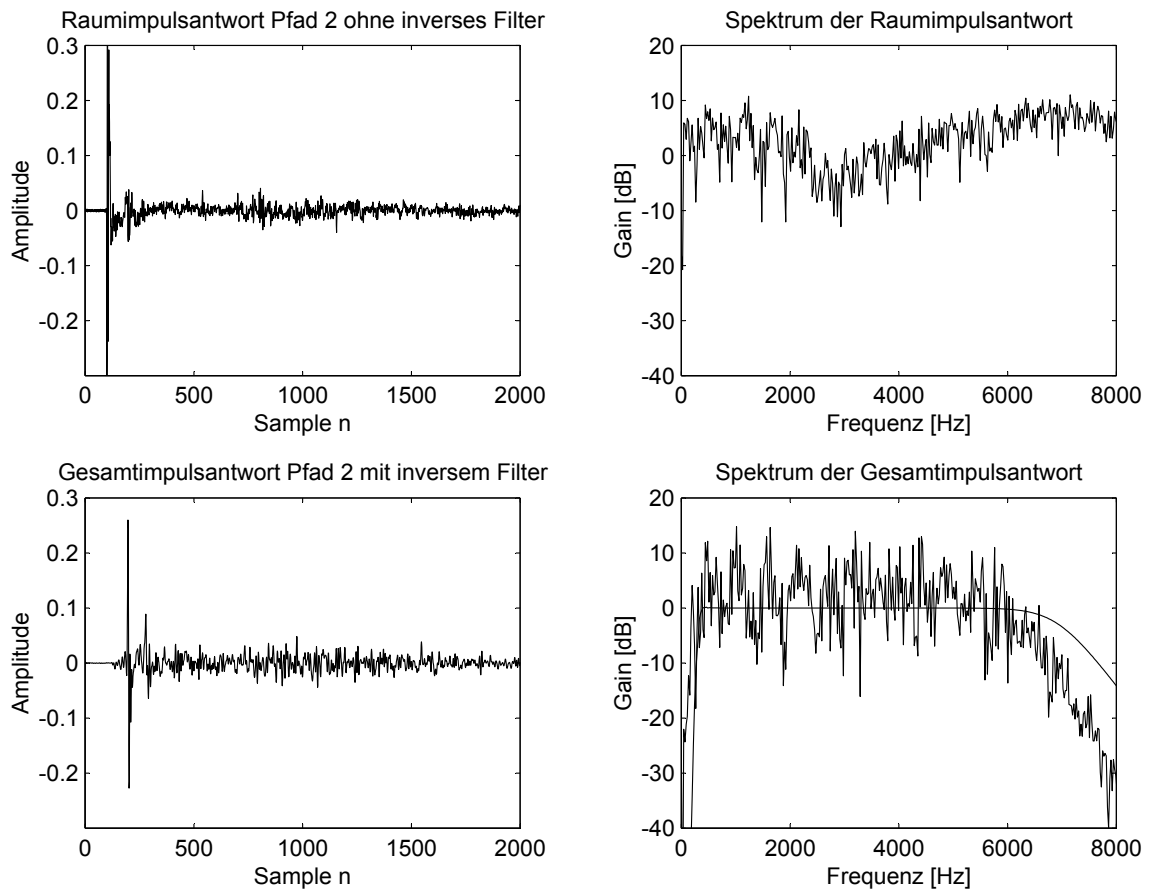


Abbildung 3.19: Messung des Pfades 2 mit inversem Filter.

gemessene. Vergleicht man das «Rauschen» im Verhältnis zum Maximalwert der Impulsantwort¹³, so fällt der Unterschied noch viel deutlicher aus.

- Die gemessenen Spektren fallen viel steiler ab, als die vorgegebene Bandpassübertragungsfunktion. Das AAF und das AIF mit der Grenzfrequenzen bei 7 kHz bzw. 9 kHz haben hier auch einen entscheidenden Einfluss. Dies stört aber nicht, da sich die Betrachtungen auf ein «Nutzband» beschränken sollen, das egalisiert werden soll.

Bei etwas näherer Betrachtung der noch vorhandenen Reflexionen (Peaks) in der gemessenen Impulsantwort stellt man fest, dass einige Peaks abgeschwächt wurden, andere aber sogar verstärkt wurden. Dies ist bei Pfad 2 deutlich zu erkennen. Dieser Peak verursacht – gemäss den symmetrischen Gesetzmässigkeit der Fourier-Theorie – im Spektrum die erwähnte Welligkeit.

3.4.2 Zweikanalige Inversion

Die Resultate der zweikanaligen Inversion zeigen ein etwas anderes Bild (siehe Abbildung 3.20). Das Spektrum der Gesamtimpulsantwort hat im Nutzband einen weitgehend gradlinigen Verlauf. Zwar ist es recht «verrauscht», weist jedoch keine Welligkeit auf wie jene der einkanaligen

¹³Bei der originalen Raumimpulsantwort beträgt der ca. 0.7.

