

# Masterarbeit

Jens Reermann

Echtzeit-Lokalisierung von Audioquellen mittels  
zirkularen Mikrofonarrays und des  
UCA-ESPRIT-Algorithmus

**Jens Reermann**  
**Echtzeit-Lokalisierung von Audioquellen mittels  
zirkularen Mikrofonarrays und des  
UCA-ESPRIT-Algorithmus**

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im gemeinsamen Masterstudiengang Mikroelektronische Systeme  
am Fachbereich Technik  
der Fachhochschule Westküste  
und  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Hans Peter Kölzer  
Zweitgutachter : Prof. Dr.-Ing. Hans-Dieter Schütte

Abgegeben am 11. Oktober 2013

**Jens Reermann**

**Thema der Masterarbeit**

Echtzeit-Lokalisierung von Audioquellen mittels zirkularen Mikrofonarrays und des UCA-ESPRIT-Algorithmus

**Stichworte**

Echtzeit-Lokalisierung, UCA-ESPRIT-Algorithmus, zirkulares Mikrofonarray, variabler Phasenschieber, Quellensimulator

**Kurzzusammenfassung**

In dieser Arbeit wird die Echtzeit-Lokalisierung von Audioquellen auf Basis eines zirkularen Mikrofonarrays und dem UCA-ESPRIT-Algorithmus diskutiert. Dazu erfolgt zunächst die Betrachtung der Grundlagen, bevor die Konzeptionierung mit Hilfe von Simulationen durchgeführt wird. Das entwickelte Konzept wird anschließend simuliert und auf dem DSP TMS320C6713B umgesetzt. Die Ergebnisse werden zu einem Endgerät übertragen, visualisiert oder protokolliert, bewertet und zur Fehlerkorrektur genutzt. Weiter wird ein variabler Phasenschieber zur Quellensimulation diskutiert und implementiert.

**Jens Reermann**

**Title of the paper**

Real-time localization of audio sources based on a circular microphone array and the UCA-ESPRIT algorithm

**Keywords**

real-time localization, UCA-ESPRIT algorithm, circular microphone array, fractional delay filter, source simulator

**Abstract**

In this thesis the real-time localization of audio sources is discussed based on circular microphone arrays and the UCA-ESPRIT algorithm. Therefore the principles are considered, before a concept is developed by simulations. The developed concept is simulated and implemented on the DSP TMS320C6713B afterwards. The results are transmitted to a terminal, visualized or recorded, rated and used for error correction. Furthermore a fractional delay filter is discussed and implemented for the simulation of sources.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>8</b>
<b>Bildverzeichnis</b>	<b>9</b>
<b>Abkürzungsverzeichnis</b>	<b>12</b>
<b>Symbolverzeichnis</b>	<b>13</b>
<b>1 Einleitung</b>	<b>18</b>
1.1 Motivation . . . . .	18
1.2 Stand des Forschungsprojektes . . . . .	19
1.3 Problemstellung . . . . .	20
1.4 Struktur der Arbeit . . . . .	21
1.5 Notation . . . . .	22
<b>2 Hardwarenahe Implementierung</b>	<b>23</b>
2.1 Mikrofonarray und -verstärker . . . . .	23
2.2 DSP-System . . . . .	25
2.2.1 D.Module.C6713 . . . . .	26
2.2.2 D.Module.PCM3003 . . . . .	27
2.2.3 Kommunikation . . . . .	29
2.3 Blockweise Programmierung . . . . .	29
2.4 Compiler Optimierungsstufen . . . . .	31
2.5 TI C67x DSPLib . . . . .	34
2.6 Komplexwertige Zahlen . . . . .	35
2.7 Vorgehensweise . . . . .	35
<b>3 Theoretische Grundlagen</b>	<b>36</b>
3.1 Signalmodell . . . . .	36
3.1.1 Ansatz . . . . .	36
3.1.2 Schmalbandigkeit . . . . .	38
3.1.3 Matrixnotation . . . . .	40
3.2 Räumlicher Alias-Effekt . . . . .	42
3.3 Autokorrelationsmatrix . . . . .	42

3.4	Unterraumbasierte Verfahren . . . . .	44
3.4.1	MUSIC . . . . .	44
3.4.2	ESPRIT . . . . .	47
<b>4</b>	<b>Variabler Phasenschieber</b>	<b>50</b>
4.1	Grundlegende Überlegungen . . . . .	50
4.2	Methode nach Pei und Lai . . . . .	53
4.3	Methode mittels Spiegelung . . . . .	53
4.4	Blockweise Verarbeitung . . . . .	55
4.5	Implementierung . . . . .	60
4.6	Messungen . . . . .	62
4.7	Quellensimulator . . . . .	65
<b>5</b>	<b>UCA-ESPRIT-Algorithmus</b>	<b>69</b>
5.1	Zirkulares Array . . . . .	69
5.2	Phasen-Moden Anregung . . . . .	71
5.2.1	Kontinuierliche zirkulare Mikrophonarrays . . . . .	72
5.2.2	Diskrete zirkulare Mikrophonarrays . . . . .	74
5.2.3	Beamformer . . . . .	78
5.3	UCA-ESPRIT-Algorithmus . . . . .	82
5.4	Simulationen . . . . .	88
<b>6</b>	<b>Konzepterarbeitung</b>	<b>93</b>
6.1	Sprachsignale . . . . .	93
6.2	Anwendung auf breitbandige Signale . . . . .	95
6.2.1	Übergang in den Spektralbereich . . . . .	96
6.2.2	Kohärenter und inkohärenter Ansatz . . . . .	97
6.2.3	Konzeptansatz . . . . .	98
6.3	Vorverarbeitung . . . . .	98
6.3.1	Energiebetrachtung . . . . .	98
6.3.2	Fensterung . . . . .	99
6.3.3	Überhang . . . . .	101
6.3.4	Frequenzauswahl . . . . .	101
6.3.5	Normierung . . . . .	102
6.4	Diskussion des UCA-ESPRIT-Algorithmus . . . . .	104
6.4.1	Eigenwertanalyse . . . . .	104
6.4.2	Anzahl der betrachteten Phasen-Moden . . . . .	112
6.4.3	Frequenzabhängigkeit . . . . .	115
6.4.4	Struktur der Ergebniswerte . . . . .	117
6.5	Ergebnisaufbereitung . . . . .	119
6.5.1	Mittelwert- und Medianfilter . . . . .	119
6.5.2	Zweistufige Histogrammauswertung . . . . .	119

6.5.3	Ergebnismittelung pro Datensatz . . . . .	121
6.5.4	Merkmalsraum . . . . .	121
6.5.5	Fehlerkorrektur . . . . .	122
6.6	Ergebnisausgabe . . . . .	122
6.7	Zusammenfassung - Das Konzept . . . . .	123
<b>7</b>	<b>Konzeptumsetzung</b>	<b>125</b>
7.1	Parameter . . . . .	125
7.2	Simulationen . . . . .	127
7.2.1	Simulationen mit einer aktiven Quelle . . . . .	128
7.2.2	Simulationen mit zwei aktiven Quellen . . . . .	129
7.2.3	Variation der Parameter . . . . .	133
7.3	Implementierung . . . . .	136
7.3.1	Übersicht über die Programmstruktur . . . . .	136
7.3.2	Funktionen . . . . .	138
7.3.3	Nachweis der Echtzeitfähigkeit . . . . .	143
<b>8</b>	<b>Visualisierung</b>	<b>145</b>
8.1	Wahl der Programmiersprache . . . . .	145
8.2	Unterstützende Bibliotheken . . . . .	145
8.2.1	RXTX . . . . .	145
8.2.2	Java3D . . . . .	146
8.3	Programmübersicht . . . . .	146
8.4	Programmimplementierung . . . . .	148
<b>9</b>	<b>Tests, Fehlerkorrektur und Evaluation</b>	<b>152</b>
9.1	Bewertungskriterien . . . . .	152
9.2	Testsystem . . . . .	152
9.3	Tests im reflexionsarmen Raum . . . . .	153
9.3.1	Aufbau und Systemparametrisierung . . . . .	153
9.3.2	Testreihe und Fehlerkorrektur mit einer aktiven Quelle . . . . .	155
9.3.3	Testreihen mit zwei aktiven Quellen . . . . .	160
9.3.4	Untersuchung der Reaktionszeit . . . . .	162
9.4	Tests in realer Umgebung . . . . .	165
9.4.1	Aufbau und Systemparametrisierung . . . . .	165
9.4.2	Testreihe mit einer aktiven Quelle . . . . .	167
9.4.3	Testreihe mit drei Sprechern . . . . .	169
9.5	Bewertung . . . . .	172
<b>10</b>	<b>Zusammenfassung und Ausblick</b>	<b>173</b>
	<b>Literaturverzeichnis</b>	<b>176</b>

<b>A</b>	<b>Mathematische Zusammenhänge</b>	<b>179</b>
A.1	Mittelwert und Varianz von Winkeln . . . . .	179
A.2	Eigen- und Singulärwertzerlegung . . . . .	180
<b>B</b>	<b>Inhalt des Datenträgers</b>	<b>183</b>

# Tabellenverzeichnis

1.1	Notation . . . . .	22
1.2	Matrixeigenschaften . . . . .	22
2.1	Optimierungsstufen . . . . .	33
7.1	Einstellung der Systemparameter (Simulationen) . . . . .	127
9.1	Einstellung der Systemparameter (reflexionsarmen Raum) . . . . .	155
9.2	Umgebungsparameter der realen Umgebung . . . . .	166
9.3	Sollwerte der Quellen (ein Sprecher) . . . . .	167
9.4	Sollwerte der Quellen (drei Sprecher) . . . . .	169

# Bildverzeichnis

1.1	Anwendungsszenarien . . . . .	18
2.1	Mikrofonarray . . . . .	23
2.2	Schaltplan Mikrofonverstärker . . . . .	24
2.3	Mikrofonverstärkerplatine . . . . .	24
2.4	DSP-System . . . . .	25
2.5	Blockschaltbild D.Module.C6713 . . . . .	26
2.6	Blockschaltbild TMS320C6713B . . . . .	27
2.7	Blockschaltbild D.Module.PCM3003 . . . . .	28
2.8	Datenformat D.Module.PCM3003 . . . . .	29
2.9	Flussdiagramm EDMA-Grundgerüst . . . . .	30
2.10	Ping-Pong-Technik . . . . .	31
2.11	Ablauf des Programmierungsprozesses . . . . .	32
3.1	Uniform Linear Array . . . . .	37
3.2	Amplitudengänge bandbegrenzter Signale . . . . .	39
3.3	Prinzip des <i>ESPRIT</i> -Algorithmus . . . . .	47
4.1	Impulsantwort . . . . .	52
4.2	Testsequenz . . . . .	55
4.3	Fehler und Zeitverläufe bei einer Verzögerung von $1,5T$ . . . . .	56
4.4	Blockweise Verarbeitung des variablen Phasenschiebers . . . . .	57
4.5	Fehler und Zeitverläufe bei einer Verzögerung von $1,5T$ (mit Erweiterungen) . . . . .	58
4.6	Fehler in Abhängigkeit der Verzögerung . . . . .	59
4.7	Grundprogramm des variablen Phasenschiebers . . . . .	60
4.8	Ablaufdiagramm des variablen Phasenschiebers . . . . .	61
4.10	Amplitudengang und Gruppenlaufzeit . . . . .	64
4.11	Ablaufdiagramm des Quellensimulators . . . . .	66
4.12	Vergleich der Fehler der Methoden mit realisierbaren Parametereinstellungen . . . . .	67
4.13	Zeitverläufe und Rechenzeit des Quellensimulators bei der Methode mittels Spiegelung . . . . .	67
5.1	Gleichverteiltes, zirkulares Mikrofonarray . . . . .	70
5.2	Besselfunktionen . . . . .	73
5.3	Vergleich von Nutz- und erstem Restterm über $m$ und $\zeta$ . . . . .	75

5.4	Realteil des Verhältnisses des Restterms und der Richtcharakteristik bei einem kontinuierlichen Array je Phasen-Modus . . . . .	77
5.5	UCA-ESPRIT-Algorithmus . . . . .	87
5.6	Fehler- und Varianzverläufe über $\bar{\varphi}$ und $\bar{\theta}$ (komplexwertige, schmalbandige Eingangssignale, $M = 3$ ) . . . . .	89
5.7	Histogramm der Ergebnisse bei $N_S = 1$ . . . . .	90
5.8	Histogramm der Ergebnisse bei $N_S = 2$ . . . . .	91
5.9	Verlauf der Eigenwerte bei komplexwertigen, schmalbandigen Eingangssignalen . . . . .	92
6.1	Zeitverläufe der Testsequenzen . . . . .	94
6.2	Teilspektrogramme der Testsequenzen ( $N_{FFT} = 512$ , 50% Überhang, Hamming Fenster) . . . . .	95
6.3	Konzeptansatz . . . . .	98
6.4	Energie der empfangenen Testsequenzen ( $N = 512$ ) . . . . .	99
6.5	Normierter Prinzipamplitudengang Fensterfunktion . . . . .	100
6.6	Vergleich verschiedener Fenster bei $N_{DFT}=512$ . . . . .	100
6.7	Fenster zur Frequenzauswahl . . . . .	102
6.8	Amplituden- und Phasengang der Strecke Soundkarte, Lautsprecher, Mikrofonarray, Vorverstärker und Analog-Digital-Umsetzer . . . . .	103
6.9	Verlauf der Eigenwerte bei Sprachsignalen . . . . .	105
6.10	Anteil des männlichen Sprechers an Peaks im Amplitudengang . . . . .	106
6.11	Numerischer Aufwand des UCA-ESPRIT-Algorithmus . . . . .	107
6.12	Zeitverlauf der Lokalisierung (generierte Signale, $N_S = 2$ ) . . . . .	108
6.13	Histogramm unter Verwendung generierter Signale ( $N_S = 2$ , Auflösung $1^\circ$ ) . . . . .	109
6.14	Histogramm unter Verwendung von Aufnahmen ( $N_S = 2$ , Auflösung $1^\circ$ ) . . . . .	110
6.15	Verhältnis des zweiten zum ersten Eigenwert ( $\hat{N}_S = 1$ ) . . . . .	111
6.16	Ergebnishistogramme für $M = 2$ und $M = 3$ (generierte Signale, Auflösung $1^\circ$ ) . . . . .	113
6.17	Ergebnishistogramme für $M = 2$ und $M = 3$ (Aufnahmen, Auflösung $1^\circ$ ) . . . . .	114
6.18	Amplituden-Frequenzabhängigkeit . . . . .	116
6.19	Amplituden-Frequenzabhängigkeit mit angepasstem Frequenzauswahlfenster . . . . .	116
6.20	Histogramm der Amplituden (Aufnahmen, $N_S = 2$ ) . . . . .	117
6.21	Eigenwertkennlinie des Elevationswinkels . . . . .	118
6.22	Ergebnishistogramm der Eigenwerte . . . . .	118
6.23	Histogrammaufbau . . . . .	120
6.24	Histogrammanalyse . . . . .	121
6.25	Protokoll . . . . .	122
6.26	Konzept . . . . .	124
7.1	Fehler- und Varianzverläufe über $\varphi$ und $\theta$ . . . . .	128
7.2	Simulationsergebnisse bei zwei aktiven Quellen mit generierten Signalen . . . . .	130
7.3	Ergebnishistogramme (Auflösung $1^\circ$ ) . . . . .	131

7.4	Simulationsergebnisse bei zwei aktiven Quellen mit Aufnahmen . . . . .	132
7.5	Ergebnisse bei zwei aktiven Quellen (statisch und dynamisch) . . . . .	133
7.6	Ergebnisse bei zwei aktiven Quellen (Aufnahmen) . . . . .	134
7.7	Ablaufdiagramm der Lokalisierung . . . . .	137
7.8	Ablaufdiagramm des UCA-ESPRIT-Algorithmus . . . . .	139
7.9	Ablaufdiagramm der zweistufigen Histogrammanalyse . . . . .	142
7.10	Nachweis der Echtzeitfähigkeit . . . . .	143
8.1	Programmansicht . . . . .	147
8.2	UML-Diagramm . . . . .	149
8.3	Automat des RS232-Empfängers . . . . .	150
9.1	Testsystem . . . . .	153
9.2	Versuchsaufbau im reflexionsarmen Raum . . . . .	154
9.3	Fehler- und Varianzverläufe ohne Korrektur . . . . .	156
9.4	Vergleich von Fehlerverläufen und modellierten Fehlerverläufen . . . . .	157
9.5	Fehler- und Varianzverläufe mit Korrektur . . . . .	158
9.6	Fehlerverläufe bei zwei aktiven Quellen bei $\theta_{Soll} = 67^\circ$ über $\bar{\varphi}$ . . . . .	160
9.7	Separierbarkeit zweier Quellen bei $\varphi_{Soll1} = 0^\circ$ und $\theta_{Soll1} = 60^\circ$ . . . . .	161
9.8	Definition der Reaktionszeiten . . . . .	162
9.9	Synchronisation . . . . .	163
9.10	Messergebnisse der Reaktionszeiten . . . . .	164
9.11	Versuchsaufbau in einer realen Umgebung . . . . .	166
9.12	Fehlerverläufe bei zwei aktiven Quellen bei $\theta_{Soll} = 67^\circ$ . . . . .	168
9.13	Ergebnisverlauf bei chronologisch aktiven Quellen . . . . .	170
9.14	Ergebnisverlauf bei parallel aktiven Quellen . . . . .	171
10.1	Gesamtsystem . . . . .	173
A.1	Zirkulare Varianz . . . . .	180

# Abkürzungsverzeichnis

CPU	.....	Central Processor Unit
DFT	.....	Discrete Fourier Transformation
DSP	.....	Digital Signal Processor
EDMA	.....	Enhanced Direct Memory Access
ESPRIT	.....	Estimation of Signal Parameters via Rotational Invariance Techniques
FFT	.....	Fast Fourier Transformation
LD	.....	Leistungsdichte
McBSP	.....	Multichannel Buffered Serial Port
MCCC	.....	Multichannel Cross-Correlation Coefficient
MERIT	.....	Mensch-Roboter-Interaktionssystem
MUSIC	.....	Multiple Signal Classification
RS232	.....	Übertragungsstandard einer seriellen Schnittstelle
SNR	.....	Signal to Noise Ratio
TDFT	.....	Time Discrete Fourier Transformation
TLS	.....	Total-Least-Squares
TLS-ESPRIT	.	Realisierung des ESPRIT-Algorithmus mittels dem TLS-Kriterium
UART	.....	Universal Asynchronous Receiver Transmitter
UCA	.....	Uniform Circular Array
UCA-ESPRIT	.	Algorithmus basierend auf einem UCA und dem ESPRIT-Algorithmus
ULA	.....	Uniform Linear Array
UML	.....	Unified Modeling Language

# Symbolverzeichnis

$\mathbf{0}_{P \times N}$ .....	Nullmatrix mit $P$ Zeilen und $N$ Spalten
$\alpha$ .....	Hilfsvariable
$\alpha_D$ .....	Dämpfungsfaktor
$\alpha_{min}$ .....	minimale Sperrdämpfung des Amplitudenganges von Zeitsignalfenstern
$\mathbf{A}_p$ .....	Richtungsmatrix (Index $p$ entsprechend Array oder Beamformer)
$\mathbf{a}_p$ .....	Richtungsvektor (Index $p$ entsprechend Array oder Beamformer)
$\mathbf{A}_{p,i}$ .....	Unterrichtungsmatrix (Index $p$ entsprechend Array oder Beamformer)
$\mathbf{a}_{p,i}$ .....	Unterrichtungsvektor (Index $p$ entsprechend Array oder Beamformer)
$\tilde{\mathbf{A}}_{ULA}$ .....	zusammengefasste Richtungsmatrix beim <i>ESPRIT</i> -Algorithmus
$\mathbf{B}$ .....	Hilfsmatrix
$B$ .....	Bandbreite
$b$ .....	Hilfsvariable
$b^c$ .....	Fernfeld Richtcharakteristik eines kontinuierlichen zirkularen Arrays
$b_m^c$ .....	Fernfeld Richtcharakteristik eines kontinuierlichen zirkularen Arrays des $m$ -ten Phasen-Modus
$\beta$ .....	Hilfsvariable
$\mathbf{B}_p$ .....	Beamformer (Index $p$ entsprechend Typs gerade, reell, <i>UCA-ESPRIT</i> )
$b_m^s$ .....	Fernfeld Richtcharakteristik eines kontinuierlichen zirkularen Arrays des $m$ -ten Phasen-Modus
$\mathbf{C}$ .....	Hilfsmatrix
$\mathbf{Q}$ .....	Hilfsmatrix
$c$ .....	Hilfsvariable
$\mathbf{C}_g$ .....	Hilfsmatrix
$\mathbf{C}_o$ .....	Hilfsmatrix
$\mathbf{C}_U$ .....	Hilfsmatrix
$\mathbf{D}$ .....	Hilfsmatrix, $\mathbf{D} = \text{diag} \{ \lambda_1, \dots, \lambda_{2N_S} \}$
$d_{array}$ .....	Ausmaße des Mikrofonarrays
$\Delta_p$ .....	Selektionsmatrix (Index $p$ entsprechend Selektion)
$\Delta d$ .....	geometrischer Versatz beim <i>ESPRIT</i> -Algorithmus

---

$\Delta\gamma$ .....	Winkelabstand zweier Mikrofone beim UCA
$\det \{\cdot\}$ .....	Determinante einer Matrix
$d_{mic}$ .....	Mikrofonabstand
$d_n$ .....	Verzögerung in Abtastpunkten
$d_{nN}$ .....	Nenner der rationalen Verzögerung $d_n = d_{nz}/d_{nN}$
$d_{nR}$ .....	restliche Verzögerungen, $d_{nR} = d_n - d_{nT}$
$d_{nT}$ .....	Anzahl ganzzahliger Verzögerungen von $T$
$d_{nZ}$ .....	Zähler der rationalen Verzögerung $d_n = d_{nz}/d_{nN}$
$d_{s-a}$ .....	Abstand der Mittelpunkte der Quellen und des Mikrofonarrays
$\mathbf{E}$ .....	Eigenvektormatrix
$\mathbf{e}$ .....	Eigenvektor
$\mathbf{E}_{G1G2}$ .....	zusammengesetzte Signalraummatrix
$\mathbf{E}_n$ .....	Eigenvektormatrix des Rauschunterraums
$E_N$ .....	Energie von $N$ Werten
$\varepsilon$ .....	Einfallswinkel der Wellenfront bei einem ULA
$E \{\cdot\}$ .....	Erwartungswertoperator
$\mathbf{E}_x$ .....	Eigenvektormatrix des Signalunterraums
$\mathbf{E}_{x,p}$ .....	Eigenvektormatrix des Signalunterraums (Index $p$ entsprechend Beamformer)
$\mathbf{E}_{x,p,i}$ .....	Untereigenvektormatrix des Signalunterraums (Index $p$ entsprechend Beamformer), $\mathbf{E}_{x,p,i} = \Delta_i \cdot \mathbf{E}_{x,p}$
$\mathcal{F} \{\cdot\}$ .....	Fouriertransformierte
$\mathbf{F}$ .....	Hilfsmatrix
$f$ .....	Frequenz
$\mathcal{F}^{-1} \{\cdot\}$ .....	inverse Fouriertransformierte
$f_c$ .....	Mittelfrequenz
$f_{lb\_h}$ .....	untere Grenze des zweiten Übergangs des Frequenzauswahlfensters
$f_{lb\_l}$ .....	untere Grenze des ersten Übergangs des Frequenzauswahlfensters
$f_{max}$ .....	Grenzfrequenz
$f_{SW}$ .....	halbe Frequenzselektionsbreite
$f_{ub\_h}$ .....	obere Grenze des zweiten Übergangs des Frequenzauswahlfensters
$f_{ub\_l}$ .....	obere Grenze des ersten Übergangs des Frequenzauswahlfensters
$\mathbf{F}_{yp}$ .....	Hilfsmatrix (Index $p$ entsprechend Arraygruppe)
$\Gamma$ .....	Hilfsmatrix
$\gamma_i$ .....	Positionswinkel des $i$ -ten Mikrofons im UCA
$\Gamma_F$ .....	Hilfsmatrix

---

$\Gamma_{F, TLS}$ .....	Hilfsmatrix
$H_p$ .....	Frequenzgang (Index $p$ entsprechend Algorithmus)
$h_p$ .....	Impulsantwort (Index $p$ entsprechend Algorithmus)
$\mathbf{I}_N$ .....	Einheitsmatrix mit $N$ Spalten und Zeilen
$\tilde{\mathbf{I}}_N$ .....	gespiegelte Einheitsmatrix mit $N$ Spalten und Zeilen
$\Im \{ \cdot \}$ .....	Imaginärteil
$j$ .....	imaginäre Einheit
$J_i$ .....	Besselfunktion erster Art $i$ -ter Ordnung
$k_0$ .....	Wellenzahl, $k_0 = 2\pi f/v$
$\Lambda$ .....	Eigenwertmatrix
$\lambda$ .....	Eigenwert
$\mathbf{M}$ .....	Hilfsmatrix
$M$ .....	größter betrachteter Phasen-Modus
$\mu$ .....	Eigenwert, $\mu = \sin(\theta)e^{j\varphi}$
$\mathbf{n}$ .....	Rauschvektor (Index entsprechend Signal denkbar)
$N$ .....	Blocklänge
$N_{BLOCK}$ .....	(EDMA-) Blocklänge
$n_i$ .....	Rauschen am $i$ -ten Mikrofon
$N_m$ .....	Anzahl betrachteter Phasen-Moden
$N_{mic}$ .....	Mikrofonanzahl
$N_m^s$ .....	Hilfsvariable, $N_m^s = N_m - 2$
$N_S$ .....	Quellenanzahl
$N_{VPS}$ .....	DFT-Länge der Methode Pei und Lai, halbe DFT-Länge der Methode mittels Spiegelung
$\omega$ .....	Kreisfrequenz
$\Omega_s$ .....	normierte Breite der halben Hauptkeule des Amplitudenganges von Zeitsignalfenstern
$\omega_a^c$ .....	Anregungsfunktion eines kontinuierlichen zirkularen Arrays
$\omega_{a,m}^c$ .....	$m$ -ter Anteil der Anregungsfunktion eines kontinuierlichen zirkularen Arrays
$\omega_m^c$ .....	$m$ -ter Fourierkoeffizient der Anregungsfunktion $\omega_a^c$ eines kontinuierlichen zirkularen Arrays
$\omega_s$ .....	Abtastkreisfrequenz
$\omega_a^s$ .....	Anregungsfunktion eines kontinuierlichen zirkularen Arrays
$\omega_{a,m}^s$ .....	$m$ -ter Anteil der Anregungsfunktion eines kontinuierlichen zirkularen Arrays
$\mathbf{p}_i$ .....	Positionsvektor des $i$ -ten Mikrofons im UCA

---

$\Phi$ .....	Eigenwertmatrix
$\varphi$ .....	Azimutwinkel der einfallenden Wellenfront bei einem UCA
$\Psi$ .....	Hilfsmatrix
$g$ .....	Hilfsvariable
$h$ .....	Hilfsvariable
$q$ .....	Hilfsvariable
$Q_p$ .....	p-te Quelle
$\mathbf{r}$ .....	Normalenvektor der Wellenfront
$R$ .....	Radius eines UCA
$\text{rang}\{\cdot\}$ .....	Rang einer Matrix
$\Re\{\cdot\}$ .....	Realteil
$\mathbf{R}_{kk}$ .....	Autokorrelationsmatrix des Signalvektors $\mathbf{k}$ (andere Indizes entsprechend Signal)
$r_{kk}$ .....	Autokorrelation des Signalvektors $\mathbf{k}$ (andere Indizes entsprechend Signal)
$r_m$ .....	Aliasfehler
$\mathbf{s}$ .....	Quellensignalvektor
$S$ .....	Fouriertransformierte des Zeitsignals $s$
$s$ .....	Zeitsignal
$s_e$ .....	Zeitsignal, das sich aus $s$ und der gleichen Anzahl Nullen zusammensetzt
$\sigma^2$ .....	Rauschleistung
$s_{MRRR}$ .....	Zeitsignal, das sich aus $s$ und dessen gespiegelte Variante zusammensetzt
$\text{span}\{\cdot\}$ .....	Lineare Hülle einer Matrix
$s_q$ .....	Quellensignal
$\mathbf{T}$ .....	Hilfsmatrix
$T$ .....	Periodendauer
$\tau$ .....	Laufzeitdifferenz
$\tau_i$ .....	Laufzeitdifferenz des i-ten Mikrofons
$\theta$ .....	Elevationswinkel der einfallenden Wellenfront bei einem UCA
$\theta_{err}$ .....	Fehlersignal des Elevationswinkels
$\mathbf{T}_U$ .....	Hilfsmatrix
$\mathbf{V}$ .....	Hilfsmatrix
$\mathbf{v}$ .....	Hilfsvektor
$v$ .....	Ausbreitungsgeschwindigkeit (Schallgeschwindigkeit)
$\varphi_{err}$ .....	Fehlersignal des Azimutwinkels
$\mathbf{W}^\dagger$ .....	Anregungsfunktion Matrix
$x$ .....	Hilfsvariable

$\mathbf{E}$ .....	komplexe Skalierungsmatrix bei ESPRIT
$Y_{\text{cmplx}}$ .....	Frequenzgang eines komplexwertigen Zeitsignals
$y_{\text{cmplx}}$ .....	komplexwertiges Zeitsignal
$y_{i,p}$ .....	Empfangssignal des $i$ -ten Mikrofons (Index $p$ entsprechend Algorithmus bzw. Array)
$\mathbf{y}_p$ .....	Empfangssignalvektor (Index $p$ entsprechend Algorithmus bzw. Array)
$Y_{\text{real}}$ .....	Frequenzgang eines reellwertigen Zeitsignals
$y_{\text{real}}$ .....	reelwertiges Zeitsignal
$\mathbf{z}$ .....	zusammengesetzter Signalvektor aus $\mathbf{y}_{G1}$ und $\mathbf{y}_{G2}$
$\zeta$ .....	Argument der Besselfunktionen, $\zeta = k_0 R \sin \theta$
$\text{Var}^c \{ \cdot \}$ .....	zirkulare Varianz

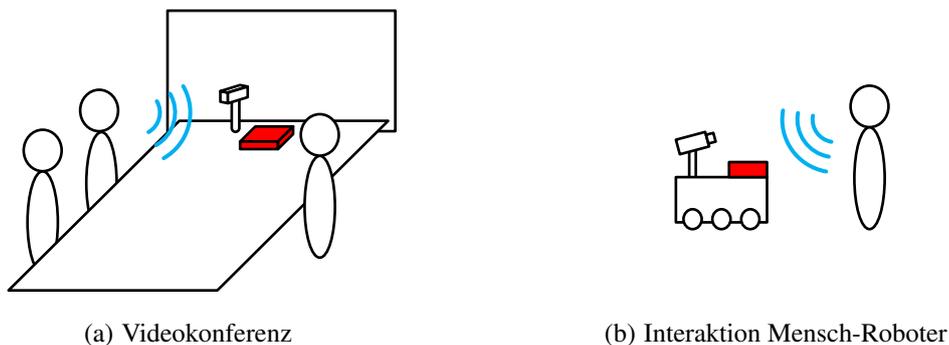
# 1 Einleitung

In dieser Arbeit wird die Entwicklung eines Echtzeit-Systems zur Lokalisierung von Audioquellen mittels dem UCA-ESPRIT-Algorithmus auf einen digitalen Signalprozessoren (DSP) vorgestellt. Dieses Kapitel dient zur Darstellung der Motivation sowie zur Vorstellung vorheriger Arbeiten. Darauf aufbauend wird die Problemstellung formuliert. Das Kapitel schließt mit Erläuterungen zur Notation und Struktur dieser Arbeit.

## 1.1 Motivation

In den letzten Jahrzehnten haben Systeme mit einer gewissen Intelligenz eine enorme Entwicklung vollzogen. Ziel dieser Entwicklung ist es zumeist das System den menschlichen Fähigkeiten anzunähern. Dies trifft auch auf das menschliche Gehör und dessen Fähigkeit der Ortung von Audioquellen zu. Wäre diese Ortung mittels eines Systems realisierbar, wären die sich dadurch ergebenden Möglichkeiten vielfältig.

Zwei denkbare Szenarien sind in Bild 1.1 veranschaulicht. Dabei repräsentiert der rote Quader das intelligente System.



**Bild 1.1:** Anwendungsszenarien

Im Unterbild (a) ist der Anwendungsfall einer Videokonferenz dargestellt. Aufgrund des Wissens über die Position des Sprechers können eingesetzte Kameras sich zum Sprecher ausrichten. Ebenso kann die Sprachqualität durch die sich ergebende Informationsredundanz

deutlich verbessert werden. Kurzum die Qualität von und die Interaktion in Videokonferenzen könnte deutlich verbessert werden.

Aber auch die Kommunikation zwischen Mensch und Roboter kann einen deutlich persönlicheren Charakter erhalten. Im Unterbild (b) ist ein solches Szenario dargestellt. Falls der Roboter Informationen über die Position des Kommunikationspartners erfasst, kann dieser entsprechende Interaktionssysteme in Richtung des Kommunikationspartners - als Reaktion auf die Lokalisierung - ausrichten. Dies sollte zu einer besseren subjektiven Wahrnehmung und Akzeptanz von Robotern beim Menschen führen.

Dieses Anwendungsszenario stellt gleichzeitig die Motivation des Forschungsprojektes Mensch-Roboter-Interaktionssystem (MERIT) des Departments Informations- und Elektrotechnik der HAW Hamburg dar. Ein Teilaspekt ist dabei die „Detektion, Lokalisation und Verfolgung menschlicher Kommunikationspartner“ [10].

Diese Masterarbeit ist Bestandteil des Forschungsprojektes MERIT. Sie soll die Möglichkeit der Echtzeit-Lokalisierung von Audioquellen, in besonderer Hinsicht von Sprachsignalen, auf Basis des 1993 veröffentlichten Algorithmus UCA-ESPRIT [39] untersuchen.

## 1.2 Stand des Forschungsprojektes

Im Rahmen des Forschungsprojektes MERIT sind bereits einige Diplom- und Masterarbeiten verfasst worden. Die Aspekte der wesentlichen Arbeiten und der aktuelle Stand des Forschungsprojektes seien hier kurz vorgestellt.

Grundsätzlich lassen sich die Forschungsarbeiten in zwei Gruppen einteilen: Die erste Gruppe betrachtet auf Korrelation basierte Verfahren; die zweite Gruppe befasst sich hingegen mit unterraumbasierten Verfahren. Diese beiden Verfahren unterscheiden sich in der prinzipiellen Art der Berechnung und weisen unterschiedliche Eigenschaften auf. Der numerische Aufwand der auf Korrelation basierten Verfahren ist zumeist geringer, allerdings unterliegt die Auflösung einem Raster. Dies ist bei unterraumbasierten Verfahren nicht der Fall, da eine kontinuierliche Auflösung erreicht wird. Auch ist mit einer höheren Robustheit der Verfahren gegenüber Störungen zu rechnen. Der numerische Aufwand ist jedoch deutlich größer und die Verfahren sind in der Regel lediglich für schmalbandige Signale definiert. Folglich ist stets die Erweiterung auf breitbandige Signale notwendig.

Der zurzeit verwendete, auf Korrelation basierende Algorithmus *Multichannel Cross-Correlation Coefficient*-Algorithmus (MCCC-Algorithmus) ist zuerst von Kolja Pikora in [24] untersucht worden. In [15] ist dieser Algorithmus von Markus Müller auf zwei Sprecher erweitert und um eine Kreuzpeilung ergänzt worden. Diese Ergänzung ermöglicht auch die Abstandsschätzung.

Im Rahmen der Betrachtung der unterraumbasierten Verfahren sind eine Reihe Algorithmen untersucht worden. Yuan Cheng und Bing Li haben zuerst den *Multiple Signal Classification*-Algorithmus (MUSIC-Algorithmus) für schmalbandige Audiosignale in [5] studiert. In [29] wird von Anshul Kant Saxena dieser MUSIC-Algorithmus auf breitbandige Signale (eine Überlagerung von Sinussignalen verschiedener Frequenzen) angewendet. Von Kolja Pikora wird in [23] eine Variante dieses Algorithmus (Root-MUSIC) betrachtet. In [40] werden von Zhonglei Zou erstmalig ein zirkulares Mikrofonarray verwendet und entsprechende Algorithmen untersucht. Dabei handelt es sich auch um den UCA-ESPRIT-Algorithmus. Es handelt sich dabei um eine auf zirkulare Arrays abgewandelte Version des Estimation of Signal Parameters via Rotational Invariance Techniques-Algorithmus (ESPRIT-Algorithmus). Die grundsätzliche Funktionsfähigkeit dieses Algorithmus bei der Betrachtung von Sprachsignalen ist nachgewiesen worden. Allerdings ist sich unter anderem auf die azimutale Richtungsbestimmung beschränkt worden, obwohl der Algorithmus prinzipiell die Möglichkeit bietet die Elevationsrichtung zu bestimmen.

### 1.3 Problemstellung

Aufbauend auf den Untersuchungen von Zhonglei Zou [40] soll in dieser Arbeit der UCA-ESPRIT-Algorithmus weiter betrachtet werden. Ziel ist es, den Algorithmus auf Audiosignale - insbesondere auf Sprachsignale - anzuwenden und gegebenenfalls anzupassen. Im Fokus dieser Entwicklung soll sowohl die Echtzeitfähigkeit als auch eine Studie über das Potenzial des Algorithmus stehen. Wünschenswert ist eine Lokalisierung sowohl in Azimut- als auch in Elevationsrichtung und die visuelle Darstellung der Ergebnisse der Lokalisierung.

Als Zielhardware ist der digitale Signalprozessor (DSP) TMS320C6713B der Firma *Texas Instruments* im D.Module.C6713-Modul in Verbindung mit dem achtkanaligen Analog-Digital-Umsetzermodule D.Module.PCM3003 der Firma *D-SignT* angedacht. Der UCA-ESPRIT-Algorithmus impliziert die Verwendung eines gleichverteilten, zirkularen Mikrofonarrays.

Die Studie soll die Genauigkeit, Stabilität im Sinne der Varianz, Robustheit in realer Umgebung und Reaktionsfähigkeit des entwickelten Systems untersuchen. Grundsätzliche Aussagen über diese Eigenschaften bei der Verwendung von Sprachsignalen werden angestrebt.

Für eine quantitative Einschätzung über wünschenswerte Eigenschaften seien die zuvor beschriebenen Anwendungsszenarien betrachtet:

Im Rahmen einer Videokonferenz beschreibt die Vorstellung von vier anwesenden Sprechern an einem Tisch ein realistisches Szenario. Ausgehend von einem Abstand von 1 m bis 2 m zwischen Array und Sprecher, erscheint eine Genauigkeit von  $10^\circ$  in beide Richtungen als ausreichend. Der relevante Bereich in Elevationsrichtung kann auf  $20^\circ$  bis  $70^\circ$  eingeschränkt werden. Die Reaktionszeit sollte unter 0,5 s liegen. Stabilität und Robustheit sollten möglichst hoch sein. Die Kommunikation beruht in der Regel auf einem Dialog. Die Wahrscheinlichkeit,

dass zwei Quellen gleichzeitig über einen längeren Zeitraum (mehrere Sekunden) aktiv sind, kann daher als sehr gering eingestuft werden. Dies schließt das Vorhandensein von zwei oder mehreren aktiven Quellen über einen kurzen Zeitraum (weniger als eine Sekunde) jedoch nicht aus.

Betrachtet man hingegen die Interaktion zwischen Mensch und Roboter, hat die Positionierung des Arrays und die Umgebung großen Einfluss auf die entsprechende Spezifikation. Allerdings sind für die Systemeigenschaften ähnliche Größenordnungen sinnvoll. Auch die Aussage über die Anzahl der Sprecher trifft in einigen Fällen zu.

An dieser Stelle sei darauf hingewiesen, dass die Spezialisierung auf ein konkretes Anwendungsszenario nicht Teil dieser Arbeit ist.

Neben den Entwicklungen bezüglich der Lokalisierung soll für Test- und Demonstrationszwecke ein variabler Phasenschieber entwickelt werden. Aufbauend auf diesem Phasenschieber soll es möglich sein, Quellen zu simulieren und die Einfallswinkel während der Ausführung der Simulation zu variieren.

## 1.4 Struktur der Arbeit

In Kapitel 2 werden die entsprechenden Grundlagen und zu berücksichtigenden Aspekte zur hardwarenahen Implementierung aufgeführt.

Anschließend erfolgt in Kapitel 3 die Darstellung der allgemeinen Problematik der Lokalisierung in theoretischer Hinsicht. Es folgt die Vorstellung grundlegender unterraumbasierter Verfahren.

Die Entwicklung des variablen Phasenschiebers bzw. des Quellensimulators wird in Kapitel 4 vorgestellt.

In Kapitel 5 wird die Theorie des UCA-ESPRIT-Algorithmus vorgestellt. Anhand entsprechender Simulationen wird die Leistungsfähigkeit bei komplexwertigen, schmalbandigen Signalen bei gegebener Arraystruktur bestimmt.

Die Erarbeitung eines Konzepts zur Verarbeitung von Audiosignalen und entsprechender Aufbereitung der Rohdaten wird in Kapitel 6 theoretisch und anhand von Simulationen diskutiert.

In Kapitel 7.3 erfolgen entsprechende Simulationen bezüglich des entwickelten Konzeptes. Des Weiteren wird die Implementierung auf dem DSP dargestellt.

Die abschließende Evaluation inklusive Tests und Fehlerkorrektur wird in Kapitel 9 durchgeführt. Kapitel 10 fasst die Ergebnisse zusammen und gibt einen Ausblick auf mögliche weitere Arbeiten.

## 1.5 Notation

Um die Lesbarkeit dieser Arbeit zu erhöhen, wird eine einheitliche Notation verwendet. Diese ist in Tabelle 1.1 zusammengefasst.

Bezeichnung	Kennzeichnung	Beispiel
Skalar	kleines Zeichen	$a$
Vektor	kleines, fettes Zeichen	$\mathbf{a}$
Matrix	großes, fettes Zeichen	$\mathbf{A}$
transponierte(r) Matrix / Vektor	hochgestellter Buchstabe T	$\mathbf{A}^T, \mathbf{a}^T$
konjugiert komplexe(r) Matrix / Vektor	hochgestellter Stern	$\mathbf{A}^*, \mathbf{a}^*$
adjungierte(r) Matrix / Vektor	hochgestelltes Kreuz	$\mathbf{A}^\dagger, \mathbf{a}^\dagger$
zeitdiskrete Signalfolge	eckige Klammern	$a[n] = a(nT)$
zeitkontinuierliche Signalfunktion	runde Klammern	$a(t)$
Schätzgröße	Dach über Symbol	$\hat{a}$
Mittelwert	Strich über Symbol	$\bar{a}$

**Tabelle 1.1:** Notation

Um Missverständnisse bei der Bezeichnung von gewissen Matriceigenschaften auszu-schließen, sind die in dieser Arbeit verwendeten Begriffe in Tabelle 1.2 zusammengefasst.

Bezeichnung	Eigenschaft
symmetrische komplexe Matrix	$\mathbf{A}^T = \mathbf{A}$
hermitesche Matrix	$\mathbf{A}^\dagger = \mathbf{A}$
orthogonale Matrix	$\mathbf{A}^T \mathbf{A} = \mathbf{I}$
unitäre Matrix	$\mathbf{A}^\dagger \mathbf{A} = \mathbf{I}$

**Tabelle 1.2:** Matriceigenschaften

Weiter sei an dieser Stelle darauf hingewiesen, dass beim Mitteln von Winkelgrößen stets das zirkulare Mitteln gemeint ist. Gleiches gilt für die Varianz. Für Details sei auf Anhang A.1 verwiesen.

Sämtliche Personenbezeichnungen werden in der männlichen Form angegeben. Die Verwendung der männlichen Form impliziert in diesem Kontext die weibliche Form.

## 2 Hardwarenahe Implementierung

Wie im vorherigen Abschnitt 1.3 dargelegt, ist die hardwarenahe, echtzeitfähige Implementierung auf einem DSP C6713 von *Texas Instruments* ein wesentlicher Bestandteil dieser Arbeit. Um entsprechende Schlussfolgerungen in den folgenden Kapiteln nachvollziehen zu können, seien hier die wesentlichen Elemente zur hardwarenahen Implementierung diskutiert. Dazu zählt neben der Vorstellung und Erläuterung der Hardware auch die Beschreibung der Vorgehensweise der Implementierung eines echtzeitfähigen Systems.

### 2.1 Mikrofonarray und -verstärker

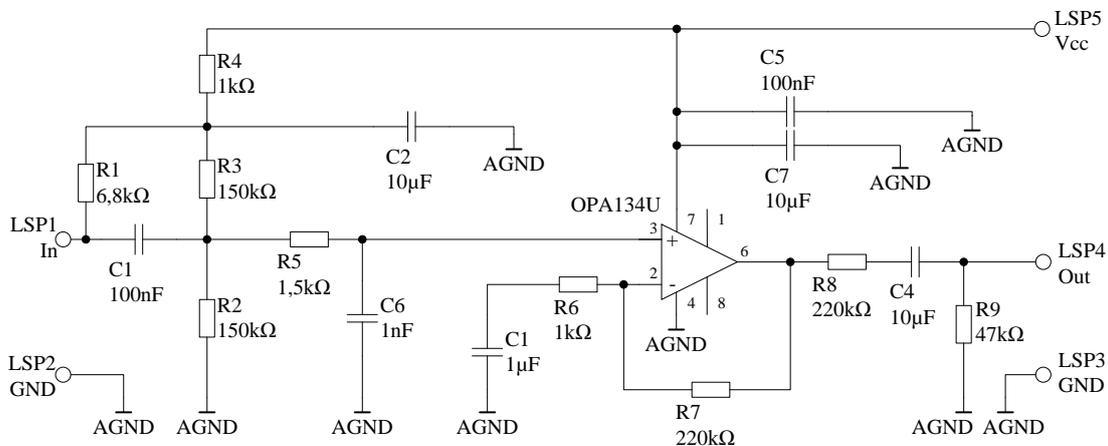
Das in dieser Arbeit verwendete Mikrofonarray ist in Bild 2.1 abgebildet. Es handelt sich dabei um ein gleichförmig verteiltes, achtkanaliges Mikrofonarray mit einem Durchmesser von  $2R = 9$  cm. Die Mikrofonkapseln sind omnidirektionale Elektret-Kondensator-Mikrofonkapseln des Typs WM-52BT. Ihr Frequenzbereich liegt zwischen 20 Hz und 16 kHz, wenn die Versorgungsspannung bei 1,5 V liegt [20]. Sie sind somit grundsätzlich für die Aufnahme von Audiosignalen - vor allem von Sprachsignalen - geeignet.



**Bild 2.1:** Mikrofonarray

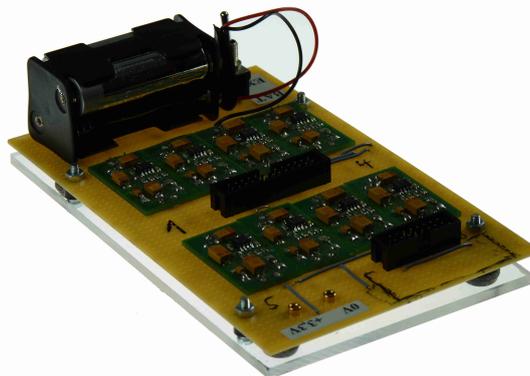
Da die Ausgangssignale eine sehr geringe Amplitude aufweisen und somit bei der Analog-Digital-Umsetzung keineswegs mit Vollaussteuerung zu rechnen ist, werden die Signale mit

Hilfe eines gegebenen Mikrofonsignalverstärkers deutlich verstärkt. Die dafür verwendete Schaltung ist in Bild 2.2 zu sehen. Die Verstärkung erfolgt dabei im Wesentlichen mit Hilfe einer nicht-invertierenden Operationsverstärker-Schaltung. Ebenfalls ist es notwendig das Mikrofonsignal um die halbe Versorgungsspannung anzuheben. So können positive als auch negative Mikrofonsignale mit Hilfe der hier verwendeten Operationsverstärkerschaltung (lediglich positive Versorgungsspannung) verarbeitet werden. Mit Hilfe des Ausgangskondensators wird dieser Gleichspannungsanteil anschließend entfernt. Weiter werden die Mikrofone mit der geforderten Versorgungsspannung entsprechend dem Datenblatt versorgt. Die Versorgungsspannung der Verstärkerschaltung liegt bei  $V_{cc} = 3,3\text{ V}$ . Die Schaltung gibt somit Wechselspannungssignale mit einer maximale Amplitude von  $1,65\text{ V}$  aus.



**Bild 2.2:** Schaltplan Mikrofonverstärker

Um acht Kanäle gleichzeitig verarbeiten zu können, werden acht Verstärkerschaltungen parallel geschaltet. Ein Bild der resultierenden Gesamtplatine ist in Bild 2.3 abgebildet.

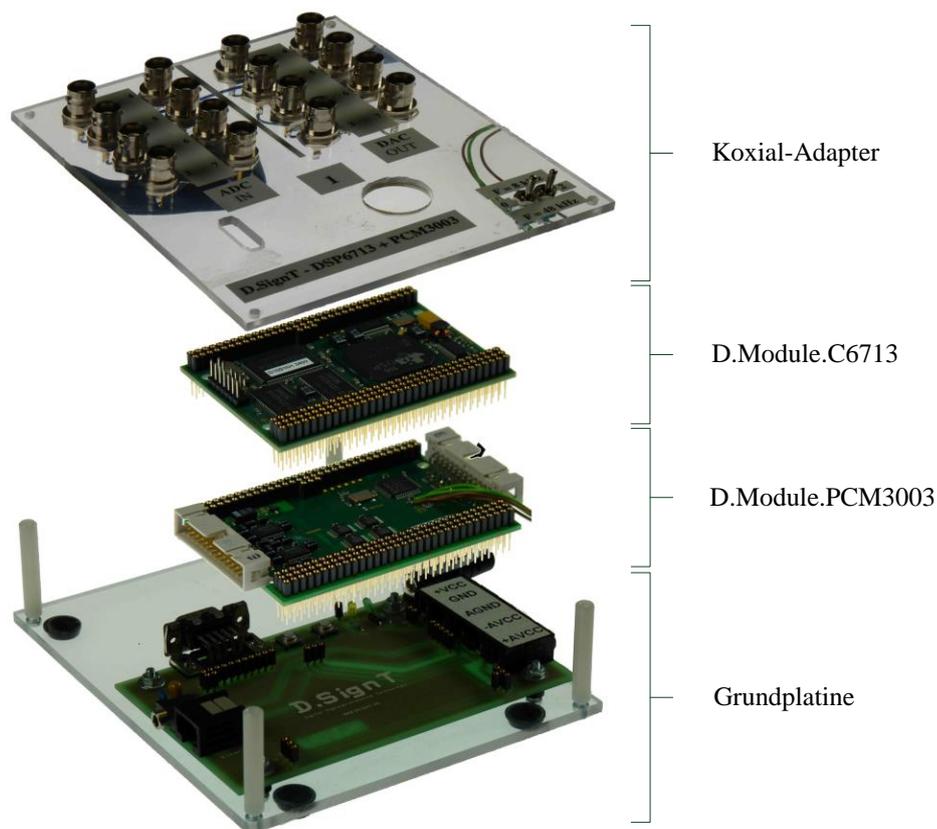


**Bild 2.3:** Mikrofonverstärkerplatine

Neben den Ein- und Ausgangssteckern und den acht Ausführungen der Verstärkerplatine ist weiter ein Batteriehalter zur autonomen Versorgung der Schaltung zu sehen.

## 2.2 DSP-System

Die Lokalisierung der Audioquellen soll in Echtzeit auf einem DSP TMS320C6713B erfolgen. Das dafür verwendete DSP-System ist in Bild 2.4 abgebildet.



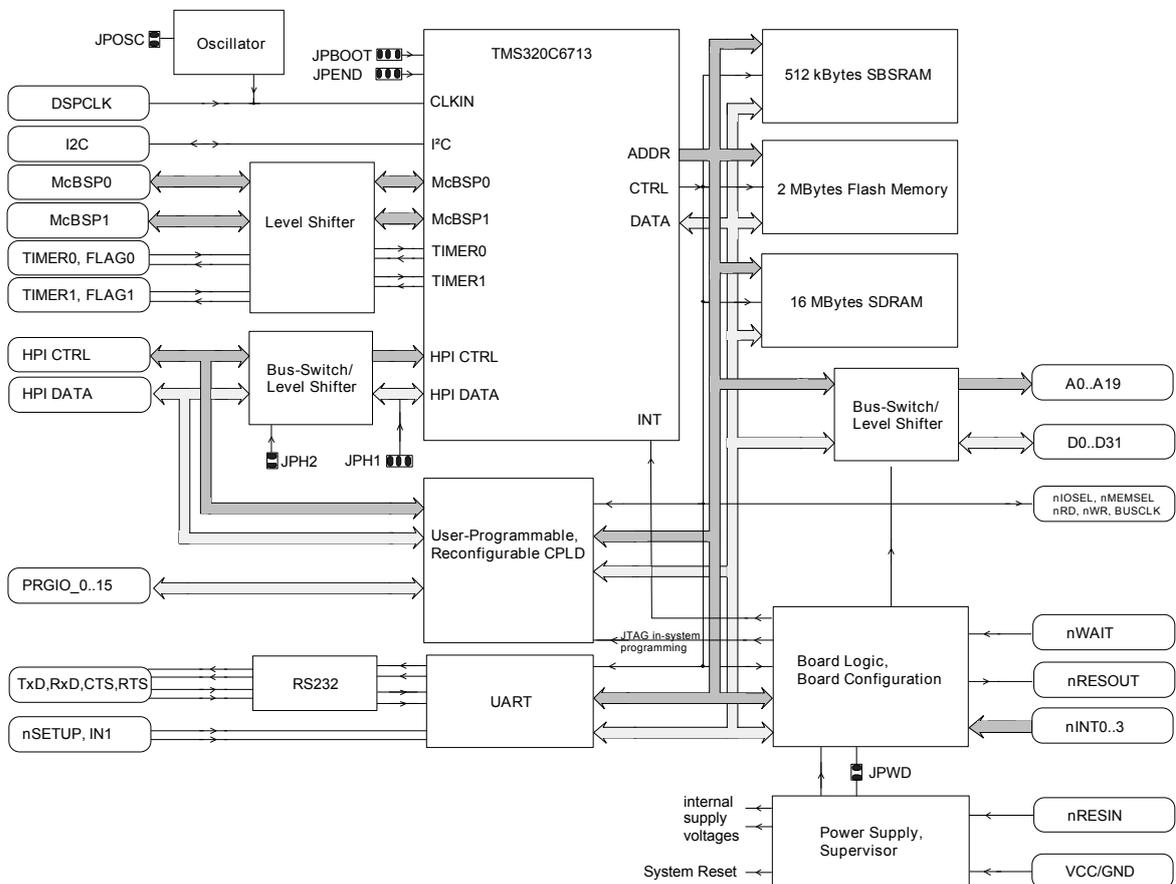
**Bild 2.4:** DSP-System

Wie Bild 2.4 zu entnehmen ist, setzt sich das DSP-System aus vier Modulen zusammen. Die Koxial-Adapter werden für den Anschluss entsprechender Koxialkabel benötigt. Die Grundplatine stellt weitere entsprechende Anschlüsse zur Verfügung. Diese umfassen unter anderem Anschlüsse zur Spannungsversorgung und RS232-Kommunikation. Bei den beiden weiteren Modulen handelt es sich einerseits um das eigentliche DSP-Modul mit entsprechen-

der Peripherie, andererseits um das Modul mit den Analog-Digital-Umsetzern. Aufgrund der Komplexität dieser Module seien sie und ihre Kommunikation nachfolgend näher erläutert.

### 2.2.1 D.Module.C6713

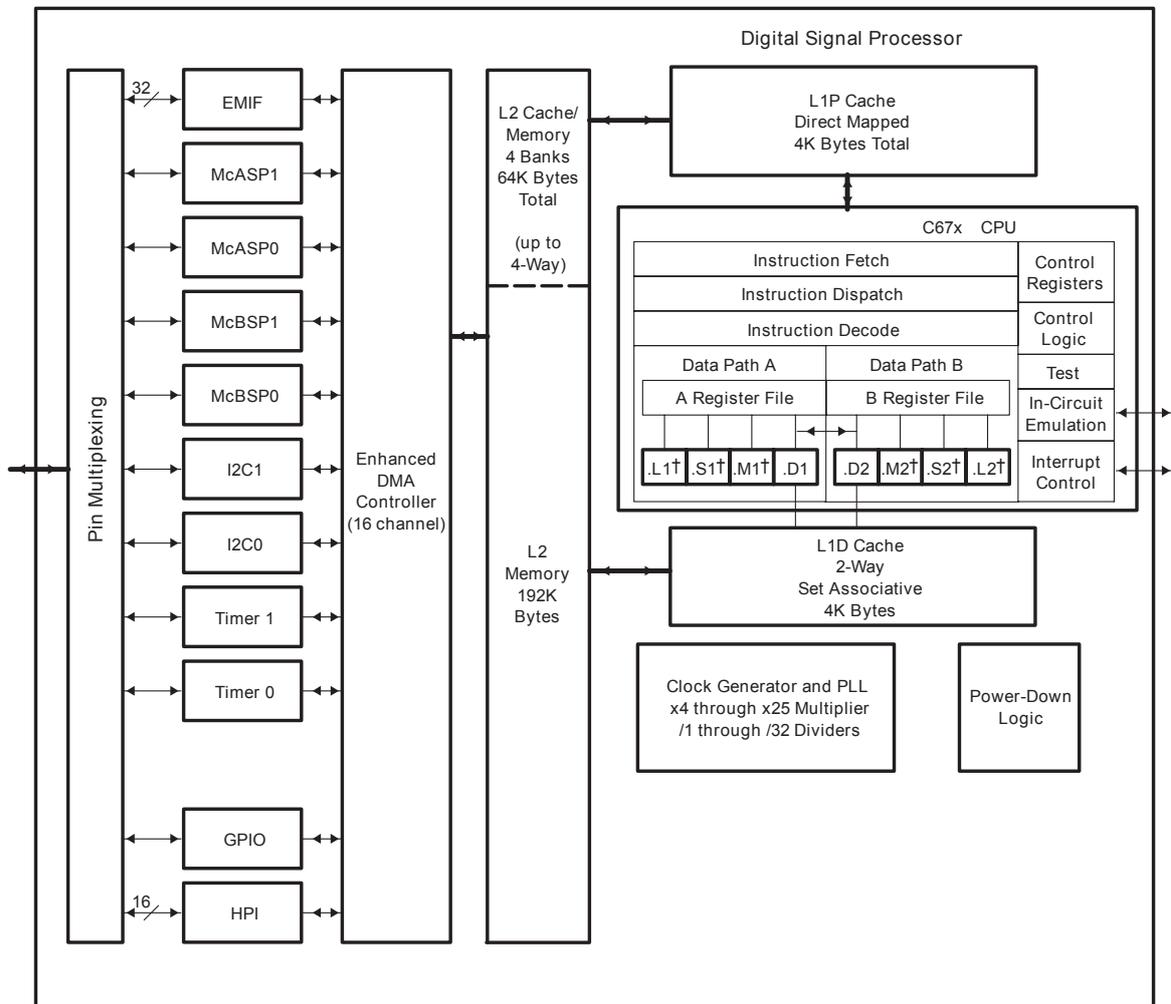
Das Modul D.Module.C6713 ist ein Entwicklungsboard. Es beinhaltet die notwendige Peripherie für den Betrieb des DSP TMS320C6713B aus der TMS320C6000-Familie von *Texas Instruments*. Von besonderer Bedeutung in dieser Arbeit ist die *Universal Asynchronous Receiver Transmitter*-Schnittstelle (UART-Schnittstelle) sowie die *Multichannel Buffered Serial Port* Datenverbindungen (McBSP Datenverbindungen). Alle weiteren Peripherie-Bausteine können dem Blockschaltbild aus Bild 2.5 entnommen werden.



**Bild 2.5:** Blockschaltbild D.Module.C6713, Quelle [17, S. 7] (angepasst)

Der TMS320C6713B DSP ist ein *Floating-Point* DSP mit acht parallelen Recheneinheiten. Diese ermöglichen eine zeitgerechte Verarbeitung rechenaufwendiger Algorithmen. Bei einer

Taktfrequenz von 300 Hz sind 1800 MFLOPS<sup>1</sup> möglich. In Bild 2.6 ist das Blockschaltbild des TMS320C6713B abgebildet.



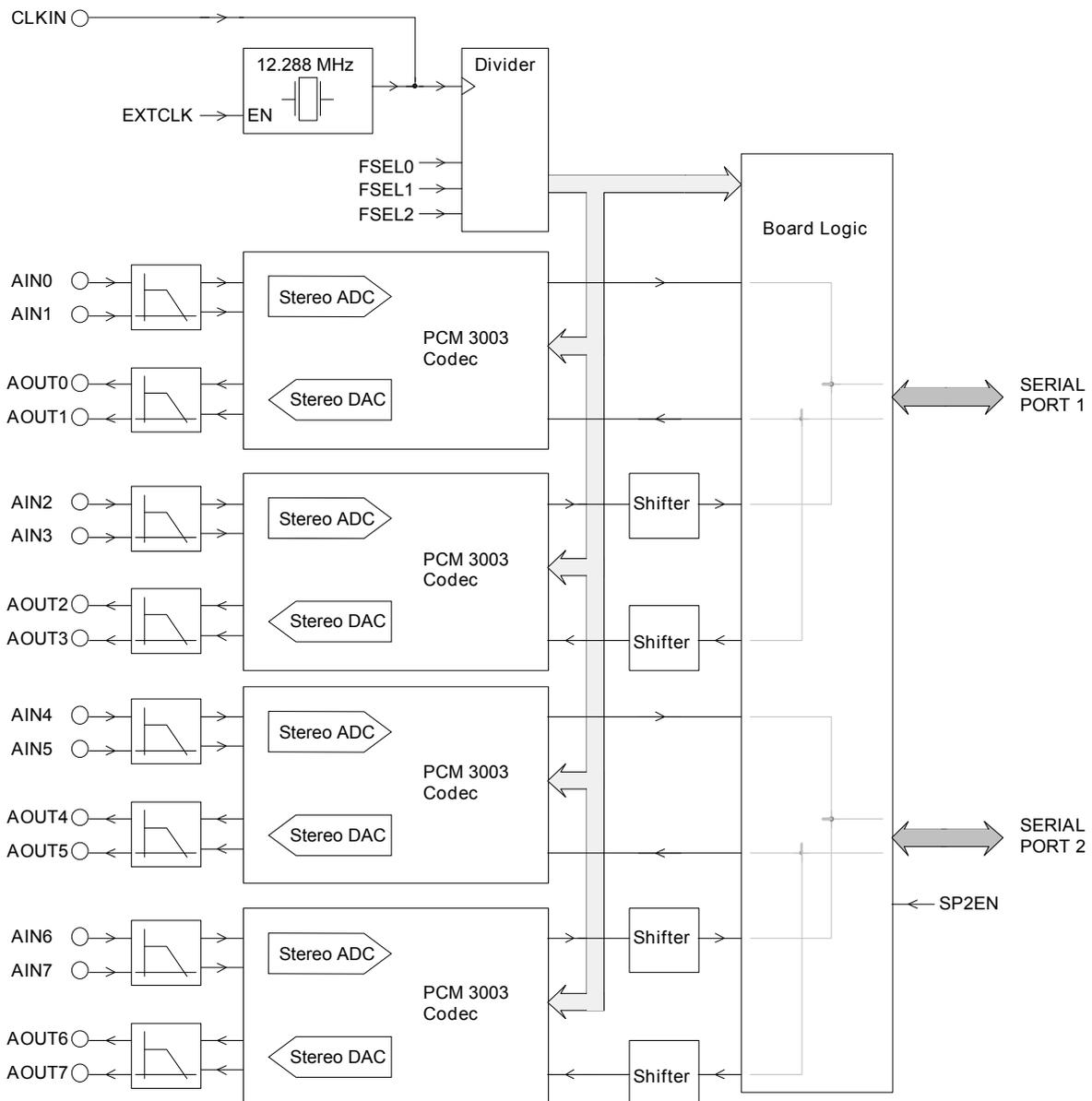
**Bild 2.6:** Blockschaltbild TMS320C6713B, Quelle [33, S. 13]

## 2.2.2 D.Module.PCM3003

Das D.Module.PCM3003 ist eine Audiocodex Platine, die unter anderem besonders für Audio-Verarbeitung an Mikrofonarrays geeignet ist. Das Blockschaltbild ist in Bild 2.7 abgebildet. Es besteht im Wesentlichen aus vier PCM3003-Modulen von *Texas Instruments*. Diese ermöglichen bis zu acht Kanäle gleichzeitig auszulesen. Die Abtastrate kann dabei in sechs Schritten zwischen 8 kHz und 48 kHz eingestellt werden. Das analoge Tiefpassfilter

<sup>1</sup>engl.: **M**ega **F**loating-Points **O**perations **P**er **S**econd, dt.: Mega Floating-Point Operationen pro Sekunde

hat eine Grenzfrequenz von 150 kHz. Die Analog-Digital- bzw. Digital-Analog-Umsetzung erfolgt mit einem Delta-Sigma-Umsetzer mit 64-facher bzw. achtfacher Überabtastung.



**Bild 2.7:** Blockschaltbild D.Module.PCM3003, Quelle [18, S. 4] (angepasst)

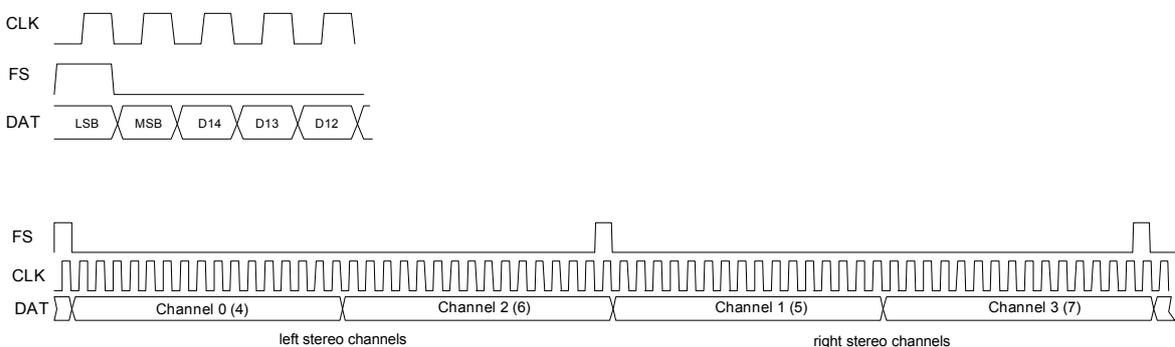
Der Eingangsspannungsbereich liegt zwischen -1,8 V und 1,8 V. Bei der Verwendung des Mikrofonverstärkers aus Abschnitt 2.1 ist somit nahezu mit Vollaussteuerung zu rechnen. Der Dynamikbereich wird annähernd vollständig genutzt.

Für die Ausgabe der Daten verfügt das D.Module.PCM3003 über zwei serielle Schnittstellen.

### 2.2.3 Kommunikation

Die Kommunikation der beiden Module erfolgt über serielle Schnittstellen.

Das D.Module.PCM3003-Modul besitzt zwei serielle Schnittstellen. Diese verfügen für das Senden über drei Pfade je Schnittstelle. Dabei handelt es sich um den Takt-, den Synchronisations- und den Datenpfad. Das Format dieser Signale für die erste und zweite Schnittstelle kann Bild 2.8 entnommen werden. Das Format für die zweite serielle Schnittstelle ist durch Klammern abgegrenzt. Deutlich wird, dass die Reihenfolge nicht mit der numerischen Reihenfolge der Zahlen 0 bis 7 übereinstimmt. Dies kann jedoch durch eine einfache *Look-Up*-Tabelle im Programm korrigiert werden.



