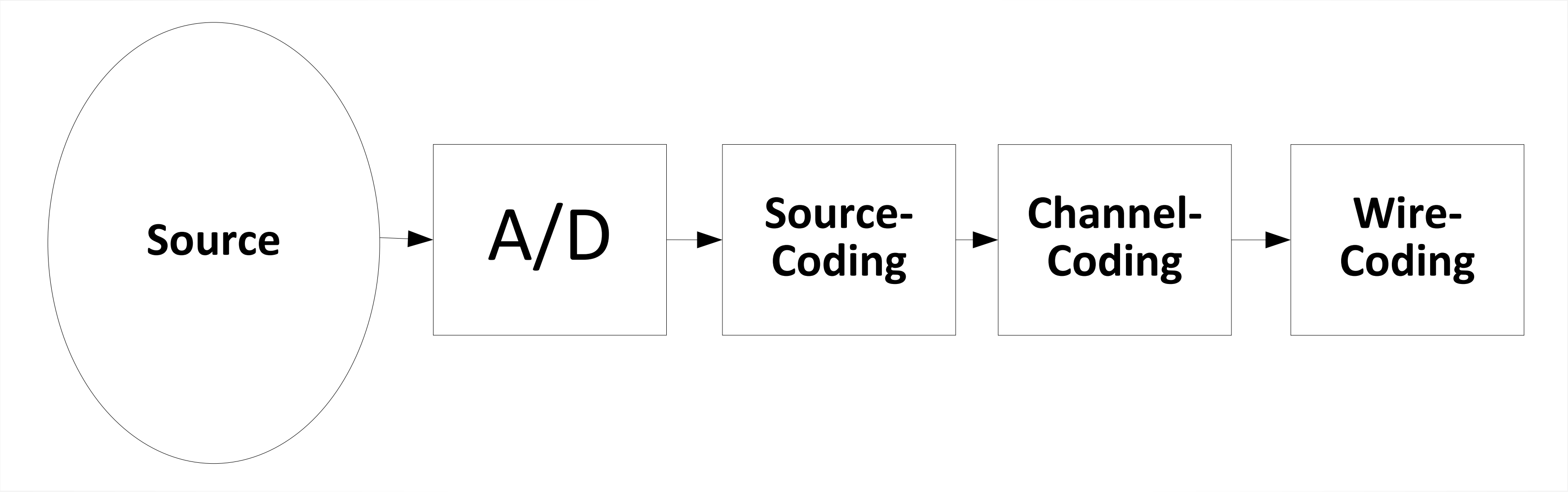
Netztechnik-Vorlesung Teil-3

Netztechnik-Vorlesung Teil-3
Inhalt

- Coding (Source-Coding / Channel-Coding / Wire-Coding)
- Fehler
- Modulation

Codierung



Source Coding

Source Coding dient zur Reduzierung der zu transportierenden Datenmenge.
Eine Reduzierung von Speicherplatz ist eng verbunden mit weniger zu übertragenden Daten und damit eine Kostenfrage.
Andererseits ist die Codierung ein komplexes, aufwändiges Verfahren, das einen erheblichen technischen Aufwand bedeutet.

Beispiel:
Eine Datenquelle kennt 4 Symbole (A, B, C und D)
Die Wahrscheinlichkeit des Auftretens der Symbole ist:

P(A) = 0,5
P(B) = 0,25
P(C) = 0,125
P(D) = 0,125

Damit ist der durchschnittliche Informationsgehalt (die Entropie) H = 1,75 bit / Symbol

H = 0,5*ld(1/0,5) + 0,25*ld(1/0,25) + 2* (0,125*ld(1/0,125)) = 0,5+0,5+2*0,375 = 1,75bit/Symbol

Source Coding:
Beispiel

```
..0         1           2
..123456789012345678901234..
..AAACACABABBABADAACBDAABD..
```

Eine erste, einfache Codierung könnte so aussehen:
A = 00
B = 01
C = 10
D = 11

Die obige Nachricht mit 24 Symbolen hat einen Informationsinhalt von 24 * 1,75 = 42 Bit.
Mit dem obigen Code erhält man jedoch 2 * 24 = 48 Bit. Dies bedeutet, dass 6 Bit redundant sind.

Ein besseres Ergebnis bringt eine weitere Codierung
A = 1
B = 01
C = 001
D = 000

Damit werden 42 Bit benötigt. Dies entspricht dem Informationsgehalt.
Damit ist die Redundanz eliminiert.

Fano Bedingung:
Wenn kein Codewort existiert, das der Anfang eines anderen Codewortes ist, dann ist jeder Satz von Codewörtern korrekt.

Source Coding: Beispiel 2 (Morse-Code)

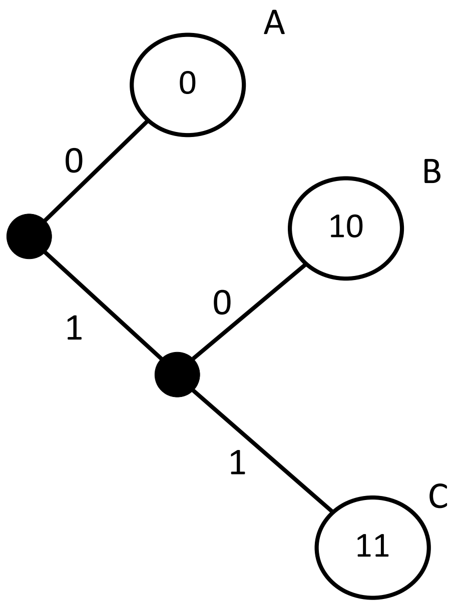
Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
a	.-	l	.-..	w	.--	8	---..
b	-...	m	--	x	-..-	9	----.
c	-.-.	n	-.	y	-.--	0	-----
d	-..	o	---	z	--..	Punkt	.-.-.-
e	.	p	.--.	1	.----	Komma	--..--
f	..-.	q	--.-	2	..---	Fragezeichen	..--..
g	--.	r	.-.	3	...--	Doppelpunkt	---...
h	....	s	...	4	....-	Semikolon	-.-.-.
i	..	t	-	5	.....	Trennung	-...-
j	.---	u	..-	6	-....	Bruchstrich	-..-.
k	-.-	v	...-	7	--...	Anführungszeichen	.-..-.

Source Coding: Fano-Bedingung

Fano Bedingung:
Wenn kein Codewort existiert, das der Anfang eines anderen Codewortes ist, dann ist jeder Satz von Codewörtern präfixfrei und damit korrekt.

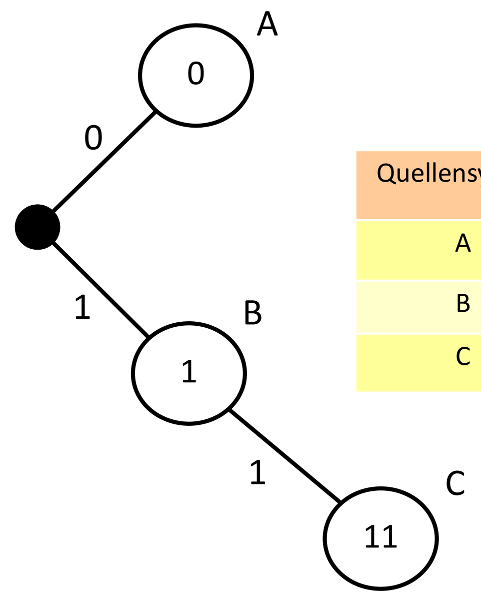
Ein D-närer päfixfreier Code mit den Codewortlängen (w1, w2, w3, … , wL) existiert genau dann, wenn die Kraft-Millan-Ungleichung erfüllt wird:

$$\sum_{i=1}^{L} D^{-wi} \leqslant 1$$



Quellensymbol	Codewort
A	0
B	10
C	11

Codebaum präfixfrei



Quellensymbol	Codewort
A	0
B	1
C	11

Codebaum nicht präfixfrei

Source Coding:
Huffman-Codierung

Der Huffman-Code erfüllt die Fano-Bedingung und wird bei Codierungsalgorithmen wie JPEG verwendet.
Um einen Text zu codieren sind folgende Schritte zu durchlaufen:

1.
Die Symbole einer gegebenen Quelle (z. B. ASCII-Text) werden nach der Häufigkeit ihres Auftretens in einer Tabelle angeordnet.

2.
Die Symbole mit der niedrigsten Häufigkeit werden mit 0 und 1 codiert und in der Tabelle gekennzeichnet.

3.
Die beiden Symbole werden zu einem neuen Symbol zusammengefasst und die Wahrscheinlichkeit ihres Auftretens wird addiert. Da die beiden Symbole zusammengefasst wurden, ist die Tabelle um eine Zeile kürzer. Die Wahrscheinlichkeiten werden in der Tabelle berücksichtigt indem sie neu einsortiert werden. Die Schritte 2 und 3 werden so lange wiederholt, bis nur noch zwei Zeilen übrig sind.

4.
Man nimmt die letzte Tabelle und geht die Tabellen rückwärts bis zur ersten Tabelle durch und druckt den Code-Baum aus. Mit jeder Tabelle bekommt man eine Code-Entscheidung und zwei Zweige des Code Baumes. Man liest jetzt den Code-Baum vom Start bis zu jedem Knoten und erhält den Code für jedes Symbol.

Source-Coding:
Huffman-Codierung mit Tabellen-Beispiel 1

Es gibt 10 Symbole (x1, x2, .. x10) mit den folgenden Wahrscheinlichkeiten (P) ihres Auftretens

Symbol	:
x1	0,25
x2	0,15
X3	0,2
X4	0,2
X5	0,05
X6	0,07
X7	0,025
X8	0,02
X9	0,025
X10	0,01

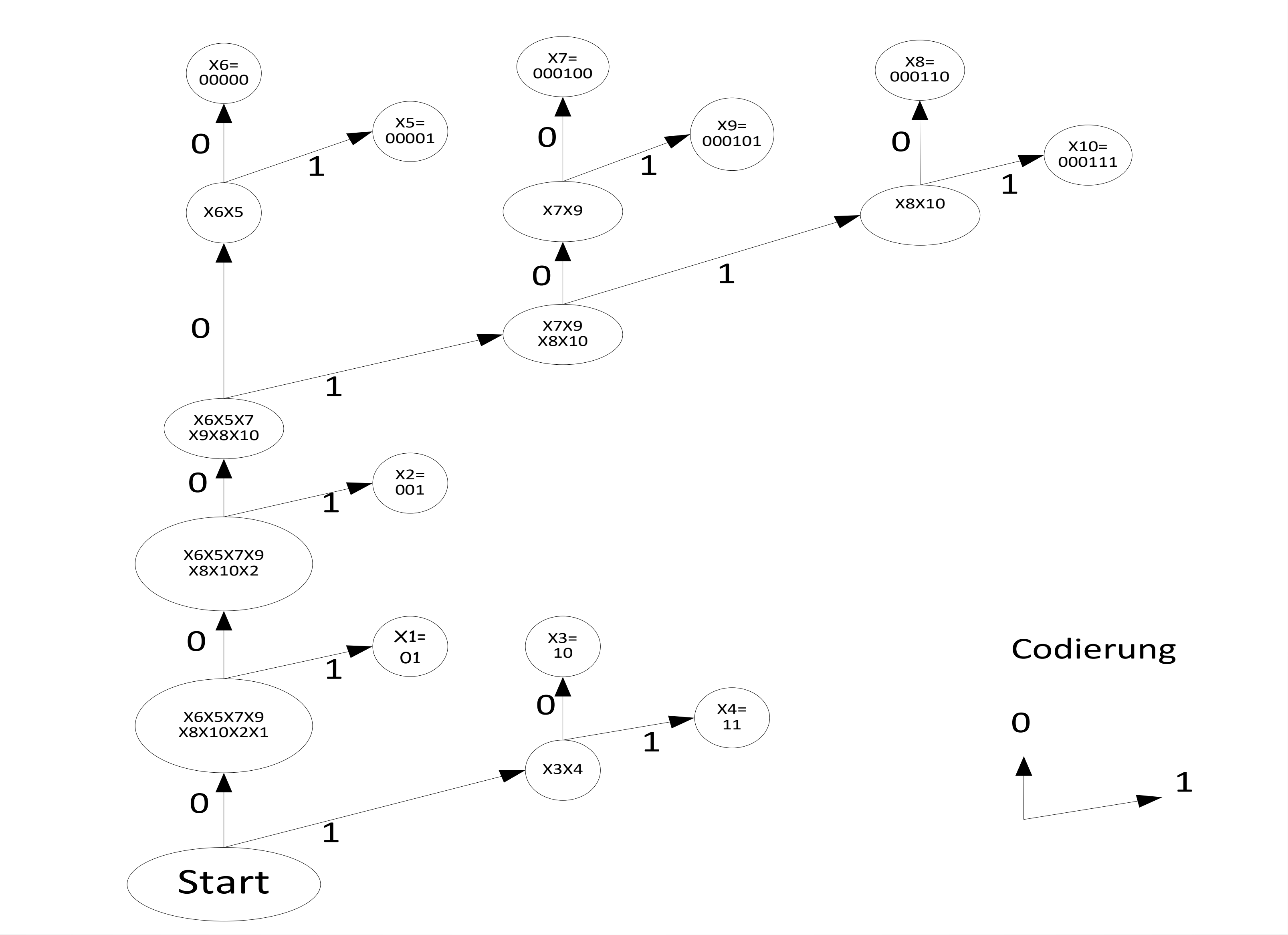
Source-Coding:
Huffman-Codierung mit Tabellen-Beispiel Teil 2

Pos	Tabelle 1	Tabelle 2	Tabelle 3	Tabelle 4	Tabelle 5
1	X1 = 0,25	X1 = 0,25	X1 = 0,25	X1 = 0,25	X1 = 0,25
2	X3 = 0,2	X3 = 0,2	X3 = 0,2	X3 = 0,2	X3 = 0,2
3	X4 = 0,2	X4 = 0,2	X4 = 0,2	X4 = 0,2	X4 = 0,2
4	X2 = 0,15	X2 = 0,15	X2 = 0,15	X2 = 0,15	X2 = 0,15
5	X6 = 0,07	X6 = 0,07	X6 = 0,07	X7X9X8X10 = 0,08	X6X5 = 0,12 -> 0
6	X5 = 0,05	X5 = 0,05	X5 = 0,05	X6 = 0,07 -> 0	X7X9X8X10 = 0,08 → 1
7	X7 = 0,025	X8X10 = 0,03	X7X9 = 0,05 -> 0	X5 = 0,05 → 1	
8	X9 = 0,025	X7 = 0,025 -> 0	X8X10 = 0,03 -> 1		
9	X8 = 0,02 -> 0	X9 = 0,025 → 1			
10	X10 = 0,01 → 1				

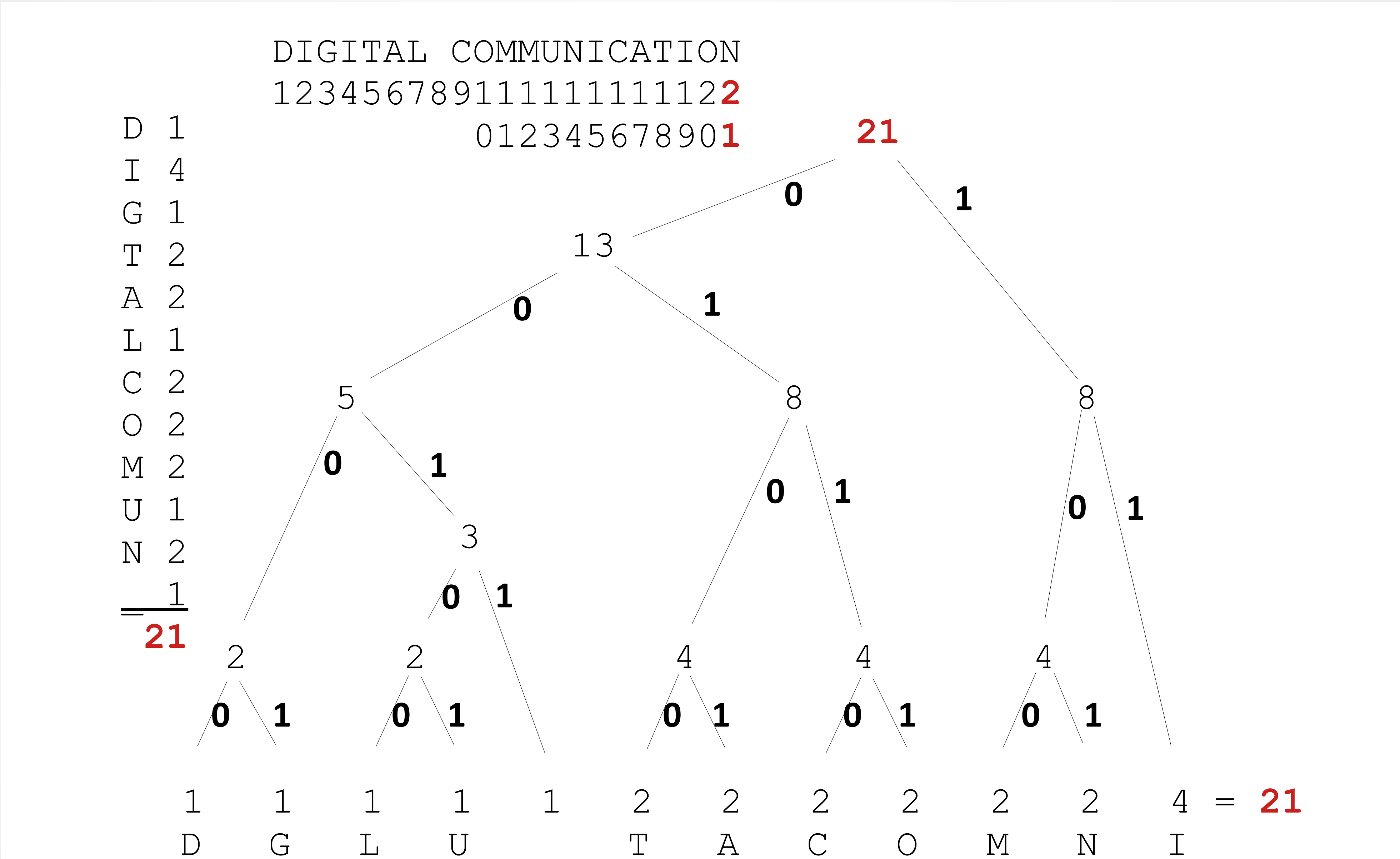
Source-Coding: Huffman-Codierung Tabellen-Beispiel Teil 3

Pos	Tabelle 6	Tabelle 7	Tabelle 8	Tabelle 9
1	X1 = 0,25	X6X5X7X9X8X10X2 = 0,35	X3X4 = 0,4	X6X5X7X9X8X10X2X1 = 0,6 → 0
2	X3 = 0,2	X1 = 0,25	X6X5X7X9X8X10X2 = 0,35 -> 0	X3X4 = 0,4 → 1
3	X4 = 0,2	X3 = 0,2 → 0	X1 = 0,25 -> 1	
4	X6X5X7X9X8X10 = 0,2 -> 0	X4 = 0,2 -> 1		
5	X2 = 0,15 → 1			

Source-Coding:
Huffman-Codierung Tabellen-Beispiel Teil 4



Source-Coding: Huffman-Codierung
Graphisches Beispiel: „DIGITAL COMMUNICATION“



Channel-Coding: Fehlererkennung mit Paritätsbits

Wort der Länge n	Gerade Parität	Ungerade Parität
	Wert des Paritätsbits	
Anzahl der Einsen ist gerade	0	1
Anzahl der Einsen ist ungerade	1	0

Channel-Coding:
Fehlererkennung mit Paritätsbits

Code ohne Parität		Code mit Parität	
Hexadezimal-Darstellung	Binär-Darstellung	Gerade Parität	Ungerade Parität
0	0000	00000	00001
1	0001	00011	00010
2	0010	00101	00100
3	0011	00110	00111
4	0100	01001	01000
5	0101	01010	01011
6	0110	01100	01101
7	0111	01111	01110
8	1000	10001	10000
9	1001	10010	10011
A	1010	10100	10101
B	1011	10111	10110
C	1100	11000	11001
D	1101	11011	11010
E	1110	11101	11100
F	1111	11110	11111

Channel-Coding:
Fehlererkennung (Hamming-Distanz)

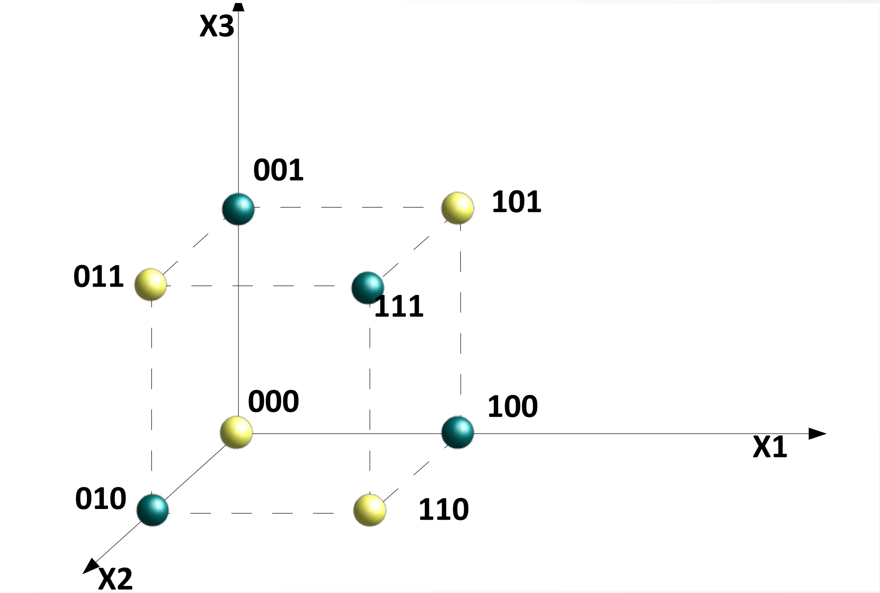
Die Distanz zweier Codewörter ist die Anzahl der Bits in denen sich die
beiden Codewörter unterscheiden.
So haben die Codewörter 0000 und 1111 die Distanz 4.
Die Codewörter 0000 und 0001 haben die Distanz 1.
Die Hamming-Distanz ist der minimale Abstand aller möglichen Codewörter eines Codes.

Allgemein gilt:
Die Fehlererkennungs- und -korrektureigenschaften eines Codes hängen
von seinem Hamming Abstand ab.

Zum Auffinden von d Fehlern benötigt man einen Hamming-Abstand von d + 1.
Zur Korrektur von d Fehlern braucht man einen Hamming-Abstand von 2d + 1.

Wird ein 2 Bit-Binärcode um ein Prüfbit erweitert,
entsteht folgende Zuordnung:
2 Bit-Binärcode + Parity-Bit -> 3 Bit-Binärcode

00	0	000	\	
01	1	011	|	gültige Codes
10	1	101	|	
11	0	110	/	
		001	\	
		010	|	ungültige Codes
		100	|	
		111	/	



Channel-Coding:
Fehlererkennung mit 2-dimensionaler Parität

Datenbits

Quer paritäts bits

0	1	0	0	1	1	0	1	0
0	0	0	0	1	0	0	1	0
1	0	0	0	1	0	1	1	0
1	1	1	0	1	1	0	0	1
0	0	1	1	0	1	1	1	1
0	0	0	1	0	1	0	0	0

Längsparitätsbits

0	1	0	0	1	1	0	1	0
0	0	1	0	1	0	1	1	0
0	0	0	0	1	0	1	1	0
1	1	1	0	1	1	1	0	1
0	0	1	1	0	1	1	1	1
0	0	0	1	0	1	0	0	0

4 Bitfehler erkennbar

0	1	0	0	1	1	0	1	0
0	0	1	0	1	0	1	1	0
1	0	0	0	1	0	1	1	0
1	1	0	0	1	1	1	0	1
0	0	1	1	0	1	1	1	1
0	0	0	1	0	1	0	0	0

4 Bitfehler nicht erkennbar

Channel-Coding: Fehlererkennung mit CRC (Cyclic Redundancy Check)

Beispiel

Wir haben eine 8-Bit-Nachricht M. Ihr wert sei 10011010.
Das entspricht folgendem Polynom.
$M(x)= 1*x^7 + 0*x^6 + 0*x^5 + 1*x^4 + 1*x^3 + 0*x^2 + 1*x^1 + 0*x^0 = x^7 + x^4 + x^3 + x^1$

Das Generatorpolynom G sei vom Grad 3.
$G(x) = x^3 + x^2 + x^0$ In diesem Beispiel sei G(x) = 1101.

Da das Generatorpolynom vom Grad 3 ist, wird M(x) zuerst mit x^3 multipliziert denn .
Dies entspricht einer Verschiebung um 3 Stellen nach links.
Damit wird aus :10011010 der Wert 10011010000.
Danach wird M(x) durch G(x) dividiert, was einer blockweisen XOR-Operation entspricht.

Original-Polynom 10011010

Multiplikation mit X³ 10011010000

Division durch Generatorpolynom

```
10011010000 / 1101   Generator-Polynom
1101
 1001
 1101
  1000
  1101
   1011
   1101
    1100
    1101
      1000
      1101
       101  -> Rest = 101
```

Rest = 101 (wird über die 3 eingeschobenen Stellen geschrieben)

Der Rest mit dem Wert 101 wird mit $M(x)*x^3$ XOR-verknüpft.

Das Ergebnis lautet 10011010101.
Dieser Wert wird als N(x) gesendet.

Übertragenes-Polynom 10011010101

Division durch Generatorpolynom

```
10011010101 / 1101   Generator-Polynom
1101
 1001
 1101
  1000
  1101
   1011
   1101
    1100
    1101
       1101
       1101
       0000
```

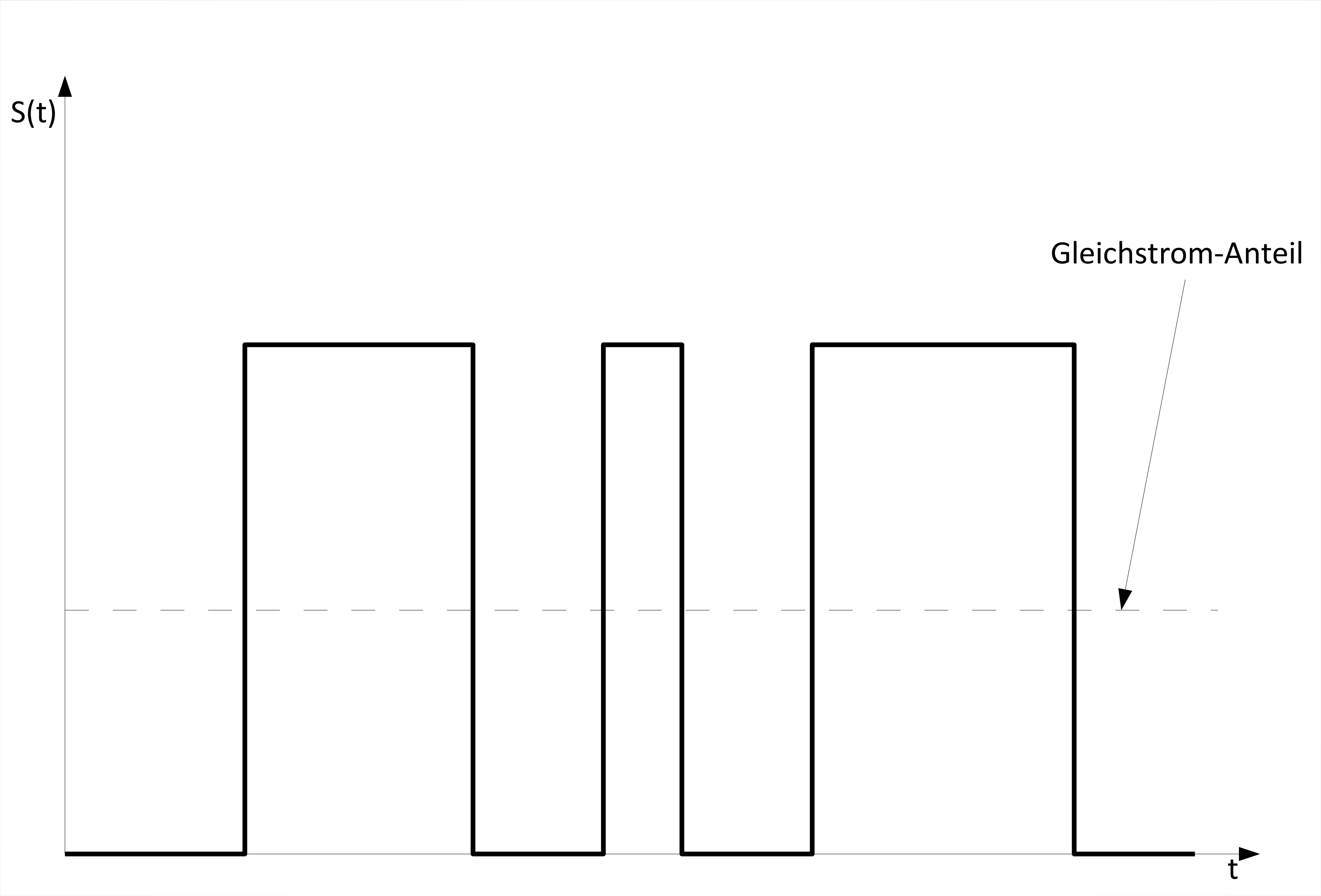
Rest = 0 !
-> Kein Fehler bei der Übertragung

Channel-Coding:
Fehlererkennung mit CRC (Cyclic Redundancy Check)

Übliche CRC-Generatorpolynome sind:

CRC	C(x) Generatorpolynom
CRC-8	$X^8 + x^2 + x^1 + 1$
CRC-10	$X^{10} + x^9 + x^5 + x^4 + x + 1$
CRC-12	$X^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$X^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$X^{16} + x^{12} + x^5 + 1$
CRC-32	$X^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Wire-Coding: Gleichstromanteil



Wire-Coding: Grundvoraussetzungen

Möglichst kein Gleichstromanteil denn dadurch muss mehr Leistung übertragen werden.

Dies führt z. B. zu einer Erwärmung von Kupferleitungen.

Eine einfache Takt-Rückgewinnung sollte möglich sein.

Es sollte eine hohe Effizienz und Stabilität erreicht werden.

