Grundkurs Codierung

Lösungsvorschläge zu den Fragen in den Unterkapiteln „Was blieb?“

Stand 25.08.2025

Unterkapitel 3.14 Seite 240

Zu Frage 1:

BCH-Codes sind die zyklischen Sonderfälle der im allgemeinen nicht zyklischen Goppa-Codes. Jedoch sind sie systematisch, siehe Bild 3.21, v'(1), v'(2), v'(4) und v'(8). Gemäß Unterkapitel 3.8.3 entsteht der BCH-Code aus der allgemeinen Berechnungsvorschrift für den Goppa-Code, wenn man als erzeugendes Polynom $c(z) = z^{2t}$ wählt. Dabei ist t die Anzahl der zu korrigierenden 1-Bitfehler.

Hinweis: Das Kapitel “Goppa-Code” wurde in Auflage 5 herausgenommen, da sich dieses Verfahren im praktischen Einsatz nicht durchgesetzt hat. Es ist aber als Basis für den BCH-Code weiter theoretisch interessant, da dieser einen Sonderfall des Goppa-Codes darstellt. Sie finden das vollständige Kapitel als .pdf jetzt auf der Homepage vkfco.de.

Zu Frage 2:

Der Goppacode ist systematisch, aber im allgemeinen nicht zyklisch. Für $c(z) = z^{2t}$ entsteht allerdings als Sonderfall ein zyklischer Code, der BCH-Code.

Zu Frage 3:

Die Länge n der Codewörter wird beim Goppa-Code durch die Anzahl der Elemente des zugrunde liegenden Galoisfeldes $GF(2^m)$ festgelegt, $n = 2^m$, siehe auch Unterkapitel 3.8.1, Seite 161. Man beachte aber, dass nach Unterkapitel 3.8.3 bei Verwendung von bestimmten Polynomen $c(z) = z^{2t}$ für die Berechnung der modularen inversen Polynome $h_i(z)$ mit

$$h_i(z),\ i = 1, 2, ..., 2^m$$

$$h_i(z) \cdot (z - \beta_i) \text{ MOD } c(z) = 1$$

die erste Gleichung $h_1(z)$ - siehe Tabelle 3-26 auf Seite 169 – wegen $c(\beta_1 = 0) = 0$ in

$$h_i(z) = \frac{c(z) \cdot c(\beta_i)}{z - \beta_i} \cdot c(\beta_i)^{-1}$$

nicht definiert ist und deshalb weg fällt, siehe Seite 183. Daher hat diese spezielle Version des Goppacodes nur $n = 2^m - 1$ Codewortelemente. Bei der weiteren Untersuchung stellt man z. B. für m=4 und t=3 fest, dass von den verbleibenden $2^4 - 1 = 15$ Gleichungen nur noch 10 Paritätsgleichungen übrig bleiben, siehe Seite 185. Damit ist der so gebildete Code dem BCH-Code der Länge n = 15 für t=3 ähnlich.

Zu Frage 4:

Die Paritätsgleichungen werden mit $n = 2^m$ und dem Codewortaufbau

$$v = (v_1\ v_2 \ldots v_{n-1}\ v_n)$$

nach Unterkapitel 3.8.1 über die Beziehung

$$v_1\, h_1(z) + v_2\, h_2(z) + \ldots + v_n\, h_n(z) = 0$$

gebildet. Dabei stellt man fest, dass für einen Goppa-Code für t-Fehler-Korrektur zur Berechnung der $n = 2^m$ Codewortelemente $m \cdot t < n$ Gleichungen zur Verfügung stehen. Es können also $k = n - m \cdot t$ Elemente frei festgelegt werden. Diese sind dann die k Info-Stellen, die restlichen die $m \cdot t$ Prüfbits. Im Beispiel von Unterkapitel 3.8.1 ist m = 4, t =3, also hat dieser Goppacode die Länge 16 mit 4 Info- und 12 Prüfstellen.