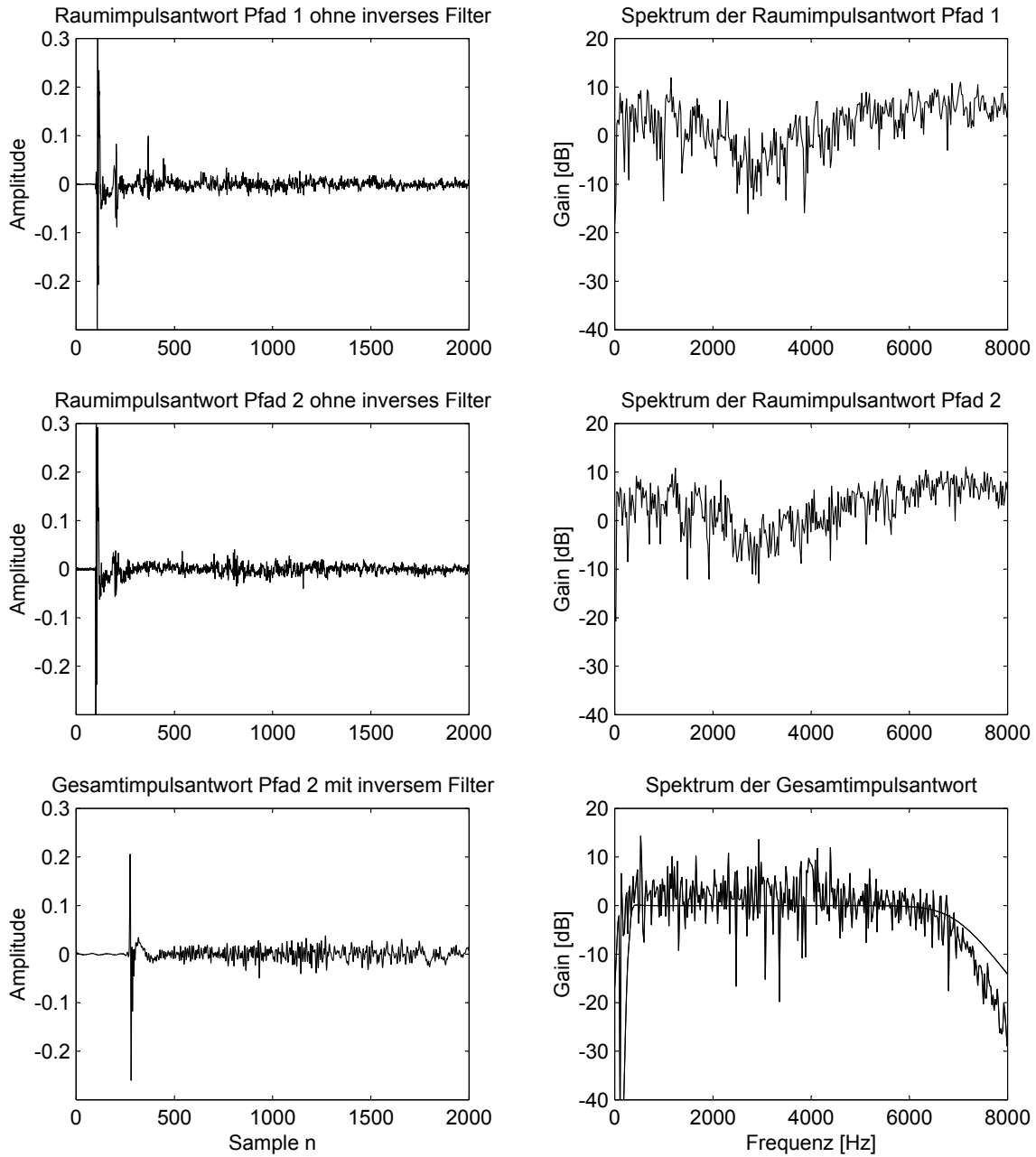


Abbildung 3.20: Messung der zweikanaligen Inversion mit inversem Filter.

Inversion. Die «Löcher» in der Raumimpulsantwort bei 3 kHz sind vollständig kompensiert. Die Gesamtimpulsantwort enthält praktisch keine Reflexionen mehr. Auffallend genau wird auch der typische «Schwinger» der geforderten Bandpass-Impulsantwort reproduziert; einzig das nachfolgende «Rauschen» hat höhere Amplituden als die Raumimpulsantworten. Wieder zu beobachten ist auch das schnelle Abfallen des Spektrums nach 7 kHz, was auf die Anti-Aliasing- und Anti-Interference-Filter zurückzuführen ist.

3.4.3 Vergleich und Diskussion

einkanalige vs. zweikanalige Inversion

Das gute Ergebnis der zweikanaligen Inversion verblüfft, zumal sie Resultate der Messungen am einkanaligen System eher enttäuschten. Wo die Simulation für beide Methoden eine ähnliche Performance voraussagte, klaffen die Ergebnisse der Messungen am Echtzeitsystem erheblich auseinander. Die zweikanalige Inversion liefert ein signifikant besseres Ergebnis. Im Vergleich zur entsprechenden Simulation jedoch ist das «Rauschen» in der Impulsantwort viel stärker vorhanden. Der Mensch empfindet diesen Teil der Impulsantwort als den «Nachhall» der Raumes.

Nachhall

Ohne sich auf die Äste der Physik der Raumakustik hinaus zu wagen kann gesagt werden, dass es bei diesem Nachhall um ein vielfach komplexeres Verhalten geht, als dies bei den ersten Reflexionen der Fall ist. Der Nachhall besteht aus der Überlagerung von dutzenden von Reflexionen mit unterschiedlichen Laufzeiten und Dämpfungsfaktoren an allen möglichen Gegenständen in Raum. Es ist denkbar, dass kleine Veränderungen, z.B. Verschiebung von Gegenständen in diesem Raum, das Muster dieses Nachhalls entscheidend verändern könnten, aber ohne dass der Mensch den Nachhall anders empfinden würde. Überhaupt ist ein Nachhall eher etwas chaotisches als etwas einfach determiniertes.

Weiter liegt die Vermutung nahe, dass die Fehler der verwendeten Messmethode mit weissem Rauschen einen entscheidenden Einfluss auf die Genauigkeit der Messung haben. Starke erste und zweite Reflexionen («early reflections») werden sicherlich gut in der Schätzung wiedergegeben, doch kann ein so komplexes Signal wie ein Nachhall mit einem realen weissen Rauschen, das sicherlich nicht perfekt ist, in einer endlichen Zeit mit hoher Genauigkeit gemessen werden? Vermutlich nicht. Wenn also die Schätzung (Messung) eines solchen Nachhalls nicht sehr genau ist¹⁴, kann von einem auf dieser Schätzung basierend berechneten Filter nicht erwartet werden, dass dieses in einer «real world»-Situation den Nachhall entfernen kann. Ich der Simulation mag das funktionieren, da hier ja immer mit genau der selben Schätzung gearbeitet wird. In [1] und [2] wurde mit fast schalltoten Räumen mit einigen gezielt aufgestellten Reflektoren gearbeitet, die noch weniger Nachhall aufweisen, als der in dieser Arbeit verwendete. In dieser «künstlicheren» Umgebung ist auch mit weniger Problemen mit dem Nachhall zu rechnen. So könnten ev. die besseren Resultate, die dort erzielt wurden, erklärt werden.

Auch stellt sich die Frage nach dem Sinn diesen Nachhall entfernen zu wollen. In einer praktischen Anwendung ist die Forderung häufig nicht die, dass an *einem* speziellen Punkt eine vollständige Entzerrung herrschen soll. Vielmehr ist diese an mehreren Punkten gewünscht. Man denke z.B. an eine Beschallungsanlage für Sprache oder Musik wo mehrere Zuhörer ein möglichst unverzerrtes Signal empfangen möchten. Ein inverses Filter muss in diesem Falle eine für alle Zuhörer im Mittel möglichst gute Entzerrung realisieren. Der Berechnungsaufwand für ein solches Filter ist mit Sicherheit auf ein Vielfaches grösser. Diese Tatsache lässt abschätzen, dass es in einen solchen Fall schwer möglich sein wird den Nachhall in den Berechnungen zu berücksichtigen.

