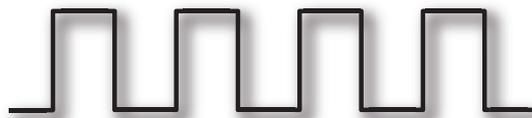


# Fachpraktikum Pulscodemodulation

Versuch KT 40



*Die theoretischen Fragen in Kapitel 6.1 müssen vor dem Praktikum gelöst werden. Die praktischen Aufgaben von den Kapiteln 6.2 werden während des Praktikums gelöst.*

---

Ausgabe: Herbst 2015

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Die Pulscodemodulationsverfahren im Überblick</b>	<b>2</b>
<b>3</b>	<b>Wichtige Systemaspekte der Pulscodemodulation</b>	<b>6</b>
3.1	Zeitliche Quantisierung . . . . .	6
3.2	Amplitudenquantisierung . . . . .	10
3.3	Codierung . . . . .	10
3.4	Kanalcodierung . . . . .	12
3.5	Signalregeneration und Zwischenverstärkung . . . . .	12
3.6	Demodulation und Fehlerkorrektur . . . . .	13
3.7	Decodierung . . . . .	13
3.8	Signalrekonstruktion . . . . .	14
3.9	Übertragung mehrerer Signale im Zeitmultiplex . . . . .	14
3.10	Synchronisationsaspekte . . . . .	15
<b>4</b>	<b>Bestimmung des Signal-Rauschleistungsverhältnisses in einem PCM-System</b>	<b>17</b>
4.1	Kanalrauschen und Übertragungsfehlerwahrscheinlichkeit . . . . .	17
4.2	Quantisierungsrauschen . . . . .	19
<b>5</b>	<b>Das PCM-Experimentalsystem</b>	<b>21</b>
5.1	Anschlüsse für die Stromversorgung . . . . .	21
5.2	Funktionsgeneratorblock . . . . .	22
5.3	Taktgeberschaltblock . . . . .	22
5.4	Synchronisationscode-Generator . . . . .	23
5.5	PCM-Sender . . . . .	23
5.6	Anschlüsse für die Stromversorgung . . . . .	26
5.7	Taktregenerationsschaltung . . . . .	26
5.8	Synchronisationscode-Detektor . . . . .	27
5.9	Demultiplexierlogik . . . . .	27
5.10	PCM-Empfänger . . . . .	28
<b>6</b>	<b>Aufgaben</b>	<b>30</b>
6.1	Theoretischer Teil (Vorbereitung) . . . . .	30
6.2	Praktischer Teil . . . . .	31

---

<b>A</b>	<b>Ausgewählte Systemaspekte</b>	<b>35</b>
A.1	Rahmensynchronisation im vorliegenden Versuchssystem . . . . .	35
A.2	Fehlerkorrigierende Block-Codes . . . . .	37
	<b>Bibliografie</b>	<b>42</b>

# Kapitel 1

## Einleitung

Die Aufgabe eines jeden Kommunikationssystems besteht in der Übermittlung informationstragender Signale von Nachrichtenquellen über einen Kanal zu den entsprechenden Senken. In den früheren *analogen* Systemen modulieren die zu übertragenden Signale unmittelbar die Parameter Amplitude, Phase oder Momentanfrequenz eines sinusförmigen Hilfssignals. Demgegenüber wandelt man ein informationstragendes Signal in einem *digitalen* System für die Übertragung in eine Folge diskreter Nachrichtensymbole um, was in jedem Fall zuerst eine Analog-Digital-Wandlung erfordert.

Der vorliegende Versuch befasst sich mit dem wichtigsten digitalen Pulsmodulationsverfahren, nämlich mit der *Pulscodemodulation* (PCM). Bei den hierbei erzeugten und übertragenen Signalen handelt es sich um codierte Pulsfolgen zur Darstellung von amplitudenquantisierten Abtastwerten eines analogen Signals. Die Pulsfolgefrequenz eines PCM-Signals ist dabei durch das Abtasttheorem und das zulässige Quantisierungsrauschen gegeben. Die Mehrfachausnützung eines Übertragungskanals im *Zeitmultiplex* bietet sich als einfachstes Multiplexverfahren für digitale Systeme an.

Der folgende Abschnitt enthält eine Übersicht der verschiedenen analogen und digitalen Pulsmodulationsverfahren und ordnet das in diesem Versuch betrachtete PCM-System in einen grösseren Rahmen ein. Anschliessend folgt eine kurze Diskussion der wesentlichen Systemaspekte, die in einem PCM-Übertragungssystem zu berücksichtigen sind. Ein weiterer Abschnitt befasst sich mit dem Signal-Geräuschleistungsverhältnis bei PCM-Übertragung. Anschliessend wird das eingesetzte PCM-Experimentalsystem beschrieben. In einem Anhang erfolgt die Diskussion einiger ausgewählter Systemaspekte, die für das tiefere Verständnis des vorliegenden Versuchsaufbaus von Bedeutung sind.

## Kapitel 2

# Die Pulscodemodulationsverfahren im Überblick

Während man *bei den analogen Modulationsverfahren* (Amplituden- und Phasen- bzw. Frequenzmodulation) *sinusförmige Hilfssignale* benutzt, bei denen jeweils ein oder mehrere Parameter (Amplitude, Phase bzw. Momentanfrequenz) proportional zum Nachrichtensignal variiert werden, verändert man *bei analogen Pulsmodulationssystemen* bestimmte Parameter einer *Pulsfolge*. Die *analogen Pulsmodulationsverfahren* umfassen Pulsamplitudenmodulation (PAM), Pulspositionsmodulation (PPM), Pulsfrequenzmodulation (PFM) und Pulsdauermodulation (PDM). Pulscodemodulation (PCM) und Deltamodulation ( $\Delta M$ ) werden den digitalen Verfahren zugeordnet.

Abbildung 2.1 enthält eine Gegenüberstellung der vier analogen Pulsmodulationsverfahren. Bei PAM werden im wesentlichen Pulse gleicher Form in konstantem Abstand  $T$ , gesendet, wobei deren Amplituden proportional zum jeweiligen Momentanwert des Nachrichtensignals  $s(t)$  sind. Bei PPM und PFM ist der Momentanwert des Signals jeweils in der Abweichung der zeitlichen Lage des Impulses bzw. der Impulsfolgefrequenz von einem Mittelwert enthalten. Diese beiden Verfahren stehen in direkter Analogie zur Phasen- (PM) bzw. Frequenzmodulation (FM); die PDM hingegen ist ohne direkte Parallele bei den analogen Modulationsverfahren. Im letztgenannten Fall wird bei der Modulation die Pulsbreite um einen Mittelwert variiert, wobei je nach gewählter Variante die zeitliche Lage der Vorderflanke, der Rückflanke oder beider Flanken gleichzeitig verändert wird. Bei den eben vorgestellten Verfahren können innerhalb des linearen Teils der Modulationskennlinie alle Amplitudenwerte des zu übertragenden Nachrichtensignals  $s(t)$  dargestellt werden, weshalb man sie auch als *analoge Pulsmodulationsverfahren* bezeichnet. Bei PCM werden hingegen die Amplitudenwerte der Nachrichtenfunktion zunächst *quantisiert*, also auf den jeweils nächstliegenden aus einer endlichen Auswahl möglicher Werte gerundet. Anschliessend erfolgt deren (zumeist binäre) Codierung und Darstellung als Impulsfolge. Aufgrund der beschränkten Anzahl möglicher Binärstellen pro Codewert - was endlich vielen Amplitudenstufen entspricht - kann das ursprüngliche Signal *nur näherungsweise reproduziert werden*. Die Abweichung des nach der Demodulation wiedergewonnenen Signalverlaufs von der ursprünglichen Signalform macht sich als *Quantisierungsrauschen* bemerkbar. Es ist jedoch darauf hinzuweisen, dass die Übertragungsqualität bei genügend tiefer Bitfehlerwahrscheinlichkeit lediglich vom Quantisierungsrauschen beeinträchtigt wird, also nur vom Abstand oder von der Anzahl der Quantisierungsniveaus abhängt; im Gegensatz zum Empfängerrauschen, das bei den analogen Systemen häufig als Hauptursache für den Qualitätsverlust auftritt, kann jedoch der Pegel des Quantisierungsrauschens beim Entwurf eines digitalen Systems nach Massgabe der zur Verfügung stehenden Übertragungsbandbreite gewählt werden. Abbildung 2.2 veranschaulicht die drei Schritte der zeitlichen Quantisierung, der Amplitudenquan-

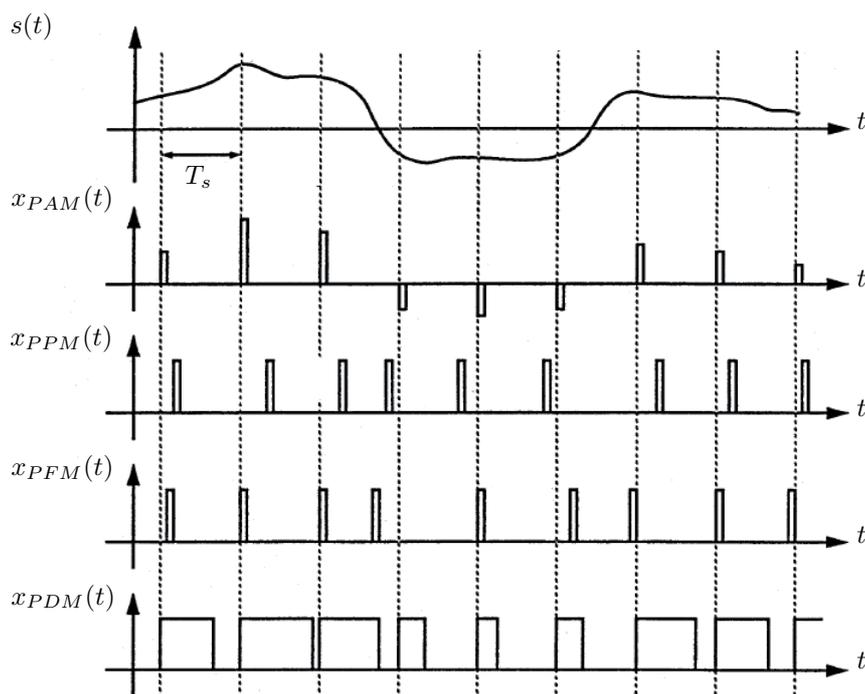


Abbildung 2.1: Übersicht der analogen Pulsmodulationsverfahren (Pulsamplitudenmodulation, Pulspositionsmodulation, Pulsfrequenzmodulation, Pulsdauermodulation).

tisierung sowie der Codierung bei der PCM, auf die in den kommenden Abschnitten noch näher eingegangen wird.

Auch bei der Deltamodulation ( $\Delta M$ ) wird eine Amplitudenquantisierung vorgenommen; man versucht hier, die Nachrichtenfunktion durch eine Treppenkurve mit vorgegebener Amplitudenstufung zu approximieren, wobei im Falle der adaptiven  $\Delta M$  die Grösse der Stufen dem Funktionsverlauf angepasst werden kann. Das Prinzip der Deltamodulation kann Abbildung 2.3 entnommen werden: Übersteigt der Momentanwert der Nachrichtenfunktion  $s(t)$  zu einem bestimmten Abtastzeitpunkt den Pegel der Treppenkurve, so wird letztere um eine weitere Amplitudenstufe erhöht; unterschreitet er ihn, so wird die Treppenkurve um einen Quantisierungsschritt erniedrigt. Um dies dem Empfänger mitzuteilen, wird im ersten Fall eine logische Eins (Senden eines Pulses), im letzteren eine logische Null (im vorliegenden Beispiel kein Puls) übermittelt.

Bei den Pulsmodulationsverfahren kann man ferner die Pausen zwischen den einzelnen, mit einem Nachrichtensignal modulierten Pulsen mit weiteren informationstragenden Pulsfolgen belegen und auf diese Weise auf dem gleichen Übertragungskanal *mehrere Nachrichtensignale im Zeitmultiplex* übertragen. Dabei kann die Übertragung pulsmodulierter Signale entweder im Basisband oder trägerfrequent erfolgen. In beiden Fällen müssen jedoch bestimmte Anforderungen an die Charakteristik des Übertragungskanals gestellt werden, um bei Beschränkungen der Bandbreite die Signalverzerrungen und das Übersprechen klein zu halten.

Der nächste Abschnitt geht näher auf einzelne Aspekte des PCM Verfahrens ein. Die dabei auftretenden Signalverarbeitungsschritte werden in der Reihenfolge ihrer Anwendung diskutiert.

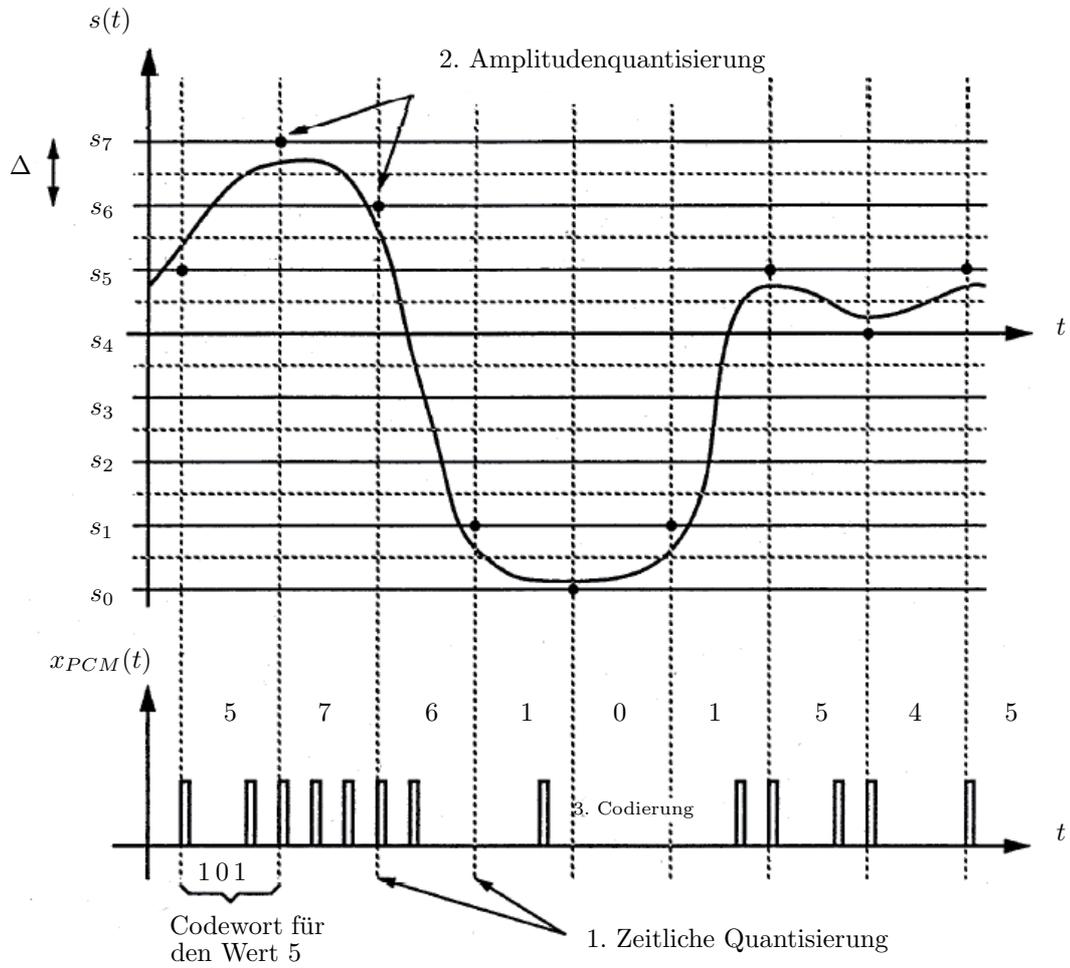


Abbildung 2.2: Pulscode modulation (PCM) als digitales Pulsmodulationsverfahren.

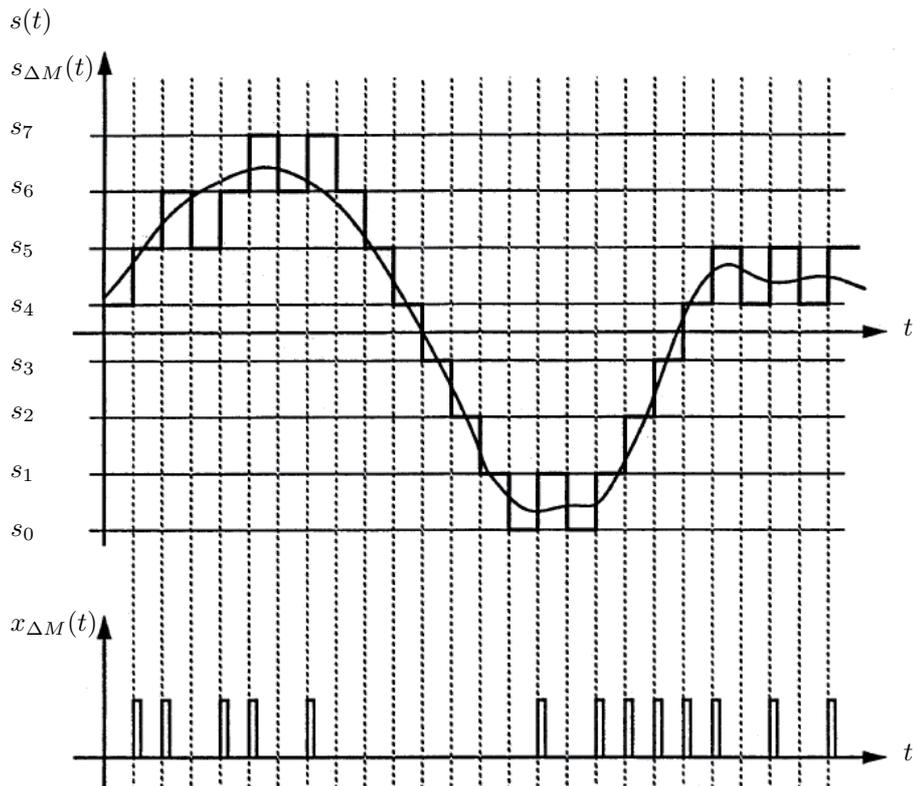


Abbildung 2.3: Deltamodulation ( $\Delta M$ ) als digitales Pulsmodulationsverfahren.

