

Vergleich Robuster Mikrofonarrays

Diplomarbeit

durchgeführt von

Lutz Pape

Institut für Elektronische Musik und Akustik
der Universität für Musik und darstellenden Kunst, Graz

Leiter: o.Univ.-Prof. Mag. DI Dr. Robert Höldrich

Begutachter: o.Univ.-Prof. Mag. DI Dr. Robert Höldrich

Betreuer: DI Markus Noisternig

Graz, im Juli 2005

Zusammenfassung

Ziel dieser Diplomarbeit ist die Entwicklung eines Mikrofonarrays zur verbesserten Sprachwiedergabe in der Konferenzsituation. Diesbezüglich wird ein Mikrofonarray fix vor einem Sprecher installiert (z.B. Tisch, Wand).

Zuerst wird ein Überblick über die grundlegenden Theorien des Array-Processing vor allem hinsichtlich des Beamformings mittels Mikrofonarrays geliefert. Ausgehend vom *Delay & Sum* Beamforming (D&S-B) werden zwei weitere Methoden, der Minimum Varianz Ansatz wie auch der *Filter & Sum* Beamformer (F&S-B), erläutert, die eine Möglichkeit zur Optimierung hinsichtlich der Keulenformung bieten. Zusätzlich werden jeweils kurz die Vor- und Nachteile eines jeden Ansatzes aufgezeigt.

In weiterer Folge werden die Verhaltensweisen bezüglich der Richtcharakteristik grundsätzlicher Sensorgeometrien in einem Mikrofonarray beschrieben. Durch Untersuchung der Keulenformung und Steeringigenschaften eines Delay & Sum Beamformers bei unterschiedlichen Anordnungen wird eine Arraystruktur mit kleiner Mikrofonanzahl gefunden, die sich sehr gut für die gestellten Anforderungen eignet.

Mit der gefundenen Geometrie werden verschiedene aus der aktuellen Literatur ausgewählte Beamforming Algorithmen mathematisch erläutert. Diese werden mit MATLAB simuliert und schließlich in der Echtzeit Programmierumgebung *Pure Data* (PD) implementiert, um dort jeweils eine Realisierung der Ansätze zu erhalten. Die Simulationen sollen die prinzipiellen Möglichkeiten der Optimierung der jeweiligen Ansätze zur Realisierung von Breitbandbeamformern veranschaulichen. Die Echtzeitrealisierungen werden sowohl mittels akustischer Vermessung, als auch subjektiv in einer typischen Anwendungssituation getestet. Anschließend erfolgt ein Vergleich mit den Ergebnissen aus den Simulationen. Jeder Ansatz wird bezüglich seiner Robustheit evaluiert. Für einen der behandelten adaptiven Beamformer wird eine Verbesserung der Robustheit gegenüber Arrayimperfektionen und Umwelteinflüssen vorgestellt.

Im letzten Kapitel wird ein Ausblick weitere Möglichkeiten zur Verbesserung der Algorithmen und Anmerkungen zu alternativen Ansätzen geben.

Abstract

The aim of this master thesis is the development of a microphone array to improve speech capture capabilities in conference situations. Therefore, a microphone with several capsules and small aperture will be installed in front of a speaker (e. g. stand alone on the desktop or surface mounted on walls).

First of all, an overview over basic theorems relating to acoustical array processing and beamforming algorithms is given. Beginning with the simple delay and sum (D&S) approach two more fundamental algorithms, the minimum variance (MV) as well as the filter and sum (F&S) beamformer, are discussed in detail. The advantages and disadvantages of each approach, considering robust beamforming applications, are going to be shown as well.

Using delay and sum principles an array structure is derived that optimizes steering capabilities over a broad frequency range (150 Hz – 8 kHz) under the constraints of small aperture size and minimum number of capsules.

Next, several beamforming theories, shown in recent literature, are illustrated due to mathematical derivation of the algorithms as well as simulations in MATLAB. Furthermore, the simulations shall point out the abilities of the approaches to optimize properties of the beamforming structures concerning to broadband applications. Real time implementations, using Miller Puckettes software Pure Date (PD), are used for acoustical measurements on the one hand and informal listening tests on the other. The results will be compared to the theoretical studies. Each approach is evaluated regarding to its robustness against array imperfections. A method to improve the overall robustness will be proposed at the end of this master thesis.

In the last chapter, possibilities to further improve the proposed Robust Adaptive Beamformer (RAB) and prospects to alternative approaches are given.

Inhaltsverzeichnis

1	Einleitung	10
1.1	Motivation	10
1.2	Aufgabenstellung.....	11
2	Beamforming	12
2.1	Theorie des Array-Processing	12
2.1.1	Kontinuierliche Aperturen.....	12
2.1.2	Diskretisierte Aperturen	16
2.2	Delay & Sum Beamformer.....	17
2.2.1	Prinzip	17
2.2.2	Vorteile.....	19
2.2.3	Nachteile.....	20
2.3	Minimum Varianz Beamformer	21
2.3.1	Prinzip	21
2.3.2	Vorteile.....	25
2.3.3	Nachteile.....	25
2.4	Filter & Sum Beamformer.....	26
2.4.1	Feststehende Filter.....	26
2.4.2	Adaptive Filter.....	27
3	Arraygeometrie	29
3.1	Anforderungen an das Array	29
3.2	Verschiedene Geometrien	29
3.2.1	Simulationen.....	30
3.3	Das Mikrofonarray	33
3.3.1	Simulationen.....	33
3.3.2	Bau des Arrays	34

4	Ausgewählte Algorithmen	36
4.1	Fixed Filter & Sum Beamformer.....	36
4.1.1	Theorie	36
4.1.2	Simulationen.....	37
4.1.3	Auswertung und Vergleich der Simulationen	40
4.1.4	Realisierung in PD und akustische Vermessung.....	40
4.1.5	Evaluierung	41
4.2	Frost Beamformer.....	41
4.2.1	Theorie	41
4.2.2	Simulationen.....	44
4.2.3	Auswertung und Vergleich der Simulationen	46
4.2.4	Realisierung in PD.....	47
4.2.5	Evaluierung	47
4.3	Frost Beamformer mit Erhöhung von G_w	47
4.3.1	Theorie	47
4.3.2	Simulationen.....	49
4.3.3	Auswertung und Vergleich der Simulationen	50
4.3.4	Realisierung in PD und akustische Vermessung.....	51
4.3.5	Evaluierung	51
4.4	Frost Beamformer mit „ <i>Steering Vector Correction</i> “	52
4.4.1	Theorie	52
4.4.2	Simulationen.....	56
4.4.3	Auswertung und Vergleich der Simulationen	58
4.4.4	Realisierung in PD.....	58
4.4.5	Evaluierung	59
4.5	Griffiths-Jim Beamformer.....	59
4.5.1	Überleitung vom Frost Beamformer	59
4.5.2	CCAF-NCAF Struktur mit AMC.....	61
4.5.3	Simulationen.....	65
4.5.4	Auswertung und Vergleich der Simulationen	68
4.5.5	Verbesserung des AMC.....	69
4.5.6	Realisierung in PD und akustische Vermessung.....	76
4.5.7	Evaluierung	81
5	Messaufbau	82
5.1	Aufbau.....	82
5.2	Methoden:.....	83
5.3	Varianzabschätzung der Messergebnisse	84

6	Zusammenfassung und Ausblick	86
6.1	Zusammenfassung.....	86
6.2	Ausblick.....	87
7	Literaturverzeichnis	89
8	Anhang: MATLAB Funktionen	90
8.1	Listen und Kurzbeschreibungen.....	90
8.2	Datensätze und Testsignale	91
8.3	Aufgenommene Messdaten.....	93

Abbildungsverzeichnis

Abb. 2.1: direktive Systemantwort einer Apertur [7].....	13
Abb. 2.2 links: Geometrie des Directivity Pattern, rechts: lin. Apertur auf der x-Achse [7].....	14
Abb. 2.3 oben: Aperturfunktion aus (2.5), unten: Directivity Pattern aus (2.6) [7]	15
Abb. 2.4: Polardiagramme für versch. Frequenzen aus (2.7) [7].....	15
Abb. 2.5: diskrete, lineare Apertur, $N = 5$ [7]	16
Abb. 2.6: räumliches Aliasing links: $d = \frac{\lambda_{\min}}{2}$, rechts: $d = \lambda_{\min}$	17
Abb. 2.7: Beam-Steering für $\Phi' = 45^\circ$, $f = 1\text{kHz}$, $N = 10$, $d = 0,15\text{m}$ [7]	18
Abb. 2.8: Polardiagramme einer D&S-B Messung vs. Simulation [9]	20
Abb. 2.9: Directivity-Pattern, $400\text{Hz} \leq f \leq 3\text{kHz}$ ($N = 5$, $d = 1\text{m}$) [7]	21
Abb. 2.10: Signalphasenverschiebung an den Mikrofonen, Schallquelle = $S(j\omega)$ [11].....	22
Abb. 2.11: Strukturbild eines Schmalband Beamformers [11].....	22
Abb. 2.12: Vergleich von MV-B und D&S-B (MATLAB).....	24
Abb. 2.13: Beampatterns von MV-B (li.) und D&S-B (re.) über die Frequenz (MATLAB)	24
Abb. 2.14: Gewichte für einen Sensor im MV-B (MATLAB).....	25
Abb. 2.15: Strukturbild eines einfachen Breitband Filter & Sum Beamformers [11].....	26
Abb. 2.16: berechnete Frequenzgänge für ein 4 Elemente Array (MATLAB).....	27
Abb. 2.17: mit Least Squares approx. Frequenzg. für ein 4 El. Array (MATLAB).....	27
Abb. 2.18: FIR-Filterkoeffizienten für ein 4 Elemente Array (MATLAB).....	27
Abb. 2.19: Strukturbild eines einfachen adaptiven Beamformers [10]	28
Abb. 3.1: 3D Richtplot eines lin. 4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB)	30
Abb. 3.2: 3D Richtplot eines lin. 4x4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB).....	30
Abb. 3.3: 3D Richtplot eines lin. 4x4x4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB)	30
Abb. 3.4: Vergleich von Steeringigenschaften linear (li.) vs. Bogen (re.) (MATLAB)	32
Abb. 3.5: Steeringverhalten des 6 Elemente Sterns (MATLAB).....	33
Abb. 3.6: 6 Elemente Stern, Beampattern vs. Frequenz, $200\text{Hz} \leq f \leq 8\text{kHz}$ (MATLAB)	34

Abb. 3.7: verwendete Mikrofonkapsel (li), Vorverstärker (re)	35
Abb. 3.8: Bild von Array (li), Geometrie (re)	35
Abb. 4.1: Richtplot für opt. Filtergewichte, $\mu=0$ (MATLAB)	38
Abb. 4.2: Richtplots für $\mu=0,0001$, $\mu=0,0005$, $\mu=0,1$, $\mu=100$ (MATLAB)	39
Abb. 4.3: Richtplots F&S-B bei Verstärkungs- und Positionierungsfehlern (MATLAB).....	39
Abb. 4.4: gemessene Ausgangsleistung des F&S-B mit weißem und pinkem Rauschen.....	41
Abb. 4.5 oben: Strukturbild des Frost Beamformers, unten: Ersatzprozessor	42
Abb. 4.6: Vergleich: Richtplots von D&S-B, F&S-B und Frost Beamformer (MATLAB).....	45
Abb. 4.7: Frost Beamformer mit Positionierungs- und Verstärkungsfehlern (MATLAB).....	46
Abb. 4.8: Frost-B mit versch. G_w -Erhöhungen (MATLAB).....	49
Abb. 4.9: Vergleich von LMS zu NLMS	49
Abb. 4.10: Frost-B mit Positionierungs- und Verstärkungsfehlern, G_w -Erhöhung: 5dB (MATLAB).....	50
Abb. 4.11: gemessene Ausgangsleistung des Frost-B mit angehobenem $G_w = 5\text{dB}$	51
Abb. 4.12: Frost Beamformer mit adaptiver Steering Vector Correction.....	56
Abb. 4.13: Adaption des Algorithmus bei 10° Targetverfehlung (MATLAB)	57
Abb. 4.14: Richtplot: Frost Beamformer mit adapt. Steering Vector Correction (MATLAB).....	57
Abb. 4.15: Strukturbild des Griffiths-Jim Beamformers	60
Abb. 4.16: G-J-B mit CCA-, bzw. NCA-Filtern und AMC.....	61
Abb. 4.17: Schematische Darstellung der Arbeitsweise von CCAFs [6].....	63
Abb. 4.18: Darstellung der Koeffizientenlimits und der zulässigen Target Region (MATLAB).....	65
Abb. 4.19: Richtplot des RAB mit untersch. Coeff. Constr. und mit Verstärkungsfehler (MATLAB).....	66
Abb. 4.20: Darstellung der Arbeitsweise des RAB mit AMC (MATLAB)	67
Abb. 4.21: Schätzungen der SIR bei verschiedenen Einfallswinkeln der Störsignale (MATLAB).....	67
Abb. 4.22: Verhalten des AMC bei Sprachsignal aus -30° als Störer, 3dB SIR (MATLAB)	68
Abb. 4.23: Schätzung des Rauschlevels durch P_z und P_b bei untersch. Winkeln (MATLAB)	70
Abb. 4.24: Verschiebung des SIR-Schätzers durch Limiter/ Expander mit Sidechain (MATLAB).....	71
Abb. 4.25: Schema der SRPR mit 4 Seitenbeams und einem Hauptbeam.....	73
Abb. 4.26: linker Seitenbeam (li) und benutzter Frequenzbereich (re) des AMC (MATLAB)	74
Abb. 4.27: Verlauf der $SRPR_{\text{left}}$ (MATLAB).....	74
Abb. 4.28: verbesserter Adaptation Mode Controller	75

Abb. 4.29: Leistungsplots von Einzelmikrofon, D&S-B, RAB.....	76
Abb. 4.30: gemessener Leistungsplot mit strenger/ lockerer Koeffizientenbeschränkung, weißes Rauschen.....	77
Abb. 4.31: Leistungsplot mit lockerer Koeffizientenbeschränkung, rosa Rauschen.....	77
Abb. 4.32: RAB Beampattern für verschiedene Oktavbänder (125Hz bis 8kHz).....	78
Abb. 4.33: Frequenzverlauf des RAB in Oktavbänder unterteilt.....	78
Abb. 4.34: Leistungsplot bei eingeschaltetem AMC mit versch. Steuerparametern, weißes Rauschen.....	79
Abb. 4.35: Leistungsplot mit AMC bei verschiedenen SIRs, weißes Rauschen	80
Abb. 4.36: Plots mit AMC, Sprachsignal und weißes Rauschen.....	80
Abb. 5.1: Messaufbau.....	82
Abb. 5.2: gemessene Richtcharakteristik der Mikrofonkapseln (weißes Rauschen).....	84
Abb. 6.1: alle vermessenen Beamformer im Vergleich.....	86
Abb. 6.2: RAB mit AMC, Steering:-30°, bei untersch. SRPR sidebeam Gewichtungen	88

1 Einleitung

1.1 Motivation

Die uneingeschränkte Kommunikation zwischen Menschen an beliebigen Orten bildet heutzutage eine der wichtigsten Grundlagen für das soziale Leben. Niemand möchte mehr auf die Vorteile der modernen Telekommunikation verzichten müssen und für viele Leute ist sie zu einem zentralen Lebensbestandteil geworden. Auch wird der Ruf nach einer intuitiven Mensch-Maschine Interaktion immer lauter und zieht somit einen riesigen Forschungsbereich auf diesem Gebiet mit sich, wie zum Beispiel die Sprachsignalerfassung und digitale Verarbeitung. Daher liegt die Idee oder die Motivation nahe, die an einen Benutzer gestellten Anforderungen und Einschränkungen zu minimieren und so gering wie möglich zu halten.

Ein Ansatz dabei ist es, schon in einer der ersten Stufen der digitalen Kommunikation größt mögliche Freiheiten anbieten zu können: Die Aufnahme der Sprache. Nicht selten ist dies damit verbunden, ein Mikrofon in unmittelbarer Nähe des Sprechers platzieren zu müssen, da sonst aufgrund von störenden Umwelteinflüssen keine ausreichende Signalqualität erzielt werden kann. Diese Notwendigkeit ist in den meisten Fällen mit einer nicht immer angenehmen Einschränkung des Benutzers verbunden, wie zum Beispiel der Verlust der freien Beweglichkeit durch einen Telefonhörer, ein Headset oder ein Ansteckmikrofon.

Um einen Sprecher davon befreien zu können, muss das Mikrofon aus dem direkten Aktionsbereich des Benutzers entfernt werden. Das geschieht heute schon zum Teil durch Freisprecheinrichtungen und Richtmikrofone. Nur stoßen diese Techniken sehr schnell an ihre Grenzen. Die Sprache wird bei Freisprecheinrichtungen mit steigenden Umgebungsgeräuschen immer unverständlicher und Richtmikrofone müssen immer exakt auf die Position des Sprechers treffen, was bei einer Bewegung eine mechanische Korrektur der Position des Mikrofons erfordert. Seit einigen Jahren existiert ein aus der Radartechnik bekannter Ansatz der Sensor-Arrays. Hiermit ist es durch die so genannte Technik des „*Beamformings*“ möglich mittels eines Mikrofon-Arrays Sprachsignale (*Targets*) zu lokalisieren, zu betonen und zu verfolgen und gleichzeitig unerwünschte Störsignale (*Interferences*) auszublenden.

Da diese Methode heute durch ausreichende Rechenleistung der Signalprozessoren und billige Mikrofonpreise umsetzbar geworden ist, scheint die Möglichkeit, solche Systeme zu entwickeln und einzusetzen in greifbare Nähe gerückt zu sein.

1.2 Aufgabenstellung

Unterschiedliche Methoden des Mikrofonarray Beamformings sollen bezüglich der Möglichkeit der Separation von Schallquellen in der Konferenzsituation untersucht und vergleichend gegenübergestellt werden. Die Minimierung der Array-Apertur, bei gleichzeitiger Beschränkung auf eine maximale Anzahl von sechs Array-Elementen, steht dabei im Vordergrund. Anhand von Simulationen ist die Wirkungsweise der Algorithmen zur Verarbeitung von Sprachsignalen darzustellen, sowie die gefundenen Algorithmen auf Robustheit gegenüber Array-Imperfektionen zu untersuchen. Durch Implementierung in einer Echtzeitumgebung soll eine Evaluierung der Simulationsergebnisse erfolgen.

2 Beamforming

Dieses Kapitel soll einen kurzen Überblick über die grundlegenden Theorien des Array-Processing vor allem hinsichtlich des Beamformings mittels Mikrofonarrays liefern. Ausgehend vom Delay & Sum Beamforming (D&S-B) werden zwei weitere Methoden, der Minimum Varianz Ansatz wie auch der Filter & Sum Beamformer, erläutert, die eine Möglichkeit zur Optimierung hinsichtlich der Keulenformung bieten. Zusätzlich werden jeweils kurz die Vor- und Nachteile eines jeden Ansatzes aufgezeigt. Da auf die mathematischen Hintergründe der Array-Theorie nur teilweise und so tief wie nötig eingegangen wird, verweise ich für detaillierte mathematische Erläuterungen auf [7].

2.1 Theorie des Array-Processing

2.1.1 Kontinuierliche Aperturen

Ein Mikrofonarray kann als eine diskrete passive Apertur betrachtet werden. Als passive Apertur bezeichnet man einen definierten Bereich, in den Wellen eintreten. Eine kontinuierliche Apertur kann man mathematisch durch ihre Aperturfunktion zum Ausdruck bringen:

In einem infinitesimal kleinen Volumen dV (passive Apertur) an einem Ort \mathbf{r} tritt zu einer Zeit t das Signal $x(t, \mathbf{r})$ ein. Wenn man dV an der Stelle \mathbf{r} als linearen Filter mit der Impulsantwort $a(t, \mathbf{r})$ betrachtet, kann das empfangene Signal $x_R(t, \mathbf{r})$ als Faltung von $x(t, \mathbf{r})$ mit $a(t, \mathbf{r})$ betrachtet werden (s. Abb. 2.2).

$$x_R(t, \mathbf{r}) = \int_{-\infty}^{+\infty} x(\tau, \mathbf{r}) a(t - \tau, \mathbf{r}) d\tau \quad (2.1)$$

Oder im Frequenzbereich:

$$X_R(f, \mathbf{r}) = X(f, \mathbf{r}) A(f, \mathbf{r}) \quad \text{mit} \quad A(f, \mathbf{r}) = F\{a(t, \mathbf{r})\} \quad (2.2)$$

Der Term $A(f, \mathbf{r})$ wird als die Aperturfunktion oder Empfindlichkeitsfunktion bezeichnet und definiert die Systemantwort als eine Funktion der räumlichen Position auf der Apertur [7].

Für die weiteren Betrachtungen wird zur Vereinfachung der mathematischen Ableitung von einer ebenen Wellenausbreitung ausgegangen. Da in dieser Arbeit Arrays mit kleinen Aperturen untersucht werden sollen, spielt die Theorie des Nahfeld Beamforming eine untergeordnete Rolle. Die meisten Anwendungen sind nur für Fernfeldbedingungen relevant.

Die Systemantwort einer passiven Apertur ist direktiv und somit wird, abhängig von der Einfallsrichtung des Signals, nur ein gewisser Anteil des Signals „gesehen“ (s. Abb. 2.1). Die Systemantwort der Apertur als Funktion der Frequenz wird als *Directivity Pattern* oder *Beam Pattern* bezeichnet. Die Aperturfunktion und das Directivity Pattern sind über die Fouriertransformation verknüpft [7].

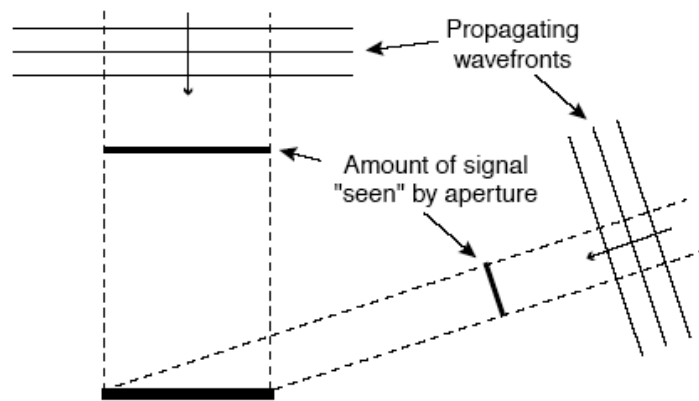


Abb. 2.1: direkte Systemantwort einer Apertur [7]

Das Fernfeld Directivity Pattern $D(f, \boldsymbol{\alpha})$ mit der Aperturfunktion $A(f, \mathbf{r})$ berechnet sich aus (F_r ist die räumliche Fourier-Transformation):

$$D(f, \boldsymbol{\alpha}) = F_r \{ A(f, \mathbf{r}) \} = \int_{-\infty}^{+\infty} A(f, \mathbf{r}) e^{j2\pi \boldsymbol{\alpha} \mathbf{r}} d\mathbf{r} \quad (2.3)$$

$$\mathbf{r} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \quad \boldsymbol{\alpha} = f \frac{\mathbf{k}}{\omega} = \frac{1}{\lambda} [\cos \theta \cos \phi \quad \cos \theta \sin \phi \quad \sin \theta]$$

$$\text{Wellenzahlvektor } \mathbf{k} = \frac{2\pi}{\lambda} [\cos \theta \cos \phi \quad \cos \theta \sin \phi \quad \sin \theta]$$

$$\text{Wellenlänge } \lambda = \frac{c}{f} \quad \text{Kreisfrequenz } \omega = 2\pi f$$

\mathbf{r} ist der Vektor auf einen Punkt entlang der Apertur, $\boldsymbol{\alpha}$ ist die Einfallsrichtung des Schalls mit dem Azimutwinkel ϕ und dem Elevationswinkel θ (Abb. 2.2 links). Die Frequenzabhängigkeit liegt in der Wellenlänge λ .

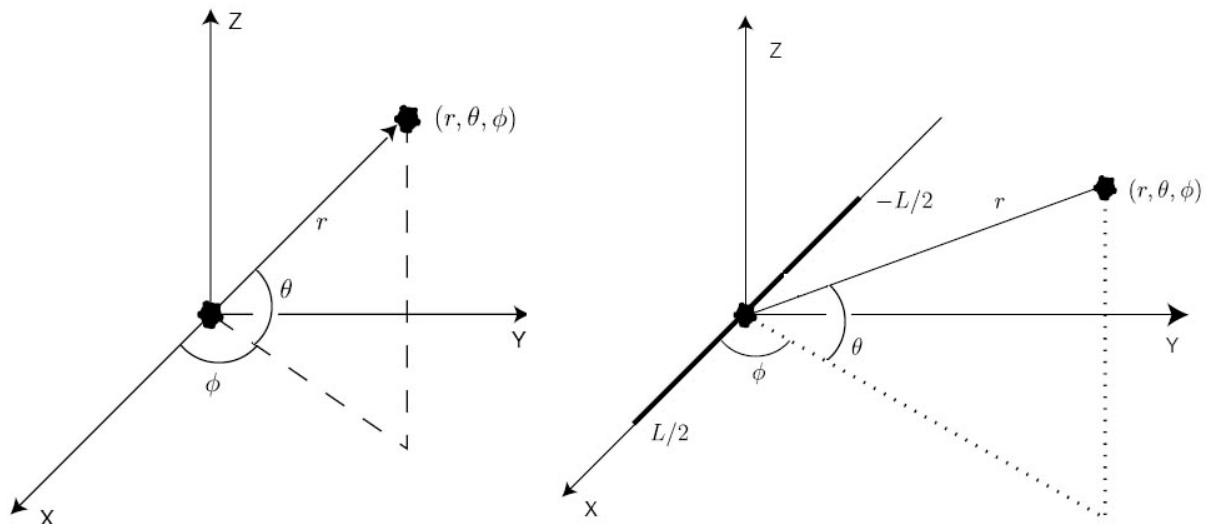


Abb. 2.2 links: Geometrie des Directivity Pattern, **rechts:** lin. Apertur auf der x-Achse [7]

Um die Eigenschaften eines Directivity Pattern zu verdeutlichen wird eine lineare Apertur der Länge L entlang der x-Achse angenommen (Abb. 2.2 rechts) [7].

$$\mathbf{r} = \begin{bmatrix} x_a \\ 0 \\ 0 \end{bmatrix} \Rightarrow D(f, \alpha_x) = \int_{-L/2}^{L/2} A(f, x_a) e^{j2\pi\alpha_x x_a} dx_a$$

$$\rightarrow \alpha_x = \frac{\cos\theta \cos\phi}{\lambda}$$

$$\Rightarrow D(f, \theta, \phi) = \int_{-L/2}^{L/2} A(f, x_a) e^{j\frac{2\pi}{\lambda} \cos\theta \cos\phi x_a} dx_a \quad (2.4)$$

Besitzt diese Apertur eine einheitliche, frequenzunabhängige Aperturfunktion (Abb. 2.3 oben)

$$A(x_a) = \text{rect}\left(\frac{x_a}{L}\right) \hat{=} \begin{cases} 1 & |x| \leq \frac{L}{2} \\ 0 & |x| \geq \frac{L}{2} \end{cases} \quad (2.5)$$

so ergibt sich aus Gl. (2.4) siehe auch Abb. 2.3 unten:

$$D(f, \alpha_x) = F_r \left\{ \text{rect}\left(\frac{x_a}{L}\right) \right\} \Rightarrow D(f, \alpha_x) = L \text{sinc}(\alpha_x L) \quad (2.6)$$

Normalisiert mit $1/L$, auf Winkelform gebracht und nur unter Betrachtung des Azimutwinkels Φ ergibt sich:

$$D(f, \frac{\pi}{2}, \phi) = \text{sinc}\left(\frac{L}{\lambda} \cos\phi\right) \quad (2.7)$$

Dies entspricht einem räumlichen Filter, bei dem die Nullstellenbreite (*Beam Width*) der sinc-Funktion von der Wellenlänge und damit von der Frequenz abhängt. Je höher die Frequenz, desto enger wird der Beam und umso mehr Nebenkeulen und Nullstellen werden sichtbar. Die Darstellung im Polardiagramm verdeutlicht diese Tatsache (Abb. 2.4).

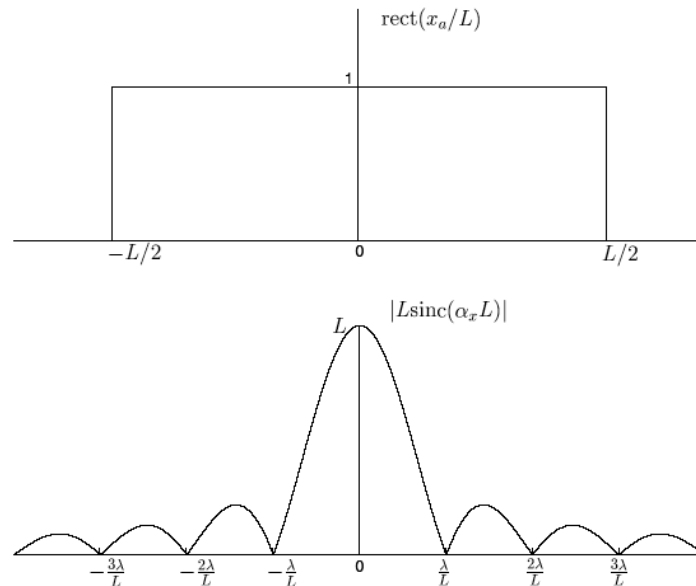


Abb. 2.3 oben: Aperturfunktion aus (2.5), unten: Directivity Pattern aus (2.6) [7]

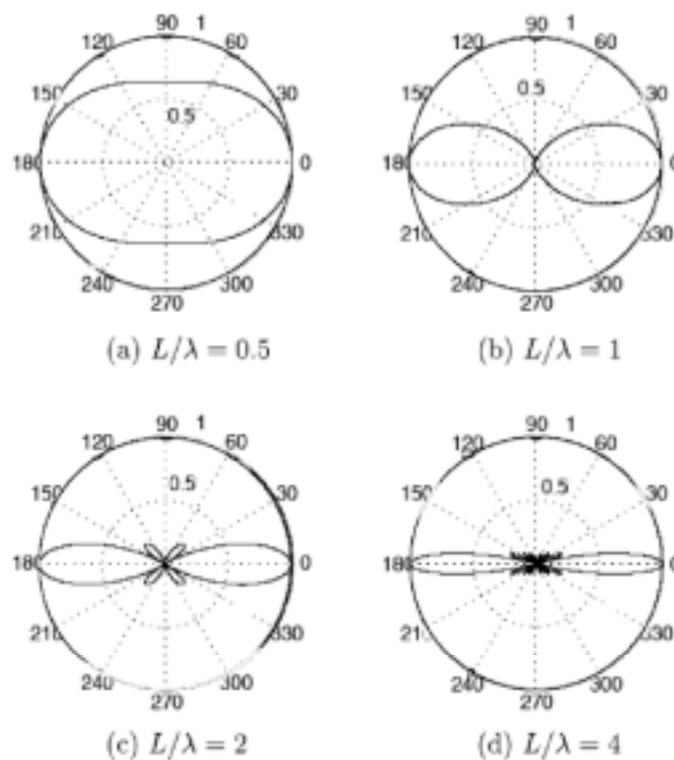


Abb. 2.4: Polardiagramme für versch. Frequenzen aus (2.7) [7]

2.1.2 Diskretisierte Aperturen

Betrachtet man nun die lineare Apertur aus Abb. 2.2 nur an einer endlichen Anzahl N von Punkten, so lässt sich von einer diskretisierten Apertur - einem Sensorarray sprechen (Abb. 2.5). Ein Sensor-Array kann als räumlich abgetastete Version einer kontinuierlichen Apertur angesehen werden. Jeder Punkt bildet dabei selbst eine kontinuierliche Apertur. Die gesamte Antwort des Arrays ergibt sich aus der Superposition der einzelnen Sensorpunkte. Das daraus resultierende Ergebnis ist eine Approximation an die äquivalente kontinuierliche Apertur. Bei einem Sensor-Array hat jeder Sensor an seiner Position x_n eine eigene komplexe Frequenzantwort $e_n(f, x)$, die zusätzlich mit einer Fensterfunktion $w_n(f)$ gewichtet wird (die Fensterfunktion wird zu einem späteren Zeitpunkt genauer betrachtet). Die diskretisierte Aperturfunktion lässt sich wie folgt anschreiben [7]:

$$A(f, x_a) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e_n(f, x_a - x_n) \Rightarrow D(f, \alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) E_n(f, \alpha_x) e^{j2\pi\alpha_x x_n} \quad (2.8)$$

Wird $E_n(f, \alpha_x)$ für alle Sensoren gleich 1 vorausgesetzt, kann (2.8) vereinfacht werden und es ergibt sich das entsprechende Directivity-Pattern (2.9). Haben alle N Sensoren auch noch den gleichen Abstand d , lässt sich (2.9) unter ausschließlicher Betrachtung des Azimutwinkels (s. 2.7) zu (2.10) reduzieren (s. auch Abb. 2.5).

$$A(f, x_a) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) \delta(x_a - x_n) \Rightarrow D(f, \alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j2\pi\alpha_x x_n} \quad (2.9)$$

$$D(f, \alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j2\pi\alpha_x nd} \quad \text{mit } x_n = nd$$

$$\Rightarrow D(f, \phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi}{\lambda} nd \cos\phi} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} w_n(f) e^{j\frac{2\pi f}{c} nd \cos\phi} \quad (2.10)$$

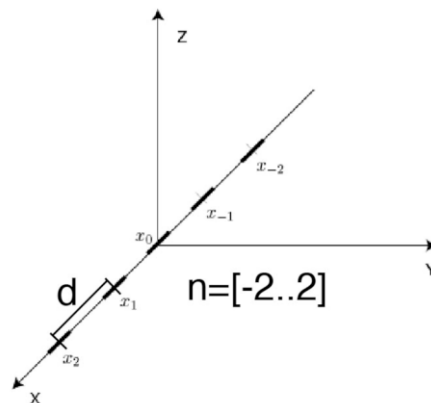


Abb. 2.5: diskrete, lineare Apertur, $N = 5$ [7]

Durch diese Diskretisierung kann es jedoch genau wie auch bei Abtastung in der zeitdiskreten Signalverarbeitung zu Aliasing kommen (vgl. Abtastung in der zeitdiskreten Signalverarbeitung). Im Fall der räumlichen Signalverarbeitung (Space-Time Signal Processing) spricht man von räumlichem Aliasing. Der minimale Abstand d zwischen 2 Sensoren in einer Anordnung bestimmt das an einer maximalen Frequenz $f_{x_{\max}}$ noch korrekt abgebildete Beampattern [7]. $f_{x_s} = \alpha_x$ und ergibt für die lineare Apertur entlang der x -Achse (2.4).

$$f_{x_s} = \frac{1}{d} \geq 2f_{\max} \quad \text{mit } f_{x_s} = \frac{\cos\theta \cos\phi}{\lambda} \Rightarrow f_{x_{\max}} = \frac{1}{\lambda_{\min}} \Rightarrow d < \frac{\lambda_{\min}}{2}$$

$$f_s = \frac{1}{t_s} \geq 2f_{\max} \Leftarrow \text{zeitliches Samplingtheorem}$$

$$f_{x_s} \rightarrow \text{räumliche Samplingfrequenz} \left[\frac{\text{Samples}}{m} \right] = \alpha_x$$

$$f_{\max} \rightarrow \text{obere Grenzfrequenz (Nyquist - Frequenz beim zeitl. Sampling)}$$

$$\lambda_{\min} \rightarrow \text{kleinste mögliche Wellenlänge}$$

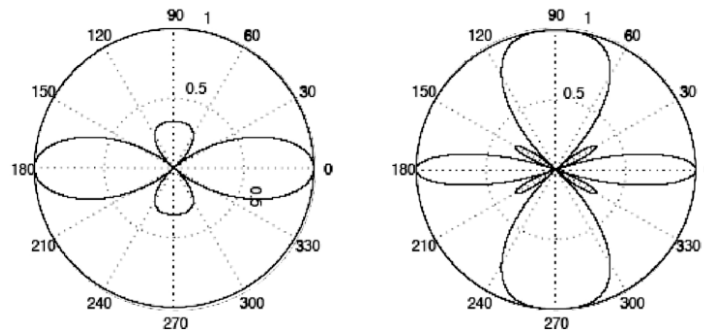


Abb. 2.6: räumliches Aliasing links: $d = \frac{\lambda_{\min}}{2}$, rechts: $d = \lambda_{\min}$

Es ist also möglich mit einer endlichen Anzahl von Sensoren das Verhalten einer kontinuierlichen Apertur anzunähern. Diese Tatsache führt, wie im folgenden Kapitel beschrieben, zum ersten Ansatz des Beamforming, dem Delay & Sum Beamforming.

2.2 Delay & Sum Beamformer

2.2.1 Prinzip

Bis jetzt wurde angenommen, dass die Fensterfunktion aus Gleichung (2.10) alle Sensoren gleich gewichtet und zwar mit $w_n(f)=1/N$. Die Fensterfunktion kann aber auch aus komplexen Gewichten bestehen [7].

$$w_n(f) = a_n(f)e^{j\varphi_n(f)} \quad (2.11)$$

$$a_n(f) \text{ (Amplitude)}, \varphi_n(f) \text{ (Phase)} \in \Re$$

Mit $w_n(f) \in \mathbb{C}$ ist es nun möglich Amplitude und Phase der jeweiligen vom Sensor aufgenommenen Signale unterschiedlich zu gewichten. Mit der Variation der Amplitudengewichte ist es möglich die Form des Beam Patterns zu verändern. Durch unterschiedliche Verzögerung der Phasen kann die Richtung der Hauptkeule variiert werden (*Steering*) (s. Abb. 2.7). Das Directivity Pattern lässt sich mit (2.11) anders anschreiben:

$$D(f, \phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j(2\pi\alpha_x nd + \varphi_n(f))} \quad \text{mit } a_n(f) = 1 \text{ und } \varphi_n(f) = -2\pi\alpha'_x nd$$

$$\alpha'_x = \frac{\cos\theta' \cos\phi'}{\lambda} \quad \text{mit } \alpha' = \text{Steeringwinkel}$$

$$\Rightarrow D'(f, \alpha_x) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{j\frac{2\pi}{\lambda} nd(\alpha_x - \alpha'_x)} = D(f, \alpha_x - \alpha'_x) \quad (2.12)$$

Die Betrachtung des Phasendelays im Zeitbereich zeigt, dass ein negativer Phase-Shift im Frequenzbereich eine Verzögerung im Zeitbereich entspricht (hier nur für ϕ'). Es ist also möglich, ein Steering nur durch unterschiedliche Verzögerungen der einzelnen Sensorsignale (bezogen auf einen Referenzsensor) zu erzielen. Wird das Delay entsprechend der Laufzeit der ebenen Welle vom Referenzsensor zum n-ten Sensor gewählt (Gl. 2.13), ergibt sich bei Summation der Signale eine konstruktive Überlagerung (Azimutwinkel betrachtet).

$$\tau_n = \frac{\varphi_n}{2\pi f} = \frac{2\pi f n d \cos\phi'}{2\pi f c} = \frac{n d \cos\phi'}{c} \quad (2.13)$$

Diese Verzögerung und anschließende Summation der Sensorsignale bezeichnet man als Delay & Sum Beamforming.