Simulation vs. Echtzeitsystem

In der Simulation konnte eine deutliche Verminderung des «Rauschens» im Spektrum der Raumimpulsantwort und eine Kompensation der Absenkung bei 3 kHz festgestellt werden. Bei der Messung am Echtzeitsystem konnte aber nur letzteres beobachtet werden. Bezüglich der ersteren hat sich eher das Gegenteil manifestiert. Dies lässt den Schluss zu, dass die Simulation keine gute Vor-

¹⁴Um genauere Ergebnisse zu erhalten müsste eine andere Messmethode basierend auf MLS (maximum length sequence) zusammen mit mehreren gemittelten Messungen verwendet werden.

aussage über das Verhalten im Nachhallbereich machen kann. Zwei Gründe könnten dafür verantwortlich sein: Die Messungenauigkeit der Impulsantwortmessung und sie sich schnell ändernden «Nachhallmuster», die von vielen Faktoren abhängen. Sicherlich wären noch weitere Studien um diesen Schluss zu verifizieren.

3.5 Resultate und Schlussfolgerungen

Die Simulation lieferte gute Ergebnisse, die auch so erwartet wurden. Das multiple input/output-inversion-theorem (MINT) konnte verifiziert werden. In der Simulation mit zwei realen Pfaden ist eine exakte Inversion mit voller Filterlänge möglich. Kürzere Filterlängen aber bringen aber wieder «Rauschen» in das Spektrum der Gesamtimpulsantwort. Ein mehr praxisorientierter Algorithmus zur Bestimmung der Filterkoeffizienten, der für eine beliebige Filterlänge Filterkoeffizienten nach dem LSE-Kriterium berechnet, wäre wünschenswert. In [2] wurde ein solcher Algorithmus entwickelt basierend auf der Kalman-Rekursion.

Es konnte auch mittels Simulation gezeigt werden, dass ein einkanaliges System keine exakte Inversion realisieren kann, zumindest nicht für die gemessenen Raumimpulsantworten; obschon die Inversion des Pfades 2 schon recht gut an eine exakte Inversion herankommt – doch nur in der Simulation. Die Ergebnisse der DSP-Implementation zeigten ein anderes Bild: Zwar wurde im *Mittel* eine gute Entzerrung des Amplitudenganges erreicht, doch die lokalen Abweichungen nahmen erheblich zu. So nicht bei der zweikanaligen Inversion. Die Messungen ergaben dort, dass die dominanten Reflexionen («early reflections») fast vollständig entfernt eine somit auch eine gute Entzerrung des Amplitudenganges erreicht wurde. Die lokalen Abweichungen («Rauschen») nahmen nur in geringem Masse zu, sind jedoch signifikant grösser als in der Simulation.

Der einkanaligen wie der zweikanaligen Echtzeit-Inversion gemeinsam ist, dass Sie den «Nachhall» nicht entfernen können, sondern ihn sogar verstärken. Es ist denkbar, dass absichtlich kurz gehaltene Filter ein besseres Resultat liefern könnten als möglichst lange. Diese würden nur die ersten «early reflections» eliminieren und somit eine mehr globale Korrektur des Amplitudenspektrums vornehmen. Die lokalen Abweichungen, die schon im Spektrum der originalen Raumimpulsantwort vorhanden sind, blieben jedoch erhalten, wurden aber auch nicht verstärkt. Die *exakte* Inversion ist eher von theoretischem Interesse, denn sie kann nicht realisiert werden, müsste doch das inverse Filter unendlich langes Filter eingesetzt werden und die Messung der Raumimpulsantwort unendlich genau sein. Dennoch sind die Vorteile einer zweikanaligen Inversion nicht von der Hand zu weisen. Denn auch einer nicht exakten zweikanaligen Inversion kann eine wesentlich bessere Performance attestiert werden.

Inwieweit das menschliche Hörverhalten eine Rolle spielt beim Design inverser Filter wurde in dieser Arbeit nicht beachtet. Das Ohr nimmt die Zeit etwas anders wahr, als dies die Messinstrumente tun. So wird beispielsweise der hintere Teil einer Impulsantwort auch deutlich als Nachhall empfunden und anders gewertet als das direkte Signal mit einigen ersten Reflexionen. Ein messtechnisch exakte Inversion muss für das Ohr nicht zwingend als eine gute Lösung empfunden werden. Diese Lücke zwischen menschlichem Empfinden (Psychoakustik) und der Messtechnik muss genauer untersucht werden und die Ergebnisse sollten in das Filter-Design einfließen.

Ausblick

Es können folgende Ansätze für weitere Studien ausgemacht werden:

- Implementation eines rekursiven Algorithmus zur Bestimmung der Filterkoeffizienten [2].
- Weitere Messungen an dem bestehenden Echtzeitsystem mit kürzeren Filternlängen.
- Einbezug des menschlichen Hörverhaltens in die Entwicklung von inversen Filtern.

- Adaptive Systeme könnten zur inversen Filterung eingesetzt werden. In [6] wurde dies für den einkanaligen Fall realisiert. Es könnte versucht werden eine zweikanalige Inversion mit adaptiven Filtern zu realisieren.

3.6 Schlusswort

3.6.1 Ziel erreicht?

Die theoretischen Aspekte der Raumakustik-Inversion konnten sehr schön anhand der Simulation studiert werden. Dabei konnten wichtige Erkenntnisse gewonnen und das Verständnis der Signaltheorie vertieft werden. Die Umsetzung in ein Echtzeitsystem ist von der systemtechnischen Seite her auch gelungen, doch blieb für die Messungen etwas wenig Zeit. So konnte nur eine «Messrunde» durchgeführt werden; weitere solche wären nach der Analyse der ersten wünschenswert gewesen um die aufgestellten Thesen zu überprüfen.

3.6.2 Persönliche Bemerkungen

Die Studienarbeit war interessant aber etwas theorielastig. Nicht dass die zu studierende Theorie zu umfangreich gewesen wäre, es ist mehr die Tiefe die beschäftigt. Ich hatte die Möglichkeit mich etwas an die komplexe Materie der Raumakustik heranzutasten, die sehr interessant aber auch schwierig ist. An dieser Stelle möchte ich mich für die Betreuung und Unterstützung bei Prof. Dr. Alex Schüeli und dem Laborassistenten A. Rüegg bedanken.

