

Diplomarbeit

Mikrofonarray mit adaptivem Postfilter zur Sprachsignalentstörung

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs
unter der Leitung von

O.Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Mecklenbräuker
Univ.Ass.Prof. Dipl.-Ing. Dr.techn. Gerhard Doblinger

Institut für Nachrichtentechnik und Hochfrequenztechnik (E 389)

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Peter Fertl
9725081
3300 Amstetten, Raiffeisenstrasse 13

Wien, August 2005

Danksagung

An dieser Stelle möchte ich mich bei jenen Personen bedanken, die mich bei der Erstellung dieser Diplomarbeit tatkräftig unterstützt haben.

Besonderer Dank gilt meinen Eltern und Großeltern sowie meinem Bruder, die mich während meines gesamten Studiums in allen Situationen motiviert, unterstützt und aufgebaut haben.

Ich bedanke mich bei o. Univ.Prof. Dr. Wolfgang Mecklenbräuker für die Bereitstellung des Arbeitsplatzes und die freundliche Unterstützung.

Für die Themenstellung, die Betreuung und die wertvollen, konstruktiven Hinweise zur Durchführung meiner Arbeit danke ich meinem Betreuer Univ.Ass.Prof. Dr. Gerhard Doblinger. Weiters danke ich Dipl.-Ing. Manfred Lenger für die freundliche Zusammenarbeit mit der Firma *Siemens AG*.

Dank gilt auch allen Testpersonen, die sich die Zeit nahmen die unzähligen Audiosignale geduldig zu bewerten.

Für die zahlreichen fruchtbaren Diskussionen zum Thema und die Unterstützung bei anfänglichen Problemen mit Linux danke ich meinem Kollegen Markus Boigner. Ein herzliches Dankeschön gilt außerdem Johannes Maurer und Lukas Haffner als Sprecher bei den Audioaufnahmen und für die Durchsicht der Arbeit.

Schließlich danke ich meiner Freundin Doris Seger für die Motivation, die sie mir am Ende meines Studiums gegeben hat, sowie für ihre Geduld und ihr Verständnis.

Zusammenfassung

Aufgrund der rasanten, technologischen Entwicklung von digitalen Signalprozessoren gewinnen mehrkanalige Algorithmen zur Sprachentstörung mehr und mehr an Bedeutung. Diese Diplomarbeit beschäftigt sich mit dem Einsatz von Mikrofonarrays zur Verbesserung der Sprachqualität in Freisprechkommunikationssystemen. Im Speziellen wird die Methode „Fixer Beamformer mit adaptivem Postfilter“ für unterschiedliche Rauschumgebungen anhand von Simulationen in MATLAB® untersucht.

Nach einer allgemeinen Herleitung der Algorithmenstruktur basierend auf einer Breitband Minimum-Mean-Square-Error (MMSE) Lösung, wird gezeigt, dass eine Aufspaltung dieser Struktur in einen fixen Beamformer mit nachfolgendem, einkanaligen Wiener-Filter (Postfilter) möglich ist. Es werden Modellannahmen getroffen und eine effiziente, echtzeitfähige Implementierung mittels einer FFT-Filterbank vorgestellt. Der Entwurf von Arrays und Beamformer wird diskutiert und ihre Leistungsfähigkeit für verschiedene Rauschumgebungen abgeklärt. Die vielversprechendsten Postfilteralgorithmen der letzten 17 Jahre werden theoretisch hergeleitet und analysiert. Dabei stellt sich heraus, dass vor allem die Schätzung einer effizienten Postfilterfunktion die größten Schwierigkeiten für eine qualitativ hochwertige Sprachentstörung bereitet. Einige Algorithmen erzielen zwar eine hohe Verbesserung des Signal-Rausch-Verhältnisses (bis zu 13 dB), jedoch unter dem Einfluss starker Sprachverzerrungen. Infolge dessen werden Maßnahmen zur Reduktion von Signalverzerrungen und Restrauschen (Musical Noise) vorgeschlagen. Im Zuge der Diplomarbeit wurde eine Datenbank von 540 Testsignalen für die Rauschsituation in einem Büroraum erstellt, die zur Analyse der Algorithmen verwendet wird. Zusätzlich werden alle Algorithmen auch in automotiver Umgebung getestet. Zur Beurteilung der Sprachqualität werden objektive und subjektive Evaluierungsmethoden präsentiert. Im letzten Teil werden die Algorithmen anhand von individuellen Hörtests, objektiven Messungen der Sprachqualität und einer subjektiven Studie, an der 20 Testpersonen teilnahmen, miteinander verglichen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Mehrkanalige Geräuschreduktion	1
1.3	Überblick	3
2	Grundprinzip und allgemeine Eigenschaften	5
2.1	Problemstellung	5
2.2	Annahmen und Bezeichnungen	6
2.2.1	Allgemeine Modellannahmen	6
2.2.2	Modelle für Rauschfelder	7
2.3	Allgemeine Herleitung	8
2.3.1	Mehrkanaliges Wienerfilter im Frequenzbereich	8
2.3.2	Faktorisierung der Wiener Lösung	10
2.3.3	Interpretation	12
2.4	Modellaufbau	13
2.5	Schätzung des Leistungsdichtespektrums	15
2.6	Kohärenzmatrix und Kohärenzfunktion	15
2.6.1	Begriffserklärung und Definition	15
2.6.2	Räumliche Kohärenzfunktionen	16
3	Beamformer	19
3.1	Allgemeines	19
3.1.1	Grundprinzip	19
3.1.2	Analysegrößen	20
3.2	Arrayentwurf	22
3.2.1	Räumliches Aliasing	22
3.2.2	Positionierung der Mikrofone	22
3.2.3	Positionierung der Sprachquelle	23
3.3	Optimaler Beamformer	24
3.3.1	MVDR Beamformer	24
3.3.2	Delay&Sum Beamformer (DSB)	25

3.3.3	Superdirektiver Beamformer (SDB)	25
3.3.4	Vergleich DSB – SDB	26
3.4	Beschränkter Entwurf	27
3.5	Vergleich verschiedener Beamformer-Konfigurationen	28
3.5.1	Unterschiedliche Arraystrukturen	28
3.5.2	DSB und SDB	30
3.5.3	Unterschiedliche Einfallswinkel	30
3.5.4	Spezielle Beamformer	31
4	Postfilter	33
4.1	Zelinski 1988 (ZEL88)	34
4.1.1	Herleitung	34
4.1.2	Interpretation	35
4.2	Simmer 1992 (SIM92)	36
4.2.1	Herleitung	36
4.2.2	Interpretation	37
4.3	McCowan 2003 (GMCC und MCC03)	38
4.3.1	Herleitung	38
4.3.2	Interpretation	40
4.4	APES	41
4.4.1	Herleitung	41
4.4.2	Interpretation	42
4.5	APAB	43
4.5.1	Herleitung	43
4.5.2	Interpretation	45
4.6	Optimierung der Algorithmen	45
4.6.1	Begrenzung der Filterübertragungsfunktion	45
4.6.2	Maßnahmen gegen Signalverzerrungen und Musical Noise	46
4.6.3	Reduzierung der Rechenkomplexität	48
5	Testsignale und Evaluierungsverfahren	51
5.1	Aufnahme der Audiosignale	51
5.1.1	Büroraumaufnahmen	53
5.1.2	Autoaufnahmen	57
5.2	Subjektive Analyse der Sprachqualität	57
5.3	Objektive Messung der Sprachqualität	59
5.3.1	Einleitung	59
5.3.2	Messung des segmentiellen Signal-Rausch-Verhältnisses	59
5.3.3	Direkte Vergleichsmessungen	60
5.3.4	Master-Slave Simulationssystem zur Evaluierung objektiver Messgrößen	62

6	Experimente und Ergebnisse	65
6.1	Parametereinstellungen zur Durchführung der Analysen	65
6.1.1	Auswahl des Eingangssignals	66
6.1.2	Wahl des Glättungsfaktors α	67
6.1.3	Wahl des Beamformers	67
6.1.4	Einsatz von Maßnahmen zur Reduktion von Sprachverzerrungen und Musical Noise	68
6.2	Subjektive Studie zum Vergleich der Algorithmenqualität	68
6.2.1	Algorithmenwahl und Vorselektion	68
6.2.2	Vorgehensweise und Rohdaten	69
6.2.3	Auswertung	70
6.3	Objektive Messungen zum Vergleich der Algorithmenqualität	74
6.3.1	Messeinstellungen und Algorithmenwahl	74
6.3.2	Probleme bei den Messungen	74
6.3.3	Auswertung	75
6.4	Conclusio	80
7	Ausblick	83
A	Akronyme und Abkürzungen	87
B	Ergänzende Definitionen und Herleitungen	89
B.1	Ergänzungen zu Abschnitt 2.3	89
B.1.1	Ableitung nach einem Vektor	89
B.1.2	Beziehung zwischen der Ableitung nach einem Vektor und dem Gradi- entenvektor	90
B.1.3	Ableitung der Fehlerfunktion	91
B.1.4	Sherman-Morrison-Woodbury Formel	92
B.2	Levinson-Durbin Rekursion	92
C	Evaluiierungstabellen	95
D	MATLAB Funktionen	105
D.1	Einleitung und Notation	105
D.2	Beschreibung der Funktionen	106
D.2.1	Test aller untersuchten Algorithmen	106
D.2.2	Objektive Messungen	119
D.2.3	Richtcharakteristik	122
D.2.4	Kohärenzfunktion	124
	Literaturverzeichnis	128

Kapitel 1

Einleitung

1.1 Motivation

Das Aufstreben der Mobilkommunikation und der erhöhte Einsatz innovativer Multimedia-Anwendungen steigert die Nachfrage an Freisprechkommunikationseinrichtungen. Die geforderte, hohe Sprachqualität kann nur durch Unterdrückung von unerwünschten Störsignalen und Nebengeräuschen erreicht werden. Um diese Aufgabe zu erfüllen, sind geeignete Geräuschreduktionsverfahren notwendig. Neben dem Einsatzgebiet solcher Verfahren für Freisprechsysteme in Kraftfahrzeugen sowie Büroumgebungen, finden sie Anwendung in Telekonferenzsystemen, Hörgeräten, Spracherkennungs- und Sprachsteuerungssystemen.

Einkanalige Verfahren sind für diese Aufgabe bereits an ihre Leistungsgrenzen gelangt und ausgereizt. Außerdem verbessern sie weder die Sprachverständlichkeit noch ist eine Reduktion des Nachhalls möglich. Die fortgeschrittene, technologische Entwicklung und Herstellung mittlerweile kostengünstiger digitaler Signalprozessoren, sowie die heutzutage erreichbaren Rechenleistungen, ermöglichen nun auch den Einsatz komplexer, mehrkanaliger Verfahren in Echtzeit.

1.2 Mehrkanalige Geräuschreduktion

Antennengruppen (Antennen-Arrays) haben in den letzten Jahrzehnten zunehmend an Bedeutung gewonnen und werden mittlerweile in einer Vielzahl von Anwendungen eingesetzt. Darunter zählen vor allem Richtstrahlantennen für Funkübertragungen, „Phased Array“ Radar, Hydrofonketten für Sonaranwendungen und Radioteleskope für astronomische Observationen. Diese technischen Errungenschaften haben sicherlich dazu beigetragen das Interesse an Mikrofonarrays – einer Gruppe mehrerer nebeneinander angeordneter Mikrofone – in der akustischen Signalverarbeitung zu fördern und die Entwicklung voranzutreiben.

Das „einfachste“ Mikrofonarray ist in einem ganz anderen Bereich anzutreffen: in der Anatomie. Die Natur hat alle Menschen mit *zwei* Ohren ausgestattet. Ein Grund dafür ist einerseits Redundanz, falls eines der beiden Ohren nicht mehr funktionstüchtig ist. Andererseits ermög-

lichen zwei Ohren dem Menschen die Lokalisation von Schallquellen. Unter Lokalisation versteht man hierbei das Erkennen von Richtung und Entfernung einer Schallquelle. Selbst bei Anwesenheit mehrerer Schallquellen ist das menschliche Gehör in der Lage, die Schallanteile einer einzigen Schallquelle aus dem Gemisch zu extrahieren. Das Gehör erreicht dabei eine Störschallunterdrückung von 9 – 15 dB, wodurch ein Mensch die gewünschte Schallquelle 2- bis 3-mal lauter wahrnimmt als die störenden Umgebungsgeräusche. Diese Fähigkeit ist auch als *Cocktail-Party-Effekt* bekannt. Die dafür notwendige Signalverarbeitung wird von Gehör und Gehirn übernommen. Das Gehör arbeitet in sämtlichen Rauschsituationen relativ robust und optimal. Die in der Praxis eingesetzten Mikrofonarrays weisen diese Fähigkeiten nur bedingt auf.

Mit Hilfe eines Mikrofonarrays lassen sich Sprachsignale, die unterschiedliche Positionen im Raum einnehmen, trennen. Es wird sozusagen eine räumliche Filterung vorgenommen. Dieses Prinzip basiert auf der Ausnutzung der unterschiedlichen Phasenbeziehungen der Eingangssignale und deren geeigneter Kombination, womit eine gewünschte Richtcharakteristik erzielt wird. Die Verwendung eines Mikrofonarrays zur räumlichen Selektion einer bestimmten Sprachquelle im Raum wird als *Beamforming* bezeichnet [1].

Ein Breitband-Mikrofonarray erfüllt die notwendigen Anforderungen bezüglich Reduktion von Nachhall, ausreichender Sprachentstörung und Rauschunterdrückung in effektiver Art und Weise. Solche leistungsfähigen und breitbandigen Beamformer verlangen jedoch eine große Anzahl an Mikrofonen und entsprechend große Mikrofonabstände [2]. Große Arrays sind nur für eine limitierte Anzahl von Anwendungen einsetzbar. Außerdem entsprechen sie nicht den Interessen vieler Hersteller, die vor allem platzsparende und kostengünstige Arrays einsetzen wollen. Daher sind ausgefeilte Algorithmen notwendig, die es ermöglichen die notwendige Mikrofonanzahl und Arraygröße zu reduzieren und trotzdem eine ausreichende Störunterdrückung erzielen. Für diesen Zweck wird der Beamformer oft mit einem zusätzlichen Filter kombiniert. Die räumliche Selektion durch den Beamformer wird somit einer nachfolgenden, zeitlichen Filterung unterworfen. Man spricht dann von Raum-Zeit-Filter. In Folge dessen sind Mikrofonarrays mit geringen Abmessungen realisierbar. Die zeitliche Filterung wird üblicherweise mit einem einkanaligen Wiener-Filter implementiert, welches in der Literatur als *Postfilter* bezeichnet wird. Die Schwierigkeiten liegen in der Schätzung der Postfilterfunktion. Obwohl viele Postfilteralgorithmen das Störrauschen gut unterdrücken, rufen sie zusätzlich Signalverzerrungen hervor. Aus diesem Grund werden in dieser Arbeit mehrere echtzeitfähige, adaptive Postfilteralgorithmen untersucht und ihre Eigenschaften in Bezug auf unterschiedliche Array- und Rauschkonfigurationen verglichen. Zum Einsatz kommen fixe Beamformer, das heißt, die Beamformerkoeffizienten werden nicht adaptiv berechnet, sowie Mikrofongrößen von 4 bis 8 Mikrofonen.

Diese Arbeit soll an die Diplomarbeit meines Vorgängers, Markus Boigner, anschließen, der 2- bis 4-kanalige Algorithmen zur Rauschunterdrückung ohne Einsatz eines Beamformers untersucht hat [3].

1.3 Überblick

In dieser Diplomarbeit werden die Prinzipien und Eigenschaften des Rauschunterdrückungssystems *Beamformer in Kombination mit einem adaptiven Postfilter* erklärt. Weiters werden die gängigsten Postfilteralgorithmen für verschiedene Rauschumgebungen und Arraykonfigurationen analysiert und miteinander verglichen. Die Arbeit gliedert sich in folgende Teile:

- **Kapitel 2.** Dieses Kapitel stellt die Modellannahmen und die Möglichkeiten zur Implementierung eines solchen Systems vor. Außerdem wird eine allgemeine Herleitung präsentiert, sowie der Begriff und die Bedeutung der *Kohärenzfunktion* erläutert.
- **Kapitel 3.** Nach einer kurzen Einführung in die Grundprinzipien der Beamforming-Technik, werden die Eigenschaften des Delay&Sum Beamformer und des superdirektiven Beamformer diskutiert und für verschiedene Arraykonfigurationen getestet.
- **Kapitel 4.** Es werden unterschiedliche, echtzeitfähige Postfilteralgorithmen hergeleitet und diskutiert. Insbesondere werden auch Maßnahmen zur Reduktion von Sprachverzerrungen und der Rechenkomplexität vorgestellt.
- **Kapitel 5.** Dieses Kapitel beschreibt die Aufnahme der Testsignale, die für die Analyse der Algorithmen verwendet wurden. Überdies werden objektive und subjektive Evaluierungsmethoden zur Ermittlung der Sprachqualität besprochen, die einen Vergleich der untersuchten Algorithmen ermöglichen.
- **Kapitel 6.** Experimente und Parametereinstellungen für die Rauschsituation in einem Büroraum und in einem Auto werden vorgestellt. Die Algorithmen werden anhand von objektiven Messungen, wie dem Signal-Rausch-Verhältnis und dem Log-Area Ratio, miteinander verglichen. Eine subjektive Studie, an der 20 Testpersonen teilnahmen, liefert eine zusätzliche Beurteilung, um die vielversprechendsten Algorithmen ausfindig zu machen.
- **Kapitel 7.** Das letzte Kapitel gibt einen Ausblick über die Analysen und Konfigurationen, auf die in dieser Diplomarbeit nicht näher eingegangen werden konnte. Abschließend wird der Ansatz eines zeitvarianten Designs zur Rauschunterdrückung instationärer Rauschfelder präsentiert.
- **Anhang.** Im Anhang befinden sich Ergänzungen und Herleitungen mathematischer Methoden, die in einzelnen Abschnitten verwendet wurden. Anschließend findet man umfangreiche Tabellen aller Ergebnisse der objektiven Analysen sowie die Rohdatentabellen der subjektiven Studie. Zum Abschluss werden die MATLAB® -Funktionen präsentiert, die im Laufe dieser Arbeit erstellt wurden.

Kapitel 2

Grundprinzip und allgemeine Eigenschaften

In diesem Kapitel wird das Grundprinzip und die Eigenschaften der Kombination Beamformer mit adaptivem Postfilter erläutert. Es wird auf die Probleme und die Modellannahmen eingegangen und eine allgemeine Herleitung des vorgestellten Sprachentstörungsalgorithmus präsentiert. Weiters wird der Begriff der räumlichen Kohärenzfunktion erklärt und ihre Funktion als Werkzeug zur Optimierung und Analyse derartiger Mikrofonsysteme deutlich gemacht.

2.1 Problemstellung

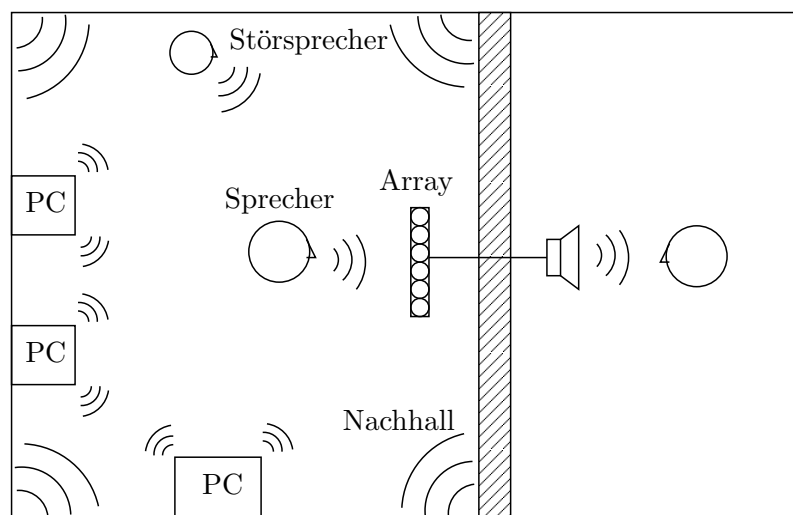


Abbildung 2.1: *Freisprechkommunikationssystem unter Verwendung eines Mikrofonarrays in einem Büroraum.*

In Abbildung 2.1 sehen sie die typische Anwendung eines Freisprechkommunikationssystems mit Mikrofonarray. Aufgabe ist es nun, das durch Umgebungsrauschen gestörte Sprachsignal wieder in ein möglichst rauschfreies und unverzerrtes Sprachsignal zu transformieren. Da-

bei ist der Eingang des angewandten Sprachsignalentstörungsverfahrens mehrkanalig und der Ausgang einkanalig.

Das Hauptproblem bei der Entstörung von Sprachsignalen bringt vor allem die Komplexität der Rauschfelder mit sich. Rauschfelder setzen sich zumeist aus einer Vielzahl unterschiedlicher, oft auch zeitvarianter, Rauschquellen zusammen. Hinzu kommt mitunter auch noch der komplexe Einfluss der Mehrwegeausbreitung. Folgende Faktoren und deren Zusammenspiel tragen zur Komplexität von Rauschfeldern bei:

- Anzahl der vorhandenen Rauschquellen
- Punkt-Rauschquellen/räumlich ausgedehnte Rauschquellen
- bewegte Rauschquellen
- unterschiedliche Rauschleistungen
- Art des Rauschspektrums (schmalbandig/breitbandig)
- Mehrwegeausbreitung/Nachhall

Um eine effiziente Entstörung zu erreichen, muss die aktuelle Beschaffenheit des Rauschfeldes in den verwendeten Algorithmus eingehen. In der Praxis begnügt man sich mit einer Modellannahme für das Rauschfeld. Abweichungen des Rauschmodells vom aktuell vorhandenen Rauschfeld beeinflussen die Leistungsfähigkeit des Algorithmus nachteilig. Modellannahmen für Rauschfelder werden im Abschnitt 2.2.2 näher erläutert.

2.2 Annahmen und Bezeichnungen

2.2.1 Allgemeine Modellannahmen

Sämtliche Prozesse werden als mittelwertfrei angenommen, da die Frequenz Null bei Sprachsignalen keine Rolle spielt. Daher wird im weiteren Verlauf dieser Arbeit nicht mehr zwischen Korrelation und Kovarianz unterschieden.

Allen durchgeführten Berechnungen liegt das Modell der *Fernfeldnäherung* zugrunde. Dabei wird davon ausgegangen, dass sämtliche Signalquellen weit genug entfernt sind, sodass die Signalamplituden an den einzelnen Mikrofonen gleich groß und die einfallenden Wellenfronten eben sind. Diese Näherung ist für viele Anwendungen von Mikrofonarrays gültig. Das Sprachsignal entspricht einer ebenen Welle aus einer bestimmten Richtung. Aufgrund der Arraygeometrie und des Einfallswinkels ergeben sich an den Mikrofoneingängen unterschiedliche Laufzeiten des Sprachsignals. Den Sprachsignalen an den Mikrofoneingängen überlagern sich additiv Störgeräusche (Rauschen) aus der Umgebung.

Für Sprachsignal, Rauschsignal und Eingangssignal werden als kurzzeitstationäre Zufallsprozesse angenommen, wobei das gewünschte Sprachsignal und die Rauschsignale unkorreliert sind. Die Annahme der Unkorreliertheit ist erfüllt, wenn sich die Sprachquelle in der Nähe des Mikrofonarrays (Abstand ca. 60–80 cm) befindet.

2.2.2 Modelle für Rauschfelder

Es werden eine Reihe theoretischer Modelle für Rauschfelder definiert. Um eine einfachere Herleitung und Analyse der betrachteten Verfahren zu ermöglichen, werden die Rauschfelder oft nicht als kurzzeitstationär, sondern als stationär angenommen.

Inkohärentes Rauschfeld

Ein inkohärentes Rauschfeld entsteht durch unkorrelierte Rauschquellen an den Mikrofoneingängen. In der Praxis treten perfekt inkohärente Felder nur selten auf, da eine Schallquelle im Raum im Allgemeinen keine unkorrelierten Signale an den Mikrofonen erzeugen kann. Es handelt sich ja nur um zeitverzögerte Versionen des gleichen Signals. Unkorrelierte Störer treten zum Beispiel durch das elektrische Eigenrauschen der Mikrofone und durch Mikrofontoleranzen auf [4].

Kohärentes Rauschfeld

Eine lokalisierte Rauschquelle erzeugt eine ebene Welle, die sich ungehindert, ohne jegliche Form von Reflexion, Dispersion oder Dissipation, im freien Raum ausbreitet und aus einer bestimmten Richtung auf das Mikrofonarray trifft. In der Praxis treten kohärente Rauschfelder in Open Air Umgebungen auf, wo es keine größere Ausbreitungshindernisse gibt und Effekte, wie Wind oder thermische Turbulenzen, minimal sind [5].

Sphärisch isotropes Rauschfeld

Das sphärisch isotrope Rauschfeld, auch diffuses Rauschfeld genannt, setzt sich aus einer unendlichen Anzahl an unkorrelierten Rauschquellen, welche isotrop an einer Kugel mit unendlichem Radius verteilt sind, zusammen. Das Rauschfeld ergibt sich daher aus der Superposition einer unendlichen Anzahl ebener Wellen aus allen Richtungen. Das diffuse Rauschfeld ist ein gutes Modell für eine große Anzahl unterschiedlicher nachhallender Umgebungen, wie zum Beispiel Büroräume oder Autos [2].

Zylindrisch isotropes Rauschfeld

Einige auftretende Rauschfelder lassen sich besser als zylindrisch isotrop modellieren. Dabei reduziert man das dreidimensionale sphärisch isotrope Rauschmodell auf ein zweidimensionales Modell, das heißt, eine unendliche Anzahl an unkorrelierten Rauschquellen ist isotrop an einem Kreis mit unendlich großem Radius verteilt. Dieses Rauschfeld tritt vor allem dann auf, wenn viele Personen in einem großen Raum sprechen, in dem die Decke und der Boden gut dämpfend sind, oder im freien Raum. Es ist auch als Cocktail-Party Noise bekannt [2].

2.3 Allgemeine Herleitung

2.3.1 Mehrkanaliges Wienerfilter im Frequenzbereich

Das optimale Filter für das Gesamtsystem im Sinne der Minimierung des mittleren quadratischen Fehlers¹ (MMSE) ist ein mehrkanaliges Wienerfilter. Für das bessere Verständnis der hier behandelten Algorithmen wird eine allgemeine Herleitung präsentiert, die in [2] ausführlicher behandelt wird. Abbildung 2.2 zeigt das Filterproblem im Frequenzbereich.

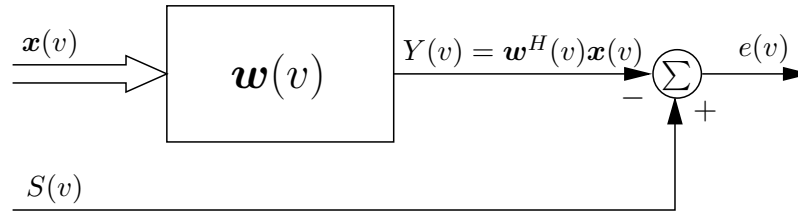


Abbildung 2.2: Wiener-Filter Problem im Frequenzbereich.

Sämtliche Eingangsgrößen werden mittels Kurzzeit-Fouriertransformation (STFT²) in den Frequenzbereich transformiert [6]. Dabei ist v der Frequenzindex, der im Bereich $[0, L - 1]$ liegt. L ist die DFT³- bzw. FFT⁴-Länge. Der zeitliche Frameindex wird zur besseren Lesbarkeit weggelassen. Außerdem werden für eine kompaktere Schreibweise Vektoren und Matrizen⁵ verwendet, deren Dimension K bzw. $K \times K$ ist. Hierbei entspricht K der Anzahl an Mikrofonkanälen. Der Index i bezeichnet den i -ten Mikrofonkanal. Im Folgenden kennzeichnet $*$ die komplexe Konjugation, H die hermitesche Transposition und $\|\cdot\|_2$ die euklidische Norm.

Die STFTs $X_i(v)$ der verrauschten Eingangssignale werden somit zum spektralen Eingangsvektor $\mathbf{x}(v) = [X_1(v), X_2(v), \dots, X_K(v)]^T$ zusammengefasst. Der komplexe, verrauschte Eingangsvektor $\mathbf{x}(v)$ soll nun durch einen optimalen Filtervektor $\mathbf{w}_{\text{opt}}(v)$ in die Spektralkomponente $S(v)$ des gewünschten, skalaren Sprachsignals transformiert werden.

Das Kurzzeitspektrum $Y(v)$ am Filterausgang entspricht für einen beliebigen Filtervektor $\mathbf{w}(v) = [w_1(v), w_2(v), \dots, w_K(v)]^T$:

$$Y(v) = \mathbf{w}^H(v)\mathbf{x}(v). \quad (2.1)$$

Der Fehler $e(v)$ berechnet sich aus der Differenz von Filterausgang $Y(v)$ und dem Kurzzeitspektrum $S(v)$ des gewünschten Sprachsignals zu

$$e(v) = S(v) - Y(v) = S(v) - \mathbf{w}^H(v)\mathbf{x}(v). \quad (2.2)$$

¹engl.: Minimum Mean Square Error

²engl.: Short-Time Fourier Transformation

³engl.: Discrete Fourier Transformation

⁴engl.: Fast Fourier Transformation

⁵Im Folgenden werden Vektoren und Matrizen fett ($\mathbf{x}$ bzw. $\mathbf{X}$) geschrieben.

Unter Verwendung des spektralen Kreuzkorrelationsvektors zwischen dem verrauschten Eingangsvektor $\mathbf{x}(v)$ und der Spektralkomponente $S(v)$ des gewünschten, skalaren Sprachsignals

$$\phi_{\mathbf{x}s} = E \{ \mathbf{x}(v) S^*(v) \}, \quad (2.3)$$

und der spektralen Korrelationsmatrix des verrauschten Eingangsvektors $\mathbf{x}(v)$

$$\Phi_{\mathbf{x}\mathbf{x}} = E \{ \mathbf{x}(v) \mathbf{x}^H(v) \} \quad (2.4)$$

erhält man das Leistungsdichtespektrum (LDS) des Fehlers e

$$\begin{aligned} \phi_{ee}(v) &= E \{ \|e(v)\|_2^2 \} = E \{ (S(v) - \mathbf{w}^H(v) \mathbf{x}(v)) (S^*(v) - \mathbf{x}^H(v) \mathbf{w}(v)) \} \\ &= \phi_{ss}(v) - \mathbf{w}^H(v) \phi_{\mathbf{x}s}(v) - \phi_{\mathbf{x}s}^H(v) \mathbf{w}(v) + \mathbf{w}^H(v) \Phi_{\mathbf{x}\mathbf{x}}(v) \mathbf{w}(v), \end{aligned} \quad (2.5)$$

wobei $E \{ \cdot \}$ der statistische Erwartungsoperator ist. Erst durch die Minimierung der Summe aller $\phi_{ee}(v)$,

$$\sum_{v=0}^{L-1} \left[\phi_{ss}(v) - \mathbf{w}^H(v) \phi_{\mathbf{x}s}(v) - \phi_{\mathbf{x}s}^H(v) \mathbf{w}(v) + \mathbf{w}^H(v) \Phi_{\mathbf{x}\mathbf{x}}(v) \mathbf{w}(v) \right], \quad (2.6)$$

ergibt sich die optimale Lösung. Da das LDS des Fehlers notwendigerweise reell und größer gleich Null ist, wird eine Minimierung der Gesamtleistung Gl. 2.6 durch eine Minimierung der Fehlerleistung $\phi_{ee}(v)$ jedes einzelnen Frequenzpunktes erreicht. Im Folgenden wird zur Vereinfachung der Frequenzindex v weggelassen.

In Gl. 2.5 und Gl. 2.6 sieht man die Abhängigkeit der Fehlerleistung ϕ_{ee} eines Frequenzpunktes vom Filtervektor $\mathbf{w}(v)$. Den optimalen Vektor $\mathbf{w}$ der das Minimum des quadratischen Fehlers bestimmt, erhält man, indem man den Gradienten von ϕ_{ee} gleich dem Nullvektor setzt [7],

$$\nabla_{\mathbf{w}} (\phi_{ee}) = 2 \frac{\partial \phi_{ee}}{\partial \mathbf{w}^*} = -2 \phi_{\mathbf{x}s} + 2 \Phi_{\mathbf{x}\mathbf{x}} \mathbf{w} \stackrel{!}{=} \mathbf{0}. \quad (2.7)$$

Eine genaue Berechnung des Gradienten von ϕ_{ee} kann im Anhang B.1 nachgelesen werden. Durch Umformung erhält man die Wiener-Hopf Gleichung in Matrizenschreibweise

$$\Phi_{\mathbf{x}\mathbf{x}} \mathbf{w}_{\text{opt}} = \phi_{\mathbf{x}s}. \quad (2.8)$$

Zu beachten ist, dass bei der Lösung der typischen Wiener-Hopf Gleichung von stationären Prozessen ausgegangen wird, was zu einem linearen, zeitinvarianten Filter führt. In dieser Herleitung wurden jedoch kurzzeitstationäre Prozesse betrachtet, daher wird in der Praxis

ein adaptives Filter verwendet. Falls die spektrale Korrelationsmatrix Φ_{xx} des verrauschten Eingangsvektors nicht singular ist, kann aus Gl. 2.8 der optimale Filtervektor

$$\mathbf{w}_{\text{opt}} = \Phi_{xx}^{-1} \phi_{xs} \quad (2.9)$$

bestimmt werden.

2.3.2 Faktorisierung der Wiener Lösung

Die mehrkanalige Wienerfilter Lösung aus Gl. 2.9 kann in einen Minimum Variance Distortionless Response (MVDR) Beamformer gefolgt von einem einkanaligen Wienerfilter zerlegt werden [2]. Der MVDR Beamformer minimiert die Ausgangsleistung mit der Nebenbedingung einer unverzerrten Wiedergabe des Signals aus Wunschrichtung. Abschnitt 3.3 zeigt eine ausführliche Untersuchung des MVDR Beamformers.

Bezogen auf unsere Modellannahmen aus Abschnitt 2.2.1 kommt es zu unterschiedlichen Laufzeitverzögerungen des gewünschten, skalaren Sprachsignals s an den Mikrofoneingängen. Diese Laufzeitverzögerungen werden im Frequenzbereich als Phasenverschiebungen im Steuervektor⁶ $\mathbf{d} = [d_1, d_2, \dots, d_K]^T$ zusammengefasst (für eine detaillierte Definition des Steuervektors, siehe Abschnitt 3.1.1). Zusätzlich wird additives Rauschen überlagert. Der spektrale Rauschvektor $\mathbf{n} = [N_1, N_2, \dots, N_K]^T$ setzt sich dabei aus den STFTs $N_i(v)$ der Rauschsignale an den Eingängen i des Mikrofonarrays zusammen. In Folge dessen ergibt sich der verrauschte Eingangsvektor

$$\mathbf{x} = S\mathbf{d} + \mathbf{n}. \quad (2.10)$$

Unter der Annahme, dass Sprachsignal und Rauschen unkorreliert sind, $\phi_{ns} = E\{\mathbf{n}S^*\} = \mathbf{0}$, berechnet sich der spektrale Kreuzkorrelationsvektor zwischen dem verrauschten Eingangsvektor und der Spektralkomponente des gewünschten Sprachsignals zu

$$\phi_{xs} = E\{\mathbf{x}S^*\} = E\{(S\mathbf{d} + \mathbf{n})S^*\} = \phi_{ss}\mathbf{d} + \underbrace{\phi_{ns}}_{=\mathbf{0}} = \phi_{ss}\mathbf{d} \quad (2.11)$$

und die spektrale Korrelationsmatrix des verrauschten Eingangsvektors zu

$$\begin{aligned} \Phi_{xx} &= E\{(S\mathbf{d} + \mathbf{n})(S^*\mathbf{d}^H + \mathbf{n}^H)\} = \phi_{ss}\mathbf{d}\mathbf{d}^H + \underbrace{\phi_{ns}}_{=\mathbf{0}}\mathbf{d}^H + \mathbf{d}\underbrace{\phi_{ns}^H}_{=\mathbf{0}} + \Phi_{nn} \\ &= \phi_{ss}\mathbf{d}\mathbf{d}^H + \Phi_{nn}. \end{aligned} \quad (2.12)$$

⁶Ergänzend ist zu sagen, dass der hier verwendete Steuervektor in dieser allgemeinen Herleitung nicht auf das Modell der Fernfeldnäherung beschränkt ist und daher jedes Element d_i ganz allgemein den akustischen Pfad von der gewünschten Sprachquelle zum jeweiligen Mikrofon i beschreiben kann.

Setzt man Gl. 2.11 und Gl. 2.12 in die mehrkanalige Wiener Lösung Gl. 2.9 ein, dann kann der optimale Filtervektor zu

$$\mathbf{w}_{\text{opt}} = \Phi_{\mathbf{x}\mathbf{x}}^{-1} \phi_{ss} \mathbf{d} = [\phi_{ss} \mathbf{d} \mathbf{d}^H + \Phi_{\mathbf{n}\mathbf{n}}]^{-1} \phi_{ss} \mathbf{d} \quad (2.13)$$

umgeformt werden. Unter Anwendung der Sherman-Morrison-Woodbury Formel⁷ (siehe dazu Anhang B.1.4 oder [7])

$$[\mathbf{A}^{-1} + \mathbf{B} \mathbf{C}^{-1} \mathbf{B}^H]^{-1} \equiv \mathbf{A} - \mathbf{A} \mathbf{B} (\mathbf{C} + \mathbf{B}^H \mathbf{A} \mathbf{B})^{-1} \mathbf{B}^H \mathbf{A} \quad (2.14)$$

kann das mehrkanalige Wiener Filter nun faktorisiert werden. Durch Substitution von

$$\mathbf{A} = \Phi_{\mathbf{n}\mathbf{n}}^{-1}, \quad \mathbf{B} = \sqrt{\phi_{ss}} \mathbf{d}, \quad \text{und} \quad \mathbf{C} = 1 \quad (2.15)$$

in Gl. 2.14 folgt

$$\begin{aligned} \mathbf{w}_{\text{opt}} &= \left[\Phi_{\mathbf{n}\mathbf{n}}^{-1} - \frac{\phi_{ss} \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d} \mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1}}{1 + \phi_{ss} \mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}} \right] \phi_{ss} \mathbf{d} \\ &= \left[1 - \frac{\phi_{ss} \mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}{1 + \phi_{ss} \mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}} \right] \phi_{ss} \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d} \\ &= \left[\frac{\phi_{ss}}{1 + \phi_{ss} \mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}} \right] \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d} \\ &= \left[\frac{\phi_{ss}}{\phi_{ss} + (\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d})^{-1}} \right] \underbrace{\frac{\Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}{\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}}_{\mathbf{b}_{\text{mvdr}}} \end{aligned} \quad (2.16)$$

Berücksichtigt man die Annahme, dass $\Phi_{\mathbf{n}\mathbf{n}}$ positiv definit ist und dadurch der hermitesche (und positive!) Ausdruck $\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}} \mathbf{d}$ skalar und reell, dann zeigt Gl. 2.16 die Zerlegung des mehrkanaligen Wienerfilters in das Produkt eines Filtervektors eines MVDR Beamformers $\mathbf{b}_{\text{mvdr}}$ und einem reellen, skalaren Faktor.

Berechnet man das LDS des gewünschten Sprachsignals,

$$\begin{aligned} \phi_{s_o s_o} &= E \{ (\mathbf{b}_{\text{mvdr}}^H \mathbf{S} \mathbf{d}) (\mathbf{S}^* \mathbf{d}^H \mathbf{b}_{\text{mvdr}}) \} \\ &= \phi_{ss} \mathbf{b}_{\text{mvdr}}^H \mathbf{d} \mathbf{d}^H \mathbf{b}_{\text{mvdr}} = \phi_{ss} \left| \frac{\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}{\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}} \right|^2 = \phi_{ss}, \end{aligned} \quad (2.17)$$

und das LDS des Rauschens,

$$\begin{aligned} \phi_{n_o n_o} &= E \{ (\mathbf{b}_{\text{mvdr}}^H \mathbf{n}) (\mathbf{n}^H \mathbf{b}_{\text{mvdr}}) \} \\ &= \mathbf{b}_{\text{mvdr}}^H \Phi_{\mathbf{n}\mathbf{n}} \mathbf{b}_{\text{mvdr}} = \frac{\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}{(\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d})^2} = \frac{1}{\mathbf{d}^H \Phi_{\mathbf{n}\mathbf{n}}^{-1} \mathbf{d}}, \end{aligned} \quad (2.18)$$

⁷In der Literatur auch als „Matrix Inversion Lemma“ bekannt.

am Ausgang des MVDR Beamformers, dann erkennt man, dass der reelle, skalare Faktor einem einkanaligen Wienerfilter H_{post} am Ausgang des MVDR Beamformers entspricht. In Gl. 2.17 sieht man außerdem, dass der MVDR Beamformer das gewünschte Sprachsignal wirklich unverzerrt lässt. Damit kann Gl. 2.16 letztendlich zu

$$\mathbf{w}_{\text{opt}} = \underbrace{\left[\frac{\phi_{s_o s_o}}{\phi_{s_o s_o} + \phi_{n_o n_o}} \right]}_{H_{\text{post}}} \underbrace{\frac{\Phi_{nn}^{-1} \mathbf{d}}{\mathbf{d}^H \Phi_{nn}^{-1} \mathbf{d}}}_{\mathbf{b}_{\text{mvdr}}}. \quad (2.19)$$

umgeformt werden, bestehend aus dem komplexen Filtervektor des MVDR Beamformers

$$\mathbf{b}_{\text{mvdr}}(v) = \frac{\Phi_{nn}^{-1}(v) \mathbf{d}(v)}{\mathbf{d}^H(v) \Phi_{nn}^{-1}(v) \mathbf{d}(v)} \quad (2.20)$$

und dem einkanaligen Postfilter

$$H_{\text{post}}(v) = \frac{\phi_{s_o s_o}(v)}{\phi_{s_o s_o}(v) + \phi_{n_o n_o}(v)}. \quad (2.21)$$

Mit Hilfe des berechneten Kurzzeitspektrums am Ausgang des Beamformers

$$Y_b(v) = \mathbf{b}_{\text{mvdr}}^H(v) \mathbf{x}(v) \quad (2.22)$$

kann die Spektralkomponente des Ausgangssignals des faktorisierten MMSE Filters gemäß

$$Y(v) = Y_b(v) H_{\text{post}}(v) \quad (2.23)$$

berechnet werden.

2.3.3 Interpretation

Die Übertragungsfunktion des Postfilters leitet sich aus den Leistungsdichtespektren von Sprach- und Rauschsignal am Ausgang des Beamformers ab. Setzt man das Ausgangs-Signal-Rausch-Verhältnis (SNR^8) des MVDR Beamformers $\text{SNR}_{\text{out}} = \frac{\phi_{s_o s_o}}{\phi_{n_o n_o}}$ in Gl. 2.21 ein, erhält man

$$H_{\text{post}}(v) = \frac{\text{SNR}_{\text{out}}(v)}{1 + \text{SNR}_{\text{out}}(v)} \quad (2.24)$$

für jeden Frequenzindex v . Das Postfilter ist also vom SNR des MVDR Beamformerausgangs abhängig. Für große SNR_{out} Werte strebt der Postfilterterm gegen 1, für kleine Werte gegen Null. Das Postfilter leistet also nur dann einen zusätzlichen Beitrag zur Verbesserung des SNR, wenn das Ausgangs-SNR des Beamformers klein ist. Das bedeutet aber auch, dass bereits der Beamformer eine entsprechende Erhöhung des SNR erzielen muss. Als Konsequenz kann man

⁸engl.: Signal-to-Noise Ratio

daraus schließen, dass das Postfilter nur effizient arbeitet, wenn der Beamformer bereits eine ausreichende Rauschunterdrückung vollbracht hat.

2.4 Modellaufbau

In Abbildung 2.3 sieht man ein allgemeines Blockschaltbild für das betrachtete Sprachsignal-entstörungsverfahren im Frequenzbereich.

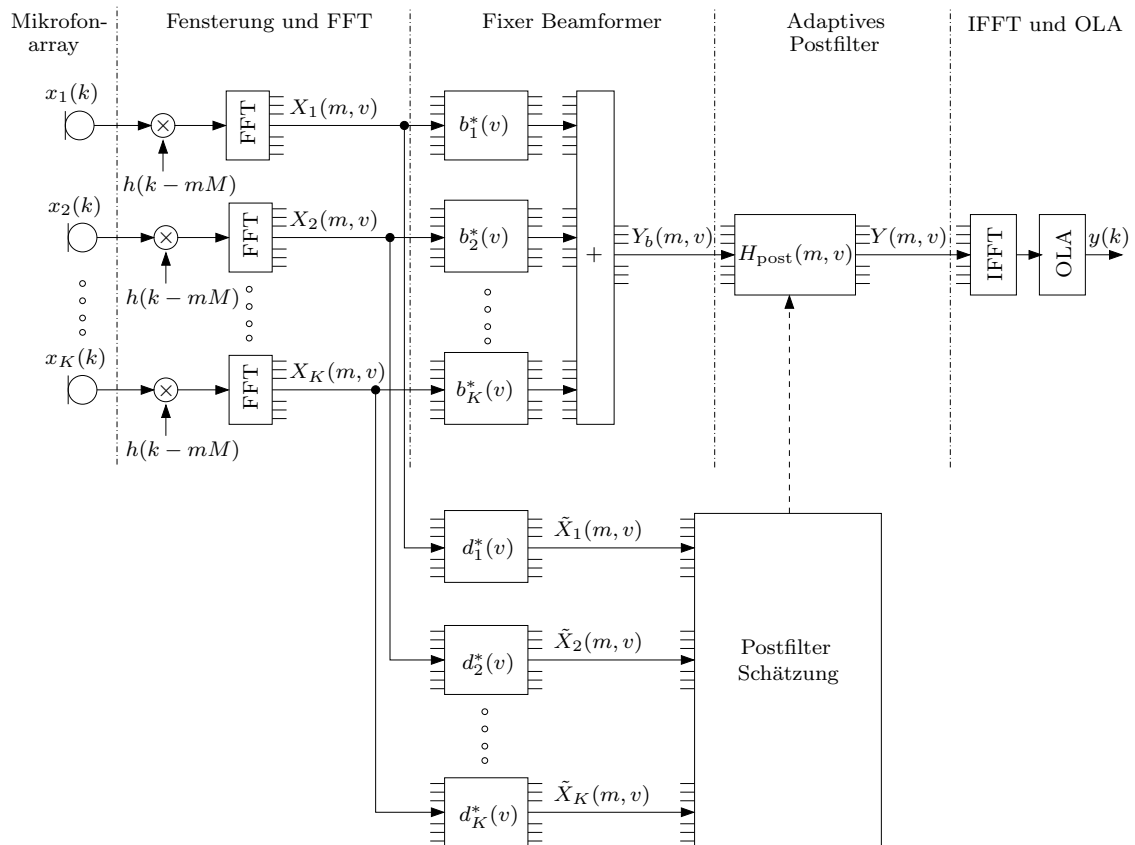


Abbildung 2.3: Block-Realisierung mittels FFT-Filterbank und OLA ($k \dots$ Zeitindex, $m \dots$ Frameindex, $v \dots$ Frequenzindex).

