

Schwerpunktlabor Nachrichtentechnik

Digitale Modulation

– SS 2013 –

Autor: Heiko Schmidt

Betreuer: Henning Paul

NW1, Raum N2360, Tel.: 0421/218-62399

E-mail: paul@ant.uni-bremen.de



Universität Bremen, FB1
Institut für Telekommunikation und Hochfrequenztechnik
Arbeitsbereich Nachrichtentechnik
Prof. Dr.-Ing. A. Dekorsy
Postfach 33 04 40
D-28334 Bremen

WWW-Server: <http://www.ant.uni-bremen.de>

Version vom April 2013

Inhaltsverzeichnis

1	Einführung	1
1.1	Vor- und Nachbereitung des Versuchs	1
1.2	Motivation	2
2	Theoretische Grundlagen	5
2.1	Übersicht über die Teilkomponenten eines Datenübertragungssystems	5
2.2	Impulsformung und Matched-Filterung	6
2.3	Lineare Modulationsverfahren	7
2.4	Kohärente Demodulation von QPSK-Signalen	8
2.5	Kohärente und inkohärente Demodulation von DQPSK-Signalen	8
2.6	Signal- und Rauschleistungsdichte im Bandpaß- und Basisbandbereich	10
2.7	Übertragung über AWGN-Kanäle	12
3	Übungsaufgaben	16
4	Versuchsdurchführung	18
4.1	Impulsformer	18
4.2	Simulation eines Symboltaktmodells	21
4.3	Simulation mit Impulsformung	26

Kapitel 1

Einführung

1.1 Vor- und Nachbereitung des Versuchs

Für die erfolgreiche und zügige Durchführung dieses Versuchs ist es unerlässlich, die theoretischen Grundlagen zu beherrschen. Deshalb werden diese Kenntnisse vor Versuchsbeginn überprüft. *Bei einer unzureichenden Vorbereitung erfolgt keine Zulassung zur Durchführung des Versuches!* Desweiteren sollten die bereits vorhandenen MATLAB-Kenntnisse – insbesondere das *Erstellen, Modifizieren und Abspeichern von Grafiken*¹ – vor dem Versuchsbeginn wieder aufgefrischt werden, damit sich die Grundlagen der MATLAB-Programmierung nicht zum Schwerpunkt dieses Nachrichtentechnik-Praktikums entwickeln!

- Lösen Sie *vor dem Versuchstermin* die Übungsaufgaben in Abschnitt 3. Eine Besprechung der Ergebnisse erfolgt vor Versuchsbeginn.
- Während des Versuches sind *alle* mit einem Punkt (●) gekennzeichneten Fragestellungen in kurzer *schriftlicher* Form zu beantworten. In regelmäßigen Abständen sind die bereits bearbeiteten Aufgaben mit dem Betreuer zu besprechen. Bei richtiger Beantwortung der Fragestellungen gelten die entsprechenden Aufgaben als erfolgreich bearbeitet.
- Nach Ablauf der für den Versuch vorgesehenen Zeit müssen alle nicht erfolgreich abgeschlossenen Aufgaben schriftlich nachbearbeitet und in Form eines Versuchsprotokolls, welches eine kurze Zusammenfassung und die Ergebnisse enthält, *innerhalb von zwei Wochen nach dem Versuchstermin* beim Betreuer abgegeben werden. Beziehen Sie sich bei den Antworten auf die in der Versuchsbeschreibung vorgegebene Nummerierung der Aufgabenstellungen!

¹MATLAB-Befehle: `figure`, `plot`, `subplot`, `axis`, `grid`, `xlabel`, `ylabel`, `title`, `legend`, etc.

1.2 Motivation

Die Übertragung digitaler Daten über analoge Kanäle, z.B. Funkkanäle, erfordert eine Anpassung des Digitalsignals an die Eigenschaften des Übertragungskanals, welcher im allgemeinen Bandpaßcharakter besitzt. In der makroskopischen Physik existieren jedoch keine digitalen Größen. Jede Spannung, jeder Strom, jede elektrische oder magnetische Feldstärke ist eine zeitkontinuierliche, analoge Größe. Die vorliegenden digitalen Daten müssen an reale Übertragungsbedingungen angepasst werden (Bild 1.1).



Bild 1.1: Übertragung digitaler Daten

Digitale Datenströme lassen sich auf unterster Ebene mit Hilfe von Bitströmen beschreiben. Obwohl digitale Signale in der Realität nur in Form von Spannungs- oder Strompegeln existieren, wird ein binärer Datenstrom $b(k)$ medienunabhängig und damit auch dimensionslos definiert.

$$b(k) \in [0, 1], \forall k \quad (1.1)$$

Durch die Einführung der Bitdauer T_b und der Bitrate

$$R_{\text{bit}} = \frac{1}{T_b} \quad (1.2)$$

erhält der Zeitindex k einen Bezug zur Zeitachse.

Der reine Bitstrom besitzt weder ein Spektrum noch eine bestimmte Bandbreite. Erst nach einer zeitkontinuierlichen Interpretation der einzelnen Bits mit Hilfe von gewichteten Impulsen erhält der Bitstrom ein Spektrum bzw. eine Bandbreite. Die zeitkontinuierliche Darstellung eines binären Datenstroms durch Aneinandereihung von Rechteckimpulsen ist bereits eine einfache Form von digitaler Modulation.

Eigentlich umfasst der hier auftretende Begriff der *digitalen Modulation* drei Stufen. Zuerst findet eine Signalraumzuordnung statt. Je nach Stufigkeit M der verwendeten Modulationsart werden $\text{ld}(M)$ Bits einem komplexen Wert zugeordnet. Die zweite Modulationsstufe beschreibt die Gewichtung von zeitkontinuierlichen Impulsen. In der dritten Stufe wird das komplexe zeitkontinuierliche Signal in die (reelle) Bandpasslage verschoben, damit es übertragen werden kann. Im Empfänger werden diese drei Stufen wieder rückgängig gemacht, um den Bitstrom möglichst originalgetreu wiederzugewinnen. Die Beschreibung von Übertragungssystemen erfolgt im allge-



Bild 1.2: Digitales Übertragungssystem im Basisband

meinen in Basisbandlage, Bild 1.2. Da die spektrale Verschiebung in die Bandpaßlage bei allen Modulationsverfahren identisch ist, soll in diesem Versuch auf die dritte Stufe verzichtet und auch der Kanal in Basisbandlage beschrieben werden. Im folgenden wird die Umformung eines

binären Datenstroms in ein komplexes zeitkontinuierliches Signal in Basisbandlage als *Modulation* und die Rückgewinnung des Bitstroms aus diesem Signal als *Demodulation* bezeichnet.

In diesem Versuch werden die folgenden *linearen* Modulationsformen betrachtet:

- ASK (Amplitude Shift Keying)
- PSK (Phase Shift Keying)
- QAM (Quadrature Amplitude Modulation)
- DPSK (Differential Phase Shift Keying)

Wird das modulierte Signal $s_{mod}(t)$ über einen idealen Kanal übertragen, ist eine fehlerfreie Reproduktion der gesendeten Daten möglich. Bei Verwendung eines AWGN-Kanals können einige Bits aufgrund von Rausch-Störungen nicht mehr fehlerfrei erkannt werden. In diesem Fall stellt sich je nach verwendeten Modulationsverfahren und Signal-Rausch-Abstand eine gewisse Bitfehlerrate ein.

Bei einem Mobilfunkkanal treten zusätzlich Störungen durch Laufzeitdifferenzen der verschiedenen Echopfade auf, die sich in Form von Intersymbolinterferenz (ISI) äußern. Mit Hilfe eines einfachen Symboltakt-Entzerrers können die störenden Einflüsse der ISI deutlich reduziert werden.

Bei vorgegebenen Quelldaten sollen die modulierten Signale im Zeitbereich berechnet und über einen Kanal im Basisband übertragen werden. Die Einflüsse möglicher Störgrößen sind mit Hilfe von Bitfehlerratenuntersuchungen zu verdeutlichen.

Die für den Versuch erforderlichen theoretischen Grundkenntnisse werden in den Kapiteln 8-11 des Lehrbuchs

Nachrichtenübertragung von *K.D. Kammeyer*, 4. Auflage, Teubner Stuttgart, 2007

detailliert beschrieben. Die wichtigsten Formeln und Zusammenhänge befinden sich in Abschnitt 2 dieser Versuchsbeschreibung. Zur Stärkung des theoretischen Grundlagenwissens befinden sich in Abschnitt 3 einige Übungsaufgaben. Die genaue Anleitung für die Versuchsdurchführung folgt in Abschnitt 4.

Ernst gemeinte Anregungen zur Verbesserung dieses Versuchs sowie Hinweise auf Rechtschreibfehler werden von mir gerne entgegengenommen.

Kapitel 2

Theoretische Grundlagen

2.1 Übersicht über die Teilkomponenten eines Datenübertragungssystems

Bild 2.1 zeigt das Blockschaltbild eines digitalen Datenübertragungssystems mit linearer Modulationsform und einem linearen Entzerrer am Eingang des Empfängers.

Am Sender erfolgt zunächst eine Seriell/Parallel-Wandlung des Bitstroms $b(k)$ einer bel. Datenquelle, wobei Bitgruppen der Länge $m = \text{ld}(M)$ gebildet werden; M bezeichnet die Stufigkeit der zugrundeliegenden Modulation. Es gilt also für die Symboldauer

$$T_s = \text{ld}(M) \cdot T_{\text{bit}}. \quad (2.1)$$

Die Modulationsform wird in dem Block “Signalraumzuordnung“ festgelegt. Real- und Imaginärteil der komplexen Symbole $d(i) = d'(i) + j \cdot d''(i)$ dienen als Gewichte für den Grundimpuls $g(t)$; man erhält die komplexe Einhüllende

$$s(t) = T \sum_i (d'(i) + j \cdot d''(i)) \cdot g(t - iT). \quad (2.2)$$

Da die Filterimpulsantwort die Dimension Zeit^{-1} aufweist, ist die Multiplikation mit T notwendig, um ein dimensionsloses Signal $s(t)$ zu erhalten.