**Bild 2.8:** Datenformat D.Module.PCM3003, Quelle [18, S. 11]

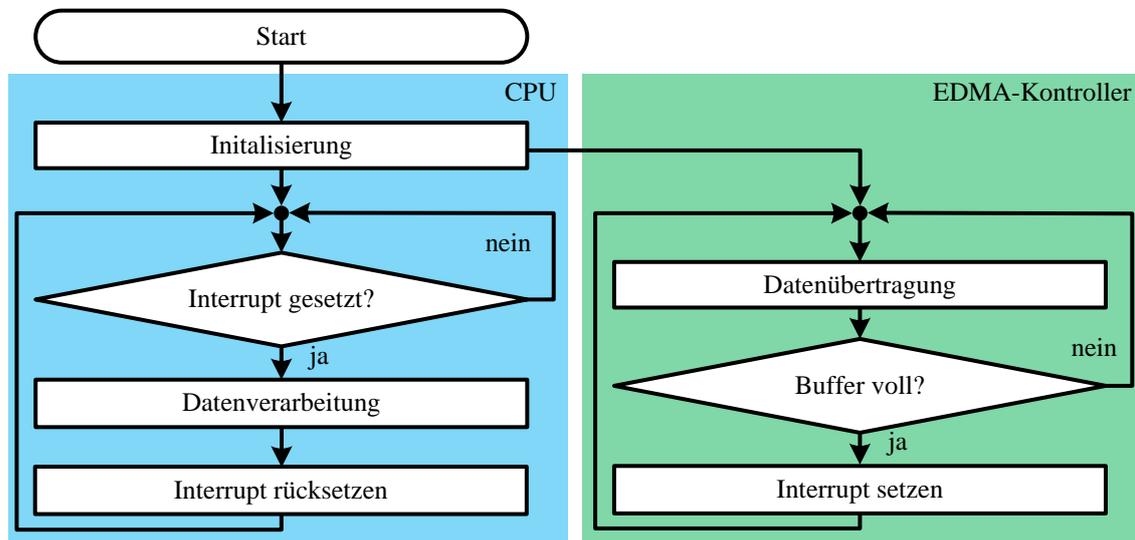
Der Empfang der Daten des D.Module.PCM3003s erfolgt über die serielle Schnittstelle McBSP des DSPs. Diese verfügt für das Empfangen ebenfalls über einen Daten-, Synchronisations- und Taktpfad.

## 2.3 Blockweise Programmierung

Wie bereits im Abschnitt 2.2.3 beschrieben, erfolgt der Datenaustausch zwischen den beiden Modulen des DSP-Systems über serielle Schnittstellen. Sobald die Daten in der McBSP-Schnittstelle vorhanden sind, kann die McBSP-Schnittstelle einen Interrupt werfen. Aufbauend auf diesem Interrupt existieren im Wesentlichen zwei Möglichkeiten die Daten zu verarbeiten.

Eine Möglichkeit besteht darin, die Interrupts direkt von der *Central Processing Unit* (CPU) verarbeiten zu lassen. Um Daten zeitgerecht abzuholen, muss die Verarbeitung des Algorithmus im Hauptprogramm fortlaufend unterbrochen werden. Dieser Effekt wird durch die Verwendung des *Enhanced Direct Memory Access*-Kontrollers (EDMA-Kontroller) vermieden. Dieser übernimmt das Verarbeiten der McBSP-Daten und speichert diese zunächst in einem Buffer. Sobald dieser voll ist, wird ein Interrupt geworfen und die CPU kann die

Daten anschließend blockweise verarbeiten. In Bild 2.9 sind diese Zusammenhänge grafisch dargestellt.



**Bild 2.9:** Flussdiagramm EDMA-Grundgerüst

Deutlich wird die Aufteilung von Datenübertragung und -verarbeitung auf zwei verschiedenen Modulen und die Kommunikation dieser Module über einen Interrupt. Die Verwendung des EDMA-Kontrollers verringert somit die Belastung der CPU mit Datenübertragung deutlich.

Allerdings muss beachtet werden, dass der EDMA-Kontroller die Daten, sobald diese vorliegen, instantan in den Speicher schreibt. Dies hat zur Folge, dass Daten zu früh überschrieben werden können, falls Datenverarbeitung und -übertragung auf denselben Speicher zugreifen. Um dies zu vermeiden wird die so genannte Ping-Pong-Technik verwendet. Dazu werden zwei Buffer - Ping und Pong - angelegt. Jeweils ein Buffer ist für die Verarbeitung und einer für die Übertragung zuständig. Sobald der Buffer der Übertragung voll ist, wechseln die Buffer ihre Zugehörigkeit. In Bild 2.10 ist dieser Zusammenhang grafisch dargestellt.

Durch diese Vorgehensweise steht fast die komplette Zeit, die für das Empfangen einer gewissen Blocklänge benötigt wird, für das Verarbeiten von Daten zur Verfügung. Allerdings sollte weiter darauf geachtet werden, dass die Verarbeitung in dieser Zeit abgeschlossen ist (Stichwort Echtzeit) oder die Daten zwischengespeichert werden.

Diese beschriebene Technik stellt die Grundlage für alle nachfolgenden Programme dar. Ein entsprechendes Grundgerüstprogramm ist vom Hersteller der Hardware-Module zur Verfügung gestellt worden. Dieses umfasst im Wesentlichen das Flussdiagramm aus Bild 2.9. Um die Übersichtlichkeit der Programme zu verbessern, sind die wesentlichen Einstellungen des EDMA-Kontrollers in der Funktion `set_edma()` zusammengefasst. Auch ist der

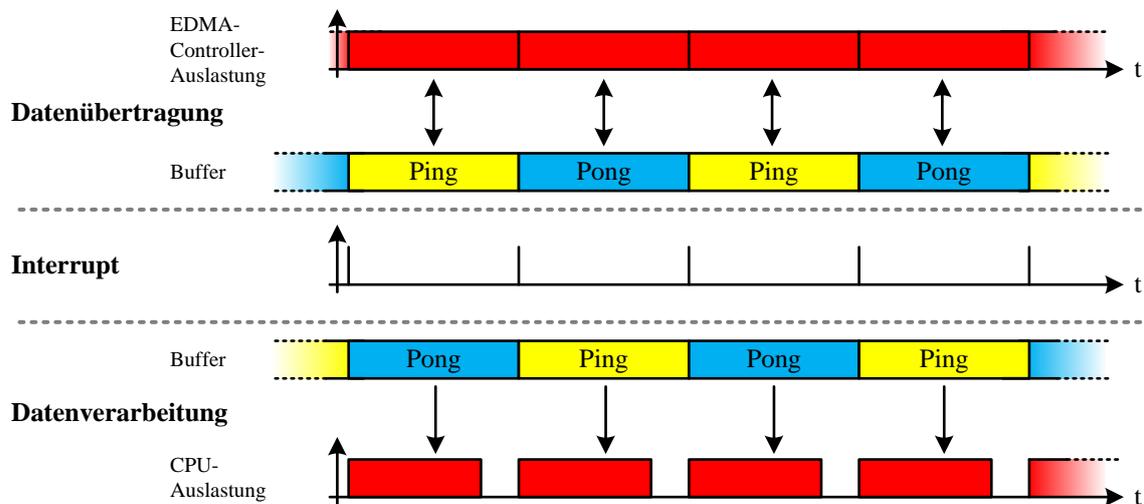


Bild 2.10: Ping-Pong-Technik

Interrupthandler ausgelagert. Die entsprechenden Funktionen und Variablen sind in den Dateien `init_edma.c` bzw. `init_edma.h` (s. Anhang B) zu finden.

## 2.4 Compiler Optimierungsstufen

Als Entwicklungsumgebung wird *Code Composer Studio* in der Version 5.1 von *Texas Instruments* verwendet. Diese Entwicklungsumgebung ist für die Programmierung der Produkte von *Texas Instruments* optimiert. Der Ablauf der Programmierung ist in Bild 2.11 dargestellt.

Um optimierte Programme für die entsprechende Hardware zu generieren, bietet der Compiler von *Texas Instruments* eine Vielzahl von Einstellungen. Die wesentlichen Möglichkeiten sind in Tabelle 2.1 zusammengefasst. Um ein echtzeitfähiges Programm zu generieren, bietet sich die Optimierungsstufe `-O3` an. Bei dieser Optimierungsstufe werden alle Optimierungsschritte durchgeführt. Das Potenzial des zugrundeliegenden C-Programms wird somit optimal ausgenutzt.

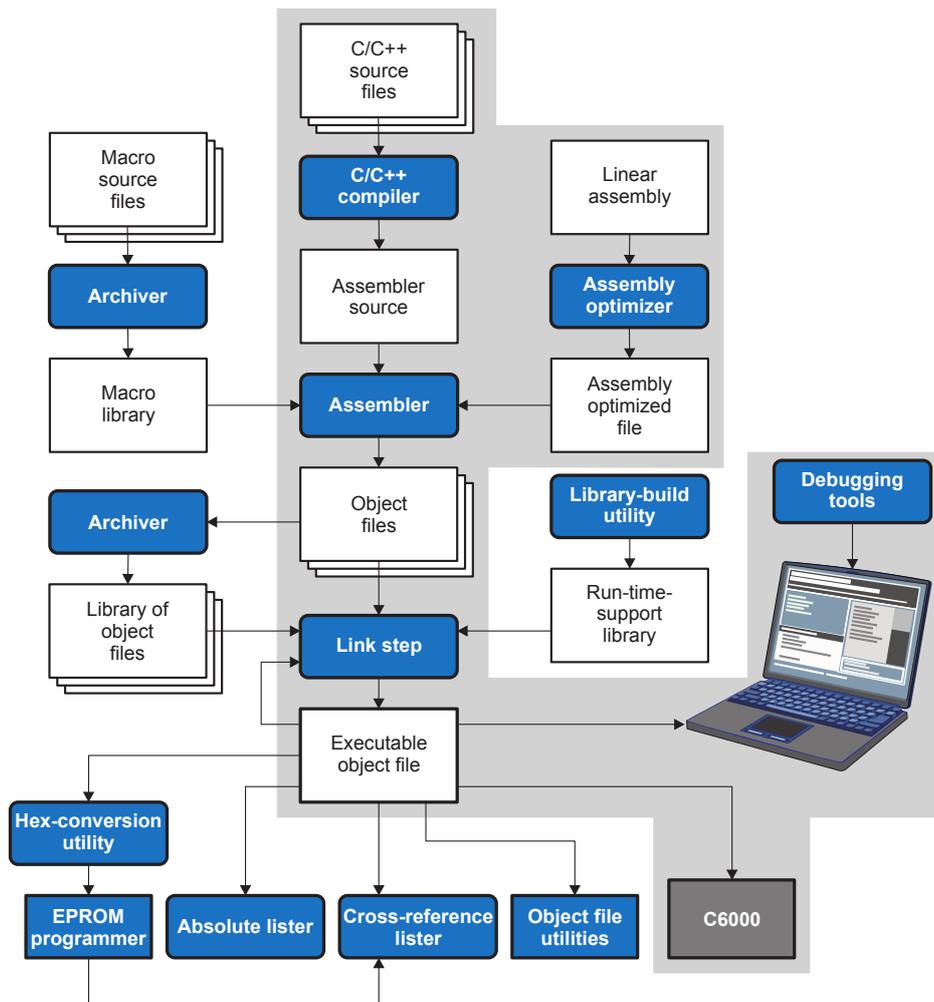


Bild 2.11: Ablauf des Programmierprozesses, Quelle [35, S. 15]

Bezeichnung	Optimierungen
-O0	<ul style="list-style-type: none"> <li>• Simplifizierung des Kontrollflussdiagramms</li> <li>• Speicherung von Variablen in Registern</li> <li>• Durchführung von <i>loop rotation</i><sup>2</sup></li> <li>• Entfernung von ungenutztem Quellcode</li> <li>• Simplifizierung von Ausdrücken und Aussagen</li> <li>• Ersetzung von <i>Inline</i>-Funktionen durch Funktionsaufrufe</li> </ul>
-O1	<ul style="list-style-type: none"> <li>• alle Optimierung aus -O0</li> <li>• Entfernung von lokalen Kopiervorgängen von Konstanten</li> <li>• Entfernung von ungenutzten Zuweisungen</li> <li>• Entfernung lokaler gemeinsamer Teilausdrücke</li> </ul>
-O2 (Default)	<ul style="list-style-type: none"> <li>• alle Optimierung aus -O1</li> <li>• Durchführung von <i>Software-Pipelining</i> und <i>loop optimizations</i><sup>3</sup></li> <li>• Entfernung globaler gemeinsamer Teilausdrücke</li> <li>• Entfernung globaler ungenutzter Zuweisungen</li> <li>• Konvertierung von Arrayreferenzierung in Pointernotation</li> <li>• Durchführung von <i>loop unrolling</i><sup>4</sup></li> </ul>
-O3	<ul style="list-style-type: none"> <li>• alle Optimierung aus -O2</li> <li>• Entfernung aller nicht aufgerufener Funktionen</li> <li>• Simplifizierung von Funktionen mit ungebrauchten <code>return</code>-Anweisungen</li> <li>• <i>Inline</i>-Aufruf von kleinen Funktionen</li> <li>• Umsortierung von Funktionsaufrufen</li> <li>• Entfernung unnötiger Parameterübergaben</li> </ul>

**Tabelle 2.1:** Optimierungsstufen (in Anlehnung an [35, S. 59f])

Durch Anwendung der umfangreichen Optimierungsverfahren wird der Unterschied zwischen der Dauer von *Floating*- und *Fixed-Point*-Arithmetik relativiert. Vor allem das *Software-Pipelining* und *loop optimizations* bewirkt, dass entsprechende Wartezeiten der Assembler-

<sup>2</sup>Durch die Durchführung von *loop rotation* werden Schleifenbedingungen erst am Ende einer Schleife überprüft. Für weitere Informationen sei auf [35, S. 103] verwiesen.

<sup>3</sup>Die Durchführung von *Software-Pipelining* und *loop optimizations* bewirkt, dass möglichst viele Rechenschritte in Schleifen parallel ausgeführt werden und somit das Potenzial paralleler Recheneinheiten optimal ausgenutzt wird. Für weitere Informationen sei auf [35] verwiesen.

<sup>4</sup>*loop unrolling* bedeutet, dass die Schleifenbedingung von kleinen Schleifen entfernt wird und stattdessen der Schleifenkörper mehrfach im Quellcode eingefügt wird.

Befehle optimal ausgenutzt werden. Durch Verwendung einer *Fixed-Point*-Arithmetik ist somit kein deutlicher Geschwindigkeitsvorteil auf einem *Floating-Point*-DSP zu erwarten.

## 2.5 TI C67x DSPLib

Die *TI C67x DSPLib* ist eine von *Texas Instruments* bereitgestellte Bibliothek, die eine Vielzahl an optimierten Assembler-Funktionen für die C67x-Familie beinhaltet. Durch die Verwendung dieser Assembler-Funktionen ist es möglich, die Ausführdauer eines Programms nochmals deutlich im Vergleich zu in ANSI C geschriebenen Programmen zu reduzieren. Dies wird durch die starke Anpassung bzw. Optimierung auf die C67x-Architektur und auf den Assemblerbefehlsatz erreicht. Ein weiterer Vorteil besteht darin, dass diese Funktionen bereits ausführlich und ausreichend gegen vergleichbare ANSI-C Funktionen getestet sind.

Die bereitgestellten Funktionen lassen sich in folgende Themengruppen aufteilen:

- Adaptive Filterung
- Korrelation
- FFT
- Filtern und Faltung
- Vektoren und Matrizen
- Sonstiges

Die Funktionen werden sowohl in *single-precision* (float) als auch in *double-precision* (double) bereitgestellt. In dieser Arbeit finden folgende Funktionen Verwendung:

- **Komplexe Radix2-FFT (DIT):** `void DSPF_sp_cfftr2_dit (float *x, float *w, short n)`
- **Komplexe Matrixmultiplikation:** `void DSPF_sp_mat_mul_cplx (float *x, int r1, int c1, float*y, int c2, float *r)`
- **Indexbestimmung des Maximums:** `int DSPF_sp_maxidx (float *x, int nx)`

Für die Verwendung dieser Funktionen müssen bestimmte Vorgaben erfüllt sein. Dies betrifft vor allem die Anordnung und die Länge der betrachteten Arrays. Für eine genauere Beschreibung dieser Vorgaben und Funktionen sei jedoch an dieser Stelle auf die Dokumentation der Bibliothek [34] verwiesen.

## 2.6 Komplexwertige Zahlen

Komplexe Zahlen lassen sich in der Programmiersprache *C* nicht direkt abbilden. Allerdings lassen sich komplexe Zahlen durch die Kombinanten zweier reeller Zahlen darstellen. Entweder handelt es sich um Betrag und Phase oder um Real- und Imaginärteil. Beide Darstellungsarten haben Vor- und Nachteile. In der *TI C67x DSPLib* (s. Abschnitt 2.5) werden die komplexen Zahlen in der Real- und Imaginärteil-Darstellung betrachtet. Um unnötige Konvertierungen zu vermeiden, wird dieses Format generell verwendet.

Für die Handhabung bietet es sich an, dazu eine Struktur `Complex` anzulegen, die zwei `float`-Variablen (Real- und Imaginärteil) enthält.

## 2.7 Vorgehensweise

Trotz deutlicher Steigerung der Rechenkapazitäten von Prozessoren, ist die Echtzeitfähigkeit von Algorithmen auch heutzutage nicht garantiert. Insbesondere bei der Implementierung komplexer Algorithmen ist mit einem gewissen Mehraufwand für die Echtzeitfähigkeit zu rechnen. Um ein echtzeitfähiges System zu entwickeln bieten sich prinzipiell drei Vorgehensweisen an:

1. Der erste Ansatz zielt auf die Erhöhung der Rechenkapazität. Diese kann durch die Verwendung von schnelleren Prozessoren oder - falls möglich - durch eine parallele Verarbeitung auf verschiedenen Prozessoren erreicht werden. Allerdings ist dieser Ansatz zumeist mit einer deutlichen Steigerung der Kosten verbunden.
2. Der zweite Ansatz besteht darin, die Rechenkapazitäten einer gegebenen Hardware optimal auszunutzen. Dies ist begrenzt durch entsprechende *C*-Programmierung und Compiler-Einstellungen möglich. Es ist jedoch eine deutliche Steigerung bei der Verwendung der Assembler-Funktionen zu erwarten. Dieser Ansatz steigert die Entwicklungszeit deutlich. Auch die Übertragung auf andere Hardware ist bei Verwendung von Assembler-Funktionen nahezu unmöglich.
3. Ein weiterer Ansatz besteht darin, den verwendeten Algorithmus auf die notwendigen Schritte zu reduzieren. Dies ist vor allem möglich, falls die Algorithmen für entsprechende Anwendungsfälle lediglich theoretisch betrachtet worden sind. Durch Fehler und Effekte, die lediglich in der praktischen Anwendung auftreten, können Schritte des ursprünglichen Algorithmus unnötig werden.

Alle drei Vorgehensweisen sind lediglich als Ansätze zu verstehen. Sie garantieren keine Echtzeitfähigkeit. Auch sind die Grenzen dieser Ansätze fließend.

In dieser Arbeit wird der zweite und der dritte Ansatz verfolgt, um die Echtzeitfähigkeit ohne eine Steigerung der Kosten zu erzielen.

## 3 Theoretische Grundlagen

In diesem Kapitel werden die grundlegenden Ansätze und Verfahren zur Lokalisierung von Quellen mittels Arraystrukturen dargelegt. Dabei wird zunächst die Verwendung eines in der angelsächsischen Literatur bezeichneten *Uniform Linear Array* (ULA) betrachtet. Dieses ist dadurch charakterisiert, dass die Mikrofone auf einer Geraden in gleichmäßigen Abständen verteilt sind.

### 3.1 Signalmodell

Das verwendete Signalmodell ist von grundlegender Bedeutung und soll zunächst hergeleitet werden. Dazu wird sich zu Beginn auf die Betrachtung einer Quelle beschränkt. Anhand dieser Quelle wird der Ansatz des Signalmodells definiert und entsprechende Bedingungen, die die Signale erfüllen müssen, aufgestellt. Anschließend wird das Modell auf mehrere Quellen in Matrixnotation verallgemeinert angegeben.

#### 3.1.1 Ansatz

Der prinzipielle Aufbau sowie die wesentlichen Zusammenhänge bei der Verwendung eines ULAs im reflexionsfreien Raum bei der Betrachtung einer Quelle sind in Bild 3.1 dargestellt.

Eine Schallwelle wird von einer Quelle in alle Richtungen ausgesendet. Diese Welle breitet sich omnidirektional aus. Erfolgt die Messung des Wellensignals jedoch in ausreichender Entfernung, so kann näherungsweise eine parallel verlaufende Wellenfront angenommen werden. Ist dies der Fall, befindet man sich im so genannten Fernfeld. Anderenfalls handelt es sich um das Nahfeld. Dabei muss näherungsweise entsprechend [32, S. 221ff] folgende Bedingung erfüllt sein:

$$d_{s-a} \gg \frac{d_{array}^2}{v} f \quad (3.1)$$

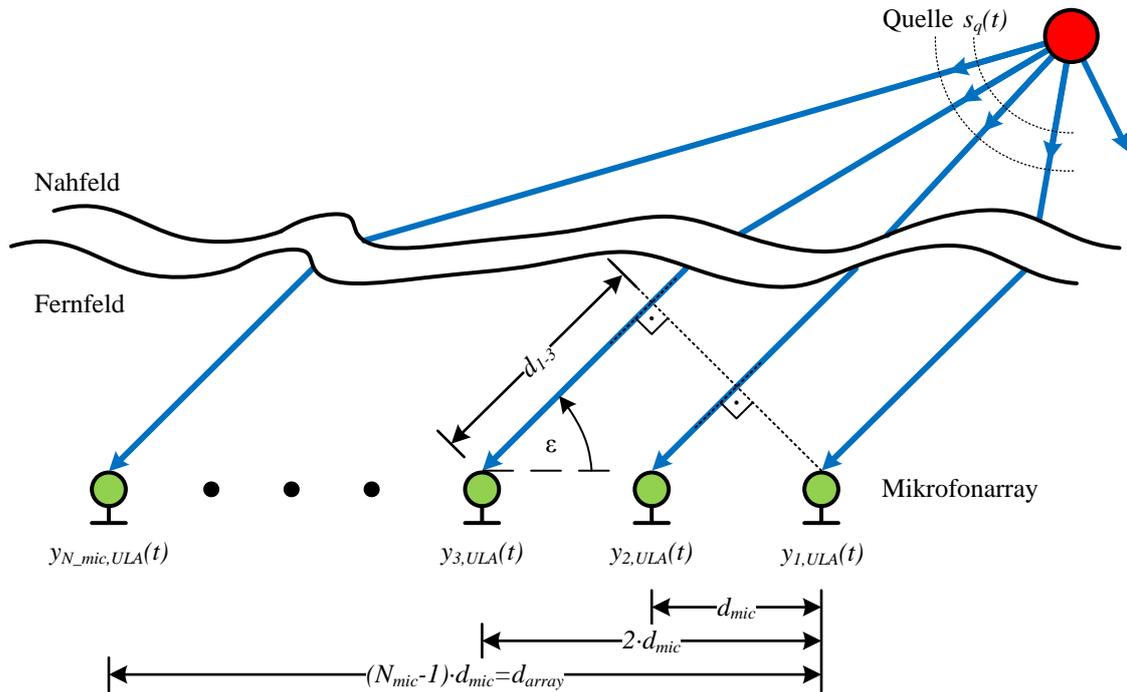


Bild 3.1: Uniform Linear Array

Dabei ist  $d_{s-a}$  der Abstand zwischen den Mittelpunkten des Mikrofonarrays und der Quelle,  $d_{array}$  die maximale Ausdehnung des Mikrofonarrays,  $v$  die Ausbreitungsgeschwindigkeit<sup>2</sup> und  $f$  die Frequenz der Schallwelle.

Die ausgesendete Welle wird zeitlich versetzt durch Mikrofone gemessen. Diese zeitliche Differenz wird als Laufzeitdifferenz bezeichnet. Sie lässt sich aus der gegebenen Geometrie des Mikrofonarrays bestimmen. Bezieht man sich auf das erste Mikrofon, so gilt für die Laufzeitdifferenz des  $i$ -ten Mikrofon in Bild 3.1:

$$\tau_i = \frac{d_{1-i}}{v} \quad (3.2)$$

$$= \frac{(i-1)d_{mic} \cos(\epsilon)}{v} \quad \text{mit } i = 1, 2, \dots, N_{mic}; \quad 0 \leq \epsilon \leq \pi \quad (3.3)$$

Dabei gibt  $N_{mic}$  die Anzahl der vorhandenen Mikrofone,  $\epsilon$  den Einfallswinkel,  $d_{mic}$  den Abstand zwischen zwei Mikrofonen und  $d_{1-i}$  die noch zurückzulegende Strecke, ab Zeitpunkt der Messung des Signals am ersten Mikrofon, an.

Mit diesem Wissen lässt sich das gemessene Signal  $y_{i,ULA}(t)$  am  $i$ -ten Mikrofon wie folgt

<sup>2</sup>Bei der Ausbreitungsgeschwindigkeit handelt es sich bei Audiosignalen um die Schallgeschwindigkeit. Diese beträgt in trockener Luft bei 20 °C 343 m/s.

beschreiben:

$$y_{i,ULA}(t) = x(t - \tau_i) + n_i(t) \quad (3.4)$$

Dabei beschreibt  $x(t - \tau_i)$  das Wellensignal der Quelle am  $i$ -ten Mikrofon, welches von weißem, mittelwertfreiem gaußschen Rauschen  $n_i(t)$  additiv überlagert gemessen wird. Die Signale  $x(t - \tau_i)$  sind lediglich zeitverzögert zum Signal  $x(t)$ , welches am ersten Mikrofon gemessen wird. Das Signal  $x(t)$  kann dabei im reflexionsfreien Raum als gedämpftes und verzögertes Signal  $s_q(t)$  angenommen werden:

$$x(t) = \alpha_D s_q(t - \tau_s) \quad (3.5)$$

Dabei beschreibt  $\alpha_D$  den Dämpfungsfaktor und  $\tau_s$  die Zeitverzögerung zwischen Quelle und erstem Mikrofon.

Nach erfolgreichem Empfang des Signals erfolgt die Analog-Digital-Umsetzung. Mit der Abtastfrequenz zu  $f_s = 1/T$  gilt:

$$y_{i,ULA}[n] = y_{i,ULA}(nT) \quad (3.6)$$

$$= x(nT - \tau_i) + n_i[n] \quad (3.7)$$

### 3.1.2 Schmalbandigkeit

Im vorherigen Abschnitt ist ein Ansatz für das Signalmodell auf Basis von Einfallswinkel und resultierender Laufzeitdifferenz hergeleitet worden. Eine Modellierung der Zeitverzögerung durch eventuelle Annahmen mittels einer Multiplikation des Quellensignals mit einem entsprechenden Faktor statt der Verschiebung des Arguments des Quellensignals wäre jedoch von Vorteil.

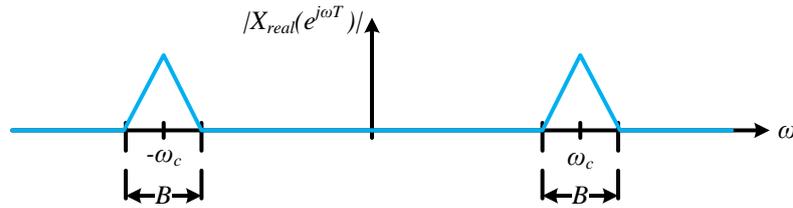
Hierzu sei ein bandbegrenzttes Signal mit einer kleinen Bandbreite  $B$  relativ zur Mittenfrequenz  $f_c = \omega_c/2\pi$  betrachtet. Es gilt:

$$\omega_c - \frac{B}{2} \leq \omega \leq \omega_c + \frac{B}{2} \quad (3.8)$$

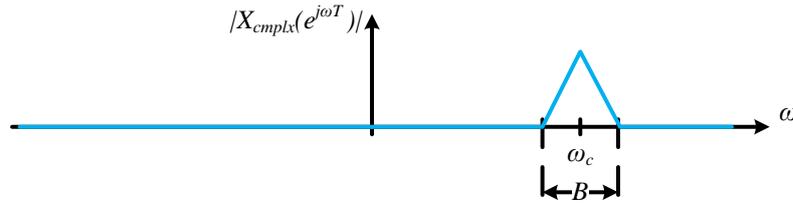
Das Spektrum eines entsprechenden reellwertigen Zeitsignals ist in Bild 3.2(a) dargestellt.

Mit Hilfe der inversen zeitdiskreten Fouriertransformation ergibt sich das entsprechende Zeitsignal  $x_{real}[n]$  zu, wobei  $\omega_s = 2\pi f_s$  gilt:

$$x_{real}[n] = \frac{1}{\omega_s} \int_{-\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} X_{real}(e^{j\omega T}) e^{j\omega n T} d\omega \quad (3.9)$$



(a) Amplitudengang eines reellwertigen Zeitsignals



(b) Amplitudengang eines komplexwertigen Zeitsignals

**Bild 3.2:** Amplitudengänge bandbegrenzter Signale

Aufgrund dessen, dass das Signal jedoch bandbegrenzt ist, können die Integralgrenzen angepasst und das Integral in zwei Summen aufgespaltet werden.

$$x_{real}[n] = \frac{1}{\omega_s} \int_{-\omega_c - \frac{B}{2}}^{-\omega_c + \frac{B}{2}} X_{real}(e^{j\omega T}) e^{j\omega n T} d\omega + \frac{1}{\omega_s} \int_{\omega_c - \frac{B}{2}}^{\omega_c + \frac{B}{2}} X_{real}(e^{j\omega T}) e^{j\omega n T} d\omega \quad (3.10)$$

Im Wesentlichen handelt es sich somit um Integrale, die jeweils um die negative und positive Mittenfrequenz gebildet werden. Nachteilig ist hierbei, dass immer zwei Frequenzen betrachtet werden müssen. Dementsprechend wird ein komplexwertiges, bandbegrenztetes Zeitsignal benutzt. Ein entsprechender Amplitudengang ist in Bild 3.2(b) dargestellt. Für das komplexwertige Zeitsignal folgt:

$$x_{complex}[n] = \frac{1}{\omega_s} \int_{\omega_c - \frac{B}{2}}^{\omega_c + \frac{B}{2}} X_{complex}(e^{j\omega T}) e^{j\omega n T} d\omega \quad (3.11)$$

Verzögert man  $x_{complex}[n] = x_{complex}(nT)$  um  $\tau$ , so lässt sich das Spektrum des verzögerten Signals  $x_{complex}(nT - \tau)$  wie folgt bestimmen:

$$x_{complex}(nT - \tau) \circ \bullet X_{complex}(e^{j\omega T}) \cdot e^{-j\omega\tau} \quad (3.12)$$

Betrachtet man für  $x_{complex}(nT - \tau)$  Gleichung 3.11, so folgt:

$$x_{complex}(nT - \tau) = \frac{1}{\omega_s} \int_{\omega_c - \frac{B}{2}}^{\omega_c + \frac{B}{2}} X_{complex}(e^{j\omega T}) e^{-j\omega\tau} e^{j\omega n T} d\omega \quad (3.13)$$

Für den Fall, dass  $B \ll \omega_s$  kann der Faktor  $e^{-j\omega\tau}$  näherungsweise als Konstante aus dem Integral gezogen werden:

$$x_{\text{cmplx}}(nT - \tau) \approx e^{-j\omega\tau} \frac{1}{\omega_s} \int_{\omega_c - \frac{B}{2}}^{\omega_c + \frac{B}{2}} X_{\text{cmplx}}(e^{j\omega T}) e^{j\omega nT} d\omega \quad (3.14)$$

$$\approx e^{-j\omega\tau} x_{\text{cmplx}}(nT) \quad (3.15)$$

Diese Näherung stellt den gewünschten Sachverhalt dar. Durch die Annahme eines komplexwertigen, bandbegrenzten Zeitsignals kann die Verschiebung des Argumentes als Darstellung der Laufzeitdifferenz durch Multiplikation einer entsprechenden Phase geschehen. Gleichung 3.7 kann folglich in folgende Form gebracht werden:

$$y_{i,ULA}[n] = x[n]e^{-j\omega\tau_i} + n_i[n] \quad (3.16)$$

Da die Laufzeitdifferenz aufgrund von Gleichung 3.3 im direkten Zusammenhang zum Einfallswinkel steht, gilt:

$$y_{i,ULA}[n] = a_i(\epsilon) \cdot x[n] + n_i[n] \quad (3.17)$$

$$a_i(\epsilon) = e^{-j\omega_c \tau_i} \quad (3.18)$$

$$= e^{-j\omega_c \frac{(i-1)d_{\text{mic}} \cos(\epsilon)}{v}} \quad (3.19)$$

In der Literatur wird zumeist noch das Modell der komplexen Einhüllenden verwendet. Dies unterscheidet sich dadurch, dass das hier verwendete komplexwertige, schmalbandige Signal in das Basisband verschoben wird. Dies hat in der Handhabung gewisse Vorteile, allerdings spielt die Unterscheidung für die nachfolgenden Algorithmen keine elementare Rolle.

### 3.1.3 Matrixnotation

Die bisherigen Betrachtungen begrenzen sich auf eine Quelle. Das entwickelte Signalmodell soll in diesem Abschnitt auf mehrere Quellen erweitert und in Matrixnotation angegeben werden.

Die Verwendung von Matrizen bietet zumeist eine kompakte Darstellung relativ komplexer Zusammenhänge. Fasst man alle Signale des Mikrophonarrays in einem Vektor zusammen,

so gilt:

$$\mathbf{y}_{ULA}[n] = \mathbf{A}_{ULA}(\boldsymbol{\varepsilon}) \cdot x[n] + \mathbf{n}[n]$$

$$\begin{bmatrix} y_{1,ULA}[n] \\ y_{2,ULA}[n] \\ \vdots \\ y_{N_{mic},ULA}[n] \end{bmatrix} = \begin{bmatrix} a_1(\boldsymbol{\varepsilon}) \\ a_2(\boldsymbol{\varepsilon}) \\ \vdots \\ a_{N_{mic}}(\boldsymbol{\varepsilon}) \end{bmatrix} \cdot x[n] + \begin{bmatrix} n_1[n] \\ n_2[n] \\ \vdots \\ n_{N_{mic}}[n] \end{bmatrix} \quad (3.20)$$

Für die Erweiterung des Modells auf  $N_S$  Quellen sei hier angenommen, dass sich die Signale der einzelnen Quellen additiv überlagern.

$$\mathbf{y}_{ULA}[n] = \mathbf{A}_{ULA} \cdot \mathbf{x}[n] + \mathbf{n}[n]$$

$$\begin{bmatrix} y_{1,ULA}[n] \\ y_{2,ULA}[n] \\ \vdots \\ y_{N_{mic},ULA}[n] \end{bmatrix} = \begin{bmatrix} a_1(\boldsymbol{\varepsilon}_1) & a_1(\boldsymbol{\varepsilon}_2) & \cdots & a_1(\boldsymbol{\varepsilon}_{N_S}) \\ a_2(\boldsymbol{\varepsilon}_1) & a_2(\boldsymbol{\varepsilon}_2) & \cdots & a_2(\boldsymbol{\varepsilon}_{N_S}) \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_{mic}}(\boldsymbol{\varepsilon}_1) & a_{N_{mic}}(\boldsymbol{\varepsilon}_2) & \cdots & a_{N_{mic}}(\boldsymbol{\varepsilon}_{N_S}) \end{bmatrix} \cdot \begin{bmatrix} x_1[n] \\ x_2[n] \\ \vdots \\ x_{N_S}[n] \end{bmatrix} + \begin{bmatrix} n_1[n] \\ n_2[n] \\ \vdots \\ n_{N_{mic}}[n] \end{bmatrix} \quad (3.21)$$

Dabei stellt das Signal  $x_i[n]$  das gemessene Signal am ersten Mikrofon der  $i$ -ten Quelle dar. Die Matrix  $\mathbf{A}_{ULA} \in \mathbb{C}^{N_S}$  stellt die sogenannte Richtungsmatrix dar und  $\boldsymbol{\varepsilon}_i$  beschreibt den Einfallswinkel der  $i$ -ten Quelle.

Um die besondere Struktur der Richtungsmatrix  $\mathbf{A}_{ULA}$  darzulegen, sei Gleichung 3.19 nochmal berücksichtigt, wobei  $\eta(\boldsymbol{\varepsilon}_i) = \omega_c d_{mic} \cos(\boldsymbol{\varepsilon}_i)^{1/v}$  gilt:

$$\mathbf{A}_{ULA} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\eta(\boldsymbol{\varepsilon}_1)} & e^{-j\eta(\boldsymbol{\varepsilon}_2)} & \cdots & e^{-j\eta(\boldsymbol{\varepsilon}_{N_S})} \\ e^{-j2\eta(\boldsymbol{\varepsilon}_1)} & e^{-j2\eta(\boldsymbol{\varepsilon}_2)} & \cdots & e^{-j2\eta(\boldsymbol{\varepsilon}_{N_S})} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j(N_{mic}-1)\eta(\boldsymbol{\varepsilon}_1)} & e^{-j(N_{mic}-1)\eta(\boldsymbol{\varepsilon}_2)} & \cdots & e^{-j(N_{mic}-1)\eta(\boldsymbol{\varepsilon}_{N_S})} \end{bmatrix} \quad (3.22)$$

Gut zu erkennen ist, dass es sich bei der Richtungsmatrix um eine transponierte Vandermonde-Matrix handeln kann. Dies ist der Fall, falls alle Quellen unterschiedliche Einfallswinkel aufweisen. Mit der Annahme  $N_{mic} > N_S$  hat die Richtungsmatrix folglich den Rang  $N_S$ .

### 3.2 Räumlicher Alias-Effekt

Bei der Betrachtung periodischer Signale sind die Signalwerte im Sinne des Abtastzeitpunkts mehrdeutig. Diese Mehrdeutigkeit ruft bei der Abtastung von Signalen den Alias-Effekt hervor. Bei der Betrachtung von zeitlich verschobenen Signalen an Mikrofonen führt die Mehrdeutigkeit zum räumlichen Alias-Effekt. Signale mit der Phase von  $\alpha$  weisen denselben Signalwert wie Signale mit der Phase  $2\pi - \alpha$  auf. Um dies zu vermeiden, soll keine größere Phasendifferenz als  $\pi$  auftreten. Für ein Signal mit der maximalen Frequenz von  $f_{max}$  gilt somit die Bedingung:

$$2\pi f_{max} < \pi. \quad (3.23)$$

Setzt man die Gleichung 3.3 ein und betrachtet lediglich die ersten beiden Mikrofone, so folgt mit elementaren Umformungen:

$$d_{mic} < \frac{v}{2f_{max} \cos \varepsilon} \quad (3.24)$$

Die größte Phasenverschiebung ergibt sich bei einem Winkel von  $\varepsilon = 0^\circ$  und der Wellenlänge  $\lambda_{max} = v/f_{max}$ . Es folgt:

$$d_{mic} < \frac{\lambda_{max}}{2} \quad (3.25)$$

Dies stellt die Bedingung für das Vermeiden des räumlichen Alias-Effekt dar. Der Abstand  $d_{mic}$  sollte somit kleiner als die halbe Wellenlänge sein. Für  $f_{max} = 3,4 \text{ kHz}$  ergibt sich beispielsweise ein Wert von  $d_{mic} < 5 \text{ cm}$ .

### 3.3 Autokorrelationsmatrix

Um eine Aussage über die Ähnlichkeit der gemessenen Signale an den Mikrofonen zu erhalten, wird die Autokorrelationsmatrix von  $\mathbf{y}_{ULA}[n]$  gebildet. Diese stimmt mit der Autokovarianzmatrix überein, falls  $\mathbf{y}_{ULA}[n]$  mittelwertfrei ist. Dies wird hier angenommen. Die Autokorrelationsmatrix ist, unter der Annahme der Ergodizität, wie folgt definiert:

$$\mathbf{R}_{yy,ULA} = E \left\{ \mathbf{y}_{ULA}[n] \cdot \mathbf{y}_{ULA}[n]^\dagger \right\} \quad (3.26)$$

Setzt man die Beziehung 3.21 ein und multipliziert die Klammerausdrücke aus, folgt:

$$\mathbf{R}_{yy,ULA} = E \left\{ (\mathbf{A}_{ULA} \cdot \mathbf{x}[n] + \mathbf{n}[n]) \cdot (\mathbf{A}_{ULA} \cdot \mathbf{x}[n] + \mathbf{n}[n])^\dagger \right\} \quad (3.27)$$

$$= E \left\{ \mathbf{A}_{ULA} \cdot \mathbf{x}[n] \cdot \mathbf{x}[n]^\dagger \mathbf{A}_{ULA}^\dagger + \mathbf{n}[n] \cdot \mathbf{n}[n]^\dagger + \mathbf{n} \cdot \mathbf{x}[n]^\dagger \cdot \mathbf{A}_{ULA}^\dagger + \mathbf{a}_{ULA} \cdot \mathbf{x}[n] \cdot \mathbf{n}[n]^\dagger \right\} \quad (3.28)$$

Mit der Annahme, dass das Rauschen und die Signale unkorreliert sind, folgt:

$$\mathbf{R}_{yy,ULA} = E \left\{ \mathbf{A}_{ULA} \cdot \mathbf{x}[n] \cdot \mathbf{x}[n]^\dagger \mathbf{A}_{ULA}^\dagger + \mathbf{n}[n] \cdot \mathbf{n}[n]^\dagger \right\} \quad (3.29)$$

Aufgrund der Linearität des Erwartungswertoperators gilt:

$$\mathbf{R}_{yy,ULA} = E \left\{ \mathbf{A}_{ULA} \cdot \mathbf{x}[n] \cdot \mathbf{x}[n]^\dagger \mathbf{A}_{ULA}^\dagger \right\} + E \left\{ \mathbf{n}[n] \cdot \mathbf{n}[n]^\dagger \right\} \quad (3.30)$$

Da für unkorreliertes weißes Rauschen die Autokorrelationsmatrix bekannt ist und  $\mathbf{a}_{ULA}$  im Sinne des Erwartungswertes eine Konstante darstellt, folgt:

$$\mathbf{R}_{yy,ULA} = \mathbf{A}_{ULA} \cdot E \left\{ \mathbf{x}[n] \cdot \mathbf{x}[n]^\dagger \right\} \cdot \mathbf{A}_{ULA}^\dagger + \sigma^2 \cdot \mathbf{I}_{N_{mic}} \quad (3.31)$$

$$= \mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger + \sigma^2 \cdot \mathbf{I}_{N_{mic}} \quad (3.32)$$

Dabei stellt  $\sigma^2$  die Leistung des weißen Rauschens dar und  $\mathbf{R}_{xx} \in \mathbb{C}^{N_s \times N_s}$  beschreibt die Autokorrelationsmatrix der verschiedenen Quellen am ersten Mikrofon.

Die Autokorrelationsmatrix weist dabei folgende Eigenschaften auf. Die Autokorrelationsmatrix  $\mathbf{R}_{ww}$  des Signalvektors  $\mathbf{w}[k]$  der Dimension  $N \times N$ ...

- ist per Definition hermitisch, da  $\mathbf{R}_{ww} = \mathbf{R}_{ww}^\dagger$
- ist per Definition positiv semidefinit, da  $\mathbf{R}_{ww}$  hermitisch ist und für alle  $\mathbf{p} \in \mathbb{C}^N, \mathbf{p} \neq \mathbf{0}$  gilt:

$$\mathbf{p}^\dagger \mathbf{R}_{ww} \mathbf{p} \geq 0 \quad (3.33)$$

$$\mathbf{p}^\dagger E \left\{ \mathbf{w}[n] \cdot \mathbf{w}[n]^\dagger \right\} \mathbf{p} \geq 0 \quad (3.34)$$

$$E \left\{ \mathbf{p}^\dagger \mathbf{w}[n] \cdot \mathbf{w}[n]^\dagger \mathbf{p} \right\} \geq 0 \quad (3.35)$$

$$E \left\{ \left( \mathbf{p}^\dagger \mathbf{w}[n] \right) \cdot \left( \mathbf{p}^\dagger \mathbf{w}[n] \right)^\dagger \right\} \geq 0 \quad (3.36)$$

Mit  $\mathbf{p} = [a, b, \dots, c]^T$  und  $\mathbf{w}[n] = [w_1[n], w_2[n], \dots, w_N[n]]^T$  folgt:

$$E \left\{ |a|^2 |w_1[n]|^2 + |b|^2 |w_2[n]|^2 + \dots + |c|^2 |w_N[n]|^2 \right\} \geq 0 \quad (3.37)$$

Da Beträge per Definition größer gleich Null sind, ist die gestellte Bedingung stets erfüllt.

- ist eine Diagonalmatrix und hat den vollen Rang  $N$ , falls die Signalvektorelemente unkorreliert sind.
- ist keine Diagonalmatrix und kann den vollen Rang  $N$  haben, falls die Signalvektorelemente nahezu unkorreliert sind.

Nachfolgend wird von unkorrelierten bzw. nahezu unkorrelierten Quellen ausgegangen.

Da jedoch nur endliche Daten vorliegen, kann die Autokorrelationsmatrix lediglich geschätzt werden:

$$\hat{\mathbf{R}}_{yy,ULA} = \frac{1}{N} \mathbf{y}_{ULA}[n] \mathbf{y}_{ULA}[n]^\dagger \approx \mathbf{R}_{yy,ULA} \quad (3.38)$$

Dabei stellt  $N$  die Anzahl der zur Verfügung stehenden Werte und  $\mathbf{y}_{ULA}[n]$  die entsprechende Wertefolge dar.

## 3.4 Unterraumbasierte Verfahren

In diesem Abschnitt werden die grundlegenden unterraumbasierten Verfahren vorgestellt. Dabei wird sowohl auf die Annahmen, auf denen die Verfahren beruhen, als auch auf die Schritte, die zur Berechnung notwendig sind, eingegangen.

### 3.4.1 MUSIC

Das Verfahren MUSIC ist 1979 von Ralph O. Schmidt vorgestellt worden. Er beschreibt das Verfahren unter anderem in [30]. An diese Veröffentlichung lehnen sich die nachfolgenden Ausführungen an.

Mit der Annahme, dass die Matrix  $\mathbf{R}_{xx}$  den Rang  $N_S$  (s. Abschnitt 3.3) hat und die Matrix  $\mathbf{A}_{ULA}$  mit der Bedingung  $N_S \leq N_{mic}$  (s. Abschnitt 3.1.3) den Rang  $N_S$  hat, hat die Matrix  $\mathbf{R}_{yy,ULA}$  maximal den Rang  $N_S^3$ . Nimmt man an, dass  $N_S < N_{mic}$ , muss die Matrix  $\mathbf{A}_{ULA}$

<sup>3</sup>Für das Produkt von Matrizen gilt für den Rang  $\text{rang}(\mathbf{A}_{ULA} \cdot \mathbf{B}) \leq \min \{ \text{rang}(\mathbf{A}_{ULA}), \text{rang}(\mathbf{B}) \}$

$\mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger$  singular sein, da der Rang der Matrix kleiner als Spalten- und Zeilenanzahl  $N_{mic}$  ist. Dies bedeutet, dass gilt:

$$\det(\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger) = 0 \quad (3.39)$$

$$\det(\mathbf{R}_{yy,ULA} - \sigma^2 \cdot \mathbf{I}_{N_{mic} \times N_{mic}}) = 0 \quad (3.40)$$

Gleichung 3.40 zeigt die charakteristische Gleichung der Matrix  $\mathbf{R}_{yy,ULA}$ , wodurch  $\sigma^2$  Eigenwert der Matrix  $\mathbf{R}_{yy,ULA}$  sein muss (s. Abschnitt A.2).

Ausgehend von der Eigenwertgleichung

$$\mathbf{R}_{yy,ULA} \cdot \mathbf{e}_i = \lambda_i \cdot \mathbf{e}_i \quad (3.41)$$

lässt sich folgende Beziehung mit Gleichung 3.32 herleiten:

$$(\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger + \sigma^2 \cdot \mathbf{I}_{N_{mic} \times N_{mic}}) \cdot \mathbf{e}_i = \lambda_i \cdot \mathbf{e}_i \quad (3.42)$$

$$\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{e}_i = (\lambda_i - \sigma^2) \cdot \mathbf{e}_i \quad (3.43)$$

$$= \lambda_{x,i} \cdot \mathbf{e}_i \quad (3.44)$$

Aus dieser Beziehung und der Gleichung 3.40 lassen sich die Charakteristika der Eigenwerte sowie die Beziehung der aufgespannten Vektorräume der Eigenvektoren herleiten. Unter der Annahme, dass  $\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger$  den Rang  $N_S$  besitzt, sind  $N_{mic} - N_S$  Eigenwerte von  $\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger$  gleich null. Aufgrund der positiven semidefiniten Eigenschaft sind demzufolge alle anderen Eigenwerte größer null. Somit folgt aus  $\lambda_i = (\lambda_{x,i} + \sigma^2)$  folgende Struktur der Eigenwerte, falls man diese der Größe nach ordnet:

$$\lambda_1 \geq \lambda_2 \geq \dots \lambda_{N_S} > \sigma^2 = \lambda_{N_S+1} = \lambda_{N_S+2} = \dots = \lambda_{N_{mic}} \quad (3.45)$$

Bezieht man diesen Zusammenhang auf Gleichung 3.43, folgt:

$$\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{e}_i = \begin{cases} (\lambda_i - \sigma^2) \cdot \mathbf{e}_i & , \text{ wenn } 1 < i \leq N_S \\ 0 & , \text{ wenn } N_S < i \leq N_{mic} \end{cases} \quad (3.46)$$

Folglich erscheint die Gruppierung der Eigenvektoren sinnvoll. Die ersten  $N_S$  Eigenvektoren der Matrix  $\mathbf{R}_{yy,ULA}$  werden dem Signalunterraum und die weiteren Eigenvektoren dem Rauschunterraum zugeordnet.

Mit der Gruppierung der Eigenvektoren bzw. Eigenwerte entsprechend der Ausführungen und der unitären Eigenschaft der Eigenvektoren einer hermiteschen Matrix folgt für Gleichung

3.41 in Matrixschreibweise

$$\mathbf{R}_{yy,ULA} = \mathbf{E} \cdot \Lambda \cdot \mathbf{E}^\dagger \quad (3.47)$$

$$= \begin{bmatrix} \mathbf{E}_x & \vdots & \mathbf{E}_n \end{bmatrix} \cdot \begin{bmatrix} \Lambda_x & \mathbf{0}_{N_S \times (N_{mic} - N_S)} \\ \mathbf{0}_{(N_{mic} - N_S) \times N_S} & \Lambda_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{E}_x & \vdots & \mathbf{E}_n \end{bmatrix}^\dagger \quad (3.48)$$

$$= \mathbf{E}_x \Lambda_x \mathbf{E}_x^\dagger + \mathbf{E}_n \Lambda_n \mathbf{E}_n^\dagger \quad (3.49)$$

wobei

$$\Lambda_x = \text{diag}(\lambda_1, \dots, \lambda_{N_S}) \quad (3.50)$$

$$\Lambda_n = \text{diag}(\sigma^2, \dots, \sigma^2) \quad (3.51)$$

gilt. Mit Gleichung 3.46 folgt bei Betrachtung des Rauschunterraumes:

$$\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{E}_n = 0 \quad (3.52)$$

Mit der Annahme, dass  $\mathbf{A}_{ULA}$  und  $\mathbf{R}_{ss}$  invertierbar sind, gilt:

$$\mathbf{A}_{ULA}^\dagger \cdot \mathbf{E}_n = 0 \quad (3.53)$$

Somit sind die Eigenvektoren des Rauschunterraumes orthogonal zu den Spaltenvektoren der Richtungsmatrix.

Die Betrachtung des Signalunterraumes in Gleichung 3.46 zeigt hingegen, falls alle Eigenwerte des Signalunterraumes unterschiedlich sind, dass

$$\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{E}_x = \mathbf{E}_x \cdot (\Lambda_x - \sigma^2 \mathbf{I}_{N_S}) \quad (3.54)$$

$$\mathbf{A}_{ULA} \cdot \mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{E}_x (\Lambda_x - \sigma^2 \mathbf{I}_{N_S})^{-1} = \mathbf{E}_x \quad (3.55)$$

gilt. Da die Matrix  $\mathbf{R}_{xx} \cdot \mathbf{A}_{ULA}^\dagger \cdot \mathbf{E}_x (\Lambda_x - \sigma^2 \mathbf{I}_{N_S})^{-1}$  vollen Rang hat und damit regulär ist, spannen die Eigenvektoren des Signalunterraumes und die Spaltenvektoren der Richtungsmatrix die gleiche lineare Hülle auf. Zusammengefasst gelten folgende Beziehungen zwischen Richtungsmatrix, Signal- und Rauschunterraum:

$$\text{span}\{\mathbf{A}_{ULA}\} \perp \text{span}\{\mathbf{E}_n\} \quad (3.56)$$

$$\text{span}\{\mathbf{A}_{ULA}\} = \text{span}\{\mathbf{E}_x\} \quad (3.57)$$

MUSIC nutzt diese Orthogonalität zwischen Rauschsignalunterraum und Richtungsunterraum aus. Dazu wird die Autokorrelationsmatrix  $\hat{\mathbf{R}}_{yy}$  geschätzt. Anschließend wird eine Eigenwertzerlegung durchgeführt. Anhand dieser werden die Eigenvektoren in Signaleigenvektor und Rauscheigenvektor entsprechend der vorherigen Überlegungen aufgeteilt. Dabei hilft eine Sortierung der Eigenwerte und der Eigenvektoren sowie die Annahme, dass

$N_{mic} - N_S$  Eigenwerte näherungsweise übereinstimmen. Anschließend wird die Menge der Richtungsvektoren und die Menge der Eigenvektoren des Rauschens auf Orthogonalität überprüft. Da jedoch keine exakte Übereinstimmung mit dem Modell zu erwarten ist, wird sich der Euklidischen Norm bedient und das Reziprok gebildet. Das Ergebnis wird als Pseudospektrum bezeichnet. Dieses definiert sich wie folgt:

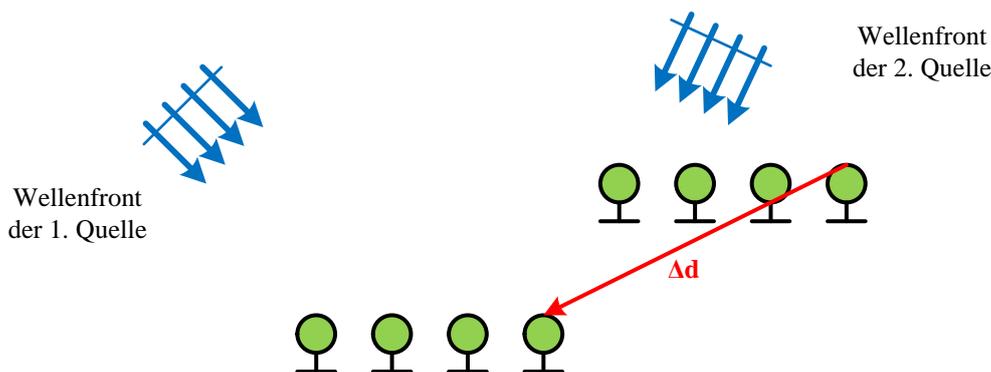
$$\hat{S}(\varepsilon) = \frac{1}{\mathbf{A}_{ULA}(\varepsilon)^\dagger \cdot \hat{\mathbf{E}}_n \cdot \hat{\mathbf{E}}_n^\dagger \cdot \mathbf{A}_{ULA}(\varepsilon)} \quad (3.58)$$

Falls ein Signal aus einer entsprechenden Richtung kommt, müssten sich deutliche lokale Maxima im Pseudospektrum ergeben.

### 3.4.2 ESPRIT

Das MUSIC-Verfahren weist im letzten Verarbeitungsschritt eine Maximasuche auf. Diese ist numerisch als aufwendig einzustufen, da stets der zu betrachtende Wert neu berechnet werden muss. Bei der Verwendung des ESPRIT-Algorithmus muss diese Suche nicht ausgeführt werden. Das Verfahren ist erstmalig [27] in der hier vorgestellten Version (TLS-ESPRIT) veröffentlicht worden. Die nachfolgenden Ausführungen orientieren sich an dieser Veröffentlichung:

ESPRIT benötigt zwei Arrays gleicher Struktur, die zueinander verschoben sind. Dies kann zum Beispiel auch durch die Aufteilung der Arrayelemente in zwei Gruppen erfolgen. In Bild 3.3 ist ein mögliches Szenario abgebildet.



**Bild 3.3:** Prinzip des *ESPRIT*-Algorithmus

Stellt man für beide Arraystrukturen die entsprechenden Signalmodelle auf und berücksich-

tigt, dass sich diese lediglich durch den geometrischen Versatz  $\Delta d$  unterscheiden, folgt:

$$\mathbf{y}_{G1}[n] = \mathbf{A}_{ULA} \cdot \mathbf{x}[n] + \mathbf{n}_{G1}[n] \quad (3.59)$$

$$\mathbf{y}_{G2}[n] = \mathbf{A}_{ULA} \cdot \mathbf{\Xi} \cdot \mathbf{x}[n] + \mathbf{n}_{G2}[n] \quad (3.60)$$

Unter Berücksichtigung der Schmalbandigkeit gilt für  $\mathbf{\Omega}$ :

$$\mathbf{\Xi} = \text{diag} \left( e^{2\pi f \frac{\Delta d}{v} \sin(\varepsilon_1)}, e^{2\pi f \frac{\Delta d}{v} \sin(\varepsilon_2)}, \dots, e^{2\pi f \frac{\Delta d}{v} \sin(\varepsilon_{N_S})} \right) \quad (3.61)$$

Diese Matrix ist unitär und setzt die beiden Arrays zueinander in Beziehung. Im Komplexen handelt es sich hierbei um eine Skalierungsmatrix. Die Idee von ESPRIT ist es,  $\mathbf{\Xi}$  zu bestimmen und somit auf die entsprechenden Einfallswinkel schließen zu können.

Beide Arraygruppen können in einem Vektor  $\mathbf{z}$  zusammengefasst werden:

$$\mathbf{z}[n] = \begin{bmatrix} \mathbf{y}_{G1}[n] \\ \mathbf{y}_{G2}[n] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{ULA} \\ \mathbf{A}_{ULA} \cdot \mathbf{\Xi} \end{bmatrix} \cdot \mathbf{x}[n] + \begin{bmatrix} \mathbf{n}_{G1}[n] \\ \mathbf{n}_{G2}[n] \end{bmatrix} \quad (3.62)$$

$$= \tilde{\mathbf{A}}_{ULA} \cdot \mathbf{x}[n] + \mathbf{n}_z[n] \quad (3.63)$$

Für die Autokorrelationsmatrix ergibt sich analog zu vorherigen Überlegungen:

$$\mathbf{R}_{zz} = \tilde{\mathbf{A}}_{ULA} \cdot \mathbf{R}_{xx} \cdot \tilde{\mathbf{A}}_{ULA}^\dagger + \sigma^2 \cdot \mathbf{I}_{2N_{mic} \times 2N_{mic}} \quad (3.64)$$

Ebenfalls können die Eigenvektoren in zwei Matrizen aufgeteilt werden, die jeweils den Signalunterraum und den Rauschunterraum aufspannen. Entsprechend den vorherigen Überlegungen folgt auch hier:

$$\text{span} \left\{ \tilde{\mathbf{A}}_{ULA} \right\} = \text{span} \left\{ \mathbf{E}_x \right\} \quad (3.65)$$

Mit der Bedingung  $N_S \leq N_{mic}$  folgt hieraus, dass eine reguläre Matrix  $\mathbf{T} \in \mathbb{C}^{N_S \times N_S}$  existieren muss, sodass gilt

$$\mathbf{E}_s = \tilde{\mathbf{A}}_{ULA} \cdot \mathbf{T}. \quad (3.66)$$

Betrachtet man die einzelnen Arraygruppen für sich, ergibt sich:

$$\mathbf{E}_x = \begin{bmatrix} \mathbf{E}_{x,y_{G1}} \\ \mathbf{E}_{x,y_{G2}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{ULA} \\ \mathbf{A}_{ULA} \cdot \mathbf{\Xi} \end{bmatrix} \cdot \mathbf{T} \quad (3.67)$$

Wobei  $\mathbf{E}_{x,y_{G1}} \in \mathbb{C}^{N_{mic} \times N_S}$  und  $\mathbf{E}_{x,y_{G2}} \in \mathbb{C}^{N_{mic} \times N_S}$  gilt. Hieraus folgt mit 3.65, da  $\mathbf{\Omega}$  lediglich

einen Skalierungsfaktor im Komplexen darstellt:

$$\text{span} \left\{ \tilde{\mathbf{A}}_{ULA} \right\} = \text{span} \left\{ \mathbf{E}_{x,y_{G1}} \right\} = \text{span} \left\{ \mathbf{E}_{x,y_{G2}} \right\} \quad (3.68)$$

Da  $\mathbf{E}_{s,x_{G1}}$  und  $\mathbf{E}_{s,x_{G2}}$  dieselbe lineare Hülle aufspannen, ist der Rang der Matrix

$$\mathbf{E}_{G1G2} = \left[ \mathbf{E}_{x,y_{G1}} \quad \vdots \quad \mathbf{E}_{x,y_{G2}} \right] \quad (3.69)$$

auf  $N_S$  begrenzt. Es existiert somit eine eindeutige Matrix  $\mathbf{F} = [\mathbf{F}_{y_{G1}} \quad \vdots \quad \mathbf{F}_{y_{G2}}]^T \in \mathbb{C}^{2N_S \times N_S}$  die den Nullraum von  $\mathbf{E}_{G1G2}$  aufspannt:

$$0 = \mathbf{E}_{G1G2} \cdot \mathbf{F} = \mathbf{E}_{x,y_{G1}} \cdot \mathbf{F}_{y_{G1}} + \mathbf{E}_{x,y_{G2}} \cdot \mathbf{F}_{y_{G2}} \quad (3.70)$$

Mit der Beziehung aus Gleichung 3.67:

$$0 = \mathbf{A}_{ULA} \mathbf{T} \mathbf{F}_{y_{G1}} + \mathbf{A}_{ULA} \mathbf{\Xi} \mathbf{T} \mathbf{F}_{y_{G2}} \quad (3.71)$$

Mit Annahme, dass  $\mathbf{F}_{y_{G2}}$  invertierbar ist, kann

$$\Gamma_F = -\mathbf{F}_{y_{G1}} \mathbf{F}_{y_{G2}}^{-1} \quad (3.72)$$

definiert werden. Eingesetzt in Gleichung 3.71 folgt mit elementaren Umformungen und unter der Annahme, dass  $\mathbf{A}_{ULA}$  invertierbar ist:

$$\mathbf{T} \Gamma_F \mathbf{T}^{-1} = \mathbf{\Xi} \quad (3.73)$$

Aus dieser Gleichung folgt, dass die Eigenwerte von  $\Gamma_F$  durch  $\mathbf{\Xi}$  gegeben sind.

Da die Eigenvektoren  $\mathbf{E}_x$  lediglich geschätzt werden können, wird Gleichung 3.66 umgeformt

$$\mathbf{E}_x - \tilde{\mathbf{A}}_{ULA} \cdot \mathbf{T} = 0 \quad (3.74)$$

und das *Total-Least-Squares*-Kriterium (TLS-Kriterium) angewendet.

## 4 Variabler Phasenschieber

Die Richtungserkennung beruht im Wesentlichen darauf, dass die sich an den verschiedenen Mikrofonen ergebenden Laufzeitunterschiede für die Berechnung verwendet werden. Um somit in einer Simulation oder in einem Gesamtsystem Quellen zu simulieren, ist es notwendig, die empfangenen Signale entsprechend der Arraystruktur und des Einfallswinkels zu verzögern.

Ebenfalls ist es denkbar, empfangene Signale so zu verzögern, dass vorhandene Laufzeitunterschiede zwischen den Signalen ausgeglichen werden. Wäre dies möglich, kann die bestehende Redundanz zur Verbesserung des Signal-Rausch-Abstands (SNR) ausgenutzt werden.

Die Entwicklung eines Systems, das entsprechende Verzögerungen generieren kann, ist somit sinnvoll. In den nachfolgenden Abschnitten wird erläutert, wie die entsprechenden Verzögerungen in einem digitalen System nachgebildet werden können.

Zunächst werden die Grundlagen bezüglich eines variablen Phasenschiebers dargestellt. Bevor anschließend auf die notwendige blockweise Verarbeitung eingegangen wird, erfolgt die Vorstellung zweier Methoden. Diese werden messtechnisch untersucht. Das Kapitel schließt mit der Implementierung eines Quellensimulators.

### 4.1 Grundlegende Überlegungen

Verzögerungen um ganzzahlige Vielfache der Abtastperiode  $T$  sind trivial durch Verzögerungsglieder zu realisieren. Allerdings müssen ebenso Verzögerungen realisiert werden können, die ungleich einem ganzzahligen Vielfachen von  $T$  sind.

Es ist jedoch unmöglich ein solches System mit theoretischen, idealen Eigenschaften zu implementieren. Die Informationen zwischen den einzelnen Abtastzeitpunkten sind nicht vorhanden und können daher lediglich geschätzt bzw. approximiert werden. Hierfür existiert ein breites Feld an Realisierungsansätzen. Um sich einen Überblick über diese zu verschaffen, bietet sich der Artikel *Splitting the Unit Delay* [11] an.

Beispielsweise kann durch eine Kette aus Interpolator, Verzögerungsglied und Decimator, bzw. der entsprechenden Polyphase, Verzögerungen von  $T/L$  erreicht werden, wobei  $L$  den

Überabtastungs-Faktor beschreibt. Allerdings beschränkt dieses Vorgehen die Anzahl der Möglichkeiten der Verzögerungen. Diese Lösung bietet ebenfalls keine ideale Realisierung, da Anti-Imaging- und Anti-Alias-Filter nicht ideal realisiert werden können.

Darüber hinaus ist es wünschenswert, dass die Verzögerung durch einfache mathematische Operationen beliebig angepasst werden kann. Dies schließt generell Methoden aus, bei denen ein komplizierter Filterentwurf notwendig ist. Allerdings bieten sich Ansätze an, die entsprechende Zusammenhänge im Spektrum ausnutzen.

Der generelle Ansatz bei der Verwendung der zeitdiskreten Fouriertransformation lautet:

$$s[n - d_n] \approx \mathcal{F}^{-1} \{ H_{d_n}(e^{j\omega T}) \mathcal{F} \{ s[n] \} \} \quad (4.1)$$

Wobei  $s[n]$  das ursprüngliche Signal,  $H_{d_n}(e^{j\omega T})$  den Frequenzgang des Filters und  $s[n - d_n]$  das verzögerte Signal darstellt.

Generell gilt bei der Verwendung der zeitdiskreten Fouriertransformation, falls  $d_n$  eine ganze Zahl beschreibt:

$$s[n - d_n] \circ \bullet S(e^{j\omega T}) e^{-j\omega d_n T} \quad (4.2)$$

Daraus lässt sich ableiten, dass ein idealer Filter, der um  $d_n T$  verzögert, den Frequenzgang

$$H_{d_n}(e^{j\omega T}) = e^{-j\omega d_n T} \quad (4.3)$$

aufweisen muss. Die entsprechende Impulsantwort lässt sich mit Hilfe der inversen zeitdiskreten Fouriertransformation und  $\omega_s = 2\pi/T$  bestimmen:

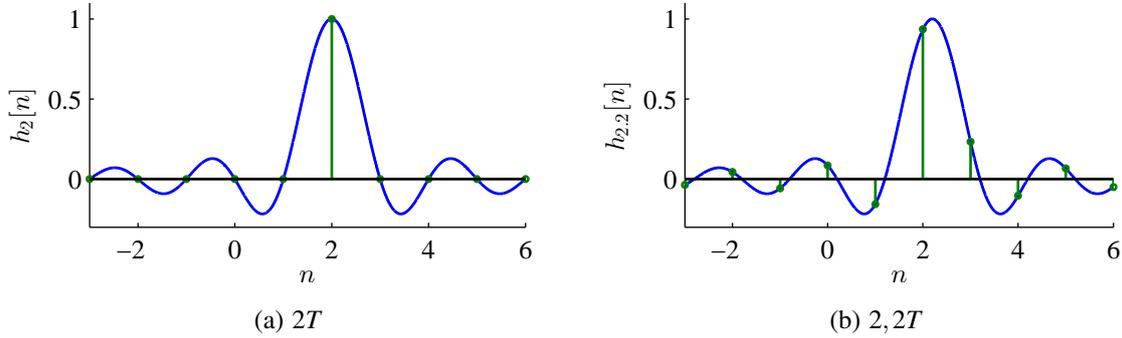
$$h_{d_n}[n] = \frac{1}{\omega_s} \int_{\frac{\omega_s}{2}}^{\frac{\omega_s}{2}} H_{d_n}(e^{j\omega T}) e^{j\omega n T} d\omega \quad (4.4)$$

Durch Lösen des Integrals ergibt sich

$$h_{d_n}[n] = si((n - d_n)\pi) = sinc(n - d_n) \quad (4.5)$$

als Ausdruck für die Impulsantwort  $h_{d_n}[n]$ . Im Unterbild (a) der Bildergruppe 4.1 ist die Impulsantwort für eine Verzögerung um  $2T$  dargestellt. Ebenfalls ist die Impulsantwort für nichtganzzahlige Vielfache von  $T$  - genauer  $2,2T$  - im Unterbild (b) dargestellt.