Andy Rohr

Rapperswil, den 12. Dezember 2001

A Listings

A.1 MATLAB Programme

A.1.1 Einkanalige Inversion

Raumimpulsantwort durch Deconvolution berechnen

File: ROOT/sim/singlechannel/decon.m

```
1 clear;
2
3 fc = 44100; % Samplingfrequenz die bei der Messung verwendet wurde
4 fc2 = 22050; % fc / 2
5
6 len = 10 * fc; % Anzahl Samples, die von dem Wavefile eingelesen werden sollen und die deconvolution zu machen (10s)
7 lena = 2000; % Anzahl Samples der zu berücksichtigenden Raumimpulsantwort
8 cutoff = 0; % Anzahl Samples, die von der gemessenen Stossantwort VORNE angeschnitten werden sollen
9
10 % erster Pfad
11
12 % Files einlesen
13 %s = wavread('s-leftpath-44k-01'); %30.11.2001
14 %f = wavread('f-leftpath-44k-01');
15 s = wavread('s-rightpath-44k-01'); % 30.11.2001
16 f = wavread('f-rightpath-44k-01');
17
18 S = fft(s(1:len) .* hanning(len)); % FFT von s berechnen. s wird vorher mit einem Hanningfenster gewichtet
19 F = fft(f(1:len) .* hanning(len)); % FFT von f berechnen. f wird vorher mit einem Hanningfenster gewichtet
20
21 PHIs = (abs(S) .* abs(S)) ./ len; % Leistungsdichtespektrum (Periodogramm-Methode)
22 PHIsf = (conj(S) .* F) ./ len; % Kreuzleistungsdichtespektrum (Periodogramm-Methode)
23
24 G1 = PHIsf ./ PHIs; % Übertragungsfunktion berechnen
25 g1 = real(ifft(G1)); % Stossantwort berechnen
26
27 gla = g1(1+cutoff:len+cutoff); % Stossantwort bescheiden. Dabei werden die vordersten 'cutoff' Samples entfernt
```

Matrizen-Inversion

File: ROOT/sim/singlechannel/inverse.m

```
1 zv = zeros(1,lena-1);
2 G1 = toeplitz([g1a' zv], [g1a(1) zv]); % Toeplitz-Matrix aufbauen
3 G1t = G1';
4
5 Gi = inv(G1t*G1);
6
7 clear G1;
```

Gleichungssystem lösen

File: ROOT/sim/singlechannel/solvesys.m

```
1 start = 250-cutoff; % modeling delay
2
3 fc2 = 22050;
4
5 % Butterworth Bandpass erzeugen
```

```

6 | n = 8; % Ordnung des BP
7 | Wn = [300 7000]/fc2; % Start- und Stop-Frequenz des BP
8 | [b,a] = butter(n,Wn); % Zähler- und Nenner-Polynom des BP
9 | [bp,t] = impz(b,a,2*lena-start-1); % Stossantwort des BP
10 | [BP,F1] = freqz(b,a,lena,'whole',fc); % Übertragungsfunktion des BP
11 |
12 | % Gewünschter Lösungsvektor
13 | %D = zeros(2*lena-1,1);
14 | %D(start) = 1; % Dirac
15 | d_soll = [zeros(start,1); bp;]; % Bandpass
16 |
17 | % Inverse Stossantwort nach der LSE berechnen.
18 | h1a = G1t*d_soll;
19 | h1 = Gi*h1a;

```

Gesamtübertragungsfunktion berechnen und Ergebnisse darstellen

File: ROOT/sim/singlechannel/convol.m

```

1 | %hlen = 2000;
2 | hlen = 1600;
3 | d1 = conv(g1a,h1(1:hlen));
4 |
5 | fc2 = 22050;
6 |
7 | figure;
8 |
9 | G1a = abs(fft(g1a));
10 | lab1 = linspace(0,fc2,length(G1a)/2);
11 |
12 | H1 = abs(fft(h1(1:hlen)));
13 | lab2 = linspace(0,fc2,hlen/2);
14 |
15 | D1 = abs(fft(d1));
16 | lab3 = linspace(0,fc2,length(D1)/2);
17 |
18 | fc4 = 11025;
19 |
20 | subplot(3,2,1);
21 | plot(g1a);
22 | axis([0 length(g1a) -0.5 0.5]);
23 | title('Raumimpulsantwort Pfad 1');
24 | xlabel('Sample n');
25 | ylabel('Amplitude');
26 |
27 | subplot(3,2,3);
28 | plot(h1(1:hlen));
29 | axis([0 hlen -0.2 0.2]);
30 | title('Impulsantwort des inversen Filtes');
31 | xlabel('Sample n');
32 | ylabel('Amplitude');
33 |
34 | subplot(3,2,5);
35 | plot(d1);
36 | axis([0 length(d1)+1 -0.3 0.3]);
37 | title('Gesamtimpulsantwort des Systems');
38 | xlabel('Sample n');
39 | ylabel('Amplitude');
40 |
41 | subplot(3,2,2);
42 | %[BP,F1] = freqz(b,a,length(G1a),'whole',fc); % Übertragungsfunktion des BP
43 | %plot(lab1, 10*log(abs(BP(1:length(G1a)/2))));
44 | %hold on;
45 | plot(lab1, 10*log(G1a(1:length(G1a)/2)));
46 | axis([0 fc4 -40 20]);
47 | title('Spektrum der Raumimpulsantwort');
48 | xlabel('Frequenz [Hz]');
49 | ylabel('Gain [dB]');
50 |
51 | subplot(3,2,2);
52 | %[BP,F1] = freqz(b,a,length(G1a),'whole',fc); % Übertragungsfunktion des BP
53 | %plot(lab1, 10*log(abs(BP(1:length(G1a)/2))));
54 | %hold on;
55 | plot(lab1, 10*log(G1a(1:length(G1a)/2)));

```

```

56 axis([0 fc4 -40 20]);
57 title('Spektrum der Raumimpulsantwort');
58 xlabel('Frequenz [Hz]');
59 ylabel('Gain [dB]');
60
61 subplot(3,2,4);
62 plot(lab2, 10*log(H1(1:length(H1)/2)));
63 axis([0 fc4 -40 20]);
64 title('Spektrum des inversen Filters');
65 xlabel('Frequenz [Hz]');
66 ylabel('Gain [dB]');
67
68 subplot(3,2,6);
69 [BP,F1] = freqz(b,a,length(D1),'whole',fc); % Übertragungsfunktion des BP
70 plot(lab3, 10*log(abs(BP(1:length(D1)/2))));
71 hold on;
72 plot(lab3, 10*log(D1(1:length(D1)/2)));
73 axis([0 fc4 -40 20]);
74 title('Spektrum der Gesamtimpulsantwort');
75 xlabel('Frequenz [Hz]');
76 ylabel('Gain [dB]');