Die modularen inversen Polynome $h_i(z)$ ergeben sich aus

$$h_i(z) \cdot (z - \beta_i) \text{ MOD } c(z) = 1$$

Dabei ist c(z) ein Polynom vom Grad $c(z) \geq t$, welches keine Nullstellen im verwendeten Galoisfeld $GF(2^m)$ besitzt. Im Beispiel auf Seite 167 wird dazu $c(z) = z^3 + z + 1$ genommen. Die Wahl von c(z) ist darüber hinaus allerdings willkürlich und es gibt auch keine einfachen Regeln, um mit c(z) die Leistungs-bestimmenden Eigenschaften des Codes beeinflussen zu können (Ausnahme: $c(z) = z^{2t}$, daraus entsteht als Sonderfall der BCH-Code, siehe Unterkapitel 3.8.3).

Zu Frage 5:

Wesentliches Hilfsmittel ist der Euklidische Algorithmus zur Bestimmung des größten gemeinsamen Teiler-Polynoms (genauer: des gemeinsamen Polynoms größten Grades) zweier gegebener Polynome sowie die Rückwärtsrechnung des Euklidischen Algorithmus zur Bestimmung des modularen inversen Polynoms, wenn das gemeinsame Teiler-Polynom den Grad 0 hat, siehe Unterkapitel 2.2, Seiten 44 – 46 und das Beispiel im Unterkapitel 3.8.1 auf den Seiten 167 und 168.

Man erhält dabei die inversen modularen Polynome $h_i(z)$ bezüglich der beiden im $GF(2^m)$ teilerfremden Polynome

$$z - \beta_i \qquad \text{und } c(z)$$

Im Beispiel ist m = 4, und $c(z) = z^3 + z + 1$.

Zu Frage 6:

Der Reed-Muller-Code verwendet das Prinzip der Mehrheitswerte zur Bestimmung der Info-Bits, siehe Unterkapitel 3.9, Seiten 194 – 197. Er ist im Übrigen ein nicht systematischer Code, da die Info-Bits keinen unmittelbaren Bestandteil der Codewörter darstellen.

Zu Frage 7:

Für die Codierung und Dekodierung des Reed-Muller-Codes werden ausschliesslich XOR-Operationen benötigt, die zu den elementaren Maschinenbefehlen jedes Prozessors gehören.

Zu Frage 8:

Falls das Rauschen rs entgegen der idealsierten Annahme nicht (ganz) unkorreliert zur Folge und zum Takt des Sendesignals vs ist, also beide Vorgänge nicht völlig unabhängig von einander sind, kann es passieren, dass trotz Interleaving auch die Verschachtelungstiefe v_d und die Fehlerbündelbreite b korrelieren. In diesem Fall würden die Fehlerbündel nicht ausreichend gut auf die einzelnen Codewörter zerlegt. Solche Abhängigkeiten lassen sich mithilfe von Kreuzkorrelationsfunktionen aufdecken, s. S. Unterkapitel 4.1, S. 243.

Man kann solche Abhängigkeiten dann Verringern oder sogar Beseitigen, wenn das Interleaving die Bits

eines Superblocks nicht über eine feste Verschachtelungstiefe sondern (pseudo-)statistisch zufällig permutiert. Allerdings muss der Sender dann zu jedem Superblock auch die zugehörige Permutationstabelle angeben (siehe DES). Der Mehraufwand lässt sich in Grenzen halten, wenn die notwendigen Zufallszahlen über ein Schieberegister erzeugt werden, für deren Berechnung nur zu Anfang die Länge des Registers, das Rückkopplungspolynom und der Startzustand bekannt sein müssen, siehe Unterkapitel 4.3.

Zu Frage 9:

Ja, die c*onstituent codes* eines Produktcodes können in jeder Dimension verschieden gewählt werden. Eine andere Frage ist allerdings, ob hierdurch besondere Vorteile entstehen. Am Markt erhältliche Chips für Fehlerkorrektur mit Produktcodes enthalten in jeder Dimension die gleichen Codes.