Es wird deutlich, warum Verzögerungen um ganzzahlige Vielfache realisiert und rationale (ausgenommen die ganzzahligen) Vielfache nicht realisiert werden können. Im Falle einer Verzögerung um ganzzahlige Vielfache von  $T$ , ist die Impulsantwort lediglich an einer Stützstelle ungleich Null. Im anderen Fall existiert für jede Stützstelle ein Funktionswert. Somit ist die Impulsantwort nicht kausal und das System nicht realisierbar.


**Bild 4.1:** Impulsantwort

Dies zeigt sich ebenso, wenn die Operation im Frequenzgang durchgeführt wird. Falls die Verzögerung ein nicht ganzzahliges Vielfaches von  $T$  ist, erzeugt die Rücktransformation ein komplexes Zeitsignal. Begründet ist dies darin, dass der Frequenzgang keine konjugiert-komplexe Struktur um  $f_s/2$  aufweist. Dazu sei angenommen, dass die Verzögerung durch den Ausdruck  $d_k T = d_{kZ}/d_{kN} T$  beschrieben werden kann. Es folgt:

$$H_{d_n}(e^{j\omega T}) = {}^{d_n N} \sqrt{e^{-j\omega d_{nZ} T}} \quad (4.6)$$

Die Wurzel komplexer Zahlen ist mehrdeutig. Aus diesem Grund wird nachfolgend stets vom Ergebnis mit der kleinsten Phase ausgegangen. Für einen konjugiert komplex symmetrischen Frequenzgang muss dieser bei  $f_s/2$  reell sein:

$$H_{d_n}\left(e^{j\frac{\omega_s}{2} T}\right) = {}^{d_n N} \sqrt{e^{-j\pi d_{nZ}}} \quad (4.7)$$

Dieser Ausdruck ist lediglich reell, falls gilt, dass  $d_{nZ}$  ein ganzzahliges Vielfaches von  $d_{nN}$  ist. Um dieses Problem zu umgehen, gibt es mehrere Ansätze. Zwei werden in den folgenden Abschnitten beschrieben.

Zuvor sei erwähnt, dass die zeitdiskrete Fouriertransformation aus Gleichung 4.1 bei der Realisierung in einem realen System durch die diskrete Fouriertransformation bzw. ihrer effizienten Implementierung der *Fast Fourier Transformation* (FFT) ersetzt werden muss. Es folgt mit  $H_{ALGO}[k]$  als Frequenzgang des verwendeten Algorithmus:

$$s[n - d_n] \approx \text{IFFT} \{H_{ALGO}[k] \text{FFT} \{s[n]\}\} \quad (4.8)$$

Für  $H_{d_n}(e^{j\omega T})$  gilt im Falle der DFT mit  $N$  vorliegenden Werten:

$$H_{d_n}[k] = H_{d_n}\left(e^{j2\pi\frac{k}{N}}\right) = {}^{d_n N} \sqrt{e^{-j2\pi d_{nZ} \frac{k}{N}}} \quad (4.9)$$

## 4.2 Methode nach Pei und Lai

Soo-Chang Pei und Yun-Chiu Lai haben in der Veröffentlichung *Closed Form Variable Fractional Time Delay Using FFT* [21] einen Filter dargelegt, der das beschriebene Problem umgeht, indem an der Stelle  $f_s/2$  lediglich der Realteil zur Berechnung verwendet wird. Für den Frequenzgang bei der Verwendung der diskreten Fouriertransformation ergibt sich somit:

$$H_{PL}[k] = \begin{cases} e^{-jd_n 2\pi k} & , k = 0, 1, \dots, \frac{N}{2} - 1 \\ \cos(d_n \pi) & , k = \frac{N}{2} \\ e^{jd_n 2\pi(N-k)} & , k = \frac{N}{2} + 1, \dots, N - 1 \end{cases} \quad (4.10)$$

Nach [21, S. 300] kann gezeigt werden, dass für die Impulsantwort des Systems gilt:

$$h_{PL}[n] = \frac{\cos\left(\frac{\pi(n-d_n)}{N}\right)}{\text{sinc}\left(\frac{(n-d_n)}{N}\right)} h_{d_n}[n] \quad (4.11)$$

Entsprechend dieser Impulsantwort kann dieser Ansatz mit Ansätzen verglichen werden, die *Finite Impuls Response*-Filter (FIR-Filter) mittels Fensterung der Impulsantwort  $h_{d_k}[k]$  realisieren [11, S. 36]. In [21] wird jedoch deutlich, dass dieser Ansatz etablierten Fenstern überlegen ist.

## 4.3 Methode mittels Spiegelung

Ein weiterer Ansatz komplexe Zeitsignale bei inversen Fouriertransformationen zu vermeiden, ist, die ursprünglichen Signale so zu verändern, dass deren Fouriertransformierte eine Nullstelle bei  $f_s/2$  bzw.  $k = N/2$  für gerade  $N$  aufweist. Es gilt für das Signal  $s[n]$  der Länge  $N$ :

$$S\left[\frac{N}{2}\right] = \sum_{n=0}^{N-1} s[n] e^{-j2\pi \frac{nk}{N}} \Bigg|_{k=\frac{N}{2}} = \sum_{n=0}^{N-1} s[n] e^{-j\pi n} \quad (4.12)$$

Dies kann aufgrund von  $e^{-j\pi n} = (-1)^n$  umgeschrieben werden in:

$$S\left[\frac{N}{2}\right] = \sum_{n=0}^{N-1} s[n] (-1)^n \quad (4.13)$$

Dieses Signal kann in zwei Summen aufgeteilt werden.

$$S\left[\frac{N}{2}\right] = \sum_{n=0}^{\frac{N}{2}-1} s[n](-1)^n + \sum_{n=\frac{N}{2}}^{N-1} s[n](-1)^n \quad (4.14)$$

Ändert man die Reihenfolge des zweiten Summanden, ergibt sich:

$$S\left[\frac{N}{2}\right] = \sum_{n=0}^{\frac{N}{2}-1} s[n](-1)^n + \sum_{n=0}^{\frac{N}{2}-1} s[N-1-n](-1)^{N-1-n} \quad (4.15)$$

$$= \sum_{n=0}^{\frac{N}{2}-1} (s[n](-1)^n + s[N-1-n](-1)^{N-1-n}) \quad (4.16)$$

Da  $N$  gerade ist und  $(-1)^n = (-1)^{-n}$  gilt, folgt:

$$S\left[\frac{N}{2}\right] = \sum_{n=0}^{\frac{N}{2}-1} (s[n](-1)^n - s[N-1-n](-1)^{-n}) \quad (4.17)$$

$$= \sum_{n=0}^{\frac{N}{2}-1} (-1)^n (s[n] - s[N-1-n]) \quad (4.18)$$

Somit ist die Fouriertransformierte des Signals  $s[n]$  an der Stelle  $k = N/2$  gleich null, falls folgende Symmetrie vorliegt:

$$s[n] = s[N-1-n] \quad (4.19)$$

Dieser Ansatz kann genutzt werden, um die Nullstelle des Frequenzganges bei  $f_s/2$  zu erzwingen. Dazu wird ein Signal  $s[n]$  der Länge  $N$  um seine gespiegelte Version  $s[-n]$  verlängert.

Indem das Signal  $s[n]$  zunächst auf die doppelte Länge mit Nullen aufgefüllt wird, ergibt sich das Signal  $s_e[n]$ . Für das um die gespiegelte Variante ergänzte Signal folgt:

$$s_{MRRR}[n] = s_e[n] + s_e[-n + (2N-1)] \quad (4.20)$$

Dieses Signal weist aufgrund seiner Konstruktion die geforderte Symmetrie auf. Nach entsprechender Filterung im Frequenzbereich mit  $H_{d_n}[k]$  werden lediglich  $N$  Werte des - durch die inverse Fouriertransformierte - gewonnenen Signals weiter verwendet.

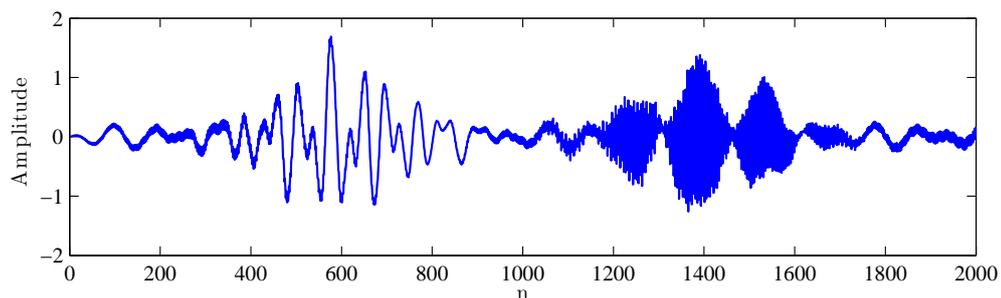
Jedoch beinhaltet dieses Vorgehen einen systematischen Fehler. Begründet ist dieser in der Spiegelung des Signals und durch die nicht endliche und nicht kausale Impulsantwort  $h_{d_n}[n]$  einerseits; andererseits in der Verwendung der diskreten Fouriertransformation (DFT), die

die Periodizität des Zeitsignals impliziert. Dies bewirkt unter anderem, dass nicht gewollte Signalanteile - zumeist stark gedämpft - das Wunschsinal überlagern.

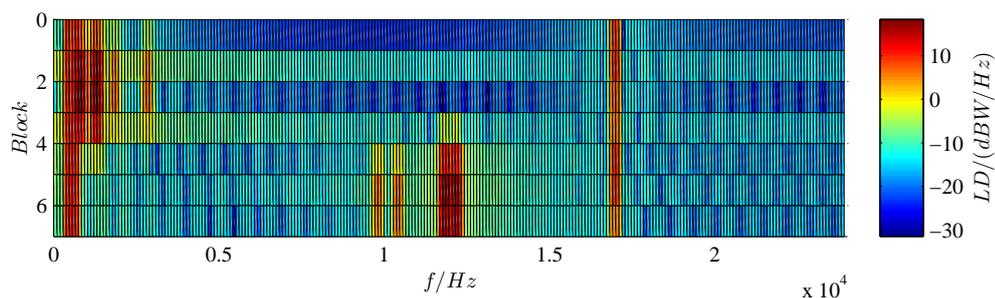
## 4.4 Blockweise Verarbeitung

Da Speicher und Rechenkapazitäten durch die gegebene Hardware begrenzt sind, ist eine blockweise Verarbeitung der Daten notwendig. In diesem Abschnitt werden die beiden zuvor vorgestellten Methoden miteinander verglichen. Dies geschieht anhand einer Simulation bei einer blockweisen Verarbeitung. Dabei wird davon ausgegangen, dass die DFT-Länge eine Potenz von Zwei hat und die verwendete Blocklänge eine gerade Zahl ist.

Um eine Bewertung der Methode durchführen zu können, wird eine synthetische Testsequenz generiert. Diese ist in Unterbild (a) der Bildergruppe 4.2 dargestellt. Das entsprechende Spektrogramm kann dem Unterbild (b) entnommen werden, wobei als Einheit des Zeitsignals Volt angenommen wird. LD steht für Leistungsdichte. Bei der Erstellung dieser Testsequenz ist darauf geachtet worden, dass eine gewisse Ähnlichkeit zu Sprachsignalen besteht (s. Abschnitt 6.1).



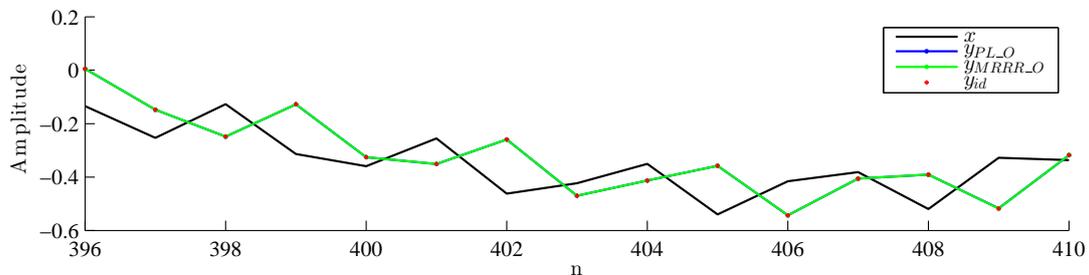
(a) Zeitverlauf



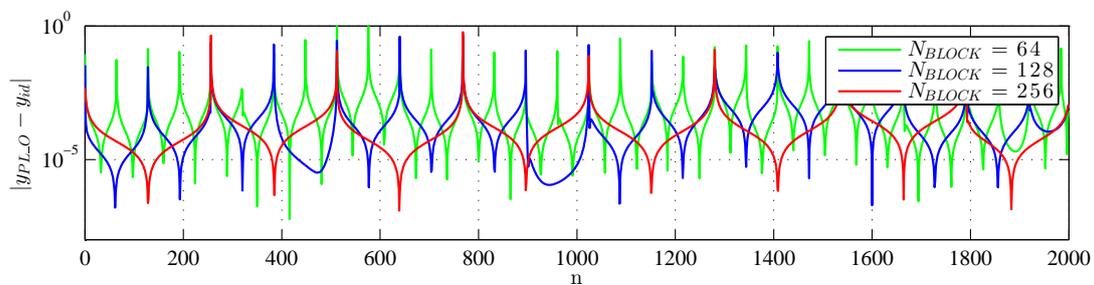
(b) Spektrogramm ( $N_{FFT} = 256$ , 50% Überhang, Hamming Fenster)

**Bild 4.2:** Testsequenz

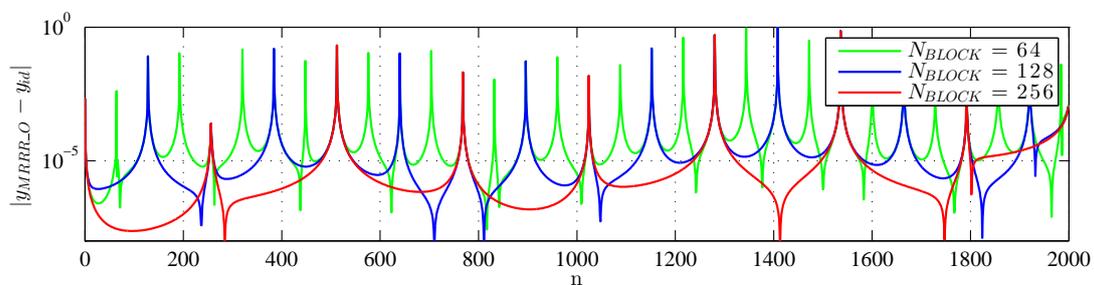
Wendet man die vorgestellten Methoden auf diese Testsequenz an, so ergeben sich die Ergebnisse aus Bild 4.3 für unterschiedliche Blocklängen. Es ist deutlich zu sehen, dass es zu Sprüngen bei den Blockübergängen kommt. Der Fehler ist in der Blockmitte minimal.



(a) Ausschnitt der Signalverläufe bei  $N_{BLOCK}=256$



(b) Fehler der Methode nach Pei und Lai



(c) Fehler der Methode mittels Spiegelung

**Bild 4.3:** Fehler und Zeitverläufe bei einer Verzögerung von  $1,5T$

Dieses Wissen kann angewendet werden, um den auftretenden Fehler zu minimieren. Dabei sind vor allem zwei Verbesserungen realisierbar:

1. Da der Fehler sich zur Blockmitte in der Regel verkleinert, ist es sinnvoll die Blocklänge gegenüber der DFT-Länge zu verkleinern. Die Differenz wird durch die letzten Werte des vorherigen Blocks aufgefüllt. Anschließend ist es möglich, lediglich die

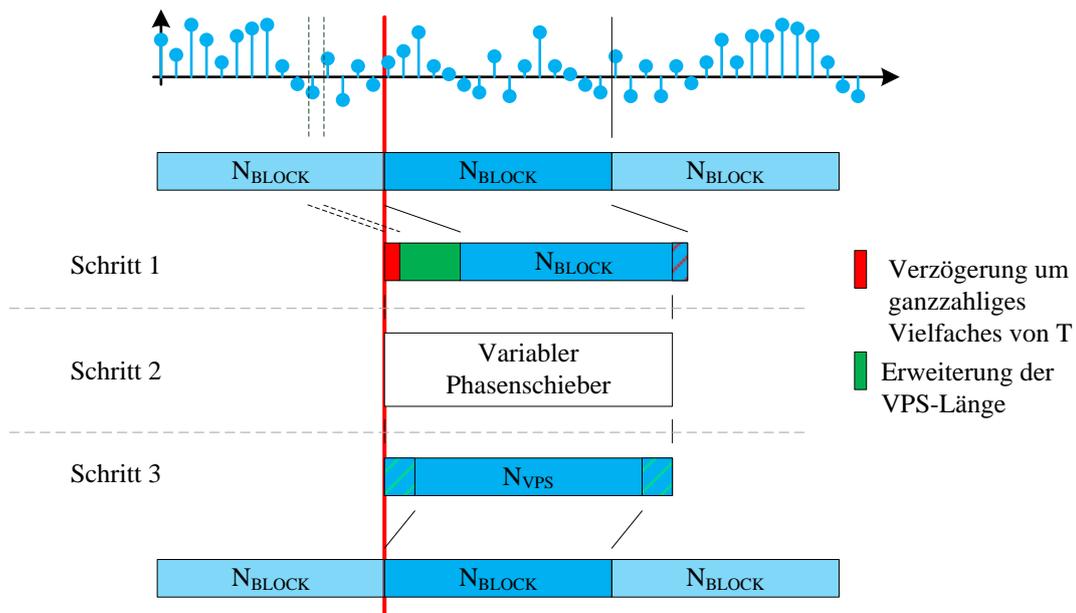
mittleren Werte zu verwenden. Durch dieses Vorgehen wird eine weitere Verzögerung von  $(N_{VPS} - N_{BLOCK})/2$  impliziert. Dabei beschreibt  $N_{VPS}$  die DFT-Länge der Pei und Lai Methode bzw. die halbe DFT-Länge der Methode mittels Spiegelung und  $N_{BLOCK}$  die Blocklänge. Die Verzögerung ist jedoch akzeptabel, da lediglich relative Verzögerungen zwischen den einzelnen Signalen von Bedeutung sind.

- Die entsprechende Verzögerung kann stets zweistufig realisiert werden. Dazu wird die Verzögerung  $d_n T$  in

$$d_n T = d_{nT} T + d_{nR} T \tag{4.21}$$

aufgeteilt. Dabei beschreibt  $d_{nT} T$  eine Verzögerung um ein ganzzahliges Vielfaches von  $T$ . Die restliche Verzögerung wird durch  $d_{nR} T$  beschrieben. Die Verzögerung  $d_{nT} T$  kann durch Verzögerungsglieder ideal realisiert werden. Für die Verzögerung um  $d_{nR} T$  werden die vorgestellten Methoden angewendet.

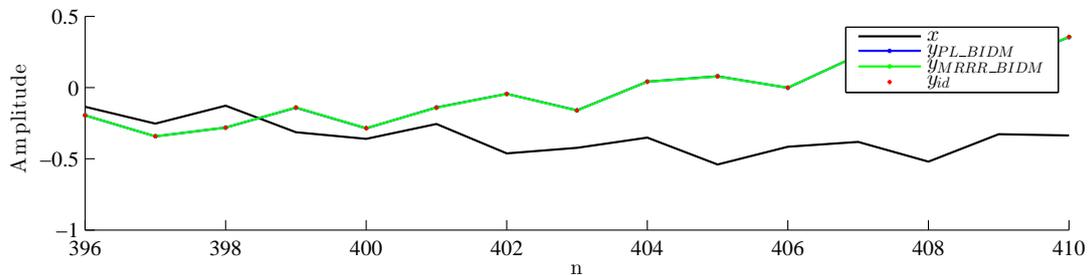
In Bild 4.4 wird dieses Vorgehen grafisch dargestellt.



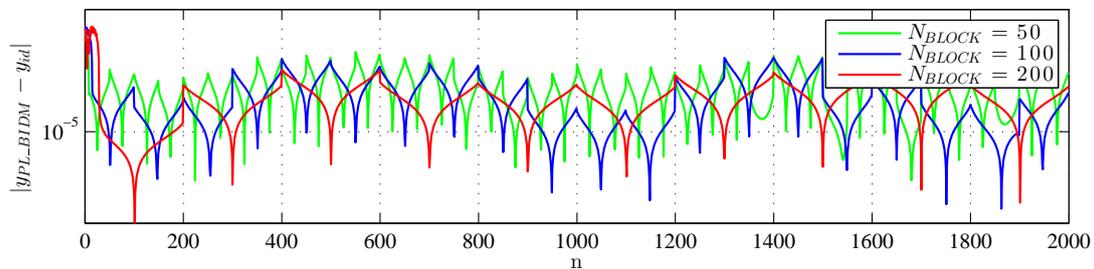
**Bild 4.4:** Blockweise Verarbeitung des variablen Phasenschiebers

Ein zu verzögernder Block wird in Schritt 1 um die  $N_{VPS} - N_{BLOCK} + d_{nT}$  letzten Werte des vorherigen Blocks erweitert. Die letzten  $d_{nT}$  Werte des aktuellen Blocks finden keine Verwendung. Mit den  $N_{VPS}$  Werten wird die Verzögerung  $d_{nT}$  durch einen variablen Phasenschieber in Schritt 2 realisiert. Anschließend werden in Schritt 3 die ersten und die letzten  $(N_{VPS} - N_{BLOCK})/2$  Werte des resultierenden Blocks verworfen. Folglich ergeben sich  $N_{BLOCK}$  verzögerte Werte.

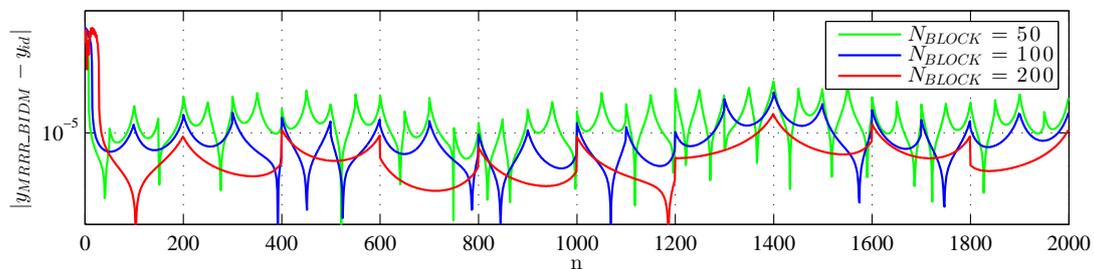
Wendet man die Methoden mit den beschriebenen Erweiterungen an, so ergeben sich die Ergebnisse aus Bild 4.5.



(a) Ausschnitt der Signalverläufe bei  $N_{BLOCK}=200$



(b) Fehler der Methode nach Pei und Lai



(c) Fehler der Methode mittels Spiegelung

**Bild 4.5:** Fehler und Zeitverläufe bei einer Verzögerung von  $1,5T$  (mit Erweiterungen)

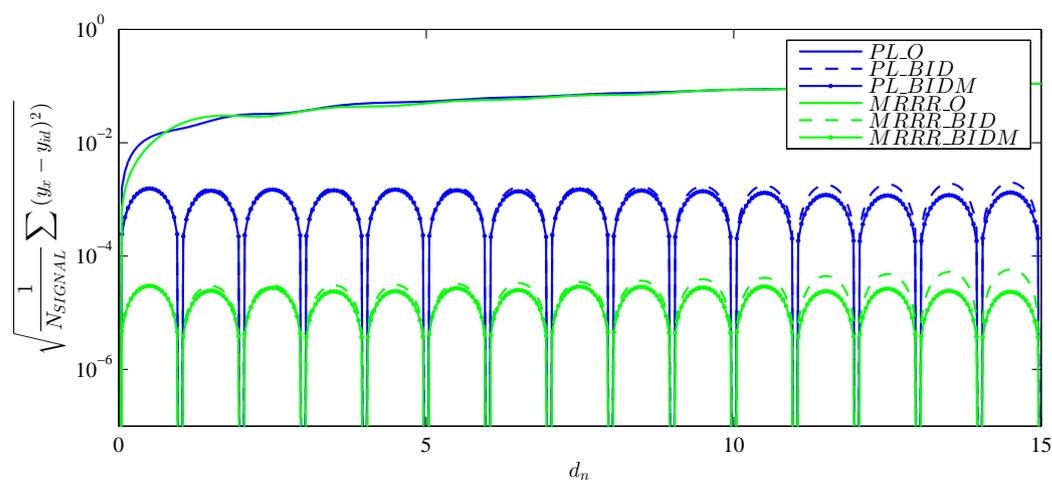
Der zuvor erhebliche Fehler hat sich durch die Erweiterungen deutlich reduziert. Lediglich beim ersten Block liegen die Fehler in der vorherigen Größenordnung. Dies ist so zu erwarten, da keine Kenntnisse über das vorherige Signal vorhanden sind.

Der Fehler verkleinert sich vor allem bei der Methode mittels Spiegelung mit zunehmender Blocklänge. Es ist somit wünschenswert, die verwendete Blocklänge möglichst zu maximieren.

Um einen besseren Vergleich zwischen den beiden Methoden zu gewährleisten, sind in Bild 4.6 die Fehler der verschiedenen Methoden gegen  $d_n$  aufgetragen, wobei der erste Block nicht mit einbezogen ist. Dabei stehen die Suffixe in der Legende für:

- $\_O$  : ursprüngliche Methoden
- $\_BID$  : ursprüngliche Methoden und Erweiterung 1
- $\_BIDM$  : ursprüngliche Methoden und Erweiterungen 1 und 2

Die Blocklänge beträgt 200. PL steht für die Methode nach Pei und Lai; MRRR für die Methode mittels Spiegelung.



**Bild 4.6:** Fehler in Abhängigkeit der Verzögerung

Betrachtet man die Fehlerverläufe, so wird deutlich, dass die Methoden ohne Erweiterungen die größten Fehler aufweisen. Die Methoden mit der Erweiterung 1 weisen einen deutlich kleineren und sich wiederholenden Fehlerverlauf auf. Die Fehler werden in Richtung ganzzahliger Verzögerungen kleiner, allerdings nehmen die Fehler mit steigenden  $d_n$  langsam zu. Dies wird mit Erweiterung 2 korrigiert. Die Fehlerverläufe bleiben auch mit zunehmender Verzögerung  $d_n$  auf einem gleichen Niveau. Die Abweichungen von der Systematik in Bild 4.6 ergeben sich dadurch, dass die Fehlerverläufe abhängig von dem betrachteten Signal sind.

Insgesamt wird deutlich, dass die Methode mittels Spiegelung für die entsprechende Testsequenz mit einem kleineren Fehler behaftet ist, als die Methode nach Pei und Lai.

Allerdings sei an dieser Stelle auch darauf hingewiesen, dass die Implementierung der Erweiterung einen deutlichen Mehraufwand bedeuten kann. Somit ist es stets abzuwägen, ob Erweiterungen für die entsprechende Anwendung vonnöten sind oder ob eine effizientere Umsetzung gefordert ist. Genauer wird auf diesen Zusammenhang in Abschnitt 4.7 eingegangen.

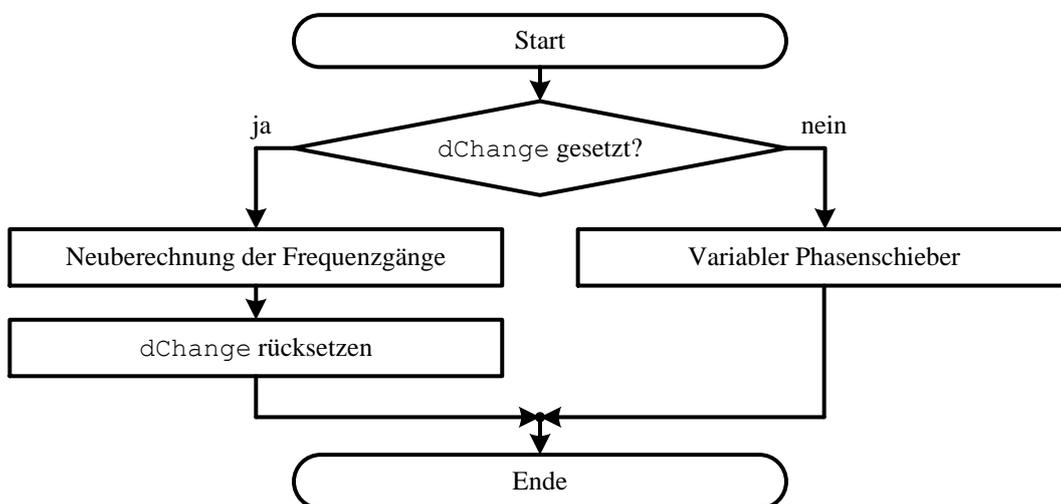
## 4.5 Implementierung

Die Implementierung kann entsprechend dem Schema aus Bild 4.4 durchgeführt werden. Für eine effiziente Berechnung der FFT wird die komplexe Radix2-FFT (DIT): `void DSPF_sp_cfftr2_dit (float *x, float *w, short n)` aus der *TI C67x DSPLib* (s. Abschnitt 2.5) verwendet.

Für die Einstellung des Programmes haben folgende Parameter, die als `#define` realisiert sind, Einfluss:

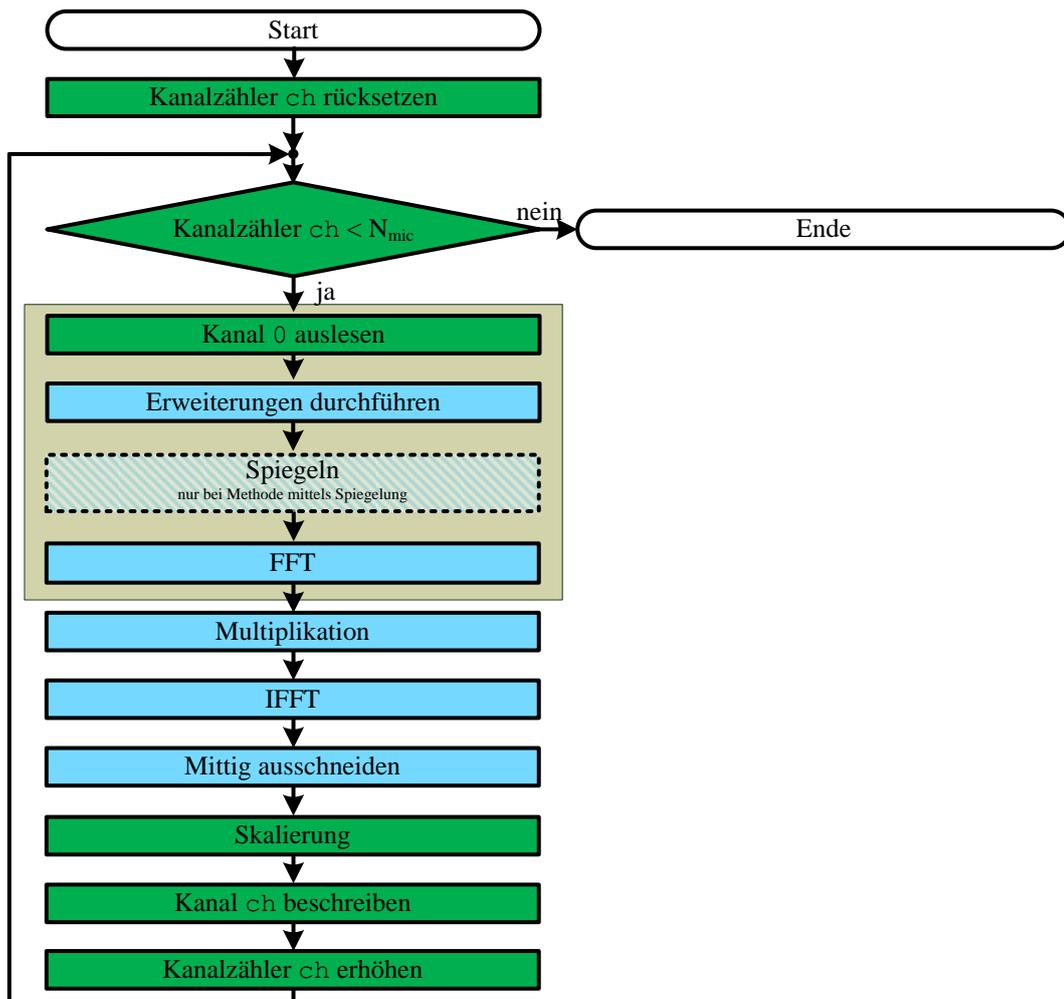
- **N\_BLOCK**: Dieser Parameter stellt die Blocklänge der EDMA ein.
- **N\_FFT**: Dieser Parameter legt die FFT-Länge fest.
- **NO\_FDF**: Falls definiert, wird keine variable Verzögerung durchgeführt.
- **MRRR**: Falls definiert, wird die Methode mittels Spiegelung durchgeführt. Falls nicht, wird die Methode nach Pei und Lai verwendet.
- **D\_MAX**: Dieser Parameter gibt die maximale Verzögerung, die durch Erweiterung Zwei berücksichtigt werden kann, an.
- **N\_CHANNEL**: Der Parameter legt die Anzahl der (Ein- und) verzögerten Ausgänge fest. Stimmt in den meisten Fällen mit  $N_{mic}$  überein.

Der Ablauf der Hauptfunktion `main` ist in Bild 4.7 als Flussdiagramm dargestellt.



**Bild 4.7:** Grundprogramm des variablen Phasenschiebers

Um die Verzögerungen während der Laufzeit des Programmes einzustellen, existiert die Variable `dChange`. Sobald diese gesetzt ist, werden die Frequenzgänge neu berechnet. Ist `dChange` nicht gesetzt, werden die Signale verzögert und ausgegeben. Der Ablauf dieser Verzögerung ist in Bild 4.8 als Flussdiagramm grafisch dargestellt. Dabei sind die Programmschritte, die durch entsprechende Methoden vorgegeben werden, in Blau und alle weiteren Schritte zur Realisierung in Grün dargestellt.



**Bild 4.8:** Ablaufdiagramm des variablen Phasenschiebers

Für alle Ausgangskanäle werden die Daten aus Kanal 0 eingelesen. Anschließend werden die entsprechenden Erweiterungen 1 und 2 sowie, falls nötig, die Spiegelung durchgeführt. Daraufhin folgt der Übergang in den Frequenzbereich, die Multiplikation mit der Übertragungsfunktion und die inverse Fouriertransformation sowie das Ausschneiden der Daten entsprechend Erweiterung 1. Des Weiteren ist nachfolgend noch eine Skalierung (entsprechend FFT-Definiton) notwendig.

Auffällig ist, dass aufgrund von Erweiterung 2 alle Aktionen für alle Ausgangskanäle durchgeführt werden müssen. Grund hierfür ist, dass das Eingangssignal des variablen Phasenschiebers bereits um das entsprechende ganzzahlige Vielfache verzögert sein muss. Folglich können sich alle Eingangssignale unterscheiden. Verzichtet man hingegen auf Erweiterung 2 ist es möglich, die Operationen inklusive der FFT (brauner Kasten) vor die Schleife zu ziehen, da die Eingangssignale aller Ausgangssignale übereinstimmen. Ob dieses Vorgehen sinnvoll ist, muss anhand der Anforderungskriterien der jeweiligen Anwendung entschieden werden.

Ebenfalls sei an dieser Stelle darauf hingewiesen, dass das Beschreiben der entsprechenden Zwischenspeicher für den nächsten Block nicht in der Schleife stattfindet.

## 4.6 Messungen

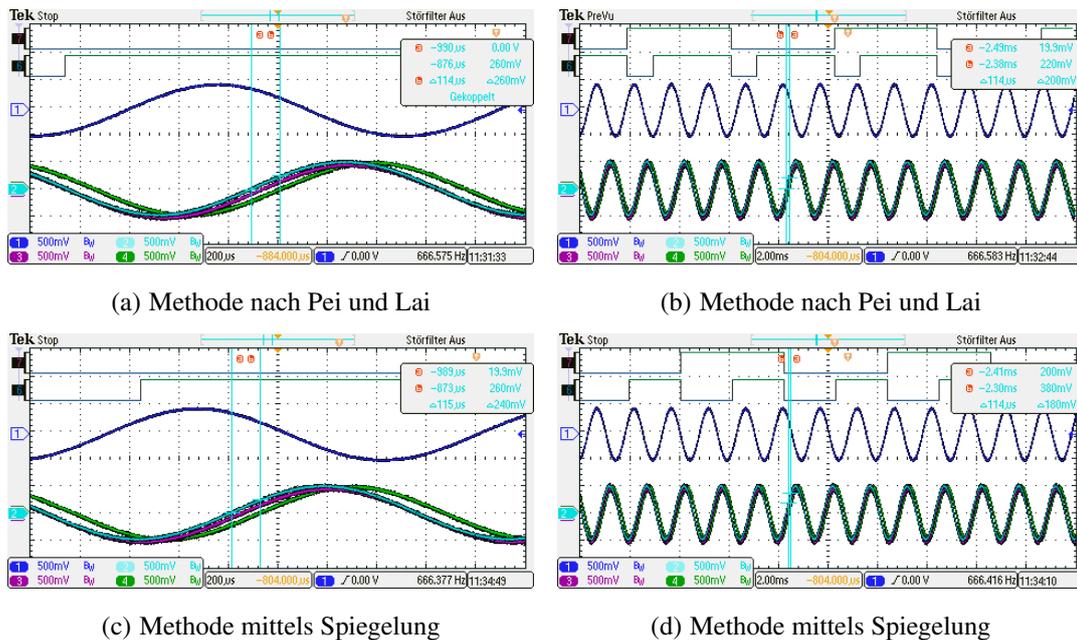
Die korrekte Funktionsweise der im vorherigen Abschnitt vorgestellten Methoden wird in diesem Abschnitt belegt. Folgende elementare Parameter werden für die Messungen gewählt:

- Abtastfrequenz  $F_S=48$  kHz
- EDMA-Blocklänge  $N_{BLOCK}=200$
- FFT-Länge  $N_{FFT}=512$  (maximale Größe - durch IRAM-Größe begrenzt) bzw.  $N_{FFT}=256$
- Anzahl der Kanäle  $N_{CHANNEL}=8$

In Bild 4.9 sind die Zeitverläufe und die entsprechenden Rechenzeiten, die durch digitale Ein- und Ausgänge gemessen werden, dargestellt. Dabei repräsentiert Kanal 7 die EDMA-Blocklänge, wobei jeder Pegel für eine Blocklänge steht. Kanal 8 beschreibt mit der Dauer des Nullpegels des Signals die Rechendauer des Algorithmus. Kanal 1 zeigt das Eingangssignal und die Kanäle 2-4 die Ausgangssignale, wobei das Signal in Kanal 2 um 0 T, das Signal in Kanal 3 um 1 T und in Kanal 4 um 5,5 T verzögert ist. Als relative Verzögerung zwischen Kanal 2 und 4 ergibt sich somit ein Sollwert von  $114,58 \mu s$ .

Analysiert man die Ergebnisse, so kann festgehalten werden, dass die Nulldurchgänge der Signalverläufe um ca.  $114 \mu s$  verzögert sind, wobei das Eingangssignal eine Frequenz von ca. 666 Hz aufweist. Betrachtet man die benötigte Rechenzeit, so wird deutlich, dass die Methode nach Pei und Lai lediglich die Hälfte der Zeit der Methode mittels Spiegelung benötigt. Dies ist auch nicht weiter verwunderlich, da bei der Methode mittels Spiegelung die doppelte FFT-Länge benötigt wird, bei gleicher EDMA-Blocklänge.

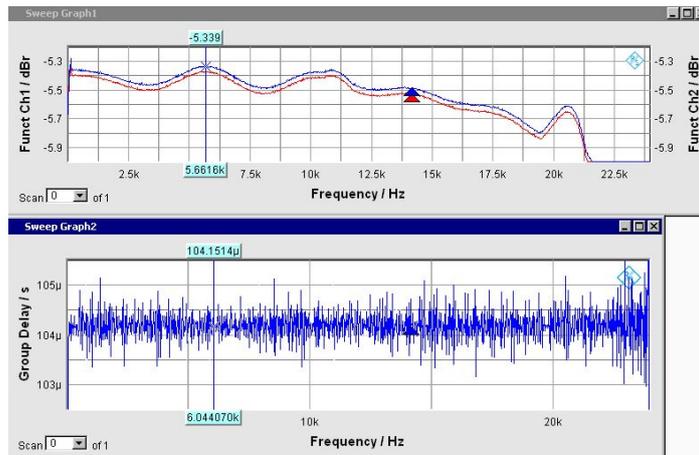
Um Aussagen über den Frequenzverlauf der betrachteten Systeme zu treffen, ist mit Hilfe des *UPV Audio Analyzers* der Amplitudengang sowie der Verlauf der Gruppenlaufzeit aufgenommen worden. In Bild 4.10 sind die Ergebnisse dargestellt. Dabei sind in Blau


**Bild 4.9:** Zeitverläufe und Rechenzeit

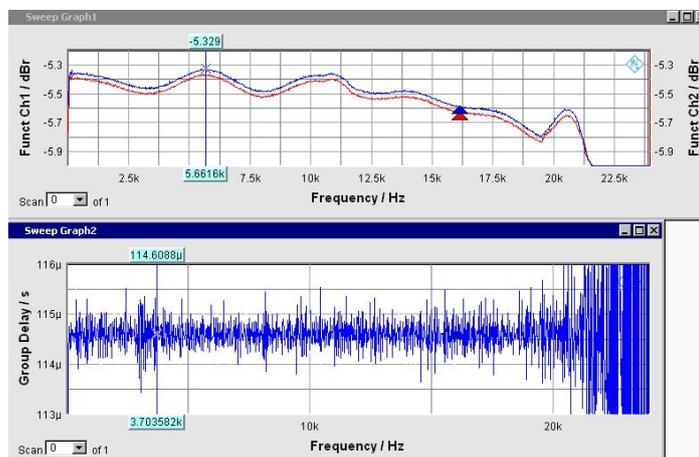
die Amplitudengänge der nicht verzögerten und in Rot der verzögerten Signale dargestellt. Weiter wird die relative Gruppenlaufzeit der Signale zueinander geplottet.

Die Ergebnisse zeigen im Unterbild (a) eine ganzzahlige Verzögerung um  $d_n = 5T$  als Referenzmessung. In den Unterbildern ist jeweils die Verzögerung  $d_n = 5,5T$  gewählt worden. Die Methode mittels Spiegelung weist einen ähnlichen Amplitudengang und Verlauf der Gruppenlaufzeit wie die ganzzahlige Verzögerung auf. Die Gruppenlaufzeit hat sich lediglich entsprechend der eingestellten Verzögerung vergrößert. Die Methode nach Pei und Lai weist deutliche Schwankungen im Amplitudengang und größere Schwankungen beim Verlauf der Gruppenlaufzeit auf.

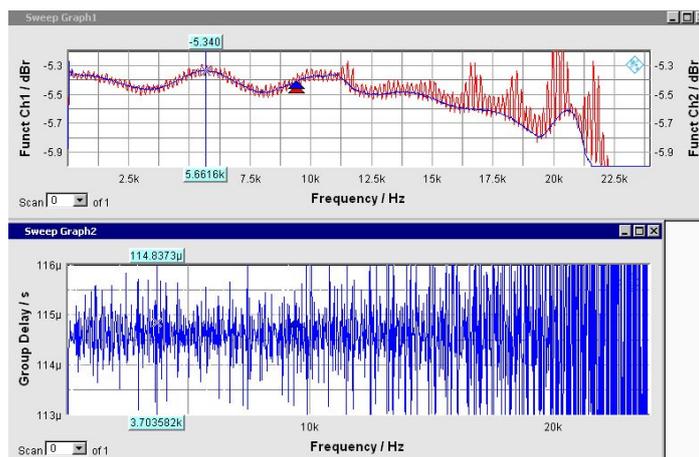
Somit stellt die Methode mittels Spiegelung laut den Messergebnissen und der Simulation die bessere Approximation des idealen Falles dar.



(a) ganzzahlige Verzögerung,  $d_n = 5T$



(b) Methode mittels Spiegelung,  $d_n = 5,5T$



(c) Methode nach Pei und Lai,  $d_n = 5,5T$

**Bild 4.10:** Amplitudengang und Gruppenlaufzeit

## 4.7 Quellensimulator

Wie bereits erwähnt, kann mit Hilfe des variablen Phasenschiebers ein Quellensimulator entwickelt werden. Dazu sind wenige Erweiterungen im Vergleich zum eigentlichen variablen Phasenschieber nötig. Ziel ist es Gleichung 3.7 nachzubilden. Dies bedeutet, dass neben der zeitlichen Verzögerung auch Rauschen additiv überlagert werden muss.

Für die Implementierung wird die maximale Quellenanzahl auf zwei beschränkt. Diese Begrenzung ist darin begründet, dass gängige PC-Systeme lediglich einen Audioausgang mit maximal zwei Audiospuren aufweisen. Weiter könnte, falls mehrere Quellen simuliert werden müssen, dies auch mittels einer Parallelschaltung von mehreren DSPs und anschließender Kombination durchgeführt werden.

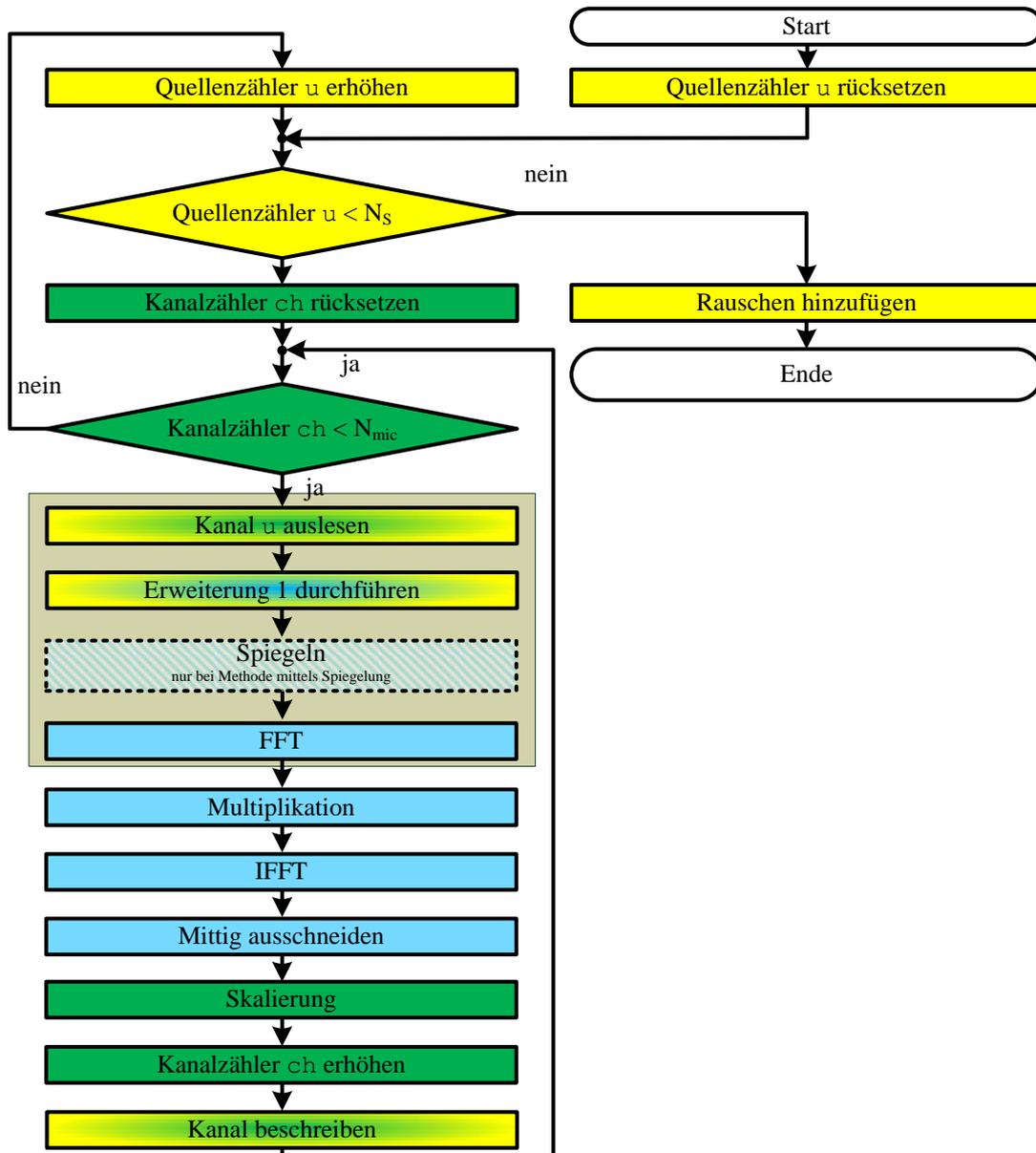
In Bild 4.11 ist das erweiterte bzw. angepasste Flussdiagramm dargestellt. Dabei sind in gelber Farbe Ergänzungen und in Farbüberläufen Änderungen eingezeichnet. Ebenso sind im braunen Kasten die Punkte aufgeführt, die, falls Erweiterung 2 nicht durchgeführt wird, vor die Kanalzähler-Schleife gezogen werden können. Ebenfalls sei an dieser Stelle darauf hingewiesen, dass das Beschreiben der entsprechenden Zwischenspeicher für den nächsten Block nicht in der Schleife stattfindet.

Während der Implementierung der verschiedenen Verfahren hat sich gezeigt, dass aufgrund der begrenzten Rechenkapazitäten unterschiedliche Parameter für die einzelnen Methoden eingestellt werden können. Um einen Vergleich der möglichen Fehler durchzuführen, sind die Methoden mit entsprechenden Parametern und der Testsequenz aus Abschnitt 4.4 simuliert worden. Das Ergebnis ist in Bild 4.12 dargestellt.

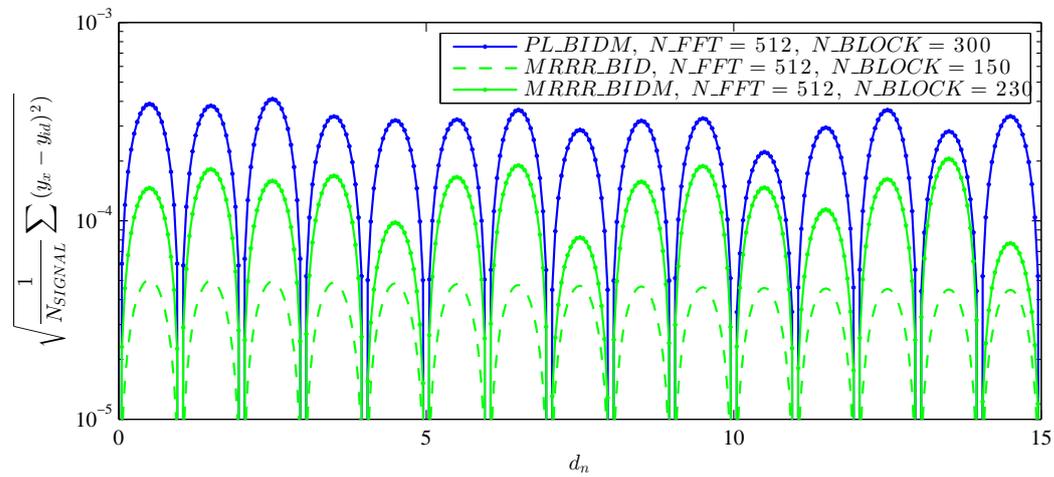
Es zeigt sich, dass die Variante der Methode mittels Spiegelung ohne Erweiterung 2 den kleinsten Fehler für die simulierte Spanne an Verzögerungen aufweist. In Bild 4.13(a) sind Messungen der Methode mittels Spiegelung mit Erweiterung 1 und mit Erweiterung 1 und 2 abgebildet. Dabei zeigen Kanal 6 und 7 die durch Nullpegel gekennzeichnete Rechendauer des Algorithmus bzw. durch Dauer der Pegel gekennzeichnete EDMA-Blocklänge. Kanal 1 zeigt das Eingangssignal und Kanäle 3 und 4 die Ausgangssignale, die relativ zueinander maximal (hier  $d_n = 12, 16$ ;  $\Delta t = 12, 16T \approx 250 \mu\text{s}$  bei  $T = 1/f_s$  mit  $f_s = 48 \text{ kHz}$ ) verzögert sind.

Die Analyse der Ergebnisse zeigt, dass ...

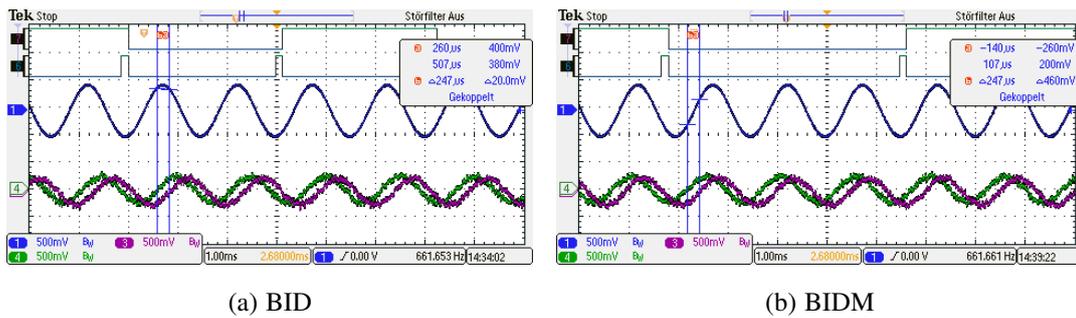
- die Blocklänge mit der Erweiterung 1 und 2 deutlich größer als die Blocklänge des Algorithmus mit lediglich der Erweiterung 1 ist.
- beide Methoden mit den entsprechenden Parametern echtzeitfähig sind.
- die relative Verzögerung im Rahmen der Cursor-Genauigkeit mit dem theoretischen Wert übereinstimmt.



**Bild 4.11:** Ablaufdiagramm des Quellensimulators



**Bild 4.12:** Vergleich der Fehler der Methoden mit realisierbaren Parametereinstellungen



**Bild 4.13:** Zeitverläufe und Rechenzeit des Quellensimulators bei der Methode mittels Spiegelung

- die Amplituden der Ausgangssignale um ca. die Hälfte im Vergleich zu den Amplituden des Eingangssignals reduziert sind. Begründet ist dies darin, dass der Algorithmus für zwei Quellen parametrisiert ist und entsprechend skaliert, jedoch nur ein Eingangssignal anliegt.
- die Ausgangssignale deutlich verrauscht sind.

Alle Ergebnisse lassen auf eine korrekte Funktionsweise des Quellensimulators schließen. Folglich wird dieser für alle nachfolgenden Simulationen verwendet. Dabei wird bei rechnerbasierten Simulationen auf die Methode mittels Spiegelung mit Erweiterung 1 und 2 zurückgegriffen. Für hardwarenahe Simulationen wird auf Erweiterung 2 verzichtet.

# 5 UCA-ESPRIT-Algorithmus

In diesem Kapitel wird der UCA-ESPRIT-Algorithmus vorgestellt. Die entsprechenden Quellen sind [2], [6], [8], [12], [13] und [39]. Dieser Algorithmus beruht ebenfalls auf der grundsätzlichen Idee der unterraumbasierten Verfahren - der Hauptkomponentenanalyse mittels Eigenwertzerlegung. Allerdings müssen die Eigenheiten einer zirkularen Struktur berücksichtigt werden. Diese sollen nachfolgend dargelegt werden, bevor der Algorithmus an sich vorgestellt wird und entsprechende Simulationen durchgeführt werden.

## 5.1 Zirkulares Array

In Kapitel 3 ist von einem gleichverteilten, linearen Array ausgegangen worden. Durch diese Struktur wird die Lokalisierung jedoch auf den Azimutwinkel begrenzt. Ein gleichverteiltes, zirkulares Array jedoch bietet die Möglichkeit auch den Elevationswinkel zu bestimmen. In der Bildergruppe 5.1 werden die Zusammenhänge zwischen den einzelnen Mikrofonen, der Wellenfront und der entsprechenden Einfallswinkel (Azimutwinkel  $\varphi$  und Elevationswinkel  $\theta$ ) deutlich.

Die Wellenfront kann dabei als Ebene beschrieben werden. Der Normalvektor der Ebene, der die Orientierung im Raum angibt, sei zu

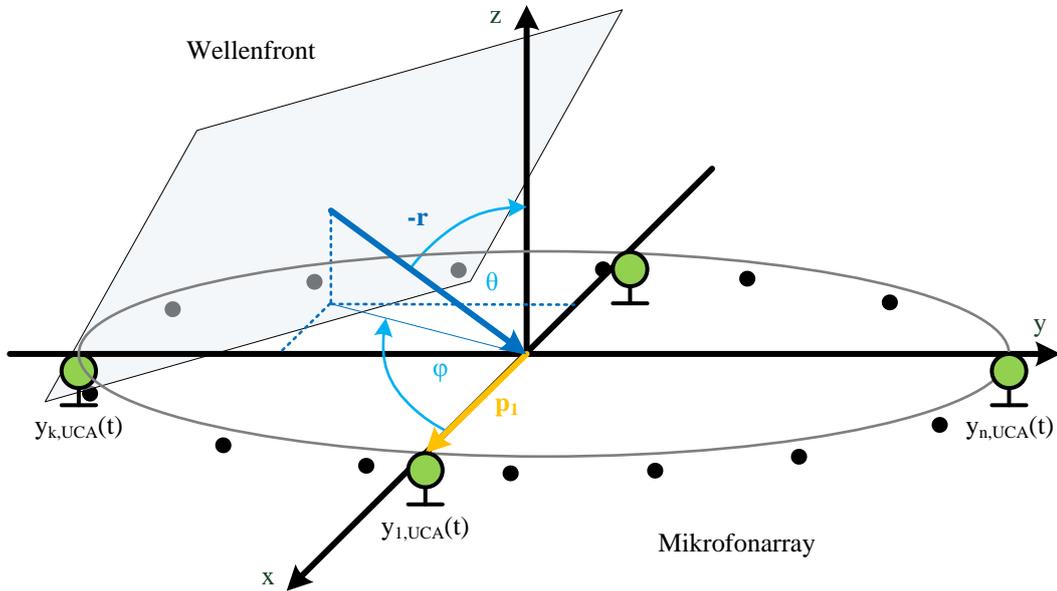
$$\mathbf{r} = [\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta]^T \quad (5.1)$$

definiert. Die Position des  $i$ -ten Mikrofons kann durch den entsprechenden Ortsvektor (s. Bild 5.1(b))

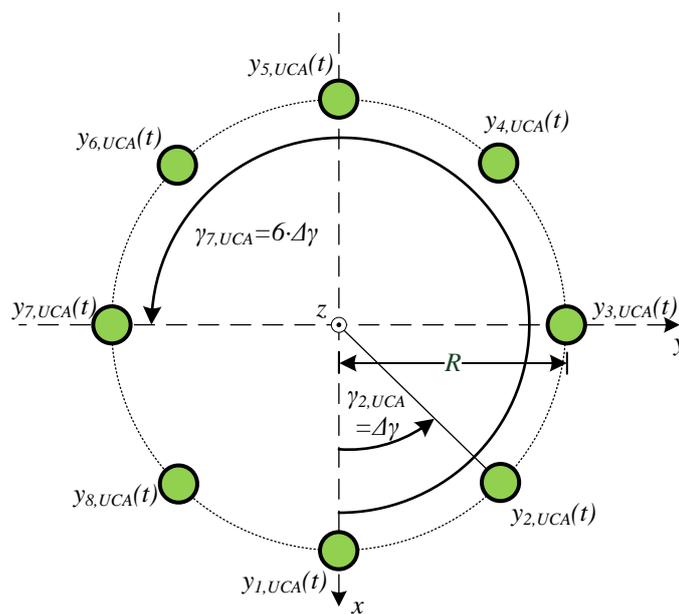
$$\mathbf{p}_i = [R\cos\gamma_i, R\sin\gamma_i, 0]^T \quad (5.2)$$

beschrieben werden. Dabei ist  $R$  der Radius der Arraystruktur. Berücksichtigt man die Gleichverteilung der Mikrofone so ergibt sich  $\mathbf{p}_i$  mit

$$\gamma_i = \Delta\gamma(i-1) \text{ mit } \Delta\gamma = \frac{2\pi}{N_{mic}} \quad (5.3)$$



(a) Raumsicht



(b) Draufsicht

**Bild 5.1:** Gleichverteiltes, zirkulares Mikrofonarray

zu

$$\mathbf{p}_i = [R \cos(\Delta\gamma(i-1)), R \sin(\Delta\gamma(i-1)), 0]^T \quad (5.4)$$

Es gilt für die Laufzeitdifferenz mit Hilfe des Skalarproduktes des Normalenvektors der Wellenfront und des jeweiligen Ortsvektors des Mikrofons

$$\tau_i = -\frac{1}{v} \mathbf{r} \mathbf{p}_i \quad (5.5)$$

Mit den Definitionen der Vektoren und dem Additionstheorem

$$\cos(\alpha - \beta) = \cos\alpha \cos\beta + \sin\alpha \sin\beta \quad (5.6)$$

ergibt sich:

$$\tau_i = -\frac{R}{v} \sin\theta \cos(\varphi - \gamma_i) \quad (5.7)$$

Verwendet man die Bezeichnungen

$$k_0 = \frac{2\pi f}{v} \quad (5.8)$$

$$\zeta = k_0 R \sin\theta \quad (5.9)$$

so folgt, für den Richtungsvektor  $\mathbf{a}(\zeta, \varphi)$ :

$$\mathbf{a}_{UCA}(\zeta, \varphi) = \left[ e^{j\zeta \cos(\varphi - \gamma_1)}, e^{j\zeta \cos(\varphi - \gamma_2)}, \dots, e^{j\zeta \cos(\varphi - \gamma_N)} \right]^T \quad (5.10)$$

Es lässt sich erkennen, dass die daraus resultierende Richtungsmatrix keine transponierte Vandermonde-Matrix mehr ist und somit die beschriebenen Methoden aus Kapitel 3 nicht angewendet werden können.

## 5.2 Phasen-Moden Anregung

In diesem Abschnitt wird die Phasen-Moden Anregung von zirkularen Arrays diskutiert. Diese Betrachtungsweise ermöglicht es, die Richtungsmatrix des zirkularen Arrays in eine Richtungsmatrix mit transponierter Vandermonde-Struktur zu transformieren.

### 5.2.1 Kontinuierliche zirkulare Mikrophonarrays

Die normalisierte Fernfeld Richtcharakteristik eines kontinuierlichen zirkularen Mikrophonarrays kann mit

$$b^c(\zeta, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} \omega_a^c(\gamma) e^{j\zeta \cos(\varphi - \gamma)} d\gamma \quad (5.11)$$

angegeben werden [6, S. 299]. Dabei beschreibt  $\omega_a^c(\gamma)$  die Anregungsfunktion beim Winkel  $\gamma$ . Da diese Anregungsfunktion periodisch mit  $2\pi$  aufgrund der angenommenen Geometrie sein muss, kann  $\omega_a^c(\gamma)$  mit Hilfe der inversen Fourierreihe beschrieben werden:

$$\omega_a^c(\gamma) = \sum_{m=-\infty}^{\infty} \omega_{a,m}^c(\gamma) = \sum_{m=-\infty}^{\infty} \omega_m^c e^{jm\gamma} \quad (5.12)$$

Dabei beschreibt  $\omega_m$  den m-ten Fourierkoeffizienten. Jeder Summand wird als Phasen-Moden Anregung des zirkularen Arrays bezeichnet [6, S. 305]. Mit der Annahme, dass alle Harmonischen gleich gewichtet werden ( $\omega_m^c = 1$ ), folgt für  $b^c(\zeta, \varphi)$  mit Gleichung 5.11 und 5.12 einer elementaren Umformung:

$$b^c(\zeta, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} \sum_{m=-\infty}^{\infty} e^{jm\gamma} e^{j\zeta \cos(\varphi - \gamma)} d\gamma \quad (5.13)$$

$$= \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_0^{2\pi} e^{jm\gamma} e^{j\zeta \cos(\varphi - \gamma)} d\gamma \quad (5.14)$$

Betrachtet man die Summanden, so können diese als normalisierte Fernfeld Richtcharakteristik der m-ten Phasen-Modus Anregung angesehen werden. Es gilt:

$$b^c = \sum_{m=-\infty}^{\infty} b_m^c(\zeta, \varphi) \quad (5.15)$$

$$b_m^c(\zeta, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} e^{jm\gamma} e^{j\zeta \cos(\varphi - \gamma)} d\gamma \quad (5.16)$$

Substituiert man  $\gamma$  mit  $\gamma' = \gamma - \varphi$ , so folgt mit  $\cos(\alpha) = \cos(-\alpha)$ :

$$b_m^c(\zeta, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} e^{j(m(\gamma' + \varphi) + \zeta \cos(\gamma'))} d\gamma' \quad (5.17)$$

$$= \frac{e^{jm\varphi}}{2\pi} \int_0^{2\pi} e^{j(m\gamma' + \zeta \cos(\gamma'))} d\gamma' \quad (5.18)$$

Berücksichtigt man, dass die Besselfunktion erster Art (s. [32, S. 211]) zu

$$j^m J_m(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{j(x\cos(\beta)+m\beta)} d\beta \quad (5.19)$$

definiert ist, kann  $b_m^c$  wie folgt

$$b_m^c(\zeta, \varphi) = j^m J_m(\zeta) e^{jm\varphi} \quad (5.20)$$

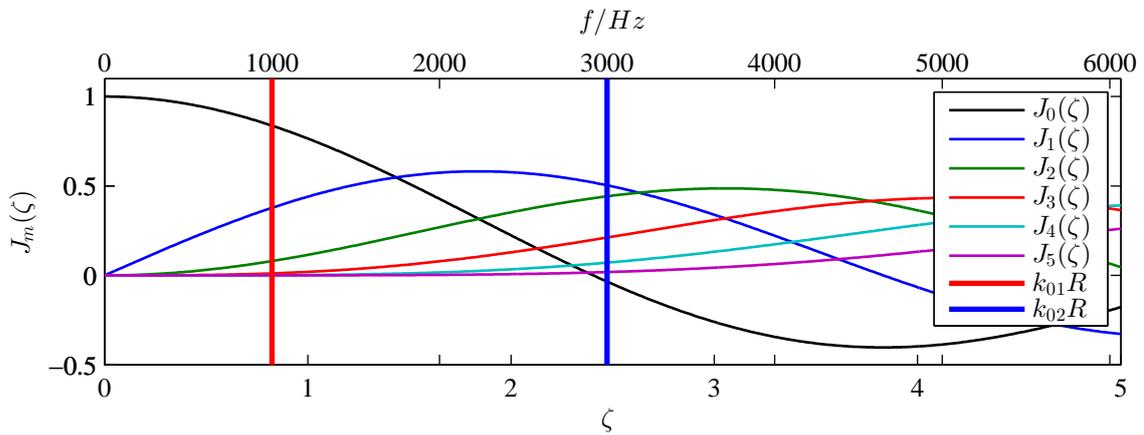
angeben werden. Für  $b^c(\zeta, \varphi)$  ergibt sich folglich:

$$b^c(\zeta, \varphi) = \sum_{m=-\infty}^{\infty} j^m J_m(\zeta) e^{jm\varphi} \quad (5.21)$$

Die Besselfunktion erster Art bestimmt somit, wie stark ein Phasen-Modus die Gesamtrichtcharakteristik beeinflusst. Dabei sei an dieser Stelle nochmals darauf hingewiesen, dass  $0 \leq \zeta \leq k_0 R$  gilt. Untersucht man, welche Funktionswerte die Besselfunktionen in diesem Wertebereich annehmen, so ergibt sich die Daumenregel [6], dass alle Phasenmoden  $|m| \geq M$  mit

$$M \approx k_0 R \quad (5.22)$$

vernachlässigt werden können. Verdeutlicht werden kann dies an einem Beispiel: In Bild 5.2 sind die Besselfunktionen erster Art bis  $|m| < 6$  dargestellt. Weiter sind  $k_{01}R = 2\pi f_1/\nu R$  und  $k_{02}R = 2\pi f_2/\nu R$  mit  $R = 0,045$  m,  $f_1 = 1$  kHz und  $f_2 = 3$  kHz eingezeichnet.