## Kapitel 3

# Wichtige Systemaspekte der Pulscodemodulation

PCM systeme weisen mehrere Signalverarbeitungsschritte auf. Zu den wesentlichen Operationen im Sender zählen die *zeitliche Quantisierung* bzw. *Abtastung* des analogen Nachrichtensignals, die *Amplitudenquantisierung*, die *Codierung* der diskreten Amplitudenwerte in eine Folge von Kanalsymbolen und gegebenenfalls der *Einsatz fehlerkorrigierender Codes*. Auf der Übertragungsstrecke erfolgt häufig *eine Regeneration und Zwischenverstärkung* der Signale. Im PCM-Empfänger werden diese nach ihrer *Demodulation einer Fehlerkorrektur* unterzogen. Aus den zurückgewonnenen Pulsfolgen gewinnt man die diskreten Amplitudenwerte zurück, eine Operation, die man als *Decodierung* bezeichnen kann. Die *Signalrekonstruktion* erfolgt in der Regel durch die Tiefpassfilterung der nun diskret amplitudenmodulierten Pulsfolge (PAM). Die *Übertragung mehrerer Signale im Zeitmultiplex* erfordert dabei zusätzliche Systemfunktionen; unter anderem tritt eine Reihe von Synchronisationsproblemen auf, die mehrere Massnahmen erforderlich machen.

Es ist zu betonen, dass die PCM kein eigentliches Modulationsverfahren im engeren Sinne darstellt, weil man unter dem Begriff *Modulation* ursprünglich die Veränderung eines oder mehrerer Parameter eines sinus- oder pulsförmigen Hilfssignals versteht. Der einzige Schritt, der dieser Definition voll entspricht, ist die zeitliche Quantisierung und Abtastung.

Im folgenden werden die oben erwähnten Signalverarbeitungsschritte in der Reihenfolge ihres Auftretens erläutert. Ausführlichere Darstellungen des PCM-Verfahrens können [1, 2] entnommen werden.

### 3.1 Zeitliche Quantisierung

Ein analoges Nachrichtensignal  $s(t)$  wird mit einer periodischen Folge  $f(t)$  sehr kurzer Impulse im Abstand  $T_s$  multipliziert, und die so erhaltenen Abtastwerte mittels eines Halteglieds während der Zeit  $T_h \leq T_s$  auf einem konstanten Wert festgehalten. Abbildung 3.1 veranschaulicht diesen Vorgang. Um beim Empfänger eine korrekte Signalrekonstruktion gewährleisten zu können, darf das Abtasttheorem nicht verletzt werden; bekanntlich muss die Abtastkreisfrequenz  $\omega_s = 2\pi/T_s$  mindestens doppelt so gross wie die höchste im Signal  $s(t)$  vorkommende Kreisfrequenz  $\omega_{\max}$  sein:  $\omega_s \geq 2\omega_{\max}$ . Die Einhaltung des Abtasttheorems wird in der Praxis sichergestellt, indem das Nachrichtensignal vor seiner Abtastung durch ein Tiefpassfilter der Grenzkreisfrequenz  $\omega_g \leq \omega_s/2$  bandbegrenzt wird (*anti-alias filtering*). Die in der Telephonie verwendeten Werte betragen  $\omega_{\max} = 2\pi \cdot 3400s^{-1}$  und  $\omega_s = 2\pi \cdot 8000s^{-1}$ .

Der beschriebene Abtastvorgang ermöglicht somit die vollständige Darstellung des Nachrichten-

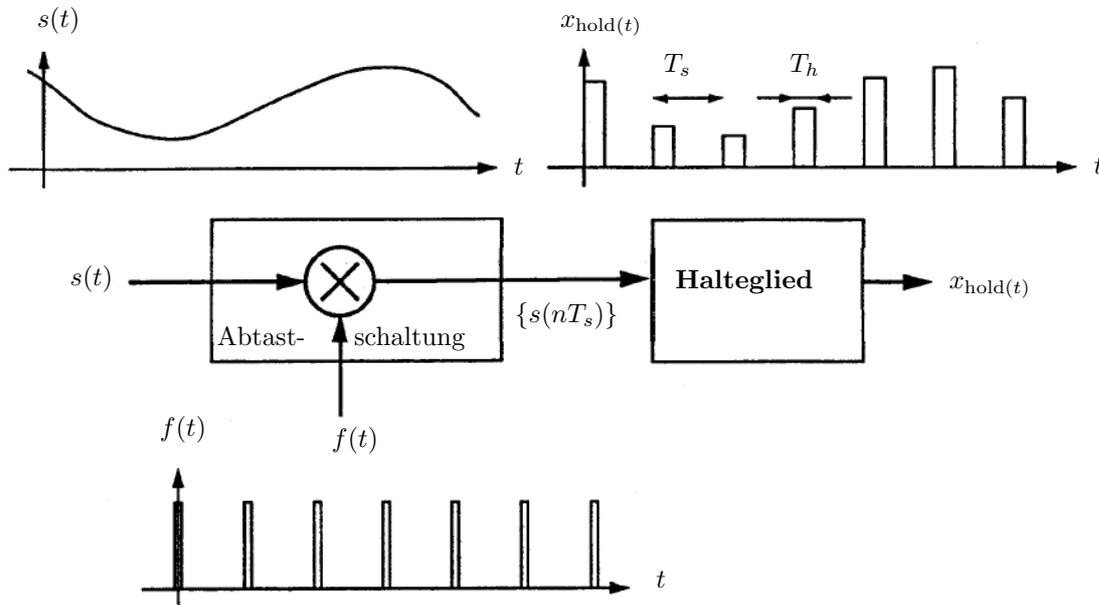


Abbildung 3.1: Blockschema des Abtastvorgangs mit Halteglied. Die Abtastung kann als Multiplikation des Nachrichtensignals  $s(t)$  mit einer Schaltfunktion  $f(t)$  und anschließender Halteoperation betrachtet werden; die Funktion  $f(t)$  ist im dargestellten Fall eine periodische Rechteckimpulsfolge.

signals  $s(t)$  mittels äquidistanter Stützwerte  $s(nT_s)$  an den Stellen  $t = nT_s$ , ( $n = 0, \pm 1, \pm 2, \dots$ ). Der Verlauf eines *abgetasteten und gehaltenen Signals*  $x_{\text{hold}}(t)$  (*sample and hold signal*) gemäss

$$x_{\text{hold}} = \sum_{n=-\infty}^{\infty} s(nT_s) q(t - nT_s) \quad (3.1)$$

mit Hilfe des nicht zentrierten Rechteckimpulses

$$q(t) = \begin{cases} A & 0 < t < T_h \\ 0 & t < 0 \text{ und } t > T_h \end{cases} \quad \circ \text{---} \bullet Q(\omega) = A \cdot T_h \frac{\sin\left(\frac{T_h}{2}\omega\right)}{\frac{T_h}{2}\omega} e^{-jT_h\omega} \quad (3.2)$$

geht aus Abbildung 3.2 (a) hervor; das entsprechende Spektrum  $X_{\text{hold}}(\omega)$  ist aus Abbildung 3.2 (b) ersichtlich.

Sobald eine dem Abtastwert  $s(nT_s)$  des Nachrichtensignals proportionale Spannung am Ausgang der Abtastung in Abbildung 3.1 vorliegt, wird diese mittels eines Halteglieds während der Zeitdauer  $T_h$  beibehalten. Anschliessend fällt die Spannung bis zur Verfügbarkeit des nächsten Stützwertes auf Null zurück.

Um das Spektrum  $X_{\text{hold}}(\omega)$  zu berechnen, ist (3.1) zunächst wie folgt umzuformen:

$$\begin{aligned} x_{\text{hold}}(t) &= \sum_{n=-\infty}^{\infty} s(nT_s)q(t - nT_s) = q(t) * \sum_{n=-\infty}^{\infty} s(nT_s)\delta(t - nT_s) \\ &= q(t) * \left( s(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \right) \end{aligned} \quad (3.3)$$

Dabei stellt  $\delta(t)$  die Dirac-Distribution dar. Mit Hilfe des Fourierpaars

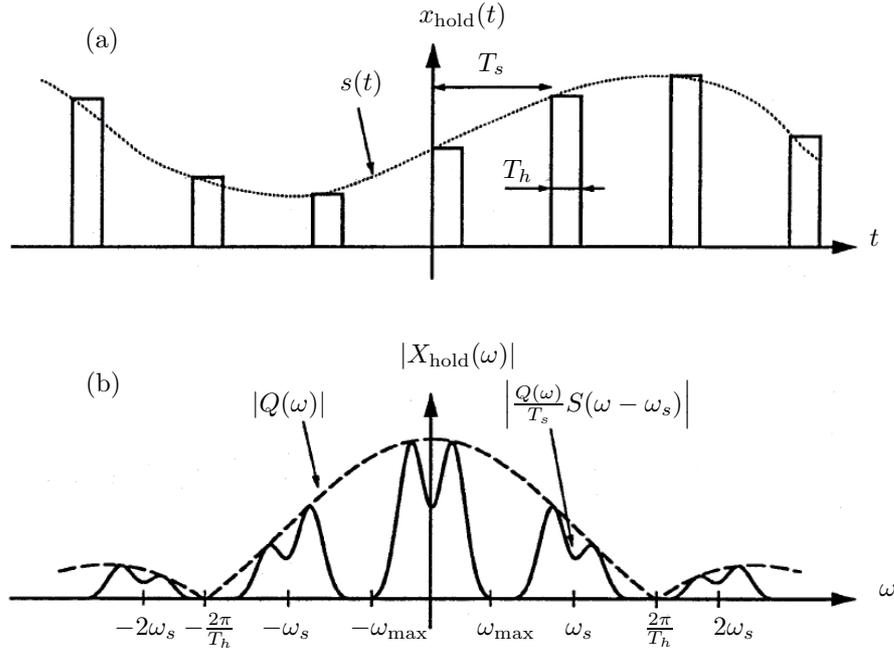


Abbildung 3.2: Signalabtastung mit Halteglied. (a) Abgetastetes und gehaltenes Signal  $x_{\text{hold}}(t)$ . (b) Amplitudengang  $|X_{\text{hold}}(\omega)|$  des abgetasteten und gehaltenen Signals  $x_{\text{hold}}(t)$ .

$$\sum_{n=-\infty}^{\infty} \delta(t - nT_s) \longleftrightarrow \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s) \quad (3.4)$$

folgt somit im Frequenzbereich

$$X_{\text{hold}}(\omega) = Q(\omega) \left( \frac{1}{2\pi} S(\omega) * \frac{2\pi}{T_s} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_s) \right) = \frac{Q(\omega)}{T_s} \sum_{k=-\infty}^{\infty} S(\omega - k\omega_s). \quad (3.5)$$

Die einzelnen Teilspektren  $S(\omega - k\omega_s)$  in Gleichung (3.5) erfahren demzufolge eine Verzerrung entsprechend dem Verlauf von  $Q(\omega)$  (siehe Abbildung 3.2 (b)).

Das Resultat gemäss (3.5) ist vor allem für die Signalrekonstruktion von Interesse und wird deshalb in Abschnitt 3.8 weiter diskutiert. Es sei hier lediglich darauf hingewiesen, dass das Signal  $x_{\text{hold}}(t)$  in (3.1) und dessen Spektrum  $X_{\text{hold}}(\omega)$  in (3.5) nicht mit dem für die Übertragung verwendeten PCM-Signal  $x_{\text{PCM}}(t)$  zu verwechseln ist. Abbildung 3.3 (a) enthält einen Ausschnitt des abgetasteten

und gehaltenen Signals; die diskreten Amplituden der Pulse, die mit  $s_0$  bis  $s_7$  bezeichnet sind, lassen sich durch dreistellige binäre Zahlen "000" bis "111" ausdrücken. Diese Binärdarstellung der Amplitudenniveaus wird nun als PCM-Signal  $x_{\text{PCM}}(t)$  übermittelt, wobei eine logische Eins mit einem Puls positiver Amplitude  $A$ , eine logische Null mit einem Puls negativer Polarität ( $-A$ ) dargestellt wird (siehe Abbildung 3.3 (b)). Alle Pulse des PCM-Signals haben im vorliegenden Beispiel die Dauer  $T = T_s/3$ .

Da das informationstragende Signal  $x_{\text{PCM}}(t)$  einen stochastischen Prozess darstellt, können hierfür

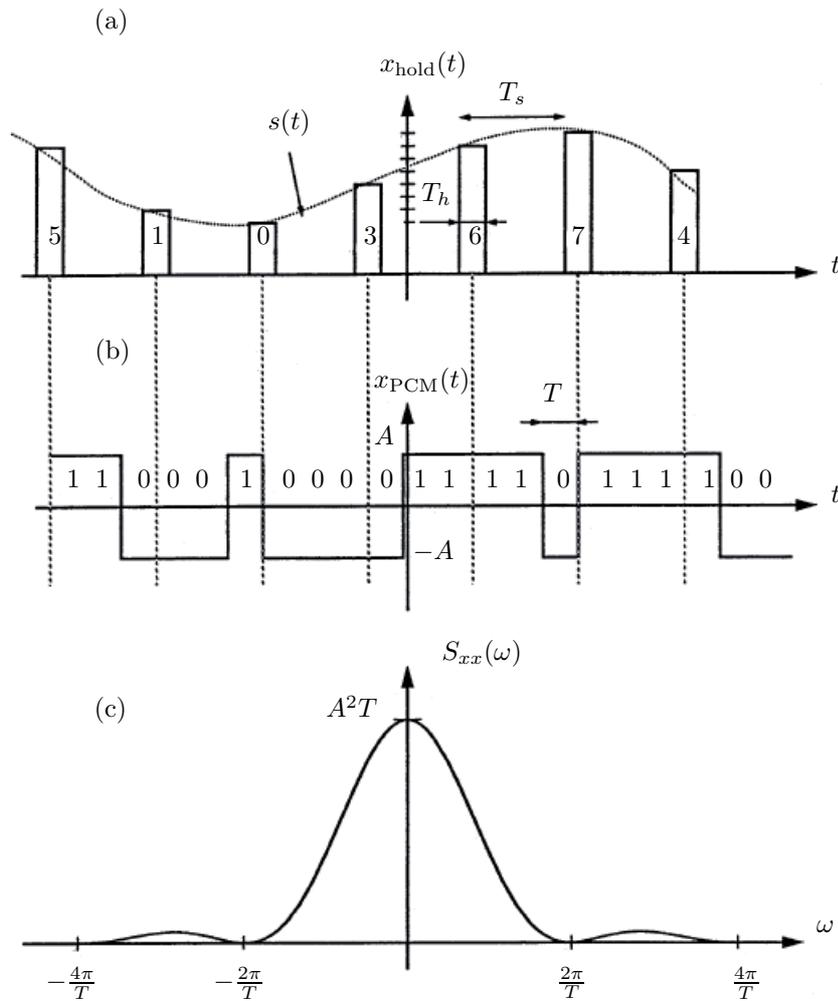


Abbildung 3.3: Veranschaulichung der Pulsmodulation. (a) Abgetastetes und gehaltenes Signal  $x_{\text{hold}}(t)$ ; die Pulsamplituden werden auf das nächstliegende Amplitudenniveau gerundet, wobei die diskreten Amplituden mit  $s_0, \dots, s_7$  bezeichnet werden. (b) Pulsmoduliertes Signal  $x_{\text{PCM}}(t)$  im Zeitbereich. (c) Leistungsdichtespektrum  $S_{xx}(\omega)$  des PCM-Signals  $x_{\text{PCM}}(t)$ .

nur gewisse globale Größen wie beispielsweise die Autokorrelationsfunktion

$$R_{xx}(\tau) = E\{x_{\text{PCM}}(t) x_{\text{PCM}}(t + \tau)\}$$

gegeben werden. Dabei bezeichnet  $E\{x\}$  den Erwartungswert des Arguments. Nach kurzer Rechnung erhält man unter Voraussetzung einer Gleichverteilung von logischen Nullen und Einsen

$$R_{xx}(\tau) = \begin{cases} A^2 \left(1 - \frac{|\tau|}{T}\right) & |\tau| < T \\ 0 & |\tau| > T \end{cases}. \quad (3.6)$$

Die Anwendung des Wiener-Khintchine-Theorems liefert das Leistungsdichtespektrum  $S_{xx}(\omega)$  als Fourier-Transformierte der Autokorrelationsfunktion  $R_{xx}(\tau)$ :

$$S_{xx}(\omega) = A^2 T \left( \frac{\sin\left(\frac{T}{2}\omega\right)}{\frac{T}{2}\omega} \right)^2. \quad (3.7)$$

Der graphische Verlauf des Leistungsdichtespektrums ist in Abbildung 3.3 (c) skizziert. Es ist deutlich zu erkennen, dass die Lage der ersten Nullstelle bei  $\omega = 2\pi/T$  als gutes Mass für die Bandbreite des PCM-Signals dienen kann.