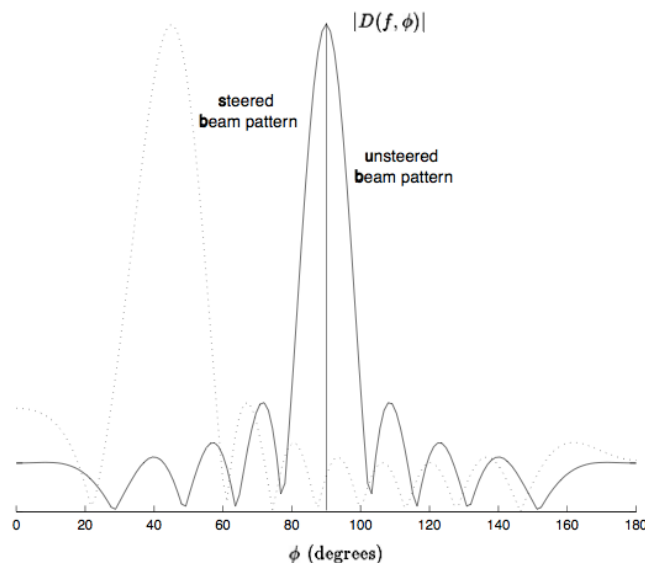


Abb. 2.7: Beam-Steering für $\Phi' = 45^\circ$, $f = 1\text{kHz}$, $N = 10$, $d = 0,15\text{m}$ [7]

Beim Delay & Sum Beamformer werden zuerst alle Sensorsignale so verzögert, dass sich die Hauptkeule in Richtung der Signalquelle ausrichtet. Danach werden alle Signale der Mikrofone addiert. Es entsteht ein räumlich gefiltertes Mono-Signal, das die Information aus der Hauptrichtung betont und Störsignale aus den anderen Richtungen durch destruktive Interferenzen unterdrückt.

$$D(f, \phi) = \frac{1}{N} \sum_{n=1}^N e^{j \frac{2\pi f (n-1)d (\cos\phi - \cos\phi')}{c}} \quad \text{mit} \quad \tau_n = \frac{(n-1)d \cos\phi'}{c}$$

$$\Rightarrow Y(f) = \frac{1}{N} \sum_{n=1}^N X_n(f) e^{j \frac{-2\pi f}{c} (n-1)d \cos\phi'}$$

$$\Rightarrow y(t) = \frac{1}{N} \sum_{n=1}^N x_n(t - \tau_n) \tag{2.14}$$

$$\tag{2.15}$$

2.2.2 Vorteile

Die Vorteile eines D&S-B sind aus obigen Beziehungen leicht ersichtlich. Diese Methode ist die einfachste und robusteste Möglichkeit zur räumlichen Filterung und somit zur Trennung von Signalen, die sich an unterschiedlichen Positionen im Raum befinden [1]. Das einfache Verzögern und Addieren der Mikrofon-signale lässt sich ohne besonders großen Rechenaufwand implementieren und realisieren. Ein Array dieser Art liefert schon eine wesentlich bessere akustische Isolierung des Target Signals als ein einzelnes, der im Array verwendeten Mikrofone - abhängig von der Arraygeometrie unter Betrachtung der Frequenzbereiche. Als Beispiel für die Robustheit und Performance werden real gemessene Polardiagramme aus [9] eines 25 Elemente Stern-Arrays ($d=3\text{cm}$, Radius= 15cm) aufgeführt, welches nur mit einem D&S-B Algorithmus belegt wurde. Die fett gezeichnete Linie stellt die Messergebnisse dar, der darunter liegende Beamplot zeigt die MATLAB Simulationsergebnisse mit Kugelmikrofonen. Das Array wurde ebenfalls mit Kugelmikrofonen realisiert. Diese waren allerdings durch eine Plexiglasscheibe nach hinten abgeschattet. Abb. 2.8 zeigt die Gegendarstellung verschiedener Polardiagramme des D&S-B gemessener Werte und simulierter Werte für verschiedene Frequenzen und Steering Richtungen.

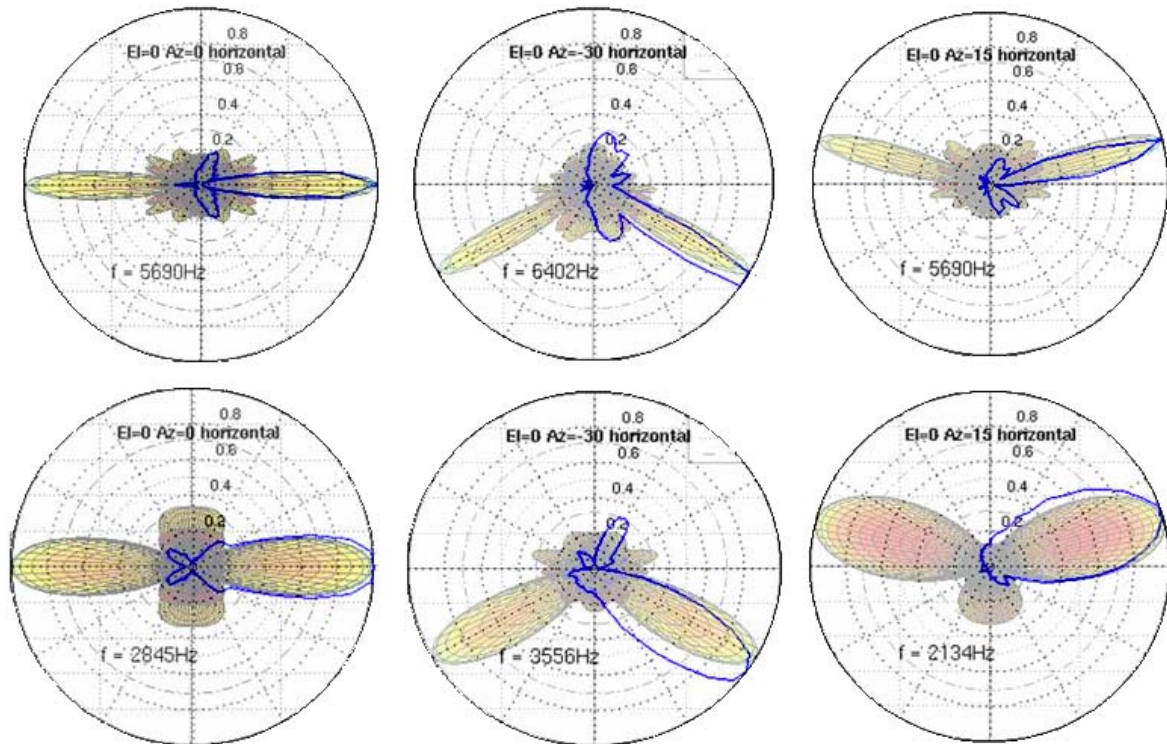


Abb. 2.8: Polardiagramme einer D&S-B Messung vs. Simulation [9]

2.2.3 Nachteile

Der größte Nachteil des D&S-B ist die starke Frequenzabhängigkeit des Directivity-Patterns. Durch den Term $2\pi f/c$ in (2.14) wird die Breite der Hauptkeule mit steigender Frequenz immer schmaler. Dies bewirkt am Ausgang des System eine Verzerrung des Spektrums des ursprünglichen Signals, was bedeutet, dass abhängig von der Arraygeometrie tiefe Frequenzen weniger stark gefiltert werden als hohe (Tiefpass) (s. Abb. 2.9). Das macht den D&S-B für breitbandige Signale wie z. B. Sprache nahezu unbrauchbar. Um eine stärkere Filterung von tiefen Frequenzen erreichen zu können, müsste man entweder den Abstand zwischen den Sensoren, oder die Anzahl der Sensoren bei gleichem Sensorabstand erhöhen. Diese Maßnahmen würden in beiden Fällen zu großen geometrischen Ausmaßen der Arrays führen und somit den Aufbau unhandlich und unverhältnismäßig aufwändig gestalten. Außerdem sinkt bei einem größeren Mikrofonabstand die obere Grenzfrequenz des Arrays und der Materialaufwand wird bei einer höheren Sensoranzahl zu groß.

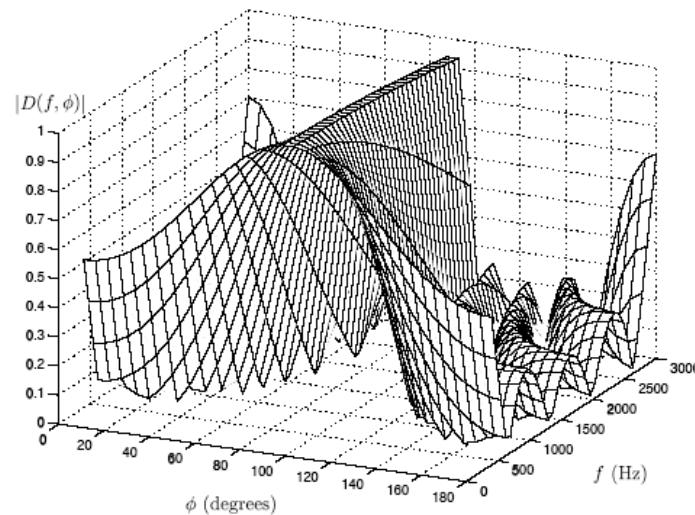


Abb. 2.9: Directivity-Pattern, $400\text{Hz} \leq f \leq 3\text{kHz}$ ($N = 5$, $d = 1\text{m}$) [7]

Die Frequenzabhängigkeit des D&S-B wird immer eine maßgebliche Einschränkung dieses Ansatzes bleiben. Deswegen versucht man durch Erweiterungen das Directivity Pattern über einen breiten Frequenzbereich konstant zu halten. Diese Bemühungen führen über verschiedene Varianten des D&S-B, wie z. B. „*Nested Arrays*“ (s. [1]) schließlich mit dem „*Minimum Variance*“ Ansatz zum „*Filter & Sum Beamformer*“.

2.3 Minimum Varianz Beamformer

2.3.1 Prinzip

Der Minimum Varianz Beamformer (MV-B) ist ein superdirektiver Ansatz. Superdirektive Mikrofonarrays erzielen eine höhere Direktivität¹ als D&S-B.

Wenn man den D&S-B aus einem anderen Blickwinkel wie in Kapitel 2.2 betrachtet, kann folgende Vektorschreibweise formuliert werden:

$$y = \mathbf{w}^H \mathbf{x} \quad (2.16)$$

$$\mathbf{w} = \mathbf{a} \cdot \mathbf{d}(\Phi, \Theta) \quad (2.17)$$

$$\mathbf{a} = [a_1, a_2, \dots, a_N]$$

$$\mathbf{d}(\Phi, \Theta) = \left[e^{j\frac{\omega}{c} \mathbf{r}_1 \cdot \mathbf{e}_r}, e^{j\frac{\omega}{c} \mathbf{r}_2 \cdot \mathbf{e}_r}, \dots, e^{j\frac{\omega}{c} \mathbf{r}_N \cdot \mathbf{e}_r} \right]$$

¹ Der Ausdruck „Direktivität“ beschreibt die Fähigkeit eines Beamformers Rauschen aus allen Richtungen zu unterdrücken, ohne das gewünschte Signal aus einer bestimmten Richtung zu beeinträchtigen [1].

Der Zeilenvektor \mathbf{w} beschreibt hierbei die komplexen Gewichte $[w_1 \dots w_N]$ der Mikrofonsignale. Sie beinhalten zum einen der Vektor der jeweiligen Phasenverschiebung \mathbf{d} für das Steering (s. Abb. 2.10), zum anderen eine Amplitudengewichtung \mathbf{a} (s. auch Gl. 2.11). H bedeutet hermitesch. Der Eingangssignalvektor \mathbf{x} stellt die Mikrofonsignale $[x_1 \dots x_N]^T$ dar. Die Amplitudengewichtung wird beim D&S-B auf den Wert $1/N$ gesetzt. Das Strukturbild des D&S-B kann wie in Abb. 2.11 dargestellt werden.

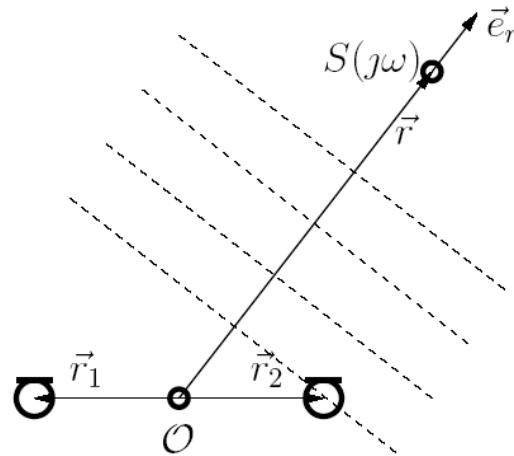


Abb. 2.10: Signalphasenverschiebung an den Mikrofonen, Schallquelle = $S(j\omega)$ [11]

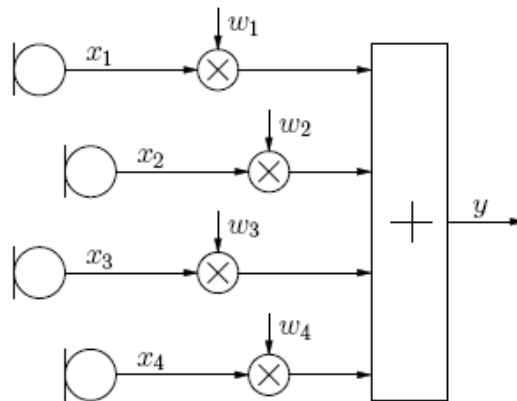


Abb. 2.11: Strukturbild eines Schmalband Beamformers [11]

Beim MV-B wird ausgehend von einer Optimierung in die Hauptrichtung (*Look Direction*) \mathbf{w} so gewählt, dass die Varianz, bzw. der Schätzer für die Gesamtleistung des Systems minimal wird. Signale aus der Look Direction sollen dabei unverändert übertragen werden.

Der Erwartungswert für die Ausgangsleistung berechnet sich zu [12]:

$$E\{|y|^2\} = E\{|\mathbf{w}^H \mathbf{x}|^2\} = E\{(\mathbf{w}^H \mathbf{x})(\mathbf{w}^H \mathbf{x})^H\} = \mathbf{w}^H E\{\mathbf{x}\mathbf{x}^H\} \mathbf{w} = \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \quad (2.17)$$

$$\text{mit } \mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^H\}$$

\mathbf{R}_{xx} ist die $N \times N$ Korrelationsmatrix des Eingangssignals. Die Optimierung erfolgt nun durch die Minimierung des Ausgangssignals mit der Nebenbedingung einer unverzerrten Wiedergabe des Signals aus der Wunschrichtung $\mathbf{d}(\Phi_0, \Theta_0)$ (Φ_0 =Azimutwinkel, Θ_0 =Elevationswinkel).

$$\min_{\mathbf{w}} \{ \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \} \quad \text{mit} \quad \mathbf{d}(\Phi_0, \Theta_0)^H \mathbf{w} = 1 \quad (2.18)$$

Dieser Optimierungsansatz ist auf eine Mehrzahl von Nebenbedingungen erweiterbar. Das bedeutet, es werden für verschiedene Steuervektoren in einer Matrix $\mathbf{L} = [\mathbf{d}_1^H \ \mathbf{d}_2^H \ \dots \ \mathbf{d}_L^H]$ unterschiedliche Bedingungen $\mathbf{b} = [1 \ 0 \ \dots \ 1]$ gefordert (z.B. Verstärkung 0 für eine andere Richtung). Die Forderung würde dann lauten: $\mathbf{L}\mathbf{w} = \mathbf{b}$. An dieser Stelle wird jedoch nur auf das eindimensionale Problem eingegangen. Für eine genaue Erläuterung der mehrdimensionalen Methode möchte ich auf [11] verweisen.

Der optimale Gewichtsvektor \mathbf{w}_{opt} ergibt sich aus (2.18) als Extremum. Die Berechnung erfolgt mittels Lagrange'scher Multiplikatoren.

$$\mathbf{w}_{opt} = \frac{\mathbf{R}_{xx}^{-1} \mathbf{d}(\Phi_0, \Theta_0)}{\mathbf{d}(\Phi_0, \Theta_0)^H \mathbf{R}_{xx}^{-1} \mathbf{d}(\Phi_0, \Theta_0)} \quad (2.19)$$

Wenn die Korrelationsmatrix nicht bekannt ist und nicht geschätzt werden kann, so lässt sich eine analytische Lösung für \mathbf{w}_{opt} finden, indem man eine bestimmte Anordnung von Störquellen im Raum voraussetzt. Hierzu wird die Korrelationsmatrix in einen Signal- und einen Störanteil aufgeteilt [12]:

$$\mathbf{R}_{xx} = \sigma_s^2 \mathbf{d}(\Phi_0, \Theta_0) \mathbf{d}(\Phi_0, \Theta_0)^H + \sigma_n^2 \mathbf{Q} \quad (2.20)$$

σ_s ist hierbei die Signalleistung, σ_n die Störleistung und \mathbf{Q} die Korrelationsmatrix des Störanteils. (2.20) in (2.18) eingesetzt ergibt das neue Optimierungsproblem, welches sich wie (2.18) mit dem Lagrange'schen Lösungsansatz ergibt:

$$\min_{\mathbf{w}} \{ \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \} = \sigma_s^2 + \sigma_n^2 \min_{\mathbf{w}} \{ \mathbf{w}^H \mathbf{Q} \mathbf{w} \} \quad \text{mit} \quad \mathbf{d}(\Phi_0, \Theta_0)^H \mathbf{w} = 1 \quad (2.21)$$

$$\Rightarrow \mathbf{w}_{opt} = \frac{\mathbf{Q}^{-1} \mathbf{d}(\Phi_0, \Theta_0)}{\mathbf{d}(\Phi_0, \Theta_0)^H \mathbf{Q}^{-1} \mathbf{d}(\Phi_0, \Theta_0)} \quad (2.22)$$

Die Matrix \mathbf{Q} ist für verschiedene Rauschquellenanordnungen analytisch bestimmbar. In dieser Arbeit wird nur die Lösung für ein sphärisch isotropes² Rauschfeld gezeigt. Weitere Lösungen sind u. a. in [1] zu finden. Die Korrelationsmatrix für ein diffuses Rauschfeld lautet [12]:

² Bei einem sphärisch isotropen oder auch diffusen Rauschfeld sind die Rauschquellen im Raum so auf einer Kugeloberfläche verteilt, dass aus jeder Richtung die gleich Energie zu messen ist.

$$\mathbf{Q}_{n,m} = \frac{1}{4\pi} \int_{\Omega} e^{j\frac{\omega}{c}(\mathbf{r}_m - \mathbf{r}_n) \cdot \mathbf{e}_r} d\Omega = \frac{\sin\left(\frac{\omega}{c} r_{ik}\right)}{\frac{\omega}{c} r_{ik}} \quad (2.23)$$

$\mathbf{r}_n, \mathbf{r}_m \rightarrow$ Ortsvektoren der Mikrofonpositionen

$r_{ik} \rightarrow$ Abstand zwischen zwei Mikrofonen

$\mathbf{e}_r \rightarrow$ Richtungsvektor zu $d\Omega$

(2.23) in (2.22) eingesetzt liefert nun die optimalen Gewichte für einen MV-B in einem sphärisch isotropen Rauschfeld. Abb. 2.12 lässt den Gewinn an Direktivität im Verhältnis zum D&S-B erkennen:

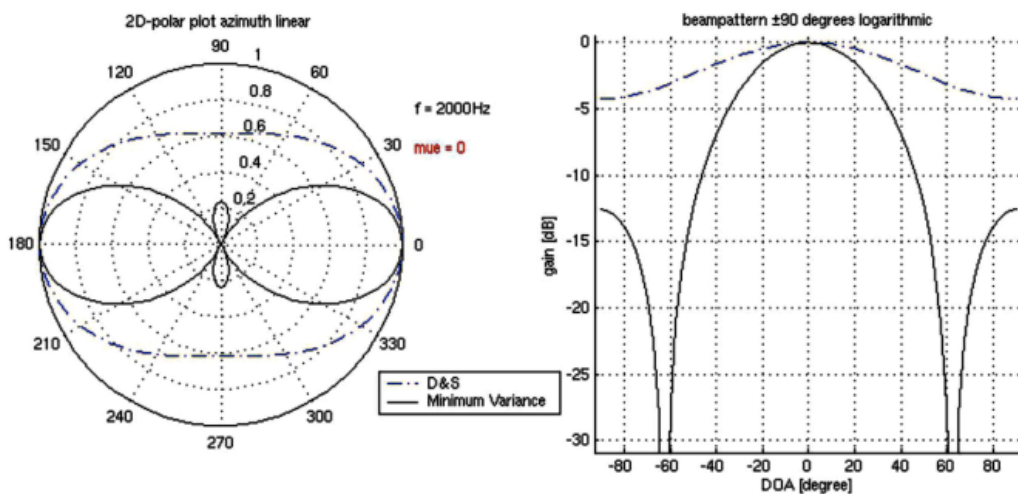


Abb. 2.12: Vergleich von MV-B und D&S-B (MATLAB)

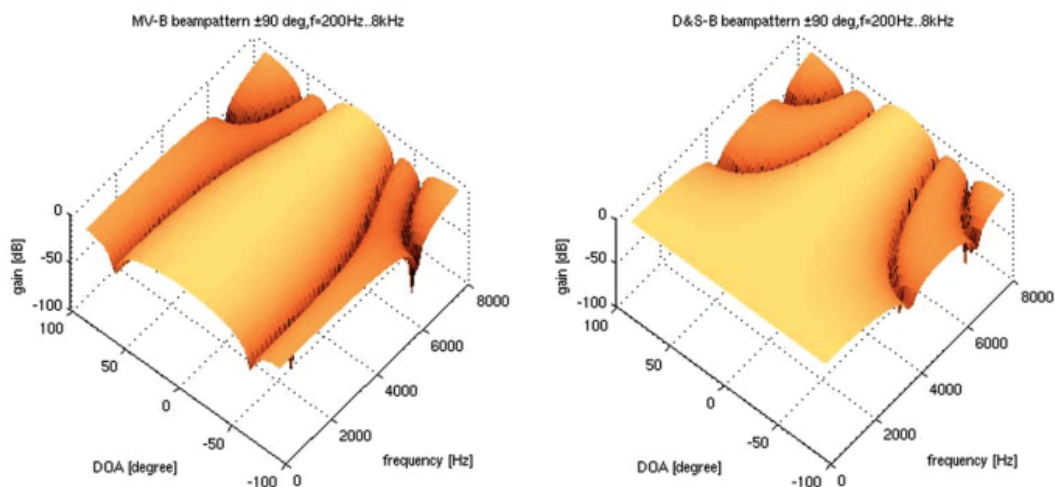


Abb. 2.13: Beampatterns von MV-B (li.) und D&S-B (re.) über die Frequenz (MATLAB)

2.3.2 Vorteile

Der größte Vorteil von MV-B ist der hohe Gewinn an Direktivität (vgl. Abb. 2.12). Durch diesen Gewinn liefert dieser Ansatz konstantere Directivity Patterns über einen großen Frequenzbereich. Während der D&S-B keine tiefen Frequenzen herausfiltern kann, dämpft der MV-B auch tiefe Frequenzen die nicht aus der Look Direction kommen gut ab (Abb. 2.13). Außerdem lässt Abb. 2.13 erkennen, dass sich der MV-B ab einer Frequenz, deren Wellenlänge kleiner als der doppelte Mikrofonabstand d ist (in diesem Beispiel etwa 4.5 kHz) wieder wie ein D&S-B verhält [1].

2.3.3 Nachteile

Eine praktische Umsetzung des Minimum Varianz Beamformers ist so gut wie unmöglich. Zum einen ist jedes Beampattern ein isolierter Schmalbandfall, d. h. es gibt für jede Frequenz einen eigenen Satz optimaler Amplitudengewichte, zum anderen funktioniert dieser Ansatz schon bei kleinsten Abweichungen vom Idealzustand, wie zum Beispiel bei Positionierungsungenauigkeiten der Sensoren, nicht mehr (vgl. [12]) und ist somit im Gegensatz zum D&S-B nicht robust³. Um die hohe Direktivität bei tiefen Frequenzen erreichen zu können, werden die Gewichte für die einzelnen Sensoren in diesem Frequenzbereich sehr hoch (s. Abb. 2.14). Dies hat gleichzeitig auch einen Anstieg von unkorrelierten Störungen, wie z. B. Sensorrauschen zur Folge. Störungen, die zwischen den einzelnen Sensoren keine Korrelation aufweisen, können von einem Beamformer nicht gefiltert werden [1]. Um diesen Problemen entgegenzuwirken, geht man auf eine Erweiterung des Minimum Varianz Ansatzes über, den Filter & Sum Beamformer.

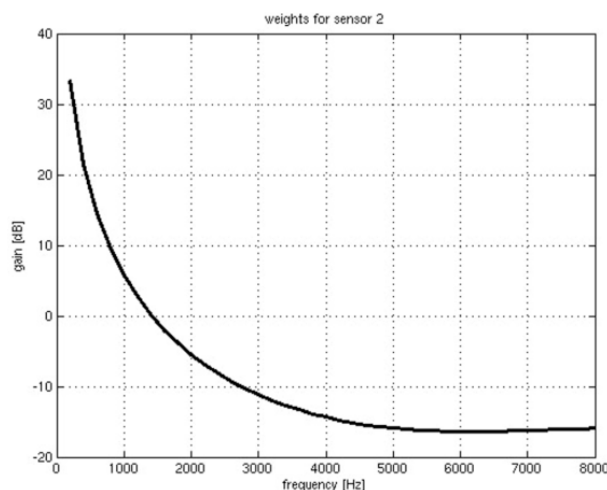


Abb. 2.14: Gewichte für einen Sensor im MV-B (MATLAB)

³ Die Robustheit eines Beamformers richtet sich nach seiner Unempfindlichkeit gegenüber zufälliger Ungenauigkeiten des Systems - wie zum Beispiel Positionierungs- oder Verstärkungsfehler der Sensoren, nicht genau die Schallquelle treffende Wunschrichtungen (*Direction of Arrival (DOA) – Mismatch, General Steering Vector Errors*) - und dadurch verzerrte Zielsignale (*Target Signals*).

2.4 Filter & Sum Beamformer

Um sich endgültig vom Schmalbandfall zu lösen, wird das in Abb. 2.11 vorgestellte Strukturbild nun durch einen Satz von Gewichten, einem FIR-Filter in jedem Kanal, erweitert (Abb. 2.15):

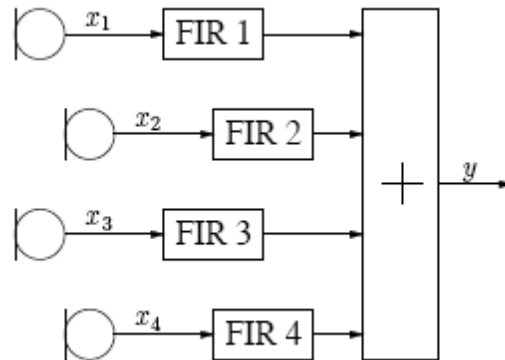


Abb. 2.15: Strukturbild eines einfachen Breitband Filter & Sum Beamformers [11]

Mit dieser Anordnung wird es möglich, verschiedene breitbandige Beamforming Ansätze und Erweiterungen umzusetzen. Es werden im folgenden Abschnitt die zwei zugrunde liegenden Prinzipien kurz erläutert.

2.4.1 Feststehende Filter

Mit feststehenden Filtern wird die Minimum Varianz Theorie fortgeführt. Man berechnet zunächst nach der in Kapitel 2.3 vorgestellten Methode die Gewichte für alle Frequenzen und erhält somit den Wunschfrequenzgang für jeden Sensor. Gewichte von gegenüberliegenden Sensoren sind konjugiert komplex. Diese Frequenzgänge können dann durch das *Least Squares* Verfahren in einem J Punkte FIR-Filter in jedem Signalpfad angenähert werden [11]. Der Satz an Gewichten im Beamformer entspricht dann in jedem Kanal genau der Impulsantwort des Filters. Da die Filtergewichte für tiefe Frequenzen sehr hohe Werte annehmen und im Grenzfalle bei 0 Hz gegen unendlich gehen, würden die benötigten FIR-Filter eine sehr hohe Dynamik bekommen und somit nur schwer bis gar nicht zu realisieren sein. Deswegen betrachtet man für die Optimierung nur einen Frequenzbereich zwischen einer unteren und oberen Grenzfrequenz. Unterhalb und oberhalb dieser Grenzfrequenzen wird das Array als D&S-B mit einer zusätzlichen Dämpfung berechnet. Als Beispiel dient hier die Berechnung eines linearen 4 Elemente Arrays mit einheitlichem Mikrofonabstand $d = 4\text{cm}$. Die Grenzfrequenzen liegen bei 200Hz und 6.8kHz, bei einer Abtastrate von 16kHz. Die Wunschrichtung ist 0° in Azimut und Elevation, senkrecht zur Achse des Arrays. Die Dämpfung der Bereiche außerhalb der Grenzfrequenzen wird bei -10 dB der Ausgangsleistung liegen. Die Filterlänge J beträgt 16 Punkte und die Anzahl der Punkte im anzunähernden Frequenzgang ist 265. Die Abbildungen Abb. 2.16 bis Abb. 2.18 verdeutlichen den Entstehungsprozess der FIR-Filterkoeffizienten.

Der Filter & Sum Beamformer verfügt genauso wenig über Robustheit, wie der schmalbandige MV-B, da er ja nur eine breitbandige Erweiterung ist. Es gibt allerdings eine Möglichkeit, die Unempfindlichkeit zu erhöhen, wenn man die angenommene Korrelationsmatrix für isotropes Rauschen regularisiert. Diese Maßnahme wird in Kapitel 4.1 genau beschrieben.

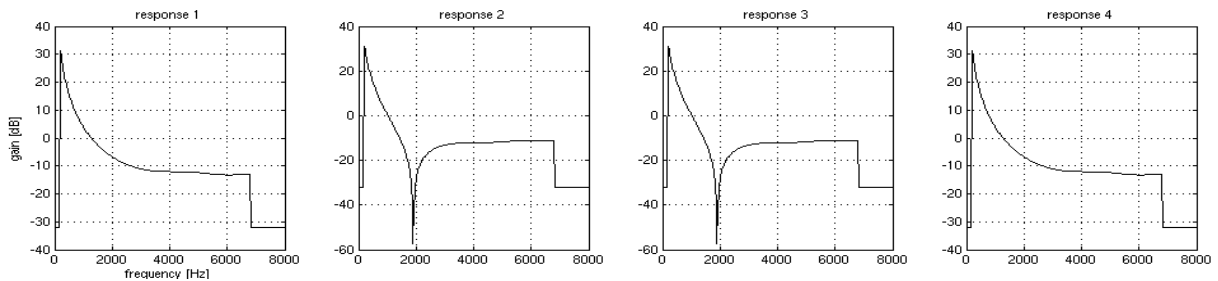


Abb. 2.16: berechnete Frequenzgänge für ein 4 Elemente Array (MATLAB)

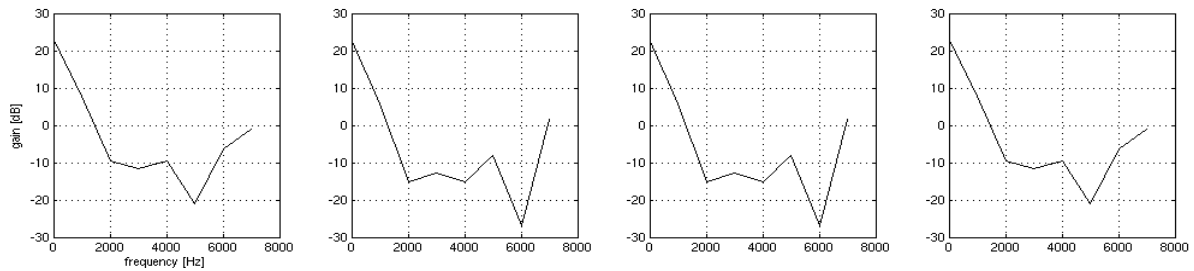


Abb. 2.17: mit Least Squares approx. Frequenzg. für ein 4 El. Array (MATLAB)

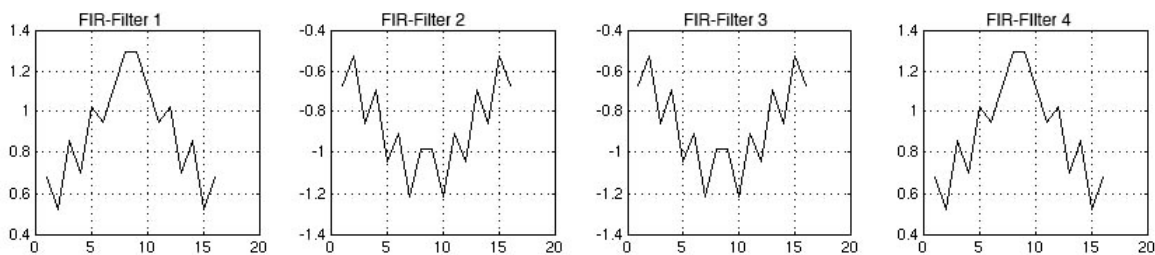


Abb. 2.18: FIR-Filterkoeffizienten für ein 4 Elemente Array (MATLAB)

2.4.2 Adaptive Filter

Für die vorangegangenen Optimierungsansätze wurde vorausgesetzt, dass die Korrelationsmatrix des Systems und somit die Beschaffenheit des Störfeldes bekannt ist. Es ist jedoch eher ein Ausnahmefall, dass ein angenommenes Störfeld wie zum Beispiel sphärisch isotropes Rauschen, die reale Umgebung ausreichend approximiert. Störfelder in realen Umgebungen sind zudem meist zeitvariant.

Adaptive Filter und Algorithmen sollen Abhilfe schaffen. Es wird versucht, unter der Annahme einer sich verhältnismäßig langsam ändernden (quasistationären) Umgebung, die aktuelle Korrelationsmatrix zu schätzen und auf dieser Schätzung basierend die Ergebnisse einem optimalen Ansatz anzunähern [11]. Zusätzlich werden adaptive Filter dazu genutzt, die Robustheit des Systems zu erhöhen. Je besser es gelingt, die in jedem Moment herrschenden akustischen Bedingungen mit einem adaptiven Mikrofonarray aufzulösen, desto robuster kann ein System gestaltet werden. Jene Bemühungen, die angestellt werden, um solch ein System zu entwickeln und zu realisieren, werden „*Robust Adaptive Beamforming*“ (RAB) genannt.

Die Arraystruktur eines adaptiven Breitband Beamformers unterscheidet sich im einfachsten Fall nur gering von der eines fixen Arrays (vgl. Abb. 2.15 mit Abb. 2.19). Die Filterkoeffizienten sind jetzt adaptiv und damit zeitvariant. Vor den FIR-Filtern wird zusätzlich in jedem Kanal eine variable Verzögerungsstufe eingebaut, um das Array-Steering von der Systemidentifikation zu trennen. Das hat den Vorteil, dass der komplexe Anteil von \mathbf{w} nun abgekoppelt wird und die Koeffizienten reell bleiben. Jeder Filter „sieht“ somit das Target Signal im Idealfall immer synchron. Störungen erreichen die Filter immer asynchron.

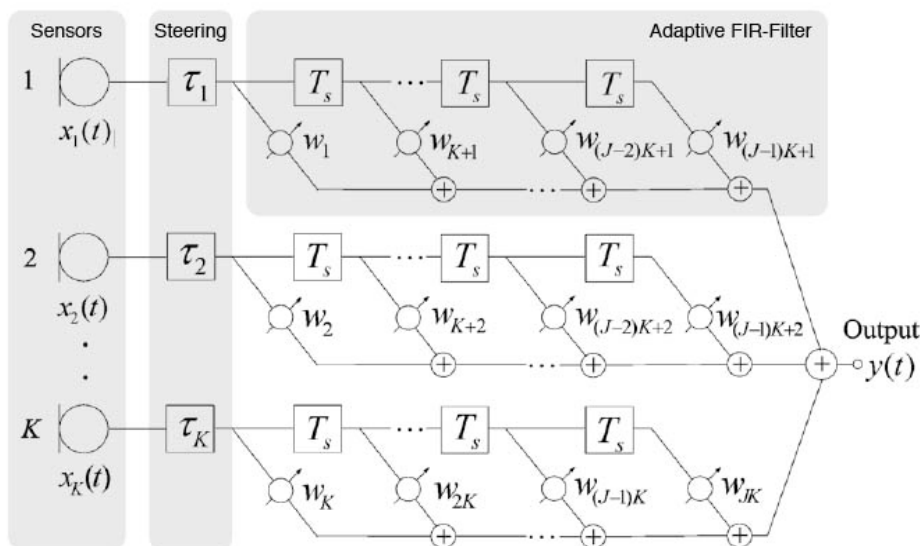


Abb. 2.19: Strukturbild eines einfachen adaptiven Beamformers [10]

Es existiert eine Vielzahl von adaptiven Beamforming Methoden und jeder dieser Ansätze besitzt seine eigenen mathematischen Theorien und Feinheiten. Deshalb wird die Mathematik an dieser Stelle nicht vertieft. Jedoch wird in Kapitel 4 ein der in Abb. 2.19 dargestellten Arraystruktur zugrunde liegender adaptiver Basisansatz und einige Varianten davon vorgestellt und ausführlich behandelt. Die Performance von Beamformern dieser Art wird nicht mehr anhand der klassischen Richtdiagramme bewertet. Sie werden als breitbandige Systeme behandelt, wobei als Bewertungskriterium die richtungsabhängige Ausgangsleistung sowie der „*Signal to Interference Ratio*“ (SIR) dient. Man könnte solche Systeme auch als räumliche Noisecanceller bezeichnen. Richtdiagramme für einzelne Frequenzen lassen sich oft gar nicht anschreiben, da die adaptiven Algorithmen für schmalbandige, hoch korrelierte Signale wie einen Sinus nicht mehr konvergieren.

3 Arraygeometrie

Die Anordnung der Sensoren in einem Array ist ein wichtiges Kriterium bei der Entwicklung eines Beamformers. Sie beeinflusst erheblich die Funktionalität des Systems. Durch die Geometrie werden die Grenzen und Möglichkeiten eines Beamformers klar definiert. Dieses Kapitel beschreibt die Verhaltensweisen von grundsätzlichen Anordnungsmöglichkeiten mit ihren Grenzen und führt im weiteren Verlauf zu einer Geometrie, die für Arraystrukturen mit kleiner Mikrofonanzahl unter bestimmten Anforderungen sehr gut geeignet ist.

3.1 Anforderungen an das Array

Neben dem Vergleich von verschiedenen robusten Beamforming Ansätzen ist es auch ein Teil dieser Diplomarbeit eine Arraygeometrie zu finden, die sich besonders gut für den Einsatz als Freisprech- und Konferenzmikrofon eignet. Hierfür ist eine kleine Bauweise unumgänglich. Sie soll es ermöglichen, das System unauffällig und ohne großen Aufwand in verschiedenste voneinander unabhängige Umgebungen, wie zum Beispiel in ein Fahrzeug oder einen Konferenzsaal, integrieren zu können. Damit das System auch mit den heute zur Verfügung stehenden Prozessorleistungen in Echtzeit realisierbar wird, darf in Anbetracht der Komplexität der verwendeten Algorithmen und der daraus entstehenden Latenzzeiten nur eine geringe Anzahl von Mikrofonen verwendet werden.

Ziel ist es also, mit möglichst wenigen Mikrofonen eine Geometrie zu finden, die für ein Steering gleichermaßen in Azimut und Elevation geeignet ist. Dabei sollte ein vorgegebener Bereich von $\pm 40^\circ$ um das Zentrum abgedeckt werden können. Das Array darf nur Signale aus der vorderen Halbebene aufnehmen. Die obere Grenzfrequenz soll über 6 kHz liegen.

3.2 Verschiedene Geometrien

Zum Entwurf der Arraygeometrie wurde ein Simulationstool in MATLAB programmiert (s. Kapitel 8), mit dem es möglich ist, das richtungsbezogene Übertragungsverhalten eines Delay & Sum Beamformers mit beliebiger 3-dimensionaler Sensorstruktur zu simulieren. Als Simulationsergebnis erhält man einen 3-dimensionalen Plot (Polardiagramm) der Richtcharakteristik des Beamformers. Es wird von Sensoren mit Kugelcharakteristik ausgegangen.

Die Simulationen von D&S-B Richtplots ist ausreichend, weil weitere Optimierungen tendenziell das gleiche Verhalten bezüglich der prinzipiellen Form des Beams aufweisen. Es verbessert sich lediglich die Direktivität bei tiefen Frequenzen.

