ulm university universität
uulm

Schichtübergreifender rechnergestützter Entwurf und Optimierung eines digitalen Kommunikationssystems – Ein modellbasierter Ansatz für die akustische Unterwasserkommunikation

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

(Dr.-Ing.)

der Fakultät für Ingenieurwissenschaften,
Informatik und Psychologie der Universität Ulm

von

Marcel Wolfgang Rieß

aus Leipzig

Gutachter: Prof. Dr.-Ing. Frank Slomka
Prof. Dr. rer. nat. Dipl.-Ing. Endric Schubert

Amtierende Dekanin: Prof. Dr. Anke Huckauf

Ulm, 29. Januar 2025

Inhalt

Symbolverzeichnis 7

Akronyme 11

Überblick 15

1. Kapitel **Einführung** 1

1.1 Unterwasserkommunikation 3
1.2 Herausforderungen 4
1.3 Entwicklung von Kommunikationssystemen 5
1.4 Wissenschaftlicher Beitrag 6
1.5 Aufbau der Dissertation 7

2. Kapitel **Stand der Forschung** 9

2.1 Aktuelle Entwicklungen 9
2.2 Modulationsverfahren 16
2.3 Modellbasierte Entwicklung 17
2.3.1 Allgemein 17
2.3.2 Unterwasserkommunikation 18
2.4 Kommunikationsstandard 19
2.5 Unterwasserkanalsimulation 20
2.6 Simulationsumgebung 22
2.7 Entwicklung und Plattformen 23

2.8 Adaptive Kommunikationssysteme 24

3. Kapitel Unterwasserkanal 27

3.1 Herausforderungen 28
3.2 Eigenschaften 31
3.2.1 Kanalimpulsantwort 31
3.2.2 Geschwindigkeitsprofil des Unterwasserkanals 33
3.2.3 Schichten des Unterwasserkanals 34
3.2.4 Signalverlust im Unterwasserkanal 37
3.2.5 Dopplereffekt 44
3.2.6 Umgebungsrauschen 45
3.2.7 Richtcharakteristik 48
3.3 Modellierung des Unterwasserkanals 50
3.3.1 Testpunkte 50
3.3.2 Modellierung der Kanalimpulseantwort (CIR) 54
3.3.3 Modellierung des Umgebungsrauschens 72

4. Kapitel Rahmenerkennung 83

4.1 Rahmensynchronisation 85
4.1.1 Barker-Code 89
4.1.2 Pseudo Noise (PN)-Folgen 89
4.1.3 Rahmenzuordnung 100
4.2 Rahmenoptimierung 100
4.3 Kasami-basierte Rahmenerkennung 102
4.3.1 Detektierbarkeit 103
4.3.2 Störanfälligkeit 116

5. Kapitel Varianten-basierter modularer Entwurf 121

5.1 Entwurfsmodell 122
5.1.1 Abhängigkeiten der Datenflussgrafen 128

5.1.2 Multivariater Datenflussgraf 130
5.1.3 Modifikationen 130
5.2 Modellbeschreibung 133
5.3 Plattformadaption 134
5.3.1 Verkettetes Variantenmodell 134
5.3.2 Flexibles Variantenmodell 135
5.3.3 Multiplexer-basiertes Variantenmodell 137
5.4 Umsetzung 139
5.4.1 Simulinkmodell 140
5.4.2 Verifikation 153
5.4.3 Referenzfunktionen und -modelle 155
5.4.4 Verifikation in Zahlen 157
5.4.5 Ressourcenanforderungen 159

6. Kapitel **Auswahl, Analyse und Optimierung** 163

6.1 Optimierungsansatz 164
6.1.1 Zielgrößen 165
6.1.2 Bewertung der Optimierung 166
6.1.3 SeeSim-Experimente 168
6.2 Reduzierung des Analyseraums 174
6.2.1 Geometrische und analytische Reduktion 176
6.2.2 Stochastische Reduktion 178
6.2.3 Ergebnisse der Analyseraumreduktion 179
6.3 Entwurfsraumbeschränkung 184
6.3.1 Kanalkapazität 185
6.3.2 Datenrate 186
6.3.3 Störabstand von Träger und Reflexion 187
6.3.4 Minimaler Signalstörabstand 188
6.3.5 Schutzabstand 188
6.3.6 Beschränkung des Entwurfsraums 193
6.3.7 Auswertung der Entwurfsraumreduzierung 196
6.4 Analyse- und Optimierungsablauf 201

7. Kapitel **Zusammenfassung und Ausblick** 205

7.1 Wissenschaftliche Methoden und Ansätze 206
7.2 Limitationen 206
7.3 Zusammenfassung und Ausblick 207

Anhang A **Anhang** 209

A.1 Umgebungen 209
A.2 Bodenbeschreibung 212
A.3 Ergebnisse der FIR-Filter-Ansätze 213
A.3.1 Rauschen von Turbulenzen 213
A.3.2 Thermalrauschen 214
A.3.3 Verkehrsrauschen 215
A.4 Ergänzungen zur Rahmenerkennung 216
A.5 Störanfälligkeit von Gold-Folgen (GF) 218
A.6 Kanalgenerierung 220
A.7 Reduktionsverfahren 235
A.8 Dokumententypedefinition (DTD) 242
A.9 Unit-Tests 243

Referenzen 249

Symbolverzeichnis

α	Absorptionskoeffizienten	dB
PL_{ab}	Absorptionsverlust im Wasser	dB
T_s	Abtastperiode	s
f_s	Abtastrate	Hz
Z_b	Akustischen Impedanz des Bodens	$\mathrm{gm/cm^3s}$
Z_w	Akustischen Impedanz des Wassers	$\mathrm{gm/cm^3s}$
A_s	Amplitude eines elektrischen Signals	V
F_{dec}	Anzahl der erfolgreich dekodierten Datenrahmen	
N_{taps}	Anzahl der Filtertaps	
F_{Tx}	Anzahl der gesendeten Datenrahmen	
F_{err}	Anzahl der nicht dekodierten Datenrahmen	
n_{SURF}	Anzahl der Oberflächenreflexionen	
B	Bandbreite des Kanals	Hz
H_{BER}	BER-Matrix für Basisband-Systeme	$\mathrm{error\,bit^{-1}}$
BER	Bitfehlerrate	$\mathrm{error\,bit^{-1}}$
bps	Bits pro Symbol	bit/s
$CINR$	Carrier-to-Interference-and-Noise Ratio	dB
$CINRR$	Carrier-to-Interference-Noise-and-Reflection Ratio	dB
R_{RS}	Codierrate Blockcodierer	
R_{CC}	Codierrate Faltungscodierer	
R_D	Datenübertragungsrate	
R_{dth}	Detektionsschwellwert der Korrelation	$\mathrm{V^2}$
ρ_b	Dichte des Seebodens	$\mathrm{g\,cm^{-3}}$
ρ_w	Dichte des Wassers	$\mathrm{g\,cm^{-3}}$
Δy	Dicke der abstrahierten Thermalschicht	
$\gamma_{xy}[f_k]$	Diskrete Kohärenzdichte von Signal x und y	
$S_{xy}[f_k]$	Diskrete Kreuzleistungsdichte von $x[n]$ und $y[n]$	$\mathrm{V^2/Hz}$
$S_{xx}[f_k]$	Diskretes Leistungsdichtespektrum $x[n]$	$\mathrm{V^2\,Hz^{-1}}$
$S_{yy}[f_k]$	Diskretes Leistungsdichtespektrum $y[n]$	$\mathrm{V^2\,Hz^{-1}}$
Δf	Doppler-Frequenzverschiebung	Hz
Δf_d	Doppler-Spreizung	Hz
θ_i	Einfallswinkel	°
OCR	Empfangsempfindlichkeit	$\mathrm{re\ V\,\mu Pa^{-1}}$
b_k	FIR-Filter-Koeffizient	
w_H	FIR-Filter-Koeffizienten des Hamming-Window	
w_K	FIR-Filter-Koeffizienten des Kaiser-Window	
w_{LS}	FIR-Filter-Koeffizienten des Least-Squares-Ansatz	
Kasami-Folge	Folge aus der kleinen Kasami-Gruppe	sym
f	Frequenz	Hz
f_k	Frequenzkomponente	Hz
f_{off}	Frequenzoffset	Hz

Symbole

HV	Hypervolumen	
h	Hypervolumen-Indikator	
T_i	i'te Senderposition	m
$h[k]$	Impulsantwort des Kanals (diskret)	
I_{BB}	Interpolationsfaktor	
H	Kanalimpulsantwortmatrix	
C	Kapazität	$\mathrm{bit\,s^{-1}}$
K_s	Kasami-Folge	sym
$\gamma_{xy}(f)$	Kohärenzdichte von Signal x und y	
R	Kommunikationsdistanz	km
R_b	Kommunikationsdistanz der Bodenreflexion	m
R_s	Kommunikationsdistanz der Oberflächenreflexion	m
R_{los}	Kommunikationsdistanz der Sichtverbindung	m
R_{th}	Korrelationsschwellwert	$\mathrm{V^2}$
ω	Kreisfrequenz	$\mathrm{rad\,s^{-1}}$
$C_{xy}(t)$	Kreuzkorrelationsfunktion zwischen x und y	
$S_{xy}(f)$	Kreuzleistungsdichte von x und y	$\mathrm{volt^2/Hz}$
μ_L	Lebesgue-Maß	
$S_{xx}(f)$	Leistungsdichtespektrum von $x(t)$	$\mathrm{V^2\,Hz^{-1}}$
$S_{yy}(f)$	Leistungsdichtespektrum von $y(t)$	$\mathrm{V^2\,Hz^{-1}}$
f^+	Maximale Frequenz	Hz
l_{gd}^+	Maximaler Schutzabstand	$\mathrm{sym\,s^{-1}}$
f^-	Minimale Frequenz	Hz
l_{gd}^-	Minimaler Schutzabstand	$\mathrm{sym\,s^{-1}}$
$\hat{\omega}$	Normierte Kreisfrequenz	rad
R_p	Nutzdatenrate	$\mathrm{bit\,s^{-1}}$
SL	Oberflächenverlust	dB
SL_{BS}	Oberflächenverlust nach Beckmann-Spizzichino	dB
SL_C	Oberflächenverlust nach Coates	dB
SL_{RL}	Oberflächenverlust nach Rayleigh	dB
R_P	Payload-Rate	
r	Radius	m
N_{est}	Rauschen durch Turbulenzen und Erdbeben	dB
R_{red}	Reduktionsrate der Testpunkte	
r_0	Referenzradius	m
R_s	Reflexionskoeffizienten der Oberfläche	
R_b	Reflexionskoeffizienten des Bodens	
θ_r	Reflexionswinkel	$\circ$
v_{TR}	Rel. Geschwindigkeit (Sender zu Empfänger)	$\mathrm{m\,s^{-1}}$
v_S	Relative strömungsspezifische Geschwindigkeit	$\mathrm{m/s}$
S	Salzgehalt	$\mathrm{g/kg}$
S	Salzgehalt	$\mathrm{g\,kg^{-1}}$
SPL	Schalldruckpegel	dB
c_0	Schallgeschwindigkeit auf der Tiefenkanal-Achse z_0	unit
c_b	Schallgeschwindigkeit im Seeboden	$\mathrm{m\,s^{-1}}$
c_w	Schallgeschwindigkeit im Wasser	$\mathrm{m\,s^{-1}}$
I	Schallintensität	$\mathrm{W/m^2}$
N_{sh}	Schiffsrauschen	dB
s	Schiffsverkehrsskala	
l_{gd}	Schutzabstand	$\mathrm{sym\,s^{-1}}$
s_{st}	Seegang	bft

TVR	Sendeempfindlichkeit	re µPa V^{-1} @ m
u	Sequenz maximaler Länge	sym
w	Sequenz maximaler Länge	sym
P	Signalleistung	W
η	Spektrale Effizients	
σ_h	Standardabweichung der Wasserhöhe	m
σ_n	Standardabweichung des Rauschens	V
k	Streufaktor des Kanals	
PL_{sp}	Streuverlust im Wasser	dB
f_s	Symbolrate	
T	Temperatur	°C
N_{th}	Thermisches Rauschen	dB
d_z	Tiefe in einem Gewässer	m
z_0	Tiefenkanal-Achse	m
θ_t	Transitionswinkel	°
$\hat{u}$	Trägerbündel	
f_c	Trägerfrequenz	Hz
N	Umgebungsrauschen	dB
Pa	Unterwasserdruck	Pa
$ACSR$	Verhältnis der Autokorrelation an der Stelle $K = 0$ zum Maximum allen anderen Korrelationswerten aller Folgen	
ASR	Verhältnis der Autokorrelation an der Stelle $k = 0$ zum Maximum des Korrelationswertes für $k \neq 0$	
R_{dec}	Verhältniss der erfolgreich dekodierten zu gesendeten Datenrahmen	
R_{err}	Verhältniss der nicht dekodierten zu gesendeten Datenrahmen	
T_{lsh}	Verschiebungsoperator (links) der Kasami-Folgen	sym
w_{sh}	Verschiebungsparameter einer m-Sequenz	
v_{SURF}	Vertikale Wellengeschwindigkeit der Wasseroberfläche	m/s
τ	Verzögerung	s
τ_H	Verzögerung der Signale (Impulsantwortmatrix)	s
τ_{FIR}	Verzögerung des Rauschens mit FIR-Filter	s
λ	Wellenlänge	m
c_{wind}	Windgeschwindigkeit	$\mathrm{m\,s}^{-1}$
N_w	Windrauschen	dB

Akronyme

ACK	ACKnowledgement
STTF	Abtastzeittranfersfunktion
ALM	Adaptive Logic Modules
ADC	Analog Digitial Converter
AUV	Autonomes Unterwasserfahrzeug
BGS	Balanced Gold Sequence
BPSK	Binary Phase Shift Keying
BER	Bitfehlerrate
BASS	Broadband Acoustic Spread Spectrum
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSMA	Carrier Sense Multiple Access
CTS	Clear To Send
CDMA	Code Division Multiple Access
CSV	Comma-Separated Values
CF-End	Contention-Free-End-Frame
CI	Continuous Integration
CSD	Cross-Spectral Density
DATA	DATA
DSE	Design Space Exploration
DSIR	Difference in Signal to Interference Ratio
DAC	Digitial Analog Converter
DTD	Dokumenttypdefinition
XML	Erweiterbare Auszeichnungssprache
EA	Evolutionary Algorithm
FIL	FPGA in the Loop
FFT	Fast Fourier Transformation
FPGA	Field Programmable Gate Array
FIR	Finite Impulse Response
FIFO	First In – First Out
FAU	Florida Atlantic University
FH-SS-Mod	Frequency-Hopping Spread Spectrum-Modulation
FSK	Frequenz-Umtastung
FH-FSK	Frequenzsprungverfahren kombiniert mit Frequenzumtastung
GEBCO	General Bathymetric Chart of the Oceans
SSP	Geschwindigkeitsprofil

Akronyme

GF	Gold-Folgen
HIL	Hardware in the Loop
HDL	Hardwarebeschreibungssprache
IEEE	Institute of Electrical and Electronics Engineers
IFI	Inter-Frame-Interferenz
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
IFFT	Inverse Fast Fourier Transformation
JANUS	Joint Acoustic Network Using Standard
CIR	Kanalimpulseantwort
CTF	Kanalübertragungsfunktion
KF	Kasami-Folgen
KLD	Kullback-Leibler-Divergenz
LFSR	Linear-Feedback Shift Register
LE	Logical Elements
MPDU	Mac Protocol Data Unit
MIT	Massachusetts-Institut für Technologie
MF	Maximalfolgen
MAC	Media Access Control
MFSK	Mehrfrequenz-Umtastung
MCO	Multi-Kriterien-Optimierung
NOAA	National Oceanic and Atmospheric Administration
NATO	North Atlantic Treaty Organization
OFDM	Orthogonales Frequenzmultiplexverfahren
PSK	Phasenumtastung
PCDU	Physical Channel Data Unit
PPDU	Physical Protocol Data Unit
PDU	Protocol Data Unit
PN	Pseudo Noise
QPSK	Quadraturphasenumtastung
RAM	Random-Access Memory
NSL	Rauschspektralpegel
RSC	Reed Solomon Codierer
RTS	Request To Send
SOFAR	SOund Fixing And Ranging
SBP	Schallstrahlmaske
SDU	Service Data Unit
LOS	Sichtverbindung
SIR	Signal to Interference Ratio
SNR	Signal-Rausch-Verhältnis
SDR	Software Defined Radio
S2C	Sweep Spread Carrier
DSC	Tiefenkanal
CNR	Träger-Rausch-Abstand

UNESCO	United Nations Educational Scientific and Cultural Organization
UWA	Unterwasserakustik
UWK	Unterwasserkanal
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VirTEX	Virtual Timeseries EXperiment
FEC	Vorwärtsfehlerkorrektur
WHOI	Woods Hole Oceanographic Institution
WMO	World Meteorological Organization
WOSS	World Ocean Simulation System
CRC	Zyklische Redundanzprüfung
ACS	acoustic channel simulator
AWGN	additives weißes gaußsches Rauschen
DSP	digitaler Signalprozessor
FC	frame control
MSE	mittlere quadratische Abweichung
OKF	orthogonale Kasami-Folgen
OKBFU	orthogonale Kasami-basierte Folgen ungerader Länge
OKBF	orthogonale Kasami-basierte Folgen
PSD	spektrale Leistungsdichte

Akronyme

Überblick

Die Dissertation untersucht die Entwicklung, Verifikation, Analyse und Optimierung eines generischen, modularen und erweiterbaren Kommunikationssystems unter Verwendung einer modellbasierten Methodik am Beispiel der akustischen Unterwasserkommunikation. Diese Methodik ermöglicht eine systematische Herangehensweise an die spezifischen Herausforderungen, die durch die physikalischen Bedingungen im Unterwasserbereich entstehen.

Dieser Forschungsbereich gewinnt zunehmend an Bedeutung, da Unterwasserkommunikation aufgrund der besonderen physikalischen Bedingungen eine erhebliche Herausforderung darstellt und immer stärker in den Fokus militärischer und ziviler Anwendungen rückt.

Die Arbeit leistet einen wichtigen Beitrag zur Weiterentwicklung von Kommunikationssystemen, die in schwierigen Umgebungen operieren müssen, und fügt sich in den Kontext der Kommunikationstechnologie sowie der systematischen Optimierung von Modellierungsansätzen und der Standardisierung von Testkonzepten ein.

Trotz signifikanter Fortschritte bestehen weiterhin bedeutende Forschungslücken, insbesondere hinsichtlich der Effizienz der Bandbreitennutzung und der Robustheit gegenüber variierenden Umweltbedingungen. Akustische Kommunikationskanäle im Wasser sind durch Mehrwegeausbreitung, Rauschen und variable Umgebungsbedingungen stark beeinträchtigt.

Diese Dissertation zielt darauf ab, bestehende Kommunikationstechniken modellbasiert in einem Modell zu kombinieren, um durch Optimierung mittels metaheuristischen Evolutionären Algorithmen über die Optimierung und festgelegten Testkonzepten eine effizientere Nutzung der Bandbreite und eine höhere Datenrate zu ermöglichen, ohne die Zuverlässigkeit der Kommunikation zu beeinträchtigen.

Der methodische Ansatz dieser Dissertation basiert auf der formalen Definition und Umsetzung eines flexiblen, parametrisierbaren und generischen Modells, das verschiedene Arten von Kommunikationssystemen durch Modularisierung und parametrisierte Varianten vereinheitlichen und synthetisierbar machen kann. Dieser Ansatz ermöglicht nicht nur die Modellierung und Analyse, sondern auch eine praktische Implementierung auf Hardware. Das Kommunikationssystem wird

dazu in Form von austauschbaren und parametrisierbaren Varianten entworfen, die verschiedene Funktionalitäten wie Modulation, Codierung und Fehlerkorrektur darstellen. Die Austauschbarkeit und Parametrisierung ermöglichen die Abbildung einer Vielzahl von Kommunikationssystemen und somit die Anpassung des Systems an spezifische Anforderungen und Umgebungen.

Für eine effizientere Kommunikation wird ein Kasami-basierter Ansatz eingeführt, der die präambelbasierte Synchronisation, Identifikation und Typisierung von Daten effizienter und robuster gestaltet. Die Präambel enthält Informationen, die normalerweise im Header platziert wären, was die Gesamtdatenrate erhöht und die Übertragungseffizienz verbessert. Verschiedene Verfahren werden eingesetzt, um sicherzustellen, dass die Präambel auch unter starkem Umgebungsrauschen und Interferenzen zuverlässig erkannt wird. Der Ansatz ist so gestaltet, dass er leicht für neue Kommunikationsszenarien erweitert werden kann.

Abschließend wird der modellbasierten Ansatz über eine definiertes Testkonzept mit dem Optimierungsframework OPT4J kombiniert und ermöglicht eine systematische Auswahl und Optimierung der umsetzbaren Kommunikationssysteme die den spezifischen Anforderungen und Umweltbedingungen gerecht werden.

Der wissenschaftliche Beitrag dieser Arbeit beinhaltet die Entwicklung eines umfassenden Testkonzepts für Unterwasserkommunikationssysteme durch die Einführung eines standardisierten Testkonzepts.Dieses Testkonzept ermöglicht die objektive Bewertung und der Vergleich verschiedener Unterwasserkommunikationssysteme und schafft eine Grundlage für die Validierung und Verbesserung von Unterwasserkommunikationssysteme ohne kostspielige und zeitaufwändige Feldtests. Gleichzeitig ermöglicht das Testkonzept eine klare Definition des Analyseraums und einheitlicher Bewertungskriterien, wodurch die Reproduzierbarkeit von Testergebnissen ermöglicht und verbessert wird.

Durch die Einführung von Kasami-basierten orthogonalen Sequenzen zur Detektion von Datenpaketen reduziert sich der Energieverbrauch und erhöht sich die Effizienz der Kommunikation, was sich insbesondere auf die Kommunikation in Netzwerken vorteilhaft auswirkt. Gleichzeitig reduziert der präambelbasierte Ansatz die Fehleranfälligkeit der Detektion und erhöht damit Zuverlässigkeit, was die Robustheit der Kommunikationssysteme in dynamischen Unterwasserumgebungen erhöht.

Der definierte erweiterte modellbasierte Ansatz ermöglicht eine flexible Anpassung und Erweiterung von Kommunikationssystemen und damit die kontinuierliche Entwicklung und Anpassung an neue Technologien und Anwendungen. Durch die modellbasierte Simulation und Verifikation können teure und aufwen-

dige Feldtests minimiert werden, was nicht nur Kosten einspart sondern auch Entwicklungszyklen beschleunigt.

Die Integration von Opt4J und OpenDSE in den Optimierungsworkflow ermöglicht die effiziente Analyse und Optimierung von Kommunikationssystemen hinsichtlich verschiedener Ziele, wie z.B. Energieeffizienz und Robustheit. Durch die spezifizierten Reduktionsverfahren wird der Aufwand zur Optimierung auf Basis kapazitärer Schranken signifikant reduziert, was die Entwicklung und Auswahl der Optimalen Kommunikationsstrategien erleichtert.

Die entwickelten Methoden ermöglichen die Anpassung der Kommunikationssysteme an zukünftige Herausforderungen und technologische Entwicklungen. Dies stellt sicher, dass die Systeme auch langfristig effektiv und relevant bleiben.

Der eingeführte methodische Rahmen hilft, die Komplexität der Entwicklung und Optimierung von Unterwasserkommunikationssystemen zu beherrschen und strukturierte Entwicklungsprozesse zu gewährleisten. Die Arbeit eröffnet damit neue Möglichkeiten für die Anwendung und Auswahl adaptiver Kommunikationssysteme, was zu einer breiteren Nutzung in verschiedenen wissenschaftlichen und praktischen Kontexten führen kann.

1

Einführung

„Das Meer bedeckt sieben Zehntel der Erdoberfläche, und der Seewind ist rein und gesund. In dieser unermesslichen Einöde ist der Mensch doch nie allein; denn er fühlt das Leben um ihn herum; ein übernatürliches, wundervolles Dasein rührt sich in demselben; es ist nur Bewegung und Liebe.“

Jules Verne (1828–1905), Schriftsteller

Unsere Erde ist zu etwa 70% von Wasser bedeckt, wodurch die Ozeane die größten Lebensräume darstellen. Trotz dieser immensen Ausdehnung sind lediglich etwa 5% der Ozeane erforscht worden. Die United Nations Educational Scientific and Cultural Organization (UNESCO) hat im Jahr 2021 mit dem Start einer Dekade der Ozeanforschung [1] das Ziel, ein tiefergehendes Verständnis dieses Ökosystems zu fördern.

Die Erforschung der Weltmeere umfasst verschiedene Aspekte, angefangen bei der Überwachung des Klimawandels und der Nutzung von Frühwarnsystemen zur rechtzeitigen Erkennung von Seebeben und der Warnung vor den resultierenden Flutwellen. Weitere Schwerpunkte liegen auf dem Kartografieren und Navigieren, der Höhlenforschung und der ozeanografischen Datenerfassung.

Die Erforschung der Ozeane erstreckt sich aber nicht nur auf rein wissenschaftliche Aspekte, sondern umfasst auch anwendungsorientierte Bereiche: Die Abbildung 1.1 zeigt dazu beispielhaft mögliche Anwendungen wie die Offshore-Explorationen, Höhlenerforschung, Überwachung der Wasserqualität von Fischfarmen und militärische Anwendungen wie ein Unterwasser-GPS und Kommunikationsbojen.

Grundsätzlich sind für die Verarbeitung von Daten zwei Ansätze möglich: Die Daten können entweder durch eine aufwendige Bergung und Extraktion der Unterwassersysteme oder durch eine direkte Übertragung verarbeitet werden.

Wobei die Extraktion der Daten nach einer Mission das Risiko birgt, dass die Systeme und damit die gesammelten Informationen verloren gehen[2].

Der Bedarf an Kommunikationsmitteln hängt aber von der jeweiligen Anwendung ab, wie der Erforschung des Verhaltens von Meerestieren wie Walen oder Delfinen. Hier spielt die Kommunikation zwischen Unterwassermodems eine entscheidende Rolle, um Informationen über Wanderungsmuster, Sozialstrukturen und Kommunikationsverhalten zu erhalten, welche das Verständnis für die Ökosysteme und deren Schutz verbessern sollen.

Die Unterwassermodems ermöglichen zudem eine Echtzeit-Überwachung von Ozeanströmungen, Temperaturprofilen und anderen Umweltfaktoren. Die Echtzeitinformationen sind für die Vorhersage von Wetterereignissen, das Verständnis des Klimas, die Entwicklung nachhaltiger Fischereistrategien und insbesondere eine Verbesserung von Frühwarnsystemen bei Flutwellen und anderen Ereignissen wichtig, deren rechtzeitige Kenntnis Menschenleben retten kann.

Heutzutage werden auch zunehmend Tauchroboter für die Inspektion und Reparatur von Unterwasserinfrastrukturen wie Öl- und Gasleitungen, Unterwasserkabeln oder Bohrsystemen eingesetzt. Die Kommunikation zwischen den Robotern und dem Kontrollzentrum ermöglicht eine Echtzeit-Überwachung, präzise Steuerung, effiziente Fehlerbehebung und frühzeitige Reaktion auf umweltschädliche Probleme wie Lecks.

Auch die Wichtigkeit von maritimen Verteidigungsstrategien spielt vermehrt eine Rolle. Dazu wird der Informationsaustausch mittels Unterwassermodems zur Koordinierung von U-Booten, Überwachungsdrohnen und Unterwasserabwehrsystemen genutzt. Eine Echtzeitkommunikation ermöglicht taktische Manöver, die Echtzeit-Überwachung feindlicher Aktivitäten und eine effektivere Durchführung von Operationen.

An diesen Beispielen wird deutlich, dass die Kommunikation zwischen Unterwassersystemen von zentraler Bedeutung ist, um das Verständnis und die Nutzung der Meeresumwelt sowie die Sicherheit von Menschen durch eine direkte Kommunikation und Datenaustausch in Echtzeit zu verbessern.

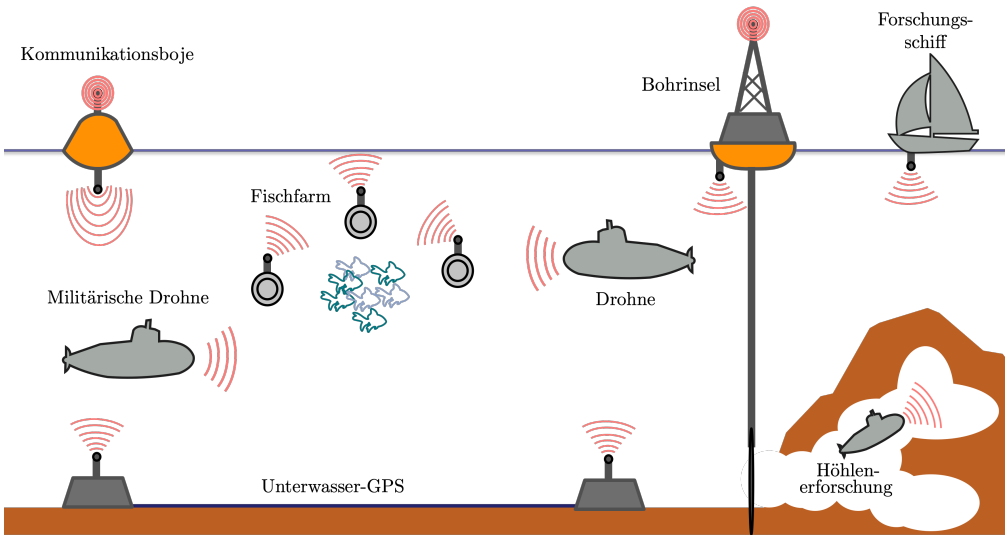


Abbildung 1.1: Schematischer Überblick der Unterwasserkommunikation

1.1 Unterwasserkommunikation

Die Unterwasserkommunikation unterscheidet zwischen der akustischen, elektromagnetischen und optischen Signalübertragung. Bei der terrestrischen drahtlosen Signalübertragung werden elektromagnetische Wellen in Abhängigkeit von der Frequenz absorbiert und sind nur für kurze Distanzen zwischen Sender und Empfänger effektiv nutzbar [3–9]. Eine mögliche Lösung wäre die Erhöhung der Sendeleistung und die Verwendung einer kleineren Trägerfrequenz, was jedoch eine entsprechende Antennengröße und Kapazität erfordert. Diese Option ist für große Trägersysteme wie bemannte U-Boote mit eigener Stromversorgung geeignet, jedoch nicht für autonome Fahrzeuge, deren Zuglast und Kapazität stark begrenzt sind. Hybrid-Systeme, die Daten zwischen den Medien Wasser und Luft übermitteln, sind eine Ausnahme und verwenden dazu mehrere Kommunikationssysteme. Ähnlich wie im terrestrischen Bereich kann die optische Signalübertragung auch unter Wasser für Hochgeschwindigkeitskommunikation genutzt werden. Allerdings sind aufgrund der Streuung und Absorption sowie der Notwendigkeit einer Sichtlinienkommunikation die Entfernungen im Wasser begrenzt[10–13].

Im Vergleich dazu hat sich die akustische Signalübertragung in den vergangenen Jahrzehnten als Hauptkommunikationsmittel für Kurz-, Mittel- und Langstrecken etabliert, trotz ihrer geringen Ausbreitungsgeschwindigkeit und begrenzten Datenrate [9, 14]. Die akustische Übertragung des Schalls im Wasser mit einer

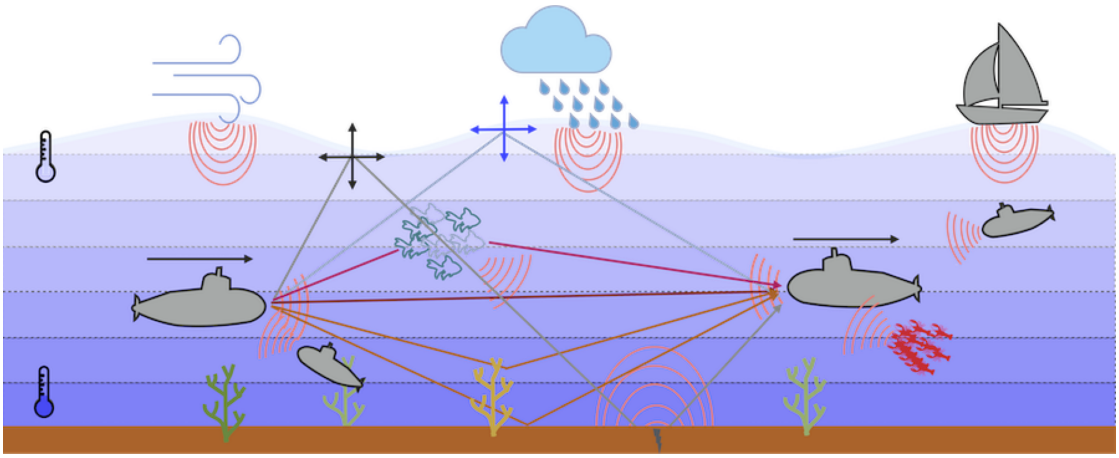


Abbildung 1.2: Herausforderungen in der Unterwasserkommunikation

Geschwindigkeit von etwa $c_w \approx 1500\,\mathrm{m/s}$ ist im Vergleich zur terrestrischen elektromagnetischen Kommunikation über Funk langsam, aber dennoch die bevorzugte Methode aufgrund ihrer Zuverlässigkeit.

1.2 Herausforderungen

Die akustische Unterwasserkommunikation, schematisch in Abbildung 1.2 dargestellt, wird über die Absorption im Wasser und die Streuung und Reflexion der Schallwellen an den natürlichen Grenzen (Seeboden, Seeoberfläche) gedämpft und verzerrt. Jegliche Bewegung von Sender, Empfänger und der Umgebung selbst (Flora, Fauna und Oberflächenwellen) führt zu einer zeitlich variierenden Beeinflussung der Schallwelle.

Die geringe Übertragungsgeschwindigkeit des Schalls im Wasser variiert zudem in Abhängigkeit der Umgebungsfaktoren wie Temperatur, Druck, Salinität, Strömungen und der relativen Geschwindigkeit zwischen Sender und Empfänger. Die unterschiedliche Ausbreitungsgeschwindigkeit, kombiniert mit Interferenzen von anderen Schallquellen, führt zu variierenden Verzögerungen, Signalpegel und Verzerrungen des Signals.

Neben den Herausforderungen des Unterwasserkanals und der gemeinsamen Nutzung des Mediums durch mehrere Kommunikationssysteme beeinflusst auch das von der Umgebung abhängige Unterwasserrauschen die Signaldetektion in Abhängigkeit von der Frequenz, der Tiefe und den Eigenschaften des Wassers selbst.

1.3 Entwicklung von Kommunikationssystemen

Die Herausforderungen des Unterwasserkanals zur akustischen Informationsübertragung legen nahe, dass für jede Umgebung ein Kommunikationssystem entwickelt und getestet wird, das an die Anforderungen der Anwendung und die Gegebenheiten des örtlichen Mediums angepasst ist. Dies hat zu einer Vielzahl verschiedener Kommunikationssysteme geführt [15]-[16]. Die entwickelten Systeme können nur mit mäßigem Erfolg, wie beispielhaft in [17] durch aufwendige Umrüstungen der Unterwassermodems von Evologic [18] und Woods Hole Oceanographic Institution (WHOI) [19], für alternative Anwendungen und Umgebungen genutzt werden.

Auch eine frühzeitige Designentscheidung, basierend auf Parametern wie Kanalkapazität, Datenrate und theoretischer Bitfehlerrate für eine festgelegte Anwendung, ist für den generellen Einsatz in verschiedenen Umgebungen nicht zielführend. Die aufgezeigten Parameter und Maxima der verfügbaren Kommunikationssysteme können lediglich als Ausschlusskriterium dienen. Diese lassen jedoch keine Aussagen über die vorherrschenden örtlichen Bedingungen und die dynamische Konstellation der Systeme zu. Daher kann mithilfe der von Herstellern und Forschung zur Verfügung gestellten Maxima und Parameter keine Aussage über die orts- und anwendungsspezifische Einsatzfähigkeit der Systeme getroffen werden.

Bei der Analyse der wissenschaftsorientierten Entwicklung von Unterwasserkommunikationssystemen fällt auf, dass der Fokus auf die Sicherungsschicht des International Organization for Standardization/Open Systems Interconnection (ISO/OSI)-Modells gelegt wird, wodurch der Einfluss und die Beschränkungen angrenzender Schichten oder komplexerer Szenarien vernachlässigt werden. Die Ergebnisse dieser eingeschränkten Perspektive bei der Entwicklung von Kommunikationssystemen können somit nicht auf eine spezifische Umgebung und Anwendung transferiert werden. Für die Wahl eines geeigneten Kommunikationssystems müssen schichtübergreifende Einflüsse wie die Herausforderungen der physikalischen Schicht, die Verzögerung des Mediums, die positions- und umgebungsbedingten Änderungen der Mehrwegeausbreitung, die von der Frequenz und Bandbreite abhängige Mehrnutzerkommunikation sowie der Einfluss von Peripherie berücksichtigt werden.

Die Entwicklung eines omnipotenten Systems zur Unterwasserkommunikation, das alle auftretenden Effekte kompensiert und allen Anforderungen entspricht, scheint nahezu unmöglich. Dennoch enthalten die vielen unterschiedlichen Kommunikationssysteme [15, 16] grundlegende Funktionen und Eigenschaften, die vermehrt auftreten.

1.4 Wissenschaftlicher Beitrag

Das zentrale Ziel dieser Arbeit ist die Entwicklung eines Konzepts zur Umsetzung, Verifizierung, Optimierung und dem Vergleich von synthetisierbaren Kommunikationssystemen am Beispiel der akustischen Unterwasserkommunikation. Zum Erreichen dieses Ziels werden in dieser Arbeit vier Hauptbereiche behandelt: das Testkonzept, die Detektion von Datenpaketen, die Konzeption, Umsetzung und Verifikation des Kommunikationsmodells sowie die Analyse, Optimierung und Auswahl von Unterwasserkommunikationssystemen.

Für die Bewertung und den Vergleich von akustischen Unterwasserkommunikationssystemen wird ein umfassendes Testkonzept eingeführt. Dieses basiert auf der Definition und Erzeugung eines Analyseraums, der die speziellen Herausforderungen des Unterwasserkanals, wie Rausch- und Ausbreitungseigenschaften, berücksichtigt. Der Analyseraum wird durch die Kombination theoretischer Konzepte der Schallausbreitung basierend auf BELLHOP [20–22] erstellt.

Das Signalrauschen wird mittels eines variablen, Finite Impulse Response (FIR)-Filter-basierten Modells generiert, was die hardware- oder simulationsbasierte Erzeugung eines umgebungsabhängigen und Szenario-basierten Rauschprofils ermöglicht.

Die Definition eines standardisierten Testkonzepts erlaubt einen objektiven Vergleich verschiedener Systeme durch die klare Definition des Analyseraums sowie einheitliche Bewertungskriterien und Maßstäbe. Dies erleichtert die Vergleichbarkeit und Validierung der Systeme und ersetzt teure, aufwendige Feldtests, was sowohl Kosten reduziert als auch Zeit spart.

Durch diesen Ansatz werden die Kontrollpakete verkürzt und die Fehlerquote bei der Decodierung gesenkt. Überdies wird ein Konzept zur Bestimmung der Störanfälligkeit und einer sicheren Strategie eingeführt, das falsche Detektionen vermeidet und gleichzeitig die Transparenz der Informationen gewährleistet.

Es wird gezeigt, dass das eingeführte innovative Kasami-basierte Verfahren im Vergleich zu anderen Präambel-basierten Methoden auch für größere, zukünftige Netzwerke in der akustischen Unterwasserkommunikation geeignet ist und sich durch überlegene Korrelationseigenschaften auszeichnet.

Die modellbasierte und parametrisierbare Abbildung ermöglicht die Umsetzung und den Vergleich der meisten derzeit verfügbaren Systeme. Der eingeführte Ansatz dient als Grundlage für die Entwicklung eines dynamischen Kommunikationssystems, das verifiziert und wahlweise als Simulation oder hardwarebasiert ausgeführt werden kann. Es lässt sich auf Field Programmable Gate Array

(FPGA)s synthetisieren und gegen definierte Modelle sowie Referenzfunktionen testen und vergleichen.

Die definierte Architektur ermöglicht eine flexible Anpassung und Erweiterung des Kommunikationssystems. Durch die Möglichkeit der Synthese des Modells mit numerisch und zeitlich identischem Verhalten wird eine Analyse der Performanz und der Einsetzbarkeit ohne den Einsatz von Trägersystemen ermöglicht. Dadurch werden kostspielige Feldtests und iterative Anpassungen minimiert, und der Bereich der adaptiven Kommunikationssysteme wird umsetzbar.

Durch die eingeführte Methodik zur Entwicklung adaptiver Kommunikationssysteme wird eine Vielzahl von Umsetzungen ermöglicht, die anwendungs- und umgebungsspezifisch unterschieden und ausgewählt werden können. Die daraus resultierende Mehrzieloptimierung wird mit Opt4J [23, 24] und OpenDSE [25] sowie den eingeführten Modellierungs- und Testkonzepten zu einem umfassenden Analyse- und Optimierungsworkflow zusammengeführt.

Ergänzt wird diese Analyse durch Verfahren zur Reduktion des Analyseraums, Beschränkung des Explorationsraums sowie die Nutzung des erstellten hybriden Referenzmodells und des geometrischen Kanalmodells. Dies trägt zu einer erheblichen Reduzierung des zeitlichen Aufwands für die Optimierung bei.

Der methodische Rahmen dieser Arbeit ermöglicht es, die Komplexität des Unterwasserkommunikationssystems zu beherrschen und eine strukturierte Entwicklung zu gewährleisten. Der eingeführte teilautomatisierte Entwicklungsprozess bildet zudem die Grundlage für analytische und leistungsorientierte Vergleiche von Unterwasserkommunikationssystemen und -methoden in Bezug auf deren Anwendung, ortsspezifische Konstellationen und zusätzliche Hardware. Er eröffnet damit neue Möglichkeiten für die Anwendung, Nutzung und Auswahl adaptiver Kommunikationssysteme im Bereich der akustischen Unterwasserkommunikation.

1.5 Aufbau der Dissertation

Die Arbeit umfasst neben der Einführung in Kapitel 1, der Übersicht wesentlicher verwandter Arbeiten in Kapitel 2 und der zusammenfassenden Diskussion in Kapitel 7 fünf wesentliche Beiträge: Das Kapitel 3 beschreibt eine ortsspezifische Definition und Generierung von datenbankbasierten und Zeitvarianten Unterwasserkanälen zur Performanzanalyse, dem Vergleich und der abschließenden Entscheidungsfindung für die Auswahl und den Einsatz von Kommunikationssystemen im Bereich der Unterwasserkommunikation unter der Verwendung

des Programms BELLHOP. Zusätzlich wird eine FIR-Filter-basierte Methode präsentiert, die eine funktionale und zeitlich identische Simulation und Emulation des Unterwasserkanals sowie des dynamischen orts-, zeit- und frequenzabhängigen Unterwasserrauschens zur Offline-Analyse ermöglicht. In Kapitel 4 wird das schichtübergreifende Problem der eindeutigen Detektion und Zuordnung von Datenrahmen bei der drahtlosen Übertragung im Unterwasserbereich aufgezeigt. Dazu wird der Kasami-basierte Ansatz mit einer zusätzlichen Codierung für die Identität und den Rahmentyp mittels eindeutiger orthogonaler Folgen für eine verbesserte Detektion und Zuordnung eingeführt.

Kapitel 5 führt die Methodik des Varianten-basierten und modularen Entwurfs eines synthetisierbaren Kommunikationssystems mittels eines formalisierten Modells ein. Basierend auf den Freiheitsgraden des Modells wird in Kapitel 6 die modell- und kanalbezogene Anbindung einer Mehrzieloptimierung unter Verwendung von Opt4J und OpenDSE umgesetzt und diskutiert.

Um den entstehenden Optimierungsaufwand zu reduzieren, werden zudem kombinierbare Reduktionsverfahren beschrieben, die mittels geometrischer oder analytischer Betrachtung den Analyseraum abschätzen und über die stochastische Verteilung von Testpunkten den Aufwand zur Optimierung reduzieren. Erweitert wird die Reduktion des Analyseraums durch die Beschränkung des Entwurfsraums durch den Ausschluss von Kommunikationssystemen, basierend auf definierten und berechneten Schranken. Abschließend wird in diesem Kapitel für spezifische Umgebungen die komplette Analyse, Reduktion, Beschränkung und Optimierung für den statischen und dynamischen Einsatz unter Verwendung aller eingeführten Ansätze und Konzepte in dieser Dissertation bewertet und diskutiert.

2

Stand der Forschung zur Unterwasserkommunikation

Ein entscheidender Faktor in der Unterwasserkommunikation ist die effiziente Übertragung von Informationen in einer äußerst anspruchsvollen Umgebung, dem Unterwasserkanal. In den vergangenen Jahren haben Forscher sowohl im wissenschaftlichen Bereich als auch in der Industrie intensiv an der Entwicklung und Verbesserung von Kommunikationssystemen gearbeitet, die speziell für Unterwasserumgebungen optimiert sind. Dieses Kapitel beinhaltet einen zusammenfassenden Überblick über relevante Arbeiten im Bereich der Unterwasserkommunikation. Dabei liegt der Schwerpunkt auf der Entwicklung und technischen Umsetzung von Kommunikationssystemen.

2.1 Aktuelle Entwicklungen

Die Herausforderungen des Unterwasserkanals und die zunehmende Anzahl von Anwendungsmöglichkeiten für die Unterwasserkommunikation haben zu einem dynamischen Wachstum im Markt und einem erhöhten Fokus in der Forschung geführt. Die Tabellen 2.1 bis 2.4 stellen eine aktuelle Auswahl von kommerziellen und wissenschaftlichen Unterwasserkommunikationssystemen, mit dem Schwerpunkt der akustischen Signalübertragung, dar.

Diese Tabellen ermöglichen einen vergleichenden Überblick über die grundlegenden Eigenschaften der Systeme, soweit diese verfügbar sind, wie Modulation, Trägerfrequenz, maximale Bandbreite und Datenrate, Energiebedarf für das Senden, Empfangen und Warten sowie den Ruhemodus. Des Weiteren ist die maximale Reichweite R^+ des Systems unter idealen Kommunikationsbedingungen angegeben, wobei die Kommunikationsreichweite R in dieser Arbeit in kurz ($R \leq 500\,\mathrm{m}$), mittel ($500\,\mathrm{m} \leq R \leq 3500\,\mathrm{m}$) und lang ($R > 3500\,\mathrm{m}$) unterteilt wird.

Die maximale Reichweite wird in der Unterwasserkommunikation unter optimalen Bedingungen bestimmt, welche eine Signalausbreitung frei von Mehrwegausbrei-

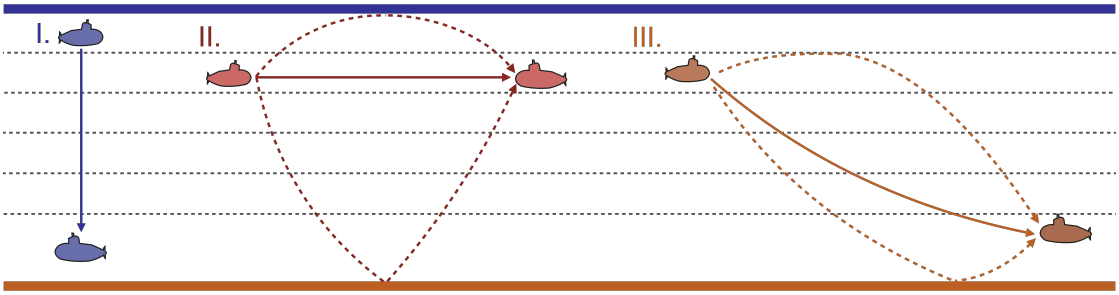


Abbildung 2.1: Schematische Abbildung der vertikalen, horizontalen und gemischten Signalausbreitung

tung, Reflexionen und Brechungen ermöglichen. Diese Bedingungen werden im Ansatz nur bei vertikaler Kommunikation zwischen Schiffen (Bojen) und autonomen Unterwasserfahrzeugen erfüllt, wie Studien von Shengjun u. a. in [26] zeigen. Dies resultiert aus der Kommunikation mit einer Signalausbreitung senkrecht zu den Temperaturschichten (Abbildung 2.1 I.), wodurch es kaum zu Brechungen, Beugungen, Transmissionen und Interaktionen mit den natürlichen Grenzen des Wassers kommt. Während die horizontale Kommunikation (Abbildung 2.1 II.), je nach Distanz der Kommunikationssysteme, der Wassertemperatur und weiterer Umgebungsparameter Brechungen und Reflexionen unterliegt, ist eine Kommunikation innerhalb einer Temperaturschicht über mehrere Kilometer und unter optimalen Bedingungen sogar über mehrere hundert Kilometer möglich, wie Freitag u. a. in einem Experiment in der Arktis gezeigt haben [27].

Die herausfordernde ist aber die vertikal und horizontal gemischte Kommunikation (Abbildung 2.1 III.), da die Signalausbreitung aufgrund der unterschiedlichen Tiefen von Sender und Empfänger über vorhandene Temperaturschichten hinweg mit unterschiedlichen Ausbreitungsgeschwindigkeiten erfolgt. Durch die Kommunikation über die Grenzen der Temperaturschichten hinweg kommt es, abhängig vom Winkel der sich ausbreitenden Schallwellen zur Temperaturschicht und dem Unterschied in der Ausbreitungsgeschwindigkeit, zu Reflexionen, Brechungen, Beugung und Verzerrungen der Schallwelle und damit dem gesendeten Signal.

Bei der Auswahl eines Kommunikationssystems aus den kommerziellen und wissenschaftlich verfügbaren Systemen sind je nach Anwendungs- und Einsatzbereich die Eigenschaften wie Zuverlässigkeit, Datenrate und Kommunikationsreichweite grundlegende Entscheidungskriterien. Aufgrund der geringen Bandbreite, der teilweise hohen Distanzen und des herausfordernden Unterwasserkanals sind auch Parameter wie Zuverlässigkeit, Robustheit und Energiebedarf, insbesondere bei autonomen Systemen mit beschränkter Kapazität, von entscheidender Bedeutung. Die Robustheit bezieht sich dabei auf die Fähigkeit,

auftretende Störungen zu kompensieren oder deren negativen Einfluss auf die Kommunikation zu minimieren, während die Zuverlässigkeit sich auf die valide Detektion von Signalen bezieht.

Die Vergleichbarkeit der Modems gestaltet sich aber schwierig, da Entwickler und Hersteller ihre Parameter uneinheitlich angeben und direkte Vergleiche maximaler Performanz ohne Kontext nur eine geringe Aussagekraft besitzen. Beispielsweise gibt Blueprint [28] die maximale Datenrate als Nutzdatenrate an, was zu vermeintlich niedrigeren Datenraten im Vergleich führt. Zudem ist die Datenrate generell abhängig von zusätzlich benötigten Daten für den Verbindungsaufbau, der Latenz, dem Kommunikationsprotokoll und der Kapazität in Abhängigkeit von Frequenz, Bandbreite und Ausrichtung der Kommunikation.

Die zusammengefassten Kommunikationssysteme in den Tabellen 2.1 bis 2.4 spezifizieren jedoch nur Modulation, Bandbreite, Trägerfrequenz, maximale Datenrate und Reichweite. Abbildung 2.2 zeigt hierzu die Relation zwischen Reichweite und Datenrate, und in Abbildung 2.3 wird die spektrale Effizienz, basierend auf Bandbreite und Datenrate, grafisch verglichen.

Die spektrale Effizienz, die obere Schranke der erreichbaren Datenrate in $\mathrm{bit\,s^{-1}}$ in Bezug auf die Bandbreite, normiert auf $1\,\mathrm{Hz}$, ist zusätzlich in Abbildung 2.3 für Frequenz-Umtastung (FSK)-, Binary Phase Shift Keying (BPSK)- und Quadraturphasenumtastung (QPSK)-Modulation abgebildet. Hierzu gibt der Hersteller Sercel (#29,#30) als Beispiel die Modems mit einer maximalen Datenrate von $7.4\,\mathrm{kbit/s}$ und $24.6\,\mathrm{kbit/s}$ für eine durch den Frequenzbereich implizierte Bandbreite von $5\,\mathrm{kHz}$ und $9\,\mathrm{kHz}$, basierend auf dem angegebenen nutzbaren Frequenzbereich. Dies entspräche aber einer spektralen Effizienz von $1.5\,\mathrm{bit/s/Hz}$ und $2.74\,\mathrm{bit/s/Hz}$. Daher liegt hier die Vermutung nahe, dass die nicht direkt angegebene Bandbreite viel größer sein muss. Eine fehlende Information, die für die Wahl des Transducers und den Einsatzbereich von enormer Bedeutung ist.

Die Auswertung der Tabellen zeigt, dass kommerzielle Modems sich auf die Hochgeschwindigkeitsdatenübertragung bei mittleren Distanzen konzentrieren, während sich forschungsorientierte Modems auf spezifische Aspekte des Systems oder der Übertragung fokussieren. Der Großteil der wissenschaftlichen Systeme ist nicht einsatzfähig, mit Ausnahme der Kommunikationssysteme wie Hermes [49], dem MIT-Prototypen [50] und μ-Modem 2 [80].

In der Forschung konnten im Bereich der Hochgeschwindigkeitsdatenübertragung bereits beeindruckende Fortschritte erzielt werden: So ermöglicht das Hermes Florida Atlantic University (FAU) Akustik-Modem [49] eine Übertragung von Bildern mit bis zu $150\,\mathrm{kbit/s}$ über eine kurze Distanz von $120\,\mathrm{m}$ und der

Tabelle 2.1: Eigenschaften ausgewählter Unterwassermodems (1. Teil, Akustik)

	Hersteller	Modell	Hard-ware	Modulation	Träger-frequenz kHz	Band-breite kHz	Daten-rate kbit/s	Energiebedarf Tx W	Rx mW	SB mW	SL mW	Reich-weite km
1	Teledyne [15]	UCM-900	DSP	PSK[1]	25.0	10	15.36	10	360	–	3.5	2
2			DSP	MFSK	25.0	10	2.4	10	360	–	3.5	2
3			DSP	FH-FSK[2]	25.0	10	0.08	10	360	–	3.5	2
4		CM-900	DSP	PSK[1]	11.5	5	15.36	5	445	–	3.5	4
5			DSP	MFSK[2]	11.5	5	2.4	5	445	–	3.5	4
6			DSP	FH-FSK[2]	11.5	5	0.08	5	445	–	3.5	4
7		ATM-9xx	DSP	PSK[1,3]	11.5	5	15.36	60	1400	–	12	6
8			DSP	MFSK[3]	11.5	5	2.4	60	600	–	12	6
9			DSP	FH-FSK[3]	11.5	5	0.08	60	600	–	12	6
10	LinkQuest[29]	UWM1000 [30]	DSP	BASS	35.7	17.9	17.8	2	750	–	8	0.35
11		UWM2000 [31]	DSP	BASS	35.7	17.9	17.8	8	800	–	8	1.5
12		UWM2200 [32]	DSP	BASS	71.4	17.9	35.7	6	1	–	8	1
13		UWM2000H [33]	DSP	BASS	35.7	17.9	17.8	8	800	–	12	1.5
14		UWM3000 [34]	DSP	BASS	10.0	17.9	5	12	800	–	8	3
15		UWM3000H [35]	DSP	BASS	10.0	17.9	5	12	800	–	8	3
16		UWM4000 [36]	DSP	BASS	17.0	17.9	8.5	7	800	–	8	4
17		UWM10000 [37]	DSP	BASS	35.7	10	5	40	900	–	9	10
18	Sonardyne	M6 SUB-Mini [38]	DSP	QPSK	26.8	11.5	9	–	–	50	–	4
19		M6 Mini [39]	DSP	QPSK	26.8	11.5	9	50	1000	1000	650	3
20		M6 Standard [40]	DSP	QPSK	26.8	11.5	9	–	–	50	–	5
21	Develogic [41]	HAM.Base	DSP	mDPSK	52.5	25	10	250	–	10	–	1.2
22			DSP	MFSK	52.5	25	0.145	250	–	10	–	1.2
23		HAM.Node	DSP	mDPSK[4]	3.5	2	7	250	–	30	3	1.95
24			DSP	n-MFSK[4]	3.5	2	0.145	250	–	30	3	30
25	EvoLogics [18]	S2C R12/24	DSP, FPGA	S2C	18.5	12	9.2	57	800	285	2.5	6
26		S2C R15/27	DSP, FPGA	S2C	21	12	9.2	65	800	285	2.5	6
27		S2C R18/34	DSP, FPGA	S2C	26	12	13.9	65	800	285	2.5	3.5
28		S2C M HS	DSP, FPGA	S2C	150	60	62.5	8.5	800	285	0.5	0.3
29	Sercel [42]	MATS3G 12 KHZ	DSP	FSK/PSK	12	5	7.4	75	600		40	15
30		MATS3G 34 KHZ	DSP	FSK/PSK	34	9	24.6	75	600		40	5
31	DiveNET [43]	Sealink M	DSP	BPSK	20	20	0.078	6	330	–	–	1
32		Sealink S	DSP	BPSK	10	10	0.08	10	400	–	–	8
33		Sealink C	DSP	BPSK	10	10	0.088	–	–	–	–	8
34	Blueprint [28]	X150	DSP	SS	28	8	0.1	6	–	600	–	1
35		X110	DSP	SS	28	8	0.1	6	–	600	–	1
36		X010	DSP	SS	28	8	0.1	6	–	600	–	1
37	WHOI [19]	μModem 2	DSP, FPGA	PSK[5]	10	5	5.4	150	2550	630	1.8	–
38			DSP, FPGA	FH-FSK[5]	10	4	0.08	150	630	630	1.8	–
39		Arktis[27]	DSP, FPGA	FSK	0.9	0.1	0.001	50				400
40	Subnero [44]	M25M [45]	SDR, DSP	OFDM	24	12	15	80	4000	–	1.5	5
41			SDR, DSP	FH-FSK	24	12	15	80	4000	–	1.5	5
42	AppliCon [46]	SEAMODEM	DSP	MFSK[1]	30	10	2.25	40	–	–	–	0.4
43	L3HARRIS [47]	GPM300	DSP	DSSS	11.5	10	1	300	1800	–	5	45
44	TriTech [48]	Micron Modem	DSP	DSSS	24	8	0.04	7.92	720	–	–	0.5
45	FAU [49]	Hermes	DSP	MPSK	360	26	150	32	–	–	–	0.12

[1] Nur Senden
[2] Optional: Trägerfrequenz: (i) 18.5 und (ii) 24.5 kHz mit der maximalen Reichweite: (i) 2 und (ii) 1 km
[3] Optional: Trägerfrequenz: (i) 18.5 und (ii) 24.5 kHz mit der maximalen Reichweite: (i) 4 und (ii) 2 km
[4] Optional: Trägerfrequenz: (i) 10, (ii) 15.5 und (iii) 23 kHz mit der Bandbreite: (i) 5, (ii) 9 und (iii) 12 kHz
[5] Optional: (i) 7, (ii) 10 und (iii) 25 kHz mit dem Energiebedarf beim Senden von $\approx$ (i) 150, (ii) 35 und (iii) 60 W

Tabelle 2.2: Eigenschaften ausgewählter Unterwassermodems (2. Teil, Akustik)

Hersteller	Modell	Hard-ware	Modulation	Träger-frequenz kHz	Band-breite kHz	Daten-rate kbit/s	Energiebedarf Tx W	Rx mW	SB mW	SL mW	Reich-weite km
46 MIT [50]	Prototype	DSP	QAM	75	15	150	–	–	–	–	0.01
47 Sanchez [51]	ITACA	MCU	FSK	0.32		1	0.012	24	–	0.003	0.02
48 Benson [52]	Benson	FPGA	FSK	35	10	0.08	0.5	500	–	–	0.02
49 Nowsheen [53]	Nowsheen	FPGA	BPSK	800	80	80	–	–	–	–	0.015
50 Sadeghi [54]	Sadeghi	SDR, USRP	OFDM	30	4	–	–	–	–	–	0.002
51 Coralcon [55]	Coralcon	SDR	FSK	19	2	1	–	–	–	–	0.09
52 Peng [56]	Peng	DSP	QPSK	12	2	0.77	–	–	–	–	0.1
53 Kongsberg[57]	cNODE 34-180	DSP	FSK	26	10	6	100	–	100	–	1
54	cNODE 34-40V	DSP	FSK	26	10	6	100	–	100	–	4
55 Jinfeng[58]	VHF-UAM		PSK	115	30	23	50	–	–	–	0.0076
56 Indriyanto[59]	UAM	MCU	FSK	41.5	3	1.2	–	–	–	–	0.0017
57 Demirors [60]	SDAM	SDR	OFDM	100	24	10	–	–	–	–	0.2
58			DSSS	125	97.6	4	–	–	–	–	0.2
59 Dong[61]	UMAM		FSK	12.5	3	0.208	–	–	–	–	0.100
60			QPSK	12.5	3	1.23	–	–	–	–	0.010
61			QAM	12.5	3	3.73	–	–	–	–	–
62 Zia[62]	UAM		FSK	31.5	2	0.3	–	–	–	–	0.001
63 Sheikh[63]	UAM	SDR	FSK	19	2	1	3	–	–	–	0.09
64 Zhang[64]	MODEM	DSP	QPSK	11.5	5	1.2	20	1200	–	20	–
65		DSP	SS	11.5	5	0.5	20	1200	–	20	–
66 Zhao[65]	UWNN	DSP	LFM	23	5	6	–	–	–	–	0.002
67 Naidu[66]	WUAM	Laptop	OFDM	15.15	6.3	–	50	–	–	–	0.002
58 Zhou [67]	Zhou	DSP	OFDM-QPSK	16	4	5.2	–	–	–	–	0.001

am Massachusetts-Institut für Technologie (MIT) entwickelte Prototyp eines Modems [50] erreichte unter optimalen Bedingungen $150\,\mathrm{kbit/s}$ allerdings nur für eine Distanz von $10\,\mathrm{m}$.

Der Fokus liegt aber nicht nur auf der Hochgeschwindigkeitsdatenübertragung, sondern auch auf der robusten Lang- und Mittelstreckenkommunikation mit entsprechend niedrigen Bitraten Diese Art der Kommunikation ist ausreichend, um geringe Datenmengen wie Statusaktualisierungen, Positionen und andere wesentliche Informationen zu übertragen. Beispielsweise bietet die ATM-9xx-Serie von Teledyne [15] (siehe Tabelle 2.1) eine Datenrate von $2.4\,\mathrm{kbit/s}$ über Mehrfrequenz-Umtastung (MFSK) und nur $80\,\mathrm{bit/s}$ über das robuster ausgelegte Frequenzsprungverfahren kombiniert mit Frequenzumtastung (FH-FSK) bei einer maximalen Reichweite von $6\,\mathrm{km}$. Die ATM-9xx unterstützen jedoch gleichzeitig über Phasenumtastung (PSK) auch eine schnelle Kommunikation von bis zu $15.36\,\mathrm{kbit/s}$, womit sich der zweigeteilte Fokus auch kommerziell widerspiegelt.

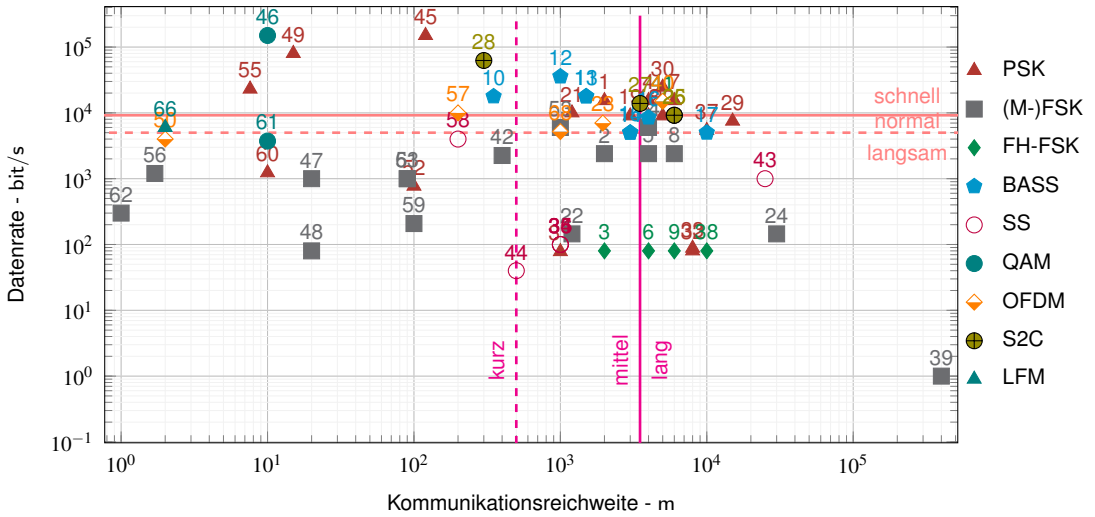


Abbildung 2.2: Datenrate in Relation zur Reichweite (Tabellen 2.1 and 2.2)

Die Blueprint-X-Serie [28] überträgt bei einer maximalen Reichweite von $1\,\mathrm{km}$ bis zu $100\,\mathrm{bit/s}$. Das HAM.NODE von Develogics [41] kann bis zu $30\,\mathrm{km}$ mit einer Bitrate von $145\,\mathrm{bit/s}$ übertragen, und eine Variante des WHOI-MicroModems [81] erreicht sogar bis zu $400\,\mathrm{km}$ bei $1\,\mathrm{bit/s}$ [27], allerdings nur unter arktischen Bedingungen.

Ein Teil der Forschungs- und Entwicklungsaktivitäten konzentriert sich aber auch darauf, Kompromisse zwischen hohen Datenraten und der Distanz zu erreichen. Beispiele hierfür sind das S2CR 18/34-Modem von Evologic [18] mit $13.9\,\mathrm{kbit/s}$ bei bis zu $3.5\,\mathrm{km}$ oder das Subnero-Modem [44] mit $15\,\mathrm{kbit/s}$ auf maximal $3\,\mathrm{km}$ Entfernung.

Mit dem Fokus auf den Aufbau von Unterwassernetzwerken, der Erforschung der Tiefsee, die Nutzung von Frühwarnsystemen, der Populationskontrolle, die Überwachung von Pipelines, Aquakulturen und weiteren Anwendungsgebieten steigt der Bedarf an höheren Datenraten aber stetig weiter an. Dies wirft zugleich das Problem der Mehrsystemkommunikation in Unterwassernetzwerken mit unterschiedlichen Ausrichtungen und Schwerpunkten auf. Der gesteigerte

Tabelle 2.3: Eigenschaften der Unterwassermodems (elektromagnetisch)

Name	Modell	Modulation	Träger-frequenz kHz	Band-breite kHz	Maximale Datenrate kbit/s	Energiebedarf Tx W	Rx mW	SB mW	Sleep mW	Maximale Reichweite m
Dalhousie Univ. Prototype [68]	MIM	QPSK	22.5	4.4	4	–	–	–	–	1.5
Clayton[6, 7]	UTM	PSK/FSK	225.0	–	–	–	–	–	–	0.010

Tabelle 2.4: Eigenschaften ausgewählter Unterwassermodems (optisch)

Name	Modell	Modulation	Licht-welle kHz	Maximale Datenrate Mbit/s	Energiebedarf Tx W	Rx mW	SB mW	Sleep mW	Maximale Reichweite m
Aquatec AQUAmodem	Op2 [69]	–	Cyan	10.2	–	–	–	–	1
	Op2L [70]	–	Cyan	10.2	–	–	–	–	1
Hydromea [71]	Luma X [72]	–	–	10.0	17	2000	2000	–	50
	Luma500ER [73]	–	–	500.0	5	1000	2000	–	50
	Luma 250LP [74]	–	–	250.0	5	500	500	3	7
	Luma 100 [75]	–	–	100.0	2	500	500	3	2
Medusa	Optical Modem[76]	4b5b,	527	0.2	24	–	–	–	10
Sonardyne[77]	BlueComm 100[78]	–	450	5.0	6	10	–	–	15
	BlueComm 200[79]	–	405	10	10	6.0	–	–	150
SPAWAR OM	[13]	–	10000	–	–	–	–	–	2
Sant'Anna	OptoCOMM[16]	–	470	10	–	–	–	–	7

Bedarf resultiert aus der Kommunikation über Knoten, der Fernkontrolle und Fernsteuerung von Unterwassersystemen durch Menschen oder Systeme sowie dem Bedarf an Echtzeitvideoübertragung und Echtzeitkommunikation. BaltRobotics [82] demonstrierte in diesem Kontext ein System für Video-Streaming mit $15\,\mathrm{fps}$ in geringer Qualität ($640 \times 480\,\mathrm{Pixel}$) und einer Datenrate von $128\,\mathrm{kbit/s}$, bei dem Entfernungen von bis zu $200\,\mathrm{m}$ überwunden werden konnten. EvoLogics hat das S2CM HS [18] entwickelt, das in diesem Bereich Datenraten von bis zu $63\,\mathrm{kbit/s}$ bei maximaler Entfernung von $300\,\mathrm{m}$ ermöglicht.

Der Aufbau von Unterwassernetzwerken ermöglicht auch die Kommunikation über feststehende Knoten an der Wasseroberfläche mit der Außenwelt über hohe Datenraten mittels elektromagnetischer Übertragung. Um die Daten der autonomen Unterwasserfahrzeuge Autonomes Unterwasserfahrzeug (AUV) weiterzuleiten, müssen diese jedoch entsprechend angebunden werden. Dies hat die Entwicklung kontaktloser und datenintensiver Ultrakurzstreckenübertragung mittels Licht oder gerichteter Laser etabliert. Beispiele für diese Form der Unterwasserkommunikation, die nicht im Fokus dieser Arbeit stehen, sind in Tabelle 2.4 abgebildet. Ein alternativer Ansatz wurde von Dalhousie verfolgt [68], der eine Kommunikation abhängig von der Anwendung entweder akustisch, optisch oder elektromagnetisch durchführt.

Die Auswertung der gelisteten Systeme zeigt zudem, dass die zur Verfügung gestellten Daten für eine umfassende und vergleichende Analyse der Kommunikationssysteme bezüglich ihrer Robustheit und ihrer Eignung für eine spezifische Anwendung mit definierten Umgebungsparametern nicht ausreichen. Es fehlen Informationen darüber, wie zuverlässig die Modems in definierten Umgebungen und unter spezifischen Bedingungen sind. Dies könnte Faktoren wie Störanfälligkeit und die Anfälligkeit für Interferenzen und Fehlerraten allgemein umfassen.

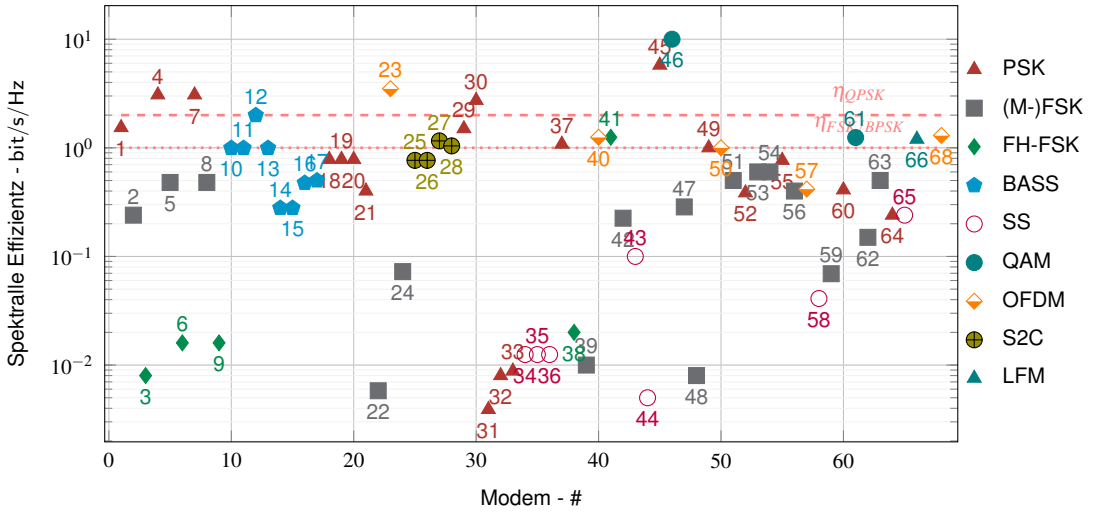


Abbildung 2.3: Bandbreiteneffizienz der Modems (Tabellen 2.1 and 2.2)

Es fehlen zudem größtenteils Informationen über die Kompatibilität mit Systemen und Kommunikationsprotokollen und Aussagen über die Flexibilität bzw. die Fähigkeit der Modems, sich an ändernde Anforderungen und Umgebungen anzupassen. Woraus sich für einen Vergleich der Systeme der Bedarf einer standardisierten Bewertung und Testung ergibt, welche definierte Zielgrößen in Abhängigkeit der Störungen und der Umgebungsspezifikationen abbilden können.

2.2 Modulationsverfahren

Die Abbildung 2.4 zeigt einen Überblick über die angewandten Modulaktionstechniken aus den Tabellen 2.1 bis 2.4. Dabei werden die verfügbaren Modulationen der Modems getrennt aus dem kommerziellen Bereich und der Forschung betrachtet. In der Forschung und bei kommerziell entwickelten Modems sind primär Modulationen wie FSK mit FH-FSK als Subdomäne sowie verschiedene Formen von PSK durchgängig präsent. Allerdings stechen LinkQuest [29] mit Broadband Acoustic Spread Spectrum (BASS) [83] und Evologics [18] mit Sweep Spread Carrier (S2C) [84] aus der Masse hervor. Diese patentierten Lösungen basieren auf Sweep-Carrier-Verfahren [85] und sind bisher nicht öffentlich zugänglich. Die teilweise verwendete Mehrträgermodulation in Form von Orthogonales Frequenzmultiplexverfahren (OFDM) kann mit allen anderen Modulationen kombiniert werden und stellt eine Erweiterung dar, die hauptsächlich, basierend auf den verfügbaren Daten, in Kombination mit QPSK eingesetzt wird. Ein System, das

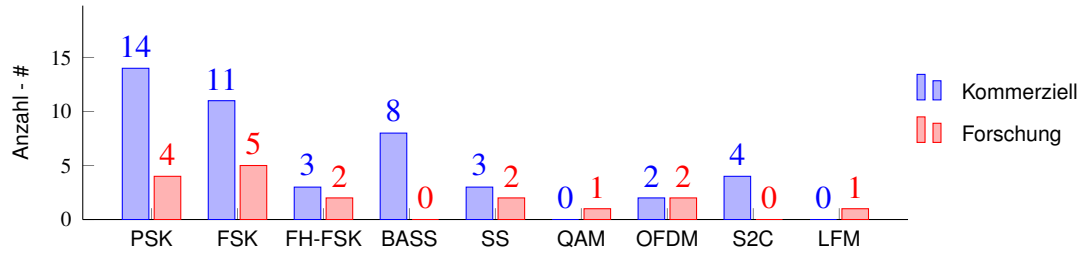


Abbildung 2.4: Verteilung der Modulationsverfahren (Tabellen 2.1 and 2.2)

potenziell PSK, FSK und FH-FSK unterstützt, könnte damit 66 % der in dieser Arbeit aufgezeigten akustischen Kommunikationssysteme abbilden.

2.3 Modellbasierte Entwicklung

Die modellbasierte Entwicklung nimmt heutzutage eine wichtige Rolle bei der Entwicklung von Technologien, speziell Kommunikationstechnologien, ein. Der Hintergrund dafür sind die zunehmende Komplexität und die stetig wachsenden Anforderungen an Leistung, Zuverlässigkeit und Sicherheit von Systemen. Der Einsatz moderner Modellierungs- und Simulationsmethoden, wie dem modellbasierten Entwicklungsansatz, ermöglicht nicht nur eine effiziente Entwicklung, sondern auch eine mit den Anforderungen mitwachsende und skalierbare Analyse und Optimierung von Systemen. Durch die damit frühzeitige Analyse, Optimierung und die Verschiebung der Designentscheidung können somit Fehler vermieden, Kosten durch fehlerhafte Prototypen reduziert und die Entwicklungszeit erheblich gekürzt werden.

2.3.1 Allgemein

So haben Gandhi u. a. [86] Modellierungstechniken und Strategien für die simulationsbasierte Entwicklung von Kommunikationssystemen und den resultierenden Nutzen für eine praktische Umsetzung zusammengefasst. Die Autoren kommen zu dem Schluss, dass die richtige problemorientierte Wahl von genutzten Techniken und Strategien für ein zuverlässiges Kommunikationssystem und dessen Verbesserung der Leistung und Zuverlässigkeit ist. Zudem sollten Ansätze die Integration von modernen Ansätzen ermöglichen.

Ein Beispiel dafür im Kontext der modellbasierten Entwicklung oder spezifischer der Testung von 5G-Kommunikationssystemen haben Lin und Lee [87] veröf-

fentlicht. Die beiden Autoren untersuchten dazu modellbasierte Testtechniken für die 5G-Kommunikation und fassten den Beitrag der modellbasierten Testentwicklung in Bezug auf Qualität und Sicherheit zusammen, indem komplexe Szenarien simuliert und die Kommunikationssysteme auf ihre Funktionalität überprüft werden. Sie zeigen damit am Beispiel von 5G, dass modellbasierte Ansätze eine robuste Methode zur Sicherstellung der Qualität und Zuverlässigkeit von Kommunikationssystemen bieten können. Allerdings wird auch deutlich, dass die Herausforderungen in der Skalierung und Anpassungsfähigkeit der Modelle liegen, um die Komplexität der Netzwerke abzubilden.

Die Autoren Yang und Chen analysieren in [88] modellgetriebene Methoden für die Entwicklung von IoT-Systemen und diskutieren die Herausforderungen, Lösungsansätze sowie die zukünftige Entwicklungsrichtung in diesem Bereich. Auch hier zielen die analysierten Entwicklungen darauf ab, die Komplexität und Skalierbarkeit der zu entwickelnden Systeme zu bewältigen und eine effiziente Verwaltung der Infrastruktur durch präzise Modellierung zu ermöglichen. Yang und Chen kommen zu dem Schluss, dass die modellgetriebenen Ansätze eine entscheidende Rolle in der Entwicklung und Verwaltung von IoT-Systemen spielen. Wie auch in den vorherigen Veröffentlichungen liegen die wesentlichen Herausforderungen, ihrer Meinung nach, in der Komplexität und Skalierbarkeit solcher Systeme, aber zusätzlich auch in der Integration heterogener Technologien.

Diese und andere aktuelle Beiträge zeigen, dass durch den systematischen Einsatz von Modellierung, Simulation und Analyse komplexe Herausforderungen abgebildet und innovative Lösungen entwickelt werden können. Damit steigert sich nicht nur die Entwicklungseffizienz, sondern es kann auch die Qualität und Sicherheit von Kommunikationssystemen verbessert werden.

2.3.2 Unterwasserkommunikation

Die Autoren Freitag, Stojanovic und Proakis haben im Laufe der vergangenen Jahre den Bereich der akustischen Unterwasserkommunikation, Kanalmodellierung und Beschreibung von Simulation, insbesondere im Bereich der OFDM-Kommunikation Unterwasser, maßgeblich mitgeprägt. So hat Stojanovic in [89] einen Überblick über mögliche Kanalmodelle der akustischen Informationsübertragung unter Wasser gegeben und geht auf deren formale Beschreibung sowie deren Vor- und Nachteile ein. Sie kommt zu dem Schluss, dass die Kenntnis und Anwendung geeigneter ortsspezifischer Kanalmodelle entscheidend für die

effiziente Entwicklung von Kommunikationssystemen im Bereich der akustischen Unterwasserkommunikation ist.

In Zusammenarbeit mit Song u. a. in [90] fassen Stojanovic und Song die Probleme und Herausforderungen der Unterwasserkommunikation allgemeiner zusammen. So arbeiteten Sie heraus, dass die Mehrheit der aktuellen Implementierungen, seien es akademische Prototypen oder kommerzielle Produkte, wie bereits eingeführt, auf proprietären Technologien basieren und keine Interoperabilität unterstützen. Sie stellen zudem fest, dass nur wenig Aufwand getrieben wurde, um Verfahrensstandards zu etablieren oder die Interoperabilität der Implementierungsergebnisse sicherzustellen. Der Implementierungsprozess müsste nach Song und Stojanovic typischerweise vier Phasen umfassen: die Spezifikationsbildung, basierend auf Missionsanforderungen und Standards, die Entwicklung von Software und Hardware, Test und Bewertung sowie Schulung und Dokumentation. Sie zeigen auch auf, dass Kommunikationstechniken und Algorithmen, die früher als zu komplex für die Umsetzung galten, nun, insbesondere auf Field Programmable Gate Array (FPGA)s, umsetzbar sind und erneut überprüft werden müssen. Damit spiegeln die Autoren den derzeitigen, kaum vorhandenen Stand der modellbasierten Entwicklung im Bereich der Unterwasserkommunikation wider.

Es gibt zwar theoretische Arbeiten und Ansätze, die aber kaum oder nur selektiv umgesetzt wurden. Zudem fehlen Standards und die Möglichkeit, Systeme effizient, objektiv und missionsspezifisch zu analysieren und zu vergleichen. Aber es gibt Bemühungen, einen gemeinsamen Standard in Form von Joint Acoustic Network Using Standard (JANUS) [91, 92], mit Schwerpunkt auf dem Unterwasserkommunikationsprotokoll zu etablieren.

2.4 Kommunikationsstandard

Der JANUS-Standard ist ein offener, international anerkannter Standard [91] für die akustische Unterwasserkommunikation. Er wurde von der North Atlantic Treaty Organization (NATO) entwickelt und im Jahr 2017 veröffentlicht und soll primär als universelles Protokoll, das verschiedene Unterwasserkommunikationssysteme miteinander verbindet und interoperabel macht, dienen. JANUS definiert dazu ein grundlegendes Paketformat der Länge $64\,\mathrm{bit}$ und beschränkt die Modulation auf Frequency-Hopping Spread Spectrum-Modulation (FH-SS-Mod) im Bereich von $9\,\mathrm{kHz}$ bis $14\,\mathrm{kHz}$ für die akustische Kommunikation im Unterwasserbereich und soll die Robustheit gegenüber Störungen und Mehrweg-Ausbreitungen ver-

bessern. Die Tabelle 2.1 zeigen aber, dass dieser Frequenzbereich und das Modulationsverfahren kaum genutzt oder unterstützt werden.

Den größten Vorteil von JANUS stellt in der Theorie die Fähigkeit dar, Geräte unterschiedlicher Hersteller und mit unterschiedlichen technischen Spezifikationen miteinander kommunizieren zu lassen. Dies fördert die Integration und den Datenaustausch in multinationalen Operationen und bei zivilen Anwendungen, wird aber derzeit kaum umgesetzt. Obwohl JANUS ein offener Standard ist, ist seine weltweite Akzeptanz und Implementierung derzeit nicht zu erkennen. Die akustische Kommunikation im Unterwasserbereich hat aufgrund der physikalischen Eigenschaften des Mediums eine begrenzte Reichweite. Auch JANUS ist davon betroffen und kann nur in relativ kurzen Distanzen effektiv eingesetzt werden, was die Kommunikationsfähigkeit in großflächigen Unterwasseroperationen einschränkt. Zudem ist JANUS darauf ausgelegt, eine universelle Basis für die Kommunikation zu bieten, was bedeutet, dass es in komplexen, spezialisierten Anwendungen und spezifischen Umgebungen nicht die erforderliche Flexibilität oder versprochene Robustheit bietet.

Der JANUS-Standard stellt damit einen bedeutenden Schritt zur Verbesserung der Unterwasserkommunikation dar, hauptsächlich durch die Förderung der Interoperabilität und der einfachen Integration in verschiedene Systeme. Seine Begrenzungen hinsichtlich Bandbreite, Reichweite und Datenrate schränken jedoch seine Anwendungsmöglichkeiten in komplexen und datenintensiven Szenarien ein. Trotz dieser Nachteile bietet JANUS durch seine offene Struktur und die Unterstützung durch internationale Organisationen wie die NATO eine erste Grundlage für die Weiterentwicklung und Standardisierung der Unterwasserkommunikation.

2.5 Unterwasserkanalsimulation

Für eine umfassende Auswertung und den Vergleich von Unterwasserkommunikationssystemen ist neben dem System selbst auch ein Umgebungsmodell des Unterwasserkanals von entscheidender Bedeutung. Der Unterwasserkanal mit seinem ortsspezifischen, frequenzabhängigen und Temperatur-sensitiven Verhalten stellt eine anspruchsvolle und sich ständig verändernde Herausforderung in der akustischen Unterwasserkommunikation dar. Die Signale, die von einem Kommunikationssystem empfangen werden sollen, unterliegen den Charakteristiken des Kanals und den daraus resultierenden, zeitvarianten Impulsantworten. Eine gängige Methode zur Analyse von Leistung und einem einfachen Vergleich zwischen Kommunikationssystemen besteht darin, vereinfachte Kanäle

zu verwenden, die beispielsweise auf dem additives weißes gaußsches Rauschen (AWGN) basieren und durch das Signal-Rausch-Verhältnis (SNR) definiert sind [93–95]. Jedoch reicht dieser vereinfachte Kanal, der hauptsächlich das thermische Rauschen berücksichtigt, nicht aus, um die Anwendbarkeit von Systemen im realistischen Unterwasserbereich zu analysieren.

Der additives weißes gaußsches Rauschen (AWGN)-Kanal kann nur begrenzte Informationen über die Leistung von Systemen in realen Unterwasserkanälen geben. Ein weitverbreitetes Modell zur Kanalmodellierung in der Unterwasserkanalsimulation ist BELLHOP [21, 22, 96], ein Beam/Raytracing-Modell zur Vorhersage der Schallausbreitung. Ursprünglich von M. Porter entwickelt und wird mit Stand 2024 von der Woods Hole Oceanographic Institution (WHOI) gepflegt. Dieses Simulationsmodell basiert auf der Strahlentheorie, welche die Ausbreitung von Schallwellen als Strahlen approximiert. Für eine genauere Modellierung des Unterwasserakustik-Kanals kann eine Simulation der zeitlich variierenden Kanalimpulsantwort der Übertragungsschicht durchgeführt werden, beispielsweise mit Virtual Timeseries EXperiment (VirTEX) [97] oder dem acoustic channel simulator (ACS), wie von P. Qarabaqi und M. Stojanovic veröffentlicht. Beide Ansätze berücksichtigen in vereinfachter Weise einen Dopplereffekt und die daraus resultierende Signalverzerrung, die durch die Bewegung von Sender, Empfänger und teilweise der Meeresoberfläche während einer Signalübertragung entsteht.

Der ACS kann mit BELLHOP oder unabhängig von BELLHOP als statistischer Ansatz zur Modellierung des Unterwasserkanals, basierend auf einer minimalistischen Geometrie, verwendet werden. Der ACS setzt aber keine Oberflächen- und Bodenverlustmodelle sowie Geschwindigkeitsprofil (SSP) um und beschränkt sich auf das Basisband mit einem fixierten Spreadingfaktor k für alle Entfernungen von $k = 1.7$. Ein weiteres Modell für die Simulation der Unterwasserkommunikation ist das Waymark-Modell. Dieses Modell simuliert eine Schallübertragung unter Annahme einer spezifizierten Trajektorie der relativen Bewegung zwischen Quelle und Empfänger. Während VirTEX auf BELLHOP basiert, ermöglicht das Waymark-Modell die Nutzung verschiedener Ausbreitungsmodelle, die Impulsantworten, basierend auf den Eingabeumgebungsparametern, generieren können, und dient somit nur als mögliche Erweiterung. Diese Simulatoren können insbesondere im Bereich der physikalischen Schicht eine detaillierte Möglichkeit zur Analyse von Kommunikationssystemen mit einzelnen Punkt-zu-Punkt-Verbindungen bieten, bilden aber den Analyseraum weder ganz ab, noch ermöglichen sie in der aktuellen Form ein standardisiertes Testen.

2.6 Simulationsumgebung

In der Unterwasserkommunikation werden verschiedene Simulationsumgebungen (Frameworks) eingesetzt, um komplexe Kommunikationssysteme oder Teile wie Protokolle oder Algorithmen in einer spezifischen Umgebung zu analysieren oder zu vergleichen. Ein Beispiel hierfür ist World Ocean Simulation System (WOSS) [98–100], welches das Beamtracing über BELLHOP zur Simulation der physikalischen Schicht über *C++*-basierten Netzwerksimulationsplattformen wie dem *ns-2* [101] oder *ns-3* [102] ermöglicht. Zusätzlich bietet World Ocean Simulation System (WOSS) die Einbindung von Datenbanken, die Meeresbodeneigenschaften und Geschwindigkeitsprofil (SSP)s beinhalten, und ermöglicht damit zum teil die Integration der Umgebung.

Ähnlich wie WOSS kombiniert auch der Aqua-Sim-Simulator [103] das *ns-2*-Framework mit einem Unterwasserakustik (UWA)-Kanalmodell, welches auf einem einfachen analytischen Dämpfungsmodell und einer konstanten Ausbreitungsgeschwindigkeit von $c_w = 1500\,\mathrm{m/s}$ basiert.

Die auf *C++* basierende Netzwerksimulatoren *ns-2* gehören zur Gruppe der diskreten Ereignissimulatoren und werden für die Erforschung von Netzwerken eingesetzt. Dieser ermöglicht die Simulation von TCP-, Routing- und Multicast-Protokollen in drahtgebundenen und drahtlosen Netzwerken sowie deren benutzerspezifische Erweiterung. Bei der Datenübertragung auf der Media Access Control (MAC)-Ebene werden hierbei komplette Pakete simuliert (rahmenbasiert) und vereinfachte Funktionen genutzt, um etwa die Kollisionsdetektion von Paketen abzubilden, oder die Bestimmung einer erfolgreichen Übertragung reduziert auf die Auswertung eines Schwellwerts. Die verwendeten Modelle zur Signalausbreitung sind stark vereinfacht und berücksichtigen weder die Komplexität noch das zeitinvariante Verhalten des Unterwasserkanals oder den Einfluss der angebundenen Hardware [104].

Der weiterentwickelte Netzwerksimulator *ns-3*, der seit 2006 in *C++* und *Python* entwickelt wird, verwendet ebenfalls eine paketbasierte Simulationsumgebung. Allerdings erfolgt die Bestimmung der erfolgreichen Übertragung in *ns-3* probabilistisch unter Berücksichtigung der Modulation, des SNR sowie des aktuellen Zustands der Übertragungsschicht [102]. Ein gemeinsames Merkmal dieser Frameworks besteht darin, dass sie die physikalische Schicht und das Übertragungsmedium im Zeitbereich vernachlässigen. Alle diese Frameworks und ihre Erweiterungen berücksichtigen nur die Schichten eines Kommunikationssystems bis zur Sicherungsschicht und ignorieren die Entwicklung und den Einfluss von Hardwarekomponenten. Die Verlagerung von Komponenten aus dem Bereich

der Software auf Hardware und die schichtübergreifende Kommunikation wird nicht berücksichtigt und kann teilweise auch nicht abgebildet werden.

Die Anbindung von Hardware oder physikalischen Modems kann zwar mit anderen Programmen wie MATLAB [105]
umgesetzt werden [106], erfordert jedoch eine spezifizierte Schnittstelle und eine entwickelte prototypische Umsetzung. Die Berücksichtigung und Unterstützung von Hardwarekomponenten, einschließlich zusätzlicher Eigenschaften des Modems wie Analog Digitial Converter (ADC), Verstärker und Signalverarbeitung auf Signalebene, sowie die möglichen gegenseitigen Beeinflussungen werden entweder nicht oder nur eingeschränkt unterstützt: Dadurch wird das volle Potenzial eines Systems, tendenziell überschätzt, wie veröffentlichte Seeexperimente gezeigt haben.

2.7 Entwicklung und Plattformen

Von den 68 akustischen Modems in den Tabellen 2.1 bis 2.4 verwenden drei Modems einen FPGA. Die übrigen Modems nutzen keine konfigurierbare Hardware, sondern realisieren die Generierung der Trägerfrequenz entweder auf einem universellen Prozessor, digitaler Signalprozessor (DSP), Mikrocontroller auf Softwareebene als sogenanntes Software Defined Radio (SDR) oder direkt in analoger Hardware. Bei Herstellern wie Teledyne, WHOI und Evologics sind die Transducer je nach Variante austauschbar, was eine gewisse Konfigurierbarkeit des analogen Bereichs im Vorfeld unter erheblichem Aufwand ermöglicht.

Mittels SDR können Kommunikationssysteme auf digitaler Ebene über universelle Prozessoren und DSPs umgesetzt werden. Bei diesen Systemen wird die Komplexität der Hardwareentwicklung in den Bereich der Software verschoben, was jedoch auf Kosten von Performanz, Parallelität und dem Energiebedarf geht. Einzig die Generierung der Trägerfrequenz und der Transfer zwischen dem digitalen und analogen Bereich (Digitial Analog Converter (DAC), ADC) werden auf Hardwareebene implementiert, abhängig von der gewählten Architektur.

Durch die Realisierung in Software erhöht sich die Latenz bei der Verarbeitung und Kommunikation erheblich [107], was zu längeren Wartezeiten, einem länger blockierten Medium und somit zu einer reduzierten Datenrate und geringerer Netzauslastung führt. Der Vorteil dieser Implementierung liegt in der Anpassbarkeit von Algorithmen und Verfahren über Software-Updates, was die Konzeption beschränkter adaptiver Systeme ermöglicht. Allerdings müssen SDR-Systeme speziell für eine festgelegte Plattform entwickelt werden und können nicht ohne Weiteres ausgetauscht oder ersetzt werden.

Es gibt jedoch Ausnahmen bei SDR-basierten Systemen, wie das Micromodem-2 [19] von WHOI, das einen FPGA für die Wake-up-Funktion nutzt, und den Hersteller Evologic [18], der vereinzelte Funktionen auf den FPGA auslagert. Die in der Hardware ausgelagerten Funktionen können benutzerspezifisch ausgeschaltet und durch Algorithmen über einen Bypass in der Software ersetzt werden [108]. Das Wake-up-System aktiviert das restliche System bei Überschreiten eines festgelegten Energielevels im Empfänger. Ein Nachteil hierbei ist eine Aufwachphase von $48\,\mathrm{s}$, bis das System empfangsbereit ist, und es ist nicht eindeutig, ob das System angesprochen wurde oder Störungen im genutzten Frequenzbereich vorliegen oder ein anderes System angesprochen werden sollte. Eine Nutzung von FPGAs im Bereich der SDR zeigt das Modem von Benson [52], das den FPGA für die FSK-Modulation in der einfachsten Form nutzt. Andere Modulationen könnten zwar umgesetzt werden, gelten jedoch nach Benson als aufwendig und langwierig. Bisher gibt es keine Ansätze für akustische Kommunikationssysteme für den Unterwassereinsatz, die eine variable Grenze zwischen Hardware und Software ermöglichen. Kommerziell erhältliche Unterwasserakustikmodems und andere veröffentlichte Forschungsansätze basieren entweder auf festen Hardwarelösungen mit proprietären Protokollen oder auf SDR-basierten Algorithmen. Daher bieten sie entweder keine oder nur sehr eingeschränkte Anpassungsfähigkeiten, eingeschränkt auf Bandbreite, Datenrate und Art der Modulation. Die sogenannten adaptiven Modi erfordern manuelle Einstellungen aus einem endlichen vorgegebenen Satz von Schemata und bieten damit keine Anpassungen an Umgebungsänderungen und sind meist stark begrenzt. Den bisherigen Umsetzungen fehlt damit die Möglichkeit, (I) zwischen den operativen Modi in Echtzeit, (II) algorithmisch oder (III) architekturbezogen zu wechseln, und damit die Fähigkeit, sich an die Umgebungsbedingungen für eine bestmögliche Kommunikation anzupassen. Hierzu muss aber der Begriff *bestmöglich* in Bezug auf akustische Unterwasserkommunikation erst noch definiert werden.

2.8 Adaptive Kommunikationssysteme

Die Herausforderungen adaptiver Systeme in der akustischen Unterwasserkommunikation sind eng mit der Komplexität des Kanals und den spezifischen Anforderungen der Anwendung verbunden. Der dynamische Unterwasserkanal unterliegt starken Schwankungen aufgrund von Temperaturunterschieden, Salzgehalt, Oberflächenreflexionen, Streuung und Absorption. Diese Schwankungen können sich schnell ändern und haben direkte Auswirkungen auf die Leistung adaptiver Systeme. Zusätzlich zu den Schwankungen des Kanals ist die Reaktionszeit des Unterwasserkanals auf lokale Veränderungen aufgrund

der langsamen Schallausbreitung verzögert. Dies stellt eine Herausforderung für die zeitnahe Anpassung der adaptiven Systeme dar. Unterwasserumgebungen werden durch Störungen und Interferenzen, einschließlich Geräuschen von Schiffen, Meerestieren und Umgebungsgeräuschen, die das SNR beeinflussen können, was wiederum die Leistung adaptiver Systeme beeinträchtigt.

Die Implementierung adaptiver Modulationswechsel erfordert unter anderem eine kontinuierliche Überwachung der Kanalbedingungen und die Berechnung optimaler Parameter. Dies kann zu einem erhöhten Energiebedarf führen. Insbesondere für batteriebetriebene Systeme mit begrenzter Kapazität stellt dies eine Herausforderung dar. Die Konzeption und Umsetzung adaptiver Systeme erfordern zudem komplexe Konzepte und eine sorgfältige Abstimmung der systemübergreifenden Parameter. Dies gilt nicht nur für Modulationsschemata, sondern auch für andere adaptive Aspekte des Systems. Die derzeit publizierten Systeme können dazu in Einzelträger- und Mehrträgersysteme unterteilt werden, wobei jeweils unterschiedliche Herausforderungen und Anforderungen bestehen:

Die Publikationen [109–113] führen das adaptive Einzelträgersystem in Form von feststehenden Systemkomponenten (Single-Carrier-System) ein. Die Systeme haben definierte Parameterbereiche, die in Abhängigkeit des berechneten SNR ausgewählt und genutzt werden. Die Publikationen lassen aber unklar, inwieweit die Interferenzen anderer Teilnehmer und die Reflexion des eigenen Signals in der Unterwasserumgebung einbezogen werden oder der Austausch der Parameter den Kommunikationspartnern mitgeteilt wird. Des Weiteren ist unklar, wie eine gemeingültige Entscheidung für die Kommunikationsteilnehmer getroffen werden kann. Abhängig von der Position, Ausrichtung und weiteren orts- und anwendungsspezifischen Parametern ist der SNR an dieser Stelle, was auch die Autoren einräumen, nicht aussagekräftig genug. Auch die Annahme, dass der Empfänger über dieselben Informationen (SNR) wie der Sender verfügt, wovon im Bereich der Unterwasserkommunikation nicht ausgegangen werden kann, muss gewährleistet sein.

Bei den sogenannten Mehrträgersystemen wird zwischen Systemen, die über mehrere Einzelsysteme verfügen, und Systemen mit umschaltbaren Modulaktionstypen unterschieden. So verfügt die Umsetzung in [114] über Einzelsysteme, die wahlweise genutzt werden können, und in [109, 113] wird nur der Modulationsmodus adaptiv gewechselt. Damit bilden auch Mehrträgersysteme einen möglichen adaptiven Ansatz, der aber die Kommunikation wie auch bei den Einzelträgersystemen vor das Problem der Systemauswahl, dem Austausch von Informationen und dem komplexen Kanal stellt. Zudem wird in den Publikationen von homogenen (identischen) Kommunikationssystemen ausgegangen, was nicht unbedingt zu einer optimalen Kommunikationsstrategie führt.

Zusammenfassung

Der modellbasierte Ansatz zur Entwicklung von Unterwasserkommunikationssystemen bietet eine theoretisch vielversprechende Methodik im Bereich der akustischen Kommunikation, eröffnet aber neue Möglichkeiten zur Simulation und Optimierung von Systemen vor deren Implementierung. Obwohl die theoretischen Grundlagen des modellbasierten Designs etabliert sind und Simulationen vielversprechende Ergebnisse liefern, steckt die komplette Umsetzung, inklusive der physischen, noch in der Anfangsphase.Die Integration und Anwendung dieser Modelle in realen, einsatzfähigen Systemen, insbesondere in der Unterwasserkommunikation, ist nach wie vor eine Herausforderung. Die komplexe und dynamische Natur des Unterwasserkanals, einschließlich Mehrwegeausbreitung, Dopplereffekten und Signalabsorption, erschwert die direkte Übertragung der theoretischen Modelle auf die Praxis. Trotz dieser Fortschritte sind die Implementierungen oft noch unvollständig, und die Interoperabilität zwischen verschiedenen Systemen ist eine Herausforderung. Die Entwicklung von Standards wie JANUS, welcher nur für eine spezifische Version von FH-FSK definiert ist, zeigt, dass es Bemühungen gibt, eine einheitliche Grundlage zu schaffen, aber es besteht noch erheblicher Bedarf an Forschung und Entwicklung, um diese Standards in breite, praktische Anwendungen umzusetzen. Damit gewinnt der Bereich der Unterwasserkommunikation durch die Fortschritte in der Anwendung von modellbasierten Ansätzen und Algorithmen auf FPGAs und ähnlichen Systemen zunehmend an Bedeutung. Dennoch befindet sich dieser Bereich noch in einer Phase des Übergangs von theoretischen Konzepten zur praktischen Anwendung. Es besteht ein großer Bedarf an weiteren Entwicklungen hin zu offenen, standardisierten und interoperablen Systemen, um die vielversprechenden Vorteile des modellbasierten Designs voll ausschöpfen und alle Phasen der Umsetzung abbilden zu können.

3

Unterwasserkanal

Die Entwicklung, Implementierung, Erprobung und der Performanzvergleich von Systemen für die akustische Kommunikation in Unterwasserumgebungen stellen aufgrund der spezifischen Eigenschaften des Unterwasserkanals eine anspruchsvolle Herausforderung dar. Der Unterwasserkanal selbst ist durch eine Vielzahl von Faktoren geprägt, die die Ausbreitung, Reflexion und Verzerrung der gesendeten Schallwellen beeinflussen. Hinzu kommen Störgeräusche, die in Kombination mit der vergleichsweise niedrigen Ausbreitungsgeschwindigkeit der Schallwellen auftreten. Die komplexen Eigenschaften des Unterwasserkanals, der die Grundlage für akustische Kommunikation unter Wasser bildet, erfordern fundierte Kenntnisse in verschiedenen Bereichen. Dazu zählen die physikalischen Eigenschaften des Wassers, die Umweltbedingungen, die Topografie des Meeresbodens, die Auswirkungen der Seeoberfläche, die vorherrschenden Störgeräusche sowie die spezifischen Anforderungen der jeweiligen Anwendungen. Diese Herausforderungen sind besonders relevant für die Entwicklung und Erforschung von akustischen Kommunikationssystemen und bilden die Basis für die Schaffung effizienter Kommunikationslösungen.

In diesem Kapitel werden die Herausforderungen, die der Unterwasserkanal hinsichtlich der Schallausbreitung und der Störgeräusche mit sich bringt, diskutiert und deren Auswirkungen formalisiert. Neben den Spezifikationen eines anwendungs- und umgebungsspezifischen Testraums mit definierten Testpunkten in Abschnitt 3.3.1 wird eine Erweiterung des etablierten Tools BELLHOP eingeführt. Diese Erweiterung ermöglicht die Abbildung der Dämpfungseigenschaften, der natürlichen Grenzen des Mediums und der Richtcharakteristik von Transducern, die einen signifikanten Einfluss auf die Performanz haben können. Zudem wird in Abschnitt 3.3.2.2 ein Alternative zu BELLHOP, dem geometrisches Kanalmodell, eingeführt, dessen Komplexität und Laufzeit im Vergleich zu BELLHOP, zusammengefasst in Abschnitt 3.3.2.3 deutlich geringer sind, während es für mittlere und tiefe Gewässer nahezu identische Ergebnisse liefern kann. Weiterhin wird ein Verfahren zur modellbasierten Simulation des

Umgebungsrauschens in Simulink und hardwarebasiert auf FPGAs in Abbildung 3.10 vorgestellt. Dieses Verfahren berücksichtigt die Hauptparameter für dynamische Szenarien und setzt diese mithilfe von programmierbaren Finite Impulse Response (FIR)-Filter um.

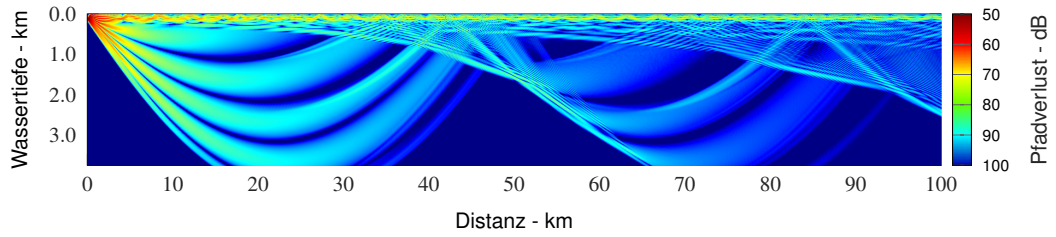
Damit legt dieses Kapitel den Grundstein für ein standardisiertes Testverfahren zur Bewertung und zum Vergleich von akustischen Unterwasserkommunikationssystemen. Durch die Definition und Erzeugung eines Analyseraums, der die speziellen Herausforderungen des Unterwasserkanals, wie Rausch- und Ausbreitungseigenschaften, berücksichtigt, wird ein objektiver Vergleich von Kommunikationssystemen ermöglicht. Die klare Definition einheitlicher Bewertungskriterien und Maßstäbe erleichtert die bisher fehlende umgebungs- und anwendungsspezifische Vergleichbarkeit und Validierung der Kommunikationssysteme. Überdies ersetzt dieses Verfahren teure und aufwendige Feldtests, was Kosten reduziert und Zeit spart, indem es die spezifischen Systemeigenschaften wie Richtcharakteristik und Sensitivität berücksichtigt.

3.1 Herausforderungen

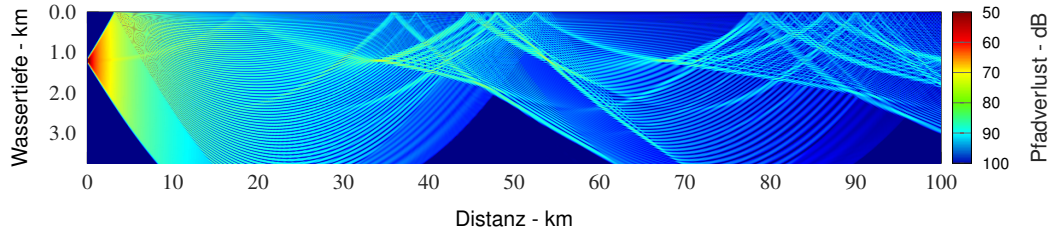
Die akustische Unterwasserumgebung ist geprägt von einer Vielzahl von Einflussfaktoren und Störquellen, die die Ausbreitung von Schallwellen erheblich beeinflussen. Zu den Hauptfaktoren gehören Umgebungsrauschen, Reflexionen an der bewegten Wasseroberfläche und dem Meeresboden, den Brechungen an den Thermoklinen, die sowohl die frequenz- und temperaturabhängigen Ausbreitungsgeschwindigkeiten als auch die frequenz- und tiefenabhängige Absorption beeinflussen. Zudem erschweren die Dämpfung, der bewegungsbasierte Dopplereffekt und die Mehrwegeausbreitung die Entwicklung und Optimierung von

Tabelle 3.1: Parameter ausgewählter Unterwasserkanäle

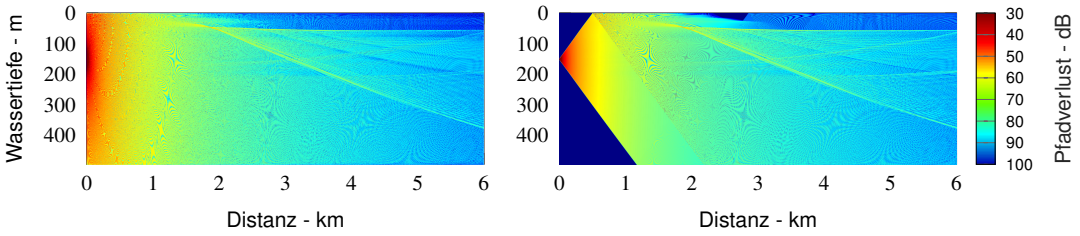
Umgebung	a	b	c	d	e	f
Sender						
Anzahl	1	1	1	1	1	1
Tiefenposition - m	100	1000	150	150	150	150
Trägerfrequenz f_c - kHz	0.2	0.2	10	10	10	10
Abstrahlwinkel - $^\circ$	$\pm$16.7	$\pm$16.7	$\pm$180	$\pm$16.7	$\pm$16.7	$\pm$16.7
Empfänger						
Distanz - km	100	100	6	6	6	6
Messpunkte (Distanz)	4001	4001	600	1001	1001	1001
Tiefe - m	3750	3750	495	495	495	495
Messpunkte (Tiefe)	761	761	761	761	761	761



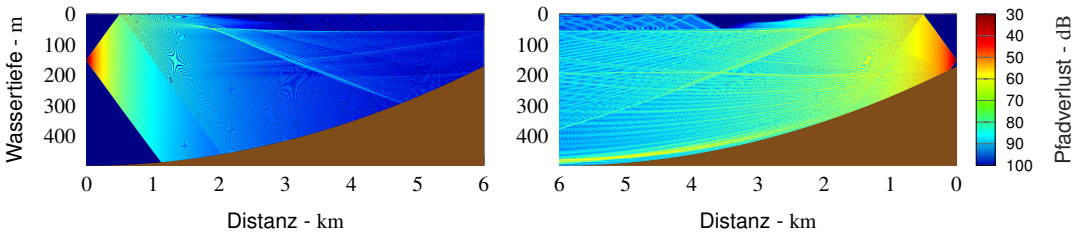
(a) Arktischen Bedingungen mit einer Sendertiefe von $150\,m$



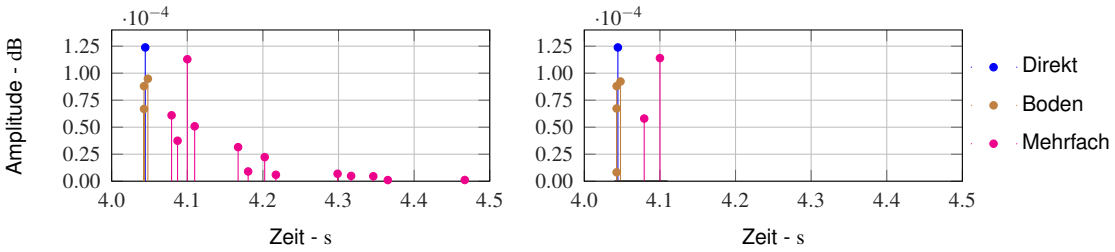
(b) Arktischen Bedingungen mit einer Sendetiefe von $1\,\mathrm{km}$



(c) Omnidirektionale (li) und beschränkte (re) Abstrahlmaske



(d) Steigenden (li) und fallenden (re) Bodenbeschaffenheit



(e) Kanalimpulsantworten basierend auf Abbildung 3.1d

Abbildung 3.1: Umgebungsbedingte Pfadverluste basierend auf Tabelle 3.1

Unterwasserkommunikationssystemen erheblich. Diese Herausforderungen spiegeln sich in der Vielfalt der verfügbaren Kommunikationssysteme wider und verdeutlichen, dass die Spezifikationen und Bewertungen eines Systems ohne Berücksichtigung der Kommunikationsumgebung unvollständig sind.

Um die Herausforderungen der Unterwasserkommunikation zu verdeutlichen, zeigen die Abbildungen 3.1a bis 3.1d, unter Verwendung von BELLHOP, die resultierenden Pfadverluste ausgewählter Umgebungen und deren Auswirkungen auf Kommunikationsreichweite und Qualität. Die Einstellungen und Parameter sind dazu in der Tabelle 3.1 und in den Listings A.1 bis A.5 zusammengefasst. Die Abbildung 3.1a, basierend auf Listing A.1, demonstriert dazu die theoretisch mögliche Kommunikationsreichweite von mehreren hundert Kilometern unter arktischen Bedingungen, während Abbildung 3.1b definiert in Listing A.2, zeigt, wie eine Änderung der Kommunikationstiefe die Reichweite erheblich reduziert. Der Pfadverlust in der Abbildung 3.1c erzeugt mit Listings A.3 and A.4 verdeutlicht die Auswirkungen von verschiedenen Transducer-Charakteristiken mit einer omnidirektionalen Abstrahlung (links) und beschränkt (rechts). Die Abbildung zeigt die daraus entstehenden Schattenzonen und Bereiche mit starker Mehrwegausbreitung. Diese Effekte führen zu einem positions- und lageabhängigen Kommunikationsverhalten, wobei die akustischen Transducer eine entscheidende Rolle innehaben, da sie akustische Signale in elektromagnetische Signale umwandeln und umgekehrt. Ihre grundlegenden Eigenschaften, wie die frequenzspezifische Signalumsetzung, Abstrahleigenschaften und Empfindlichkeit, sind für die Übertragung von Informationen in der herausfordernden akustischen Unterwasserumgebung daher von großer Bedeutung.

In der Abbildung 3.1d wird der Signalverlust eines mitteltiefen Gewässers mit einem Boden aus lehmigem Schlamm und einer quadratisch ansteigenden Bodentiefe illustriert. Für die Kommunikation über diese Umgebung (Listing A.5) wird in Abbildung 3.1e die dazugehörigen Impulsantworten dargestellt. Die jeweils rechte Abbildung in 3.1d und 3.1e zeigt die Auswirkungen einer getauschten Sender-Empfänger-Position unter den gleichen Oberflächen- und Bodenbedingungen. Beide Abbildungen beziehen sich auf eine Sender- und Empfängertiefe von $150\,\mathrm{m}$ und eine Kommunikationsentfernung von $6\,\mathrm{km}$. Sie verdeutlichen, wie Bodenbeschaffenheit und Position von Sender und Empfänger die Signalpfade, dargestellt über die Kanalimpulsantworten, beeinflussen können.

