Algebraische Codierungstheorie und Compact Discs

El. Math. *51* (1996)

0013-6018/96/020089-13$1.50 + 0.20/0

Algebraische Codierungstheorie und Compact Discs

Dietmar Dorninger

Dietmar Dorninger wurde 1945 in Oberösterreich geboren. Er studierte Mathematik und Physik an der Universität Wien, wo er 1969 promovierte. Bereits 1976 wurde er Professor an der Technischen Universität Wien; seit 1992 ist er dort Vorstand des Institutes für Algebra und Diskrete Mathematik. Seine wissenschaftlichen Interessen gelten ausser den Anwendungen der Algebra insbesondere Fragen der mathematischen Biologie, gegenwärtig vor allem im Bereich der Zell- und Molekularbiologie. In der Freizeit beschäftigt er sich gerne mit zeitgenössischer Literatur.

1 Einleitung

Im Herbst 1982 wurde in Europa und Japan die Audio-CD eingeführt, im Frühling 1983 folgten die USA nach. Seither hat die CD einen unvergleichlichen Siegeszug um die Welt angetreten. Obgleich jedem die hervorragende Wiedergabequalität der CD bekannt

Im Jahr 1878 liess Thomas Alva Edison erstmals ein Gerät zur Tonaufzeichnung patentieren, den Phonographen. Die Tonspeicherung geschah rein mechanisch, als Speichermedium diente eine Wachs- bzw. Stanniolwalze. Es ist den Berichten leicht zu glauben, dass die Wiedergabequalität sehr zu wünschen übrig liess. Im Laufe der Zeit erreichte man mit Hilfe neuer Speichermedien und neuer Aufzeichnungsverfahren elektromagnetischer Art schrittweise Verbesserungen. Trotzdem vermochten die marktgängigen Produkte die hohen Ansprüche der Benutzer nicht immer zu befriedigen. 1982, fast genau 100 Jahre nach der Patentierung des Phonographen, wurde die *Compact Disc* eingeführt. Wurden bei allen bisherigen Verfahren die Tonsignale analog gespeichert, so verwendete das neue System erstmals eine Aufzeichnung in digitaler Form. Damit wurde eine praktisch perfekte Audiowiedergabe technisch machbar. Bei der digitalen Speicherung muss eine enorme Datenmenge auf kleinem Raum untergebracht werden, und dies konnte nur dank ganz neuer Entwicklungen in der Lasertechnik und in der Elektronik realisiert werden. Die hochkompakte Datenspeicherung machte es dann aber notwendig, im Gerät ein System einzubauen, das Speicher- und Ablesefehler zu erkennen und zu korrigieren vermag. Zu diesem Zweck wurden raffinierte mathematische Methoden der algebraischen Codierungstheorie herangezogen. – Wohl nur wenige Benutzer wissen, wie ihr CD-Player wirklich funktioniert. Wer möchte nicht mehr darüber erfahren? Im vorliegenden Beitrag gibt Dietmar Dorninger einen Überblick über die technischen und mathematischen Grundlagen, die in einem CD-Gerät eingesetzt werden. *ust*

zu sein scheint, ist den wenigsten aber bewußt, daß hierbei die Algebra eine wesentliche Rolle spielt. In welcher Weise algebraische Methoden bei der CD Verwendung finden und um welche algebraischen Hilfsmittel es sich handelt, soll im folgenden ausgeführt werden.

Zunächst werden an Hand der Gegenüberstellung der Übertragung von Bild und Ton das Problem der Quellencodierung und die Frage nach einer Möglichkeit der Korrektur von Fehlern bei der Übertragung besprochen. Dann werden Elemente der Codierungstheorie unter besonderer Berücksichtigung der Anforderungen bei CDs behandelt. Es folgt eine Vorstellung der CD von der technischen Seite her, wobei die Implementierung der bei CDs verwendeten Codes, sogenannte RS-Codes, im Mittelpunkt des Interesses steht. Wie sich die RS-Codes in das Gedankengebäude der algebraischen Codierungstheorie einordnen, ist Gegenstand des letzten Abschnitts.

Beethovens 9. Symphonie stand dafür Pate, daß die Spieldauer einer Audio-CD 74 Minuten beträgt. Die Theorie der endlichen Körper und in ihrem Gefolge die algebraische Codierungstheorie tragen dazu wesentlich bei, daß es möglich ist, Beethovens Neunte in so hervorragender Wiedergabequalität von einer CD zu hören.

2 Codierung von Bild und Ton

Fast jeder von uns hat schon einmal eine Audio-CD in der Hand gehabt, und viele kennen die gestochen scharfen Bilder aus dem Weltraum, die uns via Fernsehen in unsere Wohnzimmer geliefert werden. (Abb. 1 zeigt eine CD von 12 cm Durchmesser, in Abb. 2 ist ein Bild vom Mars wiedergegeben, das durch die Marssonde Mariner 7 aufgenommen wurde.) Fragt man sich, was CD und Bilder vom Mars gemeinsam haben, so erkennt man sehr schnell, daß in jedem Fall Informationen über einen Kanal übertragen werden, der starken Störungen unterworfen sein kann, daß die Informationen aber zumeist so gut wie fehlerfrei reproduziert werden. Bei den enormen Entfernungen, die ein Signal aus dem Weltall zurücklegt, ist ganz offensichtlich, daß stark störende Einflüsse wirksam werden können, bei der CD hingegen muß man sich vor Augen halten, daß Einschlüsse beim Prägen, Verunreinigungen, Fingerabdrücke und Kratzer ähnliche Auswirkungen haben können.

Jeder digitalen Übertragung von Bild und Ton geht eine *Quellencodierung* über einem gegebenen *Alphabet* (= Zeichenvorrat) A voraus. Bilder werden zumeist in einzelne Bildpunkte zerlegt, welche sequentiell angeordnet werden. Wählt man $A = \{0, 1\}$ und ordnet jedem Bildpunkt eine Graustufe oder Farbe in Form eines r-tupels aus 0 und 1 zu, so wird der Bildinhalt durch eine Folge über A der Länge r mal Anzahl der Bildpunkte repräsentiert. (Bei den Aufnahmen durch die Mariner-Sonden 6 und 7 wurde jedes Bild in 658240 Punkte zerlegt, und jeder Punkt hatte eine Helligkeitsabstufung zwischen 1 und 2^8, welche durch $r = 8$ Bits wiedergegeben wurde, so daß pro Aufnahme etwa 5 Millionen Bits an Information notwendig waren.) Die durch die Quellencodierung eines Bildes erhaltene Folge stellt dann die Grundlage für algebraische Verfahren der Fehlererkennung und Fehlerkorrektur dar, durch welche die Folge in eine neue (längere) Folge über A übergeführt wird, welche man überträgt. Die übertragenen Bits heißen *Kanalbits*.