## 3.2 Amplitudenquantisierung

Eine analoge Nachrichtenfunktion  $s(t)$  (beispielsweise ein Sprachsignal) ist eine reellwertige Funktion einer reellen Variablen  $t$ ; auch bei einer auf ein endliches Intervall begrenzten Amplitude können die Abtastpunkte  $s(nT_s)$  ( $n = 0, \pm 1, \pm 2, \dots$ ) unendlich viele mögliche Werte annehmen. Die menschlichen Sinne verfügen jedoch nur über eine begrenzte Auflösungsfähigkeit. Dies bedeutet, dass die ursprüngliche analoge Nachrichtenfunktion  $s(t)$  durch ein Signal  $s_q(t)$  mit  $N$  möglichen Amplitudenstufen hinreichend gut approximiert werden kann. Die *Beschränkung auf eine endliche Anzahl von Amplitudenwerten* ist eine grundlegende Bedingung für den Einsatz der PCM.

Die Umwandlung eines analogen, wertekontinuierlichen Amplitudenwerts  $s(nT_s)$  in einen digitalen Abtastwert, der eine von  $N$  möglichen Stufen annehmen kann, wird als *Amplitudenquantisierung* bezeichnet. In Abbildung 2.2 sind insgesamt  $N = 8$  mögliche Quantisierungsstufen  $s_0, \dots, s_7$ , vorgesehen. Die Amplitudenquantisierung hat dabei zwei Auswirkungen:

1. Der gesamte Amplitudenbereich wird in  $N - 1$  Subintervalle der Höhe  $\Delta$  unterteilt. Die diskreten Amplitudenstufen entsprechen den Werten  $(-N/2)\Delta, \dots, 0, \dots, (N/2 - 1)\Delta$  und sind in Abbildung 2.2 durch die ausgezogenen horizontalen Linien angedeutet.
2. Es entstehen  $N - 1$  Entscheidungsschwellen zwischen den  $N$  möglichen Amplitudenwerten, und zwar bei  $\pm\Delta/2, \pm 3\Delta/2, \pm 5\Delta/2, \dots$ . Die gestrichelten horizontalen Linien in Abbildung 2.2 zeigen die Lage dieser Schwellwerte.

Da der Abstand zwischen zwei nebeneinanderliegenden Amplitudenstufen stets  $\Delta$  beträgt, spricht man von einer *linearen Quantisierung*. Die sogenannte *Transfer-Charakteristik*, die in Abbildung 3.4 (a) dargestellt ist, gibt den Ausgangswert  $s_q$  des Quantisierers in Funktion der Eingangsgrösse  $s$  wieder. Dabei ergibt sich der in Abbildung 3.4 (b) angedeutete Quantisierungsfehler  $e_q = s - s_q$ .

## 3.3 Codierung

Nach Durchführung der Zeit- und Amplitudenquantisierung liegen die Abtastwerte der Nachrichtenfunktion in diskreter Form vor; in Abbildung 2.2 entspricht dies der Zahlenfolge  $\{5, 7, 6, 1, 0, 1, 5, 4, 5\}$ . Diese Art der Signaldarstellung ist jedoch für die Übermittlung über ein Kabel, einen Funkkanal

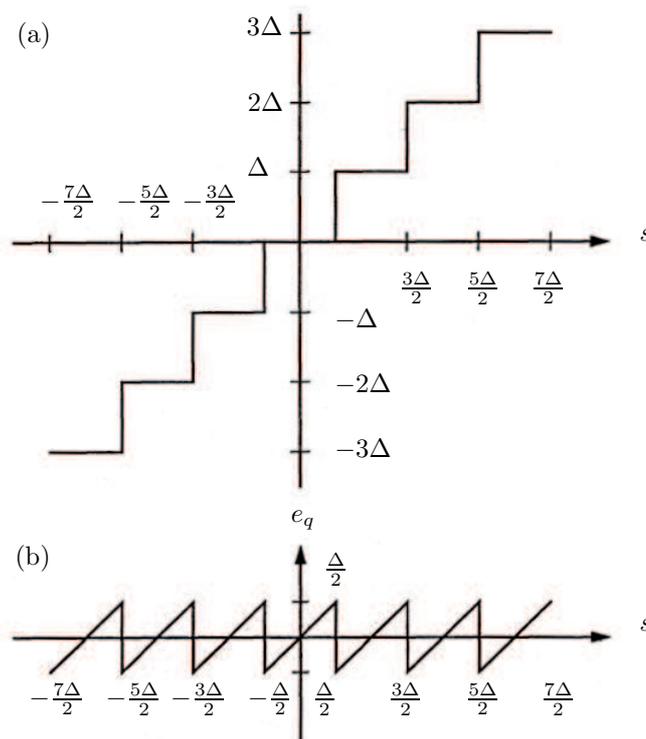


Abbildung 3.4: Amplitudenquantisierung: (a) Transferkennlinie des linearen Quantisierers. (b) Quantisierungsfehler  $e_q = s - s_q$  in Abhängigkeit der Eingangsgröße  $s$ .

oder eine Glasfaser nicht geeignet. Als *Codierung* bezeichnen wir die Abbildung der diskreten Wertefolge in ein für die Übertragung geeignetes Format; die Darstellung eines bestimmten diskreten Amplitudenwertes in einem derartigen Format wird als *Codewort* bezeichnet. Die Menge aller Codeworte bildet den *Code*.

Das in der Praxis weitaus am häufigsten anzutreffende Format ist die *Binärdarstellung*. In der konventionellen Form wird jeder der  $N$  diskreten Werte  $s_0, \dots, s_{N-1}$  einfach als  $k$ -stellige binäre Zahl ausgedrückt; die einzelnen binären Symbole lassen sich anschreiben als  $b_{k-1}b_{k-2} \dots b_1b_0$ . Dabei ist es notwendig, die Codewortlänge  $k$  so zu wählen, dass genügend Codeworte zur Darstellung aller  $N$  Werte vorhanden sind: die Ungleichung  $2^k \geq N$  muss also erfüllt sein.

Die Länge  $k$  der Codeworte beim vorliegenden PCM-Versuchssystem beträgt - je nach Einsatz von Massnahmen zur Fehlerkorrektur - 7,6 oder gar nur 4 *Bits*, was 128,64 bzw. 16 Quantisierungsniveaus ergibt ( $N = 2^k$ ). An dieser Stelle sei hervorgehoben, dass fehlerkorrigierende Codes nicht mit der in diesem Abschnitt beschriebenen Codierung der diskreten Amplitudenwerte zu verwechseln sind. Die Möglichkeit des Einsatzes fehlerkorrigierender Codes ist ein herausragendes Merkmal digitaler Übermittlungssysteme und wird im nächsten Abschnitt kurz erläutert.

### 3.4 Kanalcodierung

Beim Entwurf eines digitalen Kommunikationssystems geht es darum, digitale Daten mit einer geforderten minimalen Datenrate  $R_b$  (in Bit/s) über einen gegebenen Kanal zu übermitteln, wobei eine bestimmte Bitfehlerwahrscheinlichkeit  $P_b$  nicht überschritten werden darf. Die gewählte *Übertragungsbandbreite* sowie die *Signalleistung* sind die beiden wichtigsten Systemparameter und beeinflussen massgeblich das erzielbare Signal-Rauschleistungsverhältnis  $S/N$ , welches sich direkt auf die mittlere *Fehlerwahrscheinlichkeit*  $P_b$  auswirkt.

Bei der *Modulation* handelt es sich um eine signalverarbeitende Operation, mit der eine möglichst gute Anpassung an die Charakteristika eines gegebenen Übertragungsmediums angestrebt wird. Die *Kanalcodierung* verfolgt hingegen das Ziel, die *Zuverlässigkeit* der Übertragung zu verbessern. Der Einsatz fehlerkorrigierender Codes ist nur in digitalen Systemen durchführbar und verleiht daher digitalen Übertragungsverfahren eine *viel höhere Flexibilität* im Hinblick auf die mögliche Wahl der Übertragungsbandbreite und der Signalleistung.

Beim Einsatz der Kanalcodierung können wahlweise zwei Ziele angestrebt werden: zum einen kann man sich mit der blossen *Fehlererkennung* begnügen, worauf der Sender die als fehlerhaft erkannten Daten nochmals übermitteln muss; zum anderen ist auch der Einsatz von Codes mit der Fähigkeit zur *Fehlerkorrektur* möglich, die dem Empfänger erlauben, vereinzelt oder büschelweise auftretende Bitfehler bis zu einem gewissen Grad zu korrigieren und die ursprünglich gesendete Information aus den fehlerhaft empfangenen Daten zu rekonstruieren. Beide Varianten sind nur durch den gezielten Einsatz von Redundanz realisierbar.

In der Praxis werden sehr häufig sogenannte *Block-Codes* eingesetzt. Dabei geht man davon aus, dass die zu übermittelnde Information bereits in digitaler Form vorliegt (siehe Abschnitt 3.3). Die Kanalcodierung geschieht nun dadurch, dass Blocks von jeweils  $k$  Datenbits  $b_{k-1}b_{k-2} \dots b_1b_0$  auf Blocks von jeweils  $n$  Codebits  $c_{n-1}c_{n-2} \dots c_1c_0$  wobei  $n > k$  abgebildet werden. Jedem einzelnen der  $N = 2^k$  diskreten Nachrichtensymbole  $s_0, \dots, s_{N-1}$ , gemäss Abschnitt 3.3, die durch jeweils  $k$  Datenbits repräsentiert werden, wird also eines von  $2^n$  möglichen Kanal-Codeworten zugeordnet. Nach der Übertragung mögen zwar einige der jeweils  $n$  Codebits im Empfänger falsch detektiert werden; da es jedoch  $2^{n-k}$  Mal so viele Codeworte im Block-Code gibt, als zur ungeschützten Übermittlung der Information eigentlich notwendig wäre, kann man eine empfangene, *unerlaubte* Sequenz von Codebits als das Codewort mit dem kleinsten Abstand davon interpretieren. Die Aufgabe des Decoders für den fehlerkorrigierenden Code besteht nun darin, aus einer (gegebenenfalls unerlaubten) Codebitsequenz  $c_{n-1}c_{n-2} \dots c_1c_0$  die wahrscheinlichste Sequenz von Datenbits  $b_{k-1}b_{k-2} \dots b_1b_0$  zu ermitteln.

Die verbesserte Übertragungssicherheit wird allerdings mit einem höheren Aufwand (Einsatz von Coder- und Decoder-Bausteinen) sowie einer grösseren Übertragungsbandbreite erkauft. Da in der gleichen Zeit  $n$  anstelle von lediglich  $k$  Bits über den Kanal gesendet werden müssen, erhöht sich die erforderliche Bandbreite um einen Faktor  $n/k$  ("Code Rate") gegenüber jener des uncodierten Verfahrens. Durch sinnvollen Einsatz mächtiger fehlerkorrigierender Codes lassen sich jedoch Gewinne erzielen, die mit analogen Modulationsverfahren nicht realisierbar sind.

### 3.5 Signalregeneration und Zwischenverstärkung

Ein wesentlicher Vorteil digitaler Übertragungsverfahren bildet die Möglichkeit, die durch Einwirkung von Störungen auf dem Kanal verzerrten Impulse wieder zu regenerieren. Dies wird vor allem im Fernmeldenetz praktiziert, wo sogenannte *Repeater* zur Verstärkung und anschliessenden Regeneration des PCM-Signals  $x_{\text{PCM}}(t)$  eingesetzt werden.

Drei wesentliche Operationen werden bei der Regeneration digitaler Signale durchgeführt:

1. *Egalisation*: Nach der Verstärkung müssen die linearen Verzerrungen aufgrund des Amplituden- und des Phasengangs der Übertragungsstrecke ausgeglichen werden.
2. *Bit-Synchronisation*: Aufgrund der empfangenen und entzerrten Pulse liefert eine Zeitgeberschaltung das Clock-Signal, welches für die korrekte Funktionsweise des Detektors benötigt wird.
3. *Detektion und Regeneration*: Zunächst detektiert der Empfänger aus dem entzerrten PCM-Signal die Werte der Binärsymbole. Diese werden in der Folge verwendet, um ein entsprechend regeneriertes PCM-Signal erneut auf den Kanal zu schicken.

Ausser einer tolerierbaren zeitlichen Verzögerung entspricht das regenerierte Signal bei Abwesenheit grösserer Störanteile vollkommen dem ursprünglichen PCM-Signal. Aus zwei Gründen können sich jedoch Abweichungen ergeben:

1. Rauscheinflüsse auf dem Kanal oder in der Empfangsstufe des Verstärkers können bei der Demodulation zu *Fehldetektionen* und somit zu *Bitfehlern* führen. Fehlerkorrigierende Massnahmen müssen ergriffen werden, um die notwendige Abhilfe zu schaffen.
2. Falls der Abstand  $T$  zweier Pulse auf dem Übertragungskanal geringfügig vom vorgesehenen Wert abweicht, können die den Pulswechseln entsprechenden Zeitpunkte von der Synchronisationsschaltung nur mit einer begrenzten Genauigkeit ermittelt werden. Es resultieren Verzerrungen, deren Ursache als *timing jitter* bezeichnet wird.

## 3.6 Demodulation und Fehlerkorrektur

Die erste signalverarbeitende Operation im PCM-Empfänger entspricht der im Abschnitt 3.5 geschilderten Signalregeneration: das verzerrte Signal wird verstärkt und einem Kanalegalisator zugeführt. Anschliessend versucht man, die korrekten Codeblocks  $c_{n-1}c_{n-2} \dots c_1c_0$  aus dem entzerrten PCM-Signal  $x_{\text{PCM}}(t)$  wiederzugewinnen. Wie schon in Abschnitt 3.4 erläutert wurde, besteht die Aufgabe des Decoders für den fehlerkorrigierenden Code in der Ermittlung der wahrscheinlichsten Sequenz von Datenbits  $b_{k-1}b_{k-2} \dots b_1b_0$  aus dem möglicherweise unerlaubten, empfangenen Block  $c_{n-1}c_{n-2} \dots c_1c_0$ . Die Lösung dieser Aufgabe hängt sehr stark von der Struktur des eingesetzten fehlerkorrigierenden Codes ab. Für die Beschreibung des im vorliegenden PCM-Experimentalsystem verwendeten Block-Codes und des prinzipiellen Vorgehens bei der Decodierung sei auf Abschnitt A.2 verwiesen.

## 3.7 Decodierung

Nach der Wiederherstellung der Datenblocks  $b_{k-1}b_{k-2} \dots b_1b_0$  aus dem codierten PCM-Signal folgt deren Abbildung auf die entsprechenden diskreten Amplituden  $s_0, \dots, s_{N-1}$ . Diese Operation entspricht dem inversen Vorgehen des im Abschnitt 3.3 beschriebenen Codierungsvorgangs. Es ist jedoch zu beachten, dass der im Abschnitt 3.2 behandelte Prozess der Amplitudenquantisierung irreversibel ist; die ursprünglichen Stützwerte  $s(nT_s)$ , werden durch die diskreten Amplituden  $s_0, \dots, s_{N-1}$  unter Inkaufnahme eines maximalen Fehlers von  $\pm\Delta/2$  lediglich angenähert (siehe Abbildung 3.4 (b)), wobei die Leistung des sogenannten Quantisierungsrauschens durch den quadratischen Mittelwert  $E\{e_q^2\}$  des Quantisierungsfehlers  $e_q$  gegeben ist. Eine ausführlichere Diskussion des Quantisierungsrauschens in einem PCM-System erfolgt in Abschnitt 4.2.

### 3.8 Signalrekonstruktion

Um die ursprüngliche analoge Nachrichtenfunktion  $s(t)$  zu rekonstruieren, wird zunächst ein PAM Signal gemäss (3.1) und (3.2) gebildet. Die Amplituden der einzelnen Pulse im Abstand  $T_s$  werden proportional zu den diskreten Abtastwerten gewählt, die aufgrund der Decodierung jeweils eine der reellen Zahlen  $s_0, \dots, s_{N-1}$  annehmen können. Ein möglicher Verlauf für das sich ergebende PAM-Signal und dessen Spektrum sind in Abbildung 3.2 (a) bzw. in Abbildung 3.2 (b) dargestellt; Abbildung 3.2 (b) ist zu entnehmen, dass die ursprüngliche Nachrichtenfunktion  $s(t)$  durch Tiefpassfilterung des PAM-Signals  $x_{\text{hold}}(t)$  wiedergewonnen werden kann, falls der verzerrende Einfluss des Pulsspektrums  $Q(\omega)$  in (3.5) nicht zu gross ist.

