

# Labor zur Vorlesung Kommunikationstechnik

## Laborübung 5 (BCH und CRC)

Prof. Dr. Dirk Staehle

Die Abgabe erfolgt durch Hochladen der bearbeiteten Word-Datei in Moodle.

### **Bearbeitung in Zweier-Teams**

**Team-Mitglied 1:**

**Team-Mitglied 2:**

## 1 Einleitung

In dieser Aufgabe soll die Übertragung eines Frames simuliert werden. Die Übertragung soll durch einen BCH Codes zur Fehlerkorrektur und einen CRC Code zur Fehlererkennung gesichert werden.

Angelehnt an den WLAN Standard soll der Frames aus einem 30 Byte großen Header, einer Payload von 166 Bytes und einer Frame Check Sequence (FCS, CRC) von 4 Bytes bestehen.

Zunächst soll die Übertragung nur mit CRC und ohne BCH Code simuliert werden, um die Grenzen der Fehlererkennung festzustellen.

Im zweiten Schritt soll ein geeigneter BCH Code für die Übertragungsstrecke ausgewählt werden.

Im dritten Schritt soll die Übertragung mit BCH und CRC simuliert werden, um festzustellen, ob der BCH-Code die Zielvorgaben erfüllt und ob der CRC-Code in der Lage ist, die verbleibenden Fehler zu detektieren.

Machen Sie sich mit den Objekten und Funktionen zu BCH-Codes (unter PHY Components/Error Detection and Correction/Block Coding/BCH Codes) und CRC-Codierung (unter PHY Components/Error Detection and Correction/Block Coding/Linear Block Codes) in der Hilfe der Communication Systems Toolbox vertraut. Ich empfehle Ihnen zur Implementierung der Aufgabe, die in der Toolbox angebotenen Objekte für CRC und BCH zu verwenden, d.h. CRCGenerator/Detector und BCHEncode/Decoder.

## 2 CRC

Führen Sie in Matlab eine Simulationsstudie aus, um die Qualität der Fehlererkennung mit CRC zu validieren. Schreiben Sie dazu eine Funktion, die

1. M zufällige Nachrichten mit einer Länge von k Bits generiert
2. Für diese Nachrichten mit dem Generatorpolynom G die CRC-Bits bestimmt
3. pro Nachricht f Bitfehler verursacht (randerr)
4. die fehlerhafte Nachricht auf Fehler überprüft
5. die Anzahl der nicht-gefundenen fehlerhaften Nachrichten zurückgibt

Funktionsargumente: M, k, G, f

Rückgabewert: Anzahl der nicht-gefundenen fehlerhaften Nachrichten

Stellen Sie in einer Grafik den Zusammenhang zwischen der Anzahl Bitfehler f und dem Prozentsatz nicht-gefundener Übertragungsfehler für das WLAN Generatorpolynom (siehe Skript) und das USB Generatorpolynom  $G(D) = D^5 + D^2 + D + 1$  dar. Wählen Sie dazu eine Payload von 8 Bytes und betrachten Sie bis zu 30 Fehler pro Frame. Da die Nicht-Erkennung von Fehlern generell selten vorkommt, wählen Sie M=50000. Inwiefern hängt das Ergebnis von

der Länge der Payload ab? Vergleich Sie dazu die Ergebnisse mit einer Payload von 1 und 8 Bytes.

Hinweis: Nutzen Sie für die Implementierung die Möglichkeit, die Anzahl CRC-Checks pro Frame in den `comm.CRCGenerator` und `comm.CRCDetector` Objekten auf  $M$  zu setzen. Damit wird die Gesamtnachricht aus  $M \cdot k$  Bits in  $M$  Nachrichten aus  $k$  Bits zerlegt und an jede Nachricht die CRC Bits angehängt.

### 3 BCH Code

BCH Codes sind eine Familie von zyklischen Codes, die bei einer Codewortlänge von  $n = 2^m - 1$  Codes für verschiedene Kombinationen von korrigierbaren Fehlern  $t$  und Nutzbits  $k$  bereitstellt. Werden nicht alle Nutzbits genutzt, so handelt es sich um einen verkürzten BCH Code. Bestimmen Sie zunächst geeignete Parameter  $n$  und  $t$  für eine Framegröße von  $k = 500$  Bytes und einer Übertragungstrecke mit einer Bitfehlerwahrscheinlichkeit von  $p = 0,5\%$ . Die Framefehlerrate soll höchstens  $1\%$  betragen.

1. Die Anzahl Bitfehler in einer Nachricht folgt einer Binomialverteilung mit Parametern „Anzahl Versuche“ gleich  $k$  und „Erfolgswahrscheinlichkeit“ gleich  $p$ . Bestimmen Sie wie in der Vorlesung beschrieben mit der Matlab-Funktion `binoinv` die Parameter  $m$  und  $t$  des BCH-Codes, so dass die in der Vorlesung formulierten Bedingungen für das Vorhandensein eines Codes erfüllt werden.
2. Bestimmen Sie die minimale Codewortlänge  $n$ , für die ein BCH-Code existiert, der bei  $k$  Nutzbits mindestens  $t$  Bitfehler sicher korrigiert.

Hinweis: Vergleichen Sie die in der Vorlesung aufgestellten Bedingungen mit den Werten, die die Matlab-Funktion `bchnumerr(n)` liefert.

3. Implementieren Sie eine Simulation, die eine Übertragung mit dem ermittelten BCH-Code über die definierte Übertragungstrecke durchführt. Validieren Sie, ob die resultierende Framefehlerrate eingehalten wird.

Hinweis: Nutzen Sie dazu die Objekte `comm.BCHEncoder` und `comm.BCHDecoder`. Die Funktion `bchgenpoly(n, k)` liefert ein Generatorpolynom. Achten Sie darauf, dass hier nur Werte für  $k$  akzeptiert werden, die `bchnumerr(n)` liefert. Verkürzte Codeworte können Sie realisieren, indem Sie den Parameter „ShortMessageLength“ des Encoder- bzw. Decoder-Objekts auf einen Wert kleiner als  $k$  setzen, siehe auch die Matlab-Hilfe des Decoders.

### 4 BCH Code und CRC

Kombinieren Sie die beiden Aufgaben dieser Laborübung, indem Sie bei der Codierung zunächst eine CRC-Prüfsumme hinzufügen und dann Nutzbits und CRC-Prüfbits mit einem BCH Code versehen. Bestimmen Sie die Framefehlerrate und die Restfehlerwahrscheinlichkeit für eine Übertragung von 10000 zufälligen Payloads und Fehlermustern. Verwenden Sie dazu das WLAN Generator Polynom aus Aufgabe 2 und die übrigen Parameter aus Aufgabe 3. Die Nachrichtenlänge beträgt also 500 Bytes abzüglich der CRC-Bits und die Bitfehlerwahrscheinlichkeit beträgt  $0,5\%$ .