Die Realisierung im Frequenzbereich erfolgt mittels einer FFT-Filterbank und anschließender „Overlap-Add“ Methode (OLA) [6]. Die Analyseseite der FFT-Filterbank beruht auf der STFT. Vor der Transformation wird das Zeitsignal mit einem gleitenden Fenster $h(k - mM)$ multipliziert, wobei der Parameter k dem Zeitindex und m dem Frameindex entspricht. Der ganzzahlige Dezimationsfaktor M legt die Anzahl an Abtastwerte fest, um die das Zeitfenster zur nächsten Position weitergeschoben wird. Die FFT (Länge L) wird daher nur alle M Abtastwerte berechnet. In den Experimenten wurde hierbei ein normiertes Hann Fenster der Länge L verwendet. Der Grad an Überlappung bestimmt die Taktratenreduktion. Eine vierfache Überabtastung ($L/M \geq 4$) ist für Sprachsignale ausreichend, um die Eigenschaft einer nahezu perfekten Rekonstruktion durch Analyse- und Synthesebank zu gewährleisten. Für

eine Abtastfrequenz von $f_s = 16$ kHz eignet sich eine FFT-Länge von $L = 512$. Eine Realisierung im Zeitbereich ist prinzipiell auch möglich. Der Algorithmus von Zelinski in [8], der in Kapitel 4 noch näher erläutert wird, zeigt die Lösung im Zeitbereich. Die einfache Berechnung und Handhabung der Filter im Frequenzbereich und die rechentechnisch geringe Komplexität der analysierten Algorithmen empfiehlt jedoch die Verwendung einer FFT-Filterbank.

Das Grundmodell besteht aus der Kombination *fixer Beamformer mit adaptivem Postfilter*, welches im weiteren Verlauf der Arbeit als *Postfilterstruktur* bezeichnet wird. Der Beamformer übernimmt den Laufzeitausgleich und eine entsprechende Gewichtung der Eingangssignale an den einzelnen Mikrofonen. Dadurch entsteht eine Richtcharakteristik, die einer Keule in Richtung der Sprachquelle entspricht und damit für eine räumliche Filterung⁹ sorgt. Somit werden sämtliche Signalanteile, die nicht aus der gewünschten Richtung kommen unterdrückt. Es wird ein fixer Beamformer verwendet. Dies bedeutet, dass die Filterkoeffizienten des Beamformers w_i aufgrund der vorhandenen Arraystruktur, der Wunschrichtung und des Rauschfeldes fix berechnet werden und nicht adaptiv eingestellt werden. Dabei liegt i im Bereich $[1, K]$ und K ist die Anzahl der Mikrofone. Basierend auf unserer Herleitung Gl. 2.19 entspricht der optimale Beamformer einem MVDR Beamformer. Abhängig von der optimalen Wirkungsweise des Beamformers befinden sich auch nach der räumlichen Filterung noch Rauschanteile im bereits vorgefilterten Signal. Dabei handelt es sich um sämtliche Störungen, die aus derselben Richtung kommen wie das gewünschte Sprachsignal. Diese Störanteile werden anschließend mit dem einkanalen Postfilter H_{post} weiter reduziert und damit das Signal-Rausch-Verhältnis stark verbessert. Der Postfilterterm aus Gl. 2.21 wird anhand der Leistungsdichtespektren von Sprachsignal und Rauschsignal berechnet. Da Sprache und Rauschen kurzzeitstationären Prozessen entsprechen, muss das Postfilter H_{post} für jeden Frame neu berechnet werden. Man spricht daher von einem adaptiven Postfilter. Weiters sind die Leistungsdichtespektren von Sprache und Rauschen nicht getrennt voneinander verfügbar. Man muss versuchen eine möglichst optimale Schätzung bzw. Näherung für die Wiener Lösung abzuleiten. Die meisten Verfahren verwenden zur Schätzung des Postfilters die verrauschten Eingangssignale nach einer entsprechenden Kompensation der unterschiedlichen Laufzeiten. Dieser erfolgt im Frequenzbereich mit Hilfe des konjugiert komplexen Steuervektors $\mathbf{d}^*$ (siehe Abbildung 2.3). Die Spektralkomponenten $\tilde{X}_i(m, v)$ der Eingangssignale mit Laufzeitausgleich ergeben sich gemäß:

$$\tilde{X}_i(m, v) = X_i(m, v)d_i^*(v). \quad (2.25)$$

⁹Die Unterscheidung zwischen unterschiedlichen Signalen anhand der räumlichen Position ihrer Signalquellen.

2.5 Schätzung des Leistungsdichtespektrums

Eine übliche Methode zur Schätzung des Auto- und Kreuzleistungsdichtespektrums ist die Verwendung der rekursiven Formel von Welch [9],

$$\hat{\phi}_{x_i x_i}(m, v) = \alpha \hat{\phi}_{x_i x_i}(m-1, v) + (1-\alpha) X_i(m, v) X_i^*(m, v) \quad \text{bzw.} \quad (2.26)$$

$$\hat{\phi}_{x_i x_j}(m, v) = \alpha \hat{\phi}_{x_i x_j}(m-1, v) + (1-\alpha) X_i(m, v) X_j^*(m, v). \quad (2.27)$$

Zur Schätzung der aktuellen Auto- bzw. Kreuzleistungsdichte werden die Kurzzeitspektren $X_i(m, v)$ und $X_j^*(m, v)$ der Signale $x_i(k)$ und $x_j(k)$, sowie die Leistungsdichteschätzung des vergangenen Frames $\hat{\phi}_{x_i x_i}(m-1, v)$ ¹⁰ bzw. $\hat{\phi}_{x_i x_j}(m-1, v)$ verwendet. Mit Hilfe des Parameters α erreicht die rekursive Formel eine exponentielle Gewichtung vergangener Frames. Der Parameter liegt im Bereich $0 < \alpha < 1$ und ist gemäß

$$\alpha = \exp\left(-\frac{M}{\tau f_s}\right) \quad (2.28)$$

definiert, wobei M der Dezimationsfaktor der FFT-Filterbank, f_s die Abtastfrequenz in kHz und τ die Zeitkonstante in ms ist.

Dabei ist die Wahl von τ und damit auch von α sehr entscheidend, da sie bestimmen, wie stark die geschätzten Leistungsdichtespektren geglättet werden. Wird τ klein gewählt, und damit α auch klein, dann kommt es zu einer großen Varianz der geschätzten Leistungsdichtespektren und in weiterer Folge zu einer starken Variation der Postfilterfunktion. Dadurch treten im Ausgangssignal Zeitartefakte auf. Wird τ groß gewählt, und damit α auch groß, dann ist durch die starke Glättung der Leistungsdichtespektren die Annahme der Kurzzeitstationarität verletzt und es kommt zu einem verstärkten Nachhalleffekt im Ausgangssignal. In dieser Arbeit wird α als Glättungsfaktor bezeichnet.

Es hat sich in den Experimenten gezeigt, dass ein Wert von α bei ca. 0.8 sehr gute Ergebnisse liefert.

2.6 Kohärenzmatrix und Kohärenzfunktion

2.6.1 Begriffserklärung und Definition

Gl. 2.19 zeigt die Abhängigkeit des MVDR Beamformers von der spektralen Rausch-Korrelationsmatrix Φ_{nn} . Durch die Abhängigkeit des Postfilters H_{post} vom Ausgangs-SNR des Beamformers wird letztendlich auch das Postfilter von der Rausch-Korrelationsmatrix beeinflusst. Daher spielt die Matrix für den Entwurf und den optimalen Gewinn des Gesamtsystems eine zentrale Rolle. Die Korrelationsmatrix des Rauschens charakterisiert das vorhandene Rauschfeld. Verschiedene Konstellationen von Störquellen und damit unterschiedliche Rauschfelder ergeben unterschiedliche Matrizen. In der Praxis ist die Rausch-Korrelationsmatrix jedoch

¹⁰Im Folgenden werden alle geschätzten Größen mit einem darüber gestellten Dach ($\hat{x}$) geschrieben.

weder bekannt, noch zeitunabhängig. Daher muss man sich mit Modellen für Korrelationsmatrizen begnügen.

Im Weiteren wird die normierte, spektrale Rausch-Korrelationsmatrix als räumliche Kohärenzmatrix bezeichnet, die folgendermaßen definiert ist:

$$\mathbf{\Gamma}_{nn} = \begin{pmatrix} 1 & \Gamma_{n_1 n_2} & \Gamma_{n_1 n_3} & \cdots & \Gamma_{n_1 n_K} \\ \Gamma_{n_2 n_1} & 1 & \Gamma_{n_2 n_3} & \cdots & \Gamma_{n_2 n_K} \\ \vdots & \vdots & \ddots & \vdots & \\ \Gamma_{n_K n_1} & \Gamma_{n_K n_2} & \Gamma_{n_K n_3} & \cdots & 1 \end{pmatrix}. \quad (2.29)$$

Die Elemente der Matrix kann man durch Messen der komplexen Kohärenzfunktion

$$\Gamma_{n_i n_j} = \frac{\phi_{n_i n_j}}{\sqrt{\phi_{n_i n_i} \phi_{n_j n_j}}}, \quad (2.30)$$

bestimmen, wobei $|\Gamma_{n_i n_j}| \leq 1$ ist. Es handelt sich hierbei um die normierte, spektrale Korrelation zwischen den Signalen n_i und n_j zweier räumlich diskreter Punkte eines Rauschfeldes. Damit die spektralen Leistungsdichten des Rauschanteils mit Hilfe der Welch-Formel Gl. 2.26 bzw. 2.27 berechnet werden können, müssen jedoch die Rauschsignale getrennt vom Sprachsignal verfügbar sein. Da dies in der Praxis nicht der Fall ist, werden zur einfacheren Analyse der Postfilterstruktur die in Abschnitt 2.2.2 theoretisch definierten Rauschfeldern anhand der Kohärenzfunktion bzw. Kohärenzmatrix ausgedrückt. Diese werden als stationär angenommen.

2.6.2 Räumliche Kohärenzfunktionen

Die folgenden Kohärenzfunktionen werden hier ganz allgemein abhängig von der Frequenz f dargestellt. Sie sind unter Annahme der Fernfeldnäherung und der Verwendung omnidirektionaler Mikrofone abgeleitet. Für die Herleitung wurde von einem eindimensionalen Mikrofonarray und einem kartesischen Koordinatensystem ausgegangen, da in dieser Arbeit ausschließlich eindimensionale, lineare Mikrofonarrays untersucht wurden. Aufgrund der dadurch entstehenden Rotationssymmetrie der Anordnung reicht zur Beschreibung der Einfallsrichtung einer ebenen Welle auf das Array ein einziger Winkel (θ) aus.

Inkohärentes Rauschfeld

Da ein inkohärentes Rauschfeld durch unkorrelierte Rauschquellen an den Mikrofoneingängen entsteht, ist die Kohärenzfunktion für diesen Fall

$$\Gamma_{n_i n_j}(f) = 0, \quad \forall i \neq j \quad \text{und} \quad \Gamma_{n_i n_j}(f) = 1, \quad \forall i = j. \quad (2.31)$$

Damit entspricht die Kohärenzmatrix für alle Frequenzen der Einheitsmatrix,

$$\Gamma_{nn} = I. \quad (2.32)$$

Kohärentes Rauschfeld

Setzt sich das Rauschfeld aus nur einer Rauschquelle im freien Raum zusammen und hat man *a-priori* Wissen über die Position dieser Rauschquelle, dann kennt man auch den Einfallswinkel θ_{koh} der ebenen Welle, die diese Rauschquelle erzeugt. Unter Annahme der Fernfeldnäherung und der Verwendung der Phasendifferenz zweier Mikrofonsignale $\Delta\varphi = \frac{2\pi f}{c} l \cos(\theta)$ (siehe Abschnitt 3.1.1) ergibt sich somit die komplexe Kohärenzfunktion eines kohärenten Rauschfeldes

$$\Gamma_{n_i n_j}(f) = \exp(-j\Delta\varphi_{\text{koh}}(f)) = \exp\left(-j\frac{2\pi f}{c} l_{ij} \cos(\theta_{\text{koh}})\right). \quad (2.33)$$

Hierbei ist $c = 340$ m/s die Schallgeschwindigkeit und l_{ij} der Abstand zwischen den Mikrofonen i und j in Meter.

Sphärisch isotropes Rauschfeld

Erste Berechnungen der Korrelation zweier diskreter Punkte in einem isotropen Rauschfeld wurden von R. K. Cook *et al.* in [10] publiziert. Die Kohärenzfunktion eines sphärisch isotropen Rauschfeldes berechnet sich durch die Integration über alle Raumrichtungen, θ und ψ , zu

$$\begin{aligned} \Gamma_{n_i n_j}(f) &= \frac{1}{4\pi} \int_0^\pi \int_0^{2\pi} \exp\left(-j\frac{2\pi f}{c} l_{ij} \cos(\theta)\right) \sin(\theta) d\theta d\psi \\ &= \frac{\sin\left(\frac{2\pi f}{c} l_{ij}\right)}{\frac{2\pi f}{c} l_{ij}} = \text{sinc}\left(\frac{2\pi f}{c} l_{ij}\right) \end{aligned} \quad (2.34)$$

Abbildung 2.4 zeigt einen Vergleich der gemessenen Kohärenzfunktion des Rauschfeldes eines Büroraumes mit dem theoretisch definierten Modell aus Gl. 2.34. Zur Messung der Kohärenzfunktion wurde das Rauschfeld eines Büroraums, bestehend aus Lüftergeräusche von PCs und Klimaanlage, mit einem Mikrofonarray aufgenommen. Dann wurden mit Hilfe der Gleichungen 2.26 und 2.27 die Leistungsdichtespektren und letztendlich die komplexe Kohärenzfunktion gemäß Gl. 2.30 berechnet. Der Vergleich von Theorie und Messung bestätigt die Verwendung dieses Modells für Büroräume, aber auch für zahlreiche, ähnliche akustische Umgebungen. Weiters erkennt man, dass die spektrale Kohärenz des diffusen Rauschfeldes eine Funktion des Abstands zwischen den Mikrofonen ist. Bei größer werdendem Abstand l zwischen zwei Mikrofonen wandert die erste Nullstelle der Kohärenzfunktion aufgrund der Beziehung $f = \frac{c}{2l}$ zu niedrigeren Frequenzen. Die Kohärenz nimmt ab der ersten Nullstelle nur mehr relativ kleine Werte an. In Folge dessen wird bei der Verwendung von Arrays mit

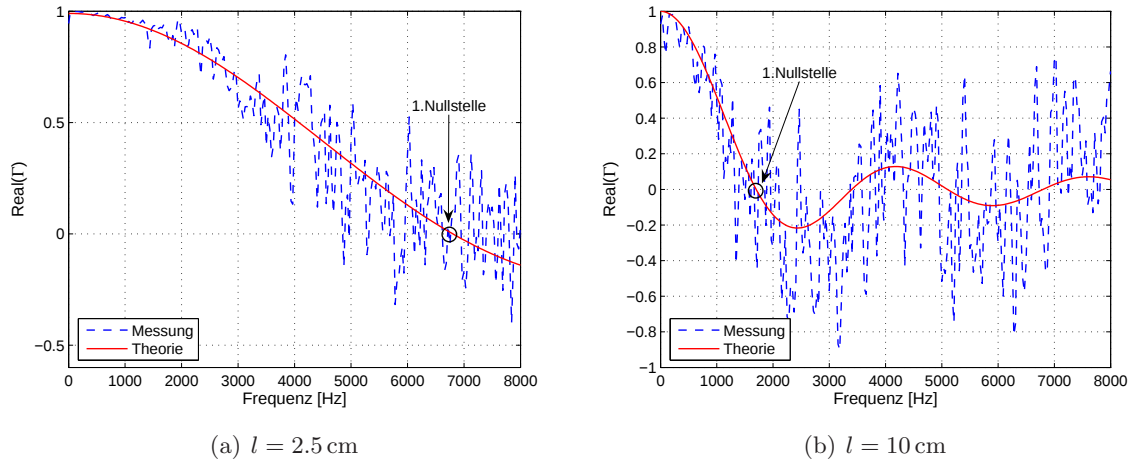


Abbildung 2.4: Realteil der gemessenen Kohärenzfunktion eines Büroraum und des theoretischen Modells bei einem Mikrofonabstand von (a) $l = 2.5 \text{ cm}$ und (b) $l = 10 \text{ cm}$.

großen Mikrofonabständen oft ein unkorreliertes Rauschfeld als Modell angenommen.

Zylindrisch isotropes Rauschfeld

Für einige akustische Rauschumgebungen (z.B. Cocktail-Party Noise) eignet sich die Verwendung eines zylindrisch isotropen Rauschfeldes als Modell. Im Gegensatz zum sphärisch isotropen Modell wird nun der zweidimensionale Fall betrachtet. Für solch ein Rauschfeld ist die Kohärenzfunktion

$$\Gamma_{n_i n_j}(f) = \frac{1}{2\pi} \int_0^{2\pi} \exp\left(-j \frac{2\pi f}{c} l_{ij} \cos(\theta)\right) d\theta = J_0\left(\frac{2\pi f}{c} l_{ij}\right), \quad (2.35)$$

wobei J_0 die Bessel Funktion nullter Ordnung ist.

Kapitel 3

Beamformer

In diesem Kapitel wird auf den Entwurf von Arraystrukturen und ihre Auswirkungen auf die Rauschunterdrückungseigenschaften des Beamformers eingegangen. Des Weiteren werden der Delay&Sum Beamformer und der superdirektive Beamformer als zwei Spezialfälle des MVDR Beamformers näher erklärt und deren Eigenschaften miteinander verglichen. Zuletzt wird eine Möglichkeit präsentiert, den Beamformer robust gegenüber einer Verstärkung des Eigenrauschens der Mikrofonsensoren zu machen.

Detaillierte mathematische Beschreibungen von Mikrofonarrays und Beamformer-Techniken findet man auch in [2, 11, 12].

3.1 Allgemeines

3.1.1 Grundprinzip

Ein Mikrofonarray besteht aus mehreren in der Ebene angeordneten Mikrofonen. Da in den Experimenten eindimensionale Arrays mit omnidirektionalen Mikrofonen verwendet wurden, bezieht sich die folgende allgemeine Beschreibung der Arrays und Beamformer ausschließlich auf diese Arraystruktur. Die Mikrofone sind dabei entlang einer Linie angeordnet. Durch die sich ergebende Rotationssymmetrie um die Arrayachse ist nur ein Winkel, der Einfallswinkel θ , notwendig, um die Einfallsrichtung einer ebenen Wellenfront zu beschreiben.

Eine Signalquelle erzeugt unter Annahme der Fernfeldnäherung eine homogene, ebene Welle im Raum. Die Signale, die von zwei Mikrofonen mit Abstand l empfangen werden, haben unterschiedliche Laufzeiten. Die Laufzeitdifferenz beträgt $\tau = \frac{l}{c} \cos(\theta)$ mit der Schallgeschwindigkeit c und dem Einfallswinkel θ . Daraus ergibt sich mit der Kreisfrequenz $\Omega = 2\pi f$ die resultierende Phasenverschiebung

$$\Delta\varphi(f, \theta) = \omega\tau = \frac{2\pi f}{c} l \cos(\theta). \quad (3.1)$$

In Abbildung 3.1 trifft eine ebene Wellenfront auf ein Mikrofonarray. Definiert man ein Referenzmikrofon, sozusagen das Mikrofon, auf das sich die Laufzeitdifferenzen und somit die

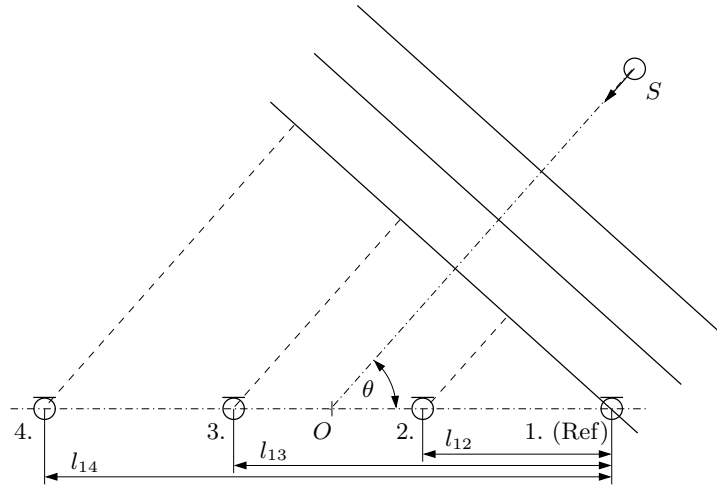


Abbildung 3.1: Phasenverschiebungen zwischen den Mikrofonensignalen.

Phasenverschiebungen beziehen, und den Einfallswinkel θ , dann können die Phasenverschiebungen der Mikrofonensignale im Steuervektor

$$\mathbf{d}(f, \theta) = [1, \exp(-j\omega\tau_{12}), \dots, \exp(-j\omega\tau_{1K})]^T \quad (3.2)$$

zusammengefasst werden. Die Laufzeitdifferenz τ_{1i} berechnet sich dabei aus dem Abstand l_{1i} zwischen dem Mikrofon i und dem Referenzmikrofon, wobei i im Bereich $[2, K]$ liegt.

Durch Beeinflussung der Laufzeiten und geeigneter Kombination der Eingangssignale kann eine beliebige Richtcharakteristik erzielt werden. Kennt man also den Einfallswinkel des gewünschten Sprachsignals, dann können durch entsprechende Keulenformung die Signale aus dieser Richtung räumlich extrahiert werden, wobei Störgeräusche aus anderen Richtungen unterdrückt werden. Im Frequenzbereich funktioniert dies prinzipiell durch Ausgleichen der Phasenverschiebungen, die im Steuervektor berechnet wurden.

3.1.2 Analysegrößen

Die folgenden, definierten Größen dienen der Analyse und dem Entwurf von Beamformer. Ausführlichere Beschreibungen dieser Größen findet man in [2, 11, 12]. Für eine kompaktere Schreibweise wurde die Frequenz f bzw. der Frequenzindex v weggelassen.

Arraygewinn

Der Arraygewinn G entspricht dem Gewinn an SNR, der durch die Keulenformung erreicht wird. Er ist definiert als Verhältnis des SNR eines Referenzsensors und des SNR am Arrayausgang:

$$G = \frac{SNR_{\text{Array}}}{SNR_{\text{Sensor}}}. \quad (3.3)$$

Mit Hilfe des LDS des reinen Sprachsignals in Wunschrichtung θ_0 am Ausgang des Arrays

$$\phi_{y_b y_b} \Big|_{\text{Signal}} = \phi_{ss} |\mathbf{b}^H \mathbf{d}(\theta_0)|^2 \quad (3.4)$$

($\mathbf{b}^H \dots$ Beamformerkoeffizienten, siehe Gl. 2.20) und dem LDS des Rauschsignals am Ausgang des Arrays

$$\phi_{y_b y_b} \Big|_{\text{Rauschen}} = \phi_{nn} \mathbf{b}^H \mathbf{\Gamma}_{nn} \mathbf{b}, \quad (3.5)$$

ergibt sich der Gewinn zu

$$G = \frac{|\mathbf{b}^H \mathbf{d}(\theta_0)|^2}{\mathbf{b}^H \mathbf{\Gamma}_{nn} \mathbf{b}}, \quad (3.6)$$

wobei ϕ_{ss} das LDS des Sprachsignals und ϕ_{nn} die mittlere Rauschsignalleistung am Eingang eines Mikrofons sind.

Gewinn für inkohärentes Rauschen

Setzt man die Kohärenzmatrix eines inkohärenten Rauschfeldes Gl. 2.32 in Gl. 3.6 ein, dann erhält man den Gewinn für inkohärentes Rauschen (WNG¹)

$$WNG = \frac{|\mathbf{b}^H \mathbf{d}(\theta_0)|^2}{\mathbf{b}^H \mathbf{b}} \leq K, \quad (3.7)$$

wobei K die Anzahl der Mikrofone ist. Das WNG beschreibt die Fähigkeit des Beamformers räumlich unkorrelierte Rauschsignale zu unterdrücken. Betrachtet man das WNG auf einer logarithmischen Skala, dann repräsentieren positive Werte eine Dämpfung und negative Werte eine Verstärkung der räumlich unkorrelierten Rauschsignale.

Direktivität

Die Direktivität beschreibt die Fähigkeit des Beamformers ein diffuses Rauschfeld zu unterdrücken. Durch Einsetzen der Kohärenzmatrix eines diffusen Rauschfeldes in Gl. 3.6 ergibt sich für das logarithmische Äquivalent, der so genannte Direktivitätsindex (DI)

$$DI = 10 \log_{10} \left(\frac{|\mathbf{b}^H \mathbf{d}(\theta_0)|^2}{\mathbf{b}^H \mathbf{\Gamma}_{nn} \Big|_{\text{diffus}} \mathbf{b}} \right). \quad (3.8)$$

Richtcharakteristik

Eine weiteres Analyseverfahren ist die Berechnung der Richtcharakteristik², auch Raum-Frequenz-Übertragungsfunktion genannt. Die Leistungsverstärkung eines Beamformers mit den

¹engl.: White Noise Gain

²engl.: Beampattern (BP)

Gewichten $\mathbf{b}$ errechnet sich für eine ebene Welle mit Einfallswinkel θ und bestimmter Frequenz f zu

$$BP(\theta, f) = |\mathbf{b}^H(\theta_0, f)\mathbf{d}(\theta, f)|^2. \quad (3.9)$$

Berechnet man diesen Wert für alle Frequenzen und Winkel θ , dann ergibt sich daraus die Richtcharakteristik.

3.2 Arrayentwurf

3.2.1 Räumliches Aliasing

Ein wichtiger Faktor beim Arrayentwurf ist *räumliches Aliasing*. Es tritt auf, wenn das räumliche Samplingtheorem [2]

$$l < \frac{\lambda_{\min}}{2} \quad (3.10)$$

nicht erfüllt ist. Die Bedingung 3.10 besagt, dass der Abstand zwischen den Mikrofonen den Wert $\lambda/2$ (halbe Wellenlänge) bei maximaler Frequenz $f_{\max} = f_s/2$ nicht überschreiten soll. Eine Verletzung dieser Bedingung führt zur Mehrdeutigkeit [1]. Andererseits sollte jedoch der Abstand zwischen den Mikrofonen möglichst groß sein, damit auch bei niedrigen Frequenzen eine gute räumliche Selektivität erreicht wird. Diese zwei Randbedingungen bestimmen die Geometrie eines Mikrofonarrays.

Je größer die Anzahl an Mikrofonen, desto größer ist auch die erreichbare räumliche Selektivität. Untersuchungen in dieser Diplomarbeit haben ergeben, dass für Arrays unter vier Mikrofonen keine nennenswerte Keulenformung erzielt wird.

3.2.2 Positionierung der Mikrofone

Lineares, äquidistantes Array

Ein häufig verwendetes Array ist ein lineares, äquidistantes Array. Die Mikrofone werden in einem konstanten Abstand zueinander positioniert. Der konstante Abstand $l = c/f_s$ wird dabei üblicherweise mit Hilfe der Bedingung 3.10 berechnet. Zum Beispiel wählt man für die Abtastfrequenz $f_s = 16$ kHz einen Abstand von ca. $l = 2.1$ cm, wobei die Schallgeschwindigkeit $c = 340$ m/s beträgt.

Harmonisches Array

Die Übertragungscharakteristik des Arrays (Hauptkeulenbreite und Nebenkeulen) bleibt nur für schmalbandige Signale konstant. Da es sich bei Sprache jedoch um ein Breitbandsignal handelt, ist ein einziges lineares Array nicht geeignet, falls eine frequenzunabhängige Richtcharakteristik erwünscht ist. Eine einfache Methode um Breitbandsignale zu verarbeiten, ist

die Verwendung von harmonischen Arrays. Die Idee dahinter ist, dass für verschiedene Frequenzbereiche verschiedene lineare Arrays, so genannte Subarrays, verwendet werden. Die Abstände zwischen den Mikrofonen jedes Subarrays werden dabei passend für den jeweiligen Frequenzbereich mit Hilfe des räumlichen Samplingtheorems (Bedingung 3.10) gewählt. Um eine konstante Keulenbreite zu garantieren, sind mit steigender Frequenz immer kleinere Abstände zwischen den Mikrofonen notwendig. Somit setzt sich ein harmonisches Array aus einem Set äquidistanter Arrays zusammen. Üblicherweise wird ein solches Array ab einer Anzahl von mehr als fünf Mikrofonen verwendet, typischerweise bei 8–11 Mikrofone. Abbildung 3.2 zeigt ein Beispiel, wie man ein harmonisches Array mit 9 Mikrofonen für eine Abtastfrequenz von $f_s = 16$ kHz dimensionieren kann.

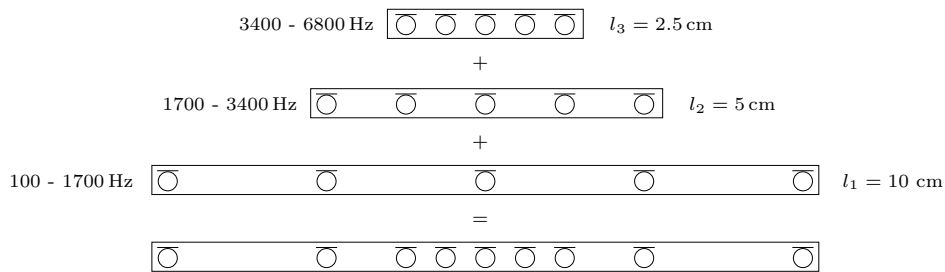


Abbildung 3.2: Harmonisches Array mit 9 Mikrofonen ($f_s = 16$ kHz).

Subband-Array

Ein Subband-Array besitzt die gleiche Struktur wie ein harmonisches Array. Es setzt sich ebenso aus einem Set äquidistanter Subarrays zusammen. Da jedes Subarray für einen gewissen Frequenzbereich steht, werden die Ausgänge von jedem Subarray zusätzlich einem zugehörigen Bandpassfilter unterworfen, um räumliches Aliasing gänzlich auszuschließen (siehe Abb. 3.3). Die Bandpassfilter müssen dabei so entworfen werden, dass die Gesamtübertragungsfunktion der Filter einen flachen Amplitudengang und eine lineare Phase aufweist. Da im Gegensatz zum harmonischen Array, abhängig von der Frequenz, immer nur die Mikrofone eines einzigen Subarrays aktiv sind und nicht alle vorhandenen Mikrofone gleichzeitig, ergibt sich auch eine geringere Keulenbreite und damit eine geringere räumliche Selektivität.

3.2.3 Positionierung der Sprachquelle

Die Sprachquelle kann prinzipiell in einem beliebigen Winkel θ_0 zum Ursprung des Mikrofonarrays positioniert werden. Damit die Annahme der Unkorreliertheit von Sprache und Rauschen erfüllt ist, wird der Abstand typischerweise in einem Bereich von 60–80 cm gewählt. In der Literatur werden jedoch meist zwei Grundkonfigurationen verwendet (siehe Abb. 3.4). Bei der Broadside-Konfiguration wird die Sprachquelle in einem Winkel von $\theta_0 = 90^\circ$ zum Array

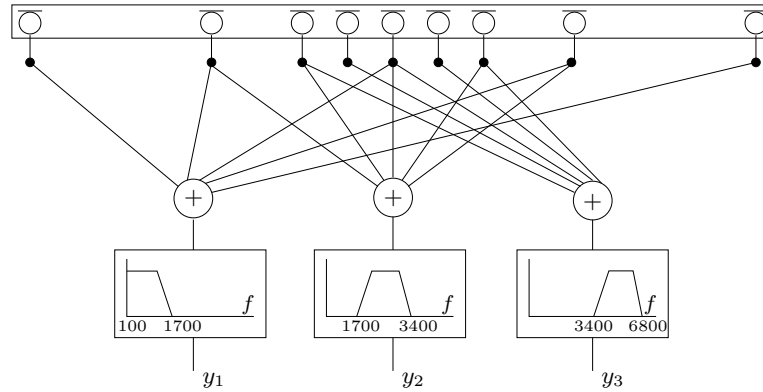


Abbildung 3.3: Subband Array bestehend aus drei Subarrays mit zugehörigen Bandpassfiltern.

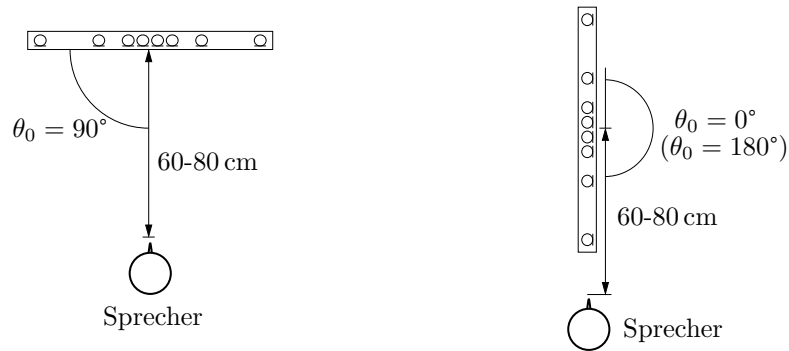


Abbildung 3.4: Broadside (links) und Endfire (rechts) Positionierung der Sprachquelle.

positioniert. Die Endfire-Konfiguration sieht eine Positionierung in einem Winkel von $\theta_0 = 0^\circ$ bzw. $\theta_0 = 180^\circ$ vor.

3.3 Optimaler Beamformer

3.3.1 MVDR Beamformer

Auch für den Beamformer ohne Postfilter kann eine optimale Lösung gefunden werden. Ein optimaler Entwurf im Sinne einer Maximierung des Arraygewinns kann durch Minimierung der Ausgangsleistung des Beamformers unter der Nebenbedingung einer unverzerrten Wiedergabe des Signals in Wunschrichtung θ_0 erzielt werden:

$$\min_{\mathbf{b}} \{ \mathbf{b}^H \Phi_{xx} \mathbf{b} \} \quad \text{mit} \quad \mathbf{b}^H \mathbf{d}(\theta_0) = 1. \quad (3.11)$$

Unter Verwendung der Lagrange'schen Multiplikation kann daraus der Minimum Variance Distortionless Response (MVDR) Beamformer mit dem optimalen Filtervektor

$$\mathbf{b}_{\text{mvdr}}(\theta_0) = \frac{\Phi_{nn}^{-1} \mathbf{d}(\theta_0)}{\mathbf{d}^H(\theta_0) \Phi_{nn}^{-1} \mathbf{d}(\theta_0)} \quad (3.12)$$

hergeleitet werden. Zumal man eine optimale Unterdrückung des Rauschens wünscht, ist nur der Rauschanteil der spektrale Korrelationsmatrix Φ_{nn} für die Minimierung maßgebend. Eine ausführliche Herleitung des optimalen Beamformerfilters findet man in [11, 12]. Gl. 3.13 stimmt exakt mit dem Beamformerterm der Postfilterstruktur, hergeleitet in Gl. 2.19, überein. Obwohl der MVDR Beamformer bereits eine optimale Verbesserung des SNR im Sinne von Maximum Likelihood (ML) erreicht, maximiert er dennoch nicht das SNR im Sinne einer Breitband MMSE Lösung. Dies wird erst durch die Kombination mit einem adaptiven Postfilter erzielt.

Ein maximaler Gewinn kann nur für jenes Rauschfeld erreicht werden, für das der MVDR Beamformer entworfen wurde. Für eine umfangreiche Analyse des MVDR Beamformers werden die theoretisch definierten Modelle der Kohärenzmatrix Γ_{nn} anstatt der Matrix Φ_{nn} in Gl. 3.12 eingesetzt:

$$\mathbf{b}_{\text{mvdr}} = \frac{\Gamma_{nn}^{-1} \mathbf{d}(\theta_0)}{\mathbf{d}^H(\theta_0) \Gamma_{nn}^{-1} \mathbf{d}(\theta_0)}. \quad (3.13)$$

Für verschiedene Rauschfelder ergeben sich daher unterschiedliche Beamformer Lösungen (siehe die folgenden Abschnitte 3.3.2 und 3.3.3).

3.3.2 Delay&Sum Beamformer (DSB)

Der DSB maximiert den Arraygewinn für räumlich unkorrelierte Rauschsignale. Die Koeffizienten lassen sich durch Einsetzen der Kohärenzmatrix eines inkohärenten Rauschfeldes $\Gamma_{nn} = \mathbf{I} = \mathbf{I}^{-1}$ in Gl. 3.13 bestimmen. Es ergibt sich

$$\mathbf{b}_{\text{DSB}} = \frac{\mathbf{I} \mathbf{d}(\theta_0)}{\mathbf{d}^H(\theta_0) \mathbf{I} \mathbf{d}(\theta_0)} = \frac{1}{K} \mathbf{d}(\theta_0), \quad (3.14)$$

wobei K die Anzahl der Mikrofone ist. Gl. 3.14 zeigt, dass das Ausgangssignal des DSB durch Ausgleichen der Laufzeitverzögerungen und anschließender Bildung des Mittelwerts berechnet wird. Aus diesem Grund wird üblicherweise die Bezeichnung *Delay&Sum Beamformer* verwendet. Das WNG ist für ein inkohärentes Rauschfeld klarerweise optimal und erreicht ihren Maximalwert $WNG = K$. Wie gut räumlich unkorreliertes Rauschen unterdrückt wird, hängt also beim DSB nur von der Anzahl der Mikrofone ab.

3.3.3 Superdirektiver Beamformer (SDB)

Passt man den MVDR Beamformer für den Spezialfall eines diffusen Rauschfeldes an, dann erhält man den klassischen, superdirektiven Beamformer. Er resultiert aus Gl. 3.13 durch Einsetzen der Kohärenzmatrix des diffusen Rauschfeldes, definiert durch Gl. 2.34. Der Begriff der „Superdirektivität“ folgt aus der, durch eine geeignete Kombination der Mikrofonsignale erzielten, optimalen Direktivität.

3.3.4 Vergleich DSB – SDB

Abbildung 3.5 zeigt das WNG und den Direktivitätsindex für den DSB und den SDB im logarithmischen Maß in Abhängigkeit von der Frequenz. Obwohl diese Größen für eine bestimmte Arraystruktur, ein 5-kanaliges, äquidistantes Array mit Mikrofonabstand $l = 2.5$ cm und einer Abtastfrequenz $f_s = 16$ kHz, berechnet wurde, spiegeln sie sehr gut das allgemeine Verhalten von DSB und SDB wieder. Wie erwartet, nimmt das WNG für den DSB für alle Frequenzen den Maximalwert $10 \log_{10}(K) = 10 \log_{10}(5) \approx 7$ dB an. Daher arbeitet der DSB in einem inkohärenten Rauschfeld optimal. Das WNG des SDB hat bei kleinen Frequenzen stark negative dB-Werte. Dies weist auf eine Verstärkung der räumlich unkorrelierten Rauschsignale (z.B. das Eigenrauschen der Mikrofone) in diesem Frequenzbereich hin. Es wird also eine höhere Direktivität zu Lasten einer Verstärkung räumlich unkorrelierter Störungen erkaufte. Dadurch werden bereits sehr kleine Störungen beim SDB stark verstärkt. Dies führt zu einer allgemeinen Verschlechterung des SNR am Beamformerausgang.

Der DI ist nahezu Null für niedrige Frequenzen, das heißt, der DSB arbeitet in einem diffusen Rauschfeld für niedrige Frequenzen nicht mehr optimal. Die Störungen, die nicht aus der Wunschrichtung kommen, werden nicht mehr ausreichend unterdrückt. Dies ist auf die hohe Korrelation des diffusen Rauschfeldes bei niedrigen Frequenzen zurückzuführen. Der SDB weist jedoch bei niedrigen Frequenzen eine sehr hohe Direktivität auf. Er arbeitet bei Vorhandensein eines diffusen Rauschfeldes optimal.

Für hohe Frequenzen nehmen der DI und das WNG beider Beamformer ähnliche Werte an, da die Kohärenz eines diffusen Rauschfeldes für hohe Frequenzen niedrig ist (siehe Gl. 2.34) und somit das Rauschfeld in diesem Fall annähernd unkorreliert ist. Bei Frequenzen ab ca. $f = 6800$ Hz macht sich, wie laut Bedingung 3.10 erwartet, räumliches Aliasing bemerkbar.

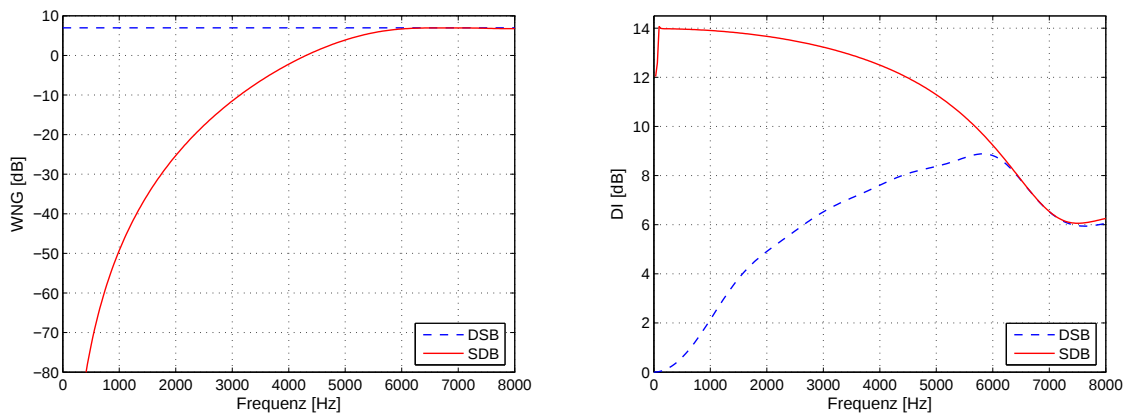


Abbildung 3.5: Links: Das WNG im logarithmischen Maß für den DSB und den SDB. Rechts: Der Direktivitätsindex (DI) für den DSB und den SDB. (Äquidistantes Array mit $K = 5$, $l = 2.5$ cm, $f_s = 16$ kHz und Endfire-Konfiguration)

3.4 Beschränkter Entwurf

Das Problem der Verstärkung des Eigenrauschens der Mikrofone beim superdirektiven Entwurf kann durch eine Beschränkung des WNG gelöst werden [13].

Für die Beschränkung des WNG wird eine kleine skalare Konstante μ zur Hauptdiagonale der Kohärenzmatrix addiert:

$$\mathbf{\Gamma}'_{nn} = \mathbf{\Gamma}_{nn} + \mu \mathbf{I}, \quad (3.15)$$

wobei $\mathbf{I}$ die Einheitsmatrix ist. Man spricht auch von einer *Regularisierung der Kohärenzmatrix*. Daraus ergibt sich das beschränkte Beamformerfilter

$$\mathbf{b}' = \frac{(\mathbf{\Gamma}_{nn} + \mu \mathbf{I})^{-1} \mathbf{d}(\theta_0)}{\mathbf{d}^H(\theta_0) (\mathbf{\Gamma}_{nn} + \mu \mathbf{I})^{-1} \mathbf{d}(\theta_0)} = \frac{\mathbf{\Gamma}'_{nn}{}^{-1} \mathbf{d}(\theta_0)}{\mathbf{d}^H \mathbf{\Gamma}'_{nn}{}^{-1} \mathbf{d}(\theta_0)}. \quad (3.16)$$

[2] zeigt eine äquivalente, mathematische Interpretation der Regularisierung, bei der jedes Element der Kohärenzmatrix mit Ausnahme der Hauptdiagonale durch den Wert $1 + \mu$ dividiert wird. In diesem Fall kann μ als Verhältnis zwischen dem unkorrelierten Eigenrauschen der Mikrofone σ^2 und der mittleren Umgebungsrauschleistung ϕ_{nn} interpretiert werden.

Die Elemente der Kohärenzmatrix eines diffusen Rauschfeldes ergeben dann

$$\Gamma'_{n_i n_j} = \frac{\text{sinc}\left(\frac{2\pi f}{c} l_{ij}\right)}{1 + \frac{\sigma^2}{\phi_{nn}}} \quad \forall i \neq j. \quad (3.17)$$

Auch die Kohärenzmatrix eines zylindrisch isotropen Rauschfeldes kann gemäß

$$\Gamma'_{n_i n_j} = \frac{J_0\left(\frac{2\pi f}{c} l_{ij}\right)}{1 + \frac{\sigma^2}{\phi_{nn}}} \quad \forall i \neq j \quad (3.18)$$

regularisiert werden. In beiden Fällen nimmt die Kohärenzmatrix demnach Werte kleiner gleich Eins an.

Der Parameter μ kann zwischen Null und Unendlich variieren, wobei $\mu = 0$ im klassischen SDB und $\mu = \infty$ im DSB resultiert. Das WNG ändert sich dabei als monotone Funktion zwischen diesen beiden Extremfällen. Typische Werte von μ reichen zwischen -10 dB und -40 dB (siehe Abbildung 3.6). Der beschränkte Entwurf erweist sich als guter Kompromiss zwischen optimalen WNG und DI. Die Regularisierung bringt noch einen weiteren Vorteil: Vor allem bei niedrigen Frequenzen treten oft Kohärenzmatrizen auf, die singulär und damit nicht invertierbar sind. Beide vorgestellten Regularisierungsmethoden lösen das Problem der Matrzensingularität.