Im Übrigen sind Produktcodes nicht nur auf 2 Dimensionen beschränkt, sonder können aus beliebig vielen bestehen. Doch auch hier ist die Frage nach dem Nutzen zu beantworten. Ausgeführte Codecs bieten die Wahl von 2 oder 3 Dimensionen.

Zu Frage 10:

Die Stärke besteht darin, dass bei Produktcodes jedes Empfangsbit aus 2 oder mehr statistisch unabhängigen Messungen beurteilt wird. Dadurch kann die Zuverlässigkeit der Aussage "*das Empfangsbit ist wahrscheinlich ein 0-Bit oder ein 1-Bit*" gegenüber einer nur eindimensionalen Messung verbessert werden. Der Preis besteht im höheren Rechenaufwand.

Zu Frage 11:

Beim ML (=Maximum Likelihood)-Prinzip werden alle Info-Bits eines Codewortes als gleich wahrscheinlich angenommen. Dies ist theoretisch nur im Mittel über viele Codewörter, nicht jedoch im Einzelfall richtig. Bei eindimensionaler Betrachtung hat man allerdings keine Möglichkeit, durch Messung erzeugte Aussagen über die tatsächliche Verteilung der Info-Bits im Codewort zu treffen. Obwohl also die Dekodierung nach dem ML-Prinzip bereits eine deutliche Verbesserung der Korrekturleistung gegenüber den HD-Verfahren bringt, holt man damit noch nicht "das Letzte" heraus (auch wenn es in einem Empfangswort viele "starke" Infobits gibt, können die restlichen "schwachen" Info- und Prüf-Bits doch zur Zuordnung eines falschen Codeworts führen).

Mithilfe des MAP (=Maximum a Posteriori)-Prinzips wird die Zuverlässigkeit zur Dekodierungsentscheidung individuell auf die einzelnen Info-Bits im Codewort aus wenigstens 2 verschiedenen, statistisch unabhängigen Messungen getroffen. Die Unzuverlässigkeit "schwacher" Info-Bit-Kandidaten schlägt dann nicht wie bei der Gesamtbetrachtung der Codewörter voll auf die "Starken" durch und zieht sie unter Umständen noch herunter. Die Summe der richtig zugeordneten Info-Bits ist hier im Mittel höher.

Zu Frage 12:

Sind zwei (zufällige) Ereignisse wie das Auftreten eines Sendewortes vs und eines Empfangswortes ws = vs + rs statistisch voneinander abhängig und hat man durch Messung in einem bestimmten Fall festgestellt, dass das eine Ereignis von den beiden sicher eintrat, so kann die Antwort auf die Frage interessieren, wie groß dann die Wahrscheinlichkeit für das andere Ereignis ist. Wurde z. B. ws gemessen, so entsteht die Frage, welche zuverlässige Aussage dann über das zugehörige Sendewort vs gemacht werden kann.

Formal drückt man dieses Zuverlässigkeitsmaß als bedingte Wahrscheinlichkeit $P(vs|ws)$ für kontinuierliche bzw $p(ws|vs)$ für diskrete Verteilungen aus, siehe Unterkapitel 3.12, Seite 217. Bei gleich bleibenden statistischen Parametern lässt sich diese bedingte Wahrscheinlichkeit z. B. dadurch ermitteln, dass man für alle möglichen Werte-Tupel ws alle möglichen Werte-Tupel von vs bestimmt und daraus die Wahrscheinlichkeiten berechnet. Selbst bei kleinen Tupeln (2 oder 3 "Bits" pro Wort) ist dies schon recht aufwändig, grafisch kann es nur für 2 Bits pro Wort übersichtlich dargestellt werden, siehe Bilder 3.32a und 3.32b.

