TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Elektrotechnik und Informationstechnik – Institut für Nachrichtentechnik – Lehrstuhl Theoretische Nachrichtentechnik

Prof. Eduard Jorswieck, Anne Wolf

Praktikum Fehlerreduktionssysteme / Codierungstheorie

Versuch 2: Reed-Solomon-Coder und -Decoder

2.1 Grundlagen und Begriffe

Die Klasse der Reed-Solomon-Codes (RS-Codes) ist die einzige Klasse mehrwertiger Codes, die technisch von Bedeutung ist. Anwendung finden RS-Codes vor allem in Systemen mit burstartigem Fehlerverhalten. Nicht selten bildet ein RS-Code den äußeren Code in einem System mit Codeverkettung. So werden beim Compact Disc Digital Audio zwei verkettete RS-Codes eingesetzt, die im Zusammenhang mit mehrfachem Interleaving (Codespreizung) Burstlängen von ≈ 4000 Bits korrigieren können. Ein Ziel des Versuches ist, an Beispielen das Rechnen in Galois-Feldern (GF) und in deren Erweiterungskörpern zu üben und so das Verständnis für die mathematischen Hintergründe der Codierung und Decodierung zu vertiefen. Ein weiterer Schwerpunkt ist die Demonstration der besonderen Eignung der RS-Codes zur Korrektur von Burstfehlern.

2.1.1 Rechenregeln in $\mathrm{GF}(2^m)$

Da in technischen Anwendungen die binäre Darstellung von Informationen dominiert, spielen Erweiterungskörper $\mathrm{GF}(2^m)$ eine besondere Rolle. Die Elemente eines $\mathrm{GF}(2^m)$ werden als Potenzen des primitiven Elementes α durch $\alpha^i \bmod p(x)$ berechnet. $p(x)$ ist dabei ein primitives Polynom mit Koeffizienten in $\mathrm{GF}(2)$.

Exponentendarstellung	Komponentendarstellung	binär	dezimal
α^0	1	0001	1
α^1	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha^3 + 1$	1001	9
α^5	$\alpha^3 + \alpha + 1$	1011	11
α^6	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^7	$\alpha^2 + \alpha + 1$	0111	7
α^8	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^9	$\alpha^2 + 1$	0101	5
α^{10}	$\alpha^3 + \alpha$	1010	10
α^{11}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{12}	$\alpha + 1$	0011	3
α^{13}	$\alpha^2 + \alpha$	0110	6
α^{14}	$\alpha^3 + \alpha^2$	1100	12
α^{15}	1	0001	1

Tabelle 1: Elemente des $\mathrm{GF}(2^4)$ für das primitive Polynom $p(x) = x^4 + x^3 + 1$

Die Darstellung der Elemente kann dann in Exponentenschreibweise, d.h. als Potenzen von α, oder Komponentenschreibweise erfolgen. Die Komponentendarstellung ergibt sich aus der Berechnung der Potenzen von $\alpha \bmod p(\alpha)$. Für das Rechnen mit den Elementen sind sowohl Exponenten- als Komponentendarstellung hilfreich. Die Addition zweier Elemente wird am einfachsten über die bitweise mod 2-Addition (XOR-Verknüpfung) der Komponentendarstellung der Elemente durchgeführt. Für die Multiplikation ist die Exponentendarstellung besser geeignet. Hier ergibt sich das Produkt zweier Elemente durch die mod n-Addition der Exponenten ($n = 2^m - 1$).

Als Beispiel dient hier das primitive Polynom $p(x) = x^4 + x^3 + 1$. Die Erzeugung der Elemente des für die Simulation verwendeten GF(2^4) ist in Tabelle 1 dargestellt. Die binäre und die dezimale Darstellung ist nur eine verkürzte Schreibweise der Komponentendarstellung und für Berechnungen eher weniger geeignet.

2.1.2 Codierung

Die Konstruktion von RS-Codes erfolgt auf der Basis von Galois-Feldern. Bei der Beschreibung der Methoden zur Codierung bzw. Decodierung wird hier nur auf die Verwendung von GF(2^m) eingegangen.

