Digital Signal Processing for Satellite Communications

Giulio Colavolpe

SPADiC Lab &
CNIT Research Unit
Dipartimento di
Ingegneria dell'Informazione
Università of Parma
Parma, Italy

January 2014

Outline

Outline

Complex envelope of a passband signal (1/3)

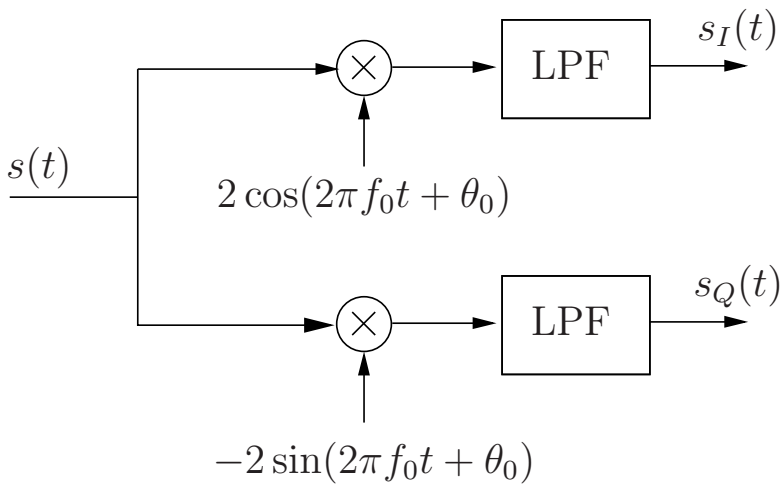
Let us consider a passband signal $s(t)$. For an arbitrary frequency f_0, usually taken as the carrier frequency, and an arbitrary phase θ_0, it can be expressed as

$$s(t) = s_I(t)\cos(2\pi f_0 t + \theta_0) - s_Q(t)\sin(2\pi f_0 t + \theta_0) = \mathcal{R}\left\{\widetilde{s}(t)e^{\jmath(2\pi f_0 t+\theta_0)}\right\}$$

where

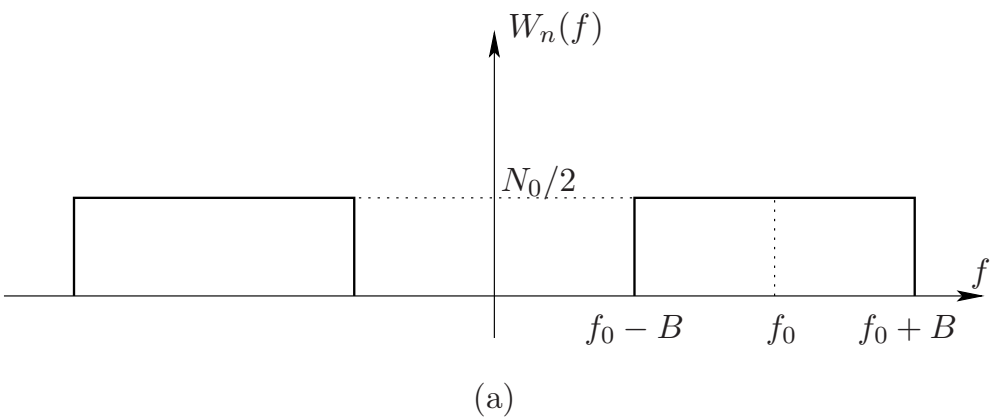
$$\widetilde{s}(t) = s_I(t) + \jmath s_Q(t)$$

is the complex envelope (or baseband equivalent signal) and can be extracted with the following scheme:

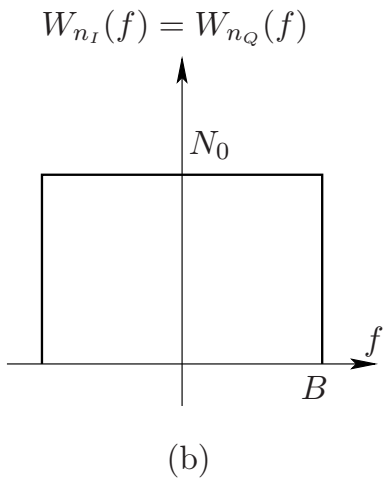


Complex envelope of a passband signal (2/3)

$s_I(t)$: in-phase component.
$s_Q(t)$: quadrature component.
The power of $s(t)$ is half that of $\widetilde{s}(t)$.
The complex envelope can be computed for both deterministic and stochastic signals.
Let us consider the case of a passband noise $n(t)$ with power spectral density (PSD) $W_n(t)$ shown in figure (a).



(a)

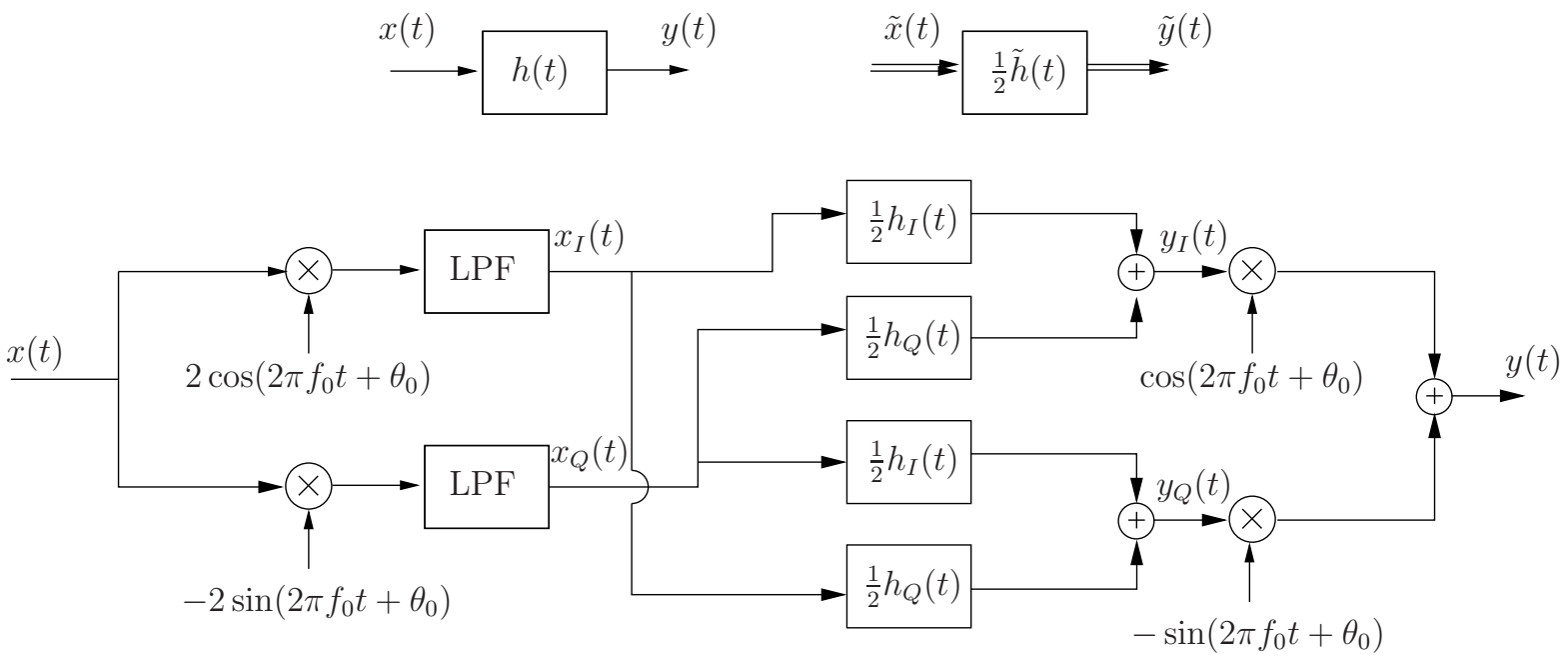


(b)

The PSD of the in-phase and quadrature components are shown in figure (b).

Complex envelope of a passband signal (3/3)

Baseband Processing Theorem:



In the following, I will always work with complex envelopes and avoid the use of $\tilde{}$ to denote them.

Discrete-time representation of a signal (1/2)

Representation of signals through vectors.
Let us consider the set of complex signals with support in (a, b) and finite energy. This set is denoted to as $L_2(a, b)$. Definitions:

- Inner (or Scalar) Product: $(x, y) = \int_a^b x(t)y^*(t)\,\mathrm{d}t$
- Energy: $E_x = (x, x) = \int_a^b x(t)x^*(t)\,\mathrm{d}t = \int_a^b |x(t)|^2\,\mathrm{d}t$
- Norm: $||x|| = \sqrt{E_x} = \sqrt{\int_a^b |x(t)|^2\,\mathrm{d}t}$
- Distance: $||x - y|| = \sqrt{E_{x-y}} = (x - y, x - y)^{1/2} = \sqrt{\int_a^b |x(t) - y(t)|^2\,\mathrm{d}t}$
- Linearly Independent Signals: $\sum_{i=1}^{M} c_i\, s_i(t) = 0$ is satisfied iff $c_i = 0 \quad \forall i$

Discrete-time representation of a signal (2/2)

Let us consider a subset of M signals $\{s_i(t)\}_{i=1}^{M}$ belonging to $L_2(a,b)$. By considering all linear combinations of these signals, we obtain a subspace S of the original $L_2(a,b)$ space. Hence, signals $\{s_i(t)\}_{i=1}^{M}$ form a basis of this subspace.
Different bases can be found. An orthonormal basis $\{\varphi_i(t)\}_{i=1}^{N}$, $N \leq M$, is such that:

$$\varphi_i(t) \in S \quad , \quad (\varphi_i, \varphi_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} ,$$

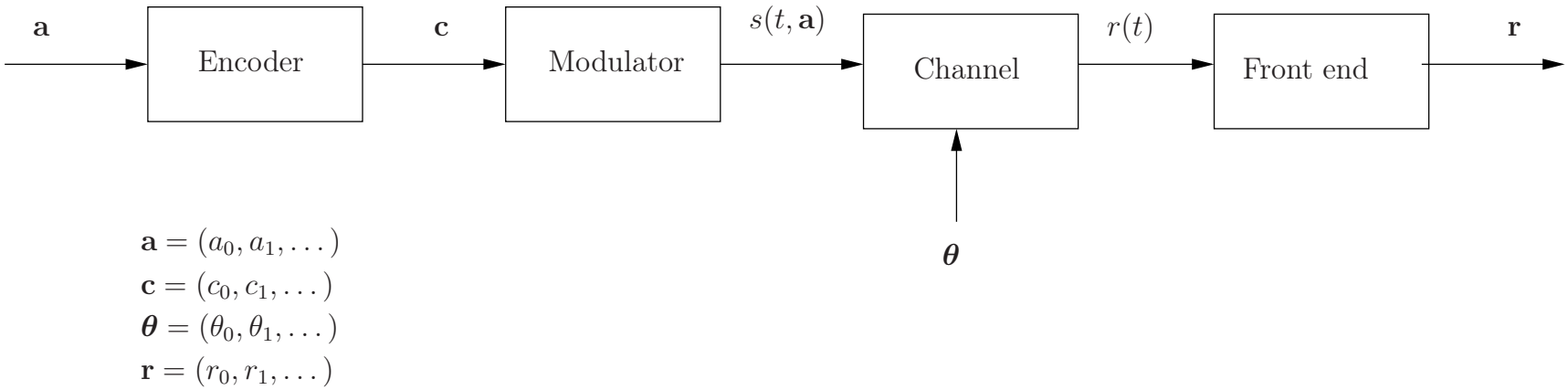
and every $x(t) \in S$ can be expressed as

$$x(t) = \sum_{i=1}^{N} x_i \varphi_i(t), \quad \text{with } x_i = (x, \varphi_i).$$

Hence, we may associate to each signal $x(t)$ a vector $\mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{C}^N$ such that

$$(x,y) = \mathbf{x}^T \mathbf{y}^*, \quad E_x = ||\mathbf{x}||^2, \quad ||x-y|| = ||\mathbf{x}-\mathbf{y}|| .$$

Detection Theory (1/3)



$\mathbf{a} = (a_0, a_1, \dots)$
$\mathbf{c} = (c_0, c_1, \dots)$
$\boldsymbol{\theta} = (\theta_0, \theta_1, \dots)$
$\mathbf{r} = (r_0, r_1, \dots)$

MAP Sequence Detection Strategy (minimizes the sequence error rate)

$$\widehat{\mathbf{a}} = \underset{\widetilde{\mathbf{a}}}{\operatorname{argmax}}\, P(\widetilde{\mathbf{a}}|\mathbf{r}) \tag{1}$$

MAP Symbol Detection Strategy (minimizes the symbol error rate)

$$\widehat{a}_k = \underset{\widetilde{a}_k}{\operatorname{argmax}}\, P(\widetilde{a}_k|\mathbf{r}) \tag{2}$$

Detection Theory (2/3)

Let us consider the case of the AWGN channel (channels with unknown parameters will be considered later):

$$r(t) = s(t, \mathbf{a}) + w(t) \, .$$

Assume that we transmit a sequence of K M-ary symbols. The possible transmitted signals are M^K and an orthonormal basis of this subspace will have $N \leq M^K$ elements. The components of $r(t)$ on this basis will be

$$r_i = s_i(\mathbf{a}) + w_i \quad , \quad i = 1, 2, \ldots, N \, .$$

We are perfectly representing the signals but not the noise. It can be shown that the noise part that cannot be represented on this orthonormal basis is irrelevant for detection. In addition, it can be shown that w_i are i.u.d. complex Gaussian random variables with mean zero and variance $2N_0$ (N_0 per component).

Detection Theory (3/3)

Collecting the components r_i, $s_i(\mathbf{a})$, w_i into vectors of N elements, we have the following vector model:

$$\mathbf{r} = \mathbf{s}(\mathbf{a}) + \mathbf{w}\,.$$

$\mathbf{r}$ is a sufficient statistic for detection and

$$p(\mathbf{r}|\widetilde{\mathbf{a}}) = \frac{1}{2\pi N_0} \exp\left\{-\frac{1}{2N_0}||\mathbf{r}-\mathbf{s}(\widetilde{\mathbf{a}})||^2\right\}\,. \tag{3}$$

Let us consider for example MAP sequence detection

$$\begin{aligned}
\widehat{\mathbf{a}} &= \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\, P(\widetilde{\mathbf{a}}|\mathbf{r}) = \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\, \frac{p(\mathbf{r}|\widetilde{\mathbf{a}})P(\widetilde{\mathbf{a}})}{p(\mathbf{r})} = \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\, \ln\left[p(\mathbf{r}|\widetilde{\mathbf{a}})P(\widetilde{\mathbf{a}})\right] \\
&= \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\left\{-\frac{1}{2N_0}||\mathbf{r}-\mathbf{s}(\widetilde{\mathbf{a}})||^2 + \ln P(\widetilde{\mathbf{a}})\right\} = \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\left\{-||\mathbf{r}-\mathbf{s}(\widetilde{\mathbf{a}})||^2 + 2N_0 \ln P(\widetilde{\mathbf{a}})\right\} \\
&= \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\left\{\mathcal{R}[\mathbf{r}^T\mathbf{s}(\widetilde{\mathbf{a}})^*] - \frac{1}{2}||\mathbf{s}(\widetilde{\mathbf{a}})||^2 + N_0 \ln P(\widetilde{\mathbf{a}})\right\} \\
&= \underset{\widetilde{\mathbf{a}}}{\text{argmax}}\left\{\mathcal{R}\left[\int_{-\infty}^{\infty} r(t)s^*(t,\widetilde{\mathbf{a}})\,\mathrm{d}t\right] - \frac{1}{2}\int_{-\infty}^{\infty}|s(t,\widetilde{\mathbf{a}})|^2\,\mathrm{d}t + N_0 \ln P(\widetilde{\mathbf{a}})\right\}
\end{aligned} \tag{4}$$

In the following, we will avoid the use of $\widetilde{\ }$ to denote the possible sequences and symbols, and we will also consider $P(\widetilde{\mathbf{a}})$ =const. $\rightarrow$ can be discarded.

Estimation Theory

Let us suppose that we would like to estimate an unknown parameter θ (the extension to multiple parameters is trivial) from noisy observations $\mathbf{r}$.

- We can model parameter θ as deterministic unknown $\rightarrow$ maximum likelihood (ML) estimation:

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, p(\mathbf{r}|\theta)$$

- Bayesian estimation can help including some a priori knowledge ($p(\theta)$) about the parameter:

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, p(\theta|\mathbf{r}) = \underset{\theta}{\text{argmax}}\, \frac{p(\mathbf{r}|\theta)p(\theta)}{p(\mathbf{r})} = \underset{\theta}{\text{argmax}}\, p(\mathbf{r}|\theta)p(\theta) \quad \text{MAP estimate}$$

$$\hat{\theta} = E\{\theta|\mathbf{r}\} \quad \text{MMSE estimate}$$

Outline

Linear Modulations (1/14)

The complex envelope of the transmitted signal is

$$s(t) = \sum_k c_k p(t - kT) \,. \tag{5}$$

Symbols c_k, possibly coded, belong to a complex M-ary constellation. T is the symbol time. $p(t)$ is called shaping pulse and is typically real. Hence,

$$s(t) = \sum_k \mathcal{R}[c_k] p(t - kT) + \jmath \sum_k \mathcal{I}[c_k] p(t - kT) \,.$$

If this signal passes through a linear channel with complex envelope of the impulse response $h(t)$, the complex envelope of the noiseless received signal becomes

$$s_R(t) = \sum_k c_k p_R(t - kT)$$

where $p_R(t) = p(t) \otimes \frac{1}{2} h(t)$. This time $p_R(t)$ can be complex because of $h(t)$. In particular, it can be easily shown that $h(t)$ is complex when the channel has a frequency response which has no Hermitian symmetry around f_0.

Linear Modulations (2/14)

Power Spectral Density:

$$W_s(f) = \frac{W_c(f)}{T}|P(f)|^2$$

where $P(f)$ is the Fourier transform of $p(t)$ and

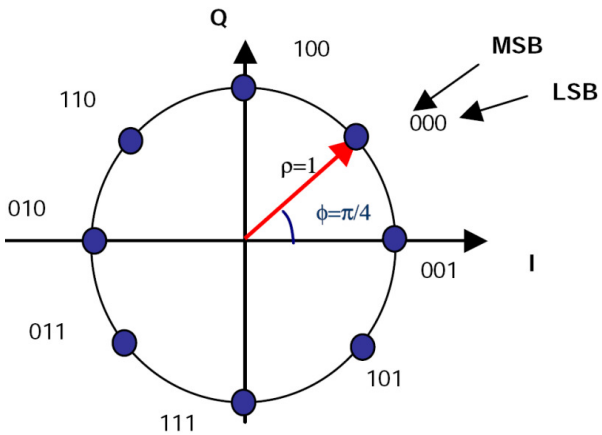
$$W_c(f) = \sum_{m=-\infty}^{+\infty} R_c(m)e^{-\jmath 2\pi fT}$$

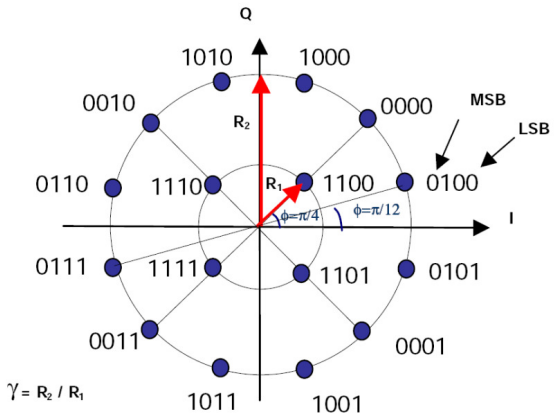
with

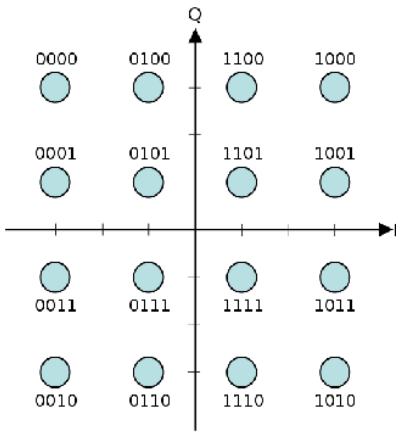
$$R_c(m) = E\{c_{k+m}c_k^*\}\,.$$

Linear Modulations (3/14)

Transmitter: trivial.
Employed constellations: PSK, APSK (QAM and many more but not for satellite communications)







APSK are used to reduce the distortions when we have a nonlinear channel (they have a better constrained capacity than QAM). We will see later why.

Gray mapping is usually employed (when possible) unless proper mapping may be more convenient in the presence of coding (see later).

Linear Modulations (4/14)

Let us consider MAP sequence detection and assume an AWGN channel. By plugging (5) into (4), we have that the first integral is (assuming that K is also the number of transmitted code symbols)

$$\int_{-\infty}^{\infty} r(t) \sum_{k=0}^{K-1} c_k^* p^*(t-kT)\,\mathrm{d}t = \sum_{k=0}^{K-1} c_k^* \underbrace{\int_{-\infty}^{\infty} r(t) p^*(t-kT)\,\mathrm{d}t}_{x_k} = \sum_{k=0}^{K-1} x_k c_k^*$$

having defined

$$x_k = \int_{-\infty}^{+\infty} r(t) p^*(t-kT)\,\mathrm{d}t = r(t) \otimes p^*(-t)|_{t=kT}$$ MF output at time $t = kT$.

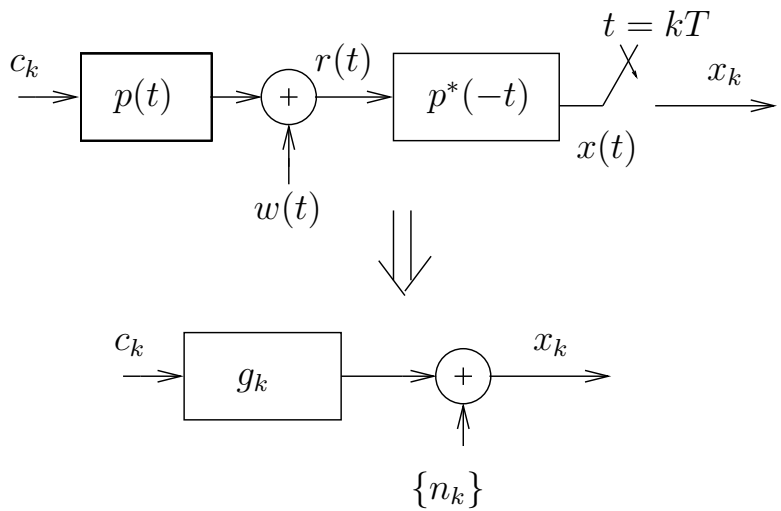
For the second integral of (4) we have

$$\int_{-\infty}^{+\infty} |s(t,\mathbf{a})|^2\,\mathrm{d}t = \int_{-\infty}^{+\infty} \sum_{k=0}^{K-1} c_k p(t-kT) \sum_{m=0}^{K-1} c_m^* p^*(t-mT)\,\mathrm{d}t$$

$$= \sum_{k=0}^{K-1}\sum_{m=0}^{K-1} c_k c_m^* \int_{-\infty}^{+\infty} p(t-kT)p^*(t-mT)\,\mathrm{d}t = \sum_{k=0}^{K-1}\sum_{m=0}^{K-1} c_k c_m^* \underbrace{p(t) \otimes p^*(-t)|_{t=(m-k)T}}_{g_{m-k}}$$

Linear Modulations (5/14)

$g(t) = p(t) \otimes p^*(-t)$ is the equivalent channel impulse response after the MF (and has Hermitian symmetry):



Assuming that $g_k \neq 0$ for $k = -L, -L+1, \ldots, 0, \ldots, L-1, L$, only, we may express

$$x_k = \sum_{i=0}^{K-1} c_i g_{k-i} + n_k = \sum_{\ell=-L}^{L} g_\ell c_{k-\ell} + n_k$$

where samples n_k have autocorrelation function $R_n(m) = E\{n_{k+m} n_k^*\} = 2N_0 g_m$ (so noise is not white).

