

Kumulantenbasierte Modulationsartenerkennung in GNU Radio

Bachelorarbeit

Douglas Weber

Hauptreferent : Univ.-Prof. Dr.rer.nat. Friedrich K. Jondral
Betreuer : Dipl.-Ing. Sebastian Koslowski

Beginn : 1. Mai 2016
Abgabe : 31. Oktober 2016

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und unter Beachtung der Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der aktuellen Fassung angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche kenntlich gemacht.

Karlsruhe, den 31.10.2016

Douglas Weber



This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/4.0/>.

Abstract

In dieser Arbeit wird die automatische Erkennung von Modulationsarten mittels Kumulanten behandelt. Um eine Übertragung ohne Kenntnis von Symbolrate, Frequenz-, Zeit- und Phasenoffset zu ermöglichen, wird zudem der Entwurf eines geeigneten Empfängers behandelt, welcher sich um die entsprechende Synchronisation kümmert. Die in GNU Radio implementierten Elemente werden zudem unter verschiedenen Gesichtspunkten evaluiert.

Der Klassifikator, welcher die Entscheidung für eine der Modulationsarten Binary Amplitude Modulation (BPSK), Quadrature Phase-Shift Keying (QPSK), 8-Phase-Shift Keying (PSK) oder 16-Quadrature Amplitude Modulation (QAM) trifft, verwendet als Merkmal den Betrag der normierten Kumulante:

$$|\tilde{\kappa}^{(4,0)}| = \left| \frac{\hat{\kappa}^{(4,0)}}{(\hat{\kappa}^{(2,1)})^2} \right|, \quad (0.1)$$

hierbei stellt $\hat{\kappa}$ die auf der Basis von N Samples geschätzte Kumulante dar. Wir wählen die Entscheidungsgrenzen wie in Abbildung (Abb.) 0.1 veranschaulicht bei 0,34, bei 0,84 und bei 1,50. Die im Schaubild eingetragenen Werte entsprechen jeweils den theoretisch erwarteten Werten des Merkmals.

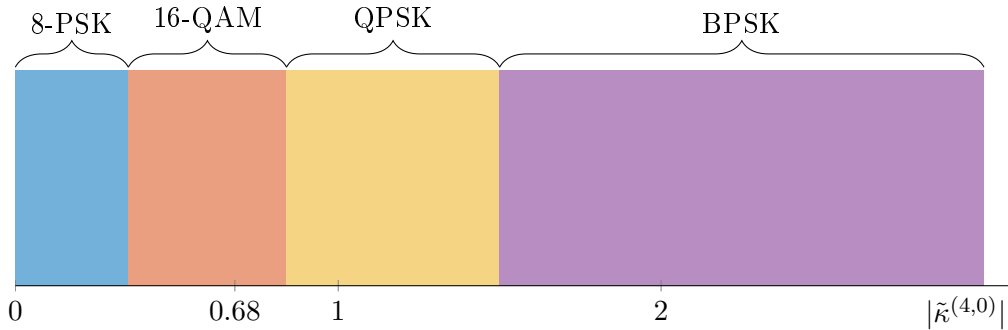


Abbildung 0.1.: Entscheidungsgrenzen des Klassifikators

Um die bei einer Übertragung auftretenden Einflüsse zu neutralisieren, ist eine Synchronisation notwendig. Die Frequenz- und Symbolratenschätzung basiert auf zyklischen Momenten. Aus deren Spektren lassen sich durch Bestimmung von Haupt- und Nebenmaximum die nötigen Werte ablesen. In Abb. 0.2 sind zwei normierte Spektren mit einem Frequenzoffset von 1kHz aufgetragen. Sie unterscheiden sich in ihrer Symboldauer, für die Beobachtungslänge gilt $N = 1 \times 10^5$. Das Hauptmaximum gibt den im Signal enthaltenen Frequenzoffset an, von den Nebenmaxima lässt sich auf die Symbolrate schließen.

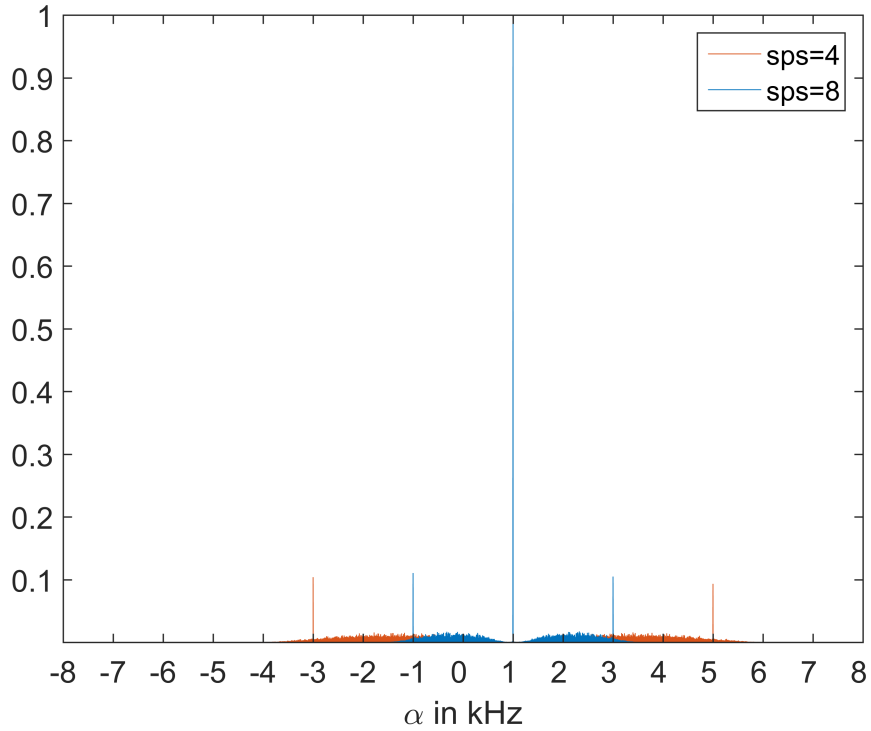


Abbildung 0.2.: Normierte Spektren mit verschiedenen Symbolraten und $f_{\text{offs}} = 1\text{kHz}$

Für eine Abtastung am idealen Zeitpunkt wird ein Phasenregelkreis zweiter Ordnung angewandt. Dabei werden M Filter mit phasenverschobenen Mached-Filtern erstellt, die Ausgabe eines dieser Filter wird an den Ausgang weitergeleitet. Um den idealen Abtastzeitpunkt zu finden, wird ein äußerer Phasenregelkreis für die Auswahl des Filters verwendet. Bei diesem Verfahren werden also Zeitsynchronisation und Matched-Filterung in einem vorgenommen.

Für die Bestimmung der Modulationsart wird der Betrag der Kumulante verwendet. Die Phasensynchronisation, welche einen Phasenoffset der Symbole beseitigt, wird daher nach der Klassifikation durchgeführt. Wir verwenden den Phasenschätzer

$$\hat{\varphi}_n = \frac{1}{r} \arg \left(\mu_i^{(r,0)} \sum_{k=1}^N \hat{s}_n[k]^r \right), \quad (0.2)$$

wobei r und $\mu_i^{(r,0)}$ von der Modulationsart abhängen. Da sie an dieser Stelle schon bekannt ist, können die entsprechenden Werte vom Schätzer eingesetzt werden.

Bei der Evaluation des Klassifikators gehen wir von einer perfekt funktionierenden Synchronisation und somit von einem reinen Additive White Gaussian Noise (AWGN) Kanal aus. Wir beobachten bei gesendetem 8-PSK die mit Abstand zuverlässigste Klassifikation, die Ergebnisse für BPSK und 16-QAM schneiden im Mittelfeld ab. Im Fall von gesendetem QPSK ist zur zuverlässigen Klassifikation mindestens eine Signal-to-Noise-Ratio (SNR) von

3dB notwendig. Die Variation der Beobachtungsdauer N hat einen geringen Einfluss auf die jeweiligen Klassifikationsraten.

In Abb. 0.3 betrachten wir die Abhängigkeit von $|\tilde{\kappa}^{(4,0)}|$ vom SNR. Darin sind die zur Klassifikation verwendeten Entscheidungsgrenzen als horizontale Linien eingezeichnet, die zugehörige Entscheidung lässt sich jeweils ablesen. Wir führen für die vier gesendeten Modulationsarten BPSK, QPSK, 8-PSK und 16-QAM jeweils 1000 Tests über den gesamten Bereich von $-15\text{dB} \leq \text{SNR} \leq 15\text{dB}$ durch, wobei jedes Ergebnis durch einen transparenten Punkt im Diagramm dargestellt ist. Die Farbe des Punktes ergibt sich anhand der Legende. Zur Schätzung verwenden wir eine Beobachtungsdauer von $N = 1000$.

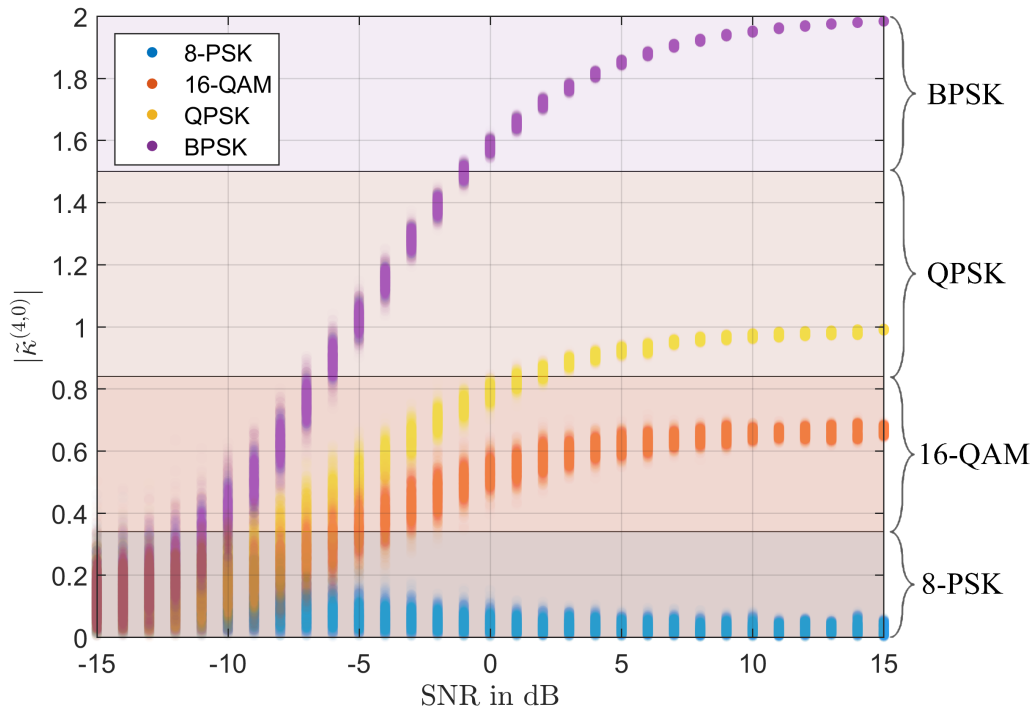


Abbildung 0.3.: Betrag der normierten Kumulante in Abhängigkeit des SNR mit $N = 1000$

Für ein hohes SNR liegen die Punktescharen bei allen Modulationen in den richtigen Entscheidungsbereichen. Sinkt das SNR ab, verzeichnen wir ein Abfallen der Scharen zu niedrigeren Werten des Merkmals. Im Fall von QPSK ist schon bei $\text{SNR} = 2\text{dB}$ eine Überschreitung der Entscheidungsgrenze und somit eine Falschentscheidung zu beobachten. Diese Grenze liegt im Fall von QPSK viel dichter an dem erwarteten Wert des Merkmals als bei den anderen Modulationsarten. Die hohe Klassifikationsrate bei niedriger Signalqualität für 8-PSK lässt sich dadurch erklären, dass das Entscheidungsmerkmal mit zunehmender Rauschähnlichkeit geringer wird und die Entscheidung daher in den Entscheidungsbereich von 8-PSK fällt.

Unter Verwendung von GNU Radio werden Empfänger und Klassifikator implementiert. Ein Flowgraph mit den realisierten Elementen ist in Abb. 0.4 gezeigt.

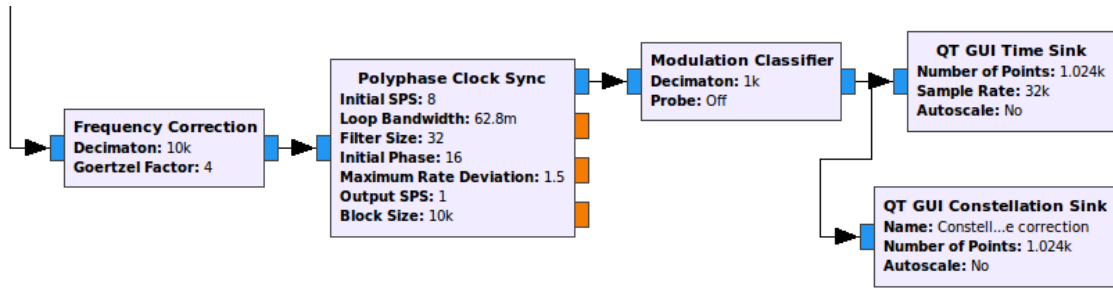


Abbildung 0.4.: Flowgraph des Empfängers

Im Block *Frequency Correction* werden Frequenzoffset und Symbolrate berechnet. Die Symbolrate wird an den nächsten Block weitergereicht. Mit dem Setzen des Parameters "Goertzel Faktor" ist es hierbei möglich, unter Verwendung des Goertzel-Algorithmus die Frequenzauflösung zu erhöhen. Die Zeitsynchronisation findet im Block *Polyphase Clock Sync* zusammen mit der Matched-Filterung statt. Dabei wird die vom vorhergehenden Block übermittelte Symbolrate verwendet. Nun liegen die rekonstruierten Symbole mit Phasenoffset vor.

Der Block *Modulation Classifier* führt die Klassifikation durch. Zudem findet dort der Ausgleich des Phasenoffsets statt. Wir erhalten somit am Ausgang der Simulation das Signal mit korrigierter Phase. Da das Modulationsverfahren nun bekannt ist, lässt sich ein Entscheider konstruieren, welcher je nach Modulation die Zuordnung der Symbole auf die Bits vornimmt.

Die Elemente *QT GUI Time Sink* und *QT GUI Constellation Sink* ermöglichen durch graphische Visualisierung die Verfolgung der Simulation in Echtzeit. Das erzeugte grafische Interface ist in Abb. 0.5 gezeigt. Im "Time Plot" sind Real- und Imaginärteil des Signals über der Zeit aufgetragen. Die Kennzeichnung "det_mod: QPSK" gibt an, dass die erkannte Modulation QPSK ist. Dies lässt sich durch das Konstellationsdiagramm im rechten Teil validieren.

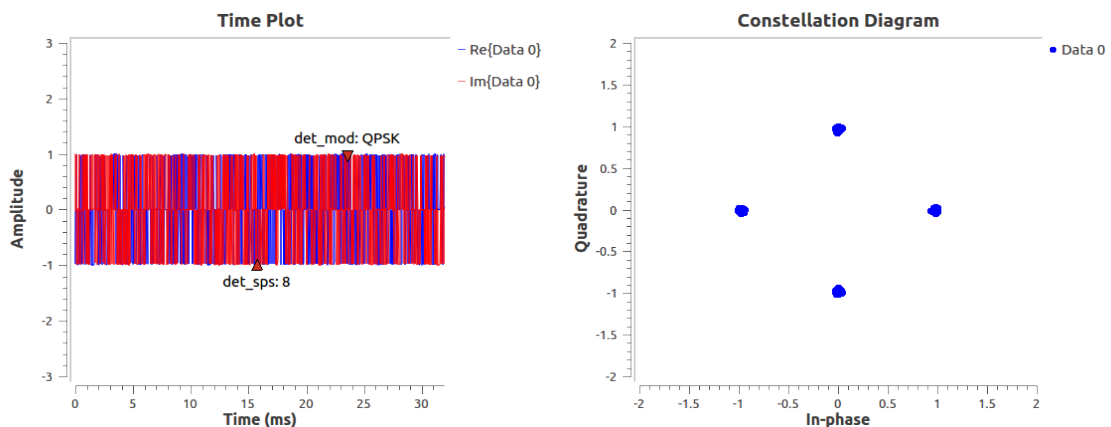


Abbildung 0.5.: Erzeugte Ausgabe beim Starten des Flowgraphs

Inhaltsverzeichnis

1. Einleitung	1
2. Grundlagen	3
2.1. Kenngrößen von Wahrscheinlichkeitsverteilungen	3
2.1.1. Konjugierte Kenngrößen bei komplexen Zufallsvariablen	4
2.1.2. Berechnung der Kumulante $\kappa^{(4,2)}$	4
2.1.3. Berechnungsvorschriften für verwendete Kumulanten	6
2.1.4. Stochastische Prozesse	6
2.2. System Modell	7
2.2.1. Sender und Modulation	7
2.2.2. Kanalmodell	9
2.2.3. Empfänger	10
2.3. Likelihood-Ratio-Tests	11
2.3.1. Zweihypothesen Problem	11
2.3.2. Bayes-Kriterium	12
2.4. Automatische Modulationsartenerkennung	13
3. Merkmalbasierter Klassifikator	15
3.1. Vorstellung des Klassifikators	15
3.2. Evaluation des Klassifikators	17
3.2.1. Konfusionsmatrizen	17
3.2.2. SNR Abhängigkeit der Klassifikationsrate	19
3.2.3. SNR Abhängigkeit des Merkmals	22
4. Entwurf eines Empfängers	25
4.1. Frequenzschätzung	25
4.1.1. Spektren der zyklischen Momente	25
4.2. Symbolratenschätzung	28
4.3. Zeitsynchronisation	29
4.4. Phasensynchronisation	31
5. Implementierung	35
5.1. GNU Radio	35
5.1.1. Beispiel Flowgraph	35
5.1.2. Stream Tags	36
5.1.3. Vector-Optimized Library of Kernels (VOLK)	36
5.2. Übersicht über den Flowgraph der Simulation	38
5.2.1. Sender und Kanal	38
5.2.2. Empfänger	40

5.2.3. Ausgabe der Simulation	41
5.3. Detaillierte Beschreibung der implementierten Elemente	42
5.3.1. Fokus auf Frequenzkorrektur und Symbolratenschätzung	42
5.3.2. Fokus auf die Zeitsynchronisation	47
5.3.3. Fokus auf den Klassifikator und Phasenoffset	48
6. Zusammenfassung und Fazit	51
A. Abkürzungsverzeichnis	53
Literaturverzeichnis	55

1. Einleitung

Es ist unbestreitbar, dass die drahtlose Datenkommunikation im heutigen Alltag eine prominente Rolle einnimmt. Quasi jeder besitzt ein mobiles Endgerät, mit dem immer größere Datenmengen übermittelt werden sollen.

Die Entwicklung zur verbesserten Störanfälligkeit und hin zu höheren Datenraten führte sich in den letzten Jahren stetig fort. Um den immer steigenden Anforderungen gerecht zu werden, ist es nötig, den Fortschritt weiter voranzutreiben und dabei auch neue Möglichkeiten der Optimierung in Betracht zu ziehen.

In dieser Arbeit wird dazu das Thema der automatischen Erkennung von Modulationsarten behandelt. Damit ist es möglich, die Modulationsart mithilfe von bestimmten Kenngrößen, sogenannten Kumulanten, zu bestimmen. Ziel dieser Arbeit ist die Realisierung einer automatischen Modulationsartenerkennung. Es soll zusätzlich ein Empfänger mit den nötigen Einheiten für die Synchronisation entworfen werden, welcher es ohne Vorkenntnisse ermöglicht, die Einflüsse des Kanals zu kompensieren. Die Implementierung von Klassifikator und Empfänger als Software-Defined Radio (SDR) soll mit GNU Radio erfolgen. Zur Demonstration sollen zudem Simulationen durchgeführt werden.

Eine solche Erkennung bringt den Vorteil mit sich, dass die Bandbreiteneffizienz erhöht wird. Bei konventionellen Übertragungen wird die Information über das Modulationsverfahren zusätzlich zu den Nutzdaten übertragen. Dieser obendrein erzeugte Overhead sorgt für eine niedrigere Effizienz. Wird die Modulationsart jedoch ohne vorherige Kenntnis aus den empfangenen Daten bestimmt, so bleibt mehr Bandbreite für die Nutzdaten. Der Vorteil fällt umso größer aus, je öfter das Modulationsverfahren bei einer Übertragung geändert wird. Dies ist der Fall, wenn sich die Signalqualität beispielsweise durch die Bewegung eines mobilen Teilnehmers häufig ändert. Dies erfordert dann den Wechsel zwischen robusteren Modulationsverfahren und komplexeren Verfahren, die mehr Übertragungskapazität bieten.

Eine weitere Anwendung findet sich im Bereich der Aufspürung und Einordnung von unbekannten Funksignalen. Nötig wird dies beispielsweise, wenn die Bundesnetzagentur Übertragungen einordnen möchte, über die keinerlei Informationen vorhanden sind.

In Kapitel 2 werden die Grundlagen und das Signalmodell, die im Verlauf der Arbeit verwendet werden, vorgestellt. Kapitel 3 beschäftigt sich mit dem Klassifikator, dabei wird dessen Funktionsweise erläutert und die Leistungsfähigkeit unter verschiedenen Bedingungen untersucht. Der Entwurf des Empfängers wird in Kapitel 4 behandelt, es wird dabei auf die Verarbeitungsschritte eingegangen, die nötig sind, um eine Übertragung trotz Kanaleinflüssen zu ermöglichen. Kapitel 5 beschäftigt sich mit der Implementierung in GNU Radio und behandelt die in der Simulation realisierten Elemente.

2. Grundlagen

In diesem Kapitel werden die verschiedenen Werkzeuge, die im Laufe der Arbeit benötigt werden, vorgestellt. Zur mathematischen Beschreibung von Empfangssignalen beschäftigen wir uns mit Themen aus der Wahrscheinlichkeitstheorie. Zudem wird ein Überblick über das angenommene Systemmodell gegeben. Themen, die die Klassifikation betreffen, werden in den Abschnitten 2.3 und 2.4 behandelt.

2.1. Kenngrößen von Wahrscheinlichkeitsverteilungen

Zufallsvariablen lassen sich durch ihre Wahrscheinlichkeitsdichten beschreiben. Zur Charakterisierung einer Zufallsvariable X mit der Dichte $f_X(x)$ kann man verschiedene statistische Kenngrößen angeben.

Das **r -te Moment** ergibt sich im Falle seiner Existenz zu:

$$\mu^{(r)} = \mathbf{E}\{X^r\} = \int_{-\infty}^{\infty} x^r f_X(x) dx . \quad (2.1)$$

Falls $r = 1$, erhält man mit $\mathbf{E}\{X\}$ den Erwartungswert der Zufallsvariablen X . Er gibt an, welchen Wert die Variable im Mittel annimmt.

