BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECRICAL ENGINEERING AND COMMUNICATION
Department of Radio Electronics

Advanced Telecommunication Systems

A. PROKES

Content

Hilbert transform (HT) of a function *s*(*t*): $\hat{s}(t) = s(t) * \frac{1}{\pi t} = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau$ (1.1)

Fourier transform of HT: $\hat{S}(f) = -j\,\mathrm{sgn}(f)\,S(f)$ (1.2)

Hilbert transform increase the phase of the negative frequencies by $\pi/2$ and to decrease the phase of the positive frequencies by $\pi/2$.

Spectrum of a function using HT *s*(*t*): $S(f) = j\,\mathrm{sgn}(f)\,\hat{S}(f)$

Function *s*(*t*) in time domain using HT : $s(t) = \hat{s}(t) * \frac{-1}{\pi t} = -\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{\hat{s}(\tau)}{t-\tau} d\tau$

Properties of the Hilbert transform: $\left|\hat{S}(f)\right| = |S(f)|, \quad \hat{\hat{s}}(t) = -s(t).$

A signal is bandpass (narrowband) if its Fourier transform satisfies

$$S(f) = 0 \quad \text{for} \quad |f| \leq f_c - B/2\,, \quad |f| \geq f_c + B/2$$

Where the signal bandwidth is given by $B \leq 2f_c$

The signal $s(t)$ is real if:

$$s(t) = s^*(t) \quad \text{and} \quad S^*(-f) = S(f) \qquad (1.3)$$

Complex analytic signal

The real bandpass signal $s(t)$ **is unique determined by** $S(f)$ for $f > 0$:

$$\boxed{S_+(f) = \begin{cases} 2S(f) & \text{for} \quad f \geq 0 \\ 0 & \text{for} \quad f < 0 \end{cases} \quad \Rightarrow \quad S_+(f) = S(f) + j\hat{S}(f)}$$

Inverse Fourier transform of $S_+(f)$ **is then:** $s_+(t) = s(t) + j\hat{s}(t)$

Conversely:

$$s(t) = \mathrm{Re}\{s_+(t)\} \qquad (1.4)$$

$$S(f) = \frac{1}{2}[S_+(f) + S_+^*(-f)]$$

Note that the Fourier transform pairs $x(t) \leftrightarrow X(\omega)$ satisfy the "shift" rule:

$x(t)e^{j\omega_c t} \leftrightarrow X(\omega - \omega_c)$

Complex baseband representation, Complex envelope

For known carrier frequency f_c the $s_+(t)$ can be translated to baseband:

$$\tilde{S}(f) = S_+(f + f_C), \quad \tilde{s}(t) = s_+(t)e^{-j2\pi f_c t} \Rightarrow s_+(t) = \tilde{s}(t)e^{j2\pi f_c t} \qquad (1.5)$$

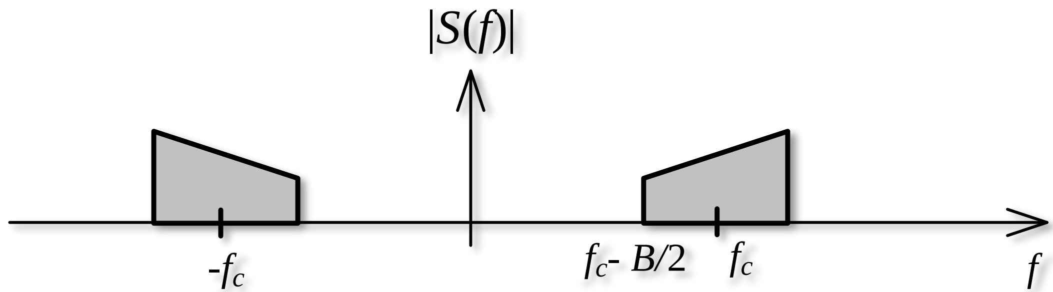
By substituting (4) into (5) the signal $s_+(t)$ can be expressed as:

$$\boxed{s(t) = \mathrm{Re}\{\tilde{s}(t)e^{j2\pi f_c t}\}, \quad S(f) = \frac{1}{2}[\tilde{S}(f - f_c) + \tilde{S}^*(-f - f_c)]} \qquad (1.6)$$

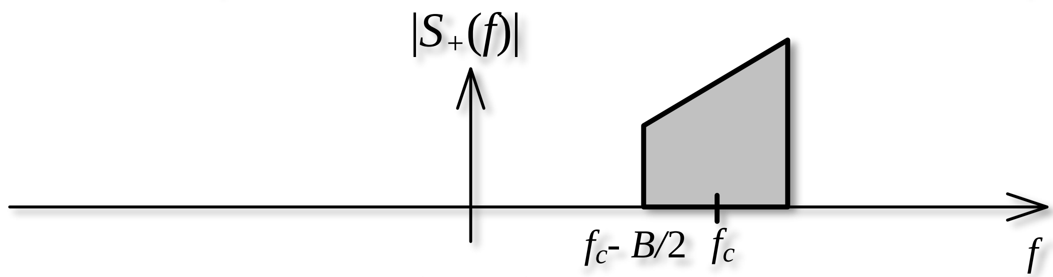
1. Representations of Communication Signals

Complex baseband representationc

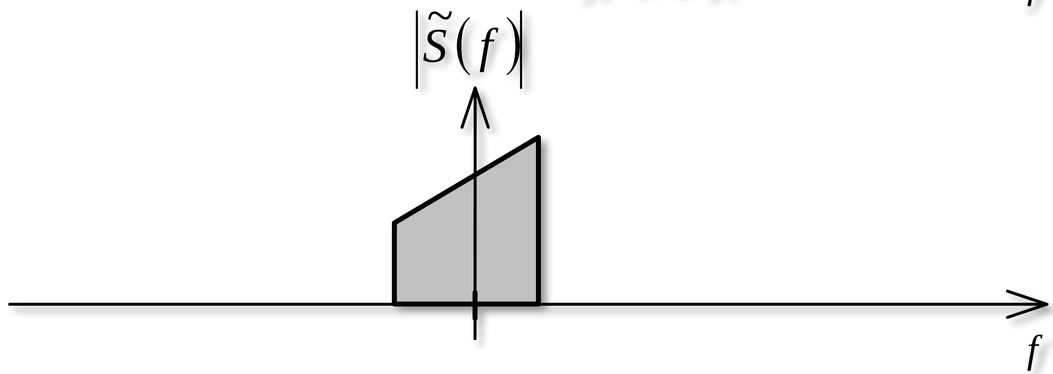
Bandpass signal:



Analytical signal:



Complex envelope:



Decomposition of the complex envelope into real and imaginary parts:

$$\tilde{s}(t) = s_I(t) + j s_Q(t)$$

$$cos(x) = \tfrac{1}{2}e^{jx} + \tfrac{1}{2}e^{-jx}$$

$$sin(x) = \tfrac{1}{2j}e^{jx} - \tfrac{1}{2j}e^{-jx}$$

Then:

$$s(t) = \mathrm{Re}\left\{\tilde{s}(t)e^{j2\pi f_c t}\right\} = s_I(t)\cos(2\pi f_c t) - s_Q(t)\sin(2\pi f_c t)$$

$$\boxed{S(f) = \frac{1}{2}[S_I(f - f_C) + S_I(f + f_C)] + \frac{j}{2}[S_Q(f - f_C) + S_Q(f + f_C)]}$$

The real and imaginary parts can be expressed as:

$$s_I(t) = \frac{1}{2}[\tilde{s}(t) + \tilde{s}^*(t)], \quad S_I(f) = \frac{1}{2}[\tilde{S}(f) + \tilde{S}^*(-f)]$$

$$s_Q(t) = \frac{1}{2}[\tilde{s}(t) - \tilde{s}^*(t)], \quad S_Q(f) = \frac{1}{2}[\tilde{S}(f) - \tilde{S}^*(-f)]$$

By applying the spectrum of a bandpass signal given by (1.6) and frequency shift

$$S(f+f_c)=\frac{1}{2}\left[\tilde{S}(f)+\tilde{S}^*(-f-2f_c)\right],$$
$$S(f-f_c)=\frac{1}{2}\left[\tilde{S}(f-2f_c)+\tilde{S}^*(-f)\right],$$

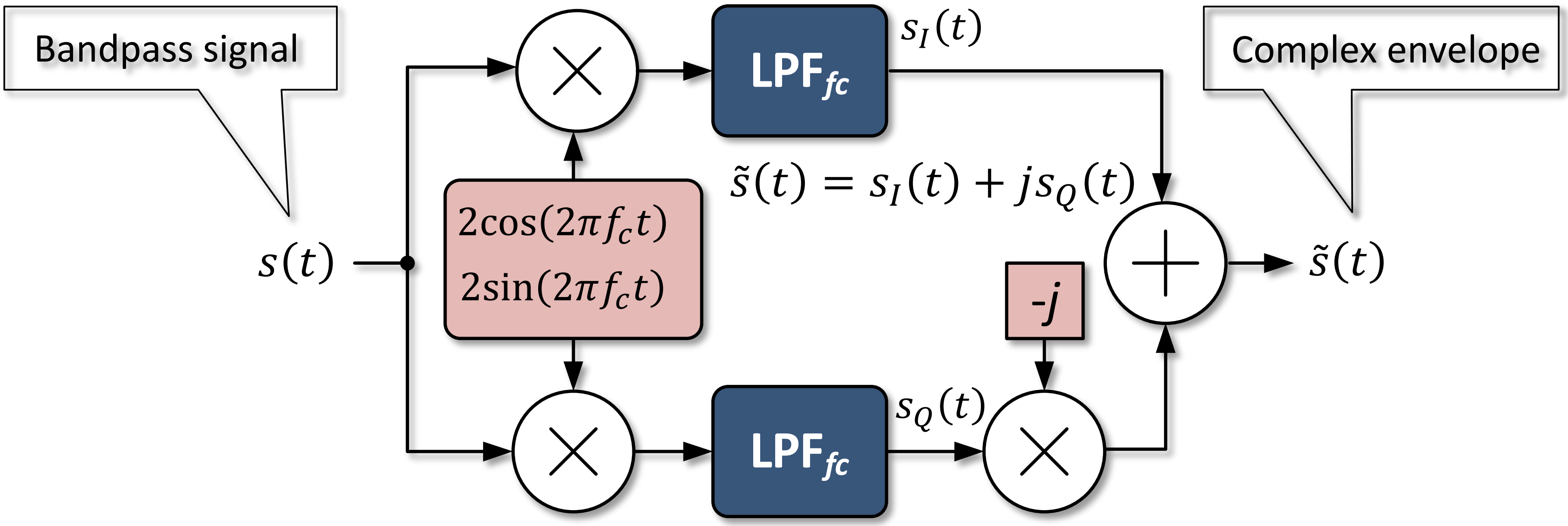
we can get:

$$S_I(f)=\mathrm{LPF}_{f_c}[S(f-f_c)+S(f+f_c)],$$
$$S_Q(f)=j\mathrm{LPF}_{f_c}[S(f-f_c)-S(f+f_c)], \tag{1.7}$$

Where LPFf_c **represents ideal lowpass filtering with cut-off frequency at** f_c**.**

By conversion (7) into the time domain, we have

$$s_I(t) = \mathrm{LPF}_{f_c}[2s(t)\cos(2\pi f_c t)], \quad s_Q(t) = -\mathrm{LPF}_{f_c}[2s(t)\sin(2\pi f_c t)]$$



A set of signals is represented by a set of vectors with respect to an orthonormal basis. Suitable for *M*-ary communications.

Functions $f_1(t)$ **and** $f_2(t)$ **are orthogonal if:**

$$\int_{-\infty}^{\infty} f_1(t) f_2(t)\, dt = 0 \qquad (1.8)$$

The norm of a function $f(t)$:

$$\|f\| = \sqrt{\int_{-\infty}^{\infty} f^2(t)\, dt} \qquad (1.9)$$

Functions $f_1(t)$ **and** $f_2(t)$ **are orthonormal if they meet** (8) **and** $\|f_1\| = \|f_2\| = 1$

For a *M*-ary signal set $\{s_i(t)\}_{i=1}^{M}$ **we need to obtain an orthonormal basis (a set of orthonormal signals)** $\{\varphi_j(t)\}_{j=1}^{N}$, **where** $N \leq M$, **which span the space formed by linear combinations of the *M* signals. Then we can represent** $s_i(t)$ **simply by the *N*-dimensional vector** $\mathbf{s}_i = \left[s_{i,1}, s_{i,2}, \dots, s_{i,N}\right]^T$.

Example 1.1 ***(M=2),*** **antipodal signals**: $s_1(t) = p(t), \quad s_2(t) = -p(t)$.

Suppose the basic pulse shape $p(t)$ **has unit norm, i.e.** $\|p(t)\| = 1$, **we can choose it to be the basis of signal set** $\varphi(t) = p(t)$, **then** $\boldsymbol{s}_1 = [1], \quad \boldsymbol{s}_2 = [-1]$.

Example 1.2 ***(M=4)*** **QPSK, considering the symbol interval** $\langle 0; T)$, **we can write**:

$$s_1(t) = \sqrt{2P}\cos(2\pi f_c t + \pi/4)\, p_T(t)$$
$$s_2(t) = \sqrt{2P}\cos(2\pi f_c t + 3\pi/4)\, p_T(t)$$
$$s_3(t) = \sqrt{2P}\cos(2\pi f_c t + 5\pi/4)\, p_T(t)$$
$$s_4(t) = \sqrt{2P}\cos(2\pi f_c t + 7\pi/4)\, p_T(t)$$

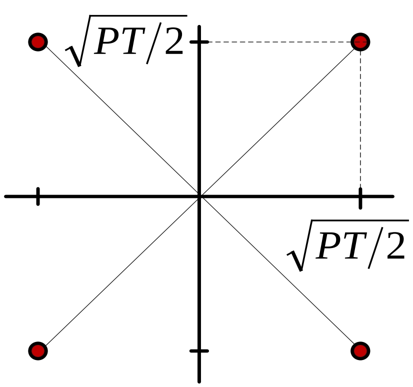
where $pT(t)$ **is the pulse shape function.**

Assuming that $T \gg f_c$ holds for the symbol period (orthogonality condition) a simple basis for this signal can be obtained in the form:

$$\varphi_1(t) = \sqrt{\frac{2}{T}}\cos(2\pi f_c t)\, p_T(t)$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T}}\sin(2\pi f_c t)\, p_T(t) \tag{1.10}$$

The corresponding signal vectors then are:



$$s_1 = \left[\sqrt{PT/2},\quad \sqrt{PT/2}\right]^T,$$
$$s_2 = \left[-\sqrt{PT/2},\quad \sqrt{PT/2}\right]^T,$$
$$s_3 = \left[-\sqrt{PT/2},\quad -\sqrt{PT/2}\right]^T,$$
$$s_4 = \left[\sqrt{PT/2},\quad -\sqrt{PT/2}\right]^T,$$

Example 1.3 ***(M=4)*** **QPSK, considering the symbol interval ⟨0; *T*) and using the complex baseband representation we can write:**

$$\tilde{s}_1(t) = \sqrt{2P}\mathrm{e}^{j\pi/4} p_T(t),$$
$$\tilde{s}_2(t) = \sqrt{2P}\mathrm{e}^{j3\pi/4} p_T(t),$$
$$\tilde{s}_3(t) = \sqrt{2P}\mathrm{e}^{j5\pi/4} p_T(t),$$
$$\tilde{s}_4(t) = \sqrt{2P}\mathrm{e}^{j7\pi/4} p_T(t).$$

A basis for this set of complex envelopes is:

$$\varphi_1(t) = \sqrt{\frac{1}{T}} p_T(t)$$

The corresponding signal vectors then have only one coordinate:

$$\boldsymbol{s}_1 = \left[\sqrt{PT} + j\sqrt{PT}\right],$$
$$\boldsymbol{s}_2 = \left[-\sqrt{PT} + j\sqrt{PT}\right],$$
$$\boldsymbol{s}_3 = \left[-\sqrt{PT} - j\sqrt{PT}\right],$$
$$\boldsymbol{s}_4 = \left[\sqrt{PT} - j\sqrt{PT}\right],$$

Gram-Schmidt procedure

Universal method of finding the orthonormal base $\{\varphi_j(t)\}_{j=1}^{N}$ of a set of finite energy signals $\{s_i(t)\}_{i=1}^{M}, \quad M \geq N$.

1. **Choose** $\varphi_1(t) = s_1(t)/\|s_1\|$.

2. **For *j* >1 calculate** $v_j(t) = s_j(t) - (s_j, \varphi_1)\varphi_1(t) - \cdots - (s_j, \varphi_{j-1})\varphi_{j-1}(t)$,

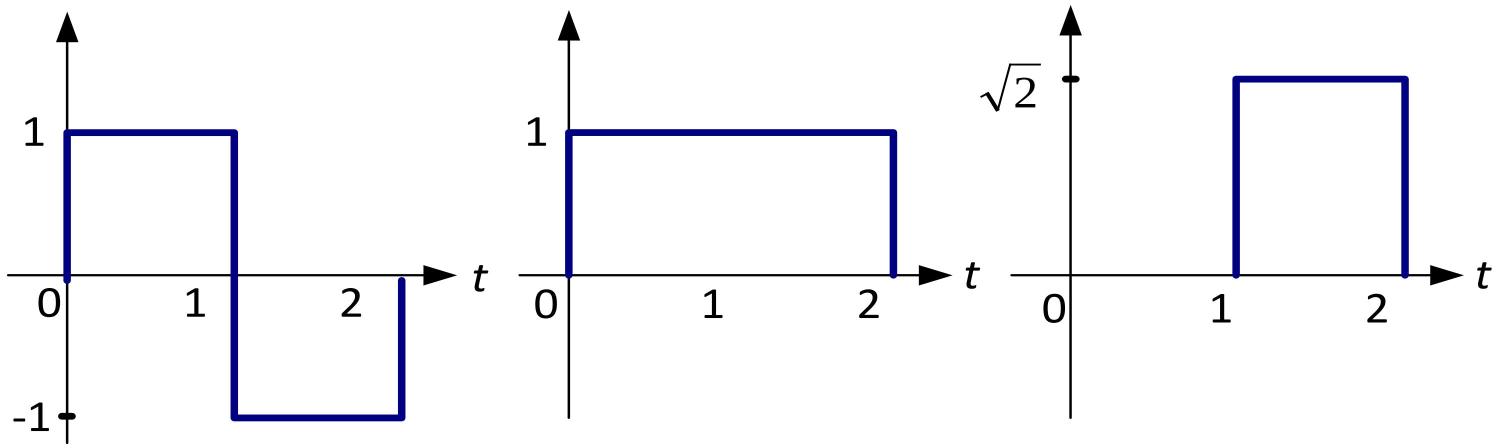
   where $(s_j, \varphi_k) = \int_{-\infty}^{\infty} s_j(t)\varphi_k^*(t)\, dt$.

   Then choose $\varphi_j(t) = v_j(t)/\|v_j\|$.

3. **Continue until all *M* functions are expressed in terms of $\varphi_j(t)$.**

   Note that all the $\{s_i(t)\}_{i=1}^{M}$ signals must be independent. In other case the norm of the signal $v_j(t)$ will be zero for some *j* and we can skip that signal and move to the next one.

Example 4:* let us consider the set of** M=3 ***signals:



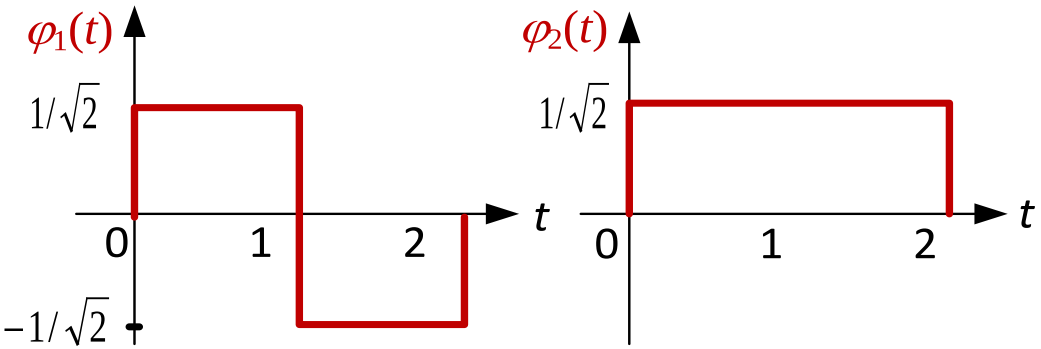
1. $\varphi_1(t) = s_1(t)/\|s_1\| = s_1(t)/\sqrt{2}$.

2. $v_2(t) = s_2(t) - \underbrace{\int_0^2 s_2(t)\varphi_1(t)dt}_{0} \cdot \varphi_1(t) = s_2(t)$,

$\varphi_2(t) = v_2(t)/\|v_2\| = s_2(t)/\sqrt{2}$.

$$v_3(t) = \underbrace{s_3(t) - \underbrace{\int_0^2 s_3(t)\varphi_1(t)dt}_{-1} \cdot \varphi_1(t) - \underbrace{\int_0^2 s_3(t)\varphi_2(t)dt}_{1} \cdot \varphi_2(t)}_{s_3(t)} = 0,$$

$$\varphi_3(t) = 0.$$



The vector representation for the signals then are:

$$s_1 = \left[\sqrt{2}, 0\right]^T, \quad s_2 = \left[0, \sqrt{2}\right]^T \quad \text{a} \quad s_3 = [-1,1]^T,$$

Orthogonal representation

A set of finite energy signals $\{s_i(t)\}_{i=1}^{M}$ can be expressed by

$$s_i(t) = \sum_{j=1}^{N} s_{i,j}\varphi_j(t)\,. \quad (1.11)$$

With the knowledge of the basis $\{\varphi_j(t)\}_{j=1}^{N}$, we can represent $s_i(t)$ by the *N*-dimensional vector

$$\mathbf{s}_i = \left[s_{i,1}, s_{i,2}, \dots, s_{i,N}\right]^T$$

where

$$s_{i,j} = \left(s_i, \varphi_j\right) = \int_{-\infty}^{\infty} s_i(t)\varphi_j^*(t)\,dt \quad (1.12)$$

Each pair of orthonormal base functions satisfies the equation

$$(\varphi_i, \varphi_j) = \int_{-\infty}^{\infty} \varphi_i(t)\varphi_j^*(t)\,dt = \delta_{ij}.$$

The inner product of two signal $s_i(t)$ and $s_j(t)$ **is equal to the inner product of the vectors** $\boldsymbol{s}_i$ **and** $\boldsymbol{s}_j$

$$(s_i, s_j) = \int_{-\infty}^{\infty} s_i(t)s_j^*(t)\,dt =$$

$$\int_{-\infty}^{\infty} \left[\sum_{k=1}^{N} s_{i,k}\varphi_k(t)\right]\left[\sum_{l=1}^{N} s_{i,l}\varphi_l(t)\right]^* dt = \sum_{k=1}^{N} s_{i,k}s_{j,k}^* = \mathbf{s}_i \cdot \mathbf{s}_j\,.$$

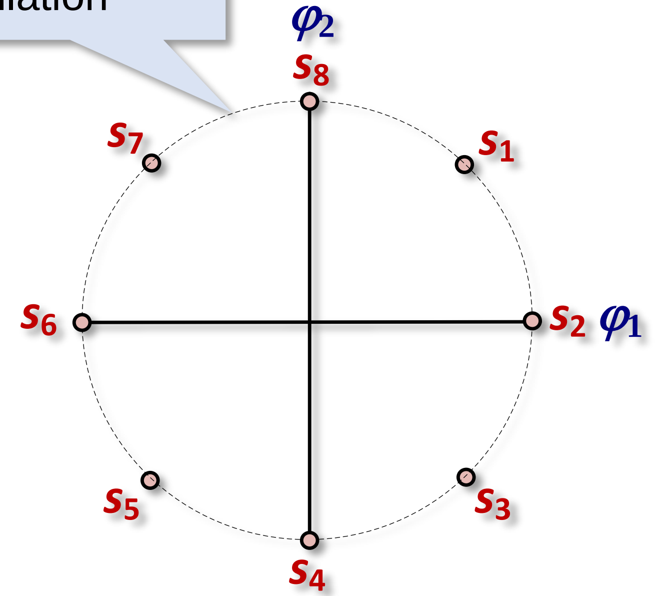
For $i = j$ **we get** $$\|s_i\| = \sqrt{\int_{-\infty}^{\infty} s_i(t)s_i^*(t)\,dt} = \|\mathbf{s}_i\|$$

M-ary communication **allows to send** $\log_2 M$ **bits at a time using one of** M **possible signals.**

MPSK modulation: **consider the symbol interval** $\langle 0; T)$. **For the amplitude** A, **carrier frequency** f_c, **and** $i = 1, 2, \ldots, M$ **we can write**

$$s(t) = A\cos\left(2\pi f_c t + \frac{2i\pi}{M}\right) p_T(t)$$

Example of 8PSK constellation



The basis functions are the same as in the case of QPSK modulation (10). **With this basis** $s_i(t)$ **can be expressed by the 2-D vector**

$$\boldsymbol{s}_i = \left[A\sqrt{T/2}\cos\left(\frac{2i\pi}{M}\right), \quad A\sqrt{T/2}\sin\left(\frac{2i\pi}{M}\right)\right]^T$$

1. Representations of Communication Signals

M-ary communications

Pulse amplitude modulation (PAM): **consider PAM with four possible symbols**

$$s(t) = \pm A\cos(2\pi f_c t) p_T(t) \text{ and } \pm 3A\cos(2\pi f_c t) p_T(t)$$

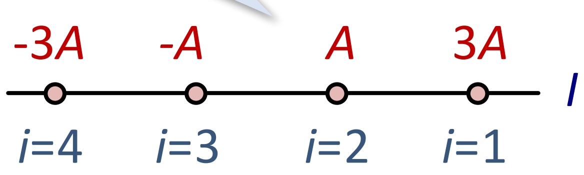
The basis of corresponding 1-D signal space consists of only one element

$$s(t) = \sqrt{2/T}\cos(2\pi f_c t) p_T(t),$$

and $s_i(t)$ can be expressed by the 1-D vector

$$\boldsymbol{s}_i = \left[A\sqrt{T/2}\,[3-2(i-1)]\right]$$

Example of 4PAM constellation



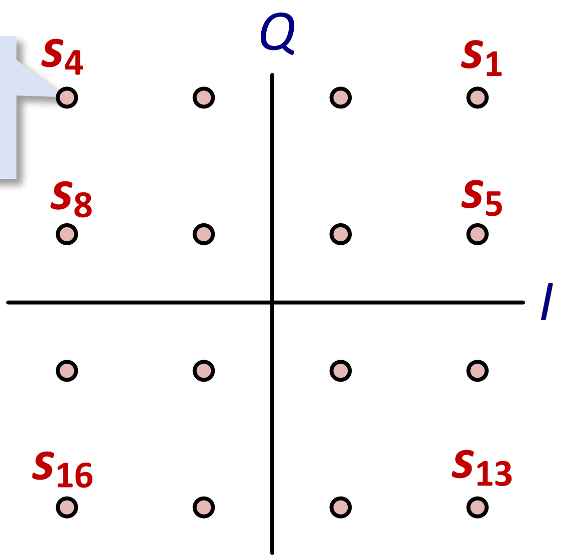
Quadrature amplitude modulation (QAM): **In QAM one data stream is modulated with PAM onto the in-phase (*I*) carrier and another data stream onto the quadrature (*Q*) carrier.**

The signal space is then 2-D. The basis functions are again the same as in (1.10),

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cos(2\pi f_c t)\, p_T(t)$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T}} \sin(2\pi f_c t)\, p_T(t)$$

Example of 16QAM constellation



and $s_i(t)$ **can be expressed by the 2-D vector.**

In the 16QAM case $s_i(t)$ **can be expressed as**

$$\boldsymbol{s}_i = \left[A\sqrt{T/2}\,[3-2\times \mathrm{mod}_4\,(i-1)],\, A\sqrt{T/2}\,[3-2\times(\mathrm{ceil}(i/4)-1)]\right]$$

A set of signals $\{s_i(t)\}$ is an orthogonal signal set over symbol interval $\langle 0; T)$ if

$$\int_0^T s_i(t)s_j^*(t)\,dt = 0 \;\text{ for all } i \neq j.$$

An orthonormal set is an orthogonal set normalized in energy, i.e.

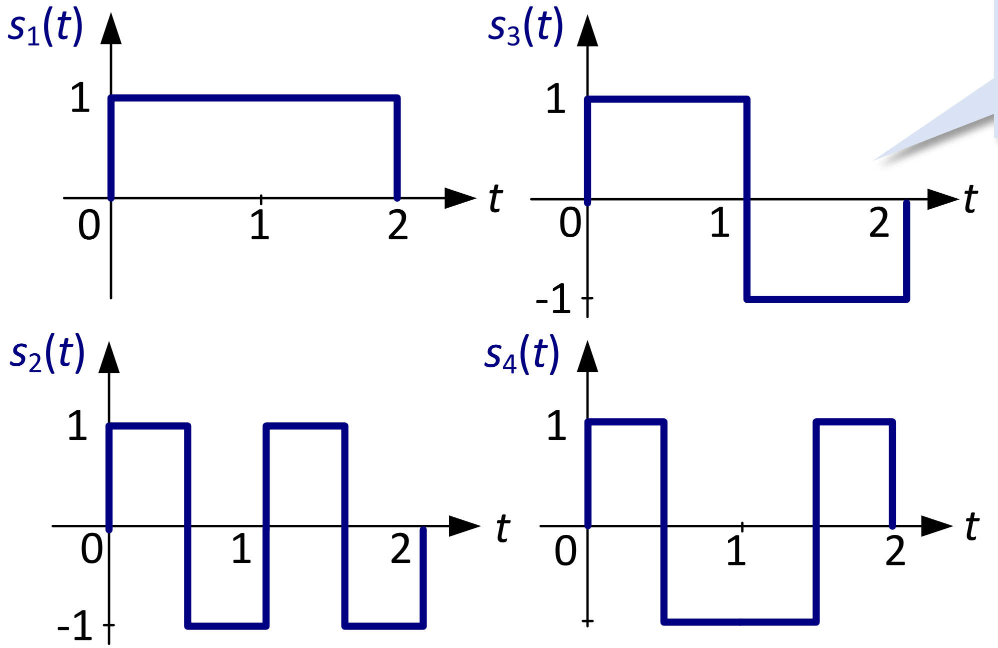
$$\varphi_j(t) = s_j(t)/\left\|s_j\right\|.$$

One possible way to generate an orthogonal set of signals is by using the *Hadamard matrices* defined by

$$H_0 = [1] \quad \textbf{and} \quad H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \quad \textbf{i.e.} \quad H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_2 = \begin{array}{c} \begin{array}{cccc} \multicolumn{2}{c}{H_1} & & \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{array}$$

Each row can be used to generate a signal.



Example of orthogonal signals generated by H_2

Let we have an orthogonal signal set $\{s_1(t), s_2(t), \dots, s_M(t)\}$, the corresponding ***biorthogonal set*** **is then $\{s_1(t), -s_1(t), s_2(t), -s_2(t), \dots, s_M(t), -s_M(t)\}$ i.e. all the sign-reversed signals are added to the set.**

Let the transmitter sends a signal chosen from the set of M finite-energy signals $s_1(t), s_2(t), \cdots, s_M(t)$, and the channel with transfer $\widehat{K} = |K|e^{j\theta}$, where $|K|$ and θ are the channel amplitude and phase response, is contaminated by AWGN $n(t)$ with noise power spectral density N_0. Hence, the received signal is

$$r(t) = \widehat{K} s_m(t) + n(t), \tag{1.13}$$

for $m \in \{0,1,\cdots,M-1\}$. By employing the Gram-Schmidt procedure, we can construct a set of N orthonormal functions $\{\varphi_n(t)\}_{n=1}^{N}$, which spans the signal space formed by $\{s_m(t)\}_{m=1}^{M}$. Then we can rewrite (1.13) as

$$\mathbf{r} = \widehat{K}\mathbf{s_m} + \mathbf{n}, \tag{1.14}$$

where

$$\mathbf{r} = [r_1, r_2, \cdots, r_N]^T, \quad \mathbf{n} = [n_1, n_2, \cdots, n_N]^T,$$

(1.15)

and

$$\mathbf{s}_m = [s_{m1}, s_{m2}, \cdots, s_{mN}]^T \quad \text{for} \quad m = 0, 1, \cdots, M-1.$$

The vector components of (1.14) **are given by** (1.12)

$$r_k = \int_{-\infty}^{\infty} r(t)\varphi_k^*(t)\,dt, \quad n_k = \int_{-\infty}^{\infty} n(t)\varphi_k^*(t)\,dt, \tag{1.16}$$

and

$$s_{mk} = (s_m, \varphi_k) = \int_{-\infty}^{\infty} s_m(t)\varphi_k^*(t)\,dt \quad \textbf{for} \quad m = 0, 1, \cdots, M-1.$$

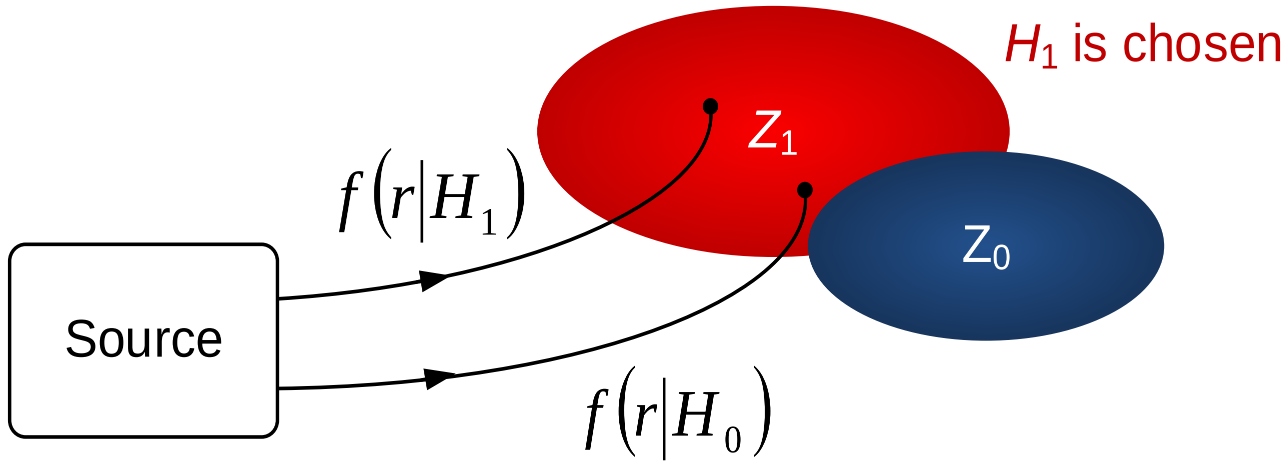
Finally, the transmitted signals are

$$s_m(t) = \sum_{k=1}^{N} s_{m,k}\varphi_k(t)\,. \tag{1.17}$$

Similar relations we can write for $r(t)$ and $n(t)$. Note that in some cases the orthonormal function set is augmented by $\{\varphi_n(t)\}_{n=N+1}^{\infty}$ to form an *orthonormal basis* for every finite-energy signal $s(t)$,

Let the signal $s(t)$ propagating through a transmission channel be disturbed by the noise $n(t)$. The receivedsignal $r(t) = s(t) + n(t)$ is then understood as a realization of a random process R with values r.

At any time the receiver must decide whether the symbol 0 or the symbol 1 has been transmitted (whether the hypothesis H_0 or H_1 holds), while maximizing the probability of correct decision.



Z_0 and Z_1: decision regions

In the decision process there are four possible outcomes:

	Decision	Truth	Notation	Cost	Designation
1	H_0	H_0	(D_0, H_0)	C_{00}	-
2	H_0	H_1	(D_0, H_1)	C_{01}	*miss*
3	H_1	H_0	(D_1, H_0)	C_{10}	*false alarm*
4	H_1	H_1	(D_1, H_1)	C_{11}	*detection*

How to get the best possible detection? We will design a detector that minimizes the expected cost of a decision:

1. **A large cost are assigned to all undesirable conditions** $C_{10} > C_{00}$, $C_{01} > C_{11}$
2. **The average loss (Bayes risk)** $R = E(C)$ **is determined and minimized**.

$$R = C_{00}P(D_0, H_0) + C_{01}P(D_0, H_1) + C_{10}P(D_1, H_0) + C_{11}P(D_1, H_1). \quad (2.1)$$

Bayes' rule: $$P(D_i, H_j) = P\left(D_i \middle| H_j\right) P(H_j) \quad (2.2)$$

where $$P\left(D_i \middle| H_j\right) = \int_{Z_i} f\left(r \middle| H_j\right) dr, \quad i,j = 0,1 \quad (2.3)$$

is a conditional probability: the likelihood of event A occurring given that B is true.

By substution (2.2) **and** (2.3) **into** (2.1) **and by using the formula**

$$\int_{Z_1} f\left(r \middle| H_j\right) dr = 1 - \int_{Z_0} f\left(r \middle| H_j\right) dr, \quad i,j = 0,1$$

we get

$$\boxed{\begin{aligned} R &= C_{10}P(H_0) + C_{11}P(H_1) \\ &+ \int_{Z_0} \left\{\left[P(H_1)(C_{01} + C_{11}) f\left(r \middle| H_1\right)\right] - \left[P(H_0)(C_{10} + C_{00}) f\left(r \middle| H_0\right)\right]\right\} dr. \end{aligned}}$$

To minimize the Bayes risk, it is necessary to select a region Z_0 that guarantees the validity of the relationship:

$$\left[P(H_1)(C_{01}+C_{11})f\left(r\middle|H_1\right)\right] < \left[P(H_0)(C_{10}+C_{00})f\left(r\middle|H_0\right)\right].$$

The validity of a hypothesis is then determined by inequality:

$$\boxed{\frac{f\left(r\middle|H_1\right)}{f\left(r\middle|H_0\right)} \mathop{\gtrless}_{H_0}^{H_1} \frac{P(H_0)(C_{10}-C_{00})}{P(H_1)(C_{01}-C_{11})}} \tag{17}$$

where $\Lambda(r)=\dfrac{f\left(r\middle|H_1\right)}{f\left(r\middle|H_0\right)}$ is a likelihood ratio and $\eta=\dfrac{P(H_0)(C_{10}-C_{00})}{P(H_1)(C_{01}-C_{11})}$ is a threshold.

Commonly we use: $$\ln\Lambda(r) \underset{H_0}{\overset{H_1}{\gtrless}} \ln\eta$$

Instead of one realization, a few of ones can be used: $\mathbf{R} = [R_1, R_2, \cdots, R_k]$.

