

# Zusammenfassung Information Theory and Coding

Markus Velm

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Informationstheorie . . . . .	1
1.2. Quellcodierung . . . . .	1
1.2.1. Huffman-Code . . . . .	1
1.2.2. Arithmetische Codierung . . . . .	1
1.3. Kanalmodell . . . . .	1
1.3.1. Binärkanal . . . . .	1
1.3.2. Bedingte Entropie . . . . .	1
1.3.3. Entropiemodell . . . . .	2
1.4. Kanalkapazität . . . . .	2
1.5. Shannon-Theoreme . . . . .	2
1.5.1. Erster Satz von Shannon . . . . .	2
1.5.2. Zweiter Satz von Shannon . . . . .	2
1.5.3. Shannon-Hartley . . . . .	2
<b>2. Blockcodes</b>	<b>2</b>
2.1. Generelles . . . . .	3
2.2. Fehlerwahrscheinlichkeit . . . . .	3
2.3. Hamming-Codes . . . . .	3
<b>3. Galois-Felder</b>	<b>4</b>
3.1. Algebraische Strukturen . . . . .	4
3.2. Eigenschaften Galois-Felder . . . . .	4
<b>4. Reed-Solomon-Code</b>	<b>5</b>
4.1. Wunsch und Idee . . . . .	5
4.2. Codierung . . . . .	5
4.2.1. Generatorpolynom . . . . .	5
4.2.2. IDFT (nicht systematisch) . . . . .	5
4.2.3. Polynommultiplikation (nicht systematisch) . . . . .	5
4.2.4. Polynomdivision (systematisch) . . . . .	5
4.2.5. Über Prüfpolynom (systematisch) . . . . .	5
4.2.6. Zyklischer Code . . . . .	6
4.3. Decodierung . . . . .	6
4.3.1. Vorgehen . . . . .	6
4.3.2. Schlüsselgleichungen . . . . .	6
4.3.3. Euklidischer Algorithmus . . . . .	6
4.3.4. Forney-Algorithmus . . . . .	7
4.4. Kürzere Codes . . . . .	7
<b>5. Erweiterungskörper</b>	<b>7</b>
5.1. Idee . . . . .	7
5.2. Eigenschaften von Erweiterungskörpern . . . . .	7
<b>6. BCH-Codes</b>	<b>7</b>
6.1. Idee . . . . .	7
6.2. Kreisteilungsklassen . . . . .	7
<b>7. Faltungscodes</b>	<b>7</b>
7.1. Ein Ausgang . . . . .	7
7.2. Mehrere Ausgänge . . . . .	8
7.3. Mehrere Ausgänge . . . . .	8
<b>A. Hilfreiches</b>	<b>9</b>
A.1. Inverse in Galois-Feldern . . . . .	9

A.2. Rechnen im Erweiterungskörper . . . . .	9
A.3. Syndromstellen aus Generatorpolynom . . . . .	9
A.4. ABC/PQ-Formel . . . . .	9
<b>B. Polynome</b>	<b>9</b>
B.1. Polynommultiplikation . . . . .	9
B.2. Polynomdivision . . . . .	9
<b>C. Lineare Algebra</b>	<b>9</b>
<b>D. Digitale Signalverarbeitung</b>	<b>10</b>
<b>E. Wahrscheinlichkeitstheorie</b>	<b>11</b>
E.1. Satz von Bayes . . . . .	11
E.2. Kombinatorik . . . . .	11
E.2.1. Permutation . . . . .	11
E.2.2. Variation . . . . .	11
E.2.3. Kombination . . . . .	12

# 1. Einleitung

## 1.1. Informationstheorie

Bit: binary unit → Einheit für Information

bit: binary digit → bit als binäres Symbol

### Informationsgehalt

je unwahrscheinlicher ein Symbol  $x$  auftritt, desto mehr Information enthält es:

$$I(x) = \lg\left(\frac{1}{P(x)}\right) = -\lg(P(x))$$

$P$ : Wahrscheinlichkeit eines Symbols

$I$ : Informationsgehalt  $[I] = \text{Bit}$

### Entropie

gemittelter Informationsgehalt einer Quelle  $X$ :

$$H(X) = \sum_i P(x_i) \cdot I(x_i) = - \sum_i P(x_i) \cdot \lg(P(x_i))$$

$H$ : Entropie  $[H] = \text{Bit/Symbol}$

### Entscheidungsgehalt

Entropie wird maximal, wenn alle Symbole gleichwahrscheinlich sind → Entscheidungsgehalt

$$H_0 = \lg(N)$$

$H_0$ : Entscheidungsgehalt  $[H_0] = \text{Bit/Symbol}$

$N$ : Anzahl der Symbole eines Alphabets

### Redundanz

$$R = H_0 - H$$

$$r = \frac{R}{H_0}$$

$R$ : Redundanz  $[R] = \text{Bit/Symbol}$

$r$ : relative Redundanz

## 1.2. Quellcodierung

### 1.2.1. Huffman-Code

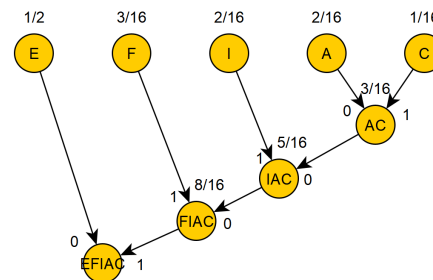
ist **Prefixcode**: ein Codewort ist niemals Anfang eines anderen Codewortes

Codebaum aufbauen:

1. Ordne die Symbole nach Auftretswahrscheinlichkeit
2. Fasse Symbole mit niedrigster Wahrscheinlichkeit zu einem Symbol zusammen und addiere die Wahrscheinlichkeiten
3. Wiederhole bis nur ein Symbol übrig bleibt

Beschrifte die Pfade mit 1 und 0

→ Codewort ergibt sich, indem man von Wurzel bis zum Blatt geht



Hinweis: Beschriftung der 0; 1 theoretisch egal aber für Überprüfung mit Onlinerechnern sollte konsistent der Pfad mit der geringeren und höheren Wahrscheinlichkeit gleich beschriftet werden

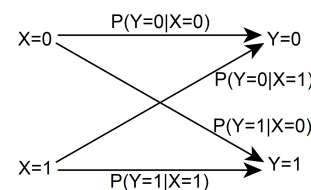
### 1.2.2. Arithmetische Codierung

Codierung eines Wortes (oder Textes) durch Zahl

Endezeichen notwendig, da keine natürliche Terminierung des Codes

## 1.3. Kanalmodell

### 1.3.1. Binärkanal



$P(Y|X)$ : »Wahrscheinlichkeit für  $Y$ , wenn  $X$  gesendet wurde« (a-priori-Wahrscheinlichkeit)

$P(X|Y)$ : »Wahrscheinlichkeit für  $X$ , wenn  $Y$  empfangen wurde« (a-posteriori-Wahrscheinlichkeit)