Die Abbildung 3.1e zeigt, dass die direkte Verbindung zwischen Sender und Empfänger (blau), auch als Sichtverbindung (LOS) bezeichnet, und die Reflexionen am Boden (braun) nahezu identisch sind, sofern angenommen wird, dass der Boden spiegelnde Eigenschaften besitzt. Die Mehrwegeausbreitung durch

mehrfache Reflexionen (magenta) hingegen zeigt signifikante Unterschiede. In diesem Beispiel würde sich das Signal to Interference Ratio (SIR) erheblich verschlechtern, je länger die Übertragung dauert und kann damit auch die Inter-Frame-Interferenz (IFI) beeinflussen, was die Leistung und Zuverlässigkeit von Kommunikationssystemen zusätzlich beeinträchtigen kann.

Diese statischen Darstellungen der Impulsantworten sind jedoch zeitlich variabel, da sie durch dynamische Faktoren wie die Bewegung des Wassers, Wellen, ortsspezifische thermische Schichten, den Meeresboden, Flora und Fauna sowie den Wind beeinflusst werden. Es wird deutlich, dass die Impulsantworten des Kanals von der Position, Lage und den Abstrahleigenschaften von Sender und Empfänger abhängen und durch die dynamischen und zeitlich veränderlichen Eigenschaften der Unterwasserumgebung beeinflusst werden. Diese Faktoren unterstreichen die Notwendigkeit eines umfassenden Verständnisses der Kommunikationsumgebung für die Entwicklung effektiver Unterwasserkommunikationssysteme.

3.2 Eigenschaften

Der Unterwasserkanal ist durch eine Vielzahl physikalischer Eigenschaften charakterisiert, die eine akustische Kommunikation maßgeblich beeinflussen. Dazu gehören der Signalverlust durch Streuung und absorbierende Effekte des Wassers, die sich negativ auf die Signalreichweite auswirken. Die Ausbreitungsgeschwindigkeit des Schalles in Form eines Geschwindigkeitsprofils (Geschwindigkeitsprofil (SSP)) kombiniert mit dem bewegungsbasierten Dopplereffekt und der Mehrwegeausbreitung über die Flora, Faune und Grenzflächen erweitert den komplexen Charakter des Unterwasserkanals. Zusätzlich beeinflussen auch die Umgebungsparameter wie die Temperatur und Salinität des Wassers sowie Störquellen wie Umgebungsrauschen die Kommunikation. Die Kenntnisse über die physikalischen Eigenschaften des Unterwasserkanals sind daher entscheidend für das Design und die Optimierung von Unterwasserkommunikationssystemen.

3.2.1 Kanalimpulseantwort (CIR)

Der Unterwasserkanal ist ein physikalisches Medium, das die Übertragung von akustischen Signalen ermöglicht. Ein akustisches Signal beschreibt eine Schallwelle, die als Druckvariationen durch ein Medium wie Luft oder Wasser übertragen wird. In der Unterwasserakustik wird das akustische Signal meist

durch den SPL (Schalldruckpegel) relativ zu einem Bezugswert von $1\,\mu\mathrm{Pa}$ angegeben. Der SPL beschreibt die Stärke des Schalles und wird in Dezibel (dB) gemessen. Die Reaktion des Kanals auf ein gesendetes Signal wird durch die Kanalimpulseantwort (CIR) beschrieben, die den zeitlichen Verlauf des empfangenen Signals am Empfänger darstellt:

$$h(t) = \sum_{n=1}^{N} a_n \cdot \delta_\sigma(t - \tau_n) \text{ mit } \delta_\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(t-\tau_n)^2}{2\sigma^2}} \tag{3.1}$$

Die Gl. (3.1) beschreibt die Kanalimpulseantwort (CIR) als Summe aller N Signalpfade des Kanals, mit der dazugehörigen Amplitude a_n und der Verzögerung τ_i zeitlich über die Dirac-Delta-Funktion, welche den Dirac-Impuls[1] in diesem Beispiel über die Annäherung an eine Gaußkurve abgebildet. In dem Unterwasserkanal mit seiner herausfordernden Komplexität ändert sich aber in Abhängigkeit der Parameter $p_n \in \mathrm{P}$ des Kanals und der Eigenschaften des gesendeten akustischen Signals auch die Kanalimpulsantwort. Um das sich ändernde Verhalten abzubilden, kann für jeden Zeitpunkt die Kanalimpulsantwort mit dem aktuellen Set von Parametern P eine CIR bestimmt und zeitlich geordnet über eine Kanalimpulsantwortmatrix, mit t_k als zeitlicher Abstand, der CIRs angegeben werden. Die Impulsantwort kann damit in Abhängigkeit der Parameter p_k zum Zeitpunkt t_k wie folgt erstellt werden:

$$h_k(t) = \sum_{n=1}^{N} a_{n,\mathrm{k}} \cdot \delta_\sigma(t - \tau_{n,\mathrm{k}}) \tag{3.2}$$

Unter der Annahme, dass die Zeitabstände von t und t_k identisch sind und alle Kanalimpulsantworten die gleiche Anzahl von Pfaden N_k haben, kann die Matrix der Kanalimpulsantwort wie in Gl. (3.3) abgebildet werden. Um aber solch eine Kanalimpulsmatrix erzeugen zu können, müssen die Amplituden und Verzögerungen der Signalpfade in Abhängigkeit der Parameter über das Geschwindigkeitsprofil, den Signalverlust, den Dopplereffekten und die Richtcharakteristik bestimmt werden.

[1] In der Signaltheorie wird der Dirac-Impuls oft als eine Funktion betrachtet, die bei der Integration über alle Zeit eine Einheit ergibt, d. h., die Bedingung $\int_{-\infty}^{\infty} \delta(t)\,dt = 1$ erfüllt. Diese Interpretation ist nützlich, um die Impulsantwort eines Systems zu beschreiben, da der Dirac-Impuls als idealisierte Modellierung eines sehr kurzen und starken Signals dient. Nach Dirac[115] wird der Dirac-Impuls $\delta(t)$ nicht als gewöhnliche Funktion, sondern als Distribution verstanden, die für jede Testfunktion $f(t)$ die Bedingung $\int_{-\infty}^{\infty} \delta(t) f(t)\,dt = f(0)$ erfüllt.

$$H = \begin{bmatrix} h_1(1) & h_1(2) & \dots & h_1(N_1) & 0 & \dots & \dots & \dots & 0 \\ 0 & h_2(1) & h_2(2) & \dots & h_2(N_2) & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \dots & h_k(1) & h_k(2) & \dots & h_k(N_k) \end{bmatrix} \quad (3.3)$$

3.2.2 Geschwindigkeitsprofil des Unterwasserkanals

Die Schallgeschwindigkeit (c_w) im Wasser ist ein grundlegender Parameter, welcher die Signalausbreitung durch die Nebenwirkung des Durchlaufens von Schichten mit unterschiedlicher Schallgeschwindigkeit beeinflusst. Die Ausbreitungsgeschwindigkeit des Schalls wurde bereits vielfach untersucht [116–120] und wird maßgeblich von den tiefenabhängigen Parametern Temperatur T, Salzgehalt S, Druck Pa und der Wassertiefe selbst d_z beeinflusst. Ein Beispiel für eine vereinfachte Berechnung basiert auf der Gleichung Gl. (3.4), wie in der Publikation von Medwin in Gl. (3.4) beschrieben.

$$c_w = \begin{array}{l} 1449.2 + 4.6T(d_z) - 0.0055T(d_z)^2 + 0.00029T(d_z)^3 \\ +(1.34 - 0.01T(d_z)) \cdot (S(d_z) - 35) + 0.016d_z \end{array} \quad (3.4)$$

Die Beschreibung der Schallgeschwindigkeit in Abhängigkeit von Tiefe und Position kann mithilfe des Geschwindigkeitsprofil (SSP) abgebildet werden. Der Vergleich der gängigsten Formeln zur Bestimmung des SSP von Mackenzie [117], Medwin [116], Wilson [120], Delgrosso [118] und Chen-Millero [119] ist in Abbildung 3.3 in Relation von Temperatur, Salzgehalt und Druck gezeigt. Ein Vergleich der Geschwindigkeitsprofile zeigt, dass die vereinfachte Formel von Medwin eine minimale Differenz von etwa $3\,\mathrm{m\,s^{-1}}$ im Verhältnis zu einem Durchschnitt von $1490\,\mathrm{m\,s^{-1}}$ aufweist, im Vergleich zu Mackenzie, mit einer Genauigkeit von $\approx 0.1\,\mathrm{m\,s^{-1}}$ wie Dushaw et al. in [121] gezeigt haben.

Für die Bestimmung der SSP bietet die National Oceanic and Atmospheric Administration (NOAA) freien Zugriff auf gesammelte orts- und tiefenspezifische Messdaten. Mit den bereitgestellten Daten[2] kann das SSP, wie beispielhaft in Abbildung 3.3 dargestellt, auf Basis der eingeführten Formeln berechnet werden.

[2] Alternativ können die Daten auch über Datenbanken wie `http://staff.washington.edu/dushaw/WOA/` mithilfe der zur Verfügung gestellten Funktionen extrahiert werden.

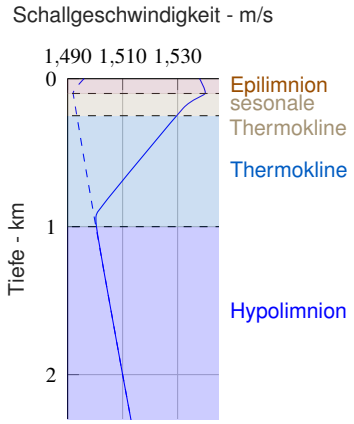


Abbildung 3.2: Munk-basierendes Tiefseeprofil

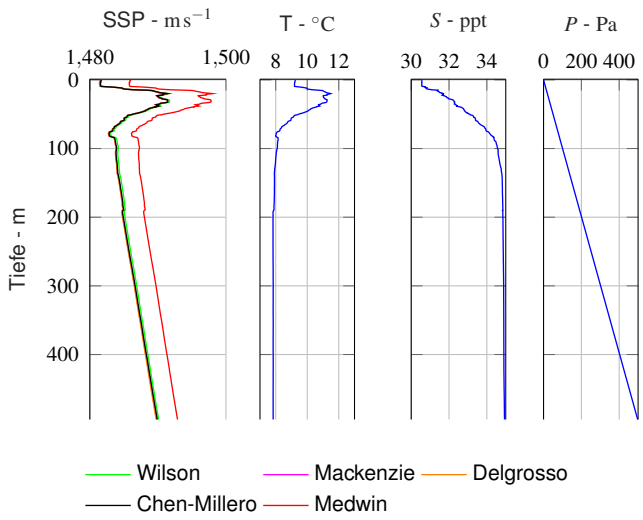


Abbildung 3.3: SSPs in Abhängigkeit von Temperatur, Salzgehalt und Druck

3.2.3 Schichten des Unterwasserkanals

Basierend auf der Temperatur und dem daraus resultierenden (SSP) können Gewässer in drei wesentliche Schichten unterteilt werden: die Oberflächenschicht (Epilimnion), die mittlere Schicht (Thermokline) und die untere Schicht (Hypolimnion). Für die Oberflächenschicht ist auch der Begriff „Oberflächenkanal“ (surface duct) gebräuchlich. Der spezielle Bereich der Oberflächenschicht mit konstanter Temperatur wird zusätzlich als Mischschicht bezeichnet und unterliegt starken Schwankungen in Abhängigkeit von Jahreszeit, lokalen Bedingungen und Wind-induzierten Wellen, die bei einer Analyse beachtet werden müssen.

Der Oberflächenkanal entsteht immer dann, wenn die Schallgeschwindigkeit in der Nähe der Wasseroberfläche geringer ist als die Schallgeschwindigkeit der oberen Thermokline, was zu einem positiven Gradienten der Schallgeschwindigkeit im Wasser führt. Die Tiefe der Thermokline variiert je nach orts- und umgebungsspezifischen Einflüssen und ist in Abbildung 3.2 als saisonale Thermokline markiert.

Abhängig von den Schallgeschwindigkeitsprofilen und der Wassertiefe treten drei wesentliche Effekte bei der Ausbreitung im Tiefenkanal (DSC) auf, der auch als Schallausbreitungskanal in großer Tiefe (SOund Fixing And Ranging (SOFAR)-Kanal) bekannt ist. Diese Effekte sind die Oberflächenkanalausbreitung, die Schattenzonen und die Ausbreitung von Schallwellen mittels Konvergenzzonen.

Der SOFAR-Kanal bildet sich zwischen dem negativen Gradienten (Thermokline) und dem positiven Gradienten (Hypolimnion) und ermöglicht eine extrem weite Ausbreitung von akustischen Signalen, sofern sich Quelle und Senke innerhalb des SOFAR-Kanals befinden. In Abbildung 3.1b wird dieses Verhalten mit einer Signalausbreitung von bis zu $400\,\mathrm{km}$ illustriert. Eine der gebräuchlichsten Parametrisierungen für Tiefenkanäle ist die Anwendung des MUNK-Profils [122] über Gl. (3.5). Hierbei repräsentiert z_0 den Übergang von negativen zu positiven Gradienten der Schallgeschwindigkeit. Die c_0 gibt die Ausbreitungsgeschwindigkeit des Schalls an, und die dimensionslose Konstante $\varepsilon = 0.007\,37$ wird verwendet.

Ein Beispiel für ein Geschwindigkeitsprofil und die zugehörige Strahlenausbreitung im SOFAR-Kanal ist in Abbildung 3.4 für $c_0 = 1500\,\mathrm{m\,s^{-1}}$ und $z_0 = 1\,\mathrm{km}$ dargestellt.

$$c(z) = c_0 \cdot (1 + \varepsilon \cdot [x - 1 + \exp(-x)]) \text{ mit } x = 2 \cdot \frac{z - z_0}{z_0} \tag{3.5}$$

Ein Oberflächenkanal entsteht bei einem Temperaturprofil, wie es in Abbildung 3.5 dargestellt ist, mit einer steigenden und beim Übergang zur Thermokline abnehmenden Schallgeschwindigkeit. Die Tiefe des Oberflächenkanals kann je nach Jahreszeit und geografischer Lage zwischen $5\,\mathrm{m}$ und $200\,\mathrm{m}$ variieren. Ein Signal, das innerhalb des Oberflächenkanals gesendet wird, bricht durch den positiven Geschwindigkeitsgradienten zur Oberfläche hin. Anschließend wird das Signal periodisch an der Wasseroberfläche reflektiert und erneut zur Oberfläche zurück gebrochen. Ein Nachteil des Oberflächenkanals ergibt sich durch die Dämpfung und Phasenverschiebung des Signals durch die Oberflächenreflexion, was die erfolgreiche Detektion und Decodierung erschwert.

Signale, welche die Grenze zur Thermokline durchbrechen, werden durch den negativen Geschwindigkeitsgradienten zum Boden hin gebrochen und ändern ihre Richtung nur bei ausreichender Tiefe über einem positiven Geschwindigkeitsgradienten der Schallgeschwindigkeit (Abbildung 3.5). Dabei entsteht eine Region, die fast frei von Signalen ist, die sogenannte Schattenzone, die in Abbildung 3.5 beispielhaft für eine Tiefe der Signalquelle von $150\,\mathrm{m}$ gezeigt wird. Schattenzonen sind zwar nie vollständig frei von Schallenergie, jedoch ist die lokale Schallenergie in Schattenzonen für eine erfolgreiche Detektion und Decodierung zu gering.

Die Schattenzone kann durch eine Änderung der Signalquellentiefe (Sendertiefe) verschoben werden. Allerdings führt dies dazu, dass die Schattenzone in Richtung Oberfläche erweitert wird, wie in Abbildung 3.6 für die Signalquellentiefe von $500\,\mathrm{m}$ dargestellt ist. Bei Gewässertiefen von $d_z > 3000\,\mathrm{m}$ werden aber die in Abbildung 3.6 zum Grund hin gebrochenen Schallwellen so lange gebrochen, bis sich der Gradient der Brechung umkehrt, ohne am Boden reflek-

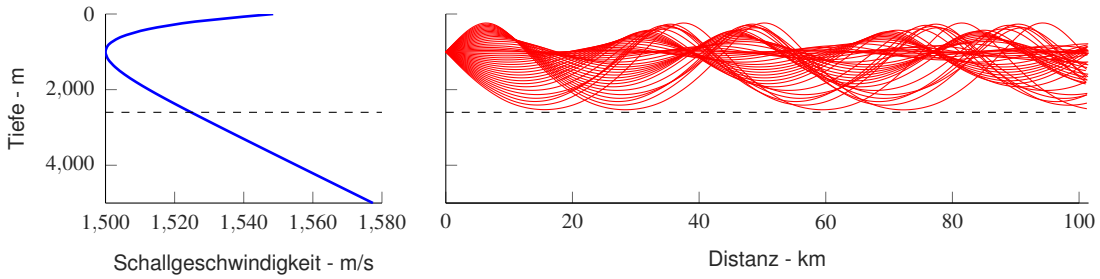


Abbildung 3.4: SSP mit Strahlenausbreitung im SOFAR-Kanal

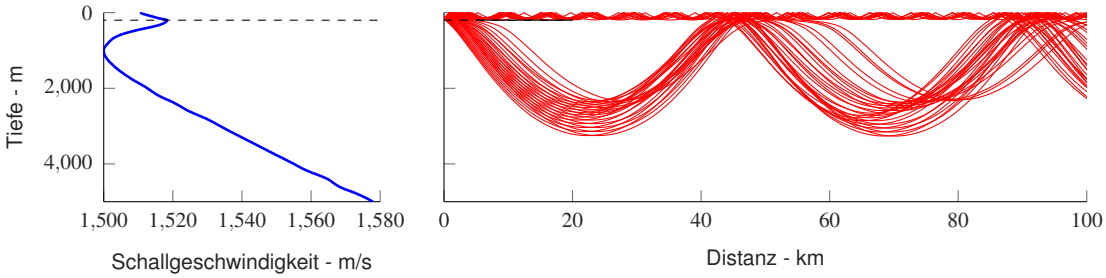


Abbildung 3.5: SSP mit Strahlenausbreitung im Oberflächenkanal

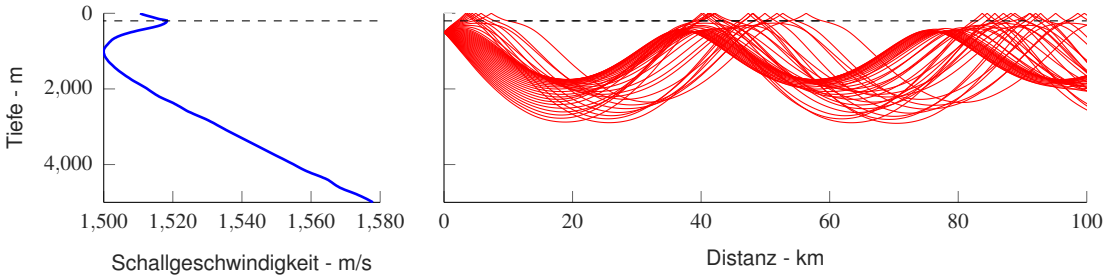


Abbildung 3.6: SSP mit Strahlenausbreitung unterhalb des Oberflächenkanals

tiert zu werden. Dieser Effekt wird hauptsächlich bei der passiven Sonarortung verwendet und als Konvergenzzone bezeichnet. Die erste Konvergenzzone bildet sich in einer Entfernung von ungefähr $30\,\mathrm{km}$ bis $50\,\mathrm{km}$ [123]. Ein Nachteil dieser Zone ist die starke Dämpfung des Signals, was die Detektion und Decodierung stark erschwert. Diese Zonen sind auf den ersten Blick für die Unterwasserkommunikation aufgrund ihrer hohen Distanz und Dämpfung uninteressant. Sie können jedoch zur Bestimmung von Kommunikationstrajektorien mit erhöhter Reichweite genutzt werden. Definierte Trajektorien könnten dadurch zu einer erhöhten Erfolgschance und Reichweite von Datenübertragungen führen oder die Lebensdauer von kapazitiv beschränkten und autarken Systemen verlängern.

3.2.4 Signalverlust im Unterwasserkanal

In der Unterwasserakustik spielen Signalverluste eine entscheidende Rolle und stellen eine Herausforderung dar. Diese Verluste im akustischen Unterwasserkanal werden durch Effekte wie Streuung, Absorption und Reflexion an der Wasseroberfläche sowie am Meeresboden verursacht.

3.2.4.1 Streuverlust

Im Gegensatz zur terrestrischen Kommunikation wird der Unterwasserkanal durch eine natürliche Oberfläche und den Seegrund begrenzt. In der akustischen Unterwasserkommunikation unterscheidet man in der Theorie zwischen sphärischen und zylindrischen Ausbreitungen. Die sphärische Ausbreitung beschreibt, wie auch in der terrestrischen Kommunikation, basierend auf der Formel nach Friis und der abgewandelten Variante des Log-Path-Loss-Models von Rappaport [124] den Verlust des Signals, wenn sich eine Schallwelle gleichmäßig in alle Richtungen (Omnidirektional) von einer Quelle weg ausbreitet. Somit entspricht die Gesamtleistung der Intensität multipliziert mit der Fläche der Kugel in Gl. (3.6) mit P_{sph} als Gesamtleistung über einen Kugelreferenzradius von $r_0{=}1\,\mathrm{m}$ angegeben. Erreicht die omnidirektionale Ausbreitung die natürlichen Grenzen des Mediums (Boden, Oberfläche), nähert sich die Ausbreitung mit steigendem Radius r der zylindrischen Ausbreitung, schematisch in Abbildung 3.7 dargestellt. Die zylindrische Leistung P_{zyl} kann, vorausgesetzt der Schall ist gleichmäßig über die Oberfläche mit einer Meerestiefe d_z verteilt, über Gl. (3.6) angenähert werden.

$$P_{sph} = 4\pi r^2 I \quad , \quad P_{zyl} = 4\pi r d_z I \qquad \text{mit} \quad I = I_0 \left(\frac{r_0^2}{r^2} \right) \tag{3.6}$$

Basierend auf diesen Annahmen kann der idealisierte Streuverlust PL_{sp} über dem Streufaktor k[3] mit einem Wert zwischen $k{=}\,2$ für die sphärische und $k{=}\,1$ für die zylindrische Ausbreitung in Abhängigkeit der Distanz zwischen Sender und Empfänger R (entspricht dem Radien r_z, r_s in Abbildung 3.7) über die Gl. (3.7) beschrieben werden [125]:

$$PL_{sp}(R,k) = k \cdot 10\,log(R) \tag{3.7}$$

Die Literatur nimmt bei Gedankenspielen, Berechnungen und Simulationen einen Mischwert von $k{=}\,1.7$ an, wie die Publikationen [125–128] beispielhaft zeigen, was einem mitteltiefen Gewässer entsprechen soll. Bei einer genaueren

[3] Aus dem Bereich der terrestrischen kabellosen Kommunikation als Pfadverlustexponent geläufig.

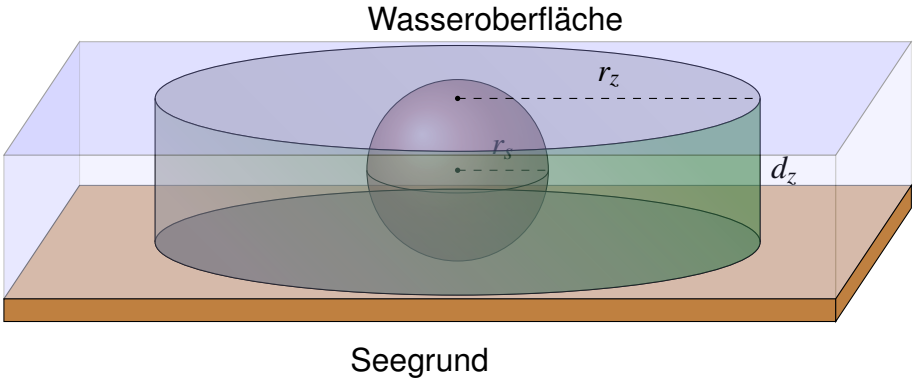


Abbildung 3.7: Schematische Abbildung der sphärischen und zylindrischen Ausbreitung in Bezug auf die natürliche Begrenzung des Kanals

Betrachtung wird aber klar, dass dies zu viel zu optimistischen Ergebnissen im Bereich der sphärischen Ausbreitung ($R \leq d_z$) und zu einem erheblichen Pessimismus für sehr große Kommunikationsdistanzen ($R >> d_z$) führt.

3.2.4.2 Absorptionsverlust

Der in Wasser auftretende Absorptionsverlust entsteht durch die Umwandlung der Schallwellen in andere Formen der Energie infolge ihrer Wechselwirkung. Eine einfache Formel für den Absorptionsverlust PL_{ab} in Abhängigkeit von der Distanz zwischen Sender und Empfänger R in km, der Frequenz f_c in kHz und dem Absorptionskoeffizienten α ist in der Gl. (3.8) dargestellt.

$$PL_{ab}(R, f_c) = 10log(\alpha(f)) \cdot R \tag{3.8}$$

Der Absorptionskoeffizient α für Schall unter Wasser ist frequenzabhängig und kann empirisch durch verschiedene Formeln beschrieben werden. Vier häufig verwendete Gleichungen, abgeleitet von Thorp [129], Fisher-Simons [130], Francois-Garrison [131, 132] und Van Moll [133], beruhen auf Feldversuchen. Gemein haben die Ansätze, dass im Frequenzbereich von $100\,\mathrm{Hz}$ bis $1000\,\mathrm{kHz}$ die Schallabsorption im Meerwasser als Summe der Beiträge von reinem Wasser, Borsäure und Magnesiumsulfat betrachtet werden kann.

Die von Thorp entwickelte Formel (Gl. (3.9)) wurde für den Bereich von $0.1\,\mathrm{kHz}$ bis $3\,\mathrm{kHz}$ definiert und wurde im Jahr 1967 für Frequenzen unter $50\,\mathrm{kHz}$, für

Temperaturen bis $4\,^{\circ}\mathrm{C}$ und einer Wassertiefe von ca. $1000\,\mathrm{m}$ eingeführt.

$$\alpha(f) = 1.0936\left[\frac{0.1f^2}{1+f^2} + \frac{40f^2}{4100+f^2}\right] \tag{3.9}$$

Fisher-Simon beschreibt eine Formel (Gl. (3.10)) für einen tiefen- und temperaturabhängigen Frequenzbereich von $3\,\mathrm{kHz}$ bis $500\,\mathrm{kHz}$. Sie repräsentiert den Einfluss der drei Hauptbestandteile von Meerwasser – Borsäure $*_1$, Magnesiumsulfat $*_2$ und reinem Wasser $*_3$ über die Koeffizienten des Temperatureffektes A_1, A_2 und A_3 und P_1, P_2 und P_3 als druckabhängigen Effekte von Meerwasser, ergänzt um die Relaxationsfrequenzen von Borsäure f_1 und Magnesiumsulfat f_2.

$$\alpha(f) = \frac{A_1P_1f_1f^2}{f_1^2+f^2} + \frac{A_2P_2f_2f^2}{f_2^2+f^2} + A_3P_3f^2 \tag{3.10}$$

Eine breitere Abdeckung des Frequenzbereichs mit $400\,\mathrm{Hz}$ bis $1000\,\mathrm{kHz}$ bietet die Formel von Francois-Garrison, welche von Van Moll zu einer hybriden Absorptionsgleichung (Gl. (3.11)) vereinfacht wurde. Damit kann die Absorption in Abhängigkeit von der Frequenz f in kHz, dem pH-Wert (gemessen auf der NBS-Skala), dem Salzgehalt S und der Temperatur T sowie der aktuellen Tiefe d_z, über Gl. (3.11) für den begrenzten Parameterbereich in Tabelle 3.2) bestimmt werden. Van Moll trennt dazu auch die Hauptbestandteile in die Bereiche der Borsäure (Gl. (3.12)), des Magnesiumsulfats (Gl. (3.13)) und reinem Wasser (3.14).

$$\alpha = \alpha_1 + \alpha_2 + \alpha_3 \tag{3.11}$$

$$\alpha_1 = 0.101\frac{f_1f^2}{f_1^2+f^2}e^{(pH-S)/0.57} \tag{3.12}$$

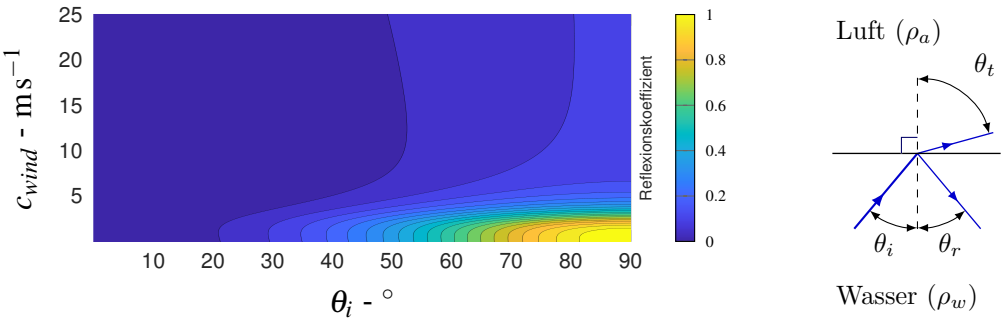
$$\alpha_2 = 0.56\left(1+\frac{T}{76}\right)\left(\frac{S}{35}\right)\left(\frac{f_2f^2}{f_2^2+f^2}\right)e^{-(z/4.9)} \tag{3.13}$$

$$\alpha_3 = A_3P_3f^2 \tag{3.14}$$

$$f_1 = 0.91\left(\frac{S}{35}\right)^{0.5}e^{T/33} \tag{3.15}$$

$$A_3 = \begin{cases} 4.937\,10^{-4} - 2.59\,10^{-5}T \\ +9.11\,10^{-7}T^2 - 1.5\,10^{-8}T^3 & T \leq 20°C \\ 3.964\,10^{-4} - 1.146\,10^{-5}T \\ +1.45\,10^{-7}T^2 - 6.5\,10^{-10}T^3 & T > 20°C \end{cases} \tag{3.16}$$

$$P_3 = 1 - 3.83\,10^{-2}z + 4.9\,10^{-4}z^2 \tag{3.17}$$



(a) Oberflächenreflexionskoeffizient R_b **(b)** Refl.-Schema

Abbildung 3.8: Der Oberflächenreflexionskoeffizient R_b als Kontur-Diagramm mit schematischer Abbildung der Winkel am Übergang von Wasser zu Luft

3.2.4.3 Signalverluste an den Grenzflächen

Die natürliche Begrenzung des Unterwasserkanals wirkt sich nicht nur auf den Streuverlust, sondern auch auf die Mehrwegeausbreitung durch Reflexion an den Grenzflächen (Seeoberfläche und Seegrund) aus. Beide Grenzflächen sind dazu mehrfach im Fokus von Forschung und Entwicklung untersucht und formalisiert worden:

Oberflächenverlust: Die See und damit auch die Wasseroberfläche ist ständig in Bewegung. Dabei umfasst die Oberfläche Muster von Wellen, Strömungen und Strudeln. Ein statisches Abbild dieser Oberfläche wird im Folgenden als Oberflächenprofil bezeichnet.

Die Dynamik von Oberflächenwellen wird durch das Zusammenspiel von Gravitation, Kapillarkräften und Reibung bestimmt. Eine grundlegende lineare Wasserwellengleichung zur Beschreibung der Wellendynamik in Form einer partiellen Differenzialgleichung wurde bereits im 19. Jahrhundert von Lord Rayleigh entwickelt [134]. Eine analytische Lösung dieser Gleichung ist aber

Tabelle 3.2: Gültigkeitsbereich der Absorption nach Moll [133]

Name	Parameter	Bereich			Einheit
Frequenz	f	0.160	-	650.00	kHz
PH-Wert	pH	7.690	-	8.18	NBS-Skala
Salzgehalt	S	8.000	-	40.50	g/kg
Temperatur	T	-1.750	-	22.00	°C
Wassertiefe	d_z	0.013	-	3.35	km

nur für einfache Randbedingungen möglich, weshalb für komplexere Szenarien numerische Verfahren eingesetzt werden.

Bei der Unterwasserkanalsimulation der Oberfläche stehen sich der Genauigkeit der Oberflächenbeschreibung, die Generalität der Aussage und die Komplexität der Berechnung gegenüber. Das heißt, je genauer die Oberflächenmodellierung ist, desto aussagekräftiger ist das Ergebnis, aber beschränkt auf genau das spezifische Szenario. Es gilt daher, ein Gleichgewicht zwischen der Modellgenauigkeit und der Effizienz der Berechnung zu finden, um realitätsnahe Ergebnisse zu erzielen, die dennoch praktisch anwendbar sind. Die Oberflächenwellen in Form der Oberflächenbeschaffenheit spielen im Bereich der akustischen Unterwasserkommunikation eine zentrale Rolle, insbesondere in Bezug auf zwei Aspekte der Signalausbreitung: den Oberflächenverlust SL und die Doppler-Spreizung Δf_d. Beide Eigenschaften beeinflussen unmittelbar die Signalstärke, den Signalpfad, die Signalfrequenz und Verzögerung zwischen Quelle und Senke.

Die gängigsten Modelle zur Bestimmung des Oberflächenverlustes sind das Rayleigh-Verlustmodell [135] in Gl. (3.19), das Oberflächenverlustmodell nach Beckmann und Spizzichino [136, 137] in Gl. (3.18) und die in dieser Arbeit genutzte und von Coartes [138] angepasste Beschreibung des Verlustes in Gl. (3.20) für die Anwendung in der Unterwasserakustik.

Die von Coates angepasste Berechnung des Oberflächenverlustes Gl. (3.20) berücksichtigt die Signalfrequenz f und die Windgeschwindigkeit c_{wind} als Parameter für den Oberflächenverlust und unterstützt eine vereinfachte Nutzung des Beckmann-Spizzichino-Modells. Die resultierenden Reflexionskoeffizienten R_s, welche das Verhältnis von einfallender zur reflektierten Schallintensität angibt, können über die Gl. (3.22) in Abhängigkeit vom Einfallwinkel θ_i, der Trägerfrequenz f_c und der Windgeschwindigkeit c_{wind} bestimmt werden. Die Abbildung 3.8a zeigt den Reflexionskoeffizienten R_s für die Trägerfrequenz f_c=10 kHz in Abhängigkeit von Windgeschwindigkeit und Einfallwinkel. Die Abbildung zeigt, dass die Dämpfung des reflektierten Signals mit steigender Windgeschwindigkeit oder dem fallenden Einfallswinkel θ_i rapide zunimmt. Insbesondere für Signale, deren Einfallswinkel den kritischen Winkel unterschreiten, oder bei einer erhöhten Windgeschwindigkeit, ist eine Reflexion nach Coates sehr stark gedämpft

und hat damit weniger Einfluss auf die Signalrekonstruktion am Empfänger.

$$SL_{BS} = -10\log_{10}\left(\frac{1}{1+\frac{4\pi\sigma_h}{\lambda}}\right) \tag{3.18}$$

$$SL_{RL} = -10\log_{10}\left(\frac{4\pi h}{\lambda}\right)^2 \tag{3.19}$$

$$SL_C(\theta_i)\begin{cases} = 10\log\left(\frac{I_2}{I_1}\right) \\ = 10\log\left(\frac{1+\left(\frac{f}{f_1}\right)^2}{1+\left(\frac{f}{f_2}\right)^2}\right) - \left(1+\frac{90-\frac{c_{wind}}{0.514\,m/s}}{60}\right)\cdot\left(\frac{\theta_i}{30}\right)^2 \end{cases} \tag{3.20}$$

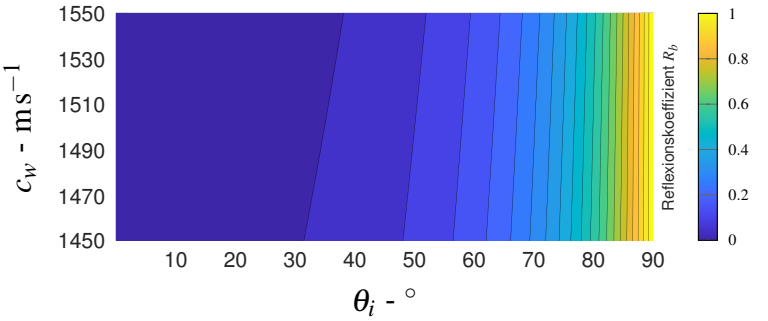
$$f_1 = \sqrt{10}f_2 \text{ und } f_2 = 378w^{-2} = 378\left(\frac{c_w}{0.514\,m/s}\right)^{-2} \tag{3.21}$$

$$R_s(\theta_i) = 10^{\frac{SL_C}{20}} \tag{3.22}$$

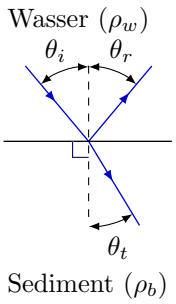
Bodenverlust: Die Bodenbeschaffenheit hat einen maßgeblichen Einfluss auf die Reflexion und Streuung des akustischen Signals über selbigen. Für eine reproduzierbare Bestimmung des Verlustes der Bodenreflexion in Form von Bodenreflexionskoeffizienten wird in dieser Arbeit ein ebenes Bodenprofil, basierend auf der durchschnittlichen Wassertiefe und dem viskoelastischen Modell [123], genutzt. Darauf basierend kann der Bodenreflexionskoeffizient direkt in Abhängigkeit der Sedimenteigenschaften und des Einfallswinkels θ_i berechnet werden.

Die Tabelle 3.3 gibt Beispiele von Sedimenteigenschaften an, die zu den gängigsten Sedimenten im Ozean gehören [123], und beinhaltet den Typ und den Bereich der Dichte sowie die Schallausbreitungsgeschwindigkeit in dem Medium als Longitudinalwellen (P-Wave, Ausbreitung in Richtung der Wellenausbreitung) und zur Vollständigkeit die Transversalwellen (S-Wave, Ausbreitung senkrecht zur Richtung der Wellenausbreitung). Für die Reflexion am Boden müssen aber, da in Wasser keine Transversalwellen auftreten können, nur die Longitudinalwellen im Sediment betrachtet werden. Die Ausbreitungsgeschwindigkeit der Longitudinalwellen im Sediment wird im Weiteren als Schallgeschwindigkeit c_b des Bodens bezeichnet.

Der Reflexionskoeffizient R_b des Bodens wird über die Fresnel-Formel in Gl. (3.23) in Verbindung mit Snells Law in Gl. (3.24) berechnet. Das heißt, der Bodenreflexionskoeffizient R_b wird über die Impedanz des Wassers Z_w und des Bodens Z_b, in Abhängigkeit der jeweiligen Dichte des Wassers ρ_w und des Bodens ρ_b, sowie der jeweiligen Schallgeschwindigkeiten c_w und c_b, des Einfallswinkels θ_i und dem Reflexionswinkels θ_r bestimmt.



(a) Bodenreflexionskoeffizienten R_b



(b) Refl.-Schema

Abbildung 3.9: Kontur-Diagramm des Bodenreflexionskoeffizienten R_b und der schematischen Abbildung der Winkel am Übergang von Wasser zu Sediment

Ein beispielhafter Bodenreflexionskoeffizient R_b ist in der Abbildung 3.9a dargestellt und zeigt R_b in Abhängigkeit der Schallgeschwindigkeit in Wasser c_w in Sediment-nähe im Bereich von $1450\,\mathrm{m/s}$ bis $1550\,\mathrm{m/s}$, einer Bodendichte von $\rho_b = 2\,\mathrm{g/cm^3}$ und einer Ausbreitungsgeschwindigkeit des Schalls im Boden von $c_b = 1800\,\mathrm{m/s}$.

Der Einfallswinkel θ_i bezieht sich dabei auf das Lot des Bodens (siehe Abbildung 3.9b). Die Abbildung zeigt, dass der R_b mit fallendem Einfallswinkel θ_i stark abnimmt, während der Einfluss der Schallgeschwindigkeit des Wassers über dem Sediment in diesem Bereich nur dezente Auswirkungen aufzeigt.

$$R_b(\theta_i) = \left(\frac{Z_w \cdot \cos(\theta_i) - Z_b \cdot \cos(\theta_t)}{Z_w \cdot \cos(\theta_i) + Z_b \cdot \cos(\theta_t)}\right)^2 \quad \text{mit } Z_w = \rho_w \cdot c_w \text{ , } Z_b = \rho_b \cdot c_b \tag{3.23}$$

$$\theta_t = asin\left(\frac{c_w}{c_b} \cdot sin(\theta_i)\right) \tag{3.24}$$

Tabelle 3.3: Physikalische Eigenschaften von Sedimenttypen

Sedimenttyp	**Dichte**			Ausbreitungsgeschwindigkeit **P-Wave**			Ausbreitungsgeschwindigkeit **S-Wave**		
Ton	2.5	-	$2.8\,\mathrm{g\,m^{-3}}$	1500	-	$1800\,\mathrm{m/s}$	800	-	$1000\,\mathrm{m/s}$
Schluff	2.4	-	$2.7\,\mathrm{g\,m^{-3}}$	1200	-	$1800\,\mathrm{m/s}$	600	-	$1000\,\mathrm{m/s}$
Sand	1.5	-	$2.6\,\mathrm{g\,m^{-3}}$	2000	-	$3500\,\mathrm{m/s}$	1000	-	$1800\,\mathrm{m/s}$
Kies	1.5	-	$2.6\,\mathrm{g\,m^{-3}}$	2500	-	$4000\,\mathrm{m/s}$	1500	-	$2500\,\mathrm{m/s}$
Schlamm	2.2	-	$2.6\,\mathrm{g\,m^{-3}}$	1000	-	$1800\,\mathrm{m/s}$	300	-	$800\,\mathrm{m/s}$

3.2.5 Dopplereffekt

In der akustischen Unterwasserkommunikation können Dopplereffekte auftreten, die durch die Relativbewegung zwischen Sender und Empfänger, dem Medium und dessen natürlichen Grenzen induziert werden. Die Dopplereffekte der Unterwasserkommunikation können dabei in Eigen-, Wellen- und Sedimentationsbewegungseffekte sowie thermische Schichtungseffekte unterschieden werden. Diese Dopplereffekte können in jeder Unterwasserumgebung auftreten, und da die Kenntnisse und Wirkung dieser Effekte entscheidend für die Entwicklung von robusten und zuverlässigen Kommunikationssystemen und -protokollen sind, sollten diese bei der Analyse und Auswahl von Kommunikationssystemen berücksichtigt und simuliert werden.

Der Eigenbewegungseffekt entsteht durch die relative Bewegung von Sender oder Empfänger v_{TR}, wobei das übertragene Signal in Abhängigkeit der Frequenz und Schallgeschwindigkeit gestreckt oder gestaucht wird. Beim Wellenbewegungseffekt werden Effekte des sich selbst bewegenden Mediums berücksichtigt, wie die Transversalwellen bzw. Longitudinalwellen (Strömung v_S). Wie bei der Eigenbewegung kommt es zu einer Verschiebung der Frequenz, wenn sich Sender oder Empfänger relativ zur Strömung bewegen. Dieser Effekt erweitert die Gl. (3.25) um die strömungsspezifische Verschiebung der relativen Geschwindigkeit v_S.

Wird eine Schallwelle an der Wasseroberfläche oder dem Boden reflektiert oder gebrochen, ändert sich Ihre Richtung und in Abhängigkeit der Interaktion Ihre Geschwindigkeit. Wird das Wellenprofil weiter als spiegelnd und der Boden als unbeweglich interpretiert, ändert sich der Dopplereffekt über eine Bodenreflexion aber nicht. Der Einfluss der Wasseroberfläche wird über eine zeitlich variierende Wellengeschwindigkeit v_{SURF} abgebildet.

Die Gl. (3.25) bestimmt über Phasenverschiebung in Abhängigkeit der zusätzlichen relativen Geschwindigkeit des Schalls aus Gl. (3.29) und moduliert dieses im Frequenzbereich auf die Kanalübertragungsfunktion (CTF), welche über die Fast Fourier Transformation (FFT) erzeugt wird, summiert über alle Signalpfade $k \in K$ und transformiert diese abschließend über die Inverse Fast Fourier Transformation (IFFT) in die doppler-basierte Kanalimpulsantwort $\mathrm{CIR}_{\mathrm{dop}}$. Die Gl. (3.26) bildet damit die Auswirkungen der Reflexionen an der Oberfläche mittels des Vektors n_{SURF} mit der zeitlich abhängigen Be- oder Entschleunigung sowie die relative Geschwindigkeit der Kommunikationsteilnehmer auf die CIR

ab.

$$\Delta f[k,t] = f \cdot {c_w}^{-1} \cdot v_{TR}[k,t] \text{ mit } f := [0 \quad f_s/2] \tag{3.25}$$

$$CIR_{\text{dop}}[t] = \mathscr{F}^{-1}\left\{\sum_{k=1}^{N} \mathscr{F}\{CIR[k,t]\} \cdot e^{-i\cdot 2\pi \cdot \frac{\Delta f[k,t]}{f_s}}\right\} \tag{3.26}$$

$$v_{TRx}[k,t] = v_{T_x}[t] - v_{R_x}[t] + v_{Sx}[t] + n_{SURF}[k] \cdot v_{SURFx}[t] \tag{3.27}$$

$$v_{TRy}[k,t] = v_{T_y}[t] - v_{R_y}[t] + v_{Sy}[t] + n_{SURF}[k] \cdot v_{SURFy}[t] \tag{3.28}$$

$$v_{TR}[k,t] = \left((v_{TRx}[k,t])^2 + (v_{TRy}[k,t] \cdot \tan(\theta_k))^2\right)^{\frac{1}{2}} \tag{3.29}$$

Thermische Schichtungseffekte, hervorgerufen durch Temperaturschichtungen im Ozean, führen zu vertikalen Geschwindigkeitsgradienten, welche die Ausbreitungsgeschwindigkeit von Schallwellen beeinflussen. Dadurch können Dopplerverschiebungen entstehen, wenn sich Sender oder Empfänger durch diese Schichtungen bewegen. Die Simulation dieser Dopplereffekte erfordert jedoch eine genaue Kenntnis der Temperatur sowie der Bewegungsrichtung und Geschwindigkeit der Objekte und wird im Rahmen dieser Arbeit nicht weiter behandelt.

3.2.6 Umgebungsrauschen

Neben der Kanalimpulsantwort ist das Umgebungsrauschen, charakterisiert durch den Rauschspektralpegel (NSL), ein weiterer wesentlicher Hauptbestandteil des Unterwasserkanals. Die NSL beschreibt die akustische Intensität von Geräuschen relativ zu einer Schallwellen-Referenzintensität von $1\,\mu\text{Pa}$ Schalldruck.

Das Umgebungsrauschen (N) wird in der Unterwasserakustik generell in vier Bereiche mit ihren jeweils dominierenden Frequenzbereichen unterteilt: das von Wind und Seegang abhängige Oberflächengeräusch (N_w), das thermische Rauschen (N_{th}), Schiffsgeräusche (N_{sh}) sowie Geräusche, die durch Erdbeben, Turbulenzen, Stürme und Regen verursacht werden (N_{est}).

Durch eine getrennte Betrachtung der einzelnen Frequenzspektren hat Waite [139] die dominierenden Geräuschquellen identifiziert und empirisch beschrieben.

Erdbeben, Stürme und Turbulenzen sind dominante akustische Quellen für Frequenzen unter $10\,\text{Hz}$ (Gl. (3.30)), während Frequenzen zwischen $10\,\text{Hz}$ und $100\,\text{Hz}$ vom Schiffslärm (Gl. (3.31)) dominiert werden. Der Schiffslärm wurde dabei über die Schiffsverkehrsskala s definiert, wobei $s = 1$ für hohen und $s = 0$

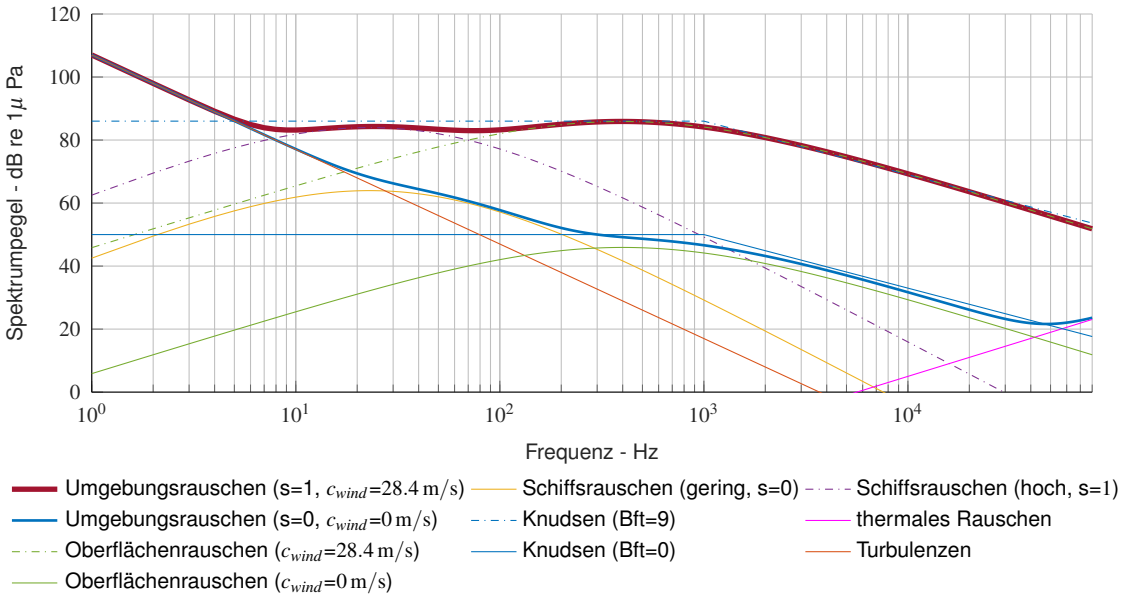


Abbildung 3.10: Berechneter Pegel des Umgebungsrauschens und ausgewählter Kommunikationssignalpegel von $1\,\mathrm{Hz}$ bis $80\,\mathrm{kHz}$.

für geringen Verkehr steht. Das thermische Rauschen (Gl. (3.33)) dominiert den Frequenzbereich ab $100\,\mathrm{kHz}$. Im Frequenzbereich zwischen $500\,\mathrm{Hz}$ und $50\,\mathrm{kHz}$ dominiert das Oberflächenrauschen (Gl. (3.32)), das durch Seegang induziert wird.

$$N_{est} = 17 - 30log(f) \tag{3.30}$$
$$N_{sh} = 40 + 20(s - 0.5) + 26log(f) - 60log(f + 0.03) \tag{3.31}$$
$$N_{w} = 50 + 7.5{c_{wind}}^{0.5} + 20log(f) - 40log(f + 0.44) \tag{3.32}$$
$$N_{th} = -15 + 20log(f) \tag{3.33}$$
$$N = N_{est} + N_{sh} + N_{w} + N_{th} \tag{3.34}$$

Der Seegang kann anhand der Wellenhöhe in Bezug auf die Windgeschwindigkeit unter Verwendung der Beaufort-Windstärkeskala [140] beschrieben werden. Die zugrundeliegende Beziehung zwischen dem Seegang s_{st}, der Windgeschwindigkeit c_{wind} und der Wellenhöhe wird von der World Meteorological Organization (WMO) [141] definiert. Diese wurde von der Douglas See-Skala übernommen und wird in dieser Arbeit, zusammengefasst in Tabelle 3.4, angewendet.

In Abbildung 3.10 ist die spektrale Leistungsdichte (PSD) des Umgebungsrauschens N (Gl. (3.34)) mit dem Oberflächenrauschen für die Seezustände

Beaufort $[bft]$	Bezeichnung	Windgeschwindigkeit $[\mathrm{m\,s^{-1}}]$			Sig. Wellenhöhe $H_{1/3}$ [141] [m]
0	Windstille	0.0	-	0.2	0.0
1	leiser Zug	0.3	-	1.5	0.1
2	leichte Brise	1.6	-	3.3	0.2
3	schwache Brise	3.4	-	5.4	0.6
4	mäßige Brise	5.5	-	7.9	1.0
5	frische Brise	8.0	-	10.7	2.0
6	starker Wind	10.8	-	13.8	3.0
7	steifer Wind	13.9	-	17.1	4.0
8	stürmischer Wind	17.2	-	20.7	5.5
9	Sturm	20.8	-	24.4	6.0
10	schwerer Sturm	24.5	-	28.4	7.0
11	orkanartiger Sturm	28.5	-	32.6	9.0
12	Orkan		ab	32.7	11.5

Tabelle 3.4: Relation zwischen Beaufort, Wind und signifikanter Wellenhöhe

1 und 7, dem Schiffsrauschen aus wenigem und hohem Schiffsverkehr, Turbulenzen und thermischem Rauschen über das Frequenzspektrum von $1\,\mathrm{Hz}$ bis $80\,\mathrm{kHz}$ abgebildet. Neben dem Umgebungsrauschen kann das Kommunikationssignal auch durch ortsspezifisches Rauschen wie Eisbruch [142] oder Snapping-Shrimps [143] gestört werden, welche in dieser Arbeit nicht weiter betrachtet werden.

Bei der Betrachtung der formal beschriebenen dominanten Rauschquellen stand auch vermehrt die Tiefenunabhängig im Fokus von wissenschaftlichen Untersuchungen, da diese dem intuitiven Grundverständnis des Ausbreitungsverlustes entgegen spricht. Dazu wurden Experimente durchgeführt, wie zum Beispiel von Barclay [144], die zeigen, dass die tiefenunabhängigen Annahmen zum Wind induzierten Rauschen halten. Dem entgegen haben Marshall [145] und Morris [146] aufgewiesen, dass sich das Rauschen im niedrigen Frequenzbereich (Schiffsverkehr) in Abhängigkeit der Tiefe ändern kann, aber stark von dem Szenario abhängt. Im Laufe der Zeit haben sich auch die Szenarien auf hoher See geändert. In der heutigen Zeit wird die Umgebung zusätzlich von weiteren Geräuschquellen gestört, wie Pipelines, Bohrtürme (Inseln), erhöhter Schiffslärm durch Motoren, Flugzeuge und weitere, was in [147] allgemein betrachtet und eingeführt wird. Die geänderten Umgebungsbedingen sollten daher bei der Auslegung von zukünftigen Systemen und der Modellierung von Rauschen je nach Einsatzszenario in Betracht gezogen werden.

3.2.7 Richtcharakteristik

Die Schallausbreitung wird auch in Abhängigkeit der Position und Lage von Quelle und Senke durch die genutzte Hardware in Form der Transducer beeinflusst. Ein Transducer ist eine Vorrichtung, die eine physikalische Größe in eine andere transferiert. Dabei formt ein Transducer, eine nichtelektrische Größe, wie in der akustischen Unterwasserkommunikation, Schalldruck, in ein elektrisches Signal um und umgekehrt. Damit ermöglicht der Transducer die Erfassung, Messung und Verarbeitung der physikalischen Größe durch elektronische Systeme. Die Transducer zur akustischen Unterwasserkommunikation kann mittels spezifischen Eigenschaften wie der Sendeempfindlichkeit TVR, Empfangsempfindlichkeit OCR sowie einer winkelabhängigen vertikalen Abstrahlcharakteristik, welche vom genutzten Frequenzbereich abhängen, beschrieben werden.

In dieser Arbeit wird die Wahl der Transducer auf den *M27-930 FFR* [148] im Bereich von $7\,\mathrm{kHz}$ bis $17\,\mathrm{kHz}$ und den T218 von Neptun [149] im Frequenzbereich von $16\,\mathrm{kHz}$ bis $30\,\mathrm{kHz}$ beschränkt.

Tabelle 3.5: Sende- und Empfangssensitivität

Transducer	TVR	OCR	f		
M27-930 [148]	$134.0\,\mathrm{dB\ re\ \mu Pa/V@1m}$	$-183\,\mathrm{dB\ re\ 1V/\mu Pa}$	$7\,\mathrm{kHz}$	-	$17\,\mathrm{kHz}$
T218 [149]	$151.0\,\mathrm{dB\ re\ \mu Pa/V@1m}$	$-183\,\mathrm{dB\ re\ 1V/\mu Pa}$	$16\,\mathrm{kHz}$	-	$30\,\mathrm{kHz}$

Die Abbildung 3.11 zeigt eine interpolierte vertikale Abstrahlcharakteristik auf Basis der Datenblätter, welche die winkelabhängige Dämpfung des Sendesignals der ausgewählten Transducer vergleicht. Die Auswahl der Transducer hängt von den jeweiligen Spezifikationen und Anforderungen des Kommunikationssystems, dem Frequenzbereich und der Bandbreite ab. Die resultierende Dämpfung durch den Transducer in dB in Relation zum Austrittswinkel θ wird im folgenden über $D(\theta)$ abgebildet.

Beide Transducer haben Frequenz-spezifische Eigenschaften und Abstrahlverhalten, die Ihre Anwendung beschränken. Der T218 hat starke Dämpfungseigenschaften bei einer ungerichteten Kommunikation, was Ihn für die horizontale oder vertikale Kommunikation und nur mit bekannter Empfängerposition auch in der schrägen Kommunikation bedingt einsetzbar macht. Der M27-930 hingegen ist ein omnidirektionaler Transducer, der kaum gedämpft wird und kann vorzugsweise bei unbekannter Empfängerposition genutzt werden. Ein Vergleich der Spektrumpegel ist in Abbildung 3.12 abgebildet und zeigt, dass der Signal-Rausch-Verhältnis (SNR) von der Frequenz, Entfernung und Ausrichtung der Transducer zwischen Sender und Empfänger abhängt. Es zeigt auch,

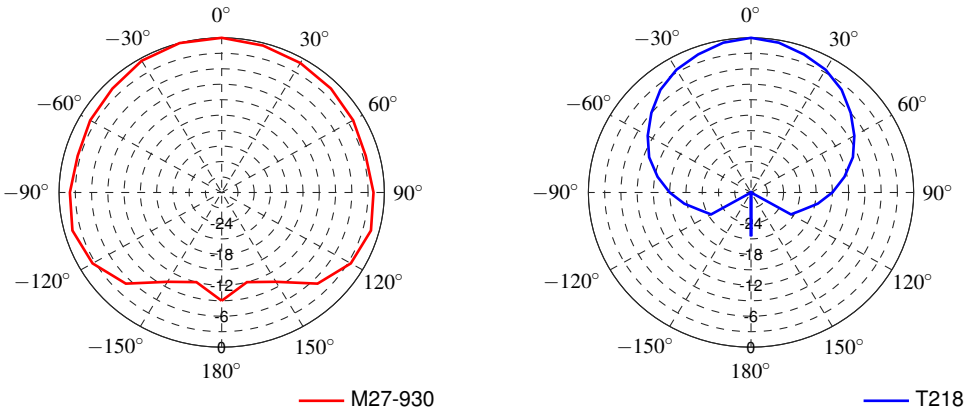


Abbildung 3.11: Vertikale Richtcharakteristik des M27-930 [148] und T218 [149]

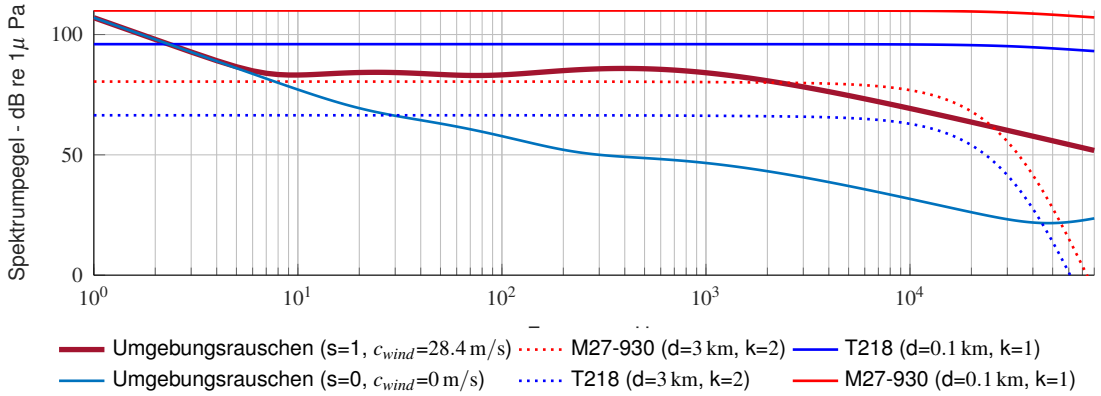


Abbildung 3.12: Berechneter Pegel des Kommunikationssignals und des Umgebungsrauschens für 1 Hz bis 80 kHz

dass zusätzlich die Kommunikationsentfernung und Art des Gewässers über die Absorption und Tiefe einen wesentlichen Einfluss auf den SNR haben können.

3.3 Modellierung des Unterwasserkanals

Bei der Modellierung eines Unterwasserkanals wird der verfügbare Analyseraum durch Abwägen der Genauigkeit im Kontrast zur benötigten Simulationszeit auf ein Set von repräsentativen Testpunkten reduziert. Diese Testpunkte bilden das Fundament für die Untersuchung, Bewertung und Vergleich von Kommunikationssystemen hinsichtlich der definierten akustischen Eigenschaften des Unterwasserkanals.

Dazu werden im Folgenden die Testpunkte im Analyseraum definiert, um eine umfassende Abdeckung der relevanten Bereiche zu gewährleisten. Anschließend werden für diese Testpunkte die akustischen Eigenschaften des Unterwasserkanals mittels des Raytracers BELLHOP modelliert. Dies beinhaltet die Berechnung der Kanalimpulsantwort, wofür im Vorfeld alle Randbedingungen berechnet und BELLHOP zur Verfügung gestellt werden. In der Nachbearbeitung werden die generierten CIR um die Effekte wie den Dopplereffekt, die hardwarespezifische Richtcharakteristik und das zeitlich variable Verhalten erweitert.

Um eine realistische Modellierung zu gewährleisten, wird abschließend das allgemeine Unterwasserrauschen berücksichtigt. Dies erfolgt durch einen im Folgenden eingeführten FIR-Filteransatz, mit dem ein Unterwasserrauschen gemäß dem definierten Rauschpegel für die numerische Simulation erzeugt oder auf Hardware synthetisiert direkt genutzt werden kann.

3.3.1 Testpunkte

Der Unterwasserkanal und die Kommunikationssysteme selbst besitzen Eigenschaften, die mithilfe von unterschiedlichen Parametern beschrieben werden können. Die Kombination all dieser möglichen N Parameter P spannen den Parameterraum $\mathscr{P} = P_1 \times P_2 \times \cdots \times P_n$ auf. Der im Folgenden genutzte Begriff „Analyseraum“ beschreibt die definierten Bereiche des Parameterraums, die für eine Analyse von Interesse sind.

Der Analyseraum wird durch festgelegte Ziele, Fragestellungen und Einschränkungen durch die Hardware und Umgebung bestimmt und variiert je nach Kontext und Anwendungsfall. Ein spezifisches Set von Parameterwerten des Analyseraums definiert im Folgenden einen Testpunkt und beschreibt den zu simulierenden Kanal und dessen Einschränkungen durch die Hardware und Eigenschaften basierend auf der Umgebung.

Dazu gehören die durchschnittliche Bodentiefe, der saisonale Bereich der Windgeschwindigkeiten bzw. Seegänge und mögliche Positionskombinationen von Sender und Empfänger wie auch Trajektorien, die den Analyseraum erheblich eingrenzen können. Zusätzlich werden die maximalen Geschwindigkeiten und auch Strömungsgeschwindigkeiten zur Bestimmung der Testsphären benötigt. Parameter der Kommunikationssysteme, welche die Eigenschaften des Kanals beeinflussen, wie die nutzbaren Trägerfrequenzen und Bandbreiten, schränken den Analyseraum ebenfalls ein. Soll die Analyse eines Systems auch einen längeren Zeitraum abdecken, müssen langfristig ändernde Parameter wie Salzgehalt, Wassertemperatur und andere, in dieser Arbeit als nicht veränderlich angenommen, im Analyseraum abgebildet werden.

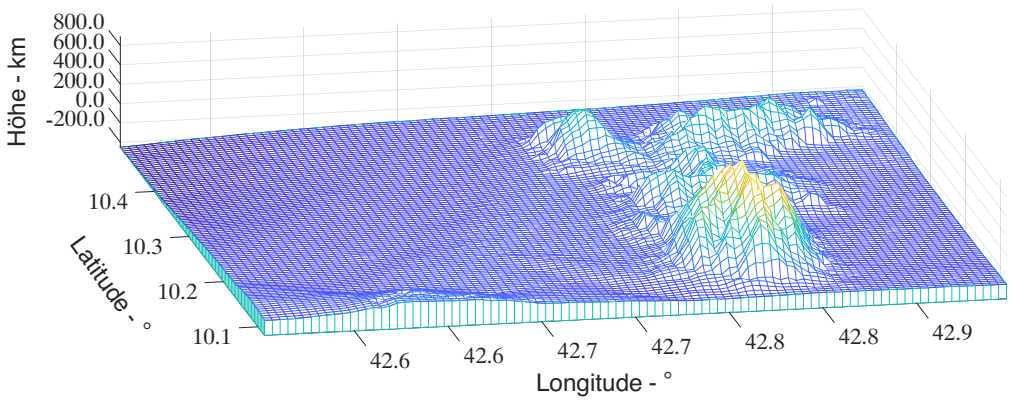
Für die benötigten Umgebungsdaten ermöglichen die NOAA und General Bathymetric Chart of the Oceans (GEBCO) einen freien Zugriff auf gesammelte Messdaten[4], die es erlauben, die SSP und Bathymetrie (Bodentopografie) für den Bereich der Analyse abzufragen oder zu berechnen.

Die gewonnenen Daten sind in den Abbildungen 3.13 bis 3.15 für die experimentellen Umgebungen in dieser Arbeit abgebildet. Die Umgebung, mit der Bezeichnung als *AComm10*, basiert auf der Umgebung des North Atlantic Treaty Organization (NATO) ACommsNet10 Experiment in der Nähe von Pionosa (Italien) und steht stellvertretend für flache Gewässer mit einer durchschnittlichen Wassertiefe von $d_z \approx 93\,\mathrm{m}$. *AComm13* hingegen basiert auf dem Experiment NATO CommsNet13 und repräsentiert tiefe Gewässer mit einer Wassertiefe von $d_z \approx 2619\,\mathrm{m}$ in der Nähe der Küste von Palmaria (Italien). Für die Repräsentation von mitteltiefen Gewässern, im folgenden mit *Trond* bezeichnet, mit einer Wassertiefe von $d_z \approx 501\,\mathrm{m}$ wird die Umgebung nahe der Küste von Trondheim (Norwegen) genutzt. Die Abbildungen Abbildungen 3.13a, 3.14a, und 3.15a zeigen für die Umgebungen jeweils die Topologie als 3D-Konturlinien-Profil, die Bathymetrie (Bodentopographie) und SSP in Abbildungen 3.13c, 3.14c, und 3.15c, basierend auf dem markierten Bereich in Abbildungen 3.13b, 3.14b, und 3.15b.

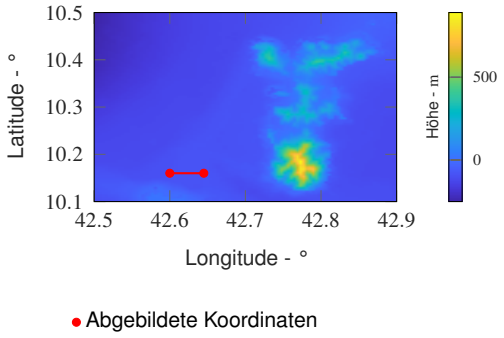
Die Auswahl der Testpunkte im Parameterraum ist ein Kompromiss zwischen vollständiger Abdeckung und schneller Simulation. Für die Tiefenpunkte des Senders und Empfängers wird das Standard-Tiefenprofil in der World Ocean Atlas ($d_z \in \mathbb{D}_z$), der wissenschaftlichen Datenbank (NOAA,Gl. (3.35)), genutzt.

$$\mathbb{D}_z := \{10, 20, 30, 50{:}25{:}150, 200, 250, 300{:}100{:}1500, 1750, 2000{:}500{:}5500\} \quad (3.35)$$

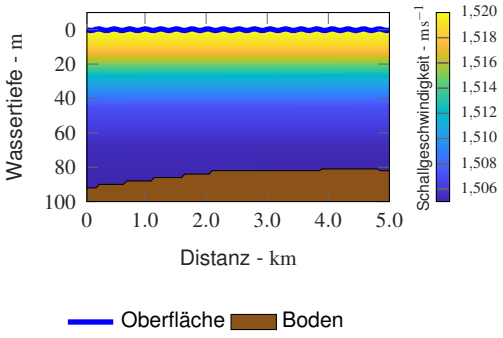
[4] Alternativ können die Daten auch über Datenbanken (zum Beispiel: `http://staff.washington.edu/dushaw/WOA/`) mithilfe der zur Verfügung gestellten Funktionen extrahiert werden.



(a) 3D-Konturlinien-Profil



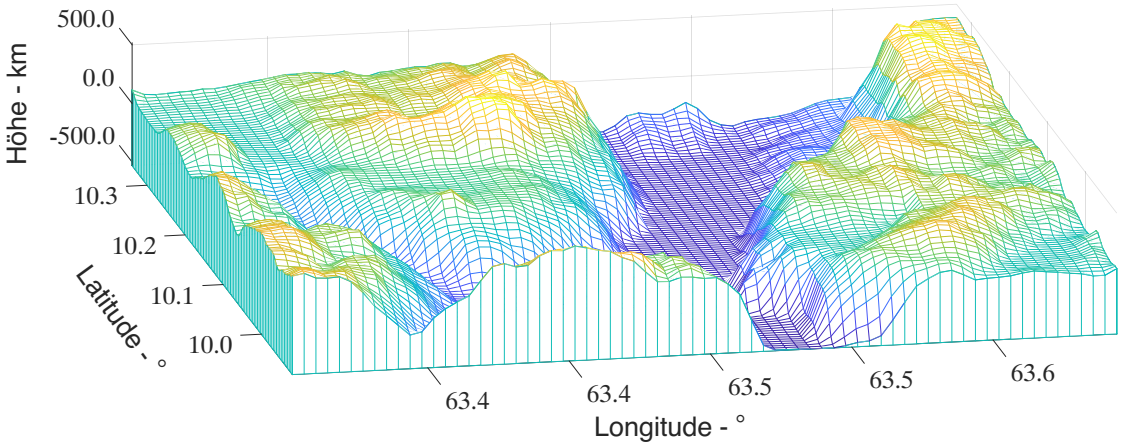
(b) 2D-Tiefenprofil mit Markierung (rot)



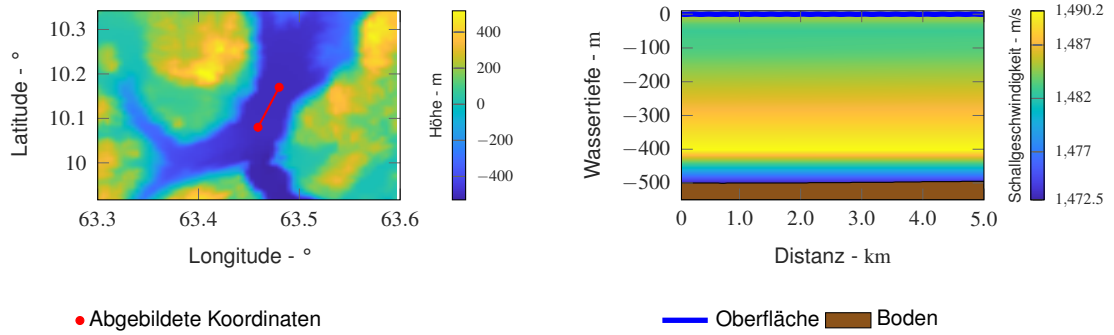
(c) Bathymetrie mit SSP

Abbildung 3.13: *AComm10*-Umgebung basierend auf [150] (Insel Palmaria)

Die Messpunkte sind über alle Bereiche entsprechend der verteilten Auflösung angeordnet. Die Verteilung in der Epilimnion und der saisonalen Thermokline, wo sich der Geschwindigkeitsgradient stark ändert, ist daher wesentlich dichter als in der konstant variierenden Hypolimnion, wo es ohne äußere Einflüsse keine signifikanten Änderungen des Gradienten gibt. In dieser Arbeit ist die räumliche Distanz zwischen den Testpunkten auf $200\,\mathrm{m}$ festgelegt. Die räumliche Distanz zwischen den Testpunkten ist in dieser Arbeit auf $200\,\mathrm{m}$ festgelegt. Um jeden dieser Testpunkte variiert jeweils die Position von Sender und Empfänger und die Wassertiefe in Abhängigkeit des Seegangs zufällig. Die Testbereiche um den Testpunkt werden mit zufälligen Werten, beschränkt über die maximale Geschwindigkeit, über die zeitlich variierende Kanalmatrix abgebildet. Die Abbildung Abbildung 3.16 zeigt dazu die räumliche Abbildung der Testpunkte bezogen auf die Relation zwischen Sender und Empfänger. Diese Testpunkte werden, zusätzlich zur räumlichen Beschreibung, mehrdimensional über den



(a) 3D-Konturlinien-Profil



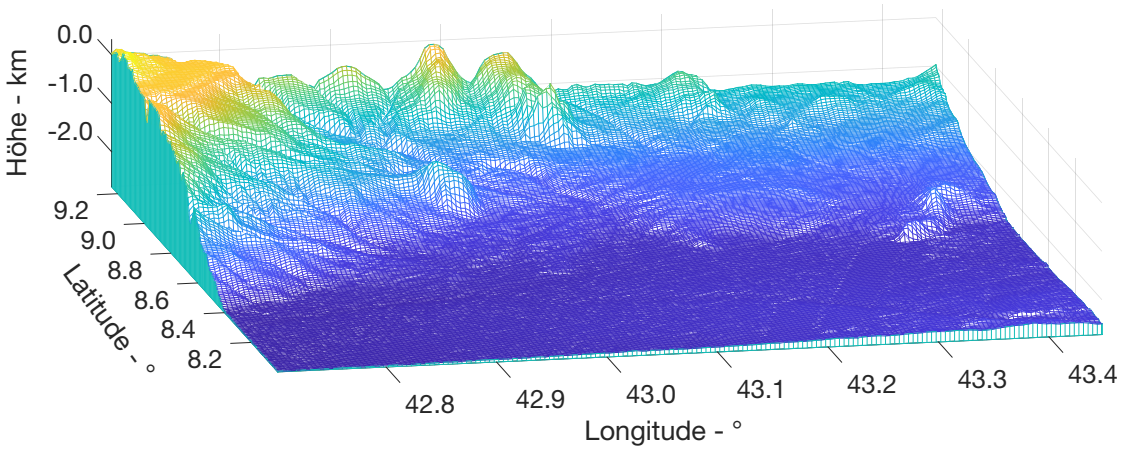
(b) 2D-Tiefenprofil mit Markierung (rot)

(c) Bathymetrie mit SSP

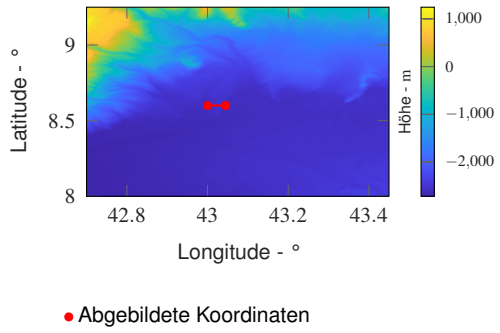
Abbildung 3.14: *Trond*-Umgebung basierend auf [150] (Trondheim)

Seegang, Trägerfrequenz, Bandbreite und ggf. über Salzgehalt, Temperatur und den veränderlichen SSP beschrieben.

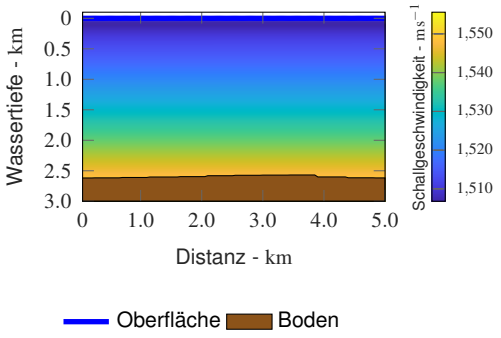
Für die Analyse von Kommunikationssystemen für einen Bereich des Seegangs von $0\,\mathrm{Bft}$ bis $9\,\mathrm{Bft}$ mit festem Salzgehalt, Temperatur, SSP und einer Auswahl von drei Trägerfrequenzen ($f_c := \{10\,\mathrm{kHz}, 12\,\mathrm{kHz}, 20\,\mathrm{kHz}\}$) und zwei Bandbreiten ($B := \{5\,\mathrm{kHz}, 1\,\mathrm{kHz}\}$) werden für das *AComm10*-Umgebung 22500, für das *Trond*-Umgebung 129600 und für das *AComm13*-Umgebung sogar 608400 Kanalmatrizen benötigt, deren Größe von der zeitlichen Auflösung und der abzubildenden Dauer abhängt.



(a) 3D-Konturlinien-Profil



(b) 2D-Tiefenprofil mit Markierung (rot)



(c) Bathymetrie mit SSP

Abbildung 3.15: *AComm13*-Umgebung basierend auf [150] (Insel Palmaria)

3.3.2 Modellierung der Kanalimpulseantwort (CIR)

Für die Modellierung des Unterwasserkanals muss für jeden Testpunkt im vordefinierten Analyseraum eine Systemantwort in Form der Kanalimpulseantwort (CIR) erstellt werden, welche die Eigenschaften des Unterwasserkanals abbildet. Eine der gängigsten Methoden zur Bestimmung der Verzögerungen und Amplituden der Mehrwegausbreitung ist BELLHOP.

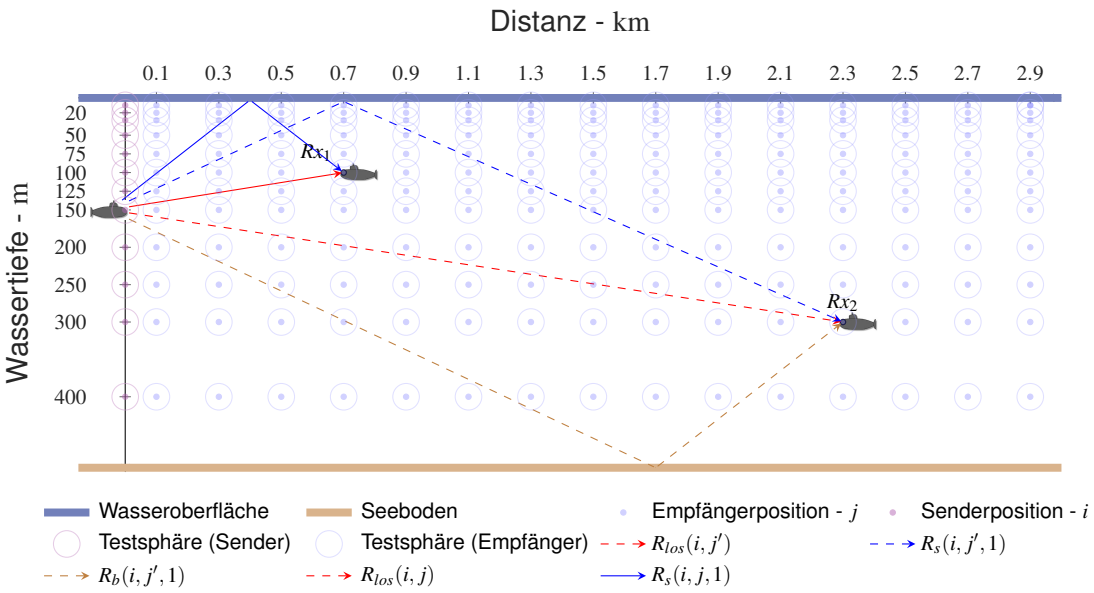


Abbildung 3.16: Positionsbezogene Testpunkte im Analyseraums

3.3.2.1 Modellierung mit BELLHOP

Eine Kanalimpulseantwort (CIR) kann grundsätzlich mit BELLHOP, dem etablierten Kanalschätzer im Bereich der akustischen Unterwasserkommunikation, erzeugt werden. Für eine detaillierte Erstellung einer CIR benötigt BELLHOP die Informationen über die Signalparameter und Kanaleigenschaften in spezifisch definierten Dateiformat [96]. Dazu werden ein Schallgeschwindigkeitsprofil (Geschwindigkeitsprofil (SSP)), die Beschreibung der Umgebungs- und Simulationsparameter, Oberflächen- und Bodenbeschaffenheit, die als flach angenommen wird (soweit nicht auf andere Art definiert), die Reflexionskoeffizienten der Oberfläche und des Bodens für alle Einfallswinkel, beschränkt auf den aktuell festgelegten Analyseraum (Bodendichte, P-Wave, Windgeschwindigkeit, SSP, Frequenz) sowie das Abstrahlverhalten des zugeordneten Transducer als gerichtetes Profil benötigt.

Die Eigenschaften der Simulationsparameter sind ausführlich in der Dokumentation [96] beschrieben. Die Abbildung 3.17 zeigt den Generierungsprozess einer CIR und verweist auf den benötigten Dateityp und zusätzlich auf die im Anhang genutzten Funktionen (Listing A.8 – Listing A.12), welche im Rahmen dieser Arbeit zur Verwendung von BELLHOP geschrieben wurden und welche nicht native zur Verfügung gestellt werden. Die Nutzung der externen Daten, dem Schreiben und dann Lesen und Ausführen mittels BELLHOP schlägt sich in der

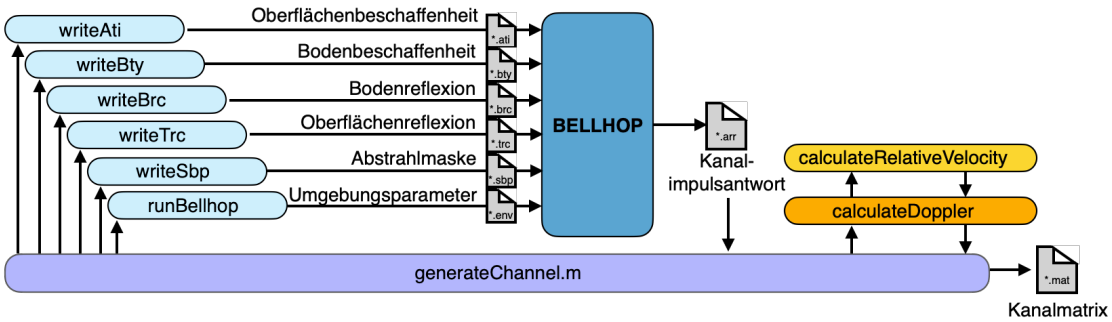


Abbildung 3.17: Arbeitsablauf zur Generierung der Kanalmatrix

benötigten Simulationszeit nieder. Abhängig von der simulierten Umgebung, der Anzahl von erzeugten Strahlen des Raytracers und der resultierenden Pfade variiert die benötigte Zeit von BELLHOP zur Erstellung einer CIR.

Nach der Generierung der Amplituden und Laufzeiten der Signalpfade kann in einem Nachbearbeitungsschritt der Dopplereffekt (*calculateDoppler* in Listing A.12) und im Falle von digitalen Systemen ein Diskretisieren der Kanalimpulseantwort (CIR) auf die Abtastrate f_s für die weitere Nutzung erfolgen.

3.3.2.2 Modellierung mit dem geometrischen Modell

Für eine beschleunigte Umsetzung und Anwendung für die Analyseraumreduktion in Kapitel 6, wurde in dieser Arbeit ein geometrisches Kanalmodell entwickelt, erstmals vereinfacht eingeführt in [151], welches ursprünglich zur Bestimmung von optimalen Paketlängen und zur Reduktion von Testpunkten [152] verwendet wurde. Nach einer Überarbeitung des Modells ermöglicht dieses nun nicht nur eine Reduktion von Testpunkten, sondern auch eine MATLAB-integrierte Kanalabschätzung in Form einer Kanalimpulsmatrix mit einer signifikant schnelleren Performanz.

Der Ansatz des geometrischen Kanalmodells ermöglicht zudem eine Anwendung von alternativen Verlustmodellen, die optional ersetzt oder zusätzlich verglichen werden können. Zudem lassen sich über das verbesserte geometrische Modell die Zeitpunkte der Interaktion mit dem Seeboden und der Wasseroberfläche bestimmen und das Ergebnis in eine Kanalimpulsmatrix überführen. Sie kann zusätzlich die Angabe einer Zeitvarianten-Wellengeschwindigkeit zur Optimierung des Dopplereffekts verwenden.

Damit kann in ihrer Gesamtheit eine zeitliche und ortsspezifische Kanalimpulsantwortmatrix H für jeden festgelegten Testpunkt im definierten Analyseraum bestimmt und analysiert werden.

Für eine reproduzierbare und zeitsparende Analyse wurden die Kanalimpulsantworten automatisiert über das Skript in Listing A.8 für die drei See-Experiment-Umgebungen (*AComm10*, *AComm13* und *Trond*) generiert und für Testwiederholungen vorgehalten.

Die Generierung der Kanalimpulseantwort (CIR) basiert, je nach Methode der Modellierung, entweder auf der Funktion *runBellhop* in Listing A.11 mittels BELLHOP oder auf der geometrisch basierten Umsetzung in *runGeoEstimation* in Listing A.13. Die Funktion *calculateRelativeVelocity* verwendet die Gleichungen 3.27 bis 3.27, um die relative Geschwindigkeitsänderung des Schalls zu bestimmen, während die Funktion *calculateDoppler* zur Bestimmung des Pfad-basierten Dopplers in Listing A.12 genutzt wird.

Die Berechnung des Oberflächen- und Bodenverlustes werden über die symbolischen Funktionen *SLRC* (Surface Loss Reflection Coeffizient, basierend auf Gl. (3.20)) und *BLRC* (Bottom Loss Reflection Coeffizient, basierend auf Gl. (3.24)), sowie in BELLHOP über die Funktionen *writeBCR* (Listing A.15) und *writeTCR* (Listing A.14) bereitgestellt.

Im Folgenden wird angenommen, dass der Seegrund und die Seeoberfläche spiegelnde Eigenschaften haben und die durchschnittliche Wassertiefe sowie Sender- und Empfängerpositionen T_i und R_j gegeben sind. Damit können über das geometrische Modell die Vektoren (Gl. (3.36) und Gl. (3.37)) und damit die Länge der jeweiligen Signalpfade in Abhängigkeit der Anzahl von Reflexionen m das direkte Kommunikationssignal (R_{los}, Gl. (3.38)), die Reflexion an der Oberfläche R_s in Gl. (3.40) und am Boden R_b in Gl. (3.39) berechnet werden. Der geometrische Ansatz dafür ist schematisch in Abbildung 3.18 veranschaulicht und zeigt die definierten Vektoren aus Gl. (3.36) and (3.37).

$$T_i = \begin{bmatrix} T_{i,x} \\ T_{i,y} \\ T_{i,z} \end{bmatrix} R_j = \begin{bmatrix} R_{j,x} \\ R_{j,y} \\ R_{j,z} \end{bmatrix} h = \begin{bmatrix} 0 \\ 0 \\ d_z \end{bmatrix} \text{ mit } T_i, R_j, h \in \mathbb{R}^3 \tag{3.36}$$

$$r_{(0,i,j)} = \begin{bmatrix} T_{i,x} - R_{j,x} \\ T_{i,y} - R_{j,y} \\ 0 \end{bmatrix}, r_{(1,i,j)} = \begin{bmatrix} 0 \\ 0 \\ T_{i,z} - R_{j,z} \end{bmatrix}, r_{(2,i,j)} = \begin{bmatrix} 0 \\ 0 \\ R_{j,z} \end{bmatrix} \text{ mit } r \in \mathbb{R}^3 \tag{3.37}$$

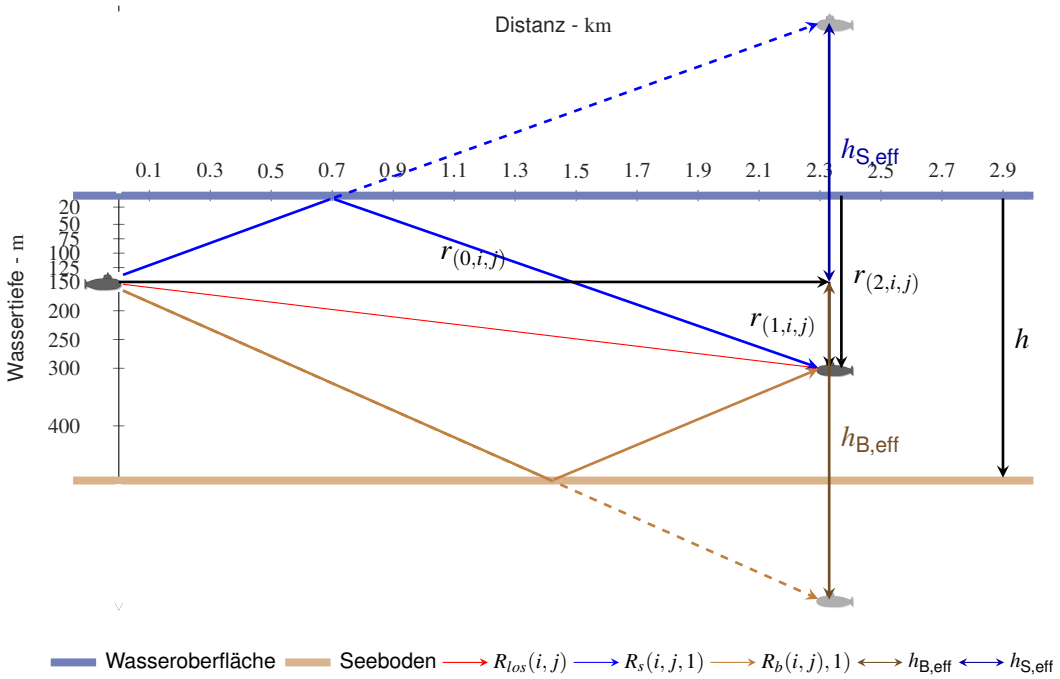


Abbildung 3.18: Schemata des geometrischen Modells

$$R_{los}(i,j)=|r_{(0,i,j)}+r_{(1,i,j)}| \tag{3.38}$$

$$R_s(i,j,m)=|r_{(0,i,j)}+r_{(1,i,j)}-2\cdot(m\,mod2)\cdot r_{(2,i,j)}-h\cdot(m-(m\,mod2)))| \tag{3.39}$$

$$R_b(i,j,m)=|r_{(0,i,j)}+r_{(1,i,j)}+2\cdot(m\,mod2)\cdot(h-r_{(2,i,j)})+h\cdot(m-(m\,mod2))| \tag{3.40}$$

$$\theta_{LOS}(i,j)=\tan^{-1}\left(\frac{|r_{(1,i,j)}|}{|r_{(0,i,j)}|}\right) \tag{3.41}$$

$$h_{\text{S,eff}}=\begin{array}{l}|r_{(2,i,j)}-r_{(1,i,j)}|+(m-1)\cdot|h| \\ +(1-(m\,mod2))\cdot|h-r_{(2,i,j)}|+(m\,mod2)\cdot|r_{(2,i,j)}|\end{array} \tag{3.42}$$

$$h_{\text{B,eff}}=\begin{array}{l}|h-r_{(2,i,j)}+r_{(1,i,j)}|+(m-1)\cdot|h| \\ +(m\,mod2)\cdot|h-r_{(2,i,j)}|+(1-m\,mod2)\cdot|r_{(2,i,j)}|\end{array} \tag{3.43}$$

$$\theta_t(i,j,m)=\tan^{-1}\left(\frac{h_{\text{t,eff}}}{|r_{(0,i,j)}|}\right) \text{ mit } t:=\{S,B\} \tag{3.44}$$

Für die Bestimmung der Kanalimpulsantwort wird, wie bei der Erzeugung durch BELLHOP, eine Pfad-bezogene Amplitude, die Verzögerungen, die Art und Häufigkeit der Reflexion, der Einfallswinkel an den Grenzflächen und der Austrittswinkel am Transducer benötigt. Durch den geometrischen Ansatz kann die Anzahl der Reflexion m sukzessive erhöht werden, bis eine

Abbruchbedingung erfüllt wird. In dieser Arbeit wird die Abbruchbedingung durch das minimale Verhältnis des LOS-Pfads und dem Pfad der m-ten Reflexion von $-45\,\mathrm{dB}$ festgelegt. Der Austrittswinkel am Transducer (θ_{LOS},$\theta_S(m)$ und $\theta_B(m)$) für die Position i des Senders und j des Empfängers wird über die effektive Höhe $|r_{(1,i,j)}|$, $h_{\mathrm{S,eff}}$ in Gl. (3.42) und $h_{\mathrm{B,eff}}$ in Gl. (3.43) abhängig von der Anzahl der Reflexionen m und der Distanz zwischen Sender und Empfänger $r_{(0,i,j)}$ in Rad mit Gl. (3.41) und 3.44 bestimmt. Mit einer Anpassung der Berechnung des Reflexionskoeffizienten kann dieser direkt über den Austrittswinkel am Transducer bestimmt werden.

Die Amplitude a_i der Kanalimpulsantwort des n-ten Signalpfades kann über die Gl. (3.45) des direkten Pfades zwischen Sender und Empfänger, der nur von der Absorption, Streuung und Dämpfung des Transducer abhängt, berechnet werden. Diese Pfade werden unterschieden in Pfade mit der ersten Reflexion an der Oberfläche (l_S) und mit der ersten Reflexion am Bodens (l_B). Die Amplituden dieser Pfade werden zusätzlich von den Reflexionskoeffizienten des Bodens und der Oberfläche sowie der Anzahl der jeweiligen Reflexionen beeinflusst.

$$a(n)=\begin{cases} 10^{\frac{(-D(\theta_{LOS}(i,j))-PL_{sp}(k,R_{los}(i,j))-PL_{ab}(R_{los}(i,j),fc))}{20}} & \text{für } n=1 \\ 10^{\frac{(-D(\theta_S(i,j,n-1))-PL_{sp}(k,R_s(i,j,n-1))-PL_{ab}(R_s(i,j,n-1),fc))}{20}} \cdot R_s(\theta_S(i,j,n-1))^{\lceil\frac{n-1}{2}\rceil}\cdot R_b(\theta_S(i,j,n-1))^{\lfloor\frac{n-1}{2}\rfloor} & \text{für } 1<n\leq l_S+1 \\ 10^{\frac{(-D(\theta_B(i,j,n-l_S-1))-PL_{sp}(k,R_b(i,j,n-l_S-1))-PL_{ab}(R_b(i,j,n-l_S-1),fc))}{20}} \cdot R_s(\theta_B(i,j,n-l_S-1))^{\lfloor\frac{n-l_S-1}{2}\rfloor} \cdot R_b(\theta_B(i,j,n-l_S-1))^{\lceil\frac{n-l_S-1}{2}\rceil} & \text{sonst} \end{cases} \tag{3.45}$$

Die Verzögerung der Pfade folgt aus der tiefen-bezogenen Schallgeschwindigkeit c_w des Wassers. Unter der Annahme, dass die Schichten gleicher Schallgeschwindigkeit gleich dick sind, im folgenden mit Δy bezeichnet, kann die Verzögerung τ_n in Abhängigkeit der Signalpfadlänge, der Art der ersten Reflexion (Boden, Wasseroberfläche) sowie der Position von Sender und Empfänger über Gl. (3.47), (3.51), und (3.52) bestimmt werden.

Die Berechnung der Verzögerung τ für die LOS in Gl. (3.47) setzt sich aus den komplett durchlaufenen Schichten und den jeweiligen anteiligen Schichten von Sender und Empfänger zusammen. Da die Verzögerung unter den getroffenen Annahmen unabhängig von der Richtung bestimmt werden kann, wird die

Verzögerung über das Minimum der Tiefe von Sender und Empfänger bestimmt.

$$y_1 = \left\{ \left\lceil \frac{min(T_{i,z}, R_{j,z})}{\Delta y} \right\rceil \cdot \Delta y : \Delta y : \left\lfloor \frac{max(T_{i,z}, R_{j,z})}{\Delta y} \right\rfloor \cdot \Delta y - \Delta y \right\} \tag{3.46}$$

$$\begin{aligned} \tau_n = & \frac{\left\lceil \frac{min(T_{i,z}, R_{j,z})}{\Delta y} \right\rceil \cdot \Delta y - min(T_{i,z}, R_{j,z})}{cos(1 - \theta_{LOS}(i,j)) \cdot c_w \left(\left\lfloor \frac{min(T_{i,z}, R_{j,z})}{\Delta y} \right\rfloor \cdot \Delta y \right)} \\ & - \frac{\left\lceil \frac{max(T_{i,z}, R_{j,z})}{\Delta y} \right\rceil \cdot \Delta y - max(T_{i,z}, R_{j,z})}{cos(1 - \theta_{LOS}(i,j)) \cdot c_w \left(\left\lfloor \frac{max(T_{i,z}, R_{j,z})}{\Delta y} \right\rfloor \cdot \Delta y \right)} \\ & + \sum_{y \in y_1} \frac{\Delta y}{cos(1 - \theta_{LOS}(i,j)) \cdot c_w(y)} \end{aligned} \quad \text{für } n = 1 \tag{3.47}$$

Für die Berechnung der Verzögerungen $\tau(n)$ für alle l_S Signalpfade, beginnend über die Oberfläche, wird in Gl. (3.51) abgebildet und beruht auf der Bestimmung der Verzögerung in den durchlaufenen Schichten. Dazu wird die Verzögerung zusammengesetzt aus der benötigten Zeit zwischen dem Sender und der Oberfläche, $n_{ref} - 2$ mal zwischen dem Boden und Oberfläche bei einer Anzahl von n_{ref} Reflexionen sowie der Verzögerung zwischen der Oberfläche und dem Empfänger, für eine ungerade oder dem Boden und Empfänger bei einer geraden Anzahl von Reflexionen n_{ref} berechnet. Durch die Auf- bzw. Abrundungen der Gleichungen, womit die Verzögerung des Signals für das Durchlaufen der kompletten Schicht berechnet wird, müssen noch die anteiligen Verzögerungen innerhalb der Schicht des Empfängers und Senders hinzugefügt werden. Die Verzögerungen $\tau(n)$ für alle restlichen Signalpfade, über den Boden beginnend, sind in Gl. (3.52) entsprechend angepasst formalisiert. Die jeweiligen durchlaufenden Schichten sind über y_i mit $i := \{1,2,3,4\}$ in den Gl. (3.46) and (3.48) bis (3.50) und die Dicke der abstrahierten Thermalschicht mit Δy definiert sind.

Damit können die Kanalimpulsantworten über das geometrische Modell erzeugt werden und wie bei der Verwendung von BELLHOP in einem Nachbearbeitungsschritt der Dopplereffekt (*calculateDoppler* in Listing A.12) und eine Diskretisierung erfolgen.

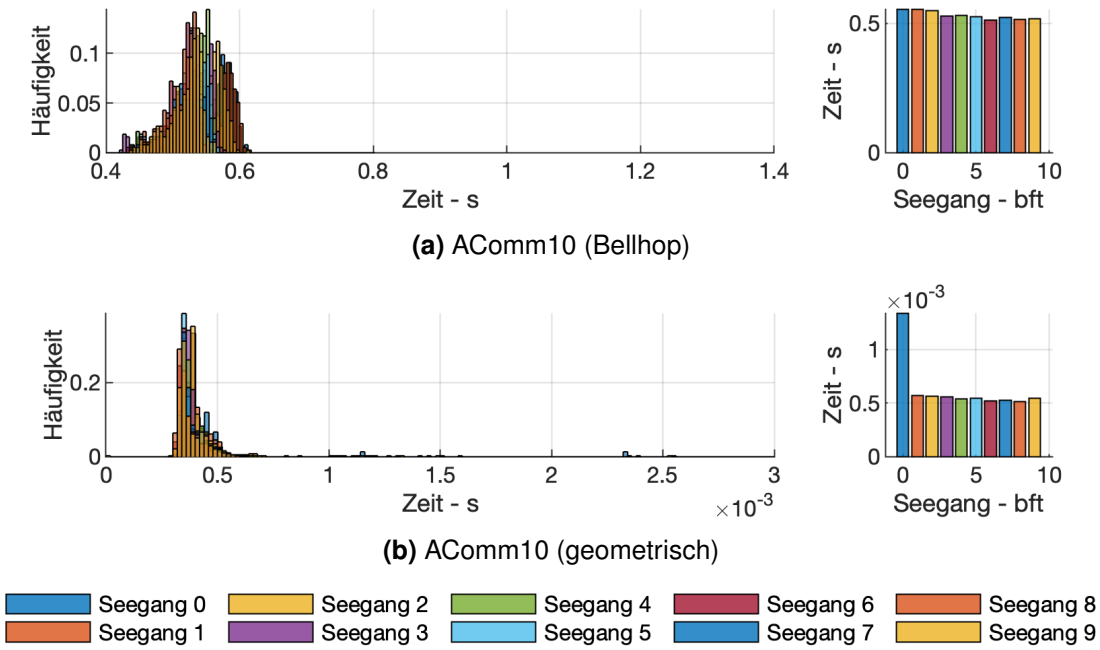
$$y_2 = \left\{ 0 : \Delta y : \left\lfloor \frac{T_{i,z}}{\Delta y} \right\rfloor \cdot \Delta y - \Delta y \right\} \tag{3.48}$$

$$y_3 = \left\{ 0 : \Delta y : \left\lfloor \frac{R_{j,z}}{\Delta y} \right\rfloor \cdot \Delta y - \Delta y \right\} \tag{3.49}$$

$$y_4 = \left\{ 0 : \Delta y : \left\lfloor \frac{h}{\Delta y} \right\rfloor \cdot \Delta y - \Delta y \right\} \tag{3.50}$$

$$\tau_n = \frac{T_{i,z} - \left\lfloor \frac{T_{i,z}}{\Delta y} \right\rfloor \cdot \Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w \left(\left\lfloor \frac{T_{i,z}}{\Delta y} \right\rfloor \cdot \Delta y \right)}$$
$$+ \sum_{y \in y_2} \frac{\Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w(y)}$$
$$+ (n-2) \cdot \sum_{y \in y_4} \frac{\Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w(y)}$$
$$+ ((n-1) mod 2) \cdot \sum_{y \in y_3} \frac{\Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w(y)} \qquad \text{für } 1 < n \leq l_S+1 \quad (3.51)$$
$$+ (1-((n-1) mod 2)) \cdot \sum_{y \in y_4 \setminus y_3} \frac{\Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w(y)}$$
$$+ ((n-1) mod 2) \cdot \frac{R_{j,z} - \left\lceil \frac{R_{j,z}}{\Delta y} \right\rceil \cdot \Delta y}{cos(1-\theta_S(i,j,n-1)) \cdot c_w \left(\left\lfloor \frac{R_{j,z}}{\Delta y} \right\rfloor \cdot \Delta y \right)}$$
$$+ (1-((n-1) mod 2) \frac{\left\lceil \frac{R_{j,z}}{\Delta y} \right\rceil \cdot \Delta y - R_{j,z}}{cos(1-\theta_S(i,j,n-1)) \cdot c_w \left(\left\lfloor \frac{R_{j,z}}{\Delta y} \right\rfloor \cdot \Delta y \right)}$$

$$\tau_n = \frac{\left\lceil \frac{T_{i,z}}{\Delta y} \right\rceil \cdot \Delta y - T_{i,z}}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w \left(\left\lfloor \frac{T_{i,z}}{\Delta y} \right\rfloor \right)}$$
$$+ \sum_{y \in y_4 \setminus y_2} \frac{\Delta y}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w(y)}$$
$$+ (n-2) \cdot \sum_{y \in y_4} \frac{\Delta y}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w(y)}$$
$$+ ((n-1) mod 2) \cdot \sum_{y \in y_4 \setminus y_3} \frac{\Delta y}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w(y)} \qquad \text{für } l_S+1 < n \quad (3.52)$$
$$+ (1-(n-1) mod 2) \cdot \sum_{y \in y_3} \frac{\Delta y}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w(y)}$$
$$+ (1-(n-1) mod 2) \cdot \frac{R_{j,z} - \left\lceil \frac{R_{j,z}}{\Delta y} \right\rceil \cdot \Delta y}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w \left(\left\lfloor \frac{R_{j,z}}{\Delta y} \right\rfloor \cdot \Delta y \right)}$$
$$+ ((n-1) mod 2) \frac{\left\lceil \frac{R_{j,z}}{\Delta y} \right\rceil \cdot \Delta y - R_{j,z}}{cos(1-\theta_B(i,j,n-l_S-1)) \cdot c_w \left(\left\lfloor \frac{R_{j,z}}{\Delta y} \right\rfloor \cdot \Delta y \right)}$$



(a) AComm10 (Bellhop)

(b) AComm10 (geometrisch)

Abbildung 3.19: Zeitliche Auftrittsverteilung der Kanalgenerierung (AComm10)

3.3.2.3 Vergleich der Modellierungsansätze

Für den Vergleich und die Bewertung des verbesserten geometrischen Modells wurden die CIR unter Verwendung identischer Eingabedaten sowie desselben Transducers mit seinem spezifischen Schallstrahlmaske (SBP) für drei See-Experiment-Umgebungen (siehe Abbildungen 3.13 bis 3.15) erzeugt und miteinander verglichen. Dabei wurden die drei Zielgrößen Rechenzeit, Ähnlichkeit und Wertigkeit ermittelt.

Die zur Bestimmung einer CIR benötigte Rechenzeit beschränkt sich auf die methodenspezifischen Funktionen und berücksichtigt nicht die Zeit für die Erstellung zusätzlicher Dateien (z. B. BELLHOP) oder das Speichern der erzeugten Daten. Die Abbildungen 3.19a, 3.20a, und 3.21a zeigen die Verteilung der Rechenzeiten zur Generierung mittels BELLHOP in Abhängigkeit vom Seegang. Zum Vergleich sind die entsprechenden Ergebnisse des verbesserten geometrischen Kanalmodells in den Abbildungen 3.19b, 3.20b, und 3.21b dargestellt.

Aus der Verteilung der benötigten Rechenzeit zur Erzeugung der Kanalimpulsantworten ergeben sich drei wesentliche Aspekte: Gaußverteilung mit Schwanz, Seegang-abhängige Rechenzeit und die signifikante schnelle Berechnung über das geometrische Kanalmodell.

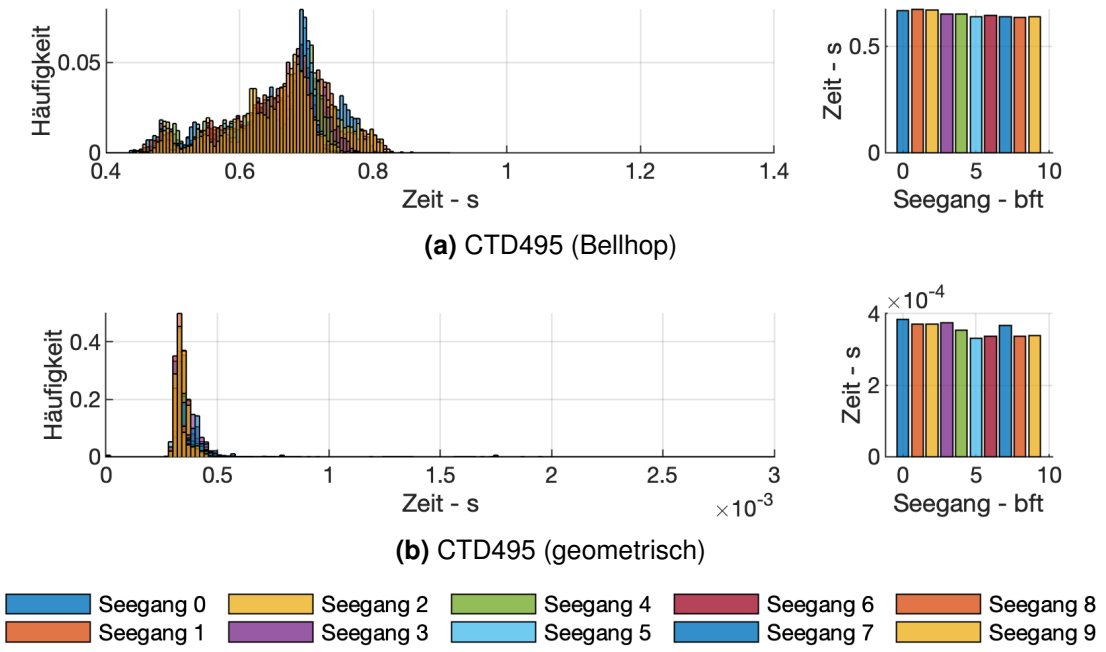
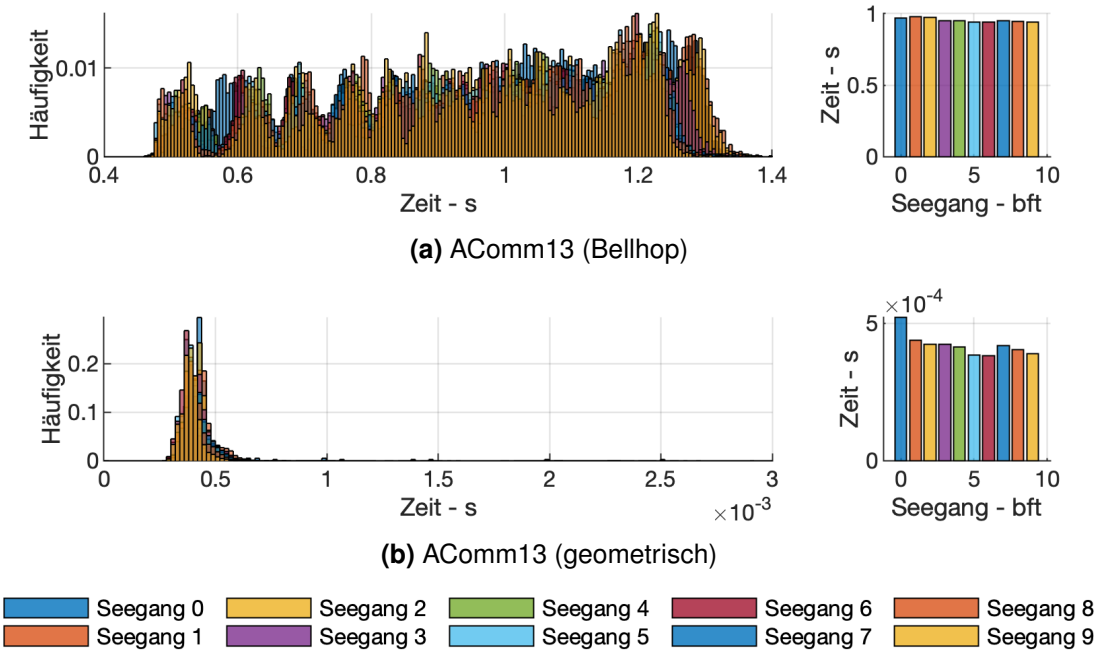


Abbildung 3.20: Zeitliche Auftrittsverteilung der Kanalgenerierung (Trond)

Der grundsätzliche Aufwand verteilt sich in beiden Ansätzen ähnlich und setzt sich aus einer Gauß-ähnlichen Verteilung und einem zusätzlichen Schwanz zusammen. Der zusätzliche Schwanz der Verteilung wird durch einen vermehrten Aufwand von Reflexionen über den Boden und die Oberfläche für mittig ausgerichtete Kombinationen von Sender- und Empfängerpositionen verursacht. Der zweite Aspekt ist, dass bei einem Seegang der Stufe 0 mit dem geometrischen Ansatz, durch die kaum vorhandenen Dämpfungseigenschaften, ein erhöhtes Mehrwegverhalten berechnet werden muss. Das wirkt sich auf die benötigte Rechenzeit aus, welche sich mit höherer Seegangstufe immer weiter reduziert.

Für das Experiment *AComm10* benötigt der geometrische Ansatz mit Seegang der Stufe 0 im Durchschnitt $1.4\,\mathrm{ms}$, was in Bezug zur Stufe 1 mit $0.69\,\mathrm{ms}$, fast der doppelten Rechenzeit entspricht. Insgesamt benötigt der Ansatz mit BELLHOP im Vergleich zum geometrischen Ansatz für die Umgebung *AComm10* 850 mal, für *Trond* 1827 mal und für *AComm13* 2268 mal mehr Rechenzeit. Neben der Erweiterbarkeit und der Beschränkung auf eine Software bietet der geometrische Ansatz damit einen zeitlich signifikanten Vorteil und ermöglicht eine ressourcensparende Online-Generierung für dynamische Testsysteme durch den Verzicht auf datenintensive vorgehaltene Kanalmatrizen. Der geringe



(a) AComm13 (Bellhop)

(b) AComm13 (geometrisch)

Abbildung 3.21: Zeitliche Auftrittsverteilung der Kanalgenerierung (AComm13)

Aufwand würde zudem eine Nutzung während autarker Missionen an Bord ermöglichen und die Auswahl und das dynamische Reprogrammieren der Kommunikationssysteme sowie die Entscheidung, ob kommuniziert werden soll, effizienter gestalten.

Eine Aussage über die Ähnlichkeit der generierten CIR kann über die Bestimmung der periodischen Kreuzkorrelation basierend auf Definition 3.3 mit einer Normierung auf das Maximum der CIR beider Ansätze getroffen werden. Je näher das Ergebnis der Kreuzkorrelation an $100\,\%$ liegt, umso größer ist die zeitliche und numerische Übereinstimmung der jeweiligen CIR der beiden Ansätze. Eine kleinere Korrelation wird damit nur über eine falsche Skalierung der Amplituden und abweichende Verzögerungen verursacht. Eine erste Auswertung zeigte einen Mittelwert der Korrelation im oberen Drittel, der CIR nur schwache Ähnlichkeit aufweist.

