10 Zyklische Codes

10 Zyklische Codes

- Zyklische Codes sind spezielle lineare Codes,
- bei denen jede zyklische Verschiebung eines Codeworts wieder ein Codewort ergibt.

Beispiel 10.1 (Wiederholungscode)

Der schon bekannte Wiederholungscode ist offensichtlich ein zyklischer Code, da jede zyklische Verschiebung von $0 \ldots 0$ wieder $0 \ldots 0$ und jede zyklische Verschiebung von $1 \ldots 1$ wieder $1 \ldots 1$ ergibt.

Beispiel 10.2 (Parity-Check-Code)

Auch der Parity-Check-Code ist ein zyklischer Code: Da die zykliche Verschiebung die Anzahl der Einsen im Codewort nicht ändert, wird aus jedem Codewort (gerade Anzahl von Einsen) wieder ein Codewort.

Bei der Codierung mit zyklischen Codes ist es zweckmäßig, das Codewort $\boldsymbol{x} = x_{n-1}x_{n-2}\ldots x_0$ als Polynom

$$x(\chi) = x_{n-1}\chi^{n-1} + x_{n-2}\chi^{n-2} + \cdots + x_0 = \sum_{i=0}^{n-1} x_i\chi^i \tag{10.1}$$

vom Grad $n-1$ zu interpretieren.

Beispiel 10.3

Das Codewort $\boldsymbol{x} = 10110$ entspricht dem Polynom

$$x(\chi) = 1\cdot\chi^4 + 0\cdot\chi^3 + 1\cdot\chi^2 + 1\cdot\chi + 0 = \chi^4 + \chi^2 + \chi.$$

Beispiel 10.4

Das Codewort $\boldsymbol{x} = 010101$ entspricht dem Polynom

$$x(\chi) = 0\cdot\chi^5 + 1\cdot\chi^4 + 0\cdot\chi^3 + 1\cdot\chi^2 + 0\cdot\chi + 1 = \chi^4 + \chi^2 + 1.$$

Satz 10.1

Eine zyklische Verschiebung von $x(\chi)$ um eine Stelle nach links lässt sich dann als

$$x'(\chi) = x(\chi)\cdot\chi \mod (\chi^n + 1) \tag{10.2}$$

schreiben.

Beweis.

$$x(\chi)\cdot\chi \mod (\chi^n+1) = \left(\sum_{i=0}^{n-1} x_i\chi^{i+1}\right) \mod (\chi^n+1) \tag{10.3}$$

$$= \left(x_{n-1}\chi^n + \sum_{i=0}^{n-2} x_i\chi^{i+1}\right) \mod (\chi^n+1) \tag{10.4}$$

$$= \left(x_{n-1}(\chi^n+1) + \sum_{i=0}^{n-2} x_i\chi^{i+1} + x_{n-1}\right) \mod (\chi^n+1) \tag{10.5}$$

$$= \sum_{i=0}^{n-2} x_i\chi^{i+1} + x_{n-1} \tag{10.6}$$

□

Korollar 10.1

Eine zyklische Verschiebung um m Stellen lässt sich als

$$x'(\chi) = x(\chi) \cdot \chi^m \mod \chi^n + 1 \tag{10.7}$$

schreiben.

Beispiel 10.5

Das Codewort $\boldsymbol{x} = 01001$ ($n = 5$) soll um $m = 2$ Stellen verschoben werden. Mit $x(\chi) = \chi^3 + 1$ ergibt sich $x(\chi) \cdot \chi^2 = \chi^5 + \chi^2$. Durchführung der Polynomdivision ergibt:

$$\begin{array}{rl} (\chi^5 + & \chi^2 \quad\quad\;\;) \quad : (\chi^5 + 1) = 1 \\ -\,(\chi^5 + & \quad\quad\;\; 1) \\ & \chi^2 + \quad 1 \end{array}$$

mit dem Rest $\chi^2 + 1$, entsprechend $\boldsymbol{x'} = 00101$.