3.2.1 Simulationen

Im Folgenden werden zur Verdeutlichung des Richtverhaltens von bestimmten Anordnungen einige Simulationsergebnisse von einfachen Geometrien dargestellt:

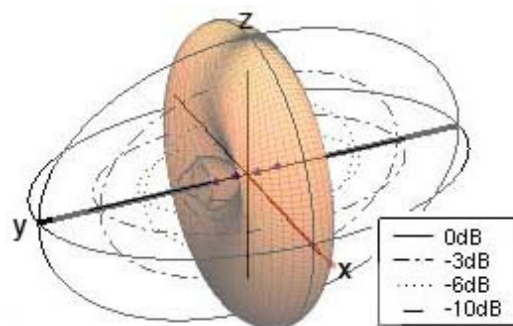


Abb. 3.1: 3D Richtplot eines lin. 4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB)

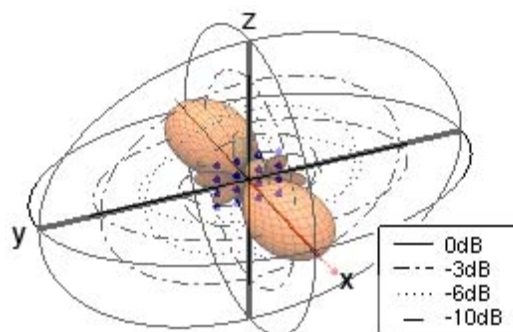


Abb. 3.2: 3D Richtplot eines lin. 4x4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB)

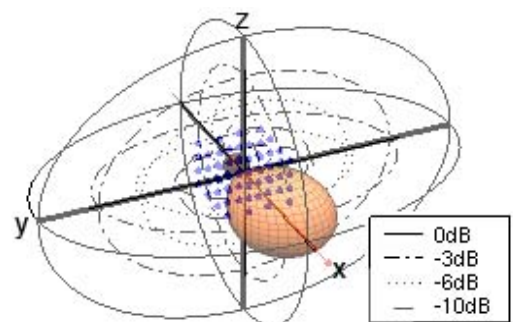


Abb. 3.3: 3D Richtplot eines lin. 4x4x4 El. Arrays, $f = 4\text{kHz}$, $d = 4\text{cm}$ (MATLAB)

In den Abbildungen Abb. 3.1 bis Abb. 3.3 ist ersichtlich, dass mit zunehmender Dimension der Arraygeometrie die Richtplots immer stärker in Bezugsrichtung (hier die positive x-Achse) gerichtet sind. Das lineare 4 Elemente Array hat eine um die y-Achse rotationssymmetrische Richtcharakteristik, bei 4x4 Elementen entsteht eine an der y-z-Ebene gespiegelte Richtcharakteristik, die um die x-Achse rotationssymmetrisch ist. Im 3-dimensionalen Fall mit 4x4x4 Elementen ist der Plot nur noch bezüglich der x-Achse symmetrisch. Bei Realisierung eines Arrays, welches seinen Hauptbeam nur in eine Richtung ausbilden soll, muss eine 3-dimensionale Elementanordnung verwendet werden. Bei 2-dimensionalen Anordnungen müssen in diesem Fall die Mikrofone gegenüber von hinten einfallenden Schallwellen abgeschattet werden. Dies kann sowohl durch den Einsatz von Absorbern, als auch der Verwendung gerichteter Kapseln erreicht werden. Nachdem man für 3-dimensionale Geometrien sehr viel mehr Sensoren braucht als für 2-dimensionale, ist es in diesem Fall besser sich auf letzteres zu beschränken, oder zumindest das Hauptaugenmerk auf 2 Dimensionen zu werfen.

Die nachfolgenden Simulationen zeigen einige Versuche zur Einschätzung des Steeringverhaltens bei 2-dimensionalen Geometrien mit geringer Mikrofonanzahl. Die Überlegungen bei der Auswahl der Anordnungen basieren auf den Erkenntnissen aus [9]. Es hat sich herausgestellt, dass eine sternförmige Anordnung mit einer ungeraden Anzahl von Armen die besten Ergebnisse bezüglich des Steeringverhaltens und dem Verhältnis von Mikrofonanzahl und geometrischen Ausmaßen zu oberer Grenzfrequenz liefert. Da aber die Arrays aus [9] zu viele Sensoren und zu große Aperturen aufweisen, muss getestet werden, ob die Ergebnisse auch auf kleinere Arrays mit weniger Mikrofonen übertragbar sind. Auch ist es denkbar eine Anordnung zu finden, deren Beam sich schwerpunktmäßig mit einer schmaleren Keulbreite als Keulenhöhe ausbildet, wenn man dadurch besseres Azimutsteering gewinnt. Bei Tischarrays z. B. ist die untere Halbebene durch den Tisch abgeschattet. Außerdem hält sich ein Sprecher vor einem Tisch nur in der oberen Halbebene auf. Es gibt in diesem Fall keine Veranlassung, eine Geometrie zu finden, die gut nach unten gesteuert werden kann. Wenn die Keulenhöhe also etwas größer wäre, so können damit Höhenunterschiede des Target Signals ausgeglichen werden und es ist möglich sich in der Mikrofonverteilung auf ein gutes Azimutsteering zu konzentrieren. Die Keulenausbildung in die hintere Halbebene ist zu vernachlässigen, da in der Realisation Nierenmikrofone verwendet werden.

Als erstes wurde versucht eine lineare Anordnung zu verwenden, die durch 2 vertikal darüber und darunter angeordnete Mikrofone gestützt wird, um die Keulenhöhe einzuschränken. Zusätzlich wurde getestet, wie sich Krümmungen der y-z-Ebene auf die Keulformung und das Steering auswirken. Abb. 3.4 zeigt die Simulationsergebnisse. Die Plots auf der linken Seite zeigen eine lineare Geometrie mit 7 Sensoren. 5 Mikrofone mit einem Abstand von 4cm auf der y-Achse und 2 mit einem Abstand von 6cm zum Mittelpunkt. Auf der rechten Seite der Abbildung stehen die Ergebnisse der gleichen Anordnung, nur dass die y-z-Ebene in einen 120° Kreisbogen gebogen wird. Es ist zu erkennen, dass die Einschränkung der Keulenhöhe durch 2 zusätzliche Mikrofone funktioniert. Das Azimut-Steering ist über den geforderten Bereich von $\pm 40^\circ$ brauchbar. Die Krümmung der y-z-Ebene bewirkt aber nur eine Abschwächung der hinteren Keule. Das Steeringverhalten verbessert sich dadurch nicht. Beide Anordnungen können nicht für Elevationssteering verwendet werden, da sie bei Verlassen der x-y-Ebene die Hauptkeule an dieser Ebene spiegeln und somit unbrauchbar werden.

Es zeigt sich, dass solche Anordnungen für die geforderte Anwendung nicht geeignet sind und dass leichte Krümmungen nur die Dämpfung der rückwärtigen Keule des Beamformers verbessern. Zumindest sollte das Array auch in einem beschränkten Bereich die Möglichkeit zu Elevationssteering bieten. Ähnliche Ergebnisse haben sich auch schon in [9] mit der 21 Element Kreuzanordnung gezeigt.

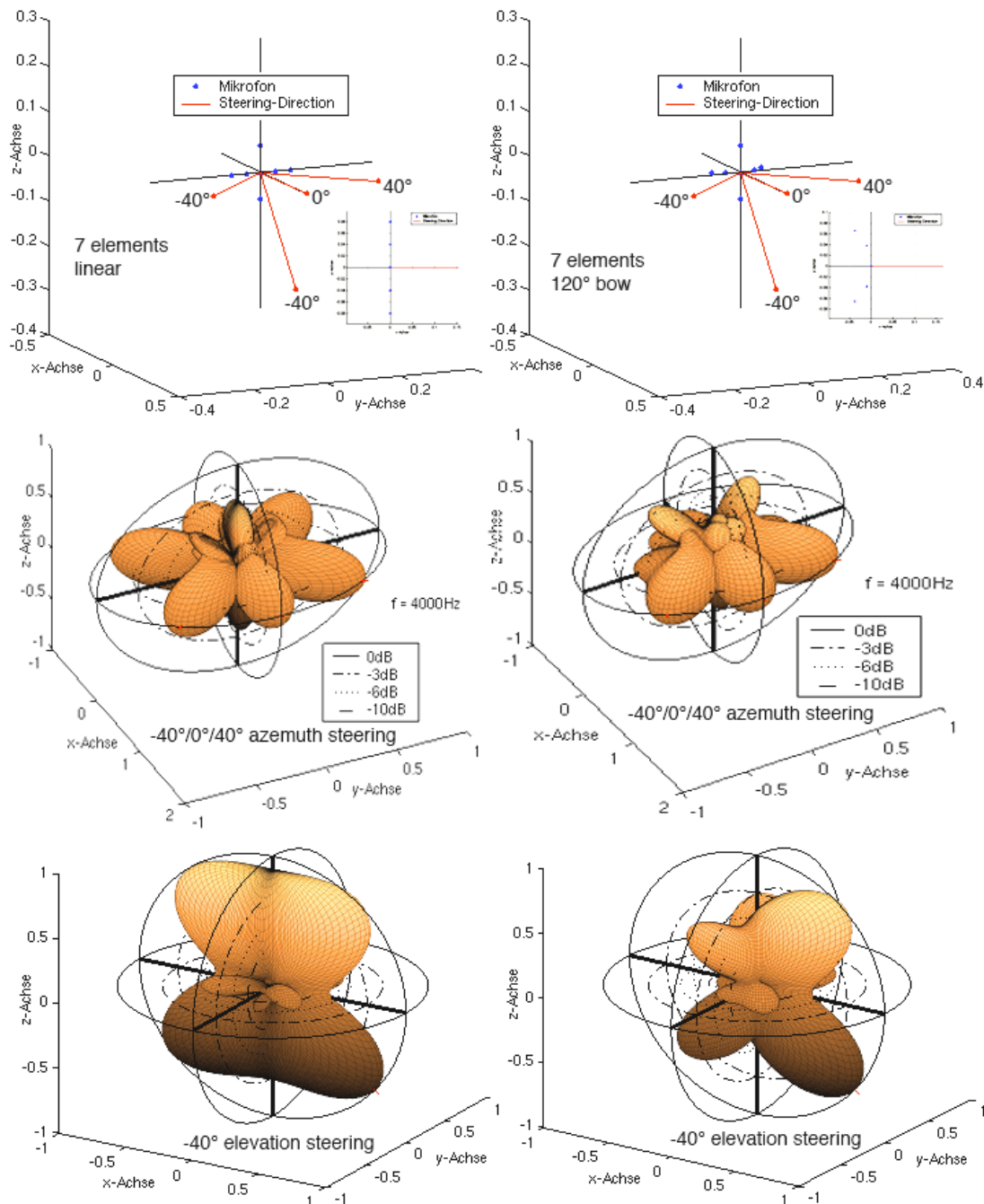


Abb. 3.4: Vergleich von Steingeigenschaften linear (li.) vs. Bogen (re.) (MATLAB)

3.3 Das Mikrofonarray

Nachdem auch weitere Versuche mit Anordnungen dieser Art zu keinen befriedigenden Ergebnissen geführt haben, wird nun versucht eine Geometrie zu finden, welche die Sternanordnung aus [9] am besten approximiert. Dafür werden 5 Mikrofone jeweils in einem Winkel von 72° zum nächsten Sensor mit einem Radius von 4cm um ein Mittelpunktmikrofon angeordnet.

3.3.1 Simulationen

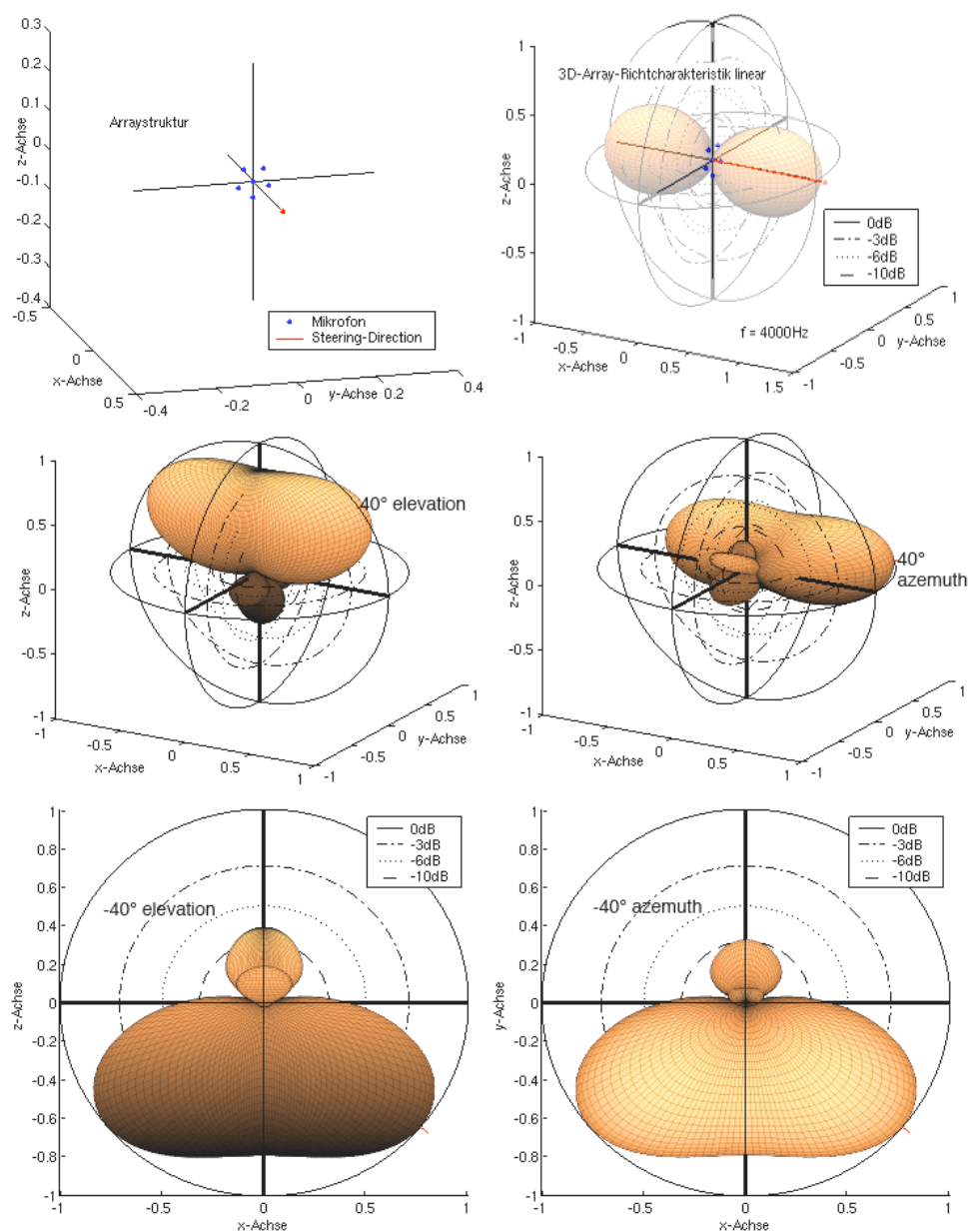


Abb. 3.5: Steeringverhalten des 6 Elemente Sterns (MATLAB)

Das Array ist rotationssymmetrisch um die x-Achse und besitzt in der reduzierten Anordnung dieselben Asymmetrien im „*Embedded Spacing*⁴“, wie das 25 Element Sternarray aus [9]. In Abb. 3.5 lässt sich gut erkennen, dass das Verhalten für Anordnungen wie sie in [9] beschrieben wurden auch auf kleinere Arrays zu übertragen ist. Das simulierte Array weist für diese Größe sehr gutes Steeringverhalten auf. Die Keule gleichermaßen in Azimut- wie in Elevationsrichtung im vorgegebenen Bereich variiert werden. Natürlich beginnt die Richtwirkung aufgrund der Größe erst bei einer für Sprachsignale hohen unteren Grenzfrequenz (s. Abb. 3.6). Diese Begebenheit soll aber durch signaltheoretische Verfahren optimiert werden. Aus Abb. 3.6 ist ersichtlich, dass im Gegensatz zu anderen vergleichbaren Strukturen bis 8kHz, noch kein räumliches Aliasing auftritt.

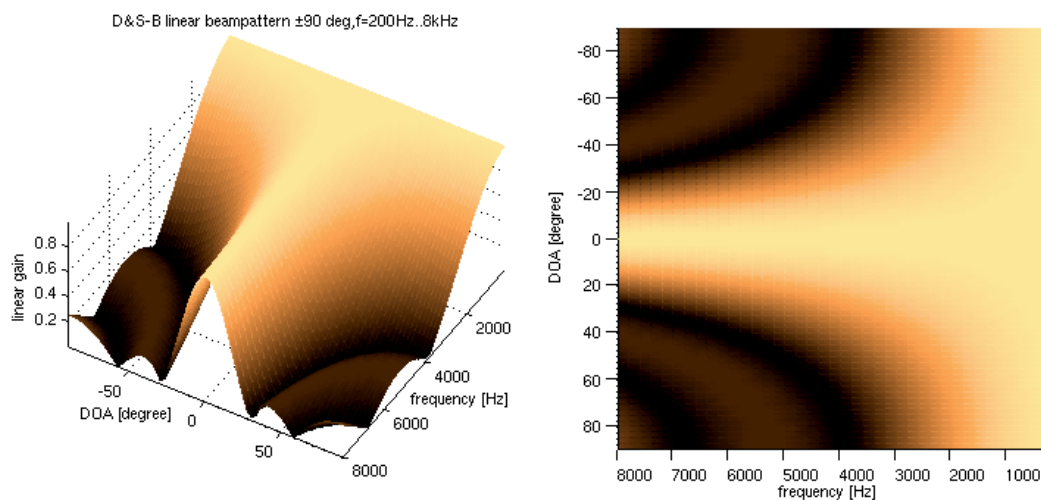


Abb. 3.6: 6 Elemente Stern, Beampattern vs. Frequenz, $200\text{Hz} \leq f \leq 8\text{kHz}$ (MATLAB)

3.3.2 Bau des Arrays

Für den Bau des Arrays wurden 6 Nierenkapseln vom Typ I15UBL47/2 mit Vorverstärker von AKG⁵ zur Verfügung gestellt (s. Abb. 3.7). Die Trägerkonstruktion wurde aus Plexiglas gefertigt, wobei die Mikrofone in der Rohrung mittels Heißkleber fixiert wurden. Die Verstärker sind ebenfalls mit Heißkleber auf einer Plastikschiene angebracht, die gleichzeitig den Standfuß des Tischarrays bildet. Kapseln und Verstärker wurden mit Kabeln verlötet, Verstärkerausgänge mit XLR-Kabeln versehen (s. Abb. 3.8, links). Die Geometrie ist wie in den Simulationen dimensioniert. Leichte Abweichungen von den Idealmaßen wurden bewusst nicht vermieden, um bei der akustischen Vermessung des Arrays die Robustheit zu testen (s. Abb. 3.8, rechts).

⁴ *Embedded Spacing* definiert die Mikrofonanordnung in der Projektionsebene der Schalleinfallrichtung. Je weniger Mikrofone sich in der Projektionsebene überlagern und je unterschiedlicher die projizierten Abstände der Sensoren, desto besser ist das Embedded Spacing. Ein gutes Embedded Spacing ist z. B. dann gegeben, wenn sich die Mikrofonabstände zum Zentrum der Apertur hin verengen.

⁵ AKG Acoustics, Lehmböckgasse 21-25, A-1230 Wien – www.akg.com

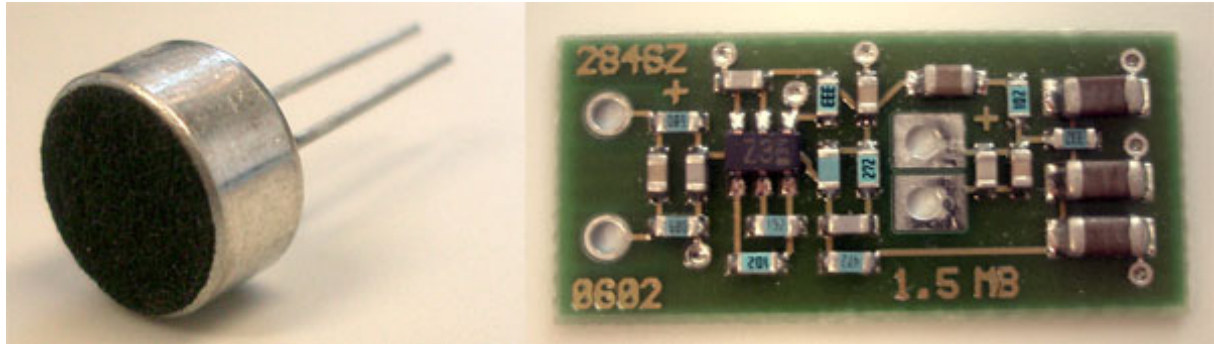


Abb. 3.7: verwendete Mikrofonkapsel (li), Vorverstärker (re)

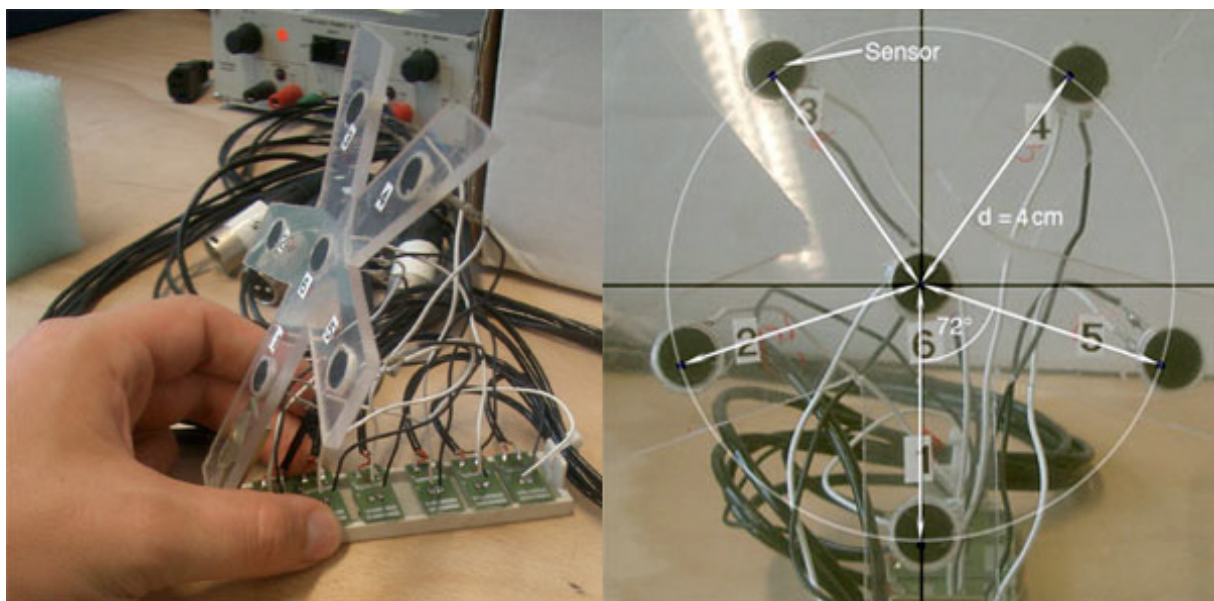


Abb. 3.8: Bild von Array (li), Geometrie (re)

4 Ausgewählte Algorithmen

In diesem Kapitel werden verschiedene aus der aktuellen Literatur ausgewählte Beamforming Algorithmen mathematisch erläutert. Diese werden mit MATLAB simuliert und schließlich in der Programmierumgebung Pure Data⁶ implementiert, um dort jeweils eine Echtzeitrealisierung der Ansätze zu erhalten. Die Simulationen sollen die prinzipiellen Möglichkeiten der Optimierung, die jeweiligen Ansätze zur Realisierung von Breitbandbeamformern veranschaulichen. Die Echtzeitrealisierungen werden sowohl mittels akustischer Vermessung, als auch subjektiv in einer typischen Anwendungssituation getestet. Anschließend erfolgt ein Vergleich mit den Ergebnissen aus den Simulationen. Jeder Ansatz wird bezüglich seiner Robustheit evaluiert. Für einen der behandelten adaptiven Beamformer wird eine Verbesserung vorgestellt.

4.1 Fixed Filter & Sum Beamformer

4.1.1 Theorie

Die prinzipielle Theorie des Filter & Sum Beamformers (F&S-B) wurde bereits in Kapitel 2.4 ausführlich vorgestellt. Ausgehend von einem Störfeld wird für jede Frequenz über eine analytisch bestimmte Korrelationsmatrix ein Satz von Gewichten über den Minimum Varianz Ansatz für jedes Mikrofon bestimmt. Der daraus resultierende Frequenzgang wird über die Least Squares Methode mittels eines J-Punkte FIR-Filters angenähert. Da die Filtergewichte mit tiefer werdenden Frequenzen gegen unendlich gehen und sich das Array ab einer bestimmten Frequenz wie ein D&S-B verhält, werden die optimalen Frequenzgänge nur in einem Bereich zwischen einer oberen und unteren Grenzfrequenz berechnet. Unterhalb der unteren und oberhalb der oberen Grenzfrequenz wird der Frequenzgang des D&S-B gedämpft beibehalten. Wie schon erwähnt ist das grundlegende Problem der Methode, dass sie eine breitbandige Version des Minimum Varianz Beamformers darstellt. Deswegen ist sie nicht robust. Selbst in der Simulation bei idealen Bedingungen liefert diese Methode keinen robusten Ansatz (s. Abb. 4.1). Die Frequenzgänge der FIR-Filter bilden eine zu ungenaue Näherung an den echten Minimum Varianz Fall. Es gibt aber eine Möglichkeit die Robustheit des F&S-B abhängig von einem Parameter zu erhöhen.

⁶ Pure Data ist eine Open Source Programmierumgebung zur Echtzeit-Signalverarbeitung von Audiosignalen [14].

Geht man von einem Störfeld aus, das nur in sich unkorrelierte Störungen beinhaltet, so ergibt sich als Korrelationsmatrix \mathbf{R}_{xx} die Einheitsmatrix $\mathbf{I}=\mathbf{I}^{-1}$, da jedes Signal nur mit sich selbst korreliert ist. Die Hauptdiagonale von \mathbf{R}_{xx} ergibt die Punkte für die Autokorrelation des jeweiligen Sensors. Dieser Punkt ist genau an der Stelle, an der die Autokorrelation eines Sensors berechnet wird (Hauptdiagonale der Korrelationsmatrix). Unkorrelierte Störungen sind zum Beispiel das Eigenrauschen der Mikrofone. Berechnet man nun nach Gleichungen (2.18) und (2.19) die optimalen Gewichte für unkorrelierte Störungen, so entsteht mit $|\mathbf{d}|^2 = N$ genau der D&S-B [11].

$$\mathbf{w}_{opt} = \frac{\mathbf{I}\mathbf{d}(\Phi_0, \Theta_0)}{\mathbf{d}(\Phi_0, \Theta_0)^H \mathbf{I}\mathbf{d}(\Phi_0, \Theta_0)} = \frac{1}{N} \mathbf{d}(\Phi_0, \Theta_0) \quad (4.1)$$

Wird zu der Korrelationsmatrix für ein angenommenes Störfeld ein Anteil $\mu\mathbf{I}$ von unkorrelierten Störungen addiert, so wird die Matrix regularisiert. Das bewirkt eine Verschiebung der optimalen Gewichte hin zum D&S-B. Dadurch nähert sich der optimale Frequenzgang an den des D&S-B und gleichzeitig auch an die Robustheit desselben. Gleichung (2.22) wird somit zu:

$$\mathbf{w}_{opt} = \frac{(\mathbf{Q} + \mu\mathbf{I})^{-1} \mathbf{d}(\Phi_0, \Theta_0)}{\mathbf{d}(\Phi_0, \Theta_0)^H (\mathbf{Q} + \mu\mathbf{I})^{-1} \mathbf{d}(\Phi_0, \Theta_0)} \quad (4.2)$$

Man erhält somit für $\mu=0$ den optimalen Minimum Varianz Beamformer und für $\mu \rightarrow \infty$ den D&S-B.

Diese Maßnahme entspricht einer Beschränkung der Verstärkung von unkorrelierten Störungen („White Noise Gain“ [1]), oder einer Erhöhung des Gewinnes gegenüber von unkorrelierten Störungen, da durch die Regularisierung der Korrelationsmatrix die hohen Filterkoeffizienten im tiefen Frequenzbereich abgeschwächt werden und somit das Eigenrauschen (unkorrelierte Störungen) der Sensoren eingedämmt wird [1].

4.1.2 Simulationen

Die folgenden MATLAB Simulationen zeigen das Verhalten des verwendeten Stern-Arrays unter Beschaltung eines F&S-B. Die Richtplots und Filterkoeffizienten bei verschiedenen Regularisierungsfaktoren sind in Abb. 4.2 dargestellt. Dabei wird mit einer Abtastfrequenz von 16kHz und einer Look Direction von 0° Azimut und Elevation simuliert. Die Filterlänge beträgt 16 Punkte. Das Eingangssignal ist gaussverteilter weißes Rauschen und wird von 200Hz bis 7.4kHz bandbegrenzt. Es wird nacheinander ein aus jeder Richtung einfallendes Signal simuliert und die Ausgangsleistung des Systems gemessen. Zusätzlich wird die Nieren-Richtcharakteristik der verwendeten Sensoren berücksichtigt. Dafür wird das Ausgangssignal in jeder Richtung mit der richtungsabhängigen Formel $(0.5+0.5\cos(\Phi))$ für Nierenmikrofone multipliziert (Φ =Azimutwinkel). Zur Vergleichsmöglichkeit ist auch das Beampattern des D&S-B auf diesem Array aufgeführt (fette Linie in Abb. 4.2).

Abb. 4.1 zeigt den Beamformer mit den optimalen Filtergewichten und Abb. 4.3 das Verhalten bei Verstärkungs- und Positionierungsfehlern der Sensoren für verschiedene Regularisierungsfaktoren im Vergleich zur idealen Anordnung und zum D&S-B. Dabei wird für einen $\pm 3\text{dB}$ Verstärkungsfehler das Eingangssignal mit entsprechenden zufälligen Faktoren multipliziert. Für den Positionierungsfehler werden die Koordinaten der Arraygeometrie einfach jeweils in x- und y-Richtung um einen Bereich von $\pm 5\text{mm}$ verschoben. Die Fehler sind rein zufällig gewählt. Sie sollen eine realistische, und gleichzeitig grenzwertige Größenordnung für Fehler dieser Art repräsentieren. Beide Abbildungen sind unter den selben Rahmenbedingungen wie Abb. 4.2 entstanden.

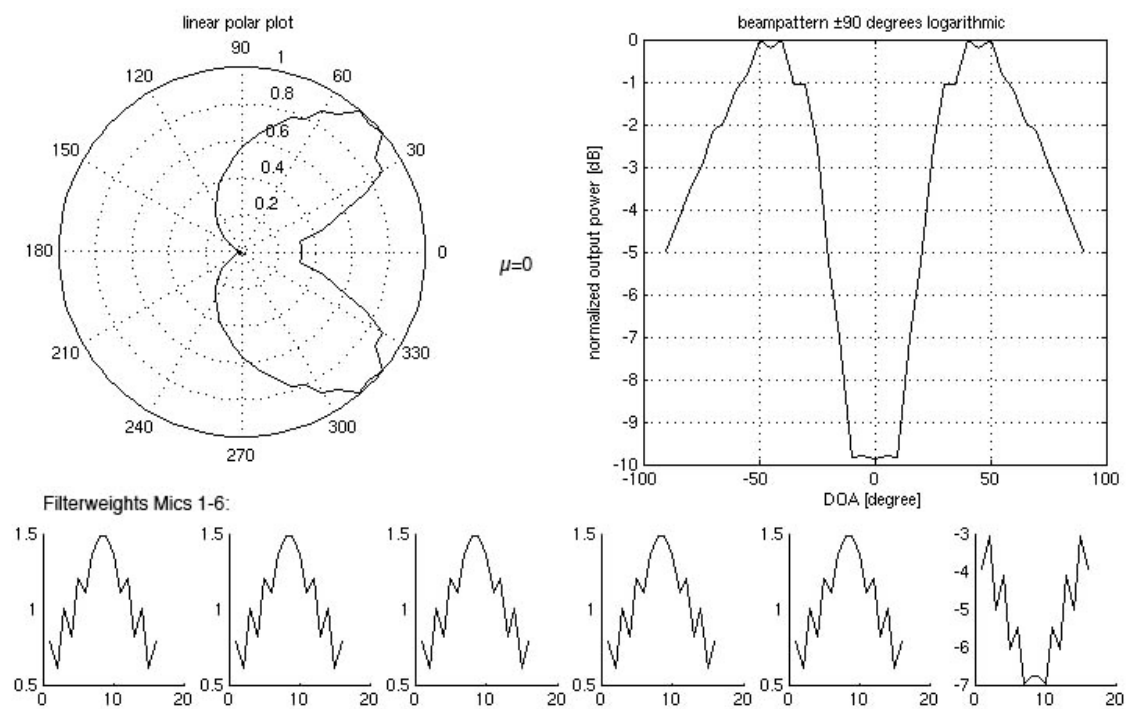


Abb. 4.1: Richtplot für opt. Filtergewichte, $\mu=0$ (MATLAB)

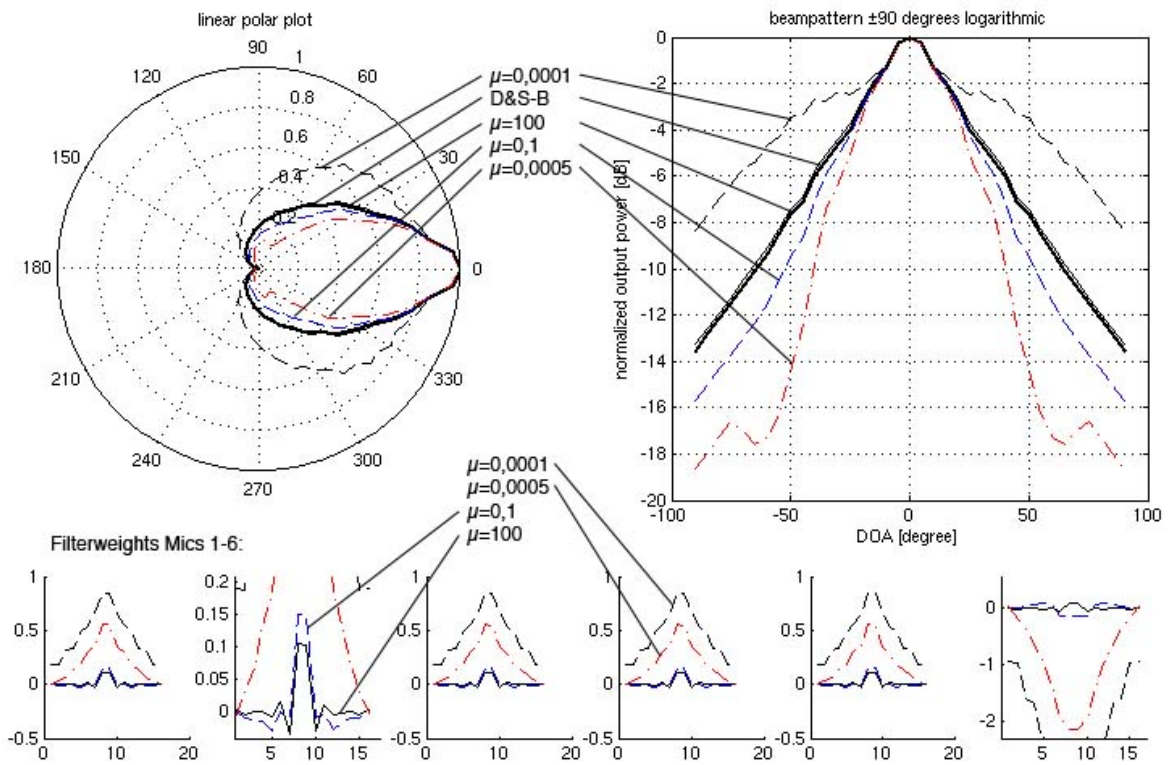


Abb. 4.2: Richtplots für $\mu=0,0001$, $\mu=0,0005$, $\mu=0,1$, $\mu=100$ (MATLAB)

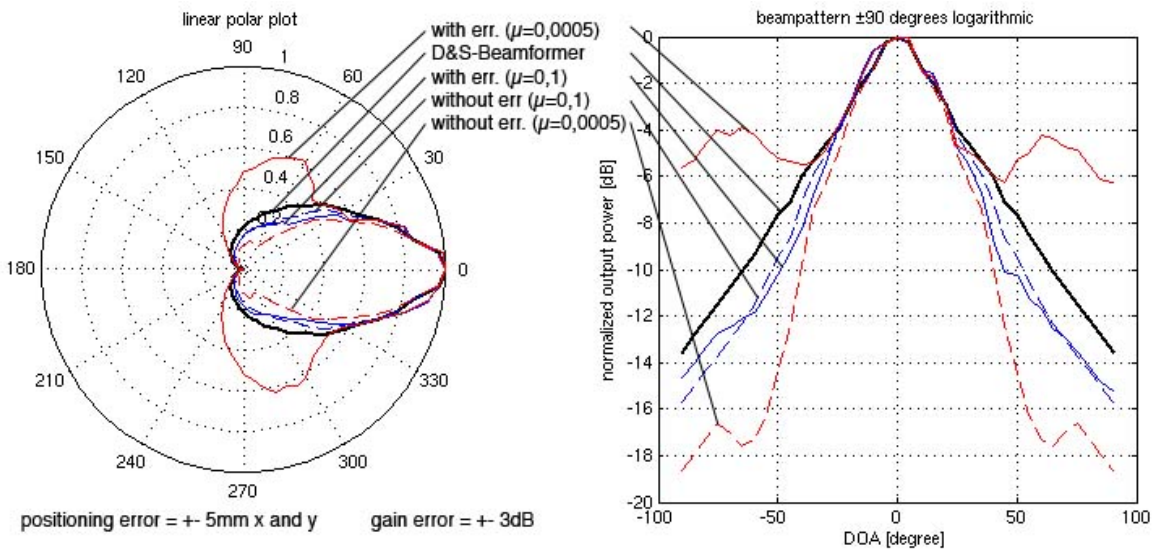


Abb. 4.3: Richtplots F&S-B bei Verstärkungs- und Positionierungsfehlern (MATLAB)

4.1.3 Auswertung und Vergleich der Simulationen

In Abb. 4.1 erkennt man, dass die Approximation des Frequenzganges mit FIR Filtern nach dem Least Squares Verfahren nicht mehr ausreicht und somit die Methode im Extremfall nicht funktioniert. Anstatt dass Signale aus seitlicher Richtung abgeschwächt werden und Schall aus der Hauptrichtung unverändert übertragen wird, klappt das Beampattern um und das Richtverhalten wird umgekehrt. Ohne Regularisierung der Korrelationsmatrix kann man zu keinem verwendbaren Filtersatz gelangen.

Die Diagramme in Abb. 4.2 zeigen, dass mit kleiner werdendem Regularisierungsfaktor μ die Keulbreite abnimmt und somit die Dämpfung von Signalen aus Richtungen die nicht der Hauptrichtung entsprechen höher wird. Gleichzeitig erhöhen sich aber auch die Beträge der Filterkoeffizienten. Somit wird das System aufgrund der hohen Dynamik immer instabiler [11]. Wird μ zu klein, so bewegt sich das Beampattern wieder in Richtung des Richtplots bei $\mu=0$, da die Genauigkeit der FIR-Filter nicht mehr ausreicht um das System richtig abzubilden. (im Diagramm an der äußersten Kurve für $\mu=0,0001$ zu erkennen). Mit steigender Regularisierung μ nähert sich der F&S-B immer mehr an den D&S-B an.

Abb. 4.3 verdeutlicht das Verhalten bei Positionierungs- und Verstärkungsfehlern der Sensoren. Je niedriger der Regularisierungsfaktor wird, desto mehr Stördämpfung geht verloren. Das Beampattern verliert seine eigentliche Richtcharakteristik und wird instabil. Erst im Bereich einer Regularisierung von $\mu=0,1$ ist kein signifikanter Unterschied zwischen fehlerbehaftetem und idealem Array zu erkennen. Dieses Verhalten ist nachvollziehbar, da die Filter für eine optimale Arraygeometrie mit einer einheitlichen Sensorverstärkung entworfen wurden. Je höher allerdings die Regularisierung wird, desto mehr Einfluss erhält der Beamformer vom sehr robusten D&S Ansatz. Deshalb machen sich im höher regularisierten Fall Ungenauigkeiten nicht so drastisch bemerkbar.

4.1.4 Realisierung in PD und akustische Vermessung

Der F&S-B wurde in PD auf die oben beschriebene Weise realisiert. Der Patch erlaubt es verschiedene Regularisierungsfaktoren auszuwählen. Vermessen wurde der Beamformer mittels der in Kapitel 5 beschriebenen Messmethode Nr.1, bei einem Regularisierungsfaktor von $\mu=0,005$ (Abb. 4.4). Als Vergleichsmöglichkeit ist noch der gemessene Richtplot des D&S-B im Diagramm abgebildet.

Wie auch schon in den Simulationen beobachtet, liefert der F&S-B eine höhere Dämpfung als der D&S-B. Bei weißem Rauschen ist die Dämpfung höher als bei rosa Rauschen. Man kann eine stärkere Keulenausbildung nach hinten betrachten, als in den Simulationen. Das ist damit zu erklären, dass sich die verwendeten Mikrofonkapseln als Supernieren herausgestellt haben (vgl. Kapitel 5), in den Simulationen aber mit Nierencharakteristik gerechnet wurde.