Es gibt auch die Möglichkeit ein frequenzabhängiges μ zu bestimmen, welches das WNG auf einen konstanten Mindestwert beschränkt. Auf diese Methode wurde in der Diplomarbeit nicht näher eingegangen. Eine Lösungsmöglichkeit dazu findet man in [12], die eine ausführliche

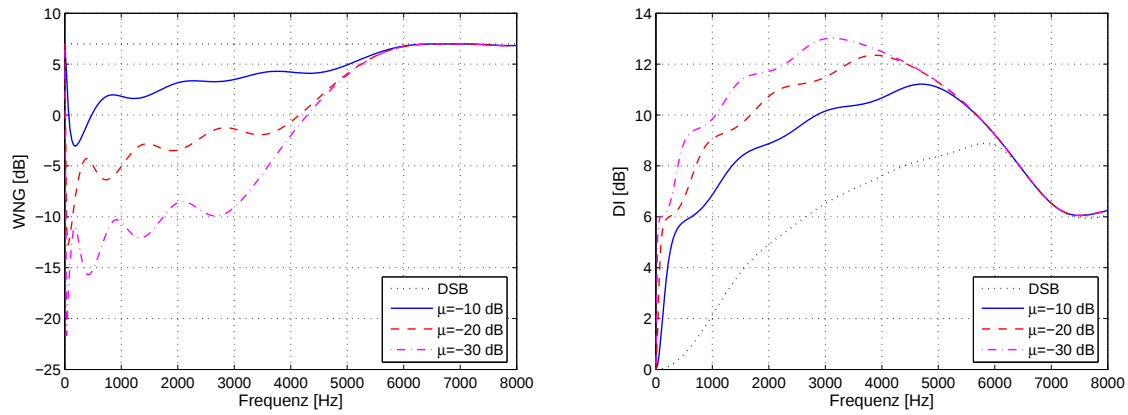


Abbildung 3.6: Links: Das WNG im logarithmischen Maß für verschiedene beschränkte Entwürfe im Vergleich zum DSB. Rechts: Der Direktivitätsindex (DI) für verschiedene beschränkte Entwürfe im Vergleich zum DSB. (Äquidistantes Array mit $K = 5$, $l = 2.5$ cm, $f_s = 16$ kHz und Endfire-Konfiguration)

Beschreibung der Regularisierung im Zusammenhang mit der Toleranzempfindlichkeit der Mikrofone bietet.

3.5 Vergleich verschiedener Beamformer-Konfigurationen

3.5.1 Unterschiedliche Arraystrukturen

Betrachtet man die Richtcharakteristik verschiedener Arraygeometrien, dann erkennt man allgemein eine schmalere Keulenbildung bei steigender Anzahl der verwendeten Mikrofone (siehe Abbildung 3.7 (a), (b) und (d) auf der nächsten Seite). Erst ab 4 Mikrofone zeigt sich eine nennenswerte Keulenbildung, woraus man schließen kann, dass bei Arrays mit weniger als 4 Mikrofone die Verwendung eines Beamformers nicht sinnvoll ist.

Jedoch spielt nicht nur die Anzahl der Mikrofone eine wichtige Rolle, sondern auch der Abstand zwischen den Mikrofonen. Ist die Bedingung 3.10 nicht mehr erfüllt, dann tritt räumliches Aliasing auf, welches sich in der Richtcharakteristik als so genannte „Grating Lobes“ bemerkbar macht. Dabei handelt es sich um Nebenkeulen, die sich über einen weiten Frequenz- und Winkelbereich erstrecken, und dadurch eine effiziente Rauschunterdrückung unmöglich machen. Abbildung 3.7 (c) zeigt das Auftreten von Grating Lobes bei einem äquidistanten DSB mit Mikrofonabstand $l = 5$ cm und der Abtastrate $f_s = 16$ kHz. Ab 7 Mikrofone wählt man meist eine harmonische Arraygeometrie. Dadurch wird eine konstante Leistungsfähigkeit über den gesamten Frequenzbereich erzielt. Außerdem wird eine noch höhere Direktivität bei niedrigen Frequenzen erreicht, wodurch man dort auch eine bessere Rauschunterdrückung erwarten kann. Abbildung 3.8 vergleicht die Direktivität eines 9-kanaligen, harmonischen Arrays (mit der Arraygeometrie aus Abbildung 3.2) mit einem 9-kanaligen, äquidistanten Array mit $l = 2.5$ cm bei einer Abtastrate von $f_s = 16$ kHz. Das äquidistante Array weist nur bei hohen

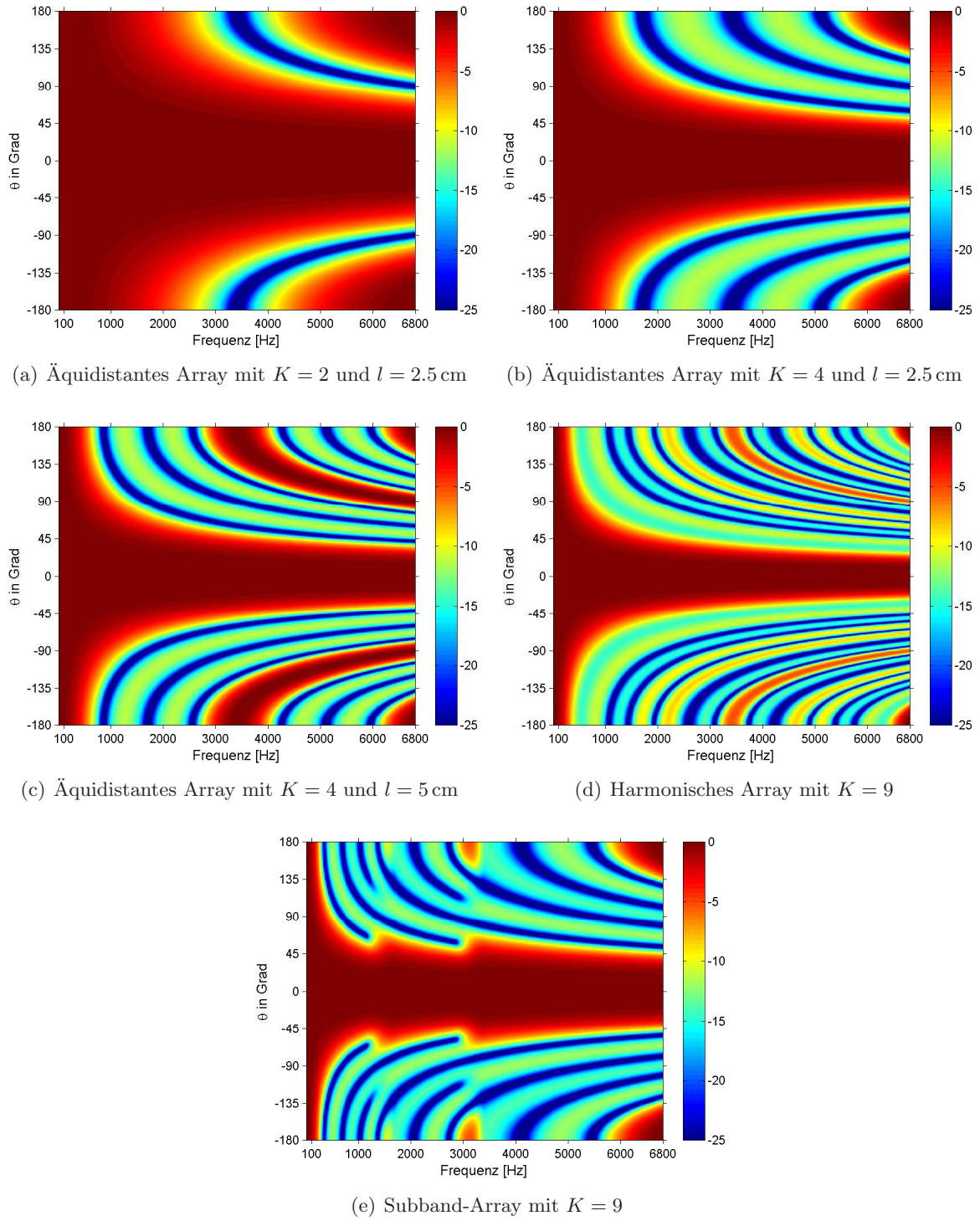


Abbildung 3.7: Richtcharakteristiken eines DSB bestehend aus unterschiedlichen Arraygeometrien ($f_s = 16$ kHz und Endfire-Konfiguration).

Frequenzen eine höhere Direktivität auf als das harmonische Array. Das harmonische Array zeigt allerdings höhere Werte bei den, für die Rauschunterdrückung wichtigen, niedrigeren Frequenzen und insgesamt einen eher konstanten Verlauf der Direktivität. In der Literatur ist daher er auch öfter von „directivity-controlled“ Arrays die Rede.

Die Unterschiede zwischen einem harmonischen Array und einem Subband-Array werden in Abbildung 3.7 (d) und (e) deutlich. Beim Subband-Array sind in jedem Frequenzbereich nie alle Mikrofone, sondern nur die Mikrofone eines Subarrays, aktiv, wodurch sich eine breitere Hauptkeule ergibt. Da die Direktivität von der Anzahl der aktiven Mikrofone abhängig ist, zeigt das Subband-Array eine geringere Direktivität als das harmonische Array, bei dem immer alle Mikrofone aktiv sind. Außerdem treten bei höheren Frequenzen stärkere Nebenkeulen auf.

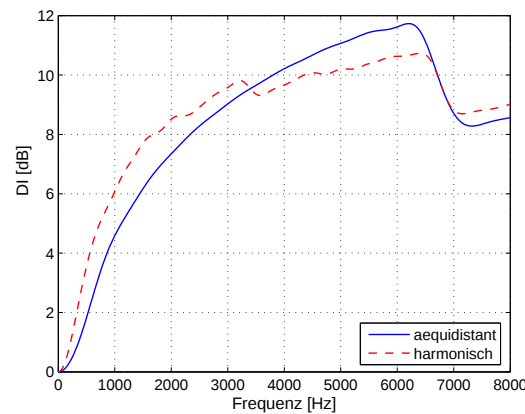


Abbildung 3.8: *DI eines äquidistanten Arrays mit $l = 2.5$ cm und eines harmonischen Arrays (DSB, $K = 9$, $f_s = 16$ kHz, Endfire-Konfiguration).*

3.5.2 DSB und SDB

Abbildung 3.7 hat gezeigt, dass der DSB bei niedrigen Frequenzen keine zufriedenstellende Keulenformung erzielt. Der Einsatz von superdirektiven Beamformer behebt dieses Problem durch eine optimale Gewichtung der Eingangssignale. Abbildung 3.9 zeigt die Richtcharakteristik eines äquidistanten Arrays und eines harmonischen Arrays für einen SDB.

3.5.3 Unterschiedliche Einfallswinkel

Abhängig vom Einfallswinkel θ_0 ändern sich die Eigenschaften des DSB und des SDB. Abbildung 3.10 (links) zeigt das WNG und den DI für den superdirektiven Beamformer für unterschiedliche Einfallswinkel. Die höchste Direktivität weist der Beamformer in Endfire-Konfiguration ($\theta_0 = 0^\circ$) auf. Auch bei niedrigen Frequenzen ergeben sich ausreichend hohe Werte. Ein Einfallswinkel von 45° und die Broadside-Konfiguration ($\theta_0 = 90^\circ$) zeigen niedrigere Werte, wobei in der Broadside-Konfiguration die Direktivität bei niedrigen Frequenzen stark abnimmt. Das bedeutet, dass der Beamformer in der Endfire-Konfiguration die beste

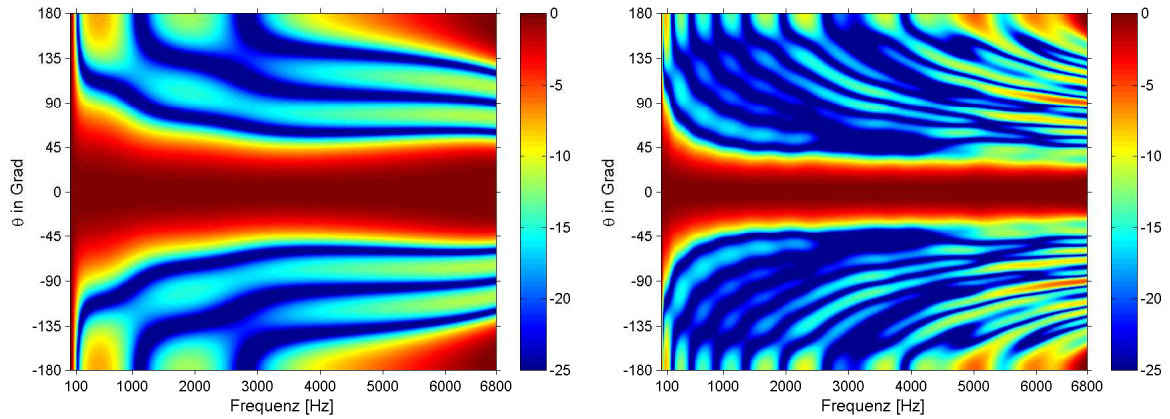


Abbildung 3.9: Richtcharakteristik zweier superdirektiver Beamformer ($f_s = 16 \text{ kHz}$, $\mu = -20 \text{ dB}$ und Endfire-Konfiguration). Links: 4-kanaliges äquidistantes Array mit $l = 5 \text{ cm}$. Rechts: 9-kanaliges harmonisches Array.

räumliche Selektivität erzielt und daher die beste Rauschunterdrückung zu erwarten ist. Bei Ausrichtung der Hauptkeule auf einen Einfallswinkel zwischen $\theta_0 = 0^\circ$ und $\theta_0 = 90^\circ$ ist Vorsicht geboten. Betrachtet man zum Beispiel die Richtcharakteristik eines SDB für einen Einfallswinkel von $\theta_0 = 45^\circ$ (siehe Abbildung 3.10, rechts), dann kann man erkennen, dass für bestimmte Winkel und Frequenzen Leistungsverstärkungen auftreten, die größer als 0 dB sind. Es handelt sich hierbei um extrem starke Nebenkeulen. In Folge dessen werden in diesen Bereichen sämtliche Signale, ebenso Rauschen, verstärkt, wodurch die selektive Wirkung des Beamformers verloren geht. Verwendet man solche Arraykonfigurationen, dann sollte man sichergehen, dass in den Winkel- und Frequenzbereichen, in denen die Leistungsverstärkungen vorkommen, keine Störer vorhanden sind. Vor allem im Auto werden solche Arraykonfigurationen üblicherweise eingesetzt. Da hier jedoch Störer aus allen Richtungen zu erwarten sind, kommt es oft zu einer verminderten Leistungsfähigkeit des gesamten Rauschunterdrückungssystems.

3.5.4 Spezielle Beamformer

Setzt man in die Formel für den MVDR Beamformer Gl. 3.13 eine spezielle Kohärenzmatrix ein, dann erhält man den optimalen Beamformer genau für die Rauschsituation, welche die Kohärenzmatrix charakterisiert.

MVDR Beamformer - optimal für zylindrisch isotropes Rauschfeld

In Abbildung 3.11 (links) sieht man die Richtcharakteristik eines MVDR Beamformers, der optimal für ein zylindrisch isotropes Rauschfeld ist. Im Vergleich zum SDB in Abbildung 3.9 (rechts) auf dieser Seite weist dieser Beamformer keine größeren Unterschiede. Jedoch zeigt sich eine verbesserte Richtwirkung für Quellen von hinten. Daher kommt dieser Beamformer oft bei der Rauschunterdrückung für Hörgeräte zum Einsatz.

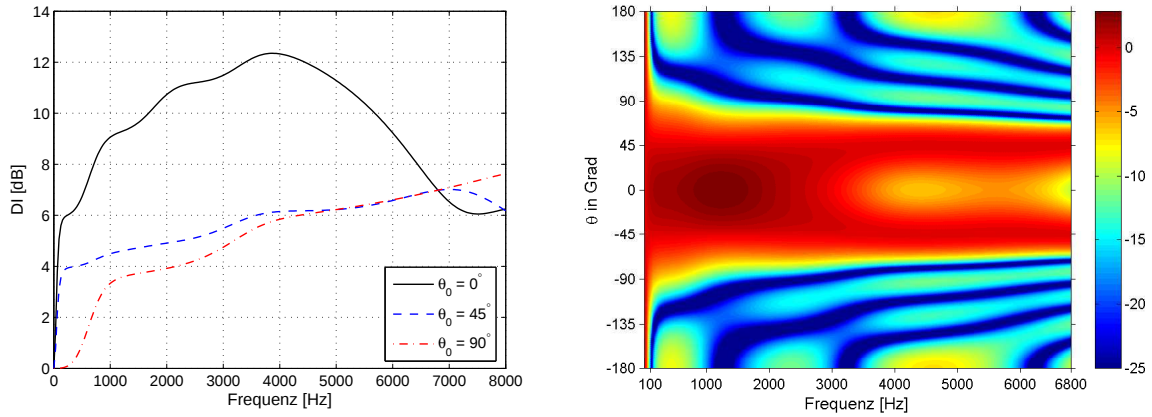


Abbildung 3.10: Links: Der Direktivitätsindex (DI) für verschiedene Einfallswinkel. Rechts: Richtcharakteristik für einen Einfallswinkel von 45° . (Äquidistantes Array mit $K = 5$, $l = 2.5$ cm, SDB, $f_s = 16$ kHz und Endfire-Konfiguration)

MVDR Beamformer - optimal kohärentes Rauschfeld

Abbildung 3.11 (rechts) zeigt die Richtcharakteristik für eine kohärente Rauschquelle mit Einfallswinkel $\theta_{\text{koh}} = 66^\circ$ in einer Entfernung von ca. 2 m. Der Beamformer ist zwar für diese Situation optimal, jedoch ist *a-priori* Wissen über die Position der Rauschquelle notwendig. Für alle anderen Rauschsituationen zeigt dieser Beamformer keine optimale Rauschunterdrückung. Ferner sollte beachtet werden, dass sich ebenfalls Winkel- und Frequenzbereich ausbilden, in denen Leistungsverstärkungen größer 0 dB auftreten.

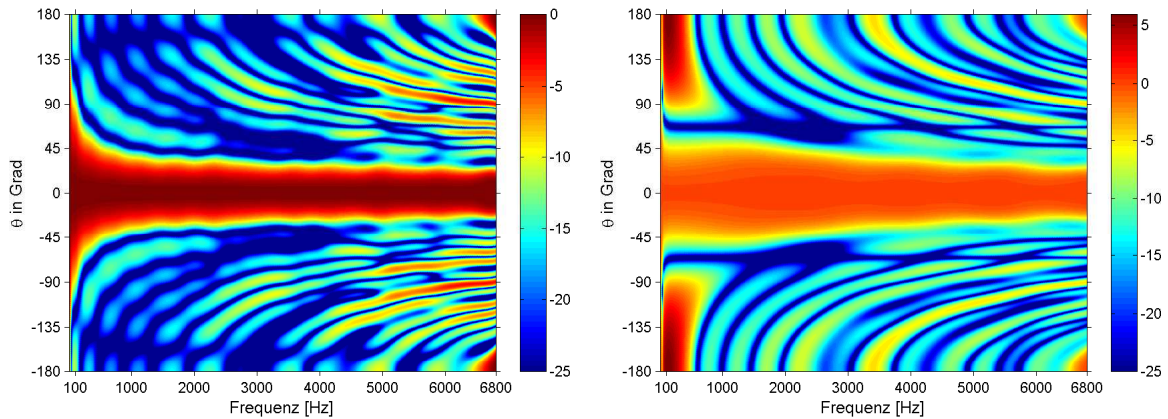


Abbildung 3.11: Richtcharakteristik zweier Spezialfälle des MVDR Beamformers (harmonisches Array, $K = 9$, $f_s = 16$ kHz, Endfire-Konfig.). Links: optimal für ein zylindrisch isotropes Rauschfeld. Rechts: optimal für eine kohärente Rauschquelle mit Einfallswinkel $\theta_{\text{koh}} = 66^\circ$.

Kapitel 4

Postfilter

Dieses Kapitel behandelt Herleitung und theoretische Analyse verschiedener Postfilteralgorithmen. Weiters werden Methoden und Maßnahmen zur Optimierung dieser Algorithmen vorgestellt.

Sämtliche Herleitungen werden im Frequenzbereich durchgeführt. Zur einfacheren Lesbarkeit und kompakteren Schreibweise wurden der zeitliche Frameindex m und der Frequenzindex v weggelassen. Die Mikrofonkanäle i und j liegen im Bereich $[1, K]$, wobei K die Anzahl der Mikrofone ist. Die Phasenverschiebungen der Eingangssignale werden durch den Steuervektor $\mathbf{d}$ aus Gl. 3.2 beschrieben. Der Operator $\Re\{\cdot\}$ kennzeichnet die Verwendung des Realteils. Im Übrigen wird die Definition für das Leistungsdichtespektrum

$$\phi_{yy} = E\{YY^*\}, \quad (4.1)$$

für den spektralen Kreuzkorrelationsvektor

$$\phi_{xy} = E\{\mathbf{x}Y^*\} \quad (4.2)$$

und die spektrale Korrelationsmatrix

$$\Phi_{xx} = E\{\mathbf{x}\mathbf{x}^H\} \quad (4.3)$$

verwendet, wobei Y die STFT eines skalaren Signals y kennzeichnet und $\mathbf{x}$ ein Vektor ist, der sich aus den Spektralkomponenten X_i der skalaren Signale x_i , gemäß $\mathbf{x} = [X_1, X_2, \dots, X_K]$, zusammensetzt.

4.1 Zelinski 1988 (ZEL88)

4.1.1 Herleitung

Der in [8] von Zelinski beschriebene Postfilteralgorithmus basiert auf der Annahme einer schwachen Kohärenz zwischen den verrauschten Eingangssignalen eines Mikrofonarrays. Da ein perfektes, inkohärentes Rauschfeld selten vorliegt, kann eine schwache Kohärenz unter Annahme des, viel häufiger vorkommenden, diffusen Rauschfeldes mit Hilfe von entsprechend große Mikrofonabstände simuliert werden. Wie schon in Abschnitt 2.6.2 erwähnt, nimmt die Kohärenzfunktion für Frequenzen von $f > \frac{c}{2l_{ij}}$ nur mehr niedrige Werte an.

Für die folgende Herleitung wird das Modell eines inkohärenten Rauschfeldes $\mathbf{\Gamma}_{nn} = \mathbf{I} = \mathbf{I}^{-1}$ angenommen. Sind außerdem die Rauschleistungsdiestespektren an allen Mikrofonen identisch

$$\phi_{n_i n_i} = \phi_{nn} , \quad \forall i, \quad (4.4)$$

dann ergibt sich die spektrale Rauschkorrelationsmatrix zu

$$\mathbf{\Phi}_{nn} = \phi_{nn} \mathbf{\Gamma}_{nn} = \phi_{nn} \mathbf{I}. \quad (4.5)$$

Zelinski geht bei seiner Herleitung nicht von der allgemeinen Filterlösung der Postfilterstruktur Gl. 2.19 aus, sondern benutzt eine modifizierte Version:

$$\mathbf{w}_{\text{ZEL88}} = \underbrace{\left[\frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} \right]}_{H_{\text{ZEL88}}} \underbrace{\frac{\mathbf{\Phi}_{nn}^{-1} \mathbf{d}}{\mathbf{d}^H \mathbf{\Phi}_{nn}^{-1} \mathbf{d}}}_{\mathbf{b}_{\text{mvdr}}}. \quad (4.6)$$

Der Postfilterterm H_{ZEL88} berechnet sich nun nicht aus den Leistungsdiestespektren von Sprachsignal und Rauschen am Ausgang des Beamformers, sondern wird über die Leistungsdiestespektren an den Mikrofoneingängen geschätzt. In Abschnitt 4.2 und in [14] wird gezeigt, dass dieser Ansatz zu einer Überschätzung der spektralen Rauschleistungsdieste führt. Durch Einsetzen von Gl. 4.5 in Gl. 4.6 lässt sich $\mathbf{w}_{\text{ZEL88}}$ in einen DSB, gefolgt von dem modifizierten, einkanaligen Wienerfilter, gemäß

$$\mathbf{w}_{\text{ZEL88}} = \underbrace{\left[\frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} \right]}_{H_{\text{ZEL88}}} \underbrace{\frac{\mathbf{d}}{K}}_{\text{DSB}}, \quad (4.7)$$

überführen.

Die Schätzung der LDS von Sprachsignal und Rauschen zur Berechnung von H_{ZEL88} erfolgt anhand der verrauschten Eingangssignale, deren Laufzeiten zuvor ausgeglichen wurden. Mit Hilfe der Spektralkomponenten

$$\tilde{X}_i = d_i^* (S d_i + N_i) = S + d_i^* N_i , \quad \forall i. \quad (4.8)$$

lassen sich die Auto- und Kreuzleistungsdichtespektren der Eingangssignale mit Laufzeitkompensation nach Gl. 4.1 durch

$$\phi_{\tilde{x}_i \tilde{x}_i} = \phi_{ss} + \phi_{n_i n_i} + d_i \phi_{sn_i} + d_i^* \phi_{n_i s}, \quad \forall i, \quad \text{und} \quad (4.9)$$

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss} + d_i^* d_j \phi_{n_i n_j} + d_j \phi_{sn_j} + d_i^* \phi_{n_i s}, \quad \forall i \neq j, \quad (4.10)$$

anschreiben.

Zusätzlich zur Annahme eines inkohärenten Rauschfeldes Gl. 4.5, wird vorausgesetzt, dass das Sprachsignal und die Rauschsignale unkorreliert sind:

$$\phi_{n_i s} = 0 \quad \text{bzw.} \quad \phi_{sn_i} = 0, \quad \forall i. \quad (4.11)$$

Dadurch reduzieren sich die Leistungsdichtespektren von Gl. 4.9 und Gl. 4.10 zu

$$\phi_{\tilde{x}_i \tilde{x}_i} = \phi_{ss} + \phi_{nn} \quad (4.12)$$

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss}. \quad (4.13)$$

Diese Gleichungen zeigen im Vergleich mit Gl. 4.7, dass das Postfilter H_{ZEL88} allein anhand der verrauschten Eingangssignale mit vorherigem Laufzeitausgleich geschätzt werden kann. Die Leistungsdichtespektren werden mit der rekursiven Welch-Formel Gl. 2.26 bzw. Gl. 2.27 berechnet. Um die Schätzung robuster zu machen, werden die Leistungsdichtespektren über alle möglichen Mikrofonkombinationen $i > j$ gemittelt. Zumal das Leistungsdichtespektrum des Sprachsignals notwendigerweise real und positiv ist, wird der Realteil von $\phi_{\tilde{x}_i \tilde{x}_j}$ für die Schätzung verwendet. Der Schätzer für das Postfilter von Zelinski ergibt sich zu

$$\hat{H}_{\text{ZEL88}} = \frac{\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \Re \left\{ \hat{\phi}_{\tilde{x}_i \tilde{x}_j} \right\}}{\frac{1}{K} \sum_{i=1}^K \hat{\phi}_{\tilde{x}_i \tilde{x}_i}}. \quad (4.14)$$

4.1.2 Interpretation

Durch die schon erwähnte Überschätzung der Rauschleistungsdichte erreicht man zwar eine bessere Rauschunterdrückung, allerdings auf Kosten einer stärkeren Verzerrung des Wunschsingals. Der Algorithmus von Zelinski zeichnet sich durch eine einfache und effiziente Implementierbarkeit aus, welche mit einer geringen Komplexität auskommt. Abschnitt 4.6.3 zeigt eine zusätzliche Möglichkeit zur Verringerung der Rechenkomplexität bei der Berechnung des Postfilters $\hat{H}_{\text{ZEL88}}$ Gl. 4.14. Die Postfilterstruktur arbeitet sehr gut bei Vorhandensein eines inkohärenten Rauschfeldes. Diese Annahme ist aber nur für große Mikrofonabstände hinreichend erfüllt. Für alle anderen Rauschfeld- und Arraykonfigurationen ist eine schlechte Rauschunterdrückung bei niedrigen Frequenzen zu erwarten, da als Beamformer ein DSB zum Einsatz kommt (siehe die zugehörige Richtcharakteristik in Abb. 3.5 auf Seite 26). In dieser Arbeit wurde daher für diffuse Rauschfelder das Postfilter $\hat{H}_{\text{ZEL88}}$ in Kombination mit einem

SDB verwendet. Diese Struktur ist zwar im Sinne der MMSE Lösung nicht optimal, liefert aber vor allem bei niedrigen Frequenzen ganz gute Ergebnisse.

In [8] zeigt Zelinski die Lösung im Zeitbereich und eine Post-Processing Methode, welche die Eigenschaften des Postfilters und damit die Rauschunterdrückung weiter verbessern soll. In dieser Arbeit wurde aufgrund der einfacheren Implementierbarkeit die Lösung für den Frequenzbereich adaptiert, wodurch es aber in keiner Weise zur Einschränkung der Leistungsfähigkeit des Algorithmus kommt. Auf die Post-processing Methode wurde nicht näher eingegangen, da sie ausschließlich für große Mikrofonabstände Wirkung zeigt, und es Ziel dieser Arbeit ist die Arraygeometrien möglichst gering zu halten.

Eine ausführliche Analyse dieses Postfilters findet man in [4].

4.2 Simmer 1992 (SIM92)

4.2.1 Herleitung

Simmer und Wasiljeff präsentieren in [14] einen Postfilteralgorithmus, der ebenfalls auf der Annahme einer schwachen Kohärenz zwischen den verrauschten Eingangssignalen basiert. Für die folgende Herleitung wird daher wieder das Modell eines inkohärenten Rauschfeldes mit identischen Rauschleistungsdichtespektren an allen Mikrofonen,

$$\Phi_{nn} = \phi_{nn} \mathbf{\Gamma}_{nn} = \phi_{nn} \mathbf{I}, \quad (4.15)$$

verwendet. Setzt man nun Gl. 4.15 in die allgemeine Lösung für die Postfilterstruktur Gl. 2.16 ein, dann erhält man den von Simmer und Wasiljeff beschriebenen Algorithmus

$$\mathbf{w}_{\text{SIM92}} = \frac{\phi_{ss}}{\phi_{ss} + (\mathbf{d}^H \phi_{nn}^{-1} \mathbf{I}^{-1} \mathbf{d})^{-1}} \frac{\phi_{nn}^{-1} \mathbf{I}^{-1} \mathbf{d}}{\mathbf{d}^H \phi_{nn}^{-1} \mathbf{I}^{-1} \mathbf{d}} = \underbrace{\frac{\phi_{ss}}{\phi_{ss} + \frac{1}{K} \phi_{nn}}}_{H_{\text{SIM92}}} \underbrace{\frac{\mathbf{d}}{K}}_{\text{DSB}}, \quad (4.16)$$

der sich aus einem gewöhnlichen DSB und dem einkanaligem Postfilter

$$H_{\text{SIM92}} = \frac{\phi_{ss}}{\phi_{ss} + \frac{1}{K} \phi_{nn}} \quad (4.17)$$

zusammensetzt. Im Sinne des gewählten MMSE Ansatzes ist diese Lösung optimal für inkohärente Rauschfelder, liefert jedoch für alle anderen Rauschsituationen eine suboptimale Lösung.

Für die praktische Implementierung muss eine Schätzung für den Postfilterterm H_{SIM92} gefunden werden. Dafür werden die Spektralkomponenten der verrauschten Eingangssignale mit Laufzeitausgleich berechnet

$$\tilde{X}_i = d_i^* (S d_i + N_i) = S + d_i^* N_i, \quad \forall i \quad (4.18)$$

und daraus die Kreuzleistungsdichtespektren zwischen den Mikrofonkanälen i und j

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss} + d_i^* d_j^* \phi_{n_i n_j} + d_j \phi_{s n_j} + d_i^* \phi_{n_i s}, \quad \forall i \neq j \quad (4.19)$$

ermittelt. Mit Hilfe des Ausgangssignals des DSB

$$Y_b = \mathbf{b}^H \mathbf{x} = \frac{\mathbf{d}^H}{K} (S\mathbf{d} + \mathbf{n}) = S + \frac{1}{K} \mathbf{d}^H \mathbf{n} \quad (4.20)$$

wird das Leistungsdichtespektrum des Signals am Beamformerausgang

$$\phi_{y_b y_b} = \phi_{ss} + \frac{1}{K^2} \mathbf{d}^H \mathbf{\Phi}_{nn} \mathbf{d} + \frac{1}{K} \mathbf{d}^H \phi_{ns} + \frac{1}{K} \mathbf{d} \phi_{ns}^H \quad (4.21)$$

kalkuliert.

Zusätzlich zur Annahme eines inkohärenten Rauschfeldes wird von der Unkorreliertheit von Sprachsignal und Rauschsignalen

$$\phi_{ns} = \mathbf{0} \quad (4.22)$$

ausgegangen, wodurch sich Gl. 4.19 und Gl. 4.21 zu

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss} \quad (4.23)$$

$$\phi_{y_b y_b} = \phi_{ss} + \frac{1}{K^2} \mathbf{d}^H \mathbf{\Phi}_{nn} \mathbf{d} = \phi_{ss} + \frac{1}{K} \phi_{nn} \quad (4.24)$$

reduzieren.

Betrachtet man Gl. 4.23 und Gl. 4.24, dann wird klar, dass der Zähler des Postfilters in Gl. 4.17 anhand der Kreuzleistungsdichtespektren der Eingangssignale mit Laufzeitkompensation $\phi_{\tilde{x}_i \tilde{x}_j}$ und der Nenner des Postfilters anhand des Leistungsdichtespektrums des Signals am Beamformerausgang $\phi_{y_b y_b}$ geschätzt werden können. Die Berechnung der Leistungsdichtespektren erfolgt mit Hilfe der rekursiven Formel von Welch Gl. 2.26 bzw. Gl. 2.27. Wie bei Zelinski, werden nur die Realteile der Kreuzleistungsdichtespektren für die Schätzung verwendet. Eine entsprechende Robustheit des Schätzers kann durch Mittelung der Kreuzleistungsdichtespektren $\phi_{\tilde{x}_i \tilde{x}_j}$ über alle möglichen Mikrofonkanalkombinationen $i > j$ erreicht werden. Damit erhält man als Schätzer für das Postfilter von Simmer und Wasiljeff

$$\hat{H}_{\text{SIM92}} = \frac{\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \Re \left\{ \hat{\phi}_{\tilde{x}_i \tilde{x}_j} \right\}}{\hat{\phi}_{y_b y_b}}. \quad (4.25)$$

4.2.2 Interpretation

Vergleicht man das Postfilter von Zelinski H_{ZEL88} in Gl. 4.7 mit dem Postfilter von Simmer und Wasiljeff Gl. 4.17, dann unterscheiden sich die Rauschleistungsdichten im Nenner um den

Faktor K . Da es sich bei dem Postfilter von Simmer und Wasiljeff um ein optimales Filter für ein inkohärentes Rauschfeld im Sinne der allgemeinen MMSE Lösung handelt, konnten die beiden Autoren somit zeigen, dass das Postfilter von Zelinski zu einer Überschätzung der Rauschleistungsdichte um den Faktor K führt.

Dieses Postfilter zeichnet sich durch eine einfache und effiziente Implementierbarkeit aus. Abschnitt 4.6.3 zeigt außerdem eine zusätzliche Möglichkeit zur Verringerung der Rechenkomplexität bei der Berechnung des Postfilters $\hat{H}_{\text{SIM92}}$ Gl. 4.25. Die Rauschunterdrückung ist nicht ganz so stark, wie bei dem Postfilter von Zelinski, allerdings kommt es auch zu keiner nennenswerten Sprachsignalverzerrung. Aufgrund der Annahme einer schwachen Kohärenz sind große Mikrofonabstände notwendig, damit der Beamformer zusammen mit dem Postfilter auch bei niedrigen Frequenzen gute Ergebnisse erzielt. Für diffuse Rauschfelder wurde deshalb in dieser Arbeit die Kombination mit einem SDB gewählt.

Eine ausführliche Analyse dieses Postfilters findet man in [4].

4.3 McCowan 2003 (GMCC und MCC03)

4.3.1 Herleitung

Ein inkohärentes Rauschfeld kommt in der Praxis selten vor. McCowan und Bourlard stellten deshalb in [15] und [16] ein Postfilter vor, das auf einem exakteren Modell eines Rauschfeldes basiert. Aus diesem Grund wird die komplexe Kohärenzfunktion Gl. 2.30 in die Schätzung des Postfilters einbezogen. In Folge dessen, ergibt sich eine Postfilterstruktur, die nicht auf ein bestimmtes Rauschfeld beschränkt ist.

McCowan und Bourlard verwenden, wie Zelinski, eine modifizierte Version der allgemeinen Postfilterstruktur Gl. 2.19

$$\mathbf{w}_{\text{GMCC}} = \underbrace{\left[\frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} \right]}_{H_{\text{GMCC}}} \underbrace{\frac{\mathbf{\Gamma}_{nn}^{-1} \mathbf{d}}{\mathbf{d}^H \mathbf{\Gamma}_{nn}^{-1} \mathbf{d}}}_{\mathbf{b}_{\text{mvdr}}}, \quad (4.26)$$

wobei für die Rauschkorrelationsmatrix $\mathbf{\Phi}_{nn}$ eine durch die komplexe Kohärenzfunktion definierte Kohärenzmatrix $\mathbf{\Gamma}_{nn}$ eingesetzt wurde. Das Postfilter H_{GMCC} berechnet sich also direkt aus den Leistungsdichtespektren von Sprachsignal und Rauschen und nicht aus den Leistungsdichtespektren von Sprachsignal und Rauschen am Ausgang des Beamformers. Dies entspricht zwar nicht der exakten Filterlösung im Sinne von MMSE, ermöglicht jedoch vor allem im Bezug auf den Rechenaufwand eine geringe Komplexität und eine einfache Schätzung des Postfilters.

Der Algorithmus von McCowan und Bourlard setzt sich somit aus einem MVDR Beamformer gefolgt von einem einkanaligen Postfilter zusammen.

Zur Schätzung des Postfilters H_{GMCC} werden die Spektralkomponenten der verrauschten Eingangssignale mit Laufzeitausgleich

$$\tilde{X} = d_i^* (Sd_i + N_i) = S + d_i^* N_i, \quad \forall i \quad (4.27)$$

verwendet. Daraus lassen sich die Autoleistungsdichtespektren

$$\phi_{\tilde{x}_i \tilde{x}_i} = \phi_{ss} + \phi_{n_i n_i} + d_i \phi_{s n_i} + d_i^* \phi_{n_i s}, \quad \forall i \quad (4.28)$$

und die Kreuzleistungsdichtespektren

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss} + d_i^* d_j \phi_{n_i n_j} + d_j \phi_{s n_j} + d_i^* \phi_{n_i s}, \quad \forall i \neq j \quad (4.29)$$

herleiten. Zusätzlich wird die komplexe Kohärenzfunktion

$$\Gamma_{n_i n_j} = \frac{\phi_{n_i n_j}}{\sqrt{\phi_{n_i n_i} \phi_{n_j n_j}}} \quad (4.30)$$

einbezogen.

Mit der Annahme

- der Unkorreliertheit zwischen Sprachsignal und Rauschen, $\phi_{n_i s} = 0$ bzw. $\phi_{s n_i} = 0$, $\forall i$,
- und identischer Rauschleistungsdichten an allen Mikrofonen, $\phi_{n_i n_i} = \phi_{nn}$, $\forall i$,

reduzieren sich die obigen Gleichungen 4.28, 4.29 und 4.30 zu

$$\phi_{\tilde{x}_i \tilde{x}_i} = \phi_{ss} + \phi_{nn} \quad (4.31)$$

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{ss} + d_i^* d_j \phi_{n_i n_j} \quad (4.32)$$

$$\Gamma_{n_i n_j} = \frac{\phi_{n_i n_j}}{\phi_{nn}}. \quad (4.33)$$

Löst man diese 3 Gleichungen mit ihren 3 Unbekannten für ϕ_{ss} , dann erhält man einen Schätzer für das Sprachleistungsdichtespektrum und damit für den Zähler des Postfilters H_{GMCC} :

$$\hat{\phi}_{ss}^{(ij)} = \frac{\Re \left\{ \hat{\phi}_{\tilde{x}_i \tilde{x}_j} \right\} - \frac{1}{2} \Re \left\{ d_i^* d_j \Gamma_{n_i n_j} \right\} \left(\hat{\phi}_{\tilde{x}_i \tilde{x}_i} + \hat{\phi}_{\tilde{x}_j \tilde{x}_j} \right)}{\left(1 - \Re \left\{ d_i^* d_j \Gamma_{n_i n_j} \right\} \right)}. \quad (4.34)$$

Um die Robustheit zu verbessern, wird das Mittel der Leistungsdichtespektren $\phi_{\tilde{x}_i \tilde{x}_i}$ und $\phi_{\tilde{x}_j \tilde{x}_j}$ verwendet. Da ϕ_{ss} real und positiv sein muss, wird nur der Realteil der Schätzung verwendet. Die Schätzung der Leistungsdichtespektren erfolgt wiederum mit Hilfe der Formel von Welch Gl. 2.26 bzw. Gl. 2.27.

Der Nenner des Postfilters kann, wie bei der Technik von Zelinski, anhand der Leistungsdichtespektren der Eingangssignale mit Laufzeitkompensation Gl. 4.31 geschätzt werden. Durch

Mittelung über alle möglichen Mikrofonkombination $i > j$ erhält man den Schätzer für das Postfilter

$$\hat{H}_{\text{GMCC}} = \frac{\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \Re \left\{ \hat{\phi}_{ss}^{(ij)} \right\}}{\frac{1}{K} \sum_{i=1}^K \hat{\phi}_{\tilde{x}_i \tilde{x}_i}}. \quad (4.35)$$

4.3.2 Interpretation

Es ist zu beachten, dass für die Berechnung des Schätzers $\hat{\phi}_{ss}^{(ij)}$ in dieser Arbeit ein etwas anderer Ansatz gewählt wurde als von McCowan und Bourlard. Die Herleitung des Algorithmus von McCowan und Bourlard vernachlässigt die Einbeziehung des Steuervektors $\mathbf{d}$ in Gl. 4.34, beziehungsweise setzt diesen gleich $\mathbf{d} = \mathbf{1}^1$. Diese Annahme ist allerdings nur für die Broadside-Konfiguration gültig, da bei dieser Konfiguration (unter Beachtung der Fernfeldnäherung) kein Laufzeitunterschied zwischen den Eingangssignalen auftritt. Daraus ergibt sich der in [16] beschriebene Schätzer für das LDS des Sprachsignals

$$\hat{\phi}_{ss}^{(ij)} = \frac{\Re \left\{ \hat{\phi}_{\tilde{x}_i \tilde{x}_j} \right\} - \frac{1}{2} \Re \left\{ \Gamma_{n_i n_j} \right\} \left(\hat{\phi}_{\tilde{x}_i \tilde{x}_i} + \hat{\phi}_{\tilde{x}_j \tilde{x}_j} \right)}{\left(1 - \Re \left\{ \Gamma_{n_i n_j} \right\} \right)}. \quad (4.36)$$

Das Postfilter, welches sich aus diesem Schätzer berechnet, wird im Weiteren als *MCC03*-Postfilter bezeichnet. Für die Broadside-Konfiguration ergeben also das *GMCC*-Postfilter und das *MCC03*-Postfilter die gleiche Funktion. Verwendet man diesen Algorithmus für andere Einfallswinkel als für Broadside, zeigt er eine starke Rauschunterdrückung. Diese folgt aus der Überschätzung des Terms $\frac{1}{2} \Re \left\{ \Gamma_{n_i n_j} \right\} \left(\hat{\phi}_{\tilde{x}_i \tilde{x}_i} + \hat{\phi}_{\tilde{x}_j \tilde{x}_j} \right)$ in Gl. 4.36, wobei aber dementsprechend starke Signalverzerrungen auftreten.

Beide Postfilterstrukturen sind so allgemein definiert, dass sich beliebige Modelle für Rauschfelder in Gl. 4.34 bzw. Gl. 4.36 einsetzen lassen. Je genauer das gewählte Kohärenzmodell mit dem aktuellen Rauschfeld übereinstimmt, desto optimaler arbeitet das Postfilter.

Setzt man das Modell eines inkohärenten Rauschfeldes $\mathbf{\Gamma}_{nn} = \mathbf{I}$ in Gl. 4.34 bzw. Gl. 4.36 ein, dann erhält man

$$\hat{\phi}_{ss}^{(ij)} = \hat{\phi}_{\tilde{x}_i \tilde{x}_j}. \quad (4.37)$$

Somit ergibt sich das Postfilter *ZEL88* als Spezialfall des Postfilters von McCowan und Bourlard. Die Komplexität des McCowan-Algorithmus erhöht sich dabei nur um vier zusätzliche Vektoroperationen für jedes Paar (i, j) verglichen mit der Methode von Zelinski.

Um ein optimale Postfilterstruktur für ein diffuses Rauschfeld zu erhalten, werden die Kohärenzwerte eines diffusen Rauschfeldes

$$\Gamma_{n_i n_j} = \text{sinc} \left(\frac{2\pi f}{c} l_{ij} \right) \quad (4.38)$$

¹ $\mathbf{1}$ ist ein Einsvektor bestehend aus K Elementen.

in Gl. 4.34 bzw. Gl. 4.36 eingesetzt. Benutzt man diese zusätzlich für den Beamformer, dann transformiert sich der MVDR Beamformer zum superdirektiven Beamformer, der für ein solches Rauschfeld optimal ist. Das diffuse Rauschmodell wurde in dieser Arbeit, sowie von McCowan und Boulard in [16], zur Analyse dieses Algorithmus herangezogen. Durch den SDB kann man eine gute Rauschunterdrückung auch bei niedrigen Frequenzen erwarten. Da in die Schätzung der Postfilterfunktion auch die Beschaffenheit des Rauschfeldes eingeht, wird zusätzlich eine starke Verbesserung des SNR erreicht.

Eine unbestimmte Lösung von $\hat{\phi}_{ss}^{(ij)}$ Gl. 4.34 bzw. 4.36 ergibt sich, falls für die Kohärenzfunktion $\Gamma_{n_i n_j} = 1$ für $\forall i \neq j$ ist. In der Praxis kann dieses Problem gelöst werden, indem die Kohärenzmatrix $\mathbf{\Gamma}_{nn}$ regularisiert wird (siehe Abschnitt 3.4). Dabei wird jene Regularisierungsmethode verwendet, bei der alle Elemente der Kohärenzmatrix mit Ausnahme der Hauptdiagonalen durch den Faktor $1+\mu$ dividiert werden. Der Parameter μ wird üblicherweise zwischen -10 dB und -40 dB gewählt.