**Unsymmetrisch**: Fehlerwahrscheinlichkeit für »0« und »1« unterschiedlich

$$(P(Y|X)) = \begin{pmatrix} P(0|0) & P(0|1) \\ P(1|0) & P(1|1) \end{pmatrix}$$

**Symmetrisch**: Bitfehlerwahrscheinlichkeit  $P_e$  für »0« und »1« gleich

$$(P(Y|X)) = \begin{pmatrix} P(0|0) & P(0|1) \\ P(1|0) & P(1|1) \end{pmatrix} = \begin{pmatrix} 1 - P_e & P_e \\ P_e & 1 - P_e \end{pmatrix}$$

### 1.3.2. Bedingte Entropie

$$H(Y|X) = - \sum_j \sum_i P(x_i, y_i) \lg(P(y_i|x_i)) \quad \text{Irrelevanz}$$

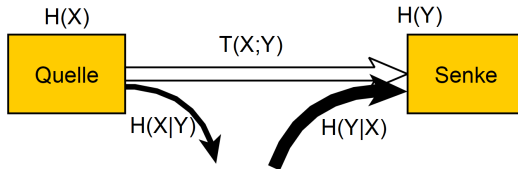
$$H(X|Y) = - \sum_j \sum_i P(x_i, y_i) \lg(P(x_i|y_i)) \quad \text{Äquivokation}$$

$H(Y|X)$ : **Irrelevanz (Fehlinformation)**, mittlere Unsicherheit über empfangene Symbole, wenn Sendesymbole bekannt

$H(X|Y)$ : **Äquivokation (Informationsverlust)**, mittlere

Unsicherheit über gesendete Symbole, wenn Empfangssymbole bekannt

### 1.3.3. Entropiemodell



für Empfänger ist  $H(Y)$  (also der Ausgang des Kanals) eine neue Quelle, allerdings mit zusätzlicher Fehlinformation

Die Transinformation ist dann die Entropie des Kanals minus der Fehlinformation

$$T(X;Y) = H(Y) - \underbrace{H(Y|X)}_{\text{Fehlinformation}}$$

oder die Entropie der Quelle minus des Informationsverlusts

$$T(X;Y) = H(X) - \underbrace{H(X|Y)}_{\text{Informationsverlust}}$$

Fehlinformation größer als Informationsverlust (außer es passieren keine Fehler, oder die Entropie der Quelle ist maximal, dann  $H(X|Y) = H(Y|X)$ )

$\hookrightarrow$  Entropie am Ausgang des Kanals größer als die der Quelle (außer es passieren keine Fehler, oder die Entropie der Quelle ist maximal, dann  $H(X) = H(Y) = 1$ )

### 1.4. Kanalkapazität

Die Kanalkapazität ist das Maximum an Information, welche über den Kanal geht

$$\begin{aligned} C &= \max(T(X;Y)) = \max(H(X) - H(X|Y)) \\ &= \max(H(Y) - H(Y|X)) \\ &= 1 - H(X|Y) = 1 - H(Y|X) \end{aligned}$$

für binären, symmetrischen Kanal:

$$H(Y|X) = -p \cdot \log(p) - (1-p) \cdot \log(1-p)$$

$$C = 1 + p \cdot \log(p) + (1-p) \cdot \log(1-p)$$

$$C_S = C \cdot R_S$$

$p$ : Bitfehlerwahrscheinlichkeit

$C$ : Kanalkapazität in Information/Symbol  $[C] = \text{Bit}/\text{Symbol}$

$C_S$ : Kanalkapazität in Information/Zeit  $[C_S] = \text{Bit}/s$

$R_S$ : Symbolrate  $[R_S] = \text{Symbol}/s$

## 1.5. Shannon-Theoreme

### 1.5.1. Erster Satz von Shannon

Quellcodierung

Ist ein Informationsfluss von  $H_S$  gewünscht, welcher kleiner als die Kanalkapazität  $C_S$  ist, ist es möglich einen Code zu finden, sodass der Informationsfluss übertragen werden kann

### 1.5.2. Zweiter Satz von Shannon

Kanalcodierung

Ist der gewünschte Datenfluss  $R_S$  ( $[R_S] = \text{Symbol}/s = \text{bit}/s$ ) kleiner als die Kanalkapazität  $C_S$ , dann gibt es einen Code, sodass die Fehlerwahrscheinlichkeit beliebig klein wird

### 1.5.3. Shannon-Hartley

Kanalkapazität abhängig vom SNR

$$C_S = W \cdot \log\left(1 + \frac{S}{N}\right)$$

$C_S$ : Kanalkapazität in Information/Zeit  $[C_S] = \text{Bit}/s$

$W$ : Bandbreite  $[W] = \text{Hz}$

$S$ : Signalleistung

$N$ : Rauschleistung

### Bandbreiteneffizienz

Wird mit  $R_S = C_S$  übertragen, kann die Bandbreiteneffizienz bestimmt werden

$$\frac{R_S}{W} = \log\left(1 + \frac{S}{N}\right)$$

$$S = \frac{E_b}{T_b} = E_b \cdot R_S \quad N = N_0 \cdot W$$

$$\frac{S}{N} = \frac{E_b \cdot R_S}{N_0 \cdot W}$$

$\hookrightarrow$  damit S/N und Bandbreiteneffizienz abhängig von  $W$

Ist eine Bandbreiteneffizienz gewünscht, kann das erforderliche S/N ausgerechnet werden

$$\frac{E_b}{N_0} = \frac{W}{R_S} \log\left(2^{\frac{R_S}{W}} - 1\right)$$

$$\frac{E_b}{N_0} = -1,6 \text{ dB}$$

$\frac{R_S}{W}$ : Bandbreiteneffizienz  $[\frac{R_S}{W}] = \left(\frac{\text{bit}/s}{\text{Hz}}\right)$

$E_b$ : Energie die für die Übertragung von 1 bit aufgewendet wird

$[E_b] = J = Ws = W/\text{Hz}$

$T_b$ : Zeitdauer eines bits  $[T_b] = s = 1/\text{Hz}$

## 2. Blockcodes

Code beschrieben durch  $C(n, k, d)$

$n$ : Länge Codewort

$k$ : Länge Informationswort

$d$ : Mindestabstand

$$\text{Coderate: } CR = \frac{k}{n}$$

## 2.1. Generelles

### Generatormatrix

Erzeugung eines Codewortes über Multiplikation eines Informationsvektors

$$\vec{c} = i \cdot G$$

$c$ : Codewort

$i$ : Informationswort

$G$ : Generatormatrix

### Prüfmatrix

$$H\vec{c}^T = 0$$

$$H \cdot G^T$$

$H$ : Prüfmatrix

### Syndrom

wenn Empfangswort  $r$  fehlerhaft ist (und nicht zum Coderaum gehört) dann ist das Produkt aus Prüfmatrix und Empfangswort das *Syndrom* und nicht mehr 0

$$H\vec{r}^T = H(\vec{c} + \vec{f})^T = H\vec{f}^T = \vec{s} \neq 0$$

### Systematische Codes

Systematischer Code: Einheitsmatrix  $I$  ist Teil der Generatormatrix

$$G = [I|G']$$

$$\text{Bsp.: } C(7, 4, 3): G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