Satz 10.2

Jeder zyklische Code enthält ein eindeutiges von Null verschiedenes Polynom minimalen Grades als Codewort.

Beweis.

Offensichtlich gibt es unter den Codewörtern mindestens ein von Null verschiedenes Polynom minimalen Grades.
Angenommen, es gäbe (mindestens) zwei Polynome

$$x(\chi) = \chi^r + x_{r-1}\chi^{r-1} + \cdots + x_1\chi + x_0 \tag{10.8}$$

$$x'(\chi) = \chi^r + x'_{r-1}\chi^{r-1} + \cdots + x'_1\chi + x'_0 \tag{10.9}$$

minimalen Grades r (also $x_r = x'_r = 1$). Aufgrund der Linearität des Codes wäre dann auch

$$x(\chi) + x'(\chi) = (x_{r-1} + x'_{r-1})\chi^{r-1} + \cdots + (x_1 + x'_1)\chi + x_0 + x'_0 \tag{10.10}$$

ein Codewort, hätte aber (höchstens) den Grad $r - 1$, womit $x(\chi)$ und $x'(\chi)$ nicht minimalen Grades wären.
Also ist das Polynom minimalen Grades eindeutig. □

10.1 Darstellung mittels Generatorpolynom

Definition 10.1 (Generatorpolynom)

Das von Null verschiedene Polynom $g(\chi)$ minimalen Grades eines zyklischen Codes heißt Generatorpolynom des Codes.

Satz 10.3

Ist $g(\chi)$ das Generatorpolynom des Grades k eines zyklischen Codes, so ist $x(\chi)$ genau dann ein Codewort, wenn es ein Polynom $a(\chi)$ höchstens des Grades $n-k-1$ gibt, sodass $x(\chi) = a(\chi) \cdot g(\chi)$, also $g(\chi)$ ein Teiler von $x(\chi)$ ist.

Beweis zu "$x(\chi) = a(\chi) \cdot g(\chi) \Rightarrow x(\chi)$ ist ein Codewort".

Da es sich um einen zyklischen Code handelt, ist neben $g(\chi)$ auch jede zyklische Verschiebung $g(\chi) \cdot \chi^m \mod (\chi^n + 1)$ ein Codewort. Für $m \leq n-k-1$ kann dabei die Modulo-Operation entfallen, da das Ergebnis der Multiplikation maximal vom Grad $n-1$ ist.

Da es sich um einen linearen Code handelt, ist auch jede Summe der durch Verschiebung (bzw. Multiplikation) gebildeten Codewörter wieder ein Codewort. Diese Summe von Termen, die durch Multiplikation mit χ^m ($m \leq n-k-1$) gewonnenen werden, entspricht gerade der Multiplikation mit einem Polynom des Grades $n-k-1$ (oder weniger), also $a(\chi)$. □

Beweis zu "$x(\chi)$ ist ein Codewort $\Rightarrow x(\chi) = a(\chi) \cdot g(\chi)$".

Angenommen, $x(\chi)$ ist ein Codewort. Sei $r(\chi)$ der Rest, der bei Division durch $g(\chi)$ bleibt. (Beachte: Der Grad von $r(\chi)$ ist geringer als der von $g(\chi)$.)

Dann lässt sich $x(\chi) = a(\chi) \cdot g(\chi) + r(\chi)$ schreiben. Dabei sind sowohl $x(\chi)$, als auch $a(\chi) \cdot g(\chi)$ Codewörter, womit auch $r(\chi)$ ein Codewort sein muss (wegen der Linearität).

Da aber $g(\chi)$ per Definitionem das von Null verschiedene Codewort minimalen Grades ist und $r(\chi)$ ein Codewort noch geringeren Grades ist, muss $r(\chi) = 0$, also $x(\chi)$ durch $g(\chi)$ teilbar sein. □

Satz 10.4

Ein zyklischer Code der Länge n mit einem Generatorpolynom des Grades k hat einen Coderaum der Dimension $l = n-k$, codiert also $l = n-k$ Nutzdatenbits pro Codewort.