Es sind unterschiedliche Codierungsverfahren im Gebrauch:

Zweistufige (Bi-Phase) Leitungscodierungen können durch zwei Spannungen erzeugt werden.
Der Vorteil hierbei ist eine einfache Takt-Rückgewinnung

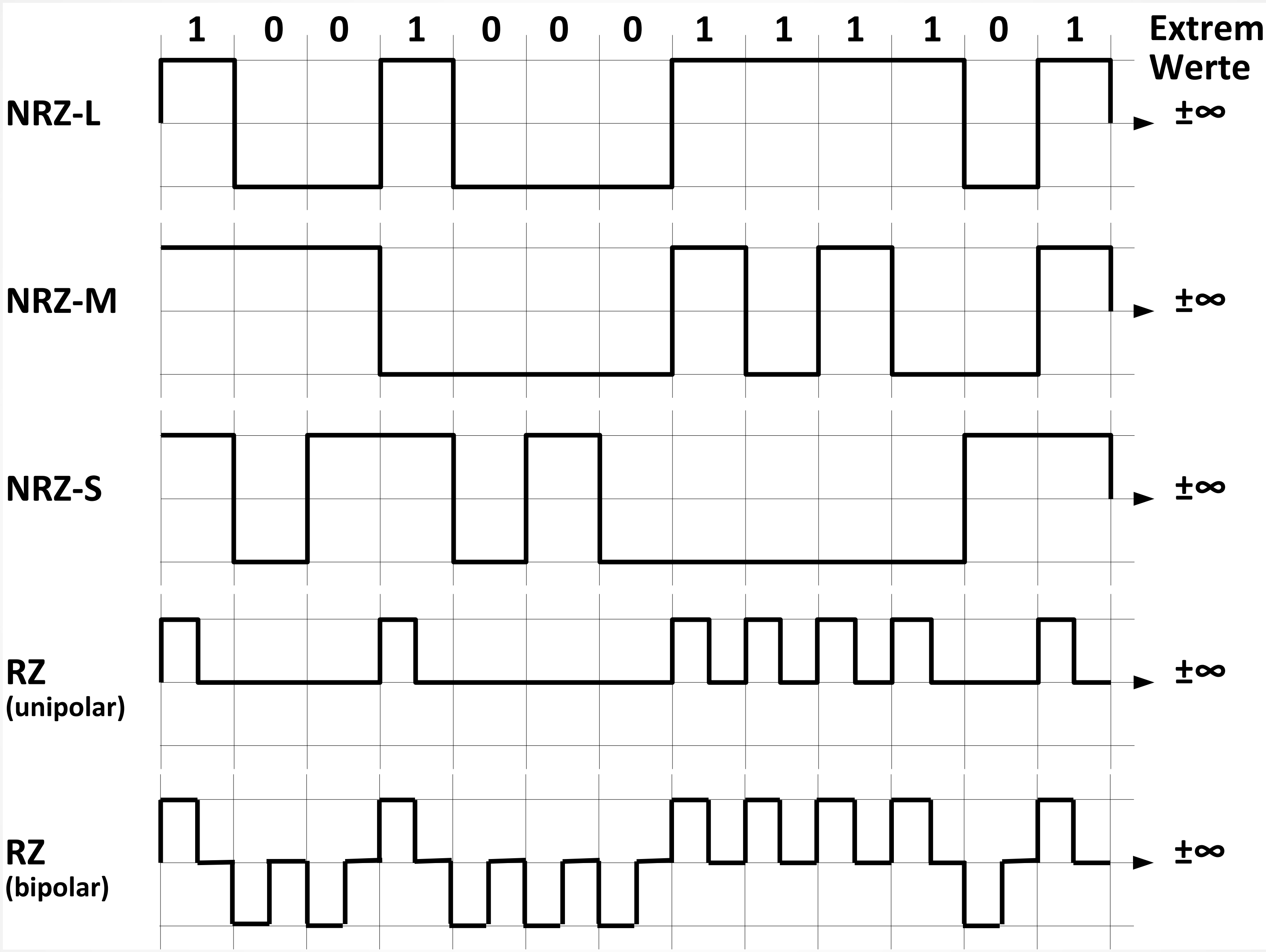
Dreistufige Leitungscodierung
Diese auch ternäre Codes genannten Wire-Codes haben 3 unterschiedliche Pegel. Z. B. MLT3.
Dazu werden 3 unterschiedliche Spannungen verwendet. +1, -1 und 0 Volt.

Vierstufige Leitungscodierung
Quarternäre Leitungscodes

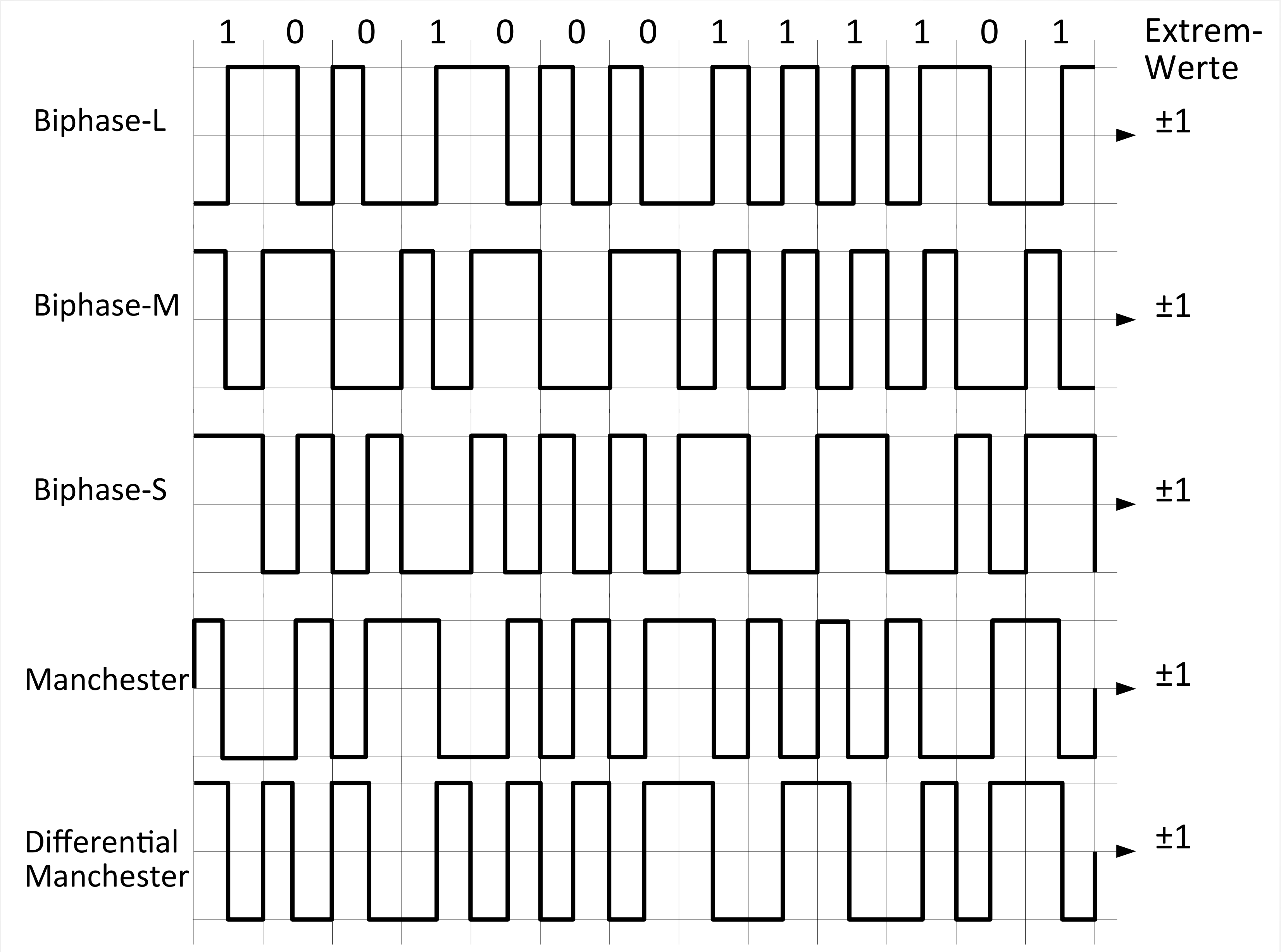
Fünfstufige Leitungscodes
Quinäre Leitungscodes

Mehrwertige Leitungscodes werden da eingesetzt wo die Fundamentalfrequenz auf den Leitung reduziert werden soll. Dafür sind dann auf der Empfängerseite hochwertige Empfänger-Bausteine notwendig, denn diese Codes sind störanfälliger.

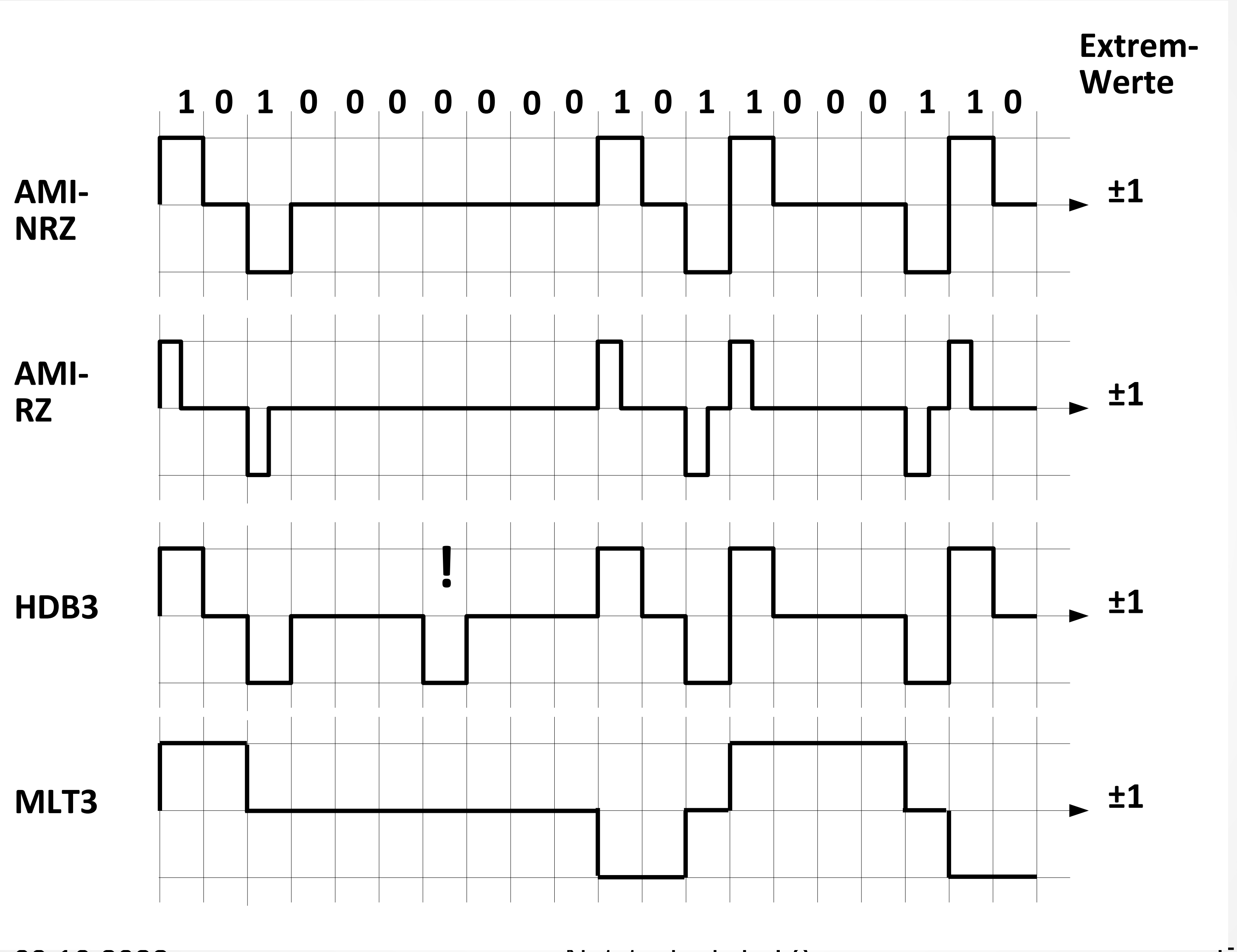
Wire-Coding:
RZ- / NRZ-Codes



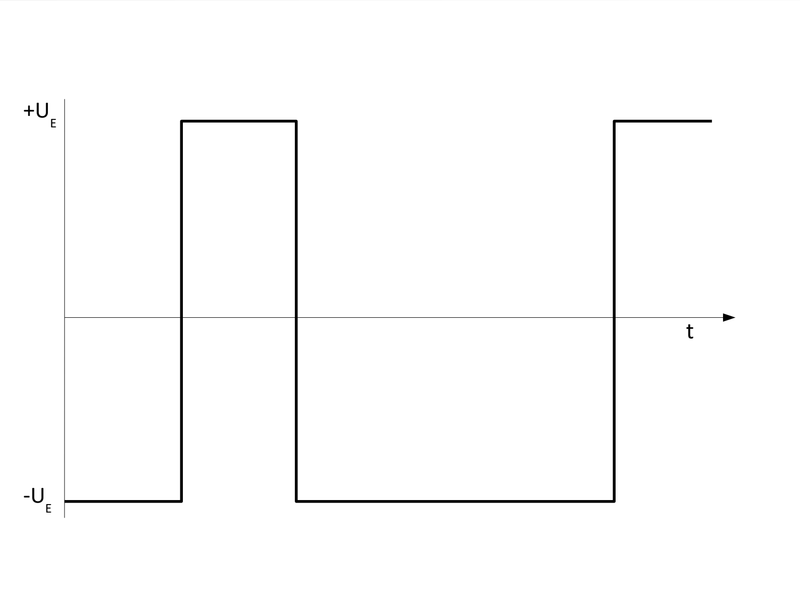
Wire-Coding: Biphase-Codes



Wire-Coding:
Tenary-Wire-Codes

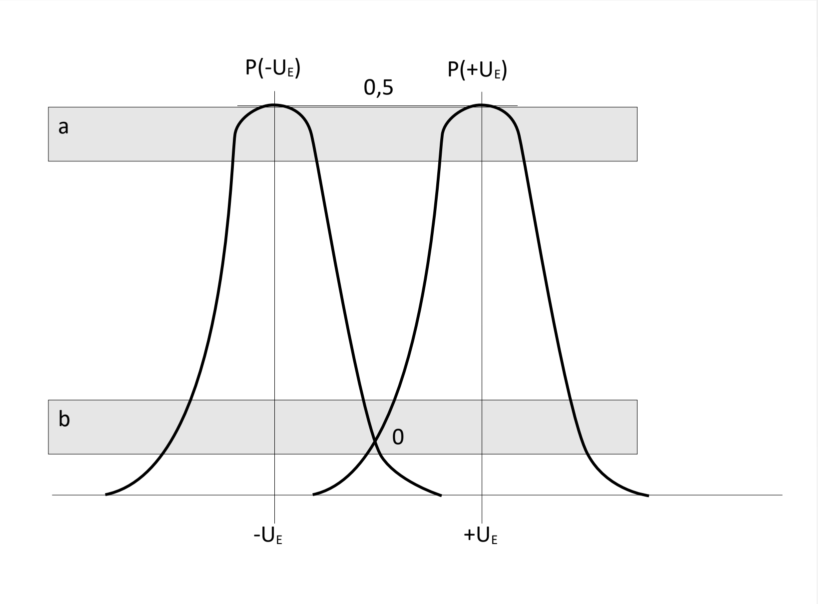


Bitfehler



Ein ideales Signal

Durch Störungen wie z. B. Rauschen wird die Wahrscheinlichkeit, dass das Signal so aussieht folgendermaßen aussehen.

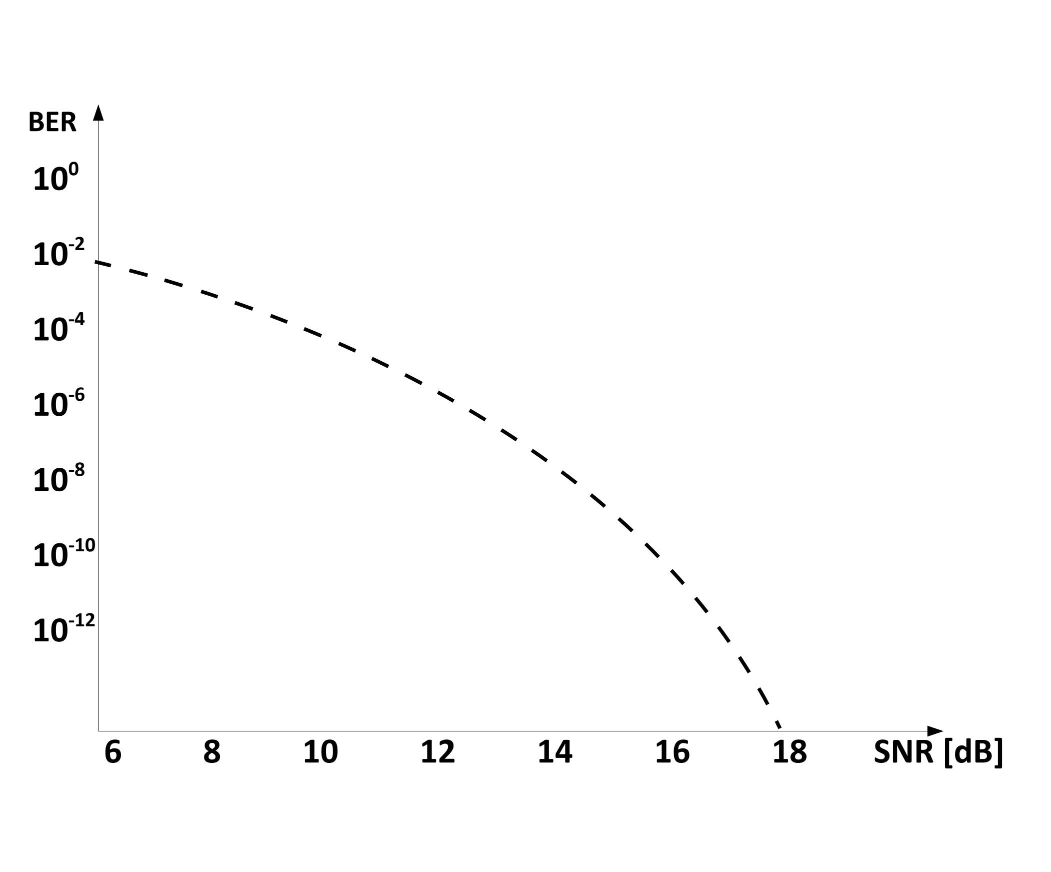


a
Es ist sehr selten ein Signal zu messen, das weit vom Wert $+U_E$ oder $-U_E$ entfernt ist.
b
Für ein rauschbehaftetes Signal ist es wahrscheinlich in der engen Umgebung von $+U_E$ oder $-U_E$ zu messen.
Bei einem Schwellwert von 0 kann ein Wert $+U_E$ nicht von einem Wert $-U_E$ unterschieden werden.
Dies ist der Bereich für Bitfehler.

Weitere Gründe sind :

- Ein-/Ausschalten von Geräten (Spannungsspitzen)
- Übersprechen zwischen Adernpaaren
- Reflexionen bei schlecht abgeschlossenen Leitungen
- Lichtbögen durch vorbeifahrende Züge
- Defekte Leitungen

Bitfehlerrate-2



$$BER = \frac{Anzahl\ der\ Bitfehler}{Anzahl\ aller\ übertragenen\ Bits}$$

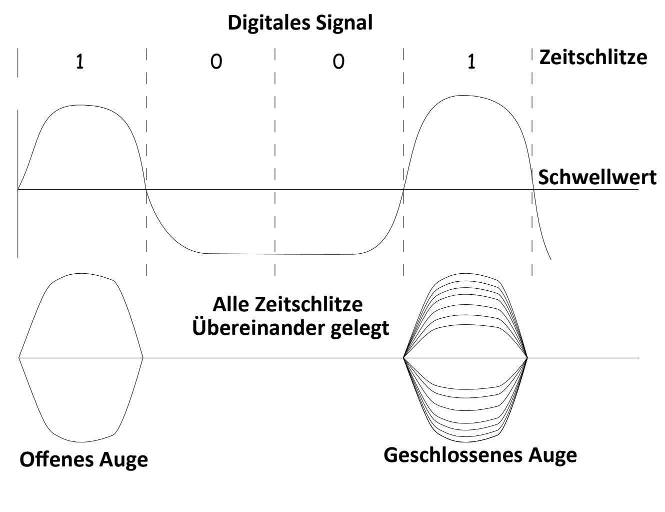
Wenn ein Bitfehler auftritt, ist es wahrscheinlich, dass ein weiterer Bitfehler auftritt.

Bitfehler haben ein Burst-Verhalten.
Dies bedeutet, dass sie gehäuft auftreten.

$$SNR = \frac{Nutzsignal - Leistung}{Rausch - Leistung} = \frac{P_N}{P_R} = 10 \log\left(\frac{P_N}{P_R}\right)[dB]$$

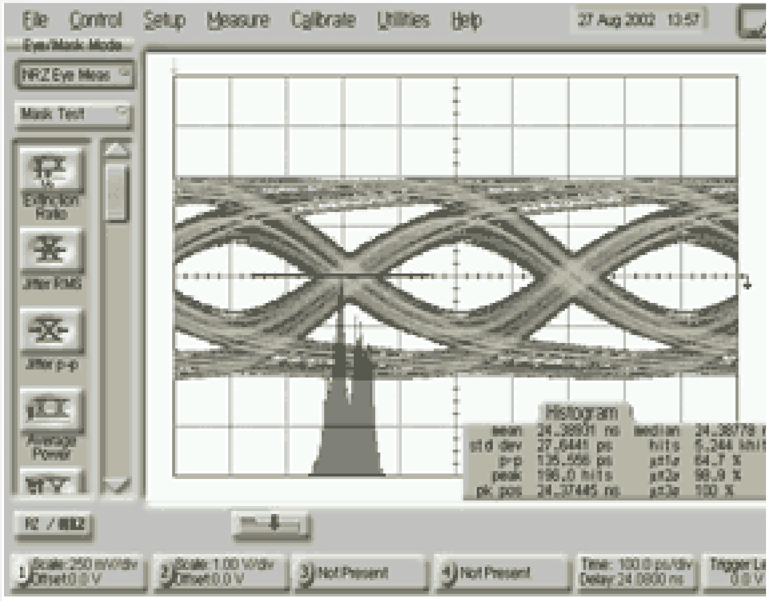
Je größer das SNR (Signal to Noise Ratio), desto weniger Bit-Fehler treten auf.

Augenmuster

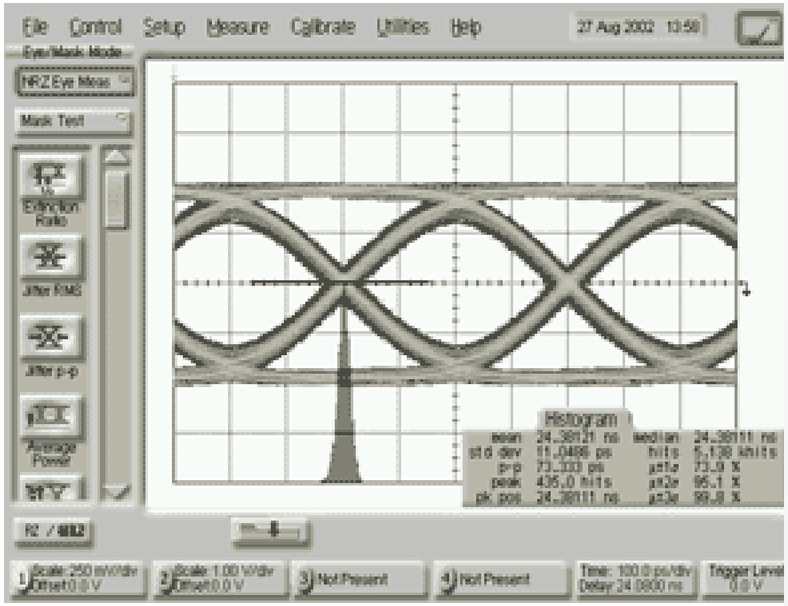


Die Qualität einer Übertragung kann auch mit den Augenmustern dargestellt werden.

Werden die Zeitschlitze übereinandergelegt, erhält man das typische Augenmuster.

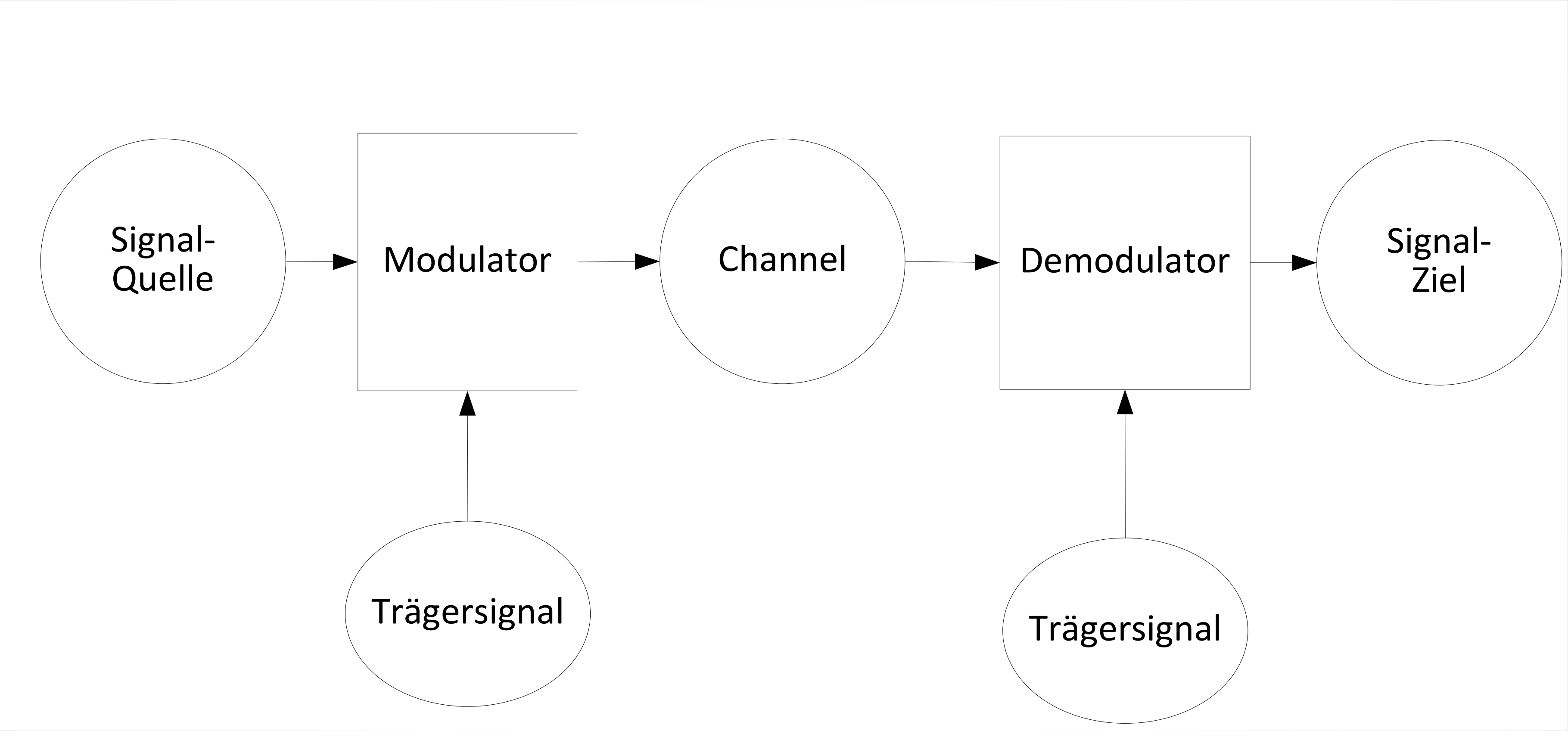


Geschlossenes Auge bei einer zweistufigen Leitungscodierung



Offenes Auge bei einer zweistufigen Leitungscodierung

Modulation


Signal-
Quelle
Modulator
Channel
Demodulator
Signal-
Ziel
Trägersignal
Trägersignal

Modulation: Möglichkeiten-1

Für **analoge Signale** werden folgende Parameter des Trägersignals beeinflusst:

Amplitude
Eine Beeinflussung der Amplitude wird Amplitudenmodulation genannt.

Frequenz
Wird die Frequenz durch das zu übertragende Signal beeinflusst, spricht man von Frequenzmodulation.

Phase
Eine Veränderung der Phase des Trägersignals wird Phasenmodulation genannt.

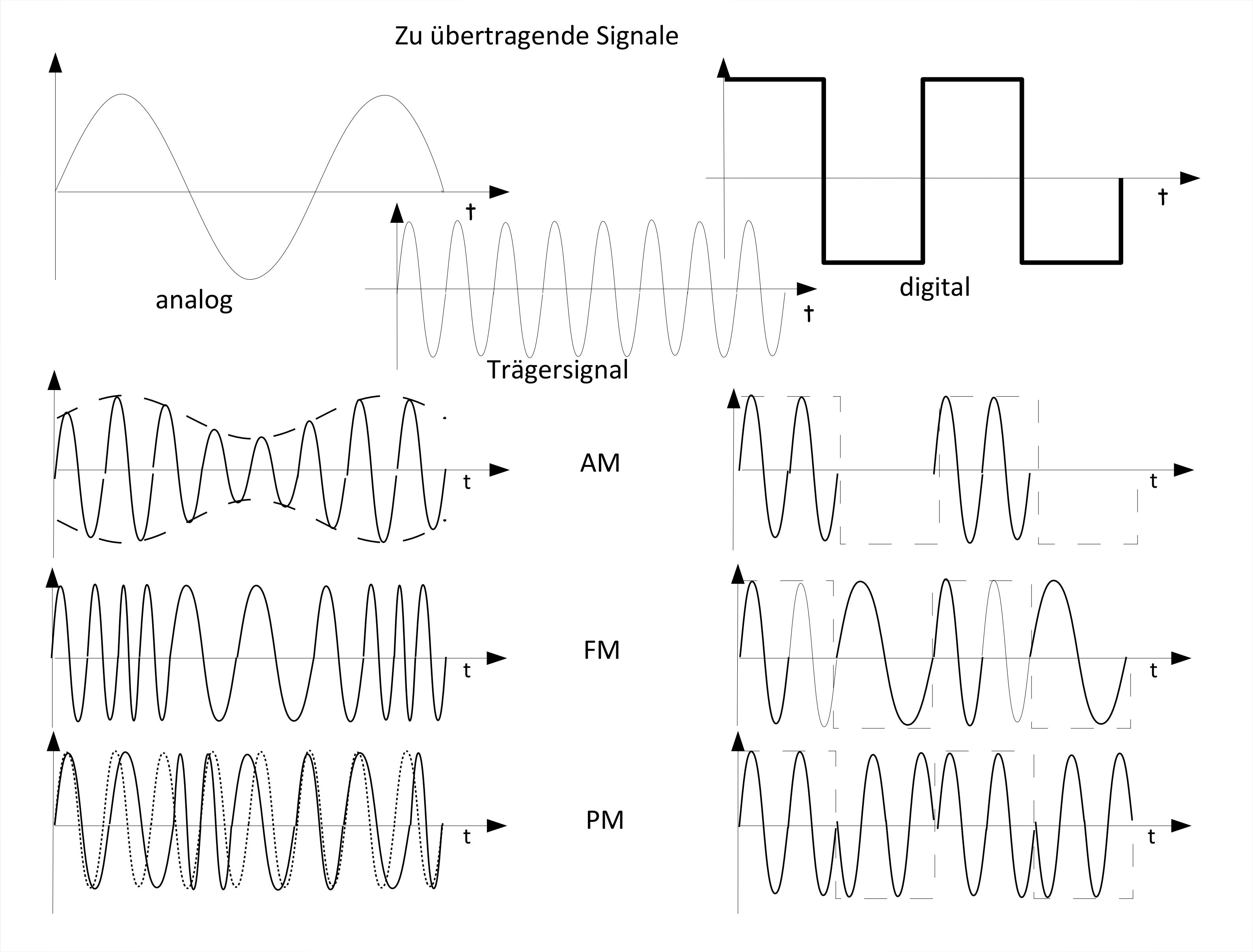
Bei **digitalen zu übertragenden Signalen** wird das Trägersignal abrupt geändert. Deshalb spricht man hier auch von einer Umtastung. Dabei gilt:

Amplitude
Eine Beeinflussung der Amplitude wird Amplitudenumtastung (engl. Amplitude Shift Keying) genannt.