Das Eingangsalphabet für einen RS-Code über GF(2^m) umfasst 2^m Symbole zu m Bit. Ist k die Anzahl der zu codierenden Symbole und $n = 2^m - 1$ die Länge des gebildeten Codewortes, so ergibt sich für ein Informationspolynom

$$a(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-2} x^{k-2} + a_{k-1} x^{k-1} \text{ mit } a_i \in \mathrm{GF}(2^m)$$

das zugehörige Codewort zu

$$\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-2}, c_{n-1}) \text{ mit } c_i = a(\alpha^i) \text{ und } i = 0, 1, \ldots, n-2, n-1.$$

Die Anzahl der Informationssymbole k berechnet sich dabei entsprechend der geforderten Minimaldistanz d zu

$$k \leq n - d + 1.$$

Da ein beliebiges Polynom $a(x)$ mit $\mathrm{grad}(a(x)) \leq k$ maximal k Nullstellen in GF(2^m) besitzen kann, ist gewährleistet, dass jedes gültige Codewort mindestens d Symbole $\neq 0$ besitzt. Somit hat der gebildete Code die Minimaldistanz d und ist in der Lage, maximal

$$t = \frac{d-1}{2}$$

Fehler zu korrigieren.

Das folgende Beispiel demonstriert die Vorgehensweise bei der Bildung des Codewortes. Dabei werden die Elemente während der einzelnen Schritte in der jeweils günstigsten Darstellung verwendet.

Beispiel: 4-Fehler-korrigierender RS-Code über GF(2^4)

Mit $t = 4$ und $m = 4$ erhält man

$$\begin{aligned} n &= 2^m - 1 = 2^4 - 1 = 15 \quad \text{und} \\ k &= n - d + 1 = n - 2 \cdot t = 15 - 2 \cdot 4 = 7 \end{aligned}$$

Somit ergibt sich ein (15,7)-RS-Code.

Wir wählen ein beliebiges Informationswort / -polynom, z.B.

$$\begin{aligned} \boldsymbol{a} &= (2, 6, 8, 12, 15, 13, 1) \\ a(x) &= 2 + 6x + 8x^2 + 12x^3 + 15x^4 + 13x^5 + x^6. \end{aligned}$$

Dafür berechnen wir die Codewortsymbole

$$\begin{aligned}
c_0 = a(\alpha^0) &= 2 + 6 + 8 + 12 + 15 + 13 + 1 \\
&= 3 \\
c_1 = a(\alpha^1) &= 2 + 6\alpha + 8\alpha^2 + 12\alpha^3 + 15\alpha^4 + 13\alpha^5 + 1\alpha^6 \\
&= \alpha + \alpha^{13}\alpha + \alpha^3\alpha^2 + \alpha^{14}\alpha^3 + \alpha^6\alpha^4 + \alpha^{11}\alpha^5 + \alpha^0\alpha^6 \\
&= \alpha + \alpha^{14} + \alpha^5 + \alpha^2 + \alpha^{10} + \alpha + \alpha^6 \\
&= 2 + 12 + 11 + 4 + 10 + 2 + 15 \\
&= 6 \\
&\vdots \\
c_{14} = a(\alpha^{14}) &= 2\alpha^{14\cdot 0} + 6\alpha^{14\cdot 1} + 8\alpha^{14\cdot 2} + 12\alpha^{14\cdot 3} + 15\alpha^{14\cdot 4} + 13\alpha^{14\cdot 5} + 1\alpha^{14\cdot 6} \\
&= \alpha + \alpha^{13}\alpha^{14} + \alpha^3\alpha^{13} + \alpha^{14}\alpha^{12} + \alpha^6\alpha^{11} + \alpha^{11}\alpha^{10} + \alpha^0\alpha^9 \\
&= \alpha + \alpha^{12} + \alpha + \alpha^{11} + \alpha^2 + \alpha^6 + \alpha^9 \\
&= 2 + 3 + 2 + 13 + 4 + 15 + 5 \\
&= 0
\end{aligned}$$