Linear Modulations (6/14)

We said that

$$\int_{-\infty}^{+\infty} |s(t,\mathbf{a})|^2 \, \mathrm{d}t = \sum_{k=0}^{K-1} \sum_{m=0}^{K-1} c_k c_m^* g_{m-k} \, .$$

This is the sum of all the elements of an Hermitian matrix and can also be expressed as

$$\int_{-\infty}^{+\infty} |s(t,\mathbf{a})|^2 \, \mathrm{d}t = \sum_{k=0}^{K-1} |c_k|^2 g_0 + 2\mathcal{R}\left[\sum_{k=0}^{K-1} \sum_{\ell=1}^{L} c_k^* c_{k-\ell} g_\ell\right].$$

The strategy thus becomes

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} \underbrace{\sum_{k=0}^{K-1} \left\{ \underbrace{\mathcal{R}\left[x_k c_k^* - \frac{1}{2}|c_k|^2 g_0 - \sum_{\ell=1}^{L} c_k^* c_{k-\ell} g_\ell\right]}_{\lambda_k(c_k, c_{k-1}, \ldots, c_{k-L})} \right\}}_{\Lambda(\mathbf{a})} . \tag{6}$$

Linear Modulations (7/14)

Let us assume that the encoder can be described through a FSM with equations

$$c_k = o(a_k, \mu_k) \tag{7}$$

$$\mu_{k+1} = ns(a_k, \mu_k) \tag{8}$$

where μ_k is the encoder's state, belonging to an alphabet of cardinality S_c. This model describes a TCM encoder. We will see later the case of convolutional encoders. In the case of a TCM encoder,

$$\lambda_k(c_k, c_{k-1}, \ldots, c_{k-L}) = \lambda_k(a_k, a_{k-1}, \ldots, a_{k-L}, \mu_{k-L}) = \lambda_k(a_k, \sigma_k)$$

where we have defined

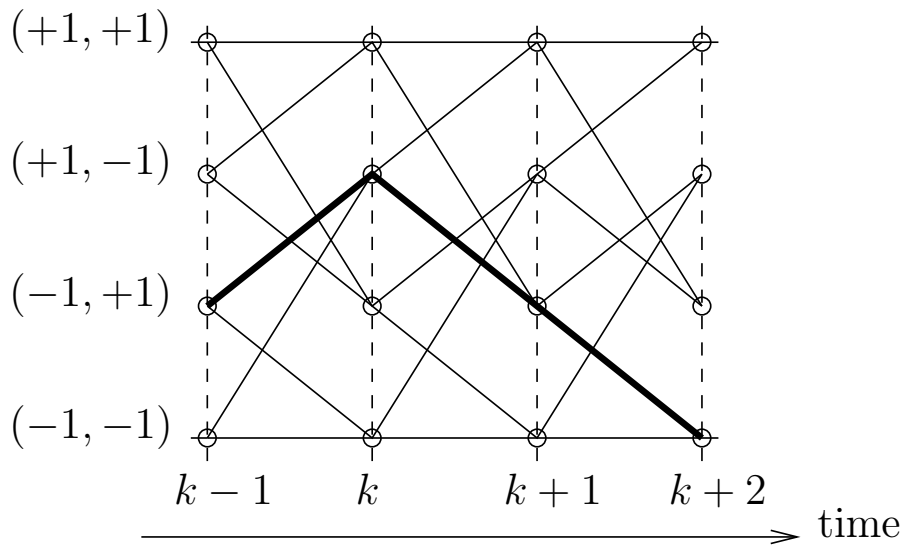
$$\sigma_k = (a_{k-1}, \ldots, a_{k-L}, \mu_{k-L})$$

as the state of the system (taking into account both the ISI and the encoder). The total number of states is $S = S_c M^L$. The temporal evolution of the states can be represented through a trellis diagram.

Linear Modulations (8/14)

Example: uncoded transmission with $L = 2$ and $M = 2$:

$$\sigma_k = (a_{k-1}, a_{k-2})$$



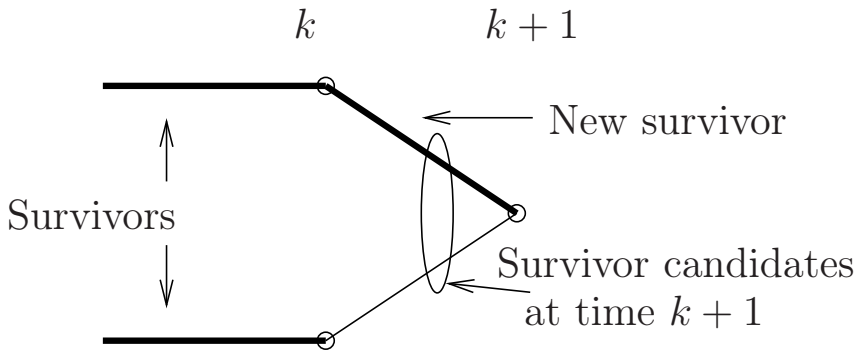
Sequence $\mathbf{a}$ is in a one-to-one correspondence with a path on the trellis, whereas (a_k, σ_k) uniquely identifies a trellis branch:

$$\lambda(a_k, \sigma_k) \rightarrow \text{branch metric}$$

$$\Lambda(\mathbf{a}) \rightarrow \text{path metric}$$

Linear Modulations (9/14)

The detection strategy reduces to the search of the path with the largest (lowest, if we change the sign) metric. This can be done through the Viterbi algorithm (survivors and their extension through Add-Compare-Select)



Going back, the survivors merge $\rightarrow$ decision can be taken with a delay (the VA decision delay).

Linear Modulations (10/14)

Condition for absence of ISI:

$$g_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} .$$

Samples at the output of the MF:

$$x_k = c_k + n_k$$

and now the noise is white. Moreover, the receiver trellis coincides with the code trellis, i.e.,

$$\sigma_k = \mu_k \qquad S = S_c$$

and the branch metric becomes

$$\lambda_k(a_k, \mu_k) = \mathcal{R}\{x_k c_k^*\} - \frac{1}{2}|c_k|^2 .$$

By adding a term which is irrelevant for detection and by changing the sign, we get an equivalent branch metric whose sum has to be minimized (it is the Euclidean distance between $\mathbf{x}$ and $\mathbf{c}$):

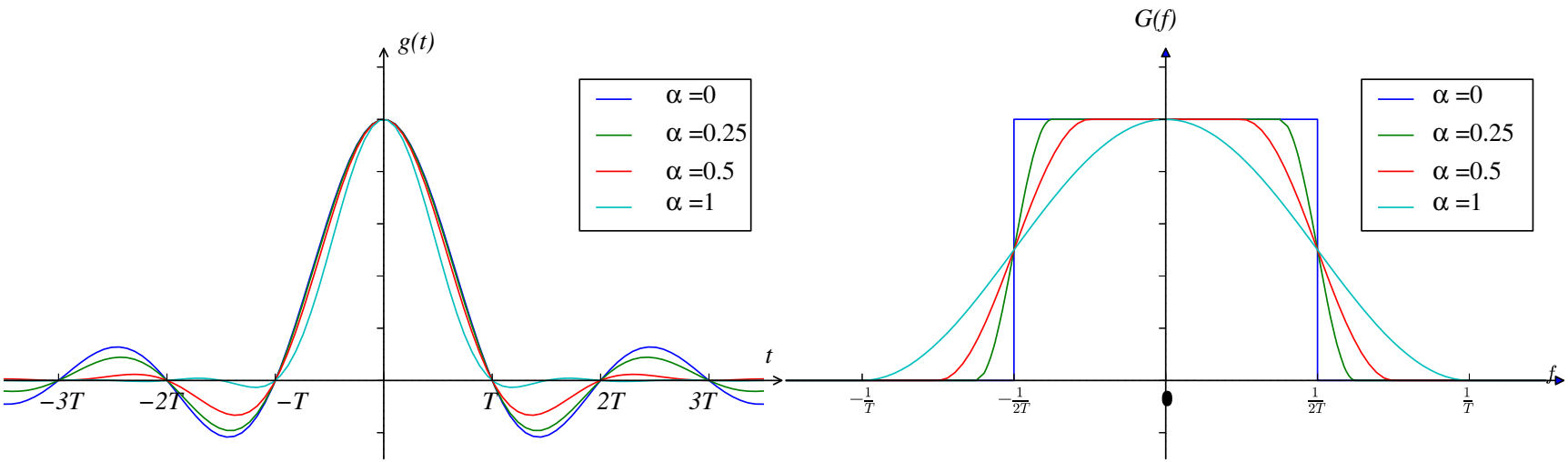
$$\lambda'_k(a_k, \mu_k) = |x_k - c_k|^2 .$$

Linear Modulations (11/14)

In absence of coding, the receiver is based on a symbol-by-symbol detection strategy:

$$\widehat{a}_k = \underset{a_k}{\operatorname{argmin}} \left[|x_k - a_k|^2 - 2N_0 \ln P(a_k) \right].$$

Absence of ISI is ensured when $g(t)$ has a raised cosine spectrum:



Bandwidth $(1+\alpha)/2T$. Roll-off $\alpha = 0.2, 0.25, 0.35$ in DVB-S2. The corresponding $p(t)$ will have root raised cosine (RRC) spectrum.

Linear Modulations (12/14)

When running simulations, performance is often expressed as a function of E_b/N_0, where E_b is the mean energy per information bit. Let us start from the continuous-time model:

$$r(t) = \sum_{\ell} c_\ell p(t - \ell T) + w(t) \, .$$

Unless we have an ideal channel with no synchronization errors, we need an oversampled model to simulate the received signal $\rightarrow$ we have to imagine that an anti-aliasing filter (AAF) is present, otherwise the noise samples will have an infinite power. A sampling interval of T_c is sufficient to represent signals of bandwidth equal to $1/2T_c \rightarrow$ the AAF will have bandwidth $1/2T_c \rightarrow$ complex noise samples will have power $2N_0 \cdot 2B = 2N_0/T_c \rightarrow$ the variance of each component will be $\sigma^2 = N_0/T_c$ and the discrete time model is

$$r(kT_c) = \sum_{\ell} c_\ell p(kT_c - \ell T) + n(kT_c)$$

where $n(t)$ is $w(t)$ filtered by the AAF. How to relate σ^2 to E_b/N_0?

Linear Modulations (13/14)

Assuming $T/T_c = \eta$ (oversampling factor)

$$r(kT_c) = \sum_{\ell} c_\ell p(kT_c - \ell\eta T_c) + n(kT_c)\,.$$

The power of the complex envelope of the received useful signal $\sum_\ell c_\ell p(t-\ell T)$ is

$$P_s = \int_{-\infty}^{\infty} W_s(f)\,\mathrm{d}f = \int_{-\infty}^{\infty} \frac{W_c(f)}{T}\,|P(f)|^2\,\mathrm{d}f\,.$$

Hyp: i.u.d. symbols belonging to an M-ary alphabet of power $E\{|c_k|^2\} = C_2 \rightarrow$

$$R_c(m) = C_2\delta_m \quad \rightarrow \quad W_c(f) = C_2$$

and

$$P_s = \frac{C_2}{T}E_p$$

where E_p is the energy of $p(t)$.

Linear Modulations (14/14)

$$E_s = \frac{P_s T}{2} = \frac{C_2}{2} E_p = E_b N_b$$

where E_s is the received energy per coded symbol (remember that the power of the passband signal is half that of its complex envelope) and N_b is the number of bits per coded symbols (it takes into account the code rate and the cardinality of the alphabet). Hence,

$$\sigma^2 = \frac{N_0}{T_c} = \frac{N_0 \eta}{T} = \frac{N_0 \eta}{T} \frac{C_2 E_p}{2 E_b N_b} = \frac{N_0}{E_b} \frac{\eta C_2 E_p}{2 T N_b}$$

Continuous Phase Modulations (1/8)

Constant envelope and continuous phase $\rightarrow$ insensitive to NL amplifiers and smooth spectrum. Drawback: introduction of memory.
Complex envelope:

$$s(t, \mathbf{a}) = \sqrt{\frac{2E_s}{T}}\, e^{\jmath\left[2\pi h \sum_i a_i q(t-iT)\right]}$$

h is the modulation index, $a_i \in \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$ are the information symbols, $q(t)$ is the phase smoothing response such that

$$\begin{cases} q(t) = 0 & \text{for} \quad t < 0 \\ q(t) = \dfrac{1}{2} & \text{for} \quad t > LT \end{cases}$$

where L is the CPM correlation length

$$\begin{cases} L = 1 & \text{full response CPM} \\ L > 1 & \text{partial response CPM.} \end{cases}$$

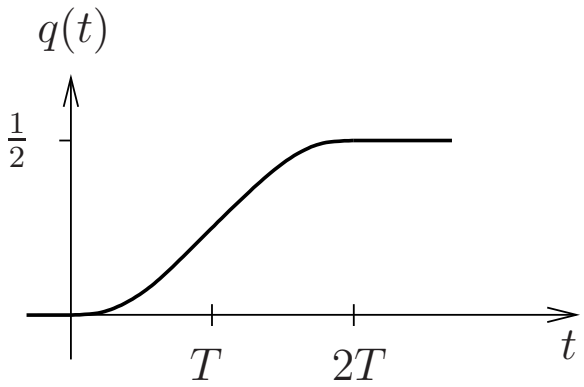
Continuous Phase Modulations (2/8)

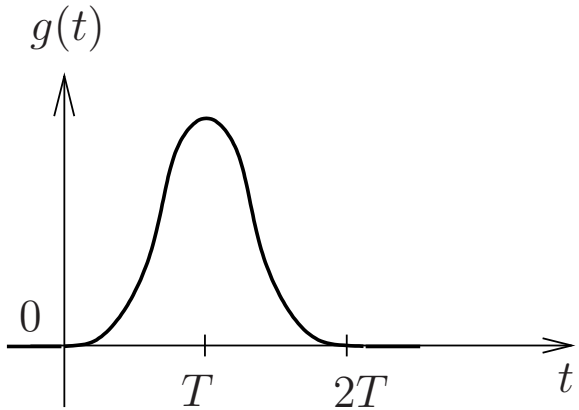
Frequency pulse:

$$g(t) = \frac{\mathrm{d}q(t)}{\mathrm{d}t} .$$

Hence,

$$g(t) = 0 \quad \text{for} \quad t < 0 \quad \text{or} \quad t > LT \quad \text{and} \int_0^{LT} g(t)\,\mathrm{d}t = \frac{1}{2} .$$

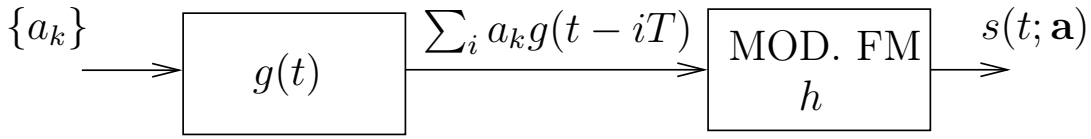




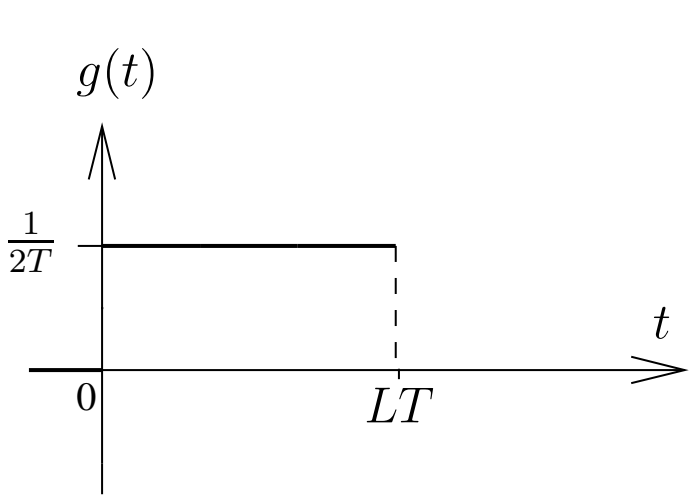
Continuous Phase Modulations (3/8)

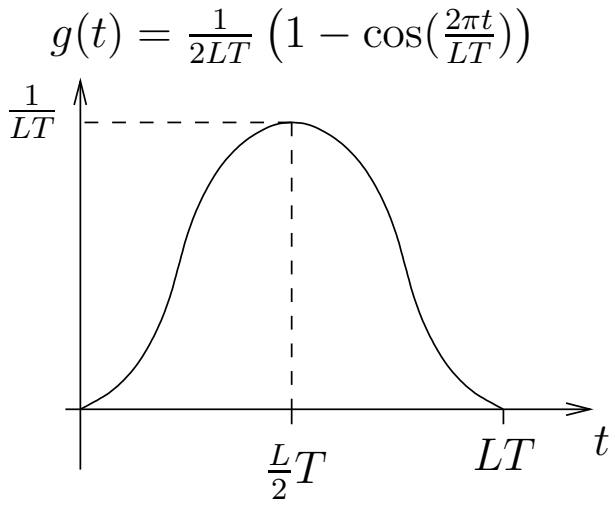
Hence,

$$s(t,\mathbf{a}) = \sqrt{\frac{2E_s}{T}}\ \exp\left\{\jmath\left[2\pi h\int_{-\infty}^{t}\sum_i a_i g(\tau - iT)\,\mathrm{d}\tau + \theta\right]\right\}$$



Usually employed frequency pulses: L-REC, L-RC, Gaussian, mixed RC-REC.





Continuous Phase Modulations (4/8)

Notable CPM formats:

- CP-FSK: 1-REC
- MSK: 1-REC, $h = 1/2$, $M = 2$
- GMSK: similar to MSK but $g(t)$ is obtained by filtering a 1-REC pulse with a Gaussian filter (used in GSM)

MAP sequence detection: from (4), we obtain

$$\widehat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, \mathcal{R}\left[\int_{-\infty}^{\infty} r(t)s^*(t,\mathbf{a})\,\mathrm{d}t\right]. \tag{9}$$

Before further modifications, we need an alternative model for the CPM signal. Let us consider the argument $\phi(t,\mathbf{a})$ of the complex exponential in the generic symbol interval $[kT,(k+1)T)$:

$$\phi(t,\mathbf{a}) = \underbrace{2\pi h \sum_{i=0}^{k-L} a_i \frac{1}{2}}_{\varphi_k} + 2\pi h \sum_{i=k-L+1}^{k-1} a_i q(t-iT) + 2\pi h a_k q(t-kT) \qquad kT \le t < (k+1)T$$

Continuous Phase Modulations (5/8)

Phase state:

$$\varphi_k = \pi h \sum_{i=0}^{k-L} a_i \mod 2\pi$$

$$\varphi_{k+1} = \varphi_k + \pi h a_{k-L+1} \mod 2\pi \ .$$

How many phase states? Infinite in principle. When $h = n/p$ (n and p relatively prime), it can be shown that φ_k takes on p values:

$$\varphi_k \in \left\{ 0, 2\pi\frac{n}{2p}, 2\pi\frac{2n}{2p}, \ldots, 2\pi\frac{(p-1)n}{2p} \right\} \quad n \text{ even}$$

$$\varphi_k \in \left\{ 0, 2\pi\frac{n}{2p}, 2\pi\frac{2n}{2p}, \ldots, 2\pi\frac{(2p-1)n}{2p} \right\} \quad n \text{ odd}$$

(in this second case, half states in even symbol instants and half states in odd symbol instants).

Continuous Phase Modulations (6/8)

We thus have a chip

$$s(t-kT, a_k, \sigma_k) = \sqrt{\frac{2E_s}{T}}\ \exp\{\jmath\phi(t, a_k, \sigma_k)\}$$

of signal in $[kT, (k+1)T)$ which depends on (a_k, σ_k), with

$$\sigma_k = (\underbrace{a_{k-1}, a_{k-2}, \ldots, a_{k-L+1}}_{\text{correlative state}}; \underbrace{\varphi_k}_{\text{phase state}})$$

and

$$s(t, \mathbf{a}) = \sum_k s(t-kT, a_k, \sigma_k)\,.$$

Substituting this expression into (9), we have

$$\widehat{\mathbf{a}} = \underset{\mathbf{a}}{\text{argmax}} \sum_k \mathcal{R}\left[\int_{-\infty}^{\infty} r(t) s^*(t-kT, a_k, \sigma_k)\,\mathrm{d}t\right] = \underset{\mathbf{a}}{\text{argmax}} \sum_k \underbrace{\mathcal{R}\left[z_k(a_k, \sigma_k)\right]}_{\lambda_k(a_k, \sigma_k)}$$

with $z_k(a_k, \sigma_k) = \int_{-\infty}^{\infty} r(t) s^*(t-kT, a_k, \sigma_k)\,\mathrm{d}t \rightarrow$ bank of matched filters and Viterbi algorithm. MS matched filters with $S = pM^{L-1}$.

Continuous Phase Modulations (7/8)

Laurent Decomposition (Laurent 1986 - Mengali& Morelli 1995). A CPM signal can be expressed as the superposition of linearly modulated components:

$$s(t,\mathbf{a}) = \sum_{m=0}^{F-1} \sum_{k} \alpha_{m,k} p_m(t - kT) \tag{10}$$

where $F = (M-1)2^{(L-1)\log_2 M}$, $\{p_m(t)\}$ are the shaping pulses and $\{\alpha_{m,k}\}$ are the pseudo-symbols.
$\{p_m(t)\}$ can be obtained in closed form from $q(t)$ and h.
$\{\alpha_{m,k}\}$ can be expressed in closed form from $\{a_k\}$ and h. Example:

$$\alpha_{0,k} = \exp\left\{\jmath\pi h \sum_{i=0}^{k} a_i\right\} = \alpha_{0,k-1} \exp\{\jmath\pi h a_k\} \ .$$

The argument of the exponential has the same expression of the phase state $\rightarrow p$ values. In addition, there is an intrinsic differential encoding.

Continuous Phase Modulations (8/8)

Most of the signal power is concentrated in the first $M-1$ linearly modulated signals (the principal components), i.e., the approximation

$$\tilde{s}(t;\mathbf{a}) \simeq \sum_{m=0}^{M-2} \sum_{k} \alpha_{m,k} p_m(t-kT)$$

is very accurate $\rightarrow$ the receiver can be designed considering only the principal components.
Plugging (10) into (9), we obtain

$$\lambda(a_k, \alpha_{0,k-1}) = \mathcal{R}\left[\sum_{m=0}^{M-2} x_{m,k}\alpha_{m,k}^*\right]$$

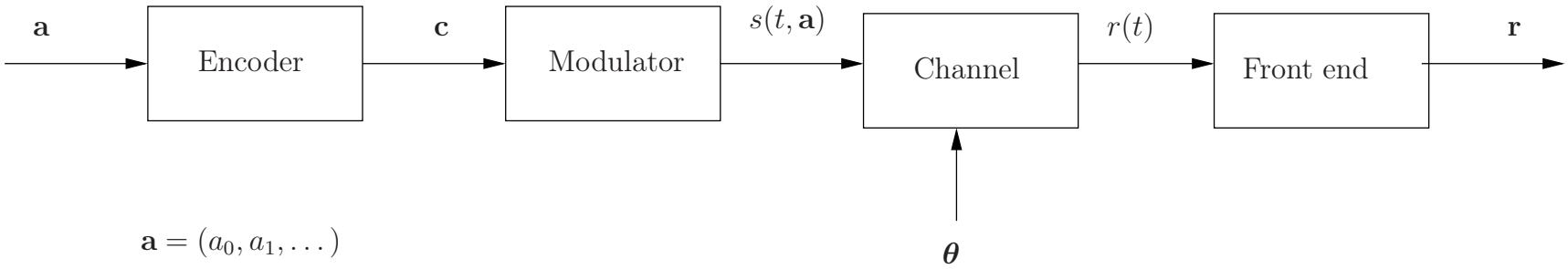
where

$$x_{m,k} = r(t) \otimes p_m(-t)|_{t=kT}$$

$\rightarrow (M-1)$ MFs are required. It can be verified that $\{\alpha_{m,k}\}_{m=0}^{M-2}$ can be expressed as a function of a_k and $\alpha_{0,k-1} \rightarrow$ Viterbi algorithm with p states.

Outline

Problem Statement (1/3)



$\mathbf{a} = (a_0, a_1, \dots)$
$\mathbf{c} = (c_0, c_1, \dots)$
$\boldsymbol{\theta} = (\theta_0, \theta_1, \dots)$
$\mathbf{r} = (r_0, r_1, \dots)$

Problems of interest:

- MAP Symbol Detection:

$$\widehat{a}_k = \underset{a_k}{\operatorname{argmax}}\, P(a_k|\mathbf{r})$$

- Bayesian Estimation: $p(\theta_k|\mathbf{r})$ is required. Examples:

$\widehat{\theta}_k = \underset{\theta_k}{\operatorname{argmax}}\, p(\theta_k|\mathbf{r})$ (MAP Estimation) $\quad \widehat{\theta}_k = E\{\theta_k|\mathbf{r}\}$ (MMSE Estimation)

Problem Statement (2/3)

- All problems have, in general, exponential complexity in the transmission length (NP-hard), unless proper conditions occur (finite memory channel).
- They have in common a marginalization of the joint distribution $P(\mathbf{a}, \mathbf{c}, \boldsymbol{\theta}|\mathbf{r})$.
- When is it possible to implement a marginalization in a more effective way?

Problem Statement (3/3)

- Algorithms that must deal with complicated global functions of many variables often exploit the manner in which the given functions factor as a product of “local" functions, each of which depends on a subset of the variables.
- Such a factorization can be visualized with a factor graph, a bipartite graph that expresses which variables are arguments of which local functions.
- The sum-product algorithm works on the factor graph and computes—either exactly or approximately—the marginal functions derived from the global function.
- A wide variety of algorithms developed in artificial intelligence, signal processing, and digital communications can be derived as specific instances of the sum-product algorithm.