Durch Multiplikation mit der komplexen Trägerschwingung $e^{j\omega_0 t}$, anschließender Realteilbildung und Skalierung mit dem Faktor $\sqrt{2}$, um die Halbierung der Leistung des Sendesignals durch die Realteilbildung zu kompensieren entsteht das Modulationssignal $x(t)$ im Bandpaß-Bereich.

Am Empfänger erfolgt zunächst die Transformation ins Basisband durch Trägermultiplikation und Tiefpaßfilterung mit dem Filterpaar $h(t)$. Es folgt eine Abtastung, wobei – je nach Wahl des nachfolgenden Entzerrertyps – ein oder zwei Abtastwerte pro Symbolintervall erfaßt werden ($w = 1$ oder $w = 2$). Der nachfolgende Entzerrer (in Bild 2.1 als lineare Struktur dargestellt) ist komplex und weist demgemäß die dargestellte Konfiguration mit vier paarweise gleichen Teilsystemen auf. Am Entzerrerausgang erfolgt im Falle eines $T/2$ -Entzerrers eine Abtastratenreduktion um den Faktor zwei. Anschließend findet eine Symbolentscheidung und die Zuordnung der m -Bit-Gruppen statt. Eine Parallel/Seriell-Wandlung liefert wieder den seriellen Bitstrom.

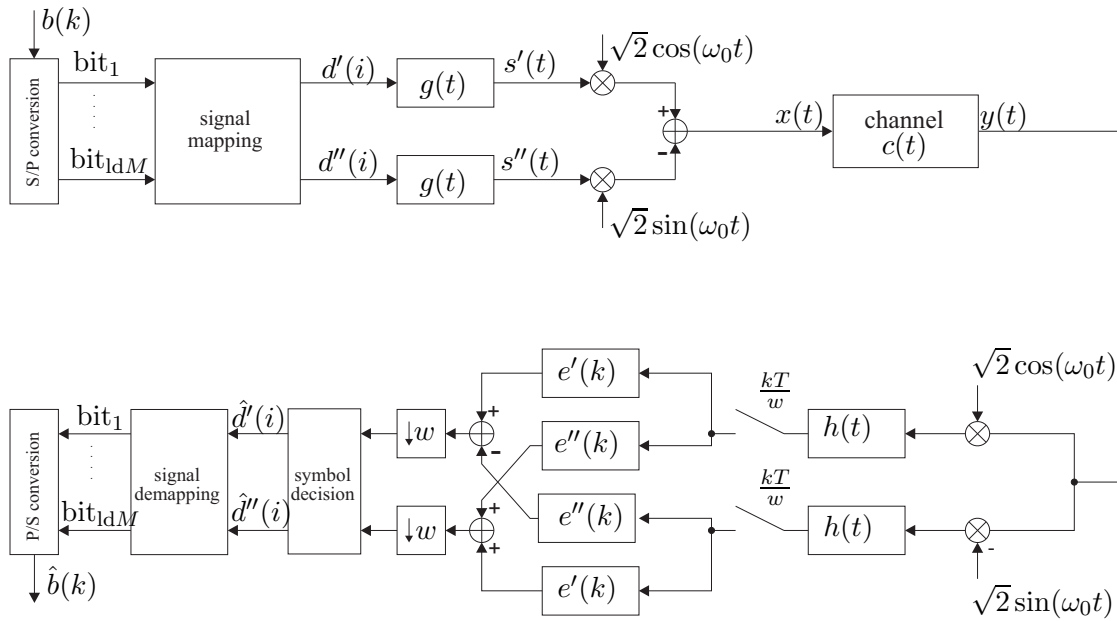


Bild 2.1: Blockschaltbild eines Datenübertragungssystems

2.2 Impulsformung und Matched-Filterung

Um im Falle eines idealen Übertragungskanals eine verzerrungsfreie Übertragung zu erreichen, müssen die Impulsformer $g(t)$ und $h(t)$ gemeinsam die 1. Nyquist-Bedingung erfüllen, also

$$g(t) * h(t) = \frac{1}{T} \begin{cases} 1 & \text{für } t = i_0 \cdot T \\ 0 & \text{für } t = i \cdot T, \quad i \neq i_0. \end{cases} \quad (2.3)$$

Zusätzlich muß die Matched-Filter-Bedingung erfüllt sein

$$h(t) = K \cdot g^*(i_0 T - t), \quad K \in \mathbb{R}, K > 0, \quad (2.4)$$

um bei additiver Rauschüberlagerung auf dem Übertragungsweg am Empfangsfilterausgang das maximale S/N -Verhältnis zu erzielen. Setzt man für $h(t)$ und $g(t)$ *reelle, linearphasige* Impulse an, so vereinfacht sich (2.4) bei nichtkausaler Formulierung zu

$$h(t) = K \cdot g(t); \quad (2.5)$$

im Sender und Empfänger werden also die gleichen Filter eingesetzt. Üblicherweise wird für das Gesamtsystem eine Cosinus-roll-off-Charakteristik angesetzt, so daß am Sende- und Empfangsfilter jeweils Wurzel-Cosinus-roll-off-Filter vorgesehen werden. Die Impulsantworten sind geschlossen zu formulieren.

Cosinus-roll-off-Impulsantwort:

$$g(t) * h(t) = \frac{1}{T} \frac{\sin(\pi \frac{t}{T})}{\pi \frac{t}{T}} \cdot \frac{\cos(r \cdot \pi \frac{t}{T})}{1 - (2r \frac{t}{T})^2} \quad (2.6)$$

Wurzel-Cosinus-roll-off-Impulsantwort (Buch, (8.3.17b)):

$$g(t) = \frac{4r \frac{t}{T} \cos[\pi(1+r)\frac{t}{T}] + \sin[\pi(1-r)\frac{t}{T}]}{[1 - (4r \frac{t}{T})^2] \pi t} \quad (2.7)$$

Diese Darstellungen sind nichtkausal und zeitlich unbegrenzt. In praktischen Realisierungen werden Approximationen durch digitale Filter mit endlicher Impulsantwort – linearphasige FIR-Filter – vorgenommen.

Vielfach werden Filter eingesetzt, die neben der ersten auch die zweite Nyquist-Bedingung erfüllen: Dazu ist für den Roll-off-Faktor $r = 1$ zu setzen; die relative horizontale Augenöffnung beträgt dann 100%. Für Impulse, die die erste und zweite Nyquist-Bedingung erfüllen, gilt (bei nichtkausaler Formulierung, also $i_0 = 0$):

$$g(t) * h(t) = \frac{1}{T} \begin{cases} 1 & \text{für } t = 0 \\ \frac{1}{2} & \text{für } |t| = \frac{T}{2} \\ 0 & \text{für } |t| = \nu \cdot \frac{T}{2}; \nu = 2, 3, 4, \dots \end{cases} \quad (2.8)$$

2.3 Lineare Modulationsverfahren

Wie bereits erwähnt, wird durch die Signalraum-Zuordnung in Bild 2.1 die Modulationsform festgelegt. Für die Versuchsdurchführung werden die Beispiele QPSK (Quarternary PSK: 4-PSK), 8-PSK, 16-QAM und 16-ASK/PSK vorgesehen. Eine Sonderrolle spielt das Offset-QPSK-Verfahren, bei dem Real- und Imaginärteil der Datensymbole zeitlich um $\frac{T}{2}$ versetzt übertragen werden.

$$s_{OQPSK}(t) = T \left[\sum_{i=-\infty}^{\infty} d'(t)g(t - iT) + j \sum_{i=-\infty}^{\infty} d''(t)g(t - \frac{T}{2} - iT) \right] \quad (2.9)$$

Der Sinn dieser Maßnahme besteht darin, die Phasenübergänge von π durch Zerlegung in zwei $\pi/2$ -Phasenübergänge zu vermeiden und damit eine bessere Konstanz der Betragseinhüllenden zu erreichen.

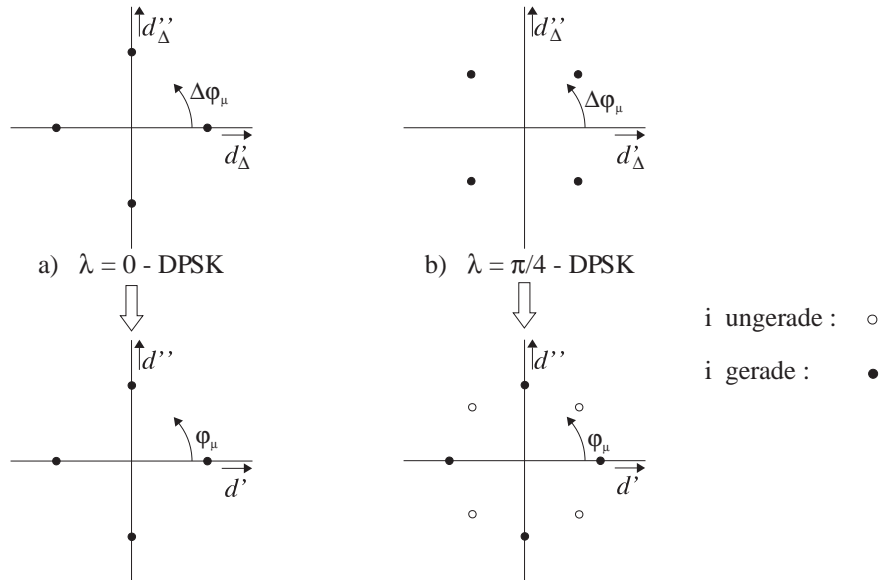


Bild 2.2: Differenzphasen und absolute Phasen bei DQPSK

In praktischen Übertragungssystemen ist die absolute Phase des gesendeten Signals in der Regel nicht bekannt. Dies gilt auch bei der Anwendung einer Trägerphasenregelung, da hierbei eine

Phasenunsicherheit von $\frac{2\pi}{M}$ verbleibt. Aus diesem Grunde wird im allgemeinen eine differentielle Phasenmodulation angewendet, bei der die Differenz zweier aufeinanderfolgender Phasenwerte die zu übertragende Information enthält.