Eine genauere Betrachtung zeigte, dass die Abweichungen durch Rundungsfehler und minimale Differenzen bei der Bestimmung der Signalankunftszeiten verursacht wurden. Durch eine zeitliche Korrektur des direkten Signalpfades (LOS) in dem Bereich von $[\frac{-40}{f_s}\ \frac{40}{f_s}]$, bei einer Abtastrate von $160\,\mathrm{kHz}$ zwischen $-2.5\,\mu\mathrm{s}$ und $2.5\,\mu\mathrm{s}$, konnte die mittlere Kreuzkorrelation durch die zeitliche An-

passung korrigiert werden. Ein weiterer Grund für die teils abweichenden Werte ergab sich aus dem fehlenden LOS-Signal über den Ansatz mit BELLHOP. Das Fehlen des LOS-Signal wird durch positive und negative Brechungsgradienten bewirkt, was die als Strahlen interpretierten Schallwellen von der Empfängerposition wegbricht und derzeit im geometrischen Ansatz vernachlässigt wird. Die zweites konträre Ursache liegt in der Verstärkung der Signalpfade, dabei werden vermehrt Strahlen, durch eine günstige Verteilung von Brechungsgradienten am Empfänger kumuliert und damit hat das mittels BELLHOP bestimmte Signal im Gegensatz zum geometrischen Ansatz eine ausgeprägtere Amplitude.

Die Bewertung wurde daher um eine komplementäre Metrik in Form der mittleren quadratischen Abweichung basierend auf Definition 3.2 mit einer Normierung über das Quadrat der von BELLHOP erzeugten CIR angegeben in Prozent für die Vergleichbarkeit als relative Fehlermetrik ergänzt. Während die Korrelation nur eine Aussage bezüglich der Ähnlichkeit trifft, kann die mittlere quadratische Abweichung (MSE) Informationen darüber liefern, wie weit die erzeugten CIR tatsächlichen numerisch, beeinflusst von zeitlichen Differenzen der CIR, abweichen.

Eine niedrige Korrelation bei gleichzeitig niedrigem MSE deutet darauf hin, dass die Signale ähnliche Signalamplituden aufweisen, aber ihre zeitliche Positionierung unterschiedlich ist, was durch die unterschiedlichen Ansätze möglich ist. Eine hohe Korrelation hingegen zeigt, dass die CIR eine zeitliche und numerisch starke Ähnlichkeit aufweisen. Eine niedrige Korrelation bei gleichzeitig hohem MSE zeigt, dass die generierten CIR keine Ähnlichkeit aufweist.

Beide Metriken werden folgend im Kontext der dritten Zielgröße, der Wertigkeit in Form des Difference in Signal to Interference Ratio (DSIR), interpretiert. In drahtlosen Kommunikationssystemen können Interferenzen von verschiedenen Quellen kommen, zum Beispiel durch ander Sendern im selben Frequenzband oder von Störquellen in Form von Reflexionen. Beim SIR wird dazu nur das LOS im Verhältnis zur Interferenz durch die umgebungsspezifischen Reflexion betrachtet. Der eingeführte DSIR in Definition 3.1 gibt damit den Unterschied, des SIR der generierten CIR über die Funktion *calculateDSIR* in Listing A.10 an. Je niedriger der bestimmte DSIR ist, desto ähnlicher ist die zu erwartende resultierende Performanz und damit die Wertigkeit des geometrischen Ansatzes im Vergleich zur Umsetzung mit BELLHOP. Die Abbildungen 3.22 bis 3.23 zeigen dazu Beispielhaft links die Korrelation für eine gegebene Sendetiefe und Seegang über die definierten Testpunkte für die Trägerfrequenz f_c von $10\,\mathrm{kHz}$, mittig die normierte MSE und rechts die dazugehörige Differenz des SIR (DSIR).

Definition 3.1 — DSIR:
Die DSIR ist eine Kennzahl zur Quantifizierung des Unterschieds zweier Signale auf Basis des SIR der geschätzten und erwarteten CIR. Für die angenommene Länge t_{sig} des Signals x wird die SIR mittels CIR wie folgt über das zeitlich relative quadratische Verhältnis der LOS und aller anderen resultierenden Pfade und deren Amplitude a_i definiert:

$$SIR_{\mathrm{dB}} = 10 \cdot log_{10}(\mathrm{SIR}) \text{ mit SIR} = \frac{(a_1 \cdot t_{sig}))^2}{\left(\sum_{n=2}^{N} a_n \cdot (t_{sig} - (\tau_n - \tau_1))\right)^2} \tag{3.53}$$

(3.54)

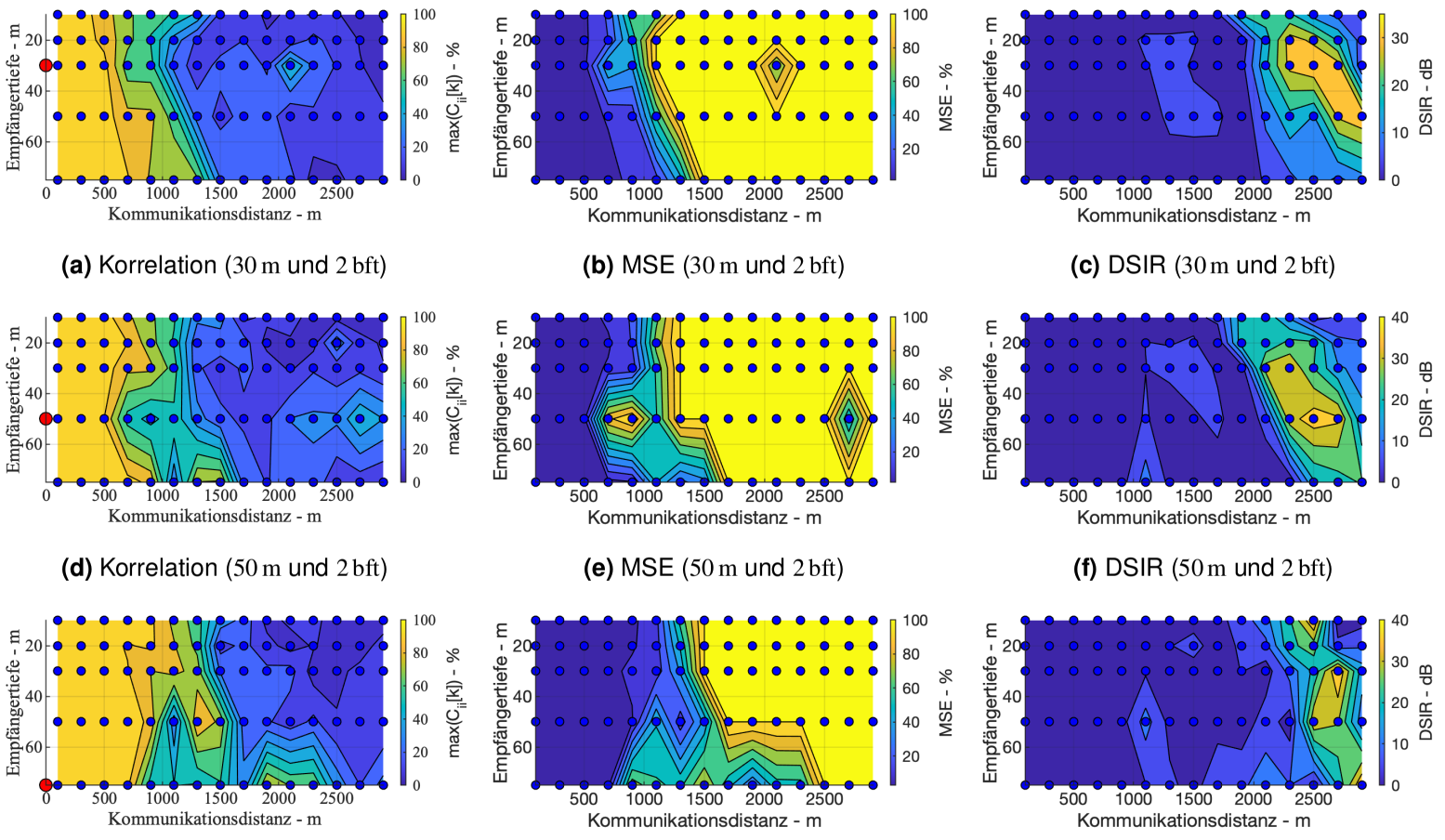
Darauf basierend wird der DSIR über die Differenz zwischen dem geschätzten SIR′ und dem erwarteten SIR bestimmt:

$$DSIR = SIR_{dB} - SIR'_{dB} = 10 \cdot \log_{10} \left(\frac{\left(\frac{a_1 \cdot t_{\mathrm{sig}}}{\sum_{n=2}^{N} a_n \cdot (t_{\mathrm{sig}} - (\tau_n - \tau_1))} \right)^2}{\left(\frac{a'_1 \cdot t_{\mathrm{sig}}}{\sum_{n=2}^{N'} a'_n \cdot (t_{\mathrm{sig}} - (\tau'_n - \tau'_1))} \right)^2} \right) \tag{3.55}$$

in Dezibel angegeben. Eine obere Abschätzung kann über $t_{\mathrm{sig}} \to \infty$ bestimmt werden:

$$DSIR \overset{t_{\mathrm{sig}} \to \infty}{\leq} 10 \cdot \log_{10} \left(\frac{\left(\frac{a_1}{\sum_{n=2}^{N} a_n} \right)^2}{\left(\frac{a'_1}{\sum_{n=2}^{N'} a'_n} \right)^2} \right) \tag{3.56}$$

Die Korrelationen und der MSE in Abbildung 3.22 zeigen, dass diese in flachen Gewässer mit steigender Entfernung für den geometrischen Ansatz erheblich abnimmt, aber bis zu einer Distanz von ungefähr $700\,\mathrm{m}$ eine zeitliche und numerische Übereinstimmung von über $96\,\%$ erreicht wird. Der restliche Bereich weist zwar eine geringere Korrelation auf, verursacht durch eine zeitliche Diskrepanz, aber bis $1700\,\mathrm{m}$ mit einem relativ geringen DSIR. Es verdeutlicht das die geringe Korrelation durch einen minimalen Unterschied in der ZEIT verursacht wird, der kaum Einfluss auf die Wertigkeit des geometrischen Ansatzes hat. Allerdings liegt die durchschnittlich Korrelation über alle Sendetiefen und Seegänge in Abbildungen 3.25a bis 3.25c bei $48.3\,\%$ mit einem DSIR von $12.3\,\mathrm{dB}$ über alle Kommunikationsdistanzen und einer Korrelation von $98\,\%$ mit einem DSIR von $1.23\,\mathrm{dB}$ beschränkt auf eine maximale Kommunikationsdistanz bis einschließlich $900\,\mathrm{m}$.

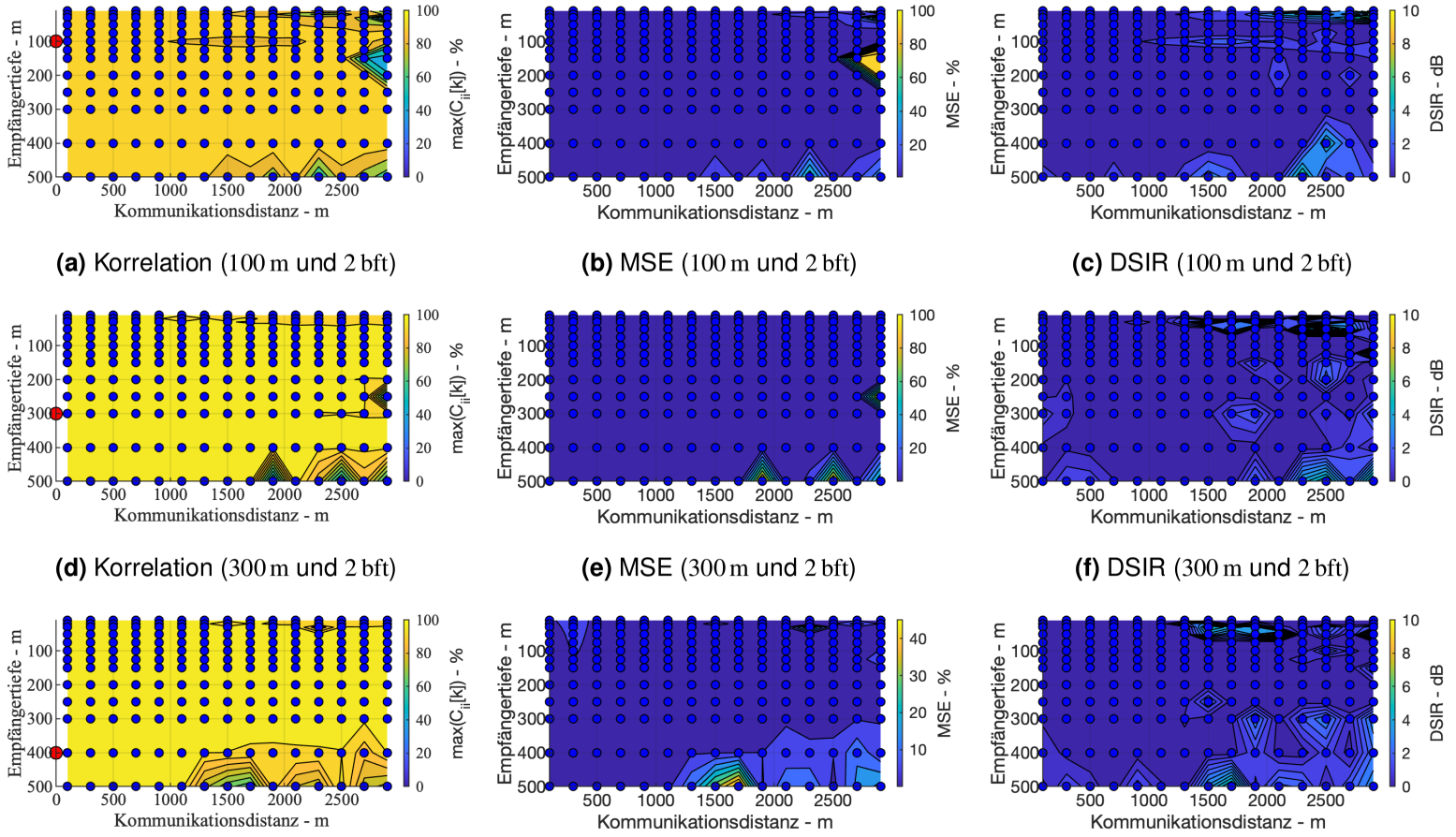


(a) Korrelation (30 m und 2 bft) **(b)** MSE (30 m und 2 bft) **(c)** DSIR (30 m und 2 bft)

(d) Korrelation (50 m und 2 bft) **(e)** MSE (50 m und 2 bft) **(f)** DSIR (50 m und 2 bft)

(g) Korrelation (75 m und 2 bft) **(h)** MSE (75 m und 2 bft) **(i)** DSIR (75 m und 2 bft)

Abbildung 3.22: Abbildung der Korrelation, MSE und dem DSIR zwischen dem geometrischen Ansatz und BELLHOP für die Umgebung *AComm10*

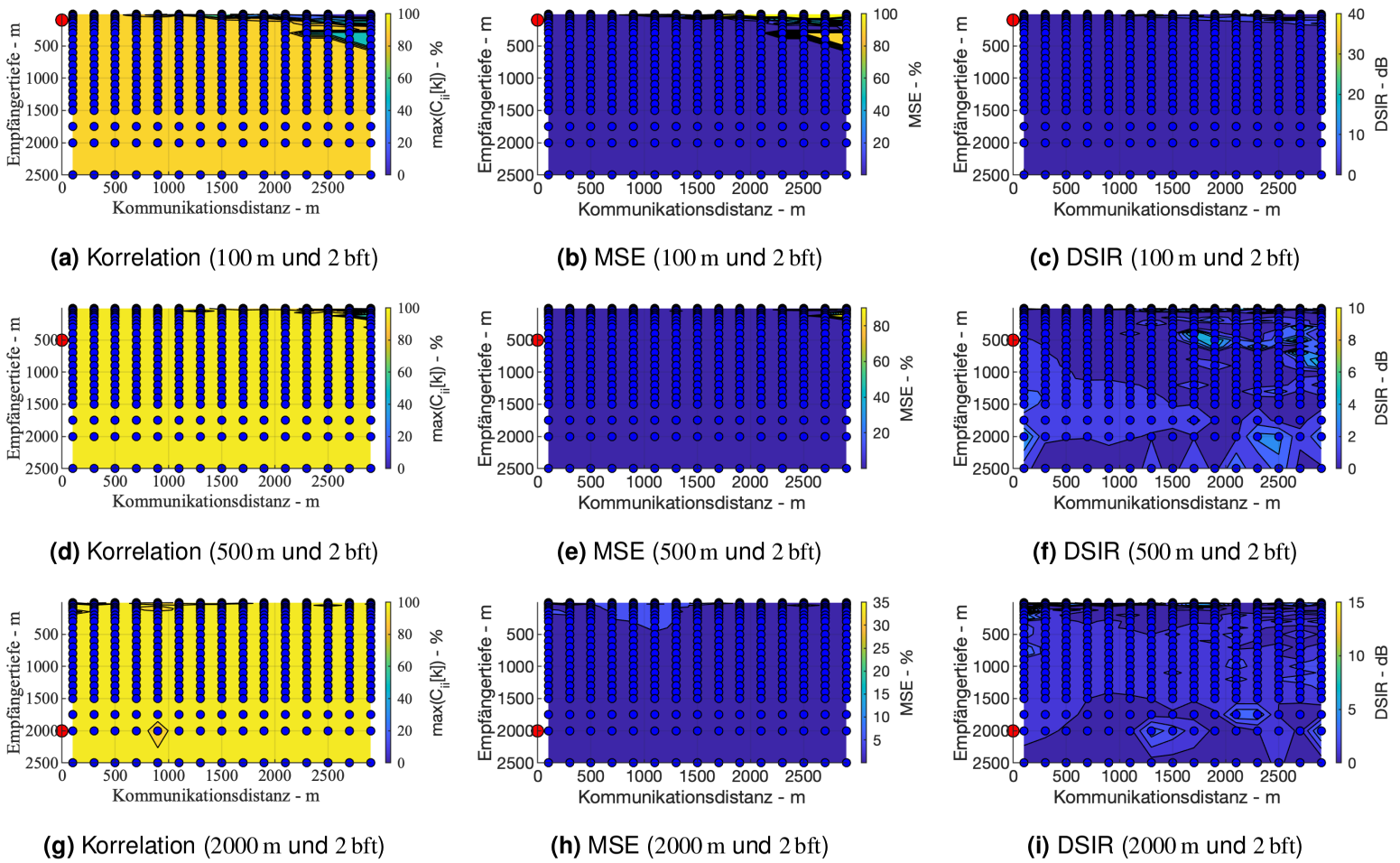


(a) Korrelation (100 m und 2 bft) **(b)** MSE (100 m und 2 bft) **(c)** DSIR (100 m und 2 bft)

(d) Korrelation (300 m und 2 bft) **(e)** MSE (300 m und 2 bft) **(f)** DSIR (300 m und 2 bft)

(g) Korrelation (400 m und 2 bft) **(h)** MSE (400 m und 2 bft) **(i)** DSIR (400 m und 2 bft)

Abbildung 3.23: Abbildung der Korrelation, MSE und dem DSIR zwischen dem geometrischen Ansatz und BELLHOP für die Umgebung *Trond*



(a) Korrelation (100 m und 2 bft)

(b) MSE (100 m und 2 bft)

(c) DSIR (100 m und 2 bft)

(d) Korrelation (500 m und 2 bft)

(e) MSE (500 m und 2 bft)

(f) DSIR (500 m und 2 bft)

(g) Korrelation (2000 m und 2 bft)

(h) MSE (2000 m und 2 bft)

(i) DSIR (2000 m und 2 bft)

Abbildung 3.24: Abbildung der Korrelation, MSE und dem DSIR zwischen dem geometrischen Ansatz und BELLHOP für die Umgebung *AComm13*

Definition 3.2 — MSE:
Der MSE ist ein Maß für die durchschnittliche quadratische Abweichung zwischen den geschätzten und erwarteten Werten. Für eine Stichprobe von n Beobachtungen, mit den erwarteten Werten y_i und den geschätzten Werten $\hat{y}_i$, wird der MSE wie folgt definiert:

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{3.57}$$

Definition 3.3 — Kreuzkorrelation:
Die Kreuzkorrelation kann in die periodische (auch bekannt als "Circular Cross-Correlation") und aperiodische Kreuzkorrelation unterteilt werden. Bei der aperiodischen Kreuzkorrelation sind die Werte vor und nach einer Folge nicht bekannt und werden mittels Nullsetzen (Zero-Padding) ausgeblendet. Sei M die Matrix aller Folgen und $M(i,:)$ die i'te und $M(j,:)$ die j'te diskrete Folge der Länge N mit den definierten Symbolwerten $\{-1,+1\}$, dann ist die normierte aperiodische Kreuzkorrelation in abhängig der Verschiebung k in Symbolen gegeben durch:

$$C_{i,j}[k]=\begin{cases}\frac{1}{N-|k|}\sum_{n=0}^{N-k-1}M(i,n)M(j,n+k) & k\geq 0\\ \frac{1}{N-|k|}\sum_{n=-k}^{N-1}M(i,n)M(j,(n+k) \mod N) & k<0\end{cases} \tag{3.58}$$

Unter der Annahme das die diskreten Folgen der Matrix M mit der Länge N periodisch auftreten, kann die periodischen Kreuzkorrelation in Gl. (3.59), als Ergebnis von Sarwate und Pursley [153], über Gl. (3.60) bestimmt werden.

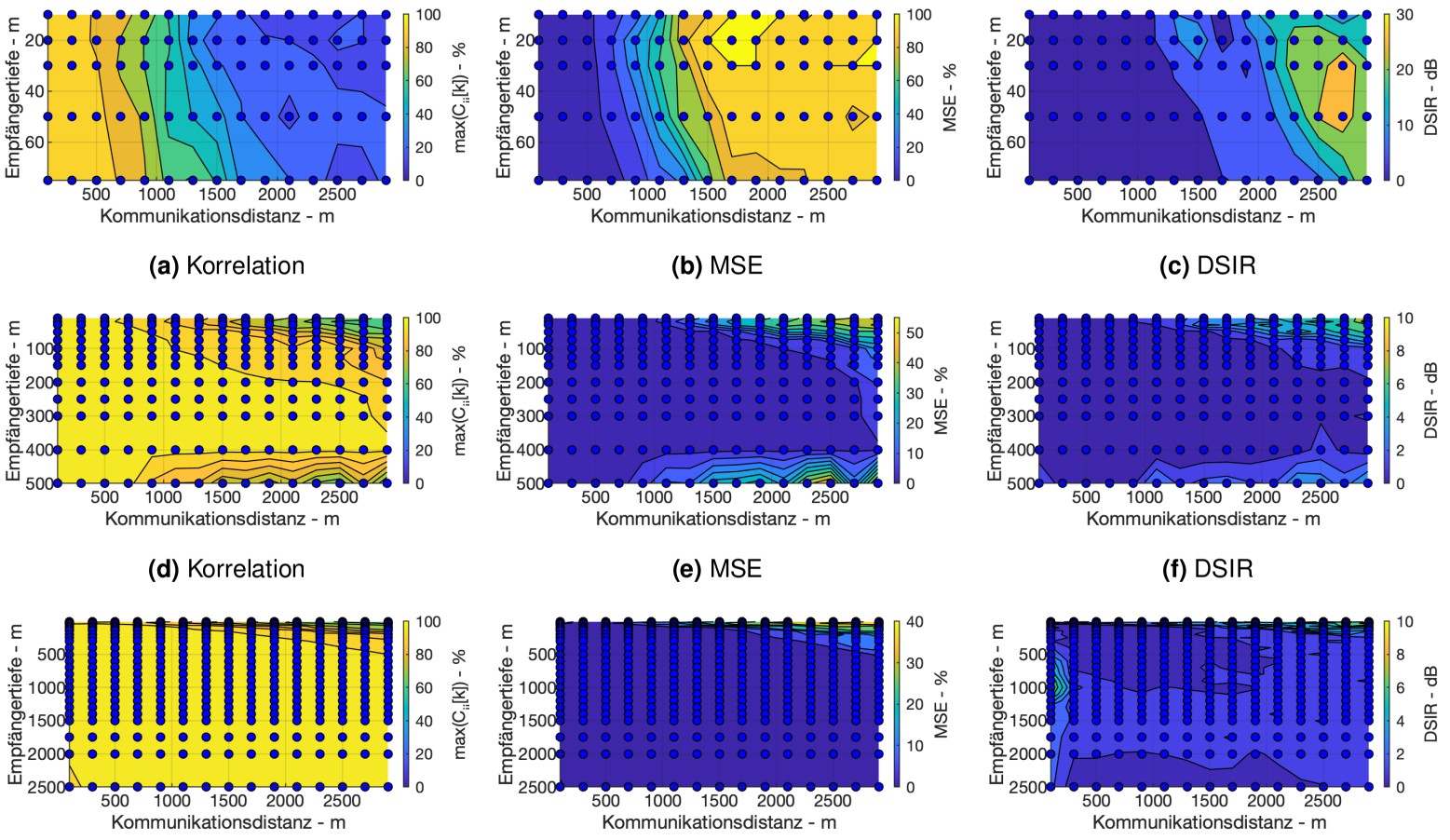
$$C^p_{i,j}[k]=C^p_{j,i}[-k]^*C^p_{i,j}[k+N] \tag{3.59}$$

$$C^p_{i,j}[k]=\begin{cases}\frac{1}{N}\sum_{n=0}^{N-1}M(i,n)M(j,(n+k) \mod N) & k\geq 0\\ \frac{1}{N}\sum_{n=0}^{N-1}M(j,n)M(i,(n+|k|) \mod N) & k<0\end{cases} \tag{3.60}$$

Für $k=0$ ist das Ergebnis beider Kreuzkorrelationen identisch:

$$\begin{aligned}C^p_{i,j}[0]&=\frac{1}{N}\sum_{n=0}^{N-1}M(i,n)M(j,(n+0) \mod N)\\ C^p_{i,j}[0]&=\frac{1}{N-|0|}\sum_{n=0}^{N-0-1}M(i,n)M(j,n+0)=C_{i,j}[0]\end{aligned} \tag{3.61}$$

Für die Umgebung *Trond* zeigen die Beispiele in Abbildung 3.23 und der Durchschnitt über alle Sendetiefen und Seegänge in Abbildungen 3.25d bis 3.25f eine zeitliche und numerische Übereinstimmung von über $90\,\%$ mit einem DSIR von$1.48\,\text{dB}$. Die auftretenden Abweichung mit einem Vorzeichenwechsel des



(a) Korrelation **(b)** MSE **(c)** DSIR

(d) Korrelation **(e)** MSE **(f)** DSIR

(g) Korrelation **(h)** MSE **(i)** DSIR

Abbildung 3.25: Durchschnitt der Korrelation, MSE und dem DSIR zwischen dem geometrischen Ansatz und BELLHOP über alle Sendetiefen und Seegänge der Experiment-Umgebungen

Geschwindigkeitsgradienten werden insbesondere im Bereich der Oberfläche bei einer ungefähren Wassertiefe von $50\,\mathrm{m}$ verursacht. Damit werden die Strahlen von BELLHOP die Oberhalb des Vorzeichenwechsels liegen nach oben und Unterhalb nach unten weggebrochen und erreichen den Empfänger bei höherer Entfernung, entgegen dem geometrischen Ansatz nicht mehr. Die Auswirkung auf den DSIR hält sich aber aufgrund der trotzdem ähnliche bestimmten Mehrwegreflexion relativ klein. Einzig die LOS wird geschwächter und verzögert Empfangen und reduziert damit einzig die Korrelation signifikant.

Für tiefe Gewässer, wie Beispielhaft für die Umgebung *AComm13* in Abbildung 3.24 gezeigt, wird eine durchschnittliche Korrelation von $93.5\,\%$ bei einer Abweichung des DSIR von ca. $1.78\,\mathrm{dB}$ erreicht. Entgegen der Umgebung *Trond* wird im Bereich der Oberfläche, ähnlich dem eingeführten SOFAR-Kanal, wird durch die entgegengesetzten Geschwindigkeitsgradienten das Signal über BELLHOP hin und her gebrochen und verbleibt bei $50\,\mathrm{m}$. Damit wird das LOS am Empfänger, entgegen dem geometrischen Verfahren, mehrfach Empfangen und reduziert die resultierende Korrelation bei einer signifikanten Differenz des DSIR.

Für flache Gewässer bildet damit, für Distanzen die maximal dem 10-fachen der Wassertiefe entsprechen, der geometrische Ansatz eine valide Alternative, wie die durchschnittlichen Werte über alle Seegänge und Sendetiefen in der Abbildung 3.25 aufzeigen. Für mittlere und tiefe Gewässer hingegen bietet der geometrische Ansatz eine effiziente Alternative mit einer signifikant kürzeren Generationszeit zur erstellen und Auswerten von Umgebungsspezifischen CIR.

3.3.3 Modellierung des Umgebungsrauschens

Die Modellierung des Umgebungsrauschens ist von entscheidender Bedeutung in der Unterwasserakustik und sollte den akustischen Kanal um ein realistisches Rauschen ergänzen. In diesem Zusammenhang werden folgend allgemein genutzte Ansätze diskutiert, um das Rauschen zu modellieren und zu simulieren.

Ein etabliertes Verfahren ist die Knudsen-Kurve, die jedoch den Einfluss anderer dominanter Geräuschquellen nicht vollständig berücksichtigt. Die Knudsen-Kurve [154] adressiert hauptsächlich den Einfluss des Windes auf das Umgebungsrauschen. Der Ansatz nimmt an, dass der Wind die dominante Quelle des Umgebungsrauschens ist, und gibt das Rauschspektrum als Funktion Gl. (3.62) des Seegangs s_{st} wieder. Dazu wird die Knudsen-Kurve bis zu $1\,\mathrm{kHz}$ als konstant angenommen und nimmt pro Dekade $17\,\mathrm{dB}$ ab. Abhängig vom Seegang s_{st} wird der Pegel des Rauschspektrums über den gesamten Frequenzbereich

angehoben oder gesenkt.

$$N_{Knud} = min\{50, 101 - 17log10(f)\} + 4 \cdot s_{st} \tag{3.62}$$

Der zweite gängige Ansatz ist die Nutzung des additives weißes gaußsches Rauschen (AWGN), wodurch der Einfluss im Unterwasserkanal komplett ignoriert wird. Durch die Annahme eines konstanten Rauschpegels mit Gaußverteilung kann es zu einer Verfälschung der Ergebnisse kommen [155]. Bei der Modellierung des Umgebungsrauschens mit dem AWGN, definiert über das Verhältnis zwischen Rausch- und Signalleistung (SNR) oder alternativ über die Signalamplitude A_s und die Standardabweichung σ_n (Gl. (3.63)), wird das Rauschen mit konstanter PSD und gaußverteilter Amplitude mit dem empfangenen Kommunikationssignal überlagert. Bei der Frequenz- und Phasenmodulation, bei der sich die Signalenergie in einem festgelegten Frequenzspektrum konzentriert, wird auch der Begriff Träger-Rausch-Abstand (CNR) genutzt.

$$SNR = 10log\left(\frac{P_s}{P_n}\right) = 10log\left(\frac{A_s}{\sigma_n}\right) \tag{3.63}$$

Um eine präzisere Modellierung des Umgebungsrauschens zu ermöglichen, wird ein für diese Arbeit entwickelter Ansatz zur Rauscherzeugung unter Verwendung von FIR-Filtern eingeführt [156]. FIR-Filter sind lineare, diskrete Systeme, die es ermöglichen, das Rauschen frequenzspezifisch zu manipulieren. Durch die Anpassung der Filterkoeffizienten kann das Rauschspektrum gezielt beeinflusst werden.

3.3.3.1 Modellierung mit FIR-Filtern

Neben der Übereinstimmung mit dem definierten und experimentell vielfach nachgewiesenen Unterwasserrauschen, soll in dieser Arbeit ein Ansatz für ein Szenario-basiertes (zeit-variantes) Rauschen aufgezeigt werden. Dieser soll sowohl für die Simulation als auch auf Hardware wie den Field Programmable Gate Arrays (FPGAs) reproduzierbar und vorhaltbar sein. Die grundlegende Idee ist die Erzeugung des Umgebungsrauschens durch frequenzspezifische Manipulation des Leistungspegels eines gleichmäßig verteilten Pseudo-Noise-Signals mithilfe digitaler FIR-Filter. Ein FIR-Filter ist ein lineares, diskretes System, bei dem die diskrete Ausgabe $y[n]$ auf die Faltung des diskreten Eingangssignals $x[n]$ und der diskreten Impulsantwort h bezogen ist (Gl. (3.64)). Der resultierende Frequenzgang des FIR-Filters (Gl. (3.65)) wird über Kreisfrequenz ω und die Abtastperiode f_s definiert.

Für digitale Systeme lässt sich der Frequenzgang über die normierte Kreisfrequenz $\hat{\omega}=\omega f_s$ (Gl. (3.66)) ausdrücken. Die Gl. (3.64) kann wegen der Gleichheit der Koeffizienten b_k mit diskreter Impulsantwort $h[k]$, mit den FIR-Filter-Koeffizienten b_k, der Ausgabe $y[k]$ und der Eingabe $x[k]$ für ein diskretes System (Gl. (3.67)) direkt konvertiert werden.

$$y[n]=\sum_{k=0}^{K} x[n-k]h[k] \tag{3.64}$$

$$H(\omega)=\sum_{n=0}^{N-1} h(n)e^{-j\omega n f_s} \tag{3.65}$$

$$H(\hat{\omega})=\sum_{n=0}^{N-1} b_k e^{-j\hat{\omega}n} \tag{3.66}$$

$$y[n]=\sum_{k=0}^{K} b_k x[n-k]=\sum_{k=0}^{K} h[k]x[n-k] \tag{3.67}$$

Die Gleichheit beruht auf der fundamentalen Eigenschaft der linearen Zeitinvarianz von FIR-Filter, die besagt, dass die Ausgangsantwort auf einen Impuls die Impulsantwort selbst ist (Gl. (3.68)).

$$y[n]=\sum_{k=0}^{M} b_k \cdot \delta[n-k]=h[n] \tag{3.68}$$

$$x[n]=Ae^{j(\hat{\omega}+\theta)}=Ae^{j\theta}e^{j\hat{\omega}n} \quad \text{für } -\infty<n<\infty \tag{3.69}$$

Unter der Annahme, dass das eingehende Signal $x[k]$ eine komplexe Exponentialfolge (Gl. (3.69)) ist, ergibt sich Gl. (3.70). Bei Gl. (3.70) und Gl. (3.71) entspricht die ausgehende der eingehenden Sequenz multipliziert mit dem Frequenzgang aus Gl. (3.67) und ändert nur die Amplitude. Damit kann die Impulsantwort $h[k]$, welche das gesuchte Spektrum $S_y(\hat{\omega})$, definiert in Gl. (3.72), erzeugt, berechnet werden [157].

$$y[n]=\sum_{k=0}^{K} b_k A e^{j\theta}e^{j\hat{\omega}(n-k)}=\sum_{k=0}^{K} b_k e^{-j\omega k}Ae^{j\theta}e^{j\hat{\omega}n} \tag{3.70}$$

$$y[n]=\left(H(\hat{\omega})e^{-j\omega k}Ae^{j\theta}\right)\cdot e^{j\hat{\omega}n} \tag{3.71}$$

$$h[n]=\frac{1}{N}\sum_{k=0}^{N/2-1} 2A[k]cos(2\pi(n-M)(k-1/2)/N) \tag{3.72}$$

Mit der erzeugten Impulsantwort kann mit einem gaußverteilten Eingangssignal und dessen PSD $S_{xx}(f)(\hat{\omega})$ die gewünschte PSD des Umgebungsgeräusches für N Samples mithilfe eines FIR-Filters über die Output-Input-PSD-Relation in Gl. (3.73) erzeugt und mit dem Korrekturfaktor $S_c(\hat{\omega})$ angepasst werden.

$$S_{yy}(f)(\hat{\omega}) = |H(\hat{\omega})|^2 S_{xx}(f)(\hat{\omega}) - S_c(\hat{\omega}) \tag{3.73}$$

Der spektrale Korrekturfaktor $S_c(\hat{\omega})$ in Gl. (3.73) kann mit einer ausreichenden Anzahl von erzeugten Signalen über die Impulsantwort und dem FIR-Filter, mithilfe der Welch-Methode (Gl. (3.74)) [158] und den berechneten Werten aus Gl. (3.30) bis Gl. (3.33) für den definierten Frequenzbereich bestimmt werden.

$$S_{xx}(f)(\hat{\omega}) = \frac{1}{N} \left| \sum_{k=0}^{N-1} x[k] \cdot w[k] \cdot e^{-j\hat{\omega}k} \right|^2 \tag{3.74}$$

Die Gl. (3.74) bestimmt dabei das PWELCH-Spektrum (PSD) $S_{xx}(f)$ von $x[k]$ mit $N = N_{\text{taps}}$ Samples in $x[k]$, der Fensterfunktion $w[k]$ zur Abschwächung der Eckpunkte und somit einer Glättung des Spektrums mit der normierten Kreisfrequenz $\hat{\omega}$ und der Abtastrate f_s. Das Umgebungsrauschen mit FIR-Filter kann auf drei Arten implementiert werden. Die erste Möglichkeit ist die Nutzung von nur einem Filter für das Gesamtrauschen. Der zweite Ansatz zerlegt das Rauschen in seine vier Hauptkomponenten Gl. (3.30) bis (3.33), die jeweils separat mit einem Filter erzeugt und abschließend überlagert werden. Die dritte Variante basiert auf der Überlagerung von getrennt erzeugten Koeffizienten der Rauschkomponenten mit einer abschließenden Überlagerung der Koeffizienten und Generierung des Rauschens mit einem Filter. Für die Bestimmung der Koeffizienten $b[k]$ der FIR-Filter gibt es unterschiedliche Ansätze, von denen speziell drei in Betracht gezogen werden: die Nutzung des Haming-Windows [159], des Kaiser-Windows [160] und der Anwendung von Least-Square [161, 162].

Das Hamming-Window w_H in Gl. (3.75) der Länge N_{taps} wird mit n als Index des aktuellen Samples und N_{taps} der Anzahl der Samples bestimmt und kann Werte zwischen 0 und 1 annehmen. Die FIR-Filter-Koeffizienten, basierend auf dem Hamming-Window (im Folgenden abgekürzt zu H-FIR), haben einen sanfteren Übergang an den Grenzbereichen auf Kosten der Flankensteilheit, also der Dämpfung im Übergang zum Sperrbereich, und begrenzen die Auswirkungen des spektralen Leck-Effekts (leakage).

Das Kaiser-Window w_K in Gl. (3.76) mit dem Index n, der Anzahl der Samples N_{taps}, der modifizierten Bessel-Funktion erster Art und nullter Ordnung I_0 und dem Parameter β, welcher die Form des Kaiser-Windows bestimmt. Je höher der Wert von β, desto schmaler wird das Kaiser-Window. Durch die wählbare Breite des Kaiser-Window kann ein FIR-Filter, basierend auf diesem Fenster, im Folgenden K-FIR genannt, wahlweise eine höhere Auflösung der Amplitude oder

der Frequenz bei einer stärkeren Dämpfung der Nebenkeulen erreichen.

$$w_H(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \tag{3.75}$$

$$w_K(n) = \frac{I_0\left[\beta \cdot \sqrt{1-\left(\frac{n-(N-1)/2}{(N-1)/2}\right)^2}\right]}{I_0(\beta)} \tag{3.76}$$

Im Gegensatz zu den Windows-Methoden wird bei Least-Squares basierenden FIR-Filtern, im folgenden LS-FIR genannt, ein lineares Gleichungssystem gelöst, um die Koeffizienten w_{LS} des FIR-Filters zu bestimmen. Die Impulsantwort des Filters wird als Linearkombination von Basisfunktionen mit den Koeffizienten w_{LS} so gewählt, dass die Summe der quadrierten Abweichungen zwischen der gewünschten und der erzeugten Frequenzantwort minimiert wird.

3.3.3.2 Auswertung des FIR-Filter-Ansatzes

Um die Ähnlichkeit der PSDs zu bestimmen, stehen Methoden wie die Bestimmung des MSE, des Kullback-Leibler-Divergenz (KLD) [163], der Cross-Spectral Density (CSD) [164] und der Kreuzkorrelation Leistungsdichten zur Verfügung. Während die MSE (Definition 3.2) einfach zu bestimmen ist und auch empfindlich genug für niedrige Frequenzen erscheint, ist diese Methode sehr empfindlich bei Störungen und Rauschen. Die KLD, welche auf der MSE basiert, berücksichtigt zusätzlich die Verteilung der PSD, kann aber bei Null-Frequenzen zu unendlichen Werten führen, was die Aussage über die Ähnlichkeit von verglichenen Signalen entwertet. Zudem ist die Methode unempfindlich gegenüber ähnlichen, aber variierenden PSDs und dadurch für die Bestimmung der Ähnlichkeit nur bedingt nutzbar. Die Kreuzkorrelation (Definition 3.3) hingegen kann verwendet werden, um Ähnlichkeiten in der PSD zu bestimmen. Die Kohärenzanalyse unter Verwendung der Kreuzleistungsdichte ermöglicht die Bestimmung der Ähnlichkeit von Folgen oder Muster zueinander. Allerdings kann bei der Kreuzleistungsdichte die Ähnlichkeit zwischen den PSDs überschätzt werden, wenn die PSDs eine ähnliche Form aufweisen, aber über unterschiedliche Amplitudenverteilung verfügen.

Daher wird die Kohärenz (Ähnlichkeitsbetrachtung) unter Verwendung der normierten Kreuzleistungsdichte (Kohärenzdichte,Definition 3.4) in Kombination mit dem MSE, bestimmt.

Definition 3.4 — Kohärenzdichte :
Die Kohärenzdichte $\gamma_{xy}(f)$, auch unter dem Begriff normierte Kreuzleistungsdichte bekannt, ist eine dimensionslose Größe, die die Korrelation zwischen zwei Signalen in Bezug auf die Frequenz beschreibt. Die diskrete Kohärenzdichte zwischen den Signalen $x[n]$ und $y[n]$ wird durch die diskrete Kreuzspektraldichte $S_{xy}[f_k]$ und die diskreten Leistungsspektraldichten $S_{xx}[f_k]$ und $S_{yy}[f_k]$ definiert:

$$\gamma_{xy}[f_k] = \frac{S_{xy}[f_k]}{\sqrt{S_{xx}[f_k] \cdot S_{yy}[f_k]}} \tag{3.77}$$

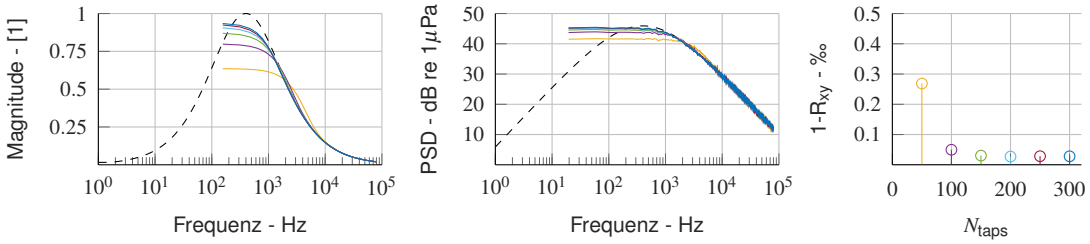
Die Kreuzspektraldichte $S_{xy}(f)$ und die Leistungsspektraldichten $S_{xx}(f)$ und $S_{yy}(f)$ können durch die diskrete Fourier-Transformation der Signale berechnet werden. Die diskreten Kreuzspektraldichte $S_{xy}[f_k]$ der Signale $x[n]$ und $y[n]$ bei der Frequenzkomponente f_k mit Länge N und Abtastperiode T_s wird wie folgt berechnet:

$$S_{xy}[f_k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot y^*[n] \cdot e^{-j2\pi f_k n T_s} \tag{3.78}$$

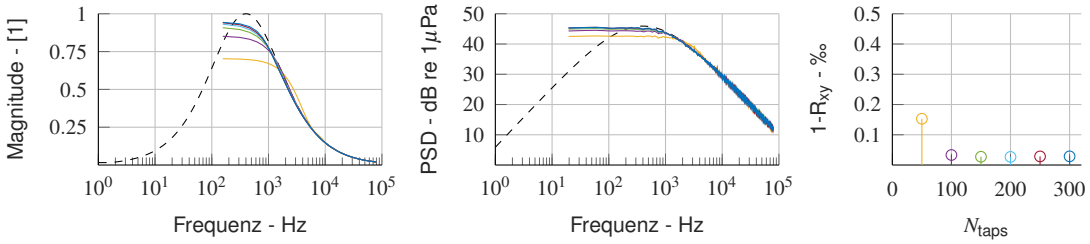
Die Abbildung 3.26 enthält die Resultate für die Bestimmung der FIR-Filter-Koeffizienten zur Generierung des Oberflächenrauschens für die jeweilige Methode H-FIR (3.26a), K-FIR (3.26b) und LS-FIR (3.26c). Die Abbildungen zeigen (links) den Vergleich zwischen der theoretischen und idealen Filterantwort, (Mitte) die tatsächliche Filterantwort als PSD im Vergleich zur berechneten PSD des Oberflächenrauschens und (rechts) die Abweichung voneinander über $1-C_{xy}(t)$ in für die jeweilige Anzahl von $50\ N_{\text{taps}}$ bis $300\ N_{\text{taps}}$ im Abstand von $50\ N_{\text{taps}}$.

Die Abbildung der Resultate zur Bestimmung der FIR-Filter-Koeffizienten für die restlichen drei dominanten Rauschkomponenten befindet sich im Anhang A.1, A.2 und A.3. Die Ergebnisse zeigen, dass die Bestimmung der Koeffizienten und deren Länge variieren für einen modellbasierten Einsatz einer dynamischen Rauschgenerierung in zeitabhängig der simulierten Parameter führt zu drei Ansätzen, der FIR-FilterNutzung.

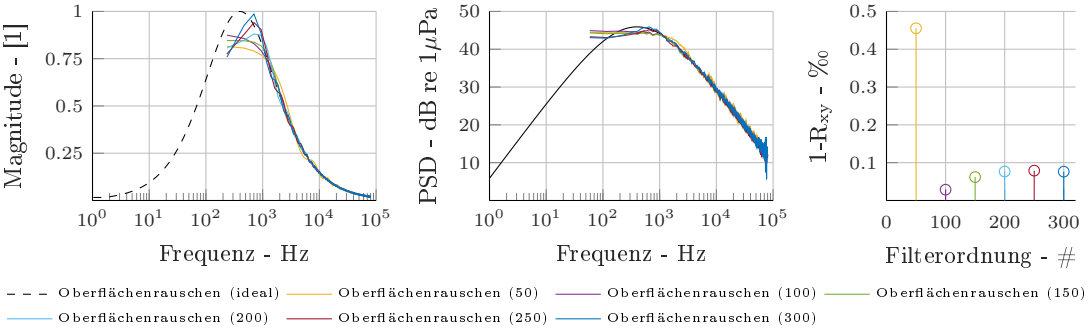
Der erste Ansatz bestimmt über die Parameter das resultierende Gesamtspektrum des Rauschens und bildet über diese Amplitude die Koeffizienten und generiert das Rauschen mit einem FIR-Filter. Die Abbildung 3.28a zeigt dazu die simulierten und das mittels FIR-Filter bestimmten Gesamtrauschen mit einer Filterordnung von 200 und die Verwendung des Hamming-Windows, des bestimmten Korrekturfaktors und dies in Abhängigkeit von Windgeschwindigkeit c_{wind} und Schiffsaktivität s ab. Die Abbildung 3.28c gibt dazu die Abweichung in Promille für das gesamte Frequenzspektrum $f^- = 1\,\text{Hz} \leq f \leq f^+ = 80\,\text{kHz}$ und Abbildung 3.28c für das begrenzte Frequenzspektrum von $f^- = 500\,\text{Hz} \leq f \leq 50\,\text{kHz} = f^+$.



(a) Oberflächenrauschen (H-FIR)



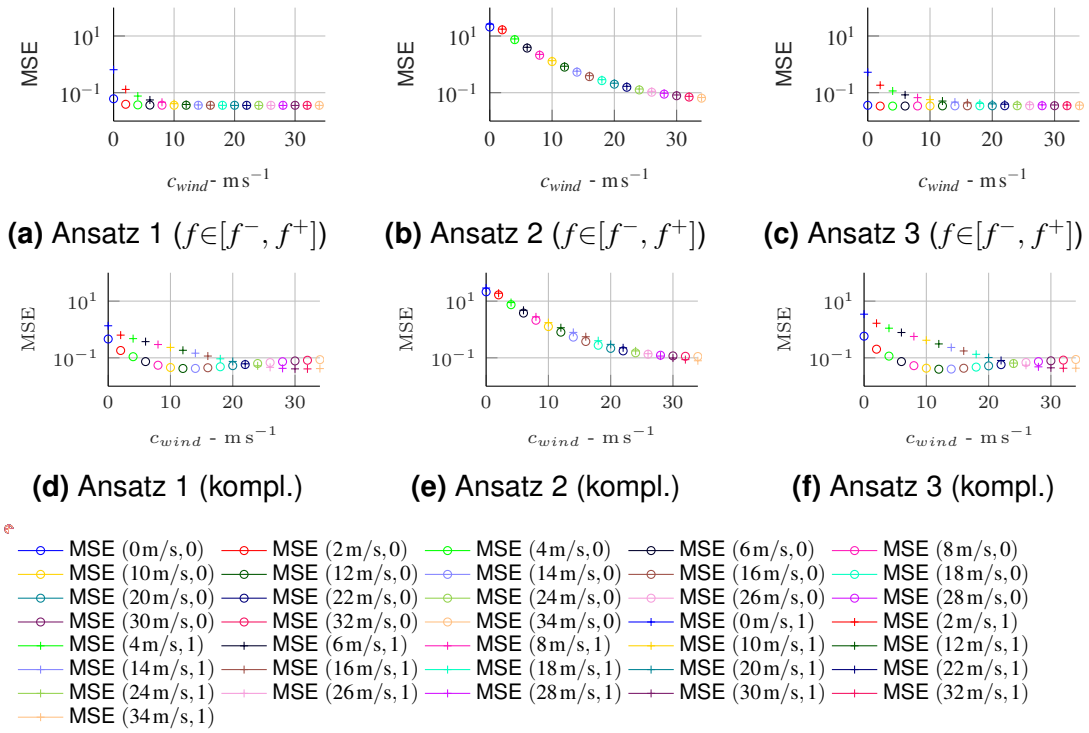
(b) Oberflächenrauschen (K-FIR)



(c) Oberflächenrauschen (LS-FIR)

Abbildung 3.26: Frequenzantwort und Ähnlichkeit (Oberflächenrauschen)

Der zweite Ansatz bestimmt für jede Rauschkomponente individuell bestmöglich die Koeffizienten und generiert jede Rauschquelle separat das Rauschen (vier FIR-Filter) und überlagert diese zu einem Gesamtrauschen. Der dritte Ansatz hingegen erzeugt für jede Komponente die Koeffizienten jeweils separat auf eine festgelegte Länge (150) und überlagert diese und generiert mit einem FIR-Filter das Gesamtrauschen. Die Ergebnisse der Rauschgenerierung für den ersten Ansatz sind in Abbildung 3.28a-Abbildung 3.28c, in Abbildung 3.28d-Abbildung 3.28f für den zweiten und in Abbildung 3.28g bis 3.28i für den dritten Ansatz dargestellt.

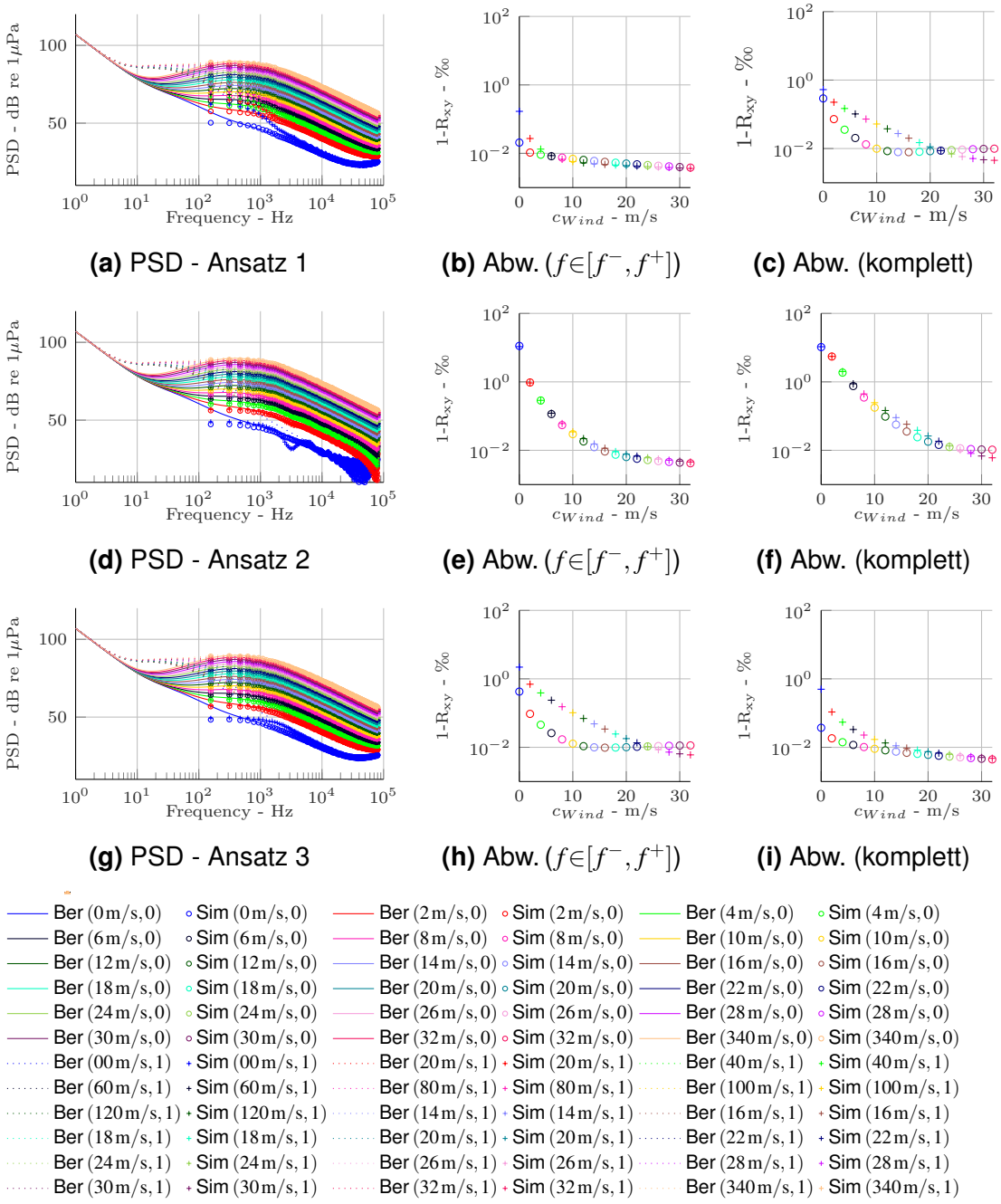


(a) Ansatz 1 ($f \in [f^-, f^+]$) **(b)** Ansatz 2 ($f \in [f^-, f^+]$) **(c)** Ansatz 3 ($f \in [f^-, f^+]$)

(d) Ansatz 1 (kompl.) **(e)** Ansatz 2 (kompl.) **(f)** Ansatz 3 (kompl.)

Abbildung 3.27: MSE der PSDs (Ansatz 1–3) im Vergleich

Die Ergebnisse in Abbildung 3.28 zeigen, dass der erste und dritte Ansatz ähnlich gute Resultate liefert und die Ähnlichkeit in beiden Fällen bei über $99.9\,\%$ liegt. Der zweite Ansatz hingegen verfügt im Schnitt zwar auch über eine Ähnlichkeit von über $99\,\%$, allerdings sind hier die Abweichungen und die variierende PSD erkennbar. Die Abbildungen 3.27a-3.27f geben dazu die jeweilige MSE (in $dB\,re\,1\mu$Pa) an und zeigen, dass der erste Ansatz mit einem Durchschnittswert der MSE von $0.1306\,\text{dB}$ ($0.2253\,\text{dB}$) für den beschränkten (kompletten) Frequenzbereich deutlich unter den Durchschnittswert von $2.7395\,\text{dB}$ ($2.9758\,\text{dB}$) des zweiten und $0.1713\,\text{dB}$ ($0.3995\,\text{dB}$) des dritten Ansatzes liegen. Der bevorzugte Ansatz ist der Ansatz 1, jedoch ist der Vorteil des dritten Ansatzes die geringere Anzahl an Koeffizienten und Korrekturwerten, die bei der Implementierung über den ersten Ansatz für alle Kombinationen von Parametern bestimmt und bereitgestellt werden müssen. Die Wahl des Ansatzes in Bezug auf die Abweichung und den Ressourcenbedarf muss in der Umsetzungsphase unter Betrachtung der zur Verfügung stehenden Ressourcen und der Zielsetzung somit abgewogen werden.



(a) PSD - Ansatz 1 **(b)** Abw. ($f \in [f^-, f^+]$) **(c)** Abw. (komplett)

(d) PSD - Ansatz 2 **(e)** Abw. ($f \in [f^-, f^+]$) **(f)** Abw. (komplett)

(g) PSD - Ansatz 3 **(h)** Abw. ($f \in [f^-, f^+]$) **(i)** Abw. (komplett)

Ber (0 m/s, 0) Sim (0 m/s, 0) Ber (2 m/s, 0) Sim (2 m/s, 0) Ber (4 m/s, 0) Sim (4 m/s, 0)
Ber (6 m/s, 0) Sim (6 m/s, 0) Ber (8 m/s, 0) Sim (8 m/s, 0) Ber (10 m/s, 0) Sim (10 m/s, 0)
Ber (12 m/s, 0) Sim (12 m/s, 0) Ber (14 m/s, 0) Sim (14 m/s, 0) Ber (16 m/s, 0) Sim (16 m/s, 0)
Ber (18 m/s, 0) Sim (18 m/s, 0) Ber (20 m/s, 0) Sim (20 m/s, 0) Ber (22 m/s, 0) Sim (22 m/s, 0)
Ber (24 m/s, 0) Sim (24 m/s, 0) Ber (26 m/s, 0) Sim (26 m/s, 0) Ber (28 m/s, 0) Sim (28 m/s, 0)
Ber (30 m/s, 0) Sim (30 m/s, 0) Ber (32 m/s, 0) Sim (32 m/s, 0) Ber (340 m/s, 0) Sim (340 m/s, 0)
Ber (00 m/s, 1) Sim (00 m/s, 1) Ber (20 m/s, 1) Sim (20 m/s, 1) Ber (40 m/s, 1) Sim (40 m/s, 1)
Ber (60 m/s, 1) Sim (60 m/s, 1) Ber (80 m/s, 1) Sim (80 m/s, 1) Ber (100 m/s, 1) Sim (100 m/s, 1)
Ber (120 m/s, 1) Sim (120 m/s, 1) Ber (14 m/s, 1) Sim (14 m/s, 1) Ber (16 m/s, 1) Sim (16 m/s, 1)
Ber (18 m/s, 1) Sim (18 m/s, 1) Ber (20 m/s, 1) Sim (20 m/s, 1) Ber (22 m/s, 1) Sim (22 m/s, 1)
Ber (24 m/s, 1) Sim (24 m/s, 1) Ber (26 m/s, 1) Sim (26 m/s, 1) Ber (28 m/s, 1) Sim (28 m/s, 1)
Ber (30 m/s, 1) Sim (30 m/s, 1) Ber (32 m/s, 1) Sim (32 m/s, 1) Ber (340 m/s, 1) Sim (340 m/s, 1)

Abbildung 3.28: PSDs (Ansatz 1-3), mit Abweichung des Ähnlichkeitsmaßes

3.3.3.3 Modellierung mit programmierbaren FIR-Filtern

Um die Performanz-Analyse von Simulation und Emulation zu verbessern, wurde in den Veröffentlichungen [165] und [166] eine Implementierung des Umgebungsrauschens über FIR-Filter eingeführt, wodurch die qualitative Aussage über die Robustheit des analysierten Kommunikationssystems in einem definierten Frequenzbereich mit bekannten Rauscheigenschaften deutlich erhöht werden konnte.

Die Implementierung der FIR-Filter basiert auf der Programmierbarkeit während der Laufzeit. Diese (Online-) Programmierung ermöglicht es, vorgegebene oder dynamisch erzeugte Szenarien des Umgebungsrauschens abzubilden. Durch eine Erweiterung kann so auch die sich stetig ändernde Impulsantwort abgebildet werden. Dieser modellbasierte FIR-Filter-Ansatz ist damit sowohl für Simulationen als auch für Hardware-beschleunigte Emulation nutzbar und ermöglicht eine Vielzahl von Anwendungsmöglichkeiten.

Es ist jedoch wichtig, bei der Umsetzung der programmierbaren FIR-Filter für den Unterwasserkanal die Latenz zu berücksichtigen, insbesondere bei dynamischen Echtzeit-Anwendungen, um eine Verfälschung der Ergebnisse zu vermeiden. Die Latenz τ_{FIR} der FIR-Filter hängt von der Anzahl der zu ladenden Koeffizienten N_{taps} und der Abtastrate f_s des Filters ab und kann direkt durch $\tau_{FIR}{=}N_{\text{taps}}{\cdot}f_s$ bestimmt werden. Durch geschicktes Vorladen der Koeffizienten kann die Latenz verschleiert und der Einfluss auf die Simulation nahezu negiert werden. Es gibt neben der Latenz jedoch zwei wesentliche Einschränkungen des Ansatzes, i) die Schnelligkeit der Änderungen und ii) die Genauigkeit der Szenarien.

Die minimale Zeit zwischen den Änderungen, basierend auf der Anzahl der Koeffizienten, ergibt sich aus der ursprünglichen Latenz τ_{FIR} für das Rauschen und τ_H für die Impulsmatrix. Dabei wird die Abtastrate bei geschickter Umsetzung nicht durch den FIR-Filter, sondern durch die genutzte Hardware bestimmt. Die Genauigkeit der Umsetzung auf einem FPGA ist begrenzt durch die maximal nutzbare Anzahl von Stellen der Festkommazahlen und die damit verbundenen Anforderungen an die Hardwareressourcen. Diese Überlegungen sind entscheidend, um die Leistungsfähigkeit der Simulation zu gewährleisten und die realistische Abbildung des Unterwasserkanals zu erreichen.

Zusammenfassung

In diesem Kapitel wurden die Herausforderungen, Probleme und Limitationen des akustischen Unterwasserkanals und dessen Simulation eingeführt und diskutiert. Ein Arbeitsablauf zur Generierung eines komplexen, zeitlich variierenden Kanals mit einem definierten Set von Testpunkten wurde vorgestellt und definiert. Dabei wurde der Kanal sowohl mit BELLHOP dem innovativen geometrischen Modell generiert und verglichen.

Das für diese Arbeit entwickelte geometrische Kanalmodell, welches sich insbesondere bei mittleren und tiefen Gewässern als hervorragende Alternative erweist, kann zudem zur Verifizierung und für Prä-Evaluation-Tests von Kommunikationssystemen verwendet werden. Dieses Modell zeichnet sich durch eine Verringerung der benötigten Rechenzeit um über $99.8\,\%$ bei nahezu identischen Ergebnissen aus, was die Effizienz erheblich steigert.

Durch die gleichzeitige Reduktion der Komplexität kann der Ansatz auch von einfachen, autarken Systemen mit begrenzter Rechenleistung bewältigt und zur Verbesserung der Kommunikation genutzt werden. Diese Anwendungsmöglichkeit bedarf jedoch noch weiterer Untersuchungen, um ihre praktische Umsetzbarkeit vollständig zu validieren.

Der eingeführte Ansatz der Finite Impulse Response (FIR)-Filter zur Erzeugung von Umgebungsrauschen ermöglicht eine Szenario-basierte Generierung und Anpassung des Rauschens und der Positions- und Geschwindigkeitsbedingen CIR anhand veränderlicher umgebungsbedingter Parameter. Dies ist für die Performanz-Analyse unerlässlich und kann sowohl für die Frame-basierte Simulation in MATLAB, die Sample-basierte Simulation in Simulink als auch für Hardware-beschleunigte Simulationen aufFPGAs genutzt werden. Diese Flexibilität zeigt, dass mit diesem Ansatz ein Testset erstellt werden kann, das sowohl Hardwaretests als auch Simulationen für Frame- und Sample-basierte Ansätze unterstützt. Dadurch werden die generische Erzeugung und Testung sowie der Vergleich und die Abschätzung von Kommunikationssystemen jeglicher Art im Bereich der Unterwasserkommunikation ermöglicht, die in dieser Form und Komplexität bisher nicht möglich waren.

4

Rahmenerkennung

Die drahtlose akustische Kommunikation in Unterwasserumgebungen stellt eine komplexe und anspruchsvolle Herausforderung dar, die durch die Eigenschaften des Unterwasserkanals bedingt ist. Zusätzlich sind die Verzögerungen bei der Schallausbreitung unter Wasser aufgrund der niedrigen Ausbreitungsgeschwindigkeit des Schalls signifikant größer als bei elektromagnetischen Wellen in der Luft. Angesichts dieser Anforderungen benötigt die Entwicklung von effizienten und zuverlässigen Unterwasserkommunikationssystemen eine sorgfältige Gestaltung der Übertragungstechniken und eine genaue Berücksichtigung der physikalischen Eigenschaften des Unterwasserkanals. Dabei spielt auch die Begrenzung der Ressourcen eine wesentliche Rolle, insbesondere bei der Nutzung autonomer Systeme, nur über begrenzte Energie- und Rechenkapazitäten verfügen. Damit ist die Notwendigkeit, den Energieverbrauch zu minimieren, von der Signalverarbeitung hin zu den Protokollen zur Übertragung und Erkennung von Nachrichten, von zentraler Bedeutung.

Eine Schlüsselstrategie zur Reduzierung des Energiebedarfs ist die Implementierung verschiedener Betriebszustände für das Kommunikationssystem. Besonders die Nutzung von Stand-by- oder Sleep-Zuständen, bekannt als Low-Power-Detect für das μ-Modem [81], trägt zur Energieeinsparung bei, indem die Systemfunktionalitäten vorübergehend eingeschränkt werden. Im Stand-by-Modus befindet sich das System in einem Energiesparzustand, in dem nur ein minimaler Teil der Hardware aktiv bleibt, um auf ein Weck-Signal zu reagieren. Bei Empfang eines vordefinierten Weck- oder Detektionssignals wechselt das System in den aktiven Zustand, wodurch die gesamte Schaltung aktiviert wird und der Energiebedarf entsprechend steigt.

Neben der Energieeinsparung ist ein weiteres Hauptziel dieser Verfahren die zuverlässige Detektion und Decodierung von Daten. Damit bildet die Rahmenerkennung einen zentralen Aspekt der Energieeinsparungstechniken ab und spielt eine entscheidende Rolle in der drahtlosen akustischen Kommunikation, um sicherzustellen, dass Systeme nur dann aktiviert werden, wenn sie

tatsächlich ein relevantes Signal empfangen. Hierbei werden abhängig von der unterstützten Modulation spezifische Weck- oder Aktivierungssequenzen umgesetzt. Gängige Verfahren sind die Nutzung von CW-Pulsen mit unterschiedlichen Frequenzen [167, 168], Hüllkurvendetektion im Zeitbereich [169], PN-Sequenzen mit festem Trägersignal [170, 171] oder LFM-Signale [172, 173]. Die richtige Gestaltung und Implementierung von Weck-Sequenzen ist daher essenziell, um den Energieverbrauch niedrig zu halten und gleichzeitig eine zuverlässige Signalverarbeitung zu gewährleisten.

In der drahtlosen akustischen Unterwasserkommunikation haben sich derzeit zwei gängige Varianten für die Implementierung von Weck- oder Signalerkennung durchgesetzt:

(i) Die Weck- oder Aktivierungssequenz ist für alle Kommunikationssysteme in einem Netzwerk gleich, sodass bei Empfang einer definierten Sequenz alle Systeme in Reichweite in den aktiven Zustand wechseln.

(ii) Die Weck- oder Aktivierungssequenz ist spezifisch für jedes Modem definiert, sodass nur der adressierte Empfänger aktiviert wird. Diese Variante führt zu einem geringeren Leistungsbedarf, verhindert jedoch, dass nicht adressierte Kommunikationssysteme Informationen aus den empfangenen Paketen gewinnen, was insbesondere bei Handshake-Protokollen wie Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mit Request To Send (RTS)/Clear To Send (CTS)-Handshaking erforderlich ist, um den Kanal zu reservieren und Kollisionen zu vermeiden.

Ein Nachteil dieser Verfahren mit Weck-Sequenzen ist der Aufwand in Form des Datenaustauschs sowie der zusätzlich benötigten Rahmentypen. Daher sollten diese genutzten Verfahren einen signifikanten Energiebedarfsvorteil bieten, um den zusätzlichen Aufwand zu rechtfertigen.

In diesem Kapitel wird ein neuer Ansatz basierend auf Kasami-Folgen vorgestellt, der eine spezifische und Rahmen-bezogene Detektion für Handshake-basierte MAC-Protokolle ermöglicht. Dieses Verfahren, dessen Grundkonzept bereits in [174] veröffentlicht wurde, verbessert die energetische Effizienz, die Nutzdatenrate und ermöglicht die spezifische Auswahl von Präambel-Mengen, die den Anforderungen und der Umgebung entsprechen.

Im Abschnitt 4.1 wird das allgemeine Konzept zur Erkennung von Rahmentypen in der drahtlosen akustischen Unterwasserkommunikation vorgestellt. Abschnitt 4.2 führt die möglichen Sequenzen für eine Präambel-basierte Kommunikation ein, und Abschnitt 4.1.3 erläutert den Kasami-Ansatz und die Anpassung der Rahmentypen hinsichtlich Detektierbarkeit und Störanfälligkeit.

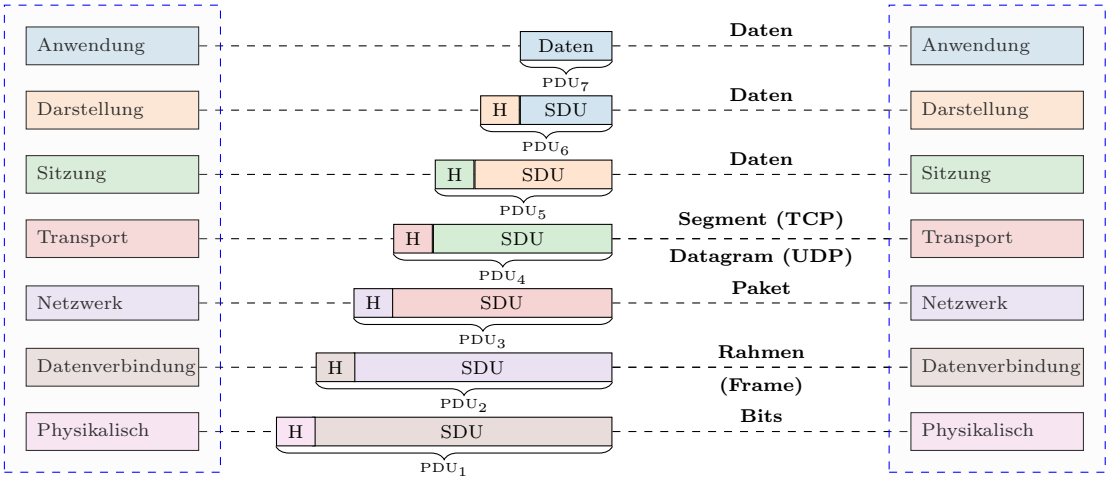


Abbildung 4.1: Schematische Kommunikation nach ISO/OSI [175, 176]

4.1 Rahmensynchronisation

Die schichtübergreifende Kommunikation im drahtlosen Bereich orientiert sich am genormten International Organization for Standardization/Open Systems Interconnection (ISO/OSI)-Referenzmodell (ISO/IEC 7498-1), dennoch haben sich bei der drahtlosen Kommunikation die Begriffe geändert. Im ISO-OSI-Modell ist die Protocol Data Unit (PDU) (Protocol Data Unit) die kleinste Datenstruktur, die von einer Schicht des Modells verarbeitet wird. Jede Schicht empfängt eine Service Data Unit (SDU) (Service Data Unit) von der darunterliegenden Schicht. Die empfangene SDU wird um spezifische Steuerinformationen der aktuellen Schicht ergänzt, um die PDU zu bilden. Diese PDU wird dann an die darüber liegende Schicht weitergegeben. Die genaue Bezeichnung der PDU für jede Schicht ist in Abbildung 4.1 beschrieben und wird im Folgenden verwendet.

Im Bereich der drahtlosen Kommunikation haben sich aber abweichend von der Norm eigene Begriffe entwickelt. Die kleinste Datenübertragungseinheit in drahtlosen Netzwerken auf der Media Access Control (MAC)-Schicht ist die Mac Protocol Data Unit (MPDU). Diese enthält sowohl Nutzdaten als auch MAC-spezifische Steuerinformationen, die für die Verwaltung des Zugriffs auf das Übertragungsmedium und die Fehlererkennung, wie die Quell- und Ziel-Adresse sowie den Rahmen-Typ, erforderlich sind.

Auf der physischen Schicht wird die MPDU zur Physical Channel Data Unit (PCDU) erweitert und entspricht damit der MPDU mit Informationen über

die Präambel. Auf der physikalischen Schicht wird die Physical Channel Data Unit (PCDU) abschließend in das Physical Protocol Data Unit (PPDU) eingebettet. Die PPDU repräsentiert die kleinste Datenübertragungseinheit (Bit-Folge) nach der anteiligen Kanalcodierung der PCDU, wobei nur die MPDU codiert wird. Die PPDU wird in der Regel direkt nach der Modulation über das Übertragungsmedium gesendet.

Eine Kommunikation in einem drahtlosen Medium zwischen zwei oder mehr Modems setzt voraus, dass die Empfänger die empfangenen Rahmen-Typen erkennen und synchronisieren können. Dazu werden hauptsächlich Präambeln mit guten Autokorrelationseigenschaften, wie die Barker-Codes und Unterklassen von Pseudo Noise (PN)-Folgen wie die Gold-Folgen (GF), Maximalfolgen (MF) oder Kasami-Folgen (KF) genutzt. Diese sind für eine synchronisierte Kommunikation wie bei Code Division Multiple Access (CDMA) in der Funktechnik, bei dem der Start des Rahmens bekannt ist, ausreichend. Im Gegensatz dazu stellen die Synchronisation und das Erkennen von Rahmentypen im Wasser höhere Anforderungen an die Eigenschaften der Präambeln.

In dieser Arbeit wird dazu ein neues Verfahren, basierend auf Kasami-Folgen, eingeführt. Dieses Verfahren verbessert die Synchronisation und die Erkennung von Rahmentypen über einen skalierbaren, an das Unterwasserkommunikationsnetz anpassbaren Präambel-Coderaum und dem dazugehörigen bestimmbaren Detektions-Schwellwert. Die über das Kasami-basierte Verfahren generierten Folgen können aufgrund der definierten Korrelationseigenschaften, sowohl periodisch als auch aperiodisch, für einen vordefinierten Schwellwert sicher detektiert werden können. Mittels des aufgespannten Coderaumes kann jeder generierten Folge ein Kommunikationssystem mit einer eindeutigen physischen Adresse (ID) oder Rahmen-Typ (DT) mit definierter Länge N zugeordnet werden.

Bei Unterwasserkommunikationsnetzwerken wird zwischen kleinen (2 bis 10 Teilnehmern), mittelgroßen (11 bis 100 Teilnehmern), großen (über 100 bis 1000 Teilnehmern) und sehr großen Netzwerken (über 1000 Teilnehmern für zukünftige Unterwasser-Internetinfrastrukturen) unterschieden. Der Vergleich zwischen dem Barker-Code, den M-Folgen, GF, KF der kleinen Gruppe und den orthogonalen KF sowie den Kasami-basierten Folgen gerader und ungerader Länge erfolgt anhand der Anzahl der zuordenbaren Folgen (Definition 4.1) im Verhältnis zum Minimum der Autokorrelationsgüte ASR definiert in Definition 4.3 und der Kreuzkorrelationsgüte $ACSR$ in Definition 4.4. Diese Definitionen basieren auf der allgemeinen Auto- und Kreuzkorrelation (Definition 4.2 und Definition 3.3) für (a)periodische diskrete Signale.

Definition 4.1 — Zuordbarkeit:
Für eine eindeutige Zuordnung aller Folgen i, j in der Menge der Folgen $F{:=}\{ID, DT\}$ und z in der Menge der kombinierten Folgen $KF{:=}\{ID{\times}DT\}$ muss gelten, dass die (a)periodische Kreuzkorrelation zwischen jeder Folge i und allen anderen Folgen in F und KF, sowie die Autokorrelation von i mit einer Verschiebung von $k{>}0$, unterhalb eines Schwellenwerts R_{th} liegen müssen. Gleichzeitig muss die Autokorrelation $C_i[k]$ bei einer Verschiebung von $k{=}0$, größer oder gleich dem Schwellwert R_{th} sein.

$$\max\left(\max_{i\neq j||k\neq 0}(C_{i,j}[k], C^p_{i,j}[k]), \max_{k\neq 0}(C_i[k], C^p_i[k]), \max_{i\not\subset z||k\neq 0}(C^p_{i,z}[k])\right) {<} R_{th} {\leq} C_i[0]$$

Die Autokorrelationsgüte ASR_i (Definition 4.3) ermöglicht eine generelle Aussage, wie gut sich die Folge $i(t)$ gegenüber einer zeitlich verschoben Folge $i(t{+}\tau)$ unterscheidet. Je höher die Autokorrelationsgüte ist, desto eindeutiger ist die zeitliche Synchronisation und umso geringer kann ein Detektions-Schwellwert gesetzt werden. Autokorrelationsgüten mit einem Wert nahe 1 zeigen, dass es sehr starke Ähnlichkeiten zwischen der Folge $i(t)$ und ihrer Verzögerung $i(t{+}\tau)$ gibt, was eine Synchronisation stark erschwert und bei Gütewerten kleiner 1 nahezu unmöglich macht.

Definition 4.2 — Autokorrelation:
Die Autokorrelation kann in die periodische und aperiodische Autokorrelation unterteilt werden. Bei der aperiodischen Autokorrelation sind die Werte vor und nach einer Folge nicht bekannt und werden mittels Nullsetzen (Zero-Padding [177]) ausgeblendet. Sei M die Matrix aller Folgen und $M(i,:)$ die i'te diskrete Folge der Länge N mit den definierten Symbolwerten $\{-1, +1\}$, dann ist die normierte aperiodische Autokorrelation in abhängig der Verschiebung k in Symbolen wie folgt gegeben:

$$C_i[k] = \begin{cases} \frac{1}{N-|k|}\sum_{n=0}^{N-k-1} M(i,n)M(i,n{+}k) & k{\geq}0 \\ \frac{1}{N-|k|}\sum_{n=-k}^{N-1} M(i,n)M(i,n{+}k) & k{<}0 \end{cases} \tag{4.1}$$

Unter der Annahme, dass die diskrete Folge x der Länge N periodisch ist, wird die periodische Autokorrelation, nach Sarwate und Pursley [153] über $C^p_i[k] {=} C^p_i[-k]^*$ wie folgt bestimmt:

$$C^p_i[k] = \frac{1}{N}\sum_{n=0}^{N-1} M(i,n)M(i,(n{+}k) \mod N) \qquad |k|{\geq}0 \tag{4.2}$$

Definition 4.3 — Autokorrelationsgüte ASR:
Sei $M(i,:)$ die i'te diskreten Folge, dann beschreibt die ASR_i das Verhältnis zwischen der aperiodischen Autokorrelation der diskreten Folge i der Länge N ohne Verschiebung ($k{=}0$) und dem Nebenmaximum, welcher dem maximalen aperiodischen Autokorrelationswert der Folge $M(i,:)$ für $k{\neq}0$ entspricht:

$$ASR_i = \frac{C_i[0]}{\max_{k\neq 0}\left(|C_i[k]|\right)} \tag{4.3}$$

Die periodische ASR_i^p entspricht dann der Güte unter Verwendung der periodischen Autokorrelation:

$$ASR_i^p = \frac{C_i^p[0]}{\max_{k\neq 0}\left(|C_i^p[k]|\right)} \tag{4.4}$$

Die Kreuzkorrelationsgüte $ACSR_i$ (Definition 4.4) erweitert die Aussage der Güte einer Folge auf ein Set von Folgen und gibt das Verhältnis der normierten Autokorrelation (Gl. (4.1)) von i zum Maximum der absoluten Kreuzkorrelation $C_{i,j}[k]$ aller anderen zu betrachteten Folgen $j{\in}M$ an. Damit quantisiert die $ACSR_i$, wie gut eine eindeutige Zuordnung der Folge möglich ist.

Definition 4.4 — Kreuzkorrelationsgüte $ACSR$:
Sei $M(i,:)$ die i'te diskreten Folge, dann beschreibt die $ACSR_i$ das Verhältnis der aperiodischen Autokorrelation der diskreten Folge i ohne Verschiebung ($k{=}0$) von $M(i,:)$ zum Maximum der aperiodischen Kreuzkorrelation aller anderen Folgen $j{\in}M$ für $i{\neq}j$:

$$ACSR_i = \frac{C_i[0]}{\max_{i\neq j}\left(|C_{i,j}[k]|\right)} \tag{4.5}$$

Die periodische $ACSR_i^p$ entspricht dann der Güte unter Verwendung der periodischen Kreuzkorrelation:

$$ACSR_i^p = \frac{C_i^p[0]}{\max_{i\neq j}\left(|C_{i,j}^p[k]|\right)} \tag{4.6}$$

Ein Wert der Kreuzkorrelationsgüte $ACSR_i{>>}1$ indiziert eine ausgezeichnete Codezuordnung, was bei Werten nahe und kleiner 1 nicht mehr gegeben ist, da sich die Folgen entweder stark ähneln oder es sich nur um eine Phase der Folge i handelt. Um Folgen für eine Zuordnung zwischen den Rahmen-Typen und der System-ID auf die Code-Folgen zu nutzen, müssen die Auto- und Kreuzkorrelationsgüte größer als 1 sein und bestimmen damit auch die untere

Schranke des Detektions-Schwellwert R_{dth} in Gl. (4.7) über den Kehrwert der minimalen Güte:

$$R_{dth} \geq \frac{1}{\min\left(ACSR_i, ASR_i\right)} \tag{4.7}$$

4.1.1 Barker-Code

Die nach Ronald Hugh Barker benannten Barker-Codes [178] der Länge N, abgebildet auf die Symbole $\{+1,-1\}$, sind optimal, da diese, mit Ausnahme des Null-Verschiebung $v=0$ immer kleiner gleich 1 sind:

$$|C_x[k]| = |\sum_{k=1}^{N-v} x[k] \cdot x[k+v]| \leq 1 \qquad \forall k \neq 0 \text{ und } 1 \leq v \leq N \tag{4.8}$$

Die Längen bekannter Barker-Codes sind auf $N=\{2,3,4,5,7,11,13\}$ beschränkt und wird vermutet, dass es keine weiteren Barker-Codes mehr gibt [178]. Der Barker-Code wird aufgrund der hervorragenden Autokorrelation zur Rahmensynchronisation in digitalen Kommunikationssystemen hauptsächlich in Bereichen der P2P-Kommunikation verwendet, in denen die Kreuzkorrelationseigenschaften keine Rolle spielen. Die Barker-Codes sind daher für eine Adressierung von Modems oder der Rahmentypen über unterschiedliche Code-Folgen aufgrund der geringen Anzahl nicht geeignet. Barker-Codes verfügen aber über ausgezeichnete Autokorrelationseigenschaften für eine generelle Synchronisation bei optimalen Übertragungsbedingungen.

4.1.2 Pseudo Noise (PN)-Folgen

PN-Folgen hingegen sind Sequenzen von Symbolen, die den Anschein von Zufälligkeit haben, aber durch deterministische Algorithmen, wie durch Linear-Feedback Shift Register (LFSR), erzeugt werden können [179, 180]. Sie werden in verschiedenen Anwendungen wie Multi-User-Kommunikation, Kryptografie, Radar und GPS eingesetzt. Die durch Schieberegister erzeugten PN-Folgen sind periodische Signale, wodurch eine Änderung des Initialworts des Schieberegisters keine neue Folge generiert, sondern nur die ausgehende Folge maximaler Länge zyklisch verschiebt und als Phase bezeichnet wird. Die bekanntesten Beispiele der Klasse der PN-Folgen sind die Maximalfolgen (MF) [181], Gold-Folgen (GF) [182, 183] und der Kasami-Folgen (KF) [184].

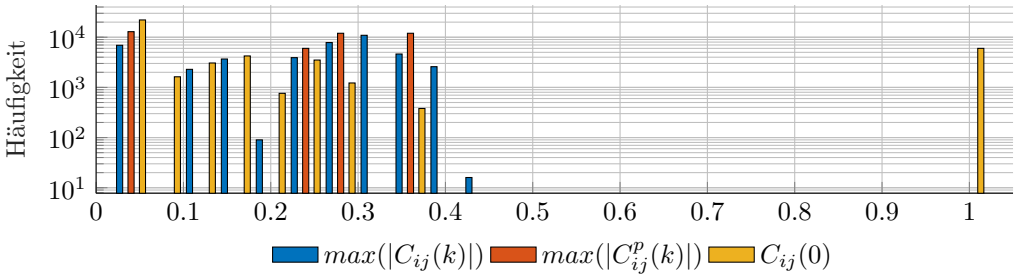


Abbildung 4.2: Häufigkeitsverteilung der maximalen Kreuzkorrelation von ausgeglichenen MF für $n{=}6$

4.1.2.1 Maximalfolgen (MF)

Die Maximalfolgen (MF), auch unter den Begriffen Maximum-Length-Sequences, Folgen maximaler Länge oder m-Sequenzen bekannt, können durch lineare Rückkopplung mithilfe von LFSR und dazugehöriger irreduzibler Polynome (Tabelle A.1) erzeugt werden:

$$x_n{=}x_{n-a}{\oplus}x_{n-b}{\oplus}\ldots{\oplus}x_{n-k} \tag{4.9}$$

Hierbei steht x_n für die Bits der MF, $\oplus$ für die XOR-Operation und $a,b,\ldots,k$ definiert, die Rückkopplungsstellen des Generaltorpolynom $g(x)$ im Register. Von Ziemer wurde gezeigt [185], dass die längste mögliche sich nicht wiederholende Folge aus maximal $2^m{-}1$ Nullen und Einsen besteht, wenn das Schieberegister über m Stufen (Speicherelemente) verfügt. Die Wahl der Rückkopplungsstellen und die Anordnung der XOR-Verknüpfungen bestimmen dabei entscheidend die Länge und die Eigenschaften der MF.

Durch die maximale Periodenlänge besitzen MF hervorragende Autokorrelationseigenschaften mit einem Korrelationswert von -1 für alle verschobenen Korrelationen basierend auf der ungeraden Anzahl von Symbolen und sind somit nach Gl. (4.8) optimal. Die MF sind aber nicht notwendigerweise orthogonal zueinander und unterschiedlich ausgeprägte Kreuzkorrelationseigenschaften aufweisen. Allerdings ist die Anzahl der möglichen Folgen beschränkt auf die Anzahl der irreduziblen primitiven Polynome mit jeweils einer möglichen Folge. Die erzeugbaren Folgen durch zyklische Verschiebung der längsten Folge weisen nur eine schlechte periodische Kreuzkorrelationsgüte von $ACSR^p_i{\approx}1$ auf. Auch die Folgen, erzeugt mit verschiedenen irreduziblen primitiven Polynomen, zeigen untereinander eine experimentell bestimmte aperiodische Kreuzkorrelationsgüte von $ACSR^p_i{\approx}\frac{1}{0.36}{\approx}2.78$, wie die Abbildung 4.2, über alle Polynome (Tabelle A.1) und über Definition 4.5 bestimmte Initialworte (Tabelle A.3), auf [185, 186].

Definition 4.5 — M-Folgen :
Eine M-Folge ist eine binäre Folge (nicht notwendigerweise 0 und 1), die einer linearen Rekursionsrelation (Rekurrenzrelation) genügt, welche durch ein primitives Polynom definiert wird. Die Periodenlänge N einer M-Folge entspricht $N=2^m-1$, für ein primitives Polynom m'ter Ordnung. Eine M-Folge muss zusätzlich folgende Eigenschaften, definiert von Golomb [179], erfüllen:

i) Die Anzahl der binären Symbole muss ausgeglichen sein. Für die ungerade maximale Länge von $N=2^m-1$ müssen 2^{m-1} Symbole der Eins und $2^{m-1}-1$ Symbole der Null entsprechen, respektive ± 1.

ii) Die Anzahl der auftretenden Untersequenzen m_i mit i gleicher aufeinanderfolgender Symbole ist auf $m_i=2^(m-i)$ beschränkt.

iii) Die periodische Autokorrelation $C_i^p[k]$ (Definition 4.2) nähert sich, für eine bipolare duale Symbolmenge $\{+1,-1\}$, der Kronecker Delta Funktion:

$$C_i^p[k]=\begin{cases}1 & \text{für } k=0,\\ -\frac{1}{N} & \text{für } 0<k<N-1.\end{cases} \tag{4.10}$$

Aus der Häufigkeitsverteilung der MF in Abbildung 4.2 ergibt sich zudem die aperiodische Kreuzkorrelationsgüte von $ACSR_i\approx\frac{1}{0.39}\approx 2.56$ was eine Unterscheidung von MF grundsätzlich auf Kosten der guten Autokorrelation ermöglicht. Die normierten 1-Werte der Häufigkeitsverteilung in Abbildung 4.2 ergeben sich für die Kreuzkorrelation von identischen Folgen ($i=j$) mit einer Verschiebung von $k=0$, was der Autokorrelation entspricht. Für die Bestimmung aller (a)periodischen Kreuzkorrelation in dieser Arbeit wurde bei $i=j$ der höchste Wert der Korrelation für $k\neq 0$ ausgewählt, was dem Nebenmaximum entspricht. Auch bei einer synchronisierten Übertragung von Daten ($k=0$) ist die Güte, vorgreifend, in jedem Bereich gleich gut oder schlechter bei einem wesentlich kleineren Set von Folgen im Vergleich zu Kasami- und Gold-Folgen. Damit sind die MF nur für sehr kleine Netze unter idealen Bedingungen, die in der akustischen Unterwasserkommunikation überwiegend nicht gegeben sind, oder für die generelle Synchronisation von Rahmen oder Spreizverfahren nutzbar. Eine individuelle und unabhängige Detektierung und Synchronisation von System-Identifikation und Datenrahmen ist mit MF somit erst ab einer Ordnung $n\geq 7$ für kleine Netze umsetzbar.

4.1.2.2 Gold-Folgen (GF)

Die nach Robert Gold benannten GF werden über die Kombination zweier MF; u und w der Länge N durch die XOR-Verknüpfung über Definition 4.6 erzeugt. Durch die Kombination von MF können die positiven Eigenschaften selbiger beibehalten und gleichzeitig eine längere Pseudozufallsfolge generiert werden. Allerdings weisen die GF keine so guten Autokorrelationseigenschaften wie die MF auf [182]. Die Werte der Kreuzkorrelation und der Autokorrelation mit $k \neq 0$ von GF nehmen nur die drei Werte -1, $-t(n)$ und $t(n)-2$ an [183]. Für den Grad n des Polynoms ist $t(n)$ definiert als:

$$t(n) = \begin{cases} 2^{(n+1)/2}+1 & n \text{ ist ungerade} \\ 2^{(n+2)/2}+1 & n \text{ ist gerade} \end{cases} \tag{4.11}$$

Definition 4.6 — Gold-Folgen [182, 183] :
Die 2^n+1 GF $G(u,w)$, der Länge $N=2^n-1$ mit $n \in \mathbb{N}^+$, werden über die Multiplikation von u und w jeweils aus einem Polynom des bevorzugten Polynompaares (Tabelle A.2) der Ordnung n und der zyklischen Verschiebung T_{lsh} generiert:

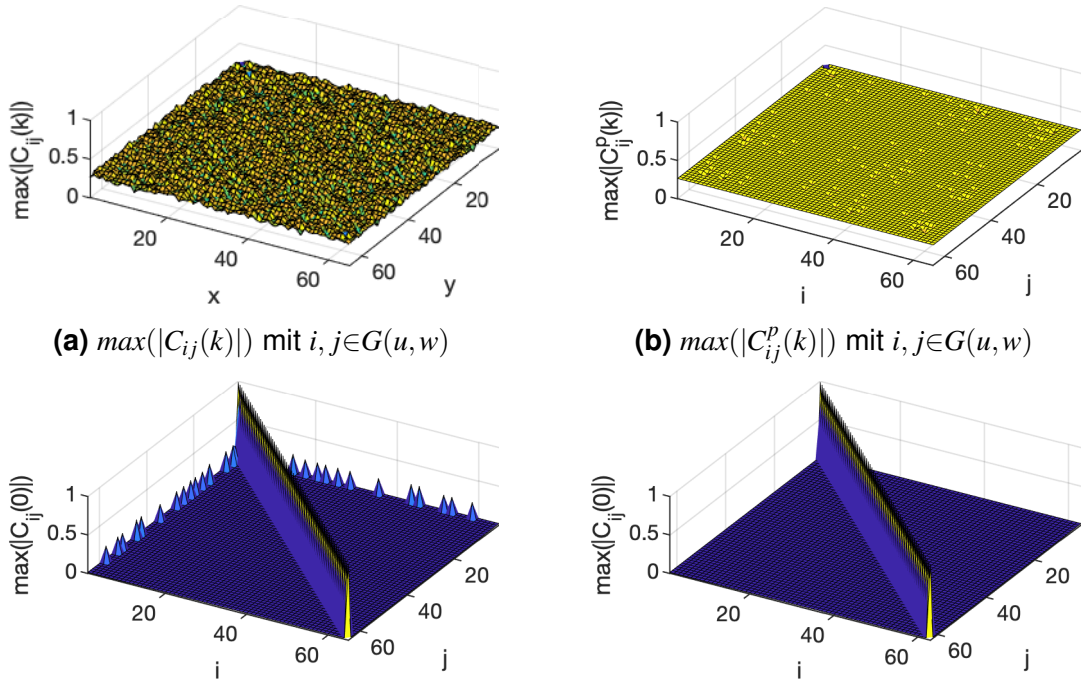
$$G(u,w) = u \oplus {T_{lsh}}^m w \qquad \text{mit } m := (0:N-1) \tag{4.12}$$

$$G(u,w) = (u, w, u \oplus w, u \oplus T_{lsh} w, \ldots, u \oplus {T_{lsh}}^{N-1} w) \tag{4.13}$$

Eine Unterklasse der GF sind die Balanced Gold Sequence (BGS) bei diesen ist das Verhältnis der verwendeten Symbole ausgeglichen, was sich insbesondere bei Spreizverfahren vorteilhaft auswirkt. Bestimmt durch die obere Schranke der Nebenwerte, mit einer periodischen Autokorrelationsgüte von $ASR_i^p = 63$ und einer periodischen Kreuzkorrelationsgüte von $ACSR_i^p = 3.7$ zeigen die GF innerhalb eines Sets ausgezeichnete Autokorrelations- und Kreuzkorrelationseigenschaften (Abbildung 4.3). Bei einer Reduzierung des Sets der Ordnung $n=6$ auf die GF $G(u,w)^*$, über Gl. (4.14), kann die periodische Kreuzkorrelationsgüte für $k=0$ auf $ACSR_i^p = ASR_i^p = 63$, was der Länge N der Folgen entspricht, erhöht werden.

$$G(u,w)^* = (u \oplus w, u \oplus T_{lsh} w, \ldots, u \oplus {T_{lsh}}^{N-1} w) \tag{4.14}$$

Die Reduzierung des Sets hat keinerlei Einfluss auf die (a)periodische Kreuzkorrelationsgüte. Die aperiodische Kreuzkorrelationsgüte wurde, für alle Verschiebungen, experimentell unter Verwendung einer zufälligen Auswahl bevorzugter Polynompaare (Tabelle A.2) und Initialworte (Tabelle A.3) für ausgeglichene MF ermittelt, wobei die Kreuzkorrelationsgüte $ACSR_i \approx \frac{1}{0.39} \approx 2.56$ beträgt. Diese Werte



(a) $max(|C_{ij}(k)|)$ mit $i,j \in G(u,w)$ **(b)** $max(|C^p_{ij}(k)|)$ mit $i,j \in G(u,w)$

(c) $max(|C_{ij}(0)|)$ mit $i,j \in G(u,w)$ **(d)** $max(|C_{ij}(0)|)$ mit $i,j \in G(u,w)^*$

Abbildung 4.3: Kreuzkorrelationen einer (reduzierten) Gold-Folge für $n=6$

sind identisch mit den Set-übergreifenden periodischen $ACSR$-Eigenschaften der MF. Die periodische Kreuzkorrelationsgüte hingegen beträgt $ACSR^p_i \approx \frac{1}{0.28} \approx 3.57$. Daher eignen sich die Gold-Folgen hervorragend für eine individuelle und unabhängige Detektierung sowie Synchronisation von System-Identifikation und Datenrahmen, besonders in kleinen Netzen. Bei einer optionalen Erweiterung durch die Kombination von Sets, die durch die Anforderungen an die Korrelationseigenschaften begrenzt sind (Abbildung 4.9), zeigen sie auch für mittelgroße Netze herausragende Leistung. Einschränkungen ergeben sich lediglich durch die ungerade Anzahl von Symbolen und die Begrenzung der Ordnung.

4.1.2.3 Kasami-Folgen (KF)

Die Kasami-Folgen (KF), eingeführt von Tadao Kasami [187], leiten sich von den MF ab und lassen sich in zwei Hauptgruppen unterteilen: die kleine und die große Gruppe von KF. Diese Sequenzen werden mithilfe von Linear-Feedback Shift Register (LFSR) erzeugt und finden aufgrund ihrer ausgezeichneten

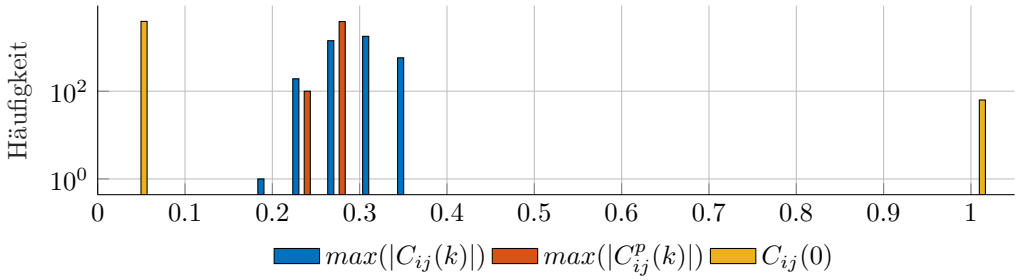


Abbildung 4.4: Häufigkeitsverteilung von $G(u,w)$ für n=6

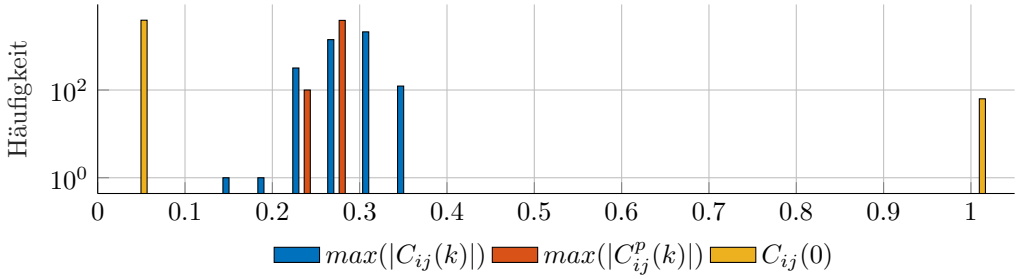


Abbildung 4.5: Häufigkeitsverteilung von $G(u,w)^*$ für n=6

Kreuzkorrelationseigenschaften vielfältige Anwendungen, insbesondere im Bereich von CDMA und Spreizverfahren.

Die in dieser Arbeit verwendete kleine Gruppe von KF (Definition 4.7) kann ähnlich wie die Gold-Folge mithilfe von zwei LFSR erzeugt werden. Die Beschränkung auf eine gerade Ordnung schränkt zwar die Länge und Anzahl der kleinen Gruppe von KF stark ein, jedoch zeichnen sich diese Sequenzen durch ihre hervorragenden Auto- und Kreuzkorrelationseigenschaften aus.

Definition 4.7 — Kleine Kasami-Gruppe [188]:
Die $2^{n/2}$ Kasami-Folgen der kleinen Kasami-Gruppe mit der Länge $N=2^n-1$, wobei $n\in\mathbb{N}^+$ eine gerade Anzahl an Speicherstellen des LFSR bezeichnet, können wie folgt erzeugt werden:

$$K_s(u,n,m)=u\oplus {T_{lsh}}^m w, \quad \text{mit } m:=0,1,\ldots,2^{n/2}-2, \tag{4.15}$$

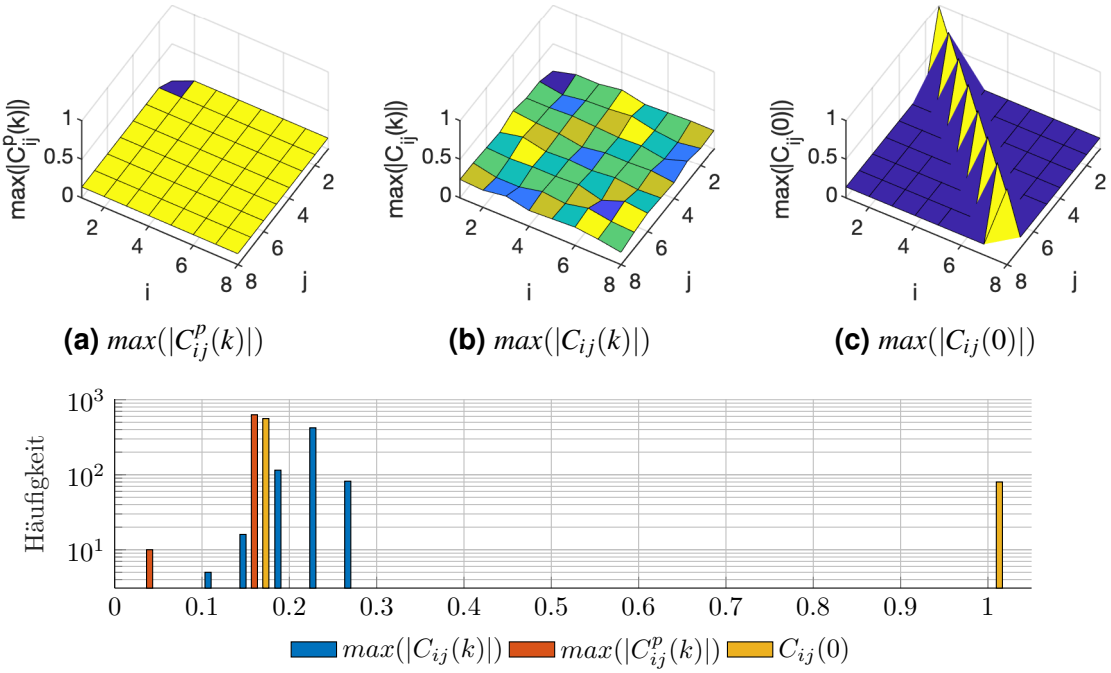
$$K_s(u,n,m)=(u,u\oplus w_{sh},u\oplus T_{lsh}w_{sh},\ldots,u\oplus {T_{lsh}}^{2^{n/2}-2}w_{sh}) \tag{4.16}$$

Dabei ist u eine M-Sequenz der Länge N (in Bit) und w eine aus u durch Dezimierung mit dem Faktor $(2^{n/2}+1)$ abgeleitete M-Sequenz. Der T_{lsh} bezeichnet den Linksverschiebungsoperator, und $m\in\mathbb{N}$ ist der Verschiebungsparameter für die Sequenz w.

Die Werte der periodischen Kreuzkorrelation der kleinen Gruppe von KF sind gemäß Gl. (4.11) [184] auf $-\frac{1}{2}(1+t(n))$, -1 und $\frac{1}{2}(1+t(n))-2$ beschränkt. Dies bedeutet, dass der maximale Kreuzkorrelationswert, als obere Schranke, für die kleine Gruppe von KF niedriger ist als die obere Schranke der GF gleicher Ordnung.

Die große Gruppe von KF bezieht sich auf eine umfassendere Menge von Kasami-Sequenzen, die nicht notwendigerweise geringe Kreuzkorrelationseigenschaften aufweisen müssen und mit einem zusätzlichen LFSR erzeugt werden. Die Ordnung n der großen Gruppe ist dabei noch stärker beschränkt, nämlich auf $n \bmod 4=2$, wird jedoch an dieser Stelle nicht weiter behandelt. Kasami-Folgen zeigen durch ihre nahezu halb so großen Kreuzkorrelationswerte, wie im Beispiel der Ordnung $n=6$ in Abbildung 4.6 dargestellt, eine bessere Eignung als die M- und Gold-Folgen. Allerdings sind die Kasami-Folgen aufgrund ihrer begrenzten Anzahl und der Verwendung nur gerader Ordnungen wesentlich eingeschränkt.

Die reinen Kasami-Folgen eignen sich daher nur für winzige Netze und bieten eine begrenzte Anwendbarkeit für eine individuelle und unabhängige Detektierung sowie Synchronisation von System-Identifikation und Datenrahmen, ähnlich den MF. Jedoch kann durch die Anwendung eines hybriden Ansatzes [184], basierend auf der kleinen Kasami-Gruppe (Definition 4.7), die Anzahl der erzeugten KF mit orthogonalen Eigenschaften auf Kosten der Korrelationsgüte erhöht werden.



(a) $max(|C^p_{ij}(k)|)$ **(b)** $max(|C_{ij}(k)|)$ **(c)** $max(|C_{ij}(0)|)$

(d) Häufigkeitsverteilung

Abbildung 4.6: Häufigkeitsverteilung Kreuzkorrelation von KF für $n{=}6$ mit zufälligen Initialworten

4.1.2.4 Orthogonale Kasami-Folgen

Aus einer MF (u der Länge N können über die kleine Kasami-Gruppe $N{\cdot}\sqrt{N{+}1}$ orthogonale Kasami-Folgen (OKF) erzeugt werden. Dies basiert auf dem Ansatz von Chandra und Chattopadhay [184], ähnlich dem von Donelean und O'Farrell für orthogonale M-Folgen [189]. Die Länge der orthogonalen Kasami-Folgen beträgt dabei $N{+}1$, und sie weisen einen maximalen periodischen Kreuzkorrelationswert von $C^p_{i,j}[0]{=}\sqrt{N{+}1}$ auf (Definition 4.8). Die orthogonale Eigenschaft der Kasami-Folgen bezieht sich dabei nur auf die Verschiebung von $k{=}0$.

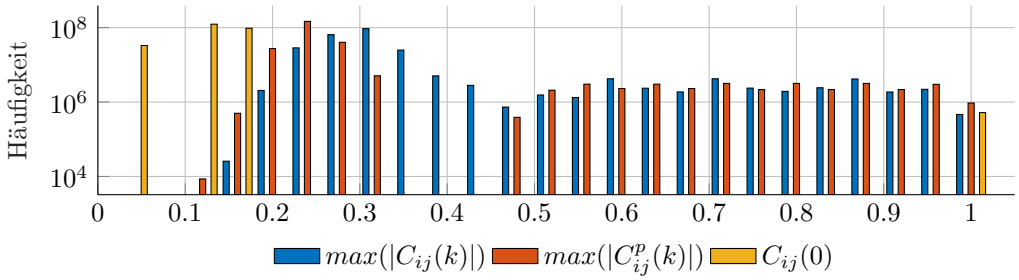


Abbildung 4.7: Häufigkeitsverteilung der Kreuzkorrelation von KF für $n=6$

Definition 4.8 — orthogonale Kasami-Folgen (OKF) [184]:
Die n OKF der Matrix ID werden über die Zwischenmatrix H und der Neuordnung über Gl. (4.17), mit einer Länge von $N+1$ und einem maximalen periodischen Kreuzkorrelationswert von $C^p_{i,j}[0]=\sqrt{N+1}$ erzeugt.

$$ID(k,:)=(1,H(k,1{:}n-1),-H(k,0)) \tag{4.17}$$

Die Zwischenmatrix H wird über die Multiplikation der KF $a=(a_0,a_1,\ldots,a_{n-1})$ und $b=(b_0,b_1,\ldots,b_{n-1})$ der Länge N aus einem Set der kleinen Kasami-Gruppe und dem Linksschiebeoperator T_{lsh} erzeugt:

$$H(k,:)=\begin{cases} {T_{lsh}}^l a \times {T_{lsh}}^k\{b_i\} & 0\leq k,l<n-1 \\ {T_{lsh}}^l a & k=n-1 \text{ und } 0\leq l<n-1 \end{cases} \tag{4.18}$$

Mit k und l als Linksschiebeparameter der KF a und b.

Die OKF haben eine periodische Kreuzkorrelation ohne Verschiebung (synchronisiert) von $C^p_{i,j}[0]=\sqrt{N+1}$. Allerdings weisen sie eine periodische und aperiodische Kreuzkorrelationsgüte von $ACSR_i=ACSR^p_i=1$ auf (Abbildung 4.7). Das disqualifiziert diese Folgen für die gewünschte Anwendung, da keine eindeutige Zuordnung gewährleistet werden kann. Bei einem direkten Vergleich der Anzahl der Folgen zwischen den orthogonalen Kasami-Folgen und den Gold-Folgen für $n=6$, erzeugt durch ein primitives irreduzibles Polynompaar (Abbildung 4.9), im folgendem als Set bezeichnet, sind die Gold-Folgen aufgrund ihrer besseren Korrelationseigenschaften zu bevorzugen.

Dennoch lässt sich aus dem Set der erzeugten orthogonale Kasami-Folgen (OKF) eine Teilmenge bilden, die einer vordefinierten Auto- bzw. Korrelationsgüte entspricht und ist in Relation zur Güte in Abbildung 4.9 abgebildet. Bei einer Güte von $\min(ASR,ACSR)<3$ beinhalten diese Teilmenge mehr Folgen als die KF bei äquivalenter Güte.

4.1.2.5 Orthogonale Kasami-basierte Folgen

Die im Rahmen dieser Arbeit durchgeführten Experimente haben gezeigt, dass eine Teilmenge von Folgen der Länge 2^n mit $n{:=}\{5,6,7,8\}$, erzeugt aus den orthogonalen Kasami-Folgen höherer Ordnung eine hervorragende Auto- und Kreuzkorrelationseigenschaft, wie die Abbildung 4.8 beispielhaft zeigt, aufweisen.

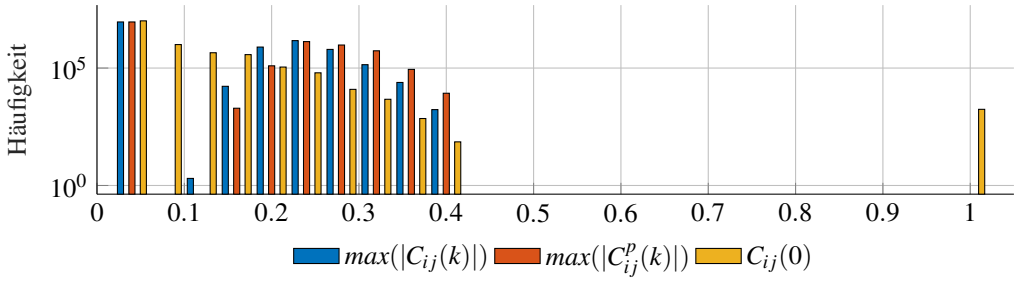


Abbildung 4.8: Häufigkeitsverteilung der Kreuzkorrelation von KF für $n{=}7$

Mit diesem neuen Ansatz ist zudem die Erzeugung von Kasami-basierten Präambeln variabler Länge möglich, welche in dieser Arbeit auf die Längen $N{=}\{16,32,64,128,256\}$ beschränkt sind. Diese Längen sind im Allgemeinen keine Beschränkung des Verfahrens. Für die Abbildung von Folgen auf System-ID und Rahmen-Typen sind die Folgen nicht auf die einzelnen Sets beschränkt und können unter Berücksichtigung der Korrelationseigenschaften selektiv ausgewählt und zusammengefasst werden.