Beweis.

Da $a(\chi) \cdot g(\chi)$ den gesamten Coderaum abdeckt und $a(\chi)$ maximal $n-k$ Koeffizienten hat, hat der Coderaum einerseits maximal die Dimension $n-k$.

Andererseits sind $g(\chi), \chi \cdot g(\chi), \ldots, \chi^{n-k-1} \cdot g(\chi)$ linear unabhängige Codewörter und bilden so eine Basis der Dimension $n-k$, womit der Coderaum mindenstens $n-k$ Dimensionen haben muss.

Folglich hat der Coderaum genau $n-k$ Dimensionen. □

Der Beweis zu Satz 10.4 liefert gleichzeitig eine Vorschrift, wie sich aus einem Generatorpolynom eine Generatormatrix des gleichen Codes bilden lässt.

Beispiel 10.6

Sei $\chi^4 + \chi^3 + \chi^2 + 1$ das Generatorpolynom eines Codes der Länge $n = 7$. Die Generatormatrix muss also $n = 7$ Spalten und $l = n - k = 7 - 4 = 3$ Zeilen haben, die sich durch Verschieben des Generatorpolynoms bilden lassen:

$$\boldsymbol{G} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Ein Vorteil zyklischer Codes ist, dass sich mit dem Generatorpolynom eine wesentlich kompaktere Darstellung als mit der Generatormatrix anbietet.

Satz 10.5

Ein Polynom $g(\chi)$ ist genau dann Generatorpolynom eines zyklischen Codes der Länge n, wenn es Teiler von $\chi^n + 1$ ist.

Beweis zu "$g(\chi)$ ist ein Generatorpolynom $\Rightarrow$ $g(\chi)$ teilt $\chi^n + 1$".

Sei $g(\chi)$ ein Generatorpolynom vom Grad k, dann hat $\chi^{n-k} \cdot g(\chi)$ den Grad n. Die Division $\chi^{n-k} \cdot g(\chi)/(\chi^n + 1)$ ergibt damit 1 und einen Rest $x(\chi)$, also

$$\chi^{n-k} \cdot g(\chi) = \chi^n + 1 + x(\chi). \tag{10.11}$$

Dabei ist $x(\chi) = \chi^{n-k} \cdot g(\chi) \mod (\chi^n + 1)$ eine Verschiebung von $g(\chi)$ und damit eine Codewort, es gibt also ein Polynom $a(\chi)$, sodass

$$a(\chi) \cdot g(\chi) = x(\chi). \tag{10.12}$$

Die Summe beider Gleichungen ergibt

$$\chi^{n-k} \cdot g(\chi) + a(\chi) \cdot g(\chi) = \chi^n + 1 + x(\chi) + x(\chi) \tag{10.13}$$

$$\left(\chi^{n-k} + a(\chi)\right) \cdot g(\chi) = \chi^n + 1, \tag{10.14}$$

womit $g(\chi)$ Teiler von $\chi^n + 1$ ist. □

Beweis zu "$g(\chi)$ teilt $\chi^n + 1 \Rightarrow g(\chi)$ ist ein Generatorpolynom".

Sei

$$x(\chi) = a_{n-k-1}\chi^{n-k-1}g(\chi) + \cdots + a_1\chi g(\chi) + a_0 g(\chi) = a(\chi) \cdot g(\chi) \qquad (10.15)$$

ein beliebiges Codewort. Es ist zu zeigen, dass dann auch die zyklische Verschiebung $\chi \cdot x(\chi) \mod (\chi^n + 1)$ ein Codewort ist, sich also als Summe von $g(\chi), \chi g(\chi), \ldots, \chi^{n-k-1}g(\chi)$ darstellen lässt.
Ist $a_{n-k-1} = 0$, so ist

$$\chi \cdot x(\chi) \mod (\chi^n + 1) = a_{n-k-2}\chi^{n-k-1}g(\chi) + \cdots + a_1\chi^2 g(\chi) + a_0\chi g(\chi) \qquad (10.16)$$

offenbar wieder ein Codewort.
Ist hingegen $a_{n-k-1} = 1$, so ist

$$\chi \cdot x(\chi) \mod (\chi^n + 1) = \chi \cdot x(\chi) + \chi^n + 1 = a(\chi) \cdot \chi \cdot g(\chi) + \chi^n + 1. \qquad (10.17)$$