Eine andere Kenngröße stellen **Kumulanten** dar. Die Kumulanten erzeugende Funktion ist nach [Rin08, S. 197] definiert zu:

$$\kappa^{(r)} = \left. \frac{d^r}{dt^r} \ln \mathbf{E}\{e^{sX}\} \right|_{s=0} . \quad (2.2)$$

Mit den Kumulanten erster, zweiter und dritter Ordnung erhält man Erwartungswert, Varianz und Schiefe, jene der Ordnung vier entspricht der Wölbung [Mü14, S. 8].

Außerdem ist es möglich, die Kumulante r -ter Ordnung mithilfe von höheren Momenten darzustellen. Die zugehörige Ausführung orientiert sich an [Mü14, S. 9-12]. Zuerst wollen wir die nötigen Voraussetzungen definieren um die Berechnung danach anschaulich in einem Beispiel durchzuführen. Zunächst wird die Menge P_r aller möglichen Partitionen ρ der Menge $\{1, \dots, r\}$ eingeführt:

$$P_r = \left\{ \rho : \rho = \{\nu_1, \dots, \nu_{|\rho|}\}, \nu_i \subseteq \{1, \dots, r\}, \nu_i \cap \nu_j = \emptyset, i \neq j, \bigcap_{i=1}^{|\rho|} \nu_i = \{1, \dots, r\} \right\} . \quad (2.3)$$

Damit lässt sich die Kumulante der Ordnung r nach [GS94] schreiben als:

$$\kappa^{(r)} = \sum_{\rho \in P_r} (-1)^{|\rho|-1} (|\rho| - 1)! \prod_{j=1}^{|\rho|} \mu^{\nu_j}. \quad (2.4)$$

2.1.1. Konjugierte Kenngrößen bei komplexen Zufallsvariablen

Handelt es sich um komplexe Zufallsvariablen, so lässt sich das Moment r -ter Ordnung auf Momente mit unterschiedlichen Konjugationen erweitern:

$$\mu^{(r,s)} = \mathbf{E}\{X^{r-s} X^{*s}\}, \text{ mit } s \in \{0, \dots, r\}. \quad (2.5)$$

Es sind $r + 1$ verschiedene Konjugationen möglich. Wie man erkennen kann, erhält man für $s = 0$ keine Konjugation und damit genau das Moment, welches in Gleichung (2.1) definiert wurde. Diese Erweiterung lässt sich auch für Kumulanten durchführen, indem man in Gleichung (2.4) die konjugierten Momente verwendet. Es ergibt sich die folgende Berechnungsvorschrift:

$$\kappa^{(r,s)} = \sum_{\rho \in P_r} (-1)^{|\rho|-1} (|\rho| - 1)! \prod_{j=1}^{|\rho|} \mu^{(r_j, s_j)}. \quad (2.6)$$

Die Menge P_r beinhaltet wie zuvor alle möglichen Partitionen der Elemente $\{1, \dots, r\}$. Die ersten $r - s$ Elementen der Menge $\{1, \dots, r\}$ werden nun ohne Beschränkung der Allgemeinheit den nicht konjugierten Termen und die letzten s Elemente den konjugierten Termen zugeordnet. Die Anzahl an Elementen in ν_j , die den nicht konjugierten Termen zugeordnet wurden, ist mit r_j bezeichnet. Für die Anzahl s_j gilt Gleiches bei konjugierten Termen.

2.1.2. Berechnung der Kumulante $\kappa^{(4,2)}$

Es soll nun anschaulich am Beispiel der Kumulante $\kappa^{(4,2)}$ erklärt werden, wie die Formel (2.6) Anwendung findet. Es gilt also $r = 4$ und $s = 2$.

Zuerst wird die Menge P_4 aller möglichen Partitionen der Menge $\{1, 2, 3, 4\}$ ermittelt. Die ersten $r - s$ Elemente, also 1 und 2, sind nicht konjugierten, die übrigen Elemente, 3 und 4, sind den konjugierten Termen zugeordnet. Bei jeder Partition ρ wird die Menge $\{1, 2, 3, 4\}$ in Teilmengen ν_i aufgeteilt, dabei kommt jedes Element der Menge genau einmal in einer Teilmenge vor (vgl. Gleichung (2.3)).

In Tabelle 2.1 ist P_4 aufgelistet, also alle möglichen Partitionen. Der Übersichtlichkeit halber beginnen wir bei der Partition mit nur einer Teilmenge ν_1 , welche alle Elemente enthält. Es folgen die Partitionen, die zwei Teilmengen ν_1 und ν_2 enthalten, also $|\rho| = 2$ gilt. Dabei gibt es sieben verschiedene Kombinationsmöglichkeiten, wie die Elemente der Menge $\{1, 2, 3, 4\}$ auf die zwei Teilmengen so aufgeteilt werden können, dass jedes Element genau einmal vorkommt. Ähnlich ist es auch bei den Partitionen mit drei und vier Teilmengen.

$\rho_1 = \{\nu_1\}$	$\nu_1 = \{1, 2, 3, 4\}$
$\rho_2 = \{\nu_1, \nu_2\}$	$\nu_1 = \{1\}, \nu_2 = \{2, 3, 4\}$
$\rho_3 = \{\nu_1, \nu_2\}$	$\nu_1 = \{2\}, \nu_2 = \{1, 3, 4\}$
$\rho_4 = \{\nu_1, \nu_2\}$	$\nu_1 = \{3\}, \nu_2 = \{1, 2, 4\}$
$\rho_5 = \{\nu_1, \nu_2\}$	$\nu_1 = \{4\}, \nu_2 = \{1, 2, 3\}$
$\rho_6 = \{\nu_1, \nu_2\}$	$\nu_1 = \{1, 2\}, \nu_2 = \{3, 4\}$
$\rho_7 = \{\nu_1, \nu_2\}$	$\nu_1 = \{1, 3\}, \nu_2 = \{2, 4\}$
$\rho_8 = \{\nu_1, \nu_2\}$	$\nu_1 = \{1, 4\}, \nu_2 = \{2, 3\}$
$\rho_9 = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1, 2\}, \nu_2 = \{3\}, \nu_3 = \{4\}$
$\rho_{10} = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1, 3\}, \nu_2 = \{2\}, \nu_3 = \{4\}$
$\rho_{11} = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1, 4\}, \nu_2 = \{2\}, \nu_3 = \{3\}$
$\rho_{12} = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1\}, \nu_2 = \{2, 3\}, \nu_3 = \{4\}$
$\rho_{13} = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1\}, \nu_2 = \{2, 4\}, \nu_3 = \{3\}$
$\rho_{14} = \{\nu_1, \nu_2, \nu_3\}$	$\nu_1 = \{1\}, \nu_2 = \{2\}, \nu_3 = \{3, 4\}$
$\rho_{15} = \{\nu_1, \nu_2, \nu_3, \nu_4\}$	$\nu_1 = \{1\}, \nu_2 = \{2\}, \nu_3 = \{3\}, \nu_4 = \{4\}$

Tabelle 2.1.: Menge aller Partitionen P_4 der Menge $\{1, \dots, 4\}$ [Mü14, S.11]

Nun wenden wir Gleichung (2.6) für die Berechnung von $\kappa^{(4,2)}$ an. Das Ergebnis sehen wir in Gleichung (2.7). Wir beginnen mit dem ersten Summanden, also der Partition ρ_1 . Sie enthält eine Teilmenge, für die Mächtigkeit gilt daher $|\rho_1| = 1$ und der Vorfaktor wird zu $(-1)^{(|\rho|-1)}(|\rho| - 1)! = +1$. Wie in Tabelle 2.1 zu sehen enthält die Teilmenge ν_1 bei ρ_1 die Elemente $\{1, 2, 3, 4\}$. Die Elemente 1 und 2 sind den nicht konjugierten Termen zugeordnet, da es zwei Elemente sind, gilt $r_1 = 2$. Entsprechendes gilt für die Elemente 3 und 4, womit wir $s_1 = 2$ erhalten. Für den ersten Summanden ergibt sich also das Produkt $1 \cdot \mu^{(r_1, s_1)} = \mu^{(2,2)} = \mathbf{E}\{X^2 X^{*2}\}$.

Nach gleichem Vorgehen erhalten wir beim zweiten Summanden für ρ_2 mit $|\rho_2| = 2$ einen Faktor von -1 vor dem Produktzeichen. Bei der Teilmenge ν_1 ergibt sich durch Zählen der Elemente $r_1 = 1$ und $s_1 = 0$ und bei ν_2 $r_2 = 1$ und $s_2 = 2$. Wir erhalten also $-1 \cdot \mu^{(r_1, s_1)} \cdot \mu^{(r_2, s_2)} = -\mathbf{E}\{X\} \mathbf{E}\{X X^{*2}\}$ als zweiten Summanden. Wird dies immer so fortgeführt, erhält man den in Gleichung (2.7) gezeigten Term.

$$\begin{aligned}
\kappa^{(4,2)} &= \mathbf{E}\{X^2 X^{*2}\} - \mathbf{E}\{X\} \mathbf{E}\{X X^{*2}\} - \mathbf{E}\{X\} \mathbf{E}\{X X^{*2}\} - \mathbf{E}\{X^*\} \mathbf{E}\{X^2 X^*\} \\
&\quad - \mathbf{E}\{X^*\} \mathbf{E}\{X^2 X^*\} - \mathbf{E}\{X^2\} \mathbf{E}\{X^{*2}\} - \mathbf{E}\{X X^*\} \mathbf{E}\{X X^*\} - \mathbf{E}\{X X^*\} \mathbf{E}\{X X^*\} \\
&\quad + 2\mathbf{E}\{X^2\} \mathbf{E}\{X^*\} \mathbf{E}\{X^*\} + 2\mathbf{E}\{X X^*\} \mathbf{E}\{X\} \mathbf{E}\{X^*\} + 2\mathbf{E}\{X X^*\} \mathbf{E}\{X\} \mathbf{E}\{X^*\} \\
&\quad + 2\mathbf{E}\{X\} \mathbf{E}\{X X^*\} \mathbf{E}\{X^*\} + 2\mathbf{E}\{X\} \mathbf{E}\{X X^*\} \mathbf{E}\{X^*\} + 2\mathbf{E}\{X\} \mathbf{E}\{X\} \mathbf{E}\{X^{*2}\} \\
&\quad - 6\mathbf{E}\{X\} \mathbf{E}\{X\} \mathbf{E}\{X^*\} \mathbf{E}\{X^*\} \\
&= \mathbf{E}\{X^2 X^{*2}\} - 2\mathbf{E}\{X\} \mathbf{E}\{X X^{*2}\} - 2\mathbf{E}\{X^*\} \mathbf{E}\{X^2 X^*\} - \mathbf{E}\{X^2\} \mathbf{E}\{X^{*2}\} \\
&\quad - 2\mathbf{E}\{X X^*\}^2 + 2\mathbf{E}\{X^2\} \mathbf{E}\{X^*\}^2 + 8\mathbf{E}\{X X^*\} \mathbf{E}\{X\} \mathbf{E}\{X^*\} + 2\mathbf{E}\{X\}^2 \mathbf{E}\{X^{*2}\} \\
&\quad - 6\mathbf{E}\{X\}^2 \mathbf{E}\{X^*\}^2
\end{aligned} \tag{2.7}$$

2.1.3. Berechnungsvorschriften für verwendete Kumulanten

Für einen mittelwertfreien stochastischen Prozess $X(t)$ ergeben sich die folgenden Berechnungsvorschriften für die Kumulanten der Ordnung zwei und vier [Mü14, S. 115]:

$$\kappa^{(2,0)}\{X(t)\} = \mathbf{E}\{X(t)^2\} \quad (2.8)$$

$$\kappa^{(2,1)}\{X(t)\} = \mathbf{E}\{X(t)X(t)^*\} \quad (2.9)$$

$$\kappa^{(4,0)}\{X(t)\} = \mathbf{E}\{X(t)^4\} - 3\mathbf{E}\{X(t)^2\}^2 \quad (2.10)$$

$$\kappa^{(4,1)}\{X(t)\} = \mathbf{E}\{X(t)^3X(t)^*\} - 3\mathbf{E}\{X(t)^2\}\mathbf{E}\{X(t)X(t)^*\} \quad (2.11)$$

$$\kappa^{(4,2)}\{X(t)\} = \mathbf{E}\{X(t)^2X(t)^{*2}\} - \mathbf{E}\{X(t)^2\}\mathbf{E}\{X(t)^{*2}\} - 2\mathbf{E}\{X(t)X(t)^*\}^2. \quad (2.12)$$

Gleichung (2.12) ergibt sich, wenn man für die im Beispiel hergeleitete Berechnungsvorschrift (2.7) einen mittelwertfreien Prozess annimmt. Da $\mathbf{E}\{X\} = \mathbf{E}\{X^*\} = 0$ gilt, werden alle Summanden in Gleichung (2.7), die einen dieser Faktoren enthalten, zu Null.

2.1.4. Stochastische Prozesse

Häufig ist es sinnvoll, keine Zufallsvariable, sondern einen stochastischen Prozess $X(t, \xi)$ zu betrachten. Man beschreibt damit eine Familie von Zufallsvariablen, die mit dem Parameter t indiziert ist. Interpretiert man t als die Zeit, erhält man für ein festes ξ_0 eine Realisierung des Prozesses [JW02, S.153].

Man kann beispielsweise eine empfangene Folge von Symbolen als stochastischen Prozess betrachten. Unter der Voraussetzung, dass der Prozess ergodisch ist, man also alle statistischen Eigenschaften aus einer einzigen Realisierung ableiten kann, lässt sich durch Zeitmittelwertbildung auf statistische Kenngrößen schließen.

Die Berechnung des r -ten Moments ergibt sich somit zu

$$\mu^{(r)} = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t)^r dt \quad (2.13)$$

Gehen wir nun von einer diskreten Folge der Länge N aus, wie im Falle von empfangenen Symbolen, so ist eine Schätzung möglich durch:

$$\hat{\mu}^{(r)} = \frac{1}{N} \sum_{n=1}^N x_n^r \quad (2.14)$$

Betrachtet man ein Empfangssignal vor der Abtastung und Filterung, kann man von einem zyklstationären Prozess $X(t)$ ausgehen, daher ergeben sich mit der Symboldauer T periodische statistische Kennwerte. Wichtig für die Berechnung des Frequenzoffsets wird später die Autokorrelationsfunktion, sie ist gegeben durch [JW02, S. 159]

$$\varphi_{XX}(\tau) = \mathbf{E}\{X(t_1)X(t_1 + \tau)^*\}. \quad (2.15)$$

2.2. System Modell

In diesem Abschnitt wird das Übertragungsmodell vorgestellt, das in dieser Arbeit angenommen wird. Unter anderem soll erklärt werden, welche Einflüsse bei der Übertragung auf das Signal wirken.

Das grundlegende Modell der Nachrichtenübertragung ist in Abbildung (Abb.) 2.1 dargestellt. In der Quelle erzeugte Nachrichten sollen zur Senke übertragen werden. Dazu sind die später genauer beschriebenen Elemente Sender, Empfänger und Kanal erforderlich.



Abbildung 2.1.: Grundlegendes Modell in der Nachrichtenübertragung [JW02, S. 13]

Da das Augenmerk dieser Arbeit auf der Klassifikation der Modulationsart liegt, wird der Entscheider im Empfänger hier nicht behandelt. Die Aufgabe des Entscheiders ist die Zuordnung der Empfangssymbole zu den Bits, also die Umwandlung der Symbolfolge in die an der Quelle eingebrachte Information. Der Empfänger beinhaltet einen Klassifikator, der die empfangene Modulationsart bestimmt und ausgibt.

2.2.1. Sender und Modulation

Nun soll näher auf die Modulation, die im Sender stattfindet, eingegangen werden. Die Information kommt als Bitfolge von der Quelle in den Sender und wird dort in ein Basisbandsignal umgewandelt, welches dann übertragen wird.

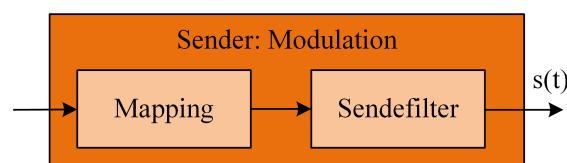


Abbildung 2.2.: Grundlegendes Modell in der Nachrichtenübertragung [JW02, S. 13]

Dazu ist zum einen das Mapping und zum anderen die Filterung mit einem Pulsformungsfilter nötig (vgl. Abb. 2.2). Beim Mapping werden ein oder mehrere Bits der Eingangsfolge auf Symbole abgebildet. Diese können auch komplexwertig sein. Die Zuordnung erfolgt nach einem festgelegten Schema, welches je nach Modulationsart unterschiedlich ist. Trägt man die möglichen Symbole einer Modulationsart auf einer komplexen Ebene auf, so erhält man das Konstellationsdiagramm der entsprechenden Modulation.

In Abb. 2.3 sind Konstellationsdiagramme der hier betrachteten Modulationsarten Binary Amplitude Modulation (BPSK), Quadrature Phase-Shift Keying (QPSK), 8-Phase-Shift

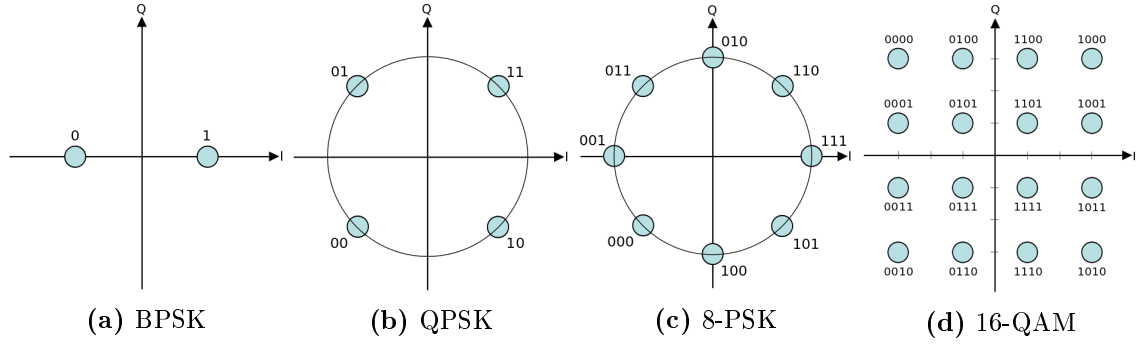


Abbildung 2.3.: Konstellationsdiagramme der Modulationen [wik06]

Keying (PSK) und 16-Quadrature Amplitude Modulation (QAM) abgebildet. In allen Fällen liegt eine Gray-Codierung vor, in dieser Arbeit wird auf Kodierung jedoch nicht eingegangen. Es ist zu erkennen, dass es bei BPSK zwei verschiedene Symbole gibt, denen jeweils entweder eine Eins oder eine Null zugeordnet ist. Bei QPSK in Abb. 2.3b hingegen kodiert jedes der vier Symbole zwei Bits, was einer Bitrate von Zwei entspricht.

Würde man die durch das Mapping gewonnene Folge von Sendesymbolen $s[kT]$ direkt (mit einem Rechtecksfiler) für eine Übertragung nutzen, so würde eine starke Außenbandstrahlung auftreten. Das bedeutet, dass die steilen Flanken des Zeitsignals zwischen den einzelnen Symbolen ein breites Frequenzspektrum verursachen und damit eine potenzielle andere Übertragung in benachbarten Frequenzbereichen stören könnte. Genauer dazu in [JW02, Kap. 7]. Um ein kompaktes Spektrum zu erreichen wird ein Filter zur Pulsformung verwendet.

Bei der Wahl des Sendefilters gibt es einen Trade-of zwischen einem möglichst kompakten Spektrum und geringer Intersymbolinterferenz (ISI) im Zeitbereich. Die Wahl eines Root-Raised-Cosine (RRC) Filters stellt hierbei einen guten Kompromiss dar. Die Impulsantwort $h(t)$ dieses Filters ist gegeben durch [Kam11, S. 56]

$$h(t) = \frac{2}{T} \cdot \frac{\sin(2\pi \frac{t}{T})}{2\pi \frac{t}{T}} \cdot \frac{\cos(r2\pi \frac{t}{T})}{1 - (4r \frac{t}{T})^2}. \quad (2.16)$$

Der Faktor r nennt sich Roll-off-Faktor, er beschreibt die Flankensteilheit des Filters. Für $r = 0$ ergibt sich ein ideales Tiefpassfilter und für $r = 1$ eine reine Kosinusflanke. Für den Fall, dass am Empfänger ebenfalls ein RRC Filter verwendet wird, liegt ein Matched Filter Empfang vor, was in Abschnitt 2.2.3 näher beleuchtet wird.

Das Signal wird bei der Filterung überabgetastet, enthielt vorher jedes Sample ein Symbol, wird ein Symbol nach der Filterung mit $r_{\text{symb}} \in \mathbb{R}$ Samples dargestellt. Die Symbolrate r_{symb} ist am Empfänger nicht bekannt.

Das Signal $s(t)$ am Ausgang des Senders ergibt sich also zu:

$$s(t) = \sum_{n=0}^{N-1} s[nT]g(t - nT), \quad 0 \leq t \leq NT. \quad (2.17)$$

Hierbei stellt t die Zeit, T die Symboldauer und $g(t)$ die Pulsform der Übertragung dar.

2.2.2. Kanalmodell

Der Kanal modelliert alle Einflüsse, die zwischen Senderausgang und Empfängereingang auf das Tiefpasssignal wirken. Diese Einflüsse erschweren den fehlerfreien Empfang. Sie haben jeweils physikalische Ursachen und lassen sich daher im Allgemeinen nicht vermeiden. In diesem Modell wird angenommen, dass stets eine Übertragung vorliegt, es muss also nicht erkannt werden, ob ein Signal vorliegt. Es wird zudem vorausgesetzt, dass im Falle einer Mehrwegausbreitung stets ein Pfad dominiert. Wir wollen im Folgenden näher auf die vorhandenen Kanaleinflüsse und deren jeweilige Ursachen eingehen.

Bei einer Funkübertragung ist es notwendig, das Tiefpasssignal in den Bandpassbereich zu überführen, da sich sonst keine elektromagnetische Wellen von der Antenne ablösen. Dies wird durch Hinzumischen einer hochfrequenten Trägerfrequenz eines Lokalen Oszillators erreicht. Man erhält somit das Bandpasssignal, es ist in Abb. 2.4 im Vergleich zum Tiefpasssignal dargestellt.