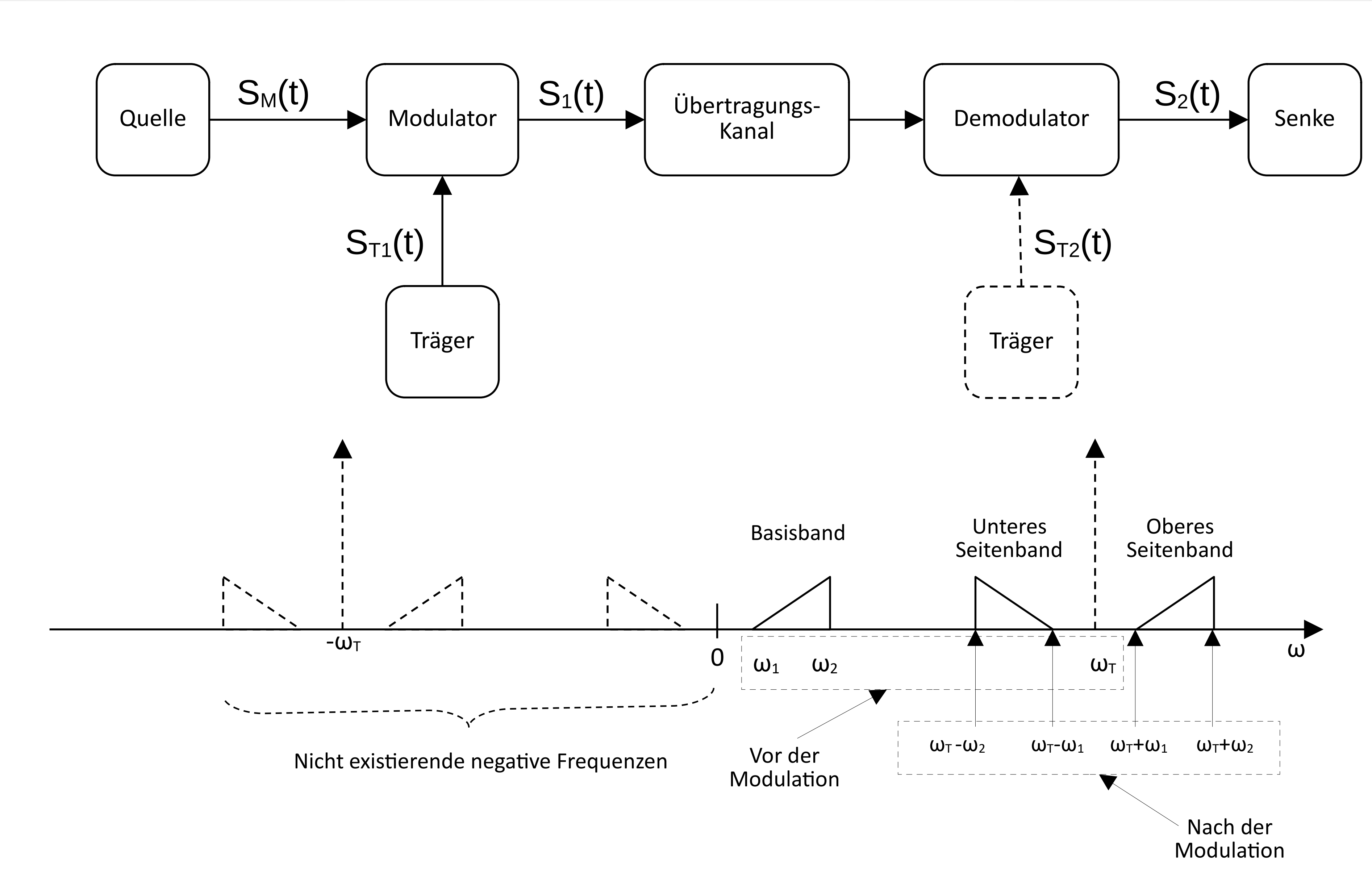
Frequenz
Wird die Frequenz durch das zu übertragende Signal beeinflusst, spricht man von Frequenzumtastung (engl. Frequency Shift Keying).

Phase
Eine Veränderung der Phase des Träger-Signals wird Phasenumtastung (engl. Phase Shift Keying) genannt.

Modulation: Möglichkeiten-2



Lineare Modulation



Mathematische Betrachtung

(1) $s_T(t)=\hat{S}_T\cos(\omega_T(t)+\varphi_T)$ Sinusträger

(2) $s_M(t)=\hat{s}_M(t)\cos(\omega_M t+\varphi_M)$ Modulierendes Signal

(3) $s_1(t)=k_1\cdot S_M(t)\cdot\hat{S}_T\cos(\omega_T t+\varphi_T)$ Modulationsprodukt

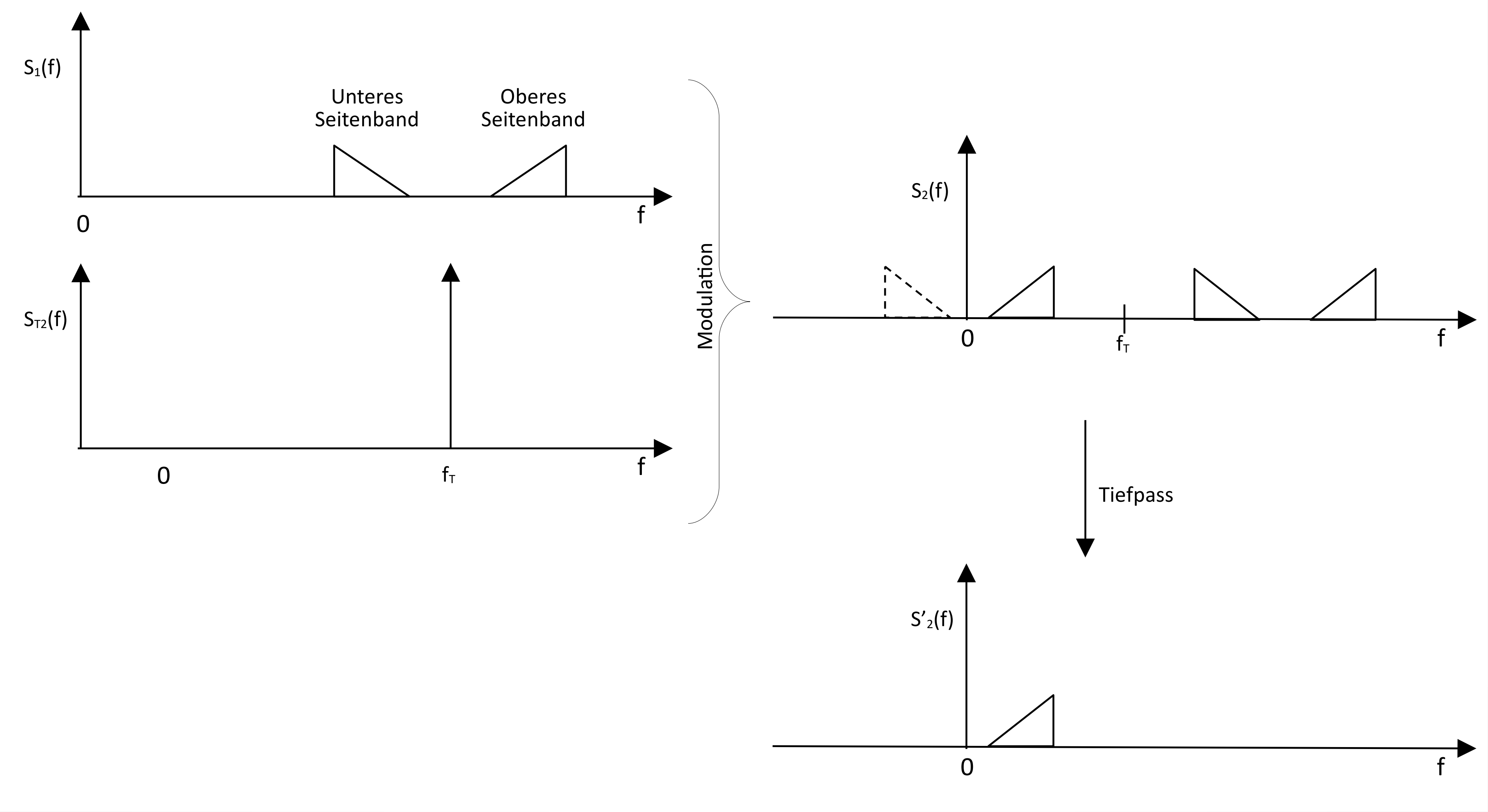
(4) $\cos\alpha\cdot\cos\beta = \frac{1}{2}[\cos(\alpha+\beta)+\cos(\alpha-\beta)]$ Trigonomische Beziehung

(5) $s_1(t)=\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T\cdot\cos(\omega_T+\omega_M)+(\varphi_T+\varphi_M)+\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T\cdot\cos(\omega_T-\omega_M)+(\varphi_T-\varphi_M)$ Modulations-produkt

(6) $\varphi_T = \varphi_M = 0$ $\hat{S}_1=\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T$ Vereinfachungen

(7) $S_1(t)=\hat{S}_1\cdot\cos(\omega_T+\omega_M)+\hat{S}_1\cdot\cos(\omega_T-\omega_M)$ Vereinfachtes Modulationsprodukt

Demodulation



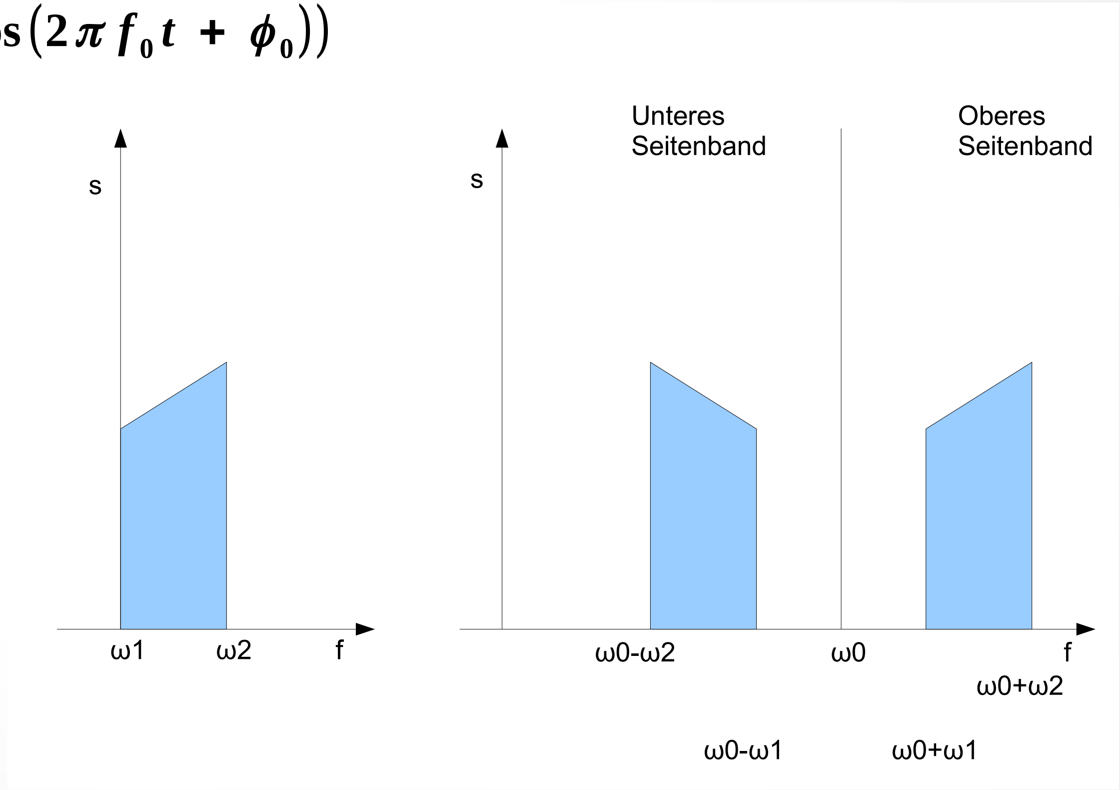
Modulation: Amplitudenmodulation

Hierbei wird die Amplitude des Träger-Signals $S_C(t)$ mit dem zu übertragenden Signal moduliert.

$$S_C(t) = (a_0 \cdot \cos(2\pi f_0 t + \phi_0))$$

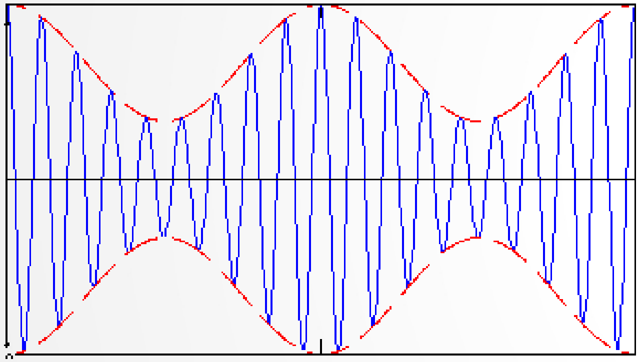
Das Amplitudenmodulierte Signal S_{AM} ergibt sich zu:

$$S_{AM}(t)=(a_0 + a1 \cdot x(t) \cdot \cos(2\pi f_0 t + \phi_0))$$



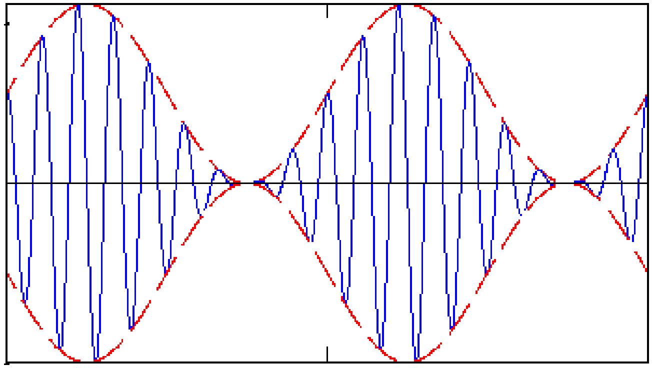
Modulationsindex / Modulationsgrad

$$m = \frac{a_1}{a_0}$$ Der Modulationsindex errechnet sich aus der vorigen Formel



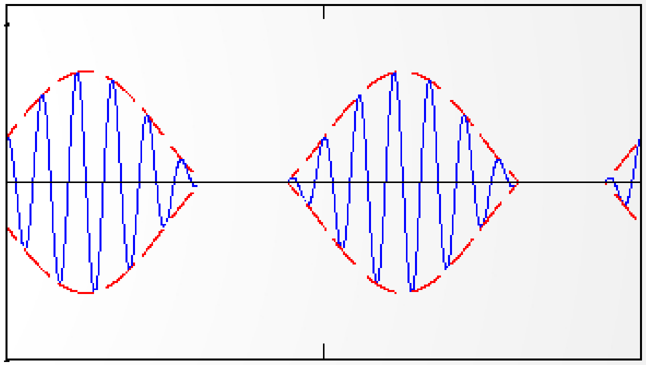
Modulationsindex M = 0,5

50% Modulation



Modulationsindex M = 1

100% Modulation



Modulationsindex M = 1,5

Übermodulation

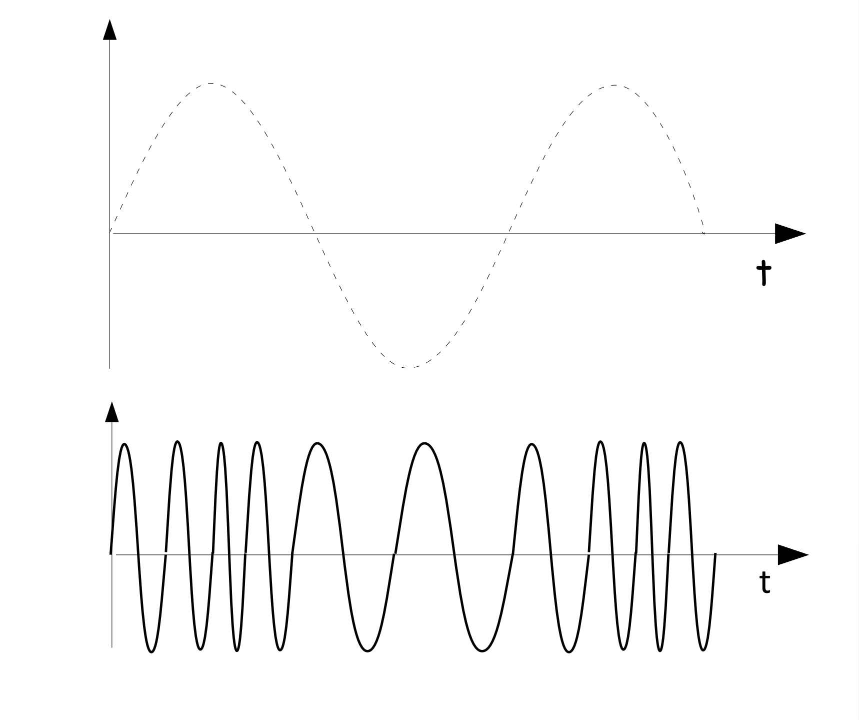
Modulation: Frequenzmodulation (FM)

Die Momentankreisfrequenz ω(t) = 2πf(t) der Trägerschwingung wird mit dem Quellensignal x(t) beeinflusst.

$$\omega(t) = \omega_0 + \Delta\Omega \cdot x(t)$$

Das FM-Signal kann vereinfacht folgendermaßen beschrieben werden.

$$S_{FM}(t) = a_0 \cdot \cos(\Phi(t) + \varphi_0)$$



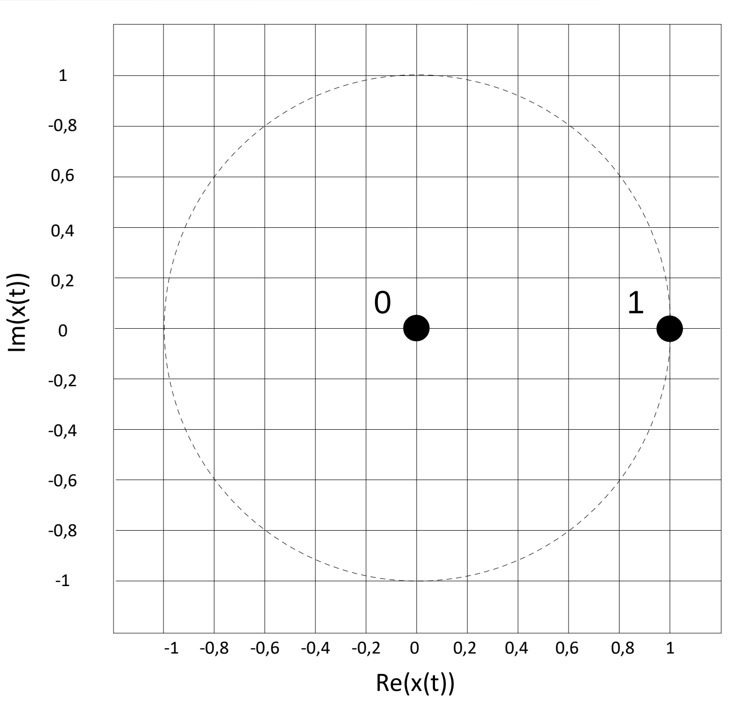
Binäre Modulationsverfahren

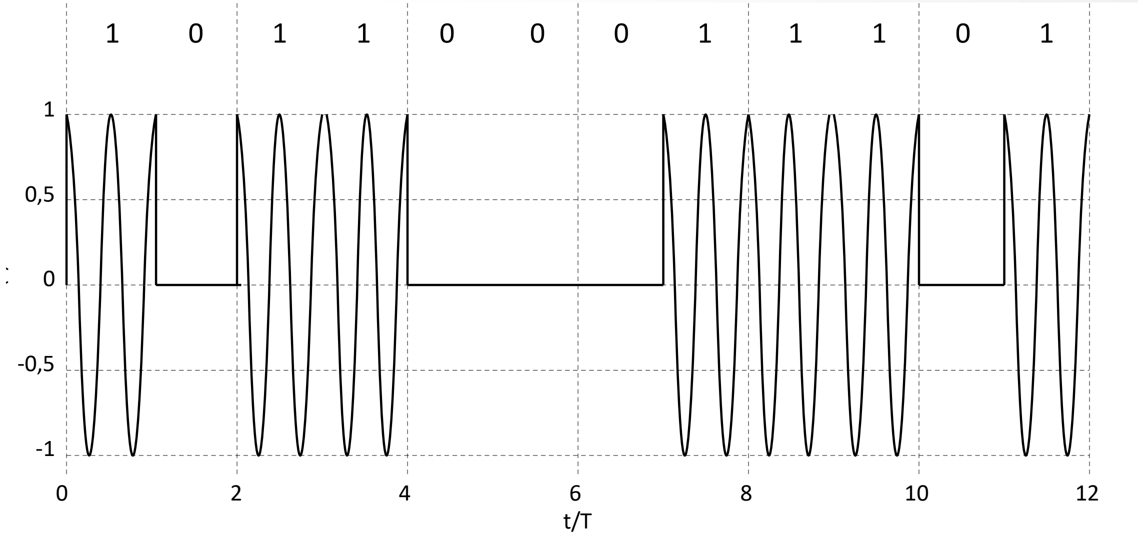
Wie bereits bei den Modulationsverfahren beschrieben wird die Modulation - da nur Einsen oder Nullen übertragen werden - auch Umtastung (eng. shift keying) genannt.

Wird nur ein Bit übertragen, muss das übertragene Signal nur 2 Zustände ($x_0(t)$ und $x_1(t)$) annehmen.

In den folgenden Beispielen soll immer die gleiche Folge an Informationsbits ((101100011101) übertragen werden.

Unipolare Modulation

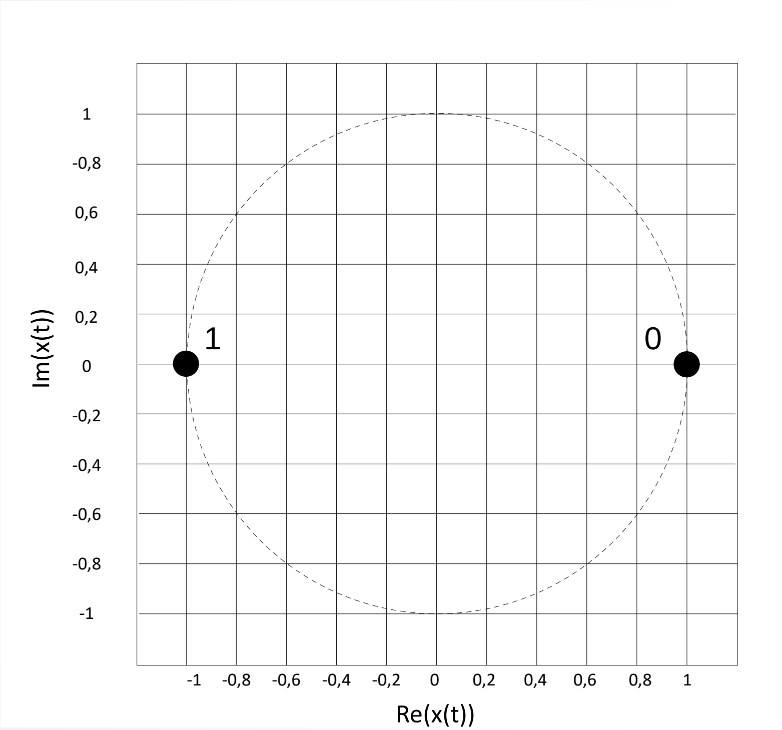




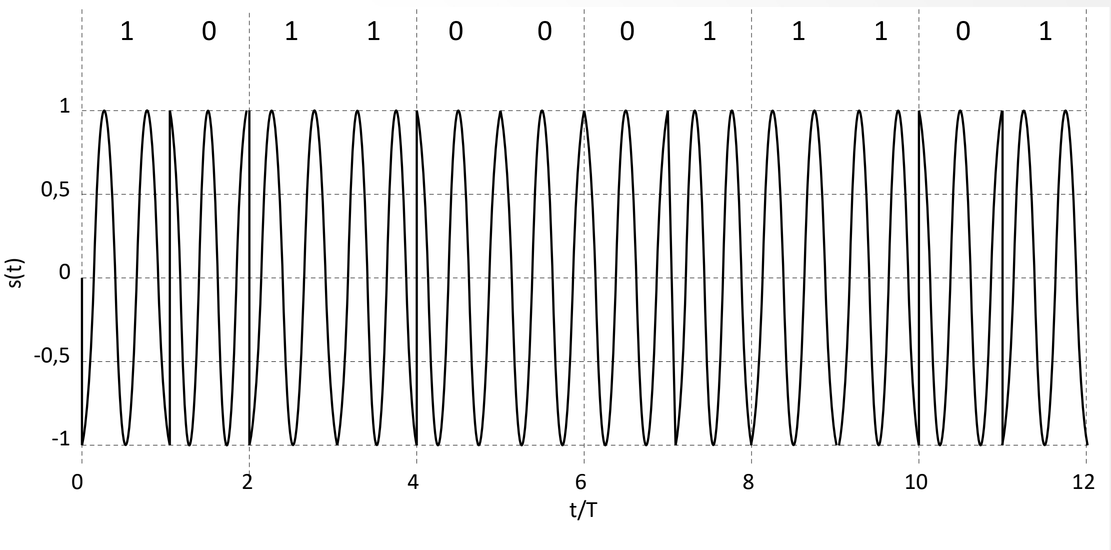
Signalverlauf der unipolaren Übertragung

Komplexe Ebene bei der unipolaren Übertragung

Bipolare Modulation

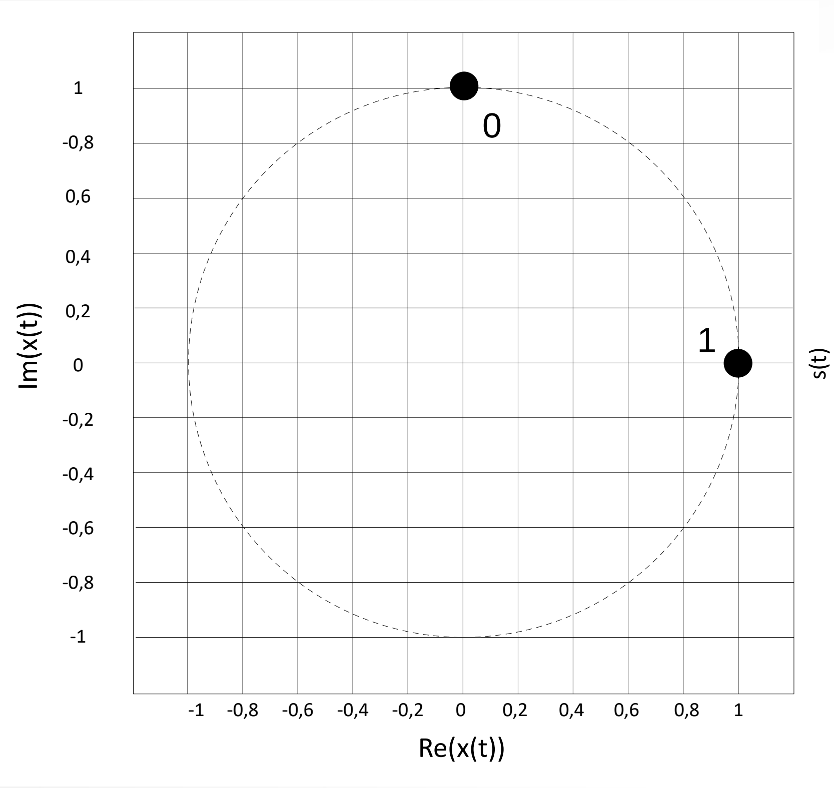


Komplexe Ebene bei der bipolaren Übertragung

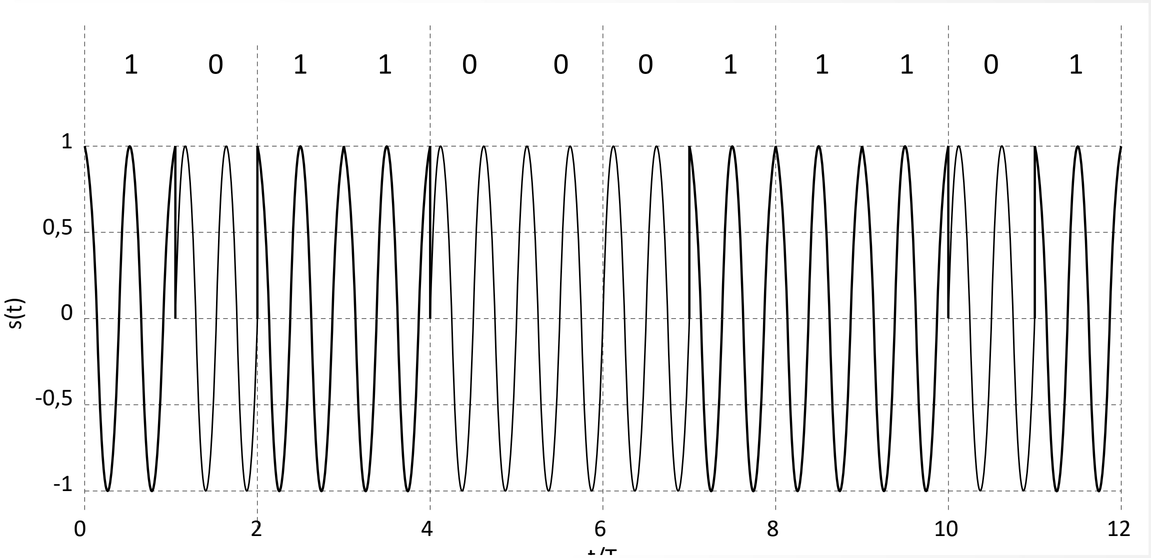


Signalverlauf der bipolaren Übertragung

Orthogonale Modulation



Komplexe Ebene bei der orthogonalen Übertragung



Signalverlauf einer orthogonalen Übertragung

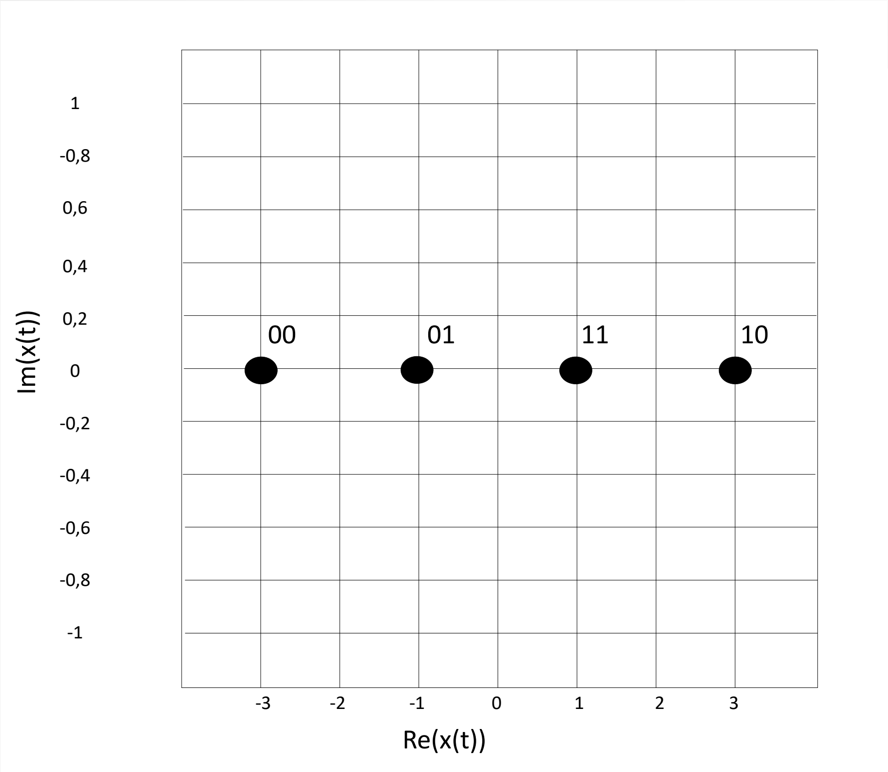
Mehrwertige Modulationsverfahren

Um mehr als nur ein Bit pro Symbol zu übertragen, können mehrwertige Modulationsverfahren verwendet werden.