Nach Voraussetzung ist $g(\chi)$ Teiler von $\chi^n + 1$, es gibt also ein Polynom $h(\chi)$ vom Grad $n - k$, sodass $h(\chi) \cdot g(\chi) = \chi^n + 1$. Damit gilt

$$\chi \cdot x(\chi) \mod (\chi^n + 1) = a(\chi) \cdot \chi \cdot g(\chi) + h(\chi) \cdot g(\chi) = (\chi \cdot a(\chi) + h(\chi)) \cdot g(\chi), \qquad (10.18)$$

und da $\chi \cdot a(\chi) + h(\chi)$ höchstens vom Grad $n - k - 1$ ist, ist damit wieder eine Darstellung als Summe von $g(\chi), \chi g(\chi), \ldots, \chi^{n-k-1}g(\chi)$ gefunden, also ist $\chi \cdot x(\chi) \mod (\chi^n + 1)$ ein Codewort. □

Beispiel 10.7 (Wiederholungs- und Parity-Check-Code)

Für jedes n gilt

$$\chi^n + 1 = (\chi + 1) \cdot (\chi^{n-1} + \chi^{n-2} + \cdots + \chi + 1).$$

Es gibt also zu jedem n wenigstens zwei zyklische Codes mit den Generatorpolynomen

$$g_1(\chi) = (\chi + 1)$$
$$g_2(\chi) = (\chi^{n-1} + \chi^{n-2} + \cdots + \chi + 1).$$

Dabei beschreibt $g_1(\chi)$ den Parity-Check-Code mit der Generatormatrix

$$\boldsymbol{G_1} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 0 & \cdots & 1 & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

und $g_2(\chi)$ den Wiederholungscode mit der Generatormatrix

$$\boldsymbol{G_2} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}.$$

Beispiel 10.8

Das Polynom $g(x) = x^4 + x^3 + x^2 + 1$ ist ein Generatorpolynom eines Codes der Länge $n = 7$, da

$$
\begin{array}{rl}
(x^7 + & 1)/(x^4 + x^3 + x^2 + 1) = x^3 + x^2 + 1, \\
-(x^7 + x^6 + x^5 + & x^3) \\
\hline
x^6 + x^5 + & x^3 + 1 \\
-(x^6 + x^5 + x^4 + & x^2) \\
\hline
x^4 + x^3 + x^2 + & 1 \\
-(x^4 + x^3 + x^2 + & 1) \\
\hline
0 &
\end{array}
$$

also $g(x)$ Teiler von $x^7 + 1$ ist.
Gleichzeitig ist festzustellen, dass auch $x^3 + x^2 + 1$ ein Generatorpolynom eines Codes der Länge $n = 7$ ist.

10.2 Codierung

Um ein Nutzdatenwort $\boldsymbol{a}$ zu codieren, wird dieses analog zu den Codewörtern als Polynom $a(x)$ dargestellt, das maximal den Grad $l - 1 = n - k - 1$ besitzt.
Die einfachste Form der Codierung ist die Multiplikation

$$x(x) = a(x) \cdot g(x). \tag{10.19}$$

Beispiel 10.9

Es soll das Nutzdatenwort 101 mit dem Generatorpolynom $g(x) = x^4 + x^3 + x^2 + 1$ eines Codes der Länge $n = 7$ codiert werden.

$$
\begin{aligned}
x(x) &= a(x) \cdot g(x) \\
&= (x^2 + 1) \cdot (x^4 + x^3 + x^2 + 1) \\
&= (x^6 + x^5 + x^4 + x^2) + (x^4 + x^3 + x^2 + 1) \\
&= x^6 + x^5 + x^3 + 1
\end{aligned}
$$

Also lautet das zugehörige Codewort 1101001.