Bei der CD ist alles wesentlich komplizierter: Bei der Quellencodierung wird das Audiosignal 44100 mal/sec (pro Audiokanal) abgetastet, und der beim Abtasten gefundene

Abb. 1

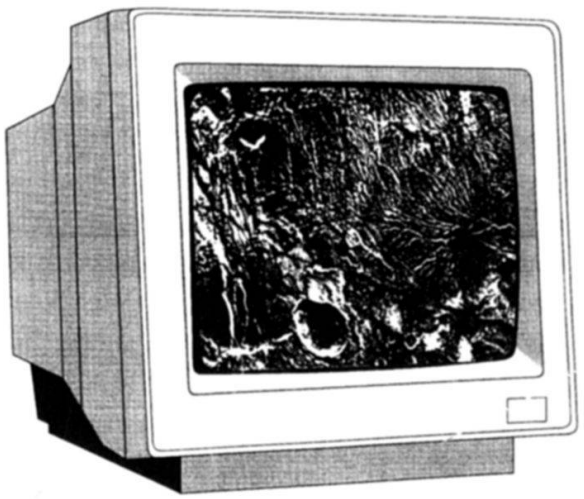


Abb. 2

Wert wird in eines von $2^{16} = 65536$ Niveaus eingeordnet (siehe Abb. 3). Jedes Niveau wird durch zwei Bytes (1 Byte = 8 Bits) charakterisiert, so daß ein Datenstrom aus Bytes entsteht, d.h., das Alphabet A, über dem die Quellencodierung erfolgt, besteht aus 256 Zeichen, nämlich den $2^8 = 256$ verschiedenen Bytes. (Um das analoge Signal aus der digitalen Information eindeutig rückgewinnen zu können, muß die Abtastfrequenz mindestens doppelt so groß sein wie die größte vorkommende Frequenz; siehe z.B. [8]. Damit steht pro Kanal eine Bandbreite von 20 kHz zur Verfügung, was dem Hörbereich eines Menschen entspricht. Höhere Frequenzen müssen vorweg ausgefiltert werden.)

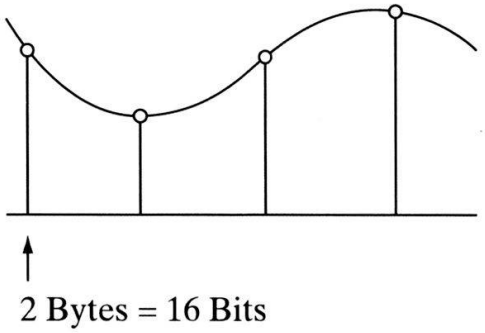


Abb. 3

Auf den die Audioinformation repräsentierenden Datenstrom aus Bytes werden anschließend mehrere, in den nachstehenden Abschnitten ausführlich beschriebene Verfahren zur Fehlerentdeckung und -korrektur angewandt, wodurch nach Hinzufügen von Synchronisationsdaten aus der ursprünglichen Folge wieder eine Folge aus Bytes entsteht. Um letztere auf der CD technisch zu realisieren (siehe Abschnitt 4), wird sie allerdings einer weiteren Codierung unterworfen, welche *Kanalcodierung* heißt.

Bei der Kanalcodierung wird jedes Byte in eine Folge von Bits übergeführt, wobei aber, nicht wie zu erwarten, jedes Byte durch 8 Bits dargestellt wird, sondern durch 14 Bits, wodurch man erreichen kann, daß zwischen zwei Einsen mindestens zwei, aber maximal 10 Nullen zu stehen kommen, eine Forderung, die man durch Einschieben von

drei weiteren "Verbindungsbits" zwischen zwei 14-tupeln auf den gesamten Datenstrom ausdehnt. Diese Erweiterung auf Bit-Ebene, welche für die Datendichte auf der CD von wesentlicher Bedeutung ist (siehe Abschnitt 4), wird *EF-Modulation* (eight to fourteen modulation) genannt. Die EF-Modulation baut darauf auf, daß es 267 14-tupel mit der Eigenschaft gibt, daß zwischen zwei Einsen mindestens zwei, maximal aber 10 Nullen stehen. Die Zuordnung der 256 Bytes zu 256 dieser 267 14-tupel erfolgt dadurch, daß an Hand eines ROM-Wörterbuches den Bytes 14-tupel zugewiesen werden. (Von den 11 nicht verwendeten 14-tupeln werden zwei als Synchronisationssymbole benützt; auch eines der drei Verbindungsbits findet hierfür Verwendung.) – In Abb. 4 ist der Weg vom Audiosignal bis hin zum Lochmuster auf der CD (siehe Abschnitt 4) wiedergegeben. Schon hier sei darauf verwiesen, daß genau bei jedem Wechsel zwischen einer Vertiefung und einer Erhebung eine 1 steht.

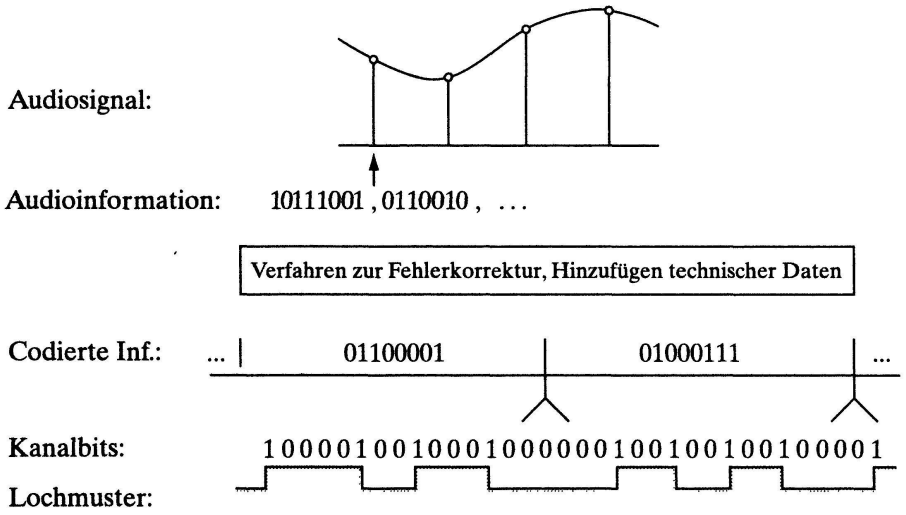


Abb. 4

Zurück zur Quellencodierung über dem Alphabet A: So wie bei natürlichen Sprachen wird der Datenstrom über A (die "Nachricht") in *Wörter*, welche auch *Blöcke* genannt werden, unterteilt. Im Gegensatz zu natürlichen Sprachen nimmt man aber an, daß jedes Wort gleichviele Symbole besitzt. Die Anzahl der Symbole eines Wortes (bzw. Blocks) wird als *Länge* des Worts (bzw. *Blocklänge*) bezeichnet.