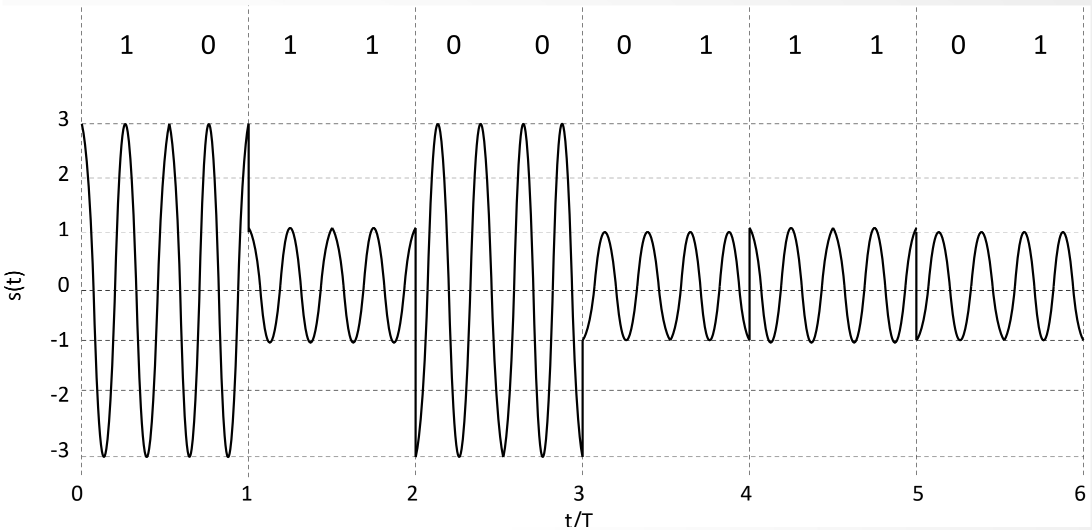
Ein Unterscheidungsmerkmal, um die unterschiedlichen Verfahren zu bewerten, ist die Euklidische Distanz

$$\delta = \sqrt{\left(\Re\{x_i(t)\} - \Re\{x_j(t)\}\right)^2 + \left(\Im\{x_i(t)\} - \Im\{x_j(t)\}\right)^2}$$

Mehrwertige ASK (Amplitude Shift Keying)

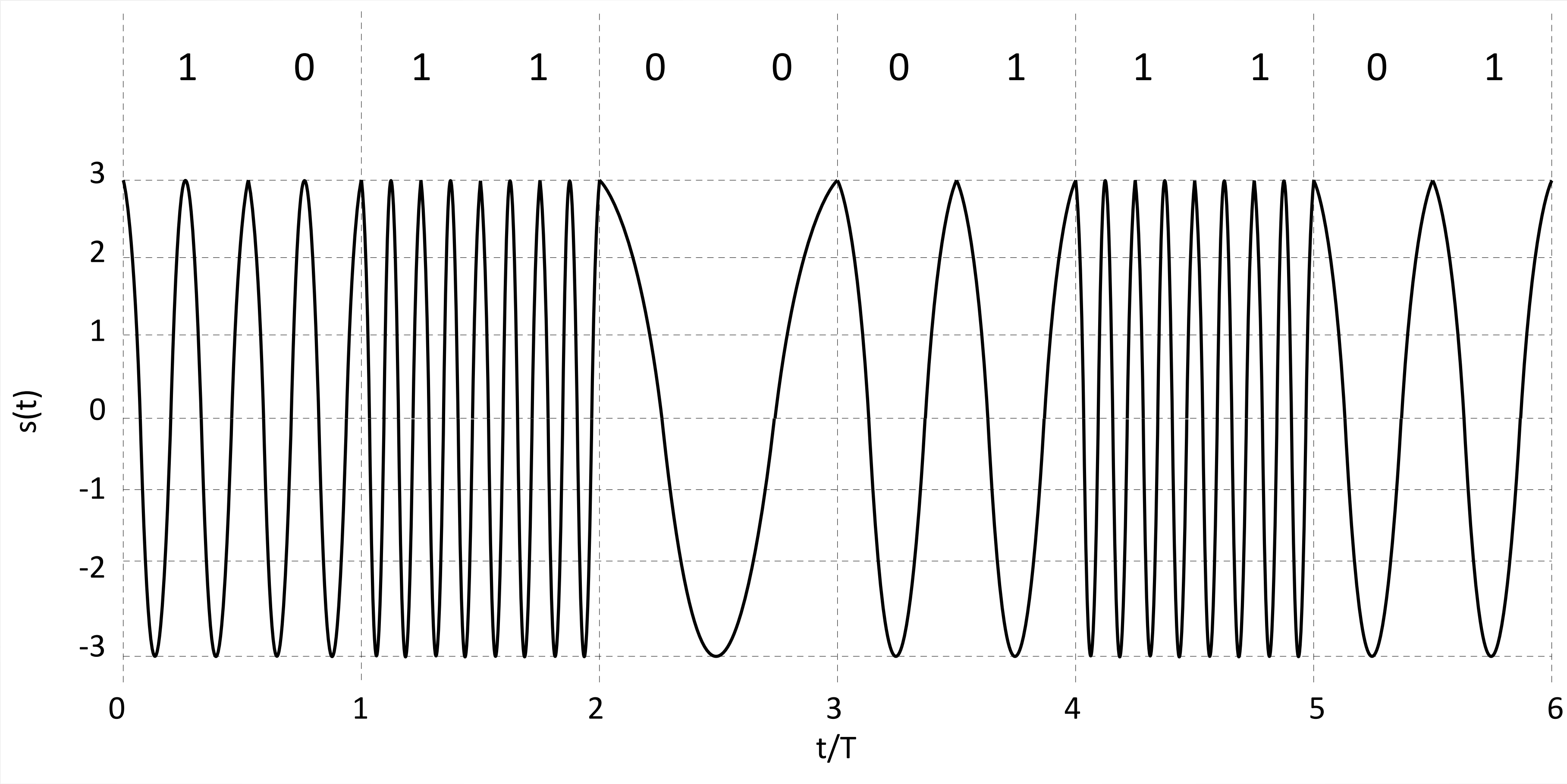


Komplexe Ebene bei der Übertragung von 4-ASK

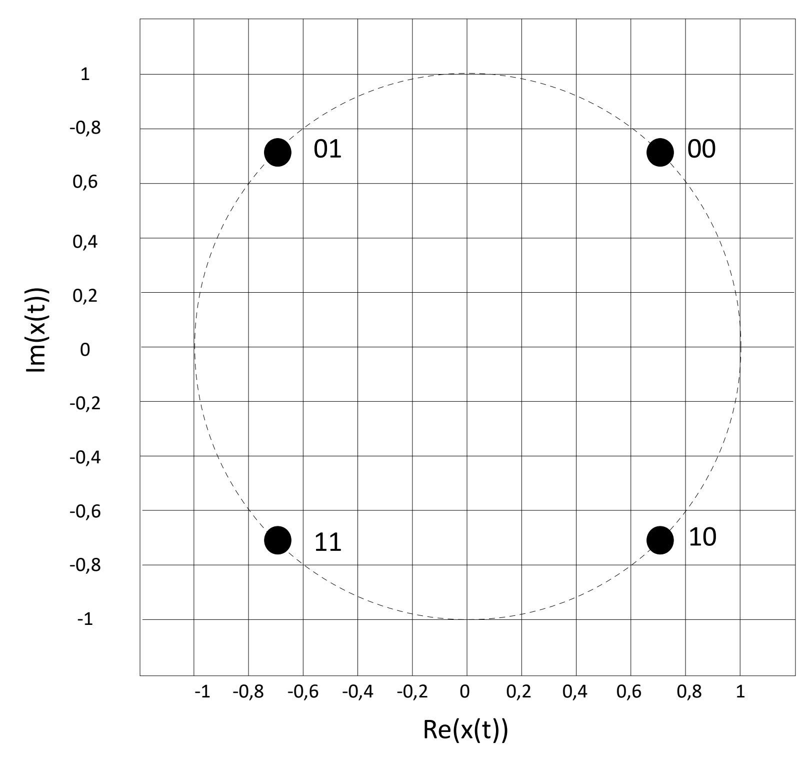


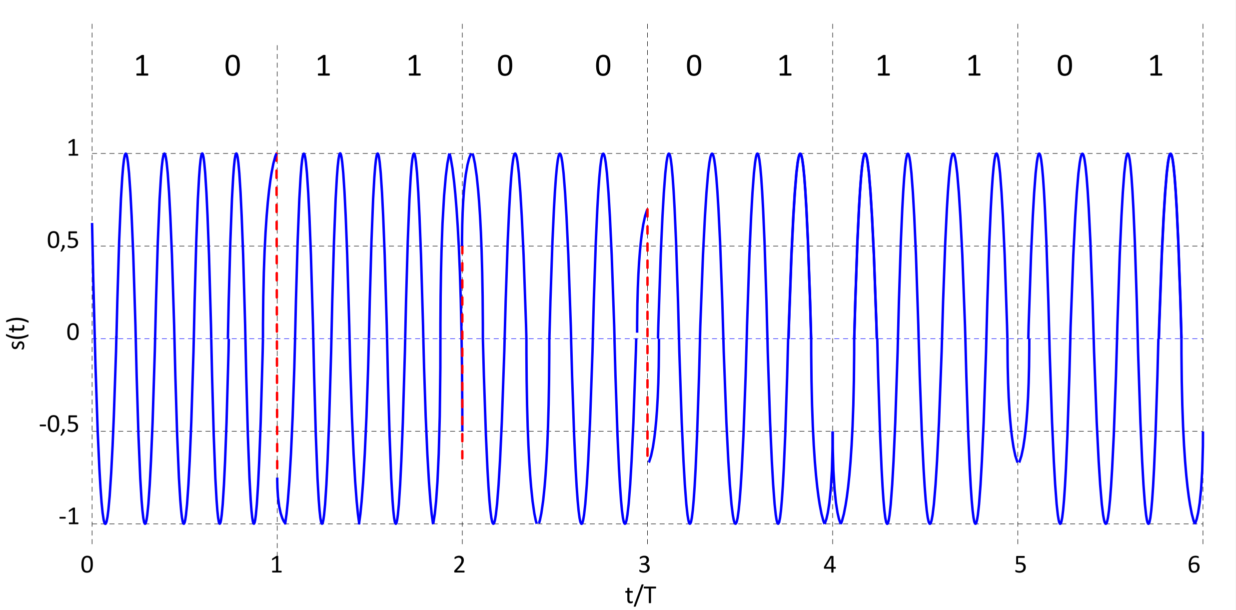
Signalverlauf einer 4-ASK-Übertragung

Mehrwertige FSK (Frequency Shift Keying)

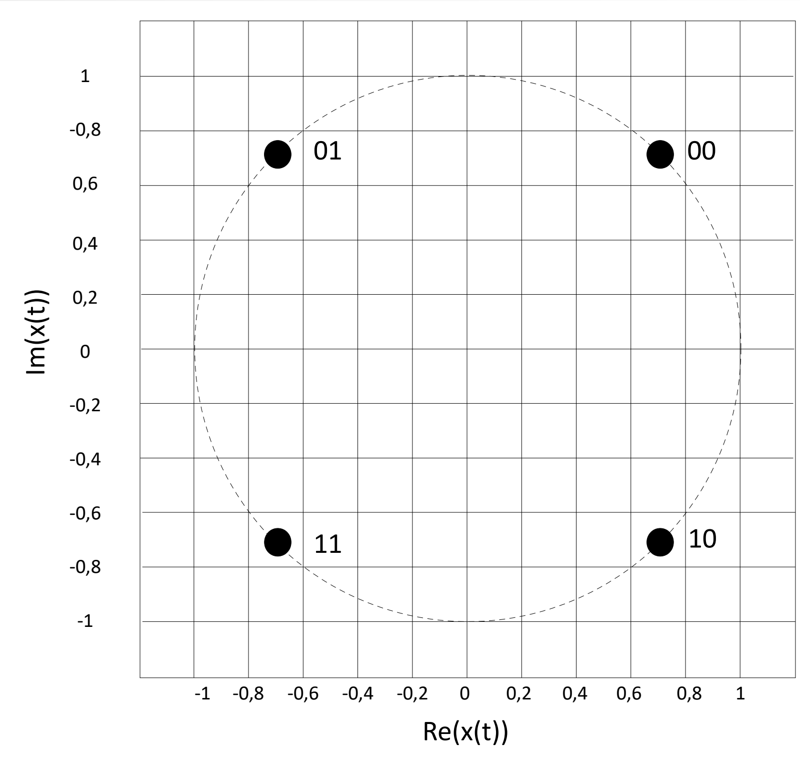


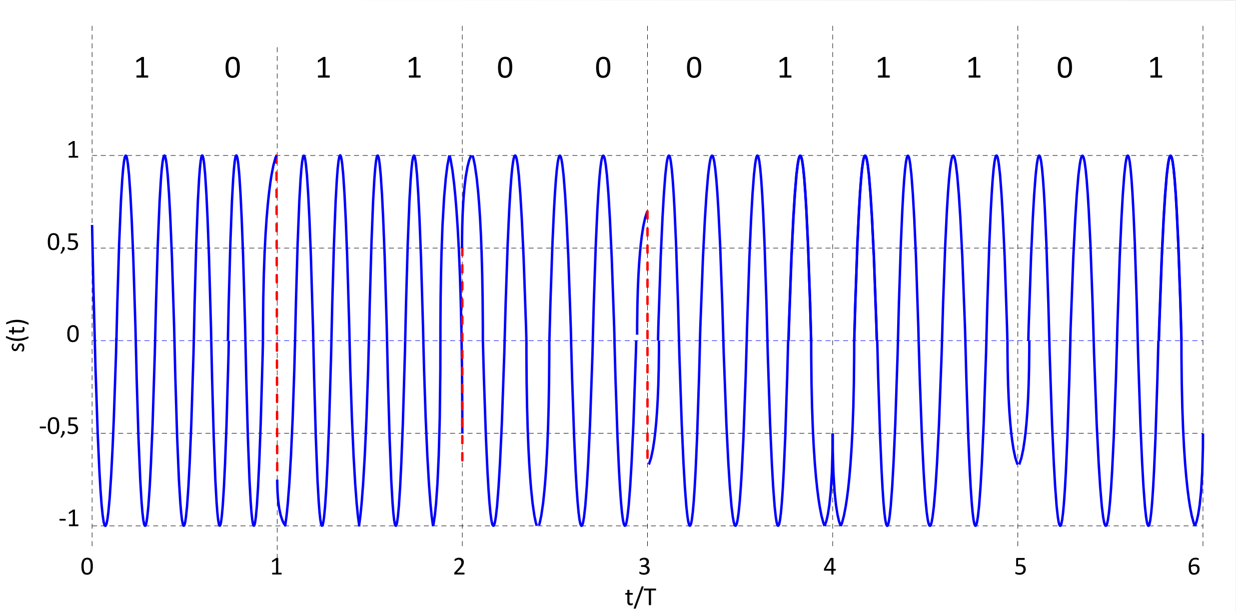
Mehrwertige PSK (Phase Shift Keying)



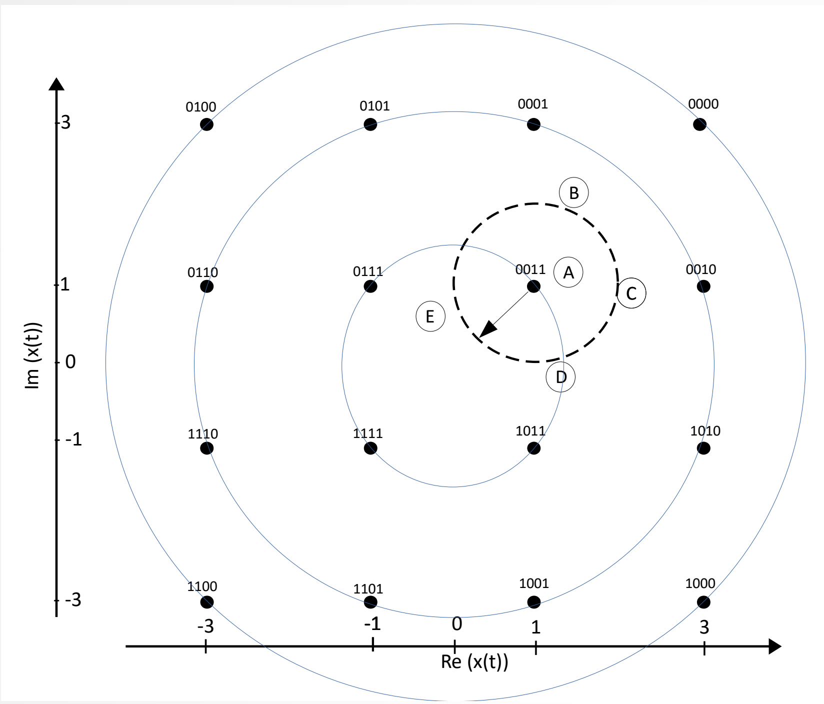


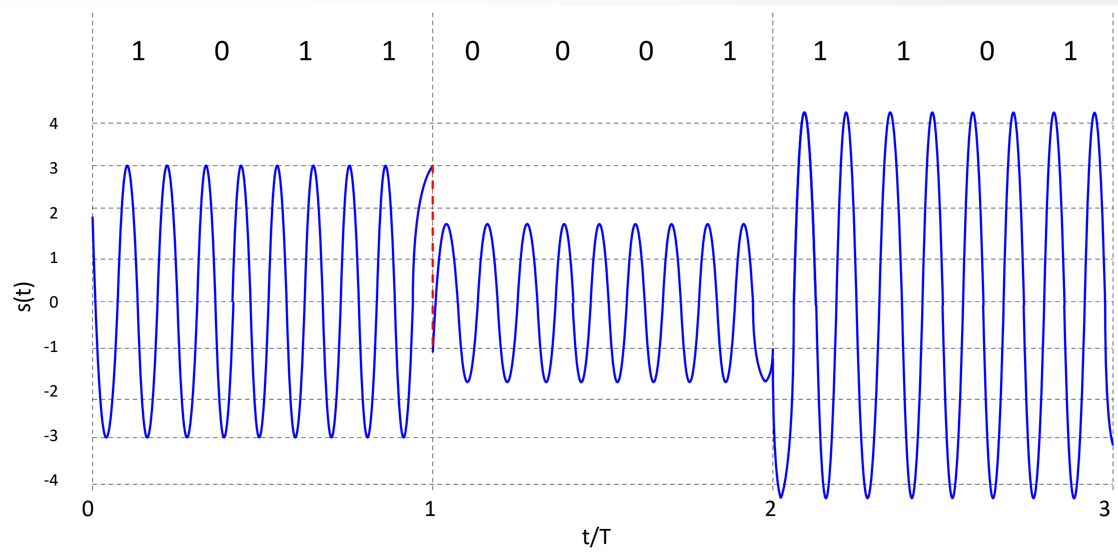
Mehrwertige PSK (Phase Shift Keying)





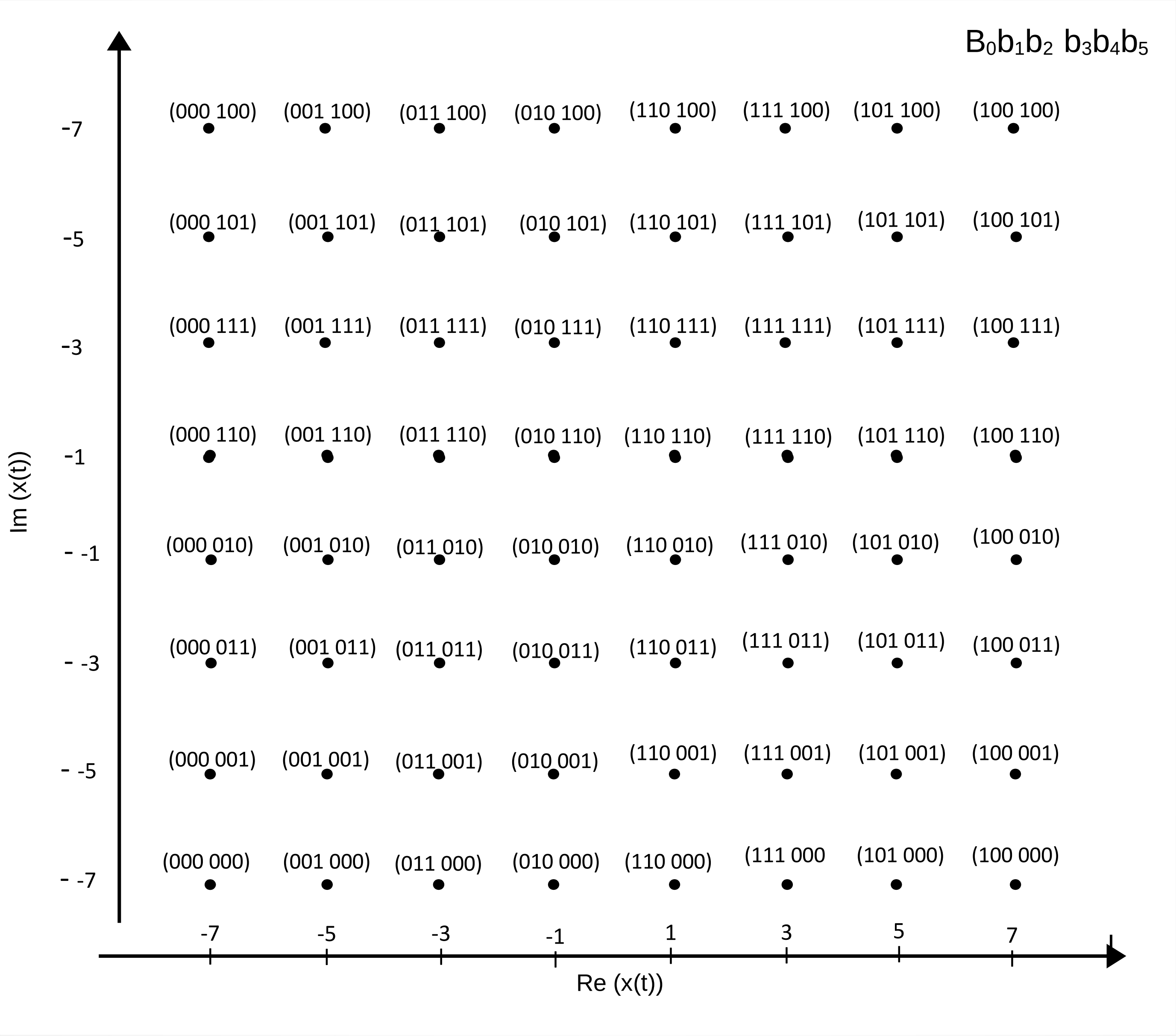
Modulation:
Quadrature Amplitude Modulation (QAM)





Kombiniert man die PSK noch mit unterschiedlichen Amplituden kommt man zur Quadrature Amplitude Modulation (QAM).
Hier werden 3 unterschiedliche Radien (Amplituden) und damit unterschiedliche Energien, sowie 12 Phasenwinkel verwendet.

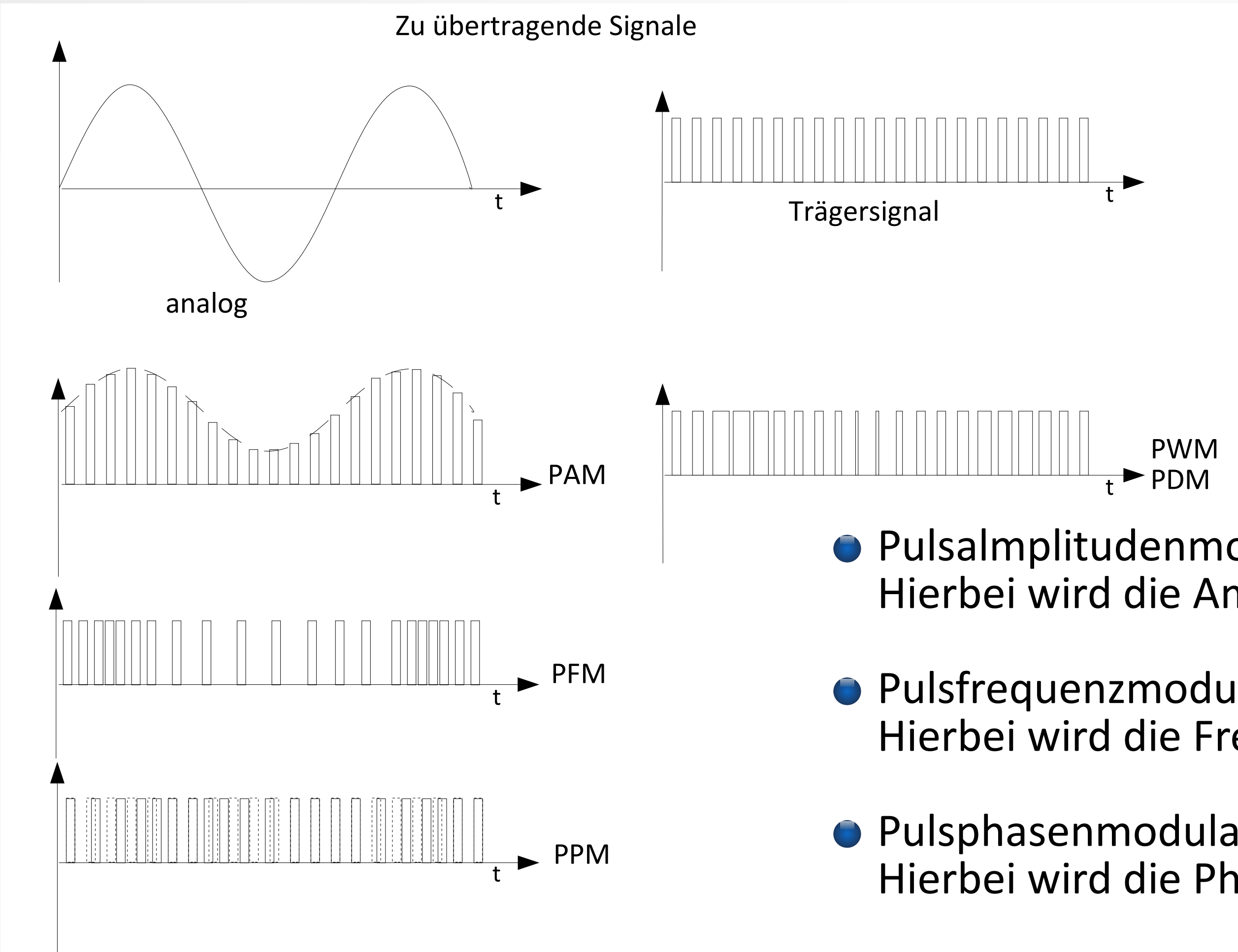
64 -QAM



$B_0b_1b_2\ b_3b_4b_5$
(000 100) (001 100) (011 100) (010 100) (110 100) (111 100) (101 100) (100 100)
(000 101) (001 101) (011 101) (010 101) (110 101) (111 101) (101 101) (100 101)
(000 111) (001 111) (011 111) (010 111) (110 111) (111 111) (101 111) (100 111)
(000 110) (001 110) (011 110) (010 110) (110 110) (111 110) (101 110) (100 110)
(000 010) (001 010) (011 010) (010 010) (110 010) (111 010) (101 010) (100 010)
(000 011) (001 011) (011 011) (010 011) (110 011) (111 011) (101 011) (100 011)
(000 001) (001 001) (011 001) (010 001) (110 001) (111 001) (101 001) (100 001)
(000 000) (001 000) (011 000) (010 000) (110 000) (111 000 (101 000) (100 000)
-7
-5
-3
-1
- -1
- -3
- -5
- -7
Im (x(t))
-7 -5 -3 -1 1 3 5 7
Re (x(t))

Modulation: Pulsmodulation

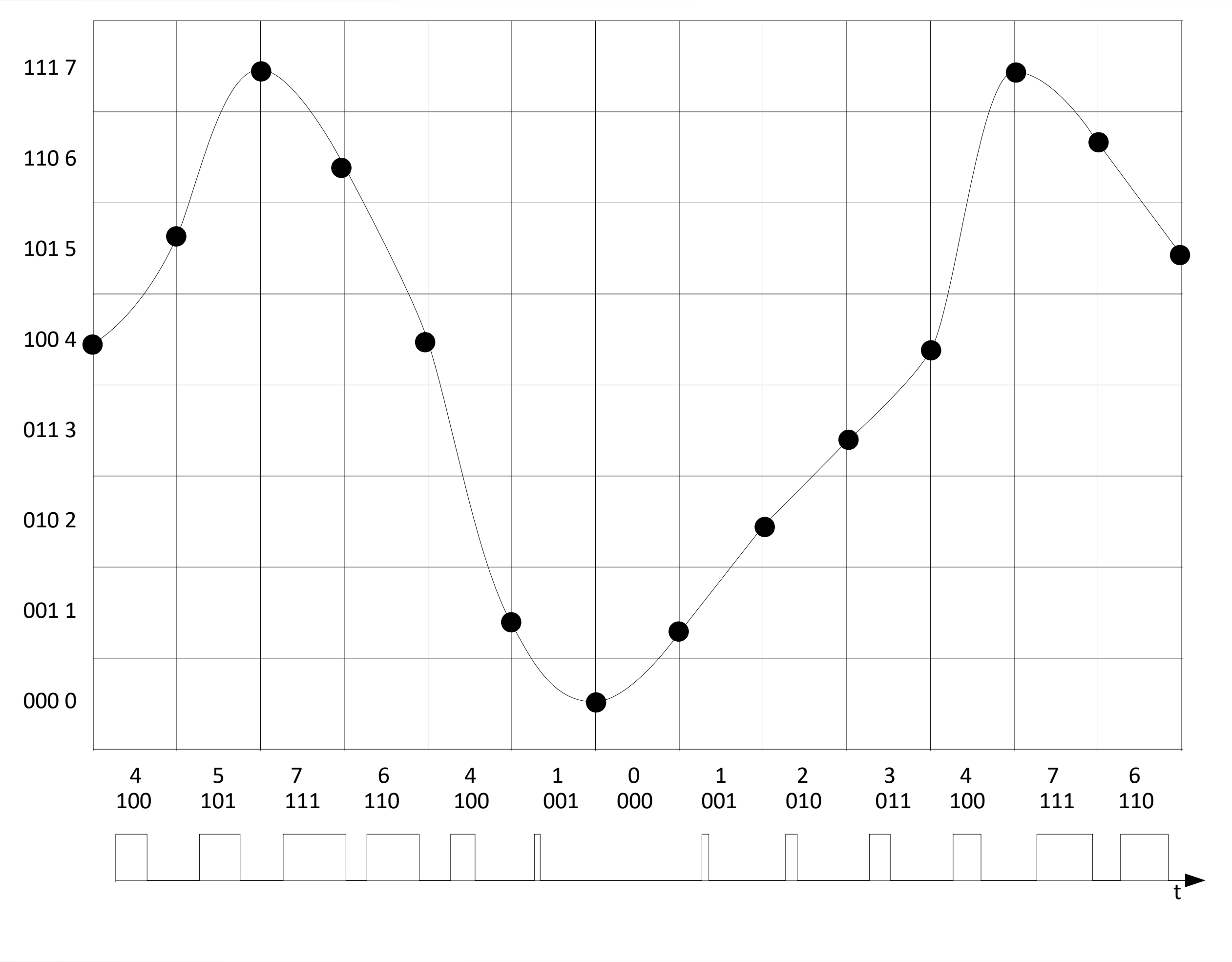
Bei der Pulsmodulation wird als Träger keine Sinuswelle sondern eine Pulsfolge verwendet.