Definitions

- Let $x_1, x_2, \ldots, x_n$ be a collection of variables.
- x_i takes on values in some (usually finite) domain (or alphabet) A_i.
- Let $g(x_1, x_2, \ldots, x_n)$ be an $\mathbb{R}$-valued function of these variables, i.e., a function with domain $S = A_1 \times A_2 \times \cdots \times A_n$ and codomain $\mathbb{R}$.
- Associated with $g(x_1, x_2, \ldots, x_n)$ there are n marginal functions $g_i(x_i)$ defined as

$$g_i(x_i) = \sum_{x_1 \in A_1} \cdots \sum_{x_{i-1} \in A_{i-1}} \sum_{x_{i+1} \in A_{i+1}} \cdots \sum_{x_n \in A_n} g(x_1, x_2, \ldots, x_n)$$

- This operation is called not-sum or summary and will be denoted by

$$g_i(x_i) = \sum_{\sim\{x_i\}} g(x_1, x_2, \ldots, x_n)$$

Factor Graphs (FGs)

- Suppose that $g(x_1, x_2, \ldots, x_n)$ factors into a product of several local functions, each having a subset of $\{x_1, x_2, \ldots, x_n\}$ as argument:

$$g(x_1, x_2, \ldots, x_n) = \prod_{j \in J} f_j(X_j)$$

  where J is a discrete index set, X_j is a subset of $\{x_1, x_2, \ldots, x_n\}$, and $f_j(X_j)$ is a function having the elements of X_j as arguments.

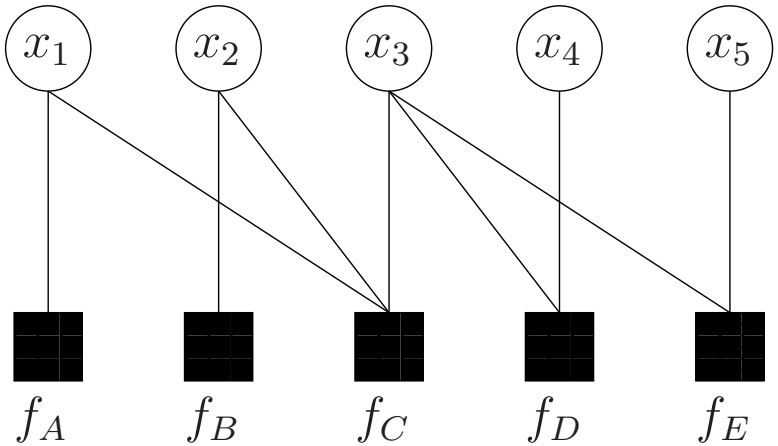
- A factor graph is a bipartite graph that expresses the structure of this factorization.
- It has a variable node for each variable x_i, a factor node for each local function f_j, and an edge connecting variable node x_i to function node f_j if and only if x_i is an argument of f_j.

Example

- Let $g(x_1, x_2, \ldots, x_n)$ be a function of five variables, and suppose that g can be expressed as a product

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5) .$$

- In this case, $J = \{A, B, C, D, E\}$, $X_A = \{x_1\}$, $X_B = \{x_2\}$, $X_C = \{x_1, x_2, x_3\}$, $X_D = \{x_3, x_4\}$, and $X_E = \{x_3, x_5\}$.



Marginalization of a Joint PMF

- If $g(x_1, x_2, \ldots, x_n)$ is a joint probability mass function (PMF), we are interested in computing the marginal functions $g_i(x_i)$.
- In the case of the previous example, by using the distributive law:

$$g_1(x_1) = f_A(x_1)\Big\{\sum_{x_2} f_B(x_2)\Big[\sum_{x_3} f_C(x_1,x_2,x_3)\Big(\sum_{x_4} f_D(x_3,x_4)\Big)\Big(\sum_{x_5} f_E(x_3,x_5)\Big)\Big]\Big\}$$

  where $a(b+c)$ is more cost-effective than $ab+ac$.
- In summary notation

$$g_1(x_1) = f_A(x_1)\sum_{\sim\{x_1\}}\Big\{f_B(x_2)f_C(x_1,x_2,x_3)\Big(\sum_{\sim\{x_3\}} f_D(x_3,x_4)\Big)\Big(\sum_{\sim\{x_3\}} f_E(x_3,x_5)\Big)\Big\}.$$

- Similarly

$$g_3(x_3) = \Big[\sum_{\sim\{x_3\}} f_A(x_1)f_B(x_2)f_C(x_1,x_2,x_3)\Big]\Big[\sum_{\sim\{x_3\}} f_D(x_3,x_4)\Big]\Big[\sum_{\sim\{x_3\}} f_E(x_3,x_5)\Big].$$

- We have some operations in common! We can compute all marginals with a single algorithm by exploiting the operations in common.

Sum-Product Algorithm (1/4)

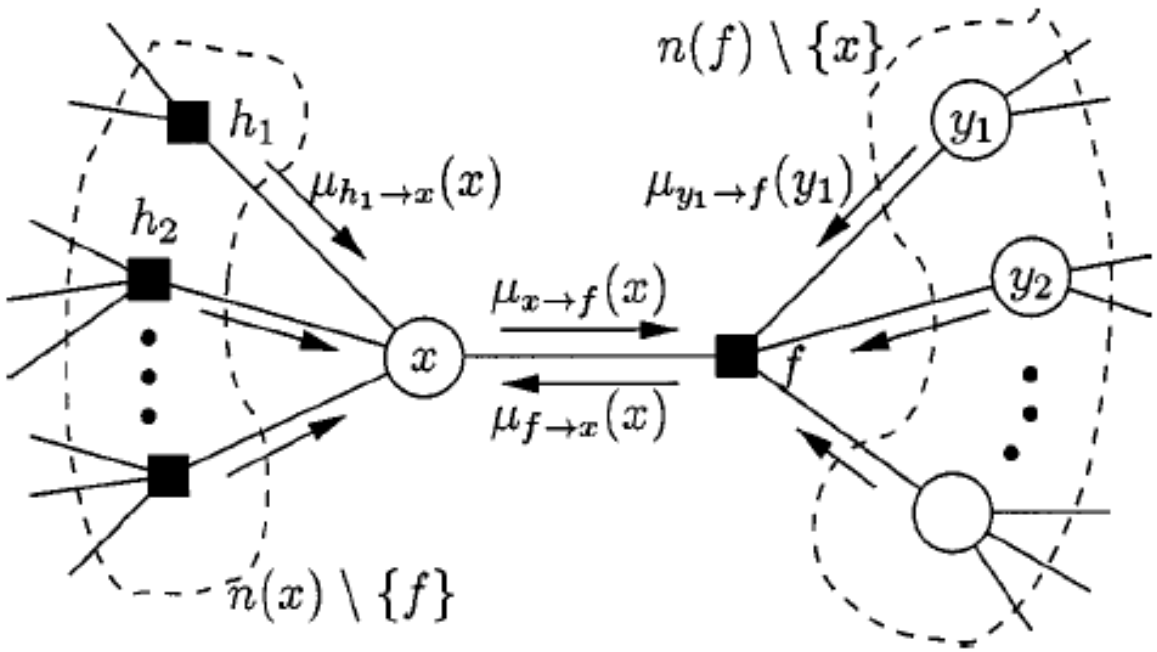
- When a factor graph is cycle free, the factor graph not only encodes in its structure the factorization of the global function, but also encodes the arithmetic expressions by which the marginal functions associated with the global function may be computed.
- Let us imagine that there is a processor associated with each vertex of the factor graph and that the factor-graph edges represent channels by which these processors may communicate. We have to follow these rules:
  - Each variable node in the factor graph computes the product of the messages at its input.
  - Each factor node computes the product of the messages at its input times f and then, when computing the message to be sent to a variable node x, it has to compute the $\sum_{\sim\{x\}}$ summary operator.
- Trivial products (those with one or no operand) act as identity operators.
- A summary operation $\sum_{\sim\{x\}}$ applied to a function with a single argument x is also a trivial operation, and may be omitted.

Sum-Product Algorithm (2/4)

- Message passing is initiated at the leaves.
- Each vertex v remains idle until messages have arrived on all but one of the edges incident on v; then, it is able to compute the message on the remaining edge to another vertex w. After that, the vertex v returns to the idle state, waiting for a return message from w. Once this message has arrived, the vertex is able to compute and send messages to each of its neighbors (other than w).

Sum-Product Algorithm (3/4)

- The algorithm terminates when two messages have been passed over every edge, one in each direction.
- At the variable node x_i, the product of all incoming messages is the marginal function $g_i(x_i)$.



Sum-Product Algorithm (4/4)

- The resulting algorithm is called sum-product algorithm and is based on the following two rules. Let us denote by $n(v)$ the set of neighbors of a given node v.
  - variable to local function

$$\mu_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x)$$

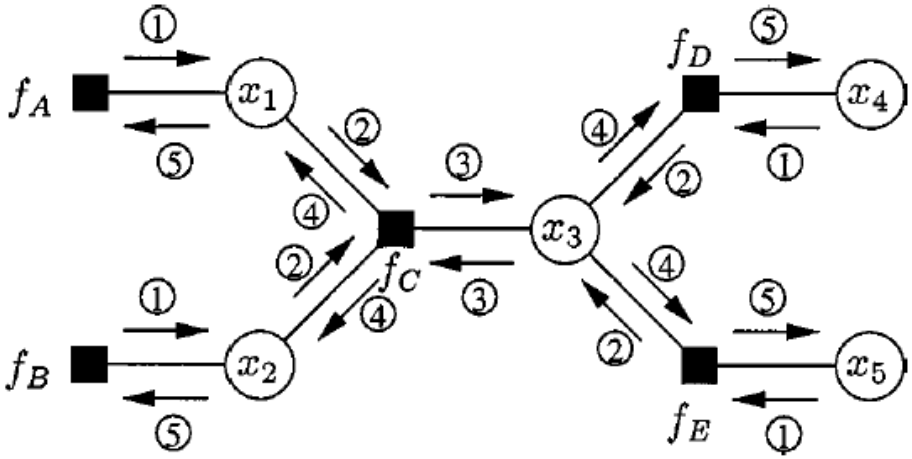
  - local function to variable

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right)$$

    where $X = n(f)$ is the set of arguments of the function f.

- Note that all messages are functions.
- Concept of extrinsic information.

Example (1/3)



- Step 1

$$\mu_{f_A \to x_1}(x_1) = \sum_{\sim\{x_1\}} f_A(x_1) = f_A(x_1) \qquad \mu_{f_B \to x_2}(x_2) = \sum_{\sim\{x_2\}} f_B(x_2) = f_B(x_2)$$

$$\mu_{x_4 \to f_D}(x_4) = 1 \qquad \mu_{x_5 \to f_E}(x_5) = 1$$

- Step 2

$$\mu_{x_1 \to f_C}(x_1) = \mu_{f_A \to x_1}(x_1) \qquad \mu_{x_2 \to f_C}(x_2) = \mu_{f_B \to x_2}(x_2)$$

$$\mu_{f_D \to x_3}(x_3) = \sum_{\sim\{x_3\}} \mu_{x_4 \to f_D}(x_4) f_D(x_3, x_4) \qquad \mu_{f_E \to x_3}(x_3) = \sum_{\sim\{x_3\}} \mu_{x_5 \to f_E}(x_5) f_E(x_3, x_5)$$

Example (2/3)

- Step 3

$$\mu_{f_C \to x_3}(x_3) = \sum_{\sim\{x_3\}} \mu_{x_1 \to f_C}(x_1)\mu_{x_2 \to f_C}(x_2) f_C(x_1, x_2, x_3)$$

$$\mu_{x_3 \to f_C}(x_3) = \mu_{f_D \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$

- Step 4

$$\mu_{f_C \to x_1}(x_1) = \sum_{\sim\{x_1\}} \mu_{x_3 \to f_C}(x_3)\mu_{x_2 \to f_C}(x_2) f_C(x_1, x_2, x_3)$$

$$\mu_{f_C \to x_2}(x_2) = \sum_{\sim\{x_2\}} \mu_{x_3 \to f_C}(x_3)\mu_{x_1 \to f_C}(x_1) f_C(x_1, x_2, x_3)$$

$$\mu_{x_3 \to f_D}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_E \to x_3}(x_3)$$

$$\mu_{x_3 \to f_E}(x_3) = \mu_{f_C \to x_3}(x_3)\mu_{f_D \to x_3}(x_3)$$

- Step 5

$$\mu_{x_1 \to f_A}(x_1) = \mu_{f_C \to x_1}(x_1)$$

$$\mu_{x_2 \to f_B}(x_2) = \mu_{f_C \to x_2}(x_2)$$

$$\mu_{f_D \to x_4}(x_4) = \sum_{\sim\{x_4\}} \mu_{x_3 \to f_D}(x_3) f_D(x_3, x_4)$$

Example (3/3)

- Termination

$$
\begin{aligned}
g_1(x_1) &= \mu_{f_A \to x_1}(x_1)\,\mu_{f_C \to x_1}(x_1) \\
g_2(x_2) &= \mu_{f_B \to x_2}(x_2)\,\mu_{f_C \to x_2}(x_2) \\
g_3(x_3) &= \mu_{f_C \to x_3}(x_3)\,\mu_{f_D \to x_3}(x_3)\,\mu_{f_E \to x_3}(x_3) \\
g_4(x_4) &= \mu_{f_D \to x_4}(x_4) \\
g_5(x_5) &= \mu_{f_E \to x_5}(x_5)
\end{aligned}
$$

- Equivalently

$$
\begin{aligned}
g_3(x_3) &= \mu_{f_C \to x_3}(x_3)\,\mu_{x_3 \to f_C}(x_3) \\
&= \mu_{f_D \to x_3}(x_3)\,\mu_{x_3 \to f_D}(x_3) \\
&= \mu_{f_E \to x_3}(x_3)\,\mu_{x_3 \to f_E}(x_3)
\end{aligned}
$$

Marginalization in Graphs with Cycles

- When a graph has cycles, the sum-product algorithm does not have a natural termination.
- Because of the cycles in the graph, an iterative algorithm with no natural termination will result, with messages passed multiple times on a given edge.
- It is not possible to obtain an exact marginalization of the global function.
- Some of the most exciting applications of the sum-product algorithm (turbo codes, LDPC codes) arise precisely in situations where the underlying factor graph does have cycles.
- Extensive simulation results show that such decoding algorithms can achieve astonishing performance even though the underlying factor graph has cycles.
- The adopted schedule assumes a key role (ex. flooding schedule).

Outline

Block Codes

Let us consider, for the moment, the case of binary codes.

Block Codes (BC): An input block of K information bits is transformed into an output block of N output (coded) bits. $N > K$, in such a way redundancy is introduced. K/N is the rate of the code.

Main idea: the possible 2^K input sequences are univocally associated to 2^K of the possible 2^N sequences at the output (the codebook). Not all output sequences are thus transmitted. In particular, the sequences which are "less similar" are transmitted making the transmission more robust to noise.

Linear block codes are usually employed.

Definitions:

Input Sequence: $\mathbf{a} = (a_0, a_1, \ldots, a_{K-1})$

Coded Sequence: $\mathbf{c} = (c_0, c_1, \ldots, c_{N-1})$

Generator Matrix: a $K \times N$ matrix $\mathbf{G}$ with elements in GF(2) such that

$$\mathbf{c} = \mathbf{aG} \quad \text{in GF(2)}$$

Parity Check Matrix: a $(N-K) \times N$ matrix $\mathbf{H}$ with elements in GF(2) such that

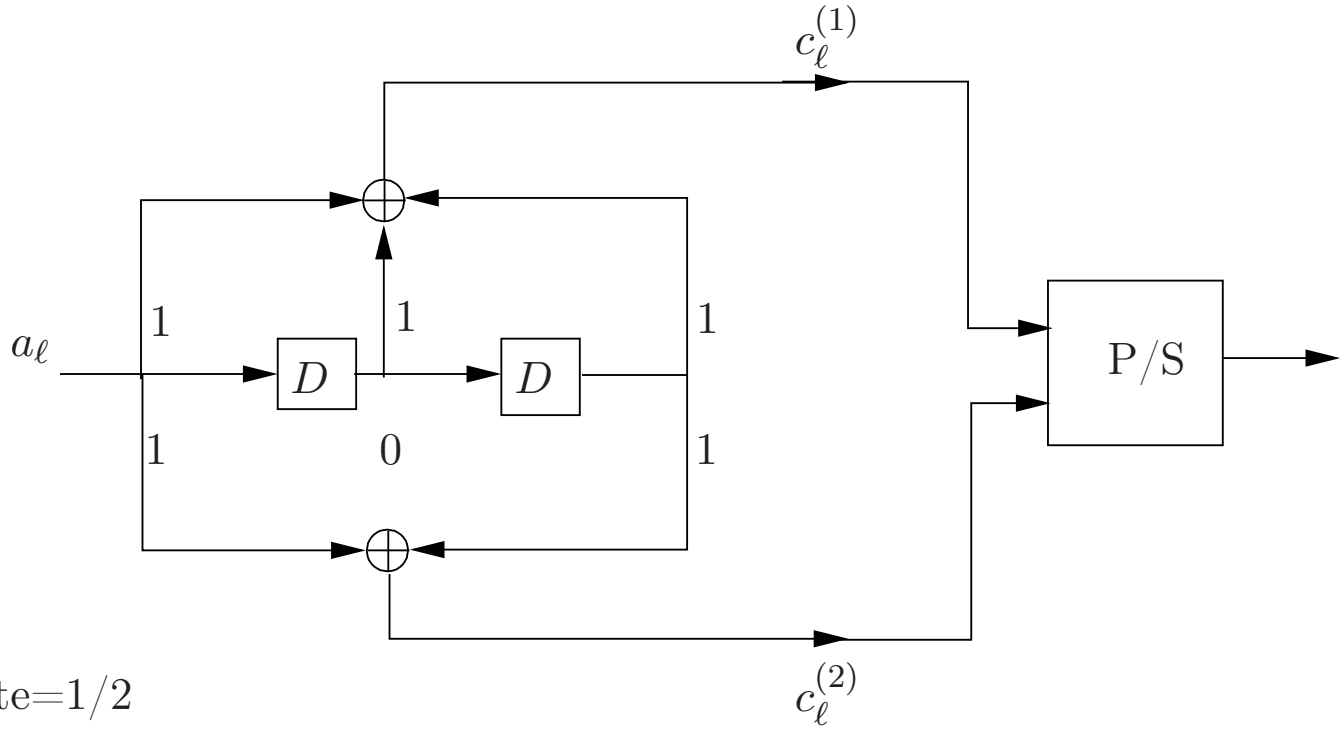
$$\mathbf{Hc}^T = \mathbf{0} \quad \text{in GF(2)}$$

when $\mathbf{c}$ is a codeword.

BC are designed by trying to maximize the minimum Hamming distance.

Convolutional Codes (1/2)

Convolutional Codes (CC): They operate continuously. k bits are received at time ℓ and, at that time instant, n coded bits are produced based on the received input bits and on the input bits received at the previous $\nu - 1$ instants (at least in non-recursive codes). ν is the constraint length. k/n is the rate of the code.



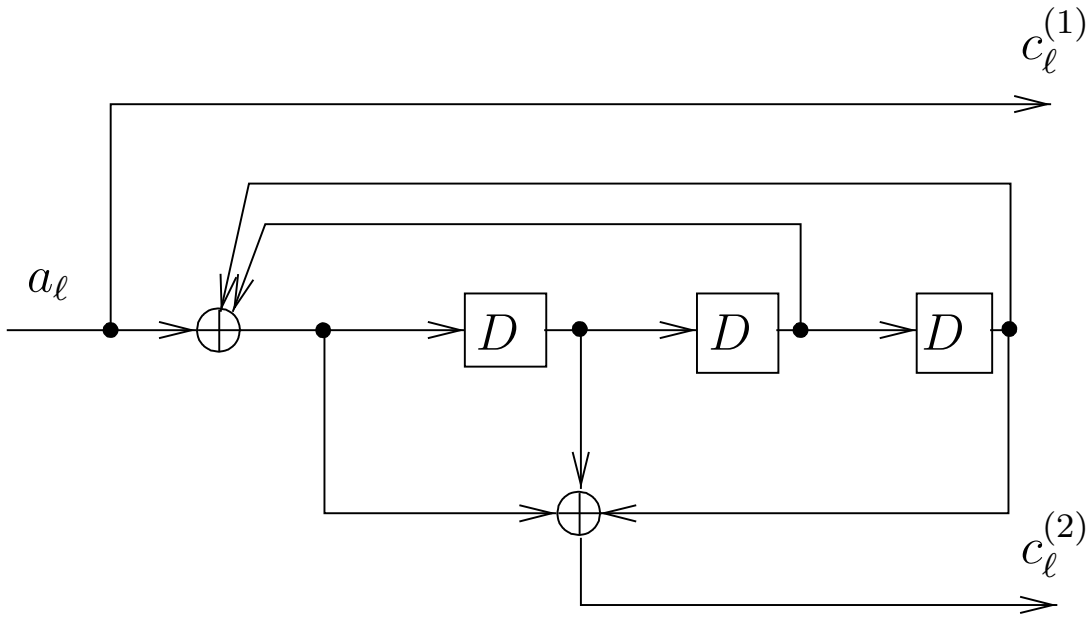
rate=$1/2$

$\nu = 3$

generator polynomial $\{7, 5\}_8 = \{111, 101\}_2$

Convolutional Codes (2/2)

Recursive codes (employed in turbo codes) have feedback connections:



Convolutional codes are designed by trying to maximize the minimum Hamming distance. Generators of the optimal codes can be found on textbooks.

Decoding of Convolutional Codes (1/6)

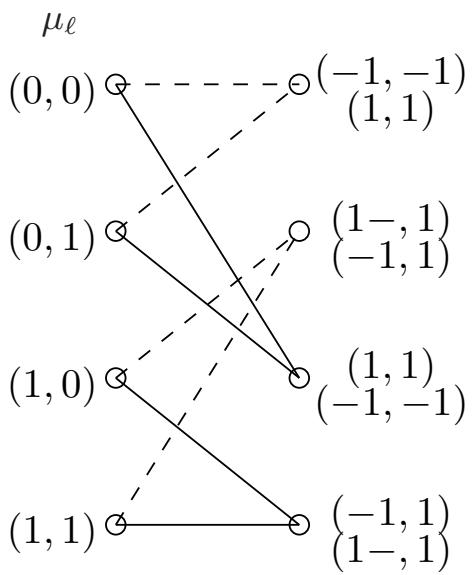
Let us consider, as an example, the CC with generators $(7,5)_8$ and assume that the coded symbols are transmitted through a BPSK:

$$\begin{aligned} 0 &\rightarrow -1 \\ 1 &\rightarrow 1\,. \end{aligned}$$

The code can be described through a FSM with state:

$$\mu_\ell = (a_{\ell-1}, a_{\ell-2})$$

and output symbols $c_\ell^{(1)}$ and $c_\ell^{(2)}$. The code trellis is the following:



Decoding of Convolutional Codes (2/6)

The MAP sequence detection strategy is based on the minimization of the Euclidean distance between the received sequence and the code sequence:

$$\widehat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmin}} \underbrace{\sum_{\ell} \left\{ \underbrace{|x_\ell^{(1)} - c_\ell^{(1)}|^2 + |x_\ell^{(2)} - c_\ell^{(2)}|^2}_{\lambda_\ell(a_\ell, \mu_\ell)} \right\}}_{\Lambda(\mathbf{a})} .$$

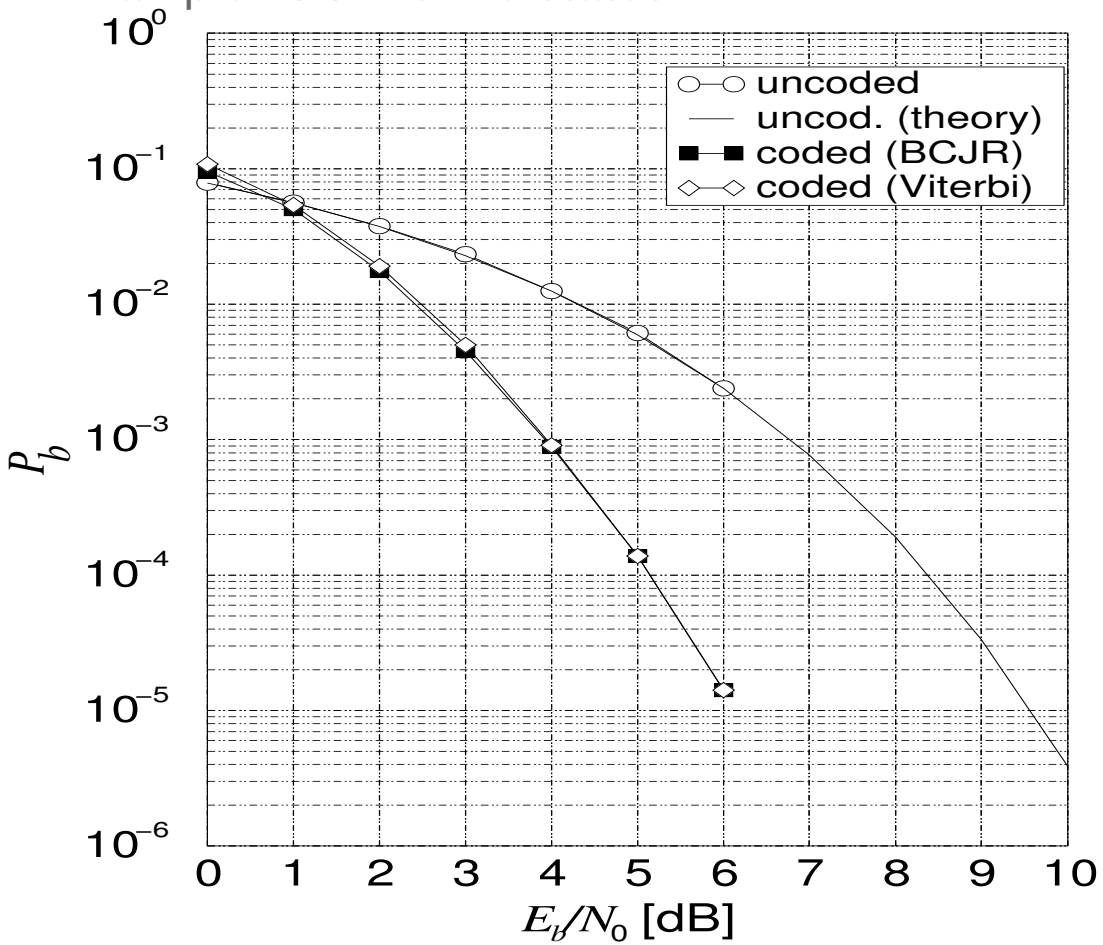
The performance of this strategy depends on the minimum Euclidean distance between two codewords.
Binary codes have been designed by trying to maximize the minimum Hamming distance between codewords. Performance, however, depends on the minimum Euclidean distance $\rightarrow$ no problems when the BPSK is employed since the two distances are proportional (true only for BPSK).
When the binary code is mapped onto an M-ary constellation, coding and mapping must be designed jointly (Ungerboeck paradigm for trellis coded modulations).