Warum verstehen wir den verballhornten Satz "Viela Leute heute tind am dif ETH Zsrich ... kommen" völlig richtig? Die Antwort ist, weil unsere Sprache a) Redundanz besitzt und b) nach grammatikalischen Regeln konstruiert ist, was gemeinsam die Rekonstruktion des Textes ermöglicht. Dieselben beiden Prinzipien wendet man nun an, um bei Übertragungen von Datenströmen über einen störanfälligen Kanal Fehler zu entdecken bzw. zu korrigieren[1]). Um Redundanz zu erreichen, fügt man zu jedem Nachrichtenwort der Länge k weitere $n - k$ Symbole als "Prüfstellen" hinzu, so daß aus einem Nachrichtenwort der Länge k ein *Codewort* der Länge n entsteht. O.B.d.A. nehmen wir an, daß die Prüfstellen alle nach dem (oder vor das) Nachrichtenwort zu stehen kommen. (Bei der Übertragung der Bilder vom Mars war z.B. $k = 6$ und $n = 32$, d.h., zu jedem

1) Der Vergleich stammt von H.K. Kaiser (mündl. Mitteilung)

Nachrichtenwort der Länge 6 wurden 26 Prüfbits hinzugefügt; bezüglich der CD siehe Abschnitt 4.) Als Analogie zur Grammatik bei natürlichen Sprachen kann gesehen werden, daß man dem Alphabet A eine algebraische Struktur aufprägt, mit deren Hilfe man A^k (Menge der k-tupel von Elementen aus A) und A^n (Menge der n-tupel von Elementen aus A) ebenfalls zu algebraischen Strukturen machen kann. A^k stellt die Menge aller Nachrichtenwörter dar, und A^n ist eine Obermenge der Menge C aller Codewörter. Die injektive Abbildung $f_C : A^k \to A^n$, welche jedem Nachrichtenwort $(a_1, \ldots, a_k)$ das zugehörige Codewort $(a_1, \ldots, a_k, c_{k+1}, \ldots, c_n)$ zuordnet, heißt *Codierungsfunktion*, und n wird als *Länge* des Codes bezeichnet. Ein Code der Länge n, genauer ein (n, k)-*Code*, ist dann nichts anderes als die Teilmenge $C = \{f_C(a_1,\ldots,a_k) | (a_1,\ldots,a_k) \in A^k\}$ von A^n.

Tab. 1

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

In den meisten Fällen verlangt man, daß A ein *endlicher Körper* GF(q) ist. (GF(q): *Galoisfeld* mit q Elementen; siehe z.B. [2], [5]). Im einfachsten Fall ist $q = 2$. Dann ist GF(q) der Restklassenring modulo 2 und hat die in Tab. 1 dargestellte Additions- und Multiplikationstafel. (Da in GF(2) gilt $1 + 1 = 0$, ist $-x = +x$ für jedes x.) Ist $A = GF(2)$, so heißt der (n, k)-Code *binär*. Der bei der Übertragung der Bilder vom Mars verwendete Code ist binär. Bei der CD hingegen nimmt man für A das Galoisfeld GF(256). (Ein solches Galoisfeld existiert, denn genau zu den Primzahlpotenzen $q = p^m$, p prim, $m \geq 1$, gibt es einen (bis auf Isomorphie eindeutig bestimmten) endlichen Körper mit p^m Elementen, nämlich das Galoisfeld GF(p^m), und $256 = 2^8$; siehe z.B. [2].)

Wir bemerken, daß man A^k und A^n für jeden Körper A als Vektorräume über A auffassen kann und daß bei den von uns im folgenden ausschließlich betrachteten *linearen Codes*, bei denen f_C als lineare Abbildung vorausgesetzt ist, C zu einem Untervektorraum wird. Eine wesentliche Konsequenz davon ist z.B., daß mit zwei Codewörtern auch deren (komponentenweise definierte) Summe und Differenz wieder Codewörter sind, eine andere, daß man die Codierung und Decodierung mit Hilfe von Matrizen beschreiben kann, worauf wir hier aber nicht weiter eingehen wollen. (Für eine diesbezügliche Einführung siehe z.B. [1].)

3 Elemente der algebraischen Codierungstheorie

Die folgenden Ausführungen orientieren sich an binären Codes, sämtliche Aussagen sind aber für Codes über jedem beliebigen Alphabet $A = \mathrm{GF}(q)$ gültig.

Zunächst zwei Beispiele:

1) *Quersummenprüfcode* über GF(2): Bei diesem Code ist $n = k+1$ und $f_C(a_1, \ldots, a_k) = (a_1, \ldots, a_k, \sum_{i=1}^{k} a_i)$, wobei $\sum_{i=1}^{k} a_i$ in GF(2) zu bilden ist. Wählen wir z.B. $k = 2$, so ist f_C die Funktion, die in Tab. 2 wiedergegeben ist, und wir können C wie folgt graphisch veranschaulichen: Wir stellen die Elemente von A^3, so wie aus Abb. 5 hervorgeht, als Gitterpunkte in einem dreidimensionalen Koordinatensystem dar. Die durch volle Kreise

$$(0,0) \to (0,0,0)$$
$$(0,1) \to (0,1,1)$$
$$(1,0) \to (1,0,1)$$
$$(1,1) \to (1,1,0)$$

Tab. 2.

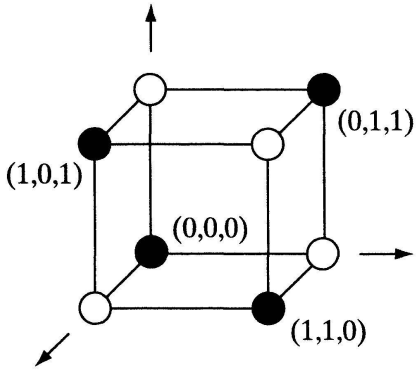


Abb. 5

gekennzeichneten Knoten repräsentieren dann die Elemente des Codes C. Man sieht dabei: Tritt höchstens ein Fehler bei der Übertragung der Codewörter auf, so wird dieser Fehler erkannt; das übertragene Wort gehört dann nämlich nicht zur Menge C.

Der Quersummenprüfcode wird beim Datentransfer zwischen den Komponenten eines Computers verwendet.