Für den Grenzwert  $\omega_{\text{max}} = \omega_s/2$  ergibt

$$\left| \frac{Q(\omega_{\text{max}})}{Q(0)} \right| = \frac{\sin\left(\frac{\pi T_h}{2T_s}\right)}{\frac{\pi T_h}{2T_s}} \quad (3.8)$$

Werte grösser als 0.9, falls das Tastverhältnis  $T_s/T_h$  kleiner als 0.5 eingestellt wird. Bei dieser Wahl lässt sich also die relative Verzerrung  $(Q(0) - Q(\omega_{\text{max}}))/Q(0)$  an der Bandgrenze auf weniger als 10% beschränken. Wählt man nun eine etwas höhere Abtastkreisfrequenz bei gleichbleibendem Verhältnis  $T_h/T_s$ , so verringern sich die Verzerrungen, wie man anhand der Abbildung 3.2 (b) nach kurzer Überlegung feststellen kann. Dies bedeutet, dass in jedem Fall auf den *Einsatz eines Tiefpassfilters mit Entzerrercharakteristik verzichtet* werden kann, falls die hinreichende Bedingung  $T_h/T_s < 0.5$  eingehalten wird. In einem Zeitmultiplexsystem mit zwei und mehr Übertragungskanälen ist dies stets der Fall.

### 3.9 Übertragung mehrerer Signale im Zeitmultiplex

Weist der Übertragungskanal eine genügend hohe Bandbreite auf, so ist es möglich, zur Übertragung eines Abtastwerts nur einen Bruchteil  $T_x/T_s$  der Zeit des Abtastintervalls  $T_s$  zu belegen. Mit  $T_x = nT$  ist das Zeitintervall bezeichnet, das für die Übermittlung einer Folge von  $n$  Code-Pulsen der Länge  $T$  beansprucht wird; im Falle der Übertragung von  $K$  Nachrichtenkanälen mit gleicher Bitrate gilt  $T_s = KT_x$ . Die Zeitintervalle zwischen den einzelnen,  $n$  Bit langen Pulsfolgen eines PCM-Signals können somit zur Übertragung anderer codierter Pulsfolgen verwendet werden. Dies ermöglicht die Mehrfachausnutzung des Übertragungskanals, indem die codierten Pulsfolgen mehrerer PCM-Signale in ihrer Zeitdauer jeweils um den Faktor  $K$  verkürzt und zeitlich verschachtelt übertragen werden. Man bezeichnet ein derartiges Vorgehen als *Zeitmultiplexverfahren* bzw. *Time-Division Multiplexing* (TDM).

Abbildung 3.5 illustriert das Konzept des TDM-Verfahrens anhand eines allgemeinen Blockdiagramms. Jede einzelne der Nachrichtenfunktionen  $s_k(t)$ ,  $k = 1, \dots, K$ , wird zunächst mit Hilfe eines *Tiefpassfilters* auf die Bandbreite  $\omega_{\text{max}}$  begrenzt, um unerwünschte Frequenzanteile zu eliminieren bzw. die Einhaltung des Abtasttheorems zu gewährleisten. Die Ausgänge der Filter werden anschliessend einem *Multiplexer* zugeführt. Er hat zwei wesentliche Funktionen:

1. Erzeugung kurzer, amplitudenmodulierter Pulse mit der Abtastkreisfrequenz  $\omega_s$ , die etwas über der doppelten Grenzkreisfrequenz  $2\omega_{\text{max}}$  des Tiefpassfilters liegt. Dieser Vorgang entspricht der zeitlichen Quantisierung und Signalabtastung.
2. Sequentielle zeitliche Verschachtelung  $K$  der Abtastwerte innerhalb eines Abtastintervalls der Länge  $T_s$ .

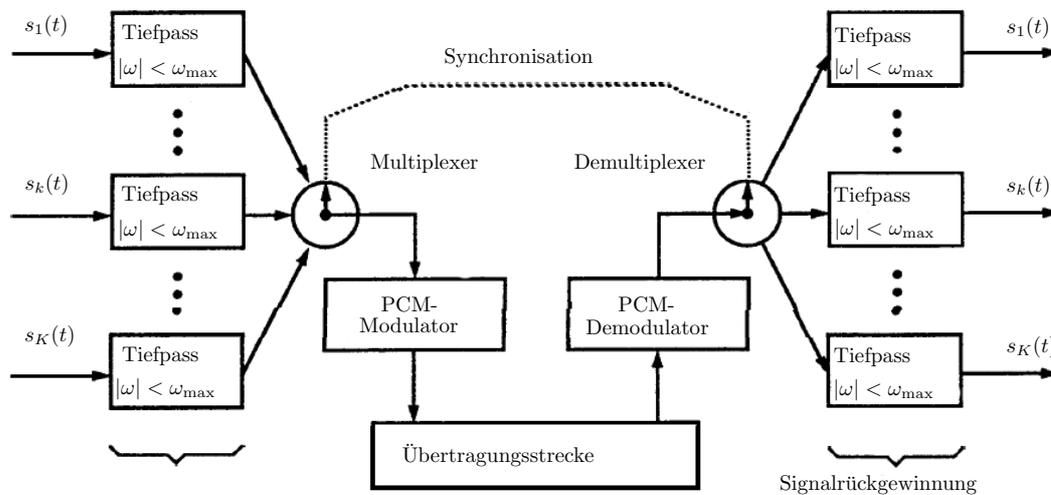


Abbildung 3.5: Blockdiagramm eines PCM-Zeitmultiplexsystems. Der PCM-Modulator sowie die entsprechende Baugruppe im Empfänger enthalten eine Vielzahl von Funktionen.

Die zweite Funktion ist die Essenz des Zeitmultiplexverfahrens. Nach dem Multiplexer wird das derart gebildete PAM-Zeitmultiplexsignal an den Eingang eines PCM-Modulators angelegt, welcher an seinem Ausgang ein (im allgemeinen trägerfrequentes) Signal erzeugt, das an die Charakteristiken der Übertragungsstrecke angepasst ist. Die Funktionen des PCM-Modulators in Figur 3.5 umfassen die in den Abschnitten 3.2 bis 3.4 diskutierten Operationen, nämlich die *Amplitudenquantisierung*, die Codierung sowie gegebenenfalls den Einsatz fehlerkorrigierender Codes. Auf der Empfangsseite erfolgt zuerst die Demodulation des PCM-Signals. Reziprok zur Funktionsweise des PCM-Modulators resultiert der Ausgang des PCM-Demodulators, zu dessen Aufgaben in diesem Falle die Demodulation (inklusive Signalregeneration), die Fehlerkorrektur aufgrund des fehlerkorrigierenden Codes sowie die Wiederherstellung der diskreten Amplitudenwerte (Decodierung) gemäss den Abschnitten 3.6 und 3.7 gehören. Das dabei erzeugte PAM-Zeitmultiplexsignal wird anschliessend mit Hilfe des Demultiplexers, der synchron zum Multiplexer arbeiten muss, zeitlich aufgefächert und den einzelnen Tiefpassfiltern zur *Signalrekonstruktion* zugeführt. Die *Synchronisation* des Demultiplexers ist für die korrekte Funktionsweise des Zeitmultiplexsystems unerlässlich und verdient besondere Beachtung; für die Wiedergewinnung der Takt- sowie der Rahmensynchronisation sind verschiedene Massnahmen notwendig, die in Abschnitt 3.10 kurz diskutiert werden.

### 3.10 Synchronisationsaspekte

Bei der Übertragung von Zeitmultiplexsignalen sind verschiedene Probleme in Bezug auf die zeitliche Synchronisation zwischen PCM-Sender und -Empfänger zu lösen. Zum einen muss der PCM-Empfänger die Zeitpunkte der Nulldurchgänge der einzelnen Pulsflanken mit einer ausreichenden Genauigkeit ermitteln, damit der Demodulator korrekt arbeiten kann; diese Aufgabe wird im allgemeinen als *Symbol synchronisation* bezeichnet und ist auch in digitalen Übertragungssystemen mit nur einem Benutzer notwendig. Als nächstes ist es erforderlich, die *Wortsynchronisation* sicherzustellen, also den Beginn eines Codewortes in einem Bitstrom zu erkennen. Bei Systemen

mit mehreren Benutzern müssen schliesslich die einzelnen Codeworte den richtigen Benutzern zugeordnet werden können; diese Funktion wird gemeinhin *Rahmensynchronisation* genannt und ermöglicht im selben Zuge auch die Wortsynchronisation.

Es erweist sich in der Praxis als zweckmässig, die einzelnen, im Zeitmultiplex angesteuerten Kanäle in eine übergeordnete Rahmenstruktur einzugliedern. Abbildung 3.6 verdeutlicht den Aufbau eines sogenannten Datenrahmens (*frame*): die zu den einzelnen Kanälen gehörenden, mit der Periode  $T_s$  sich wiederholenden  $K$  codierten Pulsfolgen werden zeitlich hintereinander angeordnet, wobei eine Rahmensynchronisationssequenz (*frame header*) oder *Präambel* vor dem zu Kanal 1 gehörenden Datenpaket gesendet wird. Wahlweise kann der Datenrahmen mit einer (in der Praxis häufig zur Fehlererkennung eingesetzten) *Postambel* oder Schlusssequenz abgeschlossen werden (*frame trailer*). Um die Rahmensynchronisation zu bewerkstelligen, sind verschiedene Methoden anwendbar.

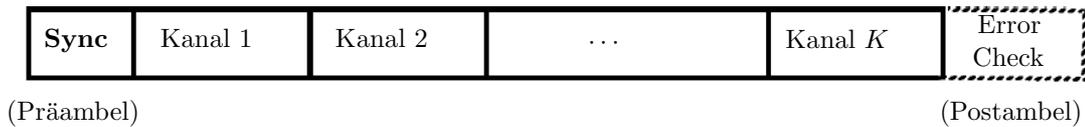


Abbildung 3.6: Allgemeine Rahmenstruktur für die Ansteuerung mehrerer Kanäle im Zeitmultiplex. Als erstes wird eine Synchronisationssequenz gesendet, anschliessend die Datenpakete der Kanäle 1 bis  $K$ , gefolgt von einer (optionalen) Schlusssequenz, die häufig zur Fehlererkennung eingesetzt wird.

## Kapitel 4

# Bestimmung des Signal-Rauschleistungsverhältnisses in einem PCM-System

Die Übertragungsqualität in einem PCM-System wird von zwei unterschiedlichen Ursachen beeinträchtigt, die als unabhängig betrachtet werden können und in den folgenden Abschnitten behandelt werden. Im ersten Abschnitt wird der Einfluss des *Rauschens auf dem Übertragungskanal* kurz erläutert; der nachfolgende Abschnitt befasst sich mit dem *Quantisierungsrauschen* und vergleicht die erzielbare Übertragungsqualität eines PCM-Systems mit derjenigen einiger klassischer analoger Modulationsverfahren.

### 4.1 Kanalrauschen und Übertragungsfehlerwahrscheinlichkeit

Eine wichtige Kenngrösse digitaler Übertragungssysteme bildet die *Bitfehlerwahrscheinlichkeit*  $P_b$ . Unter der Annahme eines binären PCM-Signals  $x_{\text{PCM}}(t)$  mit den Pulsamplituden  $A$  und  $-A$  (wie beispielsweise in Abbildung 3.3 (b) dargestellt), dem normalverteiltes, mittelwertfreies Rauschen  $n(t)$  mit der Varianz  $\sigma^2$  überlagert ist, ergibt sich  $P_b$ , zu

$$P_b = \frac{1}{2}P\{n(t) > A\} + \frac{1}{2}P\{n(t) < -A\}, \quad (4.1)$$

wobei angenommen wurde, dass beide Amplitudenwerte  $A$  und  $-A$  mit derselben Wahrscheinlichkeit  $1/2$  auftreten. Abbildung 4.1 zeigt die mit dem Faktor  $1/2$  gewichteten bedingten Wahrscheinlichkeitsdichten  $p_r(\rho|x_{\text{PCM}}(t) = \pm A)$  des empfangenen Summensignals  $r(t) = x_{\text{PCM}}(t) + n(t)$  für die beiden Fälle  $x_{\text{PCM}}(t) = A$  und  $x_{\text{PCM}}(t) = -A$ ; die schraffierten Flächen entsprechen den Wahrscheinlichkeiten  $P\{n(t) > A\}$  bzw.  $P\{n(t) < -A\}$  und sind gleich gross.

Mit der Wahrscheinlichkeitsdichte der Normalverteilung

$$p_n(\eta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\eta^2}{2\sigma^2}} \quad (4.2)$$

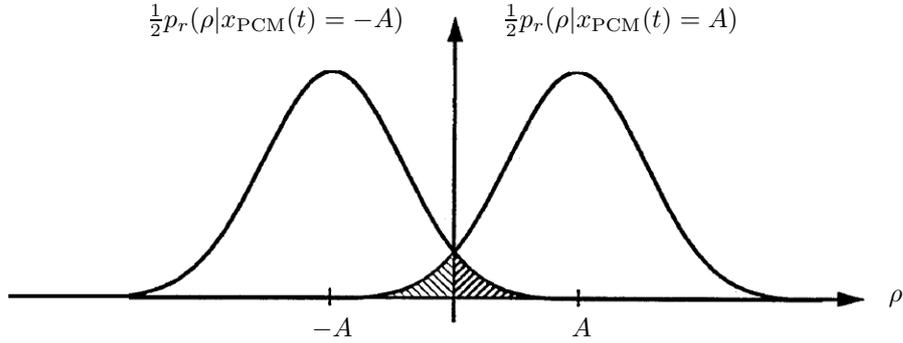


Abbildung 4.1: Bedingte Wahrscheinlichkeitsdichtefunktion eines empfangenen binären PCM-Signals  $r(t) = x_{\text{PCM}}(t) + n(t)$  mit den Pulsamplituden  $A$  und  $-A$ , das von normalverteiltem, mittelwertfreiem Rauschen gestört wird.

und dem Fehlerintegral

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{x}} \int_0^x e^{-y^2} dy \quad (4.3)$$

folgt die Fehlerwahrscheinlichkeit  $P_b$  schliesslich zu

$$P_b = \frac{1}{2} \left( 1 - \operatorname{erfc} \left( \frac{A}{\sqrt{2}\sigma} \right) \right). \quad (4.4)$$

Wird die Signalleistung gemäss  $S_1 = A^2$  und die Rauschleistung als  $N_1 = \sigma^2$  definiert, so erhält man das Signal-Rauschleistungsverhältnis

$$\gamma = \frac{S_1}{N_1} = \frac{A^2}{\sigma^2} \quad (4.5)$$

und damit die Fehlerwahrscheinlichkeit

$$P_b = \frac{1}{2} \left( 1 - \operatorname{erfc} \left( \sqrt{\frac{\gamma}{2}} \right) \right). \quad (4.6)$$

Abbildung 4.2 zeigt  $P_b$ , in Funktion von  $\gamma$ , wobei Abszisse und Ordinate in logarithmischem Massstab dargestellt sind.

Aus (4.6) ergibt sich, dass die Fehlerwahrscheinlichkeit am Eingang des Empfängers für  $\gamma > 20$  dB verschwindend klein ist, bei einer Abnahme des Signal-Rauschverhältnisses auf 15 dB jedoch rapide zunimmt. In der Regel gelten Fehlerwahrscheinlichkeiten von  $10^{-5}$  und höher für eine Datenübertragung als nicht akzeptabel. Dieser Fall tritt bei einem Signal-Rauschleistungsverhältnis von rund 12.5 dB ein.

Zusammenfassend lässt sich feststellen, dass die Wahrscheinlichkeit von Übertragungsfehlern ab einem bestimmten Signal-Rauschleistungsverhältnis praktisch *nicht mehr ins Gewicht fällt*. Unterhalb einer kritischen Grenze nehmen jedoch die Bitfehler rapide zu. Dieser *stark ausgeprägte Schwelleffekt* ist für das Verhalten aller digitalen Übertragungsverfahren charakteristisch. Fehlerkorrigierende Codes haben lediglich den Effekt, die Schwelle zu senken; auf Abbildung 4.2 übertragen

bedeutet dies eine Verschiebung der Kurve um einige Dezibel nach links, je nach Wirksamkeit des eingesetzten fehlerkorrigierenden Codes. Am grundsätzlichen Verhalten des digitalen Übertragungssystems ändert sich jedoch nichts: Bitfehler auf dem Übertragungskanal können bei Gewährleistung eines ausreichenden Signal-Rauschleistungsverhältnisses praktisch ausgeschlossen werden.