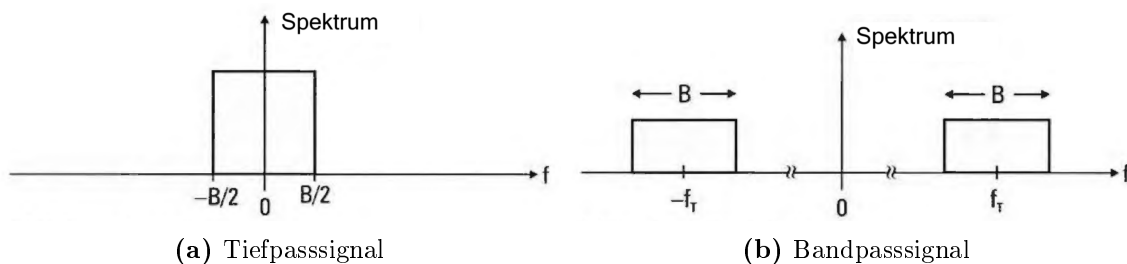


Abbildung 2.4.: Verschiedene Darstellungen eines Signals [JW02, S. 2]

Um am Empfänger wieder ein Tiefpasssignal zu erhalten, wird erneut eine hohe Frequenz hinzugemischt. Wir nehmen hierbei an, dass die ungefähre Trägerfrequenz am Empfänger bekannt ist, oder beispielsweise durch ein Leistungsdichtespektrum geschätzt wurde. Durch Schätzfehler und Bauteiltoleranzen der Oszillatoren sind die Frequenzen an Sender und Empfänger jedoch nicht exakt gleich. Daher bleibt ein geringer **Frequenzoffset** Δf im Tiefpasssignal des Empfängers bestehen.

Zusätzlich zu den nicht perfekt synchronisierten Frequenzen der Sende- und Empfängeroszillatoren besteht zwischen ihnen ein **Phasenoffset** Θ . Daher liegen die am Empfänger erkannten Symbole um Θ gedreht in der komplexen Ebene.

Da das Sampeln an Sender und Empfänger nicht taktsynchron abläuft, beinhaltet das am Empfänger anliegende Signal außerdem einen **Zeitoffset**. Dieser entsteht, da am Empfänger keine Information darüber vorliegt, an welcher Stelle des empfangenen Signals ein neues Symbol anfängt, beziehungsweise an welcher Stelle dieses abgetastet werden soll. Die Modellierung des Zeitoffsets wird durch den Parameter ε bewerkstelligt. Ist er eins, so liegt ideale Synchronisation vor, eine Abweichung von eins beschreibt einen Zeitoffset.

Einen weiteren bei Raumtemperatur unvermeidbaren Effekt stellt **thermisches Rauschen** dar, es tritt zum Beispiel in Verstärkern auf. Die Modellierung erfolgt durch Additive White Gaussian Noise (AWGN). Es tritt bei allen Frequenzen auf, aufgrund der Bandpasscharakteristik des Empfängers hängt die Rauschleistung zusätzlich von der Bandbreite des Systems ab.

Somit erhalten wir am Kanalausgang das Signal:

$$r(t) = Ae^{j2\pi\Delta ft}e^{j\Theta}s(t - \varepsilon T) + n(t) . \quad (2.18)$$

Hierbei stellt A die Amplitude, N die Anzahl übertragener Symbole, Δf den Frequenzoffset, Θ den Phasenoffset und ε den Zeitoffset dar. Durch $n(t)$ wird das AWGN Rauschen berücksichtigt.

In dem verwendeten Modell gehen wir von Block Fading aus, das bedeutet, dass die Änderung der Übertragungsparameter über die Beobachtungslänge eines Empfangsblocks vernachlässigbar gering ist.

2.2.3. Empfänger

Die Herausforderung am Empfänger ist allgemein die Rekonstruktion der Nachricht aus dem empfangenen Signal. Wie in Abb. 2.5 verdeutlicht, wird der Entscheider, der die entsprechende Empfangsbitfolge erzeugt, in dieser Arbeit nicht betrachtet. Durch Kanalkompensation und Filterung werden die Symbole rekonstruiert, der Klassifikator bestimmt außerdem die Modulationsart.

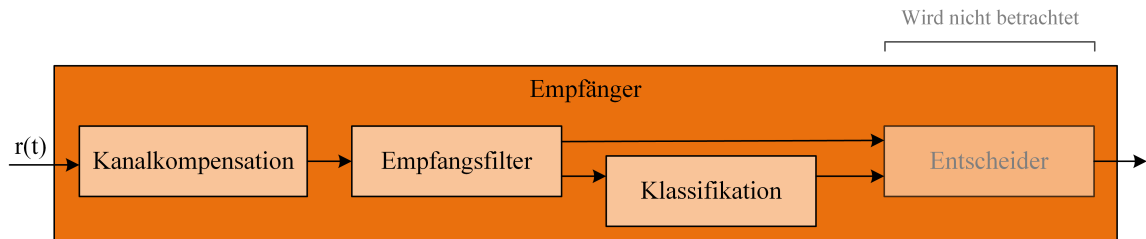


Abbildung 2.5.: Grundlegendes Modell in der Nachrichtenübertragung [JW02, S. 13]

Die Sendesymbole können jedoch trotz Phasenschätzung noch um 180° beziehungsweise um ein Vielfaches von 90° gedreht vorliegen. Das liegt an der unbekannten Absolutphase, das Problem kann durch geeignete Kodierung gelöst werden. Dies stellt jedoch nicht Gegenstand dieser Arbeit dar.

Im Empfänger wird zuerst die Kanalkompensation vorgenommen, es werden also die in Abschnitt 2.2.2 erläuterten Effekte rückgängig gemacht. Anschließend findet die Filterung am Empfangsfilter statt und danach wird die Klassifikation durchgeführt. Da das verwendete Modulationsverfahren am Empfänger nicht bekannt ist, wird es durch den Klassifikator bestimmt. Die einzelnen Schritte müssen nicht strikt voneinander getrennt ablaufen, beispiels-

weise kann es vorteilhaft sein, die Filterung mit Teilen der Kanalkompensation gemeinsam durchzuführen.

In dieser Beschreibung wird Matched Filterung vorausgesetzt. Dies bedeutet, dass Sende- und Empfangsfilter die gleiche RRC Form mit gleichem Roll-off-Faktor haben. Durch die zweimalige RRC Filterung ergibt sich durch deren Multiplikation ein Raised-Cosine (RC) gefiltertes Signal. Die Pulsform der Übertragung $g(t)$ entspricht demnach dem RC Filter. Es stellt sich heraus, dass sich die Wahl dieses Filters als besonders geschickt darstellt, am idealen Abtastzeitpunkt ist die ISI dann nämlich genau Null.

Das kontinuierliche Zeitsignal wird am Empfänger mit einer Samplerate r_{samp} abgetastet. Die Symbolrate r_{symb} ist am Empfänger unbekannt, das liegt zum einen daran, dass der Faktor der Überabtastung im Sendefilter nicht bekannt ist und zum anderen am Einfluss der unterschiedlichen Sampleraten von Sender und Empfänger. Da die Filtercharakteristik von der Symbolrate abhängt, muss diese dort geschätzt werden. Ebenso sind Schätzungen für unbekannten Frequenzoffset Δf , Phasenoffset Θ und Zeitoffset ε erforderlich. Durch Normierung wird der Einfluss der Amplitude A eliminiert.

Das nächste Element in der Kette stellt der Klassifikator dar. Er soll aus den rekonstruierten Symbolen bestimmen, welche Modulationsart bei dem Signal verwendet wurde. Der Klassifikator wird in Kapitel 3 ausführlich behandelt. Das Ergebnis der Klassifikation wird an den Entscheider weitergegeben. Er erzeugt aus den Symbolen und der erkannten Modulationsart den Bitstrom mit der Information.

2.3. Likelihood-Ratio-Tests

Um sich genauer mit Modulationsartenerkennung befassen zu können, werden nun einige Grundlagen zur Entscheidungstheorie behandelt. Zuerst betrachten wir ein System, bei dem sich das Problem auf eine Entscheidung zwischen zwei Hypothesen beschränkt. Das Verfahren lässt sich auch auf einen M Hypothesen Test erweitern, was dann der Entscheidung zwischen mehreren Modulationsarten entspricht. Die folgende Ausführung orientiert sich an [Kro11, S. 62 - 68].

2.3.1. Zweihypothesen Problem

Eine Nachrichtenquelle nimmt entweder den Zustand M_1 oder den Zustand M_2 an. Bezogen auf Modulationsartenerkennung entsprechen die Zustände M_i den Modulationsverfahren, die der Sender verwendet. Die *A-priori-Wahrscheinlichkeiten* P_1 und P_2 sind durch $P_1 = P(M_1)$ und $P_2 = P(M_2)$ gegeben. Am Empfänger liegt der gestörte Empfangsvektor \mathbf{r} vor. Die Aufgabe des Empfängers ist es, sich auf Grundlage von \mathbf{r} zwischen den Hypothesen H_1 (M_1 an der Quelle) und H_2 (M_2 an der Quelle) so zu entscheiden, dass das Ergebnis im Sinne eines noch festzulegenden Qualitätskriteriums optimal ist.

Die in Abb. 2.6 verwendeten bedingten Wahrscheinlichkeiten $P(H_j|M_i)$ lassen sich fol-

gendermaßen veranschaulichen: sie geben die Wahrscheinlichkeit dafür an, dass sich der Empfänger für Hypothese H_j entscheidet, unter der Bedingung, dass M_i an der Quelle anliegt [JW02, S. 27]. Falls also $j = i$ gilt, so liegt eine korrekte Entscheidung vor, ansonsten eine Fehlentscheidung.

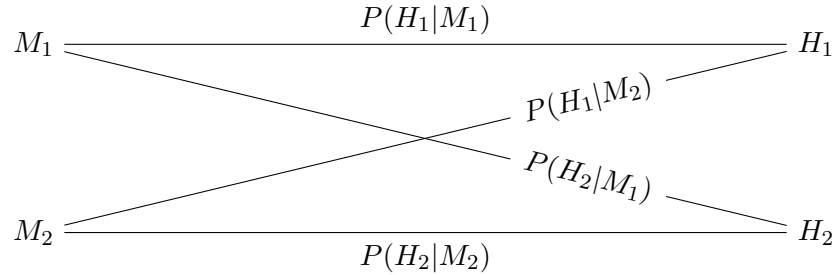


Abbildung 2.6.: Bedingte Wahrscheinlichkeiten bei Zweihypothesen Problem

2.3.2. Bayes-Kriterium

Neben den *A-priori-Wahrscheinlichkeiten* P_1 und P_2 sind für das Bayes-Kriterium außerdem die *Kosten*, die bei den jeweiligen Entscheidungen entstehen nötig. Die Kosten für einen korrekten Übergang, also von M_1 zu H_1 bzw. von M_2 zu H_2 sind naturgemäß gering, oder sogar negativ, die Kosten für Fehlentscheidungen sind höher.

Das Bayes-Kriterium besagt, dass das *Risiko* R minimiert werden soll. Es setzt sich aus den mit den Kosten gewichteten *A-priori-Wahrscheinlichkeiten* zusammen:

$$R = C_{11}P_1P(H_1|M_1) + C_{21}P_1P(H_2|M_1) + C_{12}P_2P(H_1|M_2) + C_{22}P_2P(H_2|M_2) . \quad (2.19)$$

Der Empfänger trifft eine Entscheidung für eine der Hypothesen H_j . Er teilt also den Beobachtungsraum \mathbf{R} , in dem alle möglichen Vektoren \mathbf{r} liegen, in zwei Teilräume R_1 und R_2 auf. Die Teilräume sind jeweils den Hypothesen H_1 und H_2 zugeordnet. Mit den bedingten Dichten $f_{\mathbf{R}|M_i}(\mathbf{r}|M_i)$ lassen sich die bedingten Wahrscheinlichkeiten in Gleichung (2.19) durch Integrale über die Entscheidungsräume R_1 und R_2 angeben:

$$R = C_{11} P_1 \int_{R_1} f_{\mathbf{R}|M_1}(\mathbf{r}|M_1) d\mathbf{r} + C_{21} P_1 \int_{R_2} f_{\mathbf{R}|M_1}(\mathbf{r}|M_1) d\mathbf{r} + C_{12} P_2 \int_{R_1} f_{\mathbf{R}|M_2}(\mathbf{r}|M_2) d\mathbf{r} + C_{22} P_2 \int_{R_2} f_{\mathbf{R}|M_2}(\mathbf{r}|M_2) d\mathbf{r} \quad (2.20)$$

Es gilt $\mathbf{R} = R_1 \cup R_2$, da R_1 und R_2 sich nicht überschneiden und zusammen \mathbf{R} ergeben. Daher folgt

$$\int_{R_2} f_{\mathbf{R}|M_i}(\mathbf{r}|M_i) d\mathbf{r} = 1 - \int_{R_1} f_{\mathbf{R}|M_i}(\mathbf{r}|M_i) d\mathbf{r} . \quad (2.21)$$

Für Gleichung (2.20) erhalten wir somit:

$$R = C_{22} P_2 + C_{21} P_1 + \int_{R_1} (P_2 \cdot [C_{12} - C_{22}] \cdot f_{\mathbf{R}|M_2}(\mathbf{r}|M_2) - P_1 \cdot [C_{21} - C_{11}] \cdot f_{\mathbf{R}|M_1}(\mathbf{r}|M_1)) d\mathbf{r} . \quad (2.22)$$

Die ersten beiden Summanden hängen nicht von der Wahl der Teilräume ab. Daher stellt sich die Optimierung des Entscheiders durch die Wahl der Teilräume und somit als Minimierung des Integrals in Gleichung (2.22) dar. Wie oben bereits erwähnt, sind die Kosten für eine richtige Entscheidung (C_{11} und C_{22}) geringer, als die für eine falsche Entscheidung (C_{12} und C_{21}). Daher sind die Terme in den eckigen Klammern positiv. Das Integral wird negativ, falls

$$P_1 \cdot [C_{21} - C_{11}] \cdot f_{\mathbf{R}|M_1}(\mathbf{r}|M_1) > P_2 \cdot [C_{12} - C_{22}] \cdot f_{\mathbf{R}|M_2}(\mathbf{r}|M_2) \quad (2.23)$$

gilt. Dann wird Hypothese H_1 angenommen, ansonsten H_2 . Durch Umformung von (2.23) erhält man:

$$\frac{f_{\mathbf{R}|M_1}(\mathbf{r}|M_1)}{f_{\mathbf{R}|M_2}(\mathbf{r}|M_2)} \underset{H_2}{\overset{H_1}{\geq}} \frac{P_2 \cdot [C_{12} - C_{22}]}{P_1 \cdot [C_{21} - C_{11}]} \quad (2.24)$$

Den linken Teil der Gleichung bezeichnet man als Likelihood-Ratio $\Lambda(\mathbf{r})$ und den rechten Teil als Schwelle η . Daher spricht man auch vom sogenannten Likelihood-Ratio-Test:

$$\Lambda(\mathbf{r}) \underset{H_2}{\overset{H_1}{\geq}} \eta . \quad (2.25)$$

Ein Empfänger berechnet hier also aus dem Vektor \mathbf{r} das skalare Likelihood-Verhältnis $\Lambda(\mathbf{r})$ und entscheidet dann auf Basis des Vergleichs mit der vorgegebenen Schwelle.

2.4. Automatische Modulationsartenerkennung

In diesem Abschnitt wird eine Übersicht über automatische Modulationsartenerkennungen gegeben. Grundsätzlich wird zwischen zwei Arten unterschieden, den likelihood- und den merkmalsbasierten Verfahren.

Erstere basieren auf der Likelihood Funktion des empfangenen Signals. Die Entscheidung fällt durch den Vergleich der Likelihood-Ratio mit einem Schwellenwert. Dazu werden die Wahrscheinlichkeitsdichten des Empfangssignals für jede Hypothese bestimmt. Dies geschieht durch die Modellierung des Kanals. Der Ansatz liefert eine optimale Lösung im Bayesschen Sinn, minimiert also die Fehlerwahrscheinlichkeit. Die optimale Lösung wird durch hohen Rechenaufwand erkauft [Mü14, S. 27 ff.].

Bei der merkmalsbasierten Klassifikation werden aus dem Signal Merkmale extrahiert, aufgrund derer die Entscheidung getroffen wird. Diese Methode liefert bei geringerem Aufwand

noch sehr gute Ergebnisse [DABNS07]. Diese Merkmale können zum Beispiel das Maximum der spektralen Leistungsdichte, die Standardabweichung der Amplitude, Phase oder Momentanfrequenz sein [Mü14, S. 28 ff.]. Eine weitere Möglichkeit besteht darin, Kumulanten zu verwenden. Hiermit lassen sich höherstufige Modulationen differenzieren. Ein merkmalsbasierter Klassifikator ist zwar nicht optimal, kann aber eine Performance nahe dem Optimum erreichen. Der Vorteil ist, dass der Berechnungsaufwand vergleichsweise gering ist, wodurch er für Echtzeitanwendungen interessant wird.

	BPSK	QPSK	8-PSK	16-PSK	16-QAM	64-QAM
$\kappa^{(2,0)}$	1	0	0	0	0	0
$\kappa^{(2,1)}$	1	1	1	1	1	1
$\kappa^{(4,0)}$	-2	1	0	0	-0,68	-0,62
$\kappa^{(4,1)}$	-2	0	0	0	0	0
$\kappa^{(4,2)}$	-2	-1	-1	-1	-0,68	-0,62
$\kappa^{(6,0)}$	16	0	0	0	0	0
$\kappa^{(6,1)}$	16	-4	0	0	2,08	1,80
$\kappa^{(6,2)}$	16	0	0	0	0	0
$\kappa^{(6,3)}$	16	4	4	4	2,08	1,80
$\kappa^{(8,0)}$	-272	-34	1	0	-13,98	-11,50
$\kappa^{(8,1)}$	-272	0	0	0	0	0
$\kappa^{(8,2)}$	-272	34	0	0	-13,98	-11,50
$\kappa^{(8,3)}$	-272	0	0	0	0	0
$\kappa^{(8,4)}$	-272	-34	-33	-33	-13,98	-11,50

Tabelle 2.2.: Kumulanten bis zu Ordnung acht für ausgewählte Modulationsarten

Tabelle 2.2 zeigt die Kumulanten bis zu Ordnung acht für ausgewählte Modulationsarten. Kumulanten ungerader Ordnung sind im Falle symmetrischer Verteilungsdichten und mittelwertfreier Zufallsvariablen immer null [Kö05, S. 15]. Außerdem gilt $\kappa^{(r,r-s)} = \kappa^{(r,s)}$ für $s \leq r$, weswegen diese Kumulanten nicht aufgeführt werden. Zur Unterscheidung von 8-PSK und 16-PSK werden mindestens die Kumulanten der Ordnung acht benötigt, da sich die theoretischen Werte bei niedrigerer Ordnung nicht unterscheiden. 16-PSK oder gar 32-PSK werden in der Praxis jedoch selten verwendet, daher findet deren Klassifikation in dieser Arbeit keine Anwendung.

3. Merkmalbasierter Klassifikator

In diesem Kapitel soll der verwendete Klassifikator untersucht werden. Er soll, basierend auf extrahierten Merkmalen, die Modulationsart eines empfangenen Signals bestimmen. Dazu wird jeweils eine bestimmte Anzahl von Samples herangezogen, der Klassifikator arbeitet also blockweise.

In diesem Kapitel gehen wir dazu von einem Additive White Gaussian Noise (AWGN) Kanal aus, siehe Abbildung (Abb.) 3.1. Wir nehmen also an, dass die Kanalkompensation perfekt arbeitet. Dadurch seien Sender und Empfänger ideal synchronisiert und die Symbolrate bekannt.

Um die Leistungsfähigkeit des Klassifikators zu untersuchen, wird in Abschnitt 3.2 die Genauigkeit der Entscheidungen bei fester Signal-to-Noise-Ratio (SNR) und verschiedenen Beobachtungsdauern N evaluiert. Wir betrachten außerdem die Abhängigkeit der Klassifikationsrate vom SNR. Zum Verständnis, wie der Klassifikator die Entscheidungen unter den gegebenen Bedingungen trifft, wird zusätzlich das zur Entscheidung verwendete Merkmal in Abhängigkeit des SNR untersucht.

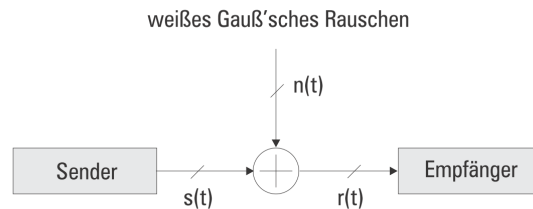


Abbildung 3.1.: AWGN-Kanal [JW02, S. 221]

3.1. Vorstellung des Klassifikators

Der hier vorgestellte Klassifikator orientiert sich an einer in [Mü14] vorgeschlagenen Methode. Es soll zwischen den vier Modulationsarten Binary Amplitude Modulation (BPSK), Quadrature Phase-Shift Keying (QPSK), 8-Phase-Shift Keying (PSK) und 16-Quadrature Amplitude Modulation (QAM) unterschieden werden können. Die zugehörigen Konstellationsdiagramme sind in den Abb. 2.3a bis 2.3d dargestellt. Wir nehmen an, dass immer eine dieser vier Modulationsarten vorliegt.

Für die Entscheidung wird der Betrag der normierten Kumulante $\tilde{\kappa}^{(4,0)}$ des Signals herangezogen. Nach [SS00] erhält man diesen durch:

$$|\tilde{\kappa}^{(4,0)}| = \left| \frac{\hat{\kappa}^{(4,0)}}{(\hat{\kappa}^{(2,1)})^2} \right| \quad (3.1)$$

Die Schätzung der Kumulante $\hat{\kappa}$ erfolgt auf Basis von je N Samples (vgl. Abschnitt 2.1.4). Die für die Schätzung verwendete Formel ergibt sich aus der Berechnungsvorschrift für $\kappa^{(4,0)}$ aus Gleichung (2.11) wie folgt:

$$\hat{\kappa}^{(4,0)} = \left[\frac{1}{N} \sum_{n=1}^N x_n^4 \right] - 3 \left[\frac{1}{N} \sum_{n=1}^N x_n^2 \right]^2 \quad (3.2)$$

Wie man in Tabelle 2.2 ablesen kann, werden für die verschiedenen Modulationen die folgenden Werte des Merkmals $|\tilde{\kappa}^{(4,0)}|$ erwartet, es unterscheidet sich für jede Modulation und ist daher als Unterscheidungsmerkmal geeignet:

$$\begin{aligned} \text{8-PSK : } & |\tilde{\kappa}^{(4,0)}| = 0 \\ \text{16-QAM : } & |\tilde{\kappa}^{(4,0)}| = 0,68 \\ \text{QPSK : } & |\tilde{\kappa}^{(4,0)}| = 1 \\ \text{BPSK : } & |\tilde{\kappa}^{(4,0)}| = 2 \end{aligned} \quad (3.3)$$

Basierend auf der in Abschnitt 2.3 eingeführten Theorie betrachten wir nun einen Sonderfall des Bayes-Kriteriums. Dazu setzen wir die Kosten für richtige Entscheidungen zu null und die Kosten für falsche Entscheidungen zu eins und erhalten damit das Maximum-a-posteriori-Kriterium (MAP). Die Schwere einer potenziellen Falschentscheidung ist also bei allen Modulationsarten gleich.