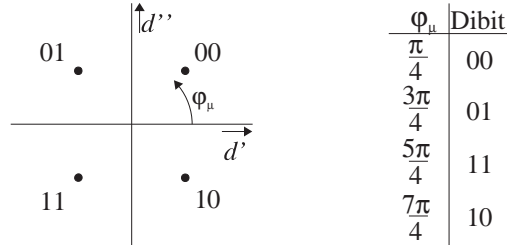


Bild 2.3: QPSK in $\pi/4$ -Phasenlage mit Gray-Codierung

$$\Delta\varphi_\mu(i) = \varphi(i) - \varphi(i-1) \quad ; \quad \Delta\varphi_\mu(i) = \frac{2\pi}{M}\mu + \lambda \quad , \quad \lambda \in \{0, \frac{\pi}{M}\} \quad (2.10)$$

Als Beispiel wird $M = 4$, also eine DQPSK, betrachtet. Für den Phasenoffset λ können hier die Werte 0 und $\frac{\pi}{4}$ vorgegeben werden; man erhält damit zwei verschiedene Formen der DPSK-Modulation: Im Falle $\lambda = 0$ ergeben sich für die absolute Phase gemäß (2.10) die diskreten Werte $\varphi(i) \in [0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}]$; es entsteht also ein Vierphasensignal (siehe Bild 2.2a). Für $\lambda = \frac{\pi}{4}$ hingegen erhält man aufgrund von (2.10) ein 8-stufiges Signal, denn mit dem Anfangswert $\varphi(0) = 0$ gilt

$$\varphi(2i+1) \in \left\{ \frac{\pi}{4}, \frac{3}{4}\pi, \frac{5}{4}\pi, \frac{7}{4}\pi \right\} \quad (2.11)$$

$$\varphi(2i) \in \left\{ 0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi \right\} . \quad (2.12)$$

Diese alternierende Veränderung des absoluten Phasenrasters wird in Bild 2.2b veranschaulicht. Der Vorteil des $\frac{\pi}{4}$ -DQPSK-Verfahrens besteht darin, daß Phasenübergänge von $\pm\pi$ vermieden werden; ähnlich wie bei der Offset-QPSK-Modulation wird damit die Konstanz der Betragshüllenden verbessert.

2.4 Kohärente Demodulation von QPSK-Signalen

Es wird grundsätzlich von einer Empfängerstruktur gemäß Bild 2.1 ausgegangen, in der das empfangene Signal zunächst in die komplexe Basisbandebene transformiert wird. Für die Symbolentscheidung bestehen dann die beiden prinzipiellen Möglichkeiten der kohärenten und der inkohärenten Demodulation. Eine kohärente Demodulation eines QPSK-Signals erfordert bei ideal angenommener Trägersynchronisation (mit korrekter absoluter Phasenregeneration) lediglich die Vorzeichenbestimmung des Real- und Imaginärteils. Liegt eine Gray-Codierung und eine $\frac{\pi}{4}$ -Phasenlage gemäß Bild 2.3 vor, so liefern diese Entscheidungen unmittelbar die zugeordneten Dibits. Das Blockschaltbild eines möglichen QPSK-Demodulators ist in Bild 2.4 zu sehen.

2.5 Kohärente und inkohärente Demodulation von DQPSK-Signalen

Auch für differentielle PSK ist eine kohärente Demodulation möglich (DECPSK, Differentially Encoded PSK). Wir betrachten hierzu das Beispiel einer DQPSK mit $\lambda = 0$; die absoluten Pha-

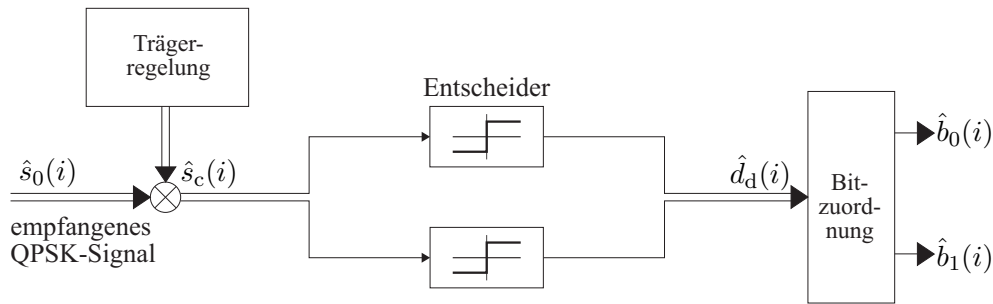


Bild 2.4: Blockschaltbild eines QPSK-Demodulators

senlagen des modulierten Signals sind dann $\varphi(i) \in \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$. Am Empfänger wird das Signal zunächst durch eine Trägerregelung in die $\frac{\pi}{4}$ -Phasenlage gemäß Bild 2.3 gebracht. Nach einer Symbolentscheidung durch Vorzeichenbestimmung des Real- und Imaginärteils erfolgt dann die Differenzphasenbestimmung gemäß Bild 2.5a.

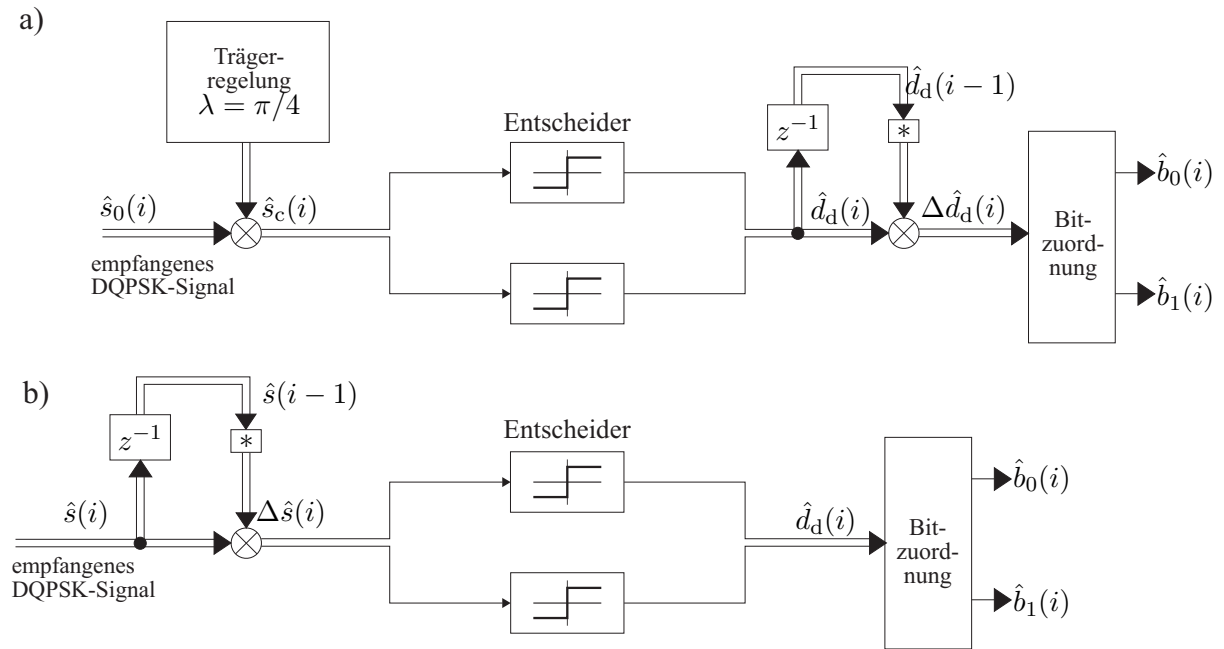


Bild 2.5: a) Kohärenter DQPSK-Demodulation (DECQPSK, $\lambda = 0$)
b) Blockschaltbild eines inkohärenten DQPSK-Demodulators

Bei inkohärenter Demodulation differentieller PSK-Signale ist die Momentanphase des Trägers nicht bekannt. Folglich kann keine Symbolentscheidung vor der differentiellen Dekodierung durchgeführt werden. Durch die Multiplikation zur Phasendifferenzbildung geht das Kanalrauschen mehrfach in die Phasendifferenz zeitlich benachbarter Symbole ein, so daß bei inkohärenter DQPSK ein deutlich schlechteres Bitfehlerverhalten als bei kohärenter DQPSK zu erwarten ist. Bezieht man den Verlust auf die Bitfehlerkurve einer QPSK-Modulation, so beträgt dieser ≈ 2.3 dB. Bild 2.5b zeigt das Blockschaltbild eines inkohärenten DQPSK-Modulators.