### Lineare Codes

Eigenschaften:

- Mindestgewicht = Mindestabstand
- Codewort + anderes Codewort = wieder Codewort
- Nullwort ist teil des Codes

### Gewicht

Gewicht eines Codewortes: Anzahl der von 0 verschiedenen Stellen

Mindestgewicht: Minimale Anzahl an Stellen, die von 0 verschieden sind

### Distanz/Abstand

Hamming-Distanz: Anzahl verschiedener Stellen zweier Codewörter

Mindestdistanz: Mindestanzahl verschiedener Stellen zweier beliebiger Codewörter eines Codes

### Fehlerkorrektur/Fehlererkennung

Anzahl der korrigierbaren Fehler  $e$  bei gegebenem Abstand  $d$

$$e = \left\lfloor \frac{d-1}{2} \right\rfloor$$

bei ungeradem  $d$ :

es werden  $d - 1$  Fehler erkannt

bei geradem  $d$ :

es werden alle ungeraden Anzahlen an Fehlern erkannt

## 2.2. Fehlerwahrscheinlichkeit

### Restblockfehlerwahrscheinlichkeit

Wahrscheinlichkeit, dass von  $n$  Symbolen beliebige  $m$  falsch und die übrigen Symbole  $n - m$  richtig sind:

$$\binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

für ungerades  $d$ :

alle Fehler werden entweder korrigiert, oder nicht erkannt

Kann ein Code  $e$  Fehler korrigieren, dann verbleibt eine Restblockfehlerwahrscheinlichkeit von:

$$P_{Block} = \sum_{m=e+1}^n \binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

$$= 1 - \sum_{m=0}^e \binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

$p$ : Fehlerwahrscheinlichkeit vor der Decodierung

$P_{Block}$ : Wahrscheinlichkeit, für Symbolfehler im decodierten Block

für gerades  $d$ :

alle ungeraden Fehler werden erkannt, da sie nicht in Korrekturkugeln liegen; können damit aber auch nicht korrigiert werden

→ für die Wahrscheinlichkeit, dass Fehler nicht erkannt werden, tragen also nur die geraden Fehleranzahlen, welche größer als  $e$  sind, bei

$$P_{Block,erkennen} = \sum_{m=e+1}^n \binom{n}{2m} \cdot p^{2m} \cdot (1-p)^{n-2m}$$

$P_{Block}$  gleich wie bei gerader Anzahl

### Symbolfehlerwahrscheinlichkeit

Block besteht aus  $k$  Symbolen

Bei gegebener Restblockfehlerwahrscheinlichkeit, gibt es eine Symbolfehlerwahrscheinlichkeit von

$$P_{Symb} = P_{S|Block} \cdot P_{Block} = \frac{2k-1}{2^k-1} P_{Block}$$

## 2.3. Hamming-Codes

immer  $d = 3$ , damit immer  $e = 1$

$$n = 2^h - 1$$

$$k = n - h$$

h	n	k	d
2	3	1	3
3	7	4	3
4	15	11	3
5	31	26	3
⋮	⋮	⋮	⋮

### Konstruktion

#### 1. Erstellung Prüfmatrix

Spalten sind Dualdarstellung der Spaltennummer

$$H_{n-k \times n}$$

Beispiel:  $C(7, 4, 3)$

$$H_{3 \times 7} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

2. Spalten tauschen, sodass hinten Einheitsmatrix

$$H = [A_{n-k \times k} | I_{n-k \times n-k}]$$

Spalte 1 mit 7

2 mit 6

4 mit 5

$$\rightarrow H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} = [A_{3 \times 4} | I_{3 \times 3}]$$

3. Generatormatrix aufstellen

$$G_{k \times n} = [I_{k \times k} | A_{k \times n-k}^T]$$

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

systematisch, da Informationswort zusammenhängend im Codewort steht

4. Rücktauschen der Spalten

Spalte 1 mit 7

2 mit 6

4 mit 5

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

quasi-systematisch, da Informationswort zwar im Codewort, aber nicht zusammenhängend

**Fehler und Syndrom**

Fehler gibt an, an welcher Stelle des Empfangsworts ein Fehler aufgetreten ist

Bsp.:

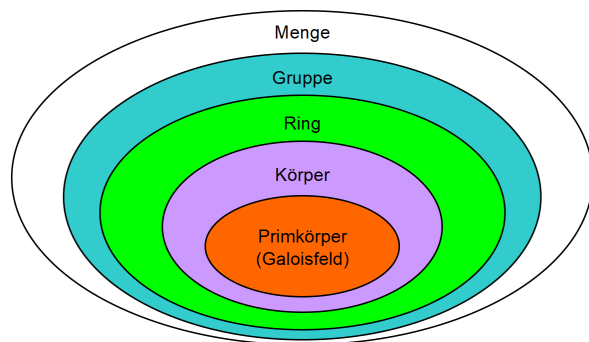
$$\vec{s} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

→ Fehler an Stelle  $(011)_b = 3$  im Empfangswort

→ da binärer Code, muss für Fehlerkorrektur das dritte bit nur invertiert werden

### 3. Galois-Felder

#### 3.1. Algebraische Strukturen



#### Menge

Verbund von Elementen, welche keine Operationen beinhalten (Möbel können eine Menge sein, es kann aber nicht Tisch + Stuhl gerechnet werden)

#### Halbgruppe

Menge  $A$  mit Verknüpfung  $\gg + \ll$  ist eine Halbgruppe, wenn

- Abgeschlossenheit (+ zweier Elemente von  $A$  ergibt wieder ein Element von  $A$ )
- Assoziativität (Reihenfolge der Operation mit  $+$  spielt keine Rolle,  $a + (b + c) = (a + b) + c$ )
- Existenz eines neutralen Elements (Element  $a$  + neutrales Element  $n$  ergibt wieder Element  $a$ )

#### Gruppe

Halbgruppe plus

- Existenz eines additiven inversen Elements ( $a + b = n$ )

#### Abelsche oder kommutative Gruppe

Gruppe plus

- Kommutativität (Reihenfolge der Operanden spielt keine Rolle,  $a + b = b + a$ )

#### Ring

abelsche Gruppe plus

- Abgeschlossenheit bezüglich  $\gg \cdot \ll$
- Assoziativität bezüglich  $\gg \cdot \ll$
- Distributivität ( $a \cdot (b + c) = a \cdot b + a \cdot c$ )

#### Körper

Ring plus

- Kommutativität bezüglich  $\gg \cdot \ll$  ( $a \cdot b = b \cdot a$ )
- Neutrales Element bezüglich  $\gg \cdot \ll$
- Inverses Element bezüglich  $\gg \cdot \ll$  für jedes Element

#### Primkörper/Galois-Feld

Körper, indem Addition und Multiplikation  $\mod p$  gerechnet wird ( $p$  muss dabei eine Primzahl sein)  
 $\hookrightarrow GF(p)$