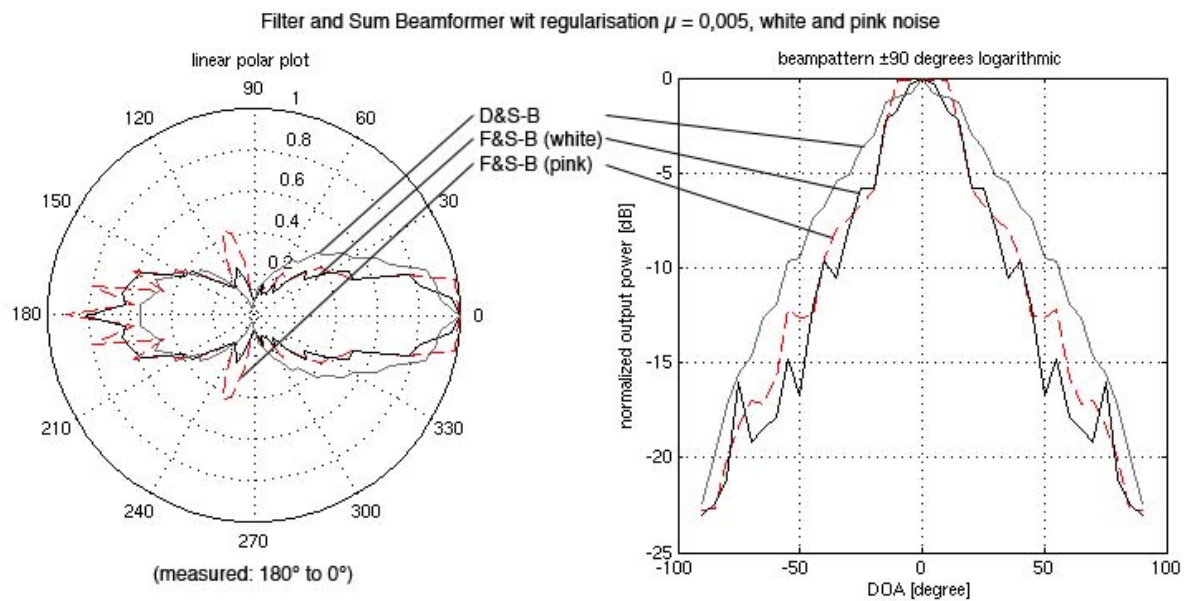


Abb. 4.4: gemessene Ausgangsleistung des F&S-B mit weißem und pinkem Rauschen

4.1.5 Evaluierung

Prinzipiell kann man diesen Ansatz ab einem bestimmten Regularisierungsfaktor durchaus als eine robuste und realisierbare Methode betrachten. Nur hat sich in den Simulationen sowie in den Messungen der Echtzeitanwendung gezeigt, dass für eine stabile Variante des F&S-B der Regularisierungsfaktor so hoch sein muss, dass es subjektiv keinen bemerkenswerten Gewinn an Stördämpfung mehr gibt. Mit weißem Rauschen als Testsignal dämpft der F&S-B bei einer Regularisierung von $\mu = 0,005$ zwar bis zu 5dB mehr als der D&S-B, im subjektiven Hörtest ist dieser Gewinn aber fast nicht auszumachen. Das bedeutet, man beansprucht die Rechenleistung von 6 Filtern, erhält jedoch keine wesentliche Verbesserung des Ergebnisses im Vergleich zu einem D&S-B ohne zusätzliche Filter. Deshalb wird in weiterer Folge eine ebenso robuste Lösung mit der Hilfe von adaptiven Filtern gesucht, die aber eine höhere Dämpfung von Signalen außerhalb der Look Direction aufweist.

4.2 Frost Beamformer

4.2.1 Theorie

Der Frost Beamformer ist die adaptive Weiterführung des F&S-B. Er wurde 1972 von O. L. Frost entwickelt und veröffentlicht. Auch bei diesem Algorithmus wird versucht, die Energie des Ausgangssignals zu minimieren, unter der Nebenbedingung einer unverzerrten Wiedergabe aus der Wunschrichtung. Diese Bedingung wird als so genannte „*Hard Constraint*“ bezeichnet.

Die Herleitung des Algorithmus erfolgt nach [10], [11] und [13], da das originale Paper von Frost nicht mehr in der IEEE Datenbank zu finden ist. Abb. 4.5 oben zeigt das Strukturbild des Frost Beamformers. Die Verzögerungsstufen τ_1 bis τ_K sollen eine Synchronisation des Target Signals bei unterschiedlicher Ausrichtung der Hauptkeule bewirken („*Prestearing*“). Diese Struktur lässt sich auch als äquivalentes System mit einem einzigen Ersatzprozessor darstellen (Abb. 4.5 unten). Der Ersatzprozessor hat als Eingang das Target Signal. Sein Ausgang wird genau dann dem Anteil des Target Signals des Systemausgangs entsprechen, wenn die Filtergewichte f_1 bis f_J der Summe der einzelnen Gewichte aus dem Frost Prozessor entsprechen [13]. Die Ersatzgewichte f_1 bis f_J werden zum Vektor \mathbf{F} zusammengefasst. \mathbf{F} wird als „Frost-Beschränkung“ („*Frost Constraint*“) bezeichnet. \mathbf{F} kann so gewählt werden, dass das Target Signal, während es den Beamformer passiert, einer linearen Filterung mit einer bestimmten Frequenzantwort unterzogen wird (z.B. Tiefpass). Wenn keine Filterung erwünscht ist, so können alle Gewichte bis auf eines auf 0 gesetzt werden. Das übrige Gewicht wird auf 1 gesetzt [13]. Die Filter des Frost Beamformers können sich beliebig adaptieren, solange die Summe der Gewichte jeder Filterstufe [1..j..J] dem entsprechenden Ersatzgewicht f_j entspricht.

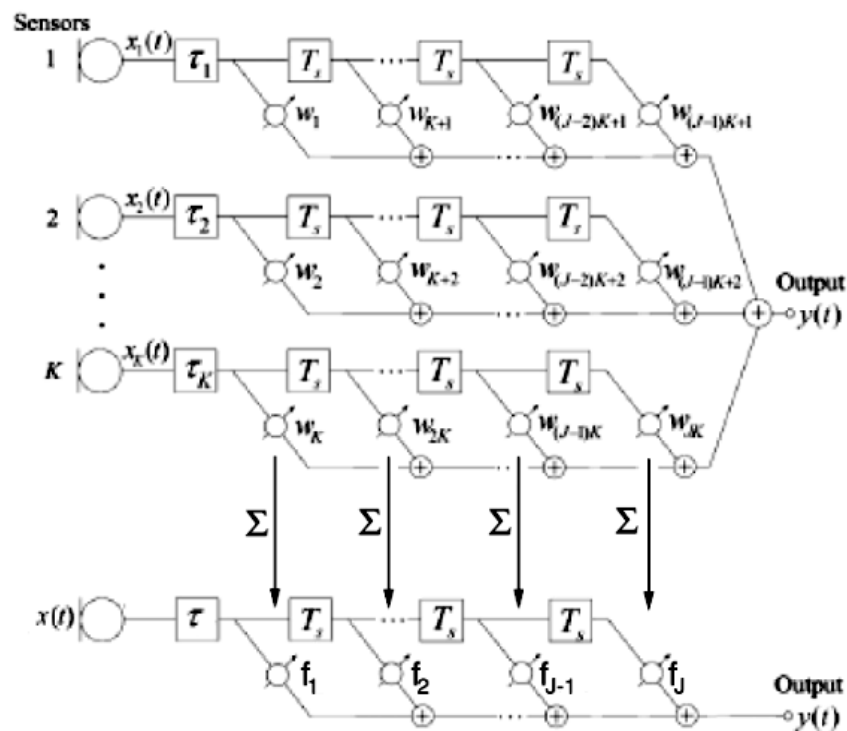


Abb. 4.5 oben: Strukturbild des Frost Beamformers, unten: Ersatzprozessor

Um die Filter im Beamformer zu adaptieren, hat Frost einen LMS Algorithmus verwendet, bei dem der mittlere quadratische Fehler, unter Berücksichtigung von allgemeinen linearen Beschränkungen („*Constraints*“), minimiert wird. Unter der Annahme von K Sensoren mit jeweils J Filterkoeffizienten und bereits an die Look Direction angepasste Steering Delays - sodass das empfangene Target Signal synchron an den Filtereingängen anliegt - lassen sich die Filtergewichte und das Eingangssignal („*Filterload*“) als Vektoren in folgender Form anschreiben (vgl. hierzu mit Abb. 4.5) [10]:

$$\mathbf{X}(n) = [x_1(n) \dots x_K(n) \quad x_1(n-1) \dots x_K(n-1) \quad x_1(n-J+1) \dots x_K(n-J+1)]^T \quad (4.3)$$

$$\mathbf{W} = [w_1 \dots w_{KJ}]^T \quad \rightarrow \quad \text{Dimension: jeweils } KJ \times 1 \quad (4.4)$$

Das Ausgangssignal $y(n)$ lässt sich durch das Produkt

$$y(n) = \mathbf{W}^T \mathbf{X}(n) \quad (4.5)$$

darstellen. Der Vektor \mathbf{F} der Ersatzgewichte hat folgende Form:

$$\mathbf{F} = [f_1 \dots f_J \dots f_J]^T \quad \rightarrow \quad \text{Dimension: } J \times 1 \quad (4.6)$$

Um die Nebenbedingung mathematisch formulieren zu können wird eine Constraint-Matrix \mathbf{C}_0 eingeführt. Die transponierte Constraint-Matrix \mathbf{C}_0^T multipliziert mit dem Gewichtsvektor \mathbf{W} ergibt genau den Constraint-Vektor \mathbf{F} . Daraus ergibt sich folgendes Optimierungsproblem für den Frost Beamformer:

$$\min_{\mathbf{W}} P = \min \mathbf{W}^T \mathbf{R}_{xx} \mathbf{W} \quad \text{mit} \quad \mathbf{C}_0^T \mathbf{W} = \mathbf{F} \quad (4.7)$$

$$\mathbf{C}_0 = [c_1 \dots c_j \dots c_J] \quad (4.8)$$

$$c_j = \begin{bmatrix} 0 \dots 0 & 1 \dots 1 & 0 \dots 0 \\ (j-1)K & K & (j-j)K \end{bmatrix}^T \quad \text{mit } j = 1, \dots, J \quad \rightarrow \quad \text{Dimension: } KJ \times J$$

Die Korrelationsmatrix $\mathbf{R}_{xx} = E\{\mathbf{X}(n)\mathbf{X}(n)^T\}$ wird vorerst als bekannt vorausgesetzt. Die Lösung lässt sich wieder über die Methode der Lagrange'schen Multiplikatoren gewinnen.

$$L(\mathbf{W}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{W}^T \mathbf{R}_{xx} \mathbf{W} + \boldsymbol{\lambda}^T (\mathbf{C}_0^T \mathbf{W} - \mathbf{F}) \quad (4.9)$$

Den Gradienten von (4.9) in Richtung \mathbf{W} erhält man über (4.10). Dieser wird 0 gesetzt und liefert somit den Vektor für die optimalen Filtergewichte. Diese Gewichte müssen nun natürlich die Nebenbedingung aus (4.7) erfüllen. Daraus ergeben sich die Lagrange'schen Multiplikatoren in Gleichung (4.12). (4.12) in (4.11) eingesetzt ergibt schließlich die Lösung des Problems (4.7), dargestellt in Gleichung (4.13) [11].

$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{R}_{xx} \mathbf{W} + \mathbf{C}_0^T \boldsymbol{\lambda} \stackrel{!}{=} 0 \quad (4.10)$$

$$\Rightarrow \mathbf{W}_{opt} = -\mathbf{R}_{xx}^{-1} \mathbf{C}_0^T \boldsymbol{\lambda} \quad (4.11)$$

$$\mathbf{C}_0^T \mathbf{W}_{opt} \stackrel{!}{=} \mathbf{F} \quad \xrightarrow{\text{Lagrange}} \quad \boldsymbol{\lambda} = -[\mathbf{C}_0^T \mathbf{R}_{xx}^{-1} \mathbf{C}_0]^{-1} \mathbf{F} \quad (4.12)$$

$$\Rightarrow \mathbf{W}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{C}_0^T [\mathbf{C}_0^T \mathbf{R}_{xx}^{-1} \mathbf{C}_0]^{-1} \mathbf{F} \quad (4.13)$$

Der Schritt zum adaptiven Filter erfolgt nun so, dass man den Gewichtsvektor \mathbf{W} bei einem Anfangswert starten lässt, der die Nebenbedingung erfüllt. Beispielsweise kann als Startwert für die Korrelationsmatrix die Einheitsmatrix angenommen werden. Somit ergibt sich für \mathbf{W} aus (4.13) dann $\mathbf{W}(0) = \mathbf{C}_0 [\mathbf{C}_0^T \mathbf{C}_0]^{-1} \mathbf{F}$. Mit jeder Iteration wird nun der Gewichtsvektor über einen Adaptionsfaktor α (Schrittweite) ein Stück weiter in die negative Richtung des Gradienten verändert. Je kleiner α gewählt wird, desto genauer kann der Algorithmus konvergieren. Jedoch verlängert sich dabei auch die Konvergenzzeit.

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \alpha \frac{\partial L(\mathbf{W}, \boldsymbol{\lambda})}{\partial \mathbf{W}} = \mathbf{W}(n) - \alpha [\mathbf{R}_{xx} \mathbf{W}(n) + \mathbf{C}_0 \boldsymbol{\lambda}(n)] \quad (4.14)$$

Wird (4.14) in die Nebenbedingung eingesetzt und nach $\boldsymbol{\lambda}$ aufgelöst, erhält man die Lagrange'schen Multiplikatoren für (4.14).

$$\boldsymbol{\lambda} = \frac{1}{\alpha} (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{C}_0^T \mathbf{W}(n) - (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{C}_0^T \mathbf{R}_{xx} \mathbf{W}(n) - \frac{1}{\alpha} (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{F} \quad (4.15)$$

(4.15) eingesetzt in (4.14) liefert die iterative Gleichung zur Bestimmung der neuen Filtergewichte [11]:

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \alpha [\mathbf{I} - \mathbf{C}_0 (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{C}_0^T] \mathbf{R}_{xx} \mathbf{W}(n) + \mathbf{C}_0 (\mathbf{C}_0^T \mathbf{C}_0)^{-1} [\mathbf{F} - \mathbf{C}_0^T \mathbf{W}(n)] \quad (4.16)$$

$$\text{mit } \mathbf{C}_0 (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{F} \equiv \mathbf{W}_c \quad \text{Dimension: KJ x 1} \quad (4.17)$$

$$\text{und } \mathbf{I} - \mathbf{C}_0 (\mathbf{C}_0^T \mathbf{C}_0)^{-1} \mathbf{C}_0^T \equiv \mathbf{P} \quad \text{Dimension: KJ x KJ} \quad (4.18)$$

$$\Rightarrow \mathbf{W}(n+1) = \mathbf{P} [\mathbf{W}(n) - \alpha \mathbf{R}_{xx} \mathbf{W}(n)] + \mathbf{W}_c \quad (4.19)$$

In Gleichung (4.19) bleibt somit lediglich die Korrelationsmatrix $\mathbf{R}_{xx} = E\{\mathbf{X}(n)\mathbf{X}(n)^T\}$ als Unbekannte. Aus diesem Grund wird ein Schätzwert $\hat{\mathbf{R}}_{xx} = \mathbf{X}(n)\mathbf{X}(n)^T$ für die Korrelationsmatrix eingeführt (Dimension: KJ x KJ), der sich aus dem Produkt der aktuellen Filterloads ergibt [11]. Wird dieser Schätzwert in (4.19) eingesetzt, so erhält man mit Gleichung (4.3) die folgende Beziehung zur Berechnung der Filtergewichte:

$$\mathbf{W}(n+1) = \mathbf{P} [\mathbf{W}(n) - \alpha \mathbf{X}(n)y(n)] + \mathbf{W}_c \quad (4.20)$$

Mit dieser mathematischen Beziehung ist es möglich den Frost Prozessor in einer Simulation zu testen und das Verhalten dieses Beamformers zu beschreiben.

4.2.2 Simulationen

Obige Beziehungen wurden in MATLAB implementiert. Folgende Diagramme (Abb. 4.6) zeigen einen Vergleich der Richtplots von D&S-B, F&S-B und Frost Beamformer. Als Arraygeometrie wurde das 6 Element Array aus Kapitel 3.3 verwendet. Die Abtastrate beträgt 16kHz, die Wunschrichtung des Beamformers beträgt 0° für Azimut und Elevation.

Alle adaptiven Filter haben eine Länge von $J=16$ Punkten und die Schrittweite α beträgt 0,1. Als Eingangssignal wurde gaussverteiltes weißes Rauschen verwendet, dessen Zeitdauer größer als die Konvergenzzeit der Filter ist. Um repräsentative Testdaten zu erzeugen, wurde dasselbe Monosignal an jeden Eingang des Prozessors gelegt und die jeweiligen Kanäle abhängig von der Einfallrichtung entsprechend zueinander verzögert. Die Beampattern in allen Simulationen erhält man, indem der Beamformer aus jeder Richtung zur Konvergenz gebracht und anschließend die Ausgangsleistung ermittelt wird.

Die Frost – Constraint F bildet einen FIR - Bandpass ab, mit den Grenzfrequenzen 200Hz und 7,4 kHz. Der Bandpass wurde mit dem MATLAB Filter Design Tool berechnet. Die Nieren-Richtcharakteristik der Mikrofone wird mit berücksichtigt.

In Abb. 4.7 ist der Richtplot des Frost Beamformers unter dem Einfluss derselben Positionierungs- und Verstärkungsfehler wie in Kapitel 4.1.2 dargestellt. Der Unterschied bei Positionierungsfehlern ist hier allerdings, dass es bei 0° Steering keinen Positionierungsfehler geben kann. Veränderungen der Mikrofonkoordinaten entsprechen einfach einer Änderung der Arraygeometrie und nicht einem Fehler. Um einen richtigen Fehler zu erzeugen, muss die Hauptkeule des Arrays mittels eines, in der idealen Geometrie erzeugten, Steeringvektors in eine bestimmte Richtung gesteuert werden und anschließend der Richtplot mit der fehlerbehafteten Geometrie berechnet werden. Es entsteht so ein Offset, der eine nicht vollständige Überlagerung der Signale in der Steering Direction bewirkt.

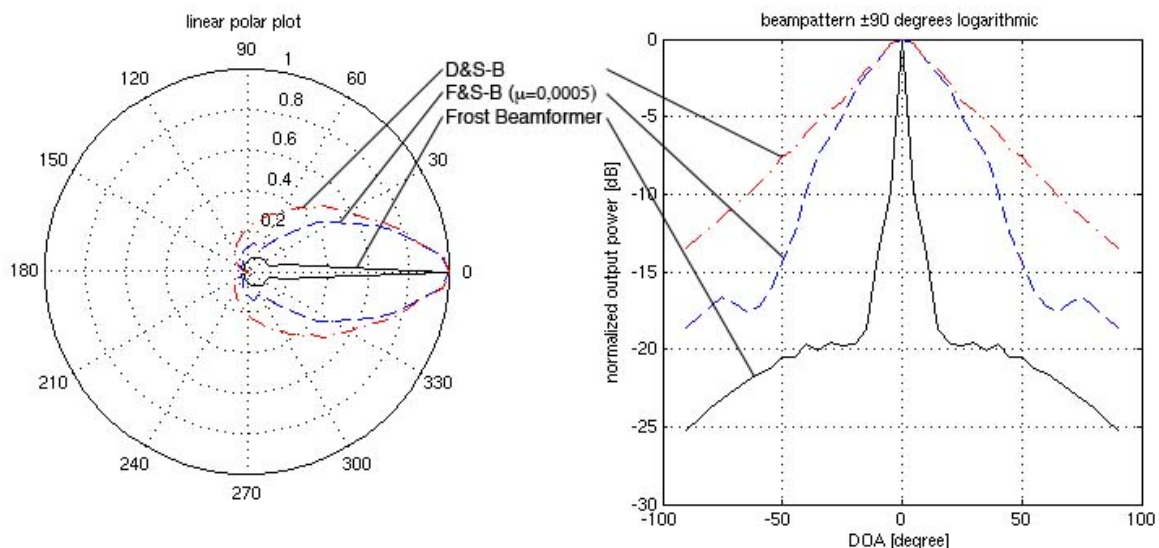


Abb. 4.6: Vergleich: Richtplots von D&S-B, F&S-B und Frost Beamformer (MATLAB)

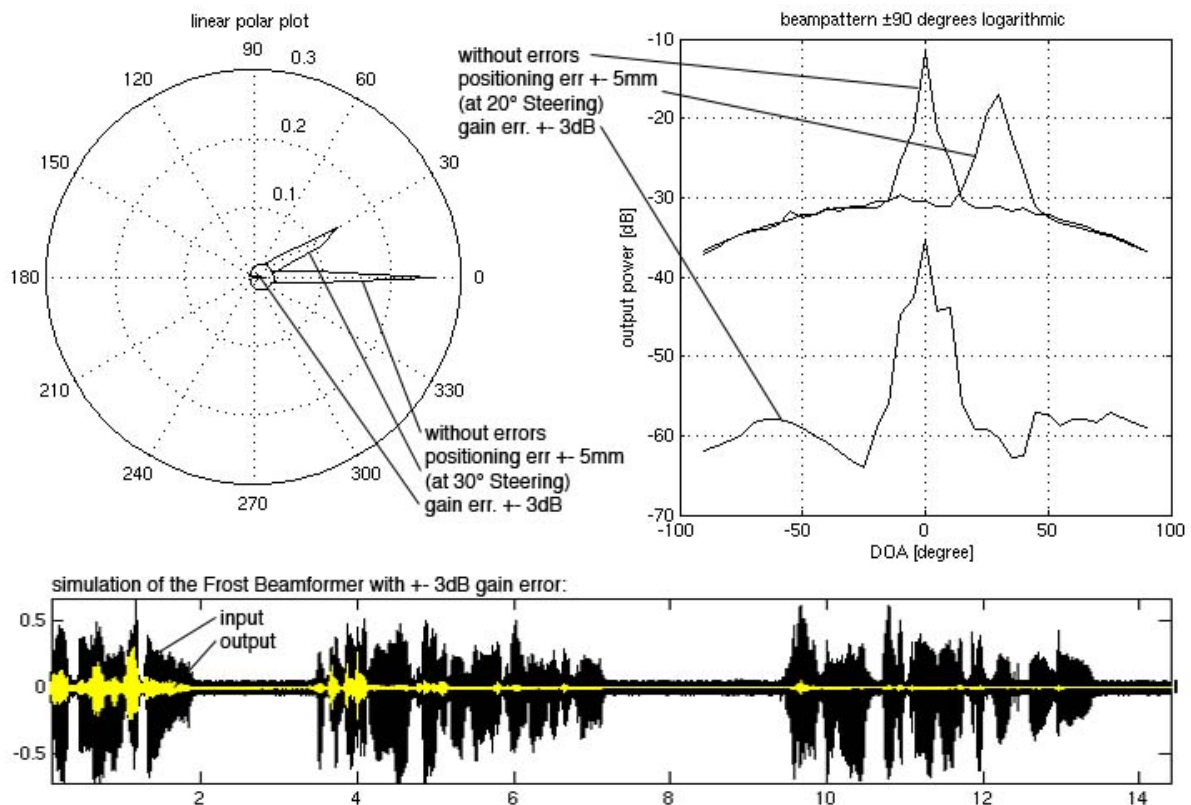


Abb. 4.7: Frost Beamformer mit Positionierungs- und Verstärkungsfehlern (MATLAB)

4.2.3 Auswertung und Vergleich der Simulationen

Der Frost Beamformer weist eine viel höhere Dämpfung von Signalen auf, die nicht aus der Wunschrichtung kommen, als die bisher vorgestellten Beamformer. Gleichzeitig zieht diese Tatsache aber auch mit sich, dass nahezu keine Toleranz gegenüber Richtungsschwankungen des Target Signals herrscht. Durch die extrem schmale Hauptkeule werden nur Signale, die zu 100% synchron am Prozessor ankommen unverzerrt wiedergegeben. In einer halligen Umgebung zum Beispiel können Anteile vom Target Signal als Reflexionen asynchron am Prozessor ankommen und eliminiert werden. Da diese mit dem Wunschsignal korreliert sind, ist es unvermeidbar, dass auch Teile des davon gedämpft werden. Wie in Abb. 4.7 ersichtlich, ist keine Toleranz gegenüber von Verstärkungsfehlern an den Sensoren gegeben. Der Algorithmus minimiert bei Verstärkungsfehlern das gesamte Ausgangssignal. Um diesen Effekt sichtbar machen zu können, sind die Beampattern in Abb. 4.7 nicht, wie in den anderen Plots, auf 0dB normalisiert. Unterhalb der Beampattern befindet sich eine Darstellung des Eingangssignals eines Sensors und des Ausgangssignals aus einer Simulation des Beamformers bei einem Verstärkungsfehler. Sprache ist dabei als Target, weißes Rauschen als Störer aus -20° verwendet worden. Die Abbildung zeigt, dass das Ausgangssignal inklusive Target minimiert wird. Dagegen verhält sich der Beamformer robuster gegenüber Positionierungsfehlern. Hier wird lediglich die Hauptkeule aufgrund der Streuung des Steering Vektors kleiner. Insgesamt zeigen die Simulationsergebnisse aber, dass der reine Frost Beamformer nicht robust ist und somit für reale Anwendungen nicht umsetzbar.

4.2.4 Realisierung in PD

Der Beamformer wurde in PD realisiert, es hat sich aber, wie schon in den Simulationen erwartet, gezeigt, dass diese Methode nicht robust ist. Durch informelle Hörversuche ist erkennbar, dass es zu extremer Target Signal Auslöschung und zu äußerst geringer Rauschunterdrückung kommt. Man kann sogar eher eine Verschlechterung des Signal-Rausch Verhältnisses feststellen. Die Hörversuche wurden mit nur wenigen Versuchspersonen durchgeführt. Dabei wurde aus Target Richtung in das Array gesprochen und aus Interference Richtung weißes Rauschen zugespielt. Die Versuchspersonen mussten die klangliche Verbesserung, vor allem hinsichtlich der SNR im direkten Vergleich (umschaltbar) mit einem D&S-B beurteilen.

4.2.5 Evaluierung

Durch die fehlende Robustheit kann dieser Algorithmus nur als Basis für weitere Entwicklungen gesehen werden. Bei der Verwendung mit einem realen Array wird immer das Problem der Wunschsingalunterdrückung eintreten. Das Wunschsingal kann im Realfall die Bedingung nicht erfüllen, exakt aus Target Richtung zu kommen und Verstärkungsfehler der Sensoren sind unvermeidbar. Außerdem ist bei diesem Ansatz die Stabilität der adaptiven Filter nicht gewährleistet. Die Schrittweite muss dafür, abhängig vom Eingangssingal, immer unter einem bestimmten Wert bleiben muss ($0 < \alpha < 2/3E\{\mathbf{X}^T\mathbf{X}\}$) [11]. Für sich alleine ist diese Methode also nicht realisierbar, weil sie nicht die nötige Robustheit für Echtzeitanwendungen und reale Umweltbedingungen aufweist. Sie wurde aber trotzdem in dieser Arbeit vorgestellt, weil alle noch folgenden Algorithmen theoretisch auf diesem adaptiven Beamformer aufbauen und deshalb die beschreibende Mathematik sowie die Erkenntnis über die Arbeitsweise dieses Ansatzes unabdingbar sind.

4.3 Frost Beamformer mit Erhöhung von G_w

4.3.1 Theorie

Um der unsicheren Stabilität der Filter beim normalen Frost Beamformer entgegenzuwirken, werden die adaptiven Prozesse zunächst auf eine stabile Form gebracht. Dafür wird die Schrittweite α mit der geschätzten Energie $\|\mathbf{X}\|^2$ des Eingangssingals normalisiert (Gl. 4.21). Im Nenner wird eine kleine Konstante ε addiert, um eine Division durch null zu verhindern. Durch diese Erweiterung wird der LMS zu einem „Normalized“ LMS (NLMS) Algorithmus.

$$\alpha_{\text{norm}} = \frac{\alpha}{\|\mathbf{X}\|^2 + \varepsilon} \quad (4.21)$$

Als weitere Verbesserung versucht man wieder (vgl. Kapitel 4.1) durch Erhöhung und Fixierung des Gewinns gegenüber unkorrelierten Störungen G_w die Empfindlichkeit für

Aufbautoleranzen zu verringern, indem man eine Regularisierung einführt. Allerdings lassen sich hier nicht so einfach wie beim fixen Array die unkorrelierten Störungen durch Regularisierung der Korrelationsmatrix auf einen konstanten Wert beschränken, da sich $\hat{\mathbf{R}}_{xx}$ bei jedem Update ändert und damit auch der Faktor μ für die Regularisierung. Man geht deshalb dazu über die Norm der Filterkoeffizienten zu beschränken („*Norm Constrained Adaptive Filter*“ – NCAF). Der Kehrwert der Norm ergibt für unkorrelierte Störungen den Gewinn G_w (4.22). Die Filtergewichte werden in einen konstanten Anteil \mathbf{W}_c und einen zeitvarianten Teil \mathbf{V} aufgeteilt [11]:

$$G_w(n) = \frac{1}{\mathbf{W}(n)^T \mathbf{W}(n)} = \frac{1}{\|\mathbf{W}(n)\|^2} \quad (4.22)$$

$$\mathbf{W}(n+1) = \mathbf{P}[\mathbf{W}(n) - \alpha_{\text{norm}} \mathbf{X}(n)y(n)] + \mathbf{W}_c \quad \text{mit} \quad \mathbf{V} = \mathbf{P}[\mathbf{W}(n) - \alpha_{\text{norm}} y(n)\mathbf{X}(n)] \quad (4.23)$$

$$\mathbf{W}(n+1) = \mathbf{V}(n) + \mathbf{W}_c \quad (4.24)$$

Die beiden Vektoren \mathbf{W}_c und \mathbf{V} sind wegen der Projektionsmatrix \mathbf{P} orthogonal zueinander ($\mathbf{P}\mathbf{W}_c = 0$). Mit dieser Aufspaltung in zwei orthogonale Anteile ist es möglich eine zusätzliche quadratische Nebenbedingung zu formulieren [11]:

$$G_w = \frac{1}{\|\mathbf{W}(n)\|^2} = \frac{1}{\|\mathbf{W}_c\|^2 + \|\mathbf{V}(n)\|^2} \leq \delta^2 \quad (4.25)$$

$$\Rightarrow \|\mathbf{V}(n)\|^2 \leq \frac{1}{\delta^2} - \|\mathbf{W}_c\|^2 = b^2 \quad (4.26)$$

Die Berechnung der neuen Gewichte erfolgt in zwei Schritten. Zuerst werden temporäre Gewichte $\tilde{\mathbf{V}}$ berechnet. Im zweiten Schritt wird $\tilde{\mathbf{V}}$ abhängig von der Nebenbedingung skaliert und schließlich zum konstanten Filteranteil dazu addiert. Da die Nebenbedingung aus Gleichung (4.26) eine im Ursprung zentrierte Kugel definiert und \mathbf{V} orthogonal zur Oberfläche steht, reicht eine einfache Skalierung aus. Gleichungen (4.27) bis (4.30) zeigen die Vorgehensweise bei der Adaption auf [11]:

$$\mathbf{W}(0) = \mathbf{W}_c \quad \text{und} \quad \mathbf{V}(0) = 0 \quad \rightarrow \quad \text{Startwerte}$$

$$b = \sqrt{10^{\left(\frac{-G_w}{10}\right)} - \mathbf{F}^T (\mathbf{C}_0^T \mathbf{C})^{-1} \mathbf{F}} \quad (4.27)$$

$$\tilde{\mathbf{V}}(n+1) = \mathbf{P}[\mathbf{V}(n) - \alpha_{\text{norm}} y(n)\mathbf{X}(n)] \quad (4.28)$$

$$\mathbf{V}(n+1) = \begin{cases} \tilde{\mathbf{V}}(n+1) & \text{wenn} \quad \|\tilde{\mathbf{V}}(n+1)\|^2 \leq b^2 \\ \frac{b\tilde{\mathbf{V}}(n+1)}{\|\tilde{\mathbf{V}}(n+1)\|} & \text{wenn} \quad \|\tilde{\mathbf{V}}(n+1)\|^2 > b^2 \end{cases} \quad (4.29)$$

$$\mathbf{W}(n+1) = \mathbf{W}_c + \mathbf{V}(n+1) \quad (4.30)$$

4.3.2 Simulationen

Die folgenden Diagramme bilden das Verhalten des Frost Beamformers mit normalisierten Filtern bei verschiedenen Anhebungen und Fixierungen des Gewinns an unkorrelierten Störungen (Abb. 4.8) und unter Einfluss von Positionierungs- und Verstärkungsfehlern (Abb. 4.10) ab. Als Vergleichsmöglichkeit ist in Abb. 4.8 zusätzlich das Beampattern des Frost Beamformers ohne normalisierte und normbeschränkte Filter abgebildet. Abb. 4.9 soll die Auswirkungen auf den Beamformer bei einer Normalisierung der Schrittweite verdeutlichen. Die Simulations- und Fehlerparameter sind identisch mit jenen aus Kapitel 4.2.2. und können dort nachgeschlagen werden.

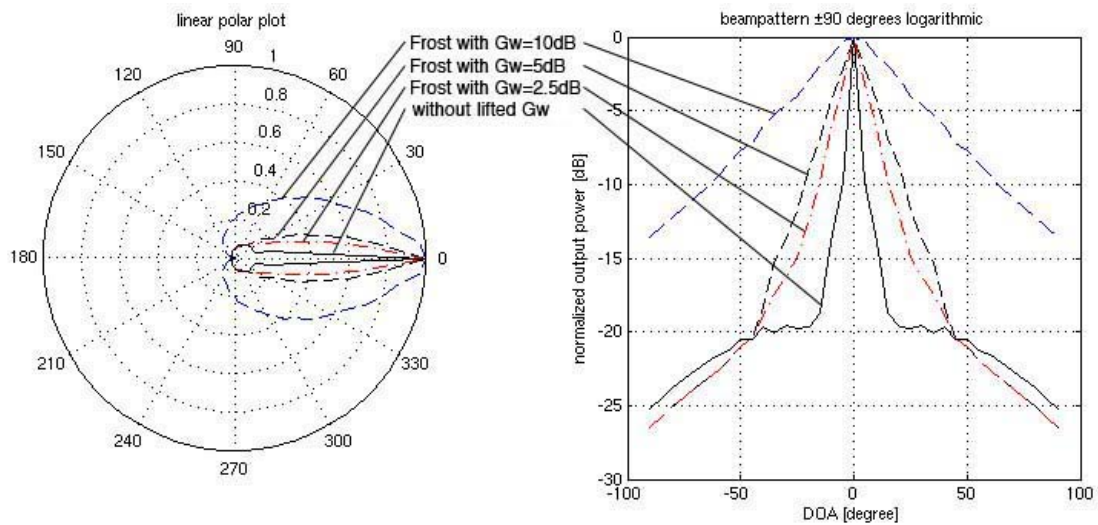


Abb. 4.8: Frost-B mit versch. G_w -Erhöhungen (MATLAB)

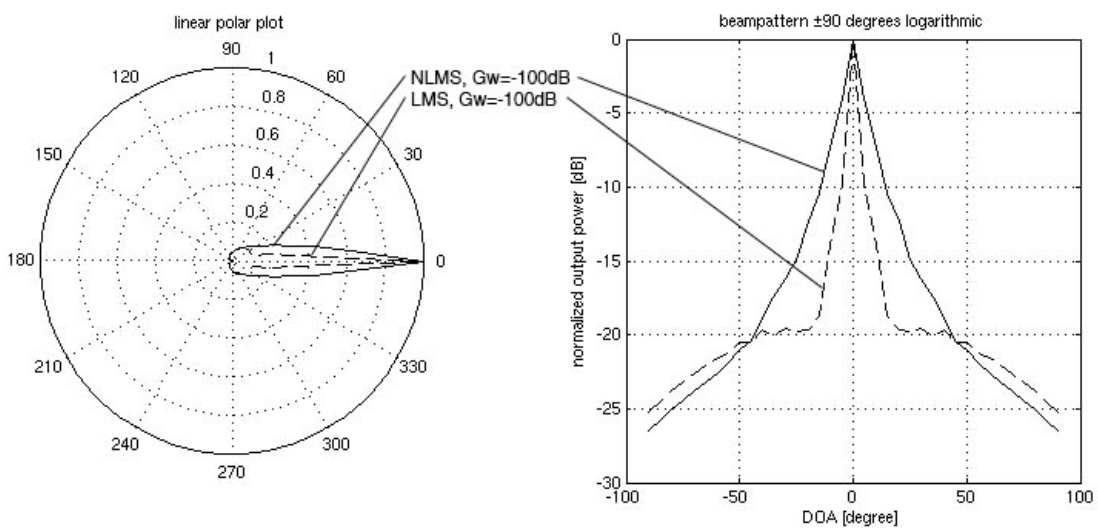


Abb. 4.9: Vergleich von LMS zu NLMS

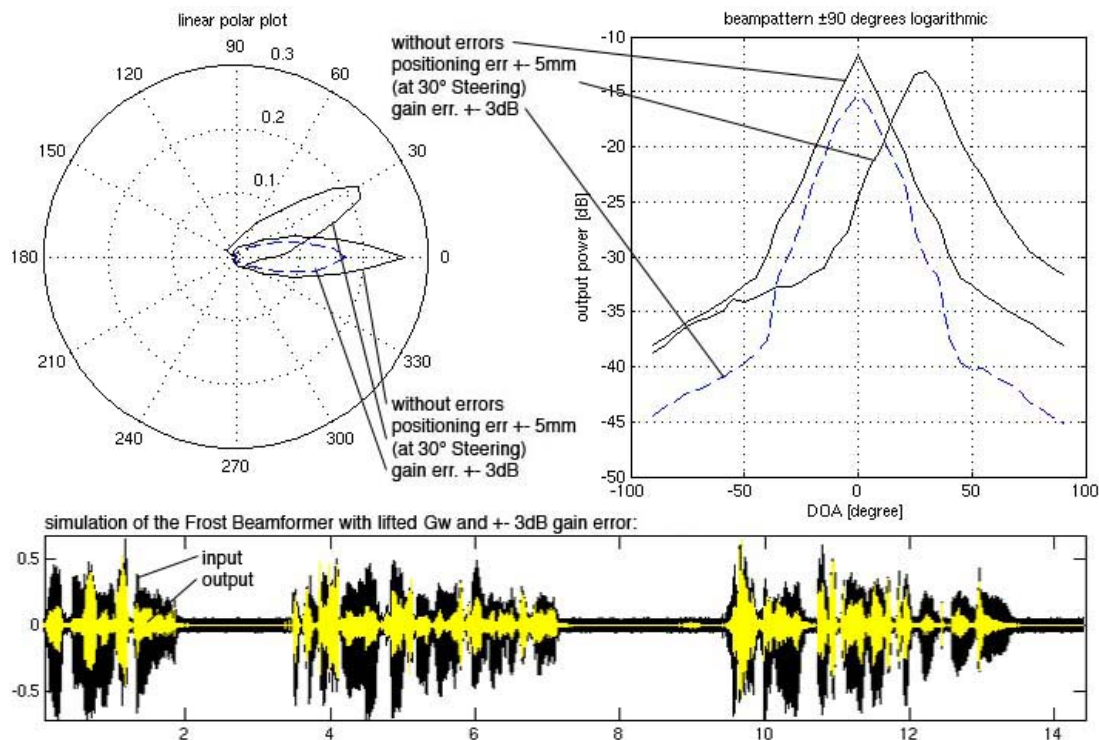


Abb. 4.10: Frost-B mit Positionierungs- und Verstärkungsfehlern, G_w -Erhöhung: 5dB (MATLAB)

4.3.3 Auswertung und Vergleich der Simulationen

In Abb. 4.8 erkennt man, dass die Keulbreite mit steigenden Werten von G_w zunimmt und sich immer weiter an die Richtcharakteristik des D&S-B annähert. Umgekehrt wandert die Richtcharakteristik bei geringeren Anhebungen von G_w hin zu der des konventionellen Frost Algorithmus. Geht man zur Dämpfung des Gewinnes über, so wird das Limiting der Filternorm immer geringer. Verhindert man zusätzlich die Normalisierung der Schrittweite, so kann man schließlich eine Anpassung an den einfachen Frost Beamformer beobachten (Abb. 4.9). Dies ist offensichtlich, da mit steigendem Gewinn die Norm der veränderlichen Filterkoeffizienten immer geringer gehalten wird und sich die endgültigen Koeffizienten $\mathbf{W}(n)$ an den konstanten Teil \mathbf{W}_c annähern, welcher genau den Koeffizienten eines bandpassgefilterten D&S-B entspricht. Verringert man hingegen G_w , so lässt man damit in Gleichung (4.29) höhere Normen für \mathbf{V} zu. Die variablen Koeffizienten wandern zur optimierten Frost Lösung. Durch die Normalisierung der Schrittweite wird jedoch ein beliebiges Ansteigen der Koeffizienten verhindert. Somit bleibt die Optimierung an einer bestimmten Schwelle hängen. Ohne Normalisierung können die Koeffizienten das Ausgangssignal ungehindert nach der Frost Beschränkung optimieren.

