

This document is an exam summary that follows the script of the Communication Systems lecture at ETH Zurich. The contribution to this is a short summary that includes the most important formulas, algorithms and proof ideas. This summary was created during the fall semester 2017. Due to updates to the syllabus content, some material may no longer be relevant for future versions of the lecture.

This work is published as CC BY-NC-SA.



I do not guarantee correctness or completeness, nor is this document endorsed by the lecturers. Feel free to point out any erratas.

CommSys

Yannick Merkli HS 2017

1 Wahrscheinlichkeitstheorie

1.1 Theorie

PDF (probability density function):

$$\int_{-\infty}^{\infty} f_X(x) \cdot dx \stackrel{!}{=} 1, \quad f_X(x) \geq 0, \quad \forall x$$

CDF (cumulative distribution function):

$$F_X(x) = P[X \leq x] = \int_{-\infty}^x f_X(x') \cdot dx'$$

Bedingte Wahrscheinlichkeit:

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

$$f_{X,Y}(x,y) = f_{X|Y}(x|y) \cdot f_Y(y) = f_{Y|X}(y|x) \cdot f_X(x)$$

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x,y') \cdot dy'$$

Formel von Bayes:

$$P[B|A] = \frac{P[A|B] \cdot P[B]}{P[A]} = \frac{P[A,B]}{P[A]}$$

Unabhängigkeit:

$$X, Y \text{ unabhängig} \Rightarrow \begin{aligned} f_{X,Y}(x,y) &= f_X(x) \cdot f_Y(y) \\ f_{X|Y}(x|y) &= f_X(x) \end{aligned}$$

Erwartungswert:

Definition:

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) \cdot dx$$

Funktion einer ZV:

$$E[Z] = E[g(X)] = \int_{-\infty}^{\infty} g(x) \cdot f_X(x) \cdot dx$$

Funktion zweier ZV:

$$E[Z] = E[g(X,Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) \cdot f_{X,Y}(x,y) \cdot dx dy$$

Wenn die ZV stat. unabhängig sind:

$$E[g(X,Y)] = E_X \left[\int_{-\infty}^{\infty} g(x,y) \cdot f_Y(y) \cdot dy \right]$$

Linearkombination von ZV:

$$E[\alpha X + \beta Y] = \alpha E[X] + \beta E[Y]$$

Unabhängigkeit:

$$X, Y \text{ unabhängig} \Rightarrow E[X \cdot Y] = E[X] \cdot E[Y]$$

Für einen Zufallsprozess $X(t)$ gilt

$$E[X(t_k)] = \mu_X(t_k) = \int_{-\infty}^{\infty} x(t_k) \cdot f_{X(t)}(x(t_k)) \cdot dx(t_k)$$

Ist der Zufallsprozess WSS, dann

$$E[X(t_k)] = \mu_X(t_k) = \mu_X \quad \forall t_k$$

Varianz:

$$\sigma^2 = \text{var}[X] = E[(X - \mu_X)^2] = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x) \cdot dx = E[X^2] - \mu_X^2$$

Für einen Zufallsprozess $X(t)$ gilt

$$\begin{aligned} \text{var}[X(t_k)] &= \sigma_X^2(t_k) = \int_{-\infty}^{\infty} (x(t_k) - \mu_X(t_k))^2 \cdot f_{X(t)}(x(t_k)) \cdot dx(t_k) \\ &= C_X(t, t) \end{aligned}$$

Korrelation:

$$E[X \cdot Y] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xy \cdot f_{X,Y}(x,y) \cdot dx dy$$

Für zwei Zufallsprozesse $X(t)$ und $Y(t)$ heisst dies Kreuzkorrelation:

$$R_{XY}(t_1, t_2) = E[X(t_1)Y(t_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t_1) \cdot y(t_2) \cdot f_{X(t_1), Y(t_2)}(x(t_1), y(t_2)) \cdot dx(t_1) dy(t_2)$$

Sind $X(t)$ und $Y(t)$ stationäre Zufallsprozesse, so gilt

- $R_{XY}(t_1, t_2) = R_{XY}(t_2 - t_1) = R_{XY}(\tau)$
- $R_{XY}(\tau) = R_{YX}(-\tau)$

Kovarianz:

$$\text{cov}(X, Y) = \sigma_{X,Y} = E[(X - \mu_X)(Y - \mu_Y)] = E[X \cdot Y] - \mu_X \mu_Y$$

Es gilt

$$\text{stat. unabhängig} \Rightarrow \text{cov}(X, Y) = 0$$

Aber:

Wenn X, Y gaussverteilt und $\sigma_{X,Y} = 0 \Rightarrow X, Y$ unabhängig

2D-Verteilung: Gemeinsame Dichte

$$P[a_1 \leq X \leq b_1 \cap a_2 \leq Y \leq b_2] = \int_{a_2}^{b_2} \int_{a_1}^{b_1} f_{X,Y}(x,y) \cdot dx dy$$

2 Gauss'sche Normalverteilung

PDF von $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{(x - \mu_X)^2}{2\sigma_X^2}\right)$$

Multivariate Gaussverteilung:

Es seien x_1, x_2, \dots, x_n Realisationen der gaussverteilten ZV X_1, X_2, \dots, X_n . Dann schreibt sich der Beobachtungsvektor $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$ und damit die gemeinsame Verteilung:

$$f_{X_1, X_2, \dots, X_n}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

$$\boldsymbol{\mu}^T = [\mu_X(t_1), \dots, \mu_X(t_n)]^T$$

mit der Kovarianzmatrix Σ , deren Einträge wie folgt definiert sind:

$$\Sigma_{ij} = \text{cov}(X_i, X_j)$$

Im 2D-Fall:

$$f_{X,Y}(x,y) = \frac{1}{2\pi\sqrt{|\Sigma|}} \cdot e^{-\frac{1}{2} \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \Sigma^{-1} \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}}$$

$$\text{mit } \Sigma = \begin{bmatrix} \text{cov}(X,X) & \text{cov}(X,Y) \\ \text{cov}(Y,X) & \text{cov}(Y,Y) \end{bmatrix}$$

Linearkombination von Gaussverteilungen:

Sei $Y = \sum_i a_i X_i$ eine Linearkombination von gaussverteilten ZV X_i mit Erwartungswerten μ_{X_i} und Varianzen $\sigma_{X_i}^2$ und Kovarianzen $\sigma_{X_i X_j} = \text{cov}(X_i, X_j)$.

Dann gilt $Y \sim \mathcal{N}(\sum_i a_i \mu_{X_i}, \sum_i a_i^2 \sigma_{X_i}^2 + \sum_i \sum_{j \neq i} a_i a_j \sigma_{X_i X_j})$.Falls die X_i unabhängig sind, dann gilt $Y \sim \mathcal{N}(\sum_i a_i \mu_{X_i}, \sum_i a_i^2 \sigma_{X_i}^2)$.

2.1 Zufallsprozesse

Zufallsprozess $X(t)$ Ereignisraum bestehend aus Zeitfunktionen $\{x_1(t), \dots, x_n(t)\}$, und beobachtet zu einer fixen Zeit t_k .

- Mittelwert Funktion**

$$\mu_X(t_k) = E[X(t_k)] = \int_{-\infty}^{\infty} x \cdot f_{X(t_k)}(x) dx$$

- Autokorrelation**

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] = \iint_{-\infty}^{\infty} x_1 x_2 \cdot f_{X(t_1), X(t_2)}(x_1, x_2) dx_1 dx_2$$

Achtung! Bevor Definition angewendet wird, erst prüfen, ob mit Regeln des Erwartungswerts gerechnet werden kann! (Wahrscheinlich muss nämlich kein Integral ausgerechnet werden!)

Mittelwertsschätzung: $\hat{\mu}_X(T) = \frac{1}{2T} \int_{-T}^T x(t) dt$, $x(t)$: Samplepkt. von $X(t)$

Für WSS Zufallsprozesse gilt:

- $R_X(t_1, t_2) = R_X(t_2 - t_1) = R_X(\tau)$
- $R_X(\tau = 0) = E[X(t)X(t)] = E[X^2(t)]$ (durchschn. Energie)
- $\text{var}[X(t)] = E[X^2(t)] - \mu_X^2(t) = R_X(0) - \mu_X^2$
- $R_X(\tau) = R_X(-\tau)$ und $|R_X(\tau)| \leq R_X(0)$

Wenn $X(t)$ ein stationärer Gaussprozess ist, dann ist er SSS.

Autokovarianz

$$\text{cov}(X(t_1), X(t_2)) = C_X(t_1, t_2) = E[(X(t_1) - \mu_X(t_1))(X(t_2) - \mu_X(t_2))] = R_X(t_1, t_2) - \mu_X(t_1)\mu_X(t_2)$$

Power Spectral Density (PSD), Unit [W/Hz] Falls wir die Autokorrelationsfunktion in den Fourier Raum transformieren:

$$R_X(\tau) \leftrightarrow S_X(f) = \int_{-\infty}^{\infty} R_X(\tau) \cdot e^{-j2\pi f \tau} d\tau$$

$$R_X(\tau) = \int_{-\infty}^{\infty} S_X(f) \cdot e^{j2\pi f \tau} df$$

Eigenschaften:

- $E[X^2(t)] = R_X(0) = \int_{-\infty}^{\infty} S_X(f) df$
- $S_X(-f) = S_X(f)$
- $S_X(f) \geq 0, \quad \forall f$

2.2 Gaussprozess

 $X(t)$ ist ein Gaussprozess,

- wenn $\forall t_1, t_2, \dots, t_k$ die gemeinsame Verteilung der ZV $\{X(t_1), X(t_2), \dots, X(t_k)\}$ eine multivariate Gaussverteilung ist.
- gdw alle Linearkombinationen der ZV $X(t_1), X(t_2), \dots, X(t_k)$ gaussverteilt sind.
- D.h. wenn $\forall t_1, t_2, \dots, t_k$ und $\forall c_1, c_2, \dots, c_k \in \mathbb{R}$ die ZV $Z = c_1 X(t_1) + c_2 X(t_2) + \dots + c_k X(t_k)$ eine univariate Gaussverteilung hat.

Ein Gaussprozess ist alleine durch den Mittelwert $E[X(t)] = \mu_X(t)$ und die Kovarianz $C_X(t_1, t_2)$ bestimmt, weil diese ausreichen um, die multivariate Gaussverteilung der ZV $\{X(t_1), X(t_2), \dots, X(t_k)\}$ aufzustellen. (Dafür braucht's weder Ergodizität, noch, dass der Prozess stationär ist.)

Eigenschaften:

- Wenn $X(t)$ stationär, dann ist er auch SSS (gilt nicht allgemein!)
- Ein stationärer Gaussprozess mit kontinuierlichem Spektrum ist ergodisch.
- AWGN ist komplett durch einen konstanten Mittelwert μ_X und die Autokovarianz $C_X(\tau)$ bzw. die Autokorrelation $R_X(\tau)$ definiert.
- Wenn ein Gaussprozess LTI-gefiltert wird, ist der resultierende Prozess ebenfalls Gauss'sch.

2.2.1 How-To

Gemeinsame Verteilung von zwei Beobachtungen eines Gaussprozesses:

- Check, dass die beiden ZV Gaussverteilt sind.
- Multivariate Gaussverteilung hinschreiben
- Mittelwert der beiden ZV berechnen
- Kovarianzmatrix aufstellen (Autokorrelation benötigt)

2.3 stationäre Prozesse

SSS (strict sense stationary):

Falls $X(t)$ und $X(t+\tau)$ dieselbe Statistik für alle τ haben. SSE \Rightarrow WSS
Zudem: $\mu_X(t) = \mu_X \quad \forall t$ (const.)

WSS (wide sense stationary):

Wenn für einen Prozess folgende zwei Bedingungen gelten:

- $\mu_X(t) = \mu_X \quad \forall t$
- $R_X(t_1, t_2) = R_X(t_1 - t_2) = R_X(\tau) \quad \forall t_1, t_2$
- $R_X(\tau) = R_X(-\tau)$
- $|R_X(\tau)| \leq R_X(0)$

dann bezeichnet man den Prozess WSS und er hat eine Autokovarianz:

$$C_X(t_1, t_2) = E[(X(t_1) - \mu_X)(X(t_2) - \mu_X)] = R_X(t_1, t_2) - \mu_X^2$$

2.4 Ergodische Prozesse

Der Mittelwert eines Ensembles $E[X(t_k)]$ und der zeitliche Mittelwert einer Realisation $x(t)$ sind gleich.D.h. ich kann eine Realisation aufintegrieren und der so erhaltene Schätzer $\hat{\mu}_X(T) = \int_0^T x(t) \cdot dt$ wird immer genauer, je grösser T ist.

- $\lim_{T \rightarrow \infty} \hat{\mu}_X(T) = \mu_X$
- $\lim_{T \rightarrow \infty} \text{var}[\hat{\mu}_X(T)] = 0$

Achtung! Ergodizität impliziert nicht stationär und umgekehrt!

Bsp. eines ergodischen Prozesses: Würfel

$$\begin{aligned} x_1[n] &= \begin{pmatrix} 1, & 2, & 4, & 1, & 1, & 3, & \dots \end{pmatrix} \phi_t \\ x_2[n] &= \begin{pmatrix} 3, & 2, & 2, & 1, & 3, & 6, & \dots \end{pmatrix} \\ x_3[n] &= \begin{pmatrix} 6, & 1, & 6, & 3, & 4, & 5, & \dots \end{pmatrix} \\ &\vdots \\ x_k[n] &= \dots \dots \dots \phi_s \end{aligned}$$

Wenn die Mittelwerte ϕ_s und ϕ_t übereinstimmen, so ist der Prozess ergodisch, was hier im Falle von Würfeln der Fall ist.

2.5 Errorfunction

$$\text{erfc}(u) = \frac{2}{\sqrt{\pi}} \int_u^{\infty} \exp(-z^2) dz$$

$$\text{erf}(u) = 1 - \text{erfc}(u) = \frac{2}{\sqrt{\pi}} \int_0^u \exp(-z^2) dz$$

$$\Rightarrow \text{erfc}(0) = \frac{2}{\sqrt{\pi}} \int_0^{\infty} \exp(-z^2) dz = 1, \quad \text{erfc}(u) < \frac{e^{-u^2}}{\sqrt{\pi}u} \quad (u > 0)$$

$$Q(x) = \frac{1}{2} \text{erfc}\left(\frac{1}{\sqrt{2}} x\right) \quad \text{erfc}(x) = 2Q(\sqrt{2}x)$$

2.6 White Noise

Wird benutzt, um thermal noise zu modellieren.

$$E[W] = 0, \quad E[W^2] = \sigma_W^2 = \frac{N_0}{2}$$

$$R_W(\tau) = \frac{N_0}{2} \delta(\tau) \leftrightarrow S_W(f) = \frac{N_0}{2}$$

White Gaussian Noise wird als gaussverteilt angenommen. Die Verteilung ist gemäss der multivariaten Gaussverteilung.

2.7 Gefilterte Zufallsprozesse

Prinzipiell lässt sich jede Eingangs- / Ausgangsbeziehung mittels einer Faltung im Zeitbereich darstellen, sofern das System LTI ist.

$$X(t) \xrightarrow{h(t)} Y(t) \quad Y(t) = \int_{-\infty}^{\infty} h(\tau) X(t - \tau) \cdot d\tau$$

Jedoch ist diese Beschreibung bei (statischen) Zufallsprozessen nicht sinnvoll. Man beschränkt sich daher darauf, statische Gaussprozessen, und damit Mittelwert und die Autokorrelation, zu betrachten (was für eine statistische Beschreibung in den allermeisten Fällen ausreicht).

Mittelwert:

$$\mu_Y = E[Y(t)] = \dots = \int_{-\infty}^{\infty} h(\tau) E[X(t - \tau)] \cdot d\tau = \mu_X \int_{-\infty}^{\infty} h(\tau) \cdot d\tau \\ \Rightarrow \mu_Y = \mu_X \cdot H(f = 0)$$

Autokorrelation:

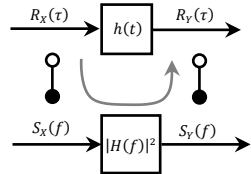
Für einen stationären Prozess $X(t)$ gilt

$$R_Y(\tau) = \iint_{-\infty}^{\infty} h(\tau_1) h(\tau_2) R_X(\tau - \tau_1 + \tau_2) \cdot d\tau_1 d\tau_2$$

LTI Input Output Beziehung:

$$S_Y(f) = |H(f)|^2 S_X(f)$$

Die Autokorrelation direkt im Zeitbereich auszurechnen ist jedoch schwierig, weshalb zuerst über den Frequenzbereich, und damit die PSD (power spectral density, spektrale Leistungsdichte) ausgewichen wird.



Man berechnet also

$$R_Y(\tau) = \mathcal{F}^{-1}\{S_Y\}(\tau) = \int_{-\infty}^{\infty} S_Y(f) \cdot e^{2\pi i f \tau} \cdot df$$

$$R_X(\tau) = R_X(t_1, t_2) = E[X(t_1)X(t_2)]$$

$$S_Y(f) = |H(f)|^2 \cdot S_X(f)$$

$$S_X(f) = \mathcal{F}\{R_X\}(f) = \int_{-\infty}^{\infty} R_X(\tau) \cdot e^{-2\pi i f \tau} \cdot d\tau$$

Die Leistung am Ausgang ist dann gegeben durch:

$$E[Y^2(t)] = \int_{-\infty}^{\infty} |H(f)|^2 \cdot S_X(f) \cdot df$$

$$S_X(0) = \int_{-\infty}^{\infty} R_X(\tau) \cdot d\tau$$

$$E[X^2(t)] = R_X(0) = \int_{-\infty}^{\infty} S_X(f) \cdot df$$

Bandbegrenztes weisses Gauss'sches Rauschen:

Sei PSD des Rauschens $S_W(f) = \frac{N_0}{2}$ und der Mittelwert $\mu_W(t) = 0$.

Dann ist PSD des gefilterten Rauschens $S_N(f) = \begin{cases} \frac{N_0}{2} & |f| \leq B \\ 0 & \text{sonst} \end{cases}$ und damit $R_N(\tau) = N_0 B \text{sinc}(2B\tau)$.

Die Varianz eines Samples berechnet sich zu

$$\text{Var}[N(t)] = R_N(0) - \mu_N^2 \stackrel{\mu_N=0}{=} R_N(0) = N_0 \cdot B = \sigma_N^2$$

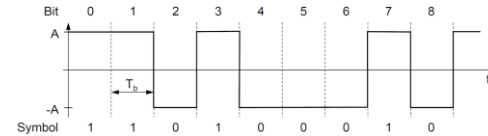
Also ist das Rauschsample Gaussverteilt mit $N(t_k) \sim \mathcal{N}(0, \sqrt{N_0 B})$. Falls mit einer Rate von $2B$ abgetastet wird, sind zwei Samples $N(t_1)$ und $N(t_2)$ statistisch unabhängig, weil die Autokorrelation bei diesem Delay verschwindet. (slide 1.32)

2.7.1 PSD berechnen

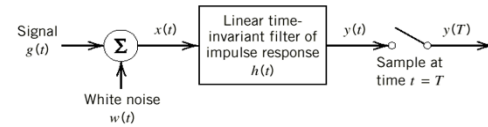
- vergewissern, dass stationärer Prozess vorliegt, also check
- $\mu_X(t) = \text{const.}$
- $R_X(t_1, t_2) = R_X(\tau)$
- wenn Input PSD gefragt:
 - noch $\mathcal{F}\{R_X(\tau)\} = S_X(f)$ berechnen
- wenn Output PSD gefragt:
 - $S_Y(f) = |H(f)|^2 \cdot S_X(f)$, wobei $H(f) = \mathcal{F}\{h(t)\}$

3 Baseband Pulse Transmission

NRZ Signal



3.1 Matched Filter (S.248ff)



Signal

Wir gehen davon aus, dass die Signalform $x_0(t)$ dadurch entsteht, dass ein Sendebit (modelliert als $\delta(t)$) mit einem Sendefilter $g(t)$ gefaltet wird.

$$x_0(t) = (\delta * g)(t) = g(t)$$

AWGN Channel

$$x(t) = g(t) + w(t), \quad 0 \leq t \leq T_b$$

$$y(t) = (x * h)(t) = ((g + w) * h)(t) = (g * h)(t) + (w * h)(t) = g_0(t) + n(t)$$

Matched Filter

Ziel: maximieren der empfangenen Nutzsignalenergie im Vergleich zu Rauschenergie

$$\max\{\eta\} = \max\left\{\frac{|g_0(T)|^2}{E[n^2(t)]}\right\}$$

$g_0(T)$ instantane Leistung des Nutzsignals zum Samplingzeitpunkt
 $\rightarrow g_0(T) = (g * h)(T)$ wobei g : Sendefilter, h : Empfangsfilter
 $E[n^2(t)]$ durchschn. Leistung des Rauschens (stationär) $n(t) = (c * w)(t)$
 $\rightarrow E[n^2(t)] = \frac{N_0}{2} \int_{-\infty}^{\infty} |h(t)|^2 dt$ wobei h : Empfangsfilter

Umformen der Terme (Betrachtung im Frequenzbereich) und Anwenden der Schwarz-Ungleichung resultiert in

$$H_{\text{opt}}(f) = k \cdot G^*(f) \cdot e^{-2\pi i f T} \quad \leftrightarrow \quad h_{\text{opt}}(t) = k \cdot g(T - t)$$

Damit wird

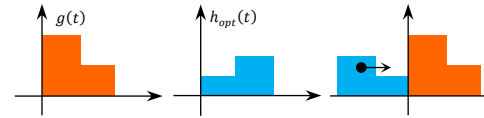
$$\text{SNR} = \frac{|g_0(T)|^2}{E[n^2(t)]} = \eta_{\text{max}} = \frac{2}{N_0} \int_{-\infty}^{\infty} |G(f)|^2 \cdot df = \frac{2E}{N_0}$$

Für Rechteckimpulse mit Energie $E = A^2 T$ (Amplitude A , Länge T), können wir den Matched Filter mit einem integrate and dump circuit designen, i.e. wir berechnen die Fläche unter dem Impuls und sampeln @ $t = T$. Der Wert ist dann $kA^2 T$.

3.1.1 How-To

Finde Matched Filter zu Eingangssignal und die resultierende Impulsantwort

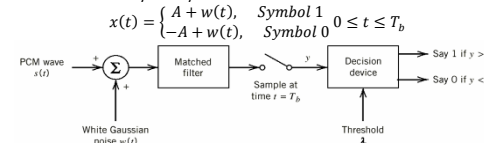
1. Zeichne $g(t)$
2. Zeichne das Matched Filter $h_{\text{opt}}(t)$:
 $g(t)$ um Nullpunkt spiegeln und um k skalieren
3. Grafische Faltung von $(g * h_{\text{opt}})(t)$



Bei der Amplitudenberechnung nicht die Zeitachse vergessen! (z.B. $\frac{T}{2}$)

3.2 Error of Threshold detector

Betrachte ein binary NRZ system

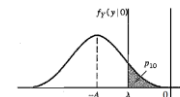


Mit AWGN in y ein Sample der ZV $Y \sim \mathcal{N}(-A, \frac{N_0}{2T_b})$ (S.254/255).