#### 3.2. Eigenschaften Galois-Felder

##### Primitives Element

Element  $\alpha$ , welches durch ihre  $p - 1$  Potenzen alle Elemente (außer 0) des  $GF(p)$  erzeugt

Bsp.  $GF(5), \alpha = 2$ :

$$\begin{aligned} 2^0 &= 1 \mod 5 = 1 \\ 2^1 &= 2 \mod 5 = 2 \\ 2^2 &= 4 \mod 5 = 4 \\ 2^3 &= 8 \mod 5 = 3 \end{aligned}$$

ab hier zyklische Wiederholung:

$$2^4 = 16 \mod 5 = 1$$

##### Polynome

Folge an  $n$  Zahlen im Galois-Feld wird als Polynom vom Grad  $n - 1$  geschrieben

$$\hookrightarrow \{1; 4; 3; 1\} \rightarrow A(x) = 1x^3 + 4x^2 + 3x + 1$$

Auswertung des Polynoms an verschiedenen Stellen von  $\alpha^i$  ergibt ihre Fouriertransformierte  $a(x)$

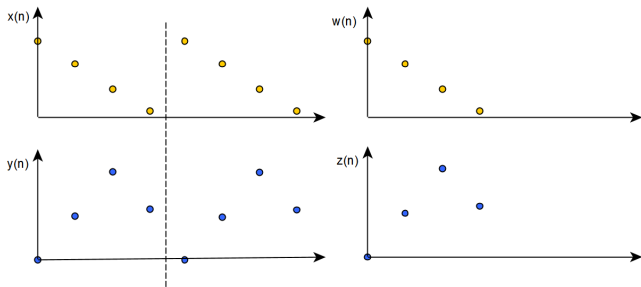
$$a_i = A(\alpha^i)$$

### Zyklische Faltung

Polynommultiplikation im Galois-Feld  $\rightarrow$  zyklische Faltung

Normale Faltung mit endlichen Signalen  $\rightarrow$  endliches Faltungsergebnis

Zyklische Faltung: Signale sind periodisch, damit Faltungsergebnis ebenfalls periodisch (und damit unendlich lang)



links: zyklische Faltung

rechts: normale Faltung

## 4. Reed-Solomon-Code

### 4.1. Wunsch und Idee

#### Wunsch

Konstruktion eines Codes mit vorgegebener Korrekturfähigkeit  $\rightarrow$  Vorgabe des Mindestabstandes  $d$

$$e = \left\lfloor \frac{d-1}{2} \right\rfloor$$

$$d = 2e + 1$$

bei linearem Code ist Mindestabstand = Mindestgewicht

$\rightarrow$  Codeworte haben mind.  $d$  von 0 verschiedene Koeffizienten

d'Alembert: Polynom vom Grad  $n$  hat  $n$  komplexe (oder höchstens  $n$  reelle) Nullstellen; auch im Galois-Feld

#### Idee

Konstruktion des Informationswortes als Polynom  $A(x)$  mit Grad  $k-1$  (damit höchstens  $k-1$  Nullstellen)

Im  $GF(p)$  mit Ordnung  $n = p-1$  kann man  $A(x)$  an  $n$  Stellen auswerten, danach wiederholen sich die Werte

$\rightarrow$  Auswertung des Polynoms für verschiedene  $x$  (bzw.  $\alpha^i$ ) ergeben die Koeffizienten  $a_i$  des Polynoms  $a(x)$

$$a_i = A(\alpha^i) \quad \text{IDFT}$$

von diesen sind höchstens  $k-1$  Null (weil  $\text{grad}(A(x)) = k-1$ )  
von diesen sind also mind.  $n - (k-1)$  von Null verschieden  $\rightarrow$  Mindestgewicht  $d$

$$d = n - (k-1) = n - k + 1$$

### 4.2. Codierung

Verschiedene Möglichkeiten aus einem Informationswort ein Codewort zu generieren

#### 4.2.1. Generatorpolynom

$$g(x) = \prod_{i=k}^{n-1} (x - \alpha^{-i})$$

Syndromstellen beginnen hier bei  $k$ , es sind aber alle anderen Stellen möglich, solange sie zusammenhängen

$$\text{grad}(g(x)) = d - 1 = n - k = \text{Anzahl Syndromstellen}$$

$g(x)$ : Generatorpolynom  
 $i(x)$ : Informationspolynom

#### 4.2.2. IDFT (nicht systematisch)

$$a_i = A(\alpha^i)$$

$A(x)$ : Informationswort  
 $a_i$ : Koeff. des Codewortes

#### 4.2.3. Polynommultiplikation (nicht systematisch)

$$a_i = g(x) \cdot i(x)$$

#### 4.2.4. Polynomdivision (systematisch)

Informationswort ist Teil des Codewortes (an den hohen Potenzen)

$$a^*(x) = i_{k-1}x^{n-1} + i_{k-2}x^{n-2} + \dots + i_1x^{n-k+1} + i_0x^{n-k}$$

jedes Codewort muss durch Generatorpolynom teilbar sein  $\rightarrow$  ist für  $a^*(x)$  i.A. nicht der Fall

$$\frac{a^*(x)}{g(x)} = b(x) + \frac{\text{rest}(a^*(x))}{g(x)}$$

$$\rightarrow \frac{a^*(x) - \text{rest}(a^*(x))}{g(x)} = b(x)$$

$$a(x) = a^*(x) - \text{rest}(a^*(x))$$

$\text{rest}(a^*(x))$ : Divisionsrest

#### 4.2.5. Über Prüfpolynom (systematisch)

Prüfpolynom:

$$h(x) = \prod_{i=0}^{k-1} (x - \alpha^{-i})$$

Produkt aus Generator- und Prüfpolynom ist 0

$$g(x) \cdot h(x) = 0$$

und Produkt aus Codepolynom und Prüfpolynom ist 0

$$a(x) \cdot h(x) = 0$$

genau da, wo  $g(x)$  (oder  $a(x)$ ) Nullstellen hat (also  $G_i = 0$  ist) hat das Prüfpolynom  $h(x)$  keine Nullstellen (ist also  $H_i$  nicht 0) und umgekehrt

#### 4.2.6. Zyklischer Code

Multiplikation eines Polynoms mit  $x^i$  verschiebt Koeff. des Polynoms um  $i$ -Stellen

durch mod-Rechnung des Exponenten verschieben sich höhere Exponenten wieder an den Anfang des Polynoms

Bsp.:

$$x \cdot a(x) = x \cdot (2x^2 + x + 1) = 2x^3 + x^2 + x = x^2 + x + 2$$

#### 4.3. Decodierung

Idee:

Addition des Fehlerpolynoms  $f(x)$  mit  $t$  Koeffizienten (d.h.  $t$  Fehler sind auf dem Kanal aufgetreten) zum gesendeten Codewort  $a(x)$

im Zeitbereich:

$$r(x) = a(x) + f(x)$$

im Frequenzbereich:

$$R(x) = A(x) + F(x)$$

gedanklich wird ein Polynom  $c(x)$  aufgestellt, welches  $t$  Nullen an den Fehlerstellen hat

