Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Friedrich-Alexander-Universität

Erlangen-Nürnberg

Prof. Dr.-Ing. W. Kellermann

Blinde adaptive MIMO-Filterung zur simultanen Lokalisierung mehrerer Schallquellen

Jochen Stenglein

Studienarbeit, September 2004

Betreuer: Dipl.-Ing. Herbert Buchner

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Jochen Stenglein
Hinterer Bach 11
96049 Bamberg

Erlangen, den 3. September 2004

Jochen Stenglein

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

Zusammenfassung

In dieser Studienarbeit wird eine neue Methode vorgestellt, mit der mehrere Quellen gleichzeitig lokalisiert werden können. Diese Methode baut auf den Ergebnissen der blinden Quellentrennung (BSS = Blind Source Separation) auf. Da durch die adaptive Trennung der Quellsignale auch eine räumliche Filterung ausgeführt wird, kann aus den Filterkoeffizienten auch räumliche Information für die Lokalisierung gewonnen werden, um die Laufzeitunterschiede zwischen den Mikrophonen abzuschätzen. Mit einigen geometrischen Überlegungen können diese in räumliche Quellpositionen umgesetzt werden. Da die Mikrophongeometrie hierbei eine gewichtige Rolle spielt, werden zwei geeignete Mikrophonanordnungen diskutiert.

Um die Genauigkeit der Lokalisierung zu beurteilen, werden einige Szenarien mit den bekannten Algorithmen Generalized Cross Correlation (GCC) und Adaptive Eigenvalue Decomposition (AED) sowie dem hier vorgestellten Algorithmus bearbeitet. Die errechneten Laufzeitunterschiede werden sowohl untereinander als auch mit den als Referenz genutzten Infrarotdaten der Szenarien verglichen. Dabei wird die Varianz als Maß für die Genauigkeit der Lokalisierung bestimmt.

Verwendete Formelzeichen und Abkürzungen

GCC	Generalized Cross Correlation
AED	Adaptive Eigenvalue Decomposition = Adaptive Eigenwertzerlegung
BSS	Blind Source Separation = Blinde Quellentrennung
TDOA	Time Difference Of Arrival = Laufzeitunterschied
MIMO	Mulitiple Input Multiple Output
ICA	Independent Component Analysis
CC	Cross-Correlation
PHAT	Phase Transform
VAD	Voidce Activity Detector
FFT	Fast Fourier Transformation
IFFT	Inverse Fast Fourier Transformation
LMSE	Least Mean Squared Error = kleinster quadratischer Fehler
$x_i(t)$	Signal am i-ten Mikrophon (zeitkontinuierlich)
$x_i(n)$	Signal am i-ten Mikrophon (zeitdiskret)
$x_i(m)$	Signal am i-ten Mikrophon (Blockzeit m)

α_i	Dämpfungsfaktor am i-ten Mikrophon
$s_i(t)$	Signal der i-ten Quelle
$b_i(t)$	Rauschsignal am i-ten Mikrophon
τ	Laufzeitunterschied (in Sekunden)
$\hat{\tau}$	geschätzter Laufzeitunterschied (in Abtastwerten)
$\mathcal{F}$	siehe FFT
$\mathcal{F}^{-1}$	siehe IFFT
$\underline{\mathbf{s}}_{x_1x_2}$	Kreuzleistungsdichtespektrum von x_1 und x_2
$\underline{\Phi}$	Gewichtungsfunktion für die GCC-Methode
$\mathbf{r}_{x_1x_2}$	Kreuzkorrelierte von x_1 und x_2
m	Blockindex
$\mathbf{\Psi}$	GCC-Funktion im Zeitbereich
$\underline{\psi}$	GCC-Funktion im Frequenzbereich
$\mathbf{h}_i$	Impulsantwort von der Quelle zum i-ten Mikrophon
$\hat{\mathbf{h}}_i$	geschätzte Impulsantwort von der Quelle zum i-ten Mikrophon
$\mathbf{h}_{ij}$	Impulsantwort von der i-ten Quelle zum j-ten Mikrophon
$\hat{\mathbf{h}}_{ij}$	geschätzte Impulsantwort von der i-ten Quelle zum j-ten Mikrophon
$\mathbf{u}$	Vektor aus den Impulsantworten
μ	Schrittweite
μ_{norm}	normierte Schrittweite
μ_{off}	Schrittweite im offline-Teil
$\mathbf{P}$	Leistung
M	Filterlänge
N	Blocklänge

x_s, y_s, z_s	karthesische Koordinaten der Quelle
x_i, y_i, z_i	karthesische Koordinaten des i-ten Mikrophones
$y_i(t)$	Signal am i-ten Ausgang
$\mathbf{H}$	Mischsystem
$\hat{\mathbf{H}}$	geschätztes Mischsystem
$\mathbf{W}$	Entmischsystem
$\mathbf{W}^{-1}$	invertiertes Entmischsystem
$\Im(m)$	Kostenfunktion für den m-ten Block
$\beta(i,m)$	Gewichtsfunktion zur Variation zwischen online- und offline-Modus
∇_W^{NG}	Ableitung mit dem natürlichen Gradienten nach W
$\mathcal{O}$	Mächtigkeit der Rechenekomplexität
K	Anzahl der offline-Blöcke in einem online-Block
L	Anzahl der Zeitverschiebungen
$\mathbf{I}$	Einheitsmatrix
c_s	Schallgeschwindigkeit in Luft ($\approx 334\frac{m}{s}$)
f_s	Abtastrate
r	Mikrophonabstand
N_{TDOA}	Anzahl der bestimmbaren TDOA-Werte an einem Mikrophonapaar
N_{pos}	Anzahl der bestimmbaren Positionen
τ_{max}	maximaler Laufzeitunterschied zwischen zwei Mikrophonen (in Abtastwerten)
$\sigma^2_{AED}, \sigma^2_{GCC}, \sigma^2_{BSS}$	Varianzen der Algorithmen AED, GCC und BSS

Kapitel 1

Einleitung

Für zukünftige Mensch-Maschine-Schnittstellen werden zunehmend nicht nur einkanalige, sondern auch mehrkanalige Aufnahme- und Wiedergabemöglichkeiten von Interesse sein. Daher ist es erstrebenswert, die Position von mehreren Quellen simultan zu bestimmen, um ein Mikrophonarray oder Kameras darauf ausrichten zu können. Im Verlauf dieser Arbeit soll bestimmt werden, ob die blinde Quellentrennung (BSS) für diese Aufgabe geeignet ist.
Im Allgemeinen kann die Positionsbestimmung entsprechend Abbildung 1.1 in zwei Teilprobleme unterteilt werden. Im ersten Schritt müssen die Lauf-

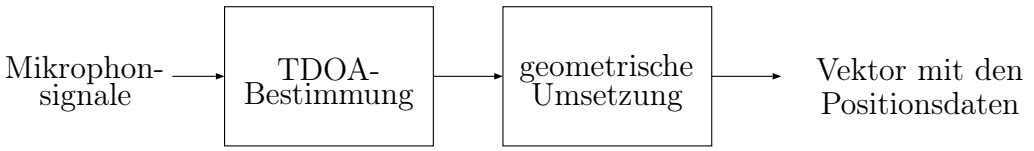

Abbildung 1.1: Vorgehen zur Positionsbestimmung einer Quelle

zeitunterschiede (TDOA = Time Difference Of Arrival) zwischen den Mikrophonen berechnet und im zweiten Schritt unter Einbeziehung der Mikrophongeometrie in mehrdimensionale (z.B.zweidimensionale) Quellpositionen umgesetzt werden.

Schritt 1: Zur Abschätzung der Laufzeitunterschiede werden mindestens zwei Mikrophone, bei mehreren Quellen mindestens so viele Mikrophone wie Quellen benötigt. Zwei häufig angewandte Verfahren sind AED [2] und GCC [1], die jeweils nur den Laufzeitunterschied einer Quelle bestimmen können. Eine dritte hier vorgestellte Methode basiert auf den Ergebnissen der blinden Quellentrennung und kann mehrere Quellen gleichzeitig handhaben. Dazu wird zuerst die Funktionsweise der blinden Quellentrennung betrachtet. Mittels adaptiver Filterung, basierend auf ICA (Independant Component Analysis), kann die BSS zwei voneinander statistisch unabhängige Quellsignale, die in den Mikrophonsignalen vermischt vorliegen, voneinander trennen. Dies kann man als adaptives Beamforming [12] interpretieren, bei dem jedoch zusätzlich entsprechend Abbildung 1.2 das Mikrophonarray auf die Quellen automatisch ausgerichtet wird (Steering). Anschließend werden die Signale durch räumliche Filterung getrennt. Aus den Steering-Informationen können

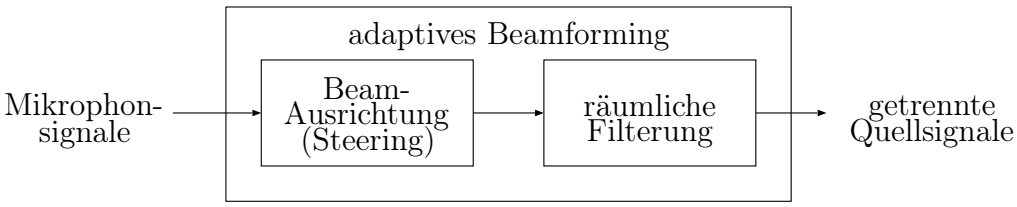

Abbildung 1.2: Interpretation von BSS als adaptiven Beamformer

die Laufzeitunterschiede gewonnen werden, da sie ebenfalls die räumliche Information der Quelle enthalten.

Schritt 2: Da alle möglichen Punkte, die an einem Mikrophonpaar den selben Laufzeitunterschied verursachen, auf einer Hyperbel liegen, kann die zweidimensionale Position der Quelle als Schnittpunkt zweier Hyperbeln berechnet werden. Dazu ist es notwendig, den Laufzeitunterschied gemäß Schritt 1 an zwei Mikrophonpaaren zu bestimmen. Analog kann mit drei Mikrophonpaa-

ren durch die Berechnung des Schnittpunkts dreier Hyperboloide die dreidimensionale Position der Quelle bestimmt werden [3]. Die Form der Hyperbeln ist von den Laufzeitunterschieden und der Mikrophongeometrie abhängig. Da die in Kapitel 2 besprochenen Methoden nur Laufzeitunterschiede berechnen können, die ganzzahlige Vielfache der Abtastrate sind, ist die Positionsbestimmung auf diskrete Punkte begrenzt. Dies resultiert in einem Positionsraster, auf das alle Punkte im Raum abgebildet werden. Um eine möglichst hohe räumliche Auflösung zu gewährleisten, muss eine geeignete Mikrophonanordnung gefunden werden. Deshalb werden in Kapitel 3 zwei verschiedene Anordnungen vorgestellt und deren Vor- und Nachteile aufgezeigt.
Im daran anschließenden Kapitel 4 werden einige Szenarien vorgestellt, deren Daten als Grundlage dienen, die besprochenen Lokalisierungs-Algorithmen zu vergleichen und zu bewerten. Um den Vergleich ohne die nichtlineare Verzerrung der Umsetzung auf räumliche Koordinaten auszuführen werden die Ergebnisse auf TDOA-Basis verglichen. Als Maß für die Genauigkeit der Ergebnisse wird die Varianz (mittlerer quadratischer Fehler zwischen den errechneten Werten und den als Referenz genutzten Infrarotdaten der Szenarien) betrachtet.

Kapitel 2

Schätzung der Laufzeitunterschiede

In diesem Kapitel werden drei verschiedene Verfahren beschrieben, die zur Schätzung von Laufzeitunterschieden genutzt werden. Aus den Eingangssignalen eines Mikrophonpaares wird mittels Korrelation (GCC), blinder Systemidenitfikation (AED) oder blinder Quellentrennung (BSS) die relative Zeitverzögerung des direkten Pfades des Quellsignals zwischen den Mikrophonen geschätzt. Dabei ist zu erwähnen, dass die ersten beiden behandelten Algorithmen jeweils nur den Laufzeitunterschied einer Quelle, BSS dagegen (ebenfalls mit zwei Mikrophonen) die Laufzeitunterschiede für zwei Quellen gleichzeitig bestimmen kann.

2.1 Generalized Cross Correlation (GCC)

2.1.1 Grundlagen

Die GCC schätzt den Laufzeitunterschied mittels Korrelation der Mikrophonsignale. Sie wird oft in verhallten Umgebungen genutzt, funktioniert aber bei starker Verhallung aufgrund des Signalmodells

$$x_j(t) = \alpha_j s(t - \tau_j) + b_j(t) \qquad \text{mit} \quad j \in \{1, 2\}, \tag{2.1}$$

das dieser Methode zugrunde liegt, oft nicht mehr zufriedenstellend [2]. Die Mikrophonsignale $x_j(t)$ werden hier als um τ_j verzögerte Varianten des Quellsignals $s(t)$ modelliert und die Verhallung wird völlig vernachlässigt. In dieser Gleichung bezeichnet α_j die Dämpfung und $b_j(t)$ ein additives Rauschsignal am j-ten Mikrophon. Der Laufzeitunterschied $\tau = \tau_1 - \tau_2$ entspricht der relativen Verzögerung zwischen den Mikrophonen. Man ermittelt einen Schätzwert $\hat{\tau}$ für den Laufzeitunterschied τ gemäß

$$\hat{\tau} = \arg\max_{\hat{\tau}} \{\mathcal{F}^{-1}\{\underline{\Phi} \cdot \underline{s}_{x1x2}\}\}, \tag{2.2}$$

indem man die inverse Fourier-Transformation des mit einer Gewichtungsfunktion $\underline{\Phi}$ multiplizierten Kreuzleistungsdichtespektrums $\underline{s}_{x1x2}$ maximiert. Als Varianten können die Kreuzkorrelations-Methode (CC) mit der Gewichtungsfunktion $\underline{\Phi} = 1$ oder die Phasen-Transformation (PHAT) mit $\underline{\Phi} = \frac{1}{|\underline{s}_{x1x2}|}$ angewendet werden [2].

2.1.2 Algorithmus

Der Algorithmus kann durch die Implementierung der folgenden Schritte realisiert werden, wobei FFT für 'Fast Fourier Transformation' und IFFT für 'Inverse Fast Fourier Transformation' stehen.

1. Erfassen von je M Werten für einen Block $\mathbf{x}_j(m)$ mit Blockindex m eines Mikrophonsignals $j = 1, 2$.
2. Falls die Leistung des aktuellen Blocks $m \geq$ Schranke des VAD
   3. Fensterung der aktuellen Blöcke mit Hamming-Fenstern
   4. Berechnung der Kreuzkorrelierten $\mathbf{r}_{x_1x_2}$ der Blöcke
   5. Transformation in den Frequenzbereich $\underline{\mathbf{s}}_{x_1x_2} = \text{fft}(\mathbf{r}_{x_1x_2})$
   6. Multiplikation mit der Gewichtungsfunktion $\underline{\psi} = \underline{\mathbf{\Phi}} \cdot \underline{\mathbf{s}}_{x_1x_2}$
   7. Rücktransformation in den Zeitbereich $\mathbf{\Psi} = \text{ifft}(\underline{\psi})$
   8. Bestimmung des Laufzeitunterschieds $\hat{\tau}$ (in Samples) mittels Gleichung 2.2
9. Falls die Leistung des aktuellen Blocks $m <$ Schranke des VAD
   10. Übernehmen des Wertes für den Laufzeitunterschied des Vorgängerblocks $\hat{\tau}(m) = \hat{\tau}(m-1)$

Tabelle 2.1: GCC-Algorithmus

Da für jeden einzelnen Block ein von den vorhergehenden Ergebnissen unabhängiger Wert für den Laufzeitunterschied ermittelt wird, ist es erforderlich, einen (z.B. leistungsbasierten) Sprach-Aktivitäts-Detektor (VAD = Voice Activity Detector) in den Schritten 2 und 9 zu nutzen. Als Begründung hierfür kann ein Block betrachtet werden, der auf Grund einer Sprechpause keine signifikanten Signalanteile aus Richtung der Sprecherposition enthält. Nutzt man diesen Block zur Quellenlokalisierung, so ist das Ergebnis hauptsächlich vom Hintergrundrauschen bestimmt und kann daher sehr weit vom wahren Wert abweichen.