In Abb. 4.10 wird das Array mit den selben Fehlern wie in Kapitel 4.2.2 behaftet. Es sind deutliche Verbesserungen gegenüber dem konventionellen Algorithmus zu erkennen. Mit einem Positionierungsfehler tritt eine deutlich niedrigere Abschwächung der Hauptkeule ein. Auffallend ist auch, dass im Falle von Verstärkungsfehlern an den Sensoren nun das Ausgangssignal nicht mehr komplett minimiert wird, sondern der Beamformer robust gegenüber von Verstärkungsfehlern ist.

Es ist nur eine leichte Abschwächung des Signals zu beobachten. Aus den theoretischen Betrachtungen heraus müsste dieser Ansatz als Echtzeitanwendung unter realen Umweltbedingungen umsetzbar sein.

4.3.4 Realisierung in PD und akustische Vermessung

Die akustische Vermessung des Ansatzes nach Methode 1 (s. Kapitel 5) hat andere Ergebnisse als die Simulation gezeigt. Obwohl der Algorithmus als robust gilt, hat das System im Echtzeitbetrieb nicht richtig funktioniert. Eine leistungsmäßige Verbesserung im Vergleich zum F&S-B ist nur bei rosa Rauschen zu sehen. Ansonsten ist eine sehr schmale Hauptkeule zu beobachten, die niemals genau auf einen Sprecher ausgerichtet werden kann. Das hat sich auch durch informelle Hörversuche in halliger Umgebung bestätigt. Es kommt zu starker Wunschsignalauslöschung. Die steuernden Parameter des Beamformers waren auf $\alpha = 0,1$ und $G_w = 5\text{dB}$ eingestellt.

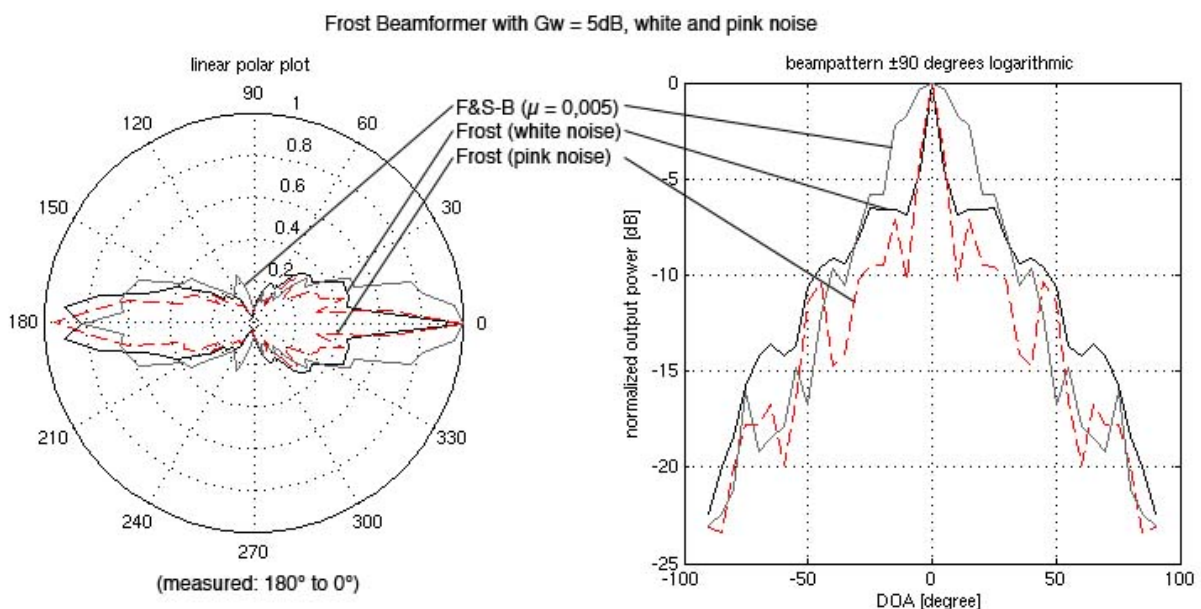


Abb. 4.11: gemessene Ausgangsleistung des Frost-B mit angehobenem $G_w = 5\text{dB}$

4.3.5 Evaluierung

Obwohl dieser Ansatz in den Simulationen viel versprechende Ergebnisse liefert, kann er unter realen Bedingungen nicht überzeugen. Die Anhebung von G_w liefert keine wesentlichen Verbesserungen, da die Hauptkeule genau so spitz zuläuft, wie beim herkömmlichen Frost Algorithmus. Lediglich die Steigung der Dämpfungskurve ändert sich. Deswegen bleibt es schwierig, den Beam wirklich exakt auf ein Target auszurichten und Wunschsignalauslöschung zu vermeiden. Es muss eine Methode gefunden werden, die eine breitere Hauptkeule aufweist und auf diese Weise die Robustheit des Arrays erhöht.

4.4 Frost Beamformer mit „Steering Vector Correction“

4.4.1 Theorie

Dieser Ansatz ist eine Erweiterung des Frost Algorithmus, mit dem Ziel, durch zusätzlich vorgeschaltete adaptive Filter Verzögerungsfehler von Signalen und Steering Vector Errors auszugleichen. Die Neuerung wurde im Januar 2004 in den IEEE Signal Processing Letters vorgestellt und ist in [10] ausführlich beschrieben.

Ausgehend von Abb. 4.5 stellt sich das Eingangssignal an einem Sensor folgendermaßen dar:

$$x_k(t) = s(t - \tau_k) + n_k(t) \quad k = 1, 2, \dots, K \quad (4.31)$$

$s(t)$ beschreibt das Wunschsinal aus der Look Direction, τ_k bestimmt die Verzögerung am k -ten Sensor, die das Signal aufgrund der Einfallsrichtung zum Referenzsensor hat und $n_k(t)$ steht für alle Störungen, die vom k -ten Sensor aufgenommen werden. Um die Einfallsrichtung des Schalls auszugleichen, wird, wie bereits oben erwähnt, das Array vor dem Prozessor am k -ten Sensor um die Zeit τ_k verzögert und somit in Richtung Target gesteuert. Da diese Verzögerungen in der Realität aber nie so genau eingestellt werden können, dass sie das Wunschsinal optimal zur Deckung bringen und auch die „Direction of Arrival“ (DOA) nie genau bekannt ist, führt das zwangsläufig zur Auslöschung von nicht optimal synchronem Target Signal. Man kann das Signal nach dem Ausgleich der Einfallsrichtung beschreiben als:

$$x_k(t) = s(t - \Delta_k) + n_k(t + \tau_k - \Delta_k) \quad k = 1, 2, \dots, K \quad (4.32)$$

$$\Rightarrow x_k(n) = s(nT_s - \Delta_k) + n_k(nT_s + \tau_k - \Delta_k) \quad \text{mit } T_s = \text{Samplingperiode} \quad (4.33)$$

Δ_k repräsentiert die restliche Verzögerung, haftend auf dem Wunschsinal. Das bedeutet, das Wunschsinal ist nur dann perfekt überlagert, wenn alle Δ_k für $k=1,2,\dots,K$ gleich sind. In diesem Fall würde der herkömmliche Frost Algorithmus ausreichen. Für den Fall, dass nicht alle Δ_k gleich sind wird folgende Erweiterung vorgeschlagen:

Angenommen $\mathbf{X}'(n)$ ist der Signalvektor⁷, der den Prozessor erreicht, wenn Verzögerungsfehler vorhanden sind. Das bedeutet, das Wunschsinal in $\mathbf{X}'(n)$ ist nicht ideal überlagert und der Gewichtsvektor \mathbf{W}_{opt} aus Gl. (4.13), entstanden aus \mathbf{C}_0 führt zu keiner optimalen Problembeschreibung. Somit gibt es eine lineare Beziehung zwischen dem idealen Vektor $\mathbf{X}(n)$ und dem fehlerbehafteten Vektor $\mathbf{X}'(n)$, die sich in einer Zeitverschiebung äußert. Diese Operation kann näherungsweise durch eine lineare Transformationsmatrix \mathbf{Y} durchgeführt werden.

$$\mathbf{X}'(n) = \mathbf{Y}\mathbf{X}(n) \quad (4.34)$$

⁷ Alle Signalvektoren und Matrizen wie $\mathbf{X}(n)$ oder \mathbf{C}_0 haben dieselben Dimensionen wie schon in Kapitel 4.2.1 beschrieben und können dort nachgeschlagen werden.

Wenn man vorerst davon ausgeht, dass alle Δ_k bekannt sind, lässt sich die Matrix \mathbf{Y} der Größe $KJ \times KJ$ (wobei: K = Anzahl der Sensoren, J = Anzahl der Filterstufen) wie in Gleichung (4.35) definieren.

$$\mathbf{Y} = [\mathbf{y}_{1,1}^T \dots \mathbf{y}_{K,1}^T \quad \mathbf{y}_{1,2}^T \dots \mathbf{y}_{K,2}^T \quad \dots \quad \mathbf{y}_{1,J}^T \dots \mathbf{y}_{K,J}^T]^T \quad (4.35)$$

$$\mathbf{y}_{k,j} = [\mathbf{g}_1 \dots \mathbf{g}_i \dots \mathbf{g}_J] \quad \text{mit} \quad k = 1 \dots K, \quad j = 1 \dots J \quad \text{und} \quad i = 1 \dots J \quad (4.36)$$

$$\mathbf{g}_i = [0 \dots 0 \quad h_k(i-j) \quad 0 \dots 0] \quad \text{Dimension: } 1 \times K \quad (4.37)$$

Das einzige Element, das im Vektor \mathbf{g}_i nicht 0 ist, befindet sich an der k -ten Stelle und ist ein Koeffizient des Interpolationsfilters $h_k(n)$, der das Signal entlang der Zeitachse um Δ_k verschiebt.

$$h_k(n) = \frac{\sin\left[\pi\left(n - \frac{\Delta_k}{T_s}\right)\right]}{\pi\left(n - \frac{\Delta_k}{T_s}\right)} \quad \text{mit} \quad n = \dots -1, 0, 1, \dots \quad (4.38)$$

Um eine Zeitverzögerung Δ_k vollständig kompensieren zu können, würde man eine unendlich lange Impulsantwort und damit eine unendliche Filterlänge benötigen. Trotzdem ist Gl. (4.35) eine gute Näherung, wenn J groß genug ist und der Betrag von $h_k(n)$ mit $1/|n|$ abnehmen kann.

Mit dem fehlerbehafteten Eingangsvektor stellt sich folgendes erweitertes Optimierungsproblem (4.41), abgeleitet von Gl. (4.7).

$$\min_{\mathbf{W}'} P' = \min_{\mathbf{W}'} \mathbf{W}'^T \mathbf{R}_{x'x'} \mathbf{W}' \quad (4.38)$$

\mathbf{W}' ist der Gewichtsvektor aufgebaut nach (4.4) und P' steht für die Ausgangsleistung des Systems mit $\mathbf{X}'(n)$ als Eingang. Mit Gleichung (4.34) wird (4.38) zu

$$\min_{\mathbf{W}'} P' = \min_{\mathbf{W}'} \mathbf{W}'^T \mathbf{Y} \mathbf{R}_{xx} \mathbf{Y}^T \mathbf{W}' \quad (4.39)$$

Vergleicht man nun Gleichung (4.39) mit (4.7), stellt sich heraus, dass \mathbf{C}_0 nicht mehr die richtige Constraint Matrix für \mathbf{W}' sein kann, sondern \mathbf{C} . Daraus ergibt sich ein neuer Ansatz:

$$\mathbf{C} = \mathbf{Y} \mathbf{C}_0 \quad (4.40)$$

$$\Rightarrow \min_{\mathbf{W}'} P' = \min_{\mathbf{W}'} \mathbf{W}'^T \mathbf{R}_{x'x'} \mathbf{W}' \quad \text{mit} \quad \mathbf{C}^T \mathbf{W}' = \mathbf{F} \quad (4.41)$$

In der Praxis sind die restlichen Verzögerungen Δ_k aber leider nicht bekannt. \mathbf{Y} kann also nicht direkt mit den Gleichungen (4.35) bis (4.38) bestimmt werden. Das bedeutet, \mathbf{C} kann auch nicht direkt mit (4.40) aufgestellt werden. Unter der Annahme eines sehr kleinen Δ_k wird die Norm $\|\mathbf{C} - \mathbf{C}_0\|$ ebenso kleine Werte annehmen, d. h. \mathbf{C} wird somit in direkter Nachbarschaft von \mathbf{C}_0 liegen. Wenn \mathbf{C} der aktuellen Situation des Arrays mit Verzögerungsfehlern genügt wird das Wunschsignal mit minimaler Verzögerung übertragen. Andernfalls wird es als Störsignal behandelt ausgelöscht.

Das bedeutet, dass die optimale Ausgangsleistung P'_{opt} unter der Berücksichtigung von verschiedenen \mathbf{C} -Matrizen ein lokales Maximum haben muss. Deshalb wird eine Methode angestrebt um das richtige \mathbf{C} zu finden. Diese Methode sucht in der Nachbarschaft von \mathbf{C}_0 , um die Ausgangsleistung des Beamformers P'_{opt} zu maximieren. Das Kriterium für den Suchalgorithmus ergibt sich zu:

$$\max_{\mathbf{C}} P'_{opt} \quad \text{mit} \quad \|\mathbf{C} - \mathbf{C}_0\| \leq \sigma \quad (4.42)$$

Dabei ist σ eine kleine reelle, positive Zahl, die die Weite der zulässigen Suchregion um \mathbf{C}_0 herum bestimmt und experimentell ermittelt werden kann. Die optimale Ausgangsleistung des Beamformers P'_{opt} ist durch Gleichung (4.43) gegeben.

$$P'_{opt} = \mathbf{F}^T \left(\mathbf{C}^T \mathbf{R}_{x'x'}^{-1} \mathbf{C} \right)^{-1} \mathbf{F} \geq 0 \quad (4.43)$$

Damit lässt sich Gl. (4.42) auch äquivalent anschreiben als

$$\min_{\Delta_k} \frac{1}{P'_{opt}} = \min \frac{1}{\mathbf{F}^T \left(\mathbf{C}^T \mathbf{R}_{x'x'}^{-1} \mathbf{C} \right)^{-1} \mathbf{F}} \quad \text{mit} \quad \|\mathbf{C} - \mathbf{C}_0\| \leq \sigma \quad (4.44)$$

Es ist numerisch stabiler den Kehrwert der Leistung zu minimieren, als die Leistung zu maximieren. Unter Zuhilfenahme von Gl. (4.40) wird aus (4.44)

$$\min_{\Delta_k} \frac{1}{P'_{opt}} = \min \frac{1}{\mathbf{F}^T \left(\mathbf{C}_0^T \mathbf{Y}^T \mathbf{R}_{x'x'}^{-1} \mathbf{Y} \mathbf{C}_0 \right)^{-1} \mathbf{F}} \quad \text{mit} \quad \|\mathbf{Y} \mathbf{C}_0 - \mathbf{C}_0\| \leq \sigma \quad (4.45)$$

Das in Gl. (4.45) dargestellte Problem ist mehrdimensional und nichtlinear und kann mit der Gradientenmethode gelöst werden. Ziel ist es, die Restverzögerung Δ_k für jeden Sensor zu finden. Die partielle Ableitung von $1/P'_{opt}$ nach Δ_k ist gegeben durch:

$$\frac{\partial}{\partial \Delta_k} \left(\frac{1}{P'_{opt}} \right) = - \frac{1}{P'^2_{opt}} \frac{\partial P'_{opt}}{\partial \Delta_k} \quad (4.46)$$

$$\text{mit} \quad \frac{\partial P'_{opt}}{\partial \Delta_k} = -\mathbf{F}^T \mathbf{Q}^{-1} \frac{\partial \mathbf{Q}}{\partial \Delta_k} \mathbf{Q}^{-1} \mathbf{F} \quad (4.47)$$

$$\text{und} \quad \mathbf{Q} = \mathbf{C}_0^T \mathbf{Y}^T \mathbf{R}_{x'x'}^{-1} \mathbf{Y} \mathbf{C}_0 \quad (4.48)$$

$$\Rightarrow \quad \frac{\partial \mathbf{Q}}{\partial \Delta_k} = \mathbf{C}_0^T \frac{\partial \mathbf{Y}^T}{\partial \Delta_k} \mathbf{R}_{x'x'}^{-1} \mathbf{Y} \mathbf{C}_0 + \mathbf{C}_0^T \mathbf{Y}^T \mathbf{R}_{x'x'}^{-1} \frac{\partial \mathbf{Y}}{\partial \Delta_k} \mathbf{C}_0 \quad (4.49)$$

Die partielle Ableitung von \mathbf{Y} kann über die Gleichungen (4.35) bis (4.38) zusammen mit (4.50) und (4.51) berechnet werden.

$$\frac{\partial h_k(n)}{\partial \Delta_k} = \begin{cases} 0 & \text{wenn } n - \frac{\Delta_k}{T_s} = 0 \\ v & \text{wenn } n - \frac{\Delta_k}{T_s} \neq 0 \end{cases} \quad (4.50)$$

$$v = -\frac{\cos\left[\pi\left(n - \frac{\Delta_k}{T_s}\right)\right]}{T_s\left(n - \frac{\Delta_k}{T_s}\right)} + \frac{\sin\left[\pi\left(n - \frac{\Delta_k}{T_s}\right)\right]}{\pi T_s\left(n - \frac{\Delta_k}{T_s}\right)^2} \quad (4.51)$$

Nachdem die Restverzögerungen Δ_k relative Größen darstellen, ist es nötig, ein Δ_k (mit $\Delta_1 = 0$) als Bezugspunkt zu definieren, um das lokale Maximum von P'_{opt} zu finden.

Zusammenfassend lässt sich der Algorithmus in 5 iterativen Schritten darstellen:

Schritt 1: Aktualisierung der Korrelationsmatrix

$$\mathbf{R}_{X'X'}^{[n]} = \eta \mathbf{R}_{X'X'}^{[n-1]} + (1 - \eta) \mathbf{X}'(n) \mathbf{X}'(n)^T \quad n = 1, 2, \dots \quad (4.52)$$

η ist der Aktualisierungskoeffizient und sollte nahe bei 1 liegen.

Schritt 2: Berechnung der partiellen Ableitungen von $1/P'_{\text{opt}}$ nach Δ_k für alle Δ_k , $k = 2 \dots K$, unter Zuhilfenahme der Gleichungen (4.46) bis (4.49) und anschließendem Aktualisieren der alten Δ_k .

$$\Delta_k^{[n]} = \Delta_k^{[n-1]} - u \frac{\partial}{\partial \Delta_k} \left(\frac{1}{P'_{\text{opt}}} \right) \quad n = 1, 2, \dots \quad (4.53)$$

u ist eine kleine positive Zahl und bestimmt die Adaptionsgeschwindigkeit. $\Delta_1^{[n]}$ wird auf 0 gesetzt, ebenso die Startwerte von Δ_k , $k = 2 \dots K$.

Schritt 3: Die Constraint Matrix wird aktualisiert zu:

$$\mathbf{C}^{[n]} = \mathbf{Y}^{[n]} \mathbf{C}_0 \quad (4.54)$$

$\mathbf{Y}^{[n]}$ wird dabei mit den neuen $\Delta_k^{[n]}$ nach den Gleichungen (4.35) bis (4.38) berechnet.

Schritt 4: Überprüfen der Beschränkung $\|\mathbf{C}^{[n]} - \mathbf{C}_0\| \leq \sigma$. Wenn diese Bedingung verletzt ist, dann befindet sich das einfallende Signal nicht in der zugelassenen Richtung und wird als Störung gewertet. Dafür setzt man einfach \mathbf{C} zurück auf \mathbf{C}_0 . Wenn die Bedingung nicht verletzt wird, wiederholen sich die Schritte 2 bis 4 solange, bis $\mathbf{C}^{[n]}$ zu der gewünschten Constraint Matrix \mathbf{C}_{opt} konvergiert.

Schritt 5: Der konventionelle Frost Algorithmus wird mit der neuen Matrix \mathbf{C}_{opt} berechnet. Alternativ können auch einfach zusätzliche Filter mit den Impulsantworten $h_k(n)$ zwischen die Steering Stufe und dem Frost Prozessor geschaltet und der Frost Algorithmus wie ursprünglich mit \mathbf{C}_0 bestimmt werden.

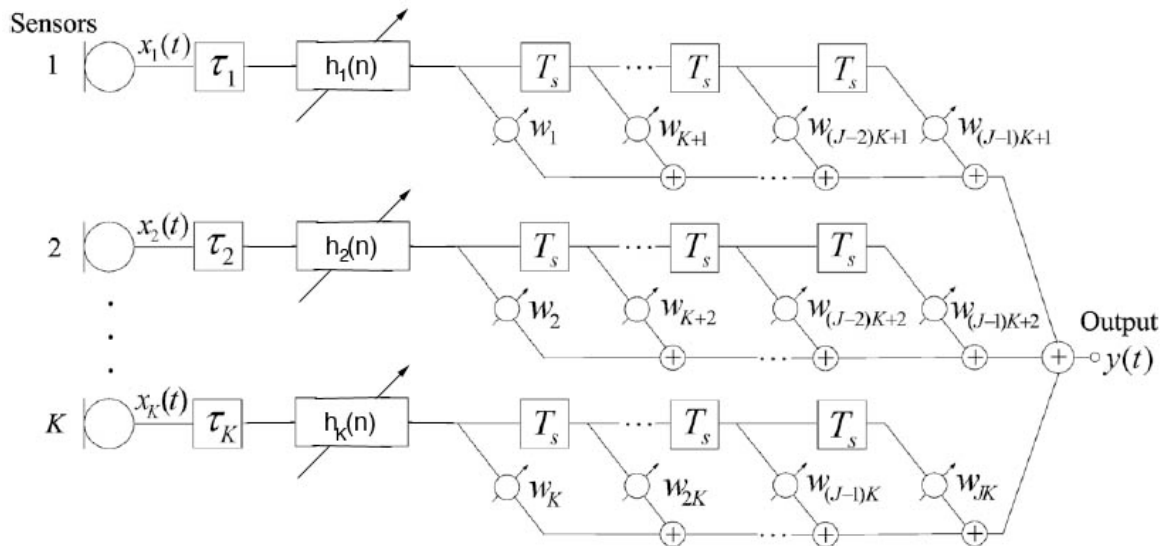


Abb. 4.12: Frost Beamformer mit adaptiver Steering Vector Correction

4.4.2 Simulationen

Die MATLAB Simulationen wurden alle mittels des Algorithmus der Aktualisierung der Constraint Matrix durchgeführt. Die Realisierung mit zusätzlichen Filtern genügt eher den Anforderungen einer Echtzeitanwendung und ist für offline Simulationen nicht notwendig.

Für die Simulationen des Frost Algorithmus wurden die gleichen Parameter wie schon in Kapitel 4.2.2 und 4.3.2 verwendet. Abb. 4.13 verdeutlicht den Adaptionsprozess der vorgestellten Erweiterung bei einem Steering Vector Error von $+10^\circ$. Im oberen Teil sind der Verlauf der Ausgangsleistung P'_{opt} und der sich adaptierenden Restverzögerungen Δ_k dargestellt. Die konstanten Linien im rechten Diagramm sind die wirklichen Steering Delays, die das Array direkt auf das Target ausrichten würden. In den unteren 6 Diagrammen sind die Interpolationsfilter $h_k(n)$ nach der Konvergenz des Algorithmus abzulesen. Die Anzahl an Wiederholungen der Schritte 2 bis 4 wurde auf 200 begrenzt. Der Koeffizient η zur Aktualisierung der Korrelationsmatrix in Schritt 1 wurde auf 0,995 gesetzt. Um eine ausreichende Mittelung für $\mathbf{R}_{X'X'}$ zu erhalten, wurde Schritt 1 zuerst 200 mal durchgeführt. Die Korrelationsmatrix wurde durch Addition einer Einheitsmatrix mit dem Faktor 0,01 reguliert und anschließend auf 1 normalisiert. Diese Maßnahme entspricht einem Hinzufügen von unkorreliertem Sensorrauschen und ist sehr wichtig für die Stabilität des adaptiven Prozesses. Für Schritt 3 wurde die Konstante $u = 0,8e^{-11}$ gewählt. Abb. 4.14 zeigt einen Vergleich der Richtplots des konventionellen Frost Beamformers, des Algorithmus mit angehobenem und fixiertem G_w und des Ansatzes mit adaptiver Steering Vector Correction bei einer Toleranz von $\pm 15^\circ$. Für Schritt 4 ist dafür die Abbruchbedingung σ auf 2,5 gesetzt worden.

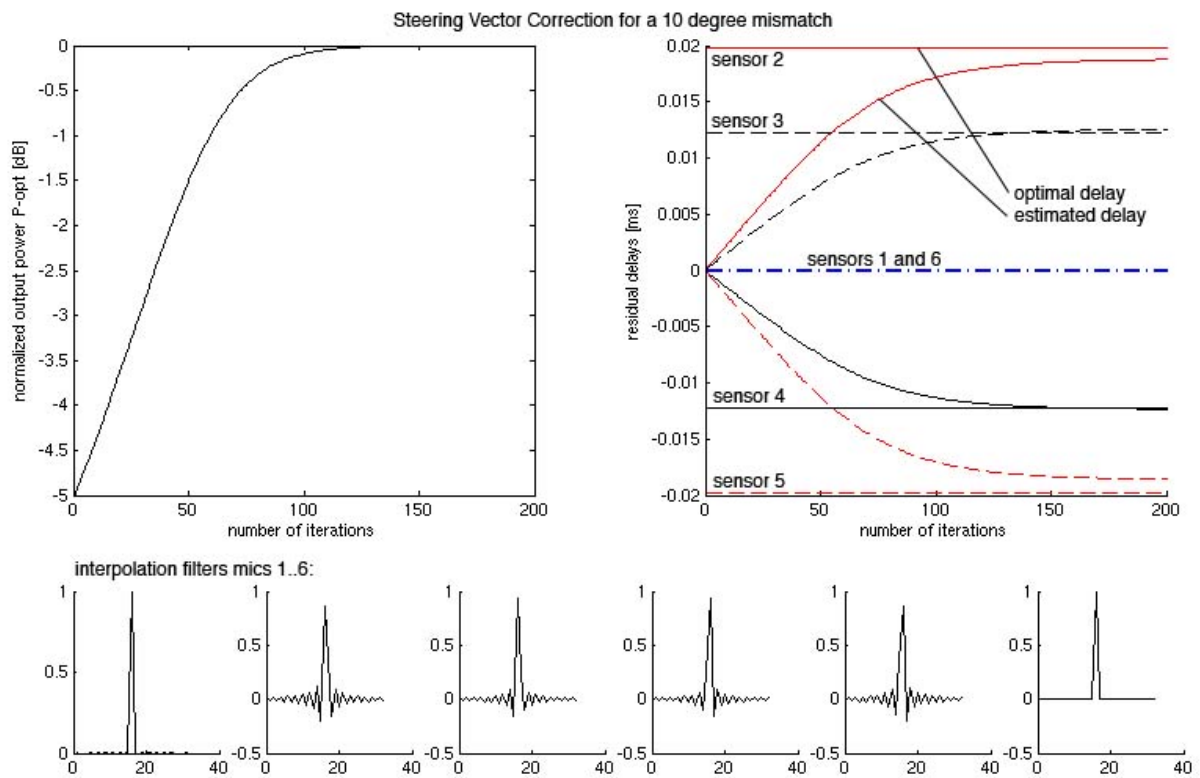


Abb. 4.13: Adaption des Algorithmus bei 10° Targetverfehlung (MATLAB)

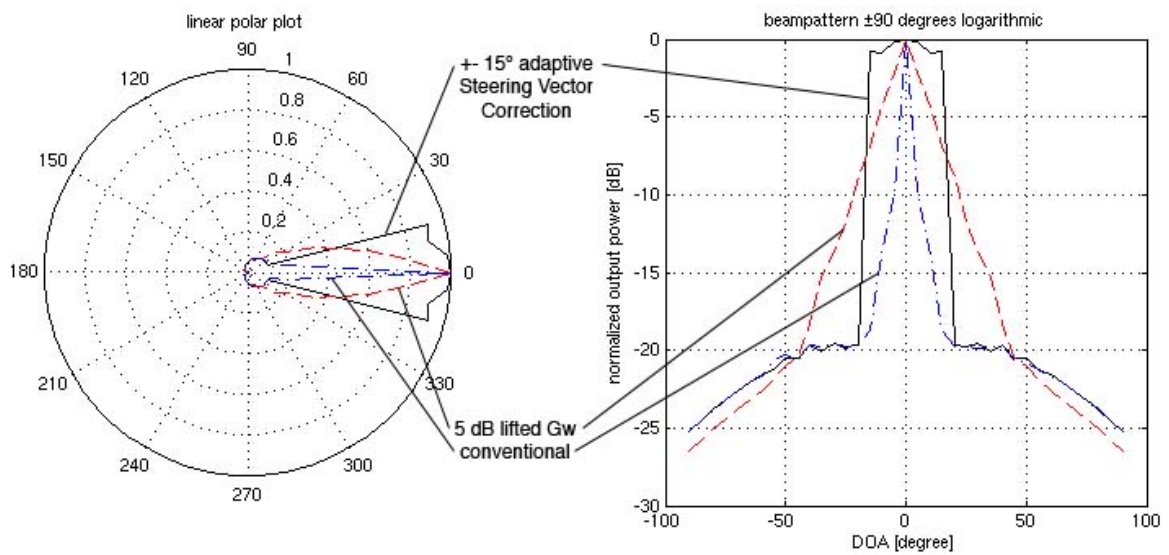


Abb. 4.14: Richtplot: Frost Beamformer mit adapt. Steering Vector Correction (MATLAB)

4.4.3 Auswertung und Vergleich der Simulationen

In Abb. 4.13 erkennt man, dass der Algorithmus für die verwendeten Parameter stabil ist. Die adaptiven Delays nähern sich mit fortschreitenden Wiederholungen immer mehr an die wirklichen Steering Delays an. Die Ausgangsleistung des Systems steigt. Das Resultat ist in Abb. 4.14 zu erkennen. Durch die adaptive Steering Vector Correction wächst der Akzeptanzwinkel, wie in der Abbruchbedingung festgelegt, auf insgesamt 30° . Einfallende Signale innerhalb dieser Region werden nun nach der Frost-Bedingung unverzerrt wiedergegeben. Hingegen sieht man bei Betrachtung der vorherigen Methoden, dass diese nicht in der Lage sind, bei einem Steering Vector Error ein wirklich unverzerrtes Signal zu liefern. Der Ansatz mit der Anhebung und Fixierung des Gewinns gegenüber von unkorrelierten Störungen hat zwar eine breitere Hauptkeule, liefert aber trotzdem nur für einen Winkel ein unverzerrtes Signal. Während er bei -15° schon eine Dämpfung von etwa 8dB aufweist, ist das Signal im Falle der adaptiven Steering Vector Correction noch nahezu unbedämpft.

Allerdings hat sich in weiteren Simulationen gezeigt, dass diese Methode bei der Verwendung von Sprache als Targetsignal nicht immer konvergiert, d. h. instabil wird. Die Simulationsparameter mussten, um eine stabile Lösung bei unterschiedlichen Signalausschnitten zu erhalten, in jedem Fall neu ermittelt werden. Bei Konvergenz haben sich in den meisten Fällen nicht die richtigen Steering Delays eingestellt. Es konnte beobachtet werden, dass die Verzögerungen nicht gegen die der Geometrie entsprechenden Werte konvergierten. D. h. ein Sensor konvergiert gegen das Delay eines anderen Sensors, wobei dieser dann nicht mehr optimiert werden kann. Die Ursachen hierfür sind nicht erwiesen.

4.4.4 Realisierung in PD

Dieser Algorithmus wurde aus mehreren Gründen nicht in PD realisiert. Der Hauptgrund ist sicherlich die fehlende Stabilität für Sprachsignale. Da die Methode schon in den Simulationen zeigt, dass sie eigentlich für die gewünschte Anwendung zu keinem befriedigendem Ergebnis führt, ist eine Überführung in eine Echtzeitumgebung nicht sinnvoll. Selbst bei ausreichender Stabilität, ist es immer noch schwierig einen Ansatz mit einem so hohen Rechenaufwand in einer Echtzeitanwendung zu realisieren. In Verbindung mit dem vorgestellten 6 Elemente Array werden hier 12 adaptive Filter und die ständige Berechnung der Filterkoeffizienten über hochrangige Matrizenoperationen erforderlich. Das kann zu einem Rechenleistungsproblem führen. Es müsste ein Weg gefunden werden, der die Berechnung von so hochrangigen Matrizen umgeht oder zumindest vereinfacht.

4.4.5 Evaluierung

Diese Methode führte in der Simulation zu keiner allgemein gültigen Stabilität. Selbst wenn man als Target Signal nur in sich unkorrelierte Signale erwarten würde, tritt immer noch das Problem fehlender Robustheit gegen Verstärkungsfehler auf. Bei Implementierung könnte versucht werden mittels der Methode aus Kapitel 4.3 höhere Stabilität zu erzielen. Fraglich ist, ob es durch diese Methode in der Praxis überhaupt zu einer Verbesserung der Robustheit gegenüber von Ausrichtungsfehlern des Arrays kommen kann. Es wird nur die Hauptkeule des Beamformers, abhängig von der Korrelationsmatrix hin und her geschoben. Das bedeutet, wenn eine Korrektur der Look Direction statt gefunden hat, dann darf sich die Situation bis zur nächsten Korrektur nicht verändern, sonst gibt es wieder Auslöschungen des Wunschssignals. Jede Korrektur dauert erstens eine gewisse Zeit an und zweitens muss die Frequenz an Korrekturen wegen der erwarteten hohen benötigten Rechenleistung wahrscheinlich gering gehalten werden. Die Frage ist deshalb, ob die Korrekturfrequenz ausreicht, sodass man die Bedingungen, die zwischen den Korrekturen herrschen, als stationär betrachten kann.

4.5 Griffiths-Jim Beamformer

4.5.1 Überleitung vom Frost Beamformer

Der Griffiths-Jim Beamformer (G-J-B) ist ein Derivat des Frost Beamformers. Er erreicht ebenso die harte Frost Beschränkung in der Wunschrichtung, arbeitet dabei allerdings mit einem unbeschränkten „least-squares“ Algorithmus. Das Blockdiagramm ist in Abb. 4.15 dargestellt. Man kann seine Struktur in einen fixen Beamformer (z.B. ein D&S-B), eine Blocking Matrix (BM) und einen Multiple Input Canceller (MC) unterteilen. Die BM formt eine Null in die Richtung des Target Signals und spielt dem MC somit Referenzsignale aller Störungen zu, im idealen Fall ohne jegliche Anteile von Wunschsinal. Der MC bildet dann mit Hilfe des Ausgangsignals Reproduktionen der Störsignale und subtrahiert diese vom Ausgang des Beamformers. Wie beim zuvor vorgestellten Frost Prozessor werden die Signale zuerst synchronisiert (im Blockschaltbild bereits geschehen). In der Terminologie der adaptiven Rauschunterdrückung kann man das primäre Signal als eine gefilterte Version der Summe der Sensorsignale betrachten. Der Filter hat die gleiche Impulsantwort wie der Vektor \mathbf{F} in Kapitel 4.2 [13]. Möchte man keine Filterung erreichen, setzt man den Filter so, dass er bei allen Frequenzen die gleiche Verstärkung liefert. In diesem Fall enthält das primäre Signal das Target aus der Wunschrichtung, sowie alle Störungen. Wenn man die K synchronisierten Signale $x_k(n)$ in Paaren voneinander abzieht (Delay & Subtract Beamformer), erhält man am Ausgang der BM $K-1$ Referenzsignale $z_k(n)$, die ein Minimum an Target (das Target wird „geblockt“) und sonst alle restlichen Signale aus anderen Richtungen (Störer) ungedämpft enthalten. Diese Referenzsignale werden im MC adaptiven Filtern zugeleitet, mit deren Hilfe Nachbildungen der mit den Störsignalen korrelierten Komponenten gebildet werden, die man schließlich summiert und vom Ausgangssignal des fixen Beamformers abzieht.

Das Ergebnis ist ein Systemausgang, der ein unverzerrtes (oder vorbestimmt gefiltertes) Signal aus der Wunschrichtung enthält und ein Minimum an Störungen aus allen anderen Richtungen. Man kann den G-J-B als einen adaptiven Rauschunterdrücker mit mehreren Referenzsignalen betrachten, die alle in der BM vorbearbeitet werden. In einem adaptiven Rauschunterdrücker wird das Referenzsignal möglichst nahe an der Störquelle entnommen. Hingegen werden beim G-J-B durch die Direktivität der BM Signale, die mit den Störungen korreliert sind, extrahiert [1]. Betrachtet man K Sensoren, dann sind $K-1$ adaptive Filter mit jeweils J Taps vorhanden. Wenn keines dieser Filtergewichte beschränkt ist, ist die Anzahl der Freiheitsgrade $(K-1)J$ die gleiche wie beim originalen Frost Beamformer mit KJ Gewichten [13]. Griffiths und Jim haben in [4] gezeigt, dass der G-J-B das gleiche Verhalten wie der Frost Beamformer zeigt, obwohl sie verschiedenen mathematischen Beschreibungen folgen. Der Frost Beamformer ist die „*direct form implementation of linearly constrained adaptive array processing*“ (Abb. 4.5) und der G-J-B wird als „*generalized sidelobe cancelling form*“ bezeichnet (Abb. 4.15) [4]. Die Vorteile bei diesem Algorithmus liegen im Gegensatz zum klassischen Frost darin, dass jede Verarbeitungsstufe strukturell sehr leicht ersichtlich und zugänglich ist und man nahezu jeden adaptiven Algorithmus in der Implementierung verwenden kann.

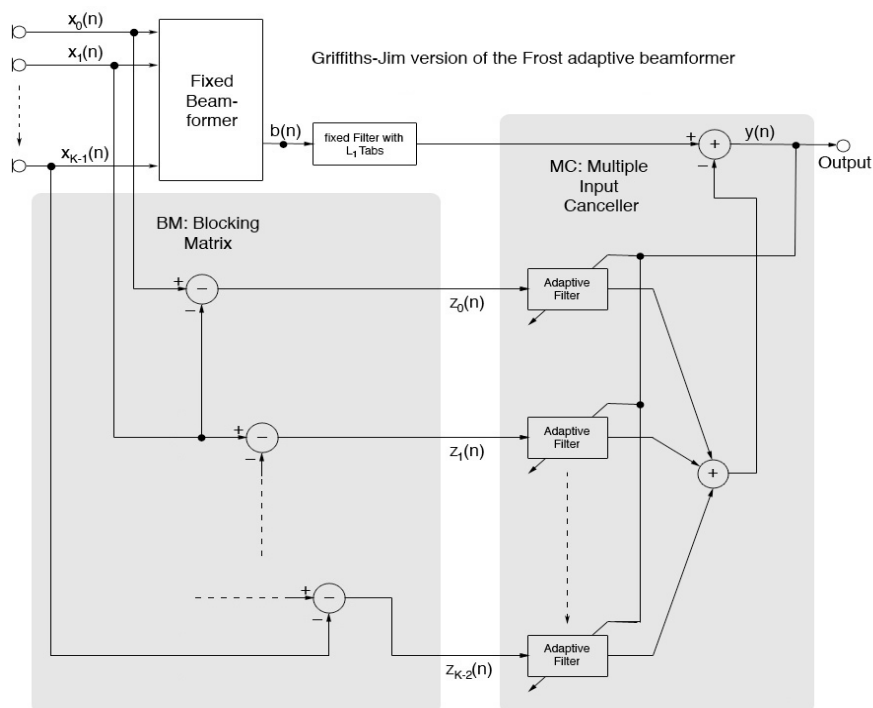


Abb. 4.15: Strukturbild des Griffiths-Jim Beamformers

Genau wie beim konventionellen Frost Algorithmus, ist der G-J-B ebenso wenig robust gegenüber Verzögerungs- und Verstärkungsfehlern an den Sensoren. Solche Fehler führen zwangsläufig zu einem *Signal Leakage* in der Blocking Matrix. Das bedeutet, dass Teile des Wunschssignals im Referenzsignal $z_k(n)$ erhalten bleiben. Das hat eine Auslöschung des Wunschssignals durch den MC zufolge. Es existiert eine Vielzahl von Ansätzen, die versuchen Target Leakage so gering wie möglich zu halten (siehe [1]).