Symbol 1 wurde gesendet da noise gaussverteilt ist, ist der sample value y einer ZV Y auch gaussverteilt:

$$y = \frac{1}{T_b} \int_0^{T_b} x(t) dt = A + \frac{1}{T_b} \int_0^{T_b} w(t) dt$$

- $E[Y] = \pm A$
- $\sigma_y^2 = E[(Y - A)^2] = \frac{1}{T_b^2} \iint_0^{T_b} E[w(t)w(u)] dt du = \frac{1}{T_b^2} \iint_0^{T_b} R_w(t, u) dt du = \frac{N_0}{2T_b}$
- $R_w(t, u) = \frac{N_0}{2} \delta(t - u)$



Interessant ist die W' keit, dass man auf ein falsches Symbol entscheidet, also z.B. gesendet 0, entschieden 1:

$$p_{10} = P[y > \lambda | \text{"0" gesendet}]$$

$$= \frac{1}{\sqrt{\pi \frac{N_0}{2T_b}}} \cdot \int_{\lambda}^{\infty} \exp\left(-\frac{(y + A)^2}{\frac{N_0}{2T_b}}\right) \cdot dy = \frac{1}{2} \text{erfc}\left(\frac{A + \lambda}{\sqrt{\frac{N_0}{2T_b}}}\right)$$

umgekehrt

$$p_{01} = \frac{1}{\sqrt{\pi \frac{N_0}{2T_b}}} \cdot \int_{-\infty}^{\lambda} \exp\left(-\frac{(y - A)^2}{\frac{N_0}{2T_b}}\right) \cdot dy = \frac{1}{2} \text{erfc}\left(\frac{A - \lambda}{\sqrt{\frac{N_0}{2T_b}}}\right)$$

Maximiere die gesamte Fehlerwahrscheinlichkeit $P_e = p_0 \cdot p_{10} + p_1 \cdot p_{01}$ um λ_{opt} zu finden: $\lambda_{\text{opt}} = \frac{N_0}{4AT_b} \ln\left(\frac{p_0}{p_1}\right)$

Die gesamte Fehlerwahrscheinlichkeit ist:

$$P_e = p_0 \cdot p_{10} + p_1 \cdot p_{01} \\ = \frac{p_0}{2} \text{erfc}\left(\frac{A + \lambda}{\sqrt{\frac{N_0}{2T_b}}}\right) + \frac{p_1}{2} \text{erfc}\left(\frac{A - \lambda}{\sqrt{\frac{N_0}{2T_b}}}\right) \stackrel{p_0=p_1}{=} \frac{1}{2} \text{erfc}\left(\frac{E_b}{\sqrt{N_0}}\right)$$

Mit Energie pro Bit $E_b = A^2 T_b$

Falls die Signalarate erhöht werden soll $\rightarrow T_b$ (und somit auch E_b) verkleinern sich (Bsp: Doppelte Signalarate \rightarrow halbe Bitdauer)

3.2.1 Complementary Error Function

Sei die Dichtefunktion einer Normalverteilten Zufallsvariable

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{(x - \mu_X)^2}{2\sigma_X^2}\right)$$

dann ist die Wahrscheinlichkeit, dass die ZV den Schwellenwert λ überschreitet

$$P[X > \lambda] = \int_{\lambda}^{\infty} f_X(x) \cdot dx \\ = \frac{1}{\sqrt{2\pi}\sigma_X} \int_{\lambda}^{\infty} \exp\left(-\frac{(x - \mu_X)^2}{2\sigma_X^2}\right) \cdot dx \\ = \frac{1}{\sqrt{2\pi}\sigma_X} \int_{\lambda}^{\infty} \exp\left(-\left(\frac{x - \mu_X}{\sqrt{2}\sigma_X}\right)^2\right) \cdot dx \\ = \frac{1}{2\sqrt{\pi}} \int_{\frac{\lambda - \mu_X}{\sqrt{2}\sigma_X}}^{\infty} \exp(-x'^2) \cdot dx' \\ = \frac{1}{2} \cdot \text{erfc}\left(\frac{\lambda - \mu_X}{\sqrt{2}\sigma_X}\right)$$

Dabei wurde die Regel

$$\frac{2}{\sqrt{\pi}} \int_u^{\infty} \exp(-z^2) \cdot dz = \text{erfc}(u)$$

verwendet.

3.2.2 Energie pro Bit

Falls nach der Energie pro Bit/Symbol bei einem Baseband System gefragt ist: Berechne Energie des/ der Sendepuls(e): $E_i = \int_0^{T_b} p_i^2(t) dt$

Falls mehrere Sendepulse: $E_s = P_1 E_1 + \dots + P_n E_n$, wobei P_i die W' keit, dass Symbol (und somit Puls) i gesendet wird.

Falls ein Bit pro Symbol codiert $\rightarrow E_b = E_s$

3.3 Intersymbol-Interferenz (ISI)

Das empfangene und durch den Empfangsfilter angepasste Signal $y(t)$ wird zuerst abgetastet. Die Multiplikation von $y(t)$ mit einem Kammsignal ergibt eine Repetition des Spektrums im Frequenzbereich.

$$y_{\delta}(t) = y(t) \cdot \sum_k \delta(t - kT_b) \quad \leftrightarrow \quad Y_{\delta}(f) = \frac{1}{T_b} \cdot \sum_k Y(f - \frac{k}{T_b})$$

Wenn man davon ausgeht, dass $y(t)$ aus einer Folge von Pulsen plus AWGN besteht, also

$$y(t) = \mu \sum_k a_k p(t - kT_b) + n(t)$$

und die Pulse $p(t)$ aus der Faltung eines Deltastosses (repräsentativ für ein Bit) mit dem Sendefilter und dem Kanal entsteht,

$$\mu \cdot p(t) = (\delta * g * c * h)(t) \quad \leftrightarrow \quad \mu \cdot P(f) = 1 \cdot G(f) \cdot C(f) \cdot H(f)$$

dann ist die Amplitude am Eingang des Samplers zu einem Abtastzeitpunkt durch

$$y(t_i = i \cdot T_b) = \mu \cdot \sum_k a_k p[(i - k)T_b] + n(t_i) \\ = \mu a_i p(0) + \mu \cdot \sum_{k \neq i} a_k p[(i - k)T_b] + n(t_i)$$

gegeben.

Normalisation der Amplitude des Pulses auf $p(0) \stackrel{!}{=} 1$ vereinfacht den ersten Term der rechten Seite noch auf $\mu a_i p(0) = \mu a_i$.

Lässt man in dieser Betrachtung nun den Einfluss des Rauschens ausser Acht und fordert, dass ein Puls nur zum Zeitpunkt seiner Abtastung einen Einfluss auf den Sampler haben soll, ergibt sich

$$n(t_i) \approx 0, \quad p(iT_b - kT_b) \stackrel{!}{=} \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \quad \Rightarrow \quad y(t_i) = \mu a_i$$

Welches $P(f)$ erfüllt diese Anforderungen?

Dazu ruft man sich in Erinnerung, dass jeder Puls gesampelt wird mit Rate $R_b = 1/T_b$. Also

$$p_b(t) = p(t) \cdot \sum_k \delta(t - kT_b) \Leftrightarrow P_b(f) = \frac{1}{T_b} \cdot \sum_k P\left(f - \frac{k}{T_b}\right)$$

Nutzt man die Definition der Fouriertransformation, kommt man auf eine zweite Gleichung

$$P_b(f) = \int_{-\infty}^{\infty} p(t) \cdot \sum_k \delta(t - kT_b) \cdot e^{-2\pi i f t} dt$$

Unter Ausnutzung der Anforderungen an $p(t)$ und der Siebeigenschaft der Deltafunktion muss schliesslich gelten (**Nyquist-Kriterium**)

$$P_b(f) = p(0) = 1 \Leftrightarrow T_b \cdot \sum_{k=-\infty}^{\infty} P\left(f - \frac{k}{T_b}\right)$$

In Worten: damit die Pulsform genau gleich 1 zum relevanten Abtastzeitpunkt und 0 bei jedem anderen ist, muss dessen Spektrum durch den Sende- und Empfangsfilter so hingebogen werden, dass das periodische Spektrum des Pulses gerade eine Konstante, nämlich T_b , ergibt.

Der Frequenzgang eines idealen Nyquistkanals erfüllt diese Bedingung, denn

$$P(f) = \begin{cases} \frac{1}{2W}, & |f| \leq W \\ 0, & \text{sonst} \end{cases} \Leftrightarrow p(t) = \text{sinc}(2Wt)$$

Ein Nyquistpuls wird damit definiert als:

$$P_{\text{nyq}}(f) = \frac{1}{2W} \text{rect}\left(\frac{f}{2W}\right) = \begin{cases} \frac{1}{2W}, & |f| \leq W \\ 0, & \text{sonst} \end{cases} \Leftrightarrow T_b, |f| \leq \frac{1}{2T_b}$$

$$W = \frac{R_b}{2} = \frac{1}{2T_b}$$

Die Nullstellen des sinc sind genau zu den Abtastzeitpunkten anderer Pulse.

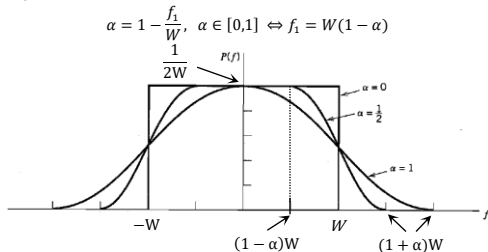
Ein idealer Nyquistkanal ist lediglich ein theoretisches Konstrukt. Die sinc-Funktion erstreckt sich zu beiden Seiten der Zeitachse ins Unendliche. Ausserdem fallen die Überschwinger nur mit $\frac{1}{|t|}$ ab.

Bessere Eigenschaften weist das „**Raised Cosine Spectrum**“ auf. Es ist definiert als (mit $B = 2W - f_1$, $W = 1/2T_b \Rightarrow B = W(1 + \alpha)$)

$$P(f) = \begin{cases} \frac{1}{2W}, & |f| \leq f_1 \\ \frac{1}{4W} \cdot \left(1 - \sin\left(\frac{\pi(|f| - W)}{2\alpha W}\right)\right), & (1 - \alpha)W \leq |f| \leq (1 + \alpha)W \\ 0, & \text{sonst} \end{cases}$$

$$p(t) = \text{sinc}(2Wt) \cdot \frac{\cos(2\pi\alpha Wt)}{1 - 16\alpha^2 W^2 t^2}$$

mit



Für eine fixe **Datenrate R_b** haben wir **Bandbreite B** :

- Nyquist channel: $B = W = R_b/2$
- Raised cosine: $B = (1 + \alpha)W = (1 + \alpha)R_b/2$

3.3.1 Bandbreite

Im Basisband

Sender sendet mit Symboldauer $T_b \Rightarrow R_b = \frac{1}{T_b}$ im Basisband.

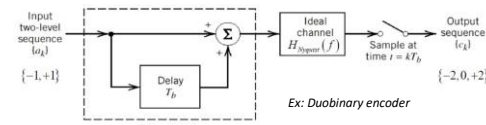
Nach der Abtastung im Empfänger landet die 1. Kopie des Spektrums bei $\frac{1}{T_b} = R_b$, d.h. das Spektrum eines Pulses darf maximal $\frac{R_b}{2} = W$ (einseitig) breit sein.

(Bezeichne ab hier mit W die einseitige Bandbreite, die ein Stream von Nyquistpulsen mit Rollof-Faktor $\alpha = 0$ hat, um eine Bitrate von R_b durchzubringen)

Das Raised Cosine Spectrum schafft einen Überlappungsbereich um $\pm\alpha W$ auf beiden Seiten von W , verbreitert also das Pulsspektrum auf $B = (1 + \alpha) \cdot W$, was dann auch physikalisch auf den Äther geht.

Trotz der Spektrumsverbreiterung wird das empfangene Signal aber weiter mit der Rate R_b abgetastet.

3.4 Correlative-Level Coding



$$c_k = a_k + a_{k-1}$$

$$H(f) = H_i(f)H_{nyq}(f) = 2T_b \cos(\pi f T_b) \exp(-j\pi f T_b) \mathbb{1}_{|f| \leq \frac{1}{2T_b}}$$

$$h(t) = \frac{T_b^2 \sin(\pi t / T_b)}{\pi t (T_b - t)}$$

Decision device kann korrekt codieren, falls c_k und a_{k-1} korrekt empfangen wurden.

Precoding gegen Fehlerfortpflanzung mittels modulo2 Addierer und Zeitverzögerung T_b wird die binäre Datensequenz $\{b_k\}$ zu $\{d_k\}$ codiert mittels:

$$d_k = b_k \oplus d_{k-1} \Rightarrow b_k = d_k \oplus d_{k-1} \quad (xor)$$

Damit: $c_k = \begin{cases} 0, & b_k = 1 \\ \pm 2, & b_k = 0 \end{cases}$ und das decision device muss vergangen

Signale nicht speichern (keine Fehlerfortpflanzung).

Beispiel für Correlative Coding mit Precoding

b_k	d_{k-1}	$d_k = d_{k-1} \oplus b_k$	a_{k-1}	$a_k = d_{k-1} \oplus f(d_k)$	c_k
init \rightarrow	0	init \rightarrow	-1		
0	0	0	-1	-1	-2
0	0	0	-1	-1	-2
1	0	1	-1	1	0
1	1	0	1	-1	0
1	0	1	-1	1	0
1	1	0	1	-1	0
0	1	1	1	1	2
0	1	1	1	1	2
0	1	1	1	1	2
1	1	0	1	-1	0
0	0	0	-1	-1	-2
0	0	0	-1	-1	-2
0	0	0	-1	-1	-2

$$f(x) = \text{map}(x, [0,1] \rightarrow [-1,1])$$

3.5 Baseband M-ary PAM (Pulse Amplitude Modulation)

• **Binary PAM:** Zwei Amplituden Levels

Transmit signal: $s(t) = \sum_k a_k \cdot g(t - k \cdot T_b)$

Receiver filter output: $y(t) = \mu \sum_k a_k \cdot p(t - k \cdot T_b) + n(t)$

Der Puls $p(t)$ ist normalisiert, so dass $p(0) = 1$ und

$$\mu p(t) = g(t) * h(t) * c(t) \Leftrightarrow \mu P(f) = G(f)H(f)C(f)$$

• **M-ary PAM:** M Amplituden Levels, somit codiert jedes Symbol $\log_2(M)$ bits (Bsp: jedes 4-ary PAM Symbol codiert $\log_2(4)=2$ bits).

Symboldauer T \rightarrow signaling rate $1/T$, $T = T_b \cdot \log_2(M)$, T_b : bit Dauer

4 Signal-Space Analysis



Annahmen:

- $p_i = p(m_i) = \frac{1}{M} \forall i$
- $s_i(t) \in \{s_k(t) | 1 \leq k \leq M\}$, real mit endlicher Energie
- AWGN: $x(t) = s_i(t) + w(t)$
- $w(t)$ ist ein Gaussprozess mit $\text{PSD} \frac{N_0}{2}$ und $\mu_w(t) = \mu_w = 0$
- damit ist $R_w(t_1, t_2) = R_w(t_2 - t_1) = R_w(\tau) = \frac{N_0}{2} \delta(\tau)$

Das Ziel ist es, die Fehlerwahrscheinlichkeit p_e zu minimieren:

$$p_e = \sum_{i=1}^M p_i \cdot P[\hat{m} \neq m_i | m_i]$$

Geometrische Darstellung von Signalen mit $\{\phi_j(t)\}_{j=1 \dots N}$ orthonormalem

Set an Basisfunktionen des Signalraums $\{s_i(t)\}_{i=1 \dots N}$

M Signale lassen mit N orthonormalen Basisfunktionen wie folgt darstellen:

$$s_i(t) = \sum_{j=1}^N s_{ij} \phi_j(t)$$

Wobei jedes s_{ij} der Ausgang des j-ten Korrelators ist:

$$s_{ij} = \int_0^T s_i(t) \phi_j(t) \cdot dt$$

Signalvektor Jedes Signal ist eindeutig bestimmt durch den Signalvektor

$$s_i(t) = [s_{i1} \dots s_{iN}]^T$$

- $\langle s_i(t), s_k(t) \rangle = \int_0^T s_i(t) s_k(t) dt = s_i^T s_k$
- $\int_0^T \phi_i(t) \phi_j(t) \cdot dt = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
- $E_i = \int_0^T s_i^2(t) dt = \|s_i\|^2 = \langle s_i(t), s_i(t) \rangle$
- distance: $\|s_i - s_k\| = \int_0^T (s_i(t) - s_k(t))^2 dt$

4.1 Gram-Schmidt Orthonormalisierung

Wie findet man für einen Satz an Signalfunktionen einen Satz von orthonormalen Basisfunktionen?

Entweder, indem man es einfach sieht, oder mit System:

- $\phi_1(t) = s_1(t) / \sqrt{\int_0^T s_1^2(t) dt}$
- $s_{21} = \int_0^T s_2(t) \phi_1(t) dt = \langle s_2(t), \phi_1(t) \rangle$
 $g_2(t) = s_2(t) - s_{21} \phi_1(t)$
 $\phi_2(t) = g_2(t) / \sqrt{\int_0^T g_2^2(t) dt}$ (if $g_2(t) \neq 0$)
- ...
- M. $g_M(t) = s_M(t) - s_{M1} \phi_1(t) - s_{M2} \phi_2(t) - \dots - s_{M,M-1} \phi_{M-1}(t)$
 $\phi_N(t) = g_M(t) / \sqrt{\int_0^T g_M^2(t) dt}$

4.1.1 How-To

ZUERST: Check auf lineare Abhängigkeit der Signale!

(Wenn aus linear abhängigen Signalen eine Basis konstruiert wird, hat die Matrix $[s_1 \ s_2 \ \dots \ s_M]$ keinen vollen Rang. (Wittneben))

Satz von trigonometrischen Funktionen gegeben

\Rightarrow Verschiedene Phasen und Frequenzen ergeben orthogonale Funktionen

Treppenfunktion über $[0, T]$ gegeben

\Rightarrow Setze für jeden Abschnitt eine neue Rechteckfunktion als Basisfunktion fest.

4.2 AWGN channel to vector channel

Mit $x(t) = s_i(t) + w(t)$ wie zuvor. Mittels einer Bank von Korrelatoren $j = 1 \dots N$.

Unter AWGN gilt für die Komponenten des Beobachtungsvektors \mathbf{x} :

$$x_j = \int_0^T x(t) \phi_j(t) \cdot dt = \int_0^T s_i(t) \phi_j(t) \cdot dt + \int_0^T w(t) \phi_j(t) \cdot dt$$

$$= s_{ij} + w_j$$

und damit lässt sich $x(t)$ wie folgt aufteilen:

$$x(t) = \sum_{j=1}^N x_j \phi_j(t) + w'(t) = \sum_{j=1}^N (s_{ij} + w_j) \phi_j(t) + w'(t)$$

wobei $w'(t) = w(t) - \sum_{j=1}^N w_j \phi_j(t)$ der Restterm des Rauschens ist (nicht im Signalraum), der für die Entscheidungsfindung irrelevant ist. Der Zufallsprozess w' hat die folgenden Eigenschaften:

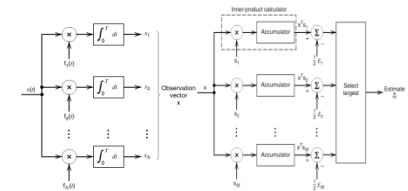
- $E[w'] = 0$
- statistisch unabhängig von allen $X_j \forall t$
- $E[w_i w_j] = \begin{cases} \frac{N_0}{2}, & i = j \\ 0, & \text{else} \end{cases}$

Wir können den relevanten Teil im Signalraum schreiben als:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \mathbf{s}_i + \mathbf{w} = \begin{bmatrix} s_{i1} \\ \vdots \\ s_{iN} \end{bmatrix} + \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}$$

Das empfangene Signal ist ein Sample des Zufallsprozess $X(t)$ und der Output jedes Korrelators ist eine Zufallsvariable X_j . Es gilt:

- $\mu_{x_j} = E[X_j] = E[s_{ij} + w_j] = s_{ij}$
- $\sigma_{x_j}^2 = E[w_j^2] = N_0/2$
- $\text{cov}(X_j, X_k) = 0$ (mutually uncorrelated)
- $w_j = X_j - s_{ij} = X_j - \mu_{x_j}$, $w_j = \int_0^T w(t) \phi_j(t) dt$



4.3 Detektion/ Fehlerwahrscheinlichkeit

Likelihood function Da die X_i gaussverteilt und mutually uncorrelated sind, haben wir lineare Unabhängigkeit und wir können schreiben:

$$L(s_i) = f_x(x|s_i) = f_w(W = x - s_i) = \prod_{j=1}^N f_w(w_j = x_j - s_{ij}) = \frac{1}{(\pi N_0)^{N/2}} \exp\left(-\frac{1}{N_0} \sum_{j=1}^N (x_j - s_{ij})^2\right)$$

$$l(s_i) = \ln(L(s_i)) = -\frac{1}{N_0} \sum_{j=1}^N (x_j - s_{ij})^2 + c$$

$$\Rightarrow c = -\frac{N}{2} \ln(\pi N_0), \text{ kann ignoriert werden für Entscheidung.}$$

Maximum a posteriori rule Fehlerwahrscheinlichkeit in der Entscheidungsregion $\hat{m}_i = m_i$ ist $P_e(m_i|x) = P(m_i \text{ not sent}|x)$. Minimal für

$$P(m_k \text{ sent}|x) = \frac{p_{k|x}(x|m_k)}{f_X(x)}, \text{ falls maximal für } k = i$$

Ermittelt durch Formel von Bayes.

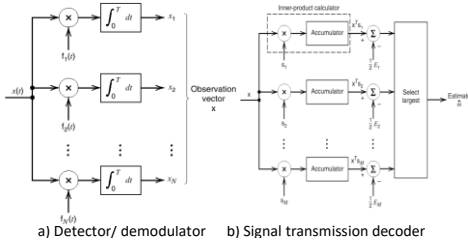
(p_k : a priori W'keit, dass m_k übertragen wird)

Falls $p_k = p \forall k$ gilt wir die Entscheidungsregel:

$$\text{setze } \hat{m} = m_i \text{ falls } l(m_k) \text{ maximal für } k = i$$

Simplified ML rule durch ausmultiplizieren von $l(s_i)$ und ignorieren von statistisch unabhängigen Termen erhält man:

$$\begin{aligned} & x \text{ liegt in Region } Z_i \text{ falls} \\ & \sum_{j=1}^N x_j s_{kj} - \frac{1}{2} E_k \text{ maximal für } k = i \\ & \Rightarrow E_k = \sum_{j=1}^N s_{kj}^2 \end{aligned}$$



Paarweise Fehlerwahrscheinlichkeit

$$P_2(s_1, s_2) = \frac{1}{2} \operatorname{erfc}\left(\frac{d_{ik}}{2\sqrt{N_0}}\right) = Q\left(\frac{d_{ik}}{\sqrt{2N_0}}\right)$$

$d_{ik} = \|s_2 - s_1\|$ die euklidische Distanz. Total ergibt sich

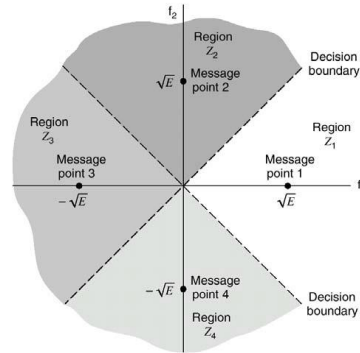
$$P_e = 1 - \frac{1}{M} \sum_{i=1}^M \int_{Z_i} f_x(x|m_i) dx$$

Die Formel kann vereinfacht werden und es ergibt sich der upper bound

$$P_e \leq \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M p_i \cdot \operatorname{erfc}\left(\frac{d_{ik}}{2\sqrt{N_0}}\right)$$

4.4 Detektion: Korrelationsempfänger

Der N-dimensionale Signalraum wird, ausgehend von der Positionierung der möglichen Signalvektoren, in M Entscheidungsregionen Z_i aufgeteilt.