Nutzt man die CC-Methode, so können die Schritte 5 bis 7 entfallen und direkt der Laufzeitunterschied bestimmt werden, indem $\hat{\tau} = \arg\max_{\hat{\tau}} \mathbf{r}_{x_1x_2}$ bestimmt wird (Schritt 8).

2.2 Adaptive Eigenwertzerlegung (AED)

2.2.1 Grundlagen

Die in [2] veröffentlichte Methode der adaptiven Eigenwertzerlegung basiert auf dem Prinzip der blinden Systemidentifikation mittels Statistik zweiter Ordnung [13]. Die Mikrophonsignale $x_j(t)$ werden als mit den Impulsantworten $h_j(t)$ gefilterte Versionen des Quellsignals $s(t)$ modelliert:

$$x_j(t) = s(t) * h_j(t) + b_j(t) \qquad \text{mit} \quad i \in \{1, 2\} \tag{2.3}$$

Im rauschlosen Fall $b_j(t) = 0$ können die Mikrophonsignale durch Filterung ineinander überführt werden als

$$x_1(t) * h_2(t) = s(t) * h_1(t) * h_2(t) = x_2(t) * h_1(t). \tag{2.4}$$

Bringt man beide Terme auf eine Seite und überführt diese Gleichung von der kontinuierlichen Zeit t in die diskrete Zeit n und nimmt dabei Impulsantworten endlicher Länge an, so erhält man die Gleichung

$$\mathbf{x}_1^T(n)\mathbf{h}_2 - \mathbf{x}_2^T(n)\mathbf{h}_1 = 0. \tag{2.5}$$

Da die Impulsantworten $\mathbf{h_i}$ nicht bekannt sind, werden sie durch Schätzungen $\hat{\mathbf{h}_\mathbf{i}}$ ersetzt und das Fehlersignal

$$e(n) = \mathbf{x}_1^T(n)\hat{\mathbf{h}}_2 - \mathbf{x}_2^T(n)\hat{\mathbf{h}}_1 \tag{2.6}$$

definiert. Das Systemidentifikations-Problem kann nun gelöst werden, indem durch Fehlerminimierung die Impulsantworten adaptiert werden. Ein Blockschaltbild von diesem Algorithmus ist in Abbildung 2.1 dargestellt. Als Ansatz zur Realisierung dieses Algorithmus wird in [2] die Minimierung des

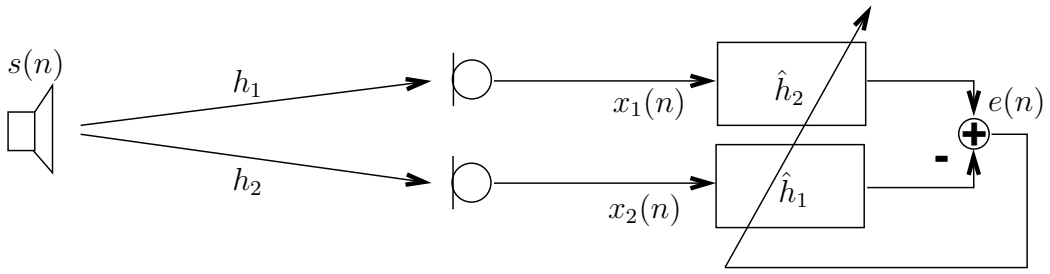

Abbildung 2.1: Blockdiagramm der adaptiven Eigenwertzerlegung

Erwartungswerts des quadratischen Fehlers (Least Mean Squared Error) gemäß

$$\min_{\hat{h}_1,\hat{h}_2} E\{e^2(n)\} \tag{2.7}$$

verwendet. In dieser Gleichung bezeichnet $E\{\cdot\}$ den mathematischen Erwartungswert und $e(n)$ den Fehler entsprechend Gleichung 2.6.

Auch nach mehreren Blöcken können die geschätzten Impulsantworten beschränkter Länge die unendlich langen, wahren Impulsantworten nicht exakt reproduzieren. Da jedoch nur der Laufzeitunterschied des direkten Pfades von Interesse ist, nicht aber die komplette Impulsantwort, hat dies keinen negativen Einfluss auf das Ergebnis der Schätzung.

Unter Verwendung der Gleichungen 2.6 und 2.7 sollen die geschätzten Impulsantworten sukzessive an die wahren Impulsantworten angenähert werden. Aus dem Fehlersignal, den Mikrophonsignalen $\mathbf{x}_i(n)$ und dem letzten Schätzwert wird die neue Abschätzung der Impulsantworten berechnet. Mit der Abkürzung

$$\mathbf{u} = \begin{bmatrix} \mathbf{h}_2 \\ -\mathbf{h}_1 \end{bmatrix} \tag{2.8}$$

für den Vektor aus den aneinandergereihten Impulsantworten und der Definition $\mathbf{x}(n) = [\mathbf{x}_1^T(n)\mathbf{x}_2^T(n)]^T$ wurde in [2] die Update-Gleichung

$$\mathbf{u}(n+1) = \frac{\mathbf{u}(n) - \mu\left(e(n)\mathbf{x}(n) - e^2(n)\mathbf{x}(n)\right)}{\|\mathbf{u}(n) - \mu\left(e(n)\mathbf{x}(n) - e^2(n)\mathbf{x}(n)\right)\|} \tag{2.9}$$

hergeleitet. Der Parameter μ (Schrittweite) ist für die Geschwindigkeit der Konvergenz entscheidend, kann aber auch zur Divergenz führen, falls er zu groß gewählt wird [3]. Es hat sich im Verlauf dieser Arbeit bei mehreren Versuchen mit unterschiedlicher Schrittweite und Filterlänge gezeigt, dass bei größeren Filterlängen auch eine größere Schrittweite gewählt werden kann, ohne den Algorithmus instabil werden zu lassen. Als Werte haben sich in dieser Arbeit $0 < \mu < 0.2$ als geeignet erwiesen.
Um den Algorithmus zu verbessern und die Rechenkomplexität zu senken wurden in [2] und [3] die folgenden weiteren Schritte unternommen:

- Vernachlässigung der Terme bei $e^2(n)$
- Blockverarbeitung: statt diskreter Zeit n Blockzeit m
- Verwendung eines Frequenzbereichs-Algorithmus
- Nutzen der Overlap-Save-Methode
- Vernachlässigung einer Constraint-Matrix im Frequenzbereichs-Algorithmus
- Normierung der Schrittweite μ mittels Division durch die Signalenergie des jeweiligen Frequenzpunktes: $\mu_{norm}(m) = \mathrm{diag}\{\frac{\mu}{\mathbf{P}(m)}\}$
- Einführung eines Vergessensfaktors λ

Die daraus resultierenden Update-Gleichungen

$$\underline{\hat{\mathbf{h}}}_{\mathbf{2}}(m+1) = \underline{\hat{\mathbf{h}}}_{\mathbf{2}}(m) - 2\mu_{norm}(m)(1-\lambda)\underline{\mathbf{X}}_2^H(m)\underline{\mathbf{e}}(m)\text{und} \tag{2.10}$$

$$-\underline{\hat{\mathbf{h}}}_{\mathbf{1}}(m+1) = -\underline{\hat{\mathbf{h}}}_{\mathbf{1}}(m) - 2\mu_{norm}(m)(1-\lambda)\underline{\mathbf{X}}_1^H(m)\underline{\mathbf{e}}(m) \tag{2.11}$$

entsprechen dem UFDAF-Algorithmus (Unconstrained Frequency-Domain Adaptive Filtering) [11]. Die unterstrichenen Größen bezeichnen Frequenzbereichsgrößen.

2.2.2 Algorithmus

Der in [8] veröffentlichte echtzeitfähige Algorithmus wurde gemäß Tabelle 2.3 so erweitert, dass er zur Bestimmung der Laufzeitunterschiede genutzt werden kann.

Für die Performance des Algorithmus ist die Initialisierung der Impulsantworten von großer Bedeutung [3]. Sie werden im Zeitbereich mit Nullen initialisiert, wobei eine Impulsantwort (hier $\hat{\mathbf{h}}_2$) an einer Stelle (in der Mitte des Blocks) auf den Wert 1 gesetzt wird (Schritt 2). Dies repräsentiert den direkten Pfad von der Quelle zum einen Mikrophon. Würden beide Impulsantworten nur mit Nullen initialisiert werden, so würde die Adaption nie starten, da der Fehler entsprechend Gleichung 2.6 von Beginn an Null wäre. Im Anschluss an die Initialisierung wird dieser Fehler bestimmt (Schritte 3-9) und mit den konjugiert komplexen Mikrophonsignalen (Schritt 10) multipliziert. Mit dem Resultat dieser Operation wird das sogenannte Gradienten-Constraint durchgeführt [11]. Dieses sorgt mittels IFFT, zu Null setzten der zweiten Hälfte des Blocks und anschließender FFT (Schritt 13) dafür, dass das richtige Filterupdate entsprechend dem Zeitbereich durchgeführt wird.

In Schritt 11 wird der Vektor $\underline{\mathbf{P}}_j$ mit der Leistung der Frequenzpunkte bestimmt und anschließend die Schrittweite (Schritt 12) durch diesen Vektor normiert. Die Konstante δ dient zur Regulierung, falls in einem Frequenzpunkt die Leistung sehr gering ist. Die Filter werden schließlich durch das Ergebnis des Gradienten-Constraints zusammen mit der normierten Schrittweite und dem Vergessensfaktor λ adaptiert (Schritt 14). Zuletzt werden die

Filter in den Zeitbereich transformiert (Schritt 15), um den Laufzeitunterschied $\hat{\tau}$ zu berechnen (Schritt 16). Dazu werden die Positionen der Maxima von $\hat{\mathbf{h}}_2$ und $\hat{\mathbf{h}}_1$ gesucht, da die Differenz dieser Positionen dem Laufzeitunterschied $\hat{\tau}$, gemessen in Abtastwerten, entspricht. Es ist darauf zu achten, dass bei der Adaption $-\hat{\mathbf{h}}_1$ ermittelt wird und deshalb entweder das Minimum gesucht werden muss oder die erhaltene Impulsantwort mit -1 zu multiplizieren ist.

Um diesen Algorithmus zusätzlich zu verbessern kann analog zur GCC-Methode ein Sprach-Aktivitäts-Detektor (VAD) genutzt werden. Dazu muss Tabelle 2.2 durch die Schritte 2, 9 und 10 aus Tabelle 2.1 erweitert werden indem Schritt 2 (von GCC) zwischen Schritt 1 und 2 eingefügt und die Schritte 9 und 10 (von GCC) am Ende angehängt werden.

2.3 Blinde Quellentrennung (BSS)

In diesem Abschnitt wird eine Methode vorgestellt, mit der die Laufzeitunterschiede mehrerer Quellen simultan bestimmt werden können. Analog zur adaptiven Eigenwertzerlegung [2], die auf den Ergebnissen der blinden Systemidentifikation [13] aufbaut und diese zur Bestimmung der Laufzeitunterschiede nutzt, wird im Folgenden die blinde Quellentrennung [4] dementsprechend erweitert, dass sie zur TDOA-Bestimmung verwendet werden kann.

2.3.1 Grundlagen

Die BSS trennt statistisch unabhängige Signale mittels Independent Component Analysis (ICA). Entsprechend [4] und [8] kann folgende Problemstellung (hier für 2 Quellen und 2 Mikrophone) gelöst werden. Die Quellensignale $s_i(t), i = 1, 2$ werden mit einem linearen MIMO-System (multiple input mul-

1.	Erfassen von je N = 2M Werten für einen Block $\mathbf{x}_j$ eines Mikrophonsignals $j = 1, 2$. (bestehend aus einem alten und einem neuen Block mit je M Werten)
2.	Initialisierung der Impulsantworten $\mathbf{u}_j$ im Zeitbereich
3.	Transformation der Impulsantworten in den Frequenzbereich $\underline{\mathbf{u}}_j = \mathrm{FFT}(\mathbf{u}_j)$ und $\underline{\mathbf{X}}_j = \mathrm{diag}\{\mathrm{FFT}(\mathbf{x}_j)\}$
4.	Berechnung der Ausgangssignale $\underline{\mathbf{y}}_j = \underline{\mathbf{X}}_j \cdot \underline{\mathbf{u}}_j$
5.	Transformation in den Zeitbereich $\mathbf{y}_j = \mathrm{IFFT}(\underline{\mathbf{y}}_j)$
6.	Verwerfen der ersten M Werte und speichern der letzten M Werte $\mathbf{y}_j \leftarrow \mathbf{y}_j(M+1:N)$
7.	Fehlersignal berechnen $\mathbf{e} = \mathbf{y}_1 + \mathbf{y}_2$ (Addition, da $\mathbf{u}_2 = -\mathbf{h}_1$)
8.	Dem Fehlersignal einen Block mit M Nullen vorstellen $\mathbf{e} \leftarrow [\mathbf{0}_{1xM}\ \mathbf{e}^T]^T$
9.	Transformation in den Frequenzbereich $\underline{\mathbf{e}} = \mathrm{FFT}(\mathbf{e})$
10.	Bestimmen der konjugiert komplexen Blöcke $\underline{\mathbf{X}}_j^* = \mathrm{conj}\underline{\mathbf{X}}_j$
11.	Berechnen der Signalleistung $\underline{\mathbf{P}}_j = 0.1\underline{\mathbf{P}}_j + 0.9(\underline{\mathbf{X}}_j\underline{\mathbf{X}}_j^*)$
12.	Normierung der Schrittweite in jedem Frequenzpunkt ν mittels Division durch dessen Leistung $\mu_{norm}^{(\nu)} = \frac{\mu}{\max(\underline{P}_j^{(\nu)}, \delta)}$
13.	Gradienten-Constraint entsprechend [11]
14.	Update der Filter entsprechend den Gleichungen 2.11 und 2.10, wobei $\underline{\mathbf{u}}_2 = -\underline{\mathbf{h}}_1$, $\underline{\mathbf{u}}_1 = \underline{\mathbf{h}}_2$ entsprechen und statt $\underline{\mathbf{X}}_p^H\underline{\mathbf{e}}$ das Ergebnis des Gradienten-Constraints verwendet wird.
15.	Transformation der Filter in den Zeitbereich $\mathbf{u}_p = \mathrm{IFFT}(\underline{\mathbf{u}}_p)$
16.	Bestimmung des Laufzeitunterschieds als Differenz der Positionen des Maximums von $\mathbf{u}_1$ und des Minimums von $\mathbf{u}_2$ $\hat{\tau} = \arg\max(\mathbf{u}_1) - \arg\min(\mathbf{u}_2)$

Tabelle 2.2: AED-Algorithmus

tiple output) $\mathbf{H}$ gefiltert und auf die Mikrophonsignale $x_j(t), j = 1, 2$ abgebildet. Dieses Mischsystem $\mathbf{H}$ besteht aus Filtern $\mathbf{h}_{ij}$ mit je M Filterkoeffizienten von der i-ten Quelle zum j-ten Mikrophon. Das Ziel der blinden Quellentrennung ist der Entwurf eines Entmischsystems $\mathbf{W}$ mit den Filterkoeffizienten $\mathbf{w}_{ji}$ vom j-ten Mikrophon zum i-ten Ausgang. Durch dieses Entmischsystem sollen die Quellsignale aus den Mikrophonsignalen extrahiert werden, sodass die Ausgangssignale $y_i(t), i = 1, 2$ im Idealfall bis auf eine verhallungsbedingte Filterung den Quellsignalen $s_i(t)$ entsprechen. Der prinzipielle Aufbau ist in Abbildung 2.2 dargestellt.