4.5.2 CCAF-NCAF Struktur mit AMC

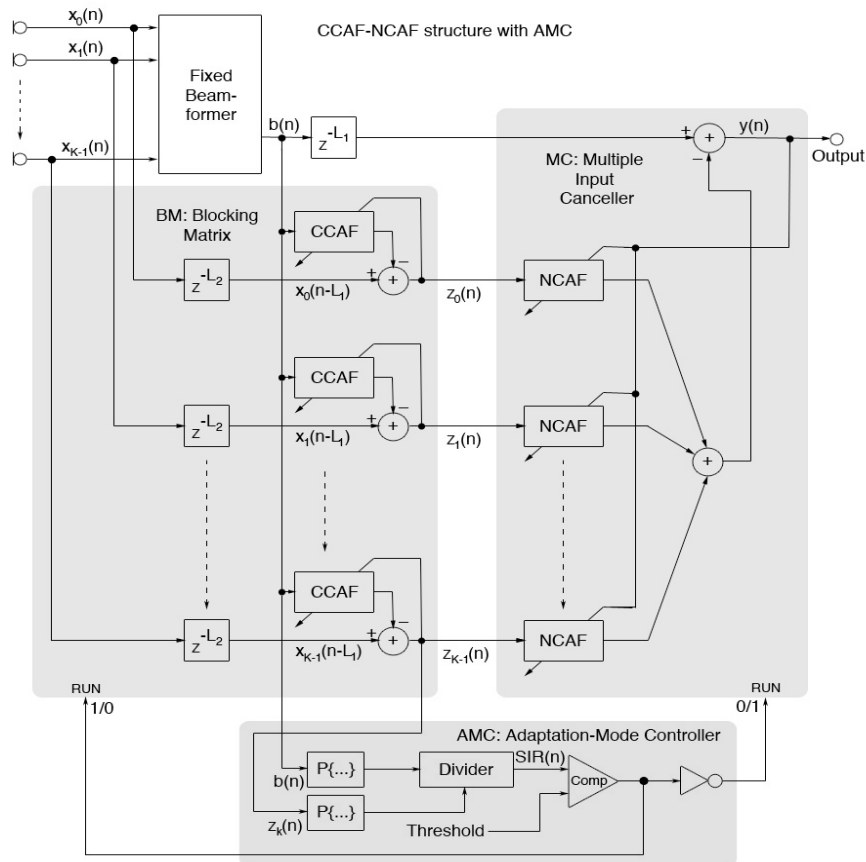


Abb. 4.16: G-J-B mit CCA-, bzw. NCA-Filtern und AMC

Ausgehend von der oben erklärten, mit dem Frost Beamformer verwandten Struktur wird in [1] ein sehr viel versprechender „*Robust Adaptive Beamformer*“ (RAB) als Variante des G-J-B vorgestellt. Angefangen bei der Erweiterung der Blocking Matrix mit adaptiven Filtern auf K Kanäle, führt [1] über verschiedene Variationen der Filtertypen in beiden Adaptionstufen hin zu einer Struktur, die in der BM mit K so genannten „*Coefficient Constrained Adaptive Filters*“ (CCAF) und im Multiple Input Canceller mit ebenfalls K „*Norm Constrained Adaptive Filters*“ (NCAF) arbeitet (s. Abb. 4.16). CCA-Filter verhalten sich wie adaptive Rauschunterdrücker und ermöglichen einen größeren Bereich für ein Null-Steering in Target Richtung. NCA-Filter reduzieren unerwünschte Auslöschungen des Target Signals, wenn die Eingänge des MCs Reste davon enthalten. Das Eingangssignal von jedem CCAF ist der Output des fixen Beamformers $b(n)$. Der k -te Ausgang jedes CCA-Filters wird vom mit um L_2 Samples verzögerten Mikrofonsignal $x_k(n)$ subtrahiert. Die Verzögerung der Mikrofonsignale wird benötigt, um die Kausalität des Systems beizubehalten. Als Ausgang eines jeden Pfades der Blocking Matrix ergibt sich so das Referenzsignal $z_k(n)$. Die J Koeffizienten der CCA-Filter $\mathbf{h}_k(n)$ werden mit Beschränkungen adaptiert. Die Adaption des dem Filter zugrunde liegenden NLMS Algorithmus kann man wie folgt beschreiben:

$$z_k(n) = x_k(n - L_2) - \mathbf{h}_k^T(n) \mathbf{b}(n) \quad \text{mit} \quad k = 1 \dots K \text{ Sensoren} \quad (4.55)$$

$$\mathbf{b}(n) = [b(n), b(n-1), \dots, b(n-J+1)]^T \quad (4.56)$$

$$\mathbf{h}_k(n) = [h_{k,0}(n), h_{k,1}(n), \dots, h_{k,J-1}(n)]^T \quad (4.57)$$

$$\mathbf{h}'_k(n+1) = \mathbf{h}_k(n) + \alpha \frac{z_k(n)}{\mathbf{b}(n)^T \mathbf{b}(n) + \varepsilon} \mathbf{b}(n) \quad (4.58)$$

$$\mathbf{h}_k(n+1) = \begin{cases} \mathbf{\Xi}_k, & \text{für } \mathbf{h}'_k(n+1) > \mathbf{\Xi}_k \\ \mathbf{\Psi}_k, & \text{für } \mathbf{h}'_k(n+1) < \mathbf{\Psi}_k \\ \mathbf{h}'_k(n+1), & \text{sonst} \end{cases} \quad (4.59)$$

$$\mathbf{\Xi}_k \hat{=} [\Xi_{k,0}, \Xi_{k,1}, \dots, \Xi_{k,J-1}]^T \quad (4.60)$$

$$\mathbf{\Psi}_k \hat{=} [\Psi_{k,0}, \Psi_{k,1}, \dots, \Psi_{k,J-1}]^T \quad (4.61)$$

$\mathbf{h}'_k(n)$ ist ein temporärer Koeffizientenvektor, der durch eine Beschränkungsabfrage geschickt wird. $\mathbf{\Xi}_k$ und $\mathbf{\Psi}_k$ sind die oberen und unteren Limits für die Filterkoeffizienten. α ist die Schrittweite der Adaption und ε ist eine kleine Konstante, nahe an 0, die eine Division durch Null verhindert. Alle Komponenten in $z_k(n)$, die mit $\mathbf{b}(n)$ korrelieren, werden durch die CCA-Filter ausgelöscht. Die Koeffizienten jedes CCA-Filters variieren charakteristisch abhängig von der Einfallsrichtung des Signals. Beschränkt man die Koeffizienten auf eine obere und untere Schwelle, kann man verhindern, dass Signale außerhalb einer dadurch entstandenen Region in der Blockingmatrix minimiert werden. Durch die Coefficient Constraints kann man so den maximal tolerierbaren Verzögerungs-/ Look Direction Fehler bestimmen.

Nur wenn die Filterkoeffizienten innerhalb der „*Constrained Region*“ bleiben und somit das Target Signal aus der definierten DOA-Region einfällt, wird es am Ausgang der BM minimiert. Abb. 4.17 verdeutlicht diese Tatsache schematisch. Für eine erfolgreiche Minimierung des Target Signals müssten bei z.B. -30° Einfallsrichtung die Koeffizienten über die Schwellen hinauswachsen.

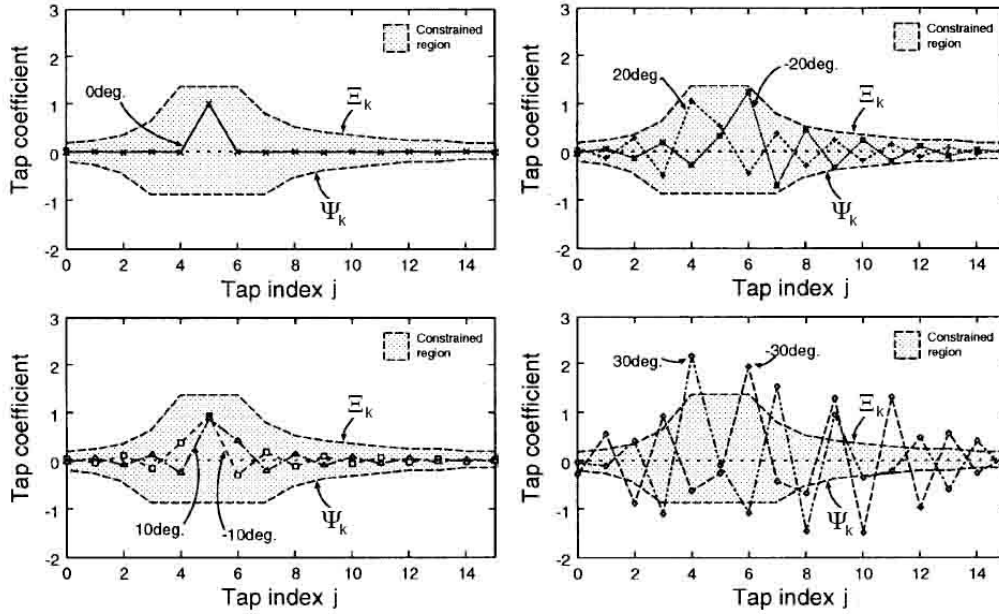


Abb. 4.17: Schematische Darstellung der Arbeitsweise von CCAFs [6]

Im Multiple Input Canceller befinden sich NCA-Filter, die alle mit $z_k(n)$ korrelierten Anteile vom verzögerten Ausgang $b(n - L_1)$ des fixen Beamformers subtrahieren (vgl. Abb. 4.16). Angenommen, man hat K Filter mit J Taps, dann lassen sich der Adaptionprozesse der NCAFs im MC mit $\mathbf{w}_k(n)$ als Koeffizientenvektor und $\mathbf{z}_k(n)$ als Filterload folgendermaßen beschreiben:

$$y(n) = b(n - L_1) - \sum_{k=0}^{K-1} \mathbf{w}_k^T(n) \mathbf{z}_k(n) \quad \text{mit } k = 1 \dots K \quad (4.62)$$

$$y(n) = \text{Ausgangssignal des RAB}, \quad (4.63)$$

$$\mathbf{w}_k(n) = [w_{k,0}(n), w_{k,1}(n), \dots, w_{k,J-1}(n)]^T \quad (4.64)$$

$$\mathbf{z}_k(n) = [z_k(n), z_k(n-1), \dots, z_k(n-J+1)]^T \quad (4.65)$$

$$\mathbf{w}'_k = \mathbf{w}_k(n) + \beta \frac{y_k(n)}{\mathbf{z}_k(n)^T \mathbf{z}_k(n) + \varepsilon} \mathbf{z}_k(n) \quad (4.66)$$

$$\Omega = \mathbf{w}_k^T \mathbf{w}'_k \quad (4.67)$$

$$\mathbf{w}_k(n+1) = \begin{cases} \sqrt{\frac{\Gamma}{\Omega}} \mathbf{w}'_k, & \text{für } \Omega > \Gamma \\ \mathbf{w}'_k, & \text{sonst} \end{cases} \quad (4.68)$$

β ist die Schrittweite der Adaption und w_k ist ein temporärer Vektor, der durch die Normbeschränkung eingeschränkt wird. Ω ist die quadratische Norm der Filterkoeffizienten und Γ ist der Threshold, oberhalb dessen die Filterkoeffizienten $w_{k(n+1)}$ durch eine Skalierung beschränkt werden. ε ist eine kleine Konstante, nahe an 0, die eine Division durch Null verhindert. Die Beschränkung der Norm verhindert das unkontrollierte Anwachsen der Filterkoeffizienten. Das Zurückhalten der Filterkoeffizienten hindert den Filter an der unerwünschten Auslöschung von Wunschsinal, das durch unzureichendes Blocken in der BM im MC dann ungewollt in die NCAFs gelangt. Im Idealfall ist eine Beschränkung der Norm nicht notwendig, wenn die BM jegliches Target Signal unterdrückt. In der realen Umgebung jedoch ist das aufgrund von Raumreflexionen und Hall vollkommen unmöglich. Deshalb ist eine Beschränkung der Norm der Filterkoeffizienten unbedingt notwendig. In [1] wird die CCAF-NCAF Struktur als eine robuste Lösung mit einer großen Toleranz zu Look Direction Errors bei einer kleinen Anzahl von Mikrofonen beschrieben.

Die Adaption der BM und des MC soll abwechselnd erfolgen. Die Beziehung zwischen angestrebtem Signal und Störsignal für die Blocking Matrix ist genau gegengleich zu der im Multiple Input Canceller. Für den adaptiven Prozess in der BM ist das Target das angestrebte und der Störer das unerwünschte Signal. Im MC hingegen ist der Störer das angestrebte und das Target das unerwünschte Signal.

Das bedeutet, dass die Blocking Matrix nur sinnvoll adaptiert, wenn Target Signal vorhanden ist und der MC, wenn kein Target vorhanden ist. Eine Adaption der BM, während kein Target vorhanden ist, kann zu einer unvollständigen Blockierung des Wunschsinal führen. Die Folge daraus ist eine mögliche Auslöschung des Target durch den MC. Der MC darf prinzipiell niemals adaptieren, wenn die BM nicht ausreichend adaptiert ist. Deswegen ist es am einfachsten und robustesten, den MC nur dann adaptieren zu lassen, wenn kein Wunschsinal vorhanden ist. Um zwischen der Aktivität oder Inaktivität eines Target Signals unterscheiden zu können ist ein „*Adaptation Mode Controller*“ (AMC) notwendig (s. Abb. 4.16 unten).

Die AMC kontrolliert die Adaption der BM und des MC mittels Wunschsinalerkennung über eine Schätzung der „*Signal to Interference Ratio*“ (SIR). Die SIR wird über ein Verhältnis der Leistung $P_b(n)$ des Ausgangssignals des fixen Beamformers $b(n)$ zu der Leistung $P_z(n)$ eines Ausgangs der BM $z_k(n)$ abgeschätzt. Die Hauptkomponente im Signal des fixen Beamformers ist das Target Signal und der Ausgang der BM sollte vorrangig Störsignal liefern. Deshalb kann das Leistungsverhältnis $s(n)$ als direkter Schätzer für die SIR betrachtet werden. Übersteigt das Verhältnis einen Threshold Q , dann adaptiert die BM, andernfalls adaptiert der MC. Die Berechnung des AMC wird in [5] beschrieben:

$$\text{run_BM} = \begin{cases} 1 & \text{für } s(n) \geq Q \\ 0 & \text{sonst} \end{cases} \quad (4.69)$$

$$\text{run_MC} = \begin{cases} 1 & \text{für } s(n) < Q \\ 0 & \text{sonst} \end{cases} \quad (4.70)$$

$$s(n) = \frac{P_b(n)}{P_z(n)} \quad (4.71)$$

$$P_b(n) = (1 - \gamma)P_b(n-1) + \gamma b^2(n) \quad (4.72)$$

$$P_z(n) = (1 - \gamma)P_z(n-1) + \gamma z_k^2(n) \quad (4.73)$$

4.5.3 Simulationen

Der Beamformer wurde mit der vorgestellten 6 Elemente Geometrie in MATLAB simuliert. Zuerst müssen die obere und untere Grenze für die Koeffizienten in den CCAFs bestimmt werden. Als zulässige Region wird in etwa ein Kreis festgelegt, der $\pm 15^\circ$ Azimuth und Elevation durchläuft. Um die Beschränkungen für die CCAFs zu erhalten, lässt man für die in Abb. 4.18 dargestellten Steerings nur die Blocking Matrix adaptieren und zieht bei jedem Filter eine Linie über die Maxima (Ξ_k) und eine Linie unter die Minima (Ψ_k) aller entstanden Filterkoeffizienten (Abb. 4.18). Als Eingangssignal wurden 3s eines zwischen 200Hz und 7,4kHz bandbegrenzten, gaußverteilten, weißen Rauschen verwendet. Die Schrittweite α wurde mit 0,2 gewählt und die Abtastrate beträgt 16kHz.

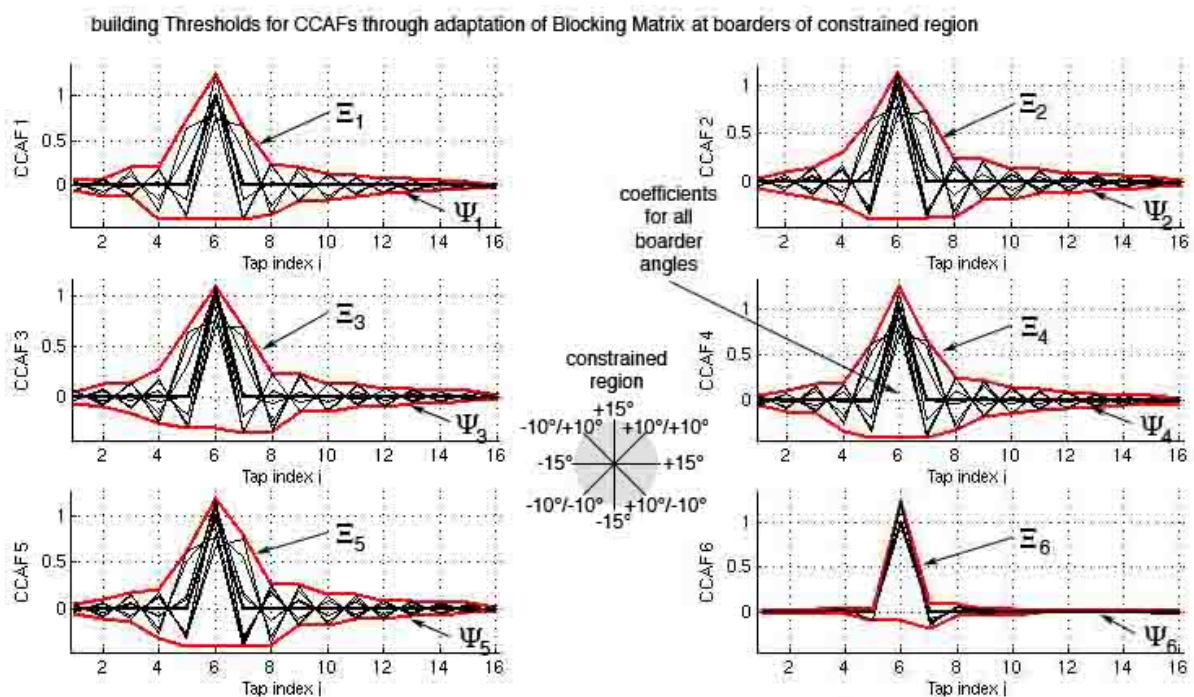


Abb. 4.18: Darstellung der Koeffizientenlimits und der zulässigen Target Region (MATLAB)

Um die Leistungsfähigkeit des Beamformers darstellen zu können, bringt man für alle Einfallsrichtungen einer Quelle nacheinander die Blocking Matrix und dann den Multiple Input Canceller zur Konvergenz. Befindet sich die Quelle innerhalb der Constrained Region, die durch die Coefficient Constraints vorgegeben wurde, so wird das Signal zum größten Teil von der BM geblockt und der MC wird keine Anteile aus dem Ausgangssignal entfernen. Mit wachsendem Winkel der Quelle wandert diese aus der Constrained Region heraus. Die BM kann immer weniger Anteil des Signals blocken, was zur Folge hat, dass der MC immer mehr am Ausgang abzieht.

Dadurch entsteht ein Richtplot, der die prinzipielle Leistungsbereitschaft der Filter im robusten adaptiven Beamformer (RAB) darstellt. Simuliert wurde mit, zwischen 200Hz und 7,4kHz bandbegrenztem, gaußverteilterm weißen Rauschen mit einer Dauer von 6s. Jede Stufe hat genau 3s Zeit zu adaptieren. Zuerst adaptiert die Blocking Matrix mit einer Schrittweite $\alpha = 0,2$ unter den oben festgelegten Constraints, dann der Multiple Input Canceller mit einem Threshold $\Gamma = 10$ für die Filternorm, ebenfalls mit einer Schrittweite $\beta = 0,2$. Vorherige Simulationstests haben gezeigt, dass die Algorithmen mit diesen Parametern bei einer Sampling Frequenz von 16kHz genug Zeit haben um vollständig zu adaptieren. Die Simulation wird nur in Azimutrichtung für die Winkel 0° bis 360° in 5° Schritten durchgeführt. Der Adaptaion Mode Controller wurde nicht in die Simulation mit einbezogen. Abb. 4.19 zeigt das Verhalten des Bamformers bei unterschiedlichen Koeffizientenbeschränkungen der CCAFs. Eine Kurve zeigt den Richtplot bei genau den in Abb. 4.18 dargestellten Schranken und die andere Kurve stellt das Verhalten dar, wenn dieselben Schranken einfach um 0,5 erhöht, beziehungsweise erniedrigt wurden. Die Simulation mit den lockereren Schranken wurde nur von 0° bis 180° durchgeführt und dann an der y-Achse gespiegelt, da jede Simulation dieser Art sehr viel Zeit beansprucht und man, vorausgesetzt es existiert kein Presteering, bei der verwendeten Arraygeometrie davon ausgehen kann, dass sich das Beampattern in der positiven Halbebene von 0° bis 180° gleich wie in der negativen von 0° bis -180° verhält.

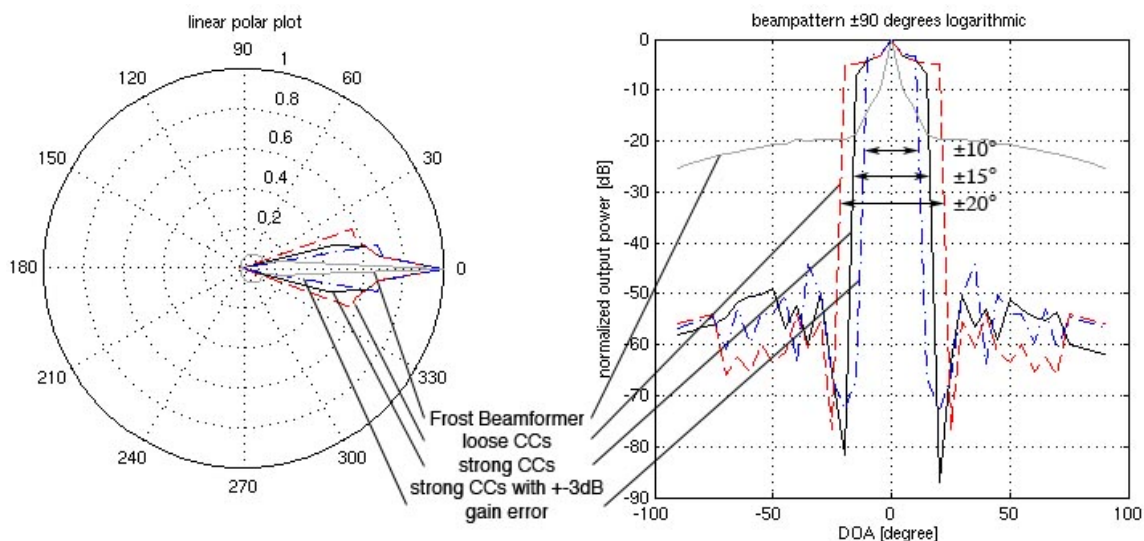


Abb. 4.19: Richtplot des RAB mit untersch. Coeff. Constr. und mit Verstärkungsfehler (MATLAB)

In Abb. 4.20 ist die Arbeitsweise des RAB dargestellt, wenn er durch einen Adaptation Mode Controller gesteuert wird. Die Simulation hat mit Sprachsignal aus 0° als Target und weißem Rauschen als Störer aus -30° bei einer SIR von 10dB stattgefunden. α und β wurden beide auf 0,2 gesetzt. Die Filter-Thresholds sind unverändert zu denen in der vorherigen Simulation (strenge Koeffizientenbeschränkung, $\Gamma=10$). Das oberste Diagramm zeigt Eingangs-, Ausgangssignal und die Schaltvorgänge des AMC. Im zweiten Plot wird der lineare Schätzer der SIR zusammen mit dem Threshold $Q = 5$ dargestellt. Plot 3 und 4 zeigen den Ausgang eines BM-Kanals und des MC.

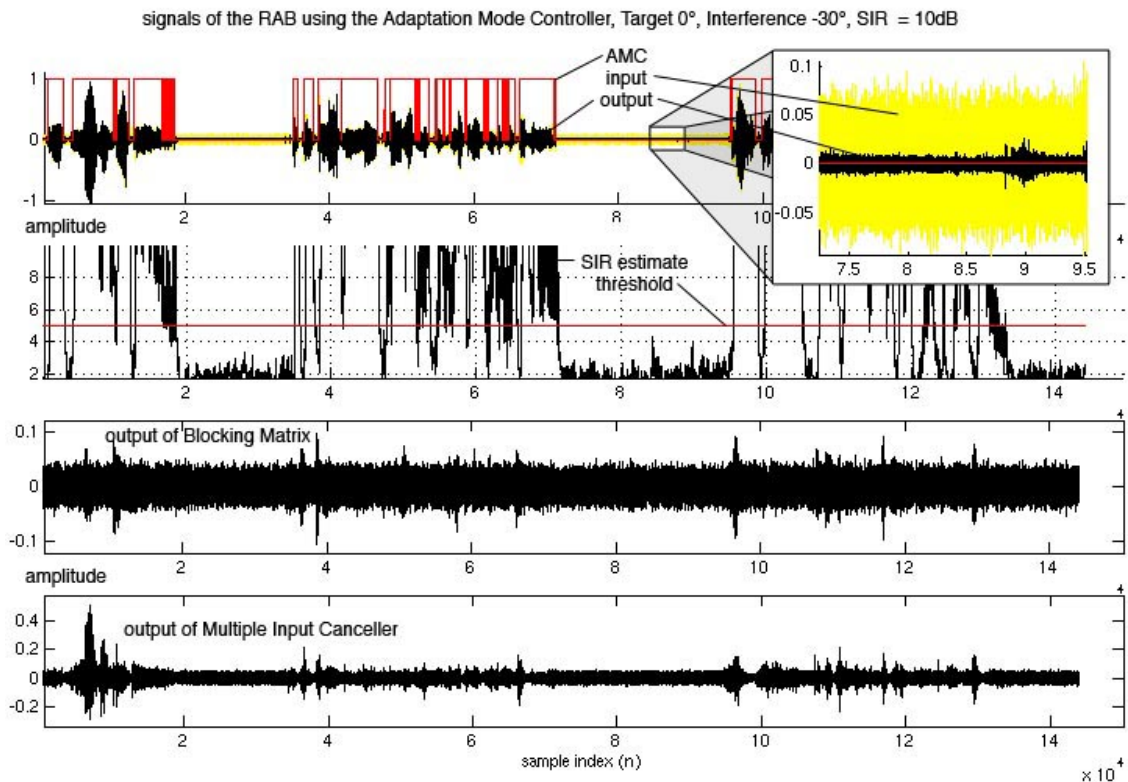


Abb. 4.20: Darstellung der Arbeitsweise des RAB mit AMC (MATLAB)

Abb. 4.21 stellt SIR Schätzungen bei verschiedenen Einfallsrichtungen von Störsignalen dar. Die Simulationsparameter sind die gleichen wie in Abb. 4.20 verwendet wurden. Das obere Diagramm zeigt die Schätzung für ein Target aus 0° und eine 10dB schwächere Interferenz aus -40° . Das untere Diagramm zeigt die Schätzung für eine Quelle aus -20° .

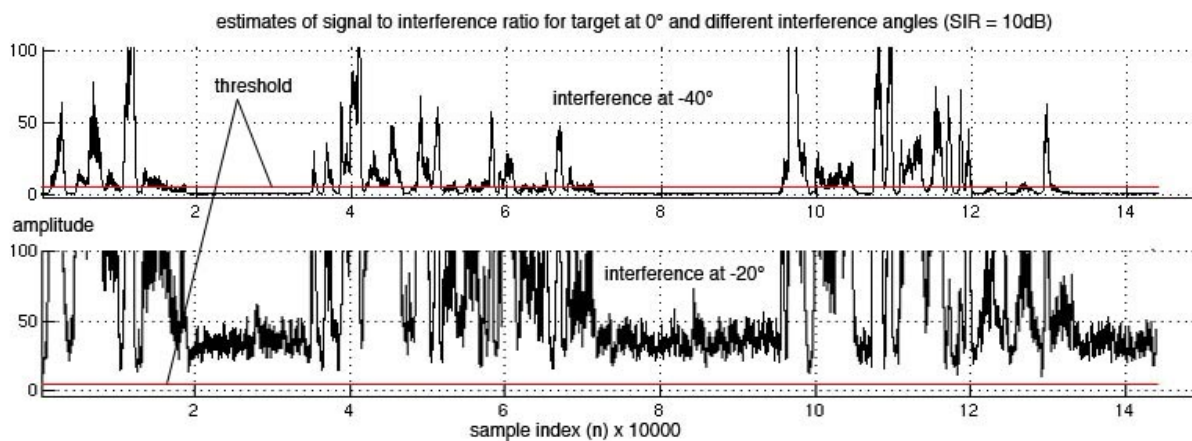


Abb. 4.21: Schätzungen der SIR bei verschiedenen Einfallswinkeln der Störsignale (MATLAB)

Abb. 4.22 zeigt das Verhalten des AMC bei Sprachsignalen als Target und Interference. Die Simulationsparameter haben sich im Vergleich zu den vorherigen Simulationen mit dem AMC nicht verändert. Nur das Signal zu Rausch Verhältnis wurde auf 3dB abgesenkt.

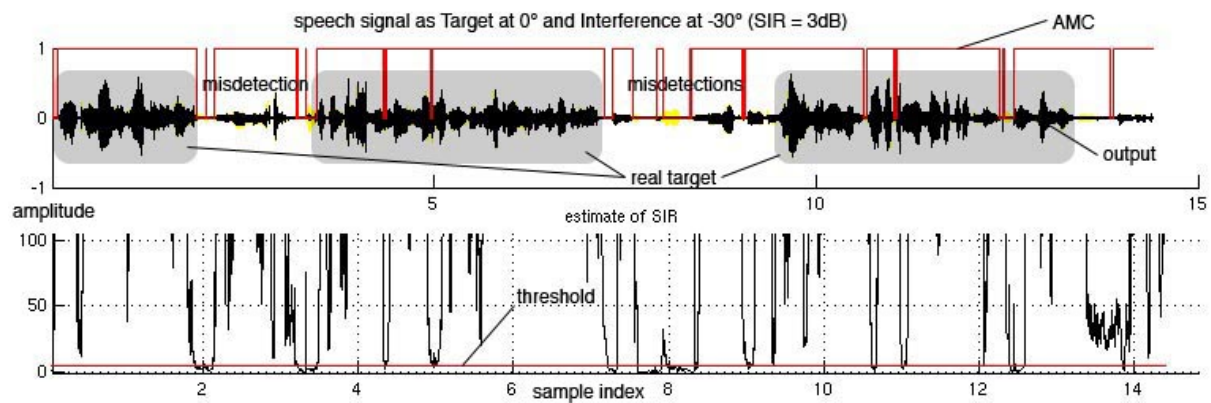


Abb. 4.22: Verhalten des AMC bei Sprachsignal aus -30° als Störer, 3dB SIR (MATLAB)

4.5.4 Auswertung und Vergleich der Simulationen

Man kann deutlich erkennen, dass dieser Ansatz der bisher leistungsstärkste ist. Die Simulationen zeigen mit der verwendeten Geometrie eine Dämpfungsbereitschaft für Störsignale von etwa 50dB. Wie die Implementierung in PD gezeigt hat, ist diese Performance jedoch nur in der Simulation zu erreichen. Gut sichtbar ist, dass sich der Akzeptanzbereich für Targets mit dem Aufweiten der Koeffizientenbeschränkungen in den CCAFs auch erweitert ($\pm 15^\circ$ mit den oben ermittelten Thresholds und $\pm 20^\circ$ mit lockereren Thresholds). Der Algorithmus ist sehr robust gegenüber von Positionierungsfehlern, da diese durch die Blocking Matrix ausgeglichen werden. Auch Verstärkungsfehler an den Sensoren nehmen nur geringen Einfluss auf das Beampattern, da die verwendeten Filter im Gegensatz zu allen Frost-Ansätzen unabhängig voneinander arbeiten können und keiner harten Frost Constraint unterliegen. In Abb. 4.19 lässt sich qualitativ bei einem Verstärkungsfehler von ± 3 dB nur eine Verengung der für das Target zugelassenen Region auf $\pm 10^\circ$ erkennen.

Der Adaptation Mode Controller scheint auf den ersten Eindruck gut zu funktionieren. Prinzipiell ergibt sich aus den Schätzern eine höhere SIR wenn Target vorhanden ist als wenn es nur Störsignal gibt. Somit ist für den Einzelfall der Beamformer über diese Methode erst einmal steuerbar. Problematisch wird es jedoch, wenn man versucht den Threshold für den AMC über einen größeren Bereich konstant zu lassen. In Abb. 4.21 kann man sehen, dass die SIR mit unterschiedlichen Einfallswinkeln von Störern einen Offset aufweist. Das bedeutet, dass sich der Level des Rauschpegels ändert, obwohl sich die echte SIR (konstant auf 10dB) nicht verändert. Es ist also unmöglich, mit einem einzigen Threshold die Funktionalität des AMC zu garantieren. Im oben gezeigten Fall übersteigt der Ruhepegel der SIR bei einem Einfallswinkel von -20° den Threshold. Es adaptiert nur noch die BM. Der MC bleibt im Stillstand, obwohl nicht permanent Wunschsignal vorhanden ist. Unter diesen Umständen wird der Beamformer funktionslos. Den einzigen Gewinn kann dann nur der fixe Beamformer liefern. Dieses Phänomen ist damit zu begründen, dass sich, abhängig von der Einfallrichtung die Leistung der Signale am Ausgang der Blocking Matrix und somit der Schätzer für die Störleistung verändert. Wandert eine Störquelle in Richtung der zugelassenen Region für Target Signale, dann werden aufgrund der richtungsbezogenen Frequenzabhängigkeit des fixen Beamformers immer mehr Anteile des Störers von der BM

geblockt. Da tiefe Frequenzen vom fixen D&S-B mit kleiner werdendem Azimutwinkel immer schlechter gedämpft werden, wird das Referenzsignal, das die BM erzeugt immer hochfrequenter. Die Leistung nimmt ab. Im fixen Beamformer hingegen ist in Situationen ohne Target das volle Spektrum und somit die volle Leistung des Störers, abzüglich des Gewinns des D&S-B, vorhanden. Wenn am Ausgang des fixen Beamformers der Störanteil und somit die Störleistung größer wird und gleichzeitig die Leistung des Ausgangssignals der BM geringer, dann steigt der Rauschpegel in der SIR-Schätzung und wird somit zu einer nicht repräsentativen Größe für eine Steuerung des AMC. Zusätzlich ist diese Methode zur Steuerung des Beamformers beschränkt auf Störsignale, die auf jeden Fall eine geringere Leistung als das Wunschsinal haben. Sind Störer und Target gleich laut, so ergibt sich kein Unterschied mehr zwischen beiden. In Abb. 4.22 ist ein Fall dargestellt, bei dem ein Sprecher aus 0° und gleichzeitig ein Sprecher aus -30° mit einem Rauschabstand von $+3\text{dB}$ in das Mikrofon sprechen. Der Sprecher aus -30° ist ganz klar außerhalb der Target Region und soll deswegen nicht übertragen werden. Zwar wird er nicht von der Blocking Matrix geblockt, kann aber durch den MC nicht vom Ausgang abgezogen werden, da dieser nicht zur Adaption kommt, wenn der AMC nicht erkennt, dass es sich um Störsignal handelt.

Es gilt also den AMC in zwei Punkten zu verbessern. Die richtungsabhängigen Schwankungen der SIR-Schätzung müssen minimiert werden. Signale die gleich laut außerhalb der zugelassenen Region einfallen, sollen vom AMC auch als Störer erkannt werden.

4.5.5 Verbesserung des AMC

Zuerst wird versucht die richtungsabhängigen Schwankungen der SIR-Abschätzung zu minimieren. Um die vorkommenden Werte der SIR besser im Überblick zu haben wird der Schätzer fortlaufend nun in dB angegeben.

Um Schwankungen zu minimieren, wurde das Verhalten der SIR für verschiedene Winkel und Rauschabstände betrachtet. Es war zu beobachten, dass die Filterenergie der CCAFs mit steigendem Winkel zunimmt und somit das SIR abnimmt. Abb. 4.23 zeigt den richtungsabhängigen Rauschpegel, der SIR-Schätzung. Dazu wurde nacheinander aus den Richtungen -90° bis 90° in 5° Schritten bandpassgefiltertes, gaußverteiltes weißes Rauschen durch den Beamformer geschickt und nach der Adaption jeweils die Leistung von P_z und P_b aufgenommen. Die linke Hälfte zeigt die beiden Leistungsverläufe, die rechte Hälfte zeigt die SIR-Schätzung, die in diesem Fall eine Schätzung für den Rauschpegel ist, da es kein Target Signal gibt.

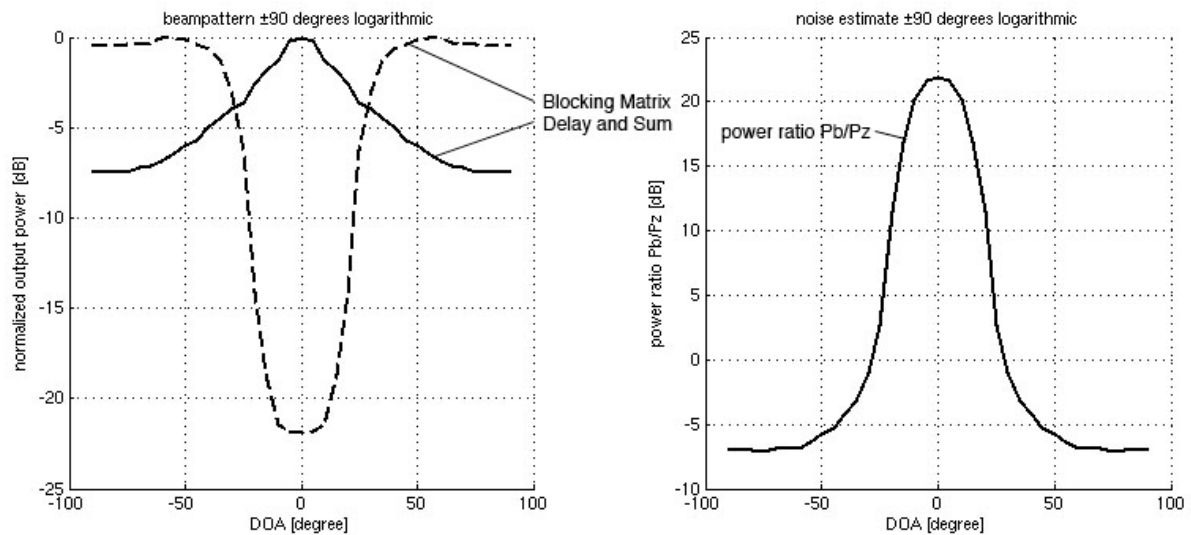


Abb. 4.23: Schätzung des Rauschlevels durch P_z und P_b bei untersch. Winkeln (MATLAB)

Es gibt die Möglichkeit, einen zusätzlichen Threshold S einzufügen, der die SIR, abhängig vom Abstand der Leistung P_z vom Threshold, um die Differenz zwischen P_z und Threshold verschiebt. Dadurch können die SIR-Schwankungen in einem kleineren Bereich gehalten werden. Wenn die Leistung von P_z über diesen Threshold geht, dann wird die SIR-Schätzung um die Differenz nach unten verschoben, wenn P_z unter dem Threshold liegt, so wird die SIR um die Differenz nach oben verschoben. Prinzipiell geht es hier um einen Limiter/ Expander für die SIR-Schätzung, der über eine Sidechain - die Leistung P_z - gesteuert wird.

$$\text{SIR}'(n) = 10 \log \left(\frac{P_b(n)}{P_z(n)} \right) \quad (4.74)$$

$$\text{SIR}(n) = \begin{cases} \text{SIR}'(n) - |10 \log(P_z(n)) + S| & \text{für } 10 \log(P_z(n)) < S \\ \text{SIR}'(n) + |10 \log(P_z(n)) + S| & \text{für } 10 \log(P_z(n)) \geq S \\ \text{SIR}'(n) & \text{für } 10 \log(P_z(n)) < S_2 \end{cases} \quad (4.75)$$

Der Threshold S_2 ist nicht unbedingt notwendig. Er stellt nur eine zusätzliche Schwelle dar, ab der es für den Beamformer nicht mehr nötig ist, auf den MC zu schalten und die BM offen zu lassen, weil das Störgeräusch zu leise ist.