Die Schwelle des Likelihood-Verhältnis-Tests ergibt sich durch Einsetzen in Gleichung (2.24) zu

$$\eta = \frac{P_1}{P_2} \quad (3.4)$$

Im Falle eines M Hypothesen-Problems entscheidet sich der Empfänger für die Hypothese H_j , bei der der Ausdruck

$$P_j f_{R|M_j}(\mathbf{r}|M_j) \quad (3.5)$$

maximal wird. Dabei gehen wir von gleichen Auftrittswahrscheinlichkeiten P_j aus, wir nehmen also an, dass alle Modulationsarten gleich häufig gesendet werden. Das wird als Maximum-Likelihood-Ratio-Test bezeichnet [Mü14, S. 17 f.]. Zusätzlich nehmen wir an, dass etwaige Störungen bei der Schätzung der Kumulante mittelwertfrei sind und die Varianzen bei allen Modulationsarten gleich sind. Daher wählen wir die Entscheidungsgrenzen immer mittig zwischen den zu erwartenden Werten wie folgt:

$$\begin{aligned} |\tilde{\kappa}^{(4,0)}| < 0,34 & \Rightarrow \text{8-PSK} \\ 0,34 \leq |\tilde{\kappa}^{(4,0)}| < 0,84 & \Rightarrow \text{16-QAM} \\ 0,84 \leq |\tilde{\kappa}^{(4,0)}| < 1,50 & \Rightarrow \text{QPSK} \\ 1,50 \leq |\tilde{\kappa}^{(4,0)}| & \Rightarrow \text{BPSK} \end{aligned} \quad (3.6)$$

Die Entscheidungsgrenzen sind in Abb. 3.2 veranschaulicht.

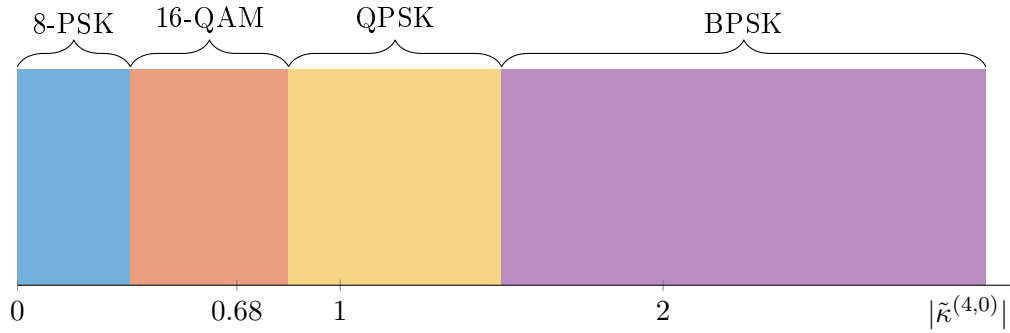


Abbildung 3.2.: Entscheidungsgrenzen des Klassifikators

Da der Betrag der Kumulante verwendet wird, liegen die Grenzen näher beieinander als bei einer etwaigen Verwendung der komplexwertigen Kumulante. Dies bringt eine Verschlechterung der Klassifikationsrate mit sich, da Störungen schneller zu Fehlentscheidungen führen können. Dies wird jedoch in Kauf genommen, da es sonst zu Problemen mit der Phasenschätzung kommen kann, Näheres dazu in Abschnitt 5.3.3.

3.2. Evaluation des Klassifikators

Nun soll untersucht werden, wie gut der Klassifikator unter gegebenen Bedingungen funktioniert. Im Folgenden werden, falls nicht anders angegeben, immer 1000 Tests durchgeführt, die Anzahl der Samples pro gesendetem Symbol r_{symb} sei acht. Hierzu werden im ersten Teil bei festem SNR Konfusionsmatrizen vorgestellt, mithilfe derer sich erkennen lässt, ob sich der Klassifikator für die korrekte Modulationsart entscheidet oder welcher Art eine potenzielle Falschentscheidung ist. In einem zweiten Teil soll die SNR-Abhängigkeit der Klassifikationsrate untersucht werden. Um die Gründe für potenzielle Falschentscheidungen des Klassifikators zu verstehen, wird in Abschnitt 3.2.3 die Abhängigkeit des zur Klassifikation verwendeten Merkmals vom SNR untersucht.

3.2.1. Konfusionsmatrizen

In einer Konfusionsmatrix ist je Zeile eine gesendete Modulation aufgetragen, in den Spalten wird angezeigt, wie oft welche Modulation erkannt wurde. Tabelle 3.1 zeigt die Konfusionsmatrizen des verwendeten Klassifikators bei der Symbolrate $r_{\text{symb}} = 8$ und einem SNR von 3dB. Die Anzahl der in jedem Test zur Entscheidung verwendeten Samples beträgt hier $N = 250, 500$, oder 1000.

Bei einer optimalen Klassifikation ist die Konfusionsmatrix eine Diagonalmatrix, bei der jeder Eintrag stets der Anzahl an Tests entspricht. In Tabelle 3.1 erkennt man, dass es im Fall von gesendetem QPSK zu Fehlentscheidungen kommt. Für alle anderen Modulationen

liegt bei allen betrachteten Beobachtungsdauern immer die korrekte Entscheidung vor. Die Problematik der Falschentscheidung bei QPSK wird umso geringer, je größer die Beobachtungsdauer wird. So liegen für $N = 250$ noch bei ca. 10% der Tests Fehlentscheidungen vor, bei $N = 1000$ nur noch 0,3%. Trotz des für Datenübertragungen verhältnismäßig niedrigen SNR von 3dB ist bei $N = 1000$ also eine einigermaßen verlässliche Klassifikation des Signals möglich.

Erkannt Gesendet	8-PSK	16-QAM	QPSK	BPSK
$N = 250$				
8-PSK	1000	0	0	0
16-QAM	0	1000	0	0
QPSK	0	95	905	0
BPSK	0	0	0	1000
$N = 500$				
8-PSK	1000	0	0	0
16-QAM	0	1000	0	0
QPSK	0	12	988	0
BPSK	0	0	0	1000
$N = 1000$				
8-PSK	1000	0	0	0
16-QAM	0	1000	0	0
QPSK	0	3	997	0
BPSK	0	0	0	1000

Tabelle 3.1.: Konfusionsmatrizen bei SNR = 3dB, $N = 250, 500, 1000$

Es fällt auf, dass die Falschentscheidungen bei gesendetem QPSK immer auf 16-QAM fallen. Anschaulich lässt sich dies auch durch zunehmende Ähnlichkeit der Konstellationsdiagramme der beiden Modulationsarten bei geringem N und niedrigem SNR erklären.

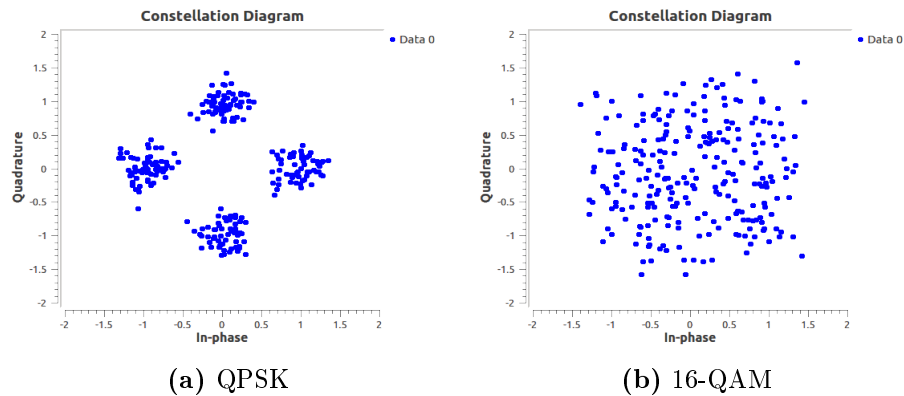


Abbildung 3.3.: Konstellationsdiagramme zum Vergleich bei SNR = 3dB mit $N = 250$ Symbolen.

Die normierten Konstellationsdiagramme der Signale nach Matched-Filter Abtastung sind zum Vergleich in Abb. 3.3 dargestellt. In den Diagrammen sind jeweils $N = 250$ Symbole aufgezeigt. Der Grund, warum sich der Klassifikator unter diesen Bedingungen falsch entscheidet, wird in Abschnitt 3.2.3 genauer beleuchtet.

3.2.2. SNR Abhängigkeit der Klassifikationsrate

Bisher wurde ein konstantes SNR von 3dB angenommen. Der Klassifikator soll jedoch auch bei niedrigem SNR eine möglichst hohe Klassifikationsrate aufweisen. Dazu wird bei festen Beobachtungsdauern N die Wahrscheinlichkeit für eine korrekte Entscheidung für die vier Modulationsarten über dem SNR aufgetragen. Dieses variiert von -15dB bis 15dB .

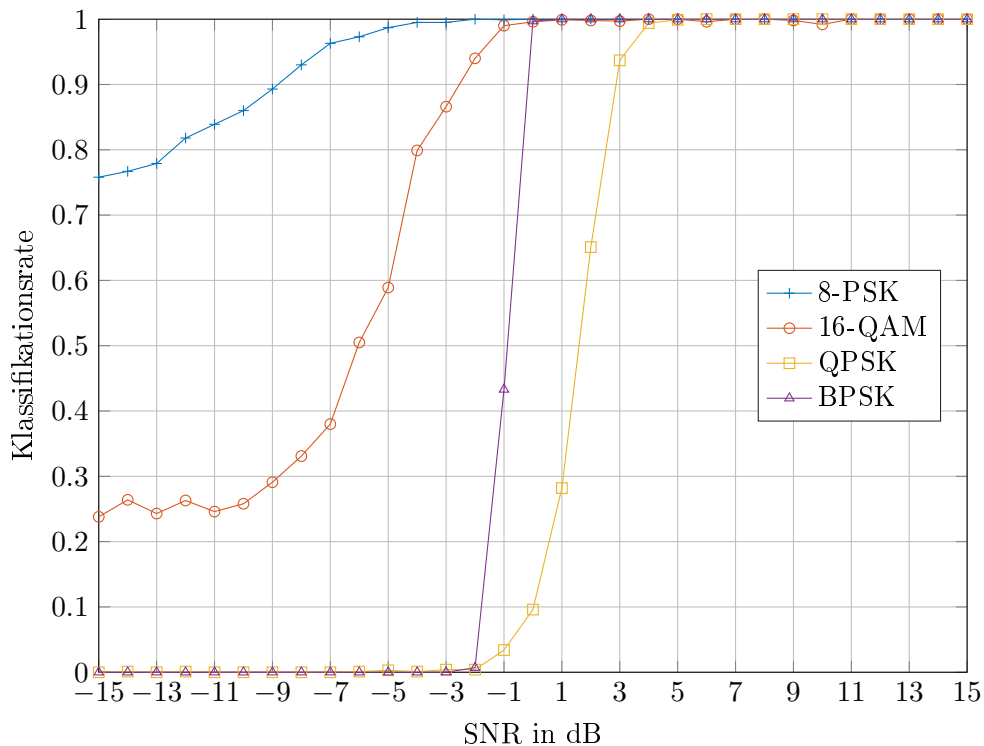


Abbildung 3.4.: Klassifikationsrate in Abhängigkeit des SNR mit $N = 250$

In Abb. 3.4 wurde $N = 250$ gewählt. Es ist zu erkennen, dass mit 8-PSK bei $\text{SNR} \geq -4\text{dB}$ eine nahezu optimale Klassifikation gewährleistet ist. Für darunter sinkendes SNR fällt die Rate vergleichsweise langsam ab. Im Fall von 16-QAM liegt schon für $\text{SNR} \leq -1\text{dB}$ keine zuverlässige Klassifikation mehr vor. Die Kurve fällt etwas steiler ab als bei 8-PSK, sinkt jedoch auch für eine sehr niedrige Signalqualität nicht unter 20%. Bei 16-QAM kommt es auch bei sehr hohem SNR vereinzelt schon zu Falschentscheidungen.

Anders bei QPSK und BPSK, hier fällt die Klassifikationsrate innerhalb eines schmalen SNR-Bereichs auf null ab. Bei gesendetem BPSK ist ein Abfallen ab $\text{SNR} = 0\text{dB}$ zu

beobachten, bei QPSK hingegen schon ab $\text{SNR} = 4\text{dB}$.

Bei einer auf $N = 500$ erhöhten Sampleanzahl ergibt sich das in Abb. 3.5 dargestellte Diagramm. Mit der längeren Beobachtungslänge erwarten wir auch bessere Raten für eine korrekte Klassifikation. Diese Erwartung bestätigt sich am deutlichsten für gesendetes 8-PSK, die Kurve bewegt sich nun für $\text{SNR} \geq -8\text{dB}$ am Optimum und fällt für ein niedrigeres SNR nicht unter 90%.

Im Fall von gesendetem 16-QAM ist bei hohem SNR nun ein einwandfreies Ergebnis zu beobachten und es treten keine vereinzelt Fehlentscheidungen mehr auf. Wie bei $N = 250$ gelingt für $\text{SNR} \leq -1\text{dB}$ keine optimale Klassifikation mehr und die Kurve fällt ab, stagniert diesmal jedoch nicht bei 20%, sondern fällt knapp unter 10%.

Bei den anderen beiden Modulationsarten QPSK und BPSK ändert sich kaum etwas an den jeweiligen Kurven, lediglich der Übergang der Klassifikationsrate von 100% zur 0% ist jeweils steiler als bei geringerer Beobachtungsdauer.

Durch eine erneute Verdopplung der Sampleanzahl auf $N = 1000$ erhalten wir die in Abb. 3.6 gezeigte Darstellung. Durch diese Modifikation verbessert sich die Klassifikation für 8-PSK nochmals. Nun ergibt sich eine nahezu optimale Klassifikationsrate über den gesamten getesteten Bereich von $-15\text{dB} \leq \text{SNR} \leq 15\text{dB}$. Bei den übrigen Modulationsverfahren ergeben sich vergleichbare Änderungen wie bei der vorherigen Erhöhung der Beobachtungslänge. Die SNR Werte, ab denen die Klassifikationsrate jeweils einbricht, ändern sich kaum, die Kurven fallen jedoch schneller ab. Bei 16-QAM fällt die Rate der korrekten Klassifikation nun auch auf 0% ab.

Zusammenfassend lässt sich feststellen, dass die Klassifikation von 8-PSK am zuverlässigsten funktioniert. Bei einer Beobachtungsdauer von $N = 1000$ ist hier sogar bei sehr niedrigem SNR eine quasi optimale Klassifikation gewährleistet. Bei der Modulation 16-QAM ist erst für $N \geq 500$ bei guter Signalqualität eine perfekte Klassifikation möglich. Bei den anderen Modulationsverfahren bringt eine Verlängerung der Beobachtungsdauer eine geringe Verbesserung.

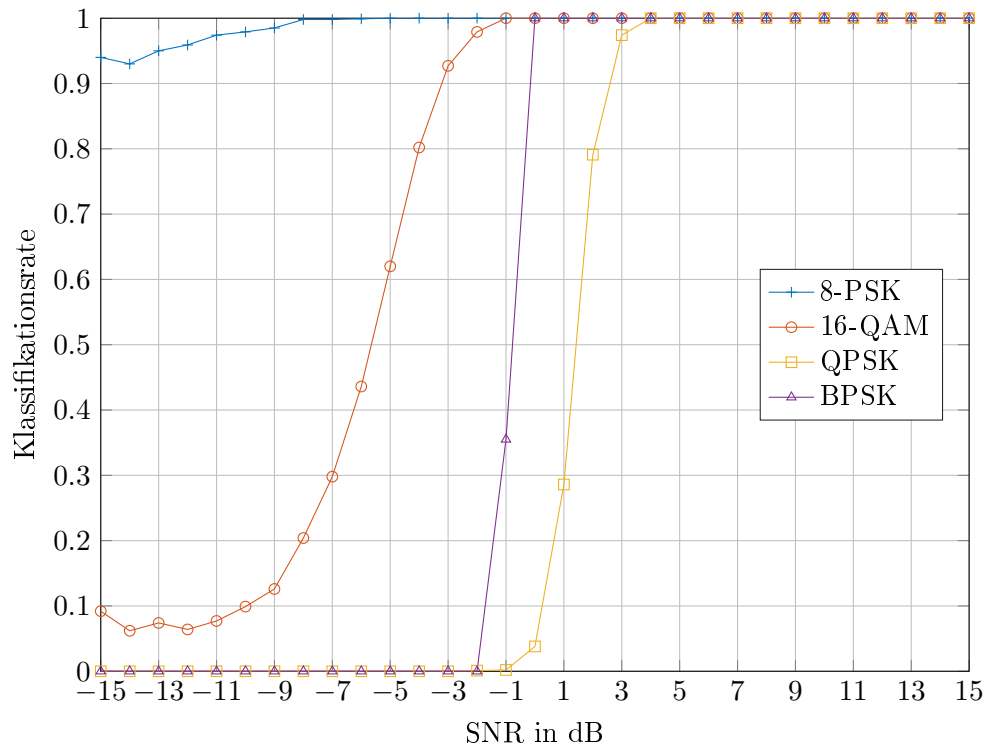


Abbildung 3.5.: Klassifikationsrate in Abhängigkeit des SNR mit $N = 500$

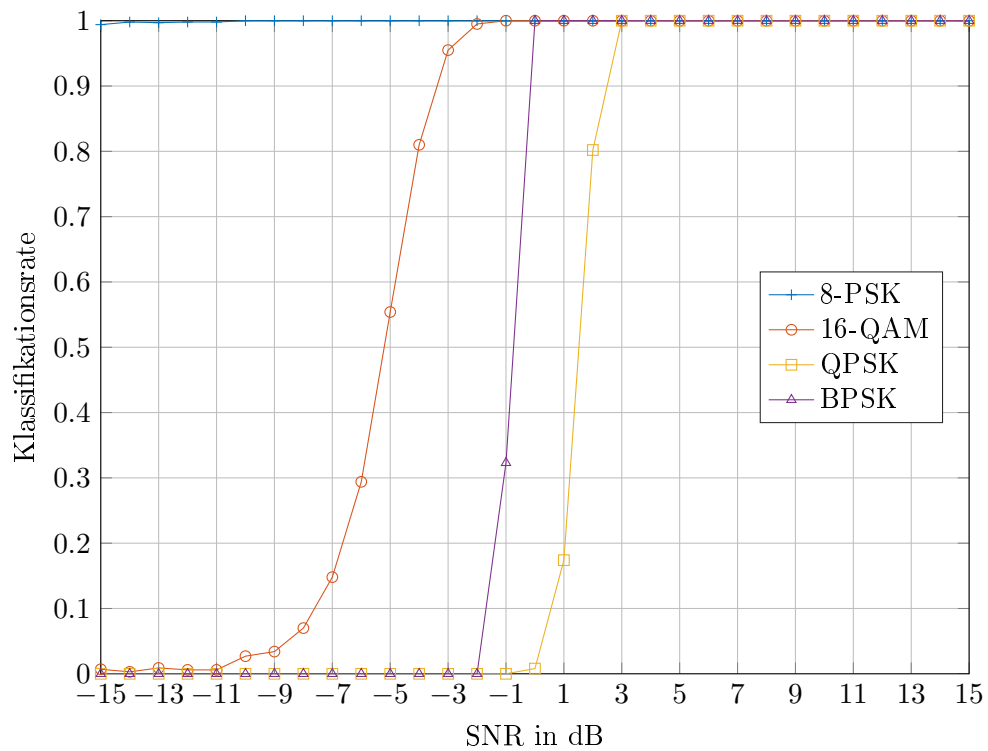


Abbildung 3.6.: Klassifikationsrate in Abhängigkeit des SNR mit $N = 1000$

3.2.3. SNR Abhängigkeit des Merkmals

Zur Erklärung der vorangegangenen Ergebnisse wollen wir nun die Abhängigkeit des Betrags der normierten Kumulante vom SNR untersuchen. Dazu wählen wir zuerst eine Beobachtungsdauer von $N = 250$ und erhalten das Diagramm in Abbildung 3.7. Das Ergebnis eines jeden Tests wird darin durch einen transparenten Punkt beim entsprechenden SNR visualisiert.

Die Entscheidungsgrenzen, die zur Klassifikation verwendet werden, sind als horizontale Linien eingezeichnet, die jeweilige Entscheidung ist rechts neben dem Diagramm notiert. Die Farben der Punkte in dem Diagramm geben entsprechend der Legende die gesendete Modulationsart an. Solange sich ein Punkt einer gesendeten Modulationsart also im entsprechenden Entscheidungsbereich der Modulation befindet, entscheidet sich der Klassifikator korrekt. Von dem Moment an, ab dem ein Punkt außerhalb des Entscheidungsbereichs ist, liegt eine Fehlentscheidung vor.

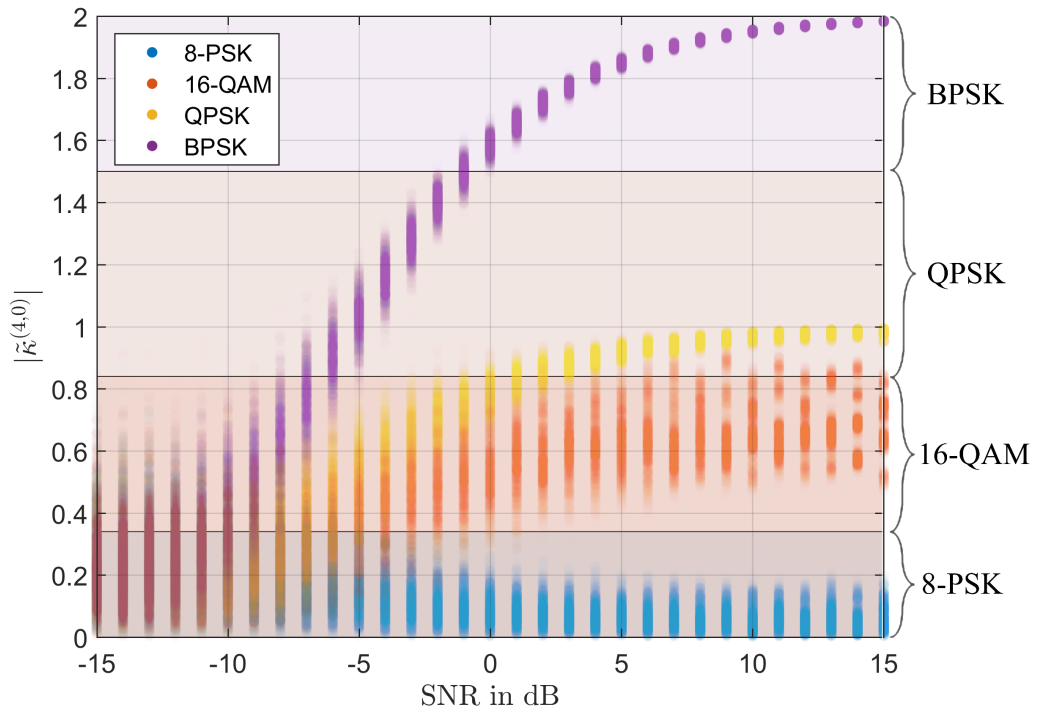


Abbildung 3.7.: Betrag der normierten Kumulante in Abhängigkeit des SNR mit $N = 250$

Es lässt sich beobachten, dass die berechnete Kumulante bei hohem SNR stets den Wert annimmt, den wir nach Gleichung (3.3) für die jeweilige Modulation erwarten. Zusätzlich fällt auf, dass die Streuung des Merkmals im Fall von 16-QAM stärker ausfällt als bei den anderen Modulationen. Wie man sieht, befinden sich auch bei hohem SNR einige Punkte außerhalb des Entscheidungsbereichs von 16-QAM, was Falschentscheidungen anzeigt. Für ein absinkendes SNR wird der Wert des Merkmals zunehmend geringer, was dazu führt,

dass unterhalb eines bestimmten SNR keine korrekte Klassifikation mehr vorliegt. Eine Ausnahme stellt hierbei die Klassifikation von 8-PSK dar. Die erwartete Kumulante ist null, deshalb ist auch bei sehr niedriger Signalqualität noch eine hohe Klassifikation zu erwarten.