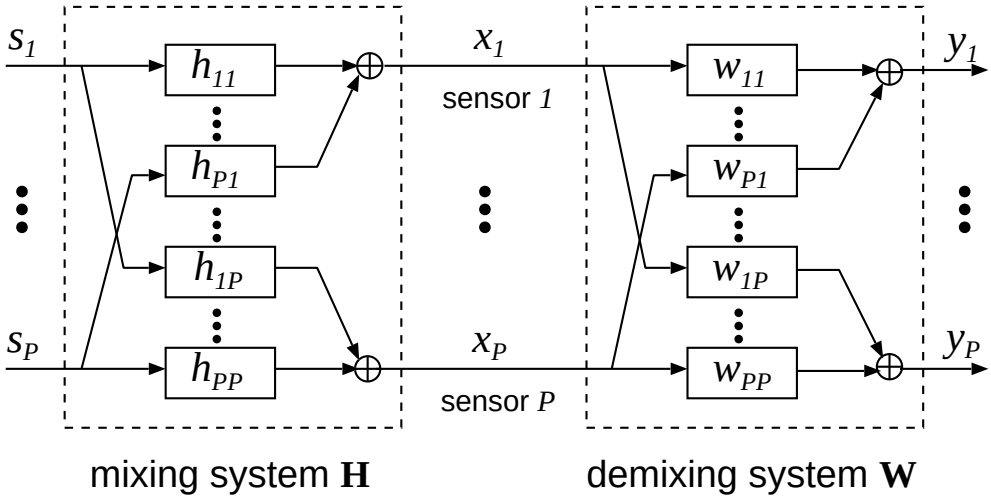

Abbildung 2.2: Lineares MIMO-Modell für die blinde Quellentrennung

Bei den ursprünglichen Algorithmen der blinden Quellentrennung für Faltungsmixturen wurde meist eine Frequenzbereichs-Realisierung genutzt, bei der die Signale binweise voneinander getrennt wurden. Beim Zuordnen der unabhängigen Bins trat jedoch ein Permutationsproblem auf, das mit Reparaturmechanismen rückgängig gemacht werden musste. Um diese Problematik zu umgehen wurde in [4] ein breitbandiger Ansatz vorgestellt, der in

[8] als echtzeitfähiger Algorithmus implementiert wurde und auch in dieser Arbeit genutzt wird. Dieser Algorithmus nutzt die Nichtweißheit und die Nichtstationarität (basierend auf Statistik zweiter Ordnung) aus, um die Ausgangssignale breitbandig voneinander statistisch zu entkoppeln. Das Vorgehen entsprechend [8] ist im Folgenden zusammengefasst.
Das Signal am q-ten Ausgang wird als Matrix

$$\mathbf{Y}_q(m) = \begin{bmatrix} y_q(mL) & \cdots & y_q(mL-L+1) \\ y_q(mL+1) & \cdots & y_q(mL-L+2) \\ \vdots & \ddots & \vdots \\ y_q(mL+N-1) & \cdots & y_q(mL-L+N) \end{bmatrix} \tag{2.12}$$

beschreiben. Des Weiteren wird die Faltung der Mikrophonsignale mit den Filtern des Entmischsystems formuliert als

$$\mathbf{Y}_q(m) = \sum_{p=1}^{P} \mathbf{X}_p(m)\mathbf{W}_{pq}. \tag{2.13}$$

Dabei ist m der Blockindex und N die Blocklänge. Die NxL Matrix $\mathbf{Y}_q(m)$ berücksichtigt L Zeitverzögerungen, um die Eigenschaft der Nichtweißheit auszunutzen. Um lineare Faltung für alle Elemente von $\mathbf{Y}_q(m)$ sicherzustellen, werden die Matrizen $\mathbf{X}_p(m)$ und $\mathbf{W}_{pq}$ festgelegt als

$$\mathbf{X}_p(m) = \begin{bmatrix} x_p(mL) & \cdots & x_p(mL-2L+1) \\ x_p(mL+1) & \cdots & x_p(mL-2L+2) \\ \vdots & \ddots & \vdots \\ x_p(mL+N-1) & \cdots & x_p(mL-2L+N) \end{bmatrix} \text{ und} \tag{2.14}$$

$$\mathbf{W}_{pq} = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{pq,0} & \ddots & \vdots \\ \vdots & w_{pq,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{pq,0} \\ 0 & w_{pq,L-1} & \ddots & w_{pq,1} \\ \vdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{pq,L-1} \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{2.15}$$

Dabei haben die Matrizen $\mathbf{X}_p(m)$ Toeplitz-Struktur und die Matrizen $\mathbf{W}_{pq}$ sind Sylvester-Matrizen. Zusammenfassend kann als Darstellung für alle Kanäle die Schreibweise

$$\mathbf{Y}(m) = \mathbf{X}_(m)\mathbf{W} \tag{2.16}$$

mit den Matrizen

$$\mathbf{Y}(m) = [\mathbf{Y}_1(m), \cdots, \mathbf{Y}_P(m)], \tag{2.17}$$

$$\mathbf{X}(m) = [\mathbf{X}_1(m), \cdots, \mathbf{X}_P(m)] \quad \text{und} \tag{2.18}$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1P} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{P1} & \cdots & \mathbf{W}_{PP} \end{bmatrix} \tag{2.19}$$

angegeben werden. Die Kostenfunktion kann laut [8] dargestellt werden als

$$\Im(m) = \sum_{i=0}^{m} \beta(i,m)\{\log\det \operatorname{bdiag}\mathbf{Y}^H(i)\mathbf{Y}(i) - \log\det \mathbf{Y}^H(i)\mathbf{Y}(i)\}. \tag{2.20}$$

Mit der Gewichtsfunktion $\beta(i,m)$ kann der Algorithmus zwischen online- und offline-Modus variiert werden, wobei auch eine block-online-Variante möglich ist, die im Weiteren genutzt wird. Der Operator bdiag setzt alle Untermatrizen zu Null, die nicht auf der Diagonalen der Matrix liegen. Er wird genutzt um die Kostenfunktion zu Null zu bringen, da hierfür die Kreuzkorrelierten

der Ausgangssignale über alle Zeitverzögerungen Null sein müssen. Dies kann im folgendem Bild veranschaulicht werden.

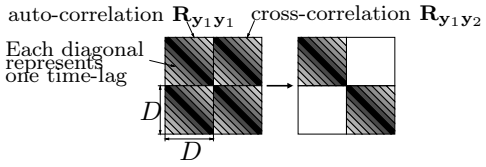

Abbildung 2.3: Darstellung der Kostenfuntion für den 2×2 - Fall

Die Ableitung der Kostenfunktion 2.20 mit dem natürlichen Gradienten nach $\mathbf{W}$ liefert einen iterativen Algorithmus mit dem Koeffizienten-Update

$$\nabla_{\mathbf{W}}^{NG} \Im(m) = 2 \sum_{i=0}^{m} \beta(i,m) \mathbf{W}(i) \{ \mathbf{R}_{\mathbf{yy}}(i) - \text{bdiag} \mathbf{R}_{\mathbf{yy}}(i) \} \text{bdiag}^{-1} \mathbf{R}_{\mathbf{yy}}(i), \tag{2.21}$$

wobei die Korrelationsmatrizen $\mathbf{R}_{\mathbf{yy}}$ aus den Untermatrizen für jeden Kanal $\mathbf{R}_{\mathbf{y_p y_q}}(m) = \mathbf{Y}_p^H(m) \mathbf{Y}_q(m)$ bestehen.
Da die Berechnung dieses Updates eine sehr hohe Rechenlast bedeutet, werden in [8] folgende Näherungen eingeführt, um die Echtzeitfähigkeit des Algorithmus zu gewährleisten.

- Abschätzung der Korrelationsmatizen $\mathbf{R}_{\mathbf{y_p y_q}}(m)$ mit der Korrelations-Methode. Entsprechend dem Ansatz aus der linearen Sprach-Prädiktion in [10] kann alternativ auch die Kovarianz-Methode gewählt werden, die zwar genauer arbeitet, dafür aber auch rechenaufwändiger ist.

- Für die Normierung werden die Korrelationsmatrizen durch die Leistung des Ausgangssignals angenähert: $\mathbf{R}_{\mathbf{y_p y_q}}(m) = \sigma_{y_q}^2(m) \mathbf{I}$
  Durch diesen Schritt erspart man sich die Inversion der Matrizen $\mathbf{R}_{\mathbf{y_p y_q}}(m)$

- Durch die Realisierung einer effizienteren Multiplikation der Matrizen kann die Rechenlast weiter gesenkt werden. Dabei nutzt man die regelmäßigen Strukturen der Sylvester- und Toeplitz-Matrizen aus. Das ka-

nalweise ausgeführte Matrixprodukt der Sylvestermatrix $\mathbf{W}_{pt}$ mit den Toeplitz-Matrizen $\frac{\mathbf{R_{y_p y_q}}(m)}{\sigma^2_{y_q}(m)}$ kann als lineare Faltung der Filterkoeffizienten mit jeder Spalte von $\frac{\mathbf{R_{y_p y_q}}(m)}{\sigma^2_{y_q}(m)}$ interpretiert werden und daher effizient mit einer FFT-Operation berechnet werden.

Durch die oben genannten Schritte kann die Rechenkomplexität auf eine Mächtigkeit von $\mathcal{O}(\log L)$ gesenkt werden.

2.3.2 Algorithmus

Der in [8] beschriebene echtzeitfähige Block-Online-Zeitbereich-Algorithmus kann erweitert werden, um aus den Filtern die Laufzeitunterschiede zweier Quellen simultan zu bestimmen. Die prinzipiell zu implementierenden Schritte sind in Tabelle 2.3 zusammengefasst, wobei von jeweils 2 Quellen und 2 Mikrophonen ausgegangen wurde. Die Schritte 1, 2, 3 und 11 werden online durchgeführt, während die Schritte 4 bis 10 offline bestimmt werden. Die letzten Schritte 12 bis 15 entsprechen der Erweiterung und bestimmen aus den online-Filtern die Laufzeitunterschiede.

In Schritt 1 werden online-Blöcke x_p mit $KL + N$ Werten erfasst. Diese online-Blöcke werden im offline-Teil in je K offline-Blöcke unterteilt und separat voneinander durch a_{max}-faches Durchlaufen der Iteration adaptiert. Durch die simultane Verarbeitung von K offline-Blöcken im online-Teil wird die Nichtstationarität der Signale ausgenutzt.

Die Entmischfilter $\mathbf{w}_{ji}$ sind analog zu den Impulsantworten der adaptiven Eigenwertzerlegung geeignet zu initialisieren (Schritt 2). Hierzu werden in der hier verwendeten Realisierung alle Filtertaps mit Nullen belegt und die Filter $\mathbf{w}_{11}$ und $\mathbf{w}_{22}$ an der ersten Stelle auf den Wert 1 gesetzt. Diese Einheitsimpulse repräsentieren jeweils den direkten Pfad zwischen einer Quelle und einem Mikrophon.

Aus den Filtern und den offline-Blöcken wird (Schritt 4) anschließend ein Ausgangssignal bestimmt, um die Kreuzkorrelationsmatrix bilden zu können (Schritt 6). Diese wiederum wird durch die Signalenergie (Schritt 5) normalisiert (Schritt 7) und anschließend die Änderung $\Delta \mathbf{W}_{ji}^{a}$ der a-ten Iteration als Matrixprodukt entsprechend Schritt 8 berechnet. Die Änderungen werden über alle a_{max} Iterationen aufsummiert (Schritt 9) und für das eigentliche Update im offline-Teil (Schritt 10) zusammen mit der offline-Schrittweite μ_{off} genutzt.

Das Update aus dem offline-Teil (Schritt 10) wird anschließend als Ausgangspunkt für den online-Teil genutzt. In Schritt 11 werden die Filterkoeffizienten $\mathbf{w}_{ji}$ durch die rekursiven Update-Gleichungen des online-Teils analog zu AED mit den Parametern Schrittweite μ und Vergessensfaktor λ adaptiert.

Die so gewonnenen Filterkoeffizienten $\mathbf{w}_{ji}$ können genutzt werden, um die Signale voneinander zu trennen oder, wie in diesem Fall, die Laufzeitunterschiede zu schätzen. Dazu wird erneut vom Misch- und Entmischsystem entsprechend Abbildung 2.2 ausgegangen. Die Ausgangssignale $y_i(t), i = 1, 2$ entsprechen im Idealfall bis auf eine verhallungsbedingte Filterung den Quellsignalen $s_i(t)$. Nun wird als Näherung davon ausgegangen, dass die Filter $\mathbf{w}_{ji}$ des Entmischsystems die Filterung mit Impulsantorten $\mathbf{h}_{ij}$ komplett rückgängig machen. Dies entspricht blinder Quellentrennung und blinder Enthallung [5]. Durch den verwendeten Algorithmus erreicht man zwar keine blinde Enthallung, da aber (analog zu AED) nicht die kompletten Impulsantworten, sondern nur die Position des direkten Pfades zur Schätzung der Laufzeitunterschiede von Interesse ist, kann diese Näherung gemacht werden. In der Umsetzung bedeutet dies, dass Misch- und Entmischsystem eine Einheitsmatrix entsprechend $\hat{\mathbf{H}}\mathbf{W} = \mathbf{I}$ bilden und daher das Mischsystem durch $\hat{\mathbf{H}} = \mathbf{W}^{-1}$ angenähert wird (Schritt 13). Um die Inversion effizient zu gestalten, wird

sie im Frequenzbereich vorgenommen (Schritt 12). Dabei ist es sehr wichtig, die Fourier-Transformation mit einer wesentlich längeren als der eigentlichen Filterlänge durchzuführen, da die Impulsantworten sonst verfälscht werden. Betrachtet man nun die in den Zeitbereich zurücktransformierten (Schritt 14) geschätzten Impulsantworten $\hat{\mathbf{h}}_{11}$ und $\hat{\mathbf{h}}_{12}$ von einer Quelle zu beiden Mikrophonen, so kann wie bei der AED-Methode der Laufzeitunterschied bestimmt werden (Schritt 15). Im Gegensatz zur adaptiven Eigenwertzerlegung ist jedoch auch der Laufzeitunterschied der zweiten Quelle bestimmbar, indem die beiden anderen Impulsantworten $\hat{\mathbf{h}}_{21}$ und $\hat{\mathbf{h}}_{22}$ zur Bestimmung genutzt werden.