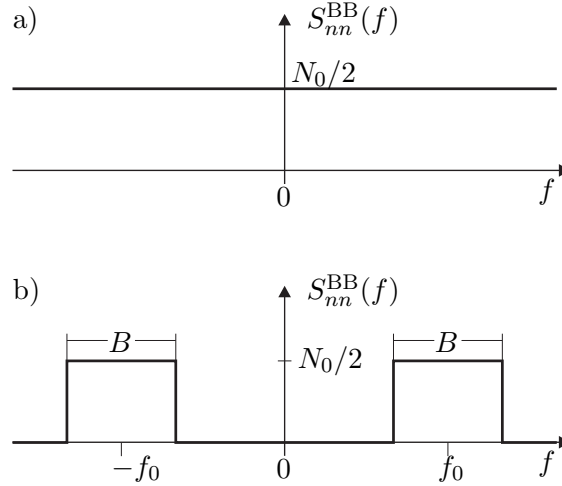


Bild 2.6: a) Rauschleistungsdichte bei weißem Rauschen; b) Rauschleistungsdichte von bandbegrenztem Rauschen

2.6 Signal- und Rauschleistungsdichte im Bandpaß- und Basisbandbereich

Gegeben sei ein Sendesignal $s^{BP}(t)$, daß durch additives weißes Gaußverteiltes Rauschen $n^{BP}(t)$ mit der Rauschleistungsdichte $S_{nn}^{BP}(f) = N_0/2$ überlagert werden soll. Ohne Bandbegrenzung, wie Bild 2.6a zeigt, geht die Leistung des Rauschens σ_n^2 gegen unendlich:

$$\lim_{f_g \rightarrow \infty} \sigma_n^2 = \lim_{f_g \rightarrow \infty} \int_{-f_g}^{f_g} \frac{N_0}{2} df = \infty \quad (2.13)$$

Im Falle eines in Bild 2.6b dargestellten bandbegrenzten Rauschens mit der Bandbreite B um die Mittenfrequenz f_0 ergibt sich für die Rauschleistung

$$\sigma_{n^{BP}}^2 = \int_{-f_0-B/2}^{-f_0+B/2} \frac{N_0}{2} df + \int_{f_0-B/2}^{f_0+B/2} \frac{N_0}{2} df = B \cdot N_0. \quad (2.14)$$

Häufig besteht die Aufgabe darin, aus einem gegebenem Verhältnis E_s/N_0 die resultierende Rauschleistung des bandbegrenzten Rauschens, bei bekannter Signalleistung zu berechnen. Dabei ist zu beachten, daß E_s/N_0 üblicherweise im Bandpaßbereich angegeben wird, da dort die tatsächliche, physikalische Störung stattfindet. Die mittlere Signalleistung im Bandpaßbereich \bar{P}^{BP} wird durch folgende Beziehung gegeben:

$$\bar{P}^{BP} = \frac{\bar{E}_s}{T_s} \quad (2.15)$$

Bei gegebenem E_s/N_0 -Verhältnis ergibt sich mit (2.14):

$$\begin{aligned} \frac{\bar{E}_s}{N_0} &= \frac{\bar{P}_s^{BP} \cdot T_s}{\sigma_{n^{BP}}^2 / B} = \frac{\bar{P}^{BP} \cdot T_s \cdot B}{\sigma_{n^{BP}}^2} \\ \sigma_{n^{BP}}^2 &= \frac{\bar{P}^{BP} \cdot T_s \cdot B}{\bar{E}_s / N_0} \end{aligned} \quad (2.16)$$

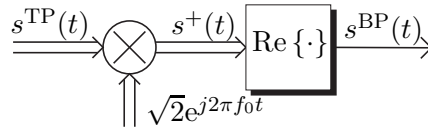


Bild 2.7: Modulation eines komplexen Basisbandsignals in den reellwertigen Bandpaßbereich

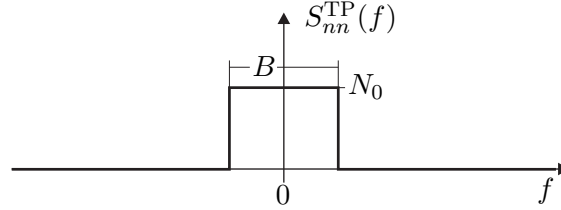


Bild 2.8: Rauschleistungsdichte im Tiefpaßbereich

Somit ist bei gegebener Symboldauer T_s und Bandbreite B die Rauschleistung des bandbegrenzten Rauschens zu bestimmen.

Das Basisbandsignal $s^{\text{TP}}(t)$ wird durch Multiplikation mit $e^{j2\pi f_0 t}$ vom Tiefpaß- in den Bandpaßbereich hochgemischt. Die Realteilbildung $\text{Re}\{\cdot\}$ erzeugt hieraus das Sendesignal $s^{\text{BP}}(t)$. Um sicherzustellen, daß das Basisband- und Bandpaßsignal dieselbe Leistung aufweisen, muß eine Skalierung mit dem Faktor $\sqrt{2}$ durchgeführt werden, da die Realteilbildung die Signalleistung halbiert. Abbildung 2.7 zeigt einen solchen Mischer.

Wenn davon ausgegangen werden kann, daß diese Korrektur bei allen Hochmischvorgängen angewendet wird, gilt

$$\bar{P}^{\text{TP}} = \bar{P}^{\text{BP}}. \quad (2.17)$$

Damit die Leistungsgleichheit auch beim Heruntermischen gewährleistet ist, ist auch dort eine Skalierung nötig. Die Multiplikation des Bandpaßempfangssignals $y(t)$ mit $e^{-j2\pi f_0 t}$ verschiebt die Spektralanteile um $-f_0$ herum nach $-2f_0$, wo sie vom folgenden Empfangsfilter entfernt werden. Dies hat zur Folge, daß auch hier die Leistung des Empfangssignals halbiert wird. Also ist auch hier eine Skalierung mit $\sqrt{2}$ nötig.

Diese Skalierung bewirkt eine Erhöhung der Rauschleistungsdichte am Empfänger auf N_0 , in Abbildung 2.8 ist sie für den bandbegrenzten Fall und nach Entfernung der Spektralanteile bei $-2f_0$ zu sehen. Für die Rauschleistung $\sigma_{n^{\text{TP}}}^2$ am Empfänger erhält man damit den Ausdruck

$$\sigma_{n^{\text{TP}}}^2 = BN_0 = \sigma_{n^{\text{BP}}}^2. \quad (2.18)$$

Offensichtlich sind neben den Signal- auch die Rauschleistungen in Bandpaß- und Basisbanddarstellung identisch.

Daher gilt (2.16) auch für den Tiefpaßbereich, wodurch man zu

$$\sigma_{n^{\text{TP}}}^2 = \frac{\bar{P}^{\text{TP}} \cdot T_s \cdot B}{\bar{E}_s / N_0} \quad (2.19)$$

gelangt.

Aus der Gleichheit der Signalleistungen ergibt sich mit (2.15) auch unmittelbar auch die Gleichheit der Symbolenergien in Basisband- und Bandpaßbereich. In vielen Fällen wird die Symbolenergie durch Signalraumverteilung $d(k)$ und Filterfunktion $g(t)$ im Basisband angegeben.

Hiermit ergibt sich somit für die Symbolenergie

$$\bar{E}_s = T^2 \cdot \sigma_d^2 \cdot \int_{t=-\infty}^{\infty} |g(t)|^2 dt, \quad (2.20)$$

wobei σ_d^2 die mittlere Leistung der Sendesymbole angibt.

Bei Simulationen eines Übertragungssystems wird das komplexe Rauschen zeitdiskret mit der Frequenz f_a abgetastet simuliert. Es ergibt sich daher eine Bandbegrenzung des Rauschens im Tiefpaßbereich mit $B = f_a$. Handelt es sich bei der Simulation um ein sog. Symboltaktmodell, d.h. die simulierte Kanalabtastrate entspricht der Symbolrate ($f_a = \frac{1}{T_s}$), so vereinfacht sich (2.19) zu

$$\sigma_{n^{\text{TP}}}^2 = \frac{\bar{P}^{\text{TP}}}{\bar{E}_s/N_0}. \quad (2.21)$$

Falls das Sendesignal und das Störsignal überabgetastet werden, d.h. $f_a = w \cdot \frac{1}{T_s}$, so ergibt sich für die Leistung des Rauschens

$$\sigma_{n^{\text{TP}}}^2 = \frac{w \cdot \bar{P}^{\text{TP}}}{\bar{E}_s/N_0}. \quad (2.22)$$

Die Rauschleistung hängt also linear vom Überabtastrfaktor w ab.

2.7 Übertragung über AWGN-Kanäle

Für die folgenden Betrachtungen wird ein Übertragungssystem gemäß dem Blockschaltbild 2.1 zugrundegelegt. Dabei wird der Kanal als verzerrungsfrei angenommen, d.h. $c(t) = \delta(t)$, so daß am Empfänger kein Entzerrer und keine Phasenregelung erforderlich ist. Sende- und Empfangsfilter erfüllen gemeinsam die 1. Nyquist-Bedingung und genügen der Matched-Filter-Bedingung (2.4).

Auf dem Übertragungsweg wird weißes, gaußverteiltes Rauschen additiv überlagert (AWGN). Die spektrale Leistungsdichte wird im Bandpaßbereich mit $\frac{N_0}{2}$ festgelegt. Zur meßtechnischen Beschreibung der Rauschverhältnisse ist am Empfängereingang (bzw. Kanalausgang) die Angabe eines Signal/Stör-Leistungsverhältnisses ungeeignet: Das Nutzsignal $x(t)$ besitzt zwar eine endliche Leistung, nicht aber das Störsignal, dessen Leistung wegen der nicht vorhandenen Bandbegrenzung unendlich ist. Eine endliche Rauschleistung erhält man erst am Ausgang des Empfangsfilters $h(t)$; sie beträgt im äquivalenten Tiefpaßbereich ¹

$$\text{E}\{|n(t)|^2\} = N_0 \int_{-\infty}^{\infty} |h(t)|^2 dt. \quad (2.23)$$

Wegen dieser Probleme wird der Rauscheinfluß auf dem Übertragungswege durch die in jedem Fall endliche Größe E_s/N_0 beschrieben, wobei

$$E_s = T^2 \text{E}\{|d(i)|^2\} \cdot \int_{-\infty}^{\infty} |g(t)|^2 dt \quad (2.24)$$

¹Man beachte, daß bei Festlegung der spektralen Leistungsdichte $N_0/2$ im Bandpaßbereich sich nach der Tiefpaß-Transformation der doppelte Wert, also N_0 , ergibt.

die Energie des gesendeten Symbols im Bandpaß-Bereich und $\frac{N_0}{2}$ die spektrale Leistungsdichte des Kanalrauschens ist. Es sei noch angemerkt, daß oftmals auch die Größe E_b/N_0 angegeben wird, wobei

$$E_b = \frac{E_s}{\text{ld}(M)} \quad (2.25)$$

die auf ein Bit des übertragenen Symbols bezogene Energie bezeichnet. Für QPSK gilt also $E_b = E_s/2$. Wie erwähnt ergibt sich nach der Matched-Filterung am Empfänger eine endliche Rauschleistung und somit ein angebbares S/N -Leistungsverhältnis. Das S/N -Verhältnis am Empfangsfilter-Ausgang soll im folgenden für einen festgelegten $\frac{E_s}{N_0}$ -Wert errechnet werden.