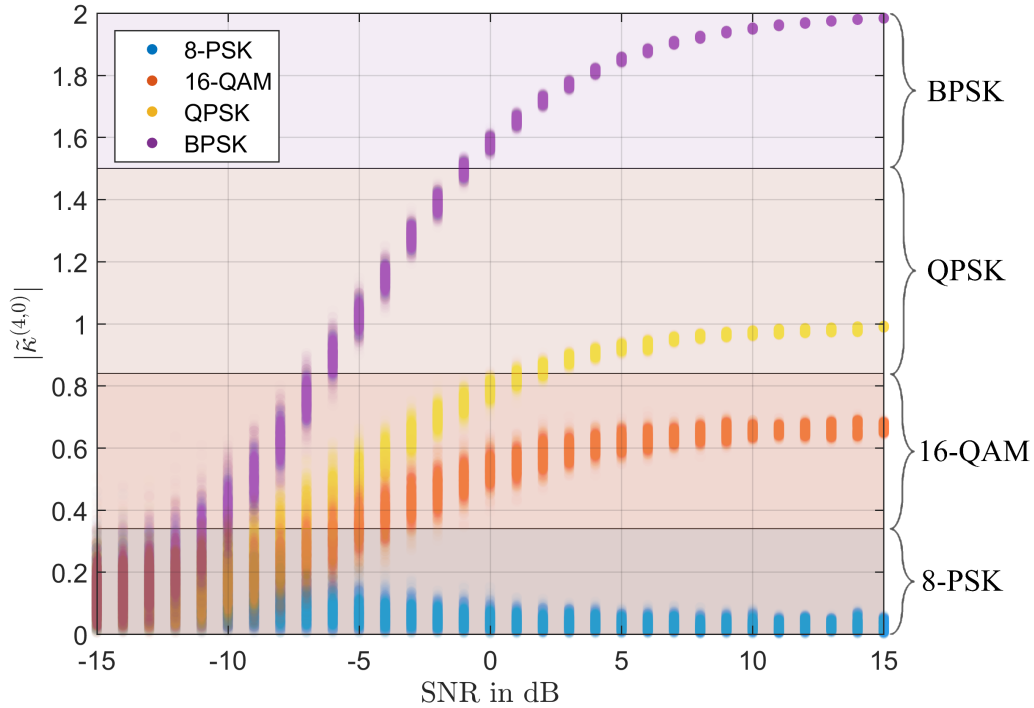


Abbildung 3.8.: Betrag der normierten Kumulante in Abhängigkeit des SNR mit $N = 1000$

Nun betrachten wir das Merkmal bei einer Beobachtungsdauer von $N = 1000$ und stellen dabei fest, dass die Varianzen des geschätzten Merkmals bei allen gesendeten Modulationen geringer sind. Dies spiegelt sich in der Schärfe der jeweiligen Punkteschar wider. Auch im Fall von 16-QAM sorgt die Schärfe nun dafür, dass bei hohem SNR immer korrekt klassifiziert wird. Dadurch erklärt sich die in Abschnitt 3.2.2 verzeichnete minimal erforderliche Beobachtungslänge. Für die anderen Modulationsarten führt eine Erhöhung also kaum zu einer Verbesserung der Klassifikationsrate, weil der Übergang in ein falsches Entscheidungsgebiet durch die erhöhte Schärfe lediglich innerhalb eines kleineren SNR Bereichs vonstatten geht. Auch dies deckt sich mit den Ergebnissen aus Abschnitt 3.2.2.

Der Grund, warum wir für die Klassifikation von QPSK ein vergleichsweise hohes SNR benötigen, liegt offenbar an dem geringen Abstand zu der Entscheidungsgrenze zu 16-QAM. Nach Gleichung (3.3) und (3.6) beträgt dieser lediglich 0,16. Wie im Diagramm zu erkennen, wird diese schon überschritten, sobald ein SNR von ca. 3dB unterschritten wird.

Es ist nun ebenfalls möglich, zu erklären, warum die Klassifikation von 8-PSK auch bei niedrigem SNR so gut funktioniert: Die Entscheidung fällt in diesem Fall immer auf 8-PSK, weil solch ein rauschähnlicher Prozess bei ausreichender Beobachtungslänge eine Kumulante

von null erzeugt. Dies lässt sich sehr schön durch die Punkte erkennen, die bei einem $\text{SNR} \leq -13$ alle nahe null liegen und somit im Entscheidungsbereich von 8-PSK. Dabei spielt auch mit, dass es keine Hypothese gibt, die keine der vier Modulationsarten annimmt.

4. Entwurf eines Empfängers

Bisher wurde vorausgesetzt, dass ein Matched-Filter-Empfang mit idealer Zeit- und Frequenzsynchronisation vorliegt. Bei einer realen Übertragung ist dies nicht mehr erfüllt, da bei einer Übertragung durch die Luft verschiedene Kanaleinflüsse hinzukommen. Von nun an soll daher das in Abschnitt 2.2 definierte Systemmodell gelten, darin sind die Einflüsse erläutert. In diesem Kapitel geht es nun darum, wie diese Kanaleinflüsse rückgängig gemacht werden.

4.1. Frequenzschätzung

Es wird davon ausgegangen, dass das empfangene Signal ungefähr um die Mittenfrequenz null zentriert ist. Dies geschieht durch Heruntermischen des Bandpasssignals in den Tiefpassbereich unter Voraussetzung der groben Kenntnis der Trägerfrequenz. Es bleibt ein kleiner Frequenzoffset, dieser muss korrigiert werden, da er sehr starken Einfluss auf die Empfangsqualität hat.

Zur Korrektur wird ein Verfahren angewandt, welches in [Mü14, S. 80-89] vorgestellt wird. Da die Korrektur vor der Matched-Filterung stattfindet, ist das Signal zyklstationär und die Momente des Signals zeitabhängig. Zur Bestimmung des Frequenzoffsets verwenden wir die zyklischen Momente $\mu_\alpha^{(r,s)}(\tau)$, welche sich durch Fouriertransformation des r -ten Moments des Empfangssignals berechnen lassen:

$$\mu_\alpha^{(r,s)}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} \mu^{(r,s)}(t, \tau) e^{-j2\pi\alpha t} dt \quad (4.1)$$

Hierbei stellt τ , entsprechend der Autokorrelationsdefinition (vgl. Abschnitt 2.1.4), den Wert dar, um welchen die Momente bei der Berechnung gegeneinander verschoben werden. Die zyklische Frequenz α ist gegeben durch $\alpha = \frac{m}{T}$. Nach [Mü14, Abschnitt 2.1.4] treten bei den zyklischen Momenten Spektrallinien bei $\frac{m}{T} + f_T$, $m \in \mathbb{Z}$ auf, der Frequenzoffset f_T lässt sich durch Suche des Hauptmaximums bestimmen.

4.1.1. Spektren der zyklischen Momente

Wir wollen nun näher auf die Spektren der zyklischen Momente $\mu_\alpha^{(r,s)}(\tau)$ und deren Abhängigkeit von τ und der zur Schätzung verwendeten Fensterbreite K eingehen. Außerdem

soll die Charakteristik der Spektren verschiedener Ordnungen in Abhängigkeit der Modulationsart beleuchtet werden.

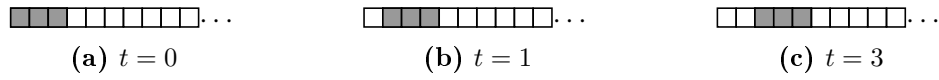


Abbildung 4.1.: Visualisierung der Fensterung bei $K = 3$

Bei jedem Zeitschritt t wird die Schätzung immer mit Ausschnitten der Fensterbreite K durchgeführt. Zur Veranschaulichung ist der Ablauf für die ersten drei Zeitschritte bei $K = 3$ in Abbildung (Abb.) 4.1 veranschaulicht. Die für die Berechnung verwendeten Samples sind grau eingefärbt.

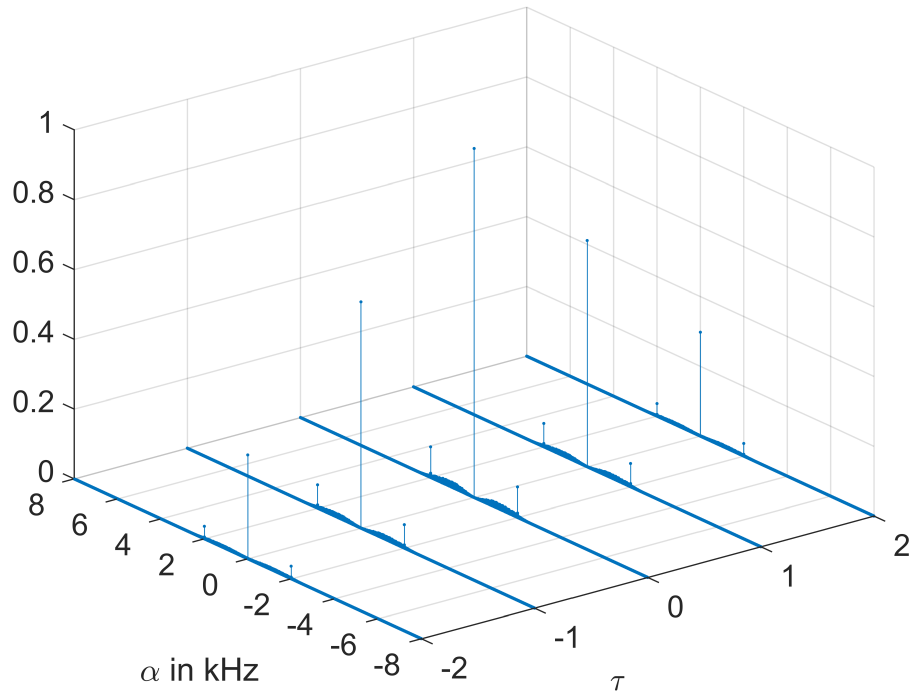


Abbildung 4.2.: Normierte Spektren der zyklischen Momente mit τ -Abhängigkeit, $N = 1 \times 10^5$, $K = 3$

Zusätzlich lassen sich Signale vor der Berechnung um τ gegeneinander verschieben. Für jedes τ wird nun mittels Fouriertransformation das Spektrum in Zeitrichtung berechnet und wir erhalten einen Graphen mit den Spektren von $\mu_{\alpha}^{(r,s)}(\tau)$ in α - und τ -Abhängigkeit. In Abb. 4.2 sind die auf eins normierten Spektren bei Verwendung von $N = 1 \times 10^5$ Samples, einer N Punkte Fast Fourier Transform (FFT) und $r_s = 32\text{kHz}$ dargestellt. In diesem Beispiel ist der Frequenzoffset null und es wird Binary Amplitude Modulation (BPSK) eingesetzt.

Man erkennt, dass die Maxima für alle τ bei den gleichen zyklischen Frequenzen α auftreten. Auch erkennt man deutlich, dass die Ausprägungen der Maxima mit größer werdender Abweichung von $\tau = 0$ geringer ausfallen. Aus diesem Grund wird der Verschiebungsfaktor hier und im Folgenden $\tau = 0$ gewählt. Dadurch fällt in dem Plot die τ -Abhängigkeit weg, wir betrachten im Folgenden zweidimensionale Darstellungen.

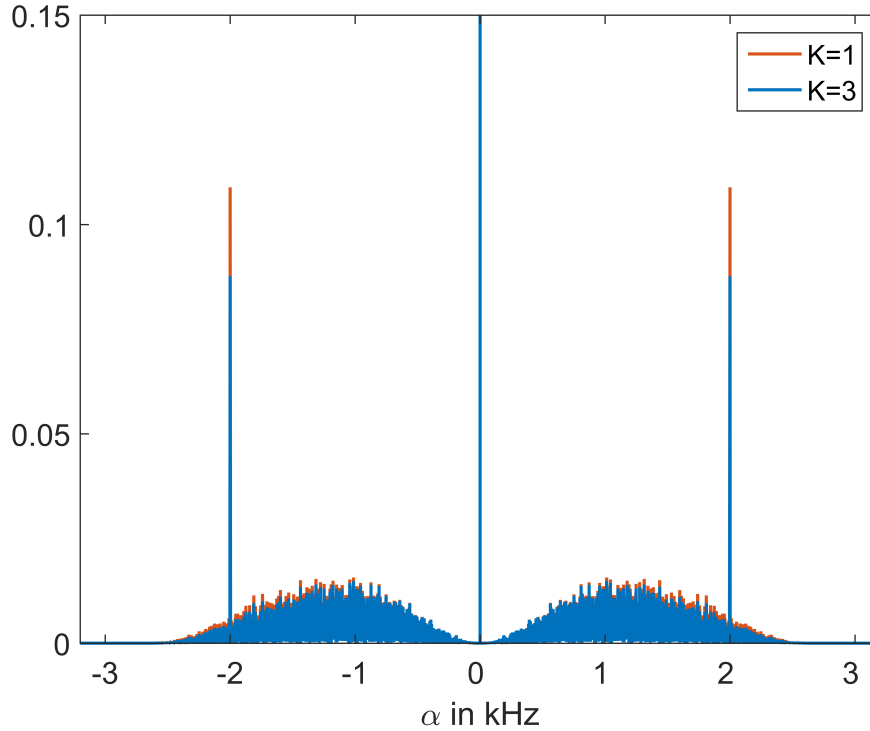


Abbildung 4.3.: Spektrum der zyklischen Momente bei $N = 1 \times 10^5$ und $K \in \{1, 3\}$

Es soll nun auf den Einfluss des Faktors K eingegangen werden. Zur Verdeutlichung ist in Abb. 4.3 ein Ausschnitt des vergrößerten Spektrums für $K = 1$ und $K = 3$ dargestellt. Die Grenzen der X- und Y-Achse in Abb. 4.3 wurden so gewählt, dass man die Nebenmaxima noch erkennen kann. Das Hauptmaximum ist in der Mitte noch zu sehen, ist jedoch nach oben abgeschnitten. Der blaue Plot entspricht dem mittleren Spektrum in Abb. 4.2 bei $\tau = 0$. Er verdeckt den orangen Plot, bei dem $K = 1$ gilt. Bei $K = 1$ fallen die Nebenmaxima wesentlich deutlicher aus. Da eine deutliche Ausprägung von Vorteil ist, wählen wir von nun an $K = 1$.

Wir untersuchen nun die Spektren von $\mu_{\alpha}^{(r,s)}$ bei verschiedenen Ordnungen $r = 2, 4, 8$ und $s = 0$ unter Verwendung verschiedener Modulationsarten. In Abb. 4.4 sind die zyklischen Momente der betreffenden Modulationsarten dargestellt.

Im Fall von Quadrature Phase-Shift Keying (QPSK) erkennt man, dass eine minimale Ordnung von vier nötig ist, dass sich Spektrallinien ausbilden. Erhöht man die Ordnung auf acht, so sind die Linien immer noch zu erkennen, jedoch stärker von Rauschen überlagert,

was eine eindeutige Identifikation erschwert. Die optimale Wahl der Ordnung ist in diesem Fall also vier. Entsprechendes gilt auch bei den anderen Modulationsarten.

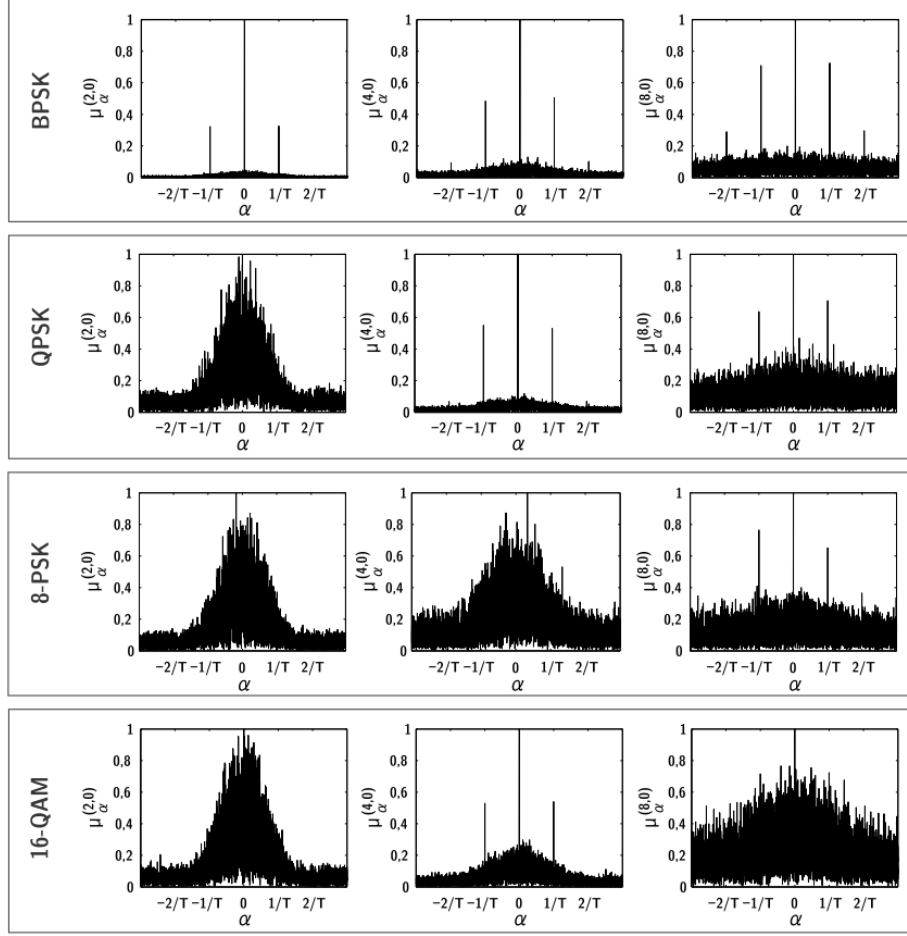


Abbildung 4.4.: Zyklische Momente der Ordnung $r = 2, 4, 8$ und $s = 0$ bei unterschiedlichen Momenten [Mü14, S. 86]

Da die Modulationsart am Empfänger nicht bekannt ist, wird das verwendete Spektrum mithilfe eines Gütekriteriums ausgewählt. Wir wählen das Spektrum, bei dem der Scheitelfaktor, also das Verhältnis von Maximalwert zum Mittelwert, am größten ist.

4.2. Symbolratenschätzung

Die Symbolrate gibt an, mit wie vielen Samples pro Symbol das Signal abgetastet wird. Die Rate ist am Empfänger nicht bekannt, da physikalisch ein zeit-kontinuierliches Signal übertragen wird. Bei der Modulation am Sender wird eine unbekannte Symbolrate verwendet und auch unterschiedliche Samplerraten von Sender und Empfänger beeinflussen die Symbolrate.

Die Schätzung der Symbolrate erfolgt auf ähnliche Weise, wie die Frequenzschätzung. So wird auch die gleiche Ordnung des zyklischen Moments gewählt. Der Abstand zwischen den Nebenpeaks und dem Hauptpeak der zyklischen Momente in Abb. 4.4 entspricht genau der Symbolrate. Auf Implementierungsaspekte zur Frequenz- und Symbolratenschätzung wird in Abschnitt 5.3.1 eingegangen.

4.3. Zeitsynchronisation

Es liegt keine ideale Synchronisation zwischen Sender und Empfänger vor. Daher ist die Bestimmung des optimalen Abtastzeitpunkts nötig, um Intersymbolinterferenz (ISI) zu reduzieren und mit der Symbolenergie das Signal-to-Noise-Ratio (SNR) zu maximieren.

Für die Zeitsynchronisation wird ein Phasenregelkreis zweiter Ordnung angewandt. Die Grundlagen für das genutzte Verfahren werden in [HR01] beschrieben. Bei diesem Ansatz ist keine Rückkopplung zum analog-to-digital converter (ADC) nötig, die gesamte digitale Datenverarbeitung (Digital Signal Processing (DSP)) findet mit fester Abtastrate r_s statt.

Es werden M Filter der Form des Mached-Filters erstellt, jeder mit einer anderen Phase des Filters. Des weiteren werden M Filter mit den jeweiligen Differenzialquotienten dieser Filter angelegt. Man möchte nun den Filter mit der Phase wählen, bei dem das Signal zum bestmöglichen Zeitpunkt abgetastet wird.

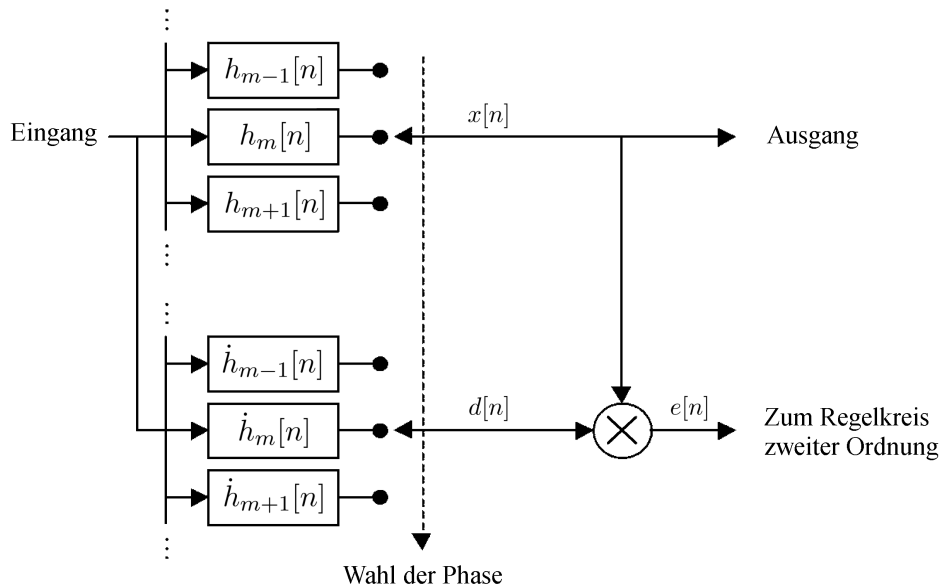


Abbildung 4.5.: Aufbau und Funktionsweise der Polyphasen Synchronisation [HR01]

Zur Verdeutlichung ist in Abb. 4.5 ein Blockschaltbild dargestellt. Die aktuelle Phase beträgt hier m . Man sieht, dass aus den M Filtern jenes mit der Impulsantwort $h_m[n]$ gewählt wird. Von den Filtern mit den Ableitungen wird jenes mit der Impulsantwort $\dot{h}_m[n]$

ausgewählt. Außerdem sind jeweils die Filter vor und nach dem gewählten abgebildet.