1. Erfassen von $KL + N$ neuen Werten an jedem Mikrophon $j = 1, 2$ um Blöcke $\mathbf{x}_j$ zu bilden
2. Initialisieren der Filter im Zeitbereich
3. Generieren von K offline - Blöcken

   Berechne für jede Iteration $a = 1, ..., a_{max}$

   Berechne für jeden offline-Block

   4. Berechne die Ausgangssignale $\mathbf{y}_p$ durch Faltung von $\mathbf{x}_p$ mit den Filterkoeffizienten $\mathbf{w}_{ji}$ des letzten Durchgangs
   5. Berechne die Signalenergie von jedem Block
   6. Berechne die erste Spalte jeder Kreuzkorrelationsmatrix
   7. Normalisiere elementweise die Korrelationsmatrix durch Division durch die Signalenergie
   8. Berechne die Änderung $\Delta\mathbf{W}_{ji}^{a}$ der Filter als Matrixprodukt $\mathbf{W}_{ji}(m)\frac{\mathbf{R}_{\mathbf{y_i y_j}}(m)}{\sigma_{y_j}^2(m)}$

   9. Summiere die Änderung der Filter über alle Iterationen $\Delta\mathbf{W}_{ji} = \sum_{v=1}^{a_{max}} \Delta\mathbf{W}_{ji}^{v}$
   10. Update-Gleichung des offline-Teils mit offline-Schrittweite μ_{off}
11. Berechne das rekursive Update für den online-Teil mit μ und λ
12. Transformieren der Filter in den Frequenzbereich $\underline{\mathbf{w}}_{pq} = \mathrm{FFT}\{\mathbf{w}_{pq}\}$
13. Annäherung der Impulsantworten durch die invertierten Entmischfilter $\underline{\hat{\mathbf{H}}}_{(\nu)} = \underline{\mathbf{W}}_{(\nu)}^{-1}$, wobei $\underline{\mathbf{W}}_{(\nu)}$ die Entmischmatrix im ν-ten Frequenzbin bezeichnet.
14. Rücktransformation der Impulsantworten $\hat{\mathbf{h}}_{pq} = \mathrm{IFFT}\ \{\underline{\hat{\mathbf{h}}}_{pq}\}$
15. Bestimmen der Laufzeitunterschiede der Quellen als Differenz der Positionen der Maxima bei den Impulsantworten von einer Quelle zu beiden Mikrophonen: $\hat{\tau}_p = \arg\max(\hat{\mathbf{h}}_{p1}) - \arg\max(\hat{\mathbf{h}}_{p2}), p = 1, 2$

Tabelle 2.3: BSS-Algorithmus

Kapitel 3

Positionsbestimmung

Zur Bestimmung der räumlichen Position einer Quelle entsprechend Abbildung 1.1 müssen die Laufzeitunterschiede an mehreren Mikrophonpaaren bestimmt werden und mit der Mikrophongeometrie zusammen verrechnet werden. Dazu werden im Folgenden zwei Mikrophonpaare betrachtet, um die zweidimensionale Position einer Quelle (x_s,y_s) bestimmen zu können. Prinzipiell kann unter Verwendung eines dritten Mikrophonpaares, das mindestens ein Mikrophon mit $y_j \neq 0$ oder $z_j \neq 0$ enthält, auch die dreidimensionale Position bestimmt werden [3].

3.1 Geometrische Umsetzung

Alle möglichen Positionen, die an einem Mikrophonpaar den selben Laufzeitunterschied hervorrufen, können bestimmt werden, indem die Gleichung

$$\tau_{12} c_s = \sqrt{(x_s - x_1)^2 + (y_s - y_1)^2} - \sqrt{(x_s - x_2)^2 + (y_s - y_2)^2} \qquad (3.1)$$

ausgewertet wird. Die linke Seite dieser Gleichung entspricht dem Wegunterschied zwischen der Quelle und den Mikrophonen, der wiederum dem mit der

Schallgeschwindigkeit c_s multiplizierten Laufzeitunterschied τ_{12} entspricht. Die Variablen x_j und y_j stehen für die x- und y- Koordinaten der Mikrophone $j = 1, 2$. Löst man nun diese Gleichung nach y_s auf und setzt verschiedene

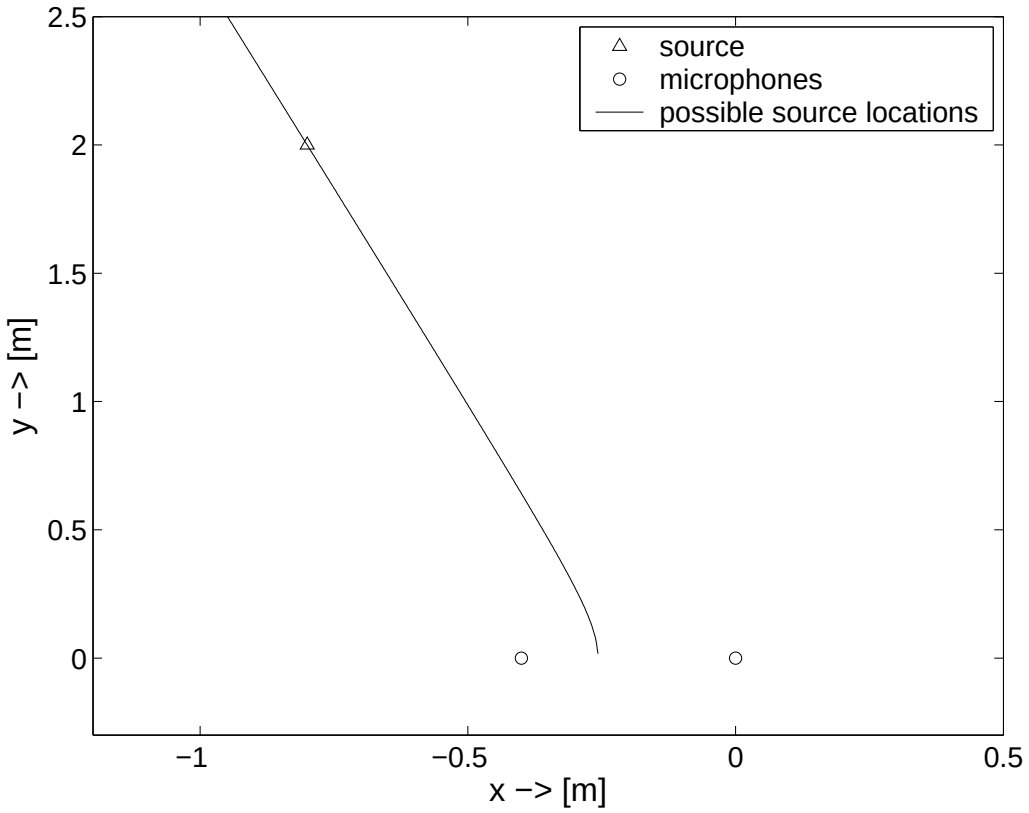

Abbildung 3.1: Mögliche Quellpositionen für ein Mikrophonpaar

Werte für x_s ein, so ergibt sich bei einem bestimmten Laufzeitunterschied τ_{12} eine Hyperbel entsprechend Abbildung 3.1, auf der alle möglichen Quellpositionen liegen. Um nun die richtige Position bestimmen zu können, muss ein zweites Mikrophonpaar betrachtet werden. Analog zu Gleichung 3.1 ergibt sich

$$\tau_{34} c_s = \sqrt{(x_s - x_3)^2 + (y_s - y_3)^2} - \sqrt{(x_s - x_4)^2 + (y_s - y_4)^2} \qquad (3.2)$$

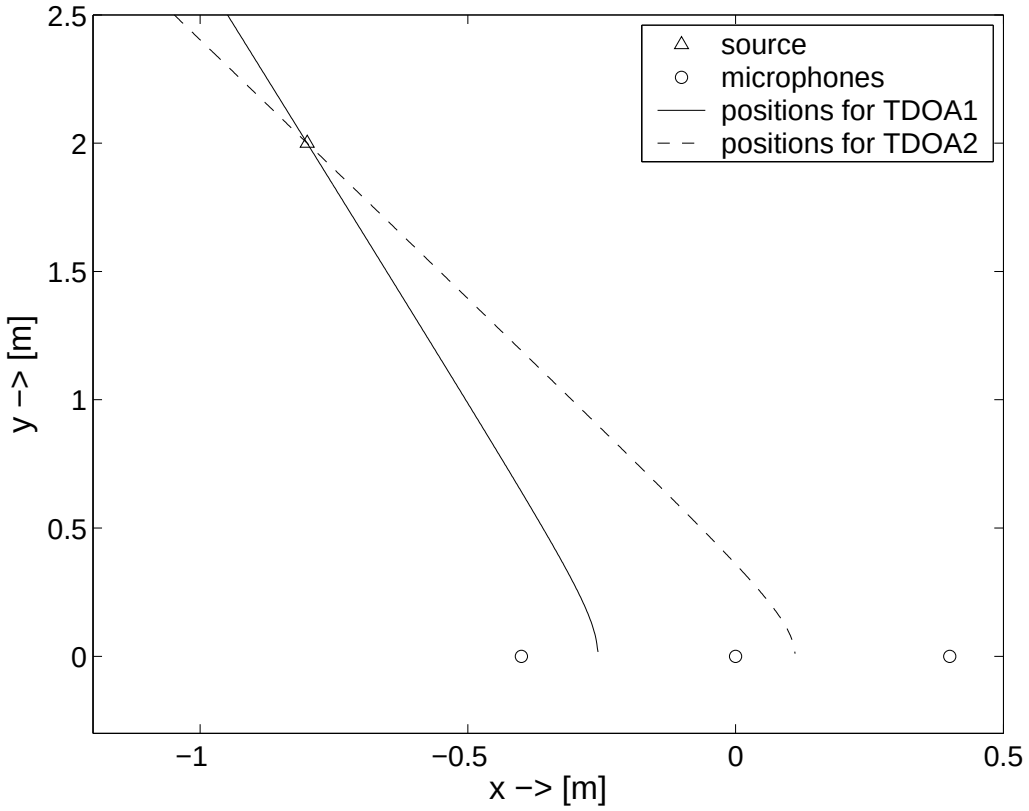

Abbildung 3.2: Bestimmung der Quellpositionen mit zwei Mikrophonpaaren

für das zweite Mikrophonpaar. Da nun mit 3.1 und 3.2 ein Gleichungssystem mit zwei Gleichungen und zwei Unbekannten vorliegt, kann man die zweidimensionale Position der Quelle bestimmen. Geometrisch entspricht dies dem Schnittpunkt der beiden Hyperbeln. Dies ist in Abbildung 3.2 dargestellt, wobei das mittlere Mikrophon sowohl für das linke, als auch für das rechte Mikrophonpaar genutzt wurde. Es ist anzumerken, dass sich je ein Schnittpunkt vor ($y_s > 0$) und hinter ($y_s < 0$) dem Mikrophonarray ergibt, wobei der Schnittpunkt im negativen y-Bereich vernachlässigt werden kann, da das Mikrophonarray den Raum nach hinten begrenzt und somit nur positive Werte für y_s sinnvoll sind.

3.2 Mikrophongeometrie

Betrachtet man die Gleichungen 3.1 und 3.2, so ist gut zu erkennen, dass die Mikrophongeometrie entscheidenden Einfluss auf die Berechnung der Positionen hat, weshalb im Folgenden zwei geeignete Mikrophonanordnungen diskutiert werden.

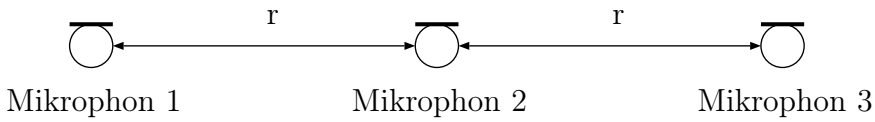

Abbildung 3.3: Mikrophonanordnung 1

Die Anordnung in Abbildung 3.3 nutzt das mittlere Mikrophonsignal doppelt, indem der Laufzeitunterschied zwischen den Mikrophonen 1 und 2 sowie zwischen den Mikrophonen 2 und 3 bestimmt wird. Dies hat neben der Ersparnis eines Mikrophons den Vorteil, dass die Berechnung der Koordinaten stark vereinfacht wird, da bei den Gleichungen 3.1 und 3.2 einige Variablen entfallen, wenn man für die Berechnung den Ursprung des Koordinatensystems auf das mittlere Mikrophon setzt. Jedoch hat diese Anordnung auch einen Nachteil, wie das folgende Problem zeigt.

Da die Verfahren aus Kapitel 2 nur ganzzahlige Vielfache der Abtastrate bestimmen können, ist in Gleichung 3.1 der Laufzeitunterschied τ_{12} durch $\frac{\hat{\tau}_{12}}{f_s}$ zu ersetzen, wobei $\hat{\tau}_{12}$ eine ganze Zahl ist und f_s die Abtastfrequenz darstellt (analoge Ersetzung bei Gleichung 3.2). Als Folge dieser Diskretisierung können die Mikrophonpositionen nicht beliebig genau bestimmt werden, sondern nur noch auf diskrete Punkte im Raum abgebildet werden. In Abbildung 3.4 sind alle möglichen Positionen dargestellt, die bei einer Mikrophonanordnung entsprechend Abbildung 3.3 mit Mikrophonabstand r=16cm bestimmt

werden können. Wie man sehen kann, nimmmt die örtliche Auflösung mit zunehmendem Abstand zwischen Quelle und Mikrophonarray ab.

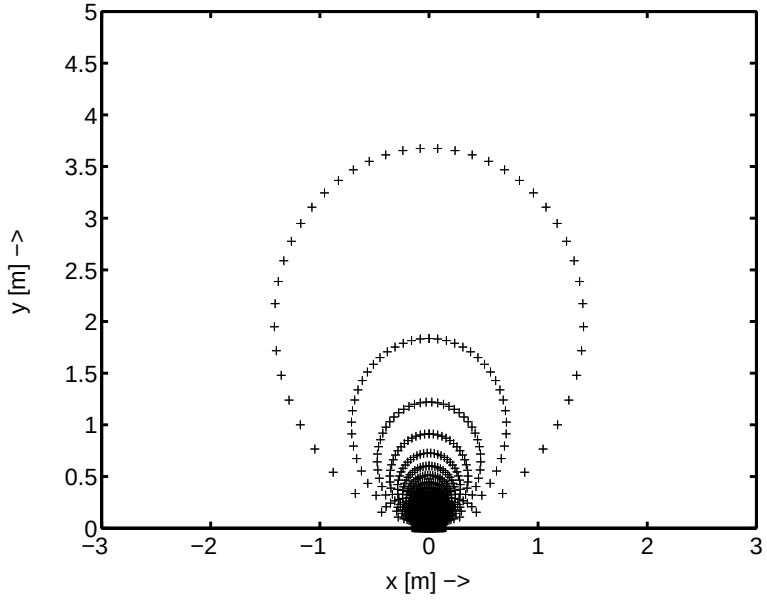

Abbildung 3.4: Positionenraster bei drei Mikrophonen mit je 16 cm Abstand

Um eine möglichst gute Auflösung für den gesamten Raum zu erreichen, kann die Mikrophongeometrie verändert werden. Die erste Möglichkeit besteht darin, den Abstand r zwischen den Mikrophonen zu erhöhen. Bei r=30cm ist die Auflösung erheblich verbessert wie in Abbildung 3.5 zu sehen ist. Der Grund hierfür ist, dass sich der maximale Laufzeitunterschied zwischen den Mikrophonen $\tau_{max} = \frac{c_s r}{f_s}$ erhöht. Mit steigendem τ_{max} erhöhen sich ebenfalls die Anzahl der möglichen TDOA-Werte $N_{TDOA} = 2\tau_{max} + 1$ pro Mikrophonpaar und damit auch die Anzahl der möglichen Positionen $N_{pos} = (N_{TDOA})^2$ erheblich. Der Nachteil des größeren Mikrophonabstandes ist jedoch, dass mit wachsendem r die TDOA-Bestimmung ungenauer wird.
Deshalb wird in Abbildung 3.6 eine Alternative zur ersten Mikrophonanordnung vorgestellt. Bei dieser Anordnung wird kein Mikrophon doppelt genutzt.