<u>Example 2.1</u> **(binary signal)**

$$H_1:\ R = m + N,$$
$$H_0:\ R = N$$

where m **is the signal amplitude for log 1 and** N **is the noise with standard deviation** σ.

$$\left.\begin{aligned} f(r|H_1) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(r-m)^2}{2\sigma^2}\right) \\ f(r|H_0) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{r^2}{2\sigma^2}\right) \end{aligned}\right\} \Rightarrow \Lambda(y) = \exp\left(-\frac{m^2-2rm}{2\sigma^2}\right)$$

$$\ln \Lambda(r) = -\frac{m^2}{2\sigma^2} - \frac{rm}{\sigma^2} \underset{H_0}{\overset{H_1}{\gtrless}} \ln \eta \quad \Rightarrow \quad r \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\sigma^2}{m} \ln \eta + \frac{m}{2} = \gamma$$

Bayes criteria: **the hypothesis is chosen using** (17) **where**

$$\eta = \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})}$$

ML (Maximum Likelihood) criteria: **the hypothesis is chosen using** (17) **where:**

$$\eta = 1 \quad \Rightarrow \quad \underbrace{P(H_0)}_{0.5} \underbrace{(C_{10} - C_{00})}_{1} = \underbrace{P(H_1)}_{0.5} \underbrace{(C_{01} - C_{11})}_{1}$$

Example 2.2 **(binary signal)**

$C_{00} = C_{11} = 0, \quad C_{10} = C_{01} = 1. \quad P(H_1) = 0.45, \; P(H_0) = 0.55$

$$r \underset{H_0}{\overset{H_1}{\gtrless}} \frac{\sigma^2}{m} \ln\left[\frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})}\right] + \frac{m}{2}$$

Bayes criteria

$$r \underset{H_0}{\overset{H_1}{\gtrless}} \ln\left[\frac{0.45}{0.55}\right] + \frac{1}{2} \Rightarrow r \underset{H_0}{\overset{H_1}{\gtrless}} \underbrace{-0.2 + \frac{1}{2}}_{0.3}$$

ML criteria

$$r \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1}{2}$$

M-ary hypotheses testing

Generally, the signal can be represented by *M* states. Then it is necessary to decide on the validity of *M* hypotheses: $H_0, H_1, \cdots, H_M$ they offer M^2 possible decission.

The Bayes criterion requires the cost of C_{ij} to be assigned to each combination of the decision D_i and the hypothesis H_j for $i,j = 0,1,\ldots M\text{-}1$.

Bayes risk is then:

$$R = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(H_j) C_{ij} \underbrace{P(D_i, H_j)}_{\text{relation (16)}}$$

The next step is to minimize *R* and to determine the conditions for the selection of individual hypotheses (similar to the binary signal)

Presumption: **the receiver has decided in favor of a true hypothesis, but a certain signal parameter is unknown.**

Objective: **to estimate the unknown parameter *f* from the finite number of signal samples.**

Given: k **implementations** $R_1, R_2, \cdots, R_k$ **of the random variable *R* corresponding to the received signal and their samples** $r_1, r_2, \cdots, r_k$.

If *f* is random, the *Bayesian estimation* is used, if *f* is non-random, the *ML Maximum Likelihood (ML) estimation is used*.

ML estimation: **if the conditional function** $f(\mathbf{r}|\varphi)$ **of the density of the random variable *R* depends on the parameter *f*, the likelihood function will be**

$$L(\varphi) = f(\mathbf{r}|\varphi) = f(r_1, r_2, \cdots, r_K|\varphi) = \prod_{k=1}^{K} f(r_k|\varphi)$$

… and the parameter estimation

$$\hat{\varphi}_{ML} = \arg \max_{\varphi} f(\mathbf{r}|\varphi) \tag{2.5}$$

For the $f(\mathbf{r}|\varphi)$ maximalization the standard procedure

$$\frac{\partial}{\partial \varphi} f(\mathbf{r}|\varphi) = 0$$

can be used. In many cases, it is preferable to use

$$\frac{\partial}{\partial \varphi} \ln f\,(\mathbf{r}|\varphi) = 0$$

It holds

$$\hat{\varphi}_{ML} = \arg \max_{\varphi} \ln f(\mathbf{r}|\varphi) = \arg \max_{\varphi} f(\mathbf{r}|\varphi)$$

Example 2.3

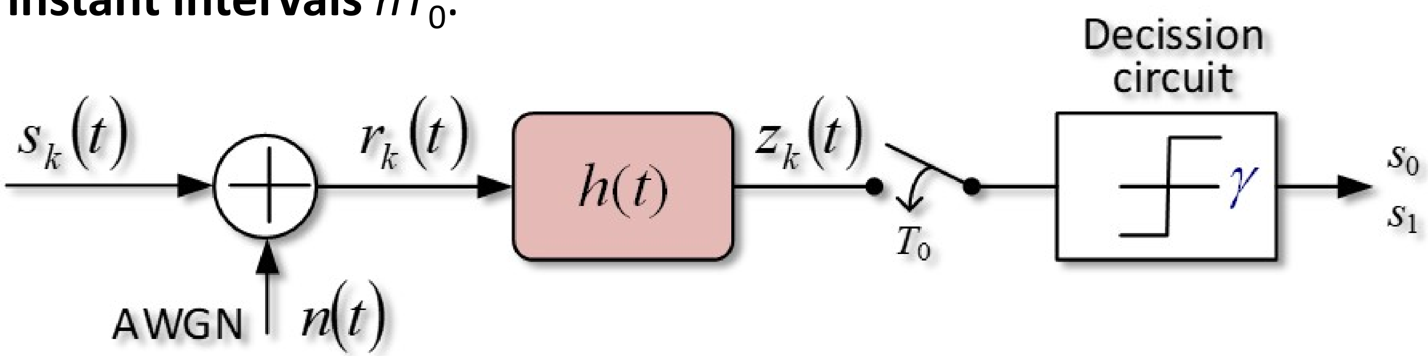
$$H_1:\ R_k = m + N_k, \qquad k = 1,2,\ldots, K$$
$$H_0:\ R_k = N_k \qquad k = 1,2,\ldots, K$$
$$m = ?$$

$$f(\mathbf{r}|m) = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(r_k - m)^2}{2\sigma^2}\right) = \frac{1}{(2\pi)^{K/2}\sigma^K} \exp\left(-\sum_{k=1}^{K} \frac{(r_k - m)^2}{2\sigma^2}\right)$$

$$\ln f\,(\mathbf{r}|m) = \ln\left[\frac{1}{(2\pi)^{K/2}\sigma^K}\right] + \sum_{k=1}^{K} \frac{(r_k - m)^2}{2\sigma^2}$$

$$\frac{\partial \ln f\,(\mathbf{r}|m)}{\partial m} = \sum_{k=1}^{K} \frac{r_k}{\sigma^2} - \frac{Km}{\sigma^2} = \frac{K}{\sigma^2}\left(\frac{1}{K}\sum_{k=1}^{K} r_k - m\right) = 0 \quad \Rightarrow \quad m = \frac{1}{K}\sum_{k=1}^{K} r_k$$

- **Let we have a binary signal:** $s_0(t)$, $s_1(t)$ $\Rightarrow H_0, H_1$
- **Then let we add AWGN:** $n(t)$ **with a double-sided power spectral density** $N_0/2$
- **The response** $z_k(t)$ **of an LTI filter with the impulse response** $h(t)$ **is sampled at the instant intervals** nT_0.



$$z_k(t) = \underbrace{s_k(t) * h(t)}_{\hat{s}_k(t)} + \underbrace{n(t) * h(t)}_{\hat{n}(t)}, \quad k = 0,1$$

Let it holds:

$$E[\hat{n}(T_0)] = 0, \qquad E[\hat{n}^2(T_0)] = R_{\hat{n}}(0), \quad \Rightarrow \quad E[\hat{z}_0(T_0)] = \hat{s}_0(T_0), \qquad E\left[\hat{z}_0{}^2(T_0)\right] = R_{\hat{n}}(0)$$

$R_{\hat{n}}$: noise autocorrelation function

Conditional probability of false reception:

$$P(e|H_1) = P[z_1(T_1) < \gamma] = \Phi\left(\frac{\gamma - \hat{s}_1(T_0)}{\sqrt{R_{\hat{n}}(0)}}\right) = \quad Q\left(\frac{\hat{s}_1(T_0) - \gamma}{\sqrt{R_{\hat{n}}(0)}}\right), \qquad (2.6)$$

$$P(e|H_0) = P[z_0(T_0) \geq \gamma] = Q\left(\frac{\gamma - \hat{s}_0(T_0)}{\sqrt{R_{\hat{n}}(0)}}\right) \qquad (2.7)$$

Total error probability: $P(e) = P(H_0)P(e|H_0) + P(H_1)P(e|H_1)$

$P(H_i)$: probability that the hypothesis H_i is valid

The threshold setting

$$P(e) = P(H_0)Q\left(\frac{\hat{s}_0(T_0) - \gamma}{\sqrt{R_{\hat{n}}(0)}}\right) + P(H_1)Q\left(\frac{\gamma - \hat{s}_1(T_0)}{\sqrt{R_{\hat{n}}(0)}}\right)$$

$$\frac{dP(e)}{d\gamma} = 0 \quad \Rightarrow \quad \gamma = \frac{\hat{s}_0(T_0) + \hat{s}_1(T_0)}{2} + \frac{R_{\hat{n}}(0)}{\hat{s}_1(T_0) - \hat{s}_0(T_0)} \ln \frac{P(H_0)}{P(H_1)}$$

Filter transfer optimization

For the case $P(H_0) = P(H_1) = 0.5$ **the treshold** γ **is:** $\gamma = \dfrac{\hat{s}_0(T_0) + \hat{s}_1(T_0)}{2}$ (2.8)

$$P(e) = Q\left(\frac{\hat{s}_0(T_0) - \hat{s}_1(T_0)}{2\sqrt{R_{\hat{n}}(0)}}\right)$$

$$R_{\hat{n}}(0) = \frac{N_0}{2} \int_{-\infty}^{\infty} |H(f)|^2 df = \frac{N_0}{2} \int_{-\infty}^{\infty} |h(t)|^2 dt = \frac{N_0}{2} \|h\|^2 \qquad (2.9)$$

$Q(.)$ is monotonically decreasing $\Rightarrow \min[P(e)] = \max\left[\frac{\hat{s}_0(T_0) - \hat{s}_1(T_0)}{2\sqrt{R_{\hat{n}}(0)}}\right]$

$$= \max\left[\frac{1}{\sqrt{2N_0}\|h\|} \int_{-\infty}^{\infty} [s_0(T_0 - t) - s_1(T_0 - t)]\, h(t)dt\right] \quad (2.10)$$

Cauchy-Schwarz inequality

$$\left|\int_{-\infty}^{\infty} f(t)g(t)dt\right| \leq \|f\| \cdot \|g\| \quad (2.11)$$

By substituting $f(t) = s_0(T_0 - t) - s_1(T_0 - t)$ **and** $g(t) = h(t)$ **into** (24) **we get:**

$$\int_{-\infty}^{\infty} [s_0(T_0 - t) - s_1(T_0 - t)]\, h(t)dt \leq \|s_0(T_0 - t) - s_1(T_0 - t)\| \cdot \|h\|$$

If $f(t) = \lambda g(t), \quad \lambda = const.$ **then we use the equality in** (24).

Under the assumption: $\boxed{h(t) = \lambda[s_0(T_0 - t) - s_1(T_0 - t)]}$ (2.12)

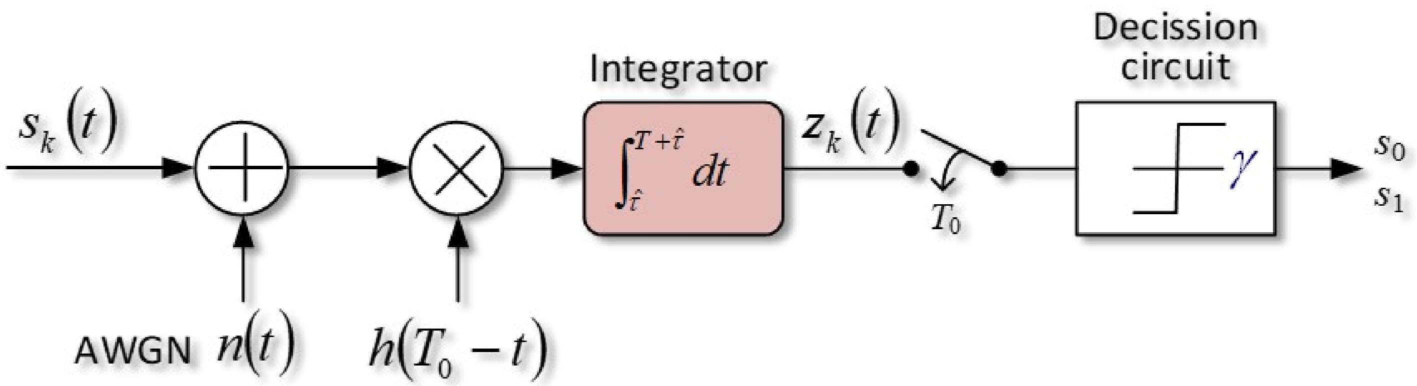
we can write $P(e) = Q\left(\dfrac{\|s_0(T_0 - t) - s_1(T_0 - t)\|}{\sqrt{2N_0}}\right) = Q\left(\sqrt{\dfrac{\|s_0 - s_1\|^2}{2N_0}}\right)$

And in the special case: $s_0(t) = -s_1(t)$ **the impulse response is**

$$\boxed{h(t) = s_0(T_0 - t), \qquad P(e) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right),} \tag{2.13}$$

where $E_b = E_0 = E_1$ **is the energy per bit**

The same decision characteristics $z_k(T_o)$ can be obtained by *correlation receiver*



$$\boxed{h(T_0 - t) = s_0[T_0 - (T_0 - t)] - s_1[T_0 - (T_0 - t)] = s_0(t) - s_1(t)}$$

The numerator of (2.10) **will then be equal to the same convolution given by the last term of** (2.10):

$$\hat{s}_0(t) - \hat{s}_1(t) = \int_{-\infty}^{\infty} [s_0(t) - s_1(t)]\, h(T_0 - t)dt$$

$$= \int_{-\infty}^{\infty} [s_0(T_0 - t) - s_1(T_0 - t)]\, h(t)dt$$

Let the bandpass binary signal be of the form $s_0(t) = Av_0(t)\cos(2\pi f_c t + \theta)$
$s_1(t) = Av_1(t)\cos(2\pi f_c t + \theta)$

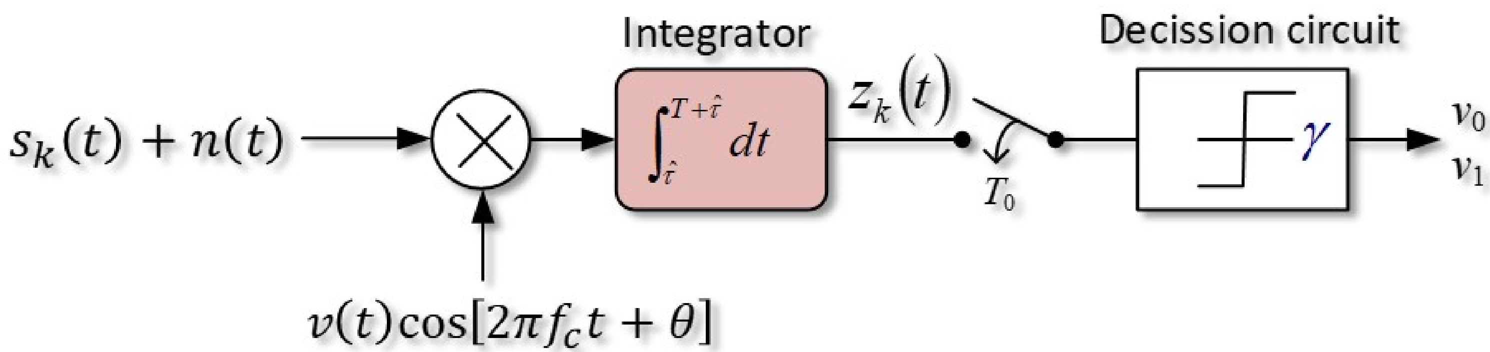
where $v_0(t)$ **and** $v_1(t)$ **are the baseband signals "0" and "1", then using** (2.11) **we can form the matched filter impulse response in the form**

$$h(t) = s_0(T_0 - t) - s_1(T_0 - t)$$
$$= [v_0(T_0 - t) - v_1(T_0 - t)]\cos[2\pi f_c(T_0 - t) + \theta]$$

If we further assume that $v_0(t) = -v_1(t) = v(t)$ **the matched filter impulse response is** $h(t) = v(T_0 - t)\cos[2\pi f_c(T_0 - t) + \theta]$ **and**

$$h(T_0 - t) = v(t)\cos[2\pi f_c t + \theta]$$

Bandpass correlation receiver



The average bit error probability is $P(e) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$

where E_b is energy per bit given by

$$E_b = \int_{-\infty}^{\infty} A^2 v^2(t)\cos^2[2\pi f_c t + \theta]dt$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} A^2 v^2(t)dt + \frac{1}{2}\int_{-\infty}^{\infty} A^2 v^2(t)\cos^2[4\pi f_c t + 2\theta]dt = \boxed{\frac{A^2}{2}\int_{-\infty}^{\infty} v^2(t)dt}$$

Synchronization in communication systems include:

1. carrier recovery (phase and frequency recovery),
2. symbol timing recovery,
3. frame synchronization.

Carrier Recovery (CR)

- **The *carrier frequency* of the received signal may differ from that of the nominal transmitter carrier frequency due to the deviation of the transmitter oscillator from the nominal frequency and due to the Doppler effect.**
- **The *carrier phase* of the received signal consists of the three major components:**
  - ✓ **the random phase of the transmitter oscillator,**
  - ✓ **the phase due to the transmission delay,**
  - ✓ **the channel phase response.**

The received signal affected by the above phenomena can be described as

$$r(t) = Av(t-\tau)\cos[2\pi(f_c - f_d)(t-\tau) + \theta] + n(t), \tag{3.1}$$

where $v(t)$ is the baseband signal, f_d and τ are the deviation of the received carrier frequency and the transmission delay, respectively and $n(t)$ is an AWGN.

The total phase error is $\varphi = \theta - 2\pi(f_c - f_d)\tau$

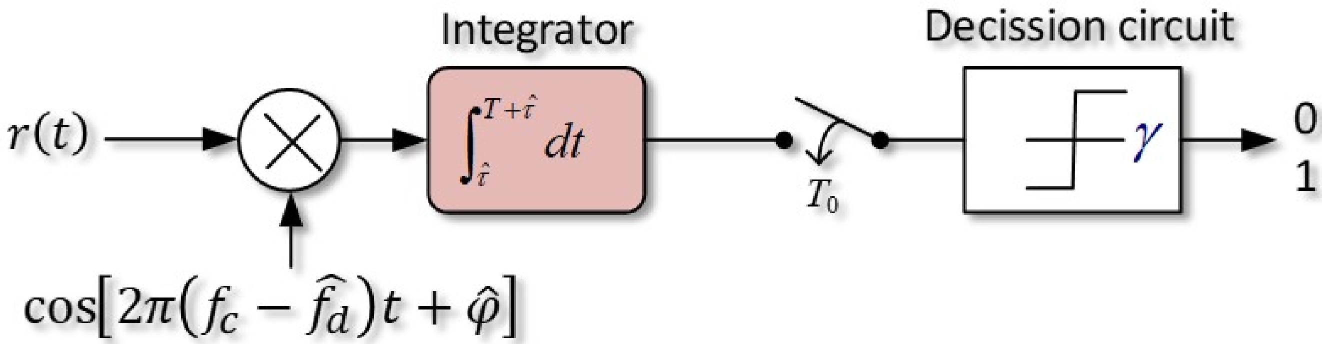
Estimating the parameters f_d, τ, and θ is called ***synchronization.***

Effect of Synchronization Errors: let us consider a BPSK system

$$r(t) = \pm A p_T(t-\tau)\cos[2\pi(f_c - f_d)t + \varphi] + n(t),$$

with an AWGN process $n(t)$ having a noise spectral density $N_0/2$ and symbols $p_T(t)$. Our aim is to obtain estimates $\hat{\varphi}, \hat{f}_d$, and $\hat{\tau}$ of the parameters φ, f_d, and τ, respectively.

We can form the *correlator demodulator* based on these estimated parameters



For the case $|\hat{\tau} - \tau| \leq T$ the average error probability can be expressed in the form

$$P_b = Q\left(\alpha\sqrt{\frac{2E_b}{N_0}}\right)$$

where $$\alpha = \frac{1}{T}\int_{\max(\tau,\hat{\tau})}^{T+\min(\tau,\hat{\tau})} \cos\left[2\pi\left(f_d - \hat{f}_d\right)t + (\varphi - \hat{\varphi})\right] dt$$

Note that $\alpha \leq 1$ and $\alpha = 1$ only if $\hat{f}_d = f_d, \hat{\varphi} = \varphi, \hat{\tau} = \tau$.

In the case when $|f_c - f_d| << 1/T$ i.e., the data rate is much higher than the estimation error in f_d, then

$$\alpha \approx \left(1 - \frac{|\hat{\tau} - \tau|}{T}\right)\cos(\hat{\varphi} - \varphi)$$

Maximum likelihood carrier phase estimation

Under simplifying assumption $f_d = 0$ and $\tau = 0$ the equation (3.1) reduces to

$$r(t) = Av(t)\cos[2\pi f_c t + \varphi] + n(t)$$

Our aim is to obtain the carrier phase estimation $\hat{\varphi}$ based on the ML principle
The signal space in this case is 2-D and it is given by the basis functions

$$\varphi_1(t) = \frac{\sqrt{2}}{\|v\|} v(t)\cos(2\pi f_c t), \qquad \varphi_2(t) = \frac{\sqrt{2}}{\|v\|} v(t)\sin(2\pi f_c t).$$

Using (1.12) **the elements of the vector representing** $r(t)$ **are**

$$r_1 = \int_{-\infty}^{\infty} r(t)\varphi_1(t)\,dt = \frac{A\|v\|\cos\varphi}{\sqrt{2}} + n_1,$$
$$r_2 = \int_{-\infty}^{\infty} r(t)\varphi_2(t)\,dt = \frac{A\|v\|\sin\varphi}{\sqrt{2}} + n_2$$

where n_1 **and** n_2 **are zero-mean Gaussian random variable with variance** $N_0/2$**.**

The likelihood function is given by

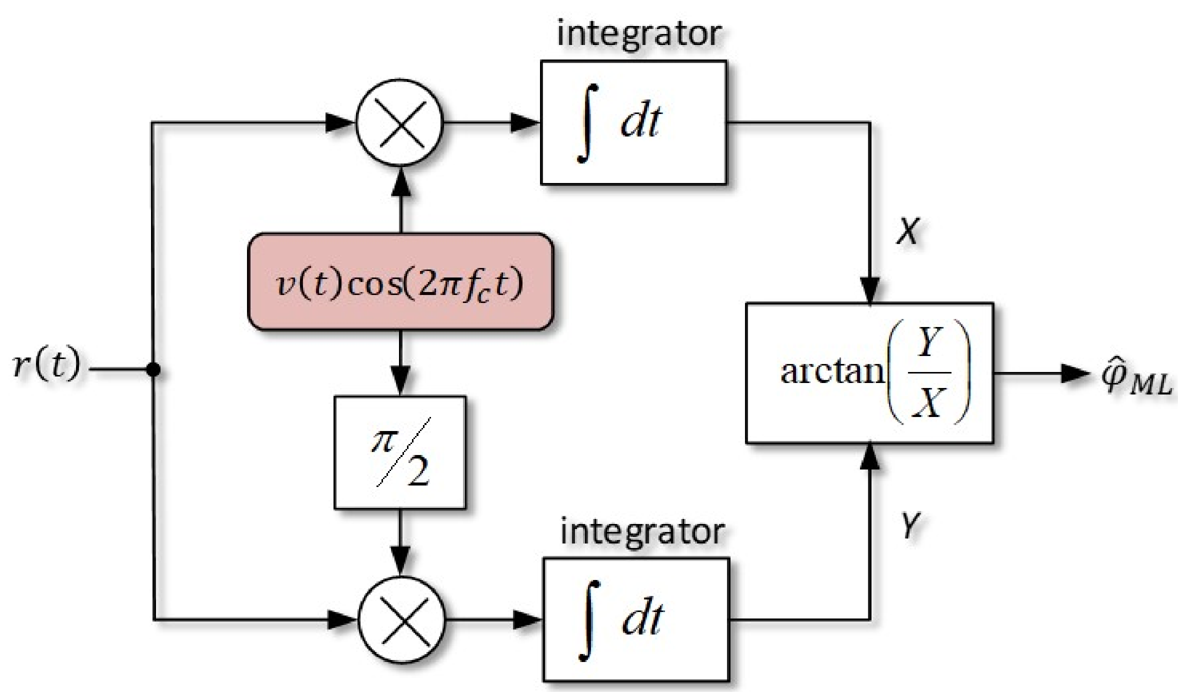
$$f(r_1, r_2|\varphi) = \frac{1}{\pi N_0}\exp\left[-\frac{\left(r_1 - A\|v\|\cos\varphi/\sqrt{2}\right)^2 + \left(r_2 + A\|v\|\sin\varphi/\sqrt{2}\right)^2}{N_0}\right]$$

The ML estimator then maximizes the likelihood function or its log-likelihood equivalent.

$$\hat{\varphi}_{ML} = \arg\max_{\varphi}\left(r_1 \cos\varphi - r_2 \sin\varphi\right) = \arg\max_{\varphi} \ln f\left(r_1, r_2 | \varphi\right)$$

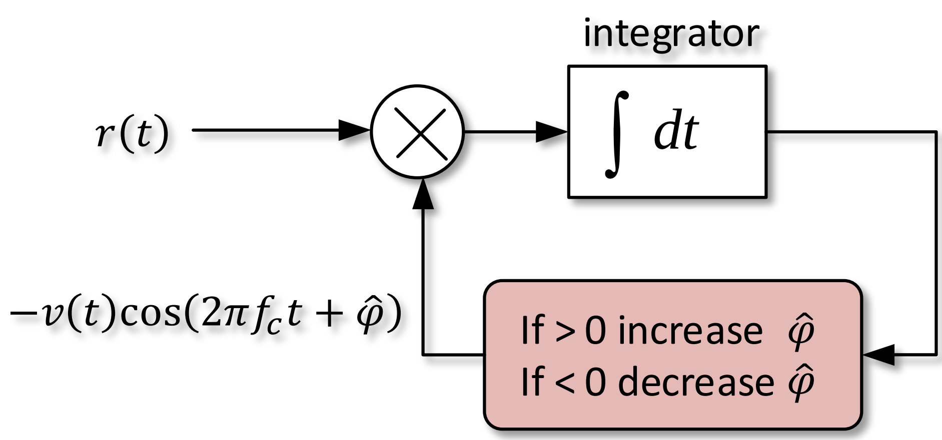
$$= \tan^{-1}\left(-\frac{r_2}{r_1}\right) = \tan^{-1}\left[\frac{\int_{-\infty}^{\infty} r(t)v(t)\sin(2\pi f_c t)\,dt}{\int_{-\infty}^{\infty} r(t)v(t)\cos(2\pi f_c t)\,dt}\right] \qquad (3.2)$$

ML phase estimator can be implemented by the circuit



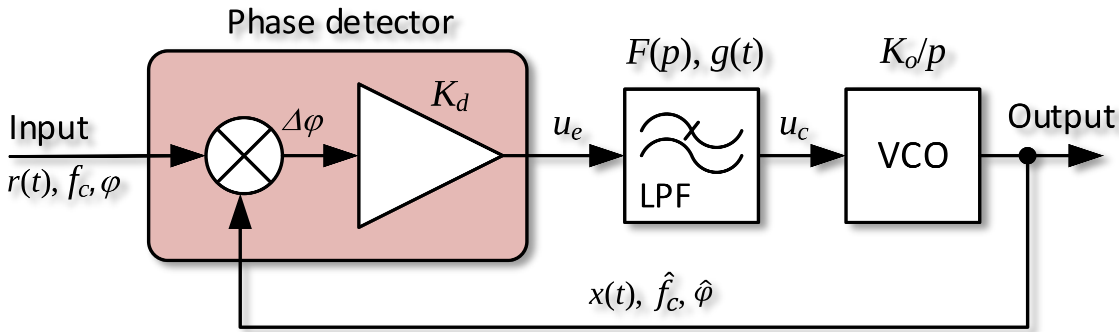
Other (more popular) way to get the ML phase estimator is obtained by differentiating the metric $r_1 \cos\varphi - r_2 \sin\varphi$ with respect to φ and setting the derivative to zero i.e. $d(r_1 \cos\varphi - r_2 \sin\varphi)/d\varphi = 0$. The ML estimator has then to satisfy

$$-\int_{-\infty}^{\infty} r(t)v(t)\sin(2\pi f_c t + \varphi)_{ML}\, dt = 0 \qquad (3.3)$$



Such structure is generally known as *phase-locked loop.*

Analysis of a simplified linearized PLL model



The carrier $r(t)$ **to be "locked" the and reference signal** $x(t)$ **are**

$$r(t) = A\cos[2\pi f_c t + \varphi], \quad x(t) = \sin\left[2\pi \hat{f}_c t + \hat{\varphi}(t)\right]$$

The phase detector consists of a multiplier followed by a low-passed filter (LPF) that suppress the double frequency term. The detector converts the phase error $\Delta\varphi(t) = \varphi - \hat{\varphi}(t)$ **into a voltage** $u_e(t) = K_d f_e(t)$,

where

$$K_d = \frac{u_e}{\Delta\varphi} = \frac{u_e}{\varphi - \hat{\varphi}}$$

VCO (*Voltage controlled oscillator*) gain is defined as $K_o = \frac{\Delta\omega_o}{u_c}$,

where $\Delta\omega_o$ **is the frequency deviation from *natural frequency*** ω_o, **on which the VCO oscillates at** u_c = 0.

As the phase deviation at the VCO output is $\Delta\widehat{\omega} = \frac{d\widehat{\varphi}(t)}{dt}$ **, we can write**

$$\frac{d\hat{\varphi}(t)}{dt} = K_o K_d[\varphi - \hat{\varphi}(t)] * g(t) \quad \overset{LT}{\rightarrow} \quad p\widehat{\Phi}(p) = K_o K_d\left[\Phi(p) - \widehat{\Phi}(p)\right]F(p)$$

where $g(t) = L^{-1}\{F(p)\}$ **is the LPF impulse response (operators** *L* **and** *LT* **denote the *Laplace Transform*)**

The ratio of phase images is: $$H(p) = \frac{\widehat{\Phi}(p)}{\Phi(p)} = \frac{K_o p^{-1} K_d F(p)}{1 - K_o p^{-1} K_d F(p)} \qquad (3.4)$$

Because $K_d\left[\Phi(p)-\widehat{\Phi}(p)\right]=K_d[1-H(p)]\Phi(p)$ **we can simply find that the error voltage satisfies the relation**

$$\boxed{U_e(p)=\frac{\Phi(p)K_d}{1-K_o p^{-1}K_d F(p)}} \qquad (3.5)$$

To investigate the PLL response to various stimuli we will apply the Limit theorem for a function and its Laplace image

$$\lim_{t\to\infty} u_e(t)=\lim_{p\to 0} p\,U_e(p)$$

PLL response to a step change of the phase $\Phi(p) = \Delta\varphi/p$

$$\lim_{t\to\infty} u_e(t)=\lim_{p\to 0}\frac{p\Delta\varphi K_d}{p-K_o K_d F(p)}=0$$

<u>PLL response to a step change of the frequency</u> $\Phi(p) = \Delta\varphi/p^2$

$$\lim_{t\to\infty} u_e(t) = \lim_{p\to 0} \frac{\Delta f_c K_d}{p - K_o K_d F(p)} = \frac{\Delta f_c}{K_o F(0)}$$

velocity error

The error voltage $u_e(t)$ **will approach zero, if** $F(p)=\mathrm{const.}/p$ **i.e. if the LPF behaves like an integrator.**

Phase locked loop for modulated signal

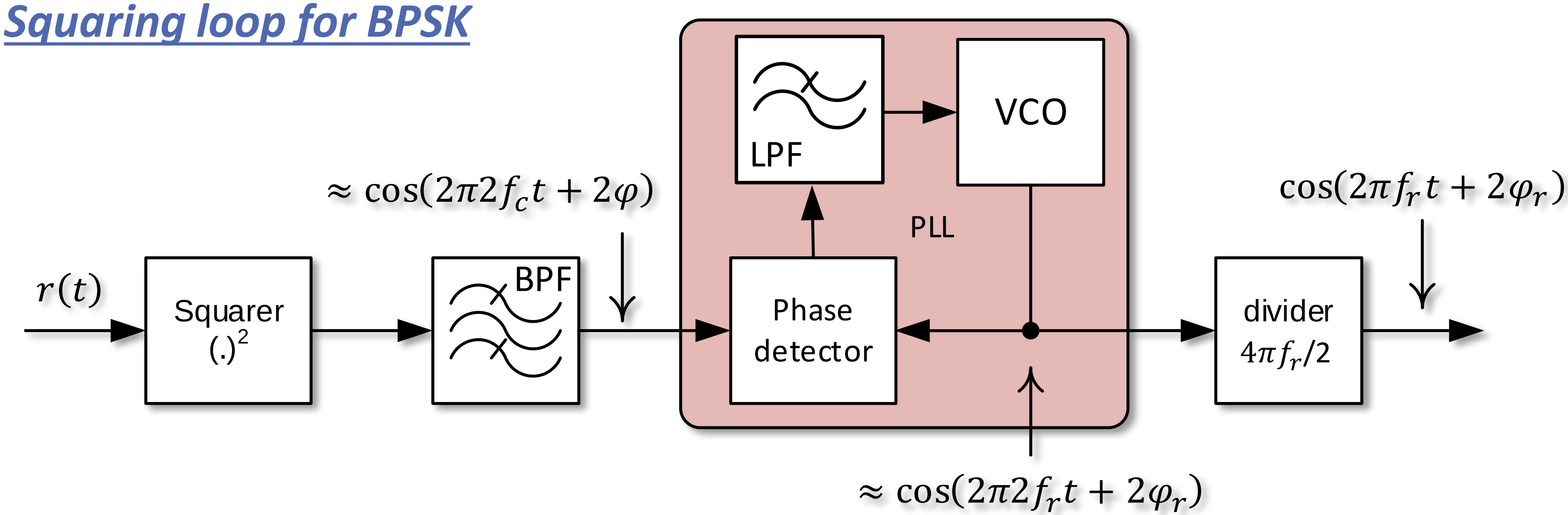
For the BPSK modulated signal the Carrier phase can be tracked using a squarer. The received signal can be expressed as

$$r(t) = A\left[\sum_n b_n p_T(t - nT)\right]\cos(2\pi f_c t + \varphi) \tag{3.6}$$

The effect of the data signal $(b_n = \pm 1)$ **can be removed by squaring**

$$r^2(t) = \frac{A^2}{2}[1 + \cos(2\pi 2 f_c t + 2\varphi)]$$

Squaring loop for BPSK

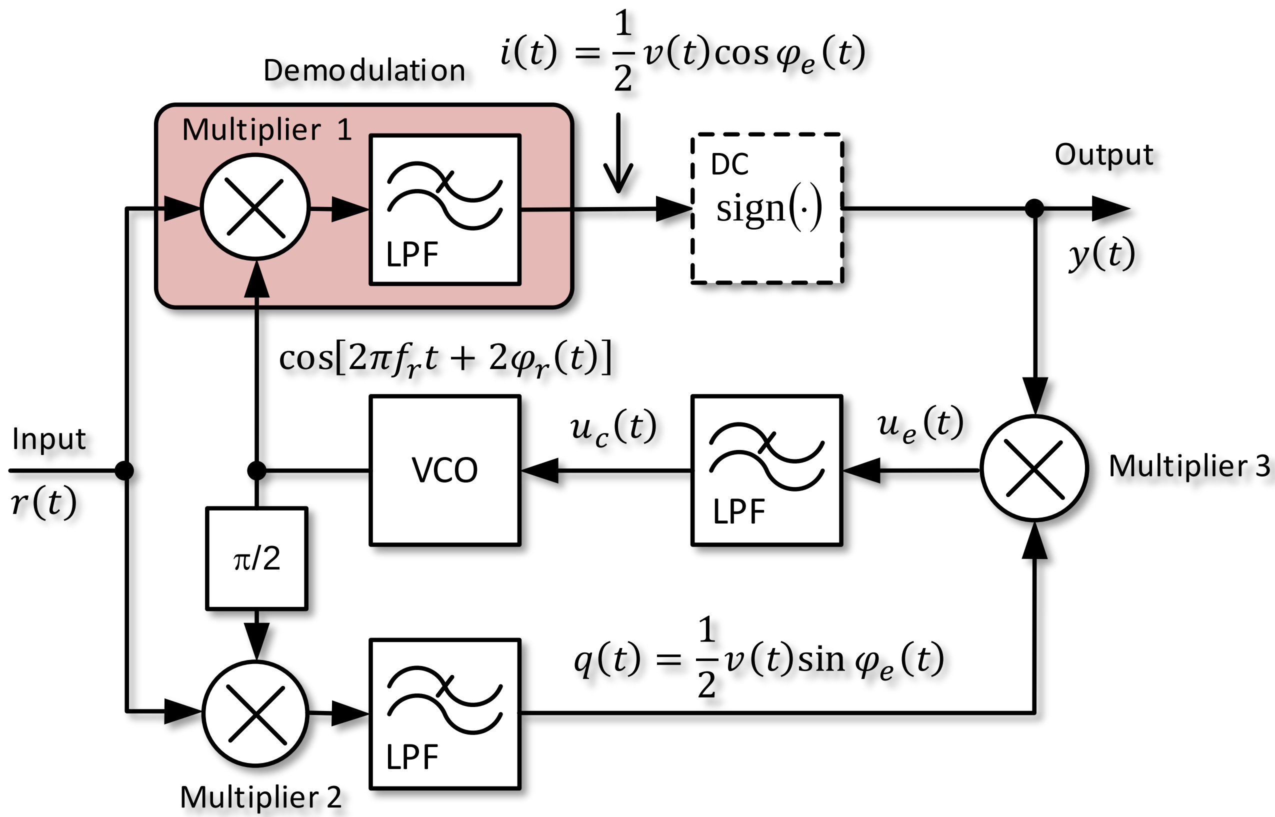


Note that since we are only able to determine $2\varphi_r$, there is a phase ambiguity of π in our estimate. To overcome this problem, the data must be differentially encoded and decoded.

For the case of *M*-ary PSK modulation where $\varphi_i = 2\pi i/M \quad i = 0,1,2,\dots,M$ we use squaring $r^M(t)$ and dividing by *M*.

Costas loop for BPSK

It uses similar idea to the squaring loop - the data signal is removed by the multiplication (in Multiplier 3) before the loop filter.