Decoding of Convolutional Codes (3/6)

The MAP symbol detection strategy minimizes the symbol error probability:

$$\widehat{a}_n = \underset{a_n}{\operatorname{argmax}}\, P(a_n|\mathbf{r}) \,.$$

What is better? Example: CC with 16 states.



Decoding of Convolutional Codes (4/6)

This strategy produces, as a by-product, the probabilities $\{P(a_n|\mathbf{r})\}$ ($P(a_n = 0|\mathbf{r})$ and $P(a_n = 1|\mathbf{r})$) that are employed in turbo decoding.
We know that the MF output is a sufficient statistics:

$$r_n^{(1)} = x_n^{(1)} = c_n^{(1)} + w_n^{(1)} \qquad r_n^{(2)} = x_n^{(2)} = c_n^{(2)} + w_n^{(2)}$$

$$p(x_n^{(1)}, x_n^{(2)}|c_n^{(1)}, c_n^{(2)}) = p(x_n^{(1)}|c_n^{(1)})p(x_n^{(2)}|c_n^{(2)}) \propto \exp[(-|x_n^{(1)} - c^{(1)}|^2 - |x_n^{(2)} - c_n^{(2)}|^2)/2N_0] .$$

We can write

$$\begin{aligned} P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{a}, \boldsymbol{\mu}|\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) &\propto p(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}|\mathbf{a}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \boldsymbol{\mu})P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \boldsymbol{\mu}|\mathbf{a})P(\mathbf{a}) \\ &= \Big[\prod_{n=1}^{K} p(x_n^{(1)}, x_n^{(2)}|c_n^{(1)}, c_n^{(2)})\Big] P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \boldsymbol{\mu}|\mathbf{a}) \Big[\prod_{n=1}^{K} P(a_n)\Big] \end{aligned}$$

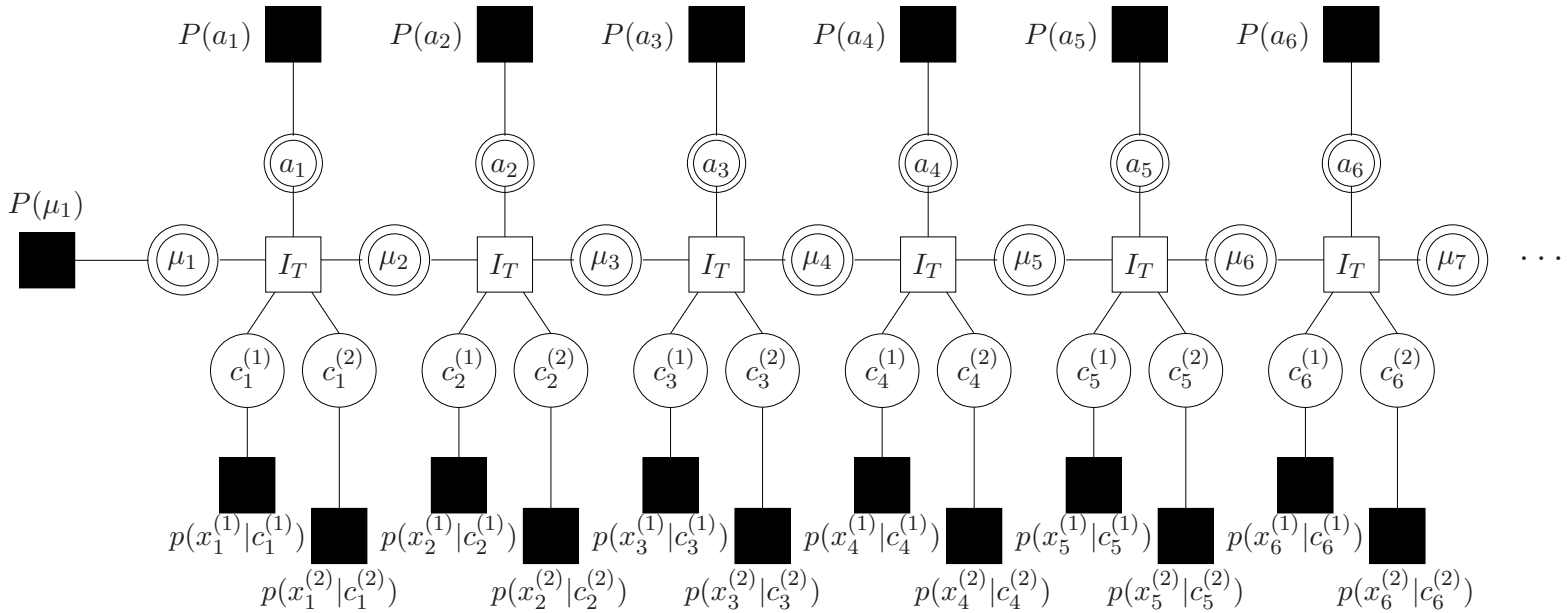
$P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \boldsymbol{\mu}|\mathbf{a})$ is an indicator function, equal to one when $\mathbf{a}, \mathbf{c}^{(1)}, \mathbf{c}^{(2)}$ and $\boldsymbol{\mu}$ are in a one-to-one correspondence, and to zero otherwise. It can be expressed as the product of indicator functions for each trellis section (Wiberg graph)

$$P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \boldsymbol{\mu}|\mathbf{a}) = P(\mu_1) \prod_{n=1}^{K} I_T(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1})$$

$I_T(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) = 1$ iff $(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) \in T$ (0 otherwise)

Decoding of Convolutional Codes (5/6)

$$P(\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{a}, \boldsymbol{\mu} | \mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \propto P(\mu_1) \prod_{n=1}^{K} p(x_n^{(1)}|c_n^{(1)}) p(x_n^{(2)}|c_n^{(2)}) I_T(a_n, c_n, \mu_n, \mu_{n+1}) P(a_n)$$



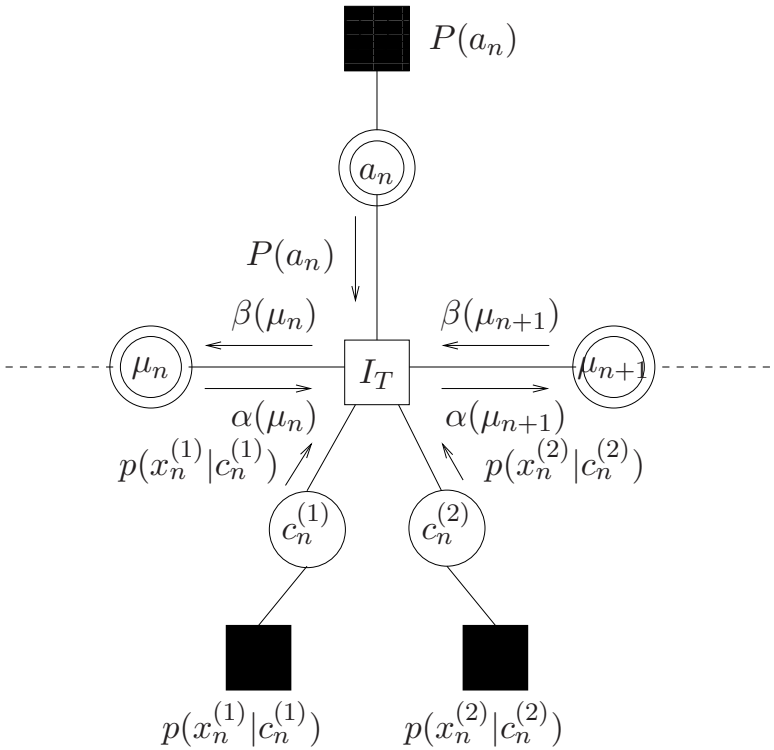
- The application of the SP algorithm leads to the BCJR algorithm.
- The graph is cycle-free $\Rightarrow$ the exact marginal pmfs $\{P(a_n|\mathbf{r})\}$ are obtained (also pmfs $\{P(c_n^{(1)}|\mathbf{r})\}$ and $\{P(c_n^{(2)}|\mathbf{r})\}$, useful in serially concatenated schemes, and $\{P(\mu_n|\mathbf{r})\}$ are also obtained).

Decoding of Convolutional Codes (6/6)

$$\alpha(\mu_{n+1}) = \sum_{\sim\{\mu_{n+1}\}} \alpha(\mu_n) p(x_n^{(1)}, x_n^{(2)} | c_n^{(1)}, c_n^{(2)}) I_T(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) P(a_n)$$

$$\beta(\mu_n) = \sum_{\sim\{\mu_n\}} \beta(\mu_{n+1}) p(x_n^{(1)}, x_n^{(2)} | c_n^{(1)}, c_n^{(2)}) I_T(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) P(a_n)$$

$$P(a_n|\mathbf{r}) = P(a_n) \sum_{\sim\{a_n\}} \alpha(\mu_n) \beta(\mu_{n+1}) p(x_n^{(1)}, x_n^{(2)} | c_n^{(1)}, c_n^{(2)}) I_T(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1})$$



BCJR in the Log Domain (1/2)

The BCJR algorithm can be conveniently implemented in the logarithmic domain.
A few maths: let us suppose that $x_1 > x2$

$$\ln[e^{x_1} + e^{x_2}] = \ln[e^{x_1}(1 + e^{x_2 - x_1})] = \ln[e^{x_1}(1 + e^{-|x_2 - x_1|})] = x_1 + \ln[1 + e^{-|x_2 - x_1|}] .$$

In general,

$$\ln[e^{x_1} + e^{x_2}] = \max(x_1, x_2) + \ln[1 + e^{-|x_2 - x_1|}] \triangleq x_1 \boxplus x_2 .$$

Note that $x_1 \boxplus x_2 = x_1$ when $(x_2 \to -\infty)$. If we have more than two exponentials, we can operate recursively:

$$\ln[\underbrace{e^{x_1} + e^{x_2}}_{e^x} + e^{x_3}] = \ln[e^x + e^{x_3}] = x \boxplus x_3 .$$

Since

$$e^x = e^{x_1} + e^{x_2} \to x = \ln[e^{x_1} + e^{x_2}] = x_1 \boxplus x_2$$

we finally have

$$\ln[e^{x_1} + e^{x_2} + e^{x_3}] = (x_1 \boxplus x_2) \boxplus x_3 .$$

BCJR in the Log Domain (2/2)

Let us now define

$$\overline{\alpha}(\mu_n) = \ln \alpha(\mu_n) \quad , \quad \overline{\beta}(\mu_n) = \ln \beta(\mu_n) \quad , \quad \overline{\gamma}(c_n^{(1)}, c_n^{(2)}) = \ln p(x_n^{(1)}, x_n^{(2)} | c_n^{(1)}, c_n^{(2)})$$

(note that $\overline{\gamma}(c_n^{(1)}, c_n^{(2)})$ is the same branch metric of the VA) and

$$\overline{I_T}(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) = \ln I_T(a_n, c_n, \mu_n, \mu_{n+1}) = \begin{cases} 0 & (a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) \in T \\ -\infty & \text{otherwise} . \end{cases}$$

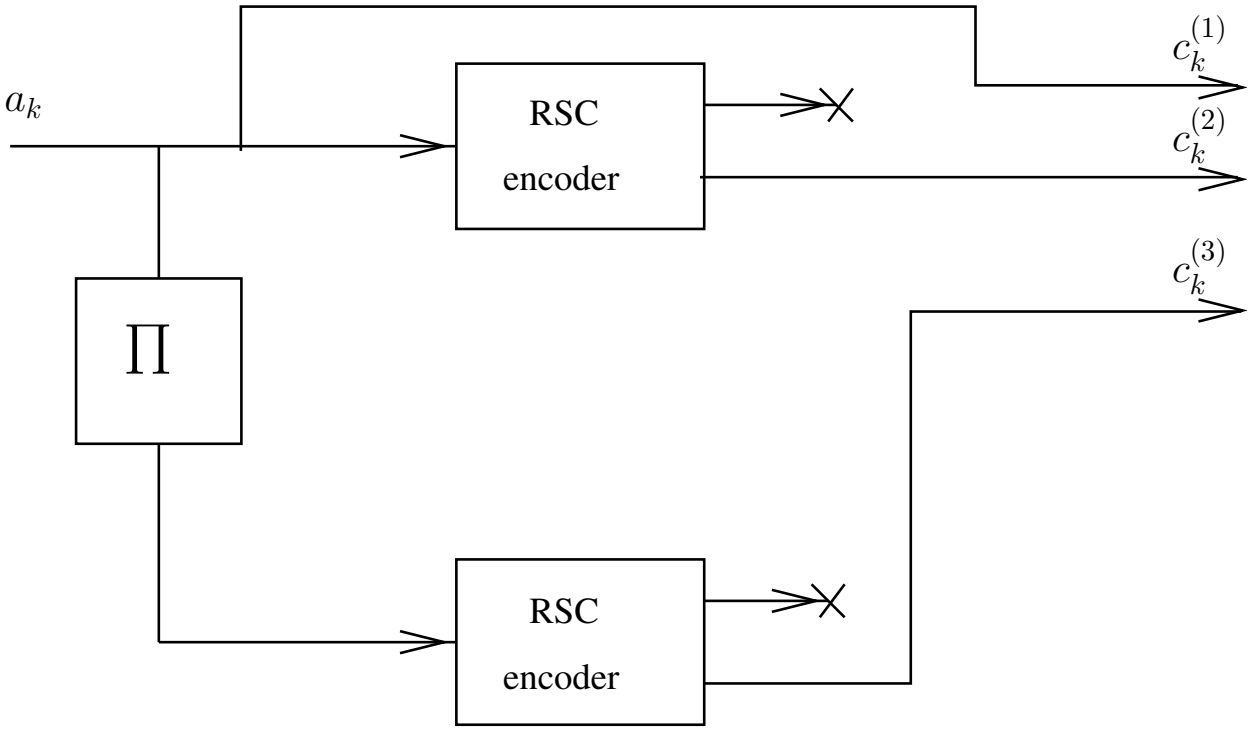
We have, considering for example the forward recursion only,

$$\begin{aligned} \overline{\alpha}(\mu_{n+1}) &= \ln \sum_{\sim\{\mu_{n+1}\}} \exp\left\{ \overline{\alpha}(\mu_n) + \overline{\gamma}(c_n^{(1)}, c_n^{(2)}) + \overline{I_T}(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) + \ln[P(a_n)] \right\} \\ &= \boxplus_{\sim\{\mu_{n+1}\}} \left[\overline{\alpha}(\mu_n) + \overline{\gamma}(c_n^{(1)}, c_n^{(2)}) + \overline{I_T}(a_n, c_n^{(1)}, c_n^{(2)}, \mu_n, \mu_{n+1}) + \ln[P(a_n)] \right] . \end{aligned}$$

The same for the backward recursion and the completion. By approximating $x_1 \boxplus x_2 \simeq \max(x_1, x_2)$ (max-log-map approximation), we obtain the VA.

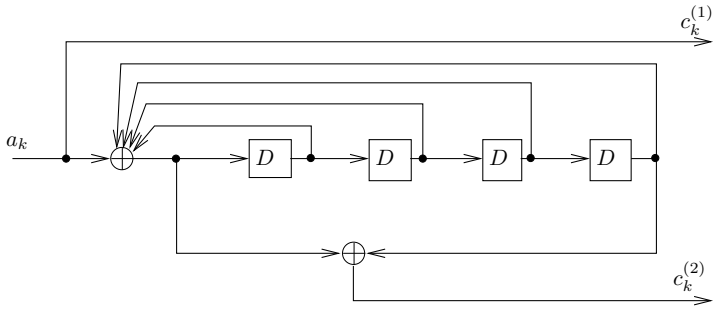
Turbo Codes (1/2)

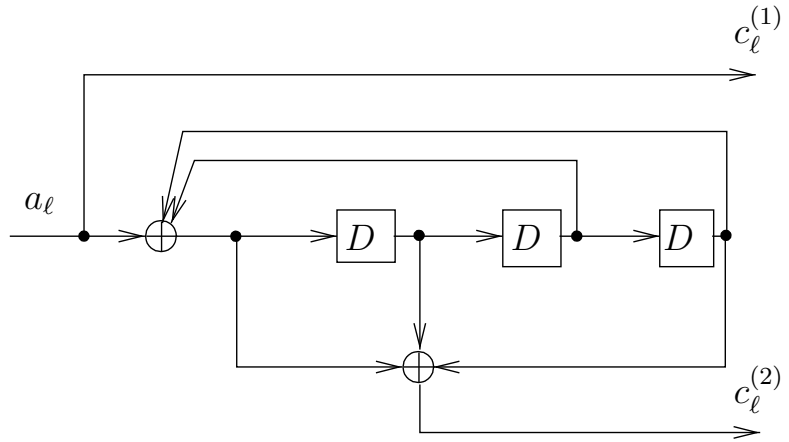
They represent an evolution of Forney's concatenated codes.
Original rate-1/3 turbo code by Berrou and Glavieux:



Turbo Codes (2/2)

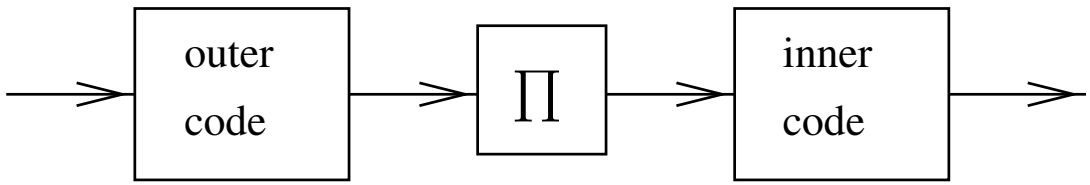
- Puncturing allows to obtain the desired rate.
- The component encoders must be recursive to have an interleaver gain ($\propto 1/N$).
- The interleaver is non-uniform. The minimum distance strongly depends on it. It is not critical for medium SNR values.
- Instead of working on the minimum distance, they attack multiplicity (they shape the distance spectrum).





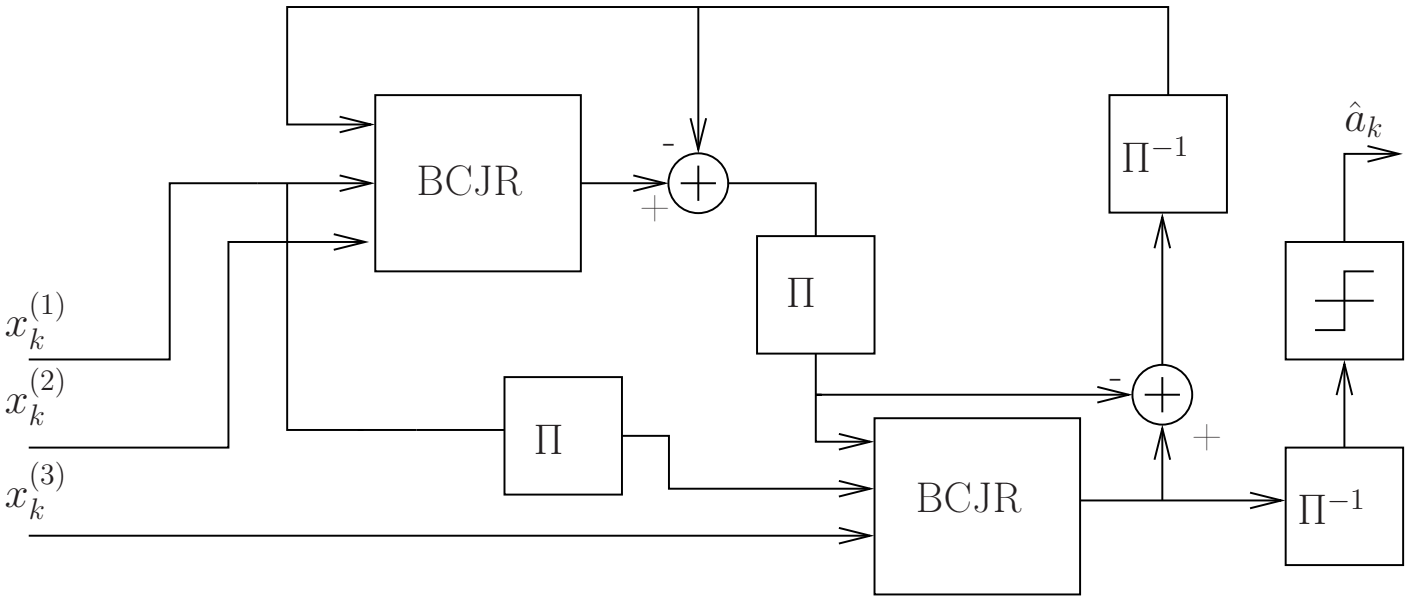
Serially Concatenated Codes

- The inner code must be recursive.
- Larger interleaver gain ($\propto 1/N^3$).



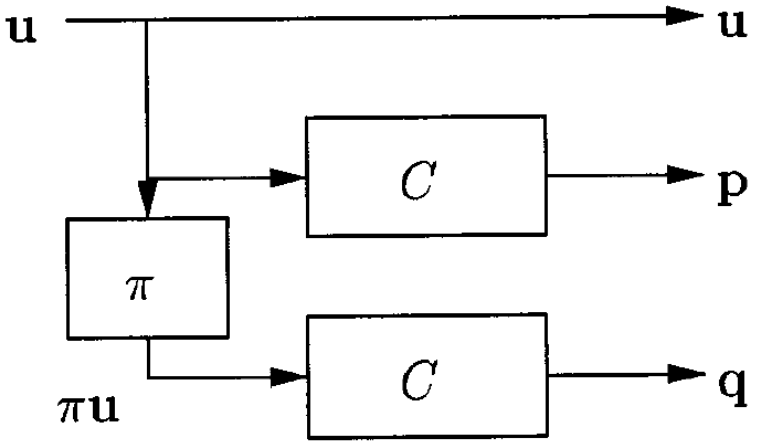
Iterative Decoding of Turbo Codes (1/3)

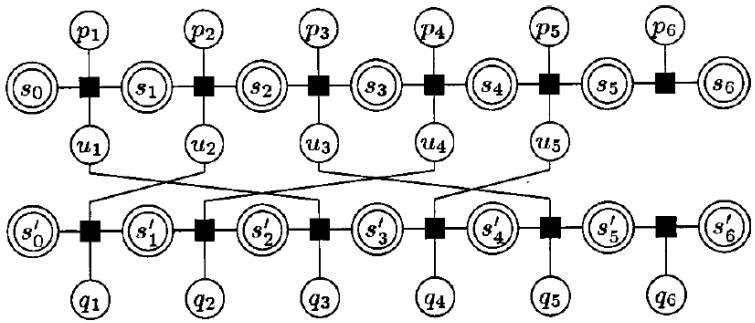
- Suboptimal iterative decoding: the two component decoders exchange soft information.
- Concept of extrinsic information.