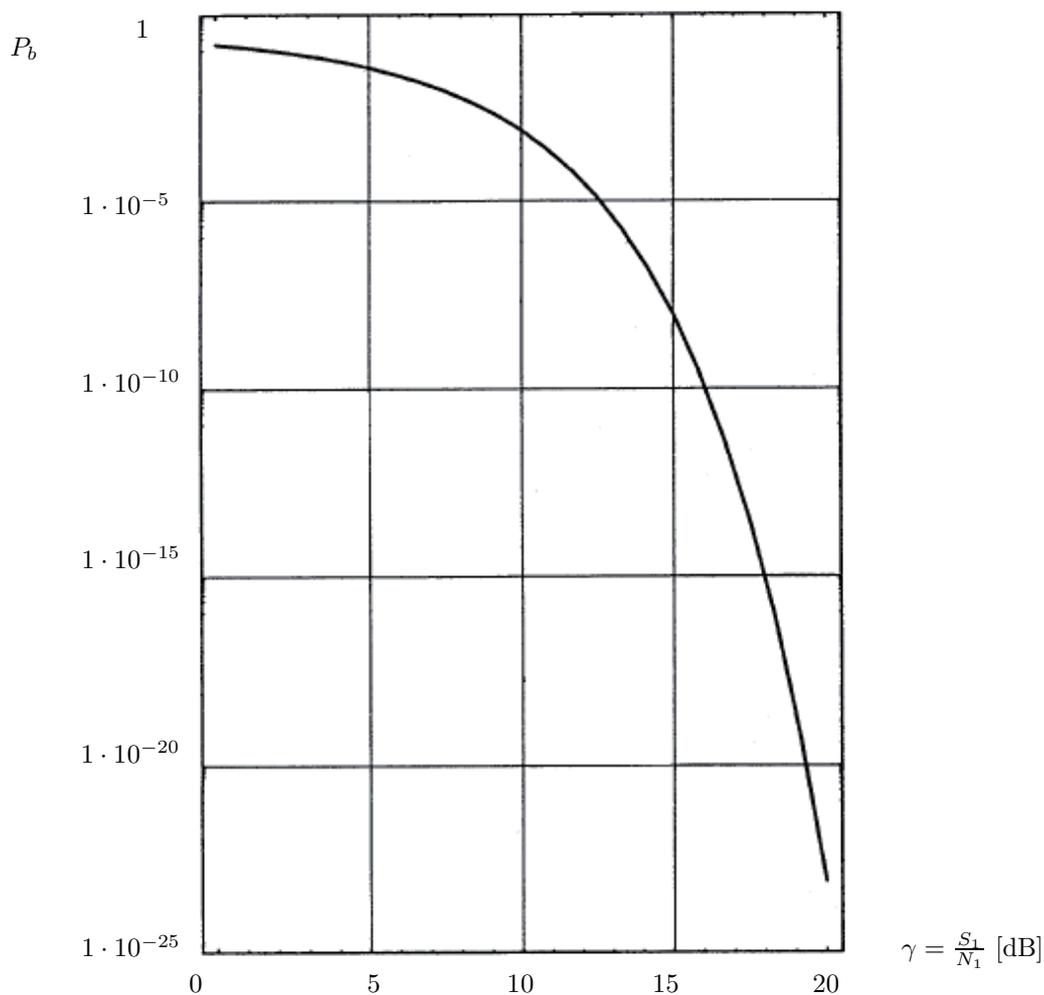


Abbildung 4.2: Fehlerwahrscheinlichkeit  $P_b$  in Funktion des Signal-Geräuschleistungsverhältnisses  $\gamma = S_1/N_1$  in dB.

## 4.2 Quantisierungsrauschen

Unter der Voraussetzung, dass keine Bitfehler auf dem Übertragungskanal auftreten, soll nun der Einfluss der Amplitudenquantisierung auf das Signal-Rauschleistungsverhältnis  $S_2/N_2$  am Empfänger Ausgang des PCM-Systems betrachtet werden.

Ausgehend von der *Annahme*, dass die Amplitude der analogen Nachrichtenfunktion  $s(t)$  im sym-

metrischen Intervall  $[-A_0, A_0]$  *gleichverteilt* ist, ergibt sich für die mittlere Leistung  $S_2$  des Signals (auch nach dessen Rekonstruktion)

$$S_2 = \int_{-A_0}^{A_0} \frac{1}{2A_0} s^2 ds = \frac{A_0^2}{3}. \quad (4.7)$$

Angesichts des Verhaltens des Quantisierungsfehlers  $e_q = s - s_q$  in Abbildung 3.4 (b) und der obigen Annahme über die Verteilung des Eingangspiegels  $s$  ergibt sich ebenfalls eine Gleichverteilung für  $e_q$ , im symmetrischen Intervall  $[-\Delta/2, \Delta/2]$  und man erhält für die mittlere Leistung des Quantisierungsfehlers nach ähnlicher Rechnung wie in (4.7)

$$N_Q = \int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} q^2 dq = \frac{\Delta^2}{12}. \quad (4.8)$$

Zieht man nun in Betracht, dass für eine grosse Anzahl von Quantisierungsstufen  $N$  die Beziehung  $2A_0 = N \cdot A$  gilt, ergibt sich das Signal-Rauschleistungsverhältnis  $S_2/N_2$  am Empfängeranfang unter Berücksichtigung von (4.7) und (4.8) zu

$$\frac{S_2}{N_2} = \frac{S_2}{N_Q} = N^2. \quad (4.9)$$

Dieses einfache Resultat besagt, dass das Verhältnis der Signalleistung  $S_2$  zur Leistung  $N_Q$  des Quantisierungsfehlers *lediglich von der Anzahl Quantisierungsstufen abhängt*. Sind  $k$  Bits für die Darstellung der  $N$  Quantisierungsstufen notwendig, so folgt mit  $N = 2^k$

$$\frac{S_2}{N_2} = \frac{S_2}{N_Q} = 2^{2k}, \text{ oder} \quad (4.10)$$

$$10 \log \left( \frac{S_2}{N_Q} \right) [\text{dB}] \approx 6k. \quad (4.11)$$

# Kapitel 5

## Das PCM-Experimentalsystem

Dieser Abschnitt enthält eine Übersicht der MODICOM 3/1 - und MODICOM 3/2 - Steckplatinen (siehe Abbildungen 5.1 und 5.2) und ihrer wichtigsten Funktionseinheiten. Das PCM-Experimentalsystem besteht aus den folgenden Baugruppen: Auf MODICOM 3/1:

- 1) Anschlüsse für die Stromversorgung (*power supply inputs*),
- 2) Funktionsgeneratorblock (*function generator*),
- 3) Taktgeberschaltblock (*transmitter timing logic*),
- 4) Synchronisationscode-Generator (*sync code generator*),
- 5) PCM-Sender,

auf MODICOM 3/2:

- 6) Anschlüsse für die Stromversorgung (*power supply inputs*),
- 7) Taktregenerationsschaltung (*clock regeneration circuit*),
- 8) Synchronisationscode-Detektor(*sync code detector*);
- 9) Demultiplexerlogik (*receiver timing logic*),
- 10) PCM-Empfänger.

Die Block-Diagramme in Abbildung 5.3 verdeutlichen die Anordnung der Baugruppen auf den MODICOM 3 - Platinen.

### 5.1 Anschlüsse für die Stromversorgung

Die erforderlichen Versorgungsspannungen für beide Versuchsanordnungen können einem regelbaren Netzteil entnommen werden. Die mit 0 V bezeichneten Klemmen der MODICOM 3 - Platinen (siehe Abbildung 5.4) sind mit dem Masseanschluss des Netzteils zu verbinden; daraufhin ist das Netzteil ohne weitere Anschlüsse in Betrieb zu nehmen. An den verschiedenen Ausgängen sind zweimal die Spannung 12 V und einmal die Spannung 5 V einzustellen. Anschliessend wird das Netzteil wieder ausgeschaltet. Die in Abbildung 5.4 gezeigte Verkabelung des Netzteils mit den entsprechenden Anschlüssen der MODICOM - Platinen stellt die korrekte Stromversorgung sicher. **Auf die richtige Ausführung der obengenannten Schritte ist sorgfältig zu achten!**

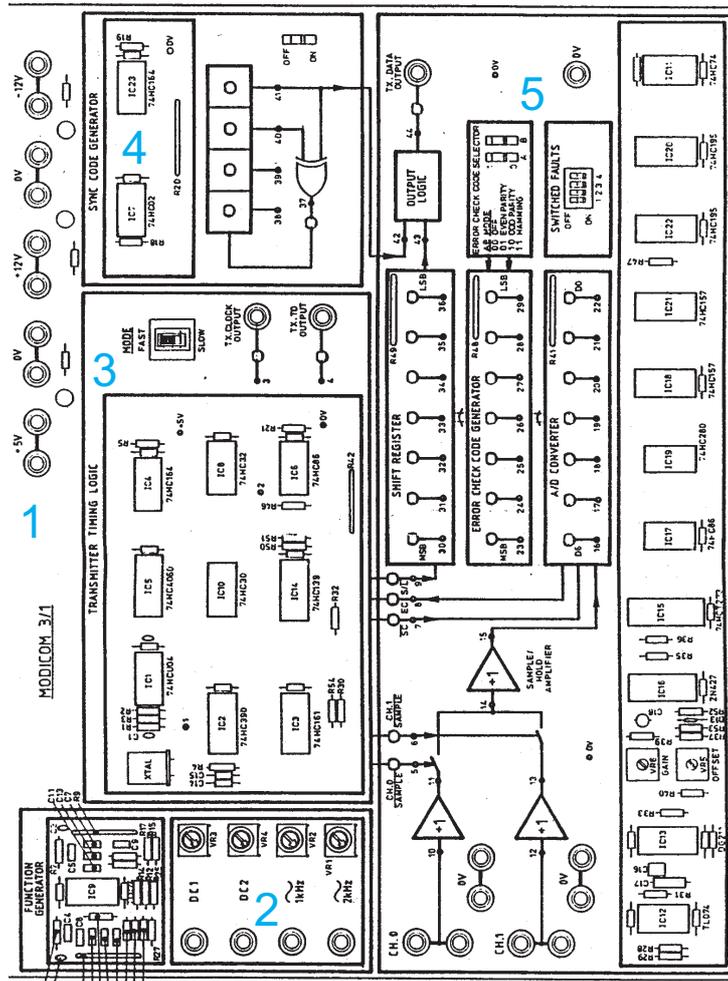


Abbildung 5.1: Komplettes Layout der MODICOM 3/1 - Steckplatine im Detail.

## 5.2 Funktionsgeneratorblock

Diese Baugruppe stellt vier Signale zur Verfügung, welche an die verschiedenen Eingänge des PCM-Senders angeschlossen werden können. Zwei Gleichspannungen und zwei sinusförmige Signale der Frequenzen 1 kHz sowie 2 kHz mit einstellbarer Amplitude (siehe Abbildung 5.5) sollen als Nachrichtensignale verwendet werden.

## 5.3 Taktgeberschaltblock

Diese Funktionseinheit erzeugt zwei zeitlich gegeneinander versetzte Schaltfunktionen, um die Erzeugung von zwei PAM-Signalen im Zeitmultiplex zu ermöglichen. Zwei zusätzliche Clock-Signale, die zur Takt- bzw. Rahmensynchronisation verwendet werden können, unterstützen wahlweise die Empfängersynchronisation über zwei von der Datenleitung getrennte Verbindungen (siehe Abbil-

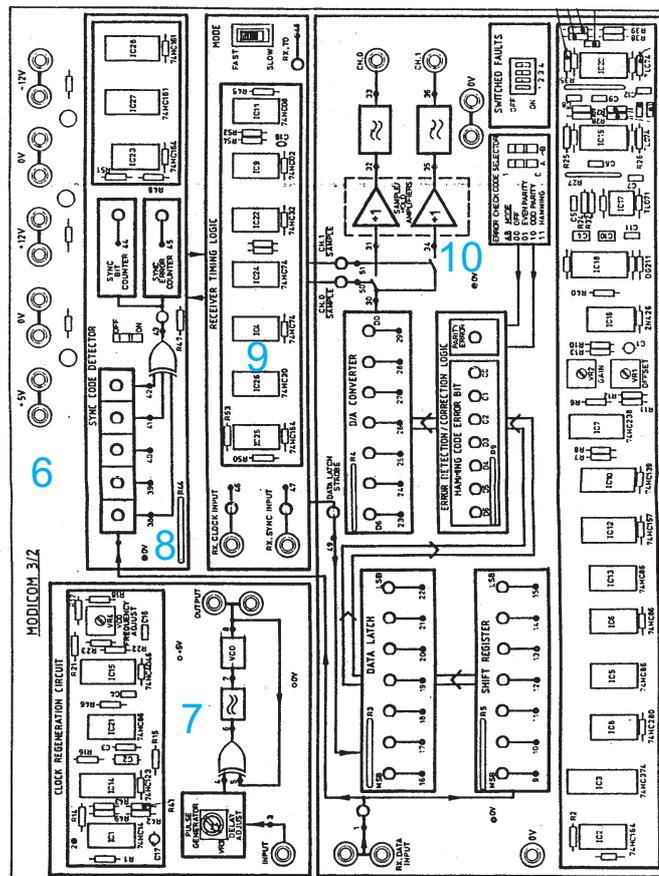


Abbildung 5.2: Komplettes Layout der MODICOM 3/2 - Steckplatine im Detail.

dung 5.6). Zum besseren Verständnis kann der Betrieb im *SLOW MODE* erfolgen; zur Umschaltung ist ein Schiebeschalter vorgesehen.

## 5.4 Synchronisationscode-Generator

Ein linear rückgekoppeltes Schieberegister erzeugt eine Maximallängensequenz, welche zur Rahmensynchronisation verwendet und mit dem Schiebeschalter ON/OFF aktiviert wird (siehe Abbildung 5.7). Die Funktionsweise der Rahmensynchronisation ist in Abschnitt A.1 detailliert beschrieben.

## 5.5 PCM-Sender

Zwei Nachrichtensignale (CH.0 und CH.1) werden im analogen Teil des Senders alternierend abgetastet und zu einem PAM-Zeitmultiplexsignal zusammengefügt (siehe Abbildung 5.8). Nach einer Analog/Digital-Wandlung können wahlweise Massnahmen zur Fehlererkennung oder -korrektur ergriffen werden; die beiden Schiebeschalter ermöglichen die Wahl des gewünschten Verfahrens.

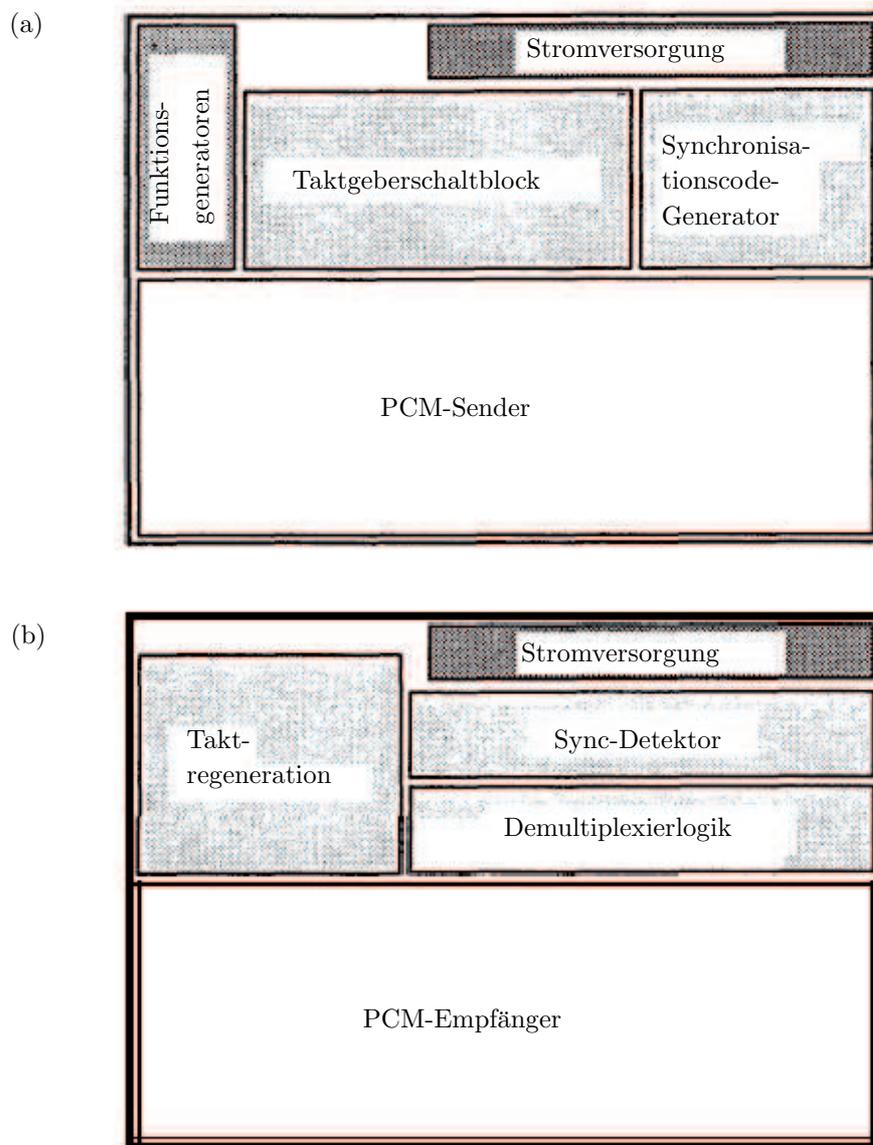


Abbildung 5.3: Anordnung der Baugruppen. (a) MODICOM 3/1; (b) MODICOM 3/2.

Das Schieberegister dient der Parallel/Seriell-Wandlung und erzeugt somit ein PCM-Signal. In einer Logikschaltung vor dem PCM-Senderausgang TX. DATA OUTPUT werden wahlweise Synchronisationsimpulse hinzugefügt. Schaltbare Fehler können durch die Kippschalter *SWITCHED FAULTS* erzeugt werden.