Das gesendete Datensignal lautet in der äquivalenten Tiefpaß-Darstellung

$$s(t) = T \sum_{\ell=-\infty}^{\infty} d(\ell)g(t - \ell T). \quad (2.26)$$

Im störungsfreien Fall ergibt sich am Ausgang des Empfangsfilters bei zeitkontinuierlicher Formulierung

$$s(t) * h(t) = T \sum_{\ell=-\infty}^{\infty} d(\ell) \int_{-\infty}^{\infty} g(\tau - \ell T)h(t - \tau) d\tau. \quad (2.27)$$

Setzt man die Matched-Filter-Beziehung (2.4) ein (in nichtkausaler Schreibweise, d.h. $i_0 = 0$) und führt eine Symbolabtastung durch, so erhält man für die ungestörte Entscheider-Eingangsfolge

$$\begin{aligned} \hat{d}(i) &= K \cdot s(t) * g^*(-t)|_{t=iT} = KT \sum_{\ell=-\infty}^{\infty} d(\ell) \underbrace{\int_{-\infty}^{\infty} g(\tau - \ell T)g^*(\tau - iT) d\tau}_{=0 \text{ für } i \neq \ell} \\ &= KT \cdot d(i) \int_{-\infty}^{\infty} |g(\tau - iT)|^2 d\tau \\ &= KT \cdot d(i) \int_{-\infty}^{\infty} |g(\tau)|^2 d\tau. \end{aligned} \quad (2.28)$$

Für die Leistung dieser Folge erhält man

$$\sigma_d^2 = K^2 T^2 \cdot \mathbb{E}\{|d(i)|^2\} \left[\int_{-\infty}^{\infty} |g(\tau)|^2 d\tau \right]^2. \quad (2.29)$$

Weiterhin gilt für die Leistung des Rauschens am Matched-Filter-Ausgang mit (2.23)

$$\begin{aligned} \sigma_n^2 &= N_0 \int_{-\infty}^{\infty} |h(t)|^2 dt = N_0 K^2 \int_{-\infty}^{\infty} |g(-t)|^2 dt, \\ &= N_0 K^2 \int_{-\infty}^{\infty} |g(t)|^2 dt, \end{aligned} \quad (2.30)$$

so daß man für das S/N -Verhältnis folgende Beziehung erhält:

$$\begin{aligned}
 \frac{S}{N} &= \frac{\sigma_d^2}{\sigma_n^2} = \frac{K^2 T^2 \cdot \mathbb{E}\{|d(i)|^2\} \left[\int_{-\infty}^{\infty} |g(\tau)|^2 d\tau \right]^2}{K^2 \cdot N_0 \int_{-\infty}^{\infty} |g(t)|^2 dt} \\
 &= \frac{T^2 \mathbb{E}\{|d(i)|^2\} \int_{-\infty}^{\infty} |g(\tau)|^2 d\tau}{N_0} .
 \end{aligned} \tag{2.31}$$

Setzt man für den Zähler von (2.31) die Definition aus (2.24) ein, so ergibt sich schließlich

$$\frac{S}{N} = \frac{E_s}{N_0} = \text{ld}(M) \frac{E_b}{N_0} . \tag{2.32}$$

Die vorangegangenen Betrachtungen gestatten es, ein Datenübertragungssystem vollständig durch ein Symboltaktmodell im äquivalenten komplexen Tiefpaßbereich zu beschreiben; in dieser Form werden die Simulationen in Abschnitt 4 durchgeführt. Dabei wird von einer idealen Erfüllung der 1. Nyquist- und der Matched-Filter-Bedingung ausgegangen. Die analytischen Ausdrücke für die Symbol- und Bitfehlerwahrscheinlichkeiten verschiedener linearer Modulationsformen werden im folgenden zusammengestellt. Die Herleitung der folgenden Beziehungen ist im Abschnitt 11.4 des Lehrbuchs *Nachrichtenübertragung* zu finden.

BPSK-Übertragung

kohärente Demodulation $P_s = P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$

DBPSK-Übertragung

kohärente Demodulation $P_s = P_b = \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$

inkohärente Demodulation $P_s = P_b = \frac{1}{2} e^{-\frac{E_b}{N_0}}$

QPSK-Übertragung

kohärente Demodulation $P_s = \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) - \frac{1}{4} \left[\operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \right]^2$
 $P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (\text{Gray-Codierung})$

DQPSK-Übertragung

kohärente Demodulation $P_s \approx 2 \cdot \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$
 $P_b \approx \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \quad (\text{Gray-Codierung})$
 inkohärente Demodulation $P_s \approx \operatorname{erfc}\left(\sqrt{0.59 \cdot \frac{E_b}{N_0}}\right) \quad (2.3\text{dB-Verlust})$
 $P_b \approx \frac{1}{2} \operatorname{erfc}\left(\sqrt{0.59 \cdot \frac{E_b}{N_0}}\right) \quad (\text{Gray-Codierung})$

M-PSK-Übertragung

kohärente Demodulation $P_s \approx \operatorname{erfc}\left(\sqrt{\operatorname{ld}(M) \frac{E_b}{N_0}} \cdot \sin \frac{\pi}{M}\right)$
 $P_b \approx \frac{1}{\operatorname{ld}(M)} \operatorname{erfc}\left(\sqrt{\operatorname{ld}(M) \frac{E_b}{N_0}} \cdot \sin \frac{\pi}{M}\right) \quad (\text{Gray-Codierung})$

Kapitel 3

Übungsaufgaben

Die folgenden Übungsaufgaben sollten Sie **vor** dem Versuchstermin bearbeiten. Benutzen Sie dazu auch die angegebenen Kapitel des Lehrbuchs *Nachrichtenübertragung*.

Aufgabe 1: Cosinus-roll-off-Impuls im Zeitbereich (Kap. 9.1)

Diskutieren Sie die Zeitbereichsformulierung (2.6) eines Cosinus-roll-off-Impulses mit $r = 1$. Bestimmen Sie die Nulldurchgangszeitpunkte (führen Sie gegebenenfalls eine Grenzwertbetrachtung durch). Erläutern Sie, welche Terme in (2.6) die erste und welche die zweite Nyquist-Bedingung sicherstellen.

Aufgabe 2: Differentielle Demodulation (Kap. 10.2)

Erläutern Sie die Wirkungsweise der in Bild 2.5a angegebenen Schaltung zur Differenzphasen-Bestimmung. Welche Auswirkungen hat eine einzelne Symbol-Fehlentscheidung (z.B. infolge von überlagertem Rauschen) nach der Differenz-Decodierung?

Aufgabe 3: Demodulation (Kap. 10)

Erläutern Sie mit eigenen Worten die Begriffe „*differentiell*“ und „*kohärent*“ im Zusammenhang mit digitaler Nachrichtenübertragung.

Welche Bitfehlerrate ergibt sich bei kohärenter Demodulation von a) QPSK-, b) DQPSK-Signalen, wenn die Trägerphasenregelung um $\pi/2$ versetzt einrastet.

Aufgabe 4: S/N -Verhältnis im Basis- und Bandpaßbereich (Kap. 11)

Über einen AWGN-Kanal wird eine QPSK-Übertragung ($d(i) \in \{(1, j); (-1, j); (-1, -j); (1, -j)\}$) durchgeführt, wobei eine rechteckförmige Impulsformung mit

$$g(t) = \begin{cases} \frac{1}{T} & \text{für } -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{sonst} \end{cases} \quad (3.1)$$

vorgenommen wird. Auf dem Bandpaß-Übertragungskanal wird weißes gaußverteiltes Rauschen mit der spektralen Leistungsdichte $N_0/2$ überlagert.

Am Ausgang des Integrate-and-dump-Filters im Empfänger wird ein S/N -Verhältnis von 6dB gemessen. Bestimmen Sie die spektrale Leistungsdichte $N_0/2$ im Bandpaß-Bereich.

Aufgabe 5: Symbolfehlerwahrscheinlichkeit bei PSK (Kap. 11.4)

Überprüfen Sie die allgemeine Näherung für die Symbolfehlerwahrscheinlichkeit bei M -stufiger PSK-Übertragung für $M = 2$ und $M = 4$ und erläutern Sie anhand der benutzten Modell-Vereinfachung die Abweichungen von der exakten Lösung (Tabelle auf Seite 15). Benutzen Sie für die Erklärung Bild 11.4.8 des Lehrbuchs *Nachrichtenübertragung*.

Aufgabe 6: Rauschanpassung — Mismatching (Kap. 11)

Es wird eine Datenübertragung (Symbolrate $1/T$) über einen verzerrungsfreien Kanal mit dem rechteckförmigen Sendefilter

$$g(t) = \begin{cases} \frac{1}{T} & 0 \leq t \leq T \\ 0 & \text{sonst} \end{cases} \quad (3.2)$$

durchgeführt. Bestimmen Sie die Rauschanpassung (in Form eines S/N -Verlusts γ^2), wenn das Empfangsfilter

$$h(t) = \begin{cases} \frac{K}{T} & 0 \leq t \leq T_h \\ 0 & \text{sonst} \end{cases} \quad (3.3)$$

$$\text{mit } T_h < T \quad (3.4)$$

nicht die Matched-Filterbedingung erfüllt.

Kapitel 4

Versuchsdurchführung

In diesem Laborversuch wird Ihnen ein Einblick in die Simulationspraktiken digitaler Übertragungssysteme gegeben. Als Einführung sollen unter Abschnitt 4.1 verschiedene Impulsantworten der Sende- und Empfangsfilter eines Übertragungssystems erzeugt und bzgl. der Nyquist-Kriterien beurteilt werden. Unter Abschnitt 4.2 wird ein Symboltaktmodell zur Simulation von Bitfehlerraten eingeführt.