4.1.2.6 Orthogonale Kasami-basierte Folgen ungerader Länge

Bei der Betrachtung der orthogonalen Kasami-basierte Folgen gerader Länge fällt auf, dass die Eigenschaften hervorragend sind und die Anzahl möglichen Folgen sehr groß ist, aber für eine höhere ASR und $ACSR$ von den Kasami-Folgen (KF) der kleinen Gruppe übertroffen werden. Durch eine Anpassung des Verfahrens zur Erzeugung von Präambeln mit einer ungeraden Länge $2^n{-}1$ kann sichergestellt werden, dass auch für höhere Güten die Menge Kasami-Folgen der kleinen Gruppe eine untere Schranke bilden, da diese nun eine Teilmenge der orthogonale Kasami-basierte Folgen ungerader Länge (OKBFU) sind. Eine Regel zur Erzeugung der Teilmengen, die den Güteanforderungen definiert über die den Kehrwert des Detektionsschwellwertes R_{dth} zur sicheren Zuordnung entspricht, kann derzeit mit keinen bekannten Regeln erzeugt werden. Für die

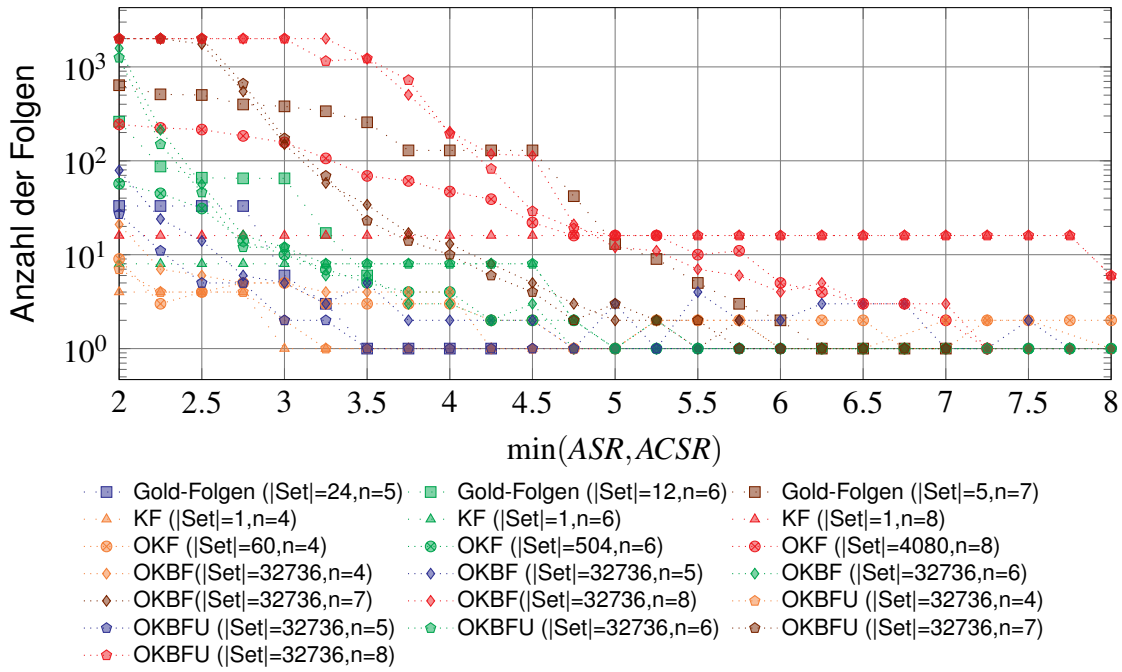


Abbildung 4.9: Vergleich der Anzahl von Folgen in Relation zur Güte

Anwendung, die nur vorgehaltene Folgen nutzt, reicht die derzeitige Erzeugung und Güte-abhängige Selektion der Teilmenge aus.

Die Abbildung 4.9 dazu Teilmenge von systematisch erzeugten Folgen, aus verschiedenen irreduziblen primitiven Polynomen in Relation zur angeforderten Korrelationsgüte ermittelt wurden. Die abgebildete Güte auf der x-Achse bildet das Minimum aus periodischer und aperiodischer Korrelations- und Autokorrelationsgüte. Für die Erzeugung der Teilmengen von Folgen, welche die Anforderungen an die Güte erfüllen, wurde eine obere Schranke von ≈ 2000 Folgen festgelegt. Die Abbildung zeigt, dass die Teilmengen der orthogonalen Kasami-basierten Folgen bei gleicher Güte zum Teil eine viel höhere Anzahl von Folgen im Vergleich zu M-, Gold-, Kasami- und orthogonalen Kasami-Folgen zur Verfügung stellen können. Die in der Arbeit genutzte Güte wurde auf 3.0 und die Präambel auf orthogonale Kasami-basierte Folgen (OKBF) gerader Länge festgelegt, womit über den Kasami-basierten Ansatz Netze von über 2000 Teilnehmern erstellt werden könnten.

4.1.3 Rahmenzuordnung

Nach erfolgreicher Generierung einer ausreichenden Teilmenge von orthogonalen Kasami-basierten Folgen kann jedem Modem eine Folge als Adresse (ID), eine Folge als allgemeine Broadcast-Adresse und jeweils eine Folge für jeden genutzten Rahmen-Typen eindeutig zugeordnet werden.

Somit hält jedes Modem bei der Verwendung von RTS/CTS-basierte Protokolle (Carrier Sense Multiple Access (CSMA), CSMA/CA) mindestens sechs OKBF im Empfänger und eine Folge pro Modems des Netzwerks im Sender vor. Bei n Kommunikationssystemen sind daher maximal $n+6$ orthogonale Kasami-basierte Folgen (OKBF) erforderlich, welche von den eingeführten OKBF abgedeckt werden können. Die Skalierbarkeit der OKBF ermöglicht auch eine einfache Erweiterung für Rahmen-Typen, die unter anderem Parameter für die Decodierung spezifizieren oder bereits etablierte Rahmen-Typen wie Contention-Free-End-Frame (CF-End) oder Rahmen-Typen zur Kollisionsdetektion hinzufügen.

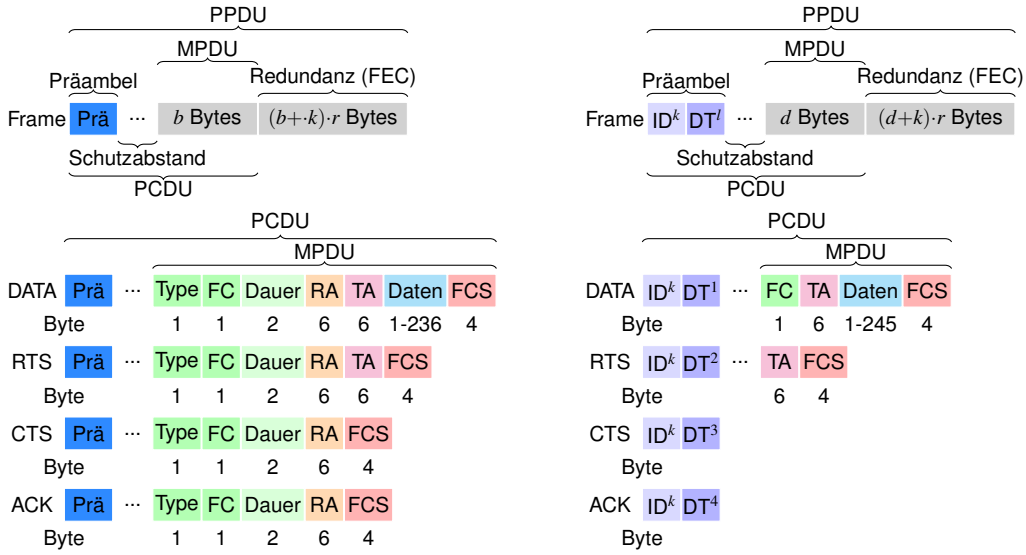
4.2 Rahmenoptimierung

Durch die Implementierung einer zweifachen Präambel mit einer eindeutigen Zuordnung der Rahmen-Typen (DT^k) und der Adresse des Modems (ID^l) kann auf die direkte Kennzeichnung des Typs in der ursprünglichen Datenrahmenstruktur (vgl. Abbildung 4.10a) verzichtet werden. Alle Flags in der Rahmentypkennung (Abbildung 4.10, frame control (FC)) sind im Fall der Unterwasserkommunikation auf Eins gesetzt und daher für den ersten Ansatz vernachlässigbar, können aber, wie in Abbildung 4.10b illustriert, einmalig im Datenrahmen übertragen werden.

Die Festlegung der Kommunikationsdauer (Abbildung 4.10, Dauer) stellt im Bereich der akustischen Unterwasserkommunikation eine Herausforderung dar, insbesondere dann, wenn die Positionen der Kommunikationsteilnehmer unbekannt sind. Zudem erschweren die Reflexionen und die gemeinsame Nutzung des Mediums die Dauer zusätzlich und sind schwer zu erfassen.

Anstatt einer dynamischen Anpassung kann die maximale Kommunikationsdauer auch im Voraus abgeschätzt werden, basierend auf wesentlichen Einflussfaktoren wie der Symbolrate, der maximalen Kommunikationsreichweite und dem Schutzabstand zwischen Präambel und Datenrahmen.

Unter der Annahme, dass die Übertragungszeiten der einzelnen Rahmen-Typen entweder im Vorfeld festgelegt oder dynamisch bestimmt werden, erübrigt sich die Angabe der Dauer innerhalb des Rahmens. Durch den Kasami-basierten



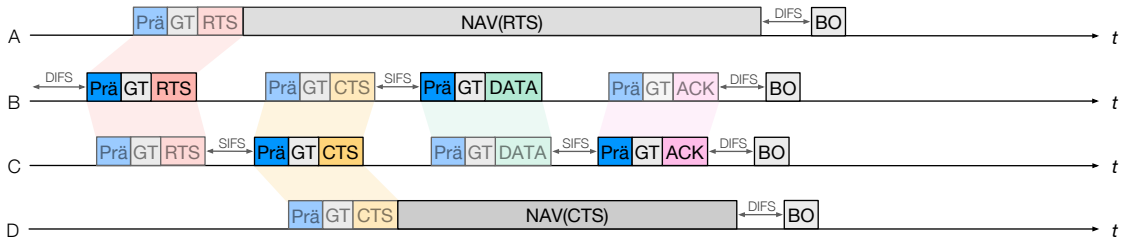
(a) Rahmenstruktur mit einfacher Präambel **(b)** Rahmenstruktur mit OKBF

Abbildung 4.10: Rahmenstrukturen des RTS/CTS Kommunikationsprotokoll

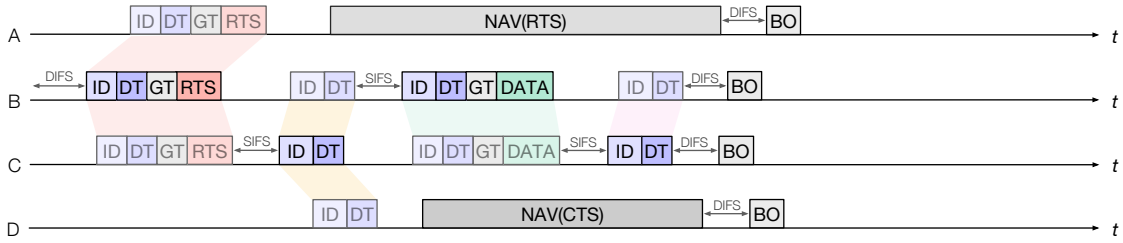
Ansatz können die CTS- und ACKnowledgement (ACK)-Pakete somit ohne zusätzliche Daten (Abbildung 4.10b) übertragen werden, was den Aufwand für Codierung, Decodierung und den optionalen Wechsel aus dem Stand-by-Modus deutlich reduziert. Auch der bisher erforderliche Schutzabstand zwischen Präambel und dem Rest des Rahmens wird nur noch benötigt, um die Überlagerung beim Austausch von RTS- und DATA (DATA)-Rahmen bei Bedarf zu verhindern.

Die Implementierung dieses Verfahrens reduziert zudem den Energieaufwand und die Belegungsdauer des Unterwassermediums bei gleichbleibender Präambel-Länge deutlich, was bereits in [174] aufgezeigt wurde. Die energetische Betrachtung wurde am Beispiel des etablierten Mikromodems (Version 1 und 2) der Woods Hole Oceanographic Institution (WHOI) [190] theoretisch berechnet. Diese Arbeit hat gezeigt, dass in Abhängigkeit der Größe des Kommunikationsnetzwerkes mit dem Einsatz der Kasami-basierten Folgen ohne zusätzliche Wake-up-Mechanismen der Energiebedarf im Vergleich zu anderen Präambel-basierten Ansätzen für die Mehrnutzerkommunikation erheblich reduziert werden kann.

Für den Daten- und RTS-Rahmentypen ermöglicht das Kasami-basierte Verfahren zudem eine zeitnahe Informationsübermittlung (Cross-Layer), was eine effizientere Kommunikation mit der übergeordneten Sicherheitsschicht und



(a) Austausch der RTS/CTS-Kontrollpakete mit jeweils einer Präamble



(b) Austausch der RTS/CTS-Kontrollpaketen unter Verwendung von OKBF zur Detektion von Rahmen-Typ und System-ID

Abbildung 4.11: Gantt-Diagramme zum Austausch der PPDUs unter Verwendung des RTS/CTS-Handshake-Protokolls mit B als Sender (mit A und C in Empfangsreichweite) und C als Empfänger (mit B und D in Empfangsreichweite)

damit eine Reduzierung der Verarbeitungszeit ermöglicht. Weiterhin erlaubt der Kasami-basierten Ansatz eine implizite Kollisionserkennung. Im Falle der Erkennung eines anderen oder mehrfach empfangenen Datenrahmens aufgrund der Korrelationseigenschaften der Präambeln können Kommunikationssysteme innerhalb der Reichweite oder der adressierte Sender unmittelbar über einen zusätzlich einzuführenden Rahmentyp informiert werden, um Gegenmaßnahmen zu ergreifen.

4.3 Rahmenerkennung

Im Folgenden wird die Verwendung von orthogonalen Kasami-basierten Folgen zur verbesserten Detektion von Datenrahmen unter dem Fokus der Robustheit und Skalierbarkeit für Kommunikationsnetze betrachtet und diskutiert.

In der akustischen Unterwasserkommunikation ergeben sich für die Robustheit einer Präambel, je nach Anwendung, neben der Zuorden- und Skalierbarkeit

zusätzliche Anforderungen an die Detektierbarkeit und Störanfälligkeit im genutzten Medium. Es wird im Folgenden gezeigt, wie diese Eigenschaften die Leistungsfähigkeit des Kommunikationssystems beeinflussen und welche Auswirkungen diese auf die Zuverlässigkeit der Datenübertragung haben können.

Für eine Menge von Folgen, für die ein Schwellwert existiert, der kleiner der Autokorrelation mit einer Verschiebung von $k{=}0$ und größer als das Maximum der (a)periodischen Autokorrelation für $k{\neq}0$ und der Kreuzkorrelation mit allen anderen Folgen der Menge ist, können die Folgen eindeutig detektiert und zeitlich zugeordnet werden. Durch die fallenden Werte der Autokorrelation an der Stelle $k{=}0$ mit sinkenden Signal-Rausch-Verhältnis (SNR) muss der Schwellwert zusätzlich auf einen gültigen SNR-Bereich begrenzt werden, in dem die Folgen detektierbar sind.

4.3.1 Detektierbarkeit

Für die Detektierbarkeit der OKBF über die Autokorrelation, wobei die Bedingung gemäß Definition 4.1 erfüllt sein muss, sollen die Folgen auch resistent gegenüber Störungen, in Form von Umgebungsrauschen, sein. Damit auch unter den widrigen Umständen des Unterwasserkanals die Rahmen eindeutig zugeordnet werden können. Über die Formalisierung der Detektierbarkeit in Axiom 4.1 in Relation zum SNR kann ein Schwellenwert zur Detektion R_{dth} der Folgen bestimmt werden.

Axiom 4.1 — Detektierbarkeit:
Für alle orthogonalen Kasami-basierten Folgen (OKBF) $i, j{\in}KF$ in der Menge $KF{:=}ID, DT$ und einer Rauschsequenz n mit einem festgelegten SNR muss gelten, dass die Kreuzkorrelation zwischen jeder Folge i und allen anderen Folgen $j^*[k]{=}j[k]{+}n[k]$ und $z^*[k]{=}z[k]{+}n[k]$ mit $z{\in}ID{\times}DT$ sowie die Autokorrelation von i und $i^*[k]{=}i[k]{+}n[k]$ mit einer Verschiebung $k{>}0$ gleich oder kleiner dem Detektions-Schwellwert R_{dth} sein müssen. Die Autokorrelation $C_{i,i}[k]$ bei einer Verschiebung von $k{=}0$ muss größer sein als der Detektions-Schwellwert:

$$\max(\max_{i\neq j}(C_{i,j^*}[k]), \max_{i\ni z}(C_{i,z^*}[k]), \max_{k\neq 0}(C_{i,i^*}[k])) \leq R_{dth} < C_{i,i^*}[0] \tag{4.19}$$

Für den Nachweis wurden in dieser Arbeit empirisch, für die Präambel-Längen von $N{:=}13$ für den Barker-Code und $N{:=}\{32, 64, 128, 256\}$ für die OKBF, OKBFU

und GF bezüglich Ihrer Auto- und Kreuzkorrelationseigenschaften untersucht. Dazu wurden jeweils Tausend Testläufe pro SNR-Wert mittels dem in das Basisband transferierte additives weißes gaußsches Rauschen (AWGN) genauer gesagt Unterwasserrauschen berechnet und visualisiert (Abbildungen 4.12 bis 4.23). Die daraus resultierende Übersicht von Schwellwerten basierend auf dem niedrigsten SNR-Wert (SNR^{-}), an dem eine eindeutige Detektierbarkeit noch gegeben ist, wurden tabellarisch in Tabellen 4.1 bis 4.4 zusammengefasst.

Für die Generierung des bandbegrenzten Basisband-Rauschens $n_{BB}(t)$ mit definierten SNR wurde das AWGN-basierte Rauschen im Passband durch die Multiplikation mit der Trägerfrequenz und Tiefpassfilterung (Butterworth-Filter der Ordnung 4) umgesetzt. Abschließend wurde die durchschnittliche Rauschleistung im gewählten Frequenzband, entsprechend der jeweils definierten SNR bezüglich der Präambel, korrigiert.

Die Autokorrelation der Präambeln (für die Kasami-basierten Folgen ID^i) wurde über den SNR-Bereich von $15\,\mathrm{dB}$ bis $-30\,\mathrm{dB}$ gebildet. Die Abbildung 4.12 zeigt die durchschnittliche periodische und Abbildung 4.14 die aperiodische Autokorrelation. In Abbildung 4.13 und Abbildung 4.15 wird jeweils das Maximum für $k{\neq}0$ und das Minimum für $k{=}0$ der Autokorrelation dargestellt. Während die Abbildung 4.16 und Abbildung 4.17 das Maximum der periodischen und aperiodischen Kreuzkorrelation aller Folgen mit der Folge ID^i in Abhängigkeit des SNR visualisieren. Die zu vergleichenden Folgen ergeben sich aus den systematisch gesuchten und vorgehaltenen Folgen, deren Minimum der ASR und $ACSR$ größer oder gleich 3.0 ist. Eine Ausnahme bildet der Barker-Code, da es keine weiteren Folgen der Länge 13 für eine Kreuzkorrelation gibt und daher die Anwendung für ein Präambel-basiertes Mehrbenutzerschema nicht infrage kommt.

Alle Abbildungen werden (rechts) mit dem SNR-abhängigen Minimum der Autokorrelation für $k{=}0$ (in Blau) und dem maximalen Wert der Autokorrelation mit $k{\neq}0$ und der Kreuzkorrelation (in Magenta), für einen visuell vereinfachten Vergleich ergänzt. In Rot ist jeweils zusätzlich der ermittelte Detektions-Schwellwert R_{dth} aus Tabellen 4.1 bis 4.4 zur Orientierung abgebildet.

Für eine realistische Betrachtung der Folgen bezüglich der aperiodischen Auto- und Kreuzkorrelation wurde die Funktion der Korrelation angepasst. Die mittels Zero-Pading hinzugefügten Null-Werte wurden basierend auf dem definierten SNR verrauscht. In der Unterwasser-Kommunikation wird immer Rauschen empfangen und ein Zero-Pading ohne Rauschen würde die Ergebnisse zu optimistisch verfälschen. Des Weiteren wird in dieser empirischen Untersuchung davon ausgegangen, dass das genutzte Kommunikationssystem

einen Entscheider besitzt, der die eingehenden verrauschten Signale einem eindeutigen Symbol (in diesem Fall $\{-1, 1\}$) zuordnet.

Zusammenfassend zeigen die Abbildungen 4.12 bis 4.23 das für alle Längen der OKBF, im Gegensatz zum Barker-Code, ein Schwellwert ungleich 1 existiert, für den eine AWGN verrauschte Präambel, periodisch als auch aperiodisch, mit hoher Wahrscheinlichkeit ohne Falsch-Detektion die Identität und der Datenrahmentyp detektiert werden kann. Es besteht aber immer eine von der Länge der Präambel abhängige Wahrscheinlichkeit, dass das Rauschen so geformt ist, dass die Abfolge der Präambel im Rauschen zu erkennen ist.

Bei einer Analyse der durchschnittlichen Korrelation in Abbildung 4.12 und Abbildung 4.14 zeigt sich, dass eine theoretisch sichere Detektion für alle Präambeln bis zu einer Leistung von $-12\,\mathrm{dB}$ möglich ist. Allerdings ergibt die Untersuchung, dass die aperiodische Kreuzkorrelation den maßgeblichen Detektionsschwellwert bestimmt, während die durchschnittliche Korrelation keine signifikante Aussagekraft bezüglich der Detektierbarkeit besitzt. Mit zunehmender Präambellänge nimmt die Wahrscheinlichkeit ab, dass das Rauschen die Präambel verfälscht, was durch die abfallenden Schwellwerte der aperiodischen Kreuzkorrelation in Abbildung 4.17 deutlich wird. Auch eine Untersuchung ohne verrauschte Null-Werte (Zero-Padding) bei der aperiodischen Kreuz- und Autokorrelation würde zu einem optimistischeren (niedrigeren) Detektionsschwellwert führen und damit die Wahrscheinlichkeit einer falschen Detektion erhöhen, was nicht vernachlässigt werden darf. Zusätzlich zur Verwendung des etablierten AWGN wurde die Detektierbarkeit auch hinsichtlich des eingeführten Unterwasserrauschens untersucht. Vorangegangene simulationsbasierte Experimente haben bereits gezeigt [155], dass die Verwendung von AWGN im Vergleich zum anzunehmenden Unterwasserrauschen zu optimistischeren Ergebnissen führen kann.

Die Nutzung des FIR-Filter-Ansatzes zur Erzeugung des Unterwasserrauschens und die Anwendung desselben Ablaufs resultieren in den kompakten Abbildungen 4.18 bis 4.21 and 4.23. Diese Abbildungen zeigen im Vergleich zu den Abbildungen 4.12 bis 4.15 and 4.17, dass der Einfluss des Unterwasserrauschens nicht vernachlässigbar ist.

Durch die Auswertung des gesamten Testbereichs kann der Detektionsschwellwert basierend auf der Länge N und dem Minimum der Güten ASR und $ACSR$ über das Maximum der Korrelation bestimmt werden. Diese Werte sind in den Tabellen 4.1, 4.3 und 4.4 für die OKBF, GF und OKBFU für AWGN sowie zusätzlich für die OKBF unter Verwendung des Unterwasserrauschens in Tabelle 4.2 zusammengefasst. Dadurch kann die Präambellänge nun auch basierend auf den Anforderungen der Kommunikationssysteme, wie dem abgedeckten Bereich des SNR, ausgewählt und gegen die Kosten in Form der Präambellänge und

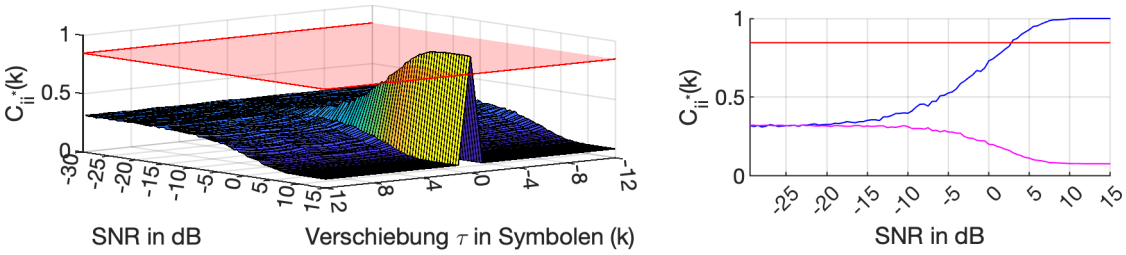
der Komplexität des Korrelators abgewogen werden. Eine zusammenfassende Visualisierung der Korrelationen von GF sowie der Ergebnisse in Tabelle 4.4 für eine minimale Güte von 3.0 ist zusätzlich in Abbildung A.4 bis Abbildung A.9 zum ergänzenden Vergleich abgebildet.

Die Zusammenfassung der Ergebnisse in den Tabellen 4.1 and 4.2 zeigt deutlich, dass die Wahl und Entscheidung hinsichtlich der Präambellänge und -güte unter Berücksichtigung des Unterwasserrauschens getroffen werden sollte. Der Schwellwert der sicheren Detektion weicht zwischen den Ergebnissen der beiden Analysen um bis zu $2\,\mathrm{dB}$ positive wie negativ ab und führt zu einer Abweichung des Schwellwerts um bis zu $6\,\%$, was Auswirkungen auf die Performanz zur Folge haben kann. Entgegen der ersten Annahme weisen die Präambeln damit im Unterwasserkanal nur für kurze Präambeln schlechtere Eigenschaften auf.

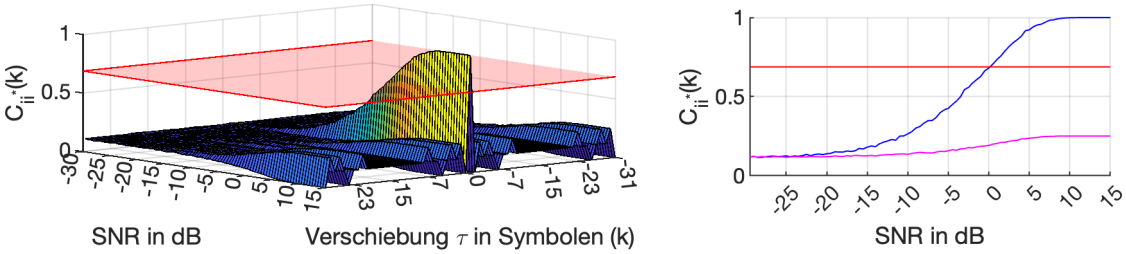
Für längere Präambeln hingegen zeigen die Präambeln bessere Detektionseigenschaften auf. Die Betrachtung mit dem AWGN kann daher weder als obere noch untere Schranke genutzt werden und erfordert bei einer notwendigen genauen Analyse des ortsspezifischen Unterwasserrauschens.

	32 Bit			64 Bit			128 Bit			256 Bit		
Güte	$\mathbf{R_{dth}}$	$\mathbf{SNR^{-}}$	#	$\mathbf{R_{dth}}$	$\mathbf{SNR^{-}}$	#	$\mathbf{R_{dth}}$	$\mathbf{SNR^{-}}$	#	$\mathbf{R_{dth}}$	$\mathbf{SNR^{-}}$	#
8.00	–	–	0	–	–	0	–	–	0	0.328	1.5	1
7.75	–	–	0	0.500	5.0	1	–	–	0	0.313	−0.5	1
7.50	–	–	0	0.469	4.0	1	–	–	0	0.320	0.0	1
7.25	–	–	0	0.469	5.5	1	–	–	0	0.313	0.0	1
7.00	–	–	0	0.438	4.0	1	–	–	0	0.313	0.0	1
6.75	–	–	0	0.438	4.0	1	–	–	0	0.336	0.5	1
6.50	–	–	0	0.500	6.5	1	0.500	5.0	1	0.336	0.0	2
6.25	–	–	0	0.438	6.5	1	0.422	3.0	1	0.359	1.5	1
6.00	0.500	8.5	1	0.469	6.0	1	0.406	3.5	1	0.352	1.0	2
5.75	0.500	9.5	1	0.438	5.5	1	0.375	2.5	1	0.305	−1.0	3
5.50	0.563	10.5	1	0.469	5.5	1	0.438	4.0	1	0.313	−0.5	6
5.25	0.625	8.5	1	0.500	5.5	1	0.406	4.0	2	0.313	0.0	7
5.00	0.500	7.0	1	0.438	4.5	1	0.422	3.5	2	0.352	0.0	10
4.75	0.625	9.5	1	0.500	6.0	1	0.422	3.5	3	0.336	−1.0	11
4.50	0.688	10.0	1	0.531	6.0	1	0.438	4.0	3	0.336	−0.5	32
4.25	0.625	10.0	1	0.438	3.5	1	0.422	4.0	7	0.352	0.0	122
4.00	0.688	10.0	1	0.500	6.5	2	0.438	3.5	9	0.359	0.0	305
3.75	0.688	11.5	1	0.531	6.0	3	0.469	3.5	12	0.375	0.5	974
3.50	0.563	7.0	2	0.594	9.5	5	0.438	4.0	26	0.414	2.5	1888
3.25	0.563	7.5	2	0.563	8.0	4	0.500	3.5	51	0.391	0.5	2000
3.00	0.688	10.0	3	0.563	8.0	9	0.500	4.0	270	0.391	1.0	2000
2.75	0.563	9.5	2	0.563	7.5	14	0.500	5.0	1136	0.555	3.5	2000
2.50	0.688	9.5	8	0.625	9.0	39	0.578	6.0	2000	0.469	2.0	2000
2.25	0.813	9.5	14	0.625	8.0	257	0.547	5.5	2000	0.492	3.0	2000
2.00	0.813	10.0	50	0.656	9.0	2000	0.672	6.5	2000	0.602	4.0	2000

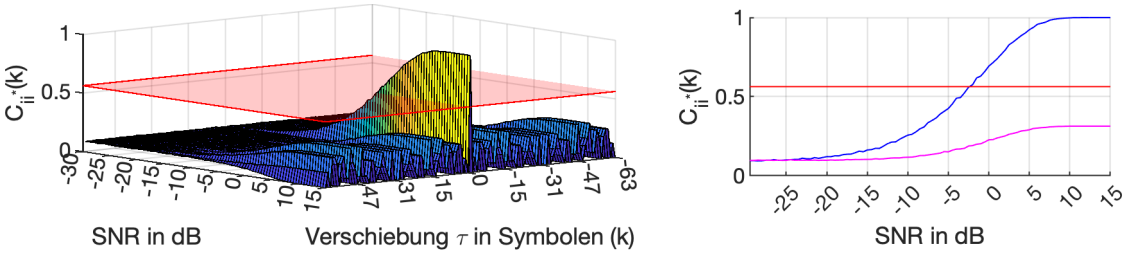
Tabelle 4.1: Detektionsschwellwerte der OKBF mit minimalen SNR^{-} (AWGN)



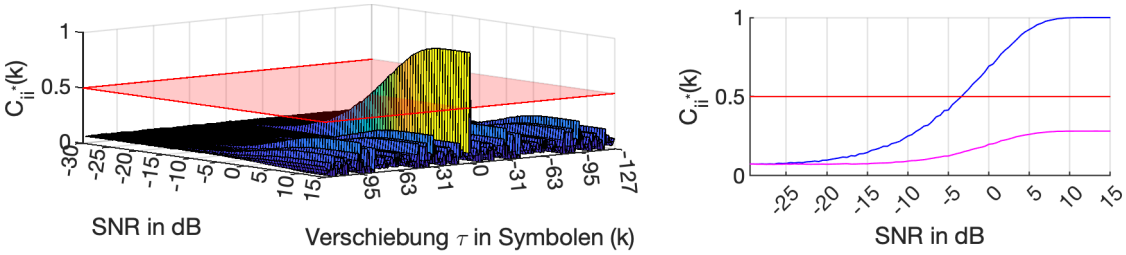
(a) Barker-Code (13 Bit)



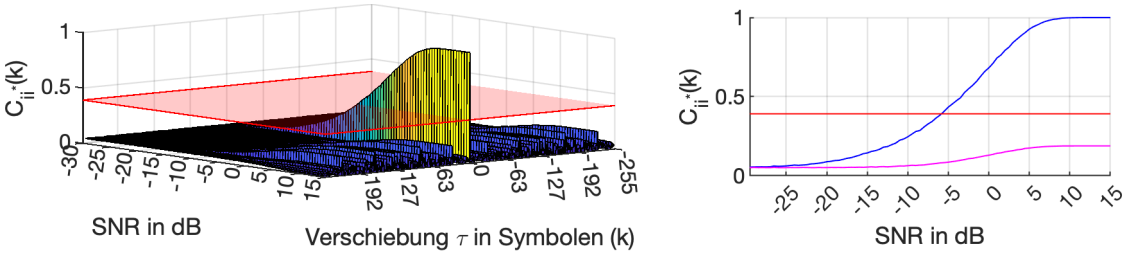
(b) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(c) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

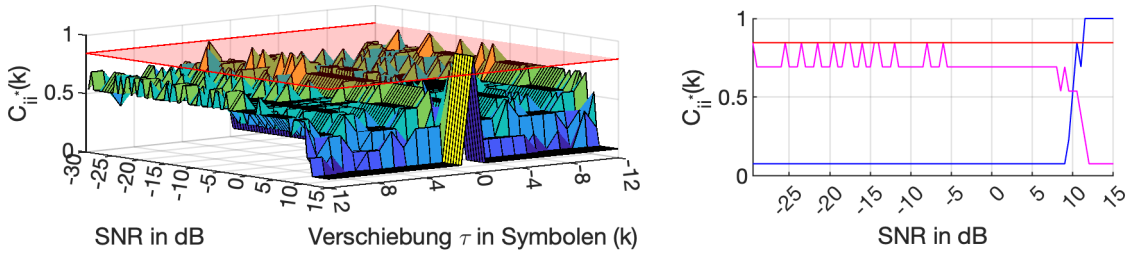


(d) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)

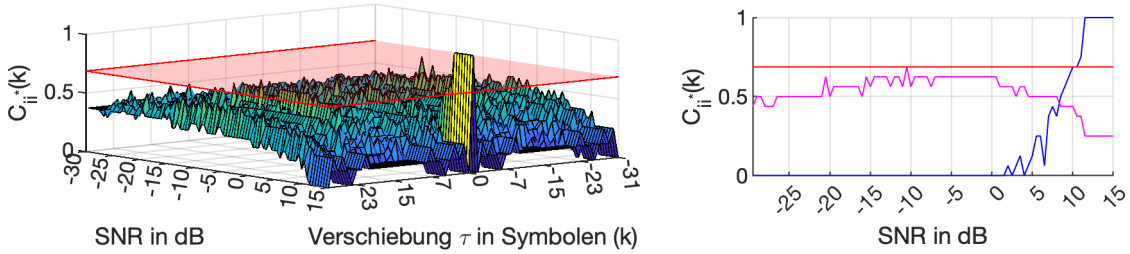


(e) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

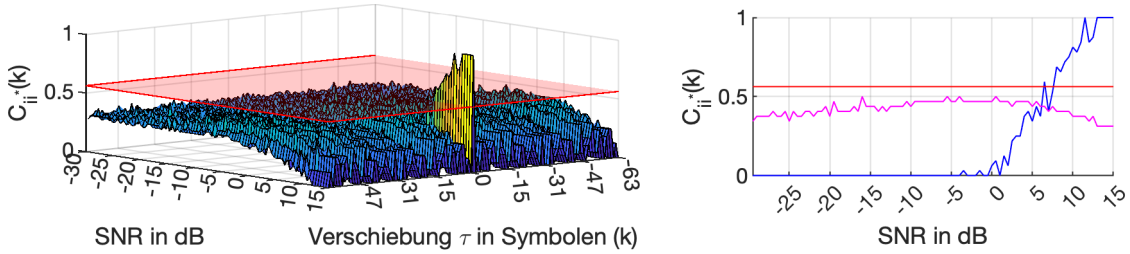
Abbildung 4.12: Durchschnittliche periodische Autokorrelation für $k{=}0$ (rechts Blau) und dem für $k{\neq}0$ (rechts Magenta) über 1000 Testläufe



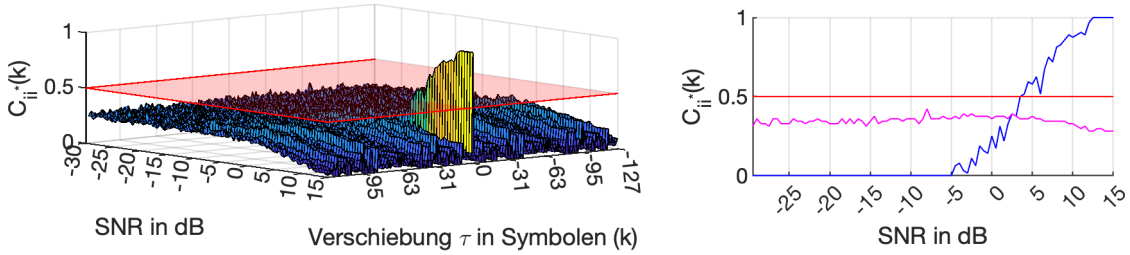
(a) Barker-Code (13 Bit)



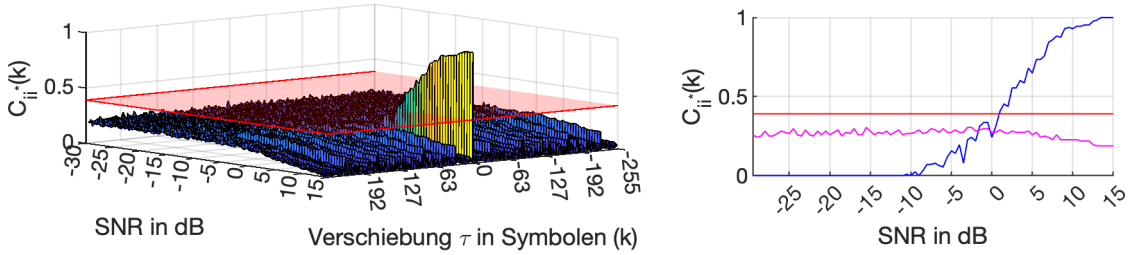
(b) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(c) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

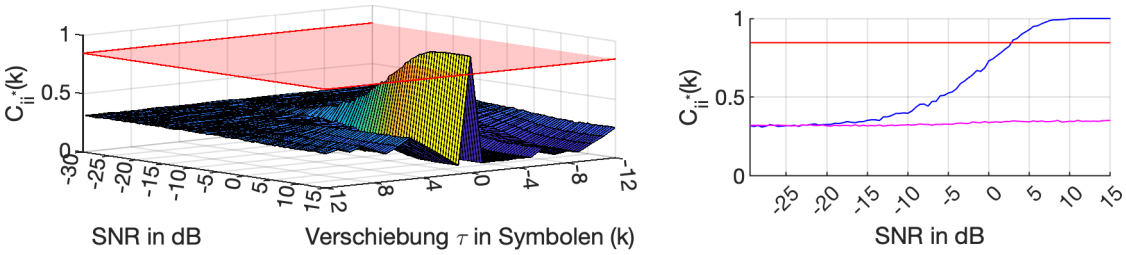


(d) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)

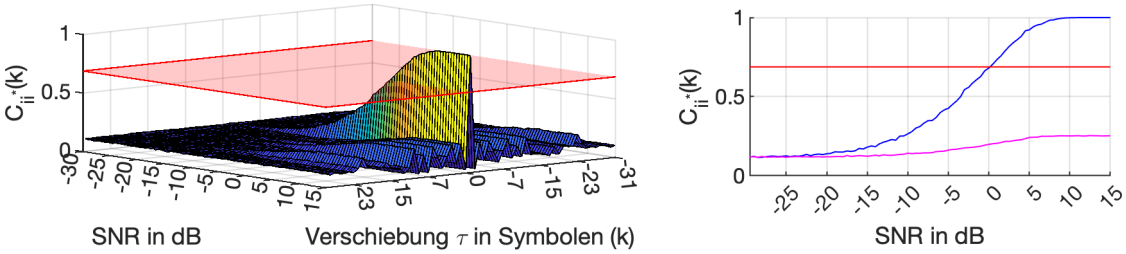


(e) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

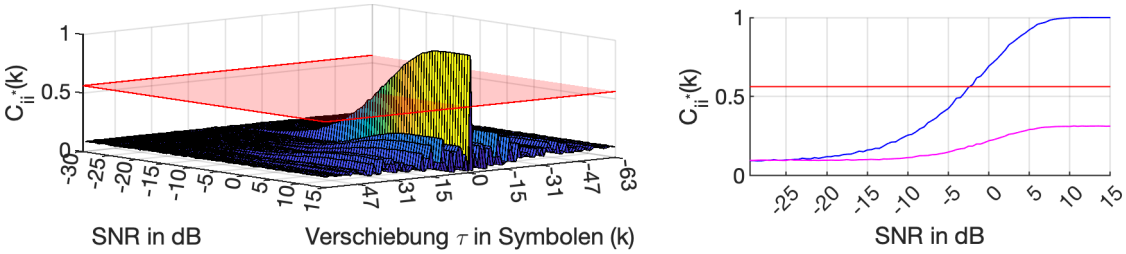
Abbildung 4.13: Minimum der periodische Autokorrelation für $k{=}0$ (rechts Blau) und dem Maximum für $k{\neq}0$ (rechts Magenta) über 1000 Testläufe



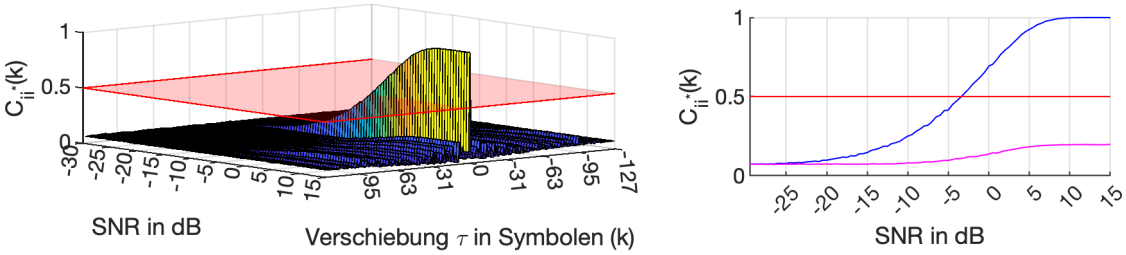
(a) Barker-Code (13 Bit)



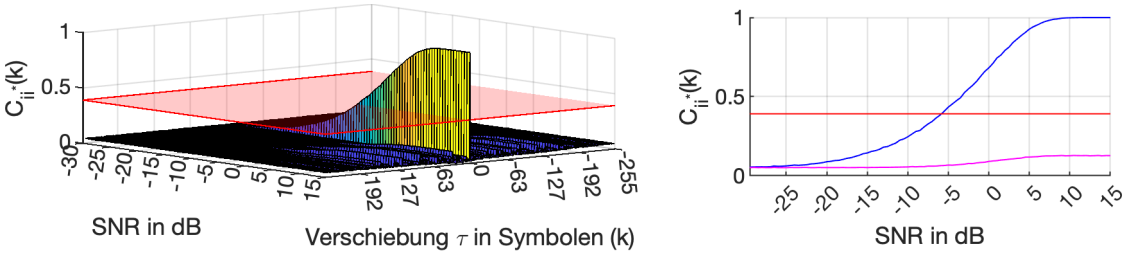
(b) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(c) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

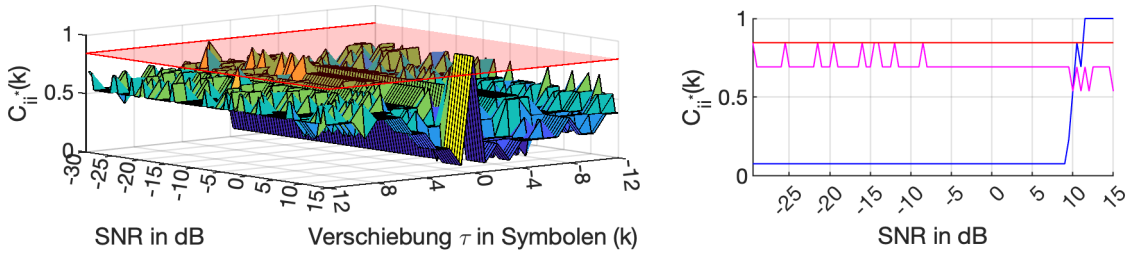


(d) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)

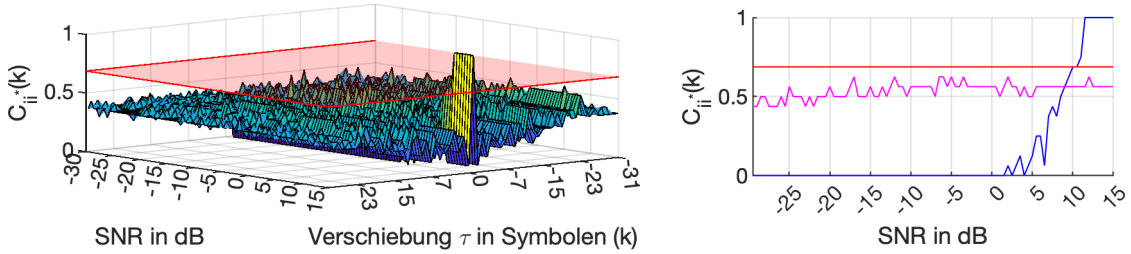


(e) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

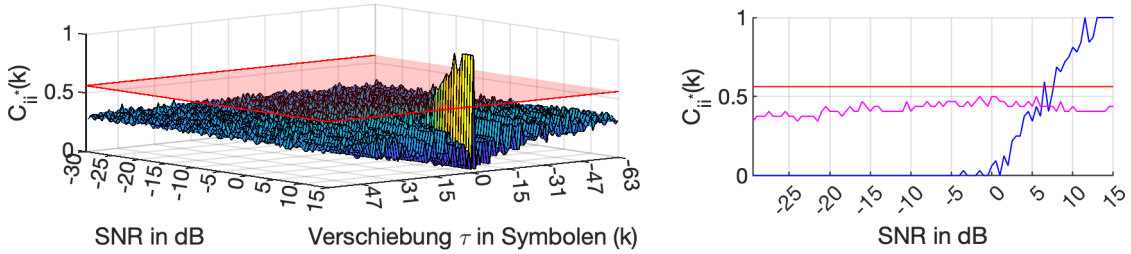
Abbildung 4.14: Durchschnittliche aperiodische Autokorrelation für $k{=}0$ (rechts Blau) und für $k{\neq}0$ (rechts Magenta) über 1000 Testläufe



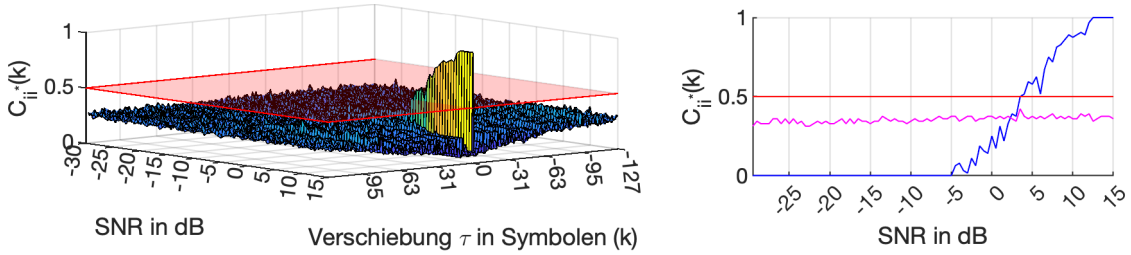
(a) Barker-Code (13 Bit)



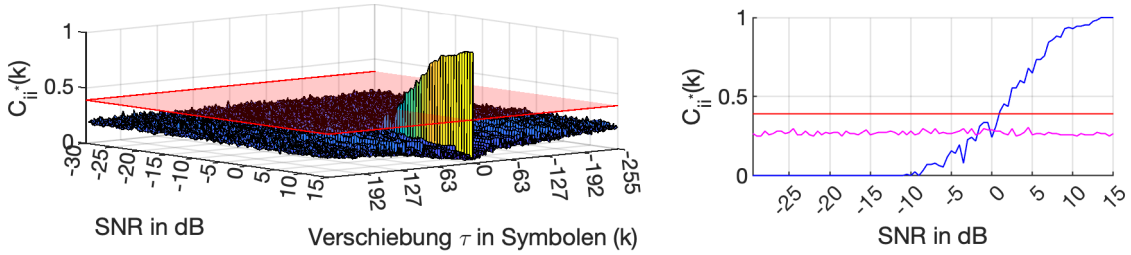
(b) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(c) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

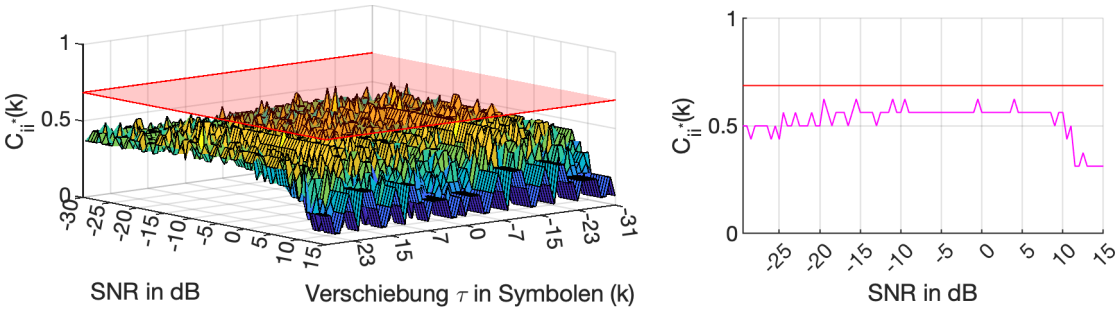


(d) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)

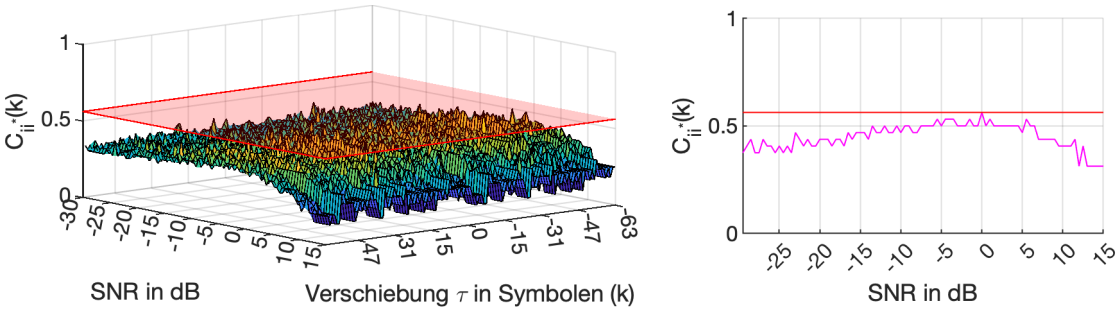


(e) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

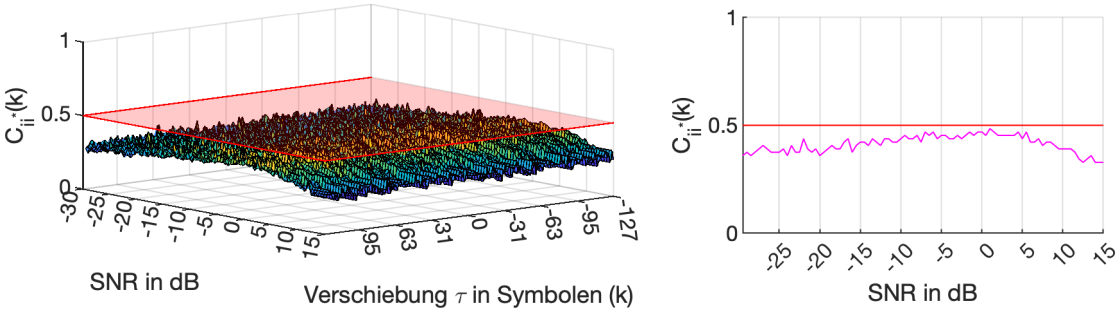
Abbildung 4.15: Maximum der aperiodische Autokorrelation für $k \neq 0$ (rechts Magenta) und dem Minimum für $k = 0$ (rechts Blau) über 1000 Testläufe



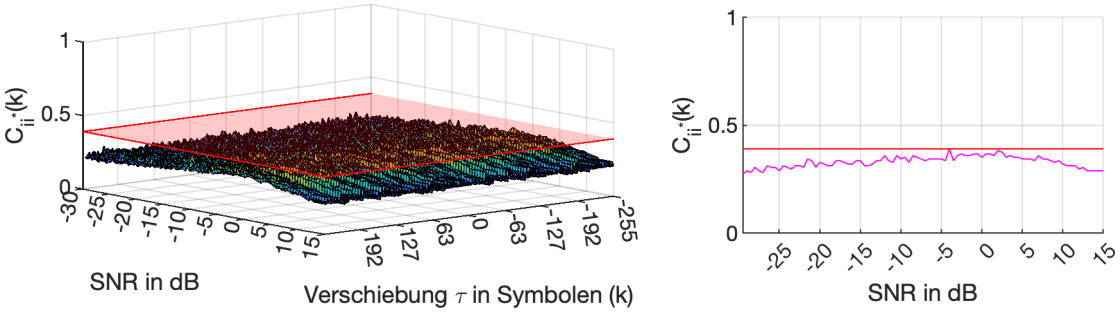
(a) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(b) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

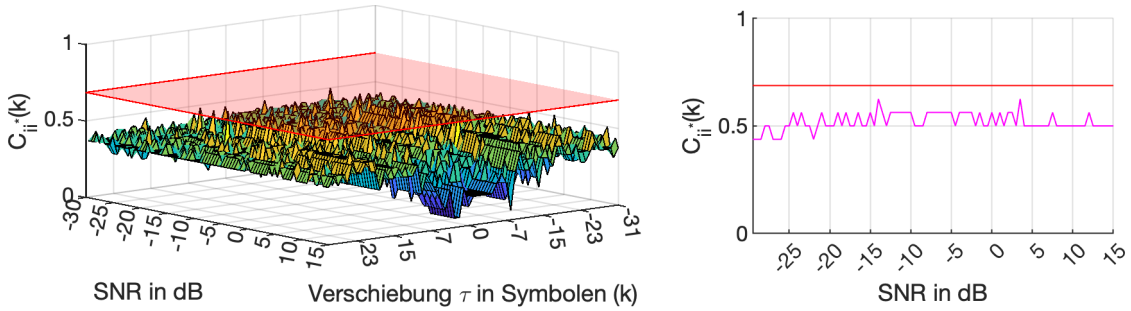


(c) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)

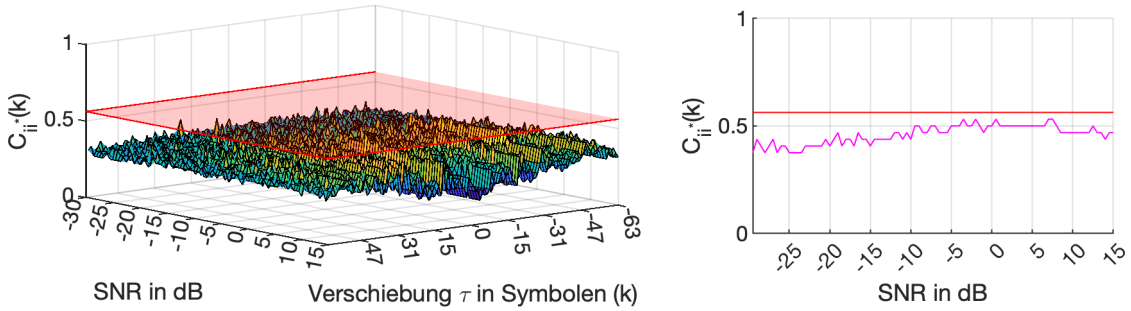


(d) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

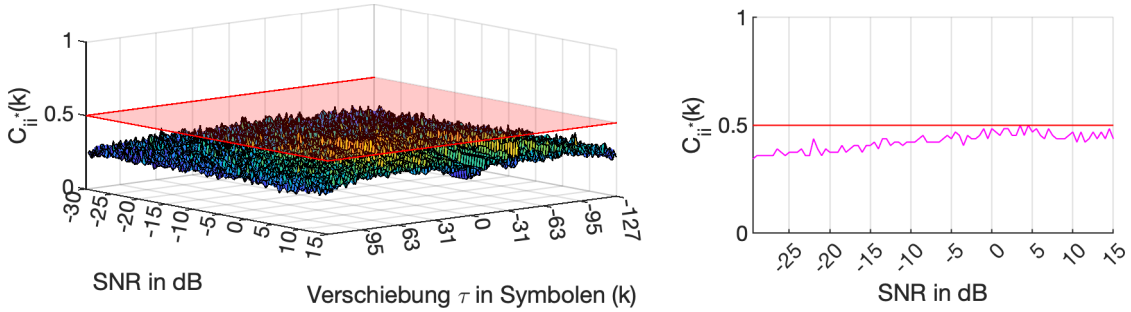
Abbildung 4.16: Maximum der periodische Kreuzkorrelation (rechts Magenta) über 1000 Testläufe



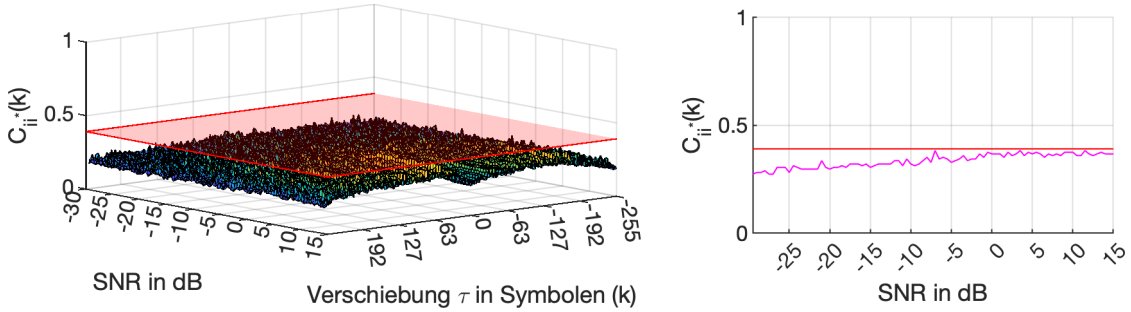
(a) orthogonale Kasami-basierte Folgen (OKBF) (32 Bit)



(b) orthogonale Kasami-basierte Folgen (OKBF) (64 Bit)

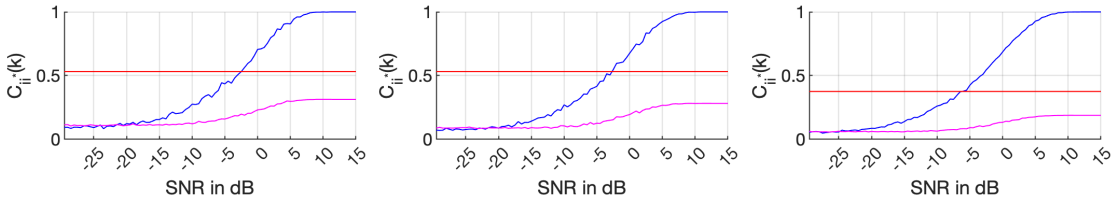


(c) orthogonale Kasami-basierte Folgen (OKBF) (128 Bit)



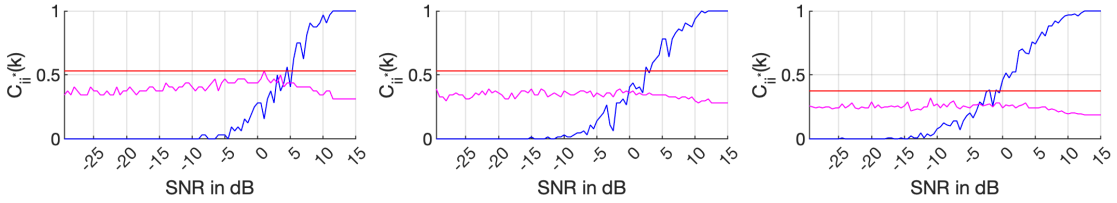
(d) orthogonale Kasami-basierte Folgen (OKBF) (256 Bit)

Abbildung 4.17: Maximum der aperiodische Kreuzkorrelation (rechts Magenta) über 1000 Testläufe



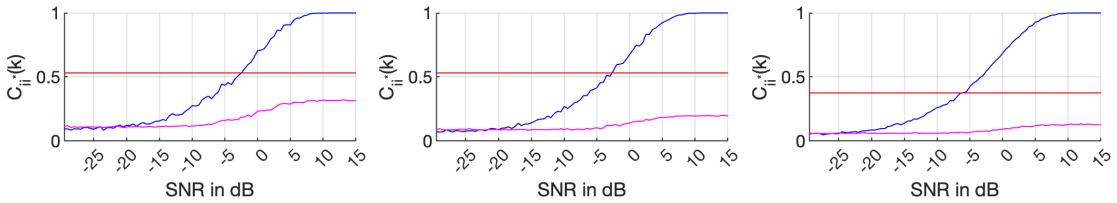
(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.18: Durchschnittliche periodische Autokorrelation im Unterwasserkanal (UWK) für $k{=}0$ (Blau) und $k{\neq}0$ (Magenta)



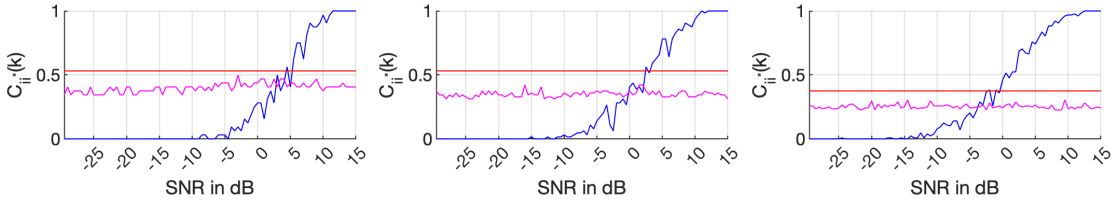
(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.19: Maximum für $k{\neq}0$ (Magenta) und Minimum für $k{=}0$ (Blau) der periodische Autokorrelation der OKBF



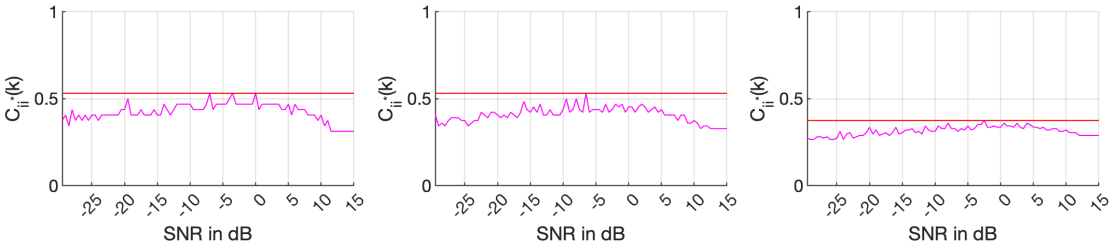
(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.20: Durchschnittliche aperiodische Autokorrelation im UWK



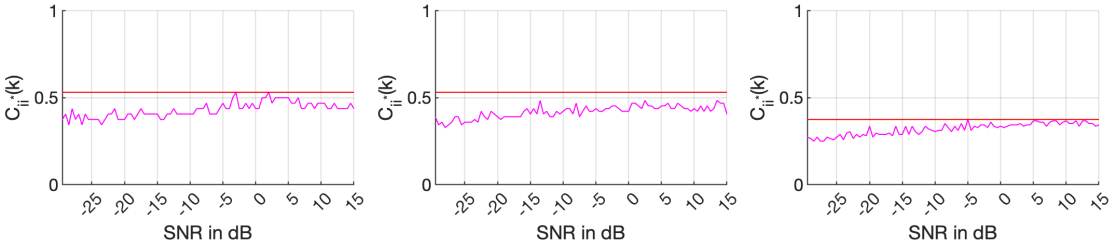
(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.21: Maximum für $k{=}0$ (Blau) und Minimum für $k{\neq}0$ (Magenta) der aperiodische Autokorrelation der OKBF im UWK



(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.22: Maximum der periodische Kreuzkorrelation im UWK



(a) OKBF (64 Bit) **(b)** OKBF (128 Bit) **(c)** OKBF (256 Bit)

Abbildung 4.23: Maximum der aperiodische Kreuzkorrelation im UWK

	32 Bit			64 Bit			128 Bit			256 Bit		
Güte	$\mathbf{R_{dth}}$	**SNR**$^-$	#	$\mathbf{R_{dth}}$	**SNR**$^-$	#	$\mathbf{R_{dth}}$	**SNR**$^-$	#	$\mathbf{R_{dth}}$	**SNR**$^-$	#
8.00	–	–	0	–	–	0	–	–	0	0.313	−2.5	1
7.75	–	–	0	0.500	6.0	1	–	–	0	0.297	−2.5	1
7.50	–	–	0	0.500	4.5	1	–	–	0	0.313	−0.5	1
7.25	–	–	0	0.531	4.5	1	–	–	0	0.297	−2.0	1
7.00	–	–	0	0.500	8.5	1	–	–	0	0.289	−2.5	1
6.75	–	–	0	0.469	4.0	1	–	–	0	0.297	−1.5	1
6.50	–	–	0	0.500	5.0	1	0.422	1.5	1	0.336	−1.5	2
6.25	–	–	0	0.469	5.5	1	0.406	2.0	1	0.344	−0.5	1
6.00	0.563	8.5	1	0.500	4.0	1	0.391	1.5	1	0.367	−1.0	2
5.75	0.625	8.0	1	0.469	5.5	1	0.391	1.0	1	0.273	−3.0	3
5.50	0.563	8.5	1	0.469	4.5	1	0.406	2.5	1	0.297	−2.5	6
5.25	0.750	9.5	1	0.594	5.0	1	0.391	1.5	2	0.297	−3.0	7
5.00	0.625	7.5	1	0.500	4.0	1	0.406	2.0	2	0.320	−1.5	10
4.75	0.625	7.5	1	0.531	5.0	1	0.406	2.5	3	0.320	−2.0	11
4.50	0.688	8.0	1	0.594	6.5	1	0.422	1.5	3	0.328	−2.0	32
4.25	0.688	10.0	1	0.469	4.5	1	0.406	1.0	7	0.320	−2.0	122
4.00	0.625	11.0	1	0.469	6.0	2	0.406	2.0	9	0.336	−1.5	305
3.75	0.688	9.5	1	0.531	4.0	3	0.469	2.5	12	0.359	−1.0	974
3.50	0.625	7.5	2	0.625	7.5	5	0.422	4.0	26	0.391	0.0	1888
3.25	0.625	8.5	2	0.531	5.0	4	0.484	2.5	51	0.383	−0.5	2000
3.00	0.625	8.0	3	0.531	5.5	9	0.531	3.5	270	0.375	0.0	2000
2.75	0.625	8.0	2	0.531	6.0	14	0.484	3.0	1136	0.539	2.0	2000
2.50	0.750	11.0	8	0.625	7.0	39	0.547	4.0	2000	0.453	0.5	2000
2.25	0.875	11.5	14	0.594	6.5	257	0.547	3.5	2000	0.469	1.0	2000
2.00	0.750	10.5	50	0.656	6.5	2000	0.672	6.0	2000	0.555	3.0	2000

Tabelle 4.2: Detektionsschwellwerte der OKBF mit minimalen SNR$^-$ (UWN)

Güte	32 Bit R_{dth}	32 Bit SNR^-	32 Bit #	64 Bit R_{dth}	64 Bit SNR^-	64 Bit #	128 Bit R_{dth}	128 Bit SNR^-	128 Bit #	256 Bit R_{dth}	256 Bit SNR^-	256 Bit #
8.00	–	–	0	–	–	0	–	–	0	0.313	−2.5	1
7.75	–	–	0	0.500	–	1	–	–	0	0.297	−2.5	1
7.50	–	–	0	0.500	–	1	–	–	0	0.313	−2.5	1
7.25	–	–	0	0.531	–	1	–	–	0	0.297	−2.5	1
7.00	–	–	0	0.500	–	1	–	–	0	0.289	−2.5	1
6.75	–	–	0	0.469	–	1	–	1.0	0	0.297	−3.0	1
6.50	–	–	0	0.500	–	1	0.422	0.5	1	0.336	−1.5	2
6.25	–	–	0	0.469	–	1	0.406	0.5	1	0.344	−2.5	1
6.00	0.563	8.5	1	0.500	–	1	0.391	0.5	1	0.367	−2.0	2
5.75	0.625	8.5	1	0.469	5.5	1	0.391	1.0	1	0.273	−2.5	3
5.50	0.563	9.5	1	0.469	4.5	1	0.406	1.0	1	0.297	−2.5	6
5.25	0.750	8.5	1	0.594	4.0	1	0.391	1.0	2	0.297	−2.5	7
5.00	0.625	8.5	1	0.500	5.0	1	0.406	1.0	2	0.320	−2.0	10
4.75	0.625	8.5	1	0.531	3.0	1	0.406	0.0	3	0.320	−1.5	11
4.50	0.688	8.0	1	0.594	4.5	1	0.422	1.5	3	0.328	−2.0	32
4.25	0.688	11.0	1	0.469	4.5	1	0.406	1.0	7	0.320	−1.5	122
4.00	0.625	7.5	1	0.469	4.5	2	0.406	1.5	9	0.336	−1.0	305
3.75	0.688	9.5	1	0.531	5.0	3	0.469	2.0	12	0.359	−1.0	974
3.50	0.625	10.0	2	0.625	4.0	5	0.422	2.5	26	0.391	−1.0	1888
3.25	0.625	10.0	2	0.531	5.0	4	0.484	1.5	51	0.383	−1.0	2000
3.00	0.625	11.0	3	0.531	5.5	9	0.531	2.0	270	0.375	0.0	2000
2.75	0.625	11.0	2	0.531	5.5	14	0.484	2.5	1136	0.539	−1.0	2000
2.50	0.750	8.5	8	0.625	6.0	39	0.547	2.0	2000	0.453	1.0	2000
2.25	0.875	13.5	14	0.594	5.5	257	0.547	3.0	2000	0.469	2.0	2000
2.00	0.750	11.0	50	0.656	6.5	2000	0.672	3.5	2000	0.555	4.0	2000

Tabelle 4.3: Detektionsschwellwerte der OKBFU mit minimalen SNR^-

Güte	32 Bit R_{dth}	32 Bit SNR^-	32 Bit #	64 Bit R_{dth}	64 Bit SNR^-	64 Bit #	128 Bit R_{dth}	128 Bit SNR^-	128 Bit #
8.00	–	–	0	–	–	0	–	–	0
7.75	–	–	0	0.500	–	1	–	–	0
7.50	–	–	0	0.500	–	1	–	–	0
7.25	–	–	0	0.531	–	1	–	–	0
7.00	–	–	0	0.500	–	1	–	–	0
6.75	–	–	0	0.469	–	1	–	1.0	0
6.50	–	–	0	0.500	–	1	0.422	0.5	1
6.25	–	–	0	0.469	–	1	0.406	0.5	1
6.00	0.563	8.5	1	0.500	–	1	0.391	0.5	1
5.75	0.625	8.5	1	0.469	5.5	1	0.391	1.0	1
5.50	0.563	9.5	1	0.469	4.5	1	0.406	1.0	1
5.25	0.750	8.5	1	0.594	4.0	1	0.391	1.0	2
5.00	0.625	8.5	1	0.500	5.0	1	0.406	1.0	2
4.75	0.625	8.5	1	0.531	3.0	1	0.406	0.0	3
4.50	0.688	8.0	1	0.594	4.5	1	0.422	1.5	3
4.25	0.688	11.0	1	0.469	4.5	1	0.406	1.0	7
4.00	0.625	7.5	1	0.469	4.5	2	0.406	1.5	9
3.75	0.688	9.5	1	0.531	5.0	3	0.469	2.0	12
3.50	0.625	10.0	2	0.625	4.0	5	0.422	2.5	26
3.25	0.625	10.0	2	0.531	5.0	4	0.484	1.5	51
3.00	0.625	11.0	3	0.531	5.5	9	0.531	2.0	270
2.75	0.625	11.0	2	0.531	5.5	14	0.484	2.5	1136
2.50	0.750	8.5	8	0.625	6.0	39	0.547	2.0	2000
2.25	0.875	13.5	14	0.594	5.5	257	0.547	3.0	2000
2.00	0.750	11.0	50	0.656	6.5	2000	0.672	3.5	2000

Tabelle 4.4: Detektionsschwellwerte der GF mit minimalen SNR^-

4.3.2 Störanfälligkeit

Wegen des Dopplereffekts und der Mehrwegeausbreitung im Unterwasserkanal, muss das Kasami-basierte Verfahren auf die umgebungsspezifische Wirksamkeit und den Einfluss der Störfaktoren hin untersucht werden.

Wie bei jeder anderen Präambel muss dies spezifisch für das Kommunikationssystem und dessen Anwendungsgebiet und Trägerplattform bestimmt und geprüft werden. Zudem hängt die Störanfälligkeit, beschrieben in Axiom 4.2, davon ab, ob Verfahren wie die Doppler-Vorabschätzung und -Kompensation integriert sind, da durch die Vorabschätzung und meist nachfolgende Kompensation der Einfluss des Dopplereffekts auf das empfangene Signal reduziert werden kann. Alternativ kann die Übertragungsfunktion des Kanals geschätzt und zur Kompensation des Dopplers, durch die Anpassung der Filterkoeffizienten kompensiert werden, was unter dem Begriff adaptive Equalization geläufig ist und nur bedingt in ruhigen kaum veränderlichen Unterwasserumgebungen Erfolg versprechend ist. Auch die Nutzung von Doppler-robusten Modulaktionsschemata wie Orthogonales Frequenzmultiplexverfahren (OFDM) oder Frequenz-Umtastung (FSK) können die Auswirkungen des Dopplereffekts signifikant reduzieren [191–196].

Axiom 4.2 — Störanfälligkeit:
Für alle OKBFs $i, j \in KF$ in der Menge der OKBFs $KF := \{ID, DT\}$, eine Frequenzverschiebung f_{off} (in Hz) und eine Abtastrate f_s (in Hz) gilt:

Die Kreuzkorrelation zwischen jeder Folge i und allen anderen j^* bei beliebiger Verschiebung $k > 0$ sowie die Autokorrelation von i und i^* bei $k > 0$ müssen kleiner oder gleich dem Detektionsschwellwert R_{dth} sein. Die Autokorrelation bei $k = 0$ muss diesen übersteigen.

Für $x \in \{i, j\}$ ist $x^*[k]$ definiert durch:

$$x^*[k] = \begin{cases} x[k] & \text{für } k = 0 \\ x[k] \cdot \exp\left(\mathrm{i} \cdot 2\pi \sum_{n=0}^{k-1} \frac{f_{off}}{f_s} \right) & \text{für } k > 0 \end{cases} \tag{4.20}$$

Es muss gelten:

$$\max_{\substack{i \neq j^* \\ k \neq 0}} \left(C_{i,j^*}[k] \right) \leq R_{dth} < C_{i,i^*}[0] \tag{4.21}$$

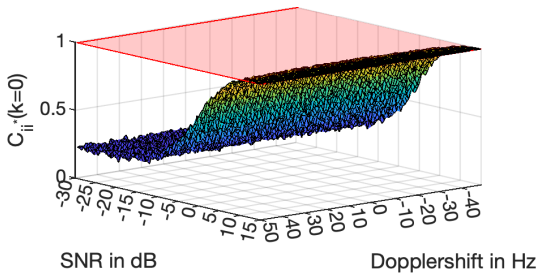
Die Störanfälligkeit wird in Bezug auf den Dopplereffekt anhand eines exemplarischen Modems aufgezeigt und diskutiert. Die Abbildungen 4.24 bis 4.27 zeigen dazu die normierte Autokorrelation für $k = 0$ in einer dreidimensionalen Darstel-

lung in a), die Autokorrelation mit Draufsicht für $k{=}0$ in b), die Autokorrelation mit für $k{\neq}0$ in c) und abschließend den Wert der Korrelationen, welcher über dem berechneten Detektionsschwellwert R_{dth} für die minimale Güte von 3.0 liegt in d). Alle Abbildungen zeigen die Ergebnisse in Relation zwischen dem SNR und der relativen Geschwindigkeit in einem idealen Kanal ohne Mehrwegeausbreitung ab. Für das gezeigte Beispiel wird die Binary Phase Shift Keying (BPSK)-Modulation mit einer Trägerfrequenz f_c von $10\,\mathrm{kHz}$, die Nutzung einer Impulsformung mit einem Interpolationsfaktor I_{BB} von 4 gegen Übersprechen, eine Abtastrate f_s von $160\,\mathrm{kHz}$ im Passband, eine Bandbreite B von $5\,\mathrm{kHz}$ und einer Symbolrate f_s von $5\,\mathrm{kHz}$ definiert. Jede getestete Länge wurde 1000 mal für den SNR-Bereich von $-30\,\mathrm{dB}$ bis $15\,\mathrm{dB}$ in 0.5-dB und $1\,\mathrm{Hz}$-Schritten für eine Frequenzverschiebung von $-50\,\mathrm{Hz}$ bis $50\,\mathrm{Hz}$ im Basisband ausgewertet. Die Verschiebung in diesem Beispiel deckt einem Bereich der relativen Geschwindigkeit v_{TR} von $-15\,\mathrm{m\,s^{-1}}$ bis $15\,\mathrm{m\,s^{-1}}$ bei einer angenommenen durchschnittlichen Schallgeschwindigkeit c_w von $1500\,\mathrm{m\,s^{-1}}$ ab. Eine typische Geschwindigkeit von autonomen Unterwasserfahrzeugen liegt derzeit bei $1\,\mathrm{m\,s^{-1}}$ bis $2\,\mathrm{m\,s^{-1}}$, was bei einem maximalen Dopplershift für das exemplarische Modem von $\pm 13.4\,\mathrm{Hz}$ entspricht, wenn sich die Kommunikationssysteme in entgegengesetzte Richtungen bewegen.

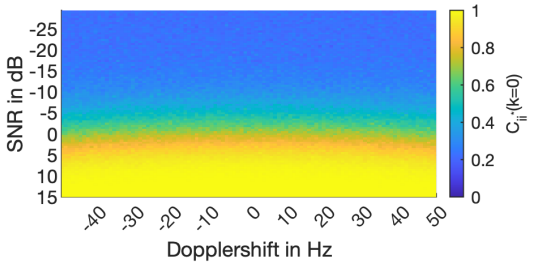
Die Abbildungen 4.24 bis 4.27 zeigen jeweils in a) und b), dass die Präambeln ohne zusätzliche Mechanismen zur Kompensation des Dopplereffektes zwar erkannt werden kann, aber die Länge der Präambel einen wesentlichen Einfluss auf den abdeckbaren Frequenz-Bereich hat. Mit einer größer werdenden Länge wird die Korrelationseigenschaft der Folgen bei immer geringeren Dopplerverschiebungen maßgeblich gestört. Dies liegt daran, dass mit einer längeren Folge der Einfluss der Dopplerverschiebung über die Zeit kumulativ ist, bis es zu einer Änderung der rekonstruierten Folge und der Korrelationseigenschaft kommt. Dabei reicht, ohne Kompensation, eine Verschiebung in Abhängigkeit des Interpolationsfaktors I_{BB} um $1/I_{BB}$ eines Symbols durch den Dopplereffekt mittels Stauchung oder Zerrung, um die Korrelationseigenschaften störend zu beeinflussen. Der maximale Dopplershift Δf^* bei dem die Folgen noch detektiert werden können, kann über die Gl. (4.22) mit dem Interpolationsfaktor I_{BB}, der Bandbreite B und der Länge N der Präambel in Symbolen bestimmt werden.

$$\pm\Delta f^* = (1 - N \cdot I_{BB} / (N \cdot I_{BB} \pm 1)) \cdot B \tag{4.22}$$

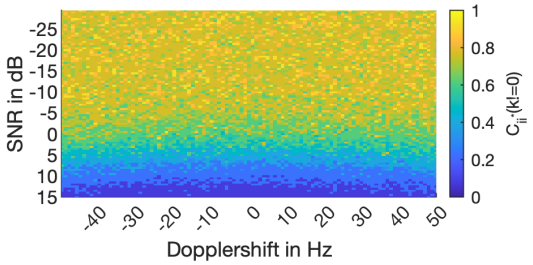
Für das gegebene Beispiel sind die Kommunikationssysteme bei einer Länge von $13\,\mathrm{Bit}$ des Barker-Codes nicht einsetzbar. Barker-Codes sind zwar aufgrund ihrer Kürze unempfindlich gegenüber dem Dopplershift, können aber bei einem verrauschten Medium ohne Synchronisation der Rahmen, wie die Abbildungen 4.24c and 4.24d zeigen, nicht eindeutig detektiert werden. Kommunikationssysteme können unter Verwendung von OKBF mit $64\,\mathrm{Bit}$ hingegen für kleine Netze genutzt



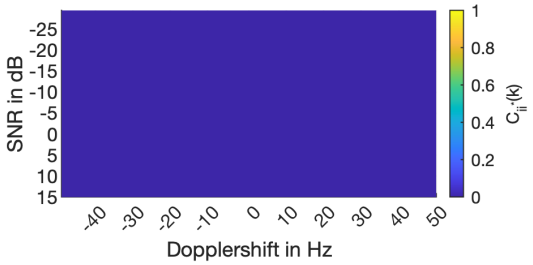
(a) $C_{i,i^*}(k{=}0)$ (3D)



(b) $C_{i,i^*}(k{=}0)$ (Top)

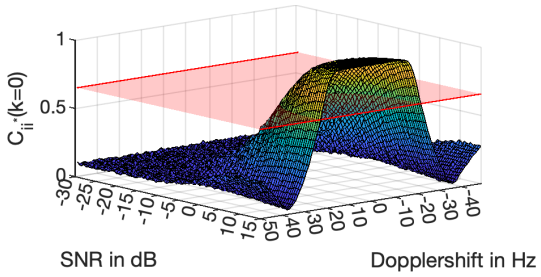


(c) $C_{i,i^*}(k{\neq}0)$

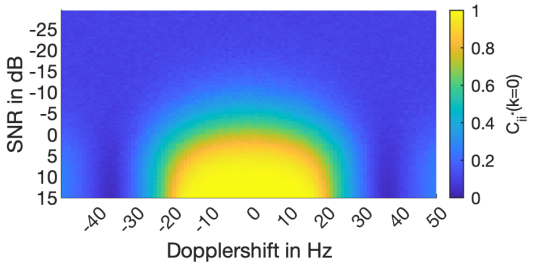


(d) $max(C_{i,i^*}(k){-}R_{dth}(0.4,13),0)$

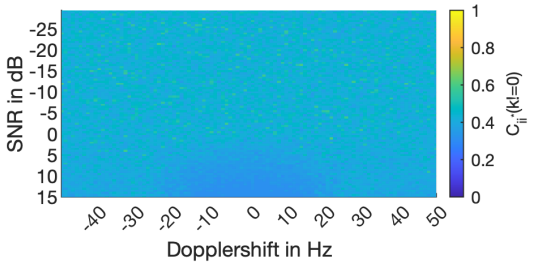
Abbildung 4.24: Auswirkungen des Dopplers für den Barker-Code



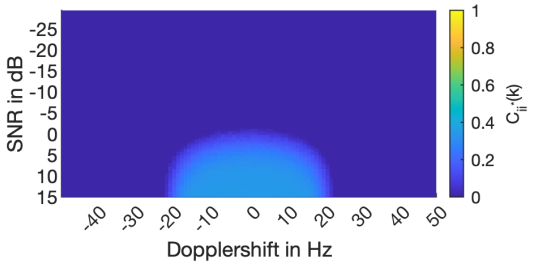
(a) $C_{i,i^*}(k{=}0)$ (3D)



(b) $C_{i,i^*}(k{=}0)$ (Top)

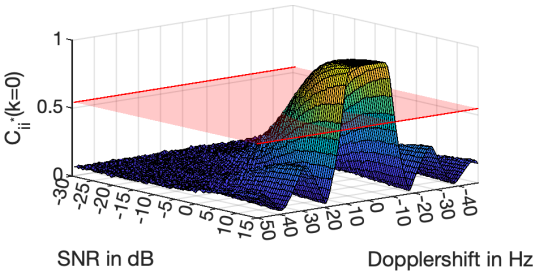


(c) $C_{i,i^*}(k{\neq}0)$

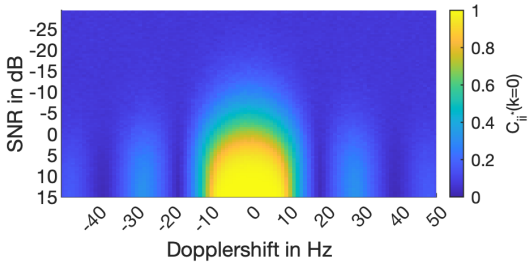


(d) $max(C_{i,i^*}(k){-}R_{dth}(0.4,64),0)$

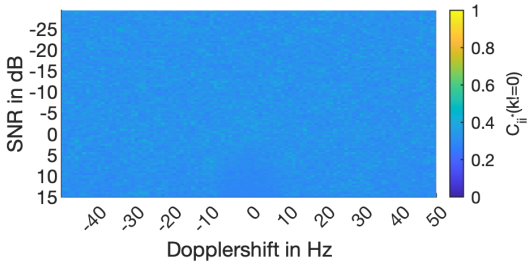
Abbildung 4.25: Auswirkungen des Dopplers für OKBF der Länge $64\,\mathrm{Bit}$



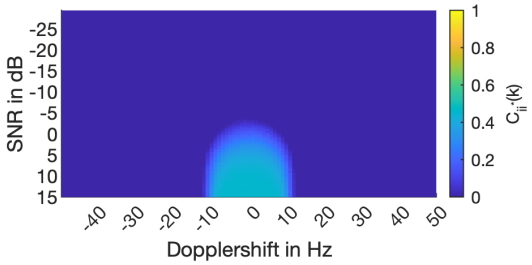
(a) $C_{i,i^*}(k{=}0)$ (3D)



(b) $C_{i,i^*}(k{=}0)$ (Top)

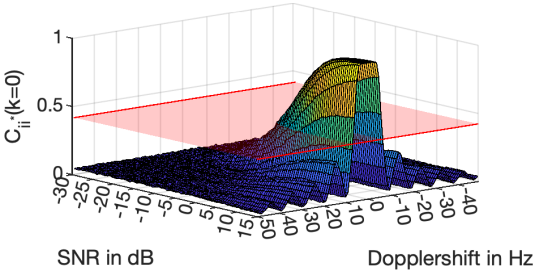


(c) $C_{i,i^*}(k{\neq}0)$

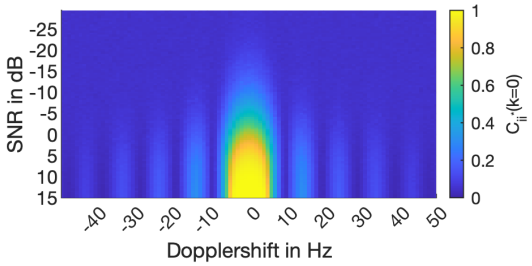


(d) $max(C_{i,i^*}(k){-}R_{dth}(0.4,128),0)$

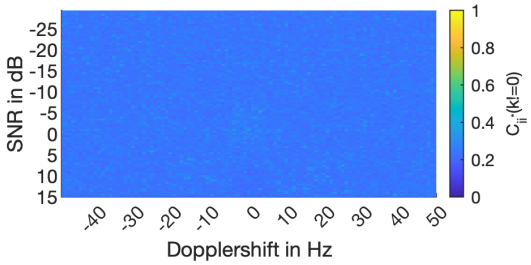
Abbildung 4.26: Auswirkungen des Dopplers für OKBF der Länge 128 Bit



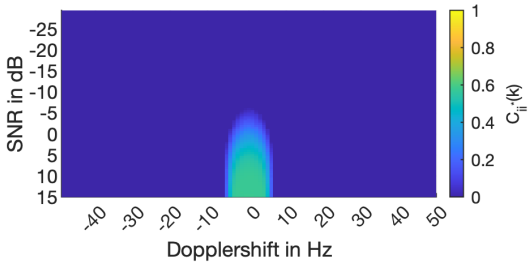
(a) $C_{i,i^*}(k{=}0)$ (3D)



(b) $C_{i,i^*}(k{=}0)$ (Top)



(c) $C_{i,i^*}(k{\neq}0)$



(d) $max(C_{i,i^*}(k){-}R_{dth}(0.4,256),0)$

Abbildung 4.27: Auswirkungen des Dopplers für OKBF der Länge 256 Bit

und auch bei maximaler Geschwindigkeit der Unterwasserfahrzeuge eindeutig detektiert und zugeordnet werden. Die Nutzung von OKBF der Länge $128\,\mathrm{Bit}$ (Abbildung 4.26) für mittlere bis große Netze ist in diesem Beispiel beschränkt auf eine maximale Geschwindigkeit von $1.47\,\mathrm{m\,s^{-1}}$ und OKBF der Länge $256\,\mathrm{Bit}$ (Abbildung 4.27) auf $0.74\,\mathrm{m\,s^{-1}}$ für eine zuverlässige und eindeutige Detektion und Rahmentypenkennung.

Zusammenfassung

In diesem Kapitel wurde die Herausforderung der Präambel-basierten Detektion von Datenrahmen in der akustischen Unterwasserkommunikation eingehend behandelt. Die Analyse fokussierte sich auf die Robustheit möglicher Sequenzen gegenüber allgemeinem und speziell unterwasserspezifischem Rauschen sowie auf die Auswirkungen des Dopplereffekts. Um die Kommunikationsqualität zu verbessern, wurde ein Kasami-basierten Ansatz eingeführt, der eine skalierbare Menge von Sequenzen bietet. Diese Sequenzen sind durch die Anpassung der geforderten Güte in der Lage, Kommunikationsnetze unterschiedlicher Größenordnung effizient abzudecken, von kleinen hin zu großen Netzwerken.

Ein weiterer zentraler Aspekt dieses Kapitels war die systematische Bestimmung der erforderlichen Detektionsschwellwerte durch empirische Tests, um eine sichere Erkennung zu gewährleisten. Das Kasami-basierte Verfahren ermöglicht es, die Detektion und Zuordnung von Datenrahmen in den Coderaum der Präambeln selbst zu verlagern. Dies macht die Decodierung von CTS- und ACK-Rahmen im Kontext einer Kommunikation mittels RTS/CTS-Austausch überflüssig. Dieses Verfahren bietet zudem die Vorteile eines erhöhten Datendurchsatzes in Netzwerken, einer verringerten Rate falscher Detektionen und einer reduzierten Latenz bei vergleichbarer Präambellänge. Zudem ermöglicht das Verfahren die Implementierung systemspezifischer Aufwachphasen und eine effektive Kollisionserkennung von empfangenen Datenrahmen.

Die Einführung der Kasami-basierten Präambel bietet damit einen bedeutenden Fortschritt in der Unterwasserkommunikation. Die Methode bietet eine effiziente Lösung zur Verbesserung der Kommunikationszuverlässigkeit und -geschwindigkeit in unterwasserbasierten Netzwerken. Der Ansatz kann maßgeblich zur Optimierung bestehender Kommunikationssysteme beitragen und eröffnet neue Möglichkeiten für die Entwicklung zukünftiger Technologien in der akustischen Unterwasserkommunikation für mehrnutzerbasierte Netzwerke.

5

Varianten-basierter modularer Entwurf

Die Entwicklung eines Modems für schwierige Kanalbedingungen, wie sie in dem akustischen Unterwasserkanal herrschen, ist nach wie vor ein herausforderndes Forschungsfeld. Der Kanal ist von einem sich schnell ändernden Mehrwegausbreitungsverhalten betroffen, das durch Signalstreuung und Reflexionen an der Wasseroberfläche und am Boden hervorgerufen wird. Hinzu kommen Störungen durch Flora und Fauna. Zusätzlich verursachen, insbesondere im Meerwasser, Schichten mit unterschiedlichem Salzgehalt und Wassertemperaturen neben Phasenverschiebungen, Dopplereffekt und Interferenzen auch Brechungen und Reflexionen der akustischen Signale. Die Entwicklung von Modems lässt sich gemäß Tabellen 2.1 and 2.2 generell in zwei Bereiche einteilen:

- Rapid-Prototyping und die Hardware/Software-orientierte Entwicklung für spezielle Bedingungen und Anforderungen, welche auf die genutzte Hardware manuell zugeschnitten und mithilfe von Expertenwissen und Erfahrung aus Tests optimiert wird.
- Task-orientierte Entwicklung, welche die Hardware/Software und die Eigenschaften des Systems auf die Missionsaufgabe(n) mithilfe von Expertenwissen und Erfahrung aus langwierigen und kostspieligen Tests optimiert.

Beide Möglichkeiten müssen zur Optimierung auf See mit der entwickelten Hard- und Software ausprobiert werden. Die Auswertung und Überprüfung des Kommunikationssystems benötigt zusätzlich neben der eingesetzten Hardware einen Träger (Drohne, Boje) und entsprechende Testumgebungen oder eine See-Expedition, welche aber mit ungewissem Erfolg und hohen Kosten verbunden ist. Zusätzlich ist in beiden Fällen eine Anpassung an die sich stetig ändernde Umgebung oder eine Neuorientierung der Anforderungen nur unter erheblichen Aufwand und Kosten möglich. Der Entwurf und die Entwicklung eines Modems für ein benutzerdefiniertes Kommunikationsszenario

erfordern jedoch einen flexiblen Ansatz. Vielversprechender ist es daher, modellbasiert vorzugehen und so Spezifikation, Validierung und Implementierung in einer rechnergestützten Entwurfs- und Simulationsumgebung durchzuführen. In diesem Kapitel wird das Konzept und die Umsetzung für ein solches modellbasiertes und synthetisierbares Kommunikationssystem eingeführt. Mit diesen können wahlweise Simulationen oder Emulationen mit definierte Tests ortsungebunden unter variierenden Bedingungen durchgeführt werden.

5.1 Entwurfsmodell

Für den modellbasierten Entwurf wird das Kommunikationssystem in die Module Sender, Empfänger und Kanal zerlegt (Abbildung 5.1). Die Module Sender und Empfänger werden dabei in weitere Komponenten gegliedert (Top-Down-Entwurf). Für jede dieser Komponenten existieren unterschiedliche Varianten der Implementierung.

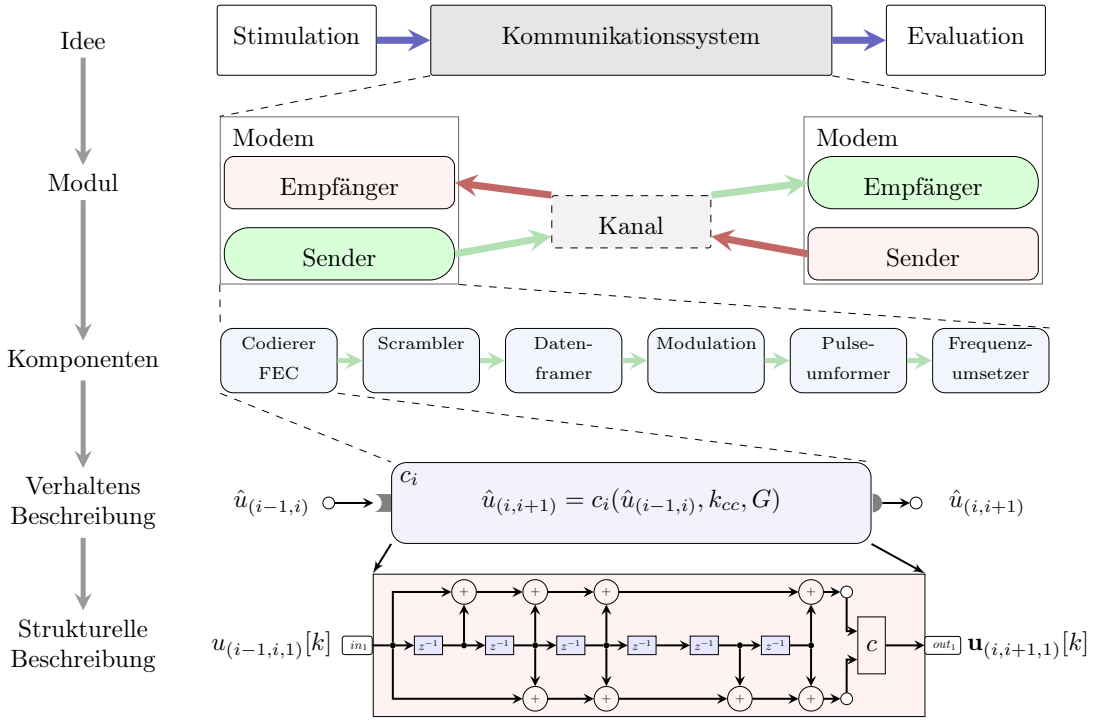


Abbildung 5.1: Entwicklungsebenen des Kommunikationsmodells

Definition 5.1 — Parameter:
Der Parameter $p \in P$, mit P als Parametersatz, repräsentiert eine weiche oder harte Eigenschaft oder Konfiguration des Systems. Dabei können die Parameter p in Ihrem Gültigkeits- und Auflösungsbereich beschränkt sein.

- Dynamische Parameter können zur Laufzeit (Online), gemäß vorgegebenen Einschränkungen, angepasst werden.
- Statische Parameter müssen vor der Simulation/Synthese (Offline) festgelegt werden und sind zur Nutzungszeit nicht änderbar.

Auf der Abstraktionsebene des Gesamtsystems wird das Kommunikationssystem, in dieser Arbeit, auf den digitalen Bereich mit zwei Modulen $G_{M_i} \in M := \{G_{M_{Tx}} G_{M_{Rx}}\}$, dem Sendemodul $G_{M_{Tx}}$ und dem Empfängermodul $G_{M_{Rx}}$, beschränkt. Die Kommunikation zwischen zwei Modems (M_1, M_2) erfolgt in der Regel im Duplex-Verfahren. Moderne akustische Unterwasserkommunikationssysteme können aber nur senden oder empfangen und reduzieren damit die Kommunikation auf Halb-Duplex. Damit kann die Kommunikation zwischen Sender und Empfänger mit D als zu sendenden Datenvektor, S als das resultierende akustische Signal im Passband, P dem Parameterset (Definition 5.1) und f_{s_i} als Sende- und f_{e_i} als Empfangsfunktion über zwei Kommunikationspfade beschrieben werden:

$$D = D' = f_{e_2}(S_1, P_2) = f_{e_1}(S_2, P_1) \text{ mit } S_2 = f_{s_2}(D, P_2) \text{ und } S_1 = f_{s_1}(D, P_1) \tag{5.1}$$

Handelt es sich um eine Kommunikation zwischen homogenen Systemen ($P_1 = P_2$), mit identischer Parametrisierung und Struktur, muss bei identischen Datenvektor D zusätzlich $S_1 = S_2$ gelten.

Das Modem mit einem Sender- und Empfängermodul kann als Komponenten-Graf $G_M(C, \hat{U})$ (Definition 5.2) spezifiziert werden [197], wobei die Knoten $c \in C$ die Komponenten (Definition 5.3) des Modems darstellen und damit das Grundgerüst bilden. Der Datenfluss des jeweiligen Moduls wird durch die Kanten $\hat{u} \subseteq C\,x\,C$ des Graphen G_M spezifiziert und verknüpft die ein- und ausgehenden Interaktionspunkte der Komponenten über Trägerbündel miteinander. Ein schematischer Komponenten-Graf des Sender- und Empfängermoduls ist in Abbildung 5.3 abgebildet. Die zusätzlichen Kanten $d_c \in D_c$ beschreiben die Abhängigkeiten der Komponenten zwischen dem Sender- und Empfänger-Modul (Abschnitt 5.1.1). Der Komponenten-Graf lässt sich basierend auf Gl. (5.1), mit f_{c_i} als Funktion der Komponenten c_i des Senders mit $i := \{1, \ldots, k\}$ und des Empfängers für $i := \{k+1, \ldots, l\}$ mit $p_j \in P_i$ als Parameterset der i-ten Komponente, wie folgt formal beschreiben:

$$D = D' = f_{c_{k+l}}(\ldots f_{c_k}(f_{c_{k-1}}(\ldots f_{c_2}(f_{c_1}(D, P_1), P_2), \ldots, P_{k-1}), P_k) \ldots, P_{k+l}) \tag{5.2}$$

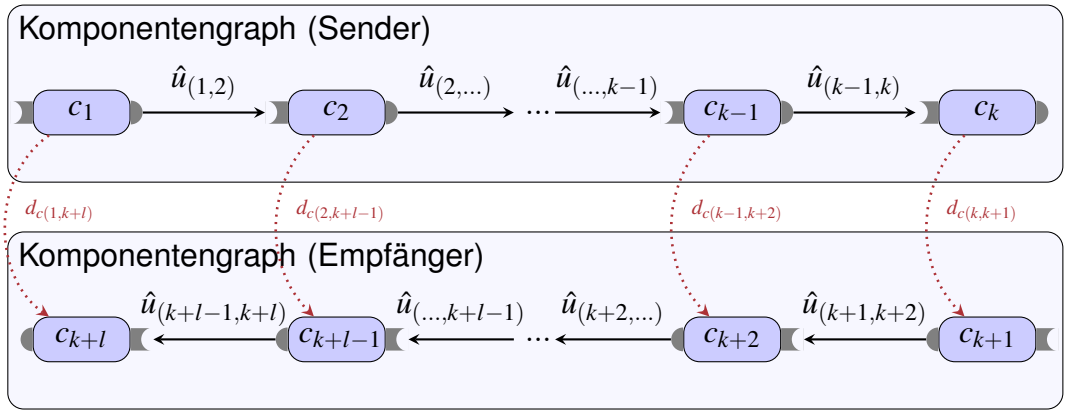


Abbildung 5.2: Schematisches Kommunikationsmodell als Graph mit Abhängigkeiten auf Komponentenebene (ohne Parameterzuweisung)

Um den Informations- oder Datenfluss zwischen den Komponenten zu beschreiben, nutzen Simulink und Hardwarebeschreibungssprachen den Begriff Signal. Der Begriff Signal beschreibt jedoch aus Sicht der Systemtheorie, wie auch bisher in dieser Arbeit genutzt, die Beschreibung einer Funktion, die Informationen über die Zeit abbildet. Für die klare Trennung wird daher im Folgenden der Begriff des Trägerbündels $\hat{u}i,j{\in}\hat{U}$ verwendet, der alle Träger ui,j,k umfasst, welche den k'ten Informationsfluss zwischen den Komponenten i und j beschreiben. Damit lässt sich der Begriff Träger direkt auf das Signal im Kontext von Simulink und Very High Speed Integrated Circuit Hardware Description Language (VHDL) abbilden.

Definition 5.2 — Komponenten-Graph:
Ein gültiger Komponenten-Graf $G_M(C,\hat{U})$ definiert die gerichtete Verknüpfung von Komponenten $c{\in}C$, wobei jede Komponente über genau ein eingehendes und ein ausgehendes Trägerbündel $\hat{u}{\in}\hat{U}$, als Repräsentation des Datenflusses, verfügt. Die modulübergreifenden Abhängigkeiten werden über $d_c{\in}D_c$ definiert.

Definition 5.3 — Komponente:
Eine Komponente $c_i{\in}C$ ist ein Objekt, dem eine Verhaltensbeschreibung oder Eigenschaft über das Parametersets P_i mit den Parametern $p{\in}P$ zugeordnet ist.

Die formale oder strukturelle Umsetzung der Beschreibung einer Komponente ist die Variante (Definition 5.4). Die Umsetzung der Komponente c_i durch die Varian-

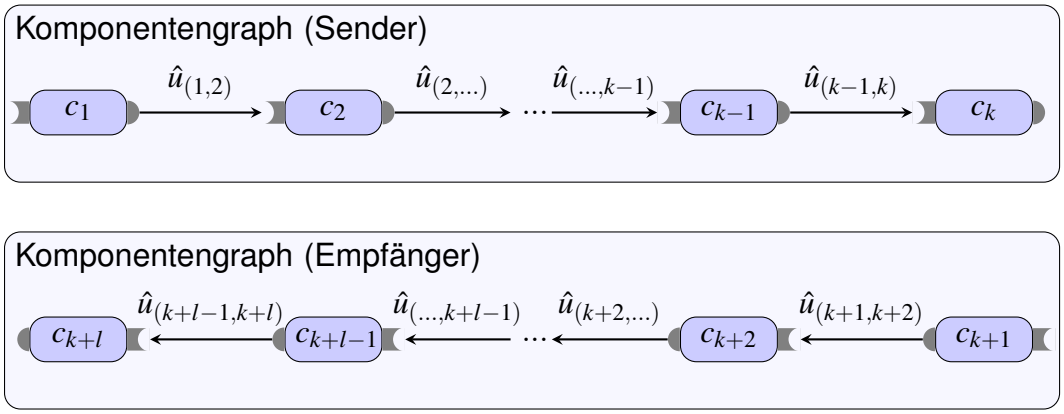


Abbildung 5.3: Schematisches Kommunikationsmodell als Graph mit Abhängigkeiten auf Komponentenebene (ohne Parameterzuweisung)

te v_i weist zusätzlich dem gerichteten Träger u einen festen Eingangsport $in_i \in I$ und Ausgangsport $out_i \in O$ zu. Eine beispielhafte Darstellung der Komponenten und zugehörigen Varianten ist schematisch in Abbildung 5.4 dargestellt.

Definition 5.4 — Variante:
Eine Variante $v_{c,i} \in V (i \in \mathbb{N})$ ist eine algorithmische, grafische, strukturelle oder funktionale Beschreibung des Verhaltens einer Komponente $c \in C$, welche nach einer Parametrisierung eine synthetisierbare und simulierbare Struktur oder Funktion abbildet. Die Zuordnung zwischen Variante und Komponente erfolgt über die Kanten $d_m \in D_m$ wobei die Kante $d_{m(i,j)}$ die Variante v_i an die Komponente c_j bindet.

Definition 5.5 — Varianten-Graf:
Der Varianten-Graf $G_V(V,D)$ beschreibt die Verknüpfung von Varianten $v \in V$, wobei jeder Komponente genau eine Variante zugeordnet ist und die Abhängigkeiten $d_v \in D_v$ zwischen den Varianten in Form der Kanten $D \subseteq V\,x\,V$ erfüllt werden.

Bei der Zerlegung des Sendemoduls müssen, sowohl die speziellen Anforderungen an die Unterwasserkommunikation, als auch die Beschränkung des Mediums berücksichtigt werden. Neben einer Präambel (d_{pre}) und der Gleichverteilung von Symbolen im unipolaren Symbolraum benötigt der Sender einen Modulator, um das Sendesignal zu modulieren. Durch den bandbegrenzten Frequenzbereich $f_0 \pm f_{bb}$ ist eine zusätzliche Impulsanpassung empfehlenswert.

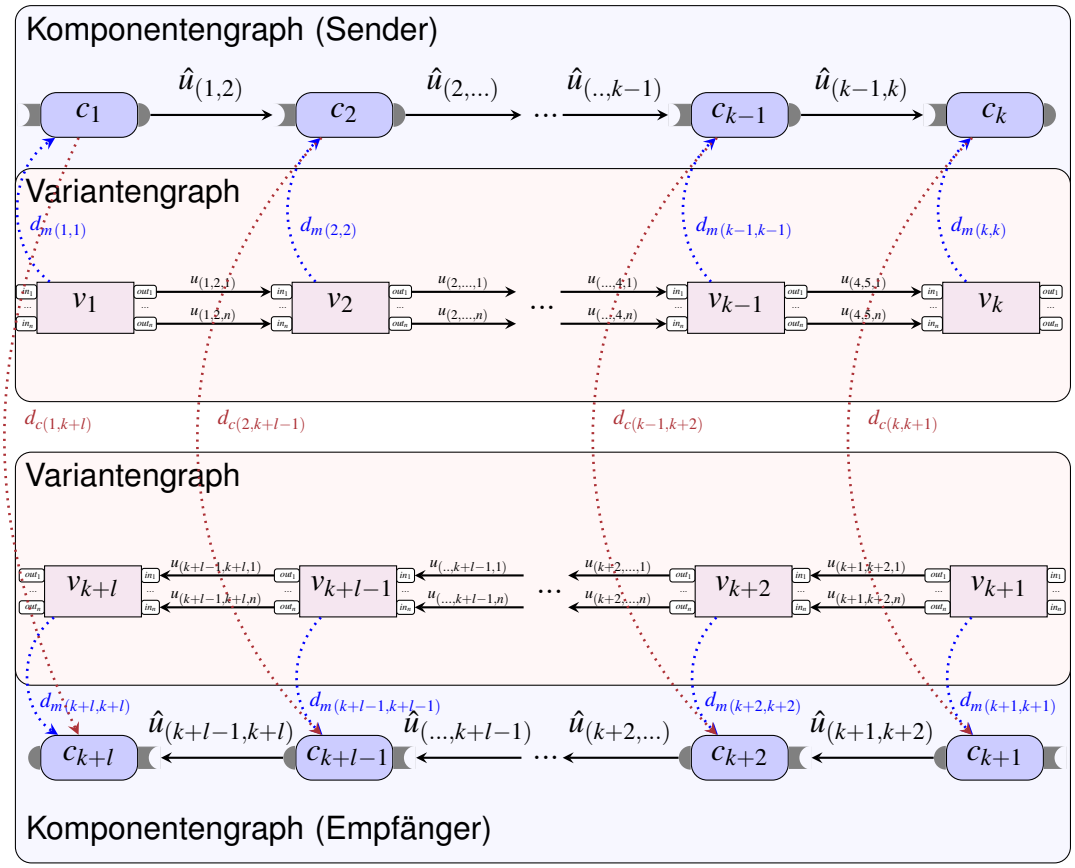


Abbildung 5.4: Schematisches Kommunikationsmodell als Graph mit Abhängigkeiten auf Komponenten- und Variantenebene (ohne Parameterzuweisung)

Für die Übertragung des Nutzsignals in dem zugewiesenen Frequenzbereich (Passband) wird ein Frequenzumsetzer (Up-Converter) benötigt. Während die terrestrische Kommunikation häufig den Schwerpunkt auf eine datenintensive Übertragung legt, wird die Kommunikation unter Wasser auf Robustheit ausgelegt. Aufgrund der langsamen Ausbreitungsgeschwindigkeit des Schalls im Wasser mit $c = 1500\,\mathrm{m\,s^{-1}}$, werden die Systeme so ausgelegt, dass die Wahrscheinlichkeit einer zeitlich teuren Sendewiederholung durch redundante Daten mithilfe der Fehlerkorrektur (Vorwärtsfehlerkorrektur (FEC)) minimiert wird.

Beispiel 5.1 — Fehlerkorrektur:
Aus den Annahmen für ein Kommunikationssystem kann eine Beschreibung der Eigenschaft bzw. des Verhaltens der Komponenten (hier am Beispiel der Fehlerkorrektur des Senders) abgeleitet und formalisiert werden.

Im folgenden wird für das eingehende Trägerbündel der Variante v_i die Notation $\hat{u}_{(in,i)}$ und für das ausgehende Trägerbündel die Notation $\hat{u}_{(out,i)}$ genutzt.

Komponentenbeschreibung: Erweitern der zu sendenden Informationen um redundante Daten, die eine Rekonstruktion der empfangenen Daten in die ursprünglich gesendeten Informationen, nach einer Manipulation durch den Kanal, ermöglicht. Durch das Auftreten von Bündel- und Streufehlern sollte das System anwendungs- und ortsspezifisch über eine Auswahl von Codierern und Kombinationen dieser verfügen. Zur Integritätsprüfung der Datenpakete muss die Komponente mindestens über eine Zyklische Redundanzprüfung (CRC), zur Analyse der Daten ohne Korrekturmöglichkeit am Empfänger, verfügen.

Formalisierte Variante: Für eine Umsetzung der Fehlerkorrektur gibt es mehrere Verfahren wie die Faltungscodierung [198] oder die Blockcodierung wie z. B. den Reed Solomon Codierer (RSC) [199], welche auch kombiniert werden können. Der Faltungskodierer, als Beispiel für die Fehlerkorrektur, kann mit einer Beschränkungslänge k_{cc}, einem Satz von j Generatorpolynomen $G{:=}G_1, G_2, ..., G_j$ und einer daraus resultierenden Coderate von $R_{cc}{=}1/||G||{=}1/j$ formalisiert werden. Die definierte Variante (Gl. (5.3)) kann somit über Gl. (5.3) den Zusammenhang zwischen dem eingehenden Trägerbündel ($\hat{u}_{(i-1,i)}$) und dem ausgehenden Trägerbündel ($\hat{u}_{(i,i+1)}$) der Variante t_i im n-ten Ausführungsschritt generell und mit Gl. (5.4) über die Träger u beschrieben werden:

$$\hat{u}_{(i,i+1)} = v_{i+1}(\hat{u}_{(i-1,i)}, k_{cc}, G) \tag{5.3}$$

$$\hat{u}_{(i,i+1)}[n] = \begin{pmatrix} u_{(i,i+1,(1)[n])} \\ \dots \\ u_{(i,i+1,(j)[n])} \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{k-1} u_{(i-1,i,[n-i])} \cdot G_1[i+1] \\ \dots \\ \sum_{i=0}^{k-1} u_{(i-1,i,[n-i])} \cdot G_j[i+1] \end{pmatrix} \tag{5.4}$$

Durch die Parametrisierung und Definition der Variablen und Polynome in Gl. (5.5), die auf dem Codierer des Voyager-Programms [200] basieren, kann der Faltungscodierer, dargestellt in Abbildung 5.1 (Variante), als Schieberegister umgesetzt werden. Eine weitere Anwendung des Codierer ist die WLAN-Kommunikation über den Institute of Electrical and Electronics Engineers (IEEE) 802.11a Standard [201] wird, Diese spezifische Variante des Codierers ist aber nur eine mögliche parametrisierte Form der vorgegebenen Komponente zur Fehlerkorrektur beim Empfänger (FEC) mittels Blockcodes. Alternativ sind Varianten nicht nur mit anderer Parametrisierung sondern auch durch andere Strukturen in Form von Automaten oder anderen möglich.

$$k_{cc}{=}7,\ G{:=}\{G_1, G_2\} \text{ mit } \begin{pmatrix} G_1{=}x^6{+}x^5{+}x^4{+}x^3{+}1 \\ G_2{=}x^6{+}x^4{+}x^3{+}x^1{+}1 \end{pmatrix} \tag{5.5}$$

5.1.1 Abhängigkeiten der Datenflussgrafen

Eine weitere Herausforderung der Implementierung eines Kommunikationssystems sind die Abhängigkeiten. Während die Beschreibung der Eigenschaften einer Komponente noch textuell umschrieben werden können, müssen die Eigenschaften und Abhängigkeiten auf Ebene der Varianten explizit spezifiziert werden. Die Bedingtheit des Kommunikationssystems, beschränkt auf den digitalen Bereich des Kommunikationssystems, umfasst daher zwei Bereiche von Abhängigkeiten, zum einen die modulübergreifenden Abhängigkeiten zwischen den Trägerbündeln der Komponenten und zum anderen die Abhängigkeiten der Träger verknüpfter Varianten.