**Bild 5.2:** Besselfunktionen

Diese Funktionsstellen zeigen den sichtbaren Bereich für  $M = 2$  bzw.  $M = 3$  auf, da die Daumenregel 5.22 für  $M$  den Wert  $0,82 \approx 1$  bzw.  $2,72 \approx 3$  ausgibt. Man erkennt, dass alle Besselfunktionen höherer Ordnung ( $m > M$ ) stets einen kleinen Funktionswert im sichtbaren

Bereich aufweisen. Ebenfalls ist im Bild 5.2 die Frequenzabhängigkeit verdeutlicht. Die obere x-Achse gibt den maximalen Aussteuerungsbereich der angegebenen Frequenz bei einem Radius von  $R = 0,045$  m an.

## 5.2.2 Diskrete zirkulare Mikrofonarrays

Zuvor ist die Annahme getroffen worden, dass ein kontinuierliches Mikrofonarray vorliegt. Lässt man diese Annahme fallen, so ist  $\gamma$  nicht kontinuierlich, sondern diskret. In Analogie zu Gleichung 5.16 gilt für  $N_{mic}$  Mikrofone:

$$b_m^s(\zeta, \varphi) = \frac{1}{N_{mic}} \sum_{i=1}^{N_{mic}} e^{jm\gamma_i} e^{j\zeta \cos(\varphi - \gamma_i)} \quad (5.23)$$

Diese Gleichung kann auch als Skalarprodukt geschrieben werden. Definiert man  $\omega_{a,m}$  zu

$$\omega_{a,m}^s \dagger = \frac{1}{N_{mic}} [e^{jm\gamma_1}, e^{jm\gamma_2}, \dots, e^{jm\gamma_{N_{mic}}}] \quad (5.24)$$

folgt mit Gleichung 5.10:

$$b_m^s(\zeta, \varphi) = \omega_{a,m}^s \dagger \cdot \mathbf{a}_{UCA}(\zeta, \varphi) \quad (5.25)$$

Drückt man  $b_m^s(\zeta, \varphi)$  ebenfalls mit Hilfe von Besselfunktionen aus, so ergibt sich entsprechend [39, S. 170] folgender Ausdruck:

$$b_m^s(\zeta, \varphi) = j^m J_m(\zeta) e^{jm\varphi} + \sum_{q=1}^{\infty} \left( j^g J_g(\zeta) e^{-jg\varphi} + j^h J_h(\zeta) e^{-jh\varphi} \right) \quad (5.26)$$

$$= b_m^c(\zeta, \varphi) + r_m(\zeta, \varphi, N_{mic}) \quad (5.27)$$

Dabei gilt  $g = N_{mic}q - m$  und  $h = N_{mic}q + m$ . Der erste Summand entspricht  $b_m^c(\zeta, \varphi)$  (s. Gleichung 5.20). Der zweite Summand stellt einen Rest dar und wird als Aliasfehler bezeichnet. Damit der erste Summand dominant ist, muss die Bedingung  $N_{mic} > 2|m|$  erfüllt sein. Ist  $M$  der größte erregte Phasen-Modus, so ergibt sich die wichtige Bedingung

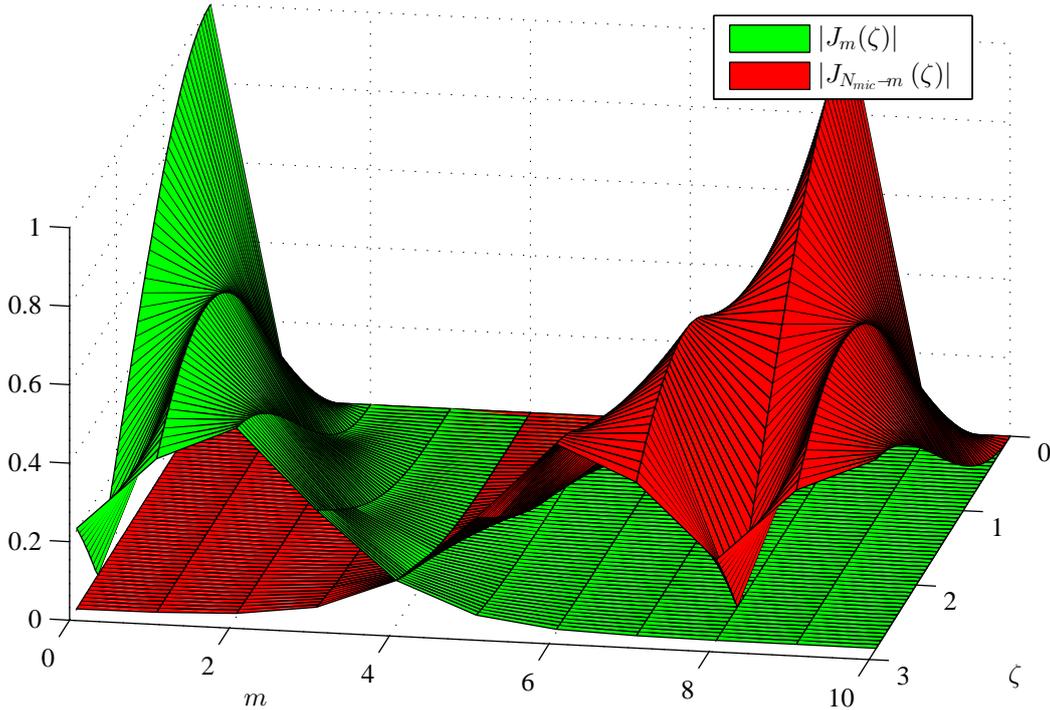
$$M < \frac{N_{mic}}{2} \quad (5.28)$$

für  $M$ . Mit  $M = k_0 R$  (s. Gleichung 5.22) führt dies zu dem Zusammenhang

$$\frac{2\pi R}{N_{mic}} < \frac{\lambda}{2} \quad (5.29)$$

der in Analogie zum ULA die Bedingung für das Vermeiden des räumlichen Alias-Effekts vorgibt. Weiterhin bleibt festzuhalten, dass insgesamt  $N_m = 2M + 1$  Phasen-Moden aufgelöst werden können, da auch stets der negative Phasen-Modus berücksichtigt werden kann.

In Bild 5.3 ist der Betrag von  $b_m^c(\zeta, \varphi)$  gegen den Betrag des Hauptrestterms  $|J_{N_{mic}-m}(\zeta)|$  über  $m$  und  $\zeta$ , wobei  $R = 0.045$  m,  $f < 3,5$  kHz und  $N_{mic} = 8$  angenommen worden ist, aufgetragen.



**Bild 5.3:** Vergleich von Nutz- und erstem Restterm über  $m$  und  $\zeta$

Deutlich ist zu erkennen, dass, solange Gleichung 5.28 erfüllt ist, der Betrag des Nutzterms einen deutlich höheren Wert als der Betrag des ersten Restterms aufweist. Die Restterme können somit vernachlässigt werden. In der Literatur wird für praktische Anwendungen die Formel

$$N_{mic} \geq 2M + x \text{ mit } x \geq 1 \quad (5.30)$$

angegeben, wobei nach [8]  $x = 3$ , nach [13]  $x = 5$  und nach [12]  $x = 6$  erfüllt sein sollte. Allerdings sei an dieser Stelle darauf hingewiesen, dass bei Einhalten dieser Bedingung mit nahezu keinem Aliasfehler mehr zu rechnen ist.

Um auch den Effekt der Phasen zu berücksichtigen, werden die Realteile der Verhältnisse der entsprechenden Restterme und Richtcharakteristik bei einem kontinuierlichen Array je

Phasen-Modus über die Einfallswinkel  $\varphi$  und  $\theta$  aufgetragen. Die entsprechenden Ergebnisse sind mit denselben Parametern bei einer Frequenz von  $f = 3,5$  kHz (schwarze Achse) und  $f = 1$  kHz (blaue Achse) in Bild 5.4 dargestellt. Ein besonderes Augenmerk sei hierbei auf die Skalierung der Achsen gelegt.

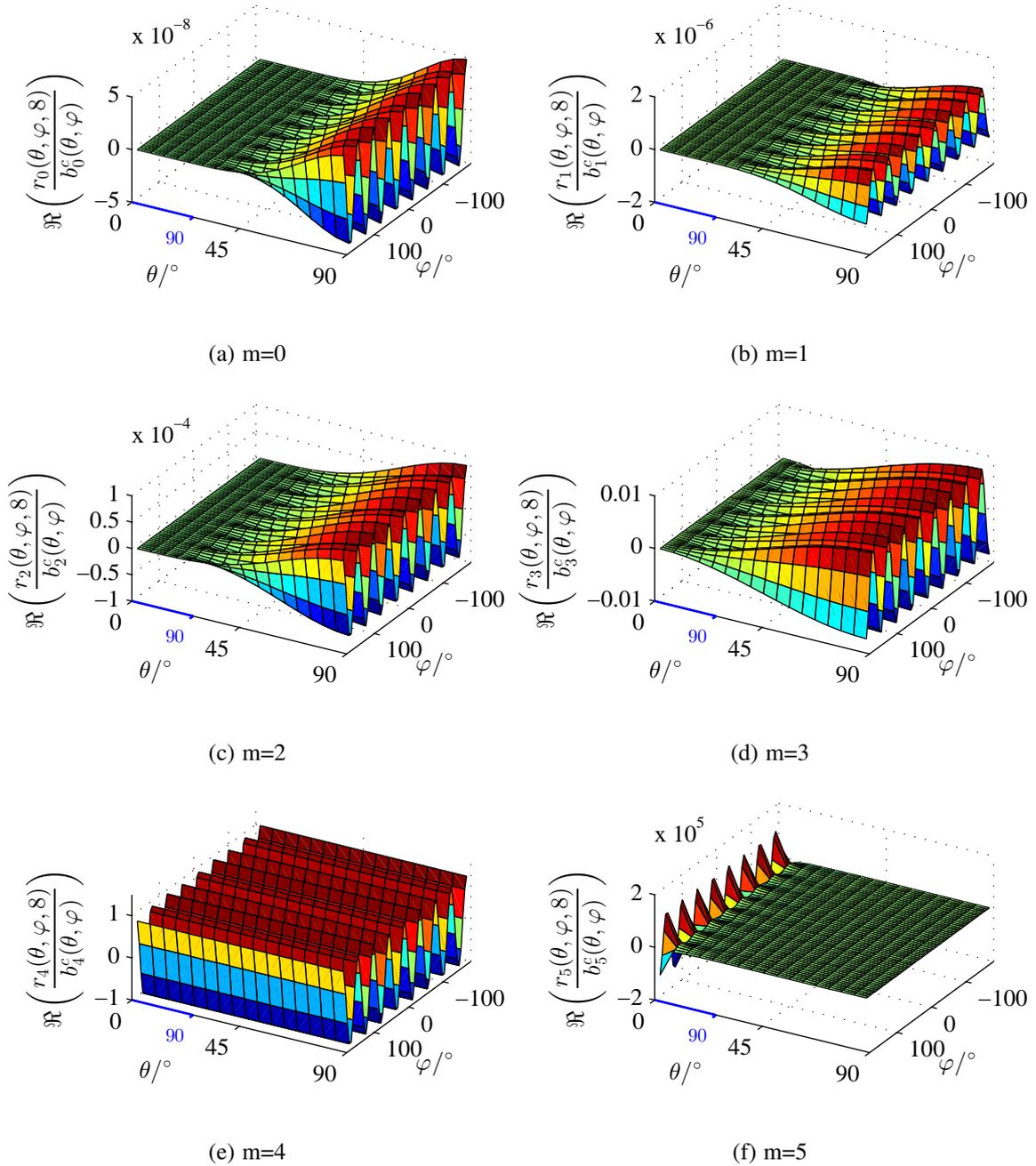
Die Ergebnisse bestätigen die vorherigen Ergebnisse. Der Anteil der Restterme steigt mit jedem betrachteten Phasen-Modus an. Solange Gleichung 5.28 erfüllt ist, ist dieser jedoch vernachlässigbar. Weiter zeigt sich eine Periodizität über dem Azimutwinkel  $\varphi$ .

Bei der Realisierung von Algorithmen auf Basis eines diskreten zirkularen Arrays ist aus diesem Grund mit periodischen Fehlern zu rechnen.

Zusammenfassend lässt sich folgern, dass, solange Gleichung 5.28 eingehalten wird, folgende Gleichung angenommen werden kann:

$$b_m^s(\zeta, \varphi) = \omega_{a,m}^s \dagger \cdot \mathbf{a}(\zeta, \varphi) \approx b_m^c(\zeta, \varphi) = j^m J_m(\zeta) e^{jm\varphi} = j^{|m|} J_{|m|}(\zeta) e^{jm\varphi}, \quad |m| \leq M \quad (5.31)$$

Wobei die Beziehungen  $J_{-m}(x) = (-1)^m J_m(x)$  (vgl. [32, S. 212]) und  $j^{-1} = -j$  genutzt werden.



**Bild 5.4:** Realteil des Verhältnisses des Restterms und der Richtcharakteristik bei einem kontinuierlichen Array je Phasen-Modus

### 5.2.3 Beamformer

Wie bereits in Abschnitt 5.1 gezeigt, weist die Richtungsmatrix von zirkularen Arrays keine Vandermonde-Struktur auf. Um diese Struktur trotzdem zu erreichen, wird die Richtungsmatrix mittels Beamformer im Phasen-Moden-Raum transformiert. In der Literatur werden drei verschiedene Beamformer verwendet, die hier kurz vorgestellt werden sollen. Die Ausführungen lehnen sich an [13, S. 41ff] an.

Die Beamformer  $\mathbf{B}_g^\dagger$ ,  $\mathbf{B}_r^\dagger$  und  $\mathbf{B}_U^\dagger$  transformieren den Richtungsvektor  $\mathbf{a}(\zeta, \varphi)$  nach  $\mathbf{a}_g(\zeta, \varphi)$ ,  $\mathbf{a}_r(\zeta, \varphi)$  und  $\mathbf{a}_U(\zeta, \varphi)$ . Die Indizes stehen dabei für gerade, reell und UCA-ESPRIT. Alle drei Beamformer sind unitär und beruhen auf der orthogonalen Matrix  $\mathbf{W} \in \mathbb{C}^{N_m \times N_{mic}}$ , die wiederum auf  $\omega_{a,m}^s$  aus Gleichung 5.24 beruht.

$$\mathbf{W}^\dagger = \sqrt{N_{mic}} \begin{bmatrix} \omega_{a,-M}^s \\ \vdots \\ \omega_{a,-1}^s \\ \omega_{a,0}^s \\ \omega_{a,1}^s \\ \vdots \\ \omega_{a,M}^s \end{bmatrix}^\dagger = \frac{1}{\sqrt{N_{mic}}} \begin{bmatrix} e^{-jM\gamma_1} & e^{-jM\gamma_2} & \dots & e^{-jM\gamma_{N_{mic}}} \\ \vdots & \vdots & & \vdots \\ e^{-j\gamma_1} & e^{-j\gamma_2} & \dots & e^{-j\gamma_{N_{mic}}} \\ 1 & 1 & \dots & 1 \\ e^{j\gamma_1} & e^{j\gamma_2} & \dots & e^{j\gamma_{N_{mic}}} \\ \vdots & \vdots & & \vdots \\ e^{jM\gamma_1} & e^{jM\gamma_2} & \dots & e^{jM\gamma_{N_{mic}}} \end{bmatrix} \quad (5.32)$$

Berücksichtigt man für  $\gamma_i$  die Definition aus 5.3, so folgt:

$$\mathbf{W}^\dagger = \frac{1}{\sqrt{N_{mic}}} \begin{bmatrix} 1 & e^{-jM\Delta\gamma} & \dots & e^{-jM\Delta\gamma(N_{mic}-1)} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-j\Delta\gamma} & \dots & e^{-j\Delta\gamma(N_{mic}-1)} \\ 1 & 1 & \dots & 1 \\ 1 & e^{j\Delta\gamma} & \dots & e^{j\Delta\gamma(N_{mic}-1)} \\ \vdots & \vdots & & \vdots \\ 1 & e^{jM\Delta\gamma} & \dots & e^{jM\Delta\gamma(N_{mic}-1)} \end{bmatrix} \quad (5.33)$$

Diese Strukturierung der Matrix entspricht einer IDFT-Matrix.



ausgedrückt werden. Dabei gilt für  $\tilde{\mathbf{I}}$ :

$$\tilde{\mathbf{I}} = \begin{bmatrix} 0 & & 1 & 1 \\ & \ddots & & \\ 1 & & 0 & \end{bmatrix} \quad (5.41)$$

Diese Beziehung kann genutzt werden, um einen reellwertigen Beamformer  $\mathbf{B}_r^\dagger$  zu generieren. Dazu wird die Matrix  $\mathbf{V}$  zu

$$\mathbf{V} = \frac{1}{N_m} \left[ \mathbf{v}(\alpha_{-M}) \vdots \dots \vdots \mathbf{v}(\alpha_0) \vdots \dots \vdots \mathbf{v}(\alpha_M) \right] \quad (5.42)$$

mit

$$\alpha_i = 2\pi \frac{i}{N_m} \quad (5.43)$$

definiert. Der Beamformer  $\mathbf{B}_r^\dagger$  ergibt sich wie folgt:

$$\mathbf{B}_r^\dagger = \mathbf{V}^\dagger \mathbf{B}_g^\dagger \quad (5.44)$$

Für  $\mathbf{a}_r(\zeta, \varphi)$  ergibt sich entsprechend:

$$\mathbf{a}_r(\zeta, \varphi) = \mathbf{V}^\dagger \mathbf{a}_g(\zeta, \varphi) \quad (5.45)$$

$$= \sqrt{\frac{N_{mic}}{N_M}} \begin{bmatrix} g(\zeta, \varphi - \alpha_{-M}) \\ \vdots \\ g(\zeta, \varphi - \alpha_{-1}) \\ g(\zeta, \varphi) \\ g(\zeta, \varphi - \alpha_1) \\ \vdots \\ g(\zeta, \varphi - \alpha_M) \end{bmatrix} \quad (5.46)$$

wobei

$$g(\zeta, \varphi) = \left[ J_0(\zeta) + 2 \sum_{m=1}^M J_m(\zeta) \cos(m\varphi) \right] \quad (5.47)$$

gilt. Durch diese Wahl des Beamformers werden mehrere, rotierte Summen der Elemente von  $\mathbf{a}_g(\zeta, \varphi)$  betrachtet.

Weiter führt dieser Beamformer zu einem deutlich geringeren numerischen Aufwand, da anschließende Operationen reellwertig durchgeführt werden können.



werden kann. Für die Richtungsvektoren folgt:

$$\mathbf{a}_U(\zeta, \varphi) = \mathbf{C}_o \mathbf{V} \mathbf{a}_r(\zeta, \varphi) \quad (5.55)$$

Dieser Zusammenhang ermöglicht es mit reellwertigen Eigenwertzerlegungen bei der Verwendung des UCA-ESPRIT-Algorithmus zu arbeiten.

Eine wichtige Eigenschaft, die alle drei vorgestellten Beamformer gemein haben, ist, dass sie unitär sind. Es gilt:

$$\mathbf{B}_U^\dagger \mathbf{B}_U = \mathbf{B}_r^\dagger \mathbf{B}_r = \mathbf{B}_g^\dagger \mathbf{B}_g = \mathbf{I}_{N_m} \quad (5.56)$$

### 5.3 UCA-ESPRIT-Algorithmus

Der UCA-ESPRIT-Algorithmus unterscheidet sich prinzipiell stark vom ESPRIT-Algorithmus. Er basiert nicht auf der bekannten Arrayverschiebung, sondern nutzt viel mehr die rekursive Beziehung der Besselfunktionen aus. Allerdings ähneln die notwendigen Schritte zur Berechnung der Ergebnisse denen des TLS-ESPRIT-Algorithmus. Auf dieser Gemeinsamkeit gründet die Namensgebung.

Ausgehend von der Verwendung des Beamformers  $\mathbf{B}_U^\dagger$  und der Gleichung 5.52

$$\mathbf{a}_U(\zeta, \varphi) = \begin{bmatrix} J_{-M}(\zeta) e^{-jM\varphi} \\ \vdots \\ J_{-1}(\zeta) e^{-j\varphi} \\ J_0(\zeta) \\ J_1(\zeta) e^{j\varphi} \\ \vdots \\ J_M(\zeta) e^{jM\varphi} \end{bmatrix}$$

lassen sich die grundsätzlichen Beziehungen zur Berechnung herleiten. Um die rekursiven Beziehungen der Besselfunktionen ausnutzen zu können, werden drei Untervektoren generiert, die anstatt  $N_m$  nur  $N_m^s = N_m - 2$  Elemente enthalten. Es gilt

$$\mathbf{a}_{U,i}(\zeta, \varphi) = \Delta_i \cdot \mathbf{a}_U(\zeta, \varphi) \text{ mit } i = -1, 0, 1 \quad (5.57)$$

mit

$$\Delta_{-1} = \begin{bmatrix} \mathbf{I}_{N_m^s \times N_m^s} & \vdots & \mathbf{0}_{N_m^s \times 2} \end{bmatrix} \quad (5.58)$$

$$\Delta_0 = \begin{bmatrix} \mathbf{0}_{N_m^s \times 1} & \mathbf{I}_{N_m^s \times N_m^s} & \mathbf{0}_{N_m^s \times 1} \end{bmatrix} \quad (5.59)$$

$$\Delta_1 = \begin{bmatrix} \mathbf{0}_{N_m^s \times 2} & \vdots & \mathbf{I}_{N_m^s \times N_m^s} \end{bmatrix}. \quad (5.60)$$

Es folgt:

$$\mathbf{a}_{U,-1}(\zeta, \varphi) = \begin{bmatrix} J_{-M}(\zeta)e^{-jM\varphi} \\ \vdots \\ J_{-1}(\zeta)e^{-j\varphi} \\ J_0(\zeta) \\ J_1(\zeta)e^{j\varphi} \\ \vdots \\ J_{M-2}(\zeta)e^{j(M-2)\varphi} \end{bmatrix} = e^{-j\varphi} \begin{bmatrix} J_{-M}(\zeta)e^{-j(M-1)\varphi} \\ \vdots \\ J_{-1}(\zeta) \\ J_0(\zeta)e^{j\varphi} \\ J_1(\zeta)e^{j2\varphi} \\ \vdots \\ J_{M-2}(\zeta)e^{j(M-1)\varphi} \end{bmatrix} \quad (5.61)$$

$$\mathbf{a}_{U,0}(\zeta, \varphi) = \begin{bmatrix} J_{-(M-1)}(\zeta)e^{-j(M-1)\varphi} \\ \vdots \\ J_{-1}(\zeta)e^{-j\varphi} \\ J_0(\zeta) \\ J_1(\zeta)e^{j\varphi} \\ \vdots \\ J_{M-1}(\zeta)e^{j(M-1)\varphi} \end{bmatrix} \quad (5.62)$$

$$\mathbf{a}_{U,1}(\zeta, \varphi) = \begin{bmatrix} J_{-(M-2)}(\zeta)e^{-j(M-2)\varphi} \\ \vdots \\ J_{-1}(\zeta)e^{-j\varphi} \\ J_0(\zeta) \\ J_1(\zeta)e^{j\varphi} \\ \vdots \\ J_M(\zeta)e^{jM\varphi} \end{bmatrix} = e^{j\varphi} \begin{bmatrix} J_{-(M-2)}(\zeta)e^{-j(M-1)\varphi} \\ \vdots \\ J_{-1}(\zeta)e^{-j2\varphi} \\ J_0(\zeta)e^{-j\varphi} \\ J_1(\zeta) \\ \vdots \\ J_M(\zeta)e^{j(M-1)\varphi} \end{bmatrix} \quad (5.63)$$

Werden lediglich die Phasen der einzelnen komplexen Vektoren betrachtet, so lässt sich erkennen, dass die Vektoren  $\mathbf{a}_{U,0}(\zeta, \varphi)$ ,  $e^{-j\varphi}\mathbf{a}_{U,1}(\zeta, \varphi)$  und  $e^{j\varphi}\mathbf{a}_{U,-1}(\zeta, \varphi)$  die gleichen Phasen aufweisen. Verwendet man die rekursive Beziehung (vgl. [32, S. 215])

$$J_{m-1}(x) + J_{m+1}(x) = \frac{2m}{x}J_m(x) \quad (5.64)$$

kann folgende Beziehung für die Vektoren aufgestellt werden:

$$e^{j\varphi} \mathbf{a}_{U,-1}(\zeta, \varphi) + e^{-j\varphi} \mathbf{a}_{U,1}(\zeta, \varphi) = \frac{2}{\zeta} \mathbf{M} \mathbf{a}_{U,0}(\zeta, \varphi) \quad (5.65)$$

Wobei  $\mathbf{M}$  wie folgt definiert ist:

$$\mathbf{M} = \text{diag} \{ -(M-1), \dots, -1, 0, 1, \dots, M-1 \} \quad (5.66)$$

Dies kann mit  $\zeta = k_0 R \sin(\theta)$  in die Form

$$\mu \mathbf{a}_{U,-1}(\zeta, \varphi) + \mu^* \mathbf{a}_{U,1}(\zeta, \varphi) = \Gamma \mathbf{a}_{U,0}(\zeta, \varphi) \quad (5.67)$$

überführt werden. Wobei

$$\Gamma = \frac{\lambda}{\pi R} \mathbf{M} \quad (5.68)$$

$$\mu = \sin(\theta) e^{j\varphi} \quad (5.69)$$

gilt. Dies kann auf mehrere Quellen erweitert werden. Dazu werden die transformierte Richtungsmatrix

$$\mathbf{A}_U = \left[ \mathbf{a}_U(\zeta_1, \varphi_1) \ : \ \mathbf{a}_U(\zeta_2, \varphi_2) \ : \ \dots \ : \ \mathbf{a}_U(\zeta_{N_S}, \varphi_{N_S}) \right] \quad (5.70)$$

und die entsprechenden Untermatrizen  $\mathbf{A}_{U,i} = \Delta_i \mathbf{A}_U$  gebildet. Dies führt zu der Gleichung

$$\mathbf{A}_{U,-1} \Phi + \mathbf{A}_{U,+1} \Phi^* = \Gamma \mathbf{A}_{U,0} \quad (5.71)$$

mit

$$\Phi = \text{diag} \{ \mu_1, \dots, \mu_{N_S} \}. \quad (5.72)$$

Dies ist die wesentliche Gleichung des UCA-ESPRIT-Algorithmus. Aufbauend auf dieser Gleichung sind nachfolgende Algorithmusschritte angelegt.

Um jedoch zunächst den Signalraum in Beziehung zur transformierten Richtungsmatrix zu setzen, sei das Signalmodell aus Kapitel 3 betrachtet. Da jedoch ein UCA anstatt einem ULA verwendet wird, muss die Richtungsmatrix  $\mathbf{A}_{ULA}$  aus Gleichung 3.21 entsprechend durch

$$\mathbf{A}_{UCA} = \left[ \mathbf{a}_{UCA}(\zeta_1, \varphi_1) \ : \ \mathbf{a}_{UCA}(\zeta_2, \varphi_2) \ : \ \dots \ : \ \mathbf{a}_{UCA}(\zeta_{N_S}, \varphi_{N_S}) \right] \quad (5.73)$$

ersetzt werden. Es folgt:

$$\mathbf{y}_{UCA}[n] = \mathbf{A}_{UCA} \cdot \mathbf{x}[n] + \mathbf{n}[n] \quad (5.74)$$

Um eine Vandermonde-Struktur der Richtungsmatrix zu erzielen, wird der Beamformer  $\mathbf{B}_U^\dagger$  angewendet:

$$\mathbf{B}_U^\dagger \mathbf{y}_{UCA}[n] = \mathbf{B}_U^\dagger \mathbf{A}_{UCA} \cdot \mathbf{x}[n] + \mathbf{B}_U^\dagger \mathbf{n}[n] \quad (5.75)$$

Da der Beamformer unitär ist, ergibt sich für die Autokorrelationsmatrix in Analogie zu Abschnitt 3.3:

$$\mathbf{B}_U^\dagger \mathbf{R}_{yy,UCA} \mathbf{B}_U = \mathbf{B}_U^\dagger \mathbf{A}_{UCA} \mathbf{R}_{xx} \mathbf{A}_{UCA}^\dagger \mathbf{B}_U + \sigma^2 \cdot \mathbf{I}_{N_{mic}} \quad (5.76)$$

$$\mathbf{B}_U^\dagger \mathbf{R}_{yy,UCA} \mathbf{B}_U = \mathbf{A}_U \mathbf{R}_{xx} \mathbf{A}_U^\dagger + \sigma^2 \cdot \mathbf{I}_{N_{mic}} \quad (5.77)$$

Entsprechend den Ausführungen in den Abschnitten 3.4.1 und 3.4.2, spannen die transformierte Richtungsmatrix  $\mathbf{A}_U$  und der bestimmte Signalunterraum  $\mathbf{E}_{x,U}$  dieselben lineare Hülle auf:

$$\text{span} \{ \mathbf{A}_U \} = \text{span} \{ \mathbf{E}_{x,U} \} \quad (5.78)$$

Ebenfalls ist es möglich die Matrix  $\mathbf{E}_{x,U}$  über eine reellwertige Eigenwertzerlegung zu bestimmen. Dazu wird anstatt des Beamformers  $\mathbf{B}_U^\dagger$  der Beamformer  $\mathbf{B}_r^\dagger$  verwendet. Für den Signalunterraum  $\mathbf{E}_{x,r}$  gilt ebenfalls:

$$\text{span} \{ \mathbf{A}_r \} = \text{span} \{ \mathbf{E}_{x,r} \} \quad (5.79)$$

Aufgrund dieses Zusammenhanges existiert eine quadratische, reguläre Matrix  $\mathbf{T}_U \in \mathbb{R}^{N_s \times N_s}$  für die gilt:

$$\mathbf{A}_r = \mathbf{E}_{x,r} \mathbf{T}_U^{-1} \quad (5.80)$$

Benutzt man den Zusammenhang 5.55, so folgt:

$$\mathbf{A}_U = \mathbf{C}_o \bar{\mathbf{I}} \mathbf{A}_r \quad (5.81)$$

$$= \mathbf{C}_o \bar{\mathbf{I}} \mathbf{E}_{x,r} \mathbf{T}_U^{-1} \quad (5.82)$$

$$= \mathbf{E}_{x,U} \mathbf{T}_U^{-1} \quad (5.83)$$

Aufgrund dieses Zusammenhanges kann in Gleichung 5.71 mit

$$\mathbf{E}_{x,U,i} = \Delta_i \cdot \mathbf{E}_{x,U} \text{ mit } i = -1, 0, 1 \quad (5.84)$$

$\mathbf{A}_{x,U,i}$  durch  $\mathbf{E}_{x,U,i}\mathbf{T}_U^{-1}$  ersetzt werden:

$$\mathbf{E}_{x,U,-1}\mathbf{T}_U^{-1}\Phi + \mathbf{E}_{x,U,+1}\mathbf{T}_U^{-1}\Phi^* = \Gamma\mathbf{E}_{x,U,0}\mathbf{T}_U^{-1} \quad (5.85)$$

Mit der Multiplikation von rechts mit  $\mathbf{T}_U$  und der Definition

$$\Psi = \mathbf{T}_U^{-1}\Phi\mathbf{T}_U \quad (5.86)$$

ergibt sich der Zusammenhang

$$\mathbf{E}_{x,U,-1}\Psi + \mathbf{E}_{x,U,+1}\Psi^* = \Gamma\mathbf{E}_{x,U,0} \quad (5.87)$$

,da  $\mathbf{T}_U$  reellwertig ist. In kompakter Matrixschreibweise folgt:

$$\begin{bmatrix} \mathbf{E}_{x,U,-1} & \vdots & \mathbf{E}_{x,U,+1} \end{bmatrix} \begin{bmatrix} \Psi \\ \Psi^* \end{bmatrix} = \Gamma\mathbf{E}_{x,U,0} \quad (5.88)$$

$$\underline{\mathbf{E}}\Psi = \Gamma\underline{\mathbf{E}}_{x,U,0} \quad (5.89)$$

So lange  $N_m^s > 2 \cdot N_S$  bzw.  $N_S < M$  gilt, ist diese Gleichung überbestimmt und es kann eine eindeutige Lösung für  $\Psi$  gefunden werden. Damit die Gleichung nicht unterbestimmt wird, können folglich maximal  $M - 1$  Quellen unterschieden werden.

Sobald  $\Psi$  bestimmt ist, kann aufgrund von Gleichung 5.86 mit Hilfe einer Eigenwertanalyse  $\Phi$  bestimmt werden (s. Anhang A.2). Diese Matrix führt unmittelbar zu den entsprechenden Richtungswinkeln, da die Eigenwerte die Struktur  $\mu_i = \sin(\theta_i)e^{j\phi_i}$  (s. Gleichung 5.69) haben.

In der Realität kann der Signalunterraum lediglich geschätzt werden. Folglich muss die Lösung von  $\underline{\Psi}$  - als  $\hat{\underline{\Psi}}_{LS}$  bezeichnet - aus Gleichung 5.89 mit Hilfe der Methode der kleinsten Quadrate bestimmt werden:

$$\min_{\underline{\Psi}} \left\| \hat{\underline{\mathbf{E}}}\underline{\Psi} - \Gamma\hat{\underline{\mathbf{E}}}_{x,U,0} \right\|_2^2 \quad (5.90)$$

Nach [13, S. 54, 105f.] kann gezeigt werden, dass dazu folgendes reellwertiges Gleichungssystem gelöst werden muss:

$$\begin{bmatrix} \Re(\mathbf{B} + \mathbf{C}) & \Im(\mathbf{C} - \mathbf{B}) \\ \Im(\mathbf{B} + \mathbf{C}) & \Re(\mathbf{B} - \mathbf{C}) \end{bmatrix} \begin{bmatrix} \Re\hat{\underline{\Psi}}_{LS} \\ \Im\hat{\underline{\Psi}}_{LS} \end{bmatrix} = \begin{bmatrix} \Re\mathbf{Q} \\ \Im\mathbf{Q} \end{bmatrix} \quad (5.91)$$

Dabei gilt für die Matrizen  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}$  und  $\hat{\Psi}_{LS}$ :

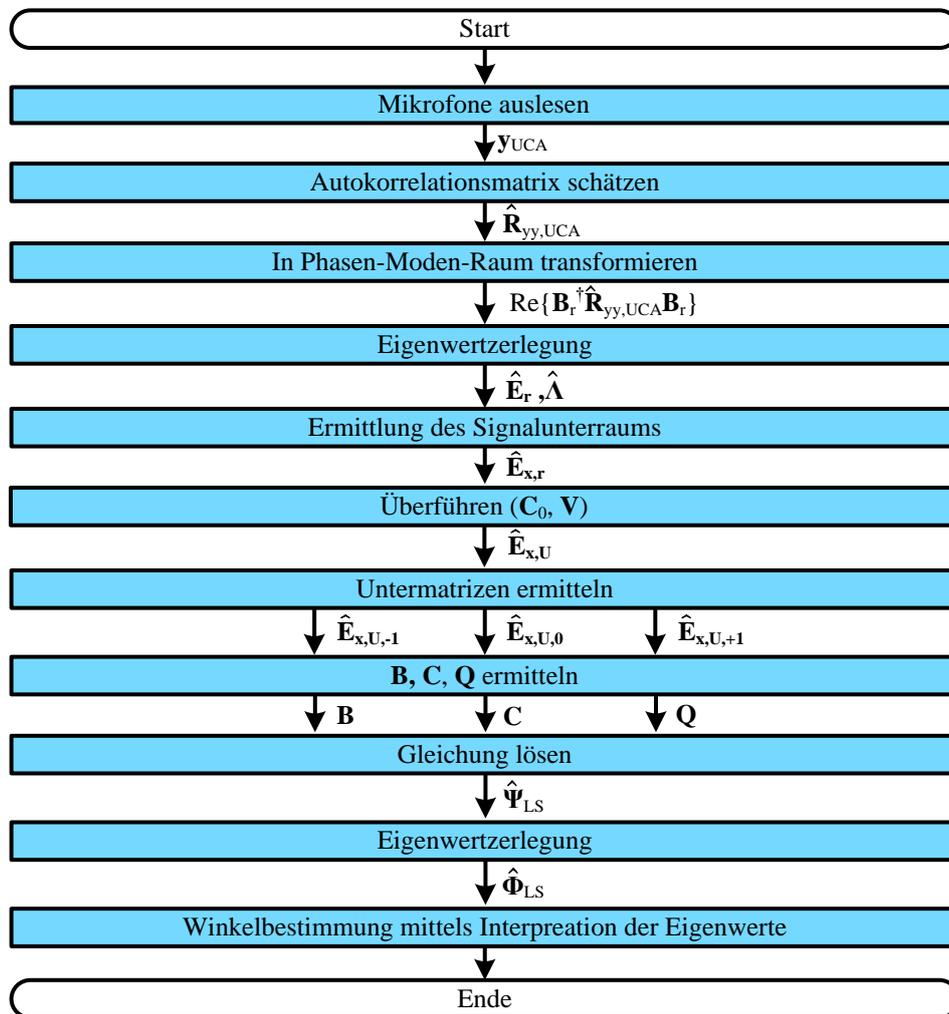
$$\mathbf{B} = \hat{\mathbf{E}}_{x,U,-1}^\dagger \hat{\mathbf{E}}_{x,U,-1} \tag{5.92}$$

$$\mathbf{C} = \hat{\mathbf{E}}_{x,U,-1}^\dagger \hat{\mathbf{E}}_{x,U,+1} \tag{5.93}$$

$$\mathbf{Q} = \hat{\mathbf{E}}_{x,U,-1}^\dagger \Gamma \hat{\mathbf{E}}_{x,U,0} \tag{5.94}$$

$$\hat{\Psi}_{LS} = \Re \hat{\Psi}_{LS} + j \cdot \Im \hat{\Psi}_{LS} \tag{5.95}$$

Anschließend können - entsprechend den vorherigen Überlegungen - die Winkel mit einer Eigenwertzerlegung und einer Interpretation der Eigenwerte (s. Gleichung 5.69) bestimmt werden. In Bild 5.5 ist der Ablauf des gesamten Algorithmus zusammengefasst:



**Bild 5.5:** UCA-ESPRIT-Algorithmus

## 5.4 Simulationen

Im vorherigen Abschnitt sind die theoretischen Grundlagen zum UCA-ESPRIT-Algorithmus vorgestellt worden. Um die beschriebenen Eigenschaften und Effekte in der Simulation für verschiedene Parameter nachzuweisen, werden entsprechende Simulationen mit schmalbandigen, komplexwertigen Signalen durchgeführt. Dabei wird von einem Array mit dem Radius  $R = 4,5$  cm mit  $N_{mic} = 8$  Mikrofonen ausgegangen.

Die zu verwendende Abtastfrequenz lässt sich gut mit der Formel 5.29 abschätzen. Für die gegebenen Werte, die den maximalen Phasen-Modus auf  $M = 3$  begrenzen, erhält man eine maximale Frequenz von  $f_{max} \approx 4,8$  kHz. Da dies lediglich eine Schätzung darstellt und der Korrekturfaktor aus Gleichung 5.30 noch nicht berücksichtigt ist, scheint eine Abtastfrequenz von  $f_s = 8$  kHz als eine gute Wahl.

Durch die endliche Anzahl der Mikrofonarrays hat sich der sogenannte Aliasfehler ergeben. Die aufgezeigte Systematik lässt die Hypothese zu, dass die Bestimmung der Winkel von einem systematischen Fehler überlagert ist. Dieser ist vom Argument der Besselfunktionen  $\zeta$  und somit von Frequenz und Elevationswinkel  $\theta$ , sowie vom Azimutwinkel  $\varphi$  abhängig.

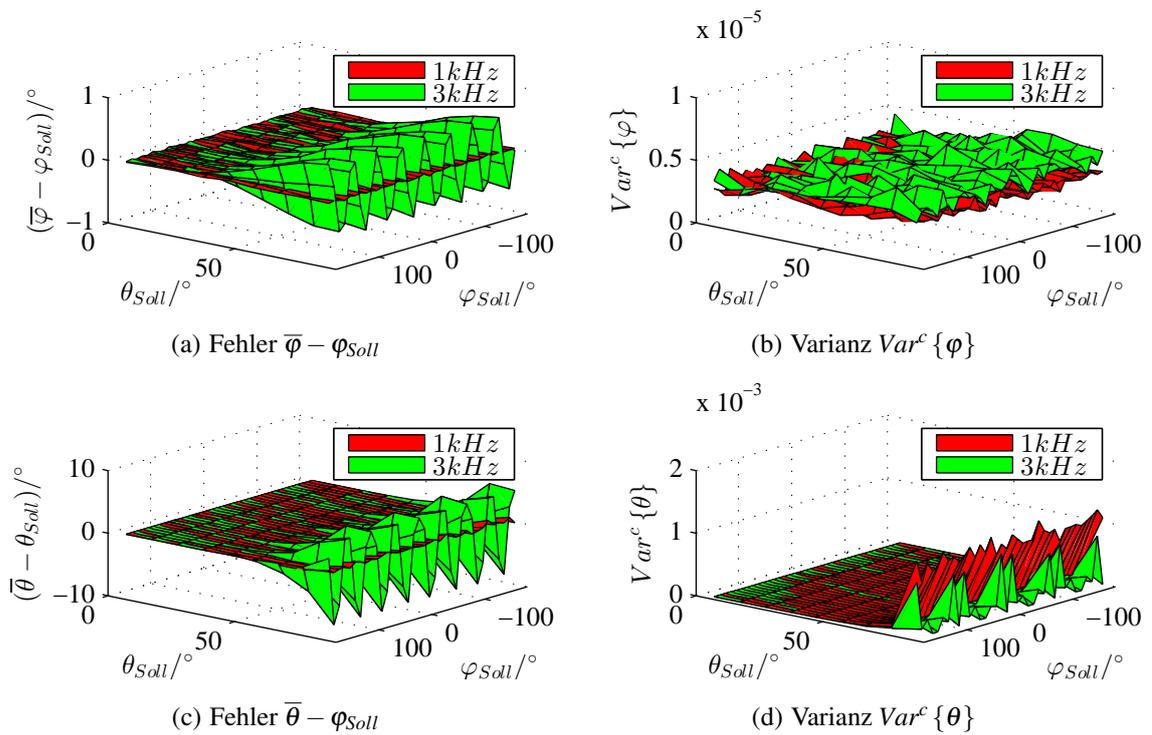
Um diese Hypothese zu überprüfen, werden Simulationen über nahezu den kompletten Wertebereich der Einfallswinkel durchgeführt. Dazu wird eine komplexe Schwingung verzögert und additiv mit weißem gaußschen Rauschen überlagert. Die Standardabweichung des Rauschens beträgt für alle nachfolgenden schmalbandigen Simulationen 0,2236. Dies entspricht bei einer komplexen Schwingung mit einer Amplitude von eins als Nutzsignal einen Signalrauschabstand (SNR) von 13 dB.

Die Theorie hat eine Abhängigkeit des Fehlers von der maximalen Anzahl der betrachteten Phasen-Moden aufgezeigt. Allerdings legt dieser Parameter auch die maximale Anzahl unterscheidbarer Quellen fest. Gleichungen 5.28 und 5.30 erlauben prinzipiell zwei Möglichkeiten:  $M = 2$  und  $M = 3$ . Um generell zwei Quellen unterscheiden zu können, sei  $M = 3$  gewählt.

Die Ergebnisse sind in den Bildern der Bildergruppe 5.6 für zwei Frequenzen dargestellt.

Die Ergebnisse zeigen die Abweichung des Mittelwerts vom Sollwert sowie die zirkulare Varianz. Bei näherer Betrachtung der Abweichung des Mittelwerts vom Sollwert, zeigt sich neben einem gewissen zufälligen Fehler auch eine deutliche Systematik für die höhere Frequenz. Der Fehler ist über den Azimutwinkel periodisch und steigt mit dem Elevationswinkel. Dies ist sowohl im Azimut- als auch im Elevationswinkel erkennbar. Da die beschriebene Systematik jedoch nur für die höhere Frequenz aus den Ergebnissen ableitbar ist, scheint die Abweichung frequenzabhängig zu sein. Eine Steigerung mit zunehmender Frequenz ist zu erwarten.

Die zirkulare Varianz weist ebenfalls eine Systematik auf. Für den Azimutwinkel ist ein konstantes Niveau erkennbar. Beim Elevationswinkel nimmt die Varianz hingegen mit dem Wert des Elevationswinkels deutlich zu. Auch die Periodizität über den Azimutwinkel ist

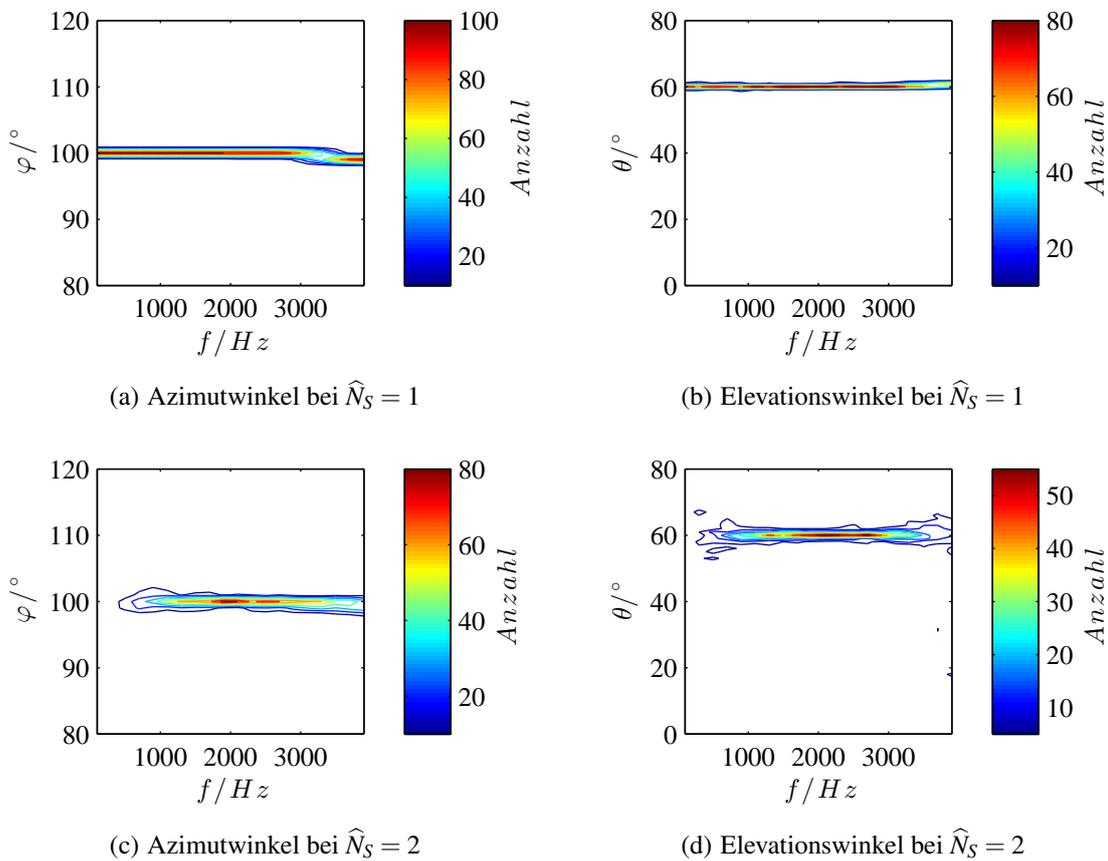


**Bild 5.6:** Fehler- und Varianzverläufe über  $\bar{\varphi}$  und  $\bar{\theta}$  (komplexwertige, schmalbandige Eingangssignale,  $M = 3$ )

ansatzweise erkennbar. Der Grund für den Anstieg der Varianz liegt in der Kennlinie zwischen dem Betrag des Eigenwertes und des Elevationswinkels (s. Abschnitt 6.4.4).

Somit bleibt festzuhalten, dass der systematische Aliasfehler, der aufgrund der Diskretisierung des Mikrofonarrays entsteht, und dessen Frequenzabhängigkeit sich auch im Endergebnis des Algorithmus widerspiegelt.

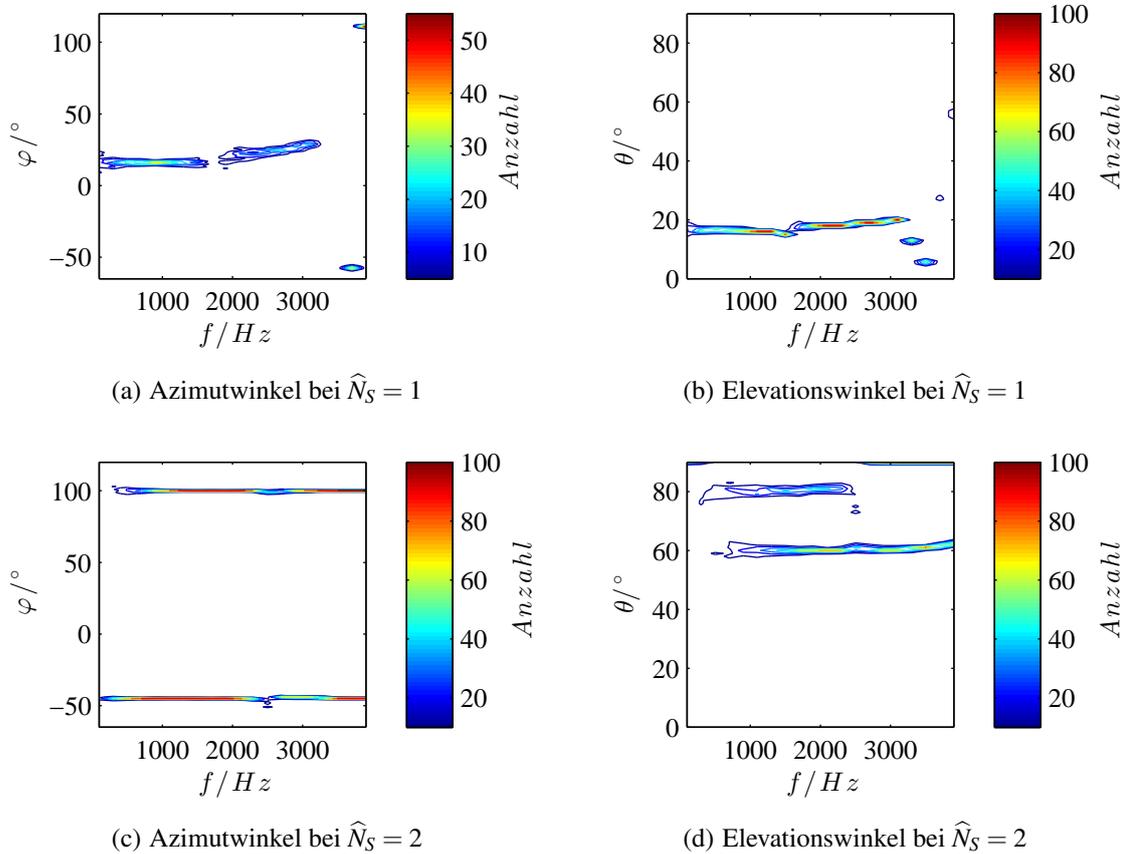
Um den Effekt der Wahl der Anzahl aktiver Quellen weiter zu untersuchen, werden Simulationen für einen festen Einfallswinkel durchgeführt. Dabei wird weiter zwischen einer und zwei aktiven Quellen unterschieden. Der SNR beträgt in Bezug auf weißes gaußsches Rauschen und einer komplexen Schwingung 13 dB. Der größte, betrachtete Phasen-Modus ist  $M = 3$ . Anschließend werden entsprechende Histogramme gebildet. Die Verläufe sind in der Bildergruppe 5.7 und 5.8 dargestellt.



**Bild 5.7:** Histogramm der Ergebnisse bei  $N_S = 1$  ( $\varphi_{Soll} = 100^\circ$ ,  $\theta_{Soll} = 60^\circ$ )

Die Auswertung der Ergebnisse führt zu folgenden Punkten: Falls eine aktive Quelle vorliegt, unterliegen die Ergebnisse unter der Annahme einer aktiven Quellen einer geringeren Streuung, als die Ergebnisse unter der Annahme zweier aktiver Quellen. Ebenfalls können

unter der Annahme von zwei aktiven Quellen lediglich die Ergebnisse im Frequenzfenster von 1 kHz bis 3 kHz als sinnvolle eingestuft werden.



**Bild 5.8:** Histogramm der Ergebnisse bei  $N_S = 2$  ( $\varphi_{Soll1} = 100^\circ$ ,  $\theta_{Soll1} = 60^\circ$ ;  $\varphi_{Soll2} = -45^\circ$ ,  $\theta_{Soll2} = 80^\circ$ )

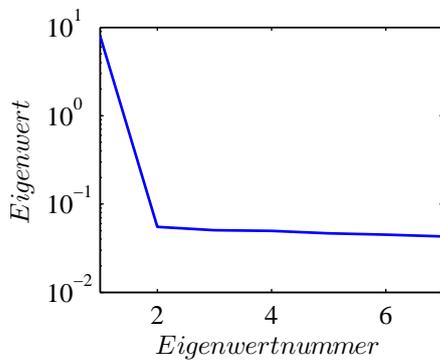
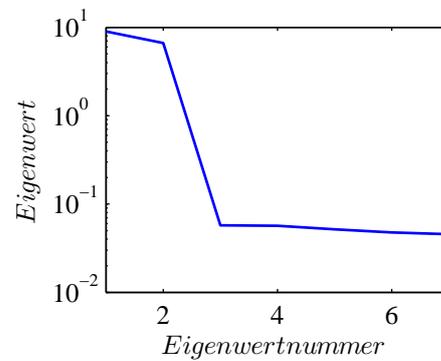
Sind jedoch zwei Quellen aktiv, so kehrt sich das Bild um. Falls die Annahme einer Quelle getroffen wird, führt dies zu einer Art Mittelung. Die Ergebnisse weichen stark vom Sollwert ab. Dieser Effekt tritt bei Annahme zweier Quellen nicht auf.

Weiter lässt sich feststellen, dass es insbesondere bei der Annahme einer Quelle zu einem Sprung in den Ergebnissen bei einer Frequenz von 3,5 kHz kommt. Dies lässt sich durch den Aliasfehler begründen.

Generell unterliegen die Ergebnisse des Elevationswinkels einer größeren Streuung als die Ergebnisse des Azimutwinkels. Dies lässt sich durch die Zusammensetzung des Argumentes der Besselfunktionen  $\zeta$  begründen. Für weitere Ausführungen sei auf 6.4.4 verwiesen.

Die Ergebnisse spiegeln wieder, dass es sich bei der Schätzung der Anzahl der Quellen um einen wichtigen Schritt im Algorithmus handelt. Wie bereits in der Theorie aufgezeigt, ist

dies prinzipiell durch die Analyse der Eigenwerte möglich. In der Bildergruppe 5.9 sind zwei Verläufe der Eigenwerte dargestellt. Dabei ist einmal eine und ein anderes Mal sind zwei Quellen aktiv.

(a)  $N_S = 1$ (b)  $N_S = 2$ 

**Bild 5.9:** Verlauf der Eigenwerte bei komplexwertigen, schmalbandigen Eingangssignalen

Deutlich ist zu erkennen, dass die Eigenwerte den nahezu idealen Verlauf der Theorie aufweisen. Die Anzahl der erhöhten Eigenwerte stimmt mit der Anzahl der aktiven Quellen überein und alle weiteren Eigenwerte haben nahezu ein konstantes Niveau. Somit ist es prinzipiell möglich, die Anzahl der Quellen korrekt zu bestimmen.

## 6 Konzepterarbeitung

Im vorherigen Kapitel sind die theoretischen Grundlagen gelegt worden. Aufbauend auf diesen Grundlagen soll in diesem Kapitel ein Konzept zur Umsetzung der Lokalisierung erfolgen.

Dazu werden zunächst Sprachsignale näher betrachtet, um aufbauend auf diesen Erkenntnissen einen Konzeptansatz zu formulieren. Anschließend werden notwendige Vorverarbeitungsschritte diskutiert. Der UCA-ESPRIT-Algorithmus wird in Hinblick auf die Verwendung von Sprachsignalen und Optimierungsmöglichkeiten vor allem unter Berücksichtigung der gewollten Echtzeitfähigkeit hinterfragt. Dazu werden umfangreiche *MATLAB*-Simulationen mit generierten Signalen (s. Abschnitt 4.7) als auch mit Aufnahmen durchgeführt. Aufbauend auf diesen Erkenntnissen wird eine Ergebnisaufbereitung diskutiert. Zum Schluss wird das entwickelte Konzept zusammengefasst.

### 6.1 Sprachsignale

Stimmhafte Sprachsignale lassen sich als statische Signale beschreiben. Sie sind „[...] quasi-periodisch und weisen folglich eine Grundwelle und Oberwellen auf [...]“ [22, S. 52f]. Die Frequenz dieser Grundwelle wird als Grundfrequenz bezeichnet. Ihr Mittelwert wird zumeist als Stimmlage und ihre zeitliche Variation als Sprachmelodie bezeichnet. Nach [38, S. 9] liegt diese Grundfrequenz bei Männern zwischen 50 Hz und 250 Hz und bei Frauen zwischen 120 Hz und 500 Hz. Weiterhin treten weitere lokale Maxima auf. Diese werden als Formante bezeichnet.

Durch die Variation der Frequenzen und die Quasi-Periodizität ist ein Sprachsignal im Allgemeinen nicht stationär. Allerdings kann für eine bestimmte Signaldauer - zumeist die Dauer einer Peridiozität - Stationarität angenommen werden. Die Wahl dieser Länge ist stets ein Kompromiss zwischen guter Statistik und Stationarität. In [7, S. 31ff.] wird ein Wert für die Blocklänge von 15 ms - 20 ms angegeben. In [37, S. 12] wird eine Dauer von 20 ms - 50 ms angenommen. Die Dauer von 50 ms entspricht bei einer Abtastfrequenz von 8 kHz 400 Abtastpunkten.

Um der Variation auch in der Darstellung im Spektralraum gerecht zu werden, wird zumeist das Spektrogramm verwendet. Dieses trägt die spektrale Leistungsdichte über die Zeit auf.

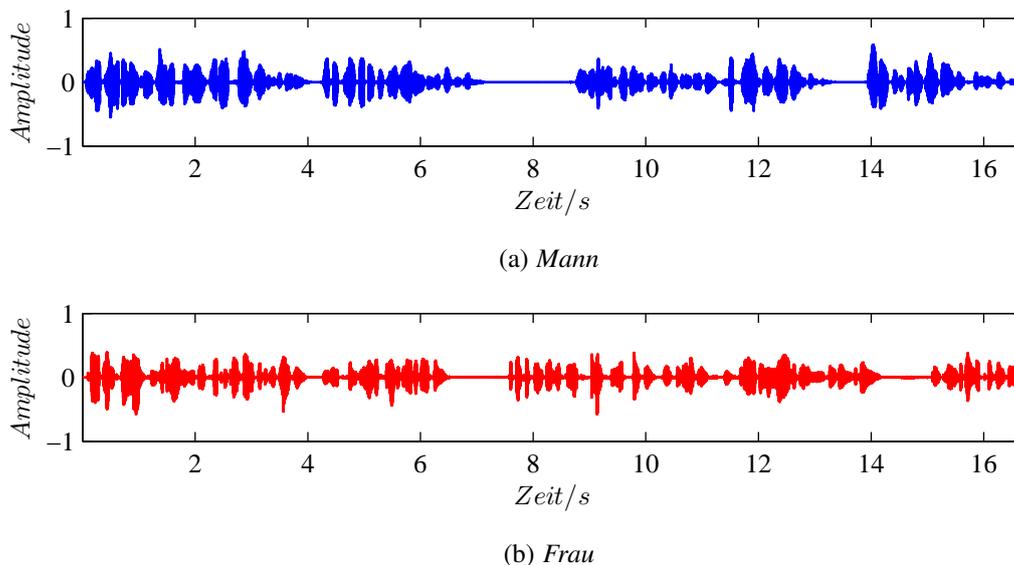
In dieser Arbeit werden zwei Sprachsignale als Testsignale verwendet. Es handelt sich dabei um jeweils einen männlichen und einen weiblichen Sprecher. Weiter unterscheidet sich die Sprache in Deutsch und Englisch. Der genaue Wortlaut für den Mann lautet:

*„Die Natur hat den Menschen eine Zunge aber zwei Ohren gegeben, auf das wir doppelt so viel hören, wie sprechen können. In diesem Ausspruch eines antiken Philosophen steckt vielleicht ein Teil Wunschdenken. Aber es ist auch ein Teil Wahrheit darin enthalten.“*

Der Wortlaut der Frau lautet:

*„To administer medicine to animals is frequently a very difficult matter and yet sometimes it is necessary to do so. The simplest method is to mix the medicine with butter or some other grease and smirr it on the noise of the animal from time to time. Naturally it will lick the grease off.“*

Der Zeitverlauf dieser Signale ist in Bild 6.1 dargestellt.



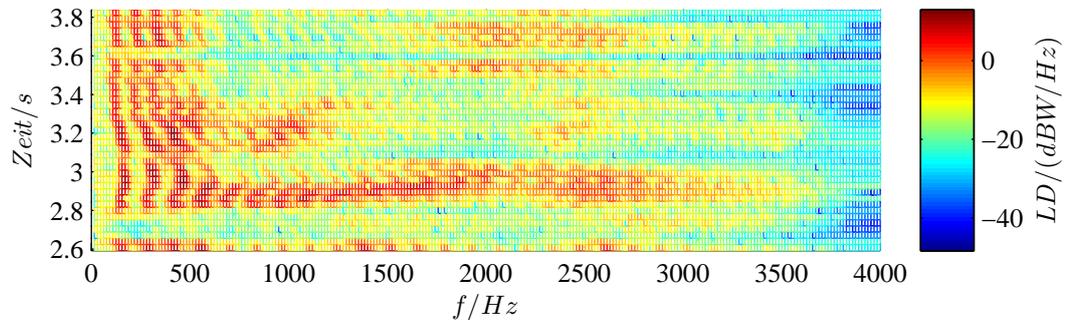
**Bild 6.1:** Zeitverläufe der Testsequenzen

Es ist zu erkennen, dass der Verlauf der Sprachsignale durch sichtbare Pausen geprägt wird. Es handelt sich dabei um Pausen, die zur Worttrennung, zur Betonung von Satzenden und zum Luftholen vonnöten sind.

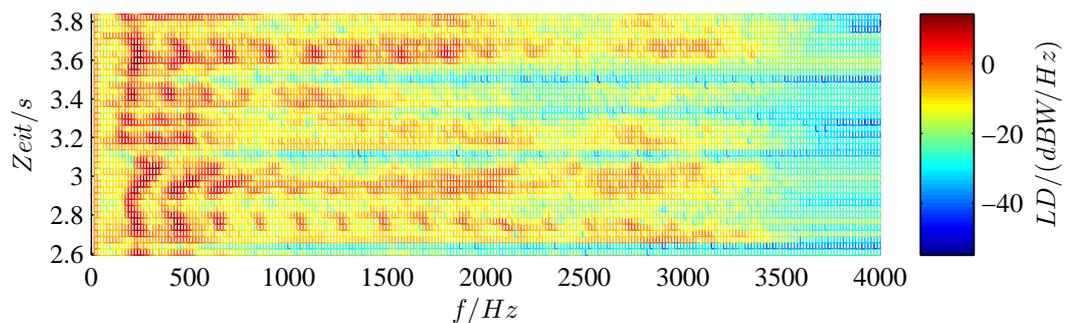
Um auch Aussagen über die Signale im Spektralbereich treffen zu können, werden die entsprechenden Spektrogramme gebildet. Sie sind in Bild 6.2 abgebildet und die Einheit des Zeitsignals wird zu Volt angenommen.

Die Spektrogramme in Bild 6.2 zeigen, dass sich die Signale aus unterschiedlichen Frequenzen zusammensetzen. Diese variieren mit der Zeit. Die Grundfrequenz weist dabei eine

sehr geringe Schwankung (Sprachmelodie) auf, wohingegen die Formanten einer größeren Variation unterliegen. Ebenfalls ist erkennbar, dass sich die spektrale Leistungsdichte für gewisse Zeitabschnitte nicht verändert. In diesen Zeitabschnitten können Sprachsignale als quasistationär angesehen werden.



(a) Mann



(b) Frau

**Bild 6.2:** Teilspektrogramme der Testsequenzen ( $N_{FFT} = 512$ , 50% Überhang, Hamming Fenster)

Durch die Struktur der Sprachsignale (Grundfrequenz und Formante) ist stets mit Peaks im Spektralbereich zu rechnen. Diese Erkenntnis bestätigt sich ebenfalls im Spektrogramm aus Bild 6.2.

## 6.2 Anwendung auf breitbandige Signale

Der UCA-ESPRIT-Algorithmus ist im Ansatz für schmalbandige Signale definiert. Allerdings ist in Abschnitt 6.1 gezeigt worden, dass Sprachsignale keineswegs schmalbandig sind. Die auftretenden Frequenzen können sich vielmehr - abhängig von der Abtastfrequenz - über das gesamte Basisband verteilen. Sprachsignale sind deshalb als breitbandig zu bezeichnen.

Folglich ist prinzipiell der UCA-ESPRIT-Algorithmus nicht für Sprachsignale geeignet. Dies wird in [32, S. 167] explizit angesprochen. Diese Problematik kann jedoch mit dem Übergang in den Frequenzbereich gelöst werden.

### 6.2.1 Übergang in den Spektralbereich

Betrachtet man die Schätzung der Autokorrelationsmatrix (s. Gleichung 3.38)

$$\hat{\mathbf{R}}_{yy} = \frac{1}{N} \mathbf{y}[n] \mathbf{y}[n]^\dagger \approx \mathbf{R}_{yy}$$

so wird für jede Mikrofonkombination die Korrelation berechnet. Betrachtet man nur eine Kombination, gilt für die Korrelation  $\hat{r}_{y_u y_o}(0)$

$$\hat{r}_{y_u y_o}(0) = \frac{1}{N} \sum_{n=0}^{N-1} y_u[n] y_o[n]^* \quad (6.1)$$

wobei  $y_u[n]$  bzw.  $y_o[n]$  das u-te bzw. o-te Mikrofonsignal der Länge  $N$  repräsentiert. Mit dem Parseval-Theorem (vgl. [26, S. 479]) folgt:

$$\hat{r}_{y_u y_o}(0) = \frac{1}{N} \sum_{n=0}^{N-1} y_u[n] y_o[n]^* \quad (6.2)$$

$$= \frac{1}{N^2} \sum_{k=0}^{N-1} Y_u[k] Y_o[k]^* \quad (6.3)$$

Geht man davon aus, dass die Quelle ein schmalbandiges, komplexwertiges Signal näherungsweise der Frequenz  $f_s^{k_x}/N$  aussendet, so tragen lediglich die Anteile des Frequenzganges um diese Frequenz zur Korrelation bei. Es folgt somit

$$\hat{r}_{y_u y_o}(0) = \frac{1}{N^2} \sum_{k=0}^{N-1} Y_u[k] Y_o[k]^* \approx \frac{1}{N^2} \sum_{p=-f_{sw}}^{f_{sw}} Y_u[k_x + p] Y_o[k_x + p]^* \quad (6.4)$$

wobei mit  $f_{sw}$  eine Bandbreite des Signals berücksichtigt wird. Durch die Verwendung der DFT tritt der sogenannte Leckeffekt auf, wodurch sich die Signalenergie über mehrere Frequenzstützstellen erstrecken kann. Die Wahl dieser Bandbreite wird in Abschnitt 6.3.2 näher erläutert.