```

Differenz ist- zu soll-Amplitudengang berechnen und darstellen

File: ROOT/sim/singlechannel/convol.m

```

1 %hlen = 2000;
2 hlen = 1600;
3 d1 = conv(g1a,h1(1:hlen));
4
5 fc2 = 22050;
6
7 figure;
8
9 G1a = abs(fft(g1a));
10 lab1 = linspace(0,fc2,length(G1a)/2);
11
12 H1 = abs(fft(h1(1:hlen)));
13 lab2 = linspace(0,fc2,hlen/2);
14
15 D1 = abs(fft(d1));
16 lab3 = linspace(0,fc2,length(D1)/2);
17
18 fc4 = 11025;
19
20 subplot(3,2,1);
21 plot(g1a);
22 axis([0 length(g1a) -0.5 0.5]);
23 title('Raumimpulsantwort Pfad 1');
24 xlabel('Sample n');
25 ylabel('Amplitude');
26
27 subplot(3,2,3);
28 plot(h1(1:hlen));
29 axis([0 hlen -0.2 0.2]);
30 title('Impulsantwort des inversen Filtes');
31 xlabel('Sample n');
32 ylabel('Amplitude');
33
34 subplot(3,2,5);
35 plot(d1);
36 axis([0 length(d1)+1 -0.3 0.3]);
37 title('Gesamtimpulsantwort des Systems');
38 xlabel('Sample n');
39 ylabel('Amplitude');
40
41 subplot(3,2,2);
42 [BP,F1] = freqz(b,a,length(G1a),'whole',fc); % Übertragungsfunktion des BP
43 %plot(lab1, 10*log(abs(BP(1:length(G1a)/2))));
44 %hold on;
45 plot(lab1, 10*log(G1a(1:length(G1a)/2)));
46 axis([0 fc4 -40 20]);
47 title('Spektrum der Raumimpulsantwort');
48 xlabel('Frequenz [Hz]');

```

```

49 ylabel('Gain [dB]');
50
51 subplot(3,2,2);
52 %[BP, F1] = freqz(b, a, length(G1a), 'whole', fc); % Übertragungsfunktion des BP
53 %plot(lab1, 10*log(abs(BP(1:length(G1a)/2))));
54 %hold on;
55 plot(lab1, 10*log(G1a(1:length(G1a)/2)));
56 axis([0 fc4 -40 20]);
57 title('Spektrum der Raumimpulsantwort');
58 xlabel('Frequenz [Hz]');
59 ylabel('Gain [dB]');
60
61 subplot(3,2,4);
62 plot(lab2, 10*log(H1(1:length(H1)/2)));
63 axis([0 fc4 -40 20]);
64 title('Spektrum des inversen Filters');
65 xlabel('Frequenz [Hz]');
66 ylabel('Gain [dB]');
67
68 subplot(3,2,6);
69 [BP, F1] = freqz(b, a, length(D1), 'whole', fc); % Übertragungsfunktion des BP
70 plot(lab3, 10*log(abs(BP(1:length(D1)/2))));
71 hold on;
72 plot(lab3, 10*log(D1(1:length(D1)/2)));
73 axis([0 fc4 -40 20]);
74 title('Spektrum der Gesamtimpulsantwort');
75 xlabel('Frequenz [Hz]');
76 ylabel('Gain [dB]');

```

MSE berechnen

File: ROOT/sim/singlechannel/calcmse.m

```

1 errsum = 0;
2 for i = 1:length(d1)
3     errsum = errsum + (d_soll(i) - d1(i))^2;
4 end
5
6 mse = errsum / length(d1)

```

Filterkoeffizienten für DSP vorbereiten und herunterladen

File: ROOT/sim/singlechannel/dez_ch1.m

```

1 errsum = 0;
2 for i = 1:length(d1)
3     errsum = errsum + (d_soll(i) - d1(i))^2;
4 end
5
6 mse = errsum / length(d1)

```

A.1.2 Zweikanalige Inversion

Raumimpulsantwort durch Deconvolution berechnen

File: ROOT/sim/twochannel/decon.m

```

1 clear;
2
3 fc = 44100; % Samplingfrequenz die bei der Messung verwendet wurde
4 fc2 = 22050; % fc / 2
5
6 len = 10 * fc; % Anzahl Samples, die von dem Wavefile eingelesen werden sollen und die deconvolution zu machen
7 lena = 2000; % Anzahl Samples der zu berücksichtigenden Raumimpulsantwort
8 cutoff = 0; % Anzahl Samples, die von der gemessenen Stossantwort VORNE angeschnitten werden sollen
9
10 % erster Pfad
11
12 % Files einlesen
13 s = wavread('s-leftpath-44k-01'); %30.11.2001

```

```

14 f = wavread('f-leftpath-44k-01');
15 %s = wavread('left-44k-01-s'); % 28.11.2001
16 %f = wavread('left-44k-01-f');
17
18 S = fft(s(1:len) .* hanning(len)); % FFT von s berechnen. s wird vorher mit einem Hanningfenster gewichtet
19 F = fft(f(1:len) .* hanning(len)); % FFT von f berechnen. f wird vorher mit einem Hanningfenster gewichtet
20
21 PHIs = (abs(S) .* abs(S)) ./ len; % Leistungsdichtespektrum (Periodogramm-Methode)
22 PHIsf = (conj(S) .* F) ./ len; % Kreuzleistungsdichtespektrum (Periodogramm-Methode)
23
24 G1 = PHIsf ./ PHIs; % Übertragungsfunktion berechnen
25 g1 = real(ifft(G1)); % Stossantwort berechnen
26
27 %Simulation eines stärker gedämpften Raumes
28 g1a = g1(1+cutoff:lenna+cutoff); % Stossantwort bescheiden. Dabei werden die vordersten 'cutoff' Samples entfernt
29
30
31 % zweiter Pfad
32
33 % Files einlesen
34 s = wavread('s-rightpath-44k-01'); % 30.11.2001
35 f = wavread('f-rightpath-44k-01');
36 %s = wavread('right-44k-01-s'); % 28.11.2001
37 %f = wavread('right-44k-01-f');
38
39 S = fft(s(1:len) .* hanning(len)); % FFT von s berechnen. s wird vorher mit einem Hanningfenster gewichtet
40 F = fft(f(1:len) .* hanning(len)); % FFT von f berechnen. f wird vorher mit einem Hanningfenster gewichtet
41
42 PHIs = (abs(S) .* abs(S)) ./ len; % Leistungsdichtespektrum (Periodogramm-Methode)
43 PHIsf = (conj(S) .* F) ./ len; % Kreuzleistungsdichtespektrum (Periodogramm-Methode)
44
45 G2 = PHIsf ./ PHIs; % Übertragungsfunktion berechnen
46 g2 = real(ifft(G2)); % Stossantwort berechnen
47
48 %Simulation eines stärker gedämpften Raumes
49 g2a = g2(1+cutoff:lenna+cutoff); % Stossantwort bescheiden. Dabei werden die vordersten 'cutoff' Samples entfernt
50
51
52 clear G1 G2 g1 g2 S F PHIs PHIsf s f; % nicht mehr benötigter Speicherplatz freigeben

```

Matrizen-Inversion

File: ROOT/sim/twochannel/inverse.m

```

1 zv = zeros(1,lenna-2);
2
3 G1 = toeplitz([g1a' zv], [g1a(1) zv]);
4 G2 = toeplitz([g2a' zv], [g2a(1) zv]);
5
6 Gi = inv([G1 G2]);
7
8 clear G1 G2;
9
10 %Gi = inv([toeplitz([g1a' zv], [g1a(1) zv]), toeplitz([g2a' zv], [g2a(1) zv])]); % Kurzform