Bei der Annahme, dass zur Kommunikation zwischen zwei oder mehr homogenen Systemen eine identische Parametrisierung genutzt werden, können die Abhängigkeiten zwischen Komponenten und Varianten vereinfacht zwischen Sender- und Empfängermodule formalisiert und abgebildet werden. Bei einer Erweiterung des Entwicklungsansatzes auf inhomogenen Strukturen muss die Abhängigkeit auf Komponenten- und Variantenebene systemübergreifend zwischen dem Sender- und Empfängermodul abgebildet werden.

Durch eine Komponente für die Faltungscodierung, mit festgelegter Parametrisierung wie in dem Beispiel 5.1, müssen die Module des Senders und Empfängers je eine Komponente, implementiert als Variante, beinhalten, welche die definierten Eigenschaften der Fehlerkorrektur erfüllt. Die Empfänger-Komponente mit ihrer Implementierung, im folgendem als inverse Variante bezeichnet, muss dazu die Rekonstruktion des ursprünglichen Trägers, am Beispiel der Fehlerkorrektur, abhängig von der Anzahl und Art der auftretenden Fehler, definiert in (Gl. (5.6)) und (Gl. (5.7)) für die fehlerfreie Kommunikation, ermöglichen.

$$\hat{u}_{(c_{FEC}^{-1},c_{FEC+1}^{-1})} = c_{FEC}^{-1}(c_{FEC}(\hat{u}_{(FEC-1,FEC)}, k_{cc}, G), k_{cc}, G) \tag{5.6}$$

$$u_{(v_{FEC}^{-1},v_{FEC+1}^{-1},)} = v_{FEC}^{-1}(v_{FEC}(u_{(FEC-1,FEC,)}, k_{cc}, G), k_{cc}, G) \tag{5.7}$$

Diese modulübergreifenden Abhängigkeiten einer Variante v_i und ihrer inversen Variante v_j gilt nur in einer Richtung und kann generell über die Gl. (5.8) mit den Parametern $p \in P$ über die gerichtete Abhängigkeit $d_{c(i,j)} \in D_v$ auf Trägerbündel und über Gl. (5.9) die Trägerabhängigkeit $d_{v(i,j)} \in D_s$ beschrieben werden:

$$d_{c(i,j)} := c_j(c_i(\hat{u}_{(i-1,i)}, P_i), P_j) = \hat{u}_{(j,j+1)} \tag{5.8}$$

$$d_{v(i,j)} := v_j(v_i(u_{(i-1,i,)}, P_i), P_j) = u_{(j,j+1,)} \tag{5.9}$$

Zusätzlich muss die Gleichheit der eingehenden Träger von v_i und des Trägerbündels von C_i mit den ausgehenden Trägern von v_j und dem Trägerbündel

von C_j (Gl. (5.10) and (5.11)), wenn das ursprüngliche Trägerbündel die Träger vollständig rekonstruiert werden soll, gelten.

$$\hat{u}_{(i-1,i)} = \hat{u}_{(j,j+1)} \tag{5.10}$$

$$u_{(i-1,i,)} = u_{(j,j+1,)} \tag{5.11}$$

Die Abhängigkeiten $d_{v(i,j)}$ unterscheiden sich dabei durch die explizite Spezifikation der umgesetzten Parameter von $d_{c(i,j)}$, wie das Beispiel des Faltungscodierer in Gl. (5.4) and (5.5) zeigt.

Durch das Vererben der Trägereigenschaften der Trägerbündel benötigen die Varianten der Komponenten auch eine Spezifikation der Schnittstellen. Wird von einem diskreten System ausgegangen, können die Zeitschritte, in der Signale übertragen werden (sample-time, T_s), der jeweiligen Variante mit $T_{s(dir,i,m,n,k)}$ sowie die numerische Darstellung mit $D_{Type(dir,i,m,k)}$ definiert werden. Die Beschreibung der Schnittstellen beinhaltet neben der zeitlichen und numerischen Auflösung noch die Parameter wie die Dimension und numerische Beschränkungen, wie die Auflösung, den maximal oder minimal darzustellenden Wert. Während die Variable $dir := \{in, out\}$ die Richtung, beschränkt auf ein- oder ausgehend, des Trägers beschreibt, gibt i die verknüpfte Komponente, m die gewählte Variante und k die Schnittstelle des Trägers an. Da die Träger der Trägerbündel, repräsentiert über die Kanten $\hat{u}_{(i,j)}$, die Spezifikationen der ausgewählten Variante erben, muss gelten, dass die zeitlichen, numerischen und sonstigen definierten Spezifikationen durch Träger verknüpfter Schnittstellen identisch sind. Die Verbindungen der Schnittstellen können mit $d_{P(i,k,j,l)} \in D_P$ (Gl. (5.12)) zusammengefasst und kompakt beschrieben werden. Dabei beschreibt $d_{P(i,k,j,l)}$ die Verknüpfung des Ausgangsports k der Komponente i und mit dem Eingangsport l der Komponente j.

$$d_{P(i,k,j,l)} := \begin{cases} T_{ss(out,i,m,k)} = T_{s(in,j,p,l)} \\ D_{Type(out,i,m,k)} = D_{Type(in,j,p,l)} \\ \dots \end{cases} \tag{5.12}$$

Komponenten können nur verknüpft werden, wenn eine direkte Nachbarschaft, definiert über die gerichteten Kanten $e_i \in E$, besteht oder die Verknüpfung vom Ein- bzw. zum Ausgang des übergeordneten Moduls besteht. Eine generelle direkte modulübergreifende Abhängigkeit der Varianten bezüglich der zeitlichen und numerischen Auflösung oder der Funktion ist nicht immer gegeben und muss in diesem Modell daher explizit angegeben werden.

Entgegen der Annahme, wie sie im terrestrischen Bereich gemacht werden, dass die Kommunikation in beiden Richtungen zwischen zwei Systemen identisch sei, kann es im Bereich der akustischen Unterwasserkommunikation zu erheblichen Unterschieden in der Signalausbreitung kommen. Wodurch eine Wahl von

unterschiedlichen Eigenschaften des jeweiligen Sende- und Empfangspfads Abschnitt 1.2 motiviert wird. Daher werden die definierten Abhängigkeiten immer auf einen Kommunikationspfad zwischen zwei Systemen bezogen und gelten nur dann für alle Kommunikationssysteme, wenn allen Kommunikationspfaden die gleichen Eigenschaften zugeordnet werden kann und es sich um eine homogene Kommunikationsstruktur oder homogenes Kommunikationsnetzwerk handelt.

5.1.2 Multivariater Datenflussgraf

Entgegen der direkten Hardwareentwicklung bietet der Ansatz des modellbasierten Entwurfs, neben der hohen Wiederverwendbarkeit und der automatisierten Codeerstellung, die Möglichkeit auch formal beschriebenes Verhalten und Strukturen zu analysieren und adaptieren. Allerdings benötigt auch bei der modellbasierten Entwicklung jede alternative Interpretation, einer Komponente, eine eigene Umsetzung, welche gepflegt und ggf. angepasst werden muss. Zudem können Komponenten durch unterschiedliche Varianten, die sich in ihrer Funktion, Struktur oder Algorithmus unterscheiden, optional umgesetzt werden. Um auch Varianten flexibler abzubilden und damit alternative Umsetzungen zu unterstützen, wurde der eingeführte modellbasierte Ansatz zu einer auf Varianten basierenden modularen Beschreibung zur Interpretation und Integration wählbarer Ansätze erweitert. Mittels der multivariaten und multivariaten Eigenschaft des Modells können optionale Ansätze, sowie Kombinationen aus bereits erstellten Modulen, Komponenten und Varianten einfacher umgesetzt, analysiert und verglichen werden, ohne ein grundlegend neues Modell zu definieren und zu pflegen. Dieser Aufbau erlaubt es nicht nur eine schnelle Hardwareentwicklung zu fördern, sondern vereinfacht die Analyse von komplexeren Algorithmen oder Ansätzen, deren Umsetzbarkeit und Nutzbarkeit bislang nicht feststeht, ohne den Aufwand einer teuren Implementierung in Hardware und der damit verbundenen kostspieligen Testreihen in Gewässern. Hinzu kommt die Zeitersparnis bei der Entwicklung und Analyse neuer Ansätze durch die Wiederverwendung bereits erstellter Komponenten, Varianten, deren Abhängigkeiten und Beschränkungen, sowie schon erstellter Testszenarien und Testumgebungen.

5.1.3 Modifikationen

Die eingeführte Variante v_i einer Komponente c_i wird dazu durch ein nicht leeres Set von n Varianten $v_{(i,j)} \in C\, i \in \mathbb{N} | 1 \leq i \leq n$ der Komponente c_i ersetzt, dabei definiert $v_{(i,j)}$ die j-te Variante der i-ten Komponente. Durch die modulare

multivariate Varianten-basierte Anpassung des Modells müssen sich zusätzliche ergebende Abhängigkeiten über den Varianten-Grafen $G_V(V,D)$ definiert und Einschränkungen festgelegt werden. Eine zusätzliche Einschränkung über die Menge von Varianten für ein Kommunikationssystems, basierend auf einem gültigen Komponenten- und Varianten-Graf, ist die Voraussetzung, dass jeder Komponente genau eine aktive Variante zugeordnet wird und kann über die Gl. (5.13) formuliert werden:

$$\forall c_i \in C: \sum_{t_{i,j} \in G_V} v_{i,j} \tag{5.13}$$

Durch die Erweiterung der Module durch zusätzliche oder optionale Komponenten würden sich auch Kombinationen ergeben, welche nicht alle zur Verfügung stehenden Komponenten und deren definierten Varianten beinhalten können. Ein Beispiel wäre die Phasenrekonstruktion einer Phasenumtastung (PSK), welche für Frequenz-Umtastung (FSK) nicht genutzt werden kann. Um die bereits bestehende Anforderung der direkten Nachbarschaft, das alle Komponenten c_i nur mit dem Vorgänger c_{i-1} und Nachfolger c_{i+1} verknüpft werden können, und die Anforderung einer nicht leeren Komponente (genau eine aktive Variante) zu erfüllen, wird zusätzlich die Null-Variante eingeführt. Die Null-Variante ist eine Variante, deren Funktion über das Durchschleifen des Eingangssignales, ohne Verzögerung oder numerische Manipulation, auf das Ausgangssignal definiert ist (Gl. (5.14)) und entspricht immer der Variante mit Indiz $j=0$.

$$\hat{u}_{(i,i+1)} = v_{(i,0)}(\hat{u}_{(i-1,i)}) = \hat{u}_{(i-1,i)} \tag{5.14}$$

Diese Null-Variante steht nur in Komponenten zur Verfügung, die für mindestens eine mögliche Kombination von Varianten optional sind oder die Komponente theoretisch nicht genutzt werden kann. Im Beispiel der Abbildung 5.5 beinhaltet die Komponente c_2 eine Null-Variante; in diesem Fall könnte es für die möglichen Codierungen stehen, mit denen die Kommunikation verbessert werden soll. Hier ermöglicht die Null-Variante nun auch die Auswahl keiner Codierung, welches beim Empfänger auch keine Decodierung impliziert, abgebildet über die Abhängigkeit d_v der Variante.

Damit muss sichergestellt werden, dass die gerichteten Abhängigkeiten d_v über den Graphen G_V erfüllt werden. Für eine gerichtete Abhängigkeit $d_{TA(i,k,j,l)}$ zwischen einer Variante $v_{(i,j)}$ und $v_{(i,j)}$ muss die Gl. (5.15) gelten.

$$\forall v_{(i,j)} \in G_V : \forall d_{TA(i,k,j,l)} \in D_V : v_{(i,j)} \in G_V \tag{5.15}$$

Bei der Wahl einer Variante $v_{(i,j)}$ mit einer gerichteten Abhängigkeit $d_{v(k,l)}$ auf mindestens eine andere Variante müssen die definierten Abhängigkeiten Komponentenweise erfüllt werden und die abhängige(n) Variante(n) $\{v_{(j,l)}\}$ Element(e)

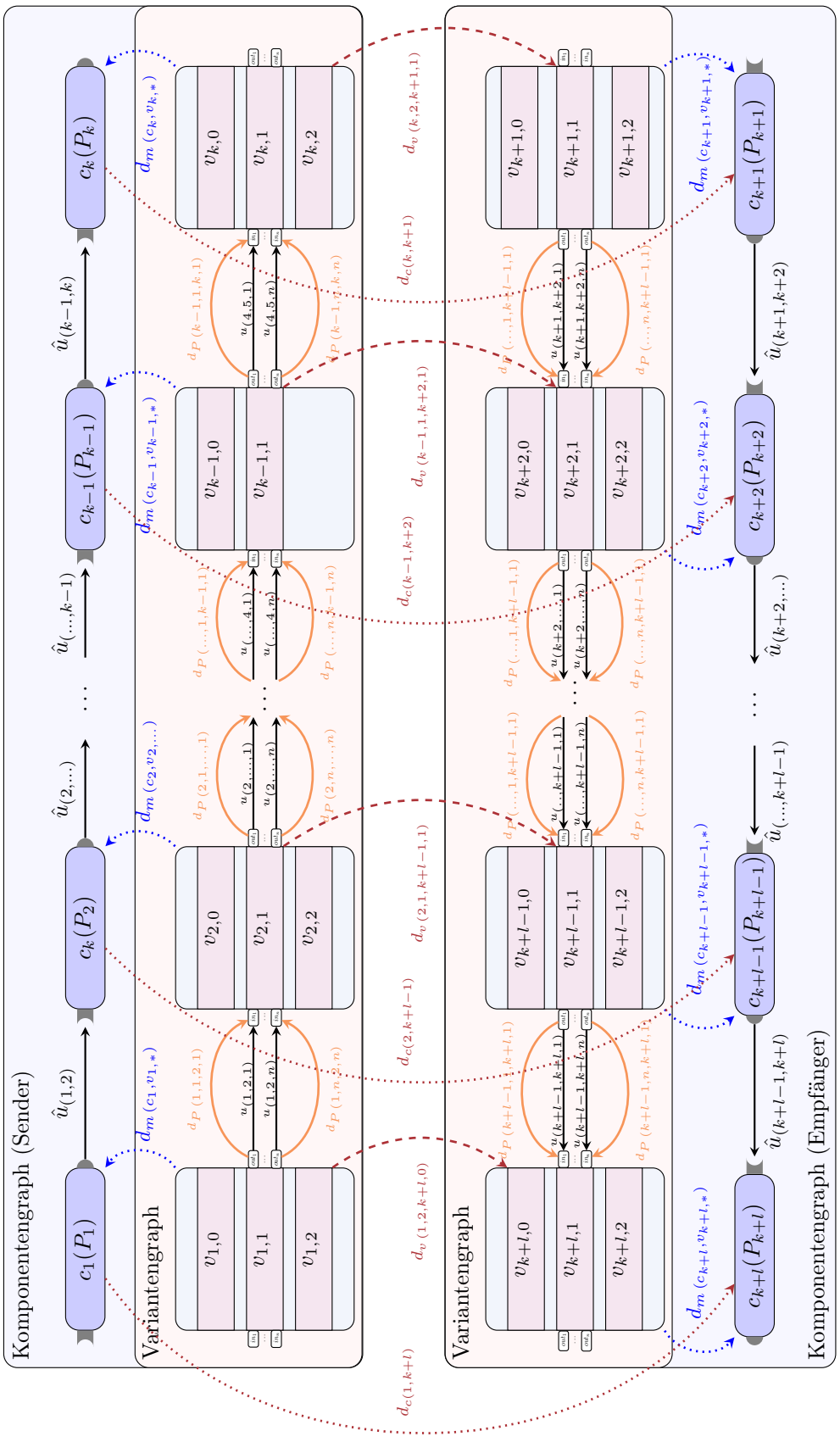


Abbildung 5.5: Schematische modulare multivariate Darstellung eines verketteten Kommunikationsmodells auf Komponenten- und Variantenebene

des Variantengrafen G_V bei der Auswahl von $v_{(i,j)}$, aber nicht umgekehrt, sein. Eine angepasste Darstellung für das schematische Kommunikationsmodell aus Abbildung 5.4 gibt die Abbildung 5.5 wieder. Die Abbildung 5.5 zeigt eine beispielhafte gültige Kombination von Varianten (G_V) der schematisch zur Verfügung stehenden Komponenten unter Berücksichtigung der Abhängigkeiten.

5.2 Modellbeschreibung

Unabhängig von der späteren Umsetzung und den genutzten Entwurfsumgebungen oder Programmen muss sowohl das Modell ganzheitlich als auch die jeweilige genutzte Implementierung im Speziellen eindeutig beschrieben werden. Diese Beschreibung kann dann zur Kommunikation, Reproduktion oder dem Datenaustausch zwischen Werkzeugen oder Programmen genutzt werden. Für den abbildbaren Raum des Modells sowie der Beschreibung einer gültigen Umsetzung wird in dieser Arbeitdie Erweiterbare Auszeichnungssprache (XML) genutzt und die Struktur und Grammatik über eine Dokumenttypdefinition (DTD) (Listing A.20) vorgegeben.

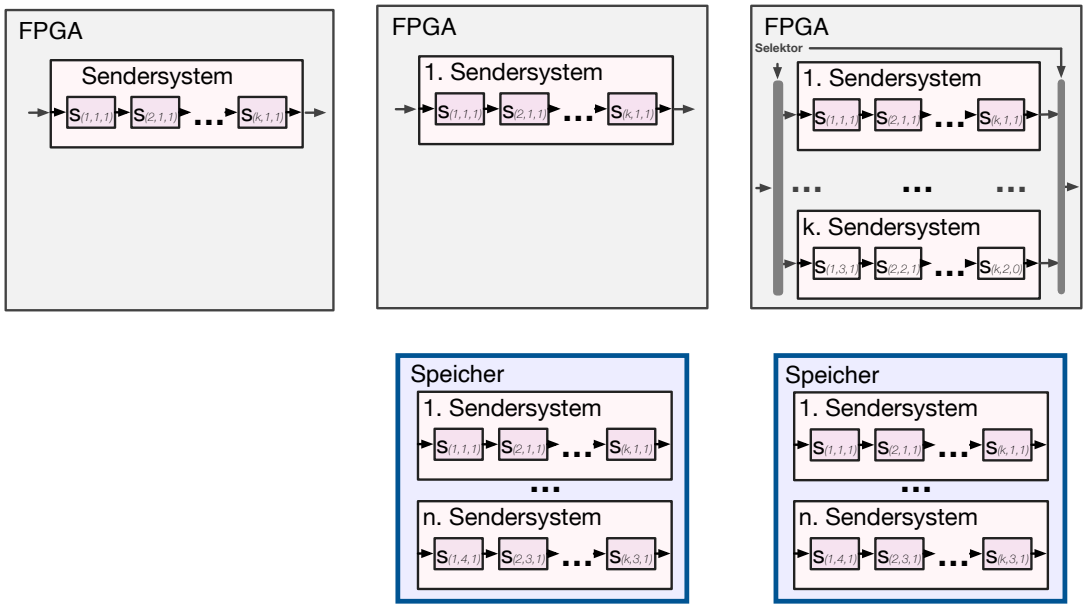
Die Basis der Modell- und Implementierungsbeschreibung ist die Auflistung und Verknüpfung der Komponenten mit den untergeordneten Varianten, ihren zugeordneten Parametern (-Bereichen) sowie die Abhängigkeiten der Varianten. Die Varianten werden dazu den übergeordneten Komponenten zugeordnet und die Abhängigkeiten der Varianten über gerichtete (Abhängigkeits-) Kanten abgebildet. Sowohl in Komponenten und Varianten können Parameter gesetzt werden. Die Parameter werden über einen Parameter-Bereich (*rangeattribute*), einen festen unveränderlichen Wert (*fixedattribute*) oder durch selektive Wertepunkte (*selectattribute*) basierend auf der DTD und beschränkt über die vorgegebene Typendefinition beschrieben. Die Parameterwerte und -Bereiche sind zentral in der Klasse *AUVModem* als Konstanten hinterlegt und werden bei der Generierung der Gesamtmodellbeschreibung (*GenerateXML.m*) inkludiert. Nach einer automatischen oder manuellen Auswahl der genutzten Varianten, Parameterwerte, der Auflösung des abbildbaren Wertebereiches und dem Erfüllen der Abhängigkeiten $d_{s(i,j)}$ kann mit einem Synthese- bzw. Simulationstool das definierte Modell generiert, simuliert und analysiert werden.

5.3 Plattformadaption

Unter der Prämisse, dass das Modell zur Generierung von Hard- und Software genutzt wird, muss dies auch bei der Erstellung und Modellierung berücksichtigt werden. Aktuelle Field Programmable Gate Array (FPGA) unterstützen hierbei unterschiedliche Methoden der Rekonfiguration. Grundlegend können diese Methoden in drei Hauptbereiche untergliedert werden: Der vollständigen, partiellen und dynamisch partiellen Rekonfiguration [202, 203]. Abhängig von der angestrebten Nutzung dieser Konfigurationsmethoden und der Verwendung des Modells müssen weitere Einschränkungen und Abhängigkeiten definiert werden. Mit dem eingeführten Modell lassen sich Varianten, wie beispielsweise in Abbildung 5.4, umsetzen. Um den Bereich der dynamischen und partiell dynamischen Rekonfiguration zu nutzen, muss das Modell über austauschbare Varianten verfügen, die jeweils innerhalb einer Komponente dynamisch zur Laufzeit gewechselt werden können und an die Kommunikationsstruktur des Modems angepasst sind. Diese auswechselbaren Varianten können auch über ein entsprechendes Multiplexer-Netzwerk gesteuert werden und benötigen, auf Kosten eines erhöhten Energie- und Platzbedarfs, keine Rekonfiguration mehr.

5.3.1 Verkettetes Variantenmodell

Eine Anwendung unter der Annahme, dass ein komplettes Modell der physikalischen Schicht des Modems umgesetzt und i) das komplette FPGA rekonfiguriert oder ii) partiell immer ein komplettes Modem ausgetauscht wird, kann das bereits definierte Modell des Modems bestehen bleiben. Durch die gleichbleibenden und im Vorfeld festgelegten äußeren Schnittstellen lassen sich die verkettete Varianten mit allen drei Methoden der Rekonfiguration integrieren. Die Abbildung 5.6 zeigt schematisch mögliche Anwendungen auf einem FPGA am Beispiel eines Senders. In der Abbildung 5.6a wird dazu genau ein Modell umgesetzt und kann zur Laufzeit nicht rekonfiguriert werden, während der rekonfigurierbare Ansatz in Abbildung 5.6b mehrere mögliche vordefinierte Modems zur Verfügung stellt, die zur Laufzeit durch identische Schnittstellen ausgetauscht werden können. Der dritte Ansatz in Abbildung 5.6c zeigt eine selektive Nutzung von vordefinierten Modems, die über ein Multiplexer-Netzwerk zur Laufzeit ausgewählt und die einzelnen Modems in den vorgesehenen Slot dynamisch mit vordefinierten Varianten rekonfiguriert werden können.



(a) Starr **(b)** Rekonfigurierbar **(c)** Rekonfigurierbar und selektiv wählbar

Abbildung 5.6: Schematische Darstellung der Nutzung eines verketteten Variantenmodells des Senders auf einen FPGA

5.3.2 Flexibles Variantenmodell

Um den Vorteil einer partiellen Rekonfiguration von Systemkomponenten auf FPGAs zu nutzen, muss das Systemmodell zusätzlich an die Anforderungen der partiellen Rekonfiguration, schematisch in Abbildung 5.7 dargestellt, angepasst werden. Durch die erforderliche Austauschbarkeit von allen Varianten innerhalb einer Komponente, schematisch in Abbildung 5.7 abgebildet, müssen die zeitlichen und numerischen Anforderungen und Abhängigkeiten für alle wählbaren Varianten einer Komponente identisch sein. Dazu werden die Abhängigkeiten der verknüpften Schnittstellen, im flexiblen Variantenmodell, mit $d_{P(i,k,j,l)} \in D_P$ (Gl. (5.12)) allgemeingültig und für alle Varianten einer Komponente bindend beschrieben werden. Die Schnittstellenverknüpfung $d_{P(i,k,j,l)} \in D_P$ in Gl. (5.12) muss daher allgemeiner über Gl. (5.16) angepasst werden. Damit gelten die Spezifikationen der Schnittstellen für alle Varianten der verknüpften Komponenten und müssen für alle Varianten einer jeweiligen Komponente identisch sein.

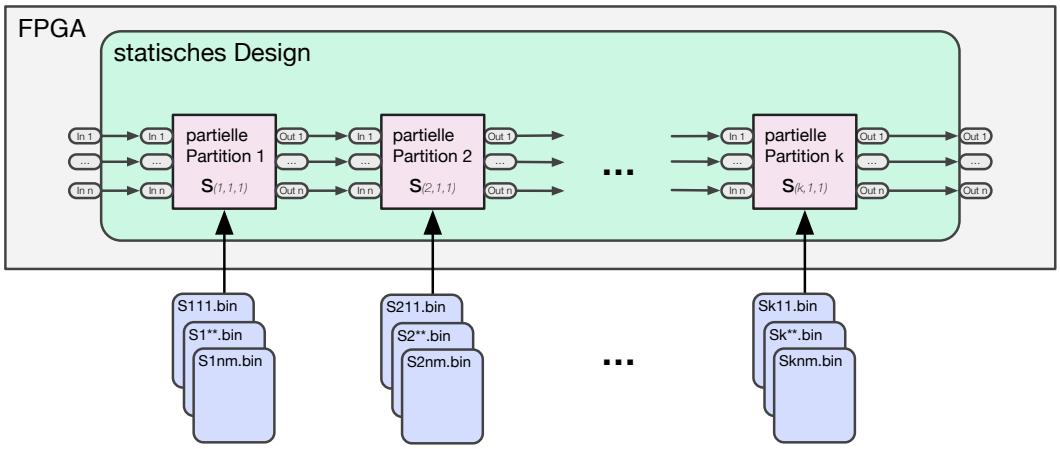


Abbildung 5.7: Schematische Darstellung der partiellen Nutzung eines flexiblen Variantenmodells auf einem FPGA

Sie werden idealerweise extern vorgegeben.

$$d_{P(i,k,j,l)} := \begin{cases} T_{s(out,i,k)} = T_{s(in,j,l)} \\ DT_{(out,i,k)} = DT_{(in,j,l)} \\ \dots \end{cases} \tag{5.16}$$

Eine komplette Überarbeitung der Varianten und die Anpassung an eine generelle Abtastzeit wäre sehr aufwendig und würde den Grundsatz der Wiederverwendung und Wartbarkeit verletzen. In dieser Arbeit wird daher für die Erweiterung des verketteten (Abbildung 5.6) zu einem flexiblen Variantenmodell (Abbildung 5.7) eine diskrete Abtastzeittransferfunktion für bestehende Varianten definiert. Die Abtastzeittranfersfunktion (STTF) verzögert oder beschleunigt die Abtastzeit der jeweiligen Varianten ohne die Ordnung oder die Integrität der Daten zu ändern und angepasst an die Vorgabe der Komponente auf Kosten der benötigten Ressourcen und einer Verzögerung bei einer Beschleunigung. Die Abbildung 5.8 zeigt am Beispiel der Quadraturphasenumtastung (QPSK)-Modulation den zeitlichen Transfer über eine schematische Abbildung mit einer möglichen Umsetzung mit Simulink. Generell wird am im Modul des Senders, mit Ausnahme des Frequenzumsetzers, die Abtastzeit der Varianten erhöht, was die benötigte Verarbeitungszeit erheblich reduziert. Erst im Frequenzumsetzer wird die Abtastrate auf die zur Übertragung erforderliche Abtastzeit zurück transferiert. Durch das Erben einer Variante und der Abtastzeittranfersfunktion (STTF) bleiben die Eigenschaften und Funktionalität erhalten und kann auch auf die partielle Rekonfiguration angewendet werden. Zusätzlich kann die Taktung aller Varianten

auf einen einheitlichen Wert festgelegt und die Verarbeitungszeit entsprechend skaliert werden, wie am Beispiel der Modulation in beispiel 5.2 gezeigt wird.

Beispiel 5.2 — QPSK-Modulation:
Im Beispiel der Modulation, dargestellt in Abbildung 5.8, wird ein einfachen First In – First Out (FIFO)-Speicher, dessen benötigte Anzahl von zu speichernden Daten k von der Anzahl der Daten, der Verwendung und der Beschleunigung $a>1$ oder Verzögerung $0<a<1$mit $a\in\mathbb{N}*$ abhängt, genutzt. Die maximale Anzahl zu transferierenden Daten n^* bestimmt dabei die benötigte Größe des Zwischenspeichers mit $k=\frac{n^+}{a}+1$. Bei einem Transfer unbestimmter Größer und einer Steuerung, die auf den Eingang *validIn* und *Num* des FIFO-Speichers sensitiv ist, kann die Verzögerung T_{del} somit die Werte im Intervall $[2{:}1{:}k=\lceil\frac{n^+}{a}\rceil+1]\cdot T_s$, mit T_s als diskreter Zeitschritt, annehmen. Wird der Steuerung der QPSK-Variante zusätzlich die Information der aktuell zu transferierenden Datenmenge d in Symbolen zur Verfügung gestellt, kann die Verzögerung mit $T_{del}=(\lceil\frac{d}{a}\rceil+1)\cdot T_s$ bestimmt werden.

Anmerkung: Die Variante der QPSK-Modulation kann mittels der Transferrate $a=2$, die Rate der ausgehenden Signale $T_{s_{out}}$ der Rate der eingehenden Signale $T_{s_{in}}$ gleichsetzen. Hierbei ist die Platzierung der Transferfunktion vor oder nach dem QPSK-Modulator möglich. Die Variante der QPSK-Modulation setzt jeweils zwei Bits auf ein komplexes Symbol um, womit die Platzierung zwar Auswirkungen auf die Anzahl der zu speichernden Daten hat, aber durch die Verwendung eines komplexen Datentyp wieder ausgeglichen wird. Damit gibt es bezüglich des benötigten Speicherplatzes keinen Unterschied bei der Wahl der Platzierung.

5.3.3 Multiplexer-basiertes Variantenmodell

Durch eine Erweiterung des flexiblen Variantenmodells über eine statische Multiplexer-Struktur, schematisch in Abbildung 5.9 gezeigt, können Teile oder, je nach Größe der Plattform, gegebenenfalls alle Varianten einer Komponente des multivariaten modellbasierten Systems abgebildet und genutzt werden. Die Umsetzung beschränkt sich dabei, wie auch beim flexiblen Variantenmodell, auf die festgelegten statischen Parameter zur Kompilierzeit. Die Abbildung 5.9 zeigt einen schematischen Ansatz, der einen ausgewählten Teil der Varianten integriert und optional die restlichen Varianten nach Bedarf partiell rekonfiguriert. Neben den erweiterten Varianten (Abschnitt 5.3.2) muss die generelle Struktur nur noch um ein Multiplexer-Netzwerk erweitert werden. Optional kann die Taktrate

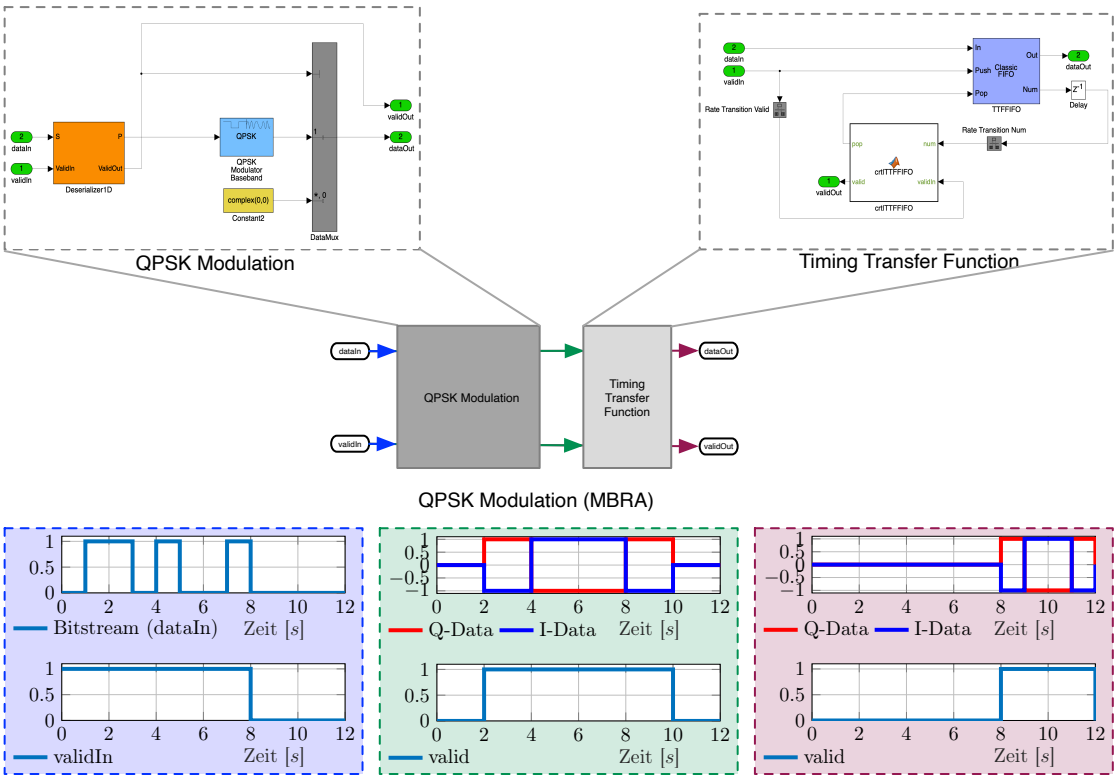


Abbildung 5.8: Umsetzung der QPSK-Modulation mit STTF in Simulink

der einzelnen Komponenten identisch gesetzt werden, was eine Ansteuerung des Netzwerkes vereinfachen würde. Der Nachteil des Multiplexer-Netzwerkes allerdings die Anforderung an gleiche Schnittstellen der Varianten, womit die Signalverknüpfungen der Multiplexer aufbläht werden müssen. Eine Umsetzung mittels des Multiplexer-Netzwerkes in Simulink ist in beispiel 5.3 abgebildet.

Beispiel 5.3 — Multiplex-Netzwerk (Block-Codierer):
Die Multiplexer-basierte Umsetzung des Senders wird in Abbildung 5.10 am Beispiel der Varianten des Blockcodierer gezeigt. Der wesentliche Unterschied ist, dass alle Varianten aktiv sind und Signale verarbeiten, aber nur die Verbindungen der über den Selektor ausgewählten Variante werden über den Multiplexer weitergeleitet. Die Auswahl der genutzten Verbindungen erfolgt jeweils über einen Bus-Selektor. Die Implementierung setzt voraus, dass die Signalnamen, Signaldatentypen und Signalperioden der einzelnen Varianten identisch sind und wird mittels der Abtastzeittransferfunktion STTF und vordefinierter Datentypen und Auflösungen umgesetzt.

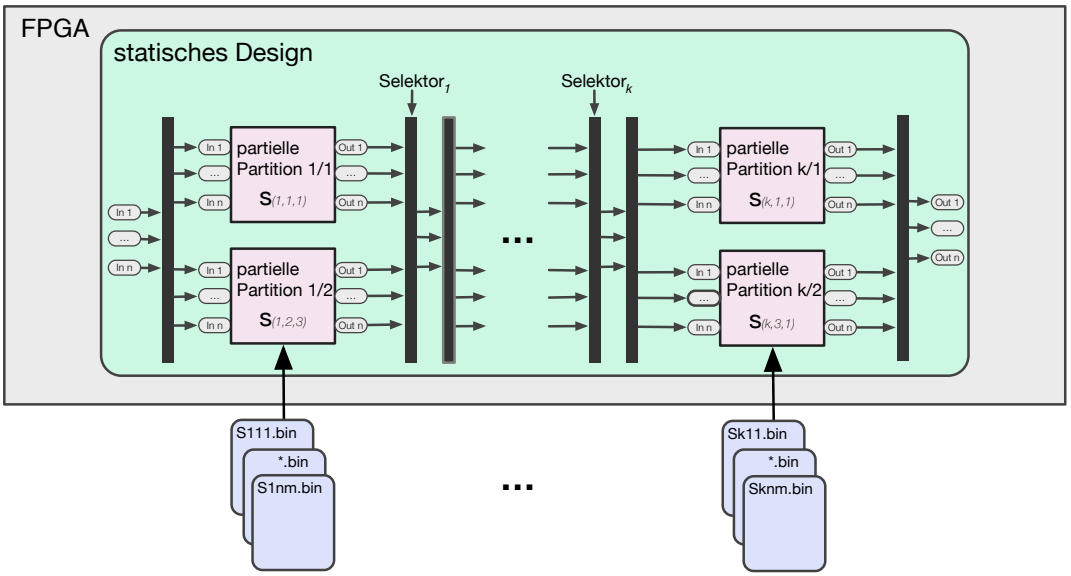


Abbildung 5.9: Schematische Darstellung des Multiplexer-Modells auf einem FPGA

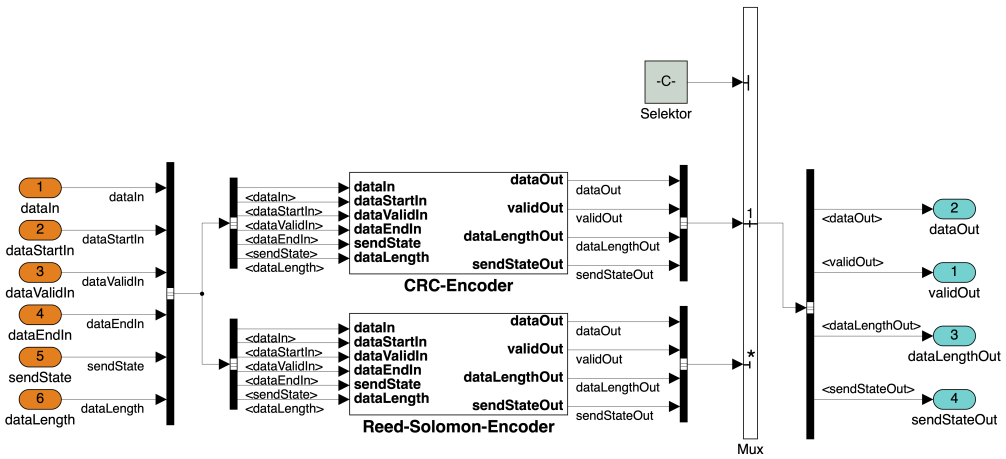


Abbildung 5.10: Multiplexernetzwerks am Beispiel des Block-Codierers

5.4 Umsetzung

Für die Umsetzung des modellbasierten Kommunikationssystems wird MATLAB/Simulink [105] verwendet, welches eine Reihe von integrierten Tools und Anwendungen bietet. Die Software unterstützt die Entwicklung mithilfe einer eigenen Hochsprache, einer grafischen Schnittstelle oder einer Kombination

aus beidem. Zusätzlich stellt MATLAB/Simulink Compiler für eine Vielzahl von Sprachen, wie Hardwarebeschreibungssprache (HDL) und Hardwarearchitekturen wie Teilserien der FPGA-Hersteller Xilinx [204] und Intel [205] zur Verfügung. Überdies verfügt MATLAB/Simulink über einen HDL-Coder [206], der es ermöglicht, aus einem eingeschränkten Umfang von optimierten oder selbst erstellten Blöcken und Funktionen VHDL oder Verilog für die abschließende Synthese der Hardware zu erzeugen.

5.4.1 Simulinkmodell

Um die Bitübertragungsschicht des digitalen Passbandbereiches des Unterwassermodems synthetisierbar umzusetzen, wurde jede Variante, basierend auf einer Referenzfunktion getestet und in einer Bibliothek zentral hinterlegt. Die Komponenten des Sender- und Empfängermoduls referenzieren mittels der Komponenten auf diese zentral verwalteten Varianten. Über das Einlesen und Auswerten einer spezifischen Modelldefinition in Form einer definierten XML-Struktur werden in MATLAB mittels Skript-basiert erzeugter Enum-Klassen die definierten Varianten aktiviert und bilden ein vollständiges System im Gesamtmodell ab. Das Sender- und Empfängermodul, definiert über die XML-Struktur, ist als erweitertes Simulink-Modell in Abbildung 5.11 auf physikalischer Ebene abgebildet und wird im Folgenden als Test-Modell bezeichnet.

Das Test-Modell basiert auf der Stimulation der Steuersignale und dem Datenrahmen, dem Kommunikationssystem auf physikalischer Ebene, dem Sender- und Empfängermodul zur Verarbeitung und Aufbereitung der Signale sowie dem Kanal zur Abbildung des umgebungsspezifischen Verhaltens der akustischen Signale im Wasser. Das Modell wird durch zwei Bereiche ergänzt: die Auswertung der Datenrahmen und die Statistik in Form von Rahmen- und Fehlerzählern.

5.4.1.1 Komponenten

Die Abbildungen 5.12 and 5.13 zeigen die im Rahmen dieser Arbeit implementierten Komponenten des gerichteten Kommunikationsmodells des Senders und Empfängers, die im folgenden kurz eingeführt werden: Die Ansätze der Varianten basierend auf gängigen bzw. etablierten Konzepten. Drei Ausnahmen bilden hierzu die Datenrahmenerkennung, der variable Blockcodierer und die flexiblen Up- und Down-Converter.

RAM-Schnittstelle:

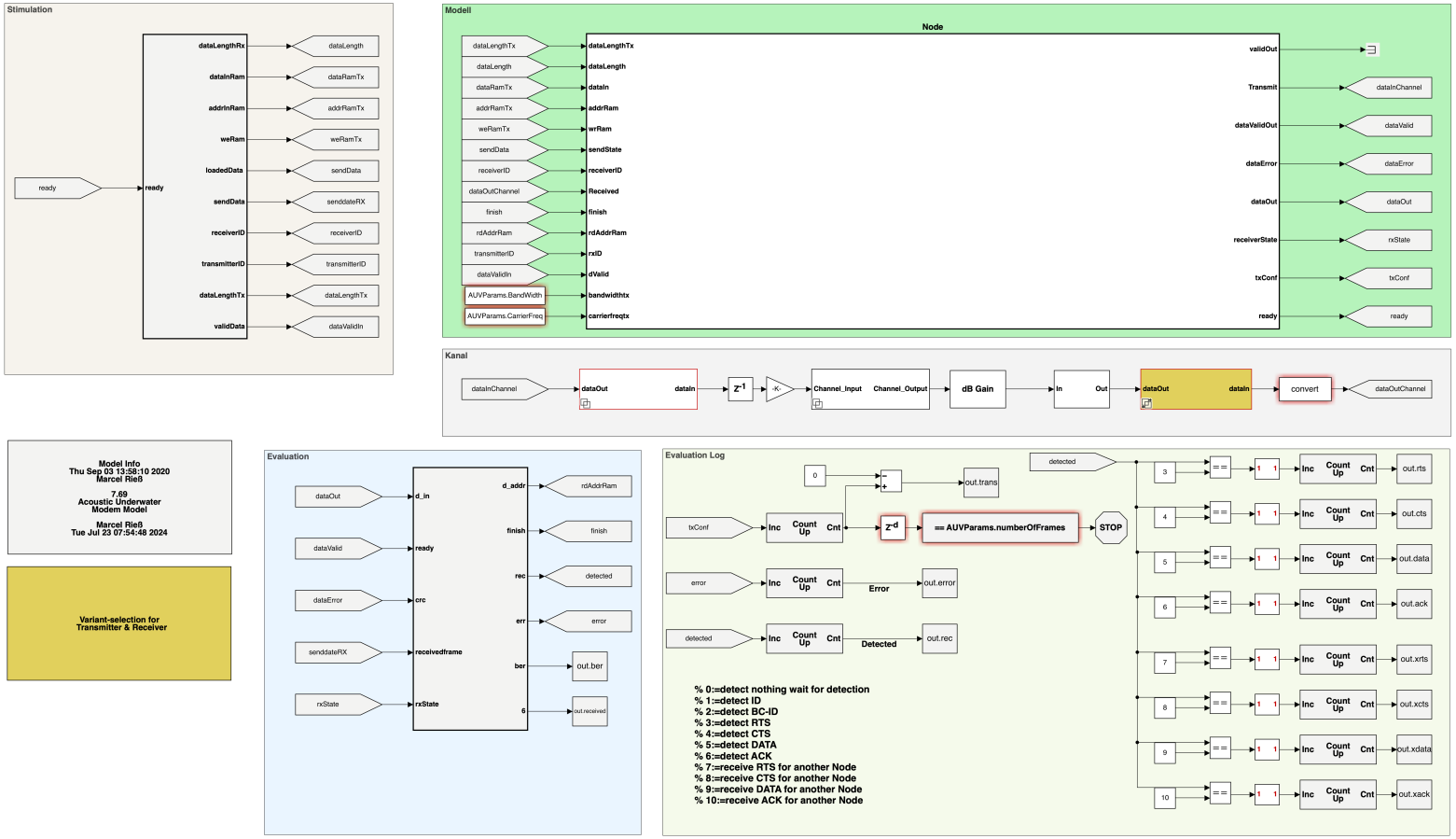


Abbildung 5.11: Test-Modell in Simulink

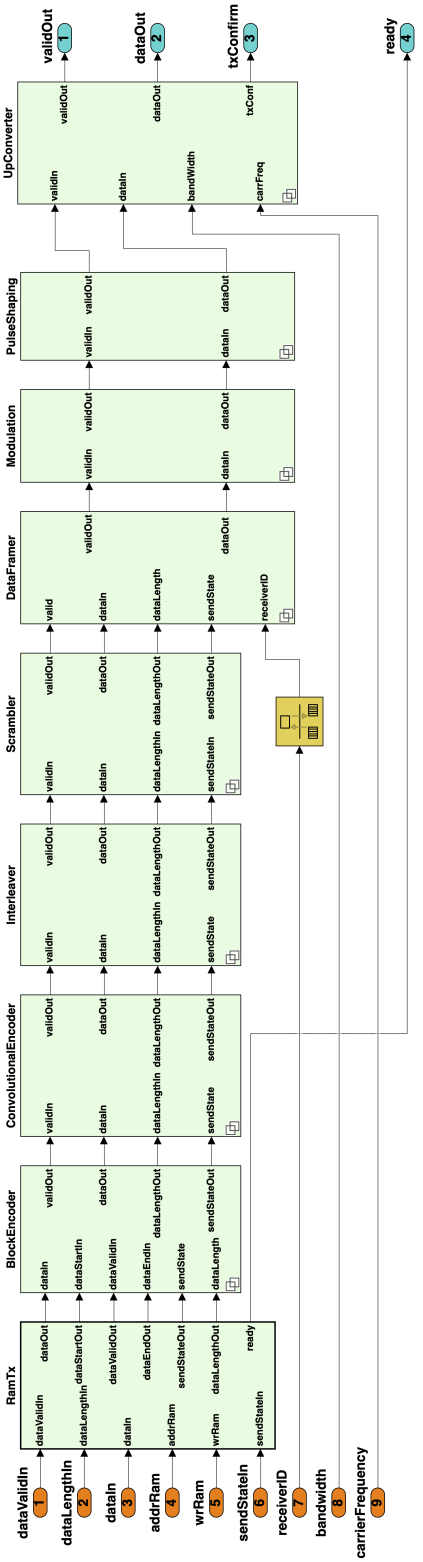


Abbildung 5.12: Simulink-Modell des Sender-Moduls

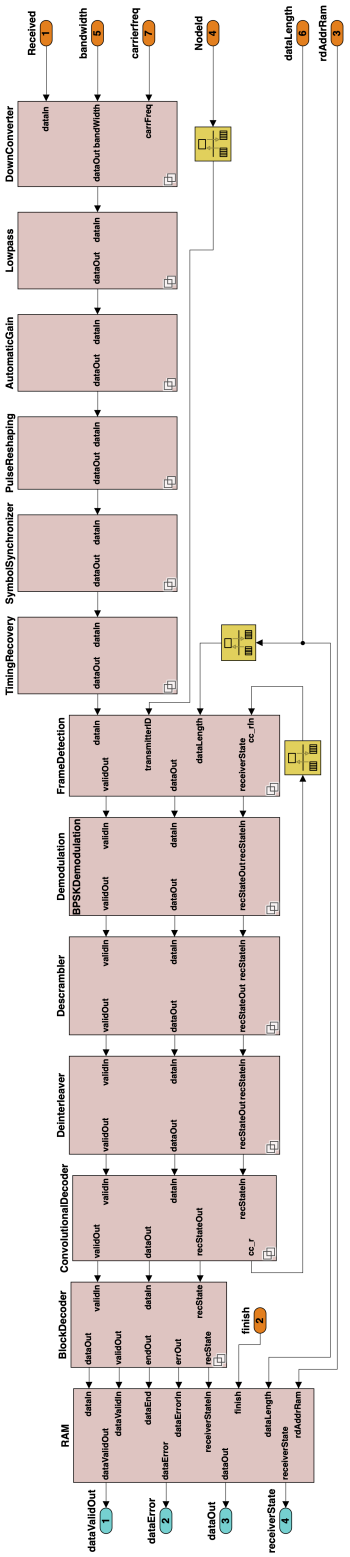
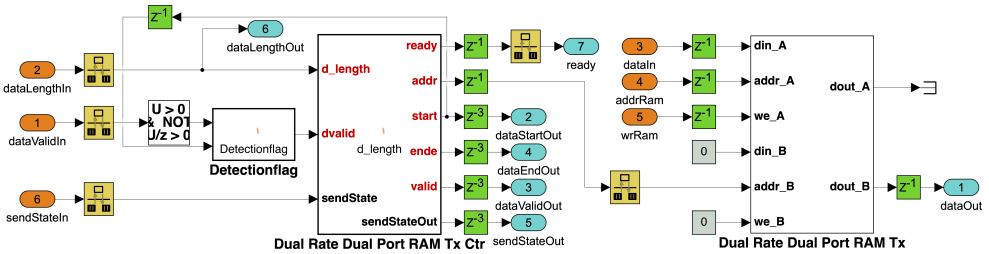


Abbildung 5.13: Simulink-Modell des Empfänger-Moduls

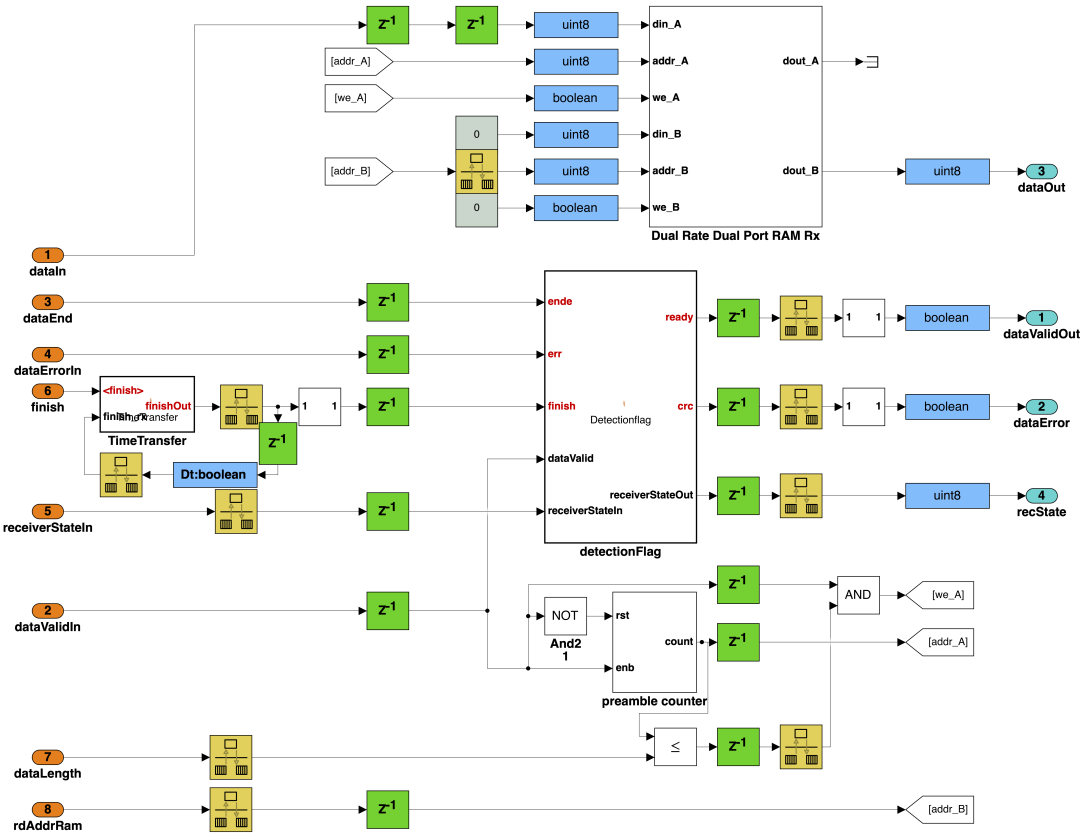
Die in Abbildung 5.14a dargestellte Komponente *RamTx* dient als Schnittstelle zwischen der nächsten Schicht,dem Bus oder Speicher und dem senderseitigen Kommunikationssystem. Die Daten werden dazu byteweise übergeben, gespeichert und nach dem Abschluss der Übertragung der übergeordneten Schicht mit der Datenrate des Sendesystems übertragen und weitergeleitet. Das Trennen von Kommunikationsschichten mithilfe eines Random-Access Memory (RAM) ist ein gängiger Ansatz in der Kommunikationstechnik, um die Effizienz und Modularität der Datenverarbeitung und -übertragung zu steigern. Der RAM dient hier als Puffer zu drüber liegenden Schicht des Kommunikationssystems. Damit können die Daten asynchron gesendet oder empfangen werden, d. h. Daten, die in einer Schicht verarbeitet werden, können zwischengespeichert werden, bis die physikalische Schicht bereit ist, diese zu verarbeiten (Senden) oder die übergeordnete diese abzurufen (Empfänger). Damit werden die unterschiedlichen Geschwindigkeiten der Schichten ausgeglichen, Überlastungen durch unterschiedliche Datenraten vermieden, die Reihenfolge der Datenpakete beibehalten und somit die Konsistenz und Integrität der Daten bewahrt. Zusätzlich kann durch die spezifizierte Entkopplung die physikalische Schicht auch unabhängig von anderen Schichten entwickelt, getestet und modifiziert werden. Die Größe des Speichers ist derzeit auf eine anpassbare Größe von $255\,\mathrm{Byte}$ begrenzt, was der maximalen Größe des Datenframes in der aktuellen Systemspezifikation entspricht, die durch den RSC definiert ist.

Block(de)codierer:

Ein Blockcodierer ist ein Kanalcodierer, der Daten in Blöcke fester Größe segmentiert und auf jeden Block eine festgelegte Kodierungsoperation durch Hinzufügen von Redundanz anwendet. Die Redundanz ermöglicht es, Fehler, die während der Übertragung auftreten, zu erkennen und optional zu korrigieren. In dieser Arbeit wurden dazu beispielhaft zwei Blockcodierer umgesetzt, den CRC abgebildet in 5.15a und den RSC [199] in Abbildung 5.15b. Der Blockcodierer bildet die Informationssymbole d über die Redundanzsymbole k auf die Blockgröße von n Symbolen ab. Der Blockdecodierer hingegen prüft die eingehenden n Symbole und rekonstruiert die d Informationssymbole mit einer optionalen Fehlerkorrektur, die beim RSC auf $\lfloor \frac{k-1}{2} \rfloor$ beschränkt ist. Zusätzlich stellt der Decodierer die Informationen zur Verfügung, ob die decodierten Daten valide, also wahrscheinlich fehlerfrei sind. Der Blockcodierer wurde in dieser Arbeit so angepasst, dass die Blockgröße in Abhängigkeit der Menge von Informationssymbolen variabel genutzt werden kann. Damit zählt d zu den dynamischen (Definition 5.1) und k zu den statischen Parametern. Für die Validierung der eingehenden Datenpakete auf physikalischer Ebene ist einer der beiden Blockcodierer nötig, um bereits auf der physikalischen Ebene fehlerhafte Pakete zu verwerfen. Bei der Verlagerung der Validierung auf die folgende



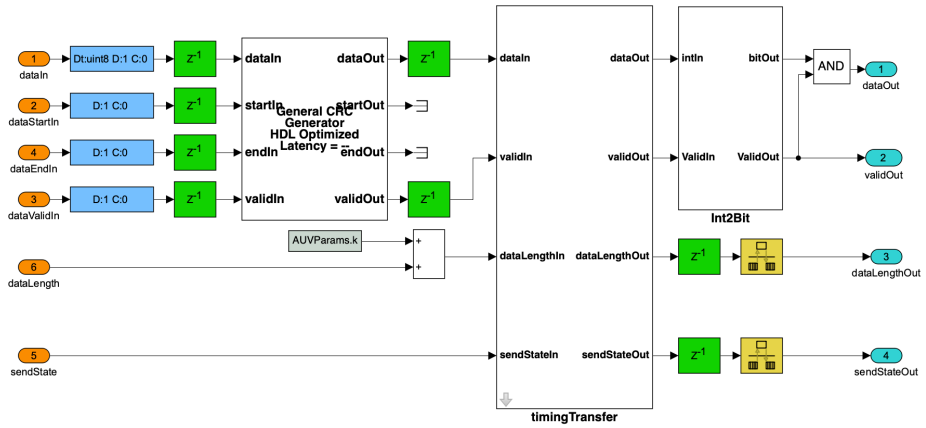
(a) Sender



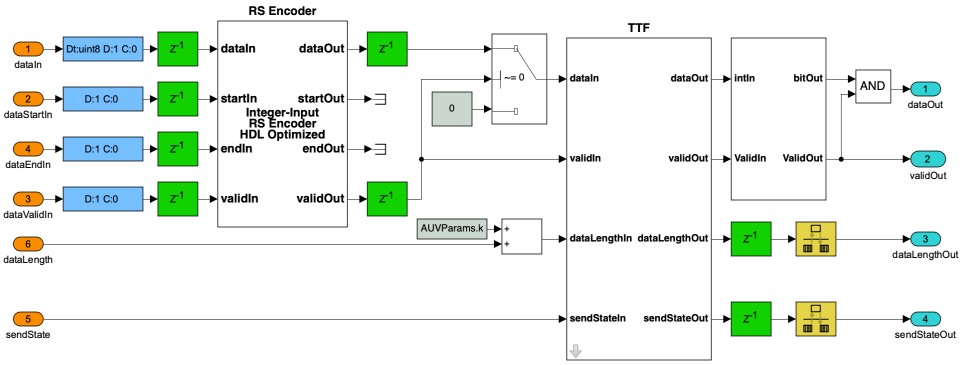
(b) Empfänger

Abbildung 5.14: Simulink-Modell der RAM-Schnittstelle

Schicht und dem Verzicht auf die Korrektureigenschaften kann auch eine Variante ohne Codierung eingeführt werden, die lediglich den Transfer von Symbolen in Bitfolgen und umgekehrt umsetzt und zeitlich anpassen muss.



(a) Simulink-Modell des CRC



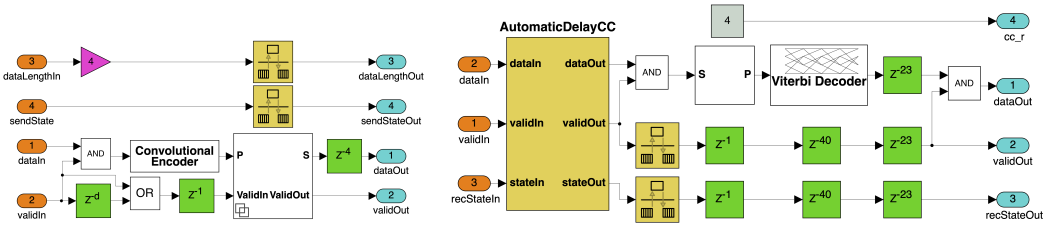
(b) Simulink-Modell des RSC

Abbildung 5.15: Simulink-Modell der Blockcodierer

Faltungs(de)codierer:

Ein Faltungscodierer wurde bereits in Beispiel 5.1 eingeführt und in den Abbildung 5.16b für die Rate $R_{CC}=0.25$ dargestellt. Der Faltungscodierer konnte im wesentlichen mit Basisblöcken in Simulink umgesetzt werden. Einzig die zeitliche Anpassung der Daten- und Kontrollströme, sowie die Synthese-spezifischen Verzögerungen hängen von der jeweiligen Umsetzung und Codierrate ab. Zusätzlich verfügt die umgesetzte Komponente um eine

Variante, die auf den optionalen Faltungscodierer verzichtet und das Signal durchschleift.



(a) Faltungscodierer mit Rate $R_{CC}=0.25$

(b) Viterbi-Decodierer mit Rate $R_{CC}=0.25$

Abbildung 5.16: Simulink-Modell des Faltungs(de)codierer der Rate $R_{CC}=0.25$

Scrambler:

Ein Scrambler ist eine Vorrichtung, die Datenbits so modifiziert, dass lange Folgen gleicher Bits vermieden werden, um die Gleichstromkomponente zu reduzieren und die Bitmuster besser an die Anforderungen der Übertragungskanäle anzupassen.

Formelle Darstellung: Sei $\mathbf{x}=(x_1,x_2,\ldots,x_n)$ die Eingangsbitfolge. Ein Scrambler erzeugt eine Ausgangsfolge $\mathbf{y}$, die durch eine polynomiale Feedback-Funktion $f(\cdot)$ modifiziert wurde:

$$y_i=x_i\oplus f(x_{i-1},x_{i-2},\ldots,x_{i-m})$$

wobei $\oplus$ die XOR-Operation und m die Länge des Feedback-Registers abbildet.

(De)Interleaver:

Der Interleaver ändert im Idealfall die Reihenfolge der Information, ohne den Informationsgehalt oder die Länge dieser zu ändern. Die Nutzung ermöglicht es auftretende Burst-Fehler über größere Zeitspannen zu verteilen. Dies hat sich insbesondere bei der Nutzung von Faltungscodierern etabliert [207–209]. In dieser Arbeit werden zwei grundsätzliche Typen von Interleavern genutzt, dem Matrix-Interleaver und dem Multiplex-Interleaver, dargestellt in Abbildung 5.18 und Abbildung 5.19a, sowie der Möglichkeit, keinen zu nutzen.

Der Matrix-Interleaver [210, 211] verwendet für die Permutation der zu sendenden Signale eine Matrixstruktur, um zusammenhängende Signalströme aufzuspalten und über einen Bereich definiert zu verteilen. Sowohl Empfänger wie Sender müssen dazu über die gleiche Matrixgröße mit R Zeilen und C Spalten, in Gl. (5.17) beschrieben, verfügen. Die Dimensionierung der Matrix hängt von der

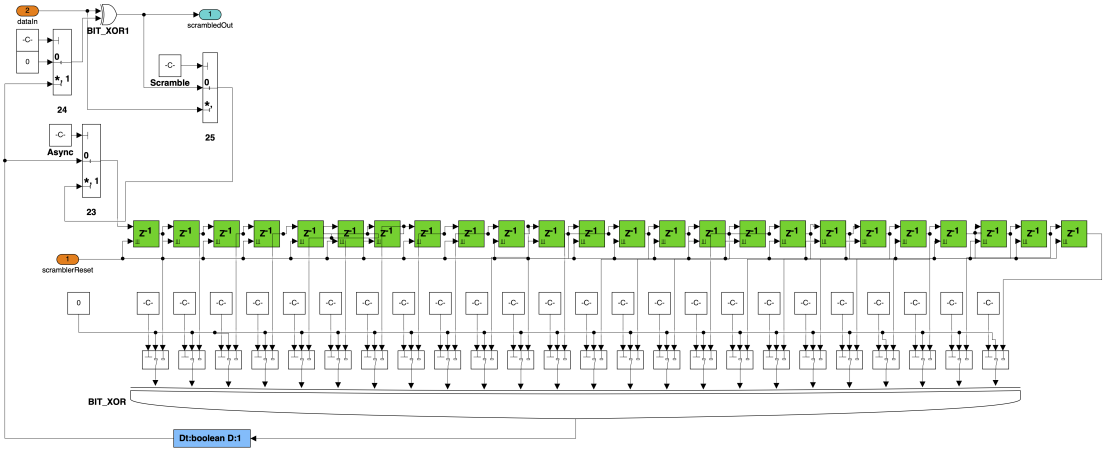


Abbildung 5.17: Simulink-Modell des Scramblers

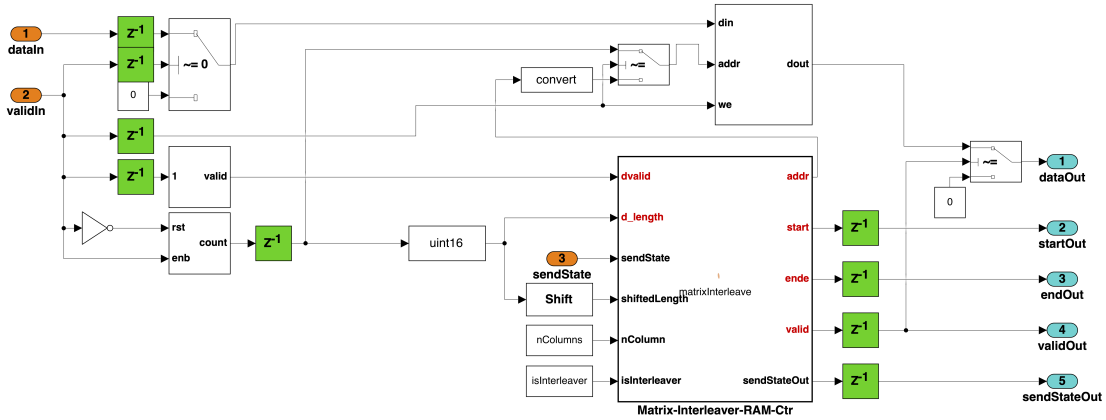
Größe der Informationen und der Verwendung ab. Die grundlegende Definition des Matrix-Interleavers sieht vor, dass die Anzahl der Informationssignale der Multiplikation von R und C entsprechen muss. Damit kann das komplette Datenpaket in der Matrix abgelegt werden. Dies vereinfacht die Nutzung und Ansteuerung erheblich, allerdings auf Kosten der Veränderlichkeit, was die Datenlänge oder Redundanz betrifft. Um den Matrix-Interleaver zu nutzen und die Veränderlichkeit der Datenlänge als dynamischen Patrameter aufrechtzuerhalten, wurde in dieser Arbeit der Matrix-(De)Interleaver mittels eines RAM und die Ansteuerung über eine adaptive datenlängenabhängige Permutationsfunktion umgesetzt. Die Größe des RAMs entspricht dazu der maximal möglichen Datenlänge der Systemspezifikation und lädt und entlädt die Daten gemäß der berechneten Permutation. Durch die Festlegung der Spaltengröße auf $N{=}8$, was der Symbolgröße in Bit entspricht, kommt es zu keinerlei Unter- oder Überbesetzung der so definierten virtuellen Matrix.

Damit kann die gleiche Funktionalität bei variabler Definition von Redundanz und Nutzdatenlänge erreicht werden, und eröffnet die Nutzung adaptiver Kommunikationsmodelle. Während der Sender die virtuelle Matrix zeilenweise füllt und spaltenweise, beschrieben über die Permutationsfunktion Gl. (5.18), ausgibt, füllt der Empfänger spaltenweise und liest zeilenweise die Signale aus. Der Empfänger rekonstruiert abschließend die Ursprungsreihenfolge der Informationssignale, über die inverse Permutationsfunktion in Gl. (5.19) über den Index i der $R{\cdot}C$ empfangenen Signale.

$$\mathbf{X}=\begin{bmatrix} x[1] & x[2] & \cdots & x[C] \\ x[C+1] & x[C+2] & \cdots & x[2C] \\ \vdots & \vdots & \ddots & \vdots \\ x[(R-1)\cdot C+1] & x[(R-1)\cdot C+2] & \cdots & x[CR] \end{bmatrix} \tag{5.17}$$

$$\pi(i)=(i \mod R)\cdot C+\left\lfloor \frac{i}{R} \right\rfloor +1 \tag{5.18}$$

$$\pi^{-1}(i)=(i \mod C)\cdot R+\left\lfloor \frac{i}{C} \right\rfloor \tag{5.19}$$



(a) RAM-basierte Umsetzung

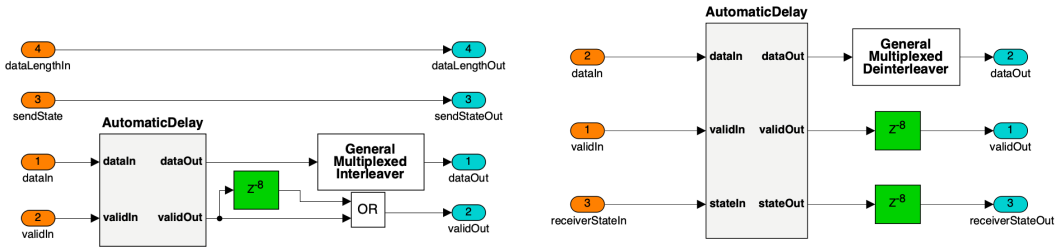
Abbildung 5.18: Simulink-Modell des Matrix-Interleavers

Der Multiplex-Interleaver verteilt die Eingabedaten hingegen auf mehrere parallele Kanäle und permutiert die Signale durch zyklisches Multiplexen. Kombiniert, wie in dieser Arbeit, mit einer vorgeschalteten Verzögerung wird aus dem Multiplex-Interleaver ein sogenannter Convolutional-Interleaver [212, 213]. Dieser verzögert die einzelnen Kanäle gemäß einer vorgegebenen Verzögerungs-Matrix. An dieser Stelle verstößt der Interleaver aber gegen die Beibehaltung der Informationslänge, welche um die Anzahl der Kanäle multipliziert, mit der maximalen Verzögerung definiert ist. Diese zusätzlichen Informationen bestehen aber bei einer mit Nullen vorinitialisierten Matrix aus eben diesen Nullen. Die Verlängerung muss damit entsprechend bei der Übertragung, Verarbeitung und Rekonstruktion berücksichtigt werden muss.

Durch die zyklische Multiplexerfunktion müssen Interleaver und Deinterleaver zeitlich abgestimmt werden. Daher verfügt der Deinterleaver, abgebildet in

Abbildung 5.19b, über eine adaptive Verzögerung der Steuersignale und eine eingangsseitige Verzögerung, welche das Signal, so verzögert, dass immer mit dem ersten Register wie im Sender begonnen wird. Der Multiplex-Interleaver bietet eine ganze Reihe von Möglichkeiten [210, 211, 214], die in dieser Arbeit, aufgrund des bereits hohen Explorationsraums, auf die zwei eingeführten Interleaver beschränkt ist, aber zukünftig angepasst werden kann.

Des Weiteren bietet der Matrix-Interleaver durch eine nahezu identische Struktur im Empfänger und Sender, welche sich nur durch die Ansteuerung des RAM über das Steuersignal *isInterleaver* unterscheidet, Optimierungspotenzial für die gleichzeitige Nutzung von Sender und Empfänger oder einer superskalaren Struktur.



(a) Multiplex-Interleaver **(b)** Multiplex-Deinterleaver

Abbildung 5.19: Simulink-Modell des Multiplex-(De)Interleaver

(De)Modulation:

Modulation und Demodulation sind grundlegende Komponenten in der digitalen Kommunikation, deren Funktion die Umwandlung von binären Datenfolgen in den benötigten Symbolraum der jeweiligen Übertragungsart und umgekehrt ist. In dieser Arbeit wurden beispielhaft die Modulationsarten Binary Phase Shift Keying (BPSK), QPSK, FSK und Frequenzsprungverfahren kombiniert mit Frequenzumtastung (FH-FSK) umgesetzt. Beim BPSK wird die Information in der Phase des Trägersignals ($0°$ und $180°$) übertragen. Dies kann erreicht werden durch die Umwandelung der binären Zustände $x \in \{0,1\}$ in den bipolaren Raum mit den Symbolen $f[x] \in \{-1,1\}$ über die Gl. (5.20) und die Umkehrfunktion des Demodulators über die Gl. (5.21).

$$f[x] = 2 \cdot x - 1 \text{ mit } x \in \{0,1\} \text{ und } f[x] \in \{-1,1\} \tag{5.20}$$

$$f^{-1}(y) = \frac{y+1}{2} \text{ mit } yin\{-1,1\} \text{ und } f^{-1}(y) \in \{0,1\} \tag{5.21}$$

Die QPSK-Modulation im Basisband wandelt zwei binären Daten $n \in \{00,01,10,11\}$ direkt in komplexe Symbole $f[n] \in \{\frac{1}{\sqrt{2}} + i\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} + i\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} - i\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} - i\frac{1}{\sqrt{2}}\}$ über

die Gleichung Gl. (5.22) um, die für die Übertragung im Basisband geeignet sind. Die Rückformung über den Demodulator ist entsprechend umgekehrt definiert:

$$f[n]=\begin{cases} \frac{1}{\sqrt{2}}+i\frac{1}{\sqrt{2}} & \text{für n } =00 \\ -\frac{1}{\sqrt{2}}+i\frac{1}{\sqrt{2}} & \text{für n } =01 \\ -\frac{1}{\sqrt{2}}-i\frac{1}{\sqrt{2}} & \text{für n } =11 \\ \frac{1}{\sqrt{2}}-i\frac{1}{\sqrt{2}} & \text{für n } =10 \end{cases} \tag{5.22}$$

Bei der FSK- Modulation wird im Up-Converter die Frequenz des Trägersignals in Abhängigkeit von den binären Daten gewählt und benötigt keine zusätzliche Umformung. Die Demodulation bedient sich hingegen derselben Umkehrfunktion wie bei BPSK, da die empfangenen Daten für eine bessere Detektion der Präambel im bipolaren Raum verarbeitet werden.

Datenframer und Rahmendetektion:

Der Datenframer erhält in der derzeitigen Implementierung die erzeugten Kasami-basierten Präambeln für die verfügbaren Adressen und Datenrahmentypen in Form eines Vorratsspeichers und basiert wesentlich auf dem Kapitel 4.

Abhängig von Status, der angibt welcher Datenrahmentyp und welche Zieladresse angesprochen werden soll, wird der Nutzdatenrahmen um die entsprechende Länge der Präambel und dem Schutzabstand verzögert und gemeinsam versendet. Der Sender verfügt dazu nur über einen Datenframer für Komplexe (QPSK) und eine Variante für alle anderen Übertragungen.

Der Empfänger hingegen verfügt für die Rahmendetektion im bipolaren Bereich über verschiedene Implementierungen, die sich in Ihrer Struktur und insbesondere in Ihrem Ressourcenbedarf unterscheiden. Grundsätzlich basieren aber alle Varianten auf der Korrelation des empfangenen Signals im Zeitbereich mit der eigenen Zielpräambel und der Korrelation mit den vier Präambeln, welche die Datenrahmentypen repräsentieren.

Bei einer erfolgreichen Detektion der Adresse und nachfolgend einem validen Datenrahmen wird das Signal weitergeleitet und in der Zeit und Phase synchronisiert. Zusätzlich sind die Varianten in der Lage Rahmentypen anderer Empfänger zu erkennen und die Informationen über den Status weiterzugeben. Zudem kann über die Varianten eine Kollisionserkennung von Rahmentypen erfolgen.

Pulseshaping:

Neben einer Variante, die das Signal nach einer spezifizierten Vervielfachung der Samplingrate nur durchschleift, verfügt die Komponente auch über eine Impulsformung über Finite Impulse Response (FIR)-Filter. Die Impulsformung

im digitalen Bereich der Kommunikationstechnik dient der Anpassung des zu übertragenden Signals für eine verbesserte spektrale Effizienz, der Reduzierung von Interferenzen und der Optimierung von Übertragungseigenschaften.

Durch die Gestaltung der Impulsform können unerwünschte Frequenzkomponenten im Spektrum, verursacht durch die Sprünge im Basisband, reduziert und damit mehr Informationen in einem begrenzten Frequenzband übertragen werden. Was im Bereich der Unterwasserkommunikation von besonderem Interesse ist. Zudem verringert die gezielte Formung der Sendesignale auftretende Interferenzen mit benachbarten Frequenzbereichen und den nachfolgenden Sendesignalen und erhöht die Robustheit gegenüber dem Rauschen und Verzerrungen.

Neben der Umsetzung mittels Raised-Cosine-Filter für die Impulsformung, definiert in Gl. (5.23), hat sich besonders im Bereich der Bandbegrenzten Kommunikation mit starkem Rauschen und stetiger Änderung des Kanals der Root-Raised-Cosine-Filter mittels Umsetzung als FIR-Filter in Gl. (5.24) und dessen Gegenpart im Empfänger etabliert [89, 215, 216].

$$h(t)=\begin{cases}\frac{1}{T}\cdot\frac{\sin(\pi t/T)}{\pi t/T}, & \text{für } |t|\leq T\\ 0, & \text{sonst}\end{cases} \tag{5.23}$$

$$h(t)=\begin{cases}\frac{1}{\sqrt{T}}\cdot\frac{\sin(\pi t/T)}{\pi t/T}, & \text{für } |t|\leq T\\ 0, & \text{sonst}\end{cases} \tag{5.24}$$

Die Voraussetzung für den Umformer ist, dass das gesendete Signal nahezu identisch am Empfänger rekonstruiert werden kann. Wird der Filter des Pulseshapers im Senders mit $h_{\text{tx}}(t)$ und der Filter des Pulsreshapers mit $h_{\text{rx}}(t)$ definiert, muss gelten, dass das Produkt dieser beiden Filterantworten die Impulsantwort eines Root-Raised-Cosine-Filters $h_{\text{rc}}(t)$ (Gl. (5.25)) ergibt. Dies wird am einfachsten erreicht, in dem Sender und Empfänger über identische Root-Raised-Cosine-Filters verfügen.

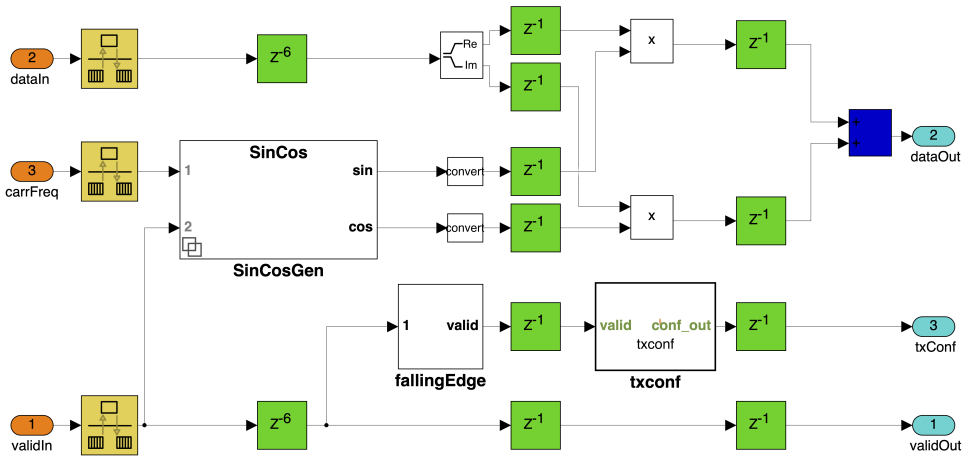
$$h_{\text{rc}}(t)=h_{\text{tx}}(t)*h_{\text{rx}}(t) \text{ mit } h_{\text{tx}}(t)=h_{\text{rx}}(t)=h_{\text{rc}}(t) \tag{5.25}$$

Auch der Pulseshaper ermöglicht, durch Filterlängen und Filterkoeffizienten die Kommunikation zukünftig weiter zu optimieren. In dieser Arbeit werden die Filter aber über eine feststehende Spezifikation umgesetzt.

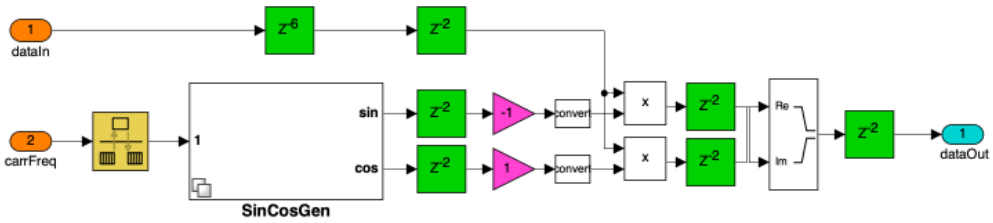
Up- und Down-Converter:

Das beispielhafte Kommunikationsmodell in dieser Arbeit umfasst zwei Arten von Signalübertagungen, dem PSK und dem FH-FSK. Ein Up- bzw. Down-Converter transferiert bei PSK-Signalen [217] das Basisbandsignal in ein höherfrequentes

Signal dem Passband um, das für die Übertragung geeignet oder genutzt werden soll, durch Multiplikation mit einem Trägersignal. Bei der Variante mit QPSK werden orthogonale Trägersignale derselben Frequenz genutzt und anschließend addiert übertragen. Die Konvertierung ist am Beispiel des Up- und Down-Converters für QPSK in Abbildungen 5.20a and 5.20b dargestellt. Die Umsetzung für BPSK ist identisch, wobei nur der reale Part der Konvertierung benötigt wird.



(a) Up-Converter

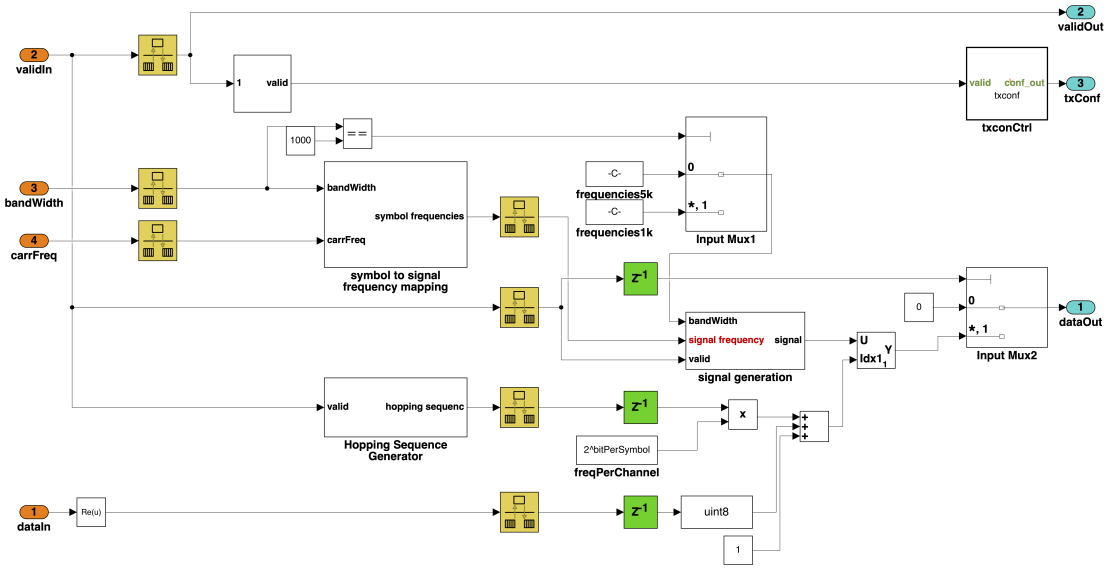


(b) Down-Converter

Abbildung 5.20: Simulink-Modell des Up- und Down-Converters für QPSK

Die Konvertierung für FSK hingegen gestaltet sich komplexer, insbesondere bei der Nutzung von mehreren Kanälen und damit der Nutzung von FH-FSK [218, 219]. Der Up-Converter in Abbildung 5.21a zeigt eine mehrstufige Umsetzung, welche die Frequenzen, Sprungsequenzen des Hoppings und der Trägersignalgenerierung und Selektion für die Übertragung aufzeigt. Der Down-Converter bildet das entsprechende Gegenstück am Empfänger, mit dem

Unterschied, dass hier das empfange Signal mit allen definierten Frequenzen multipliziert, über die Symbollänge integriert und weitergeleitet wird.



(a) Up-Converter

Abbildung 5.21: Simulink-Modell des Up-Converters für FH-FSK

Tiefpass:

Während der Ansatz mittels FSK die Energie in den Frequenzbändern betrachtet und durch einen Bandfilter unterstützt werden kann, benötigt der Ansatz mittels Phasen einen Tiefpass, um unerwünschte Frequenzkomponenten zu entfernen und die Phasenwechsel im Basisbandsignal zu extrahieren. Dies ist notwendig, da durch die Multiplikation unerwünschte Oberwellen und Mischprodukte, die außerhalb des gewünschten Frequenzbereichs liegen, entstehen. Der Tiefpassfilter entfernt diese hochfrequenten Komponenten und lässt nur die Signalkomponenten im benötigten Basisband passieren. Der Tiefpass wurde mittels drei Varianten umgesetzt, zum einen das Durchschleifen und Reduzieren der Sample-Rate des Signals, einen FIR-Filter und einen Tiefpass in Form einer diskreten Integration.

5.4.2 Verifikation

Die Verifikation der erstellten Simulink-Modelle ist ein wesentlicher Aspekt für die modellbasierte Entwicklung des akustischen Unterwasserkommunikations-

systems. Durch die Einhaltung der Spezifikationen können nicht nur potenzielle Fehler frühzeitig erkannt und behoben werden, sondern auch sichergestellt werden, dass die Simulation in Software und Hardware numerisch und zeitlich identisch ist und somit austauschbar ist.

In diesem Kontext werden Unit-Tests genutzt, da sie eine systematische Methode zur Überprüfung der Modellfunktionalität bieten. Diese Tests werden basierend auf den erzeugten Referenzfunktionen der Varianten in MATLAB durchgeführt, um die Korrektheit des Modells zu gewährleisten.

Die systematische Testung der Modelle auf Varianten-Ebene ermöglicht es, die Test-Komplexität der erzeugbaren Kommunikationsmodelle zu vermeiden, und fördert damit eine schnelle Fehlerfindung und -behebung. Durch die Integration von Unit-Tests in den Verifizierungsprozess mittels Continuous Integration (CI) kann sichergestellt werden, dass Änderungen an Varianten und dem Gesamtmodell keine unerwarteten Auswirkungen haben und dass die Referenzen konsistent und fehlerfrei bleiben. Die Umsetzung der Unit-Tests erfolgt mithilfe des MATLAB-Unit-Test-Frameworks. Zusätzlich bietet die Verwendung von MATLAB-Funktionen als Referenz die Möglichkeit einer Überprüfung der Modellleistung und trägt zur Validierung der abbildbaren Implementierung bei.

Für den Verifikationsprozess wurden zwei Arten von Testfällen definiert, welche die erwarteten Kombinationen (Sender, Empfänger) und Parameterbereiche des Modells für eine schnelle und für eine ausführliche Verifikation abdecken.

Jeder definierte Unit-Test vergleicht das Verhalten der Umsetzung in Simulink, der zu testenden Komponenten, für jede verfügbare und definierte Variante auf Basis eines Referenzmodells mit dazugehörigen MATLAB-Referenzfunktionen basierend auf den definierten Eingabeparametern für Fließkommazahlen und Festkommazahlen (optional mit definierten Toleranzen). Für die test- oder analyseabhängige Auswahl der Zahlenbereiche wurden eigene Datenformate definiert und die Umsetzungen so konzipiert, dass eine Simulation des Testmodells, der Referenzmodelle und Referenzfunktionen variabel mit beiden Zahlenbereichen (Gleit- und Festkommazahlen) möglich ist.

Die Referenzfunktionen wurden in Simulink so umgesetzt, dass die Modelle eine zeitlich und numerisch identische Funktionalität aufweisen, jedoch unabhängig voneinander genutzt werden können. Durch den Vergleich der Ausgaben beider Implementierungen können potenzielle Abweichungen und Fehler in den Umsetzungen identifiziert werden. Nach einer erfolgreichen separaten Testung der einzelnen Varianten der jeweiligen Komponente im Empfänger und im Sender wird jede Variante der Komponente des Senders mit den möglichen Varianten

des Empfängers (Umkehrfunktion), ebenfalls für Fließ- und Festkommazahlen, getestet.

Optional, in Abhängigkeit des genutzten Betriebssystems und der verfügbaren Software, wird, in dieser Arbeit automatisiert, aus dem Simulink-Modell HDL-Code generiert und ein sogenanntes *Validierungs-Modell*, welches die zeitlichen Änderungen bei der Umsetzung in HDL beinhaltet erstellt werden. Das Validierungs-Modell dient zur Überprüfung ob die zeitliche Anpassung an die HDL-Umsetzung übertragen auf das Original-Modell zu einem Fehlverhalten oder einer zeitlichen bzw. numerischen Abweichung von Original-Modell führt. Damit entspricht das generierte *Validierungs-Modell* einem Verifkations-Modell

Des Weiteren wird ein FPGA in the Loop (FIL)-Modell, welches das extern synthetisierte Modell auf der Hardware laufen lässt und gegen das Original-Modell testet, und ein Raw-File zur Programmierung eines FPGAs erstellt. Die erstellten Unit-Tests vergleichen dazu das *Validierungs-Modell* mit den Referenzfunktionen und simulieren abschließend das synthetisierte Modell auf dem FPGA und verifizieren dieses mittels des FIL-Modells. Durch die Verifikation des Original-Modells mit den Referenzfunktionen wird sichergestellt, dass die Umsetzung in HDL damit numerisch und zeitlich identisch zur Referenzfunktion ist.

Die Verwendung von Unit-Tests in Kombination mit MATLAB-Referenzfunktionen, Referenzmodellen und CI ermöglicht eine umfassende Überprüfung der Modellintegrität, die Identifizierung von Abweichungen und Fehlern und das automatisierte Erstellen von Testberichten.

5.4.3 Referenzfunktionen und -modelle

Die Referenzfunktion stellt eine mathematische oder algorithmische Darstellung eines Kommunikationssystems dar und bildet die Grundlage für Vergleiche, Tests und Validierungen der synthetisierten modellbasierten Implementierung auf Varianten-Ebene mithilfe der Referenzmodelle. Das Referenzmodell wird gemäß den Spezifikationen des zu entwickelnden Systems erstellt. Dies kann durch mathematische Modellierung, Implementierung neuer oder bekannter Algorithmen oder, falls verfügbar, durch die Verwendung vorhandener Bibliotheken erfolgen.

Auf Basis der Referenzfunktion können Unit-Tests erstellt und automatisiert werden. Diese Unit-Tests sind Tests, die sicherstellen, dass die Implementierung des modellbasierten Systems sich wie spezifiziert verhält. Sie prüfen das

Verhalten des Referenzmodells in Simulink gegenüber den definierten Testfällen unter Verwendung der Referenzfunktionen.

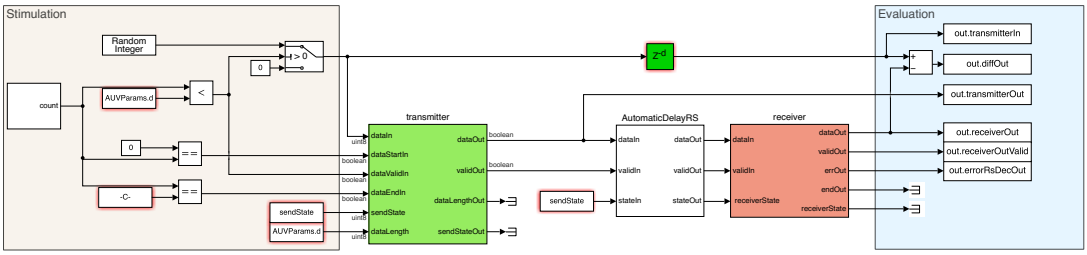


Abbildung 5.22: Referenz-Modell für den Block-(De-)Codierer in Simulink

Für jede definierte Komponente wurden Referenzfunktionen, Referenzmodelle in Simulink und Unit-Tests erstellt, welche die Varianten aus der erstellten und vorgehaltenen Bibliothek referenzieren und abbilden. In jedem Referenzmodell werden die definierten Varianten der Komponente separat sowie die Funktionspaare (mögliche Kombinationen von Umkehrfunktionen) für einen vordefinierten Parameterraum und Zahlenraumbeschränkungen (Fließ- und Festkommazahlen) getestet. Die Abbildung 5.22 zeigt das beispielhaft für das Referenzmodell des Block-Codierers.

Das Referenzmodell besteht, wie auch das Test-Modell, aus der Stimulation, der ausgewählten Variante der Komponente des Senders, einer Verzögerung, der ausgewählten Variante des Empfängers und einem Bereich zur Auswertung. Die Stimulation wird maßgeblich durch die Testparameter der Unit-Tests (Tabelle 5.1) definiert. Die Evaluierung erfolgt über die gespeicherten Signal- bzw. Datenströme der Sender- und Empfängervariante im Testskript der Unit-Tests selbst. Für die Ermittlung der Verzögerung wird das Referenzmodell zweimal simuliert. In der ersten Simulation wird die Verzögerung über die Korrelation der Signalströme bestimmt und in der zweiten Simulation wird unter Verwendung der Verzögerung das Referenzmodell ausgewertet. Für ein generisches Testskript wurden die referenzierten Varianten des Empfängers als *receiver* und für den Sender als *transmitter* in den Unit-Tests und den Referenzmodellen benannt. Eine beispielhafte Anwendung am Beispiel der Komponente des Block-Kodierers ist in Beispiel A.1 beschrieben.

Die Auswertung erfolgt für alle Varianten und Varianten-Kombinationen identisch. In der ersten Phase wird das Referenzmodell, wie bereits eingeführt, zweimal simuliert mit den Parametern durch den aktuellen Unit-Test definiert. Der Test gilt als erfolgreich, wenn die Kombination von Sender und Empfänger die Gl. (5.8) bis 5.10 erfüllt. Die zweite Testphase überprüft unter Verwendung der gleichen Eingangssignale, ob sowohl der Sender als auch der Empfänger identische oder

ggf. im Toleranzbereich liegende Ausgaben erzeugen. Wird die Referenzfunktion mit f_{ref} definiert, muss gelten:

$$f(D,P)==f_{ref}(D,P) \text{ und } f^{-1}(f(D,P),P)==f_{ref}^{-1}(f(D,P),P)$$

woraus folgt, dass die Umsetzung der Variante das gleiche numerische und zeitliche Verhalten bezüglich der Signalverarbeitung aufweist.

Optional wird unter Verwendung der Referenzmodelle mit Festkommazahlen überprüft, ob die Varianten synthetisierbar sind und das zeitliche und numerische Verhalten, basierend auf dem generierten Validierungs-Modell, identisch oder innerhalb der definierten Toleranz zum Referenzmodell liegt. Das Validierungs-Modell erzeugt aus dem Referenzmodell eine identische Kopie, die zusätzlich zeitspezifische Anpassungen in Form von Verzögerungen bezüglich der Umsetzung in HDL enthält und kann gegen das Original getestet werden. Durch eine auf den Unit-Test-basierten Erzeugung einer Testbench des zu testenden Teilsystems kann mittels Simulatoren plattformunabhängig der HDL-Code über Co-Simulationsmodelle geprüft und verifiziert werden. Das Co-Simulationsmodell vergleicht dazu die simulierten Daten des Referenz-Modells mit den Ergebnissen derHDL-Simulation unter Verwendung identischer Eingabedaten. Abhängig vom genutzten Betriebssystem (nur Windows und Linux) kann der erzeugte HDL-Code plattformspezifisch (im Rahmen der Arbeit beschränkt auf DE10-Standard, DE115 und Cyclone V SoC Development Kit) plattformspezifisch kompiliert, transferiert und mittels automatischer Generierung des Modells als FIL, allgemeiner bekannt unter dem Begriff Hardware in the Loop (HIL), über das FIL-Modell validiert werden.

Mittels des FIL-Modells können die Varianten in Hardware simuliert und das Ergebnis gegen das Referenzmodell und somit indirekt gegen die Referenzfunktion getestet und verifiziert werden. Abschließend kann aus den erzeugten Metadaten der Synthese zusätzlich der Platz- und Energiebedarf extrahiert werden.

5.4.4 Verifikation in Zahlen

Die Qualität der Methode und der resultierenden verifizierten Komponenten und Kombinationen wird anhand der erfolgreichen Verifikation gegen die jeweilige Referenzfunktion, der erfolgreichen HDL-Codeerzeugung, sowie der erfolgreichen Synthese bewertet. Durch die Zerlegung in Komponenten kann die Zahl der Tests je nach Anforderung angepasst werden. Für die Verifikation der mit der zeitlich aufwendigen Synthese wurden nur für die Kurztest auf Windows durchgeführt.

Tabelle 5.1: Testparameter der Unit-Tests in MATLAB

Variable	Kurztest	Erweiterter Test	Beschreibung
k	4,8,32	4,8:8:64	Anzahl der redundante Symbole des Block-Codierers
d	32,64,128	14,20,11+8:8:251,251	Länge des Datenrahmen in Byte
speedup	1,8	1,2,4,8,16	Beschleunigungsfaktoren für den Block-Codierer für eine kürzere Verarbeitungszeit
sendState	5	3,4,5,6	Festgelegte Werte für die aktuell zu sendenden Datenrahmen (RTS (3), CTS (4),DATA (5), ACK (6))
BandWidth	1 kHz,5 kHz		Derzeit unterstützte und zu testende Bandbreite (keine Limitation)
CFreq	10 kHz,12.3 kHz,20 kHz		Ausgewählte Trägerfrequenzen
r	0.25,0.5	1, 0.5, 0.25	Rate des Faltungscodierers
scramble	true, false		Ansteuerung des Scramblers
async	true, false		Scramblertyp (Synchron (false), Asynchron (true))
ID_L	64,128		Längen der Präambel zur Detektion der System-ID
DT_L	64,128		Längen der Präambel zur Detektion des Datenrahmentyp
fixpoint	true,false		Festlegung des genutzten Zahlenraums (Festkommazahlen (true), Gleitkommazahlen (false))
num_channels	1,2	1,2,3,5	Anzahl der genutzten Kanäle der FSK-Modulation

Die Abbildung Abbildung 5.23 gibt einen Überblick über die jeweils (kurz, erweitert) durchgeführte Anzahl von Tests. Die Abbildung zeigt dazu die erfolgreich durchgeführten und fehlerhaften Tests. Bei der Umsetzung mit Windows inklusive der Synthese von HDL-Code und dem Testen auf dem FPGA gegen die Referenzfunktion (FIL) konnte sichergestellt werden, dass alle Implementierungen gemäß der definierten Referenzen arbeiten. Insgesamt konnten alle Varianten erfolgreich und automatisiert verifiziert und, beschränkt auf den Kurztest, auf Synthetisierbarkeit geprüft und gegen die Referenzfunktion auf der Hardware getestet werden.

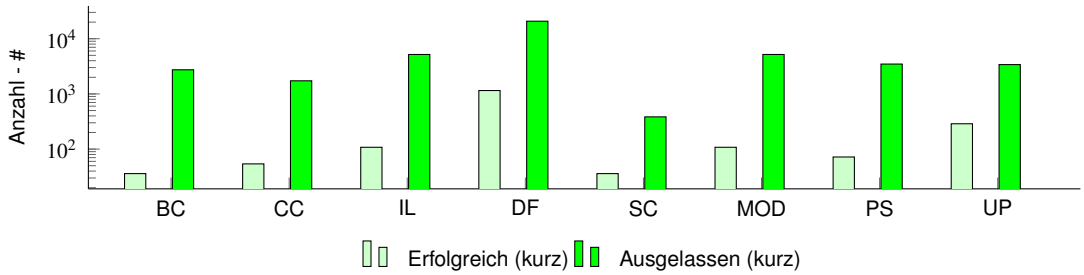


Abbildung 5.23: Auswertung der durchgeführten Unit-Tests

Die Herausforderung der Integrationstest für jede valide Kombination bleibt an dieser Stelle bestehen. Für die Verifizierung der synthetisierbaren Kombinationen von Varianten werden zwei Tests umgesetzt: Es werden zufällige valide Kombinationen von Varianten in Form von Modem-Spezifikationen erstellt, deren Umsetzungen als Kombinationen von Referenzfunktion und im Testmodell selbst für einen idealen Kanal (kein Rauschen, keine Störungen und keine Mehrwegeausbreitung) getestet werden. Für beide Tests muss für eine Verifizierung

Komponente	Variante	ALM	Register	Memory	RAM	DSP	I/O
Scrambler (Tx)	*NoneScrambler*	1	0	0	0	0	56
	GenScrambler (None)	1	0	0	0	0	56
	GenScrambler (Async)	6	15	0	0	0	56
	GenScrambler (Sync)	6	15	0	0	0	56
Scrambler (Rx)	*NoneScrambler*	1	0	0	0	0	24
	GenScrambler (None)	1	0	0	0	0	24
	GenScrambler (Async)	6	15	0	0	0	24
	GenScrambler (Sync)	6	15	0	0	0	24

Tabelle 5.2: Ressourcenbedarf des Scramblers

gelten, dass der gesendete Datenrahmen dem rekonstruierten Datenrahmen entspricht. Dem Kapitel der Optimierung vorgreifend wurde der systematische Integrationstest über das Optimierungsframework Opt4J [23] mit einem idealen Kanal (störungsfrei) umgesetzt und stichprobenartig geprüft.

5.4.5 Ressourcenanforderungen

Jede Hardware hat begrenzte Ressourcen. Das in dieser Dissertation hauptsächlich verwendete FPGA von Altera, der *Cyclone V 5CSXFC6D6F31C6*, verfügt über 77600 Adaptive Logic Modules (ALM)s und 110000 Logical Elements (LE) , die die kleinste nutzbare Einheit darstellen. Ferner verfügt das FPGA über 342 DSP-Blöcke, einen Speicher von bis zu $5662\,\mathrm{kBit}$ des Typ M9K und $5688\,\mathrm{kBit}$ des Typ M144K und 480 Ein- und Ausgabe-Pins.

Die genaue Berechnung des Bedarfs an Chip-Elementen (ALM, I/O-Pins, RAM-Blöcke, digitaler Signalprozessor (DSP)-Blöcke, Register, Memory-Bits) durch das Synthesetool erfordert einen vollständigen Technologie-spezifischen Durchlauf des gesamten Syntheseprozesses. Entgegen diesem Ansatz bietet auch MATLAB/Simulink eine Abschätzung der wahrscheinlich benötigten Ressourcen in abstrakter Form von benötigten Rechenelemente (Multiplizierern, Addier, Subtrahierer, Register, 1-Bit Register, RAMs, Multiplexer, I/O-Bits, Statische bzw. Dynamische Shift-Operationen) an. Der hardwarespezifische Bedarf der Varianten wird im folgenden parameterabhängig quantitativ aufgelistet, die Werte der benötigten Ressourcen und die Verhältnisse zwischen den Varianten unterscheiden sich dabei teils erheblich.

Der Ressourcenbedarf des Scramblers, sowohl für die Komponenten des Senders und des Empfängers kombiniert in einer optimierten Variante mit umfassender Funktionalität (Abbildung 5.17), fällt kaum ins Gewicht, mit jeweils 15 Register und 6 ALM. Die Komponenten wie die Decodierer und Rahmendetektion (Tabellen 5.3 bis 5.5) hingegen benötigen erheblich mehr

Komponente	Variante	r	ALM	Register	Memory	RAM	DSP	I/O
Faltungscodierer (Tx)	*CC*	1	1	0	0	0	0	56
		0.5	12	42	0	0	0	56
		0.25	16	49	0	0	0	56
Faltungsdecodierer (Rx)	*Viterbi*	1	1	0	0	0	0	27
		0.5	1798	4354	1088	3	0	27
		0.25	2175	4893	1152	3	0	27

Tabelle 5.3: Ressourcenbedarf des Faltungs(de)codierers

Komponente	Variante	d	k	ALM	Register	Memory	RAM	DSP	I/O
Blockcodierer (Tx)	*CRC*	32	4	106	206	2048	1	0	49
		64	4	106	206	2048	1	0	49
		128	4	106	206	2048	1	0	49
	RSC	32	16	212	370	2048	1	0	49
		64	16	212	370	2048	1	0	49
		128	16	212	370	2048	1	0	49
		32	32	346	554	2048	1	0	49
		64	32	346	554	2048	1	0	49
		128	32	346	554	2048	1	0	49
Blockdecodierer (Rx)	*CRC*	32	4	1354	911	24576	3	0	33
		64	4	1354	911	24576	3	0	33
		128	4	1354	911	24576	3	0	33
		32	16	4388	2409	24576	3	0	33
		64	16	4388	2409	24576	3	0	33
		128	16	4388	2409	24576	3	0	33
		32	32	8669	4397	24576	3	0	33
		64	32	8669	4397	24576	3	0	33
		128	32	8669	4397	24576	3	0	33

Tabelle 5.4: Ressourcenbedarf des Block(de)codierers

Ressourcen. Der Bedarf des Block(de)codierers zeigt nochmals, dass die Datenlänge, im Gegensatz zur Redundanz, in den Umsetzungen keinen Einfluss hat. Während der Registerbedarf sich nahezu proportional verhält, ist der Bedarf an Speicher konstant. Der feste Bedarf an Speicherplatz liegt an der Umsetzung der Decodierung über das Galois-Feld ($GF(2^8)$) und dem Vorhalten der dafür implementierungsspezifisch benötigten Daten.

Auch die Varianten des Datenframers im Empfänger weisen einen Parameter-spezifisch erhöhten Bedarf an Ressourcen auf. Hier ist die Differenz durch die benötigte Länge und Anzahl der Präambeln bedingt. Auch die Struktur der Varianten hat bei gleicher Funktionalität einen erheblichen Einfluss auf den Ressourcenbedarf. Dies zeigt beispielsweise die Implementierung des Datenframers als Variante mit einer Multiplikations- und Additionsfunktionen im Vergleich über die Implementierung der Korrelation mit FIR-Filtern. Die Variante basierend auf FIR-Filtern weist einen mehr als doppelt so hohen Bedarf an ALMs und mindestens einen sechsfachen Bedarf an Registern auf.

Damit lassen sich in Abhängigkeit von Geschwindigkeit und Komplexität auch unterschiedliche Strukturen implementieren die zusätzlich über die Kosten in Form

Komponente	Variante	ID	DT	ALM	Register	Memory	RAM	DSP	I/O
Datenframer (Tx)	*Kasami (Complex)*	64	64	122	79	512	3	0	40
		64	128	129	82	896	3	0	40
		128	64	136	82	640	3	0	40
		128	128	136	85	1024	3	0	40
	Kasami (Real)	64	64	108	123	321	2	0	40
		64	128	129	82	704	3	0	40
		128	64	120	126	385	2	0	40
		128	128	138	85	768	3	0	40
Datenframer (Rx)	*Kasami (Complex, V1)*	64	64	3099	1492	997	7	0	101
		64	128	4330	1748	997	7	0	101
		128	64	4251	1918	997	7	0	101
		128	128	5334	1918	997	7	0	101
	Kasami (Complex, FIR)	64	64	6317	8922	1299	9	0	101
		64	128	8902	14033	1299	9	0	101
		128	64	8981	12002	1299	9	0	101
		128	128	11547	17113	1299	9	0	101
	Kasami (Real, V1)	64	64	998	300	592	5	0	69
		64	128	1439	428	592	5	0	69
		128	64	1347	428	592	5	0	69
		128	128	1774	428	592	5	0	69
	Kasami (Real, FIR)	64	64	2236	4053	781	7	0	69
		64	128	3510	6613	781	7	0	69
		128	64	3027	5358	781	7	0	69
		128	128	4211	8378	781	7	0	69

Tabelle 5.5: Ressourcenbedarf des Datenframers

von Ressourcen- und Energiebedarf optimiert und durch die plattformspezifische Limitationen, wie die verfügbaren Ressourcen, eingeschränkt werden können.

Zusammenfassung

In diesem Kapitel wurde ein erweiterter, modellbasierter Ansatz zur Optimierung und Validierung der akustischen Unterwasserkommunikation umgesetzt und in MATLAB/Simulink realisiert. Dieser Ansatz ermöglicht es, eine Vielzahl von verschiedenen Varianten und Konzepten in ein ganzheitliches Modell zu integrieren. Am Beispiel der Unterwasserkommunikation zeigt dieses Kapitel, wie durch eine Modellierung und Formalisierung der Kommunikationstechniken eine umfassende und flexible Lösung geschaffen wird die derzeit $60\,\%$ der akustischen Kommunikationssysteme abbilden kann.

Das entwickelte Modell umfasst dazu verschiedene Kommunikationsvarianten und -konzepte, und bietet zusätzlich die Möglichkeit diese in Form von Multiplexernetzwerken und weiterer Kombinationen für die Nutzung von partielle Rekonfiguration vorzubereiten. Diese Vielfalt an Kommunikationsstrategien wird in einem einzigen erweiterbaren und anpassungsfähigen Modell zusammengefasst. Das modellbasierte Verfahren stellt sicher, dass das System flexibel

genug ist, um neue Technologien und Methoden zu integrieren, wodurch es zukunftssicher und skalierbar wird.

Ein zentraler Aspekt dieses Verfahrens ist die Synthetisierbarkeit des Modells. Dies bedeutet, dass das Modell in eine praktische Implementierung auf Hardware übertragen werden kann, was eine schnelle und effiziente Test- und Validierungsphase ermöglicht. Durch die Nutzung von Referenzfunktionen kann das Modell umfassend und effizient getestet werden, um sicherzustellen, dass es den Anforderungen der akustischen Unterwasserkommunikation gerecht wird.

Zusätzlich wird die bidirektionale Anbindung an externe Frameworks ermöglicht, indem das Modell über Austauschformate wie XML definiert werden kann. Diese flexible Modellbeschreibung erleichtert die Integration in bestehende Systeme und unterstützt die Anbindung an Optimierungsframeworks und eröffnet damit die Möglichkeit, komplexe Modelle effizient zu optimieren und die besten Lösungen für spezifische Kommunikationsanforderungen und Umgebungen zu finden.

Durch die präzise Modellierung und die Möglichkeit zur Synthese bietet dieser Ansatz einen bedeutenden Fortschritt in der Entwicklung, Auswahl und Implementierung komplexer Kommunikationssysteme. Die Integration verschiedener Kommunikationskonzepte in ein einziges, flexibles Modell ermöglicht es, neue Technologien und Methoden effizient zu testen und zu validieren, was die Innovationsrate in der Forschung und Entwicklung erheblich steigert.

Darüber hinaus erleichtert die Möglichkeit zur Anbindung an externe Optimierungsframeworks die Implementierung von komplexen, wissenschaftlich fundierten Lösungen in der Praxis. Dies bedingt eine schnellere Adaption neuer Forschungsergebnisse und Technologien in realen Anwendungen, was den technologischen Fortschritt beschleunigt und die Lücke zwischen Theorie und Praxis verringert.

Die methodische Grundlage dieses Verfahrens bietet auch eine neue Perspektive der Systemmodellierung und -synthese, die weit über die Unterwasserkommunikation hinaus anwendbar ist. Sie schafft eine Basis für interdisziplinäre Forschungsansätze, die in verschiedenen Bereichen der Ingenieurwissenschaften und darüber hinaus Anwendung finden können. Durch die Kombination von präziser Modellierung, Synthese und der Integration externer Frameworks wird eine umfassende Plattform für die Optimierung und Validierung von Kommunikationssystemen geschaffen.

6

Auswahl, Analyse und Optimierung

Die modulare multivariate modellbasierte Entwicklung und deren Spezifikation ermöglicht eine Vielzahl gültiger parametrisierbarer Kombinationen von Fehlerkorrekturverfahren, Rahmenerkennungen und Modulationsformen. Bedingt durch die Vielzahl an möglichen Kombinationen entsteht ein großer Entwurfsraum. Optimale Entwürfe sind unmöglich manuell auszuwählen und eine erschöpfende Exploration zeitlich zu aufwendig. Kombiniert mit der Tatsache, dass die simulative Auswertung von möglichen Lösungen für realitätsnahe Kanalmodelle sehr rechenintensiv ist, wird eine Design Space Exploration (DSE)-Technik für eine effizientere Mehrzieloptimierung benötigt.

In diesem Kapitel wird das Konzept zur Anbindung für ein standardisierbares Verfahren zum Zielgrößen-basierten Leistungsvergleich und zur Optimierung von Implementierungen für die modellbasierte Entwicklung am Beispiel des akustischen Passband-Modems für die Unterwasserkommunikation eingeführt. Diese anwendungs- und umgebungsspezifische Optimierung ist jedoch sehr zeitaufwendig und kostenintensiv [220, 221].

Daher beinhaltet das Kapitel neben einer allgemeinen Einführung in das Optimierungsverfahren, die Zielgrößen und den zu betrachtenden Analyseraum auch Ansätze zur Reduzierung der Simulationszeit. Dazu gehören ein hybrides Referenzmodell, Testpunktreduktion im Analyseraum [152] und eine Beschränkung des Entwurfsraums [222], deren Eigenschaften nicht den definierten Anforderungen entsprechen.

Dieses Kapitel verdeutlicht damit die Wirksamkeit und Anwendbarkeit der eingeführten Konzepte und zeigt deren positiven Effekte auf die Entwicklung und Auswahl von Kommunikationssystemen für zukünftige Entwicklungsprozesse und Standards.

6.1 Optimierungsansatz

Während der Optimierung werden definierte Umsetzungen des Modems bezüglich generisch festzulegender Zielgrößen, welche die Kommunikationsqualität erfassen, optimiert. Dazu wird im Vorfeld eine geeignete (gültige) Umsetzung eines Modems definiert und über die Kombination eines Optimierungsframeworks und Matlab/Simulink über die definierten Testpunkte analysiert und evaluiert.