Hätte man ein solches Tabellenwerk erstellt, so könnte man theoretisch bei Messung eines bestimmten Wortes ws der Länge n (n-Tupel) daraus die zugehörigen bedingten Wahrscheinlichkeiten für bestimmte n-Tupel Kombinationen der Sendewörter vs ablesen - und würde das mit der größten Wahrscheinlichkeit als beste Schätzung $\hat{vs}$ wählen (jedoch hiesse das nicht unbedingt, dass es auch das richtige wäre !). Wegen des Aufwandes ist dieses Vorgehen zur Lösung der praktisch vorliegenden Aufgaben aber nicht gangbar. Ein Beispiel für den (7,4,3)-Hammingcode. Für vs gibt es 16 Kombinationen. Würde man die kontinuierlichen Werte für die 7 Komponenten des Empfangswortes ws in z. B. 15 diskrete Wertebereiche von "-" nach "+" einteilen, so hätte man 15^7= 170.859.375 Kombinationen zu untersuchen und dies in 16 Tabellen mit 171 Milliarden Zeilen nieder zu legen. Für signifikante Ergebnisse müssten dazu deutlich mehr als 16 x 171 Milliarden Messung vorgenommen werden, also vielleicht wenigsten 100 Mal so viele, rund 300 Billionen.

Die Bayes'schen Beziehungen $P(vs|ws)=\frac{p(ws|vs)\cdot P(vs)}{p(ws)}$ ermöglichen nun unter bestimmten Voraussetzungen, diesen Aufwand zu verkleinern. Wenn das Rauschen rs mittelwertfrei und normalverteilt mit bekannter Standardabweichung ist, dann können sowohl $p(ws)$ als auch $p(ws|vs)$ als mathematischer Ausdruck angegeben werden, siehe S.215 und S. 221. Für die diskrete Auftrittswahrscheinlichkeit $P(vs)$ der Codewörter gilt $1/(2^k)$. Der Vorteil liegt zunächst also darin, dass die gesuchte bedingte Wahrscheinlichkeit $P(vs|ws)$ nicht mehr durch Messungen ermittelt werden muss, sondern sich berechnen lässt. Damit ist sie auch gut als Ausgangspunkt für weitere Betrachtungen geeignet, unter anderem für die Beantwortung der Frage, welche Zuverlässigkeitsaussagen über die Schätzwerte der Info-Bits getroffen werden können. Dies ist zugleich der Ausgangspunkt für die Korrekturverbesserung mithilfe der vertikalen (genauer orthogonalen) Dimension.

Zu Frage 13:

Es wird das allgemein gültige **Bellman'sche Optimalitätsprinzip** genutzt. Im vorliegenden Fall lässt es sich anhand von Bild 3.36 erkennen. Bis zum Takt i=3 muss jeder der bis hierhin entstandenen Wege für sich geführt werden, es entstanden bis zu 8 verschiedene Hammingabstände, im Beispiel sind es dg = 1, 2, 3, 4 und 6.

Ab i=4 gibt es zu jedem der 8 Vorgängerzustände aus Takt i = 3 je zwei, die zum **selben** Folgezustand führen. Nach dem oben genannten Optimalitätsprinzip kann aber nur derjenige der "Richtige" sein, der den kleineren Abstand aufweist, da der andere Weg seinen größeren Abstand im weiteren Verlauf nicht mehr verkleinern würde. So etwa geht der Zustand x=00 von Takt i=3 im Takt i=4 über u=0 in den Zustand x=00 über (unterste Zeile), aber auch der Zustand x=01 (dritte Zeile von unten) über u=1 in den Zustand x=00. Im ersteren Fall beträgt der aufgelaufene Gesamtabstand dg=3, im letzteren dg=2. Deshalb gehört dieser zu den 4 grau unterlegten Kandidaten für die weitere Rechnung. Und ebenfalls aus diesem Grund verändert sich die Kandidatenzahl in den nächsten Schritten nicht.

Bei diesem Vorgehen wächst also die Zahl zu betrachtender Wege nicht unbegrenzt in jedem Takt auf das doppelte, sondern bleibt bei 2^{m+1} stehen.

Zu Frage 14:

Nein, es gibt solche mathematisch begründeten Verfahren nicht. Geeignete Generatorpolynome müssen durch Probieren gefunden werden.