und erhalten so das komplette Codewort

$$\boldsymbol{c} = (3, 6, 15, 6, 6, 3, 13, 14, 3, 12, 15, 2, 11, 1, 0).$$

Die Tatsache, dass ein Symbol eines RS-Codes über GF(2^m) durch m Bits dargestellt wird, hat besonderen Einfluss auf das Fehlerkorrekturverhalten dieser Codes. Es ist leicht einzusehen, dass sich RS-Codes gut zur Behandlung von burstartigen Störungen eignen. Ein t-Fehler-korrigierender RS-Code ist geeignet, Fehlerbursts von $(t-1)\cdot m + 1$ Bits auf jeden Fall zu korrigieren, im besten Fall können sogar $t \cdot m$ aufeinanderfolgende Bitfehler korrigiert werden. Da mehrere verfälschte Bits zu dem selben Symbol gehören können, zieht nicht jeder Bitfehler auch einen Symbolfehler nach sich. Bei einer gegebenen Bitfehlerwahrscheinlichkeit p berechnet sich die Symbolfehlerwahrscheinlichkeit zu

$$P_s = 1 - (1-p)^m.$$

2.1.3 Decodierung und Fehlerkorrektur

Zur Erkennung eines Übertragungsfehlers erfolgt die Berechnung des Syndroms $\boldsymbol{s}$. Für ein empfangenes Codewort $r(x) = c(x) + f(x)$ ergibt sich das Syndrom

$$\boldsymbol{s} = (s_0, s_1, \ldots, s_{d-3}, s_{d-2}) \text{ mit } s_{i-k} = r(\alpha^{-i}) \text{ und } i = k, k+1, \ldots, n-2, n-1.$$

Für fehlerfreie Übertragung, d.h. $f(x) = 0$, ergibt sich $s(x) = 0$. Ansonsten erhält man ein nur vom Fehler abhängiges Syndrom $s(x) = s_0 + s_1 x + \ldots + s_{d-3}x^{d-3} + s_{d-2}x^{d-2}$. Soll eine Fehlerkorrektur erfolgen, so sind die folgenden Verarbeitungsschritte durchzuführen:

1. Ermittlung des Fehlerstellenpolynoms $C(x)$
2. Berechnung der Fehlerstellen (Nullstellen von $C(x)$)
3. Berechnung der Fehlerwerte

Konnte der aufgetretene Fehler $f(x)$ bestimmt werden, so kann durch Addition von $r(x)$ und $f(x)$ das ursprüngliche Codewort $c(x)$ restauriert werden. Für ein fehlerfrei übertragenes bzw. korrigiertes Codewort $c(x)$ kann mit

$$\boldsymbol{a} = (a_0, a_1, \ldots, a_{k-2}, a_{k-1}) \text{ mit } a_i = c(\alpha^{-i}) \text{ und } i = 0, 1, \ldots, k-2, k-1.$$

die gesendete Information bestimmt werden.

2.1.4 Interleaving

Ein häufig bei Kanälen mit burstartigen Störungen angewendetes Verfahren ist das Interleaving (Codespreizung). Das Prinzip besteht darin, ein Codewort über einen längeren Zeitraum verteilt zu senden. Zu diesem Zweck werden ℓ Codewörter zeilenweise in eine Matrix eingelesen und dann spaltenweise ausgelesen. Am Empfänger wird diese Operation in umgekehrter Reihenfolge durchgeführt, um die ursprüngliche Anordnung der Codesymbole wieder herzustellen. Ein t-Fehler korrigierender Code kann auf diese Weise Bursts mit der maximalen Länge $t \cdot \ell$ korrigieren. Man nennt ℓ die Interleavingtiefe.

Es werden auch andere Interleavingmuster, z.B. über spezielle Tabellen festgelegtes Ein- und Auslesen eines Speichers, verwendet (s.a. Versuch 4: Faltungscodierung).

2.2 Versuchsdurchführung

2.2.1 Simulationsprogramm

Zur Untersuchung der Eigenschaften von RS-Codes wurde ein Simulationsprogramm erstellt. Dieses Programm ist Bestandteil der WINDOWS-Anwendung *FRS für Windows*. Als Beispielcode wird ein (15, k)-Code ($k = 3, 5, 7, 9, 11, 13$) verwendet, dessen Fehlerkorrektur-Bereich zwischen 1 und 6 Symbolfehlern wählbar ist. Gestartet wird die Simulation, indem zunächst das Hauptprogramm aufgerufen wird und dann der Menübefehl *Reed-Solomon-Codes – Simulation* ausgewählt wird.