Die Fehlerwahrscheinlichkeit ist gegeben durch

$$p_e = \sum_{i=1}^M p_i \cdot P[x \text{ liegt nicht in } Z_i | m_i \text{ gesendet}]$$

Die Berechnung dieser Integrale ist numerisch aufwendig, weshalb man sich damit behilft, einfach alle paarweisen Fehlerwahrscheinlichkeiten p_{ij} aufzusummieren.

$$p_e(m_i) \leq \sum_{j \neq i} p_{ij}$$

4.4.1 Äquivalenz mit Matched Filter

Die Impulsantwort eines MF ist bekanntlich $h_j(t) = k \cdot s_j(T - t)$.

Wenn der Amplitudenfaktor k so gewählt wird, dass $E_{h_j} = 1$, dann wird $h_j(t) = \phi_j(T - t)$.

Damit wird der Ausgang des MF

$$y_j(t) = \int_{-\infty}^{\infty} x(\tau) \phi_j(T - t + \tau) \cdot d\tau$$

Wird das Signal zum Zeitpunkt T abgetastet, findet man die Signaldefinition des j-ten Korrelators wieder, wenn man beachtet, dass $\phi_j(t) = 0$ ausserhalb dem Intervall $[0, T]$:

$$y_j(T) = \int_0^T x(\tau) \phi_j(\tau) \cdot d\tau = x_j$$

4.4.2 Union Bound

Sei $P_e(m_i)$ die Fehlerw'keit gegeben, dass m_i (Signalvektor s_i) gesendet wurde. Sei A_{1k} das Ereignis, dass der receive vector x näher beim Signalvektor s_k als beim Signalvektor s_i liegt (hat W'keit $P(A_{1k})$) Dann:

$$P_e(m_i) \leq \sum_{k=1}^M \sum_{k \neq i} P(A_{ik}), i \in \{1, \dots, M\}$$

5 Basisband-Repräsentation

5.1 Hilbert-Transformation

Auszug aus Haykin p.723

Definitionen:

$$\hat{g}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{g(\tau)}{t - \tau} \cdot d\tau$$

$$g(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\hat{g}(\tau)}{t - \tau} \cdot d\tau$$

Die Fouriertransformierte von $\hat{g}(t)$ ist

$$\hat{G}(f) = -j \operatorname{sgn}(f) G(f)$$

Eigenschaften:

- Ein Signal $g(t)$ und dessen Hilberttransformierte $\hat{g}(t)$ besitzen das selbe Magnitudenspektrum.
- Wenn $\hat{g}(t)$ die Hilberttransformierte von $g(t)$ ist, dann ist die Hilberttransformierte von $\hat{g}(t)$ wiederum $-g(t)$.
- Ein Signal $g(t)$ und dessen Hilberttransformierte $\hat{g}(t)$ sind über das gesamte Intervall $(-\infty, \infty)$ orthogonal, wie $\int_{-\infty}^{\infty} g(t) \hat{g}(t) \cdot dt = 0$ zeigt.

5.2 Basisband-Repräsentation

5.2.1 Pre-Envelope / Analytical signal $g_+(t)$

für positive Frequenzen

$$g_+(t) = g(t) + j\hat{g}(t) \quad I)$$

$$\mathcal{F}\{\cdot\} \Leftrightarrow \mathcal{F}^{-1}\{\cdot\}$$

$$G_+(f) = G(f) + \operatorname{sgn}(f) G(f)$$

und damit

$$G_+(f) = \begin{cases} 2G(f), & f > 0 \\ G(0), & f = 0 \\ 0, & f < 0 \end{cases} \quad II)$$

Berechne das analytische Signal auf zwei Arten:

- Berechne die Hilberttransformierte $\hat{g}(t)$ von $g(t)$ und verwende I).
- Bestimme die Fouriertransformierte $G(f)$ von $g(t)$ und benütze II).

Für negative Frequenzen

$$g_-(t) = g(t) - j\hat{g}(t)$$

Es ist offensichtlich, dass das analytische Signal für negative Frequenzen einfach das komplex konjugierte analytische Signal für positive Frequenzen ist. Formal

$$g_-(t) = g_+^*(t)$$

Und damit

$$G_-(f) = \begin{cases} 0, & f > 0 \\ G^*(0), & f = 0 \\ 2G^*(f), & f < 0 \end{cases} \quad III)$$

5.2.2 Complex envelope $\tilde{g}(t)$

Nehme das analytische Signal ins Basisband

$$\tilde{g}(t) = \frac{1}{\sqrt{2}} g_+(t) e^{-j2\pi f_0 t}$$

$$\tilde{G}(f) = \frac{1}{\sqrt{2}} G_+(f + f_0)$$

$\tilde{g}(t)$ wird als „complex envelope“ bezeichnet.

Oder mit der Identität:

$$\begin{aligned} x(t) &= \operatorname{Re}\{\sqrt{2}\tilde{x}(t)e^{-j2\pi f_0 t}\} = \operatorname{Re}\{\sqrt{2}(\tilde{x}_I(t) + j\tilde{x}_Q(t))e^{-j2\pi f_0 t}\} = \\ &= \sqrt{2}\tilde{x}_I(t) \cos(2\pi f_0 t) - \sqrt{2}\tilde{x}_Q(t) \sin(2\pi f_0 t) \end{aligned}$$

Originalsignal als Funktion von In-Phase- und Quadratur-Teil

$$g(t) = \operatorname{Re}\{\sqrt{2} \cdot \tilde{g}(t) \exp(j2\pi f_0 t)\}$$

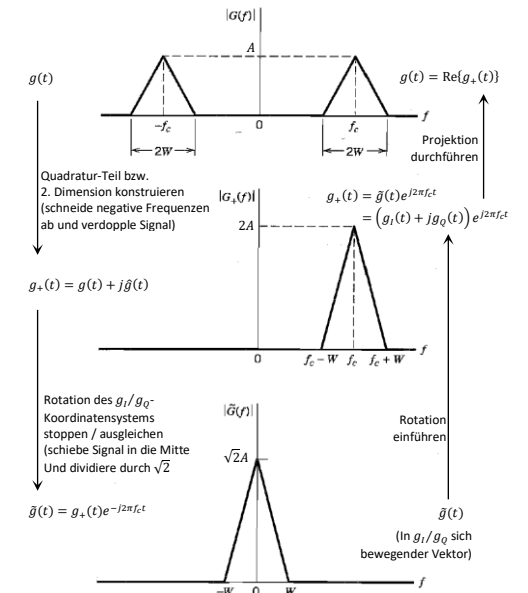
mit

$$\tilde{g}(t) = g_I(t) + jg_Q(t)$$

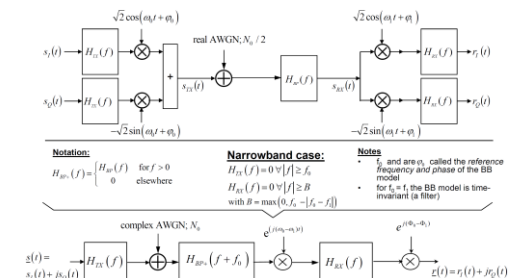
Damit

$$g(t) = g_I(t) \cos(2\pi f_0 t) - g_Q(t) \sin(2\pi f_0 t)$$

Spektrum des äquivalenten Bandpasssignals:



Die Aufteilung des Bandpass-Signals $g(t)$ in einen In-Phase und Quadraturteil erlaubt es, die PSD von solchen Modulationsarten einfach zu berechnen. Man kann jeden Signalanteil separat für sich betrachten und dabei die PSD von sinusoidalen Signalen berechnen, die durch einen stationären Zufallsprozess moduliert werden.



5.2.3 PSD eines amplitudenmodulierten sinusoidalen Prozesses

Sei $X(t)$ ein stationärer Zufallsprozess und

$$Y(t) = X(t) \cdot \cos(2\pi f_c t + \Theta)$$

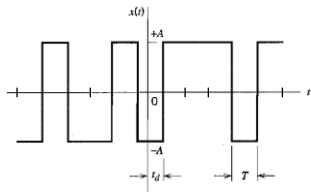
Θ ist eine Zufallsvariable, welche ausdrücken soll, dass $X(t)$ und der sinusoidale Anteil keinen definierten Phasenzusammenhang haben.

$X(t)$ könnte z.B. eine zufällige binäre Folge sein.

$$R_Y(\tau) = \frac{1}{2} R_X(\tau) \cos(2\pi f_c \tau)$$

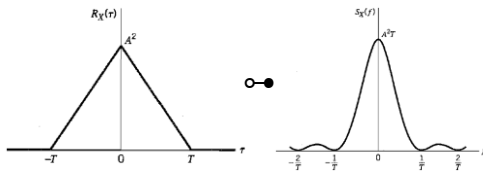
$$S_Y(f) = \frac{1}{4} [S_X(f - f_c) + S_X(f + f_c)]$$

5.2.4 PSD einer zufälligen binären Folge (Welle)



Eigenschaften:

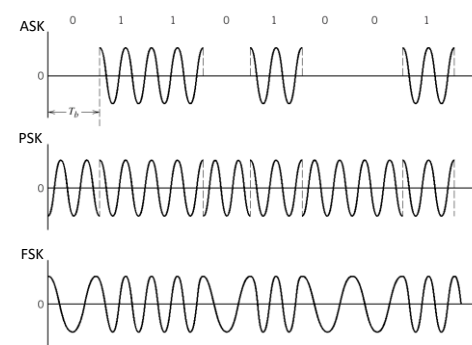
- Die Amplitudenwerte für die Bits 1 und 0 nehmen $\pm A$ an.
- Der Start des ersten Bits findet zufällig innerhalb einer Abtastperiode statt. D.h. t_d ist gleichverteilt im Intervall $[0, T]$.
- Jedes Bit wird statistisch unabhängig vom vorigen, mit gleicher Wahrscheinlichkeit 0 oder 1, gezogen.



$$R_X(\tau) = \begin{cases} A^2 \left(1 - \frac{|\tau|}{T}\right), & |\tau| < T \\ 0, & \text{sonst} \end{cases} \quad \leftrightarrow \quad S_X(f) = A^2 T \operatorname{sinc}^2(fT)$$

6 Kohärente Bandpass-Übertragung

Grundlegende Modulationsverfahren:



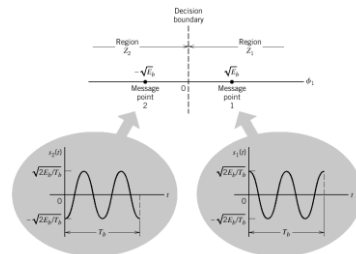
6.1 BPSK

Binary Phase Shift Keying

$$\phi_1(t) = \sqrt{\frac{2E_b}{T_b}} \cdot \cos[2\pi f_c t], \quad 0 \leq t \leq T_b$$

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t) = +\sqrt{E_b} \phi_1(t)$$

$$s_2(t) = \sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}} \cdot \cos(2\pi f_c t) = -\sqrt{E_b} \phi_1(t)$$



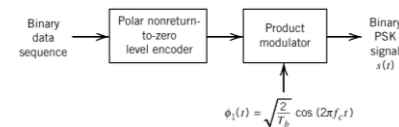
$$s_{11} = \int_0^{T_b} s_1(t) \phi_1(t) \cdot dt = +\sqrt{E_b}$$

$$s_{21} = \int_0^{T_b} s_2(t) \phi_1(t) \cdot dt = -\sqrt{E_b}$$

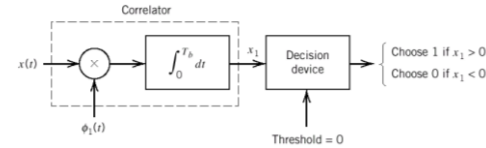
$$\Rightarrow \text{Distanz im Signalraum: } d = 2\sqrt{E_b}$$

Der Detektor beobachtet $\int_0^T x(t) \phi_1(t) dt$ und Entscheidet mit threshold = 0 (für gleich wahrscheinliche Symbole)

BPSK Transmitter:



Kohärenter BPSK Empfänger:



Fehlerwahrscheinlichkeiten:

$$\begin{aligned} P_e &= p_{0 \rightarrow 1} + p_{1 \rightarrow 0} \\ &= \frac{1}{2} \cdot \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) + \frac{1}{2} \cdot \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \\ &= \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = \frac{1}{2} \operatorname{erfc}\left(\frac{d}{2\sqrt{N_0}}\right) \end{aligned}$$

Spektrale Leistungsdichte (PSD):

Die spektrale Leistungsdichte im Basisband ist

$$S_B(f) = 2E_b \operatorname{sinc}^2(T_b f)$$

Benütze Fakt, dass der in-Phase-Teil von $\tilde{g}(t)$ mit einer zufälligen Binärfolge mit Amplitude $\sqrt{\frac{E}{T_b}}$ ausgedrückt werden kann.

6.2 QPSK

Quadrature Phase Shift Keying

$$s_i(t) = \begin{cases} \sqrt{\frac{2E}{T}} \cdot \cos[2\pi f_c t + (2i-1) \cdot \frac{\pi}{4}], & 0 \leq t \leq T; 1 \leq i \leq 4 \\ 0, & \text{sonst} \end{cases}$$

Dies lässt sich zu

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[(2i-1) \cdot \frac{\pi}{4}\right] \cos[2\pi f_c t] - \sqrt{\frac{2E}{T}} \sin\left[(2i-1) \cdot \frac{\pi}{4}\right] \sin[2\pi f_c t] \quad 0 \leq t \leq T$$

umschreiben.

Damit sind die beiden Basisfunktionen durch

$$\phi_1(t) = \sqrt{\frac{2E}{T}} \cos[2\pi f_c t] \quad 0 \leq t \leq T, \quad \phi_2(t) = \sqrt{\frac{2E}{T}} \sin[2\pi f_c t] \quad 0 \leq t \leq T$$

gegeben.

Also ergeben sich folgende Signalpunkte mittels Gray-Code:

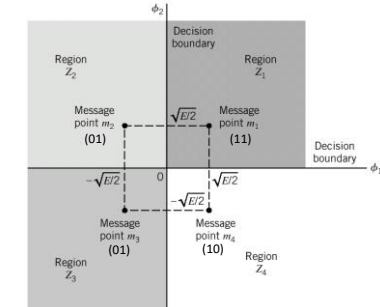
Gray-Code	Phase	S_{i1}	S_{i2}
10	$\pi/4$	$+\sqrt{E/2}$	$-\sqrt{E/2}$
00	$3\pi/4$	$-\sqrt{E/2}$	$-\sqrt{E/2}$
11	$5\pi/4$	$-\sqrt{E/2}$	$+\sqrt{E/2}$
01	$7\pi/4$	$+\sqrt{E/2}$	$+\sqrt{E/2}$

Jedes QPSK Symbol trägt 2 Informationsbits somit ist die Symbolenergie doppelt so gross wie die Energie pro Informationsbit:

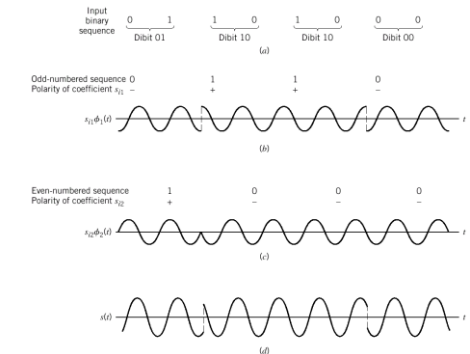
$$E = 2E_b$$

- $BER = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$, die gleiche Error Rate wie BPSK aber doppelte Bitrate
- $S_B(f) = 4E_b \operatorname{sinc}^2(2T_b f) \rightarrow$ QPSK braucht nur die halbe Bandbreite von BPSK.

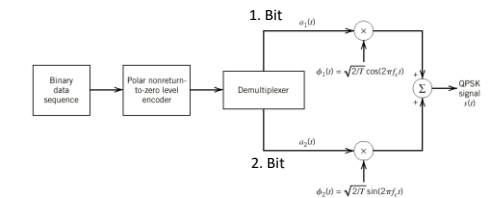
Das Phasendiagramm schaut wie folgt aus:



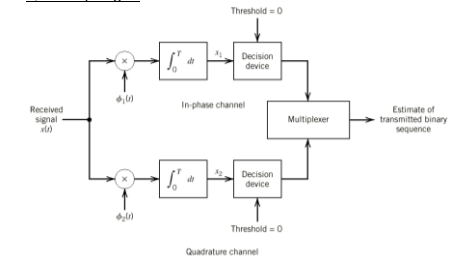
Für ein Beispiel:



QPSK Sender:

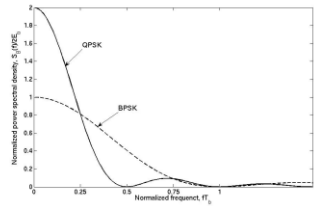


QPSK Empfänger:



Spektrale Leistungsdichte im Basisband

$$S_B(f) = 4E_b \operatorname{sinc}^2(2T_b f)$$



6.3 M-ary QAM (gerade # Bit)

Quadrature-Amplitude Shift Keying

In der Vorlesung wird nur der Fall behandelt, wenn die Anzahl Signalvektoren eine Zweierpotenz mit geradem Exponent ist.

Bsp: 4 bits können auf $2^4 = 16$ Symbole gemapped werden für ein 4-ary QAM in einem Grid angeordnet und detektiert mit L-PAMs ($L = \sqrt{M}$). Jedes 16-QAM Symbol überträgt 4 Bit.

M-ary QAM: $M = 2^n$
 \rightarrow Symboldauer: $T = nT_b$
 \rightarrow Bandbreite $B = \frac{1}{nT_b}$

4-QAM ist äquivalent zu QPSK.

Die Basisfunktionen sind wie bei QPSK

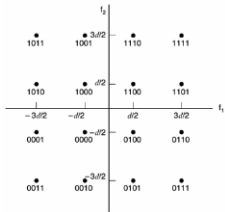
$$\phi_1(t) = \sqrt{\frac{2}{T}} \cos[2\pi f_c t] \mathbb{1}_{0 \leq t \leq T}, \quad \phi_2(t) = \sqrt{\frac{2}{T}} \sin[2\pi f_c t] \mathbb{1}_{0 \leq t \leq T}$$

Die Signalvektoren sind durch

$$s_i = \frac{d_{\min}}{2} \begin{bmatrix} a_i \\ b_i \end{bmatrix}, \quad a_i, b_i \text{ ungerade, } \in \mathbb{Z}^{>0}, \quad i = 1, \dots, M$$

gegeben. Die Zuordnung der Bits erfolgt dann wieder nach einem Gray-Schema.

Bspw. in der Abbildung die ersten zwei Bit für den Quadranten, die zweiten zwei Bit für die Position innerhalb des Quadranten.



Fehlerberechnung

(äußere Symbole einer Reihe haben halbe Fehlerwahrscheinlichkeit wie innere Symbole)

$$P_{\text{korrekt}} = (1 - P_e')^2 \Rightarrow P_e = 1 - P_{\text{korrekt}} = 1 - (1 - P_e')^2 \approx 2P_e'$$

$$\text{mit } P_e' = \left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc}\left(\sqrt{\frac{d_{\min}^2}{4N_0}}\right)$$

oder

$$P_e \approx 2 \left(1 - \frac{1}{\sqrt{M}}\right) \operatorname{erfc}\left(\sqrt{\frac{3E_{av}}{2(M-1)N_0}}\right)$$

mit durchschnittlicher Symbolenergie: $E_{av} = \frac{(M-1)d_{\min}^2}{6}$

Die Bitfehlerwahrscheinlichkeit bleibt im 2-dimensionalen gleich, weil bei der Verdoppelung der Übertragungsrate sich auch die Fehlerrate verdoppelt. Jedoch nehmen die Symbolfehler zu.

6.4 FSK

Frequency Shift Keying

$$s_i(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_i t], & 0 \leq t \leq T_b \\ 0 & \text{sonst} \end{cases}$$

Um Phasendiskontinuitäten zu vermeiden sind die Frequenzen wie folgt gewählt:

$$f_i = \frac{n_c + i}{T_b}, \quad i \in \{1, 2\}, \quad n_c \in \mathbb{Z}_{\geq 0}$$

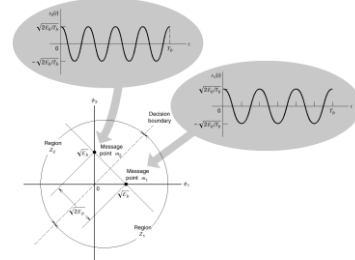
Damit ergeben sich die Basisfunktionen

$$\phi_1(t) = \sqrt{\frac{2}{T_b}} \cos[2\pi f_1 t] \mathbb{1}_{0 \leq t \leq T}, \quad \phi_2(t) = \sqrt{\frac{2}{T_b}} \cos[2\pi f_2 t] \mathbb{1}_{0 \leq t \leq T}$$

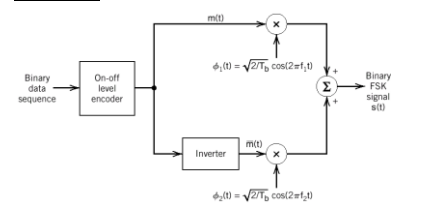
mit den Signalvektoren

$$s_1 = \begin{bmatrix} \sqrt{E_b} \\ 0 \end{bmatrix}, \quad s_2 = \begin{bmatrix} 0 \\ \sqrt{E_b} \end{bmatrix}$$

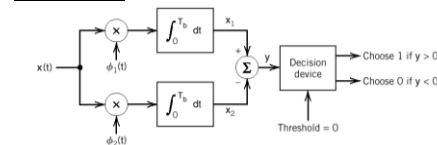
Damit ist die Distanz im Signalraum: $d_{\min} = \sqrt{2E_b}$



BFSK Sender:



BFSK Empfänger:



Fehlerrate:

Im Gegensatz zu BPSK nehmen wir eine 3dB-Abnahme des SNR in Kauf, weil der euklidische Abstand zwischen den beiden Signalpunkten bei gleicher Aussteuerung der Basisfunktion kleiner ist.

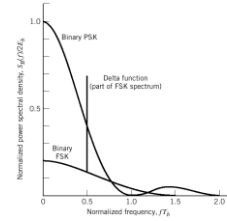
Es berechnet sich leicht der Abstand zwischen den Punkten $d_{\min} = \sqrt{2E_b}$.

$$P_e = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{d_{\min}^2}{4N_0}}\right) = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

Spektrale Leistungsdichte im Basisband

$$S_B(f) = \frac{E_b}{2T_b} \left[\delta\left(f - \frac{1}{2T_b}\right) + \delta\left(f + \frac{1}{2T_b}\right) \right] + \frac{8E_b \cos^2[\pi T_b f]}{\pi^2 (4T_b^2 f^2 - 1)^2}$$

\rightarrow decays with $\sim f^{-4}$ (PSK f^{-2})



Die PSD von BFSK enthält zwei Deltapulse bei f_1 und f_2 .

Diese Anteile des konstanten Trägers enthalten keine Information, können jedoch zur Synchronisation des Empfängers benutzt werden.

Das Spektrum von BFSK fällt viel schneller ab ($\propto \frac{1}{f^4}$) als jenes von

BPSK ($\propto \frac{1}{f^2}$). Jedoch ist das Hauptband durch die Trägerdeltas weiter.