Der Vorfaktor  $1/N^2$  stellt lediglich einen Proportionalitätsfaktor dar. Folglich hat er auf das Endergebnis keinen Einfluss, falls Gleitkomma-Arithmetik - wie in dieser Arbeit - verwendet wird. Die Größenordnung der Eigenwerte der ersten Eigenwertzerlegung würde

sich entsprechend diesem Faktor verschieben. Die relativen Größenordnungen der Eigenwerte zueinander bleiben allerdings erhalten.

Gleichung 6.4 beschreibt den notwendigen Ansatz für die Verarbeitung breitbandiger Signale mit Peaks im Spektrum. Die Korrelation eines schmalbandigen, komplexwertigen Zeitsignales kann im Spektralbereich näherungsweise durch die Berücksichtigung jener Frequenzstelle, die das Signal aufweist, berechnet werden. Dies ermöglicht im Umkehrschluss, dass bei einem breitbandigen, jedoch mit Peaks behafteten Spektrum, jeder Peak bzw. jede auftretende Frequenz separat betrachtet werden kann. Dies stimmt näherungsweise mit der Berechnung verschiedener schmalbandiger, komplexwertiger Zeitsignale überein.

## 6.2.2 Kohärenter und inkohärenter Ansatz

Durch den Übergang in den Spektralbereich und der Annahme breitbandiger Signale mit mehreren Peaks kann, entsprechend den vorherigen Überlegungen, jeder einzelne Peak zur Berechnung der Einfallswinkel herangezogen werden. Es entsteht eine gewisse Redundanz, die in zweierlei Hinsicht, inkohärent oder kohärent, verarbeitet bzw. genutzt werden kann.

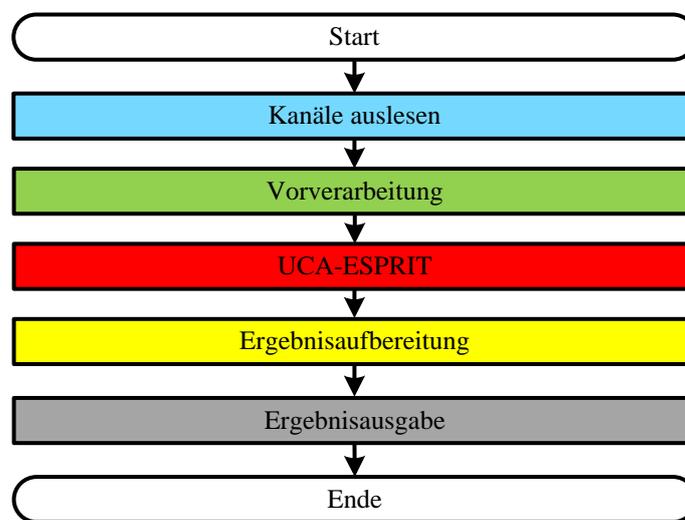
Der kohärente Ansatz versucht die verschiedenen Autokorrelationsmatrizen der Frequenzstützstellen vor der Berechnung der Einfallswinkel miteinander zu verknüpfen. Dazu wird für jeden Peak eine Autokorrelationsmatrix gebildet. Die einzelnen Matrizen werden anschließend entsprechend zusammengerechnet. Ein Hauptvorteil dieses Vorgehens ist, dass die eigentliche Berechnung des Azimutwinkels lediglich einmal pro Datensatz erfolgen muss. Bei der Anwendung dieses Ansatzes auf den UCA-ESPRIT-Algorithmus ist es jedoch notwendig, die Abhängigkeit der Autokorrelationsmatrix von der Frequenz herauszurechnen. Da die Frequenz zusammen mit dem Elevationswinkel  $\theta$  und dem Radius des Arrays das Argument der Besselfunktionen  $\zeta$  bildet, rechnet man ebenfalls die Information über den Elevationswinkel heraus. Somit bleibt lediglich die Möglichkeit der Bestimmung des Azimutwinkels erhalten. Da es aber wünschenswert ist, beide Einfallswinkel zu bestimmen, wird diese Variante der Verarbeitung hier nicht weiter verfolgt. Weitere Ausführungen können jedoch [40, S. 60ff] entnommen werden.

Werden die einzelnen Peaks hingegen getrennt betrachtet und wird gegebenenfalls nach der Berechnung der Einfallswinkel eine Mittelung durchgeführt, so handelt es sich um einen inkohärenten Ansatz. Der Vorteil dieser Variante ist, dass weiterhin beide Einfallswinkel berechnet werden können. Allerdings ist die Berechnung für jeden Peak separat notwendig. Aufgrund der Möglichkeit beide Einfallswinkel zu betrachten, wird die inkohärente Variante in dieser Arbeit verwendet und als Ansatz für die Verarbeitung breitbandiger Signale benutzt.

### 6.2.3 Konzeptansatz

Aufbauend auf den vorherigen Überlegungen kann ein grundsätzliches Konzept abgeleitet werden. Nach dem Einlesen eines entsprechenden Datensatzes, ist eine Vorverarbeitung der Daten sinnvoll. Dies schließt eine gewisse Vorauswahl als auch den dafür vorgesehenen Übergang in den Spektralbereich mit ein. Anschließend kann für alle Peaks des Spektrums der UCA-ESPRIT-Algorithmus angewendet werden. Um die Ergebnisse zu filtern und nachzubearbeiten, wird eine Ergebnisaufbereitung durchgeführt. Zum Schluss erfolgt die Ergebnisausgabe.

Dieser grundsätzliche Ablauf ist in Bild 6.3 grafisch dargestellt.



**Bild 6.3:** Konzeptansatz

## 6.3 Vorverarbeitung

Die Vorverarbeitung der Signale dient dazu, die eingelesenen Daten entsprechend zu verarbeiten und auszuwerten, sodass die Anwendung des UCA-ESPRIT-Algorithmus möglich ist.

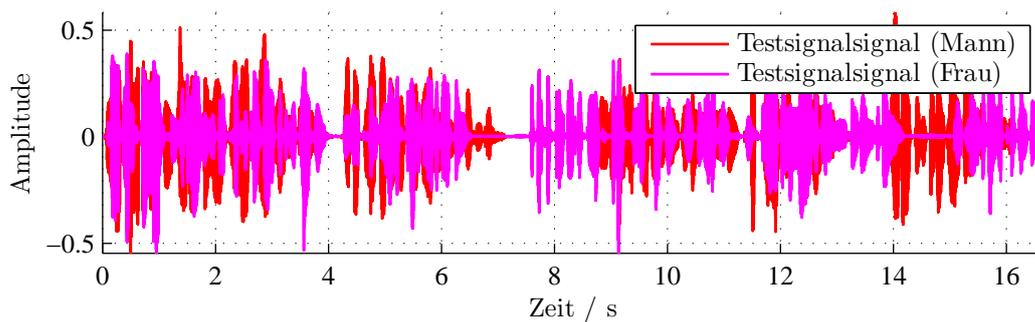
### 6.3.1 Energiebetrachtung

Um zu vermeiden, dass Einfallswinkel auch berechnet werden, falls keine Audioquelle aktiv ist, erscheint es sinnvoll, vor der Berechnung Entscheidungsschwellen einzubauen. Eine erste Möglichkeit einer solchen Schwelle stellt die Energie des Signals dar. Aktive Audioquellen

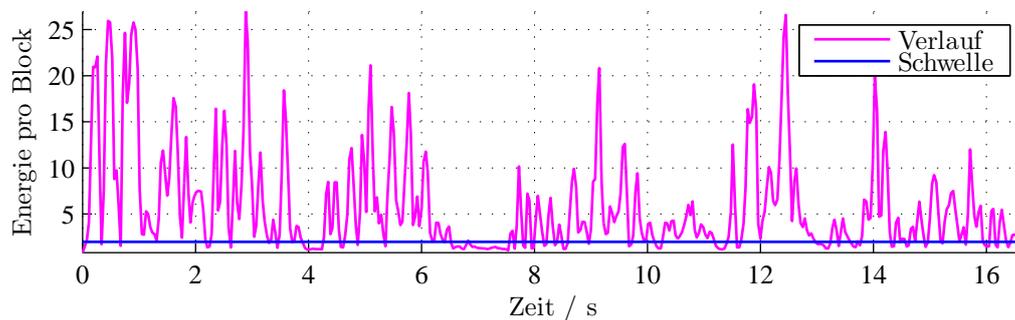
heben die Energie der aufgenommenen Mikrofonsignale  $y_i[n]$  deutlich an. Die Energie ist dabei wie folgt definiert:

$$E_N = \sum_{n=0}^{N-1} y_i[n]^2 \quad (6.5)$$

In 6.4 sind sowohl beide Testsequenzen als auch die Energie eines empfangenen Mikrofonsignals dargestellt. Weiter ist eine mögliche Energieschwelle eingezeichnet.



(a) Zeitverlauf



(b) Energie

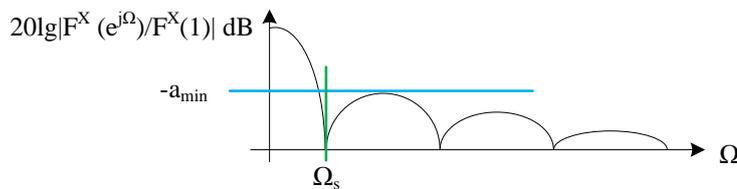
**Bild 6.4:** Energie der empfangenen Testsequenzen ( $N = 512$ )

Die Energie eines Mikrofonsignals steigt deutlich, sobald ein Sprecher aktiv ist. Die zuvor aufgestellte Theorie bestätigt sich. Eine entsprechende Energieschwelle einzuführen ist daher sinnvoll.

### 6.3.2 Fensterung

Durch den Übergang in den Spektralbereich ist die Verwendung der DFT notwendig. Dabei tritt der sogenannte Leckeffekt auf. Um diesen zu minimieren und gleichzeitig eine hohe

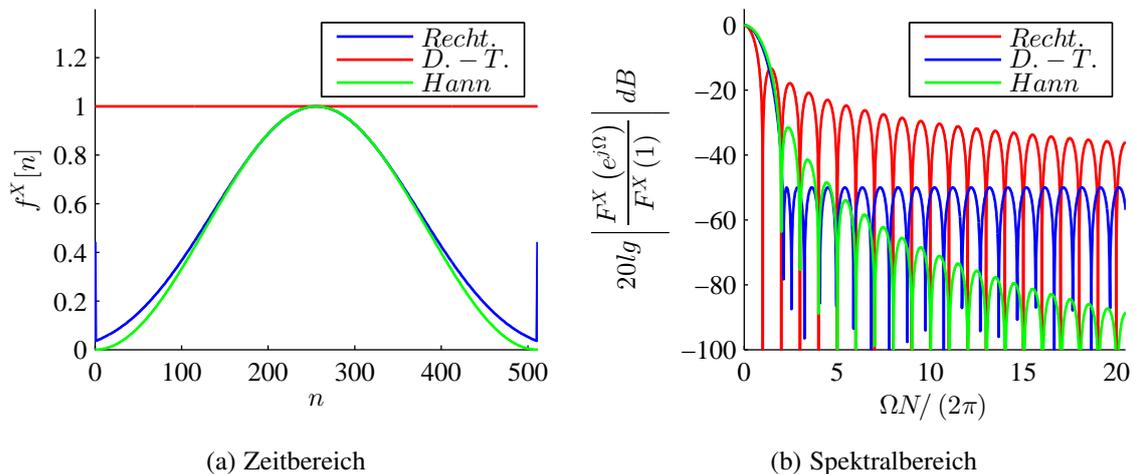
Frequenzauflösung zu erhalten, bietet sich eine Fensterung an. Die Frequenzgänge dieser Fenster weisen alle einen ähnlichen, si-artigen Verlauf auf. Sie lassen sich durch zwei Kennzahlen beschreiben: der minimalen Sperrdämpfung und der halben Breite der Hauptkeule. In Bild 6.5 ist ein typischer Frequenzgang abgebildet und die entsprechenden Kennzahlen sind eingezeichnet.  $\Omega$  beschreibt dabei die normierte Frequenz  $\Omega = 2\pi f/f_s$ .



**Bild 6.5:** Normierter Prinzipamplitudengang Fensterfunktion

Neben den traditionellen Fenstern, wie dem Hann-, Hamming- oder Blackmanfenster, wird mit Hilfe der Dolph-Tschebyscheff-Fenster eine gleichmäßige Approximation im Sperrbereich erzielt. Diese Approximation weist zumeist bessere Eigenschaften auf als die traditionellen Fenster. Weiterhin lassen sie sich parametrisieren. Für genauere Ausführungen bezüglich Leckeffekt und Fensterung sei an dieser Stelle auf [9, S. 292ff.] verwiesen.

In Bild 6.6 werden drei verschiedene Fenstertypen bei einer Fensterlänge von  $N_{DFT} = 512$  miteinander verglichen. Es handelt sich dabei um das Rechteck-Fenster (keine Fensterung), das Hann-Fenster und eine Dolph-Tschebyscheff-Fenster Approximation mit  $\alpha_{min} = 48dB$ .



**Bild 6.6:** Vergleich verschiedener Fenster bei  $N_{DFT}=512$

Der Vergleich im Spektralbereich dieser drei Fenster zeigt, dass das Rechteckfenster die kleinste Hauptkeule und die kleinste minimale Sperrdämpfung aufweist. Das Hann-Fenster

weist näherungsweise die doppelte Hauptkeulenbreite auf, allerdings ist die minimale Sperrdämpfung größer. Das Dolph-Tschebyscheff-Fenster hat die gleiche Hauptkeulenbreite, wie das Hann-Fenster. Allerdings ist die minimale Sperrdämpfung nochmals größer. Sie ist aufgrund der gleichmäßigen Approximation über den gesamten Sperrbereich gleich und größer diesem Wert. Die Sperrdämpfung der anderen Fenster nimmt mit der Frequenz zu. Aufgrund dieser Ergebnisse ist der Einsatz des Dolph-Tschebyscheff-Fensters sinnvoll.

Die Dimensionierung des Dolph-Tschebyscheff-Fensters kann über die minimale Sperrdämpfung ( $\alpha_{min}$  in dB) oder der halben Breite der Hauptkeule  $\Omega_s$  und der DFT-Länge  $N_{DFT}$  erfolgen. Nach [9, S. 304] gilt näherungsweise:

$$\Omega_s \approx \frac{1,46\pi (\lg(2) + \alpha_{min}/(20dB))}{N_{DFT} - 1} \quad (6.6)$$

$$\alpha_{min} \approx 20 \left( \frac{\Omega_s (N_{DFT} - 1)}{1,46\pi} - \lg(2) \right) \quad (6.7)$$

Als guter Kompromiss zwischen minimaler Sperrdämpfung und halber Breite der Hauptkeule erweist sich eine minimale Sperrdämpfung von  $\alpha_{min} = 48$  dB. Die halbe Breite der Hauptkeule ergibt sich dabei zu  $0,0242 = 3,85 \cdot 10^{-3}\pi$ . Bezieht man diesen Wert auf das DFT-Raster von  $2\pi/N_{DFT}$ , ergibt sich eine halbe Breite von näherungsweise zwei DFT-Stützstellen.

Diese Information kann genutzt werden, um das  $f_{SW}$  aus Gleichung 6.4 zu bestimmen. Um die gesamte Hauptkeule eines komplexwertigen, schmalbandigen Signals unter Verwendung der DFT zu berücksichtigen, wird  $f_{SW} = 2$  (halbe Breite der Hauptkeule) gewählt.

### 6.3.3 Überhang

Durch die Fensterung werden die betrachteten Signale am Anfang und am Ende eines Blocks gedämpft. Die Signalinformationen gehen in diesem Bereich für den Datensatz verloren. Um diesem Effekt entgegen zu wirken und um zu vermeiden, dass diese potentiell interessanten Bereiche gar nicht vom Algorithmus erfasst werden, wird jeder betrachtete Block um einen Teil des vorherigen Blocks ergänzt. Dies bewirkt, dass die zuvor stark gedämpften Bereiche im aktuellen oder im folgenden Block schwächer gedämpft werden. Der Informationsverlust nimmt mit der Größe des Überhangs ab.

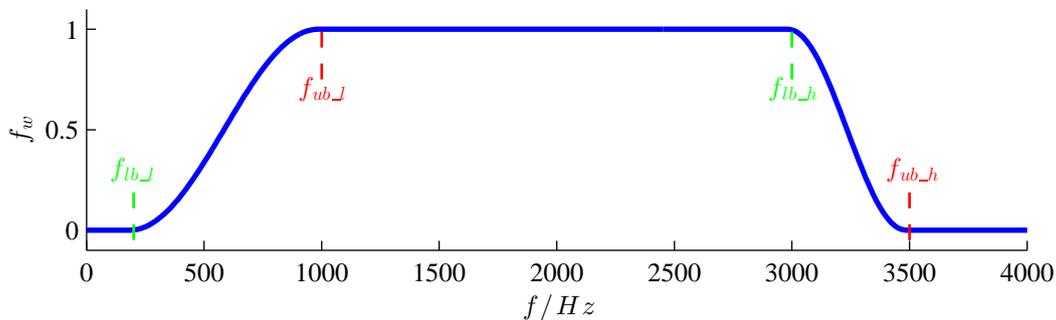
### 6.3.4 Frequenzauswahl

Der grundlegende Ansatz zur Berechnung besteht in der Auswertung verschiedener Peaks im Spektralraum. So kann auf die jeweiligen Autokorrelationsmatrizen geschlossen werden. Allerdings wird bereits bei schmalbandigen Simulationen deutlich, dass nicht jede Frequenz

gleichermaßen geeignet ist. Dies zeigt sich in der Proportionalität der Frequenz zum Argument  $\zeta$  der Besselfunktionen. Es muss daher sichergestellt sein, dass gewisse Frequenzen nicht berücksichtigt werden.

Die Grenzen zwischen sinnvollen und weniger sinnvollen Frequenzen sind fließend. Bedingt dadurch gestaltet sich die Verwendung entsprechender Intervalle ungünstig. Vielmehr bietet sich die Anwendung eines Fensters an, das zum Amplitudengang multipliziert wird. Die Übergänge können dabei durch *Cosinus-Roll-Off*-Funktionen fließend gestaltet werden.

In Bild 6.7 ist ein entsprechendes Fenster dargestellt. Die Anwendung dieses Fensters führt zur Unterdrückung sehr niedriger als auch sehr hoher Frequenzen.



**Bild 6.7:** Fenster zur Frequenzwahl

Das Fenster lässt sich über die Frequenzen  $f_{lb\_l}$ ,  $f_{ub\_l}$ ,  $f_{lb\_h}$  und  $f_{ub\_h}$  parametrisieren.

Die Frequenz  $f_{lb\_l}$  wird dabei zu 200 Hz gewählt. Diese Wahl stellt sicher, dass eventuelle Störungen, die durch die Spannungsversorgung (entsprechend nahe der Netzfrequenz von 50 Hz) ausgeblendet werden.

Die Frequenz  $f_{ub\_h}$  wird zu 3,5 kHz gewählt, um einerseits den Aliasfehler gering zu halten (s. Kapitel 5); andererseits wird die Berücksichtigung von Frequenzen vermieden, die deutlich durch den Anti-Alias-Tiefpassfilter der Analog-Digital-Umsetzung beeinflusst werden.

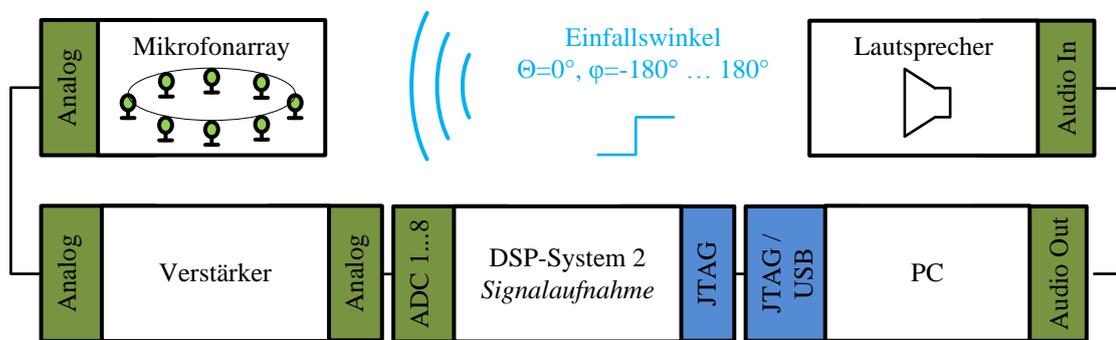
Die beiden mittleren Frequenzen  $f_{ub\_l}$  und  $f_{lb\_h}$  werden in den nächsten Abschnitten dimensioniert.

### 6.3.5 Normierung

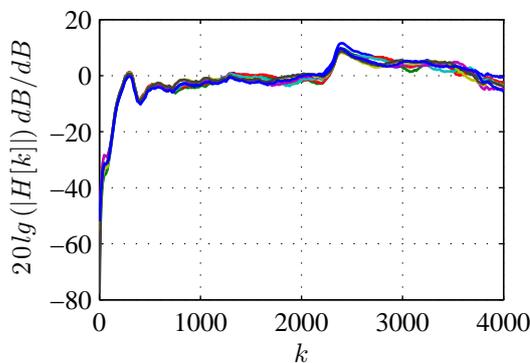
Durch Bauteil- und Fertigungstoleranzen entstehen bei der Signalaufnahme Abweichungen und Fehler im messbaren Bereich. Dies betrifft neben der Arraystruktur auch die Mikrofone sowie alle analogen Bauteile. Um diese Fehler gering zu halten, sind entsprechende Gegenmaßnahmen vorzusehen.

Die durch die gegebene Arraystruktur auftretenden Fehler können durch entsprechende Fehlerkorrekturmaßnahmen minimiert werden. Dazu sei auf das Kapitel 9 verwiesen. Um die restlichen Fehler im Amplituden- und Phasengang zu korrigieren, ist die Sprungantwort in der neutralen Stellung des Arrays (keine relativen Verzögerungen zwischen den Mikrofonsignalen) aufgenommen worden. Durch die Ableitung kann auf die Impulsantwort und folglich auf den Amplituden- und Phasengang geschlossen werden. Dabei handelt es sich um den Amplituden- und Phasengang der gesamten Messanordnung inklusive Lautsprecher, Mikrofon, Verstärker und Analog-Digital-Umsetzer.

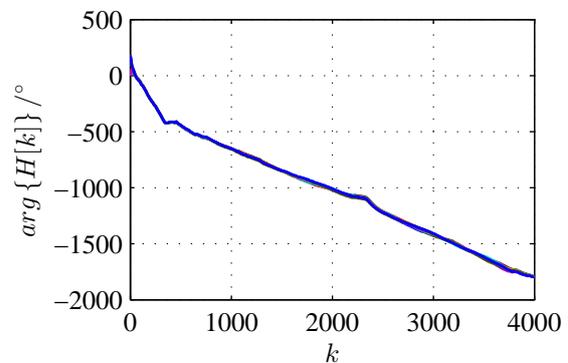
In Bild 6.8 sind sowohl die Messanordnung, als auch die entsprechenden Amplituden- und Phasengänge aller Mikrofone dargestellt.



(a) Aufbau



(b) Amplitudengang



(c) Phasengang

**Bild 6.8:** Amplituden- und Phasengang der Strecke Soundkarte, Lautsprecher, Mikrofonarray, Vorverstärker und Analog-Digital-Umsetzer

Die Ergebnisse zeigen, dass lediglich mit sehr geringen Abweichungen zu rechnen ist. Die deutliche Erhöhung bei 2,4 kHz im Amplitudengang wird durch den verwendeten Lautsprecher hervorgerufen. Dies hat sich durch Vergleichsmessungen mit anderen Lautsprechern gezeigt. Dennoch kann eine gewisse Schwankung der Amplituden zwischen den Kanälen

den restlichen Bauteilen zugeschrieben werden. Ein Ausgleich dieser Schwankungen ist demnach wünschenswert. Die Verwendung des Frequenzgangs erscheint, aufgrund des starken Einflusses des Lautsprechers, ungeeignet.

Es bietet sich jedoch die Möglichkeit, den verwendeten Frequenzbereich auf den auftretenden Peak zu normieren.

$$\hat{r}_{y_u y_o}(0) \approx \frac{1}{N^2 |Y_u[k_x]| |Y_o[k_x]|} \sum_{p=-f_{sw}}^{f_{sw}} Y_u[k_x + p] Y_o[k_x + p]^* \quad (6.8)$$

Dies bewirkt einen Ausgleich der Unterschiede in den Amplitudengängen der Mikrofone. Lediglich die Phasendifferenz zwischen den einzelnen Kanälen trägt zur Berechnung bei. Da ebenso idealerweise lediglich von Phasenunterschieden zwischen den Kanälen ausgegangen wird, stellt dies einen verbesserten Zustand dar.

## 6.4 Diskussion des UCA-ESPRIT-Algorithmus

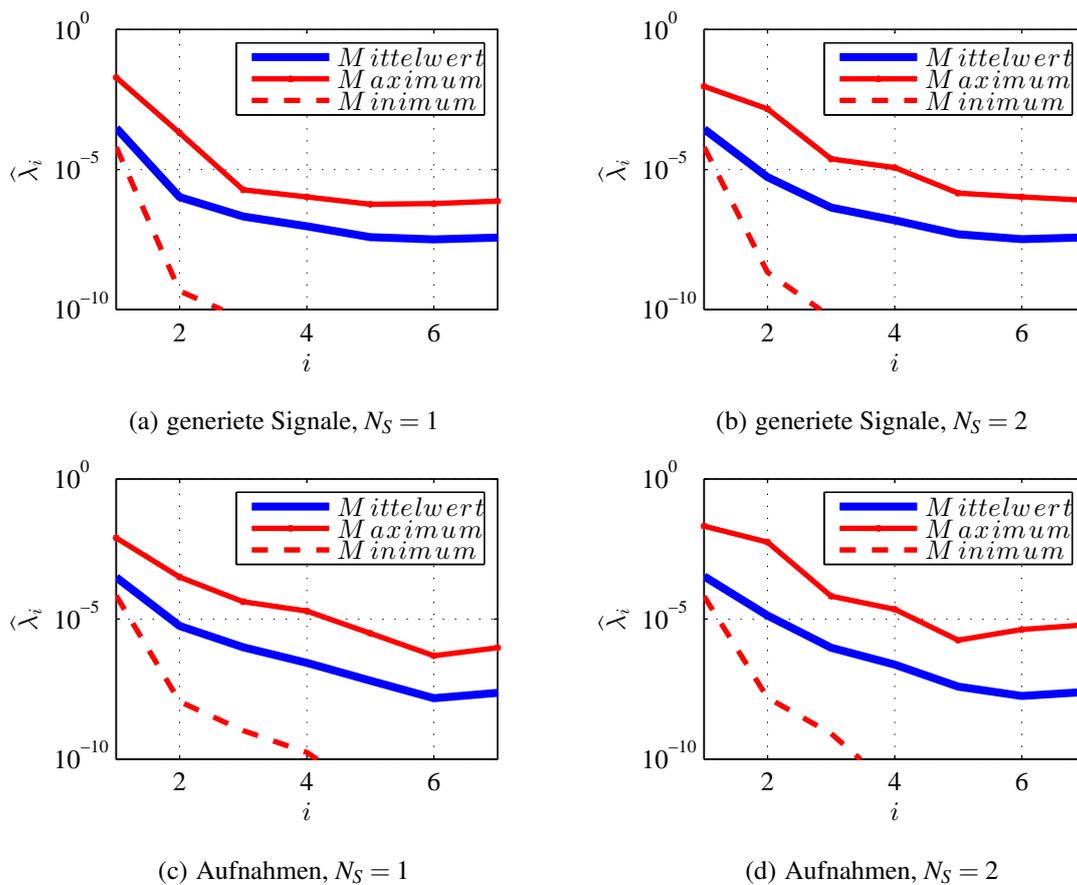
Die bisherigen Überlegungen bezüglich des UCA-ESPRIT-Algorithmus basieren stets auf schmalbandigen, komplexwertigen Signalen. Da die Sprachsignale als breitbandig anzusehen sind, wird der Algorithmus in diesem Abschnitt hinsichtlich der Verwendung von Sprachsignalen diskutiert. Dazu werden umfangreiche Simulationen durchgeführt, die bereits alle Vorverarbeitungsschritte beinhalten. Die mittleren Frequenzen des Auswahlfensters werden so gewählt, dass lediglich eine Übergangsbreite von 100 Hz zwischen Durchlass- und Sperrbereich vorliegt. Die Standardabweichung des weißen gaußschen Rauschen liegt für alle nachfolgenden Simulationen bei 0,01. In Bezug auf eine der Testsequenzen ergibt sich näherungsweise ein SNR von 16 dB.

### 6.4.1 Eigenwertanalyse

Das Wissen über die Anzahl aktiver Quellen ist zumeist nicht a-priori bekannt. Folglich muss diese Größe geschätzt werden. Die theoretischen Ausführungen eröffnen die Möglichkeit, die Anzahl der Quellen über die Eigenwertanalyse bei der Trennung zwischen Signal- und Störunterraum zu schätzen. Die Eigenwerte sollten, falls sortiert, einen deutlichen Sprung aufzeigen. Dieser Sprung teilt die Eigenwerte in zwei Gruppen - Eigenwerte des Signal- und Eigenwerte des Störunterraums - auf. Die Eigenwerte des Störunterraums sind nahezu konstant und liegen deutlich unter denen des Signalunterraums. In den schmalbandigen Simulationen (s. Abschnitt 5.4) bestätigt sich die angenommene Theorie, da diese Simulation die in der Theorie aufgezählten Einschränkungen (schmalbandiges, komplexwertiges Signal und

weißes, mittelwertfreies gauß'sches Rauschen) berücksichtigt. Es ist somit prinzipiell möglich, die Anzahl der Quellen zu schätzen, falls entsprechende Einschränkungen eingehalten werden.

Bei der Betrachtung von Sprachsignalen und unter Verwendung des Ansatzes über den Spektralbereich, ist der Verlauf der Eigenwerte nicht mehr ganz eindeutig. In Bild 6.9 sind die gemittelten, maximalen und minimalen Verläufe der Eigenwerte sowohl bei generierten Testsequenzen als auch bei realen Aufnahmen abgebildet.

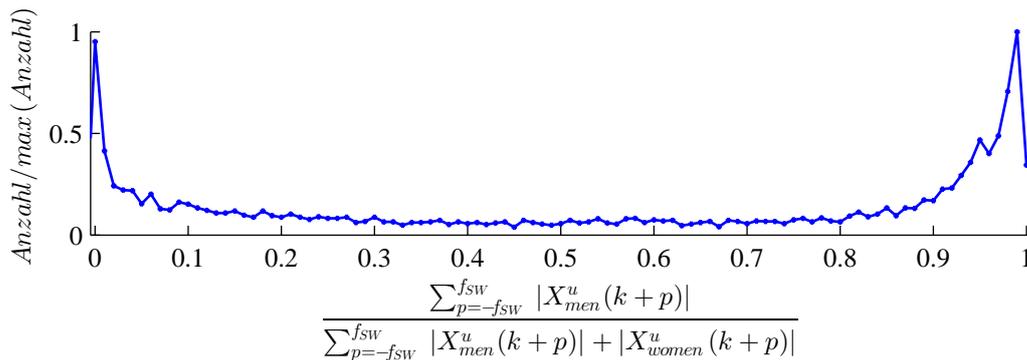


**Bild 6.9:** Verlauf der Eigenwerte bei Sprachsignalen

Bei Begutachtung der Unterbilder sind die vorherigen Überlegungen lediglich in den maximalen und minimalen Verläufen erkennbar. Allerdings werden diese Verläufe in den seltensten Fällen erreicht. Im gemittelten Verlauf ist hingegen kein deutlicher Sprung zwischen Signal- und Störraum erkennbar. Vielmehr ist der Verlauf der Eigenwerte durch eine große Streuung und eine systematische, zunehmende Steigerung der Eigenwerte charakterisiert. Ein Kriterium für die Schätzung über die Anzahl aktiver Quellen ist schwer ableitbar.

Begründet ist dies vor allem darin, dass einerseits das Rauschen nicht als weiß angesehen

werden kann und die schmale, notwendige Auswahl der Frequenzstützstellen zur Berechnung lediglich einen kleinen Teil des Rauschens berücksichtigt. Andererseits ist die Wahrscheinlichkeit für zwei aktive Quellen in einem relativ kleinen Frequenzrahmen gering. In Bild 6.10 ist dazu das Histogramm des Verhältnisses der Summe der Amplitudengangswerte der Mänbertestsequenz zu der Summe der Amplitudengangswerte der überlagerten Testsequenzen für die entsprechende Auswahlbreite dargestellt. Dabei werden nur die Amplitudengangswerte mit einer gewissen Amplitude berücksichtigt. Die FFT-Länge beträgt 512 und der Überhang beläuft sich auf 212.



**Bild 6.10:** Anteil des männlichen Sprechers an Peaks im Amplitudengang

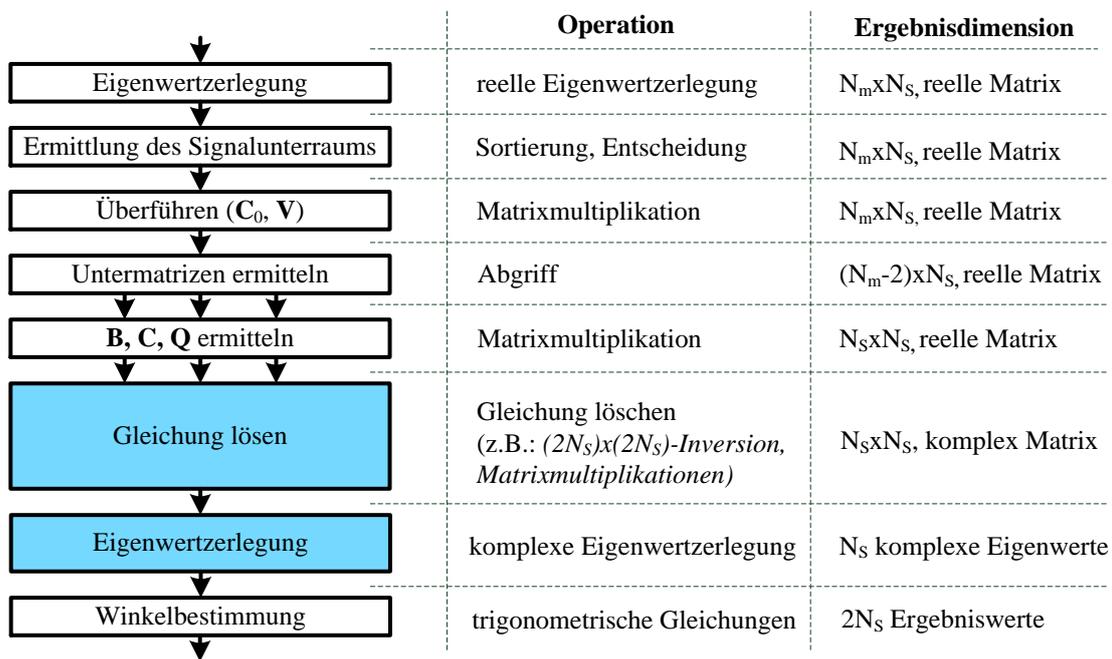
In Bild 6.10 zeigt sich durch die Peaks bei Null und Eins, dass nahezu stets ein Sprecher zum Amplitudengangswert beiträgt bzw. dass es nur zu geringen Überlagerungen kommt. Es ist somit sinnvoll zu prüfen, ob die Hypothese von stets einer Quelle als grundsätzliche Annahme haltbar ist. Dies erscheint auch in Hinblick auf die geringe Wahrscheinlichkeit des Vorhandenseins zweier gleichzeitig aktiver Sprecher als sinnvoll.

Insbesondere hinsichtlich des numerischen Aufwandes ist dies ein interessanter Aspekt. Um dies zu verdeutlichen sind in Bild 6.11 die wesentlichen Schritte und Ergebnisdimensionen des UCA-ESPRIT-Algorithmus nach der Transformation der Autokorrelationsmatrix dargestellt.

In Bild 6.11 wird deutlich, dass sich bei der Beschränkung auf eine Quelle die Dimension der Ergebnisse reduziert. Die Lösung der Gleichung beschränkt sich auf die Inversion einer  $2 \times 2$ -Matrix mit anschließender Multiplikation. Die zweite Eigenwertzerlegung entfällt komplett. Dies stellt einen deutlich geringen numerischen Aufwand dar.

Um zu kontrollieren, ob die Annahme haltbar ist, werden sowohl Simulationen mit generierten Signalen als auch mit Aufnahmen durchgeführt. Dabei sind jeweils zwei Quellen aktiv. Weiter wird sowohl die Annahme getroffen, dass eine bzw. zwei Quellen vorhanden sind. Auf eine Schätzung wird folglich verzichtet.

Die Zeitverläufe für die generierten Mikrofon-signale sind in Bild 6.12 abgebildet. Für die Aufnahmen ergeben sich ähnliche Verläufe, allerdings beträgt die Aufnahmedauer 8 s anstatt



**Bild 6.11:** Numerischer Aufwand des UCA-ESPRIT-Algorithmus

16 s. Die Menge  $K$  enthält alle Ergebnisse, die lediglich in beiden Richtungen um maximal  $10^\circ$  vom korrekten Ergebnis abweichen. Die Menge  $I$  enthält alle weiteren Ergebnisse.

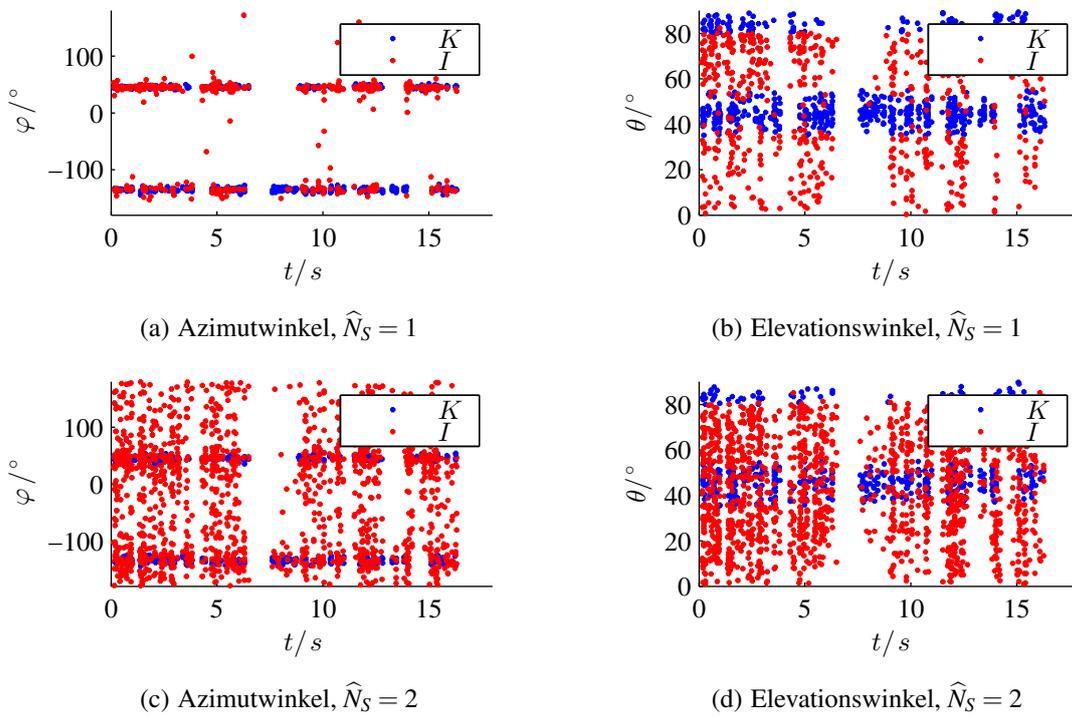
Betrachtet man die Ergebnisse, so wird deutlich, dass die Verläufe, die unter der Annahme einer Quelle berechnet worden sind, weniger streuen. Über die genauen Verhältnismäßigkeiten ist jedoch keine Aussage möglich. Abhilfe schafft die Übertragung der Ergebnisse in ein Histogramm. Dies ermöglicht die Ableitung quantitativer Aussagen.

In Bild 6.13 und 6.14 sind die entsprechenden Histogramme dargestellt. Dabei sind unter der Annahme, dass zwei Quellen vorliegen, die ersten und zweiten Ergebnisse getrennt abgebildet.

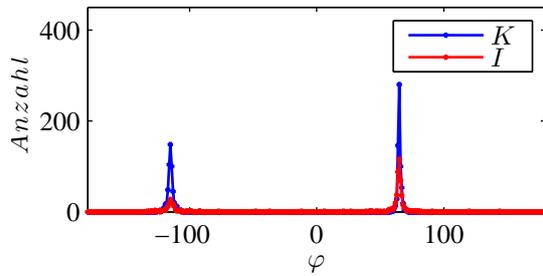
Die Bilder zeigen sowohl für die generierten Mikrofon-signale als auch für die Aufnahmen ähnliche Verläufe. Unter der Annahme, dass lediglich eine Quelle vorliegt, bilden sich Peaks aus. Diese sind im Azimutwinkel schmaler und höher als im Histogramm des Elevationswinkels. Die Streuung im Elevationswinkel ist folglich höher.

Werden zwei Quellen angenommen, weisen die ersten Ergebnisse einen ähnlichen Verlauf auf. Allerdings sind die Histogrammwerte geringer und streuen mehr. Die zweiten Ergebnisse bilden hingegen sehr kleine Peaks aus. Sie streuen über den gesamten Wertebereich.

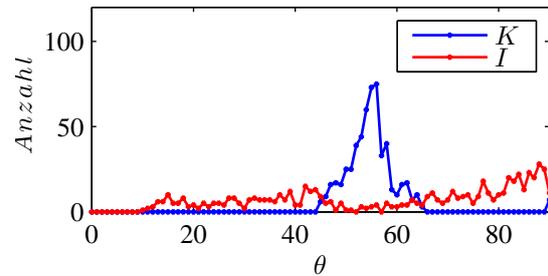
Die Bilder 6.13 und 6.14 zeigen somit, dass die Hypothese einer Quelle haltbar ist. Die Peaks der Histogramme, die mit der Annahme des Vorliegens einer Quelle getroffen worden



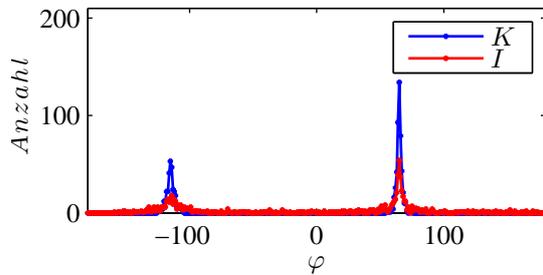
**Bild 6.12:** Zeitverlauf der Lokalisierung (generierte Signale,  $N_S = 2$ )



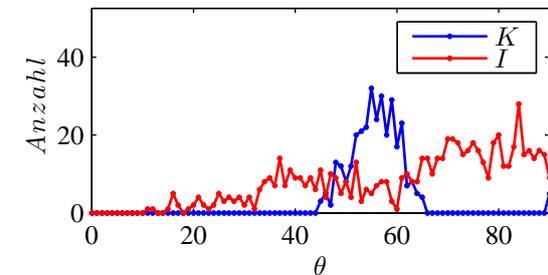
(a) Azimutwinkel,  $\hat{N}_S = 1$



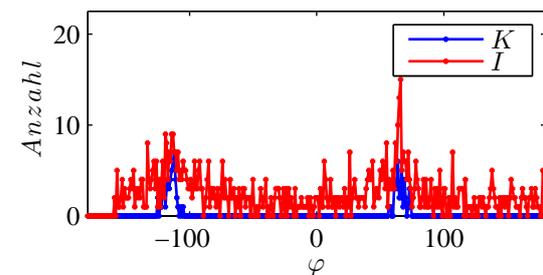
(b) Elevationswinkel,  $\hat{N}_S = 1$



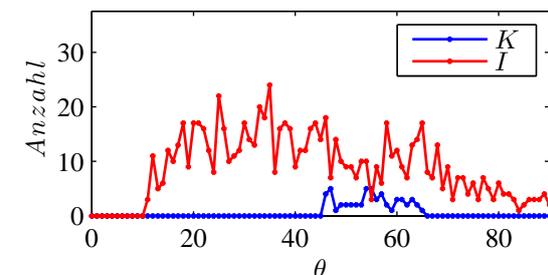
(c) Azimutwinkel,  $\hat{N}_S = 2$ , Ergebnisse des ersten Eigenwertes



(d) Elevationswinkel,  $\hat{N}_S = 2$ , Ergebnisse des ersten Eigenwertes

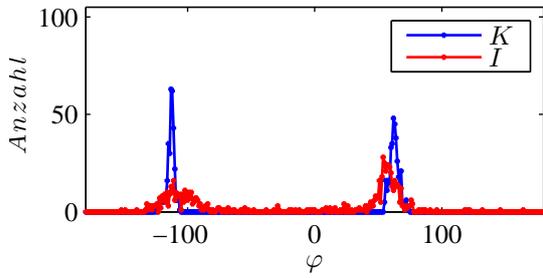


(e) Azimutwinkel,  $\hat{N}_S = 2$ , Ergebnisse des zweiten Eigenwertes

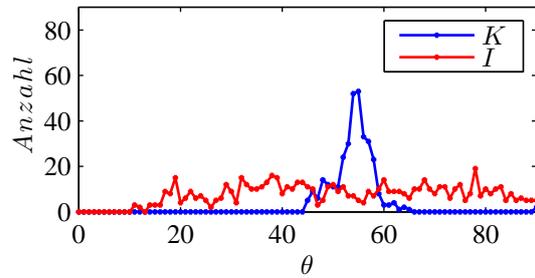


(f) Elevationswinkel,  $\hat{N}_S = 2$ , Ergebnisse des zweiten Eigenwertes

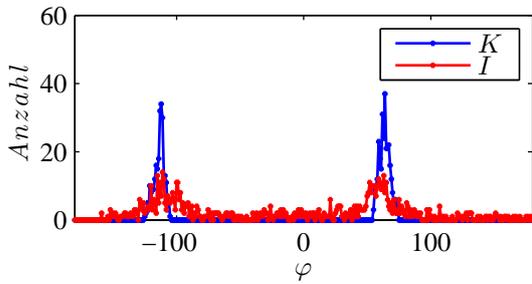
**Bild 6.13:** Histogramm unter Verwendung generierter Signale ( $N_S = 2$ , Auflösung  $1^\circ$ )



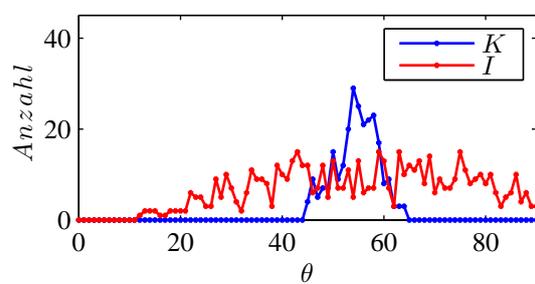
(a) Azimutwinkel,  $\hat{N}_S = 1$



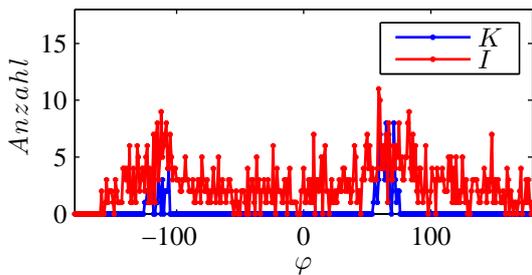
(b) Elevationswinkel,  $\hat{N}_S = 1$



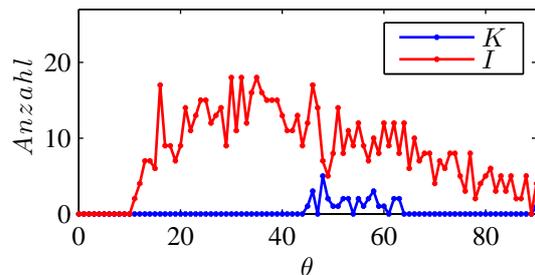
(c) Azimutwinkel,  $\hat{N}_S = 2$ , Ergebnisse des ersten Eigenwertes



(d) Elevationswinkel,  $\hat{N}_S = 2$ , Ergebnisse des ersten Eigenwertes



(e) Azimutwinkel,  $\hat{N}_S = 2$ , Ergebnisse des zweiten Eigenwertes

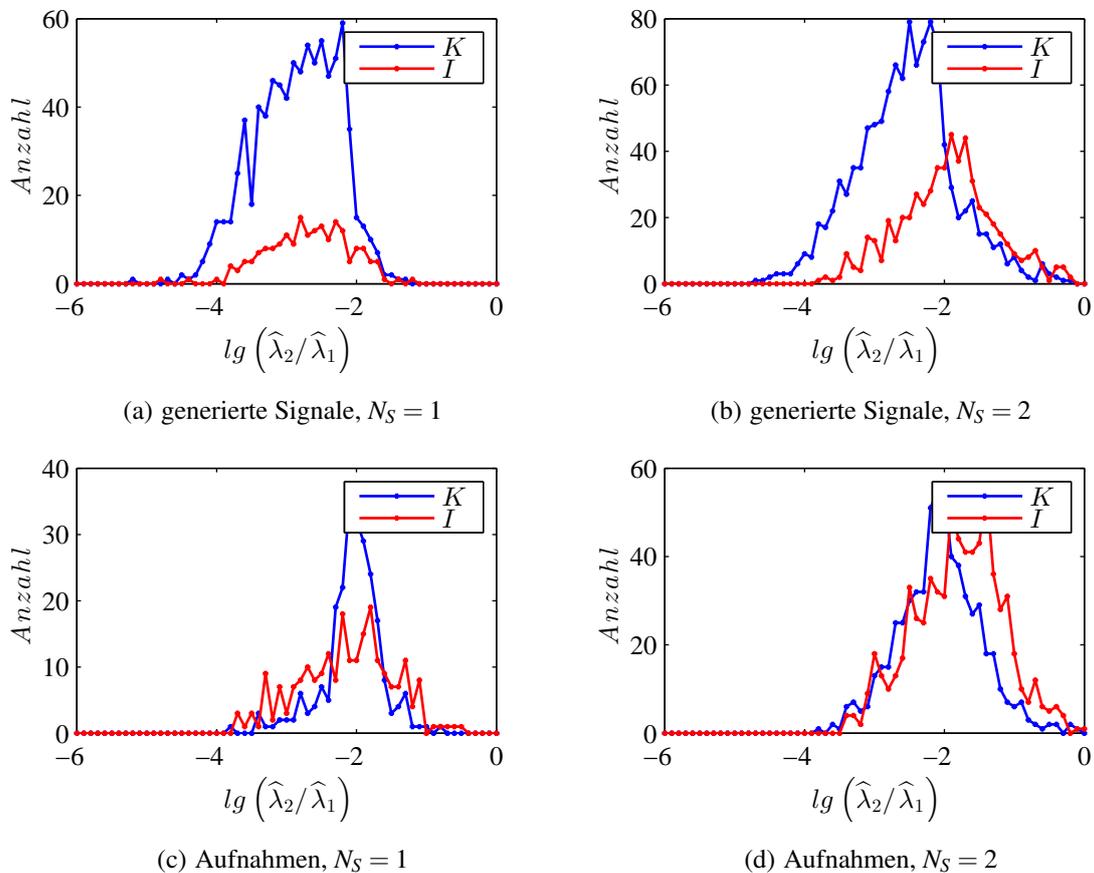


(f) Elevationswinkel,  $\hat{N}_S = 2$ , Ergebnisse des zweiten Eigenwertes

**Bild 6.14:** Histogramm unter Verwendung von Aufnahmen ( $N_S = 2$ , Auflösung  $1^\circ$ )

sind, weisen deutlich höhere Werte auf als die Histogramme, die unter der Annahme zweier Quellen getroffen worden sind.

Weiter sind die Erkenntnisse aus den schmalbandigen, komplexwertigen Simulationen zu beachten (s. Abschnitt 5.4). Falls die Größenordnungen des ersten und zweiten Eigenwertes sich ähneln, ist mit einer Mittelung zwischen den korrekten Werten als Ergebnis zu rechnen. Um dies zu umgehen, ist es denkbar, die Berechnungen nicht fortzuführen, wenn das Verhältnis zwischen dem ersten und dem zweiten Eigenwert ungünstig ist. Der Verlauf der Eigenwerte würde somit als relativ schwaches Kriterium in den Algorithmus miteinbezogen. In der Bildergruppe 6.15 sind die Histogramme des Verhältnisses des zweiten zum ersten Eigenwert dargestellt.



**Bild 6.15:** Verhältnis des zweiten zum ersten Eigenwert ( $\hat{N}_S = 1$ )

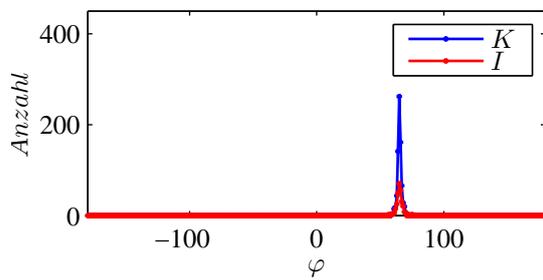
Die Ergebnisse zeigen, dass bei fehlerhaften Ergebnissen (Menge I) das Verhältnis des zweiten zum ersten Eigenwert tendenziell größer ist. Allerdings zeigt sich auch, dass es größtenteils zu Überschneidungen zwischen den Mengen K und I kommt. Durch diese Erkenntnis bestätigt sich, dass der Verzicht der Schätzung über die Anzahl der Quellen

sinnvoll ist, da dieser Ansatz der Schätzung stark fehlerbehaftet wäre. Das Wissen über die tendenzielle Verschiebung kann genutzt werden, indem das Verhältnis unter einem gewissen Wert liegen muss, falls die Berechnung fortgeführt werden soll. Allerdings muss beachtet werden, dass dadurch auch korrekte Ergebnisse ausgelassen werden. Somit ist die Schwelle behutsam zu wählen. Entsprechend der Ergebnisse aus Bild 6.15 wären ein oder zwei Dekaden als minimaler Unterschied denkbar.

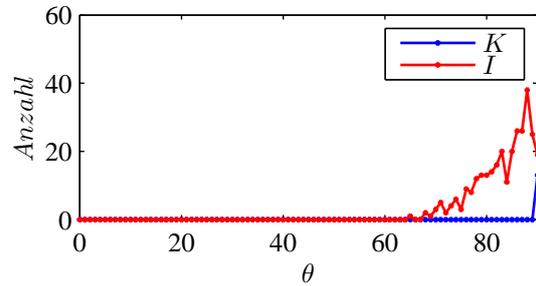
#### 6.4.2 Anzahl der betrachteten Phasen-Moden

Durch die Annahme, dass lediglich eine Quelle vorliegt, ist die Frage nach der Anzahl der betrachteten Phasen-Moden bzw. nach dem größten betrachteten Phasen-Modus zu hinterfragen. Wie zuvor beschrieben, lässt sich  $M$  prinzipiell zu  $M = 2$  oder  $M = 3$  wählen. Die Wahl von  $M = 2$  würde den numerischen Rechenaufwand weiter reduzieren, da weniger Phasen-Moden in die Berechnung mit einbezogen würden. Der auftretende Alias-Effekt wäre ebenfalls kleiner. Allerdings könnte durch die Reduzierung der betrachteten Phasen-Moden die Genauigkeit abnehmen. Welche Effekte die größeren Auswirkungen auf das Endergebnis haben, wird mit Hilfe einer Simulation untersucht. Dazu werden die vorherigen Simulationen mit  $M = 2$  wiederholt und mit den Simulationen mit  $M = 3$  verglichen. Die Ergebnisse sind in den Bildern 6.16 und 6.17 dargestellt.

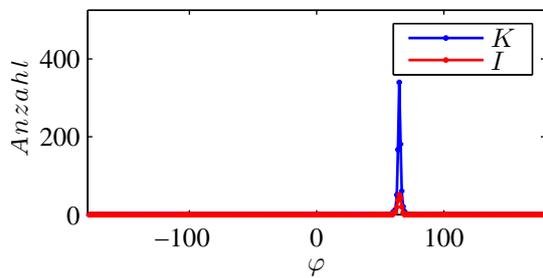
Die Ergebnisse zeigen, dass für  $M = 3$  die Peaks höher ausfallen. Dies lässt darauf schließen, dass eine höhere Anzahl an Phasen-Moden - hier  $M = 3$  - vorzuziehen ist.



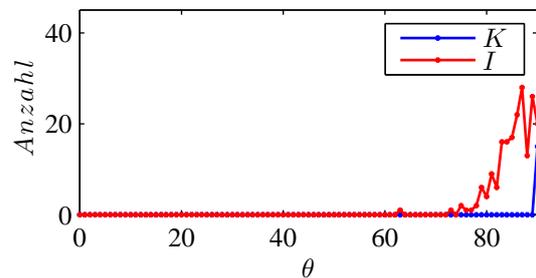
(a) Azimutwinkel,  $N_S = 1, M = 2$



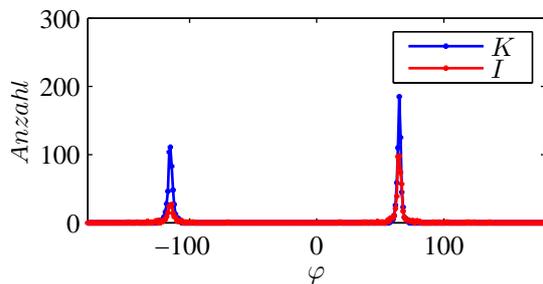
(b) Elevationswinkel,  $N_S = 1, M = 2$



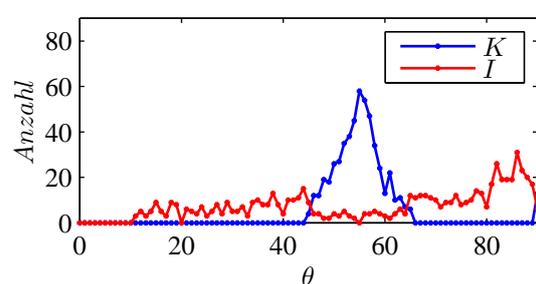
(c) Azimutwinkel,  $N_S = 1, M = 3$



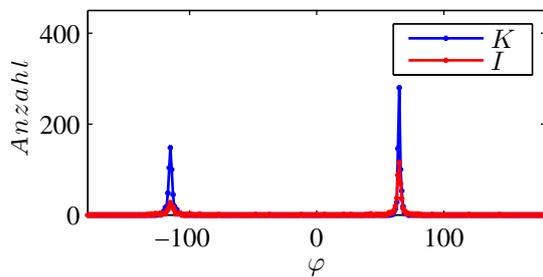
(d) Elevationswinkel,  $N_S = 1, M = 3$



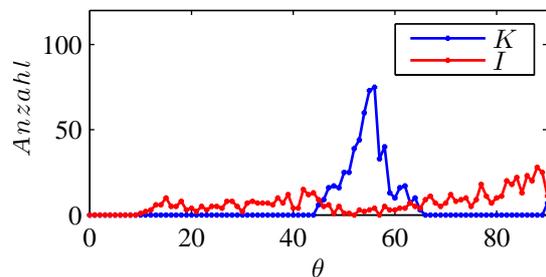
(e) Azimutwinkel,  $N_S = 2, M = 2$



(f) Elevationswinkel,  $N_S = 2, M = 2$

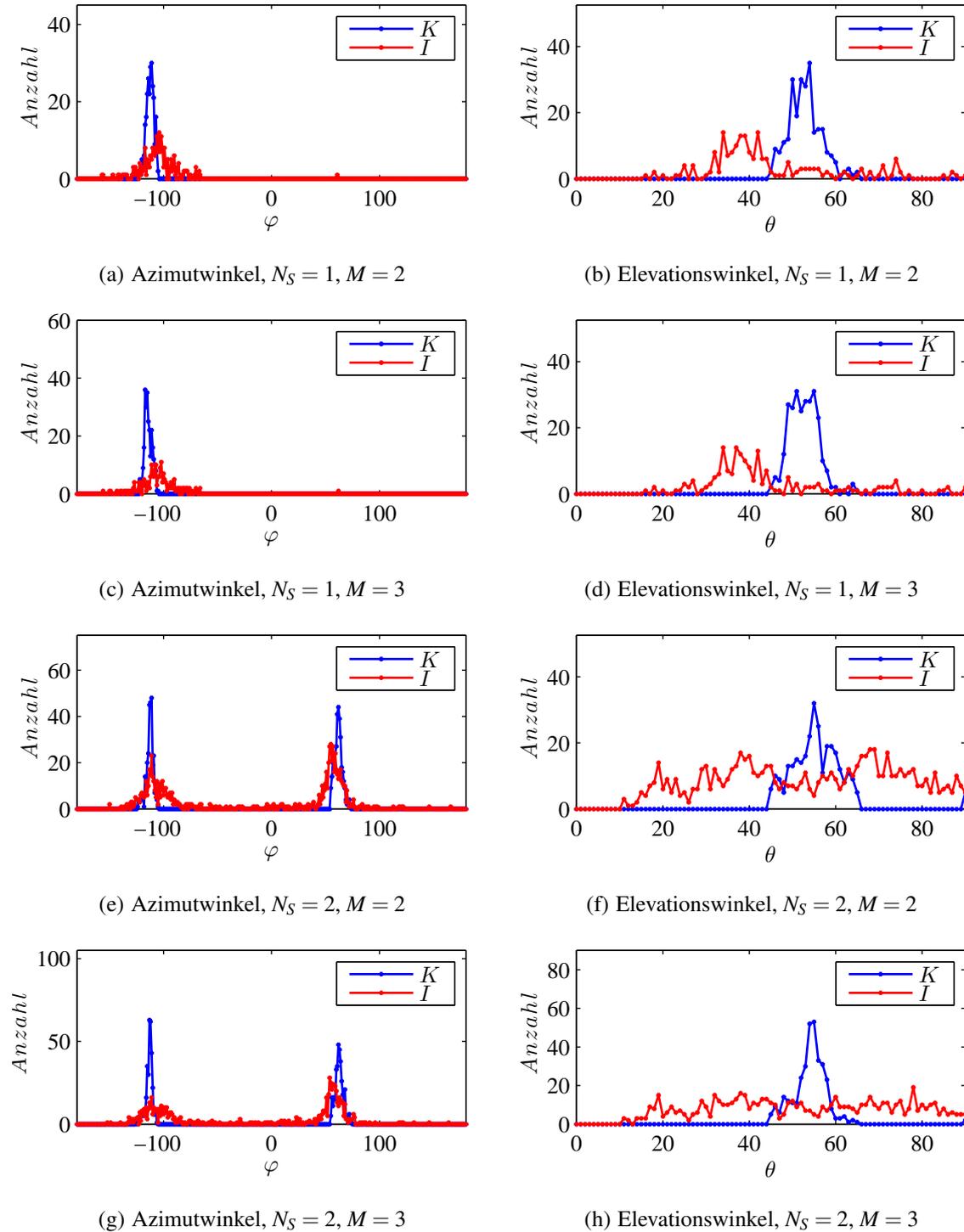


(g) Azimutwinkel,  $N_S = 2, M = 3$



(h) Elevationswinkel,  $N_S = 2, M = 3$

**Bild 6.16:** Ergebnishistogramme für  $M = 2$  und  $M = 3$  (generierte Signale, Auflösung  $1^\circ$ )



**Bild 6.17:** Ergebnishistogramme für  $M = 2$  und  $M = 3$  (Aufnahmen, Auflösung  $1^\circ$ )

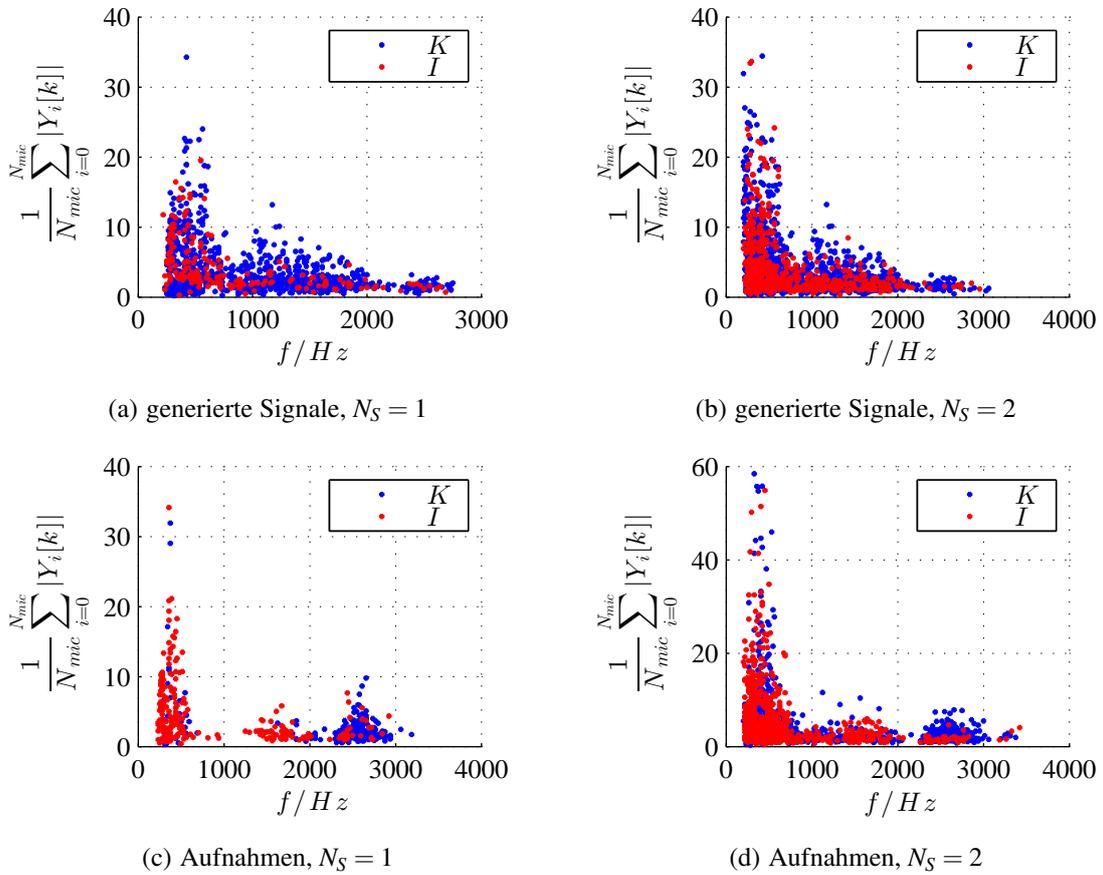
### 6.4.3 Frequenzabhängigkeit

Wie bereits in Abschnitt 6.3.4 angedeutet, ist der Algorithmus frequenzabhängig. Dies ist bedingt durch die Proportionalität zwischen der Frequenz und dem Argument der Besselfunktionen.

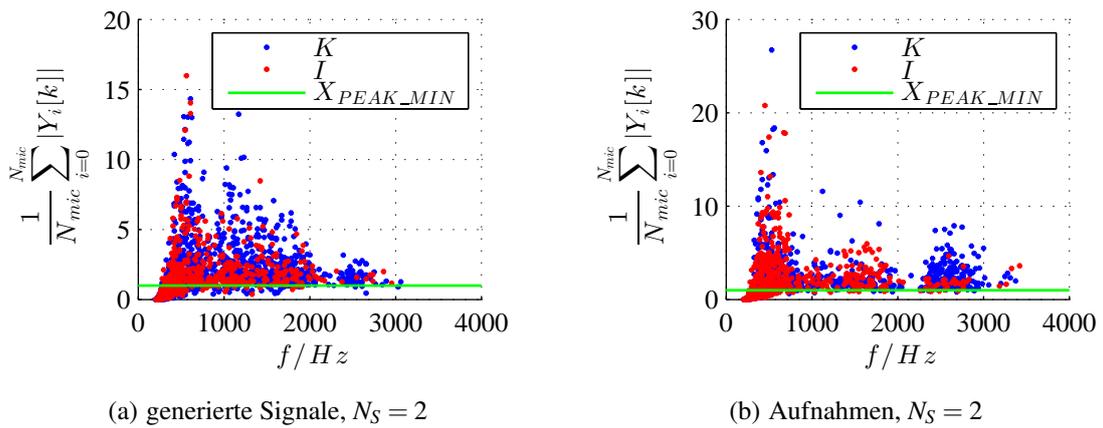
Um die Auswirkungen dieser Frequenzabhängigkeit zu zeigen, sind bereits schmalbandige Simulationen durchgeführt worden. Dabei zeigte sich, dass der Aliasfehler mit höherer Frequenz zunimmt. Allerdings sollte eine gewisse Frequenz vorhanden sein, wenn zwei Quellen voneinander getrennt werden sollen. Da dies jedoch aufgrund der vorherigen Überlegungen entfällt, ist im Wesentlichen die obere Frequenz interessant. Um die Auswirkungen des Alias-Effekts gering zu halten, sollte die Frequenz nicht deutlich größer als 3 kHz gewählt werden. Somit ist es sinnvoll die Grenze  $f_{lb\_h}$  auf diesen Frequenzwert zu legen.