- Pulsalmplitudenmodulation (PAM)
  Hierbei wird die Amplitude der Pulsfolge moduliert.
- Pulsfrequenzmodulation (PFM)
  Hierbei wird die Frequenz der Pulsfolge moduliert.
- Pulsphasenmodulation (PPM)
  Hierbei wird die Phase der Pulse moduliert.
- Pulsweitenmodulation (PWM) Pulsdauermodulation (PDM)
  Hierbei wird die Dauer / Weite der Pulse durch die Modulation beeinflusst.

Modulation:
Puls-Code-Modulation

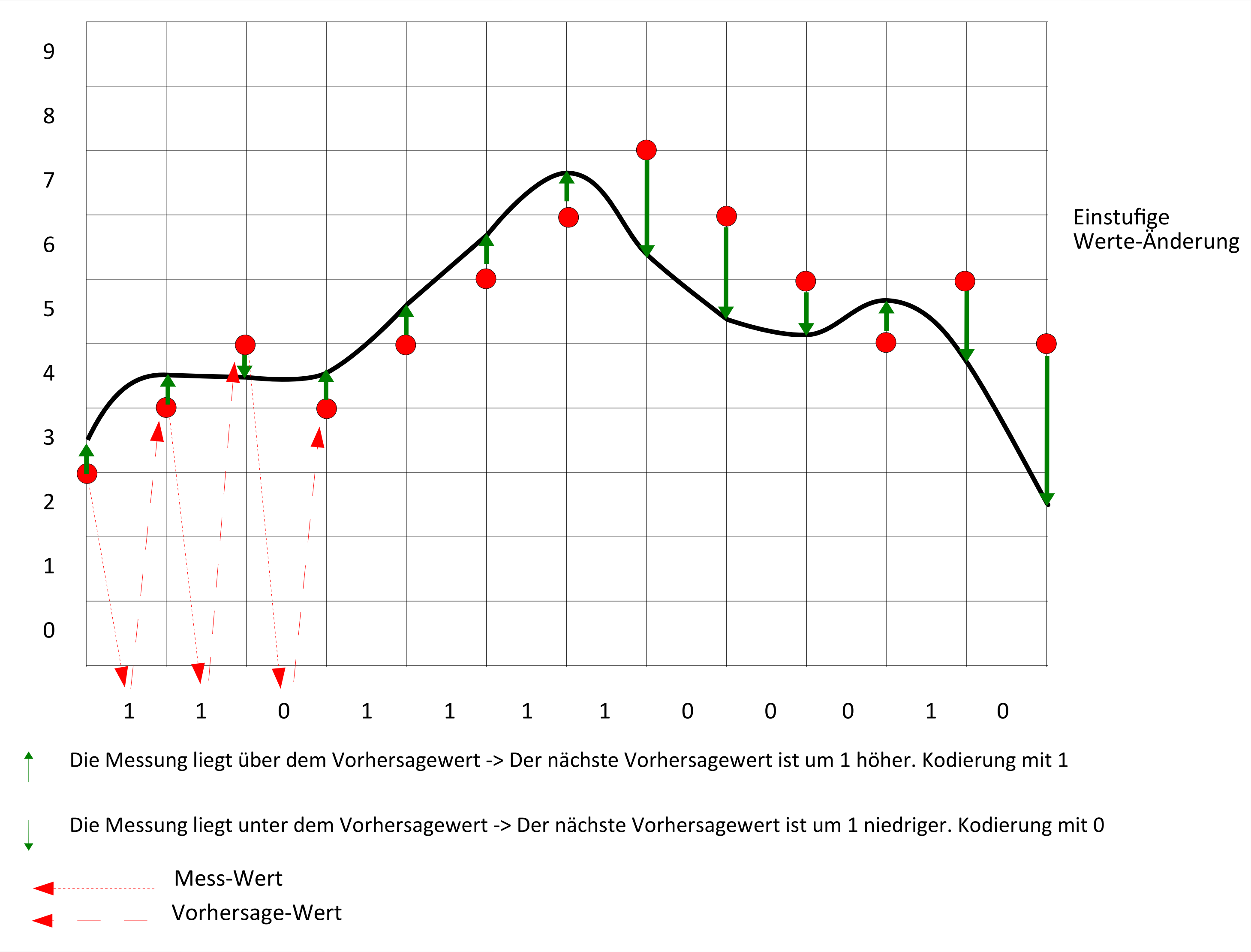
Hierbei wird das Ergebnis der Quantisierung bei der Abtastung für die Modulation verwendet.
Das Verfahren wird PCM (Pulse Code Modulation) genannt.



In der Abbildung ist der Kurvenverlauf in der PCM moduliert.

Der resultierende Wire-Code ist ein NRZ-Code.

Modulation:
Deltamodulation (DM) / Differenz Puls Code Modulation (DPCM)



Zusammenfassung

Codierung (Source Coding / Channel Coding / Wire-Coding)

Source-Coding (Huffman-Codierung (tabellarisch / graphisch))

Channel-Coding (Paritätsbits / Hamming-Code / CRC)

Wire-Coding (Gleichstromanteil / RZ-Codes / NRZ-Codes / Biphase-Codes / Tenary-Codes / ..)

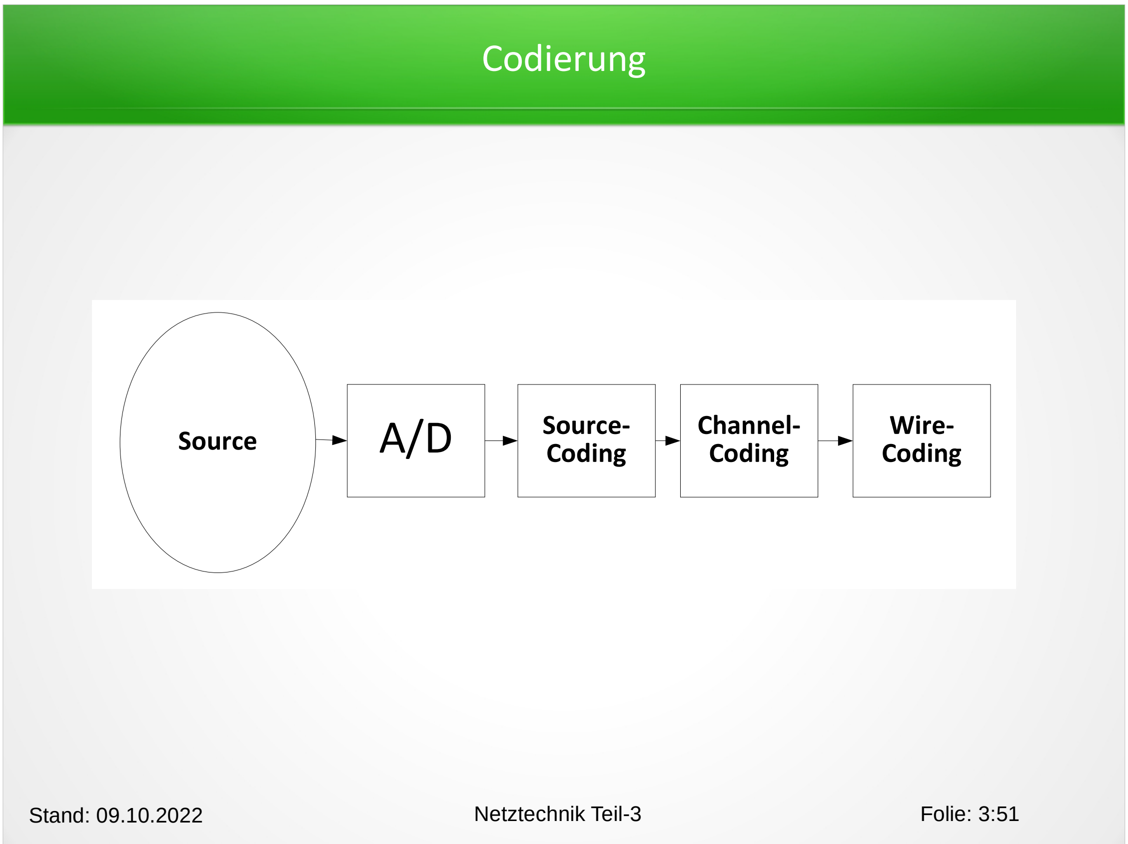
Bitfehlerrate / Augenmuster

Modulation (AM / FM /PM /ASK / FSK / PSK / BPSK / QAM / xQAM /…
… / Puls- / Delta- / Adaptive DM / Differenz Puls Code Modulation (DPCM))

Netztechnik-Vorlesung Teil-3

Netztechnik-Vorlesung Teil-3
Inhalt

- Coding (Source-Coding / Channel-Coding / Wire-Coding)
- Fehler
- Modulation



Nachdem die AD-Wandlung stattgefunden hat gibt es mindestens 3 Bearbeitungsblöcke die vor einer Übertragung von Daten erforderlich sind:

Source-Coding:
Minimierung der Daten auf das absolute Minimum
Stichworte: Redundanz entfernen Relevanz herausarbeiten.

Channel-Coding:
Gezieltes Hinzufügen von redundanten Bits um Fehler zu erkennen und zu beheben.

Wire-Coding:
Einsatz von Codes die möglichst viele Bits gleichzeitig übertragen und eine Taktrückgewinnung sicher stellen.

Source Coding

Source Coding dient zur Reduzierung der zu transportierenden Datenmenge.
Eine Reduzierung von Speicherplatz ist eng verbunden mit weniger zu übertragenden Daten und damit eine Kostenfrage.
Andererseits ist die Codierung ein komplexes, aufwändiges Verfahren, das einen erheblichen technischen Aufwand bedeutet.

Beispiel:
Eine Datenquelle kennt 4 Symbole (A, B, C und D)
Die Wahrscheinlichkeit des Auftretens der Symbole ist:

P(A) = 0,5
P(B) = 0,25
P(C) = 0,125
P(D) = 0,125

Damit ist der durchschnittliche Informationsgehalt (die Entropie) H = 1,75 bit / Symbol

H = 0,5*ld(1/0,5) + 0,25*ld(1/0,25) + 2* (0,125*ld(1/0,125)) = 0,5+0,5+2*0,375 = 1,75bit/Symbol

Source Coding:
Beispiel

```
..0         1          2
..123456789012345678901234..
..AAACACABABBBABADAACBDAABD..
```

Eine erste, einfache Codierung könnte so aussehen:
A = 00
B = 01
C = 10
D = 11

Die obige Nachricht mit 24 Symbolen hat einen Informationsinhalt von 24 * 1,75 = 42 Bit.
Mit dem obigen Code erhält man jedoch 2 * 24 = 48 Bit. Dies bedeutet, dass 6 Bit redundant sind.

Ein besseres Ergebnis bringt eine weitere Codierung
A = 1
B = 01
C = 001
D = 000

Damit werden 42 Bit benötigt. Dies entspricht dem Informationsgehalt.
Damit ist die Redundanz eliminiert.

Fano Bedingung:
Wenn kein Codewort existiert, das der Anfang eines anderen Codewortes ist, dann ist jeder Satz von Codewörtern korrekt.

Wenn alle Symbole die gleiche Häufigkeit aufweisen, ist keine Reduzierung der Redundanz möglich.

Im obigen Fall treten die Symbole unterschiedlich oft auf:
A = 12
B = 6
C = 3
D = 3

Je häufiger ein Symbol auftritt desto weniger Bits bekommt es bei der Kodierung spendiert.

A = 1
B = 01
C = 001
D = 000

Source Coding:
Beispiel 2 (Morse-Code)

Symbol	Code	Symbol	Code	Symbol	Code	Symbol	Code
a	.-	l	.-..	w	.--	8	---..
b	-...	m	--	x	-..-	9	----.
c	-.-.	n	-.	y	-.--	0	-----
d	-..	o	---	z	--..	Punkt	.-.-.-
e	.	p	.--.	1	.----	Komma	--..--
f	..-.	q	--.-	2	..---	Fragezeichen	..--..
g	--.	r	.-.	3	...--	Doppelpunkt	---...
h	....	s	...	4	....-	Semikolon	-.-.-.
i	..	t	-	5	.....	Trennung	-...-
j	.---	u	..-	6	-....	Bruchstrich	-..-.
k	-.-	v	...-	7	--...	Anführungszeichen	.-..-.

Im deutschen (und englischen) Alphabet ist der Buchstabe „e“ häufiger als der Buchstabe „q“.

Deshalb hat der Buchstabe „e“ die Codierung „.“ und der Buchstabe „q“ die Codierung „--.-“

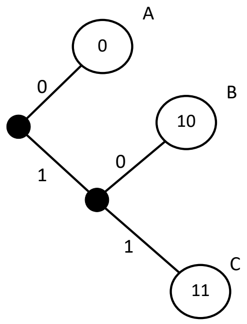
Siehe hierzu auch die Folie 59:63 in der Vorlesung Netztechnik Teil-2

Source Coding:
Fano-Bedingung

Fano Bedingung:
Wenn kein Codewort existiert, das der Anfang eines anderen Codewortes ist, dann ist jeder Satz von Codewörtern präfixfrei und damit korrekt.

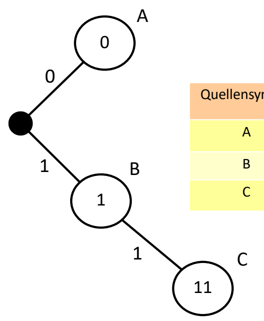
Ein D-närer päfixfreier Code mit den Codewortlängen (w1, w2, w3, … , wL) existiert genau dann, wenn die Kraft-Millan-Ungleichung erfüllt wird:

$$\sum_{i=1}^{L} D^{-wi} \leqslant 1$$



Quellensymbol	Codewort
A	0
B	10
C	11

Codebaum präfixfrei



Quellensymbol	Codewort
A	0
B	1
C	11

Codebaum nicht präfixfrei

Stand: 09.10.2022 Netztechnik Teil-3 Folie: 7:51

Den Beweis, ob ein Code die Fano-Bedingung erfüllt, kann über die Kraft-Millan-Ungleichung erbracht werden. Sie gilt auch für nicht binäre Codes.

In den obigen Beispielen sind Codes mit 3 Codewörtern mit variabler Länge.

Beim präfixfreien Codewort links, entsprechen alle Codewörter einem Endknoten.

Beim nicht präfixfreien Codewort rechts, kann man sehen, dass nicht alle Codewörter einem Endknoten entsprechen.

Wird auf die Codes die Kraft-Millan-Ungleichung angewandt, kann man sehen, dass im Fall des präfixfreien Codes mit den Längen für A=1, B=2 und C=2 :
$2^{-1} + 2^{-2} + 2^{-2} =$ 0,5 + 0,25 + 0,25 = 1 ≤ 1 erfüllt ist.

Im Fall des nicht präfixfreien Codes mit den Längen für A=1, B=1 und C=2
$2^{-1} + 2^{-1} + 2^{-2} =$ 0,5 + 0,5 + 0,25 = 1,25 ≤ 1 ist die Kraft-Millan-Ungleichung nicht erfüllt ist.

Source Coding:
Huffman-Codierung

Der Huffman-Code erfüllt die Fano-Bedingung und wird bei Codierungsalgorithmen wie JPEG verwendet.
Um einen Text zu codieren sind folgende Schritte zu durchlaufen:

1.
Die Symbole einer gegebenen Quelle (z. B. ASCII-Text) werden nach der Häufigkeit ihres Auftretens in einer Tabelle angeordnet.

2.
Die Symbole mit der niedrigsten Häufigkeit werden mit 0 und 1 codiert und in der Tabelle gekennzeichnet.

3.
Die beiden Symbole werden zu einem neuen Symbol zusammengefasst und die Wahrscheinlichkeit ihres Auftretens wird addiert. Da die beiden Symbole zusammengefasst wurden, ist die Tabelle um eine Zeile kürzer. Die Wahrscheinlichkeiten werden in der Tabelle berücksichtigt indem sie neu einsortiert werden. Die Schritte 2 und 3 werden so lange wiederholt, bis nur noch zwei Zeilen übrig sind.

4.
Man nimmt die letzte Tabelle und geht die Tabellen rückwärts bis zur ersten Tabelle durch und druckt den Code-Baum aus. Mit jeder Tabelle bekommt man eine Code-Entscheidung und zwei Zweige des Code Baumes. Man liest jetzt den Code-Baum vom Start bis zu jedem Knoten und erhält den Code für jedes Symbol.

Die Huffman-Codierung stellt einen Codierungsalgorithmus zur Verfügung der die Fano-Bedingung erfüllt.

Source-Coding:
Huffman-Codierung mit Tabellen-Beispiel 1

Es gibt 10 Symbole (x1, x2, .. x10) mit den folgenden Wahrscheinlichkeiten (P) ihres Auftretens

Symbol	:
x1	0,25
x2	0,15
X3	0,2
X4	0,2
X5	0,05
X6	0,07
X7	0,025
X8	0,02
X9	0,025
X10	0,01

Im Beispiel gibt es 10 Symbole (xi) mit unterschiedlichen Wahrscheinlichkeiten (P) des Auftretens.

Source-Coding:
Huffman-Codierung mit Tabellen-Beispiel Teil 2

Pos	Tabelle 1	Tabelle 2	Tabelle 3	Tabelle 4	Tabelle 5
1	X1 = 0,25	X1 = 0,25	X1 = 0,25	X1 = 0,25	X1 = 0,25
2	X3 = 0,2	X3 = 0,2	X3 = 0,2	X3 = 0,2	X3 = 0,2
3	X4 = 0,2	X4 = 0,2	X4 = 0,2	X4 = 0,2	X4 = 0,2
4	X2 = 0,15	X2 = 0,15	X2 = 0,15	X2 = 0,15	X2 = 0,15
5	X6 = 0,07	X6 = 0,07	X6 = 0,07	X7X9X8X10 = 0,08	X6X5 = 0,12 -> 0
6	X5 = 0,05	X5 = 0,05	X5 = 0,05	X6 = 0,07 -> 0	X7X9X8X10 = 0,08 → 1
7	X7 = 0,025	X8X10 = 0,03	X7X9 = 0,05 -> 0	X5 = 0,05 → 1	
8	X9 = 0,025	X7 = 0,025 -> 0	X8X10 = 0,03 -> 1		
9	X8 = 0,02 -> 0	X9 = 0,025 → 1			
10	X10 = 0,01 → 1				

Die Symbole werden nach ihrer Häufigkeit in die erste Spalte der obigen Tabelle einsortiert. Die Symbole mit der größten Häufigkeit zuerst (oben)

Danach werden im ersten Durchgang die beiden untersten Einträge mit einer 1 und einer 0 markiert und zu einem neuen Symbol, mit addierter Häufigkeit, zusammengefasst.

Das neue Symbol wird für den nächsten Durchgang in der nächsten Spalte der Tabelle an der Stelle mit der addierten Häufigkeit einsortiert.

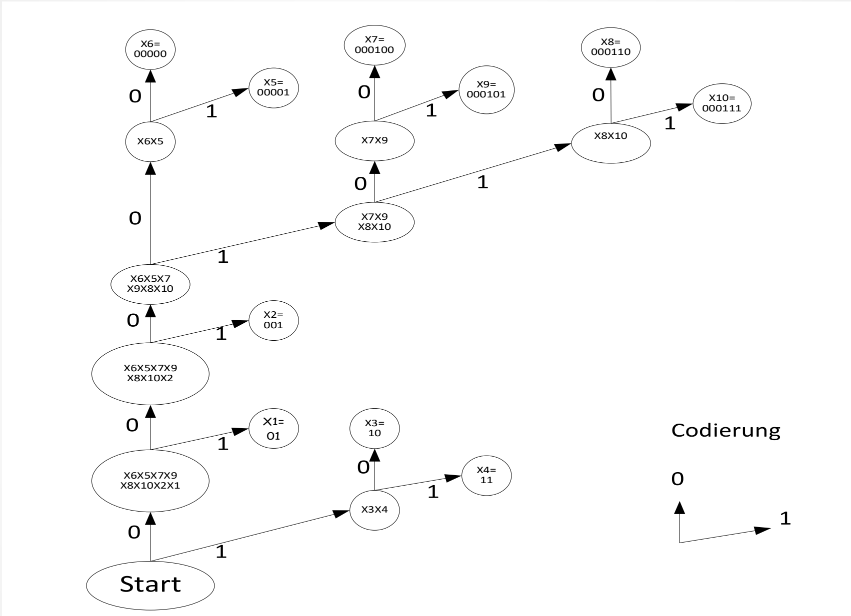
Source-Coding:
Huffman-Codierung Tabellen-Beispiel Teil 3

Pos	Tabelle 6	Tabelle 7	Tabelle 8	Tabelle 9
1	X1 = 0,25	X6X5X7X9X8X10X2 = 0,35	X3X4 = 0,4	X6X5X7X9X8X10X2X1 = 0,6 → 0
2	X3 = 0,2	X1 = 0,25	X6X5X7X9X8X10X2 = 0,35 -> 0	X3X4 = 0,4 → 1
3	X4 = 0,2	X3 = 0,2 → 0	X1 = 0,25 -> 1	
4	X6X5X7X9X8X10 = 0,2 -> 0	X4 = 0,2 -> 1		
5	X2 = 0,15 → 1			

Der Vorgang wird wiederholt bis nur noch 2 Einträge übrig sind.

Auch diese Einträge werden mit einer 0 und einer 1 gekennzeichnet.

Source-Coding:
Huffman-Codierung Tabellen-Beispiel Teil 4



Danach kann ein Baum aufgebaut werden.

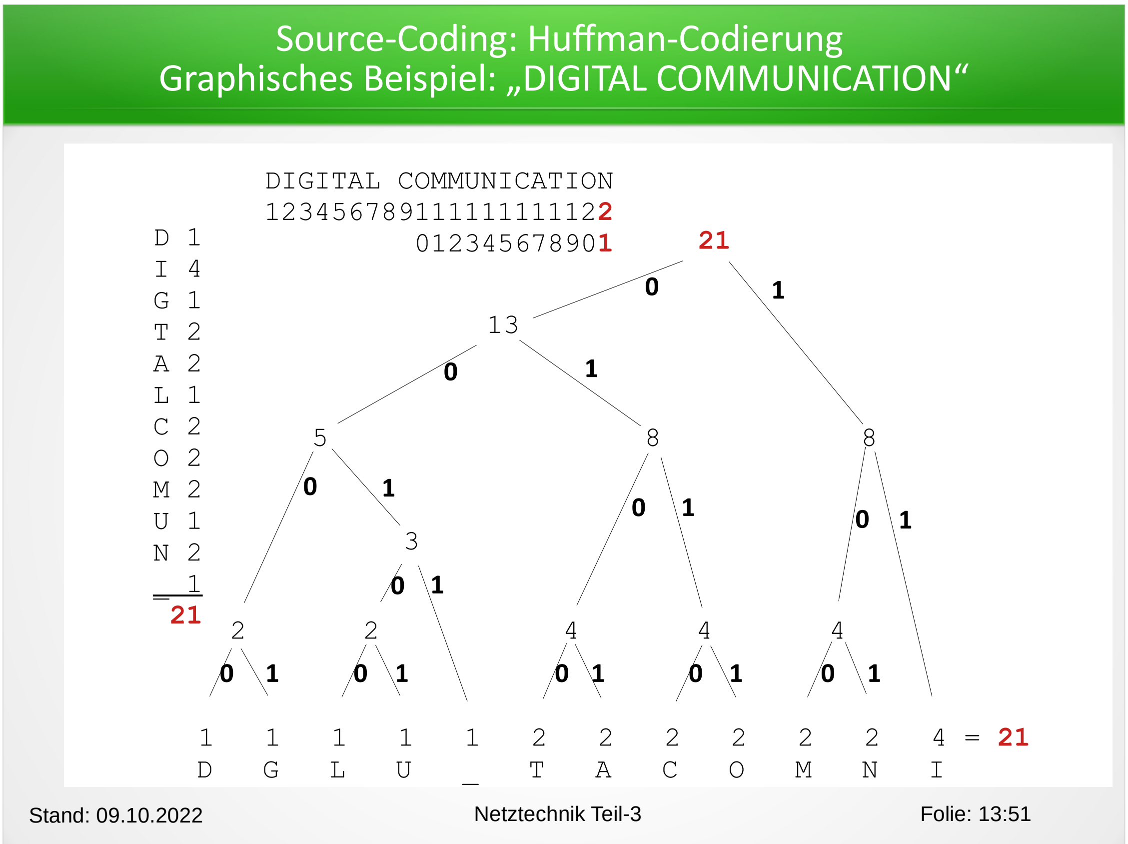
Ausgehend vom Start werden die beiden letzten Einträge als Knoten angehängt. Die Verbindung zu den Knoten mit der Eins werden nach rechts aufgezeichnet und mit einer Eins beschriftet.
Die Verbindung zum Knoten mit der Null wird nach oben gezeichnet und mit einer Null beschriftet.

Die Knoten stellen die Symbole dar, die entweder ein einzelnes Symbol beinhalten oder eine Zusammenfassung von Symbolen sind.

Ist es ein einzelnes Symbol, endet der Zweig an dieser Stelle.
Ist es eine Zusammenfassung, wird der Knoten wie oben weiter aufgelöst.

Am Ende steht der Baum mit allen Symbolen an den Enden fest.

Die Kodierung für ein Symbol erfolgt vom Start aus mit den Nullen und Einsen mit denen die Verbindungen beschriftet wurden.



Vorgehensweise:

1.
Zählen aller Symbole (auch die Leerzeichen werden mitgezählt) Summe = 21

2.
Ermitteln der Häufigkeit des Auftretens aller Symbole. Addieren der Häufigkeiten. Summe = 21

3.
Auflisten der Symbole beginnend mit den Symbolen die am seltensten auftreten und beschriften der Symbole mit der Anzahl des Auftretens. Die Anzahlen können addiert werden und ergeben wiederum 21.

4.
Zusammenfassen der Symbole mit der niedrigsten Häufigkeit beginnend zu einem neuen Symbol mit addierter Häufigkeit. Dabei sind Überschneidungen möglich. Sie fördern allerdings nicht die Übersicht und sind deshalb zu vermeiden.

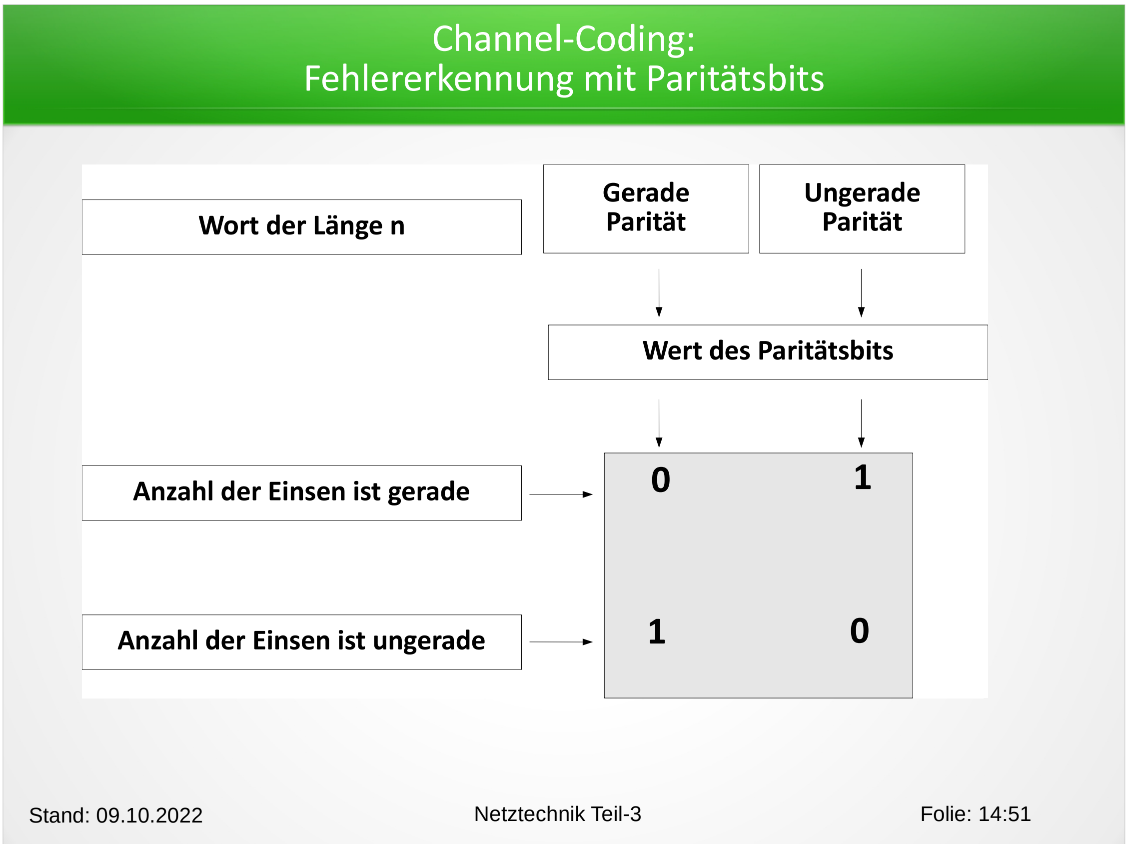
5.
Am Ende entsteht ein Knoten mit allen addierten Häufigkeiten = 21

6.
Bei allen neu entstandenen Knoten werden die Verbindungen die von links kommen mit 0 beschriftet und die Verbindungen die von rechts kommen mit 1.