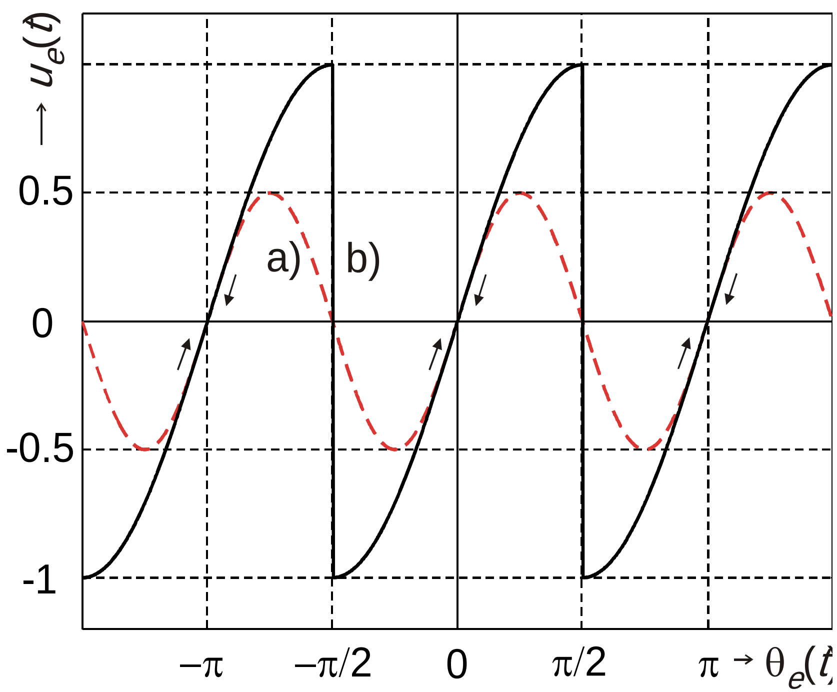


Output voltage is $$y(t) = \operatorname{sign}\left[\frac{1}{2}v(t)\cos\varphi_e(t)\right] = \operatorname{sign}[v(t)]\Big|_{\varphi_e(t)\to 0}$$

Error voltage is $$u_e(t) = \frac{1}{2}|v(t)|\sin\varphi_e(t)\Big|_{\varphi_e\to 0} \approx \sqrt{\frac{E_b}{2T}}\varphi_e$$

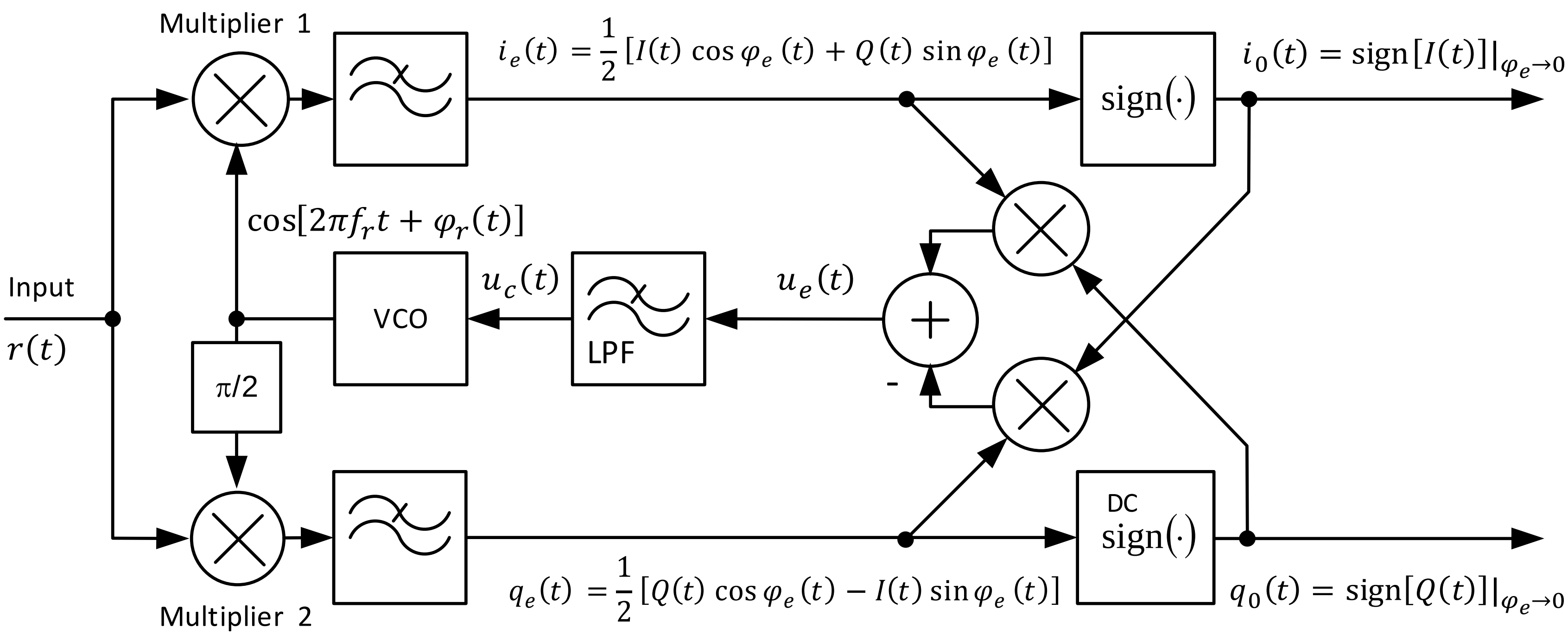
Dependence of the error voltage $u_e(t)$ on the phase error $\varphi_e(t)$.

- a) **without** sign(.) **block,**
- b) **with** sign(.) **block**



Costas loop for QPSK

$$r(t) = I(t)\cos(2\pi f_c t + \varphi) + Q(t)\sin(2\pi f_c t + \varphi)$$



$$u_e(t) = \frac{1}{2}[|I(t)| + |Q(t)|]\sin\varphi_e(t) + \frac{1}{2}[I(t)\,\text{sign}\,Q(t) - Q(t)\,\text{sign}\,I(t)]\cos\varphi_e(t)$$

Let f_d = 0 and carrier phase synchronization was achieved i.e. the value of φ is known. Then we can write

$$r(t) = Av(t-\tau)\cos[2\pi f_c t + \varphi] + n(t), \tag{3.7}$$

ML symbol timing estimation with training signal

Let the baseband signal $v(t)$ be known i.e., a training signal is sent to allow the receiver to ensure the symbol timing synchronization. The goal of STR is to work out the ML estimator for τ. First we will create a basis $\{\varphi_i(t)\}_{i=1}^{\infty}$ and represent the received signals by vectors

$$\mathbf{r} = \mathbf{s}(\tau) + \mathbf{n}.$$

Where the i-th elements of vector $\mathbf{r}$ is

$$r_i = s_i(\tau) + n_i = \int_{-\infty}^{\infty} Av(t-\tau)\cos[2\pi f_c t + \varphi]\,\varphi_i(t)dt + \int_{-\infty}^{\infty} n(t)\,\varphi_i(t)dt$$

A sufficient statistic for the estimation of τ is the whole vector $\mathbf{r}$. Thus we start by the truncated observation vector r_K and then let K increase to infinity.

$$f(\mathbf{r}_K|\tau) = \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi N_0}} \exp\left[-\frac{\left(r_k - s_k(\tau)\right)^2}{N_0}\right] = \frac{1}{(2\pi N_0)^{K/2}} \exp\left[-\sum_{k=1}^{K} \frac{[r_k - s_k(\tau)]^2}{N_0}\right]$$

then (3.8)

$$\ln f(\mathbf{r}_K|\tau) = -\frac{K}{2}\ln(2\pi N_0) - \frac{1}{N_0}\sum_{k=1}^{K}[r_k - s_k(\tau)]^2$$

The maximum of $\ln f(\mathbf{r}|\tau)$ over τ is equivalent to finding the signal $s_k(\tau)$ that minimizes the Euclidean distance

$$D\left(\mathbf{r}_K, s_k(\tau)\right) = \sum_{k=1}^{K}[r_k - s_k(\tau)]^2 = \sum_{k=1}^{K} r_k^2 - 2\sum_{k=1}^{K} r_k s_k(\tau) + \sum_{k=1}^{K} s_k(\tau)^2$$

The term $\sum_{k=1}^{K} {r_k}^2$ is common to all decision metrics and, hence, it may be ignored.

The ML estimator is then given by $\hat{\tau}_{ML} = \arg\max_{\tau} \sum_{k=1}^{K} [2r_k s_k(\tau) - s_k(\tau)^2]$

Now, letting *K* go to infinity, we have

$$\hat{\tau}_{ML} = \arg\max_{\tau} \left[2A \int_{-\infty}^{\infty} r(t)v(t-\tau)\cos(2\pi f_c t + \varphi)dt - \right.$$
$$\left. -A^2 \int_{-\infty}^{\infty} v^2(t-\tau)\cos^2(2\pi f_c t + \varphi)dt \right]$$
$$= \arg\max_{\tau} \left[\int_{-\infty}^{\infty} r(t)v(t-\tau)\cos(2\pi f_c t + \varphi)dt \right]$$

It is obvious that the ML estimator has to satisfy

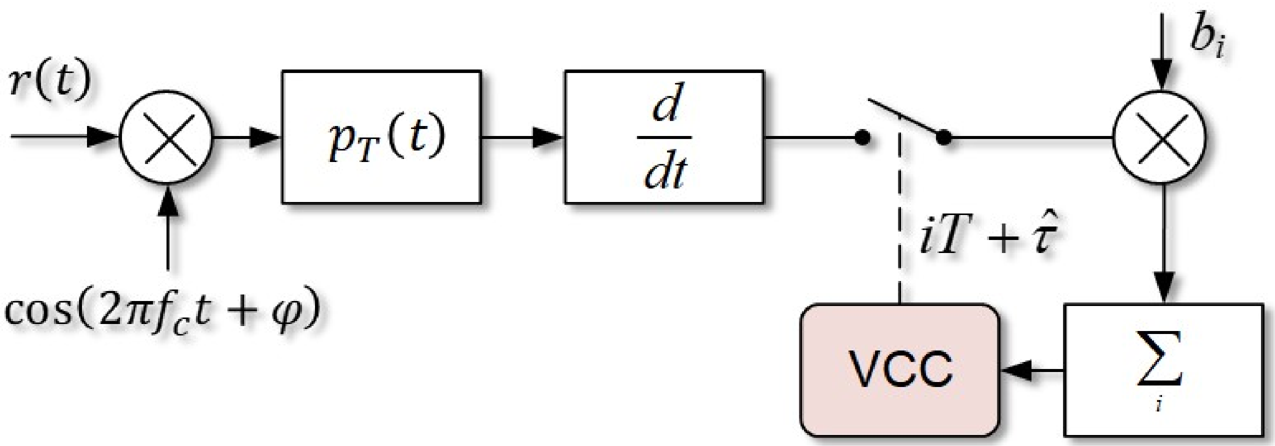
$$\boxed{\frac{d}{d\tau} \int_{-\infty}^{\infty} r(t)v(t-\tau)\cos(2\pi f_c t + \varphi)dt \bigg|_{\tau=\hat{\tau}_{ML}} = 0} \qquad (3.9)$$

For the case where the training signal is a known sequence of BPSK pulses, i.e. $v(t) = \sum_i b_i p_T(t - iT)$, the necessary condition in (3.9) **becomes**

$$\sum_i b_i \frac{d\tilde{r}}{d\tau}(\hat{\tau}_{ML} - iT) = 0, \qquad (3.10)$$

where $\tilde{r} = [r(t)\cos(2\pi f_c t + \varphi) * p_T](\tau)$

To solve for $\hat{\tau}_{ML}$ **we can employ a feedback structure, known as *delay-locked loop* (DLL)**



VCC: voltage controlled clock

Non-decision-directed ML symbol timing estimation

Let a finite number of bits $v(t) = \sum_i b_i p_T(t - iT)$ defined by a vector of symbols $\mathbf{b} = [b_1, b_2, \dots, b_I]$ be transmitted using the BSPK. Then for each bit sequence b, we can create an equation describing the received signal using vector notation based on the basis in the previous section

$$\mathbf{r} = \mathbf{s}(\tau, \mathbf{b}) + \mathbf{n}.$$

As in the previous case we assume the truncated observation vector r_K and then let K increase to infinity. Symbol vector b will be modelled as a random vector with equal-probable bit patterns. The likelihood function is obtained by averaging the conditional density function over all possible bit patterns

$$f(\mathbf{r}_K|\tau) = \overline{f(\mathbf{r}_K|\tau, \mathbf{b})} = \frac{1}{2^I} \sum_{\mathbf{b}} f(\mathbf{r}_K|\tau, \mathbf{b})$$

$$= \frac{1}{2^I} \sum_{\mathbf{b}} \prod_{k=1}^{K} \frac{1}{\sqrt{2\pi N_0}} \exp\left[-\frac{[r_k - s_k(\tau, \mathbf{b})]^2}{N_0}\right]$$

Letting K approaches infinity and removing constant term we have

$$\begin{aligned}\hat{\tau}_{ML} &= \arg\max_{\tau} \ln\left\{\frac{1}{2^I}\sum_{\mathbf{b}}\exp\left[-\frac{1}{N_0}\sum_{k=1}^{\infty}[r_k - s_k(\tau,\mathbf{b})]^2\right]\right\} \\ &= \arg\max_{\tau} \ln\left\{\frac{1}{2^I}\sum_{\mathbf{b}}\exp\left[\frac{2A}{N_0}\sum_{k=1}^{\infty} b_i\tilde{r}(\tau+iT)\right]\right\} \qquad (3.11) \\ &= \arg\max_{\tau} \ln\left\{\prod_i \cosh\left[\frac{2A}{N_0}\tilde{r}(\tau+iT)\right]\right\} = \arg\max_{\tau}\sum_i \ln\cosh\left[\frac{2A}{N_0}\tilde{r}(\tau+iT)\right]\end{aligned}$$

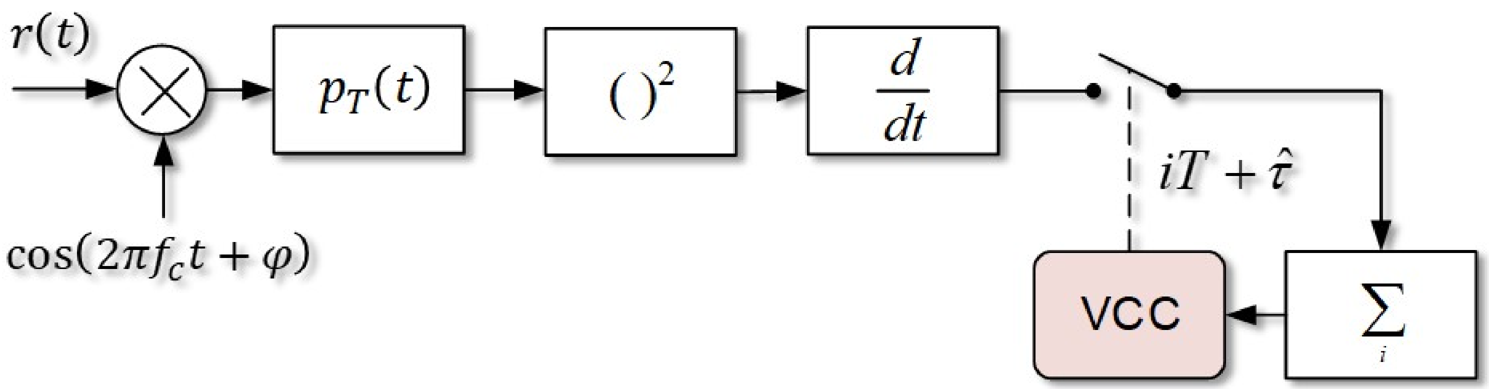
For small x (in our case for low SNR) we can use approximation $\ln\cosh x \approx \frac{1}{2}x^2$ **then we can rewrite** (3.11) **to**

$$\hat{\tau}_{ML} \approx \arg\max_{\tau}\sum_i \tilde{r}^2(\tau+iT)$$

The ML estimator must satisfy condition

$$\sum_i \frac{d\tilde{r}^2}{d\tau}(\hat{\tau}_{ML} - iT) = 0,$$

which can be implemented by a feedback structure known as *Delay-locked loop* (DLL)

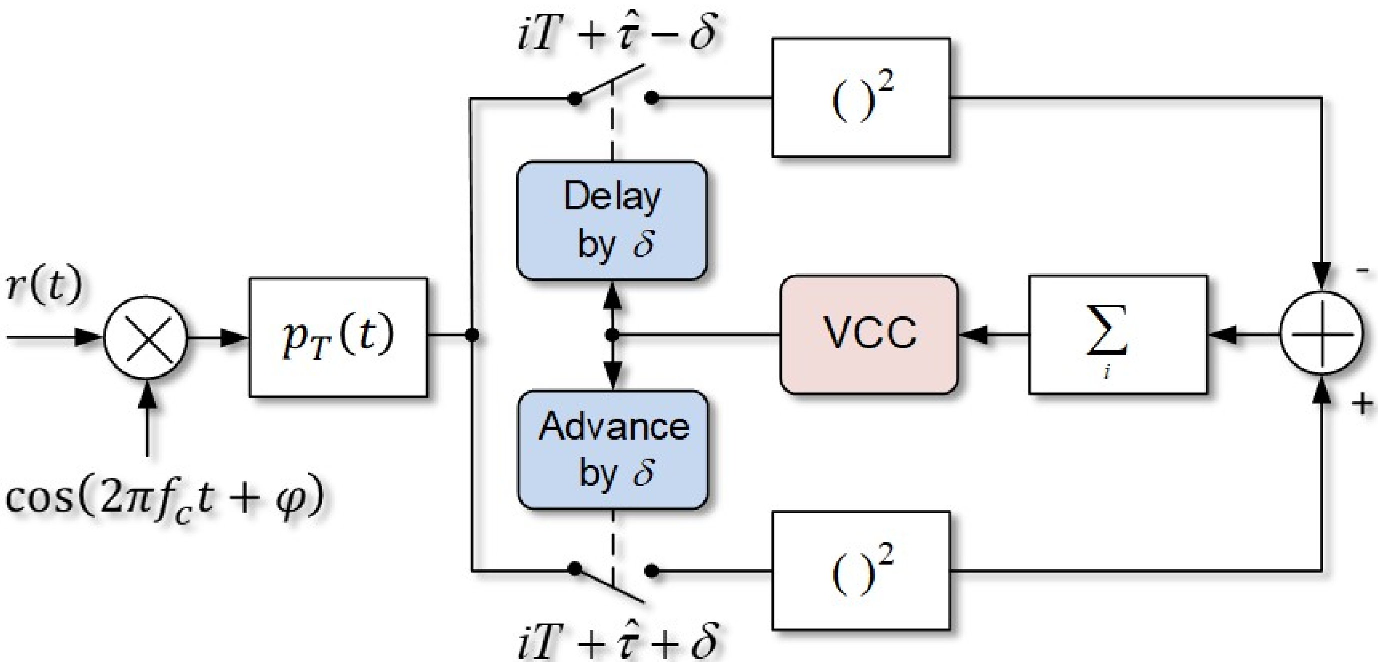


VCC: voltage controlled clock

The derivative of $\tilde{r}^2(\tau + iT)$ can be further approximated by

$$\frac{d\tilde{r}^2}{d\tau}(\tau) = \lim_{\delta \to 0}\left[\frac{\tilde{r}^2(\tau + iT + \delta) - \tilde{r}^2(\tau + iT - \delta)}{2\delta}\right], \tag{3.12}$$

where δ is in reality a short time interval before and after the sampling points $\tau + iT$. It allows us to construct the ML estimator known as the *early-late gate DLL.*



Channels containing transmitter (TX), propagation environment, and receiver (RX) have a non-flat frequency response due to:

- Transmitting filter (TF): pulse-shaping, signal bandwidth limitation.
- Channel filter (CF): distributed reactances (wire cable), multipath propagation (wireless system), chromatic dispersion (fiber optic system).
- Receiving filter (RF): input noise elimination.

The goal of equalization is:

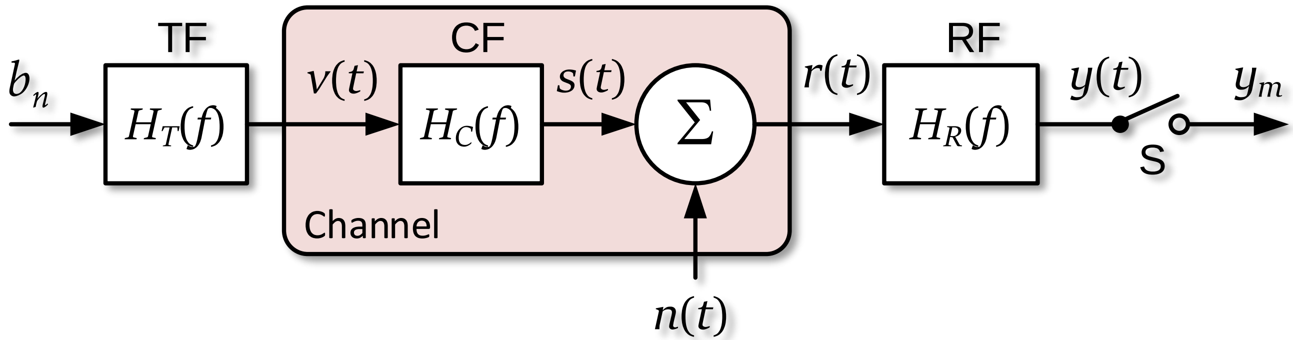
- Inter-symbol interferences (ISI) elimination.
- Flattening of the channel frequency response.

Equalizer types:

- Linear: Zero forcing, Minimum mean square (MMSE),
- Nonlinear: Decision-feedback (DFE), Maximum likelihood sequence estimation (MLSE).

As some (mobile) channels have time-varying characteristics adaptive equalizers must be used.

Let us consider a digital communication system by means of *M*-ary PAM that consists of a TF having a transfer function $H_T(f)$, the linear CF $H_C(f)$ with AWGN $n(t)$, and an RF with transfer function $H_R(f)$, a sampler S that periodically samples the output of the receiving filter.



Signals at the TF output and RF input may be expressed as

$$v(t) = \sum_{n=-\infty}^{\infty} b_n h_T(t - nT) \quad \text{and} \quad r(t) = \sum_{n=-\infty}^{\infty} b_n h_T(t - nT) * h_C(t) + n(t) \tag{4.1}$$

where b_n is the n-th transmitted symbol, and $*$ stands for convolution.

Our aim is design of a bandlimited TF. It will be done first under the assumption that there is no channel distortion i.e. $H_C(f) = 1$. The RF will then be realized as a matched filter with the transfer function $H_R(f)=H_T^*(f)$. Let its output $y(t)$ will be *periodically sampled* at the times $t=mT$ then

$$y(mT) = h(0)b_m + \sum_{\substack{n=-\infty \\ m\neq n}}^{\infty} b_n h(mT - nT) + \nu(mT), \tag{4.2}$$

where $h(t) = h_T(t) * h_R(t)$ and $\nu(t)$ is the output response of the matched filter to the input AWGN process $n(t)$.

The second term in (4.2) expresses the undesirable effect of all other transmitted bits on decoding the m-th bit. To remove the effect of ISI, it is necessary and sufficient that $h(iT) = h(mT - nT)$ must satisfy the condition

$$h_i = \begin{cases} 1, & \text{for} \quad i = 0, \\ 0 & \text{for} \quad i \neq 0. \end{cases} \tag{4.3}$$

For the TF design it is better to express the condition (4.3) in the frequency domain. It is generally known that the spectrum of sampled signal $h(t)$ with the sampling period T is

$$H_s(f) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H\left(f - \frac{k}{T}\right) \tag{4.4}$$

It is obvious that the sampled response $h(iT)$ given by (4.3) corresponds to the Dirac impulse $\delta(t)$ whose spectrum is $F\{\delta(t)\} = 1$. Thus, under assumption that $R = 1/T$ we can rewrite (4.4) to the form known as the ***Nyquist pulse-shaping criterion***

$$\boxed{\sum_{k=-\infty}^{\infty} H(f - kR) = T} \tag{4.5}$$

The criterion (4.5) can be met if the spectrum of the sampled signal $h(iT)$ is constant over all frequencies i.e.

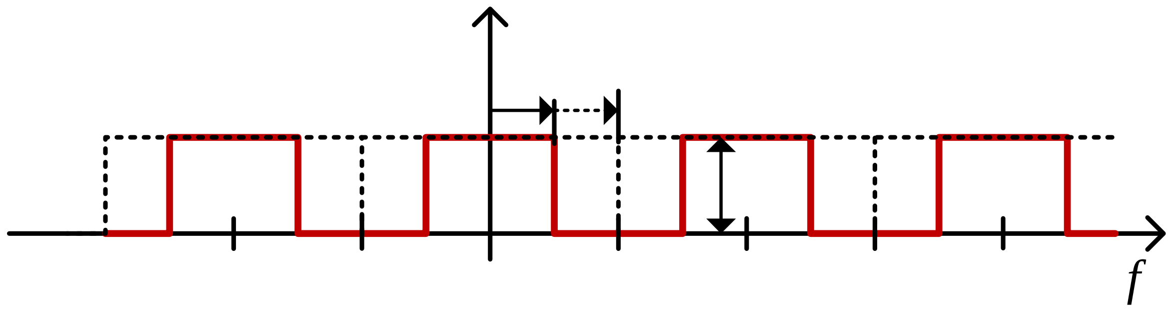
$$\mathrm{H}(f) = \begin{cases} T, & \text{for} \quad |f| \le 0.5R, \\ 0, & \text{for} \quad |f| > 0.5R. \end{cases} \tag{4.6}$$

4. Equalization

The relation (4.6) **is the transfer function of the *Nyquist filter* (ideal lowpass filter) with the cut-off frequency of** $B_T = 0.5R$**. Its impulse response is**

$$h(t) = \frac{\sin(2\pi B_T t)}{2\pi B_T t} = \mathrm{sinc}(2\pi B_T t)$$

Frequency response of sampled Nyquist filter transfer function



Note that the Nyquist filter is unrealizable. The constant spectrum of the sampled impulse response can be realized also by so called *raised-cosine filter* (RC filter) whose transfer function is

$$H_{RC}(f) = \begin{cases} T, & \text{for} \quad 0 \leq |f| < \dfrac{1-\beta}{2T}, \\ \dfrac{T}{2}\left[1 - \sin\dfrac{\pi T}{\beta}\left(|f| - \dfrac{1}{2T}\right)\right], & \text{for} \quad \dfrac{1-\beta}{2T} \leq |f| < \dfrac{1+\beta}{2T}, \\ 0, & \text{for} \quad \dfrac{1+\beta}{2T} \leq |f|, \end{cases}$$

where the *roll-off factor* β is a measure of the excess bandwidth of the filter, i.e. the bandwidth occupied beyond the Nyquist bandwidth B_T.

The impulse response of the raised-cosine filter is

$$h_{RC}(t) = \operatorname{sinc}(\pi R t)\,\frac{\cos \beta\,\pi R t}{1 - (2\beta R t)^2}$$

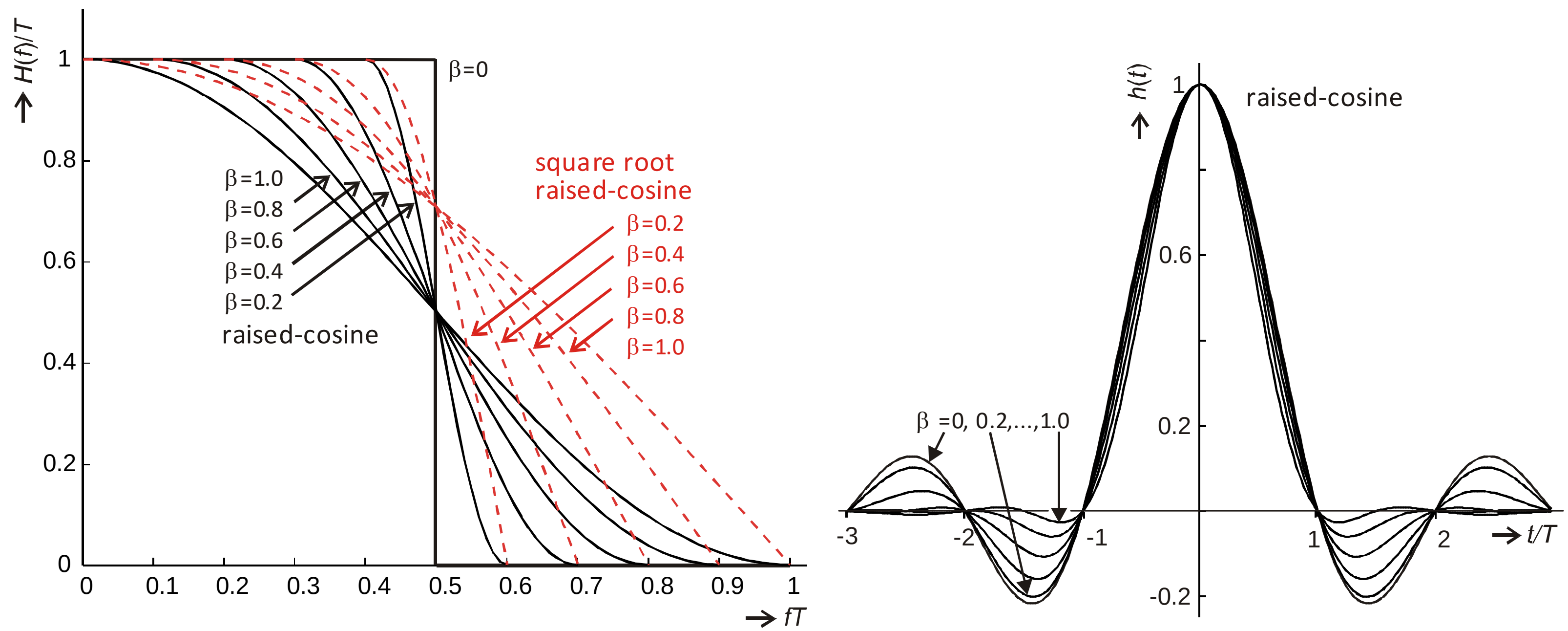
As mentioned above $H_R(f)=H_T^*(f)$ and **for** $H_C(f)=1$ **the total impulse response is**

$$H_{RC}(f) = H_T(f)H_T^*(f). \tag{4.7}$$

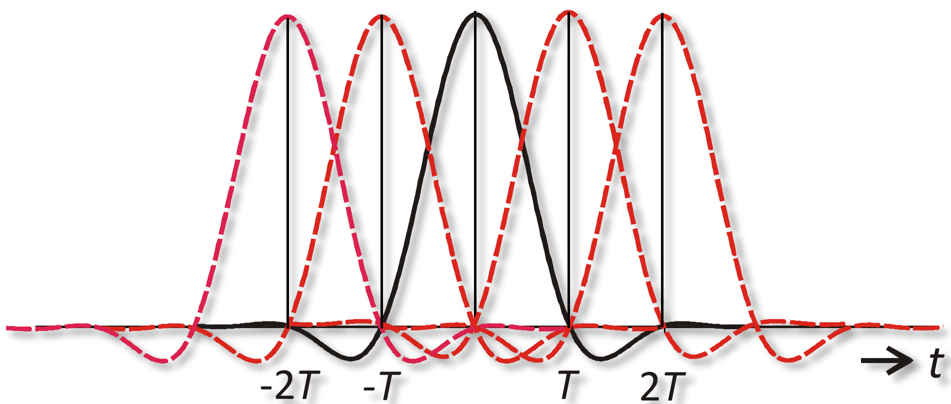
Hence, the TF filter transfer function is

$$H_T(f) = \sqrt{H_{RC}(f)}.$$

It is known as the *square root raised-cosine (SRC)* response.



It is obvious from the raised-cosine impulse response that pulse sidelobes go through zero at the sample times adjacent to the main lobe of the pulse.



Let we have the sequence of information symbols denoted by $\{I_k\}$ and a common (not constant) channel transfer function $H_C(f)$. Further, let overall transfer function is

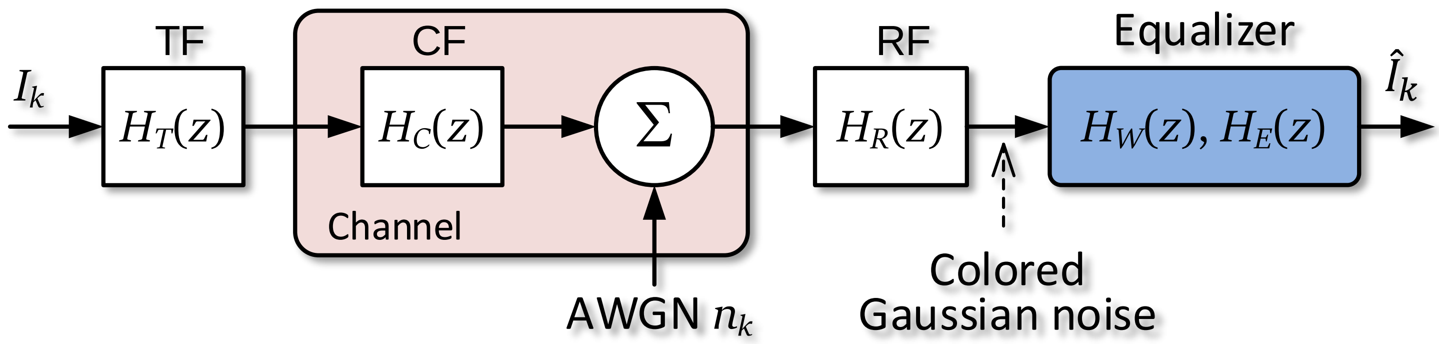
$$H(f) = H_T(f)H_C(f)H_R(f).$$

To achieve the Nyquist condition the filter cascade $H_T(f)H_C(f)H_R(f)$ must fullfil the relation (4.5). It can be achiewed when

- **RF is constructed to be the matched filter, i.e. $H_R(f) = H_T{}^*(f)H_C{}^*(f)$. The TF is then chosen so that (4.5) is satisfied.**
- **The TF is fixed and the RF is chosen so that (4.5) is satisfied.**

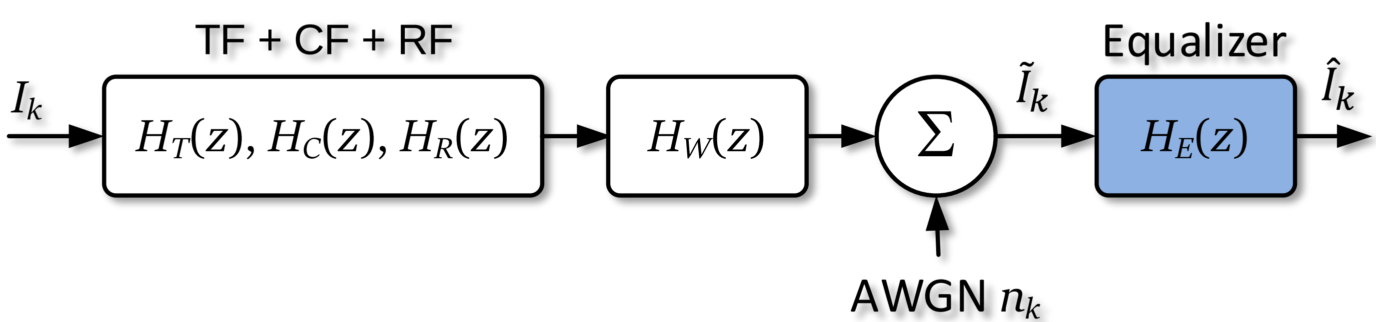
An analog realization of TF and RF filters is very difficult $\Rightarrow$ we use digital filter called *equalizer* which can remove (or suppress) ISI. The $H_T(f)$, $H_C(f)$, and $H_R(f)$ filters are represented by an equivalent digital LTI filters.

Discrete-time linear filter model



Usually, the equalizer consists of two parts a *noise-whitening filter* $H_W(z)$ and an equalizing filter $H_E(z)$. The effect of $H_W(z)$ is to "whiten" the noise sequence so that the noise samples are uncorrelated.

We can reorganize the blocks to get the linear model suitable for analysis.



It is obvious that $H_W(z)$ depends only on $H_R(z)$. Let

$$G(z) = \underbrace{H_T(z)\, H_C(z)\, H_R(z)}_{H(z)}\, H_W(z). \tag{4.8}$$

The communication system from the input to the noise whitening filter output can then be represented by the *discrete-time white-noise linear filter model.*

$$\tilde{I}_k = \sum_j I_j g_{k-j} + n_k = I_k g_0 + \sum_{j \neq k} I_j g_{k-j} + n_k$$

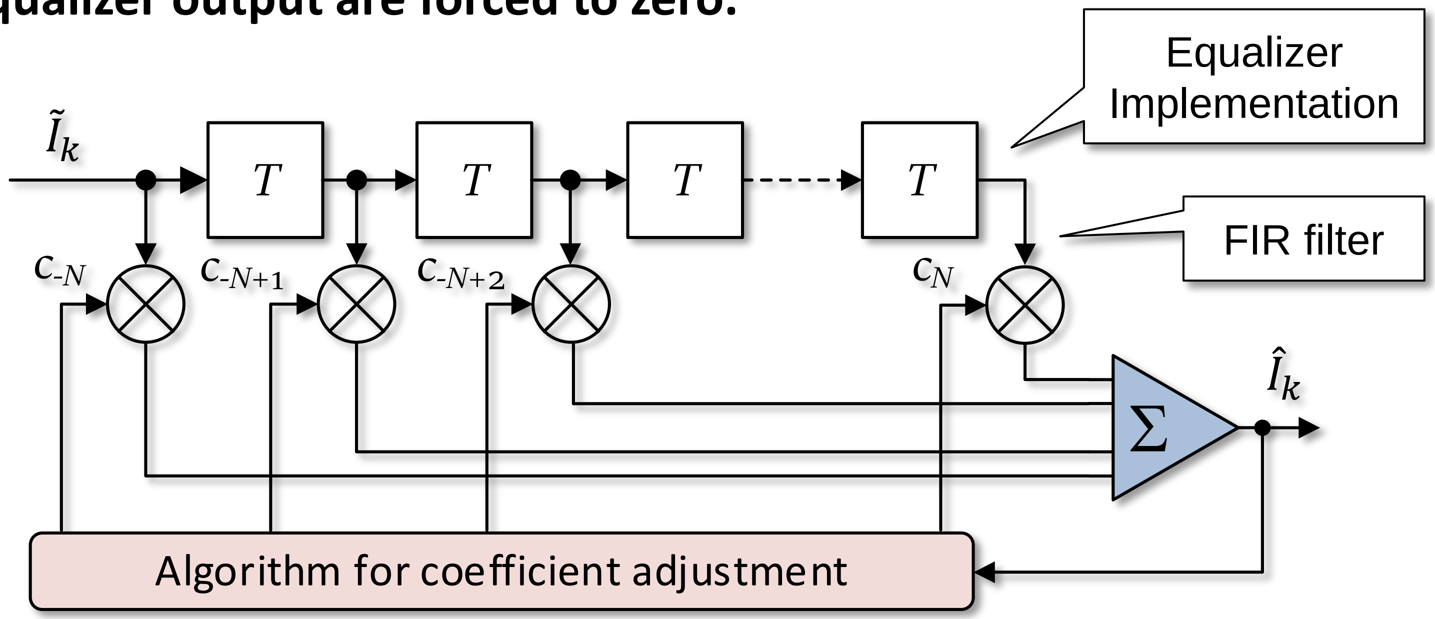
where $\{g_k\}$ is the impulse response corresponding to the transfer function $G(z)$, and $\{n_k\}$ is an AWGN sequence. The equalizer output sequence is given by

$$\hat{I}_k = \sum_j \tilde{I}_{k-j} h_{Ej} \tag{4.9}$$

Let the ISI be completely removed i.e. $\hat{I}_k = I_k$ **(while noise is not present) and in compliance with** (4.7) $H_T(z)\,H_R(z)\,H_W(z)$ **satisfy** (4.5)**. Then** $H_E(z)G(z) = 1$ **and**

$$\boxed{H_E(z) = \frac{1}{G(z)}}$$

This technique is known as *zero-forcing equalization* since the ISI components at the equalizer output are forced to zero.