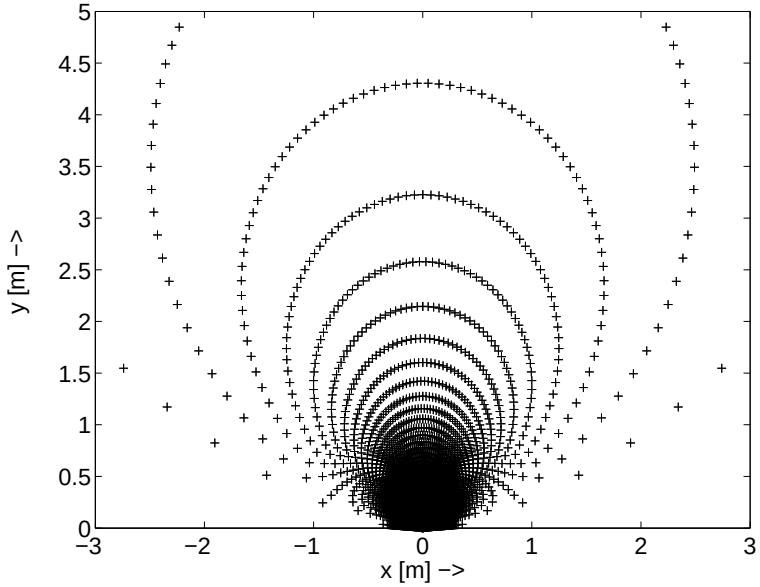

Abbildung 3.5: Positionenraster bei drei Mikrophonen mit je 30 cm Abstand

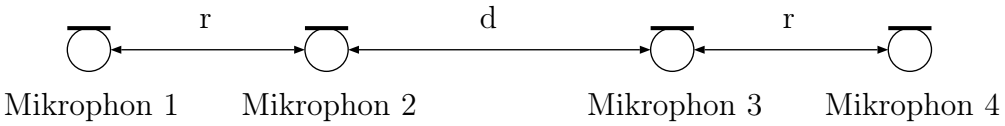

Abbildung 3.6: Mikrophonanordnung 2

Anstatt nun den Abstand r zu erhöhen, wird d vergrößert und somit die Mikrophonpaare weiter auseinander positioniert. Dadurch steigt zwar nicht die Anzahl der möglichen Positionen, jedoch sind die möglichen Positionen besser im Raum verteilt. In Abbildung 3.7 sind die Positionen dargestellt, die mit der Mikrophonanordnung entsprechend 3.6 mit $r = 16$ cm und $d = 32$ cm ermittelt werden können. Diese Anordnung ermöglicht genauere Schätzungen der Laufzeitunterschiede an den Mikrophonpaaren, benötigt allerdings vier Mikrophone und zieht eine kompliziertere Berechnung der Koordinaten nach sich. Für den zweidimensionalen Fall ist dies noch handhabbar, jedoch ist

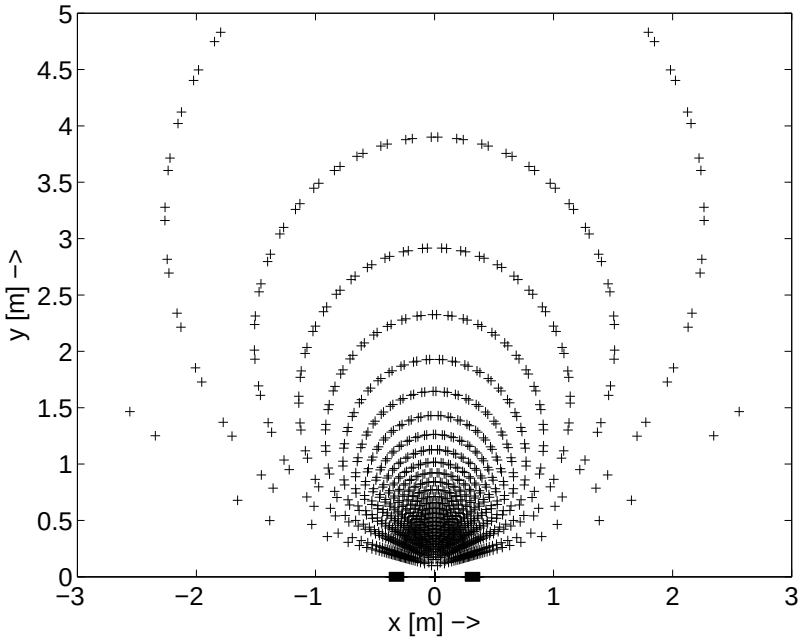

Abbildung 3.7: Positionenraster bei zwei Mikrophonpaaren

im dreidimensionalen Fall die Positionsberechnung extrem aufwendig, da zu viele Variablen in die Berechnung eingehen.

Nach der Betrachtung der Vor- und Nachteile beider Mikrophonanordnungen ist festzustellen, dass für den zweidimensionalen Fall die Mikrophonanordnung 2 (Abb. 3.6) leichte Vorteile gegenüber der Mikrophonanordnung 1 (Abb. 3.3) aufweist. Die größere Varianz der TDOA-Bestimmung bei Mikrophonanordnung 1 fällt bei der Umsetzung auf die räumliche Position jedoch nicht so stark ins Gewicht wie zuerst vermutet. Der Grund liegt darin, dass mehr mögliche Quellenpositionen existieren und deshalb die Abbildung auf eine benachbarte Position weniger Abweichung von der tatsächlichen Quellenposition bedeutet als bei dem gleichen Fehler des Laufzeitunterschieds bei Mikrophonanordnung 2. Bei dreidimensionaler Positionsbestimmung ist da-

her auf Grund der einfacheren Berechnung die mit einem Mikrophon ($y_s \neq 0$ oder $z_s \neq 0$) entsprechend [3] erweiterte Mikrophonanordnung 1 vorzuziehen.

Kapitel 4

Vergleich der Lokalisierungs-Algorithmen

4.1 Szenenbeschreibung

Zur Bestimmung der Genauigkeit der Algorithmen wurden zwei verschiede Szenarien betrachtet. Im ersten Szenario stehen durch Infrarot-Sensoren ermittlete Messdaten zur Verfügung, die als Referenzpositionen herangezogen werden. Daher können feste und auch bewegte Quellen betrachtet werden. Im zweiten Szenario wurden Sprachsignale mit Impulsantworten gefaltet, was örtlich festen Quellen entspricht. Die Länge der betrachteten Signale beträgt im ersten Szenario 29 Sekunden und im zweiten Szenario 9 Sekunden.

4.1.1 Szenario mit Infrarotdaten

Aus der in [6] beschriebenen Datenbank wurden zwei geeignete Szenen (Scene4-take1 und Scene51-take1) ausgewählt, um die Funktion der Algorithmen zu untersuchen. In Szene 4 befindet sich der Sprecher (Speaker2) an einer festen Stelle (geringe Abweichung im Bereich von 20 cm) im Abstand von etwa

zwei Metern vor dem Mikrophonarray. Szene 51 enthält die Aufzeichnung eines Sprechers (Speaker1), der sich zu Beginn 1.5 Meter vor dem Mikrophonarray befindet und sich während dem Sprechen weiter von ihm weg bewegt. In Abbildung 4.1 sind die Trajektorien der beiden Sprecher in eine Abbildung gezeichnet. Die kreisförmigen Markierungen entsprechen den Positionen des betrachteten Mikrophonpaares (mic4 und mic5). Die Aufnahmen dieser Datenbank wurden bei einer Abtastfrequenz von 48 kHz aufgezeichnet und als Nachhallzeit wurde $T_{60} = 700$ ms bestimmt. Des Weiteren ist anzumerken, dass ein Störsignal, das in allen Mikrophonsignalen enthalten war, herausgefiltert wurde bevor die Mikrophonsignale zur Lokalisierung genutzt wurden.

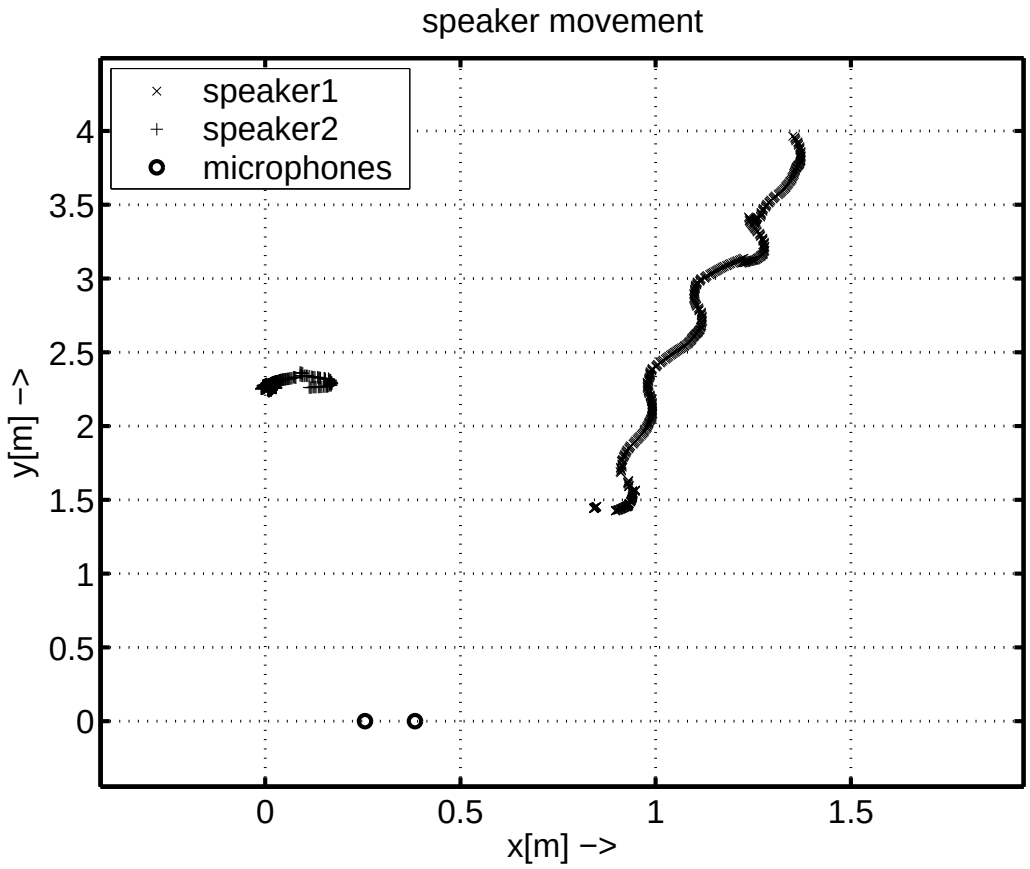

Abbildung 4.1: Trajektorien beider Sprecher von Szenario 1

4.1.2 Szenario mit bekannten Impulsantworten

Als zweites Szenario wurden Sprachsignale mit Impulsantworten gefaltet, die im Multimedia-Raum des Lehrstuhls mit einer Abtastrate $f_s = 48$ kHz aufgenommen wurden. Die Nachhallzeit dieses Raumes beträgt 200 ms. Als Mikrophonanordnug wurden zwei Mikrophonpaare entsprechend Abbildung 3.6 verwendet. In Abbildung 4.2 ist die Positionierung des Mikrophonarrays und der Quellen dargestellt, wobei die Mikrophonpositionen durch Kreise und die Quellpositionen durch Dreiecke gekennzeichnet sind.

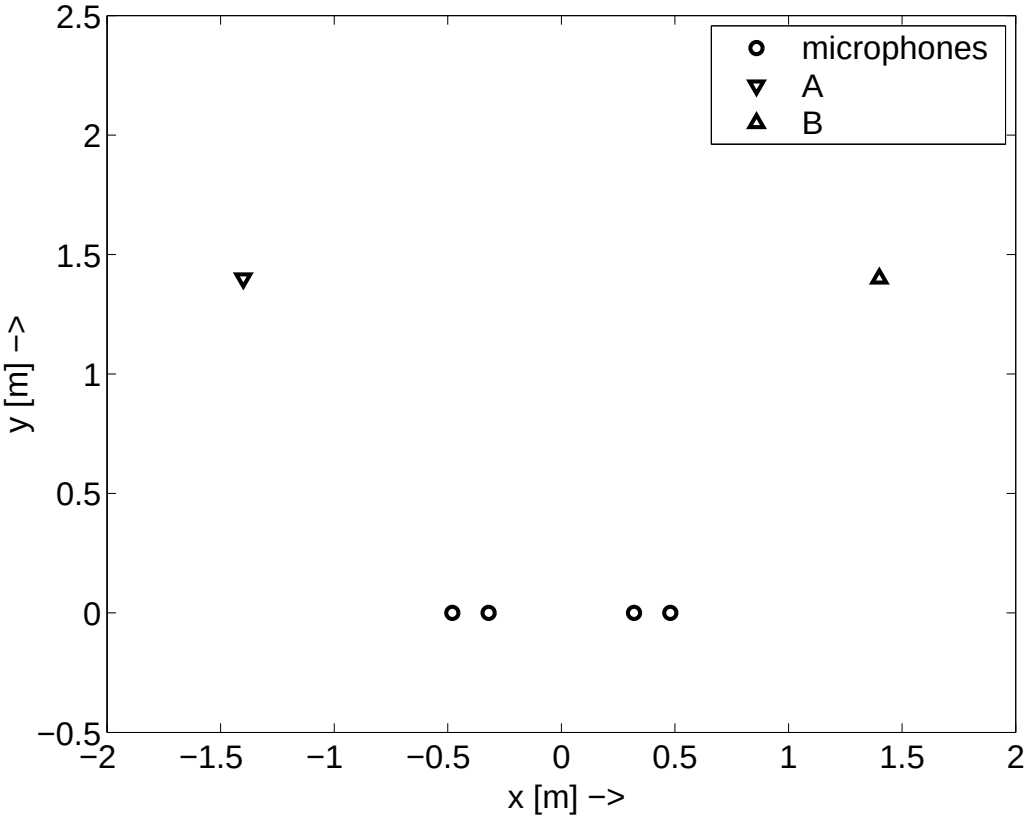

Abbildung 4.2: Szenario mit Impulsantworten

4.1.3 Parameter der Algorithmen

Für die Bestimmung der Laufzeitunterschiede mit der GCC-Methode wurde ein Algorithmus entsprechend Tabelle 2.1 implementiert. Die Filterlänge wurde $M = 1024$ gewählt und die letzen 512 Werte jedes Blocks wurden für den nächsten Block erneut genutzt, was einer Überlappung um den Faktor $\frac{1}{2}$ entspricht. Zur Gewichtung wurde die Kreuzkorrelations-Methode genutzt, da die Gewichtung mittels PHAT bei den getesteten Szenarien zu keiner Verbesserung der Schätzung der Laufzeitunterschiede geführt hat.
Der AED-Algorithmus ist entsprechend Tabelle 2.2 implementiert und mit einer Filterlänge von $M = 1024$ verwendet worden. Die Schrittweite wurde $\mu = 0.09$ und der Vergessensfaktor $\lambda = 0.78$ gewählt. Ebenso wie bei der GCC-Methode wurde auch bei dem AED-Algorithmus ein leistungsbasierter VAD genutzt, um die Ergebnisse zu verbessern.
Zur Lokalisierung mittels BSS wurde der in [8] veröffentlichte und gemäß Tabelle 2.3 erweiterte Algorithmus genutzt. Der Block-Online-Zeitbereich-Algorithmus wurde mit einer Filterlänge von $M = 256$ verwendet. Diese Filterlänge ist für die eigentliche Aufgabe der blinden Quellentrennung zwar zu klein, reicht aber zur Bestimmung der Laufzeitunterschiede aus. Denn analog zum AED-Algorithmus ist nicht das gesamte Filter von Interesse, sondern nur die Position des direkten Pfades. Die Schrittweite wurde sowohl im online- wie auch im offline-Teil $\mu = 0.0002$ gewählt. Des Weiteren wurde die Anzahl der Iterationen im offline-Teil des Algorithmus auf $a_{max} = 10$ begrenzt und der Vergessensfaktor $\lambda = 0.1$ gewählt.