Da die Koeffizienten von  $c(x)$  die Auswertung ihrer Fouriertransformierten  $C(x)$  ist, ist der Grad von  $C(x)$   $t$

Da  $c(x)$  gerade dort 0 ist, wo  $f(x)$  ungleich 0, ist das Produkt  $f_i \cdot c_i$  immer 0 (Achtung, keine Polynommultiplikation gemeint, sondern punktweise Multiplikation)

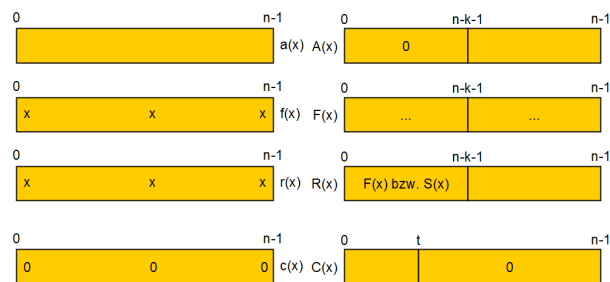
$$f_i \cdot c_i = 0$$

wenn Zeitbereich = 0  $\rightarrow$  Frequenzbereich = 0

$$F(x) \cdot C(x) = 0$$

Achtung: hier Polynommultiplikation/ Faltung/ Filterung gemeint

$\hookrightarrow$  Aufstellen der Schlüsselgleichungen



##### 4.3.1. Vorgehen

1. Fouriertransformation des empfangenen Codewortes  $r(x) \rightarrow R(x)$
2. Auslesen der Koeff. des Syndrompolynoms ( $S_0, \dots, S_n$ ) aus  $R(x)$  und Aufstellen des Syndrompolynoms
3. Berechnung des  $C(x)$  aus Schlüsselgleichungen oder euklidischem Algorithmus
4. Berechnung der Fehlerstellen durch Nullstellensuche von  $C(x)$
5. Berechnung des Fehlerwertes über Schlüsselgleichungen oder Forney-Algorithmus

##### 4.3.2. Schlüsselgleichungen

beschreiben, dass Faltung von  $C(x)$  und  $F(x)$  Null ist (Achtung: zyklische Faltung, siehe Abschnitt 3)

$F_0$  bis  $F_{n-k-1}$  (bzw.  $F_{d-2}$ ) sind bekannt, da diese direkt an den Syndromstellen von  $R(x)$  stehen

Alle  $C$ -Koeff. sind unbekannt, außer  $C_t$ , dieser wird zu 1 gesetzt

$$C_t = 1$$

da Anzahl der Fehler ( $t$ ) unbekannt ist, muss ausprobiert werden, welche minimale Anzahl an Fehlern die Schlüsselgleichungen widerspruchsfrei erfüllt

Lösen der Schlüsselgleichungen nach  $C(x)$

$\hookrightarrow$  Nullstellensuche von  $C(x)$  ergibt die Nullen des  $c(x)$

$\hookrightarrow$  wenn Grad von  $C(x)$  nicht mit Anzahl der Nullstellen übereinstimmt  $\rightarrow$  Decodierungsversagen

Lösen der Schlüsselgleichungen nach  $F(x)$

$\hookrightarrow f(x)$  aus Rücktransformation von  $F(x)$

$\hookrightarrow f(x)$  von  $r(x)$  abziehen, man erhält  $a(x)$

$$a(x) = r(x) - f(x)$$

##### 4.3.3. Euklidischer Algorithmus

Suche des ggT zweier Zahlen

Kann zur Lösung der Schlüsselgleichungen verwendet werden

Rest:

$$r_n = v_n a_n + w_n b_n$$

Rekursionsformeln für  $v_n$  und  $w_n$ :

$$v_n = v_{n-2} - q_n v_{n-1}$$

$$w_n = w_{n-2} - q_n w_{n-1}$$

$q_n$ : Quotient des vorherigen Schrittes

Initialisierung:

$$\begin{matrix} v_{-1} = 1 & v_0 = 0 \\ w_{-1} = 0 & w_0 = 1 \end{matrix}$$

Suche des  $C(x)$  und damit den Fehlerstellen

Polynomdivision von  $x^{d-1}$  und des Syndrompolynoms  $S(x)$

$$x^{d-1} : S(x)$$

Wenn Rest der Division im Grad nicht kleiner ist als die Anzahl der Fehler  $e$ , die maximal korrigiert werden können  $\rightarrow$  weiter:  $S(x) : r_1(x)$

usw.

ist Grad des Restes kleiner als  $e \rightarrow$  Berechnung des  $C(x)$  und des  $T(x)$

$$\hookrightarrow C(x) = w_n$$

$$\hookrightarrow T(x) = -r_n$$



#### 4.3.4. Forney-Algorithmus

Fehlerwertberechnung aus gegebenem  $C(x)$  und  $T(x)$

$$f_i = x^q \cdot n \cdot x^{-1} \frac{T(x)}{C'(x)} \Big|_{x=\alpha^i}$$

$q$ : Verschiebung der Syndromstellen ( $q = 5$ , wenn Syndrom an Stelle 5)

Achtung: Fehlerwert an den Stellen, an dem keine Fehler passiert sind, ist i.A. nicht 0

#### 4.4. Kürzere Codes

##### Verkürzung

Streichen von Informationswortstellen und Codewortstellen

Distanz und damit Fehlerkorrigierbarkeit bleibt gleich

Code ist nicht mehr zyklisch

Bsp.: Verkürzung eines  $C(6, 2, 5)$  um 1 auf  $C(5, 1, 5)$

##### Punktierung

### 5. Erweiterungskörper

#### 5.1. Idee

Erweitern des Grundkörpers (z.B. 2) mit Exponent (z.B. 4)  $\rightarrow GF(2^4)$

Irreduzibles Polynom ist die Primzahl des Erweiterungskörpers z.B. in  $GF(2^4)$ :

$$p(x) = x^4 + x + 1$$

Irreduzibles Polynom:  $ggT(p(x), b(x)) = 1$

größter gemeinsamer Teiler mit einem beliebigen Polynom  $b(x)$  ist 1

d.h.  $p(x)$  kann nicht in Linearfaktoren zerlegt werden

für irreduzible Polynome gilt:

- ist durch kein Polynom ohne Rest teilbar
- hat keine Nullstellen

**aber**: Nullstellen sind wichtig für Nutzung des RS-Codes, daher »Erfindung« des Elements  $\alpha$ , welches Nullstelle von  $p(x)$  ist

$$p(\alpha) = 0$$

am Beispiel:

$$p(\alpha) = \alpha^4 + \alpha + 1 = 0$$

Analogie: »Erfindung« von  $j$ , sodass gilt:

$$j^2 + 1 = 0$$

primitives Polynom: Nullstelle ( $\alpha$ ) des primitiven Polynoms erzeugt alle Elemente (außer 0) des Erweiterungskörpers

primitives Element: Nullstelle  $\alpha$  des primitiven Polynoms

#### 5.2. Eigenschaften von Erweiterungskörpern

Ordnung des primitiven Elements:  $2^m - 1$  im  $GF(2^m)$

Erzeugung der Elemente über Potenzieren des primitiven Elements  $\alpha$

zum Körper  $GF(2^m)$  gehören  $2^m$  Elemente ( $2^m - 1$  dieser wird durch Potenzieren von  $\alpha$  erzeugt)