Das verwendete DSE-Verfahren nutzt das Optimierungsframework Opt4J [23, 24] unter Verwendung eines metaheuristischen Evolutionary Algorithm (EA) zur Durchquerung des Entwurfsraums und der Mehrzieloptimierung. Dieser Algorithmus eignet sich besonders zur Bestimmung des Lösungsraums von Mehrzieloptimierungsproblemen durch die Fähigkeit, mehrere Lösungen parallel zu verarbeiten und zu optimieren. Zu den bekanntesten evolutionären Algorithmen für Mehrzieloptimierung gehören der in dieser Arbeit verwendete NSGA-II (Non-dominated Sorting Genetic Algorithm II) [223] und der SPEA2 (Strength Pareto Evolutionary Algorithm 2) [224]. Der NSGA-II verwendet eine nicht dominierte Sortierung und eine Crowding-Distanz-Messung und erzeugt eine diverse und gut verteilte Pareto-Front. Der NSGA-II besteht aus der Initialisierungsphase, in der eine festgelegte Startpopulation zufällig aus dem Entwurfsraum generiert oder ausgewählt wird. Diese werden über die sogenannte Fitness der jeweiligen Population, basierend auf den eval10548874862104329418rten Optimierungszielen, ausgewertet. Aus den dominierenden Populationen wird eine feste Anzahl (Eltern) ausgewählt. Basierend auf diesen dominanten Populationen werden über den Kreuzungsfaktor mittels Mutation eine festgelegte Anzahl von Kindern erzeugt. Abschließend wird eine diverse und gut verteilte Pareto-Front über eine nicht dominierte Sortierung und Crowding-Distanz erzeugt.

Die Optimierung durchläuft dazu eine automatische Erstellung des Entwurfsraums (in Matlab/Simulink) über eine allgemeine XML-Beschreibung der möglichen Modems, aus dem der Optimierer eine gültige Lösung erstellt. Die gültige Lösung wird von Matlab/Simulink eingelesen, parametrisiert, erstellt, kompiliert und über einen vorgehaltenen Analyseraum über definierte Testpunkte simuliert.

Eine erste Verifikation, mit minimalem Testset, wurde bereits in [225] veröffentlicht und zeigt, dass die aufgestellte These zur Optimierung von Kommunikationssystemen für den Einsatz im Bereich der komplexen akustischen Unterwasserkommunikation genutzt werden kann.

6.1.1 Zielgrößen

Die Ergebnisse der jeweiligen Simulation werden abschließend dem Optimierer, in dieser Arbeit als Comma-Separated Values (CSV)-Datei, zurückgegeben und verwendet, um die Kommunikationsqualität einer Umsetzung über vorgegebene Zielgrößen zu bestimmen. Die Zielgrößen sind nicht statisch definiert und können von Anwender frei festgelegt werden. Beschränkt wird die Auswahl dabei auf verfügbare oder vorgegebene Werte für die Zielgrößen, welche über die Kopfzeile der CSV-Datei definiert und auch spezifiziert werden. In der anpassbaren Konfiguration werden Kanalparameter, die den Analyseraum aufspannen, mit dem Prefix „*chan:*“angegeben. Die Zielgrößen hingegen benötigen den Prefix „*obj:*“, gefolgt von Parameternamen der jeweiligen Zielgröße. Da die Zielgrößen bei der Optimierung entweder maximiert oder minimiert werden sollen, wird dies in Form des Postfix „(MAX)“für das Maximieren und „(MIN)“für die Minimierung der Zielgröße angegeben. Bei einem Testraum mit mehr als einer Simulation, was in dieser Arbeit gegeben ist, wird bei der Auswertung und Optimierung der Durchschnitt aller Simulationswerte einer Zielgröße bestimmt.

Die erweiterbaren durchschnittlichen Zielgrößen über das Testset sind in dieser Arbeit auf die Dekodierrate (Definition 6.1), Nutzdatenraten für einen Broadcast sowie Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mit und ohne Handshake-Verfahren (beschrieben in Abschnitt 6.3.2), Decodierfehlerrate (Definition 6.2) und Bitfehlerrate (Definition 6.3) beschränkt, wobei die Nutzdatenrate damit implizit auch die Nutzdatenlänge d_l als zu maximierende Optimierungsziel mit gleicher Wertigkeit wie die Nutzdatenraten gewählt werden könnte. Die durchschnittliche Nutzdatenrate setzt sich aus allen Nutzdatenraten der simulierten Testpunkte zusammen. Für Testpunkte, in denen kein Datenrahmen erfolgreich detektiert und decodiert wurde, wird eine Nutzdatenrate von $0\,\mathrm{bit/s}$ angenommen. Die Optimierung kann über zusätzliche Optimierungsziele wie den Bedarf an Fläche, Kosten, Energieverbrauch und weitere Zielgrößen aus den Metadaten oder Expertenwissen erweitert werden.

Definition 6.1 — Durchschnittliche Dekodierrate $\overline{R_{dec}}$:
Die durchschnittliche Dekodierrate ist eine Zielgröße, die maximiert werden soll. Sie gibt das Verhältnis der durchschnittlich korrekt dekodierten Datenframes F_{dec} zur durchschnittlichen Anzahl der gesendeten Datenrahmen F_{Tx} in der physikalischen Schicht über ein Testset wieder:

$$\overline{R_{dec}} = \frac{\frac{1}{n}\sum_{i=1}^{n} F_{deci}}{\frac{1}{n}\sum_{i=1}^{n} F_{Txi}} \tag{6.1}$$

Hierbei bezeichnet i den Testpunkt und n die Gesamtanzahl der Testpunkte. Die Anzahl der korrekt dekodierten Datenrahmen ist dabei auf maximal eins pro gesendeten Datenrahmen beschränkt.

Definition 6.2 — Durchschnittliche Decodierfehlerrate $\overline{R_{err}}$:
Die zu minimierende Zielgröße der durchschnittlichen Decodierfehlerrate gibt das Verhältnis der detektierten, aber nicht erfolgreich dekodierten F_{err} zur Anzahl der gesendeten Datenrahmen F_{Tx} über das komplette Testset wieder:

$$\overline{R_{err}} = \frac{\frac{1}{n}\sum_{i=1}^{n} F_{erri}}{\frac{1}{n}\sum_{i=1}^{n} F_{Txi}} \tag{6.2}$$

Hierbei bezeichnen F_{erri} und F_{Txi} die Anzahl der nicht dekodierten und gesendeten Datenrahmen im i-ten Testpunkt der insgesamt n Testpunkte.

Definition 6.3 — Durchschnittliche Bitfehlerrate $\overline{BER}$:
Die durchschnittliche Bitfehlerrate (BER) ist ein Standard-Qualitätsmaß in der Nachrichtentechnik und quantifiziert den Anteil der Übertragungsfehler in Bits. Die zu minimierende durchschnittliche Bitfehlerrate ist definiert über die Anzahl der fehlerhaft empfangenen Bits $\#Bit_{err}$ im Verhältnis zu den insgesamt empfangen Bits $\#Bit_{rx}$ (Gl. (6.3)) über das gesamte Testset mit n Testpunkten.

$$\overline{BER} = \frac{1}{n}\sum_{i=1}^{n} \frac{\#Bit_{err}}{\#Bit_{rx}} \text{ mit } \#Bit_{err}, \#Bit_{rx} \in \mathbb{N}^* \tag{6.3}$$

6.1.2 Bewertung der Optimierung

In der Multi-Kriterien-Optimierung (MCO) steht die Identifizierung von Lösungen im Vordergrund, die (konkurrierende) Ziele gleichzeitig optimieren. Besonders in komplexen Anwendungen oder Problemen, wie bei einer DSE, müssen Entscheidungen getroffen werden, die nicht nur die Qualität einer einzelnen Lösung, sondern auch die Diversität und Effizienz des gesamten Lösungsraums berücksichtigen. Es muss also sichergestellt werden, dass der Lösungsraum

vollständig abgedeckt wird. Eine vollständige Abdeckung führt aber zu langen und unzumutbaren Rechenzeiten. Daher versucht die Heuristik die Abdeckung so zu reduzieren, dass die Qualität der Lösungen nur geringfügig verschlechtert wird. Der Hypervolumen-Indikator stellt eine Metrik dar, die nicht die Lösungen, sondern die Qualität des durchsuchten Lösungsraums der Optimierung bewertet. Ergänzt wird, der Hypervolumen-Indikator durch den ε-Indikator dessen Ziel der qualitative Vergleich von Lösungsmengen ist. Der ε-Indikator gibt damit an, wie ähnlich die gefundenen Mengen von Lösungen sind.

6.1.2.1 Hypervolumen-Indikator

Das Hypervolumen, auch als S-Metrik oder Lebesgue-Maß bekannt, spielt eine zentrale Rolle in der Bewertung von Lösungsräumen bei Multi-Kriterien-Optimierungsproblemen. Es misst das Volumen des mehrdimensionalen Raums, der von einer Menge von Lösungen im Vergleich zu einem Referenzpunkt dominiert wird. Dabei umfasst es alle Punkte im Zielraum, die durch die Lösungen dominiert werden, und bietet eine solche quantitative Maßzahl für die Abdeckung des Lösungsraums. Der Hypervolumen-Indikator (Definition 6.4) betrachtet die Differenz der Hypervolumen zueinander ohne Berücksichtigung einer Referenzlösung und gibt damit an, wie sich diese zueinander unterscheiden. Ein großer Wert deutet auf einen großen Unterschied im Hypervolumen und ein niedriger zu einem ähnlich großen Volumen des Lösungsraums hin.

Definition 6.4 — Hypervolumen-Indikator [224, 226, 227] :
Mit $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ als Menge von n Lösungen im Zielraum, wobei jede Lösung $\mathbf{x}_i=(f_1(\mathbf{x}_i),f_2(\mathbf{x}_i),\ldots,f_m(\mathbf{x}_i))$ durch m Zielfunktionen f_j definiert ist. Sei $\mathbf{r}=(r_1,r_2,\ldots,r_m)$ ein Referenzpunkt mit $r\in X$.

Das Hypervolumen HV ist dann definiert als das m-dimensionale Volumen des Raums, über das Lebesgue-Maß μ_L, der von den Lösungen in $\mathbf{X}$ und dem Referenzpunkt $\mathbf{r}$ dominiert wird:

$$HV(\mathbf{X},\mathbf{r})=\mu_L\left(\bigcup_{\mathbf{x}\in\mathbf{X}}[f_1(\mathbf{x}),r_1]\times[f_2(\mathbf{x}),r_2]\times\cdots\times[f_m(\mathbf{x}),r_m]\right) \tag{6.4}$$

Damit wird der Hypervolumen-Indikator h über das Maximum des Unterschieds im Hypervolumen zwischen Sets von Lösungen definiert:

$$h=\max_{i\neq j}|HV(\mathbf{X}_i)-HV(\mathbf{X}_j)| \tag{6.5}$$

6.1.2.2 ε-Indikator

Der ε-Indikator wird über die gefundenen Lösungsmengen bestimmt. Ein sich ergebener niedriger ε-Indikator-Wert deutet darauf hin, dass die Lösungsmenge nahe der zu vergleichenden Lösungsmenge liegt und eine gute Näherung der optimalen Pareto-Front darstellen kann. Ein großer ε-Indikator-Wert hingegen deutet darauf hin, dass sich die betrachtete Lösungsmengen hinsichtlich der Zielfunktionen erheblich schlechter sind als die Lösungen der Vergleichsmenge.

Definition 6.5 — ε-Indikator [224, 228]:
Der ε-Indikator für zwei Mengen von Lösungen A und B ist der kleinste Wert ε, für den jede Lösung in B von mindestens einer Lösung in A um den Faktor ε dominiert wird:

$$\varepsilon(A,B) = \inf_{\varepsilon \in \mathbb{R}} \{\forall \mathbf{b} \in B \exists \mathbf{a} \in A\} \tag{6.6}$$

Für ε muss für alle Zielfunktionen i gelten das $f_i(\mathbf{a}) \leq \varepsilon \cdot f_i(\mathbf{b})$ ist.

6.1.3 SeeSim-Experimente

Ausgehend von den aus der Literatur [229–234] bekannten maritimen Umgebungen *AComm10*, *Trond* und *AComm13* für Meßkampangnen und Tests auf hoher See, eingeführt in Abschnitt 3.3.1, wird der Lösungsraum für die Unterwassermodems durchsucht. Dazu wird basierend auf der Gesamtmodellbeschreibung eine vollständige Optimierung als Basis durchlaufen. Mit dieser Referenz wird die Effektivität der Reduktionsverfahren des Analyseraums nachgewiesen und bewertet.

6.1.3.1 Testdefinitionen

Die Experimente werden aufgrund der Komplexität und unterschiedlichen Anzahl der Testpunkte im Analyseraum differenziert betrachtet. Dazu wird die Durchführung des Analyse- und Optimierungsworkflows auf eine Seegangsstufe von $2\,\mathrm{bft}$ bei einer Windstärke von $2\,\mathrm{m\,s^{-1}}$ festgelegt. Die initiale Population (Eltern) beträgt 20 Lösungen. In jeder neuen Generation (beschränkt auf 25 für die Umgebung *AComm10* und 15 für *Trond*) werden 10 neue Individuen (Lösungen) über den evolutionären Ansatz mit einer Kreuzungsrate von $95\,\%$

erzeugt und ausgewertet. Das bedeutet: $95\,\%$ der erzeugten Lösungen werden über die Kreuzung und $5\,\%$ über Mutation erzeugt.

Jeder Testpunkt im Analyseraum wird für die Umgebung *AComm10*, und *Trond* mit jeweils 20 Datenframes mittels des hybriden Modells, sowie der Umgebungsrauschen definiert über Abschnitt 3.3.3 getestet und ausgewertet. Die Testpunkte wurden im Vorfeld über das in Abschnitt 3.3.2.1 eingeführte Verfahren mittels BELLHOP erzeugt und in der Nachbearbeitung um den Dopplereffekt (Abschnitt 3.2.5) erweitert.

Die Kanalimpulseantwort (CIR)-Matrix wird über einen Zeitraum von $3\,\mathrm{s}$ im Abstand von $200\,\mathrm{ms}$, einer Geschwindigkeit von Empfängern und Sendern von $1\,\mathrm{ms}^{-1}$ in horizontaler Richtung und $0.5\,\mathrm{ms}^{-1}$ in vertikaler Richtung, einer vertikalen Wellengeschwindigkeit von $0.2\,\mathrm{ms}^{-1}$ und einer horizontalen Strömungsgeschwindigkeit von $1\,\mathrm{ms}^{-1}$.

In Abhängigkeit der Geschwindigkeit des Senders und Empfängers ändert sich deren Postion im Raum und resultiert in einer entsprechend angepassten CIR-Matrix. In der ersten Testphase wird angenommen, dass der Kanal für die Dauer einer Übertragung nahezu stabil ist und sich kaum merkliche Änderungen ergeben. Zudem wird angenommen das sich Sender und Empfänger identisch bewegen.

6.1.3.2 Hybride Simulation

Für eine Beschleunigung der Simulation wurde ein hybrides Passband-Modell in MATLAB, basierend auf den Referenzfunktionen der Verifikation, mit einer identischen numerischen und zeitlich geordneten Funktionalität entworfen, wobei die Sender- und Empfängerabbildung Frame-basiert implementiert wurden. Die hybride Simulation ermöglicht, im Vergleich zur Simulation in Simulink, eine schnelle Überprüfung auf Basis des identischen Funktionsverhaltens der abgebildeten Modems.

Die Simulationszeit einer Serie von 20 Datenframes für einen Testpunkt kann im Vergleich zur Modellsimulation um bis zu $98.75\,\%$ reduziert werden [220, 221] und bietet damit einen zeitlich umsetzbaren Ansatz für die Optimierung. Damit muss bei dem eingeführten Optimierungsansatz einzig in der abschließenden Bewertung und Überprüfung sowie für die Hardwaresynthese auf das Simulink-Modell des Kommunikationssystems zurückgegriffen werden.

6.1.3.3 Bewertung der Optimierung

In der Analyse wurden die Leistungsmetriken, definiert in Abschnitt 6.1.1, optimiert. Dazu wurde die Nutzdatenrate für die Verwendung des RTS/CTS-Austauschs und Decodierrate maximiert und die BER und Fehlerrate minimiert. Die Effizienz der Optimierung abschließend über des Hypervolumens und der Epsilon-Dominanz untersucht und bewertet.

Das Ergebnis der Optimierung für *AComm10* über alle Testpunkte ist in der Abbildung 6.1 mit allen dominierenden Lösungen über alle Generationen hinweg dargestellt. Die Abbildungen zeigen dazu die Verteilung der Nutzdatenrate in Relation zur BER (Abbildung 6.1a) und die Nutzdatenrate in Relation zur Decodierrate in Abbildung 6.1c.

Die Abbildung 6.1d zeigt die Verteilung in Bezug auf die Zielgrößen der Decodierrate und der BER und die Abbildung 6.1b zeigt die Verteilung der Ergebnisse, abgebildet auf den Zielgrößen der Decodierrate und dem BER. Die Grafiken bilden dazu alle generationsweise dominierenden Lösungen der durchlaufenden Optimierung in Grau und die resultierenden dominierenden Lösungen farblich ab.

Für eine gezielte Analyse und Bewertung der Optimierung ist die normierte Distanz über alle Generationen für das Hypervolumen in Abbildung 6.1f, für die ε-Dominanz in Abbildung 6.1e und die benötigte Optimierungszeit in Abbildung 6.1g mit der Standardabweichung, abgebildet.

Die Nutzdatenrate zeigt eine weite Verteilung, was auf eine signifikante Variabilität in den zu optimierenden Kommunikationsparametern hinweist. Diese weite Streuung ist bezeichnend für Szenarien, in denen die Kommunikationsbedingungen stark fluktuieren, was auf die spezifische Umgebungsbedingung, wie dem Geschwindigkeitsprofil, und der stark geprägten Mehrwegeausbreitung zurückzuführen ist. Die weite Spreizung der Nutzdatenrate zeigt sich direkt in der Entwicklung des ε-Dominanz über die hohe Standardabweichung. Zudem weist die Verteilung der Zielgrößen darauf hin, dass breit gefächert im Lösungsraum gesucht und qualitativ hochwertige Lösungen gefunden werden.

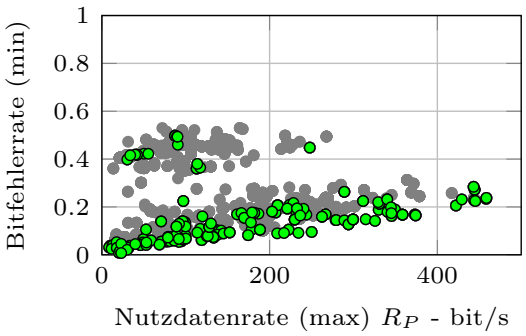
Die BER weist eine charakteristische Verteilung in zwei markanten Fronten auf, die direkt von der Decodierrate und der Fehlerrate beeinflusst wird. Diese Verteilung lässt sich darauf zurückführen, dass die jeweiligen Datenrahmen zwar erkannt (detektiert), aber nicht erfolgreich decodiert wurden. Eine geringe Diversität in den Lösungen kann über die Betrachtung der Indikatoren (Hypervolumen, ε-Dominanz) ausgeschlossen werden und weist auf eine robuste Optimierung der Kommunikationsparameter hin.

Die Decodierrate und die Nutzdatenrate zeigen eine breite Front, was zusätzlich darauf hinweist, dass verschiedene Optimierungspfade verfolgt werden. Die Spreizung zeigt, dass die Optimierung eine Vielzahl von Lösungen exploriert, um somit die besten Kompromisse zwischen Datenrate, BER und Decodierrate zu finden. Eine gut definierte und funktionierende Detektion der Datenrahmen impliziert die größtenteils niedrige Fehlerrate. Die dennoch auftretenden Fehler lassen sich auf eine starke Störung und die nicht vorhandene Sichtverbindung (LOS)-Komponente, insbesondere für große Kommunikationsdistanzen, zurückführen.

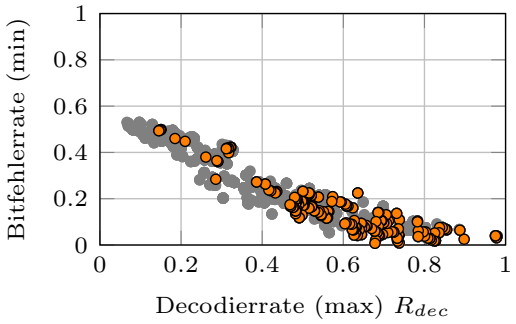
Ein globales Optimum für das Modem ist nicht bekannt, aber mit einer Decodierrate von $R_{dec}{=}0.975$ legt die Vermutung nahe das es in Anbetracht der Komplexität des Kanals für diese Zielgröße kaum noch bessere Lösungen zu geben.

Zusammenfassend zeigt die Optimierung über fünf Läufe mit jeweils 25 Generationen, dass zuverlässige Lösungen gefunden werden, die einen guten Kompromiss zwischen Datenrate, BER, Decodierrate und Fehlerrate bieten. Die Konvergenz des Hypervolumens und die rasche Abnahme der Epsilon-Dominanz sind zudem Indikatoren für die Effektivität und Robustheit der Optimierung und die Qualität der gefundenen Lösungen.

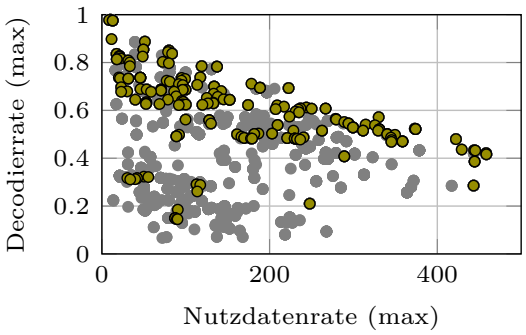
Die Abbildung 6.2 visualisiert die Optimierung für die Umgebung *Trond* begrenzt auf die Sendertiefen von $150\,\mathrm{m}$ und $300\,\mathrm{m}$ für eine Windstärke von $2\,\mathrm{m\,s^{-1}}$. Auch hier zeigt die Optimierung ihre Stärke in der Diversität und der Erzeugung qualitativ hochwertigen Lösungen, welche zeigen, dass die Optimierung erfolgreich durchlaufen wird und Lösungen gefunden werden, die ausgeglichen ausgelegt auf die vorgegebenen Zielgrößen ausgelegt sind. Aus den erhaltenen Lösungen können in Folgeschritten adaptive Strategien und weitere Optimierungen bezüglich Mehrnutzer-Netzwerke, Kosten in Form von Ressourcen- und Energiebedarf und weiterer Eigenschaften erfolgen. Damit bildet die Entwicklungs- und Optimierungsworkflow die Basis für die effektive Auswahl und Strategie einer adaptiven Netzwerkwerkkommunikation im Bereich der akustischen Unterwasserkommunikation und ähnlich komplexer Kanäle.



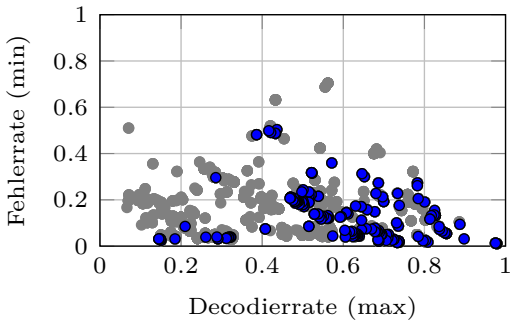
(a) Nutzdatenrate zu BER



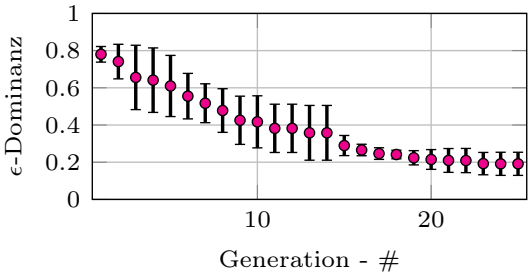
(b) Decodierrate zu BER



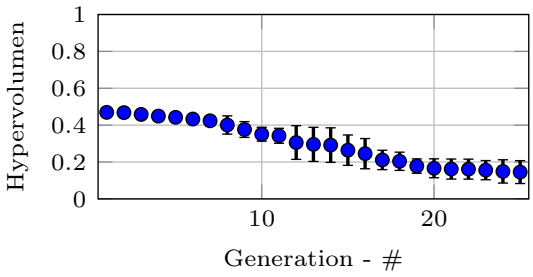
(c) Nutzdatenrate zu Decodierrate



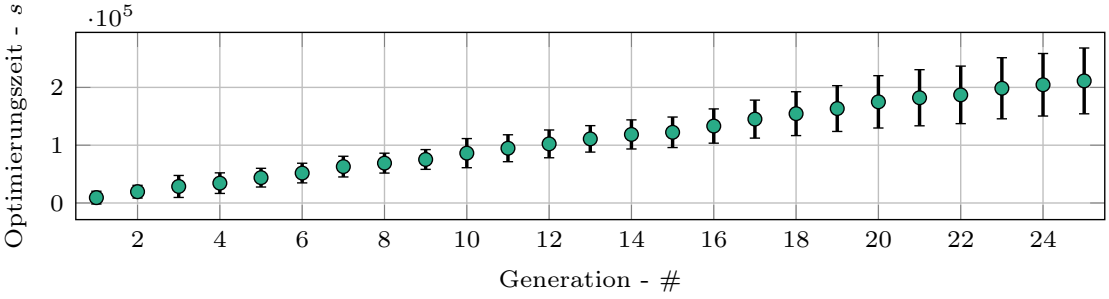
(d) Decodierrate zu Decodierfehlerrate



(e) ε-Dominanz über die Generationen

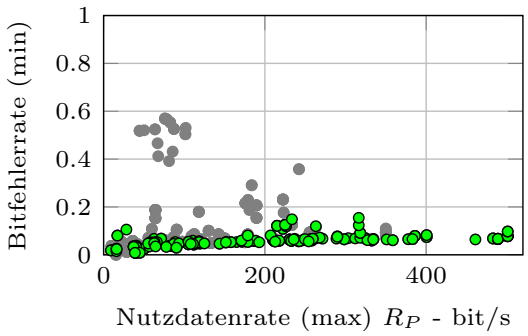


(f) Hypervolumen über die Generationen

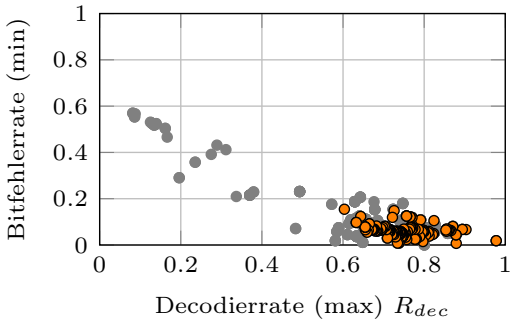


(g) Optimierungszeit

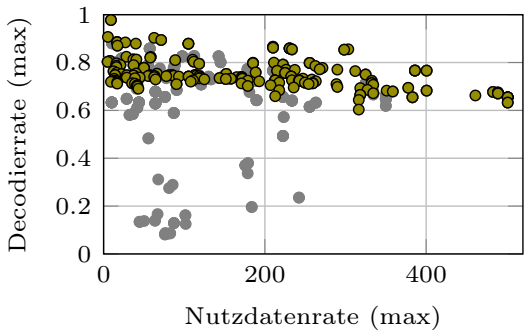
Abbildung 6.1: Ergebnis der Optimierung für die Umgebung *AComm10*



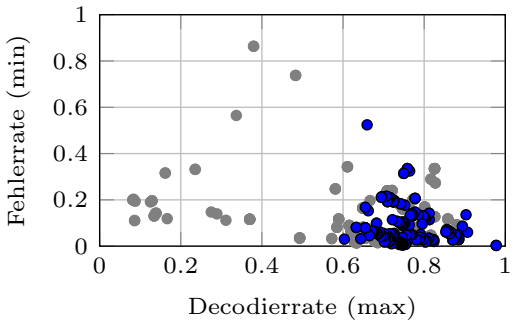
(a) Nutzdatenrate zu BER



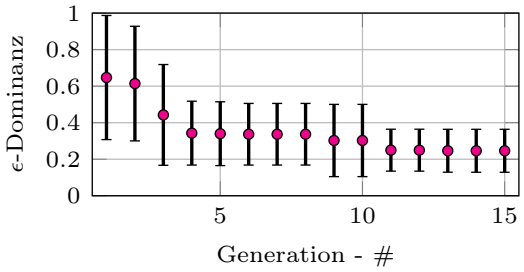
(b) Decodierrate zu BER



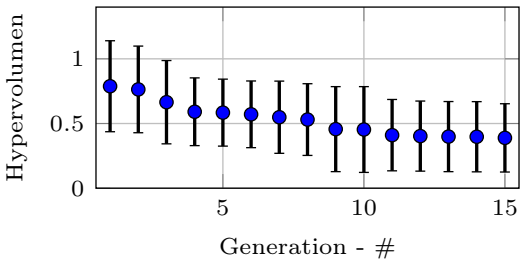
(c) Nutzdatenrate zu Decodierrate



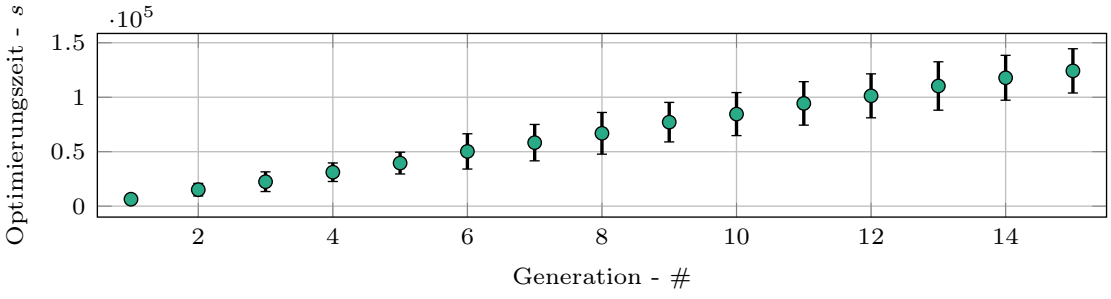
(d) Decodierrate zu Decodierfehlerrate



(e) ε-Dominanz über die Generationen



(f) Hypervolumen über die Generationen

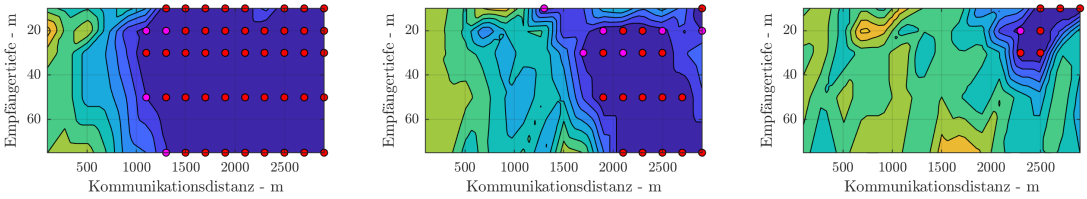


(g) Optimierungszeit

Abbildung 6.2: Ergebnis der Optimierung für die Umgebung *Trond*

6.2 Reduzierung des Analyseraums

Ein herausforderndes Problem der DSE im Bereich der akustischen Unterwasserkommunikation ist die Größe des Analyseraums geschickt zu beschränken. Der Analyseraum wächst mit jeden Parameter, der zusätzlich berücksichtigt wird. Zudem zeigen die Ergebnisse der Analyse, wie beispielhaft in den Abbildung 6.3, die Konstellationen im Kanal eine erfolgreiche Codierung aufgrund der Eigenschaften gar nicht ermöglichen (R_{dec}=0 %, rot) oder vernachlässigbar gering (0 %<R_{dec}<5 %, magenta) sind. Für die Beispiele würde ein Ausschluss aller Testpunkte mit der Decodierrate von 0 % eine Simulationsreduktion von ≈74.5 % (links), ≈21 % (mitte) und 8 % (rechts) ergeben. Im gleichen Verhältnis wurde sich auch die benötigte Simulationszeit reduzieren, unter der Annahme, dass diese für alle Testpunkte gleich ist.



(a) 1. Beispiel (AComm10) **(b)** 2. Beispiel (AComm10) **(c)** 3. Beispiel (AComm10)

Abbildung 6.3: Decodierraten basierend auf der Umgebung *AComm10*

Dies führt zu der Annahme, dass durch geeignete Analysen und formalisiertes Expertenwissen des Kanals in Kombination mit den Eigenschaften der ausgewählten Modems die Testpunkte reduziert und die Simulationszeit erheblich verbessert werden kann, ohne die Qualität der Simulation zu beeinträchtigen. Als Lösungsansätze für die Mittelwert-basierten Optimierung wird die Analyseraumreduktion mittels Ausschluss von Testpunkten durch einen geometrischen und analytischen Ansatz kombinierbar mit einer optionalen stochastischen Testpunktauswahl eingeführt.

Das Ziel dieser Reduktionen ist die Vermeidung von Simulationen einzelner definierter Testpunkte, deren negative Ergebnisses mit einer sehr hohen Wahrscheinlichkeit im Vorfeld bestimmbar sind. Hierzu werden alle Testpunkte (CIR-Matrixen), bei denen davon ausgegangen wird, dass eine Decodierrate von ≈0 % erreicht wird, ohne Simulation auf das erwartete Ergebnis gesetzt.

Um diese Testpunkte, definiert in Abschnitt 3.3.1, mit einer Decodierrate von $0\,\%$ zu finden, werden die folgenden Annahmen getroffen:

- Liegt der theoretische Detektionswert der Präambel unterhalb des geforderten minimalen Detektionsschwellwerts R_{dth} kann der Datenrahmen nicht detektiert und infolgedessen nicht dekodiert und somit von der Simulation ausgeschlossen werden.
- Nach einer erfolgreichen Detektierung eines Datenrahmens kann für die definierte Framelänge kein weiterer Datenrahmen erfolgreich detektiert und dekodiert werden.
- Für eine erreichbare BER, die größer oder gleich einem Fehler pro Datenrahmen nach den Fehlerkorrekturmaßnahmen ist, kann eine erfolgreiche Decodierung mit hoher Wahrscheinlichkeit ausgeschlossen werden.
- Ein Datenrahmen dessen $CINRR$, dem Verhältnis des Schallintensitätspegels zwischen dem LOS-Signal und den Störungen (Reflexionen) mit Rauschen, geringer als der Systemspezifikation minimal benötigten Signal-Rausch-Verhältnis (SNR)$^-$ ist, wird mit hoher Wahrscheinlichkeit nicht erfolgreich dekodiert und kann ausgeschlossen werden.
- Impulse und Amplituden der CIR, die unterhalb des Signalpegels vom berechneten Pegel des Unterwasserrauschens liegen, werden nicht berücksichtigt.

Für die Bestimmung des minimal benötigten SNR wird im Vorfeld der Optimierung eine einmalige umfassende Analyse aller Komponenten im Basisband durchgeführt und eine BER-Matrix H_{BER} erstellt. Diese kann bei Erweiterungen des Modells ggf. ergänzt und angepasst werden. Die BER-Matrix bildet den SNR in Relation zum BER nach den Fehlerkorrekturmaßnahmen für die getesteten Basisbandsystem unter idealen Bedingen in einem additives weißes gaußsches Rauschen (AWGN)-Kanal ab. Ein Kommunikationssystem, welches unter idealen Bedingungen nicht die geforderte BER erfüllt, wird nicht in der Lage sein eine fehlerfreie erfolgreiche Kommunikation über einen stärker gestörten Kanal durchzuführen.

Für die Reduktion des Analyseraums wurden basierend auf diesen Annahmen drei Ansätze zur Analyseraumreduktion untersucht; (i) die inverse geometrische Reduktion, (ii) die analytische Reduktion und (iii) die stochastische Reduktion des Testraums.

6.2.1 Geometrische und analytische Reduktion

Bei der inversen geometrischen Reduktion, wie in [222, 235] beschrieben, werden alle Signalpfade vom Empfänger zum Sender rekonstruiert, basierend auf dem in Abschnitt 3.2.1 vorgestellten geometrischen Kanalmodell. Diese Rekonstruktion erfolgt unter Verwendung der verfügbaren Parameter und wird anschließend bearbeitet.

Durch die Basisband-Analyse der BER-Matrix kann der minimale erforderliche SNR unter idealen Bedingungen bestimmt werden. Dieser Wert wird mit dem berechneten $CINRR$, der auf der jeweiligen Kanalimpulseantwort (CIR) basiert, für jeden Testpunkt verglichen. Ist der $CINRR$ kleiner als der erforderliche SNR^{-}, wird der Testpunkt von der Simulation ausgeschlossen und entsprechend bewertet. Der verwendete $CINRR$ ist eine Anpassung des $CINR$, die zwischen Interferenzen und Reflexionen unterscheidet. Diese Unterscheidung ist sowohl im Bereich der drahtlosen Kommunikation im Allgemeinen [124, 208] als auch in der akustischen Unterwasserkommunikation im Besonderen [89] von Bedeutung.

Bei Impulsantworten, die aus Messkampagnen gewonnen werden oder ohne zusätzliche Informationen vorliegen, ist es schwierig, zwischen LOS, Reflexionen und Interferenzen zu unterscheiden. Das geometrische Kanalmodell ermöglicht jedoch, durch die zusätzlich verfügbaren Kanalinformationen, eine eindeutige Zuordnung. Dies erlaubt die Anwendung der erweiterten Qualitätsmetrik [236] in Form des $CINRR$. Der $CINRR$ ist ein präziseres Maß zur Beurteilung der Signalqualität, wenn sowohl Rauschen als auch Reflexionen und eventuell auftretende Interferenzen vorhanden und eindeutig zuordenbar sind.

Die Implementierung der vorgestellten geometrischen Reduktion wird beispielhaft durch die Funktion *geoReduction.m* in Listing A.16 für die Anwendung in MATLAB demonstriert. Der Ablauf der geometrischen Reduktion ist für alle Testpunkte des Analyseraums wie folgt definiert:

0. Generierung der BER-Matrix basierend auf dem AWGN in Abhängigkeit der definierten Parameter P_i.
1. Bestimmen des minimalen $SNR^{-}(BER(P_i))$
2. Erzeugen der CIR über die definierten Parameter.
3. Berechnen des Signalpegels für das Unterwasserrauschens N in Abhängigkeit der Parameter.
4. Bestimmen des Korrellationswert der Präambel x_{pre} (Ausschluss des Testpunktes von der Simulation, wenn gilt: $R_{dth} \geq \max(corr(x_{pre}, CIR * x_{pre}))$)

5. Bestimmen des $CINRR$ (Ausschluss des Testpunktes von der Simulation, wenn Gl. (6.7) nicht gilt).

$$SNR^{-} \leq CINRR + c_{abw} = 10\log_{10}\left[\frac{\frac{1}{\Delta t}\int_{t=t_0}^{\Delta t+t_0} P_{los}(t)dt}{\frac{1}{\Delta t}\int_{t=t_0}^{\Delta t+t_0}(P_I(t)+N(t)+P_R(t))dt}\right] + c_{abw} \qquad (6.7)$$

Der Ansatz beruht darauf, dass mittels des geometrischen Modells die CIR und somit eindeutig der Signalpegel P_{los} der LOS-Komponente und P_R der Reflexionen für den Datenrahmen bestimmt werden kann. Über einen Vergleich des resultierenden $CINRR$ mit dem benötigten SNR über Gl. (6.7) und der Wahrscheinlichkeit, ob der Datenrahmen detektiert wird, kann bewertet werden, ob eine erfolgreiche Detektierung möglich ist. Die Experimente und der Vergleich der Kanalgenerierung (Abschnitt 3.3.2.3) haben aber gezeigt, dass aufgrund des Geschwindigkeitsprofil (SSP) des Schalls in Wasser die LOS-Komponente nicht immer das erste eintreffende Signal am Empfänger ist. Daher wird in den Reduktionsverfahren, das Signal welches als erstes die Bedingung des Detektionsschwellwerts erfüllt als Referenz zur Bestimmung des $CINRR$ genutzt. Zusätzlich wird die Abweichung durch die Abschätzung mittels des geometrischen Modells mit c_{abw} eingerechnet. Um zu verhindern das Testpunkte fälschlich reduziert werden, da deren berechneter $CINRR$ die Anforderungen durch die grenznahen Werte nicht erfüllt, wurde zusätzlich eine Sicherheitsmarge von $3\,\mathrm{dB}$ festgelegt.

Nach der geometrischen Reduktion werden die Ergebnisse der ausgeschlossenen Testpunkte vorgetragen und mit der Reduktionsrate R_{red}, welche das Verhältnis von tatsächlich zu testenden Punkten zur Anzahl der gesamten Testpunkte wiedergibt, zurückgegeben.

Entgegen der geometrischen Analyse wird bei der analytischen Reduktion die Kanalimpulseantwort (CIR) der jeweiligen Testpunkte als vorhanden betrachtet. Die Kanalmatrix ist dazu mit BELLHOP und Beachtung der definierten Variationen der Position und der Geschwindigkeit generiert worden. Die CIR-Matrix kann daher direkt in das Basisband transferiert und anschließend analysiert werden. Basierend auf der Abschätzung des Kanals wird entschieden, ob eine Simulation durchgeführt werden muss oder nicht. Die Schritte zum Ausschließen eines Testpunktes sind an dieser Stelle identisch zur geometrischen Reduktion, mit dem Unterschied, dass der Testpunkt nur ausgeschlossen wird, wenn keine CIR der Matrix die Bedingungen für eine Simulation erfüllt. Durch die Verarbeitung der Matrix und der CIR im Passband benötigt die analytische Reduktion erheblich mehr Zeit im Vergleich zur geometrischen Analyse.

6.2.2 Stochastische Reduktion

Abhängig von der zu testenden Lösungen und des Kanals können trotz Reduktion noch eine große Menge von Testpunkten vorhanden sein. Ein weiterer Ansatz, welcher mit der analytischen und geometrischen Reduktion kombinierbar ist, ist die zufällige Auswahl einer festen Anzahl von validen zu simulierenden Testpunkten, in folgenden benannt als stochastischer Reduktion.

Durch die randomisierte Auswahl von validen Testpunkten wird angenommen, dass die Anzahl der zu testenden Punkte reduziert werden kann, ohne signifikant die Qualität der Zielgrößen zu beeinträchtigen. Das heißt, die Qualität der Ergebnisse, die durch stochastische Verfahren erzielt werden, soll vergleichbar sein mit den Ergebnissen, die durch umfassende Auswertung des Testraums erzielt werden, besonders bei großen und komplexen Suchräumen.

Damit kann theoretisch über eine stochastische Reduktionstechnik der Rechenaufwand und somit die benötigte Zeit zur Auswertung erheblich reduziert [237] werden. Dazu werden n gültige CIR aus dem Analyseraum zufällig ausgewählt.

Für einen fairen Vergleich der Testsysteme müssten alle Lösungen mit demselben Satz von gültigen reduzierten Tests analysiert werden. Bei einer Kombination der Reduktionsmethoden kann es sein, dass es keinen gemeinsamen Satz an gültigen Testpunkten gibt. Neben dem Satz gültiger Testpunkte, Γ_{CIR}, muss zusätzlich die Reduktionsrate r_{red} der analytischen und geometrischen Reduktion berücksichtigt werden, welche auf der Anzahl der als Null interpretierten Testpunkte in Gl. (6.8) im Verhältnis zur Gesamtanzahl der Testpunkte $\#CIR$ für die analytische ($x{=}a$), geometrische ($x{=}g$) oder ohne ($x{=}o$;) Reduktion resultiert. Die Anzahl der zu simulierenden Testpunkte ist das Minimum aus validen Testpunkten nach der Reduktion und der geforderten Anzahl von Testpunkten der stochastischen Reduktion ($n'{=}\min(\Gamma_{CIR}, n)$).

$$\bar{r}_{sx}{=}\frac{\sum_1^n r_{dec}(\Gamma_{CIR,i})}{n{+}(\frac{n}{\#CIR}{\cdot}r_{red})} \qquad \text{mit } x{:=}\{a,g,o\} \tag{6.8}$$

Die Ermittlung, ob ein Testpunkt geometrisch, analytisch, stochastisch oder in Kombination ausgeschlossen werden kann, wird über die Funktionen *geoReduction.m* in Listing A.16,*anaReduction.m* in Listing A.18 und *noneReduction.m* in Listing A.17 implementiert. Nach der jeweiligen Reduktion wird optional die festgelegte Anzahl von Testpunkten über den stochastischen Reduktionsansatz bestimmt. Bei einer vorherigen geometrischen bzw. analytischen Reduktion wird das Verhältnis der Reduktion auf die zu simulierenden Testpunkte abgebildet und vorgetragen.

6.2.3 Ergebnisse der Analyseraumreduktion

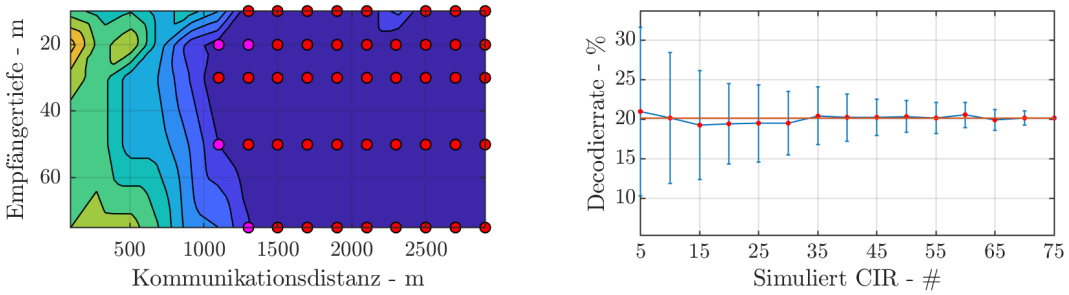
Für eine Bewertung der Reduktionsverfahren wurden für die aus der Literatur bekannten und in Abschnitt 3.3.1 eingeführten Umgebungen die Decodierrate stichprobenartig hinsichtlich der Reduktionsverfahren überprüft und bewertet. Mit dieser explorativen Analyse wird mit Hilfe von Stichproben untersucht, ob die Annahmen zur Reduktion schwach belegt werden kann und in welchem Umfang die Reduktionsverfahren die Ergebnisse verfälschen und die Optimierungszeit verkürzen.

Die Abbildungen 6.4e, 6.5e, 6.6e, und 6.7e zeigen dazu jeweils die Decodierrate in Prozent für einen Analyseraum mit definierter Sendetiefe für ein spezifisches Modem aus der Optimierung in Abschnitt 6.1.3. Jede Abbildung umfasst dazu acht Diagramme. Die oberste linke Seite gibt nach unten den kompletten Analyseraum mit ausgewerteter Decodierrate an, wobei die Testpunkte mit $R_{dec}=0\,\%$ rot und für Werte mit $0\,\%<R_{dec}\leq 5\,\%$ magenta markiert sind. Die von oben jeweils zweite und dritte Abbildung zeigt die von der Simulationen ausgeschlossen Testpunkte über rot markierte Kreuze für die geometrische und analytische Reduktion. Die vierte Abbildung zeigt die Kombination von geometrischer und analytischer Reduktion. Die rechte Abbildungsreihe zeigt jeweils die Standardabweichung S der Dekodierrate und dem Mittelwert μ (Gl. (6.9)) über die Anzahl der simulierten Testpunkt von über $10,000$ zufällig ausgewählte Kombinationen der n simulierten Testpunkte. Der Mittelwert wird über alle N_c Werte des jeweiligen Analyseraums für die geometrische (g), analytische (a) und ohne (o) Reduktion berechnet.

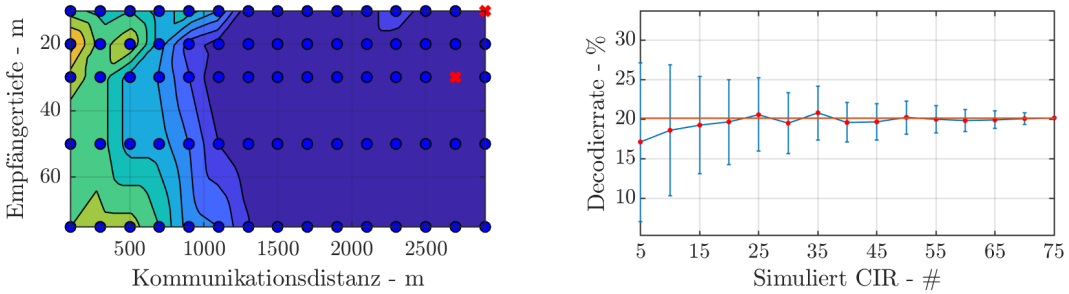
$$S=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|\bar{r}_{sx,i}-\mu|^2} \qquad \text{mit } x:=\{a,g,o\} \text{ und } \mu=\frac{1}{N_c}\sum_{i=1}^{N_C} r_{s,i} \tag{6.9}$$

Mit der so bestimmten Standardabweichung kann die Qualität der Lösungen interpretiert werden. Eine niedrige Standardabweichung zeigt, dass die Ergebnisse der durchschnittlichen Decodierrate relativ nahe am Mittelwert liegen und die Ergebnisse über den entsprechenden Ansatz auch bei einer zufälligen Auswahl und Reduktion konsistent sind. Die hohe Standardabweichung zeigt hingegen das die Ergebnisse mehr streuen und somit anfälliger für eine zufällige Auswahl.

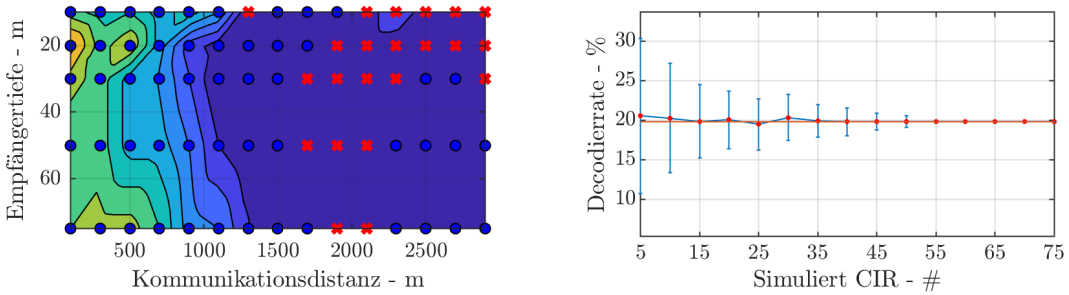
Die Ergebnisse der stichprobenartigen Untersuchung zeigen deutlich, dass sowohl die sehr schnelle geometrische als auch die analytische Reduktion den Analyseraum reduzieren können. Allerdings weichen die geometrische und analytische Reduktion in den Stichproben um bis zu $3\,\%$ von der durchschnittlichen Decodierrate $\overline{R_{dec}}$ ohne Reduktion ab. Bei beiden Reduktion traten nach bisherigen Untersuchungen die größeren Abweichungen ($>1\,\%$) nur sporadisch



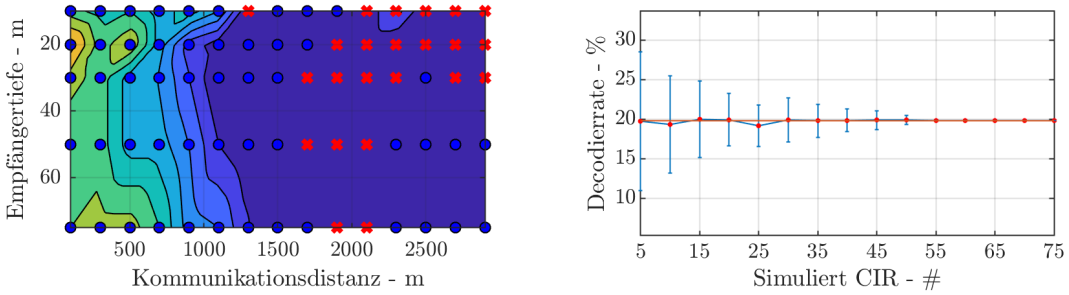
(a) Keine Reduktion



(b) Geometrische Reduktion

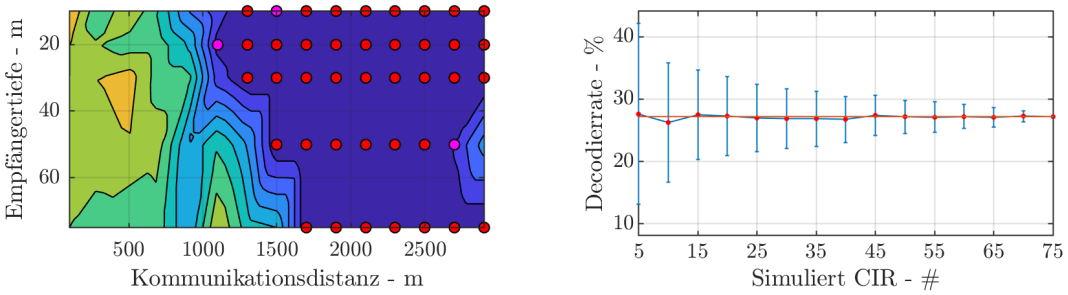


(c) Analytische Reduktion

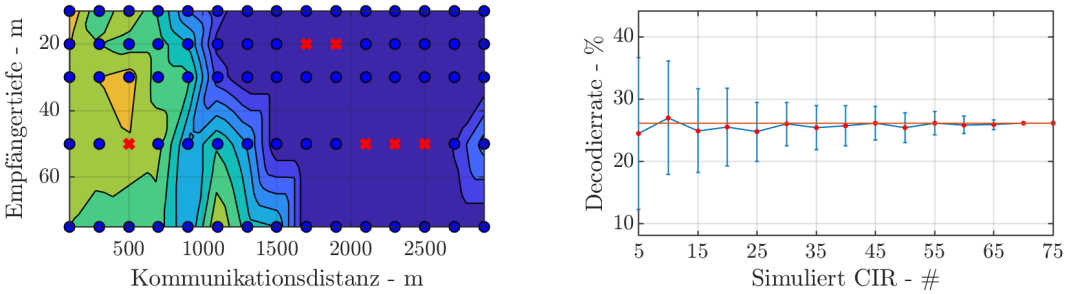


(d) Geometrisch und analytisch kombinierte Reduktion

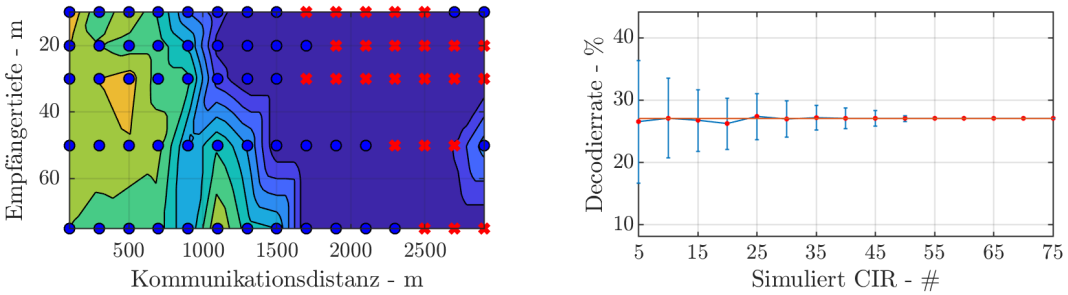
(e) Decodierraten des 1. Beispiels basierend auf der Umgebung *AComm10*



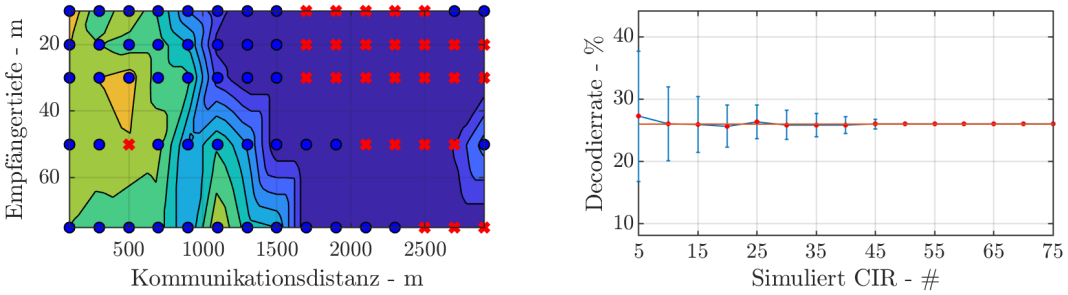
(a) Keine Reduktion



(b) Geometrische Reduktion

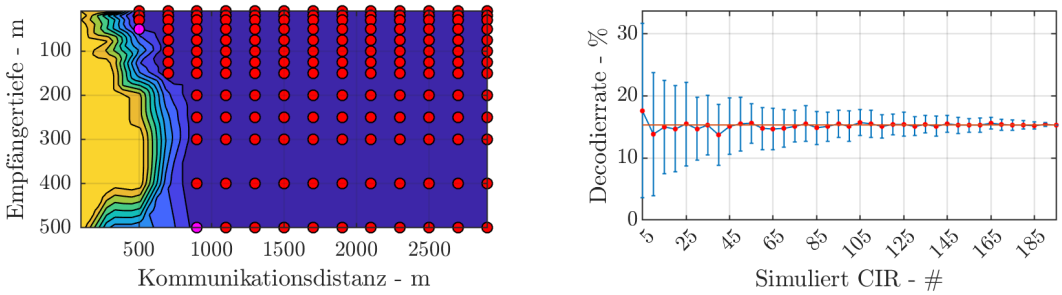


(c) Analytische Reduktion

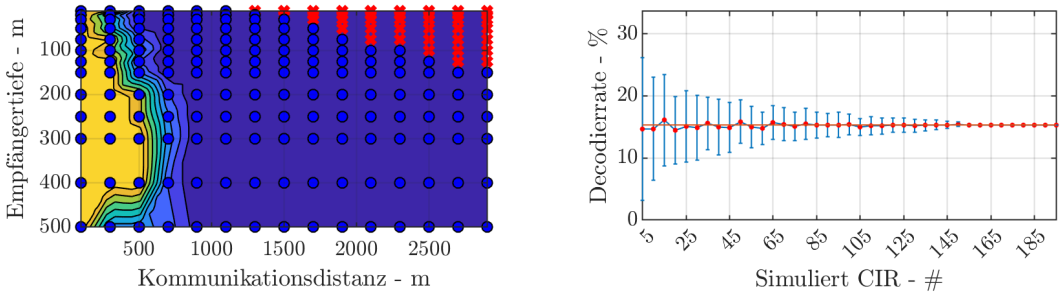


(d) Geometrisch und analytisch kombinierte Reduktion

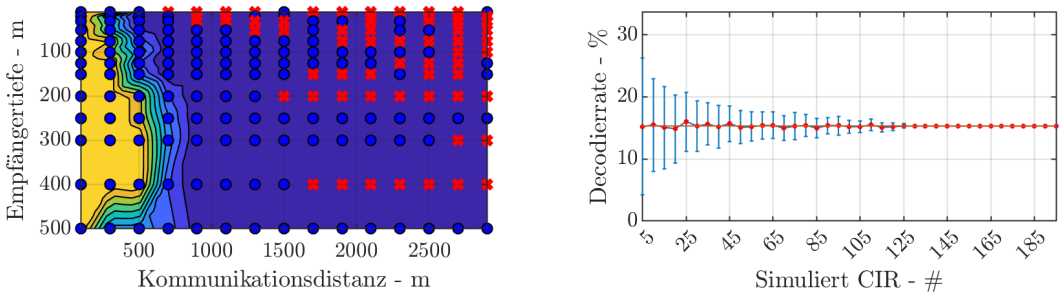
(e) Decodierraten des 2. Beispiels basierend auf der Umgebung *AComm10*



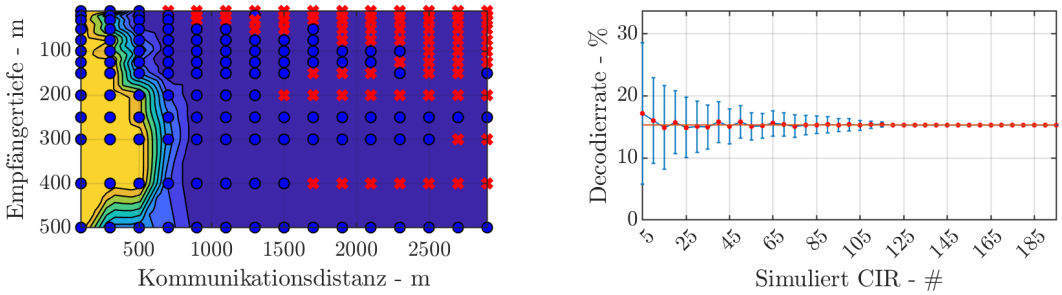
(a) Keine Reduktion



(b) Geometrische Reduktion

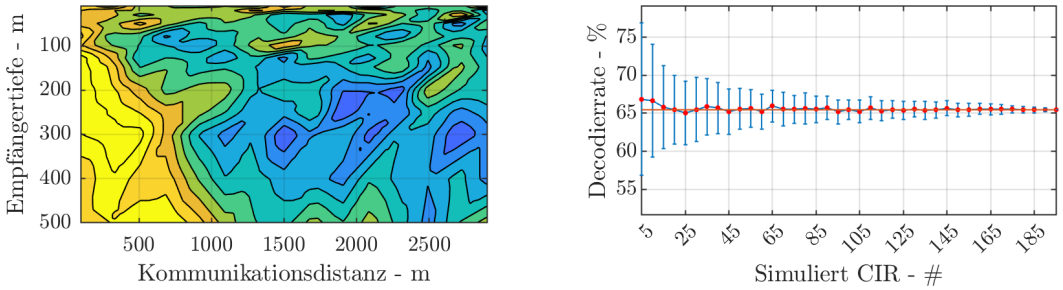


(c) Analytische Reduktion

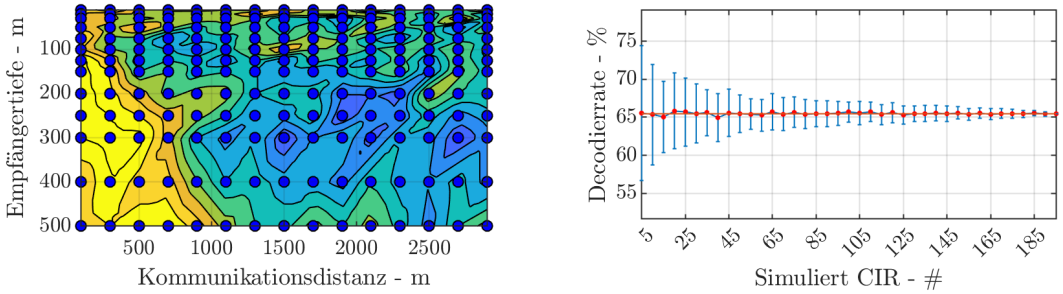


(d) Geometrisch und analytisch kombinierte Reduktion

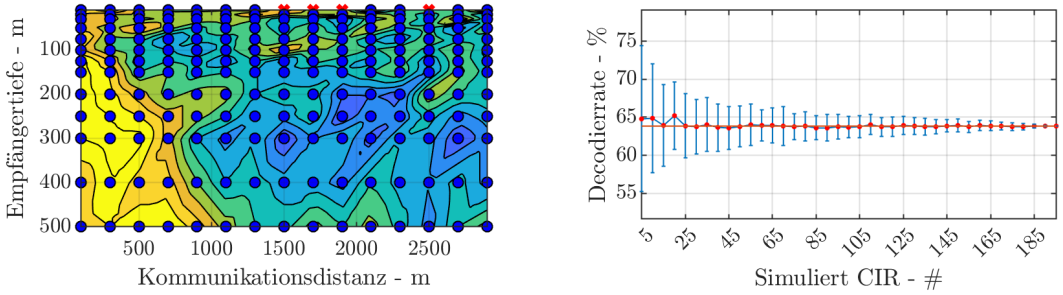
(e) Decodierraten des 3. Beispiels basierend auf der Umgebung *Trond*



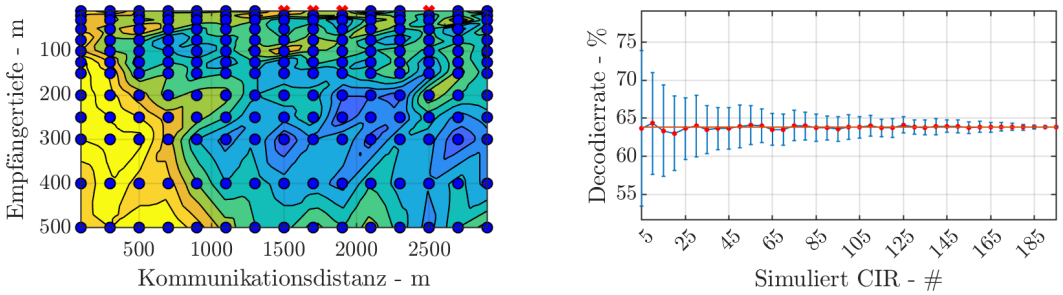
(a) Keine Reduktion



(b) Geometrische Reduktion



(c) Analytische Reduktion



(d) Geometrisch und analytisch kombinierte Reduktion

(e) Decodierraten des 4. Beispiels basierend auf der Umgebung *Trond*

auf. Damit kann der Analyseraum mit der analytische Reduktion nicht reduziert werden ohne den Durchschnittswert der Decodierrate zu verfälschen.

Aber trotz der Abweichung stechen beide Reduktionsverfahren durch eine hervorragende Annäherung der durchschnittlichen Decodierrate bei gleichzeitiger Reduktion des Analyseraums hervor.

Die zufällige Auswahl von Testpunkten zeigt darüber hinaus, bei 10000 zufälligen Kombinationen, dass sich der Mittelwert für alle drei Varianten (ohne, geometrisch, analytisch) dem jeweiligen Mittelwert der Decodierrate über alle Testpunkte entspricht. Unter der Annahme, dass eine Abweichung von $\approx 3\,\%$ eine akzeptable Abweichung vom Mittelwert für eine Optimierung ist, können die Testpunkte über die stochastische Reduktion um mind. $35\,\%$ reduziert werden. Bei einer genauen Betrachtung wird auch deutlich, dass die beiden Verfahren (geometrisch, analytisch) teils diverse Testpunkte ausschließen, daraus ergibt sich die Möglichkeit alle drei Reduktionen zu kombinieren und die Reduktion zu maximieren bei einer maximalen Abweichung von $5\,\%$. Da die geometrische Reduktion im Durchschnitt weniger als $\approx 3\,\%$ und die analytische um weniger als $1\,\%$ vom originalen Mittelwert abweichen, können die Verfahren kombiniert werden. Proportional zum Testraum lässt sich damit auch die notwendige Zeit für die Optimierung damit um mindestens $35\,\%$ reduzieren und entspricht im Beispiel der Optimierung von *AComm10* einer Reduzierung von $47\,\mathrm{h}$ auf maximal $30.5\,\mathrm{h}$.

6.3 Entwurfsraumbeschränkung

Die Auswahl der optimalen Kombination aus Modulationstyp, Fehlerkorrektureigenschaften, Synchronisationseigenschaften und anderen Parametern ist auch unter Verwendung der eingeführten Reduktionsverfahren sehr zeitaufwändig und komplex. Zudem hat die Vielzahl der Experimente gezeigt das viele Lösungen zu keinen zufriedenstellenden Ergebnissen führen, da Sie entweder kaum oder nur selektiv eine Decodierrate über $0\,\%$ oder eine extrem niedrige Datenrate und entsprechende lange Sendezeit aufweisen.

Für den Ausschluss dieser ungewollten und simulativ teuren Lösungen wurde der Ansatz zur Beschränkung des Entwurfsraum, zu teilen publiziert in [222], anhand von szenario-, anwendungs- und leistungsbezogenen Schranken eingeführt. Ziel dieses Verfahrens ist es somit nicht nur einzelne Testpunkte von der teuren Simulation auszuschließen, sondern die komplette Lösung auf Basis von Schranken die durch den Anwender, die Anwendung oder der Umgebung vorgegeben sind.

Das Verfahren basiert, wie die anderen Reduktionsansätze auf der BER-Matrix H_{BER} für das Basisbandsystem unter optimalen Bedingungen. Die Methode unterscheidet hier zwischen der statischen und dynamischen Modemauswahl. Das statische Modem bleibt für eine Mission unveränderlich bezüglich der Systemkomponenten und Eigenschaften, während beim Einsatz eines dynamischen Modems davon ausgegangen wird, dass es die Komponenten und Systemeigenschaften während der Mission ändern kann. Damit müssen Implementierungen des dynamischen Modems nur einen Teil der zu erwartenden Kanaleigenschaften abdecken.

Die Hauptmerkmale für eine Beschränkung des Entwurfsraumes konnten durch Experimente auf die Kanal-Kapazität, das obere Limit der übertragbaren Datenmenge über einen Kanal basierend auf dem $CINRR$, der verfügbaren Bandbreite und die benötigte Datenrate, definiert durch aufgabenspezifische Anforderungen oder durch die Nutzlastlänge, beschränkt werden. Zudem hat sich gezeigt das die Hardware-spezifische Einschränkungen, Umweltfaktoren, Schutzabstand zwischen Präambel und Daten, der Datenrahmens-Abstand und die Kommunikationsprotokolle die erreichbare Datenrate im Unterwasserkanal, aufgrund der spezifischen Eigenschaften des Kanals, erheblich beeinflusst. Das benötigten Kapazitäten werden für einen idealen Unterwasserkanal, ähnlich der eingeführten Reduktionen im Analyseraum, basierend auf der Entfernung zwischen Sender und Empfänger, Trägerfrequenz, Umgebungscharakteristiken und der zugehörigen Dämpfung des Signals durch Ausbreitungsverlust, Absorption, Dopplereffekte und der Sensitivität der Transducer mit Hilfe des geometrischen Kanalmodells und das Umgebungsrauschen über die formalisierten Funktionen für alle Testpunkte berechnet werden.

6.3.1 Kanalkapazität

Die Kanalkapazität, bezeichnet als C, ist das obere Limit für die über einen Kanal übertragbare Datenmenge, basierend auf dem SNR und der verfügbaren Bandbreite B. Gemäß dem Shannon-Theorem [238] muss die Kapazität C die erforderliche Datenrate R_D überschreiten (Gl. (6.10)). Die verfügbare Kapazität kann durch die minimale theoretische Kapazität bestimmt werden, diese steigt mit zunehmendem SNR und wird durch die Bandbreite, definiert über die maximalen Anzahl von Bits pro Symbol bps und der Symbolrate f_{sym} in Symbolen pro Sekunde, beschränkt. Eine Alternative zur Shannon ist die von der Bandbreite

unabhängige spektrale Effizienz η η:

$$R_D \leq C = \min(B \cdot \log_2(1+SNR), f_s \cdot f_s) \tag{6.10}$$

$$\frac{R_D}{B} \leq \eta = \min(\log_2(1+SNR), \frac{f_{sym} \cdot f_s}{B}) \tag{6.11}$$

6.3.2 Datenrate

Um die erreichbare Datenrate zu bestimmen, muss das verwendeten Codierungsschemata und dessen Redundanz berücksichtigt werden. Dazu gehören die Rate des Faltungscodierers R_{CC} und R_{RS} für die Rate des Blockcodierers. Zusätzlich fließen die erforderliche Länge des Headers l_h, die Präambellänge l_{pre} für die Signaldetektion, der Schutzabstand l_{gd} zur Vermeidung von Überlappungen zwischen Daten und Präambel sowie die maximale Anzahl von Rahmen, die pro Sekunde übertragen werden können $R_f = \frac{1}{t_{\mathrm{frame}}}$, also der Kehrwert der Rahmendauer t_{frame} bei der Bestimmung der Datenrate in Gl. (6.12) ein.

$$R_D = \left(\frac{R_P \cdot t + l_h}{R_{cc} \cdot R_{rs}} + l_{pre} + l_{gd} \right) \cdot R_f = \left(\frac{l_p + l_h}{R_{rs} \cdot R_{cc}} + l_{pre} + l_{gd} \right) \cdot R_f \tag{6.12}$$

Die erforderliche Datenrate R_D kann basierend auf aufgabenspezifischen Anforderungen an die Nutzdatenrate R_p oder durch Berücksichtigung der Nutzdatenlänge l_p (siehe Gl. (6.12)) in Verbindung mit der Rahmenrate R_f, gemessen in Datenrahmen pro Sekunde, bestimmt werden.

Zur Berechnung der Datenrate müssen aber weitere Faktoren, einschließlich hardware-spezifischer Einschränkungen, Umgebungsvariablen und Kommunikationsprotokolle betrachtet werden. Diese umfassen Aspekte wie die Struktur von Daten- und Steuerrahmen sowie den Datenfluss. Eine primäre Herausforderung bei der Festlegung der erreichbaren Datenrate für die Unterwasserkommunikation ist die variable, ortsabhängige Verzögerung (t_d), die die Daten während der Übertragung zwischen Sender und Empfänger erfahren.

Für eine gegebene durchschnittliche Verzögerung $\overline{t_d}$ über alle Testpunkte kann die zur Übertragung der Nutzlast erforderliche Zeit mithilfe der Gleichung 6.13 ohne zusätzlichen Austausch von Kontrollrahmen (t_{cdt}) und mit der Gleichung 6.14 unter Einbezug der zusätzliche Kontrollrahmen (t^*_{cdt}) genauer bestimmt werden. Damit ergibt sich über Gl. (6.15) die durchschnittliche Datenübertragungsrate R_D und über Gl. (6.16) den durchschnittlichen Nutzdatenrate R_p. Diese Raten unterscheiden sich in Ihrer Anwendung. Für die Kommunikation über Broadcast nur Daten gesendet werden und es zu keinem Austausch sonstiger Informationen kommt hat die Latenz kaum Einfluss auf die Datenrate. bei der Kommunikation

mit und ohne RTS/CTS-Austausch werden zusätzliche Informationen ausgetauscht und je nach Entfernung wird durch die auftretende Verzögerung im Kanal die Nutzdatenrate entsprechend reduziert.

$$t_{\text{cdt}} = t_{\text{BO}} + t_{\text{difs}} + t_{\text{sifs}} + t_{\text{data}} + t_{\text{ack}} + 2 \cdot \overline{t_d} \tag{6.13}$$

$$t^*_{\text{cdt}} = t_{\text{BO}} + t_{\text{difs}} + 3 \cdot \text{sifs} + t_{\text{rts}} + t_{\text{cts}} + t_{\text{data}} + t_{\text{ack}} + 4 \cdot \overline{t_d} \tag{6.14}$$

$$R_D = \begin{cases} l_{\text{data}}/t_{\text{data}} & \text{Broadcast} \\ (l_{\text{rts}} + l_{\text{cts}} + l_{\text{data}} + l_{\text{ack}})/t^*_{\text{cdt}} & \text{mit RTS/CTS} \\ (l_{\text{data}} + l_{\text{ack}})/t_{\text{cdt}} & \text{ohne RTS/CTS} \end{cases} \tag{6.15}$$

$$R_P = \begin{cases} l_{\text{p}}/t_{\text{data}} & \text{Broadcast} \\ l_{\text{p}}/t^*_{\text{cdt}} & \text{mit RTS/CTS} \\ l_{\text{p}}/t_{\text{cdt}} & \text{ohne RTS/CTS} \end{cases} \tag{6.16}$$

6.3.3 Störabstand von Träger und Reflexion

Die alleinige Verwendung des SNR basierend auf dem AWGN, die Leistungsfähigkeit des Systems in Unterwasserumgebungen. Um eine präzisere Leistungsanalyse durchzuführen, ist es notwendig, die Reflexion des Kommunikationssignals durch Objekte und die natürlichen Grenzen im Kanal und die Interferenzen durch andere Kommunikationsteilnehmer und Störquellen zu berücksichtigen, insbesondere in Mehrknotenkommunikationsszenarien. Daher führen wir das $CINRR$ in Gleichung 6.17 ein. Der $CINRR$ berücksichtigt den minimalen Abstand des Trägersignals und aller anderen Signale des drahtlosen Kommunikationssignals unter Berücksichtigung der Leistung des Umgebungsrauschen N, Interferenzen P_I und Reflexionen P_R, die durch natürliche Barrieren und Objekte verursacht werden. Der Frequenzbereich wird durch die Trägerfrequenz f_c definiert und durch die Bandbreite B begrenzt.

Insbesondere stellen Reflexionen in der Unterwasserkommunikation eine Herausforderung dar, da sie während der Kommunikation schwer zu identifizieren und zu trennen sind. Die Offline-Analyse, basierend auf der Basisbandanalyse, bei der alle Parameter bekannt sind, umgeht dieses Problem. Dieser Ansatz ermöglicht die Schätzung von Effekten im Voraus mithilfe des $CINRR$ (Gl. (6.17)) für ein definiertes Sendeintervall Δt.

$$\text{CINRR} = 10 \log_{10} \left[\frac{\frac{1}{\Delta t} \int_{t=t_0}^{\Delta t + t_0} P_s(t) dt}{\frac{1}{\Delta t} \int_{t=t_0}^{\Delta t + t_0} (P_I(t) + N(t) + P_R(t)) dt} \right] \tag{6.17}$$

Der $CINRR$ bietet damit eine Grenze zur Eliminierung von möglichen Umsetzungen, welche die Anforderungen nicht erfüllen werden. Es wird angenommen, dass die Signalleistung der Interferenzen P_I durch zusätzliche Schallquellen anderer Modems nicht vorhanden ist und daher nicht explizit berücksichtigt werden braucht. Der Faktor muss aber in Analysen von Mehrknotennetzwerken betrachtet und ausgewertet werden.

6.3.4 Minimaler Signalstörabstand

Für die Datenanalyse wurde im Vorfeld die BER-Matrix über den definierten Parameterbereich des Modells in Tabelle 6.1, beschränkt auf das Basisband, erstellt. Diese mehrdimensionale Matrix setzt die BER in Relation zum SNR, basierend auf dem AWGN, und repräsentiert den notwendigen SNR für einen definierten BER im Basisband. Bei der Kommunikation im Passband treten jedoch zusätzliche Fehler auf, die durch den Kanal und andere Komponenten wie Up- und Down-Converter erzeugt werden. Ein System, welches diese optimalen Bedingungen im Basisband nicht erfüllt, wird diese im Passband auch nicht erfüllen können. Damit bilden die Werte der Matrix die untere Schranke der Kapazität ab und können auf alle Passband-Kombinationen und -Erweiterungen angewendet werden. In dieser Arbeit wird Verfahren mit Kasami-basiertem Präambel genutzt. Es reduziert damit die Kontrolldaten des Datenrahmen auf $11\,\mathrm{Byte}$ [174] während die originale Länge gemäß 802.11 $28\,\mathrm{Byte}$ umfasst. Dies wirkt sich zusätzlich positiv auf die Fehlerkorrektur durch den kleineren Datenrahmen bei gleicher Nutzdatengröße aus und führt damit zu einem geringeren BER.

Die BER-Matrix wird mittels einer Datenrahmen-basierten Hybridsimulation im Basisband erstellt, welche den BER nach Decodierung und Detektion im AWGN-Kanal, basierend auf einer Nutzlast von $1\,\mathrm{Mbit}$ über den SNR-Bereich von $-10\,\mathrm{dB}$ bis $20\,\mathrm{dB}$ bestimmt. Mit den vorgehaltenen Werten dieser Matrix kann für jede gültige Implementierung im Passband generisch geprüft werden, ob diese die notwendigen Kriterien für eine definierte BER mittels Bestimmung des $CINRR$ auf Basis des geometrischen Kanalmodells erfüllt.

6.3.5 Schutzabstand

Der Schutzabstand l_{gd} verhindert, dass die Datenrahmen mit der Präambel am Empfänger überlappen. Der Schutzabstand kann über die Zeitdifferenz zwischen dem direkt eintreffenden und dem am längsten verzögerten Signal

Komponente	Variante	Parameter		Beschreibung
Blockcodierer	CRC	k	4	Anzahl der Symbole zur Validierung
		d	$[(11+8):8:(255-k)]$	Datenlänge
	RSC	k	[4 8:8:128]	Anzahl der redundanten Symbole
		d	$[(11+8):8:(255-k)]$	Datenlänge
Faltungscodierer	none CC	r	1	Codierrate
	CC 1/2	r	$1/2$	Codierrate
		cl	7	constraint length
		gp	$[171\ 133]_8$	Generatorpolynom
	CC 1/4	r	$1/4$	Codierrate
		cl	7	Konstruktionslänge
		gp	$[133\,171\,117\,165]_8$	Generatorpolynom
Datenframer	Kasami	l_{ID}	$64, 128$	Länge der Präambel zur Identifikation
		l_{DT}	$64, 128$	Länge der Präambel des Datenrahmentyps
Interleaver	none			
	multiplex	$delay$	$[1;0;2;1]$	Verzögerungsmatrix
	matrix	$matrRow$	8	Anzahl von Zeilen der Matrix
Scrambler	none			
	generic	$sync$	[0 1]	Selektor (Asynchron, Synchron)
		gp	$[00014000]_8$	Generatorpolynom des Scramblers
		ip	$[00020510]_8$	Initialwort des Schieberegisters
Modulation	QPSK	M	4	Anzahl der Symbole
		sps	1	Abtastwerte pro Symbol
	BPSK	M	2	Anzahl von Symbolen
		sps	1	Abtastwerte pro Symbol
	FH-FSK	M	2	amount of symbols
		sps	4	Werte pro Symbol
		nc	[1 2 4]	Anzahl genutzter Kanäle (Frequenzbereiche)
		B	[1 5]	Bandbreite

Tabelle 6.1: Modemparameter im Basisband

bestimmt werden. Unter der Annahme, dass das übertragene Signal einem Dirac-Impuls entspricht ($x(t)=\delta(t)$) und dass die Amplitude durch $\rho_n e^{j\phi_n}$ abgebildet wird (Gl. (6.18)), ergibt sich der Schutzabstand l_{gd} über Gl. (6.19) mittels der maximalen Verzögerung τ_n der einzelnen Signalpfade.

$$y(t)=h(t)*x(t)=h(t)*\delta(t)=h(t)=\sum_{n=0}^{N-1}\rho_n e^{j\phi_n}\delta(t-\tau_n) \tag{6.18}$$

In der Unterwasserkommunikation kann die Berücksichtigung von Reflexionen, insbesondere in Gewässern mit mittleren Tiefen, für die Bestimmung des Schutzabstands ein Problem darstellen, da Signalverzögerungen aufgrund mehrfacher Reflexionen im Bereich von Sekunden liegen können. Lange Schutzabstände verringern die Netzwerk-Durchsatzrate und die Nutzlastrate und erhöhen die Verzögerung. Daher wird in dieser Arbeit ein angepasstes Verfahren zur Bestimmung des Schutzabstands genutzt. Mithilfe des geometrischen Kanalmodells wird die maximalen positionsabhängige nötige Verzögerungen bestimmt bei denen die Störungen des Rahmens durch die Präambel unterhalb des minimal benötigten SNR, für eine fast fehlerfreie Kommunikation, liegt. Aus der erhaltenen Verteilung von Verzögerungen für einen Kanal werden abschließend die Ausreißer mit geringer Wahrscheinlichkeit ausgeschlossen.

Das heißt für die Berechnung werden nur Reflexionen, die zu einem $CINRR$ führen, der geringer ist als das erforderliche minimale SNR für fehlerfreie Übertragung in die Bestimmung einbezogen, womit der Schutzabstand $\tilde{l_{gd}}$ über Gl. (6.19) unter Berücksichtigung der Symbolrate f_{sym} und der durchschnittlichen Schallgeschwindigkeit $\bar{c}$ mithilfe des Q_{75}-Quantils und 1.5-mal des Interquartilbereichs $IQR_{Turkey}=(Q_{75}-Q_{25})\cdot 1.5$ zum Ausschluss oberer Ausreißer (Tukey's Fence [239]), bestimmt werden kann. Für die Bestimmung des notwendigen Schutzabstandes wird für alle definierten Empfänger-Sender-Konstellation die Zeit τ_n bestimmt, in der Reflexionen auftreten welche das Signal so stören das die untere Schranke des minimal benötigten SNR^- verletzt wird. Die so bestimmte Verzögerung kann über die Symbolrate f_s in den benötigten Schutzabstand in Symbolen überführt werden. Hierbei bildet der minimale Schutzabstand l_{gd}^-, den System-spezifisch definierter Abstand der eingehalten werden muss, wenn zusätzliche Berechnungen oder eine Aufwachzeit erforderlich sind, die untere Grenze. Der maximale Schutzabstand l_{gd}^+ ist durch die maximale Verzögerung durch Mehrwegeausbreitung definiert und bildet damit die obere Schranke (Gl. (6.19)).

$$l_{gd}^- \leq \tilde{l_{gd}} = \max(\text{IQR}_{Turkey}\left(l_{gd} | CINRR \geq SNR^-\right), l_{gd}^-) \leq \lceil f_s \cdot max(\tau_n) \rceil = l_{gd}^+ \qquad (6.19)$$

Damit können nun zwei, in der Literatur nur wage beschriebene, Schutzabstände bestimmt werden. Zum einem der Schutzabstand für eine spezifische Umsetzung und zum anderen über das Maximum aller bestimmten Schutzabstände den generellen Schutzabstand der für alle Systemkonfigurationen angewendet werden kann. Beide Schutzabstände sind immer gleich oder geringer wie die maximale Verzögerung die in einem solchen Szenario möglich ist.

6.3.5.1 Bestimmung des Schutzabstandes

Das Ziel des Verfahrens ist, durch eine Verkürzung des Schutzabstandes die Optimierung der Datenrate bei gleicher Performanz in Form der erfolgreichen Decodierung zu erreichen. Dazu zeigen die Abbildungen 6.8 und 6.9 eine Darstellung der Verteilung des positionsabhängigen Schutzabstandes in Relation zu dem minimal benötigten SNR^- für die Umgebungen *AComm10* und *AComm13* über die definierten maximalen Kommunikationsdistanzen. Die Abbildungen zeigen über die untere Grenze der blauen Box das 25. Perzentil (1. Quartil) und die obere Grenze das 75. Perzentil (3. Quartil) [240, 241]. Der Bereich zwischen diesen beiden Grenzen ist der Interquartilabstand und umfasst damit den mittleren $50\,\%$ der Daten, dessen Median, also das 50. Perzentil, über den schwarz markierten Punkt angezeigt wird. Die Whisker (Antennen) der Abbildung erstre-

cken sich von den Enden des Interquartilsabstands bis zur jeweils am weitesten entfernten Beobachtung, die noch innerhalb der Whiskerlänge liegt, die dem $1\,5$-Fachen des Interquartilsabstands entspricht. Alle Datenpunkte, die außerhalb dieses Bereichs liegen, werden als Ausreißer markiert (rot), interpretiert und bilden damit ungewöhnliche Werte des Datensatzes ab.

Die durchgeführten Tests haben gezeigt, dass die Hauptfaktoren, die den Schutzabstand beeinflussen, der erforderliche minimale SNR^{-} welcher direkt von der benötigten BER und der Modulation, Fehlerkorrekturmaßnahmen und der Datenlänge abhängt, die Windgeschwindigkeit, die Meerestiefe, maximale Reichweite und Kommunikationsdistanz sind.

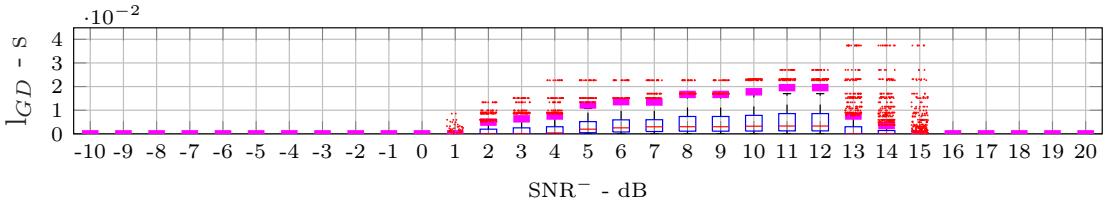
Die Visualisierungen in Abbildung 6.8 zeigen den benötigten Schutzabstand, dessen maximal resultierende $CINRR$ unterhalb des minimalen SNR liegt. Durch diese Einschränkung, unter Ausnutzen der geometrisch bestimmten Kapazitäten reduziert sich der Schutzabstand im Vergleich zur maximalen Verzögerung der Signalpfade des Kanals erheblich. Allerdings wird auch über diesen Ansatz der Schutzabstand von den auftretenden Ausreißern dominiert, welche durch die Anwendung von Tukey's Fence [239] (in Magenta) ausgeschlossen und der Schutzabstand noch weiter reduziert werden.

Bei dem resultierenden Schutzabstand zeichnen sich zwei Extreme ab. Einerseits gibt es Konstellation, wie in Abbildung 6.8a, in denen der berechnete Schutzabstand eine Nähe zu den Ausreißern aufweist und andererseits Konstellationen, in denen kein Schutzabstand benötigt wird, obwohl die Ausreißer an dieser Stelle einen hohen Wert des Schutzabstandes implizieren würden.

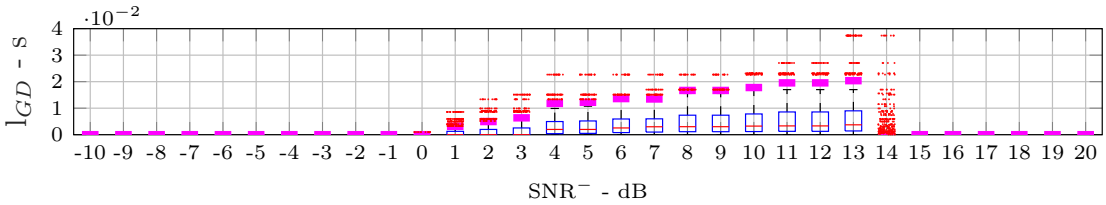
$$l_{gd}{}^{*} \leq \left\lceil f_s \cdot \max_{i \in I, j \in J} \left(\frac{R_{sb}^{+}(i,j,m) - R_{\mathrm{los}}(i,j)}{\overline{c_w}} \right) \right\rceil \tag{6.20}$$

Der Vergleich den ermittelten Schutzabstand mit dem in der Literatur allgemein definierten Schutzstand, optimistisch beschränkt auf maximal zwei Reflexionen über Boden und Oberfläche (Gl. (6.20)), bei einer durchschnittlichen Schallgeschwindigkeit von $\overline{c_w} = 1500\,\mathrm{m\,s^{-1}}$, zeigt das erhebliche Einsparpotenzial auf, ohne die Performanz zu reduzieren.

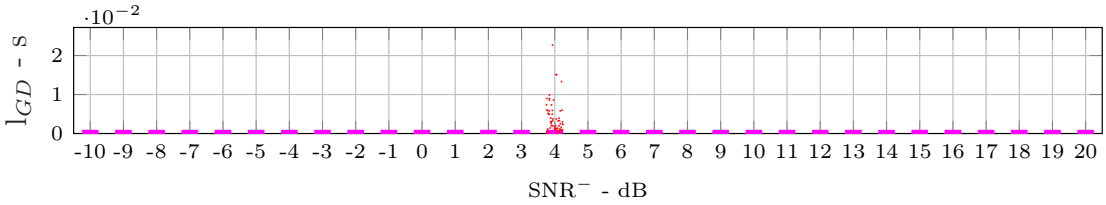
Die maximale Verzögerung beträgt, unabhängig von der Kommunikationsdistanz, bei zwei Reflexionen $112.7\,\mathrm{ms}$ für eine Gewässertiefe von $92\,\mathrm{m}$ (Abbildung 6.8). Die maximale Verzögerung beträgt $21\,\mathrm{ms}$ und reduziert damit den Schutzabstand um $\approx 82.9\,\%$ und bei einer Gewässertiefe von $2600\,\mathrm{m}$ von $3466\,\mathrm{ms}$ auf $100\,\mathrm{ms}$ um $\approx 97.1\,\%$ (Abbildung 6.9). Da der Schutzabstand je nach Kommunikationsprotokoll mehrfach vorkommen kann, wirkt sich die Einsparung damit erheblich auf die Nutzdatenrate und damit die Performanz aus.



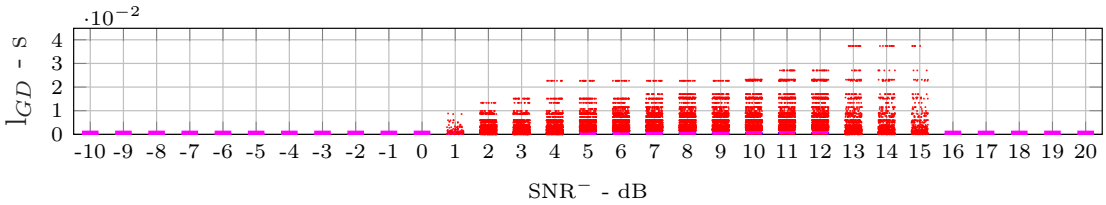
(a) QPSK mit einer maximalen Kommunikationsdistanz von 0.5 km



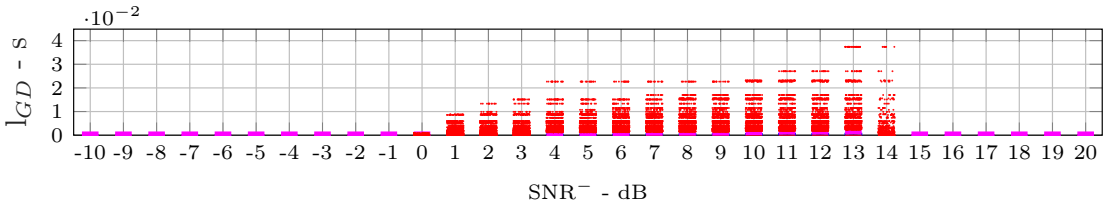
(b) BPSK mit einer maximalen Kommunikationsdistanz von 0.5 km



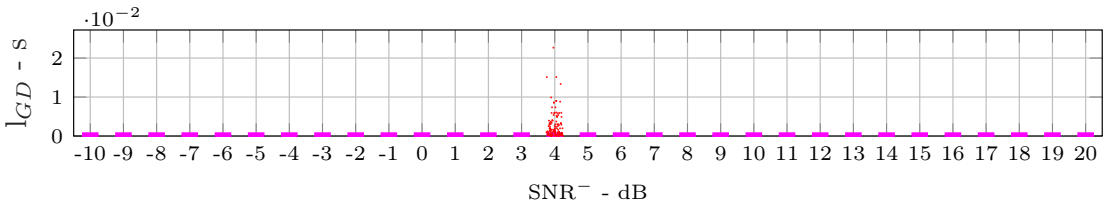
(c) FH-FSK mit einer maximalen Kommunikationsdistanz von 0.5 km



(d) QPSK mit einer maximalen Kommunikationsdistanz von 3 km



(e) BPSK mit einer maximalen Kommunikationsdistanz von 3 km



(f) FH-FSK mit einer maximalen Kommunikationsdistanz von 3 km

Abbildung 6.8: Schutzabstand in Abhängigkeit des SNR für AComm10

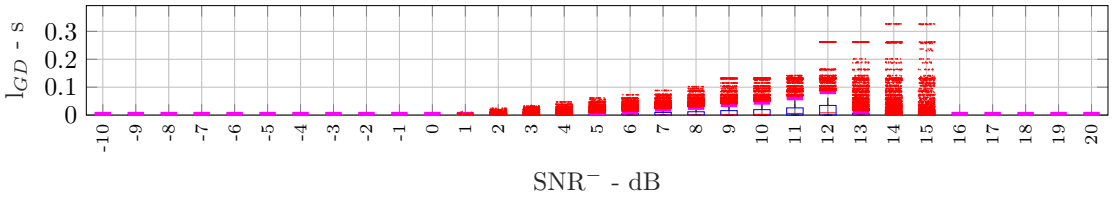
Eine weitere Konsequenz dieser Experimente ist die Notwendigkeit des Schutzabstandes. Der über die $CINRR$ bestimmte minimale Schutzabstand der Verzögerung des Signals definiert den nötigen Abstand so damit eine Überlagerung mit der Präambel den minimal benötigten SNR^{-} nicht unterschreitet. Die Verletzung dieser Grenze, durch einen Verzicht auf den Schutzabstand, führt zu einem erhöhten Fehleraufkommen und somit zu einer geringeren Decodierrate. Eine Ausnahme bildet die Kommunikation mittels Frequenzsprungverfahren kombiniert mit Frequenzumtastung (FH-FSK), für diese Modulation zeigen die Experimente, dass der Schutzabstand nicht benötigt wird.

Die Festlegung des Schutzabstandes hängt damit von der Umsetzung ab und kann über die eingeführte Methodik bestimmt werden. Für die Anwendung in statischen Szenarien, in denen eine feste homogene Konfiguration von Kommunikationssystemen vorherrscht, kann der Schutzabstand damit explizit festgelegt und bei der dynamischen Konstellation hingegen adaptiv angepasst werden, was mit dem in dieser Arbeit eingeführten Kommunikationsmodell Umsetzbar ist und die Nutzdatenrate weiter erhöhen kann.

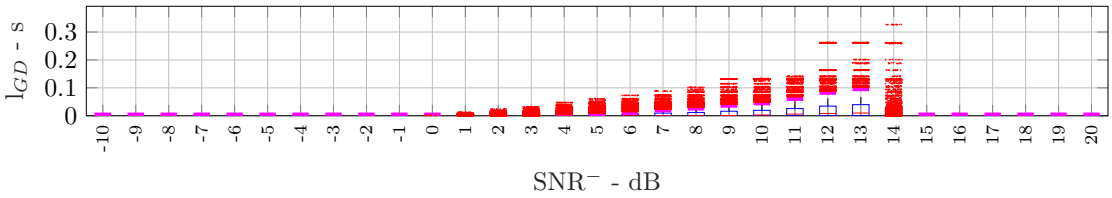
6.3.6 Beschränkung des Entwurfsraums

Die Entscheidung, ob eine Implementierung simuliert wird, erfolgt über die Funktion *capacitybasedReduction.m* in Listing A.19. Diese Funktion lädt alle implementierungsspezifischen Parameter über die vorgegebene Konfigurationsdatei, die vorgehaltene BER-Matrix des Basisbands und die erforderlichen Umgebungsdaten wie die durchschnittliche Wassertiefe, Kommunikationsreichweite und Transducer-Eigenschaften. Anschließend wird einmalig eine Matrix für die positionsspezifischen Kanalimpulsantworten und der daraus resultierende Schutzabstand für den minimal benötigten SNR^{-} mit Hilfe des eingeführten geometrischen Kanalmodells erstellt.

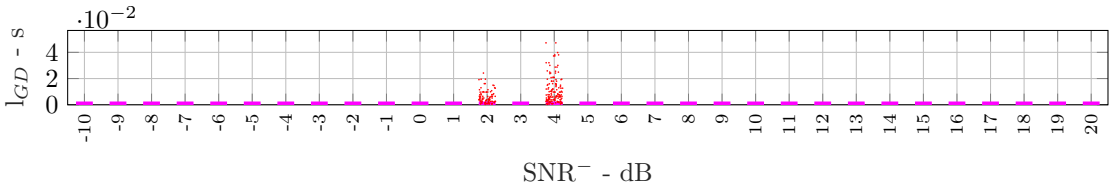
Damit kann die durchschnittliche Nutzdatenrate und die Verteilung des $CINRR$ im Analyseraum berechnet werden. Bei der Beschränkung des Entwurfsraums gibt es zwei grundlegende Ansätze für Wahl der Kommunikationssysteme zu betrachten; den statischen und dynamischen Ansatz.



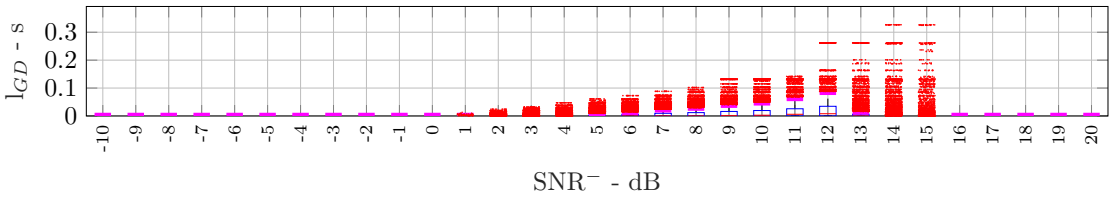
(a) QPSK mit einer maximalen Kommunikationsdistanz von 0.5 km



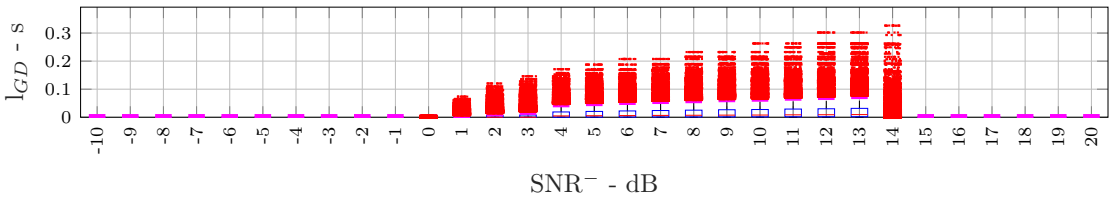
(b) BPSK mit einer maximalen Kommunikationsdistanz von 0.5 km



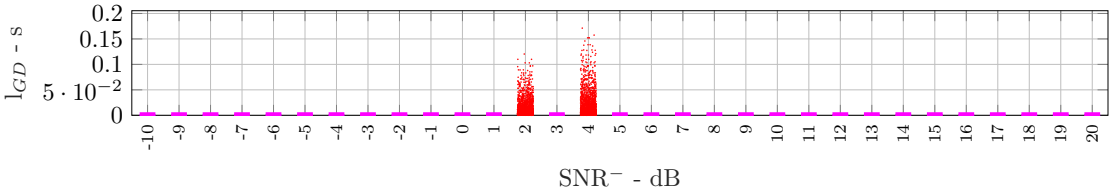
(c) FH-FSK mit einer maximalen Kommunikationsdistanz von 0.5 km



(d) QPSK mit einer maximalen Kommunikationsdistanz von 3 km



(e) BPSK mit einer maximalen Kommunikationsdistanz von 3 km



(f) FH-FSK mit einer maximalen Kommunikationsdistanz von 3 km

Abbildung 6.9: Schutzabstand in Abhängigkeit des SNR für AComm13

6.3.6.1 Statischer Ansatz

Ziel des Entwurfs eines statischen Modems ist es, eine gültige Implementierung (eine Kombination aus Komponenten und Parametern) zu finden, welche eine optimale Kommunikation im gesamten Kanal, definiert über die Zielgrößen, ermöglicht. Vereinfacht heißt das, dass in dem Großteil der Sender-Empfänger-Konstellationen eine erfolgreiche Detektion und Decodierung möglich sein sollte. Dazu muss für die Implementierung gelten, dass alle $CINRR$, definiert über die Funktion $CINRR(i,j)$, für die möglichen Testpunkte $\mathbb{P}$ mit Positionen (i,j) immer größer oder gleich dem minimal erforderlichen SNR sein müssen (Gl. (6.21)). Weiterhin muss gelten, dass die erforderliche protokollspezifische Nutzlastrate R_P^* (Gl. (6.22)) für alle Konstellationen $(i,j)\in\mathbb{P}$ im Durchschnitt mindestens erreicht wird (Gl. (6.22)).

$$\forall(i,j)\in\mathbb{P}{:}CINRR(i,j)\geq\mathrm{SNR}^- \tag{6.21}$$

$$\frac{1}{|\mathbb{P}|}\sum_{(i,j)\in\mathbb{P}} R_{P,i,j}\geq R_P^* \tag{6.22}$$

$$\mathbb{P}=\{(i,j)|i\in I{:=}\{d_z\},\, j\in J{:=}\{d_z\times\{100{:}200{:}R\}\}\} \tag{6.23}$$

Diese harten Anforderungen sind aber durch den komplexen Kanal kaum bis gar nicht möglich. Des Weiteren muss berücksichtigt werden, dass es auch Konstellation zwischen Empfänger und Sender gibt, die diese Anforderungen rein physisch in dem Kanal nicht abbilden können. Zur Abschwächung der Anforderungen wird ein 80-Prozent-Quantil eingeführt. Damit müssen die Anforderungen an die statische Umsetzung nur noch in $80\,\%$ der Testpunkte für den $CINRR$ erfüllt werden (Gl. (6.24)), damit die Umsetzung als gültig definiert wird.

$$\mathrm{IQR}_{80}(\{CINRR(i,j)|i\in I, j\in J\})\geq SNR^- \tag{6.24}$$

6.3.6.2 Dynamischer Ansatz

In einem dynamischen Kommunikationssystem, wie der Online-Auswahl oder der teilweisen Programmierung vorgewählter Komponenten, wird eine Implementierung als gültig definiert, wenn die Grenzen und Anforderungen für eine definierte Umgebung mindestens einmal erfüllt wurden.

Die niedrigen Anforderungen würden aber im Extremfall dafür sorgen, dass im Testraum für jeden Testpunkt eine andere Lösung infrage kommen könnte. Für eine zukünftige Ausrichtung auf sogenannte Kommunikationscluster, Bereiche

in denen eine Implementierung dominiert und genutzt werden kann, wird die Anforderung auf das 10-Prozent-Quantil erhöht. Das heißt, die Implementierung muss die Anforderungen in $10\,\%$ der Testpunkte erfüllen (Gl. (6.25)), um als gültig definiert zu werden.

$$\mathrm{IQR}_{10}(\{CINRR(i,j)|i\in I, j\in J\})\geq SNR^{-} \tag{6.25}$$

6.3.7 Auswertung der Entwurfsraumreduzierung

Die Bewertung der Entwurfsraumreduzierung wird auf Basis der maritimen Umgebungen *AComm10* (flaches Gewässer), *CDT* (mitteltiefes Gewässer) sowie *AComm13* (Tiefsee) durchgeführt. Für die Reduktion wurden die benutzerspezifischen Anforderungen an die erforderliche maximale R (Gl. (6.26)) und die benötigten R_p (Gl. (6.27)) festgelegt. Die Bewertung umfasst damit den Anforderungsraum von RxR_p unter der Verwendung des CSMA/CA-Protokoll mit Request To Send (RTS)/Clear To Send (CTS)-Austausch für den Einsatz als statisches oder dynamisches System.

$$R=\{3\,\mathrm{km}, 2\,\mathrm{km}, 1\,\mathrm{km}, 0.5\,\mathrm{km}\} \tag{6.26}$$

$$R_p=\{10\,\mathrm{bit/s}, 25\,\mathrm{bit/s}, 50\,\mathrm{bit/s}, 100\,\mathrm{bit/s}, 200\,\mathrm{bit/s}, 400\,\mathrm{bit/s}, 800\,\mathrm{bit/s}\} \tag{6.27}$$

Für den Ansatz der Entwurfsraumbeschränkung werden alle Systeme, definiert über die Tabelle 6.1, für das Basisband mit einer Trägerfrequenz von $f_c=10\,\mathrm{kHz}$, über alle benutzerdefinierten Entfernungen, notwendigen Nutzdatenraten, maritimen Umgebungen und einen minimalen Umgebungsrauschen für die dynamischen und statischen Verwendung analysiert und bewertet. Daraus ergaben sich jeweils 12636 Lösungen aus dem Basisband für Quadraturphasenumtastung (QPSK) und Binary Phase Shift Keying (BPSK) und 37908 für FH-FSK, die für jede Kombination der Anforderung berechnet und überprüft werden. Für die Bewertung der Beschränkung des Entwurfsraums wurden die sich ergebenden ≈ 5.3 Millionen Ergebnisse in vier Kategorien unterteilt. Die Kategorie *A* (■) umfasst Implementierungen, welche die erforderliche Nutzdatenrate und den minimal benötigten SNR^{-} in mindestens $80\,\%$ der Testpunkte für den statischen und $10\,\%$ für den dynamischen Ansatz erreichen. Die zweite Kategorie *B* (■) umfasst Umsetzungen, welche die durchschnittliche Nutzlastrate und das erforderliche SNR^{-} in $80\,\%$ der Testfälle für die statische und $10\,\%$ für die dynamische Vorauswahl nicht erreichen. Die dritte Kategorien *C* (■) umfasst Implementierungen, welche nur die Anforderungen an die Nutzdatenrate nicht erfüllen, und die vierte Kategorie *D* (■) erfüllt die Anforderung an die Nutzdatenrate, aber nicht den $CINRR$, der über dem minimalen SNR^{-} liegen muss.

Die Ergebnisse der Reduzierung sind in den Abbildungen 6.10 bis 6.18 in Prozent abgebildet. Jede Abbildung zeigt den prozentualen Anteil der Kategorien in Relation der geforderten Kommunikationsdistanz und Nutzdatenrate für ein statisches (oben) und dynamisches Modem (unten). Die Abbildungen 6.10 bis 6.12 zeigen auf, dass die Anzahl der möglichen gültigen Lösungen, basierend auf den spezifischen Anforderungen für das flache Gewässer, erheblich reduziert werden können. Die Reduktion für eine statische Nutzung zeigt, dass die Verwendung von QPSK durch den hohen, minimal benötigten SNR^{-} bezogen auf die Anforderungen, nur gültige Lösungen in einer kurzen Kommunikationsdistanz von maximal $1\,\mathrm{km}$ und eine Reduktion von mindestens $84\,\%$ aufzeigt. Die Anwendungen von BPSK und FH-FSK zeigen für eine maximale Nutzdatenrate von $200\,\mathrm{bit/s}$ für alle Kommunikationsreichweiten valide Lösungen und für höhere Datenraten keine für FH-FSK und eingeschränkt für BPSK. Dies führt zu einer Reduzierung des Erkundungsraums von mindestens $81\,\%$, hauptsächlich durch zu starke Störungen durch Reflexionen und zu niedrige Nutzdatenraten für BPSK und mindestens $12\,\%$ für FH-FSK, ausschließlich durch die nicht zu erreichenden Nutzdatenraten.

Insgesamt kann die Anzahl der gültigen Lösungen am Beispiel der Trägerfrequenz von $10\,\mathrm{kHz}$ über alle Umgebungen hinweg für die statische Auswahl um mindestens $12\,\%$ für die niedrigsten Anforderungen reduziert und zu Teilen sogar komplett ausgeschlossen werden. Das ermöglicht, je nach Anwendung eine erhebliche Verkürzung die Simulationszeit oder eine gezieltere Auswahl aus dem Entwurfsraum. Darüber hinaus stellt dieser Ansatz eine Verbindung zwischen den Anforderungen und Implementierungen in der frühen Entwurfsphase her, was eine fortgeschrittene Optimierung erleichtert und hilft, kostspielige Experimente und fehlerhafte Parameter zu vermeiden.

Obwohl diese Vorauswahl keinen Erfolg garantieren kann, wird damit sichergestellt, dass ein System, das die Anforderungen unter den bestmöglichen Bedingungen nicht erfüllt und damit impliziert, dass das System damit auch in einer herauszufordernden Umgebung scheitern wird, von der Optimierung ausgeschlossen wird. Darüber hinaus erlaubt die statistische Modemauswahl nur gültige Implementierungen, die den Analyseraum (Kommunikationsgebiet) zu $80\,\%$ abdecken. Dies ist bei der dynamischen Auswahl nicht der Fall, da diese nur eine anpassbare Abdeckung von garantiert $10\,\%$ bietet, was keine vollständige Abdeckung impliziert. Die Wahl der dynamischen Systeme muss abschließend so gewählt werden, dass der benötigte Analyseraum, den Anforderungen entsprechend, ausreichend abgedeckt wird.