7.
Die Kodierung eines Symbols kann von oben nach unten mit den Nullen und Einsen der Verbindungen vorgenommen werden. Z. B. G = 0001, I = 11

Auch hier bekommen die öfter vorkommenden Symbole eine kürzere Kodierung. Mit der Summenbildung kann für jeden Schritt überprüft werden, ob richtig gearbeitet wurde (Die Summe ist immer 21).

Je nach Sortierung der Symbole mit gleicher Wahrscheinlichkeit, können sich für die Symbole unterschiedliche Codierungen ergeben. Wichtig ist dabei nur dass auf der Sender und der Empfängerseite die gleiche Codierung erfolgt.



Nachdem die Information auf ein Minimum reduziert wurde, kann mit dem gezielten Hinzufügen von Paritätsbits erreicht werden, dass beim Empfänger Fehler erkannt und evtl. auch korrigiert werden können.

Dabei kann entweder mit einer geraden oder mit einer ungeraden Parität gearbeitet werden.

Die Daten werden in Worte der Länge n aufgeteilt.
Für jedes Wort wird die Anzahl der Einsen gezählt.

Gerade Parität bedeutet, dass das neu erzeugte Wort eine gerade Anzahl von Einsen aufweisen muss. Bei einer Ungerade Parität muss das neu erzeugte Wort eine ungerade Anzahl von Einsen aufweisen.,

Bei gerader Parität muss damit bei einer geraden Anzahl von Einsen eine Null angefügt werden da keine Änderung mehr erforderlich ist.

Bei einer ungeraden Parität muss folglich bei einer geraden Anzahl von Einsen eine Eins angefügt werden, da eine Änderung erforderlich ist.

Channel-Coding: Fehlererkennung mit Paritätsbits

Code ohne Parität		Code mit Parität	
Hexadezimal-Darstellung	Binär-Darstellung	Gerade Parität	Ungerade Parität
0	0000	00000	00001
1	0001	00011	00010
2	0010	00101	00100
3	0011	00110	00111
4	0100	01001	01000
5	0101	01010	01011
6	0110	01100	01101
7	0111	01111	01110
8	1000	10001	10000
9	1001	10010	10011
A	1010	10100	10101
B	1011	10111	10110
C	1100	11000	11001
D	1101	11011	11010
E	1110	11101	11100
F	1111	11110	11111

In der Tabelle sind die zugeordneten Codes für eine gerade oder ungerade Parität aufgeführt.

Channel-Coding:
Fehlererkennung (Hamming-Distanz)

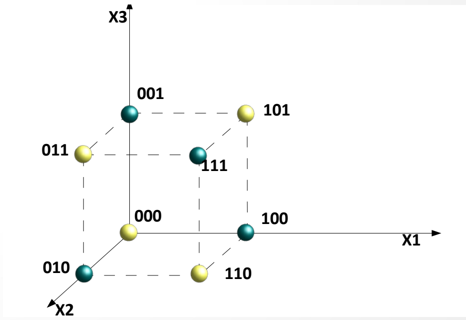
Die Distanz zweier Codewörter ist die Anzahl der Bits in denen sich die beiden Codewörter unterscheiden.
So haben die Codewörter 0000 und 1111 die Distanz 4.
Die Codewörter 0000 und 0001 haben die Distanz 1.
Die Hamming-Distanz ist der minimale Abstand aller möglichen Codewörter eines Codes.

Allgemein gilt:
Die Fehlererkennungs- und -korrektureigenschaften eines Codes hängen von seinem Hamming Abstand ab.

Zum Auffinden von d Fehlern benötigt man einen Hamming-Abstand von d + 1.
Zur Korrektur von d Fehlern braucht man einen Hamming-Abstand von 2d + 1.

Wird ein 2 Bit-Binärcode um ein Prüfbit erweitert, entsteht folgende Zuordnung:
2 Bit-Binärcode + Parity-Bit -> 3 Bit-Binärcode

00	0	000	\	
01	1	011	|	gültige Codes
10	1	101	|	
11	0	110	/	
		001	\	
		010	|	ungültige Codes
		100	|	
		111	/	

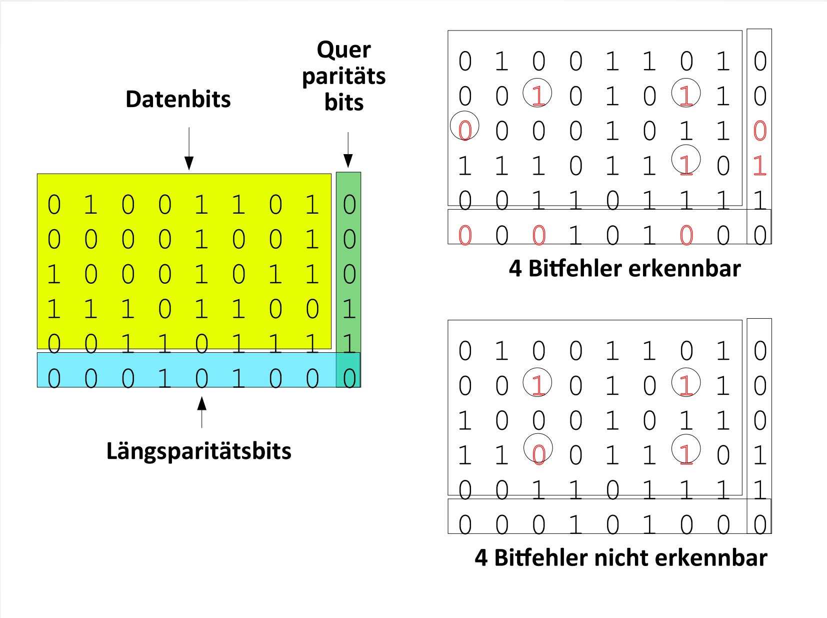


Abhängig von der Hamming-Distanz können bei der Übertragung Fehler erkannt oder sogar korrigiert werden

Die Hamming-Distanz zweier Codewörter ist die Anzahl der Bits, in denen sie sich unterscheiden.

Mit der obigen Vorgehensweise können Einzelbitfehler erkannt werden. Wenn ein einzelnes Bit kippt, entsteht ein ungültiger Code.

Channel-Coding:
Fehlererkennung mit 2-dimensionaler Parität



Beispiel einer 2-dimensionalen geraden Parität. Dabei werden für die Zeilen und die Spalten jeweils ein Paritätsbit hinzugefügt. (linke Tabelle)

Treten Bitfehler auf, können je nach Position bis zu 4 Bitfehler erkannt werden. (rechts oben)

Bei ungünstigen Positionen werden überhaupt keine Fehler erkannt, da sie sich gegenseitig aufheben. (rechts unten)

Channel-Coding:
Fehlererkennung mit CRC (Cyclic Redundancy Check)

Beispiel

Wir haben eine 8-Bit-Nachricht M. Ihr wert sei 10011010.
Das entspricht folgendem Polynom.
$M(x)= 1*x^7 + 0*x^6 + 0*x^5 + 1*x^4 + 1*x^3 + 0*x^2 + 1*x^1 + 0*x^0 = x^7 + x^4 + x^3 + x^1$

Das Generatorpolynom G sei vom Grad 3.
$G(x) = x^3 + x^2 + x^0$ In diesem Beispiel sei G(x) = 1101.

Da das Generatorpolynom vom Grad 3 ist, wird M(x) zuerst mit x^3 multipliziert denn .
Dies entspricht einer Verschiebung um 3 Stellen nach links.
Damit wird aus 10011010 der Wert 10011010000.
Danach wird M(x) durch G(x) dividiert, was einer blockweisen XOR-Operation entspricht.

```
Original-Polynom          10011010
Multiplikation mit X³     10011010000
Division durch            10011010000 / 1101   Generator-
Generatorpolynom          1101                 Polynom
                           1001
                           1101
                            1000
                            1101
                             1011
                             1101
                              1100
                              1101
                                 1000
                                 1101
                                  101 ---> Rest = 101
                                           (wird über die 3
                                           eingeschobenen
                                           Stellen geschrieben)
```

Der Rest mit dem Wert 101 wird mit $M(x)*x^3$ XOR-verknüpft.

Das Ergebnis lautet 10011010101.
Dieser Wert wird als N(x) gesendet.

```
Übertragenes-Polynom      10011010101
Division durch            10011010101 / 1101   Generator-
Generatorpolynom          1101                 Polynom
                           1001
                           1101
                            1000
                            1101
                             1011
                             1101
                              1100
                              1101
                                 1101
                                 1101
                                 0000   Rest = 0 !
                                        -> Kein Fehler bei
                                           der Übertragung
```


Auf der Ebene 2 wird der CRC Cyclic Redundancy Check) durchgeführt. Dabei werden die zu sichernden Daten um den Grad eines Generatorpolynoms nach links verschoben.

Danach wird über die gesamten Daten eine Division (XOR-Operation) mit dem Generatorpolynom durchgeführt.

Der Rest wird in die 3 Bytes am Ende der Daten geschrieben und übertragen.

Auf der Empfängerseite wird mit dem selben Generatorpolynom die gleiche XOR-Operation durchgeführt.

War bei der Übertragung kein Fehler aufgetreten, ist das Ergebnis der zweiten XOR-Operation = Null

Ist ein Fehler aufgetreten, bleibt bei der zweiten XOR-Operation ein Rest ungleich Null übrig. Damit ist der Frame zu verwerfen .

Channel-Coding: Fehlererkennung mit CRC (Cyclic Redundancy Check)

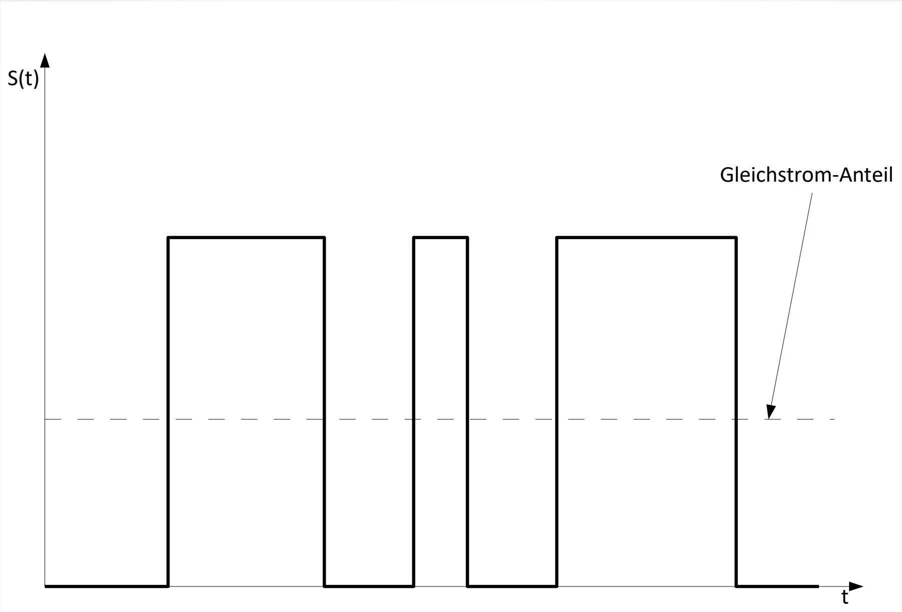
Übliche CRC-Generatorpolynome sind:

CRC	C(x) Generatorpolynom
CRC-8	$X^8 + x^2 + x^1 + 1$
CRC-10	$X^{10} + x^9 + x^5 + x^4 + x + 1$
CRC-12	$X^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$X^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$X^{16} + x^{12} + x^5 + 1$
CRC-32	$X^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Es existieren für unterschiedliche Übertragungssysteme, unterschiedliche Generatorpolynome.

Ethernet verwendet CRC-32

Wire-Coding: Gleichstromanteil



Stand: 09.10.2022 Netztechnik Teil-3 Folie: 20:51

Bei den Wirecodes ist darauf zu achten, dass der Gleichstromanteil so gering wie möglich ist da damit die Leitung unnötig belastet und damit erwärmt wird.

Wire-Coding: Grundvoraussetzungen

Möglichst kein Gleichstromanteil denn dadurch muss mehr Leistung übertragen werden.

Dies führt z. B. zu einer Erwärmung von Kupferleitungen.

Eine einfache Takt-Rückgewinnung sollte möglich sein.

Es sollte eine hohe Effizienz und Stabilität erreicht werden.

Es sind unterschiedliche Codierungsverfahren im Gebrauch:

Zweistufige (Bi-Phase) Leitungscodierungen können durch zwei Spannungen erzeugt werden.
Der Vorteil hierbei ist eine einfache Takt-Rückgewinnung

Dreistufige Leitungscodierung
Diese auch ternäre Codes genannten Wire-Codes haben 3 unterschiedliche Pegel. Z. B. MLT3.
Dazu werden 3 unterschiedliche Spannungen verwendet. +1, -1 und 0 Volt.

Vierstufige Leitungscodierung
Quarternäre Leitungscodes

Fünfstufige Leitungscodes
Quinäre Leitungscodes

Mehrwertige Leitungscodes werden da eingesetzt wo die Fundamentalfrequenz auf den Leitung reduziert werden soll. Dafür sind dann auf der Empfängerseite hochwertige Empfänger-Bausteine notwendig, denn diese Codes sind störanfälliger.

Stand: 09.10.2022 Netztechnik Teil-3 Folie: 21:51

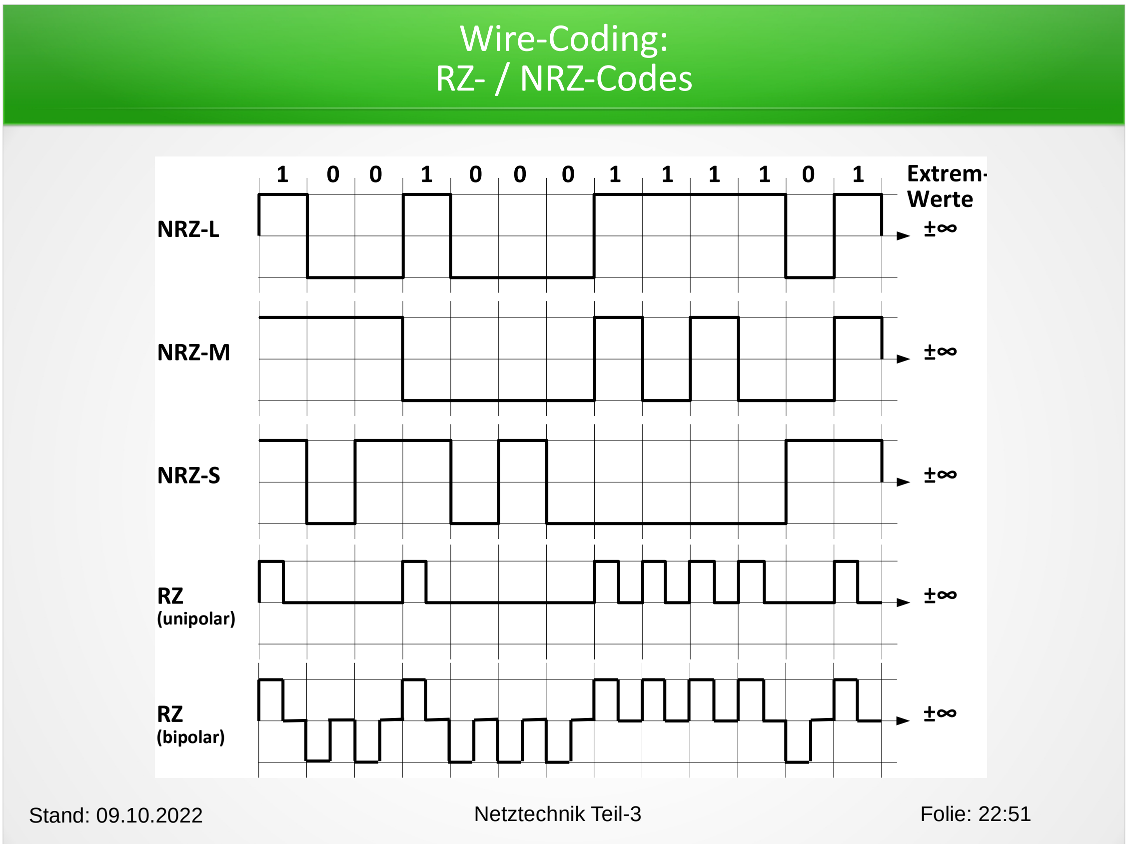
Hier geht es um die physikalische Darstellung von codierten Symbolen auf:

Kupferleitungen
Auf einer Kupferleitung können Nutzdaten durch Spannungsschwankungen übertragen werden.

LWL
Bei Lichtwellenleitern (LWL) kommen Lichtimpulse zu Einsatz.

Funk
Bei Funk werden Elektromagnetische Wellen übertragen.

Je mehr Stufen ein Code hat, desto mehr Bits können pro Takt übertragen werden.



RZ (Return to Zero)

Unipolar

Bei jeder 1 wird einen halben Takt lang ein +Pegel angelegt.

Bei 0 bleibt der Pegel auf 0

Bipolar

Bei jeder 1 wird einen halben Takt lang ein +Pegel angelegt.

Bei jeder 0 wird einen halben Takt lang ein -Pegel angelegt

NRZ (Non Return to Zero)

NRZ-Verfahren arbeiten nach dem Prinzip der

differentiellen Codierung.

D. h. nicht der Spannungspegel ist ausschlaggebend, sondern der Pegelwechsel.

Die NRZ-Codierung hat einen hohen Gleichspannungsanteil.

Im Extremfall ± ∞.

Diese Codierung wird bei Magnetaufzeichnungsverfahren verwendet.

Es besteht keine Möglichkeit zur Takt-Rückgewinnung.

NRZ-L

Wechsel von -Pegel auf +Pegel bedeutet 1.

Wechsel von +Pegel auf -Pegel bedeutet 0.

NRZ-M

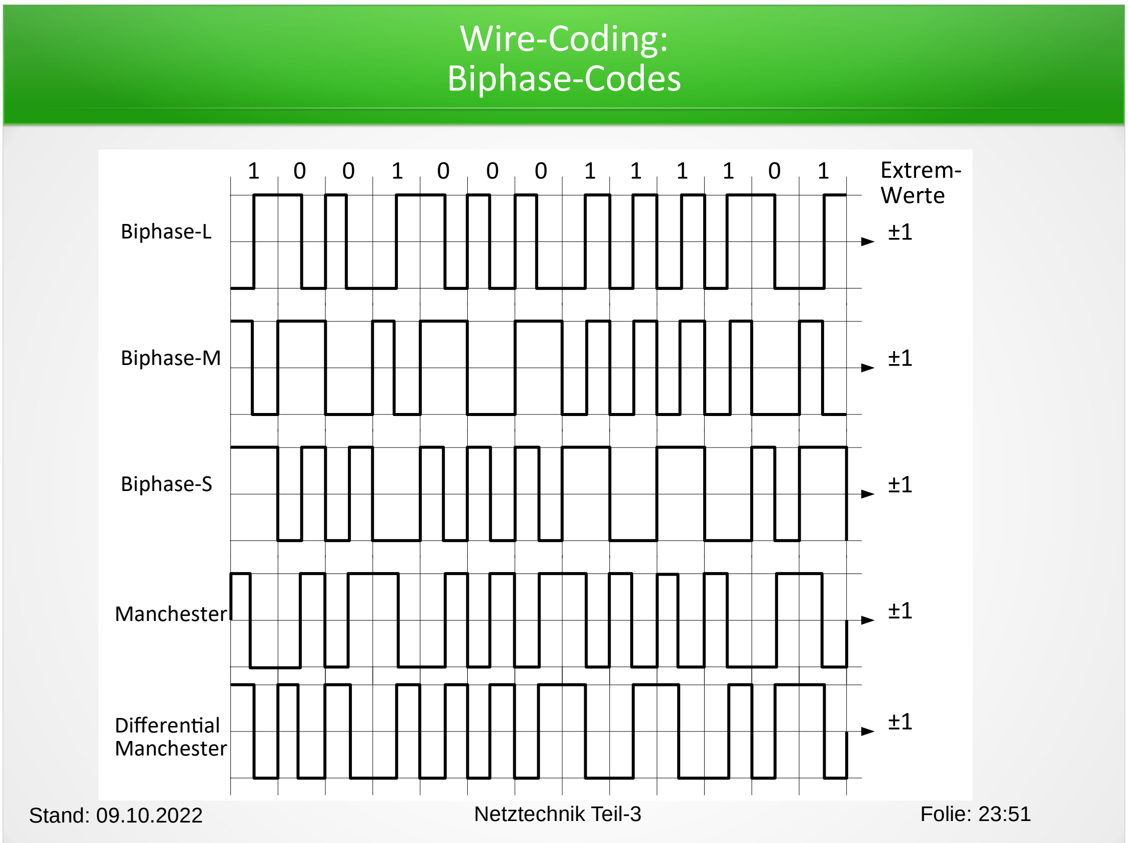
Wechsel des Pegel bedeutet 1.

Kein Pegelwechsel bedeutet 0.

NRZ-S

Wechsel des Pegel bedeutet 0.

Kein Pegelwechsel bedeutet 1.



Biphase-Codierung
Bei der Biphase-Codierung liegen folgende Prinzipien zugrunde:
Bei jedem Bit-Intervall findet ein Übergang statt.
Dadurch ist eine Taktrückgewinnung einfach möglich.
Einfachere Fehlererkennung. Das Ausbleiben eines Übergangs kann für die Fehlererkennung genutzt werden.
Kein Gleichstrom-Anteil.

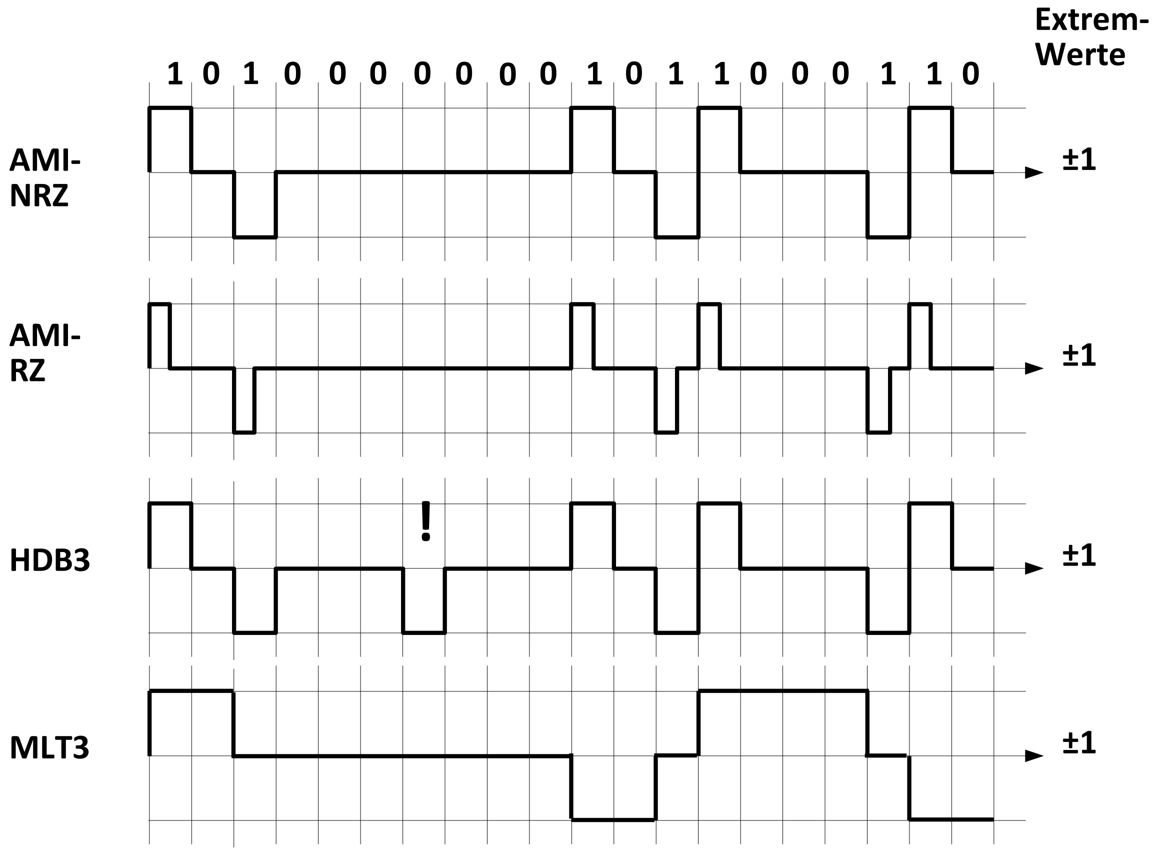
Biphase-L (Level)
Eine 0 wird durch eine, in der Mitte des Intervalls fallende, Flanke dargestellt.
Eine 1 wird durch eine, in der Mitte des Intervalls steigende, Flanke dargestellt.

Biphase-M (Mark)
Eine 1 wird durch eine Flanke in der Mitte des Intervalls gekennzeichnet.
Eine 0 wird durch das Fehlen einer Flanke in der Mitte des Intervalls gekennzeichnet.

Biphase-S (Space)
Eine 0 wird durch eine Flanke in der Mitte des Intervalls gekennzeichnet.
Eine 1 wird durch das Fehlen einer Flanke in der Mitte des Intervalls gekennzeichnet.

Manchester
Eine 1 wird durch eine hohe Spannung in der ersten Intervall-Hälfte dargestellt. Eine 0 wird durch eine hohe Spannung in der zweiten Intervall-Hälfte dargestellt.

Differential Manchester
Eine 1 wird dadurch angezeigt dass am Intervall-Anfang keine Änderung statt findet. Eine 0 wird dadurch angezeigt dass am Intervall-Anfang eine Änderung statt findet.



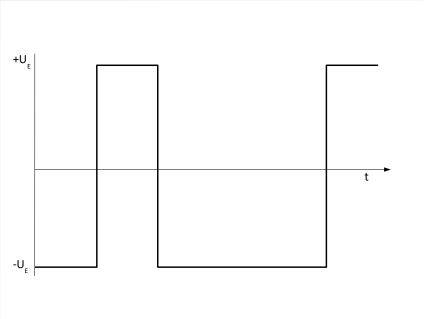
AMI-NRZ
Bei jeder 1 wird ein Impuls während des gesamten Intervalls im Wechsel mit + und – übertragen.
Bei 0 bleibt der Pegel auf 0.

AMI-RZ
Bei jeder 1 wird ein Impuls während eines halben Intervalls im Wechsel mit + und – übertragen.
Bei 0 bleibt der Pegel auf 0.

HDB3
Entspricht einer 1 einem AMI-NRZ
Bei einer 0 wird bei 4 Nullen in Folge der letzte Impuls wiederholt.
Dies entspricht einer Codeverletzung.

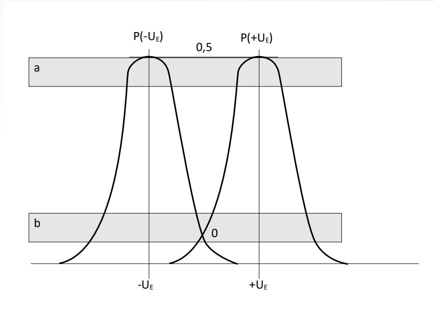
MLT3 (Multiple Level 3)
Hierbei wird zuerst eine Umkodierung von 2 auf 3 Pegel vorgenommen. Damit werden 3 anstelle von 2 Spannungspegeln ausgenutzt.
Dadurch reduziert sich bei FDDI die Fundamentalfrequenz von 62,5 MHz auf 31.25 MHz. Somit sind Leitungslängen bis zu 100 m möglich.
Bei jeder 1 wird ein Pegelwechsel von 0,+1,0,-1 vorgenommen.
Bei einer 0 bleibt der Pegel erhalten. Damit ist bei langen 0-Folgen keine Taktrückgewinnung möglich! Deshalb muss bei 100BASE-TX , also Fastethernet auf Twisted-Pair-Leitungen, vorher erst eine 4B/5B-Codierung erfolgen.

Bitfehler



Ein ideales Signal

Durch Störungen wie z. B. Rauschen wird die Wahrscheinlichkeit, dass das Signal so aussieht folgendermaßen aussehen.



a
Es ist sehr selten ein Signal zu messen, das weit vom Wert $+U_E$ oder $-U_E$ entfernt ist.
b
Für ein rauschbehaftetes Signal ist es wahrscheinlich in der engen Umgebung von $+U_E$ oder $-U_E$ zu messen.
Bei einem Schwellwert von 0 kann ein Wert $+U_E$ nicht von einem Wert $-U_E$ unterschieden werden.
Dies ist der Bereich für Bitfehler.

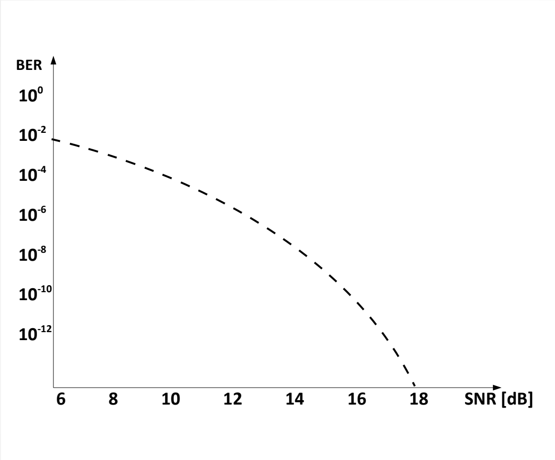
Weitere Gründe sind :
- Ein-/Ausschalten von Geräten (Spannungsspitzen)
- Übersprechen zwischen Adernpaaren
- Reflexionen bei schlecht abgeschlossenen Leitungen
- Lichtbögen durch vorbeifahrende Züge
- Defekte Leitungen

Je mehr Stufen ein Wirecode hat desto kleiner werden die Unterschiede zwischen U- und U+. (links)
Damit wirken sich Störungen immer gravierender aus.