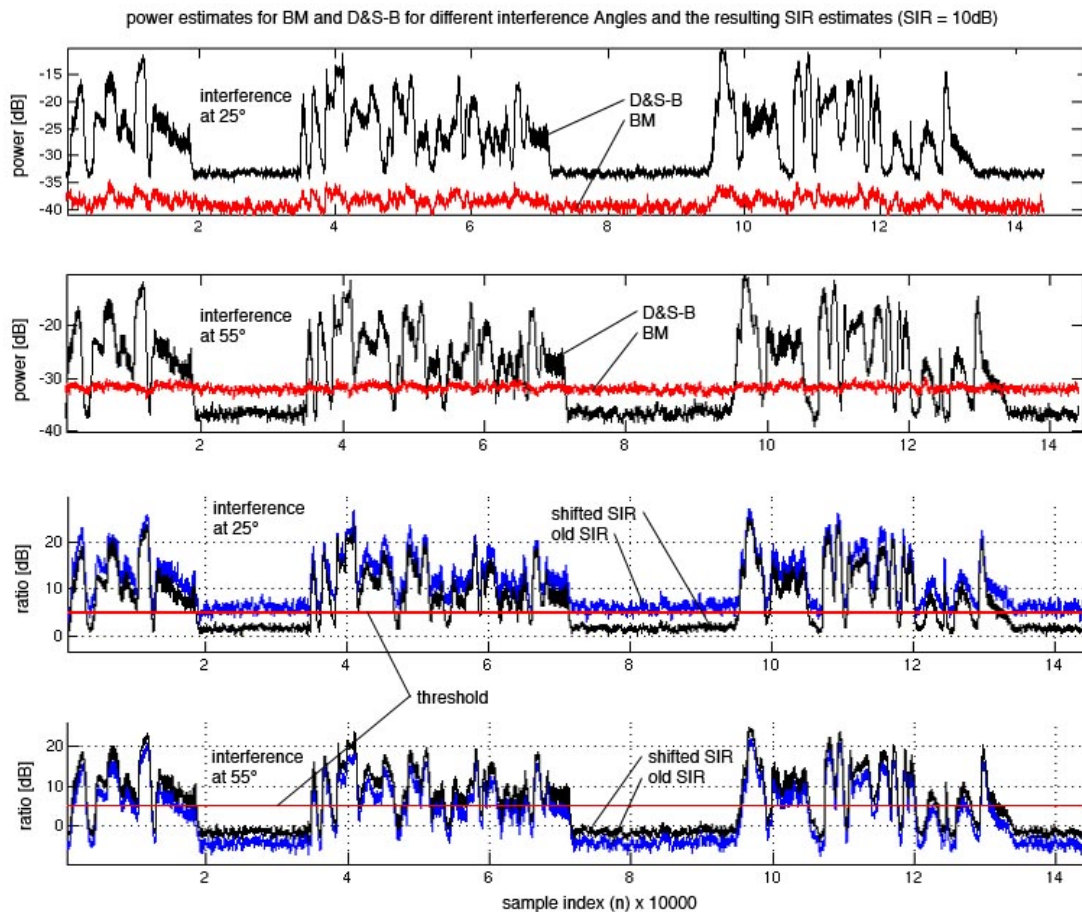


Abb. 4.24: Verschiebung des SIR-Schätzers durch Limiter/ Expander mit Sidechain (MATLAB)

Abb. 4.24 zeigt die Arbeitsweise der vorgeschlagenen Erweiterung des AMC. Simuliert wurde mit den gleichen Parametern wie zuvor in Abb. 4.21. Das Wunschsinal kam aus 0° . Die Einfallswinkel für die Störsignale wurden einmal auf 25° und einmal 55° gesetzt. Das entspricht etwa dem Punkt, ab dem Interference erkannt werden soll und dem Ende der Kurvenkrümmung in Abb. 4.23 rechts. S wurde mit -35dB gewählt und Q mit 5dB . Man kann deutlich sehen, wie im einen Fall die SIR nach unten und im anderen Fall nach oben korrigiert wird. Ohne Korrekturfunktion wäre im ersten Fall bereits Fehldetektion eingetreten.

Im zweiten Fall wird die SIR gerade noch so stark angehoben, dass es leichte Verbesserungen in der Target Erkennung gibt. Durch diese Methode werden starke SIR-Schwankungen vermindert. Um den Ansatz noch robuster zu machen, wird später in der Echtzeitrealisierung der Threshold Q dynamisch variiert, d. h. wenn die SIR unter Q fällt, dann folgt dieser der SIR im Abstand von 6dB , so lange, bis er durch einen plötzlichen Anstieg der SIR (\Rightarrow Target vorhanden) wieder „überholt“ wird. Nach Überschreitung durch die SIR bleibt er fix, bis die SIR wieder unter den Level fällt. Das setzt natürlich einen stationären oder sich nur langsam ändernden Rauschpegel voraus. Gleichzeitig muss der Threshold Q beim Einschalten des Systems natürlich oberhalb des Rauschpegels liegen, sonst funktioniert der AMC nicht. Ab einem gewissen Rauschpegel funktioniert dieser Ansatz prinzipiell nicht mehr, da es in der SIR-Abschätzung keinen ausreichenden Unterschied mehr zwischen Target und Interference gibt.

Um den AMC auch funktionstüchtig zu halten, wenn zwei Quellen gleicher Lautstärke vorhanden sind (eine Quelle als Target aus 0° und die andere als Interference), muss eine zusätzliche Entscheidungsstufe eingeführt werden, die es ermöglicht, Signale, die nicht aus der zulässigen Region kommen als Störer zu erkennen. Man geht davon aus, dass bei der Anwesenheit eines Targets die SIR auf jeden Fall groß genug ist, um über dem Threshold Q zu liegen. Gleichzeitig kann es passieren, dass die SIR über dem Threshold liegt, obwohl kein Target vorhanden ist. Das geschieht zum Beispiel, wenn ein Sprecher mit der gleichen Lautstärke aus einer anderen Richtung in das Mikrofon hineinspricht. Um diesen Fehler verhindern zu können wurde zusätzlich vor jeder Entscheidung des AMC eine „*Time Delay Estimation*“ (TDE) berechnet, die bei Existenz eines Targets in der zugelassenen Region ein positives Ergebnis liefert (1 für positiv, 0 für negativ). Das bedeutet, die Blocking Matrix wird nur dann zur Adaption frei gegeben, wenn SIR und TDE positive Werte haben. Es hat sich in den Simulationen und in der Echtzeitumgebung allerdings gezeigt, dass ein TDE-Filter bei Aperturen mit solch kleinen Abmessungen wie sie hier verwendet werden, keine zuverlässigen Abschätzungen liefern kann (verwendet wurde jener TDE-Filter aus [3]). Zwar ist es näherungsweise möglich, die Richtung des Störers zu bestimmen, wenn dieser aus Rauschen besteht und somit einen eindeutigen Peak in der Kreuzkorrelation zwischen 2 Sensoren aufweist. Jedoch ist es nicht gelungen über diese Methode das Target identifizieren zu können, da es meistens gleichzeitig mit dem Störer auftritt und der TDE-Filter bei gleicher Lautstärke keine Entscheidung mehr treffen kann. Aber auch bei leiseren Störsignalen ist es schwierig, weil der Filter sich nicht sehr gut auf Sprache adaptiert. Das führt zu häufigen Auslöschungen des Wunschssignals, weil der AMC auch bei der Existenz eines Targets auf den MC schaltet. Deswegen wurde von dieser Methode wieder abgesehen.

Es gibt eine sehr einfache Methode, über die man auch bei kleinen Aperturen, unabhängig vom Signal sehr gut die Einfallsrichtung des Signals bestimmen kann. Aufgrund der verwendeten Geometrie ist es möglich, Beams in alle Richtungen mit etwa gleichen Eigenschaften auszubilden. Belegt man das Array zusätzlich zum Hauptbeamformer noch mit vier D&S-B, die mittels eines Presteerings nach oben, unten, links und rechts gerichtet sind, kann man die aufgenommenen Leistungen des Hauptbeams und der zusätzlichen Nebenbeams zueinander ins Verhältnis setzen und erhält so die „*Steered Response Power Ratio*“ (SRPR). Je kleiner eines dieser Verhältnisse wird, desto weiter ist man aus dem Zentrum entfernt und umgekehrt. Abb. 4.25 verdeutlicht den Ansatz. Vier Seitenbeams ergeben vier SRPRs. Damit erhält man eine zusätzliche Entscheidungsstufe für den AMC. Die Blocking Matrix darf nur noch adaptieren, wenn die SIR oberhalb des Thresholds Q liegt und gleichzeitig alle Werte der SRPR innerhalb der Constrained Region, das bedeutet unterhalb eines Thresholds B , liegen. In allen anderen Fällen adaptiert der Multiple Input Canceller (MC).

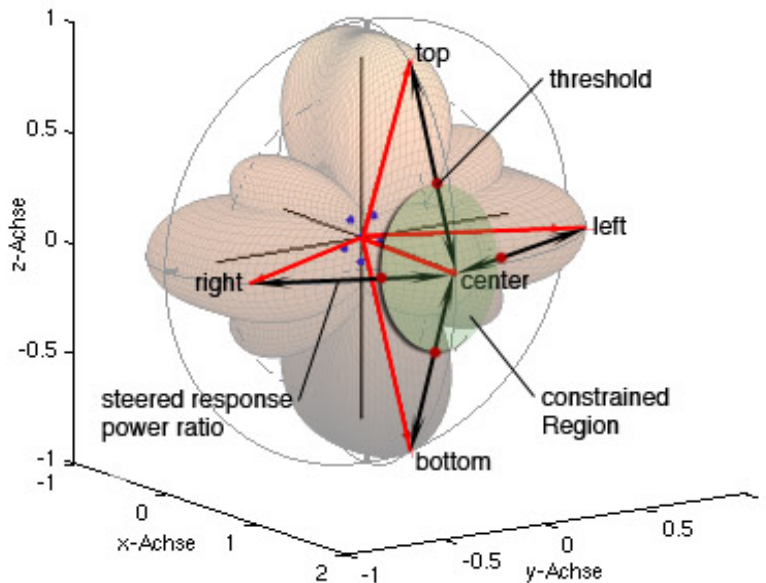


Abb. 4.25: Schema der SRPR mit 4 Seitenbeams und einem Hauptbeam

$$SRPR_{side}(n) = 10 \log \left(\frac{P_{center}(n)}{P_{side}(n)} \right) \quad \text{mit } side = left, right, top, bottom \quad (4.76)$$

$$P_{center}(n) = (1 - \zeta) P_{center}(n-1) + \zeta b_{center}^2(n) \quad (4.77)$$

$$P_{side}(n) = (1 - \zeta) P_{side}(n-1) + \zeta b_{side}^2(n) \quad (4.78)$$

mit $b_{side}(n)$ = Ausgang des jeweiligen Seitenbeams, $b_{center}(n)$ = Ausgang des Hauptbeams
und ζ = Aktualisierungskonstante, nahe bei 0

$$CENTER = \begin{cases} 1 & \text{für } SRPR_{side} < B \\ 0 & \text{sonst} \end{cases} \quad (4.79)$$

Diese Neuerung wurde nicht in MATLAB simuliert, sondern sofort in der Echtzeitumgebung implementiert. Es hat sich gezeigt, dass man eine höhere Genauigkeit bekommt, wenn man die Signale der einzelnen Beams nur in einem relativ schmalen Band betrachtet. Nachdem ein Array bei kleinen Aperturen zu tiefen Frequenzen hin kugelförmige Richtcharakteristik annimmt, wird die SRPR in diesem Bereich nur sehr schlecht auflösen. Als Steering Winkel für die Seitenbeams wurden $\pm 60^\circ$ gewählt, weil so direkt eine Nullstelle des Beams auf die 0° Richtung gelegt wird. Der betrachtete Frequenzbereich geht von 2kHz bis 7kHz. In diesem Bereich sind am wenigsten Nebenkeulen zu finden (vgl. Abb. 4.26). Den Verlauf einer SRPR kann man in Abb. 4.27 sehen. Dafür wurden die richtungsabhängigen Ausgangsleistungen eines D&S-B bei 0° und 60° Steering über bandbegrenzt (2kHz und 7kHz), gaußverteiltes weißes Rauschen ermittelt und anschließend nach (4.76) ins Verhältnis gebracht.

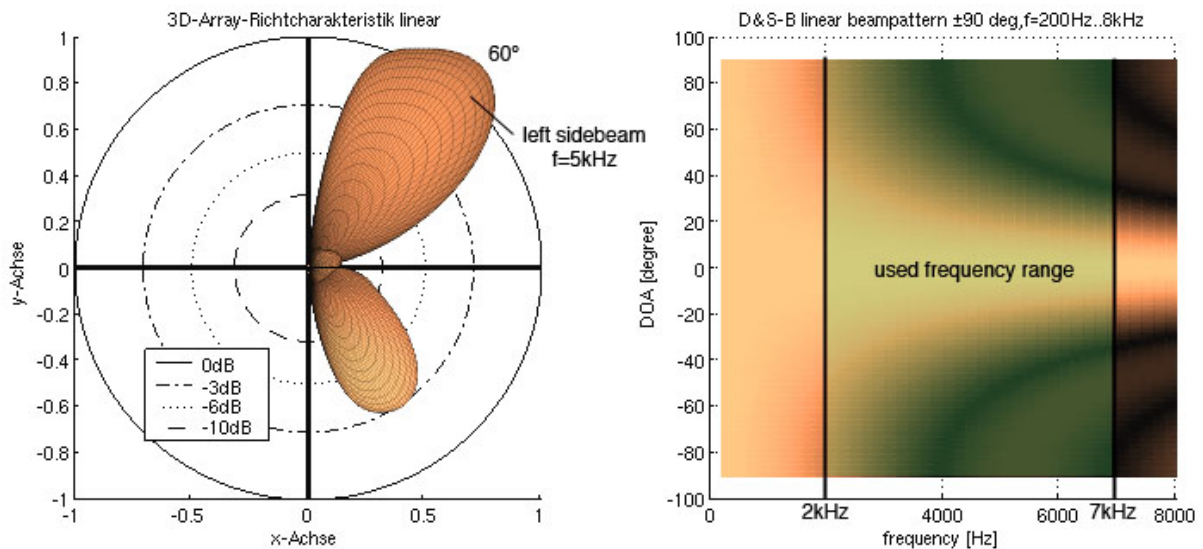


Abb. 4.26: linker Seitenbeam (li) und benutzter Frequenzbereich (re) des AMC (MATLAB)

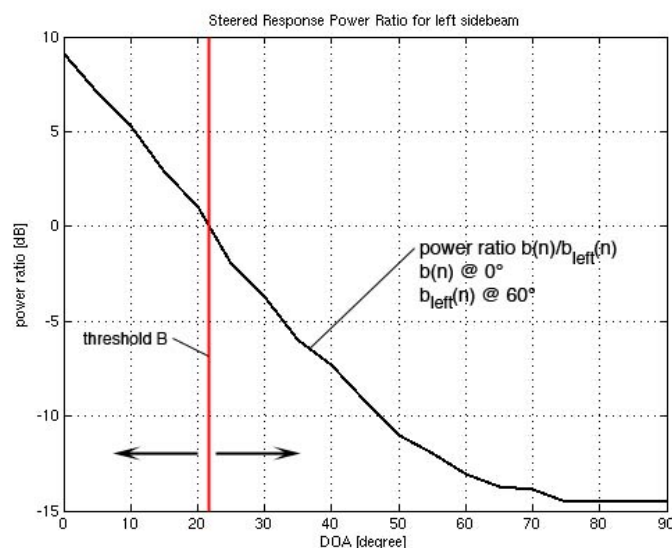


Abb. 4.27: Verlauf der $SRPR_{left}$ (MATLAB)

Durch diese Erweiterung wird es möglich, die Einfallsrichtung eines Signals stabiler zu identifizieren. Damit hat der AMC eine weitere zuverlässige Entscheidungsstufe. Leider ist die SRPR auch nicht in der Lage, immer ein eindeutiges Ergebnis zu liefern, wenn 2 Quellen gleichzeitig aus verschiedenen Richtungen mit gleicher Lautstärke auf den Beamformer einwirken. Ist eine Störquelle ungünstig positioniert (z. B. direkt in der Hauptkeule eines Seitenbeams), dann wird die betroffene SRPR nach außen gezogen. Auch in halligen Umgebungen kann es durch Reflexionen zu einer unsicheren Entscheidung kommen. Unsicherheiten sind dadurch definiert, dass während Adaption der Blocking Matrix die Variable CENTER auf 0 springt, die SIR aber weiterhin über dem Schwellenwert bleibt. Bei der bisherigen Umsetzung würde der AMC sofort auf MC schalten und es käme zu Target Auslöschungen (vorausgesetzt es ist wirklich Target da).

Um solchem Verhalten entgegenwirken zu können, wird der AMC noch um eine letzte Entscheidungsstufe erweitert. Diese soll, im Falle einer Unsicherheit, die Adaption des Beamformers anhalten („*No Operation*“) und entweder bei rechtzeitiger Rückkehr von der SRPR in den erlaubten Bereich innerhalb von B die BM wieder starten oder im anderen Fall nach einer gewissen Zeitdauer auf den MC schalten. Abb. 4.28 zeigt ein Blockschaltbild des AMC mit allen Entscheidungsstufen.

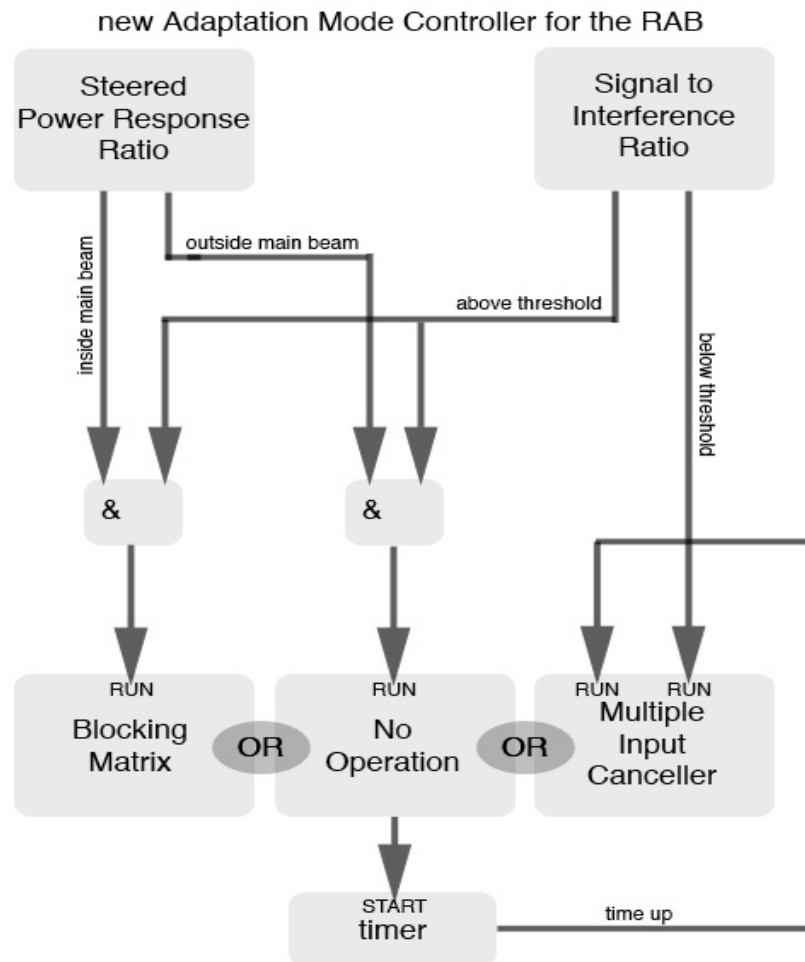


Abb. 4.28: verbesserter Adaptation Mode Controller

Mit dem neu vorgestellten AMC wird es möglich nahezu alle Signale, die nicht in der Constrained Region liegen, als Interference zu identifizieren. Wie man an den folgenden Messergebnissen der akustischen Vermessung des Arrays erkennen kann, funktioniert das System auch noch bei einem Rauschabstand von 0dB mit zwei Sprachsignalquellen.

4.5.6 Realisierung in PD und akustische Vermessung

Der RAB wurde mit dem im vorigen Kapitel vorgestellten AMC in PD realisiert. Zusätzlich ist noch eine Normalisierung der einzelnen Sensorverstärkungen bezogen auf einen Referenzsensor (Mikrofon 6) eingebaut worden, um Verstärkungsfehlern und der Langzeitdrift der Vorverstärker entgegenzuwirken. Dafür wurden die Leistungen an allen Sensoren in dB überwacht und die Verstärkung in dB der Mikrofone 1 bis 5 jeweils um die Differenz zu Sensor 6 korrigiert. Dieser Vorgang wird alle 5 Sekunden durchgeführt. Die Leistung wird, wie schon in den vorherigen Kapiteln, über einen Tiefpass gemittelt bestimmt (s. z. B. Gl. 4.77). Der Beamformer wurde mit beiden in Kapitel 5 beschriebenen Methoden vermessen. Methode 1 dient dabei zur Leistungsbeurteilung der Filter und zum Vergleich mit den durchgeführten Simulationen. Methode 2 soll die Leistung des neuen AMC darstellen. Abb. 4.29, Abb. 4.30 und Abb. 4.31 sind durch die Vermessung mit Methode 1 entstanden. Während das Messsignal abgespielt wurde, ist zuerst für 7 Sekunden die Adaption der Blocking Matrix und dann für 11 Sekunden des Multiple Input Cancellers erzwungen worden. Die Plots sind in ihrer Entstehung mit den Simulationen aus Abb. 4.19 zu vergleichen.

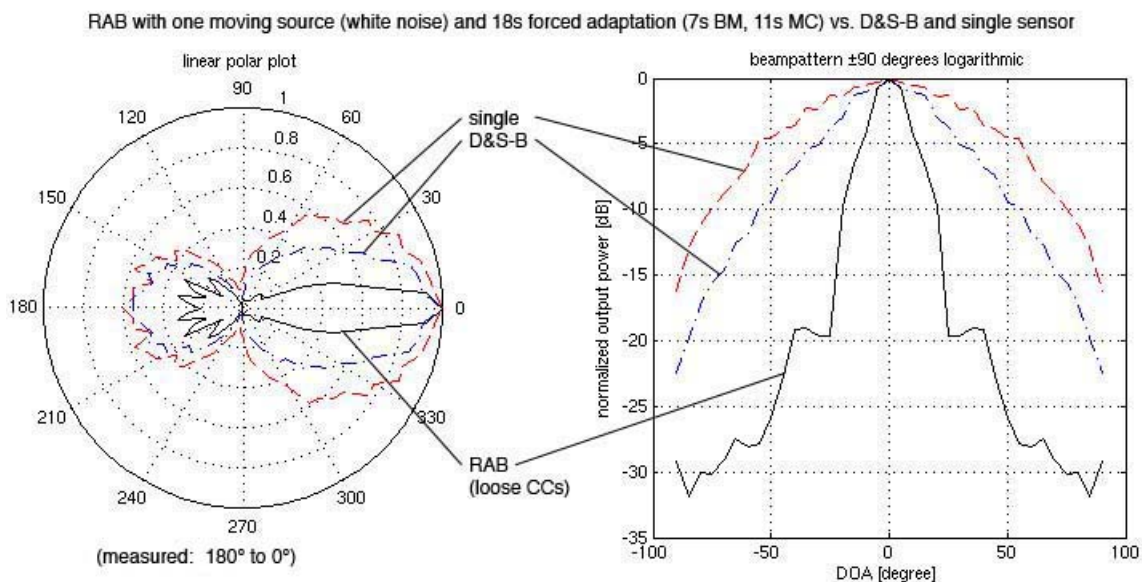


Abb. 4.29: Leistungsplots von Einzelmikrofon, D&S-B, RAB

In Abb. 4.29 kann man den großen Gewinn bezüglich der Dämpfung außerhalb des Target Bereichs des RAB im Vergleich zum Einzelmikrofon und D&S-B erkennen. Gemessen wurde mit den Koeffizientenbeschränkungen aus Kapitel 4.5.3 für die CCAFs und einer Normbeschränkung der NCAFs auf $\Gamma = 10$, die Schrittweiten waren auf $\alpha = 0,02$ und $\beta = 0,004$ gesetzt. Die Messergebnisse können die Leistung der Simulation aus Abb. 4.19 nicht erreichen, zeigen aber die gleiche Tendenz und bestätigen somit die Robustheit des Algorithmus. Abb. 4.30 zeigt die Grenzen der Robustheit auf, wenn man die Coefficient Constraints enger macht. Auch hier kommt es zu Wunschsignalauslöschung und einem Sturz der Richtcharakteristik. In Abb. 4.31 sind der Richtplot und das Beampattern des RAB mit rosa Rauschen als Eingangssignal dargestellt.

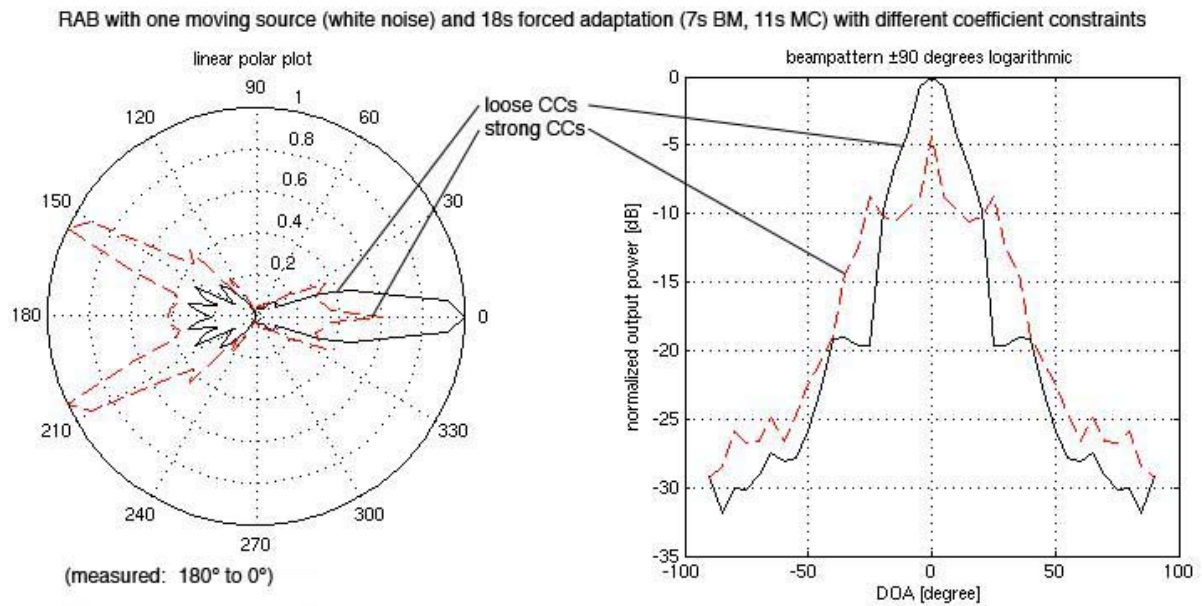


Abb. 4.30: gemessener Leistungsplot mit strenger/ lockerer Koeffizientenbeschränkung, weißes Rauschen

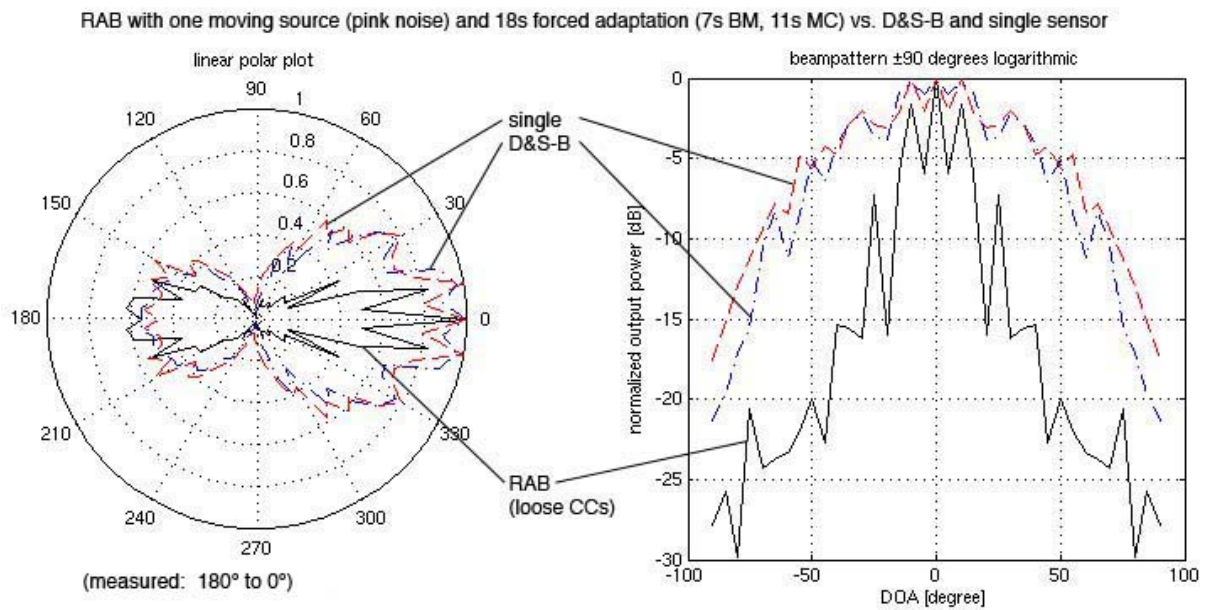


Abb. 4.31: Leistungsplot mit lockerer Koeffizientenbeschränkung, rosa Rauschen

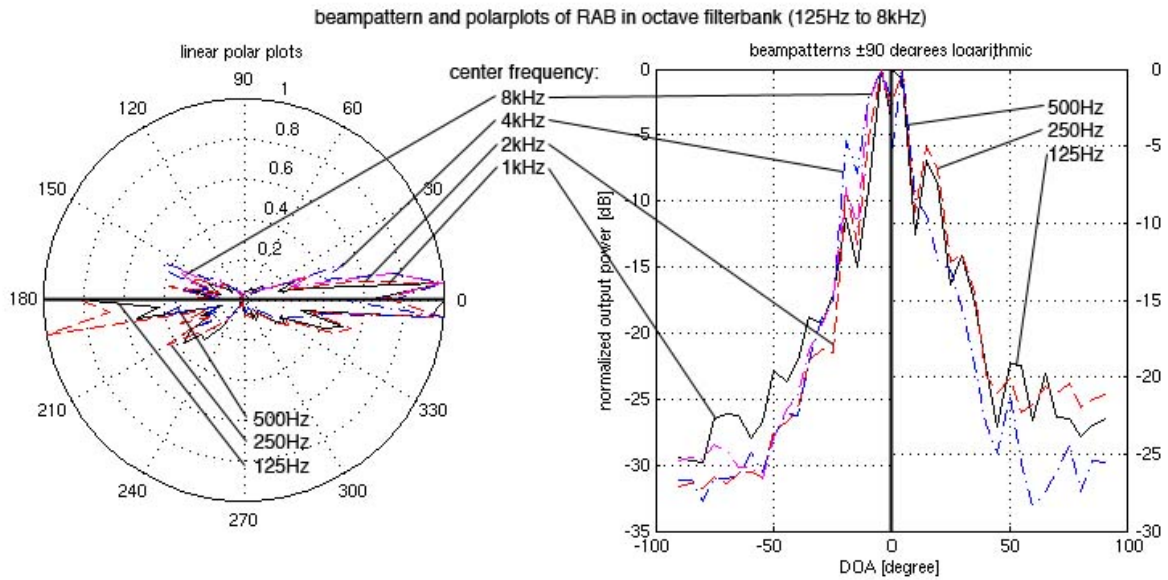


Abb. 4.32: RAB Beampattern für verschiedene Oktavbänder (125Hz bis 8kHz)

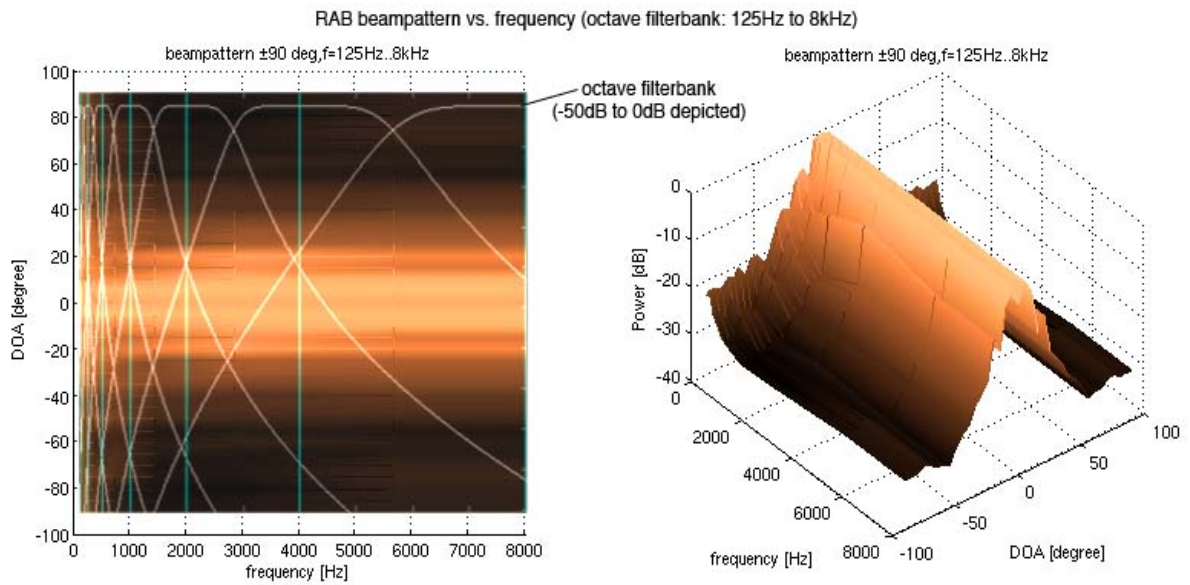


Abb. 4.33: Frequenzverlauf des RAB in Oktavbänder unterteilt

In Abb. 4.32 und Abb. 4.33 sind die Polardiagramme und Beampattern des RAB für verschiedene Frequenzbänder dargestellt. Die Diagramme sind unter der Verwendung von Messmethode 1 entstanden, jedoch wurde vor der Ermittlung der Ausgangsleistung das Ausgangssignal des RAB zuerst in sieben Oktavbänder von 125Hz bis 8kHz unterteilt. Die Oktavfilterbank ist mittels des Filter Design And Analysis Tools in MATLAB bestimmt worden. Jeder Filter ist ein Butterworth Bandpass sechster Ordnung der Direct Form II, Second Order Section. Die Grenzfrequenzen eines Oktavbandes ergeben sich stets am geometrischen Mittel (Faktor $\sqrt{2}$ für obere Grenzfrequenz und $1/\sqrt{2}$ für untere Grenzfrequenz) zwischen den Mittenfrequenzen zweier benachbarter Oktavbänder. Wie zu erkennen ist, bleibt die Richtcharakteristik des Beamformers über den gesamten Frequenzbereich annähernd konstant. Tiefe Frequenzen (bis 500Hz) haben eine geringfügig schlechtere Stördämpfung als hohe (ab 500Hz).

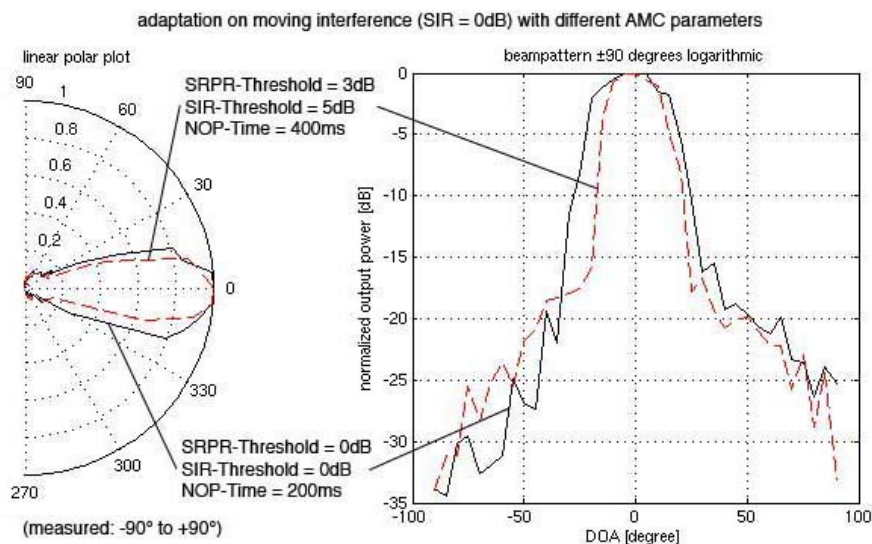


Abb. 4.34: Leistungsplot bei eingeschaltetem AMC mit versch. Steuerparametern, weißes Rauschen

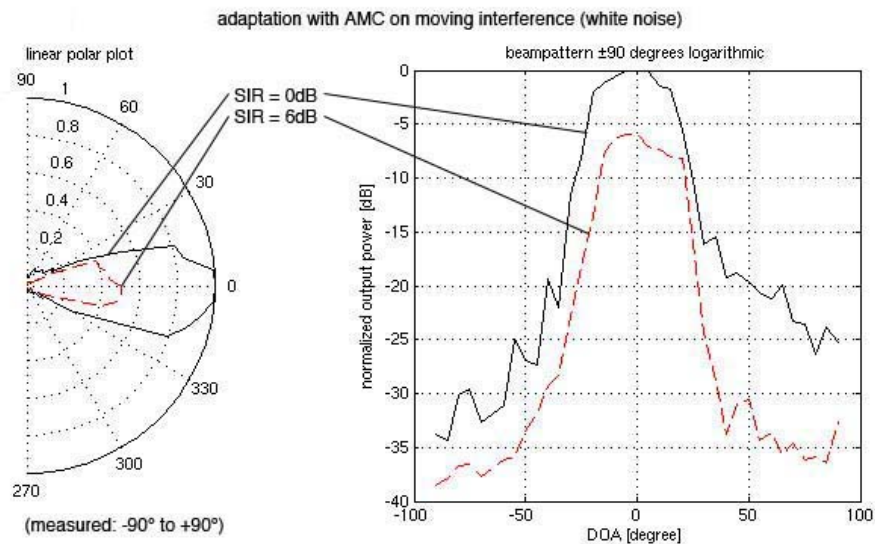


Abb. 4.35: Leistungsplot mit AMC bei verschiedenen SIRs, weißes Rauschen

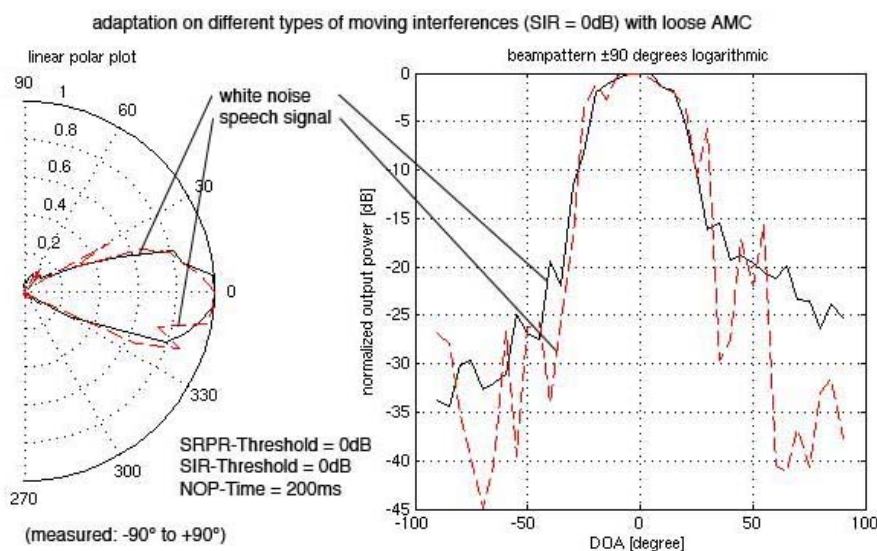


Abb. 4.36: Plots mit AMC, Sprachsignal und weißes Rauschen

Abb. 4.34, Abb. 4.35 und Abb. 4.36 zeigen die Messergebnisse bei eingeschaltetem AMC unter der Messung durch Methode 2. Zu erkennen ist im Prinzip die richtungsabhängige Dämpfung einer Störquelle (bei 0dB SIR auch als Signalquelle zu betrachten) mit unterschiedlichen AMC Parametern, Signal-Rausch Abständen und Eingangssignalen. Es wurden die Koeffizientenbeschränkungen aus Kapitel 4.5.3 für die CCAFs benutzt. Die Normbeschränkung der NCAFs lag bei $\Gamma = 10$ und die Schrittweiten bei $\alpha = 0,02$ und $\beta = 0,004$. In Abb. 4.34 erkennt man die Leistung des AMC bei verschiedenen Thresholds Q und S. Erhöht man die Schwellen ($Q = 5\text{dB}$, $S = 3\text{dB}$), so lässt sich eine Verkleinerung der Constrained Region beobachten. Die NOP-Time gibt an, wie lange der Beamformer bei einer Unsicherheit eingefroren bleiben soll. Wird die Constrained Region verkleinert, muss die NOP-Time erhöht werden, um Wunschsignalauslöschung zu vermeiden, da es öfter und

länger zu Unsicherheiten des AMC kommen kann. Abb. 4.35 liefert den Vergleich zwischen einer Störquelle mit 0dB SIR und einer Störquelle mit 6dB SIR. Betrachtet man die 0dB Quelle als Signalquelle, so kann man erkennen, wie stark eine Störquelle mit 6dB SIR im Verhältnis zur Signalquelle gedämpft wird. Die Diagramme in Abb. 4.36 zeigen das Verhalten des RAB bei Sprachsignal, verglichen mit weißem Rauschen. Man kann sehr gut erkennen, dass der AMC unabhängig vom Eingangssignal richtig zwischen Target und Interference unterscheiden kann.