Welches Filters verbunden wird, also die Wahl der Phase, wird von einem äußeren Regelkreis kontrolliert. Das Fehlersignal $e[n]$ kann als Phase des äußeren Regelkreises interpretiert werden und wird auf null geregelt. Dieses Fehlersignal ist gegeben durch

$$e[n] = \frac{\Re(x[n]) \cdot \Re(d[n]) + \Im(x[n]) \cdot \Im(d[n])}{2}, \quad (4.2)$$

wobei $x[n]$ die Ausgabe der Filter und $d[n]$ die Ausgabe der Differenzial-Filter ist. Der Fehler berechnet sich durch Mittelung der jeweils multiplizierten Imaginär- und Realteile der Filterausgänge.

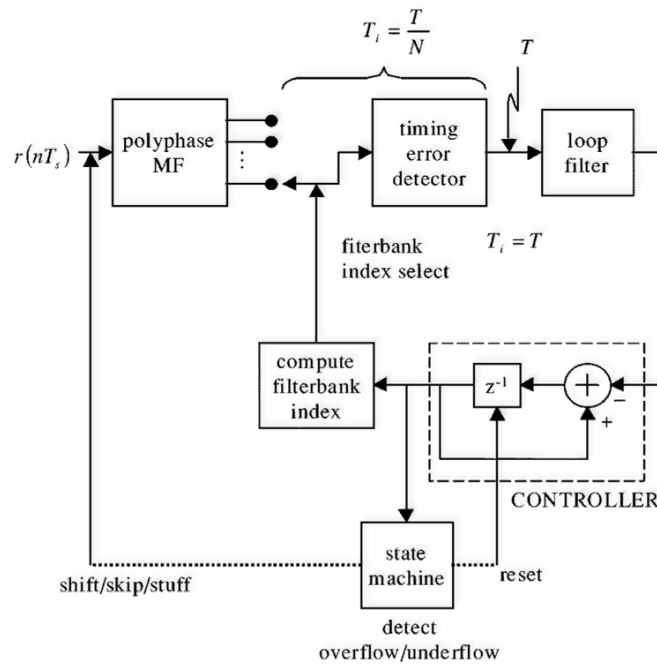


Abbildung 4.6.: Äußerer Regelkreis der Polyphasen Synchronisation [HR01]

Das in Abb. 4.6 aufgezeichnete Strukturbild des äußeren Regelkreises dient zur Verdeutlichung der groben Funktionsweise. Der "Controller" im Schaltbild stellt ein Register dar, von dessen Inhalt sich auf den Index m des Filters schließen lässt. Für den Fall, dass dabei der optimale Abtastzeitpunkt getroffen wird und das Fehlersignal null wird, bleibt der Index gleich.

Sobald sich eine Abweichung davon ergibt, ändert sich der Index in die entsprechende Richtung, um den Fehler auf null zu regeln. Läuft der Index außerhalb des erlaubten Bereichs, zeigt also zum Beispiel hinter das letzte Filter, liegt ein Überlauf vor ("overflow"). Der Index muss in den Grenzen gehalten werden, entsprechend wird M vom Index abgezogen. Korrespondierend wird M zum Index hinzu addiert, falls mit einer Unterschreitung des Bereichs ein Unterlauf vorliegt ("underflow"). Die "state machine" in Abb. 4.6 erkennt Über- und

Unterläufe und gewährleistet eine korrekte Funktion. Dazu gehört auch das Überspringen von Samples am Filtereingang ("skip") beim Überlauf, respektive ein erneutes Verwenden ("stuff") im Falle eines Unterlaufs.

Der Zeitoffset kann nur bis zu einer gewissen Auflösung ausgeglichen werden. Diese ergibt sich aus der Anzahl der verwendeten Filter. Stellt man sich einen Einheitskreis vor, wird dieser in diese Anzahl von Abschnitten aufgeteilt. Dadurch ergibt sich die Phasenaufösung $\Delta\varphi$. Um eine brauchbare Auflösung zu erhalten, sind meistens 32 Filter ausreichend.

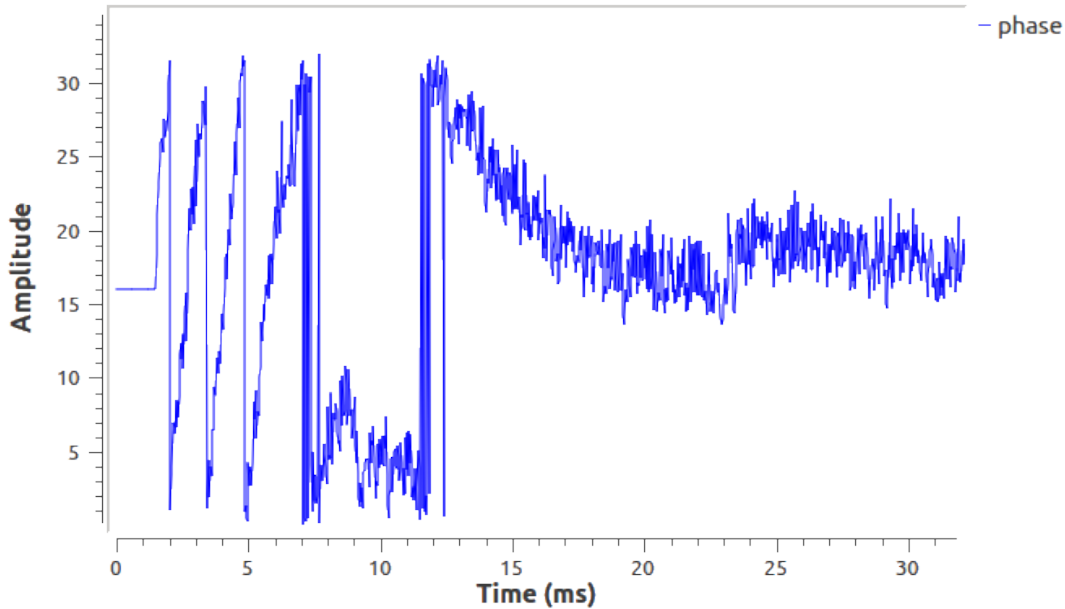


Abbildung 4.7.: Zeitsynchronisation: Verlauf der Phase beim Einschwingen

Der Graph in Abb. 4.7 zeigt die Phase des äußeren Regelkreises beim Einschwingvorgang über der Zeit. Die Phase entspricht dem Wert, der in dem Register des "Controllers" aus Abb. 4.6 gespeichert ist. Dessen quantisierte Form gibt den Index des verwendeten Filters an. Die Werte variieren zwischen 1 und 32, da wir $M = 32$ phasenverschobene Filter verwenden. Am Anfang beträgt die Phase 16 und nimmt ab ca. 1,5ms stetig zu, da das Fehlersignal offenbar ungleich null ist. Wir beobachten bei etwa 2ms einen Überlauf. Damit der Index nicht aus dem zulässigen Bereich läuft, wird M von der Phase subtrahiert und ein Sample am Eingang übersprungen. Dies wiederholt sich nun bis $t \approx 5$ ms, hier trifft der Regelkreis die Phase mit dem momentan korrekten Abtastzeitpunkt und pendelt sich schließlich auf diesen ein.

4.4. Phasensynchronisation

Wurden alle vorhergegangenen Korrekturen erfolgreich durchgeführt, so liegt das Konstellationsdiagramm des Signals um einen unbekannten Phasenversatz gedreht in der IQ Ebene.

Zum Ausgleich des Phasenoffsets wird der Phasenschätzer [MdJ94]

$$\hat{\varphi}_n = \frac{1}{r} \arg \left(\mu_i^{(r,0)} \sum_{k=1}^N \hat{s}_n[k]^r \right) \quad (4.3)$$

verwendet. Die Werte r und $\mu_i^{(r,0)}$ werden vor der Berechnung festgelegt und hängen vom verwendeten Modulationsverfahren ab. Die zugehörigen Größen sind in Tabelle 4.1 aufgelistet.

Modulationsart	r	$\mu_i^{(r,0)}$
BPSK	2	1
QPSK	4	1
8-PSK	8	1
m-PSK	m	1
16-QAM	4	-0,68
m-QAM	4	$\mu^{(4,0)}$

Tabelle 4.1.: Parameter des Phasenschätzers bei verschiedenen Modulationsarten [Mü14, S. 72]

Die geschätzte Phase kann nun zur Korrektur verwendet werden:

$$\bar{s}_n[k] = e^{-j\hat{\varphi}_n} \hat{s}_n[k]^r \quad (4.4)$$

Da sich die Parameter zur Phasenschätzung ändern, muss die Schätzung für jede Hypothese, also jedes Modulationsverfahren separat erfolgen. Auf die Anwendung der Phasenschätzung wird in 5.3.3 näher eingegangen.

In Abb. 4.8 ist der quadratische Mittelwert (QMW) des Phasenschätzfehlers in Abhängigkeit des SNR aufgetragen. Es stellt sich heraus, dass die Schätzung für BPSK am robustesten funktioniert, der maximale Fehler bei $\text{SNR} = -15\text{dB}$ beträgt etwa $0,12\text{rad}$, also weniger als 7° . Im Fall von QPSK wird der Schwellwert von $0,12\text{rad}$ bei einem $\text{SNR} \leq -9\text{dB}$ unterschritten und bei 16-Quadrature Amplitude Modulation (QAM) für $\text{SNR} \leq -7\text{dB}$. Bei 8-Phase-Shift Keying (PSK) hingegen wird die Schätzung schon für $\text{SNR} \leq -3\text{dB}$ schlechter als diese Schwelle, der Fehler wird jedoch nicht größer als etwa $0,23\text{rad}$.

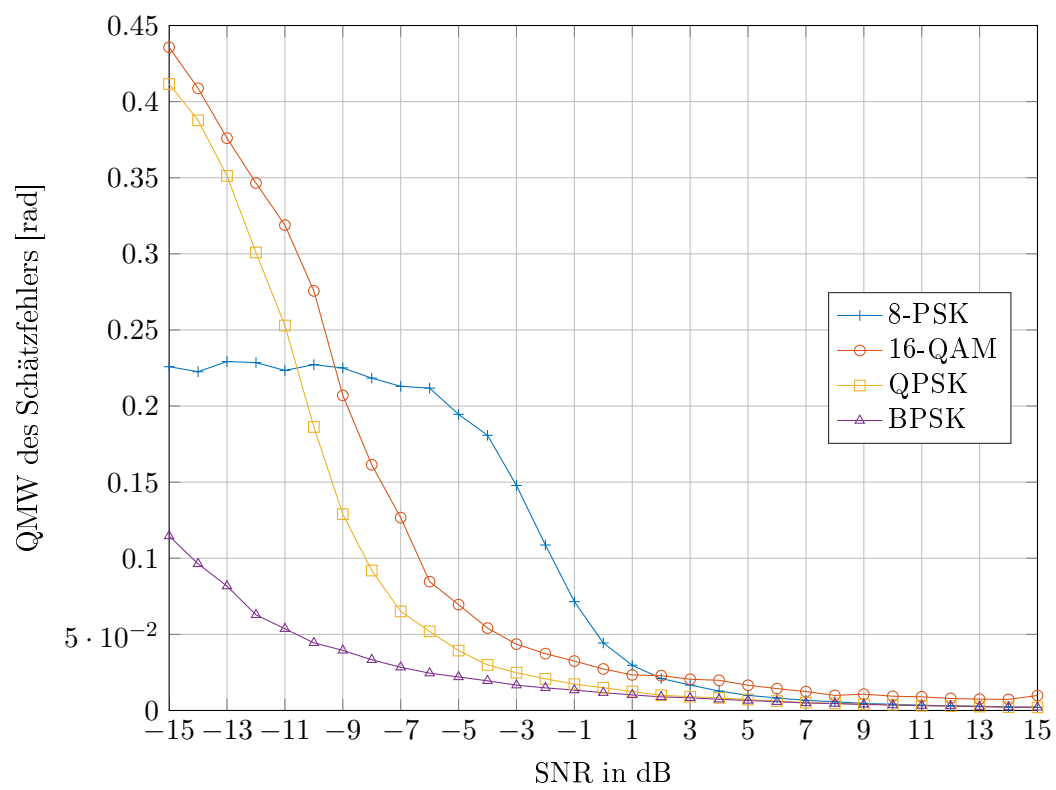


Abbildung 4.8.: Phasenschätzfehler in Abhängigkeit des SNR mit $N = 500$

5. Implementierung

In diesem Kapitel geht es um die Umsetzung von Empfänger und Klassifikator. Zuerst wird das Programm GNU Radio vorgestellt, mit dem Simulation und Implementierung realisiert sind. Dabei wird anhand von Beispielen die Funktionsweise und der Aufbau einer Simulation in Form eines Flowgraphs erklärt. Auf dieser Grundlage soll eine Übersicht über den erstellten Flowgraph mit der Simulation gegeben werden. Dabei wird kurz auf die unterschiedlichen Einheiten im Flussbild eingegangen, um ein grobes Verständnis zu erlangen. In einem nächsten Schritt wird dann detaillierter auf die Implementierung der einzelnen Elemente eingegangen.

5.1. GNU Radio

GNU Radio ist ein Werkzeug zur baukastenartigen Entwicklung von Software-Defined Radio (SDR). Es ist möglich, verschiedenste Modulationsarten oder Filter in Form von Verarbeitungsblöcken zu implementieren und sich Signale auf beliebige Art anzeigen zu lassen. Die Organisation der Signale wird durch sogenannte Flowgraphs realisiert. Um den Fluss der Signale zu veranschaulichen, können Blöcke mit Hilfe der grafischen Oberfläche, dem GNU Radio Companion (GRC), angeordnet und verbunden werden. Zur einfachen Visualisierung von Signalen stehen Blöcke mit einem Graphical User Interface (GUI) zur Verfügung [GR14].

5.1.1. Beispiel Flowgraph

Ein einfacher Flowgraph besteht klassischerweise aus einer Quelle, der Signalverarbeitung und einer Senke. Die Einheiten haben einen oder mehrere Ports, welche miteinander verbunden werden, um einen Signalfluss herzustellen. Die Signale können von verschiedenen Datentypen sein, oft sind es komplexe Gleitkommazahlen.

Es besteht die Möglichkeit, alle Signale virtuell zu erzeugen und den Flowgraph durchlaufen zu lassen. Ein Beispiel ist in Abbildung (Abb.) 5.1 zu sehen. Im Block "Signal Source" wird ein komplexes Kosinussignal mit der Frequenz 1kHz und der Amplitude 1 erzeugt. Die Signalverarbeitung besteht hier lediglich aus dem Block "Multiply Const", bei dem das Signal mit dem Faktor 0,5 multipliziert wird.

Der Block "QT GUI Time Sink" dient zur Visualisierung des Signals, die Ausgabe des ausgeführten Programms ist in Abb. 5.2 zu sehen.

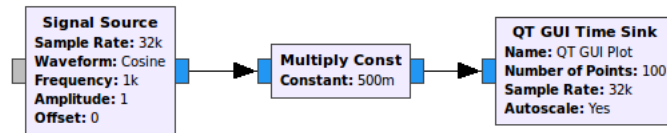


Abbildung 5.1.: Einfacher Beispiel-Flowgraph

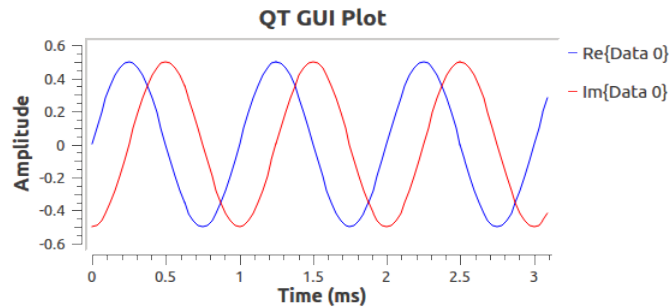


Abbildung 5.2.: Ausgabe des Beispiel-Flowgraphs

Ein großer Vorteil von GNU Radio besteht darin, dass sich Hardware, wie Universal Software Radio Peripheral (USRP), zur Übertragung durch die Luft ganz einfach als Block in den Flowgraph einfügen lässt. Möchte man ein Signal aus der Luft empfangen, geschieht das durch Austauschen des Blocks "Signal Source" in Abb. 5.1 mit einem Block, welcher die Daten von der Hardware akquiriert und in den Flowgraph einbringt. Neben der Nutzung der vielen bereits vorhandenen Blöcke lassen sich auch eigene Blöcke entwickeln und mit wenig Aufwand verwenden. Als Programmiersprachen sind C++ und Python möglich.

5.1.2. Stream Tags

Mit GNU Radio ist es möglich, neben dem eigentlichen Datenstrom, der zwischen den Blöcken fließt, zusätzliche Informationen synchron zu übertragen. Dies ist zum Beispiel mit sogenannten Stream Tags möglich. Dabei handelt es sich um eine Datenstruktur, welche einem bestimmten Sample im Datenstrom zugeordnet ist. Ein Stream Tag besteht zum einen aus einem Key, der bezeichnend für die Information ist, und zum anderen aus dem Wert, der die eigentliche Information darstellt. Der Datentyp kann verschiedener Art sein, wie zum Beispiel ein String oder eine Gleitkommazahl. Ein Stream Tag eignet sich, um Informationen, die bestimmte Stellen oder Bereiche des Signals betreffen, zwischen verschiedenen Blöcken auszutauschen.

5.1.3. Vector-Optimized Library of Kernels (VOLK)

Die Vector-Optimized Library of Kernels (VOLK) dient zur einfachen Verwendung des Single Instruction Multiple Data (SIMD) Befehlssatzes des Prozessors. Mithilfe von SIMD-

Befehlen lassen sich schnell viele Daten parallel verarbeiten, was einen enormen Geschwindigkeitsvorteil mit sich bringt. Diese Befehle sind so effizient, weil sie sehr hardwarenah ablaufen. Dies bringt jedoch auch mit sich, dass die Befehle mitunter für verschiedene Prozessoren unterschiedlich sind.

VOLK besteht aus einer Reihe von Funktionen, die auf einen Vektor angewandt werden können. VOLK weiß, welche SIMD-Befehle beim gegenwärtigen Prozessor verfügbar sind. Auf dieser Basis wird zur Laufzeit entschieden, welche Routine zur Berechnung der Funktion verwendet wird. Anstatt einer gewöhnlichen Schleife, die jedes Element des Datenvektors abarbeitet, wird die Berechnung nun also parallelisiert auf den Datenvektor angewandt.

Um die Parallelisierbarkeit zu ermöglichen, ist eine Ausrichtung der zu verarbeitenden Daten im Speicher nötig. Diese Anforderung hängt vom System ab, oft weist die Architektur eine Länge von 32 Bit auf. Sollten die einzelnen Samples der Datenströme nun kürzer als diese Länge sein, so kann es passieren, dass beim Aufruf einer VOLK-Funktion ein Array übergeben wird, welches nicht an dieser Speichergröße ausgerichtet ist. In diesem Fall muss zusätzlich entsprechend ausgerichteter Speicher für die Daten vergeben werden. Eine performante Berechnung ist nur möglich, falls dieser Aspekt beachtet wird.

Beispielimplementierung

Es soll nun nur anhand eines kurzen Beispiels gezeigt werden, wie die Implementierung von VOLK-Funktionen aussieht. Es soll ein Verarbeitungsblock des Signals quadriert werden. Die Ausgangsdaten sind in einem *n_samples* langen Array *samples* gespeichert. Das Ergebnis soll in einem Array mit derselben Länge mit dem Namen *samples_squared* abgespeichert werden. Klassisch würde die Quadrierung für jedes Sample nacheinander in einer Schleife ablaufen:

```
1 for (int i=0; i<n_samples; i++) {  
2     samples_squared[i]=pow(samples[i],2);  
3 }
```

Für eine parallelisierte Datenverarbeitung soll diese Schleife nun durch den Aufruf einer VOLK-Funktion ersetzt werden. Sofern der Kernel eingebunden wurde, geschieht dies durch die folgende Codezeile:

```
1 volk_32fc_s32f_power_32fc(samples_squared, samples, 2, n_samples);
```

Die Namensgebung der Funktion erfolgt nach dem Schema [VOL15]:

`volk_(Eingangsgroßen)_[Operation]_(Ausgangsgroßen)`

Die durchgeführte Operation ist eine Potenzierung (**power**). Die Bezeichnung **32fc** steht für komplexe float-Vektoren der Länge 32-Bit. Dies sind Ein-, sowie Ausgangsdatenströme. Des Weiteren wird **s32fc** eingegeben, was einem reellen Skalar der Länge 32-Bit entspricht. Die übergebene 2 setzt den Exponenten auf zwei, wodurch die Quadrierung erreicht wird.

5.2. Übersicht über den Flowgraph der Simulation

Hier wird eine Übersicht über den von mir erstellten Flowgraph gegeben. Dabei werden die einzelnen Blöcke erklärt und es wird kurz auf deren Funktion und die Realisierung eingegangen, tiefer gehende Erläuterungen sind im Abschnitt 5.3 zu finden.

5.2.1. Sender und Kanal

In Abb. 5.3 ist der erste Teil des Flussbildes zu sehen, der den Sender und Kanal beinhaltet. Hier wird die Erzeugung und Modulation eines Signals sowie der Kanaleinfluss simuliert. Es wird nun nacheinander auf die verschiedenen Stufen von Blöcken eingegangen.