Zur Durchführung der einzelnen Versuchspunkte stehen die folgenden Routinen zur Verfügung. Die notwendigen Informationen bezüglich der entsprechenden Eingabesyntax und den dazu erforderlichen Parametern können Sie mit der `help`-Funktion abrufen. Ansonsten dürfen Sie natürlich alle internen MATLAB-Funktionen verwenden.

Abschnitt 4.1	
<code>cosroll</code>	Erzeugung eines Cosinus-roll-off-Impulses
<code>wurzc</code>	Erzeugung eines Wurzel-Cosinus-roll-off-Impulses
<code>symtrunc</code>	Symmetrische zeitliche Verkürzung eines Signals
Abschnitt 4.2	
<code>mapping</code>	Signalraumzuordnung
<code>demapping</code>	Harte Bitentscheidung

4.1 Impulsformer

Die darzustellenden Kurven sollen pro Teilaufgabe jeweils in ein Bild gezeichnet und anschließend ausgedruckt werden.

4.1.1 Entwurf von Cosinus-roll-off-Filtern im Zeitbereich

Erzeugen Sie drei Impulsformer-Filter mit den Roll-off-Faktoren $r_1 = 1$, $r_2 = 0.5$ und $r_3 = 0.2$. Die Beobachtungslänge beträgt jeweils $L = 8$ Symbolintervalle, bei $w = 8$ Abtastwerten pro

Symbol. Erstellen Sie die grafische Darstellung der Filterimpulsantworten durch die Eingabe des folgenden Befehls:

`exercise_1_1`

Bearbeiten Sie dazu folgende Punkte:

- **1.1** - Überprüfen Sie die Gültigkeit der 1. Nyquist-Bedingung im Zeitbereich.
- **1.2** - Welche Aussage läßt sich pauschal über die Abklingeigenschaften in Abhängigkeit vom Roll-off-Faktor treffen ?

4.1.2 Entwurf von Wurzel-Cosinus-roll-off-Filtern

Entwerfen Sie zwei Wurzel-Cosinus-roll-off-Filter mit den Roll-off-Faktoren $r_2 = 0.5$ und $r_3 = 0.2$. Erzeugen Sie die normierten Impulsantworten mit folgender Eingabe:

`exercise_1_2`

- **1.3** - Überprüfen Sie auch hier die Gültigkeit der 1. Nyquist-Bedingung.

4.1.3 Gesamtimpulsantwort

Zur Erfüllung der Matched-Filter-Bedingung befindet sich in einem Datenübertragungssystem am Sender und am Empfänger jeweils ein Wurzel-Cosinus-roll-off-Filter. Erzeugen Sie für die beiden Roll-off-Faktoren $r_2 = 0.5$ und $r_3 = 0.2$ die Gesamtimpulsantworten (bei idealem Kanal) durch Faltung von zwei Wurzel-Cosinus-Impulsantworten.

`exercise_1_3`

- **1.4** - Überprüfen Sie die Erfüllung der 1. Nyquist-Bedingung in beiden Fällen.
- **1.5** - Erklären Sie etwaige Abweichungen von der 1. Nyquistbedingung.

4.1.4 Spektren von Cosinus-roll-off-Impulsen

Erzeugen Sie für die drei unter 4.1.1 erzeugten Cosinus-roll-off-Impulse die Betragsspektren und stellen Sie sie in einem Diagramm dar.

`exercise_1_4`

- **1.6** - Erklären Sie etwaige Abweichungen gegenüber einem idealen Cosinus-roll-off-Spektrum.
- **1.7** - Überprüfen Sie die 1. Nyquist-Bedingung im Spektralbereich. Ist sie in allen drei Fällen erfüllt?
- **1.8** - Welche Maßnahmen müßten zur Verbesserung der Entwürfe ergriffen werden?

4.1.5 Spektren eines Wurzel-Cosinus-roll-off-Impulses, des Gesamtimpulses und des zugehörigen Cosinus-roll-off-Impulses

Erzeugen Sie das Betragsspektrum des unter 4.1.2 entworfenen Wurzel-Cosinus-roll-off-Impulses mit $r_2 = 0.5$ und stellen Sie es graphisch dar. Bilden Sie das Betragsquadrat des Spektrums und stellen Sie es ebenfalls graphisch dar. Tragen Sie diese beiden Funktionen zusammen mit dem oben ermittelten Betragsspektrum eines Cosinus-roll-off-Impulses ($r_2 = 0.5$) in ein gemeinsames Diagramm ein.

`exercise_1_5`

- **1.9** - Vergleichen Sie die drei Spektren und erläutern Sie Ihre Beobachtungen. Stellen Sie eine Verbindung zu Ihren Untersuchungen unter Punkt 4.1.2 her.

4.2 Simulation eines Symboltaktmodells

Für die weiteren Untersuchungen in diesem Abschnitt werden aus dem vorherigen Abschnitt 4.1 keine Ergebnisse benötigt. Das Ziel der folgenden Übung ist es, ein paar grundlegende Techniken zur Simulation von Bitfehlerraten zu erlernen.

Beim Symboltaktmodell wird jedem Sendesymbol genau 1 Abtastwert zugeordnet. Alle analogen Komponenten seien dabei ideal. Es wird lediglich AWGN zugelassen¹. Für Sende- und Empfangsfilter bedeutet dies die gemeinsame Erfüllung der 1. Nyquistbedingung und eine optimale Rauschanpassung.

In den folgenden Teilaufgaben findet schrittweise der Entwurf einer Simulationskette statt. Erzeugen Sie **für jede Teilaufgabe eine neue Datei** und **kopieren Sie die benötigten Zeilen der jeweils vorangegangenen Teilaufgabe hinein**. Im Versuchsbeschreibungstext werden jeweils nur die Zeilen aufgeführt, die sich gegenüber der jeweils vorangegangenen Teilaufgabe geändert haben.

4.2.1 Signalraumzuordnung

Erzeugen Sie zunächst einen Datenvektor, der aus `n_bit` binären Daten besteht. Die Länge sollte für spätere Benutzung dieser Programmzeilen variabel gehalten werden:

```
n_bit = 1200;
s_bit = round(rand(n_bit,1));
```

Die Signalraumzuordnung kann mit Hilfe der Funktion `mapping.m` geschehen, wobei noch der Modulationstyp festzulegen ist.

```
m_typ = 2;
s_sym = mapping(s_bit,m_typ);
```

Die Funktion `mapping.m` erlaubt insgesamt 7 verschiedene Modulationstypen, wobei der Typ 2 eine Gray-codierte $\pi/4$ -QPSK vorgibt.

- **2.1** - Finden Sie heraus, welche Signallräume durch die Typen 1,3,4,5,6 und 8 eingestellt werden, und stellen Sie jeweils den entstehenden Signalraum graphisch dar.
Hinweis: Erhöhen Sie gegebenenfalls die Anzahl der Daten.

```
figure(1);
subplot(231);
m_typ = 1;
s_sym = mapping(s_bit,m_typ);
plot(s_sym,'o');
axis('square'); axis([-1.5 1.5 -1.5 1.5]);
title('xPSK');
%
```

¹Bei Symboltaktsimulationen können natürlich auch noch Störungen durch Kanalmodelle simuliert werden. Da in diesem Laborversuch keine Entzerrung stattfindet, sei angenommen, daß ein idealer Kanal vorliegt.

```
subplot(232);
m_typ = 3;
s_sym = mapping(s_bit,m_typ);
plot(s_sym,'o');
axis('square'); axis([-1.5 1.5 -1.5 1.5]);
title('xPSK');
... usw
```

Für die Darstellung der Signlräume sollte evtl. ein neues (separates) Programm erzeugt werden, denn die nachfolgenden Aufgaben benötigen diese Programmzeilen nicht mehr. Im rauschfreien Fall sind die empfangenen Symbole gleich den gesendeten Symbolen.

```
r_sym = s_sym;
r_bit = demapping(r_sym,m_typ);
```

Mit Hilfe der Routine `demapping.m` können die empfangenen Symbole entschieden und decodiert werden, so daß der Vektor `r_bit` eigentlich den ausgewürfelten Quelldaten `s_bit` entsprechen müsste. Die bisher erzeugte Übertragungskette sollte folgendermaßen aussehen:

```
n_bit = 1200;
m_typ = 2;

s_bit = round(rand(n_bit,1));
s_sym = mapping(s_bit,m_typ);

r_sym = s_sym;

r_bit = demapping(r_sym,m_typ);
```

4.2.2 Erzeugung eines Rauschsignals

Zur Überlagerung des Sendesignals mit AWGN muß zunächst ein komplexer Rauschvektor generiert werden. Die Schwierigkeit liegt dabei in dessen richtiger Skalierung. Angenommen, es sei ein E_b/N_0 -Verhältnis von 7 dB vorgegeben. Mit Hilfe der Bits pro Symbol ($\text{ld}(M)=2$) und der mittleren Sendeleistung $\sigma_d^2 = 1$ kann man nun die mittlere Leistung des Rauschvektors berechnen.

```
eb_n0 = 7;
ldM = 2;

sigma = ???

rausch = (randn(length(s_sym),1)+j*randn(length(s_sym),1))*(sigma/sqrt(2));

r_sym = s_sym + rausch;
```

- **2.2** - Bestimmen Sie $\sigma := \sigma_n$ im Programm und geben Sie σ_n^2 für die vorgegebenen Einstellungen an.
- **2.3** - Warum ist hier noch eine Normierung des Rauschvektors mit $1/\sqrt{2}$ erforderlich?
- **2.4** - Stellen Sie außerdem den gestörten Signalraum graphisch dar.

Hinweis: Zur Bestimmung von σ ist der **lineare** S/N -Wert von Bedeutung.