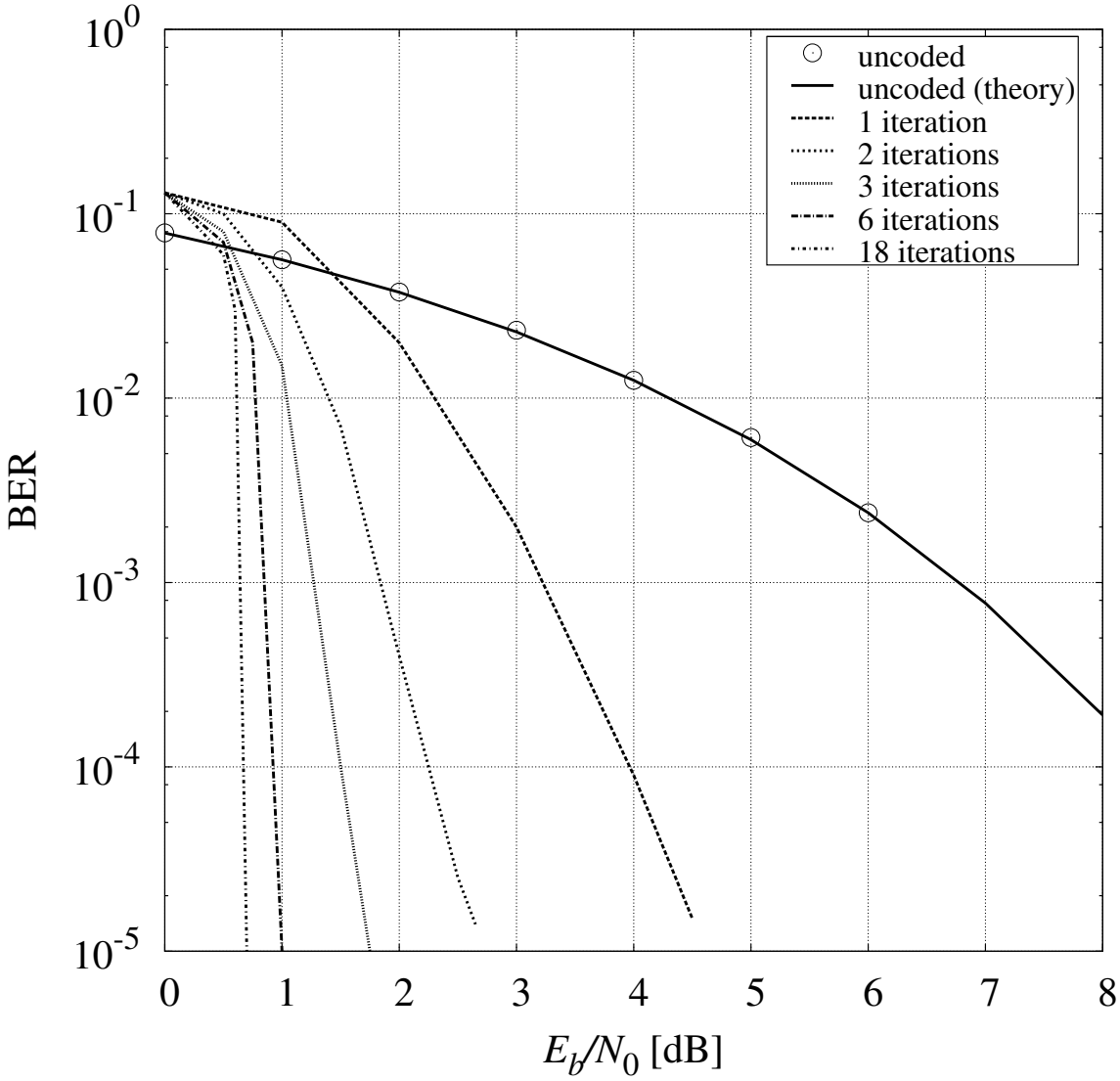
Iterative Decoding of Turbo Codes (2/3)

- Decoding is not performed on the Wiberg graph of the overall code $\Rightarrow$ cycles $\Rightarrow$ iterative decoding. Extrinsic information comes naturally.





Iterative Decoding of Turbo Codes (3/3)



Decoding of LDPC Codes (1/7)

- Definition: Iverson's convention. If P is a predicate (Boolean proposition), then

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{otherwise.} \end{cases}$$

- Definition: characteristic function of a code with codebook C

$$\chi_C(c_1, c_2, \ldots, c_N) = [(c_1, c_2, \ldots, c_N) \in C] .$$

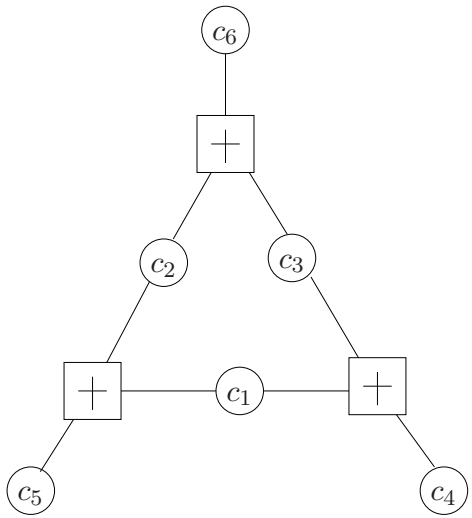
- Tanner graph for a linear code: it represents the characteristic function of the code. Ex.: a (6, 3) block code with parity check matrix $\mathbf{H}$ ($\mathbf{H}\mathbf{c}^T = \mathbf{0}$)

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

Decoding of LDPC Codes (2/7)

- Membership in C is completely determined by checking whether each of the three equations is satisfied

$$\chi_C(c_1, c_2, c_3, c_4, c_5, c_6) = [(c_1, c_2, c_3, c_4, c_5, c_6) \in C]$$
$$[c_1 \oplus c_2 \oplus c_5 = 0][c_2 \oplus c_3 \oplus c_6 = 0][c_1 \oplus c_3 \oplus c_4 = 0]$$



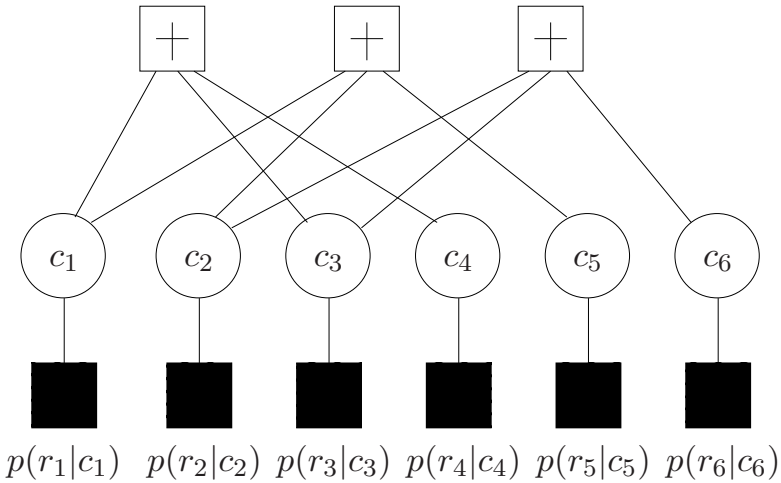
- Usually, this graph has cycles.

Decoding of LDPC Codes (3/7)

- In the MAP symbol decoding the aim is the computation of the marginal probabilities $\{P(c_n|\mathbf{r})\}$ from the joint pmf $P(\mathbf{c}|\mathbf{r})$.
- On an AWGN channel with received samples $r_n = c_n + w_n$ $(n = 1, \ldots, N)$, we have

$$P(\mathbf{c}|\mathbf{r}) \propto P(\mathbf{c})p(\mathbf{r}|\mathbf{c}) = \frac{1}{|C|}\chi_C(\mathbf{c})\prod_{n=1}^{N} p(r_n|c_n)$$

where $p(r_n|c_n) = \frac{1}{2\pi N_0}\exp\{-\frac{|r_n - c_n|^2}{2N_0}\}$



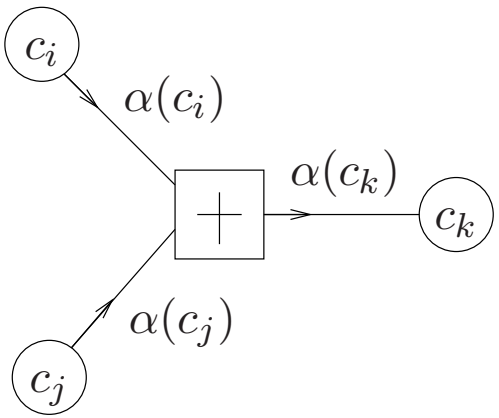
Decoding of LDPC Codes (4/7)

- The application of the SP algorithm to this factor graph allows the computation of the marginal probabilities $\{P(c_n|\mathbf{r})\}$ (the code is assumed to be systematic).
- The graph has cycles: the algorithm proceeds iteratively until all checks are satisfied (or a maximum number of iterations is reached).
- Usually, the flooding schedule is adopted.
- The messages on the edges of the graph are functions of discrete variables.
- In the case of binary block codes, if log-likelihood ratios (LLR) are used instead of probabilities, the passed messages become real numbers.

Decoding of LDPC Codes (5/7)

- A variable node computes the product of the incoming messages. In terms of LLRs, a variable node simply adds the incoming messages.
- Check node to variable node

$$\begin{aligned}\alpha(c_k) &= \sum_{\sim\{c_k\}} \alpha(c_i)\alpha(c_j)[c_i \oplus c_j \oplus c_k = 0] \\ &= \sum_{c_i}\sum_{c_j} \alpha(c_i)\alpha(c_j)[c_i \oplus c_j \oplus c_k = 0]\,.\end{aligned}$$



Decoding of LDPC Codes (6/7)

- Hence

$$\begin{aligned}\alpha(c_k=0) &= \alpha(c_i=0)\alpha(c_j=0)+\alpha(c_i=1)\alpha(c_j=1)\\ \alpha(c_k=1) &= \alpha(c_i=0)\alpha(c_j=1)+\alpha(c_i=1)\alpha(c_j=0)\,.\end{aligned}$$

- By defining (log-likelihood ratio, LLR)

$$\ell_k = \ln\frac{\alpha(c_k=0)}{\alpha(c_k=1)}$$

one has

$$\ell_k = 2\tanh^{-1}\left[\tanh\left(\frac{\ell_i}{2}\right)\tanh\left(\frac{\ell_j}{2}\right)\right] \simeq \mathrm{sgn}(\ell_i)\mathrm{sgn}(\ell_j)\min(|\ell_i|,|\ell_j|)\,.$$

- Alternatively, instead of using LLRs, we can use the signed log-likelihood difference (SLLD):

$$\delta_k = \mathrm{sgn}[\alpha(c_k=1)-\alpha(c_k=0)]\ln|\alpha(c_k=1)-\alpha(c_k=0)|\,.$$

In this case, the operation at variable node is quite complex whereas at a check node it is

$$\delta_k = \mathrm{sgn}(\delta_i)\mathrm{sgn}(\delta_j)[|\delta_i|+|\delta_j|]$$

Decoding of LDPC Codes (7/7)

The most effective way to implement a binary LDPC decoder must:

- work with LLRs at variable nodes $\rightarrow$ each variable node has to perform a sum of the incoming messages;
- before sending the messages to the check nodes, transform each LLR into a SLLD with the NL trasformation (implemented through a LUT):

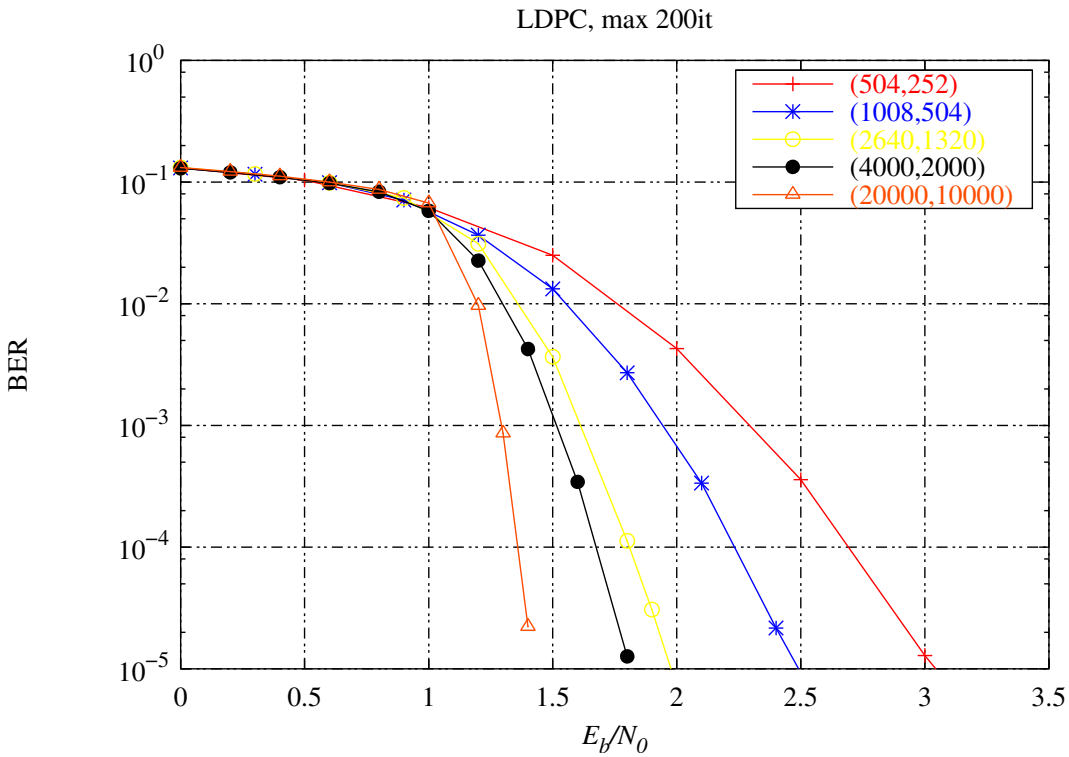
$$\delta_k = -\mathrm{sgn}(\ell_k) \ln \frac{1 - e^{-|\ell_k|}}{1 + e^{-|\ell_k|}} ;$$

- work with SLLDs at check nodes and transform back the SLLDs into LLRs before sending the messages back to the variable nodes.

This decoding algorithm is used to decode Low-Density Parity-Check (LDPC) codes, very long block codes with sparse (to reduce the decoding complexity) parity-check matrix.
In practice, only cycles of length 4 must be avoided.
Note that an “implicit” interleaver is present.

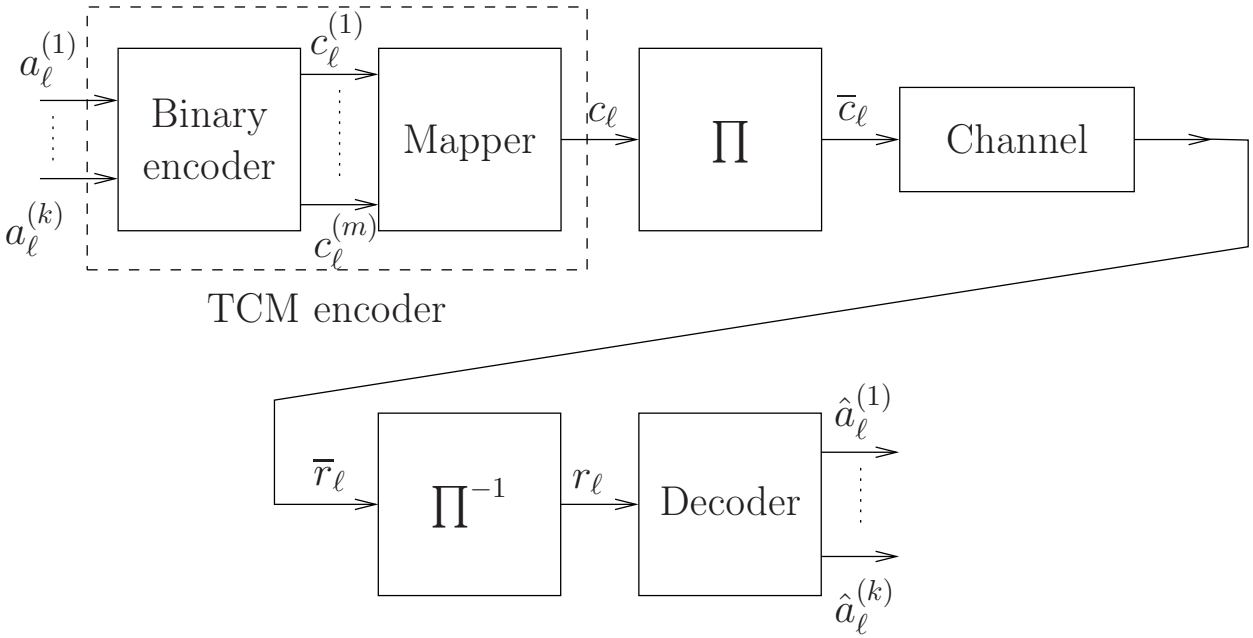
Performance of LDPC Codes

- Regular LDPC codes were proposed by Gallager in 1963.



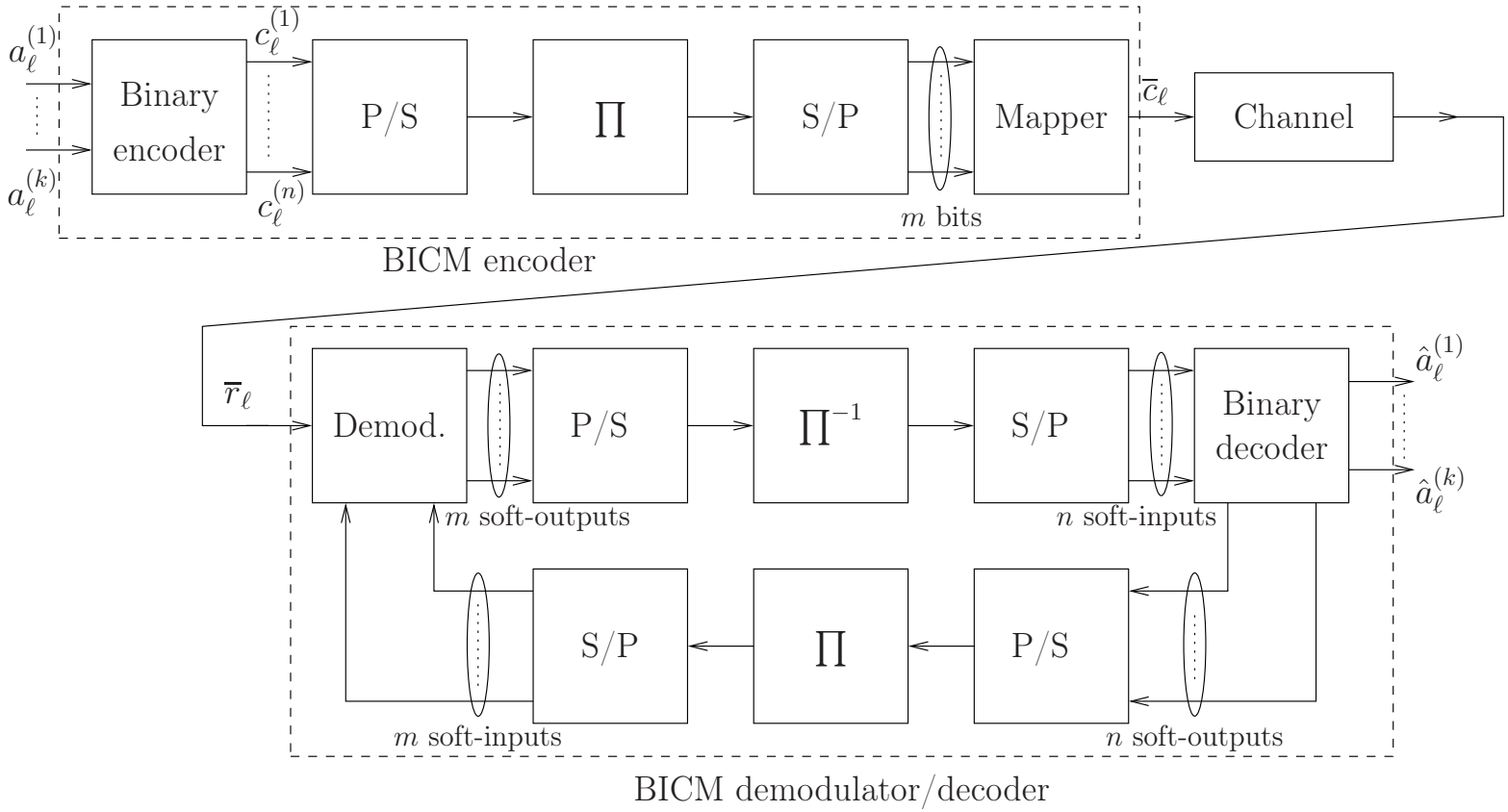
- Irregular LDPC codes, proposed in 2001, outperform the best known turbo codes. The degree distribution of variable and check nodes is properly optimized.

TCM on Fading Channels



The interleaver is required against deep fade events. On fading channels the asymptotic performance depends on the minimum Hamming distance between coded sequences $\rightarrow$ a redesign of the code is required with respect to classical TCMs.

BICM (1/3)



It was born to simplify the code design for fading channels. BICM performs well also for AWGN channels (see DVB-S2 standard). When Gray mapping is used at the receiver, there is no gain in iterating between demodulator (demapper) and decoder. For other maps, the gain can be significant.

BICM (2/3)

Toy example with a block code mapped onto a 8PSK modulation.
Let us consider a block code with $N = 6$ (the one in the previous example).
Bits c_1, c_2, c_5, are mapped onto an 8PSK symbol Δ_1 whereas bits c_3, c_4, c_6 are mapped onto Δ_2.
The received samples are

$$r_1 = \Delta_1 + w_1 \quad , \quad r_2 = \Delta_2 + w_2 \, .$$

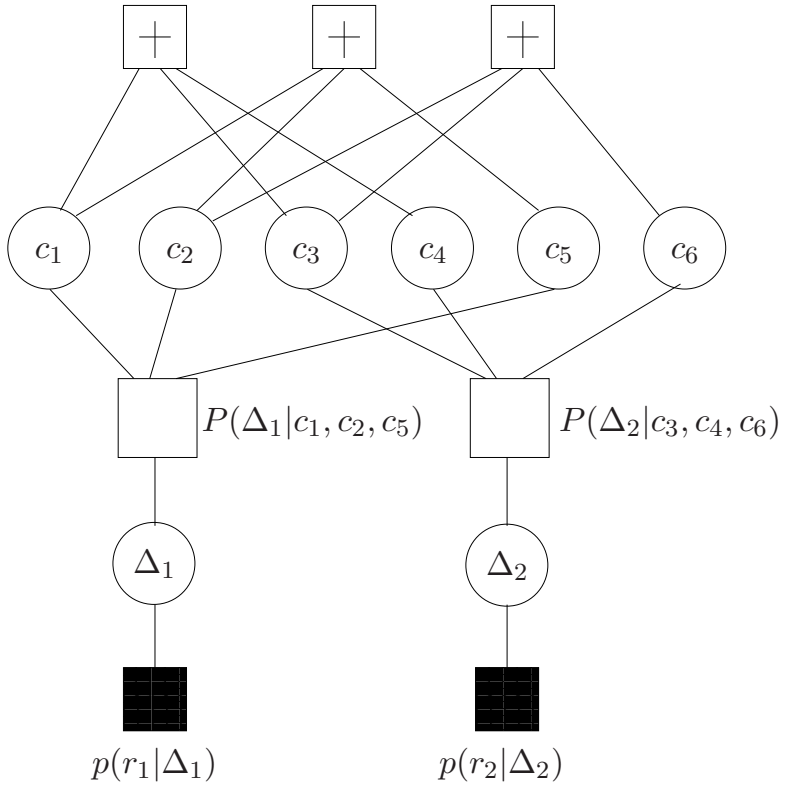
We have

$$P(\boldsymbol{\Delta}_1^2, \mathbf{c}_1^6 | \mathbf{r}_1^2) \propto p(r_1|\Delta_1) p(r_2|\Delta_2) P(\Delta_1|c_1, c_2, c_5) P(\Delta_2|c_3, c_4, c_6) P(\mathbf{c}_1^6)$$

$P(\Delta_1|c_1, c_2, c_5)$ and $P(\Delta_2|c_3, c_4, c_6)$ are two indicator functions and $P(\mathbf{c}_1^6) = \frac{1}{|C|} \chi_C(\mathbf{c})$.

BICM (3/3)

Overall FG



Iterative demapping and decoding may be convenient when a mapping different from the Gray one is adopted.

Outline

Channels with Unknown Parameters (1/10)

In addition to unknown data, we usually have other unknown parameters (collected into vector θ): phase and frequency of the carrier, the propagation delay, the channel attenuation.
We will see the main concept by concentrating on a particular discrete-time model. We will see later the cases when a discrete-time model cannot be used (e.g., timing estimate).
Model: linear modulations without ISI on the non-coherent time-varying channel:

$$r_k = c_k e^{\jmath\theta_k} + w_k \, .$$

Channels with Unknown Parameters (2/10)

We can model the parameters as random variables with known pdf. In this case, an optimal receiver exists (considering for example MAP sequence detection):

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{a}) = \underset{\mathbf{a}}{\operatorname{argmax}} \int p(\mathbf{r}|\mathbf{a},\boldsymbol{\theta}) \underbrace{p(\boldsymbol{\theta}|\mathbf{a})}_{p(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}\,.$$

Example: non-coherent, time-invariant channel (θ has uniform distribution)

$$p(\mathbf{r}|\mathbf{a}) = \left(\frac{1}{2\pi N_0}\right)^N \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{N-1}\left[|r_k|^2 + |c_k|^2\right]\right\} \mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{k=0}^{N-1} r_k c_k^*\right|\right)$$

$$\begin{aligned}\hat{\mathbf{a}} &= \underset{\mathbf{a}}{\operatorname{argmax}} \left[\mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{k=0}^{N-1} r_k c_k^*\right|\right) \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{N-1}|c_k|^2\right\}\right] \\ &= \underset{\mathbf{a}}{\operatorname{argmax}} \left[\ln \mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{k=0}^{N-1} r_k c_k^*\right|\right) - \frac{1}{2N_0}\sum_{k=0}^{N-1}|c_k|^2\right]\,.\end{aligned}$$

Channels with Unknown Parameters (3/10)

Problems:

Often $p(\mathbf{r}|\mathbf{a})$ cannot be computed in closed form. When possible, the strategy has a complexity which is exponential in $N \rightarrow$ the system has infinite memory.

Solutions:

Introduction of approximations, e.g., strategy of memory truncation.

Examples of receivers based on memory truncation:

Noncoherent sequence detection and linear predictive receivers for fading channels.