6.5 CPFSK

Continuous Phase Frequency Shift Keying

Bei CPFSK steht im Vordergrund, dass die Wellenform keine Diskontinuitäten, also keine Phasensprünge enthält. Ein allgemeines CPFSK-Signal lässt sich wie folgt beschreiben:

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \theta(t)]$$

Der Phasenterm ist stetig und von der Form

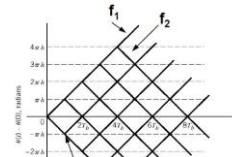
$$\theta(t) = \theta(0) \pm \frac{\pi h}{T_b} t, \quad 0 \leq t \leq T_b$$

Durch Umformung erkennt man, dass dieses Verfahren einer FSK-Modulation entspricht, die die Beschränkung aufgehoben hat, dass eine ganze Anzahl an Schwingungen der beiden Frequenzen f_1 und f_2 in die Symboldauer passt. Die Stetigkeit wird durch den Term $\theta(0)$ garantiert, welcher die kumulative Phase, die sich bis zum Zeitpunkt $t = 0$ angehäuft hat, repräsentiert.

Da $s_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_i t + \theta(0))$ (von BFSK) und mit einem Parametervergleich mit der BFSK lassen sich h und f_c wie folgt bestimmen:

$$h = T_b(f_1 - f_2), \quad f_c = \frac{1}{2}(f_1 + f_2)$$

Die Phase des Trägers ändert sich also linear über eine Symboldauer, was anders ausgedrückt auch einfach einer geänderten Frequenz entspricht. $h = 1/2 \rightarrow$ minimale Differenz für welche $s_1(t)$ und $s_2(t)$ orthogonal sind. Dies entspricht **MSK**.



Jede Spur (trace) gehört zu einer Bitsequenz, keine Phasensprünge. Für obiges Beispiel: $h = 1 \rightarrow f_1 - f_2 = \frac{1}{T_b}$

Bei $t_k = k \cdot T_b$:

- k gerade: $\theta(t_k) \bmod 2\pi = 0$
- k ungerade: $\theta(t_k) \bmod 2\pi = \pi$

Remember: $3dB \cong \frac{1}{\sqrt{2}}$

6.5.1 MSK

Minimum Shift Keying

Bei MSK wird $h = \frac{1}{2}$ gewählt (so sind die zwei Symbole noch orthogonal), sodass sich die Phase während einer Symboldauer jeweils um $\pm \frac{\pi}{4}$ ändert. Zudem ist wegen $h = \frac{1}{2} \Rightarrow f_1 - f_2 = \frac{1}{2T_b}$ (Differenz ist die halbe Bitrate). Für MSK: $WT_b = \infty$ und $\alpha = 2$ (siehe GMSK)

Setzt man den Term für $\theta(t)$ in der Symboldefinition ein und benutzt das Additionstheorem $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$, so ergibt sich

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos[\theta(t)] \cos[2\pi f_c t] - \sqrt{\frac{2E_b}{T_b}} \sin[\theta(t)] \sin[2\pi f_c t]$$

- Die Phase $\theta(0)$ bestimmt die cos-Komponente für 2 Bitdauern. Somit kann man $\theta(0)$ schätzen indem wir den ersten Teil beobachten
- In gleicher Weise kann man $\theta(T_b)$ schätzen, indem man den zweiten Teil beobachtet

Umformen des In-Phase-Terms ergibt

$$s_i(t) = \sqrt{\frac{2E_b}{T_b}} \cos[\theta(0)] \cos\left[\frac{\pi}{2T_b} t\right]$$

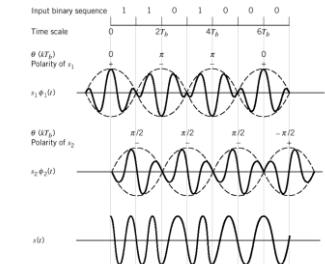
Man stellt nun fest, dass $s_i(t=0) = \sqrt{\frac{2E_b}{T_b}} \cos[\theta(0)]$ und damit Stetigkeit von $s_i(t)$ nur dann gegeben ist, wenn $\theta(0)$ sowohl für das vorangegangene als auch das aktuelle Symbol gilt. Damit können wir also $s_i(t)$ auf dem Intervall $-T_b \leq t \leq T_b$ betrachten und erhalten

$$s_i(t) = \pm \sqrt{\frac{2E_b}{T_b}} \cos\left[\frac{\pi}{2T_b} t\right]$$

in Abhängigkeit von $\theta(0)$, der Anfangsphase des aktuellen Symbols (entweder 0 oder π).

Einer ähnlichen Argumentation folgend muss der Quadraturterm auf dem Intervall $0 \leq t \leq 2T_b$ mit gleichem $\theta(T_b)$ gelten.

$\cos\left[\frac{\pi}{2T_b} t\right]$ bzw. $\sin\left[\frac{\pi}{2T_b} t\right]$ sind $4T_b$ -periodisch. Sie dienen letztlich als Hüllkurve für den Träger, jeweils $2T_b$ lange und um $1T_b$ zeitlich versetzt. In den Nulldurchgängen der Hüllkurven kann das Vorzeichen dann gewechselt werden.



Man erhält einen 2D Signalraum für die Schätzungen $\theta(0), \theta(T_b)$:

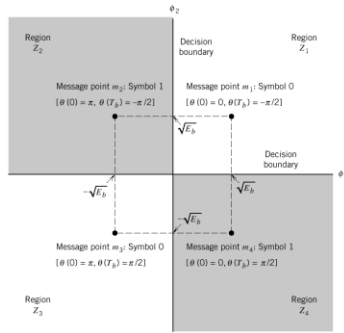
$$\phi_1(t) = \sqrt{\frac{2}{T_b}} \cos\left[\frac{\pi}{2T_b} t\right] \cos[2\pi f_c t], \quad -T_b \leq t \leq T_b$$

$$\phi_2(t) = \sqrt{\frac{2}{T_b}} \sin\left[\frac{\pi}{2T_b} t\right] \sin[2\pi f_c t], \quad 0 \leq t \leq 2T_b$$

die zugehörigen empfangenen Signalkoeffizienten

$$x_1 = \int_{-T_b}^{T_b} (s(t) + w(t)) \phi_1(t) \cdot dt = \sqrt{E_b} \cos[\theta(0)] + w_1$$

$$x_2 = \int_0^{2T_b} (s(t) + w(t)) \phi_2(t) \cdot dt = -\sqrt{E_b} \sin[\theta(T_b)] + w_2$$

**Fehlerrate:**

Jedes gesendete Symbol kann entweder $\theta(0) = 0$ oder $\theta(0) = \pi$ haben, das heisst pro Symbol muss beachtet werden, dass Symbol 0 sowohl durch m_1 als auch m_3 repräsentiert werden könnte. Analoges gilt auch für Symbol 1.

Die Fehlerwahrscheinlichkeit mit differential precoding berechnet sich schliesslich zu

$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$$

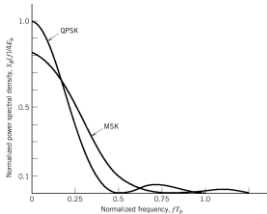
was der gleichen Rate wie bei PSK und QPSK entspricht. Das Verfahren geniesst diese gute Performance, weil jeweils über zwei Bitdauern pro Symbol integriert wird.

Ohne differential precoding wäre die Fehlerwahrscheinlichkeit:

$$P_e = \operatorname{erfc}(\sqrt{E_b/N_0}) - \frac{1}{2} \operatorname{erfc}(\sqrt{E_b/N_0})^2$$

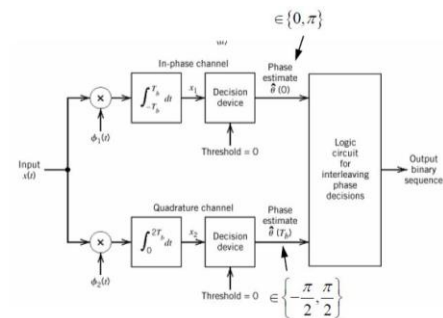
Spektrale Leistungsdichte im Basisband (p.395)

$$S_B(f) = \frac{32E_b}{\pi^2} \left[\frac{\cos(2\pi f T_b)}{16T_b^2 f^2 - 1} \right]^2$$



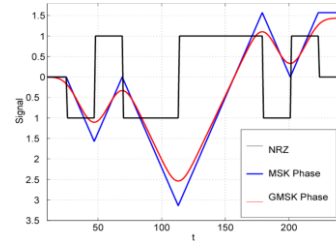
Das Spektrum fällt im Vergleich zu QPSK ($\propto \frac{1}{f^2}$) mit der vierten Potenz ab, was insgesamt zu weniger starken Seitenbändern führt, was eine sehr willkommene Eigenschaft von MSK ausmacht.

Blockdiagramm eines MSK receiver:

**6.5.2 GMSK****Gaussian Filtered Minimum Shift Keying**

In der Regel wird das **MSK-Signal** generiert, indem einem Spannungsgesteuerter Oszillator (VCO) ein **bipolares NRZ-Signal** gefüttert wird. Es wird also hart zwischen zwei Frequenzen umgeschaltet. Dies führt dazu, dass das Spektrum an den Rändern nur sehr langsam abfällt.

Eine Strategie, die besser in den Griff zu kriegen, ist das **Filtern des NRZ-Signals mit einem Gauss'schen Tiefpass**. Ein Nachteil ist, dass GMSK nicht mehr ISI-frei ist. Es muss deshalb eine sorgfältige Abwägung zwischen den Bandbreitenanforderungen und ISI gemacht werden.



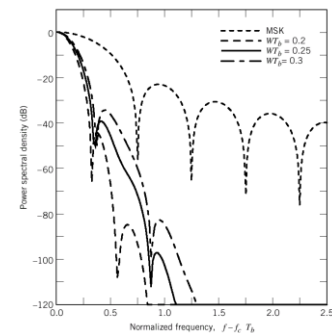
Pulse-shaping filter:

$$H(f) = \exp \left(-\frac{\log(2)}{2} \left(\frac{f}{W} \right)^2 \right)$$

nach Faltung von $h(t)$ mit dem NRZ-Signal:

$$g(t) = \frac{1}{2} \operatorname{erfc} \left(\pi \sqrt{\frac{2}{\log(2)}} W T_b \left(\frac{t}{T_b} - \frac{1}{2} \right) \right) - \operatorname{erfc} \left(\pi \sqrt{\frac{2}{\log(2)}} W T_b \left(\frac{t}{T_b} + \frac{1}{2} \right) \right)$$

Das ergibt die folgenden Spektren für verschiedene Filterbreiten $W T_b$.



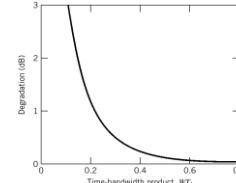
Beim Mobilfunkstandard GSM wird $W T_b = 0.3$ verwendet. Das Modulationsverfahren ist in der Lage 99% der Sendeleistung in einer Bandbreite von 250 kHz zu konzentrieren.

Fehlerwahrscheinlichkeit

Bei GMSK gibt es eine empirische Formel zur Berechnung der Bitfehler:

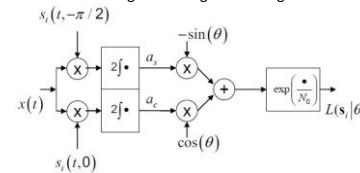
$$P_e = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{\alpha E_b}{2N_0}} \right)$$

mit dem Parameter α , den wir aus der folgenden Grafik mit Hilfe der Beziehung $\operatorname{degr}[\text{dB}] = 10 \cdot \log_{10} \left(\frac{\alpha}{2} \right) \Leftrightarrow \alpha = 10^{\frac{\operatorname{degr}[\text{dB}]}{10}}$ errechnen können. Dabei bezeichnet die Grösse «degr» den Performanceverlust im Vergleich zu herkömmlichem MSK.

**7 Inkohärente Bandpass-Übertr.****7.1 Noncoherent FSK**

$$s_i(t, \theta) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_i t + \theta)$$

Wobei f_1, \dots, f_n orthogonal sein müssen. Die Likelihoodfunktion für eine Bank von Korrelatoren mit gleich-energetischen Signalen



Für gleichverteilten phase offset.

Likelihood function: (I_0 : modifizierte Besselfunktion, order 0)

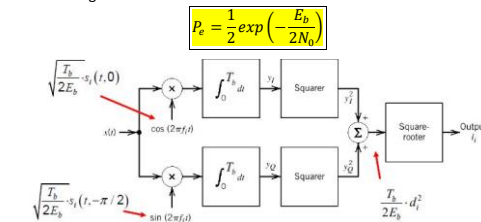
$$\begin{aligned} \bar{L}(s_i) &\equiv E_\theta \{ L(s_i|\theta) \} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp \left(\frac{1}{N_0} \sqrt{a_z^2 + a_z^2} \cdot \cos(\theta + \varphi) \right) d\theta = \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp \left(\frac{1}{N_0} \sqrt{a_z^2 + a_z^2} \cdot \cos(\theta) \right) d\theta = I_0 \left(\frac{1}{N_0} \sqrt{a_z^2 + a_z^2} \right) \end{aligned}$$

Wir wählen Signale für welche die Likelihoodfunktion maximal ist. Da die Besselfunktion I_0 monoton steigend ist, können wir vereinfachen zu:

$$\hat{i} = \operatorname{argmax}_i (a_z^2 + a_z^2)$$

Anstatt Integratoren kann man auch Matched Filters brauchen und da $s_i(t, -\frac{\pi}{2})$, $s_i(t, 0)$ Hilbert-Transformations-Paare sind, können wir das Signal $s_i(t, 0)$ und ein Hilbert Transforme benutzen.

Für $i = 1, 2$ jeder Branch braucht so einen Detektor und die Outputs werden verglichen



Korrelations Decodierer für noncoherent FSK

Wir brauchen einen Korrelator mit $\theta = 0$ (inphase) und einen mit $\theta = \frac{\pi}{2}$ (quadrature). $i = 1, 2$ brauchen beide je solche Branches.

7.2 DPSK

Differential Phase Shift Keying, binary:

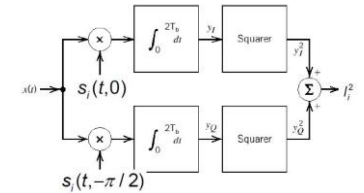
$$s_1(t, \theta) = \begin{cases} \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + \theta), & 0 \leq t \leq T \\ \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + \theta), & T \leq t \leq 2T \end{cases}$$

$$s_2(t, \theta) = \begin{cases} \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + \theta), & 0 \leq t \leq T \\ \sqrt{\frac{2E}{T}} \cos(2\pi f_c t + \pi + \theta), & T \leq t \leq 2T \end{cases}$$

Mit unbekannter Phase.

- Symbol 0: Phasenverschiebung vom 180°
- Symbol 1: keine Phasenverschiebung

Der Receiver speichert die fortschreitende Phase in einem Buffer und wir nehmen an, dass die Phase zwischen zwei Sampling Instanzen nicht erheblich ändert.



Der Receiver braucht im binären Fall 2 solche Branches, für jede Basisfunktion ein Branch.

Dies ist der grundsätzliche Aufbau eines decision device für 2 orthogonale Signale (hier: s_1, s_2) mit unbekannter Phase. Da

$$\int_0^{2T_b} s_1(t) s_2(t) dt = 0$$

Haben wir orthogonale Signale und somit können wir es verwenden.

Precoding Strategie

- Wähle ein zufälliges erstes Bit
- Falls das einkommende binäre Symbol b_k :
 - $b_k = 1$: $d_k = d_{k-1}$
 - $b_k = 0$: $d_k = (d_{k-1} + 1) \bmod 2$

Dies entspricht: $d_k = d_{k-1} \oplus b_k$

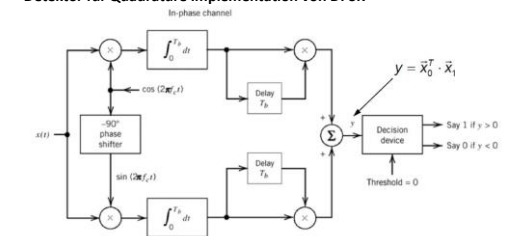
Dann:

- $d_k = 0 \rightarrow \text{phase } \pi$
- $d_k = 1 \rightarrow \text{phase } 0$

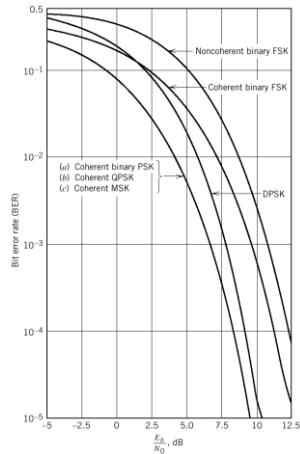
DPSK ist ein Spezialfall von noncoherent orthogonal modulation mit $T = 2T_b, E = 2E_b$

Fehlerrate

$$P_e = \frac{1}{2} \exp \left(-\frac{E}{2N_0} \right) = \frac{1}{2} \exp \left(-\frac{E_b}{N_0} \right)$$

Detektor für Quadrature Implementation von DPSK

8 Vergleich von Übertragungsverfahren



Übertragungsverfahren	Fehlerrate
kohärentes BPSK, QPSK, MSK	$\frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$
kohärentes BFSK	$\frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{2N_0}}\right)$
DPSK	$\frac{1}{2} \exp\left(-\frac{E_b}{N_0}\right)$
nichtkohärentes FSK	$\frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right)$

tieferer SNR
höhere BER

Die Verfahren haben alle ihre Vorteile:

- QPSK hat eine doppelte Bitrate wie BPSK und MSK.
- Die spektrale Effizienz von MSK ist höher wie bei QPSK und BPSK.
- BPSK ist sehr robust.

9 Multiuser-Übertragung

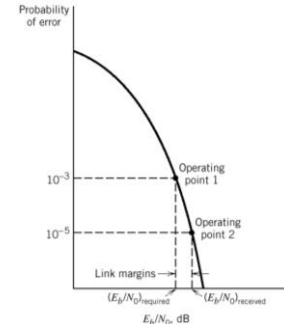
Leistungsberechnung:

$$P[\text{dBm}] = 10 \cdot \log_{10}\left(\frac{P[\text{W}]}{10^{-3} \text{ W}}\right) \Leftrightarrow P[\text{mW}] = 10^{\frac{P[\text{dBm}]}{10}}$$

Trennung der User mittels

1. **FDMA**: frequency division
Teile das verfügbare Spektrum auf (Guard Bands einplanen!).
FDMA führt in der Summe zu AWGN, welches hohe Peaks führen kann. Nichtlinearitäten in Transpondern können dazu führen, dass trotz Orthogonalität der Signale Übersprecher vorkommen können.
⇒ Empfänger nicht ganz aussteuern (ineffizient!)
2. **TDMA**: time division
Serieller Zugriff auf das Medium (Guard Times!)
Man kann bandbegrenzte analoge Signale kritisch abtasten, puffern und dann in gewissen Zeitintervallen mit höherer Bitrate übertragen. Die restliche Zeit, die frei wird, kann für weitere Nutzer verwendet werden.
Abbildung
3. **CDMA**: code division
Z.B. Frequency Hopping
4. **SDMA**: spatial division
Richtantennen, Antennenarrays

9.1 Link Margin



OP1: höchste erlaubte Fehlerrate, minimal benötigter SNR

OP2: effektiv gefahrene Fehlerrate, aktueller SNR

Link Margin: Differenz zwischen benötigtem und empfangenen Signallevel (Fehlerwahrscheinlichkeit)

$$M(\text{dB}) = \left(\frac{E_b}{N_0}\right)_{\text{rec}} (\text{dB}) - \left(\frac{E_b}{N_0}\right)_{\text{req}} (\text{dB})$$

$\left(\frac{E_b}{N_0}\right)_{\text{req}} = \left(\frac{C/R}{N_0}\right)$. Da die received carrier power C als Energie pro Bit E_b mal die Datenrate R ausgedrückt werden kann.

Zudem:

$$\left(\frac{E_b}{N_0}\right)_{\text{req}} = \left(\frac{P_r}{N_0}\right)_{\text{req}} \cdot \frac{1}{R}$$

Mit Link Margin M :

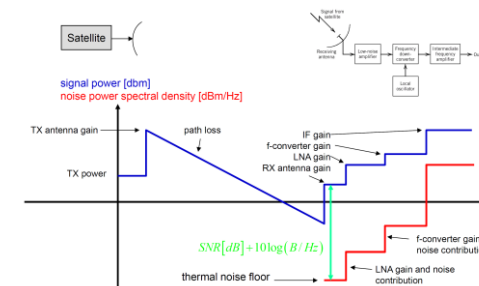
$$\left(\frac{P_r}{N_0}\right)_{\text{DL}} = \left(\frac{E_b}{N_0}\right)_{\text{req}} \cdot R \cdot M = EIRP \cdot \frac{G_r}{T} \cdot \left(\frac{\lambda}{4\pi d}\right)^2 \cdot \frac{1}{k}$$

Beachte:

- $N_0 = k \cdot T$
Falls noise durch Verstärker geht: $N_0 = kTF$
- Noise power: $P_n = N_0 \cdot B$, B : bandwidth
- $P_{r,\min} = SNR \cdot P_n$ (minimum receive power)

Falls E_b/N_0 für zB den Input angegeben ist → dies ist das Input SNR!!

9.2 Link Budget



Obere Kurve: Signalleistungsdichte
Untere Kurve: Rauschleistungsdichte

Die Gesamt-rauschzahl wird durch die frühen Verstärker, also vor allem den Low Noise Amplifier LNA, dominiert! Alle weiteren F s werden durch G_1 geteilt.

Gain-Berechnungen (Leistung!):

$$G[\text{dB}] = 10 \cdot \log_{10}(G[1]) \Leftrightarrow G[1] = 10^{\frac{G[\text{dB}]}{10}}$$

Signal-to-Noise:

$$SNR_{\Delta f}(f_c) = \frac{P_{\text{out}}(f_c)}{S_{N_{\text{out}}}(f_c) \cdot \Delta f}, \quad SNR[\text{dB}] = 10 \cdot \log_{10}\left(\frac{P_{\text{sig}}}{P_n}\right)$$

Sofern die Bandbreite eng genug ist, damit lineare Näherung akkurat.

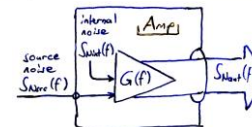
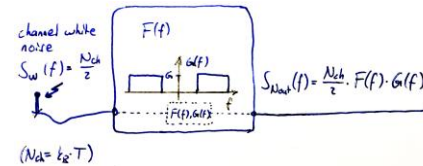
Downlink Budget eines Satelliten

$$\left(\frac{C}{N_0}\right)_{\text{rec}} = \left(\frac{P_r}{kT_e}\right) = EIRP_{\text{sat}} \left(\frac{G_r}{T_e}\right)_{\text{earth}} \left(\frac{\lambda}{4\pi d}\right)^2 \frac{1}{k} = \left(\frac{P_r}{N_0}\right)_{\text{DL}}$$

9.3 Rauschzahl (Noise Figure) p.523/526

Das Modell geht davon aus, dass am Eingang des i -ten Verstärkers die Rauschleistung N_i vorliegt. Alles Rauschen, das durch die internen Komponenten des Verstärkers verursacht wird, wird in Relation zu N_i , $[N_i] = W$ angegeben und resultiert schliesslich in der Rauschzahl F_i , welche das Verhältnis von Eingangs- und totalem Rauschen widerspiegelt. Vor der Verstärkung mit dem Gain G_i und Bandlimitierung liegt also eine Rauschleistung von $F_i N_i$ an.