Obwohl bei den niedrigen Frequenzen der Alias-Effekt deutlich geringer ist, ist eine kurze Betrachtung, ob ein weiterer Zusammenhang zwischen Ergebnis und Frequenz existiert, sinnvoll. Dazu sind die inkorrekten und korrekten Ergebnisse in Bild 6.18 gegenüber der Amplitude und der Frequenz aufgetragen. Es handelt sich um Ergebnisse, die sowohl durch generierte Testsequenzen als auch mit Testaufnahmen erzielt worden sind.

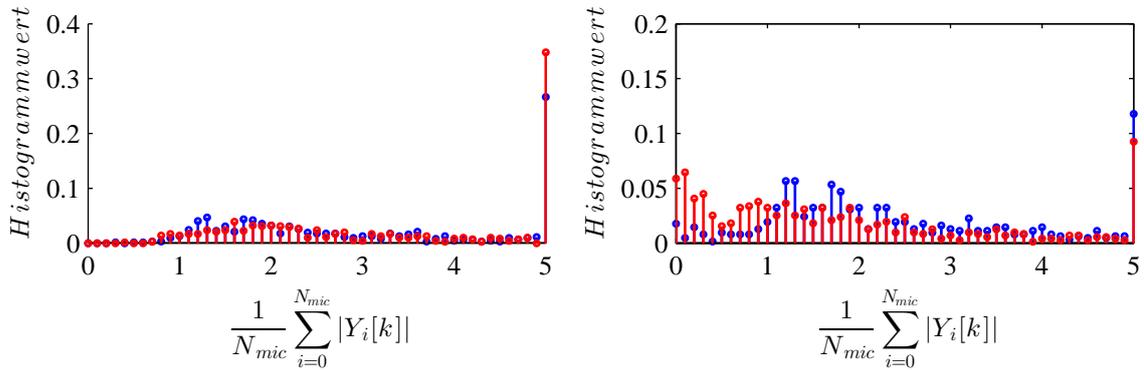
Die Graphen zeigen, dass vor allem bei niedriger Frequenz und niedriger Amplitude mit fehlerhaften Ergebnissen zu rechnen ist. Dieser Zusammenhang kann genutzt werden, um weitere fehlerhafte Ergebnisse auszuschließen. Dazu wird die Amplitude mit Hilfe des Frequenzfensters im Bereich von 200 Hz bis 800 Hz vermindert, wodurch ein Teil der Peaks unter die Schwelle fällt. Dies ist beispielsweise für zwei der Szenarien aus Bild 6.18 durchgeführt worden. Die Ergebnisse sind in Bild 6.19 zu sehen. Es ist erkennbar, dass eine entsprechende Anzahl an inkorrekten Ergebnissen unter die Schwelle fallen sollen. Leider werden auch korrekte Ergebnisse verworfen. Dennoch erscheint die Dämpfung der Amplituden sinnvoll, da das Verhältnis zwischen korrekten und inkorrekten Ergebnissen deutlich ohne Verlust eines Großteils der Ergebnisse verbessert wird (s. Histogramm aus der Bildergruppe 6.20).



**Bild 6.18:** Amplituden-Frequenzabhängigkeit



**Bild 6.19:** Amplituden-Frequenzabhängigkeit mit angepasstem Frequenzauswahlfenster



(a) schmale Bereichsübergänge im Frequenzauswahlfenster

(b) breite Bereichsübergänge im Frequenzauswahlfenster

**Bild 6.20:** Histogramm der Amplituden (Aufnahmen,  $N_S = 2$ )

### 6.4.4 Struktur der Ergebniswerte

Die Ergebnisse in den vorherigen Simulationen haben deutliche Unterschiede in den Ergebnissen des Azimut- und des Elevationswinkels ergeben. Begründet ist dies in der Struktur der zugrundeliegenden Eigenwerte. Die Problematik sei hier genauer erläutert.

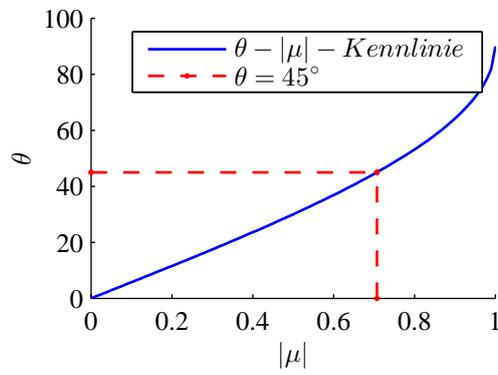
Die Ergebniswinkel werden aus den ermittelten Eigenwerten bestimmt. In Gleichung 5.69 wird die Struktur der Eigenwerte angegeben:

$$\mu = \sin(\theta) e^{j\varphi}$$

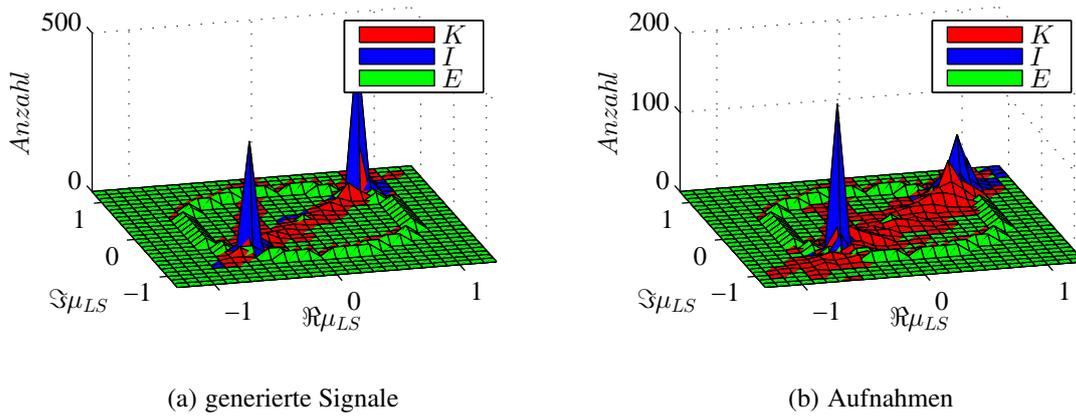
Dabei ist der Azimutwinkel in der Phase des Eigenwertes und der Elevationswinkel im Betrag enthalten. Allerdings geht der Elevationswinkel als Argument der Sinusfunktion in den Betrag des Eigenwertes ein. Dies bewirkt eine Verzerrung, wodurch sich der Dynamikbereich mit dem Elevationswinkel ändert. Um die Verzerrung zu verdeutlichen ist in Bild 6.21 die entsprechende Kennlinie dargestellt.

Es ist deutlich zu sehen, dass die Elevationswinkel von  $0^\circ$  bis  $45^\circ$  einen Betrag von 0 bis etwa 0,7 ergeben. Der Bereich von  $45^\circ$  bis  $90^\circ$  umfasst hingegen einen Betrag von 0,7 bis 1. Die Steigung der Kennlinie nimmt mit der Steigerung des Elevationswinkels zu. Aus dieser Erkenntnis folgt, dass der Dynamikbereich im unteren Elevationswinkelbereich deutlich höher ist als im oberen Bereich.

Um die Auswirkungen dieses Zusammenhanges zu verdeutlichen sind in Bild 6.22 Histogramme der Eigenwerte bei zwei Quellen abgebildet. Die Menge  $K$  entspricht den korrekten und  $I$  den inkorrekten Ergebnissen entsprechend den vorherigen Ausführungen. Die Menge  $E$  deutet den Einheitskreis an.



**Bild 6.21:** Eigenwertkennlinie des Elevationswinkels



(a) generierte Signale

(b) Aufnahmen

**Bild 6.22:** Ergebnishistogramm der Eigenwerte

Es ist deutlich zu sehen, dass die Eigenwerte einer gewissen Streuung unterliegen, die näherungsweise einer Normalverteilung entspricht. Der Anstieg der Steigung der Kennlinie bewirkt, dass diese Streuung einen deutlich größeren Fehler für größere Elevationswinkel als für kleinere hervorruft.

Ebenfalls zeigt sich, dass auch Eigenwerte auftreten, die im Betrag größer als Eins sind. Dies ist in der Theorie ausgeschlossen, muss jedoch in der Praxis betrachtet werden. Ein erster Ansatz würde alle Ergebnisse mit Beträgen größer Eins verwerfen. Dies würde praktisch das Erreichen des Elevationswinkels  $90^\circ$  ausschließen. Um diesem Effekt entgegen zu wirken, könnte die Schwelle erst bei einem Betrag von über Eins greifen. Für alle Ergebnisse in diesem Übergangsbereich werden die Elevationswinkel auf  $90^\circ$  gesetzt. Ebenfalls wäre es denkbar, auch Werte über  $90^\circ$  zuzuweisen und erst in der entsprechenden Nachbearbeitung auf  $90^\circ$  zu begrenzen. Allerdings müssten dazu weitere Untersuchungen durchgeführt werden, mit welcher Kennlinie die beste Korrektur erzielt wird. Weiter darf nicht vernachlässigt werden, dass ein Betrag in der Theorie einen Fehler darstellt. Aus diesen Gründen sei zunächst der Ansatz mit der Verwerfung der Ergebnisse ab einem bestimmten Betrag und einer vorherigen Begrenzung auf einen Winkel von  $90^\circ$  gewählt.

## 6.5 Ergebnisaufbereitung

Die Ergebnisse des UCA-ESPRIT-Algorithmus unterliegen Streuungen und weisen Fehler auf. Um diesen Effekten entgegen zu wirken, ist eine Ergebnisaufbereitung sinnvoll. Weiter können durch geschicktes Verarbeiten der Ergebnisse mehrere Quellen, trotz der Annahme einer Quelle bei der Berechnung, detektiert werden.

### 6.5.1 Mittelwert- und Medianfilter

Um Schwankungen auszugleichen, bietet sich zunächst ein Mittelwert- oder Medianfilter an. Bei diesen Filtern wird davon ausgegangen, dass sich die Schwankungen um das korrekte Ergebnis streuen. Dies trifft im Wesentlichen auf die auftretenden Streuungen zu. Allerdings führen diese Filter beim Vorhandensein mehrerer Quellen zu deutlichen Fehlern.

### 6.5.2 Zweistufige Histogrammauswertung

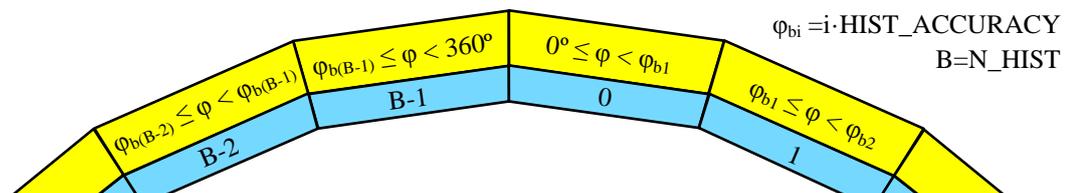
Als Maßnahme gegen diesen Effekt ist ein fortlaufendes Histogramm geeignet. Eine gewisse Anzahl der letzten Werte werden zwischengespeichert. Anschließend werden die Häufigkeiten der Ergebnisse über dem Zwischenspeicher bestimmt und die Maxima gesucht. Diese Maxima

beschreiben die Bereiche der Endergebnisse. Dieses Vorgehen ermöglicht es somit mehrere Quellen zu detektieren.

Die vorherigen schmalbandigen als auch breitbandigen Simulationen haben gezeigt, dass die Streuungen des Azimutwinkels unter denen des Elevationswinkels liegen. Somit eignet sich der Azimutwinkel als Leitwinkel, über den das Histogramm gebildet wird. Dies bedeutet gleichsam, dass keine zwei Quellen mit gleichem Azimutwinkel unterschieden werden können. Allerdings kann dieser Fall als sehr unwahrscheinlich angenommen werden, da dies bei entsprechender Arraylage lediglich bei zwei übereinander liegenden Sprechern gegeben ist.

Die Auflösung des Histogramms ist schwer zu wählen. Eine geringe Auflösung führt zwar zu deutlicheren Peaks, allerdings ist die Ergebnisauflösung ebenfalls gering. Eine hohe Auflösung kann hingegen dazu führen, dass sich gar keine Peaks bilden. Um dieses Problem zu umgehen, wird mit einer zweistufigen Histogrammauswertung gearbeitet:

Das Histogramm wird mit einer geringen Auflösung gebildet. Diese Auflösung wird durch den kleinsten Abstand zweier Quellen definiert, bei den die Quellen vom System noch getrennt werden sollen. Der prinzipielle Aufbau des Histogramms ist in Bild 6.23 dargestellt. Um den zirkularen Wertebereich des Azimutwinkels zu verdeutlichen, wird das Histogramm als Kreis visualisiert.



**Bild 6.23:** Histogrammaufbau

Die geringe Auflösung führt - wie bereits erwähnt - zu deutlicheren Peaks. Anschließend wird über alle Ergebnispaaare, die zum Peak beigetragen haben, zirkular gemittelt. Um Ergebnisse nicht indirekt auszuschließen, die an den Grenzen des Histogramms liegen, werden ebenfalls die benachbarten Histogrammstufen beim Mitteln berücksichtigt. Dazu werden aus dem Histogramm entsprechende Grenzen definiert. Mit Hilfe dieser Grenzen können alle zwischengespeicherten Ergebnisse durchsucht und lediglich die innerhalb der Grenzen liegenden Ergebnisse zur Mittelung herangezogen werden. Dieser Ablauf ist in Bild 6.24 visualisiert.

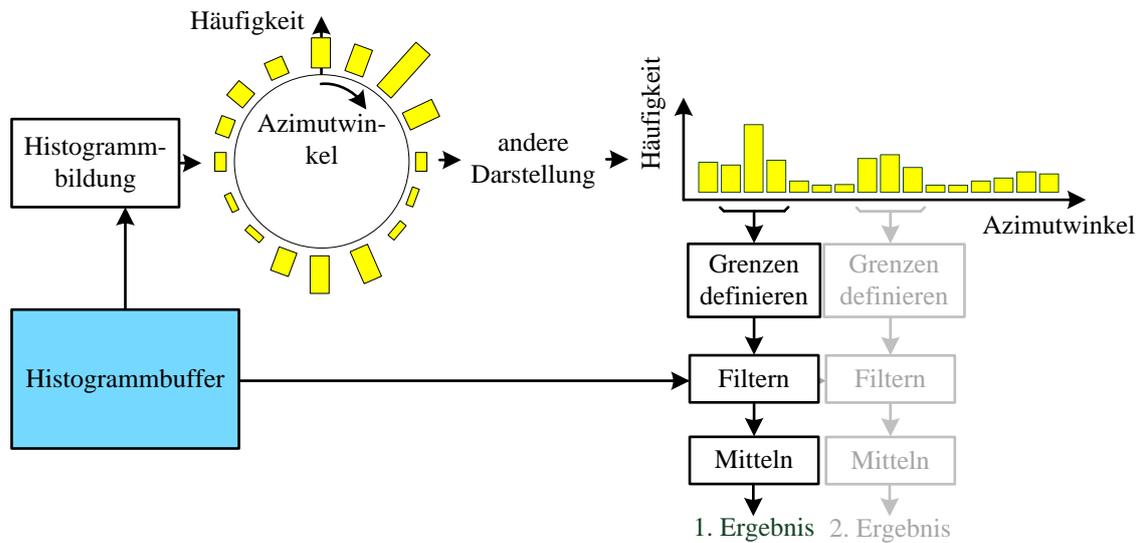


Bild 6.24: Histogrammanalyse

### 6.5.3 Ergebnismittelung pro Datensatz

Als weitere Stufe der Ergebnisaufbereitung ist das Mitteln über alle Ergebnisse eines Blocks denkbar. Diese Stufe würde dazu dienen, nah beieinander liegende Ergebnisse im Histogramm hervorzuheben. Dazu werden, falls die Ergebnisse unterschiedlicher Frequenzstützstellen in einem gewissen Rahmen liegen, die Ergebnisse zirkular gemittelt und entsprechend der Anzahl der zugrundeliegenden Winkel in den Zwischenspeicher des Histogramms gespeichert. Dies bewirkt, dass eine gewisse Peakbildung erzwungen wird. Allerdings ist diese Maßnahme durch die Verwendung des zweistufigen Histogramms nicht nötig. Die geringe Auflösung des Histogramms begünstigt bereits die Peakbildung. Somit kann diese Maßnahme - auch im Hinblick der Echtzeitfähigkeit und der Separierbarkeit von Quellen - vernachlässigt werden.

### 6.5.4 Merkmalsraum

Ebenfalls ist die Durchführung einer Ergebnisaufbereitung im Merkmalsraum denkbar. Dazu werden die Ergebnisse als Merkmal interpretiert und anschließend klassifiziert. Die Klasse repräsentiert dabei eine detektierte Quelle. Durch dieses Vorgehen ist es prinzipiell möglich, komplexe Verfahren zu implementieren, die unter anderem auch die Verfolgung von Quellen - Tracking - ermöglichen. Es ist allerdings zu beachten, dass die Methoden stark auf die Anwendung parametrisiert werden können, teils sogar müssen. Weiter sind diese Verfahren zumeist sehr komplex und sollten nicht auf stark schwankende Daten angewendet werden. Sie eignen sich somit als zweite Stufe, die Besonderheiten der entsprechenden Anwendung

berücksichtigen. Folglich werden diese Verfahren in dieser Arbeit nicht weiter betrachtet, da keine Spezialisierung vorgesehen ist.

### 6.5.5 Fehlerkorrektur

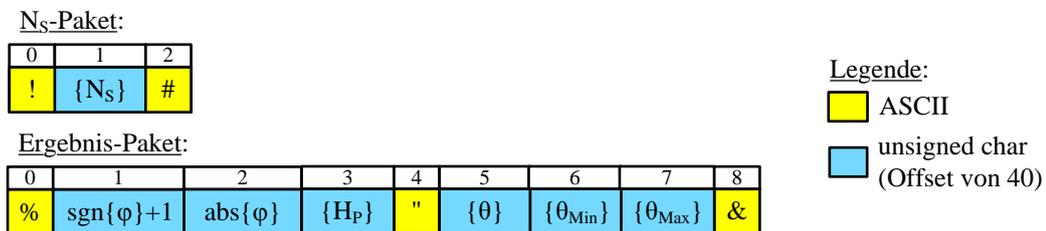
Als weiterer Aufbereitungsblock sei hier schon die Fehlerkorrektur erwähnt. Diese kann vor oder nach der Histogrammauswertung erfolgen. Für eine Korrektur nach der Histogrammauswertung spricht, dass die zugrundeliegenden Fehlerkurven auch nach der Histogrammauswertung gemessen worden sind. Allerdings ist eine Korrektur vor dem Histogramm vorzuziehen, da so, aufbauend auf korrekteren Ergebnissen, das Histogramm gebildet wird. Für weitere Details sei auf das Kapitel 9 verwiesen.

## 6.6 Ergebnisausgabe

Das DSP-Echtzeitsystem bietet prinzipiell zwei Möglichkeiten Daten zwischen dem System und einem Endgerät zu übertragen. Es handelt sich dabei um eine JTAG-Schnittstelle sowie um eine serielle Schnittstelle. Da jedoch die Übertragung über die JTAG-Schnittstelle langsam ist, erfolgt die Kommunikation über die serielle Schnittstelle.

Um die CPU des Echtzeitsystems jedoch nicht mit der Kommunikation auszulasten, ist ein schlankes Protokoll vonnöten. Dennoch sollten alle wichtigen Informationen wie Anzahl der Quellen, Azimut- und Elevationswinkel, Histogrammwert sowie Minima und Maxima des Elevationswinkels übertragen werden.

Um dies zu erreichen, werden lediglich die Steuerzeichen im Echtzeitsystem als ASCII-Zeichen codiert. Alle Werte werden hingegen in einem `unsigned char` verpackt. Die Auflösung wird dabei auf eine ganze Zahl begrenzt. Dies erscheint hinsichtlich der zu übertragenden Datenmenge und des systematischen Fehlers sinnvoll. Um Verwechslungen mit einem Steuerzeichen auszuschließen, werden alle Ergebniswerte um 40 erhöht. Das verwendete Protokoll ist in Bild 6.25 dargestellt.



**Bild 6.25:** Protokoll

Das Protokoll gliedert sich in zwei Pakete. Das  $N_S$  – *Paket* überträgt die Anzahl der Quellen bzw. die Anzahl der folgenden *Ergebnis – Pakete*. Das *Ergebnis – Paket* übermittelt alle wichtigen Informationen zum ermittelten Ergebnis. Die Information bezüglich Azimut- und Elevationswinkel werden zusätzlich mit einem Steuerzeichen getrennt.

## 6.7 Zusammenfassung - Das Konzept

In den vorherigen Abschnitten sind die wesentlichen Gedankengänge zum Konzept vorgetragen worden. Um die Übersicht zu gewähren, wird in diesem Abschnitt ein Überblick über das Gesamtkonzept gegeben. Grafisch ist das Konzept in Bild 6.26 dargestellt. Die Farben orientieren sich an denen, die im Konzeptansatz verwendet worden sind.

Nachdem alle Kanäle ausgelesen sind, erfolgt zunächst die Signalkonstruktion. Dies betrifft vor allem die Handhabung des Überhangs.

Anschließend erfolgt die Betrachtung der Energie, um sicherzustellen, dass auch aktive Quellen zum Datensatz beigetragen haben. Bei der Energieberechnung kann gleichzeitig die Fensterung des Signals durchgeführt werden.

Falls die Signalenergie größer als das Minimum ist, erfolgt der Übergang in den Spektralbereich. Dazu werden alle Frequenzgänge unter Berücksichtigung eines entsprechenden Fensters gebildet. Zur Auswahl der Frequenzstützstellen wird der gemittelte Amplitudengang bestimmt. Auf diesen gemittelten Amplitudengang wird das Fenster zur Frequenzauswahl angewendet.

Nach dem Übergang in den Spektralbereich wird das Maximum des gemittelten Amplitudengangs aller Kanäle gesucht. Danach wird überprüft, ob der Peak größer als das gewünschte Minimum ist. Ist dies der Fall, wird die Normierung, die Bildung der Autokorrelationsmatrix und der UCA-ESPRIT-Algorithmus mit den beschriebenen Modifikationen durchgeführt. Anschließend erfolgt die Fehlerkorrektur und die Speicherung im Histogrammbuffer. Um auszuschließen, dass der betrachtete Peak im nächsten Durchlauf abermals verwendet wird, wird der gemittelte Amplitudengang über dem aktuellen Bereich maskiert. Nach dieser Maskierung beginnt die Maximumsuche abermals, solange, bis das gefundene Maximum die gesetzte Schwelle unterschreitet.

Sind alle Peaks ausgewertet, folgt die Histogrammauswertung für den Fall, dass die Anzahl neuer Werte im Histogramm über einem gewissen Minimum liegt. Nach der Berechnung wird das Maximum gesucht. Ist dies größer als das gesetzte Minimum, wird das Mitteln der Werte durchgeführt, das Ergebnis zur Ergebnisausgabe zwischengespeichert und das Histogramm maskiert. Ist der Peak jedoch kleiner als das Minimum, folgt die Ergebnisausgabe und das Ende des Algorithmus für den aktuellen Block. Sobald ein neuer Block vorliegt, wird der geschilderte Ablauf wiederholt.

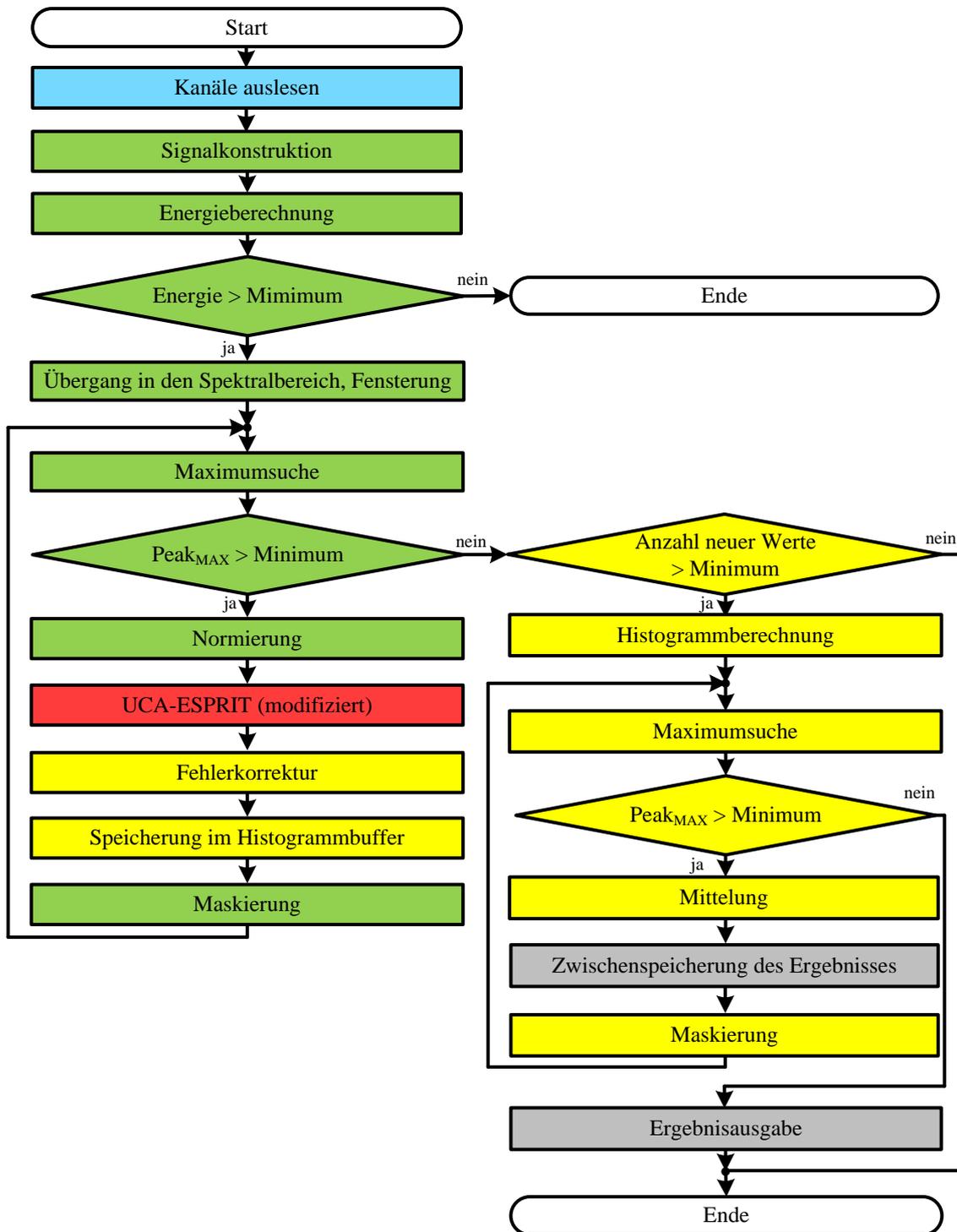


Bild 6.26: Konzept

# 7 Konzeptumsetzung

Das in Kapitel 6 entwickelte Konzept wird in diesem Kapitel umgesetzt. Dazu erfolgt zunächst die Vorstellung der wesentlichen Parameter und die Darstellung eventueller Abhängigkeiten. Anschließend werden Simulationen basierend auf generierten Mikrofonsignalen und Aufnahmen von Mikrofonsignalen durchgeführt. Sie dienen zum Nachweis der grundsätzlichen Funktionsfähigkeit und zur Variation und Bestimmung der Parameterwerte. Anschließend wird die hardwarenahe Implementierung erläutert und explizit auf Besonderheiten dieser eingegangen.

## 7.1 Parameter

Das im vorherigen Kapitel vorgestellte Konzept bietet eine Vielzahl an Parametern. Um eine Übersicht über diese zu erhalten, seien diese kurz vorgestellt:

Die Zusammenstellung des Datensatzes, der zur Berechnung herangezogen wird, kann über drei Parameter mit zwei Freiheitsgraden eingestellt werden:

- **N\_BLOCK**: Der Parameter **N\_BLOCK** gibt die EDMA-Blocklänge an.
- **N\_ALGO**: Dieser Parameter gibt die Länge des Blocks an, der zur Berechnung verwendet wird. Er sollte in Hinblick auf die Anwendung der FFT einer Potenz von Zwei entsprechen.
- **N\_OVERLAP**: Der Parameter **N\_OVERLAP** gibt die Anzahl der zu speichernden Werte für den nächsten Durchlauf an.

Für die drei Parameter besteht folgender Zusammenhang:

$$N\_ALGO = N\_OVERLAP + N\_BLOCK \quad (7.1)$$

Nach der Erstellung des Datensatzes folgen die Schritte bis einschließlich der Berechnung der Autokorrelationsmatrix. Dabei beeinflussen folgende Parameter den Algorithmus:

- **X\_ENERGY\_MIN**: Die minimale Energie, die für eine Berechnung im Signal erhalten sein muss, wird über diesen Parameter definiert.

- **X\_PEAK\_MIN**: Dieser Parameter gibt den minimalen Wert der Peaks an, die für eine Berechnung herangezogen werden.
- **f\_window**: Bei diesem Parameter handelt es sich weniger um einen Parameter als vielmehr um das Frequenzauswahlfenster. Allerdings lassen sich die Grenzen parametrisieren. Sie sollten entsprechend der Ausführungen aus Kapitel 6 gewählt werden - können jedoch beliebig variiert werden.
- **F\_SW, F\_DW**: Die Auswahl- und Maskierungsbreite im Spektralbereich werden über diese Parameter festgelegt und sollten entsprechend des verwendeten Fensters gewählt werden. Bei Wahl des Dolph-Tschebyscheff-Fensters mit entsprechender Parametrisierung (s. Abschnitt 6.3.2) ist ein Wert von Zwei sinnvoll.
- **N\_PEAKS**: Dieser Parameter gibt die maximale Anzahl an auszuwertenden Peaks an. Er sollte so gewählt werden, dass keinerlei Echtzeitprobleme auftreten können.

Der anschließende UCA-ESPRIT-Algorithmus wird stets mit der Annahme einer Quelle berechnet, dennoch ergeben sich drei Parameter:

- **M**:  $M$  gibt den höchsten betrachteten Phasen-Modus an, der zur Berechnung berücksichtigt werden soll. Folglich werden  $N_m = 2M + 1$  Phasen-Moden in die Berechnung mit einbezogen. Entsprechend der vorherigen Ausführungen aus Kapitel 6 ist ein Wert von  $M = 3$  sinnvoll.
- **EV\_ABS\_MAX**: Dieser Parameter beschreibt den maximal erlaubten Betrag des Eigenwertes  $\mu_{LS}$ . Dabei ist ein Wert von knapp über Eins sinnvoll. In dieser Arbeit wird ein Wert von 1,2 verwendet.
- **SVD\_VALUE\_DIVISOR**: Der Parameter SVD\_VALUE\_DIVISOR gibt das maximale Verhältnis zwischen erstem und zweitem Eigenwert an. Er sollte entsprechend den Ausführungen in Abschnitt 6.4.1 gewählt werden. In dieser Arbeit wird ein Wert von Zehn (eine Dekade) verwendet.

Die folgende Histogrammauswertung bestimmt wesentliche Eigenschaften wie Reaktionszeit und Robustheit des Systems und lässt sich wie folgt parametrisieren:

- **HIST\_ACCURACY**: Dieser Parameter bestimmt die Auflösung des Histogramms und sollte die Hälfte des minimalen Abstandes zweier Quellen, die voneinander unterschieden werden sollen, betragen.
- **N\_HIST**: N\_HIST gibt die Länge des Histogrammbuffers an, aus dem das Histogramm berechnet wird.
- **HIST\_PEAK\_MIN**: Dieser Parameter gibt den minimalen Wert eines zu berücksichtigenden Peaks im Histogramm an.

- **HIST\_SW**: Dieser Parameter gibt die halbe Anzahl der Stufen an, die neben der Stufe des Peaks zum Mitteln herangezogen werden. Entsprechend der Ausführungen aus Abschnitt 6.5.2 ist dieser Parameter zu Eins zu wählen.
- **HIST\_DW**: Die Breite der Maskierung wird durch diesen Parameter angegeben. Um den minimalen Abstand zweier Quellen nicht zu beeinträchtigen, sollte dieser Parameter ebenfalls zu Eins gewählt werden.
- **N\_HISTPEAKS**: Die maximale Anzahl an Ergebniswerten ist durch diesen Parameter begrenzt.
- **N\_HIST\_WAIT**: Dieser Parameter gibt an, wie viele neue Ergebniswerte mindestens im Histogramm enthalten sein müssen, um eine Histogrammauswertung durchzuführen. In dieser Arbeit wird er zu Eins gewählt, damit pro Block stets eine Histogrammauswertung durchgeführt wird.

## 7.2 Simulationen

In diesem Abschnitt werden Simulationen durchgeführt, um Aussagen über die grundsätzliche Funktionsfähigkeit des Systems und über die Auswirkungen entsprechender Parameter treffen zu können.

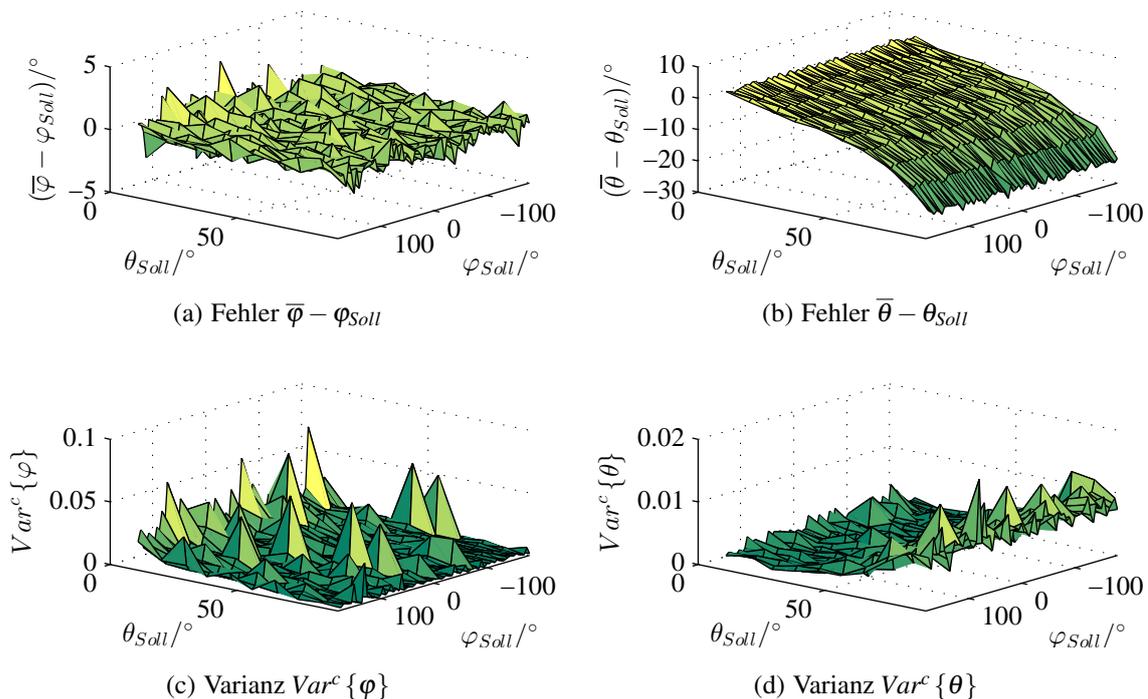
Falls nicht explizit andere Systemparameter angegeben werden, so beruhen die Simulationen auf den Systemparametern aus Tabelle 7.1. Die Standardabweichung des weißen gaußschen Rauschens liegt weiterhin für alle nachfolgenden Simulationen bei 0,01 (s. Abschnitt 6.4).

Parameter	Parameterwert	Parameter	Parameterwert
N_BLOCK	300	N_M (M)	7 (3)
N_OVERLAP	212	EV_ABS_MAX	1,2
N_ALGO	512	SVD_VALUE_DIVISOR	10
X_ENERGY_MIN	N_ALGO/256	N_HIST	50
X_PEAK_MIN	N_ALGO/512	N_HIST_WAIT	1
N_PEAKS	10	HIST_PEAK_MIN	5
		N_HISTPEAKS	5
		HIST_ACCURACY	10

**Tabelle 7.1:** Einstellung der Systemparameter (Simulationen)

### 7.2.1 Simulationen mit einer aktiven Quelle

Die schmalbandigen Simulationen haben einen systematischen Fehler aufgezeigt. Um zu überprüfen, ob dieser systematische Fehler auch in dem Gesamtsystem enthalten ist, wird eine aktive Quelle simuliert. Deren Mikrofonsignale werden entsprechend aller möglichen Einfallswinkel, wobei der Elevationswinkel mindestens  $10^\circ$  beträgt, variiert. Ein Einfallswinkel wird dabei für die komplette Testsequenz konstant gehalten. Die Ergebnisse sind in Bild 7.1 dargestellt.



**Bild 7.1:** Fehler- und Varianzverläufe über  $\varphi$  und  $\theta$

Der systematische Fehler im Azimutwinkel ist nahezu nicht mehr erkennbar. Er wird deutlich von einem zufälligen Fehler überlagert. Eine Steigerung mit dem Elevationswinkel ist ebenfalls nicht mehr deutlich ersichtlich. Begründet ist dies in erster Linie durch den unterschiedlichen systematischen Fehler bei verschiedenen Frequenzen. Diese Frequenzstützstellen weisen jeweils einen unterschiedlichen systematischen Fehler auf. Die Varianz im Azimutwinkel fällt gering aus, ist jedoch deutlich höher als bei den komplexwertigen, schmalbandigen Simulationen.

Der Fehler im Elevationswinkel zeigt ansatzweise die Systematik der schmalbandigen Simulationen. Vor allem die Steigerung mit dem Elevationswinkel ist im Gegensatz zum Azimutwinkel noch erkennbar. Allerdings zeigt sich ein weiterer systematischer Fehler im

Elevationswinkel ab einem Wert von über  $80^\circ$ . Der Fehler nimmt ab diesem Punkt deutlich zu. Erklärbar ist dies durch die Streuung der Eigenwerte und durch die Deckelung des Ergebniswertes (s. Abschnitt 6.4.4). Dieser Effekt bewirkt einen maximalen Fehler von  $20^\circ$ . Die Varianz des Elevationswinkels steigt mit dem Elevationswinkel. Dies lässt sich ebenfalls durch die Struktur der Eigenwerte erklären (s. Abschnitt 6.4.4).

Die Zunahme der Fehler und der Varianzen ist einerseits in der Variation des SNRs im Gegensatz zu den schmalbandigen, komplexwertigen Simulationen begründet. Durch die realen Testsequenzen variiert die Amplitude des Nutzsignals stark. Dies hat zwangsläufig eine Veränderung des SNRs und entsprechend der Fehler und der Varianzen zur Folge. Andererseits entspricht der Übergang in den Spektralbereich mit anschließender Auswertung eines schmalen Bandes nicht gänzlich den idealen theoretischen Bedingungen der Simulationen mit schmalbandigen, komplexwertigen Signalen.

Trotz der Erhöhung können die Fehler und die Varianzen insgesamt als klein angesehen werden. Die größeren Abweichungen im Elevationswinkel können aufgrund der Systematik durch eine anschließende Fehlerkorrektur (s. Kapitel 9) behoben werden.

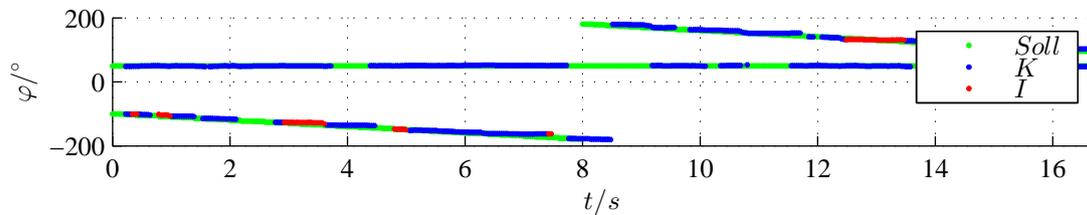
## 7.2.2 Simulationen mit zwei aktiven Quellen

Der vorherige Abschnitt hat die Genauigkeit des Systems bei einer aktiven Quelle betrachtet. Insgesamt ist der auftretende Fehler - ausgenommen bei Elevationswinkeln von über  $80^\circ$  - sehr gering. Die Varianz fällt ebenfalls sehr klein aus.

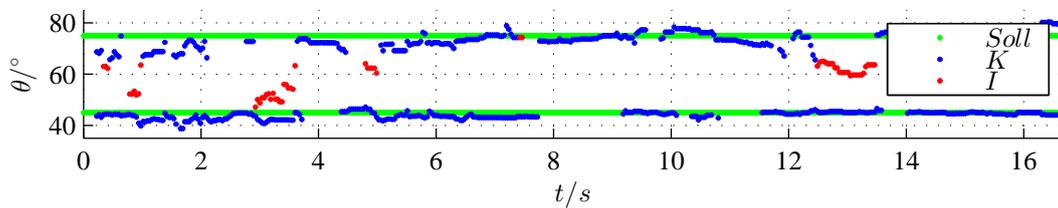
Um zu überprüfen, wie das System auf die Anwesenheit zweier aktiver Quellen reagiert, werden entsprechende Simulationen durchgeführt. Um auch Aussagen über das System bei nicht statischen Quellen treffen zu können, vollzieht eine Quelle eine Kreisbewegung im Azimutwinkel und stellt somit eine dynamische Quelle dar. Um die zeitliche Synchronisation zwischen Ergebnis- und Sollwertaufnahme gewährleisten zu können, wird die Simulation mit generierten Mikrofonsignalen durchgeführt. Die Ergebnisse sind in Bild 7.2 dargestellt. Dabei beschreibt die Menge  $K$  alle korrekten Ergebnisse (maximale Abweichung vom Sollwert  $10^\circ$ ),  $I$  alle inkorrekten Ergebnisse (minimale Abweichung vom Sollwert  $10^\circ$ ) und  $Soll$  die wahren Werte.

Bei Begutachtung der Ergebnisse aus Bild 7.2, lassen sich folgende Erkenntnisse ableiten:

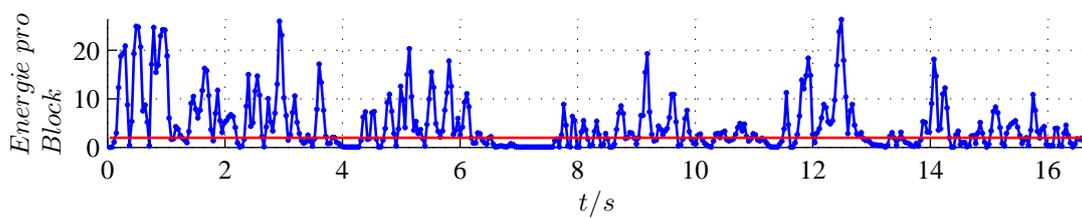
Der Azimutwinkel ist relativ genau detektierbar. Für die statische Quelle ist der Fehler nie größer als  $10^\circ$ . Für die dynamische Quelle ist hingegen ein Treppenstufenverlauf erkennbar und die Ergebnisse sind äußerst selten der Menge  $I$  zugeordnet. Der Treppenstufenverlauf lässt sich dadurch begründen, dass es einerseits zu einer zeitlichen Verzögerung aufgrund der Histogrammauswertung kommt und sich entsprechende Zwischenergebnisse folglich erst später bemerkbar machen. Andererseits bewirkt das Histogramm, dass ein Ergebnis länger



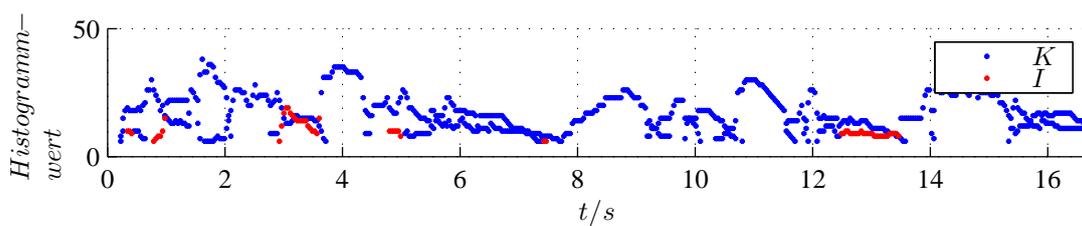
(a) Azimutwinkel



(b) Elevationswinkel



(c) Energie



(d) Histogrammwert

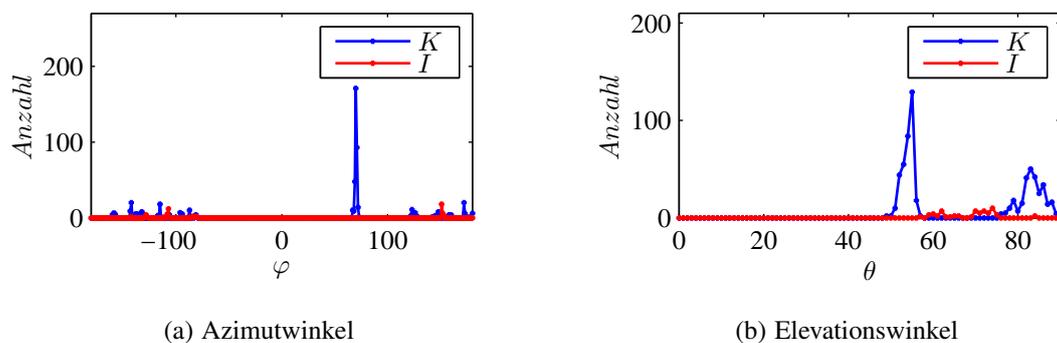
**Bild 7.2:** Simulationsergebnisse bei zwei aktiven Quellen mit generierten Signalen

erhalten bleibt, wenn keine neuen Zwischenergebnisse erzeugt werden. Die Zuordnung zur Menge  $I$  ist hingegen durch den größeren Fehler im Elevationswinkel erklärbar.

Der Elevationswinkel weist hingegen eine deutlich größere Streuung auf. Diese ist stark abhängig vom wahren Wert. Die Streuung nimmt deutlich mit dem Wert des Elevationswinkels zu. Erklärbar ist dies abermals mit der Kennlinie des Elevationswinkels.

Die Histogrammwerte der Peaks verhalten sich entsprechend der Erwartungen. Die Werte unterliegen Schwankungen, die sich durch die Streuung der Zwischenergebnisse begründen lassen. Allerdings ist ein deutlicher Abfall des Histogrammwerts erkennbar, falls für einen längeren Zeitraum kein Zwischenergebnis berechnet wird. Ursächlich hierfür sind Pausen in den Testsequenzen und eine dadurch bedingte zu geringe Signalenergie.

Wird das Histogramm über die Ergebniswerte dieser Simulation gebildet, ergeben sich die Ergebnisse aus Bild 7.3.

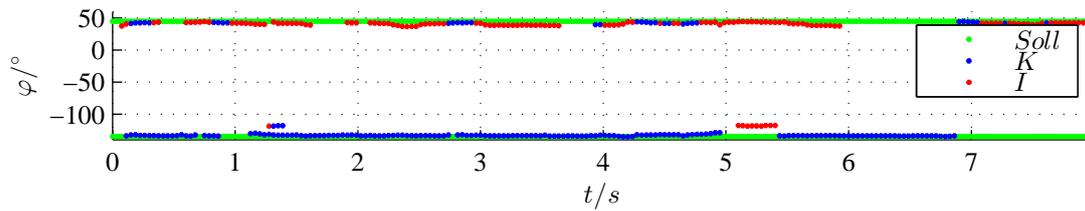


**Bild 7.3:** Ergebnishistogramme (Auflösung 1 °)

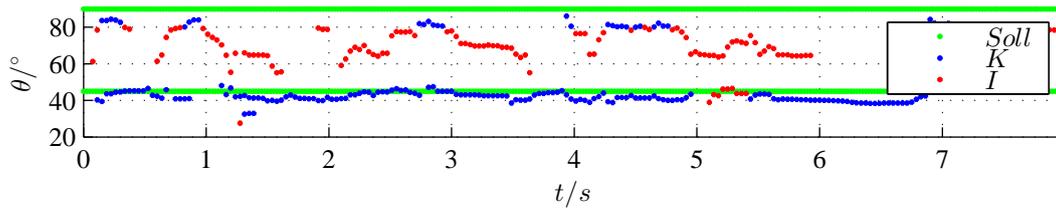
Werden die Histogramme begutachtet, sind deutliche Peaks für die statische Quelle zu erkennen. Für die dynamischen Quellen bildet sich ebenfalls im Elevationswinkel ein Peak aus. Für den Azimutwinkel ergibt sich jedoch eine gewisse Streuung. Allerdings sind durch den Treppenstufenverlauf kleine Peaks erkennbar.

Weiter ist die Abhängigkeit der Streuung des Elevationswinkels vom Sollwert des Elevationswinkels erkennbar.

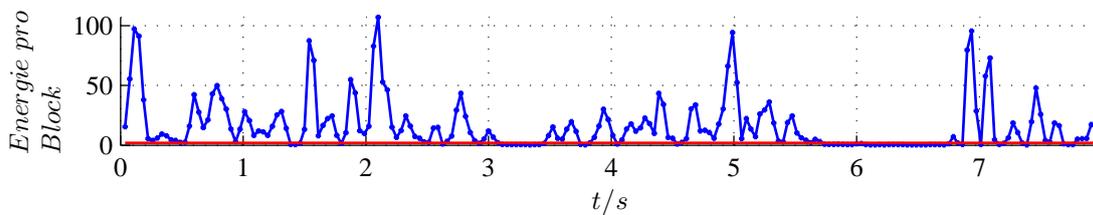
Um zu überprüfen, ob dieselben Erkenntnisse auch für Simulationen, basierend auf Aufnahmen, ableitbar sind, sind entsprechende Simulationen durchgeführt worden. Es ist zu beachten, dass zwei statische Quellen verwendet werden, da sonst keine Synchronisation zwischen Soll- und Ergebniswert gewährleistet werden kann. Die Ergebnisse sind in Bild 7.4 dargestellt.



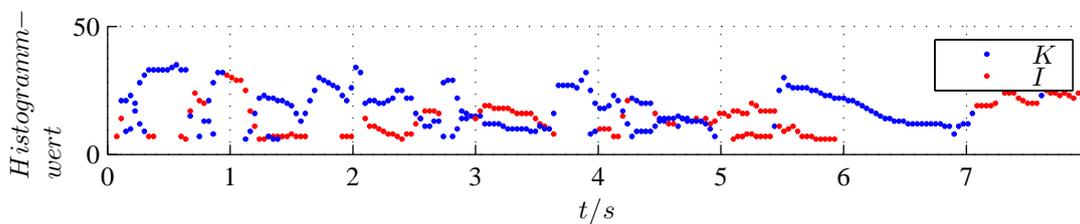
(a) Azimutwinkel



(b) Elevationswinkel



(c) Energie



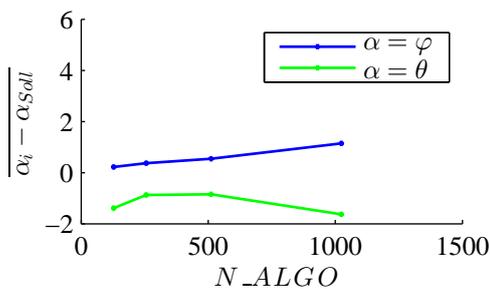
(d) Histogrammwert

**Bild 7.4:** Simulationsergebnisse bei zwei aktiven Quellen mit Aufnahmen

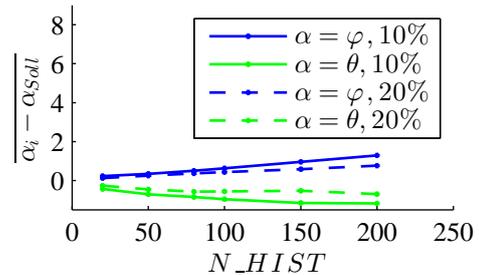
Die Begutachtung der Ergebnisse zeigt, dass die vorherigen Aussagen ebenfalls auf diese Simulation zutreffen. Die Ergebnisse des Azimutwinkels sind konstant. Allerdings erscheint der Fehler deutlich größer, da viele Ergebnisse der Menge  $I$  zugeordnet sind. Diese Zuordnung lässt sich jedoch durch den großen Fehler im Elevationswinkel erklären. Die Streuung des Elevationswinkels nimmt mit dem Wert dieses Winkels zu. Der Verlauf der Histogrammwerte unterliegt ebenso Schwankungen und nimmt deutlich ab, falls die Energie unter das entsprechende Minimum fällt. Neben diesen Effekten ist vor allem im Elevationswinkel ein zusätzlicher, konstanter Fehler erkennbar. Ursache hierfür sind die Abweichungen der realen von der idealen Arraystruktur. Für weitere Ausführungen sei auf das Kapitel 9 verwiesen.

### 7.2.3 Variation der Parameter

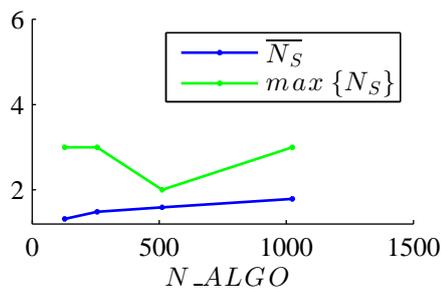
Die größten Freiheitsgrade in den Parametern ergeben sich hinsichtlich der Länge des Datensatzes ( $N\_ALGO$ ) und in der Histogrammauswertung. Bei der Histogrammauswertung handelt es sich insbesondere um die Länge des Histogrammbuffers  $N\_HIST$  und um das Verhältnis des Minimums eines auszuwertenden Peaks zu der Länge des Histogrammbuffers  $HIST\_PEAK\_MIN / N\_HIST$ . Sie bestimmen vor allem die Robustheit und die Reaktionszeit des Systems. Um diese Parameterwerte in Verbindung mit den entsprechenden Eigenschaften zu bringen, werden die vorherigen Testszenerien mit entsprechender Variation der Parameter simuliert. Die Ergebnisse, basierend auf generierten Mikrofonsignalen und einer statischen und einer dynamischen Quelle, sind in Bild 7.5 abgebildet. Bild 7.6 zeigt die Ergebnisse der Aufnahmen. Bei der Variation der Länge des betrachteten Datensatzes  $N\_ALGO$  wird die Histogrammbufferlänge auf  $N\_HIST=50$  mit einem Verhältnis von  $HIST\_PEAK\_MIN / N\_HIST=10\%$  konstant gehalten. Wird hingegen  $N\_HIST$  und  $HIST\_PEAK\_MIN / N\_HIST$  variiert, wird  $N\_ALGO$  auf einen Wert von 512 festgelegt.



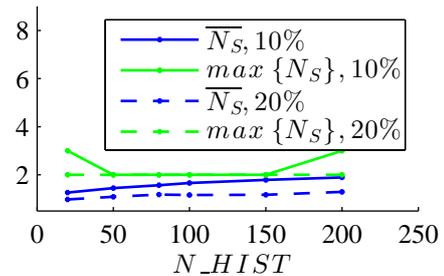
(a) Mittlere Abweichung über  $N\_ALGO$



(b) Mittlere Abweichung über  $N\_HIST$

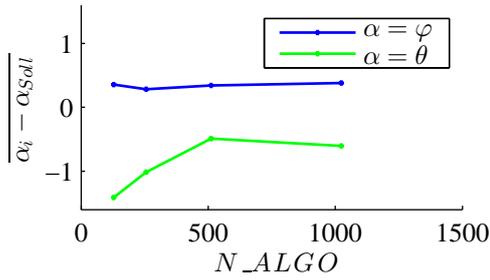


(c) Ergebnisanzahl über  $N\_ALGO$

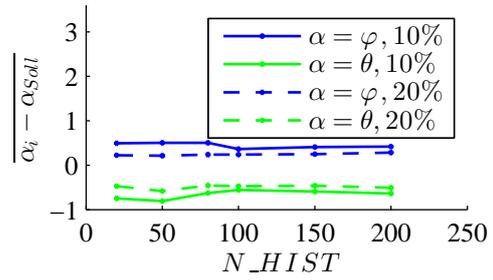


(d) Ergebnisanzahl über  $N\_Hist$

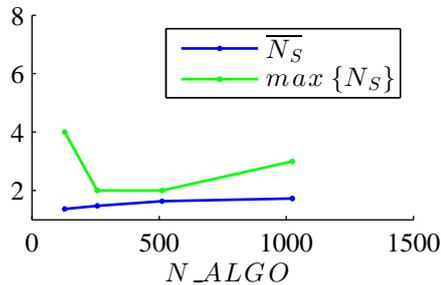
**Bild 7.5:** Ergebnisse bei zwei aktiven Quellen (statisch und dynamisch)



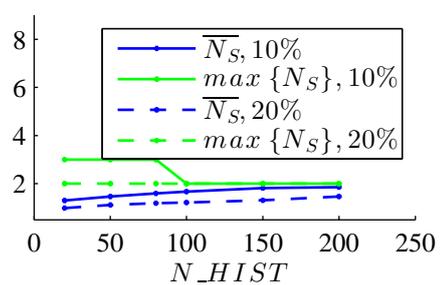
(a) Mittlere Abweichung über  $N_{ALGO}$



(b) Mittlere Abweichung über  $N_{HIST}$



(c) Ergebnisanzahl über  $N_{ALGO}$



(d) Ergebnisanzahl über  $N_{Hist}$

**Bild 7.6:** Ergebnisse bei zwei aktiven Quellen (Aufnahmen)

Bei Betrachtung der Ergebnisbilder aus Bild 7.5 und 7.6 sind deutliche Unterschiede zwischen den einzelnen Parametereinstellungen erkennbar. Zwischen den generierten Signalen und Aufnahmen ergeben sich viele Gemeinsamkeiten, andererseits auch einige Unterschiede.

Die Verlängerung der Datensatzlänge führt zu ähnlichen Ergebnissen zwischen den generierten Signalen und Aufnahmen (s. Unterbilder (a) und (c)) außer beim Fehler im Azimutwinkel. Dieser steigt bei den generierten Signalen mit der Datensatzlänge. Der Grund liegt in der Treppenstufenbildung. Der Fehler im Elevationswinkel und die maximale Anzahl an detektierten Quellen verlaufen parabelförmig und haben jeweils ihr Minimum bzw. den korrekten Wert bei einer ungefähren Datensatzlänge von 512. Dieses Verhalten ergibt sich wie folgt: Aufgrund der Verwendung der DFT steigt die Auflösung im Frequenzbereich mit der Datensatzlänge. Dies führt dazu, dass die Wahrscheinlichkeit zweier Quellen in einem betrachteten Peak sinkt. Folglich sind die Fehler geringer. Allerdings führt die Verlängerung des Datensatzes dazu, dass die generelle Wahrscheinlichkeit zweier aktiver Quellen in einem Signal mit der Datensatzlänge steigt. Ebenfalls ist die Stationarität des Signals ab einer gewissen Datensatzlänge nicht mehr gegeben. Da der Fehler des einen Effektes mit der Verlängerung der Datensatzlänge steigt und der andere sinkt, ergeben sich die parabelförmigen Verläufe. Die mittlere Anzahl an detektierten Quellen steigt aufgrund der größeren generellen Wahrscheinlichkeit zweier aktiver Quellen im Signal mit der Datensatzlänge.

Die Variation der Parameter der Histogrammanalyse und die Betrachtung der Ergebnisan-

zahl ergibt ebenfalls ähnliche Verläufe zwischen generierten Signalen und Aufnahmen (s. Unterbilder (d)). Die mittlere Anzahl der Ergebnisse steigt mit der Histogrammbufferlänge. Dies lässt sich dadurch erklären, dass mit der Steigerung der Histogrammbufferlänge auch mehr Zwischenergebnisse mehrerer Quellen ausgewertet werden können. Die Wahrscheinlichkeit des Vorhandenseins der Ergebnisse zweier Quellen im Histogrammbuffer vergrößert sich somit mit der Histogrammbufferlänge. Das die Verläufe mit einem Verhältnis von  $HIST\_PEAK\_MIN / N\_HIST=10\%$  einen höheren Wert aufzeigen, ist in der höheren Empfindlichkeit begründet. Für das Maximum sind besonders erhöhte Werte bei einer kleineren Histogrammbufferlänge erkennbar. Dies ist durch das entsprechend zu kleine Minimum eines berücksichtigten Histogrammpeaks erklärbar.

Die Betrachtung der Fehlerkurven (s. Unterbilder (b)) bei der Variation der Parameter der Histogrammanalyse zeigt hingegen in den Verläufen Unterschiede zwischen generierten Testsequenzen und Aufnahmen. Bei den generierten Testsequenzen steigt der Fehler mit der Zunahme der Histogrammbufferlänge sowie mit der Verringerung des Verhältnisses  $HIST\_PEAK\_MIN / N\_HIST$ . Ein konstantes Niveau zeigen sie hingegen bei der Verwendung der Aufnahmen. Allerdings ist ebenfalls die Erhöhung des Fehlers bei Verringerung des Verhältnisses  $HIST\_PEAK\_MIN / N\_HIST$  zu erkennen. Begründet ist dieser Unterschied in der Treppenstufenbildung bei der Lokalisierung der dynamischen Quelle. Durch Erhöhung der Histogrammbufferlänge wird das System träger und die Treppenstufenbildung verstärkt sich. Dies führt zu einem höheren Fehler. Das die Fehler mit der Verringerung des Verhältnisses  $HIST\_PEAK\_MIN / N\_HIST$  steigen, ist mit der entsprechend höheren Empfindlichkeit des Systems zu Lasten der Genauigkeit zu erklären.

Entsprechend der aufgezeigten Verläufe können die Eigenschaften des Systems parametrisiert werden. Als guter Kompromiss zwischen Genauigkeit und Empfindlichkeit ist die Wahl von  $N\_ALGO=512$ ,  $N\_HIST=50$  und ein Verhältnis  $HIST\_PEAK\_MIN / N\_HIST$  zwischen 10 % und 20 % sinnvoll.

## 7.3 Implementierung

In diesem Abschnitt wird die hardwarenahe Implementierung des entwickelten Algorithmus vorgestellt. Dabei sind die Grundlagen aus Kapitel 2 berücksichtigt worden. Für eine entsprechende Echtzeitfähigkeit ist der UCA-ESPRIT-Algorithmus modifiziert worden. Für aufwendige Berechnungen, wie zum Beispiel die FFT oder komplexe Matrixmultiplikationen, wird die *TI C67x DSPLib* verwendet. Um einen effizienten Algorithmus zu erhalten, wird die -O3 Optimierung verwendet. Die Abtastfrequenz wird zu  $f_s = 8 \text{ kHz}$  gewählt (s. Abschnitt 5.4).

### 7.3.1 Übersicht über die Programmstruktur

Das Einlesen der Mikrofonsignale erfolgt über den EDMA-Kontroller. Der entsprechende Algorithmus ist in die beschriebene Grundarchitektur der EDMA-Programmierung entsprechend Abschnitt 2.3 eingebettet. Nach dem Empfang eines Datenblocks erfolgt die Berechnung des Algorithmus. Das Flussdiagramm ist in Bild 7.7 abgebildet und spiegelt die konkrete Realisierung des Konzeptes aus Bild 6.26 wieder. Dieses Grundgerüst ruft eine Vielzahl an Funktionen auf, die nachfolgend näher erläutert werden. Auf die in gelb markierten Blöcke wird, aufgrund ihrer Komplexität, besonders eingegangen. Es handelt sich dabei um die Durchführung des UCA-ESPRIT-Algorithmus und der Histogrammanalyse.

Die Parametereinstellung und Berechnung der Beamformer und Fenster erfolgt in *MATLAB* und wird in das Projekt anschließend eingebunden. Dabei handelt es sich um die Dateien `global_param.h` und `globals.c` (s. Anhang B).

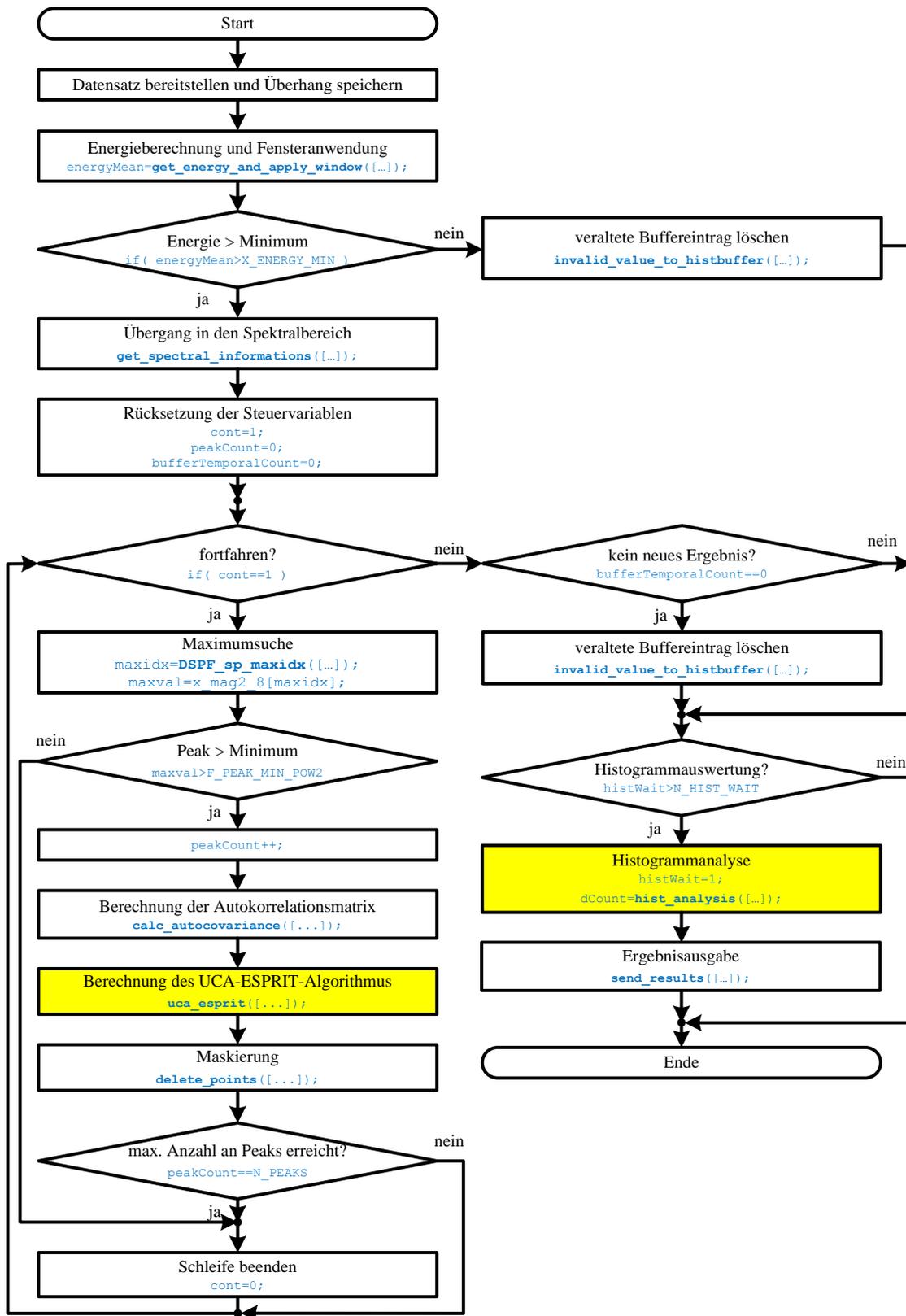


Bild 7.7: Ablaufdiagramm der Lokalisierung

### 7.3.2 Funktionen

Die verwendeten Funktionen der Implementierung - Dateien `algo_func.h`, `algo_func.c`, `svdcmp_opt.h`, `svdcmp_opt.c`, `rs232_comm.h` und `rs232_comm.c` (s. Anhang B) - seien im Folgenden kurz erläutert:

**Energieberechnung und Fensteranwendung** Die Energieberechnung und Fensteranwendung erfolgt in der Funktion

```
float get_energy_and_apply_window([...])
```

und wird mittels entsprechender Schleifen durchgeführt. Das Fenster wird mit *MATLAB* berechnet und als statisches Array angelegt.