4.2.3 Aufnehmen einer Bitfehlerkurve

Ein wichtiger Punkt bei der Simulation einer Bitfehlerkurve ist die Bestimmung der Bitfehlerrate. Dabei ist ein Vergleich zwischen gesendeten und entschiedenen Daten erforderlich.

```
d_bit = abs(s_bit-r_bit);
ber    = mean(d_bit);
```

- **2.5** - Bestimmen Sie nun simulativ die Bitfehlerrate mit dem bisher eingestellten E_b/N_0 -Wert von 7 dB.
- **2.6** - Ermitteln Sie die theoretischen Werte für die Bitfehlerwahrscheinlichkeit unter der Annahme, daß Symbolfehler **nicht** auf diagonal gegenüberliegende Signalpunkte führen; nehmen Sie hierfür eine Gray-Codierung an.
- **2.7** - Vergleichen Sie den theoretischen Wert mit den Ergebnissen der Simulationen.

Für die Darstellung einer Bitfehlerkurve sind Simulationen bei mehreren E_b/N_0 -Werten erforderlich. Dazu soll um das bisherige Programm eine Schleife gelegt werden:

```
v_eb_n0 = [0:1:8];
n_bit   = 1200;
m_typ   = 2;
ldM     = 2;

v_ber   = [];
for eb_n0 = v_eb_n0
    s_bit = round(rand(n_bit,1));
    s_sym = mapping(s_bit,m_typ);

    sigma = ???
    rausch=(randn(length(s_sym),1)+j*randn(length(s_sym),1))*(sigma/sqrt(2));
    r_sym = s_sym+rausch;

    r_bit = demapping(r_sym,m_typ);
    d_bit = abs(s_bit-r_bit);
    ber    = mean(d_bit);
    v_ber = [v_ber ber];
```

```
end;

figure;
semilogy(v_eb_n0,v_ber);
grid;
xlabel('E_b/N_0');
ylabel('BER');
```

Die Funktion `semilogy` sorgt für eine logarithmische Darstellung der y -Achse.

- **2.8** - Stellen Sie die Bitfehlerkurve mit den oben angegebenen Einstellungen dar und diskutieren Sie ihre Aussagekraft.

Wenn Sie die Funktion des bisherigen Programms verstanden haben, überlegen Sie, wie man die Genauigkeit der Bitfehlerkurve auch für hohe E_b/N_0 -Werte verbessern kann.

Wenn Sie in dem obigen Programm einfach nur die Anzahl der zu übertragenden Bits erhöhen, treten zwei Probleme auf. Durch besonders große Vektoren steigt der Arbeitsspeicherbedarf der Simulation so stark an, daß die Recheneffizienz durch Überlastung des Rechners sehr gering wird². Außerdem würde man für kleine E_b/N_0 -Werte *mit Kanonen auf Spatzen schießen*, denn bei hoher Bitfehlerrate wären auch nur wenige Symbole zu simulieren.

Betrachten Sie nun das folgende Programm³, das eine Möglichkeit zur effizienten Bitfehlerraten-Simulation darstellt.

```
% -----
%
% Programm dm423.m
%
% -----
%
% Schwerpunktlabor Nachrichtentechnik
%
% Universitaet Bremen
%
% by Heiko Schmidt, 15.3.2001
%
% -----
%
% Bestimmung der Simulationsparameter
%
v_eb_n0 = [0:1:10];    % zu simulierende Eb/N0-Werte
n_bit   = 1200;        % Bits pro Schleife
m_typ   = 2;           % Modulationstyp (QPSK)
ldM     = 2;           % Bits pro Symbol
max_error = 100;       % maximale Fehleranzahl
max_loop = 1000;       % maximale Schleifendurchl{\a}ufe
```

²Durch Auslagerung (SWAP) auf die Festplatte erhöht sich die benötigte Rechenzeit dramatisch.

³Damit Sie es nicht neu eingeben müssen ist es auch unter dem Namen **dm423.m** auf der Festplatte gespeichert.

```

%
% -----
%
v_ber = [];          % Initialisierung
for eb_n0 = v_eb_n0  % Schleife {\u}ber alle Eb/N0-Werte
    vv_ber = [];      % Initialisierung
    loop = 1;         % Startwert f{\u}r innere Schleife
    n_error = 0;      % Fehlerz{\a}hler zur{\u}cksetzen
    while loop > 0     % Innere Schleife
        %
        s_bit = round(rand(n_bit,1));    % Quellbits erzeugen
        s_sym = mapping(s_bit,m_typ);    % Sendesymbole
        %
        % ----- Uebertragung von Symbolen -----
        %
        sigma = sqrt(10^(-eb_n0/10)/ldM); % Wurzel aus Rauschleistung
        rausch=(randn(length(s_sym),1)+j*randn(length(s_sym),1))*(sigma/sqrt(2));
        r_sym = s_sym+rausch;             % Rauschen addieren
        %
        % ----- Empfang von Symbolen -----
        %
        r_bit = demapping(r_sym,m_typ);

        d_bit = (abs(s_bit-r_bit));
        n_error = n_error + sum(d_bit);
        vv_ber = [vv_ber mean(d_bit)];
        %
        % ----- Abbruchkriterium pruefen -----
        %
        if (loop >= max_loop)|(n_error >= max_error)
            n_loop = loop;
            loop = 0;
        else
            loop = loop + 1;
        end;

    end;
    ber = mean(vv_ber);
    v_ber = [v_ber ber];
    % ----- Ausgabe von Zwischenergebnissen auf dem Bildschirm -----
    disp(sprintf('Eb/N0: %4.1f dB   BER: %5.1e   N_bit: %7d   N_error: %7d',...
        eb_n0,ber,n_loop*n_bit,n_error));
end;

figure;
semilogy(v_eb_n0,v_ber);
grid;
xlabel('E_b/N_0');

```

```
ylabel('BER');
% EOF
```

Diskutieren Sie folgende Punkte:

- **2.9** - Welche Bitfehlerrate ergibt sich bei $E_b/N_0 = 8$ dB ?
- **2.10** - Welche Aufgabe haben die Variablen `max_error` und `max_loop`?
- **2.11** - Welches Abbruchkriterium greift bei welchen E_b/N_0 -Werten?
- **2.12** - Beurteilen Sie qualitativ die Auswirkungen der Bitfehlerrate auf die Rechenzeit.

Machen Sie mit dem nächsten Abschnitt erst weiter, wenn Sie das zuletzt diskutierte Symboltakt-Simulationsmodell verstanden haben. Bei Fragen wenden Sie sich bitte an den Betreuer.

4.3 Simulation mit Impulsformung

Für diesen Versuchsabschnitt benötigen Sie den im ersten Teil eingeführten Impulsformer (`wurzc`) und das im letzten Abschnitt betrachtete Programm `dm423.m`. Ziel ist es, verschiedene Impulsformer bei der Bitfehlerraten-Simulation zu berücksichtigen. Doch zunächst stehen ein paar kleinere Übungen im Umgang mit „analogen“ Sendesignalen auf dem Programm.

4.3.1 Überabtasten mit verschiedenen Impulsformern

Beginnen Sie diesen Abschnitt mit einer neuen Programmdatei und generieren Sie sich einen zufälligen QPSK-Symbolvektor (Spaltenvektor):

```
n_bit = 1200;
m_typ = 2;
s_bit = round(rand(n_bit,1));
s_sym = mapping(s_bit,m_typ);
```

Das Überabtasten besteht aus zwei Teilen. Zuerst müssen zwischen den Abtastwerten Nullen eingefügt werden, um die Abtastrate zu erhöhen. Anschließend erfolgt eine Faltung mit einem Impulsformer. Je höher der Überabtastfaktor w gewählt wird, desto mehr nähert man sich bei der Simulation dem realen, analogen Signal, wobei auch der Rechenaufwand enorm ansteigt. Wir beschränken uns daher auf einen Überabtastfaktor $w = 8$.

Bild 4.1 zeigt eine Folge von Abtastwerten $x(k)$, zwischen denen Nullen ($w = 4$) eingefügt werden. Da es sich bei x um ein zeitdiskretes Signal handelt, ist das Spektrum im Ω -Bereich von $-\pi$ bis π vollständig beschreibbar. Nehmen Sie an, das Spektrum von x habe den ebenfalls in Bild 4.1 skizzierten Verlauf.

- **3.1** - Skizzieren Sie nun das Spektrum des diskreten Signals nach dem Einfügen von Nullen.

Erhöhen Sie nun die Abtastfrequenz des Symbolvektors `s_sym`, indem Sie Nullen einfügen.

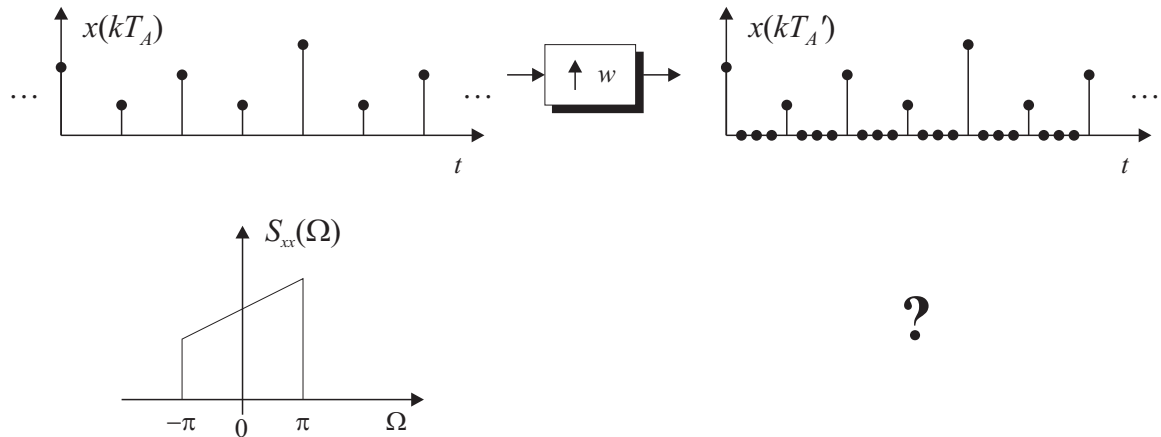


Bild 4.1: Einfügen von Nullen

```

w          = 8;
s_sym      = mapping(s_bit,m_typ);
n_sym      = length(s_sym);
s_0_mat    = zeros(w,n_sym);
s_0_mat(1,:) = s_sym.';
s_0_sig    = s_0_mat(:);

```

Der nächste Schritt umfaßt die Faltung mit dem Sendefilter $g(t)$, das wir hier nur zeitdiskret, w -fach überabgetastet realisieren. Für die folgende Auswertung sollen 4 verschiedene Impulsformer betrachtet werden, die im folgenden Text mit g_1 bis g_4 bezeichnet werden.