```

Gleichungssystem lösen

File: ROOT/sim/twochannel/solvesys.m

```

1
2 start = 300-cutoff; % modeling delay
3
4 % Butterworth Bandpass erzeugen
5 n = 8; % Ordnung des BP
6 Wn = [300 7000]/22050; % Start- und Stop-Frequenz des BP
7 [b,a] = butter(n,Wn); % Zähler- und Nenner-Polynom des BP
8 [bp,t] = impz(b,a,2*lenna-start-2); % Stossantwort des BP
9 [BP,Fl] = freqz(b,a,2*lenna-2,'whole',fc); % Übertragungsfunktion des BP
10
11 % Gewünschter Lösungsvektor: Dirac
12 %d_soll = zeros(2*lenna-2,1);
13 %d_soll(start) = 1;

```

```

14
15 % Gewünschter Lösungsvektor: Stossantwort des BP verzögert um $start
16 d_soll = [zeros(start,1); bp;]
17
18 h = Gi*d_soll;
19
20 h1 = h(1:lenna-1);
21 h2 = h(lenna:2*(lenna-1));

```

Gesamtübertragungsfunktion berechnen und Ergebnisse darstellen

File: ROOT/sim/twochannel/convol.m

```

1 fc = 44100;
2 fc2 = 22050;
3
4 hlen = 1999;
5 %hlen = 1600;
6 %hlen = 1000;
7 d1 = conv(g1a,h1(1:hlen));
8 d2 = conv(g2a,h2(1:hlen));
9 d = d1 + d2;
10
11 figure;
12
13 G1a = abs(fft(g1a));
14 G2a = abs(fft(g2a));
15 lab1 = linspace(0,fc2,length(G1a)/2);
16
17 H1 = abs(fft(h1(1:hlen)));
18 H2 = abs(fft(h2(1:hlen)));
19 lab2 = linspace(0,fc2,hlen/2);
20
21 D = abs(fft(d));
22 lab3 = linspace(0,fc2,length(D)/2);
23
24 fc4 = 11025;
25
26 subplot(3,2,1);
27 plot(h1(1:hlen));
28 axis([0 hlen -0.2 0.2]);
29 title('Impulsantwort des inversen Filtes Pfad 1');
30 xlabel('Sample n');
31 ylabel('Amplitude');
32
33 subplot(3,2,3);
34 plot(h2(1:hlen));
35 axis([0 hlen -0.2 0.2]);
36 title('Impulsantwort des inversen Filtes Pfad 2');
37 xlabel('Sample n');
38 ylabel('Amplitude');
39
40 subplot(3,2,5);
41 plot(d);
42 axis([0 length(d) -0.3 0.3]);
43 title('Gesamtimpulsantwort des Systems');
44 xlabel('Sample n');
45 ylabel('Amplitude');
46
47
48 subplot(3,2,2);
49 plot(lab2, 10*log(H1(1:length(H1)/2)));
50 axis([0 fc4 -60 20]);
51 title('Spektrum des inversen Filtes Pfad 1');
52 xlabel('Frequenz [Hz]');
53 ylabel('Gain [dB]');
54
55 subplot(3,2,4);
56 plot(lab2, 10*log(H2(1:length(H2)/2)));
57 axis([0 fc4 -60 20]);
58 title('Spektrum des inversen Filtes Pfad 2');
59 xlabel('Frequenz [Hz]');
60 ylabel('Gain [dB]');
61

```

```

62 subplot(3,2,6);
63 [BP,Fl] = freqz(b,a,length(D),'whole',fc); % Übertragungsfunktion des BP
64 plot(lab3, 10*log(abs(BP(1:length(D)/2))));
65 hold on;
66 plot(lab3, 10*log(D(1:length(D)/2)));
67 axis([0 fc4 -60 20]);
68 title('Spektrum der Gesamtimpulsantwort');
69 xlabel('Frequenz [Hz]');
70 ylabel('Gain [dB]');

```

Differenz ist- zu soll-Amplitudengang berechnen und darstellen

File: ROOT/sim/twochannel/convol.m

```

1 fc = 44100;
2 fc2 = 22050;
3
4 hlen = 1999;
5 %hlen = 1600;
6 %hlen = 1000;
7 d1 = conv(g1a,h1(1:hlen));
8 d2 = conv(g2a,h2(1:hlen));
9 d = d1 + d2;
10
11 figure;
12
13 G1a = abs(fft(g1a));
14 G2a = abs(fft(g2a));
15 lab1 = linspace(0,fc2,length(G1a)/2);
16
17 H1 = abs(fft(h1(1:hlen)));
18 H2 = abs(fft(h2(1:hlen)));
19 lab2 = linspace(0,fc2,hlen/2);
20
21 D = abs(fft(d));
22 lab3 = linspace(0,fc2,length(D)/2);
23
24 fc4 = 11025;
25
26 subplot(3,2,1);
27 plot(h1(1:hlen));
28 axis([0 hlen -0.2 0.2]);
29 title('Impulsantwort des inversen Filtes Pfad 1');
30 xlabel('Sample n');
31 ylabel('Amplitude');
32
33 subplot(3,2,3);
34 plot(h2(1:hlen));
35 axis([0 hlen -0.2 0.2]);
36 title('Impulsantwort des inversen Filtes Pfad 2');
37 xlabel('Sample n');
38 ylabel('Amplitude');
39
40 subplot(3,2,5);
41 plot(d);
42 axis([0 length(d) -0.3 0.3]);
43 title('Gesamtimpulsantwort des Systems');
44 xlabel('Sample n');
45 ylabel('Amplitude');
46
47
48 subplot(3,2,2);
49 plot(lab2, 10*log(H1(1:length(H1)/2)));
50 axis([0 fc4 -60 20]);
51 title('Spektrum des inversen Filtes Pfad 1');
52 xlabel('Frequenz [Hz]');
53 ylabel('Gain [dB]');
54
55 subplot(3,2,4);
56 plot(lab2, 10*log(H2(1:length(H2)/2)));
57 axis([0 fc4 -60 20]);
58 title('Spektrum des inversen Filtes Pfad 2');
59 xlabel('Frequenz [Hz]');
60 ylabel('Gain [dB]');

```

```

61 subplot(3,2,6);
62 [BP,F1] = freqz(b,a,length(D),'whole',fc); % Übertragungsfunktion des BP
63 plot(lab3, 10*log(abs(BP(1:length(D)/2))));
64 hold on;
65 plot(lab3, 10*log(D(1:length(D)/2)));
66 axis([0 fc4 -60 20]);
67 title('Spektrum der Gesamtimpulsantwort');
68 xlabel('Frequenz [Hz]');
69 ylabel('Gain [dB]');
70