Im rechten Bild ist bei a die Wahrscheinlichkeit groß, dass ein Signal sich entweder bei U+ oder U- aufhält. Damit ist ein Symbol gut erkennbar.

Je größer die Störungen werden, desto eher kann beim Empfänger das Signal bei b gemessen werden.
Je weiter die Wahrscheinlichkeit nach unten wandert desto kleiner wird der Unterschied zwischen den Symbolen.
Ab einer Wahrscheinlichkeit von 0 ist keine Unterscheidung mehr möglich und es treten Bitfehler auf.

Bitfehlerrate-2



$$BER = \frac{Anzahl\ der\ Bitfehler}{Anzahl\ aller\ übertragenen\ Bits}$$

Wenn ein Bitfehler auftritt, ist es wahrscheinlich, dass ein weiterer Bitfehler auftritt.

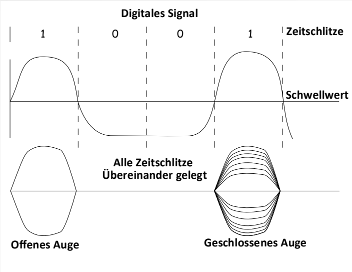
Bitfehler haben ein Burst-Verhalten.
Dies bedeutet, dass sie gehäuft auftreten.

$$SNR = \frac{Nutzsignal-Leistung}{Rausch-Leistung} = \frac{P_N}{P_R} = 10\log\left(\frac{P_N}{P_R}\right)[dB]$$

Je größer das SNR (Signal to Noise Ratio), desto weniger Bit-Fehler treten auf.

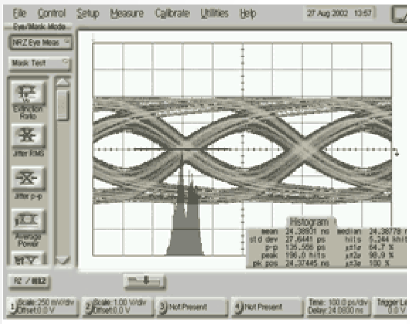
Das Signal-Rausch-Verhältnis auch Störabstand oder Signal-to-Noise-Ratio (SNR) ist ein Maß für die Qualität eines Nutzsignals das in einem Rausch Signal eingebettet ist.

Augenmuster

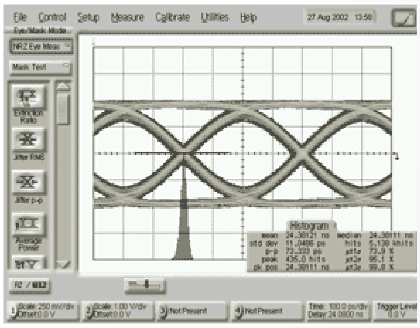


Die Qualität einer Übertragung kann auch mit den Augenmustern dargestellt werden.

Werden die Zeitschlitze übereinandergelegt, erhält man das typische Augenmuster.



Geschlossenes Auge bei einer zweistufigen Leitungscodierung



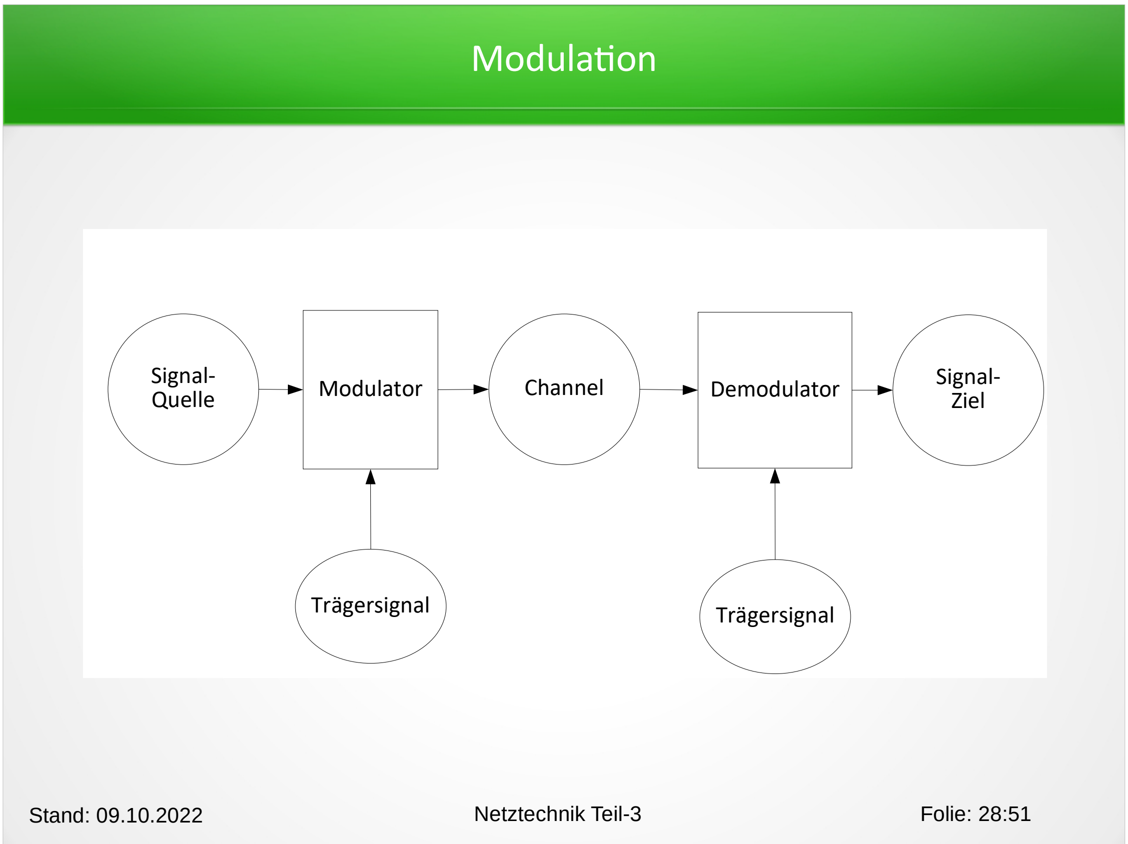
Offenes Auge bei einer zweistufigen Leitungscodierung

Die Signalqualität kann auch mit einem so genannten Augenmuster dargestellt werden.

Dabei werden die Takte eines Signals übereinandergelegt. Durch die unterschiedlichen Signalverläufe der einzelnen Takte ergibt sich eine verschmierte Darstellung.

Je größer die Störungen sind, desto verwischter ist der Signalverlauf und der Unterschied zwischen einer Eins (oben) und einer Null (unten) wird immer Kleiner.

Durch die überlagerte Darstellung der Takte sehen die Takte aus wie Augen. Je geschlossener das Auge ist, desto schlechter ist das Signal zu Rauschverhältnis. (links = schlecht / rechts = gut)



Zuerst wird ein Trägersignal erzeugt, das über den Kanal leicht übertragen werden kann.

Dieses Trägersignal wird mit den zu übertragenden Daten manipuliert (moduliert). Dadurch werden die Daten mit dem Trägersignal als Transportmedium übertragen.

Auf der Empfängerseite wird das modulierte Trägersignal empfangen und mit dem ursprünglichen Trägersignal wieder demoduliert.

Damit können die Daten aus dem übertragenen Signal wieder extrahiert und dem Ziel übergeben werden.

Modulation: Möglichkeiten-1

Für **analoge Signale** werden folgende Parameter des Trägersignals beeinflusst:

Amplitude
Eine Beeinflussung der Amplitude wird Amplitudenmodulation genannt.

Frequenz
Wird die Frequenz durch das zu übertragende Signal beeinflusst, spricht man von Frequenzmodulation.

Phase
Eine Veränderung der Phase des Trägersignals wird Phasenmodulation genannt.

Bei **digitalen zu übertragenden Signalen** wird das Trägersignal abrupt geändert. Deshalb spricht man hier auch von einer Umtastung. Dabei gilt:

Amplitude
Eine Beeinflussung der Amplitude wird Amplitudenumtastung (engl. Amplitude Shift Keying) genannt.

Frequenz
Wird die Frequenz durch das zu übertragende Signal beeinflusst, spricht man von Frequenzumtastung (engl. Frequency Shift Keying).

Phase
Eine Veränderung der Phase des Träger-Signals wird Phasenumtastung (engl. Phase Shift Keying) genannt.

Stand: 09.10.2022 Netztechnik Teil-3 Folie: 29:51

Als Trägersignal werden Sinusschwingungen (Sinusträger) oder Pulsträger verwendet.

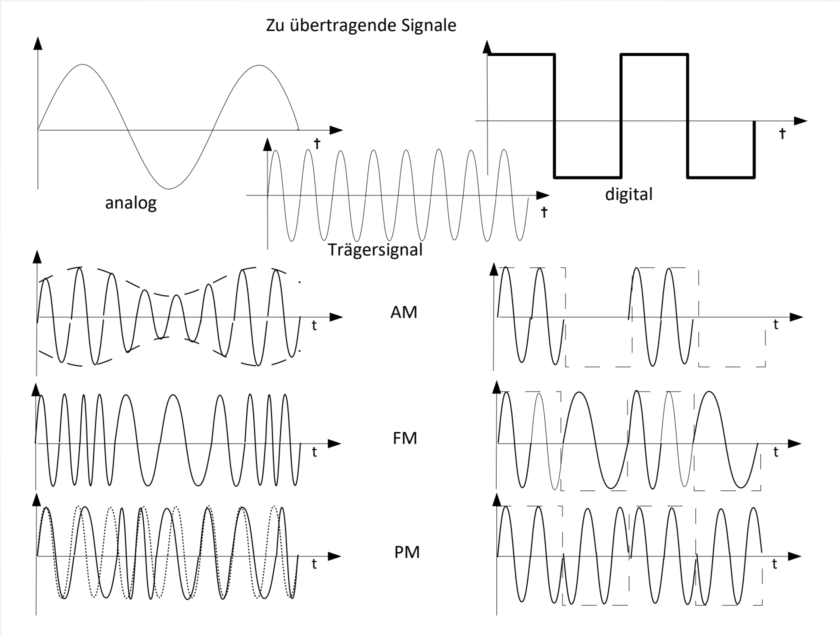
Ein Sinus-Trägersignal s(t) hat folgende Zusammensetzung:

$$s(t) = a(t)\cos(2\pi f(t)+\varphi(t)) = a(t)\cos(\omega(t)+\varphi(t))$$

Dieses Trägersignal kann in den folgenden Parametern moduliert werden.:
- Amplitude (a)
- Frequenz (f) Wobei 2πf der Kreisfrequenz ω des Trägers entspricht.
- Phase (φ)

Bei einem Pulsträger kommt noch die Pulsweite als modulierbarer Parameter Dazu.

Modulation: Möglichkeiten-2

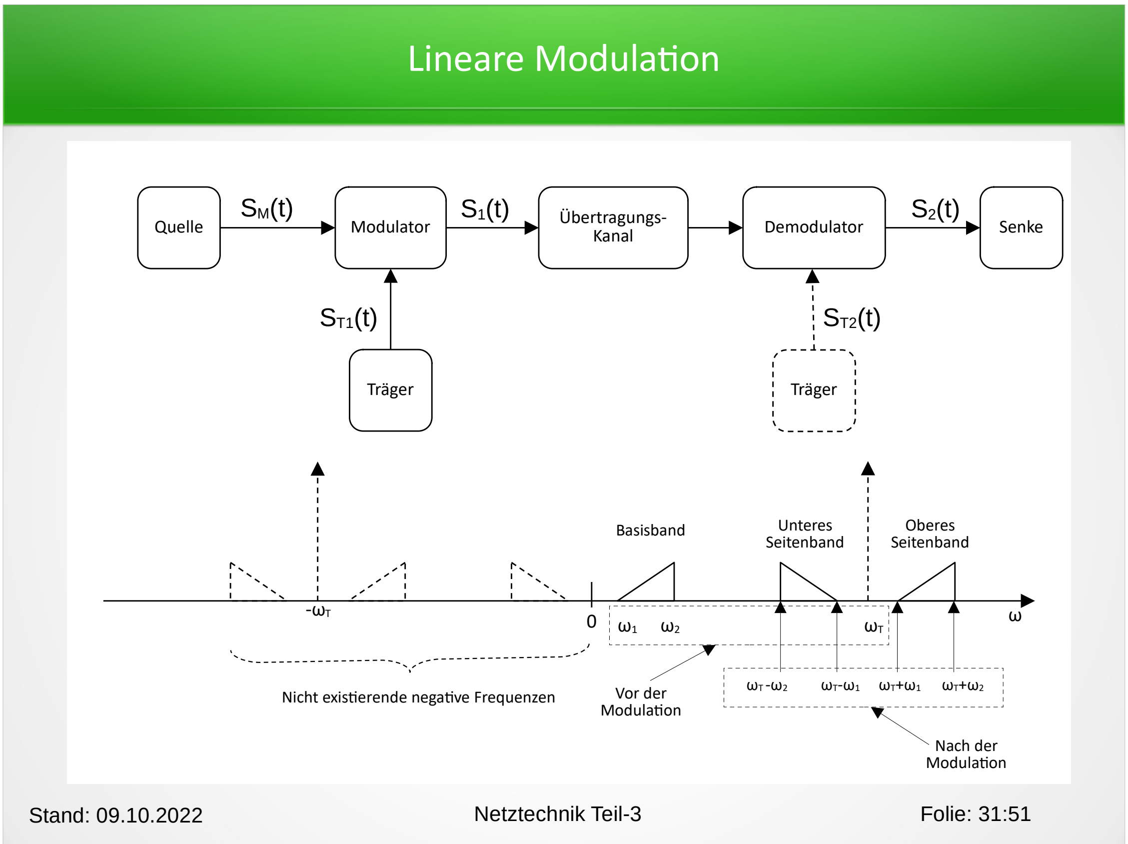


Stand: 09.10.2022 Netztechnik Teil-3 Folie: 30:51

Mit einem sinusförmigen Trägersignal (oben in der Mitte) können sowohl analoge als auch digitale Signale übertragen zu werden.

Auf der linken Seite wird das Trägersignal in der Amplitude, der Frequenz und der Phase **analog** moduliert.

Auf der rechten Seite wird das Trägersignal in der Amplitude, der Frequenz und der Phase **digital** moduliert.



In der Folie ist oben das von der Quelle erzeugte modulierende Signal **S_M(t)** zu sehen. Es wird **Basisband-Signa**l genannt. In ihm ist die zu übertragende Information enthalten.

Das modulierte Trägersignal S_T(t) ist ein Sinus- oder Pulsträger. Bei den Sinusträgern können die Amplitude die Frequenz oder die Phase moduliert werden. Bei den Pulsträgern können die Amplitude, die Frequenz, die Phase, oder die Pulsweite(Dauer) moduliert werden.

Unten ist der Frequenzbereich zu sehen. Ausgehend von der Frequenz ω = 0, sind auf der rechten Seite vor der Modulation nur die beiden Signale (Basisband (ω_1 bis ω_2) und Trägersignal (ω_T)) vorhanden.

Nach der Modulation gibt es zusätzlich die beiden Seitenbänder (unteres (ω_T – ω_2 bis ω_T - ω_1) und oberes (ω_T + ω_1 bis ω_{T+} ω_2) Seitenband). Zu mathematischen Zwecken wurden noch die negativen Frequenzen eingeführt. Da sie oft in der Literatur auftreten wurden sie hier der Vollständigkeit halber gestrichelt dargestellt. In der Realität existieren sie jedoch nicht und werden ab hier auch nicht weiter verwendet.

Im vorliegenden Abschnitt soll der folgende Sinusträger verwendet werden.

Mathematische Betrachtung

(1) $s_T(t)=\hat{S}_T\cos(\omega_T(t)+\varphi_T)$ Sinusträger

(2) $s_M(t)=\hat{s}_M(t)\cos(\omega_M t+\varphi_M)$ Modulierendes Signal

(3) $s_1(t)=k_1\cdot S_M(t)\cdot\hat{S}_T\cos(\omega_T t+\varphi_T)$ Modulationsprodukt

(4) $\cos\alpha\cdot\cos\beta = \frac{1}{2}[\cos(\alpha+\beta)+\cos(\alpha-\beta)]$ Trigonomische Beziehung

(5) $s_1(t)=\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T\cdot\cos(\omega_T+\omega_M)+(\varphi_T+\varphi_M)+\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T\cdot\cos(\omega_T-\omega_M)+(\varphi_T-\varphi_M)$ Modulations-produkt

(6) $\varphi_T = \varphi_M = 0$ $\hat{S}_1=\frac{1}{2}k_1\cdot\hat{S}_M\cdot\hat{S}_T$ Vereinfachungen

(7) $S_1(t)=\hat{S}_1\cdot\cos(\omega_T+\omega_M)+\hat{S}_1\cdot\cos(\omega_T-\omega_M)$ Vereinfachtes Modulationsprodukt

Stand: 09.10.2022 Netztechnik Teil-3 Folie: 32:51

In den ersten zwei Zeilen sind die verwendeten Signale (Trägersignal $S_T(t)$ und Modulierendes Signal $S_M(t)$) aufgeführt.

Das Ergebnis der Modulation in Formel (3) ist das Modulationsprodukt $S_1(t)$. Dabei ist der Faktor K_1 eine Modulationskonstante.

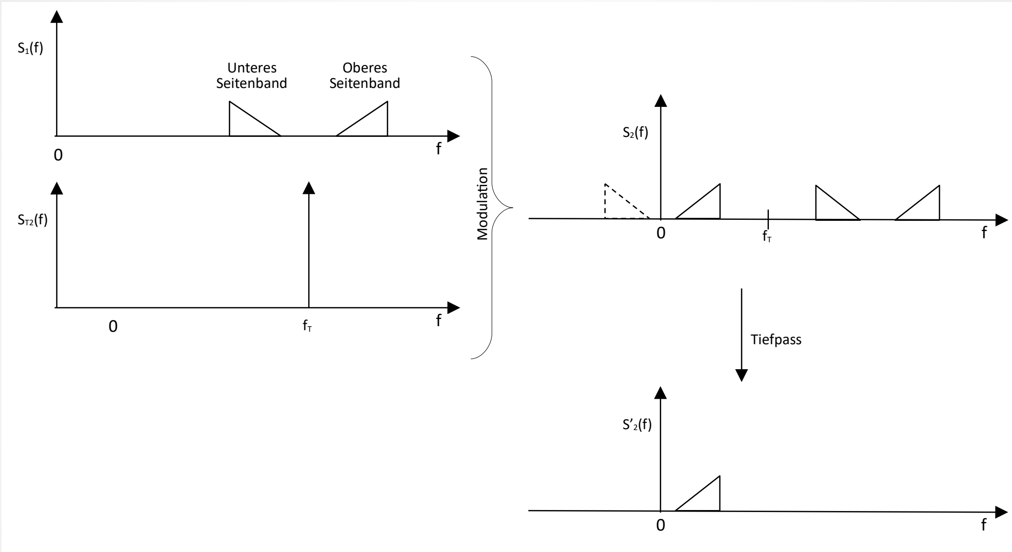
Mit der trigionometrischen Beziehung für die Multiplikation für Cosinus Signale in Formel (4) erhält man das Modulationsprodukt in Formel (5)

Die Vereinfachungen (6) führen zum vereinfachten Modulationsprodukt in Formel (7)

Die Kreisfrequenz ω_M steht stellvertretend für die Kreisfrequenzen eines von ω_1 bis ω_2 begrenzten Basisbandes. Entsprechend Gleichung (7) erzeugt die Kreisfequenz ω_M des Basisbandes eine obere Seitenbandfrequenz ($\omega_T + \omega_M$) und eine untere Seitenbandfrequenz ($\omega_T + \omega_M$). Jedes der beiden Seitenbänder hängt linear mit dem Basisband zusammen.

Zusammen mit den nicht existierenden negativen Frequenzen kann man erkennen, dass sich eine lineare Modulation als einfache Anwendung des Frequenz Verschiebungssatzes anwenden lässt.

Demodulation



Auf der Empfängerseite erfolgt die Demodulation durch eine weitere Modulation mit einem Träger der selben Frequenz und Phase, wie auf der Senderseite. Die Demodulation mit Hilfe des frequenz- und phasenrichtigen Trägers nennt man kohärente Modulation oder Synchronmodulation. Wird der Träger nicht mitübertragen, lässt er sich aus den beiden Seitenbändern erzeugen.

Daraus resultiert eine weitere Frequenzverschiebung und das ursprüngliche Basisband kann durch einen Tiefpassfilter ausgefiltert werden.

Da die bei den Seitenbänder die gleiche Information beinhalten kann durch Ausfiltern eines Seitenbandes Bandbreite und somit auch Übertragungsenergie eingespart werden. Dann wird von einer Single-Side Band-Modulation (SSB) gesprochen.

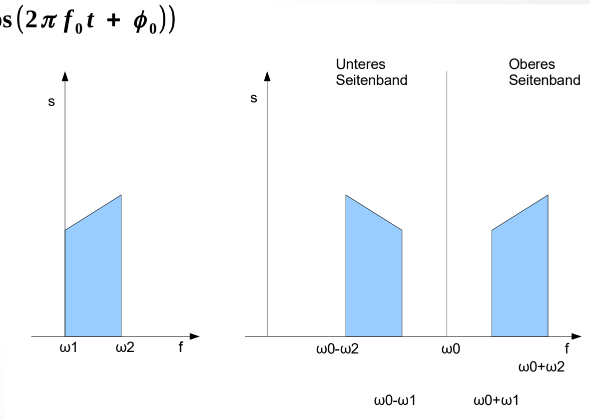
Modulation: Amplitudenmodulation

Hierbei wird die Amplitude des Träger-Signals $S_C(t)$ mit dem zu übertragenden Signal moduliert.

$$S_C(t) = (a_0 \cdot \cos(2\pi f_0 t + \phi_0))$$

Das Amplitudenmodulierte Signal S_{AM} ergibt sich zu:

$$S_{AM}(t)=(a_0 + a1 \cdot x(t) \cdot \cos(2\pi f_0 t + \phi_0))$$



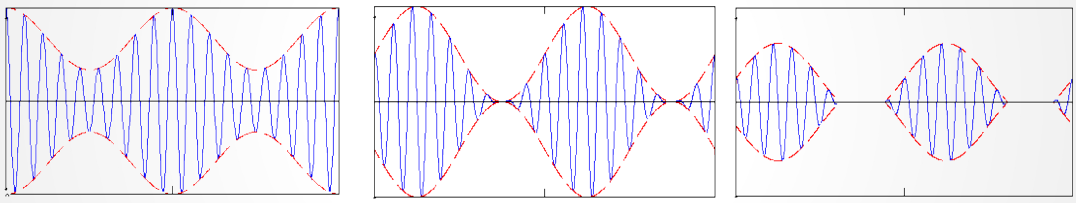
Links ist das Originalsignal und rechts das modulierte Signal im Frequenzbereich dargestellt.

Das modulierte Signal wird ober und unterhalb der Trägerfrequenz ω_0 in Form eines oberen und unteren Seitenbandes gesendet.

Durch Filterung eines Seitenbandes kann Bandbreite und somit auch Übertragungsenergie eingespart werden.
Dann wird von einer SSB- Modulation (Single-Side-Band) gesprochen.

Modulationsindex / Modulationsgrad

$m = \frac{a_1}{a_0}$ Der Modulationsindex errechnet sich aus der vorigen Formel



Modulationsindex M = 0,5	Modulationsindex M = 1	Modulationsindex M = 1,5
50% Modulation	100% Modulation	Übermodulation

Bei der Amplitudenmodulation ist es anzustreben, den Modulationsindex von m = 1 zu realisieren. Dies entspricht einer 100%-Modulation.
Bei dieser Modulation verschwindet der unmodulierte Träger. Dies wird als Zweiseitenband-AM ohne Träger bezeichnet.

Bei einer Abweichung nach unten (m<1) spricht man von einer Untermodulation. Dabei nutzt man die mögliche Amplitude nicht aus.
Das AM-modulierte Signal enthält hierbei die beiden Seitenträger und den unmodulierten Träger. Dies wird als Zweiseitenband-AM mit Träger bezeichnet.

Bei einer Abweichung nach oben (m > 1) spricht man von einer Übermodulation. Dabei geht Information verloren.

Die Amplitudenmodulation ist empfindlich gegen Rauschen!

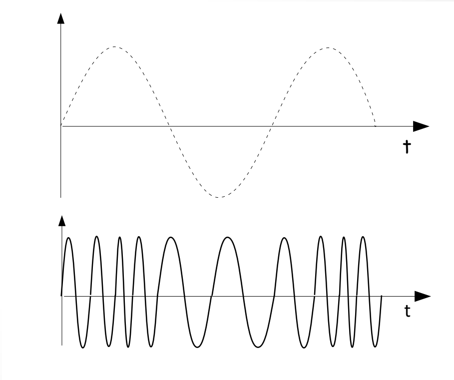
Modulation: Frequenzmodulation (FM)

Die Momentankreisfrequenz $\omega(t) = 2\pi f(t)$ der Trägerschwingung wird mit dem Quellensignal $x(t)$ beeinflusst.

$$\omega(t) = \omega_0 + \Delta\Omega \cdot x(t)$$

Das FM-Signal kann vereinfacht folgendermaßen beschrieben werden.

$$S_{FM}(t) = a_0 \cdot \cos(\Phi(t) + \varphi_0)$$



Der Kurvenverlauf oben zeigt das modulierende Signal des Basisbandes.

Darunter ist das frequenz-modulierte Signal dargestellt.

Während die Frequenzmodulation (FM) bei der Übertragung von Musik im UKW-Bereich noch genutzt wird, hat sie für die Übertragung von digitalen Signale kaum eine Bedeutung.

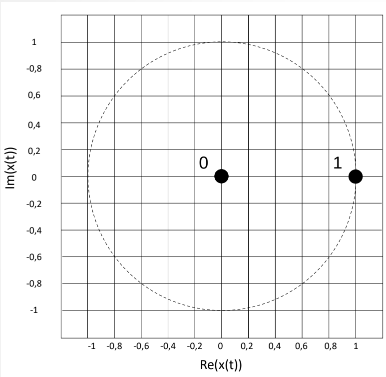
Binäre Modulationsverfahren

Wie bereits bei den Modulationsverfahren beschrieben wird die Modulation - da nur Einsen oder Nullen übertragen werden - auch Umtastung (eng. shift keying) genannt.

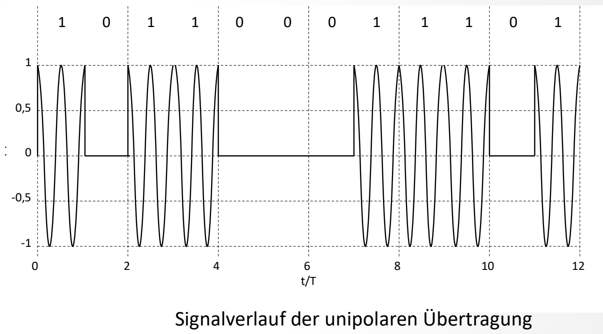
Wird nur ein Bit übertragen, muss das übertragene Signal nur 2 Zustände ($x_0(t)$ und $x_1(t)$) annehmen.

In den folgenden Beispielen soll immer die gleiche Folge an Informationsbits ((101100011101) übertragen werden.

Unipolare Modulation



Komplexe Ebene bei der unipolaren Übertragung



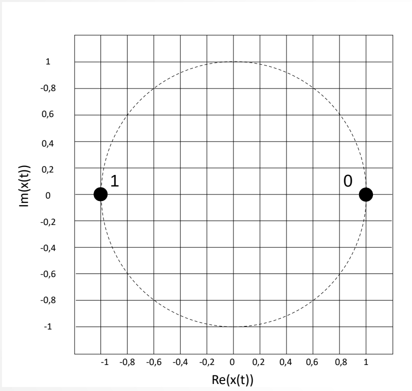
Signalverlauf der unipolaren Übertragung

Bei der unipolaren Modulation gilt $x_0(t) = 0$ und $x_1(t) = x(t)$.
Dabei entsteht entweder kein Signal ($x_0(t) = 0$), oder nur ein Signal mit einen Realteil ohne einen Imaginäranteil ($x_1(t) = x(t)$).

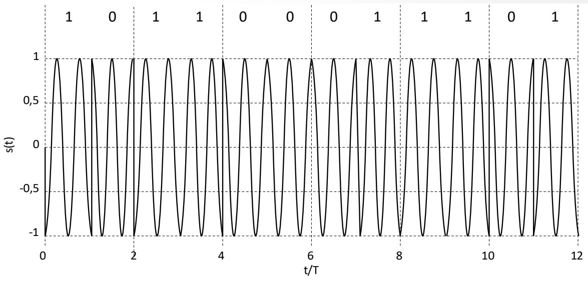
Wie beim Signalverlauf zu sehen ist wird bei mit jedem Signal nur ein Bit übertragen. Das Ausbleiben eines Signals bedeutet eine Null. Das übertragen eines Cosinus-Signals bedeutet eine Eins.

Hierbei wird sichtbar, dass es beim Senden langer Nullfolgen problematisch wird, den Takt auf der Empfängerseite zurückzugewinnen.

Bipolare Modulation



Komplexe Ebene bei der bipolaren Übertragung



Signalverlauf der bipolaren Ubertragung

Bei der bipolaren Modulation gilt $x_0(t) = x(t)$ und $x_1(t) = -x(t)$. Dabei entsteht entweder das Signal $x(t) = 0$, oder nur das Signal $-x(t) = 1$. Beide Signale haben nur einen Realteil ohne einen Imaginäranteil.

In den obigen Darstellungen wird eine Binary Phase Shift Keying-Modulation (BPSK) gezeigt. Die Amplitude und die Frequenz bleiben gleich nur die Phase ändert sich.

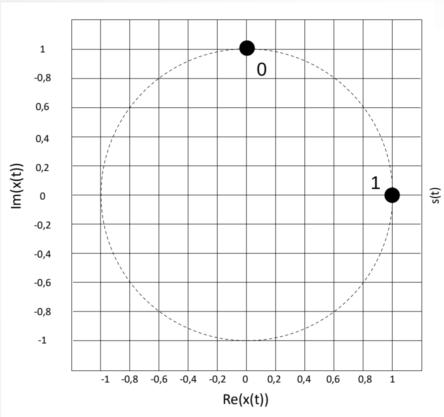
Eine Null wird mit einem Cosinus-Signal übertragen. Eine Eins wird mit einem negierten Cosinus-Signal übertragen. Der Wechsel von einer Null zu einer Eins, und umgekehrt, bedeutet somit einen Phasensprung von 180°.

Auch hier wird mit jedem Signal nur ein Bit übertragen. Lange Folgen von Nullen oder Einsen sind hier jedoch kein Problem mehr.