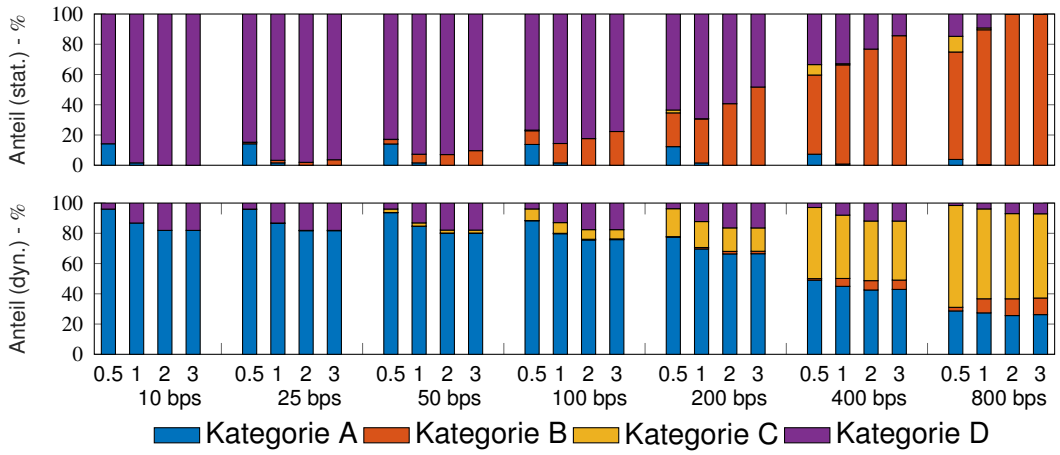


Abbildung 6.10: Beschränkung des Entwurfsraum für AComm10 (QPSK)

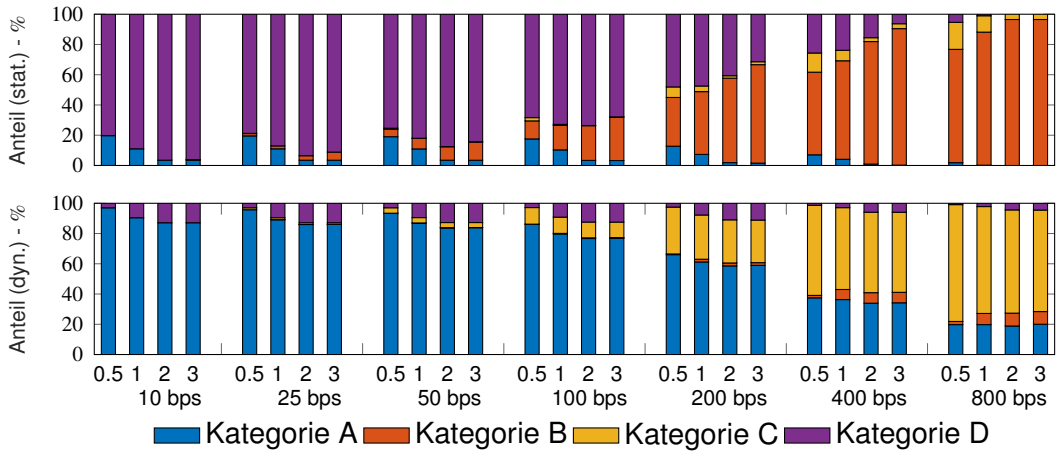


Abbildung 6.11: Beschränkung des Entwurfsraums für AComm10 (BPSK)

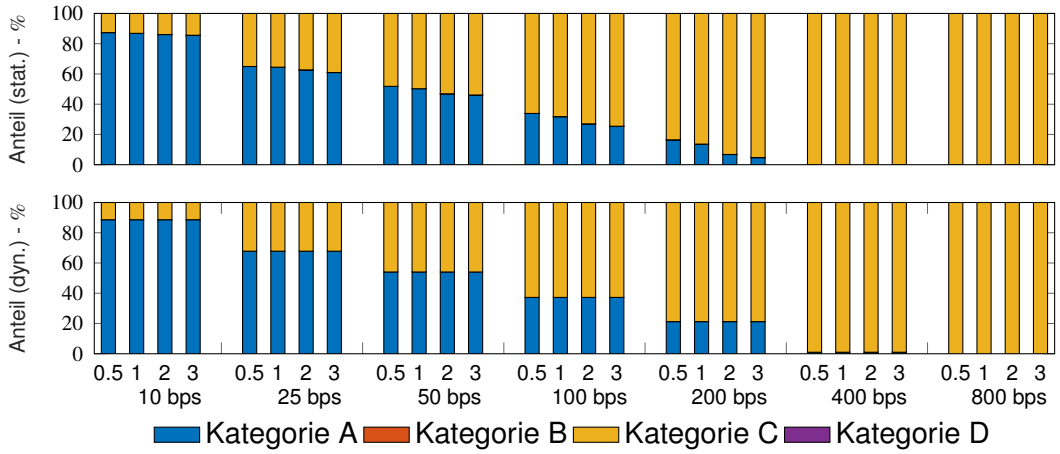


Abbildung 6.12: Beschränkung des Entwurfsraums für AComm10 (FH-FSK)

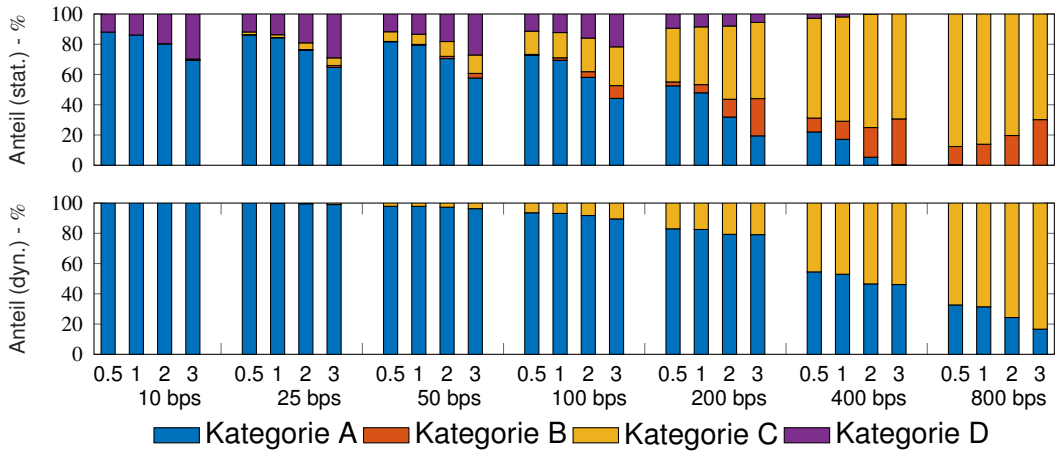


Abbildung 6.13: Beschränkung des Entwurfsraum für Trond (QPSK)

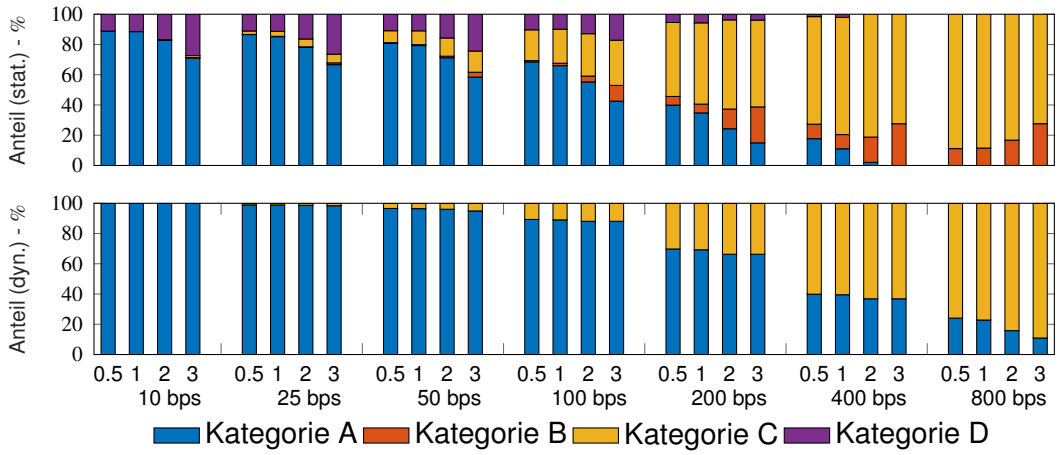


Abbildung 6.14: Beschränkung des Entwurfsraum für Trond (BPSK)

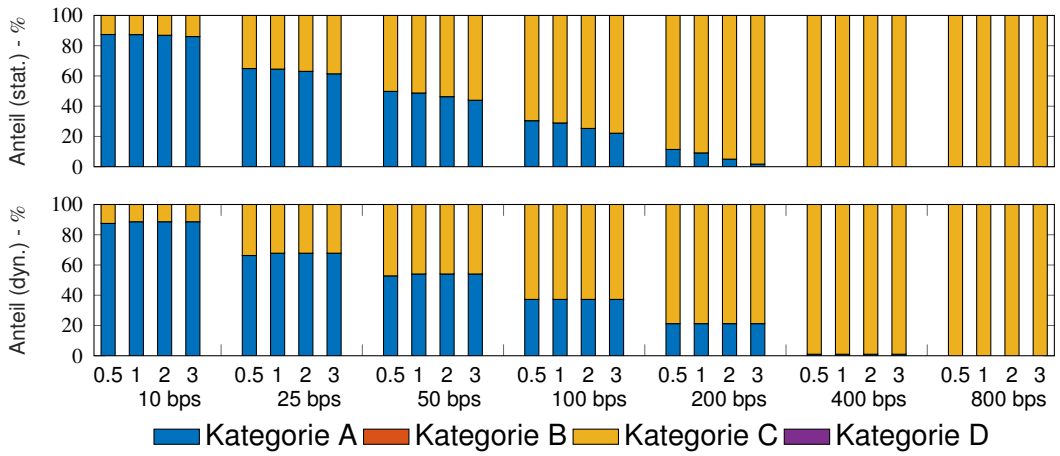


Abbildung 6.15: Beschränkung des Entwurfsraum für Trond (FH-FSK)

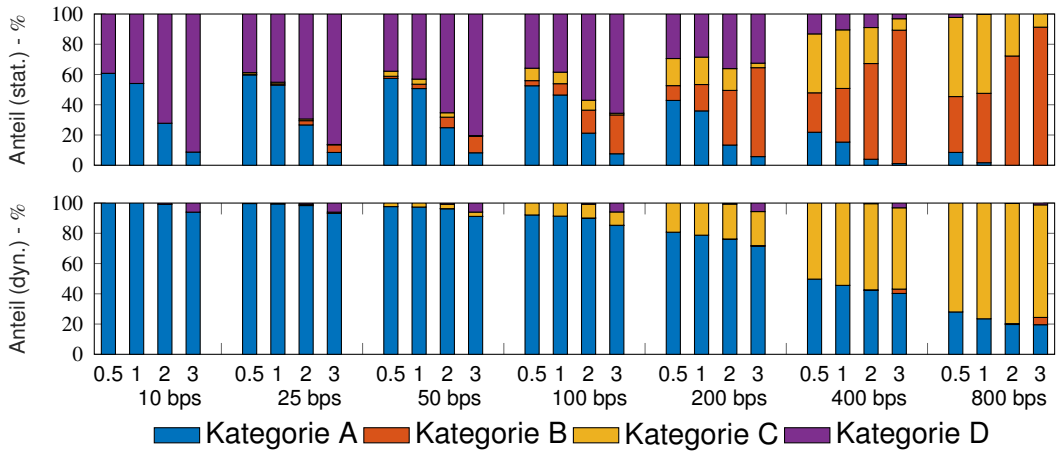


Abbildung 6.16: Beschränkung des Entwurfsraum für AComm13 (QPSK)

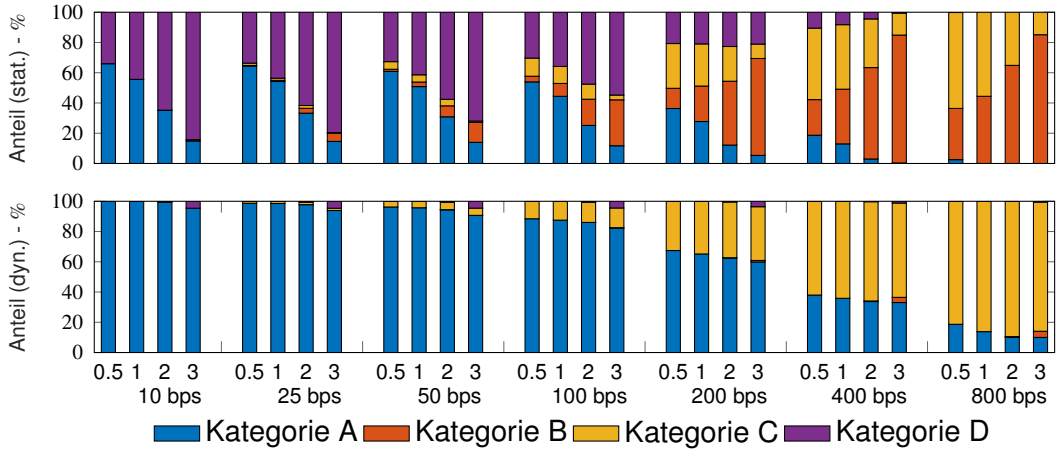


Abbildung 6.17: Beschränkung des Entwurfsraum für AComm13 (BPSK)

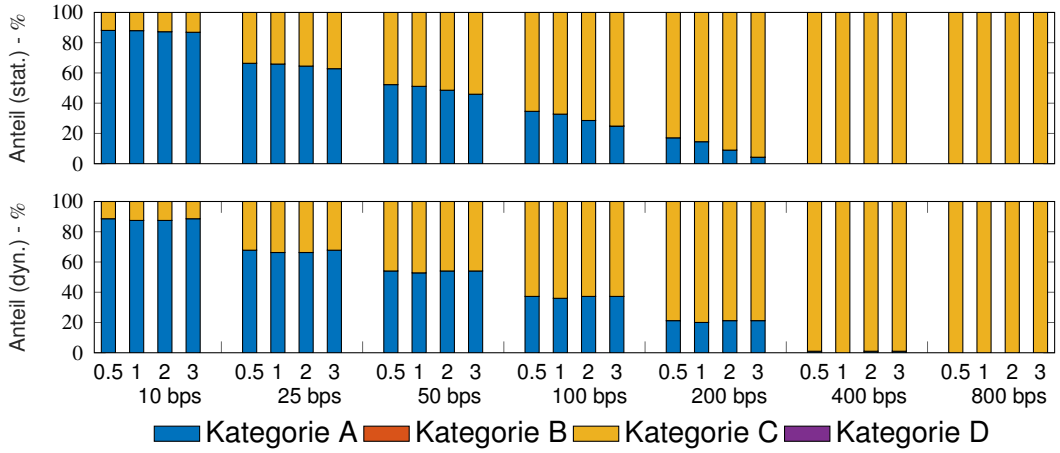


Abbildung 6.18: Beschränkung des Entwurfsraum für AComm13 (FH-FSK)

Die Ergebnisse verdeutlichen zwei wesentliche Aspekte: Das Hauptausschlusskriterium für robuste Verfahren wie Frequenz-Umtastung (FSK) und FH-FSK ist die niedrige Nutzdatenrate. Infolgedessen wird damit die Wichtigkeit der Latenz-Reduzierung über die Optimierung des Schutzabstandes, der Verkürzung von Informationen in der Protocol Data Unit (PDU) und eine ausgewogenen Wahl der Nutzdatenlänge in Bezug auf Redundanz verdeutlicht. Alle diese Verbesserungen haben eine erhebliche Auswirkung auf die Nutzdatenrate und damit auf die Nutzbarkeit im Bereich der Unterwasserkommunikation.

Ein zweiter Aspekt, der hier deutlich wird ist das die Verfahren wie QPSK und BPSK erhebliche Defizite in flachen und tiefen Gewässern aufweisen, insbesondere wenn nach einer einzigen Lösung (statischer Ansatz) gesucht wird. Bei einer dynamischen Auswahl, die sich nicht nur auf ein Kommunikationssystem beschränkt, können die jeweiligen ortsspezifischen Vorteile ausgenutzt werden und eine vollständige Abdeckung (mind. $80\,\%$) erreicht werden.

6.4 Analyse- und Optimierungsablauf

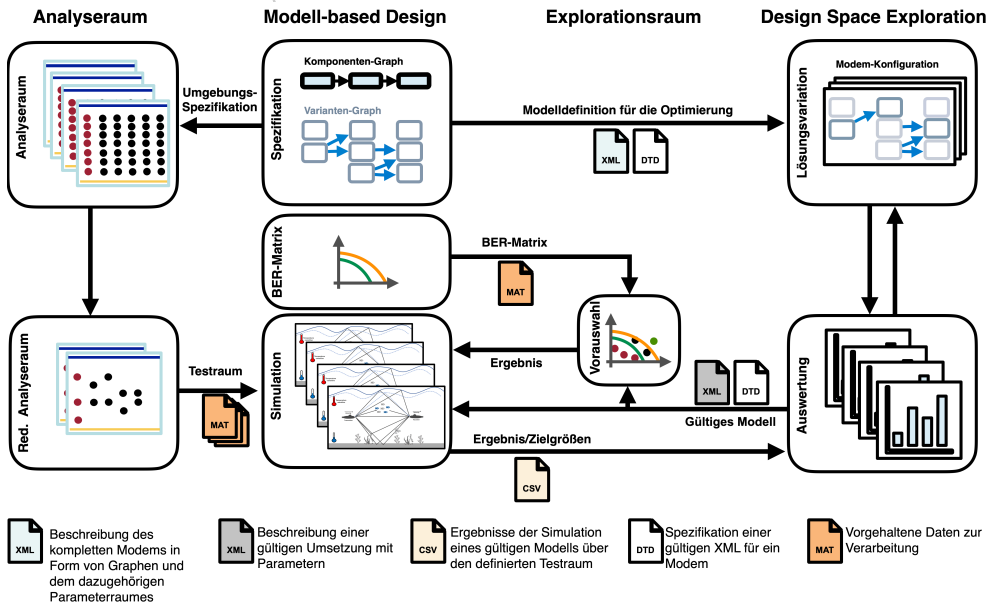


Abbildung 6.19: Analyseworkflow (schematisch)

Der resultierende Ansatz zur Entwicklung, Analyse und Optimierung eines Kommunikationssystem am Beispiel der akustischen Unterwasserkommunikation

über ein umgebungs- und anforderungsbasiertes Testkonzept ist schematisch in Abbildung 6.19 abgebildet und kann in sechs Phasen zusammengefasst werden:

Die erste Phase beinhaltet die Spezifikation der Anforderungen an das Kommunikationssystem und definiert Datenraten, Modulations- und Codierungsverfahren sowie der notwendigen Schnittstellen. Zusätzlich werden die physikalischen Eigenschaften und Einschränkungen des Kommunikationskanals im Unterwasserbereich, berücksichtigt. Diese Spezifikation bildet die Grundlage für die nachfolgenden Phasen und gewährleistet, dass das Kommunikationssystem den spezifischen Anforderungen der Anwendung und Umgebung entspricht.

In der zweiten Phase erfolgt die Erstellung der Referenzfunktionen und deren modellbasierte Implementierungen der Komponenten. Hierbei werden die Eigenschaften der Komponenten in Form von Varianten umgesetzt, die anschließend simuliert und gegen die Referenzfunktion und Anforderungen getestet werden. Der modellbasierte Ansatz ermöglicht es, Parameter und Varianten zu variieren, um die Leistungsfähigkeit der Kombinationen unter unterschiedlichen Bedingungen zu überprüfen. Durch den Einsatz von Simulationstools kann die Funktionalität der einzelnen Module bereits vor einer aufwändigen und kostenintensiven Hardware-Implementierung validiert werden. Erweitert um den Aspekt der Synthetisierbarkeit lassen sich die Varianten direkt in Hardware (Field Programmable Gate Array (FPGA)) realisiert und validiert werden.

Die dritte Phase konzentriert sich auf die Verifikation der Implementierungen durch die Generierung valider Varianten-Kombinationen und deren Testung über einen idealen Kanal. In dieser Phase werden die Varianten der Komponenten zusammengefasst zu einem Kommunikationssystem und unter idealisierten Bedingungen getestet, um ihre grundsätzliche Funktionsfähigkeit zu bestätigen. Die Verifikation stellt sicher, dass das System den spezifizierten Anforderungen entspricht. Zudem werden mögliche Fehlerquellen identifiziert und gegebenenfalls behoben.

Die vierte Phase umfasst die Definition und Generierung des Testraums. Dieser Testraum wird anhand von Umgebungsparametern und anwendungsspezifischen Daten, hier am Beispiel der akustischen Unterwasserkommunikation, erstellt. Dazu werden die Faktoren wie Umgebungsdruck, Temperatur, Salzgehalt und Strömungsverhältnisse in die Analyseraumgenerierung einbezogen, um realitätsnahe Testbedingungen zu schaffen. Die Generierung des Testraums ermöglicht es, die Leistung des Kommunikationssystems unter verschiedenen Umgebungsbedingungen und Anforderungen vergleichend zu evaluieren und stellt sicher, dass die Anforderungen erreicht werden können.

In der fünften Phase wird der Analyse- und Entwurfsraum durch die Einführung von Reduktionsansätzen optimiert. Diese Ansätze zielen darauf ab, die Anzahl der Testpunkte im Analyseraum und die Anzahl der möglichen Implementierungen im Erkundungsraum zu reduzieren, ohne die Aussagekraft der Evaluation zu beeinträchtigen. Durch den Einsatz von stochastischen Methoden und Expertenwissen werden die Testpunkte identifiziert die nicht simuliert werden müssen. Dies führt zu einer effizienteren Nutzung von Ressourcen und einer Verkürzung der Testzeiten, während gleichzeitig die Validität der Ergebnisse gewährleistet bleibt.

In der abschließenden sechsten Phase erfolgt die Optimierung und Selektion der besten Lösungen, die den Anforderungen entsprechen. Durch die Anwendung von Optimierungsalgorithmen werden die Parameter und Varianten-Kombinationen so angepasst, dass eine über die Zielgrößen definiertes optimales Ergebnis erzielt wird. Die Modems werden somit hinsichtlich ihrer Effizienz, Zuverlässigkeit und Robustheit bewertet. Damit wird sichergestellt, dass die dominierenden Kommunikationssystem nicht nur funktionsfähig, sondern auch optimal für die spezifischen Einsatzbedingungen gewählt und parametrisiert sind.

7

Zusammenfassung und Ausblick

Diese Dissertation widmet sich der modellbasierten Entwicklung eines akustischen Unterwasserkommunikationssystems, das rechnergestützt implementiert wurde. Die Arbeit umfasst die theoretische Konzeption eines modularen Kommunikationssystems, die Erstellung eines anwendungs- und ortsbezogenen Unterwasserkanals sowie die Verbesserung der Kommunikationsstrategie für RTS/CTS-basierte Protokolle in Unterwasserumgebungen. Zudem wird eine Methode zur Auswahl optimaler Lösungen für spezifische Anforderungen vorgestellt.Die wesentlichen wissenschaftlichen Beiträge können wie folgt zusammengefasst werden:

Modularer Ansatz für Kommunikationssysteme: Die Dissertation präsentiert ein modular aufgebautes Kommunikationssystem, das flexibel an verschiedene Anwendungen angepasst werden kann.

Geometrischer Ansatz zur Kanalschätzung: Der entwickelte geometrische Ansatz ermöglicht eine präzise und reproduzierbare Analyse des Unterwasserkanals, sowohl in Simulationen als auch auf Hardware.

Optimierung mittels automatisierter DSE: Die Einführung einer automatisierten Optimierung (Design Space Exploration (DSE)) mithilfe eines evolutionären Algorithmus ermöglicht es, analytische, geometrische und stochastische Ansätze effektiv zu kombinieren, wodurch der Analyse- und Explorationsraum reduziert wird.

Die Implementierung des Systems in MATLAB/Simulink und die empirische Validierung des erweiterten Modells ermöglichen eine optimale Auswahl des Systems für die Herausforderungen der Unterwasserumgebung. Das entwickelte System kann direkt auf programmierbaren Systemen wie Field Programmable Gate Array (FPGA) eingesetzt und in bestehende Systeme integriert werden.

7.1 Wissenschaftliche Methoden und Ansätze

Die Dissertation nutzt die modellbasierte Entwicklung eines umfassenden Kommunikationssystems am Beispiel der akustischen Unterwasserkommunikation. Durch die Anwendung von Simulationsmodellen und Varianten-Management-Ansätzen wurde ein systematisches Modell erstellt, das den Passband-Bereich der akustischen Kommunikation präzise erfasst. Diese Methodik ermöglicht eine effektive, wiederverwendbare, testbare und erweiterbare Systemarchitektur.

Robuste Kommunikationsmethoden: Zur Steigerung der Kommunikationsrobustheit wurde ein Kasami-basiertes Verfahren zur Rahmentypen-Detektion eingeführt. Kontrollrahmen wurden in den Coderaum der Präambel verschoben, um die Detektionsschwellen basierend auf Länge und Güte zu optimieren.

Vielfältige Anwendungsmöglichkeiten: Der entwickelte Workflow ermöglicht einen analytischen und leistungsbasierten Vergleich von Unterwasserkommunikationssystemen in spezifischen Anwendungs- und Umgebungssettings. Umfangreiche numerische Simulationen analysieren die Ausbreitungscharakteristika akustischer Signale und bestimmen optimale Systemparameter. Dieser Workflow ist nicht nur auf Unterwasserkommunikation beschränkt, sondern kann auch auf Hochgeschwindigkeitssysteme, Weltraumkommunikation und urbane mobile Kommunikation angewendet werden, insbesondere in herausfordernden oder sich stetig ändernden Kanälen.

Energieeffiziente und umweltfreundliche Kommunikation: Die vorgenommenen Optimierungen ermöglichen eine nachhaltige Nutzung von Energiereserven und schonen Flora und Fauna durch grüne Kommunikation, auch im Bereich der akustischen Unterwasserkommunikation.

7.2 Limitationen

Trotz der erfolgreichen Umsetzung gibt es einige Limitationen, die in dieser Dissertation festgestellt wurden:

Abstrakte Modelle: Die Simulationen basieren auf Modellen der Unterwasserumgebung, die nicht alle realen Bedingungen vollständig abbilden. Variablen wie Strömungsverhältnisse, Temperaturgradienten und biologisches Rauschen wurden berücksichtigt, jedoch könnte eine spezifischere Analyse der Auswirkungen dieser Parameter auf die Systemperformance die Validität der Ergebnisse weiter verbessern.

Eingeschränkte Empirie: Die empirischen Simulationstests wurden in spezifischen Gewässern durchgeführt, die nicht alle möglichen Einsatzszenarien abdecken und deren Topologie vereinfacht wurde. Unterschiede in den geografischen Besonderheiten könnten die Systemleistung zusätzlich beeinflussen und sollten gesondert betrachtet werden.

7.3 Zusammenfassung und Ausblick

Zusammenfassend wurde ein leistungsfähiges und zuverlässiges Unterwasserkommunikationssystem entwickelt, das die spezifischen Herausforderungen der Unterwasserumgebung adressiert. Die Dissertation zeigt die Wirksamkeit der verwendeten wissenschaftlichen Methoden und bietet eine solide Grundlage für weiterführende Forschung:

Modellverfeinerung und Optimierungsprozess: Zukünftige Arbeiten sollten darauf abzielen, die Modelle und den Optimierungsprozess weiter zu verfeinern. Insbesondere Feldtests zur Validierung der Optimierung und der Kanalmodelle sind erforderlich, um eine breitere Vielfalt von Unterwasserumgebungen abzudecken und die Validität der Ergebnisse zu gewährleisten.

Erweiterung der Forschungsbereiche: Der entwickelte Workflow eröffnet Möglichkeiten zur Erschließung neuer Forschungsbereiche. Die resultierenden Metadaten ermöglichen die Bildung von Kommunikationsclustern, die eine adaptive Änderung und Auswahl von Hardware und Funktionalität in Abhängigkeit von Umgebung und Anforderungen erlauben.

Standards für Systemtest und Bewertung: Ein weiterer Forschungsbereich ist die Entwicklung von Standards für die Testung, Bewertung und den Vergleich von Unterwasserkommunikationssystemen, abhängig von Umgebung, Missionsauftrag und Anwendungsanforderungen. Dies ermöglicht objektive Vergleiche und reduziert das Risiko kostspieliger Fehlschläge.

Optimierung der Kommunikationsprotokolle: Es besteht weiterhin Potenzial zur Verbesserung der Zuverlässigkeit und Störresistenz von Kommunikationsprotokollen. Das Kasami-basierte Konzept bietet vielversprechende Ansätze, insbesondere zur Kollisionserkennung in Mehrnutzerszenarien mit präambelbasierten Strategien.

Die umfassende Darstellung der entwickelten Methoden und Ergebnisse sowie der Ausblick auf zukünftige Forschungsrichtungen unterstreicht den wissenschaftlichen Beitrag dieser Dissertation und deren Relevanz für die Weiterentwicklung der Unterwasserkommunikationstechnologie. Die vorgestellten Methoden sind

nicht nur für die Unterwasserkommunikation relevant, sondern könnten auch wertvolle Erkenntnisse für andere Disziplinen wie Meeresbiologie, Umweltüberwachung und Weltraumforschung liefern. Die Methoden können zur Verbesserung der Kommunikationstechnologie in verschiedenen anspruchsvollen Umgebungen beitragen.

A

Anhang

A.1 Umgebungen

Auflistung aller Quellcodes für die Definition der genutzten Umgebungen (**.env*). Jeweils die Umgebung a und b sowie c bis f haben ein identisches Geschwindigkeitsprofil (SSP) und die Umgebung e und f sind komplett identisch und unterscheiden sich nur durch die genutzte Bodendefinition (**.bty*). Die Umgebungen a und b Verfügen über ein flache spiegel-ähnliche Wasseroberfläche ohne Einflussnahme von Wellen verursacht durch Wind und Wetter. Die Umgebungen b bis f nutzen eine Identische Oberflächenbeschreibung auf Basis des Seeganges der Stufe 3. Mit Ausnahme der Umgebung e und f verfügen alle Umgebungen über einen flachen Boden mit den in der jeweiligen Eigenschaften über die Umgebungsbeschreibung definierten Eigenschaften.

Listing A.1: Umgebungdefinition a

```
'Arctic profile SOFAR',  ! TITLE
200.0                    ! FREQ (Hz)
1                     ! NMEDIA
'CVWT'                   ! SSPOPT (Analytic or C-linear interpolation)
51  0.0  3750.0            ! DEPTH of bottom (m)
   0.0  1436.0 /
  200.0  1458.4 /
  300.0  1460.5 /
  1000.0  1466.7 /
  2000.0  1479.6 /
  2500.0  1487.9 /
  3750.0  1510.4 /
'A'  0.0
3750  1507  0  1.49  0.5  /
1                     ! NSD
100  /                   ! SD(1:NSD)  (m)
761                   ! NRD
0  3750  /                ! RD(1:NRD)  (m)
1001                  ! NR
0  100  /                ! R(1:NR )  (km)
'CB'                  ! Run type:  'Ray/Coh/Inc/Sem'
5000                  ! NBEAMS IBEAM
-16.7  16.7  /             ! ALPHA1,2  (degrees)
0  3800  101              ! STEP  (m),  ZBOX  (m),  RBOX  (km)
'MS'  1.0  100.0            ! 'Min/Fill/Cer,  Sin/Doub/Zero', EPMULT
```

Listing A.2: Umgebungdefinition b

```
'Arctic profile',   ! TITLE
200.0           ! FREQ (Hz)
1             ! NMEDIA
'CVWT'          ! SSPOPT (Analytic or C-linear interpolation)
51 0.0 3750.0     ! DEPTH of bottom (m)
  0.0 1436.0 /
 200.0 1458.4 /
 300.0 1460.5 /
 1000.0 1466.7 /
 2000.0 1479.6 /
 2500.0 1487.9 /
 3750.0 1510.4 /
'A' 0.0
3750 1507 0.0 1.49 0.5 /
1             ! NSD
1000.0 /          ! SD(1:NSD) (m)
761           ! NRD
0.0 3750.0 /      ! RD(1:NRD) (m)
1001          ! NR
0.0 100.0 /       ! R(1:NR ) (km)
'CB'          ! Run type: 'Ray/Coh/Inc/Sem'
5000          ! NBEAMS IBEAM
-16.7 16.7 /      ! ALPHA1,2 (degrees)
0.0 3800.0 101.0   ! STEP (m), ZBOX (m), RBOX (km)
'MS' 1.0 100.0     ! 'Min/Fill/Cer, Sin/Doub/Zero', EPMULT
```

Listing A.3: Umgebungdefinition c

```
'Arctic profile',   ! TITLE
10000.0         ! FREQ (Hz)
1             ! NMEDIA
'CVWT'          ! SSPOPT (Analytic or C-linear interpolation)
51 0 494.6 0 1481.89 7 1483.03 12 1487.4 17 1491.35 22 1490.65 32 1490.84
37 1489.51 47 1486.18 52 1485.32 57 1484.71 67 1483.62 77 1483.25 87 1484.11
92 1484.18 107 1484.28 112 1484.3 122 1484.42 127 1484.45 132 1484.52 137 1484.61
147 1484.79 157 1484.95 167 1485.11 177 1485.26 182 1485.33 192 1485.33 197 1485.32
207 1485.48 217 1485.65 222 1485.74 227 1485.82 232 1485.91 237 1485.99 252 1486.25
267 1486.52 277 1486.69 287 1486.88 297 1487.05 302 1487.13 312 1487.3 317 1487.38
327 1487.54 337 1487.71 347 1487.88 352 1487.97 362 1488.12 367 1488.21 377 1488.37
387 1488.54 392 1488.62 397 1488.71 407 1488.88 417 1489.04 427 1489.21 432 1489.30
442 1489.46 457 1489.71 462 1489.80 467 1489.89 477 1490.06 492 1490.32 494.6 1490.46
'A' 0.0
495 1507 0 1.49 0.5 /
1             ! NSD
150.0 /           ! SD(1:NSD) (m)
761           ! NRD
0.0 495 /         ! RD(1:NRD) (m)
1001                                ! NR
0.0 6.0 /       ! R(1:NR ) (km)
'CB'          ! Run type: 'Ray/Coh/Inc/Sem'
5000          ! NBEAMS
-16.7 16.7 /      ! ALPHA1,2 (degrees)
0.0 495.0 6.1     ! STEP (m), ZBOX (m), RBOX (km)
'MS' 1 6        ! 'Min/Fill/Cer, Sin/Doub/Zero', EPMULT
```

Listing A.4: Umgebungdefinition d

```
'Shallow water omni'! TITLE
10000.0          ! FREQ (Hz)
1              ! NMEDIA
'CVWT'           ! SSPOPT (Analytic or C-linear interpolation)
51 0 494.6          0 1481.89 7 1483.03      12 1487.4        17 1491.35       22 1490.65
 32 1490.84 37 1489.51    47 1486.18       52 1485.32       57 1484.71       67 1483.62
 77 1483.25       87 1484.11       92 1484.18       107 1484.28      112 1484.3       122 1484.42
127 1484.45      132 1484.52      137 1484.61      147 1484.79      157 1484.95      167 1485.11
177 1485.26      182 1485.33      192 1485.33      197 1485.32      207 1485.48      217 1485.65
222 1485.74      227 1485.82      232 1485.91      237 1485.99      252 1486.25      267 1486.52
277 1486.69      287 1486.88      297 1487.05      302 1487.13      312 1487.3       317 1487.38
327 1487.54      337 1487.71      347 1487.88      352 1487.97       362 1488.12     367 1488.21
377 1488.37      387 1488.54      392 1488.62      397 1488.71      407 1488.88      417 1489.04
427 1489.21      432 1489.30      442 1489.46      457 1489.71      462 1489.80      467 1489.89
477 1490.06      492 1490.32      494.6 1490.46    'A' 0.0
495 1507 0 1.49 0.5 /
1              ! NSD
150.0 /          ! SD(1:NSD) (m)
761            ! NRD
0.0 495 /        ! RD(1:NRD) (m)
1001           ! NR
0.0 6.0 /        ! R(1:NR ) (km)
'CB'           ! Run type: 'Ray/Coh/Inc/Sem'
5000           ! NBEAMS
-180 180 /       ! ALPHA1,2 (degrees)
0.0 495.0 6.1    ! STEP (m), ZBOX (m), RBOX (km)
'MS' 1 6         ! 'Min/Fill/Cer, Sin/Doub/Zero', EPMULT
```

Listing A.5: Umgebungdefinition e/f

```
'Shallow water with inc'! TITLE
10000.0            ! FREQ (Hz)
1                ! NMEDIA
'CVWT'             ! SSPOPT (Analytic or C-linear interpolation)
51 0.0 494.6         ! DEPTH of bottom (m)
 0 1481.89         7 1483.03         12 1487.4         17 1491.35        22 1490.65
 32 1490.84        37 1489.51        47 1486.18        52 1485.32        57 1484.71        67 1483.62
 77 1483.25        87 1484.11        92 1484.18       107 1484.28       112 1484.3        122 1484.42
127 1484.45       132 1484.52       137 1484.61       147 1484.79       157 1484.95       167 1485.11
177 1485.26       182 1485.33       192 1485.33       197 1485.32       207 1485.48       217 1485.65
222 1485.74       227 1485.82       232 1485.91       237 1485.99       252 1486.25       267 1486.52
277 1486.69       287 1486.88       297 1487.05       302 1487.13       312 1487.3        317 1487.38
327 1487.54       337 1487.71       347 1487.88       352 1487.97       362 1488.12       367 1488.21
377 1488.37       387 1488.54       392 1488.62       397 1488.71       407 1488.88       417 1489.04
427 1489.21       432 1489.30       442 1489.46       457 1489.71       462 1489.80       467 1489.89
477 1490.06       492 1490.32       494.6 1490.46
'A*' 0.0
494.6 1507 0 1.49 0.5 /
1              ! NSD
150 /            ! SD(1:NSD) (m)
761            ! NRD
0.0 494.6 /        ! RD(1:NRD) (m)
1001           ! NR
0.0 6.0 /        ! R(1:NR) (km)
'CB'           ! 'R/C/I/S'
4001 10          ! NBEAMS
-16.7 16.7 /       ! ALPHA1,2 (degrees)
0.0 495 7.0      ! STEP (m), ZBOX (m), RBOX (km)
```

A.2 Bodenbeschreibung

Listing A.6: Bodendefinition e

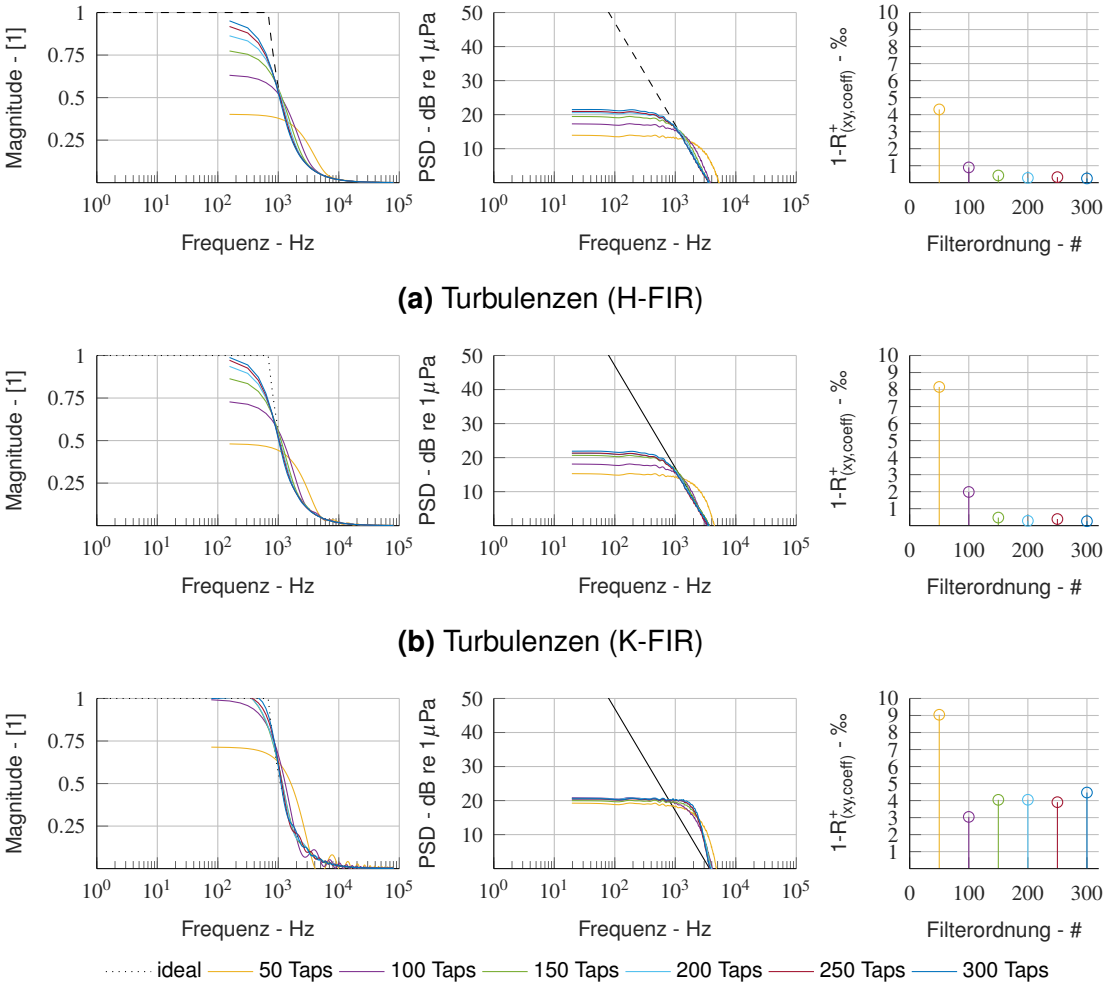
```
'L'
61
0.0  494.0
0.1  493.91
0.2  493.64
0.3  493.19
0.4  492.56
0.5  491.75
0.6  490.76
0.7  489.59
0.8  488.24
0.9  486.71
1.0  485.0
1.1  483.11
1.2  481.04
1.3  478.79
1.4  476.36
1.5  473.75
1.6  470.96
1.7  467.99
1.8  464.84
1.9  461.51
2.0  458.0
2.1  454.31
2.2  450.44
2.3  446.39
2.4  442.16
2.5  437.75
2.6  433.16
2.7  428.39
2.8  423.44
2.9  418.31
3.0  413.0
3.1  407.51
3.2  401.84
3.3  395.99
3.4  389.96
3.5  383.75
3.6  377.36
3.7  370.79
3.8  364.04
3.9  357.11
4.0  350.0
4.1  342.71
4.2  335.24
4.3  327.59
4.4  319.76
4.5  311.75
4.6  303.56
4.7  295.19
4.8  286.64
4.9  277.91
5.0  269.0
5.1  259.91
5.2  250.64
5.3  241.19
5.4  231.56
5.5  221.75
5.6  211.76
5.7  201.59
5.8  191.24
5.9  180.71
6.0  170.0
```

Listing A.7: Bodendefinition f

```
'L'
61
0.0    170.0
0.1    180.71
0.2    191.24
0.3    201.59
0.4    211.76
0.5    221.75
0.6    231.56
0.7    241.19
0.8    250.64
0.9    259.91
1.0    269.0
1.1    277.91
1.2    286.64
1.3    295.19
1.4    303.56
1.5    311.75
1.6    319.76
1.7    327.59
1.8    335.24
1.9    342.71
2.0    350.0
2.1    357.11
2.2    364.04
2.3    370.79
2.4    377.36
2.5    383.75
2.6    389.96
2.7    395.99
2.8    401.84
2.9    407.51
3.0    413.0
3.1    418.31
3.2    423.44
3.3    428.39
3.4    433.16
3.5    437.75
3.6    442.16
3.7    446.39
3.8    450.44
3.9    454.31
4.0    458.0
4.1    461.51
4.2    464.84
4.3    467.99
4.4    470.96
4.5    473.75
4.6    476.36
4.7    478.79
4.8    481.04
4.9    483.11
5.0    485.0
5.1    486.71
5.2    488.24
5.3    489.59
5.4    490.76
5.5    491.75
5.6    492.56
5.7    493.19
5.8    493.64
5.9    493.91
6.0    494.0
```

A.3 Ergebnisse der FIR-Filter-Ansätze

A.3.1 Rauschen von Turbulenzen



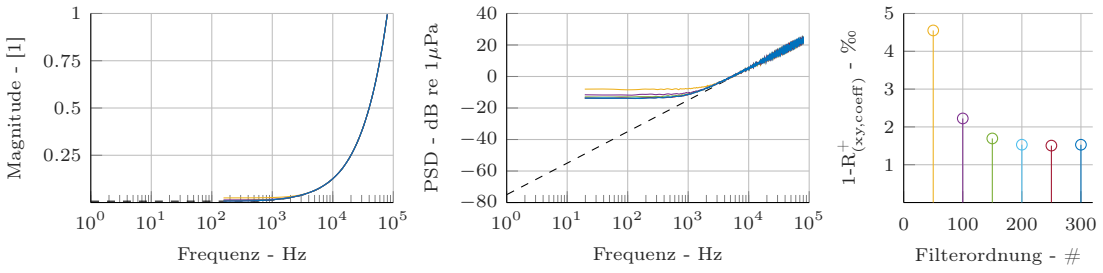
(a) Turbulenzen (H-FIR)

(b) Turbulenzen (K-FIR)

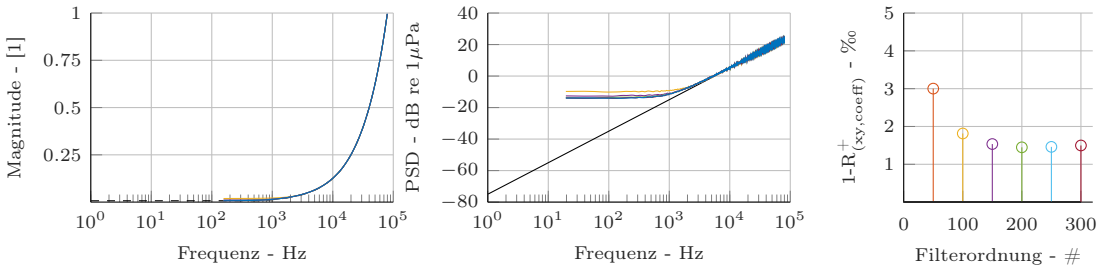
(c) Turbulenzen (LS-FIR)

Abbildung A.1: Frequenzantwort und Ähnlichkeitsmaß für das Rauschen von Turbulenzen der FIR-Filter-Ansätze im Vergleich

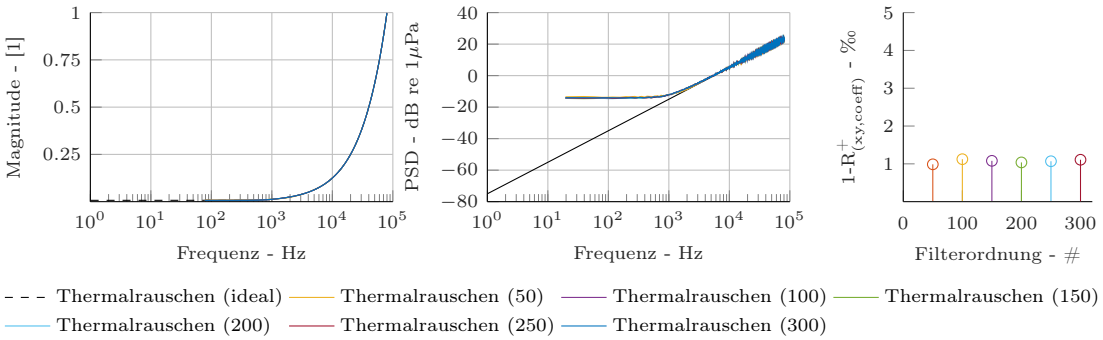
A.3.2 Thermalrauschen



(a) Thermalrauschen (H-FIR)



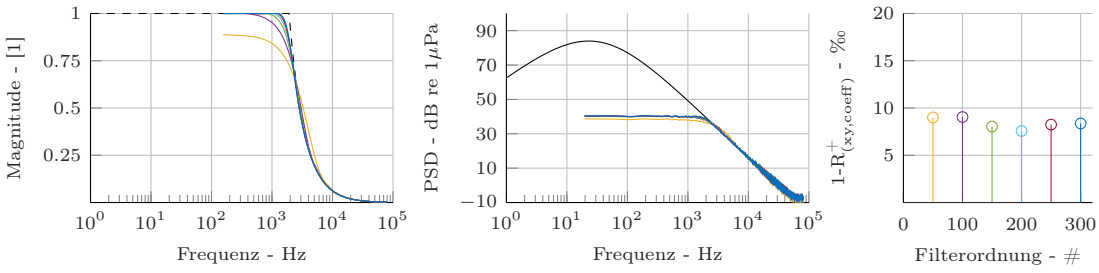
(b) Thermalrauschen (K-FIR)



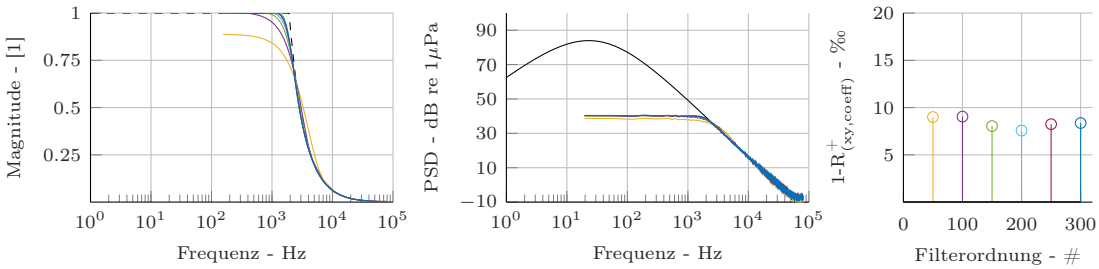
(c) Thermalrauschen (LS-FIR)

Abbildung A.2: Frequenzantwort und Ähnlichkeitsmaß für das Thermalrauschen der FIR-Filter-Ansätze im Vergleich

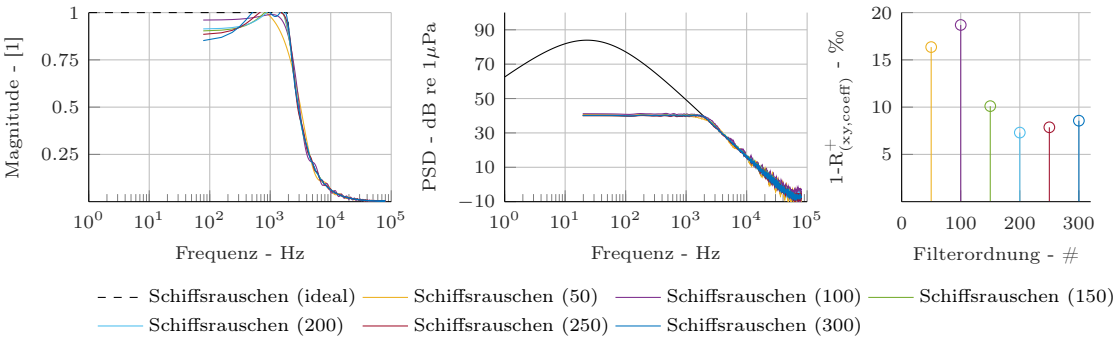
A.3.3 Verkehrsrauschen



(a) Verkehrsrauschen (H-FIR)



(b) Verkehrsrauschen (K-FIR)



(c) Verkehrsrauschen (LS-FIR)

Abbildung A.3: Frequenzantwort und Ähnlichkeitsmaß für das Verkehrsrauschen der FIR-Filter-Ansätze im Vergleich

A.4 Ergänzungen zur Rahmenerkennung

Die Polynome basieren auf der Polynome-Tabelle von Petersons [186] und den reziproken nach Ziemer und Peterson [185] auch primitiven und irreduzible, Polynomen zur Erzeugung der Pseudo Noise (PN)-Folgen mit Schieberegistern.

#	Polynom	#	Polynom	#	Polynom
201	x^2+x+1	701	x^7+x+1	801	$x^8+x^4+x^3+x+1$
		702	x^7+x^3+1	802	
301	x^3+x^2+1	703	$x^7+x^3+x^2+x+1$	803	$x^8+x^5+x^3+x+1$
302	x^3+x+1	704	x^7+x^4+1	804	$x^8+x^5+x^3+x^2+1$
		705	$x^7+x^4+x^3+x^2+1$	805	$x^8+x^5+x^4+x^3+1$
401	x^4+x^3+1	706	$x^7+x^5+x^2+x+1$	806	
402	x^4+x+1	707	$x^7+x^5+x^3+x+1$	807	$x^8+x^6+x^3+x^2+1$
403	$x^4+x^3+x^2+x+1$	708	$x^7+x^5+x^4+x^3+1$	808	$x^8+x^6+x^4+x^3+x^2+x+1$
		709	$x^7+x^5+x^4+x^3+x^2+x+1$	809	$x^8+x^6+x^5+x+1$
501	x^5+x^2+1	710	x^7+x^6+1	810	$x^8+x^6+x^5+x^2+1$
502	x^5+x^3+1	711	$x^7+x^6+x^3+x+1$	811	$x^8+x^6+x^5+x^3+1$
503	$x^5+x^3+x^2+x+1$	712	$x^7+x^6+x^4+x+1$	812	$x^8+x^6+x^5+x^4+1$
504	x^5+x^4+1	713	$x^7+x^6+x^4+x^2+1$	813	$x^8+x^6+x^5+x^4+x^2+x+1$
505	$x^5+x^4+x^2+x+1$	714	$x^7+x^6+x^5+x^2+1$	814	$x^8+x^6+x^5+x^4+x^3+x+1$
506	$x^5+x^4+x^3+x^2+1$	715	$x^7+x^6+x^5+x^3+x^2+x+1$	815	$x^8+x^7+x^2+x+1$
		716	$x^7+x^6+x^5+x^4+1$	816	$x^8+x^7+x^3+x+1$
601	x^6+x^5+1	717	$x^7+x^6+x^5+x^4+x^2+x+1$	817	$x^8+x^7+x^3+x^2+1$
602	$x^6+x^5+x^4+x+1$	718	$x^7+x^6+x^5+x^4+x^3+x^2+1$	818	$x^8+x^7+x^4+x^3+x^2+x+1$
603	$x^6+x^4+x^3+x+1$			819	$x^8+x^7+x^5+x+1$
604	x^6+x+1			820	$x^8+x^7+x^5+x^3+1$
605	$x^6+x^5+x^2+x+1$			821	$x^8+x^7+x^5+x^4+1$
606	$x^6+x^5+x^3+x^2+1$			822	$x^8+x^7+x^5+x^4+x^3+x^2+1$
				823	$x^8+x^7+x^6+x+1$
				824	$x^8+x^7+x^6+x^3+x^2+x+1$
				825	$x^8+x^7+x^6+x^4+x^2+x+1$
				826	$x^8+x^7+x^6+x^4+x^3+x^2+1$
				827	$x^8+x^7+x^6+x^5+x^2+x+1$
				828	$x^8+x^7+x^6+x^5+x^4+x+1$
				829	$x^8+x^7+x^6+x^5+x^4+x^2+1$
				830	$x^8+x^7+x^6+x^5+x^4+x^3+1$

Tabelle A.1: Irreduzible Polynome für Ordnungen $n := \{2,3,...,8\}$

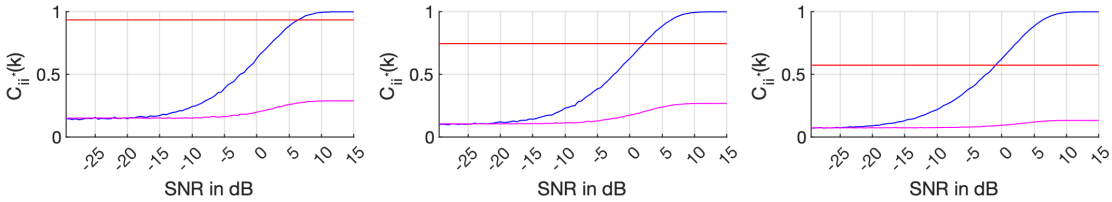
Ordnung n	m-Folge 1	m-Folge 2
5	$x^5+x^4+x^3+x^2+1$	x^5+x^2+1
6	$x^6+x^5+x^2+x+1$	x^6+x+1
7	$x^7+x^3+x^2+x+1$ x^7+x^3+x+1 $x^7+x^5+x^4+x^3+x^2+x+1$	x^7+x^3+1 x^7+x+1 $x^7+x^3+x^2+x+1$

Tabelle A.2: Bevorzugte Polynom-Paare für die Erzeugung von Gold-Folgen

Nr.	Initialwerte
401	1,3,4,6,8,10,13,15
402	8,9,10,11,12,13,14,15
403	8,9,10,11,12,13,14,15
501	1,3,4,6,9,11,12,14,16,18,21,23,24,26,29,31
502	2,3,4,5,10,11,12,13,16,17,22,23,24,25,30,31
503	4,5,6,7,8,9,10,11,16,17,18,19,28,29,30,31
504	1,2,5,6,9,10,13,14,16,19,20,23,24,27,28,31
505	1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31
506	2,3,6,7,8,9,12,13,16,17,20,21,26,27,30,31
601	1,3,5,7,9,11,13,15,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,49,51,53,55,57,59,61,63
602	8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,32,33,34,35,36,37,38,39,56,57,58,59,60,61,62,63
603	4,5,6,7,8,9,10,11,20,21,22,23,24,25,26,27,32,33,34,35,44,45,46,47,48,49,50,51,60,61,62,63
604	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
605	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63
606	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63
701	1,3,5,7,8,10,12,14,17,19,21,23,24,26,28,30,33,35,37,39,40,42,44,46,49,51,53,55,56,58,60,62,64,66,68,70,73,75,77,79,80,82,84,86,89,91,93,95,96,98 100,102,105,107,109,111,112,114,116,118,121,123,125,127
702	1,3,5,7,8,10,12,14,16,18,20,22,25,27,29,31,32,34,36,38,41,43,45,47,49,51,53,55,56,58,60,62,64,66,68,70,73,75,77,79,81,83,85,87,88,90,92,94,97,99 101,103,104,106,108,110,112,114,116,118,121,123,125,127
703	1,3,4,6,8,10,13,15,16,18,21,23,25,27,28,30,33,35,36,38,40,42,45,47,48,50,53,55,57,59,60,62,64,66,69,71,73,75,76,78,81,83,84,86,88,90,93,95,96,98 101,103,105,107,108,110,113,115,116,118,120,122,125,127
704	2,3,4,5,10,11,12,13,16,17,22,23,24,25,30,31,32,33,38,39,40,41,46,47,50,51,52,53,58,59,60,61,64,65,70,71,72,73,78,79,82,83,84,85,90,91,92,93,98,99 100,101,106,107,108,109,112,113,118,119,120,121,126,127
705	1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31,32,35,37,38,41,42,44,47,49,50,52,55,56,59,61,62,64,67,69,70,73,74,76,79,81,82,84,87,88,91,93,94,97,98 100,103,104,107,109,110,112,115,117,118,121,122,124,127
706	4,5,6,7,12,13,14,15,16,17,18,19,24,25,26,27,36,37,38,39,44,45,46,47,48,49,50,51,56,57,58,59,64,65,66,67,72,73,74,75,84,85,86,87,92,93,94,95,96,97 98,99,104,105,106,107,116,117,118,119,124,125,126,127
707	1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,88,90,92,94,97,99 101,103,105,107,109,111,113,115,117,119,121,123,125,127
708	8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,48,49,50,51,52,53,54,55,64,65,66,67,68,69,70,71,80,81,82,83,84,85,86,87,104 105,106,107,108,109,110,111,120,121,122,123,124,125,126,127
709	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,34,35,38,39,42,43,46,47,48,49,52,53,56,57,60,61,64,65,68,,69,72,73,76,77,82,83,86,87,90,91,94,95,96,97 100,101,104,105,108,109,114,115,118,119,122,123,126,127
710	1,3,4,6,9,11,12,14,17,19,20,22,25,27,28,30,33,35,36,38,41,43,44,46,49,51,52,54,57,59,60,62,64,66,69,71,72,74,77,79,80,82,85,87,88,90,93,95,96,98 101,103,104,106,109,111,112,114,117,119,120,122,125,127
711	2,3,4,5,10,11,12,13,18,19,20,21,26,27,28,29,34,35,36,37,42,43,44,45,50,51,52,53,58,59,60,61,64,65,70,71,72,73,78,79,80,81,86,87,88,89,94,95,96,97 102,103,104,105,110,111,112,113,118,119,120,121,126,127
712	1,2,4,7,8,11,13,14,17,18,20,23,24,27,29,30,33,34,36,39,40,43,45,46,49,50,52,55,56,59,61,62,64,67,69,70,73,74,76,79,80,83,85,86,89,90,92,95,96,99 101,102,105,106,108,111,112,115,117,118,121,122,124,127
713	2,3,6,7,8,9,12,13,16,17,20,21,26,27,30,31,32,33,36,37,42,43,46,47,50,51,54,55,56,57,60,61,64,65,68,69,74,75,78,79,82,83,86,87,88,89,92,93,98,99 102,103,104,105,108,109,112,113,116,117,122,123,126,127
714	2,3,4,5,8,9,14,15,16,17,22,23,26,27,28,29,34,35,36,37,40,41,46,47,48,49,54,55,58,59,60,61,64,65,70,71,74,75,76,77,82,83,84,85,88,89,94,95,96,97 102,103,106,107,108,109,114,115,116,117,120,121,126,127
715	1,2,5,6,8,11,12,15,17,18,21,22,24,27,28,31,32,35,36,39,41,42,45,46,48,51,52,55,57,58,61,62,64,67,68,71,73,74,77,78,80,83,84,87,89,90,93,94,97,98 101,102,104,107,108,111,113,114,117,118,120,123,124,127
716	64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96 97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127
717	4,5,6,7,12,13,14,15,20,21,22,23,28,29,30,31,32,33,34,35,40,41,42,43,48,49,50,51,56,57,58,59,64,65,66,67,72,73,74,75,80,81,82,83,88,89,90,91 100,101,102,103,108,109,110,111,116,117,118,119,124,125,126,127
718	1,2,5,6,9,10,13,14,16,19,20,23,24,27,28,31,32,35,36,39,40,43,44,47,49,50,53,54,57,58,61,62,64,67,68,71,72,75,76,79,81,82,85,86,89,90,93,94,97,98 101,102,105,106,109,110,112,115,116,119,120,123,124,127
801	1,3,5,7,9,11,13,15,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,49,51,53,55,57,59,61,63,
802	8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,32,33,34,35,36,37,38,39,56,57,58,59,60,61,62,63,
803	4,5,6,7,8,9,10,11,20,21,22,23,24,25,26,27,32,33,34,35,44,45,46,47,48,49,50,51,60,61,62,63,
804	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
805	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
806	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
807	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
808	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
809	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
810	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
811	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
812	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
813	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
814	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
815	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
816	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
817	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
818	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
819	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
820	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
821	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
822	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
823	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
824	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
825	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
826	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
827	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,
828	32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,
829	2,3,6,7,10,11,14,15,16,17,20,21,24,25,28,29,32,33,36,37,40,41,44,45,50,51,54,55,58,59,62,63,
830	1,2,4,7,9,10,12,15,16,19,21,22,24,27,29,30,32,35,37,38,40,43,45,46,49,50,52,55,57,58,60,63,

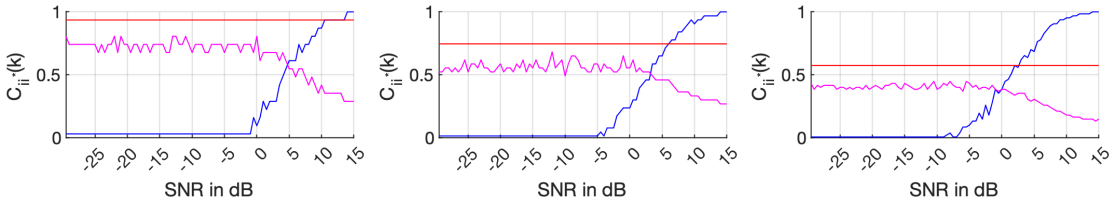
Tabelle A.3: Berechnete Initalworte der Linear-Feedback Shift Register (LFSR) zur Erzeugung ausgeglichener Maximalfolgen (MF)

A.5 Störanfälligkeit von Gold-Folgen (GF)



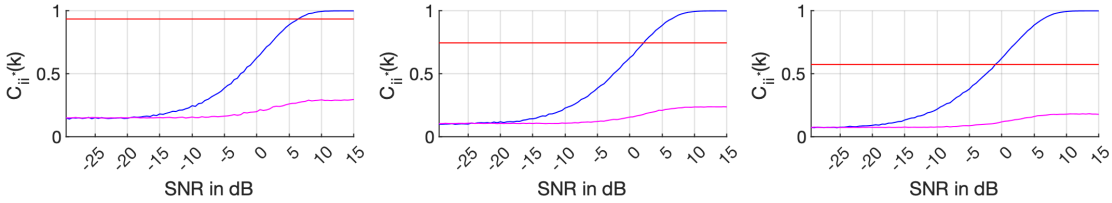
(a) Gold-Folgen (GF) (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.4: Durchschnittliche periodische Autokorrelation der GF



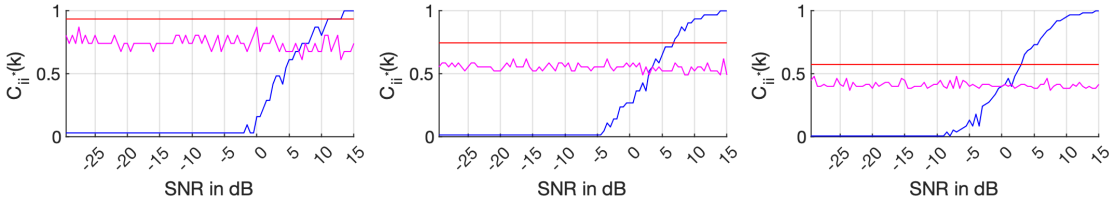
(a) GF (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.5: Maximum der periodische Autokorrelation der GF



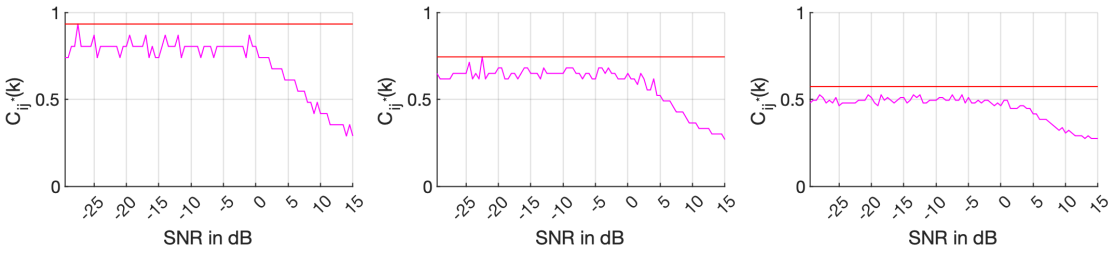
(a) GF (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.6: Durchschnittliche aperiodische Autokorrelation der GF



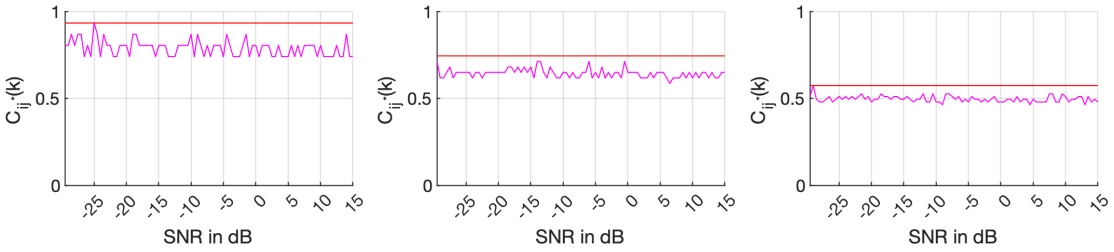
(a) GF (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.7: Maximum der Autokorrelation der GF



(a) GF (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.8: Maximum der periodische Kreuzkorrelation der GF



(a) GF (31 bit) **(b)** GF (63 bit) **(c)** GF (127 bit)

Abbildung A.9: Maximum der aperiodische Kreuzkorrelation der GF

A.6 Kanalgenerierung

Listing A.8: Berechnung aller definierten Kanalimpulseantwort (CIR)-Matrixen für definierte Testset

```
function generateChannel(modSSP, profilSSP, channeLocationRoot, channelLocationOutput, fast, rebuild, seastate_a)
      % Author: Marcel Riess
      % Description: This function simulates the underwater channel for a given
      % sound speed profile (SSP) using various simulation methods. It generates
      % the channel impulse response considering surface and bottom reflections
      % and saves them in .mat files accordingly.
      %
      % Input Parameters:
      %  - modSSP: SSP model ('CDT' or 'GEBCO')
      %  - profilSSP: Name of the SSP profile file
      %  - channeLocationRoot: Root directory of the channel storage location
      %  - channelLocationOutput: Output subdirectory for channel simulation results
      %
      % Output Parameters: None
      %
      % ASSUMPTIONS AND CONSTRAINTS:
      %  Data is saved in the form of a .mat file

      % Global variables
      global time_diff transmitAngleArray transmitAngleArrayLoss  ht_bnd_a hr_bnd_a d_bnd_a h_bnd_a matrix_max
      Abtastrate diff_d0 bottomType_a Carrierfreq_a BandWidth_a time_axis densitySurface densityBottom cBottom title
      %#ok<GVMIS>
      % Display options
      display = true; % Output information
      calculateAddionalAnalsis = false; % Compute target objects for comparison of CIR methods

      % Change to channel root directory
      actpath=pwd;
      if ismac
        path(genpath(['..' filesep '..' filesep '..' filesep 'core' filesep 'ChannelGeneration' filesep 'at_mac']),
      path)
      else
        path(genpath(['..' filesep '..' filesep '..' filesep 'core' filesep 'ChannelGeneration' filesep 'at']), path
      )
      end
      cd([channeLocationRoot filesep 'CTD'] );

      % Load SSP
      switch (modSSP)
        case 'CDT'
          ctd=strcat(profilSSP, '.cnv');
          [SSP_z,SSP]= readcnv(ctd);
        case 'GEBCO'
          ctd=strcat(profilSSP, '.txt');
          [SSP, SSP_z] = readSpeedAndDepth(ctd);
        otherwise
      end

      % Grid-Daten der Testpunkte des Kanals
      h0=ceil(max(SSP_z));% Wassertiefe [m]
      ht0_a=AUVModem.d_z(AUVModem.d_z<h0);% Sendertiefeen [m]conturf
      hr0_a=AUVModem.d_z(AUVModem.d_z<h0);% Empf'e4ngertiefen [m]
      d0_a=(100:diff_d0:3000);% Kommunikationsreichweiten[m]

      % Definiere m'f6gliche Sendergeschwindigkeiten
      % Hinzufuegen von Abweichungen der Position 'fcber die Zeit
      % Initialisierung
      ht_bnd_matrix = zeros(matrix_max, 1); % Initialisierung der Koordinaten
      hr_bnd_matrix = zeros(matrix_max, 1);
      h_bnd_matrix = zeros(matrix_max, 1);
      v_Tx_a = zeros(matrix_max, 1);
      v_Rx_a = zeros(matrix_max, 1);
      v_Ty_a = zeros(matrix_max, 1);
      v_Ry_a = zeros(matrix_max, 1);
      % Zuf'e4llige Startkoordinaten
      ht_bnd_matrix(1) = 0;
      hr_bnd_matrix(1) = 0;
      h_bnd_matrix(1) = 0;
      d_bnd_matrix(1)=0;
      % Schleife 'fcber die Zeit
      for timeCount = 2:matrix_max
        max_step_x = time_diff*(1); % Maximale Schrittweite der 'c4nderung pro Zeitschritt
        max_step_y = time_diff*0.5; % Maximale Schrittweite der 'c4nderung pro Zeitschritt

```

```
71          % Zuf'e4llige 'c4nderung berechnen, begrenzt durch maximale Schrittweite
72          delta_Ty = ceil(randi([ceil(-max_step_y*10) , ceil(max_step_y*10) ])./10);
73          delta_Ry = ceil(randi([ceil(-max_step_y*10) , ceil(max_step_y*10) ])./10);
74          delta_Tx = ceil(randi([ceil(-max_step_x*10) , ceil(max_step_x*10) ])./10);
75          delta_Rx = ceil(randi([ceil(-max_step_x*10) , ceil(max_step_x*10) ])./10);
76          % Neue Koordinate berechnen und sicherstellen, dass sie im Bereich liegt
77          ht_bnd_matrix(timeCount) = ht_bnd_matrix(timeCount - 1) + delta_Ty;
78          ht_bnd_matrix(timeCount) = min(max(ht_bnd_matrix(timeCount), -ht_bnd_a), ht_bnd_a);
79
80          hr_bnd_matrix(timeCount) = hr_bnd_matrix(timeCount - 1) + delta_Ry;
81          hr_bnd_matrix(timeCount) = min(max(hr_bnd_matrix(timeCount), -hr_bnd_a), hr_bnd_a);
82
83          d_bnd_matrix(timeCount) = hr_bnd_matrix(timeCount - 1) + delta_Tx+delta_Rx;
84          d_bnd_matrix(timeCount) = min(max(d_bnd_matrix(timeCount), -d_bnd_a), d_bnd_a);
85          v_Tx_a(timeCount) =0.5;
86          v_Rx_a(timeCount) =0.5;
87          v_Ty_a(timeCount) =0;
88          v_Ry_a(timeCount) =0;
89        end
90
91        tempFolder=['tmp' profilSSP char(num2str(seastate_a(1)))];
92        % Erzeugen der Reflektionskoeffizienten f'fcr die CIR-Generation
93        cd(channeLocationRoot);
94        if exist(tempFolder,'dir')==0; mkdir (tempFolder); end
95        cd (tempFolder);
96        AUVModem.writeBCR(SSP, densitySurface, cBottom, densityBottom,title);
97
98        for bottomTypeCount=1:1%length(bottomType_a) % Schleife ueber alle Bodenprofile
99          % Schleife ueber alle Traegerfrequenzen definiert 'fcber die Klasse AUVModem
100         for freqCarrCount=length(Carrierfreq_a):-1:1
101           Carrierfreq=Carrierfreq_a(freqCarrCount);
102           % Laden des zu Traegerfrequenz zugeordneten Transducers
103           AUVParams.transducer=AUVModem.Transducer(freqCarrCount);
104           transducer=AUVParams.transducer;
105           AUVParams.TransducerTxSens=str2double(char(AUVModem.TransducerTxSens(freqCarrCount)));
106           %% Abstrahlchrakteristik des Trancducer generieren
107
108           % Daten aus der Datei lesen
109           filepath = char(strcat((AUVParams.transducer), '.sbp'));
110           %'M27-930-FFR-10kHz.sbp';
111           if exist(filepath, 'file')
112             delimiterIn = ' ';
113             headerlinesIn = 1;
114             data = importdata(filepath, delimiterIn, headerlinesIn);
115
116           % Winkel und D'e4mpfungswerte extrahieren
117             angles = data.data(:, 1);
118             loss = data.data(:, 2);
119             transmitAngleArray = -180:1:180;
120
121           % D'e4mpfung interpolieren (lineare Interpolation)
122             transmitAngleArrayLoss = interp1(angles, loss, transmitAngleArray, 'linear');
123           else
124             transmitAngleArrayLoss = 0;
125           end
126
127           cd(channeLocationRoot);
128           cd (tempFolder);
129           % Erstellen des Bodenreflektionskoeffizienten fuer BELLHOP in der Datei *.bcr
130           AUVModem.writeBCR(SSP, densitySurface, cBottom, densityBottom,title);
131           % Schleife 'fcbner alle Bandbreiten definiert 'fcber die Klasse AUVModem
132           for bandCount=1:length(BandWidth_a)
133             BandWidth=BandWidth_a(bandCount);
134             % Schleife 'fcber alle Seegangsstufen
135             for ssCount=1:length(seastate_a)
136               % Variation der Wasserhoehe basierend auf den Seegangsstufen
137               if matrix_max>1;h_bnd_matrix(2:end)= randi([-ceil(AUVModem.waveHeight(seastate_a(ssCount)+1)/2) ceil(
AUVModem.waveHeight(seastate_a(ssCount)+1)/2)],matrix_max-1,1);end
138               seastate=seastate_a(ssCount);
139               % Bestimmen der Windgeschwindigkeit basierend auf der Seegangsstufe
140               c_wind=AUVModem.beaufortTable(find(AUVModem.beaufortTable(:,1)==(seastate)),2);
141
142               cd(channeLocationRoot); cd (tempFolder);
143               % Erstellen des Oberflaechenreflektionskoeffizienten fuer BELLHOP in der Datei *.tcr
144               AUVModem.writeTCR(Carrierfreq,c_wind);
145               % Schleife 'fcber alle Kommunikationsdistanzen
146               for distanzCount=1:1:length(d0_a)
147                 cd(channeLocationRoot); cd (tempFolder);
148                 % Erstelle Bodenprofil
149                 AUVModem.writeBTY(d0_a(distanzCount),ceil(max(SSP_z)),bottomType_a(bottomTypeCount));
150                 % Erstelle Oberflaechenprofil
151                 AUVModem.writeATI(d0_a(distanzCount),0);
152                 cd(channeLocationRoot);
```

```
153                 % Schleife 'fcber alle Sendertiefen
154                 for transDeepCount=[find(ht0_a==150) find(ht0_a==300)];%1:1:length(ht0_a)
155                   % Schleife 'fcber alle Empfaengertiefen
156                   for recDeepCount=1:1:length(hr0_a)
157                     % Definiere restliche Geschwindigkeiten fest
158
159                     v_WF=[1 0 0]; % Stroemungsgeschwindigkeit in x y z
160                     v_W=[0 0 0.2]; % Wellengeschwindigkeit in x y z
161                     ht0=ceil(ht0_a(transDeepCount)); % Sendertiefe [m]
162                     hr0=hr0_a(recDeepCount); % Empfaengertiefe [m]
163                     d0=d0_a(distanzCount); % Kommunikationsdistanz [m]
164                     method_char='bel_dir';
165
166                     % Bestimme Kommunikationsdistanz-abhaengig die
167                     % Laenge der CIR + 1 Sekunde fuer Mehrwegeausbreitung
168                     time_axis = 0:1:ceil((sqrt(d0^2+h0^2)+2*h0)/min(SSP)+0.1)*Abtastrate;
169                     cd(channeLocationRoot);
170
171                     % Zwischenspeichern der Ausgangswerte
172                     % fuer die Postionen und Distanzen
173                     tmp_ht0=ht0; tmp_hr0=hr0; tmp_d0=d0;tmp_h0=h0;
174
175                     % Erstellen des spezifischen Dateinamens
176                     filename=strcat(profilSSP, '_surfh_', num2str(h0), '_Txh_', num2str(ht0), '_Rxh_', num2str(hr0),
 '_dist_', num2str(d0),'_seaststate_',num2str(seastate), '_', char(bottomType_a(bottomTypeCount)),'_',
method_char,'.mat');
177                     folder=[channelLocationOutput filesep 'FC_' num2str(Carrierfreq) '_BW_' num2str(BandWidth)
filesep 'Channels_Tx' num2str(ht0) '_SS' num2str(seastate) '_' char(bottomType_a(bottomTypeCount)) '_'
method_char];
178                     if exist(folder,'dir')==0; mkdir (folder); end
179                     cd (folder);
180
181                     % Erstelle die Dateien wenn diese nicht vorhanden sind
182                     if isfile(filename)==0&&~fast||rebuild
183
184                       % CIR-Generation mit SBP-Profil des Transducers
185                       CIRM_BELLHOP_SBP=zeros(matrix_max,length(time_axis));
186                       CIRM_BELLHOP_SBP_doppler=zeros(matrix_max,length(time_axis));
187                       % Erstellen einer CIRM (Matrix) mit BELLHOP
188                       % zur Abbildung des zeitvarianten
189                       % verhaltens unter dem Einfluss des SBP
190                       for i_matrix=1:1:matrix_max
191                         v_T=[v_Tx_a(i_matrix) v_Ty_a(i_matrix) 0];% Sendergeschwindigkeit in x y z
192                         v_R=[v_Rx_a(i_matrix) v_Ry_a(i_matrix) 0]; % Empfaengergeschwindigkeit in x y z
193                         % Zuweisung der zeitlich variierenden Werte der Postionen und Hoehe
194                         h0=tmp_h0;%+h_bnd_matrix(i_matrix);
195                         ht0=round(min(max(1,tmp_ht0+ht_bnd_matrix(i_matrix)),ceil(max(SSP_z))));
196                         hr0=round(min(max(1,tmp_hr0+hr_bnd_matrix(i_matrix)),ceil(max(SSP_z))));
197                         d0=round(tmp_d0+d_bnd_matrix(i_matrix));
198                         cd([channeLocationRoot filesep tempFolder]);
199
200                         % generieren der notwendigen CIRM Daten mit BELlHOP und SBP-Profil
201                         if i_matrix==1
202                           tStart = tic;
203                           [taumeanBell,thetaBell,nsBell,nbBell,hpBell]=AUVModem.runBellhop(h0,ht0,hr0,d0,Carrierfreq
,title,bottomType_a(bottomTypeCount),strcat(AUVParams.transducer, ".sbp"),SSP,SSP_z,AUVModem.d_z(AUVModem.d_z<
h0), time_axis,tempFolder);
204                           tend=toc(tStart);
205                           tau_0=max((d0/max(SSP))-0.100,0);
206                           matrixTimeBellSbp(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,
distanzCount) = full(tend);
207                         else
208                           [taumeanBell,thetaBell,nsBell,nbBell,hpBell]=AUVModem.runBellhop(h0,ht0,hr0,d0,Carrierfreq
,title,bottomType_a(bottomTypeCount),strcat(AUVParams.transducer, ".sbp"),SSP,SSP_z,AUVModem.d_z(AUVModem.d_z<
h0), time_axis,tempFolder);
209                         end
210                         taumeanBell=real(taumeanBell-tau_0);
211
212
213
214                         % Finde LOS
215                         ind_0=find(and(nsBell==0,nbBell==0));
216                         if length(ind_0)>1;[~, id]=max(hpBell(ind_0));ind_0=ind_0(id); end
217                         % Korrigiere Position damit LOS-Pfad an erster Stelle steht
218                         if length(ind_0)==1&&ind_0>1
219                           thetaBell=rad2deg(thetaBell);
220                           taumeanBell=[taumeanBell(ind_0) taumeanBell(1:ind_0-1) taumeanBell(ind_0+1:end)];
221                           thetaBell=[thetaBell(ind_0) thetaBell(1:ind_0-1) thetaBell(ind_0+1:end)];
222                           nsBell=[nsBell(ind_0) nsBell(1:ind_0-1) nsBell(ind_0+1:end)];
223                           nbBell=[nbBell(ind_0) nbBell(1:ind_0-1) nbBell(ind_0+1:end)];
224                           hpBell=[hpBell(ind_0) hpBell(1:ind_0-1) hpBell(ind_0+1:end)];
225                           ind_0=1;
226                         end
```

```
227                         if ~isempty(taumeanBell)
228                           thetaBell=thetaBell(taumeanBell<=length(time_axis)*1/Abtastrate );
229                           nsBell=nsBell(taumeanBell<=length(time_axis)*1/Abtastrate );
230                           nbBell=nbBell(taumeanBell<=length(time_axis)*1/Abtastrate );
231                           hpBell=hpBell(taumeanBell<=length(time_axis)*1/Abtastrate );
232                           taumeanBell=taumeanBell(taumeanBell<=length(time_axis)*1/Abtastrate );
233                         end
234
235
236                         if ~isempty(tau_0)
237                           tmp=AUVModem.calculateDoppler(ht0,hr0,thetaBell,taumeanBell+tau_0,hpBell,nsBell,nbBell,v_T
, v_R, v_WF, v_W,length(time_axis)+ceil(tau_0/Abtastrate),Abtastrate,SSP);
238                           CIRM_BELLHOP_SBP_doppler(i_matrix,:)=tmp(ceil(tau_0/Abtastrate)+1:end);
239                         else
240                           tmp=AUVModem.calculateDoppler(ht0,hr0,thetaBell,taumeanBell,hpBell,nsBell,nbBell,v_T, v_R,
 v_WF, v_W,length(time_axis),Abtastrate,SSP);
241                           CIRM_BELLHOP_SBP_doppler(i_matrix,:)=tmp(1:end);
242                         end
243
244
245                         for i_count=1:length(find(taumeanBell<ceil(2*h0/min(SSP)+0.1)))
246                           CIRM_BELLHOP_SBP(i_matrix,int32(fix((round(real(abs(taumeanBell(i_count))),6)*Abtastrate))
))= (CIRM_BELLHOP_SBP(i_matrix,int32(fix((round(real(abs(taumeanBell(i_count))),6)*Abtastrate))))+real(abs(
hpBell(i_count))));
247                         end
248
249                         if i_matrix==1
250                           % Speichere Daten f'fcr
251                           % analyse
252                           thetaBell_analyse=thetaBell;
253                           taumeanBell_analyse=taumeanBell;
254                           thetaBell_analyse=thetaBell;
255                           nsBell_analyse=nsBell;
256                           nbBell_analyse=nbBell;
257                           hpBell_analyse=hpBell;
258                           ind_0_analyse=ind_0;
259                         end
260                       end
261                       ht0=tmp_ht0;hr0=tmp_hr0; d0=tmp_d0;h0=tmp_h0;
262                       cd (folder);
263                       CIRM_BELLHOP_SBP_doppler = single(CIRM_BELLHOP_SBP_doppler);
264                       CIRM_BELLHOP_SBP = single(CIRM_BELLHOP_SBP);
265                       save(filename,'CIRM_BELLHOP_SBP_doppler','h0','tau_0', 'c_wind', 'h_bnd_a' ,'ht0','hr0','d0','
v_T', 'v_R', 'v_W', 'v_WF', 'ind_0_analyse', 'nsBell_analyse','nbBell_analyse','seastate','CIRM_BELLHOP_SBP','
hpBell_analyse','taumeanBell_analyse','thetaBell_analyse','transducer','-v7.3');
266                       if display;disp(filename);end
267                       % save([actpath filesep '..' filesep '..' filesep '..' filesep 'scripts' filesep '
Eva_Geo_Bell_' profilSSP '.mat'],'matrixCosine', 'matrixXCORR','matrixXcorrCor','matrixMSE','matrixTimeGeo','
matrixTimeBellSbp','matrixDSIR','-v7.3');
268                     else
269                       disp([filename ' already generated'])
270                       if calculateAddionalAnalsis&&~fast
271                         load(filename); %#ok<LOAD>
272                       end
273                     end
274
275                     method_char='fast';
276                     tStart = tic;
277                     %% genrate CIRM mat-file
278                     filename=strcat(profilSSP, '_surfh_', num2str(h0), '_Txh_', num2str(ht0), '_Rxh_', num2str(hr0),
 '_dist_', num2str(d0),'_seastate_',num2str(seastate),'_', char(bottomType_a(bottomTypeCount)),'_',method_char
,'.mat');
279                     folder=[channelLocationOutput filesep 'FC_' num2str(Carrierfreq) '_BW_' num2str(BandWidth)
filesep 'Channels_Tx' num2str(ht0) '_SS' num2str(seastate) '_' char(bottomType_a(bottomTypeCount)) '_'
method_char];
280                     if exist(folder,'dir')==0; mkdir (folder);end; cd (folder);
281                     if exist(filename,'file')==0||rebuild
282
283                       CIRM_GEO=zeros(matrix_max,length(time_axis));
284                       CIRM_GEO_doppler=zeros(matrix_max,length(time_axis));
285                       for i_matrix=1:1:1%matrix_max
286                         v_T=[v_Tx_a(i_matrix) v_Ty_a(i_matrix) 0];% Sendergeschwindigkeit in x y z
287                         v_R=[v_Rx_a(i_matrix) v_Ry_a(i_matrix) 0]; % Empfaengergeschwindigkeit in x y z
288                         h0=tmp_h0+h_bnd_matrix(i_matrix);
289                         ht0=max(1,tmp_ht0+ht_bnd_matrix(i_matrix));
290                         hr0=max(1,tmp_hr0+hr_bnd_matrix(i_matrix));
291                         d0=tmp_d0+d_bnd_matrix(i_matrix);
292
293                         %% genrate CIRM mat-file
294                         if i_matrix==1
295                           tStart = tic;
296                           [lmeanGeo,taumeanGeo,thetaGeo,nsGeo,nbGeo,hpGeo]=AUVModem.runGeoEstimation(h0,ht0,hr0,d0,
Carrierfreq,(SSP),c_wind,AUVModem.d_z(AUVModem.d_z<h0));
```

```
297                         tend=toc(tStart);
298                         tau_0_Geo=max(min(taumeanGeo)-0.100,0);
299                         matrixTimeGeo(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)
 = full(tend);
300                       else
301                         [lmeanGeo,taumeanGeo,thetaGeo,nsGeo,nbGeo,hpGeo]=AUVModem.runGeoEstimation(h0,ht0,hr0,d0,
Carrierfreq,(SSP),c_wind,AUVModem.d_z(AUVModem.d_z<h0));
302                       end
303                       taumeanGeo=taumeanGeo-tau_0_Geo;
304                       if ~isempty(taumeanGeo)
305                         lmeanGeo=lmeanGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
306                         thetaGeo=thetaGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
307                         nsGeo=nsGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
308                         nbGeo=nbGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
309                         hpGeo=hpGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
310                         taumeanGeo=taumeanGeo(taumeanGeo<=length(time_axis)*1/Abtastrate);
311                       end
312
313                       for i_count=1:length(find(taumeanGeo<ceil(2*h0/min(SSP)+0.1)))
314                         CIRM_GEO(i_matrix,int32(fix((round(real(abs(taumeanGeo(i_count))),6)*Abtastrate))))= abs(
CIRM_GEO(i_matrix,int32(fix((round(real((taumeanGeo(i_count))),6)*Abtastrate))))+real(abs(hpGeo(i_count))));
315                       end
316                       if i_matrix==1
317                         % Speichere Daten f'fcr
318                         % analyse
319                         thetaGeo_analyse=thetaGeo;
320                         taumeanGeo_analyse=taumeanGeo;
321                         nsGeo_analyse=nsGeo;
322                         nbGeo_analyse=nbGeo;
323                         hpGeo_analyse=hpGeo;
324                       end
325                     end
326                     ht0=tmp_ht0; hr0=tmp_hr0; d0=tmp_d0; h0=tmp_h0;
327                     cd (folder);
328                     CIRM_GEO_doppler = single(CIRM_GEO_doppler);
329                     CIRM_GEO = single(CIRM_GEO);
330                     save(filename,'h0','tau_0_Geo', 'c_wind', 'h_bnd_a', 'ht0','hr0','d0','v_T', 'v_R', 'v_W', '
v_WF','nsGeo_analyse','nbGeo_analyse', 'seastate','CIRM_GEO','CIRM_GEO_doppler','hpGeo_analyse','
taumeanGeo_analyse','thetaGeo_analyse','transducer','-v7.3');
331                     if display;disp(filename);end
332                   else
333
334                     if calculateAddionalAnalsis
335                       load(filename); %#ok<LOAD>
336                     end
337                   end
338
339                   if (calculateAddionalAnalsis)
340
341                     if ~exist('anaDataLoaded', 'var') || ~anaDataLoaded
342                       try
343                         % Pfad zur Datei erstellen
344                         dataFilePath = fullfile(actpath, '..', '..', '..', 'scripts', ['Eva_Geo_Bell_' profilSSP '
_' num2str(seastate+1) '.mat']);
345
346                         % 'dcberpr'fcfen, ob die Datei existiert
347                         if exist(dataFilePath, 'file')
348                           % Laden der Simulationsdaten
349                           load(dataFilePath);
350                           % Setzen der Variable, um anzuzeigen, dass die Daten geladen wurden
351
352                         else
353                           % Ausgabe einer Meldung, wenn die Datei nicht existiert
354                           % Initialisieren der Simulationsdaten
355                           matrixXCORR=single.empty();
356                           matrixXcorrCor=single.empty();
357                           matrixMSE=single.empty();
358                           matrixTimeGeo=single.empty();
359                           matrixTimeBellSbp=single.empty();
360                           matrixDSIR=single.empty();
361                           matrixCosine=single.empty();
362                           disp('Data file does not exist!!!');
363                         end
364                         anaDataLoaded = true;
365                       catch
366                         disp('Error loading data file!!!');
367                       end
368                       % Darstellung der Verteilung f'fcr die Korrelation und die DSIR
369                       if display
370
371                         figure;figureHandleDSIR=histogram(matrixDSIR,1000);
372                         figure;figureHandleXcorr=histogram(matrixXcorrCor,100);
373                       end
```

```
374                     else
375                       disp('Simulation data already loaded.');
376                     end
377
378                     DSIR=AUVModem.calculateDSIR(full(CIRM_BELLHOP_SBP(1,:)), full(CIRM_GEO(1,:)), int32(fix((round
(real(abs(taumeanBell_analyse)),6)*Abtastrate))), hpBell_analyse, int32(fix((round(real(abs(taumeanGeo_analyse
)),6)*Abtastrate))), hpGeo_analyse,ind_0_analyse);
379                     disp(['DSIR: ' num2str(DSIR) ' dB ']);
380
381
382                     matrixDSIR(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=DSIR;
383
384                     % Berechne den Kreuzkorrelationskoeffizienten
385                     val=xcorr(full(CIRM_BELLHOP_SBP(1,1:length(time_axis))), full(CIRM_GEO(1,1:length(time_axis)))
)./max(sum(CIRM_BELLHOP_SBP(1,1:length(time_axis)).^2),sum(CIRM_GEO(1,1:length(time_axis)).^2));
386                     idx_max=(find(abs(val)==max(abs(val))));
387
388                     [ matrixXCORR(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=
max(xcorr(full((CIRM_BELLHOP_SBP(1,1:length(time_axis)))), (full(CIRM_GEO(1,1:length(time_axis))))))./max(sum(
CIRM_BELLHOP_SBP(1,1:length(time_axis)).^2),sum(CIRM_GEO(1,1:length(time_axis)).^2)));
389
390                     % Display or XCORR as needed
391                     disp(['XCORR: ' num2str(matrixXCORR(seastate+1,freqCarrCount,bandCount,transDeepCount,
recDeepCount,distanzCount)*100) ' % ']);
392                     impulse1=full(CIRM_BELLHOP_SBP(1,1:length(time_axis)));
393                     impulse2=full(CIRM_GEO(1,1:length(time_axis)));
394                     tmp_size=max(find(abs(impulse2)~=0, 1, 'last')+1,find(abs(impulse1)~=0, 1, 'last')+1);
395
396                     impulse1=impulse1(1:tmp_size);
397                     impulse2=impulse2(1:tmp_size);
398
399                     old_shift=0;
400                     if matrixXCORR(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)
<0.99&&~isempty(tmp_size)&&matrixXCORR(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,
distanzCount)~=0
401                       old_corr=max(xcorr((impulse1), (impulse2))./max(sum(impulse1.^2),sum(impulse2.^2)));
402                       old_corr_tmp=old_corr;
403                       tmp_count=50;
404                       count_corr=-tmp_count;
405                       temp_corr=old_corr;
406
407                       while temp_corr<0.99 && count_corr<=tmp_count&&temp_corr~=0
408                         if temp_corr>old_corr;old_corr=temp_corr;old_shift=count_corr-1;end
409                         impulse2=zeros(1,length(time_axis));
410                         impulse2(int32(fix((round(real(taumeanGeo_analyse(2:end)),6)*Abtastrate)))+int32(
count_corr))= real(abs(hpGeo_analyse(2:end)));
411                         impulse2(int32(fix((round(real(taumeanGeo_analyse(1)),6)*Abtastrate))))= real(abs(
hpGeo_analyse(1)));
412                         impulse2=impulse2(1:length(impulse1));
413                         temp_corr=max(xcorr((impulse1), (impulse2(1:length(impulse1))))./max(sum(impulse1.^2),sum(
impulse2.^2)));
414                         count_corr=count_corr+1;
415                         if count_corr>tmp_count&&old_corr>old_corr_tmp
416                           impulse2=zeros(1,length(time_axis));
417                           impulse2(int32(fix((round(real(taumeanGeo_analyse(2:end)),6)*Abtastrate)))+int32(
old_shift))= real(abs(hpGeo_analyse(2:end)));
418                           impulse2(int32(fix((round(real(taumeanGeo_analyse(1)),6)*Abtastrate))))= real(abs(
hpGeo_analyse(1)));
419                           impulse2=impulse2(1:length(impulse1));
420                         elseif count_corr>tmp_count
421                           impulse2=full(CIRM_GEO(1,1:tmp_size));
422                         end
423                       end
424
425                       % Display or XCORR as needed
426                       disp(['XCORR corrected: ' num2str(max(xcorr((impulse1), (impulse2))./max(sum(impulse1.^2),
sum(impulse2.^2)))*100) ' % ']);
427                       matrixXcorrCor(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=
 max(xcorr((impulse1), (impulse2))./max(sum(impulse1.^2),sum(impulse2.^2)));
428
429                     else
430                       matrixXcorrCor(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=
matrixXCORR(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount);
431                     end
432
433                     if max(xcorr((impulse1), (impulse2))./max(sum(impulse1.^2),sum(impulse2.^2)))<0.5
434                       max(xcorr((impulse1), (impulse2))./max(sum(impulse1.^2),sum(impulse2.^2)))
435                     end
436
437                     [val idGeo]=sort(taumeanGeo_analyse);
438                     [val idBell]=sort(taumeanBell_analyse);
439                     sortGeo=zeros(1,max(length(idGeo),length(idBell)));
440                     sortBell=zeros(1,max(length(idGeo),length(idBell)));
```

```
441                         sortGeo(1:length(idGeo))=hpGeo_analyse(idGeo);
442                         sortBell(1:length(idBell))=hpBell_analyse(idBell);
443                         [valxcorr idxcorr]=max(xcorr((impulse1), (impulse2)));
444
445                         mse_tmp_percent=-1;
446                         if sum(impulse1.^2)~=0
447                           if idxcorr-length(impulse1)>0
448                             mse_tmp_percent=sum((abs(impulse1(1+idxcorr-length(impulse1):end))-abs(impulse2(1:end-(
      idxcorr-length(impulse1))))).^2)/sum(impulse1.^2)*100;
449                           else
450                             mse_tmp_percent=sum((abs(impulse1(1:end+(idxcorr-length(impulse1))))-abs(impulse2(1-(
      idxcorr-length(impulse1)):end))).^2)/sum(impulse1.^2)*100;
451                           end
452                         end
453
454                         matrixMSE(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=
      mse_tmp_percent;
455                         if isnan(matrixMSE(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount
      ))
456                           matrixMSE(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount)=100;
457                         end
458                         disp(['MSE: ' num2str(matrixMSE(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount
      ,distanzCount)) ' %' ])
459                         mse_tmp_percent_amplitude=-1;
460                         if sum(abs(sortBell))~=0
461                           mse_tmp_percent_amplitude=(sum(abs(sortBell))-sum(abs(sortGeo)))/sum(abs(sortBell))*100;
462                         end
463                         matrixMSEamplitude(seastate+1,freqCarrCount,bandCount,transDeepCount,recDeepCount,distanzCount
      )=mse_tmp_percent_amplitude;
464
465                         disp(['MSE-Amplitude: ' num2str(matrixMSEamplitude(seastate+1,freqCarrCount,bandCount,
      transDeepCount,recDeepCount,distanzCount)) ' %' ])
466
467                         % Save simulation results
468                         save([actpath filesep '..' filesep '..' filesep '..' filesep 'scripts' filesep 'Eva_Geo_Bell_'
       profilSSP '_' num2str(seastate+1) '.mat'],'matrixCosine', 'matrixXCORR','matrixXcorrCor','matrixMSE','
      matrixTimeGeo','matrixTimeBellSbp','matrixDSIR','matrixMSEamplitude','-v7.3');
469                         % Aktualisiere das Diagramm
470                         if display&&calculateAddionalAnalsis
471                           figureHandleDSIR.Data=matrixDSIR;
472                           figureHandleXcorr.Data=matrixXcorrCor;
473                         end
474                       end
475                     end
476                   end
477                 end
478               end
479             end
480           end
481         end
482       end}
```

Listing A.9: Generierung der relativen Geschwindigkeit

```
1  function relativeVelocity = calculateRelativeVelocity(v_T, v_R, v_W, v_WF, theta_k, NS_k)
2    % Author: Marcel Riess
3    % Description: Computes the relative velocity between two objects
4    % considering surface reflections with vertical movement.
5    %
6    % Input parameters:
7    %  v_T: Velocity of the transmitter (m/s)
8    %  v_R: Velocity of the receiver (m/s)
9    %  v_W: Velocity of the water surface (m/s)
10   %  v_WF: Flow velocity (m/s)
11   %  theta_k: Angle of the sound beam (radians)
12   %  NS_k: Number of surface reflections
13   %
14   % Output:
15   %  relative_velocity: Relative velocity change of the sound (m/s)
16
17   % Calculation of relative velocity
18   delta_v_x = v_T(1) - v_R(1) + v_W(1) + v_WF(1); % Relative velocity in x-direction
19   delta_v_z = v_T(3) - v_R(3) + double(NS_k) * v_W(3) + v_WF(3); % Relative velocity in z-direction considering
         surface reflections
20
21   % Compute relative velocity
22   relativeVelocity = sqrt((delta_v_x)^2 + (delta_v_z * tan(theta_k))^2);
23 end
```

Listing A.10: Berechnung des Difference in Signal to Interference Ratio (DSIR)

```
function DSIR = calculateDSIR(cirBell, cirGeo, tauBell, hpBell, tauGeo, hpGeo, ind_0)
  % Autor: Marcel Riess
  % Description: Calculates the Differential Signal-to-Interference Ratio (DSIR).
  %
  % Assumptions: tauBell(1) and tauGeo(1) represent time information
  % and hpBell(1) and hpGeo(1) represent amplitude information of the LOS path.
  %
  % Input Parameters:
  %  - cirBell: Channel impulse response generated with Bellhop
  %  - cirGeo: Channel impulse response generated with geometric model
  %  - tauBell: Delay vector for the channel impulse response (Bellhop)
  %  - hpBell: Channel amplitude vector for the channel impulse response (Bellhop)
  %  - tauGeo: Delay vector for the channel impulse response (geometric model)
  %  - hpGeo: Channel amplitude vector for the channel impulse response (geometric model)
  %  - ind_0: Index for LOS path (-1 if no LOS path exists)
  %
  % Output Parameter:
  %  - DSIR: Differential Signal-to-Interference Ratio

  % Initialize the result variable
  DSIR = [];

  % Extract interference in the impulse responses
  % if there is a LOS path
  if ind_0 ~= -1
    % Generate the signal and compute the signal-dependent difference
    % in SIR between the two CIRs
    minTempSize = min(size(cirBell(tauBell(ind_0) + 1:end), 2), size(cirGeo(tauGeo(1) + 1:end), 2));
    signal = ones(1, minTempSize);
    cirBell_tmp = cirBell;
    cirBell_tmp(tauBell(ind_0)) = 0;
    cirGeo_tmp = cirGeo;
    cirGeo_tmp(tauGeo(1)) = 0;

    % Convolve with the signal to get interference
    rx_bell_interference = conv(cirBell_tmp, signal);
    rx_geo_interference = conv(cirGeo_tmp, signal);

    % Compute DSIR
    DSIR = [DSIR (snr(signal * abs(hpGeo(1)), rx_geo_interference(1:minTempSize))) - (snr(signal * abs(hpBell(1)),
      rx_bell_interference(1:minTempSize)))];
  else
    try
      % No direct line of sight, therefore set DSIR to 40 dB
      minTempSize = min(size(cirBell(tauBell(1) + 1:end), 2), size(cirGeo(tauGeo(1) + 1:end), 2));
      signal = ones(1, minTempSize);
      cirBell_tmp = cirBell;
      cirBell_tmp(tauBell(1)) = 0;
      cirGeo_tmp = cirGeo;
      cirGeo_tmp(tauGeo(1)) = 0;

      % Convolve with the signal to get interference
      rx_bell_interference = conv(cirBell_tmp, signal);
      rx_geo_interference = conv(cirGeo_tmp, signal);

      % Compute DSIR
      DSIR = [DSIR (snr(signal * abs(hpGeo(1)), rx_geo_interference(1:minTempSize))) - (snr(signal * abs(hpBell(1)),
       rx_bell_interference(1:minTempSize)))];
    catch
      % If an error occurs, set DSIR to 40 dB
      DSIR = 40;
    end

    disp('DSIR: No direct line of sight');
  end
end
```

Listing A.11: Bestimmung der CIR über BELLHOP

```
function [taumean,theta,ns,nb,hp]=runBellhop(zmax, TX_depth, RX_depth, RX_range, freq,title,bottomType,SBP,SSP,SSP_z
    ,~,time_axis,tempFolder)
% runBellhop executes the BELLHOP program and analyzes the results.
%
% Author: Marcel Riess
%
% Input Parameters:
%  zmax: Maximum depth (m)
%  TX_depth: Transmission depths (m)
%  RX_depth: Receiver depths (m)
%  RX_range: Receiver range (m)
%  freq: Carrier frequency (Hz)
%  title: File name
%  bottomType: Bottom type ('flat' for flat, other values for non-flat)
%  SBP: Source beam pattern file (empty if not present)
%  SSP: Sound speed profile
%  SSP_z: Depth vector for SSP
%  z: Depth
%  time_axis: Time axis for analysis
%  tempFolder: Folder for generated Bellhop files
%
% Output Parameters:
%  taumean: Delay
%  theta: Launch angle at transmitter
%  ns: Number of surface reflections
%  nb: Number of bottom reflections
%  hp: Amplitude
%

% Global variables for sound speed and bottom type
global cs_surface alpha_surface cs_bottom alpha_bottom Abtastrate alpha_bottom densitySurface densityBottom
    cBottom

% BELLHOP file names
% title

% Copying SBP file to temporary directory
if ~isempty(SBP)
  cd ..
  copyfile(fullfile('Transducer', 'SBP', SBP), fullfile(tempFolder, [title '.sbp']));
  cd(tempFolder);
end

% OPTIONS for .env file
options1 = 'SFMT'; % OPTIONS1 according to BELLHOP documentation
switch(bottomType)
  case 'flat'
    options2 = 'F'; % Flat bottom surface
  otherwise
    options2 = 'F'; % Non-flat bottom surface
end
if isempty(SBP)
  options3 = 'A'; % Writes amplitudes and travel times
else
  options3 = 'A *'; % Writes amplitudes and travel times, requires *.sbp file for source beam pattern
end

% Number of transmission and reception depths and ranges
nTX_depth = length(TX_depth);
nRX_depth = length(RX_depth);
nRX_range = length(RX_range);

% Write parameters for the .env file
range_box = max(RX_range) * 1.1;
depth_box = ceil(zmax);
nbeams = 5000; % Number of rays
launching_angles = [-90 90]; % Range of launch angles

% Writing the .env file
f = fopen([title '.env'], 'w');
fprintf(f, ''%s''\n', title); % Title
fprintf(f, '%f ! Frequency (Hz)\n', freq); % Frequency
fprintf(f, '1 ! Number of media\n'); % Number of media
fprintf(f, ''%s'' ! OPTIONS1\n', options1); % OPTIONS1
if options1(2)=='A'
  fprintf(f, '%f %f %f %f %f %f ! Z(m) CP CS (m/s) RHO (g/cm3) AP AS\n', 0.0, SSP(1), cs_surface, densitySurface,
    alpha_surface(1), alpha_surface(2)); % Z(m) CP CS (m/s) RHO (g/cm3) AP AS
end
fprintf(f, '%d 0.0 %f \n', 51, ceil(zmax));
fprintf(f, '0 %f / \n', SSP(1));
d_z_temp = SSP_z(SSP_z<=ceil(max(SSP_z)));
```

```
79    for i = 2:10:length(SSP_z(1:end-1))
80      fprintf(f, '%f %f / \n', (SSP_z(i)), SSP(i));
81    end
82    fprintf(f, '%f %f / \n', ceil(zmax), SSP(d_z_temp(end)));
83    fprintf(f, ''%s'' 0 ! OPTIONS2\n', options2);
84    if options2(1)=='A'
85      fprintf(f, '%f %f %f %f %f %f /\n', ceil(zmax), cBottom, cs_bottom, densityBottom, alpha_bottom(1), alpha_bottom
        (2));
86    end
87    fprintf(f, '%d ! Number of sources\n', nTX_depth);
88    if(length(TX_depth) == 2)
89      fprintf(f, '%f %f / ! Transmitter depth\n', TX_depth(1), TX_depth(2));
90    else
91      fprintf(f, '%f / ! Transmitter depth\n', TX_depth);
92    end
93    fprintf(f, '%d ! Number of receivers in depth\n', nRX_depth);
94    if(length(RX_depth) == 2)
95      fprintf(f, '%f %f / ! Receivers depth\n', RX_depth(1), RX_depth(2));
96    else
97      fprintf(f, '%f / ! Receivers depth\n', RX_depth);
98    end
99    fprintf(f, '%d ! Number of receivers in range\n', nRX_range);
100   if(length(RX_range) == 2)
101     fprintf(f, '%f %f / ! Receivers range\n', RX_range(1)/1000, RX_range(2)/1000);
102   else
103     fprintf(f, '%f / ! Receivers range\n', RX_range/1000);
104   end
105   fprintf(f, ''%s'' ! OPTIONS3\n', options3);
106   fprintf(f, '%d ! nbeams (number of launching angles)\n', nbeams);
107   fprintf(f, '%f %f / ! Angles [min max] degrees \n', launching_angles);
108   fprintf(f, '0.0 %f %f ! STEP (m), ZBOX (m), RBOX (km)\n', depth_box+50, range_box/1000);
109   fclose(f);
110
111   % Running BELLHOP
112   bellhop(title);
113
114   % Reading the result file and analysis
115
116   arr_info = read_arrivals_asc([title, '.arr']);
117   delete([title '.env']);
118   delete([title '.arr']);
119   P = arr_info.Narr;
120   ns = arr_info.NumTopBnc(1:P);
121   nb = arr_info.NumBotBnc(1:P);
122   tbell = arr_info.delay(1:P);
123   taumean = tbell;
124   hp = arr_info.A(1:P);
125   hp = abs(hp) .* sign(real(hp));
126   theta = -arr_info.SrcDeclAngle(1:P);
127
128   % Sorting the data
129   [taumean, id] = sort(taumean);
130   theta = theta(id);
131   ns = ns(id);
132   nb = nb(id);
133   hp = hp(id);
134 end
```

Listing A.12: Berechnung der CIR mit Dopplereffekten

```
function CIR_doppler = calculateDoppler(ht0, hr0, theta, taumean, hp, nsBell, nbBell, v_T, v_R, v_WF, v_W,
    length_matrix, sample_rate, SSP)
  % Author: Marcel Riess
  % Description: Computes the Doppler effect on the channel impulse response.
  %
  % Input parameters:
  %  ht0: Transmitter depth (m)
  %  hr0: Receiver depth (m)
  %  theta: Transmitter emission angle (radians)
  %  taumean: Mean delay (s)
  %  hp: Amplitude array of all paths (1xN array)
  %  nsBell: Array of number of surface reflections per path
  %  nbBell: Array of number of bottom reflections per path
  %  v_T: Transmitter velocity (m/s)
  %  v_R: Receiver velocity (m/s)
  %  v_WF: Flow velocity (m/s)
  %  v_W: Water surface velocity (m/s)
  %  length_matrix: Length of the channel impulse response matrix
  %  sample_rate: Sampling rate (Hz)
  %  SSP: Sound speed profile
  %
  % Output:
  %  CIR_doppler: Channel impulse response with Doppler effects
  %

  % Initialize variables
  length_matrix = ceil(real(length_matrix));
  CIR_doppler = zeros(1, length_matrix);

  try
    % Limit frequencies of the channel within the specified range limited by the sampling
    % rate according to Nyquist

    frequencies = linspace(-Abtastrate / 2 + 1, Abtastrate / 2, length_matrix);
    [~, id] = find(taumean <= ((length_matrix - 1) / Abtastrate));

    % Iterate over all signal paths
    for i = 1:length(id)
      i_count = id(i);

      % Extract signal path
      CIR = zeros(1, length_matrix);
      CIR(int32(fix(round(real(abs(taumean(i_count))), 6) * Abtastrate))) = real(abs(hp(i_count)));

      % Calculate relative velocity in m/s
      rel_velocity = calculateRelativeVelocity(v_T, v_R, v_W, v_WF, deg2rad(theta(i_count)), nsBell(i_count));

      % Calculate mean acoustic speed in water in m/s
      if (nsBell(i_count) == 0 && nbBell(i_count) == 0)
        c_water = (SSP(ht0) + SSP(hr0)) / 2;
      elseif (xor(nbBell(i_count), nsBell(i_count)) == 1 && nsBell(i_count))
        c_water = (sum(SSP(1:ht0)) + sum(SSP(1:hr0))) / (length(SSP(1:ht0)) + length(SSP(1:hr0)));
      elseif ((xor(nbBell(i_count), nsBell(i_count)) == 1 && nbBell(i_count)))
        c_water = (sum(SSP(ht0:end)) + sum(SSP(hr0:end))) / (length(SSP(ht0:end)) + length(SSP(hr0:end)));
      else
        c_water = (sum(SSP(1:ht0) + sum(SSP(1:hr0))) / (length(SSP(1:ht0)) + length(SSP(1:hr0))));
      end

      % Calculate frequency shift
      frequency_shift = rel_velocity / c_water;

      % Calculate Doppler shift
      doppler_factor = exp(-1i * 2 * pi .* frequency_shift .* frequencies);

      % Step 1: Impulse response to channel transfer function
      channel_transfer_function = fft(CIR);

      % Step 2: Adding motion-based Doppler effect
      modified_transfer_function = channel_transfer_function .* doppler_factor;

      % Step 3: Channel transfer function back to impulse response
      CIR_doppler = abs(real(ifft(modified_transfer_function))) + CIR_doppler;
    end
  catch
    disp('Problem with doppler')
    disp(length_matrix)
  end
end
```