4.5.7 Evaluierung

Dieser Beamformer weist unter Verwendung des verbesserten AMC die größte Robustheit im Vergleich zu allen anderen vorgestellten Algorithmen auf. Es hat sich in den Simulationen sowie bei der Vermessung der Echtzeitanwendung gezeigt, dass es keine Beeinträchtigung der Funktionalität durch Verstärkungsfehler oder Positionierungsfehler gibt. Der RAB funktioniert als Echtzeitanwendung so, wie in der Simulation vorhergesagt. Auch die subjektive Beurteilung in einer halligen Umgebung hat die Erwartungen bestätigt. Signalunabhängig können Störquellen außerhalb eines einstellbaren Bereichs herausgefiltert werden. Signal-Stör Abstände bis hin zu 0dB werden richtig interpretiert. Somit ist dieses System unter realen Bedingungen voll einsetzbar.

5 Messaufbau

Gemessen wurde in einem reflexionsarmen Raum der Firma AKG Acoustics. Der Raum hat eine Größe von $l \times b \times h = 3\text{m} \times 2,5\text{m} \times 2,5\text{m}$. Die untere Grenzfrequenz des Raumes liegt trotz starker breitbandiger Absorber laut Angaben von AKG bei etwa 150Hz bis 200Hz.

5.1 Aufbau

In Abb. 5.1 kann man den Messaufbau sehen. Das Array wurde in einer Höhe von 1,3m (halbe Raumhöhe) auf einem Drehteller platziert, der mit einer Auflösung von 5° gesteuert werden kann. In einem Abstand von einem Meter standen eine fixe Quelle (1) und eine bewegliche Quelle (2). Die bewegliche Quelle konnte durch vorher ausgemessene Winkelmarkierungen ebenfalls in einer Auflösung von 5° umgestellt werden.

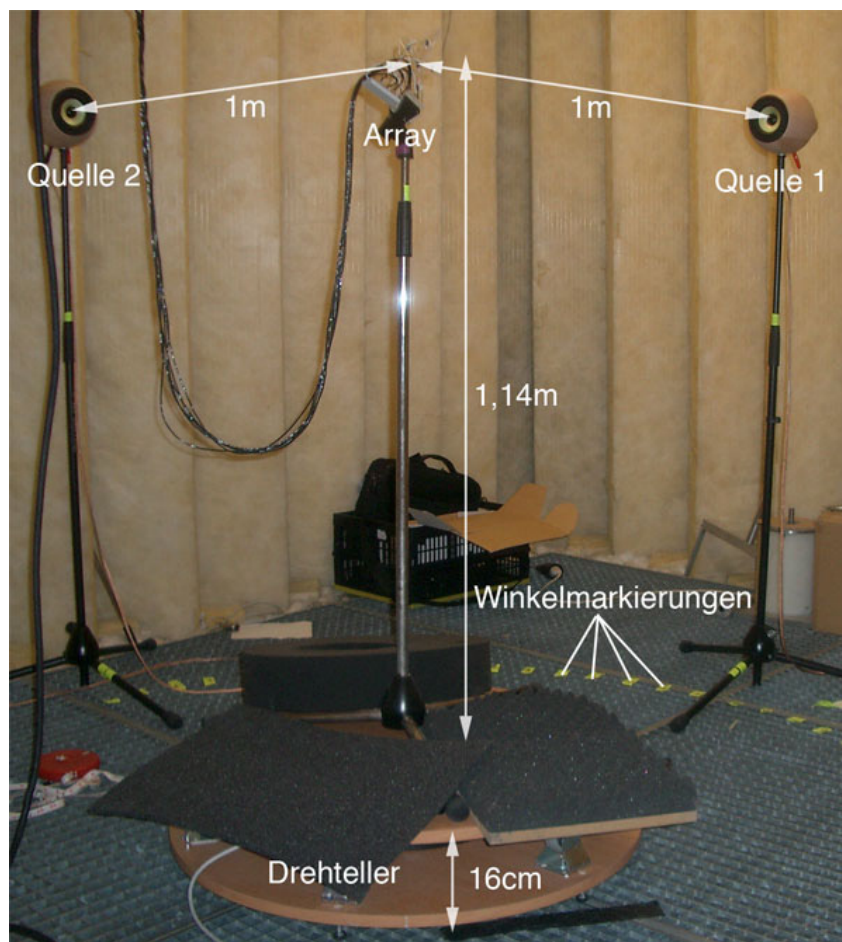


Abb. 5.1: Messaufbau

Die Messsignale wurden mit einem Schalldruck von 85 dB SPL abgespielt (gemessen am Ort des Arrays). Wenn es nötig war mit einem Rauschabstand > 0 dB zu messen, so wurde die Verstärkung der 2. Quelle entsprechend verringert. Die vom Array aufgenommenen Signale wurden über einen AD-Umsetzter mit 44,1 kHz und 16 bit in Amplitude aufgezeichnet und zwischen 150 Hz bis 7,6 kHz bandbegrenzt. Jeder Messpunkt wurde 20 s lang aufgenommen, um eine ausreichende Signaldauer für die adaptiven Algorithmen zu gewährleisten. Mit den aufgezeichneten Signalen ist es möglich jeden in PD realisierten Beamformer einzeln und mit unterschiedlichen Parametern zu vermessen.

5.2 Methoden:

Es wurden 2 Arten von Messungen durchgeführt:

Methode 1:

Durch Quelle 1 wird ein Messsignal abgespielt und über das Array aufgezeichnet. Das Array wird um 5° gedreht und der Vorgang wiederholt. Der Winkelbereich bei dieser Methode erstreckt sich von 0° bis 180° . Alle Richtplots und Beampattern sind somit an der 0° Achse gespiegelt dargestellt. Als Messsignal wird erst rosa Rauschen, dann weißes Rauschen verwendet. Die aufgezeichneten Signale dienen zur Vermessung der Leistungsrichtplots eines einzelnen Mikrofons, D&S-B, F&S-B, Frost-B und zur Ermittlung der Performance der Filter im RAB.

Methode 2:

Bei der zweiten Methode werden die ersten 10 Sekunden Signale aus beiden Quellen abgespielt. In den letzten 10 Sekunden verstummt Quelle 1 und Quelle 2 sendet alleine.

Dabei wird am Ende des gesamten Ablaufes der über 20 Punkte gemittelte Wert der Ausgangsleistung aufgezeichnet. Das bedeutet es wird nur die Leistung der Interference gemessen. Danach wird Quelle 2 um 5° versetzt und der Vorgang wiederholt. Der betrachtete Messbereich verläuft von 90° bis -90° . Als Messsignal wird zuerst weißes Rauschen aus beiden Quellen mit 0 dB SIR verwendet, dann weißes Rauschen aus beiden Quellen mit 6 dB SIR, danach männliche Sprache aus Quelle 1 und weibliche Sprache aus Quelle 2, mit einer SIR von 0 dB und schließlich mit 6 dB SIR. Diese Messung dient dazu, beim RAB das Verhalten des AMC unter verschiedenen Bedingungen zu testen.

Gemessen wurde immer die Ausgangsleistung eines Systems, abhängig von der Einfallsrichtung der Messquelle. Die Aufnahme der Messwerte erfolgte in PD und wurde dann zur Auswertung in MATLAB überführt. Dabei wurden alle adaptiven Prozesse (soweit vorhanden) für jeden Messpunkt zuerst zur Konvergenz gebracht und danach der Messwert aufgenommen. Jeder Messwert entstand durch die Mittelung von 20, innerhalb einer Sekunde aufgenommener, Einzelmesswerte.

$$P(n) = (1 - \sigma)P(n-1) + \sigma \text{Output}^2(n) \quad (5.1)$$

$$\mathbf{P} = \left[P_1 \left(n - 19 \left(\frac{0.05}{T_s} \right) \right), P_2 \left(n - 18 \left(\frac{0.05}{T_s} \right) \right), \dots, P_k \left(n - (20 - k) \left(\frac{0.05}{T_s} \right) \right), \dots, P_{19} \left(n - 1 \left(\frac{0.05}{T_s} \right) \right), P_{20}(n) \right] \quad (5.2)$$

mit $k = 1 \dots 20$, $\sigma =$ Aktualisierungsfaktor, $T_s =$ Samplingintervall

$$P_{\text{out}} = \frac{1}{20} \sum_{k=1}^{20} P_k \quad (5.3)$$

σ wurde für diese Messungen mit 0,005 gewählt. Alle Beamformer Messungen sind mit der Richtcharakteristik der verwendeten Mikrofonkapseln überlagert (Abb. 5.2.)

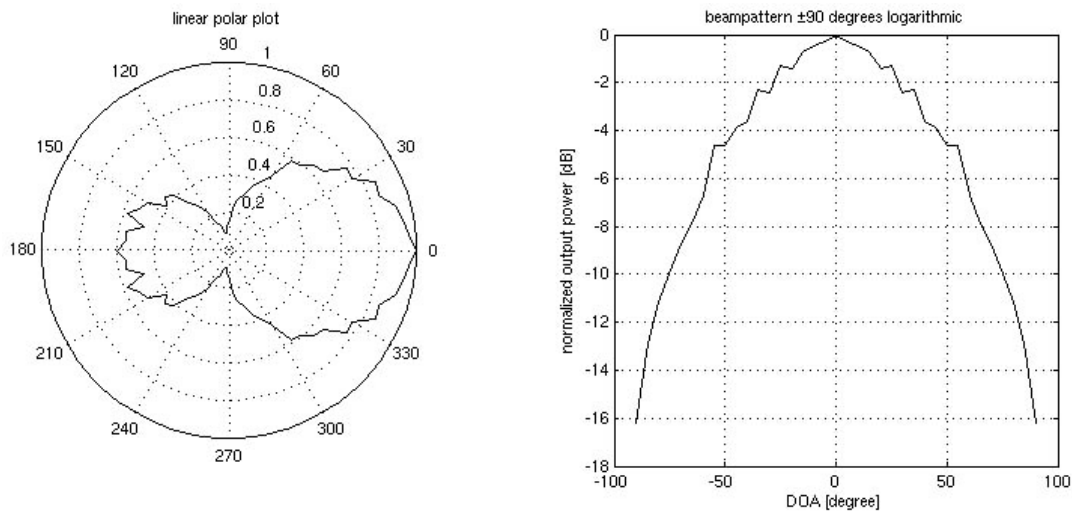


Abb. 5.2: gemessene Richtcharakteristik der Mikrofonkapseln (weißes Rauschen)

5.3 Varianzabschätzung der Messergebnisse

Nachdem jede Messung durch das gleiche System verarbeitet wird, reicht es, nur für einen Messpunkt eine Varianzabschätzung zu machen. Die Schätzung in Prozent ist auf alle Messpunkte übertragbar. Die Varianz berechnet sich zu:

$$\sigma^2(x) = E\{(x - \bar{x})^2\} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (5.4)$$

$$\text{Standardabweichung } \sigma(x) = \sqrt{\sigma^2(x)} \quad (5.5)$$

Die folgende Tabelle zeigt verschiedene Messwerte von RAB, D&S-B und Einzelmikrofon. Zusätzlich wird immer rechts von den Messwerten der Mittelwert angezeigt, unterhalb der Messwerte die Abweichung vom Mittelwert und rechts neben den Abweichungen jeweils die Standardabweichung. Für jeden Messwert wurde das System neu adaptiert, gemessen wurde mit Methode 1 bei einer Ausrichtung der Quelle von 90° zum Array.

rosa Rauschen	x_1	x_2	x_3	x_4	Mittelwert	$\sigma(x)$
RAB	-53,9dB	-52,7dB	-54,3dB	-50,7dB	-52,9dB	$\pm 3,05\%$
Abweichung	-1dB	0,2dB	-1,4dB	2,2dB		$\pm 1,612\text{dB}$
D&S-B	-36,9dB	-37,5dB	-37,2dB	-36,6dB	-37,05dB	$\pm 1,05\%$
Abweichung	0,15dB	-0,45dB	-0,15dB	0,45dB		$\pm 0,387\text{dB}$
Einzelmikro	-37,9dB	-37,6dB	-37,1dB	-37,2dB	-37,45dB	$\pm 0,99\%$
Abweichung	-0,45dB	-0,15dB	0,35dB	0,25dB		$\pm 0,37\text{dB}$
weißes Rauschen	x_1	x_2	x_3	x_4	Mittelwert	$\text{sdv}(x)$
RAB	-53,1dB	-52,5dB	-53dB	-52,7dB	-52,83dB	$\pm 0,52\%$
Abweichung	-0,27dB	0,33dB	-0,17dB	0,13dB		$\pm 0,275\text{dB}$
D&S-B	-38,4dB	-37,9dB	-38,3dB	-38,1dB	-38,18dB	$\pm 0,58\%$
Abweichung	-0,22dB	0,28dB	-0,12dB	0,08dB		$\pm 0,222\text{dB}$
Einzelmikro	-33dB	-33,5dB	-33,1dB	-33,3dB	-33,23dB	$\pm 0,67\%$
Abweichung	0,23dB	-0,27dB	0,13dB	-0,07dB		$\pm 0,222\text{dB}$

Man kann erkennen, dass die Standardabweichung zum größten Teil unter $\pm 1\%$ liegt. Der RAB hat bei rosa Rauschen eine höhere Abweichung. Das liegt daran, dass der RAB zu unterschiedlichen Zeitpunkten unterschiedlich gut adaptiert und deswegen höhere Schwankungen auftreten. Rosa Rauschen beeinträchtigt die Qualität der Adaption.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

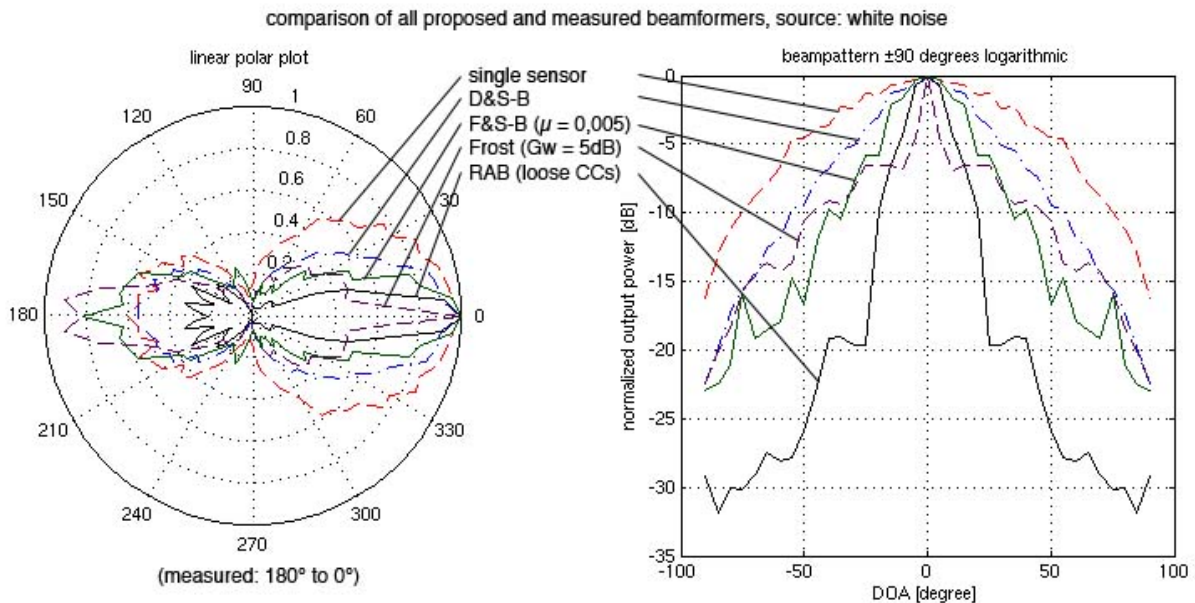


Abb. 6.1: alle vermessenen Beamformer im Vergleich

Diese Diplomarbeit führt Schritt für Schritt zu einer robusten Lösung für ein adaptives Mikrofonarray kleiner Apertur. Ausgewählte adaptive Algorithmen werden auf ihre Eignung unter realen Bedingungen untersucht und bewertet. Als Basisalgorithmus dient dafür der Frost Beamformer. Zuerst werden kurz die Grundlagen des Arrayprocessing und Beamforming behandelt und der grundlegende Optimierungsansatz für Sensorarrays vorgestellt, die Superdirektivität über „*Minimum Variance Distortionless Response*“. Mit der Hilfe von 3-dimensionalen Simulationen des Delay & Sum Beamformers wird eine Arraygeometrie mit sechs Sensoren gefunden und gebaut, die ein sehr gutes Steeringverhalten in alle Richtungen aufweist und eine hohe obere Grenzfrequenz besitzt.

Vom Minimum Varianz Beamformer ausgehend werden der breitbandige Minimum Varianz Ansatz (Filter & Sum Beamformer) mit regularisierter Korrelationsmatrix, der konventionelle Frost Beamformer [11], der Frost Beamformer mit einer Anhebung und Beschränkung des Gewinnes gegenüber unkorrelierten Störungen [11], der Frost Beamformer mit adaptiver Steering Vektor Korrektur [10] und der Griffiths Jim Beamformer mit CCAF-NCAF Struktur und AMC [1] hergeleitet, mit der vorgestellten Arraygeometrie simuliert und teilweise in die Echtzeit Programmiersprache PD überführt.

Alle in PD implementierten Ansätze werden zusammen mit dem gebauten Array akustisch vermessen und auf Anwendbarkeit und Robustheit getestet. Es stellt sich heraus, dass keine der als robust geltenden Methoden für Applikationen ausreichende Fehlertoleranz aufweist. Für den Beamformer mit CCAF-NCAF Struktur und AMC wird in weiterer Folge ein verbesserter AMC vorgestellt, der durch einen Limiter/ Expander mit Sidechain die richtungsabhängigen Schwankungen der geschätzten SIR verringert und über eine Steered Response Power Ratio (SRPR) Störquellen außerhalb der Constrained Region signalunabhängig als Störer identifiziert. Mit dem neuen AMC gelingt es schließlich, einen „Robust Adaptive Beamformer“ zu realisieren.

6.2 Ausblick

Mit der vorgestellten Arraygeometrie, der CCAF-NCAF Struktur und dem verbessertem AMC ist es möglich, einen Sprecher innerhalb der Constrained Region deutlich besser von Störgeräuschen abzuheben, als bei Verwendung eines normalen Richtmikrofones. Für eine wirklich unabhängige Kommunikation bedarf es einer Bewegungsverfolgung des Sprechers (*Target Tracking*). Die Implementierung einer Sprecherverfolgung ist nicht mehr in die Diplomarbeit mit eingeflossen und bleibt somit noch aus. Jedoch ist der Schritt zum selbst ausrichtenden Mikrofonarray nicht mehr sehr groß. Durch den Ansatz der Steered Response Power Ratio kann man die Position einer Quelle ausfindig machen. Es muss lediglich eine Heuristik entwickelt werden, die eine Verwechslung mit Interference ausschließt und zuverlässig einem Target folgt. Der Einfallswinkel eines Target kann in Predelays übersetzt werden, die den vorgeschalteten Delay & Sum Beamformer in die entsprechende Richtung auslenken. Es ist dabei noch zu beachten, dass sich in der vorgestellten Implementierung die Hauptkeule des Beamformers abhängig vom Steeringwinkel verändert. Je größer der Steeringwinkel wird, desto breiter wird die Hauptkeule. Das geschieht, weil die Steered Response Power Ratio, die auf der Seite des Beamsteerings gebildet wird, mit zunehmendem Steeringwinkel einen kleineren Wertebereich und eine geringere Steigung annimmt. Der Hauptbeam rückt immer näher an der betreffenden Seitenbeam heran. Das bewirkt, dass der SRPR Wert länger innerhalb des zulässigen Bereichs bleibt. Deshalb muss eine winkelabhängige Gewichtung des Seitenbeams eingeführt werden, die bei einem Steering die betroffene SRPR in den ursprünglichen Wertebereich zurückzieht. Die genaue mathematische Beziehung ist dafür noch zu finden. Denkbar ist ein in dB linear zunehmender Offset der Leistung des Seitenbeams für steigende Steeringwinkel. Abb. 6.2 zeigt die Verbesserungsmöglichkeit bei einem Steering des RAB. Hier wurde eine Anhebung der rechten Seitenbeamleistung um 5dB vorgenommen. Es ist eine schmälere Hauptkeule zu erkennen.

Die Anwendung des Arrays ist aufgrund der Geometrie beschränkt auf die vordere Halbebene, in einem Winkelbereich von in etwa $\pm 60^\circ$ Azimut und Elevation um die Symmetrieachse. Eine typische vorstellbare Anwendungssituation ist z. B. ein zu übertragender beweglicher Sprecher vor Publikum, oder als Teil einer Freisprecheinrichtung in einem Fahrzeug. Möchte man mehrere Sprecher versammelt um einen Tisch übertragen, so ist eine andere Arraygeometrie nötig.

Vorstellbar ist eine Ringgeometrie, die als eine Art Grenzflächenbeamformer in der Mitte des Tisches platziert wird somit um einen Azimutbereich von 360° gleichmäßig ausgerichtet werden kann. Mit solchen Geometrien wurde im Zuge der Diplomarbeit bereits experimentiert. Es hat sich in Simulationen gezeigt, dass eine ringförmige Anordnung mit wenigen Mikrofonen keine guten Ergebnisse liefert.

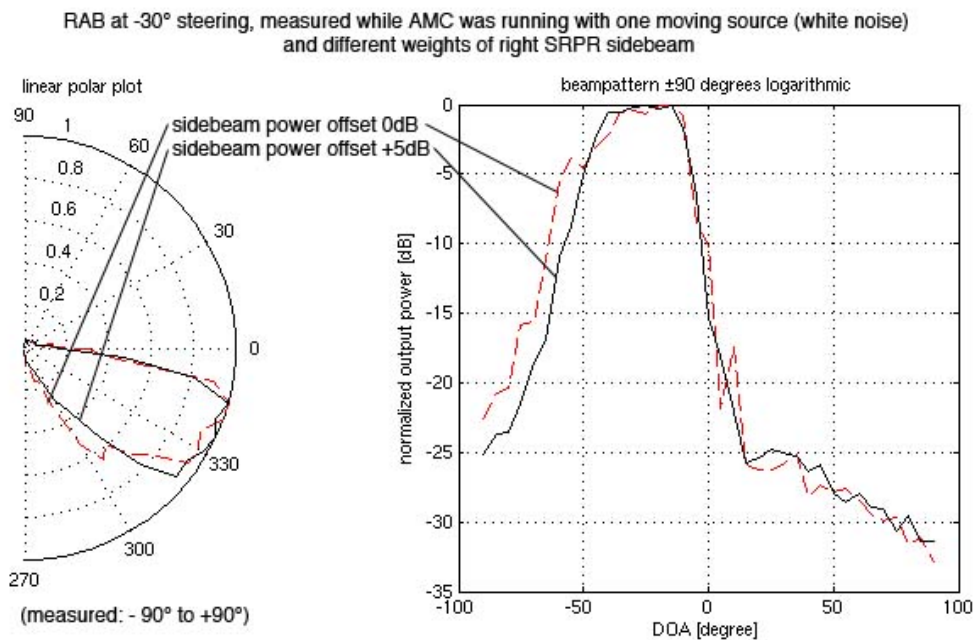


Abb. 6.2: RAB mit AMC, Steering: -30° , bei untersch. SRPR sidebeam Gewichtungen

Es ist zu überlegen, ob für eine Telekonferenzanwendung mit mehreren Personen um ein Tischarray herum nicht vom Beamforming abgesehen werden sollte. Eine wahrscheinlich effektivere und einfachere Lösung wäre es, ein Kreisarray bestehend aus Richtmikrofonen auf dem Tisch zu platzieren, das durch eine energetische Abschätzung einzelne Mikrofone gewichtet, ein, oder stumm schaltet. Ein solches System kann auf einem Array mehrmals parallel implementiert werden und sich unabhängig auf verschiedene Sprecher ausrichten.

7 Literaturverzeichnis

- [1] BRANDSTEIN Michael, WARD Darren: *Microphone Arrays – Signal Processing Techniques and Applications*, ISBN 3-540-41953-5 Springer- Verlag, Cambridge MA, USA – London, UK, January 2001
- [2] COX Henry, ZESKIND Robert M., OWEN Mark M.: *Robust Adaptive Beamforming*, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-35 No. 10, October 1987
- [3] FALCH Cornelia: *Smart Microphone Array – Final Project Report*, Universität für Musik und darstellende Kunst, Institut für elektronische Musik, Graz 2003
- [4] GRIFFITHS Lloyd J., JIM Charles W.: *An Alternative Approach to Linearly Constrained Adaptive Beamforming*, IEEE transactions on Antennas and Propagation, vol. AP-30 No. 1, January 1982
- [5] HOSHUYAMA Osamu, BEGASSE Brigitte, SUGIYAMA Akihiko, HIRANO Akihiro: *A Realtime Robust Adaptive Microphone Array Controlled by an SNR Estimate*, IEEE ICASSP'98, pp.3605-3678, May 1998
- [6] HOSHUYAMA Osamu, SUGIYAMA Akihiko, HIRANO Akihiro: *A Robust adaptive Beamformer for Microphone Arrays with a Blocking Matrix Using Constrained Adaptive Filters*, IEEE Transactions on Signal Processing, vol. 47, no. 10, October 1999
- [7] MCCOWAN I. A.: *PhD Thesis – Robust Speech Recognition using Microphone Arrays*. Queensland University of Technologie, Australia, 2001
- [8] MOSCHYTZ George, HOFBAUER Markus: *Adaptive Filter – Eine Einführung in die Theorie mit Aufgaben und MATLAB-Simulationen auf CD-ROM*, ISBN 3-540-67651-1 Springer- Verlag, Zürich, Juli 2000
- [9] PAPE Lutz: *Projektarbeit – Akustische Dialogsegmentierung mit kleinen Mikrofonarrays*. Technische Universität, Universität für Musik und darstellende Kunst, Graz, January 2005
- [10] QIYUE Zou, ZHU LIANG Yu, ZHIPING Lin: *A Robust Algorithm for Linearly Constrained Adaptive Beamforming*. IEEE Signal Processing Letters, vol. 11, no.1, January 2004
- [11] STÖBICH Gerhard: *Diplomarbeit – Entwurf und Simulation eines adaptiven, zweidimensionalen Mikrofonarrays*, Wien, September 2001
- [12] WARUM Stefan: *Diplomarbeit – Messung von Aussenohrübertragungsfunktionen und direktionalen Raumimpulsantworten als Basis der Auralisation realer Räume*, Universität für musik und darstellende Kunst, Graz, Juni 2002
- [13] WIDROW Bernard, STEARNS Samuel D.: *Adaptive Signal Processing*, ISBN 0-13-004029-0 Prentice Hall Inc., New Jersey, USA, 1985
- [14] MILLER PUKETTE, <http://crca.ucsd.edu/~msp/>

8 Anhang: MATLAB Funktionen

Alle m-Files und Datensätze sind ausdokumentiert auf der beiliegenden CD-Rom zu finden.

8.1 Listen und Kurzbeschreibungen

globale Funktionen (alphabetisch):

CreateTestData.m	erzeugt Mikrofonsignale abhängig von Arraygeometrie und Einfallswinkel [11]
Create_error_TestData.m	wie CreateTestData.m mit zusätzlicher Berücksichtigung von Presteering und Positionierungsfehlern
delaysum2.m	berechnet schmalbandigen D&S-B [11]
fircoeffs2.m	berechnet Filter eines F&S-B [11]
getdelays.m	berechnet Steeringdelay Matrix für einen Steeringbereich (Azimut und Elevation)
GetImpacts.m	berechnet Steeringdelays für einen Steeringwinkel
Impacts.m	berechnet Phasendifferenzen zwischen Mikrofonen
levin.m	FIR-Filter Approximation [11]
lslevin.m	FIR-Filter Approximation [11]
showGeo.m	zeigt Arraygeometrie
wgn.m	erzeugt weißes, gaussverteiltes Rauschen [MATLAB]

D&S, Minimum Varianz, F&S Beamformer (alphabetisch):

Beam3D.m	erstellt 3-dimensionale Richtplots von D&S Beamformern
DS_Out_Power.m	erstellt Beampattern und Richtplot der Ausgangsleistung eines D&S-B
FS_Out_Power.m	erstellt Beampattern und Richtplot der Ausgangsleistung eines F&S-B
minVar2D.m	erstellt Beampattern und Richtplot eines schmalbandigen Minimum Varianz Beamformers
minVar3D_freq.m	erstellt Beampattern vs. Frequenz für D&S-B und Minimum Varianz Beamformer
opt_constr2.m	berechnet die optimalen Gewichte eines schmalbandigen Minimum Varianz Beamformers [11]

Frost Beamformer (alphabetisch):

CoxBeam.m	Frost Beamformer mit beschränktem G_w [11]
FrostBeam.m	konventioneller Frost Beamformer
FrostBeamExt.m	konventioneller Frost Beamformer mit variablem C_0
FrostExtension.m	Frost Beamformer mit adaptiver Steering Vektor Korrektur
PlotCox.m	erstellt Beampattern/ Richtplot der Ausgangsleistung von CoxBeam.m
PlotFrost.m	erstellt Beampattern/ Richtplot der Ausgangsleistung von FrostBeam.m
PlotFrostExt.m	erstellt Beampattern/ Richtplot der Ausgangsleistung von FrostBeamExt.m
SimFrost.m	simuliert Frostbeamformer mit einer Target und einer Interference
SimFrostExt.m	simuliert adaptive Steering Vektor Korrektur

Robust Adaptive Beamformer (alphabetisch):

BlockingMatrix.m	berechnet nur die Blocking Matrix
CCAF_Thresh.m	berechnet Filterkoeffizienten der Blockingmatrix
get_Thresholds.m	Simulationsumgebung für CCAF_Thresh.m
PlotBM.m	erstellt Beampattern/ Richtplot der Ausgangsleistung der BM
PlotRAB.m	erstellt Beampattern/ Richtplot der Ausgangsleistung des RAB
RAB_AMC_TDE.m	berechnet RAB mit AMC und TDE
RAB_for_Plot.m	berechnet RAB mit gezwungenem Adaptionablauf
RAB_plus_AMC_dB.m	berechnet RAB mit AMC und Limiter/ Expander in dB
RAB_plus_konvAMC.m	berechnet RAB mit AMC nach [1]
robust0.m	Simulationsumgebung für RAB_plus_konvAMC.m
robust1.m	Simulationsumgebung für RAB_plus_AMC_dB.m
robust2.m	Simulationsumgebung für RAB_AMC_TDE.m
TDE.m	Time Dealy Estimation Filter [3]

8.2 Datensätze und Testsignale

Datensätze:

errors.mat	Beschreibung: (Geometrie: Stern6_r4)
gainerr_3dB	Versärfungsfehler ± 3 dB,
gainerr_6dB	Verstärkungsfehler ± 6 dB
poserr_5mm_xy	Positionierungsfehler ± 5 mm in x- und x-Richtung

DiplArrays.mat	Beschreibung:
Bogen_120_5	120° Bogen mit Mikrofonabstand 3cm, 5 Sensoren
Bogen_120_5plus2	120° Bogen mit Mikrofonabstand 3cm, 2 Sensoren jeweils 6 cm über und unter dem Mittelpunkt, 7 Sensoren
Bogen_60_5plus2	60° Bogen mit Mikrofonabstand 3cm, 2 Sensoren jeweils 6 cm über und unter dem Mittelpunkt, 7 Sensoren
lin4aussen2	4 Elemente, Abstand 4cm, äußere Sensoren 2cm nach vorne versetzt
lin4plus2	4 Elemente, Abstand 4cm, 2 zusätzl. Sensoren jeweils 6cm über und unter dem Mittelpunkt
lin5plus2	5 Elemente, Abstand 4cm, 2 zusätzl. Sensoren jeweils 6cm über und unter dem Mittelpunkt
Lin5plus2oben	5 Elemente, Abstand 4cm, 2 zusätzl. Sensoren 6cm und 12cm über dem Mittelpunkt
Lin5plus2versetzt	5 Elemente, Abstand 4cm, versetzt auf x-Achse um 0,9 und 3cm, 2 zusätzl. Sensoren jeweils 6cm über und unter dem Mittelpunkt
Mapped_Bogen_120_5plus2	Bogen_120_5plus2 auf eine Linie projiziert
Mapped_Bogen_60_5plus2	Bogen_60_5plus2 auf eine Linie projiziert
Stern6_r4_hor	Stern in xy-Ebene mit 6 Elementen, Radius 4cm
ring10_hor	Ring in xy-Ebene mit 10 Elementen, Radius 8cm

goodArrays.mat	Beschreibung:
Kreuz8	Kreuz mit acht Sensoren, Abstand 4cm
lin3	lineares Array mit 3 Sensoren, Abstand 4cm
lin4	lineares Array mit 4 Sensoren, Abstand 4cm
lin5	lineares Array mit 5 Sensoren, Abstand 4cm
Stern6_r4	Stern mit 6 Elementen, Radius 4cm
Stern6_r6	Stern mit 6 Elementen, Radius 6cm

SRPR_FILTER.mat	Bandpass für Steered Response Power Ratio, Grenzfrequenzen: 2kHz und 7kHz
------------------------	---

Xi_Psi.mat	Beschreibung: (Geometrie: Stern6_r4)
Psi	untere Schwellen der CCAFs, $\pm 15^\circ$ bei $F_s=16\text{kHz}$
Xi	obere Schwellen der CCAFs, $\pm 15^\circ$ bei $F_s=16\text{kHz}$
cos_Psi	untere Schwellen als Kosinusfenster
cos_Xi	obere Schwellen als Kosinusfenster

WoptStern.mat	Beschreibung: (Geometrie: Stern6_r4)
Wmue100	mit fircoeffs2.m berechnete Filterkoeffizienten eines F&S-B, 16 Filtertaps. Regularisierungsfaktor $\mu=100$
Wmue0p1	Regularisierungsfaktor $\mu=0,1$
Wmue0p005	Regularisierungsfaktor $\mu=0,005$
Wmue0p001	Regularisierungsfaktor $\mu=0,001$
Wmue0p0005	Regularisierungsfaktor $\mu=0,0005$
Wmue0p0001	Regularisierungsfaktor $\mu=0,0001$
Wopt	Regularisierungsfaktor $\mu=0$

Testsignale:

Fem_interf.wav	weibliche Stimme, mono, 16kHz, 10s
gaussmiddle.wav	gaussverteilttes weißes Rauschen, mono, 16khz, 6s
gaussmiddle2.wav	gaussverteilttes weißes Rauschen, mono, 16khz, 3s
gaussnoise.wav	gaussverteilttes weißes Rauschen, mono, 16khz, 4s
Male_Target.wav	männliche Stimme, mono, 16kHz, 10s
shortgaussnoise.wav	gaussverteilttes weißes Rauschen, mono, 16khz, 1s
white.wav	weißes Rauschen, mono, 16khz, 10s

8.3 Aufgenommene Messdaten

Frost_F&S_0_180.mat	Beschreibung: (Messmethode 1)
az_patt	Azimutwinkel für Beampattern [°], 1x37 Matrix
az_pol	Azimutwinkel für Polardiagramm [rad], 1x73 Matrix
rho_patt	normalisierte Messwerte Frost mit $G_w = 5\text{dB}$, Leistung [dB], 2x37 Matrix, 1: rosa Rauschen 2: weißes Rauschen
Rho_patt_FSB	normalisierte Messwerte F&S-B, $\mu=0,005$, Leistung [dB], 2x37 Matrix, 1: rosa Rauschen 2: weißes Rauschen
rho_pol	lineare Verstärkung Frost $G_w = 5\text{dB}$, 2x73 Matrix (Spiegelung an 0°), 1: rosa Rauschen 2: weißes Rauschen
rho_pol_FSB	lineare Verstärkung F&S-B, $\mu=0,005$, 2x73 Matrix (Spiegelung an 0°), 1: rosa Rauschen 2: weißes Rauschen
Frost_180_0	aufgenommene Messwerte Frost mit $G_w = 5\text{dB}$ 2x37 Matrix, 1: rosa Rauschen 2: weißes Rauschen
FSB_180_0	aufgenommene Messwerte F&S-B mit $\mu=0,005$ 2x37 Matrix, 1: rosa Rauschen 2: weißes Rauschen

RAB_Messung1_180_0.mat	Beschreibung: (Messmethode 1)
az_patt	Azimutwinkel für Beampattern [°], 1x37 Matrix
az_pol	Azimutwinkel für Polardiagramm [rad], 1x73 Matrix
rho_patt	normalisierte Messwerte bei lockerer Koeffizientenbeschränkung der CCAFs, Leistung [dB], 6x37 Matrix, 1: RAB, 2: D&S, 3: einzelnes Mik. – rosa Rauschen 4: RAB, 5: D&S, 6: einzelnes Mik. – weißes Rauschen
rho_patt_s	normalisierte Messwerte bei starker Koeffizientenbeschränkung der CCAFs, Leistung [dB], 2x37 Matrix, 1: RAB – rosa Rauschen 2: RAB – weißes Rauschen
rho_pol	lineare Verstärkung bei lockerer Koeffizientenbeschränkung der CCAFs, 6x73 Matrix, 1: RAB, 2: D&S, 3: einzelnes Mik. – rosa Rauschen 4: RAB, 5: D&S, 6: einzelnes Mik. – weißes Rauschen
rho_pol_s	lineare Verstärkung bei starker Koeffizientenbeschränkung der CCAFs, 2x73 Matrix (Spiegelung an 0°), 1: RAB – rosa Rauschen 2: RAB – weißes Rauschen
Messung1_180_0	aufgenommene Messwerte bei lockerer Koeffizientenbeschränkung der CCAFs, 6x37 Matrix, 1: RAB, 2: D&S, 3: einzelnes Mik. – rosa Rauschen 4: RAB, 5: D&S, 6: einzelnes Mik. – weißes Rauschen
Messung1_strong	aufgenommene Messwerte bei starker Koeffizientenbeschränkung der CCAFs, 2x37 Matrix, 1: RAB – rosa Rauschen 2: RAB – weißes Rauschen

RAB_Messung2_90_0_90.mat	Beschreibung: (Messmethode 2)
az_patt	Azimutwinkel für Beampattern [°], 1x37 Matrix
az_pol	Azimutwinkel für Polardiagramm [rad], 1x73 Matrix
rho_patt	normalisierte Messwerte bei lockeren AMC Thresholds, Leistung [dB], 4x37 Matrix, 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR
rho_patt_s	normalisierte Messwerte bei strengen AMC Thresholds, Leistung [dB], 4x37 Matrix, 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR
rho_pol	lineare Verstärkung bei lockeren AMC Thresholds, 4x73 Matrix (nur gültig von -90° bis +90°), 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR
rho_pol_s	lineare Verstärkung bei strengen AMC Thresholds, 4x73 Matrix (nur gültig von -90° bis +90°), 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR
Messung2_90_0_90	aufgenommene Messwerte bei lockeren AMC Thresholds, Leistung [dB], 4x37 Matrix, 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR
Messung2_strong	aufgenommene Messwerte bei strengen AMC Thresholds, Leistung [dB], 4x37 Matrix, 1: RAB weißes Rauschen, 2: RAB Sprache – 0dB SIR 3: RAB weißes Rauschen, 4: RAB Sprache – 6dB SIR

Anmerkung: der Datensatz RAB_Messung2_90_0_90.mat existiert auch mit einem Arraysteering von 30° für weißes Rauschen und ist unter RAB_Messung3_90_0_90_30.mat zu finden.

NoiseData_0_180.mat	Beschreibung:
BeamOut_0_180	enthält Ausgangssignale des RAB von 0 bis 180°, gemessen mit Methode 1, $F_s = 44,1\text{kHz}$
aCoeff	Filterkoeffizienten für Oktavfilterbank von 125Hz bis 8kHz (Nenner, eine Zeile entspricht einem Filter)
bCoeff	Filterkoeffizienten für Oktavfilterbank von 125Hz bis 8kHz (Zähler, eine Zeile entspricht einem Filter)

Get_Octave_Plots.m	erstellt alle Richtplots und Beampattern der Oktavfilterbank
---------------------------	--