4.4 APES

4.4.1 Herleitung

Ein weitere Kombination von SDB und adaptiven Postfilter zur Unterdrückung eines diffusen Rauschfeldes ist ein Rauschunterdrückungsverfahren namens „Adaptive Post-filter Extension for Superdirective Beamformers (APES)“. Es wurde von Bitzer, Simmer und Kammeyer in [17] vorgestellt und basiert auf dem *SIM92*-Algorithmus (siehe Abschnitt 4.2), sowie einer alternativen Implementierung des SDB mit Hilfe eines *Generalized Sidelobe Cancellers*. Auf die alternative Implementierung des SDB wird in dieser Arbeit nicht näher eingegangen. Eine ausführliche Beschreibung findet man in [18]. Stattdessen wird die herkömmliche Implementierung aus Abschnitt 3.3.3 verwendet, welche die Rechenkomplexität nicht nennenswert erhöht. Abbildung 4.1 zeigt ein Blockdiagramm des *APES*-Algorithmus. Die Struktur besteht grundsätzlich aus einem SDB gefolgt von zwei Postfilter H_1 und H_2 . Da aber die Schätzung des ersten Postfilter H_1 mit Hilfe der *SIM92*-Methode erfolgt, muss auch ein DSB implementiert werden. Die Schätzung für das erste Postfilter lautet

$$\hat{H}_1 = \frac{\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \Re \left\{ \hat{\phi}_{\tilde{x}_i \tilde{x}_j} \right\}}{\hat{\phi}_{zz}}. \quad (4.39)$$

Der Schätzer berechnet sich aus den Kreuzleistungsdichtespektren der verrauschten Eingangssignale mit Laufzeitkompensation, sowie aus dem LDS $\hat{\phi}_{zz}$ des Ausgangssignals des DSB.

Das zweite Postfilter H_2 kann mit

$$\hat{H}_2 = \frac{\hat{\phi}_{y_b y_b}}{\hat{\phi}_{zz}} \quad (4.40)$$

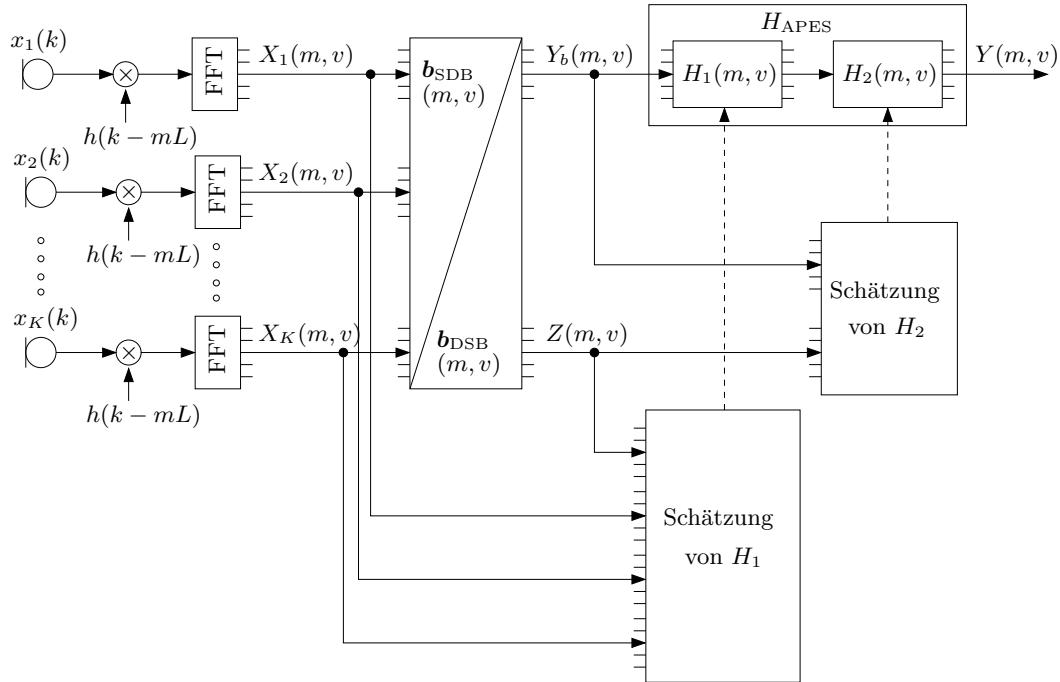


Abbildung 4.1: Blockdiagramm des APES-Algorithmus im Frequenzbereich.

geschätzt werden, wobei $\hat{\phi}_{y_b y_b}$ das LDS am Ausgang des SDB ist.

Die beiden Postfilter können durch Multiplikation zu

$$\hat{H}_{\text{APES}} = \hat{H}_1 \cdot \hat{H}_2 \quad (4.41)$$

kombiniert werden. Die Schätzung der Leistungsdichtespektren erfolgt mit Hilfe der Formel von Welch Gl. 2.26 bzw. Gl. 2.27.

4.4.2 Interpretation

Das erste Postfilter H_1 bewirkt bei kleinen Arraygeometrien und niedrigen Frequenzen in einem diffusen Rauschfeld keine ausreichende Rauschunterdrückung. Daher ist die Erweiterung mit dem zweiten Postfilter H_2 notwendig. Da der DSB und der SDB bei hohen Frequenzen ähnliche Eigenschaften aufweisen, nimmt die Übertragungsfunktion des zweiten Postfilters H_2 bei diesen Frequenzen Werte nahe 1 an, und führt daher auch zu keiner zusätzlichen Rauschunterdrückung. Bei niedrigen Frequenzen tendiert die Übertragungsfunktion des zweiten Postfilters zu Werten nahe Null. Dies liegt daran, dass der SDB im Gegensatz zum DSB bei den niedrigen Frequenzen räumlich korreliertes, diffuses Rauschen unterdrückt. Der folgende Abschnitt 4.6.3 zeigt außerdem eine Möglichkeit zur Verringerung der Rechenkomplexität für die Berechnung des Postfilters H_1 Gl. 4.39.

4.5 APAB

4.5.1 Herleitung

In [2] wurde ein anderer Ansatz zur Herleitung des Postfilters gewählt. Die Autoren Simmer, Bitzer und Marro nennen dieses Postfilter „Adaptive Post-Filter for an Arbitrary Beamformer“ (APAB).

Die Herleitung beruht auf einer allgemeinen Version des Zelinski Postfilters Gl. 4.14, die auch ausführlich in [4] diskutiert wurde. Für die Formulierung dieser allgemeinen Postfilterfunktion werden die, in [19] eingeführten, komplexen Shading Koeffizienten a_i verwendet. Der Unterschied zu den üblichen Beamformer Koeffizienten liegt darin, dass sie nicht für einen Laufzeitausgleich, sondern nur für eine reine Gewichtung der Eingangssignale sorgen. Die verrauschten Eingangssignale werden schon vor dem Beamformer einer Laufzeitkompensation unterzogen. Das verallgemeinerte Zelinski Postfilter lässt sich demnach mit Hilfe der Auto- und Kreuzleistungsdichtespektren der kompensierten, verrauschten Eingangssignale gemäß

$$H_{\text{GZEL}} = \frac{\Re \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^K a_i a_j^* \phi_{\tilde{x}_i \tilde{x}_j} \right\} \sum_{i=1}^K |a_i|^2}{\Re \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^K a_i a_j^* \right\} \sum_{i=1}^K |a_i|^2 \phi_{\tilde{x}_i \tilde{x}_i}}. \quad (4.42)$$

formulieren. Um eine robuste Schätzung zu erhalten, wurden die LDS über alle Mikrofonpaare $i > j$ anhand der Shading Koeffizienten gemittelt. Wählt man für eine gleichmäßige Gewichtung der Eingangssignale die Shading Koeffizienten mit $a_i = \frac{1}{K}, \forall i$, dann erhält man den *ZEL88*-Algorithmus aus Gl. 4.14.

Zur Vereinfachung der Herleitung wird als Steuervektor $\mathbf{d} = \mathbf{1}$ angenommen. Folglich gilt für die Auto- und Kreuzleistungsdichtespektren der Eingangssignale mit Laufzeitausgleich:

$$\phi_{\tilde{x}_i \tilde{x}_j} = \phi_{x_i x_j}, \forall i \neq j \quad \text{und} \quad \phi_{\tilde{x}_i \tilde{x}_i} = \phi_{x_i x_i}, \forall i. \quad (4.43)$$

Zur Schätzung der Postfilterfunktion können also die verrauschten Eingangssignale direkt verwendet werden.

Um Gl. 4.42 in Matrizenform anschreiben zu können, werden die Auto- und Kreuzleistungsdichtespektren der Eingangssignale zur spektralen Korrelationsmatrix $\Phi_{\mathbf{x}\mathbf{x}}$ und die Shading Koeffizienten zum Vektor $\mathbf{a}$ zusammengefasst. Mit Hilfe von

$$\Re \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^K a_i a_j^* \phi_{x_i x_j} \right\} = \sum_{i=1}^K \sum_{j=1}^K a_i a_j^* \phi_{x_i x_j} - \sum_{i=1}^K a_i a_i^* \phi_{x_i x_i} \quad (4.44)$$

lautet Gl. 4.42 in Matrizenschreibweise

$$H_{\text{GZEL}} = \frac{(\mathbf{a}^H \Phi_{\mathbf{x}\mathbf{x}} \mathbf{a} - \mathbf{a}^H \Phi_{\mathbf{x}\mathbf{x}}^D \mathbf{a}) \mathbf{a}^H \mathbf{a}}{(\mathbf{a}^H \mathbf{1} \mathbf{1}^H \mathbf{a} - \mathbf{a}^H \mathbf{a}) \mathbf{a}^H \Phi_{\mathbf{x}\mathbf{x}}^D \mathbf{a}}, \quad (4.45)$$

wobei Φ_{xx}^D die Diagonalmatrix aus den Elementen der Hauptdiagonale von Φ_{xx} besteht. Ist die Eingangsleistung aller Mikrofonen identisch, $\Phi_{xx}^D = \phi_{xx} \mathbf{I}$, folgt

$$H_{\text{GZEL}} = \frac{(\mathbf{a}^H \Phi_{xx} \mathbf{a} - \phi_{xx} \mathbf{a}^H \mathbf{a})}{\phi_{xx} (\mathbf{a}^H \mathbf{1} \mathbf{1}^H \mathbf{a} - \mathbf{a}^H \mathbf{a}^H)}. \quad (4.46)$$

Die Annahme der Unkorreliertheit von Sprache und Rauschen, $\Phi_{xx} = \Phi_{ss} + \Phi_{nn}$, führt zu

$$H_{\text{GZEL}} = \frac{(\mathbf{a}^H \Phi_{ss} \mathbf{a} - \phi_{ss} \mathbf{a}^H \mathbf{a}) + (\mathbf{a}^H \Phi_{nn} \mathbf{a} - \phi_{nn} \mathbf{a}^H \mathbf{a})}{(\phi_{ss} + \phi_{nn}) (\mathbf{a}^H \mathbf{1} \mathbf{1}^H \mathbf{a} - \mathbf{a}^H \mathbf{a}^H)}. \quad (4.47)$$

Aufgrund der Annahme identische Rauschleistungen an den Mikrofonen lässt sich die spektrale Rauschkorrelationsmatrix mit $\Phi_{nn} = \phi_{nn} \Gamma_{nn}$ angeben. Ist das gewünschte Sprachsignal kohärent, $\Phi_{ss} = \phi_{ss} \mathbf{1} \mathbf{1}^H$, und wurden die Shading Koeffizienten normiert, $\mathbf{a}^H \mathbf{1} \mathbf{1}^H \mathbf{a} = 1$, erhält man schließlich

$$H_{\text{GZEL}} = \frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} + \frac{\phi_{nn} (\mathbf{a}^H \Gamma_{nn} \mathbf{a} - \mathbf{a}^H \mathbf{a})}{(\phi_{ss} + \phi_{nn}) (1 - \mathbf{a}^H \mathbf{a})}. \quad (4.48)$$

Definiert man den Dämpfungsfaktor für die Rauschleistung² durch

$$A_{\Gamma} = \mathbf{a}^H \Gamma_{nn} \mathbf{a} \quad (4.49)$$

und mit $\Gamma_{nn} = \mathbf{I}$ den Dämpfungsfaktor für inkohärentes Rauschen durch

$$A_{\mathbf{I}} = \mathbf{a}^H \mathbf{a}, \quad (4.50)$$

folgt durch Substitution in Gl. 4.48 der Ausdruck

$$H_{\text{GZEL}} = \frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} + \frac{\phi_{nn} (A_{\Gamma} - A_{\mathbf{I}})}{(\phi_{ss} + \phi_{nn}) (1 - A_{\mathbf{I}})}. \quad (4.51)$$

Vor allem die Subtraktion von $A_{\mathbf{I}}$ in der Übertragungsfunktion des Postfilters Gl. 4.51 verursacht jedoch Probleme. Eine Analyse zeigt folgende Eigenschaften:

- $A_{\Gamma} = A_{\mathbf{I}}$: Die Differenz der beiden Dämpfungsfaktoren ist nur für ein inkohärentes Rauschfeld Null. In diesem Fall ergibt sich die gewohnte Lösung für das Zelinski Postfilter (siehe H_{ZEL88} , Gl. 4.7).
- $A_{\Gamma} < A_{\mathbf{I}}$: In einem diffusen Rauschfeld, aber auch beim Auftreten kohärenter Rauschquellen, kann die Differenz der beiden Dämpfungsfaktoren zu einer negativen Übertragungsfunktion führen. Solche negativen Werte werden üblicherweise Null gesetzt. Dies bewirkt eine starke Verzerrung des Sprachsignals.

²Bezüglich der obigen Annahmen entspricht der Dämpfungsfaktor für die Rauschleistung A_{Γ} für normierte Shading Koeffizienten der Inversen des Arraygewinns G^{-1} (siehe Abschnitt 3.1.2 auf Seite 20, Gl. 3.6).

- $A_I = 1$: Dieser Fall führt zu einer unbestimmten Lösung des Postfilters. Er tritt üblicherweise bei einem superdirektiven Design der Filterkoeffizienten auf, da es hierbei zu einer Verstärkung des inkohärenten Rauschens bei niedrigen Frequenzen kommt (siehe Abschnitt 3.3.4).

Um diese Probleme zu lösen, wird $A_I = 0$ substituiert. Daraus resultiert das APAB-Postfilter

$$H_{\text{APAB}} = \frac{\phi_{ss}}{\phi_{ss} + \phi_{nn}} + \frac{\phi_{nn}A_I}{(\phi_{ss} + \phi_{nn})}, \quad (4.52)$$

das sich mit Hilfe des LDS $\phi_{y_b y_b} = \phi_{ss} + \phi_{nn}A_I$ des Signals am Beamformerausgang durch

$$\hat{H}_{\text{APAB}} = \frac{\hat{\phi}_{y_b y_b}}{\hat{\phi}_{xx}} \quad (4.53)$$

schätzen lässt. $\hat{\phi}_{xx}$ kennzeichnet die Schätzung für das LDS des Eingangssignals des Mikrofons, welches sich am nächsten zur Sprachquelle befindet, oder, alternativ dazu, die Schätzung für das LDS des Signals, das sich durch Mittelung der Eingangssignale über alle Mikrofone ergibt. Letztere Methode wurde in dieser Diplomarbeit angewandt. Die Leistungsdichtespektren werden mit der Formel von Welch Gl. 2.26 bzw. Gl. 2.27 geschätzt.

Die Implementierung des APAB-Algorithmus innerhalb einer FFT-Filterbank zeigt Abbildung 4.2. Die Laufzeitkompensation und die Shading Koeffizienten wurden im Beamformfilter $\mathbf{b}_{\text{arb}}$ zusammengefasst. Die Koeffizienten des Beamformers können beliebig gewählt werden. Für die Tests in dieser Arbeit wurden die Koeffizienten eines regularisierten, superdirektiven Beamformers verwendet.

4.5.2 Interpretation

Durch Nullsetzen des Dämpfungsfaktors A_I erhält man ein Postfilter, das die obig genannten Probleme vermeidet. Das Design ist kompatibel für superdirektive Koeffizienten. Es ergeben sich bei diesem Entwurf immer positive Werte für die Übertragungsfunktion und es zeigen sich gute Eigenschaften für niederfrequentes Rauschen. Nichtsdestotrotz ist die Übertragungsfunktion eine Näherung des Wienerfilters für das Eingangssignal und vernachlässigt die Tatsache, dass das Rauschen bereits durch den MVDR Beamformer reduziert wurde. Dies wirkt sich negativ auf die Rauschunterdrückungseigenschaften des Postfilters aus.

4.6 Optimierung der Algorithmen

4.6.1 Begrenzung der Filterübertragungsfunktion

Bei allen vorgestellten Postfilteralgorithmen handelt es sich um Filter ohne Phase, deren Wertebereich zwischen 0 und 1 liegt. Vor allem bei instationärem Rauschen, aber auch durch eine große Varianz des Sprachsignals, kann es zu groben Schätzfehlern bei der Berechnung

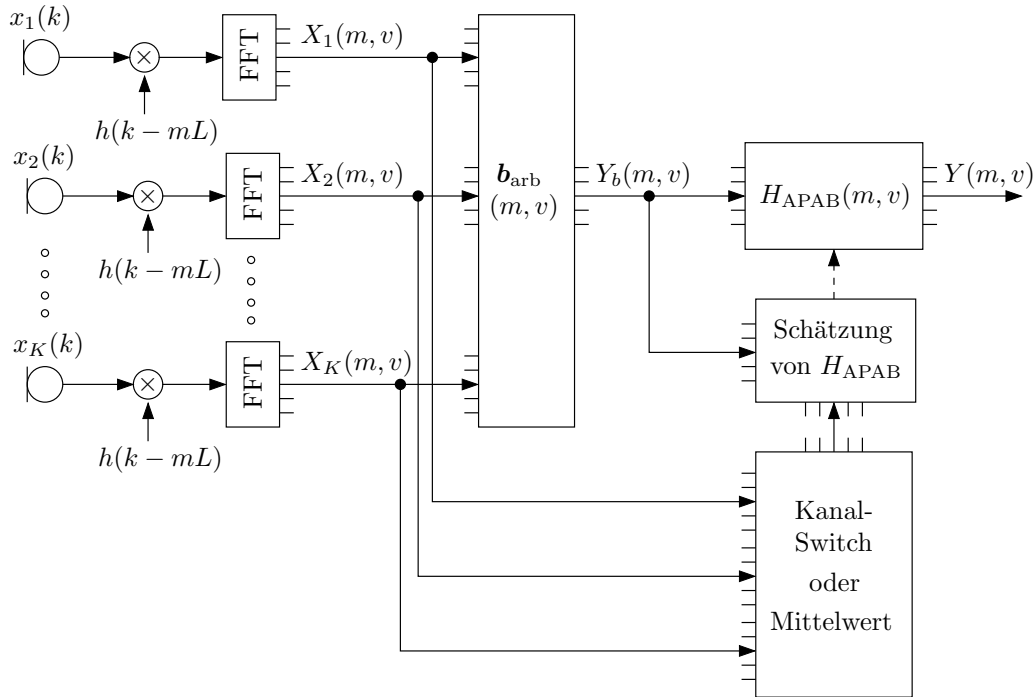


Abbildung 4.2: Blockdiagramm des APAB-Algorithmus im Frequenzbereich.

der Übertragungsfunktion des Postfilters kommen, wodurch auch negative Werte auftreten können. Daher ist es notwendig die Werte der Filterübertragungsfunktion zusätzlich zwischen

$$0.05 \leq H_{\text{post}}(m, v) \leq 1 \quad (4.54)$$

einzuschränken. Die Begrenzung der Übertragungsfunktion ist bei allen Algorithmen absolut notwendig, um deren korrekte Funktion zu gewährleisten. Die untere Grenze in Bedingung 4.54 sollte deswegen nicht Null gesetzt werden, da es ansonst bei diesen Frequenzen zu einer völligen Signalauslöschung kommt und eine starke Verzerrung des Sprachsignals zur Folge hätte. Die Wahl der unteren Grenze wird in Abschnitt 4.6.2 noch näher erläutert.

4.6.2 Maßnahmen gegen Signalverzerrungen und Musical Noise

Stark instationäres Rauschen, wie es besonders in automotiver Umgebung vorkommt, führt bei fast allen vorgestellten Algorithmen zum Auftreten von Musical Noise. Bei Musical Noise handelt es sich um gefärbtes, instationäres und für den Menschen sehr unangenehm empfundenes Rauschen. Die folgenden zwei Strategien verringern das Auftreten von Signalverzerrungen und Musical Noise:

Mindestfilterfunktion

Die Einführung einer Mindestfilterfunktion $H_{\min}$ als untere Grenze für die Filterübertragungsfunktion reduziert zu starke Verzerrungen des verarbeiteten Sprachsignals. Die Verminderung

der Signalverzerrungen wird allerdings durch das Auftreten eines Restrauschpegels „erkauft“. Dieser Restrauschpegel wird auch „Comfort Noise“ genannt. Der Restrauschpegel hat jedoch den Vorteil, dass er vorhandenes Musical Noise teilweise maskiert und dadurch gewissermaßen verringert. Die Mindestfilterfunktion wird dabei so implementiert, dass sie mit der aktuellen Filterfunktion $H_{\text{post}}(m, v)$ verglichen wird. Ist der Wert der Filterfunktion $H_{\text{post}}(m, v)$ für einen Frequenzpunkt v kleiner als der Wert der Mindestfilterfunktion $H_{\text{min}}(v)$, dann wird die Mindestfilterfunktion übernommen:

$$H_{\text{post}}(m, v) = \min \{H_{\text{min}}(v), H_{\text{post}}(m, v)\}, \quad (4.55)$$

wobei

$$\min \{x, y\} = \begin{cases} x & \text{wenn } x < y, \\ y & \text{sonst.} \end{cases} \quad (4.56)$$

In dieser Arbeit wurde eine Mindestfilterfunktion verwendet, wie sie bereits in [3] zum Einsatz kam. Es wurde ein Hochpassverlauf gemäß

$$H_{\text{min}}(f) = \begin{cases} 0.05 & \text{für } f \leq f_1, \\ (f - f_1) \frac{0.7}{f_2 - f_1} & \text{für } f_1 < f \leq f_2, \\ 0.7 & \text{für } f_2 < f \end{cases} \quad (4.57)$$

gewählt. Günstige Werte für die Grenzfrequenzen ergaben sich für $f_1 \approx 200 \text{ Hz}$ und für $f_2 \approx 4500 \text{ Hz}$. Diese Wahl des Hochpassverlaufs ist bezogen auf das vorhandene Rauschen, welches für die untersuchten Umgebungen in dieser Arbeit eine typische $\frac{1}{f}$ -Charakteristik im Spektralbereich aufweist, äußerst günstig. Bei hohen Frequenzen, bei denen das Rauschen nicht mehr so stark ist, bewirkt ein hoher Wert von H_{min} eine Anhebung des Sprachsignals und damit geringere Verzerrungen.

Adaptive Berechnung des Glättungsfaktors

Musical Noise entsteht vor allem durch eine starke Varianz der Übertragungsfunktion. Da diese von den geschätzten Leistungsdichtespektren abhängt, kann man mit einer Glättung der Leistungsdichtespektren die Fluktuationen der Übertragungsfunktion niedrig halten. Diese Glättung ist durch die Beeinflussung des Parameters α in der Formel von Welch Gl. 2.26 bzw. Gl. 2.27 möglich. Werte von α nahe 1 bewirken eine starke Glättung der LDS über die Zeit. Allerdings wird dadurch das instationäre Sprachsignal nicht mehr korrekt wiedergegeben und als Folge tritt Nachhall auf. Das Ziel ist, einen Kompromiss zwischen ausreichender Glättung und geringem Nachhall zu finden.

Eine Möglichkeit dieses Problem zu lösen, ist eine adaptive Berechnung des Parameters α , welche bereits in [3] präsentiert und aus [20] übernommen wurde:

$$\alpha(m, v) = 0.98 - 0.3 \frac{SNR(m, v)}{1 + SNR(m, v)}. \quad (4.58)$$

Für niedriges SNR nimmt α hohe Werte nahe 0.98 an. Sobald die Rauschleistung gegenüber der Sprachleistung überwiegt, werden die Leistungsdichtespektren geglättet und damit deren Varianz, die durch das stark instationäre Rauschen auftritt, niedrig gehalten. Die Folge ist begrenztes Musical Noise. Für hohes SNR, die Sprachleistung überwiegt gegenüber der Rauschleistung, nimmt α Werte nahe 0.68 an, wodurch die Schätzer den schnellen Variationen der Sprache gut folgen können.

Unter der Annahme, dass sich das SNR von einem Frame zum anderen nicht so schnell ändern wird, kann man für den Ausdruck $SNR(m, v)/(1 + SNR(m, v))$ die Näherung

$$\frac{SNR(m, v)}{1 + SNR(m, v)} \cong H_{\text{post}}(m - 1, v) \quad (4.59)$$

in Gl. 4.58 einsetzen. Mit Hilfe der Übertragungsfunktion des vorherigen Frames $H_{\text{post}}(m - 1, v)$ resultiert die Berechnung des adaptiven Parameters α in

$$\alpha(m, v) = 0.98 - 0.3 H_{\text{post}}(m - 1, v). \quad (4.60)$$

Mit dieser adaptiven Regel kann man das Musical Noise einigermaßen kontrollieren, obwohl die Verzerrungen des Signals etwas stärker hörbar werden.

4.6.3 Reduzierung der Rechenkomplexität

Die beiden Algorithmen für die Postfilter *ZEL88* und *SIM92* lassen sich durch Umformung noch weiter in ihrer Rechenkomplexität reduzieren.

Dafür berechnet man die Spektralkomponente des DSB-Ausgangs

$$Y_b = \mathbf{b}^H \mathbf{x} = \frac{\mathbf{d}^H}{K} (S\mathbf{d} + \mathbf{n}) = S + \frac{1}{K} \mathbf{d}^H \mathbf{n} = S + \frac{1}{K} \sum_{i=1}^K d_i^* N_i \quad (4.61)$$

und bildet unter der Annahme der Unkorreliertheit von Sprachsignal und Rauschen das zugehörige LDS

$$\begin{aligned} \phi_{y_b y_b} &= \phi_{ss} + \frac{1}{K^2} \sum_{i=1}^K \sum_{j=1}^K d_i^* d_j \phi_{n_i n_j} \\ &= \phi_{ss} + \frac{2}{K^2} \Re \left\{ \sum_{i=1}^{K-1} \sum_{j=i+1}^K d_i^* d_j \phi_{n_i n_j} \right\} + \frac{1}{K^2} \sum_{i=1}^K \phi_{n_i n_i}. \end{aligned} \quad (4.62)$$

Die Autoleistungsdichtespektren der verrauschten Eingangssignale mit Laufzeitausgleich sind identisch mit den Autoleistungsdichtespektren der verrauschten Eingangssignale,

$$\phi_{\tilde{x}_i \tilde{x}_i} = \phi_{x_i x_j} = \phi_{ss} + \phi_{n_i n_i}, \quad \forall i, \quad (4.63)$$

mit deren Hilfe sich der Zähler

$$\frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^K \Re \{ \phi_{\tilde{x}_i \tilde{x}_j} \} \quad (4.64)$$

des *ZEL88*-Postfilters Gl. 4.14 bzw. des *SIM92*-Postfilters Gl. 4.25 zu

$$\frac{K}{K-1} \left(\phi_{y_b y_b} - \frac{1}{K} \sum_{i=1}^K \phi_{x_i x_i} \right) \quad (4.65)$$

umformen lässt. Damit erhält man für den Schätzer des Postfilters von Zelinski

$$\hat{H}_{\text{ZEL88}} = \frac{K}{K-1} \frac{\hat{\phi}_{y_b y_b} - \frac{1}{K} \sum_{i=1}^K \hat{\phi}_{x_i x_i}}{\frac{1}{K} \sum_{i=1}^K \hat{\phi}_{x_i x_i}} \quad (4.66)$$

und für den Schätzer des Postfilters von Simmer und Wasiljeff

$$\hat{H}_{\text{SIM92}} = \frac{K}{K-1} \frac{\hat{\phi}_{y_b y_b} - \frac{1}{K} \sum_{i=1}^K \hat{\phi}_{x_i x_i}}{\hat{\phi}_{y_b y_b}}. \quad (4.67)$$

Die Rechenkomplexität für diese Schätzmethode verringert sich von $\frac{K^2-K}{2}$ zu $K+1$ notwendigen Additionen für die Berechnung des Zählers. Darüber hinaus ist für die Schätzung des Postfilters kein Laufzeitausgleich der verrauschten Eingangssignale notwendig.

Kapitel 5

Testsignale und Evaluierungsverfahren

In diesem Kapitel wird die Aufnahme der Testsignale, die zur Evaluierung der Algorithmen benötigt werden, erläutert. Es wird die Aufnahmesituation in einem Büroraum und in automotiver Umgebung dargestellt. Weiters werden objektive und subjektive Messverfahren für eine Bewertung der Sprachqualität vorgestellt.

5.1 Aufnahme der Audiosignale

Zum Testen der Algorithmen wurden zwei typische Umgebungssituationen gewählt:

- Büroraum
- Auto

Im Zuge dieser Diplomarbeit wurde eine umfangreiche Datenbank bestehend aus 540 Audiosignale erstellt. Sämtliche Rauschfelder, weibliche, sowie männliche, Sprachquellen aus verschiedenen Abständen und Winkeln zum Mikrofonarray und zahlreiche Arraykonfigurationen können damit simuliert und getestet werden. Durch Kombination der unterschiedlichen Audiosignale kann somit fast jegliche Situation in einem Büroraum nachgestellt werden. Aufgrund der großen Anzahl an möglichen Testsignalen, wird in Kapitel 6 nur auf jene Konfigurationen eingegangen, welche für die Analyse der Algorithmen besonders relevant waren.

Für die automotive Umgebung wurden die Audiosignale verwendet, die im Zuge der Arbeit von Herrn Boigner [3] erstellt wurden. Für die Aufnahmen im Büroraum sowie im Auto wurde das Array in Abbildung 5.1 verwendet. Die Haltevorrichtung für die Mikrofone hat Löcher in diskreten Abständen von 2.5 cm, in denen die Mikrofone eingesetzt und befestigt werden. Damit wird sichergestellt, dass sich nur kleine Abweichungen von den gewünschten Mikrofonabständen ergeben. Mit dieser Vorrichtung können für bis zu acht Mikrofone nahezu sämtliche Arraykonfigurationen für eine Abtastfrequenz von 8 kHz und 16 kHz realisiert

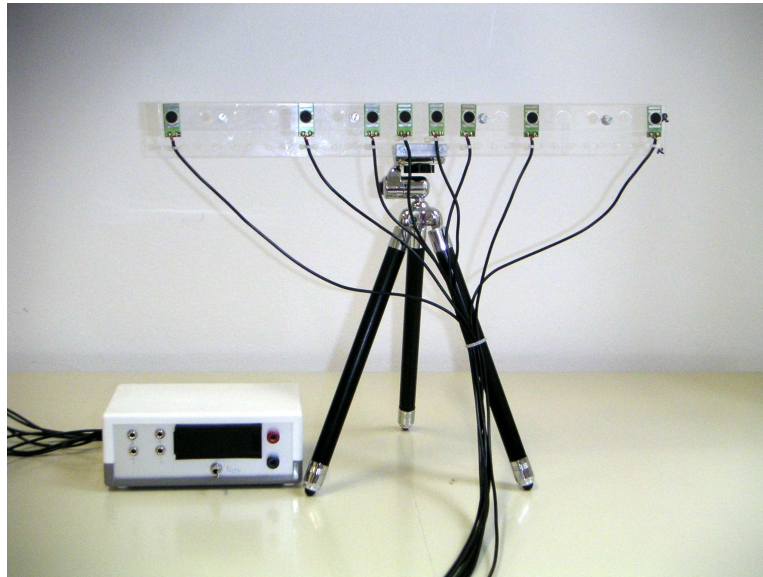


Abbildung 5.1: *Mikrofonarray mit externem Vorverstärker.*

werden. Die Mikrofone sind Elektret-Kondensator Mikrofonkapseln von *AKG Acoustics* mit der Bezeichnung Q300T. Das Gehäuse wurde entfernt, um die Abstände kleiner und genauer einstellen zu können. Abbildung 5.2 zeigt das Mikrofon im Gehäuse.



Abbildung 5.2: *Mikrofon Q300T im Gehäuse.*

Die Richtcharakteristik des Mikrofons ist omnidirektional. Der Frequenzgang des Mikrofons wird aufgrund des fehlenden Gehäuses und der fehlenden, akustischen Umwegleitungen im Gehäuse nicht exakt mit dem Datenblatt übereinstimmen¹. Jedoch spielt der Frequenzgang selbst eine untergeordnete Rolle, da nur die Abweichungen der Mikrofoncharakteristika untereinander von Bedeutung sind. Die Mikrofone besitzen einen integrierten Vorverstärker, welcher mittels Phantomspeisung über die Signalleitung versorgt wird. Die Signale wurden mittels einer externen 8-kanaligen Soundkarte auf einem PC gespeichert. Da die Signalleistungen der Mikrofone für die externe Soundkarte zu gering waren, wurden sie mittels eines externen Vorverstärkers angehoben (siehe ebenfalls Abbildung 5.1, links unten). Der externe

¹Das Datenblatt ist nicht im Internet erhältlich und kann direkt bei *AKG Acoustics* angefordert werden.

Vorverstärker wurde von einer 9 V-Batterie gespeist. Damit konnte sichergestellt werden, dass die Signale nicht durch den auftretenden Netzbrumm einer herkömmlichen Netzversorgung beeinträchtigt wurden.

5.1.1 Büroraumaufnahmen

Für die Aufnahmen in Büroraumumgebung wurden sämtliche Sprach- und Rauschsignale getrennt voneinander aufgezeichnet. Alle aufgenommenen Signale können somit additiv kombiniert werden, wodurch die gewünschte Sprach-Rausch-Situation simuliert werden kann. Es ergeben sich dadurch noch weitere Vorteile: Das Eingangs-SNR kann beliebig eingestellt werden und die getrennten Sprach- und Rauschsignale können später verwendet werden, um objektive Messungen der Sprachqualität (siehe Abschnitt 5.3) durchzuführen.

Die Testsignale wurden mit folgenden Arraykonfigurationen aufgenommen:

Mikrofonanzahl	Arraytyp	Mikrofonabstand
4	äquidistant	2.5 cm
4	äquidistant	5 cm
5	äquidistant	2.5 cm
5	äquidistant	5 cm
8	harmonisch	siehe Abb. 5.3

Abbildung 5.3 zeigt das 8-kanalige, harmonische Mikrofonarray im Detail. Bezogen auf das Auftreten von räumlichen Aliasing (siehe Abschnitt 3.2.1) sind jene Arraykonfigurationen mit einem Mikrofonabstand von 5 cm ideal für eine Abtastfrequenz von $f_s = 8$ kHz. Jene Arraystrukturen mit einem Mikrofonabstand von 2.5 cm, sowie das harmonische Array, sind für eine Abtastfrequenz von $f_s = 16$ kHz geeignet.

Um eine umfangreiche Analyse der Algorithmen zu gewährleisten, standen als Sprecher für die Sprachaufnahmen je eine weibliche und eine männliche Person zur Verfügung. Folgender Satz

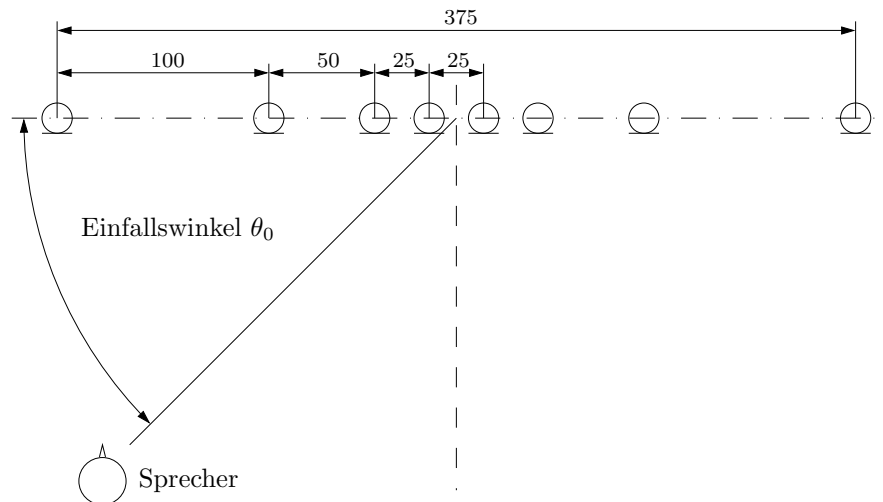


Abbildung 5.3: Mikrofonpositionierung des harmonischen 8-Kanal-Arrays.

wurde aufgenommen: „Die Relativitätstheorie befasst sich mit der Struktur von Raum und Zeit, sowie mit dem Wesen der Gravitation“. Zuerst wurde ein Prototyp des Sprachsignals in einem reflexionsarmen Raum (siehe Abschnitt 5.1.1) aufgenommen und auf einem PC gespeichert. Danach wurde für jede weitere Aufnahme die Prototyp-Sprachaufnahme über einen Lautsprecher wiedergegeben, damit garantiert werden konnte, dass immer dieselben Bedingungen für jede Aufnahme vorlagen und immer das exakt gleiche Sprachsignal aufgezeichnet wurde. Sämtliche für die Büroaufnahmen verwendeten Lautsprecher sind 2-Wege Aktivlautsprecher der Firma *Genelec* mit der Bezeichnung 1029A, die sich vor allem durch eine große Signalbandbreite auszeichnen². Da man von dem Modell der Fernfeldnäherung ausgeht, eignen sich diese Lautsprecher nicht nur für die Simulation der Rauschquellen, sondern auch für die Wiedergabe des Sprachsignals.

Aufnahme von Sprachsignalen im reflexionsarmen Raum

Ideal für die Aufnahme der Prototyp-Sprachaufnahme wäre eine „schalltote“ Umgebung, da in dieser Aufnahme weder Nachhalleffekte noch jeglicher anderer Störschall erwünscht sind. Eine solche Umgebung kann allerdings nur näherungsweise durch einen reflexionsarmen Raum realisiert werden. Für die Aufnahmen in dieser Diplomarbeit wurde der reflexionsarme Raum am *Institut für Nachrichten- und Hochfrequenztechnik (INTHFT) der TU Wien* verwendet. Alle Wände dieses Raumes, inklusive Decke und Boden, sind mit ca. 5 cm dicken, verschachtelten Platten aus Mineralwolle ausgekleidet. Die gesamte Wandverkleidung einer Wand erreicht damit eine Dicke von ca. 1 m. Durch diese schalldämmende Verkleidung werden starke Echos und lange Nachhallzeiten vermieden. Die Schallwellen werden an den Wänden dadurch möglichst wenig und möglichst diffus reflektiert. Zusätzlich wird verhindert, dass Störschall von außerhalb in den Messraum eindringen kann. Damit der Raum noch begehrbar bleibt, ist am Boden ein Gitter verlegt, welches sich akustisch möglichst neutral verhält. Der PC zur Speicherung der Signale und zur Ansteuerung des Lautsprechers befand sich außerhalb des Messraumes. Die Messanordnung und den reflexionsarmen Raum sieht man in Abbildung 5.4. Die Messanordnung in der Abbildung besteht aus folgenden Teilen:

1. Mikrofonarray
2. Lautsprecher als Sprachsignalquelle
3. externer Vorverstärker
4. schalldämmende Wandverkleidung.

Mit Hilfe des reflexionsarmen Raumes wurden für eine umfangreiche Analyse der Algorithmen alle Sprachaufnahmen auch ohne Nachhall realisiert. Dabei wurde der Lautsprecher, der als Sprachquelle diente und die Prototyp-Sprachaufnahme wiedergab, in einem Abstand von

²siehe Datenblatt, <http://www.genelec.com/pdf/OM1029a.pdf> (August 2005)



Abbildung 5.4: Messanordnung im reflexionsarmen Raum (Mikrofonarray (1), Lautsprecher (2), Vorverstärker (3), Wandverkleidung (4)).

0.7 m und danach in einem Abstand von 1.2 m vom Mikrofonarray aufgestellt. Als Einfallswinkel wurden $\theta_0 = 25^\circ, 45^\circ, 70^\circ$ und 90° gewählt. Der Abstand des Lautsprechers und des Mikrofonarrays vom Boden betrug ca. 1 m.

Aufnahme von Sprachsignalen in realer Umgebung

Sämtliche Sprachsignale wurden mit den gleichen Einstellungen, wie in Abschnitt 5.1.1, ebenso in einem herkömmlichen Büroraum aufgezeichnet. Auf diese Weise wurde den Signalen ein Nachhall, der in herkömmlichen Räumen üblich ist, hinzugefügt. Abbildung 5.5 zeigt die Messanordnung und die Maße des verwendeten Büroraums. Der PC zur Speicherung der Signale und zur Steuerung der Aufnahme befand sich außerhalb des Messraumes. Der Lautsprecher zur Wiedergabe der Prototyp-Sprachaufnahme wurde in 0.7 m und in 1.2 m Entfernung vom Mikrofonarray positioniert. Die Aufnahmen wurden in beide Positionen für die Einfallswinkel $\theta_0 = 25^\circ, 45^\circ, 70^\circ$ und 90° durchgeführt.

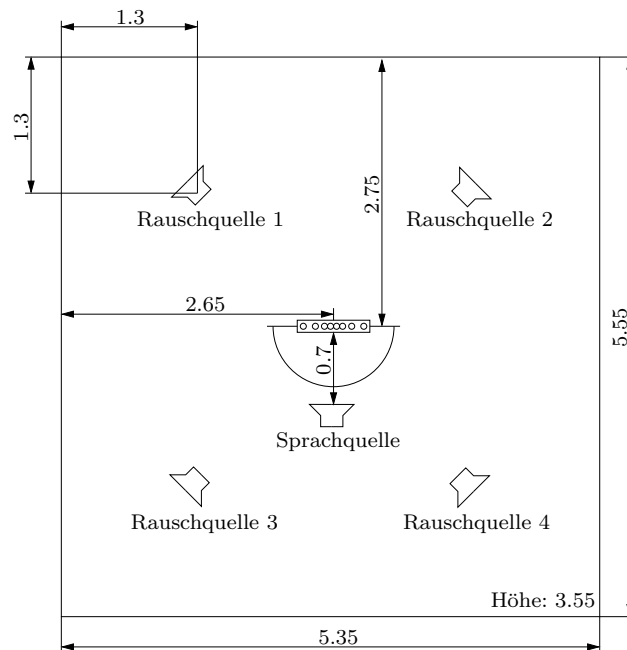


Abbildung 5.5: Messanordnung im Büroraum (Angaben in m).

Aufnahme von diffusem Rauschen in realer Umgebung

In einem Büroraum tragen viele unterschiedliche Rauschquellen zum gesamten Rauschfeld bei. Einen Großteil machen vor allem Geräusche von PCs und Klimaanlage aus. Hinzu kommen aber auch noch Tippgeräusche und mehrere im Raum verteilte, sprechende Personen. Misst man die Rauschsignale mit einem Mikrofonarray und berechnet daraus die Kohärenzfunktion zwischen zwei Mikrofonen, dann ergibt sich, wie schon in Abschnitt 2.6.2, Abbildung 2.4, gezeigt, ein $\frac{\sin(x)}{x}$ -förmiger Verlauf, der dem eines diffusen Rauschfeldes entspricht. Um ein solches Rauschfeld hinreichend zu simulieren, wurden als Rauschquellen vier Lautsprecher

in einem Büroraum gemäß Abbildung 5.5 aufgestellt. Die Lautsprecher wurden von Rauschgeneratoren gespeist, die $\frac{1}{f}$ -Rauschen (entspricht typischem Maschinenrauschen) erzeugten. Für ein möglichst homogenes Rauschfeld wurden gleiche Lautstärken für die Lautsprecher gewählt. Die Lautsprecher wurden außerdem in Richtung der Ecken des Raumes positioniert, damit sich möglichst viele Reflexionen ergaben und damit das diffuse Rauschfeld approximiert wurde. Alle Lautsprecher wurden in einer Höhe von ca. 1.5 m angebracht.

5.1.2 Autoaufnahmen

Für die Autoaufnahmen wurde die harmonische Arraygeometrie aus Abbildung 5.3 verwendet. Der männliche Sprecher befand sich am Beifahrersitz unter einem Winkel von $\theta_0 \approx 126^\circ$ in einem Abstand von ca. 58 cm vom Mittelpunkt des Mikrofonarrays. Der aufgenommene Satz lautete: „*Ein vielfacher Wunsch von Liebhabern alter Tonaufzeichnungen ist die Unterdrückung von Störungen, die der Musikaufnahme zwar eine gewisse...*“. Das Mikrofonarray wurde am Rückspiegel im Automobil befestigt. Um Vibrationen zu vermeiden, wurde eine ca. 1.5 m dicke Schicht Schaumgummi zwischen Rückspiegel und Mikrofonarray montiert. Die Aufnahmen wurden in einem Automobil der Marke *Audi*, Modell *A4* und Baujahr 1995, gemacht. Fahrgeräusche und verrauschte Sprache wurden während der Fahrt auf der Autobahn (Westautobahn Nähe Wien) bei einer Geschwindigkeit von ca. 120 km/h aufgenommen. Drei verschiedene Konfigurationen führten zu unterschiedlichen Störgeräuschen mit folgenden Bezeichnungen:

- „*Fahrgeräusche*“ – bei geschlossenem Fenster und Schiebedach.
- „*Offenes Schiebedach*“ – offenes Schiebedach (ein Spalt von 5 cm), aber geschlossene Fenster.
- „*Offenes Fenster*“ – offenes Fahrerfenster und geschlossenes Schiebedach.

Für spätere, objektive Messungen wurde das reine Sprachsignal in einer ruhigen Garage aufgezeichnet. Die Steuerung und Speicherung der Aufnahme und die Speicherung erfolgte mit Hilfe eines Laptops unter Verwendung einer externen Soundkarte.

5.2 Subjektive Analyse der Sprachqualität

Objektive Messungen reichen üblicherweise alleine nicht aus, um die Qualität der untersuchten Algorithmen ausreichend zu beurteilen. Erst die Kombination mit einer subjektiven Studie lässt einen zuverlässigen Vergleich zwischen den Algorithmen zu. Der Nachteil einer subjektiven Studie liegt in dem großen zeitlichen Aufwand, außerdem ist eine ausreichende Anzahl an Testpersonen notwendig. Die am häufigsten verwendete Methode zur Durchführung einer subjektiven Studie ist die *Category-Judgement-Methode*, die auch in dieser Arbeit angewandt wurde [21, 22]. Dabei wird die Qualität der Signale durch Kategorien beschrieben, welche eine

intuitive Bedeutung für die Testpersonen haben. Jeder Kategorie wird dabei eine Nummer zugeordnet, die in der Tabelle 5.1 ersichtlich sind.