Elemente der Erweiterungskörper sind Polynome

##### Darstellung

Erzeugung von bspw.  $\alpha^3$  in  $GF(2^4)$  mit irreduziblem Polynom  $p(x) = x^4 + x + 1$ :

$$\alpha^3 = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha^1 + 0 \cdot \alpha^0$$

dazugehörige Binärdarstellung:

1000

### 6. BCH-Codes

#### 6.1. Idee

Für Erweiterungskörper war  $\alpha$  die Nullstelle des primitiven Polynoms

ABER: d'Alembert: Polynom vom Grad  $m$  hat  $m$  Nullstellen

Wo sind die restlichen Nullstellen der primitiven Polynome höheren Grades?

$\hookrightarrow$  wenn  $\alpha$  Nullstelle von  $p(x)$  ist, dann sind auch  $\alpha^2, \alpha^{2^2}, \alpha^{2^3}, \dots, \alpha^{2^{m-1}}$  Nullstellen ( $\rightarrow$  konjugiert komplexe Nullstellen)

#### 6.2. Kreisteilungsklassen

### 7. Faltungscodes

Filterung der Eingangssequenz mit FIR-Filter

Beschreibung durch  $C(n, k, [z])$

$n$ : Anzahl Ausgänge

$k$ : Anzahl Eingänge

$z$ : Anzahl an Speicherzellen

#### 7.1. Ein Ausgang

Faltungscoder ohne Redundanz

Normaler FIR-Filter mit binären Koeffizienten, Generatorsequenz ist Impulsantwort

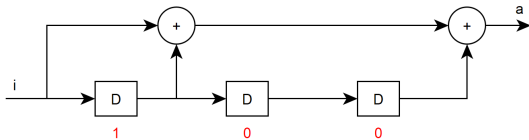
Delay:  $z^{-1} = D$

Impulsantwort:  $\vec{g}$

Inhalt der Speicherzellen:  $\vec{d}$

Ausgang:  $a = \vec{g} \cdot \begin{pmatrix} i \\ \vec{d} \end{pmatrix}$

##### Beispiel



Impulsantwort:  $g = 1 + 1D + 0D^2 + D^3 = 1 + D + D^3$

Generatorsequenz:  $\vec{g}^T = (1 \ 1 \ 0 \ 1)$

Von links nach rechts steht (100) in den Speicherzellen und es wird  $i = 1$  hineingeschrieben

$$\hookrightarrow \vec{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$a = (1 \ 1 \ 0 \ 1) \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 0$$

## 7.2. Mehrere Ausgänge

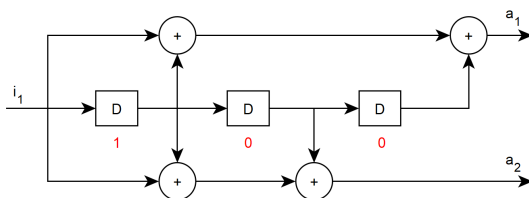
Filter mit mehreren Ausgängen

2 Impulsantworten des Filters geben 2 verschiedene Ausgänge

Generatorsequenz wird zu Generatormatrix mit 2 Generatorsequenzen

$$\vec{a} = G^T \cdot \begin{pmatrix} i \\ \vec{d} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad G = \begin{pmatrix} \vec{g}_1^T \\ \vec{g}_2^T \end{pmatrix}$$

Beispiel



Im Speicher steht wieder von links nach rechts (100) und es wird  $i = 1$  hineingeschrieben

$$\vec{a} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

## 7.3. Mehrere Ausgänge

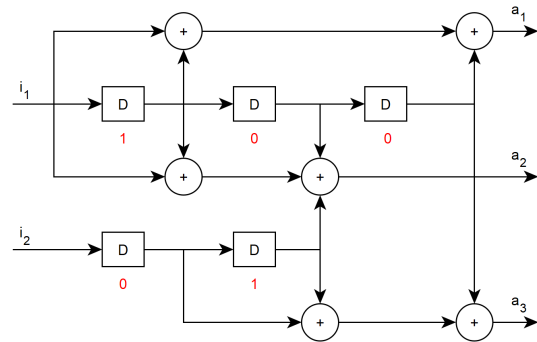
Mehrere Eingänge ( $i_1, i_2$ ) um Coderate anzupassen

für jeden Ausgang gibt es jeweils 2 Generatorsequenzen ( $g_{ij}$ )

$i$ : Eingang

$j$ : Ausgang

Beispiel



$$\vec{g}_{11}^T = (1 \ 1 \ 0 \ 1) \quad \vec{g}_{21}^T = (0 \ 0 \ 0 \ 0)$$

$$\vec{g}_{12}^T = (1 \ 1 \ 1 \ 1) \quad \vec{g}_{22}^T = (0 \ 0 \ 1 \ 0)$$

$$\vec{g}_{13}^T = (0 \ 0 \ 0 \ 1) \quad \vec{g}_{23}^T = (0 \ 1 \ 1 \ 0)$$

Zusammenfassung in  $G$

$$G = \begin{pmatrix} \overbrace{1 \ 0}^{E1 \ E2} & \overbrace{1 \ 0}^{E1 \ E2} & \overbrace{0 \ 0}^{E1 \ E2} & \overbrace{1 \ 0}^{E1 \ E2} \\ \overbrace{1 \ 0}^{D^0} & \overbrace{1 \ 0}^{D^1} & \overbrace{1 \ 1}^{D^2} & \overbrace{1 \ 0}^{D^3} \end{pmatrix} \left. \begin{array}{l} \text{Ausgang 1} \\ \text{Ausgang 2} \\ \text{Ausgang 3} \end{array} \right\}$$

## A. Hilfreiches

### A.1. Inverse in Galois-Feldern

#### Additive Inverse

gegeben:  $-3$  in  $GF(5)$

gesucht: additive Inverse

bedeutet:  $3 + x \bmod 5 = n = 0$

$n$ : neutrales Element der Addition ( $= 0$ )

$x = 2$ , da  $3 + 2 = 5$  und  $5 \bmod 5 = 0$

daher:  $-3 = 2$

oder: mit Tabelle

gegebene Zahl als Index behandeln, passenden Wert raussuchen

Index	-3	-2	-1	0	1	2	3	4	5	6
Wert	2	3	4	0	1	2	3	4	0	1

#### Multiplikative/modulare Inverse

gegeben:  $2^{-1}$  in  $GF(7)$

gesucht: multiplikative Inverse

bedeutet:  $2^1 \cdot x \bmod 7 = n = 1$

$n$ : neutrales Element der Multiplikation ( $= 1$ )