**Übergang in den Spektralbereich** Der Übergang in den Spektralbereich wird mit der Funktion

```
void get_spectral_information([...])
```

durchgeführt. Dabei wird auf die entsprechende Assemblerfunktion der *TI C67x DSPLib* (s. Abschnitt 2.5) zurückgegriffen. Anschließend wird das Quadrat des Amplitudengangs berechnet und das Quadrat des Frequenzauswahlfensters angewendet. Dieses ist als statisches Array angelegt.

**Normierung und Berechnung der Autokorrelationsmatrix** Die Normierung und Berechnung der Autokorrelationsmatrix erfolgt in der Funktion

```
void calc_autocovariance([...])
```

Dabei wird im Wesentlichen die komplexe Matrizenmultiplikation der *TI C67x DSPLib* verwendet (s. Abschnitt 2.5). Damit die Berechnung mit Hilfe der Assemblerfunktion erfolgen kann, muss `F_SW` größer als Eins und damit alle Bedingungen dieser Funktionen erfüllt sein. Ist dies nicht der Fall, wird eine entsprechende C-Funktion verwendet. Auf den Proportionalitätsfaktor  $1/N^2$  wird verzichtet.

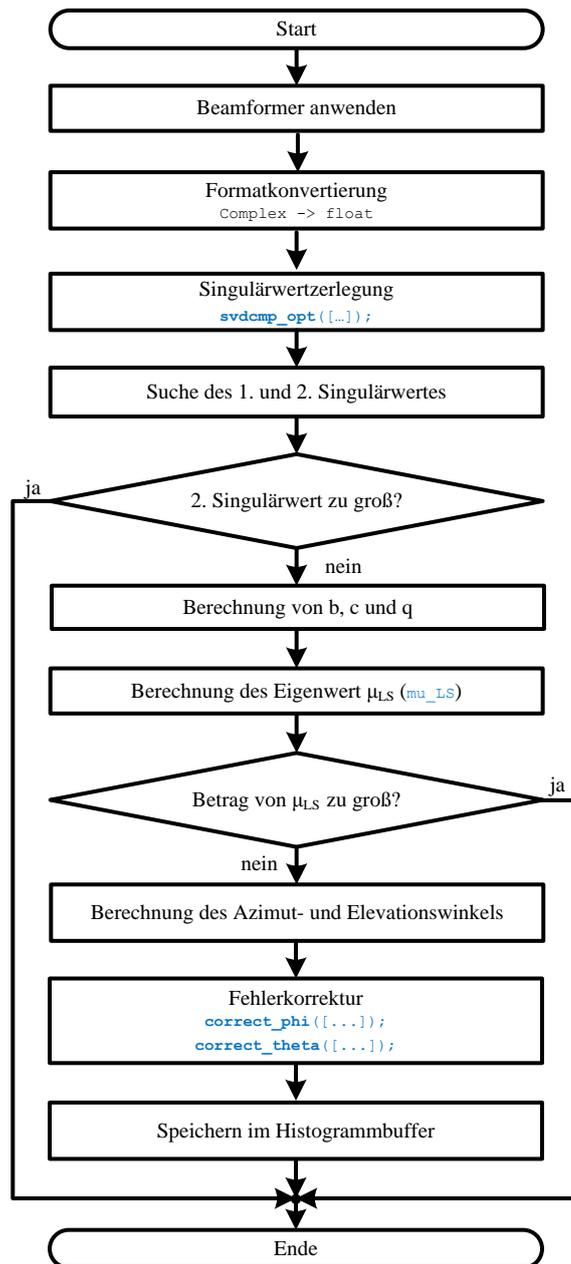
**Maskierung** Um einen Peak im nächsten Schleifendurchlauf nicht mehr zu berücksichtigen, wird dieser entsprechend ausgeblendet bzw. maskiert. Die verwendete Funktion lautet:

```
void delete_points([...])
```

**UCA-ESPRIT** Die Berechnung des UCA-ESPRIT-Algorithmus erfolgt in der Funktion

```
void uca_esprit ([...]).
```

Ein Ablauf dieser Funktion ist in Bild 7.8 dargestellt.



**Bild 7.8:** Ablaufdiagramm des UCA-ESPRIT-Algorithmus

Die Beamformer sowie weitere Vektoren und Matrizen, die für die Berechnung vonnöten sind, sind statisch angelegt.

Die Berechnung der Eigenwertanalyse erfolgt mit Hilfe der Singulärwertanalyse. Durch Definition ist die Autokorrelationsmatrix  $\mathbf{R}_{yy,UCA}$  hermitisch (s. Abschnitt 3.3). Ebenfalls ist folglich der Matrixausdruck  $\mathbf{B}_U^\dagger \mathbf{R}_{yy,UCA} \mathbf{B}_U$  hermitisch, da

$$\mathbf{B}_U^\dagger \mathbf{R}_{yy,UCA} \mathbf{B}_U = \left( \mathbf{B}_U^\dagger \mathbf{R}_{yy,UCA} \mathbf{B}_U \right)^\dagger \quad (7.2)$$

gilt. Betrachtet man anschließend lediglich den Realteil, so handelt es sich zwangsläufig um eine reelle, symmetrische Matrix. Weiter ist diese Matrix entsprechend den Überlegungen aus Abschnitt 3.3 positiv semidefinit, wodurch die Eigenwertzerlegung durch eine Singulärwertzerlegung ersetzt werden kann (s. Anhang A.2).

Durch die Verwendung der Singulärwertzerlegung kann auf einen bekannten Algorithmus zurückgegriffen werden. Dieser hat erstmalig 2004 in [5] innerhalb des Forschungsprojektes MERIT Anwendung gefunden und ist seitdem stets verwendet worden. Er entstammt dem Buch *Numerical Recipes* [25] und basiert auf der Householder-Transformation<sup>4</sup> und der QR-Zerlegung<sup>5</sup>. Er kann als erprobt angesehen werden und wird folglich auch in dieser Arbeit verwendet. Für weitere Details sei auf [5], [25] oder [31] verwiesen. Für einen weiteren Geschwindigkeitsvorteil sind alle zuvor dynamisch angelegten Hilfsarrays statisch angelegt worden. Die entsprechende Funktion ist unter dem Namen

```
void svdcmp_opt ([...])
```

zu finden.

Als weiteren wesentlichen Schritt ist das Lösen der Gleichung 5.91

$$\begin{bmatrix} \Re(\mathbf{B} + \mathbf{C}) & \Im(\mathbf{C} - \mathbf{B}) \\ \Im(\mathbf{B} + \mathbf{C}) & \Re(\mathbf{B} - \mathbf{C}) \end{bmatrix} \begin{bmatrix} \Re \hat{\Psi}_{LS} \\ \Im \hat{\Psi}_{LS} \end{bmatrix} = \begin{bmatrix} \Re \mathbf{Q} \\ \Im \mathbf{Q} \end{bmatrix}$$

durchzuführen. Da die Annahme einer Quelle getroffen wird, ergeben sich die Eigenwerte ohne anschließende Eigenwertanalyse. Die Matrizen  $\mathbf{B}$ ,  $\mathbf{C}$  und  $\mathbf{Q}$  reduzieren sich auf die komplexwertigen, skalaren Größen  $b$ ,  $c$  und  $q$ :

$$\begin{bmatrix} \Re(b + c) & \Im(c - b) \\ \Im(b + c) & \Re(b - c) \end{bmatrix} \begin{bmatrix} \Re \hat{\mu}_{LS} \\ \Im \hat{\mu}_{LS} \end{bmatrix} = \begin{bmatrix} \Re q \\ \Im q \end{bmatrix} \quad (7.3)$$

<sup>4</sup>Die Householder-Transformation transformiert mittels Spiegelung einen Spalten- oder Zeilenvektor auf das Vielfache eines Einheitsvektors.

<sup>5</sup>Die QR-Zerlegung zerlegt eine Matrix  $\mathbf{A}$  in das Produkt  $\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}$ , wobei  $\mathbf{Q}$  eine unitäre und  $\mathbf{R}$  eine obere Dreiecksmatrix ist. Diese Zerlegung kann unter anderem mit der Householder-Transformation erfolgen.

Die Lösung dieses Gleichungssystems lässt sich mit elementaren Umformungen wie folgt angeben:

$$\Re \hat{\mu}_{LS} = \frac{\Re q(\Im b + \Im c) - \Im q(\Re b + \Re c)}{|c|^2 - |b|^2} \quad (7.4)$$

$$\Im \hat{\mu}_{LS} = \frac{\Im q(-\Im b + \Im c) - \Re q(\Re b - \Re c)}{|c|^2 - |b|^2} \quad (7.5)$$

Die Quadrate der Beträge können mit

$$|c|^2 - |b|^2 = (\Im c)^2 + (\Re c)^2 - (\Im b)^2 - (\Re b)^2 \quad (7.6)$$

berechnet werden. Somit lassen sich die Ergebnisse mittels dieser Formeln aus Real- und Imaginärteil direkt bestimmen.

Die Fehlerkorrektur erfolgt in den Funktionen

```
float correct_phi([...]) und
float correct_theta([...])
```

Für eine genauere Beschreibung sei auf das Kapitel 9 verwiesen.

**Zweistufige Histogrammanalyse** Die Histogrammauswertung erfolgt mit der Funktion

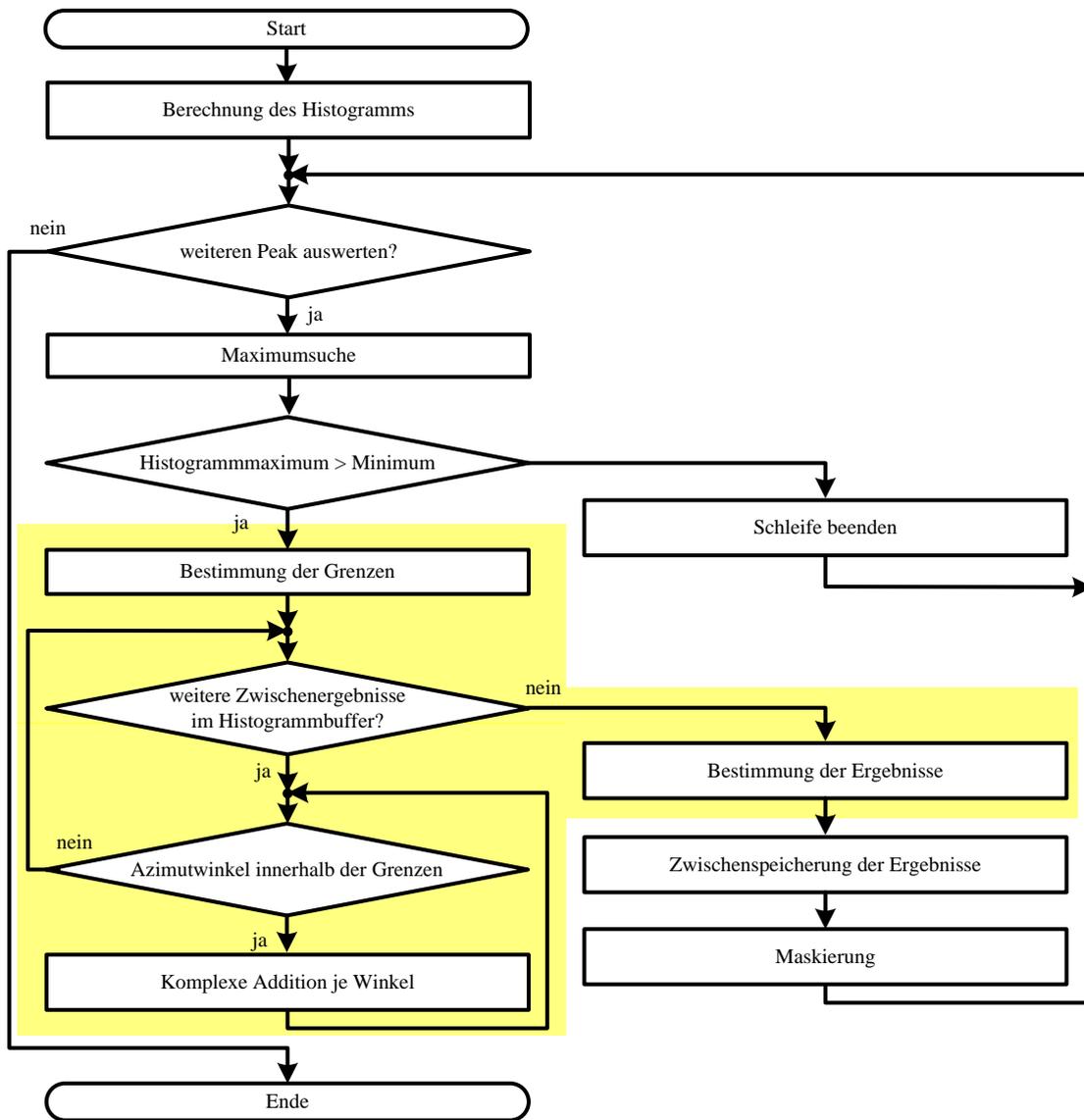
```
int hist_analysis([...]).
```

Der Ablauf dieser Funktion ist in Bild 7.9 dargestellt. Dabei ist das Mitteln mit der Farbe Gelb hinterlegt. Die Bestimmung des Minimums und Maximums des Elevationswinkels findet ebenfalls in dieser Funktion statt. Dieser Vorgang wird jedoch aus Gründen der Übersichtlichkeit nicht im Ablaufdiagramm dargestellt.

Um das Histogramm auch in den Fällen zu aktualisieren, in denen keine aktive Quelle vorliegt, existiert die Funktion

```
void invalid_value_to_histbuffer([...]),
```

die ungültige Werte in den Histogrammbuffer schreibt und somit veraltete Zwischenergebnisse löscht.



**Bild 7.9:** Ablaufdiagramm der zweistufigen Histogrammanalyse

**Ergebnisausgabe und Kommunikation** Die Ergebnisausgabe erfolgt in der Funktion

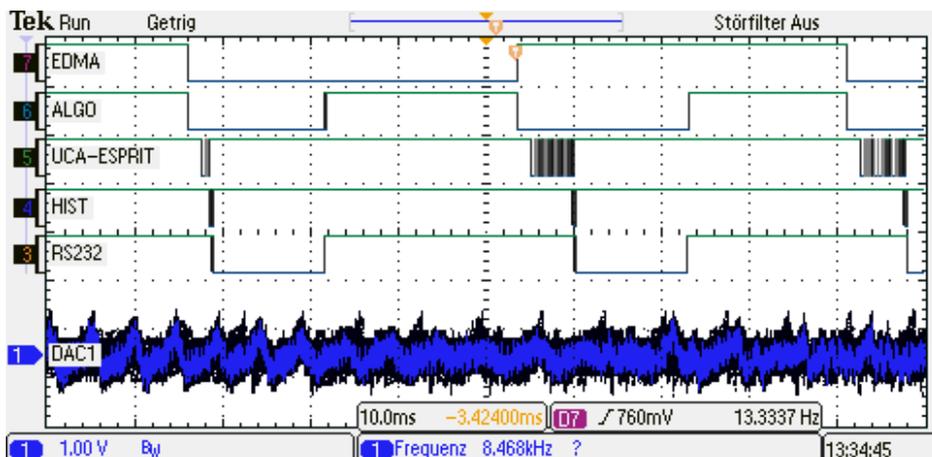
```
int send_results([...]).
```

In dieser werden die entsprechende Konvertierung und Übergabe an die RS232-Schnittstelle durchgeführt. Dabei wird das Protokoll aus Abschnitt 6.6 verwendet.

Als Übertragungsrate wird 9600 Baud gewählt. Es handelt sich um die maximale Übertragungsrate des Systems, die nahezu keine Übertragungsfehler bei einer längeren RS232-Kabellänge gewährleistet. Weiter werden acht Datenbits, ein Stopbit, ein Paritätsbit übertragen.

### 7.3.3 Nachweis der Echtzeitfähigkeit

Um den Nachweis der Echtzeitfähigkeit zu liefern, werden die Signalausgänge des DSPs entsprechend geschaltet und mit einem Oszilloskop aufgenommen. Bild 7.10 zeigt einen Verlauf der entsprechenden Signalausgänge bei aktiven Quellen und einer Algorithmuslänge von  $N_{\text{ALGO}}=512$ . Die EMMA-Blocklänge beträgt  $N_{\text{BLOCK}}=312$ ; die Histogrammbufferlänge beträgt  $N_{\text{HIST}}=50$ . Dargestellt wird die EDMA-Länge, wobei jeder Pegel eine EDMA-Länge repräsentiert. Die Länge des Gesamtalgorithmus (ALGO), die Länge der UCA-ESPRIT-Berechnung (UCA-ESPRIT), die Länge der Histogrammanalyse (HIST) und die Länge der Ergebnisübertragung (RS232) werden neben dem durchgeschleiften Signal des ersten Kanals angezeigt. Die Dauer der Nullpegel zeigt die Dauer der entsprechenden Berechnungsschritte an.



**Bild 7.10:** Nachweis der Echtzeitfähigkeit

Die Verläufe aus Bild 7.10 zeigen die Echtzeitfähigkeit des Algorithmus auf, da die Berechnung der Einfallswinkel und deren Übertragung stets weniger Zeit benötigt als durch die

EDMA-Verarbeitung zur Verfügung steht. Weiter wird deutlich, dass die Dauer des Algorithmus stark variieren kann, da die Anzahl der ausgewerteten Frequenzstützstellen, die Anzahl der zum Mitteln herangezogenen Zwischenergebnisse, die Dauer der Singulärwertzerlegung und die Anzahl der zu übertragenen Pakete zufällig ist. Allerdings kann durch die Parameter N\_PEAKS, N\_Hist und N\_HISTPEAKS und die Festlegung der maximalen Iterationsschritte der Singulärwertzerlegung die benötigte Zeit begrenzt werden. Diese können so gewählt werden, dass die Echtzeitfähigkeit stets gewährleistet ist. Dabei sollte vor allem bedacht werden, dass, falls die Ergebnisanzahl über drei steigt, sich die Übertragungsdauer deutlich verlängern kann. Andererseits hat sich gezeigt, dass eine Übertragung von bis zu sechs Ergebniswerten in Echtzeit erfolgen kann.

## 8 Visualisierung

Dieses Kapitel beschreibt die entwickelte grafische Oberfläche zur Ergebnisdarstellung. Das zugrunde liegende Programm ermöglicht neben der eigentlichen Darstellung der Ergebnisse auch die Erstellung von Protokollen für Testzwecke.

### 8.1 Wahl der Programmiersprache

Grafische Benutzeroberflächen lassen sich in einer Vielzahl an Hochsprachen und Programmen entwickeln. Da jedoch die Unabhängigkeit von einem Programm wünschenswert ist, entfallen jegliche Lösungen, die ein eventuell kostenpflichtiges Programm benötigen.

Die Hochsprachen *C* und *C++* ermöglichen lediglich mit umfangreichen Zusatzbibliotheken eine grafische Ausgabe. *C Sharp* hingegen bietet die gesuchte Möglichkeit. Allerdings sind diese Programmiersprachen nicht so plattformunabhängig wie *Java*, die lediglich auf die entsprechende *JAVA Virtual Machine* (JVM) angewiesen ist. Aufgrund der Portierbarkeit wird die Entwicklung der grafischen Benutzeroberfläche in *Java* vorgezogen.

### 8.2 Unterstützende Bibliotheken

Dieser Abschnitt stellt die verwendeten Bibliotheken vor, die nicht im *Java Software Development Kit* (JDK) enthalten sind und somit zusätzlich zu einer entsprechenden *Java*-Installation hinzugefügt werden müssen.

#### 8.2.1 RXTX

Da *Java* vom Grundgedanken her plattformunabhängig sein soll, ist im JDK keine entsprechende Bibliothek zum Ansprechen einer seriellen Schnittstelle enthalten. Allerdings existieren plattformabhängige Zusatzbibliotheken, die diese Aufgabe übernehmen. Von *Sun Microsystems*<sup>6</sup> wird unter Linux-Systemen die *Java Communications 3.0 API* bereitgestellt.

---

<sup>6</sup>*Sun Microsystems* ist seit 2010 ein Tochterunternehmen von *Oracle*.

Für Windows- oder Mac-Systeme kann auf die veraltete *Java Communications 2.0 API* zurückgegriffen werden. Neben diesen Bibliotheken existiert die *Open Source Bibliothek RXTX*, die unter der Lizenz *GNU Lesser General Public License 2.1* (LGPL 2.1) angeboten wird. Der Vorteil dieser Bibliothek ist, dass sie für alle gängigen Betriebssysteme angeboten wird, wodurch sie auch in diesem Projekt verwendet wird.

Die *RXTX*-Bibliothek stellt Funktionen zur Verfügung, die zur Kommunikation mit der seriellen Schnittstelle nötig sind. Sie basiert auf der Spezifikation der *Java Communications API* von *Sun Microsystems*.

Der entsprechende Internetauftritt ist unter [28] zu finden.

## 8.2.2 Java3D

Für die Darstellung von dreidimensionalen Objekten ist ebenfalls eine Zusatzbibliothek notwendig. Neben einer Vielzahl an sehr umfangreichen und komplexen Bibliotheken, bietet die *Java3D*-Bibliothek eine kompakte Programmierschnittstelle. Diese ermöglicht die Modellierung einfacher, dreidimensionaler Szenen. Dieses umfasst die Erzeugung, Manipulation und Darstellung.

In dieser Arbeit wird die Bibliothek in der Version 1.5.1 verwendet. Lizenzrechtlich sind die Unterbibliotheken unter der *GNU General Public License* (GPL) Lizenz und *Berkeley Software Distribution* (BSD) Lizenz angeordnet.

Der entsprechende Internetauftritt ist unter [19] zu finden.

## 8.3 Programmübersicht

Eine Ansicht des Programmes ist in Bild 8.1 dargestellt. Diese lässt sich prinzipiell in drei Bereiche einteilen:

- Im oberen Bereich werden die aktuellen Einfallrichtungen im dreidimensionalen Raum dargestellt. Zur Orientierung wird neben dem Einfallswinkel auch das Mikrofonarray eingezeichnet. Bei der Darstellung der Einfallrichtung werden alle übertragenen Informationen dargestellt. In Rot wird die aktuelle Einfallrichtung und in Hellbau werden die Toleranzbereiche der Winkel eingezeichnet. Dabei handelt es sich beim Azimutwinkel um eine Uminterpretierung des Histogrammwertes. Beim Elevationswinkel handelt es sich um das Minimum und Maximum.
- Der mittlere Bereich zeigt eine zeitliche Historie der Einfallswinkel. Dazu werden die Einfallswinkel in einem Diagramm in Abhängigkeit der Zeit dargestellt.

- Der untere Bereich dient zur Kontrolle des Programmes. Unter anderem kann die RS232-Kommunikation auf- und abgebaut, die Darstellung der Einfallswinkel oder die Bewegung des Arrays aktiviert bzw. deaktiviert und das Abspielen der Testsequenzen kann gestartet oder gestoppt werden. Neben diesen Kontrollmöglichkeiten können entsprechende Protokolle aufgezeichnet werden. Diese ermöglichen die anschließende Auswertung.

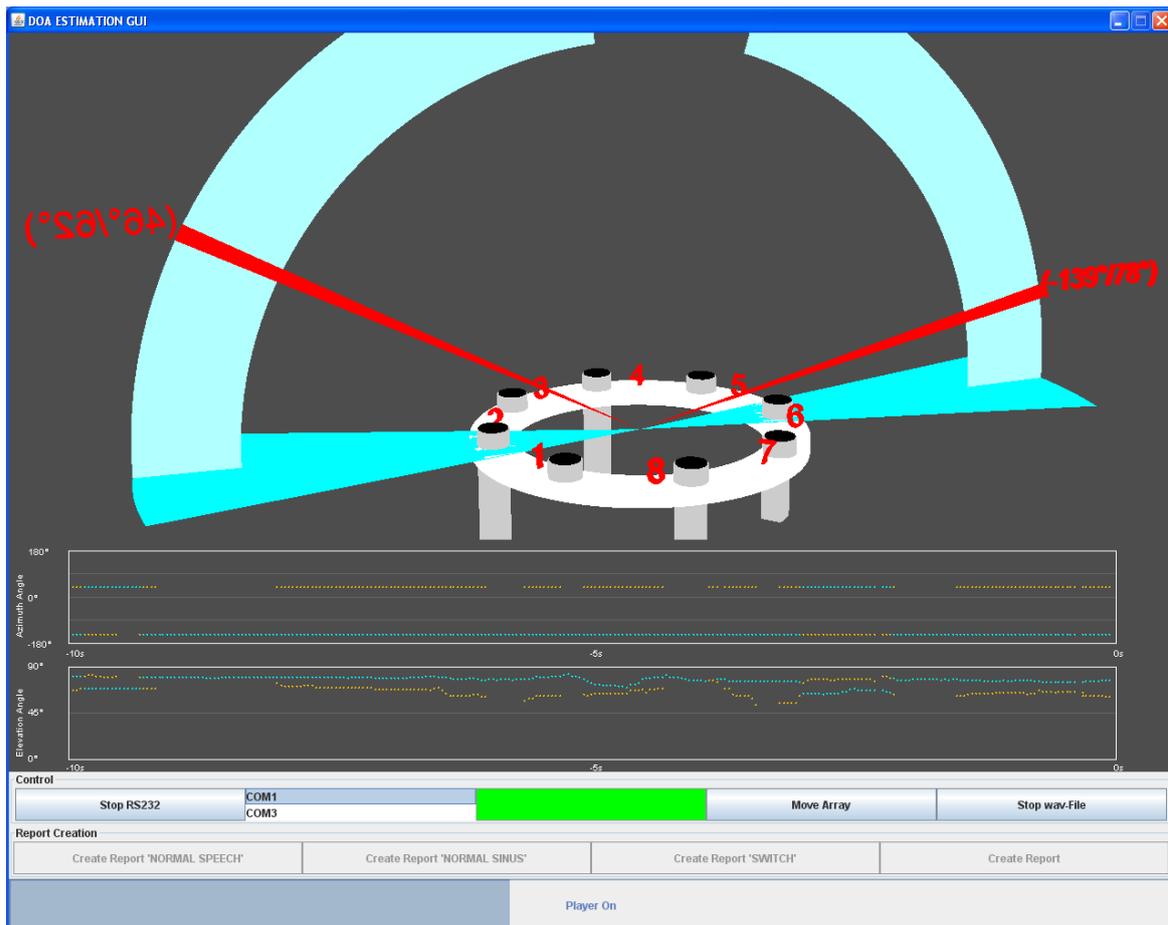


Bild 8.1: Programmansicht

Für die Erstellung der Protokolle existieren vier verschiedene Möglichkeiten. Alle haben gemein, dass stets alle empfangenen Ergebnisse protokolliert werden. Sie unterscheiden sich lediglich in der Art der abgespielte Testsequenzen:

- **Create Report 'NORMAL SPEECH'**: In dieser Variante werden beide Sprachtestsequenzen einmalig abgespielt.
- **Create Report 'NORMAL SINUS'**: Wird diese Art der Protokollierung verwendet, werden zwei Sinustöne unterschiedlicher Frequenzen (1,5 kHz und 2 kHz) abgespielt.

- **Create Report 'SWITCH'**: Bei Verwendung dieser Variante wird ein Sinuston der Frequenz 1,5 kHz in zeitlichen Abständen abwechselnd auf je einen Audiokanal ausgegeben.
- **Create Report**: In dieser Möglichkeit der Protokollierung wird keinerlei Testsequenz abgespielt.

## 8.4 Programmimplementierung

In den nachfolgenden Abschnitten wird die Implementierung des Programmes vorgestellt. Dazu wird ein UML-Diagramm aufgezeigt und die entwickelten Klassen kurz erläutert.

Das *Unified Modeling Language*-Diagramm (UML-Diagramm) des Programmes ist in Bild 8.2 dargestellt. Die verwendeten Klassen werden nachfolgend näher erläutert.

**MainGUI** *MainGUI* ist die Hauptklasse der Anwendung. Von ihr wird in der *main*-Methode ein Objekt erstellt und im Konstruktor werden durch die Objekterstellung alle weiteren relevanten Objekte erzeugt. Sie stellt das Bindeglied zwischen den anderen Objekten dar und übernimmt die grundsätzliche grafische Ausgabe.

**DoA** Die Klasse *DoA* stellt ein ermitteltes Ergebnis des Echtzeitsystemes dar. In ihr werden alle übermittelten Angaben bezüglich des Ergebnisses inkl. Zeitstempel gebündelt. Nach Empfang dieser Daten werden die Daten innerhalb der Anwendung als Objekt dieser Klassen verarbeitet und weitergegeben.

**DoAVector** *DoAVector* sammelt alle Ergebnisse (*DoA*-Objekte) eines übermittelten Datensatzes. Die Klasse ist von der Klasse *Vector<DoA>* abgeleitet.

**DoAHistory** Die Klasse *DoAHistory* verwaltet alle vergangenen Ergebnisse der letzten X Sekunden. Sie ist von der Klasse *Vector<DoAVector>* abgeleitet.

**DataLogger** Ein Objekt der Klasse *DataLogger* ermöglicht das threadsichere Speichern von Strings in Dateien und wird somit für die Erstellung von entsprechenden Protokollen verwendet.

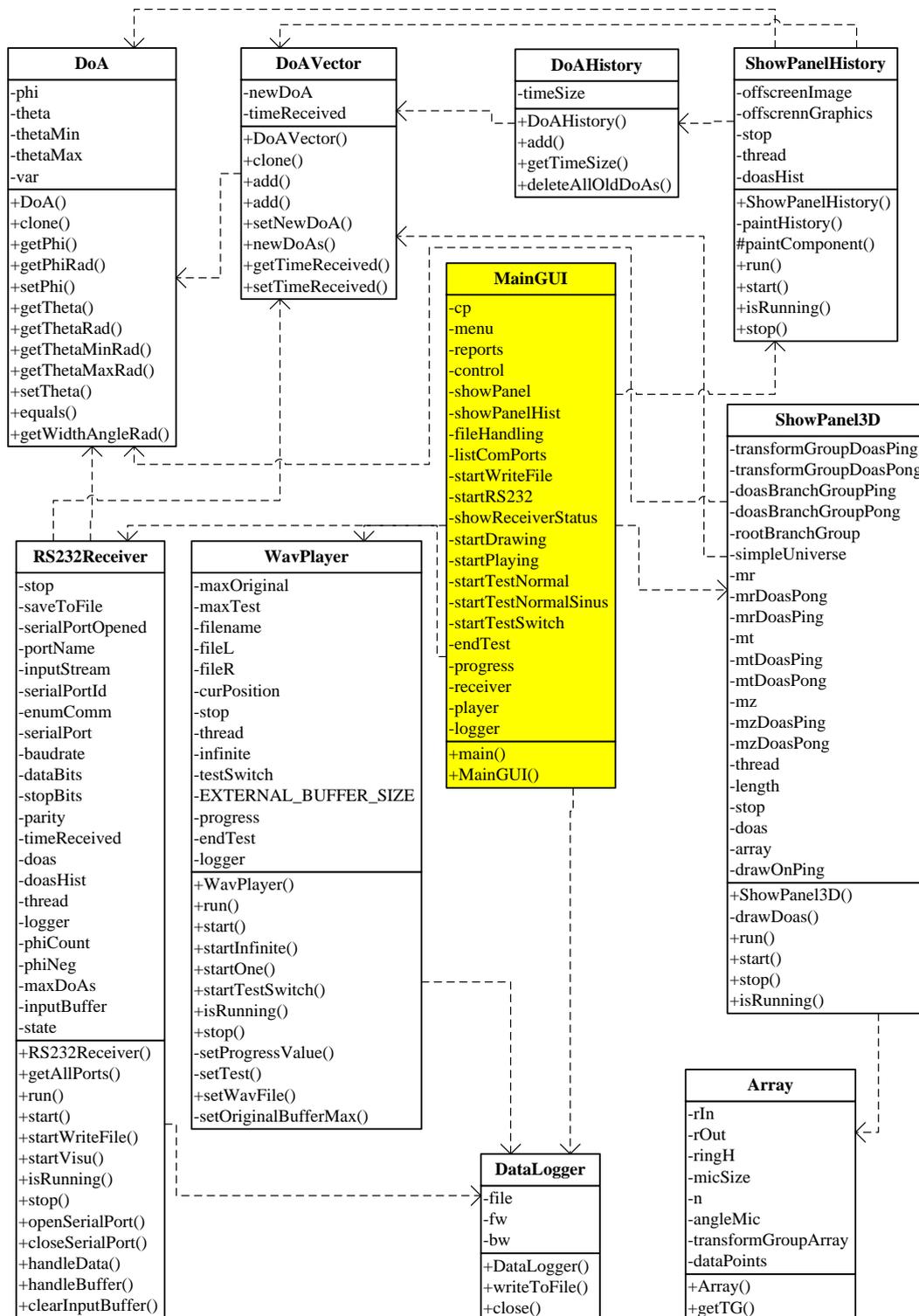
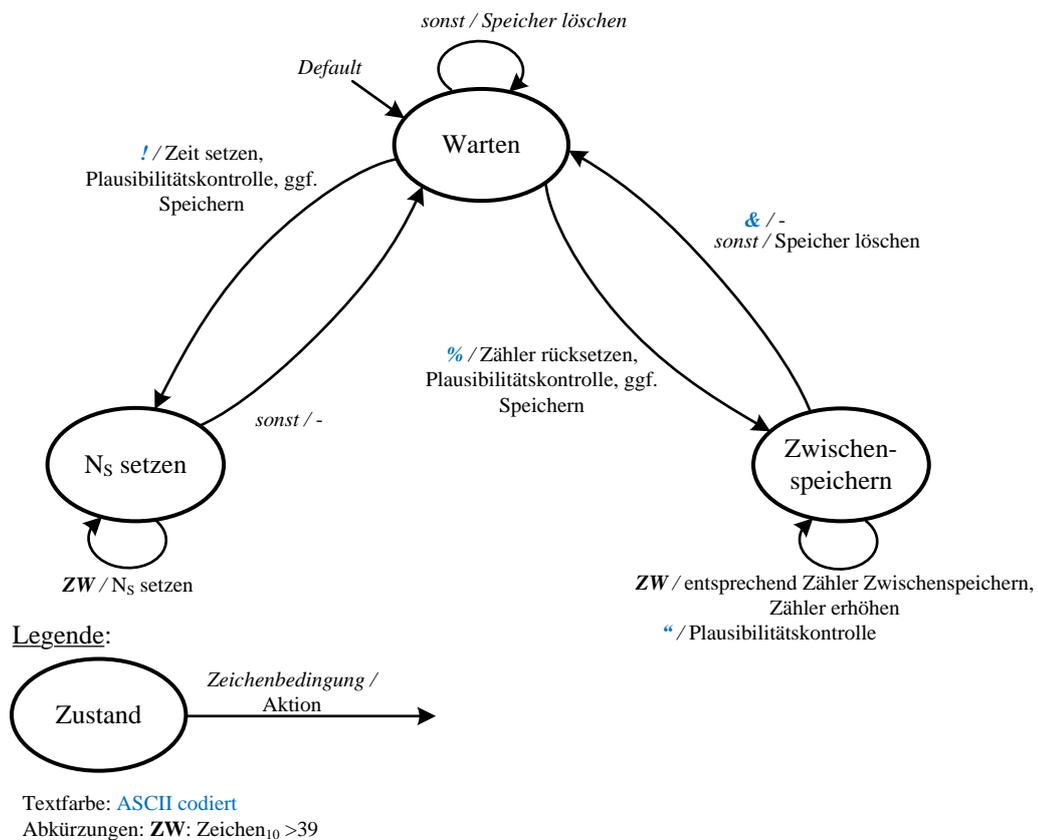


Bild 8.2: UML-Diagramm

**RS232Receiver** Die Klasse *RS232Receiver* ist für den Empfang des seriellen Datenstromes und für dessen Verarbeitung zuständig. Dazu wird die beschriebene RXTX-Bibliothek verwendet. Die anschließende Verarbeitung wird entsprechend des Automaten aus Bild 8.3 durchgeführt.

Dabei hat die Plausibilitätskontrolle bei negativem Ergebnis einen Sprung in den Zustand *WAIT* und die Durchführung der Aktion *Löschen des Speichers* zur Folge. Weiter handelt es sich im Wesentlichen um die Umsetzung des definierten Protokolls. Die empfangenen Zeichen werden dabei entsprechend dem Automat verarbeitet.



**Bild 8.3:** Automat des RS232-Empfängers

Die empfangenen Ergebnisse können als *DoA*-Objekt in einem *DoAVector*-Objekt abgelegt oder über *DataLogger*-Objekt in einer Datei abgespeichert werden.

Objekte dieser Klasse laufen in einem eigenen Thread.

**WavPlayer** Die Klasse *WavPlayer* realisiert jegliche Audioausgabe der Anwendung. Dabei kann zwischen drei verschiedenen Testsignalen gewählt werden:

- Sprachsignale
- Sinussignale bei verschiedenen Frequenzen
- zeitlich abwechselnde aktive Sinussignale zwischen den Kanälen

Für weitere Details bezüglich dieser Testsignale sei auf das Kapitel 9 verwiesen.

Objekte dieser Klasse laufen in einem eigenen Thread. Ereignisse können mit Hilfe eines *DataLogger*-Objekts in einer Datei protokolliert werden.

**ShowPanel3D** Ein Objekt der Klasse *ShowPanel3D* ist für die grafische 3D-Ausgabe zuständig. Neben dem Mikrofonarray wird stets das aktuelle *DoAVector*-Objekt grafisch dargestellt.

Sie ist von der Klasse *Canvas3D* abgeleitet und läuft in einem eigenen Thread.

**Array** Die Klasse *Array* beschreibt das gegebene Mikrofonarray. Die Klasse wird für die 3D-Grafik-Ausgabe verwendet, indem ein Objekt dieser Klasse ein entsprechendes 3D-Objekt zurückgeben kann.

**ShowPanelHistory** Ein Objekt der Klasse *ShowPanelHistory* stellt die zweidimensionale Historie der empfangenen Ergebnisse dar. Als Basis dient ein Objekt der Klasse *DoAVector*.

Die Klasse ist von der Klasse *JPanel* abgeleitet und läuft in einem eigenen Thread.

# 9 Tests, Fehlerkorrektur und Evaluation

Die Simulation auf Basis von generierten Testsequenzen oder Aufnahmen haben zumeist gute Leistungsmerkmale aufgezeigt. Allerdings ist bereits ein systematischer Fehler im ursprünglichen Algorithmus enthalten. Simulationen haben darüber hinaus gezeigt, dass durch weitere getroffene Annahmen und nicht ideale Bedingungen mit zusätzlichen Fehlern zu rechnen ist.

Die nachfolgenden Tests untersuchen die Leistungsmerkmale des entwickelten Echtzeit-Systems. Dazu werden zunächst Bewertungskriterien entwickelt. Anschließend werden in einem reflexionsarmen Raum und in einer realen Umgebung in Hinsicht auf die Bewertungskriterien Untersuchungen durchgeführt.

## 9.1 Bewertungskriterien

Um eine Bewertung des entwickelten Systems ableiten zu können, seien folgende Bewertungskriterien untersucht:

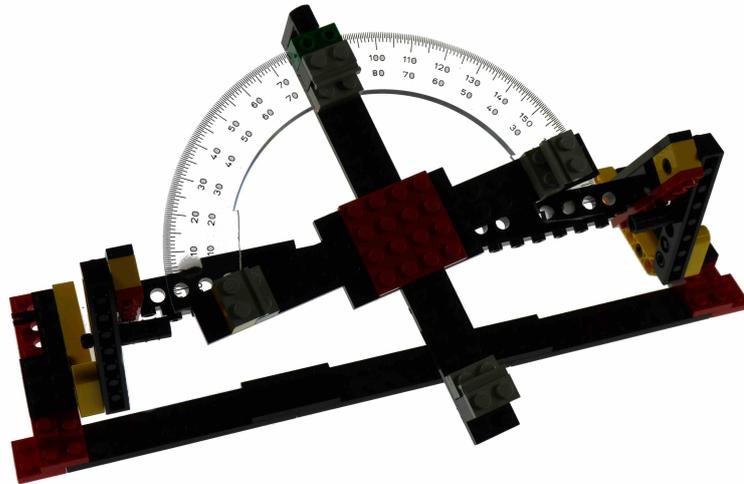
- Genauigkeit und Varianz bei einer aktiven Audioquelle
- Genauigkeit und Trennbarkeit bei zwei aktiven Audioquellen
- Reaktionszeit auf Positionsänderungen aktiver Audioquellen
- Robustheit in realer Umgebung

## 9.2 Testsystem

Um ein strukturiertes Vorgehen in den Testreihen zu ermöglichen, ist ein Testsystem auf Basis von *LEGO Technik* aufgebaut worden. Dieses ist in Bild 9.1 abgebildet.

Das Testsystem ermöglicht die Positionierung des Mikrofonarrays in vier verschiedenen Winkeln in Bezug zur Referenzebene und eine anschließende nahezu kontinuierliche Drehung.

Die verwendete Messeinheit hat eine Auflösung von  $1^\circ$ .



**Bild 9.1:** Testsystem

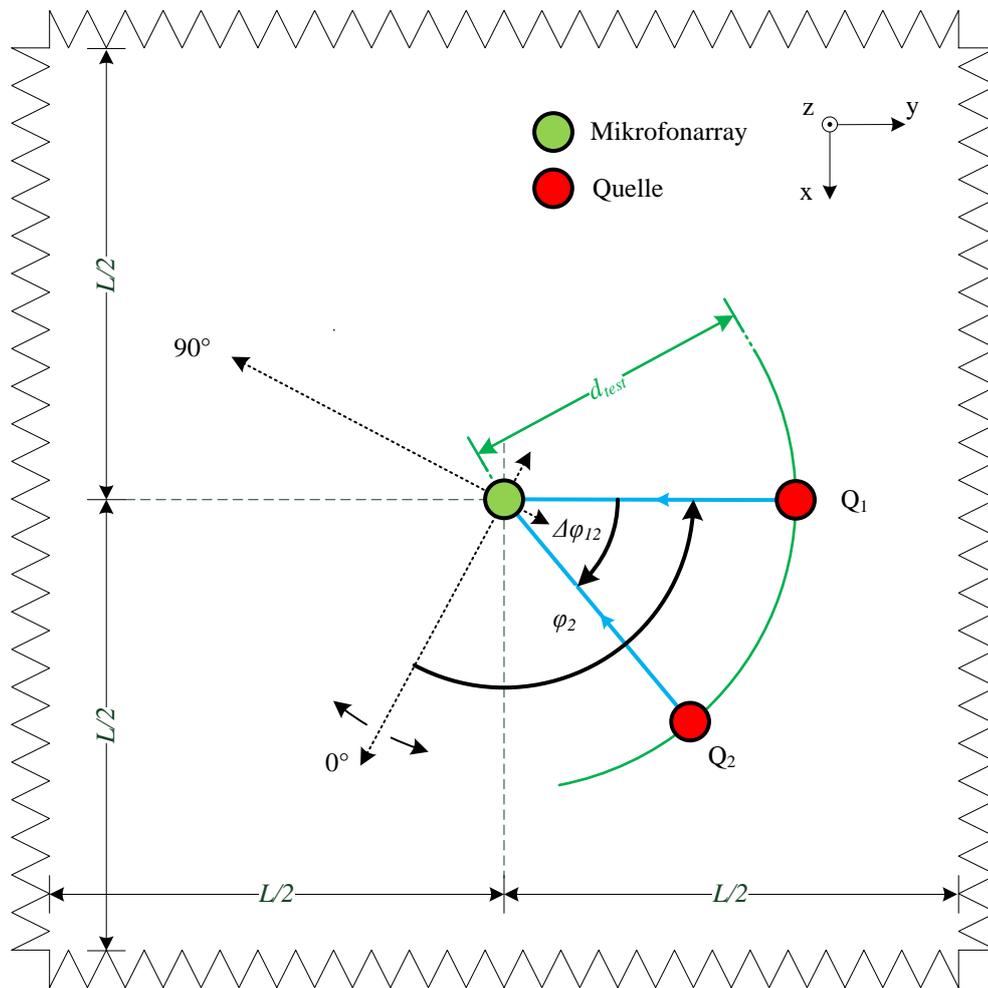
## 9.3 Tests im reflexionsarmen Raum

Um Aussagen über die Genauigkeit eines Systems treffen zu können, müssen die Tests in einem reflexionsarmen Raum stattfinden. Dabei handelt es sich um einen Raum, der alle Schallwellen absorbiert und somit Echos und weitere Störungen unterdrückt. Ebenfalls ist dieser Raum von seiner Umgebung entkoppelt, damit keine externen Störungen Einfluss auf die Messergebnisse nehmen.

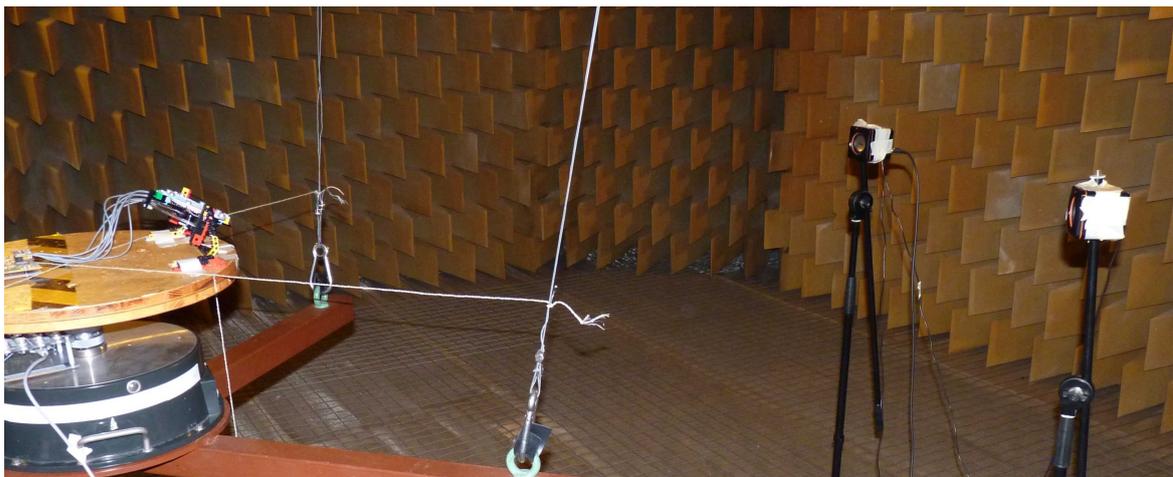
### 9.3.1 Aufbau und Systemparametrisierung

Der Aufbau für die nachfolgenden Tests ist in Bild 9.2 zu sehen. Der reflexionsarme Raum ist dabei kubisch. Die Länge einer Seite beträgt  $L = 7,1$  m. Die Quellen sind in einem Abstand von  $d_{test} = 1,37$  m aufgestellt, wodurch die Fernfeldbedingung (s. Gleichung 3.1) erfüllt sein sollte.

Das Mikrofonarray ist im Testsystem eingespannt und die Lautsprecher weisen im Referenzkoordinatensystem keine Differenz in der z-Komponente auf. Aus dieser Tatsache folgt, dass die Einfallswinkel der Quelle 1, die ebenfalls keine Differenz zum Mikrofonarray in der x-Komponente des Referenzkoordinatensystems aufweist, sich direkt aus den eingestellten Winkeln ableiten lassen. Bei der zweiten Quelle muss hingegen stets eine Koordinatensystemtransformation des Positionvektors durchgeführt werden.



(a) Aufbau und Abmessungen



(b) Aufnahme des Versuchsaufbaus

**Bild 9.2:** Versuchsaufbau im reflexionsarmen Raum

Die Systemparametrisierung bei den folgenden Tests wird in Tabelle 9.1 aufgeführt.

Parameter	Parameterwert	Parameter	Parameterwert
N_BLOCK	300	N_M (M)	7 (3)
N_OVERLAP	212	EV_ABS_MAX	1,2
N_ALGO	512	SVD_VALUE_DIVISOR	10
X_ENERGY_MIN	0,5	N_HIST	50
F_PEAK_MIN_POW2	16	N_HIST_WAIT	1
N_PEAKS	10	HIST_PEAK_MIN	5
		N_HISTPEAKS	8
		HIST_ACCURACY	10

**Tabelle 9.1:** Einstellung der Systemparameter (reflexionsarmen Raum)

### 9.3.2 Testreihe und Fehlerkorrektur mit einer aktiven Quelle

Mit Hilfe des vorgestellten Testaufbaus sind Messwerte bei verschiedenen Winkelkonfigurationen durchgeführt worden. Dabei ist stets die männliche Testsequenz als Signal abgespielt worden. In der Bildergruppe 9.3 sind die Ergebnisse (mittlere Abweichung und Varianz) dargestellt.

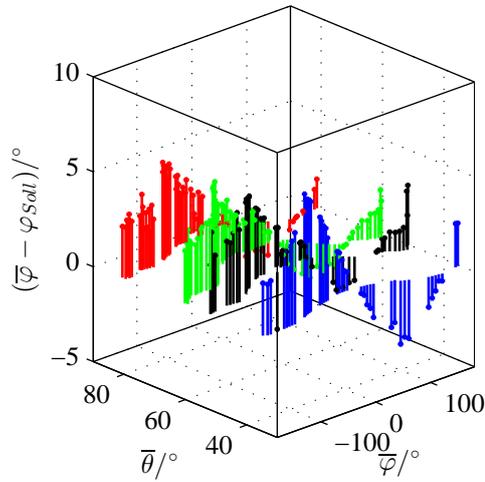
Die Ergebnisse weisen eine gewisse Systematik auf. Die Differenz zwischen mittlerem und wahren Wert beim Azimut- wie auch beim Elevationswinkel verläuft in Abhängigkeit des Azimutwinkels einem Sinus ähnlich. In Abhängigkeit des Elevationswinkels ist lediglich beim Fehler des Elevationswinkels eine Veränderung festzustellen.

Aufgrund der gegebenen Systematik lässt sich der Fehler der Mikrofonarraystruktur und dem Algorithmus zuordnen. Die sinusartigen Verläufe weisen dabei auf eine Ellipsen- anstatt eine Kreisform des Mikrofonarrays hin. Die Veränderung des Fehlers des Elevationswinkels mit dessen Wert hat sich bereits in den Simulation aus Abschnitt 7.2.1 als systematische Steigerung des Fehlers abgezeichnet.

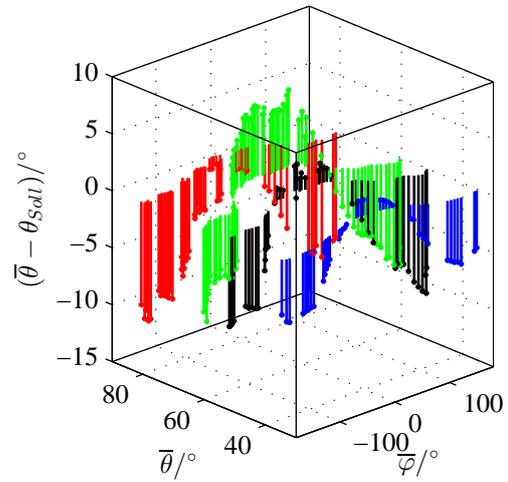
Durch die Systematik im Fehler kann dieser korrigiert werden. Die Fehler lassen sich näherungsweise mit den Gleichungen

$$\varphi_{err} = 3^\circ \cdot \sin(\varphi - 175^\circ) + 1,5^\circ \tag{9.1}$$

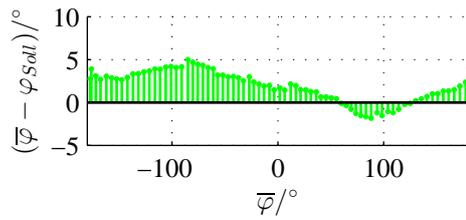
$$\theta_{err} = \begin{cases} \left(1 - e^{-0.035 \frac{\theta}{90}}\right) \cdot (6^\circ \cdot \sin(\varphi + 75^\circ) - 6^\circ + 0,12(\theta - 22^\circ)) & \text{falls } \theta < 75^\circ \\ 6^\circ \cdot \sin(\varphi + 75^\circ) - 0,4 \cdot (\theta - 75^\circ) & \text{sonst} \end{cases} \tag{9.2}$$



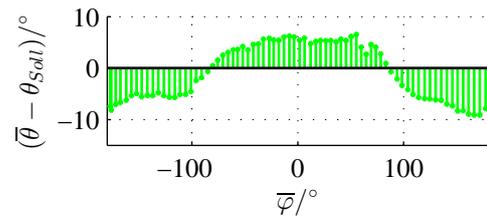
(a) Fehler  $\bar{\varphi} - \varphi_{Soll}$  über  $\bar{\varphi}$  und  $\bar{\theta}$



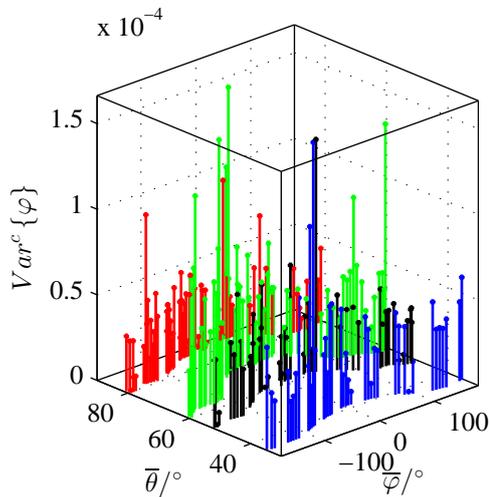
(b) Fehler  $\bar{\theta} - \theta_{Soll}$  über  $\bar{\varphi}$  und  $\bar{\theta}$



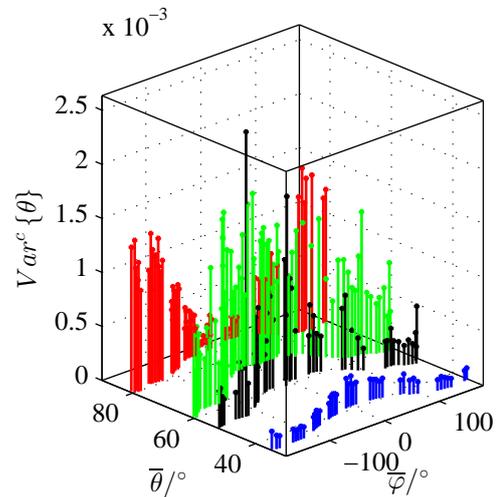
(c) Fehler  $\bar{\varphi} - \varphi_{Soll}$  über  $\bar{\varphi}$  bei  $\theta_{Soll} = 68^\circ$



(d) Fehler  $\bar{\theta} - \theta_{Soll}$  über  $\bar{\varphi}$  bei  $\theta_{Soll} = 68^\circ$



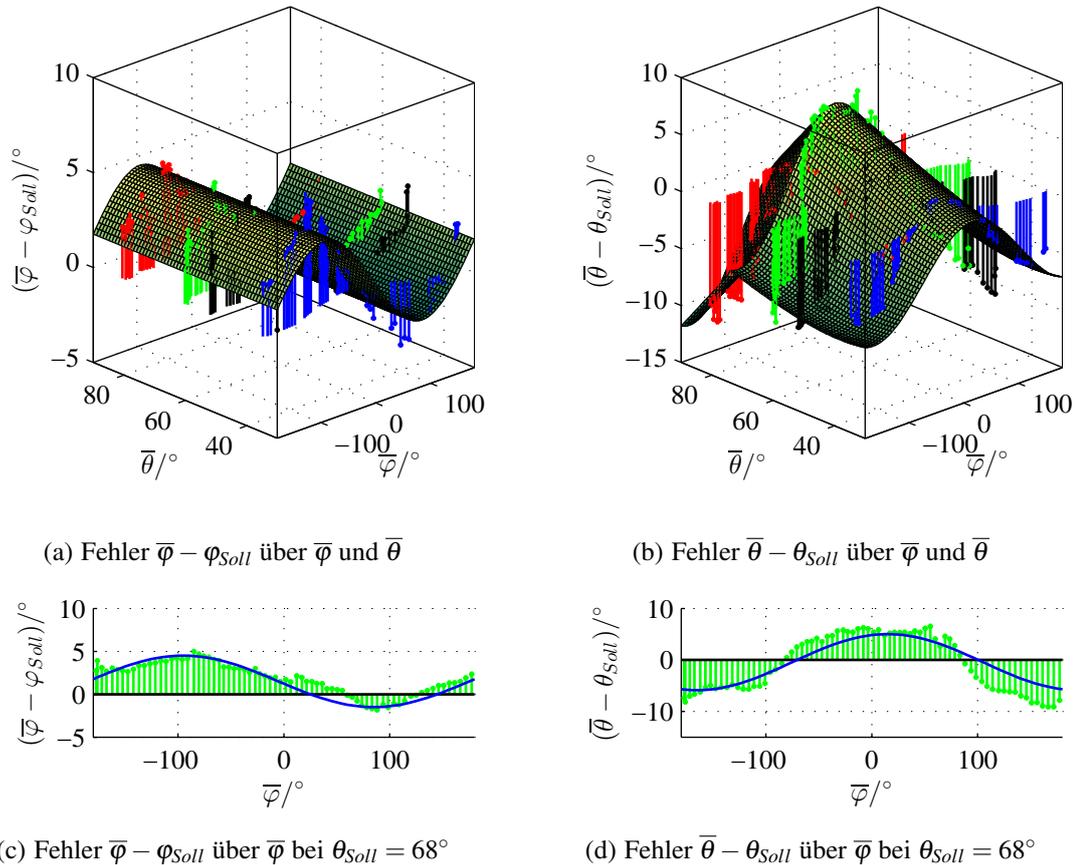
(e) Varianz von  $\varphi_i$  über  $\bar{\varphi}$  und  $\bar{\theta}$



(f) Varianz von  $\theta_i$  über  $\bar{\varphi}$  und  $\bar{\theta}$

**Bild 9.3:** Fehler- und Varianzverläufe ohne Korrektur

berechnen. Dabei stehen  $\varphi$  und  $\theta$  für die aktuellen Ergebniswinkel. In Bildergruppe 9.4 sind abermals die Fehler und zusätzlich die modellierten Fehlerkurven auf Basis der Gleichungen 9.1 und 9.2 dargestellt. Bei den modellierten Fehlerkurven handelt es sich um die kontinuierlichen Funktionsverläufe.

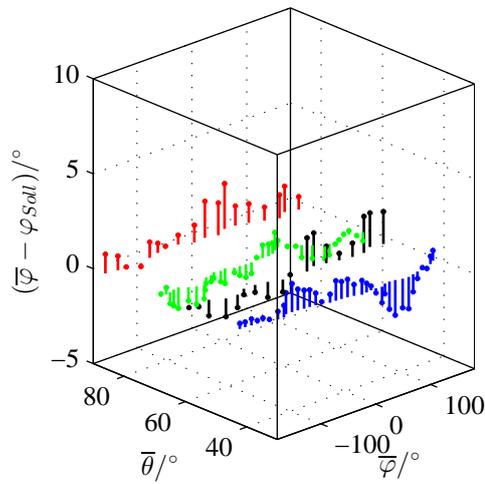


**Bild 9.4:** Vergleich von Fehlerverläufen und modellierten Fehlerverläufen

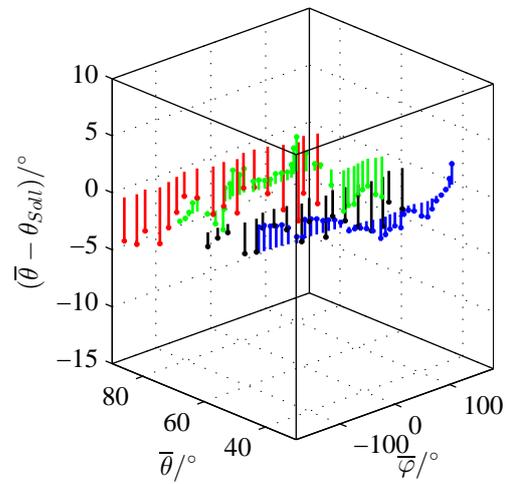
Die Bilder aus der Bildergruppe 9.4 zeigen deutlich, dass die Gleichungen 9.1 und 9.2 eine gute Näherung darstellen.

Durch eine Subtraktion des berechneten Fehlers vom aktuellen Ergebniswinkel ergibt sich der korrigierte Ergebniswinkel. Wie bereits im Abschnitt 6.5.5 erwähnt, gibt es prinzipiell zwei Möglichkeiten diese Korrektur durchzuführen. Da jedoch bei einer Korrektur vor Histogrammauswertung mit einer besseren Histogrammauswertung zu rechnen ist, sei diese Möglichkeit nachfolgend betrachtet. Die sich mit der Korrektur ergebenden Fehler und Varianzen sind in der Bildergruppe 9.5 abgebildet.

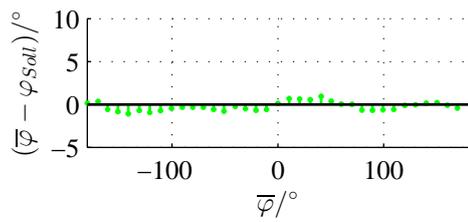
Es ist eine deutliche Verbesserung der Genauigkeit erkennbar. Vor allem die zuvor vorhandene Systematik ist nur noch ansatzweise vorhanden. Die Fehler des Azimutwinkels liegen stets



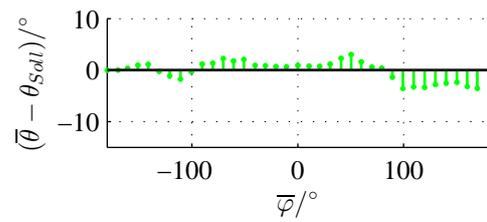
(a) Fehler  $\bar{\varphi} - \varphi_{Soll}$  über  $\bar{\varphi}$  und  $\bar{\theta}$



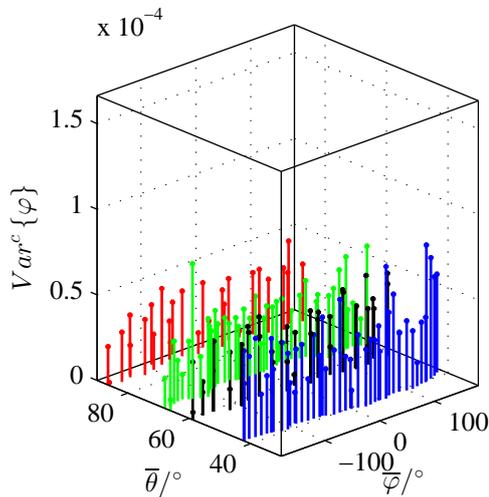
(b) Fehler  $\bar{\theta} - \theta_{Soll}$  über  $\bar{\varphi}$  und  $\bar{\theta}$



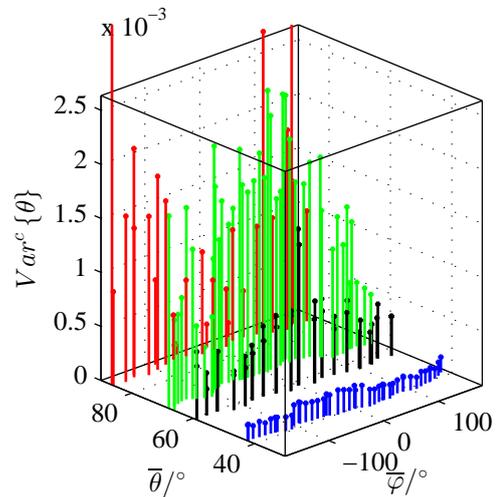
(c) Fehler  $\bar{\varphi} - \varphi_{Soll}$  über  $\bar{\varphi}$  bei  $\theta_{Soll} = 68^\circ$



(d) Fehler  $\bar{\theta} - \theta_{Soll}$  über  $\bar{\varphi}$  bei  $\theta_{Soll} = 68^\circ$



(e) Varianz von  $\varphi$  über  $\bar{\varphi}$  und  $\bar{\theta}$



(f) Varianz von  $\theta$  über  $\bar{\varphi}$  und  $\bar{\theta}$

**Bild 9.5:** Fehler- und Varianzverläufe mit Korrektur

unter  $2^\circ$ , wobei es sich dabei um einen guten Wert handelt, da sich der Messfehler in einer ähnlichen Größenordnung bewegen muss. Weiter sei zu beachten, dass die Auflösung des Systems auf  $1^\circ$  beschränkt ist.

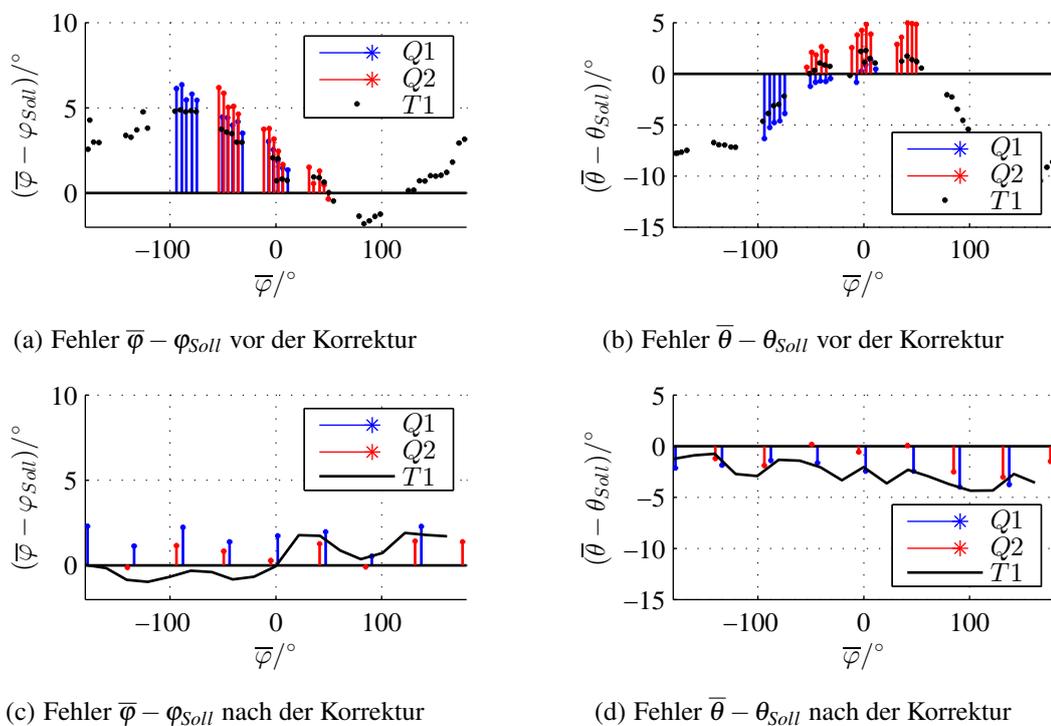
Die Fehler des Elevationswinkels sind ebenfalls gesunken. Der Fehler beläuft sich auf maximal  $5^\circ$ . Dabei muss jedoch beachtet werden, dass dieser vor allem bei einem Sollwinkel von  $90^\circ$  auftritt. Mit diesem Effekt ist ebenfalls, aufgrund des Systems, zu rechnen gewesen, da der Elevationswinkel auf  $90^\circ$  begrenzt ist (s. Diskussion aus Abschnitt 5.4).