Die Studie teilt sich in zwei Phasen:

1. **Trainingsphase.** Diese Phase ist notwendig, um den Testpersonen Referenzpunkte für ihre Bewertungen zu schaffen. Dafür werden den Testpersonen zwei Trainingssignale präsentiert. Der Testleiter definiert die Qualitätskategorie dieser Signale. Es ist sinnvoll dafür Signale zu wählen, welche die zwei Randkategorien („ausgezeichnet“ und „schlecht“) repräsentieren. Dies hilft allen Testpersonen ein gewisses „Gespür“ für die Größe des Beurteilungsbereichs zu bekommen, den sie in der Evaluierungsphase antreffen werden. Dieser Prozess wird auch als Verankerung bezeichnet.
2. **Evaluierungsphase.** In dieser Phase hören die Testpersonen die Testsignale und beurteilen sie nach ihrer allgemeinen Qualität, das heißt, wie angenehm sie von der jeweiligen Testperson empfunden werden. Die Testsignale werden in zufälliger Reihenfolge präsentiert. Es sollte eine angemessene Dauer der Testsignale und eine angemessene Zeit für die Beurteilung zwischen den Testsignalen gewählt werden. Außerdem sollten die einzelnen Lautstärken der Testsignale zuvor abgeglichen werden.

Da jede Testperson üblicherweise eine andere Vorstellung hat, wie zum Beispiel ein Signal der Kategorie „gut“ klingen soll, kann es zu großen Abweichungen der Beurteilungen unter den einzelnen Testsignalen kommen. Eine Möglichkeit diese Varianz herabzusetzen, liegt in einer sorgfältigen Auswahl der Testpersonen. Am besten eignet sich eine trainierte Hörergruppe zur Beurteilung der Testsignale. Ein trainiertes Hörerfeld besteht aus circa 10 Personen, welche mit dem Zweck des Tests vertraut sind und eine konsistente Beurteilung der Testsignale liefern.

Die Auswertung der subjektiven Studie erfolgt durch Berechnung des mittleren Meinungswertes (MOS³). Für jedes Testsignal wird der MOS nach folgender Formel berechnet:

$$MOS = \frac{(5 pa) + (4 pg) + (3 pb) + (2 pm) + ps}{\text{Gesamtzahl an Testpersonen}}. \quad (5.1)$$

Dabei bedeutet

pa die Anzahl an Testpersonen, die das Testsignal mit „ausgezeichnet“ bewerten,

pg die Anzahl an Testpersonen, die das Testsignal mit „gut“ bewerten,

pb die Anzahl an Testpersonen, die das Testsignal mit „befriedigend“ bewerten,

pm die Anzahl an Testpersonen, die das Testsignal mit „mäßig“ bewerten und

ps die Anzahl an Testpersonen, die das Testsignal mit „schlecht“ bewerten.

Die Testsignale können anhand ihres MOS aufgelistet werden, wobei das Testsignal mit dem größten Wert, das Signal und somit der Algorithmus ist, welcher vom gesamten Hörerfeld am meisten bevorzugt wurde.

³engl.: Mean Opinion Score

Nummer	Kategorie	Level an Sprachverzerrung und Rauschen
5	ausgezeichnet	Nicht wahrnehmbar.
4	gut	Gerade wahrnehmbar, aber nicht störend.
3	befriedigend	Wahrnehmbar und leicht störend.
2	mäßig	Störend, aber nicht unangenehm.
1	schlecht	Sehr störend und unangenehm.

Tabelle 5.1: Kategorien zur Bewertung der subjektiven Signalqualität.

5.3 Objektive Messung der Sprachqualität

5.3.1 Einleitung

Objektive Messmethoden werden eingesetzt, um Qualitätsvergleiche zwischen den Rauschunterdrückungsalgorithmen auf einfache Art und Weise zu ermöglichen. Viele Verfahren sind leicht zu implementieren und erlauben schnelle, quantitative Aussagen ohne großen Aufwand. Jedoch sind nicht alle Methoden mit den Ergebnissen einer subjektiven Teststudie vergleichbar bzw. können diese sogar ersetzen. Zur Bestimmung der Leistungsfähigkeit eines objektiven Messverfahrens wird die Korrelation mit der subjektiven Qualität berechnet. Dies wurde in einer umfangreichen Studie von Quackenbush *et.al.* für verschiedene Sprachcodierungsverfahren in [21] getan. Deller *et.al.* erweiterte diese Studie erstmals auf einkanale Rauschunterdrückungsverfahren [23]. Im Zuge dieser Arbeit wurde festgestellt, dass dies nicht ohne weiteres auf mehrkanaligen Rauschunterdrückungsalgorithmen zutrifft. Mitunter kann es sehr schwierig sein, aussagekräftige Messungen durchzuführen und vor allem korrekt zu interpretieren.

5.3.2 Messung des segmentiellen Signal-Rausch-Verhältnisses

Das Signal-Rausch-Verhältnis ist die am häufigsten angewandte Messmethode. Das klassische Signal-Rausch-Verhältnis in dB lässt sich gemäß [21] durch

$$SNR = 10 \log_{10} \frac{\sum_k s^2(k)}{\sum_k n^2(k)} \quad (5.2)$$

berechnen. In dieser Formel entspricht $s(k)$ dem reinen Sprachsignal und $n(k)$ dem reinen Rauschsignal.

Es hat sich jedoch herausgestellt, dass das klassische SNR für eine Vielzahl von Signalen ein schwacher Schätzer für die subjektive Sprachqualität ist. Als besseres Maß hat sich das segmentielle SNR (SSNR⁴) bewährt, welches gemäß [21] in dB durch

$$SSNR = \frac{10}{F} \sum_{m=0}^{F-1} \log_{10} \sum_{k=Lm}^{Lm+L-1} \left(\frac{s^2(k)}{n^2(k)} \right) \quad (5.3)$$

⁴engl.: Segmental Signal-to-Noise Ratio

formuliert werden kann, wobei F der Frameanzahl und L der Framelänge entspricht. Die Framedauer sollte zwischen 10–35 ms liegen. Für eine Abtastfrequenz von $f_s = 16$ kHz wurde in dieser Arbeit eine Framelänge von $L = 256$ Samples gewählt. Dies entspricht einer Framedauer von 16 ms.

Für diese Definition des SSNR treten Probleme auf, falls es Frames ohne Sprachaktivität (zum Beispiel in Sprachpausen) gibt. In Frames, in denen die Sprachleistung nahe Null ist, führt jeglicher Einfluss von Rauschen zu einem stark negativen SNR (in dB) für diesen Frame. Dies beeinflusst natürlich sehr stark das über alle Frames gemittelte SSNR. Es gibt zwei Möglichkeiten dieses Problem zu umgehen:

- Man arbeitet nur mit Signalen, die keine Sprachpausen aufweisen. Diese Methode wurde in dieser Arbeit angewandt und konnte näherungsweise durch flüssig gesprochene Sätze erfüllt werden.
- Man identifiziert Frames ohne Sprache und entfernt sie aus der Berechnung des SSNR. Dafür misst man die Energie jedes Frames des Sprachsignals und benutzt nur jene Frames zur Berechnung des SSNR, welche einen gewissen Energielevel überschreiten.

Hat man das SSNR am Eingang und am Ausgang des Rauschunterdrückungsalgorithmus berechnet, dann kann man daraus die SSNR Verbesserung (SSNRE⁵) in dB gemäß

$$SSNRE = SSNR_{\text{out}} - SSNR_{\text{in}} \quad (5.4)$$

bestimmen. Musical Noise, starker Nachhall oder ein Restrauschpegel können mit diesem Maß nicht quantifiziert bzw. festgestellt werden. Dies erfolgt am zuverlässigsten anhand von individuellen Hörtests.

5.3.3 Direkte Vergleichsmessungen

Viele objektive Messungen machen einen direkten Vergleich zwischen dem reinen Originalsignal (Referenzsignal) und dem verarbeiteten Ausgangssignal des untersuchten Systems. Dabei werden die Signale in Blöcke (Frames) mit typischen Blocklängen von 10 bis 35 ms zerteilt. Die meisten Messungen bestimmen spektrale Unterschiede zwischen den Frames der verglichenen Signale. Andere Messungen bestimmen parametrische Modelle für jeden Frame der Signale und vergleichen diese mit Hilfe einer Distanzmessung. Eine sehr wichtige Rolle spielt eine perfekte Synchronisierung der Signale. Signalverzögerungen, die aufgrund von Latenzzeiten des Systems auftreten können, müssen perfekt kompensiert werden, da ansonst die Messergebnisse keine korrekten Werte liefern und damit die Messungen unbrauchbar machen. Bei mehrkanaligen Algorithmen ist die Synchronisation ein noch größeres Problem. Die Eingangssignale der Mikrofone weisen aufgrund der Ausbreitung im Raum alle eine unterschiedliche Verzögerung zum Originalsignal auf. Hinzu kommt noch die entstehende Verzögerung durch das Rauschunterdrückungssystem selbst. Eine Möglichkeit zur Kompensation der Signalverzögerungen

⁵engl.: Segmental Signal-to-Noise Ratio Enhancement

erreicht man anhand der Berechnung der zeitlichen Kreuzkorrelation der beiden Signale. Das Maximum der Kreuzkorrelationsfunktion zeigt an, um wieviele Samples die beiden Signale gegeneinander verschoben werden müssen, damit eine entsprechende Synchronisation erzielt wird. Da in dieser Arbeit nur der Vergleich verschiedener Postfilter interessant war, wurde bei den direkten Frame-Vergleichsmessungen das reine Sprachsignal am Ausgang des Beamformers als Referenzsignal verwendet. Die untersuchten Postfilter nehmen eine reine Gewichtung der FFT-Bins vor und bringen keine weiteren Signalverzögerungen ein. Somit war eine perfekte Synchronisation der verglichenen Signale sichergestellt und trotzdem konnten sinnvolle Aussagen über die Leistungsfähigkeit der Postfilterfunktionen gemacht werden.

Es wurden zwei parametrische Distanzmessungen basierend auf einer Linear Prediction (LP) Analyse ausgewählt, um quantitative Aussagen über die Sprachqualität der behandelten Rauschunterdrückungsalgorithmen zu treffen [21, 23, 7].

Messung des Log-Area Ratio

Die Log-Area Ratio Messung basiert auf Unterschiede zwischen den PARTIAL CORRELATION (PARCOR) Koeffizienten des Referenzsignals und des verarbeiteten Ausgangssignals. Diese Koeffizienten können anhand einer LP Analyse ermittelt werden. Quackenbush *et.al.* zeigte in [21], dass die Messung des Log-Area Ratio die höchste Korrelation mit der subjektiven Qualität aufweist. Somit lässt sich mit diesem Maß allgemein die Sprachqualität der verschiedenen Algorithmen untereinander vergleichen. Die Berechnung erfolgt dabei in drei Schritten [23]:

1. Zuerst schätzt man die PARCOR Koeffizienten für jeden Frame eines Signals. Die Framelänge sollte kurz genug gewählt werden, dass die Annahme der Kurzzeitstationarität erfüllt ist, aber lang genug, dass die Varianz der geschätzten Werte möglichst gering ist. Für eine Abtastfrequenz von $f_s = 16$ kHz wurden mit einer Framelänge von $L = 256$ Samples gute Ergebnisse erzielt. Ein Algorithmus zur Bestimmung der PARCOR Koeffizienten $\kappa(p, m)$ ist die *Levinson-Durbin Rekursion*. Der Koeffizientenindex p liegt im Bereich $[1, P]$. Die Modellordnung P bestimmt die Anzahl an Koeffizienten (typische Ordnung $P = 14$). Die Berechnung ist im Anhang B.2 und in [21], sowie in [23], zu finden.

2. Danach werden die Area-Koeffizienten berechnet:

$$g(p, m) = \frac{1 + \kappa(p, m)}{1 - \kappa(p, m)}, \quad 1 \leq p \leq P. \quad (5.5)$$

3. Das LAR ist definiert als die euklidische Distanz zwischen den Area-Koeffizienten des Referenzsignals $g_{\text{ref}}(p, m)$ (üblicherweise das originale Sprachsignal) und dem verarbei-

teten Ausgangssignal $g_{y_{s+n}}(p, m)$ des Systems. Letztendlich berechnet sich das LAR für jeden Frame m in dB zu

$$LAR(m) = \sqrt{\frac{1}{P} \sum_{p=1}^P \left| 20 \log_{10} \left(\frac{g_{\text{ref}}(p, m)}{g_{y_{s+n}}(p, m)} \right) \right|^2}. \quad (5.6)$$

Durch Mittelung über alle Frames bekommt man ein allgemeines Qualitätsmaß. Hansen *et.al.* empfiehlt in [24] jene Frames, die ein besonders hohes Verzerrungslevel aufweisen, aus der Mittelung auszuschließen, um das Auftreten eines Biasfehler zu verhindern. Um diese Aufgabe zu erfüllen, werden 5% der Frames mit dem größten Verzerrungslevel entfernt und dann die Mittelung über die verbleibenden Frames durchgeführt. Die Messung des LAR beurteilt das Maß an Rauschen und der Sprachverzerrungen gemeinsam.

Messung der Sprachverzerrung

Als letzte objektive Größe wurde das Maß der reinen Sprachverzerrungen (SD^6) der getesteten Algorithmen berechnet. Das SD lässt sich wie das LAR als Verhältnis der Area-Koeffizienten definieren. In diesem Fall handelt es sich um einen Vergleich der Area-Koeffizienten des Referenzsignals $g_{\text{ref}}(p, m)$ und des reinen Sprachsignals am Ausgang des jeweiligen Postfilters $g_{y_s}(p, m)$. Das SD für jeden Frame m in dB ergibt sich aus

$$SD(m) = \sqrt{\frac{1}{P} \sum_{p=1}^P \left| 20 \log_{10} \left(\frac{g_{\text{ref}}(p, m)}{g_{y_s}(p, m)} \right) \right|^2}. \quad (5.7)$$

Wieder kann durch Mittelung über alle Frames ein globales Maß berechnet werden. Auch hier empfiehlt es sich jene 5% an Frames mit dem größten Verzerrungslevel von der Mittelung auszuschließen [24].

5.3.4 Master-Slave Simulationssystem zur Evaluierung objektiver Messgrößen

Die Verwendung des Master-Slave Simulationssystems zur Durchführung objektiver Qualitätsmessungen der einzelnen Algorithmen gewährleistet, dass die mehrkanaligen Sprach- und Rauschsignale getrennt voneinander verfügbar sind. Eine grafische Beschreibung des Simulationssystems zeigt Abbildung 5.6. Es besteht aus folgenden Blöcken:

- Der Block für die *SNR-Regelung* misst das aktuelle SNR beziehungsweise das segmentielle SNR der beiden Eingangsgrößen und stellt damit das gewünschte SNR bzw. SSNR für das verrauschte Eingangssignal ein, welches dann dem Master-Block zugeführt wird.

⁶engl.: Speech Degradation

- Der *Master-Block* enthält das „DUT“⁷, sozusagen den Algorithmus, der untersucht werden soll. Hier werden die Filterkoeffizienten des Beamformers und die adaptiven Filterkoeffizienten des Postfilters berechnet, die anschließend in die beiden Slave-Blöcke kopiert werden.
- Die beiden *Slave-Blöcke* verarbeiten nur das Sprachsignal beziehungsweise nur das Rauschsignal. Die Filterkoeffizienten werden dafür vom Master-Block berechnet und von den Slave-Blöcken übernommen.
- Die *Evaluierungseinheit* berechnet aus den Ausgangssignalen des Simulationssystems – dem verarbeiteten Sprachsignal $y_s(k)$, dem verarbeiteten Rauschsignal $y_n(k)$ und der verarbeitete Summe $y_{s+n}(k)$ – die Messgrößen zur objektiven Beurteilung der erzielten Sprachqualität.

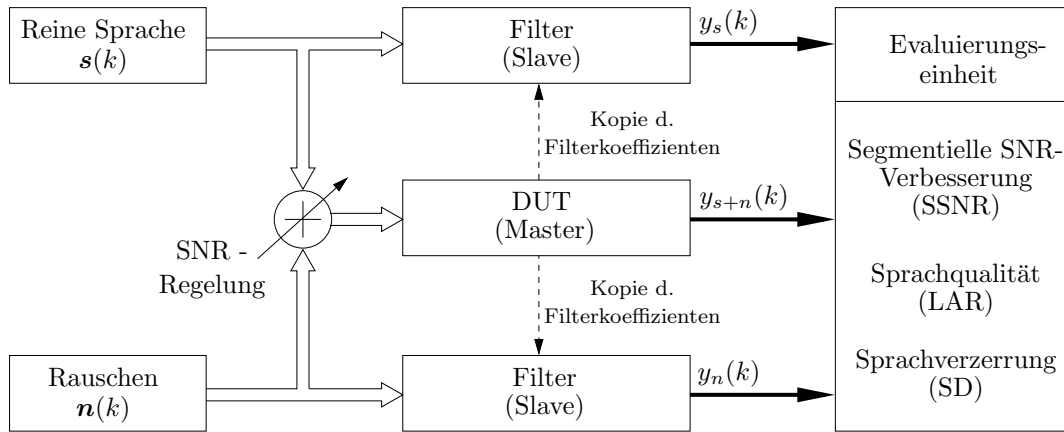


Abbildung 5.6: Objektive Messung der Algorithmenqualität mit Hilfe des Master-Slave Simulationssystems.

⁷engl.: Device Under Test

Kapitel 6

Experimente und Ergebnisse

In diesem Kapitel werden die Arraykonfigurationen und die Parametereinstellungen der Algorithmen erklärt, welche für die subjektive Analyse und die objektiven Messungen ausgewählt wurden. In Abschnitt 6.2 wird auf die subjektive Studie eingegangen, anhand derer Ergebnisse diejenigen Algorithmen evaluiert wurden, die für die Testpersonen am angenehmsten empfunden wurden. Danach wird ein Vergleich der Algorithmen bezüglich der durchgeführten objektiven Messungen präsentiert. Letztendlich werden die Postfilter mit den besten Rauschunterdrückungseigenschaften ermittelt.

Folgende Kurzbezeichnungen wurden in den weiteren Abschnitten verwendet:

BS	Broadside-Konfiguration
EF	Endfire-Konfiguration
CN	Comfort Noise: Es wurde die Mindestfilterfunktion gemäß Abschnitt 4.6.2 angewandt und damit ein gewisser Restrauschpegel hinzugefügt.
AA	Adaptives Alpha: Es wurde zur Berechnung der Leistungsdichtespektren (nach Welch) ein adaptiver Glättungsfaktor gemäß Abschnitt 4.6.2 verwendet.

6.1 Parametereinstellungen zur Durchführung der Analysen

Die subjektiven und die objektiven Analysen wurden für die Umgebung *Büroraum* und *Auto* durchgeführt. Aufgrund der Vielzahl an Konfigurationsoptionen für das Mikrofonarray und den zahlreichen Algorithmenparametern mussten für die Untersuchungen repräsentative Einstellungen getroffen werden. Außerdem ist es wichtig, dass für einen Leistungsvergleich der Algorithmen immer dieselben Einstellungen getroffen werden. Falls nicht anders angegeben, wurden für die objektive und subjektive Analyse der Algorithmen jene Konfigurationen verwendet, welche in den nachfolgenden Abschnitten beschrieben werden.

6.1.1 Auswahl des Eingangssignals

Es wurden die Aufnahmen mit dem 8-kanaligen, harmonischen Array aus Abbildung 5.3 auf Seite 53 ausgewählt, um die Postfiltersysteme zu testen. Der Vorteil liegt darin, dass durch die Verwendung der Mikrofonkanäle 3, 4, 5, 6 des 8-kanaligen Arrays automatisch auch die Aufnahmen für ein 4-kanaliges, äquidistantes Mikrofonarray mit einem Mikrofonabstand von $l = 2.5 \text{ cm}$ vorliegen. Mit einer einzigen Aufnahmekonfiguration konnten somit wahlweise zwei verschiedene Typen von Mikrofonarrays getestet werden. Die Abstände zwischen den Mikrofonen beider Arrays sind für eine Abtastfrequenz von $f_s = 16 \text{ kHz}$ optimal. Bei dieser Abtastfrequenz tritt räumliches Aliasing frühestens ab einer Frequenz von $f = 6800 \text{ Hz}$ auf. Äußerst sinnvoll ist deshalb der Einsatz eines Tiefpass mit Grenzfrequenz $f_{\text{high}} = 6800 \text{ Hz}$, womit der Effekt des räumlichen Aliasing vermieden werden konnte. Zusätzlich empfiehlt es sich einen Hochpass mit Grenzfrequenz $f_{\text{low}} = 200 \text{ Hz}$ zu verwenden. In automotiver Umgebung konnten dadurch die Rumpelgeräusche der Fahrbahn herausgefiltert werden. Für die Bürosituation ergaben sich keine Nachteile durch die Verwendung des Tiefpassfilters. Beide Filter können zu einem Bandpass vereint werden und sollten direkt nach der Berechnung der FFT implementiert werden.

Sowohl bei den Tests für den Büroraum, als auch bei jenen für das Auto, wurden die Aufnahmen eines männlichen Sprechers verwendet. Alle Analysen für den Büroraum wurden mit Aufnahmen in Broadside- und in Endfire-Konfiguration als repräsentative Einfallswinkel durchgeführt, wobei sich der Sprecher in einer Entfernung von 70 cm zum Mikrofonarray befand. Typisch für das Rauschen in einem Büroraum wurde ein diffuses Rauschfeld verwendet, welches mit Hilfe der Konfiguration gemäß Abschnitt 5.1.1 generiert wurde. Die Sprach- und die Rauschsignale wurden derart kombiniert, dass sich die Eingangs-SSNR-Werte $SSNR_{\text{in}} = -10, -5, 0, 5, 10, 15$ und 20 dB ergaben. Für die automotive Umgebung wurden die Aufnahmen gemäß den Einstellungen in Abschnitt 5.1.2 eingesetzt. Berechnet man das Eingangs-SSNR bezüglich der unterschiedlichen Fahrgeräuscharten, dann ergibt sich für die Situation „Fahrgeräusche“ $SSNR_{\text{in}} = -5.76 \text{ dB}$, die Situation „Offenes Schiebedach“ $SSNR_{\text{in}} = -6.47 \text{ dB}$ und für „Offenes Fenster“ $SSNR_{\text{in}} = -7.84 \text{ dB}$. Die ausgewählten Aufnahmen zum Testen der Algorithmen sind in Tabelle 6.1 nochmals zusammengefasst.

Eigenschaften	Büro	Auto
Arraytyp	4-kanalig, äquidistant und 8-kanalig, harmonisch	
Abtastfrequenz [kHz]	16	
Sprecher	männlich	
Entfernung [cm]	70	58
Einfallswinkel θ_0	0° (EF), 90° (BS)	126°
Rauschen	diffus	Fahrgeräusche (3 Arten)
$SSNR_{\text{in}}$ [dB]	$-10, -5, 0, 5, 10, 15, 20$	$-5.76, -6.47, -7.84$

Tabelle 6.1: Wahl der Originalaufnahmen zur Analyse der Algorithmen.

6.1.2 Wahl des Glättungsfaktors α

Der Glättungsfaktor α wird in allen Algorithmen zur Schätzung der Leistungsdichtespektren verwendet. Die Eigenschaften des Glättungsfaktors wurde bereits ausführlich in Abschnitt 2.5 beschrieben. Der genaue Wert von α wurde durch individuelle Hörtests und anhand von objektiven Messungen bestimmt. Für die objektiven Messungen wurde die SSNR Verbesserung und das Maß an Sprachverzerrungen in Abhängigkeit vom Glättungsfaktor α berechnet. Die Ergebnisse der Messungen sind in Abbildung 6.1 für die in Kapitel 4 Algorithmen dargestellt. Um das Auftreten störender Zeitartefakte zu vermeiden, sollte α nicht kleiner als 0.6 gewählt werden. Ein Glättungsfaktor nahe 1 bewirkt wiederum einen deutlich hörbaren Nachhaleffekt. Vergleicht man die SSNR Verbesserung der unterschiedlichen Algorithmen über α , dann sieht man das bei einigen Algorithmen das höchste SSNRE in der Nähe von $\alpha = 0.8$ liegt. Die übrigen Algorithmen zeigen ein abnehmendes SSNRE bei steigendem α , wobei die Abnahme für $\alpha < 0.8$ sehr gering ist. Betrachtet man zusätzlich den Einfluss der Sprachverzerrungen über α , dann erkennt man eine abnehmende Tendenz der SD-Werte mit zunehmenden α für alle Algorithmen. Aufgrund dieser Erkenntnisse und durch informelle Hörtests wurde daher für alle analysierten Algorithmen, als Kompromiss zwischen SSNR Verbesserung und Sprachverzerrungen, ein Glättungsfaktor von $\alpha = 0.8$ gewählt.

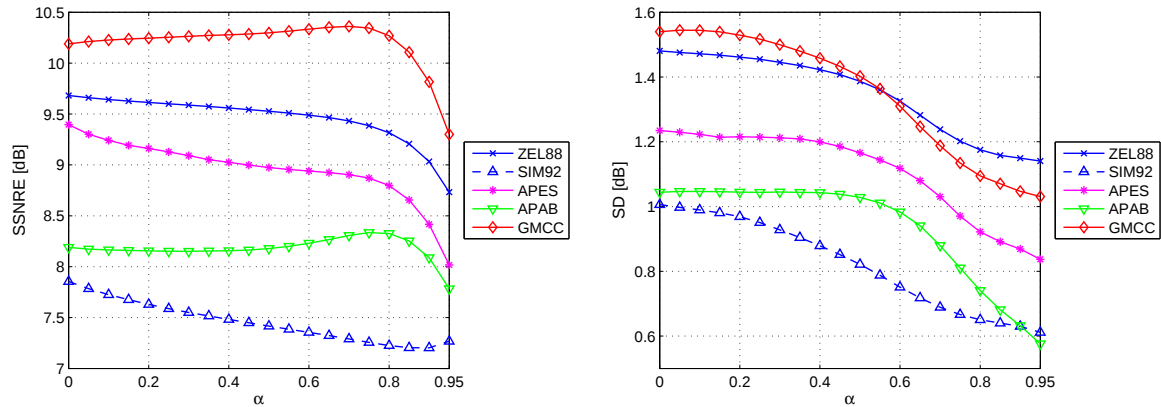


Abbildung 6.1: Links: SSNR Verbesserung (SSNRE) über den Glättungsfaktor α für verschiedene Postfilteralgorithmen. Rechts: Sprachverzerrungen (SD) über den Glättungsfaktor α für verschiedene Postfilteralgorithmen. (Einstellungen: $f_s = 16$ kHz; 8-kanaliges, harmonisches Array; Endfire; SDB; $\mu = -20$ dB)

6.1.3 Wahl des Beamformers

Da für die untersuchten Rauschumgebungen im Büro und im Auto näherungsweise diffuse Rauschfelder zutreffend sind, werden für die objektiven und subjektiven Studien die Postfilterfunktionen ausschließlich in Kombination mit einem SDB eingesetzt. Wie bereits in Abschnitt 3.4 beschrieben, ist für den Einsatz eines SDB unbedingt ein beschränkter Entwurf notwendig. In Folge dessen, ist eine geeignete Wahl des Regularisierungsparameters μ erforder-

derlich. Für Werte von μ über -10 dB wird keine ausreichende Direktivität mehr erzielt. Bei Werten von μ unter -40 dB wird das verstärkte Eigenrauschen der Mikrofone hörbar. Die besten Ergebnisse wurden anhand von individuellen Hörtests mit einem μ von -20 dB erzielt. Daher wurde dieser Wert sowohl für die objektiven als auch für die subjektiven Analysen übernommen.

6.1.4 Einsatz von Maßnahmen zur Reduktion von Sprachverzerrungen und Musical Noise

Die betrachteten Algorithmen wurden zusätzlich in Kombination mit einem adaptiven Glättungsfaktor und/oder mit einer Mindestfilterfunktion getestet. Diese wurden exakt mit den Werten aus Abschnitt 4.6.2 implementiert. Der adaptive Glättungsfaktor zur Berechnung der Leistungsdichtespektren nach Welch erreicht eine Verringerung von Musical Noise bei gleichzeitiger Verstärkung der Sprachverzerrungen. Die Mindestfilterfunktion liefert zwar geringere Sprachverzerrungen, ruft aber einen gewissen Restrauschpegel (Comfort Noise) hervor.

6.2 Subjektive Studie zum Vergleich der Algorithmenqualität

6.2.1 Algorithmenwahl und Vorselektion

Um die Konzentration der Testpersonen, die an dieser Studie teilnahmen, zu wahren, musste eine annehmbare Testdauer (nicht länger als 15 Minuten) eingehalten werden. Daher konnte nur eine bestimmte Auswahl an Testsignalen und damit an Algorithmen zur Studie zugelassen werden.

Als Eingangssignal (Originalsignal) für die Algorithmen der Bürraumsituation wurde ein Signal mit den Eigenschaften aus Tabelle 6.1 gewählt. Allerdings kam nur das 8-kanaligen, harmonische Mikrofonarray zum Einsatz und als Eingangs-SSNR wurde ein Wert von $SSNR_{\text{in}} = 10$ dB genommen. Die Algorithmenqualität wurde für Broadside- und Endfire-Konfiguration getestet.

Für die automotiv Umgebung wurde ebenfalls ein Signal mit den Eigenschaften aus Tabelle 6.1 als Eingangssignal verwendet, wobei ausschließlich das 4-kanalige, äquidistante Mikrofonarray eingesetzt wurde. Für die Störgeräusche im Auto wurde die Konfiguration „Fahrgeräusche“ gewählt. Dadurch ergab sich ein Eingangs-SSNR von ca. $SSNR_{\text{in}} = -5.76$ dB.

Alle in der Arbeit beschriebenen Postfilteralgorithmen standen zur Verfügung, wobei aufgrund des näherungsweise diffusen Rauschfeldes als Beamformer ein SDB verwendet wurde. Als Algorithmenparameter wurden $\alpha = 0.8$ und $\mu = -20$ dB eingestellt. Ein herkömmlicher SDB wurde, sozusagen als Klassiker, ebenfalls zur Studie hinzugezogen. Alle Algorithmen wurden zusätzlich in Kombination mit der Mindestfilterfunktion und/oder in Kombination mit dem adaptiven Glättungsfaktor angewendet. Diese Vielzahl an Algorithmen wurde durch individuelle Hörtests von Dr. Dobliger auf eine für die Studie annehmbare Zahl reduziert. Letztendlich wurden 25 verschiedene Algorithmen für die Bürraumsituation zur Studie zu-

gelassen, die in den Rohdatentabellen C.1, C.2 und C.3 im Anhang ersichtlich sind. Für die automotive Umgebung ergaben sich 11 Algorithmen, die in den Rohdatentabellen C.4, C.5 und C.6 im Anhang dargestellt sind.

6.2.2 Vorgehensweise und Rohdaten

Es wurde eine subjektive Studie gemäß Abschnitt 5.2 mit 20 Testpersonen im Alter von 22 bis 37 Jahren durchgeführt. Um den Testpersonen allerdings mehr Spielraum für die Bewertung zu ermöglichen, durften sie zusätzlich zu den Bewertungszahlen 1 bis 5 auch Halbnoten zur Beurteilung vergeben. Dadurch ergaben sich 9 verschiedene Bewertungskategorien. Diese Anzahl an Kategorien ist gemäß den Studien von Miller [25] als oberstes Limit vertretbar. Ein trainiertes Hörerfeld wurde näherungsweise erzielt, indem jede Testperson drei Testdurchgänge absolvierte. Die Durchgänge waren dabei zeitlich folgendermaßen verteilt: Der erste Durchgang fand am ersten Testtag statt, der zweite am zweiten Tag und der dritte am fünften Tag. Der Abstand von drei Tagen zwischen dem zweiten und dem dritten Durchgang wurde gewählt, um den Testpersonen genügend Zeit zu geben, die Höreindrücke zu verarbeiten und im Gedächtnis zu verankern. Bei insgesamt 20 Personen ergab dies eine Anzahl von 60 Durchgängen. Der erste Durchgang wurde nicht zur Auswertung herangezogen, sondern diente ausschließlich als Trainingsdurchgang. Dies wurde den Testpersonen allerdings nicht mitgeteilt.

In der Trainingsphase wurde den Testpersonen ein Trainingssignal der Kategorie 5 („schlecht“) und ein Trainingssignal der Kategorie 2 („gut“) vorgespielt. Das Signal der Kategorie 2 wurde gewählt, um die Testpersonen nicht allzu sehr in ihrer Entscheidung für das „beste“ Signal zu beeinflussen. Um eine längere Trainingsphase zu erzielen, wurden die ersten zwei Testsignale der Evaluierungsphase nochmals in den Testumfang eingestreut. Dieser Umstand war den Testpersonen jedoch nicht bekannt. Die ersten zwei Testsignale wurden später auch nicht zur Auswertung herangezogen. Weiters wurde die Originalaufnahme von den Testpersonen blind bewertet. Für die Algorithmen der Evaluierungsphase wurde eine zufällige Reihenfolge gewählt. Außerdem wurden die Lautstärken der Testsignale zuvor abgeglichen.

Der Test wurde in einem ruhigen Raum ohne Störgeräusche von einem Testleiter durchgeführt. Die Testsignale wurden auf einem *iPod* von *Apple* unkomprimiert gespeichert und den Testpersonen über Kopfhörer von *AKG Acoustics* mit der Bezeichnung *K500* vorgespielt. Vor der eigentlichen Testdurchführung wurde den Probanden der Zweck und das Beurteilungsverfahren erklärt. Jedes Testsignal dauerte ca. 10 Sekunden. Nach jedem Testsignal hatte jede Testperson ca. 5–10 Sekunden zur Verfügung, um eine Beurteilung abzugeben. Den Testpersonen wurden die Testsignale für die Büroraumumgebung und die Testsignale für die automotive Umgebung in einem Durchgang präsentiert. Inklusive den Trainingssequenzen und den eingestreuten Originalsequenzen ergaben sich insgesamt 45 Testsignale. Gemeinsam mit der einführenden Erklärung und den Beurteilungszeiten erreichte ein gesamter Testdurchgang eine Dauer von ca. 13–15 Minuten. Die Beurteilungsergebnisse der Testdurchgänge (von Bü-

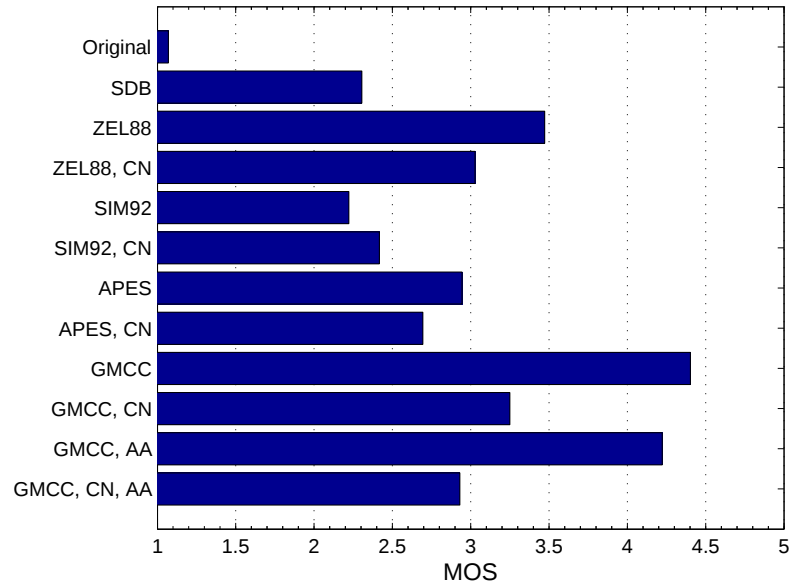
rum und Auto) sind in den Rohdatentabellen C.1, C.2, C.3, C.4, C.5 und C.6 im Anhang ersichtlich.

6.2.3 Auswertung

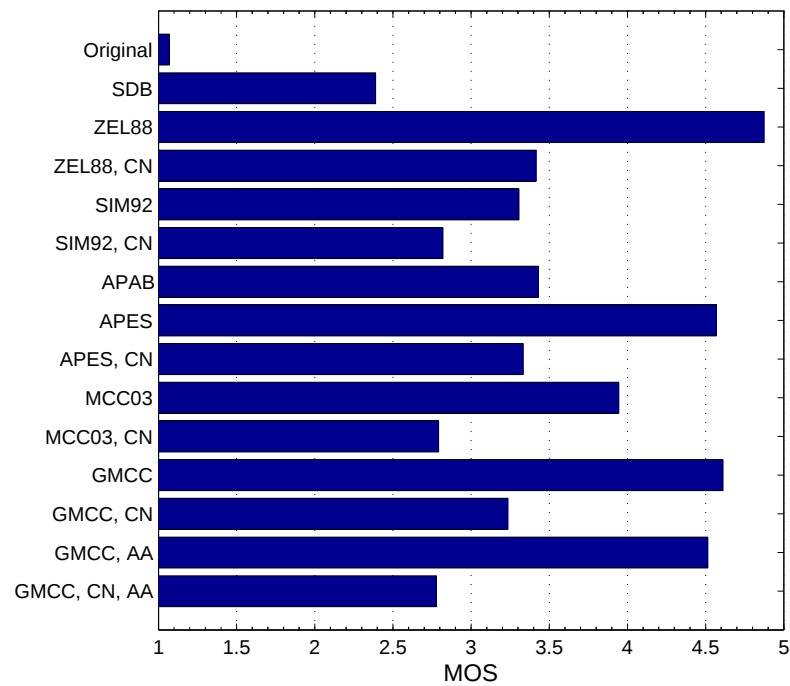
Zur Auswertung wurden nur der zweite und der dritte Testdurchgang einer jeden Testperson verwendet, da der erste Durchgang als Trainingsdurchgang galt. Dies ergab insgesamt 40 Durchgänge, die für die Auswertung relevant waren. Um grobe Abweichungen der einzelnen Bewertungen eines Testsignals zu minimieren, wurden von den 40 Durchgängen weitere 10% ausgeschieden. Dafür wurde zuerst der Mittelwert $\hat{\mu}_T = \frac{1}{40} \sum_{i=1}^{40} B_T(i)$ für jede Testsequenz berechnet, wobei $B_T(i)$ die Bewertungszahl der Testperson i für das Testsignal T ist. Danach konnte die quadratische Abweichung $\hat{\sigma}_T(i) = (B_T(i) - \hat{\mu}_T)^2$ errechnet werden. Die vier Durchgänge jeder Testsequenz mit der größten quadratischen Abweichung wurden eliminiert, wonach insgesamt 36 Durchgänge zur Berechnung des MOS zur Verfügung standen. Der MOS wurde gemäß Gleichung 5.1 aus Abschnitt 5.2, jedoch unter Einbezug der Halbnoten, ermittelt.

Abbildung 6.2(a) zeigt die Auflistung der Algorithmen bezüglich ihres MOS für Aufnahmen in der Broadside-Konfiguration. Die Testsignale, die vom *GMCC*-Algorithmus und der Kombination des *GMCC*-Algorithmus mit dem adaptiven Glättungsfaktor α verarbeitet wurden, wurden von den Testpersonen am angenehmsten empfunden und heben sich am deutlichsten von den anderen Algorithmen ab. Es ist zu beachten, dass gemäß Abschnitt 4.3 das *GMCC*-Postfilter und das *MCC03*-Postfilter für die Broadside-Konfiguration identisch sind. Alle anderen Algorithmen fallen in ein breites Mittelfeld. Für Aufnahmen in der Endfire-Konfiguration (siehe Abbildung 6.2(b)) zeigt der *ZEL88*-Algorithmus den höchsten MOS. Von dem breiten Mittelfeld heben sich außerdem noch der *GMCC*-Algorithmus, der *GMCC*-Algorithmus in Kombination mit dem adaptiven Glättungsfaktor und der *APES*-Algorithmus mit einem guten Ergebnis ab. Der herkömmliche SDB hat bezogen auf die anderen Algorithmen für beide Konfigurationen (Broadside und Endfire) den geringsten mittleren Meinungswert. Dies bestätigt die Tatsache, dass eine zusätzliche Verbesserung der Sprachqualität durch Einsatz eines adaptiven Postfilters erzielt werden kann. Vergleicht man die Algorithmen beider Konfigurationen, dann werden die Algorithmen der Endfire-Konfiguration von den Testprobanden am angenehmsten empfunden. Tabelle 6.2 zeigt eine Auflistung der Algorithmen, die entsprechend absteigendem MOS sortiert wurden. Da der Beamformer für die Endfire-Konfiguration eine höhere Direktivität aufweist und dadurch eine bessere Rauschunterdrückung hat, kann auch das nachfolgende Postfilter eine höhere Leistungsfähigkeit erzielen. Dies erklärt die Überlegenheit der Algorithmen in Endfire-Konfiguration gegenüber jenen in Broadside-Konfiguration. Der Einbau eines Arrays in Endfire-Position in ein Kraftfahrzeug ist allerdings sehr schwierig zu realisieren. Der insgesamt am angenehmsten empfundene Algorithmus der Bürosituation ist der *ZEL88*-Algorithmus. Das blind bewertete Original-

signal weist erwartungsgemäß den geringsten mittleren Meinungswert auf.



(a) Broadside



(b) Endfire

Abbildung 6.2: Algorithmenvergleich anhand des MOS für den Büroraum. Oben: Broadside-Konfiguration. Unten: Endfire-Konfiguration.

Algorithmus		MOS
ZEL88	EF	4.875
GMCC	EF	4.611
APES	EF	4.569
GMCC, AA	EF	4.514
GMCC	BS	4.403
GMCC, AA	BS	4.222
MCC03	EF	3.944
ZEL88	BS	3.472
APAB	EF	3.431
ZEL88, CN	EF	3.417
APES, CN	EF	3.333
SIM92	EF	3.306
GMCC, CN	BS	3.250
GMCC, CN	EF	3.236
ZEL88, CN	BS	3.028
APES	BS	2.944
GMCC, CN, AA	BS	2.931
SIM92, CN	EF	2.819
MCC03, CN	EF	2.792
GMCC, CN, AA	EF	2.778
APES, CN	BS	2.694
SIM92, CN	BS	2.417
SDB	EF	2.389
SDB	BS	2.306
SIM92	BS	2.222
Original	–	1.069

Tabelle 6.2: Vergleich aller getesteten Algorithmen der Büroräumumgebung anhand des mittleren Meinungswertes (MOS), absteigend sortiert.

Die Algorithmen, die in automotiver Umgebung getestet wurden, sind in Abbildung 6.3 zu sehen. Generell erzielten die Algorithmen von McCowan die besten Ergebnisse. Vor allem das *MCC03*-Postfilter in Kombination mit dem adaptiven Glättungsfaktor ist für die meisten Testpersonen am angenehmsten. Dieser Algorithmus unterdrückt am stärksten instationäres Rauschen, wie es in automotiver Umgebung vorkommt. Zusätzlich bewirkt der adaptive Glättungsfaktor eine Verringerung des störenden Musical Noise. Trotz auftretender Sprachverzerrungen konnte er die Testpersonen überzeugen. Die Algorithmen, die mit der Mindestfilterfunktion (CN) kombiniert wurden, wurden trotz der geringeren Sprachverzerrungen, jedoch aufgrund des zusätzlichen Comfort Noise, nicht so gut bewertet. Das Originalsignal wurde wiederum am schlechtesten beurteilt. Alle Algorithmen der automotiven Umgebung sind zusätzlich in Tabelle 6.3, sortiert nach absteigendem MOS, aufgelistet.

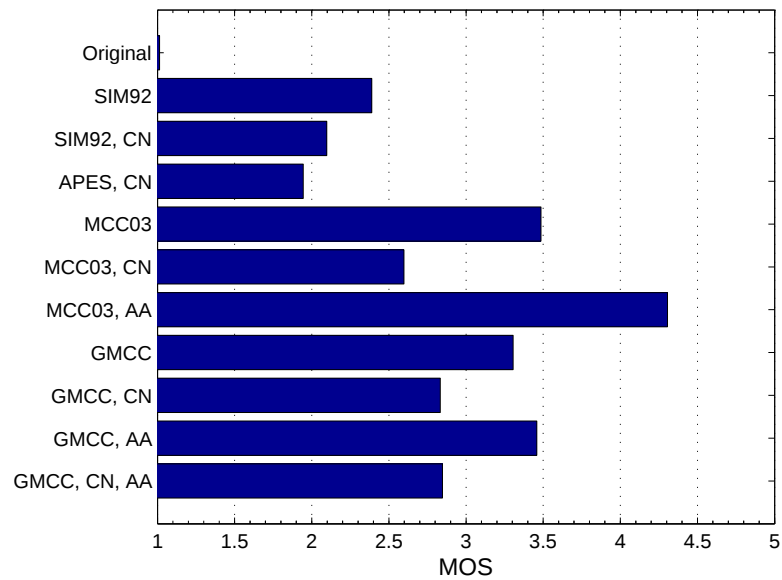


Abbildung 6.3: Graphische Darstellung des Algorithmenvergleichs für die automotiv Umgebung anhand des MOS.

Algorithmus	MOS
MCC03, AA	4.306
MCC03	3.486
GMCC, AA	3.458
GMCC	3.306
GMCC, CN, AA	2.847
GMCC, CN	2.833
MCC03, CN	2.597
SIM92	2.389
SIM92, CN	2.097
APES, CN	1.944
Original	1.014

Tabelle 6.3: Vergleich aller getesteten Algorithmen der automotiv Umgebung anhand des mittleren Meinungswertes (MOS), absteigend sortiert.

6.3 Objektive Messungen zum Vergleich der Algorithmenqualität

6.3.1 Messeinstellungen und Algorithmenwahl

Gemäß Abschnitt 5.3 wurden drei objektive Messgrößen bestimmt: die SSNR Verbesserung (SSNRE), das Log-Area Ratio (LAR) und die Sprachverzerrungen (SD). Die Framelänge wurde für die Abtastfrequenz von $f_s = 16$ kHz mit $L = 256$ Samples gewählt. Dies entspricht einer Zeitdauer von 16 ms. Als Eingangssignale zum Testen der Algorithmen wurden Aufnahmen mit den Eigenschaften aus Tabelle 6.1 eingesetzt. Es wurde darauf geachtet, dass die Sätze flüssig gesprochen wurden, damit korrekte Berechnungen des SSNRE durchgeführt werden konnten. Als Referenzsignal zur Bestimmung des LAR und SD wurde das reine Sprachsignal am Ausgang des Beamformers verwendet. Auf diesem Weg konnten Synchronisationsprobleme vermieden werden. Des Weiteren kam eine Modellordnung von $P = 14$ zum Einsatz.