2) *r-fach Wiederholungscode*: $n = rk$, und f_C ist gegeben durch $f_C(a_1, \ldots, a_k) = (a_1, \ldots, a_k, a_1, \ldots, a_k, \ldots, a_1, \ldots, a_k)$ ($a_1, \ldots, a_k$ r-mal wiederholt). Für $k = 1$ und $r = 3$ erhalten wir über GF(2) die Codierungsfunktion: $(0) \to (0,0,0)$ und $(1) \to (1,1,1)$. Wie die Veranschaulichung dieses Codes in Abb. 6 zeigt, kann der Code bis zu zwei Fehler erkennen und einen Fehler korrigieren. $(0,1,1)$ wird z.B. zu $(1,1,1)$ korrigiert – sofern wir von dem nachstehenden *Prinzip zur Decodierung* ausgehen, welches wir im folgenden stets annehmen wollen:

Das zum empfangenen, eventuell fehlerbehafteten Wort nächstliegende Codewort ist dasjenige Codewort, welches von der Nachrichtenquelle ausgesandt wurde.

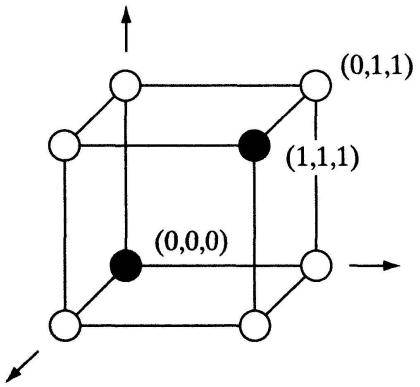


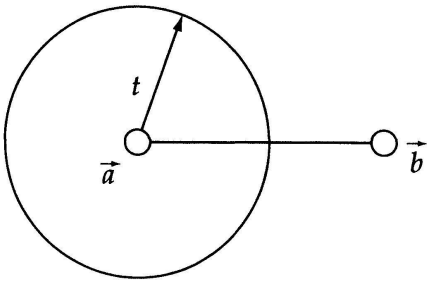
Abb. 6

Aus dem auf die beschriebene Weise bestimmten Codewort kann die ursprüngliche Nachricht abgelesen werden. Voraussetzung für die Anwendbarkeit des Decodierungs-Prinzips ist allerdings, daß die Wahrscheinlichkeit eines Übertragungsfehlers $a \to b$ für alle $a, b \in A$ gleich groß ist und daß Übertragungsfehler unabhängig voneinander erfolgen. (Wir werden daher bei der CD darauf zu achten haben, "Bündelfehler", wie sie dort zumeist auftreten, mit geeigneten Verfahren in den Griff zu bekommen.) Ferner benötigen wir natürlich einen geeigneten Abstandsbegriff.

Wir definieren: Der *(Hamming-)*Abstand $d(\vec{a},\vec{b})$ von $\vec{a} = (a_1,\ldots,a_n)$ und $\vec{b} = (b_1,\ldots,b_n)$ ist die Anzahl der Stellen $i \in \{1,2,\ldots,n\}$ mit $a_i \neq b_i$. Z.B. ist $d((1,1,0,1),(0,1,1,1)) = 2$.

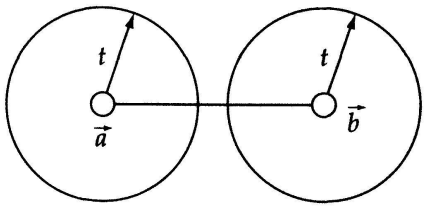
An Hand der Abbildungen 7 und 8 ist unschwer einzusehen (was auch formal leicht zu beweisen ist), daß stets gilt:

Ein Code kann genau dann jede Kombination von t oder weniger Fehlern entdecken bzw. korrigieren, wenn der Hamming-Abstand zwischen zwei beliebigen verschiedenen Codewörtern mindestens $t+1$ bzw. $2t+1$ ist.



$t \leq d(\vec{a},\vec{b}) - 1$

Abb. 7



$2t \leq d(\vec{a},\vec{b}) - 1$

Abb. 8

Bezeichnen wir den minimalen Abstand zwischen zwei verschiedenen Codewörtern, die sogenannte *Minimaldistanz* des Codes, mit d, so muß also für die Entdeckung von bis zu t Fehlern gelten $t \leq d-1$, und für die Fehlerkorrektur $t \leq \left[\frac{d-1}{2}\right]$. ($[x]$ bezeichnet die nächstkleinere ganze Zahl zu x.) Da bei einem (n,k)-Code stets Nachrichtenwörter existieren, die sich in genau einer Komponente unterscheiden, und die zu den Nachrichtenwörtern hinzugefügten $n-k$ Prüfstellen im günstigsten Fall sich in allen Komponenten unterscheiden, gilt für die Minimaldistanz d eines (n,k)-Codes $d \leq n-k+1$. Bei allen Codes ist nun wesentlich, daß d möglichst groß ist, damit der Code gute Fehlererkennungs- und -korrektureigenschaften hat, und eine große Minimaldistanz ist dadurch zu erreichen, daß man sehr viele Prüfstellen zu den Nachrichtenwörtern hinzufügt, d.h., daß $n-k$ groß wird. Im Gegensatz dazu muß man aber insbesondere bei den bei CDs verwendeten Codes trachten, $n-k$ möglichst klein zu halten, damit man soviel wie nur möglich an Audio-Information auf der CD unterbringen kann. Beide einander widersprechenden Forderungen lassen sich am besten vereinen, falls $d = n-k+1$ ist. In diesem Fall heißt der Code optimal. – Für einen optimalen (n,k)-Code gilt dann, daß er bis zu $n-k$ Fehler entdecken und bis zu $\left[\frac{n-k}{2}\right]$ Fehler korrigieren kann.

Ein weiteres Anliegen an bei CDs verwendeten Codes ist, daß die Decodierung, für welche es bei linearen Codes gute systematische Verfahren gibt, besonders schnell vonstatten gehen soll. Dies ist am besten mit Hilfe von sogenannten Polynomcodes zu erreichen, bei denen man ausnützt, daß die Multiplikation und Division von Polynomen sehr einfach und effektiv mit Hilfe von Schieberegistern technisch zu realisieren ist.