Weiter gehen wir in der Vorlesung davon aus, dass N_i im Falle eines Wireless-Empfängers aus bandlimitiertem weissen Rauschen resultiert. Dieses hat eine PSD von $S_W(f) = \frac{N_0}{2}$, wobei $N_0 = k_B T$ (Definition unterscheidet sich zu Haykin), $[N_0] = \frac{W}{\text{Hz}}$ die sogenannte „Noise Spectral Density“ ist. Sofern wir eine Übertragung im Passband betrachten, können wir die Rauschleistung mittels $\int_{-\infty}^{\infty} S_W(f) G(f) \cdot df$ zu $N_1 = k_B T B$ mit der Bandbreite B bestimmen.



Die **(mittlere) Rauschzahl F_0** (also eine Vereinfachung der frequenzabhängigen Rauschzahl $F(f)$)

$$F(f) = \frac{S_{N_{\text{out}}}(f)}{G(f) S_{N_{\text{src}}}(f)} = \frac{SNR_{\text{Source}}(f)}{SNR_{\text{Output}}(f)}$$

wird über die Rauschleistung wie folgt berechnet:

$$F_0 = \frac{\int_{-\infty}^{\infty} S_{N_{\text{out}}}(f) \cdot df}{\int_{-\infty}^{\infty} G(f) S_{N_{\text{src}}}(f) \cdot df}$$

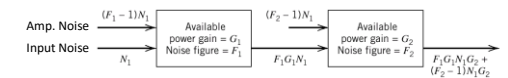
Wobei: $S_{N_{\text{src}}}(f)$ noise power des Inputs, $S_{N_{\text{out}}}(f)$ noise power des outputs, $G(f)$ der power gain.

Falls two-port noise-free: $S_{N_{\text{out}}}(f) = G(f) \cdot S_{N_{\text{src}}}(f)$ und somit $F(f) = 1$
Beachte, dass Quell- und Output PSD durch die Gainfunktion des Verstärkers bandlimitiert werden.

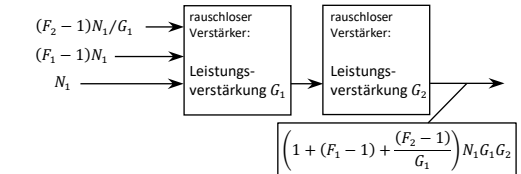
Ist die Frage nach der **Rauschleistung am Ausgang**, so ist der Term

$$P_n = \int_{-\infty}^{\infty} S_{N_{\text{out}}}(f) \cdot df \text{ gemeint.}$$

Um das Rauschen von kaskadierten Verstärkern aussagekräftig beschreiben zu können, ersetzt man der einzelnen Stufen durch ideale, rauschlose Varianten und rechnet dafür deren internes Rauschen als virtuelle Rauschquelle an den Eingang des ersten Verstärkers.



Im Beispiel mit zwei Verstärkern ergibt die Rückwärtspropagation von $(F_2 - 1)N_1$ eine zusätzliche Rauschquelle von $\frac{(F_2 - 1)N_1}{G_1}$.



Welche Rauschleistung kann ich dem Rauschprozess (mit gegebenem Quell-/Rauschwiderstand R_s) entnehmen?

$$N_1 = \left(\frac{\sqrt{4k_B T R_s \Delta f}}{2R_s}\right)^2 R_s = \frac{k_B T \Delta f}{2}, \quad k_B T|_{T=300 \text{ K}} = -174 \frac{\text{dBm}}{\text{Hz}}$$

Äquivalente Rauschtemperatur T_e :

Temperaturerhöhung, die man machen müsste, um am Ausgang N_2 zu erhalten.

$$T_e = T(F - 1), \quad F = \frac{N_2}{N_2 - N_d} = \frac{T + T_e}{T}$$

$$\text{mit } N_2 = G N_1 + N_d = G k_B T_e \Delta f \text{ und } N_d = G k_B T_e \Delta f$$

N_1 : Am Eingang des Verstärkers anliegende noise power.
 N_d : Alles durch interne Komponenten verursachtes Rauschen.
 N_2 : Am Ausgang des Verstärkers anliegende noise power.

Bei low-noise Verstärkern möglicherweise sinnvollerer Mass, weil die Rauschzahl in allen Instanzen nahe 1 ist.

Total am 1. Verstärker anliegendes Rauschen (Rückpropagation):

$$\text{Total noise figure: } F_{\text{tot}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} + \dots$$

$$T_{e,\text{tot}} = T_{e1} + \frac{T_{e2}}{G_1} + \frac{T_{e3}}{G_1 G_2} + \frac{T_{e4}}{G_1 G_2 G_3} + \dots$$

Die Rauschleistung am Ausgang ist dann:

$$P_{n,\text{out}} = P_{\text{noise}} \cdot F_{\text{tot}} \cdot \prod_i G_i$$

Signal to noise nach mehreren noisy Filtern.

$P_{\text{noise}} = k_B B T$ das Rauschen am Input.

$$SNR_{\text{out}} = \frac{P_{\text{out}}}{P_{n,\text{out}}} = \frac{P_{\text{in}} G_{\text{tot}}}{P_{\text{noise}} G_{\text{tot}} F_{\text{tot}}} = \frac{SNR_{\text{in}}}{F_{\text{tot}}}$$

$$(SNR_{\text{out}})_{\text{dB}} = (SNR_{\text{in}})_{\text{dB}} - (F_{\text{tot}})_{\text{dB}}$$

Fading mit $\gamma_0 = \frac{E_b}{N_0} E[a^2]$

Type of Signaling	Exact Formula for the Bit Error Rate P_b	Approximate Formula for the Bit Error Rate, Assuming Large γ_0
Coherent binary FSK	$\frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{1 + \gamma_0}} \right)$	$\frac{1}{4\gamma_0}$
Coherent binary PSK	$\frac{1}{2} \left(1 - \sqrt{\frac{\gamma_0}{2 + \gamma_0}} \right)$	$\frac{1}{2\gamma_0}$
Binary DPSK	$\frac{1}{2(1 + \gamma_0)}$	$\frac{1}{2\gamma_0}$
Noncoherent binary FSK	$\frac{1}{2 + \gamma_0}$	$\frac{1}{\gamma_0}$

9.4 Free-space propagation model

Operationen im Fernfeld $d_r \gg 2D^2/\lambda$. Übertragungsantenne wird als Punktquelle modelliert (uniforme Abstrahlung in alle Richtungen).

- Isotropische Quelle**, idealisierte Referenz mit Leistungsdichte $\rho(d) = \frac{P_t}{4\pi d^2} \quad [W/m^2]$
 $\phi = d^2 \rho(d) \quad [W/\text{unit solid angle}]$
- Leistung des Signals $r(t)$** : $P_r = \frac{1}{2} |r(t)|^2 \Leftrightarrow P_r = \frac{1}{T} \int_0^T |r_x(t)|^2 dt$
- Reale Antenne mit Strahlungsintensität $\phi(\theta, \varphi)$ und Leistung $P = \int \Phi(\theta, \varphi) d\Omega = \int \Phi(\theta, \varphi) \sin\theta d\theta d\varphi$

9.5 Antennencharakteristiken

Richtungsgewinn (directivity gain):

$$g(\theta, \varphi) = \frac{\Phi(\theta, \varphi)}{P_{av}} = \frac{\Phi(\theta, \varphi)}{P/(4\pi)}$$

Direktivität (directivity): Gewinn im Vergleich zu einer isotropen Antenne

$$D = \max_{\theta, \varphi} \{g(\theta, \varphi)\}$$

Leistungsgain (power gain):

$$G = \eta_{\text{radiation}} \cdot D$$

mit $\eta_{\text{radiation}}$ der Abstrahleffizienz

Effective isotropically radiated power (EIRP):

$$[EIRP = P_t G_t] \quad [EIRP] = W$$

mit P_t, G_t die abgestrahlte Leistung und der Antennengewinn beim Senden.

Die EIRP ist ein Mass für die maximale Störungsleistungsdichte, die ich erzeugen kann / darf.

Antennen-Abstrahlbreite (antenna beamwidth):

Winkel zwischen den Richtungen, in denen noch die halbe Abstrahlleistung zu verzeichnen ist.

Apertur:

Sei eine Leistungsdichte von $1 \frac{W}{m^2}$ gegeben. Falls die Antenne 1 W herausholt, sagt man, sie habe eine Wirkfläche / Apertur von $1 m^2$.

Effektive Apertur:

$$A = \frac{\lambda^2}{4\pi} G = \text{Apertur einer isotropen Antenne} \cdot \text{Leistungsgain}$$

Apertur-Effizienz (aperture efficiency):

$$\eta_{ap} = \frac{A}{A_{ph}}, \quad A_{ph}: \text{physikalische Fläche}$$

Für eine Dish Antenne: $A_{ph} = \frac{\pi D^2}{4}$

Durchschnittliche Leistung (avg. Power):

$$P_{av} = \frac{P}{4\pi}$$

Generelle Formeln:

- $\lambda = c/f$
- $P_{r,min} = SNR \cdot P_n$ (minimum receive power)
- $\Delta t = \frac{\Delta l}{c}$
- $\phi = 2\pi f \Delta t = 2\pi \frac{f}{c} \Delta l = 2\pi \frac{1}{\lambda} \Delta l$

9.6 Wellenausbreitung

Sei G_t, G_r der Antennengewinn beim Senden bzw. Empfangen.

$$G_r = \frac{4\pi A_{ph} \eta_{ap}}{\lambda^2}$$

Annahme: Der Empfänger ist im Fernfeld des Senders.

Eingehende Leistungsdichte im Abstand d (direkte Sicht):

$$P_r = \frac{EIRP}{4\pi d^2} \cdot A_r = \left[\frac{P_t G_t A_r}{4\pi d^2} \right], \quad [P_r] = W$$

Friis's free-space equation:

Die verfügbare Empfangsleistung ist gegeben durch

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d} \right)^2, \quad [P_r] = W$$

$$\Leftrightarrow d = \frac{\lambda}{4\pi} \sqrt{\frac{P_t G_t G_r}{P_r}}$$

weil $A_r = \frac{\lambda^2}{4\pi} G_r$

$$P_r [\text{dB}] = P_t [\text{dB}] + G_t [\text{dB}] + G_r [\text{dB}] + \left(\frac{\lambda}{4\pi d} \right)^2 [\text{dB}]$$

Pfadverlust (path loss):

Unterschied zwischen Sende- und Empfangsleistung in dB.

$$PL[\text{dB}] = 10 \cdot \log_{10} \left(\frac{P_t}{P_r} \right) = -10 \cdot \log_{10} (G_t G_r) + 10 \cdot \log_{10} \left(\left(\frac{4\pi d}{\lambda} \right)^2 \right)$$

10 Informationstheorie

10.1 Definitionen

Symbolsequenz

Sei s_k der k -te Wert, den die Zufallsvariable S annehmen kann (Das Alphabet (Wahrscheinlichkeitsraum) ist also $\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\}$). Die Wahrscheinlichkeit, mit der das Symbol s_k auftritt, ist definiert als $P[S = s_k] = p_s(s_k) = p_k$.

Je kleiner p_k , desto mehr Information steckt in der Beobachtung von $S = s_k$.

Informationsgehalt:

Eine Quelle emittiert eine Nachricht $S = \{s_0, s_1, \dots, s_{K-1}\}$, S ist eine Zufallsvariable mit $P[S = s_k] = p_k, \sum_{k=0}^{K-1} p_k = 1$. Der Informationsgehalt eines Symbols ist eine Funktion der Auftretenswahrscheinlichkeit des Symbols und wird in # Bit angegeben. Logarithmen sind in der Informationstheorie, sofern nicht anders vermerkt, alle zur Basis 2 zu verstehen.

$$[I(s_k)] = -\log_2(p_k), \quad [I(s_k)] = \text{Bit}$$

- $p_k = 1 \Rightarrow I(s_k) = 0$
- $0 \leq p_k \leq 1 \Rightarrow I(s_k) \geq 0$
- $p_k < p_l \Rightarrow I(s_k) > I(s_l)$
- s_k und s_l statistisch unabhängig $\Rightarrow I(s_k s_l) = I(s_k) + I(s_l)$

Entropie:

Die Entropie einer Symbolsequenz ist als deren durchschnittlicher Informationsgehalt zu interpretieren. Da die Symbolverteilung diskret und gedächtnislos ist, kann dazu sehr einfach der Erwartungswert wie folgt berechnet werden:

$$H(S) = E[I(S)] = \sum_{k=0}^{K-1} I(s_k) \cdot p_k = - \sum_{k=0}^{K-1} \log_2(p_k) \cdot p_k, \quad [H(S)] = \text{Bit}$$

- $0 \leq H(S) \leq \log_2(K)$ (*) Gleichheit, wenn S gleichverteilt.
- $H(S) = \log_2(K) \Leftrightarrow \forall k: p_k = 1/K$, maximale Entropie
Für eine Quelle $S = \{s_0, s_1, \dots, s_{K-1}\}$ mit K Symbolen.
- $H(S) = 0 \Leftrightarrow \exists! k: p_k = 1 \Rightarrow p_i = 0 \forall i \in \{0, 1, \dots, K-1\} \setminus \{k\}$
- $H(S^N) = N \cdot H(S)$ für die Nte Extension der Source (siehe 10.3)

10.2 Erweiterung einer diskreten, gedächtnislosen Quelle / Quellerweiterung

Sei der Symbolraum einer diskreten, gedächtnislosen Quelle durch $\mathcal{S} = \{s_0, s_1, \dots, s_{K-1}\}$ gegeben.
Eine Sequenz der Länge $L = N \cdot R$ von Symbolen wird nun in R Blöcke der Länge N aufgeteilt, was dann einen neuen Symbolraum $\mathcal{B} = \mathcal{S}^N = \{b_0, b_1, \dots, b_{M-1}\}$ mit $M = K^N$ Symbolen für die erweiterte Quelle ergibt.
Jeder dieser Blöcke ist nun also eine Realisation aus dem Symbolraum \mathcal{S}^N . Es gilt

$$H(\mathcal{S}^N) = N \cdot H(\mathcal{S})$$

„Wenn wir eine Sequenz von N statistisch unabhängigen Symbolen beobachten, erhalten wir N mal die Information, die ein einziges Symbol für sich geliefert hätte.“

ACHTUNG: Bei einer Quellerweiterung der Ordnung N muss die mittlere Codewortlänge normalisiert werden auf die Anzahl codierter elementar Symbole:

$$\bar{L}_{ext} = \frac{\bar{L}_{src}}{N}$$

Wobei \bar{L}_{ext} die mittlere Codewortlänge der erweiterten Quelle ist und \bar{L}_{src} ist die mittlere Codewortlänge der ursprünglichen Quelle. \bar{L}_{ext} ist dann die effektive Anzahl bit pro Quellsymbol.

Bsp: Sei das Alphabet einer Quelle $\mathcal{S} = \{s_1, s_2, s_3\}$ mit p_1, p_2, p_3 gegeben. Dann ist die Erweiterte Quelle der Ordnung 2 wie folgt:

Supersymb.	Wahrsch.	Supersymb.	Wahrsch.
$s_0 s_0$	$p_0 p_0 = q_0$	$s_2 s_0$	$p_2 p_0 = q_6$
$s_0 s_1$	$p_0 p_1 = q_1$	$s_2 s_1$	$p_2 p_1 = q_7$
$s_0 s_2$	$p_0 p_2 = q_2$	$s_2 s_2$	$p_2 p_2 = q_8$
$s_1 s_0$	$p_1 p_0 = q_3$	$= q_{K-1} = q_{32-1}$	
$s_1 s_1$	$p_1 p_1 = q_4$		
$s_1 s_2$	$p_1 p_2 = q_5$		

Wenn **sämtliche Wahrscheinlichkeiten aller Symbole der Quelle eine Zweierpotenz sind**, so wird bei einer Erweiterung der Quelle bereits bei Ordnung 2 das Shannon'sche Quellcodierungstheorem mit Gleichheit erfüllt, also $\bar{L} = H(S)$!

10.3 Quellcodierung

Idee: repräsentiere häufiger vorkommende Symbole mit kurzen Sequenzen und umgekehrt.

Die durchschnittliche Codewort-Länge (bit pro Symbol), falls das k -te Symbol Länge L_k hat, ist:

Mittlere Codewortlänge

$$\bar{L} = \sum_{k=0}^{K-1} L_k \cdot p_k$$

10.3.1 Shannon's Source Coding Theorem

Die mittlere Codewortlänge ist gegen unten durch die Entropie der Quelle begrenzt:

$$\bar{L} \geq L_{min} := H(S)$$

Wir können \bar{L} verbessern indem wir die Quelle erweitern (siehe 10.3) \rightarrow z.B. Ordnung 2, kreierte alle möglichen Symbolkombinationen aus 2 Symbolen (z.B. $s_0 s_0, s_0 s_1, s_1 s_0, s_1 s_1$)

Indem wir grössere Quellerweiterungen kreieren, kommt die durchschnittliche Codewort-Länge \bar{L} immer näher und ihr Limit $H(S)$.

Die **Codeeffizienz** berechnet sich zu $\eta = \frac{L_{min}}{\bar{L}} \leq 1$

Die Effizienz eines guten Codes geht gegen 1 für grosse Quellerweiterungen.

10.4 Präfix-Codes

Ein Präfix-Code erfüllt die Eigenschaft, dass kein Codewort der Präfix eines anderen Codeworts ist.

Ein Codewort hat n bits und l prefix bits.

Für jede diskrete, gedächtnislose Quelle existiert ein Präfix-Code, der die folgende Beziehung erfüllt:

$$H(S) \leq \bar{L} < H(S) + 1$$

Durch Quellerweiterung kann der +1-Term vernachlässigbar gemacht werden:

$$\begin{aligned} H(S^n) &\leq \bar{L}_n < H(S^n) + 1 \\ \Leftrightarrow H(S) &\leq \frac{\bar{L}_n}{n} < H(S) + \frac{1}{n} \end{aligned}$$

Im Limit $n \rightarrow \infty$ (grosse Quellerweiterungen) kommen die beiden Schranken arbiträr nahe.

10.4.1 Huffman-Code

Der Huffman Algorithmus erzeugt für eine gegebene Sequenz eine Codierung, welche die mittlere Codewortlänge minimiert.

Voraussetzung ist eine gedächtnislose Quelle.

p_k	Symbol		Codewort	Länge L
0.9	s_0		00	2
0.2	s_4		10	2
0.2	s_5		11	2
0.1	s_3		010	3
0.1	s_1		011	3

- Finde die zwei (Pseudo-)Symbole mit der geringsten Auftretenswahrscheinlichkeit.
- Fasse diese Symbole zu einem Supersymbol zusammen. Berechne dessen Auftretenswahrscheinlichkeit.
- Wiederhole 4 und 5, bis nur noch ein Supersymbol übrig ist. (vnr)
- Teile jedem Ast des entstandenen Entscheidungsbaums eine Bitfolge zu, indem jede Abzweigung eine 0 oder 1 encodiert. (vrnl)

Im Beispiel ist die mittlere Codewortlänge

$$\bar{L} = 0.4 \cdot 2 + 0.2 \cdot 2 + 0.2 \cdot 2 + 0.1 \cdot 3 + 0.1 \cdot 3 = 2.2 \text{ Bit}$$

Und die Entropie ist:

$$H(S) = -0.4 \log_2(0.4) - 0.2 \log_2(0.2) - 0.2 \log_2(0.2) - 0.1 \log_2(0.1) = 2.12$$

Nachteile von Huffman-Codes:

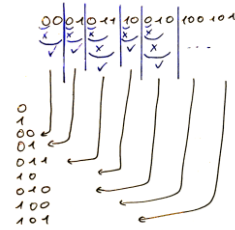
- Encoder muss a priori alle Symbolwahrscheinlichkeiten kennen
- Memory in der Quelle führt zu Redundanz und erhöht die Komplexität

Quellerweiterung anstatt S nehme die erweiterte Quelle S^N und wende Huffman-Coding auf alle Symbole $s_0^N, s_1^N, \dots, s_{N-1}^N$ an.

Die neue mittlere Codewortlänge ist: $\bar{L}_{ext} = \frac{\bar{L}_{src}}{N}$

10.4.2 Lempel-Ziv-Code

Entscheidung auf Subsequenzen:



Anfangs haben wir nur die Symbolsequenzen 0 und 1 (Innovationsbits). Wann immer dann eine neue Sequenz auftaucht, wird diese ins Wörterbuch aufgenommen. Jede neue Sequenz hat die Form: $seq_{ALT} + innov.bi$

Numerical Positions:	1	2	3	4	5	6	7	8	9
Subsequences:	0	1	00	01	011	10	010	100	101
Numerical representations:	1-1	1-2	4-2	2-1	4-1	6-1	6-2		
Binary encoded blocks:	0010	0011	1001	0100	1000	1100	1101		

Erläuterung am an numerical position 9: Die subsequence 101_{bin} setzt sich aus der subsequence 10_{bin} und dem Innovationsbit 1 zusammen. Subsequence 10_{bin} hat numerical position 6_{dec} $\hat{=}$ 110_{bin}. Innovationsbit 1 hat numerical position 2_{dec}.

- Die numerical representation ist somit 6 - 2
- Der binary encoded block ist 1101_{bin}

Erstelle nun ein Wörterbuch basierend auf den durchnummerierten Subsequenzen:

Index Bin	Index Dez	Basissequenz (Referenz)	Innovations-symbol	codierte Subsequenz
...0000	0	\emptyset	\emptyset	\emptyset
...0001	1	...0000 / 0	0	0
...0010	2	...0000 / 0	1	1
...0011	3	...0001 / 1	0	010
...0100	4	...0001 / 1	1	011
...0101	5	...0100 / 4	1	0111
...0110	6	...0010 / 2	0	110
...0111	7	...0100 / 4	0	0110
...1000	8	...0110 / 6	0	1010
...1001	9	...0110 / 6	1	1011

Typischerweise werden 12 Bit pro Codewort verwendet, was dann einer Wörterbuchgröße von 4096 Einträgen entspricht.

In einigen Varianten werden erst nur Indexreferenzen mit einer minimalen Anzahl führender Nullen gesendet und sobald die Wörterbuchgröße eine Zweierpotenz überschreitet, wird die Anzahl führender Nullen um eins erhöht.

Ist die maximale Wörterbuchgröße erreicht, muss entschieden werden, was mit diesem passiert. I.d.R. werden weniger genutzte Einträge nach einem vorher definierten Schema ersetzt.

Das Beispiel hat einen längeren Output erzeugt als das Original. Das kann passieren, wenn die Nachricht eine hohe Entropie hat (sprich: zu zufällig ist) oder zu kurz ist, dass die asymptotisch steigenden Platzersparnisse des Lempel-Ziv Algorithmus zum Tragen kommen.

10.5 Diskreter, gedächtnisloser Kanal

Notation:

Eingangsalphabet: $X \in \{x_0, x_1, \dots, x_{J-1}\}$
 Ausgangsalphabet: $Y \in \{y_0, y_1, \dots, y_{K-1}\}$
 Übergangswahrscheinlichkeiten: $p(y_k|x_j) = P[Y = y_k | X = x_j]$
 Im Allgemeinen ist $J \neq K$.

X und Y sind beides statistisch unabhängige Zufallsvariablen. Falls die I/O Alphabete finit sind, sind X und Y diskrete Zufallsvariablen (sonst reelle).

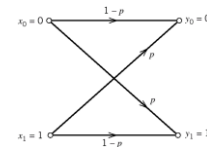
Die Übergangswahrscheinlichkeiten lassen sich in eine Matrix packen:

$$P = \begin{bmatrix} p(y_0|x_0) & \dots & p(y_{K-1}|x_0) \\ \vdots & \ddots & \vdots \\ p(y_0|x_{J-1}) & \dots & p(y_{K-1}|x_{J-1}) \end{bmatrix}$$

Es gilt

$$p(y_k) = P[Y = y_k] = \sum_{j=0}^{J-1} p(y_k|x_j) \cdot p(x_j)$$

Binary, symmetric channel (BSC) ($J = K = 2$):



Mit:

$$\begin{aligned} P(X = 0) &= p_0 \\ P(X = 1) &= p_1 = 1 - p_0 \end{aligned}$$

Bei einem asymmetrischen Kanal wäre die Fehlerwahrscheinlichkeit $p(y_0|x_1) \neq p(y_1|x_0)$.