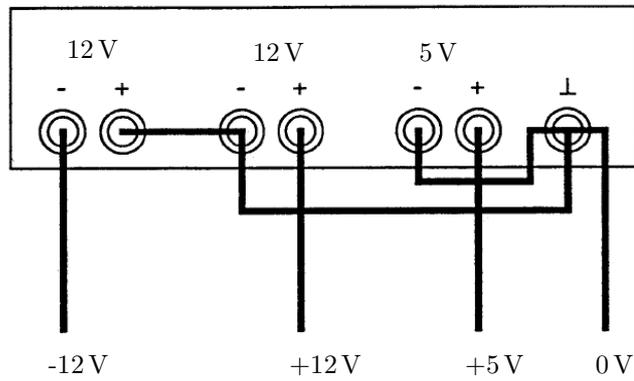


Abbildung 5.4: Verkabelung des Netzteils mit den Anschlüssen für die Stromversorgung.

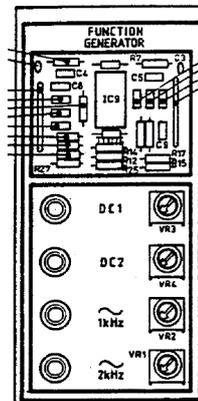


Abbildung 5.5: Funktionsgeneratorblock.

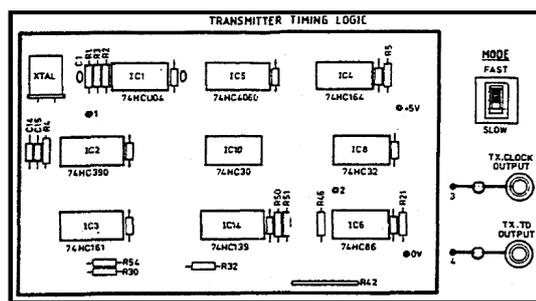


Abbildung 5.6: Taktgeberschaltblock.

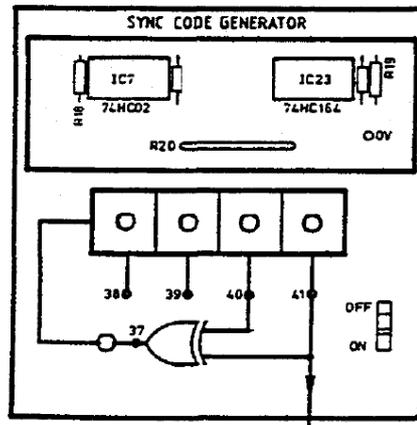


Abbildung 5.7: Synchronisationscode-Generator.

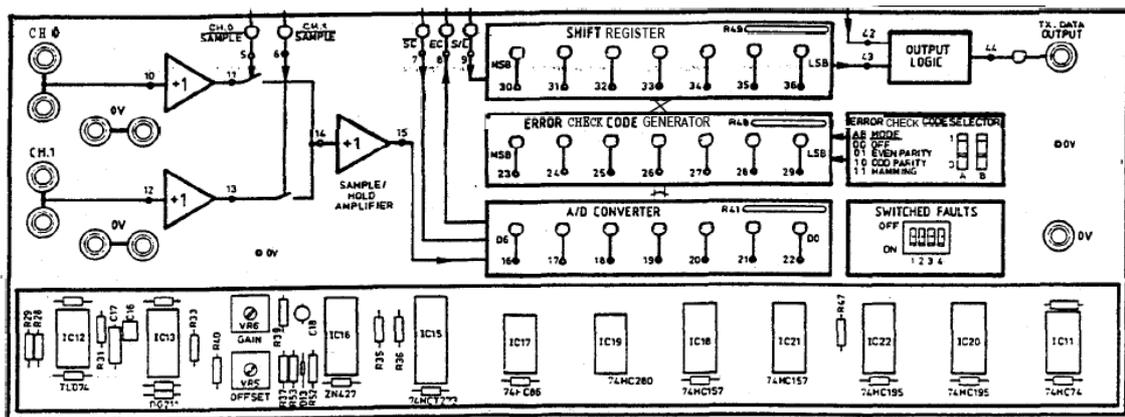


Abbildung 5.8: PCM-Sender.

## 5.6 Anschlüsse für die Stromversorgung

Die erforderlichen Versorgungsspannungen für das MODICOM 3/2 - Board sind von der MODICOM 3/1 - Platine zu beziehen (s.a. Abschnitt 5.1).

## 5.7 Taktregenerationsschaltung

Ein Phasenregelkreis oder PLL (*Phase Locked Loop*) dient der Erzeugung des Taktsignals aus dem PCM-Signal (siehe Abbildung 5.9) und gelangt bei den Synchronisationsvarianten 2 und 3 (siehe Abschnitt 5.9) zum Einsatz.

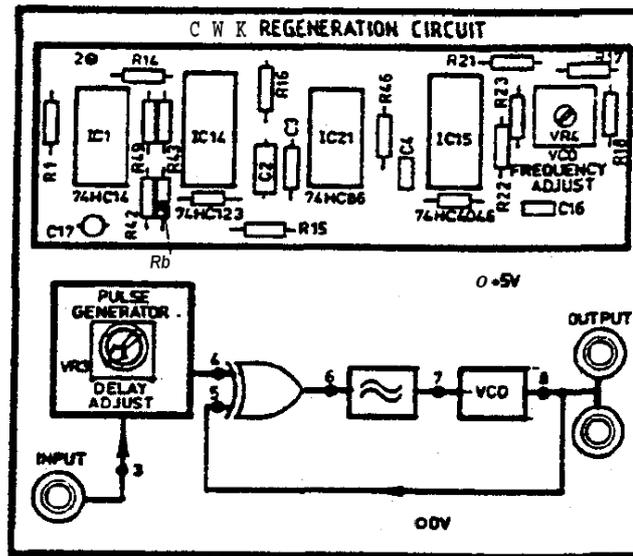


Abbildung 5.9: Taktregenerationsschaltung.

## 5.8 Synchronisationscode-Detektor

Diese Funktionseinheit kann mit dem Schiebeschalter ON/OFF hinzugeschaltet werden (siehe Abbildung 5.10) und ermöglicht die Detektion des Synchronisationssignals wie in Abschnitt 5.4 beschrieben.

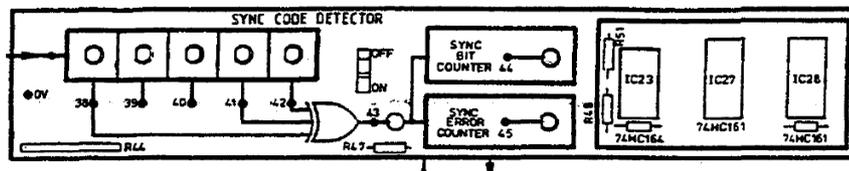


Abbildung 5.10: Synchronisationscode-Detektor.

## 5.9 Demultiplexierlogik

Diese Baugruppe stellt das empfängerseitige Komplement zum Taktgeberschaltblock dar. Drei verschiedene Varianten sind möglich, um die Takt- und Rahmensynchronisation des PCM-Empfängers mit dem Sender sicherzustellen:

1. Daten, Takt- und Rahmensynchronisationssignale werden auf drei getrennten Leitungen übermittelt.

2. Neben der Datenleitung existiert lediglich eine zweite Verbindung für das Rahmensynchronisationssignal; die Taktsynchronisation wird mit Hilfe des Phasenregelkreises in der Taktregenerationsschaltung abgeleitet.
3. Eine Leitung genügt für die Datenübertragung und Synchronisation, falls der Synchronisationscode-Generator und -Detektor verwendet werden.

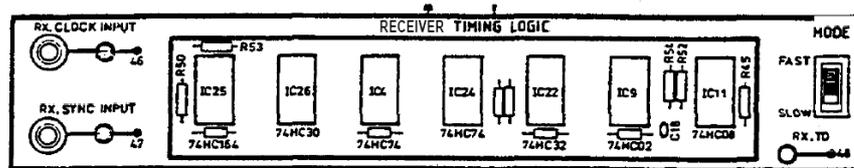


Abbildung 5.11: Demultiplexierlogik.

Es werden - wie im senderseitigen Taktgeberschaltblock - Symboltaktsignale sowie zwei zeitverschobene Schaltfunktionen erzeugt, die im PCM-Empfänger zur Ansteuerung des Demultiplexers benötigt werden. Die Anschlüsse RX CH.0 und RX CLOCK entsprechen den senderseitigen Ausgängen TX CH.0 und TX CLOCK. Zum besseren Verständnis kann der Betrieb im *SLOW MODE* erfolgen (Umschaltung mittels Schiebeschalter).

## 5.10 PCM-Empfänger

Der PCM-Empfänger führt die inversen Operationen des PCM-Senders durch: Seriell/Parallel-Wandlung, Zwischenspeicherung, Fehlerdetektion bzw. -korrektur, D/A-Wandlung in ein PAM-Zeitmultiplexsignal, Demultiplexierung sowie Signalrekonstruktion (siehe Abbildung 5.12). Es ist darauf zu achten, dass der ERROR CHECK CODE SELECTOR des Empfängers denselben Betriebsmodus wie jener des Senders verwendet.

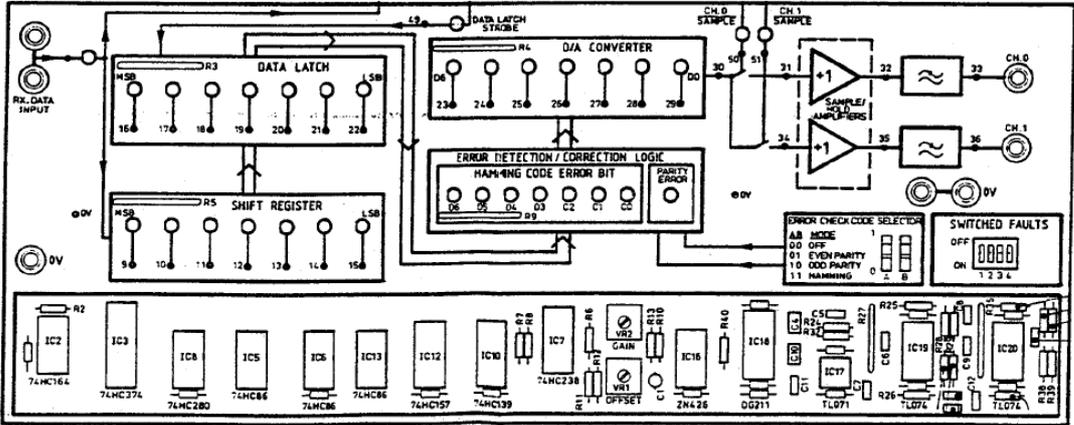


Abbildung 5.12: PCM-Empfänger.

# Kapitel 6

## Aufgaben

### 6.1 Theoretischer Teil (Vorbereitung)

#### Aufgabe 1

Ausgehend vom Codegenerator in Abbildung A.1 (a) und einem Anfangszustand “1111” ist eine Liste der hintereinander auftretenden Schieberegisterzustände zu erstellen. Man lese daraus die (periodische) Maximallängensequenz ab!

#### Aufgabe 2

Für den systematischen (7,4)-Hamming Code mit der Erzeugenden

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (6.1)$$

ermittle man die Parity-Check-Matrix  $\mathbf{H}$ . Wieviele Syndromvektoren  $\mathbf{s} = \mathbf{e}\mathbf{H}^T$  gibt es? Zwischen wievielen Schätzwerten  $\hat{\mathbf{e}}(\mathbf{s})$  für den Fehler  $\mathbf{e}$  kann folglich ein Syndrom-Decoder für diesen Code überhaupt unterscheiden? Wieviele mögliche Fehlervektoren  $\mathbf{e}$  gibt es?

Man erstelle intuitiv eine Liste der wahrscheinlichsten Fehlervektoren  $\hat{\mathbf{e}}(\mathbf{s})$ , ohne Berechnungen durchzuführen! (Hinweis:  $\hat{\mathbf{e}} = [0\ 0\ 0\ 0\ 0\ 0\ 0]$  muss neben sieben anderen Fehlervektoren dazugehören.)

Durch Einsetzen der wahrscheinlichsten Fehlervektoren  $\hat{\mathbf{e}}(\mathbf{s})$  in  $\mathbf{s} = \mathbf{e}\mathbf{H}^T$  sind die entsprechenden Syndromvektoren zu ermitteln.

#### Aufgabe 3

Aus dem demodulierten Vektor  $\mathbf{y} = [1\ 1\ 0\ 1\ 1\ 0\ 1]$  soll zunächst dessen Syndrom  $\mathbf{s}$  und anschliessend mit Hilfe der in Aufgabe 2 erstellten Syndromtabelle  $\mathbf{s} \rightarrow \hat{\mathbf{e}}(\mathbf{s})$  die Schätzung für das Codewort  $\hat{\mathbf{c}}$  gefunden werden. Auf welche Schätzung der Datensequenz  $\hat{\mathbf{b}}$  führt dies? (Hinweis: es handelt sich um einen systematischen Code!)

## 6.2 Praktischer Teil

### Aufgabe 4

Als erstes sind die beiden Steckplatten MODICOM 3/1 und MODICOM 3/2 an die Stromversorgung anzuschliessen. **Die angegebenen Speisespannungen dürfen auf keinen Fall überschritten werden!** Auf beiden Einheiten sollen die jeweils vier Kippschalter SWITCHED FAULTS auf die Stellungen OFF eingestellt werden; ausserdem ist die Übertragung ohne Fehlerkorrektur vorgesehen (ERROR CHECK CODE SELECTOR auf "00"). Als Betriebsart wähle man den FAST MODE.

Zunächst sollen die verschiedenen Signale im Sender untersucht werden. Die Signalquelle mit einem 1 kHz-Sinussignal gehe bei einer Amplitude von 2 V<sub>pp</sub> auf Kanal 0 (CH.0), die 2 kHz-Quelle auf Kanal 1 (CH.1), wobei ihre Amplitude auf 1 V<sub>pp</sub> einzustellen ist. Man stelle das PAM- "sample and hold"-Zeitmultiplexsignal (Pin 15) auf dem Oszilloskop dar und vergleiche es mit den modulierenden Eingangssignalen (Pins 10 und 12), indem man die Signalverläufe auf dem KO zur Deckung bringt.

Wie gross ist die Abtastrate  $f_s = 1/T_s$ ? Wieviel beträgt die Pulsdauer  $T_h$ , des PAM-Signals? Man ermittle ebenfalls die Pulsdauer  $T_p$ , des PCM-Signals am Senderausgang (Pin 43)!

**Wichtig: Bei Änderungen der Beschaltung der Kanaleingänge CH.0 und CH.1 ist die Stromversorgung stets auszuschalten!**

### Aufgabe 5

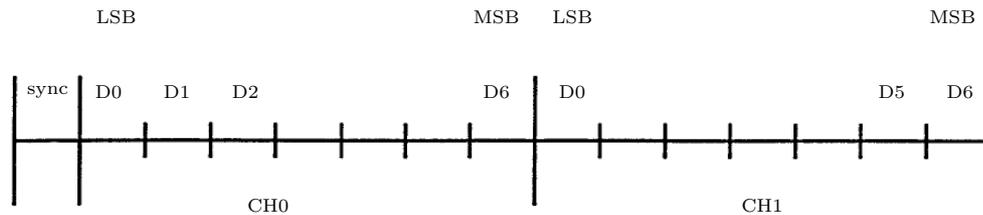
Um die Codetabelle zu ermitteln, ist die Stromversorgung auszuschalten, die Gleichspannungsquelle DC1 mit CH.0 und CH.1 zu verbinden und nach Einschalten der Speisung die Gleichspannung am Eingang über den gesamten Aussteuerungsbereich durchzustimmen. Wie wird die Eingangsspannung auf den Binärcode abgebildet? Welchem Eingang entspricht das Codewort "1000000"? Man ermittle den Quantisierungsschritt indem die Differenz der zu den Codeworten "1000000" und "1001010" entsprechenden Eingangsspannungen gemessen wird! In welchem Intervall können folglich die Spannungen am Eingang des AD-Wandlers übersteuerungsfrei quantisiert werden?

Wie hoch ist das Signal-Quantisierungsrauschleistungsverhältnis beim vorliegenden System ohne Fehlerkorrektur? Auf wieviel dB verringert es sich beim Einsatz des (7,4)-Hamming Codes (4 Bit-Quantisierung)?

### Aufgabe 6

Man betrachte nun das übertragene PCM-Signal ohne Einfügung der Synchronisationspulse am Ausgang des Senders (Pin 42). Dieses ist zusammen mit den Rahmensynchronisationsimpulsen am Ausgang TX.T0 OUTPUT (Pin 4) auf dem KO darzustellen. Am Eingang des Systems soll die Gleichspannungsquelle DC1 auf Kanal 0 bzw. DC2 auf Kanal 1 gegeben werden. Dabei verändere man die Eingangsspannungen über den gesamten Aussteuerungsbereich der Quellen.

In welcher Reihenfolge werden die einzelnen Bits der Codeworte übermittelt? Wird das MSB ("most significant bit") oder das LSB ("least significant bit") zuerst übertragen? Die Eingangsspannungen DC1 und DC2 sind derart einzustellen, dass ausschliesslich die Codeworte "1111111" gesendet werden. Auf diese Weise ist es möglich, die Positionen der Synchronisationsimpulse sichtbar zu machen. Man vervollständige das untenstehende Schema der Rahmenstruktur (CH 0, CH 1, MSB, LSB, D0...D6)!