Ausgangspunkt der Definition von *Polynomcodes* ist, daß man jedem Vektor $\vec{\omega} = (\omega_0,\omega_1,\ldots,\omega_{m-1}) \in A^m$ (man beachte, daß wir jetzt die Komponenten von 0 bis $m-1$ indizieren) in umkehrbar eindeutiger Weise ein Polynom $p_{\vec{\omega}}(x) = \omega_0 + \omega_1 x + \ldots +$

$\omega_{m-1}x^{m-1}$ zuordnen kann. Ein (n,k)-Polynomcode C über GF(q) wird dann auf folgende Weise festgelegt: Man wählt ein Polynom $g(x)$ vom Grad $n-k$ als sogenanntes *Generatorpolynom* und berechnet zu jedem Nachrichtenwort $\vec{a}=(a_0,a_1,\ldots,a_{k-1})\in A^k$ das zugehörige Codewort $\vec{c}=f_C(\vec{a})$ bzw. das entsprechende Polynom $p_{\vec{c}}(x)$, indem man $p_{\vec{c}}(x)=p_{\vec{a}}(x)\cdot x^{n-k}-R_{g(x)}(p_{\vec{a}}(x)\cdot x^{n-k})$ bildet, wobei $R_{g(x)}$ den Rest bei Division durch $g(x)$ bedeutet. Da der Grad von $p_{\vec{a}}(x)\leq k-1$ ist, folgt für Polynome $p_{\vec{a}}(x)$ ungleich dem Nullpolynom (welchem als "Codepolynom" durch obige Vorschrift das Nullpolynom zugeordnet wird), daß gilt: $n-k\leq$ Grad von $p_{\vec{a}}(x)\cdot x^{n-k}\leq k-1+n-k=n-1$, d.h., $f_C(a_0,a_1,\ldots,a_{k-1})$ ist von der Gestalt $(c_0,c_1,\ldots,c_{n-k-1},a_0,a_1,\ldots,a_{k-1})$. Mit anderen Worten: Die Prüfstellen kommen vor das Nachrichtenwort zu stehen. Im übrigen sei bemerkt, daß auf Grund der Konstruktion von $p_{\vec{c}}(x)$ jedes "Codepolynom" ein Vielfaches des Generatorpolynoms $g(x)$ ist, und weiters heben wir hervor, daß jeder Polynomcode ein linearer Code ist (was allerdings nicht unmittelbar einzusehen ist).

Beispiel: Beim (7,4)-Polynomcode über GF (2) mit dem Generatorpolynom $g(x)=1+x+x^3$ ist das Nachrichtenwort $\vec{a}=(0,1,0,1)$ zu codieren. $p_{\vec{a}}(x)=x+x^3$, $p_{\vec{a}}(x)x^{n-k}=x^4+x^6$, und $R_{g(x)}(x^4+x^6)=x+1$. Also ist $p_{\vec{c}}(x)=x^4+x^6-x-1$, was wegen $-a=+a$ in GF(2) zu $p_{\vec{c}}(x)=1+x+x^4+x^6$ führt. Damit ist das zu (0,1,0,1) gehörige Codewort gleich (1,1,0,0,1,0,1). (Die Prüfstellen sind 1,1,0.)

Um unter den Polynomcodes, für welche sehr effektive Algorithmen zur Codierung und Decodierung zur Verfügung stehen, solche mit guten Fehlerkorrektureigenschaften konstruieren zu können (also insbesondere, um optimale Codes zu finden), ist ihre Klasse noch zu allgemein. Man beschränkt sich hierbei auf sogenannte zyklische Codes:

Ein linearer (n,k)-Code heißt *zyklisch*, wenn mit $\vec{c}=(c_1,c_2,\ldots,c_n)$ auch alle jene Wörter in C liegen, welche durch zyklische Vertauschung der Symbole entstehen, also $(c_n,c_1,c_2,\ldots,c_{n-1})$, $(c_{n-1},c_n,c_1,c_2,\ldots,c_{n-2})$,.... Man kann beweisen, daß jeder zyklische lineare Code ein Polynomcode ist und daß ein (n,k)-Poynomcode genau dann zyklisch ist, wenn sein erzeugendes Polynom $g(x)$ das Polynom x^n-1 teilt. Damit ist die in Frage kommende Menge von Generatorpolynomen $g(x)$ wesentlich eingeschränkt (aber immer noch sehr umfangreich). Wie wir in Abschnitt 5 genauer ausführen werden, kann man unter den Teilern von x^n-1 Generatorpolynome $g(x)$ finden, welche zu optimalen Codes über GF(q) von vorgegebener Länge n und Minimaldistanz d führen und welche sich noch dazu gut zur Korrektur von Fehlerbündeln eignen. Zu solchen Codes zählen insbesondere die *Reed-Solomon-Codes (RS-Codes)* und die daraus hervorgehenden *verkürzten RS-Codes* (siehe Abschnitt 5). Letztere finden bei der CD Anwendung.

4 Compact Discs

Zunächst einige technische Aspekte: Der Durchmesser einer CD ist 12 cm, so daß die Disc mit einer Hand in den CD-Player eingeführt und wieder herausgenommen werden kann. Überdies kann der Player klein sein. Auf einer Seite der Disc befindet sich, geschützt durch eine 1,2 mm dicke lichtdurchlässige Schicht, eine spiralförmig angeordnete Spur, welche eine Folge von verschieden langen Vertiefungen bzw. Erhebungen enthält (siehe Abb. 9. – Abb. 9 sowie die beiden folgenden Abbildungen sind dem Konferenzbericht [7] entnommen). Die minimale Länge einer Vertiefung ist $9\cdot 10^{-4}$mm, der Abstand zwischen zwei Spurringen beträgt $1,6\cdot 10^{-3}$mm. Die Disc wird beim Abspielen durch