Die Übergangsmatrix ist:

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

$$P[Y = 1] = p_1(1-p) + p_0p = pp_0 + (1-p)(1-p_0) = z$$

$$P[Y = 0] = 1 - z$$

Gemeinsame Information:

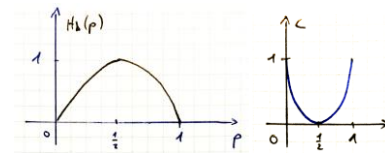
$$I(X;Y) = I(Y;X) = H(Y) - H(Y|X)$$

Binäre Entropiefunktion:

$$H(Y) = - \sum_y \log_2(p(y)) \cdot p(y) = \left[H_b(z) = -z \cdot \log_2(z) - (1-z) \log_2(1-z) \right]$$

Zudem: $H(Y|X) = H_b(p) = -p \cdot \log_2(p) - (1-p) \log_2(1-p)$

Somit: $I(X;Y) = H(Y) - H(Y|X) = H_b(z) - H_b(p)$



Die Frage nach der Kanalkapazität eines binären Übertragungskanals mit gegebener Fehlerwahrscheinlichkeit kann nun einfach mit obiger Formel beantwortet werden, nämlich

$$C = \max_{(p(x_j))} \{I(X;Y)\} = -H_b(p) + \max_{p_0} \{H_b(z)\}$$

Dies ist maximal für $z = \frac{1}{2}$, wie man dem Plot entnehmen kann.

$$z = \frac{1}{2} = p_1(1-p) + p_0p = pp_0 + (1-p)(1-p_0) \Leftrightarrow p_0 = \dots$$

Also ist $\underline{C} = -H_b(p) + H_b\left(\frac{1}{2}\right) = 1 - H_b(p)$:

10.6 gemeinsame Information

Übertrage X , beobachte Y .

„Welche Information über X kriege ich, wenn ich Y beobachtet habe?“

$$\begin{aligned} I(X;Y) &:= H(Y) - H(Y|X) = I(Y;X) \\ I(X;Y) &:= H(Y) + H(X) - H(Y,X) \end{aligned}$$

mit der **bedingten Entropie**

$$\begin{aligned} H(Y|X) &:= E[-\log_2(p(Y|X))] = - \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(y_j, x_k) \cdot \log_2(p(y_j|x_k)) = \\ &= \sum_{k=0}^{K-1} p(x_k) \cdot H(Y|X = x_k) \\ \rightarrow H(Y|X = x_k) &= - \sum_{j=0}^{J-1} P(Y = y_j | X = x_k) \cdot \log_2(P(Y = y_j | X = x_k)) \end{aligned}$$

„Je mehr X und Y gemeinsam haben, desto mehr weiss ich über X durch blosser Beobachtung von Y . D.h. $H(X|Y)$ ist klein und $I(X;Y)$ gross.“

Eigenschaften:

- Symmetrie:

$$I(X;Y) = I(Y;X)$$

$$H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- Nichtnegativität:

$$I(X;Y) \geq 0$$

$$\Leftrightarrow H(X) \geq H(X|Y)$$

Falls Gleichheit in der letzten Formel gilt, dann sind X und Y statistisch voneinander unabhängig.

Gemeinsame Entropie:

$$H(X,Y) = E[-\log_2(P(X,Y))] = - \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \cdot \log_2(p(x_j, y_k))$$

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

10.7 Channel capacity

$$\begin{aligned} I(X;Y) &= \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(x_j, y_k) \cdot \log_2\left(\frac{p(y_k|x_j)}{p(y_k)}\right) \\ &= \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} p(y_k|x_j) \cdot p(x_j) \cdot \log_2\left(\frac{p(y_k|x_j)}{\sum_{j=0}^{J-1} p(y_k|x_j) \cdot p(x_j)}\right) \end{aligned}$$

Channel capacity: maximale gemeinsame Information, die durch eine gegebene Eingangsverteilung erreicht werden kann.

$$C = \max_{\{p(x_j)\}} \{I(X;Y)\}, \quad |C| = \frac{\text{Bit}}{\text{Kanalbenutzung}}$$

10.8 Differenzielle Entropie

Wir betrachten nun nicht mehr Quellen mit diskretem Alphabet, sondern Quellen mit stetigem Alphabet. Die Zufallsvariablen X und Y sind dann reelle Zufallsvariablen mit Wahrscheinlichkeitsdichten $f_X(x)$ und $f_Y(y)$.

Die Kanalkapazität wird undefiniert: $C_{neu} := \frac{C}{R_b}$

Differenzielle Entropie

$$h(X) = - \int_{-\infty}^{\infty} f_X(x) \cdot \log_2(f_X(x)) dx, \quad I(X;Y) = h(X) - h(X|Y)$$

- $X \sim \mathcal{N}(\mu, \sigma^2)$

$$h(X) = \frac{1}{2} \log_2(2\pi e \sigma^2)$$

- $X \sim \text{unif}(0, a)$

$$h(X) = \log_2(a)$$

Bemerkung: Gauss'sche ZVs haben die grösste differentielle Entropie unter allen ZVs mit Varianz σ^2 .

10.9 Kanalcodierung

Im Gegensatz zur Quellcodierung geht es bei der Kanalcodierung nicht darum, die Quellinformation möglichst stark zu kompaktieren, sondern **eine gegebene Datenfolge durch geschicktes Hinzufügen von Redundanz weniger störanfällig zu machen**. Im Ideal- und Regelfall können Bitfehler dann nahezu ausgeschlossen werden.

10.10 Shannon's Channelkapazität-Theorem

Gegeben: Diskrete, gedächtnislose Quelle mit input alphabet S

- Ein Symbol emittiert pro T_s Sekunden. Die Informationsrate ist somit $H(S)/T_s$
 - Ein codiertes Symbol übermittelt pro T_c Sekunden
- Wenn

$$\frac{H(S)}{T_s} < \frac{C}{T_c}$$

gilt, dann existiert ein Kanalcode, der eine beliebige kleine Fehlerwahrscheinlichkeit liefert, solange die Codewortlänge gegen ∞ strebt.

$\frac{H(S)}{T_s}$ ist die Informationsrate der Quelle (Symbole mit der Informationsmenge $H(S)$, welche alle T_s generiert wird) und $\frac{C}{T_c}$ die Kanalratenkapazität (mit C der Informationsmenge, die pro T_c erfolgreich übertragen werden kann).

Umstellen der Ungleichung ergibt

$$H(S) \cdot \frac{T_c}{T_s} < C$$

wobei $r = \frac{T_c}{T_s}$ auch für die Coderate stehen kann, d.h. wie viele Datenbit auf wie viele Kanalbit gemappt werden.

Block codes Die Quelle emittiert $S = \{0,1\}$ mit gleicher W'keit, somit ist $H(S) = 1$. Die Nachricht wird unterteilt in Blöcke an k -bits (data) und diese werden auf n -bits gemapped (channel input).

Die Coderate ist $r = \frac{k}{n} \leq 1$ und ist ein Mass für die Redundanz. Shannon's Channelkapazitäts-Theorem sagt:

$$\frac{H(S)}{T_s} = \frac{C}{T_c} \Rightarrow r < C$$

$$\text{Da } r = \frac{k}{n} = \frac{1/T_s}{1/T_c} = \frac{T_c}{T_s}$$

Repetition codes Jedes Symbol wird $n = 2m + 1$ mal übertragen, der Decoder nimmt dann das Symbol, welches mindestens $m + 1$ mal übertragen wurde.

$$P_e = \sum_{i=m+1}^n \binom{n}{i} p^i (1-p)^{n-i}$$

10.11 Physikalischer Kanal

Betrachte einen Band- und Leistungs-limitierten AWGN Kanal.

- $X(t)$ bandlimitiert mit B , und sampled $X_k, k = 1, \dots, K$ mit $K = 2BT$ die Anzahl Samples nach T Sekunden
- $Y_k = X_k + N_k, k = 1, \dots, K$. Wir haben AWGN mit zero-mean und Dichte $N_0/2$. Da die Samples bandlimitiert sind, können wir N_k brauchen, das gaussian noise sample mit zero-mean und Varianz $\sigma_{N_k}^2 = \frac{N_0}{2} \cdot 2B = N_0B$. Dann: $\sigma_{Y_k}^2 = P + N_0B$

Wobei $E[X^2] = P$, Leistungslimitiert. (P : avg. Transmit power)

Somit ist die differentielle Entropie von bandlimitiertem Rauschen:

$$h(Y_k) = \frac{1}{2} \log_2(2\pi e(P + N_0B)) \quad h(N_k) = \frac{1}{2} \log_2(2\pi e N_0B)$$

- $C = \max_{p_{X_k}(x)} \{I(X_k; Y_k) : E[X_k^2] = P\}$
- $I(X_k; Y_k) = h(Y_k) - h(Y_k|X_k) = h(Y_k) - h(N_k)$

10.11.1 Informationskapazitäts-Theorem

Die Informationskapazität eines continuous channel mit Bandbreite B [hertz], gestört durch AWGN mit power spectral density $N_0/2$ und limitiert in der Bandbreite auf B , ist gegeben durch:

$$C = B \cdot \log_2 \left(1 + \frac{P}{N_0B} \right) \text{ [bits/sec]}$$

$$\text{Oder } C = \frac{1}{2} \log_2 \left(1 + \frac{P}{N_0B} \right) = \frac{1}{2} \log_2 \left(1 + \frac{2Es}{N_0} \right) \text{ bits pro Übertragung,}$$

$$\text{mit } E_s = \frac{P}{2B}$$

Annahmen (ideales System):

- Übertragungsrate [#bits/sec]: $R_b = C$
- Übertragungsleistung (Energie pro Bit): $P = E_b \cdot R_b = E_b \cdot C$

Die Bandbreiteneffizienz $\frac{C}{B}$ ist bei einem idealen System definiert durch

$$\frac{C}{B} = \log \left(1 + \frac{E_b C}{N_0 B} \right), \quad \left[\frac{C}{B} \right] = \frac{\text{Bit}}{s \cdot \text{Hz}}$$

„Wie viele Bit kann ich pro Sekunde in 1Hz Bandbreite übertragen?“

Damit kann die Fragestellung „Welcher Störabstand $\frac{E_b}{N_0}$ ist nötig um eine bestimmte Bandbreiteneffizienz zu erreichen?“ beantwortet werden.

$$\frac{E_b}{N_0} = \frac{2^{C/B} - 1}{C/B}$$

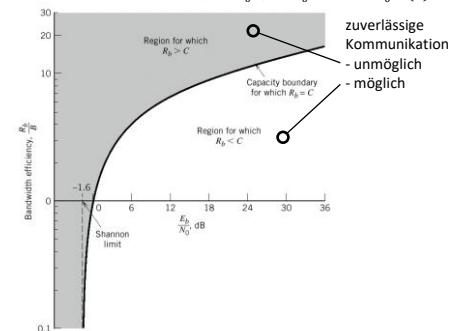
Das **Shannon-Limit**, das den minimalen Störabstand im Empfänger angibt, der nötig ist, um fehlerfrei empfangen zu können, ist

$$\frac{E_b}{N_0} \xrightarrow{C \rightarrow 0} \ln(2) \approx -1.6 \text{ dB}$$

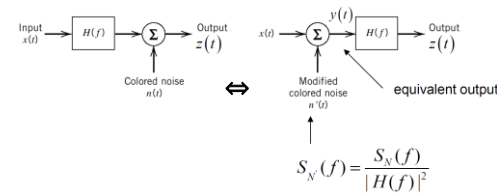
$\frac{C}{B} \rightarrow 0^+$ erhält man sinnvollerweise, indem man die Bandbreite B gegen ∞ streben lässt ($B \rightarrow \infty$). D.h. bei unendlich verfügbarer Bandbreite braucht man einen Störabstand von -1.6 dB.

Bei begrenzter Sendeleistung ergibt sich

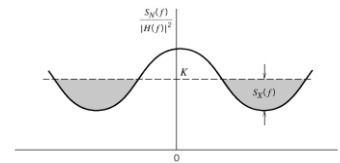
$$C_\infty = \lim_{B \rightarrow \infty} B \cdot \log \left(1 + \frac{P}{N_0B} \right) = \frac{P}{N_0} \log(e) = \frac{P}{N_0 \ln(2)}$$



10.12 Informationskapazität eines Kanals mit Colored Noise



Die optimale Verteilung der Leistung am Empfänger $S_X(f)$ richtet sich an $\frac{S_N(f)}{|H(f)|^2}$, und zwar genau gleich, wie sich Wasser in einem Gefäß füllen würde:



Die gesamte Empfangsleistung entspricht der grauen Fläche.

11 Forward Error Correction

Füge Redundanz hinzu um die Fehlerrate zu verkleinern, bei fixiertem Enegie pro Bit zu noise Verhältnis.

11.1 Linear Block Codes

k message bits werden auf n coded bits gemapped. Die coded bits werden dann übertragen.

$$\text{Coderate: } r = \frac{k}{n} < 1$$

ACHTUNG: Falls Prüfungsfrage ist, die message bits anzugeben, nehme das Codeword und schneide die Parity bits ab!

11.1.1 Systematic Linear Block Codes

Systematisch \rightarrow Message bits m_i werden unverändert übertragen, $(n - k)$ parity bits b_i werden hinzugefügt.

ACHTUNG: Wir rechnen hier immer mit mod2-Arithmetik (also $0+0=0, 0+1=1+0=1, 1+1=0$)

- Die parity bits sind linear Kombinationen der message bits, mit $p_{ij} \in \{0,1\}$

$$\begin{bmatrix} b_0, b_1, \dots, b_{n-k-1} \end{bmatrix} = \begin{bmatrix} m_0, \dots, m_{k-1} \end{bmatrix} \cdot \begin{bmatrix} p_{0,0} & \dots & p_{0,n-k-1} \\ \vdots & & \vdots \\ p_{k-1,0} & \dots & p_{k-1,n-k-1} \end{bmatrix}$$

parity matrix $P \in (k \times (n-k))$

- Das Codeword in Vektorform, die Kombination von message Vektor und parity Vektor, wird durch die Generatormatrix $G \in \mathbb{R}^{k \times n}$ erzeugt:

$$\begin{aligned} c &= [c_0, \dots, c_{n-1}] = [b \ m] = m \cdot [P \ I_k] = m \cdot G \\ &\Rightarrow G = [P \ I_k], G \in \mathbb{R}^{k \times n} \end{aligned}$$

Linearität:

$$c_i + c_j = m_i G + m_j G = (m_i + m_j) G$$

- Parity check Matrix H : Wegen mod2 Arithmetik erhält man

$$H = [I_{n-k} \ P^T], H \in \mathbb{R}^{(n-k) \times n}$$

$$\begin{aligned} HG^T &= [I_{n-k} \ P^T] \cdot \begin{bmatrix} P \\ I_k \end{bmatrix} = GH^T = P^T + P = 0 \\ &\Rightarrow c \cdot H^T = 0 \end{aligned}$$

Daraus ergibt sich der Parity check des receive vector r :

- Falls $r \cdot H^T = 0 \rightarrow$ Fehlerlose Übertragung
- Denn $r = c \Rightarrow r \cdot H^T = c \cdot H^T = m \cdot GH^T = 0$
- Falls $r \cdot H^T \neq 0 \rightarrow$ Fehlerhafte Übertragung
- Denn: $r = c + e \Rightarrow r \cdot H^T = c \cdot H^T + \underbrace{e \cdot H^T}_{\neq 0} \neq 0$
- Das **Syndrom** eines receive vectors r ist: $s = r \cdot H^T$
- Beispiel: **repetition code** mit $n = 5, k = 1: G = [11111]$
 \rightarrow Wir haben $k = 1$ message bit und reptieren dies einfach 5 mal
 Fehlerwahrscheinlichkeit bei repetition codes:

$$P_e = \sum_{i=m+1}^n \binom{n}{i} p^i (1-p)^{n-i}$$

Wobei $n = 2m + 1$ ein ungerader Integer ist und p die Fehler'keit eines binary symmetric channel (also $p = P(y_0|x_1) = P(y_1|x_0)$).

- Ein Systematic Linear Block Code ist ein **Dual Code** falls gilt:

$$HG^T = 0 \text{ und } GH^T = 0$$

- Für Linear Block codes: W'keit c_2 zu empfangen, wenn c_1 gesendet $P(c_2|c_1) = \sum_{j=0}^n \alpha_j p^j (1-p)^{n-j}$, wobei p : crossover W'keit eines BSC und α_j die Anzahl error patterns mit weight j , die von c_1 aus zu c_2 führen.
- Min. Hamming Distance, falls H gegeben: Die min. Distance eines Codes ist gleich der minimalen Anzahl an lin. unabhängigen Spalten aus H . Also: $r_1 + r_2 + \dots + r_i = 0$, wobei r Spalten aus H und dann i die min. Hamming distance. Ausnahmen: Falls H eine 0-Spalte hat $\rightarrow d = 1$. Falls H zwei identische Spalten hat $\rightarrow d \leq 2$

11.1.2 Cyclic codes

ACHTUNG: Polynom Operationen bei Cyclic codes immer mit mod2 Arithmetik.

Message bits werden mittels Shift Registern codiert.

In Polynomnotation lässt sich dies darstellen als:

$$[c_0, \dots, c_{n-1}] \Rightarrow c(X) = c_0 + c_1X + c_2X^2 + \dots + c_{n-1}X^{n-1}$$

Die Exponenten repräsentieren bit shifts (vgl. z-Transformation)

Und ein Shift entspricht einer Multiplikation:

$$\begin{aligned} c^{(i)}(X) &= X^i c(X) \mod (X^n + 1) = \\ &= c_{n-i} + \dots + c_{n-1}X^{i-1} + c_0X^i + \dots + c_{n-i-1}X^{n-1} \end{aligned}$$

- Das **Generatorpolynom** $g(X)$ hat Grad $n - k$ und ist ein Faktor von

$$X^n + 1 \rightarrow g(X) = 1 + \sum_{i=1}^{n-k-1} g_i \cdot X^i + X^{n-k}$$

- Codieren der k -bit Message zum n -bit Codeword für **nicht systematische Codes**:

$$c(X) = m(X) \cdot g(X)$$

- Codieren der k -bit Message zum n -bit Codeword für **systematische Codes**:

- message Polynom: $m(X) = m_0 + m_1X + \dots + m_{k-1}X^{k-1}$
- Multipliziere $m(X)$ mit $X^{n-k} \rightarrow X^{n-k} \cdot m(X)$
- Dividiere $X^{n-k} \cdot m(X)$ durch das Generatorpolynom $g(X)$, wobei man den Rest $b(X)$ erhält ($b(X)$ vom Grad $n - k - 1$)
- Füge $b(X)$ zu $X^{n-k} \cdot m(X)$ hinzu, was das Codepolynom ergibt:

$$\begin{aligned} c(X) &= b(X) + X^{n-k} \cdot m(X) \\ \frac{X^{n-k}m(X)}{g(X)} &= a(X) + \frac{b(X)}{g(X)} \\ c(X) &= a(X)g(X) + b(X) + X^{n-k}m(X) \end{aligned}$$

\rightarrow Beachte: Die Polynomdivision $b(X)/g(X)$ wird mit mod2 Addition gemacht.

- Parity-Check Polynom** $h(X)$: $h(X) = h_0 + h_1X + \dots + h_{n-k-1}X^{n-k-1}$
Es gilt:

$$g(X)h(X) = X^n + 1$$

11.1.3 Minimum distance considerations

Minimum distance eines Codes

- $d(c_1, c_2)$, Hamming Distanz (An wie vielen Stellen unterscheiden sich die Bits der zwei Codes c_1 und c_2)
- $w(c)$, Hamming-Gewicht (Anzahl 1-bits)
→ Es gilt: $d(c_1, c_2) = w(c_1 + c_2)$
- d_{\min} , **minimale Hamming Distanz eines linearen Blockcodes** (kleinste Hamming Distanz zwischen zwei beliebigen code vectors) bzw. (kleinstes Hamming-Gewicht der nonzero code vectors)

Error Detection/ Correction Gegeben ein $C(n, k, d_{\min})$ Block Code. Man hat:

- Error Detection falls ein error pattern mit Gewicht $t \leq d_{\min} - 1$ vorliegt.
- Error Correction falls ein error pattern mit Gewicht $t \leq \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ vorliegt.

$2^n - 2^k$ Fehler sind detectable, $2^k - 1$ Fehler sind undetectable.

Für gegebene minimum Distanz d_{\min} und Codeword Länge n , ein guter $C(n, k, d_{\min})$ Block Code hat eine grosse Anzahl möglicher Codewörter (2^k)

Hamming Bound Die Anzahl Codewörter eines binären Codes $C(n, k, d_{\min})$ muss die folgende Bedingung erfüllen:

$$2^k \left(1 + \binom{n}{1} + \dots + \binom{n}{t_0} \right) \leq 2^n, \quad \text{mit } t_0 = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$$

Anzahl receive vectors innerhalb eines decoding ball mit Radius t_0
→ Anzahl error vectors mit höchstens t_0 len

Falls ein Code die Bedingung mit Gleichheit erfüllt, so ist der Code ein **perfekter Code**.

Hamming Codes (n, k) linear block codes. Hamming codes sind single error correcting binary perfect codes.

$$\begin{array}{ll} \text{Blocklänge} & n = 2^m - 1 \\ \text{Anzahl message bits } k & k = 2^m - m - 1 \\ \text{Anzahl parity bits} & m = n - k, \quad m \geq 3 \end{array}$$

Beispiel für einen $(7,4)$ Hamming Code Generatormatrix:

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

Für Hamming Codes: $d_{\min} = 3$, unabhängig von der Anzahl Parity bits m .
→ $t_0 = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor = \left\lfloor \frac{3-1}{2} \right\rfloor = 1$

Somit sind Hamming Codes perfekte Codes:

$$2^k \left(1 + \binom{n}{1} \right) = 2^k(1+n) \Rightarrow \log_2(2^k(1+n)) = k + \log_2(1+n) = k + \log_2(2^m) \\ \Leftrightarrow 2^k \left(1 + \binom{n}{1} \right) = 2^{k+\log_2 2^m} = 2^{k+m} = 2^n$$

→ Die Bedingung für Perfect Codes ist somit erfüllt.

11.1.4 Durchsatz bei Linear Block Codes

Gegeben ein (n, k) linearer Blockcode.

Dann ist der erreichbare Durchsatz:

$$S = R \cdot P_{\text{korrekt}} \cdot \frac{k}{n}$$

Wobei:

- R : Übertragungsrate des Kanals in $\left[\frac{\text{Pakete}}{\text{sec}} \right]$ (1 Paket $\hat{=}$ n bit)
- P_{korrekt} = W'keit für korrekte Übertragung

11.2 Syndrome Decoding

Das Codeword wird über einen Kanal übertragen und das receive word ist $r = c + e \Leftrightarrow e = r + c$ (wegen mod2 Arithmetik)

11.2.1 Linear Block Codes

Das Syndrom ist definiert als

$$s = rH^T = (c + e)H^T = \underbrace{cH^T}_{=0} + eH^T = eH^T$$

→ s hängt nur von e ab → falls $s = 0 \Leftrightarrow e = 0$ (fehlerlose Übertragung)

Da $cH^T = 0$, alle möglichen 2^n error pattern, welche sich durch ein Codeword unterscheiden (2^k mögliche Codewörter), haben das selbe Syndrome. Somit ist die Anzahl Syndrome 2^{n-k} .