Leider führt die Codierung durch Multiplikation nicht auf eine systematische Codierung, das Codewort beginnt also nicht mit den Nutzdatenbits.
Wünschenswert wäre hingegen eine Darstellung der Form

$$x(\chi) = \chi^k a(\chi) + p(\chi). \tag{10.20}$$

mit einem Polynom $p(\chi)$ mit maximalem Grad $k-1$, sodass $x(\chi) \mod g(\chi) = 0$.
Für eine systematische Codierung ist also ein $p(\chi)$ vom Grad $k-1$ zu bestimmen, sodass

$$\left(\chi^k a(\chi) + p(\chi)\right) \mod g(\chi) = 0, \tag{10.21}$$

also

$$p(\chi) \mod g(\chi) = \chi^k a(\chi) \mod g(\chi), \tag{10.22}$$

und da der Grad $k-1$ von $p(\chi)$ kleiner ist als der Grad k von $g(\chi)$ kann die Modulo-Operation auf der linken Seite entfallen, also

$$p(\chi) = \chi^k a(\chi) \mod g(\chi). \tag{10.23}$$

Damit kann eine systematische Codierung gemäß

$$x(\chi) = \chi^k a(\chi) + \left(\chi^k a(\chi) \mod g(\chi)\right) \tag{10.24}$$

erfolgen.
Ein Codewort besteht also aus den l Nutzdatenbits, gefolgt von k Prüfbits, die als $\chi^k a(\chi) \mod g(\chi)$ bestimmt werden.
Hinweis: Die Berechnung der Prüfbits kann sehr effizient mittels eines rückgekoppelten Schieberegisters erfolgen. Dies soll hier aber nicht diskutiert werden.

Beispiel 10.10

Es soll das Nutzdatenwort 101 mit dem Generatorpolynom $g(\chi) = \chi^4 + \chi^3 + \chi^2 + 1$ eines Codes der Länge $n = 7$ codiert werden. Mit $\chi^4 a(\chi) = \chi^6 + \chi^4$ folgt

$$\begin{array}{rl}
(\chi^6 + \chi^4 &) \; /(\chi^4 + \chi^3 + \chi^2 + 1) = \chi^2 + \chi + 1, \\
-(\chi^6 + \chi^5 + \chi^4 + \chi^2) & \\
\hline
\chi^5 + \chi^2 & \\
-(\chi^5 + \chi^4 + \chi^3 + \chi) & \\
\hline
\chi^4 + \chi^3 + \chi^2 + \chi & \\
-(\chi^4 + \chi^3 + \chi^2 + 1) & \\
\hline
\chi + 1 &
\end{array}$$

Also lautet das zugehörige Codewort 1010011.

10.3 Decodierung

Das Polynom $h(\chi)$, mit dem $g(\chi) \cdot h(\chi) = \chi^n + 1$ erfüllt eine der Kontrollmatrix vergleichbare Funktion, da für jedes Codewort

$$x(\chi) \cdot h(\chi) \mod (\chi^n + 1) = 0 \tag{10.25}$$

gilt.
Für ein gestörtes Empfangswort $y(\chi)$ ergibt sich ein nur vom Fehlermuster abhängiges Syndrom

$$s(\chi) = y(\chi) \cdot h(\chi) \mod (\chi^n + 1). \tag{10.26}$$

Wie bei den allgemeinen linearen Codes lässt sich so eine Fehlerkorrektur anhand einer Syndromtabelle durchführen. Allerdings ist auch hier das Problem, dass die Syndromtabelle exponentiell mit k wächst und dadurch schnell inpraktikabel groß wird.