Advantages and drawbacks:

The derived detection strategies usually have a strong robustness in the presence of time-varying parameters but their complexity is usually high.

Channels with Unknown Parameters (4/10)

Strategy of memory truncation:
Starting from the optimal strategy

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{a})$$

we may observe that

$$p(\mathbf{r}|\mathbf{a}) = p(\mathbf{r}|\mathbf{c}) = \prod_{k=0}^{N-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}) = \prod_{k=0}^{N-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k)$$

having assumed that the system is causal. If we now adopt the following approximation (intuitive for a time-varying channel, because we are assuming that the received samples far from the present are less correlated with it)

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) \simeq p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k)$$

the detection strategy becomes

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} \ln p(\mathbf{r}|\mathbf{a}) \simeq \underset{\mathbf{a}}{\operatorname{argmax}} \sum_{k=0}^{N-1} \ln p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k)$$

($C \geq R$) that can be implemented using the VA with branch metrics $\lambda_k(a_k, \sigma_k) = \ln p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k)$ and state

$$\sigma_k = (c_{k-1}, c_{k-2}, \ldots, c_{k-C}) = (a_{k-1}, a_{k-2}, \ldots, a_{k-C}, \mu_{k-C})$$

Channels with Unknown Parameters (5/10)

Noncoherent time-invariant channel. In this case it is $R = C$ and

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) = \frac{p(\mathbf{r}_{k-C}^{k}|\mathbf{c}_{k-C}^{k})}{p(\mathbf{r}_{k-C}^{k-1}|\mathbf{c}_{k-C}^{k})} = \frac{p(\mathbf{r}_{k-C}^{k}|\mathbf{c}_{k-C}^{k})}{p(\mathbf{r}_{k-C}^{k-1}|\mathbf{c}_{k-C}^{k-1})}$$

where

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k-1}) = \frac{1}{2\pi N_0} \exp\left\{-\frac{1}{2N_0}\left[|r_k|^2 + |c_k|^2\right]\right\} \frac{\mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{\ell=0}^{C} r_{k-\ell} c_{k-\ell}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{\ell=1}^{C} r_{k-\ell} c_{k-\ell}^*\right|\right)} .$$

The branch metric is thus

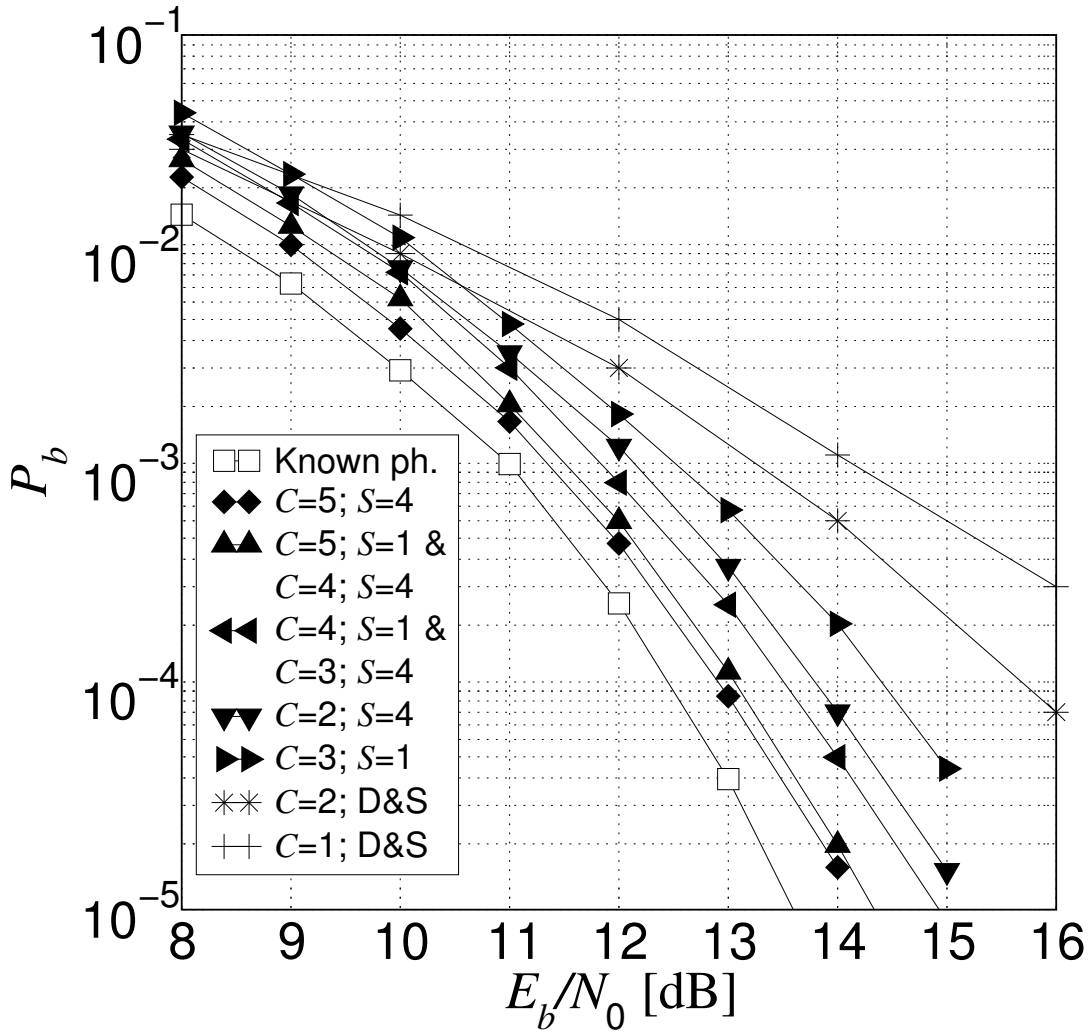
$$\ln p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k-1}) \propto \ln \mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{\ell=0}^{C} r_{k-\ell} c_{k-\ell}^*\right|\right) - \ln \mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{\ell=1}^{C} r_{k-\ell} c_{k-\ell}^*\right|\right) - \frac{1}{2N_0}|c_k|^2$$

and using the approximation $\ln \mathrm{I}_0(x) \simeq x$,

$$\ln p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k-1}) \propto \left|\sum_{\ell=0}^{C} r_{k-\ell} c_{k-\ell}^*\right| - \left|\sum_{\ell=1}^{C} r_{k-\ell} c_{k-\ell}^*\right| - \frac{1}{2}|c_k|^2 .$$

Channels with Unknown Parameters (6/10)

16QAM with quadrant differential encoding



Channels with Unknown Parameters (7/10)

When we model the unknown parameters as deterministic unknown, the optimal receiver is not defined unless a uniformly most powerful (UMP) test exists.

UMP test: $\mathbf{r}$ depends on θ and $\mathbf{a}$. Let us consider the optimal receiver under the assumption that θ is perfectly known (and be $\overline{\theta}$ its value). This receiver $\mathcal{R}_{\overline{\theta}}$ is based on the strategy

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{a}, \overline{\theta}) \,.$$

A test UMP exists when $\forall\overline{\theta}$, the receiver $\mathcal{R}_{\overline{\theta}}$ does not change. In this case $R_{\overline{\theta}}$ is the optimal receiver also when θ is unknown.

Channels with Unknown Parameters (8/10)

Examples

Unknown and time invariant phase:

Received samples $r_k = c_k e^{\jmath\theta} + w_k$.

Receiver $\mathcal{R}_{\overline{\theta}}$ is based on the strategy

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{a}, \overline{\theta}) = \underset{\mathbf{a}}{\operatorname{argmax}} \sum_{k=0}^{N-1} \left| r_k - c_k e^{\jmath\overline{\theta}} \right|^2 = \underset{\mathbf{a}}{\operatorname{argmax}} \sum_{k=0}^{N-1} \left| r_k e^{-\jmath\overline{\theta}} - c_k \right|^2$$

$\rightarrow$ an UMP test does not exist (the received samples have to be derotated).

Unknown and positive channel attenuation with uncoded PSK:

In this case an UMP test exists.

Channels with Unknown Parameters (9/10)

When an UMP test does not exist, we have a few heuristic strategies available.
Generalized likelihood.
In this case the joint pdf is maximized:

$$(\hat{\mathbf{a}}, \hat{\theta}) = \arg\max_{(\mathbf{a},\theta)} p(\mathbf{r}|\mathbf{a}, \theta)$$

Example for an unknown and time invariant phase:

$$\begin{aligned} p(\mathbf{r}|\mathbf{a}, \theta) &= \left(\frac{1}{2\pi N_0}\right)^N \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{N-1}|r_k|^2\right\} \\ &\quad \cdot \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{N-1}|c_k|^2 + \frac{1}{N_0}\left|\sum_{k=0}^{N-1} r_k c_k^*\right| \cos\left[\theta - \phi(\mathbf{r}, \mathbf{c})\right]\right\}. \end{aligned}$$

Thus, it can be maximized by choosing

$$\begin{aligned} \hat{\mathbf{a}} &= \arg\max_{\mathbf{a}} \left|\sum_{k=0}^{N-1} r_k c_k^*\right| - \frac{1}{2}\sum_{k=0}^{N-1}|c_k|^2 \\ \hat{\theta} &= \phi(\mathbf{r}, \hat{\mathbf{c}}) = \arg\left[\sum_{k=0}^{N-1} r_k \hat{c}_k^*\right] \end{aligned}$$

Channels with Unknown Parameters (10/10)

Synchronization.

It is based on the following two steps:

1. Find (in some way) an estimate of parameter $\hat{\theta}$.
2. Use $\hat{\theta}$ for detection as if it were the true value of the parameter, thus according to the strategy

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{a}, \hat{\theta}) \,.$$

How to find $\hat{\theta}$? We will concentrate on the main ideas with reference to ML estimation (all these concepts can be extended to the case of Bayesian estimation).

Synchronization

Estimators' classifications:

1. It can be based on the way estimation is performed: open loop (OL) vs closed loop (CL). We will see that for each OL estimator, it is possible to derive a corresponding CL estimator.
2. It can be based on the knowledge about data:
   1. data aided (DA);
   2. decision directed (DD);
   3. soft-decision directed (SDD);
   4. non data aided (NDA).

Omitted for a lack of time: performance limits of estimators (CRB and MCRB).

DA Estimation (1/4)

In packet transmissions, one or more fields of known data are often present (preamble, midamble, pilot fields). If the channel coherence time is larger than the packed duration, we can estimate the parameter on the pilot fields and use the estimate for the whole packet. Assuming we have L consecutive known symbols, the DA-ML estimator is

$$\hat{\theta}_{DA} = \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1}, \theta)\,.$$

Example: Frequency estimation (Rife & Boorstyn algorithm).
Let us consider the noncoherent time-varying channel with $\theta_k = 2\pi FkT + \phi$ where F is the frequency offset and $\phi \in \mathcal{U}[0, 2\pi)$. We would like to estimate F without estimating ϕ in advance. Being

$$p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1}, F) = \left(\frac{1}{2\pi N_0}\right)^L \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{L-1}\left[|r_k|^2 + |a_k|^2\right]\right\} \mathrm{I}_0\left(\frac{1}{N_0}\left|\sum_{k=0}^{L-1} r_k a_k^* e^{-\jmath 2\pi FkT}\right|\right)$$

we have

$$\hat{F}_{DA} = \underset{F}{\operatorname{argmax}}\, p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1}, F) = \underset{F}{\operatorname{argmax}} \left|\sum_{k=0}^{L-1} r_k a_k^* e^{-\jmath 2\pi FkT}\right|$$

DA Estimation (2/4)

$R(F) = \sum_{k=0}^{L-1} r_k a_k^* e^{-\jmath 2\pi FkT}$ is the Fourier transform of the sequence $r_k a_k^*$. The maximum of its magnitude can be obtained through a procedure based on the following two steps:

1. coarse search based on the samples of $R(F)$ computed through FFT;
2. fine search by interpolating the previously computed samples and followed by a local search.

The R&B algorithm is the best one in terms of performance. It reaches the CRB for medium-to-high SNR values (for low SNR values the performance degrades due to the outliers). However, it has a very high computational complexity.
The (heuristic) algorithm by Mengali & Morelli has the same performance but a much lower complexity.

DA Estimation (3/4)

Let us consider a PSK transmission and define

$$z_k = r_k a_k^* = e^{\jmath(2\pi FkT+\phi)} + w_k' = e^{\jmath(2\pi FkT+\phi)}[1+\tilde{w}_k] \tag{11}$$

($w_k' = w_k a_k^*$ and $\tilde{w}_k = w_k' e^{-\jmath(2\pi FkT+\phi)}$). The algorithm is based on the computation of the sample correlation

$$R_m = \frac{1}{L-m}\sum_{k=m}^{L-1} z_k z_{k-m}^*\,, \qquad m = 1,2,\ldots,N \tag{12}$$

where N is a design parameter ($N \leq L/2$). Substituting (11) into (12) we obtain

$$R_m = e^{\jmath(2\pi FmT)}[1+\gamma_m]$$

with

$$\gamma_m = \frac{1}{L-m}\sum_{k=m}^{L-1}[\tilde{w}_k + \tilde{w}_{k-m}^* + \tilde{w}_k\tilde{w}_{k-m}^*] \simeq \frac{1}{L-m}\sum_{k=m}^{L-1}[\tilde{w}_k + \tilde{w}_{k-m}^*]\,.$$

DA Estimation (4/4)

In high SNR conditions

$$\arg[1+\gamma_m] = \arctan\frac{\gamma_{I,m}}{1+\gamma_{R,m}} \simeq \arctan\gamma_{I,m} \simeq \gamma_{I,m}$$

and thus

$$[\arg[R_m] - \arg[R_{m-1}]]_{2\pi} \simeq 2\pi FT + \gamma_{I,m} - \gamma_{I,m-1}$$

$\rightarrow$ we have a linear dependence on F with an approximately Gaussian noise. Thus, the optimal estimator is a linear MMSE estimator and we obtain

$$\hat{F} = \frac{1}{2\pi T}\sum_{m=1}^{N}\lambda_m\left[\arg[R_m] - \arg[R_{m-1}]\right]_{2\pi}$$

where

$$\lambda_m = \frac{3[(L-m)(L-m+1) - N(L-N)]}{N(4N^2 - 6NL + 3L^2 - 1)}\,.$$

The algotithm can be further simplified by replacing R_m with

$$\overline{R}_m = \frac{1}{L-m}\sum_{k=m}^{L-1}\exp\left\{\jmath\left[\arg(z_k) - \arg(z_{k-m})\right]\right\}\,, \quad m = 1,2,\ldots,N\,.$$

OL DA Phase Estimation

Example: Phase estimation.

The channel model is

$$r_k = a_k e^{\jmath\theta} + w_k \, .$$

It is thus

$$\begin{aligned} p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1},\theta) &= \left(\frac{1}{2\pi N_0}\right)^L \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{L-1}\left|r_k - a_k e^{\jmath\theta}\right|^2\right\} \\ &= \left(\frac{1}{2\pi N_0}\right)^L \exp\left\{-\frac{1}{2N_0}\sum_{k=0}^{L-1}\left[|r_k|^2 + |a_k|^2\right]\right\} \\ &\quad \cdot \exp\left\{\frac{1}{N_0}\left|\sum_{k=0}^{L-1} r_k a_k^*\right| \cos\left[\theta - \phi(\mathbf{r},\mathbf{a})\right]\right\} \end{aligned}$$

and

$$\hat{\theta}_{DA} = \phi(\mathbf{r},\mathbf{a}) = \arg\left[\sum_{k=0}^{L-1} r_k a_k^*\right] \, .$$

This is an OL estimator that can be used when the phase can be assumed constant over the whole packet. It reaches the CRB at high SNR.

DA-PLL (1/5)

When the channel phase is time-varying, it is better to adopt a CL estimator. We would like to update the estimate as soon as a new received sample has arrived. Application: we can first estimate the phase using the preamble and then turn on the closed loop estimator in correspondence to the pilot fiels to update the estimate.
Let us define

$$\begin{aligned}\Lambda(\theta) &= \ln p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1},\theta) = \\ &= \ln\left(\frac{1}{2\pi N_0}\right)^L - \frac{1}{2N_0}\sum_{k=0}^{L-1}\left[|r_k|^2+|a_k|^2\right] + \frac{1}{N_0}\mathcal{R}\left[e^{-\jmath\theta}\sum_{k=0}^{L-1} r_k a_k^*\right]\end{aligned}$$

whose derivative w.r.t. θ is

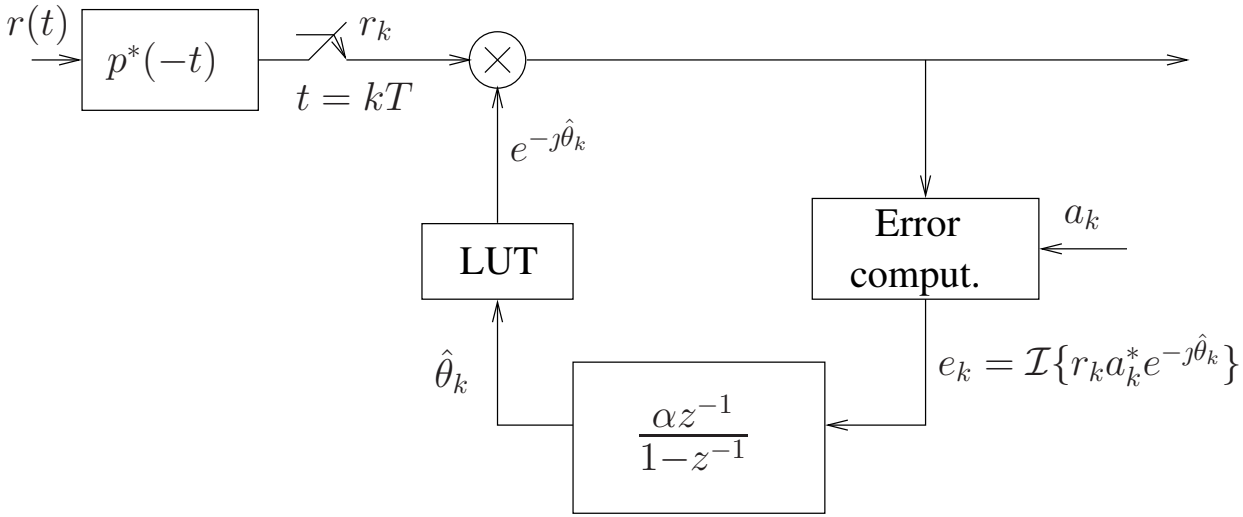
$$\frac{\mathrm{d}\Lambda(\theta)}{\mathrm{d}\theta} = \frac{1}{N_0}\mathcal{R}\left[-\jmath e^{-\jmath\theta}\sum_{k=0}^{L-1} r_k a_k^*\right] = \frac{1}{N_0}\mathcal{I}\left[e^{-\jmath\theta}\sum_{k=0}^{L-1} r_k a_k^*\right].$$

DA-PLL (2/5)

Considering the k-th term of the summation, we can recursively update the phase estimate according to the gradient algorithm:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \underbrace{\mathcal{I}\left[e^{-\jmath\hat{\theta}_k} r_k a_k^*\right]}_{e_k \text{ error signal}}$$

where α is the step-size—the larger its value, the lower the acquisition time and the capability of the estimator to track the channel variations (in case of phase noise), but the larger the variations at steady state.
The recursive equation defines a first-order phase-locked loop (PLL).



DA-PLL (3/5)

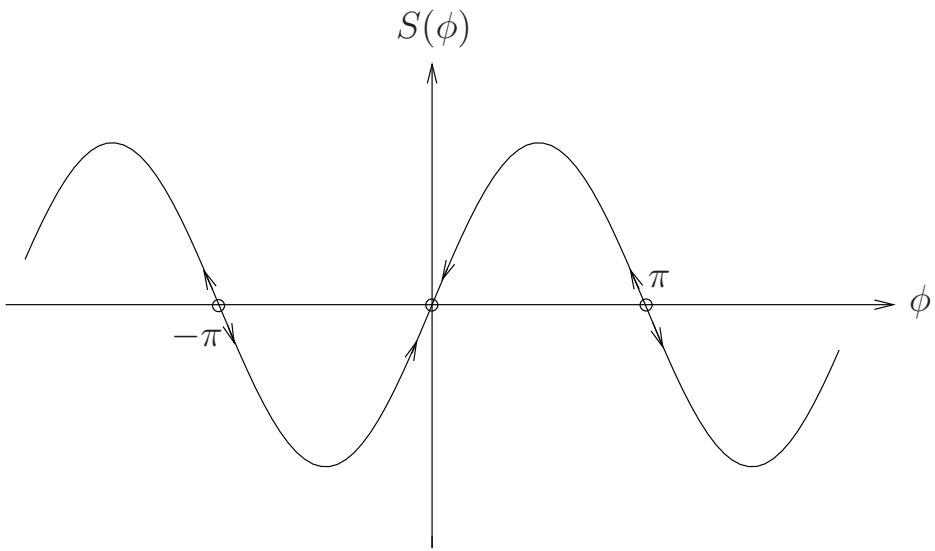
PLL Analysis:

S-curve:

$$S(\phi) = E\{e_k | \phi_k = \phi\}$$

where $\phi_k = \theta_k - \hat{\theta}_k$. It is the average of the error signal when the feedback is disconnected. For our DA-PLL and assuming a PSK, we have

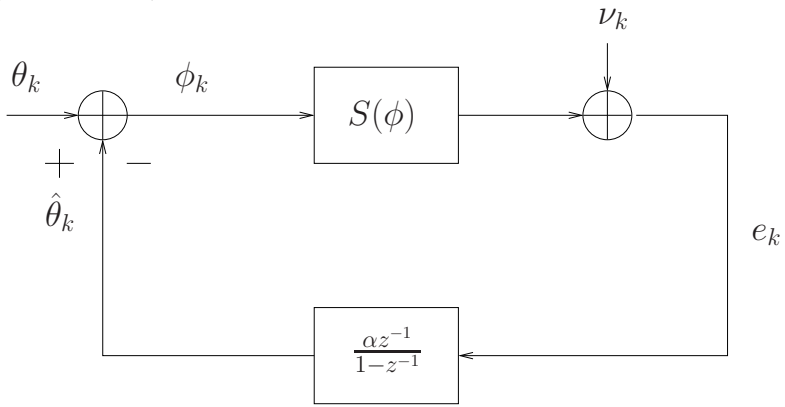
$$S(\phi) = E\{|a_k|^2\} \sin \phi \, .$$



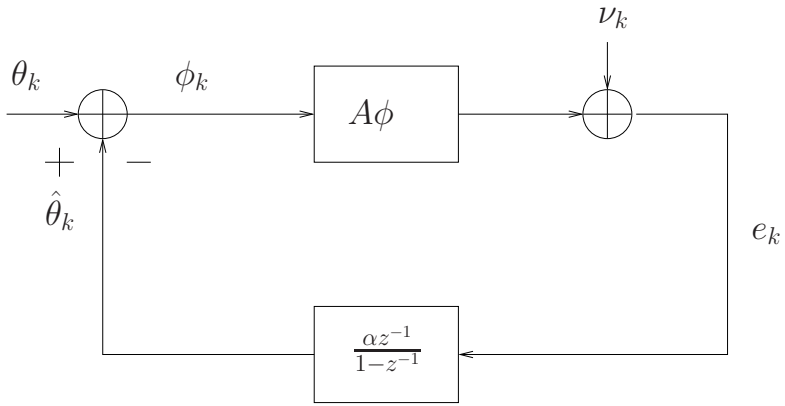
Stable and unstable equilibrium points. Hang-up.