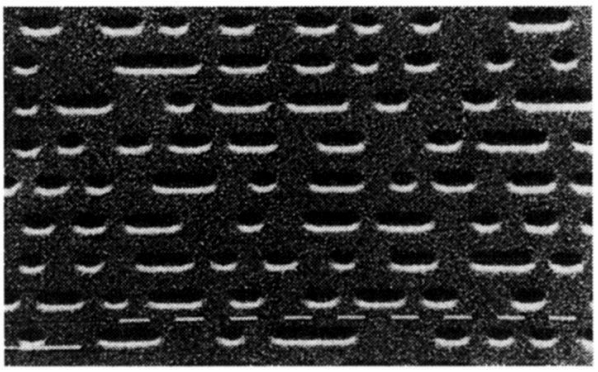


Abb. 9

einen Laserstrahl, der durch ein Servosystem exakt in der Spur gehalten wird, abgetastet (siehe Abb. 10). Fällt der Lichtstrahl auf eine Erhebung, wird das Licht fast total reflektiert und der Strahl erreicht in fast voller Stärke eine Photodiode; fällt der Lichtstrahl in eine Vertiefung (Tiefe $\approx \frac{1}{4}$ der Wellenlänge des Lichtes), so wird er nur sehr wenig reflektiert. Damit Fingerabdrücke, Staubpartikel usw. den Abtastvorgang möglichst wenig beeinträchtigen, verjüngt sich der Lichtstrahl von 0,7 mm an der Oberfläche der Schutzschicht auf 10^{-3}mm bei Erreichen der Spur (siehe Abb. 11). Jedesmal wenn ein Wechsel zwischen einer Vertiefung und einer Erhebung stattfindet, d.h., wenn von der Photodiode ein Wechsel zwischen sehr starker und geringfügiger Reflexion registriert wird, wird eine Eins angenommen, dazwischen Nullen. Die Spur stellt den Informationsstrom aus Nullen und Einsen dar, der durch die in Abschnitt 2 erklärte Kanalcodierung entsteht. Wie dort ausgeführt, bewirkt die EF-Modulation, daß zwischen zwei Einsen mindestens zwei, aber höchstens 10 Nullen zu stehen kommen, was ein Bit-Muster mit einer geringeren Anzahl von Übergängen zwischen 0 und 1 als ohne EF-Modulation entstehen läßt. Dadurch ist auf der CD eine kleinere Anzahl von Vertiefungen erforderlich, was bedeutet, daß die Datendichte auf der CD erhöht werden kann, wodurch eine längere Spielzeit erreicht wird. (Darüber hinaus werden bei der Manipulation des Frequenzspektrums durch die EF-Modulation noch einige weitere Zielsetzungen verfolgt; siehe z.B. [4] und [8].)

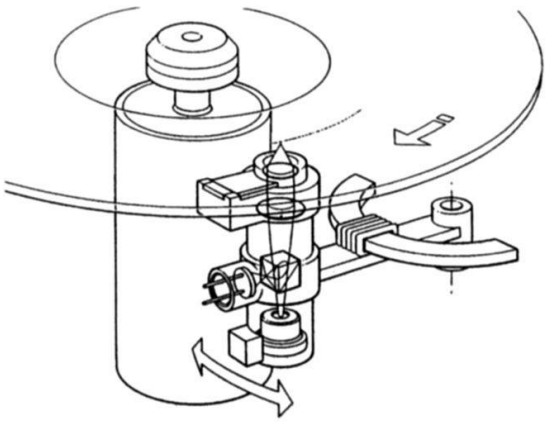


Abb. 10

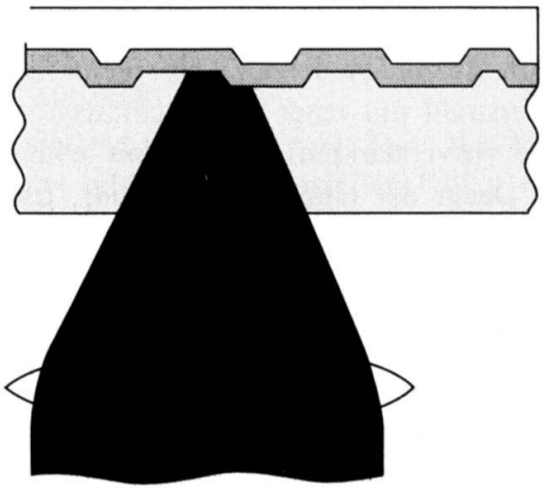


Abb. 11

Wir beschreiben nun im Detail, wie die Folge der Kanal-Bits aus der ursprünglichen Folge jener Bytes entsteht, welche man durch Digitalisierung der Audio-Information erhält (wobei jeweils zwei Bytes vom linken und dann zwei Bytes vom rechten Kanal genommen werden). Zunächst wird der Datenstrom von Bytes in Wörter der Länge 24, welche hier *Frames* heißen, unterteilt. Dann wird auf diese Frames der Wortlänge 24 ein (28,24)-Code C_1 über GF(256) angewandt, und zwar ein verkürzter Reed-Solomon-Code. Dieser ist optimal und kann daher bis zu $\left[\frac{n-k}{2}\right] = 2$ Byte-Fehler korrigieren und bis zu $n-k = 4$ Byte-Fehler erkennen. Die Korrekturmöglichkeit von zwei Byte-Fehlern bedeutet auf Bit-Ebene, daß im günstigsten Fall ein Fehlerbündel von bis zu 16 und im schlechtesten Fall ein Fehlerbündel von bis zu 9 aufeinanderfolgenden Bits korrigiert werden kann. Im nächsten Schritt wird der mit C_1 codierte Datenstrom auf Byte-Ebene neu organisiert. Aus jeweils 28 hintereinanderfolgenden Codewörtern $c_1, c_2, \ldots, c_{28}$ werden 28 neue Codewörter $c_1^*, c_2^*, \ldots, c_{28}^*$ dadurch gewonnen, daß aus $c_1, c_2, \ldots, c_{28}$ die ersten Buchstaben entnommen werden und damit c_1^* gebildet wird, dann aus $c_1, c_2, \ldots, c_{28}$ die zweiten Buchstaben entnommen werden und c_2^* gebildet wird, u.s.f. Dieser Vorgang heißt *Interleaving* und dient dem Zweck, Fehlerbündel, wie sie vornehmlich bei CDs zu erwarten sind, möglichst auseinanderzureißen.

Auf den neu organisierten Datenstrom wird nun nochmals ein verkürzter Reed-Solomon-Code angewendet, und zwar ein (32,28)-Code über GF(256), den wir mit C_2 bezeichnen. Auch der Code C_2 kann Fehler bis zu zwei Bytes korrigieren und bis zu vier Bytes erkennen. Nach der Codierung durch C_2 wird der Datenstrom noch durch Abspielinformationen ergänzt, indem jedem Wort der Länge 32 eines von zwei bestimmten Bytes (meist mit P und Q bezeichnet) angefügt wird. Der nächste Schritt ist dann die EF-Modulation, bei der aus den 33 Bytes 33×17 Kanalbits werden (Aufblähen eines jeden Bytes auf 14 Bits und Einfügen von drei Verbindungsbits). Zu den 33×17 Kanalbits kommen schließlich noch jeweils 27 Synchronisations-Bits, mit deren Hilfe gleichsam wie mit einer dem Datenstrom innewohnenden Uhr die Bitrate beim Auslesen der Daten von der CD bestimmt wird, so daß schlußendlich ein Frame von ursprünglich 24 Bytes Länge aus 588 Kanalbits besteht.