Ein Simulationszyklus umfasst die Codierung, die Übertragung und die Decodierung eines (Interleaving ausgeschaltet) oder mehrerer (Interleaving eingeschaltet) Codewörter. Für den Codierer kann die Anzahl der korrigierbaren Fehler t eingestellt werden. Abhängig von t wird die Anzahl der zu codierenden Informationssymbole k berechnet. Hierbei kann der allgemeine Zusammenhang zwischen t und k des jeweils gebildeten Codes untersucht werden.

Ebenso ist eine Veränderung der Länge des berechneten Syndroms zu beobachten. Über das Dialogfenster *Kanal-Eigenschaften* kann festgelegt werden, welche Fehlersimulation bei der Übertragung der Codewörter durchgeführt wird.

2.2.2 Versuchsaufgaben

1. Kanal mit wählbarem Fehlervektor

   Für ausgewählte Fehlermuster ist zu prüfen, ob die angegebene Korrekturleistung erreicht wird und wie sich die Codierung bei Fehlern verhält, welche die Korrekturfähigkeit überfordern.

   Protokollieren Sie für den (15,9)-RS-Code ohne Interleaving je ein Beispiel
   a) mit nicht mehr als drei,
   b) mit mehr als drei Symbolfehlern,
   die Sie in den Fehlervektor eingegeben haben!

   Vollziehen Sie für einen der beiden Fälle einen Übertragungsvorgang im Detail nach! Dabei sind folgende Teilaufgaben zu erfüllen:

   - Bestimmen eines Codewortsymbols (nicht c_0) für ein zufällig generiertes Datenwort und Vergleich mit dem vom Simulator erzeugten Codewort,
   - Nachvollziehen der Entstehung des Empfangswortes aus dem Sendewort durch Berechnung der Werte an den fehlerhaften Stellen,
   - Berechnen einer Stelle des Syndroms (nicht s_0) und Vergleich mit dem vom Simulator erzeugten Syndrom,
   - Analyse der Simulationsauswertung:
     – Wieviele Symbol- und Bitfehler sind aufgetreten?
     – Sind Decodier- und Restfehler aufgetreten?

Außerdem sollen das Verhalten des Syndroms und verschiedene Fehlerfälle betrachtet werden. Dazu sind folgende Teilaufgaben zu bearbeiten:

- Betrachten des Syndroms für verschiedene Informationswörter bei gleichem Fehlervektor,
- Unterscheiden der verschiedenen Fehlerfälle und Nachvollziehen ihrer Auswirkung auf die Statistik (Decodier- und Restfehler)

2. Symmetrischer Binärkanal (BSC)

   Es werden zufällige Bitfehler mit 5 verschiedenen Bitfehlerwahrscheinlichkeiten p erzeugt. Unter Verwendung der Zähler für Bitfehler und Symbolfehler sind die Bitfehler- und Symbolfehlerwahrscheinlichkeiten zu ermitteln und mit den theoretischen Werten zu vergleichen.

   Protokollieren Sie für einen (15,9)-RS-Code ohne Interleaving in einer Tabelle die Häufigkeiten von Symbol- und Bitfehlern für eine Versuchsreihe, in der für jede der fünf wählbaren Fehlerhäufigkeiten drei Übertragungen beobachtet werden.

3. Burst-Kanal

   Für die Untersuchung der Funktion des Interleavers ist die Simulation eines Burst-Kanals mit unterschiedlichen mittleren Burstlängen zu nutzen. Für ausgewählte Korrekturleistungen und mittlere Burstlängen sind mehrere Simulationszyklen bei ein- bzw. ausgeschaltetem Interleaver durchzuführen und anschließend die Anzahl der Decodier- und Restfehler zu vergleichen.

   Protokollieren Sie für einen (15,9)-Code in einer Tabelle die Häufigkeiten von Decodier- und Restfehlern für eine Versuchsreihe, in der für jede der vier wählbaren Burstlängen drei Übertragungszyklen aufgenommen werden. Beobachten Sie zum Vergleich die Übertragung der gleichen Datenmenge ohne Interleaving!