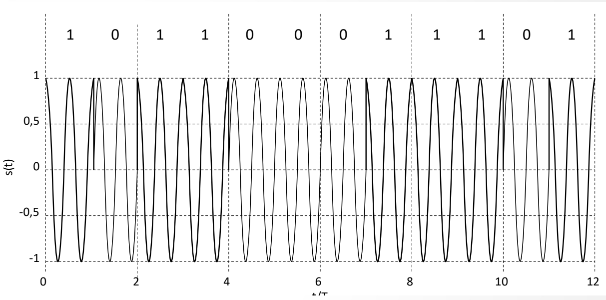
Der größere Abstand der beiden Signale führt zu einem größeren Fehlerabstand, was die Übertragung weniger anfällig auf Störeinflüsse macht.

Dies ist der Grund dafür, dass die BPSK häufig angewendet wird.

Orthogonale Modulation



Komplexe Ebene bei der orthogonalen Übertragung



Signalverlauf einer orthogonalen Übertragung

Stand: 09.10.2022 Netztechnik Teil-3 Folie: 40:51

Bei der orthogonalen Modulation gilt ‹s0(t), s1(t)› = 0. Die Null hat nur einen Imaginärteil und die Eins hat nur einen Realteil. Zwischen den beiden Signalen besteht eine Phasenverschiebung von π/2.
Damit stehen die Signale orthogonal aufeinander.

Im Beispiel ist $x_1(t)$ = rect (t/T). $x_0(t)$ = rect (t/T) · e $^{+j\pi/2}$ ist ein komplexwertiges Signal. Das Signal für S_0

$$S_0(t)=\Re\{x_0(t)\cdot e^{j2\pi f_0 t}\}=\Re\{rect(\frac{t}{T})\cdot e^{j(2\pi f_0 t\ +\ \pi/2)}\}=rect(\frac{t}{T})\cdot\cos(2\pi f_0 t\ +\ \frac{\pi}{2})$$

Ist damit orthogonal zum Signal S1

$$S_1(t)=\Re\{x_1(t)\cdot e^{j2\pi f_0 t}\}=\Re\{rect(\frac{t}{T})\cdot e^{j2\pi f_0 t}\}=rect(\frac{t}{T})\cdot\cos(2\pi f_0 t)$$

Mehrwertige Modulationsverfahren

Um mehr als nur ein Bit pro Symbol zu übertragen, können mehrwertige Modulationsverfahren verwendet werden.

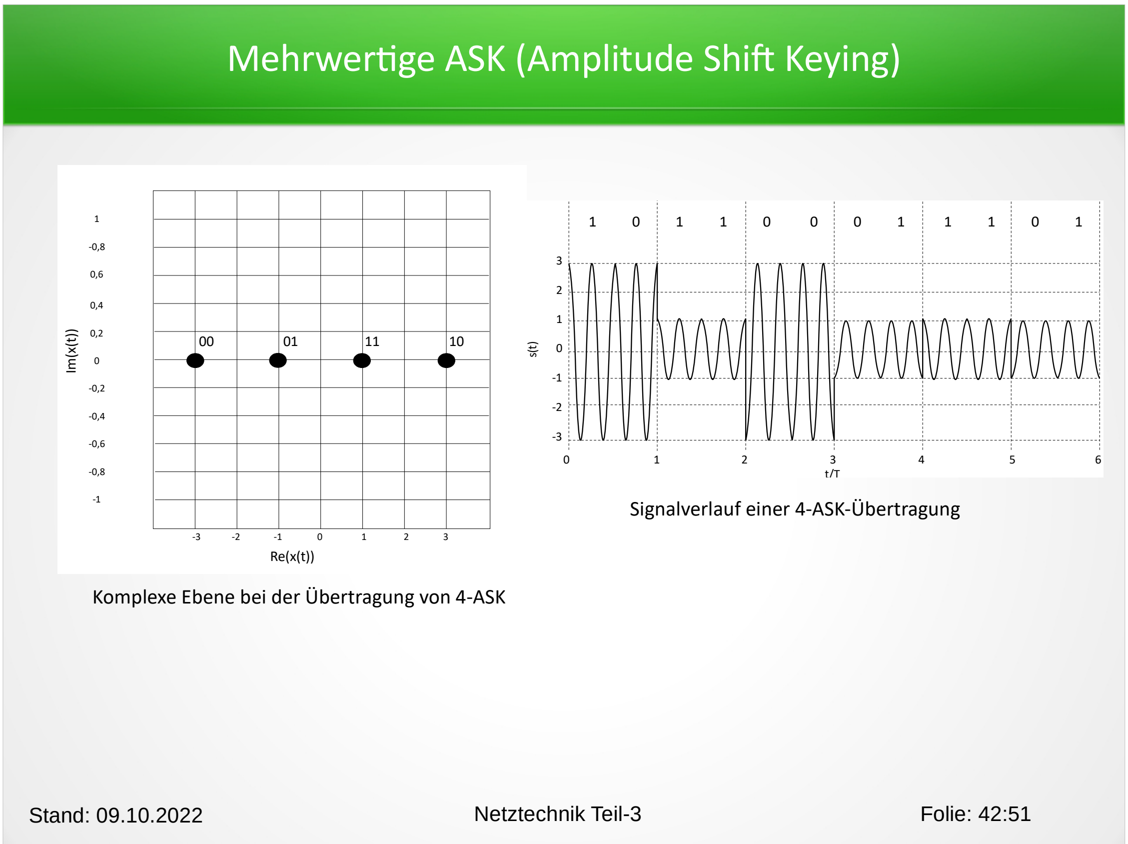
Ein Unterscheidungsmerkmal, um die unterschiedlichen Verfahren zu bewerten, ist die Euklidische Distanz

$$\delta = \sqrt{\left(\Re\{x_i(t)\} - \Re\{x_j(t)\}\right)^2 + \left(\Im\{x_i(t)\} - \Im\{x_j(t)\}\right)^2}$$

Mehrwertig bedeutet, dass ein bestimmtes Signal $s_i(t)$ aus M Signalen ausgewählt werden kann was $\log_2(M)$ Bits entspricht. Deshalb wird in der Regel $M = 2^m$ gewählt. Damit entspricht ein Signal m-bit. Die Zuordnung der Bits zu den Signalen wird **Labeling** genannt.

Bei der Verwendung von 2^m Signalen erhöht sich die binäre Datenübertragungsrate um den Faktor m gegenüber einer binären Datenübertragungsrate.

Eine größere Euklidische Distanz führt zu einer geringeren Symbolfehlerwahrscheinlichkeiten. Deshalb hat das Binary Phase Shift Keying (BPSK) Modulationsverfahren die größte Euklidische Distanz.



Sollen 2 Bits bei der Übertragung mit einem Signal übertragen werden, sind 2^2 = 4 unterschiedliche Signale erforderlich.

Das Labeling wurde so gewählt, dass sich benachbarte Signale nur in einem Bit unterscheiden.

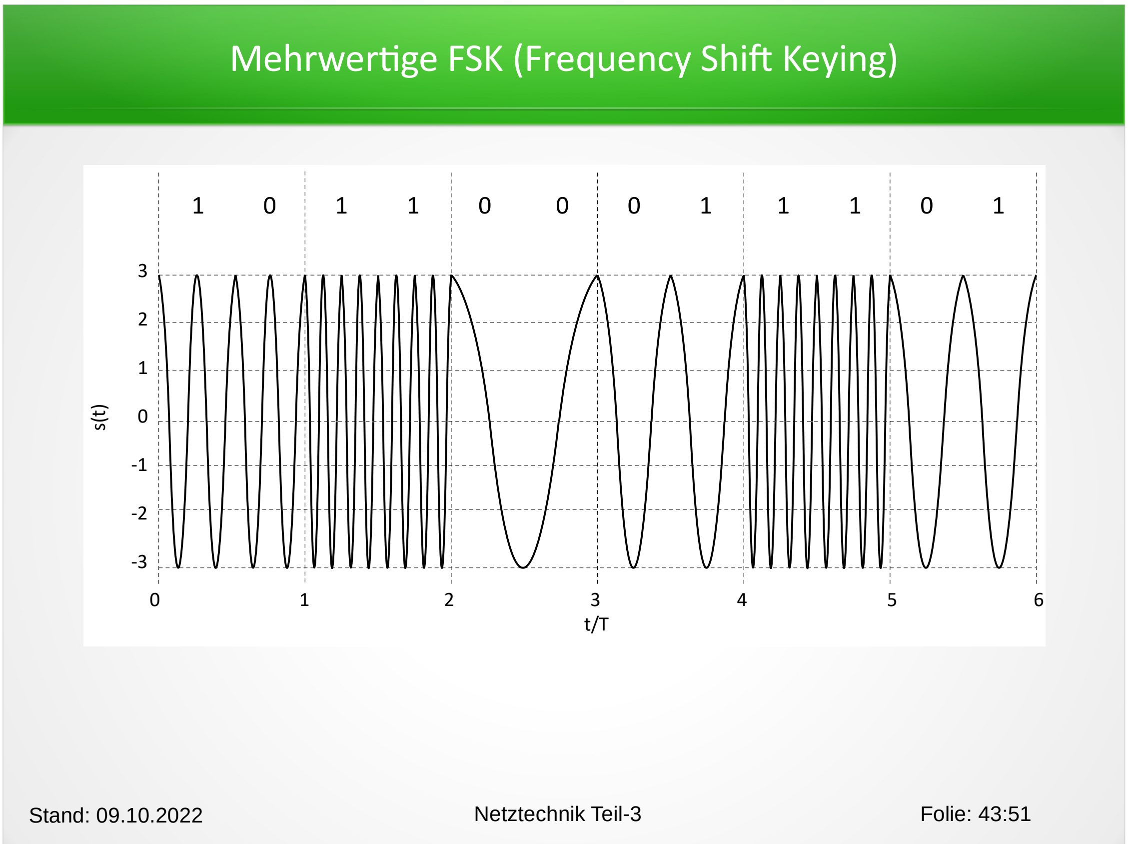
Es ist auch zu erkennen, dass mit jedem Signal zwei Bits übertragen werden. Dies führt zu einer Erhöhung der Datenübertragungsrate.

In der komplexen Ebene können die beiden Symbole 00 und 01 auch als um 180 Grad phasenverschobene Versionen von 10 und 11 verstanden werden (was einer Kombination von Amplituden und Phasenmodulation entspricht) . Bei der 4-ASK ist die Euklidische Distanz δ = 2 zwischen zwei benachbarten Signalen. Z. B.

$$\delta = \sqrt{(\Re\{3\} - \Re\{1\})^2 + (\Im\{0\} - \Im\{0\})^2} = 2$$

In der Folie ist zu erkennen, dass die Signale eine unterschiedliche Amplitude und damit auch unterschiedliche Energie haben, was eigentlich zu vermeiden ist, da manche Anwendungen damit Probleme haben. Unter der Voraussetzung, dass alle Signale gleichwahrscheinlich auftreten, kann die mittlere Energie folgendermaßen berechnet werden:

$$E[x_i^2] = \frac{1}{4}((-3)^2 + (-1)^2 + 1^2 + 3^2) = \frac{20}{4} = 5$$



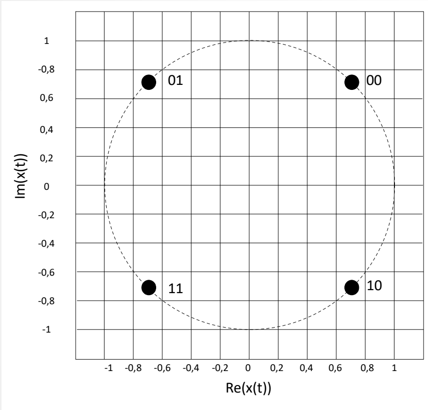
Die zu übertragende Information wird in unterschiedlichen Frequenzen codiert.

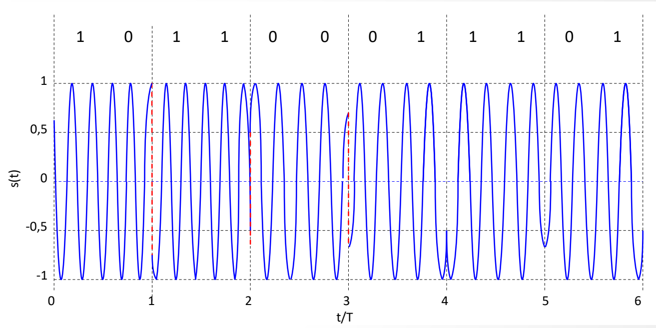
Dafür werden 2^m unterschiedliche Frequenzen benötigt um die Signale zu übertragen.

Im Beispiel werden $2^2 = 4$ Signale für die Übertragung von jeweils 2 Bit Übertragen.

Eine Darstellung in der komplexen Ebene macht hier keinen Sinn, da nur ein Punkt dargestellt werden würde. Es gibt keine Änderung in der Amplitude und der Phase.

Mehrwertige PSK (Phase Shift Keying)

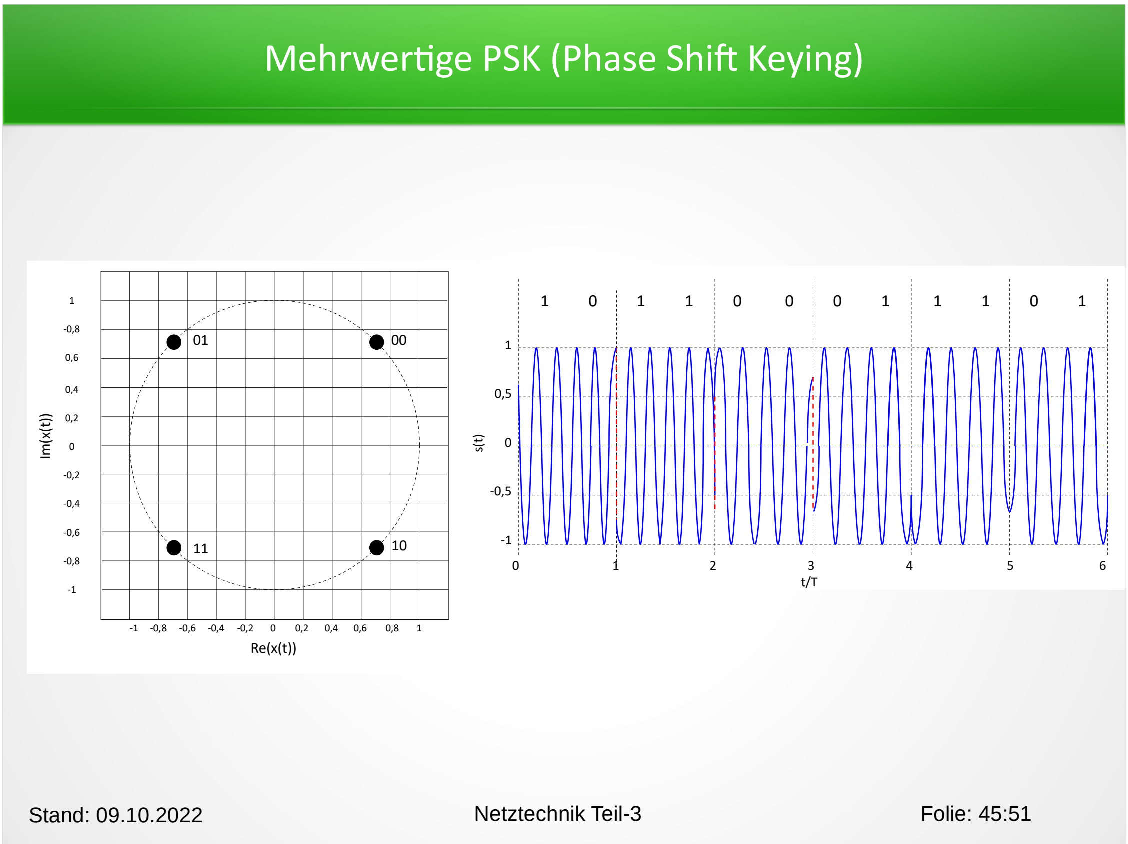




Die zu übertragende Information wird in unterschiedlichen Phasen codiert. Dies hat den Vorteil, dass alle Signale die gleiche Energie haben. Damit ändert sich am Sender die abgestrahlte Energie für unterschiedliche Symbole nicht.

Die minimale Euklidische Distanz ist $\delta = \sqrt{2}$. Dreht man die Konstellation wie in Abbildung 18 um π/4 nach rechts, dann liegt Das Signal 00 auf den Koordinaten (1,0) und das Signal 01 auf den Koordinaten (0,1). Damit ergibt sich die Distanz zu.

$$\delta = \sqrt{1+1}$$



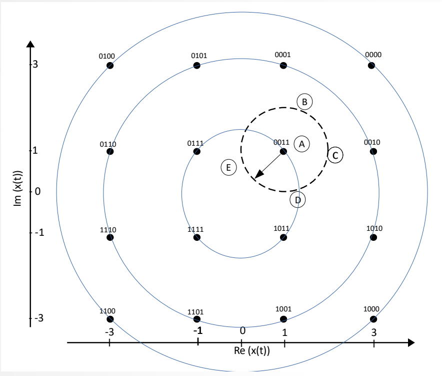
Die zu übertragende Information wird in unterschiedlichen Phasen codiert. Dies hat den Vorteil, dass alle Signale die gleiche Energie haben. Damit ändert sich am Sender die abgestrahlte Energie für unterschiedliche Symbole nicht.

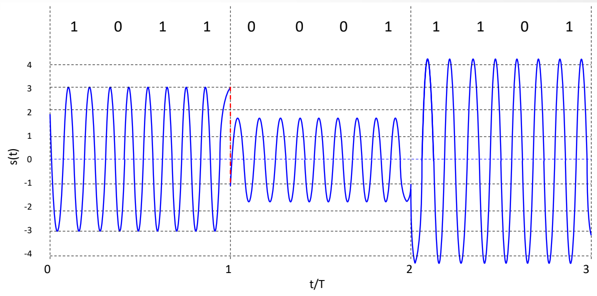
Die minimale Euklidische Distanz ist $\delta = \sqrt{2}$. Dreht man die Konstellation wie in Abbildung 18 um $\pi/4$ nach rechts, dann liegt Das Signal 00 auf den Koordinaten (1,0) und das Signal 01 auf den Koordinaten (0,1). Damit ergibt sich die Distanz zu.

$$\delta = \sqrt{1+1}$$

Man kann diesen Faden jetzt weiter spinnen und die Abstände immer feiner granulieren. Bei jeder Halbierung kann ein Bit mehr pro Symbol übertragen werden.

Modulation:
Quadrature Amplitude Modulation (QAM)



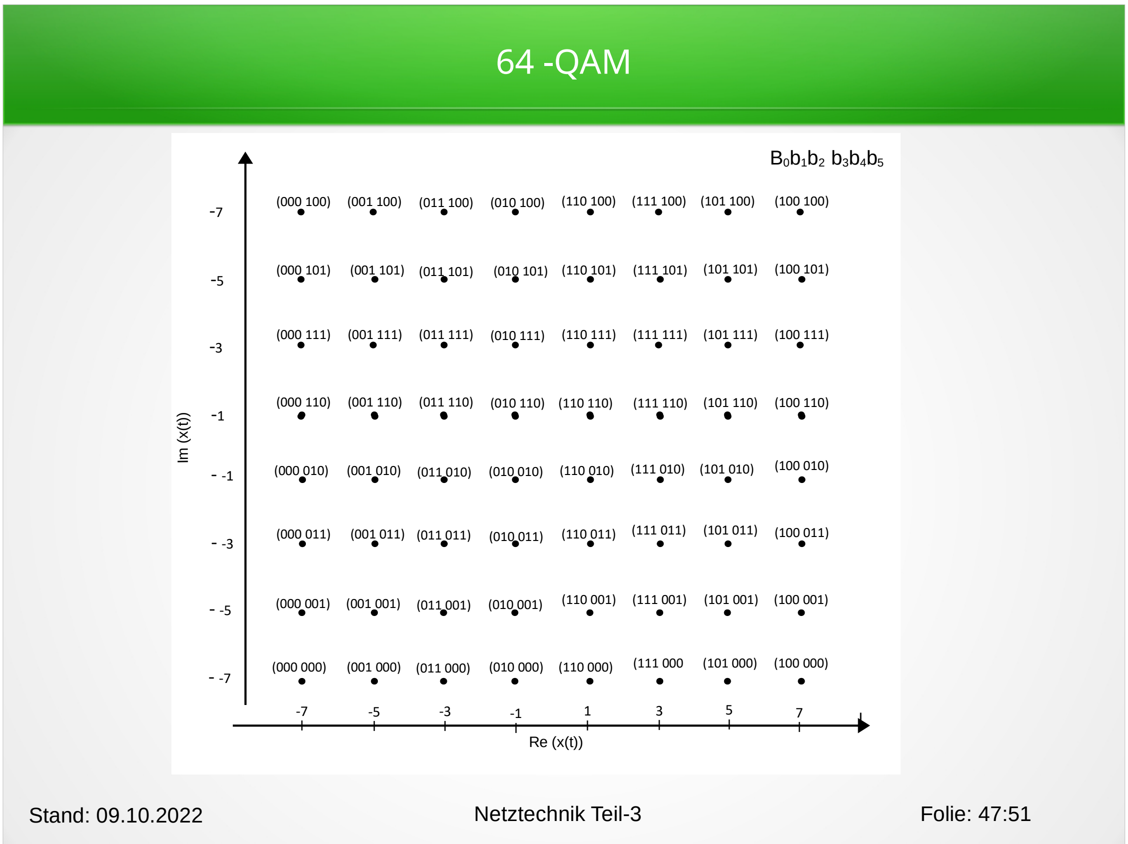


Kombiniert man die PSK noch mit unterschiedlichen Amplituden kommt man zur Quadrature Amplitude Modulation (QAM).
Hier werden 3 unterschiedliche Radien (Amplituden) und damit unterschiedliche Energien, sowie 12 Phasenwinkel verwendet.

Im Signalraum kann man die QAM als zweidimensionale ASK auffassen. In der Folie ist zu sehen, dass eine vervierfachte 4-ASK 16 Signalpunkte hat was zur Bezeichnung 16-QAM führt.

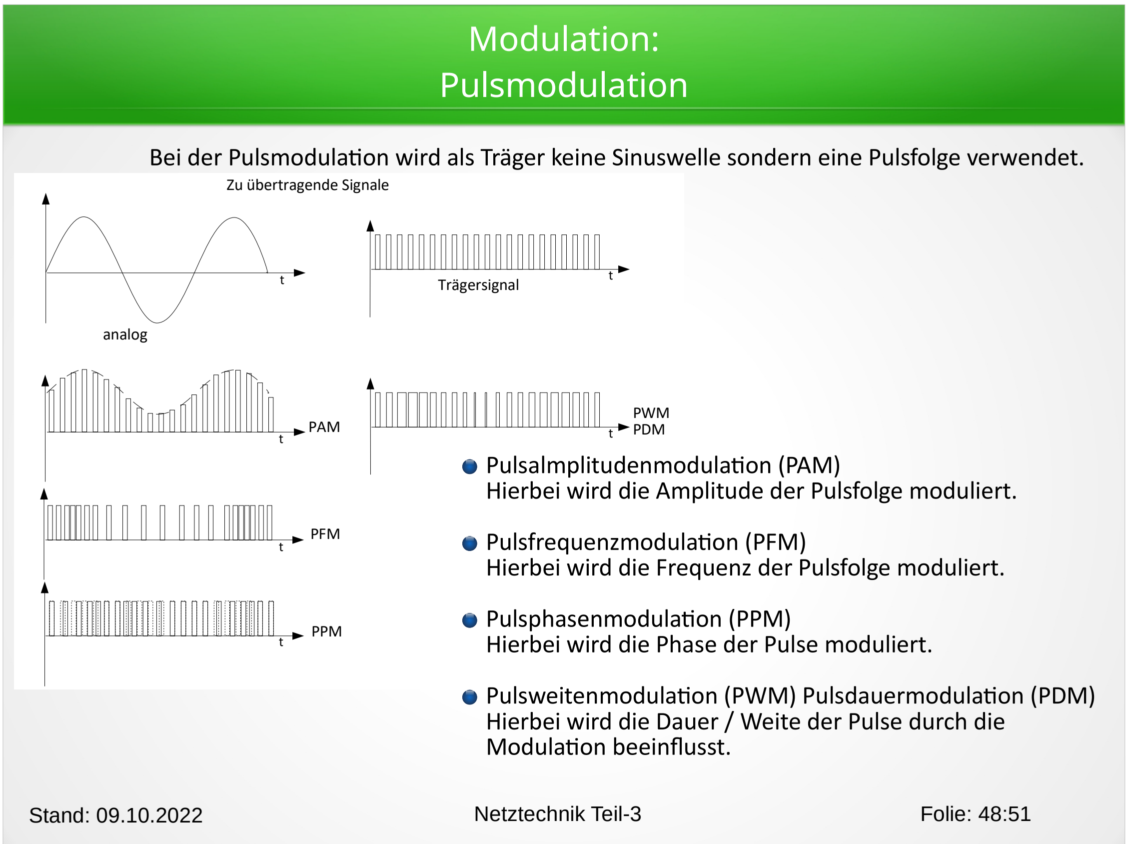
Wie bei ASK ist die minimale Euklidische Distanz δ = 2. Bei gleich wahrscheinlichen Signale ist die mittlere Energie 6.

Jeder Signalpunkt ist mit 4 Bit gelabelt, weshalb für die Übertragung der verwendeten Bitfolge nur drei Symbole erforderlich sind.



Das Spiel lässt sich immer weiter treiben. In der Folie ist eine 64-QAM dargestellt. Damit lassen sich 6 Bit übertragen. Damit sind nur zwei Symbole zur Übertragung der Bitfolge erforderlich. Welche?
Oberste Zeile das zweite Symbol von rechts und in der zweiten Zeile das dritte Symbol von links.

Mittlerweile gibt es das 4096-QAM-Modulationsverfahren. Damit lassen sich mit einem Symbol alle 12 Bits der verwendeten Bitfolge übertragen..

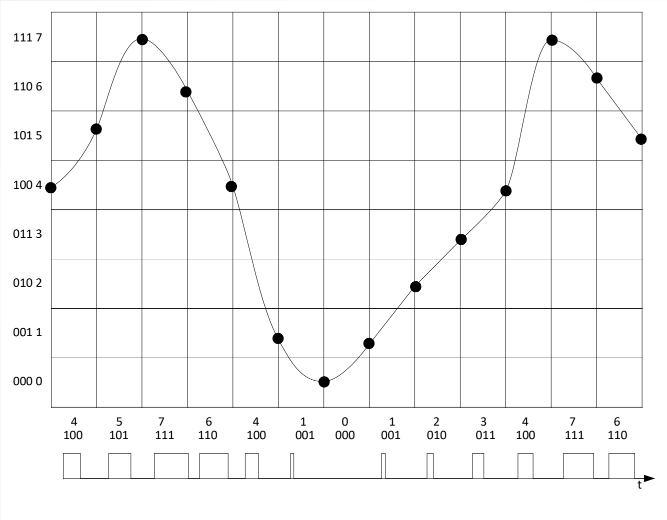


Anstelle eines Sinusträgers kann auch ein Puls verwendet werden.

Ein Puls kann in den folgenden Parametern beeinflusst (moduliert) werden:
- Amplitude
- Frequenz
- Phase
- Weite / Dauer

Modulation:
Puls-Code-Modulation

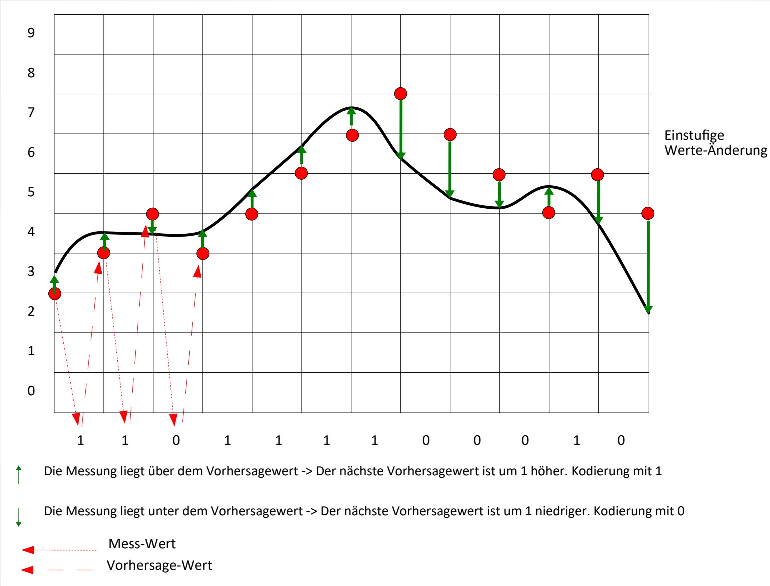
Hierbei wird das Ergebnis der Quantisierung bei der Abtastung für die Modulation verwendet.
Das Verfahren wird PCM (Pulse Code Modulation) genannt.



In der Abbildung ist der Kurvenverlauf in der PCM moduliert.

Der resultierende Wire-Code ist ein NRZ-Code.

Modulation:
Deltamodulation (DM) / Differenz Puls Code Modulation (DPCM)



Bei der Deltamodulation wird so moduliert, dass immer nur der Wert 0 oder 1 zu übertragen ist. Dabei wird ausgehend von der vorangegangenen Abtastung ein Vorhersagewert ermittelt. Der Vorhersagewert ist 1 wenn der tatsächliche Abtastwert über dem vorhergesagten Wert liegt. Der Vorhersagewert ist 0, wenn der tatsächliche Abtastwert unter dem vorhergesagten Wert liegt.

Die Differenz zwischen Signal und Vorhersagewert wird binär codiert und übertragen.

Bei großen Signalsprüngen ist dieses Verfahren zu ungenau! Abhilfe bringt hier die „Adaptive Deltamodulation ADM“. Dabei wird jeder n-te gleich lautende Wert nicht mit 1 sondern um den doppelten Wert geändert.

Ein Nachteil dieses Verfahrens ist, dass es schlecht einer Gleichspannung folgen kann. Ein Kompromiss ist hier die Differenz-Pulscode-Modulation. Hierbei wird die einstufige Werte-Ermittlung durch eine mehrstufige Quantisierung ersetzt. Eingesetzt wird dieses Verfahren bei der Übertragung von Fernsehbildern.

Zusammenfassung

Codierung (Source Coding / Channel Coding / Wire-Coding)

Source-Coding (Huffman-Codierung (tabellarisch / graphisch))

Channel-Coding (Paritätsbits / Hamming-Code / CRC)

Wire-Coding (Gleichstromanteil / RZ-Codes / NRZ-Codes / Biphase-Codes / Tenary-Codes / ..)

Bitfehlerrate / Augenmuster

Modulation (AM / FM /PM /ASK / FSK / PSK / BPSK / QAM / xQAM /…
… / Puls- / Delta- / Adaptive DM / Differenz Puls Code Modulation (DPCM))