Listing A.13: Bestimmung der CIR über BELLHOP

```
1
2  function [lmean,taumean,theta,ns,nb,hp]=runGeoEstimation(zmax,TX_depth,RX_depth,RX_range,f_c,SSP,c_wind,z)
3  % runGeoEstimation fuehrt die geometrische Schaetzung durch und berechnet die Ergebnisse.
4  %
5  % Autor: Marcel Riess
6  %
7  % Eingabeparameter:
8  %  zmax: Maximale Tiefe
9  %  TX_depth: uebertragungstiefe
10 %  RX_depth: Empfangstiefe
11 %  RX_range: Empfangsreichweite
12 %  f_c: Traegerfrequenz
13 %  SSP: Schallgeschwindigkeitsprofil
14 %  c_wind: Windgeschwindigkeit
15 %  z: Tiefe
16 %
17 % Ausgabeparameter:
18 % Arrays nach Signalpfaden sortiert
19 %  lmean: Strahlenlaenge
20 %  taumean: Mittlere Ankunftszeit
21 %  theta: Abstrahl-Winkel
22 %  ns: Anzahl der oberen Reflexionen
23 %  nb: Anzahl der unteren Reflexionen
24 %  hp: Amplitude
25 %
26 % Anmerkung:
27 %  Der Streufaktor k wird mit Quadratischer Entfernung bestimmt,
28 %  wobei jede Reflektion als neue Signalquelle interpretiert wird
29 %  der Faktor k_factor wurde empirisch bestimmt.
30
31 global transmitAngleArray transmitAngleArrayLoss BLRC SLRC alphaThorp snrContrain densitySurface densityBottom
       cBottom %#ok<GVMIS>
32
33 % Berechne LOS-Signal-Laenge (m)
34 lmean(1)=sqrt((TX_depth-RX_depth)^2+RX_range^2);
35 theta(1)=(atan((RX_depth-TX_depth)/RX_range)*180/pi);
36
37 % Spreadingfaktorkompensionfaktor
38 k_factor=0.000000026;
39
40 % Berechne den Verlust durch das Abstrahlverhalten des Transducers
41 tranducerLoss=(db2mag(transmitAngleArrayLoss(transmitAngleArray==(round((theta(1)))))));%mag2db(db2mag(
       transmitAngleArrayLoss(find(transmitAngleArray==ceil(theta))))^2);
42 k=2-((k_factor)*(lmean(1))^2);
43 hp(1)=((tranducerLoss)*db2mag(-k*10*log10(lmean(1))-alphaThorp(f_c/1000)*lmean(1)/1000));
44
45 % Berechne die Differenzen zu definierten Tiefenpunkten
46 differenzen = abs(z - [TX_depth RX_depth]');
47
48 % Finde den Index des Elements mit der kleinsten Differenz
49 [~, index] = min(differenzen,[],2);
50
51 % Bestimme Verzoegerung in Abhaengigkeit des SSP
52 if RX_depth<TX_depth
53   taumean(1)=round(lmean(1),1)/(sum(SSP(z(index(2):index(1))))/length((z(index(2):index(1)))));
54 elseif TX_depth<RX_depth
55   taumean(1)=round(lmean(1),1)/(sum(SSP(z(index(1):index(2))))/length((z(index(1):index(2)))));
56 else
57   taumean(1)=round(lmean(1),1)/(SSP(z(max(index(1),index(1)))));
58 end
59
60 % Definiert LOS-Signal an erster Stelle
61 ns(1)=0;
62 nb(1)=0;
63
64 % Bestimme alle Refektionen beginnend ueber die Oberflaeche
65 firstReflection=true;
66 % Anzahl der Reflektionen
67 nr=0;
68
69 % Schleife ueber alle Reflektionen ueber die Oberflaeche bis die Abbruchbedingung erfuellt wird
70 while firstReflection||(mag2db(hp(1))-mag2db(max(hp(end))))<=snrContrain
71   firstReflection=false;
72   nr=nr+1;
73
74  % Bestimme Anzahl der Reflektionen an Oberflaeche und Boden
75   ns(nr+1)=mod(nr,2)+floor(nr/2);
76   nb(nr+1)=floor(nr/2);
77
78  % Bestimme theoretische Hoehe
```

```
79    heff=TX_depth + (nb(nr+1)+ns(nr+1)-1)*zmax + (1-(ns(nr+1)-nb(nr+1)))*(zmax-RX_depth)+ (ns(nr+1)-nb(nr+1))*(
          RX_depth);
80
81    % Bestimme die zurueckgelegte Distanz des Strahls
82    lmean(nr+1)=sqrt(double(heff^2+RX_range^2));
83
84    % Bestimme die benoetigte Zeit in Abhaengigkeit des SSP
85    taumean(nr+1)=round(lmean(nr+1),1)/round((sum(SSP(z(1:index(1))))+ sum(SSP(z(1:index(2))))*(ns(nr+1)-nb(nr+1))+
          sum(SSP(z(index(2):end)))*(1-(ns(nr+1)-nb(nr+1)))+sum(SSP(z))*(ns(nr+1)-1+nb(nr+1)))/(length((z(1:index(1))))+
           length((z(1:index(2))))*(ns(nr+1)-nb(nr+1))+ length((z(index(2):end)))*(1-(ns(nr+1)-nb(nr+1)))+length(z)*(ns(
          nr+1)-1+nb(nr+1))),1); %#ok<*AGROW>
86
87    % Berechnen des Abstrahlwinkel
88    theta(nr+1)=(atan(heff/RX_range)*180/pi);
89
90    % Bestimme den Verlust ueber das SBP
91    tranducerLoss=(db2mag(transmitAngleArrayLoss(transmitAngleArray==(round((theta(nr+1)))))));
92
93    % Bestimme Streufaktor k in abhaengigkeit der Entfernung
94    k_temp=0;
95    if ns(end)+nb(end)==1
96      l_tmp=TX_depth/cos(deg2rad(abs(90-theta(nr+1))));
97      l_tmp=sqrt(l_tmp*conj(l_tmp));
98      k_temp=+k_temp+k_factor*(l_tmp)^2;
99      k_temp=+k_temp+k_factor*(lmean(nr+1)-l_tmp)^2;
100   elseif ns(end)+nb(end)==2
101     l_tmp=TX_depth/cos(deg2rad(abs(90-theta(nr+1))));
102     l_tmp=sqrt(l_tmp*conj(l_tmp));
103     k_temp=+k_temp+((k_factor)*(l_tmp)^2);
104     l_tmp2=zmax*asin(theta(nr+1));
105     l_tmp2=sqrt(l_tmp2*conj(l_tmp2));
106     k_temp=+k_temp+((k_factor)*(l_tmp2)^2);
107     l_tmp3=lmean(nr+1)-l_tmp-l_tmp2;
108     k_temp=+k_temp+((k_factor)*(l_tmp3)^2);
109   else
110     l_tmp=TX_depth/cos(deg2rad(abs(90-theta(nr+1))));
111     l_tmp=sqrt(l_tmp*conj(l_tmp));
112     k_temp=+k_temp+((k_factor)*(l_tmp)^2);
113     l_tmp2=zmax/cos(deg2rad(abs(90-theta(nr+1))));
114     l_tmp2=sqrt(l_tmp2*conj(l_tmp2));
115     k_temp=+k_temp+((k_factor)*(l_tmp2)^2)*(ns(end)+nb(end)-1);
116     l_tmp3=lmean(nr+1)-l_tmp2*(ns(end)+nb(end)-1)-l_tmp;
117     k_temp=+k_temp+((k_factor)*(l_tmp3)^2);
118   end
119   k=min(2,2-k_temp);
120   % Berechne die Amplitude des Pfades mit allen Verlusten
121   hp(nr+1)=tranducerLoss*((BLRC(deg2rad(theta(nr+1)),SSP(end),cBottom,densitySurface,densityBottom)))^nb(nr+1)*(SLRC
          (f_c/1000,theta(nr+1),c_wind))^ns(nr+1)*(tranducerLoss)*db2mag(-(k)*10*log10(lmean(nr+1)))*db2mag(-alphaThorp(
          f_c/1000)*lmean(nr+1)/1000);
122 end
123
124 % Entferne letzten Pfad der die Bedingung nicht erfuellt
125 p_start=nr-1;
126
127 % Bestimme alle Refektionen beginnend ueber die Oberflaeche
128 firstReflection=true;
129 nr=0;
130
131 % Schleife ueber alle Reflektionen ueber die Oberflaeche bis die Abbruchbedingung erfuellt wird
132 while firstReflection||(mag2db(hp(1))-mag2db(max(hp(end))))<=max(3,snrContrain)%G(1)/cut;
133   firstReflection=false;
134   nr=nr+1;
135
136   % Bestimme Anzahl der Reflektionen an Oberflaeche und Boden
137   nb(p_start+nr+1)=mod(nr,2)+floor(nr/2);
138   ns(p_start+nr+1)=floor(nr/2);
139
140   % Bestimme theoretische Hoehe
141   heff=(zmax-TX_depth) + (ns(p_start+nr+1)+nb(p_start+nr+1)-1)*zmax + ((nb(p_start+nr+1)-ns(p_start+nr+1)))*(zmax-
          RX_depth)+(1- (nb(p_start+nr+1)-ns(p_start+nr+1)))*(RX_depth);
142
143   % Bestimme die zurueckgelegte Distanz des Strahls
144   lmean(p_start+nr+1)=sqrt(heff^2+RX_range^2);
145
146   % Bestimme die benoetigte Zeit in Abhaengigkeit des SSP
147   taumean(p_start+nr+1)=lmean(p_start+nr+1)/((sum(SSP(TX_depth+10:10:end))+ sum(SSP(RX_depth+10:10:end)))*(nb(p_start
          +nr+1)-ns(p_start+nr+1))+ sum(SSP(10:10:RX_depth))*(1-(nb(p_start+nr+1)-ns(p_start+nr+1)))+sum(SSP(10:10:end))
          *(nb(p_start+nr+1)-1+ns(p_start+nr+1)))/(length(SSP(TX_depth+10:10:end))+ length(SSP(RX_depth+10:10:end))*(nb(
          p_start+nr+1)-ns(p_start+nr+1))+ length(SSP(10:10:RX_depth))*(1-(nb(p_start+nr+1)-ns(p_start+nr+1)))+length(
          SSP(10:10:end))*(nb(p_start+nr+1)-1+ns(p_start+nr+1))));
148
149   % Berechnen des Abstrahlwinkels
150   theta(p_start+nr+1)=(atan(heff/RX_range)*180/pi);
```

```
151
152   % Bestimmung des Verlust ueber den SBP
153   tranducerLoss=(db2mag(transmitAngleArrayLoss(transmitAngleArray==(round(-theta(p_start+nr+1))))));
154
155   % Bestimme Streufaktor k in abhaengigkeit der Entfernung
156   k_temp=0;
157   if ns(end)+nb(end)==1
158     l_tmp=(zmax-TX_depth)/sin(deg2rad(abs(theta(p_start+nr+1))));
159     l_tmp=sqrt(l_tmp*conj(l_tmp));
160     k_temp=+k_temp+((k_factor)*(l_tmp)^2);
161     k_temp=+k_temp+((k_factor)*(lmean(p_start+nr+1)-l_tmp)^2);
162   elseif ns(end)+nb(end)==2
163     l_tmp=(zmax-TX_depth)/sin(deg2rad(abs(theta(p_start+nr+1))));
164     l_tmp=sqrt(l_tmp*conj(l_tmp));
165     k_temp=+k_temp+((k_factor)*(l_tmp)^2);
166     l_tmp2=zmax/sin(deg2rad(abs(theta(p_start+nr+1))));
167     l_tmp2=sqrt(l_tmp2*conj(l_tmp2));
168     k_temp=+k_temp+((k_factor)*(l_tmp2)^2);
169     l_tmp3=lmean(p_start+nr+1)-l_tmp-l_tmp2;
170     k_temp=+k_temp+((k_factor)*(l_tmp3)^2);
171   else
172     l_tmp=(zmax-TX_depth)/sin(deg2rad(abs(theta(p_start+nr+1))));
173     l_tmp=sqrt(l_tmp*conj(l_tmp));
174     k_temp=+k_temp+((k_factor)*(l_tmp)^2);
175     l_tmp2=zmax/sin(deg2rad(abs(theta(p_start+nr+1))));
176     l_tmp2=sqrt(l_tmp2*conj(l_tmp2));
177     k_temp=+k_temp+((k_factor)*(l_tmp2)^2)*(ns(end)+nb(end)-1);
178     l_tmp3=lmean(p_start+nr+1)-l_tmp2*(ns(end)+nb(end)-1)-l_tmp;
179     k_temp=+k_temp+((k_factor)*(l_tmp3)^2);
180   end
181   k=min(2,2-k_temp);
182   % Berechne die Amplitude des Pfades mit allen Verlusten
183   hp(p_start+nr+1)=tranducerLoss*((BLRC(deg2rad(round(theta(p_start+nr+1))),SSP(end),cBottom,densitySurface,
          densityBottom)))^nb(p_start+nr+1)*(SLRC(f_c/1000,(theta(p_start+nr+1)),c_wind))^ns(p_start+nr+1)*(
          tranducerLoss)*db2mag(-(k)*10*log10(lmean(p_start+nr+1)))*db2mag(-alphaThorp(f_c/1000)*lmean(p_start+nr+1)
          /1000);
184 end
185
186 % Entferne letzten Pfad der die Bedingung nicht erfuellt
187 hp=hp(1:end-1);
188 taumean=taumean(1:end-1);
189 lmean=lmean(1:end-1);
190 theta=theta(1:end-1);
191
192 end
```

Listing A.14: Bestimmung des Reflexionskoeffizienten der Wasseroberfläche

```
function writeTCR(freq, cw)
  % Function: writeTCR
  % This function writes TCR data to a file.
  % The file is named based on the global variable 'title'.
  % The function computes SLRC (Surface Loss Reflection Coefficient) over
  % a range of angles (theta) and writes the results to the file.

  global title % Use the global variable 'title' for the file name

  % Define the SLRC function as an anonymous function
  % f: frequency
  % theta: angle in degrees
  % cw: correction weight
  SLRC = @(f, theta, cw) (db2mag(10*log10((1 + (f / (sqrt(10) * 378 * (cw / 0.514) .^ -2)) .^ 2) / (1 + (f / (378 *
      (cw / 0.514) .^ -2)) .^ 2)) - (1 + (90 - (cw / 0.514)) / (60)) * (theta / 30) .^ 2));

  % Define the range of theta from 0 to 90 degrees with a step of 0.1
  theta_range = 0:0.1:90;

  % Open a file with the name based on the global variable 'title'
  % with the extension '.trc' in write mode
  f = fopen([title '.trc'], 'w');

  % Write the number of theta values to the file
  fprintf(f, '%d \n', length(theta_range));

  % Loop over each value in the theta range
  for theta = theta_range
    % Write the theta value, SLRC, and negative theta value to the file
    fprintf(f, '%f %f %f\n', theta, (SLRC(freq / 1000, theta, cw)), -theta);
  end

  % Close the file
  fclose(f);
end
```

Listing A.15: Bestimmung des Reflexionskoeffizienten des Bodens

```
function writeBCR(SSP, density_surface, c_bottom, density_bottom, title)
  % Function: writeBCR
  % This function calculates and writes the Bottom Reflection Coefficients (BCR)
  % to a file. The file name is based on the input parameter 'title'.
  % The function computes the BCR for a range of incidence angles (theta1).

  % Define the range of incidence angles (theta1) from 0 to 90 degrees with a step of 0.1
  theta1_range = 0:0.1:90;

  % Open a file with the name based on the input parameter 'title'
  % with the extension '.brc' in write mode
  f = fopen([title '.brc'], 'w');

  % Write the number of theta1 values to the file
  fprintf(f, '%d \n', length(theta1_range));

  % Loop over each value in the theta1 range
  for i = 1:length(theta1_range)
    % Convert theta1 from degrees to radians
    theta1 = deg2rad(theta1_range(i));

    % Calculate theta2 using Snell's Law
    theta2 = asin((SSP(end) / c_bottom) * sin(theta1));

    % Calculate the acoustic impedances for the surface and bottom
    Z1 = density_surface * SSP(end);
    Z2 = density_bottom * c_bottom;

    % Calculate the reflection coefficient and write it to the file
    reflection_coefficient = ((Z1 * cos(theta1) - Z2 * cos(theta2)) / (Z1 * cos(theta1) + Z2 * cos(theta2)))^2;
    fprintf(f, '%f %f %f\n', theta1_range(i), reflection_coefficient, -theta1_range(i));
  end

  % Close the file
  fclose(f);
end
```

A.7 Reduktionsverfahren

Listing A.16: Bestimmung der geometrischen Reduktion

```
function [AUVParams, UWC] = geoReduction(AUVParams, UWC, fileNameCSV, SSP, SNR_BER, write, ~, kasSeqID, kasSeqDT)
% geoReduction - Geometrische Reduktion der Unterwasser-Kommunikationskanaele (UWC) durchfuehren.
% [AUVParams, UWC] = geoReduction(AUVParams, UWC, fileNameCSV, SSP, SNR_BER, write, ~, kasSeqID, kasSeqDT)
% fuehrt eine geometrische Reduktion basierend auf den gegebenen AUV-Parametern, UWC-Kanaelen und
% weiteren Eingaben durch. Die Funktion berechnet das Signal-Rausch-Verhaeltnis (SINR) fuer jeden Kanal
% und filtert Kanaele heraus, die die spezifizierten Kriterien nicht erfuellen.
%
% Eingaben:
%  AUVParams - Struktur mit den Parametern des autonomen Unterwasserfahrzeugs (AUV).
%  UWC - Zell-Array mit UWC-Kanal-Strings.
%  fileNameCSV - Name der CSV-Datei zum Speichern der Ergebnisse.
%  SSP - Schallgeschwindigkeitsprofil.
%  SNR_BER - Signal-Rausch-Verhaeltnis fuer die Bitfehlerquote.
%  write - (Optional) Flag, um anzuzeigen, ob Ergebnisse in CSV geschrieben werden sollen (Standard: true).
%  kasSeqID - Kasami-Sequenz-ID.
%  kasSeqDT - Kasami-Sequenz-Daten.
%
% Ausgaben:
%  AUVParams - Aktualisierte Struktur mit geometrischen Reduktionsparametern.
%  UWC - Gefiltertes Zell-Array der gueltigen UWC-Kanaele.

% Standardwert fuer 'write' setzen, falls nicht bereitgestellt
if nargin < 6
  write = true;
end

% Zaehler initialisieren und Startnachricht anzeigen
geoc = 0;
georestc = 0;
disp('##### Geometrische Reduktion!')
clear UWCgeo

% Rauschmodelle definieren
turbo_noise = @(x) (17 - 30 * log10(x / 1000)); % Turbulenzrauschen
ship_noise = @(x, ss) (40 + 20 * (ss - 0.5) + 26 * log10(x / 1000) - 60 * log10(x / 1000 + 0.03)); % Schiffsrauschen
surf_noise = @(x, sw) (50 + 7.5 * sqrt(sw) + 20 * log10(x / 1000) - 40 * log10(x / 1000 + 0.44)); % Oberflaechenrauschen
therm_noise = @(x) (-15 + 20 * log10(x / 1000)); % Thermisches Rauschen

% Zusaetzlicher Kompensationswert
add_comp = 3;

% Rahmenlaenge berechnen
framelength = (11 + AUVParams.d + AUVParams.k) / AUVParams.r * 8 / log2(AUVParams.M);
if AUVParams.ModulationActivEnum == CtrModulation.QPSKModulation || AUVParams.ModulationActivEnum == CtrModulation.BPSKModulation
  fs = AUVParams.BandWidth;
else
  fs = AUVParams.bits_per_second(AUVParams.num_channels, log2(AUVParams.M), AUVParams.BandWidth == 1000);
end

% Jeden UWC-Kanal verarbeiten
for iUWC = 1:length(UWC)
  valid = false;
  temp_str = split(UWC{iUWC}, "_");
  h = str2double(temp_str{3});
  hr = str2double(temp_str{7});
  ht = str2double(temp_str{5});
  d = str2double(temp_str{9});
  cw = ceil(AUVModem.beaufortTable(AUVModem.beaufortTable(:, 1) == str2double(temp_str{11}), 2));

  % Multipfadkomponenten schaetzen
  [~, tau, ~, ~, ~, hp] = AUVModem.runGeoEstimation(h, ht, hr, d, AUVParams.CarrierFreq, SSP, cw, AUVModem.d_z(AUVModem.d_z < h));
  [tau, id_s] = sort(tau);
  hp = hp(id_s) * db2mag(AUVParams.transmitsensitivity) * 160;

  % Signalvektor erstellen
  tau_t = round(tau * fs);
  tmp_kas = zeros(1, max(tau_t) + length(kasSeqID) + length(kasSeqDT));
  for i = 1:length(tau_t)
    tmp_kas(tau_t(i):tau_t(i) + length(kasSeqID) + length(kasSeqDT) - 1) = ...
      tmp_kas(tau_t(i):tau_t(i) + length(kasSeqID) + length(kasSeqDT) - 1) + ...
      hp(i) * ([-(kasSeqID * 2 - 1) -(kasSeqDT * 2 - 1)]);
  end
  tmp_kas(tmp_kas >= 0) = 1;
```

```
74    tmp_kas(tmp_kas < 0) = -1;
75
76    % Kreuzkorrelation mit Kasami-Sequenzen durchfuehren
77    [~, id_corrID] = find(abs(xcorr(tmp_kas, -((kasSeqID) * 2 - 1))) >= AUVParams.IDlength * 0.5);
78    [~, id_corrDT] = find(abs(xcorr(tmp_kas, -((kasSeqDT) * 2 - 1))) >= AUVParams.DTlength * 0.45);
79
80    % Erkennung validieren
81    tmp = [];
82    try
83      for i_id = 1:length(id_corrID)
84        if any(id_corrID(i_id) == (id_corrDT - AUVParams.IDlength))
85          tmp = [tmp i_id];
86        end
87      end
88    catch
89      tmp = [];
90    end
91
92    if ~isempty(id_corrID) && ~isempty(id_corrDT) && ~isempty(tmp)
93      try
94        detection_a = (id_corrID(tmp) - length(tmp_kas) + 1);
95      catch
96        detection_a = [];
97      end
98      detection_dt_a = (id_corrDT - length(tmp_kas) + 1);
99      detection = find(detection_a - detection_a(1) > framelength, 1);
100
101     % CINRR berechnen
102     Nsig = mag2db(db2mag(turbo_noise(AUVParams.CarrierFreq)) + ...
103             db2mag(ship_noise(AUVParams.CarrierFreq, 0)) + ...
104             db2mag(surf_noise(AUVParams.CarrierFreq, cw)) + ...
105             db2mag(therm_noise(AUVParams.CarrierFreq)));
106
107     tmp = [detection_a detection(1)];
108     try
109       for i_corr_tmp = tmp
110         i_corr = find(tau_t == i_corr_tmp);
111         tmp_sum = sum(hp(i_corr));
112         hp(i_corr) = 0;
113         hp(i_corr(1)) = tmp_sum;
114         i_corr = i_corr(1);
115
116         if ~isempty(find(detection_dt_a == i_corr_tmp + AUVParams.IDlength, 1))
117           tmp_id = i_corr;
118
119           if length(tau) > tmp_id && tau(tmp_id + 1) - tau(tmp_id) < 1 / fs
120             hp(tmp_id) = hp(tmp_id) + hp(tmp_id + 1) * ((tau(tmp_id + 1) - tau(tmp_id)) / (1 / fs));
121             hp(tmp_id + 1) = hp(tmp_id + 1) - hp(tmp_id + 1) * (1 - (tau(tmp_id + 1) - tau(tmp_id)) / (1 / fs));
122           end
123
124           % SINR berechnen
125           P_LOS = sum(hp(tmp_id).^2);
126           P_Interference = sum((hp(tmp_id + 1:end).^2) .* ...
127             max(0, (framelength / fs) - (tau(tmp_id + 1:end) - tau(tmp_id))) / (framelength / fs));
128
129           if tmp_id > 1
130             P_Interference = P_Interference + sum((hp(1:tmp_id - 1).^2) .* ...
131               max(0, (framelength / fs) + (tau(1:tmp_id - 1) - tau(tmp_id))) / (framelength / fs));
132           end
133
134           cinrr = pow2db(P_LOS) - mag2db(db2mag(pow2db(P_Interference)) + db2mag(Nsig));
135
136           % Gueltigkeit basierend auf SINR ueberpruefen
137           valid = SNR_BER <= round(cinrr + add_comp);
138           if valid
139             break;
140           end
141         end
142       end
143     catch
144       valid = false;
145     end
146   end
147
148   % Gueltige und ungueltige Kanaele speichern
149   if valid
150     geoc = geoc + 1;
151     UWCgeo{geoc} = UWC{iUWC};
152   else
153     georestc = georestc + 1;
154     UWCgeorest{georestc} = UWC{iUWC};
155   end
156 end
```

```
157
158 % Reduktionsrate berechnen und anzeigen
159 red_rate_geo = 1 - geoc / length(UWC);
160 disp(['##### Geo-Reduktionsrate: ', num2str(red_rate_geo)]);
161
162 % Gueltige Kanaele fuer die Simulation auswaehlen
163 if AUVParams.fastSim
164   nc = min(geoc, AUVParams.simCTFs);
165   if exist('UWCgeo', 'var')
166     randIdcs = randperm(length(UWCgeo), nc);
167     UWC = UWCgeo(randIdcs);
168   end
169
170   disp(['##### Anzahl der simulierten CTFs: ', num2str(nc)])
171   if exist('UWCgeorest', 'var') && write
172     resetAUVParams(AUVParams);
173     writeGeorestChannels(UWCgeorest, AUVParams, fileNameCSV, geoc, nc);
174   end
175 else
176   nc = geoc;
177   UWC = UWCgeo;
178
179   disp(['##### Anzahl der simulierten CTFs: ', num2str(nc)])
180   if exist('UWCgeorest', 'var') && write
181     resetAUVParams(AUVParams);
182     writeGeorestChannels(UWCgeorest, AUVParams, fileNameCSV, geoc, nc);
183   end
184 end
185
186 % AUVParams mit den Reduktionsresultaten aktualisieren
187 AUVParams.red_rate_geo = red_rate_geo;
188 AUVParams.red_rate = red_rate_geo;
189 AUVParams.nc = nc;
190 AUVParams.geoc = geoc;
191 disp('##### Geometrische Reduktion abgeschlossen!')
192 end
193
194 function resetAUVParams(AUVParams)
195 % resetAUVParams - AUV-Parameter fuer geometrische Reduktion zuruecksetzen.
196 AUVParams.throughput = 0;
197 AUVParams.error = 1000;
198 AUVParams.ber = 1;
199 AUVParams.decoded = 0;
200 AUVParams.trans = 0;
201 AUVParams.throughputDATA = 0;
202 AUVParams.throughputRTSCTS = 0;
203 AUVParams.throughputBC = 0;
204 end
205
206 function writeGeorestChannels(UWCgeorest, AUVParams, fileNameCSV, geoc, nc)
207 % writeGeorestChannels - Georest-Kanaele in eine CSV-Datei schreiben.
208 for iUWC = 1:(length(UWCgeorest) * nc / geoc)
209   temp_str = split(UWCgeorest{iUWC}, "_");
210   AUVParams.waterdepth = str2double(temp_str{3});
211   AUVParams.hr0 = str2double(temp_str{7});
212   AUVParams.ht0 = str2double(temp_str{5});
213   AUVParams.d0 = str2double(temp_str{9});
214   AUVParams.seastate = str2double(strrep(temp_str{11}, ".mat", ""));
215   AUVParams = AUVModem.writeCSV(false, false, AUVParams, fileNameCSV);
216 end
217 end
```

Listing A.17: Bestimmung der analytischen Reduktion

```
1  function [AUVParams, UWC] = anaReduction(AUVParams, UWC, fileNameCSV, channel, SNR_BER, write, kasSeqID, kasSeqDT)
2  % anaReduction - Fuehrt eine analytische Reduktion basierend auf den gegebenen AUV-Parametern, UWC-Kanaelen
3  %  und weiteren Eingaben durch. Die Funktion berechnet das Signal-Rausch-Verhaeltnis (SINR) fuer jeden Kanal
4  %  und filtert Kanaele heraus, die die spezifizierten Kriterien nicht erfuellen.
5  % Syntax: [AUVParams, UWC] = anaReduction(AUVParams, UWC, fileNameCSV, channel, SNR_BER, write, kasSeqID, kasSeqDT)
6  %
7  % Eingaben:
8  %  AUVParams - Struktur mit den AUV-Parametern.
9  %  UWC - Zell-Array von Unterwasser-Kanal-Dateien.
10 %  fileNameCSV - Name der CSV-Datei fuer die Ausgabe.
11 %  channel - Kanal-Identifikator.
12 %  SNR_BER - Signal-Rausch-Verhaeltnis fuer die Bitfehlerquote.
13 %  write - (Optional) Boolesches Flag, um Ergebnisse in CSV zu schreiben (Standard: true).
14 %  kasSeqID - KAS-Sequenz-ID.
15 %  kasSeqDT - KAS-Sequenz-DT.
16 %
17 % Ausgaben:
```

```
18 %  AUVParams - Aktualisierte Struktur mit AUV-Parametern.
19 %  UWC - Zell-Array von reduzierten Unterwasser-Kanal-Dateien.
20
21 % Standardwert fuer 'write' setzen, falls nicht bereitgestellt
22 if nargin < 6
23   write = true;
24 end
25
26 % Zaehler initialisieren und Startnachricht anzeigen
27 anac = 0;
28 anarestc = 0;
29 disp('*** Start der analytischen Reduktion!')
30
31 % Vorherige Reduktionsergebnisse loeschen
32 clear UWCana
33
34 % Rauschfunktionen definieren
35 turbo_noise = @(x) (17 - 30 * log10(x / 1000)); % Turbulenzrauschen
36 ship_noise = @(x, ss) (40 + 20 * (ss - 0.5) + 26 * log10(x / 1000) - 60 * log10(x / 1000 + 0.03)); % Schiffsrauschen
37 surf_noise = @(x, sw) (50 + 7.5 * sqrt(sw) + 20 * log10(x / 1000) - 40 * log10(x / 1000 + 0.44)); %
       Oberflaechenrauschen
38 therm_noise = @(x) (-15 + 20 * log10(x / 1000)); % Thermisches Rauschen
39
40 % Zusaetzlicher Kompensationswert
41 add_comp = 3;
42
43 % Rahmenlaenge berechnen
44 framelength = (AUVModem.dataLength + AUVParams.d + AUVParams.k) / AUVParams.r * 8 / log2(AUVParams.M);
45
46 % Abtastfrequenz basierend auf dem Modulationstyp bestimmen
47 if AUVParams.ModulationActivEnum == CtrModulation.QPSKModulation || AUVParams.ModulationActivEnum == CtrModulation.
       BPSKModulation
48   fs = AUVParams.BandWidth;
49 else
50   if AUVParams.BandWidth == 1000
51     bb = 1;
52   else
53     bb = 2;
54   end
55   fs = AUVParams.bits_per_second(AUVParams.num_channels, log2(AUVParams.M), bb);
56 end
57
58 % Traegerfrequenz
59 fc = AUVParams.CarrierFreq;
60
61 % Jeden Unterwasser-Kanal verarbeiten
62 for iUWC = 1:length(UWC)
63   % Kanal-Daten laden
64   cd([AUVParams.channelfiles filesep 'FC_' num2str(AUVParams.CarrierFreq) '_BW_' num2str(AUVParams.BandWidth)
       filesep channel])
65   load(UWC{iUWC});
66   valid = false;
67
68   % Jeden CIR analysieren
69   for i_cir = 1:size(CIRM_BELLHOP_SBP_doppler, 1)
70     % Signal-Rausch-Verhaeltnis berechnen
71     Nsig = mag2db(db2mag(turbo_noise(AUVParams.CarrierFreq)) + ...
72             db2mag(ship_noise(AUVParams.CarrierFreq, 0)) + ...
73             db2mag(surf_noise(AUVParams.CarrierFreq, c_wind)) + ...
74             db2mag(therm_noise(AUVParams.CarrierFreq)));
75     % Signifikante Bereiche des CIR finden
76     range_cir = find(mag2db(CIRM_BELLHOP_SBP_doppler(i_cir, :) * ...
77                 db2mag(AUVParams.transmitsensitivity) * 160) > ...
78                 max(mag2db(CIRM_BELLHOP_SBP_doppler(i_cir, :) * ...
79                       db2mag(AUVParams.transmitsensitivity) * 160)) - 50);
80     if isempty(range_cir)
81       tmp_bell = [0 0 0 0];
82     else
83       tmp_bell = CIRM_BELLHOP_SBP_doppler(i_cir, max(1, range_cir(1)) - 1 : range_cir(end)) * ...
84             db2mag(AUVParams.transmitsensitivity) * 160;
85     end
86
87     [val_tmp, tau] = findpeaks(tmp_bell);
88     tau_tmp = tau / 160000;
89     tau_tmp = tau_tmp(mag2db(val_tmp) > max(mag2db(val_tmp)) - 40);
90     val_tmp = val_tmp(mag2db(val_tmp) > max(mag2db(val_tmp)) - 40);
91     tau_t = ceil(tau_tmp * fs);
92     tmp_kas = zeros(1, max(tau_t) + length(kasSeqID) + length(kasSeqDT));
93     for i = 1:length(tau_t)
94       tmp_kas(tau_t(i) : tau_t(i) + length(kasSeqID) + length(kasSeqDT) - 1) = ...
95         tmp_kas(tau_t(i) : tau_t(i) + length(kasSeqID) + length(kasSeqDT) - 1) + ...
96         val_tmp(i) * ([-(kasSeqID * 2 - 1) -(kasSeqDT * 2 - 1)]);
97     end
```

```
98
99      tmp_kas(tmp_kas >= 0) = 1;
100     tmp_kas(tmp_kas < 0) = -1;
101     [~, id_corrID] = find(abs(xcorr(tmp_kas, -((kasSeqID) * 2 - 1))) >= AUVParams.IDlength * 0.5);
102     [~, id_corrDT] = find(abs(xcorr(tmp_kas, -((kasSeqDT) * 2 - 1))) >= AUVParams.DTlength * 0.45);
103
104     if ~isempty(id_corrID) && ~isempty(id_corrDT)
105       id_tmp = 1;
106       t = 1;
107       while tau_tmp(t) - tau_tmp(1) < 1 / fs * AUVParams.IDlength / 16 && t < length(val_tmp) && val_tmp(t) >
          val_tmp(1)
108         if val_tmp(t - 1) < val_tmp(t)
109           id_tmp = [id_tmp t];
110         end
111         t = t + 1;
112       end
113
114       try
115         id_tmp = find(tau_t == (id_corrID - length(tmp_kas) + 1));
116       catch
117         id_tmp = 1;
118       end
119
120       id_tmp = [id_tmp find(tau_tmp - tau_tmp(id_tmp(1)) > framelength, 1)];
121
122       for id = unique(id_tmp)
123         valid = true;
124         P_LOS = sum((val_tmp(id)) .^ 2);
125
126         P_Interference = [];
127         if id < length(val_tmp)
128           P_Interference = sum(((val_tmp(max(id) + 1:end)) .^ 2) .* ...
129             (max(0, (framelength / fs) - max(0, tau_tmp(max(id) + 1:end) - tau_tmp(id))) / (framelength / fs)));
130         end
131         if id > 1 && id < length(val_tmp)
132           P_Interference = P_Interference + sum(((val_tmp(max(1, id - framelength):id - 1)) .^ 2) .* ...
133             (max(0, (framelength / fs) + (tau_tmp(max(1, id - framelength):id - 1) - tau_tmp(id))) / (framelength /
          fs)));
134         end
135
136         if isempty(P_Interference)
137           cinrr = pow2db(P_LOS) - Nsig;
138         else
139           cinrr = pow2db(P_LOS) - mag2db(db2mag(pow2db(P_Interference)) + db2mag(Nsig));
140         end
141
142         if (SNR_BER) > round(cinrr + add_comp)
143           valid = false;
144         end
145
146         if valid
147           break;
148         end
149       end
150
151       if valid
152         break;
153       end
154     else
155       valid = false;
156     end
157   end
158
159   if valid
160     anac = anac + 1;
161     UWCana{anac} = UWC{iUWC};
162   else
163     anarestc = anarestc + 1;
164     UWCanarest{anarestc} = UWC{iUWC};
165   end
166 end
167
168 % Reduktionsrate berechnen und anzeigen
169 red_rate_ana = 1 - anac / length(UWC);
170 disp(['##### Reduktionsrate der analytischen Reduktion: ', num2str(red_rate_ana)]);
171
172 % Zufaellig gueltige Kanaele auswaehlen, wenn schnelle Simulation aktiviert ist
173 if AUVParams.fastSim
174   nc = min(anac, AUVParams.simCTFs); % Anzahl der zufaelligen CTFs
175   if exist('UWCana', 'var')
176     randIdcs = randperm(length(UWCana), nc);
177     UWC = cell(1, length(randIdcs));
178     for ic = 1:length(randIdcs)
```

```
179         UWC{ic} = UWCana{randIdcs(ic)};
180       end
181     end
182
183     if exist('UWCanarest', 'var') && write
184       AUVParams = resetAUVParams(AUVParams);
185       for iUWC = 1:(length(UWCanarest) * nc / anac)
186         temp_str = split(UWCanarest{iUWC}, "_");
187         AUVParams = updateAUVParams(AUVParams, temp_str);
188         AUVParams = AUVModem.writeCSV(false, false, AUVParams, fileNameCSV);
189       end
190     end
191   else
192     nc = anac;
193     if exist('UWCana', 'var')
194       UWC = UWCana;
195     end
196
197     if exist('UWCanarest', 'var') && write
198       AUVParams = resetAUVParams(AUVParams);
199       for iUWC = 1:length(UWCanarest)
200         temp_str = split(UWCanarest{iUWC}, "_");
201         AUVParams = updateAUVParams(AUVParams, temp_str);
202         AUVParams = AUVModem.writeCSV(false, false, AUVParams, fileNameCSV);
203       end
204     end
205   end
206
207   disp(['##### Anzahl der simulierten CTFs: ', num2str(nc)]);
208   AUVParams.red_rate_ana = red_rate_ana;
209   AUVParams.red_rate = red_rate_ana;
210   AUVParams.nc = nc;
211   AUVParams.anac = anac;
212
213   disp('*** Stop der analytischen Reduktion!')
214   end
215
216   function AUVParams = resetAUVParams(AUVParams)
217   % resetAUVParams - Setzt die AUV-Parameter fuer die schnelle Simulation zurueck.
218   AUVParams.throughput = 0;
219   AUVParams.error = 1000;
220   AUVParams.ber = 1;
221   AUVParams.decoded = 0;
222   AUVParams.trans = 0;
223   AUVParams.throughputDATA = 0;
224   AUVParams.throughputRTSCTS = 0;
225   AUVParams.throughputBC = 0;
226   end
227
228   function AUVParams = updateAUVParams(AUVParams, temp_str)
229   % updateAUVParams - Aktualisiert die AUV-Parameter aus dem Dateinamen-String.
230   AUVParams.waterdepth = str2double(temp_str{3});
231   AUVParams.hr0 = str2double(temp_str{7});
232   AUVParams.ht0 = str2double(temp_str{5});
233   AUVParams.d0 = str2double(temp_str{9});
234   AUVParams.seastate = str2double(strrep(temp_str{11}, ".mat", ""));
235   end
```

Listing A.18: Bestimmung der stochastischen Reduktion

```
1   function [AUVParams, UWC] = noneReduction(AUVParams, UWC, ~)
2   % noneReduction - Kiene Reduktion der Kanaele durchfuehren und die Eingabedaten unveraendert zurueckgeben.
3   % Diese Funktion gibt die Eingabedaten ohne aenderungen zurueck. Falls eine schnelle Simulation aktiviert ist,
4   % wird eine zufaellige Auswahl von Kanaelen basierend auf der angegebenen Anzahl der Kanaele durchgefuehrt.
5
6   % Anzeige der Nachricht, die den Beginn des Prozesses "Keine Reduktion" anzeigt
7   disp('##### Keine Reduktion!')
8
9   % Temporaere Variable UWCnone loeschen, falls vorhanden (zur Konsistenz)
10  clear UWCnone
11
12  % Reduktionsrate auf 0 setzen, da keine Reduktion durchgefuehrt wird
13  red_rate_none = 0;
14  % Formatieren und Anzeigen der Nachricht zur Reduktionsrate
15  X = sprintf('##### Reduktionsrate: %d', red_rate_none);
16  disp(X);
17
18  %% Zufaellige Auswahl von Kanaelen, wenn eine schnelle Simulation aktiviert ist
19  if AUVParams.fastSim && AUVParams.simCTFs < length(UWC)
20    % Anzahl der Kanaele, die zufaellig ausgewaehlt werden sollen
21    nc = AUVParams.simCTFs;
```

```
22
23    % Zufaellige Indizes fuer die Kanalauswahl generieren
24    randIdcs = randperm(length(UWC), nc);
25    UWC_tmp = UWC; % Original-UWC temporaer speichern
26    clear UWC; % Original-UWC loeschen, um die ausgewaehlten Kanaele neu zuzuweisen
27
28    % Die zufaellig ausgewaehlten Kanaele der Variable UWC zuweisen
29    for ic = 1:length(randIdcs)
30      UWC{ic} = UWC_tmp{randIdcs(ic)};
31    end
32  else
33    % Wenn keine zufaellige Auswahl erforderlich ist, alle verfuegbaren Kanaele verwenden
34    nc = length(UWC);
35  end
36
37  % AUVParams mit der Reduktionsrate und der Anzahl der ausgewaehlten Kanaele aktualisieren
38  AUVParams.red_rate_none = red_rate_none;
39  AUVParams.red_rate = red_rate_none;
40  AUVParams.nc = nc;
41
42  % Formatieren und Anzeigen der Nachricht zur Anzahl der simulierten CTFs
43  X = sprintf('##### Anzahl der simulierten CTFs: %d', nc);
44  disp(X);
45
46  % Anzeige der Nachricht, die das Ende des Prozesses "Keine Reduktion" anzeigt
47  disp('##### Ende der Keine Reduktion!')
48
49  end
```

Listing A.19: Bestimmung der Entwurfsraumreduktion

```
1   function [flag_back, RTSCTS_throughput_kasami, cinrr, lg_t, matrixHP, matrixTAU] = capacitybasedReduction(mode, h0_a
        , maxDistance, SNR_BER, AUVParams, RTSCTS_throughput_req, matfolder_string, SSP)
2   % Diese Funktion berechnet die Kapazitaetsreduzierung basierend auf verschiedenen Parametern und Anforderungen.
3   % Die Ergebnisse beinhalten die Rueckgabeflagge, die Durchsatzraten, CINRR-Werte und Matrizen fuer die Geometrie.
4
5   % Globale Variablen
6   global cBottom densitySurface densityBottom %#ok<GVMIS>
7
8   % Parameter fuer Abstands- und Hoehenvariablen
9   diff_d0 = 200; % Schrittweite im Abstand
10  z = AUVModem.d_z; % Array der Hoehenwerte
11  ht0_a = z(z < h0_a); % TX-Hoehe [m]
12  hr0_a = z(z < h0_a); % RX-Hoehe [m]
13  d0_a = (100:diff_d0:maxDistance); % Kanalabstand [m]
14  cw_a = 0:1:28; % Windgeschwindigkeit ueber dem Wasser
15
16  % Geschwindigkeit des Schalls in Wasser und Dichten
17  c_top = SSP(1); % Geschwindigkeit des Schalls in Wasser (oberflaeche)
18  c_bot = SSP(end); % Geschwindigkeit des Schalls in Wasser (bodennaehe)
19  c = sum(SSP)/length(SSP); % Geschwindigkeit des Schalls in Wasser
20  cBottom = 1800; % Geschwindigkeit des Schalls in Boden
21  densitySurface = 1; % Dichte des Oberflaechenwassers
22  densityBottom = 1.8; % Dichte des Bodenwassers
23  fc = AUVParams.CarrierFreq; % Traegerfrequenz
24
25  % Bandbreite und Modulationseinstellungen
26  if AUVParams.ModulationActivEnum == CtrModulation.QPSKModulation || AUVParams.ModulationActivEnum == CtrModulation.
        BPSKModulation
27    fs = AUVParams.BandWidth; % Bandbreite
28  else
29    if AUVParams.BandWidth == 1000
30      bb = 1;
31    else
32      bb = 2;
33    end
34    fs = AUVParams.bits_per_second(AUVParams.num_channels, log2(AUVParams.M), bb); % Bits pro Sekunde
35  end
36
37  % Generierung der Geometrie-Matrizen
38  [matrixHP, matrixTAU, lengthM, lg_t] = generateGeoMatrix(h0_a, ht0_a, hr0_a, d0_a, fc, SNR_BER, c, c_bot, cBottom,
        cw_a, matfolder_string, SSP);
39
40  % aendere den Ansatz von milcom23 auf 90 % Quantil
41  lg = ceil(((quantile(lg_t, 0.75) - quantile(lg_t, 0.25)) * 1.5 + quantile(lg_t, 0.75)) * fs);
42  cinrr_min = inf;
43  cinrr_max = -inf;
44
45  % Berechne die Laenge von Daten- und Steuerpaketen
46  framelength = ((AUVParams.d + AUVModem.dataLength + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M) + lg +
        AUVParams.IDlength + AUVParams.DTlength;
```

```
47 [cinrr] = calculateSingleCINRR(matrixHP, matrixTAU, lengthM, fs, fc, ht0_a, hr0_a, d0_a, framelength);
48 cinrr_min = min(cinrr_min, min(cinrr)); %#ok<NASGU>
49 cinrr_max = max(cinrr_max, max(cinrr)); %#ok<NASGU>
50
51 % Berechne die Distanzen
52 distance_a = [];
53 for i = ht0_a
54   for ii = hr0_a
55     for iii = d0_a
56       distance_a = [distance_a, sqrt((i - ii)^2 + iii^2)]; %#ok<AGROW>
57     end
58   end
59 end
60
61 % Paketlaengenberechnungen
62 length_DATA = (((AUVParams.d + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M));
63 length_RTS = (((20 + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M));
64 length_CTS = (((14 + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M)); % Gleich wie ACK
65 length_DATA_kas = (((AUVParams.d + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M));
66 length_RTS_kas = (((10 + AUVParams.k) / AUVParams.r) * 8 / log2(AUVParams.M));
67
68 % Laenge lg anpassen
69 if isempty(lg);  lg = 0;end
70
71 % Durchsatzberechnungen
72 RTSCTS_throughput_kasami = (((AUVParams.d - 11) * 8) ./ ((length_DATA_kas + length_RTS_kas + 4 * ((AUVParams.
       DTlength + AUVParams.IDlength) + 2 * lg)) / fs + 4 * distance_a / c));
73
74 % ueberpruefe, ob die Durchsatzanforderungen erfuellt sind
75   if ((cinrr(1) < SNR_BER || (sum(RTSCTS_throughput_kasami) / length(RTSCTS_throughput_kasami)) <
       RTSCTS_throughput_req) && strcmp(mode, 'stat')) || ...
76     ((cinrr(3) < SNR_BER || max(RTSCTS_throughput_kasami) < RTSCTS_throughput_req) && strcmp(mode, 'dyn'))
77     flag_back = false;
78   else
79     flag_back = true;
80   end
81
82 end
```

A.8 Dokumententypedefinition (DTD)

Listing A.20: Dokumententypedefinition

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <!-- Definition des Elements 'modemgraph', das alle anderen Elemente enthaelt -->
3  <!ELEMENT modemgraph (componentgraph, variantlist, mappings, dependencies, typedef*)>
4  <!ATTLIST modemgraph name CDATA #REQUIRED <!-- Name des Modemgraphen --> >
5
6  <!-- Definition des Elements 'componentgraph', Informationen ueber Komponenten -->
7  <!ELEMENT componentgraph (component)+>
8  <!ATTLIST componentgraph id ID #REQUIRED <!-- Eindeutige ID des Komponentengraphen --> >
9
10 <!-- Definition des Elements 'variantlist', das Varianteninformationen enthaelt -->
11 <!ELEMENT variantlist (variant)+>
12 <!ATTLIST variantlist id ID #REQUIRED <!-- Die eindeutige ID der Variantenliste --> >
13
14 <!-- Definition des Elements 'mappings', Mappings zw. Varianten und Komponenten -->
15 <!ELEMENT mappings (mapping)+>
16 <!ATTLIST mappings id ID #REQUIRED <!-- Die eindeutige ID der Zuordnungen --> >
17
18 <!-- Definition des Elements 'dependencies', Abhaengigkeiten zwischen Komponenten -->
19 <!ELEMENT dependencies (dependency)*>
20 <!ATTLIST dependencies id ID #REQUIRED <!-- Die eindeutige ID der Abhaengigkeiten --> >
21
22 <!-- Definition des Elements 'variant', das die Attribute einer Variante enthaelt -->
23 <!ELEMENT variant (fixattribute|rangeattribute|selectattribute)*>
24 <!ATTLIST variant id ID #REQUIRED <!-- Die eindeutige ID der Variante --> >
25
26 <!-- Definition des Elements 'component', enthaelt die Attribute einer Komponente -->
27 <!ELEMENT component (fixattribute|rangeattribute|selectattribute)*>
28 <!ATTLIST component id ID #REQUIRED <!-- Die eindeutige ID der Komponente --> >
29
30 <!-- Definition des Elements 'fixattribute', das ein festes Attribut darstellt -->
31 <!ELEMENT fixattribute EMPTY>
```

```
<!ATTLIST fixattribute
name CDATA #REQUIRED <!-- Der Name des Attributs -->
value CDATA #REQUIRED <!-- Der Wert des Attributs -->
type (INTEGER|DOUBLE) #REQUIRED <!-- Der Datentyp des Attributs --> >

<!-- Definition des Elements 'rangeattribute', das ein Bereichsattribut darstellt -->
<!ELEMENT rangeattribute EMPTY>
<!ATTLIST rangeattribute
name CDATA #REQUIRED <!-- Der Name des Attributs -->
default CDATA #REQUIRED <!-- Der Standardwert des Attributs -->
max CDATA #REQUIRED <!-- Der maximale Wert des Attributs -->
min CDATA #REQUIRED <!-- Der minimale Wert des Attributs -->
stepsize CDATA #REQUIRED <!-- Die Schrittgroesse des Attributs -->
type (INTEGER|DOUBLE) #REQUIRE <!-- Der Datentyp des Attributs --> >

<!-- Definition des Elements 'selectattribute', das ein Auswahlattribut darstellt -->
<!ELEMENT selectattribute EMPTY>
<!ATTLIST selectattribute
name CDATA #REQUIRED <!-- Der Name des Attributs -->
value CDATA #REQUIRED <!-- Der Wert(e) des Attributs -->
type (INTEGER|DOUBLE) #REQUIRED <!-- Der Datentyp des Attributs --> >

<!-- Definition des Elements 'mapping', das ein Mapping zwischen Varianten und Komponenten darstellt -->
<!ELEMENT mapping EMPTY>
<!ATTLIST mapping
id ID #REQUIRED <!-- Die eindeutige ID des Mappings -->
source IDREF #REQUIRED <!-- Die ID der Quellvariante -->
target IDREF #REQUIRED <!-- Die ID der Zielkomponente --> >

<!-- Definition des Elements 'dependency', das eine Abhaengigkeit zwischen Komponenten darstellt -->
<!ELEMENT dependency EMPTY>
<!ATTLIST dependency
id ID #REQUIRED <!-- Die eindeutige ID der Abhaengigkeit -->
source IDREF #REQUIRED <!-- Die ID der Quellkomponente --> >

<!-- Definition des Elements 'typedef', das die Typdefinitionen enthaelt -->
<!ELEMENT typedef EMPTY>
<!ATTLIST typedef
name ID #REQUIRED <!-- Der Name des Typs -->
type (STRING|INTEGER|DOUBLE|BOOLEAN|LONG|ATTRIBUTE) #REQUIRED <!-- Der Datentyp --> >
```

A.9 Unit-Tests

Beispiel A.1 — Unit-Test (Block-Codierer):
Die Validierung der Komponente Block-Codierer wird mittels des in Abbildung 5.22 abgebildeten Referenz-Modells, der Unit-Test-Funktion in Listing A.21 und deren zugehörigen Unterfunktionen basierend auf den Testparametern in Tabelle 5.1 für Fließ- und Festkommazahlen durchgeführt.

Nach dem Initialisieren der vorgegebenen Parameter wird das variantenspezifisch geladene Referenz-Modell des Block-Kodierers, basierend auf den definierten Parametern, simuliert (Listing A.22). Nach der erfolgreichen Simulation werden die gesendeten Signale (*out.transmitterIn*) mit denen der Sender angesteuert wird mit den empfangenen Datensignalen verglichen(Listing A.23), um die in Gl. (5.8) bis 5.10 definierten Bedingungen auf Gültigkeit zu prüfen.

Nach der erfolgreichen Prüfung, werden die Referenzfunktion des Senders und Empfängers unter Verwendung identischer Datensignale validiert (Listing A.24).

Die Validierung gilt als erfolgreich wenn die ausgehenden Signale des Senders und Empfängers jeweils numerisch und zeitlich identisch sind.

Nach der Simulation wird nur für Test mit Festkommazahlen das Referenzmodell nach einer erfolgreichen Hardwarebeschreibungssprache (HDL)-Code-Erzeugung gegen ein automatisch generiertes Validierungs-Modell für die Festkommazahlen, welche die zeitlichen Abweichungen der korrespondierenden HDL-Umsetzungen implementiert, getestet validiert. Nach der Generierung wird die Komponente zusätzlich über ein erzeugtes FPGA in the Loop (FIL)-Model validiert womit nach einer erfolgreichen Testung gezeigt wird das Referenzfunktion, Referenz-Modell und die Hardwareumsetzung (für die genutzte Plattform) zeitlich und numerisch identisch sind (Listing A.25).

Listing A.21: Unit-Test des Block-Codierer

```
function BlockCoder(testCase, k, d, fixpoint, VariantsBC, sendState, speedup)
  % Function: BlockCoder
  % This function configures and runs simulations for different block coding
  % schemes (CRC or ReedSolomon) based on the specified parameters.
  %
  % Inputs:
  %  - testCase: The test case object for unit tests.
  %  - k: Integer parameter for block coding, related to the message length.
  %  - d: Integer parameter for block coding, related to the block length.
  %  - fixpoint: Boolean indicating whether fixpoint operations should be used.
  %  - VariantsBC: String indicating the variant of block coding ('CRC' or 'ReedSolomon').
  %  - sendState: Integer state indicating the type of frame to be sent. Must be one of:
  %    3 - 'RTS' (Request to Send)
  %    4 - 'CTS' (Clear to Send)
  %    5 - 'Data'
  %    6 - 'ACK' (Acknowledgement)
  %  - speedup: Numeric parameter to speed up the simulation.
  %
  % Outputs:
  %  This function does not return any outputs, but it performs the following actions:
  %  - Configures simulation parameters in the global AUVParams structure.
  %  - Runs the simulation and verifies the results.
  %  - Optionally generates, validates, and runs HDL for fixpoint operations.

  % Load the default parameter set from the base workspace
  AUVParams = evalin('base', 'AUVParams');

  % Constraint definition for unit-tests
  % Ensure the input parameters meet the specific conditions
  testCase.assumeTrue(((sendState == 3 && d == 20) || ...
            ((sendState == 4 || sendState == 6) && d == 14) || ...
            (sendState == 5 && d > 20)) && ...
            ((k == 4 && strcmp(VariantsBC, 'CRC')) || ...
            strcmp(VariantsBC, 'ReedSolomon')));

  % Frame type definition based on sendState
  switch sendState
    case 3
      frameType = 'RTS'; % Request to Send
    case 4
      frameType = 'CTS'; % Clear to Send
    case 5
      frameType = 'Data'; % Data frame
    case 6
      frameType = 'ACK'; % Acknowledgement
    otherwise
      error('Input frame type must be one of ''RTS'', ''CTS'', ''ACK'', ''Data''.');
  end

  sc = -2; % Effects only fixpoint type

  % Data type definition
  initDataTypes;

  % Define model parameters
  AUVParams.d = min(255 - k, d);
```

```
57    AUVParams.k = k;
58    AUVParams.speedupRS = speedup;
59    count = ceil(255 * 8 * speedup) * 4;
60
61    % Define variant
62    mdl = VariantsBC;
63
64    % Set active for transmitter and receiver
65    AUVParams.BlockEncoderActivEnum = CtrBlockEncoder.([mdl 'Encoder']);
66    AUVParams.BlockDecoderActivEnum = CtrBlockDecoder.([mdl 'Decoder']);
67
68    dataDelay = 0; % Delay for first simulation
69
70    % Run simulation
71    TestVariants;
72
73    % Verify: f(x) = f^(-1)(f(x))
74    VerifyFunction;
75
76    % Generate signals based on stimulation
77    subscript_BlockCoder;
78
79    % Verify the transmitter output
80    side = 'transmitter';
81    VerifyReference;
82
83    % Verify the receiver output
84    side = 'receiver';
85    VerifyReference;
86
87    % Generate, validate and run HDL for transmitter and receiver if fixpoint is true
88    if fixpoint
89      side = 'transmitter';
90      generateHDLVariants;
91      side = 'receiver';
92      generateHDLVariants;
93    end
94  end
```

Listing A.22: Unit-Test des Block-Codierer (TestVariants)

```
1   act=pwd;
2   cd ([AUVParams.uNetSimRootFolder filesep 'tmp'])
3   load_system(mdl);
4   set_param(mdl, 'SimCompilerOptimization', 'on');
5   set_param(mdl, 'AccelVerboseBuild', 'on');
6   set_param(mdl, 'SimulationMode', 'accelerator');
7   if ~exist('time');    time=(count*1e-4);end
8   set_param(mdl, 'StopTime', num2str(time));
9
10  %% load variable in the workspace from definition
11  if ~isfield(AUVParams,'numberOfFrames');   AUVParams.numberOfFrames=1;end
12
13  in = Simulink.SimulationInput(mdl);
14  in = in.setVariable('AUVParams',evalin('caller', 'AUVParams' ),'Workspace',mdl);
15  in = in.setVariable('count',evalin('caller', 'count' ),'Workspace',mdl);
16  in = in.setVariable('sendState',evalin('caller', 'sendState' ),'Workspace',mdl);
17  in = in.setVariable('dataDelay',evalin('caller', 'dataDelay' ),'Workspace',mdl);
18
19  %% add data for dataframer (if exist)
20  if exist('kasSeqID','var')
21   in = in.setVariable('kasSeqID',evalin('caller', 'kasSeqID' ),'Workspace',mdl)
22  end
23  if ~isfield(AUVParams,'numberOfFrames')
24    AUVParams.numberOfFrames=1;
25    in = in.setVariable('AUVParams.numberOfFrames',evalin('caller', 'AUVParams.numberOfFrames' ),'Workspace',mdl);
26  end
27  if exist('kasSeqDT','var')
28    in = in.setVariable('kasSeqDT',evalin('caller', 'kasSeqDT' ),'Workspace',mdl);
29  end
30  if exist('genPeambleID','var')
31    in = in.setVariable('genPeambleID',evalin('caller', 'genPeambleID' ),'Workspace',mdl);
32  end
33  if exist('Threshold_ID','var')
34    in = in.setVariable('Threshold_ID',evalin('caller', 'Threshold_ID' ),'Workspace',mdl);
35  end
36  if exist('Threshold_DT','var')
37    in = in.setVariable('Threshold_DT',evalin('caller', 'Threshold_DT' ),'Workspace',mdl);
38  end
39  out = sim(in);
40  cd (act)
```

Listing A.23: Unit-Test des Block-Codierer (VerifyFunction)

```
[~, idx]=max(xcorr(double(out.receiverOut),double(out.transmitterIn)));
dataDelay=idx-size(out.transmitterIn,1);
testCase.verifyEqual(out.receiverOut(dataDelay+1:end)',out.transmitterIn(1:end-dataDelay)', sprintf(...
  ("Function test (f(x)=f^(-1)(f(x))) for Variant " + mdl + " with:\nd: [%d]\nk: [%d]\nsendState: [%d].") ,...
  (AUVParams.d),(AUVParams.k),(sendState)));
disp(['*** Verify function of ' mdl ' simulation ---- passed'])
```

Listing A.24: Unit-Test des Block-Codierer (VerifyReference)

```

if strcmp(side,'transmitter')
  if ( iscolumn( txRef ) ); txRef = txRef.'; end
  if ( iscolumn( out.transmitterOut ) ) out.transmitterOut = out.transmitterOut.'; end
  [~, idx]=max(xcorr(double(out.transmitterOut),double(txRef)));
  dataDelay=idx-max(size(out.transmitterOut,2),size(txRef,2));
  tmp_ref=(txRef);
  clear txRef
  txRef=tmp_ref(1:endIdx);
  testCase.verifyEqual((txRef(1:min(length(txRef),length(out.transmitterOut(dataDelay+1:end))))),...
    out.transmitterOut(dataDelay+1:dataDelay+length(txRef(1:min(length(txRef),...
    length(out.transmitterOut(dataDelay+1:end)))))), sprintf(...
    ("Referencetest of transmitter for variant " + mdl + " with:\nd: [%d]\nk: [%d]\nsendState: [%d].") ,...
    (AUVParams.d), (AUVParams.k),(sendState)));
  disp("*** Verify the transmitter (simulation) against the reference function in AUVModem --- passed")
else
  if ( iscolumn( rxRef ) ); rxRef = rxRef.'; end
  if ( iscolumn( out.receiverOut ) ); out.receiverOut = out.receiverOut.'; end
  [~, idx]=max(xcorr(out.receiverOut,rxRef));
  dataDelay=idx-size(out.receiverOut,2);
  testCase.verifyEqual((rxRef(1:min(length(rxRef),length(out.receiverOut(dataDelay+1:end))))),...
    out.receiverOut(dataDelay+1:dataDelay+length(rxRef(1:min(length(rxRef),...
    length(out.receiverOut(dataDelay+1:end)))))), sprintf(...
    ("Referencetest of receiver for variant " +mdl+ " with:\nd: [%d]\nk: [%d]\nsendState: [%d].") ,...
    (AUVParams.d), (AUVParams.k), (sendState)));
    disp("*** Verify the receiver (simulation) against the reference function in AUVModem ---- passed")
end
```

Listing A.25: Unit-Test des Block-Codierer (generateHDLVariants)

```
genHDL=evalin('base','genHDL');
if exist('genHDL','var')&&genHDL==true
  disp("*** Verify HDL generation of the " +side);
  target=[mdl '/' side];
  targetdir=[AUVParams.uNetSimRootFolder filesep 'generatedHDL' filesep side ];
  targetadd=['d' num2str(AUVParams.d) 'k' num2str(AUVParams.k) 'ss' num2str(sendState) 'fc' num2str(AUVParams.
      CarrierFreq/1000) 'B' num2str(AUVParams.BandWidth/1000) 'SU' num2str(AUVParams.speedupRS)];
  assignin('base','count',count)
  assignin('base','dataDelay',0)
  assignin('base','sendState',sendState)
  assignin('base','AUVParams',AUVParams)
  if isfield( AUVParams, 'IDlength')
    targetadd=[targetadd '_IDL' num2str(AUVParams.IDlength) '_DTL' num2str(AUVParams.DTlength)];
  end
  if exist('LowpassActivEnum','var')
    assignin('base','LowpassActivEnum',LowpassActivEnum);
  end
  if exist('genPeambleID','var')
    assignin('base','genPeambleID',genPeambleID);
  end
  if exist('kasSeqID','var')
    assignin('base','kasSeqID',kasSeqID);
  end
  if exist('kasSeqDT','var')
    assignin('base','kasSeqDT',kasSeqDT);
  end

  disp("Generate HDL for " +target+ " in folder " + [targetdir filesep targetadd] + "." )

    makehdl(target,'TargetDirectory',[targetdir filesep targetadd],'GenerateValidationModel','on','UserComment','
      This is automatically generated by Marcel Riess with a unit tests for HDL-Verification.\n')
ReadOutResourcesHLS;
% search for errors
file_path = [targetdir filesep targetadd filesep mdl];
file_contents = fileread([file_path filesep side '_report.html']) ;
expected_errors=0;
valid_indicator = contains(file_contents, 'complete with 0 errors.')||contains(file_contents,'No messages, warnings,
      or errors were found.');
```

```
testCase.verifyEqual(valid_indicator,true, ...
    sprintf(...
    ("Validationtest of makeHDL for " +side+ "of mdl " + mdl + "with:\nd: [%d]\nk: [%d]\nr: [%d]\nsendState: [%d].")
      ,...
    (AUVParams.d),...
    (AUVParams.k),...
    (AUVParams.r),...
    (sendState)));
disp("*** Verify HDL generation of "+side+" for Variant " +mdl+ " ---- passed");
 %% Test of the validationmodell
  disp("*** Simulate validationmodell for the " +side+ " of " +mdl+ ".")
  act=pwd;
  cd ([AUVParams.uNetSimRootFolder filesep 'tmp'])
  load_system([targetdir '/' targetadd '/' mdl '/gm_' mdl '_vnl']);
  in = Simulink.SimulationInput(['gm_' mdl '_vnl']);
  %% set assertion to error and stop simulation with failure
  blks=find_system(['gm_' mdl '_vnl'], 'MatchFilter', @Simulink.match.allVariants, 'LookUnderMasks', 'all','
      blocktype', 'Assertion');
  for ii=1:length(blks), set_param(blks{ii}, 'stopWhenAssertionFail', 'on'); end
  out = sim(in);

  %% Verify the validationmodell
  testCase.verifyEmpty(out.ErrorMessage, ...
    sprintf(...
    ("Validationtest for " +side+ 'of' + mdl + " with:\nd: [%d]\nk: [%d]\nsendState: [%d].") ,...
    (AUVParams.d),...
    (AUVParams.k),...
    (sendState)));
  [~, idx]=max(xcorr(double(out.receiverOut),double(out.transmitterIn)));
  dataDelay2=idx-size(out.transmitterIn,1);
  tmp=out.transmitterIn;
  disp("*** Verify validation of "+side+" for Variant " +mdl+ " against the Validationmodell ---- passed");
 %% run FIL
  genFIL=evalin('base','genFIL');
  if ismac
  elseif genFIL==true
    hdlworkflowForVariants
    ReadOutResources;
    disp("*** Simulate FIL-Modell for the " +side+ " of " +mdl+ ".")
    act=pwd;
    cd ([AUVParams.uNetSimRootFolder filesep 'tmp'])
    save_system([targetdir '/' targetadd '/' mdl '/' 'gm_' mdl '_fil']);
   %% set assertion to error and stop simulation with failure
    blks=find_system(['gm_' mdl '_fil'], 'MatchFilter', @Simulink.match.allVariants, 'LookUnderMasks', 'all','
      blocktype', 'Assertion');
    for ii=1:length(blks), set_param(blks{ii}, 'stopWhenAssertionFail', 'on'); end
    system(['quartus_pgm -m jtag -o "p;' targetdir filesep targetadd filesep mdl filesep 'fil_prj' filesep side '
      _fil.sof@2"'])
    in = Simulink.SimulationInput(['gm_' mdl '_fil']);
    out = sim(in);
   %% Verify the FIL-Modell
    testCase.verifyEmpty(out.ErrorMessage, ...
        sprintf(...
        ("Validationtest for FIL of " +side+ ' in modell ' + mdl + " with:\nd: [%d]\nk: [%d]\nsendState: [%d].") ,...
        (AUVParams.d),...
        (AUVParams.k),...
        (sendState)));
      disp("*** Verify FIL-simulation of "+side+" for Variant " +mdl+ " ---- passed");
    bdclose([targetdir '/' targetadd '/' mdl '/' 'gm_' mdl '_fil']);
  end
  cd (act)
end
```

Referenzen

Bücher

[104] Teerawat Issariyakul und Ekram Hossain. *Introduction to Network Simulator NS2*. 2nd. Springer Publishing Company, Incorporated, 2011. ISBN: 1461414059.

[115] Paul Adrien Maurice Dirac. *The principles of quantum mechanics*. 27. Oxford university press, 1981.

[116] H. Medwin. *Sounds in the Sea: From Ocean Acoustics to Acoustical Oceanography*. Cambridge University Press, 2018. ISBN: 9781108448147. URL: https://books.google.de/books?id=Hl6LtAEACAAJ.

[122] Walter Munk, Peter Worcester und Carl Wunsch. *Ocean Acoustic Tomography*. Cambridge Monographs on Mechanics. Cambridge University Press, 1995. DOI: 10.1017/CBO9780511666926.

[123] Lawrence E Kinsler u. a. *Fundamentals of acoustics*. John wiley & sons, 2000.

[124] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[134] J.W.S.B. Rayleigh. *The Theory of Sound*. The Theory of Sound Bd. 1. Macmillan, 1894. URL: https://books.google.de/books?id=hd8EAAAAYAAJ.

[136] Petr Beckmann und Andre Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. 1987.

[139] A. D. Waite. *Sonar for Practising Engineers*. 2002.

[140] Edward H Owens. *BEAUFORT WIND SCALEBeaufort wind scale*. Boston, MA: Springer US, 1982, S. 161–162. ISBN: 978-0-387-30843-2. DOI: 10.1007/0-387-30843-1_63. URL: https://doi.org/10.1007/0-387-30843-1%5C_63.

[141] Donald Groves. *Ocean world encyclopedia*. New York: McGraw-Hill, 1980. ISBN: 9780070250109.

[157] Leland B Jackson. *Digital Filters and Signal Processing: With MATLAB® Exercises*. Springer Science & Business Media, 2013.

[159] R.W. Hamming und R.W. Hamming. *Digital Filters*. Prentice-Hall signal processing series. Prentice-Hall, 1977. ISBN: 9780132125710. URL: https://books.google.de/books?id=1fxQAAAAMAAJ.

[161] T.W. Parks und C.S. Burrus. *Digital Filter Design*. Topics in Digital Signal Processing. Wiley, 1987. ISBN: 9780471828969. URL: https://books.google.de/books?id=ux9TAAAAMAAJ.

[162] A.V. Oppenheim und R.W. Schafer. *Discrete-time Signal Processing*. Always learning. Pearson, 2013. ISBN: 9781292025728. URL: https://books.google.de/books?id=LeQpnwEACAAJ.

[164] Julius S. Bendat und Allan G. Piersol. *Random Data: Analysis and Measurement Procedures*. 4th. Hoboken, NJ: John Wiley & Sons, 2011. ISBN: 978-0-470-24877-5.

[176] Andrew S. Tanenbaum und David J. Wetherall. *Computer Networks*. 6th. Boston: Pearson, 2021. ISBN: 978-0136730681.

[177] Alan V. Oppenheim, Ronald W. Schafer und John R. Buck. *Discrete-Time Signal Processing*. 3rd. Prentice Hall, 2010.

[179] S.W. Golomb. *Shift Register Sequences*. Holden-Day series in information systems. Holden-Day, 1967. URL: https://books.google.de/books?id=LqtMAAAAMAAJ.

[185] R.E. Ziemer und R.L. Peterson. *Digital Communications and Spread Spectrum Systems*. Macmillan, 1985. ISBN: 9780024316707. URL: https://books.google.de/books?id=dfpSAAAAMAAJ.

[186] William Wesley Peterson und Edward J Weldon. *Error-correcting codes*. MIT press, 1972.

[187] Tadao Kasami. *Weight distribution formula for some class of cyclic codes*. University of Illinois Champaign, IL, USA, 1966.

[192] Andrzej Zielinski. *Underwater Acoustic Digital Signal Processing and Communication Systems*. Springer, 2012.

[197] Reinhard Diestel. *Graphentheorie. Elektronische Ausgabe*. Springer Verlag, 2000.

[207] Andrew J Viterbi und Jim K Omura. *Principles of digital communication and coding*. Courier Corporation, 2009.

[208] John G. Proakis und Masoud Salehi. *Digital Communications*. McGraw-Hill, 2008.

[213] George C. Clark und J. Bibb Cain. *Error-Correction Coding for Digital Communications*. Applications of Communications Theory. New York: Plenum Press, 1981.

[236] Andreas F. Molisch. *Wireless Communications*. Wiley, 2011.

[237] Slawomir Koziel und Leifur Leifsson. *Surrogate-based Modeling and Optimization: Applications in Engineering*. New York: Springer, 2013. ISBN: 978-1-4614-7952-0.

[238] C. E. Shannon. *A mathematical theory of communication*. Bd. 27. 3. 1948, S. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

Artikel

[3] A. Shaw u. a. „Experimental Investigations of Electromagnetic Wave Propagation in Seawater“. In: (Sep. 2006), S. 572–575. DOI: 10.1109/EUMC.2006.281456.

[4] Umberto M. Cella, Ron Johnstone und Nicholas Shuley. „Electromagnetic Wave Wireless Communication in Shallow Water Coastal Environment: Theoretical Analysis and Experimental Results“. In: WUWNet '09 (2009), 9:1–9:8. DOI: 10.1145/1654130.1654139. URL: http://doi.acm.org/10.1145/1654130.1654139.

[5] A. I. Al-Shamma'a, A. Shaw und S. Saman. „Propagation of electromagnetic waves at MHz frequencies through seawater“. In: *IEEE Transactions on Antennas and Propagation* 52.11 (Nov. 2004), S. 2843–2849. ISSN: 0018-926X. DOI: 10.1109/TAP.2004.834449.

[6] Clayton R. Thurmer u. a. „Underwater Telemetry Using Electromagnetic Communication Through Fresh Water for USACE Infrastructure“. In: *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*. Mai 2018, S. 1–5. DOI: 10.1109/OCEANSKOBE.2018.8559420.

[7] Clayton R. Thurmer u. a. „Toward Underwater Wireless Telemetry for Inland Waterways using Low Frequency Electromagnetic Communication“. In: *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. 2018, S. 110–112. DOI: 10.1109/WiSEE.2018.8637319.

[8] Experimental Investigations, of Electrom und Seawater. „Proceedings of the 36th European Mic“. In: (2006).

[9] M. Stojanovic, J. A. Catipovic und J. G. Proakis. „Phase-coherent digital communications for underwater acoustic channels“. In: *IEEE Journal of Oceanic Engineering* 19.1 (Jan. 1994), S. 100–111. ISSN: 0364-9059. DOI: 10.1109/48.289455.

[10] S. Watson u. a. „Multi-gigabit data transmission using a directly modulated GaN laser diode for visible light communication through plastic optical fiber and water“. In: (Juli 2015), S. 224–225. ISSN: 1099-4742. DOI: 10.1109/PHOSST.2015.7248278.

[11] C. Pontbriand u. a. „Diffuse high-bandwidth optical communications“. In: (Sep. 2008), S. 1–4. ISSN: 0197-7385. DOI: 10.1109/OCEANS.2008.5151977.

[12] H. M. Oubei u. a. „Performance Evaluation of Underwater Wireless Optical Communications Links in the Presence of Different Air Bubble Populations“. In: *IEEE Photonics Journal* 9.2 (Apr. 2017), S. 1–9. ISSN: 1943-0655. DOI: 10.1109/JPHOT.2017.2682198.

[13] Frank Hanson und Stojan Radic. „High bandwidth underwater optical communication“. In: *Applied optics* 47.2 (2008), S. 277–283.

[14] John G Proakis u. a. „Shallow water acoustic networks“. In: *Communications Magazine, IEEE* 39.11 (2001), S. 114–119.

[16] Andrea Caiti u. a. „OptoCOMM: Introducing a new optical underwater wireless communication modem“. In: *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. 2016, S. 1–5. DOI: 10.1109/UComms.2016.7583431.

[17] Roberto Petroccia, Chiara Petrioli und John Potter. „Performance Evaluation of Underwater Medium Access Control Protocols: At-Sea Experiments“. In: *IEEE Journal of Oceanic Engineering* 43.2 (2018), S. 547–556. DOI: 10.1109/JOE.2017.2695759.

[19] Eric Gallimore u. a. „The WHOI Micromodem-2: A Scalable System for Acoustic Communications and Networking“. In: (2010), S. 1–7.

[20] Michael B Porter. „The BELLHOP Manual and User's Guide: PRELIMINARY DRAFT“. In: *Heat, Light, and Sound Research, Inc., La Jolla, CA, USA, Tech. Rep* (2011).

[22] Michael B. Porter und Homer P. Bucker. „Gaussian beam tracing for computing ocean acoustic fields“. In: *The Journal of the Acoustical Society of America* 82.4 (1987), S. 1349–1359. DOI: 10.1121/1.395269. eprint: https://doi.org/10.1121/1.395269. URL: https://doi.org/10.1121/1.395269.

[23] Martin Lukasiewycz u. a. „Opt4J - A Modular Framework for Meta-heuristic Optimization“. In: *Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 2011)*. Dublin, Ireland, Juli 2011, S. 1723–1730.

[26] Xiong Shengjun u. a. „Experimental study of Acoustic Communication over Deep-sea Vertical Channel“. In: *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. 2018, S. 1–5. DOI: 10.1109/ICSPCC.2018.8567738.

[27] Lee Freitag u. a. „Long Range Acoustic Communications and Navigation in the Arctic“. In: (Okt. 2015). DOI: 10.23919/OCEANS.2015.7401956.

[49] P. P. J. Beaujean u. a. „HERMES - A high-speed acoustic modem for real-time transmission of uncompressed image and status transmission in port environment and very shallow water“. In: (2008), S. 1–9. ISSN: 0197-7385. DOI: 10.1109/OCEANS.2008.5151835.

[50] Konstantinos Pelekanakis, Milica Stojanovic und Lee Freitag. „High rate acoustic link for underwater video transmission“. In: 2 (Okt. 2003), 1091–1097 Vol.2. DOI: 10.1109/OCEANS.2003.178494.

[51] A. Sanchez u. a. „A low cost and high efficient acoustic modem for underwater sensor networks“. In: (2011), S. 1–10. DOI: 10.1109/Oceans-Spain.2011.6003428.

[52] Bridget Benson u. a. „Design of a Low-Cost Underwater Acoustic Modem“. In: *Embedded Systems Letters* 2 (Sep. 2010), S. 58–61. DOI: 10.1109/LES.2010.2050191.

[53] Nusrat Nowsheen, Craig Benson und Michael Frater. „A high data-rate, software-defined underwater acoustic modem“. In: (2010), S. 1–5. DOI: 10.1109/OCEANS.2010.5664474.

[54] Mohammad Sadeghi, Mohammed Elamassie und Murat Uysal. „Adaptive OFDM-based acoustic underwater transmission: System design and experimental verification“. In: (2017), S. 1–5. DOI: 10.1109/BlackSeaCom.2017.8277660.

[55] Adil A. Sheikh, Emad Felemban und Adnan Ashraf. „Coralcon: An open source low-cost modem for underwater IoT applications“. In: (2017), S. 503–508. DOI: 10.1109/ICCP.2017.8117054.

[56] Bo Peng und Hefeng Dong. „DSP based real-time single carrier underwater acoustic communications using frequency domain turbo equalization“. In: *Physical Communication* 18 (2016), S. 40–48. ISSN: 1874-4907. DOI: https://doi.org/10.1016/j.phycom.2015.11.001. URL: https://www.sciencedirect.com/science/article/pii/S1874490715000634.

[58] Jinfeng Li, Joseph Halt und Y. Rosa Zheng. „Utilizing JANUS for Very High Frequency Underwater Acoustic Modem“. In: *Global Oceans 2020: Singapore – U.S. Gulf Coast*. 2020, S. 1–6. DOI: 10.1109/IEEECONF38699.2020.9389046.

[59] Slamet Indriyanto und Ian Yosef Matheus Edward. „Ultrasonic Underwater Acoustic Modem Using Frequency Shift Keying (FSK) Modulation“. In: *2018 4th International Conference on Wireless and Telematics (ICWT)*. 2018, S. 1–4. DOI: 10.1109/ICWT.2018.8527809.

[60] Emrecan Demirors u. a. „A High-Rate Software-Defined Underwater Acoustic Modem With Real-Time Adaptation Capabilities“. In: *IEEE Access* 6 (2018), S. 18602–18615. DOI: 10.1109/ACCESS.2018.2815026.

[61] Qi Dong, Yiyin Wang und Xinping Guan. „The Design and Implementation of an Underwater Multimode Acoustic Modem for Autonomous Underwater Vehicles“. In: *2018 37th Chinese Control Conference (CCC)*. 2018, S. 4201–4205. DOI: 10.23919/ChiCC.2018.8482662.

[62] Muhammad Yousuf Zia, Pablo Otero und Javier Poncela. „Design of a Low-Cost Modem for Short-Range Underwater Acoustic Communications“. In: *Wirel. Pers. Commun.* 101.1 (Juli 2018), S. 375–390. ISSN: 0929-6212. DOI: 10.1007/s11277-018-5694-5. URL: https://doi.org/10.1007/s11277-018-5694-5.

[63] Adil A. Sheikh, Emad Felemban und Adnan Ashraf. „Coralcon: An open source low-cost modem for underwater IoT applications“. In: *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. 2017, S. 503–508. DOI: 10.1109/ICCP.2017.8117054.

[64] Hongtao Zhang u. a. „Sea trials of an underwater acoustic network in the East China Sea 2015“. In: *2016 IEEE/OES China Ocean Acoustics (COA)*. 2016, S. 1–5. DOI: 10.1109/COA.2016.7535737.

[65] Ruiqin Zhao u. a. „Underwater acoustic network node design and anechoic pool network experimentation with five nodes“. In: *2016 IEEE/OES China Ocean Acoustics (COA)*. 2016, S. 1–5. DOI: 10.1109/COA.2016.7535802.

[66] Cheepurupalli Ch Naidu und E S Stalin. „Establishment of underwater wireless acoustic MODEM using C-OFDM“. In: *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*. 2016, S. 1–6. DOI: 10.1109/MicroCom.2016.7522465.

[67] Yuehai Zhou und Feng Tong. „Research and Development of a Highly Reconfigurable OFDM MODEM for Shallow Water Acoustic Communication“. In: *IEEE Access* 7 (2019), S. 123569–123582. DOI: 10.1109/ACCESS.2019.2936933.

[68] Yibin Wang, Andrew Dobbin und Jean-François Bousquet. „A Compact Low-Power Underwater Magneto-Inductive Modem“. In: WUWNet '16 (2016). DOI: 10.1145/2999504.3001064. URL: https://doi.org/10.1145/2999504.3001064.

[76] Pedro Góis u. a. „Development and validation of blue ray, an optical modem for the MEDUSA class AUVs“. In: (2016), S. 1–5. DOI: 10.1109/UComms.2016.7583455.

[80] Lee Freitag u. a. „The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms“. In: (2005), S. 1086–1092.

[81] L. Freitag u. a. „The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms“. In: (Sep. 2005), 1086–1092 Vol. 2. DOI: 10.1109/OCEANS.2005.1639901.

[83] D.L. Schilling. „Broadband spread spectrum multiple access for personal and cellular communications“. In: *IEEE 43rd Vehicular Technology Conference*. 1993, S. 819–822. DOI: 10.1109/VETEC.1993.508816.

[84] Konstantin G. Kebkal und Rudolf Bannasch. „Sweep-spread carrier for underwater communication over acoustic channels with strong multipath propagation“. In: *The Journal of the Acoustical Society of America* 112.5 (2002), S. 2043–2052. DOI: 10.1121/1.1504855. eprint: https://doi.org/10.1121/1.1504855. URL: https://doi.org/10.1121/1.1504855.

[85] K.P. Arunkumar und Chandra R. Murthy. „Soft Symbol Decoding in Sweep-Spread-Carrier Underwater Acoustic Communications: A Novel Variational Bayesian Algorithm and Its Analysis“. In: *IEEE Transactions on Signal Processing* 68 (2020), S. 2435–2448. DOI: 10.1109/TSP.2020.2983830.

[86] O. P. Gandhi und V. S. Charan. „Modelling and Simulation of Communication Systems: Techniques and Strategies“. In: *International Journal of Computer Applications* 175.2 (2020), S. 1–6. DOI: 10.5120/ijca2020919101.

[87] Y. Lin und S. Lee. „A Survey of Model-Based Testing Techniques for 5G Communication Systems“. In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), S. 1810–1832. DOI: 10.1109/COMST.2021.3069034.

[88] Y. Yang und J. Chen. „Model-Driven Engineering for IoT Systems: Challenges, Solutions, and Future Directions“. In: *IEEE Internet of Things Journal* (2023). DOI: 10.1109/JIOT.2023.3144042.

[89] M. Stojanovic und J. Preisig. „Underwater acoustic communication channels: Propagation models and statistical characterization“. In: *IEEE Communications Magazine* 47.1 (2009), S. 84–89. DOI: 10.1109/MCOM.2009.4752682.

[90] Aijun Song, Milica Stojanovic und Mandar Chitre. „Editorial underwater acoustic communications: Where we stand and what is next?“ In: *IEEE Journal of Oceanic Engineering* 44.1 (2019).

[92] Douglas B. Johnson u. a. „JANUS: The genesis, propagation, and use of an underwater standard“. In: *IEEE Journal of Oceanic Engineering* 40.4 (2014), S. 853–864. DOI: 10.1109/JOE.2014.2330193.

[93] B. Mounika und B. K Priya. „Analysis and Comparison of Different Channel Coding Techniques for Underwater Channel using AWGN and Acoustic Channel“. In: *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*. 2018, S. 1664–1669. DOI: 10.1109/ICEECCOT43722.2018.9001323.

[94] Xiaohui Yao, Honghui Yang und Yiqing Li. „Modulation Identification of Underwater Acoustic Communications Signals Based on Generative Adversarial Networks“. In: Juni 2019, S. 1–6. DOI: 10.1109/OCEANSE.2019.8867125.

[95] X. Ning, Z. Liu und X. Fu. „LMMSE channel estimation method for underwater acoustic OFDM systems under narrow-band jamming“. In: 36 (Okt. 2012), S. 1054–1058. DOI: 10.3963/j.issn.2095-3844.2012.05.042.

[97] JS Kim u. a. „Virtual time series experiment (virtex) simulation tool for underwater acoustic communications.“ In: *The Journal of the Acoustical Society of America* 126.4 (2009), S. 2174–2174.

[98] Federico Guerra, Paolo Casari und Michele Zorzi. „World Ocean Simulation System (WOSS): A Simulation Tool for Underwater Networks with Realistic Propagation Modeling“. In: *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*. WUWNet '09. Berkeley, California: Association for Computing Machinery, 2009. ISBN: 9781605588216. DOI: 10.1145/1654130.1654134. URL: https://doi.org/10.1145/1654130.1654134.

[99] Paolo Casari u. a. „Open source suites for underwater networking: WOSS and DESERT underwater“. In: *IEEE Network* 28.5 (2014), S. 38–46. DOI: 10.1109/MNET.2014.6915438.

[103] Peng Xie u. a. „Aqua-Sim: An NS-2 based simulator for underwater sensor networks“. In: *OCEANS 2009*. 2009, S. 1–7. DOI: 10.23919/OCEANS.2009.5422081.

[106] Filippo Campagnaro u. a. „The DESERT underwater framework v2: Improved capabilities and extension tools“. In: *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. 2016, S. 1–5. DOI: 10.1109/UComms.2016.7583420.

[107] Nguyen B. Truong, Young-Joo Suh und Chansu Yu. „Latency Analysis in GNU Radio/USRP-Based Software Radio Platforms“. In: *MILCOM 2013 - 2013 IEEE Military Communications Conference*. 2013, S. 305–310. DOI: 10.1109/MILCOM.2013.60.

[108] Filippo Campagnaro u. a. „Full reconfiguration of underwater acoustic networks through low-level physical layer access“. In: *Proceedings of the International Conference on Underwater Networks & Systems*. 2017, S. 1–8.

[109] Beatrice Tomasi u. a. „Performance study of variable-rate modulation for underwater communications based on experimental data“. In: *OCEANS 2010 MTS/IEEE SEATTLE*. IEEE. 2010, S. 1–8.

[110] Y-P Cong u. a. „Framework of adaptive multi-mode underwater wireless communication networks research“. In: *Periodical of Ocean University of China* 42.5 (2012), S. 115–119.

[111] Lei Wan u. a. „Adaptive modulation and coding for underwater acoustic OFDM“. In: *IEEE Journal of Oceanic Engineering* 40.2 (2014), S. 327–336.

[112] Konstantinos Pelekanakis u. a. „Decision tree-based adaptive modulation for underwater acoustic communications“. In: *2016 IEEE third underwater communications and networking conference (UComms)*. IEEE. 2016, S. 1–5.

[113] Sanjay Mani, Tolga M Duman und Paul Hursky. „Adaptive coding-modulation for shallow-water UWA communications“. In: *Journal of the Acoustical Society of America* 123.5 (2008), S. 3749.

[114] Weiqing Zhu u. a. „Signal processing in underwater acoustic communication system for manned deep submersible" Jiaolong"“. In: *Shengxue Xuebao(Acta Acustica)* 37.6 (2012), S. 505–573.

[117] Kenneth V. Mackenzie. „Nine?term equation for sound speed in the oceans“. In: *The Journal of the Acoustical Society of America* 70.3 (1981), S. 807–812. DOI: 10.1121/1.386920. eprint: https://doi.org/10.1121/1.386920. URL: https://doi.org/10.1121/1.386920.

[118] V. A. Del Grosso. „New equation for the speed of sound in natural waters (with comparisons to other equations)“. In: *The Journal of the Acoustical Society of America* 56.4 (1974), S. 1084–1091. DOI: 10.1121/1.1903388. eprint: https://doi.org/10.1121/1.1903388. URL: https://doi.org/10.1121/1.1903388.

[119] Chen-Tung Chen und Frank J. Millero. „Speed of sound in seawater at high pressures“. In: *The Journal of the Acoustical Society of America* 62.5 (1977), S. 1129–1135. DOI: 10.1121/1.381646. eprint: https://doi.org/10.1121/1.381646. URL: https://doi.org/10.1121/1.381646.

[120] Wayne D. Wilson. „Equation for the Speed of Sound in Sea Water“. In: *The Journal of the Acoustical Society of America* 32.10 (1960), S. 1357–1357. DOI: 10.1121/1.1907913. eprint: https://doi.org/10.1121/1.1907913. URL: https://doi.org/10.1121/1.1907913.

[121] Brian D. Dushaw u. a. „On equations for the speed of sound in seawater“. In: *The Journal of the Acoustical Society of America* 93.1 (Jan. 1993), S. 255–275. ISSN: 0001-4966. DOI: 10.1121/1.405660. eprint: https://pubs.aip.org/asa/jasa/article-pdf/93/1/255/11723705/255_1_online.pdf. URL: https://doi.org/10.1121/1.405660.

[125] Milica Stojanovic. „On the relationship between capacity and distance in an underwater acoustic communication channel“. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 11.4 (Okt. 2007), S. 34–43. ISSN: 1559-1662. DOI: 10.1145/1347364.1347373. URL: https://doi.org/10.1145/1347364.1347373.

[127] Youwen Zhang u. a. „Deep Learning Based Single Carrier Communications Over Time-Varying Underwater Acoustic Channel“. In: *IEEE Access* 7 (2019), S. 38420–38430. DOI: 10.1109/ACCESS.2019.2906424.

[128] K.P. Arunkumar und Chandra R. Murthy. „Soft Symbol Decoding in Sweep-Spread-Carrier Underwater Acoustic Communications: A Novel Variational Bayesian Algorithm and Its Analysis“. In: *IEEE Transactions on Signal Processing* 68 (2020), S. 2435–2448. DOI: 10.1109/TSP.2020.2983830.

[129] William H. Thorp. „Analytic Description of the Low-Frequency Attenuation Coefficient“. In: *Journal of The Acoustical Society of America - J ACOUST SOC AMER* 42 (Juli 1967). DOI: 10.1121/1.1910566.

[130] F. H. Fisher und V. P. Simmons. „Sound absorption in sea water“. In: *The Journal of the Acoustical Society of America* 62.3 (1977), S. 558–564. DOI: 10.1121/1.381574. eprint: https://doi.org/10.1121/1.381574. URL: https://doi.org/10.1121/1.381574.

[131] R. E. Francois und G. R. Garrison. „Sound absorption based on ocean measurements: Part I: Pure water and magnesium sulfate contributions“. In: *The Journal of the Acoustical Society of America* 72.3 (1982), S. 896–907. DOI: 10.1121/1.388170. eprint: https://doi.org/10.1121/1.388170. URL: https://doi.org/10.1121/1.388170.

[132] R. E. Francois und G. R. Garrison. „Sound absorption based on ocean measurements. Part II: Boric acid contribution and equation for total absorption“. In: *The Journal of the Acoustical Society of America* 72.6 (1982), S. 1879–1890. DOI: 10.1121/1.388673. eprint: https://doi.org/10.1121/1.388673. URL: https://doi.org/10.1121/1.388673.

[133] C. A. M. van Moll, M. A. Ainslie und R. van Vossen. „A Simple and Accurate Formula for the Absorption of Sound in Seawater“. In: *IEEE Journal of Oceanic Engineering* 34.4 (Okt. 2009), S. 610–616. ISSN: 0364-9059. DOI: 10.1109/JOE.2009.2027800.

[135] AA Maradudin und DL Mills. „The attenuation of Rayleigh surface waves by surface roughness“. In: *Annals of Physics* 100.1-2 (1976), S. 262–309.

[138] R. Coates. „An empirical formula for computing the Beckmann-Spizzichino surface reflection loss coefficient“. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 35.4 (1988), S. 522–523. DOI: 10.1109/58.4190.

[142] Reinhard Burger und Michael Lynch. „Evolution and Extinction in a Changing Environment: A Quantitative-Genetic Analysis“. In: *Evolution* 49.1 (1995), S. 151–163. ISSN: 00143820, 15585646. URL: http://www.jstor.org/stable/2410301.

[143] M. W. Legg u. a. „Analysis of impulsive biological noise due to snapping shrimp as a point process in time“. In: (Juni 2007), S. 1–6. DOI: 10.1109/OCEANSE.2007.4302279.

[145] S.W. Marshall. „Depth Dependence of Ambient Noise“. In: *Oceanic Engineering, IEEE Journal of* 30 (Mai 2005), S. 275–281. DOI: 10.1109/JOE.2005.850876.

[146] Morris. „Depth dependence of ambient noise in the northeastern Pacific Ocean“. In: *The Journal of the Acoustical Society of America* 64.2 (Aug. 1978), S. 581–590. ISSN: 0001-4966. DOI: 10.1121/1.382010. eprint: https://pubs.aip.org/asa/jasa/article-pdf/64/2/581/11724506/581_1_online.pdf. URL: https://doi.org/10.1121/1.382010.

[147] G Jiménez-Arranz u. a. „Review of Existing Data on Underwater Sounds Produced by the Oil and Gas Industry“. In: *JIP topic-sound source characterisation and propagation* (2020).

[151] Marcel Rieß und Frank Slomka. „Geometric Multi-path Avoiding Approach for Adaptive Acoustic Underwater Communications“. In: *Proceedings of the International Conference on Underwater Networks & Systems*. 2019, S. 1–5.

[152] Marcel Rieß und Frank Slomka. „Analysis-space-reduction for automated modem optimization in underwater communication“. In: *Global Oceans 2020*. 2020.

[153] D.V. Sarwate und M.B. Pursley. „Crosscorrelation properties of pseudorandom and related sequences“. In: *Proceedings of the IEEE* 68.5 (1980), S. 593–619. DOI: 10.1109/PROC.1980.11697.

[154] Gordon M. Wenz. „Acoustic Ambient Noise in the Ocean: Spectra and Sources“. In: *The Journal of the Acoustical Society of America* 34.12 (1962), S. 1936–1956. DOI: 10.1121/1.1909155. eprint: https://doi.org/10.1121/1.1909155. URL: https://doi.org/10.1121/1.1909155.

[155] Marcel Rieß und Frank Slomka. „Ambient Noise in Acoustic Underwater Communication Modeling and Analysis“. In: *Global Oceans 2020*. 2020.

[156] Marcel Rieß und Frank Slomka. „FIR-Filter: based Ambient Noise Implementation for Acoustic Underwater Communication Analysis“. In: *Proceedings of the International Conference on Underwater Networks & Systems*. 2019, S. 1–2.

[158] Peter D. Welch. „The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms“. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), S. 70–73.

[160] J. Kaiser und R. Schafer. „On the use of the I0-sinh window for spectrum analysis“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.1 (1980), S. 105–107. DOI: 10.1109/TASSP.1980.1163349.

[163] Solomon Kullback und Richard A. Leibler. „On information and sufficiency“. In: *The Annals of Mathematical Statistics* 22.1 (1951), S. 79–86.

[165] Marcel Rieß und Frank Slomka. „FIR-Filter: Based Ambient Noise Implementation for Acoustic Underwater Communication Analysis“. In: *Proceedings of the International Conference on Underwater Networks & Systems*. WUWNET'19. Atlanta, GA, USA: Association for Computing Machinery, 2019. ISBN: 9781450377409. DOI: 10.1145/3366486.3366536. URL: https://doi.org/10.1145/3366486.3366536.

[166] Marcel Rieß und Frank Slomka. „Ambient noise in acoustic underwater communication modeling and analysis“. In: *Global Oceans 2020: Singapore – U.S. Gulf Coast*. 2020, S. 1–6. DOI: 10.1109/IEEECONF38699.2020.9389139.

[167] D. H. Goldberg u. a. „A wake-up detector for an acoustic surveillance sensor network: algorithm and VLSI implementation“. In: *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*. 2004, S. 134–141. DOI: 10.1109/IPSN.2004.238897.

[168] M. T. R. Khan, S. H. Ahmed und D. Kim. „AUV-Assisted Energy-Efficient Clustering in Underwater Wireless Sensor Networks“. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, S. 1–7. DOI: 10.1109/GLOCOM.2018.8647278.

[169] S. Jevtic u. a. „Lucid Dreaming: Reliable Analog Event Detection for Energy-Constrained Applications“. In: *2007 6th International Symposium on Information Processing in Sensor Networks*. 2007, S. 350–359. DOI: 10.1109/IPSN.2007.4379695.

[170] D. Zhang, W. Jin und D. Sun. „Design of duty circuit on underwater transponder“. In: *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. 2017, S. 1–5. DOI: 10.1109/ICSPCC.2017.8242410.

[171] M. Yue u. a. „Microcontroller implementation of low-complexity wake-up receiver for wireless sensor nodes in severe multipath fading channels“. In: *2016 IEEE/OES China Ocean Acoustics (COA)*. 2016, S. 1–6. DOI: 10.1109/COA.2016.7535695.

[172] D. Wang u. a. „Channel-Adaptive Location-Assisted Wake-up Signal Detection Approach Based on LFM Over Underwater Acoustic Channels“. In: *IEEE Access* 7 (2019), S. 93806–93819. DOI: 10.1109/ACCESS.2019.2926531.

[173] P. M. Baggenstoss. „On detecting linear frequency-modulated waveforms in frequency- and time-dispersive channels: alternatives to segmented replica correlation“. In: *IEEE Journal of Oceanic Engineering* 19.4 (1994), S. 591–598. DOI: 10.1109/48.338395.

[174] Marcel Riess und Frank Slomka. „Improved Acoustic Underwater Communication with Orthogonal Kasami-Based Sequences for Multi-User Communication“. In: *Proceedings of the 15th International Conference on Underwater Networks and Systems*. WUWNet '21. Shenzhen, Guangdong, China: Association for Computing Machinery, 2022. ISBN: 9781450395625. DOI: 10.1145/3491315.3491319. URL: https://doi.org/10.1145/3491315.3491319.

[178] RH Barker. „Group synchronizing of binary digital systems“. In: *Communication theory* (1953), S. 273–287.

[180] James L Massey. „Shift-Register Synthesis and BCH Decoding“. In: *Information Theory, IEEE Transactions on* 15.1 (1969), S. 122–127.

[181] Bernd E. Helms. „Maximal Length Sequences“. In: *IEEE Transactions on Information Theory* (1985).

[182] R. Gold. „Optimal binary sequences for spread spectrum multiplexing (Corresp.)“ In: *IEEE Transactions on Information Theory* 13.4 (Okt. 1967), S. 619–621. ISSN: 1557-9654. DOI: 10.1109/TIT.1967.1054048.

[183] Robert Gold. „Maximal recursive sequences with 3-valued recursive cross-correlation functions (Corresp.)“ In: *IEEE transactions on Information Theory* 14.1 (1968), S. 154–156.

[184] Esmael H Dinan und Bijan Jabbari. „Spreading codes for direct sequence CDMA and wideband CDMA cellular networks“. In: *IEEE communications magazine* 36.9 (1998), S. 48–54.

[188] Abhijit Chandra und Sudipta Chattopadhyay. „Small Set Orthogonal Kasami codes for CDMA system“. In: *2009 4th International Conference on Computers and Devices for Communication (CODEC)*. 2009, S. 1–4.

[189] H. Donelan und T. O'Farrell. „Method for generating sets of orthogonal sequences“. In: *Electronics Letters* 35.18 (1999), S. 1537–1538. DOI: 10.1049/el:19991046.

[190] Eric Gallimore u. a. „The WHOI micromodem-2: A scalable system for acoustic communications and networking“. In: (2010), S. 1–7. DOI: 10.1109/OCEANS.2010.5664354.

[191] Aaron B Kilfoyle und Arthur B Baggeroer. „The state of the art in underwater acoustic telemetry“. In: *IEEE Journal of Oceanic Engineering* 25.1 (2000), S. 4–27.

[193] M. Stojanovic und J. Preisig. „Underwater acoustic communication channels: Propagation models and statistical characterization“. In: *IEEE Communications Magazine* 47.1 (Jan. 2009), S. 84–89. ISSN: 0163-6804. DOI: 10.1109/MCOM.2009.4752682.

[194] Roee Diamant, Lutz Lampe und Winston KG Seah. „Adaptive modulation and coding for underwater acoustic communications“. In: *IEEE Journal of Oceanic Engineering* 38.3 (2013), S. 486–499.

[195] Sami M F Bushnaq, Michael D Richardson und Mohamad A Ahmed. „Doppler shift compensation in underwater acoustic communication systems“. In: *IEEE Access* 2 (2014), S. 1414–1422.

[196] Paolo Casari und Michele Zorzi. „Protocol design issues in underwater acoustic networks“. In: *Computer Communications* 34.17 (2011), S. 2013–2025.

[198] G.D. Forney. „The viterbi algorithm“. In: *Proceedings of the IEEE* 61.3 (1973), S. 268–278. DOI: 10.1109/PROC.1973.9030.

[199] IS Reed und G Solomon. „Polynomial Codes Over Certain Finite Fields“. In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), S. 300–304.

[200] Jim Taylor, Kar-Ming Cheung, Dongae Seo u. a. „Galileo telecommunications“. In: *DESCANSO design and performance summary series* (2002).

[201] IEEE. „IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band“. In: *IEEE Std 802.11a-1999* (1999), S. 1–102. DOI: 10.1109/IEEESTD.1999.90606.

[202] David Dye. „Partial reconfiguration of Xilinx FPGAs using ISE design suite“. In: *White Paper* (2012).

[203] Kizheppatt Vipin und Suhaib A Fahmy. „FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications“. In: *ACM Computing Surveys (CSUR)* 51.4 (2018), S. 1–39.

[209] Amos Lapidoth. „The performance of convolutional codes on the block erasure channel using various finite interleaving techniques“. In: *IEEE Transactions on Information Theory* 40.5 (1994), S. 1459–1473.

[210] O. Y. Takeshita und Jr. D. J. Costello. „New Classes Of Algebraic Interleavers for Turbo-Codes“. In: *Proc. 1998 IEEE International Symposium on Information Theory*. Boston, Aug. 1998, S. 419.

[211] J. L. Ramsey. „Realization of Optimum Interleavers“. In: *IEEE Transactions on Information Theory* IT-16.3 (Mai 1970), S. 338–345.

[212] G. D. Forney Jr. „Burst-Correcting Codes for the Classic Bursty Channel“. In: *IEEE Transactions on Communications* COM-19.5 (Okt. 1971), S. 772–781.

[215] Yu Li, Haining Huang und Zhijie Wang. „FPGA design and implementation of pulse shaping filter for coherent underwater communication“. In: *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. 2012, S. 748–752. DOI: 10.1109/ICCSNT.2012.6526041.

[216] Yanyan Wang u. a. „Acoustic-domain self-interference cancellation for full-duplex underwater acoustic communication systems“. In: *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2019, S. 1112–1116.

[217] Rao Farhat Masood, Ali Anjum und Raza Ali. „Adaptive Modulation (QPSK, QAM)“. In: *arXiv preprint arXiv:1302.7145* (2013). DOI: 10.48550/arXiv.1302.7145. URL: https://arxiv.org/abs/1302.7145.

[218] Wen-Bin Yang und T. C. Yang. „High-frequency FH-FSK underwater acoustic communications: The environment effect and signal processing“. In: *Journal of the Acoustical Society of America* 114.4 (2004), S. 2407–2412. DOI: 10.1063/1.1843003.

[219] Ji Gang Dong u. a. „FH-FSK underwater acoustic control and communication system based on LDPC code“. In: *Applied Mechanics and Materials*. Bd. 511-512. Trans Tech Publications Ltd, 2014, S. 375–380. DOI: 10.4028/www.scientific.net/AMM.511-512.375.

[220] Valentina Richthammer u. a. „Design Space Exploration for Model-based Communication Systems“. In: *Design, Automation and Test in Europe Conference 2020* ().

[221] Marcel Rieß u. a. „Analysis-Space-reduction for Automated Modem Optimization in Underwater Communication“. In: Global OCEANS 2020. 2020.

[222] Marcel Rieß, Steffen Moser und Frank Slomka. „Online Reduction of Exploration Space for Automated Underwater Modem Optimization“. In: *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*. 2023, S. 430–437. DOI: 10.1109/MILCOM58377.2023.10356305.

[223] K. Deb u. a. „A fast and elitist multiobjective genetic algorithm: NSGA-II“. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), S. 182–197. DOI: 10.1109/4235.996017.

[224] Eckart Zitzler, Marco Laumanns und Lothar Thiele. „SPEA2: Improving the Strength Pareto Evolutionary Algorithm“. In: *Proceedings of the Eurogen 2001 Conference*. 2002, S. 95–100.

[225] Valentina Richthammer u. a. „Design Space Exploration for Model-based Communication Systems“. In: *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2020, S. 556–561. DOI: 10.23919/DATE48585.2020.9116508.

[226] Karl Bringmann und Tobias Friedrich. „Approximation of the Hypervolume Indicator“. In: *Artificial Intelligence* 182-183 (2010), S. 71–81. DOI: 10.1016/j.artint.2010.04.005.

[227] Carlos M. Fonseca, Luís Paquete und Manuel Lopez-Ibanez. „Improved Performance Assessment for Multiobjective Optimization“. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2006, S. 530–537. DOI: 10.1145/1143997.1144091.

[228] Joshua Knowles und David W. Corne. „On Metrics for Comparing Non-Dominated Sets“. In: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*. Bd. 1. IEEE, 2002, S. 711–716.

[229] B. Tomasi u. a. „Experimental study of the space-time properties of acoustic channels for underwater communications“. In: (Mai 2010), S. 1–9. DOI: 10.1109/OCEANSSYD.2010.5603667.

[230] Michael J. Hamilton, Stephanie Kemna und David Hughes. „Antisubmarine warfare applications for autonomous underwater vehicles: The GLINT09 sea trial results“. In: *Journal of Field Robotics* 27.6 (2010), S. 890–902. DOI: https://doi.org/10.1002/rob.20362. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20362. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20362.

[231] Michael Goetz u. a. „Performance analysis of underwater network protocols within international sea trial“. In: *OCEANS 2015 - Genova*. 2015, S. 1–7. DOI: 10.1109/OCEANS-Genova.2015.7271400.

[232] Petter Norgren u. a. „Tracking and remote monitoring of an autonomous underwater vehicle using an unmanned surface vehicle in the Trondheim fjord“. In: *OCEANS 2015 - MTS/IEEE Washington*. 2015, S. 1–6. DOI: 10.23919/OCEANS.2015.7401975.

[233] Martin Ludvigsen u. a. „Scientific operations combining ROV and AUV in the Trondheim Fjord“. In: *2013 MTS/IEEE OCEANS - Bergen*. 2013, S. 1–7. DOI: 10.1109/OCEANS-Bergen.2013.6608194.

[234] Konstantin G. Kebkal, Oleksiy G. Kebkal und Roberto Petroccia. „Assessment of underwater acoustic channel during payload data exchange with hydro-acoustic modems of the S2C series“. In: *OCEANS 2015 - Genova*. 2015, S. 1–4. DOI: 10.1109/OCEANS-Genova.2015.7271671.

[235] Marcel W Rieß und Frank Slomka. „Geometric Multi-path Avoiding Approach for Adaptive Acoustic Underwater Communications“. In: (Okt. 2019).

[239] John W. Tukey. „Exploratory Data Analysis“. In: *Reading, MA: Addison-Wesley* (1977).

[240] E. Langford. „Quartiles in Elementary Statistics“. In: *Journal of Statistics Education* 14.3 (2006).

[241] R. McGill, J.W. Tukey und W.A. Larsen. „Variations of box plots“. In: *The American Statistician* 32.1 (1978), S. 12–16.

Spezifikationen

[1] Aricò Salvatore. *Global Ocean Science Report 2020*. URL: https://unesdoc.unesco.org/ark:/48223/pf0000375147.

[2] *Underwater vehicle gone missing under a glacier*. Jan. 2024. URL: https://www.gu.se/en/news/underwater-vehicle-gone-missing-under-a-glacier.

[15] Teledyne Marine. *Teledyne Benthos Modems PSG*. URL: http://www.teledynemarine.com/Lists/Downloads/Benthos_Modem_PSG.pdf.

[18] Evologics S2C. *evologics*. URL: https://evologics.de/web/content/15634.

[21] Michael B Porter und Yong-Chun Liu. *Finite-element ray tracing*. 1994.

[24] Lukasiewycz, Martin and Glaß, Michael and Reimann, Felix and Teich, Jürgen. *Opt4J-Framework*. URL: https://sdarg.github.io/opt4j.

[25] Lukasiewycz, Martin and Reimann, Felix and Fedor, Smirnov and Falko, Hoefte. *OpenDSE-Framework*. URL: https://sdarg.github.io/opendse/.

[28] Blueprint Subsea. *Blueprint Subsea*. URL: https://www.blueprintsubsea.com.

[29] LinkQuest. *LinkQuest*. URL: http://www.link-quest.com.

[30] LinkQuest. *LinkQuest UWM1000*. URL: http://www.link-quest.com/html/uwm1000.htm.

[31] LinkQuest. *LinkQuest UWM2000*. URL: http://www.link-quest.com/html/uwm2000.htm.

[32] LinkQuest. *LinkQuest UWM2200*. URL: http://www.link-quest.com/html/uwm2200.htm.

[33] LinkQuest. *LinkQuest UWM2000H*. URL: http://www.link-quest.com/html/uwm2000H.htm.

[34] LinkQuest. *LinkQuest UWM3000*. URL: http://www.link-quest.com/html/uwm3000.htm.

[35] LinkQuest. *LinkQuest UWM3000H*. URL: http://www.link-quest.com/html/uwm3000H.htm.

[36] LinkQuest. *LinkQuest UWM4000*. URL: http://www.link-quest.com/html/uwm4000.htm.

[37] LinkQuest. *LinkQuest UWM10000*. URL: http://www.link-quest.com/html/uwm10000.htm.

[38] Sonardyne. *Sonardyne Modem 6 Sub-Mini*. URL: https://www.sonardyne.com/app/uploads/2016/06/Sonardyne_8377_Modem_6_Sub-Mini.pdf.

[39] Sonardyne. *Sonardyne Modem 6 Mini*. URL: https://www.sonardyne.com/app/uploads/2016/06/Sonardyne_8244_Modem_6_Mini_Subsea.pdf.

[40] Sonardyne. *Sonardyne Modem 6 Standard*. URL: https://www.sonardyne.com/app/uploads/2016/06/Sonardyne_8307_Modem_6_Standard.pdf.

[41] Develogic. *Develogic HAM.BASE/NODE*. URL: http://www.develogic.de/wp-content/uploads/2014/04/HAM.Node_HAM.Base-04-2014.pdf.

[42] Sercel. *Sercel MATS 3G*. URL: http://www.sercel.com/products/Lists/ProductSpecification/Mats3G_specifications_Sercel_EN.pdf.

[43] DiveNET. *Sealink Modems*. URL: https://www.divenetgps.com/sealink.

[44] Subnero. *Subnero*. URL: https://subnero.com.

[45] Subnero. *Subnero M25M*. URL: https://subnero.com/brochures/Subnero-Modem-Specifications.pdf.

[46] AppliCon. *SEAMODEM*. URL: http://www.applicon.it.

[47] L3Harris. *GPM 300 ACOUSTIC MODEM*. URL: https://www.l3harris.com/sites/default/files/2020-09/ims-maritime-datasheet-GPM%20300.pdf.

[48] Moog. *Micron MODEM*. URL: https://www.tritech.co.uk/media/products/micron-data-modem.pdf.

[57] Kongsberg. *CNODE Modem Modem MiniS 34-180*. URL: https://www.kongsberg.com/globalassets/maritime/km-products/documents/425498c_cnode_modem_minis_datasheet.pdf.

[69] Aquatec. *AQUAmodem Op2*. URL: https://www.aquatecgroup.com/images/products/AQUAmodemOp2Datasheet.pdf.

[70] Aquatec. *AQUAmodem Op2L*. URL: https://www.aquatecgroup.com/images/products/AQUAmodemOp2lDatasheet.pdf.

[71] Hydromea. *Hydromea LUMA*. URL: https://www.hydromea.com.

[72] Hydromea. *Hydromea LUMA X*. URL: https://www.hydromea.com/wp-content/uploads/2021/06/Hydromea_LUMA_X_datasheet.pdf.com.

[73] Hydromea. *Hydromea LUMA 500ER*. URL: https://www.hydromea.com/wp-content/uploads/2020/08/Hydromea_LUMA_500ER_datasheet.pdf.

[74] Hydromea. *Hydromea LUMA 250LP*. URL: https://www.hydromea.com/wp-content/uploads/2020/08/Hydromea_LUMA_250LP_datasheet.pdf.

[75] Hydromea. *Hydromea LUMA 100*. URL: https://www.hydromea.com/wp-content/uploads/2020/08/Hydromea_LUMA_100_datasheet.pdf.

[77] Sonardyne. *Sonardyne*. URL: https://www.sonardyne.com.

[78] Sonardyne. *BlueComm 100*. URL: https://www.sonardyne.com/wp-content/uploads/2021/06/Sonardyne_8360_BlueComm_100.pdf.

[79] Sonardyne. *BlueComm 200*. URL: https://www.sonardyne.com/wp-content/uploads/2021/07/Sonardyne_8361_BlueComm_200_UV.pdf.

[82] BaltRobotics. *UNDERWATER WIRELESS ACOUSTIC VIDEO COMMUNICATIONS CHANNEL*. URL: http://www.baltrobotics.com.

[91] NATO Centre for Maritime Research and Experimentation. *JANUS: A Digital Underwater Communication Standard for the Network-Centric Maritime Battlespace*. NATO Official Website. https://www.nato.int/cps/en/natohq/news_145256.htm. 2017.

[96] M. B. Porter. *Ocean Acoustics Library Bellhop*. URL: http://oalib.hlsresearch.com/.

[100] Guerra Federo. *WOSS - World Ocean Simulation System*. URL: http://telecom.dei.unipd.it/ns/woss/.

[101] Fall Kevin und Varadhan Kannan. *The ns Manual*. URL: https://www.isi.edu/nsnam/ns/doc/ns_doc.pdf.

[102] ns-3 project. *ns-3 Network Simulator*. URL: https://www.nsnam.org/docs/release/3.33/models/ns-3-model-library.pdf.

[126] Wenxuan Chen u. a. *Vector Approximate Message Passing based Channel Estimation for MIMO-OFDM Underwater Acoustic Communications*. 2022. arXiv: 2211.12282 [eess.SP].

[148] GeoSpectrum Technologies Inc. *M27-930 FFR Communications Transducer*. URL: https://geospectrum.ca/wp-content/uploads/2021/05/M27-930-v1.pdf.

[149] NEPTUNE SONAR LIMITED. *MODEL T218*. URL: https://www.neptune-sonar.co.uk/downloads/0f58d582-8ea4-473a-b324-c07a3d45e5be/T218.pdf.

[204] Xilinx. *Xilinx*. URL: https://www.xilinx.com/products/silicon-devices/fpga.html.

[205] Intel. *Intel FPGA*. URL: https://www.intel.de/content/www/de/de/products/details/fpga.html.

[206] *Simulink HDL Coder/User's Guide*. http://cn.mathworks.com/help/pdf_doc/hdlcoder/hdlcoder_ug.pdf. Accessed: 08.04.2016.