10.4 Cyclic Redundancy Checks (CRC)

- Einsatz in vielen Kommunikationssystemen zur Fehlererkennung: Ethernet, USB, MMC/SD, FlexRay, ZigBee, BlueTooth, PPP, IrDA, ITU G.704, ...
- Benutzung der systematischen Codierung

$$x(\chi) = \chi^k a(\chi) + \left(\chi^k a(\chi) \mod g(\chi)\right). \tag{10.27}$$

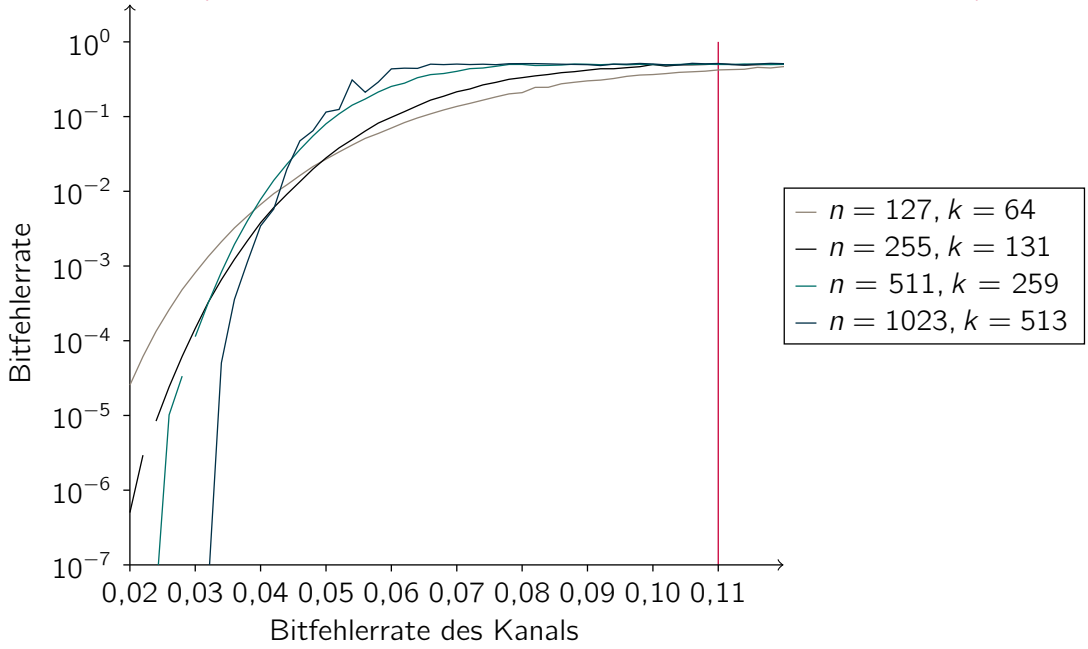
- Die Prüfbits werden auch als Prüfsumme (Checksum) bezeichnet.
- Typische Werte für die Anzahl k der angehängten Prüfbits sind 8, 16, 32.
- In den einschlägigen Standards finden Generatorpolynome Verwendung, die besonders gute Fehlererkennungseigenschaften haben.
- Ist $g(\chi)$ ein Generatorpolynom eines zyklischen Codes der Länge n (d.h. $g(\chi)$ teilt $\chi^n + 1$), so muss $l + k = n$ gelten – was tun, wenn die tatsächliche Länge der Nachricht $l_a \neq l$?