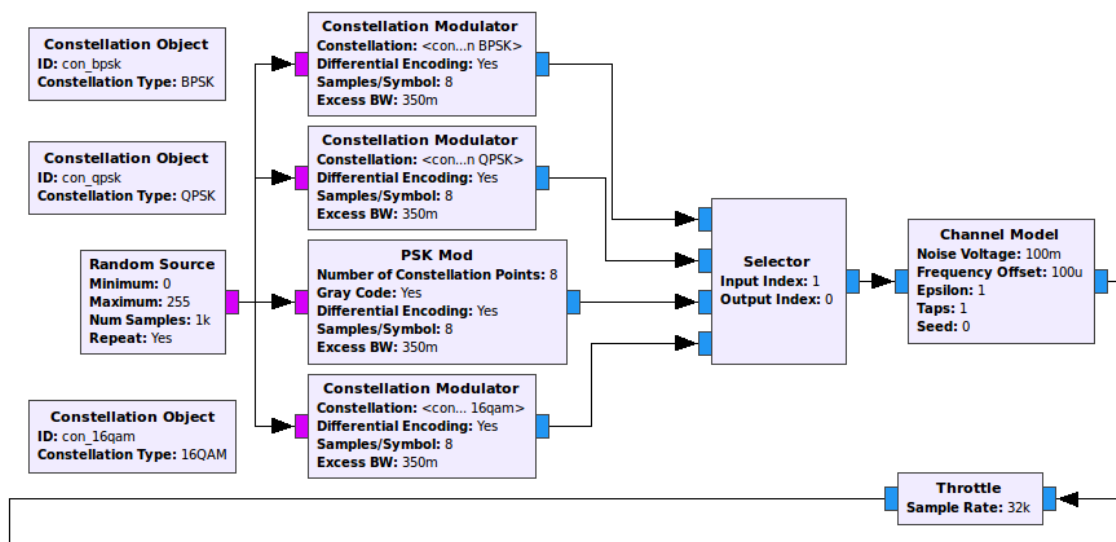


Abbildung 5.3.: Flowgraph des Senders und des Kanals

- **Signalerzeugung:** Der Block *Random Source* generiert eine Folge von Bytes, ein Byte hat einen positiven Wertebereich von 0 bis 255. Diese Werte sind für "Minimum" und "Maximum" eingetragen. Da die Quelle endlos weiterlaufen soll, ist der Parameter "Repeat" auf "Yes" gesetzt. Somit wird die Hervorbringung wiederholt, nachdem 1000 Samples ("Num Samples") erzeugt wurden.
- **Modulation:** Der erzeugte Datenstrom wandert nun parallel in vier verschiedene Blöcke, jeder für eine andere Modulation. Drei der Blöcke sind *Constellation Modulator*, der Bitstrom wird also mittels In-Phase-&-Quadrature-Verfahren (I&Q-Verfahren) auf die Symbole in der I/Q-Ebene projiziert. Die Lage dieser Symbole ist abhängig von dem Modulationsverfahren und wird im *Constellation Object* spezifiziert. Der vierte Block ("PSK Mod") ist für die 8-Phase-Shift Keying (PSK) Modulation zuständig. Die Gestalt der Konstellationsdiagramme der Modulationsarten sind in Abb. 2.3 dargestellt.

Die Rate "Samples/Symbol" entspricht der Symbolrate (vgl. Abschnitt 4.2), sie wurde auf acht festgelegt. Das Sendefilter ist ein Root-Raised-Cosine (RRC) Filter, dessen Roll-off-Faktor r mit dem Parameter "Excess BW" auf 0,35 festgelegt wird (vgl. Abschnitt 2.2).

- **Auswahl des Modulationsverfahrens:** Die Aufgabe des *Selector*-Blockes ist es, einen bestimmten Eingangsport mit dem Ausgangsport zu verbinden. Der gewünschte Port wird mit "Input Index" spezifiziert, wobei die Indizierung bei null beginnt. Die Zuordnung erfolgt intuitiv wie in Tabelle 5.1 aufgetragen.

"Input Index"	Gewählte Modulation
0	BPSK
1	QPSK
2	8-PSK
3	16-QAM

Tabelle 5.1.: Zuordnung von "Input Index" zu den Modulationen

Da es nur einen Ausgangsport gibt, entfällt die Wahlmöglichkeit und "Output Index" steht immer auf Null. Mit dieser Funktionalität ist es möglich, zur Laufzeit eines von mehreren Modulationsverfahren auszuwählen, welches gesendet werden soll.

- **Kanal Modell:** Wird ein Signal übertragen, so gibt es viele Einflüsse, die darauf wirken und es verändern. Gründe dafür sind meist physikalischer Natur, die den Übertragungskanal betreffen. Die Gesamtheit dieser Einflüsse soll mit dem *Channel Model* modelliert werden.

Ein unvermeidbarer Störfaktor stellt beispielsweise thermisches Rauschen dar. Es entsteht in Verstärkern durch die thermische Teilchenbewegung und wird durch Additive White Gaussian Noise (AWGN)-Rauschen modelliert. Die "Noise Voltage" gibt die Stärke des Rauschens an. Bei gegebenem Signal-to-Noise-Ratio (SNR) kann man den zugehörigen Wert U_N bei einer auf eins normierten Sendeleistung nach der Formel

$$U_N = \sqrt{\left(10^{\frac{\text{SNR}}{10}}\right)^{-1}} \quad (5.1)$$

berechnen. Außerdem ergibt sich ein Frequenzoffset ("Frequency Offset") und ein Zeitoffset ("Epsilon"). Für "Epsilon" gleich eins ist kein Zeitoffset vorhanden. Die Ursachen und deren Einflüsse wurden in Abschnitt 2.2.2 ausführlich behandelt. In das Feld "Taps" wird 1 eingetragen, da wir davon ausgehen, dass keine Mehrwegausbreitung vorliegt, sondern nur ein dominierender Übertragungspfad besteht.

- **Throttle:** Dieser Block wird benötigt, um die vollständige Auslastung der CPU bei Simulationen zu verhindern. Er muss sich immer einmal irgendwo im Flowgraph befinden, sofern keine Hardware verwendet wird, welche dann ihrerseits den Takt kontrollieren würde.

5.2.2. Empfänger

Der zweite Teil des Flowgraphs beinhaltet den Empfänger mit dem Klassifikator. Es wird nun zu jedem Block erklärt, wofür er benötigt wird. Der Code der Blöcke *Frequency Correction*, *Polyphase Clock Sync* und *Modulation Classifier* wurde im Rahmen dieser Arbeit implementiert, wobei der Code von *Polyphase Clock Sync* aus einem bestehenden Block übernommen und an meine Anforderungen angepasst wurde.

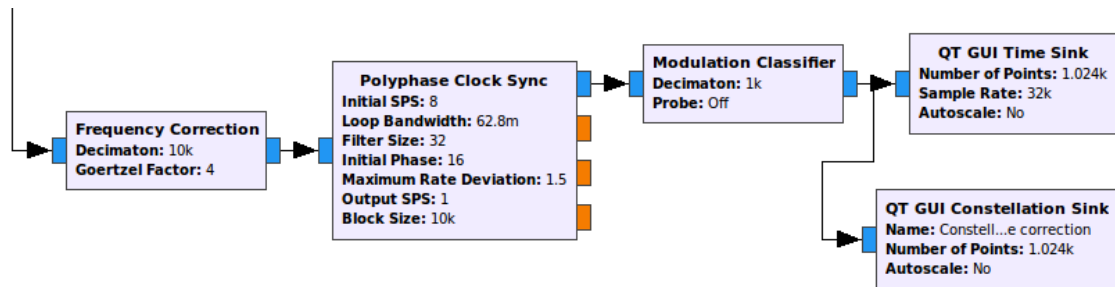


Abbildung 5.4.: Flowgraph des Empfängers

- **Frequenzkorrektur und Symbolratenschätzung:** Zuerst wird der Frequenzoffset des empfangenen Signals im Block *Frequency Correction* berechnet und ausgeglichen. Zusätzlich wird die Symbolrate geschätzt und als Stream Tag (Vergleiche Abschnitt 5.1.2) an den nächsten Block weitergereicht. Auf das verwendete Verfahren wurde in Abschnitt 4.1 eingegangen.

Der Eingangsstrom wird blockweise verarbeitet. Es wird immer eine bestimmte Anzahl ("Decimation") Samples herangezogen, um einen Frequenzoffset zu berechnen, welcher dann auf alle verwendeten Samples angewandt wird. Mit dem Setzen des Parameters "Goertzel Faktor", kann die Frequenzauflösung erhöht werden.

- **Zeitsynchronisation und Matched-Filterung:** Die Zeitsynchronisation findet im Block *Polyphase Clock Sync* zusammen mit der Matched-Filterung statt. Auf diesen Sachverhalt wurde in Abschnitt 4.3 eingegangen. Zur Berechnung wird die Symbolrate benötigt. Der Parameter "Initial SPS" (SPS steht für Samples per Symbol) bestimmt den zur Initialisierung verwendeten Wert. Die Anzahl der Filter "Filter Size" wird auf 32 festgelegt.

In Abschnitt 5.3.2 wird auf die Implementierung eingegangen. Sobald der Signalfluss beginnt, wird die vom Block *Frequency Correction* berechnete und per Stream Tag übermittelte Symbolrate verwendet und laufend aktualisiert. Die Ausgabe der Symbole erfolgt mit der Symbolrate eins.

- **Klassifikation und Phasenkorrektur:** Im Block *Modulation Classifier* findet die Klassifikation auf Grundlage der Kumulanten sowie der Ausgleich des Phasenoffsets statt. Am Ausgang liegt das Signal mit korrigierter Phase an. Dem Signal sind Stream Tags angehängt, welche die Information zur erkannten Modulation enthalten.

Die Demodulation des Datenstroms erfolgt danach auf Grundlage der Information in den Stream Tags, dies wird in dieser Arbeit jedoch nicht behandelt.

Ähnlich wie bei der Frequenzsynchronisation wird der Datenstrom hier auch blockweise bearbeitet. Die Blockgröße lässt sich mit der Variablen "Decimation" festlegen. Mit dem Parameter "Probe" wird festgelegt, ob die detektierten Modulationen gespeichert und für statistische Auswertungen bereitgestellt werden sollen.

- **Grafische Elemente:** Zuletzt sind noch zwei Einheiten zur graphischen Visualisierung des Datenstroms angefügt. Der Block *QT GUI Time Sink* trägt die Amplitude der Real- und Imaginärteile des Signals über der Zeit auf. In diesem Diagramm lassen sich auch Stream Tags ablesen. Die Anzeige von *QT GUI Constellation Sink* umfasst das Konstellationsdiagramm des Signals. Somit kann man die Simulation in Echtzeit mitverfolgen und die Funktion von Empfänger und Klassifikator verifizieren.

5.2.3. Ausgabe der Simulation

Wird der Flowgraph ausgeführt, so wird aufgrund der Blöcke *QT GUI Time Sink* und *QT GUI Constellation Sink* eine grafische Oberfläche erzeugt. Sie ist in Abb. 5.5 dargestellt. Im "Time Plot" werden die Symbole aufgespalten in Real- und Imaginärteil über der Zeit aufgetragen. In diesem Diagramm sind außerdem die angefügten Stream Tags zu sehen. Ein Stream Tag trägt den Key "det_mod" und beinhaltet die Information zu der vom Klassifikator erkannten Modulation. Das andere Stream Tag ist mit "det_sps" gekennzeichnet. Dieses Tag beinhaltet die vom Block *Frequency Correction* erkannte Symbolrate und wird im Block *Polyphase Clock Sync* für die Matched-Filterung benötigt.

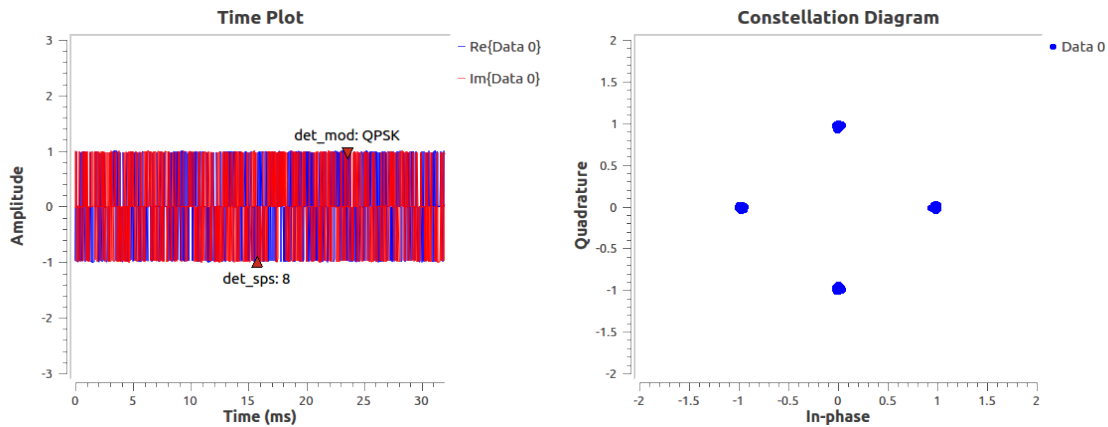


Abbildung 5.5.: Erzeugte Ausgabe beim Starten des Flowgraphs

Die detektierte Modulationsart ist, wie im Stream Tag angegeben, Quadrature Phase-Shift Keying (QPSK). Der Parameter "Input Index" des Blocks *Selector* ist, wie in Abb. 5.3 zu sehen, auf eins gesetzt. Nach Tabelle 5.1 wurde also QPSK gesendet, die Entscheidung des Klassifikators war also korrekt. Überprüfen lässt sich das Ganze auch durch Vergleichen des

"Constellation Diagram" in der Ausgabe (Abb. 5.5) mit den Konstellationsdiagrammen der Modulationen, welche in Abb. 2.3 zu sehen sind.

5.3. Detaillierte Beschreibung der implementierten Elemente

Nachdem nun ein Überblick über den Gesamtaufbau der Simulation gegeben wurde, findet hier eine detaillierte Beschreibung statt. Die Blöcke sind in der Programmiersprache C++ implementiert, um eine gute Performance zu erreichen. Des Weiteren kommen für parallelisierbare Berechnungen, wie zum Beispiel die Multiplikationen von Signalen, Funktionen aus der VOLK (vgl. Abschnitt 5.1.3) zum Einsatz. Im Folgenden wird näher auf die Implementierung der einzelnen Blöcke eingegangen.

5.3.1. Fokus auf Frequenzkorrektur und Symbolratenschätzung

Der Ausgleich des Frequenzoffsets und die Berechnung der Symbolrate ist die Aufgabe des Blockes *Frequency Correction*, siehe Abb. 5.6. Dazu findet das in Abschnitt 4.1 behandelte Verfahren Anwendung. Zusätzlich soll näher auf die Frequenzauflösung des Verfahrens eingegangen werden.

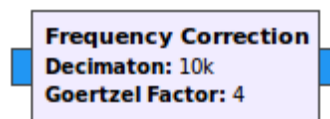


Abbildung 5.6.: Block im Flowgraph zur Frequenzkorrektur und Symbolratenschätzung

Bei der Ermittlung der höheren Momente werden die Zwischenergebnisse der niedrigeren Momente wiederverwendet. Zur effizienten Berechnung des komplexen Spektrums wird ein Fast Fourier Transform (FFT) Algorithmus verwendet. Es folgt die Betragsbildung und Maximumsuche. Wie in Abschnitt 4.1 erläutert, muss zu jedem Moment das Verhältnis aus Maximalwert des Spektrums zum Durchschnitt berechnet werden. Das Spektrum, bei dem dieses Verhältnis am größten ist, wird für die Frequenzschätzung verwendet.

Berechnung von Frequenzoffset und Symbolrate aus dem Spektrum

Wie in Abschnitt 4.1.1 erarbeitet, wählen wir für die Berechnung der zeitabhängigen Momente eine Fensterbreite von $K = 1$ und den Verschiebungsfaktor τ zu null. Diese Wahl bringt den Vorteil mit sich, dass sich die Berechnung zu einer Multiplikation vereinfacht. Die Berechnung kann also erheblich beschleunigt werden, da sich die Multiplikation von Signalen als performante VOLK-Funktion implementieren lässt.

Wir betrachten nun ein Signal mit einem Frequenzoffset von $f_{\text{offs}} = 1\text{kHz}$. Dieser Offset ist für real auftretende Szenarien zwar hoch, soll hier aber zur Demonstration dienen.

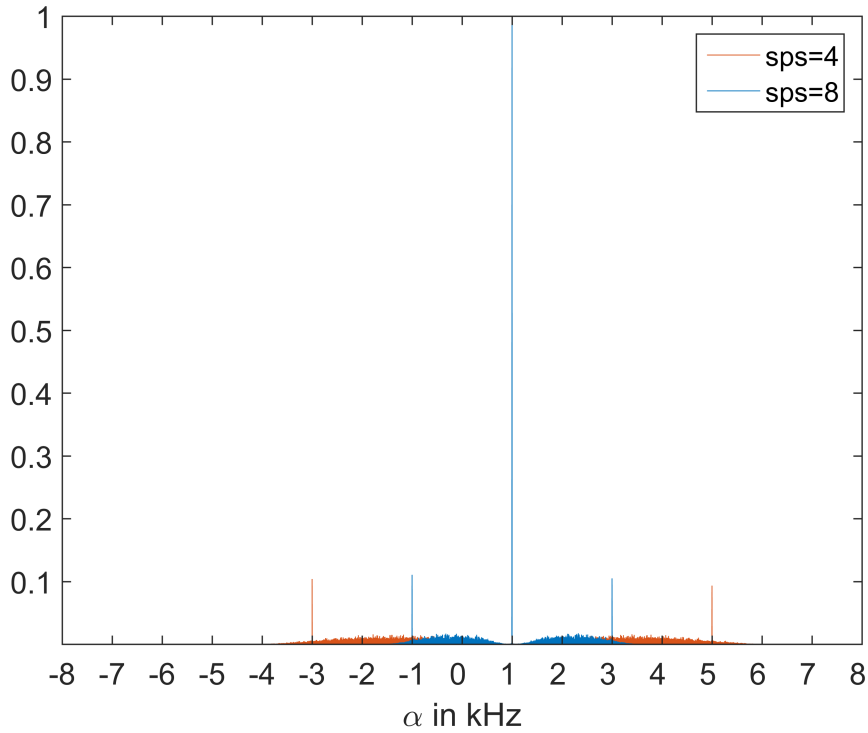


Abbildung 5.7.: Normierte Spektren mit verschiedenen Symbolraten und $f_{\text{offs}} = 1\text{kHz}$

In Abb. 5.7 sind die Spektren von Binary Amplitude Modulation (BPSK) Signalen mit den Symbolraten (samples per symbol (sps)) $r_{\text{symb}} = 4$ und $r_{\text{symb}} = 8$ im rauschfreien Fall abgebildet. Es gilt $N = 1 \times 10^5$. Die Hauptmaxima der beiden Spektren befinden sich an der Stelle $\alpha = 1\text{kHz}$, wobei die orange Kurve von der Blauen verdeckt ist. Die Frequenz an dieser Stelle entspricht dem Frequenzoffset.

Die Nebenmaxima der beiden Graphen sind repräsentativ für die jeweilige Symbolrate. Bei gegebenem Abstand von Haupt- zu Nebenmaximum $\Delta\alpha_{\text{sps}}$ berechnet sich die Symbolrate bei BPSK zu

$$r_{\text{symb}} = \frac{r_s}{2 \cdot \Delta\alpha_{\text{sps}}}. \quad (5.2)$$

Somit erhalten wir bei $\Delta\alpha_{\text{sps}} = 2\text{kHz}$ (blau)

$$r_{\text{symb}} = \frac{32\text{kHz}}{2 \cdot 2\text{kHz}} = 8 \quad (5.3)$$

und für $\Delta\alpha_{\text{sps}} = 4\text{kHz}$ (orange)

$$r_{\text{symb}} = \frac{32\text{kHz}}{2 \cdot 4\text{kHz}} = 4. \quad (5.4)$$

Dies entspricht genau den gesendeten Symbolraten, die Schätzung war also zutreffend.

Frequenzauflösung

Es wird nun die Leistungsfähigkeit der Frequenzschätzung untersucht werden. Ein wichtiges Kriterium ist die Frequenzauflösung der Schätzung. Sie ist abhängig von der Modulationsart, denn je nachdem werden andere Spektren verwendet. Wird das zyklische Moment $\mu_\alpha^{(r,s)}$ (wobei $s = 0$) verwendet, so ist die Auflösung bei einer Sample Rate von r_s und der Verwendung von N Samples gegeben durch

$$f_{\text{res}} = \frac{r_s}{r \cdot N} . \quad (5.5)$$

Für die verschiedenen Modulationsarten erwarten wir nach Abb. 4.4 folgende Werte für die Ordnung des Moments:

$$r = \begin{cases} 2 & \text{bei BPSK} \\ 4 & \text{bei QPSK und 16-QAM} \\ 8 & \text{bei 8-PSK} \end{cases} \quad (5.6)$$

Zur Veranschaulichung ist in Abb. 5.8 der quadratische Mittelwert (QMW) des Schätzfehlers in Abhängigkeit eines vorhandenen Frequenzoffsets aufgetragen. Die Anzahl der zur Mittelung herangezogenen Datensätze beträgt 1000. Zur Durchführung der Simulationen wird nun und im Folgenden ein SNR von 3dB und QPSK verwendet, daher gilt $r = 4$. Zum Vergleich werden verschiedene Sampleanzahlen N_1 und N_2 gewählt. Nach Gleichung (5.5) ergibt sich für $N_1 = 1 \times 10^5$ eine Auflösung von

$$f_{\text{res},1} = \frac{r_s}{4 \cdot 1 \times 10^5} = r_s \cdot 2,5 \times 10^{-5} \quad (5.7)$$

und für $N_2 = 5 \times 10^5$

$$f_{\text{res},2} = \frac{r_s}{4 \cdot 5 \times 10^5} = r_s \cdot 5 \times 10^{-6} . \quad (5.8)$$

Es ist deutlich zu erkennen, dass der Schätzfehler zu null wird, wenn der Frequenzoffset ein Vielfaches der Frequenzauflösung ist, also in das Raster passt. Genau in der Mitte zwischen zwei Rasterpunkten ist der Schätzfehler am größten, er entspricht etwa dem Abstand zum nächsten Rasterpunkt. Bei N_2 ist der QMW des Schätzfehlers bei einem Offset von $f_{\text{offset}} = 0,25 \cdot 10^{-5}$ etwa gleich $0,25 \cdot 10^{-5}$.