11.2.2 Systematic Cycling Codes

Der receive vector ist ein ein Polynom

$$r(X) = r_0 + r_1X + \dots + r_{n-1}X^{n-1}$$

Das Syndrom ist der Rest der Polynomdivision von $r(X)$ durch das Generatorpolynom $g(X)$, da wir $r(X)$ schreiben können als

$$r(X) = q(X)g(X) + s(X)$$

Bemerkung: Wir können schreiben $c(X) = a(X)g(X)$

Syndrome Berechnung:

Berechne die Polynomdivision $\frac{r(X)}{g(X)}$ (Achtung: mod2 Arithmetik). Der Rest der Polynomdivision ist dann das Syndrome $s(X)$.

11.2.3 Decoding

Konstruiere ein Standardarray indem man alle möglichen Codewörter in der ersten Reihe platziert, dann wählt man iterativ (wobei man beim minimum Hamming weight startet) coset-leaders mit einzigartigem Syndrome und füllt dann die Reihe auf indem man zur ersten Reihe addiert.
→ Coset-leader = «Fehler mit höchster Wahrscheinlichkeit»

$c_1 = 0$	c_2	c_3	c_4	c_5	c_6	c_7
e_2	$c_2 + e_2$	$c_3 + e_2$	$c_4 + e_2$	$c_5 + e_2$	$c_6 + e_2$	$c_7 + e_2$
e_3	$c_2 + e_3$	$c_3 + e_3$	$c_4 + e_3$	$c_5 + e_3$	$c_6 + e_3$	$c_7 + e_3$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
e_{2n-k}	$c_2 + e_{2n-k}$	$c_3 + e_{2n-k}$	$c_4 + e_{2n-k}$	$c_5 + e_{2n-k}$	$c_6 + e_{2n-k}$	$c_7 + e_{2n-k}$

Falls wir einen receive vector r' decodieren wollen, müssen wir zuerst das Syndrome berechnen: $s' = r'H^T$ und dann den entsprechenden coset leader dazu addieren: $\hat{c} = r' + e'$

Beispiel: Gegeben ein $(n, k, d_{\min}) = (6, 3, 3)$ linear Block code (kann 1-bit Fehler korrigieren)

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad H^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Receive word: $r = [1 \ 0 \ 1 \ 1 \ 0 \ 1]$, Syndrome: $s = [0 \ 1 \ 1] = rH^T$
Coset leader: $e_j = [0 \ 0 \ 0 \ 1 \ 0 \ 0]$, Estimate $c = r + e_j = [1 \ 0 \ 1 \ 0 \ 1 \ 0]$

Coset leader	2 ^k codewords
000000	011100 010110 110001 110110 101101 011011 000111
100000	111100 000110 010001 000110 001101 111011 000111
010000	001100 111010 100001 100110 111101 000111 010111
001000	010100 100010 111001 111110 100001 000011 001111
000100	011000 001110 110101 110010 100101 011111 000011
000010	011110 101000 110011 110100 101111 011001 000101
000001	011101 101011 110000 110111 101100 011010 000110
100100	111100 001110 010101 010010 001001 111111 100011

11.2.4 Fehlerwahrscheinlichkeit Block Codes

Maximum Likelihood Decoding

Fehlerwahrscheinlichkeit eines Binary Symmetric Channel, mit α_j der Anzahl an coset leader mit Hamming weight j

$$P_e = 1 - \sum_{j=0}^n \alpha_j p^j (1-p)^{n-j}$$

Die Wahrscheinlichkeit eines **undetectable errors**, mit ω_j der Anzahl codewords mit Hamming weight j

$$P_u = \sum_{j=1}^n \omega_j p^j (1-p)^{n-j}$$

11.3 Convolutional Codes

Free Hamming distance = minimum Hamming weight der Codewords

11.3.1 (n,k,K) encoder

(Alternative Notation für (n, k, K) ist (n, k) oder (n, k, m) mit $m = K - 1$)
Ein (n, k, m) encoder hat 2^{m-k} states, da die shift register $m \cdot k$ bits gross sind.

Encoder mit:

- n : Anzahl Generatoren (Outputs)
- k : Wortbreite (k-bit Worte)
- K : Anzahl shift register/stages (inkl. Input stage)

Die K Stages shiften k-bit Worte pro Zeiteinheit durch alle K Stages.

Memory depth des Convolutional encoder $m = K - 1$ (K : Anzahl Stages)

→ m = Anzahl k-bit Register

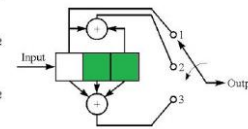
Die Coderate ist ungefähr k/n . Falls der Input Länge Nk hat, entspricht dies einem $(n(N+K-1), Nk)$ block code (am Schluss muss der Code noch komplett durchgeschliffen werden).

Beispiel: (3,1,3) encoder

$$g_1 = [100] = 1 + 0 \cdot D + 0 \cdot D^2 = 1$$

$$g_2 = [101] = 1 + 0 \cdot D + 1 \cdot D^2 = 1 + D^2$$

$$g_3 = [111] = 1 + 1 \cdot D + 1 \cdot D^2 = 1 + D + D^2$$



Der Input ist hier auch als «Kästchen» dargestellt, ist aber kein shift Register! Der state des Encoders wird durch die Inhalte der **grünen** shift register bestimmt.

Die **time domain generator matrix** dieses (3,1,3) encoders ist:

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \begin{array}{l} g_1 = [1 \ 0 \ 0] \\ g_2 = [1 \ 0 \ 1] \\ g_3 = [1 \ 1 \ 1] \end{array}$$

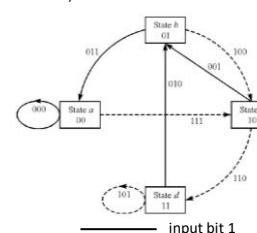
Ein (n, k, K) encoder benötigt $K - 1$ shift register (die **grünen** Blocks) welche vollständig durch ein state diagram (wobei die Transitions über die Input bits definiert sind) oder durch ein Trellis dargestellt werden können. Für einen Input der Länge N ergibt sich ein Pfad der Länge $N + K - 1$ im Trellis (Pfad kehrt immer zu all-zero zurück).

• State diagram:

Die Transitions sind mit den jeweiligen 3 coded **Output-Bits** bezeichnet.

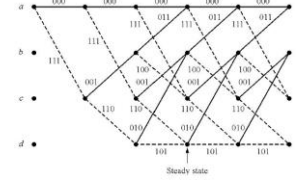
Für eine finite Anzahl an Input bits: **Füge tailing zeros am Ende des Inputs ein, damit wir im zero state enden.**

Es ist: input bit 0

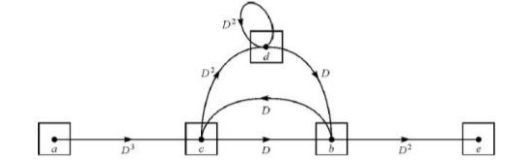


• Trellis:

Die Pfade im Trellis sind mit den Output bits gelabelt



Das distance spectrum kann mit dem state diagram bestimmt werden. Am Beispiel des (3,1,3) encoder: Der input state ist der all-zero state a . Der Loop bei a muss nicht beachtet werden (sind nur an Pfaden interessiert, die vom zero Pfad abweichen). Man geht dann durch das state diagram und zeichnet die Transitions ein. Die Transitions sind mit D^m labeled, wobei m das Hamming weight des assoziierten 3-bit Output code ist, aus der state diagram Transition.



Man erstellt eine Matrix mit Input state X_a und Output state X_e :

$$\begin{bmatrix} 0 & 0 & D & D & 0 \\ D^3 & D & 0 & 0 & 0 \\ 0 & 0 & D^2 & D^2 & 0 \\ 0 & D^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_a \\ X_b \\ X_c \\ X_d \\ X_e \end{bmatrix} = \begin{bmatrix} X_b \\ X_c \\ X_d \\ X_e \end{bmatrix}$$

Durch Gauss Elimination erhält man die Transfunktionsfunktion:

$$T(D) = \frac{X_e}{X_a} = \frac{D^6}{1 - 2D^2} = D^6 + 2D^8 + 4D^{10} + \dots$$

Man kann die Hamming distance vom distance spectrum diagram direkt auslesen: suche den kürzesten Pfad vom Startstate zum Endstate (im Beispiel von a zu e) → multipliziere alle D^m Terme → Exponent ist die Hamming distance.

Parity sequence Die Parity sequences berechnen sich zu:

$$v = u \cdot G = [v_0^{(0)} v_1^{(0)} \dots v_0^{(n-1)} v_1^{(n-1)} \dots v_1^{(n-1)} \dots]$$

Und dann die Parity Vektoren

$$\begin{array}{l} v^{(0)} = [v_0^{(0)} v_1^{(0)} v_2^{(0)} \dots] \\ v^{(1)} = [v_0^{(1)} v_1^{(1)} v_2^{(1)} \dots] \\ \vdots \end{array}$$

Transform-domain Generator Matrix G(D)

$$g(D) = g_0 + g_1 D + g_2 D^2 + \dots$$

$$G = \begin{bmatrix} g_1^{(0)}(D) & g_1^{(1)}(D) & \dots & g_1^{(n-1)}(D) \\ g_2^{(0)}(D) & g_2^{(1)}(D) & \dots & g_2^{(n-1)}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_k^{(0)}(D) & g_k^{(1)}(D) & \dots & g_k^{(n-1)}(D) \end{bmatrix}$$

Für einen (n, k, K) encoder.

Set of output sequences V(D)

$$V(D) = U(D) \cdot G(D) = [v^{(0)}(D) v^{(1)}(D) \dots v^{(k-1)}(D)]$$

wobei $G(D)$ die Transform-domain Generator Matrix ist und die Informations sequence $U(D) = [u^{(0)}(D) u^{(1)}(D) \dots u^{(k-1)}(D)]$

Das **respektive Codeword** ist dann:

$$v(D) = v^{(0)}(D^n) + D \cdot v^{(1)}(D^n) + D^2 \cdot v^{(2)}(D^n) + \dots + D^{n-1} v^{(k-1)}(D^n)$$

11.3.2 Fehlerwahrscheinlichkeit

Zwischen zwei Codewords mit Hamming distance d_h ist die Fehlerwahrscheinlichkeit

$$P_2(d_h) = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0} R_c d_h} \right) = Q \left(\sqrt{\frac{2E_b}{N_0} R_c d_h} \right)$$

(gegeben durch QPSK mapping). R_c : code rate

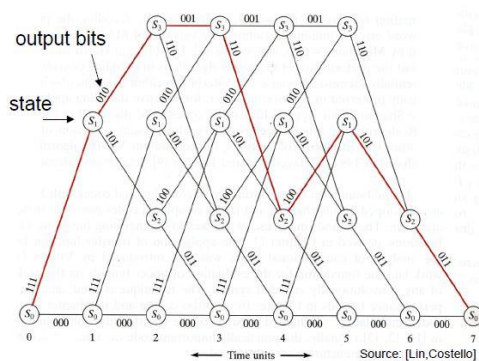
Die Energie pro coded bit ist $E_c = E_b \cdot R_c$

11.3.3 Viterbi Algorithmus

Wenn man das state diagram eines convolutional encoders in der Zeit expandiert, erhält man ein Trellis.

- Für eine Informationssequenz der Länge h und einen (n, k, m) Encoder hat ein Trellis $h + m + 1$ Zeitschritte.
- Trellis startet und endet in state S_0
- Upper leaving branch $\rightarrow u_i = 1$
- Lower leaving branch $\rightarrow u_i = 0$

Beispiel: Trellis für (3,1,2) encoder und Info.sequence $u = [1 \ 1 \ 0 \ 1]$



Vorgehen Viterbi Algorithmus:

- Berechne die Hamming distance jedes Transition Codes mit dem receive Code des jeweiligen Zeitschritts \rightarrow Branch metric
- Summiere die Branch metrics auf und wähle für jeden State den Survivor Pfad (Pfad mit kleinster Branch metric sum)
- Um den estimate receive vector zu erhalten, läuft man einfach den Pfad rückwärts

Branch Metrics Mit einem ML-decoder wollen wir die Branch metric $\sum_{i=0}^{m-1} M(r_i | v_i) = \sum_{i=0}^{m-1} \log(P(r_i | v_i))$, wobei v_i der codierte vector des i -ten branch ist. Da das Codeword nur mit allen Branches decodiert werden kann, summiert sich die Fehlerwahrscheinlichkeit auf.

- BSC: $P(r_{i,j} = 1 | v_{i,j} = 1) = P(r_{i,j} = 0 | v_{i,j} = 0) = 1 - p$
 $P(r_{i,j} = 1 | v_{i,j} = 0) = P(r_{i,j} = 0 | v_{i,j} = 1) = p$
- $\log(P(r_{i,j} = a | v_{i,j} = b)) \sim -(a - b)^2$

- Berechne branch metric für alle Pfade in States zur Zeit t eintreten
- Addiere jede metric zur branch metric des Survivors des respektiven state zur Zeit $t-1$
- Vergleiche & entscheide: Für jeden State, wähle den Pfad mit der grössten branch metric und eliminiere alle anderen Pfade

12 Multiple Access Protocol

Static Channel Allocation mit gegebenen slots im Zeit- oder Frequenzbereich.

- Frequency Division Multiplexing
- Time Division Multiplexing
- Code-Division Multiplexing: narrowband Signale werden in einem breiten Frequenzspektrum übertragen
- Space-division Multiplexing: Channels sind räumlich getrennt

Dynamic Channel Allocation zeitvariabler traffic der von verschiedenen Quellen ausgeht, mit den folgenden Annahmen:

- Station Model: N unabhängige Quellen auf dem gleichen Kanal.
- Mittlere Anzahl generierte Frames pro Zeiteinheit ist λ (konstant)
- Kollision: Falls zwei Kanäle gleichzeitig senden, überlappen sie sich in der Zeit, was zu einer Kollision führt. Beide müssen erneut senden.
- Time allocation, entweder in continuous time oder mit slotted time intervals.
- Carrier sense: Stationen hören den Kanal ab, bevor sie senden und können so detektieren, ob bereits ein Kanal sendet.

MAC Protocol medium access control protocol classification

- Random Access Method: kein koordinierter Kanalzugriff, Kollisionen können vorkommen.
- Deterministic Access Model: **keine Kollisionen** aufgrund einer zentralisierten oder dezentralisierten Kontrollereinheit.

Kollisions-freie Protokolle sind besser für high-load, weil der delay gross ist. Mit grösserer load verbessert sich die Kanaleffizienz anstatt eines generierten overhead mit den contention Protokollen (mit Kollisionen).

12.1 ALOHA

Unendliche Anzahl an unabhängigen Stationen greifen auf den gleichen Kanal zu. Kollisionen können vorkommen (werden vom Protokoll nicht ausgeschlossen). Kollidierte Pakete können erst nach einer zufällig gewählten Wartezeit erneut gesendet werden.

Der **arrival process** für ein neues Paket ist Poisson-verteilt mit **arrival rate λ** , die offered load besteht aus neuen und retransmittierten Paketen, was ein Poisson Prozess mit **channel access rate G** darstellt.

$$P[k|T] = \frac{(\lambda T)^k e^{-\lambda T}}{k!} \quad P[k|T] = \frac{(GT)^k e^{-GT}}{k!} = \frac{(G \cdot T/D)^k e^{-G \cdot T/D}}{k!}$$

Mit einer fixen Frame Länge (Frame Dauer) $D = \frac{\text{frame size [bit]}}{R_{\text{channel}}}$, $D: [s]$

Die folgenden Raten sind von Bedeutung:

- Station rate: $R_{\text{station}} = \frac{\text{\#bits outputted}}{\text{time frame}}$
- Channel rate: $R_{\text{channel}} = \dots$ (wird gegeben)
 \rightarrow wie viel kann der Kanal übertragen
- Utilisation rate: $R_{\text{util}} = S \cdot R_{\text{channel}}$, (S : Throughput)
 \rightarrow maximale effektive Datenrate im System

Die maximale Anzahl an Stationen folgt direkt:

$$\text{max. \# of stations } N_{\text{max}} = \frac{R_{\text{util}}}{R_{\text{station}}}$$

Mit $G = g \cdot D$ der durchschnittlichen Anzahl an generierten Frames pro frame Dauer.

- $G < 1$: unterloaded channel
- $G > 1$: overloaded channel

Die Wahrscheinlichkeit von k Kollisionen und einer erfolgreichen Übertragung ist: $P_k = (1 - P_0)^k \cdot P_0$

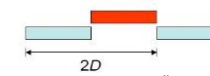
Die durchschnittliche Anzahl an Retransmissionen ist:

$$E[K] = \sum_{k=1}^{\infty} k \cdot P_k = \sum_{k=1}^{\infty} k \cdot (1 - P_0)^k \cdot P_0 = \frac{1 - P_0}{P_0}$$

12.1.1 Unslotted ALOHA

Stationen können jederzeit Pakete senden. Bei Kollisionen muss eine zufällige Zeit lang gewartet werden und dann erst kann erneut gesendet werden.

Kollisionen passieren, wenn mindestens ein neues Frame im Intervall $T = 2D$ (vulnerable period), relativ zum **roten Frame**, generiert wird:



Die Wahrscheinlichkeit einer erfolgreichen Übertragung (ein Versuch) folgt als die Wahrscheinlichkeit, dass während $T = 2D$ kein anderes Frame generiert wird ($k = 0$):

$$P[\text{"successful transmission"}] = P[k = 0|2D] = e^{-2gD} = e^{-2G} = P_0$$

Damit folgt die Kollisionswahrscheinlichkeit als:

$$P[\text{"collision"}] = 1 - P_0$$

Throughput (frames/ frame transmission) ist gegeben als

$$S = G \cdot P_0 = G \cdot e^{-2G}$$

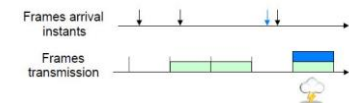
Bei $G = 1/2$ hat man den maximalen Throughput $S_{\text{max}} = 1/2e$

Der Throughput in [bits] ist die maximale Datenrate R_{util}

$$R_{\text{util}} = S \cdot R_{\text{channel}}$$

12.1.2 Slotted ALOHA

Wenn Stationen senden wollen, so können sie das Paket erst zu Beginn des nächsten time slots senden. Kollisionen passieren, falls zwei Frames im gleichen slot sind.



Damit wird die vulnerable period halbiert: $T = D$

Die Wahrscheinlichkeit einer erfolgreichen Übertragung (ein Versuch) folgt als die Wahrscheinlichkeit, dass während $T = D$ kein anderes Frame generiert wird ($k = 0$):

$$P[\text{"successful transmission"}] = P[k = 0|D] = e^{-gD} = e^{-G} = P_0$$

Damit folgt die Kollisionswahrscheinlichkeit als:

$$P[\text{"collision"}] = 1 - P_0$$

Falls die Anzahl an **idle slots β** (oft in %) gegeben ist, so ist die Wahrscheinlichkeit einer erfolgreichen Übertragung: $P_0 = \beta$

Throughput (frames/ frame transmission) ist gegeben als

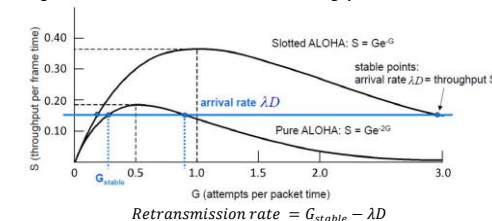
$$S = G \cdot P_0 = G \cdot e^{-G}$$

Bei $G = 1$ hat man den maximalen Throughput $S_{\text{max}} = 1/e$

Der Throughput in [bits] ist die maximale Datenrate R_{util}

$$R_{\text{util}} = S \cdot R_{\text{channel}}$$

Vergleich von ALOHA und slotted ALOHA Throughput



12.1.3 Slotted ALOHA mit endlicher Anzahl Stationen

Annahmen:

- Totale Anzahl an Stationen sei m
- Stationen buffern nur ein Frame
 - Entweder neues Frame oder Retransmission (backlogged station)
 - Eine backlogged station akzeptiert keine neuen Frames
- Jede non-backlogged station generiert ein neues Frame in nächsten slot mit Wahrscheinlichkeit $p_a = \lambda_{st}$
- Backlogged stations übertragen ihr Frame im nächsten time slot mit Wahrscheinlichkeit p_r
- Anzahl an backlogged stations: n
 \rightarrow Mittlere Anzahl an retransmittierten frames pro slot: $n \cdot p_r$
- Offered load: $G = n \cdot p_r + (m - n) \cdot p_a$

Wahrscheinlichkeit das i neue frames in einem slot gesendet werden:

$$P_a(i, n) = \binom{m-n}{i} (1 - p_a)^{m-n-i} p_a^i$$

Wahrscheinlichkeit das i Pakete, die schon kollidiert sind, in einem slot **retransmittiert** werden:

$$P_r(i, n) = \binom{n}{i} (1 - p_r)^{n-i} p_r^i$$

Durchschnittliche Anzahl an backlogged stations:

$$N = E[n] = \sum_{n=0}^m p_n \cdot n$$

Wobei p_n die steady state Wahrscheinlichkeiten sind (ermittelt mit Markov chain)

Mit Little's law ergibt sich die durchschnittliche Verzögerung bis zu einer erfolgreichen Übertragung: $\bar{T} = \frac{N}{\lambda} = \frac{N}{(m-n)p_a}$, $\bar{\lambda} = (m-n)p_a$

12.2 CSMA

Carrier Sense Multiple Access (z.B. Stationen hören den Kanal ab)

1-persistent Falls der Kanal idle ist, senden die Stationen mit Wahrscheinlichkeit 1.

- Warte bis der Kanal idle ist, dann sende ein Frame
- Falls eine Kollision auftritt, wartet die Station eine zufällige Zeit lang und beginnt dann von neuem.

Non-persistent Warte eine zufällige Zeit lang und prüfe dann, ob der Kanal idle ist, wiederhole den Algorithmus. Dadurch muss man denn Kanal nicht kontinuierlich abhören.

p-persistent (nur für slotted channels)

- Falls eine Station einen idle Kanal detektiert, sendet sie mit Wahrscheinlichkeit p , und mit Wahrscheinlichkeit $1 - p$ wartet die Station auf den nächsten Slot und wiederholt das Ganze.
- Falls die Station einen besetzten Kanal detektiert, wartet sie auf den nächsten slot und wiederholt den Algorithmus.
- Dies wird wiederholt bis das Frame übertragen wird, oder falls eine andere Station anfängt zu senden \rightarrow die erste Station verhält sich als ob eine Kollision passiert ist (wartet eine zufällige Zeit lang und versucht erneut)

Mit diesen Protokollen können Kollisionen nach wie vor vorkommen:

- Vulnerable period (signal propagation delay unter Stationen)

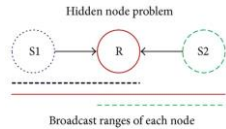
Für CSMA ist die vulnerable period

$$\tau = \frac{d}{v} = \frac{\text{Distanz zwischen Stationen}}{\text{Signal - Ausbreitungsgeschwindigkeit}}$$

\rightarrow Während dieser Zeit darf es nicht möglich sein, ein ganzes Frame zu übermitteln \rightarrow min. frame size = $\tau \cdot R_{\text{channel}}$

- Mehr als eine Station beginnt zu senden sobald die vorherige Transmission fertig ist (vor allem bei 1-persistent)

- Hidden node problem (nur für wireless transmission)



S1 und S2 sind beide in Reichweite von R, können sich untereinander aber nicht hören → carrier-sensing funktioniert nicht.