- Ist $l_a < n - k$, kann die Nachricht mit Nullen aufgefüllt werden. Da dies das Polynom $a(\chi)$ nicht ändert, brauchen diese bei der Berechnung aber nicht berücksichtigt zu werden. Auch eine Übertragung der zusätzlichen Nullen ist nicht nötig, da der Decoder ebenso verfahren kann.
  Es ist also ausreichend, die Prüfsumme $\chi^k a(\chi) \mod g(\chi)$ für die originale Nachricht zu berechnen und zu übertragen.
- Ist $l_a > n - k$, nutzt man aus, dass $(\chi^n + 1)^2 = \chi^{2n} + 1$, jeder Teiler von $\chi^n + 1$ (insbesondere $g(\chi)$) ist also auch Teiler von $\chi^{2n} + 1$. Es lässt sich also zu jedem l_a ein n finden, sodass $l_a \leq n - k$ und $g(\chi)$ ein Generatorpolynom eines Codes der Länge n ist. Dann ist wie oben zu verfahren.
- Tatsächlich ist für die Berechnung der Prüfsumme die Kenntnis von n nicht notwendig!

10.5 BCH- und RS-Codes

- BCH-Codes (Bose, Chaudhuri, Hocquenghem) erlauben durch einen speziellen Entwurf nicht nur eine Erkennung, sondern auch eine Korrektur von Fehlern im Decoder mit vertretbarem Aufwand.
- Der Entwurf erfolgt über die Wahl der Nullstellen α_i des Generatorpolynoms; die Parameter des Codes (n, k) lassen sich dabei vorgeben.
- Die Nullstellen liegen dabei nicht in $\{0, 1\}$, sondern in einem Galois-Feld, ähnlich wie Nullstellen von Polynomen mit reellen Koeffizienten im allgemeinen komplexwertig sind. Auf die Diskussion von Galois-Feldern soll an dieser Stelle verzichtet werden.
- Ein Codewort $x(\chi)$ hat an den gleichen Stellen wie $g(\chi)$ Nullstellen, d.h. $s_i = x(\alpha_i) = 0$.
- Für ein Empfangswort $y(\chi)$ stellt $s_i = y(\alpha_i)$ damit ebenfalls eine Form von Syndrom dar.
- Durch Lösen eines linearen Gleichungssystems lässt sich aus den s_i ein Polynom gewinnen, dessen Nullstellen mit den Fehlerstellen von $y(\chi)$ korrespondieren.
- Das Suchen dieser Nullstellen erlaubt somit eine Fehlerkorrektur.

Beispiel 10.11 (BCH-Code für einen symmetrisch gestörten Binärkanal)



- RS-Codes (Reed, Solomon) sind eine Variante der BCH-Codes, die nicht auf einzelnen Bits, sondern auf höherwertigen Symbolen (Blöcken von Bits) arbeiten.
- Dadurch erreichen RS-Codes sehr gute Fehlerkorrektureingeschaften; insbesondere erreichen sie zu vorgegebenen n, k den größtmöglichen minimalen Abstand $d(\mathcal{C})$. Außerdem erlauben sie die Korrektur relativ langer Folgen von Fehlern (Bursts).
- RS-Codes werden z.B. auf der CD zur Fehlerkorrektur eingesetzt.
- Leider erlauben die effizienten Decodieralgorithmen für BCH- und RS-Codes nicht die Ausnutzung von “weich” entschiedenen Bits.

10.6 Zusammenfassung

- Zyklische Codes sind eine Unterklasse der linearen Codes mit der Eigenschaft, dass zyklische Verschiebungen von Codewörtern wieder Codewörter ergeben.
- Dies erlaubt eine kompakte Beschreibung des Codes mittels eines Generatorpolynoms (anstelle einer Generatormatrix).
- Die systematische Codierung lässt sich effizient realisieren.
- Ebenso die Fehlererkennung; die Fehlerkorrektur hingegen ist vergleichbar aufwändig zu allgemeinen linearen Codes.
- Die CRC-Codes beschränken sich daher auf eine Fehlererkennung und werden zu diesem Zweck in diversen Kommunikationsprotokollen eingesetzt.
- Die BCH- und RS-Codes folgen einem speziellen Entwurf, der zu vorgegebenen n und k sehr leistungsfähige Codes liefert, für die spezielle Methoden zur Fehlerkorrektur existieren.