4.2 Vergleich auf TDOA-Basis

Wie in Kapitel 3 gezeigt wurde beinhaltet die Umsetzung der Laufzeitunterschiede auf Koordinaten eine nichtlineare Verzerrung der Ergebnisse, da die Laufzeitunterschiede nur ganzzahlige Werte annehmen können. Deshalb werden, um den Vergleich der Methoden nicht zusätzlich zu verfälschen, die Ergebnisse auf Basis der Laufzeitunterschiede verglichen.

4.2.1 Vorgehensweise

Zur Durchzuführung des Vergleichs [1] müssen zunächst die Algorithmen auf die Mikrophonsignale aufgesetzt werden. Dabei werden für GCC und AED jeweils die Mikrophonsignale einer Quelle genutzt, während bei BSS die Mikrophonsignale beider Quellen eines Szenarios additiv überlagert werden. Dies ist möglich ohne die Ergebnisse negativ zu beeinflussen, da die beiden Szenen aus [6] im gleichen Raum mit gleichen Randbedingungen aufgezeichnet wurden und auch keine Überlagerung der Sprecherpositionen auftritt. Analog kann man beim zweiten Szenario zwei Quellen mit den entsprechenden Impulsantworten falten und ebenfalls die Mikrophonsignale überlagern.
Als zweiten Schritt müssen die Laufzeitunterschiede aus den Infrarotdaten durch Einsetzen der Referenz-Koordinaten in die Gleichungen 3.1 und 3.2 bestimmt werden.
Danach müssen die errechneten Laufzeitunterschiede, die auf Grund unterschiedlicher Blocklänge und Überlappungsfaktoren unterschiedliche zeitliche Auflösung besitzen, auf dieselbe Frequenz mit den Infrarotdaten gebracht werden. Dies wird durch geeignete Überabtastung, Interpolation und anschließende Unterabtastung umgesetzt.

[1] Erstellung der Bilder mit comparison.m

Im nächsten Schritt sind die Laufzeitunterschiede mit den Referenz-TDOAs zu synchronisieren. Dies kann man bewerkstelligen, indem man die geschätzten Laufzeitunterschiede mit den gesamten Referenzdaten vergleicht. Der Ausschnitt der Infrarotdaten, zu dem die geschätzten Laufzeitunterschiede (über den gesamten Ausschnitt gemittelt) den kleinsten quadratischen Fehler aufweisen, dient als Referenz[2] [3]. In Abbildung 4.3 ist der Referenz-Laufzeitunterschied einer Quelle aufgetragen und mit gestrichelter Linie der synchronisierte Teil der Szene eingetragen.
Im letzten Schritt können die ermittelten Werte binweise miteinander verglichen werden. Als Maß für die Genauigkeit der einzelnen Verfahren wurde die Varianz σ^2 ermittelt. Sie kann bestimmt werden, indem das Fehlersignal zwischen den geschätzten Laufzeitunterschieden und den Refernzwerten quadriert und über den gesamten Abschnitt gemittelt wird.

[2]Einmalige Synchronisation mit synchronisation.m
[3]Speicherung der Referenz-TDOAs in mat-Dateien

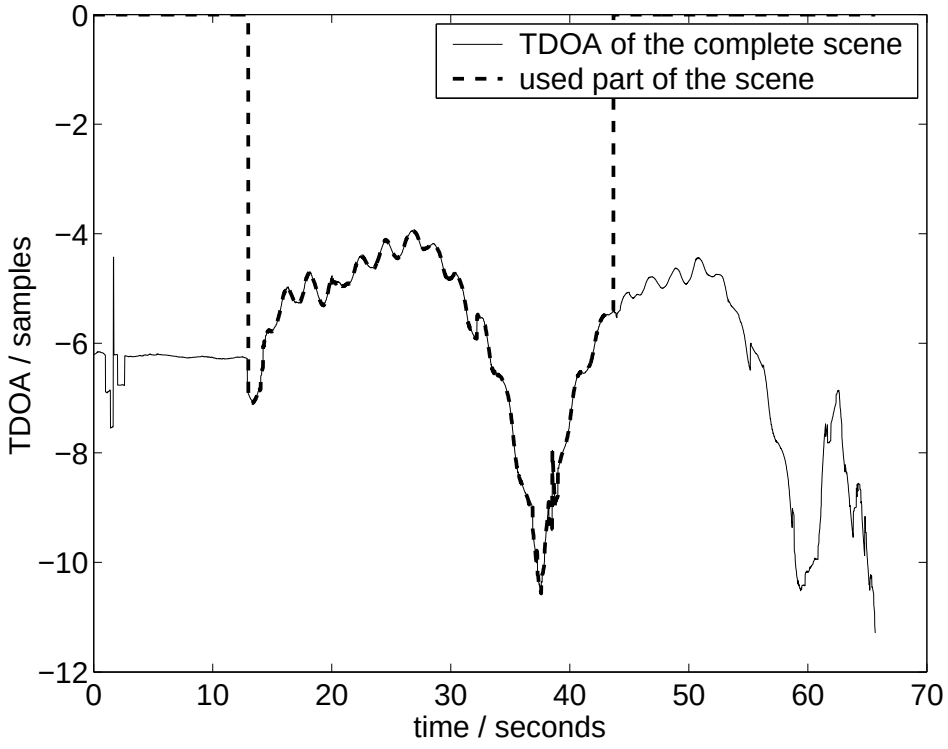

Abbildung 4.3: Synchronisation der Referenzdaten mit den berechneten TDOAs

4.2.2 BSS bei zwei Quellen

Zur besseren Übersichtlichkeit wurden nicht alle Verfahren in einer Abbildung dargestellt, sondern die BSS- und Referenz-TDOAs mit denen des AED- oder GCC-Algorithmus verglichen. In der oberen von zwei Darstellungen sind die absoluten Laufzeitunterschiede und in der unteren die absolute Abweichung vom Referenzwert der Infrarotdaten jeweils in Abtastwerten angegeben. Auf der x-Achse ist in beiden Darstellungen die Zeit in Sekunden angetragen, sodass die Veränderung der Laufzeitunterschiede über den Zeitraum von 29 Sekunden beobachtet werden kann.

In Abbildung 4.4 sind die Laufzeitunterschiede des festen Sprechers aus

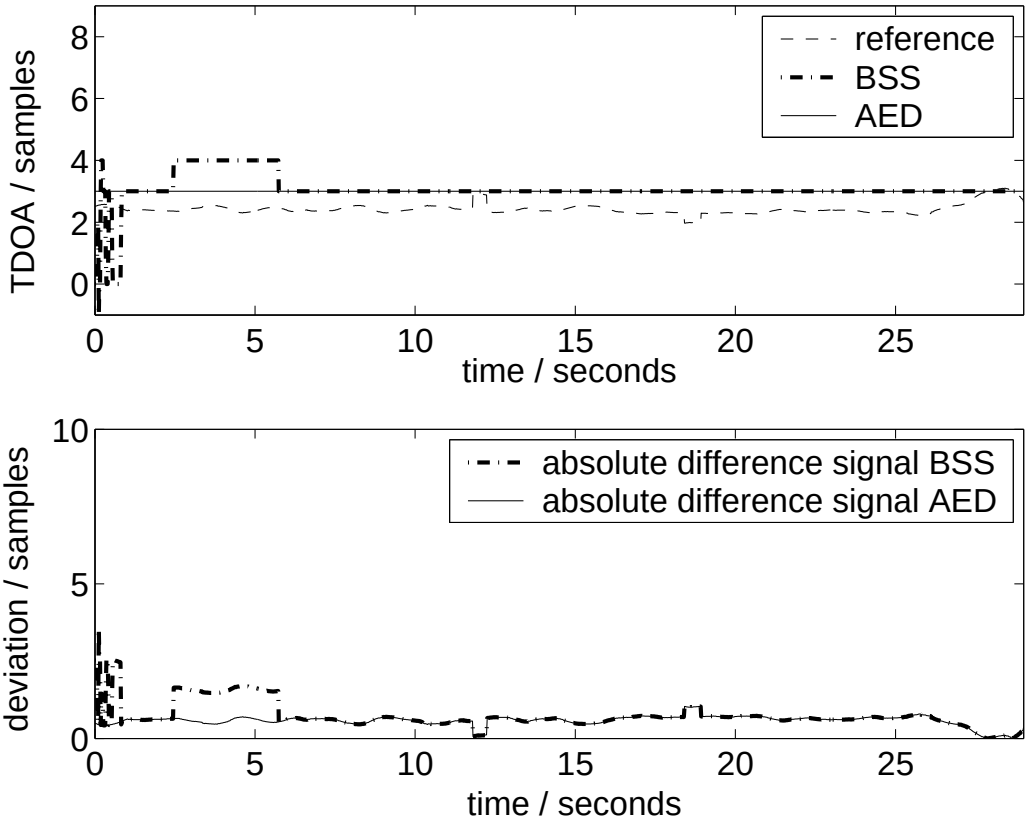

Abbildung 4.4: Vergleich von AED und BSS (2 Quellen) bei fester Quelle und r=16cm

Szene 4 aufgetragen. Daraus ist zu erkennen, dass der AED-Algorithmus (bei einer Quelle) geringfügig bessere Werte als der BSS-Algorithmus (bei zwei Quellen) liefert. Auch ist zu erkennen, dass der BSS-Algorithmus etwas Zeit für die Anfangskonvergenz benötigt, bis der erste gute Wert bestimmt werden kann. AED dagegen adaptiert schneller und liefert vom ersten Block an gute Schätzungen. Die Varianzen ergeben sich als $\sigma^2_{AED} = 0.36$ für den AED-Algorithmus und $\sigma^2_{BSS} = 0.71$ für den BSS-Algorithmus.

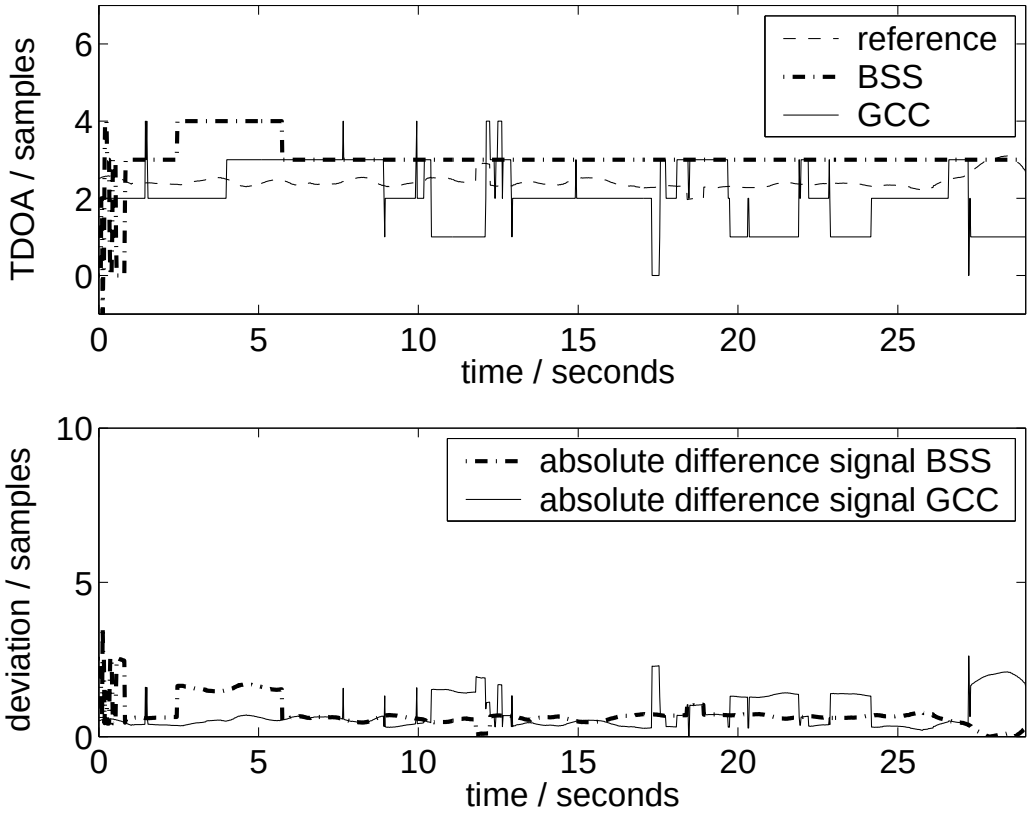

Abbildung 4.5: Vergleich von GCC und BSS (2 Quellen) bei fester Quelle und r=16cm

Betrachtet man nun bei der gleichen Szene die Ergebnisse von BSS- und GCC-Algorithmus (eine Quelle), so zeigt sich in Abbildung 4.5, dass die Schätzung mittels BSS etwas genauer ist, obwohl sie zwei Quellen gleichzeitig bewerkstelligt. Die Varianz der GCC-Methode liegt bei dieser Szene bei $\sigma^2_{GCC} = 0.86$ und ist daher höher als die der anderen Methoden.

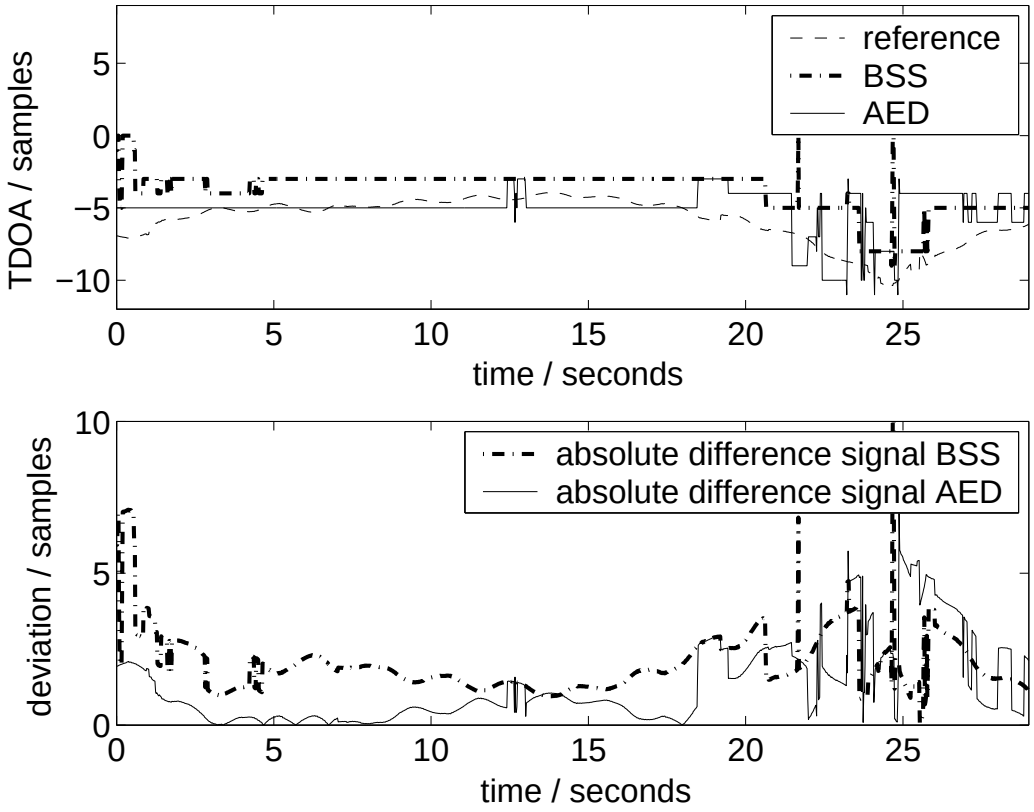

Abbildung 4.6: Vergleich von AED und BSS (2 Quellen) bei bewegter Quelle und r=16cm

Als nächstes wird die bewegte Quelle aus Szene 51 betrachtet. Abbildung 4.6 zeigt die Ergebnisse von AED- und BSS-Algorithmus im Vergleich zu den Referenz-TDOAs. Bei schneller Änderung des Wertes für den Laufzeitunterschied passt sich der BSS-Algorithmus besser als der AED-Algorithmus an. Jedoch weicht der Wert im vorderen Teil des betrachteten Ausschnitts weiter von der Referenz ab. Über die gesamte Zeit gemittelt ergeben sich die Varianzen $\sigma^2_{AED} = 3.46$ und $\sigma^2_{BSS} = 5.25$ und damit - wie bei der festen Quelle - ein etwas besserer Wert für den AED-Algorithmus.

