



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG (HTWG)
Fakultät Informatik
Rechner- und Kommunikationsnetze
Prof. Dr. Dirk Staehle

Labor zur Vorlesung Kommunikationstechnik

Laborübung 4 (Linear-systematische Blockcodes)

Prof. Dr. Dirk Staehle
Daniel Scherz (M.Sc.)

Durchführung: 25. April 2016

1 Einleitung

In der Vorlesung wurde die Fehlererkennung und Fehlerkorrektur mit Hilfe von linear-systematischen Blockcodes vorgestellt. In dieser Laborübung soll die Anwendung von Kanalcodierung in Matlab umgesetzt werden. Eine erste Simulation einer digitalen Übertragungsstrecke soll implementiert und erste Simulationen durchgeführt werden.

2 Vorbereitung

Machen Sie sich mit den Funktionen zur Kanalcodierung mit linear-systematischen Blockcodes (encode, decode, hamngen, syndtable) sowie grundlegenden Funktionen zur Erzeugung, Manipulation und Auswertung von Bit- und Fehlervektoren (bsc, randsrc, randerr, biterr, rem, etc.) vertraut.

3 Vergleich mit Theorieübung und Vorlesung

Implementieren Sie ohne Nutzung der Communication Systems Toolbox ein Matlab-Skript zur

1. Berechnung eines Codeworts bei gegebener Generatormatrix
2. Berechnung der Parity-Check-Matrix bei gegebener Generatormatrix
3. Berechnung des Fehlersyndroms bei gegebenem empfangenem Codewort und Parity-Check-Matrix
4. Korrektur bei eindeutigem Fehlersyndrom (Argumente: empfangene Nachricht, Fehlersyndrom; Rückgabewerte: korrigierte Nachricht, Anzahl korrigierter Fehler, Boolean für übrige Fehler)

Testen Sie ihre Implementierung mit

1. der Generatormatrix des (7,4)-Codes aus der Vorlesung
2. der Generatormatrix $G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$

4 Simulation

Implementieren Sie die Simulation einer digitalen Übertragungsstrecke, die folgende Komponenten beinhaltet:

1. Quelle: Erzeugung einer zufälligen binären Nachricht aus $N=M*k$ Bits.
2. Kanalcodierung (Sender): Codieren Sie die Nachricht mit einem linear-systematischen Code, indem Sie die Nachricht in M Nutzworte zu je k Bits aufteilen und diese mit einer gegebenen Generatormatrix codieren. Fügen Sie die Nutzworte nach der Codierung wieder zu einer Nachricht zusammen
3. Übertragungskanal: Simulieren Sie die Übertragung der Nachricht über einen binär-symmetrischen Übertragungskanal mit vorgegebener Bitfehlerwahrscheinlichkeit p und generieren Sie einer fehlerbehaftete Nachricht am Empfänger.
4. Kanalcodierung (Empfänger): Dekodieren/korrigieren Sie die Nachricht am Empfänger.
5. Führen Sie eine statistische Auswertung durch und bestimmen Sie

- a. die Anzahl/den Anteil fehlerhafter Bits nach der Fehlerkorrektur
 - b. die Anzahl/den Anteil fehlerfrei übertragener Nutzworte
 - c. die Anzahl/den Anteil richtig korrigierter Nutzworte
 - d. die Anzahl/den Anteil als fehlerhaft erkannter aber nicht korrigierter Nutzworte
 - e. die Anzahl/den Anteil falsch korrigierter Nutzworte
 - f. die Anzahl/den Anteil unerkannter fehlerhafter Nutzworte
6. Führen Sie eine Simulation mit $M=1000$ Nutzworten zu je 4 Bits durch. Testen Sie den (7,4)-Hamming-Code für Bitfehlerwahrscheinlichkeiten von 1%, 2%, 5%, 10% und 25%. Erstellen Sie ein Diagramm, dass die Anzahl fehlerhafter Nutzworte sowie die Anzahl erkannter aber nicht korrigierbarer Fehler zeigt.
7. Wie ändern sich der Transinformationsgehalt und die Kapazität des binär-symmetrischen Übertragungskanal mit und ohne der Kanalcodierung?

Hinweis: Nutzen Sie zur Implementierung der Simulation die Funktionen (`randsrc`, `randerr`, `encode`, `bsc`, `hammg`, `syndtable`, `decode`, `biterr`) der Communication Systems Toolbox. Jede einzelne Komponente - außer der Auswertung - sollte in wenigen Zeilen implementierbar sein. Versuchen Sie Schleifen zu vermeiden und nutzen Sie die Möglichkeit der Matlab-Funktionen mit Matrizen zu rechnen, z.B. kann `randsrc` eine Matrix erzeugen, die in jeder Zeile ein Nutzwort enthält, und `encode` kann diese Matrix zeilenweise codieren. Die Matlab-Funktion `reshape` kann genutzt werden, um Matrizen spaltenweise in einen Vektor zu schreiben, um die Gesamtnachricht zu erhalten. Dies ist aber nicht unbedingt notwendig, um die Laborübung zu lösen.