Jene Algorithmen, die für die Bürraumsituation zum Einsatz kamen, sind in Tabelle 6.4 auf Seite 78 ersichtlich. Es ist zu beachten, dass in der Broadside-Konfiguration die Algorithmen *GMCC* und *MCC03* identisch sind. Alle Algorithmen, die in der automotiven Umgebung getestet wurden, sind in Tabelle 6.5 auf Seite 80 aufgelistet. Bei sehr niedrigem EingangssNR oder bei stark instationärem Rauschen produzierten die Algorithmen von McCowan (*GMCC* und *MCC03*) teilweise starkes Musical Noise. Daher wurden sie zusätzlich ebenso in Kombination mit dem adaptiven Glättungsfaktor (AA) analysiert. Alle Algorithmen wurden überdies mit der Mindestfilterfunktion (CN) getestet. Zum Vergleich wurde für beide Rauschsituationen ein herkömmlicher DSB und ein SDB getestet.

Die Ergebnisse aller objektiven Messungen sind in Tabelle C.7, C.8 und C.9 im Anhang dargestellt.

6.3.2 Probleme bei den Messungen

Alle Messungen des LAR führten zu keinen zufriedenstellenden und aussagekräftigen Ergebnissen. Sie spiegelten nicht, wie angenommen, die subjektive Studie wieder. Zahlreiche Tests und Experimente zeigten, dass die verwendeten Rauschsignale nicht im gewünschten Maß in die Messung eingingen, sondern ausschließlich das Maß der Sprachverzerrungen die Messung beeinflusste. Deshalb eignete sich diese Messung nicht, um die allgemeine Sprachqualität, welche durch die Algorithmen erreicht wird, zu beurteilen. Es wird vermutet, dass die Ursache bei den aufgenommenen Rauschsignalen liegt. Das Leistungsdichtespektrum der Rauschsignale hat ab einer Frequenz von ca. 1000 Hz einen sehr starken exponentiellen Abfall (ungefähr gemäß f^{-3}) zu verzeichnen. Die meiste Energie des Rauschsignals befindet sich also in einem Frequenzband von ca. 100–1300 Hz. Rauschen in diesem „kritischen“ Frequenzband wird als sehr störend empfunden, jedoch wird dies nicht in der LAR-Messung sichtbar. Tests mit breitbandigem, weißem, gaußischem Rauschen brachten allerdings die gewünschten, aussagekräftigen Ergebnisse. Auch in der Literatur wird zumeist weißes, gaußisches Rauschen

zum Testen von Rauschunterdrückungsalgorithmen und zur Evaluierung der Sprachqualität verwendet. Da weißes, gaußsches Rauschen in der Praxis nicht von Maschinen oder Automotoren erzeugt wird und demnach keine realen Rauschsignale simuliert, wurde es nicht in dieser Arbeit verwendet. Aufgrund dieser Erkenntnisse wird vermutet, dass eine sinnvolle LAR-Messung nur mit breitbandigem Rauschen durchführbar ist. Diese Vermutung konnte in dieser Diplomarbeit nicht weiter überprüft werden, wodurch auf die Messung des Log-Area Ratio verzichtet werden musste.

Im Zuge der Analyse der LAR-Messung wurde ebenso festgestellt, dass die LAR-Messung kaum durch Musical Noise beeinflusst wird und die LAR-Werte quantitativ nicht verändert. Musical Noise wird allerdings von vielen Rauschunterdrückungsalgorithmen hervorgerufen. Aus diesem Grund wäre eine objektive Messmethode, die den Einfluss von Musical Noise quantifiziert, wünschenswert. Evaluierungsmethoden zur Ermittlung der objektiven Sprachqualität von mehrkanaligen Rauschunterdrückungsalgorithmen sind ein Thema, welches auf jeden Fall noch intensiver Forschung und genaueren Analysen bedarf.

6.3.3 Auswertung

Büroraum

Die Abbildung 6.4 zeigt die Messungen des SSNRE und Abbildung 6.5 das Maß an Sprachverzerrung der Grundalgorithmen für die Broadside- und Endfire-Konfiguration des 8-kanaligen, harmonischen Arrays. In der Endfire-Konfiguration erreichen die Algorithmen höhere SSNRE-Werte als in der Broadside-Konfiguration. Dies ist auf die höhere Direktivität des Beamformers in dieser Position zurückzuführen, wonach die nachfolgenden Postfilterfunktionen zusätzlich eine bessere Leistungsfähigkeit und damit eine stärkere Verbesserung des SSNR erzielen können. Generell ist zu sagen, dass bei allen Algorithmen die Sprachverzerrungen mit fallendem Eingangs-SSNR zunehmen. Mit Ausnahme des DSB und des SDB, ist diese Eigenschaft auf die starke Rauschunterdrückung zurückzuführen, welche teilweise auch das gewünschte Sprachsignal unterdrückt und verzerrt. Aus den individuellen Hörtests geht hervor, dass bei hohem Eingangs-SSNR oft störendes Musical Noise auftritt.

In der Broadside-Konfiguration hat der *GMCC/MCC03*-Algorithmus gefolgt vom *ZEL88*-Algorithmus die höchsten SSNRE-Werte. Betrachtet man die Ergebnisse der gemessenen Sprachverzerrung dieser Algorithmen (siehe Abbildung 6.5, links), dann erkennt man annehmbare Resultate im Bereich hoher Eingangs-SSNR-Werte. Mäßige Resultate zeigen der *APES*- und der *APAB*-Algorithmus in der Broadside-Konfiguration. Sie weisen ähnliche Eigenschaften auf. Für niedrige SSNR-Werte ergibt sich für die beiden Postfilter nur bedingt ein höheres SSNRE als beim herkömmlichen SDB. Dafür produzieren sie weniger Sprachverzerrungen als der *GMCC/MCC03*- und der *ZEL88*-Algorithmus. Für hohe Werte des Eingangs-SSNR sinkt das SSNRE unter jenes des SDB, obwohl es zu keiner Abnahme der Sprachverzerrungen kommt. Auch der *SIM92*-Algorithmus liefert nur geringfügig bessere SSNRE-Werte als der herkömmliche SDB. Die Sprache ist allerdings stärker verzerrt. Eine dementsprechend

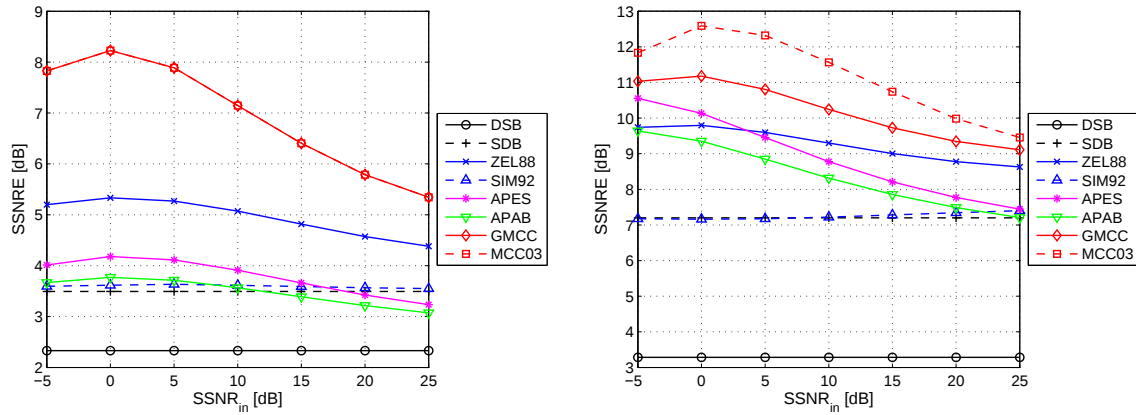


Abbildung 6.4: Vergleich der Algorithmen anhand der Messung der SSNR Verbesserung (SSNRE) in Abhängigkeit vom Eingangs-SSNR für ein harmonisches 8-Kanal-Array. Links: Broadside-Konfiguration. Rechts: Endfire-Konfiguration.

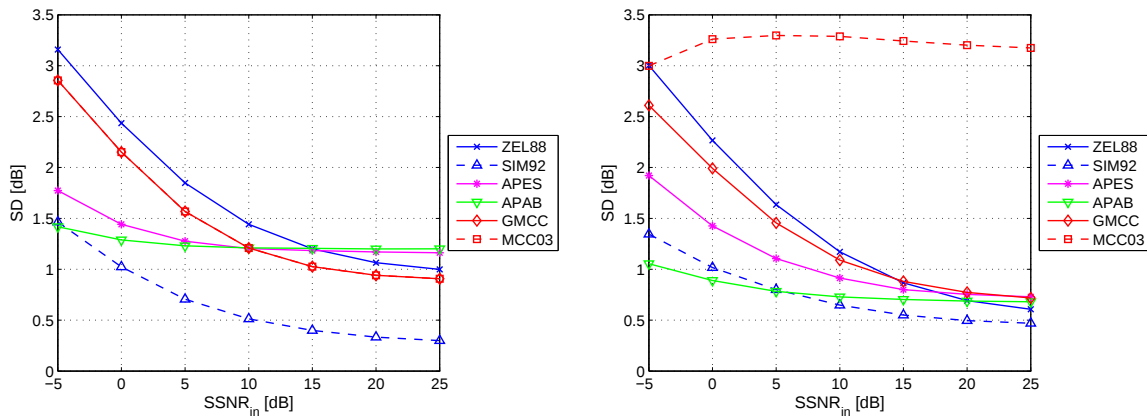


Abbildung 6.5: Vergleich der Algorithmen anhand der Messung der Sprachverzerrung (SD) in Abhängigkeit vom Eingangs-SSNR für ein harmonisches 8-Kanal-Array. Links: Broadside-Konfiguration. Rechts: Endfire-Konfiguration.

schlechte Bewertung erhielt dieses Postfilter in der subjektiven Studie (siehe Tabelle 6.2). Die dürtigen Ergebnisse sind auf das suboptimale Verhalten des *SIM92*-Postfilters in diffusen Rauschfeldern zurückzuführen.

In der Endfire-Konfiguration hat der *MCC03*-Algorithmus die höchsten SSNRE-Werte (siehe Abb. 6.4, rechts), aber es treten extrem starke Signalverzerrungen auf (siehe Abbildung 6.5, rechts). Diese Eigenschaften wurden durch individuelle Hörtests nachgewiesen und bestätigen die theoretischen Erkenntnisse aus Abschnitt 4.3. Sowohl in Bezug auf die SSNRE Verbesserung als auch in Bezug auf die Messung der Sprachverzerrung weisen der *GMCC*-, der *ZEL88*- und der *APES*-Algorithmus im Bereich hoher Eingangs-SSNR-Werte ($SSNR_{in} > 10$ dB) die besten Ergebnisse auf. Das *APAB*-Postfilter hat in der Endfire-Konfiguration ähnliche Eigenschaften wie das *APES*-Postfilter. Generell zeigt es niedrigere SSNRE-Werte, aber dafür auch eine geringere Sprachverzerrung als *APES*. Der *SIM92*-Algorithmus und der herkömmliche

SDB zeigen identische SSNRE-Verhältnisse. Hörtests bestätigten, dass das Rauschen durch das *SIM92*-Postfilter vermindert, aber zudem Musical Noise stärker hörbar wird.

Für einen besseren Vergleich wurden die SSNRE- und SD-Werte für Eingangs-SSNR-Werte von 10 dB, 15 dB und 20 dB in der Tabelle 6.4 zusammengefasst. Die Tabelle stellt ebenso die Resultate des 4-kanaligen, äquidistanten Mikrofonarrays dar. Generell werden mit einer größeren Anzahl an Mikrofonen höhere SSNRE-Werte erzielt. Die Ergebnisse aller untersuchten Algorithmen und deren Kombinationen mit der Mindestfilterfunktion und/oder dem adaptiven Glättungsfaktor für sämtliche Eingangs-SSNR-Werte sind in den Tabellen C.7 und C.8 im Anhang zu finden. Wie erwartet, lässt sich für jene Postfilter, die mit der Mindestfilterfunktion kombiniert werden, erkennen, dass die Sprachverzerrungen abnehmen. Hörtests bestätigen den Verbleib eines gewissen Restrauschpegels. Für jene Postfilter in Kombination mit einem adaptiven Glättungsfaktor nehmen die Sprachverzerrungen zu, jedoch erhöht sich das Maß der SSNR Verbesserung. Zudem wird auftretendes Musical Noise hörbar vermindert.

Auto

In Abbildung 6.6 sind die objektiven Messergebnisse aller untersuchten Algorithmen für ein 4-kanaliges, harmonisches Mikrofonarray mit der Rauschkonfiguration „Fahrgeräusche“ ersichtlich.

Die größte Verbesserung des SSNR erreichen die Algorithmen *MCC03,AA* und *MCC03*. Sie erzeugen allerdings die stärksten Sprachverzerrungen. Ebenso zeigen die Postfilter *MCC03,CN* und *MCC03,CN,AA* hohe SSNRE-Werte. Der auftretende Restrauschpegel, der nicht durch die Messungen quantifiziert werden kann, sondern durch Hörtests bewertet werden muss, wurde bei diesen Algorithmen jedoch als relativ störend empfunden. Dies galt übrigens für fast alle Algorithmen, die mit der Mindestfilterfunktion kombiniert wurden. Etwas geringere SSNRE-Ergebnisse, aber dafür auch geringere Sprachverzerrungen, wurden mit den Postfiltern *GMCC,AA* und *GMCC* erzielt. Im Mittelfeld liegen der *APES*- und der *APAB*-Algorithmus, sowie deren Kombination mit der Mindestfilterfunktion. Sie zeichnen sich durch eine geringe Sprachverzerrung aus. Die schlechteste Rauschunterdrückung zeigen der *SIM92*- und der *ZEL88*-Algorithmus. Vergleicht man die Ergebnisse des 4-kanaligen, äquidistanten Mikrofonarrays mit denen des 8-kanaligen, harmonischen Mikrofonarrays, dann sieht man, dass aus einer größeren Anzahl an Mikrofone erstaunlicherweise beinahe kein höheres SSNRE hervorgeht (siehe Tabelle 6.5). Bei manchen Algorithmen ergeben sich sogar geringere SSNRE-Werte, wie zum Beispiel für den SDB. Dies kann folgendermaßen erklärt werden: Durch Schallreflexionen aufgrund von Straßenbegrenzungen am rechten Straßenrand, wie zum Beispiel Lärmschutzwänden, Böschungen oder Wällen, kam es zur Ausbildung eines starken kohärenten Störers. Dieser Störer traf nahezu aus derselben Richtung auf das Array wie das Sprachsignal. Diese Vermutung konnte durch eine Messung der Gesamtenergie des Rauschsignals, $E_{n_i} = \sum_{\forall k} n(k)^2$, jedes Kanals i bestätigt werden. Die meiste Energie war in jenem Mikrofon vorzufinden, das dem Sprecher am nächsten war. Im Gegensatz zum DSB sind die

	4 - Kanal, äquidistant Broadside		4 - Kanal, äquidistant Endfire		8 - Kanal, harmonisch Broadside		8 - Kanal, harmonisch Endfire	
[dB]	SSNRE	SD	SSNRE	SD	SSNRE	SD	SSNRE	SD
Algorithmen	$SSNR_{in} = 10$ dB							
DSB	0.51	–	1.15	–	2.33	–	3.28	–
SDB	1.45	–	3.95	–	3.49	–	7.20	–
ZEL88	1.70	0.55	4.74	1.17	5.07	1.44	9.30	1.17
SIM92	1.35	0.46	3.83	0.70	3.61	0.51	7.22	0.65
APES	1.78	0.70	5.57	1.07	3.91	1.2	8.78	0.91
APAB	1.83	0.57	5.32	0.92	3.44	1.21	8.32	0.73
GMCC	3.84	1.19	6.83	1.20	7.15	1.21	10.25	1.09
MCC03	3.84	1.19	9.76	3.02	7.15	1.21	11.57	3.29
Algorithmen	$SSNR_{in} = 15$ dB							
DSB	0.51	–	1.15	–	2.33	–	3.28	–
SDB	1.45	–	3.95	–	3.49	–	7.20	–
ZEL88	1.67	0.34	4.67	0.86	4.82	1.2	9.01	0.87
SIM92	1.39	0.39	3.85	0.58	3.59	0.4	7.28	0.55
APES	1.69	0.56	5.19	0.95	3.66	1.18	8.21	0.80
APAB	1.71	0.50	4.97	0.90	3.28	1.21	7.86	0.70
GMCC	3.43	1.09	6.49	1.05	6.41	1.03	9.73	0.88
MCC03	3.43	1.09	9.04	2.97	6.41	1.03	10.74	3.24
Algorithmen	$SSNR_{in} = 20$ dB							
DSB	0.51	–	1.15	–	2.33	–	3.28	–
SDB	1.45	–	3.95	–	3.49	–	7.20	–
ZEL88	1.64	0.22	4.61	0.68	4.57	1.06	8.78	0.69
SIM92	1.44	0.35	3.88	0.50	3.57	0.33	7.34	0.50
APES	1.60	0.49	4.86	0.91	3.42	1.17	7.77	0.75
APAB	1.58	0.46	4.66	0.88	3.14	1.2	7.49	0.69
GMCC	3.07	1.05	6.24	0.97	5.79	0.94	9.34	0.77
MCC03	3.07	1.05	8.45	3.01	5.79	0.94	9.98	3.20

Tabelle 6.4: Vergleich der Algorithmen anhand der Ergebnisse der objektiven Messungen ($SSNRE$, SD) für die Büroraumumgebung.

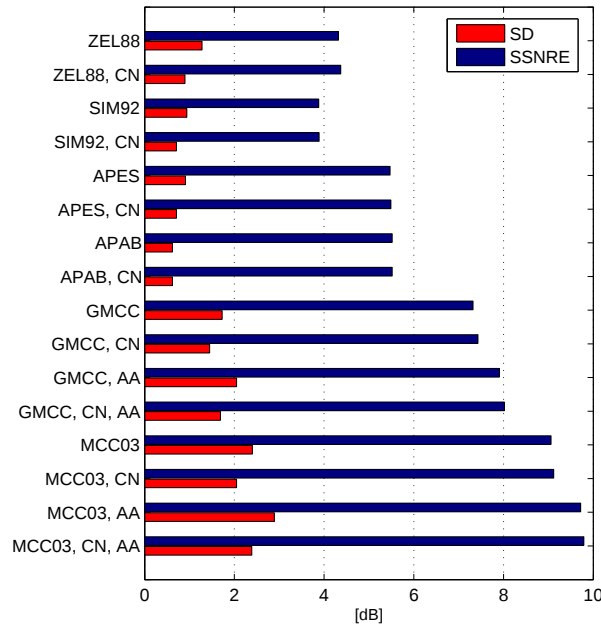


Abbildung 6.6: *SSNR Verbesserung (SSNRE) und Maß der Sprachverzerrung (SD) zum Vergleich der Algorithmenqualität in der automatischen Umgebung (4-kanaliges, äquidistantes Mikrofonarray; Rauschkonfiguration: „Fahrgeräusche“).*

Koeffizienten des SDB vom vorhandenen Rauschfeld abhängig. Da Sprache und Störer aus derselben Richtung kamen, konnte durch den SDB keine verbesserte Keulenformung erzielt werden. Die schlechten Eigenschaften des SDB wirkten sich überdies auf die Postfilter aus. Zusätzlich muss beachtet werden, dass der SDB, gemäß den Erkenntnissen aus Abschnitt 3.5.3, bei einem Einfallswinkel von $\theta_0 = 45^\circ$ Bereiche mit Leistungsverstärkungen größer 0 dB ausbildet. Rauschen, welches in diesen Bereichen auftritt, wird demnach verstärkt. All diese Effekte beeinflussten die Eigenschaften der Rauschunterdrückungsalgorithmen stark und führten dazu, dass sie nicht mehr die erwartete Leistungsfähigkeit erreichten.

Das 4-kanalige, äquidistante Array zeigte aufgrund der geringen Mikrofonabstände eine gleichmäßige Verteilung der Rauschenergie über alle Mikrofone. Der kohärente Störer wurde somit vom Array nicht als solcher „erkannt“. Größere Mikrofonabstände würden auch beim 4-kanaligen Arraydesign die Leistungsfähigkeit des SDB und damit der Postfilteralgorithmen herabsetzen. Betrachtet man die Tabelle C.9 im Anhang, dann sieht man, dass die SSNRE-Werte des 8-kanaligen, harmonischen Arrays für die Rauschsituation „Offenes Fenster“ größer sind als jene des 4-kanaligen, äquidistanten Arrays. Aufgrund des offenen Fensters auf der Fahrerseite stellte sich wieder nahezu ein homogenes Rauschfeld ein, worauf auch der SDB eine bessere Leistungsfähigkeit erzielte.

Es lassen sich nun folgende Erkenntnisse ableiten: Das Rauschfeld in automatischer Umgebung kann sehr stark variieren. Durch Straßenbegrenzungen und vorbeifahrende Autos kann ein perfekt homogenes, diffuses Rauschfeld nicht immer garantiert werden. Gerade dies wäre

Algorithmen	4 - Kanal, äquidistant		8 - Kanal, harmonisch	
	SSNRE] [dB]	SD [dB]	SSNRE [dB]	SD [dB]
DSB	1.68	–	3.15	–
SDB	4.35	–	4.05	–
ZEL88	4.32	1.28	4.62	1.82
ZEL88, CN	4.37	0.9	4.81	1.23
SIM92	3.88	0.94	4.03	1.2
SIM92, CN	3.89	0.71	4.05	0.9
APES	5.47	0.91	4.39	1.17
APES, CN	5.49	0.71	4.42	0.96
APAB	5.52	0.62	4.45	0.77
APAB, CN	5.52	0.62	4.45	0.77
GMCC	7.32	1.73	6.87	2.06
GMCC, CN	7.43	1.45	7.21	1.86
GMCC, AA	7.91	2.05	7.35	2.24
GMCC, CN, AA	8.02	1.69	7.7	2.12
MCC03	9.06	2.4	8.88	2.69
MCC03, CN	9.12	2.05	9	2.88
MCC03, AA	9.72	2.89	8.91	3.17
MCC03, CN, AA	9.79	2.39	8.49	3.67

Tabelle 6.5: Vergleich der Algorithmen anhand der Ergebnisse der objektiven Messungen (SSNRE, SD) für die automotive Umgebung (Rauschkonfiguration: „Fahrgeräusche“).

aber für eine optimale Leistungsfähigkeit des verwendeten SDB, und folglich für eine effiziente Rauschunterdrückung durch die verschiedenen Postfilter, notwendig.

6.4 Conclusio

Betrachtet man die Ergebnisse der objektiven Messungen, dann erkennt man, dass eine Interpretation, sowie der Vergleich mit der subjektiven Studie, sehr schwierig ist. Trotzdem spiegeln die quantitativen Werte der objektiven Messungen die Ergebnisse der Teststudie einigermaßen wieder und ermöglichen somit eine genauere Analyse der Algorithmen. Zusätzlich sind individuelle Hörtests notwendig, um Effekte, die nicht in die objektiven Messungen eingehen, wie Nachhall oder Musical Noise, festzustellen und in die Evaluierung der Algorithmenqualität einzubeziehen.

Das Postfilter *GMCC*, welches eine korrigierte Version des Postfilters *MCC03* von McCowan ist, zeigt in Kombination mit dem superdirektiven Beamformer in allen Konfigurationen die besten Eigenschaften, um das Rauschen einer Büroumgebung effizient zu unterdrücken. Abhängig von der Position des Sprechers zum Mikrofonarray erzielen auch die Postfilter *ZEL88* und *APES* sehr gute Ergebnisse. Für niedriges Eingangs-SSNR ist bei allen Algorithmen mit hohen Sprachverzerrungen zu rechnen. Die Anwendung der Mindestfilterfunktion und/oder des adaptiven Glättungsfaktors hilft, unerwünschte Effekte, wie starke Sprachverzerrungen, Musical Noise oder Nachhall, zu reduzieren.

In der automotiven Umgebung tritt aufgrund des stark instationären Rauschens fast bei allen Algorithmen Musical Noise auf. Das Postfilter *MCC03* wurde trotz dem erhöhten Maß an Sprachverzerrungen von den Testpersonen am angenehmsten empfunden. Eine ebenso gute Rauschunterdrückung und Bewertung durch die Teststudie zeigt das Postfilter *GMCC*, jedoch mit geringeren Sprachverzerrungen.

Die analysierten Postfilterstrukturen zeigen in diffusen Rauschfeldern sehr gute Rauschunterdrückungseigenschaften. Dies haben vor allem die Tests in der Büroraumumgebung bewiesen. Probleme treten jedoch dann auf, wenn kohärente Störer vorhanden sind. Bei allen Algorithmen wurden starke Leistungseinbrüche festgestellt, sobald starke Störer aus derselben Richtung wie das gewünschte Sprachsignal auf das Mikrofonarray gelangen. Hier kommt es auch bei hoher Mikrofonanzahl zu keiner Trennung zwischen dem gewünschten Sprachsignal und dem Rauschen, wodurch das nachfolgende Postfilter keine ausreichende Rauschunterdrückung mehr erreicht. Die Messungen, welche im Auto durchgeführt wurden, haben dies bestätigt.

Kapitel 7

Ausblick

Das abschließende Kapitel dokumentiert weitere Konfigurationen und Methoden zur Analyse der Eigenschaften von Beamformer und Postfilter, auf die in dieser Arbeit aufgrund des begrenzten Umfangs bzw. der begrenzt vorhandenen technischen Ausrüstung nicht näher eingegangen wurde. Zudem werden einige Ansätze und Verfahren zur Verbesserung von Postfilteralgorithmen vorgestellt.

Individuelle Hörtests haben gezeigt, dass die Postfilteralgorithmen abhängig von der Nachhallzeit¹ der akustischen Umgebung stark in ihrer Fähigkeit der Rauschunterdrückung variieren. Fischer und Kammeyer präsentieren diese Abhängigkeit in [26]. Demnach bewirken die Postfilter zusätzlich zum Beamformer bei kurzen Nachhallzeiten fast keine weitere Unterdrückung des Rauschfeldes. Bei langen Nachhallzeiten erscheint das Rauschfeld mehr und mehr diffus, sodass das Postfilter eine starke Verbesserung des SNR leistet. In dieser Diplomarbeit wurde diese Abhängigkeit nicht näher untersucht, da die Analyse realer Rauschumgebungen im Vordergrund stand. Unterschiedliche Nachhallzeiten können nur durch die Simulation akustischer Umgebungen erreicht werden. Die akustischen Eigenschaften eines geschlossenen Raumes werden üblicherweise mit Hilfe der Image Methode von Allen [27] simuliert. Diese Methode berechnet für einen beliebigen Raum die Raumimpulsantworten von einer Sprachquelle zu den Mikrofonen. Durch Filterung können dann die zugehörigen Eingangssignale an den Mikrofonen erzeugt werden. Es wäre daher interessant, auch eine Analyse der vorgestellten Postfilteralgorithmen bezüglich dieser Abhängigkeit durchzuführen.

In Abschnitt 3.4 wird die Möglichkeit erwähnt, den superdirektiven Beamformer anhand eines frequenzabhängigen Parameters μ zu regularisieren. Dadurch kann das WNG auf einen konstanten Mindestwert beschränkt werden. In dieser Arbeit wurde diese Methode nicht implementiert. Dennoch ist eine verbesserte Leistung des Beamformers zumindest für bestimmte Rauschsituationen vorstellbar.

Es besteht die Möglichkeit Beamformer zu entwerfen, deren Richtcharakteristik Nullstellen in bestimmte Richtungen aufweist. In Rauschumgebungen, in denen die Positionen kohärenter

¹engl.: Reverberation Time

Störquellen bekannt ist, kann durch einen solchen Entwurf eine noch effizientere Störgeräuschunterdrückung erzielt werden. Das Design dieser Beamformer wird ausführlich in [11, 12] behandelt.

Im Zuge der Arbeit wurden Subband-Arrays mit 7 Mikrofonen getestet. Da bei einem Array dieser Größe allerdings immer nur 3 Mikrofone gleichzeitig aktiv sind, ergibt sich eine sehr breite Hauptkeule und eine geringe Direktivität. Die Konfiguration mit 7 Mikrofonen zeigt trotz der Vermeidung von räumlichen Aliasing und Grating Lobes keine bessere Leistung als ein 4-kanaliges, äquidistantes Mikrofonarray. Ab 9 Mikrofone wird die Anwendung eines Subband-Arrays interessant und ist gemäß [4] zu bevorzugen. Für die Experimente dieser Diplomarbeit standen nicht mehr als 8 Mikrofone zur Verfügung, wodurch keine detaillierte Analyse von Subband-Arrays erfolgte.

Eine umfangreiche Literaturrecherche und die Ergebnisse im Zuge dieser Diplomarbeit haben ergeben, dass objektive Messgrößen zur Untersuchung der Sprachqualität von *mehrkanaligen* Rauschunterdrückungsalgorithmen noch genauer untersucht werden sollten. Die Anwendung solcher Messverfahren ist oft sehr schwierig, da viele Parameter, wie die perfekte zeitliche Synchronisierung der verglichenen Signale oder unterschiedliche Rauschfelder und Rauschspektren, die Messergebnisse stark beeinflussen und mitunter unbrauchbar machen können. Dies gilt vor allem für Verfahren, bei denen ein direkter Vergleich zweier Signale erfolgt, wie zum Beispiel bei der LAR-Messung, der Itakura-Saito Messung oder der Weighted-Spectral Slope Messung [21]. Daher wäre es notwendig geeignete, objektive Messverfahren für mehrkanalige Algorithmen zu finden, wobei eine hohe Korrelation mit subjektiven Messungen gewünscht ist.

Sämtliche untersuchte Postfilteralgorithmen wurden für stationäre, zeitinvariante Rauschfelder getestet. Nicht alle Rauschumgebungen zeigen ein stationäres Verhalten, wie die Analyse der automotiven Umgebung bestätigt hat. Besteht die Möglichkeit die Kohärenzmatrix adaptiv zu berechnen, dann kann sich der MVDR Beamformer automatisch und optimal auf die gegebene Rauschsituation einstellen. Die Lösung eines solchen robusten, zeitvarianten Designs des MVDR Beamformers wird in [28] vorgestellt. Zur Berechnung der Kohärenzmatrix ist eine Schätzung der Rauschleistungsdichtespektren erforderlich.

Dafür stehen zwei Methoden zur Verfügung:

- **Sprachpausendetektor (VAD²).** Dieser Detektor erkennt Sprachpausen und ermöglicht eine Schätzung der Rauschleistungsdichtespektren zu diesen Zeitpunkten. Die Kohärenzmatrix kann damit nur in Sprachpausen adaptiert werden. Verschiedene Sprachpausendetektoren werden in [3] vorgestellt.
- **Minimale Statistik (MS).** Dieser ausgefeilte Algorithmus von Rainer Martin, präsentiert in [29] und [30], erlaubt eine sehr genaue Schätzung der Rauschleistungsdichtespektren durch Beobachten der Minima, die im Leistungsdichtespektrum eines verrauschten

²engl.: Voice Activity Detector

Sprachsignals auftreten. Die Annahme, dass in einem Minimum die Leistung eines verrauschten Sprachsignals immer zur reinen Rauschleistung hin abnimmt, ist aber nicht für die Kreuzleistungsdichten gültig. Die Methode in [28] zeigt Lösung dieses Problems.

Nicht nur der Beamformer, sondern auch das Postfilter, sollte auf eine veränderliche Rauschsituation angepasst werden. Ein Algorithmus, der dies ermöglicht, ist das Postfilter von McCowan (siehe Abschnitt 4.3). Dieser Algorithmus benutzt direkt die Kohärenzfunktion zur Berechnung der Filterkoeffizienten und beeinflusst damit die Leistungsfähigkeit des Postfilters bezüglich des vorhandenen Rauschfeldes. Somit ist bei diesem Postfilter eine adaptive Anpassung auf das Rauschfeld möglich. Je exakter das aktuelle Rauschfeld durch die Kohärenzmatrix repräsentiert wird, desto besser sollte die Rauschunterdrückung funktionieren. Dieser Ansatz könnte einen neuen, vielversprechenden Postfilteralgorithmus zur Entstörung von Sprache hervorbringen.

Anhang A

Akronyme und Abkürzungen

AA	Adaptives Alpha
APAB	Adaptive Postfilter for an Arbitrary Beamformer
APES	Adaptive Postfilter Extension for Superdirective Beamformers
BS	Broadside-Konfiguration
CN	Comfort Noise
DFT	Diskrete Fouriertransformation
DI	Direktivitätsindex
DSB	Delay&Sum Beamformer
DUT	Device Under Test
EF	Endfire-Konfiguration
FFT	Fast Fourier Transformation
GCOW	Generalized McCowan 2003 Postfilter
GSC	General Sidelobe Canceller
GZEL	Generalized Zelinski 1988 Postfilter
LAR	Log-Area Ratio
LDS	Leistungsdichtespektrum
LP	Linear Prediction
MCC03	McCowan 2003 Postfilter
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MOS	Mean Opinion Score
MVDR	Minimum Variance Distortionless Response
OLA	Overlap-Add
PARCOR	Partial Correlation Coefficient
PC	Personal Computer
SD	Speech Degradation
SDB	Superdirektiver Beamformer

SIM92	Simmer 1992 Postfilter
SNR	Signal-to-Noise Ratio
SSNR	Segmental Signal-to-Noise Ratio
SSNRE	Segmental Signal-to-Noise Ratio Enhancement
STFT	Short-Time Fourier Transformation
WNG	White Noise Gain
ZEL88	Zelinski 1988 Postfilter

Anhang B

Ergänzende Definitionen und Herleitungen

B.1 Ergänzungen zu Abschnitt 2.3

B.1.1 Ableitung nach einem Vektor

Das i -te Element w_i eines Vektors $\boldsymbol{w}$ setzt sich aus dem Realteil a_i und dem Imaginärteil b_i zusammen, wobei i im Bereich $[1, K]$ liegt. Jedes Element des Vektors $\boldsymbol{w}$ ist somit eine komplexe Größe gemäß

$$w_i = a_i + j b_i. \quad (\text{B.1})$$

Man erhält damit eine Funktion der reellen Größen a_i und b_i . Mit Hilfe von Gl. B.1 kann man den Realteil a_i als Funktion der konjugiert komplexen Paare w_i und w_i^* durch

$$a_i = \frac{1}{2} (w_i + w_i^*) \quad (\text{B.2})$$

und den Imaginärteil b_i durch

$$b_i = \frac{1}{2j} (w_i - w_i^*) \quad (\text{B.3})$$

ausdrücken. Die reellen Größen lassen sich somit als Funktion von w_i und w_i^* darstellen.

Man kann nun die komplexen Ableitungen in Form von reellen Ableitungen definieren:

$$\frac{\partial}{\partial w_i} = \frac{1}{2} \left(\frac{\partial}{\partial a_i} - j \frac{\partial}{\partial b_i} \right) \quad (\text{B.4})$$

und

$$\frac{\partial}{\partial w_i^*} = \frac{1}{2} \left(\frac{\partial}{\partial a_i} + j \frac{\partial}{\partial b_i} \right). \quad (\text{B.5})$$

Diese Ableitungen erfüllen die folgenden zwei Bedingungen:

$$\frac{\partial w_i}{\partial w_i} = 1 \quad (\text{B.6})$$

$$\frac{\partial w_i}{\partial w_i^*} = \frac{\partial w_i^*}{\partial w_i} = 0. \quad (\text{B.7})$$

Nun betrachtet man die Ableitung nach einem Vektor. Dabei sind $w_1, w_2, \dots, w_K$ die Elemente des komplexen $K \times 1$ Vektors $\mathbf{w}$. Mit den Gleichungen B.4 und B.5 lässt sich $\frac{\partial}{\partial \mathbf{w}}$ als Ableitung bezüglich des Vektors $\mathbf{w}$ durch

$$\frac{\partial}{\partial \mathbf{w}} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial a_1} - j \frac{\partial}{\partial b_1} \\ \frac{\partial}{\partial a_2} - j \frac{\partial}{\partial b_2} \\ \vdots \\ \frac{\partial}{\partial a_K} - j \frac{\partial}{\partial b_K} \end{bmatrix} \quad (\text{B.8})$$

und $\frac{\partial}{\partial \mathbf{w}^*}$ als konjugierte Ableitung bezüglich des Vektors $\mathbf{w}$ durch

$$\frac{\partial}{\partial \mathbf{w}^*} = \frac{1}{2} \begin{bmatrix} \frac{\partial}{\partial a_1} + j \frac{\partial}{\partial b_1} \\ \frac{\partial}{\partial a_2} + j \frac{\partial}{\partial b_2} \\ \vdots \\ \frac{\partial}{\partial a_K} + j \frac{\partial}{\partial b_K} \end{bmatrix} \quad (\text{B.9})$$

anschreiben. Die Ableitungen gehorchen dabei den folgenden Beziehungen:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{w}} = \mathbf{I} \quad (\text{B.10})$$

und

$$\frac{\partial \mathbf{w}}{\partial \mathbf{w}^*} = \frac{\partial \mathbf{w}^*}{\partial \mathbf{w}} = \mathbf{0}, \quad (\text{B.11})$$

wobei $\mathbf{I}$ der Einheitsmatrix und $\mathbf{0}$ der Nullmatrix entspricht.

B.1.2 Beziehung zwischen der Ableitung nach einem Vektor und dem Gradientenvektor

Geht man von komplexen Filterkoeffizienten w_i aus, dann kann man diese gemäß Gl. B.1 ausdrücken, wobei i wieder im Bereich $[1, K]$ liegt. Man definiert nun den Gradientenoperator ∇ für den i -ten Filterkoeffizienten. Dieser kann als partielle Ableitung erster Ordnung bezüglich des Realteils a_i und des Imaginärteils b_i gemäß

$$\nabla_i = \frac{\partial}{\partial a_i} + j \frac{\partial}{\partial b_i} \quad (\text{B.12})$$

angeschrieben werden kann.

Wendet man den Gradientenoperator auf die Fehlerleistung ϕ_{ee} an, dann erhält man den komplexen Gradientenvektor $\nabla_{\mathbf{w}}(\phi_{ee})$, wobei das i -te Element durch

$$\nabla_{w_i}(\phi_{ee}) = \frac{\partial(\phi_{ee})}{\partial a_i} + j \frac{\partial(\phi_{ee})}{\partial b_i} \quad (\text{B.13})$$

definiert ist. Damit diese Definition des Gradientenvektors gültig ist, muss ϕ_{ee} notwendigerweise reell sein.

Vergleicht man nun den Gradientenvektor

$$\nabla_{\mathbf{w}}(\phi_{ee}) = \begin{bmatrix} \frac{\partial(\phi_{ee})}{\partial a_1} + j \frac{\partial(\phi_{ee})}{\partial b_1} \\ \frac{\partial(\phi_{ee})}{\partial a_2} + j \frac{\partial(\phi_{ee})}{\partial b_2} \\ \vdots \\ \frac{\partial(\phi_{ee})}{\partial a_K} + j \frac{\partial(\phi_{ee})}{\partial b_K} \end{bmatrix} \quad (\text{B.14})$$

mit der konjugierten Ableitung $\frac{\partial(\phi_{ee})}{\partial \mathbf{w}^*}$ (siehe Gl. B.9), erkennt man folgende Beziehung:

$$\nabla_{\mathbf{w}}(\phi_{ee}) = 2 \frac{\partial(\phi_{ee})}{\partial \mathbf{w}^*} \quad (\text{B.15})$$

Der Gradientenvektor ist mit Ausnahme eines skalaren Faktors gleich der konjugierten Ableitung aus Gl. B.9.

B.1.3 Ableitung der Fehlerfunktion

Um das Minimum des quadratischen Fehlers

$$\phi_{ee} = E \{ \|e\|_2^2 \} = \phi_{ss} - \mathbf{w}^H \phi_{xs} - \phi_{xs}^H \mathbf{w} + \mathbf{w}^H \Phi_{xx} \mathbf{w} \quad (\text{B.16})$$

zu bestimmen, muss der Gradientenvektor von ϕ_{ee} gebildet werden. Dieser ergibt sich mit Hilfe von Gl. B.15 zu

$$\begin{aligned} \nabla_{\mathbf{w}}(\phi_{ee}) &= 2 \frac{\partial \phi_{ee}}{\partial \mathbf{w}^*} \\ &= 2 \underbrace{\frac{\partial}{\partial \mathbf{w}^*}(\phi_{ss})}_{=0} - 2 \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H \phi_{xs}) - 2 \underbrace{\frac{\partial}{\partial \mathbf{w}^*}(\phi_{xs}^H \mathbf{w})}_{=0} + 2 \frac{\partial}{\partial \mathbf{w}^*}(\mathbf{w}^H \Phi_{xx} \mathbf{w}) \\ &= -2\phi_{xs} + 2\Phi_{xx} \mathbf{w}, \end{aligned} \quad (\text{B.17})$$

wobei Gl. B.10 und Gl. B.11 zur Vereinfachung des Ausdrucks verwendet wurden und $\mathbf{0}$ der Nullvektor ist.

B.1.4 Sherman-Morrison-Woodbury Formel

Man nimmt an, dass $\mathbf{Y}$ und $\mathbf{B}$ zwei positiv-definite $M \times M$ Matrizen sind, die gemäß

$$\mathbf{Y} = \mathbf{A}^{-1} + \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^H \quad (\text{B.18})$$

zusammenhängen, wobei $\mathbf{C}$ eine weitere positiv-definite $N \times M$ Matrix ist und $\mathbf{B}$ eine $M \times N$ Matrix.

Nun lässt sich gemäß der *Sherman-Morrison-Woodbury Formel* die Inverse von Matrix $\mathbf{Y}$ durch

$$\mathbf{Y}^{-1} = \mathbf{A} - \mathbf{A}\mathbf{B}(\mathbf{C} + \mathbf{B}^H\mathbf{A}\mathbf{B})^{-1}\mathbf{B}^H\mathbf{A} \quad (\text{B.19})$$

ausdrücken. Der Beweis dieser Formel kann durch Multiplikation von Gl. B.18 mit Gl. B.19 nachvollzogen werden. Die *Sherman-Morrison-Woodbury Formel* zeigt, dass für eine gegebene Matrix $\mathbf{Y}$, definiert durch Gl. B.18, die Inverse Matrix $\mathbf{Y}^{-1}$ mit Hilfe von Gl. B.19 berechnet werden kann. In der Literatur ist diese Formel auch als *Matrix Inversion Lemma* bekannt.

B.2 Levinson-Durbin Rekursion

Die Berechnung der PARCOR Koeffizienten erfolgt mit Hilfe der Levinson-Durbin Rekursion. Die Koeffizienten ergeben sich durch eine so genannte Linear Prediction (LP) Analyse, die für diese Arbeit nicht weiters relevant ist. Eine detailliertere Beschreibung dieser Analyse findet man in [21, 23].

Das betrachtete Signal $x(k)$ wird in Frames der Länge L zerteilt. Für jeden Signalframe $x_f(k, m)$ mit Frameindex m gilt:

$$x_f(k, m) = 0, \quad \text{für } k < 0 \text{ und } k > L. \quad (\text{B.20})$$

Pro Frame wird dann ein Set von P PARCOR Koeffizienten berechnet, wobei üblicherweise $P = 14$ gewählt wird. P wird oft auch als Modellordnung bezeichnet. Die Autokorrelationsfunktion $R(l, m)$ für jeden Frame ergibt sich durch

$$R(l, m) = \sum_{i=l}^{L-1} x_f(i, m)x_f(i-l, m), \quad \text{für } 0 \leq l \leq P. \quad (\text{B.21})$$

Sie entspricht einer gerade Funktion mit $R(l, m) = R(-l, m)$. Der Levinson-Durbin Algorithmus löst folgendes Set von linearen Gleichungen rekursiv für $p = 1, \dots, P$:

$$\kappa(p, m) = \left[R(p, m) - \sum_{i=1}^{p-1} a_i(p-1, m)R(p-i, m) \right] \frac{1}{E(p-1, m)}, \quad (\text{B.22})$$

$$a_p(p, m) = \kappa(p, m), \quad (\text{B.23})$$

$$a_i(p, m) = a_i(p-1, m) - \kappa(p, m)a_{p-i}(p-1, m), \quad \text{für } 1 \leq i < p, \quad (\text{B.24})$$

$$E(p, m) = (1 - \kappa(p, m)^2) E(p-1, m), \quad (\text{B.25})$$

wobei als Initialisierung $E(0, m) = R(0, m)$ gewählt wird. Daraus ergeben sich die gesuchten PARCOR Koeffizienten $\kappa(p, m)$ ($1 \leq p \leq P$) für jeden Frame m .

Anhang C

Evaluierungstabellen

Die nachfolgenden Tabellen zeigen die Rohdaten der *subjektiven Studie*. Die Tabellen C.1, C.2, C.3 beinhalten die Evaluierungsdaten für die Büroraumumgebung und die Tabellen C.4, C.5, C.6 die Daten für die automotive Umgebung.

Alle Messwerte der *objektiven Analyse* wurden in Tabellen zusammengefasst. Die SSNRE- und SD-Messungen der Büroraumumgebung sind für das 4-kanalige, äquidistante Array in Tabelle C.7 ersichtlich. Die Ergebnisse für das 8-kanalige, harmonische Array sind in Tabelle C.8 dargestellt. Die Messwerte für die automotive Umgebung sind in Tabelle C.9 aufgelistet.