```

MSE berechnen

File: ROOT/sim/twochannel/calcmse.m

```

1 errsum = 0;
2 for i = 1:length(d)
3     errsum = errsum + (d_soll(i) - d(i))^2;
4 end
5
6 mse = errsum / length(d)

```

Filterkoeffizienten für DSP vorbereiten und herunterladen

File: ROOT/sim/twochannel/dez_ch1.m

```

1 errsum = 0;
2 for i = 1:length(d)
3     errsum = errsum + (d_soll(i) - d(i))^2;
4 end
5
6 mse = errsum / length(d)

```

A.2 DSP-Programm

FIR-Filter

File: ROOT/imp/invfir/invfir.c

```

1
2 /******\
3 * invfir (Blacktip PCI, Toolkit 5.0) *
4 * ----- *
5 * * *
6 * reads channel 1 input (CH1IN) *
7 * outputs FIR-filtered CH1IN to CH1OUT *
8 * * *
9 * Autor: Andy Rohr, Diplomarbeit 2001 *
10 * * *
11 * * *
12 \*****/
13
14
15 /*-----*/
16 /* ADSP-21060 System Register bit definitions */
17 #include <def21060.h> // register definitions
18 #include <21060.h> // architecture
19 #include <signal.h> // interrupt
20 #include <sport.h> // serial port
21 #include <math.h> // fmod
22 #include <filters.h> // fir
23 #include "bitsibb.h" // bitsi board
24 #include "bitsi.h"
25
26 #define SetIOP(addr, val) (* (int *) addr) = (val)
27 #define GetIOP(addr) (* (int *) addr)
28
29 /* Convert a 16 bit Integer value (the 16 left bits in a 32 bit word x)
30 to a 32 bit Integer value */

```

```

31 #define convertleft16to32(result, x) \
32     asm("%0=fext %1 by 16:16 (se);" : "=d" (result) : "d" (x));
33
34 /* Convert a 16 bit Integer value (the 16 right bits in a 32 bit word x)
35    to a 32 bit Integer value */
36 #define convertright16to32(result, x) \
37     asm("%0=fext %1 by 0:16 (se);" : "=d" (result) : "d" (x));
38
39 #define MAX_U_AD 2.75      /* maximum input voltage for A/D converter */
40 #define MAX_U_DA 3.0      /* maximum output voltage on D/A converter */
41
42 #define FIR_LEN 800
43
44 /*-----*/
45 /* pointer for serial port (sport) transfer */
46 int *dms3;
47
48 /* loop counters */
49 int index;
50
51 /* set correction terms for input / output voltages */
52 float NORM = MAX_U_AD / 32767.0; /* normalization of sampled input values */
53 float CORR = 32767.0 / MAX_U_DA; /* denormalization for output values */
54
55 /* A/D, D/A converters */
56 unsigned long AD_IN; /* 32-bit A/D input value (CH1IN | CH2IN) */
57
58 /* matlab interface */
59 //float fc=44100.0; /* sampling frequency */
60 float fc=22050.0; /* sampling frequency */
61
62 int test=0;
63
64 /* fir things */
65 float pm coeffs[FIR_LEN];
66 float state[FIR_LEN+1];
67
68 unsigned int hold_tcount;
69
70 /*-----*/
71 // ISR
72 void spt0_asserted( int sig_num )
73 {
74     int temp; /* temporary variable to separate channels */
75     float sample_in, sample_out; /* input sample values */
76     int out1, out2; /* output sample values */
77
78     /* Convert 16 bit int to 32 bit int */
79     convertleft16to32 (temp, GetIOP(RX0));
80
81     /* normalize input signal from CH1IN */
82     sample_in = NORM * ((float)temp);
83
84     /* Apply FIR Filter to current sample */
85     sample_out = fir(sample_in, coeffs, state, FIR_LEN);
86
87     /* denormalize output */
88     out1 = ((unsigned) (CORR * sample_out) << 16) & 0xffff0000;
89     out2 = 0;
90
91     /* write to output */
92     SetIOP(TX0, out1 | out2);
93 }
94
95 /*-----*/
96
97 void setup_sport0( void )
98 {
99     /* Configure SHARC serial port */
100
101     /* TRANSMIT CONTROL REGISTER */
102     /* An alternate (and more efficient) way of doing this would be to */
103     /* write the 32-bit register all at once with a statement like this: */
104     /* SetIOP(STCTL0, 0x001c00f2); */
105     /* But the following is more descriptive... */
106

```

A Listings

```

107 sport0_iop.txc.mdf = 0; /* multichannel frame delay (MFD) */
108 sport0_iop.txc.schen = 0; /* Tx DMA chaining enable */
109 sport0_iop.txc.sden = 0; /* Tx DMA enable */
110 sport0_iop.txc.lafs = 0; /* Late TFS (alternate) */
111 sport0_iop.txc.ltfs = 1; /* Active low TFS */
112 sport0_iop.txc.ditfs = 0; /* Data independent TFS */
113 sport0_iop.txc.itfs = 0; /* Internally generated TFS */
114 sport0_iop.txc.tfsr = 1; /* TFS Required */
115
116 sport0_iop.txc.ckre = 0; /* Data and FS on clock rising edge */
117 sport0_iop.txc.gclk = 0; /* Enable clock only during transmission*/
118 sport0_iop.txc.iclk = 0; /* Internally generated Tx clock */
119 sport0_iop.txc.pack = 0; /* Unpack 32b words into two 16b tx's */
120
121 sport0_iop.txc.slen = 31; /* Data word length minus one */
122 sport0_iop.txc.sendn = 0; /* Data word endian 1 = LSB first */
123 sport0_iop.txc.dtype = SPORT_DTYPE_RIGHT_JUSTIFY_SIGN_EXTEND;
124 /* Data type specifier */
125 sport0_iop.txc.spen = 1; /* Enable (clear for MC operation) */
126
127 /* RECEIVE CONTROL REGISTER */
128 sport0_iop.rxc.nch = 31; /* multichannel number of channels - 1 */
129 sport0_iop.rxc.mce = 0; /* multichannel enable */
130 sport0_iop.rxc.spl = 0; /* Loop back configure (test) */
131 sport0_iop.rxc.d2dma = 0; /* Enable 2-dimensional DMA array */
132 sport0_iop.rxc.schen = 0; /* Rx DMA chaining enable */
133 sport0_iop.rxc.sden = 0; /* Rx DMA enable */
134 sport0_iop.rxc.lafs = 0; /* Late RFS (alternate) */
135 sport0_iop.rxc.ltfs = 1; /* Active low RFS */
136 sport0_iop.rxc.irfs = 0; /* Internally generated RFS */
137 sport0_iop.rxc.rfsr = 1; /* RFS Required */
138 sport0_iop.rxc.ckre = 0; /* Data and FS on clock rising edge */
139 sport0_iop.rxc.gclk = 0; /* Enable clock only during transmission*/
140 sport0_iop.rxc.iclk = 0; /* Internally generated Rx clock */
141 sport0_iop.rxc.pack = 0; /* Pack two 16b rx's into 32b word */
142
143 sport0_iop.rxc.slen = 31; /* Data word length minus one */
144 sport0_iop.rxc.sendn = 0; /* Data word endian 1 = LSB first */
145 sport0_iop.rxc.dtype = SPORT_DTYPE_RIGHT_JUSTIFY_SIGN_EXTEND;
146 /* Data type specifier */
147 sport0_iop.rxc.spen = 1; /* Enable (clear for MC operation) */
148
149
150 /* Enable sport0 xmit irqs */
151 interruptf(SIG_SPTOI, spt0_asserted);
152
153 }
154
155 /*-----*/
156
157 void init_fir() {
158     int i;
159
160     for (i=0; i<FIR_LEN+1; i++) state[i] = 0;
161
162     for (i=0; i<FIR_LEN; i++) coeffs[i] = 0;
163     coeffs[FIR_LEN-1] = 1;
164
165 }
166
167 /*-----*/
168
169 void setup_bitsibb( void )
170 {
171     BITSIBB_TIMO = BB_TIM(fc);
172     // BITSIBB_DIODAT = 0x5555;
173     // BITSIBB_DIODIR = 0xFF00; /* D15:8] OUT, D[7:0] IN */
174     BITSIBB_CTL = 1; /* Enable just one D/A */
175     BITSIBB_CLKSEL = 0x0844; /* Ch 1-2 Tim 0, Ch 3-4 Tim 1 */
176 }
177
178 /*-----*/
179
180 void main ( void )
181 {