Throughput von non-persistent CSMA

Wir nehmen an:

- sehr viele Stationen ($m \rightarrow \infty$),
- Paket Transmissionen mit fixer Länge D**
- Kanalzugriff ist Poisson verteilt mit Parameter g (g ist die Rate, mit der Stationen versuchen auf den Kanal zuzugreifen)

Die normalized offered load ist: $G = g \cdot D$

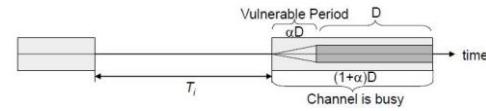
3 mögliche channel states: idle, successful transmission, collision

Die **vulnerable period** ist: $\tau = \alpha D$

→ Während dieser Periode können Kollisionen auftreten.

- Dauer eines channel access ohne Kollision: $D + \tau = (1 + \alpha)D$
→ im Falle einer Kollision erhöht sich die Dauer des channel access um αD (kann vernachlässigt werden)

T_i , die zufällige Zeit eine Station wartet, ist exponential verteilt $\sim \exp(g)$ mit Erwartungswert $T = 1/g$.



$$P(\text{success}) = e^{-\alpha G}$$

Mit der Rate an erfolgreichen Transmissionen:

$$S \approx \frac{P(\text{success})}{T + (1 + \alpha)D} = \frac{e^{-\alpha G}}{T + (1 + \alpha)D}, \quad S = sD = \frac{e^{-\alpha G}}{\frac{1}{G} + 1 + \alpha}$$

S ist der normalisierte Throughput pro Paketdauer D .

CSMA/ CD Collision Detection

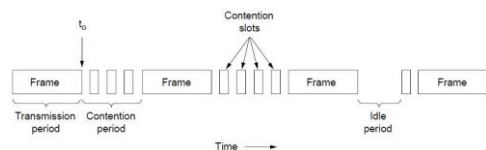
Bedingung Der Transmitter muss eine Kollision während seiner Übertragung erkennen können. Somit hat CSMA/ CD die doppelte vulnerable period wie CSMA: $2\tau = 2 \cdot \frac{v}{a} \Rightarrow \min. \text{frame size} = 2\tau \cdot R_{\text{channel}}$

Momentane Übertragungen zu detektieren reduziert die Kollisionswahrscheinlichkeit, aber Kollisionen können trotzdem auftreten.

Für lange Kommunikationsdistanz (zB Satelliten Kommunikations Links) ist diese Protokoll ineffizient, da eine lange Zeit vergeht, bis eine Kollision detektiert wird.

CSMA/ CD kann in einem von drei states sein:

contention, transmission oder idle

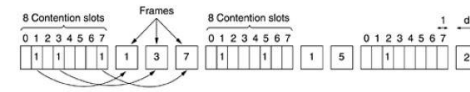


12.3 Collision-Free Protocols

Annahmen: Alle Stationen hören sich untereinander, es gibt **genau N Stationen**, jede Station hat eine einzigartige Adresse von $0, \dots, N - 1$.

12.3.1 Bit-map Protocol

- Jede Contention period hat genau N slots. Station j sendet genau 1 bit während slot j und signalisiert damit, dass sie übertragen will. Keine andere Station darf während diesem slot senden.
- Am Ende der contention Period weiss jede Station, wie viele und welche Stationen senden wollen.
- Frames werden in numerischer Reihenfolge übertragen (keine Kollisionen). Die Anzahl frames entspricht der Anzahl bits in den contention slots. Sobald alle frames übertragen worden sind, fängt das Ganze von neuem an.



12.3.2 Binary Countdown

Jede Station hat eine einzigartige Adresse, all mit derselben Länge.

	Bit
• Alle Stationen, die senden wollen, übertragen ihre binäre Adresse bit-weise	0 1 2 3
• Falls das momentane bit 1 ist, sendet die Station einen Burst aus, falls das Bit 0 ist, detektiert die Station den Kanal als busy.	0 0 1 0
• Falls eine Station den Kanal als busy detektiert, gibt die Station auf. Ansonsten sendet sie das nächste Bit der Adresse.	0 1 0 0
• Falls die Station den Kanal nicht abhörte (weil sie ein Burst sendet), sendet sie ebenfalls das nächste Bit ihrer Adresse.	1 0 0 1
	1 0 1 0
	Result 1 0 1 0

Der Gewinner ist die Station mit der höchsten Adresse (höchste binäre Zahl) → Stationen mit höhere Adresse haben grössere Priorität als Stationen mit tieferen Adressen. Indem wir Adressen dynamisch neu Allokieren, ist das ganze langfristig fair für alle Stationen.

12.3.3 Limited-contention

Falls wir eine tiefe load haben, so ist contention (pure oder slotted ALOHA) effizienter wegen dem kleineren delay. Sobald sich die load vergrößert, sind collision-free protocols effizienter, da die Kanaleffizienz besser ist (weil keine Kollisionen).

→ **Limited-contention protocols:** Protokolle, die contention bei low-load benutzen (wenig delay) und collision-free protocols bei high-load benutzen (für gute Kanaleffizienz)

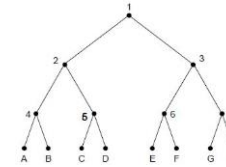
k Stationen wollen alle auf den Kanal zugreifen. Jede Station sendet während jedem time slot mit Wahrscheinlichkeit p .

Die Wahrscheinlichkeit, dass eine Station erfolgreich den Kanal bekommt, ist: $kp(1-p)^k$. Die optimale Wahrscheinlichkeit ist $p = 1/k$ und

$$P[\text{success optimal}] = \left(\frac{k-1}{k}\right)^{k-1}$$

12.3.4 Adaptive Tree Walk

Jeder Bit slot gehört zu einem Knoten im Baum. Die Blätter sind die Stationen (hier A-H).



- Im ersten Slot (nach einer erfolgreichen Übertragung) kann jede Station versuchen, den Kanal zu bekommen.
- Falls eine Kollision auftritt, so können nur Stationen, die unter Knoten 2 sind, versuchen den Kanal zu bekommen.
- Falls eine Station unter einem linken Knoten senden konnte, so ist der nächste Slot für Stationen unter dem rechten Knoten reserviert. (zB falls bei Knoten 2 eine Station senden kann, so dürfen im nächsten slot nur Stationen unter Knoten 3 senden)

12.4 Wired Protocols

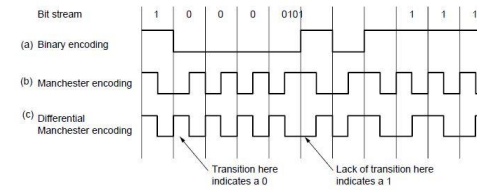
12.4.1 Manchester Coding

Wird verwendet in Ethernet (IEEE 802.3)

- Jede Bitdauer wird in zwei gleiche Intervalle aufgeteilt. Dann:
 - Bit 1: Spannung auf high im ersten Intervall und low im Zweiten.
 - Bit 0: Spannung auf low im ersten Intervall und high im Zweiten.

Differential Manchester Coding

Bei Differential Manchester Coding wird Bit 1 durch das Fehlen eines Spannungswechsels am Anfang eines Intervalls dargestellt. Bit 0 wird durch das Vorhandensein eines Spannungswechsels dargestellt.



12.4.2 Binary exponential backoff

Basiert auf CSMA/CD.

Drei channel states: idle, contention, successful transmission

- Nach einer Kollision wird die Zeit in diskrete Zeitslots unterteilt. Die **Länge eines slots ist die worst-case round-trip propagation time (2τ)**.

- Nach der ersten Kollision wartet jede Station entweder 0 oder 1 Slot bevor sie erneut versuchen zu senden.
- Nach jeder weiteren Kollision wird das backoff window verdoppelt (bis zu maximal 1024 Zeitslots)
- Grundsätzlich wird nach i Kollisionen eine zufällige Zahl zwischen 0 und $2^i - 1$ gewählt. Diese Anzahl an Slots wird dann abgewartet.
- Während Versuch j gibt es 2^{j-1} slots

Beispiel:

Zwei CSMA/CD Stationen, die jeweils auf einen Kanal zugreifen wollen. Es wird binary exponential backoff verwendet.

Während Versuch j gibt es 2^{j-1} slots → $P[\text{collision}] = 2^{-(j-1)}$

Die W'keit, dass die ersten $k - 1$ Versuche scheitern und im k ten Versuch keine Kollision auftritt, ist:

$$P_k = (1 - 2^{-(k-1)}) \prod_{j=1}^{k-1} 2^{-(j-1)} = (1 - 2^{-(k-1)}) 2^{-(k-1)(k-2)/2}$$

Dann: $E[K] = \sum k P_k$

13 Mathematischer Appendix

13.1 EM

$$c = \lambda \cdot f, \quad c \approx 300 \cdot 10^6 \frac{\text{m}}{\text{s}}$$

13.2 Definitionen

sinc-Funktion

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}, \quad \text{sinc}(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad \forall n \in \mathbb{Z}$$

Amplitude bei $x = 0$: 1
 Nullstellen bei: $x, n \in \mathbb{Z} \setminus \{0\}$
 (π gehört nicht in die Achsenbeschriftung!)
 Im Fall von $x = T f \Rightarrow \text{NS bei } \pm \frac{1}{T}, \pm \frac{2}{T}, \dots$

Rechteckfunktion per SigSys I

$$\square_T(t) = \begin{cases} 1, & |t| \leq T \\ 0, & \text{sonst} \end{cases}$$

Rechteckfunktion per KommSys

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} 1, & |t| \leq \frac{T}{2} \\ 0, & \text{sonst} \end{cases}$$

13.3 Euler Relationen

$$\sin(2\pi f t) = \frac{1}{2j} (e^{j2\pi f t} - e^{-j2\pi f t})$$

$$\cos(2\pi f t) = \frac{1}{2} (e^{j2\pi f t} + e^{-j2\pi f t})$$

$$T \text{sinc}(fT) e^{-j\pi f t} = \frac{1}{2\pi j f} (1 - e^{-j2\pi f t})$$

13.4 Additionstheoreme

$$\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \cos \alpha \sin \beta$$

$$\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta$$

$$\tan(\alpha \pm \beta) = \frac{\tan \alpha \pm \tan \beta}{1 \mp \tan \alpha \tan \beta}$$

$$\sin(\alpha) + \sin(\beta) = 2 \sin\left(\frac{\alpha + \beta}{2}\right) \cos\left(\frac{\alpha - \beta}{2}\right)$$

$$\cos(\alpha) + \cos(\beta) = 2 \cos\left(\frac{\alpha + \beta}{2}\right) \cos\left(\frac{\alpha - \beta}{2}\right)$$

$$\sin(\alpha) \sin(\beta) = \frac{1}{2} (\cos(\alpha - \beta) - \cos(\alpha + \beta))$$

$$\cos(\alpha) \cos(\beta) = \frac{1}{2} (\cos(\alpha - \beta) + \cos(\alpha + \beta))$$

$$\sin(\alpha) \cos(\beta) = \frac{1}{2} (\sin(\alpha - \beta) + \sin(\alpha + \beta))$$

$$\cos(2x) = \cos^2(x) - \sin^2(x) = 2 \cos^2(x) - 1$$

$$\sin(2x) = 2 \sin x \cos x$$

$$2 \cos^2 x = 1 + \cos(2x)$$

$$2 \sin^2 x = 1 - \cos(2x)$$

13.5 Fourier-Tabelle

Auszug aus der SigSys I Formelsammlung:

$$x(t) = \int_{-\infty}^{\infty} \hat{x}(f) e^{j2\pi f t} df \quad \hat{x}(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt$$

$g(at)$	$\circ \bullet$	$\frac{1}{ a } G\left(\frac{f}{a}\right)$
$\frac{d}{dt} g(t)$	$\circ \bullet$	$j2\pi f G(f)$
$\int_{-\infty}^t g(\tau) d\tau$	$\circ \bullet$	$\frac{1}{j2\pi f} G(f) + \frac{G(0)}{2} \delta(f)$
$\text{rect}\left(\frac{t}{T}\right)$	$\circ \bullet$	$T \text{sinc}(fT)$
$\text{sinc}(2Wt)$	$\circ \bullet$	$\frac{1}{2W} \text{rect}\left(\frac{f}{2W}\right)$
$\exp(-at)\sigma(t), \quad a > 0$	$\circ \bullet$	$\frac{1}{a + j2\pi f}$
$\exp(-a t), \quad a > 0$	$\circ \bullet$	$\frac{2a}{a^2 + (2\pi f)^2}$
$\exp(-\pi t^2)$	$\circ \bullet$	$\exp(-\pi f^2)$
$\begin{cases} 1 - \frac{ t }{T}, & T < T \\ 0, & t \geq T \end{cases}$	$\circ \bullet$	$T \text{sinc}^2(fT)$
$\delta(t)$	$\circ \bullet$	1
1	$\circ \bullet$	$\delta(f)$
$\delta(t - t_0)$	$\circ \bullet$	$\exp(-j2\pi f t_0)$
$\exp(j2\pi f_c t)$	$\circ \bullet$	$\delta(f - f_c)$
$\cos(2\pi f_c t)$	$\circ \bullet$	$\frac{1}{2} (\delta(f - f_c) + \delta(f + f_c))$
$\sin(2\pi f_c t)$	$\circ \bullet$	$\frac{1}{2j} (\delta(f - f_c) - \delta(f + f_c))$
$\text{sign}(t)$	$\circ \bullet$	$\frac{1}{j\pi f}$
$\frac{1}{\pi t}$	$\circ \bullet$	$-j \text{sign}(f)$
$\sigma(t)$	$\circ \bullet$	$\frac{1}{2} \delta(f) + \frac{1}{j2\pi f}$
$\sum_{k=-\infty}^{\infty} \delta(t - kT_0)$	$\circ \bullet$	$\frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_0}\right)$

13.6 Hilbert Transformation

$\cos(2\pi f_c t)$	$\sin(2\pi f_c t)$
$\sin(2\pi f_c t)$	$-\cos(2\pi f_c t)$
$m(t) \cos(2\pi f_c t)$	$m(t) \sin(2\pi f_c t)$
$m(t) \sin(2\pi f_c t)$	$-m(t) \cos(2\pi f_c t)$
$\frac{\sin(t)}{t}$	$\frac{1 - \cos(t)}{t}$
$\text{rect}(t)$	$-\frac{1}{\pi} \log\left \frac{t-1/2}{t+1/2}\right $
$\delta(t)$	$\frac{1}{\pi t}$
$\frac{1}{1+t^2}$	$\frac{t}{1+t^4}$
$\frac{1}{t}$	$-\pi \delta(t)$

13.7 Ableitungen

$(f \cdot g)' = f' \cdot g + f \cdot g'$	$(f/g)' = (f'g - fg')/g^2$
$(a^x)' = \ln(a) \cdot a^x$	$(\ln(x))' = 1/x$
$(f(g(x)))' = f'(g(x)) \cdot g'(x)$	

13.8 Integrale

$$\int_0^{2\pi} \sin^2(nx) \cdot dx = \int_0^{2\pi} \cos^2(nx) \cdot dx = \pi, \quad \forall n \neq 0$$

$$\int_0^{\pi} \sin^2(nx) \cdot dx = \int_0^{\pi} \cos^2(nx) \cdot dx = \frac{\pi}{2}, \quad \forall n \neq 0$$

$$\int \sin^2(ax) \cdot dx = \frac{x}{2} - \frac{1}{4a} \sin(2ax)$$

$$\int \cos^2(ax) \cdot dx = \frac{x}{2} + \frac{1}{4a} \sin(2ax)$$

$$\int x \cdot \sin(ax) dx = \frac{\sin(ax)}{a^2} - \frac{x \cdot \cos(ax)}{a}$$

$$\int x \cdot \cos(ax) dx = \frac{\cos(ax)}{a^2} + \frac{x \cdot \sin(ax)}{a}$$

$$\int \cos(ax) \cos(bx) dx = \frac{\sin((a-b)x)}{2(a-b)} + \frac{\sin((a+b)x)}{2(a+b)}, \quad (a \neq b)$$

$$\int \sin(ax) \sin(bx) dx = \frac{\sin((a-b)x)}{2(a-b)} - \frac{\sin((a+b)x)}{2(a+b)}, \quad (a \neq b)$$

$$\int \sin(ax) \cos(ax) dx = \frac{1}{2a} \sin^2(ax)$$

$$\int \sin(ax) \cos(bx) dx = -\frac{\cos((a-b)x)}{2(a-b)} - \frac{\cos((a+b)x)}{2(a+b)}, \quad (a \neq b)$$

Für $f_c = \frac{1}{T_b}$:

$$\int_0^{T_b} \sin^2(2\pi f_c t) \cdot dt = \int_0^{T_b} \cos^2(2\pi f_c t) \cdot dt = \frac{T_b}{2} = \frac{1}{2f_c}$$

Partielle Integration

$$\int_a^b f'(x) \cdot g(x) dx = [f(x) \cdot g(x)]_a^b - \int_a^b f(x) \cdot g'(x) dx$$

Orthogonalitätsrelationen

- cos, wobei die Grenzen um 2π auseinanderliegen und beliebig verschiebbar sind (z.B. $[0, 2\pi]$)

$$\int_{-\pi}^{\pi} \cos(kx) \cos(lx) dx = \begin{cases} 2\pi, & k = l = 0 \\ \pi, & k = l \neq 0 \\ 0, & k \neq l \neq 0 \end{cases}$$

- sin, wobei die Grenzen um 2π auseinanderliegen und beliebig verschiebbar sind (z.B. $[0, 2\pi]$)

$$\int_{-\pi}^{\pi} \sin(kx) \sin(lx) dx = \begin{cases} 2\pi, & k = l = 0 \\ \pi, & k = l \neq 0 \\ 0, & k \neq l \neq 0 \end{cases}$$

- Mischterm:

$$\int_{-\pi}^{\pi} \sin(kx) \cos(lx) dx = 0, \quad \forall k, l$$

Besselfunktion 0ter Ordnung

$$I_0(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp(z \cdot \cos(\theta)) d\theta$$

13.9 Fehlerwahrscheinlichkeiten

paarweise Fehlerwahrscheinlichkeit:

Es habe das AWGN die spektrale Leistungsdichte $\frac{N_0}{2}$.

Sei A_{ik} das Ereignis, dass der Beobachtungsvektor \mathbf{x} näher am Signalpunkt \mathbf{s}_k ist als bei \mathbf{s}_i .

Die Distanz zwischen den beiden Signalpunkten ist durch $d_{ik} = \|\mathbf{s}_i - \mathbf{s}_k\|$ definiert.

Dann ist

$$P[A_{ik}] = \frac{1}{2} \text{erfc}\left(\frac{d_{ik}}{2\sqrt{N_0}}\right)$$

13.10 Faltung

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau = \int_{-\infty}^{\infty} f(t - \tau) g(\tau) d\tau$$

Siebeigenschaft

$$\int_{-\infty}^{\infty} f(\tau) \delta(\tau - a) d\tau = \int_{-\infty}^{\infty} f(\tau) \delta(a - \tau) d\tau = f(a)$$

$$\Rightarrow f(t) * \delta(t - t_0) = f(t - t_0)$$

13.11 Summenformeln

Geometrische Reihe

$$\sum_{k=0}^{\infty} a_0 q^k = \frac{a_0}{1 - q}, \quad |q| < 1$$

n-te Partial Summe

$$s_n = \sum_{k=0}^n a_0 q^k = \begin{cases} a_0 \frac{1 - q^{n+1}}{1 - q}, & q \neq 1 \\ a_0(n+1), & q = 1 \end{cases}$$

Diverse

$$\sum_{k=1}^{\infty} k \cdot q^k = \frac{q}{(1 - q)^2}, \quad |q| < 1$$

13.12 dB

Für einheitenlose Grössen, z.B. Verhältnisse wie das SNR:

$$[SNR]_{dB} = 10 \cdot \log_{10}(SNR)$$

Für Amplituden Grössen (Signale, Spannung, Strom,...):

$$[A]_{dB} = 20 \cdot \log_{10}(A)$$

Grössen mit Einheiten brauchen einen Bezugswert, zB 1mW:

$$P_{dB} = 10 \cdot \log_{10}\left(\frac{P}{1\text{mW}}\right)$$

Absolut	Relativ	Absolut	Relativ
1	0dB	2	$3.01 \approx 3\text{dB}$
10	10dB	1/2	-3dB
100	20dB	4	6dB
1/10	-10dB	$10/2 = 5$	$(10 - 3) = 7\text{dB}$
10^{-12}	-120dB	$10/(2 \cdot 2) = 2.5$	$(10 - 3 - 3) = 4\text{dB}$
0	$-\infty\text{dB}$		

Umrechnung: $[P]_{dB} = 10 \cdot \log_{10}(P) \Leftrightarrow P = 10^{P_{dB}/10}$

Definitionen

- 1mW := 0dBm
- 1W := 0dBW

Zur Erinnerung:

- $\log(x \cdot y) = \log(x) + \log(y)$
- $a^{x+y} = a^x \cdot a^y$
- $(a^x)^y = a^{x \cdot y}$
- $\log(x^y) = y \cdot \log(x)$
- $\log_a(x) = \log_b(x) / \log_b(a)$

13.13 Diverses

Jensen's Ungleichung

Falls $f(\cdot)$ eine konkave Funktion (zB log) ist, so gilt:

$$E[f(X)] \leq f(E[X])$$

Oder äquivalent:

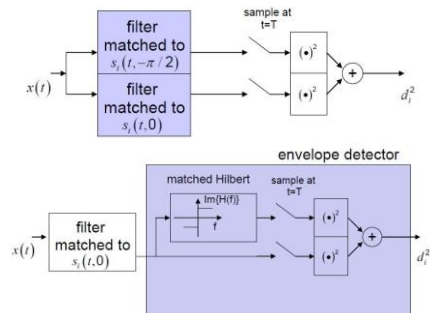
$$\sum_{k=0}^{K-1} p_k \cdot f(x_k) \leq f\left(\sum_{k=0}^{K-1} p_k \cdot x_k\right)$$

14 Appendix

14.1 Diagramme

Matched Filter Decoder für noncoherent BFSK

Anstatt den zwei-Branch Korrelatorbänken kann man auch zwei matched Filter brauchen.

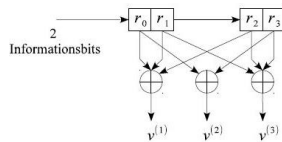


14.2 Beispiele

Convolutional encoder mit Wortbreite 2:

Falls die Wortbreite (wie viele Bits pro Shift übertragen werden) grösser als eins ist, so reiht man in den Generatoren dies einfach aneinander:

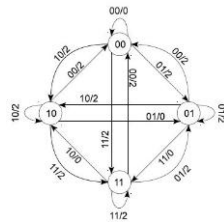
$$g = [r_0 \ r_1 \ r_2 \ r_3]$$



$$g^{(1)} = [1 \ 1 \ 1 \ 0], g^{(2)} = [1 \ 0 \ 0 \ 1], g^{(3)} = [0 \ 1 \ 1 \ 1]$$

Das dazugehörige state diagram ist:

Wobei auf den Transitions der state (Inhalt der Register $r_2 r_3$) und das Hamming weight der assoziierten Output bits steht (kann mittles Tabelle ermittelt werden)



Free Hamming distance:

Suche den Weg mit dem kleinsten summierten Hamming weight (von Transitions), der von state 00 zurück zu state 00 führt und mindestens einen anderen state (als 00) besucht.

Die aufaddierten Hamming weights ergeben die Free Hamming Distance.