DA-PLL (4/5)

Defining $\nu_k = e_k - S(\theta_k - \hat{\theta}_k)$, we have the following PLL equivalent model:



When the steady state is reached, we can linearize the model by defining $A = \left.\frac{\mathrm{d}S(\phi)}{\mathrm{d}\phi}\right|_{\phi=0}$



DA-PLL (5/5)

PLL transfer function:

$$H(z) = \left.\frac{\Phi(z)}{V(z)}\right|_{\theta_k=0} = -\frac{F(z)}{1+AF(z)}$$

where $\Phi(z)$ and $V(z)$ are the z transforms of ϕ_k and ν_k, and $F(z) = \frac{\alpha z^{-1}}{1-z^{-1}}$. Hence

$$H(z) = \frac{\alpha}{z-(1-A\alpha)} .$$

The mean square estimation error can be computed taking into account that the power spectral density of ϕ_k is $N_0 \left|H(e^{j2\pi fT})\right|^2$. Thus

$$\begin{aligned} \text{var}[\hat{\theta}_k] &= E\{[\theta_k - \hat{\theta}_k]^2\} = E\{\phi_k^2\} = N_0 T \int_{-\frac{1}{2T}}^{\frac{1}{2T}} \left|H(e^{\jmath 2\pi fT})\right|^2 df \\ &= N_0 \int_{-\frac{1}{2}}^{\frac{1}{2}} \left|H(e^{\jmath 2\pi fT})\right|^2 d(fT) = 2N_0 B_{eq} T \end{aligned}$$

having defined the PLL normalized equivalent bandwidth as

$$B_{eq}T = \frac{1}{|H(1)|^2} \int_0^{\frac{1}{2}} \left|H(e^{\jmath 2\pi fT})\right|^2 \mathrm{d}(fT) = \frac{\alpha A}{2(2-\alpha A)} .$$

DA Timing Synchronization (1/3)

Let us consider the problem of DA timing synchronization for linear modulations:

$$r(t) = \underbrace{\sum_k a_k p(t - \tau - kT)}_{s(t-\tau,\mathbf{a})} + w(t)$$

$$\widehat{\tau} = \underset{\widetilde{\tau}}{\operatorname{argmax}} \left\{ \underbrace{\mathcal{R}\left[\int_{-\infty}^{\infty} r(t) s^*(t - \widetilde{\tau}, \mathbf{a})\, \mathrm{d}t\right] - \frac{1}{2}\int_{-\infty}^{\infty} |s(t - \widetilde{\tau}, \mathbf{a})|^2\, \mathrm{d}t}_{\Lambda(\widetilde{\tau})} \right\}.$$

We can use the same recursive technique used to derive the PLL:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \left.\frac{\mathrm{d}\Lambda(\widetilde{\tau})}{\mathrm{d}\widetilde{\tau}}\right|_{\widetilde{\tau}=\hat{\tau}_k} = \hat{\tau}_k + \alpha\, \Lambda'(\widetilde{\tau})\big|_{\widetilde{\tau}=\hat{\tau}_k}$$

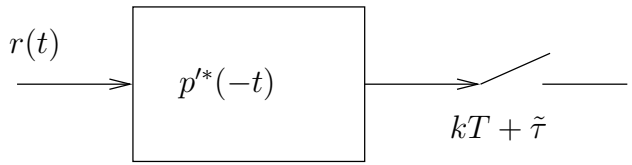
DA Timing Synchronization (2/3)

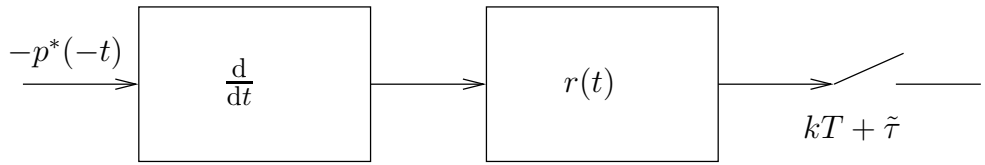
We can neglect the term which depends on the energy of the signal:

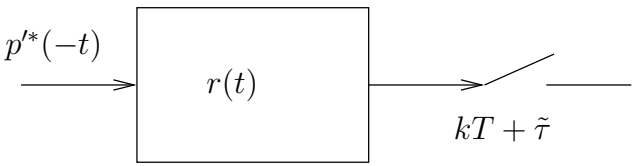
$$\Lambda(\widetilde{\tau}) \simeq \mathcal{R}\left[\int_{-\infty}^{\infty} r(t)s^*(t-\widetilde{\tau},\mathbf{a})\,\mathrm{d}t\right] = \mathcal{R}\left[\sum_k a_k^* \int_{-\infty}^{\infty} r(t)p^*(t-\widetilde{\tau}-kT)\mathrm{d}t\right].$$

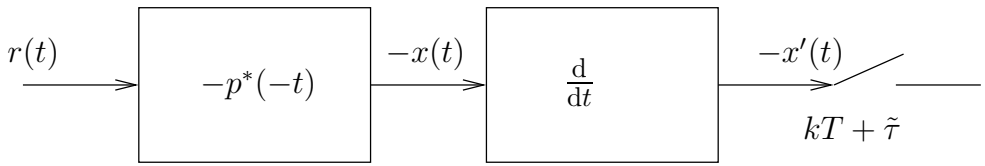
Thus

$$\Lambda'(\widetilde{\tau}) \simeq -\mathcal{R}\left[\sum_k a_k^* \int_{-\infty}^{\infty} r(t)p'^*(t-\widetilde{\tau}-kT)\mathrm{d}t\right] = \mathcal{R}\left[\sum_k x'(kT+\widetilde{\tau})a_k^*\right].$$









DA Timing Synchronization (3/3)

Thus

$$\begin{aligned}\hat{\tau}_{k+1} &= \hat{\tau}_k + \alpha\mathcal{R}\left[x'(kT+\hat{\tau}_k)a_k^*\right] \\ &\simeq \hat{\tau}_k + \alpha\mathcal{R}\left\{\left[x\left(kT+\frac{T}{2}+\hat{\tau}_k\right) - x\left(kT-\frac{T}{2}+\hat{\tau}_{k-1}\right)\right]a_k^*\right\}.\end{aligned}$$

In the approximate implementation, this algorithm requires to oversample the MF output with 2 samples per symbol interval.
A different algorithm, proposed by Gardner, is the following

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\mathcal{R}\left\{\left[a_{k-1} - a_k\right]x^*\left(kT-\frac{T}{2}+\hat{\tau}_k\right)\right\}.$$

DA Frame Synchronization (1/2)

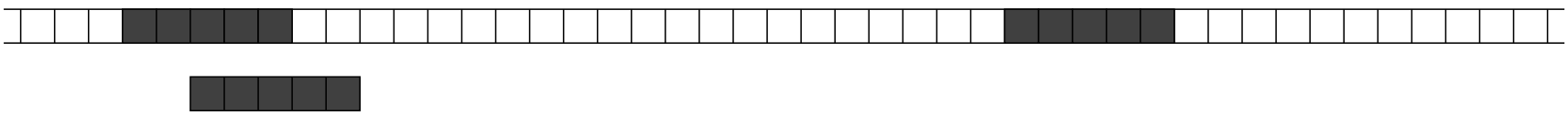
Consider a continuous transmission according to the following observation model

$$r_k = a_{k-k_0} + w_k$$

where k_0 is an integer delay introduced by the channel. Assume that, within the sequence, we have a pattern of L_p consecutive symbols $\{b_\ell\}_{\ell=0}^{L_p-1}$ periodically transmitted every N symbols. The value of k_0 can be found by correlating the last L_p received samples with the known sequence and looking for the position for which we have the maximum possible value:

$$\hat{k}_0 = \underset{k\in\mathcal{I}}{\operatorname{argmax}}\, \mathcal{R}\left\{ \sum_{\ell=k-L_p+1}^{k} r_\ell b^*_{\ell-k+L_p-1} \right\}$$

where $\mathcal{I} = [0, N-1]$. In practice, we can compute $\mathcal{R}\left\{\sum_{\ell=k-L_p+1}^{k} r_\ell b^*_{\ell-k+L_p-1}\right\}$ and compare it with a threshold, declaring that we found the alignment when we are above the threshold.



DA Frame Synchronization (2/2)

- The sequence $\{b_\ell\}_{\ell=0}^{L_p-1}$ must hold some particular properties (it must be designed properly).
- To reduce the false alarm probability, a verification phase could be useful.
- It works well when there is no phase uncertaint. In fact, in this case, when there is alignment we have (assuming M-PSK with $|a_k| = 1$)

$$\mathcal{R}\left\{\sum_{\ell=k-L_p+1}^{k} r_\ell b_{\ell-k+L_p-1}^*\right\} = L_p + \mathcal{R}\Big\{\underbrace{\sum_{\ell=k-L_p+1}^{k} w_\ell b_{\ell-k+L_p-1}^*}_{W}\Big\}.$$

If a time-varying channel phase is present (e.g., there is an uncompensated frequency offset), i.e.,

$$r_k = a_{k-k_0} e^{\jmath\theta_k} + w_k$$

the algorithm cannot work since

$$\sum_{\ell=k-L_p+1}^{k} r_\ell b_{\ell-k+L_p-1}^* = \sum_{\ell=k-L_p+1}^{k} e^{\jmath\theta_\ell} + W$$

DPDI

A possible solution can be the adoption of a strategy based on differential detection $\rightarrow$ Differential Post-Detection Integration (DPDI) algorithm.
The L_p symbols are split into B blocks of L symbols ($L_p = BL$). In each block it is assumed that the channel phase remains constant $\rightarrow$ we can compute the correlation

$$y_i = \sum_{\ell=k-L+1-iL}^{k-iL} r_\ell b^*_{\ell-k+L_p-1} \simeq L e^{\jmath\Theta_i} + W_i\,, \quad i = 0, 1, \ldots, B-1\,.$$

Then

$$y_i y^*_{i-1} = L^2 e^{\jmath\Theta_i} e^{-\jmath\Theta_{i-1}} + \underbrace{L W_i e^{-\jmath\Theta_{i-1}} + L W^*_{i-1} e^{\jmath\Theta_i} + W_i W^*_{i-1}}_{W} \simeq L^2 + W$$

if the channel phase Θ_i can be assumed constant over two consecutive blocks. Hence

$$\hat{k}_0 = \underset{k\in\mathcal{I}}{\text{argmax}} \left| \sum_{i=1}^{B-1} y_i y^*_{i-1} \right|$$

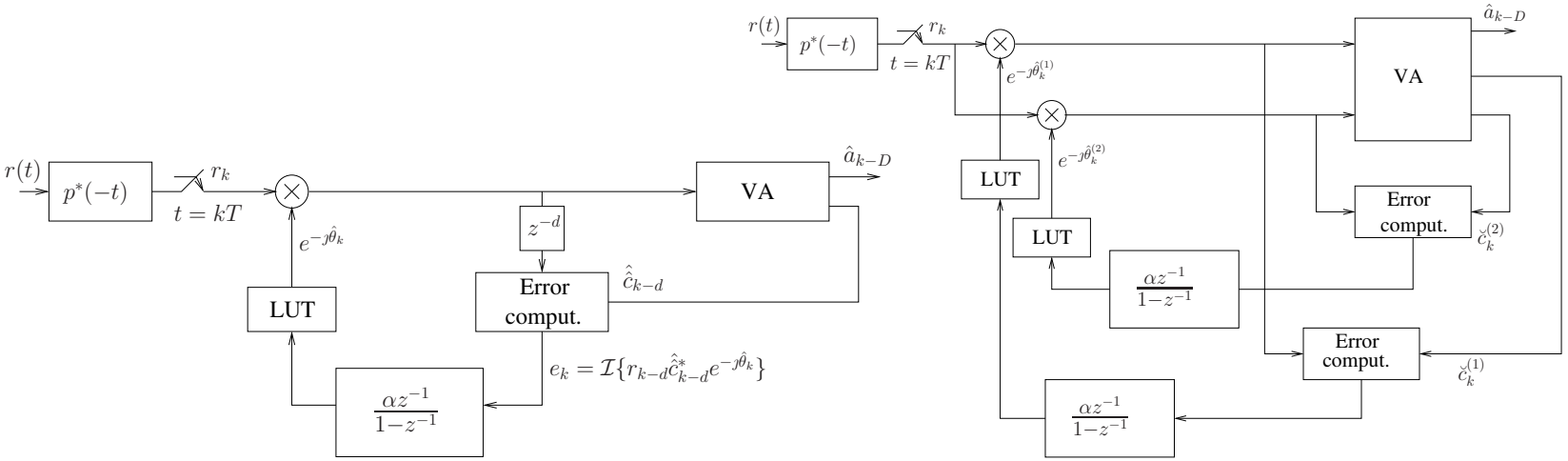
L, the coherent correlation length, depends on the phase coherence time.

DD Estimation and DD-PLL

Once the steady state has been reached and we have reliable decisions, we may switch in DD mode, bus substituting the known data with decisions and track the channel variations:

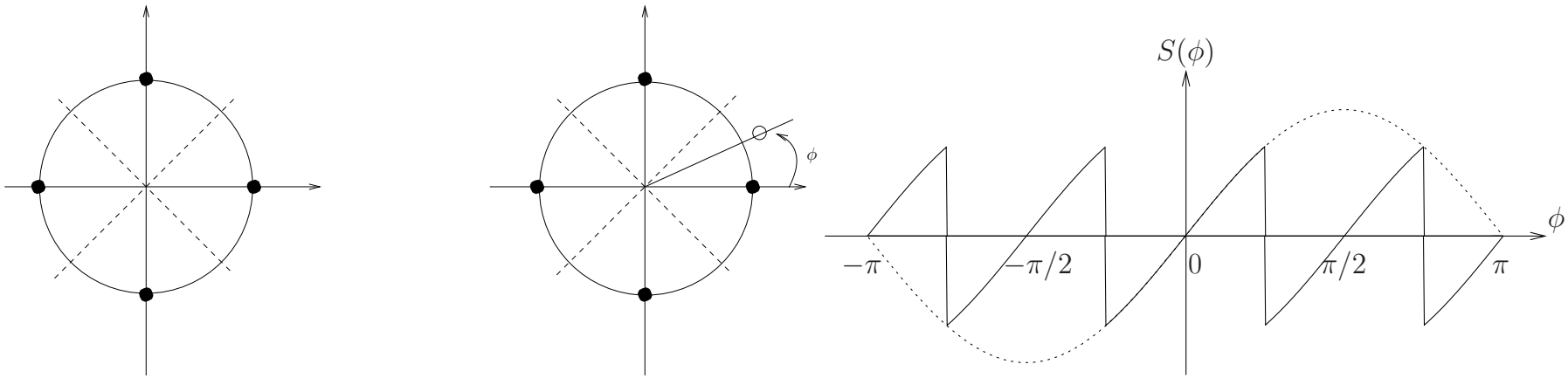
$$\hat{\theta}_{DD} = \arg\max_{\theta} p(\mathbf{r}|\hat{\mathbf{a}}, \theta) \, .$$

Sometimes (e.g., in case of phase estimate) the training phase is not required and we can directly start in DD mode.
When a VA is employed, we have a problem related to the VA decision delay.
Solutions: tentative decisions or per-survivor processing (PSP).



DD-PLL

Note: Modification of the S-curve of a DD-PLL.
A DD-PLL cannot recover the absolute channel phase: ambiguity due to the angle of symmetry of the employed constellation.



Solution: differential encoding.

NDA estimation

In the case of NDA estimation, it is assumed to have no knowledge of data.
Data are also assumed independent and uniformly distributed (i.u.d.). Thus,

$$\hat{\theta}_{NDA} = \underset{\theta}{\text{argmax}}\, p(\mathbf{r}_0^{L-1}|\theta)\,.$$

When we employ an M-ary constellation $\mathcal{A}$,

$$p(\mathbf{r}_0^{L-1}|\theta) = \sum_{a_0 \in \mathcal{A}} \sum_{a_1 \in \mathcal{A}} \cdots \sum_{a_{L-1} \in \mathcal{A}} \left(\frac{1}{M}\right)^L p(\mathbf{r}_0^{L-1}|\mathbf{a}_0^{L-1}, \theta)\,.$$

The computation of $p(\mathbf{r}_0^{L-1}|\theta)$ is often unfeasible in closed form.
Approximations are introduced or other heuristic approaches are adopted.

NDA Phase Estimation (1/2)

Example: Phase estimation.
Considering the channel

$$r_k = a_k e^{\jmath\theta} + w_k$$

and assuming an M-PSK constellation, by introducing proper approximations we obtain the M-th power estimator:

$$\hat{\theta}_{NDA} = \frac{1}{M}\arg\left(\sum_{k=0}^{L-1} r_k^M\right)$$

This OL estimator has also a heuristic interpretation. In the absence of noise, it is clearly

$$r_k^M = a_k^M e^{\jmath M\theta} = e^{\jmath M\theta}\ .$$

Thus, raising the received samples to the M-th power, we are removing the modulation. On the other hand, in this way we can only estimate phase values in the interval $[-\frac{\pi}{M}, \frac{\pi}{M}) \rightarrow$ the use of differential encoding is required.

NDA Phase Estimation (2/2)

The M-th power estimator reaches the CRB for high SNR. An algorithm with better performance is the Viterbi & Viterbi algorithm.
Observing that

$$r_\ell^M = |r_\ell|^M e^{\jmath M \arg(r_\ell)} .$$

The V&V algorithm computes

$$\hat{\theta}_{NDA} = \frac{1}{M} \arg \left(\sum_{\ell=0}^{L-1} F\left(|r_\ell|\right) e^{\jmath M \arg(r_\ell)} \right)$$

where $F(|r_\ell|)$ is a proper function of $|r_\ell|$. When $F(|r_\ell|) = |r_\ell|^M$ we obtain the M-th power estimator. We can have a better performance by choosing $F(|r_\ell|) = |r_\ell|^2$ or $F(|r_\ell|) = 1$ depending on the considered modulation or the operating SNR value.
A CL version of this estimator can be easily devised.

NDA Frequency Estimation (1/3)

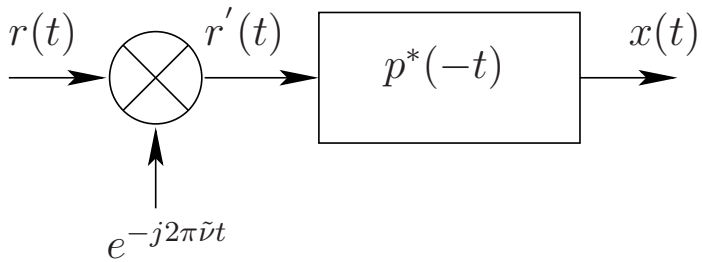
Example: Coarse frequency estimation (with no timing information).
A NDA algorithm must be used for coarse frequency estimation (large initial frequency uncertainty). In this case, we cannot use the discrete-time model working at symbol time. In fact, in case of large frequency errors, we have to compensate for the frequency before matched filtering. We can work on the continuous-time model or on an oversampled model (with sampling time T_s):
Two CL algorithms are usually employed for linear modulations. For both algorithms

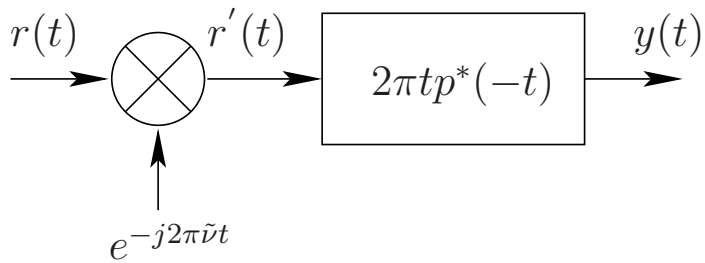
$$\hat{F}[(k+1)T_s] = \hat{F}(kT_s) + \alpha e(kT_s)\,.$$

They differ for expression of the error signal only

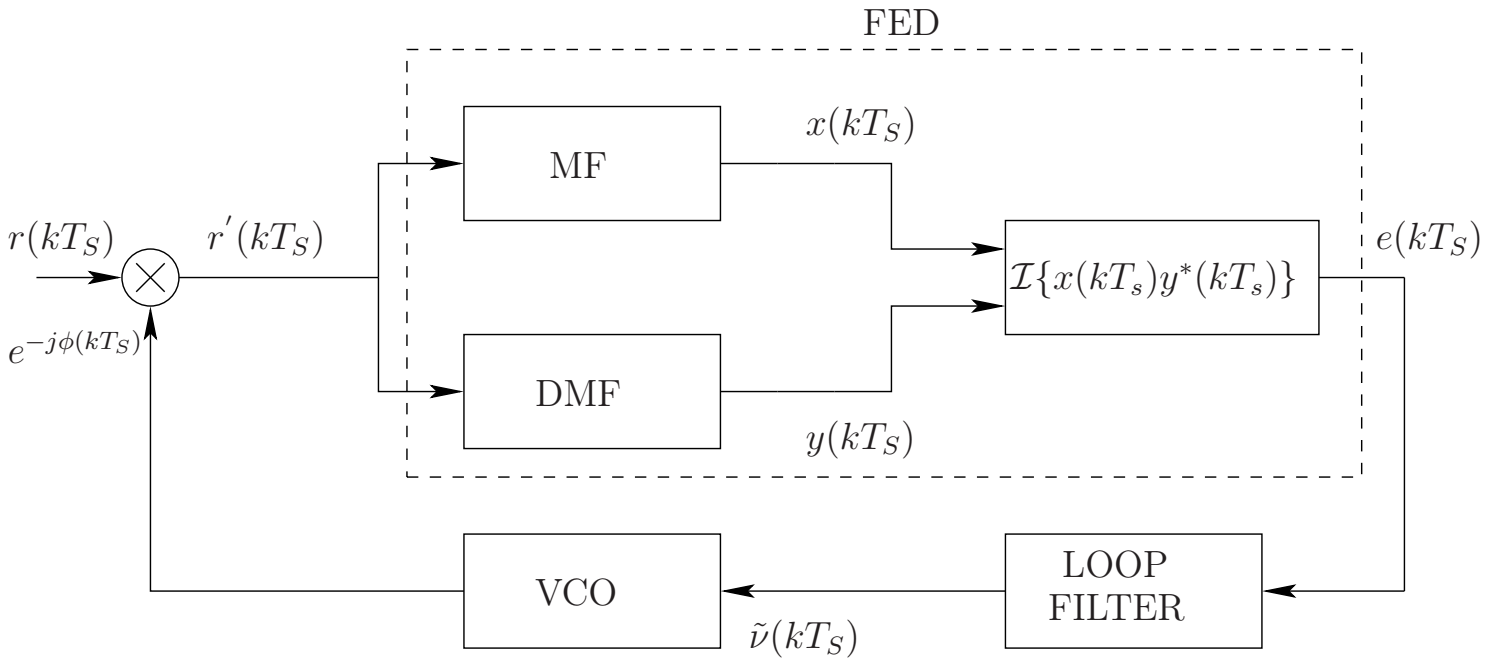
- The first one is derived from the ML criterion:

$$e(kT_s) = \mathcal{I}\{x(kT_s)y^*(kT_s)\}\,.$$





NDA Frequency Estimation (2/3)



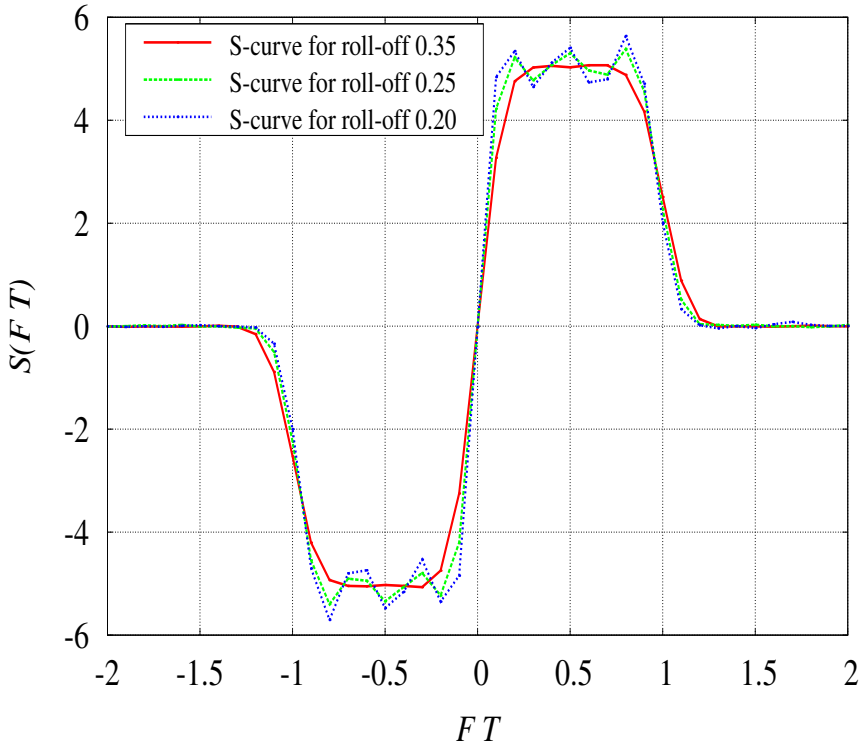
The VCO implements the following recursion

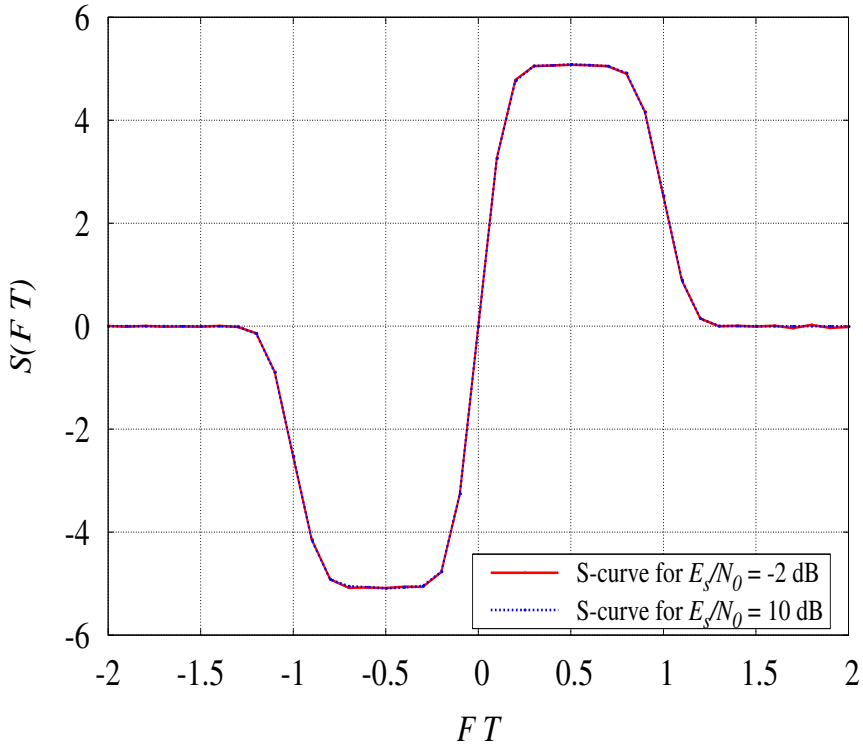
$$\phi[(k+1)T_s] = \phi(kT_s) + 2\pi\hat{F}(kT_s)T_s.$$

- The second one is heuristic (delay and multiply FED):

$$e(kT_s) = \mathcal{I}\left\{r'(kT_s)\, r'^*[(k-1)T_s]\right\} .$$

NDA Frequency Estimation (3/3)





NDA Timing Estimation (1/5)

Timing estimation is often performed in NDA mode since, even in this case, we obtain a very good performance. We will see a few algorithms commonly used in satellite communications.