Die Varianz ist nahezu konstant geblieben, zeigt jedoch leicht erhöhte Werte im Elevationswinkel auf. Dies kann durch die zusätzliche Verschiebung der Fehlerkorrektur innerhalb des kleineren oberen Kennlinienbereichs erklärt werden. Die deutliche Steigerung der Varianz mit der Erhöhung des Elevationswinkels ist weiterhin zu erkennen. Dies ist wiederum mittels der Struktur der Eigenwerte zu begründen (s. Kapitel 5).

### 9.3.3 Testreihen mit zwei aktiven Quellen

Die vorherigen Testreihen haben gezeigt, dass das System eine relativ hohe Genauigkeit aufweist. Allerdings ist dabei stets lediglich eine Quelle aktiv gewesen. Dieser Abschnitt dient der Untersuchung, welchen Einfluss eine zweite aktive Quelle auf die Ergebnisse hat. Neben der Genauigkeit wird auch untersucht, unter welchen Bedingungen zwei Quellen separierbar sind und wie lange eine entsprechende Reaktion auf eine örtliche Variation andauert.

Zu diesem Zweck sind die Fehlerverläufe in der Bildergruppe 9.6 vor und nach der Korrektur mit zwei Quellen bei einem Elevationswinkel von  $60^\circ$  dargestellt. Dabei steht  $Q1$  für die Ergebnisse der ersten,  $Q2$  für die Ergebnisse der zweiten Quellen und  $T1$  für die Ergebnisse aus den Tests mit einer Quelle.

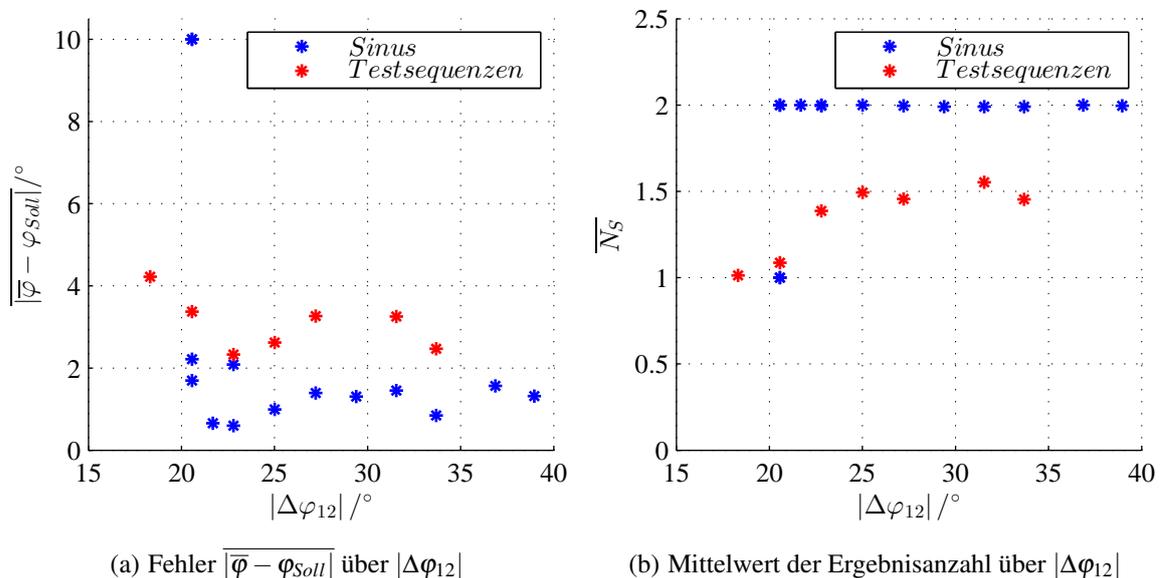


**Bild 9.6:** Fehlerverläufe bei zwei aktiven Quellen bei  $\theta_{Soll} = 67^\circ$  über  $\bar{\varphi}$

Bevor die Ergebnisse zueinander in Bezug gesetzt werden, sei zunächst erwähnt, dass die Veränderung des Elevationswinkels der zweiten Quelle, die sich aufgrund des gegebenen Versuchsaufbaus und der Koordinatensystemtransformation ergibt, in diesen Verläufen nicht dargestellt werden können. Da die Anordnung lediglich vier verschiedene Elevationswinkel ermöglicht, sind keine entsprechenden Referenzwerte aufnehmbar. Weiter ist der Messfehler bei der zweiten Quelle größer, da die Sollwerte aufgrund des Versuchsaufbaus ungenauer sind.

Dennoch zeigen die Ergebnisse, dass die Fehlerverläufe bei zwei aktiven Quellen den Fehlerverläufen bei einer aktiven Quelle ähneln. Es ist lediglich eine sehr geringe Erhöhung des Fehlers festzustellen. Diese Erhöhung ergibt sich dadurch, dass ebenfalls Frequenzstützstellen ausgewertet werden, bei denen beide Quellen zur Bildung dieser beitragen. Auch die Korrektur bewirkt eine deutliche Verbesserung der Ergebnisse.

Um die Separierbarkeit zweier Quellen zu untersuchen, wird die zweite Quelle der ersten Quelle angenähert bzw. der Winkel  $\Delta\varphi_{12}$  aus Bild 9.2(a) verringert. In der Bildergruppe 9.7 sind der absolute Fehler zwischen zirkularem Mittelwert und Sollwert sowie der Mittelwert der Ergebnisanzahl bzw. der Schätzwert der aktiven Quellen gezeigt. Die Messergebnisse sind mit einer aktiven Fehlerkorrektur aufgenommen worden. Als Testsignale werden Sinussignale unterschiedlicher Frequenzen und die Sprach-Testsequenzen verwendet.


 (a) Fehler  $|\overline{\varphi} - \varphi_{Soll}|$  über  $|\Delta\varphi_{12}|$ 

 (b) Mittelwert der Ergebnisanzahl über  $|\Delta\varphi_{12}|$ 
**Bild 9.7:** Separierbarkeit zweier Quellen bei  $\varphi_{Soll1} = 0^\circ$  und  $\theta_{Soll1} = 60^\circ$ 

Bei der Begutachtung der Ergebnisse muss zwischen den beiden Testsignalen unterschieden werden. Bei Verwendung der Sinussignale als Testsignale stimmen die Ergebnisse nahezu mit den theoretisch zu erwartenden Werten überein. Da die Histogrammauflösung auf  $10^\circ$  eingestellt ist, können Quellen unterschieden werden, die im Azimutwinkel mindestens um  $20^\circ$  (doppelte Histogrammauflösung) voneinander abweichen. Dieser Wert stimmt nahezu mit den Messergebnissen überein. Ist die Winkeldifferenz größer als  $20^\circ$  werden zwei Ergebnisse mit hoher Genauigkeit detektiert. Bei einer ungefähren Winkeldifferenz von  $20^\circ$  steigt der Fehler schlagartig an und es wird nur eine Quelle detektiert. Dies bedeutet, dass beide Quellen in benachbarte Histogrammabschnitte fallen und somit der Mittelwert gebildet wird, wie auch der Fehler von etwa  $10^\circ$  zeigt.

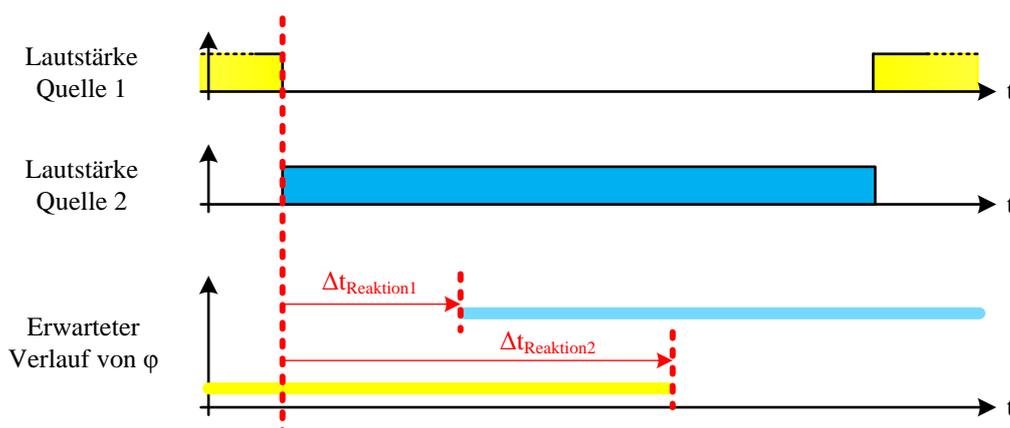
Bei der Verwendung von Sprachtestsignalen verhalten sich die Zusammenhänge ähnlich,

falls die Eigenheiten der Testsequenzen berücksichtigt werden. Bei der Verwendung der Sinussignale sind beide Quellen über den gesamten betrachteten Zeitraum aktiv. Dies ist bei den Sprachsignalen nicht der Fall. Dadurch ergibt sich für die mittlere Anzahl an detektierten Quellen ein Wert von 1,5. Allerdings ist auch in der näheren Umgebung der Winkeldifferenz von  $20^\circ$  ein Sprung auf ein Ergebnis zu erkennen. Dies wird auch im Fehlersignal deutlich, jedoch nicht so stark wie bei den Sinustestsignalen. Dadurch, dass nicht stets beide Quellen aktiv sind, ergeben sich über gewisse Zeiträume keine gemittelten Ergebnisse und folglich geringere Fehler. Dass der Fehler bis auf ein Ergebnis geringfügig größer als bei den Sinustestsignalen ist, lässt sich wiederum durch die Tatsache erklären, dass bei Sprachsignalen auch mehrere Quellen in einer Frequenzstützstelle enthalten sein können.

### 9.3.4 Untersuchung der Reaktionszeit

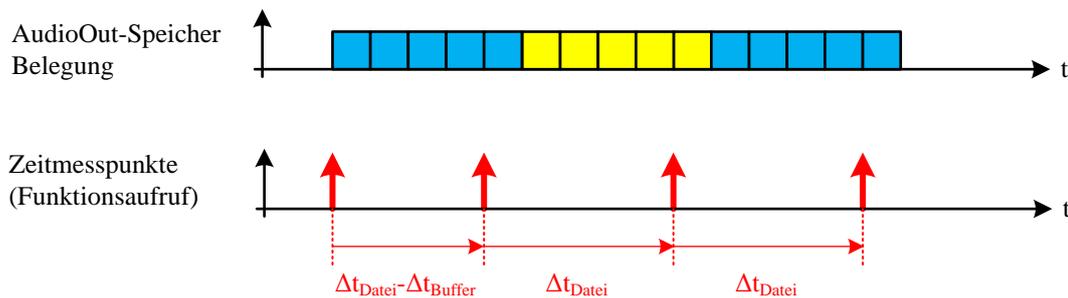
Die bisherigen Tests haben im Wesentlichen die Genauigkeit und die Stabilität des Systems manifestiert. Das Verhalten hinsichtlich des Zeitverhaltens wird in diesem Abschnitt untersucht.

Dazu seien zunächst die Kenngrößen definiert. Es handelt sich um Reaktionszeiten auf die Positionsveränderung von Quellen. Diese wird dadurch ermittelt, dass zwei Quellen unterschiedlich positioniert werden. Allerdings sind beide Quellen nicht gleichzeitig, sondern aufeinander folgend aktiv. Die Reaktionszeiten auf diese Änderung werden gemessen und als Kenngrößen definiert. Die Zeit zwischen Umschalten der Quellen und Detektion der zweiten Quelle wird als Kenngröße  $\Delta t_{\text{Reaktion1}}$  definiert. Als Kenngröße  $\Delta t_{\text{Reaktion2}}$  wird hingegen die Zeit bezeichnet, die zwischen dem Umschalten der Quellen und der Nicht-Detektion der ersten Quelle vergeht. In Bild 9.8 ist dieser Zusammenhang grafisch dargestellt. Um sich von dem zufälligen Charakter der Sprachtestsignale zu lösen, werden abermals Sinustestsignale verwendet.



**Bild 9.8:** Definition der Reaktionszeiten

Um diese Kenngrößen zu messen, muss eine Synchronisation zwischen Empfang der Ergebnisdaten und Umschalten der Quellen gewährleistet sein. Aufgrund dessen muss beides mittels eines einzigen Testsystems erfolgen. Die entwickelte grafische Oberfläche bietet diese Möglichkeit. Bei der Erstellung der Protokolle werden die empfangenen Daten mit einem Zeitstempel versehen. Ebenfalls werden Umschaltunkte sowie Start- und Endpunkt der Tests protokolliert. Allerdings muss beachtet werden, dass die Ausgabe der Testsequenz über einen Audioausgangsbuffer erfolgt. In Bild 9.9 sind sowohl der Status des Buffers als auch die Zeitmesspunkte eingezeichnet.



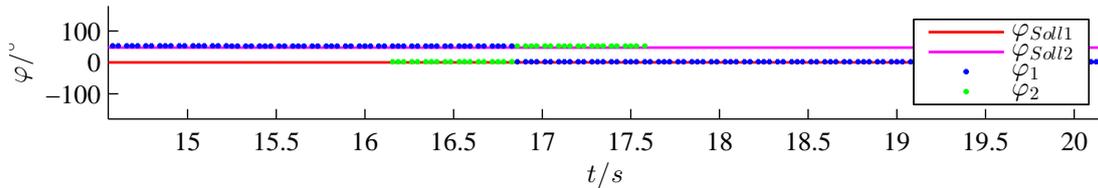
**Bild 9.9:** Synchronisation

Die erste Zeitdifferenz ist verkürzt, da bei der Zeitermittlung nicht gewartet wird, bis der Buffer leer ist. Dieser Effekt tritt jedoch auch beim zweiten Umschalten auf, wodurch die Zeitdifferenz zwischen dem zweiten und dem dritten Zeitmesspunkt mit der korrekten Zeitdifferenz übereinstimmt. Mit diesem Wissen kann die Zeit  $\Delta t_{Buffer}$  ermittelt werden. Alle Zeitstempel, ausgenommen des ersten, können anschließend um diesen Wert korrigiert werden.

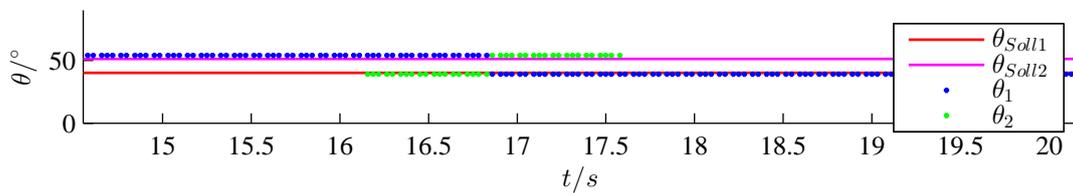
Weiterhin tritt eine zusätzliche Verzögerung zwischen dem Zeitstempel der Audiosignale und der Audioausgabe aufgrund der Bufferstruktur auf. Um diesen Effekt zu korrigieren, müssen die Zeitstempel weiter verzögert werden. Dazu wird die Erstdetektion eines Ergebnisses genutzt. Da das Histogramm vorbelastet sein kann, muss zunächst bestimmt werden, wie groß die Vorbelastung gewesen ist. Dies ist möglich, da die Zeit zwischen Erstdetektion und der ersten Messung von  $\Delta t_{Reaktion1}$  in der Theorie genau die Dateilänge betragen sollte. Mit Hilfe dieses Wissens kann auf die Bufferverzögerung geschlossen werden und alle Zeitstempel bezüglich Audiosignalen können abermals korrigiert werden.

In Bild 9.10 sind die Ergebnisse für  $\varphi_{Soll1} = 0^\circ$ ,  $\theta_{Soll1} = 40^\circ$  und  $\Delta\varphi_{12} = 35^\circ$  dargestellt.

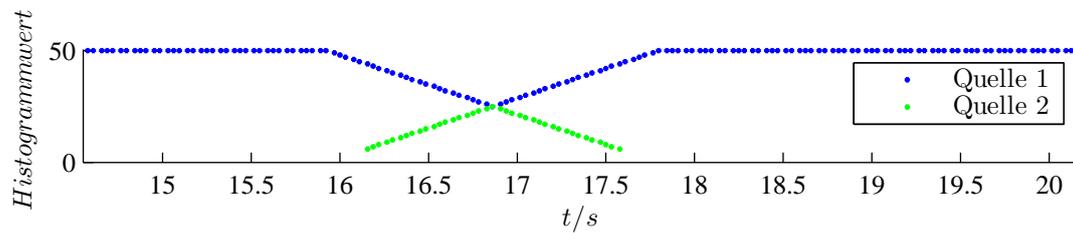
Es ist ersichtlich, dass beide Quellen mit einer guten Genauigkeit detektiert werden und dass ein Positionswechsel stattfindet. Die darauf folgende Reaktion ist deutlich am Histogrammwert abzulesen. Sofort beginnt der Wert nach dem Umschalten der Quellen zu sinken. Nach einer gewissen Zeit erscheint ein zweites Ergebnis, dessen Histogrammwert steigt. Nach einer weiteren Zeitdifferenz tauschen beide Ergebnisse ihre Reihenfolge, bevor das



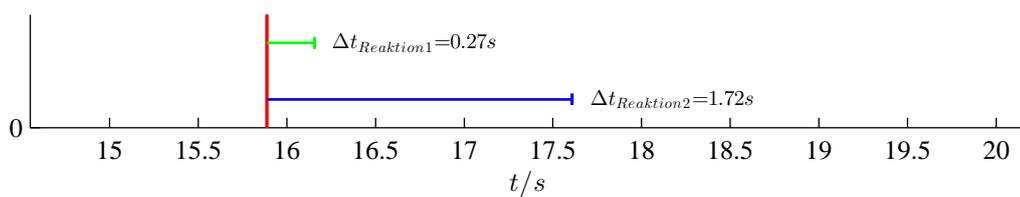
(a) Azimutwinkel über die Zeit



(b) Elevationswinkel über die Zeit



(c) Histogrammwert über die Zeit



(d) Kenngrößen

**Bild 9.10:** Messergebnisse der Reaktionszeiten

erste Ergebnis ganz verschwindet. Die Kenngrößen ergeben sich nach den entsprechenden Korrekturen zu  $\Delta t_{Reaktion1} = 270 \text{ ms}$  und  $\Delta t_{Reaktion2} = 1,72 \text{ s}$ . Vergleicht man diese mit den theoretischen Werten<sup>7</sup> bei angenommener Vollausteuering der Histogramme

$$\Delta t_{Reaktion1} = (HIST\_PEAK\_MIN + 1 + 0 \dots 1) \cdot \frac{N\_BLOCK}{8 \text{ kHz}} \quad (9.3)$$

$$= 225 \text{ ms} - 262,5 \text{ ms} \quad (9.4)$$

$$\Delta t_{Reaktion2} = (HA - HIST\_PEAK\_MIN + 0 \dots 1) \cdot \frac{N\_BLOCK}{8 \text{ kHz}} \quad (9.5)$$

$$= 1,6875 \text{ s} - 1,7250 \text{ s} |_{HA=N\_HIST} \quad (9.6)$$

zeigt sich eine nahezu vollkommene Übereinstimmung mit der Messung. Dabei steht *HA* hier für Histogrammaussteuerung. Die geringen Abweichungen lassen sich durch folgende Gründe erklären: Die teils zufällige RS232-Übertragungsdauer ist in der Theorie nicht berücksichtigt worden, die Zeitauflösung der Zeitstempel ist begrenzt und die Korrektur kann lediglich mit einer Auflösung von  $300/8 \text{ kHz} = 37,5 \text{ ms}$  durchgeführt werden.

Zur Bewertung der gemessenen Reaktionszeiten muss der Unterschied zwischen Sinustestsequenzen und Sprachsignalen beachtet werden. Da bei der Auswertung von Sprachsignalen pro Datensatz mehrere Frequenzstützstellen ausgerechnet werden, erneuert sich der Histogrammbuffer schneller. Daher ist bei der Verwendung von Sprachsignalen mit deutlich kleineren Reaktionszeiten zu rechnen.

## 9.4 Tests in realer Umgebung

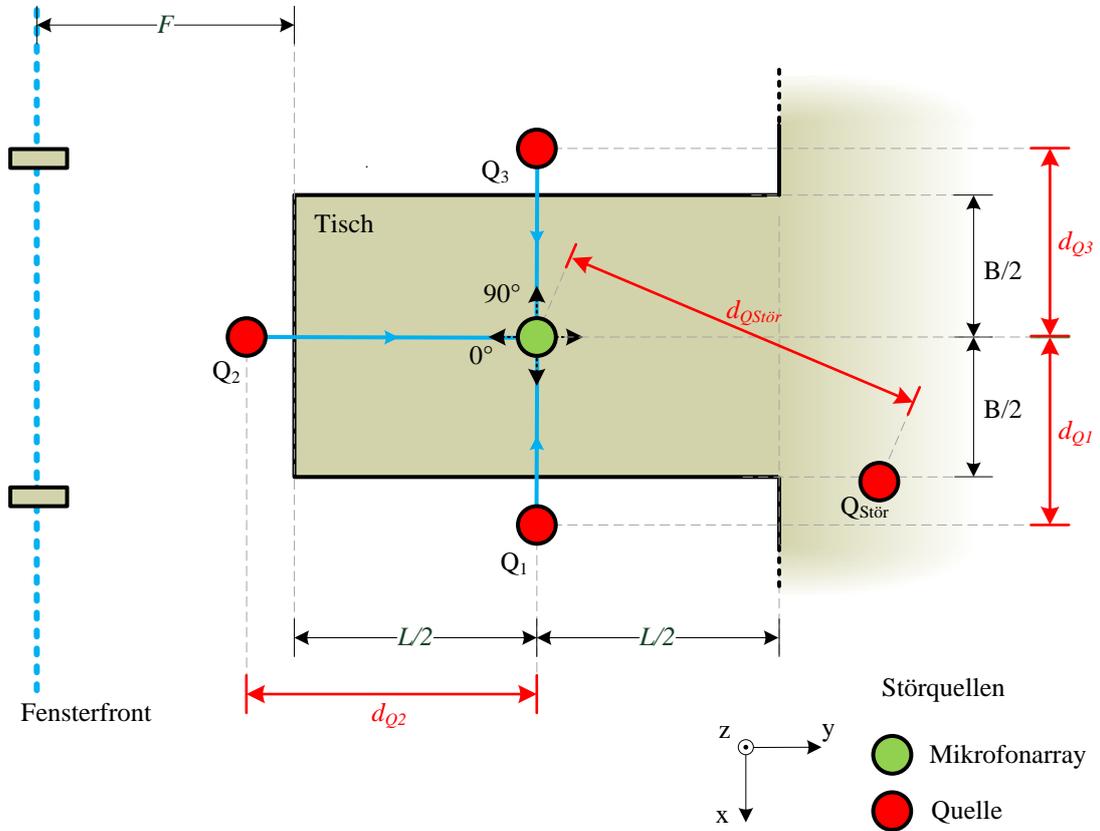
In der realen Umgebung treten deutliche Störeffekte wie beispielsweise Echos durch Mehrwegeausbreitung oder Störgeräusche durch entfernte Sprecher oder Computerlüfter auf. Um Aussagen über das System für reale Umgebung treffen zu können, erscheinen entsprechende Tests in dieser Umgebung sinnvoll.

### 9.4.1 Aufbau und Systemparametrisierung

Der verwendete Aufbau ist in Bild 9.11 abgebildet. Die grundsätzlichen Dimensionen können der Tabelle 9.2 entnommen werden. Bei dem Szenario handelt es sich um einen Tisch, auf dem mittig das Mikrofonarray befestigt ist. An diesem Tisch können rundherum bis zu drei

<sup>7</sup>Bei Berechnung der theoretischen Werte muss einerseits beachtet werden, dass für die Berücksichtigung eines Histogrammspeaks der Minimumwert *HIST\_PEAK\_MIN* überschritten werden muss; andererseits, dass die Signale erst über einen ganzen Block aufgenommen werden und anschließend in einem Bruchteil der Blocklänge das Ergebnis ermittelt wird.

Quellen positioniert werden. Der Aufbau entspricht dem in der Einleitung vorgestellten Szenario an einem Konferenztisch.



**Bild 9.11:** Versuchsaufbau in einer realen Umgebung

Die gewählte Parametrisierung ist lediglich im Parameter HIST\_PEAK\_MIN gegenüber der aus Tabelle 9.1 angepasst worden. Der Wert ist von fünf auf sieben erhöht worden, um den Störungen in der realen Umgebung gerecht zu werden.

Größe	Wert	Größe	Wert
Tischlänge $L$	1,3 m	Fensterabstand $F$	1 m
Tischbreite $B$	0,8 m	min. Abstand zu Störquellen $d_{QStr}$	1,15 m

**Tabelle 9.2:** Umgebungsparameter der realen Umgebung

### 9.4.2 Testreihe mit einer aktiven Quelle

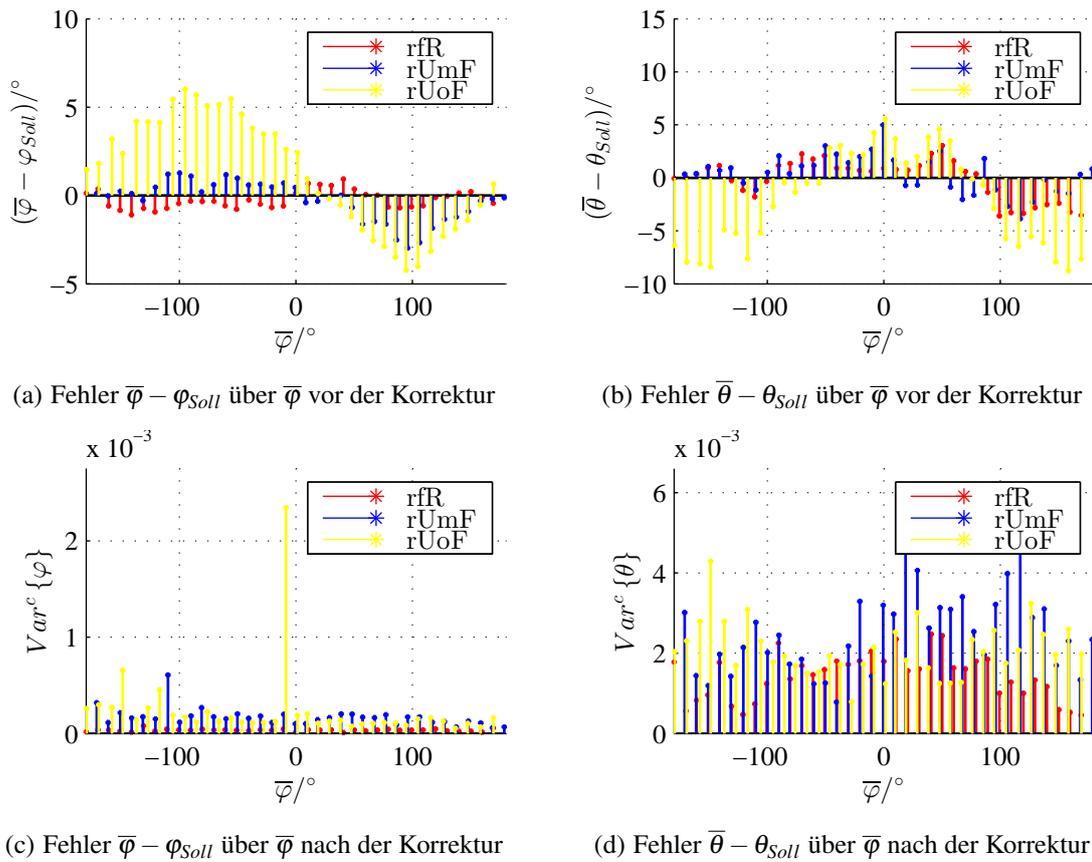
Um Aussagen über die Genauigkeiten zu treffen, ist lediglich Quelle  $Q_2$  entsprechend Tabelle 9.3 positioniert worden. Anschließend ist das Mikrofonarray gedreht worden, um den Azimutwinkel zu variieren.

Größe	$Q_1$	$Q_2$	$Q_3$
$d_{Q_x}$	-	0,8 m	-
$h_{Q_x}$	-	$\approx 0,56$ m	-
$\varphi_{Soll,Q_x}$	-	$-180^\circ$ bis $180^\circ$	-
$\theta_{Soll,Q_x}$	-	$\approx 65^\circ$	-

**Tabelle 9.3:** Sollwerte der Quellen (ein Sprecher)

Die entsprechenden Ergebnisse sind in Bild 9.12 aufgeführt. Dabei handelt es sich bei der Menge  $rfR$  um die Ergebnisse aus dem reflexionsfreien Raum, bei  $rUoF$  um die Ergebnisse in der realen Umgebung ohne Fehlerkorrektur und bei  $rUmF$  um die Ergebnisse in der realen Umgebung mit Fehlerkorrektur.

Die Auswertung der Ergebnisse führt zu folgenden Punkten: Der systematische Fehler tritt auch in der realen Umgebung auf, sobald keine Fehlerkorrektur vorgenommen wird. Falls die Fehlerkorrektur angeschaltet ist, reduziert sich der Fehler deutlich und stimmt nahezu mit den Messungen aus dem reflexionsfreien Raum überein. Die geringen Abweichungen liegen im Bereich der Messungenauigkeit. Wird die Varianz betrachtet, steigt diese leicht. Von einem deutlichen Anstieg kann jedoch nicht gesprochen werden.



**Bild 9.12:** Fehlerverläufe bei zwei aktiven Quellen bei  $\theta_{Soll} = 67^\circ$

### 9.4.3 Testreihe mit drei Sprechern

Für eine weitere Testreihe sind drei reale, männliche Sprecher verwendet worden. Diese sind entsprechend Tabelle 9.4 positioniert.

Größe	$Q_1$	$Q_2$	$Q_3$
$d_{Q_x}$	$\approx 0,75$ m	$\approx 0,65$ m	$\approx 0,77$ m
$h_{Q_x}$	$\approx 0,33$ m	$\approx 0,23$ m	$\approx 0,4$ m
$\varphi_{Soll,Q_x}$	$\approx -90^\circ$	$\approx 0^\circ$	$\approx 90^\circ$
$\theta_{Soll,Q_x}$	$\approx 66^\circ$	$\approx 70^\circ$	$\approx 62^\circ$

**Tabelle 9.4:** Sollwerte der Quellen (drei Sprecher)

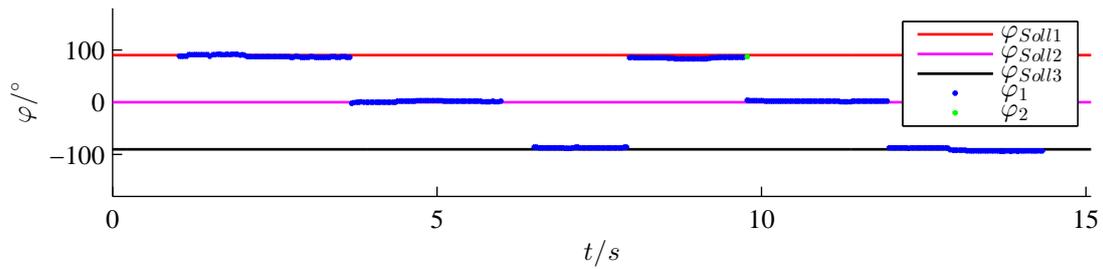
Anschließend sind zwei Tests mit unterschiedlichem zeitlichen Ablauf durchgeführt worden. Zunächst haben die Sprecher abwechselnd (entgegen dem Uhrzeigersinn) gezählt. Dabei hat jeder Sprecher um Drei weiter gezählt. Der entsprechende zeitliche Verlauf der Ergebnisse (Azimutwinkel, Elevationswinkel und Wert des Histogramms) ist in den Bildern der Bildergruppe 9.13 dargestellt.

Die Ergebnisse zeigen einen treppenstufenartigen Verlauf. Abwechselnd (entgegen dem Uhrzeigersinn) werden die entsprechenden Sprecher detektiert. Aufgrund der leichten Bewegungen der Sprecher, variieren die Ergebnisse leicht. Auch ist im Verlauf des Histogrammwerts eine Zu- und Abnahme erkennbar. Es lässt sich ansatzweise daraus schließen, dass jeder Sprecher drei Zahlen weiter gezählt hat und somit jeweils zwei Pausen eingelegt haben sollte.

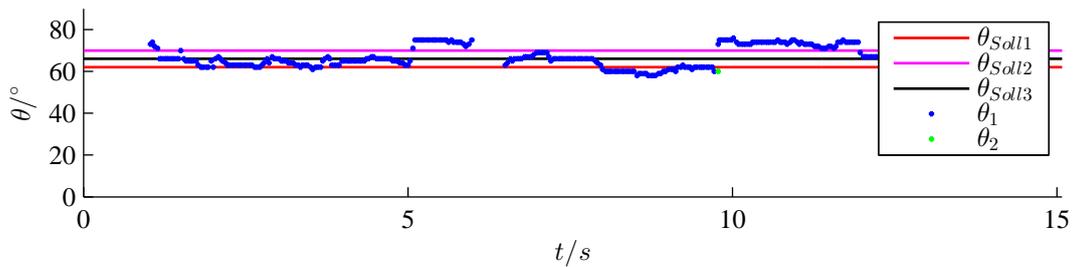
Um auch Aussagen über das System aufstellen zu können, wie es auf mehrere gleichzeitig aktive Sprecher reagiert, ist auch dieser Fall entsprechend getestet worden. Dazu zählt der Sprecher  $Q_2$  laut und deutlich, während der Sprecher  $Q_3$  schnell und leise buchstabiert und der Sprecher  $Q_1$  sich gleichzeitig vorstellt. Die Ergebnisse sind in der Bildergruppe 9.14 gezeigt.

Es ist deutlich ersichtlich, dass mehrere Sprecher über den abgebildeten Zeitraum aktiv gewesen sind. Das System detektiert bis zu zwei Sprecher gleichzeitig. Es kommt zu keiner Mittelung zwischen den Sprechern. Es handelt sich um ein positives Ergebnis, wird berücksichtigt, dass jeder Sprecher über gewisse Zeiträume zum Luftholen und zur Worttrennung pausiert bzw. keinen Schall aussendet. Auch beschränkt die Histogrammlänge die Anzahl der möglichen Detektionen. Um jedoch alle drei Quellen gleichzeitig zu detektieren, wäre die Verlängerung des Histogrammbuffers denkbar.

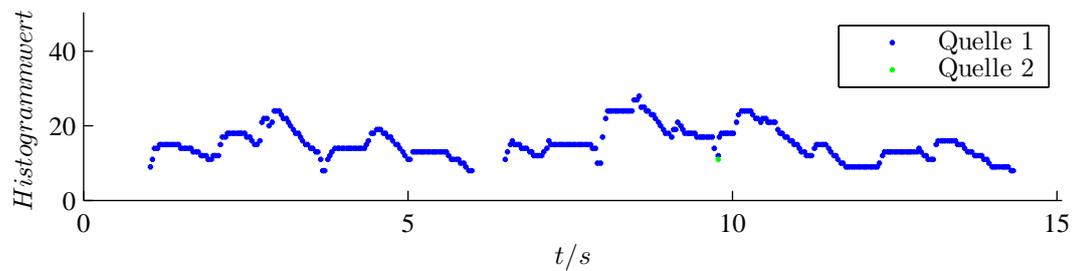
Die genauere Begutachtung der Ergebnisse bei ca. 4 s zeigt, dass es seltenst zu einer Mehrfachdetektion einer Quelle kommt. Dies lässt sich durch die Bewegung der Quelle und durch Mehrwegausbreitung erklären. In Kombination mit der Streuung der Zwischenergebnisse bilden sich im Histogramm breite Peaks aus, die die Mehrfachdetektion ermöglichen.



(a) Verlauf der Azimutwinkel

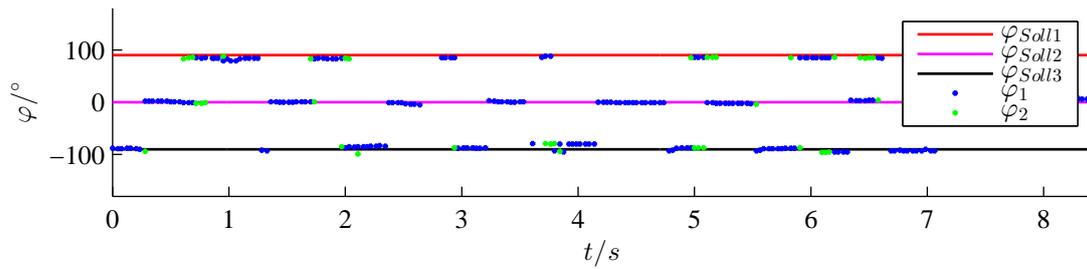


(b) Verlauf der Elevationswinkel

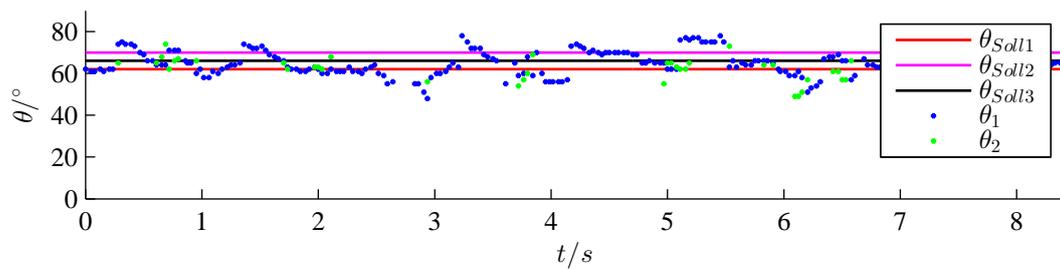


(c) Verlauf des Histogrammwerts

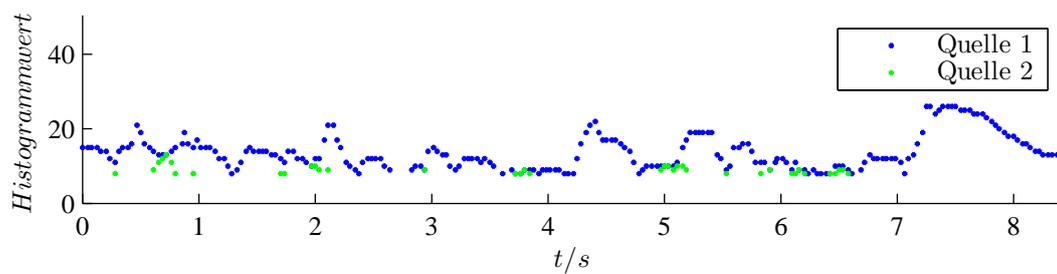
**Bild 9.13:** Ergebnisverlauf bei chronologisch aktiven Quellen



(a) Verlauf der Azimutwinkel



(b) Verlauf der Elevationswinkel



(c) Verlauf des Histogrammwerts

**Bild 9.14:** Ergebnisverlauf bei parallel aktiven Quellen

Allerdings kann dieser Effekt verringert bzw. sogar vermieden werden. Die Verringerung der Histogrammauflösung zu Lasten der Separierbarkeit verringert die Wahrscheinlichkeit dafür, dass sich Streuungen der Ergebnisse über mehrere Histogrammstützstellen erstrecken. Folglich ist eine Mehrfachdetektion seltener.

Weiter ist zu berücksichtigen, dass es sich bei allen drei Sprechern um männliche Sprecher handelt. Da sich die Grundfrequenzen ähneln, ist die Wahrscheinlichkeit dafür, dass mehrere Sprecher zu einer Frequenzstützstelle beitragen, durchaus als hoch anzusehen. Folglich ist die Stabilität des Systems als hoch einzustufen.

## 9.5 Bewertung

Die Ergebnisse der Tests in den vorherigen Abschnitten lassen sich als durchweg positiv bezeichnen.

Die grundsätzliche Genauigkeit des Systems liegt durch eine entsprechende Fehlerkorrektur in Azimutrichtung bei ca.  $1^\circ$  und in Elevationsrichtung bei ca.  $4^\circ$ . Die Genauigkeit verringert sich bei zwei aktiven Quellen oder in einer realen Umgebung lediglich geringfügig. Dies lässt auf eine hohe Robustheit schließen.

Ebenfalls hat sich gezeigt, dass die Varianz des Systems sowohl im reflexionsarmen Raum als auch in der realen Umgebung sehr gering ist. Dies ist ein Indikator für eine gute Stabilität des Systems. Auch ist kein deutlicher Anstieg der Varianz bei mehreren aktiven Quellen erkennbar.

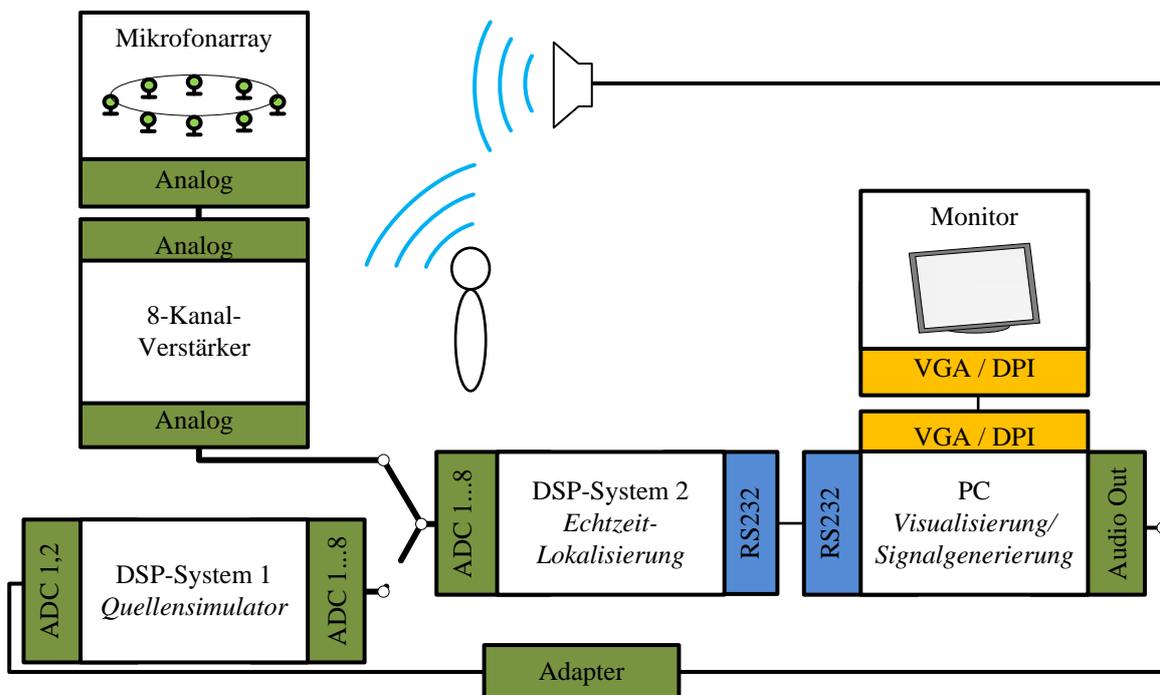
Die Trennung der einzelnen Quellen ist im Rahmen der Systemdimensionierung gegeben. Der auftretende Effekt der Mehrfachdetektion in der realen Umgebung kann mit entsprechenden Gegenmaßnahmen vermieden werden.

Die Reaktionszeit kann ebenfalls als sehr gut bezeichnet werden. Vor allem wenn beachtet wird, dass die gemessene Situation entsprechend den vorherigen Ausführungen nicht der Realität entspricht. Bei der Lokalisierung von Sprechern ist mit deutlich geringeren Reaktionszeiten zu rechnen.

Insgesamt gehen somit die gemessenen Ergebnisse über die quantitativen Zielgrößenordnungen aus Abschnitt 1.3 hinaus. Weiter zeigen die Messungen stets eine geringe Abweichung zur Theorie. Folglich kann das System entsprechend der Theorie dimensioniert werden. Dies stellt für weitere Implementierungen eine wesentliche Erkenntnis dar.

## 10 Zusammenfassung und Ausblick

In dieser Arbeit ist die Echtzeit-Lokalisierung von Audioquellen basierend auf dem UCA-ESPRIT-Algorithmus betrachtet worden. Dazu zählt unter anderem die Konzepterarbeitung für die Verarbeitung von Sprachsignalen, die Konzeptumsetzung und Untersuchungen bezüglich der Genauigkeit, Stabilität, Robustheit und Reaktionszeit. Weiter ist neben der Visualisierung der Ergebnisse auch ein Quellensimulator entwickelt worden. Das entwickelte Gesamtsystem ist in Bild 10.1 abgebildet.



**Bild 10.1:** Gesamtsystem

Prinzipiell sind drei Signalpfade möglich. Diese ergeben sich dadurch, dass die Signalgenerierung durch externe Quellen, wie Sprecher, oder über den PC erfolgen kann. Die Bereitstellung der Mikrofonsignale kann durch den Quellensimulator oder durch das Mikrofon mit einer entsprechenden Vorverstärkung erfolgen.

Für den Quellensimulator ist auf eine Methode mittels Spiegelung für die variable Verschiebung der Phasen zurückgegriffen worden. Als Alternative ist die Methode nach Pei und Lai

betrachtet worden. Diese weist in den Simulationen und Messungen jedoch größere Fehler auf.

Für die Echtzeit-Lokalisierung sind zunächst die theoretischen Grundlagen diskutiert worden, bevor durch den Übergang in den Spektralbereich ein Konzeptansatz entwickelt worden ist. Aufbauend auf diesem Ansatz ist die Konzeptionierung des Systems durchgeführt worden. Dabei ist durch die haltbare Annahme, dass stets nur eine Quelle vorhanden ist, eine deutliche Reduzierung des numerischen Aufwands erreicht worden. Dies hat in der anschließenden Implementierung das Erreichen der Echtzeitfähigkeit deutlich begünstigt. Um trotzdem mehrere Quellen detektieren zu können, ist eine anschließende, zweistufige Histogrammanalyse eingeführt worden. Diese dient darüber hinaus zur Glättung der Zwischenergebnisse.

Die anschließenden Tests im reflexionsfreien Raum haben einen deutlichen systematischen Fehler aufgezeigt, der durch eine Fehlerkorrektur nahezu vollständig entfernt worden ist. Die Abweichung vom Sollwert liegt sowohl in Azimut- als auch in Elevationsrichtung nie über und teils sogar deutlich unter  $5^\circ$ . Auch die Varianz des Systems ist als gering einzustufen. Tests in einer realen Umgebung haben ebenfalls gezeigt, dass der Fehler und die Varianz keiner relevanten Steigerung unterliegen. Auch zeigten weitere Tests, dass sich die Reaktionszeit des Systems entsprechend der Theorie verhält und somit parametrisiert werden kann. Gleiches gilt für die Trennbarkeit zweier Quellen. Ebenfalls sind die Ergebnisse in einer realen Umgebung mit Sprechern als sehr gut einzustufen.

Insgesamt bleibt somit festzuhalten, dass umfangreiche Aussagen über die Genauigkeit, Robustheit, Stabilität und Reaktionszeit dieser Arbeit entnommen werden können. Auch die aufgeführten quantitativen Zielgrößenordnungen werden deutlich erreicht.

Wie in der Einleitung dieser Arbeit beschrieben, gliedern sich die bisherigen Arbeiten des Forschungsprojekts MERIT in zwei Gruppen. Der Vorteil der auf Korrelation basierenden Verfahren ist im Wesentlichen auf den geringen numerischen Aufwand begrenzt. Sie bieten sich somit für eine echtzeitfähige Implementierung an. In dieser Arbeit ist aber gezeigt worden, dass der Aufwand des unterraumbasierten Verfahrens UCA-ESPRIT deutlich reduziert werden kann und der Algorithmus echtzeitfähig ist. Es ist sogar möglich, die RS232-Kommunikation in Echtzeit durchzuführen. Dies ist bei den auf Korrelation basierenden Verfahren zumeist nicht der Fall. Weiter weist das Verfahren eine kontinuierliche Auflösung auf, die Genauigkeit ist hoch und die Varianz ist gering. Es wird somit empfohlen die unterraumbasierten Verfahren in weiteren Arbeiten zu betrachten.

Weiter ist in dieser Arbeit eine grafische Oberfläche entwickelt worden. Diese ermöglicht die Darstellung der Ergebnisse in Echtzeit. Dabei wird sowohl der aktuelle Einfallswinkel, als auch die Historie für einen gewissen Zeitraum dargestellt. Ebenfalls ermöglicht das Programm die Erstellung von Protokollen, welche für die Durchführung und anschließende Auswertung von Tests benötigt werden.

Aufbauend auf dieser Arbeit sind einige weitere Arbeiten denkbar. Die Grundzüge dieser Ideen seien nachfolgend kurz erläutert:

- Die Messungen über den kompletten möglichen Winkelbereich könnten automatisiert und durchgeführt werden. Mit Hilfe der Messergebnisse wäre es anschließend möglich, den systematischen Fehler aufbauend auf dieser Arbeit über den kompletten Wertebereich zu modellieren und entsprechend zu korrigieren. Eine weitere Reduzierung des Fehlers ist zu erwarten und neue Mikrofonarrays könnten schnell und umfassend vermessen werden.
- Für die Spezialisierung auf eine Anwendung wäre die Implementierung von weiteren Nachbearbeitungsschritten sinnvoll. Denkbar ist dabei unter anderem die Implementierung von Tracking-, Vorhersage- und Klassifizierungsverfahren wie zum Beispiel das Kalman-Filter. Ebenso sind weitere Maßnahmen zur Fokussierung auf einen Sprecher und entsprechender Störunterdrückung denkbar.
- Während der Tests hat sich gezeigt, dass die Lokalisierung von Miniaturhubschraubern ausgezeichnet funktioniert. Denkbar wäre es durch die Verwendung eines zweiten Arrays und einer Kreuzpeilung die Position des Hubschraubers zu bestimmen. Aufbauend auf dieser Information wäre ein automatisierter Flug denkbar.
- Ein weiterer interessanter Aspekt wäre die Kombination der Lokalisierung mit einer kamerabasierten Personenerkennung. Mit Hilfe von Stellmotoren könnte die Kamera sich zum Sprecher ausrichten.

Grundsätzlich ist eine Kombination und oder Ergänzung bzw. Erweiterung der Ideen möglich.

# Literaturverzeichnis

- [1] ARENS, Tilo ; HETTLICH, Frank ; KARPFINGER, Christian ; KOCKELKOREN, Ulrich ; LICHTENEGGER, Klaus ; STACHEL, Hellmuth: *Mathematik*. 2. Heilderberg : Spektrum Akademischer Verlag, 2012. – ISBN 978-3-8274-2347-4
- [2] BELLONI, F. ; KOIVUNEN, V.: Beamspace Transform for UCA: Error Analysis and Bias Reduction. In: *Signal Processing, IEEE Transactions on* 54 (2006), Nr. 8, S. 3078–3089. – ISSN 1053-587X
- [3] BENESTY, Jacob ; CHEN, Jingdong ; HUANG, Yiteng: *Microphone Array Signal Processing*. 1. Berlin : Springer Verlag, 2008. – ISBN 978-3-540-78611-5
- [4] BERENS, Philipp ; VELASCO, Marc J.: *The circular statistics toolbox for Matlab*. April 2009. – URL [http://www.kyb.mpg.de/fileadmin/user\\_upload/files/publications/attachments/MPiK-TR-184\\_%5b0%5d.pdf](http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/MPiK-TR-184_%5b0%5d.pdf). – Abruf: 16. August 2013
- [5] CHENG, Y. ; LI, B.: *DSP based Audio Source Localization*. 2004. – Diplomarbeit
- [6] DAVIES, D. E. N.: Circular arrays. In: *The Handbook Of Antenna Design*. Peter Peregrinus, 1983
- [7] EULER, Stephen: *Grundkurs Spracherkennung*. Friedr.Vieweg & Sohn Verlag | GWV Fachverlage GmbH, 2006
- [8] HAARDT, Martin ; ZOLTOKOWSKI, Michael D. ; MATHEWS, Cherian P. ; RAMOS, Javier: ESPRIT and Closed-Form 2-D Angle Estimation with Planar Arrays. In: *Digital Signal Processing Handbook*. CRC Press, 1999
- [9] KAMMEYER, Karl-Dirk ; KROSCHER, Kristian: *Digitale Signalverarbeitung*. 8. Wiesbaden : Springer Vieweg Verlag, 2012
- [10] KÖLZER, Hans Peter ; SAUVAGERD, Ulrich: *MERIT - Mensch-Roboter-Interaktionssystem*. 2011. – URL <http://www.haw-hamburg.de/ti-ie/forschung/merit.html>. – Abruf: 22. Mai 2013
- [11] LAAKSO, T.I. ; VALIMAKI, V. ; KARJALAINEN, M. ; LAINE, U.K.: Splitting the unit delay [FIR/all pass filters design]. In: *Signal Processing Magazine, IEEE* 13 (1996), Nr. 1, S. 30–60. – ISSN 1053-5888

- [12] MATHEWS, Cherian P. ; ZOLTOWSKI, M.D.: Performance analysis of the UCA-ESPRIT algorithm for circular ring arrays. In: *Signal Processing, IEEE Transactions on* 42 (1994), Nr. 9, S. 2535–2539. – ISSN 1053-587X
- [13] MATHEWS, Cherian P. ; ZOLTOWSKI, Michael D.: Signal Subspace Techniques for Source Localization with Circular Sensor Arrays / Purdue University. 1994. – Forschungsbericht
- [14] MEYER, Carl D.: *Matrix Analysis and Applied Linear Algebra*. 1. Philadelphia : Society for Industrial and Applied Mathematics, 2004. – ISBN 978-0-898-71454-8
- [15] MÜLLER, Markus: *Sprecherlokalisierung von zwei oder mehr Sprechern im Raum unter Verwendung von Mikrofonarrays*. 2012. – Masterthesis
- [16] MOLER, Cleve: *Numerical Computing with MATLAB*. 1. Philadelphia : Society for Industrial and Applied Mathematics, 2004. – ISBN 978-0-898716-60-3
- [17] NÖLER, Norbert ; KLEMENZ, Adolf: *D.Module.C6713*
- [18] NÖLER, Norbert ; KLEMENZ, Adolf: *D.Module.PCM3003*
- [19] ORACLE: *Java3D*. – URL <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-138252.html>. – Abruf: 10. August 2013
- [20] PANASONIC: *Panasonic Sensors&Transducers*. – URL [http://www.angliac.co.uk/product\\_search/datasheets/process.asp?datasheet\\_id=16700](http://www.angliac.co.uk/product_search/datasheets/process.asp?datasheet_id=16700). – Abruf: 16. July 2013
- [21] PEI, Soo-chang ; YUN-CHIU, Lai: Closed Form Variable Fractional Time Delay Using FFT. In: *Signal Processing Letters, IEEE* 19 (2012), Nr. 5, S. 299–302. – ISSN 1070-9908
- [22] PFISTER, Beat ; KAUFMANN, Tobias: *Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer-Verlag, 2008
- [23] PIKORA, Kolja: *DSP-basiertes Echtzeitsystem zur Sprecherlokalisierung mittels Mikrofonarray und Root-MUSIC*. 2009. – Diplomarbeit
- [24] PIKORA, Kolja: *Korrelationsalgorithmen zur Sprecherlokalisierung mittels DSP-basiertem Echtzeitsystem und Mikrofonarray*. 2011. – Masterthesis
- [25] PRESS, William H. ; TEUKOLSKY, Saul A. ; VETTERLING, William T. ; FLANNERY, Brian P.: *Numerical Recipes*. 3. Cambridge : Cambridge University Press, 2007
- [26] PROAKIS, John G. ; MANOLAKIS, Dimitris G.: *Digital Signal Processing*. 4. New Jersey : Pearson Education, 2007

- [27] ROY, R. ; KAILATH, T.: ESPRIT-estimation of signal parameters via rotational invariance techniques. In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 37 (1989), Nr. 7, S. 984–995. – ISSN 0096-3518
- [28] RXTX-COMMUNITY: *RXTX-Bibliothek*. – URL <http://rxtx.qbang.org/wiki/index.php/Download>. – Abruf: 16. August 2013
- [29] SAXENA, Anshul Kant: *Wideband Audio Source Localization using Microphone Array and MUSIC Algorithm*. 2009. – Masterthesis
- [30] SCHMIDT, R.O.: Multiple emitter location and signal parameter estimation. In: *Antennas and Propagation, IEEE Transactions on* 34 (1986), Nr. 3, S. 276–280. – ISSN 0018-926X
- [31] SOFTWARE, Numerical R.: *Numerical Recipes, Webnote No. 2, Rev. 1*. 2007. – URL <http://www.nr.com/webnotes/nr3web2.pdf>. – Abruf: 16. August 2013
- [32] TEUTSCH, Heinz: *Modal Array Signal Processing: Principles and Applications of Acoustic Wavefield Decomposition*. 1. Berlin : Springer Verlag, 2007. – ISBN 978-3-540-40893-2
- [33] TEXAS INSTRUMENTS: *TMS320C6713B Floating-Point Digital Signal Processor*. October 2005
- [34] TEXAS INSTRUMENTS: *TMS320C67x DSP Library Programmers Reference Guide*. June 2005
- [35] TEXAS INSTRUMENTS: *TMS320C6000 Optimizing Compiler v7.4*. July 2012
- [36] VAN TREES, Harry L.: *Optimum Array Processing*. John Wiley & Sons, 2002
- [37] VARY, Peter ; HEUTE, Ulrich ; HESS, Wolfgang: *Digitale Sprachsignalverarbeitung*. Stuttgart : B. G. Teubner, 1998
- [38] VARY, Peter ; MARTIN, Rainer: *Digital Speech Transmission*. Chichester : John Wiley & Sons, 2006
- [39] ZOLTOWSKI, M.D. ; MATHEWS, Cherian P.: Closed-form 2D angle estimation with uniform circular array via phase mode excitation and ESPRIT. In: *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, 1993, S. 169–173 vol.1. – ISSN 1058-6393
- [40] ZOU, Zhonglei: *DSP-based System for 2D Wideband Acoustic Source Localization Using UCA-ESPRIT*. 2011. – Masterthesis

# A Mathematische Zusammenhänge

## A.1 Mittelwert und Varianz von Winkeln

Für den linearen Mittelwert  $\bar{x}$  von  $N$  Werten  $x_i$  mit  $i = 0, \dots, N - 1$  zusammengefasst im Vektor  $\mathbf{x}$  gilt:

$$\bar{x} = \frac{1}{N} \sum_{i=0}^{N-1} x_i \quad (\text{A.1})$$

Diese Gleichung kann jedoch nicht auf periodische Größen, wie Winkel, übertragen werden. Vielmehr muss der zirkulare Mittelwert gebildet werden. Für  $N$  Winkel  $\alpha_i$  zusammengefasst im Vektor  $\alpha$  mit  $i = 0, \dots, N - 1$  gilt [4, S. 4]:

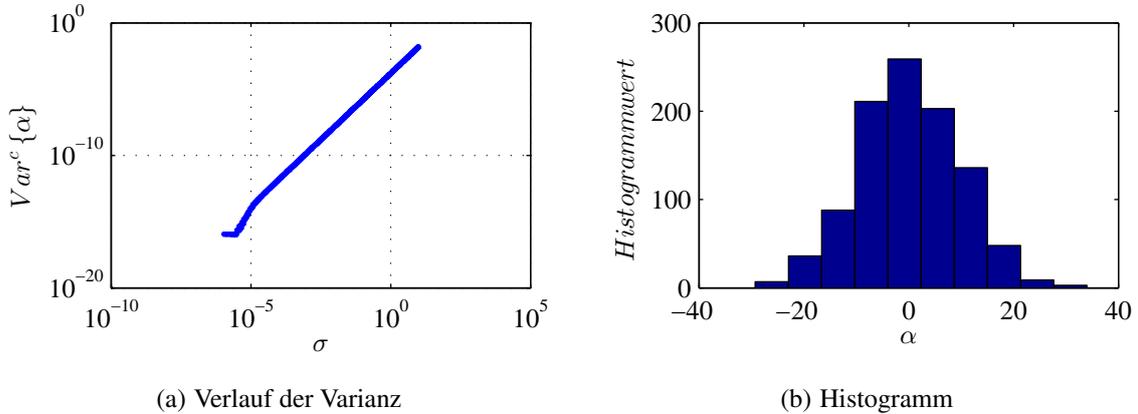
$$\bar{\alpha} = \arg \left\{ \sum_{i=0}^{N-1} e^{j\alpha_i} \right\} \quad (\text{A.2})$$

Ein Winkel  $\alpha$  wird somit in eine komplexe Zahl mit der Länge eins und der Phase  $\alpha$  uminterpretiert. Anschließend wird die Summe aller komplexen Zahlen gebildet und die Phase bestimmt.

Für die Varianz wird die Länge des resultierenden Vektors betrachtet. Es gilt nach [4, S. 5]:

$$\text{Var}^c \{ \alpha \} = 1 - \left| \frac{1}{N} \sum_{i=0}^{N-1} e^{j\alpha_i} \right| \quad (\text{A.3})$$

Falls alle Winkel nahe beieinander liegen, wird die Varianz nahezu null werden. Streuen die Winkel jedoch über einen weiteren Bereich, so wird die Varianz gegen null gehen. In Bild A.1(a) der Bildergruppe A.1 ist der Verlauf der Varianz für eine Normalverteilung der Winkel über die Standardabweichung  $\sigma$  dargestellt. Zu berücksichtigen bei diesem Verlauf ist, dass es sich bei der Normalverteilung um keine zirkulare Normalverteilung handelt und somit das Quadrat der Standardabweichung nicht der Varianz entspricht. In Bild A.1(b) der Bildergruppe A.1 ist das Histogramm des Winkeldatensatzes mit der größten Standardabweichung aus Bild A.1(a) dargestellt.



**Bild A.1:** Zirkulare Varianz

## A.2 Eigen- und Singulärwertzerlegung

Die folgenden Ausführungen basieren auf [1], [14] und insbesondere auf [16]. Sie stellen eine kurze, prägnante Zusammenfassung wichtiger Eigenschaften der Eigen- und Singulärwertanalyse dar, die in dieser Arbeit benötigt werden.

Die Eigenwertzerlegung einer Matrix  $\mathbf{A}$  ist über folgenden Zusammenhang definiert:

$$\mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{x}_i \quad (\text{A.4})$$

Dabei ist  $\mathbf{x}_i$  der  $i$ -te Eigenvektor und  $\lambda_i$  der dazugehörige Eigenwert. Die Matrix  $\mathbf{A}$  hat die Dimension  $N \times N$ . Sie ist somit quadratisch. Diese Gleichung kann auch in der Form

$$\mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{I}_N\mathbf{x}_i \quad (\text{A.5})$$

$$(\mathbf{A} - \lambda_i\mathbf{I}_N)\mathbf{x}_i = 0 \quad (\text{A.6})$$

beschrieben werden. Falls  $\mathbf{x}_i \neq 0$  ist diese Gleichung lediglich lösbar, falls

$$\det(\mathbf{A} - \lambda_i\mathbf{I}_N) = 0 \quad (\text{A.7})$$

gilt. Diese Determinante wird auch als charakteristische Gleichung der Matrix  $\mathbf{A}$  bezeichnet.

In Matrizen zusammengefasst gilt für die Eigenvektoren und Eigenwerte:

$$\mathbf{A}\mathbf{X} = \mathbf{X}\Lambda \quad (\text{A.8})$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix} \quad (\text{A.9})$$

$$\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_N\} \quad (\text{A.10})$$

Die Singulärwertzerlegung  $\mathbf{B}$  ist hingegen über den Zusammenhang

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger \quad (\text{A.11})$$

$$\mathbf{B} = \mathbf{U} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^\dagger \quad (\text{A.12})$$

beschrieben. Dabei hat  $\mathbf{B}$  die Dimension  $M \times N$  und muss somit nicht quadratisch sein. Die Matrizen  $\mathbf{U}$  und  $\mathbf{V}$  sind unitär. Sie haben die Dimension  $M \times M$  bzw.  $N \times N$ . Die Matrix  $\mathbf{\Sigma}$  ist eine Diagonalmatrix, wobei die Einträge stets größer gleich Null sind.

Bildet man das Produkt  $\mathbf{B}^\dagger \mathbf{B}$ , so gilt:

$$\mathbf{B}^\dagger \mathbf{B} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger)^\dagger (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger) \quad (\text{A.13})$$

$$= (\mathbf{V}\mathbf{\Sigma}^T \mathbf{U}^\dagger) (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger) \quad (\text{A.14})$$

Da  $\mathbf{\Sigma}$  eine Diagonalmatrix ist und  $\mathbf{U}$  unitär ist, folgt:

$$\mathbf{B}^\dagger \mathbf{B} = \mathbf{V}\mathbf{\Sigma}^2 \mathbf{V}^\dagger \quad (\text{A.15})$$

Mit der Multiplikation von rechts mit  $\mathbf{V}$  gilt, da  $\mathbf{V}$  unitär ist:

$$\mathbf{B}^\dagger \mathbf{B} \mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2 \quad (\text{A.16})$$

Ein Vergleich der Gleichungen A.8 und A.16 ergibt, dass  $\mathbf{\Sigma}^2$  die Eigenwerte von  $\mathbf{B}^\dagger \mathbf{B}$  darstellt.

Ein interessanter Sonderfall stellt die Eigen- bzw. Singulärwertzerlegung einer reellen, symmetrischen Matrix dar. Für eine reelle, symmetrische Matrix  $\mathbf{C}$  gilt:

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{A.17})$$

$$\mathbf{C}^T = \mathbf{V}\mathbf{\Sigma}\mathbf{U}^T \quad (\text{A.18})$$

Da  $\mathbf{C}$  symmetrisch ist, müssen folglich die Matrizen  $\mathbf{V}$  und  $\mathbf{U}$  übereinstimmen. Es folgt:

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T \quad (\text{A.19})$$

Da  $\mathbf{U}$  orthogonal ist, gilt ebenfalls:

$$\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^{-1} \quad (\text{A.20})$$

Wird die Eigenwertzerlegung ebenfalls auf  $\mathbf{C}$  angewendet, gilt:

$$\mathbf{C}\mathbf{X} = \mathbf{X}\mathbf{\Lambda} \quad (\text{A.21})$$

Da die Eigenvektoren einer reellen, symmetrischen Matrix orthogonal sind, existiert die inverse Matrix und es gilt:

$$\mathbf{C} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1} \quad (\text{A.22})$$

Beim Vergleich der Gleichungen A.20 und A.22 stimmen Eigen- und Singulärwertzerlegung überein, falls es sich bei  $\mathbf{C}$  um eine positiv semidefinite Matrix handelt. Des Weiteren müssen alle Eigen- bzw. Singulärwerte zueinander verschieden sein, um Eindeutigkeit zu garantieren.

## B Inhalt des Datenträgers

Der beigelegte Datenträger ist wie folgt strukturiert:

- **/01\_Masterarbeit\_JensReermann.pdf**

Es handelt sich um die vorliegende Masterarbeit im pdf-Format.

- **/02\_MATLAB/**

Dieser Ordner enthält jegliche *MATLAB*-Skripte, die in dieser Arbeit entwickelt und verwendet worden sind. In diesem Hauptordner sind alle relevanten Funktionen bzw. Skripte, die von verschiedenen Skripten verwendet werden, enthalten. Die Unterordner sind entsprechend folgender Auflistung strukturiert.:

- **AnalyzeTests/**: Skripte zur Analyse der Testreihen
- **FDF/**: Skripte bezüglich des variablen Phasenschiebers und Quellensimulators
- **Figures/**: Ordner zum Speichern von *MATLAB-Figures* als pdf-Datei
- **Data/**: Daten (Aufnahmen, Testergebnisse, etc.)
- **Localization/**: Skripte bezüglich der Lokalisierung

Die Hauptsimulationsdateien enthalten zumeist das Wort *Test* in ihrem Namen

- **/03\_Implementierung/**

Dieser Ordner beinhaltet drei *CodeComposerStudio5*-Arbeitsumgebungen.

- **fdf\_workspace/**: Variabler Phasenschieber
- **source\_simulator\_workspace/**: Quellensimulator
- **doa\_estimation\_workspace/**: Echtzeit-Lokalisierung

- **/04\_Visualisierung/** Dieser Ordner beinhaltet die *eclipse*-Arbeitsumgebung der grafischen Oberflächen, Testsequenzen, die *Java*-Datei *DOA\_ESTIMATION\_GUI.jar*, Ordner der Bibliotheken *Java3D* und *RXTX* sowie eine *ReadMe.txt* Textdatei mit Hinweisen zur Ausführung der *Java*-Datei.

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 11. Oktober 2013

Ort, Datum

Unterschrift