Bisher wurde der Fall angenommen, dass die FFT-Länge genau der Anzahl Samples N entspricht. Um ein besseres Ergebnis zu erzielen, kann der Goertzel-Algorithmus verwendet werden. Mit diesem ist es möglich, bei reduziertem Rechenaufwand nur einen bestimmten Ausschnitt des Spektrums zu berechnen. Hat man mit dem FFT-Algorithmus den gesamten spektralen Bereich abgedeckt und ein Maximum gefunden, so kann man mit Hilfe des Goertzel-Algorithmus eine höhere Auflösung wählen und nur den Bereich direkt um dieses Maximum betrachten. Dadurch erhöht sich die Frequenzauflösung auf 5.5 um den Faktor k_g bei nur wenig höherem Rechenaufwand:

$$f_{\text{res}} = \frac{r_s}{r \cdot k_g \cdot N} . \quad (5.9)$$

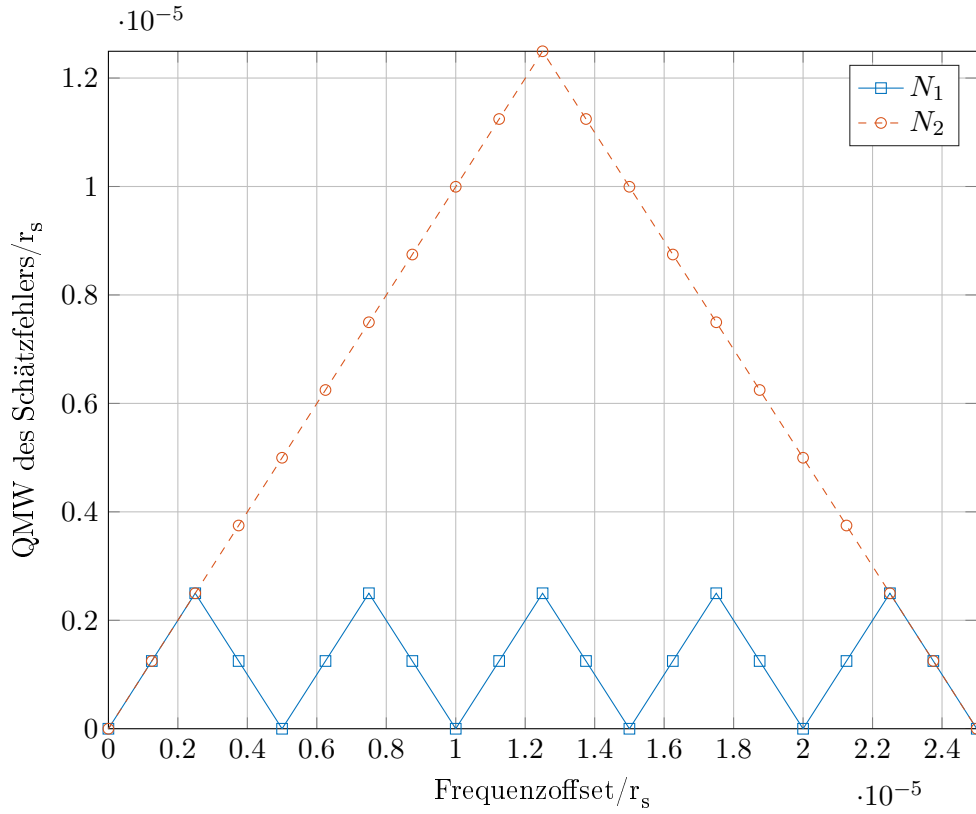


Abbildung 5.8.: QMW des Schätzfehlers in Abhängigkeit des Frequenzoffsets mit $k_g = 1$

In allen vorhergegangenen Betrachtungen galt $k_g = 1$, nun soll die Leistungsfähigkeit bei $k_g > 1$ untersucht werden. Wir führen nun zwei Tests A und B durch. Die gewählten Werte für N und k_g sind in Tabelle 5.2 dargestellt. In den beiden Tests wird jeweils die gleiche Auflösung erwartet, bei Probe 2 ist dazu jeweils $k_g = 2$ und die Anzahl an Samples N im Vergleich zu Probe 1 halbiert.

	Test A		Test B	
	Probe 1	Probe 2	Probe 1	Probe 2
N	2×10^4	1×10^4	4×10^4	2×10^4
k_g	1	2	1	2
f_{res}	$r_s \cdot 1,25 \times 10^{-5}$		$r_s \cdot 6,25 \times 10^{-6}$	

Tabelle 5.2.: Übersicht der gewählten Parameter bei verschiedenen Tests

Die für Test A erwartete Auflösung ergibt sich nach Gleichung (5.9) zu:

$$f_{\text{res,A}} = \frac{r_s}{4 \cdot 1 \cdot 2 \times 10^4} = \frac{r_s}{4 \cdot 2 \cdot 1 \times 10^4} = r_s \cdot 1,25 \times 10^{-5} \quad (5.10)$$

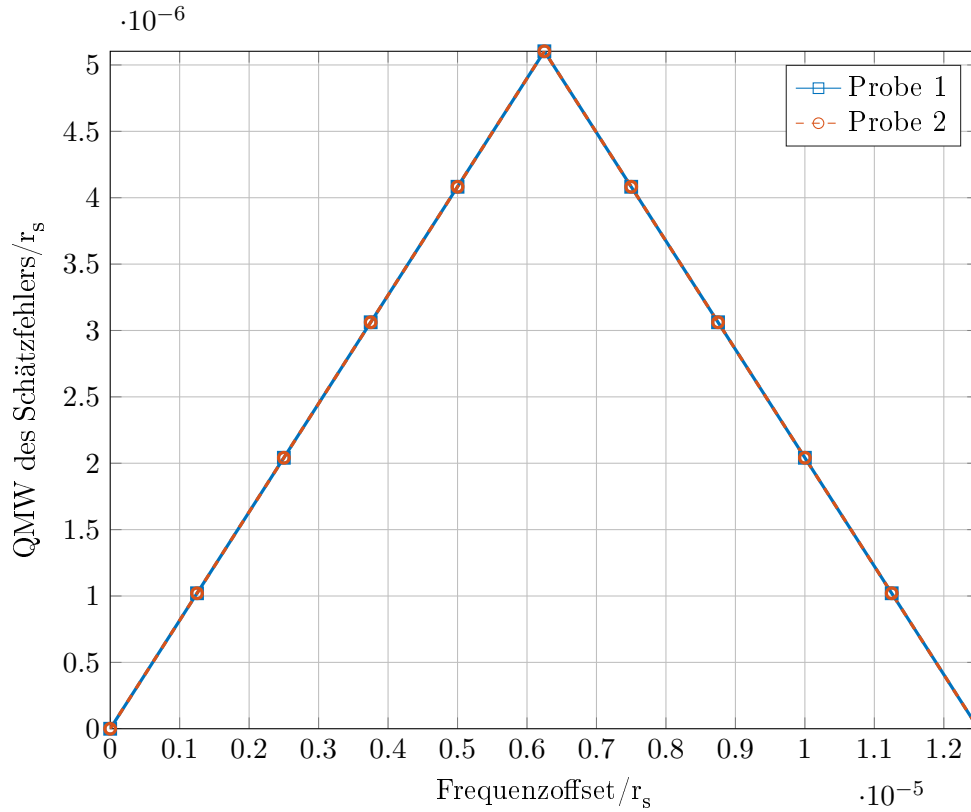


Abbildung 5.9.: Test A: QMW des Schätzfehlers in Abhängigkeit des Frequenzoffsets

In Abb. 5.9 ist der QMW des Schätzfehlers in Abhängigkeit des Frequenzoffsets für Test A aufgetragen. Das Ergebnis ist wie erwartet. Wir erhalten für beide Kurven den gleichen Verlauf mit genau der Auflösung $f_{\text{res,A}}$.

Nach entsprechendem Vorgehen ergibt sich im Fall von Test B:

$$f_{\text{res,B}} = \frac{r_s}{4 \cdot 1 \cdot 4 \times 10^4} = \frac{r_s}{4 \cdot 2 \cdot 2 \times 10^4} = r_s \cdot 6,25 \times 10^{-6} \quad (5.11)$$

Für Test B erhalten wir das in Abb. 5.10 dargestellte Diagramm. Man beachte die im Vergleich zu Abb. 5.9 geänderte x-Achse. Es fällt sofort auf, dass die Kurven bei Test B nun nicht mehr übereinander liegen. Für Probe 1 ergibt sich die erwartete Kurve, vergleichbar mit Test A. Bei Probe 2 jedoch ist die Schätzung bei kleinem Frequenzoffset viel schlechter als erwartet.

Wir stehen nun vor der Frage, warum die Schätzung für $k_g > 1$ bei hohem N nicht mehr gut funktioniert. Die Korrektheit der Implementierung des Goertzel-Algorithmus wurde in Test A gezeigt, hier gibt es keinerlei Probleme. In Test B wurde N verdoppelt, wir verdoppeln damit die Auflösung des Goertzel-Algorithmus. Dessen Implementierung enthält dann eine sehr lange Summe. Aufgrund der numerischen Berechnung treten hier offenbar Rundungsfehler auf.

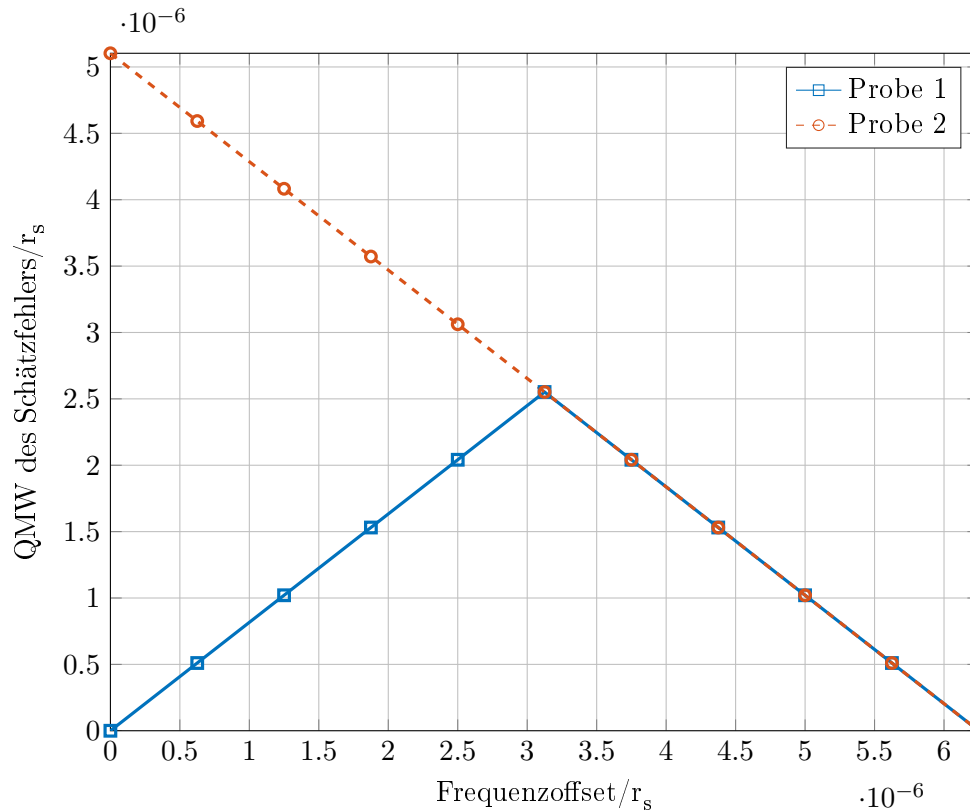


Abbildung 5.10.: Test B: QMW des Schätzfehlers in Abhängigkeit des Frequenzoffsets

Die in Test A erreichte Genauigkeit von $f_{\text{res}} = r_s \cdot 1,25 \times 10^{-5}$ reicht jedoch aus, da die Schätzung für die N verwendeten Samples gut genug ist, um die ausgegebenen Samples weiter zu prozessieren.

5.3.2. Fokus auf die Zeitsynchronisation

Zur Kompensation des Zeitoffsets werden Polyphasen Filter eingesetzt. Das angewandte Prinzip wurde in Abschnitt 4.4 erläutert. Der Code für den von mir verwendeten Block basiert im Wesentlichen auf dem über GNU Radio bereitgestellten Block *Polyphase Clock Sync* (Abb. 5.11). Um den Block meinen Anforderungen anzupassen, wurden Änderungen vorgenommen, welche im folgenden Abschnitt behandelt werden.

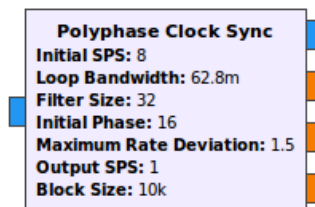


Abbildung 5.11.: Block im Flowgraph zur Zeitsynchronisation

Anpassungen des Zeitsynchronisationsblocks

Der Empfänger soll sich an eine variable Anzahl von Samples pro Symbolen anpassen. Die Voraussetzung für diese Funktionalität ist eine einstellbare Symbolrate im Block *Polyphase Clock Sync*. Vor den Anpassungen gab es lediglich die Möglichkeit, eine Symbolrate zur Initialisierung des Blocks anzugeben. Sie soll der ungefähr zu erwartenden Rate entsprechen, muss jedoch nicht genau mit der tatsächlichen übereinstimmen.

Der Block *Frequency Correction* gibt die Symbolrate vor, die verwendet werden soll. Sie wird als Stream Tag (vgl. Abschnitt 5.1.2) übergeben. Da die Erkennung blockweise abläuft, ist für jeden Block ein Stream Tag angefügt. Die Blockgröße wird an den Block *Polyphase Clock Sync* mit dem Parameter "Block Size" übergeben, somit findet er die angefügten Tags in dem entsprechend großen Fenster. Sofern sich die im Stream Tag der aufgenommenen Samples angegebene Symbolrate von der aktuellen unterscheidet, wird die Funktion zum Ändern der Symbolrate aufgerufen.

In diesem Fall werden die lokalen Variablen an die neue Symbolrate angepasst. Unter anderem ändert sich auch die Form der Filter. Diese werden bei einer Änderung der Rate entsprechend aktualisiert. Es wird immer ein RRC Filter mit Roll-off-Faktor $r = 0,35$ verwendet. Vorher war es mit dem Block *Polyphase Clock Sync* noch möglich, einen zu verwendenden Filter als Parameter zu übergeben. Der Einfachheit halber und weil ohnehin immer ein RRC Filter verwendet wird, wurde diese Funktionalität weggelassen.

5.3.3. Fokus auf den Klassifikator und Phasenoffset

Die Modulationsartenerkennung und der Ausgleich des Phasenoffsets finden im Block *Modulation Classifier* statt. Wie in Kapitel 3 genauer beschrieben, basiert die Entscheidung auf der Berechnung von Kumulanten.

Es ist wünschenswert, dass für die Klassifikation ein Signal mit korrigierter Phase vorliegt, denn so kann zur Entscheidung die komplexwertige Kumulante herangezogen werden. Muss man hingegen einen Phasenfehler annehmen, so kann man nur den Betrag der Kumulante nutzen. Dies verringert die Abstände der Entscheidungsgrenzen und somit die Rate der korrekten Klassifikation.

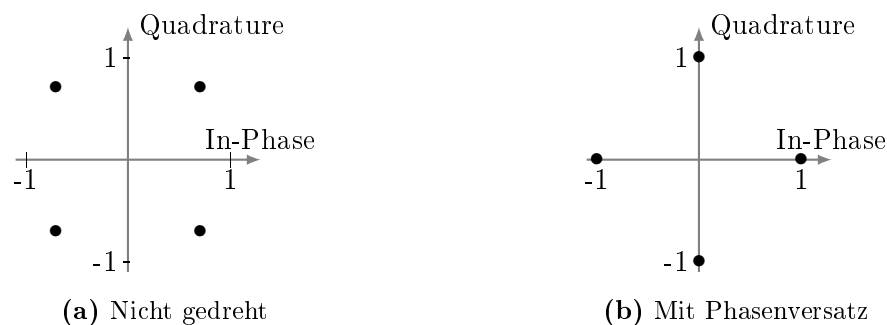


Abbildung 5.12.: Normierte QPSK-Konstellationsdiagramme in verschiedener Ausrichtung

Versucht man, zuerst den Phasenoffset unter Annahme der jeweiligen Hypothese zu berechnen, und die Prüfung danach mittels Kumulantenberechnung durchzuführen, so stellt man sich vor ein Problem bei den Modulationsarten QPSK und 16-Quadrature Amplitude Modulation (QAM). Die Entscheidung wird immer auf die zuerst gestellte Hypothese fallen, in diesem Beispiel 16-QAM. Der Grund dafür ist, dass der Phasenschätzer die Phase von QPSK so schätzt, dass das Konstellationsdiagramm wie in Abb. 5.12a aussieht. Der Klassifikator erwartet jedoch ein wie in Abb. 5.12b gedrehtes Konstellationsdiagramm und die Hypothese QPSK wird abgelehnt. Der Phasenschätzer für 16-QAM dreht die Punkte dann so wie in Abb. 5.12a und nimmt die Hypothese für 16-QAM an. Eine Korrektur durch konstante Phasenverschiebung um $\frac{\pi}{4}$ führt nicht zum gewünschten Ergebnis, da in dem Fall auch bei gesendetem 16-QAM für QPSK entschieden wird.

Sollte zuerst die Hypothese, dass QPSK gesendet wurde, geprüft werden, so geschieht das Gleiche, nur andersherum. Das Konstellationsdiagramm von 16-QAM wird um 45° gedreht und der Klassifikator nimmt die Hypothese für QPSK fälschlicherweise an.

Die Lösung für das Problem ist, wie oben schon angedeutet, die Modulation mit dem Betrag der Kumulante zu bestimmen. Ist die Modulationsart bekannt, so lässt sich unter dieser Annahme auch die Phase berechnen und ausgleichen. Der Nachteil ist, dass die Entscheidungsgrenzen (siehe Gleichung (3.6)) bei Verwendung des Betrags der Kumulante näher beieinander liegen und Fehlentscheidungen bei niedrigen SNR somit häufiger auftreten.

6. Zusammenfassung und Fazit

In dieser Arbeit wurde die Machbarkeit einer automatischen Modulationsartenerkennung in einer Simulation gezeigt. Dazu wurde auf die nötigen theoretischen Grundlagen zu statistischen Kennwerten und zur Entscheidungstheorie eingegangen. Der verwendete merkmalsbasierte Klassifikator nutzt als Entscheidungsmerkmal den Betrag der normierten Kumulante vierter Ordnung mit der Konjugation Null. Damit ist es möglich, zwischen den vier Modulationsverfahren BPSK, QPSK, 8-PSK und 16-QAM zu unterscheiden.

Um die Klassifikation durchführen zu können, wurde ein geeigneter Empfänger entworfen, welcher die nötige Synchronisation von Frequenz, Symbolrate und Zeit gewährleistet. Die Funktionsprinzipien dieser Synchronisationen wurden vorgestellt. Die verwendeten Verfahren ermöglichen eine Kompensation der Kanaleinflüsse ohne Vorinformationen zu dem empfangenen Signal. Dies macht eine Anwendung als Teil eines "Cognitive Radio" interessant, sprich, dass sich die Übertragung dynamisch an die Umgebungsbedingungen anpasst. Am Ende der Verarbeitungskette liegen die vollständig rekonstruierten Symbole mit korrigierter Phase vor.

Ferner wurde auf das Werkzeug GNU Radio eingegangen, welches zur Implementierung von Klassifikator und den realisierten Elementen des Empfängers genutzt wurde. Dabei ging es neben der grundsätzlichen Funktionsweise auch um Zusatzfunktionen wie der VOLK. Sie ermöglicht eine plattformübergreifende Nutzung von effizienten Befehlssätzen. Eine erhöhte Performance wird zudem durch den Einsatz der Programmiersprache C++ erreicht.

Die Leistungsfähigkeit des Klassifikators und der Kanalkompensation wurde jeweils durch Tests evaluiert. Dazu wurde die Klassifikationsrate bei verschiedenen Beobachtungsdauern unter variablem SNR untersucht. Die zusätzliche Betrachtung des verwendeten Merkmals unter den gegebenen Bedingungen erlaubt ein tieferes Verständnis der Ergebnisse aus vorangegangenen Klassifikationstests.

Um aus dem Ausgangsstrom letztendlich die gesendete Bitfolge zu rekonstruieren, ist ein Entscheider erforderlich, der die empfangenen Symbole je nach erkannter Modulationsart einer Bitfolge zuordnet. Ein derartiger Entscheider ist in einer Erweiterung dieser Arbeit denkbar. Außerdem ließe sich der hier konstruierte Empfänger samt Klassifikator zusätzlich zur Simulation noch "over the air", also in einer Übertragung durch die Luft, testen.

A. Abkürzungsverzeichnis

AWGN	Additive White Gaussian Noise
Abb.	Abbildung
ADC	analog-to-digital converter
BPSK	Binary Amplitude Modulation
DSP	Digital Signal Processing
FFT	Fast Fourier Transform
GRC	GNU Radio Companion
GUI	Graphical User Interface
ISI	Intersymbolinterferenz
I&Q-Verfahren	In-Phase-&-Quadrature-Verfahren
MAP	Maximum-a-posteriori-Kriterium
PSK	Phase-Shift Keying
QMW	quadratische Mittelwert
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RRC	Root-Raised-Cosine
RC	Raised-Cosine
SDR	Software-Defined Radio
SNR	Signal-to-Noise-Ratio
SIMD	Single Instruction Multiple Data

sps	samples per symbol
USRP	Universal Software Radio Peripheral
VOLK	Vector-Optimized Library of Kernels

Literaturverzeichnis

- [DABNS07] Octavia A. Dobre, Ali Abdi, Yeheskel Bar-Ness, and Wei Su. Survey of automatic modulation classification techniques: classical approaches and new trends. *IET Communications*, 1:137–156, 2007.
- [GR14] GR. What is GNU Radio. <http://gnuradio.org/redmine/projects/gnuradio/wiki/WhatIsGR>, 2014. abgerufen am 31.10.2016.
- [GS94] W. A. Gardner and C. M. Spooner. The cumulant theory of cyclostationary time-series. I. Foundation. *IEEE Transactions on Signal Processing*, 42(12):3387–3408, Dec 1994.
- [HR01] F. J. Harris and M. Rice. Multirate digital filters for symbol timing synchronization in software defined radios. *IEEE Journal on Selected Areas in Communications*, 19(12):2346–2357, Dec 2001.
- [JW02] Friedrich Jondral and Anne Wiesler. *Wahrscheinlichkeitsrechnung und stochastische Prozesse: Grundlagen für Ingenieure und Naturwissenschaftler*. Lehrbuch. Teubner, Stuttgart, 2., durchges. und aktualis. Aufl. edition, 2002.
- [Kö05] Bert-Uwe Köhler. *Konzepte der statistischen Signalverarbeitung*. Springer, Berlin, 2005.
- [Kam11] Karl-Dirk Kammeyer. *Nachrichtenübertragung : mit 38 Tabellen*. Studium. Vieweg + Teubner, Wiesbaden, 5., durchges. und erg. Aufl. edition, 2011.
- [Kro11] Kristian Kroschel. *Statistische Informationstechnik : Signal- und Mustererkennung, Parameter- und Signalschätzung*. SpringerLink : Bücher. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [Mü14] Michael Sebastian Mühlhaus. *Automatische Modulationsartenerkennung in MIMO-Systemen*. PhD thesis, 2014.
- [MdJ94] M. Moeneclaey and G. de Jonghe. ML-oriented NDA carrier synchronization for general rotationally symmetric signal constellations. *IEEE Transactions on Communications*, 42(8):2531–2533, Aug 1994.
- [Rin08] Horst Rinne. *Taschenbuch der Statistik*. Deutsch, Frankfurt am Main, 4., vollst. überarb. u. erw. Aufl. edition, 2008.

- [SS00] Ananthram Swami and Brian M. Sadler. Hierarchical digital modulation classification using cumulants. *IEEE Transactions on Communications*, 48(3):416–429, 2000.
- [VOL15] VOLK. Vector-Optimized Library of Kernels. <http://gnuradio.org/redmine/projects/gnuradio/wiki/Volk>, 2015. abgerufen am 31.10.2016.
- [wik06] Datei:{BPSK, QPSK, 8PSK, 16QAM}Gray Coded.svg. https://de.wikipedia.org/wiki/Datei:{BPSK, QPSK, 8PSK, 16QAM}_Gray_Coded.svg, 2006. abgerufen am 31.10.2016.