		Titel	Bewertung																							
Testpersonnummer			1			2			3			4			5			6			7					
Alter			29			37			22			23			32			26			26					
Geschlecht			w			m			w			w			m			m			m					
Durchgang			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3			
SDB	BS	1	1.5	2	2.5	3	3	3	2	2.5	3	3	3	3	3	4	3	2.5	2	2	3	3	2			
SIM92, CN	EF	2	2.5	2.5	3	4	3.5	4	3.5	4	4	3	3	3	4	4	3.5	3.5	3	3	3.5	4	3			
ZEL88	EF	3	4.5	4	4.5	4.5	5	5	5	5	4.5	4	5	4	4.5	4.5	4.5	5	5	5	4	5	5			
GMCC	BS	4	3.5	2.5	3.5	5	4.5	4.5	4.5	4	4	3	4	3	2.5	4.5	4	4.5	4.5	4.5	3.5	5	5			
MCC03, CN	EF	5	2	2	2.5	2.5	2.5	2.5	2.5	2.5	2	2	3	3	2.5	4	2.5	2.5	2.5	3	3	3	2			
ZEL88	BS	6	3	2.5	4	3.5	2	3.5	3	2	3	2	2	2	3	4.5	3	3.5	3	4	2.5	4	2			
ZEL88, CN	EF	7	4	3.5	2.5	3	3.5	3	4	4.5	4	4	2	3	2.5	4	4	3	3	4	2.5	4.5	4			
MCC03	EF	8	3	2.5	4	4.5	4.5	5	3.5	5	4	4	4	3	3.5	3.5	4	3.5	4	4.5	3	3.5	3			
GMCC, AA	BS	9	3	3	3.5	5	3.5	4.5	2	4.5	3	3	3	3	4	3.5	3.5	4	4	4	4	5	5			
APES	BS	10	3	2	3	1.5	1.5	2	3	2	1.5	2	2	2	4.5	4.5	4	3	3	3.5	2.5	3	3			
GMCC, CN, AA	EF	11	4	2.5	2.5	1	2	3	1.5	3	3	3	2	3	3.5	3.5	2.5	2	2.5	3.5	2.5	3	2			
GMCC, CN, AA	BS	12	3	3	2.5	1.5	3.5	2	2	2	3	2	3	3	2.5	4	3.5	2	2.5	3	2.5	4	4			
GMCC	EF	13	4	3.5	4	3.5	5	4.5	4	4	5	4	4	4	4.5	4.5	4.5	3.5	5	4.5	3.5	4	3			
GMCC, CN	EF	14	3	2.5	2.5	2	3	3	3.5	2	3.5	3	3	3	4	4	4	2.5	2.5	4	3	4	3.5			
SIM92, BS	BS	15	2	2	2	1	1	1.5	1	1	1.5	1	2	1	3	3	4.5	1.5	2	3	2	2	2			
SDB	EF	16	2.5	2.5	2	1	1.5	1	1	1.5	3	2	2	2	2	2	3	2	2.5	3	2.5	2	2.5			
GMCC, AA	EF	17	3.5	4	3.5	4.5	4.5	5	4	4.5	4.5	4	4	3	5	3	4	4	4.5	4	3.5	4.5	4			
Original	–	18	1	1.5	1	1	1	1	1	1	1	1	1	1	1.5	1	2	1	1	1	1	1	1			
SIM92, CN	BS	19	2.5	2.5	2	1.5	2	2	2.5	1.5	1	3	2	2	3	3	3	2	1.5	2	2.5	3	3			
APES, CN	EF	20	3	3	3	4	4	4	4.5	3	2.5	4	3	3	4	3.5	3.5	3	3	3	2.5	3.5	3.5			
GMCC, CN	BS	21	3	3	2.5	3	3.5	3.5	4	3	2	3	3	2	3.5	4	3.5	2.5	2.5	3.5	3.5	4	3.5			
APES, CN	BS	22	3	2.5	2.5	2	2.5	3	2.5	1.5	1	2	2	2	3.5	4	3.5	3	2	3.5	3	3	2.5			
APES	EF	23	4.5	3.5	3.5	4.5	4	4.5	5	5	4	2	3	2	4.5	4.5	4.5	4	4.5	5	3.5	4	5			
SDB	BS	24	2	2	2	1.5	2	2.5	2	1	3	2	1	2	3	4	4	2	1.5	2	2	3	1.5			
SIM92	EF	25	3.5	3	3.5	3.5	2.5	3.5	3	2.5	3	3	2	3	3.5	4	4	3.5	3	3	3	4	3			
ZEL88, CN	BS	26	2.5	2.5	3	3	3.5	3	2	3	2	3	3	3	3	3.5	4	2.5	2.5	2.5	3	3	3			
APAB	EF	27	3.5	2.5	3.5	4	4	4	4	4	4	3	3	3	4	3.5	4	2.5	3.5	3.5	2.5	4.5	3.5			
SIM92, CN	EF	28	2.5	3	2.5	3	3	2.5	2.5	2	1.5	3	3	3	4	4	2.5	2.5	3	2.5	3	3	3			

Tabelle C.1: Rohdaten der subjektiven Studie für die Büroräumumgebung, Testperson 1 – 7.

		Titel	Bewertung																				
Testpersonnummer			8			9			10			11			12			13			14		
Alter			27			32			24			27			26			26			27		
Geschlecht			m			w			m			w			m			m			m		
Durchgang			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SDB	BS	1	3.5	4	3	4	3	3.5	3.5	4	3.5	3	3.5	3.5	3.5	3	3	3.5	3	2	5	4.5	3.5
SIM92, CN	EF	2	4	4.5	3.5	4	3.5	4	4	4.5	4	4	3.5	4.5	4	4	3	3.5	3	4	4.5	4	4.5
ZEL88	EF	3	5	4.5	5	5	4.5	5	5	5	5	5	5	5	5	5	4.5	5	5	5	4	5	5
GMCC	BS	4	4.5	5	4.5	4.5	4.5	5	3.5	3.5	4	4	4	4.5	4.5	4.5	4	4.5	4.5	4.5	3	4.5	4.5
MCC03, CN	EF	5	2.5	3	4	3.5	3.5	3.5	2	2.5	3	3.5	3	3.5	3	3	3	2	2.5	4	1	2	4
ZEL88	BS	6	3	4	3	3.5	3.5	4.5	3	3	2.5	4	2.5	3.5	3.5	4	3.5	4	4	4.5	1.5	2.5	4
ZEL88, CN	EF	7	3	3	3	3.5	3.5	3.5	2.5	3.5	2.5	4	4.5	4	3	3.5	3	2.5	2.5	4	2.5	3.5	3.5
MCC03	EF	8	4	3.5	4.5	3	4.5	4	2.5	3	3	4	4.5	3	3.5	4.5	4	5	4	4.5	3	3	5
GMCC, AA	BS	9	3.5	4	4.5	4.5	4.5	4.5	3	4	3	3	4	2.5	3	4	4	5	5	5	4	4.5	4.5
APES	BS	10	4	4	4	4.5	3.5	3.5	3.5	2.5	2.5	3	2.5	2	3	3.5	3	4	3.5	3.5	3.5	4	3
GMCC, CN, AA	EF	11	3.5	3	3	3.5	3	3	3	2.5	3.5	3	3.5	4	3.5	3	2.5	3	2.5	2.5	2	1.5	2
GMCC, CN, AA	BS	12	2.5	3	3	3	4	2.5	2.5	3	3.5	3.5	2.5	3	3	3.5	2.5	3	3	2	3.5	2.5	1.5
GMCC	EF	13	5	4	4.5	5	5	4.5	4.5	4.5	5	4.5	4.5	4.5	3.5	5	4.5	5	5	5	4.5	5	5
GMCC, CN	EF	14	4	2.5	4	3	3.5	3.5	3	3.5	3.5	3.5	3	3.5	3	3.5	3	3.5	3	3.5	2.5	3	3
SIM92	BS	15	1	2.5	2.5	3	3	2.5	2.5	3	3	2.5	2.5	2.5	4	3	2	3.5	3.5	4.5	1.5	2	1
SDB	EF	16	3	3	3	2.5	3	2.5	2	2.5	3	3	3	3	3	3	2	3	2.5	2.5	2	2.5	1.5
GMCC, AA	EF	17	4	4	4.5	5	5	4.5	5	4.5	5	5	5	4.5	4.5	4.5	4	5	4.5	5	2.5	5	5
Original	–	18	1	1.5	1	2	2	1.5	1.5	1.5	1	1	1	1	2	2	1.5	1	1	1	1	1	1
SIM92, CN	BS	19	2	3	3	2.5	2.5	2	3.5	2.5	3	2	3	2.5	3	2.5	2.5	2	2.5	3.5	2.5	3.5	2.5
APES, CN	EF	20	2.5	3.5	3.5	3	3.5	3	4	3.5	3.5	3	4	4.5	3.5	3	3	2.5	3.5	4.5	3	4	3
GMCC, CN	BS	21	3	3	3.5	3	3.5	3.5	3	3	3.5	3	3.5	3	3.5	3.5	3	2.5	3	4	3.5	3.5	3
APES, CN	BS	22	3	3.5	3.5	2.5	3.5	2.5	3	2.5	3.5	2.5	2.5	2.5	3	3.5	2.5	3	4	4	2.5	2.5	2
APES	EF	23	4	4	5	5	4.5	4	5	5	5	5	5	5	5	5	4.5	5	5	5	4	5	4.5
SDB	BS	24	2.5	3.5	3	2	3.5	2.5	2.5	3	3	2.5	2	2.5	3.5	3	2.5	2	2	2	2	3	1.5
SIM92	EF	25	3.5	3.5	4	3.5	4	3.5	3.5	5	4	4	3	4	3.5	4	2.5	3	4	4	3	2.5	2.5
ZEL88, CN	BS	26	3	3	3.5	3	3.5	3	3.5	3.5	3.5	3.5	3	3	4	3.5	3	3.5	3.5	3	3	1.5	2
APAB	EF	27	3	2.5	3.5	3.5	4	2.5	3.5	4	3	4	3.5	4	4	3.5	3.5	2.5	3	3.5	4	3	2.5
SIM92, CN	EF	28	2.5	2.5	3.5	3.5	3.5	3	3	4.5	2.5	4	4	3	3	3	3	4	4	3.5	2	2	2

Tabelle C.2: Rohdaten der subjektiven Studie für die Büroräumumgebung, Testperson 8–14.

		Titel	Bewertung																	
Testpersonnummer			8			9			10			11			12			13		
Alter			27			32			24			27			26			26		
Geschlecht			m			w			m			w			m			m		
Durchgang			1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
SDB	BS	1	4	3	3	3.5	3.5	3	3	3	3	3	2	2	3	2	1.5	3	2	3
SIM92, CN	EF	2	5	3.5	4.5	3.5	4	3.5	3.5	3.5	3.5	3.5	2.5	3.5	4	3.5	3	3	3	4
ZEL88	EF	3	5	5	5	4	3.5	4	5	4.5	4.5	5	5	5	5	5	4.5	5	5	
GMCC	BS	4	4	4.5	4	4	4	3.5	4	5	4	4.5	4.5	4.5	4.5	5	4.5	4.5	4.5	4
MCC03, CN	EF	5	3	2	2.5	2.5	3.5	3	3	2.5	3.5	4	3	3	2	3.5	3.5	2	3	2
ZEL88	BS	6	1	1.5	3.5	3	3	3	4.5	3	4	4	4.5	3.5	4.5	4.5	3.5	4	4	4
ZEL88, CN	EF	7	2	1.5	2.5	2.5	3.5	3	3.5	4	3.5	3	3.5	4	2.5	4	3	2	3	3
MCC03	EF	8	3.5	4	3.5	3.5	4.5	4	3	3.5	4.5	4.5	5	4.5	3.5	4.5	5	4	3	4
GMCC, AA	BS	9	3.5	5	4.5	4	4	4	4.5	4	4.5	4	4.5	4	4	5	4.5	4	4.5	4
APES	BS	10	1.5	3.5	2	3	3	2.5	3.5	3	3	3	3	3	3	4.5	4	3	2.5	2
GMCC, CN, AA	EF	11	2	3	2	2.5	3	3	2.5	2.5	3.5	3	3	3	2	4	3	2	2	2
GMCC, CN, AA	BS	12	1.5	1.5	1.5	3	2.5	3	3.5	3.5	4	2	3	3	1	3	2.5	1.5	2	2
GMCC	EF	13	3	5	5	4	3.5	4	4	4.5	5	5	4.5	4.5	4	5	5	5	4	4.5
GMCC, CN	EF	14	1.5	1	2	3	2.5	3	3	3.5	4	2.5	3	3	2	3	4	2	2	2
SIM92	BS	15	1	1	1	3	2	2	2.5	3	2.5	2	2	1.5	2	3	3	1	3	1
SDB	EF	16	1	1.5	1.5	3	2	2.5	2	2.5	2.5	2	2	1	1	2	3	1	1	1
GMCC, AA	EF	17	5	4.5	5	4	5	5	4.5	4	4.5	4	5	4.5	4.5	4.5	5	3	4	4
Original	–	18	1	1	1	1	1	1	1	1	1.5	1	1	1	1	1	1	1	1	1
SIM92, CN	BS	19	2	1.5	2	2.5	3	3	3	3	2	2	2.5	2	4	2.5	2.5	3	2	1.5
APES, CN	EF	20	2.5	2	2.5	3	3	3.5	4.5	3.5	4	3	4	4	3	3	3.5	3	2.5	2
GMCC, CN	BS	21	2	2.5	3	3	3.5	3.5	3.5	3.5	3.5	3	3.5	3.5	2	3	3	2	2.5	2
APES, CN	BS	22	1	2.5	3	3	2.5	2.5	3	3	3	3	3	2.5	2.5	2.5	2.5	2	3	2
APES	EF	23	5	4.5	5	4	3.5	3	4.5	4	5	4.5	5	5	4	4.5	5	4	4	4
SDB	BS	24	1.5	1	2	2.5	2	2.5	2	2.5	2.5	2.5	2	2	2	2	2	2	2	2
SIM92	EF	25	3	2.5	3	3.5	3	3	3	2.5	2.5	4	3	3.5	3	4	3	3.5	3.5	3
ZEL88, CN	BS	26	2.5	2	2	3	3	3	3	3	3.5	3	3.5	3.5	2.5	3	3	2.5	2	3
APAB	EF	27	2	2.5	1.5	3.5	3	3.5	3.5	3.5	4	4	4	3	3	4	3	2	2	2
SIM92, CN	EF	28	1.5	2	1.5	3.5	2.5	3	3	3	3	3	3	3	4	2.5	2	3	2	3

Tabelle C.3: Rohdaten der subjektiven Studie für die Büroräumumgebung, Testperson 15–20.

	Titel	Bewertung																				
Testpersonnummer		1			2			3			4			5			6			7		
Alter		29			37			22			23			32			26			26		
Geschlecht		w			m			w			w			m			m			m		
Durchgang		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
MCC03, CN	1	1.5	2.5	3	2	3	3	3	2	2	2	3	2	3	3.5	3	3	2.5	3	3	2	3
APES, CN	2	2	2	3	3	4	2	2	1.5	1	2	3	3	4	3.5	3.5	2	2	2.5	4	1	2
SIM92	3	2	2.5	3.5	2	2	3.5	3	2	1	3	2	3	4	4	4	1.5	1	2	4	3	3
MCC03	4	1.5	3	3.5	4	4.5	4	4.5	3.5	3	1	1	3	2	3	3	4	4	3.5	2	2	3.5
SIM92, CN	5	3	2	2	1	2	1.5	3	2	1.5	3	3	2	2.5	3.5	4	1	2	2.5	3	2	2
GMCC, AA	6	1.5	3	3.5	4	5	4.5	4	4	3.5	3	2	2	2	2.5	3	4	4	3.5	3	3	3.5
GMCC, CN	7	2.5	1.5	3	2	3.5	4	3.5	2	2	2	1	1	2	3	3	2	2.5	3.5	4	3	2
GMCC	8	2.5	3	3.5	5	4	4.5	4	3	2.5	3	3	2	3	3	2.5	3	4	3.5	3	3	3
MCC03, CN	9	1.5	2	2.5	3	2.5	3	3	2	2	1	3	3	3	3.5	3.5	1.5	2.5	3	4	2.5	2
MCC03, AA	10	1	3.5	4	4	4	5	4.5	4.5	4	4	4	4	3	2.5	3.5	4.5	5	5	3	3	5
APES, CN	11	3	2	2.5	1	1	1	1	1	1	2	2	3	4	3	3.5	1.5	2	2	3	2	2
Original	12	2	1	1	1	1	1	1	1	1	1	1	1	3	1.5	1	1	1	1	2	1	1
GMCC, CN, AA	13	2	2.5	3	4	3	3	3	2.5	3	3	3	2	3	2.5	3	2	2.5	2	4	3	3.5

Tabelle C.4: Rohdaten der subjektiven Studie für die automotive Umgebung, Testperson 1 – 7.

	Titel	Bewertung																				
Testpersonnummer		8			9			10			11			12			13			14		
Alter		27			32			24			27			26			26			27		
Geschlecht		m			w			m			w			m			m			m		
Durchgang		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
MCC03, CN	1	3	3	3	3.5	3.5	3.5	2.5	3	3	2	3	3	2.5	3	3	1	4	3	2	4	4
APES, CN	2	2	2	2.5	3	3	3	3	3	2	3	3	3.5	3	2	1.5	2.5	3.5	3	2.5	3	2
SIM92	3	2	2	2	3.5	3.5	2.5	4	2	2	2.5	3.5	3.5	3	2.5	2	2	4	2	1.5	2.5	2.5
MCC03	4	3	3.5	3.5	2.5	3	3.5	2.5	1.5	1.5	4	3	3	2	3.5	4	3	3	4	3	2	3.5
SIM92, CN	5	3	1.5	2	3	3	3	2	2.5	2	3.5	3	2	3	2.5	1.5	2.5	1.5	1	3.5	3	1.5
GMCC, AA	6	3	2.5	3	2	3.5	4	2	2.5	2	4	3	3	2	4	3.5	1.5	4	4	3.5	3	4
GMCC, CN	7	4	3	2.5	3.5	3.5	3	1.5	2.5	3	5	3.5	4	2.5	3	3	2	1	1	2.5	2	2
GMCC	8	4	2	2.5	3	3	3	3.5	3.5	2.5	3.5	2.5	3	2	3.5	4	3	3.5	3	2.5	4	4.5
MCC03, CN	9	3.5	2.5	3.5	3	2.5	3.5	2.5	1.5	2	3	4	4	1.5	2.5	2	1.5	1.5	1	2	3.5	3.5
MCC03, AA	10	4.5	4.5	4	4	3	4	4	2	1	4.5	4	4.5	3	4	4.5	4	3.5	4	4	5	5
APES, CN	11	2	2	3.5	3	2.5	2	3	2	3	3.5	2.5	2	3	2.5	1.5	1	1.5	1.5	3	2.5	2.5
Original	12	1	1	1	2.5	2	1.5	1.5	1	1.5	1.5	1	1	2.5	1.5	1	1	1	1	2.5	1	1
GMCC, CN, AA	13	3	3	3.5	3	3	3	3.5	2.5	3	3.5	4	4	2	3	3	2.5	1.5	1.5	2	2.5	2.5

Tabelle C.5: Rohdaten der subjektiven Studie für die automotiv Umgebung, Testperson 8–14.

	Titel	Bewertung																	
Testpersonnummer		8			9			10			11			12			13		
Alter		27			32			24			27			26			26		
Geschlecht		m			w			m			w			m			m		
Durchgang		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
MCC03, CN	1	3	3	2.5	3.5	4	3	2.5	3	2.5	3	3	3	1.5	2	3	2	3	3
APES, CN	2	2	1	1.5	2	2	2.5	4	2	2	2.5	2	1	2	3	3	3	2	2
SIM92	3	2	1.5	1	1.5	2	2	3	3.5	2	2.5	2	2	2.5	3.5	3.5	2	1	2
MCC03	4	1	4	3.5	3.5	3	3.5	5	4	3	5	4	4	2.5	4.5	4.5	2	3	3.5
SIM92, CN	5	1.5	1.5	1	3	1.5	3	3	3	2.5	3	1.5	2	3.5	1.5	2	1	2	1.5
GMCC, AA	6	1	2.5	4	4	3.5	3.5	3.5	4	3.5	4.5	4	4	3	4.5	5	3	3	4
GMCC, CN	7	1	1.5	3	3	3	3.5	3	3.5	3	4	3.5	3	2	2.5	2.5	2	2	3
GMCC	8	1	4.5	4.5	2.5	3	2.5	3	3.5	3.5	4	4	5	3	3.5	3	3	3	3.5
MCC03, CN	9	2	1	1.5	3.5	2	3.5	2.5	3	3	2.5	3	3	1.5	1	1.5	1.5	1.5	2
MCC03, AA	10	4	5	5	3.5	4.5	4.5	3.5	4.5	4.5	5	5	4	2.5	5	4.5	4	2	4
APES, CN	11	2	1	1	2	3	3	3.5	3.5	2	2	2	1.5	3	2	1	1	1	1
Original	12	1.5	1	1	1	1	1	2	1	1	1	1	1	1.5	1	1	1	1	1
GMCC, CN, AA	13	1	3	2.5	3.5	4	3	3	3	3	3	3	3	2.5	2	3.5	2	2	3

Tabelle C.6: Rohdaten der subjektiven Studie für die automotive Umgebung, Testperson 15–20.

	Broadside							Endfire						
$SSNR_{in}$ [dB]	-10	-5	0	5	10	15	20	-10	-5	0	5	10	15	20
Algorithmen	SSNRE [dB]													
DSB	0.51	0.51	0.51	0.51	0.51	0.51	0.51	1.15	1.15	1.15	1.15	1.15	1.15	1.15
SDB	1.45	1.45	1.45	1.45	1.45	1.45	1.45	3.95	3.95	3.95	3.95	3.95	3.95	3.95
ZEL88	1.62	1.69	1.71	1.70	1.67	1.64	1.61	4.74	4.81	4.80	4.74	4.67	4.61	4.58
ZEL88, CN	1.63	1.69	1.7	1.69	1.65	1.62	1.6	4.74	4.8	4.79	4.73	4.65	4.59	4.56
SIM92	1.26	1.28	1.32	1.35	1.39	1.44	1.50	3.83	3.81	3.81	3.83	3.85	3.88	3.90
SIM92, CN	1.26	1.28	1.31	1.35	1.39	1.44	1.49	3.82	3.8	3.8	3.81	3.83	3.86	3.89
APES	1.84	1.87	1.85	1.78	1.69	1.60	1.52	6.16	6.17	5.94	5.57	5.19	4.86	4.60
APES, CN	1.84	1.87	1.84	1.77	1.68	1.59	1.52	6.14	6.14	5.9	5.53	5.15	4.83	4.58
APAB	2.00	2.00	1.93	1.83	1.71	1.58	1.48	5.84	5.85	5.64	5.32	4.97	4.66	4.43
APAB, CN	1.82	1.83	1.79	1.73	1.64	1.56	1.49	5.83	5.84	5.63	5.31	4.96	4.66	4.43
GMCC	3.43	4.13	4.18	3.84	3.43	3.07	2.83	6.70	7.20	7.13	6.83	6.49	6.24	6.11
GMCC, CN	3.4	4.05	4.09	3.77	3.37	3.03	2.8	6.55	7.03	6.99	6.7	6.39	6.16	6.05
GMCC, AA	4.35	5.08	5	4.51	3.96	3.49	3.19	6.84	7.57	7.59	7.28	6.91	6.61	6.44
GMCC, CN, AA	4.24	4.88	4.81	4.37	3.86	3.43	3.14	6.64	7.33	7.38	7.11	6.78	6.52	6.36
MCC03	3.43	4.13	4.18	3.84	3.43	3.07	2.83	8.93	10.36	10.41	9.76	9.04	8.45	8.02
MCC03, CN	3.4	4.05	4.09	3.77	3.37	3.03	2.8	5.91	7.56	8.17	8.1	7.79	7.45	7.21
MCC03, AA	4.35	5.08	5	4.51	3.96	3.49	3.19	7.62	9.53	10.28	10.1	9.51	8.94	8.51
MCC03, CN, AA	4.24	4.88	4.81	4.37	3.86	3.43	3.14	4.72	6.57	7.83	8.19	8.05	7.8	7.6
Algorithmen	SD [dB]													
ZEL88	1.81	1.29	0.86	0.55	0.34	0.22	0.15	2.93	2.27	1.65	1.17	0.86	0.68	0.59
ZEL88, CN	1.23	0.91	0.61	0.4	0.26	0.18	0.13	1.91	1.58	1.23	0.93	0.72	0.61	0.55
SIM92	0.83	0.69	0.56	0.46	0.39	0.35	0.33	1.41	1.13	0.88	0.70	0.58	0.50	0.46
SIM92, CN	0.5	0.46	0.42	0.39	0.35	0.33	0.33	0.93	0.79	0.66	0.56	0.5	0.46	0.44
APES	1.75	1.29	0.93	0.70	0.56	0.49	0.46	2.25	1.69	1.29	1.07	0.95	0.91	0.89
APES, CN	1.43	1.1	0.83	0.65	0.54	0.48	0.46	1.48	1.21	1.01	0.91	0.87	0.87	0.87
APAB	1.11	0.88	0.69	0.57	0.50	0.46	0.44	1.30	1.11	0.99	0.92	0.90	0.88	0.88
APAB, CN	1.1	0.87	0.69	0.57	0.5	0.46	0.44	1.25	1.08	0.97	0.91	0.89	0.88	0.88
GMCC	2.32	1.81	1.42	1.19	1.09	1.05	1.04	2.57	2.01	1.52	1.20	1.05	0.97	0.94
GMCC, CN	1.79	1.36	1.13	1.02	1	1.01	1.02	1.79	1.48	1.22	1.04	0.95	0.92	0.91
GMCC, AA	2.72	2.10	1.64	1.32	1.17	1.11	1.10	3.1	2.42	1.77	1.33	1.14	1.05	1.01
GMCC, CN, AA	2.14	1.57	1.25	1.1	1.05	1.06	1.07	2.06	1.64	1.34	1.14	1.05	1.01	1
MCC03	2.32	1.81	1.42	1.19	1.09	1.05	1.04	3.21	3.39	3.16	3.02	2.97	3.01	3.01
MCC03, CN	1.79	1.36	1.13	1.02	1	1.01	1.02	4.38	3.48	2.97	2.81	2.8	2.84	2.86
MCC03, AA	2.72	2.1	1.64	1.32	1.17	1.11	1.1	2.5	3.46	3.51	3.26	3.17	3.22	3.26
MCC03, CN, AA	2.14	1.57	1.25	1.1	1.05	1.06	1.07	5.33	4.26	3.43	3.06	3.04	3.07	3.1

Tabelle C.7: Objektive Messungen: SSNR- und SD-Messungen im Büroraum für ein 4-kanaliges, äquidistantes Mikrofonarray (alle Angaben in dB).

	Broadside							Endfire						
$SSNR_{in}$ [dB]	-10	-5	0	5	10	15	20	-10	-5	0	5	10	15	20
Algorithmen	SSNRE [dB]													
DSB	2.33	2.33	2.33	2.33	2.33	2.33	2.33	3.28	3.28	3.28	3.28	3.28	3.28	3.28
SDB	3.49	3.49	3.49	3.49	3.49	3.49	3.49	7.20	7.20	7.20	7.20	7.20	7.20	7.20
ZEL88	5.2	5.33	5.27	5.07	4.82	4.57	4.38	9.74	9.79	9.60	9.30	9.01	8.78	8.63
ZEL88, CN	5.24	5.36	5.28	5.07	4.81	4.56	4.37	9.61	9.68	9.51	9.23	8.94	8.72	8.58
SIM92	3.6	3.62	3.63	3.61	3.59	3.57	3.55	7.17	7.16	7.17	7.22	7.28	7.34	7.40
SIM92, CN	3.61	3.62	3.62	3.6	3.58	3.56	3.54	7.15	7.14	7.15	7.19	7.26	7.33	7.39
APES	4.01	4.18	4.11	3.91	3.66	3.42	3.23	10.56	10.13	9.46	8.78	8.21	7.77	7.44
APES, CN	4.06	4.19	4.1	3.89	3.64	3.41	3.22	10.37	10	9.36	8.7	8.16	7.74	7.42
APAB	3.43	3.56	3.53	3.44	3.28	3.14	2.98	9.64	9.35	8.85	8.32	7.86	7.49	7.21
APAB, CN	3.67	3.77	3.71	3.57	3.39	3.22	3.07	9.62	9.33	8.83	8.31	7.85	7.48	7.21
GMCC	7.83	8.23	7.89	7.15	6.41	5.79	5.34	11.03	11.18	10.81	10.25	9.73	9.34	9.11
GMCC, CN	7.62	8.01	7.72	7.04	6.34	5.75	5.32	10.6	10.85	10.57	10.07	9.6	9.24	9.03
GMCC, AA	8.86	9.08	8.52	7.62	6.75	6.05	5.55	11.37	11.51	11.08	10.47	9.91	9.5	9.24
GMCC, CN, AA	8.33	8.68	8.27	7.48	6.67	6	5.52	10.81	11.11	10.81	10.28	9.78	9.39	9.16
MCC03	7.83	8.23	7.89	7.15	6.41	5.79	5.34	11.84	12.59	12.32	11.57	10.74	9.98	9.45
MCC03, CN	7.62	8.01	7.72	7.04	6.34	5.75	5.32	7.8	9.14	9.78	9.81	9.53	9.14	8.84
MCC03, AA	8.86	9.08	8.52	7.62	6.75	6.05	5.55	11	12.03	12.22	11.8	11.02	10.29	9.74
MCC03, CN, AA	8.33	8.68	8.27	7.48	6.67	6	5.52	6.74	8.14	9.34	9.82	9.69	9.38	9.09
Algorithmen	SD [dB]													
ZEL88	3.16	2.44	1.85	1.44	1.2	1.06	1	3.00	2.27	1.63	1.17	0.87	0.69	0.61
ZEL88, CN	1.95	1.61	1.3	1.05	0.89	0.8	0.76	1.81	1.48	1.13	0.85	0.66	0.56	0.51
SIM92	1.46	1.02	0.71	0.51	0.4	0.33	0.3	1.35	1.02	0.80	0.65	0.55	0.50	0.47
SIM92, CN	0.92	0.67	0.5	0.39	0.33	0.29	0.27	0.94	0.75	0.63	0.54	0.5	0.47	0.46
APES	1.77	1.44	1.28	1.2	1.18	1.17	1.16	1.92	1.43	1.11	0.91	0.80	0.75	0.73
APES, CN	1.21	1.08	1.06	1.08	1.11	1.12	1.13	1.31	1.04	0.87	0.77	0.73	0.72	0.71
APAB	1.42	1.29	1.23	1.21	1.21	1.2	1.2	1.05	0.89	0.78	0.73	0.70	0.69	0.68
APAB, CN	1.42	1.29	1.23	1.21	1.2	1.2	1.2	1.03	0.87	0.77	0.72	0.7	0.68	0.68
GMCC	2.86	2.15	1.57	1.21	1.03	0.94	0.91	2.61	1.99	1.46	1.09	0.88	0.77	0.72
GMCC, CN	2.12	1.51	1.11	0.88	0.75	0.69	0.66	1.76	1.38	1.07	0.86	0.74	0.69	0.67
GMCC, AA	3.22	2.45	1.75	1.31	1.09	0.97	0.93	2.97	2.25	1.61	1.18	0.93	0.8	0.75
GMCC, CN, AA	2.57	1.72	1.24	0.96	0.82	0.75	0.73	1.88	1.47	1.15	0.92	0.8	0.74	0.72
MCC03	2.86	2.15	1.57	1.21	1.03	0.94	0.91	3.00	3.26	3.30	3.29	3.24	3.20	3.18
MCC03, CN	2.12	1.51	1.11	0.88	0.75	0.69	0.66	5.05	4.2	3.68	3.4	3.27	3.21	3.18
MCC03, AA	3.22	2.45	1.75	1.31	1.09	0.97	0.93	2.83	3.58	3.72	3.64	3.62	3.61	3.59
MCC03, CN, AA	2.57	1.72	1.24	0.96	0.82	0.75	0.73	6.03	5.29	4.47	3.93	3.75	3.67	3.63

Tabelle C.8: Objektive Messungen: SSNR- und SD-Messungen im Büroraum für ein 8-kanaliges, harmonisches Mikrofonarray (alle Angaben in dB).

	Fahrgeräusche		Offene Dachluke		Offenes Fenstern	
$SSNR_{in}$ [dB]	-5.76		-6.47		-7.84	
Algorithmen	SSNRE	SD	SSNRE	SD	SSNRE	SD
4 - Kanal, äquidistant						
DSB	1.68	–	2.66	–	1.14	–
SDB	4.35	–	5.31	–	3.88	–
ZEL88	4.32	1.28	5.25	1.22	3.95	1.7
ZEL88, CN	4.37	0.9	5.3	0.84	4.04	1.3
SIM92	3.88	0.94	4.75	0.98	4.19	1.22
SIM92, CN	3.89	0.71	4.76	0.72	4.21	0.94
APES	5.47	0.91	6.46	0.96	4.16	1.24
APES, CN	5.49	0.71	6.48	0.75	4.2	1
APAB	5.52	0.62	6.51	0.65	4.34	0.77
APAB, CN	5.52	0.62	6.51	0.65	4.34	0.77
GMCC	7.32	1.73	8.11	1.67	12.29	3.1
GMCC, CN	7.43	1.45	8.21	1.44	12.27	2.86
GMCC, AA	7.91	2.05	8.66	1.96	12.1	3.57
GMCC, CN, AA	8.02	1.69	8.77	1.67	11.91	3.45
MCC03	9.06	2.4	9.93	2.36	12.63	3.65
MCC03, CN	9.12	2.05	10.03	2.08	12.4	3.53
MCC03, AA	9.72	2.89	10.65	2.89	11.12	3.97
MCC03, CN, AA	9.79	2.39	10.74	2.39	10.76	4.41
8 - Kanal, harmonisch						
DSB	3.15	–	4.07	–	3.57	–
SDB	4.05	–	5.3	–	4.49	–
ZEL88	4.62	1.82	5.8	1.75	5.93	2.03
ZEL88, CN	4.81	1.23	5.99	1.2	6.25	1.56
SIM92	4.03	1.2	5.06	1.19	5.55	1.35
SIM92, CN	4.05	0.9	5.08	0.88	5.57	1.03
APES	4.39	1.17	5.74	1.2	4.52	1.46
APES, CN	4.42	0.96	5.77	0.98	4.57	1.26
APAB	4.45	0.77	5.81	0.81	4.6	0.86
APAB, CN	4.45	0.77	5.81	0.81	4.6	0.86
GMCC	6.87	2.06	7.93	2	11.82	3
GMCC, CN	7.21	1.86	8.29	1.82	12.42	3.41
GMCC, AA	7.35	2.24	8.33	2.22	11.08	3.42
GMCC, CN, AA	7.7	2.12	8.71	2.09	11.31	4.39
MCC03	8.88	2.69	9.76	2.68	11.23	3.59
MCC03, CN	9	2.88	9.91	2.87	11.64	4.65
MCC03, AA	8.91	3.17	9.78	3.15	9.57	3.89
MCC03, CN, AA	8.49	3.67	9.44	3.56	9.88	5.57

Tabelle C.9: Objektive Messungen: SSNR- und SD-Messungen im Auto (alle Angaben in dB).

Anhang D

MATLAB Funktionen

D.1 Einleitung und Notation

Sämtliche Funktionen benötigen Ein- und Ausgangsvariablen. Eine Hilfe zu den Funktionen und ihren Parametern kann in MATLAB® mit dem Befehl `help funktionsname` aufgerufen werden.

Ein wichtiger Eingangsparameter ist die Position der Mikrofone. Um diese zu definieren, muss ein Koordinatensystem gewählt werden. Die Position jedes Mikrofons wird dann durch ihre x-Koordinate bestimmt, wobei die Maßeinheit in Meter angegeben wird. Diese Koordinaten werden in einem Zeilenvektor zusammengefasst. Jede Zeile der Matrix steht für die Position eines Mikrofons. Die Reihenfolge der Mikrofone muss mit den Kanälen der Audiodateien übereinstimmen. Prinzipiell sind alle Funktionen auch für 2- und 3-dimensionale Arrays anwendbar. Sie wurden jedoch nicht für diese Konfigurationen getestet. In diesem Fall werden die Mikrofonpositionen in einer Matrix zusammengefasst, wobei jede Zeile die Koordinaten eines Mikrofons enthält. Als Beispiel ergibt sich für ein 4-kanaliges, lineares, äquidistantes Array mit einem Mikrofonabstand von 2.5 cm folgenden Zeilenvektor:

```
mics =  
    -0.0375  
    -0.0125  
     0.0125  
     0.0375
```

In vielen Funktionen wird die *mat*-Datei `mics_8xh.mat` verwendet. Sie enthält die Matrix über die Mikrofonpositionen des 8-kanaligen, harmonischen Arrays, das in Abschnitt 5.1.1 beschrieben wurde.

Alle Audiosignale, die in den Funktionen verwendet werden, müssen in MATLAB® importiert werden. Hierbei eignen sich am besten Audiodateien im WAV-Format, auf die in MATLAB® über die Funktion `wavread` einfach zugegriffen werden kann.

D.2 Beschreibung der Funktionen

D.2.1 Test aller untersuchten Algorithmen

Mit der Funktion `sim_system.m` können alle Algorithmen, die in dieser Arbeit behandelt wurden, für beliebige Eingangssignale und Arraykonfigurationen getestet werden. Mit Hilfe des implementierten Master-Slave Systems (siehe Abschnitt 5.3.4) werden sämtliche Ausgangssignale berechnet, die zur anschließenden Durchführung der objektiven Messmethoden notwendig sind. Zur Ausführung werden die Unterfunktionen `mvdr.m` und `post_filter.m`, sowie die Funktion `welch_est.m`, benötigt.

```
function [y_sv,varargout] = ...
    sim_system(speech,noise,phi_d,beam_nr,mue,filter_nr,cn,aa,alpha,mics,N,L,fs,flow,fhigh)

% [y_sv,y_s,y_v,y_beam,y_bs,y_bv] = ...
%     sim_system(speech,noise,phi_d,beam_nr,mue,filter_nr,cn,aa,alpha,mics,N,L,fs,flow,fhigh)
%
% Complete Simulation System for Post-Filtering techniques with
% MVDR-Beamformers in a diffuse noise field (Master-Slave-Algorithm)
%
% y_sv          processed noisy input
% y_s           processed speech-only signal (optional)
% y_v           processed noise-only signal (optional)
% y_beam        beamformer output (optional)
% y_bs          speech-only beamformer output (optional)
% y_bv          noise-only signal beamformer output (optional)
%
% speech        clean input signal matrix
% noise         input noise matrix
%
%             hint: if you don't need the Slave-Algorithm please declare
%             speech = noisy_input and noise = 0
% phi_d         desired azimuth angle (to direction of arrival)
% beam_nr       you can choose between the following beamformers
%               'DSB' ... Delay&Sum-Beamformer
%               'SDB' ... Superdirective Beamformer
%               default beam_nr = 'SDB'
% mue           for the regularization of Coherence Matrix Gamma
%               (see thesis, section 3.4)
%               mue in dB (typ. between -10 and -40dB); mue = 0 ... uses zero mue;
%               default mue = -20
% filter_nr     name of the chosen filter
%               'ZEL88' ..... Zelinski Filter based on Welch-estimated spectral
%               density functions
%               'ZEL88p' ..... Zelinski Filter based on Welch-estimation plus
%               post-processing method (see Zelinski 1988)
%               'SIM92' ..... Simmer 1992
%               'APAB' ..... Adaptive Postfilter with Arbitrary Beamformer
%               (see book "Micorphone Arrays" by Brandstein)
%               'APES' ..... Adaptive Postfilter Extension for
%               Superdirective beamformers (see Bitzer et al. 1999)
%               'MCCC' ..... McCowan Filter (see McCowan 2003)
%               'GMCC' ..... McCowan Filter, correct solution for all angles (see
%               diploma thesis, section 4.3)
%               default filter_nr = 'ZEL88'
% cn            introduce Comfort Noise to reduce speech distortions
%               (Minimum-Filter, see thesis section 4.6.1)
%               0 ..... no Comfort Noise (default)
%               1 ..... activate Comfort Noise
% aa           use adaptive Welch-parameter
%               (see thesis, section 4.6.2)
%               0 ..... use fixed Welch-parameter (default)
%               1 ..... use adaptive Welch-parameter
% alpha        factor for Welch estimation; default alpha = 0.8
```

```

%          set this value also for an adaptive Welch-parameter
% mics      microphone position matrix; default load mics_8xh.mat
% N          FFT length; default N = 512
% L          decimation factor L = N/M; default L = 4
% fs         sampling frequency in Hz; default fs = 16000 (16kHz)
% flow       lowest frequency in Hz; default 200 Hz
% fhigh      highest frequency in Hz; default 6800 Hz
%
% used functions: mvdr.m, post_filter.m, welch_est.m
%
% Syntax:
% [y_sv] = sim_system(noisy_signal,0,...)
% [y_sv] = sim_system(speech,noise,...)
% [y_sv,y_beam] = sim_system(noisy_signal,0,...)
% [y_sv,y_s,y_v] = sim_system(speech,noise,...)
% [y_sv,y_s,y_v,y_beam] = sim_system(speech,noise,...)
% [y_sv,y_s,y_v,y_beam,y_bs,y_bv] = sim_system(speech,noise,...)

if nargin<15 fhigh = 6800; end
if nargin<14 flow = 200; end
if nargin<13 fs = 16000; end
if nargin<12 L = 4; end
if nargin<11 N = 512; end
if nargin<10 load mics_8xh.mat; end
if nargin<9 alpha = 0.8; end
if nargin<8 aa = 0; end
if nargin<7 cn = 0; end
if nargin<6 filter_nr = 'ZEL88'; end;
if nargin<5 mue = -20; end;
if nargin<4 beam_nr = 'SDB'; end
if nargin<3
    help sim_system
    return;
end

% Parameters
theta_d = 90;      % elvation angle to direction of arrival (fixed)
fact = 1.3;        % additional parameter to scale the output signals
[K,Dim] = size(mics);

[Nx_orig,K_signal] = size(speech);
if K_signal ~= K
    error('number of mics does not match number of speechsignals')
end

% *****
% Zero-padding to reach a signallength to be a multiple of L
n_orig = [1:Nx_orig];
n_zeros = ceil(Nx_orig/N)*N - Nx_orig;

if noise == 0
    solo = true;
    signal = speech;
else
    signal = speech + noise;
    solo = false;
    speech = [speech;zeros(n_zeros,K)];
    noise = [noise;zeros(n_zeros,K)];
end
signal = [signal;zeros(n_zeros,K)];
Nx = length(signal);

% *****
% Initialise Vectors and Matrices, and calculation different parameters

% init Hanning-Window for FFT
h = hanning(N);
H = h(:) * ones(1,K);

```

```

% Calculate decimationfactor and half of FFT-Length
M = N/L;
N2 = N/2 + 1;
n2 = 1:(N2);

% Calculate mue
if mue ~= 0
    mue = 10^(mue/10);
end

% Initialise output vectors
y_sv = zeros(Nx,1);
switch nargout
    case 1
    case 2
        if solo
            y_beam = zeros(Nx,1);
        else
            error('Noise input has to be Zero for this choice of output variables!!')
        end
    case 3
        if ~solo
            y_s = zeros(Nx,1);
            y_v = zeros(Nx,1);
        else
            error('Noise input has to be Zero for this choice of output variables!!')
        end
    case 4
        if ~solo
            y_s = zeros(Nx,1);
            y_v = zeros(Nx,1);
            y_beam = zeros(Nx,1);
        else
            error('Noise input has to be Zero for this choice of output variables!!')
        end
    case 6
        if ~solo
            y_s = zeros(Nx,1);
            y_v = zeros(Nx,1);
            y_beam = zeros(Nx,1);
            y_bs = zeros(Nx,1);
            y_bv = zeros(Nx,1);
        else
            error('Noise input has to be Zero for this choice of output variables!!')
        end
    otherwise
        error('This choice of output variables is not possible! Please see help sim_system')
end

% Calculate Lowpass and Highpass Filter
[bh,ah] = butter(4,flow/(fs/2),'high');
h_high = freqz(bh,ah,N2,fs);
[bl,al] = butter(4,fhigh/(fs/2),'low');
h_low = freqz(bl,al,N2,fs);

% Calculate number of all possible sensor combinations
comb = 2/(K*(K-1));

% Initialize vectors necessary to calculate the different postfilters
switch filter_nr
    case 'SIM92'
        cross_dens = zeros(N2,1/comb);
        Pyy = zeros(N2,1);
    case {'ZEL88','ZEL88p'}
        auto_dens = zeros(N2,K);
        cross_dens = zeros(N2,1/comb);
    case 'MCC03'

```

```

        auto_dens = zeros(N2,K);
        cross_dens = zeros(N2,1/comb);
    case 'GMCC'
        auto_dens = zeros(N2,K);
        cross_dens = zeros(N2,1/comb);
    case 'APAB'
        Pxx = zeros(N2,1);
        Pyy = zeros(N2,1);
    case 'APES'
        Pyy = zeros(N2,1);
        Pxx = zeros(N2,K);
        Pzz = zeros(N2,1);
end

% Check microphone position matrix
if (Dim < 2) | (K < 1)
    error('bad matrix of microphine coordinates');
end

if Dim == 2
    rn = [mics zeros(K,1)];
else
    rn = mics;
end

%Calc. of angles in rad
theta_d = theta_d(:).' * pi / 180;
phi_d = phi_d(:).' * pi / 180;

% Calculate time alignment vector
ed = [sin(theta_d).*cos(phi_d); sin(theta_d).*sin(phi_d); cos(theta_d)];
tau = rn*ed;

% Define Matrix of the micorphone distances
xc = rn(:,1);
xc = xc(:,ones(K,1));
dxc = xc - xc.';
yc = rn(:,2);
yc = yc(:,ones(K,1));
dyc = yc - yc.';
if Dim == 2
    dR = sqrt(dxc.^2 + dyc.^2);
else
    zc = rn(:,3);
    zc = zc(:,ones(K,1));
    dzc = zc - zc.';
    dR = sqrt(dxc.^2 + dyc.^2 + dzc.^2);
end

% Calculate Coherencematrix, MVDR-Beamformer coefficients and the steering
% vector
for l = 1:N2
    beta = 2*(l-1)/N*fs/340;
    % Coherence function for a diffuse noise field
    Gamma_dum = sinc(beta*dR);
    % use constrained design for a SDB
    if strcmp(beam_nr,'SDB')
        Gamma_const = tril(Gamma_dum,-1)./(1+mue) + diag(diag(Gamma_dum)) + ...
            triu(Gamma_dum,1)./(1+mue);
    else
        Gamma_const = Gamma_dum;
    end
    Gamma(:,l) = Gamma_const;

    % Calculate MVDR-Beamformer Coefficients
    [W(:,l),d0(:,l)] = mvdr(tau,Gamma(:,l),(l-1)/N*fs,beam_nr);
end

