

## Übung 4: Modulationsverfahren, Forward error correction

2019-11-07

### 1. Simulation von Bitfehlerraten bei verschiedenen Modulationstechniken

Wir nutzen die Einsichten aus der vorherigen Programmieraufgabe zu Rauschen im Phasendiagramm. Implementieren Sie eine Simulation von Basisband- und Breitbandübertragungsverfahren über AWGN-Kanäle. Es reicht dazu aus, die zu übertragenden Symbole (!, also nicht die waveform) zufällig zu wählen und mit geeignetem Rauschen zu versehen.

Plotten Sie die resultierende Bitfehlerrate für variierende Signal-Rausch-Verhältnisse.

Sie können dazu das Notebook `BER-Basisband.ipynb` bzw. das entsprechende PDF heranziehen.

### 2. Hamming-Codierung

Ein wichtiger Aspekt bei der Konstruktion von fehlererkennenden und fehlerkorrigierenden Codes ist die minimale Hamming-Distanz  $d_{\min}$  zwischen den unterschiedlichen Datenworten eines Codes.

- Finden Sie den minimalen Hamming-Distanz für folgenden Code  $C = \{0000000, 1110100, 1101011, 1111111\}$ . Wie viele Fehler kann  $C$  erkennen und wie viele kann er korrigieren? Wie könnte man  $C$  verbessern?
- Finden Sie mit Hilfe der naheliegenden Dekodierung heraus, welches Codewort aus dem Code Book  $C = \{0000, 1110, 1011\}$  gesendet wurde, wenn man 0110 empfängt.
- Angenommen, dass die Bitfehlerrate eines binär-symmetrischen Kanals<sup>1</sup> 0.01 ist, wie hoch ist die Wahrscheinlichkeit dafür, dass die Bitreihe 100011 empfangen wird, wenn man 000000 sendet?

<sup>1</sup>Binary symmetric channel, BSC, alle Bits werden mit fester Wahrscheinlichkeit invertiert oder nicht; Verhalten für alle Bits unabhängig von einander, Verhalten für 0 und 1 identisch.

**Lösung:**

- (a) Der minimale Abstand ist  $d(c_3, c_4) = 2$ . Für die Erkennung gilt folgendes:  $d(C) \geq k + 1 \Rightarrow k \leq 1$ , wobei  $k$  die Anzahl der Fehlern ist. Basierend darauf, kann man bis zu 1-bit Fehlern erkennen. Für die Korrektur gilt folgendes:  $d(C) \geq 2 \cdot k + 1 \Rightarrow k \leq \frac{1}{2}$ . Man kann also allgemein keine Fehler korrigieren, da  $\lfloor \frac{1}{2} \rfloor = 0$ . Man kann  $C$  verbessern, indem man beispielsweise  $c_3 = 1001011$  setzt. Dann gilt:  $d(C) = 3$ . Damit kann man dann die Erkennung von 2-bit Fehlern und die Korrektur von 1-bit Fehlern garantieren.
- (b) Man muss den minimalen Abstand zwischen den empfangenen Codewort und den erlaubten Codesatz berechnen. Der minimale Abstand beträgt bei dem zweiten Codewort:  $d(1110, 0110) = 1$ . D.h. das gesendete Codewort ist: 1110.  
Anmerkung: Trotz  $d(C) = 2$  ist also die Fehlerkorrektur im Einzelfall möglich. Wäre jedoch z.B. 1111 empfangen worden, so wäre eine Korrektur mit Hilfe der Nächster-Nachbar-Dekodierung nicht möglich gewesen.
- (c) Die Wahrscheinlichkeit von einem Bitfehler = 0.01 und von einer korrekten Übertragung =  $1 - 0.01 = 0.99$ . Die Gesamtwahrscheinlichkeit für die empfangene Bitreihe ist  $= (0.01)^3 (0.99)^3 \approx 9.7 \cdot 10^{-7}$ .

**3. Vorwärtsfehlerkorrektur: Fehlerrate und Durchsatz**

In der Vorlesung haben wir besprochen, wie sich unterschiedliche Code raten auf Paketfehlerrate und Durchsatz auswirken. Implementieren Sie hierfür eine Simulation! Orientieren Sie sich dazu an dem Notebook BCH.ipynb oder folgen Sie der unten stehenden Anleitung.

Nutzen Sie dazu (z.B.) eine Blocklänge von 127 Bits und BCH-Codes, die von 0 bis zu 10 Fehlern in einem Block korrigieren können. Sie finden die Anzahl nutzbarer Bits pro Block in solchen Codes in den Vorlesungsfolien.

- Nehmen Sie zunächst einen AWGN-Kanal mit variabler SNR an. Nutzen Sie BPSK und die Abschätzung für die BER für gegebenes SNR aus den Folien. Gehen Sie wie folgt vor:
  - Entwerfen Sie eine Funktion, die für ein gegebenes SNR und eine gegebene Paketlänge eine zufällige Anzahl Bitfehler bestimmt, die in diesem Paket vorkommen. (So gewählt, dass die richtige Verteilung an Anzahl Bitfehlern entsteht.)
  - Entwerfen Sie eine Funktion, die für einen gegebenen Code, eine gegebene Paketlänge und eine gegebene Anzahl Bitfehlern entscheidet, ob das Paket korrekt übertragen wurde oder nicht.

- Nutzen Sie diese Funktionen, um für viele Pakete zu bestimmen, ob sie korrekt oder inkorrekt übertragen wurden (für einen gegebenen Code und eine gegebene SNR)
- Erstellen Sie daraus einen Plot, indem Sie für alle Codes und ein SNR-Intervall (z.B. von -5 dB bis 10 dB) die entstehenden Paketfehlerraten zeigen
- Bestimmen Sie aus den Paketfehlerraten den erzielten Durchsatz. Achten Sie auf die Anzahl Nutzbits der unterschiedlichen Codes! Plotten Sie auch dies.