$x = 4$ , da  $2^1 \cdot 4 = 8$  und  $8 \bmod 7 = 1$

daher:  $2^{-1} = 4$

oder: mit Logarithmentafel

gegebene Potenz als Index behandeln, passenden Wert raussuchen

Index	-3	-2	-1	0	1	2	3	4	5	6
Wert	1	2	4	1	2	4	1	2	4	1

### A.2. Rechnen im Erweiterungskörper

Bsp.:  $GF(2^4)$  mit  $p(x) = x^4 + x + 1$

#### Addition

#### Multiplikation

gegeben  $5 \cdot 6 = (\alpha^2 + 1) \cdot (\alpha^2 + \alpha)$

### A.3. Syndromstellen aus Generatorpolynom

Bsp.:  $GF(5)$  mit  $\alpha = 2$

gegeben  $g(x) = x^2 + 5x + 4$

$\hookrightarrow 2$  Syndromstellen

Suchen über stures Einsetzen der Elemente des  $GF(5)$ :

$$g(\alpha^{-0}) = \alpha^{-0^2} + 3\alpha^{-0} + 2 = 1 + 3 + 2 = 1$$

$\hookrightarrow$  Position 0 keine Syndromstelle

$$g(\alpha^{-1}) = \alpha^{-1^2} + 3\alpha^{-1} + 2 = 3^2 + 3 \cdot 3 + 2 = 0$$

$\hookrightarrow$  Position 1 ist Syndromstelle

$$g(\alpha^{-2}) = \alpha^{-2^2} + 3 \cdot \alpha^{-2} + 2 = 4^2 + 3 \cdot 3^2 + 2 = 0$$

$\hookrightarrow$  Position 2 ist Syndromstelle

## A.4. ABC/PQ-Formel

ABC:  $ax^2 + bx + c = 0$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

PQ:  $x^2 + px + q$

$$x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$

## B. Polynome

### B.1. Polynommultiplikation

### B.2. Polynomdivision

allgemein:

gegeben:  $(6x^3 - 2x^2 + x + 3) : (x^2 - x + 1)$

$$\begin{array}{r|l} 6x^3 - 2x^2 + x + 3 & x^2 - x + 1 \\ -6x^3 + 6x^2 - 6x & \\ \hline 4x^2 - 5x + 3 & \\ -4x^2 + 4x - 4 & \\ \hline -x - 1 & \end{array}$$

Quotient  $q(x) = 6x + 4$

Rest  $r(x) = -x - 1$

#### Horner-Schema

zur Polynomdivision mit Linearfaktor

Rest der Division mit  $(x - x_0)$  ist Wert des Polynoms an der Stelle  $x_0$

gegeben:  $p(x) = 3x^3 + 2x^2 - 5x - 10$

$$\begin{array}{r|rrrr} x^3 & x^2 & x^1 & x^0 & \\ 3 & 2 & -5 & -10 & 2 \\ & 6 & 16 & 22 & \\ \hline & 3 & 8 & 11 & 12 \end{array}$$

Quotient:  $q(x) = 3x^2 + 8x + 11$

Rest:  $r(x) = 12 = p(2)$

## C. Lineare Algebra

### Matrix-Multiplikation

$$A \cdot B = \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix} \begin{pmatrix} g & h \\ i & j \end{pmatrix} = \begin{pmatrix} ag + bi & ah + bj \\ cg + di & ch + dj \\ eg + fi & eh + fj \end{pmatrix}$$

$$\vec{v} \cdot \vec{w}^T = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \cdot \begin{pmatrix} d & e & f \end{pmatrix} = \begin{pmatrix} a \cdot d & a \cdot e & a \cdot f \\ b \cdot d & b \cdot e & b \cdot f \\ c \cdot d & c \cdot e & c \cdot f \end{pmatrix}$$

Skalarprodukt:

$$\vec{v}^T \cdot \vec{w} = \begin{pmatrix} a & b & c \end{pmatrix} \cdot \begin{pmatrix} d \\ e \\ f \end{pmatrix} = a \cdot d + b \cdot e + c \cdot f$$

bei komplexen Vektoren:  $\vec{v}^H \cdot \vec{w}$

#### Transponieren

Zeilen werden Spalten, Spalten werden Zeilen

$$A^T = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}^T = \begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}$$

### Invertieren

für 2x2-Matrizen:

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

### Diagonale Matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

Spalten der Matrix sind Eigenvektoren

1; -5; 3 sind die Eigenwerte der Eigenvektoren

### Hermiteische Matrix

nur für quadratische Matrizen

$$A = A^H = (A^*)^T = \begin{pmatrix} 1 & 5-j & 3j \\ 5+j & 2 & 3-2j \\ -3j & 3+2j & 3+4j \end{pmatrix}$$

Eigenvektoren von hermiteschen Matrizen sind orthogonal

### Unitäre Matrix

$$A \cdot A^H = k \cdot I$$

$k$ : Skalierungsfaktor (bei skaliert unitären Matrizen)

### Toeplitz-Struktur

Eine Matrix hat Toeplitz-Struktur, wenn alle Diagonalen parallel zur Hauptdiagonalen, die gleichen Elemente enthalten:

$$T = \begin{pmatrix} 0 & -2 & -5 & -3 \\ 1 & 0 & -2 & -5 \\ 2 & 1 & 0 & -2 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

Hermiteische Toeplitz-Matrizen sind positiv oder negativ definit, abhängig vom Vorzeichen der Elemente auf der Hauptdiagonalen

### Vandermonde-Matrix

Spalten: Indizes gleich

Zeilen: Potenzen gleich

$$S = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_0^1 & x_1^1 & \dots & x_{N-1}^1 \\ x_0^2 & x_1^2 & \dots & x_{N-1}^2 \\ \dots & \dots & \dots & \dots \\ x_0^{N-1} & x_1^{N-1} & \dots & x_{N-1}^{N-1} \end{pmatrix}$$

### Determinante

nur für quadratische Matrizen

für  $2 \times 2$ -Matrix:

$$\det(A) = \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = a \cdot d - b \cdot c$$

für  $3 \times 3$ -Matrix:

$$\det(C) = \det \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \\ = a \cdot \det \begin{bmatrix} e & f \\ h & i \end{bmatrix} - b \cdot \det \begin{bmatrix} d & f \\ g & i \end{bmatrix} + c \cdot \det \begin{bmatrix} d & e \\ g & h \end{bmatrix}$$

### Rang einer Matrix

Eine Matrix hat vollen Rang, wenn die Determinante ungleich 0 ist

Ist die Determinante gleich 0, ist die Matrix/das Gleichungssystem überbestimmt

### Eigenvektoren/ Eigenwerte

Eigenvektoren einer Matrix werden bei einer Matrixtransformation nur in ihrer Länge geändert, nicht in ihrer Richtung