For $(2N+1)$ filter taps we generally get

$$\hat{I}_k = \sum_{j=-N}^{N} \tilde{I}_{k-j} c_j \qquad k = -2N, \cdots, 2N \tag{4.10}$$

In matrix notation we get $\hat{\mathbf{I}} = \tilde{\mathbf{I}} \cdot \mathbf{c}$. (4.11)

For the *zero–forcing equalization* the input $\tilde{\mathbf{I}}$ matrix is a $(2N+1) \times (2N+1)$ square matrix.

$$\tilde{\mathbf{I}} = \begin{bmatrix} \tilde{I}_0 & \tilde{I}_{-1} & \cdots & \tilde{I}_{-2N} \\ \tilde{I}_1 & \tilde{I}_0 & \cdots & \tilde{I}_{-2N+1} \\ \vdots & \vdots & & \vdots \\ \tilde{I}_{2N} & \tilde{I}_{2N-1} & \cdots & \tilde{I}_0 \end{bmatrix}$$

Because $H_E(z)G(z) = 1$, in the time domain we can write

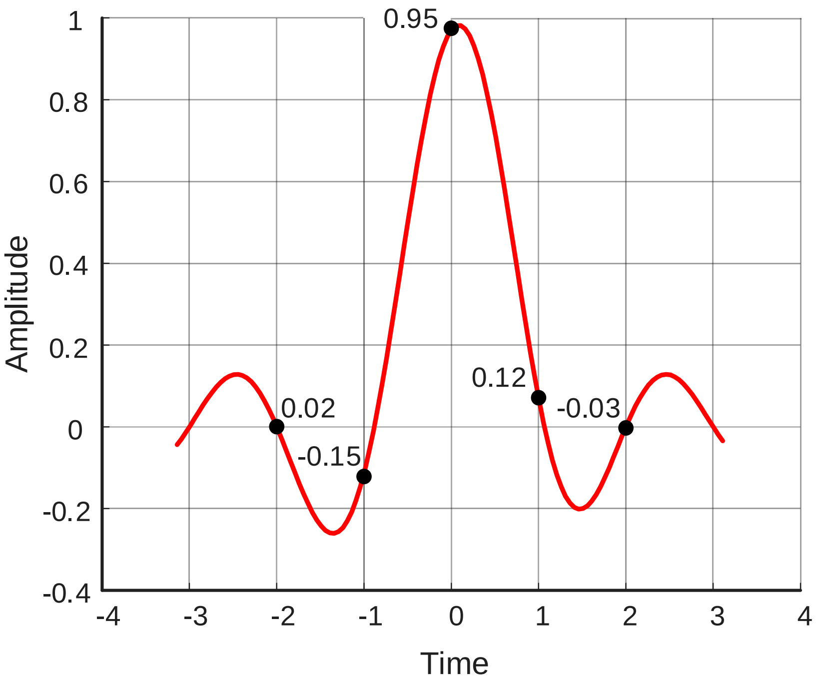
$$\sum_{j=-N}^{N} h_e(j) g(k-j) = \delta(k)$$

Then

$$\hat{I}_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k = \pm 1, \pm 2, \dots \end{cases}$$

Example 4.1 Let the distorted set of pulse samples shown in figure be received and the equalizer circuit be made up of three taps ($N = 1$).

Find the weights $\{c_{-1}, c_0, c_1\}$.



$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{I}_0 & \tilde{I}_{-1} & \tilde{I}_{-2} \\ \tilde{I}_1 & \tilde{I}_0 & \tilde{I}_{-1} \\ \tilde{I}_2 & \tilde{I}_1 & \tilde{I}_0 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.95 & -0.15 & 0.02 \\ 0.12 & 0.95 & -0.15 \\ -0.03 & 0.12 & 0.95 \end{bmatrix} \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} \Rightarrow \begin{bmatrix} c_{-1} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 0.163 \\ 1.013 \\ -0.123 \end{bmatrix}$$

After application (4.10) **we get** $\tilde{I}_{-1} = 0, \tilde{I}_0 = 1, \tilde{I}_1 = 0$ **as shown in m file** $(\tilde{I} = x, \hat{I} = y)$. **Note that** (4.10) **represents a linear convolution.**

```
x = [0.02   -0.15    0.95    0.12   -0.03];
Ik = [0 1 0]';              % required symbol samples
B = [x(3)  x(2)  x(1); x(4)  x(3)  x(2); x(5)  x(4)  x(3)];
C = linsolve(B, Ik);        % solving a linear system of equations
y = conv(C, x);             % linear convolution

C =
    0.1625
    1.0127
   -0.1228

y =
    0.0032   -0.0041    0.0000    1.0000    0.0000   -0.0451    0.0037
```

$\hat{I}_k$

The ZF equalizer removes ISI but it does not take into account noise in the system $\Rightarrow$ it may not give the best error performance.

The *minimum mean square error* (MMSE) *equalizer* assume presence of noise. Thus, each symbol I_k is modeled as a random variable and the information sequence $\{I_k\}$ as WSS. We want to minimize the MSE between $\hat{I}_k$ and I_k, i.e.

$$\mathrm{MSE} = \mathrm{E}\left[\left(I_k - \hat{I}_k\right)^2\right]$$

Using (4.8) **we get**

$$\mathrm{MSE} = \mathrm{E}\left[\left(I_k - \sum_{j=-N}^{N} \tilde{I}_{k-j} h_{Ej}\right)^2\right] = \mathrm{E}\left[\left(I_k - \tilde{\mathbf{I}}_k^T \mathbf{h}_E\right)^2\right], \tag{4.12}$$

where

$$\tilde{\mathbf{I}}_k = [\tilde{I}_{k+N}, \cdots, \tilde{I}_{k-N}], \quad \mathbf{h}_E = \left[h_{E,-N}, \cdots, h_{E,N}\right] \tag{4.13}$$

Differentiating with respect to each h_{Ej} and setting the result to zero, we get

$$-2\mathrm{E}\left[\tilde{\mathbf{I}}_k\left(I_k - \tilde{\mathbf{I}}_k^T \mathbf{h}_E\right)\right] = -2(\mathbf{R}_{\tilde{I}I} - \mathbf{R}_{\tilde{I}\tilde{I}}\mathbf{h}_E) = 0 \quad (4.14)$$

where $\mathbf{R}_{\tilde{I}\tilde{I}} = E\left[\tilde{\mathbf{I}}_k \tilde{\mathbf{I}}_k^T\right]$ and $\mathbf{R}_{\tilde{I}I} = E[I_k \tilde{\mathbf{I}}_k]$ are the autocorrelation matrix and cross-correlation vector, respectively. After simple rearranging, we get

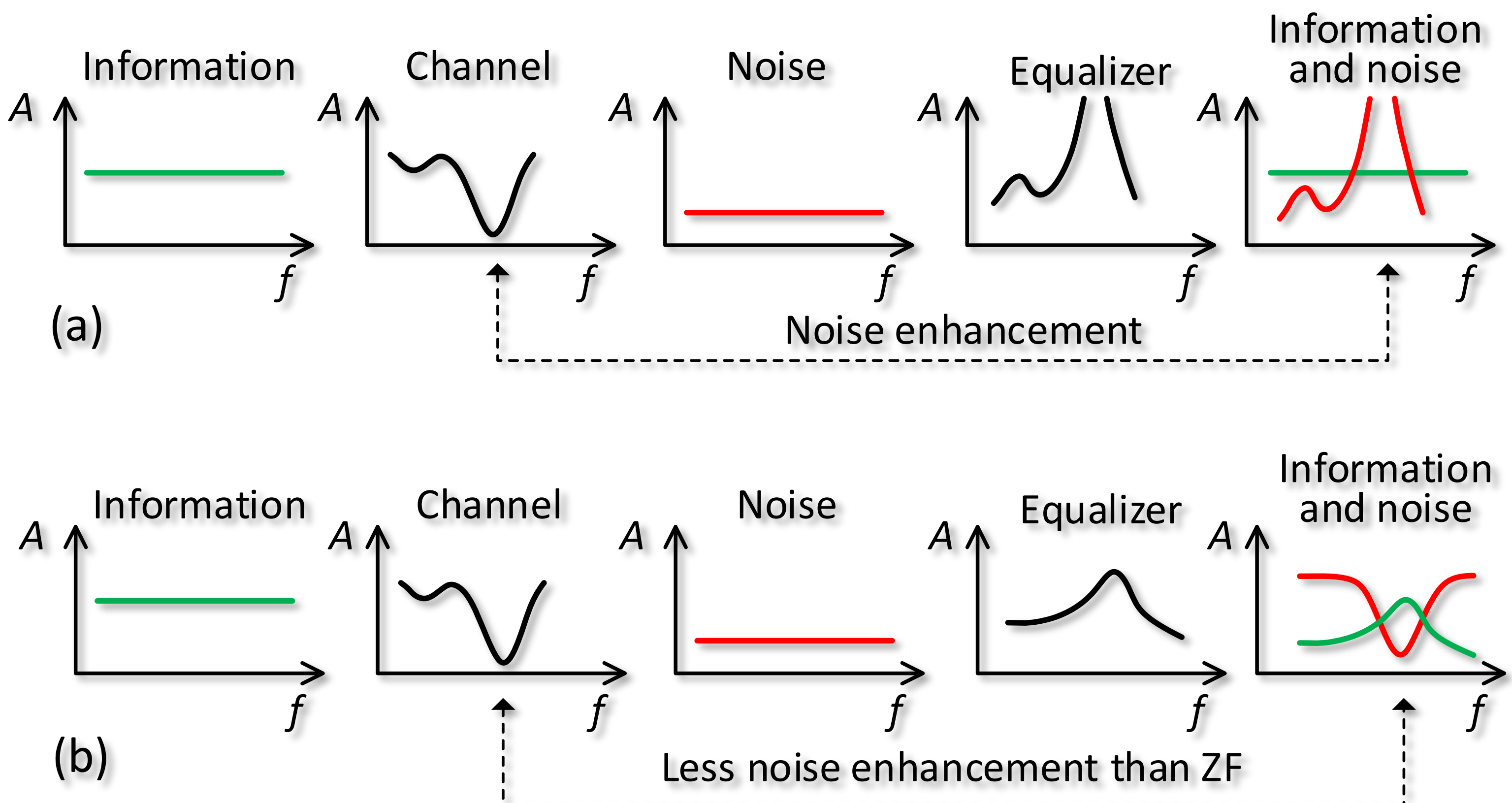
$$\mathbf{R}_{\tilde{I}\tilde{I}}\mathbf{h}_E = \mathbf{R}_{\tilde{I}I}, \quad (4.15)$$

Finally, the filter coefficients are given by

$$\mathbf{h}_E = \mathbf{R}_{\tilde{I}\tilde{I}}^{-1}\mathbf{R}_{\tilde{I}I},$$

It can be shown that the SNR at the MMSE equalizer output is better than that of the zero-forcing equalizer.

The ZF equalizer enforces a completely flat (constant) transfer function (a) ⇒ strong amplification at frequencies with small transfer ⇒ noise enhancement. The noise power of an MMSE equalizer is smaller than that of a ZF equalizer (b).



Least mean square **(LMS)** ***equalizer*** **results from iterative MMSE approach. Recall from** (4.14) **to get the MSE gradient**

$$-2\mathrm{E}\left[\tilde{\mathbf{I}}_k\left(I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E\right)\right] = -2\mathrm{E}[\tilde{\mathbf{I}}_k e_k] = -2\mathbf{R}_{\tilde{I}e}$$

where $e_k = \left(I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E\right)$ **is an** ***equalization error.*** **The MSE gradient gives the direction to change** $\mathbf{h}_E$ **for the largest increase of the MSE. To decrease the MSE** $\mathbf{h}_E$ **must be updated in the direction opposite to the gradient.**

$$\mathbf{h}_E(k) = \mathbf{h}_E(k-1) + \mu\mathbf{R}_{\tilde{I}e}, \tag{4.16}$$

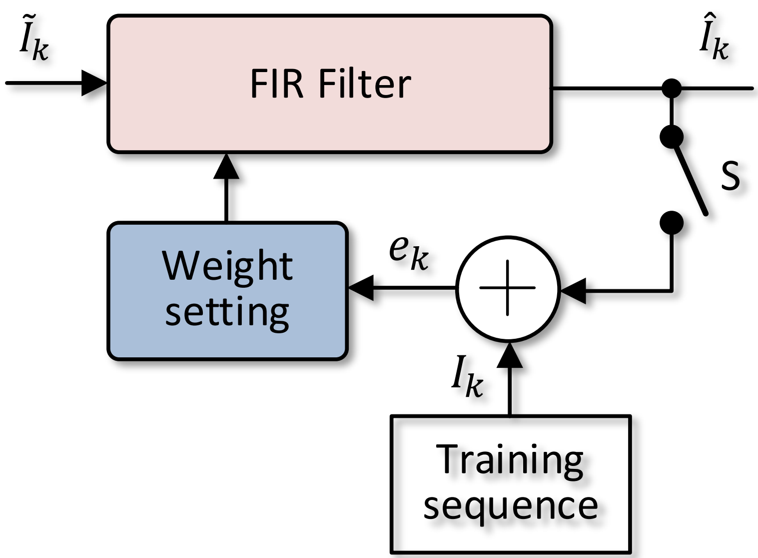
where μ **is a small positive constant that controls the rate of convergence. Note that in the iterative approach each symbol is equalized by using previous filter coefficients i.e.** $e_k = \left[I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E(k-1)\right]$**.**

To simplify the calculation, the instantaneous estimate of the cross-correlation $\mathbf{R}_{\tilde{I}e} \approx \tilde{I}_k e_k$ **may be used. Then**

$$\mathbf{h}_E(k) = \mathbf{h}_E(k-1) + \mu\tilde{I}_k e_k$$

As $\tilde{I}_k$ is unknown at the receiver, the transmitter transmits a training sequence that is known a priori by the receiver (it is stored in memory) and can be used for e_k estimation.

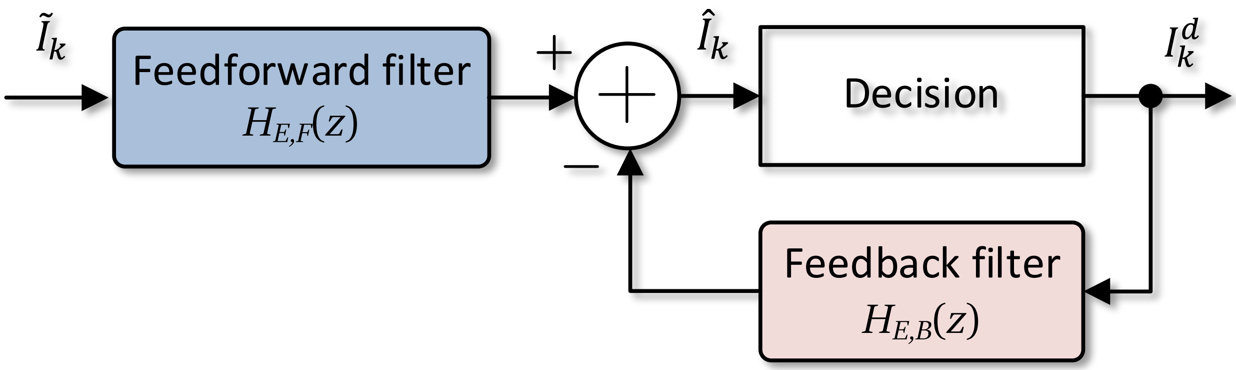
LMS equalizer implementation



***Training mode*: switch S is on. The received training sequence $\hat{I}_k$ is compared with an identical sequence I_k generated by receiver. The error signal e_k is continuously calculated, and the filter weights are adjusted.**

***Data mode:* switch S is OFF and the equalized signal is fetched from output.**

The DFE determines the ISI contribution from detected symbols I_k^d by passing them through a *feedback filter* $H_{E,B}(z)$ that approximates the channel $G(z)$ convolved with the *feedforward filter* $H_{E,F}(z)$. The resulting ISI is then subtracted from the incoming symbols.



Assuming that $H_{E,F}(z)$ has N_1+1 taps and $H_{E,F}(z)$ has N_2 taps. We can write the DFE output as

$$\hat{I}_k = \sum_{j=-N_1}^{0} \tilde{I}_{k-j} h_{Ej} - \sum_{j=1}^{N_2} I_{k-j}^d h_{Ej} \qquad (4.17)$$

It is obvious that $\mathbf{h}_E = \left[h_{E,F,-N_1}, \cdots, h_{E,F,0}, h_{E,B,1}, \cdots, h_{E,B,N_2}\right]$**.**

Note that in the case of ZF we get zero ISI and in (4.17) $\hat{I}_k$ **is equal to the second term** $\Rightarrow$ **ZF approach is not suitable for the filter coefficients calculation** $\Rightarrow$ **LMS algorithm is used bellow.**

Adaptation of the LMS coefficients

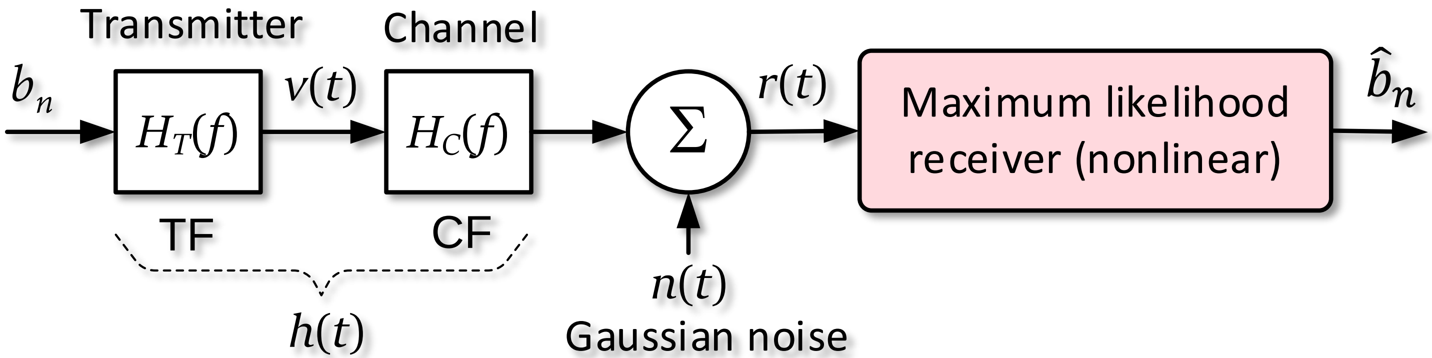
$$\mathbf{h}_{E,F}(k) = \mathbf{h}_{E,F}(k-1) + \mu \tilde{I}_k e_k \quad \textbf{and} \quad \mathbf{h}_{E,B}(k) = \mathbf{h}_{E,B}(k-1) + \mu I_k^d e_k,$$

where during the training phase $e_k = \left(I_k - \hat{I}_k\right)$. (4.18)

The elimination of ISI, and making a decision on the current symbol based on the equalizer output is simple and practical, but we have no idea whether such approach is optimal in terms of minimizing the average symbol error probability $\Rightarrow$

- We will design a receiver which decides the whole transmitted symbol sequence simultaneously from the received signal.
- We will be aimed at *minimizing the probability of choosing the wrong sequence* of symbols instead of the average symbol error probability.
- We will employ the ML principle to achieve our goal.
- All compensation for channel distortion is to be done at the receiver referred to as the *maximum likelihood sequence estimation* (MLSE) receiver

We will demonstrate the MLSE technique by derivation of the MLSE receiver for a signal set consisting of PAM waveforms.



The transmitted $v(t)$ and the received $r(t)$ PAM signals are given by (4.1), where

$$r(t) = \sum_{n=-\infty}^{\infty} b_n h(t - nT) + n(t)$$

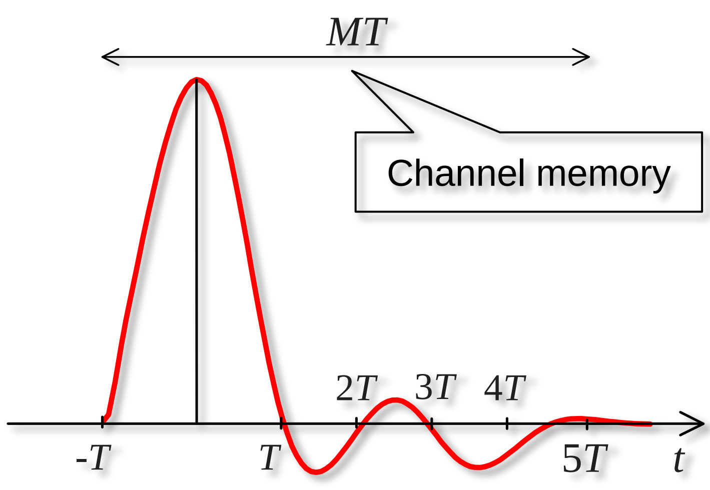
and $n(t)$ is a Gaussian process but not necessarily white. Further, we will make the following assumptions:

- **The dispersion (*memory*), of the channel is limited to a finite time MT.**
- **The transmitter is turned on at some arbitrary time $N_1 T$ and off at a later time $N_2 T$ so that $(N_2 - N_1) >> M$.**

The channel memory causes the latest signal energy arrival at the time $t = N_2 T + MT$.

Hence, we will observe the received signal $r(t)$ on the time interval $N_1 T \leq t \leq N_2 T + MT$.

The transmitted sequence is $\{b_n(t)\}$, $N_1 \leq n \leq N_2$.



The receiver produces a decision sequence $\hat{b}_{N_1}, \hat{b}_{N_1+1}, \cdots, \hat{b}_{N_2}$ which maximize *a posteriori* probability of a data sequence $\Rightarrow$ the probability to be evaluated is

$$\Pr\left[b_{N_1}, \cdots, b_{N_2} | r(t)\right], \text{ where } N_1 T \leq t \leq N_2 T + MT$$

By using Bayes's rule we have

$$\Pr\left[b_{N_1},\cdots,b_{N_2}|r(t)\right]=\Pr\left[r(t)|b_{N_1},\cdots,b_{N_2}\right]\times\frac{\Pr\left(b_{N_1},\cdots,b_{N_2}\right)}{\Pr[r(t)]} \qquad (4.19)$$

Because all the symbols are assumed equiprobable, $\Pr\left(b_{N_1},\cdots,b_{N_2}\right)$ **is independent of the data sequence, as is the denominator probability density. The maximization of** (4.19) **is equivalent to maximization of likelihood function**

$$\Pr\left[r(t)|b_{N_1},\cdots,b_{N_2}\right]=\frac{1}{(2\pi N_0)^{k/2}}\exp\left\{-\frac{1}{2N_0}\int_{-\infty}^{\infty}\left[r(t)-\sum_n b_n h(t-nT)\right]^2 dt\right\} \qquad (4.20)$$

where $K=(N_2-N_1)$**. The ML sequence estimate of** $b_{N_1},\cdots,b_{N_2}$ **is that data sequence minimizing** (4.20).

Using the bit sequence notation $\mathbf{b}_N = \left[b_{N_1}, \cdots, b_{N_2}\right]$ we can minimize the log-likelihood function $\ell\left(\mathbf{b}_N\right)$ by

$$\hat{\mathbf{b}}_N = \arg\min_{\mathbf{b}_N} \int_{-\infty}^{\infty} \left[r(t) - \sum_{n=N_1}^{N_2} b_n h(t-nT)\right]^2 dt$$

$$= \arg\max_{\mathbf{b}_N} \left\{2\int_{-\infty}^{\infty} r(t) \sum_{n=N_1}^{N_2} b_n h(t-nT)\, dt - \int_{-\infty}^{\infty} \left[\sum_{n=N_1}^{N_2} b_n h(t-nT)\right]^2 dt\right\}$$

$$= \arg\max_{\mathbf{b}_N} \left[2\sum_{n=N_1}^{N_2} b_n y_n - \sum_{n=N_1}^{N_2} \sum_{m=N_1}^{N_2} b_n b_m x_{n-m}\right] \qquad (4.21)$$

where y_n is the *matched filter* output. Assuming the above time interval $N_1 T \leq t \leq N_2 T + MT$ we can write

$$y_n = \int_{N_1T}^{N_2T+MT} r(t)h(t-nT)dt = \sum_{k=N_1}^{N_2} b_k x_{n-k} + \xi_n \qquad (4.22)$$

where ξ_n is the noise at the matched filter output, and

$$x_{n-m} = \int_{N_1T}^{N_2T+MT} h(t-nT)h(t-mT)dt, \qquad N_1 \le n, m \le N_2 \qquad (4.23)$$

It is obvious that

- **The term x_0 is the energy of the pulse at the *matched filter* output.**
- **The finite channel memory implies that $x_{n-m} = 0$ for $|n-m| > M$.**
- **The Gaussian noise variables ξ_n are generally correlated but can be changed to uncorrelated by a whitening filter.**
- **The processing of y_n requires the knowledge of the channel.**

Viterbi algorithm for the Nyquist channel

Let $h(t)$ correspond to SRC shaping of a Nyquist pulse. Then $|H(f)|^2$ is a Nyquist spectrum. The complementary SRC shaping will be provided by the matched filter at the receiver. From (4.23) we can get

$$x_k = \int_{-\infty}^{\infty} h(t)h(t-kT)dt = \int_{-\infty}^{\infty} |H(f)|^2 e^{-j2\pi fkT} dt$$

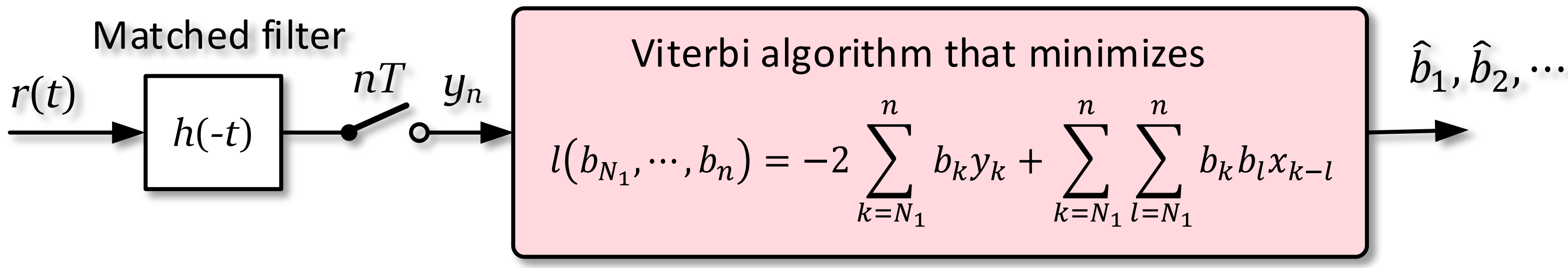
Because $|H(f)|^2$ is the transform of a Nyquist pulse, in time domain we have (see (4.3))

$$x_k = E_s\delta_k,$$

where $E_s = \int_{-\infty}^{\infty} h^2(t)$ is the pulse energy.

4. Equalization

Maximum likelihood sequence estimation



Since x_k is nonzero only for $k = 0$ the log-likelihood (4.21) **becomes**

$$l(\mathbf{b}_n) = 2\sum_{k=N_1}^{n} b_k y_k + E_s \sum_{k=N_1}^{n} b_k^2 = E_s \sum_{k=N_1}^{n} \left[\left(b_k - \frac{y_k}{E_s} \right)^2 - \frac{y_k^2}{E_s^2} \right]$$

Optimization can by done in *symbol-by-symbol* manner by choosing the quantized or sliced signal b_k closest to the scaled received sample y_k / E_s.

$$l(\mathbf{b}_n) = l(\mathbf{b}_{n-1}) - 2b_n y_n + 2b_n \sum_{k=n-M}^{n-1} b_k x_{n-k} + b_n^2 x_0$$

1. **Gitlin, R.D., Hayes, J. F., Weinstein S, B. *Data communication principles*. Plenum press, New York, 1992**

Channel coding can generally be understood as the transformation of a "useful" signal performed in order to eliminate errors caused by channel impairments.

The main causes of errors:

1. presence of interfering signals in the transmission channel (various types of noise in a channel and in a receiver, industrial interference, etc.)
2. Non-flat channel frequency response (narrow bandwidth)
3. Nonlinear behavior of electronic circuits (nonlinear power stage of the transmitter, modulator or demodulator).
4. Imperfect synchronization in the receiver

Waveform coding transforms a waveform set (representing a message) into an improved waveform set using *orthogonal* or *biorthogonal codes* to provide better bit error probability P_b.

Orthogonal codes

let M is the sequence length then waveform sets s_i and s_j are orthogonal if

$$z_{ij} = \sum_{k=0}^{M-1} s_i(kT)\, s_j(kT) = \begin{cases} 1 & \text{for} \quad i = j \\ 0 & \text{for} \quad i \neq i \end{cases}$$

where $i, j = 1,2,\ldots,M$, or using number of digit agreements N_A, number of digit disagreements N_D, and total number of digits N_T:

$$z_{ij} = \frac{N_A - N_D}{N_T} = \begin{cases} 1 & \text{for} \quad i = j \\ 0 & \text{for} \quad i \neq i \end{cases}$$

A one-bit data set can be transformed, using *orthogonal codewords* of two digits each as follows

Data set	Orthogonal codeword set
0 1	$\boldsymbol{H}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$

Hadamard matrix

To encode a 2-bit data set, we extend the foregoing set as follows

Data set	Orthogonal codeword set
0 0 0 1 --------- 1 0 1 1	$\boldsymbol{H}_2 = \left[\begin{array}{cc:cc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ \hdashline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right] = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_1 \\ \boldsymbol{H}_1 & \overline{\boldsymbol{H}_1} \end{bmatrix}$

To encode a k-bit **data set we can write:** $\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{H}_{k-1} & \boldsymbol{H}_{k-1} \\ \boldsymbol{H}_{k-1} & \overline{\boldsymbol{H}_{k-1}} \end{bmatrix}$

For equal-energy orthogonal signals the probability of codeword (symbol) error, can be upper bounded as

$$P_e(M) \leq (M-1)Q\left(\sqrt{\frac{E_s}{N_0}}\right)$$

To get bit error probability we can use

$$\frac{P_b(M)}{P_e(M)} = \frac{M}{2(M-1)} = \frac{2^{k-1}}{2^k - 1}$$

Where k is the number of data bits per codeword.

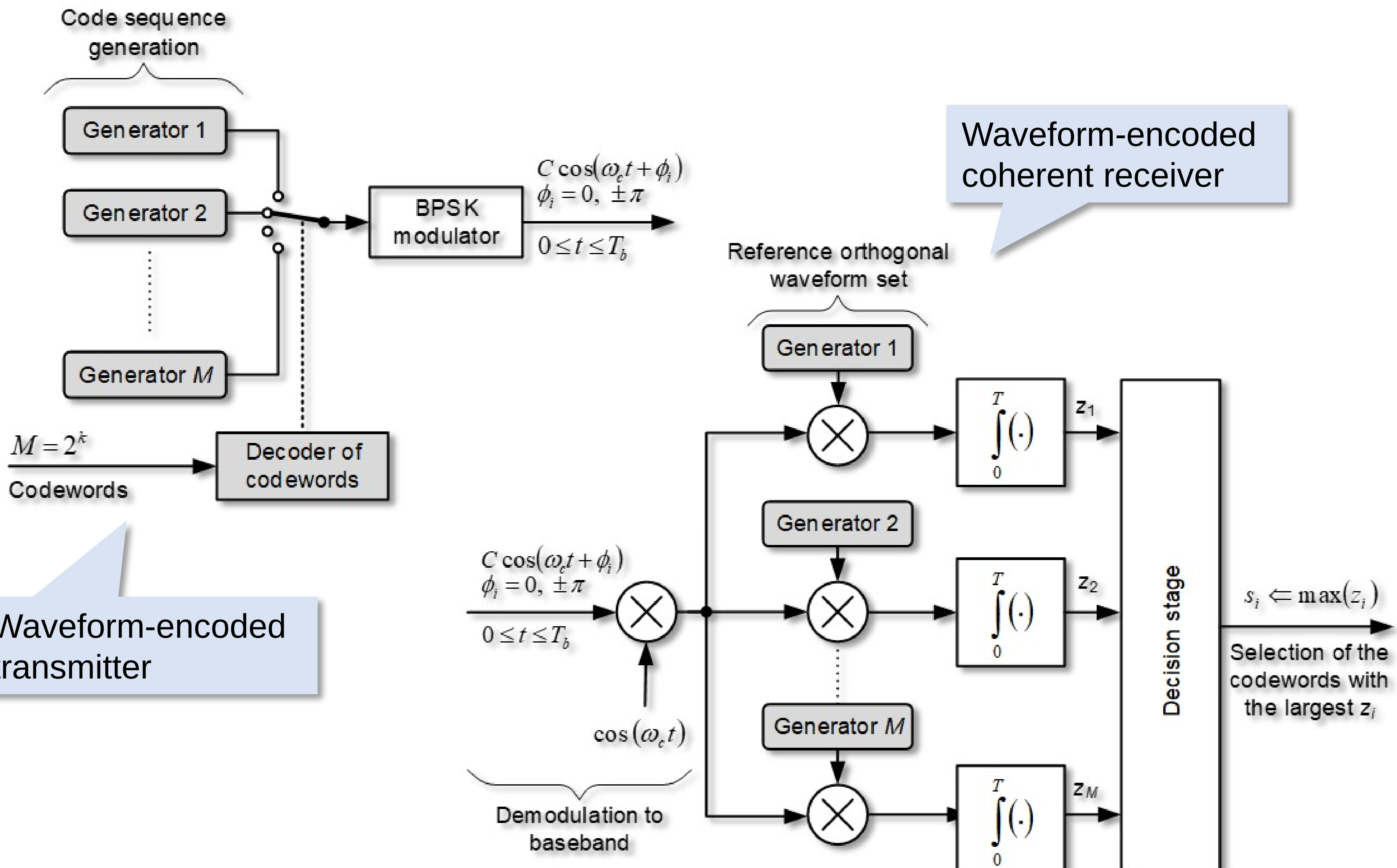
Orthogonal codes

A biorthogonal signal set of M total signals can be obtained from an orthogonal set of $M/2$ signals by augmenting it with the negative of each signal

$$B_k = \begin{bmatrix} H_{k-1} \\ \overline{H}_{k-1} \end{bmatrix} \quad \text{and} \quad P_e(M) \leq (M-2)Q\left(\sqrt{\frac{E_s}{N_0}}\right) + Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

5. Channel coding

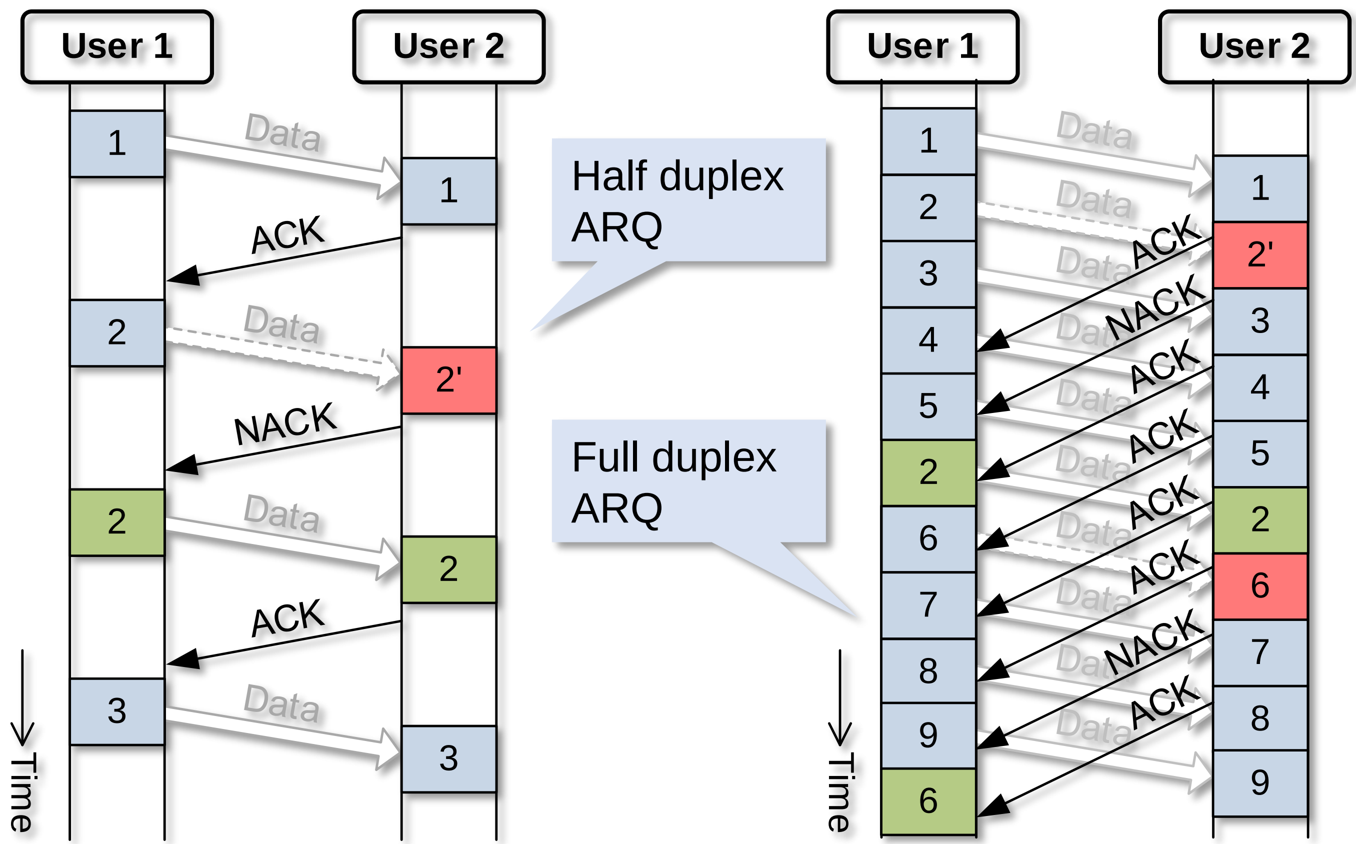
Waveform coding



Waveform-encoded transmitter

Waveform-encoded coherent receiver

In case the error control consists of *error detection only*, the communication system needs to provide a means of alerting the transmitter that an error has been detected and that a *retransmission* is necessary ⇒ ARQ



- **Introduced by Gallager in his PhD thesis in 1960**
- **Class of linear block codes**
- **Parity-check matrix contains only a few 1's**
- **Suited for implementations that make heavy use of parallelism**

Representations for LDPC codes

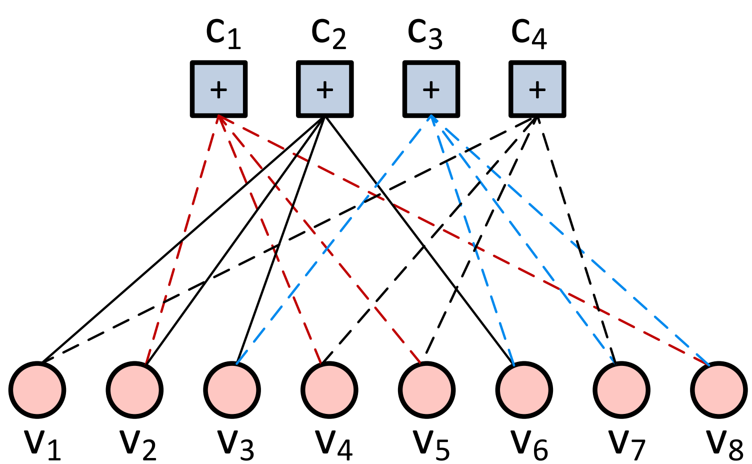
1. **Matrix representation**
2. **Graphical representation**

As for 1: parity check matrix with dimension $n \times m$. The number of 1's in each row and column must satisfy the conditions $w_r \ll m$ and $w_c \ll n$.
Example for a (8, 4) code:

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

As for 2: Effective graphical representation for LDPC codes is known as a ***Tanner graph***. It

- provides a complete representation of the code,
- helps to describe the decoding algorithm, and
- contains the two types of nodes called variable nodes (*v-nodes*) and check nodes (*c-nodes*).



$$H = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ \hline & 1 & & 1 & 1 & & & 1 \\ 1 & 1 & 1 & & & 1 & & \\ & & 1 & & & 1 & 1 & 1 \\ 1 & & & 1 & 1 & & 1 & \end{array}$$

Types of LDPC codes

- **Regular codes**: w_c is constant for every column and $w_r = w_c(n/m)$ is also constant for every row, i.e. there is the same number of incoming edges for every *v-node* and also for all the *c-nodes.*
- **Irregular codes**: the numbers of 1's in each row or column aren't constant.