Beim *Abspielen* der CD geschieht folgendes: Als erstes werden von der durch den Laserstrahl mit einer Durchschnittsgeschwindigkeit von 4,3 Megabits/sec abgetasteten (und vorverstärkten) Information jeweils die Synchronisationsdaten ausgelesen, womit die Dauer der einzelnen Kanalbits, codierten Bytes bzw. Frames festgelegt wird und wodurch auch die Umdrehungsgeschwindigkeit der CD (durch ein Referenzsignal zum Datenpuffer; siehe später) kontrolliert wird. Anschließend wird im *Demodulator* die EF-Modulation rückgängig gemacht, und die Abspielinformationen werden dem "Display" bzw. "Memory" zugänglich. Dann wird der Datenstrom, der nur noch codierte Audioinformation enthält, in den *Datenpuffer* gelenkt, durch den sichergestellt wird, daß der Datenstrom höchst gleichmäßig, d.h. mit der Genauigkeit eines Quarzkristalles, weiterfließt. Der Datenpuffer steuert auch die Umdrehungsgeschwindigkeit der CD, welche im Gegensatz zum Schallplattenspieler aber innerhalb gewisser Toleranzen von untergeordneter Bedeutung ist. Das System ist so ausgelegt, daß der Datenpuffer im Durchschnitt halb voll ist.

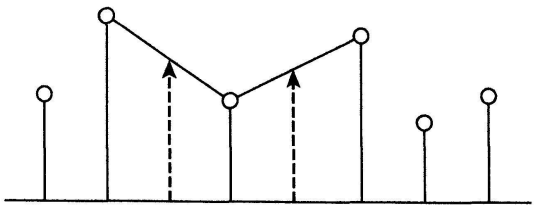


Abb. 12

Als nächstes folgen die *Decodierung* mittels C_2, die *Umkehrung* des *Interleaving* und die *Decodierung* mittels C_1. Dabei werden Fehler, soweit dies möglich ist, ausgebessert und erkannt, bzw., es wird festgestellt, daß eine Decodierung (etwa auf Grund eines unsicheren Erkennens von Bits) nicht erfolgen kann. Bytes, die als fehlerhaft oder unverläßlich eingestuft wurden, werden dem *Interpolator* zugeführt, wo sie linear interpoliert werden (siehe Abb. 12). Schließlich wird der digitale Informationsstrom wieder in Audiosignale umgewandelt. Das Endprodukt ist die bekannte ausgezeichnete Wiedergabequalität von CDs. Bezüglich genauerer Informationen verweisen wir auf [8].

5 Reed-Solomon-Codes

Bei der Lösung des Problems, Codes mit vorgegebenen Fehlerkorrektureigenschaften zu konstruieren, konzentriert man sich vor allem auf zyklische Linearcodes, welche stets Polynomcodes sind und sich daher sehr gut implementieren lassen.

Für den Fall, daß nur $t = 1$ Fehler korrigiert werden muß, haben sich die *binären Hamming-Codes* bewährt. Es sind dies $(2^m - 1, 2^m - 1 - m)$-Polynomcodes über GF(2), bei denen das Generatorpolynom $g(x)$ das Minimalpolynom $M_\alpha(x)$ eines primitiven Elements α von GF(2^m) ist. (Jedes Element $\gamma \neq 0$ eines Galoisfeldes GF(p^m), p prim, ist Nullstelle eines Polynoms mit Koeffizienten aus GF(p), und unter diesen Polynomen gibt es stets ein eindeutig bestimmtes Polynom $M_\gamma(x)$ von minimalem Grad und mit Koeffizient 1 bei der höchsten Potenz von x. $M_\gamma(x)$ heißt das *Minimalpolynom* von γ; es ist Teiler von $x^{p^m-1} - 1$. – Ein *primitives Element* α von GF(p^m) ist nichts anderes als ein erzeugendes Element der multiplikativen Gruppe der Elemente $\neq 0$ von GF(p^m); sein Minimalpolynom $M_\alpha(x)$ hat den Grad m; siehe z.B.[5].)

Beispiel für einen binären Hammingcode der Länge $n = 7$: Wegen $2^m - 1 = 7$ ist $m = 3$. Das Polynom $g(x) = x^3 + x + 1$ über GF(2) ist, wie man zeigen kann, Minimalpolynom eines primitiven Elements α von GF(8). $g(x)$ erzeugt daher einen zyklischen (7,4)-Polynomcode über GF(2), welcher einen Fehler korrigiert.

Ist α ein primitives Element von GF(p^m) und sind $M_{\alpha^i}(x)$ für $i = 1, 2, \ldots$ die Minimalpolynome der Elemente α^i von GF(p^m), dann gilt für den speziellen Fall $p = 2$, daß $M_{\alpha^i}(x) = M_{\alpha^{2i}}(x)$ für $i = 1, 2, \ldots$. Damit ergibt sich für das erzeugende Polynom $g(x) = M_\alpha(x)$ eines binären Hamming-Codes die folgende im Hinblick auf eine spätere Verallgemeinerung benötigte Darstellung: $g(x) = \text{k.g.V.}(M_\alpha(x), M_{\alpha^2}(x))$, wo k.g.V. abkürzend für "kleinstes gemeinsames Vielfaches" steht.

Sucht man für $t = 2$ binäre zyklische (n, k)-Polynomcodes der Länge $n = 2^m - 1$, welche t oder weniger Fehler korrigieren, so kann man in Verallgemeinerung des Falls $t = 1$ beweisen, daß $g(x) = \text{k.g.V.}(M_\alpha(x), M_{\alpha^2}(x), M_{\alpha^3}(x), M_{\alpha^4}(x)), = \text{k.g.V.}(M_\alpha(x), M_{\alpha^3}(x))$ das Gewünschte leistet, und für allgemeines t erhält man einen binären zyklischen (n, k)-Polynomcode der Länge $n = 2^m - 1$, der bis zu t Fehler korrigiert, wenn man als Generatorpolynom $g(x) = \text{k.g.V.}\ (M_\alpha(x), M_{\alpha^2}(x), \ldots, M_{\alpha^{2t}}(x)) = \text{k.g.V.}(M_\alpha(x), M_{\alpha^3}(x), \ldots,$

$M_{\alpha^{2t-1}}(x)$) wählt. Für diesen Code ist dann $k = n - \operatorname{grad} g(x)$, wo $\operatorname{grad} g(x)$ den Grad von $g(x)$ bezeichnet. Setzen wir $t = \left[\frac{\delta-1}{2}\right]$ für ein $\delta \in \mathbf{N}$, so bedeutet dies bei der Wahl eines ungeraden δ, daß $\delta - 1 = 2t$ ist, und für ein gerades δ, daß $\delta - 1 = 2t - 1$ ist. Damit können wir das Generatorpolynom $g(x)$ stets in der Form $g(x) = \text{k.g.V.}(M_\alpha(x), M_{\alpha^2}(x), \ldots, M_{\alpha^{\delta-1}}(x))$ schreiben. Wegen $2t + 1 \leq d$ ist δ eine untere Schranke für die Minimaldistanz d.