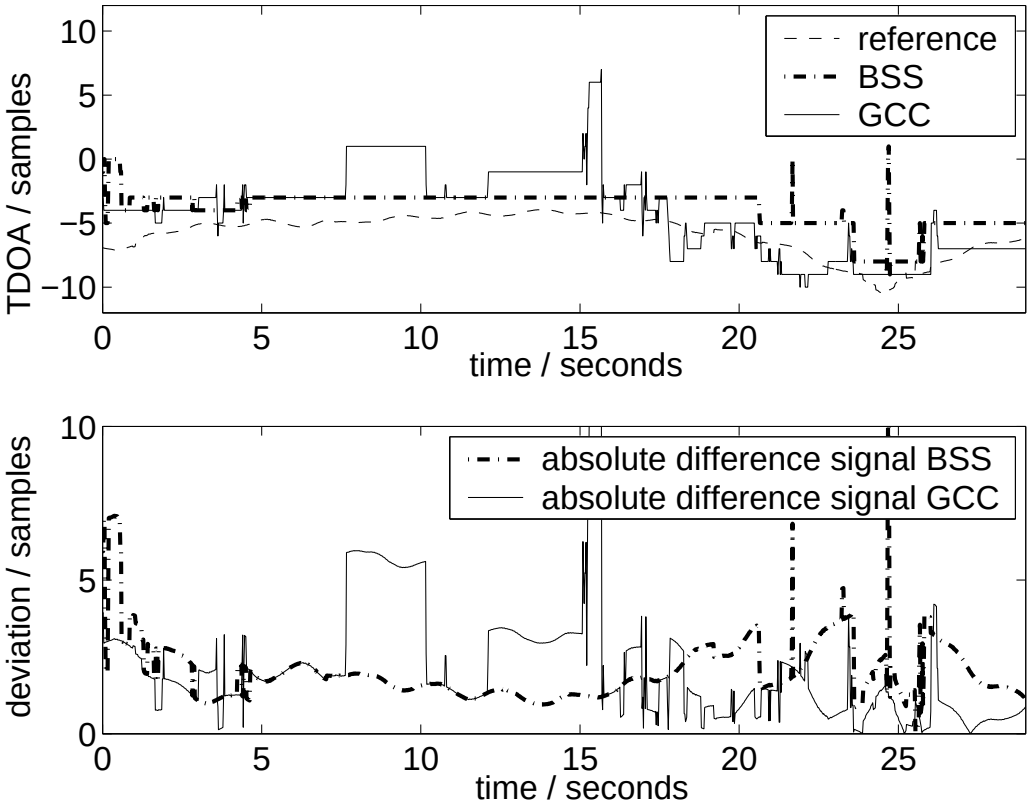

Abbildung 4.7: Vergleich von GCC und BSS (2 Quellen) bei bewegter Quelle und r=16cm

Beim Vergleich von BSS und GCC der Szene 51 (Abbildung 4.7) zeigen sich die Probleme des GCC-Algorithmus deutlicher als bei der festen Quelle. Die hohe Verhallung $T_{60} \approx 700$ ms und die Bewegung der Quelle lassen die ermittelten Werte weiter von den Referenzen abweichen als bei den anderen Algorithmen, was in einer höheren Varianz $\sigma^2_{GCC} = 7.64$ resultiert. Der BSS-Algorithmus ist meist näher am Referenzwert und liefert daher eine höhere Genauigkeit.

4.2.3 BSS bei einer Quelle

Im vorhergehenden Abschnitt wurde die BSS jeweils auf Mikrophonsignale mit Anteilen beider Quellen angesetzt, um deren Laufzeitunterschiede gleichzeitig zu bestimmen, was dem Ziel dieser Arbeit entspricht. Die Ergebnisse wurden dann mit denen der Algorithmen verglichen, die nur eine Quelle zu verarbeiten hatten. Um nun einen Vergleich mit denselben Anforderungen an die Algorithmen durchzuführen, wird die BSS in diesem Abschnitt (analog zu AED und GCC) auf die Mikrophonsignale von nur einer Quelle aufgesetzt.
Dazu ist zuerst zu klären, ob die BSS bei nur einer Quelle genutzt werden kann, da die BSS prinzipiell zwei Quellen voneinander trennt. Fehlt nun die zweite Quelle, so könnte man vermuten, dass die BSS und damit auch die TDOA-Bestimmung nicht mehr funktioniert. Diese Bedenken können jedoch durch die Betrachtung des folgenden Beispiels ausgeräumt werden.
Setzt man die BSS auf Mikrophonsignale an, die nur Anteile einer Quelle enthalten, so gibt es zwei Möglichkeiten, wie sich der Algorithmus verhalten könnte. Entweder er leitet die Mikrophonsignale zu den Ausgängen durch, ohne sie zu verarbeiten, oder er leitet alle Anteile der Quelle auf einen Ausgang und lässt den anderen Ausgang leer. Der erste Ansatz würde bedeuten, dass die Ausgangssignale Anteile der selben Quelle enthalten, die natürlich voneinander statistisch abhängig sind. Da die BSS aber die Ausgangssignale statistisch entkoppelt, kann nur die zweite Möglichkeit zutreffen. Deshalb kann auch bei nur einer Quelle der BSS-Algorithmnus zur Bestimmung des Laufzeitunterschieds genutzt werden.
In Abbildung 4.8 sind die Ergebnisse der festen Quelle aus Szene4 dargestellt. Es wurden AED und BSS auf die Mikrophonsignale dieser Quelle angesetzt. Deshalb ändert sich für die adaptive Eigenwertzerlegung gegenüber Abbildung 4.4 nichts und auch die Varianz $\sigma^2_{AED} = 0.36$ ist dieselbe. Die Ergebnisse

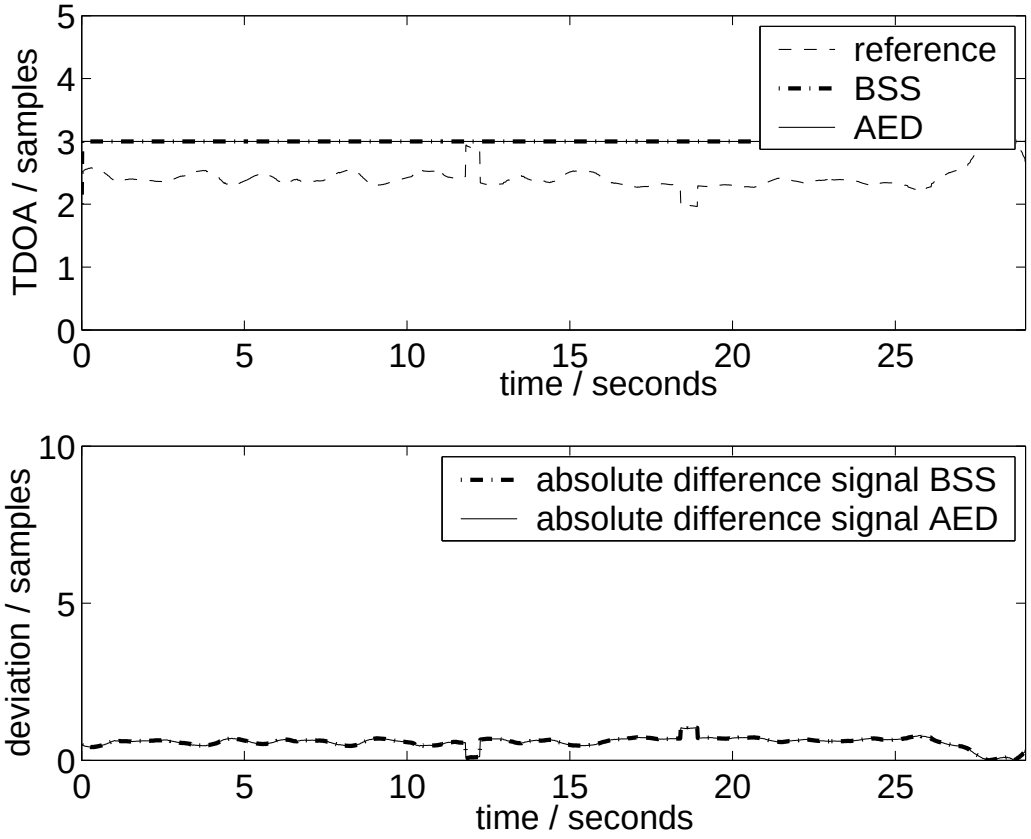

Abbildung 4.8: Vergleich von AED und BSS (1 Quelle) bei fester Quelle und r=16cm

der BSS dagegen verbessern sich, da die Anteile des Quellsignals auf einen Ausgang geleitet werden können, ohne durch das zweite Quellsignal gestört zu werden. Die Laufzeitunterschiede entsprechen stets den Werten des AED-Algorithmus. Die Varianz des BSS-Algorithmus ist daher $\sigma^2_{BSS} = 0.36$ und damit identisch mit der Varianz des AED-Algorithmus. Da die Ergebnisse der GCC-Methode dieselben wien in Abbildung 4.5 sind, wird der Vergleich zwischen BSS und GCC in diesem Abschnitt ausgelassen.
Abbildung 4.9 zeigt analog dazu die Ergebnisse der Szene51 für AED und BSS. Auch hier sind die Ergebnisse von AED und BSS etwa gleichwertig, was

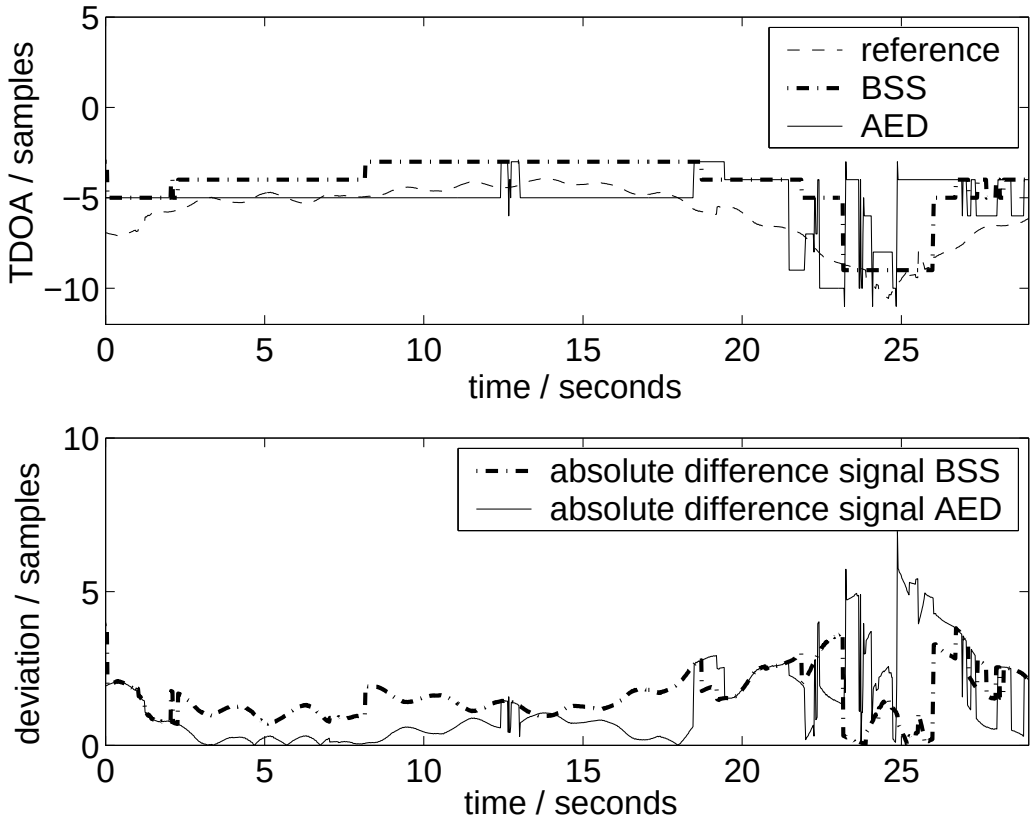

Abbildung 4.9: Vergleich von AED und BSS (1 Quelle) bei bewegter Quelle und r=16cm

auch die Varianzen $\sigma^2_{AED} = 3.49$ und $\sigma^2_{BSS} = 3.08$ bestätigen.
Zur besseren Übersicht sind die Varianzen der bisher gezeigten Ergebnisse in Tabelle 4.1 zusammengefasst.

Szene	AED	GCC	BSS bei zwei Quellen	BSS bei einer Quelle
scene4	0.36	0.85	0.71	0.36
scene51	3.49	7.64	5.25	3.08

Tabelle 4.1: Varianzen der Algorithmen bei den getesteten Szenen

4.2.4 Ergebnisse bei großem Mikrophonabstand

In diesem Abschnitt werden die Signale zweier Mikrophone verwendet, die einen größeren Abstand als die bisher betrachteten Mikrophone haben. Der Abstand wird von r=16cm auf r=50cm erhöht, um den Einfluss des Mikrophonabstandes r auf die Genauigkeit der Algorithmen zu untersuchen.