Performance and complexity of LDPC codes

- For large block lengths they approach the Shannon limit (e.g. an irregular code performs 0.04 dB of the Shannon limit at a bit error rate of 10^{-6} with a block length of 10^7).
- Encoding: it is performed in blocks of size $\mathbf{m}$, $\mathbf{y} = \mathbf{mG} = \mathbf{m}[\mathbf{I_k}|\mathbf{P}] = [\mathbf{m}|\mathbf{mP}]$,

  where $\mathbf{I}$ is a unit matrix, $\mathbf{m} = [m_1, m_2, \ldots, m_k]$, and $\mathbf{y} = [y_1, y_2, \ldots, y_n]$.

  The generator matrix $\mathbf{G} = [\mathbf{I_k}|\mathbf{P}]$ can be obtained from the parity check matrix $\mathbf{H} = [\mathbf{P^T}|\mathbf{I_{n\text{-}k}}]$ via Gaussian elimination. The sub-matrix $\mathbf{P}$ is generally not sparse. The matrix $\mathbf{H}$ is created by a random construction, a geometric construction, or by means of a Gallager parity matrix, etc.

Parity check matrix random construction

- **Each row of the matrix must contain a constant number w_r of 1's,**
- **Each column of the matrix must contain a constant number w_c of 1's**
- **The matrix must be sparse,**
- **Any two rows in the matrix must be linearly independent,**
- **There must be no cycles in the matrix (see simple loop).**

Simple loop

1				1					1					1	
		1				1						1			1
			1					1		1			1		
1				1			1						1		
					1					1				1	1
	1		1			1		1							
				1		1			1			1			
		1						1			1				1
1					1								1		
	1		1				1					1			
	1				1				1		1				
		1					1				1			1	

Gallager's construction of parity check matrix

- It is based on individual submatrices.

$$\mathbf{H} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} \tag{5.5}$$

- H_1 has a constant number w_r of 1's and w_c = 1. They are distributed as follows:

$$H_1 = \begin{bmatrix} \underbrace{11\cdots1}_{w_r} & 00\cdots0 & \cdots & 00\cdots0 \\ 00\cdots0 & \underbrace{11\cdots1}_{w_r} & & \\ \vdots & & \ddots & \vdots \\ 00\cdots0 & & \cdots & \underbrace{11\cdots1}_{w_r} \end{bmatrix} \tag{5.6}$$

- There are no two lines having any common element and two columns having more than one 1.
- Other submatrices are created from H_1 by **permutation of the columns**.
- Matrix $\mathbf{H}$ defines a (w_c, w_r) regular parity check code where $w_c = N$.

Example 5.1: regular LDPC code (3, 4).

It can be shown that it contains 13 linearly independent rows. The dimension of the code is 20 - 13 = 7, $\Rightarrow$ we have a (20,7) code with actual rate R = 0.35.

$$
\mathbf{H} = \begin{array}{cc}
\left[\begin{array}{cccccccccccccccccccc}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
1&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1
\end{array}\right]
&
\begin{array}{c}
\\ \\ H_1 \\ \\ \\ \\ \\ H_2 \\ \\ \\ \\ \\ H_3 \\ \\ \\
\end{array}
\end{array}
$$

Geometrical construction of parity check matrix

- **It is based on m-dimensional** *Euclidean geometry* **(EG) over** *Galois field* GF(2^s)
- **The LDPC are of the** EG(m, 2^s) **type, where** $m = 2$ **(for the 2-D space) and EG contains** 2^{ms} points. **The code has then the following parameters**:
  - ✓ Code length: $n = 2^{2s} - 1$
  - ✓ Number of parity bits: $n - k = 3^s - 1$
  - ✓ Number of message bits: $k = 2^{2s} - 3^s$
  - ✓ Minimum Hamming distance: $d_{min} = 2^s$
  - ✓ Sparsity: $k = 2^s/n$
- **The goal to create a vector** v **and by a cyclic shift of this vector find the square matrix** $\mathbf{H}$. **For this purpose we will use the** *finite field arithmetics* [5.1]

s	n	k	d_{min}	$w_r = w_c$	r
2	15	7	5	4	0,267
3	63	37	9	8	0,127
4	255	175	17	16	0,0627
5	1023	781	33	32	0,0313
6	4095	3367	65	64	0,01563

Example 5.2 for $s = 2$**:** because $m = 2$, we have EG(2, 2^2) over GF(2^2), thus we use the extended finite element array GF($2^{2\cdot 2}$).

The GF(2^4) is generated by the primitive polynomial (irreducible polynomial that can be used to generate a finite field) $\mathrm{g}(x) = 1 + x + x^4$.

1. We must find vector $\boldsymbol{v}$, containing $w_r = w_c = 4$ ones. The length of the vector must be equal to the length of the code ($n = 15$).
2. We determine the distribution of 1's in the vector. The positionS of 1's are given by $\{\alpha^{14} - \eta\alpha\}$, where $\eta \in \mathrm{GF}(2^2)$.

$g(\alpha) = 1 + \alpha + \alpha^4 = 0 \Rightarrow \boldsymbol{\alpha^4 = 1 + \alpha}$, $\alpha^5 = \alpha(1 + \alpha) \Rightarrow \boldsymbol{\alpha^5 = \alpha^2 + \alpha}, \cdots$

Possible representations of polynomial roots

Power r.	Polynomial r.	4-Tuple r.	Power r.	Polynomial r.	4-Tuple r.
0	0	(0000)	α^7	$1+\alpha+\alpha^3$	(1101)
1	1	(1000)	α^8	$1+\alpha^2$	(1010)
α	α	(0100)	α^9	$\alpha+\alpha^3$	(0101)
α^2	α^2	(0010)	α^{10}	$1+\alpha+\alpha^2$	(1110)
α^3	α^3	(0001)	α^{11}	$\alpha+\alpha^2+\alpha^3$	(0111)
α^4	$1+\alpha$	(1100)	α^{12}	$1+\alpha+\alpha^2+\alpha^3$	(1111)
α^5	$\alpha+\alpha^2$	(0110)	α^{13}	$1+\alpha^2+\alpha^3$	(1011)
α^6	$\alpha^2+\alpha^3$	(0011)	α^{14}	$1+\alpha^3$	(1001)

The elements of the finite field GF (4) are $\{0,\ 1,\ \cdots,\beta^{2^s-2}\} = \{0,\ 1, \beta, \beta^2\}$, where

$$\beta = \alpha^{\frac{2^{2s}-1}{2^s-1}} = \alpha^5$$

Element of an extension field GF(2^s)

Element of an extension field GF(2^{2s})

then $\{0,\ 1, \beta, \beta^2\} = \{0,\ 1, \alpha^5, \alpha^{10}\}$.

$\{\alpha^{14} - 0\alpha\} = \{1001 + 0000\} = \alpha^{14}$
$\{\alpha^{14} - 1\alpha\} = \{1001 + 0100\} = \{1101\} = \alpha^7$
$\{\alpha^{14} - \alpha^5\alpha\} = \{1001 + 0011\} = \{1010\} = \alpha^8$
$\{\alpha^{14} - \alpha^{10}\alpha\} = \{1001 + 0111\} = \{1110\} = \alpha^{10}$

***v* = (000000011010001)**

α^7 α^8 α^{10} α^{14}

3. We can now create the parity check matrix $\mathbf{H}$ by a cyclic shift of the vector ***v***.

$$H_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ \vdots & & & \cdots & & & & \ddots & & & & \cdots & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.7)$$

Construction of EG-LDPC Generator matrix

Let h be an nonnegative integer less than 2^{2s}. Then h can be expressed in radix-2^s form:

$$h = \delta_0 + \delta_1 2^s + \cdots + \delta_{m-1} 2^{(m-1)s},$$

where $0 \leq \delta_0, \cdots, \delta_{m-1} < 2^s$. The 2^s-weight of h, denoted $W_{2^s}(h)$ is given by

$$W_{2^s}(h) = \delta_0 + \delta_1 + \cdots + \delta_{m-1}. \tag{5.8}$$

For an nonnegative integer l, let $h^{(l)}$ be the remainder resulting from dividing $2^l h$ by $2^{ms} - 1$. Then $0 \leq h^{(l)} < 2^{ms} - 1$.

Let $g_{EG}(x)$ be the *generator polynomial* of the EG-LDPC code and let α be a primitive element of GF(2^{ms}). Then α^h is a root of $g_{EG}(x)$ if and only if

$$0 < \max_{0 \leq l < 2} W_{2^s}\left(h^{(l)}\right) \leq (m-1)(2^s - 1). \tag{5.9}$$

Then $g_{EG}(x)$ has the following sequence of consecutive powers of α,

$$\alpha, \alpha^2, \cdots, \alpha^{\frac{2^{ms}-1}{2^s-1}-1}$$

EG-LDPC Generator matrix is obtained by a cyclic shift of the generator polynomial.

The minimum distance of the *m*-dimensional EG-LDPC, code is lower bounded as follows

$$d_{min} \geq \frac{2^{ms} - 1}{2^s - 1} \tag{5.10}$$

To derive the generator matrix we need to apply some theorems:

Theorem 5.1. **if the element α is a root of $f(x)$, then for any $i > 0$, α^{2i} is also a root of $f(x)$. The element α^{2i} is called a *conjugate* of α.**

Theorem 5.2. **The minimal polynomial $\phi(x)$ of an element α in GF(2^s) divides $x^{2s} + x$.**

Theorem 5.3. **If $\alpha \neq 0$ is a root of $f(x)$, α^{-1} is a root of $g(x) = x^s f(x^{-1})$, the *reciprocal polynomial* of $f(x)$.**

Example 5.3 for $s = 2$ ***and*** $m = 2$**:** by substituting m and s into (5.7), (5.8), and (5.9) we get $h = \delta_0 + 4\delta_1$.

- The condition $0 < \max\limits_{0 \le l < 2} W_{2^s}\left(h^{(l)}\right) \le 3$ is satisfied for

$$h = [1, 2, 3, 4, 6, 8, 9, 12].$$

  For example $h = 6$ is obtained when $\delta_0 = 2$ and $\delta_1 = 1$ $(\delta_0 + \delta_1 = 3)$. The primitive elements [roots of $g_{EG}(x)$] then are: $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}$.

- We need to express α^h as a roots of prime polynomials in GF($2^{2\cdot 2}$) supposing that α is a root of $p(x) = 1 + x + x^4$

Minimal polynomials of the Elements in GF(16) generated by $p(x) = 1 + x + x^4$

	Conjugate roots	Minimal polynomials
1	0	x
2	1	$1 + x$
3	$\alpha, \alpha^2, \alpha^4, \alpha^8$	$1 + x + x^4$
4	$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$1 + x + x^2 + x^3 + x^4$
5	α^5, α^{10}	$1 + x + x^2$
6	$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	$1 + x^3 + x^4$

reciprocal polynomial:
$x^4(1 + x^{-1} + x^{-4}) = x^4 + x^3 + 1$

Hence $x^{16} + x = (1 + x + x^4)(1 + x^3 + x^4)(1 + x + x^2 + x^3 + x^4)(1 + x + x^2)(1 + x)\, x$

The elements α, α^2, α^4, α^8 have the same minimal polynomial $1 + x + x^4$, and the elements α^3, α^6, α^9, α^{12} have the same minimal polynomial $1 + x + x^2 + x^3 + x^4$. Thus, the generator polynomial of the EG code of length 15 is:

$$g_{EG}(x) = (1 + x + x^4)(1 + x + x^2 + x^3 + x^4) = 1 + x^4 + x^6 + x^7 + x^8$$

The generator matrix is then formed by the k - 1 cyclic shift of the $g_{EG}(x)$ and by a rearranging to a systematic form.

$$\mathbf{G} = \left|\begin{array}{ccccccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1
\end{array}\right|$$

Hard decision (HD) decoding

Based on the exchange of messages between *c-nodes* with *v-nodes* of the Tanner graph.

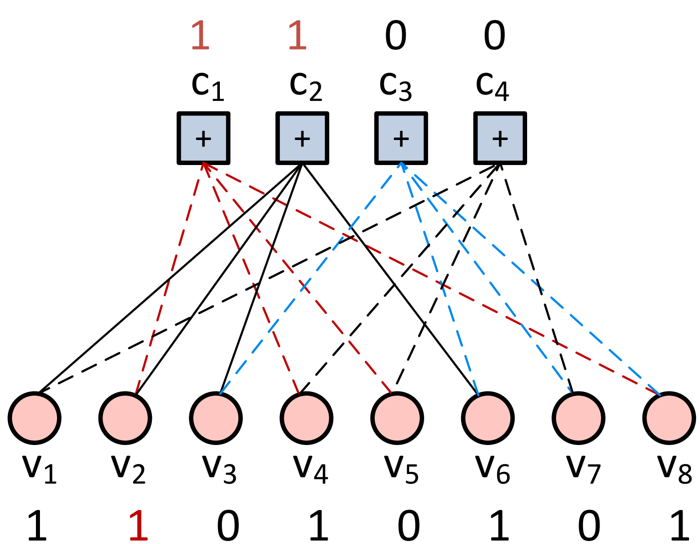
Decoding procedure:

1. *v-nodes* send their values to the corresponding *c-nodes*.
2. *c-nodes* perform parity check.
   - Check OK $\Rightarrow$ *c-nodes* return to the *v-nodes* the same value.
   - Wrong parity $\Rightarrow$ *c-nodes* return to the *v-nodes* an opposite value.
3. *v-nodes* update their values according to messages from the *c-nodes.*
4. Steps 1 to 3 are repeated.

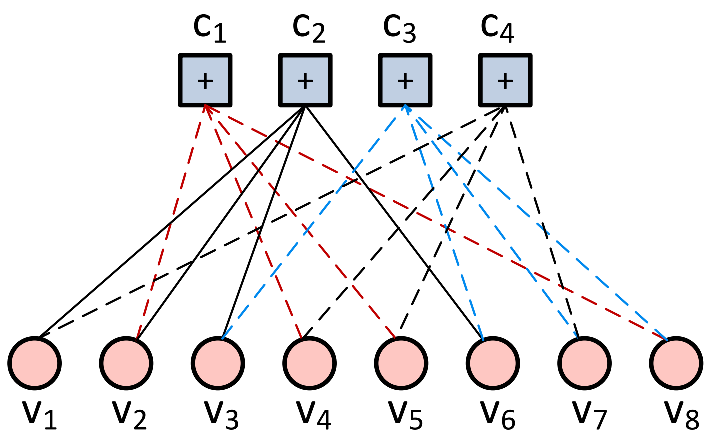
The **low internal complexity** of the LDPC decoder allows it to be used for high-speed applications. For example, 10 Gbit Ethernet (10GBASE-T), optional in Wi-Fi 802.11 standards (specifically 802.11n and 802.11ac), DVB-S2 and DVB-T2 standard.

***Example 5.4*:** Let the sequence **y** = [1 0 0 1 0 1 0 1] be sent and **y'**= [1 1 0 1 0 1 0 1] be received. Correct the sequence.

$$
H = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array}
\begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ \hline & 1 & & 1 & 1 & & & 1 \\ 1 & 1 & 1 & & & 1 & & \\ & & 1 & & & 1 & 1 & 1 \\ 1 & & & 1 & 1 & & 1 & \end{array}
$$



c_1	*received*	$v_2 \rightarrow 1$	$v_4 \rightarrow 1$	$v_5 \rightarrow 0$	$v_8 \rightarrow 1$
	sent	$0 \rightarrow v_2$	$0 \rightarrow v_4$	$1 \rightarrow v_5$	$0 \rightarrow v_8$
c_2	*received*	$v_1 \rightarrow 1$	$v_2 \rightarrow 1$	$v_3 \rightarrow 0$	$v_6 \rightarrow 1$
	sent	$0 \rightarrow v_1$	$0 \rightarrow v_2$	$1 \rightarrow v_3$	$0 \rightarrow v_6$
c_3	*received*	$v_3 \rightarrow 0$	$v_6 \rightarrow 1$	$v_7 \rightarrow 0$	$v_8 \rightarrow 1$
	sent	$0 \rightarrow v_3$	$1 \rightarrow v_6$	$0 \rightarrow v_7$	$1 \rightarrow v_8$
c_4	*received*	$v_1 \rightarrow 1$	$v_4 \rightarrow 1$	$v_5 \rightarrow 0$	$v_7 \rightarrow 0$
	sent	$1 \rightarrow v_1$	$1 \rightarrow v_4$	$0 \rightarrow v_5$	$0 \rightarrow v_7$



v-nodes	**y**	Messages from ***v-nodes***		New **y**
v_1	**1**	$c_2 \to 1$	$c_4 \to 1$	**1**
v_2	**1**	$c_1 \to 0$	$c_2 \to 0$	**0**
v_3	**0**	$c_2 \to 1$	$c_3 \to 0$	**0**
v_4	**1**	$c_1 \to 0$	$c_4 \to 1$	**1**
v_5	**0**	$c_1 \to 1$	$c_4 \to 0$	**0**
v_6	**1**	$c_2 \to 0$	$c_3 \to 1$	**1**
v_7	**0**	$c_3 \to 0$	$c_4 \to 0$	**0**
v_8	**1**	$c_1 \to 1$	$c_3 \to 1$	**1**

Node v_2 sends symbol 1 to c_1 a c_2 and both the nodes return symbol 0, while e.g. v_4 gets the opposite answer only from one node $\Rightarrow$ correction of v_2.

Soft decision (SD) decoding

Sum–product algorithm (simplified form):

- **based on a similar principle as the HD algorithm, it does not work with bits but with probabilities of 0 or 1 occurrence.**
- **the goal is to find a decoded vector *d*, which is an estimate of the code vector *c* actually transmitted able to satisfy the syndrome condition $H \cdot d = 0$.**

The estimates of the channel information are:

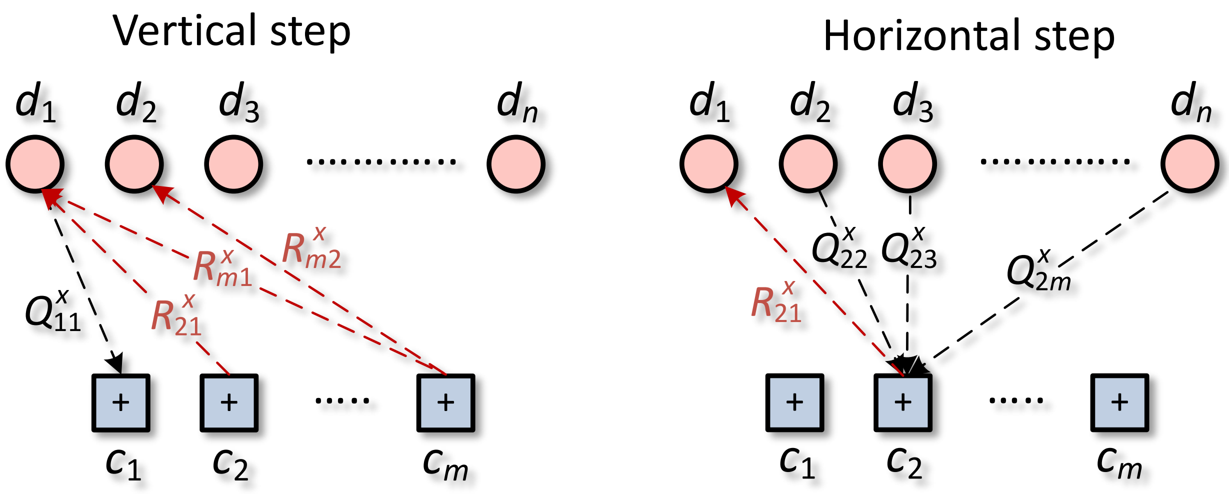
$$f_j^1 = \frac{1}{1+e^{-\frac{2Ay_j}{\sigma^2}}}, \quad f_j^0 = 1 - f_j^1, \tag{5.11}$$

where y_j is the channel output at time instant j, σ is the standard deviation of the noise, and bits are transmitted in polar format with amplitudes $\pm A$.

Now, we define coefficients $Q_{ij}^0 + Q_{ij}^1 = 1$ in simplified version, where

$$Q_{ij}^0 = f_j^0 \text{ and } Q_{ij}^1 = f_j^1 \tag{5.12}$$

Calculation of Q_{ij}^{x} (horizontal steps) and R_{ij}^{x} (vertical steps) are performed alternately:



Then we define the difference $\delta Q_{ij} = Q_{ij}^{0} - Q_{ij}^{1}$ **and the quantity**

$$\delta R_{ij} = \prod_{j \in N(i) \backslash j} \delta Q_{ij} \tag{5.13}$$

where $N(i)$ represents the set of indexes of all the symbol nodes d_j connected to the parity check node c_i, whereas $N(i)\backslash j$ represents the same set except d_j.

Coefficients R_{ij}^{x} then are:

$$R_{ij}^{0} = {}^{1}\!/_{2}\left(1 + \delta R_{ij}\right), \quad R_{ij}^{1} = {}^{1}\!/_{2}\left(1 - \delta R_{ij}\right), \tag{5.14}$$

The estimate of the decoded vector requires *a posteriori probabilities*

$$Q_{j}^{x} = \alpha_{j} f_{j}^{x} \prod_{i \in M(j)} R_{ij}^{x} \tag{5.15}$$

where $M(j)$ represents the set of indexes of all the parity check nodes connected to the symbol node d_j and constant α_j is selected so that

$$Q_{j}^{0} + Q_{j}^{1} = 1.$$

The estimate of the decoded vector $\hat{d}_j$ can be finally obtained by conditions:

$$\text{if } Q_{j}^{0} > Q_{j}^{1} \text{ then } d_j = 0, \text{else } d_j = 1. \tag{5.16}$$

Example 5.5: let the vector c = [1 0 1 0 1 1 1 1 0 0 0 1 0 0 1] was sent through the channel, A = ±1, σ = 0.8. Use the SP algorithm to check the received vector and correct errors it if necessary.

1. We calculate f_j^0 and f_j^1 using (5.11)

Position j	Codeword c	f_j^0	f_j^1	Hard decission d
1	1	0,0399	0,9601	1
2	0	0,9814	0,0185	0
3	1	0,0236	0,9764	1
4	0	0,0589	0,9410	1 (error)
5	1	0,0495	0,9504	1
6	1	0,0549	0,9451	1
7	1	0,9653	0,0347	0 (error)
8	1	0,0321	0,9680	1
9	0	0,9686	0,0314	0
10	0	0,9653	0,0347	0
11	0	0,9580	0,0419	0
12	1	0,0473	0,9526	1
13	0	0,9046	0,0954	0
14	0	0,9495	0,0505	0
15	1	0,0260	0,9740	1

2. We assign $Q_{ij}^0 = f_j^0$ and $Q_{ij}^1 = f_j^1$ using (5.12) and perform horizontal step $\Rightarrow$ R_{ij}^0 and R_{ij}^1 (5.9) using *parity check matrix* (5.7).
3. Finally, we perform vertical step to get coefficients Q_j^0 and Q_j^1 using (5.15) and apply (5.16). To do it, we need $\alpha_j = 1/(f_j^0 \prod_{i \in M(j)} R_{ij}^0 + f_j^1 \prod_{i \in M(j)} R_{ij}^1)$.

Position	c	α_j	Q_j^0	Q_j^1	d
1	1	73,99	0,0933	0,9067	1
2	0	12,11	0,9996	0,0004	0
3	1	71,82	0,0220	0,9780	1
4	0	78,47	0,8057	0,1943	0
5	1	1,89	0,00001	0,9999	1
6	1	49,34	0,0352	0,9648	1
7	1	123,09	0,2782	0,7218	1
8	1	44,25	0,0341	0,9659	1
9	0	14,59	0,9998	0,0002	0
10	0	79,22	0,9595	0,0405	0
11	0	2,08	0,9999	0,00001	0
12	1	12,82	0,0005	0,9995	1
13	0	72,30	0,9200	0,0799	0
14	0	8,11	0,9988	0,0012	0
15	1	72,42	0,0170	0,9830	1

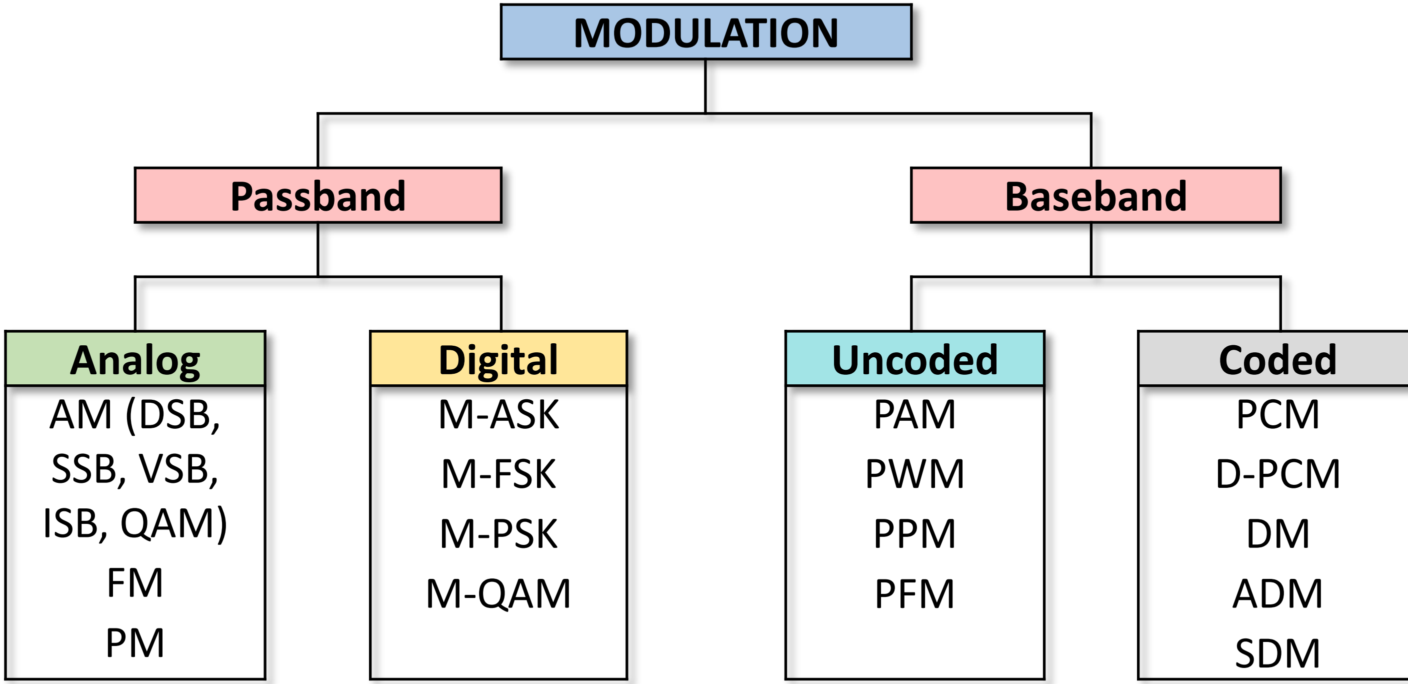
corrected

corrected

[5.1] LIN, S., COSTELLO, D. J. *Error Control Coding: Fundamentals and Applications*, second edition. Prentice Hall: Englewood Cliffs, NJ, 2005, 1271 s. ISBN: 0-13-042672-5.

6. Baseband modulation

Overview of modulations



AM (*Amplitude Modulation*), **DSB** (*Double Side Band*), **SSB** (*Single Side Band*), **VSB** (*Vestigial Side Band*), **ISB** (*Independent Side Band*), **QAM** (*Quadrature AM*), **FM** (*Frequency Modulation*), **PM** (*Phase Modulation*), ***M-*** (*M-ary*), ***x*SK** (*x Shift Keying*), **PAM** (*Pulse Amplitude Modulation*), **PWM** (*Pulse Width Modulation*), **PPM** (*Pulse Position modulation*), **PFM** (*Pulse Frequency modulation*), **PCM** (*Pulse Coded Modulation*), **D-PCM** (*Diferential – PCM*), **DM** (*Delta Modulation*), **ADM** (*Adaptive DM*), **SDM** (*Sigma DM*)

Let us assume a data signal in the form of rectangular pulses $p(t)$ of width T_b acquiring two or more states (amplitudes) a_n. Its spectrum is expressed using PSD (*Power Spectral Density*).

The data signal can be regarded as stationary ergodic random process where a_n is a random variable. Then the PSD is

$$P_s(f) = \lim_{T\to\infty}\left[\frac{\overline{|S_T(f)|^2}}{T}\right] \quad \textbf{where} \quad S_T(f) = \int_{-T/2}^{T/2} s_T(t)e^{-j\omega t}dt \qquad (6.1)$$

and $s_T(t) = \sum_{n=-N}^{N} a_n p(t - nT_b)$**. Then**

$$S_T(f) = F[s_T(t)] = \sum_{n=-N}^{N} a_n F[p(t-nT_b)] = \sum_{n=-N}^{N} a_n P(f) e^{-j\omega n T_b} \qquad (6.2)$$

where the spectral function of a rectangular pulse is

$$P(f) = T_b \mathrm{sinc}(\pi f T_b) \qquad (6.3)$$

By substituting (6.2) **into** (6.1) **we have**

$$P_s(f) = \lim_{T\to\infty}\left(\frac{1}{T}|P(f)|^2\,\overline{\left|\sum_{n=-N}^{N} a_n e^{-j\omega n T_b}\right|^2}\right)$$

$$= |P(f)|^2 \lim_{T\to\infty}\left(\frac{1}{T}\sum_{n=-N}^{N}\sum_{m=-N}^{N}\overline{a_n a_m}\,e^{-j\omega(m-n)T_b}\right) \qquad (6.4)$$

By introducing $m = n + k,\ \ T = (2N+1)T_b$, **and** $R_r = \overline{a_n a_{n+k}}$

$$P_p(f) = \frac{|P(f)|^2}{T_b}\lim_{N\to\infty}\left[\frac{1}{(2N+1)}\sum_{n=-N}^{N}\sum_{r=-N-n}^{N-n} R_r e^{-j\omega r T_b}\right] = \frac{|P(f)|^2}{T_b}\sum_{r=-\infty}^{\infty} R_r e^{-j\omega r T_b} \qquad (6.5)$$

The autocorrelation R_r **can be expressed as**

$$R_r = \overline{a_n a_{n+r}} = \sum_{i=1}^{I}(a_n a_{n+r})_i P(i) \qquad (6.6)$$

$P(i)$ **is the probability of** $(a_n a_{n+r})$ **occurrence.**

If the amplitudes a_n are statistically independent (6.6) **become to**

$$R_r = \begin{Bmatrix} \overline{a_n^2} & r = 0 \\ (\overline{a_n})^2 & r \neq 0 \end{Bmatrix} = \begin{Bmatrix} \sigma_a^2 + m_a^2 & r = 0 \\ m_a^2 & r \neq 0 \end{Bmatrix} \tag{6.7}$$

where σ_a and m_a are the standard deviation and mean value respectively. Using *Poisson formula*

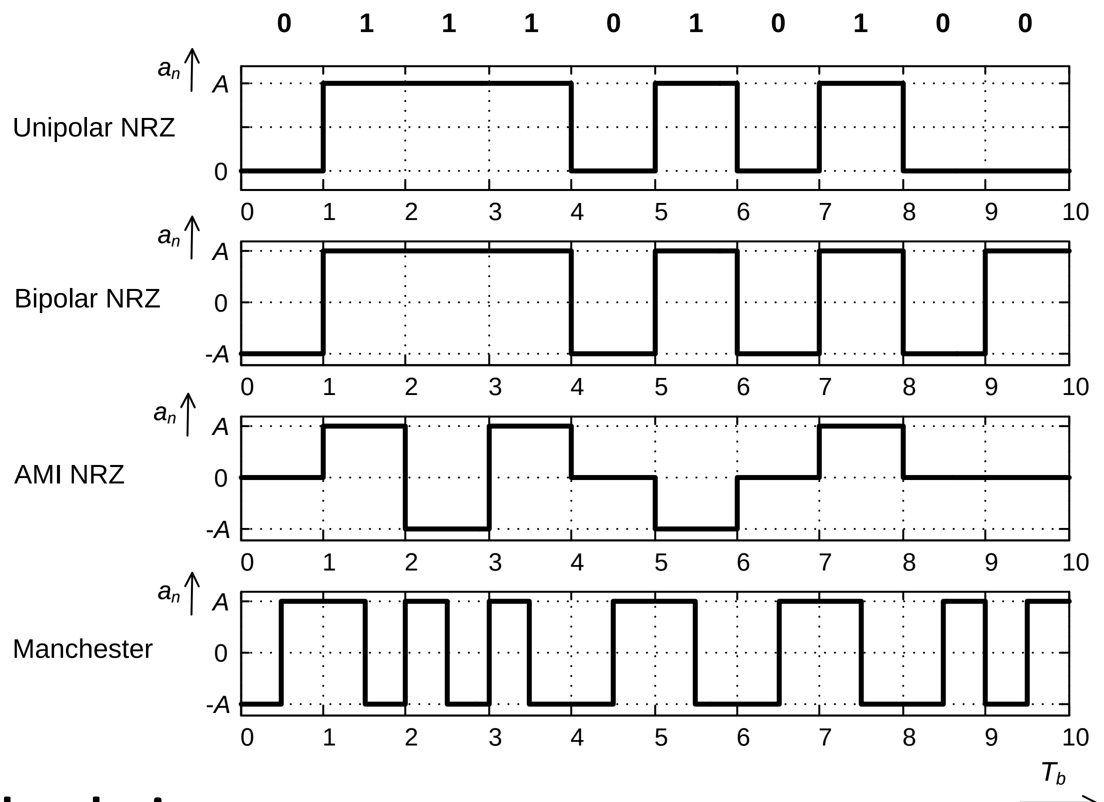
$$\sum_{r=-\infty}^{\infty} e^{jr2\pi f T_b} = \frac{1}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)$$

we can rewrite PSD to

$$P_p(f) = \frac{|P(f)|^2}{T_b}\left[\sigma_a^2 + m_a^2 \sum_{r=-\infty}^{\infty} e^{jr\omega T_b}\right] = \frac{|P(f)|^2}{T_b}\left[\sigma_a^2 + \frac{m_a^2}{T_b} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right] \tag{6.8}$$

Note 1: **the** sinc **function in $P(f)$ is zero at all multiples of $f_b = 1/T_b$.**

Basic line codes, NRZ (*No Return to Zero*)



Aims of line codes design:

- **zero DC component $\Rightarrow$ easy design of electronic circuits,**
- **easy synchronization (at least one amplitude change in each bit).**

Unipolar NRZ: $a_n = \{0, A\}$ **for** $P(0) = P(A)$ **we can write**.

$$R_r = \begin{cases} \frac{1}{2}0 + \frac{1}{2}A^2 = \frac{A^2}{2} & r = 0 \\ \left[\frac{1}{2}0 + \frac{1}{2}A\right]^2 = \frac{A^2}{4} & r \neq 0. \end{cases}$$

Using (6.8) **the PSD is**

$$P_s(f) = |P(f)|^2 \frac{A^2}{4T_b}\left[1 + \frac{1}{T_b}\sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T_b}\right)\right]$$

and assuming *Note 1*

$$P_{s\_Uni}(f) = \frac{A^2 T_b}{4}\mathrm{sinc}^2(\pi f T_b) + \frac{A^2}{4}\delta(f) \tag{6.9}$$

Transmission of a long sequence of the same bit causes nonzero DC component and difficult synchronization.

Bipolar NRZ: $a_n = \{-A, A\}$ **for** $P(-A) = P(A)$ **we can write**

$$R_r = \begin{cases} \frac{1}{2}(-A^2) + \frac{1}{2}A^2 = A^2 & r = 0 \\ \left[\frac{1}{2}(-A) + \frac{1}{2}A\right]^2 = 0 & r \neq 0. \end{cases}$$

PSD:

$$P_{s\_Bip}(f) = A^2 T_b \mathrm{sinc}^2(\pi f T_b) \qquad (6.10)$$

Transmission of a long sequence of the same bit causes nonzero DC component and difficult synchronization (similar to Unipolar NRZ).

AMI-NRZ ***(Alternate Mark Inversion - NRZ)*** **:** $a_n = \{-A, 0, A\}$.

A binary 0 is encoded as zero volts, as in unipolar encoding, whereas a binary 1 is encoded alternately as a positive voltage or a negative voltage. Thus

$$P(-A) = P(A) = \frac{1}{4}, P(0) = \frac{1}{2}$$

Although the sequence of bits in the message can be independent, the amplitudes of the binary 1's are statistically dependent due to the $-A$, A **alternation** $\Rightarrow$ **equation** (6.7) **must not be used. We can use the following considerations:**

- **If** $r = 0$, $R_0 = A^2\,{}^1/_4 + 0^2\,{}^1/_2 + (-A)^2\,{}^1/_4 = A^2/2$**.**
- **For** $r = 1$, **the two consecutive bits** (d_n, d_{n+1}) **can be** (0,0), (0,1), (1,0) a (1,1). **The corresponding products** $a_n a_{n+1}$ **then are** $0, 0, 0$ a $-A^2$ **and** $R_1 = 3 \times 0\,{}^1/_4 + (-A^2)\,{}^1/_4 = -A^2/4$**.**

- **For $r > 1$, one half of all message sequences $(d_n, d_{n+1}, \ldots, d_{n+r})$ has $d_n = 0$ $\Rightarrow a_n a_{n+r} = 0$, one quarter of messages has $d_n = 1$ and $d_{n+r} = 0 \Rightarrow a_n a_{n+r} = 0$. For the last quarter we have $d_n = 1$ and $d_{n+r} = 1 \Rightarrow a_n a_{n+r} = \pm A$.**

PSD: $$P_{s\_AMI}(f) = T_b^2 \mathrm{sinc}^2(\pi f T_b) \frac{A^2}{T_b}\left[-\frac{1}{4}e^{j\omega T_b} + \frac{1}{2} - \frac{1}{4}e^{-j\omega T_b}\right]$$

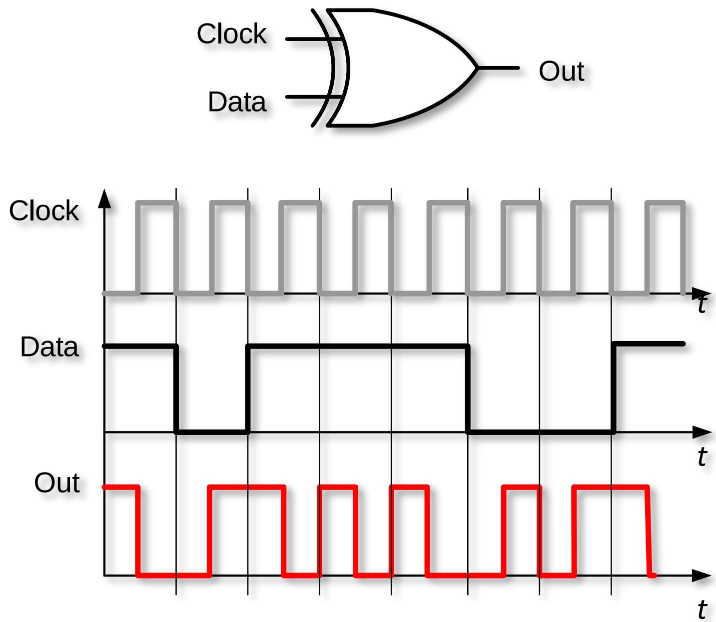
$$= A^2 T_b \mathrm{sinc}^2(\pi f T_b) \cdot \sin^2(\pi f T_b). \qquad (6.11)$$

DC component is always zero, transmission of a long sequence of zeros causes difficult synchronization.

Manchester (Bi-Φ-L): $a_n = \{-A, A\}$. Bi-phase coding guarantees at least one polarity alternation in each bit interval. $P(-A) = P(A)$, $R_0 = A^2$ and $R_r = 0$.