```

```

% Change Coherencematrix to a matrix of all sensor combinations
% needed for McCowan 2003 postfilter calculation
counter = 0;
for m = 1:(K-1)
    for n = (m+1):K
        counter = counter + 1;
        Gamma_comb(:,counter) = real(Gamma(m,n,:));
    end
end

% *****
% Main program, Master-Slave-Algorithm
for k = 1:M:(Nx-N+1)
    k1 = k:(k+N-1);

    % *****
    % proceeded Master-Algorithm and determination of Post-Filter bins

    % FFT - Filterbank with Hanning-Windowing
    X = fft(signal(k1,:) .* H,N).';

    % Multiply Beamformer weighting coefficients with signal
    X_mod = conj(W) .* X(:,n2);

    % Calculate sum of all signals (beamformer output)
    Y_sum = sum(X_mod).';

    % Calculating the post filter
    switch filter_nr
        case {'ZEL88','ZEL88p'}
            % Use time-aligned for calculating
            % post-filter
            X = (conj(d0).*X(:,n2)).';
            % Calculate Postfilter-Coefficients of the Zelinski Postfilter
            % with Welch-estimation / of the Zelinski Filter with
            % Welch-estiamtion and Zelinskis post-processing method
            [H_post,alpha,auto_dens,cross_dens] = ...
                post_filter(filter_nr,cn,aa,alpha,X,auto_dens,cross_dens);
        case 'SIM92'
            % Use time-aligned for calculating
            % post-filter
            X = (conj(d0).*X(:,n2)).';
            % Calculate Postfilter-Coefficients of the Simmer92 Postfilter
            % with Welch-estimation
            [H_post,alpha,cross_dens,Pyy] = ...
                post_filter(filter_nr,cn,aa,alpha,X,Y_sum,cross_dens,Pyy);
        case 'MCC03'
            % Use time-aligned for calculating
            % post-filter
            X = (conj(d0).*X(:,n2)).';
            % Calculate Postfilter-Coefficients of the McCowan03 Postfilter
            [H_post,alpha,auto_dens,cross_dens] = ...
                post_filter(filter_nr,cn,aa,alpha,X,Gamma_comb,auto_dens,cross_dens);
        case 'GMCC'
            % Use time-aligned for calculating
            % post-filter
            X = (conj(d0).*X(:,n2)).';
            % Calculate Postfilter-Coefficients of the McCowan03
            % Postfilter, correct solution
            [H_post,alpha,auto_dens,cross_dens] = ...
                post_filter(filter_nr,cn,aa,alpha,X,Gamma_comb,d0,auto_dens,cross_dens);
        case 'APAB'
            % Use time-aligned or beamformed signal for calculating
            % post-filter
            X = (conj(d0).*X(:,n2)).';
            % Calculate Postfilter-Coefficients of the APAB Postfilter
            [H_post,alpha,Pxx,Pyy] = post_filter(filter_nr,cn,aa,alpha,X,Y_sum,Pxx,Pyy);
        case 'APES'

```

```

    % Calculate output of a Delay&Sum Beamformer
    Y = conj(d0./K).*X(:,n2);
    Y = sum(Y).';
    X = (X(:,n2)).';
    % Calculate Postfilter-Coefficients of the APES Postfilter
    [H_post,alpha,Pxx,Py,Pzz] = post_filter(filter_nr,cn,aa,alpha,X,Y,Y_sum,Pxx,Py,Pzz);
end

% Calculate output of the postfilter
Y = Y_sum.*H_post;

% Highpass and Lowpass filter to cut frequencies
Y = h_high.*Y;
Y = h_low.*Y;

% IFFT and Overlap-Add (OLA)
Y = [Y;conj(Y(end-1:-1:2))];
yb = (ifft(Y,N));
y_sv(k1) = y_sv(k1) + yb(:);

% *****
% Slave-Algorithm for Speech- and Noise-only signals
% calculating signal at beamformer output, speech-only at beamformer output,
% noise-only at beamformer output, speech-only at postfilter output and
% noise-only at postfilter output, depending on input and output arguments
if nargout>2
    % FFT - Filterbank with Hanning-Windowing for speech- and noise-only
    S = fft(speech(k1,:)) .* H,N).';
    V = fft(noise(k1,:)) .* H,N).';

    % Multiply Beamformer weighting coefficients with signal
    S_mod = conj(W) .* S(:,n2);
    V_mod = conj(W) .* V(:,n2);

    % Calculate Beamformer outputs of speech- and noise-only
    S_sum = sum(S_mod).';
    V_sum = sum(V_mod).';

    switch nargout
        case 4
            % Calc. signal at beamformer output
            % Highpass and Lowpass filtering of beamformer output
            Y_beam = h_low.*h_high.*Y_sum;
            % IFFT and Overlap-Add (OLA)
            Y_beam = [Y_beam;conj(Y_beam(end-1:-1:2))];
            yb = (ifft(Y_beam,N));
            y_beam(k1) = y_beam(k1) + yb(:);
        case 6
            % Calc. signal at beamformer output
            % Highpass and Lowpass filtering of beamformer output
            Y_beam = h_low.*h_high.*Y_sum;
            % IFFT and Overlap-Add (OLA)
            Y_beam = [Y_beam;conj(Y_beam(end-1:-1:2))];
            yb = (ifft(Y_beam,N));
            y_beam(k1) = y_beam(k1) + yb(:);

            % Calc. speech-only signal at beamformer output
            % Highpass and Lowpass filtering of beamformer output
            Y_bs = h_low.*h_high.*S_sum;
            % IFFT and Overlap-Add (OLA)
            Y_bs = [Y_bs;conj(Y_bs(end-1:-1:2))];
            yb = (ifft(Y_bs,N));
            y_bs(k1) = y_bs(k1) + yb(:);

            % Calc. noise-only signal at beamformer output
            % Highpass and Lowpass filtering of beamformer output
            Y_bv = h_low.*h_high.*V_sum;
            % IFFT and Overlap-Add (OLA)

```

```

        Y_bv = [Y_bv;conj(Y_bv(end-1:-1:2))];
        yb = (ifft(Y_bv,N));
        y_bv(k1) = y_bv(k1) + yb(:);
    end

    % Speech-only signal
    % Calc. at output of the postfilter
    Y_s = S_sum.*(H_post);
    % Highpass and Lowpass filtering
    Y_s = h_low.*h_high.*Y_s;
    % IFFT and Overlap-Add (OLA)
    Y_s = [Y_s;conj(Y_s(end-1:-1:2))];
    yb = (ifft(Y_s,N));
    y_s(k1) = y_s(k1) + yb(:);

    % Noise-only signal
    % Calc. at output of the postfilter
    Y_v = V_sum.*(H_post);
    % Highpass and Lowpass filtering
    Y_v = h_low.*h_high.*Y_v;
    % IFFT and Overlap-Add (OLA)
    Y_v = [Y_v;conj(Y_v(end-1:-1:2))];
    yb = (ifft(Y_v,N));
    y_v(k1) = y_v(k1) + yb(:);

elseif nargout == 2 & solo
    % Calc. signal at beamformer output
    % Highpass and Lowpass filtering of beamformer output
    Y_beam = h_low.*h_high.*Y_sum;
    % IFFT and Overlap-Add (OLA)
    Y_beam = [Y_beam;conj(Y_beam(end-1:-1:2))];
    yb = (ifft(Y_beam,N));
    y_beam(k1) = y_beam(k1) + yb(:);
end
end

% Output signal of the whole postfilter algorithm
% Realpart and scaled
y_sv = real(y_sv(n_orig)*1/L*2*fact);

% Calc. realpart and scale Slave-output-signals
if nargout>2
    y_s = real(y_s*1/L*2*fact);
    y_v = real(y_v*1/L*2*fact);
    varargout{1} = y_s(n_orig);
    varargout{2} = y_v(n_orig);
    switch nargout
        case 4
            y_beam = real(y_beam*1/L*2*fact);
            varargout{3} = y_beam(n_orig);
        case 6
            y_beam = real(y_beam*1/L*2*fact);
            y_bs = real(y_bs*1/L*2*fact);
            y_bv = real(y_bv*1/L*2*fact);
            varargout{3} = y_beam(n_orig);
            varargout{4} = y_bs(n_orig);
            varargout{5} = y_bv(n_orig);
        end
    elseif nargout == 2 & solo
        y_beam = real(y_beam*1/L*2*fact);
        varargout{1} = y_beam(n_orig);
    end
end

```

Berechnung der Beamformerkoeffizienten und des Steuervektors

Das Unterprogramm `mvdr.m` berechnet den Steuervektor sowie die Koeffizienten des gewählten Beamformers.

```
function [w,d0] = mvdr(tau,Gamma,f,nr)

% [w,d0] = mvdr(tau,Gamma,f,nr)
%
% Compute weights of MVDR-Beamformer
%
% w          Weight vector of length K (K ... number of mics)
% d0         Steering vector
%
% tau        Time alignment vector
% Gamma       Coherencematrix
% f          Frequency in Hz
% nr         you can choose between the following beamformers
%            'DSB' ... Delay&Sum-Beamformer
%            'SDB' ... Superdirective Beamformer

if nargin <4
    help mvdr;
    return;
end

[K,dum] = size(Gamma);

beta = 2*pi*f/340;          % wave number

% calc. steering vector of desired direction
d0 = exp(-j * beta * tau);

switch nr
    case 'DSB'
        % Delay-Sum-Beamformer
        w = d0 / K;
    case 'SDB'
        % Superdirective Beamformer for Diffuse Noise Field
        B = (Gamma^-1)*d0;
        Lambda = (d0'*B)\1;      % Lagrange multiplicator
        w = B*Lambda;            % optimum coefficient vector at given frequency
    otherwise
        error('Please insert correct number for the beamformer you want to choose!')
end
```

Berechnung der Postfilterkoeffizienten

Das Unterprogramm `post_filter.m` enthält alle Postfilteralgorithmen und berechnet adaptiv die aktuellen Gewichtungskoeffizienten.

```
function [H_post,alpha,varargout] = post_filter(filter_nr,cn,aa,alpha,X,varargin)

% [H_post, alpha, new return vectors] = ...
%   post_filter(filter_nr,cn,aa,alpha,X, input vectors, old return vectors)
%
% This function calculates the bins of the Post-Filter depending on
% which Filter-Realization is chosen.
%
% H_post          Post-Filter bins
% alpha           new factor for Welch-estimation; if adaptive
%                 Welch-factor is used
% new return vectors  new calculated auto- and cross spectral density
%                 vectors or matrix for the current frame
```

```

%
% filter_nr          name of the chosen filter
%                    'ZEL88' ..... Zelinski Filter based on Welch-estimated spectral
%                               density functions
%                    'ZEL88p' ..... Zelinski Filter based on Welch-estimation plus
%                               post-processing method (see Zelinski 1988)
%                    'SIM92' ..... Simmer 1992
%                    'APAB' ..... Adaptive Postfilter with Arbitrary Beamformer
%                               (see book "Micorphone Arrays" by Brandstein)
%                    'APES' ..... Adaptive Postfilter Extension for
%                               Superdirective beamformers (see Bitzer et al. 1999)
%                    'MCCC' ..... McCowan Filter (see McCowan 2003)
%                    'GMCC' ..... McCowan Filter, correct solution for all angles (see
%                               diploma thesis)
% cn                 introduce Comfort Noise to reduce speech distortions
%                    (Minimum-Filter, see thesis section 4.6.1)
%                    0 ..... no Comfort Noise (default)
%                    1 ..... activate Comfort Noise
% aa                 use adaptive Welch-parameter
%                    (see thesis, section 4.6.2)
%                    0 ..... use fixed Welch-parameter (default)
%                    1 ..... use adaptive Welch-parameter
% alpha              previous factor for Welch-estimation
% X                  time-aligned input signal bins
% input vectors      other input vectors, depends on Post-Filter
% old return vectors auto- and cross spectral density vectors or matrix
%                    of the previous frame
% hint:
% Please declare and initialize the return vectors in the master program

if nargin<2
    help post_filter
    return;
end

[N2,K] = size(X);
comb = 2/(K*(K-1));      % all possible sensor combinations

% Calc. minimum Postfilter function; to avoid zeros in the transfer
% function
f1 = ceil(512/16000*200);
f2 = ceil(512/16000*4500);
f = [f1:1:f2].';
H_min_func = [zeros(f1-1,1);0.7/(f2-f1).*(f-f1); 0.7.*ones(N2-f2,1)]; % see diploma thesis Boigner

% *****
% Parameter, Minimum Postfilter function if choosen
% introduces Comfort Noise, but avoids speech degradation
if cn == 0
    H_min = 0.05;
else
    H_min = H_min_func;
end

%*****
% Calculate Postfilters
switch filter_nr
    case 'SIM92'
        %*****
        % Estimation of postfiler transfer function =>
        % Simmer with Welch-estimation
        Y = varargin{1};      % signal at beamformer output
        cross_dens = varargin{2}; % Cross-PSD of time-aligned input signals
        Pyy = varargin{3};    % Auto-PSD of beamformer output

        % Calc. Auto-PSD of beamformer output using Welch-formula
        Pyy = welch_est(Pyy,Y,Y,alpha);
        % Calc. Cross-PSD of time-aligned input signals using Welch-formula

```

```

counter = 0;
for m = 1:(K-1)
    for n = (m+1):K
        counter = counter + 1;
        cross_dens(:,counter) = welch_est(cross_dens(:,counter),X(:,m),X(:,n),alpha);
    end
end
% Calc. averaged Cross-PSD over all sensor comb.
C = sum(real(cross_dens.')).';
C = comb.*C;
% Calc. Postfilter
H_post = C./Pyy;

if nargout == 4
    varargout{1} = cross_dens;
    varargout{2} = Pyy;
else
    error('Wrong number of output arguments!!')
end
case 'ZEL88'
%*****
% Estimation of postfilter transfer function =>
% Zelinski with Welch-estimation
auto_dens = varargin{1};      % Auto-PSD of time-aligned input signals
cross_dens = varargin{2};     % Cross-PSD of time-aligned input signals

% Calc. Auto-PSD of time-aligned input signals using Welch-formula
for m = 1:K
    auto_dens(:,m) = welch_est(auto_dens(:,m),X(:,m),X(:,m),alpha);
end
% Calc. Cross-PSD of time-aligned input signals using Welch-formula
counter = 0;
for m = 1:(K-1)
    for n = (m+1):K
        counter = counter + 1;
        cross_dens(:,counter) = welch_est(cross_dens(:,counter),X(:,m),X(:,n),alpha);
    end
end
% Calc. averaged Cross-PSD over all sensor comb.
C = sum(real(cross_dens.')).';
C = comb.*C;
% Calc. averaged Auto-PSD over all sensor comb.
A = sum(auto_dens.')).';
A = A./K;
% Calc. Postfilter
H_post = C./A;

if nargout == 4
    varargout{1} = auto_dens;
    varargout{2} = cross_dens;
else
    error('Wrong number of output arguments!!')
end
case 'ZEL88p'
%*****
% Estimation of postfilter transfer function =>
% Zelinski with Welch-estimation incl. post-processing method
auto_dens = varargin{1};      % Auto-PSD of time-aligned input signals
cross_dens = varargin{2};     % Cross-PSD of time-aligned input signals

% Calc. Auto-PSD of time-aligned input signals using Welch-formula
for m = 1:K
    auto_dens(:,m) = welch_est(auto_dens(:,m),X(:,m),X(:,m),alpha);
end
% Calc. Cross-PSD of time-aligned input signals using Welch-formula
counter = 0;
for m = 1:(K-1)
    for n = (m+1):K

```

```

        counter = counter + 1;
        cross_dens(:,counter) = welch_est(cross_dens(:,counter),X(:,m),X(:,n),alpha);
    end
end
% Calc. averaged Cross-PSD over all sensor comb.
C_tilde = sum(real(cross_dens.')).';
C_tilde = comb.*C_tilde;

% post-processing method (see Zelinski 1988)
S_2 = max(C_tilde,0);
S_2 = S_2.^2;
S_2 = smooth(S_2);
V = real(cross_dens.').';
V = min(V,0);
for m = 1:N2
    M(m) = nnz(V(:,m));
end
M = max(M,1);
V = sum(V.^2);
V = (V./M).';
V = smooth(V);
alpha_k = S_2./(S_2 + V.*comb);
C = alpha_k.*C_tilde;

% Calc. averaged Auto-PSD over all sensor comb.
A = sum(auto_dens.')).';
A = A./K;
% Calc. Postfilter
H_post = C./A;

if nargout == 4
    varargout{1} = auto_dens;
    varargout{2} = cross_dens;
else
    error('Wrong number of output arguments!!')
end
case 'APAB'
%*****
% Estimation of postfilter transfer function =>
% APAB Postfilter
Y = varargin{1};           % signal at beamformer output
Pxx = varargin{2};         % Auto-PSD of time-aligned and averaged input signals
Pyy = varargin{3};         % Auto-PSD of beamformer output

% average time-aligned input signals
X = sum(X.').';
X = X./K;
% Calc. Auto-PSD of time-aligned and averaged input signals using
% Welch-formula
Pxx = welch_est(Pxx,X,X,alpha);
% Calc. Auto-PSD of beamformer output using Welch-formula
Pyy = welch_est(Pyy,Y,Y,alpha);
% Calc. Postfilter
H_post = Pyy./Pxx;

if nargout == 4
    varargout{1} = Pxx;
    varargout{2} = Pyy;
else
    error('Wrong number of output arguments!!')
end
case 'APES'
%*****
% Estimation of postfilter transfer function =>
% APES Postfilter
Y = varargin{1};           % signal at DSB output
Z = varargin{2};           % signal at SDB output
Pxx = varargin{3};         % Auto-PSD of time-aligned and averaged input signals

```

```

Pyy = varargin{4};          % Auto-PSD of DSB output
Pzz = varargin{5};          % Auto-PSD of SDB output

% Calc. Auto-PSD of DSB output using Welch-formula
Pyy = welch_est(Pyy,Y,Y,alpha);
% Calc. Auto-PSD of SDB output using Welch-formula
Pzz = welch_est(Pzz,Z,Z,alpha);
% Calc. Auto-PSD of time-aligned input signals using Welch-formula
for m = 1:K
    Pxx(:,m) = welch_est(Pxx(:,m),X(:,m),X(:,m),alpha);
end
% Calc. nominator of APES Algorithm
dum = sum(Pxx.').';
dum = (1/K^2).*dum;
nom = Pyy-dum;
% average nominator over all sensor comb.
nom = (K/(K-1)).*nom;
% Calc. First postfilter
H1 = nom./Pyy;
% Calc. Second postfilter
H2 = Pzz./Pyy;
% Calc. Postfilter
H_post = H1.*H2;

if nargin == 5
    varargout{1} = Pxx;
    varargout{2} = Pyy;
    varargout{3} = Pzz;
else
    error('Wrong number of output arguments!!')
end
case 'MCC03'
    %*****
    % Estimation of the postfilter transfer function =>
    % McCowan 2003
    Gamma_comb = varargin{1};    % Coherence matrix over all sensor comb.
    auto_dens = varargin{2};      % Auto-PSD of time-aligned input signals
    cross_dens = varargin{3};     % Cross-PSD of time-aligned input signals

    % Calc. Auto-PSD of time-aligned input signals using Welch-formula
    for m = 1:K
        auto_dens(:,m) = welch_est(auto_dens(:,m),X(:,m),X(:,m),alpha);
    end

    % Calc. estimation of PSD of the speech signal (see McCowan 2003)
    counter = 0;
    for m = 1:(K-1)
        for n = (m+1):K
            counter = counter + 1;
            % Calc. Cross-PSD of time-aligned input signals using
            % Welch-formula
            cross_dens(:,counter) = welch_est(cross_dens(:,counter),X(:,m),X(:,n),alpha);
            delta1(:,counter) = real(cross_dens(:,counter));    % realpart of PSD
            delta2(:,counter) = auto_dens(:,m) + auto_dens(:,n); % average of Auto-PSDs m and n
        end
    end
    dum = 0.5.*Gamma_comb;
    delta2 = dum.*delta2;
    dum = delta1 - delta2;
    nom = 1 - Gamma_comb;    % calc. denominator of speech PSD
    C_ss = dum./nom;         % estimated PSD of speech signal
    % Calc. averaged speech-PSD over all sensor comb.
    C = sum(C_ss.').';
    C = comb.*C;
    % Calc. averaged Auto-PSD over all sensor comb.
    A = sum(auto_dens.').';
    A = A./K;
    % Calc. Postfilter

```

```

H_post = C./A;

if nargout == 4
    varargout{1} = auto_dens;
    varargout{2} = cross_dens;
else
    error('Wrong number of output arguments!!')
end
case 'GMCC'
    %*****
    % Estimation of the postfilter transfer function =>
    % McCowan03, correct solution, calc. in my diploma thesis
    Gamma_comb = varargin{1}; % Coherence matrix over all sensor comb.
    d0 = varargin{2}.'; % steering vector
    auto_dens = varargin{3}; % Auto-PSD of time-aligned input signals
    cross_dens = varargin{4}; % Cross-PSD of time-aligned input signals

    % Calc. Auto-PSD of time-aligned input signals using Welch-formula
    for m = 1:K
        auto_dens(:,m) = welch_est(auto_dens(:,m),X(:,m),X(:,m),alpha);
    end
    % Calc. estimation of PSD of the speech signal (see McCowan 2003)
    counter = 0;
    for m = 1:(K-1)
        for n = (m+1):K
            counter = counter + 1;
            % Calc. new Gama_comb with steering vector
            Gamma_comb(:,counter) = real(conj(d0(:,m)).*d0(:,n)).*Gamma_comb(:,counter);
            % Calc. Cross-PSD of time-aligned input signals using
            % Welch-formula
            cross_dens(:,counter) = welch_est(cross_dens(:,counter),X(:,m),X(:,n),alpha);
            delta1(:,counter) = real(cross_dens(:,counter)); % realpart of PSD
            delta2(:,counter) = auto_dens(:,m) + auto_dens(:,n); % average of Auto-PSDs m and n
        end
    end
    dum = 0.5.*Gamma_comb;
    delta2 = dum.*delta2;
    dum = delta1 - delta2;
    nom = 1 - Gamma_comb; % calc. denominator of speech PSD
    C_ss = dum./nom; % estimated PSD of speech signal
    % Calc. averaged speech-PSD over all sensor comb.
    C = sum(C_ss.').';
    C = comb.*C;
    % Calc. averaged Auto-PSD over all sensor comb.
    A = sum(auto_dens.').';
    A = A./K;
    % Calc. Postfilter
    H_post = C./A;

    if nargout == 4
        varargout{1} = auto_dens;
        varargout{2} = cross_dens;
    else
        error('Wrong number of output arguments!!')
    end
otherwise
    error('Wrong Filter-Name!!!')
end

% restrict Postfilter
H_post = max(H_post,H_min);
H_post = min(H_post,1);

% Calc. current frequency-dependent alpha
if aa == 1
    alpha_fact = 0.3;
    alpha = 0.98 - alpha_fact.*H_post;
end

```

Schätzung der Leistungsdichtespektren

Die Funktion `welch_est.m` berechnet frameweise die Leistungsdichtespektren nach Welch. Gemäß Gl. 2.26 bzw. Gl. 2.27 ist das LDS des vorherigen Frames zur Berechnung des LDS des aktuellen Frames notwendig.

```
function [lds] = Welch_est(lds_prev,Xi,Xj,alpha)

% [korr] = Welch_est(korr_prev,Xi,Xj,fs,D,tau)
%
% Calculates the Welch-Estimate of Auto- or Cross-Spectral-Density
%
% lds          Auto- or Cross-Spectral Density for the current frame
%
% lds_prev     Auto- or Cross-Spectral Density for the previous frame
% Xi           signal vector of signal i
% Yi           signal vector of signal j;
%              default Xj = Xi (for Auto-Spectral density)
% alpha        Welch-factor for weighting the previous frame; default alpha = 0.8

if nargin<4 alpha = 0.8; end
if nargin<3 Xj = Xi; end
if nargin<2
    help Welch_est;
    return;
end

% Calc. recursive Welch-formula
first = alpha.*lds_prev;
delta = Xi.*conj(Xj);
delta = (1 - alpha).*delta;
% calc. the spectral density
lds = first + delta;
```

D.2.2 Objektive Messungen

Mit der Funktion `eval_unit.m` können alle objektiven Messungen (SSNRE, LAR und SD) durchgeführt werden. Zuvor müssen jedoch alle dafür notwendigen Signalvektoren mit Hilfe der Funktion `sim_system.m` ermittelt werden.

```
function [varargout] = eval_unit(speech_in, noise_in,N,varargin)

% [ssnre,lar,sd] = eval_unit(speech_in, noise_in,N,refsig,speech_out,noise_out,signal_out,porder)
%
% The Evaluation Unit calculates all possible objective measures
% (see diploma thesis, section 5.3)
%
% ssnre        SSNR Enhancement SSNRE = SSNR_out - SSNR_in
% lar          LAR (log-area-ratio distance) (optional)
% sd           SD (speech degradation) (optional)
%
% speech_in    input speech only (vector!)
% noise_in     input noise only (vector!)
% N            Block length; default N = 256
% refsig       reference signal to calc. LAR and SD (vector!)
% speech_out   output speech only (vector!)
% noise_out    output noise only (vector!)
% signal_out   output signal (vector!)
% porder       model order for Levinson-Durbin Recursion (see thesis, appendix B.2);
%              typ. porder = 14
%
% hint: to provide an accurate objective measures use signals that hardly contain
```

```

%           intervals of silence in the speech utterances
%
% used functions: calc_parcor.m
%
% Syntax:
% to calculate SSNR of a signal
% [ssnr] = eval_unit(speech_in,noise_in,N)
%
% to calc. SSNRE, LAR & SD between the input signal and the processed signal
% [ssnre,lar,sd] = eval_unit(speech_in,noise_in,N,refsig,speech_out,noise_out,signal_out,porder)

if nargin<3 N=256; end
if nargin<2
    help eval_unit
    return;
end

% Declare input vectors
switch nargin
    case 3
    case 8
        refsig=varargin{1};
        speech_out=varargin{2};
        noise_out=varargin{3};
        signal_out=varargin{4};
        porder=varargin{5};
        refsig=refsig(:);speech_out=speech_out(:); % columnvectors!
        noise_out=noise_out(:); signal_out=signal_out(:); % columnvectors!
    otherwise
        error('Wrong number of input arguments')
end

speech_in=speech_in(:); noise_in=noise_in(:); % columnvectors!

% Calc. the vector length as a multiple of the block length
Nx_orig=length(speech_in);
M=floor(Nx_orig/N);
Nx=M*N;

% Calc. the objective measures for each frame
l=1;
for i=1:N:(Nx-N)
    k1=i:(i+N-1);

    % Calculate input-SSNR for each frame
    ssnr=10*log10(sum(speech_in(k1).^2)./sum(noise_in(k1).^2));

    ssnr_in(l)=ssnr;
    if nargin==8
        % Calculate output-SSNR for each frame
        ssnr_out(l)=10*log10(sum(speech_out(k1).^2)./sum(noise_out(k1).^2));

        % Calculate PARCOR coefficients and Area coefficients
        p_refsig(:,l)=calc_parcor(refsig(k1),porder);
        p_signal_out(:,l)=calc_parcor(signal_out(k1),porder);
        p_speech_out(:,l)=calc_parcor(speech_out(k1),porder);

        g_refsig(:,l)=(1+p_refsig(:,l))./(1-p_refsig(:,l));
        g_signal_out(:,l)=(1+p_signal_out(:,l))./(1-p_signal_out(:,l));
        g_speech_out(:,l)=(1+p_speech_out(:,l))./(1-p_speech_out(:,l));

        % Calculate LAR and SD for each frame
        lar(l)=sqrt((1/porder)*sum((abs(20*log10(g_refsig(:,l)./g_signal_out(:,l))))).^2));
        sd(l)=sqrt((1/porder)*sum((abs(20*log10(g_refsig(:,l)./g_speech_out(:,l))))).^2));
    end
    l=l+1;
end
end

```

```

switch nargin
    case 3
        % Calc. the averaged SSNR of a signal over all frames
        varargout{1}=sum(ssnr_in)/length(ssnr_in);
    case 8
        % Calc. the averaged input and output SSNR over all frames
        msnr_out=sum(ssnr_out)/length(ssnr_out);
        msnr_in=sum(ssnr_in)/length(ssnr_in);
        % Calc. the averaged SSNRE over all frames
        varargout{1}=msnr_out-msnr_in;

        % Calc. the averaged LAR & SD over 95% of all frames
        lar_5=sort(lar);
        mlar_5=lar_5(1:ceil(length(lar_5)*0.95));
        sd_5=sort(sd);
        msd_5=sd_5(1:ceil(length(sd_5)*0.95));
        varargout{2}=sum(mlar_5)/length(mlar_5);
        varargout{3}=sum(msd_5)/length(msd_5);
    otherwise
        error('Wrong number of input arguments!!')
end

```

Berechnung der PARCOR Koeffizienten

Mit Hilfe der Funktion `calc_parcor.m` können die PARCOR Koeffizienten ermittelt werden. Dies geschieht anhand der Levinson-Durbin Rekursion (siehe Anhang B.2).

```

function k = calc_parcor(x,P)

% k = calc_parcor(x,P)
%
% Calculate PARCOR coefficients
% (see diploma thesis, Appendix B.2)
%
% k    vector with PARCOR coefficients for block x
%
% x    block of input signal vector x
% P    model order

% Calc. autocorrelation functions
N = length(x);
for i = 1:(P+1)
    sum_x = 0;
    for n = i:(N)
        sum_x = sum_x + x(n)*x(n-i+1);
    end
    r(i) = sum_x;
end
r = r(:);

% Initial conditions
e(1) = r(1);
a_new = 0;

% compute PARCOR coefficients and LP-parameters
for m = 1:P
    sum_a = 0;
    a = a_new;
    for i = 1:(m-1)
        sum_a = sum_a + a(i)*r(m+1-i);
    end
    k(m) = -(sum_a + r(m+1)) / e(m); % PARCOR coefficients
    for i = 1:(m-1)
        a_new(i) = a(i) + k(m)*a(m-i);
    end
end

```

```

    a_new(m) = k(m);
    e(m+1) = (1 - k(m)^2) * e(m);
end
k = k(:);

```

D.2.3 Richtcharakteristik

Um die Richtcharakteristik eines Beamformers (DSB und SDB) für verschiedene Rauschfelder zeichnen zu lassen, wird die Funktion `beampattern.m` angewandt.

```

function beampattern(beam_nr,phi_d,mue,mics,fs,varargin)

% beampattern(beam_nr,phi_d,mue,mics,fs,sim,phi_n)
%
% Plots the Beampattern of a 1-dimensional Array and the frequency response
% at a defined angle, as well as the frequency response for small
% perturbations imposed to mic positions
%
% beam_nr          you can choose between the following beamformers
%                  'DSB' ... Delay&Sum-Beamformer
%                  'SDB' ... Superdirective Beamformer
%                  default beam_nr = 'SDB';
% phi_d            angle to TDOA
% mue              for the regularization of Coherence Matrix Gamma
%                  mue in dB; default mue = -20
% mics              microphone positons; default load mics_8xh.mat
% fs               sampling frequency in Hz; default fs = 16000 (16kHz)
% sim              use simulated coherence function or sinc-function for
%                  Gamma; default sim = 'sinc'
%                  'incoh' ..... incoherent Noiseifield
%                  'sinc' ..... Sinc-Function
%                  'bessel' ..... Bessel-Function
%                  'zero' ..... puts a Zero at the angle
%                               specified by phi_n
% phi_n            angle for the plotted frequency response; if sim = 'zero'
%                  phi_n is the angle of the specified zero
%
% used functions: mvdr.m

if nargin>=7 phi_n = varargin{2}; end
if nargin>=6 sim = varargin{1}; end
if nargin<6 sim = 'sinc'; end
if nargin<5 fs = 16000; end
if nargin<4 load mics_8xh.mat; end
if nargin<3
    help beampattern
    return;
end

% Parameters
theta_d = 90; % elvation angle to direction of arrival (fixed)

% Check microphone position matrix
[K,Dim] = size(mics);
if (Dim < 2) | (K < 1)
    error('bad matrix of microphine coordinates');
end

% Calculate mue
if mue ~= 0
    mue = 10^(mue/10);
end

%Calc. of angles in rad
theta_r = theta_d(:).' * pi / 180;

```

```

phi_r = phi_d(:).' * pi / 180;
phi_n = phi_n(:).' * pi / 180;
% Calculate time alignment vector
ed = [sin(theta_r).*cos(phi_r); sin(theta_r).*sin(phi_r)];
Rc = mics*ed;
% Define Matrix of the microphone distances
xc = mics(:,1);
xc = xc(:,ones(K,1));
dR = (xc-xc.>');

% Define Frequency vector
if fs <= 12000
    f = linspace(0,3400,120);
else
    f = linspace(0,6800,240);
end
Nf = length(f);

% Calculate Coherencefunction
for l = 1:Nf
    switch sim
        case 'incoh'
            % Calc. coherencefunction for an incoherent noise field
            Gamma_const = diag(ones(1,K));
        case 'bessel'
            % Calc. coherencefunction for a cylindrical isotropic noise field
            beta = 2*pi*f(l)/340;
            Gamma_dum = besselj(0,beta*dR);
        case 'sinc'
            % Calc. coherencefunction for a diffuse noise field
            beta = 2*f(l)/340;
            Gamma_dum = sinc(beta*dR);
        case 'zero'
            % Calc. coherencefunction for a coherent noise field (interferer noise from angle phi_n)
            beta = 2*pi*f(l)/340;
            Gamma_real = cos(beta*cos(phi_n)*dR);
            Gamma_imag = -sin(beta*cos(phi_n)*dR);
            Gamma_dum = (Gamma_real + j*Gamma_imag);
    end
    if strcmp(sim,'zero') | strcmp(sim,'bessel') | strcmp(sim,'sinc')
        % regularization of the coherence matrix
        Gamma_const = tril(Gamma_dum,-1)./(1 + mue) + diag(diag(Gamma_dum)) + ...
            triu(Gamma_dum,1)./(1 + mue);
    end
    Gamma(:,l) = Gamma_const;
end

% Calculate Beampattern
phi_wav_d = [(-180):1:(180)]; % angle vector
phi_wav = phi_wav_d(:).' * pi / 180;
ed_wav = [sin(theta_r).*cos(phi_wav); sin(theta_r).*sin(phi_wav)];
Rc_wav = mics*ed_wav;
for l = 1:Nf
    % Calc. beamformer coefficients and steering vector for each angle and frequency
    [W(:,l),dum] = mvdr(Rc,Gamma(:,l),f(l),beam_nr);
    beta_wav = 2*pi*f(l)/340;
    d = exp(-j * beta_wav * Rc_wav);
    % Calc. gain for each angle and frequency
    H = abs(W(:,l)'.*d).^2;
    HdB = max(-25,10*log10(H + eps));
    H_log(:,l) = HdB;
end

% Plot beampattern
figure,surf(f,phi_wav_d,H_log);
axis tight
set(gca,'TickDir','out');

```

```

set(gca,'YTick',[-180:45:180]);
if fs <= 12000
    set(gca,'XTick',[100;1000;2000;3000;3400])
    set(gca,'XTickLabel',{'100','1000','2000','3000','3400'})
else
    set(gca,'XTick',[100;1000;2000;3000;4000;5000;6000;6800])
    set(gca,'XTickLabel',{'100','1000','2000','3000','4000','5000','6000','6800'})
end
colorbar('YLim',[-25 max(max(H_log))])
view([0,90]);
box on
shading interp
ylabel('\theta in Grad');
xlabel('Frequenz [Hz]');

% Calculate frequency response
if strcmp(sim,'zero') | ~isstr(sim)
    phi_freq = phi_n;
else
    phi_freq = phi_r;
end
figurebackcolor = 'black';
Hf = calc_freq_resp(mics,W,theta_r,phi_n,f,fs);
HfdB = max(-100,10*log10(Hf + eps));

% repeat for small perturbations imposed to mic positions
delta = 0.001; % standard deviation in m
r = mics + delta*randn(size(mics));
Hr = calc_freq_resp(r,W,theta_r,phi_n,f,fs);
HrdB = max(-100,10*log10(Hr + eps));

% Plot frequency response
pos = [0.045 0.01 0.4 0.37];
fp4 = figure('numbertitle','off','name','Frequency domain',...
    'Units','normal','Position',pos);
colordef(fp4,figurebackcolor);

plot(f,HfdB,f,HrdB);
grid on;
xlabel('f in Hz');
ylabel('magnitude in dB');
title('frequency responses of ideal (y), and perturbed array (m)');

% -----
% Calc. frequency response
function Hf = calc_freq_resp(mics,W,theta_r,phi_r,f,fs)

beta = (2*pi*f/340);

ed = [sin(theta_r).*cos(phi_r); sin(theta_r).*sin(phi_r)];
% Calc. of Constraint Matrix
d = exp(-j*(mics*ed)*beta);
Hf = abs(diag(W'*d)).^2;

```

D.2.4 Kohärenzfunktion

Die Kohärenzfunktion zweier Kanäle eines mehrkanaligen Rauschsignals kann mit der Funktion `coh_measure.m` angezeigt werden. Dafür werden die Leistungsdichtespektren des Rauschsignals mit Hilfe der Funktion `welch_est.m` geschätzt. Die LDS des letzten Frames der beiden betrachteten Rauschsignale werden zur Ermittlung der Kohärenzfunktion herangezogen.

```

function [varargout] = coh_measure(noise,ch1,ch2,alpha,mics,tit,N,L,fs)

% Measurement of the coherence function between two channels of a recorded multi-channel noise
% signal and comparison with the ideal sin(x)/x - coherence function
%
% [coh,coh_smooth] = coh_measure(noise,ch1,ch2,alpha,mics,tit,N,L,fs)
%
% coh                deliveres the measured coherence function
% coh_smooth         delivers a smoothed version of the meas. coherence
%                    function
%
% noise              input noise matrix recorded in a room
% ch1                first channel
% ch2                second channel
% alpha              factor for the exponentially weighted
%                    Welch periodogram; default = 0.8
% mics               microphone position matrix; default mics_8xh.mat
% tit                title of the plot; default tit = ''
% N                  FFT length; default = 512
% L                  decimation factor; default = 4
% fs                 sampling frequency; default = 16000 (16 kHz)
%
% Syntax:
% Plot measured coherence function:
% coh_measure(noise,ch1,ch2,alpha,mics,tit,N,L,fs)
% Save the measured coherence function and the smoothed c.f. in vectors:
% [coh,coh_smooth] = coh_measure(noise,ch1,ch2,alpha,mics,tit,N,L,fs)
%
% functions required: calc_cross.m

if nargin<9 fs = 16000; end
if nargin<8 L = 4; end
if nargin<7 N = 512; end
if nargin<6 tit = ''; end
if nargin<5 load mics_8xh.mat; end
if nargin<4 alpha = 0.8; end
if nargin<3
    help coh_measure
    return;
end

[K,Dim] = size(mics);
% Check microphone position matrix
if (Dim < 2) | (K < 1)
    error('bad matrix of microphine coordinates');
end

if Dim == 2
    rn = [mics zeros(K,1)];
else
    rn = mics;
end

% Define Distance Matrix of the Array
xc = rn(:,1);
xc = xc(:,ones(K,1));
dxc = xc - xc.';
yc = rn(:,2);
yc = yc(:,ones(K,1));
dyc = yc - yc.';
if Dim == 2
    dR = sqrt(dxc.^2 + dyc.^2);
else
    zc = rn(:,3);
    zc = zc(:,ones(K,1));
    dzc = zc - zc.';
    dR = sqrt(dxc.^2 + dyc.^2 + dzc.^2);
end

```

```

% set paramters
N2 = N/2 + 1;
n2 = 1:N2;

% Calc. spectral cross- and auto power density vectors
[cpsd,psd_x,psd_y] = calc_cross(noise,ch1,ch2,alpha,N,L,fs);

% Calc. coherence function
nom = sqrt(psd_x.*psd_y);
coh = cpsd./nom;

% Calc. microphone distance
d = dR(ch1,ch2);
if nargin == 0
    % Plot coherence functions
    coh_est = sinc((2*fs/N*d/340).*(n2-1))./(1 + 0.01);
    h = linspace(0,fs/2,N2);
    figure,plot(h,real(coh),'--b')
    hold on
    plot(h,coh_est,'r')
    hold off
    legend('Messung',['Theorie'],3)
    xlabel('Frequenz [Hz]')
    ylabel(['Real(\Gamma)'])
else
    % Save coherence functions in vectors
    varargout{1} = coh;
    varargout{2} = smooth(coh,30,'rloess');
end

% *****
function [CXX,CX1,CX2] = calc_cross(signal,ch1,ch2,alpha,N,L,fs)

% [CXX,CX1,CX2] = calc_cross(signal,ch1,ch2,alpha,N,L,fs)
%
% Calculate spectral cross- and auto power density vectors for 2 channels of a
% recorded multi-channel signal
%
% CXX          cross power density vector
% CX1          auto power density vector of channel 1
% CX2          cross power density vector of channel 2
%
% signal       input signal matrix recorded in a room
% ch1          channel 1
% ch2          channel 2
% alpha        factor for the exponentially weighted
%              Welch periodogram; default = 0.8
% N            FFT length; default = 512
% L            decimation factor; default = 4
% fs           sampling frequency; default = 16000 (16 kHz)
%
% used function: welch_est.m

if nargin<7 fs = 16000; end
if nargin<6 L = 4; end
if nargin<5 N = 512; end
if nargin<4 alpha = 0.8; end
if nargin<3
    help calc_cross
    return;
end

M = N/L;

% Zero-padding to reach a signallength to be a multiple of L
[Nx,K] = size(signal);
dum = ceil(Nx/N)*N - Nx;

```

```
signal = [signal;zeros(dum,K)];
Nx = length(signal);

% Initialise Vectors and Matrices
h = hanning(N);
H = h(:) * ones(1,K);
N2 = N/2 + 1;
y = zeros(Nx,K);
n2 = 1:(N2);

% initialise power density vectors
CXX = zeros(N2,1);
CX1 = zeros(N2,1);
CX2 = zeros(N2,1);

for k = 1:M:(Nx - 2*N + 1)
    k1 = k:k+N-1;

    % FFT - Filterbank with Hanning-Windowing
    X = fft(signal(k1,:)) .* H,N)';

    % Calc. spectral power density vectors
    Y = (X(:,n2))';
    CXX = welch_est(CXX,Y(:,ch1),Y(:,ch2),alpha);
    CX1 = welch_est(CX1,Y(:,ch1),Y(:,ch1),alpha);
    CX2 = welch_est(CX2,Y(:,ch2),Y(:,ch2),alpha);
end
```


Literaturverzeichnis

- [1] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.
- [2] M. Brandstein and D. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*. Springer-Verlag, 2001.
- [3] M. Boigner, "Mehrkanalalgorithmen zur Geräuschreduktion bei Sprachsignalen in automotiver Umgebung," Diplomarbeit, Technische Universität Wien E389, May 2005.
- [4] C. Marro, Y. Mahieux, and K. U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 3, pp. 240–259, May 1998.
- [5] I. A. McCowan, *Microphone arrays: A tutorial*, Apr. 2001.
- [6] G. Doblinger, *MATLAB - Programmierung in der digitalen Signalverarbeitung*. J. Schlembach Fachverlag Deutschland, 2001.
- [7] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Prentice-Hall International, Inc., 1996.
- [8] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc., 1988. (ICASSP-88)*, vol. 5, New York, USA, Apr. 1988, pp. 2578–2581.
- [9] J. B. Allen, D. A. Berkley, and J. Blauert, "Multimicrophone signal-processing technique to remove room reverberation from speech signals," *J. Acoust. Soc. Amer. (JASA)*, vol. 62, no. 4, pp. 912–915, Oct. 1977.
- [10] R. K. Cook, R. V. Waterhouse, R. D. Berendt, S. Edelmann, and M. C. Thompson, Jr., "Measurement of correlation coefficients in reverberant sound fields," *J. Acoust. Soc. Amer. (JASA)*, vol. 27, no. 6, pp. 1072–1077, Nov. 1955.
- [11] H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 10, pp. 1365–1376, Oct. 1987.
- [12] D. Stöbich, "Entwurf und Simulation eines adaptiven, zweidimensionalen Mikrofonarrays," Diplomarbeit, Technische Universität Wien E389, Sept. 2001.

- [13] E. N. Gilbert and S. P. Morgan, "Optimum design of directive antenna arrays subject to random variations," *Bell Syst. Tech. J.*, pp. 637–663, May 1955.
- [14] K. U. Simmer and A. Wasiljeff, "Adaptive microphone arrays for noise suppression in the frequency domain," in *Second Cost 229 Workshop Adapt. Alg. Communicat.*, Bordeaux, France, Oct. 1992, pp. 185–194.
- [15] I. A. McCowan and H. Bourlard, "Microphone array post-filter for diffuse noise field," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc., 2002. (ICASSP '02)*, vol. 1, May 2002, pp. I-905 – I-908.
- [16] I. A. McCowan and H. Bourlard, "Microphone array post-filter based on noise field coherence," *IEEE Trans. Speech Audio Processing*, vol. 11, no. 6, pp. 709–716, Nov. 2003.
- [17] J. Bitzer, K. U. Simmer, and K.-D. Kammeyer, "Multi-microphone noise reduction by post-filter and superdirective beamformer," in *Proc. Int. Workshop Acoust. Echo and Noise Control*, Pocono Manor, USA, Sept. 1999, pp. 100–103.
- [18] J. Bitzer, K. U. Simmer, and K.-D. Kammeyer, "An alternative implementation of the superdirective beamformer," in *Proc. IEEE Workshop Applicat. Signal Processing to Audio and Acoust.*, New Paltz, New York, Oct. 1999, pp. 7–10.
- [19] C. Marro, Y. Mahieux, and K. U. Simmer, "Performance of adaptive dereverberation techniques using directivity controlled arrays," in *Proc. EURASIP European Signal Proc. Conf. (EUSIPCO)*, Trieste, Italy, Sept. 1996, pp. 1127–1130.
- [20] G. F. A. Guerin, R. Le Bouquin-Jeannes, and G. Faucon, "A two-sensor noise reduction system: Applications for hands-free car kits," *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1125–1134, Mar. 2003.
- [21] S. R. Quackenbush, T. P. Barnwell III, and M. A. Clements, *Objective Measures of Speech Quality*. Englewood Cliffs, New Jersey: Prentice-Hall International, Inc., 1988.
- [22] "IEEE Recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. 17, no. 3, pp. 225–246, Sept. 1969.
- [23] J. R. Deller, Jr., J. G. Proakis, and J. H. L. Hansen, *Discrete-Time processing of speech signals*. New York, USA: Macmillan Publishing Company, 1993.
- [24] J. H. L. Hansen and B. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," in *Inter. Conf. on Spoken Language Processing (ICSLP-98)*, vol. 7, Sydney, Australia, Dec. 1998, pp. 2819–2822.
- [25] G. A. Miller, "The magical number seven, plus or minus two," *The Psychological Review*, vol. 63, pp. 81–97, 1956.

-
- [26] S. Fischer and K.-D. Kammeyer, "Broadband beamforming with adaptive postfiltering for speech acquisition in noisy environments," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc., 1997. (ICASSP-97)*, vol. 1, Munich, Germany, Apr. 1997, pp. 359–362.
 - [27] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer. (JASA)*, vol. 65, no. 4, pp. 943–950, Apr. 1979.
 - [28] J. Bitzer, K. U. Simmer, and M. Kallinger, "Robust, time-variant design of MVDR Beamformers," in *29. Jahrestagung fuer Akustik (DAGA-2003)*, Aachen, Germany, Mar. 2003, pp. 1–2.
 - [29] R. Martin, "Spectral subtraction based on minimum statistics," in *EUSIPCO 94*, Edinburgh, UK, 1994, pp. 1182–1185.
 - [30] R. Martin, "Noise power density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 5, pp. 504–512, July 2001.