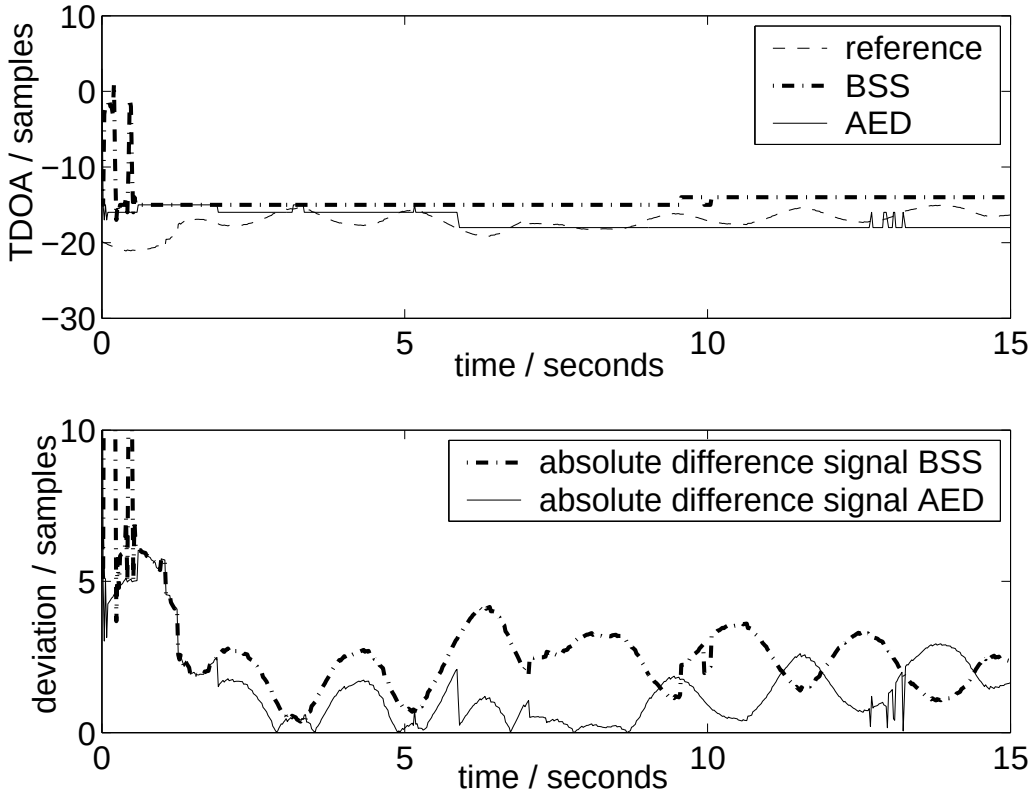

Abbildung 4.10: Vergleich von AED und BSS (2 Quellen) bei bewegter Quelle und r=50cm

In Abbildung 4.10 sind die Ergebnisse von AED- und BSS-Algorithmus dargestellt. Die TDOA-Werte der adaptiven Eigenwertzerlegung sind meist besser als die mit BSS berechneten Werte. Dies ist damit zu begründen, dass AED auf blinder Systemidentifikation, und der BSS-basierte Algorithmus auf

'blindem Beamforming' beruht. Dementsprechend ist im letzteren Fall auf räumliches Aliasing zu achten. Es ist zu erkennen, dass beide Algorithmen weiter von den Referenzwerten abweichen als bei kleinerem Mikrophonabstand.
Diese Ergebnisse bestätigen (vor allem bei der Nutzung des BSS-Algorithmus) die Berechtigung der alternativen Mikrophonanordnung entsprechend Abbildung 3.6, die auch im nächsten Abschnitt zur Positionsbestimmung genutzt wird.

4.3 Positionsbestimmung

4.3.1 Vorgehensweise

Das in 4.1.2 beschriebene Szenario wird im folgenden genutzt, um mit den besprochenen Algorithmen zuerst die Laufzeitunterschiede zu bestimmen und diese anschließend in Positionen umzusetzen[4]. Die Berechnung[5] erfolgte entsprechend 4.1.3 mit denselben Werten der Parameter wie im Szenario mit den Infrarotdaten.
Die Vorgehensweise wurde ebenso analog zum vorher betrachteten Szenario aufgebaut. Nach der Berechnung der TDOAs wurden alle Ergebnisse auf die selbe zeitliche Auflösung gebracht und anschließend miteinander verglichen. Lediglich der Schritt der Synchronisation mit den Infrarotdaten entfiel in diesem Szenario. Als Referenz konnte stattdessen der Laufzeitunterschied aus den Impulsantworten bestimmt werden, mit denen die Sprachsignale gefaltet wurden.

[4]Berechnung in MMR_pos.m
[5]Berechnung in MMR.m

4.3.2 Ergebnisse

In Abbildung 4.11 sind im oberen Teil die TDOAs aus AED- und BSS-Algorithmus, im unteren Teil GCC- und BSS-Werte eingezeichnet. Der Referenzwert für den Laufzeitunterschied ist im gesamten Ausschnitt $\hat{\tau} = -16$ Samples. Der AED-Algorithmus liefert vom ersten bis zum letzten Block

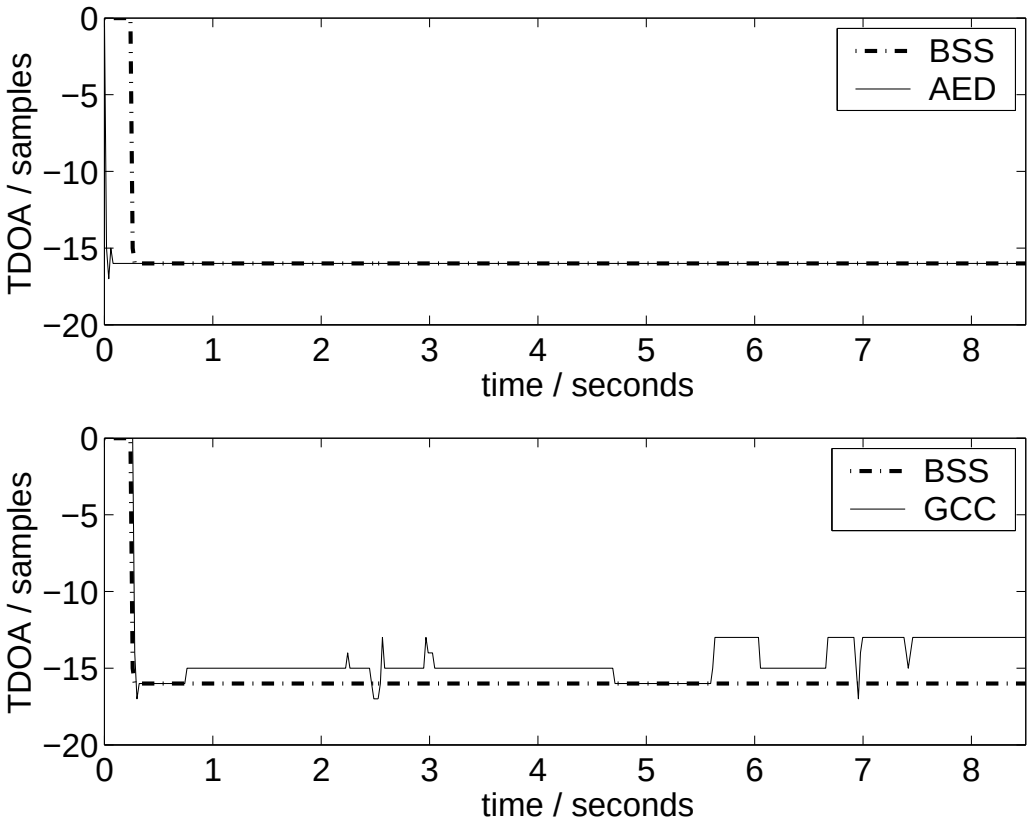

Abbildung 4.11: Vergleich von AED, BSS (2 Quellen) und GCC bei einer festen Quelle

genau diesen Wert. Der Grund dafür, dass dieses Ergebnis so gut ist, ist einerseits darin zu finden, dass die Nachhallzeit relativ gering (200ms) und außerdem der Laufzeitunterschied konstant ist.

Die BSS-Werte sind zu Beginn etwas weiter vom Referenzwert entfernt, was

darin begründet liegt, dass der Pegel der betrachteten Quelle zu diesem Zeitpunkt relativ niedrig ist und die BSS gerade in der Anfangskonvergenz empfindlich darauf reagiert. Daher kann der BSS-Algorithmus nicht so gut adaptieren, zumal er zusätzlich durch die zweite Quelle negativ beeinflusst wird. Nachdem die BSS den richtigen Wert erreicht hat, bleibt sie für die restliche Zeit genau auf dem Referenzwert und ist der AED ebenbürtig.
Die GCC liefert zu Beginn des Abschnitts den Wert $\tau = 0$, was ebenso mit der geringen Signalenergie begründet werden kann, die in diesem Bereich unterhalb der Schranke des Aktivitätsdetektors liegt. Daher werden die aktuellen Blöcke nicht zur Schätzung genutzt, sondern der Wert des letzten Blocks übernommen. Da der erste Wert auf Null gesetzt wurde sind die weiteren Werte ebenfalls Null. In dem Bereich, in dem die Signalenergie stärker ist, liefert auch der GCC-Algorithmus bessere Werte, die jedoch häufig um einen kleinen Betrag vom Referenzwert abweichen.

Die Quellposition kann berechnet werden, indem man die Gleichungen 3.1 und 3.2 so umformt, dass man je eine Gleichung für x_s und y_s erhält und die Mikrophonkoordinaten und Laufzeitunterschiede einsetzt. Abbildung 4.12 zeigt die errechneten Positionen für die Laufzeitunterschiede aus Abbildung 4.11. Die Werte des AED- (mit Plus gekennzeichnet) und BSS-Algorithmus (Kreis) sind alle auf den richtigen Punkt ($x_s \approx 1.4$ und $y_s \approx 1.7$) abgebildet. Dabei wurden die ersten BSS-Werte vernachlässigt, da die Laufzeitunterschiede der beiden Mikrophonpaare (Dreieck) keine gültige Position ergeben haben. Zwar bestimmt auch die GCC (Kreuz) für manche Blöcke die richtige Position, jedoch ergeben sich auch andere Positionen, die teilweise weit von der realen Position entfernt liegen. Was die Ergebnisse der GCC jedoch relativ gut einschränken ist der Winkelbereich vom Ursprung aus, da die Mikrophone symmetrisch um den Ursprung angordnet wurden.

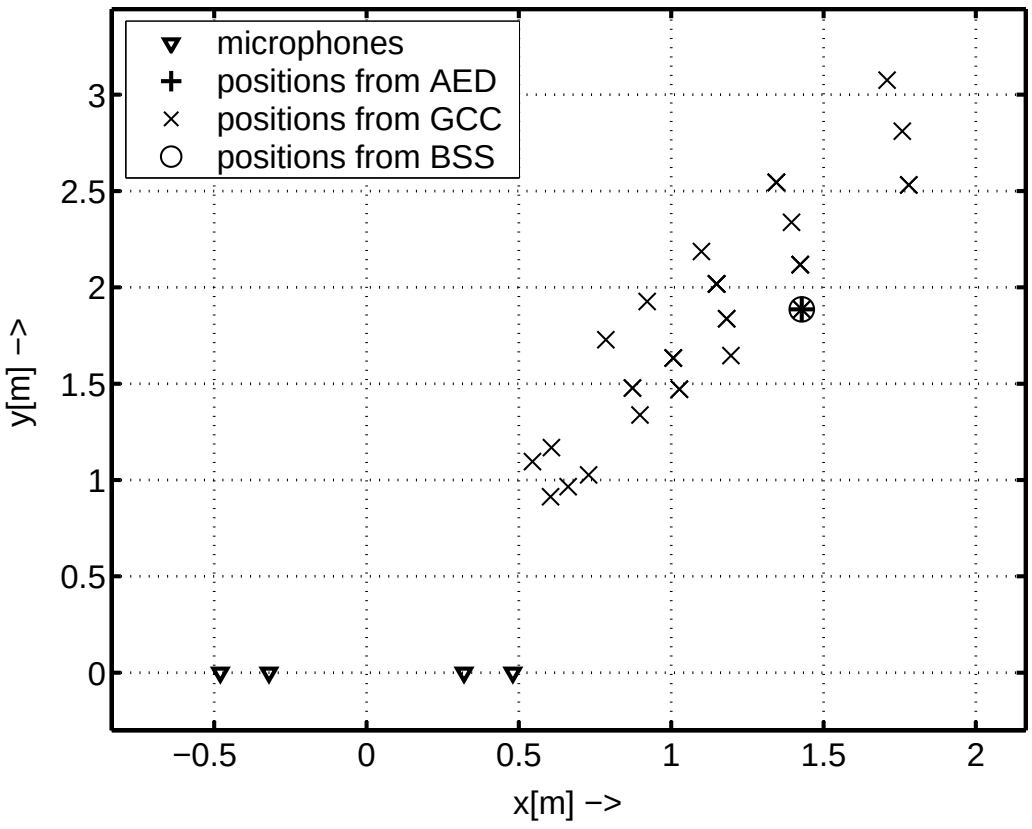

Abbildung 4.12: Von AED, BSS (2 Quellen) und GCC berechnete Positionen

Kapitel 5

Schlussfolgerungen

In dieser Arbeit wurde eine neue Methode zur Quellenlokalisierung mittels blinder Quellentrennung vorgestellt und mit bereits existierenden Methoden verglichen. Bei der Anwendung der implementierten Algorithmen auf die Daten zweier Szenen aus [6] hat sich gezeigt, dass der neue Algorithmus dem AED-Algorithmus in etwa gleichwertig ist, wenn er zur Lokalisierung einer Quelle genutzt wird. Er kann aber auch zur simultanen Lokalisierung zweier Quellen genutzt werden, wobei die Genauigkeit gegenüber dem Fall mit nur einer Quelle etwas abnimmt. Im Vergleich zu den anderen Algorithmen ist diese Methode aber sehr vorteilhaft, da AED- und GCC-Algorithmus prinzipiell nur eine Quelle handhaben können. Durch geeignete Erweiterung des BSS-Algorithmus, könnten sogar die dreidimensionalen Positionen von mehr als zwei Quellen simultan bestimmt werden.
Abschließend ist anzumerken, dass in der gesamten Arbeit keine Nachverarbeitung genutzt wurde. Durch zusätzliches Nachbearbeiten (z.B. Glättung der TDOA-Kurven mit Median-Filter oder Kalman-Filter), kann die Genauigkeit der Positionsbestimmung vermutlich noch gesteigert werden.

Literaturverzeichnis

[1] C.H.Knapp G.C.Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-24, pp.320-327, Aug. 1976

[2] J.Benesty, "Adaptive Eigenvalue Decomposition Algorithm for Passive Acoustic Source Localization," *J.Acoust.Soc.Am.*, vol.107, pp.384-391, Jan 2000.

[3] A.Schneider, *A real-time demonstrator for robust speaker localisation*, Diplomarbeit, Erlangen 2001.

[4] H.Buchner, R.Aichner, W.Kellermann, "A Generalisation of Blind Source Separation Algorithms for Convolutive Mistures Based on Second-Order Statistics," *IEEE Transactions on Speech and Audio Processing,* im Druck.

[5] H.Buchner, R.Aichner, W.Kellermann, "TRINICON: A Versatile Framework for Multichannel Blind Signal Processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* Montreal, May 2004.

[6] M.Krinidis, G.Stamou, H.Teutsch, S.Spors, N.Nikolaidis, R.Rabenstein, I.Pitas, "An audio-visual database for evaluating person tracking algorithms," eingereicht zur Veröffentlichung.

[7] H.Sawada, R.Mukai, S.Makino, "Direction of Arrival Estimation for multiple source signals using Independant Component Analysis," ISSPA 2003.

[8] R.Aichner, H.Buchner, F.Yan, W.Kellermann, "Real-Time Convolutive Blind Source Separation based on a Braodband Approach," in *Proc. Int. Symp. on Independent Component Analysis and Blind Source Separation* (ICA), Granada, Spain, Sept. 2004.

[9] F.Yan, *Real-time Blind Source Separation for Convolutive Mixtures*, Diplomarbeit, Erlangen 2004.

[10] J.D.Markel, A.H.Gray *Linear Predicition of Speech,* Springer, Berlin, 1976.

[11] J.J. Shynk, "Frequency-Domain and Multirate Adaptive Filtering," *IEEE SP Magazine,* January 1992.

[12] B.D.Van Veen, "Beamforming: A versatile approach to Spacial Filtering," *IEEE SP Magazine,* April 1988.

[13] H.Liu, G.Xu, L.Tong, "A deterministic approach to Blind Identification of Multi-Channel FIR-Systems," ICASSP 94, Adelaide, Australien, April 1994.