- g_1 : Wurzel-Cosinus-Roll-Off-Filter mit $r = 1$
- g_2 : Wurzel-Cosinus-Roll-Off-Filter mit $r = 0.25$
- g_3 : Rechteckfilter der Länge T
- g_4 : Rechteckfilter der Länge $T/2$

```

% ----- Definitionen -----
w          = 8;
l_wcos     = 8;
r1         = 1;
r2         = 0.25;
g_1        = wurzcos(r1,w,l_wcos);
g_2        = wurzcos(r2,w,l_wcos);
g_3        = ones(1,w);
g_4        = ones(1,round(w/2));

g          = g_1;
% ----- Faltung -----
s_sig      = conv(s_0_sig,g);
s_sig      = s_sig(:);      % Spaltenvektor erzwingen

```

4.3.2 Abtasten des Empfangssignals

Die nächste Aufgabe besteht darin, das empfangene Signal abzutasten um die empfangenen Symbole zu erhalten. Dafür sind wieder verschiedene Schritte notwendig. Zunächst findet eine Faltung des Empfangssignals `r_sig` mit einem Empfangsfilter `h` statt, das die Matched-Filter-Bedingung erfüllen soll. Anschließend erfolgt eine Synchronisation des Empfangssignals in Zeitrichtung. Den idealen Synchronisationszeitpunkt kann man durch Faltung von Sende- und Empfangsfilter ermitteln.

```
r_sig      = s_sig;
h          = g_1;
%
% ----- Faltung mit Empfangsfilter -----
r_0_sig    = conv(r_sig,h);
%
% ----- Abtastzeitpunkt ermitteln -----
g_all      = conv(g,h);
[g_max,max_pos] = max(abs(g_all));
%
% ----- Abtasten -----
r_sym      = r_0_sig(max_pos + w*[0:n_sym-1]') / g_max;
```

Damit liegen die empfangenen Symbole als Spaltenvektor vor und können wie im Symboltaktmodell ausgewertet werden.

- **3.2** - Welche Bedeutung haben `g_max` und `max_pos`?

4.3.3 Überlagerung des Kanalrauschens bei Überabtastung

Bei der Überlagerung mit dem Rauschvektor stellt auch hier wieder die richtige Skalierung das größte Problem dar. Gemäß Gleichung (2.22) hängt die Leistung des zu simulierenden Rauschens linear von w ab (bei konstanter Rauschleistungsdichte). Außerdem ist zu berücksichtigen, daß das Sendesignal durch die verschiedenen Impulsformer nicht normiert vorliegt.

```
sigma = sqrt(w * 10^(-eb_n0/10)/ldM); % Wurzel aus Rauschleistung
rausch=(randn(length(s_sig),1)+j*randn(length(s_sig),1))*(sigma/sqrt(2));
g_eff = sqrt(sum(abs(g).^2)/w);        % Normierung auf Sendefilter
r_sig = s_sig + (g_eff * rausch);      % Rauschen {"u"}berlagern
```

- **3.3** - Welche Bedeutung hat `g_eff` ?

Damit sind alle Komponenten für die Simulation komplett und können (im nächsten Abschnitt) in das Programm `dm423.m` eingearbeitet werden.

4.3.4 Bitfehlersimulation mit Überabtastung

Das komplette Simulationsprogramm ist unter dem Namen `dm434.m` zu finden.

```
% -----
%
% Programm dm434.m
%
% -----
%
% Schwerpunktlabor Nachrichtentechnik
%
% Univerit{"a}t Bremen
%
% by Heiko Schmidt, 15.3.2001
%
% -----
%
% Bestimmung der Simulationsparameter
%
v_eb_n0 = [0:1:10]; % zu simulierende Eb/N0-Werte
n_bit   = 1000;     % Bits pro Schleife
m_typ   = 2;        % Modulationstyp (QPSK)
ldM     = 2;        % Bits pro Symbol
max_error = 100;    % maximale Fehleranzahl
max_loop = 400;     % maximale Schleifendurchlaeufer
w       = 8;        % Ueberabtastfaktor
l_wcos  = 8;
r1      = 1;
r2      = 0.25;
g_1     = wurzcos(r1,w,l_wcos);
g_2     = wurzcos(r2,w,l_wcos);
g_3     = ones(1,w);
g_4     = ones(1,round(w/2));
%
g       = g_1;      % Sendefilter festlegen
h       = g_1;      % Empfangsfilter festlegen
%
% -----
%
v_ber   = [];       % Initialisierung
for eb_n0 = v_eb_n0 % Schleife ueber alle Eb/N0-Werte
    vv_ber = [];    % Initialisierung
    loop   = 1;     % Startwert fuer innere Schleife
    n_error = 0;    % Fehlerzaehler zuruecksetzen
    while loop > 0 % Innere Schleife
        %
```

```

s_bit = round(rand(n_bit,1));      % Quellbits erzeugen
s_sym = mapping(s_bit,m_typ);      % Sendesymbole
%
% ----- Uebertragung von Symbolen -----
%
n_sym      = length(s_sym);
s_0_mat    = zeros(w,n_sym);
s_0_mat(1,:) = s_sym.';
s_0_sig    = s_0_mat(:);
%
s_sig      = conv(s_0_sig,g);
s_sig      = s_sig(:);              % Spaltenvektor erzwingen
%
sigma = sqrt(w * 10^(-eb_n0/10)/ldM); % Wurzel aus Rauschleistung
rausch=(randn(length(s_sig),1)+j*randn(length(s_sig),1))*(sigma/sqrt(2));
g_eff = sqrt(sum(abs(g).^2)/w);      % Normierung auf Sendefilter
r_sig = s_sig + (g_eff * rausch) ;    % Rauschen ueberlagern
%
% ----- Faltung mit Empfangsfilter -----
r_0_sig    = conv(r_sig,h);
%
% ----- Abtastzeitpunkt ermitteln -----
g_all      = conv(g,h);
[g_max,max_pos] = max(abs(g_all));
%
% ----- Abtasten -----
r_sym      = r_0_sig(max_pos + w*[0:n_sym-1]') / g_max;
%
% ----- Empfang von Symbolen -----
%
r_bit = demapping(r_sym,m_typ);

d_bit = (abs(s_bit-r_bit));
n_error = n_error + sum(d_bit);
vv_ber = [vv_ber mean(d_bit)];
%
% ----- Abbruchkriterium pruefen -----
%
if (loop >= max_loop)|(n_error >= max_error)
    n_loop = loop;
    loop = 0;
else
    loop = loop + 1;
end;

end;
ber = mean(vv_ber);
v_ber = [v_ber ber];

```

```

% ----- Ausgabe von Zwischenergebnissen auf dem Bildschirm -----
disp(sprintf('Eb/N0: %4.1f dB    BER: %5.1e    N_bit: %7d    N_error: %7d',...
            eb_n0,ber,n_loop*n_bit,n_error));
end;

figure;
semilogy(v_eb_n0,v_ber);
grid;
xlabel('E_b/N_0');
ylabel('BER');
% EOF

```

- **3.4** - Rufen Sie es mehrfach auf und weisen Sie dabei den Sende- und Empfangsfiltern (g und h) jeweils die gleiche Impulsantwort $g_1 - g_4$ zu. Vergleichen Sie die simulierten Bitfehlerraten mit der Bitfehlerkurve des Symboltaktmodells.
- **3.5** - Führen Sie die Simulation erneut durch, wobei Sie nun Sende- und Empfangsfiltern unterschiedliche Impulsantworten zuweisen. Betrachten Sie die beiden Fälle $g=g_1$; $h=g_2$; und $g=g_3$; $h=g_4$;
- **3.6** - Ermitteln anhand der simulierten Bitfehlerkurven den E_b/N_0 -Verlust gegenüber der Bitfehlerkurve des Symboltaktmodells.
- **3.7** - Vergleichen Sie den Verlust im zweiten Fall mit den theoretischen Ergebnissen der Vorbereitungsaufgabe 6.

4.3.5 Auswirkungen eines Synchronisationsfehlers

Der korrekte Abtastzeitpunkt ist bei Einträgersverfahren besonders wichtig. Fügen Sie in das bestehende Programm eine Möglichkeit zur Simulation eines Offsets ein, der durch den Parameter

```
sync_offset = 0;
```

eingestellt werden soll. Als Sende- und Empfangsfilter kommen zunächst Wurzel-Cosinus-Roll-Off-Filter mit $r = 1$ zum Einsatz. Ergänzen Sie außerdem folgende Zeile:

```

% ----- Abtasten -----
r_sym      = r_0_sig(max_pos + sync_offset + w*[0:n_sym-1]') / g_max;

```

Da es sich bei r_0_sig um ein w -fach überabgetastetes, zeitdiskretes Signal handelt, sind natürlich auch nur Verschiebungen um T/w oder Vielfache davon möglich.

- **3.8** - Geben Sie an, welche Einstellungen für `sync_offset` sinnvoll sind.
- **3.9** - Ermitteln Sie jeweils die Bitfehlerraten bei $E_b/N_0=7$ dB und stellen Sie diese in einer Kurve dar.
- **3.10** - Deuten Sie die Ergebnisse.

- **3.11** - Wiederholen Sie abschließend die vorangegangene Untersuchung mit Wurzel-Cosinus-Roll-Off-Filtern mit $r = 0.25$.