Gibt man bei der Konstruktion eines zyklischen (n, k)-Polynomcodes über GF(2) der Länge $n = 2^m - 1$ an Stelle der Anzahl t der zu korrigierenden Fehler eine untere Schranke δ für d vor, so läßt sich das bisher Gesagte unmittelbar auf beliebige Galoisfelder GF(q), q eine Primzahlpotenz > 2, übertragen.

Gegeben sei ein $\delta \in \mathbf{N}$ mit $2 \leq \delta \leq n$ und $n = q^m - 1$. Wählt man für ein primitives Element α über GF(q) als Generatorpolynom $g(x) = \text{k.g.V.}(M_\alpha(x), M_{\alpha^2}(x), \ldots, M_{\alpha^{\delta-1}}(x))$, wo $M_{\alpha^i}(x)$ für $i = 1, 2, \ldots, \delta - 1$ das Minimalpolynom von α^i über GF(q) bezeichnet, so erhält man einen zyklischen (n, k)-Polynomcode über GF(q), für dessen Minimaldistanz d gilt $d \geq \delta$. Dieser Code kann $\delta - 1$ oder weniger Fehler entdecken und $\left[\frac{\delta-1}{2}\right]$ oder weniger Fehler korrigieren. Für seine Wortlänge k gilt $k = n - \operatorname{grad} g(x)$. Man kann zeigen, daß $k \geq n - m(\delta - 1)$ ist.

Eine weitere Verallgemeinerung ist wie folgt möglich: Sei n eine beliebige nichtnegative ganze Zahl, welche zu q teilerfremd ist. Dann kann man beweisen, daß es eine kleinste natürliche Zahl $m > 0$ gibt, so daß n die Zahl $q^m - 1$ teilt. Ferner kann man zeigen (siehe z.B.[5]): Die Nullstellen von $x^n - 1$ in GF(q^m) bilden eine Untergrupe der zyklischen Gruppe der Elemente $\neq 0$ von GF(q^m); diese Untergruppe ist daher ebenfalls zyklisch. Ist nun α ein beliebiges erzeugendes Element dieser Untergruppe und b eine beliebige nichtnegative ganze Zahl, dann wählen wir

$$g(x) = \text{k.g.V.}(M_{\alpha^b}(x), M_{\alpha^{b+1}}(x), \ldots, M_{\alpha^{b+\delta-2}}(x))$$

als Generatorpolynom. Der dadurch definierte (n, k)-Code ist ein zyklischer Polynomcode, für den gilt $d \geq \delta$ und $k = n - \operatorname{grad} g(x) \geq n - m(\delta - 1)$. Der Code wird nach seinen Entdeckern Bose, Chaudhuri und Hocquenghem ein *BCH-Code* der Länge n und der *designierten Distanz* δ genannt.

Wählt man $n = q - 1$, dann ist $m = 1$ und α ein primitives Element von GF(q). Das Minimalpolynom von α^{b+i} über GF(q) ist daher gegeben durch $M_{\alpha^{b+i}}(x) = x - \alpha^{b+i}$, und es folgt

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \ldots (x - \alpha^{b+\delta-2}).$$

Ferner ergibt sich: $k = n - \operatorname{grad} g(x) = n - \delta + 1$. Daher ist $n - k + 1 = \delta \leq d \leq n - k + 1$, was bedeutet, daß $\delta = d = n + k - 1$ ist. D.h., der Code ist optimal und seine designierte Distanz δ ist gleich seiner Minimaldistanz d. Es handelt sich dabei um den von uns bereits mehrfach angesprochenen *Reed-Solomon-Code*, den wir mit RS(n, d) bezeichnen. Der Code hängt natürlich auch noch von der Wahl des primitiven Elements α und von b ab; zumeist wird $b = 1$ gewählt. Die RS-Codes wurden unabhängig von den BCH-Codes gefunden und erst im nachhinein den BCH-Codes untergeordnet. Sie geben Anlaß zu einer Fülle von neuen optimalen Codes durch das im folgenden beschriebene Prinzip der *Verkürzung*, bei dem die Optimalität des Codes unberührt bleibt.

Sei C ein beliebiger zyklischer (n, k)-Code der Minimaldistanz d. Setzt man in jedem Nachrichtenwort und in jedem Codewort die ersten r Stellen gleich 0, so erhält man einen linearen $(n - r, k - r)$-Code, dessen Minimaldistanz $\geq d$ ist. Dieser Code ist wohl nicht mehr zyklisch, er kann aber technisch auf dieselbe Weise wie der zugehörige zyklische Code C implementiert werden. (Im Gegenteil: Es ergeben sich sogar gewisse Vereinfachungen.)

Die bei der CD verwendeten Codes entstehen beide durch Verkürzung von Reed-Solomon-Codes RS(255,5) über GF(256) und haben daher das Generatorpolynom $g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3)(x - \alpha^4)$. Üblicherweise nimmt man für α jenes primitive Element, dessen Minimalpolynom gegeben ist durch $M_\alpha(x) = 1 + x^2 + x^3 + x^4 + x^8$.

Literatur

[1] Dorninger D.: Codierung und Chiffrierung. Schriftenreihe zur Lehrerbildung im berufsbildenden Schulwesen, Heft 144. Päd. Inst. des Bundes in Wien, 1991

[2] Dorninger D. und Müller W.: Allgemeine Algebra und Anwendungen. Teubner, Stuttgart, 1984

[3] Carasso M.G., Peek J.B.H., Sinjou J.P.: The Compact Disc Digital Audio System. Philips tech. Rev. 40 (1982), 151–155

[4] Heemskerk J.P.J., Schouhamer Immink K.A.: Compact Disc: system aspects and modulation. Philips tech. Rev. 40 (1982), 157–164

[5] Lidl R., Niederreiter H.: Introduction to finite fields and their applications. Cambridge University Press, Cambridge, 1986

[6] Mac Williams F.J. and Sloane N.J.: The Theory of Error-Correcting Codes I, II. North Holland, Amsterdam, Oxford, 1977

[7] Peek J.B.H.: Some Features of the Compact Disc Digital Audio System. In: van der Burg A.H.P. and Matheij R.M.M. (Ed): Proceedings of the ICIAM 87, Contributions from the Netherlands

[8] Pohlmann K.C.: The Compact Disc Handbook (Second Edition). A-R Editions, Inc., Madison, 1992

Dietmar Dorninger
Inst. f. Algebra u. Diskrete Mathematik
Technische Universität Wien
Wiedner Hauptstr. 8–10/118
A-1040 Wien