### Aufgabe 7

Das Zweikanal-PCM-Zeitmultiplexsystem werde wie in Aufgabe 6 betrieben; die Gleichspannungsquellen sollen jedoch so eingestellt werden, dass das Codewort "1000000" auf dem Kanal 0 bzw. das Codewort "1010000" auf dem Kanal 1 gesendet wird. Die Einstellung wird mit Hilfe des Ausgangssignals (Pin 42) auf dem KO wesentlich erleichtert. Ferner ist der Empfänger anzuschliessen; dazu müssen die Anschlüsse TX. DATA OUTPUT und RX. DATA INPUT miteinander verbunden werden. Um die Takt- und die Rahmensynchronisation herzustellen, sind die senderseitigen Ausgänge TX. CLOCK OUTPUT und TX.T0 OUTPUT an die Buchsen RX.CLOCK INPUT beziehungsweise RX. SYNC INPUT des Empfängers anzuschliessen. Die Schalter am Synchronisationscodegenerator sowie am Synchronisationscodedetektor können auf Stellung OFF belassen werden.

Man verifiziere im SLOW MODE die Rahmenstruktur (1+7+7Bits) sowie die korrekte Funktionsweise des Multiplexvorgangs, des AD-Wandlers, der Parallel/Seriell-Wandlung sowie der entsprechenden inversen Operationen im Empfänger! **Vorsicht:** Wegen der Möglichkeit transients Vorgänge, die im SLOW MODE eventuell zu kurze Zeitkonstanten aufweisen können, wird von einer Anzapfung der Pins 14 und 15 zu Messzwecken abgeraten.

### Aufgabe 8

Ausgehend vom Betrieb im SLOW MODE können nun der Synchronisationscodegenerator sowie der Synchronisationscodedetektor eingeschaltet werden (Schalter auf ON). Man erstelle eine Tabelle der aufeinanderfolgenden Schieberegisterzustände und verifiziere das Resultat von Aufgabe 1. Welche M-Sequenz ergibt sich als Synchronisationscode, beginnend mit dem Schieberegisterzustand "1111"?

### Aufgabe 9

Das PCM-System soll nun im Zustand FAST MODE betrieben werden, wobei die Einstellung der Gleichspannungsquellen DC1 und DC2 derart zu wählen ist, dass ausschliesslich das Codewort "0000000" übermittelt wird. Man stelle die Synchronisationscodesequenz (Pin 42) auf dem KO dar. Die Synchronisationsimpulse sind mit der Codesequenz zu vergleichen. Wie lang ist ein sogenannter "multiframe"?

### Aufgabe 10

Am Eingang des PCM-Systems soll nun das 1 kHz-Sinussignal auf Kanal 0, die 2 kHz-Schwingung auf Kanal 1 gegeben werden. Dabei ist eines der Eingangssignale (z.B. CH.0 an Pin 10) mit dem entsprechenden Ausgang des DA-Wandlers (z.B. Pin 32) bzw. mit dem rekonstruierten Signal (Pin 33) zu vergleichen. Welches ist der Aussteuerungsbereich des D/A-Wandlers? Man variiere dazu die Amplitude des Eingangssignals, bis Verzerrungen des Ausgangssignals bemerkbar werden.

Bei Verwendung des (7,4)-Hamming Codes (ERROR CHECK CODE SELECTOR auf "11") überprüfe man die Verringerung der Anzahl Quantisierungsstufen auch am Ausgangssignal. Eine Gleichspannung am Eingang des PCM-Systems eignet sich gut für die Zählung der Quantisierungsniveaus am Ausgang.

### Aufgabe 11

Mit dem 1 kHz-Sinussignal auf Kanal 0 sowie dem 2 kHz-Sinus auf Kanal 1 können die verschiedenen Varianten der Synchronisation gemäss Abschnitt 5.9 überprüft werden. Auf dem KO stelle man das Eingangssignal von Kanal 0 sowie den Ausgang von Kanal 1 dar. Die Leitung für die Rahmensynchronisation (TX. T0 OUTPUT nach RX. SYNC INPUT) ist zu entfernen. Dies entspricht der Variante 2 gemäss Abschnitt 5.9. Die Schalter am Synchronisationscodegenerator sowie am Synchronisationscodedetektor stehen auf Stellung OFF.

Was geschieht, wenn die Taktsynchronisationsleitung (TX. CLOCK nach RX.CLOCK) für kurze Zeit unterbrochen und anschliessend wiederhergestellt wird? Man wiederhole diesen Vorgang mehrmals und beschreibe den beobachteten Effekt.

Weswegen hat die Veränderung der Amplitude beim Eingang von CH.0 einen Einfluss auf den Ausgang von CH.1?

Besteht die gegenseitige Abhängigkeit der Übertragungskanäle auch noch nach Wiederherstellung der Rahmensynchronisation?

### Aufgabe 12

Ausgehend vom Betrieb ohne Taktsynchronisation wird nun der Schalter am (senderseitigen) Synchronisationscodegenerator auf Stellung ON gebracht. Verändert sich der Ausgang des Systems?

Was geschieht, wenn man nun am Empfänger die Synchronisationscodedetektion einschaltet?

Ausgehend vom synchronisierten Betriebszustand ist nun jeweils der Schalter für die Synchronisation am Sender sowie jener am Empfänger zu betätigen. Welche Unterschiede sind festzustellen, und warum?

### Aufgabe 13

In einem letzten Schritt kann man sich ebenfalls der Taktsynchronisationsleitung entledigen, indem an ihrer Stelle der Empfängereingang RX.DATA INPUT zunächst mit der Taktregenerationsschaltung verbunden wird, deren Ausgang wiederum an die Klemme RX.CLOCK INPUT gelangt. Mittels des Phasenregelkreises (PLL) kann also der Bit-Takt aus dem PCM-Signal rekonstruiert werden.

Bei aktivierter Rahmensynchronisationsschaltung(ON-ON) verstelle man das Drehpotentiometer VCO FREQUENCY ADJUST des PLL solange, bis die Rahmensynchronisation gewährleistet werden kann. Dies ist erst dann der Fall, wenn die LED "SYNC BIT COUNTER" hell leuchtet. Der kleine Fangbereich des PLL ist dabei zu beachten.

### Aufgabe 14

Als Fehlererkennungsverfahren ist die Bildung von Parity-Bits sehr verbreitet. Mittels der Schalter ERROR CHECK CODE SELECTOR kann durch Einstellen von "01" die "even parity", bei der Wahl "10" die "odd parity" gebildet werden; dies geschieht auf Kosten des niederwertigsten Bits (LSB).

Man gebe die Gleichspannungsquelle DC1 auf die Eingänge von CH.0 und CH.1 und überprüfe die Parity-Bildung beim Sender und Empfänger!

---

Als zweites soll ein Codewort der Form "OXXXXXX" übermittelt werden (X steht für einen beliebigen Wert). Der zweite Kippschalter von links bei den SWITCHED FAULTS am PCM-Sender setzt ohne Rücksicht auf den tatsächlichen Wert das MSB stets auf 1 und kann so zur gezielten Erzeugung von Fehlern benutzt werden. Man beobachte und beschreibe die Wirkung der Parity-Bildung und Fehlererkennung. Weshalb ist keine Fehlerkorrektur möglich? Man überprüfe nun die Auswirkung der Fehlfunktion beim MSB auf das Ausgangssignal, indem anstatt einer Gleichspannung das 1 kHz-Sinussignal verwendet werde.

### **Aufgabe 15**

Der (7,4)-Hamming Code kann für die Korrektur von Bitfehlern verwendet werden (ERROR CHECK CODE SELECTOR auf "11"). Man überprüfe die in Aufgabe 3 erhaltenen Resultate auf experimentellem Wege. Zur Einstellung bestimmter Eingangswerte eignet sich die mit den Eingängen CH.0 und CH.1 verbundene Gleichspannungsquelle DC1. Was geschieht beim Einfügen von Fehlern mit dem zweiten SWITCHED FAULTS-Kippschalter?

Dasselbe ist mit der 1 kHz-Sinusspannung am Eingang zu wiederholen. Auf dem KO sollen das Eingangs- und das Ausgangssignal des PCM-Systems miteinander verglichen werden.

# Anhang A

## Ausgewählte Systemaspekte

Die beiden Abschnitte des Anhangs haben die Diskussion implementationsspezifischer Aspekte zum Inhalt. Nach einer Diskussion des im vorliegenden PCM-Versuchssystem verwendeten Rahmensynchronisationsverfahrens folgt eine knappe Erläuterung der Grundlagen fehlerkorrigierender Block-Codes.

### A.1 Rahmensynchronisation im vorliegenden Versuchssystem

In Anlehnung an die in Abbildung 3.6 dargestellte Rahmenstruktur wurde in Abschnitt 3.10 das Format des im PCM-Experimentalsystem verwendeten Datenrahmens kurz skizziert: die Präambel besteht aus einem einzigen Bit, gefolgt von 7 Datenbits für den ersten und weiteren 7 Datenbits für den zweiten Kanal, was eine Rahmenlänge von insgesamt 15 Bits ergibt. Im Folgenden ist kurz beschrieben, auf welche Weise der Empfänger die Rahmensynchronisation erreichen und bewahren kann.

Beim Synchronisationssymbol zu Beginn des Datenrahmens handelt es sich nicht etwa um einen Impuls von stets gleichbleibendem Wert. Vielmehr kann dieses Bit als *eigenständiger Synchronisationskanal* betrachtet werden, der - sozusagen gleichwertig zu den beiden Datenkanälen - im Zeitmultiplex übertragen wird. Das übertragene Synchronisationssignal entspricht einer periodisch sich wiederholenden, pseudozufälligen Sequenz mit einer Periode von 15 Symbolen, die nachstehend beschrieben wird. Da also pro Datenrahmen nur ein Synchronisationsbit zur Verfügung gestellt wurde, wiederholt sich der Beginn der Bitsequenz im Synchronisationskanal nach jeweils 15 Datenrahmen. Es sei hier betont, dass die Periode der Synchronisationssequenz nichts mit der Länge des Datenrahmens zu tun hat.

In Abbildung A.1 (a) ist dargestellt, wie eine sogenannte *pseudozufällige Sequenz* oder *M-Sequenz* (Maximallängensequenz) mittels eines linear rückgekoppelten Schieberegisters erzeugt wird. Ausgehend von einem beliebigen Anfangszustand des Schieberegisters (zum Beispiel "1111") ergibt sich bei der richtigen Platzierung der Abgriffe eine Sequenz von Nullen und Einsen, die eine maximale Länge von 15 Symbolen aufweisen kann und sich periodisch wiederholt. Im allgemeinen Fall erhält man bei einem Schieberegister mit  $m$  Speicherzellen bestenfalls Sequenzen der maximalen Länge  $2^m - 1$ ; der Grund liegt darin, dass das Schieberegister maximal  $2^m$  Zustände annehmen kann, wobei der Zustand "0000" zur trivialen Nullsequenz führen würde und daher nicht erlaubt ist. Tabelle A.1 enthält der Vollständigkeit halber die Zuordnungen bei der EXOR-Verknüpfung.

Die in Abbildung A.1 (b) enthaltene Schaltung des Synchronisationscode-Detektors funktioniert

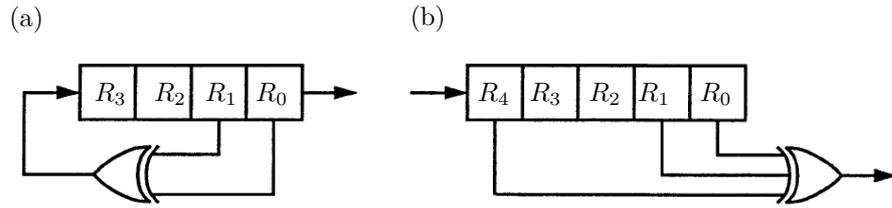


Abbildung A.1: Baugruppen der Rahmensynchronisation. (a) M-Sequenzgenerator im PCM-Sender. (b) Synchronisationscode-Detektor im PCM-Empfänger.

Tabelle A.1: EXOR-Verknüpfung  $X = A \oplus B$

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

ähnlich wie jene des Codegenerators in Abbildung A.1 (a); wie zuvor werden die Abgriffe bei den Registern  $R_0$  und  $R_1$  angebracht und EXOR-verknüpft. Nun muss jedoch zusätzlich der Inhalt des Registers  $R_4$  mit dem Resultat der vorherigen EXOR-Verknüpfung verglichen werden: falls  $R_0 \oplus R_1$  null ergibt und  $R_4$  ebenfalls null beträgt, entsteht kein Fehler; dasselbe gilt für  $R_0 \oplus R_1 = 1$  und  $R_4 = 1$ . Offenbar entspricht dieser Test auf Richtigkeit ebenfalls einer EXOR-Verknüpfung, und wegen der Assoziativität dieser Operation zeigt die EXOR-Schaltung mit drei Eingängen in Abbildung A.1 (b) mit einer logischen Eins an ihrem Ausgang an, ob ein Fehler aufgetreten ist oder nicht.

Der *Akquisitionsalgorithmus für die Synchronisation* arbeitet nun wie folgt. Ohne auf den Ausgang des Synchronisationscode-Detektors zu achten, werden zunächst 75 Bits des empfangenen Datenstroms eingelesen, was der Länge von fünf Datenrahmen entspricht; der Beginn eines Datenrahmens ist aber noch nicht bekannt. Jedes fünfzehnte Bit wird nun in das Schieberegister des Synchronisationscode-Detektors eingelesen. Sodann wird geprüft, ob es sich dabei um die richtige Synchronisationssequenz handelt. Zu diesem Zeitpunkt sind also die Register  $R_0$  bis  $R_3$  stets in einem erlaubten Zustand, ausser wenn es sich um "0000" handelt. Falls nun die Speicherzelle  $R_4$  einen erlaubten Wert enthält, wird kein Fehler angezeigt, und weitere 15 Bits können gelesen werden, wobei das erste wiederum nach  $R_4$  kopiert und auf seine Richtigkeit hin überprüft wird. Dieser Vorgang wird ständig wiederholt und erst abgebrochen, wenn der Synchronisationscode-Detektor an seinem Ausgang einen Fehler anzeigt (oder lediglich lauter Nullen eingelesen werden). Nach dem Abbruch des Suchvorgangs nimmt der Empfänger an, die Rahmengrenze beginne ein Bit später, überspringt ein Symbol im Datenstrom und wiederholt den eben beschriebenen Prüfvorgang: Einlesen von 5 Bits ins Schieberegister (entspricht der Dauer von 5 Datenrahmen) zwecks Herstellung eines Anfangszustands und sukzessive Überprüfung des (angenommenen) Synchronisationskanals auf seine Richtigkeit hin.

Es stellt sich die Frage nach der mittleren Dauer des Akquisitionsvorgangs. Nimmt man an, dass

keine Synchronisationssequenz gelesen wird, sondern lediglich Datenbits, deren Werte gleichverteilt 0 und 1 betragen, dann besteht eine gewisse Wahrscheinlichkeit, dass ein Fehler nicht schon nach fünf Datenrahmen, sondern erst etwas später detektiert wird. Die Wahrscheinlichkeit, dass  $j$  Bits hintereinander zufällig einen Ausschnitt der verwendeten M-Sequenz ergeben, beträgt  $2^{-j}$ . Die mittlere Anzahl überprüfter Synchronisationsbits (oder Datenrahmen) zwischen zwei Abbrüchen des Prüfvorgangs errechnet man zu

$$5 + \sum_{j=1}^{\infty} j2^{-j} = 5 + a \sum_{j=0}^{\infty} ja^{j-1} \Big|_{a=\frac{1}{2}} = 5 + a \frac{\partial}{\partial a} \sum_{j=0}^{\infty} a^j \Big|_{a=\frac{1}{2}} = 5 + \frac{a}{(1-a)^2} \Big|_{a=\frac{1}{2}} = 7, \quad (\text{A.1})$$

wobei die fünf zusätzlichen Bits für die Herstellung des Anfangszustands der Register  $R_4$  bis  $R_0$  benützt werden müssen.

Da man annehmen kann, dass der Akquisitionsvorgang des Empfängers mit gleicher Wahrscheinlichkeit bei irgendeinem der  $p = 15$  Bits des Datenrahmens beginnt, beträgt die mittlere Anzahl Wiederholungen des oben analysierten Prüfvorgangs

$$\frac{1}{p} \sum_{j=1}^p j = \frac{1}{p} \frac{p(p+1)}{2} = \frac{p+1}{2} = 8. \quad (\text{A.2})$$

Nach dieser Abschätzung ist also die Rahmensynchronisation im Mittel nach der Untersuchung von  $8 \cdot 7 = 56$  Datenrahmen gewährleistet.

Der offenkundige Nachteil dieser Synchronisationsmethode ist wohl die relativ lange Akquisitionszeit. Als Vorteil zu erwähnen ist jedoch der sparsame Umgang mit den Ressourcen: pro Datenrahmen wird lediglich ein einziges Bit für Synchronisationszwecke benötigt.