Faktor, um den ein Eigenvektor gedehnt oder gestaucht wird, ist der zum Eigenvektor zugehöriger Eigenwert  $\lambda$

$$A\vec{v} = \lambda\vec{v}$$

$$(A - \lambda I)\vec{v} = \vec{0}$$

$A$ : Matrix

$\vec{v}$ : Eigenvektor

$\lambda$ : Eigenwert(e)

$I$ : Einheitsmatrix

Eigenwerte von positiv (oder negativ) definiten Matrix sind immer positiv (oder negativ)

## D. Digitale Signalverarbeitung

### Diskretisierung und Fensterung

Diskretisierung  $\circ \rightarrow \bullet$  Periodische Fortsetzung

Diskretisierung  $\bullet \rightarrow \circ$  Periodische Fortsetzung

Begrenzung  $\rightarrow$  Leck-Effekt

### Fourier-Transformation (kontinuierlich)

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi f t} dt$$

### DFT

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{nk}{N}}$$

$n$ : Frequenzindex

$k$ : Zeitindex

### Auflösung DFT

$$\Delta f = \frac{f_a}{N} = \frac{1}{t_a \cdot N} = \frac{1}{\Delta t}$$

$\Delta f$ : spektrale Auflösung

$f_a$ : Abtastfrequenz

$t_a$ : Abtastzeit

$N$ : Anzahl Abtastwerte

$\Delta t$ : Messdauer

### Fensterung

Fensterung im Zeitbereich  $\rightarrow$  Multiplikation mit Fensterfunktion  $\circ \rightarrow \bullet$  Faltung mit zur Fensterfunktion zugehörigem Spektrum

### Dirichlet-Kern

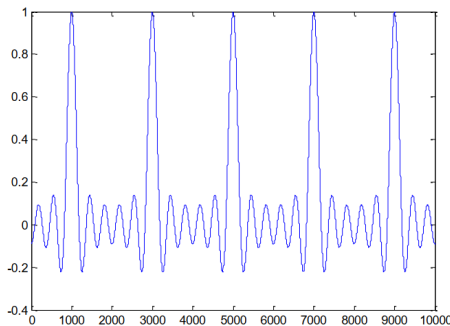
abgetastete Rechteckfunktion  $\circ \rightarrow \bullet$  Dirichlet-Kern

Definition der Dirichlet-Kerns der Länge  $N + 1$ :

$$D(x) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} e^{jnx} = \frac{\sin\left(\left(\frac{N+1}{2}\right)x\right)}{\sin\left(\frac{x}{2}\right)}$$

$$x = 2\pi f T_a$$

Bsp: 11:



Eigenschaften:

Hauptwert hat Höhe  $N + 1$

Nullstellen, bei  $f = \frac{f_a}{N+1} \cdot k$  für  $k \in \mathbb{N}$

Hauptwert periodisch mit  $f_a$

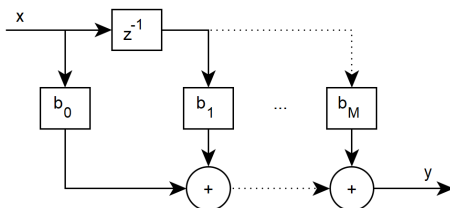
### z-Transformation

$$z = e^{j2\pi \frac{f}{f_a}}$$

$$H(f) = H(z) \Big|_{z=e^{j2\pi \frac{f}{f_a}}}$$

### Reiner FIR-Filter

kanonische Form

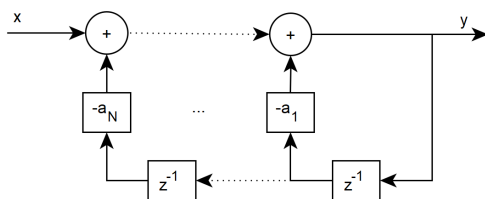


$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 \cdot z^{-1} + \dots + b_M \cdot z^{-M}$$

$$y(t) = b_0 \cdot x(t) + b_1 \cdot x(t-1) + \dots + b_M \cdot x(t-M)$$

### Reiner IIR-Filter

kanonische Form



$$H(z) = \frac{1}{1 + a_1 \cdot z^{-1} + \dots + a_N \cdot z^{-N}}$$

$$y(t) = x(t) - a_1 \cdot y(t-1) - \dots - a_n \cdot y(t-N)$$

## E. Wahrscheinlichkeitstheorie

### E.1. Satz von Bayes

$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A) = P(A, B)$$

$$P(A|B) = \frac{P(B|A)}{P(B)} \cdot P(A)$$

### E.2. Kombinatorik

#### E.2.1. Permutation

##### Ohne Wiederholung

Anordnung aller möglicher, unterscheidbarer Ereignisse, kein Ereignis tritt doppelt auf

$$M = N!$$

**Beispiel:** In einem Glas sind 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, alle aus dem Glas zu nehmen? (ohne Zurücklegen)

$$\hookrightarrow M = N! = 5! = 120$$

##### Mit Wiederholung

Anordnung aller möglicher Ereignisse, von welchen manche nicht unterscheidbar sind

$$M = \frac{N!}{K_1! \cdot \dots \cdot K_s!}$$

$K_x$ : Gibt an, wie viele ununterscheidbare Ereignisse es pro Ereignis gibt

**Beispiel:** In einem Glas liegen 2 blaue, 2 rote und ein grünes Bonbon. Wie viele Möglichkeiten gibt es alle aus dem Glas zu nehmen? (Ohne Zurücklegen)

$$\hookrightarrow \frac{5!}{2! \cdot 2! \cdot 1!} = 30$$

$N$ : Anzahl der möglichen Ereignisse  
 $M$ : Anzahl der Permutationen

#### E.2.2. Variation

##### Ohne Wiederholung

Zusammenstellung von  $K$  Ereignissen; Reihenfolge relevant

$$M = \frac{N!}{(N-K)!}$$

**Beispiel:** In einem Glas liegen 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, 3 Bonbons herauszunehmen? (Ohne Zurücklegen)

$$\hookrightarrow M = \frac{5!}{(5-2)!} = \frac{5!}{2!} = 60$$

##### Mit Wiederholung

$$M = N^K$$

**Beispiel:** In einem Glas liegen 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, 3 Bonbons herauszunehmen? (Mit Zurücklegen)

$$\hookrightarrow M = 5^3 = 125$$

$M$ : Anzahl der Variationen

$N$ : Anzahl der möglichen Ereignisse

$K$ : Anzahl der zusammengestellten Ereignisse (Ordnung)

### E.2.3. Kombination

Wie Variation, Reihenfolge aber irrelevant (vgl. Lottoziehung)

#### Ohne Wiederholung

$$M = \binom{N}{K} = \frac{N!}{K! \cdot (N - K)!}$$

#### Mit Wiederholung

$$M = \binom{N + K - 1}{K} = \frac{(N + K - 1)!}{K! \cdot (N - 1)!}$$

$M$ : Anzahl der Kombinationen

$N$ : Anzahl der möglichen Ereignisse

$K$ : Anzahl der zusammengestellten Ereignisse (Ordnung)