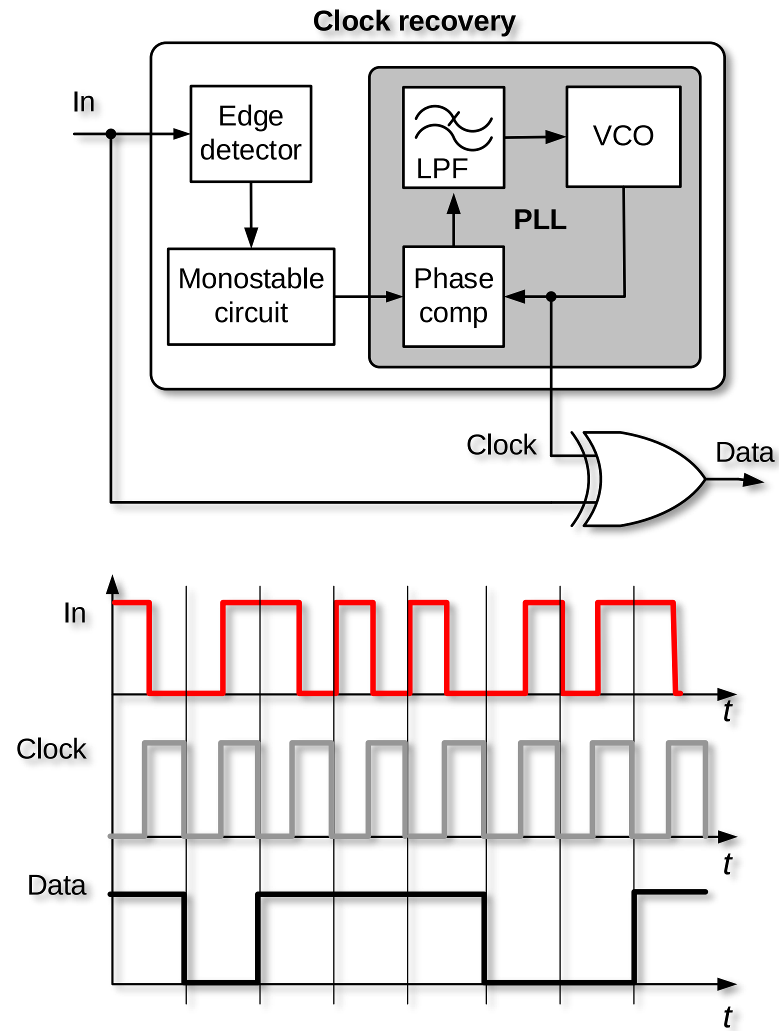
$$P_{s\_Man}(f) = A^2 T_b \mathrm{sinc}^2\left(\frac{\pi f T_b}{2}\right) \cdot \sin^2\left(\frac{\pi f T_b}{2}\right) \qquad (6.12)$$

Manchester encoder

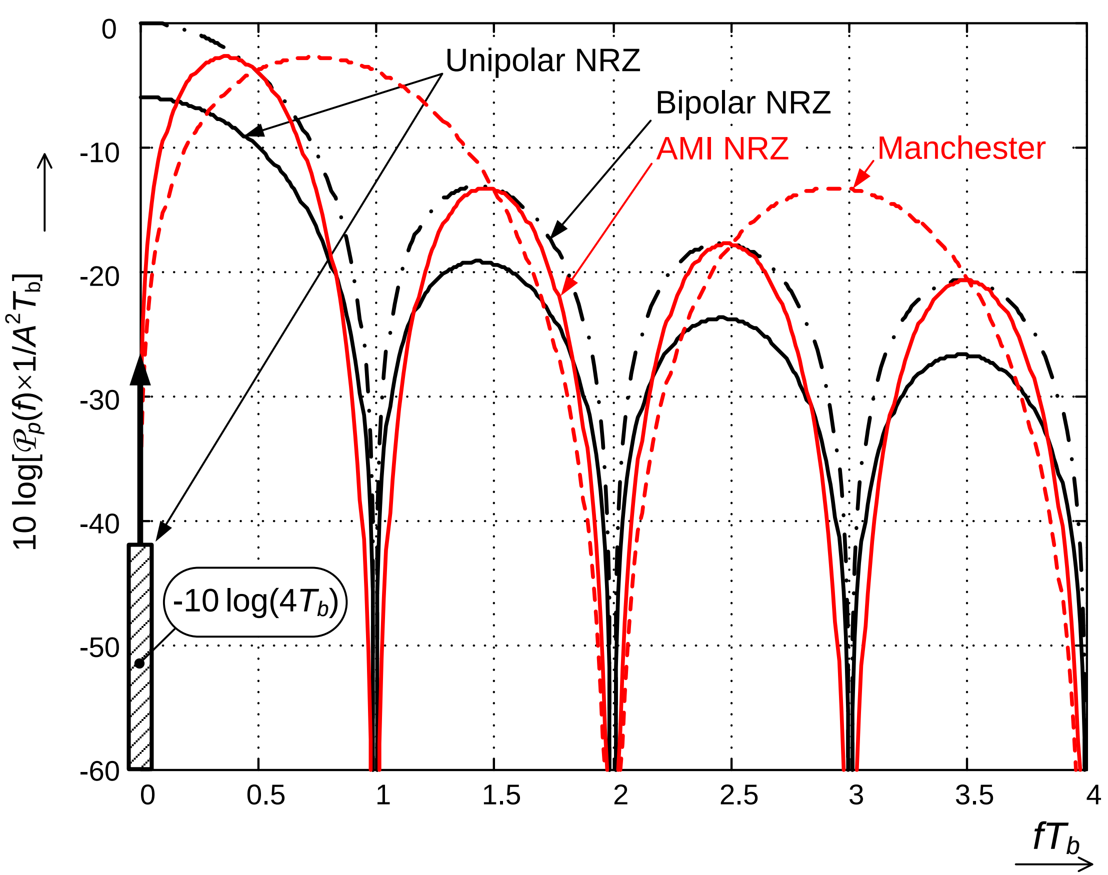


DC component is always zero, easy synchronization.

Manchester decoder



PSD of line codes



Continuous Phase FSK (CPFSK) is a member of CPM (*Continuous Phase Modulation*) family where a symbol change does not cause a step change in phase.

$$s_{CPM}(t) = \sum_{n=-\infty}^{\infty} A_c \cos[\omega_c t + \phi_n(t)]p(t - nT_s)\,, \qquad (7.1)$$

$\phi_n(t)$ **is the phase change (trajectory),** $p(t)$ **is the pulse shape function,** T_s **is the symbol period. The symbol frequency** $\omega_n = \omega_c + \Delta\omega_n$ **is constant because**

$$\Delta\omega_n = 2\pi a_n \Delta f \qquad (7.2)$$

where for *M*-ary CPFSK $a_n = \{\pm 1, \pm 3, \cdots, \pm(M-1)\}$
Relative phase change in the interval $nT_s \le t < (n+1)T_s$ **is**

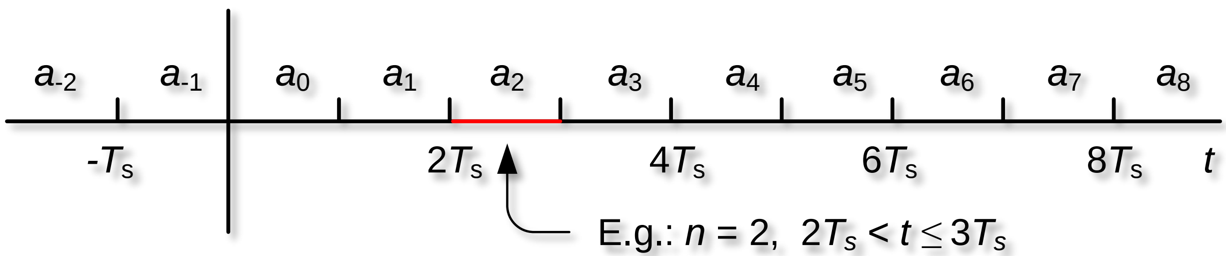
$$\Delta\phi_n(t) = \int_0^t \Delta\omega_n d\alpha = 2\pi a_n \Delta f t$$

7. Digital bandpass modulation

Continuous Phase FSK (*Frequency Shift Keying*)

For $t = T_s$ we have $\Delta\phi_n(T_s) = 2\pi a_n \Delta f T_s = \pi a_n \beta$, where $\beta = 2\Delta f T_s$ is the modulation index.

Symbol interval definition



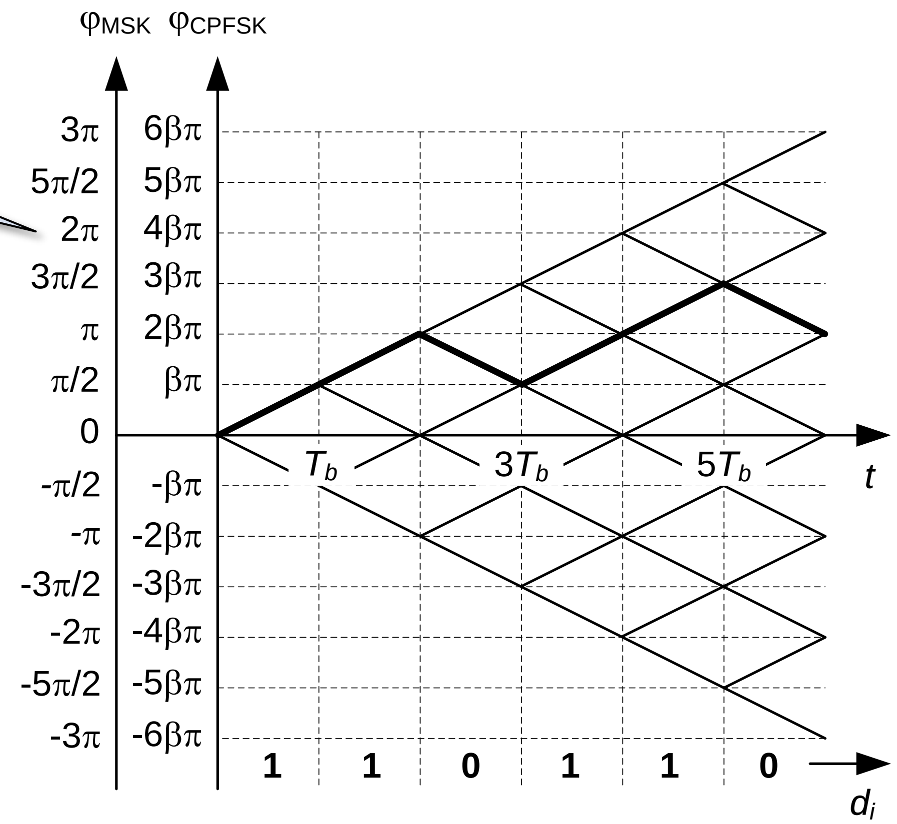
The instantaneous absolute phase during the *n*-th symbol transmission

$$\phi_n(t) = \sum_{i=0}^{n-1} \Delta\phi_i(T_s) + \Delta\phi_n(t - T_s) = \Phi_n + \frac{\pi}{T_s}\beta a_n t \qquad (7.3)$$

where $\Phi_n = \pi\beta\left(\sum_{i=0}^{n-1} a_i - n a_n\right)$

Example 7.1: let $M = 2$, then $a_n = \{1, -1\}$, bit period $T_b = T_s$, $\Delta\phi_n(T_s) = \pm\pi\beta$. For an input bit sequence d_n we have $a_n = 1$ for $d_n = 1$ and $a_n = -1$ for $d_n = 0$.

Trellis diagram for CPFSK and MSK



Bit error probability of CPFSK modulation for coherent demodulation

$$P_b = \frac{1}{2}\operatorname{erfc}\left[\sqrt{\frac{(1-\gamma)A_c^2 T_b}{4N_0}}\right] = \frac{1}{2}\operatorname{erfc}\left[\sqrt{\frac{(1-\gamma)E_b}{2N_0}}\right], \quad (7.4)$$

Where correlation coefficient $\gamma = \frac{2}{T_b}\int_0^{T_b} \cos(\omega_c - \Delta\omega)\cos(\omega_c + \Delta\omega)\, dt = \operatorname{sinc}(2\Delta\omega T_b) + \operatorname{sinc}(\omega_c T_b)\, cos(\omega_c T_b)$.

For $\omega_c T_b = k\pi$ **where** $k = 1,2,\cdots$ **or** $\omega_c T_b << 1$ **we have**

$$\gamma = \operatorname{sinc}(2\Delta\omega T_b). \quad (7.5)$$

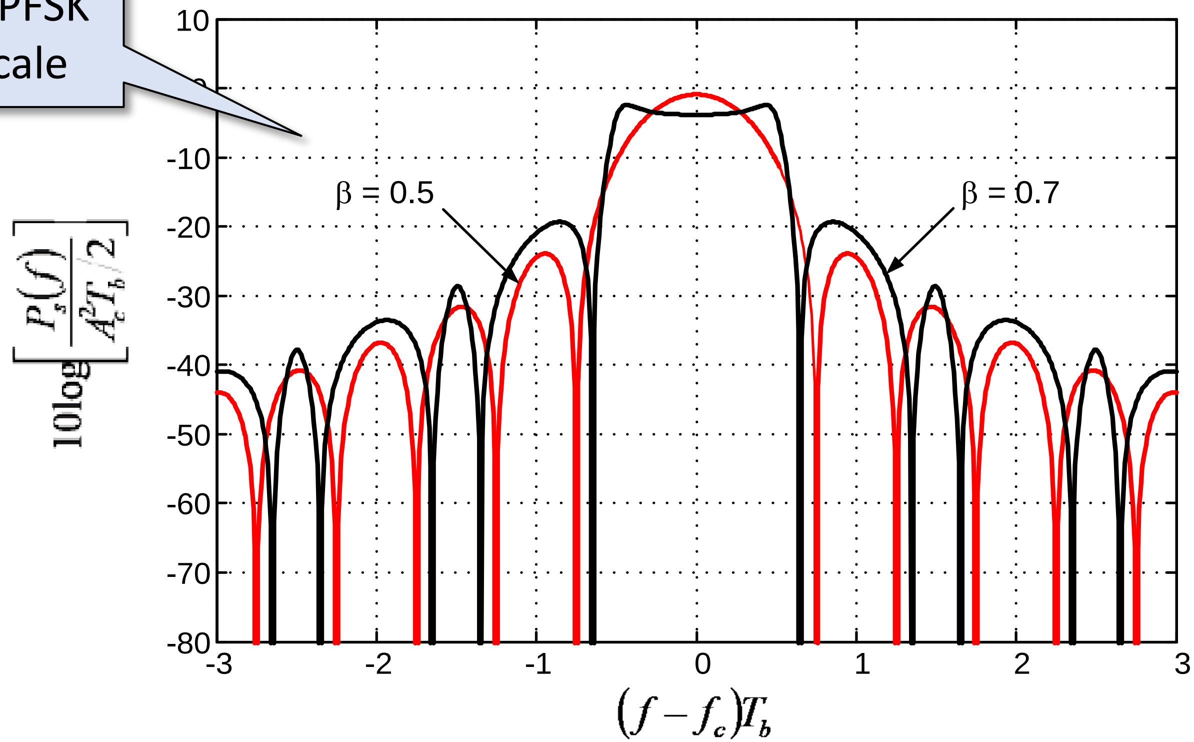
Minimal value of (7.4) can be obtained for $2\Delta\omega T_b \approx 3\pi/2$**. Then**

$$P_b = \frac{1}{2}\operatorname{erfc}\left[\sqrt{\frac{1.21 E_b}{2N_0}}\right]$$

PSD: $$P_s(f) = \frac{A_c^2 T_b}{4} \cdot \frac{(\Delta\omega T_b)^2 \operatorname{sinc}^2[\gamma_1(f)] \operatorname{sinc}^2[\gamma_2(f)]}{1 + \cos^2(\Delta\omega T_b) - 2\cos(\Delta\omega T_b)\cos[(2\pi f - \omega_c)T_b]}$$

where $\gamma_1(f) = (2\pi f - \omega_c + \Delta\omega)\frac{T_b}{2}$ $\quad \gamma_2(f) = (2\pi f - \omega_c - \Delta\omega)\frac{T_b}{2}$

PSD of CPFSK in log scale



For MSK $(M = 2)$ **the signals are orthogonal** $\Rightarrow \gamma = 0$**. From (7.5) we get** $2\Delta\omega T_b = k\pi$ **or** $\Delta\omega = k\,\omega_b/4\,,\quad k = 1,2,\cdots$ **. Assuming** $k = 1$ **we have**

$$\Delta\omega = \omega_b/4 = \pi/2T_b \quad \text{or} \quad \Delta f = f_b/4 = 1/4T_b \tag{7.6}$$

To meet the orthogonality condition, the signaling frequencies $f_1 = f_c - \Delta f$ **and** $f_2 = f_c + \Delta f$ **must be**

$$f_1 = k\frac{f_b}{2} \quad \text{and} \quad f_2 = (k+1)\frac{f_b}{2}, \quad k \in Z. \tag{7.7}$$

Then $\beta = 2\Delta f T_b = (f_1 - f_2)T_b = 0.5$

Assuming (7.1), (7.3), and $\beta = 0.5$ **for** $nT_s \le t \le (n+1)T_s$ **we get**

$$s_{nMSK}(t) = A_c \cos\left(\omega_c t + \Phi_n + a_n \frac{\pi}{2T_b} t\right), \tag{7.8}$$

where $a_n = \{1, -1\}$

To maintain continuous phase at bit transition $t = kT_b$ the condition

$$a_{n-1}\frac{\pi}{2}n + \Phi_{n-1} = a_n\frac{\pi}{2}n + \Phi_n \quad \mathrm{mod}(2\pi)$$

has to be met. This is

$$\Phi_n = \begin{cases} \Phi_{n-1} & \mathrm{mod}(2\pi)\,, \quad a_n = a_{n-1} \\ \Phi_{n-1} \pm n\pi & \mathrm{mod}(2\pi)\,, \quad a_n \neq a_{n-1} \end{cases} \tag{7.9}$$

For $\Phi_n = 0$ **or** $\pm\pi$ **using** $\cos(\alpha+\beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$ (7.8) **become to**

$$s_{nMSK}(t) = A_c s_I(t)\cos(\omega_c t) - A_c s_Q(t)\sin(\omega_c t)\,, \tag{7.10}$$

where

$$s_I(t) = \cos(\Phi_n)\cos\left(a_n\frac{\pi t}{2T_b}\right) = \cos(\Phi_n)\cos\left(\frac{\pi t}{2T_b}\right)$$

$$s_Q(t) = \cos(\Phi_n)\sin\left(a_n\frac{\pi t}{2T_b}\right) = a_n\cos(\Phi_n)\sin\left(\frac{\pi t}{2T_b}\right)$$

Conclusion:

- $s_I(t)$ can change only if n is even $[\cos(\Phi_n) = \pm 1]$ in $t = (2n+1)T_b$ when $\cos(\pi t/2T_b)$ goes through zero.
- $s_Q(t)$ can change only if n is odd $[a_n \cos(\Phi_n) = \pm 1]$ in $t = 2nT_b$ when $\sin(\pi t/2T_b)$ goes through zero.
- Both the signals can change always after the interval $2T_b$. In I branch, the moments of change relative to Q branch are shifted by the T_b.

Example 7.2:
d_n = 0 0 1 1 0 1 0 1 1 0
Calculate the coefficients $\cos(\Phi_n)$ and $a_n \cos(\Phi_n)$

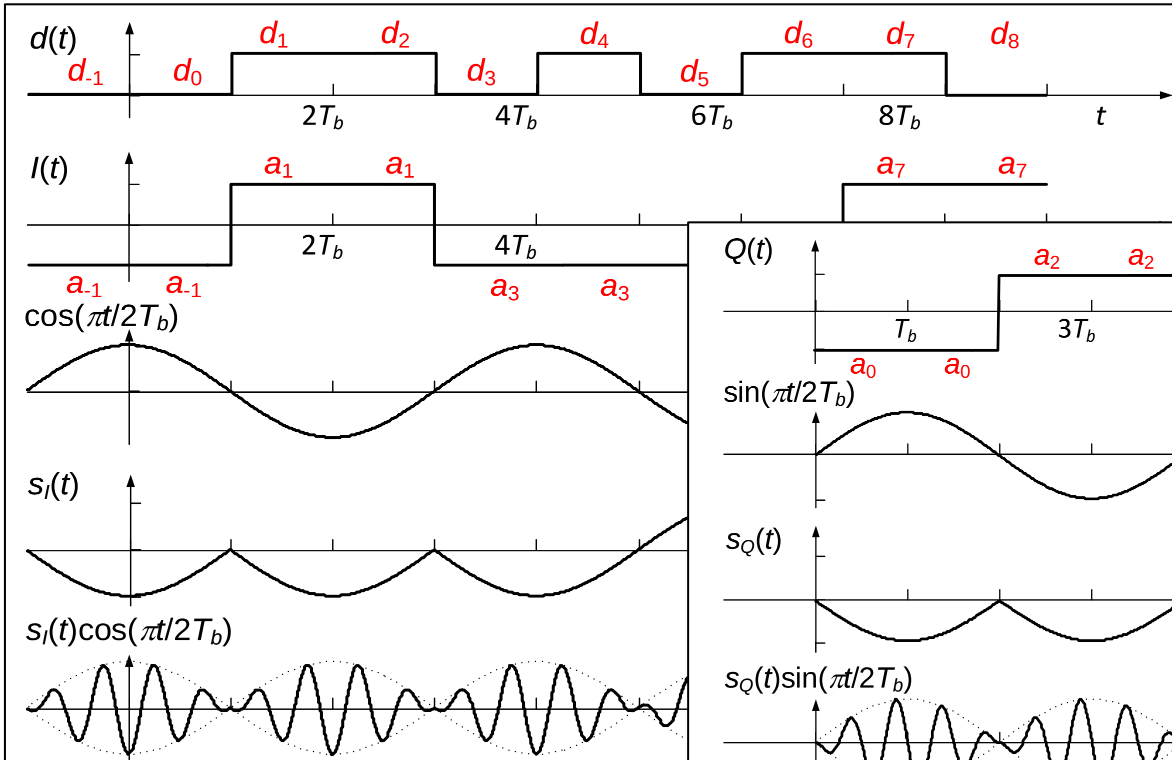
Changes for odd ***n***

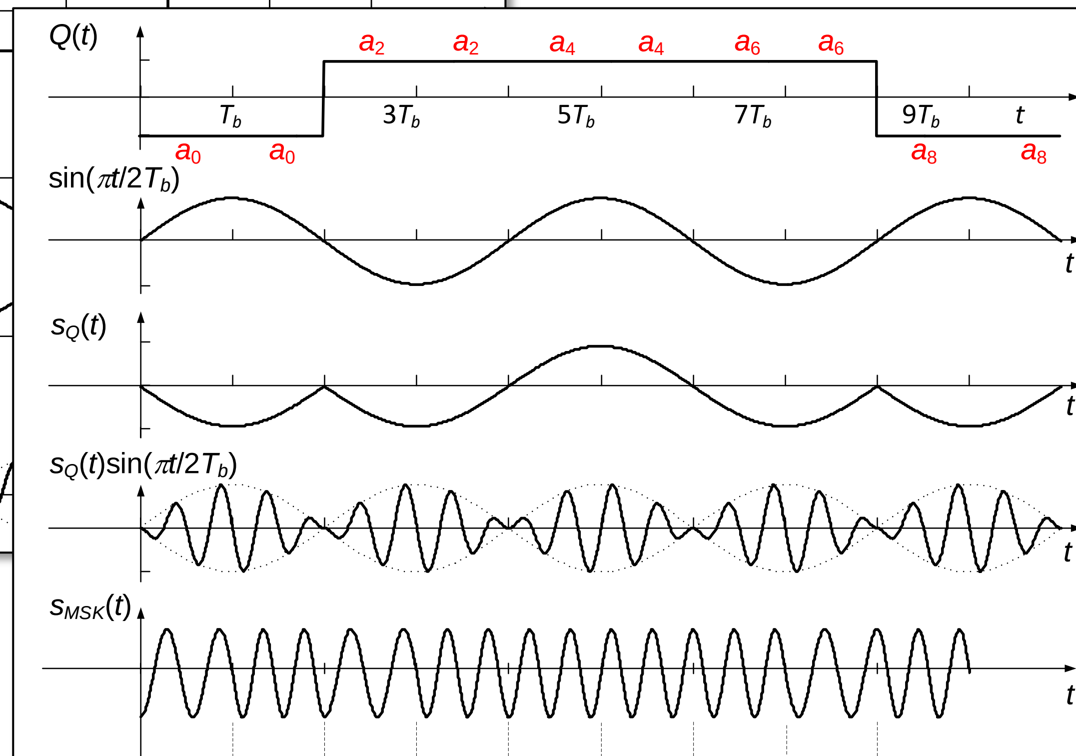
Changes for even ***n***

n	-1	0	1	2	3	4	5	6	7	8
a_{n-1}	-1	-1	-1	1	1	-1	1	-1	1	1
a_n	-1	-1	1	1	-1	1	-1	1	1	-1
Φ_{n-1}		0	0	π	π	0	0	π	π	π
Φ_n	0	0	π	π	0	0	π	π	π	π
$\cos \Phi_n$	1	1	-1	-1	1	1	-1	-1	-1	-1
$a_n \cos \Phi_n$	-1	-1	-1	-1	-1	1	1	-1	-1	1
frequency		f_0	f_1	f_0	f_1	f_1	f_1	f_1	f_0	f_1

MSK (*Minimum Shift Keying*)

MSK signal generation

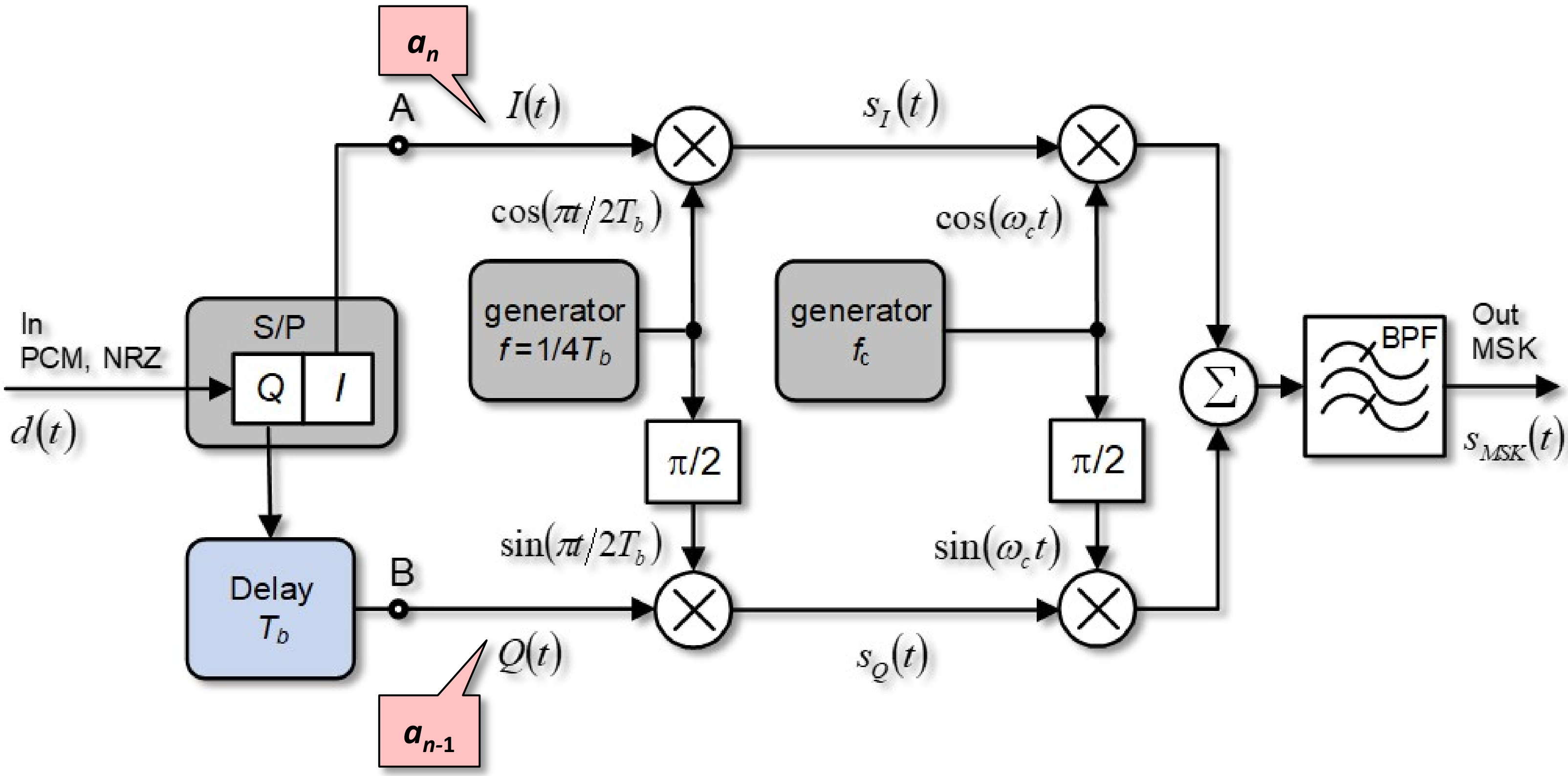




n	0	1	2	3	4	5	6	7	8
a_{n-1}, a_n	-1,-1	-1,1	1,1	1,-1	-1,1	1,-1	-1,1	1,1	1,-1

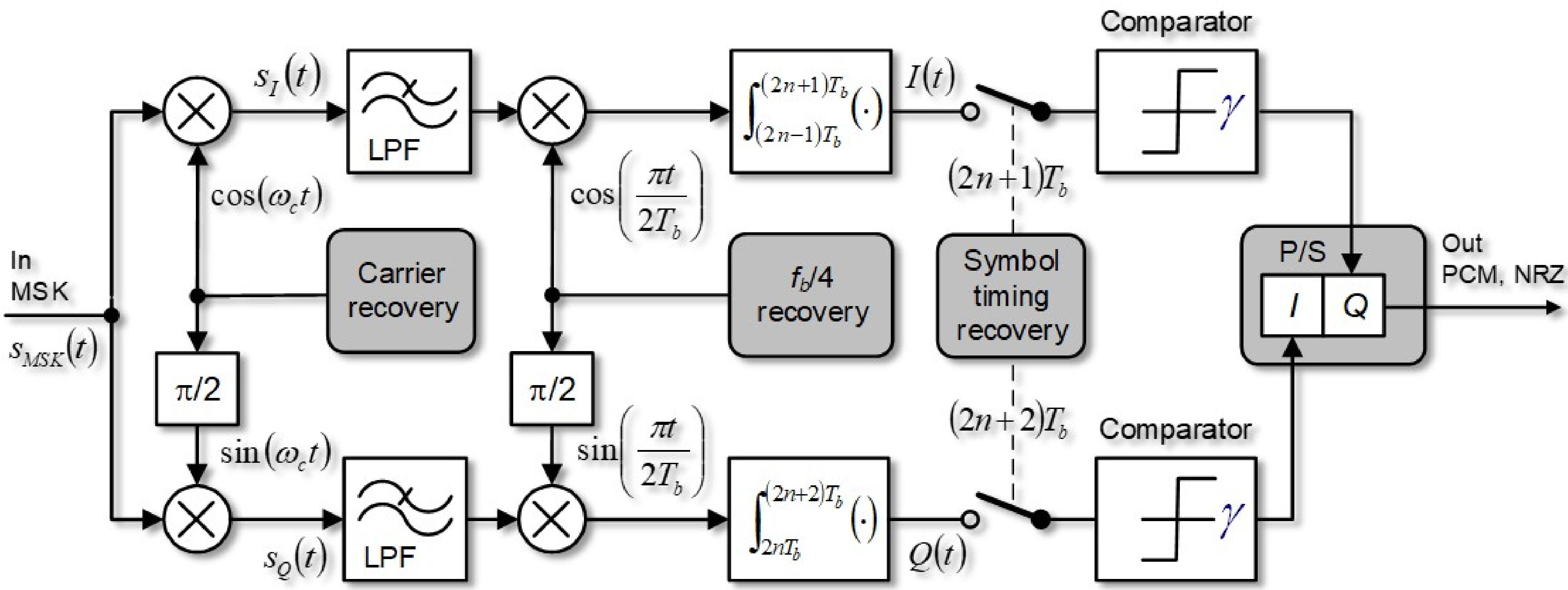
$$f = \begin{cases} f_0 = f_c - \dfrac{1}{4T_b} & \text{for} \quad a_n = a_{n-1}, \\ f_1 = f_c + \dfrac{1}{4T_b} & \text{for} \quad a_n \neq a_{n-1}. \end{cases}$$

MSK modulator



S/P: serial to parallel converter
BPF: bandpass filter

MSK demodulator



P/S: parallel to serial converter
LPF: lowpass filter

FFSK modulation: a modification of MSK, but it unambiguously assigns particular frequencies to the bit values of the input sequence $d(t)$.
By appliyng a differential coder where

$$b_n = b_{n-1} \oplus d_n,$$

each bit $d_n = 1$ would be modulated by frequency f_1 and each bit $d_n = 0$ by frequency f_0.

Example 7.3:
d_n = 0 0 1 1 0 1 0 1 1 0
Calculate the coefficients a_n and a_{n-1}.

$f_0 \equiv d_n = 0$

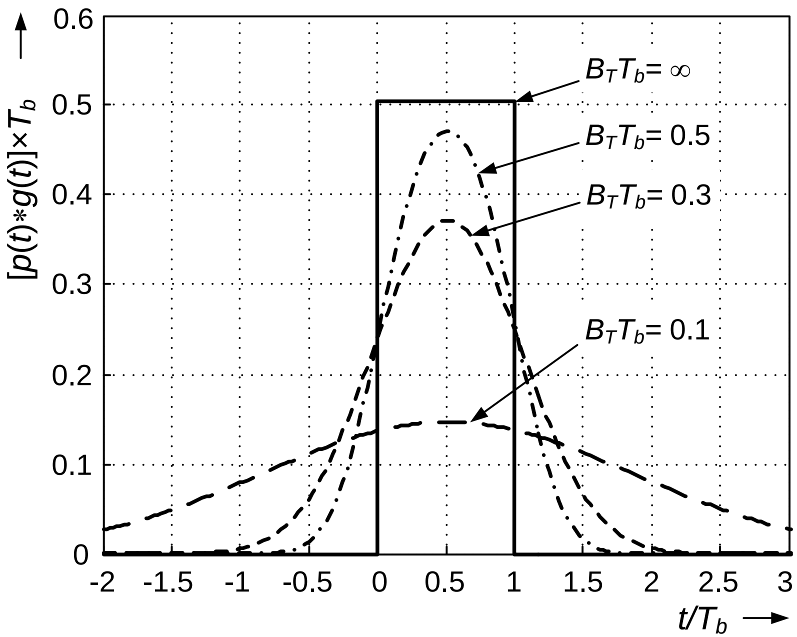
n	−1	0	1	2	3	4	5	6	7	8
d_n	0	0	1	1	0	1	0	1	1	0
b_{n-1}	–	0	0	1	0	0	1	1	0	1
$b_n = b_{n-1} \oplus d_n$	0	0	1	0	0	1	1	0	1	1
a_n	−1	−1	1	−1	−1	1	1	−1	1	1
a_{n-1}	–	−1	−1	1	−1	−1	1	1	−1	1
frequency	–	f_0	f_1	f_1	f_0	f_1	f_0	f_1	f_1	f_0

GMSK: applies symbol filtering using GLPF (*Gaussian Low-Pass*) before MSK modulation.