## A.2 Fehlerkorrigierende Block-Codes

In diesem Abschnitt sind die Grundlagen der Beschreibung linearer Block-Codes in aller Kürze zusammengefasst; danach wird die Syndrom-Decodierung anhand eines sehr einfachen Beispiels erläutert. Eine ausführliche Behandlung der Theorie und Anwendung fehlerkorrigierender Codes ist in [2–4] gegeben.

Bei einem  $(n, k)$ -Block-Code werden  $k$  Datensymbole zu sogenannten *Datenblocks*  $b_{k-1}b_{k-2} \dots b_1b_0$ , zusammengefasst und in  $n$ -elementige *Codeworte*  $c_{n-1}c_{n-2} \dots c_1c_0$  abgebildet. Diese können auch mittels der Zeilenvektoren

$$\mathbf{b} = [b_{k-1} \ b_{k-2} \ \dots \ b_1 \ b_0] \quad \text{und} \quad (\text{A.3})$$

$$\mathbf{c} = [c_{n-1} \ c_{n-2} \ \dots \ c_1 \ c_0] \quad (\text{A.4})$$

dargestellt werden. Dabei heisst ein Code *linear*, wenn jede Summe zweier beliebiger Codeworte  $\mathbf{c}_i + \mathbf{c}_j$  ebenfalls ein Codewort ergibt. Die *Addition zweier Vektorelemente* sei definiert als deren logische EXOR-Verknüpfung. Ein linearer Code lässt sich demnach mit Hilfe der Abbildungsvorschrift

$$\mathbf{c} = \mathbf{bG} \quad (\text{A.5})$$

vollständig beschreiben, wobei die Koeffizientenmatrix  $\mathbf{G}$  mit  $k$  Zeilen und  $n$  Spalten als *Erzeugende* bezeichnet wird. Man beachte die Linksmultiplikation von  $\mathbf{G}$  mit dem Zeilenvektor  $\mathbf{b}$ ; diese Schreibweise ist in der Codierungstheorie allgemein üblich. Für die *Multiplikation zweier Matrixelemente* wird deren *logische AND-Verknüpfung* gebildet.

---

Der Code heisst systematisch, falls die ersten  $k$  Stellen des Codevektors  $\mathbf{c}$  mit dem Datenvektor  $\mathbf{b}$  identisch sind; die letzten  $n - k$  Stellen von  $\mathbf{c}$  heissen in einem solchen Fall Paritätsbits und können als entsprechender Paritätsvektor  $\mathbf{p} = [p_{n-k-1} \ p_{n-k-2} \ \dots \ p_1 \ p_0]$  angeschrieben werden. Die Codeworte eines systematischen Codes lauten in symbolischer Schreibweise

$$\mathbf{c} = [\mathbf{b} \ \mathbf{p}]. \quad (\text{A.6})$$

Wegen der Linearitätseigenschaft kann durch geeignete Linearkombination der Zeilen der Erzeugenden  $\mathbf{G}$  jeder lineare nichtsystematische Code in seine systematische Form gebracht werden. Ohne Einschränkung der Allgemeinheit können sich also die Betrachtungen auf Codes mit Erzeugenden der Form

$$\mathbf{G} = [\mathbf{I}_k \ \mathbf{P}] \quad (\text{A.7})$$

beschränken, wobei  $\mathbf{P}$  die Dimensionen  $k \times (n - k)$  hat und  $\mathbf{I}_k$ , die  $k \times k$ -Einheitsmatrix darstellt:

$$\mathbf{I}_k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \quad (\text{A.8})$$

Mit der Definition  $\mathbf{p} = \mathbf{bP}$  folgt Gleichung (A.6) unmittelbar aus (A.5).

Die Transponierte  $\mathbf{P}^T$  von  $\mathbf{P}$  wird zur Bildung der sogenannten  $(n - k) \times n$  - *Parity-Check-Matrix*  $\mathbf{H}$  herangezogen:

$$\mathbf{H} = [\mathbf{P}^T \ \mathbf{I}_{n-k}]. \quad (\text{A.9})$$

Durch Ausmultiplizieren der Submatrizen in (A.7) und (A.9) ergibt sich die wichtige Beziehung

$$\mathbf{GH}^T = [\mathbf{I}_k \ \mathbf{P}] \begin{bmatrix} \mathbf{P} \\ \mathbf{I}_{n-k} \end{bmatrix} = \mathbf{P} + \mathbf{P} = \mathbf{0}. \quad (\text{A.10})$$

Dabei sei noch einmal daran erinnert, dass die Multiplikation der Matrixelemente der logischen AND-Operation, die Addition hingegen als EXOR-Verknüpfung definiert ist:  $p_{ij} + p_{ij} = 0$  für  $p_{ij} = 0$  oder  $p_{ij} = 1$ . Aus Gleichung (A.10) ergibt sich nun mit (A.5)

$$\mathbf{cH}^T = \mathbf{bGH}^T = \mathbf{b0} = \mathbf{0} \quad (\text{A.11})$$

die Rechtsmultiplikation eines beliebigen Codevektors  $\mathbf{c}$  mit  $\mathbf{H}^T$  ergibt also stets einen Nullvektor  $\mathbf{0}$ .

Als Beispiel betrachte man den einfachen *Repeat-Code* mit der Erzeugenden

$$\mathbf{G} = [\mathbf{I}_k \ \mathbf{P}] = [1 \ 1 \ 1]. \quad (\text{A.12})$$

Für eine Null werden demnach drei Nullen, für eine Eins drei Einsen auf den Übertragungskanal gegeben:  $[0] \rightarrow [0 \ 0 \ 0]$  und  $[1] \rightarrow [1 \ 1 \ 1]$ . Man erhält tatsächlich gemäss (A.12)

$$\mathbf{GH}^T = [\mathbf{I}_k \ \mathbf{P}] = [1 \ 1 \ 1] \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [0 \ 0] \quad (\text{A.13})$$

---

Die obige Rechnung beinhaltet ebenfalls das Ergebnis von (A.11), zumal  $[1\ 1\ 1]$  in diesem einfachen Fall nicht nur der Erzeugenden  $\mathbf{G}$ , sondern auch einem der beiden Codevektoren entspricht (die Rechnung mit  $[0\ 0\ 0]$  ergibt trivialerweise Null).

Um das Vorgehen bei der Decodierung zu verstehen, betrachten wir im Folgenden Operationen an Blöcken  $y$  verrauschter Kanalsymbole der Länge  $n$ , welche nicht zur Menge der Codevektoren gehören. Unter der Annahme, dass über jedes einzelne Bit nach dessen Empfang eine Entscheidung getroffen wird, ob es sich um eine logische Eins oder eine logische Null handelt, erhält man den *demodulierten Vektor*

$$\mathbf{y} = \mathbf{c} + \mathbf{e}. \quad (\text{A.14})$$

Diejenigen Symbole in  $\mathbf{y}$ , welche wegen des Rauschens auf der Übertragungsstrecke nicht mit jenen in  $\mathbf{c}$  identisch sind, können durch Einsen an den entsprechenden Stellen im *Code-Fehlervektor*  $\mathbf{e}$  bezeichnet werden. Laut Tabelle 1 ergibt die Verknüpfung eines beliebigen Elements (0 oder 1) mit einer Eins stets das inverse Element (1 bzw. 0).

Multipliziert man analog zu (A.11)  $\mathbf{y}$  anstelle von  $\mathbf{c}$  mit  $\mathbf{H}^T$ , so erhält man den sogenannten *Syndromvektor*  $\mathbf{s}$  der Länge  $n - k$ :

$$\begin{aligned} \mathbf{s} &= \mathbf{y}\mathbf{H}^T \\ &= (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{0} + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T. \end{aligned} \quad (\text{A.15})$$

Das obige Ergebnis hängt also gar nicht vom übermittelten Codewort, sondern ausschliesslich vom Fehlervektor ab. Die Berechnung des Syndromvektors der Länge  $n - k$  bildet im Decoder für den fehlerkorrigierenden Code den ersten Schritt zur Ermittlung des wahrscheinlichsten Datenvektors  $\hat{\mathbf{b}}$  aus dem empfangenen und verrauschten Vektor  $\mathbf{y}$ .

Bevor wir uns wieder einem einfachen Beispiel zuwenden, ist das Verständnis einer weiteren Eigenschaft notwendig:

Den insgesamt  $2^k$  möglichen Datenvektoren  $\mathbf{b}_i$  ( $i = 0, \dots, 2^k - 1$ ) sind ebenso viele Codevektoren  $\mathbf{c}_i$  ( $i = 0, \dots, 2^k - 1$ ) zugeordnet. Für ein bestimmtes Fehlermuster  $\mathbf{e}$  seien angesichts der Übertragung irgendeines Codevektors die entsprechenden demodulierten Vektoren

$$\mathbf{y}_i = \mathbf{e} + \mathbf{c}_i, \quad i = 0, \dots, 2^k - 1 \quad (\text{A.16})$$

definiert. Die so definierte Menge  $C_{\mathbf{e}}$  der Vektoren  $\mathbf{y}_i \in C_{\mathbf{e}}$  besteht aus  $2^k$  Elementen, was zur Folge hat, dass es insgesamt  $2^{n-k}$  disjunkte Mengen  $C_{\mathbf{e}}$  in einem  $(n,k)$ -Code gibt. Auf jeden Fall erhält man nach Multiplikation beider Seiten von (A.16) mit  $\mathbf{H}^T$

$$\mathbf{y}_i\mathbf{H}^T = \mathbf{e}\mathbf{H}^T + \mathbf{c}_i\mathbf{H}^T = \mathbf{e}\mathbf{H}^T, \quad (\text{A.17})$$

was wiederum unabhängig vom Codevektor  $\mathbf{c}_i$  ist. Man kann also sagen, dass jede einzelne der Mengen  $C_{\mathbf{e}}$  gemäss (A.16) durch einen bestimmten Syndromvektor  $\mathbf{s}$  der Länge  $n - k$  charakterisiert ist. Der Unterschied zwischen zwei Elementen  $\mathbf{y}_i$  und  $\mathbf{y}_j$  beträgt  $\mathbf{c}_i + \mathbf{c}_j$ , was wegen der Linearität des Codes wiederum ein Codewort ist; zwei Elemente  $\mathbf{y}_i$  und  $\mathbf{y}_j$  unterscheiden sich demnach genau um ein Codewort.

Diese Eigenschaft macht man sich bei der Decodierung eines  $(n,k)$ -Block-Codes zunutze. Der Syndromvektor liefert offenbar gewisse Angaben über den Fehlervektor  $\mathbf{e}$ , die jedoch nicht ausreichen, um den exakten Wert von  $\mathbf{e}$  zu ermitteln. Immerhin reduziert die Kenntnis von  $\mathbf{s}$  die Suche nach einem von  $2^n$  möglichen Fehlervektoren  $\mathbf{e}$  auf die Auswahl einer von  $2^k$  möglichen Alternativen, nämlich auf eines der Elemente der zu  $\mathbf{s}$  gehörenden Menge  $C_{\mathbf{e}}$ . Es ist die Aufgabe des Decoders,

---

die beste Auswahl aus der durch  $\mathbf{s}$  bestimmten Menge  $C_e$  zu treffen. Da es hier gemäss (A.16) um die Auswahl eines der  $2^k$  Codeworte  $\mathbf{c}_i$  und demzufolge um einen der  $2^k$  Datenvektoren  $\mathbf{b}_i$  geht, erhält man auf diese Weise die ursprünglichen Daten zurück.

Anhand eines einfachen Beispiels sei dieses Vorgehen illustriert: die Decodierung soll für den durch (A.12) definierten *Repeat Code* durchgeführt werden. Mit der Parity-Check-Matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (\text{A.18})$$

erhält man den Syndromvektor  $\mathbf{s} = [s_1 \ s_0]$  zu

$$\mathbf{s} = \mathbf{e}\mathbf{H}^T = [e_2 \ e_1 \ e_0] \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (\text{A.19})$$

Nachstehend folgt die Tabelle der Syndromvektoren in Abhängigkeit der möglichen Fehlermuster:

$e_2 \ e_1 \ e_0$	$s_1 \ s_0$
0 0 0	0 0
0 0 1	0 1
0 1 0	1 0
0 1 1	1 1
1 0 0	1 1
1 0 1	1 0
1 1 0	0 1
1 1 1	0 0

Ordnet man nun die obige Aufstellung nach den Syndromvektoren  $\mathbf{s}$ , so ergeben sich die entsprechenden Mengen  $C_e$  aus denen anschliessend eines der Elemente als wahrscheinlichstes Fehlermuster bestimmt werden muss:

---

<b>s</b>	<b>e</b>
0 0	0 0 0
	1 1 1
0 1	0 0 1
	1 1 0
1 0	0 1 0
	1 0 1
1 1	1 0 0
	0 1 1

Betrachtet man nun eine beliebige der vier Teilmengen in der rechten Spalte, so ist erkennbar, dass alle Elemente in einer Teilmenge durch Addition der Codevektoren  $\mathbf{c}_0 = [0\ 0\ 0]$  und  $\mathbf{c}_1 = [1\ 1\ 1]$  zu einem beliebigen Element der jeweiligen Teilmenge erhalten werden können, was mit (A.16) und (A.17) übereinstimmt. Anstatt eines von  $2^n = 8$  Fehlermustern bestimmen zu müssen, erhält man durch die Operation (A.19) einen von  $2^{n-k} = 4$  Syndromvektoren. Diesem sind jeweils  $2^k = 2$  mögliche Fehlermuster zugeordnet, von denen das wahrscheinlichere ausgewählt wird. Der geschätzte Fehlervektor erhält die Bezeichnung  $\hat{\mathbf{e}}(s)$ .

Nimmt man nun an, dass kein oder nur ein Übertragungsfehler wahrscheinlicher sind als zwei oder drei, erhält man die endgültige Zuordnung, die der Decoder vornehmen muss:

<b>s</b>	<b><math>\hat{\mathbf{e}}(s)</math></b>
0 0	0 0 0
0 1	0 0 1
1 0	0 1 0
1 1	1 0 0

Zusammenfassend lautet der Decodieralgorithmus wie folgt:

1. Nach Erhalt des demodulierten Vektors  $\mathbf{y} = \mathbf{c} + \mathbf{e}$  berechne man den Syndromvektor  $\mathbf{s} = \mathbf{y}\mathbf{H}^T$ , welcher eine der  $2^{n-k}$  Mengen  $C_{\mathbf{e}}$  bestimmt.
2. Aus der durch den Syndromvektor  $\mathbf{s}$  bestimmten Menge  $C_{\mathbf{e}}$  ist das wahrscheinlichste Fehlermuster  $\hat{\mathbf{e}}(s)$  zu wählen.
3. Man erhält eine Schätzung für das übertragende Codewort mittels  $\hat{\mathbf{c}} = \mathbf{y} + \hat{\mathbf{e}}(s)$  wegen der Beziehung  $\mathbf{y} = \mathbf{c} + \mathbf{e}$ .
4. Wegen der eindeutigen Zuordnung von Codeworten zu Datenblöcken erfolgt zuletzt die Bestimmung des geschätzten Datenblocks  $\hat{\mathbf{b}}$  aus  $\hat{\mathbf{c}}$ . Bei einem systematischen Code besteht einfach  $\hat{\mathbf{b}}$  aus den ersten  $k$  Komponenten von  $\hat{\mathbf{c}}$ .

Da es  $2^{n-k}$ -mal so viele mögliche Vektoren  $\mathbf{y}$  am Decodereingang geben kann, als Codevektoren  $\mathbf{c}$  vorhanden sind, führt das obige Vorgehen in jedem Fall zur Bestimmung eines Codeworts. Jedem

---

der  $2^{n-k}$  Syndromvektoren wird jeweils einer der  $2^{n-k}$  wahrscheinlichsten aus  $2^n$  möglichen Fehlern zugeordnet. Hierbei besteht jedoch die Möglichkeit eines *Decodierungsfehlers*; dieser tritt nämlich dünn auf, falls ein von den festgelegten  $\hat{\mathbf{e}}(\mathbf{s})$  abweichender Fehlervektor den tatsächlichen Fehler darstellt.

Man betrachte den Fall  $\mathbf{e} = [0 \ 1 \ 1]$ , wobei  $\mathbf{b} = [1]$  und dementsprechend  $\mathbf{c} = [1 \ 1 \ 1]$ . Am Decodiereingang ergibt sich  $\mathbf{y} = \mathbf{c} + \mathbf{e} = [1 \ 0 \ 0]$  und daraus der Syndromvektor  $\mathbf{s} = [1 \ 1]$ , der zur Schätzung des Fehlers  $\hat{\mathbf{e}}(\mathbf{s}) = [1 \ 0 \ 0]$  Anlass gibt. Die Schätzung unterscheidet sich jedoch vom tatsächlichen Fehlervektor  $\mathbf{e}$  gemäss (A.16) um ein Codewort, und man erhält die Schätzung für das übertragene Codewort  $\hat{\mathbf{c}} = \mathbf{y} + \hat{\mathbf{e}}(\mathbf{s}) = [0 \ 0 \ 0]$ , woraus  $\hat{\mathbf{b}} = [0]$  folgt.

# Literaturverzeichnis

- [1] S. Haykin, *Communication Systems*, 4th ed. John Wiley & Sons, 2001.
- [2] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill, 2008.
- [3] A. Lapidoth, “Information Theory I + II,” *Vorlesung an der ETH Zürich, D-ITET*.
- [4] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, 1983.