```

```
183 | // int t, msize;
184 |
185 | /* GET BASE LOCATION OF DMS3 (BITSI) */
186 | init_ms_bases(); /* Initialize memory select bases */
187 |
188 | init_fir();
189 |
190 | /* initialize hardware */
191 | setup_bitsibb();
192 | setup_sport0();
193 |
194 | /* do forever */
195 | while(1)
196 | {}
197 | }
```


B Messequipment

Zur Messung der Raumimpulsantworten wurden folgende Geräte verwendet:

Rauschgenerator	Funktionsgenerator Hewlett Packard 33120A
Verstärker für die Lautsprecher	Sony TA-FE530R aus der Heim HiFi-Anlage
Lautsprecher	Canton Ergo 72 DC aus der Heim HiFi-Anlage
Mikrophon	Sennheiser K6 mit Kapsel ME 62 (omnidirektionaler Charakteristik)
Vorverstärker	Kleinmischpult Behringer Eurorack MX 802A
Soundkarte	SoundBlaster Live!

Für die inverse Filterung wurde zusätzlich folgende Geräte verwendet:

Anti-Aliasing und Anti-Interference-Filter dem DSP vor- bzw. nachgeschaltet	Krohn Hite model 3343 Filter
DSP-Board für PC	Bitware Blacktip PCI-Board mit Analog Devices ADSP-21060. Taktfrequenz: 40 MHz

Literaturverzeichnis

- [1] Masato Miyoshi, Yutaka Kaneda *Inverse Filtering of Room Acoustics*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 36, No. 2, February 1988, 145-152
- [2] Alberto Gonzáles, José J. López *Fast Transversal Filters for Deconvolution in Multichannel Sound Reproduction*, IEEE Transactions Speech and Audio Processing, Vol. 9, No. 4, May 2001, 429-440
- [3] Sedki M. Raid *The Deconvolution Problem: An Overview*, Proceedings of the IEEE, Vol. 74, No. 1, January 1986
- [4] Stephen T. Neely, Jont B. Allen *Invertibility of a room impulse response*, J. Acoust. Soc. Am. 66(1), Juli 1979
- [5] Prof. Dr. A. Schüeli, Autographie zur Vorlesung Digitale Signalverarbeitung an der HSR Hochschule für Technik Rapperswil Abteilung E
- [6] Gian A. Bisatz, Markus Gisi, *Inverse Raumakustikfilterung mit DSP*, Diplomarbeit D94-06, HSR Hochschule für Technik Rapperswil, Dezember 1994
- [7] Handbücher zu ANALOG DEVICES ADSP-2106x (*User's Manual, C Tools Manual, C Runtime Library Manual*)

Abbildungsverzeichnis

2.1	Einkanaliges elektroakustisches System	3
2.2	Einkanalige Inversion.	4
2.3	Zweikanaliges elektroakustisches System mit vorgeschalteten Filtern (Pfad 1 und 2)	5
2.4	Zweikanalige Inversion basierend auf dem multiple-input/output theorem (MINT).	6
3.1	Anordnung zur Messung der Impulsantworten verschiedener Pfade in einem Raum.	9
3.2	Auswertungsergebnisse: Raumimpulsantwort und zugehöriges Leistungsspektrum des ersten Pfades	11
3.3	Auswertungsergebnisse: Raumimpulsantwort und zugehöriges Leistungsspektrum des zweiten Pfades	12
3.4	Erste 500 Samples der Impulsantwort und Leistungsspektrum des zu approximierenden Bandpasses	13
3.5	Simulationsergebnis einkanalige Inversion Pfad 1	15
3.6	Simulationsergebnis einkanalige Inversion Pfad 2	16
3.7	Pfad 1: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 2000 Samples.	17
3.8	Pfad 2: Fehler zwischen dem ist- und dem soll-Amplitudengang in Leistungs-dB dargestellt. Die Filterlänge beträgt 2000 Samples.	17
3.9	Simulationsergebnis einkanalige Inversion Pfad 1	18
3.10	Simulationsergebnis einkanalige Inversion Pfad 2	19
3.11	Einkanalige Inversion Pfad 1	20
3.12	Einkanalige Inversion Pfad 2	20
3.13	Simulationsergebnis zweikanalige Inversion	22
3.14	Simulationsergebnis zweikanalige Inversion	23
3.15	Zweikanalige Inversion: Fehler zwischen dem ist-Amplitudengang und dem soll-Amplitudengang	24
3.16	Zweikanalige Inversion: Fehler zwischen dem ist-Amplitudengang und dem soll-Amplitudengang	24
3.17	Aufbau des Echtzeitsystems	30
3.18	Messung des Pfades 1 mit inversem Filter.	31
3.19	Messung des Pfades 2 mit inversem Filter.	32
3.20	Messung der zweikanaligen Inversion mit inversem Filter.	33