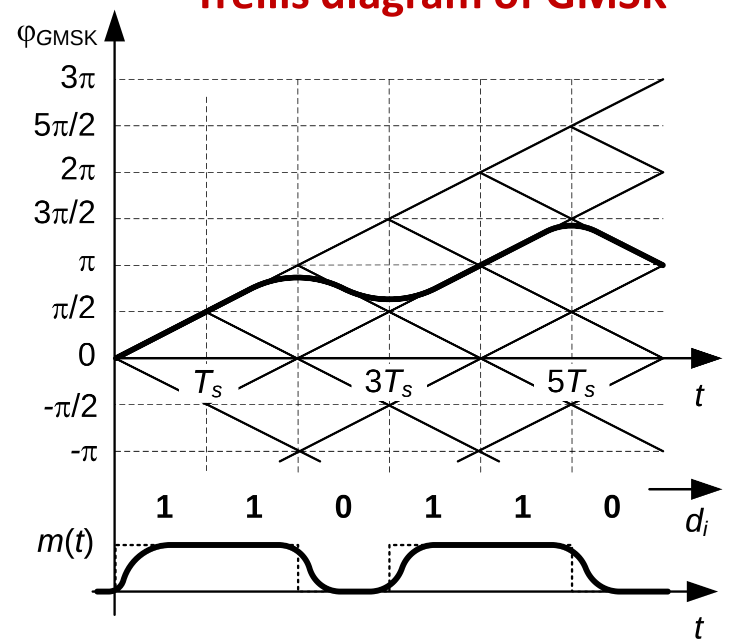
$$g(t) = \frac{K}{\sqrt{2\pi}} \exp\left(-\frac{K^2 t^2}{2}\right) \quad \text{where} \quad K = \frac{2\pi B_T}{\sqrt{\ln(2)}}$$

where B_T is 3 dB bandwidth

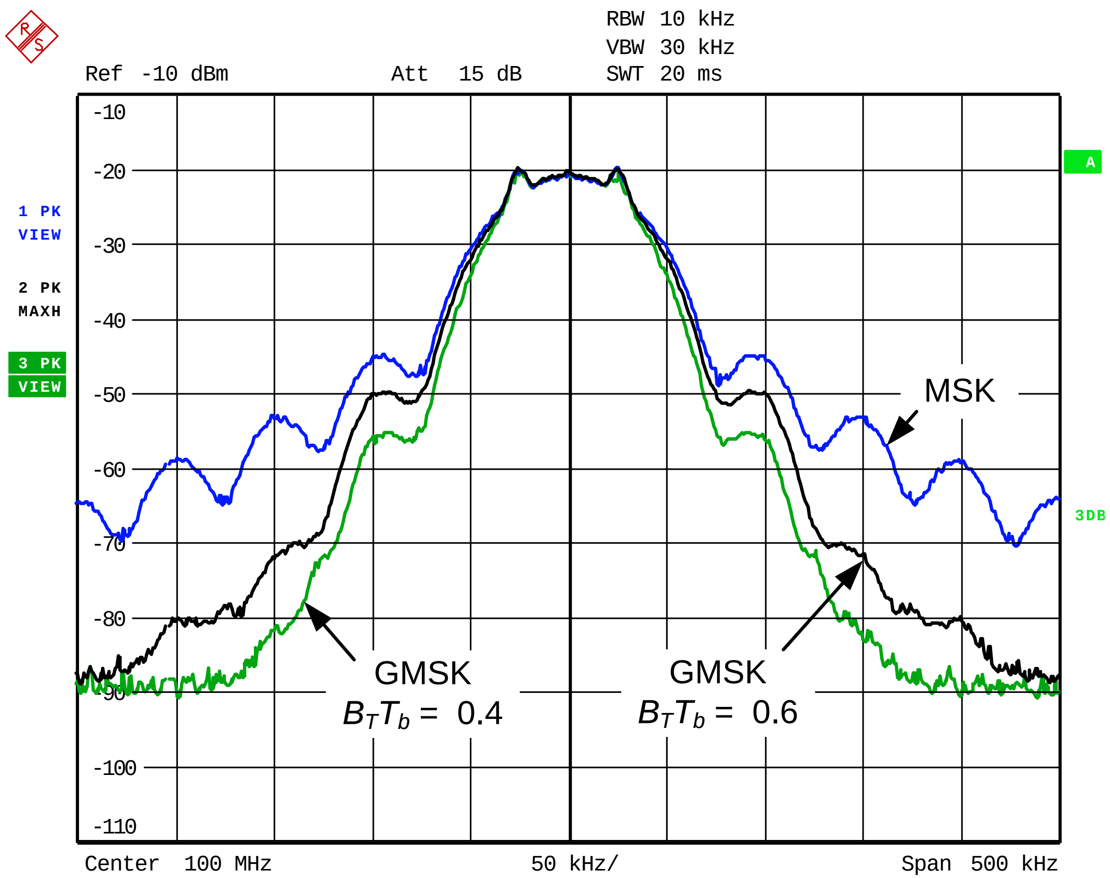
GLPF impulse response



Trellis diagram of GMSK



Measured power spectrum of MSK and GMSK

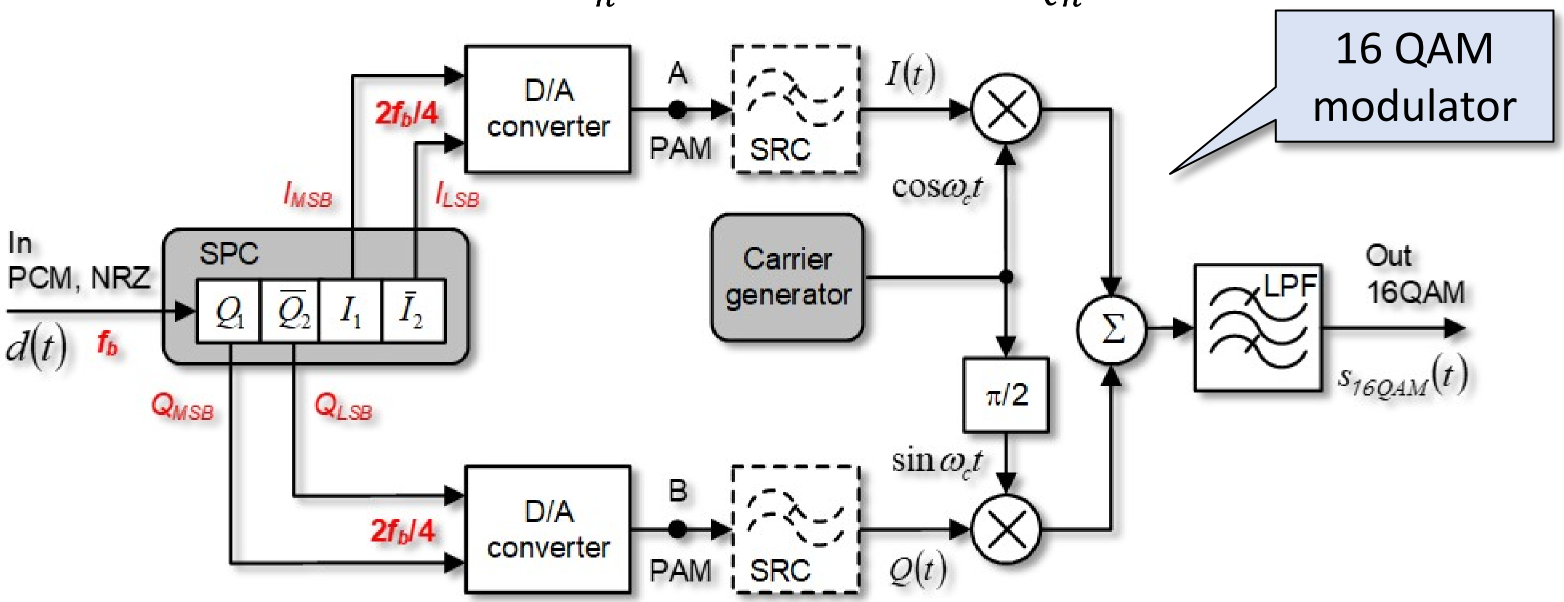


MQAM: change of amplitude A_n and phase ϕ_n depending on n-tuple signal bits

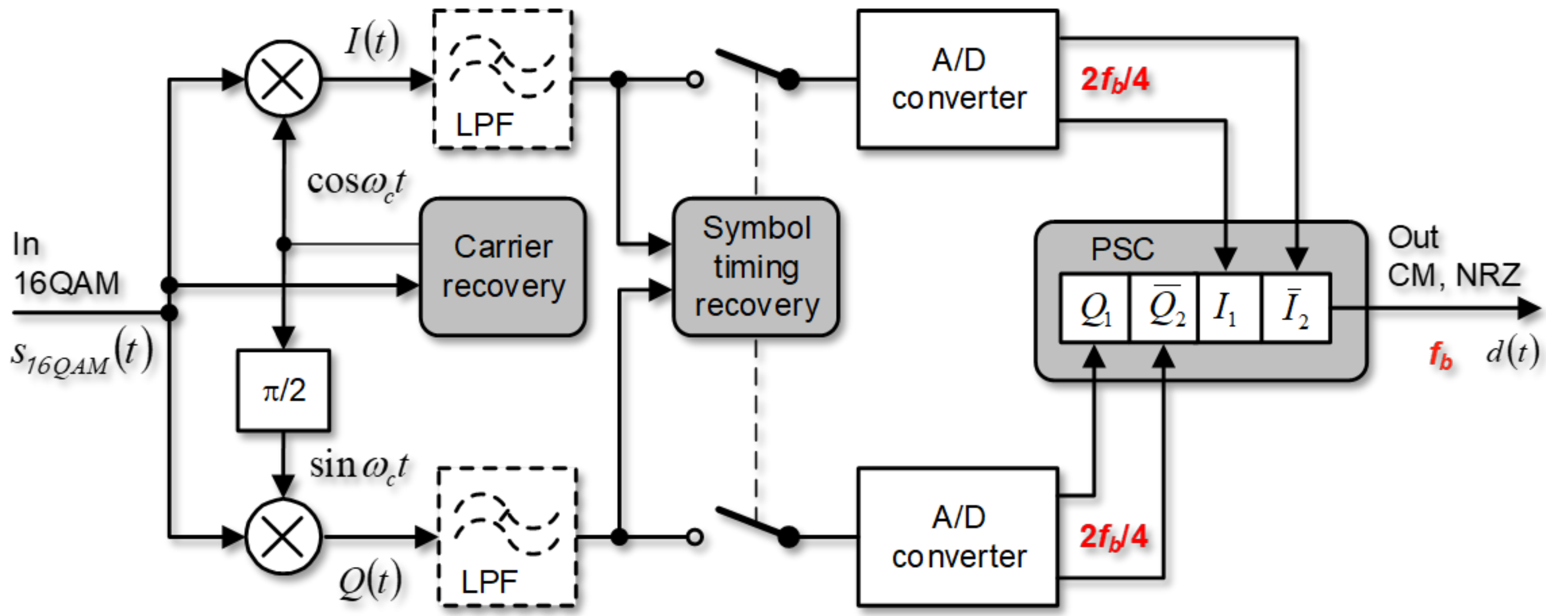
$$s_{MQAM}(t) = \sum_{n=-\infty}^{\infty} A_n \cos[\omega_c t + \phi_n](t - nT_s)$$

For the time interval $nT_s \leq t \leq (n+1)T_s$ we can write

$$s_{MQAM}(t) = \underbrace{A_n \cos\phi_n}_{I_n} \cos\omega_c t - \underbrace{A_n \sin\phi_n}_{Q_n} \sin\omega_c t$$



16 QAM demodulator

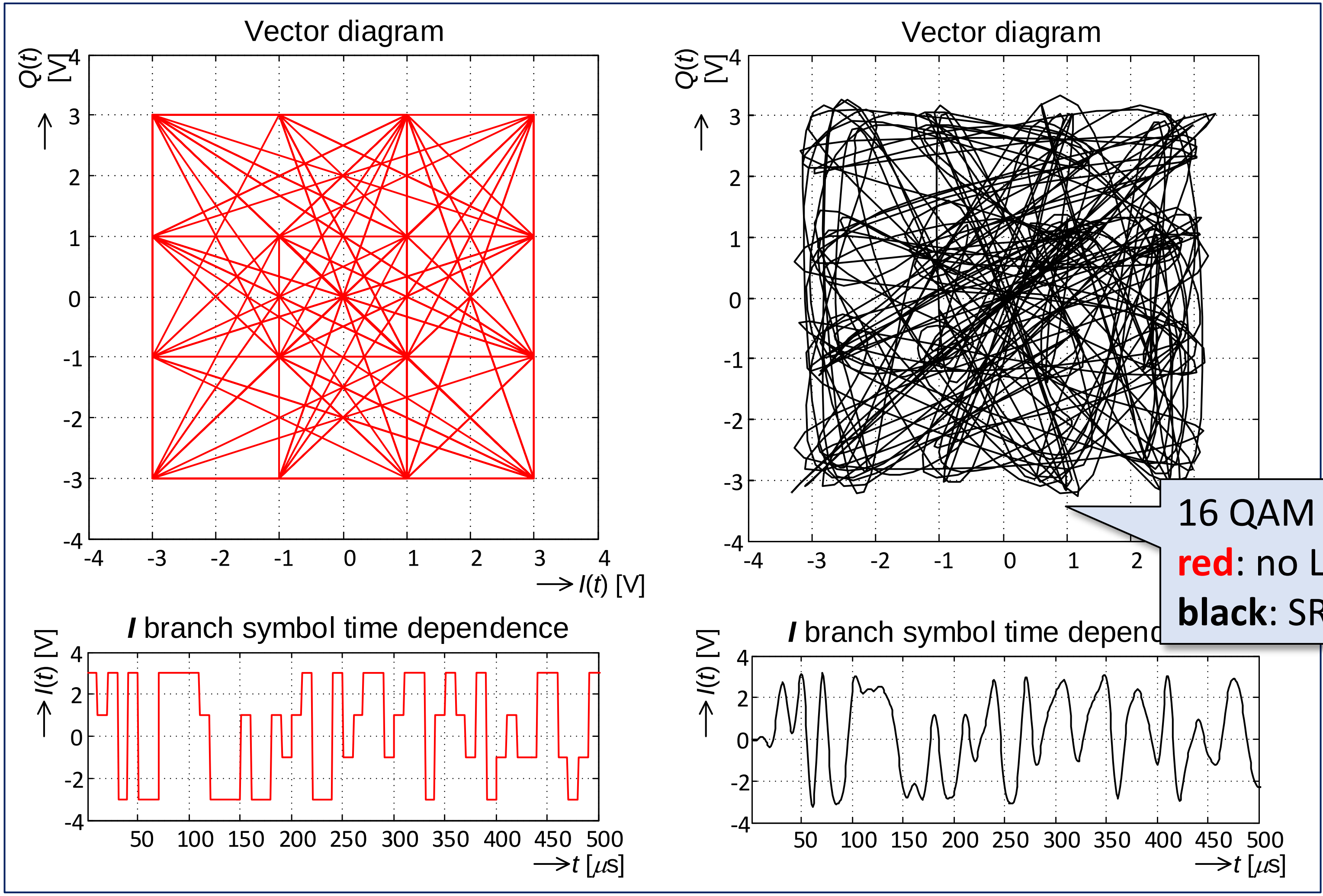


***M*QAM PSD:**

$$P_{IQ\_QAM}(f) = A_{avg}^2 T_s \mathrm{sinc}^2(\pi f T_s) = A_{avg}^2 n T_b \mathrm{sinc}^2(\pi f n T_b),$$

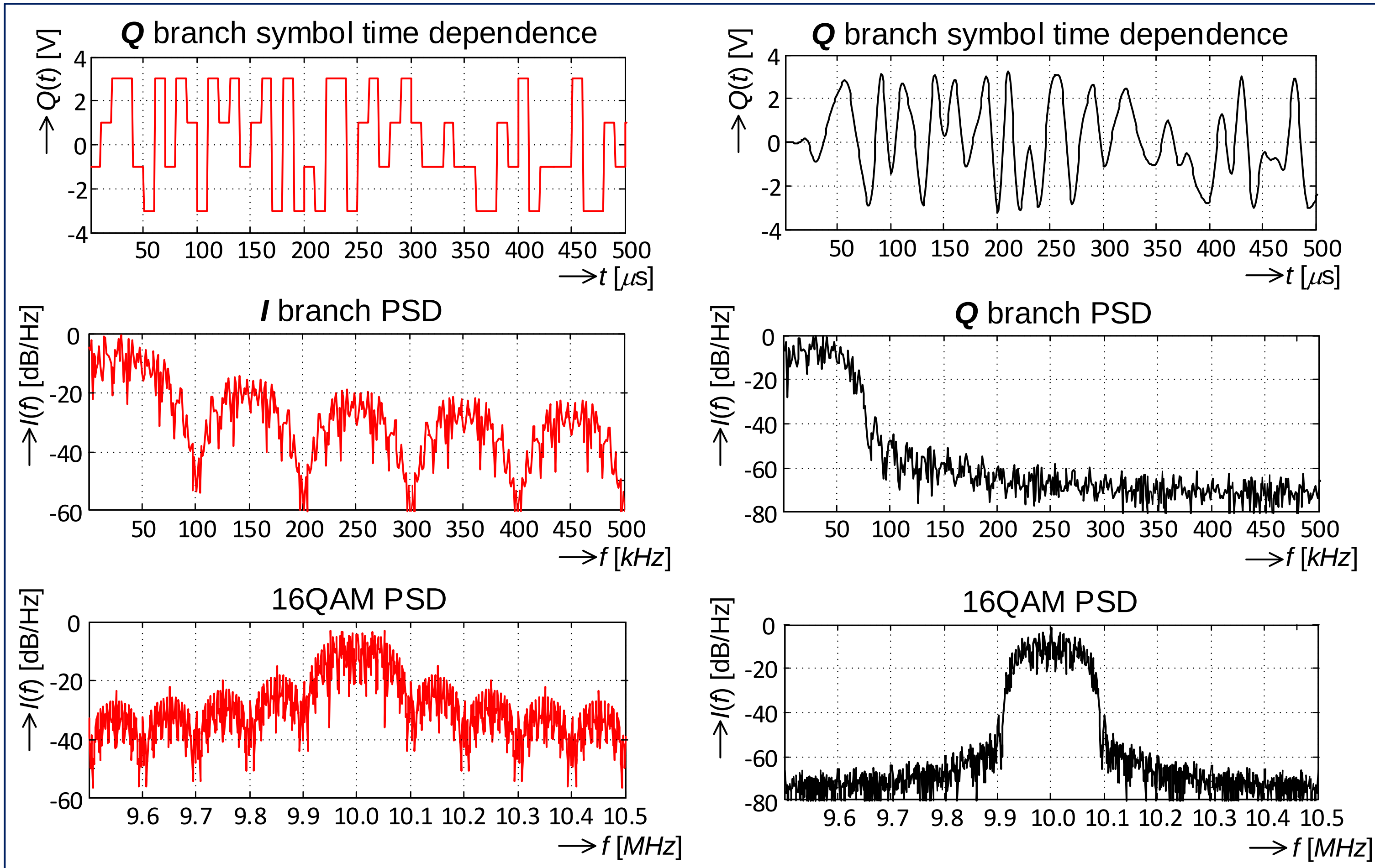
where A_{avg} is an average symbol amplitude.

*M*QAM (*M-ary Quadrature Amplitude Modulation*)



16 QAM signals
red: no LPF filter
black: SRC filter

*M*QAM (*M-ary Quadrature Amplitude Modulation*)



MQAM error probability

$$P_b = \frac{2}{\log_2 M}\left(1 - \frac{1}{\sqrt{M}}\right)\mathrm{erfc}\left[\sqrt{\frac{3}{2}\frac{\log_2 M}{(M-1)}}\sqrt{\frac{E_b}{N_0}}\right]$$

For M = 16 we get

$$P_b = \frac{3}{8}\mathrm{erfc}\left(\sqrt{\frac{2}{5}}\sqrt{\frac{E_b}{N_0}}\right)$$

MPSK error probability

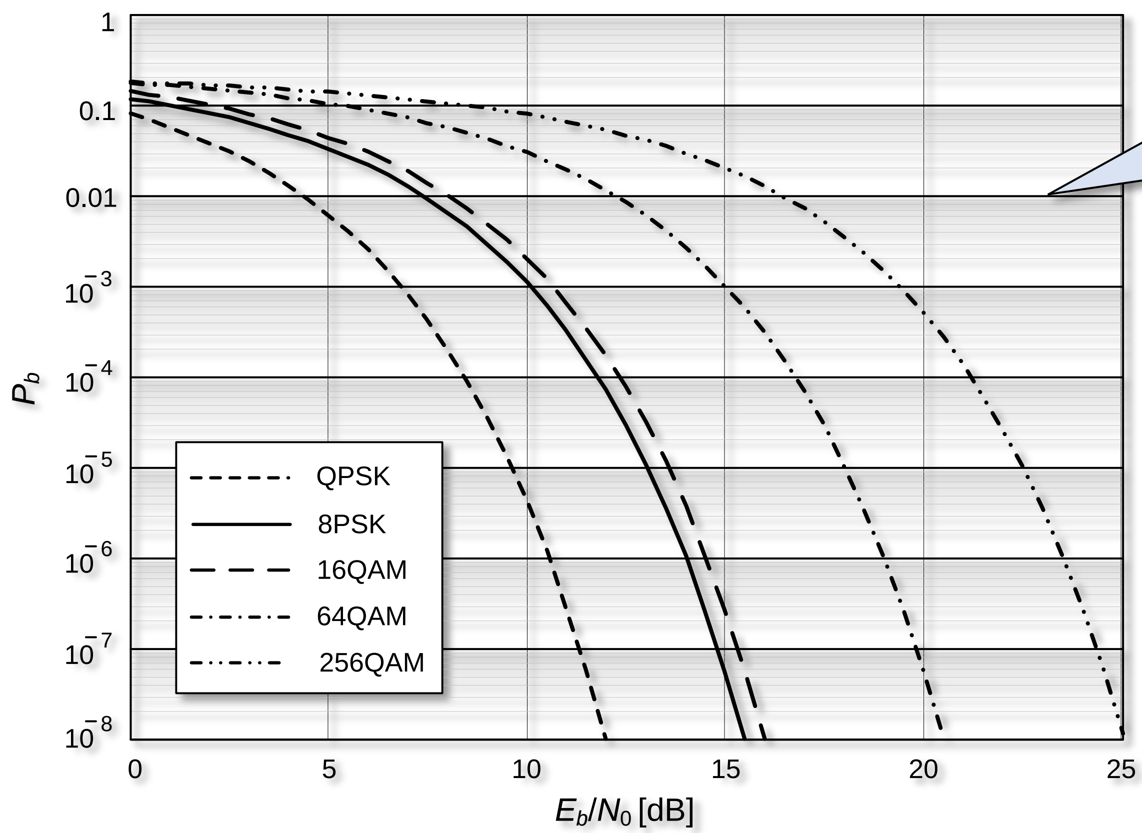
$$P_b = \frac{1}{\log_2 M}\mathrm{erfc}\left[\sin\left(\frac{\pi}{M}\right)\sqrt{\log_2 M}\sqrt{\frac{E_b}{N_0}}\right]$$

If M = 8 then

$$P_b = \frac{1}{3}\mathrm{erfc}\left[\sin\left(\frac{\pi}{8}\right)\sqrt{3}\sqrt{\frac{E_b}{N_0}}\right]$$

QPSK error probability

$$P_b = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{A_c^2 T_b}{2N_0}}\right) = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$



Bit error probability for AWGN channel

8. Multicarrier Modulation

- A transmitted bitstream is subdivided into many different substreams, which are send over different subchannels.
- The data rate and bandwidth on each of the subchannels is much less than the total data rate and the total bandwidth of the system.
- The number of substreams is chosen to ensure that bandwidth of each subchannel is less than the coherence bandwidth of the channel $\Rightarrow$ flat fading.
- Due to long symbol duration, the ISI on each subchannel is small. It can be completely suppressed using a cyclic prefix.

Typical implementation: Ortogonal frequency division multiplexing (OFDM), Discrete multitone (DMT),

MM application: Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB), Wireless LANs (802.11a, g), Digital subscriber lines (DSL).

The orthogonality of the subchannels can be impaired by frequency offset and timing jitter.

Let B be a baseband bandwidth, R be a desired data rate and N be a number of subchannels.

The coherence bandwidth and the subchannel bandwidth are assumed to be $B_c \geq B$ and $B_N = B/N \ll B_c$ respectively.

Nonovelapping channels

For nonoverlapping channels we set $f_n = f_c + n2B_N, \quad n = 0,1,\cdots,N-10$. The transmitted signal over one symbol time T_N is

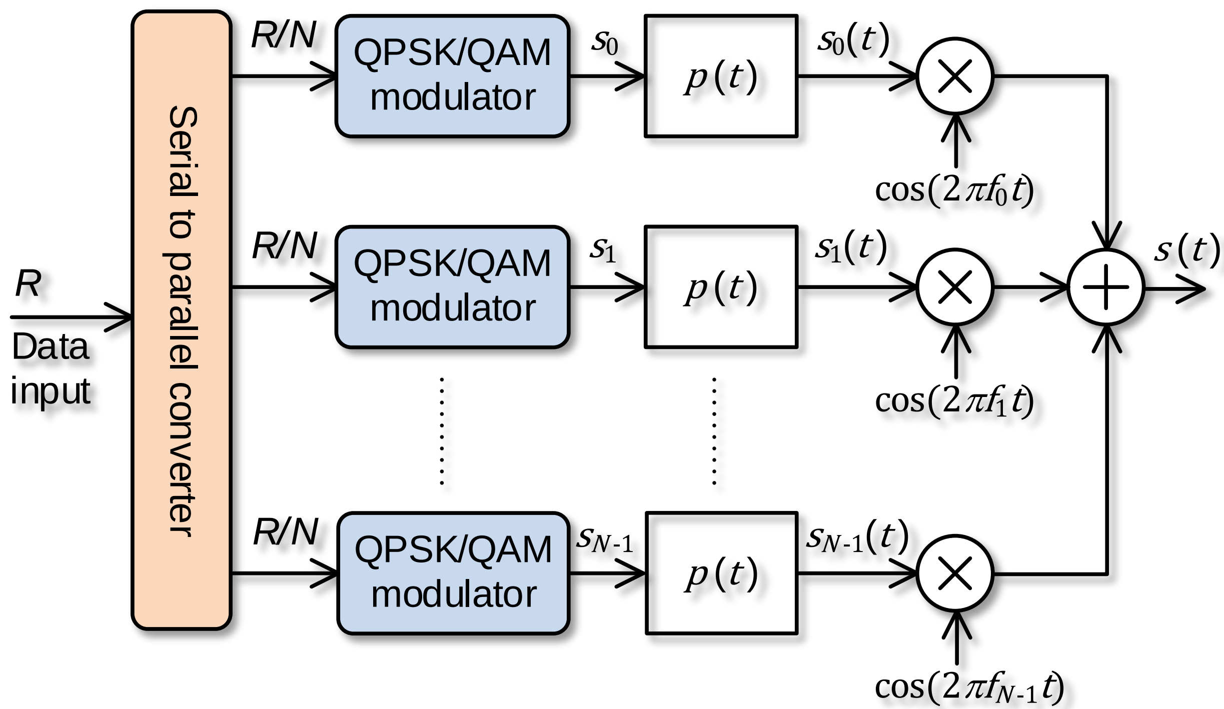
$$s(t) = \mathrm{Re}\left\{\sum_{n=0}^{N-1} s_n p(t)\, e^{j2\pi f_n t}\right\}$$

where s_n is the complex symbol associated with the n-th subcarrier. If $p(t)$ is a raised-cosine pulse, we get $T_N = (1+\beta)/2B_N$ and If $p(t)$ is a rectangular pulse $T_N = 1/2B_N$.

A total passband bandwidth: $N2B_N = 2B$.

Advantage: small frequency offsets and timing jitter have insignificant impact on the orthogonality of the subchannels.

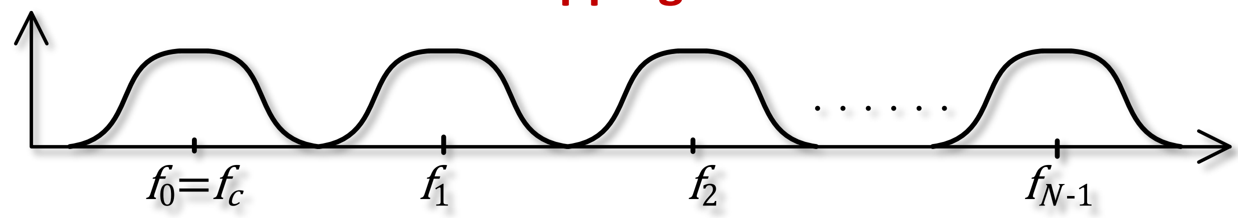
Multicarrier transmitter



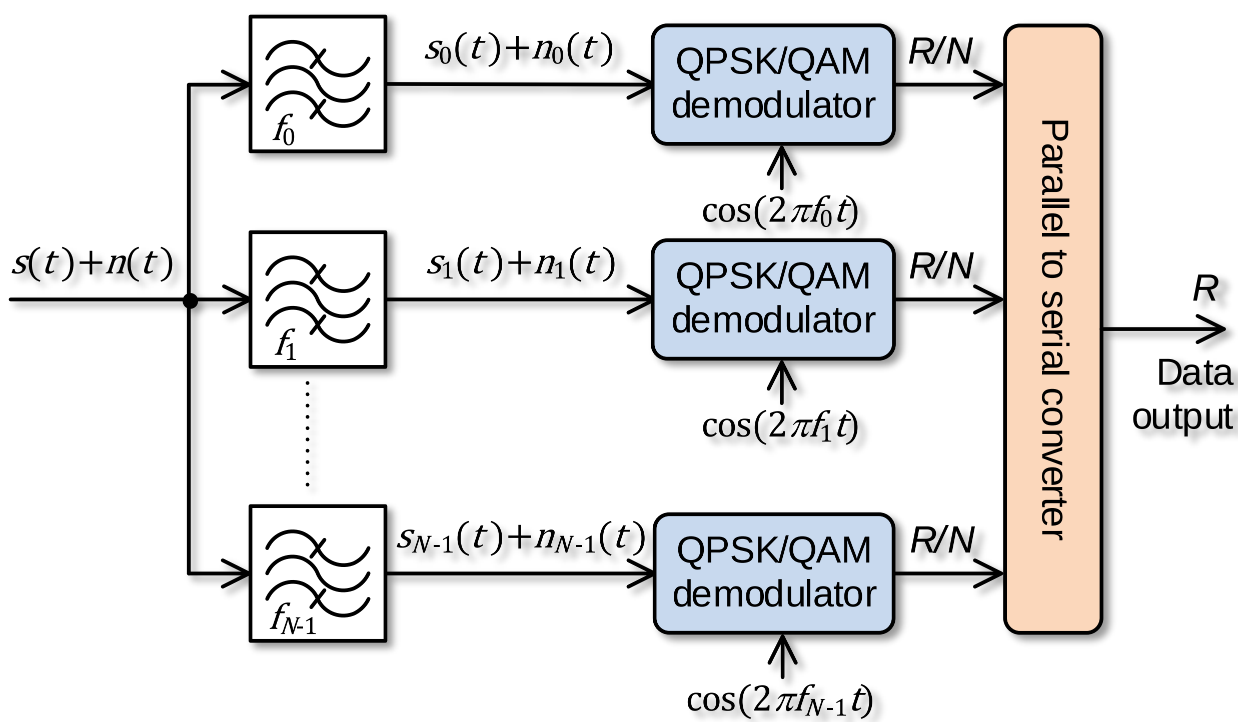
8. Multicarrier Modulation

Orthogonal frequency division multiplexing

Nonovelapping channels



Multicarrier receiver

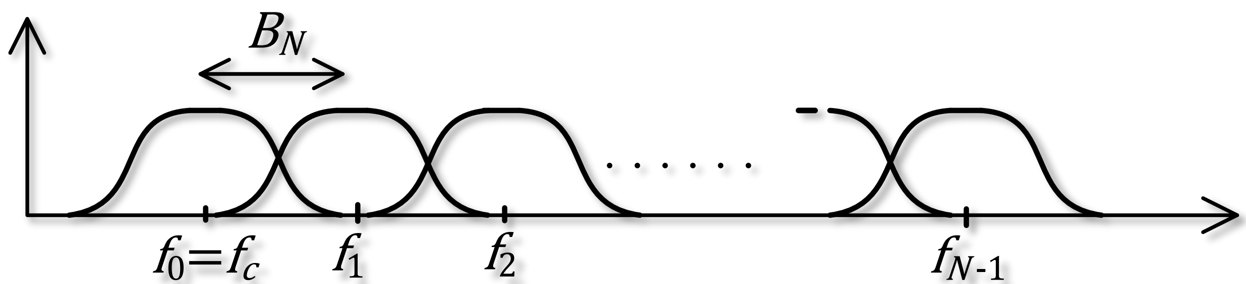


Ovelapping channels

The baseband subcarriers $\{\cos(2\pi i T_N), i = 0,1,\cdots\}$ form a set of orthonormal basis functions on the interval $[0, T_n]$.

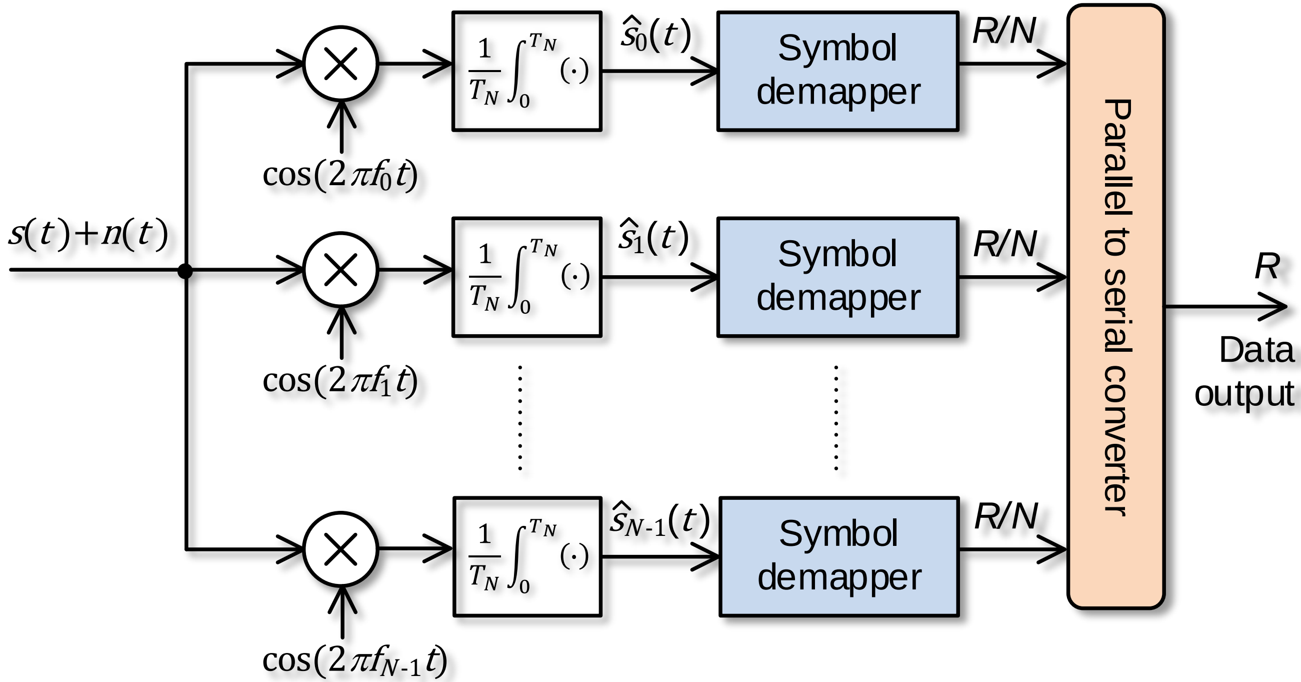
There is no set of subcarriers with a smaller frequency separation that form an orthonormal set on $[0, T_n] \Rightarrow$ minimum frequency separation $\Delta f = 1/T_N$.

If we use raised cosine pulses with $\beta = 1$, we will have $T_N = 1/B_N$ and $\Delta f = B_N \Rightarrow$ passband subchannels will overlap



$\Rightarrow$ a different receiver structure is needed.

Multicarrier receiver (in phase branch) for overlapping subcarriers



$$\hat{s}_i = \frac{1}{T_N}\int_0^{T_N}\left[\sum_{j=0}^{N-1} s_j \cos\left(2\pi f_j t\right)\right]\cos(2\pi f_i t)dt$$

For simplicity, let pulse shapes will be rectangular and in-phase signaling only be used, so that s_j be real and modulated with a cosine carrier.

Identical structure using sine carriers would be used to demodulated the quadrature signal component. The passband subcarriers are $f_j = f_c + j/T_N$

$$\hat{s}_i = \frac{1}{T_N}\int_0^{T_N}\left[\sum_{j=0}^{N-1} s_j \cos\left(2\pi f_j t\right)\right]\cos(2\pi f_i t)dt$$

$$= \frac{1}{T_N}\sum_{j=0}^{N-1} s_j \int_0^{T_N} \cos[2\pi(f_c + j/T_N)] \cos[2\pi(f_c + i/T_N)t]\, dt$$

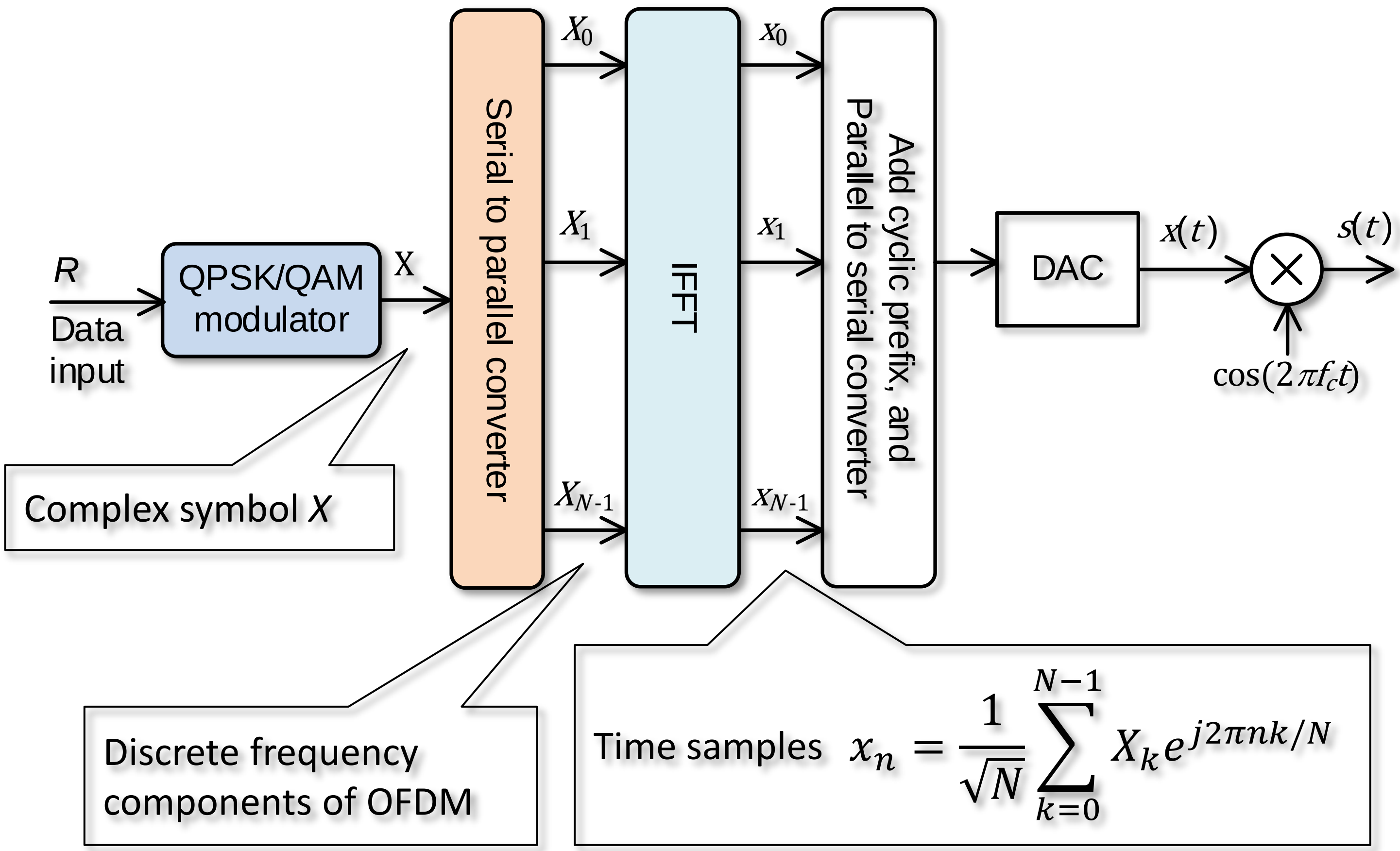
$$= \frac{1}{2T_N}\sum_{j=0}^{N-1} s_j \int_0^{T_N} \cos[2\pi(j-i)t/T_N]dt + \int_0^{T_N} \cos\{2\pi[2f_c + (j+i)/T_N]t\}dt$$

= 0 for $f_c \gg 1/T_N$

$$\approx \frac{1}{2}\sum_{j=0}^{N-1} s_j \delta(j-i) = \frac{1}{2}s_i$$

Requirement for separate modulators and demodulators on each subchannel is unrealizable for most of systems ⇒ discrete implementation of OFDM using FFT and IFFT.

Discrete multitone transmitter

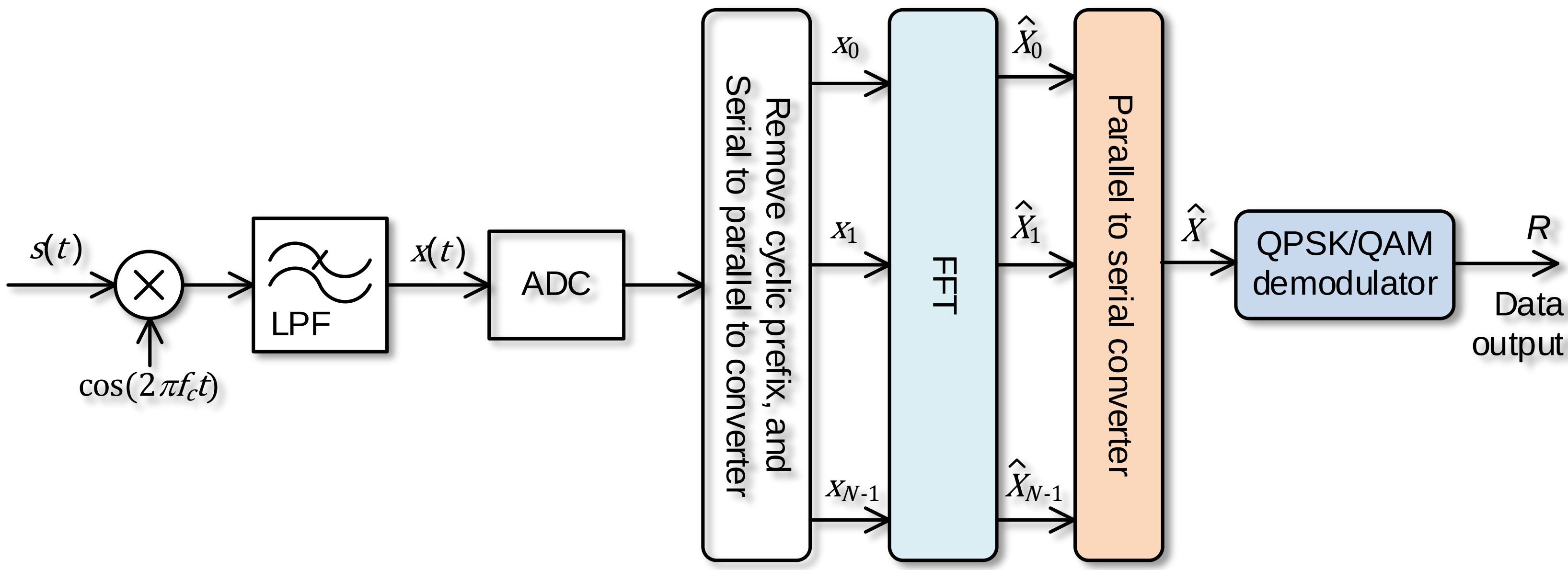


- **Baseband OFDM signal at the DAC output is**

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi kt/N} \quad 0 \leq t \leq T_N$$

- **The subcarrier frequencies are** $f_i = \frac{i}{T_N}, i = 1,2,\cdots,N-1$
- **The discrete time values** $\{x_0,\cdots,x_{N-1}\}$ **represent samples of** $x(t)$ **every** T_N/N **seconds.**
- **The baseband OFDM signal** $x(t)$ **is upconverted to the carrier frequency** f_c**, to get RF signal** $s(t)$**.**
- **Symbol duration** T_N **is chosen sufficiently large** $(B/N \ll B_c)$ **to remove most of ISI caused by the channel.**
- **ISI can be completely removed by either adding a guard time equal to the channel delay spread, or by adding a cyclic prefix after the IFFT.**

Discrete multitone receiver



Cyclic prefix (CP): let the channel delay spread has a maximum value of $\mu T_N/N$, (T_N/N is the sampling rate). The CP increases the number of samples in the $\{x_n\}$ sequence to $N+\mu$.

The new DAC input is $\{\underbrace{x_{N-\mu},\cdots,x_{N-1}}_{cyclic\ prefix},x_0,x_1,\cdots,x_{N-1}\}$.

Let us consider a received signal $r(t) = x(t) * c(t)$, where $c(t)$ is the channel impulse response. Assuming that the input to the FFT is the circular convolution of $\{x_n\}$ and $\{c_n\}$ we can introduce:

- By convolving $\{x_{N-\mu}, \cdots, x_{N-1}, x_0, x_1, \cdots, x_{N-1}\}$ with $\{c_0, c_1, \cdots, c_\mu\}$ we get $\{r_n\}$ with duration $N + \mu$ $\Rightarrow$ adding the CP at the transmitter converts the circular convolution associated with the FFT to a linear convolution.
- The FFT output is $\hat{X}_k = C_k X_k$ where C_k is the FFT of $\{c_0, c_1, \cdots, c_\mu\} \Rightarrow$ the effects of the channel $c(t)$ can be completely removed by frequency equalization by multiplying each $\hat{X}_k$ by $1/C_k$.

- **Each subchannel is relatively narrowband $\Rightarrow$ the effect of delay spread is mitigated.**
- **Each subchannel can suffer from flat-fading, which can cause large BERs on some of the subchannels $\Rightarrow$ necessity of frequency equalization, precoding, coding across subchannels, or adaptive loading.**

Let $SNR = P_i\alpha_i/(N_0B)$, where P_i and α_i are the power and fading on i-th subcarrier, then precoding (frequency equalization at the transmitter side) will cause that the transmitted power in the i-th subchannel is $P_i/\alpha_i^2 \Rightarrow$ transmitter must know the subchannel fading α_i. The received then signal will be $P_i\alpha_i^2/\alpha_i^2 = P_i$.

Adaptive loading is based on the adaptive modulation technique, where the data rate and power are assigned to each subchannel relative to that subchannel gain $\Rightarrow$ knowledge of the subchannel fading $\{\alpha_i, i = 1,2,\cdots, N\}$ is required.

Spread spectrum performance

- Increases signal bandwidth to reduce ISI and narrowband interference.
- Enables the system resistance to a narrowband jamming.
- Allows hiding the transmitted signal below the noise floor (suitable for military applications).
- With a RAKE receiver provides a form of diversity, known as code diversity.

Spread spectrum forms

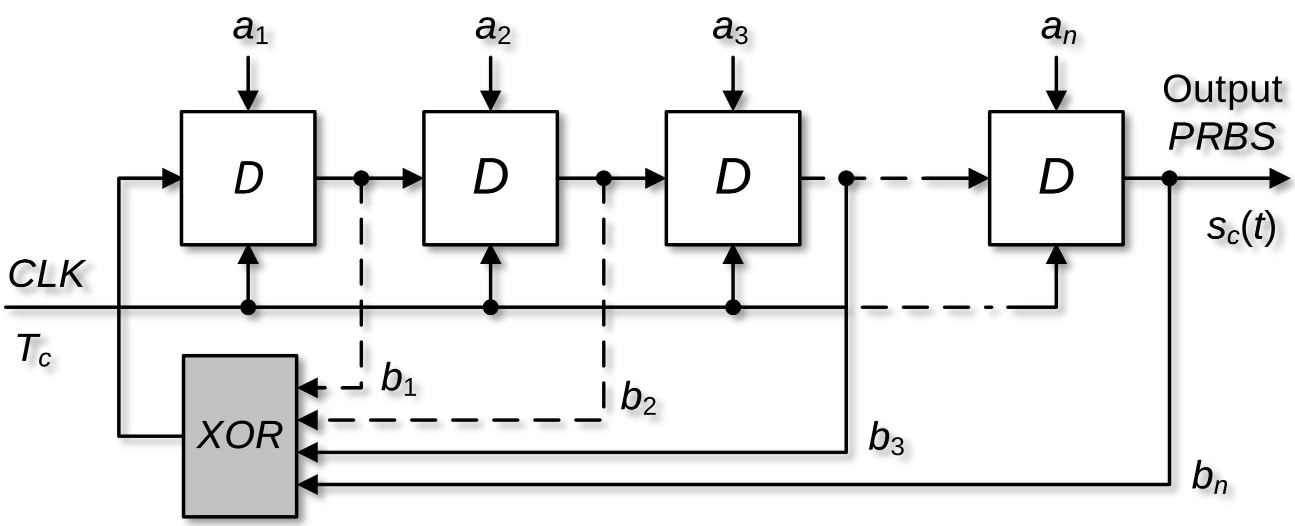
- Direct sequence: (more common) the data signal is modulated (multiplied) by a pseudorandom bit sequence (a spreading sequence).
- Frequency hopping: the carrier frequency rapidly changes among many distinct frequencies occupying a large spectral band.

In a **spread spectrum direct sequence (SSDS) system** the data signal $s_b(t)$ is multiplied by a **pseudo-random Binary sequence (PRBS)** $s_c(t)$.

The bit duration of the data signal T_b and bit duration of PRBS satisfy the condition $T_b = KT_c, K \in \mathbb{R}$. $R_c = 1/T_c$ is the **chip rate**.

Frequency response of the transmitted signal $S(f) = \mathcal{F}\{s_b(t)s_c(t)\} = S_b(f) * S_c(f)$ has a 3 dB bandwidth $B = (K+1)B_b$, where B_b is the original signal bandwidth $\Rightarrow$ the **spreading factor** is $J = B/B_b \approx K$.

PRBS generator based on a shift register



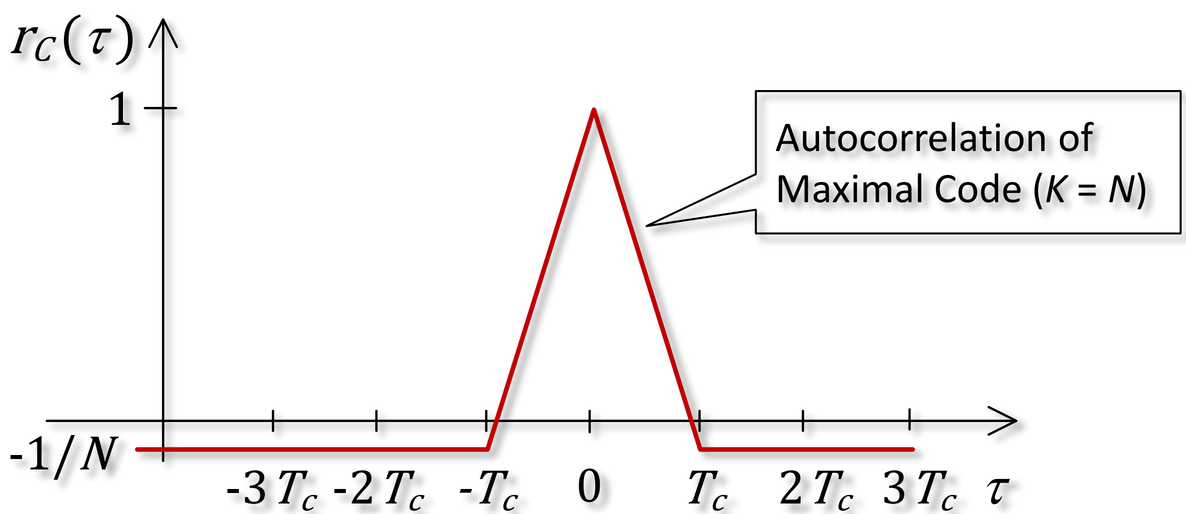
D: delay
a_i : initialization value
b_i : feedback

PRBS generator length: $N = 2^n - 1.$

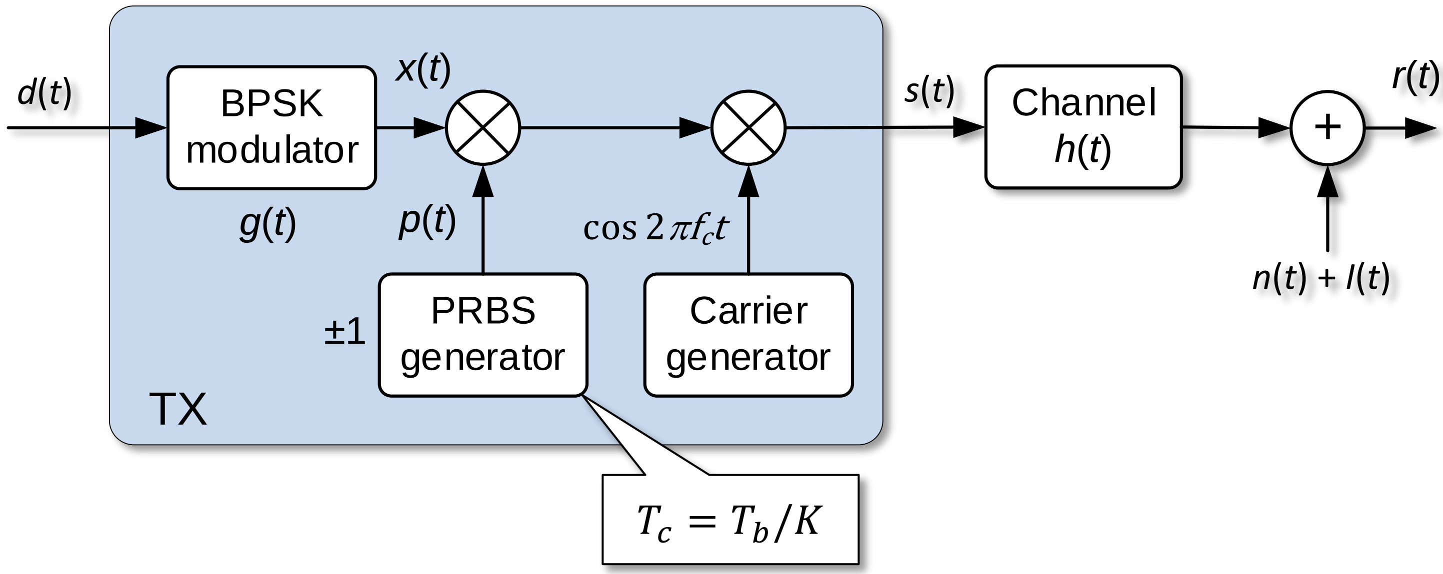
Maximal PRBS codes

The number of negative ones in a sequence is approximately equal to the number of ones. The autocorrelation of $s_c(t)$ is

$$r_c(\tau) = \frac{1}{T_b}\int_0^{KT_c} s_c(t)s_c(t-\tau)\,dt = \begin{cases} 1 - \dfrac{|\tau|(1+1/N)}{T_c} & |\tau| \le T_c \\ -1/N & |\tau| > T_c \end{cases} \qquad (9.1)$$



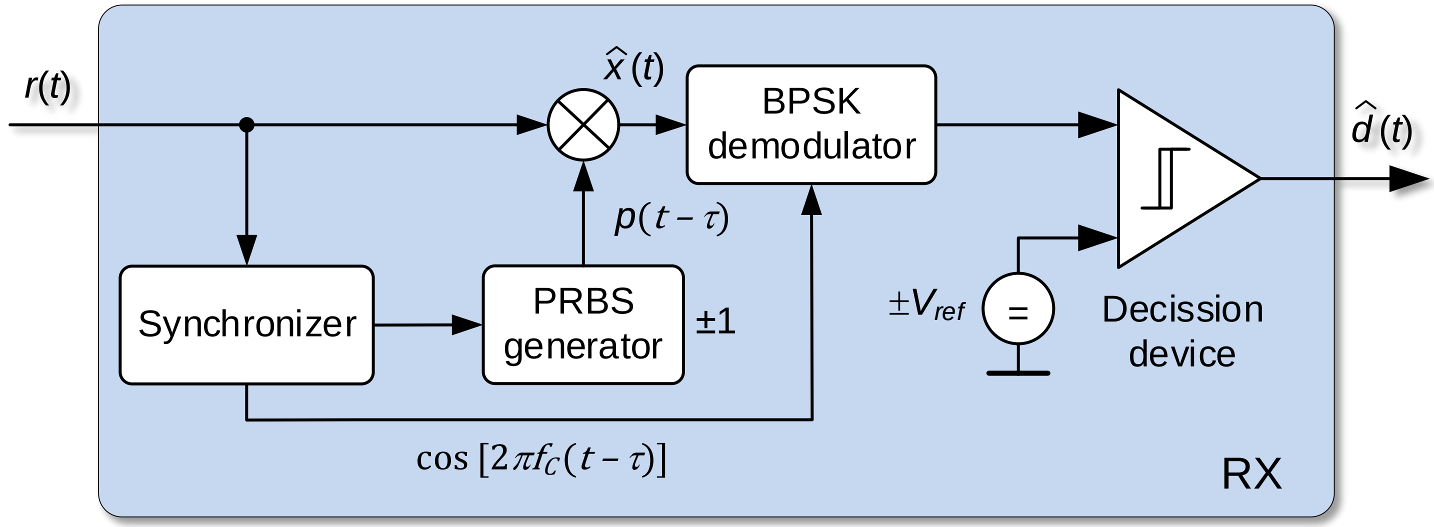
BPSK spread spectrum transmitters



Asumptions:

- **The BPSK modulator uses rectangular pulse shapes** $g(t) = \begin{cases} 1, & 0 \le t \le T_b \\ 0, & \text{otherwise} \end{cases}$
- **The baseband data signals are** $x(t) = \sum_k dg(t - kT_b)$ **,**
- **The channel introduces multipath** $h(t) = \sum_j \alpha_j \delta(t - \tau_j)$ **.**
- **The channel adds noise** $n(t)$ **and narrowband interference** $I(t)$**.**

BPSK spread spectrum receiver



Asumptions:

- The perfect synchronization generates PRBS $p(t-\tau)$.
- The synchronizer synchronizes either to the first multipath component above a given threshold or the strongest multipath component.
- The signal $\hat{x}(t)$ passes through a BPSK demodulator after despreading.

$$\hat{x}(t) = [x(t)p(t)\cos(2\pi f_c t) * h(t)]\, p(t-\tau) + n(t)p(t-\tau) + I(t)p(t-\tau)$$

Narrowband interference rejection (single TX, single RX)

Let $h_i(t) = \delta(t) \Rightarrow \tau = 0$. **Assuming that** $f(t) * \delta(t) = f(t)$ **and** $[x(t)p(t)]p(t) = x(t)$ **since** $p^2(t) = 1$ **we have**

$$\hat{x}(t) = x(t)\cos(2\pi f_c t) + n(t)p(t) + I(t)p(t). \tag{9.2}$$

Demodulator output signal is

$$\hat{d}_k = \frac{1}{T_b}\int_0^{T_b} d_k \cos^2(2\pi f_c t)\, dt + \frac{1}{T_b}\int_0^{T_b} n(t)\, p(t)\cos(2\pi f_c t)\, dt$$
$$+ \frac{1}{T_b}\int_0^{T_b} I(t)\, p(t)\cos(2\pi f_c t)\, dt = \frac{1}{2} d_k + n_k + \underbrace{I_k}_{0} \tag{9.3}$$

Integrating the product $I(t)\, p(t)$ **and the cosine term** $\cos(2\pi f_c t)$ **yield approximately zero because** $T_c = T_b/K$ **and** $f_c \gg \frac{1}{T_b} \Rightarrow I_k \approx 0$.

Delayed component reception (single TX, single RX)

$h(t) = \alpha_1 \delta(t) + \alpha_2 \delta(t - \tau_0)$. **Let $\alpha_1 > \alpha_2$ and $I(t) = 0$. Then**

$$\hat{x}(t) = \alpha_1 x(t)\cos(2\pi f_c t) + \alpha_2 x(t-\tau_0)p(t-\tau_0)p(t)\cos[2\pi f_c(t-\tau_0)] + n(t)p(t). \quad (9.4)$$

Similarly to (9.3) **we can get**

$$\hat{d}_k = \frac{1}{2}\alpha_1 d_k + \frac{1}{2}\alpha_2 d_{k0} + n_k,$$

where

$$d_{k0} = d_{k-k_0}\cos(2\pi f_c \tau_0) \underbrace{\frac{1}{T_b}\int_0^{T_b} p(t)\,p(t-\tau_0)dt}_{\rho(\tau_0)}, \quad (9.5)$$

and d_{k-k_0} is the symbol corresponding to time $t - \tau_0$.

If N = K and $\tau_0 > T_c$ then $d_{k-k_0}\cos(2\pi f_c\tau_0)\,\rho(\tau_0) = -d_{k-k_0}\cos(2\pi f_c\tau_0)\,/K$
$\Rightarrow$ the power of all multipath components at delays greater than T_c is reduced by the spreading gain $\Rightarrow$ most of the ISI is removed.

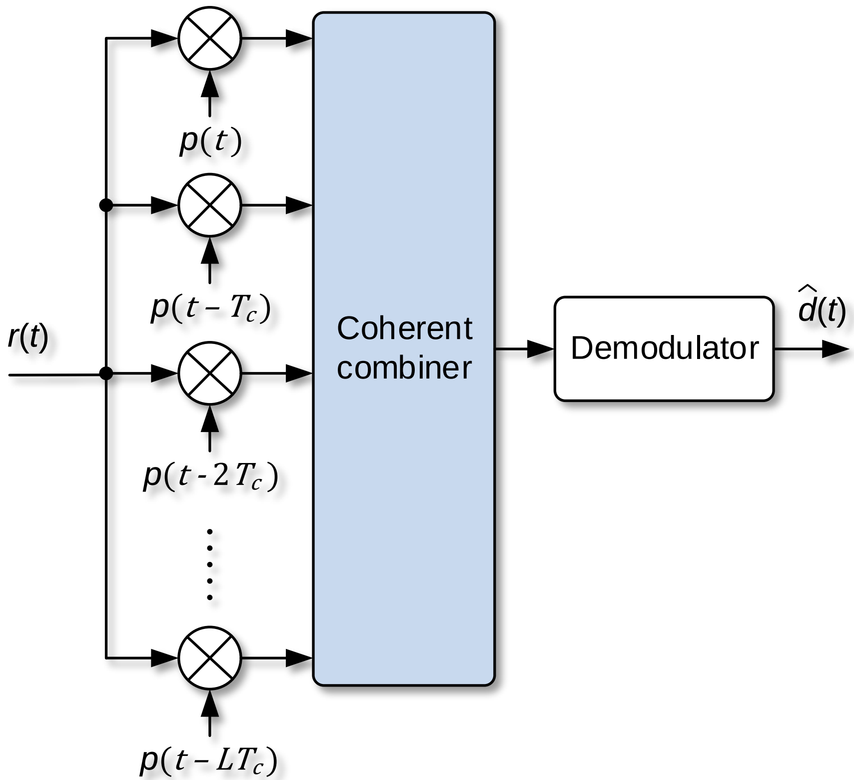
Rake receiver

Diversity receiver uses the autocorrelation properties of the PRBS to coherently combine all multipath components.

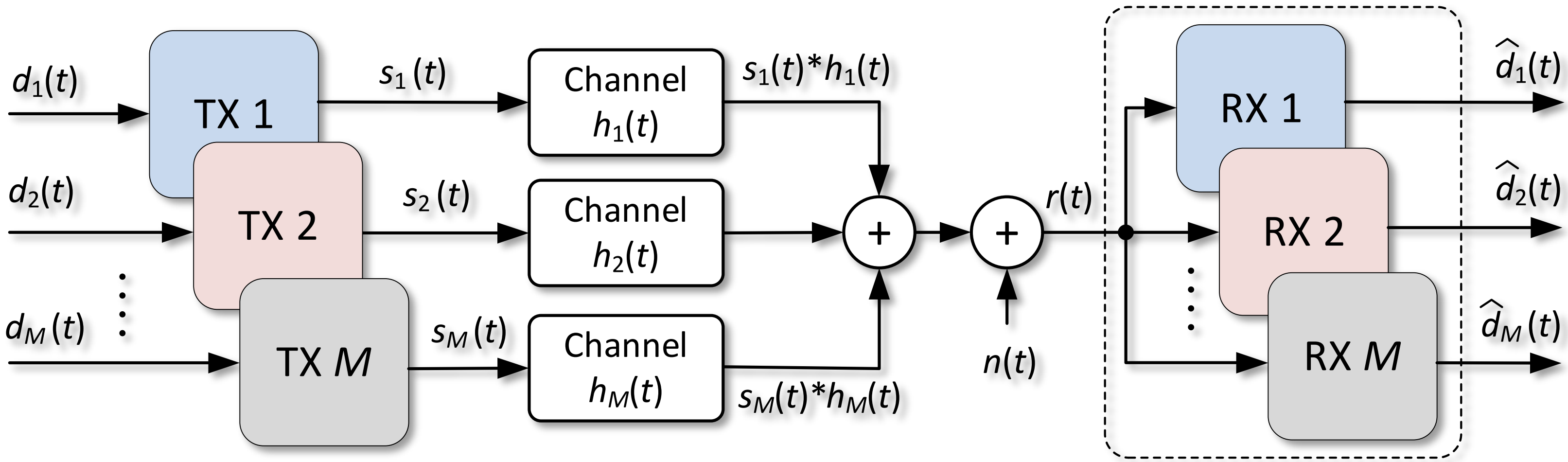
- **It has several branches synchronized to a different multipath component. The time delay of the PRBS code between branches is T_c. If the received PRBS is delayed by more than a small fraction of T_c performance is significantly decreased $\Rightarrow$ precise tracking is required.**
- **Any of the combining techniques (selection combining, threshold combining, maximal ratio combining, equal-gain combining) may be used.**

- **Equal gain combining** (which co-phases the signals on each branch and then combines them with equal weighting) is the most common, as it does not require knowledge of the multipath amplitudes.

Rake receiver



- Multiple access transmission assumes several separate TX and single common RX.



- Each user has their own unique code to spread the transmitted signal.
- The users occupy the same bandwidth.
- The spreading codes (orthogonal or semi-orthogonal) have much higher chip rate than the data rate.

- **To eliminate multiple access (MAC) interference, the cross correlation between the codes assigned to user i and user j**

$$\rho_{ij}(\tau) = \frac{1}{T_b}\int_0^{T_b} p_i(t)\, p_j(t-\tau)dt$$

should satisfy $\rho_{ij}(\tau) \approx 0$.

- **Orthogonal codes (e.g. Walsh-Hadamard codes) can support limited number of users, but MAC interference are eliminated because $\rho_{ij}(\tau) = 0$.**

- **Semi-orthogonal codes (e.g. Gold codes) can support more users than the orthogonal codes, but they exhibit nonzero MAC interference because**

$$\rho_{ij}(\tau) \approx \frac{1}{\sqrt{J}}, \tag{9.6}$$

where J is spreading (bandwidth expansion) factor.

Interference rejection

Let us consider a transmission with semi-orthogonal codes and no multipath where $h_j(t) = \alpha_j \delta(t), j = 1, 2, \cdots, M$. The desired signal component is given by (9.2) **and** (9.3). **The interference signal $I_i(t)$ for i**-th **receiver branch is**

$$I_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^{M} \frac{1}{T_b} \int_0^{T_b} \left[x_j(t) p_j(t) \cos(2\pi f_c t) * h_j\ (t) \right] \cos(2\pi f_c t)\, p_i(t) dt$$

$$= \sum_{\substack{j=1 \\ j \neq i}}^{M} \alpha_j d_j \frac{1}{T_b} \int_0^{T_b} p_i(t) p_j(t) \left[\frac{1}{2} + \frac{\cos(4\pi f_c t)}{2} \right] dt \approx \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^{M} \frac{\alpha_j d_j}{\sqrt{G}}, \quad \text{(9.7)}$$

where $\sqrt{G} \triangleq 1/\rho_{ij}(\tau)$, and α_j is the path gain of the j- th **user's channel. Integral $\int_0^{T_b} p_i(t) p_j(t) \cos(4\pi f_c t)\, dt \approx 0$ because $2f_c \gg 1/T_b$ and $p_i(t)p_j(t)$ at any chip period is ± 1.**

The interference $I_i(t)$ and signal $\hat{x}_i(t)$ are attenuated by different path gains. If $\alpha_j \gg \alpha_i$ the MAC interference can be quite large.

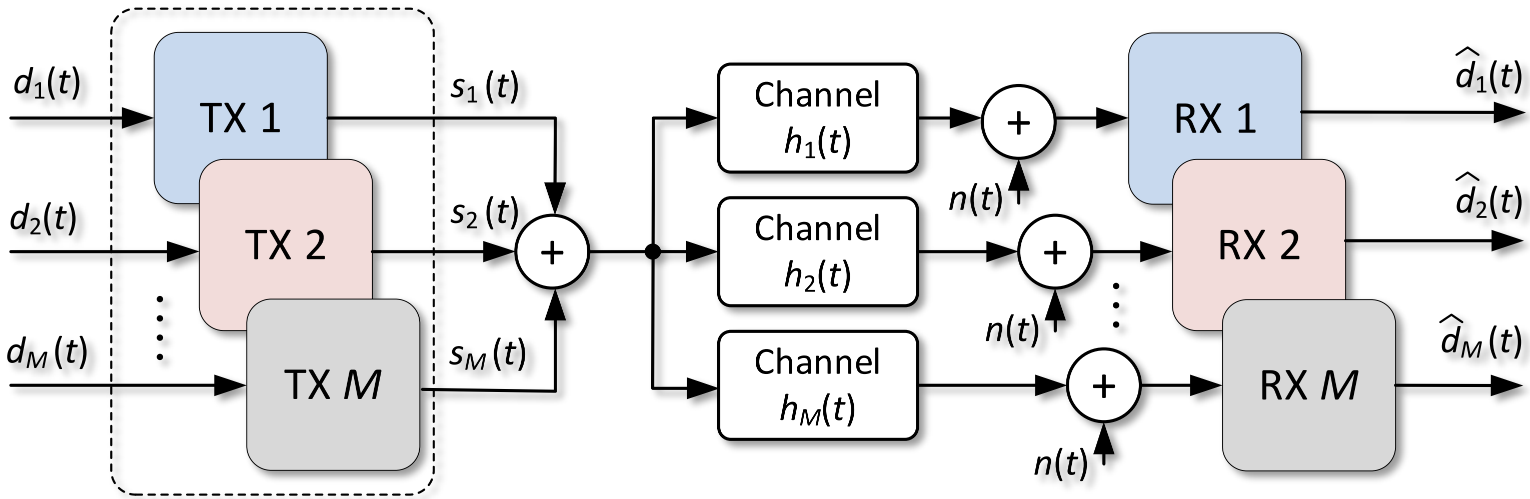
The received signal power on the i- th branch is $S_i = \alpha_i^2 S$, where S is the transmitted power.

The interference power is $I_i = \alpha_j^2 (M-1)/4G$.

The signal to-interference power ratio then is

$$\frac{S_i}{I_i} = \frac{\alpha_i^2 G}{\alpha_j^2 4(M-1)} \ll \frac{G}{(M-1)}. \tag{9.8}$$

- **Broadcast transmission assumes single common TX and several separate RX.**



- **Let us consider a broadcast with semi-orthogonal codes and no multipath where $h_j(t) = \alpha_j \delta(t), j = 1, 2, \cdots, M$. The desired signal component is given by** (9.2) **and** (9.3). **The interference signal $I_i(t)$ for i-th receiver is**

$$I_i(t) = \frac{\alpha_i}{2} \sum_{\substack{j=1 \\ j \neq i}}^{M} \frac{d_j}{\sqrt{G}}, \qquad \frac{S_i}{I_i} = \frac{G}{4(M-1)}. \tag{9.9}$$

10. Channel diversity

Channel diversity goal: achieving more reliable detection by sending signals that carry the same information through multiple independent paths.
Reliable communication is possible as long as one of the paths is strong (it has high SNR).

Channel diversity types:

- **Time diversity:** coded symbols are dispersed over time in different coherence periods so that different parts of the codewords experience independent fades.
- **Frequency diversity:** signals are transmitted over frequency-selective channel with different fading statistics
- **Space diversity:** identical signals propagate between sufficiently spaced multiple transmit and/or receive antennas.

Assumptions for analysis: the receiver has perfect knowledge of the channel gains and can coherently combine the received signals in the diversity paths.

Time diversity: achieved by averaging the fading of the channel over time. The channel is highly correlated across consecutive symbols ⇒ to ensure that the coded symbols are transmitted through independent fading gains **interleaving** of codewords is required.

Let the codeword $\mathbf{x} = [x_1, x_2, \cdots, x_L]$ is transmitted over a flat fading channel. The received signal is

$$y_l = x_l h_l + w_l, \qquad l = 1, \cdots, L$$

where w_l is additive noises and L is called the number of ***diversity branches.*** If consecutive symbols x_l are transmitted sufficiently far apart in time, we can assume that the h_l are independent.

Repetition code

Code, in which $x_l = x_1$ and

$$\mathbf{y} = \mathbf{h}x_1 + \mathbf{w}, \tag{10.1}$$

$\mathbf{x} = [x_1, x_2, \cdots, x_L]^\mathsf{T}$, $\mathbf{h} = [h_1, h_2, \cdots, h_L]^\mathsf{T}$, $\mathbf{w} = [w_1, w_2, \cdots, w_L]^\mathsf{T}$, Let the BPSK modulation, with $x_1 = \pm a$ is used. The error probability is

$$P(e) = Q\left(\sqrt{\frac{2\|\mathbf{h}\|^2 a^2}{N_0}}\right), \tag{10.2}$$

where $\|\mathbf{h}\|^2 = \sum_{l=1}^{L} |h_l|^2$, $h_l \in (0,1)$ is a sum of the squares of $2L$ independent real Gaussian random variables (squares of the real and imaginary parts) ⇒ Chi-square distributed variables with $2L$ degrees of freedom, which density is

$$f(x) = \frac{1}{(L-1)!} x^{L-1} e^{-x}, \qquad x \geq 0.$$

Then

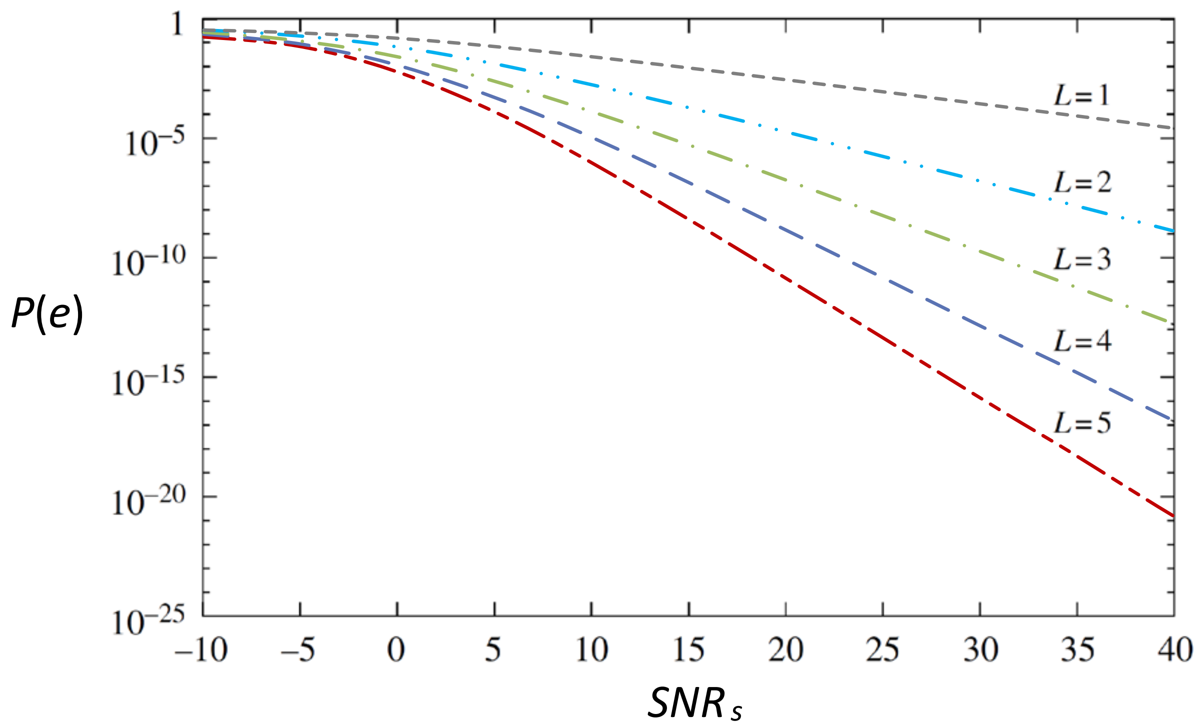
$$\overline{P(e)} = \int_0^\infty Q\left(\sqrt{2xSNR_s}\right) f(x) dx = \left(\frac{1-\mu}{2}\right)^L \sum_{l=0}^{L-1} \binom{L-1+l}{l} \left(\frac{1+\mu}{2}\right)^l, \tag{10.3}$$

where $\mu = \sqrt{\left(\frac{SNR_s}{1+SNR_s}\right)}$ and $SNR_s = a^2/N_0$ is SNR per symbol time.

At high SNR_s we get $(1+\mu)/2 \approx 1$ and $(1-\mu)/2 \approx 1/4SNR_s$

$$\sum_{l=0}^{L-1} \binom{L-1+l}{l} = \binom{2L-1}{L} \Rightarrow \overline{P(e)} = \binom{2L-1}{L} \frac{1}{(4SNR_s)^L} \quad (10.4)$$

The error probability decreases with the L - th power of SNR_s.

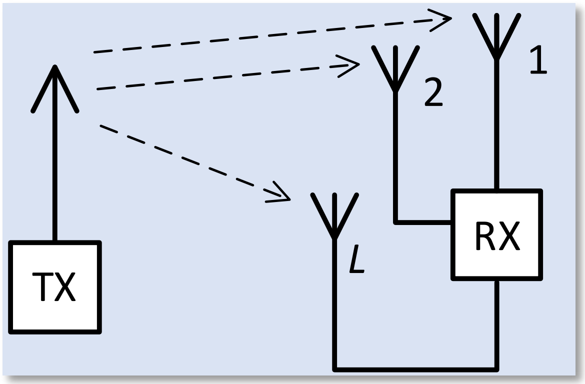


If the antennas are deployed sufficiently far apart, the channel gains between different antenna pairs fade independently.

- **Receive diversity:** **multiple antennas are at the receiver side - single input multiple output (SIMO) channel.**
- **Transmit diversity:** **multiple antennas are at the transmitter side - multiple input single output (MISO) channel.**

Receive diversity

let us have 1 **transmit antenna and** L **receive antennas**



$$y_l[m] = h_l[m]x[m] + w_l,[m], \qquad l = 1, \cdots, L \tag{10.5}$$

and we want to detect $x[1]$ based on $y_1[1], \cdots, y_L[1] \Longrightarrow$ the same detection problem as in a repetition code.

The error probability of BPSK:

$$P(e) = Q\left(\sqrt{2\|\mathbf{h}\|^2 SNR_s}\right) \tag{10.6}$$

For the coherent combining at the receiver we can write

$$\|\mathbf{h}\|^2 SNR_s = L \cdot SNR_s \frac{1}{L}\|\mathbf{h}\|^2,$$

where $L \cdot SNR_s$ is an array gain and $\frac{1}{L}\|\mathbf{h}\|^2$ is the diversity gain.

Averaging of the channel responses over multiple independent signal paths, decrease the probability that the overall gain is small.

For very high L and fully correlated channel we have:

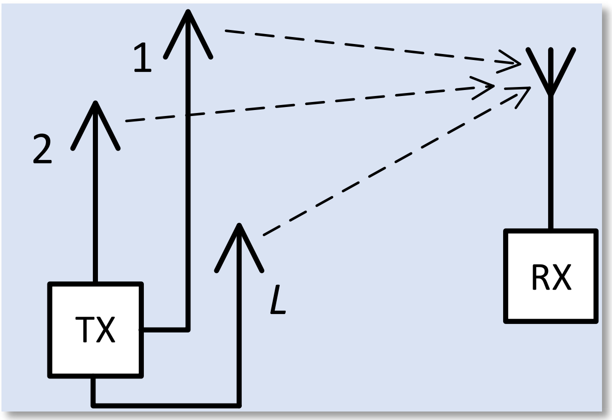
$$\frac{1}{L}\|\mathbf{h}\|^2 = \frac{1}{L}\sum_{l=1}^{L}|h_l[1]|^2 \to 1 \tag{10.7}$$

i.e. $\frac{1}{L}\|\mathbf{h}\|^2$ converges to 1.

Transmit diversity

We have L transmit antennas and 1 receive antenna.

This concept is suitable for mobile communications because it is less expensive to have multiple antennas at the base-station than to have multiple antennas at every mobile handset.



Let us transmit the same symbol over the L different antennas during L symbol periods.

During each symbol period only one antenna is turned on $\Longrightarrow$ only one antenna is used at a time and the coded symbols of the time diversity are transmitted successively over the different antennas $\Longrightarrow$ it offers the coding gain over the repetition code.

Alamouti scheme

It is designed for two TX and single RX antennas. With flat fading we have

$$y[m] = +h_1[m]x_1[m] + h_2[m]x_2[m] + w[m], \qquad l = 1, \cdots, L \tag{10.8}$$

where h_i is the channel gain for transmit antenna i. The Alamouti scheme transmits two complex symbols: $x_1[1] = a_1, x_2[1] = a_2$ and $x_1[2] = -a_2^*$, $x_2[2] = a_1^*$. If $h = h_i[1] = h_i[2], \;\; i = 1{,}2$.

Then (10.8) **can be experessed in a matrix form:**

$$[y_1[1]y_2[2]] = [h_1, h_2]\begin{bmatrix} a_1 & -a_2^* \\ a_2 & a_1^* \end{bmatrix} + [w_1[1]w_2[2]]. \tag{10.9}$$

We can rewrite (10.9) **to**

$$\begin{bmatrix} y_1[1] \\ y_2[2]^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} w_1[1] \\ w_2[2]^* \end{bmatrix}.$$

As the columns of the square matrix are orthogonal, the detection of a_1, a_2 can be decomposed into two separate procedures.

MIMO systems can significantly increase the data throughput of wireless systems without increasing transmit power or bandwidth.