Example. Remember the DA timing algorithm already discussed:

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \mathcal{R}\left\{\left[x\left(kT + \frac{T}{2} + \hat{\tau}_k\right) - x\left(kT - \frac{T}{2} + \hat{\tau}_{k-1}\right)\right] a_k^*\right\} .$$

An NDA algorithm is obtained by substituting a_k with $x\left(kT + \hat{\tau}_k\right)$. Thus

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \mathcal{R}\left\{\left[x\left(kT + \frac{T}{2} + \hat{\tau}_k\right) - x\left(kT - \frac{T}{2} + \hat{\tau}_{k-1}\right)\right] x^*\left(kT + \hat{\tau}_k\right)\right\} .$$

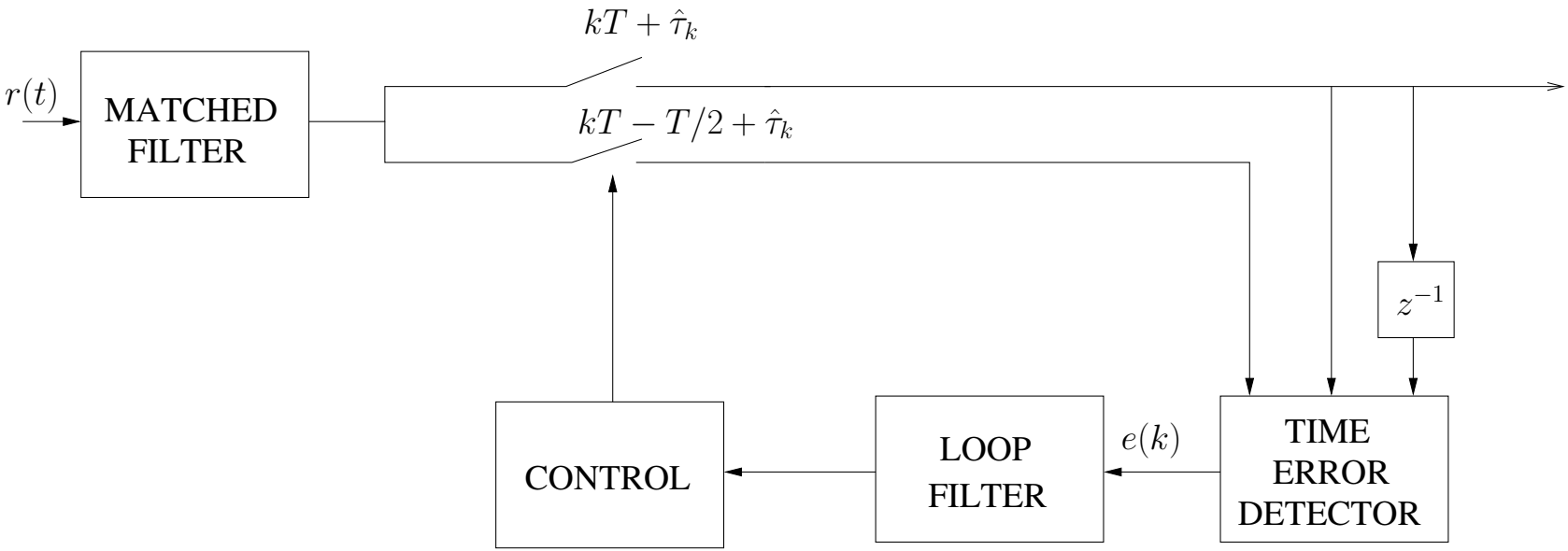
The algorithm is quite robust to uncompensated phase and frequency offsets.

NDA Timing Estimation (2/5)

Example. NDA Gardner algorithm.

It is a heuristic modification of the DA Gardner algorithm (obtained substituting the symbols with the samples):

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \mathcal{R}\left\{ \left[x\left(kT - T + \hat{\tau}_{k-1}\right) - x\left(kT + \hat{\tau}_k\right) \right] x^*\left(kT - \frac{T}{2} + \hat{\tau}_k \right) \right\} .$$



NDA Timing Estimation (3/5)

Example. Modified NDA Gardner algorithm. A modified version of the Gardner algorithm exhibits a better performance (a performance closer to the CRB) especially in the presence of NL distortions. Let us decompose $x(t)$ in polar coordinates

$$x(t) = \rho(t) e^{\jmath \beta(t)}$$

where $\rho(t) = |x(t)|$ and $\beta(t) = \arg[x(t)]$. The error signal of the modified Gardner algorithm has expression

$$e(k) = \mathcal{R}\left\{ \left[\rho^{\mu}\left(kT - T + \hat{\tau}_{k-1}\right) e^{\jmath\beta\left(kT - T + \hat{\tau}_{k-1}\right)} - \rho^{\mu}\left(kT + \hat{\tau}_{k}\right) e^{\jmath\beta\left(kT + \hat{\tau}_{k}\right)} \right] x^{*}\left(kT - \frac{T}{2} + \hat{\tau}_{k}\right) \right\}.$$

Hence, for $\mu = 1$ it reduces to the Gardner algorithm. A better performance is obtained by choosing $\mu = 0$.

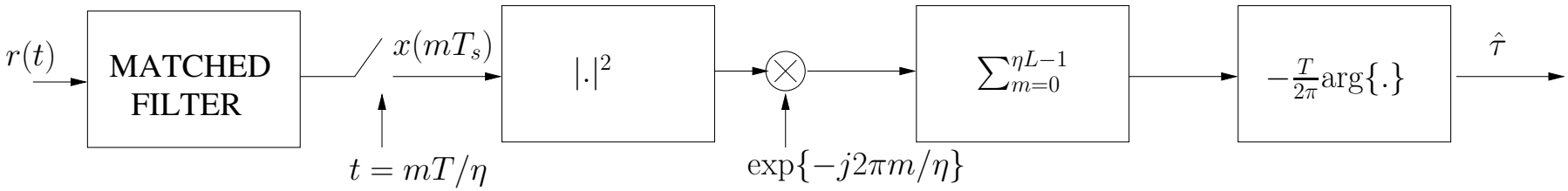
NDA Timing Estimation (4/5)

Example. Oerder & Meyr algorithm (O&M).
Another popular algorithm for timing synchronization is the one proposed by O&M. As the Gardner algorithm, it is a heuristic and employs the oversampled output of the matched filter (typically the oversampling factor is $\eta = 4$). However, it is OL and is well suited for burst-mode transmissions.
It exploits the presence in the spectrum of the squared output of the MF of a frequency component at $f = 1/T$. Then, it uses the delay property of Fourier transforms, i.e.,

$$x(t-\tau) \longleftrightarrow X(f)\, e^{-\jmath 2\pi \tau f}$$

to relate the phase of the spectrum to the timing delay.

NDA Timing Estimation (5/5)



Hence, the estimator is defined as

$$\hat{\tau} = -\frac{T}{2\pi}\arg\left\{\sum_{m=0}^{\eta L-1}|x(mT_s)|^2 e^{-j2\pi m/\eta}\right\}$$

where L is the integration window (design parameter).
In order to reduce the noise enhancement due to the square law, different nonlinear functions can be used. The most popular version of the O&M algorithm employs the absolute value, i.e.,

$$\hat{\tau} = -\frac{T}{2\pi}\arg\left\{\sum_{m=0}^{\eta L-1}|x(mT_s)| e^{-j2\pi m/\eta}\right\}.$$

SDD Estimation (1/4)

In modern communication systems, based on the use of powerful coding schemes to be decoded iteratively, we typically have more detection/decoding iterations. At each iteration, the decoder provides new updates for the probabilities of coded symbols. We will denote by $\{\hat{P}(c_k)\}_{c_k \in \mathcal{C}}$, where $\mathcal{C}$ is the constellation of coded symbols, the estimates of the a priori probabilities of symbol c_k available at a given iteration (soft decision). Using them, it is possible to update the estimates of the channel parameters using an NDA algorithm, assuming that the symbols are still independent (due to the presence of an interleaver between detector and decoder) but not equally likely since they are assumed distributed according to the probablities $\{\hat{P}(c_k)\}_{c_k \in \mathcal{C}}$.
The resulting SDD algorithm is defined as

$$\hat{\theta}_{SDD} = \arg\max_{\theta} \hat{p}(\mathbf{r}_0^{L-1}|\theta)$$

where

$$\hat{p}(\mathbf{r}_0^{L-1}|\theta) = \sum_{c_0 \in \mathcal{C}} \hat{P}(c_0) \sum_{c_1 \in \mathcal{C}} \hat{P}(c_1) \cdots \sum_{c_{L-1} \in \mathcal{C}} \hat{P}(c_{L-1}) p(\mathbf{r}_0^{L-1}|\mathbf{c}_0^{L-1}, \theta)$$

represents the estimate of the pdf of the received samples given the parameter and computed based on soft decisions.
At the beginning, and SDD algorithm thus starts in NDA mode and, at the end, when decisions become reliable, it becomes a DD estimator.

SDD Estimation (2/4)

Example: SDD phase estimation.

It is

$$\hat{p}(\mathbf{r}_0^{L-1}|\theta) = \prod_{\ell=0}^{L-1} \hat{p}(r_\ell|\theta)$$

where

$$\hat{p}(r_\ell|\theta) = \sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell) p(r_\ell|c_\ell, \theta) \tag{13}$$

having defined

$$p(r_\ell|c_\ell, \theta) = \frac{1}{2\pi N_0} \exp\left\{ -\frac{1}{2N_0} \left| r_\ell - c_\ell e^{\jmath\theta} \right|^2 \right\} .$$

The SDD phase estimate is thus

$$\begin{aligned} \hat{\theta}_{SDD} &= \underset{\theta}{\text{argmax}}\, \hat{p}(\mathbf{r}_0^{L-1}|\theta) = \underset{\theta}{\text{argmax}}\, \ln \hat{p}(\mathbf{r}_0^{L-1}|\theta) = \underset{\theta}{\text{argmax}} \sum_{\ell=0}^{L-1} \ln \hat{p}(r_\ell|\theta) \\ &= \underset{\theta}{\text{argmax}} \sum_{\ell=0}^{L-1} \ln \left\{ \sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell) \exp\left[-\frac{1}{2N_0} \left| r_\ell - c_\ell e^{\jmath\theta} \right|^2 \right] \right\} . \end{aligned}$$

SDD Estimation (3/4)

A closed form expression of $\hat{\theta}_{SDD}$ is not available. A commonly used approximation is the following. The pdf (13) is a Gaussian mixture. We can approximate it with a Gaussian pdf with the same mean and the same variance (approximation which minimizes the divergence).

$$\begin{aligned} E\{r_\ell|\theta\} &= e^{\jmath\theta}E\{c_\ell\} = e^{\jmath\theta}\gamma_\ell \\ \text{var}\{r_\ell|\theta\} &= E\{|r_\ell|^2 \mid \theta\} - |\gamma_\ell|^2 = E\{|c_\ell|^2\} + 2N_0 - |\gamma_\ell|^2 \\ &= \beta_\ell + 2N_0 - |\gamma_\ell|^2 \end{aligned}$$

having defined

$$\gamma_\ell = E\{c_\ell\} = \sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell) c_\ell \qquad \beta_\ell = E\{|c_\ell|^2\} = \sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell)\,|c_\ell|^2 \ .$$

We thus use the approximation

$$\begin{aligned} \hat{p}(r_\ell|\theta) &= \frac{1}{\pi\left[\beta_\ell + 2N_0 - |\gamma_\ell|^2\right]} \exp\left\{-\frac{\left|r_\ell - \gamma_\ell e^{\jmath\theta}\right|^2}{\beta_\ell + 2N_0 - |\gamma_\ell|^2}\right\} \\ &= \frac{1}{\pi\left[\beta_\ell + 2N_0 - |\gamma_\ell|^2\right]} \exp\left\{-\frac{|r_\ell|^2 + |\gamma_\ell|^2}{\beta_\ell + 2N_0 - |\gamma_\ell|^2}\right\} \exp\left\{\frac{2\mathcal{R}\left[r_\ell \gamma_\ell^* e^{-\jmath\theta}\right]}{\beta_\ell + 2N_0 - |\gamma_\ell|^2}\right\} . \end{aligned}$$

SDD Estimation (4/4)

It thus results

$$\hat{\theta}_{SDD} = \underset{\theta}{\text{argmax}} \sum_{\ell=0}^{L-1} \ln \hat{p}(r_\ell|\theta) = \arg\left[\sum_{\ell=0}^{L-1} \frac{r_\ell \gamma_\ell^*}{\beta_\ell + 2N_0 - |\gamma_\ell|^2}\right].$$

In the case of a CL implementation

$$\begin{aligned}\hat{\theta}_{\ell+1} &= \hat{\theta}_\ell + \alpha \left.\frac{\mathrm{d} \ln \hat{p}(r_\ell|\theta)}{\mathrm{d}\theta}\right|_{\theta=\hat{\theta}_\ell} = \hat{\theta}_\ell + \alpha \left.\frac{1}{\hat{p}(r_\ell|\theta)} \frac{\mathrm{d}\hat{p}(r_\ell|\theta)}{\mathrm{d}\theta}\right|_{\theta=\hat{\theta}_\ell} \\ &= \hat{\theta}_\ell + \alpha \frac{\sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell) \mathcal{I}\left[r_\ell e^{-\jmath\hat{\theta}_\ell} c_\ell^*\right] \exp\left\{\frac{1}{N_0}\mathcal{R}\left[r_\ell e^{-\jmath\hat{\theta}_\ell} c_\ell^*\right]\right\}}{\sum_{c_\ell \in \mathcal{C}} \hat{P}(c_\ell) \exp\left\{\frac{1}{N_0}\mathcal{R}\left[r_\ell e^{-\jmath\hat{\theta}_\ell} c_\ell^*\right]\right\}}\end{aligned}$$

or, using the minimum divergence approximation,

$$\hat{\theta}_{\ell+1} = \hat{\theta}_\ell + \alpha \mathcal{I}\left[r_\ell e^{-\jmath\hat{\theta}_\ell} \frac{\gamma_\ell^*}{\beta_\ell + 2N_0 - |\gamma_\ell|^2}\right].$$

Bayesian SDD Estimation Using FGs (1/14)

Beyesian estimation means that we have some a priori information on the unknown parameters, which is exploited when performing the estimation.
Example: Unknown channel phase modeled as a Wiener process. In this case:

$$r_k = c_k e^{\jmath\theta_k} + w_k \,. \qquad \theta_k = \theta_{k-1} + \Delta_k$$

where $\{\Delta_k\}$ are i.i.d Gaussian rvs with mean zero and variance σ^2_Δ. Hence,

$$p(\boldsymbol{\theta}) = \prod_k p(\theta_k|\theta_{k-1}, \theta_{k-2}, \ldots, \theta_0) = \prod_k p(\theta_k|\theta_{k-1})$$

where

$$p(\theta_k|\theta_{k-1}) = \frac{1}{\sqrt{2\pi\sigma^2_\Delta}} \exp\left\{-\frac{(\theta_k - \theta_{k-1})^2}{2\sigma^2_\Delta}\right\}$$

SDD estimation can be performed by using FGs, through the following steps:

- Variable nodes representing the channel parameters are explicitly introduced in the FG by considering the joint distribution of symbols and unknown parameters and the corresponding FG.
- For the computation of the marginal pmfs $P(c_k|\mathbf{r})$, the average over channel parameters of the joint conditional distribution of $\mathbf{c}$ and $\boldsymbol{\theta}$ is now performed by the SP algorithm.

Bayesian SDD Estimation Using FGs (2/14)

Problem. We need to handle the presence in the graph of continuous rvs (representing the channel parameters)
Solution. Use of canonical distributions
Example: Detection and decoding on a channel with Wiener phase noise.
The joint a posteriori distribution of symbols and unknown parameters may be expressed as (Wiener model)

$$p(\mathbf{c},\boldsymbol{\theta}|\mathbf{r}) \propto p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})\chi(\mathbf{c})p(\boldsymbol{\theta}) = \chi(\mathbf{c})p(\theta_0)\prod_{k=0}^{N-1} p(r_k|c_k,\theta_k)\prod_{k=1}^{N-1} p(\theta_k|\theta_{k-1})$$

$$\propto \chi(\mathbf{c})p(\theta_0)\prod_{k=0}^{N-1} f_k(c_k,\theta_k)\prod_{k=1}^{N-1} p(\theta_k|\theta_{k-1})$$

where

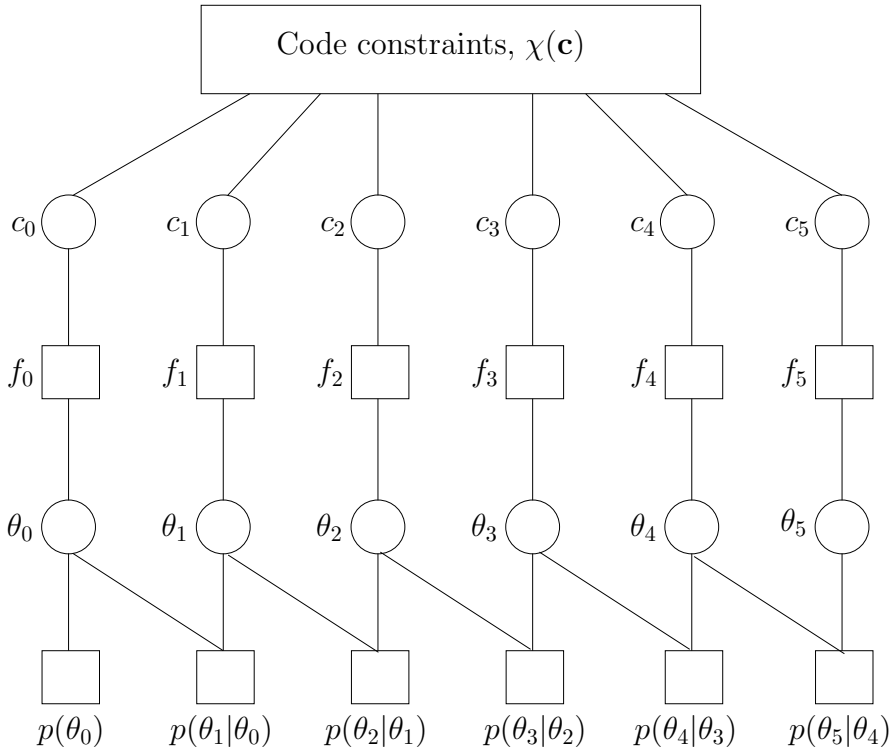
$$f_k(c_k,\theta_k) = \exp\left\{-\frac{1}{2\sigma^2}|r_k - c_k e^{j\theta_k}|^2\right\}$$

$$p(\theta_0) = \frac{1}{2\pi} \quad , \quad \theta_0 \in [0,2\pi)$$

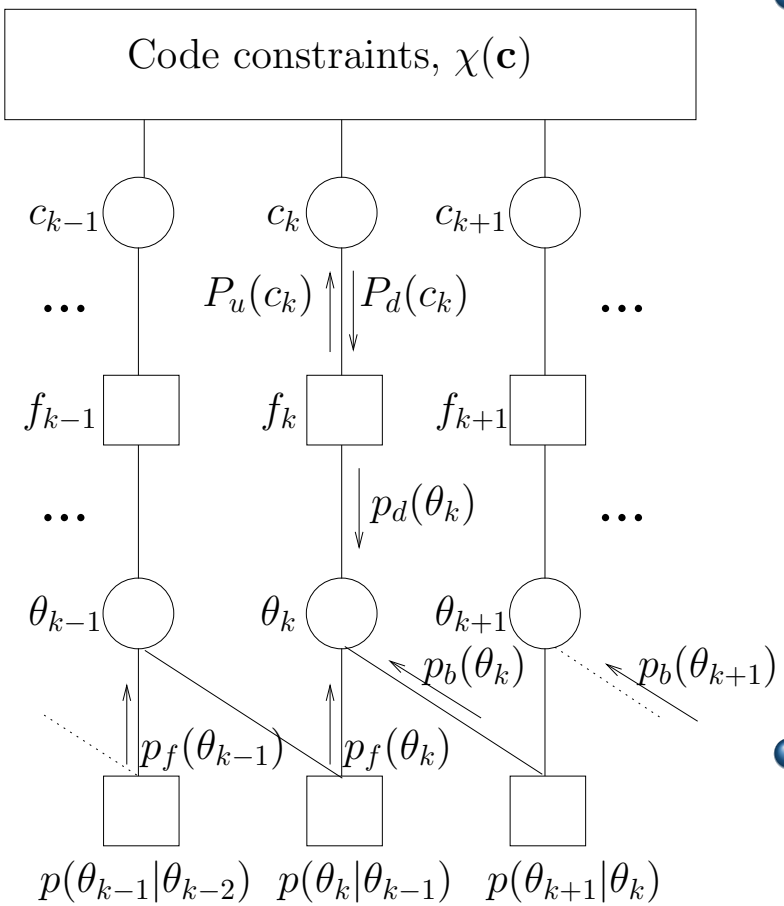
$$p(\theta_k|\theta_{k-1}) = p_\Delta(\theta_k - \theta_{k-1}) = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} e^{-\frac{(\theta_k-\theta_{k-1})^2}{2\sigma_\Delta^2}}$$

Bayesian SDD Estimation Using FGs (3/14)

Corresponding FG



Bayesian SDD Estimation Using FGs (4/14)



- Omitting the esplicit reference to the current iteration:

$$p_d(\theta_k) = \sum_{c \in \mathcal{C}} P_d(c_k = c) f_k(c_k = c, \theta_k)$$

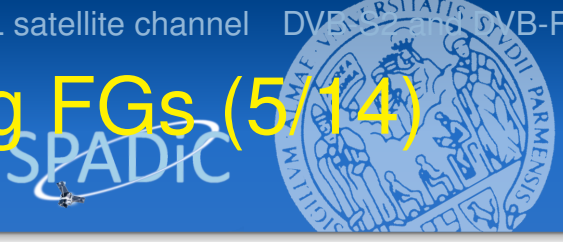
$$p_f(\theta_k) = \int_0^{2\pi} p_d(\theta_{k-1}) p_f(\theta_{k-1}) p(\theta_k|\theta_{k-1}) \, \mathrm{d}\theta_{k-1}$$

$$p_b(\theta_k) = \int_0^{2\pi} p_d(\theta_{k+1}) p_b(\theta_{k+1}) p(\theta_{k+1}|\theta_k) \, \mathrm{d}\theta_{k+1}$$

$$P_u(c_k) = \int_0^{2\pi} p_f(\theta_k) p_b(\theta_k) f_k(c_k, \theta_k) \, \mathrm{d}\theta_k$$

- The optimal SP algorithm is unfeasible

Bayesian SDD Estimation Using FGs (5/14)



- We represent the pdfs, which are the messages sent or received by nodes representing the channel parameters, with given canonical pdfs, described by some parameters
- This representation can be exact or, more often, involve some approximate assumptions
- As an example, we could assume that these messages are Gaussian pdfs which can be completely specified by their mean and variance
- Hence, the SP algorithm has just to forward the parameters of this distribution
- Quantization-based algorithm and Tikhonov algorithm

Bayesian SDD Estimation Using FGs (6/14)

- If we quantize the channel parameters, we may consider the factor graph representing the joint APP $P(\mathbf{c}, \boldsymbol{\theta}|\mathbf{r})$
- The application of the SP algorithm to this factor graph allows us to compute the desired marginal APPs $P(c_k|\mathbf{r})$ and the collateral ones $P(\theta_k|\mathbf{r})$
- From a point of view of the involved approximation, in this case we have the quantization of in general continuous channel parameters and their statistics
- This approach becomes optimal (in the sense that it approaches the performance of the exact SP algorithm) for a sufficiently large number of quantization levels, at the expenses of an increased computational complexity
- As an example, for M-PSK signals, $L = 8M$ quantization levels are sufficient

Bayesian SDD Estimation Using FGs (7/14)

- In the case of the Wiener model

$$\begin{aligned} P(\mathbf{c},\boldsymbol{\theta}|\mathbf{r}) &\propto \chi(\mathbf{c})P(\boldsymbol{\theta})p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \\ &\propto \chi(\mathbf{c})P(\theta_0)\prod_{k=1}^{N-1}P(\theta_k|\theta_{k-1})\prod_{k=0}^{N-1}f_k(c_k,\theta_k) \end{aligned}$$

  where

$$f_k(c_k,\theta_k)=\exp\left\{-\frac{1}{2\sigma^2}|r_k-c_k e^{j\theta_k}|^2\right\}$$

- The pmf $P(\theta_k|\theta_{k-1})$ is related to the quantized distribution of the increment Δ_k:

$$P(\theta_k|\theta_{k-1})=P_\Delta(\theta_k-\theta_{k-1})$$

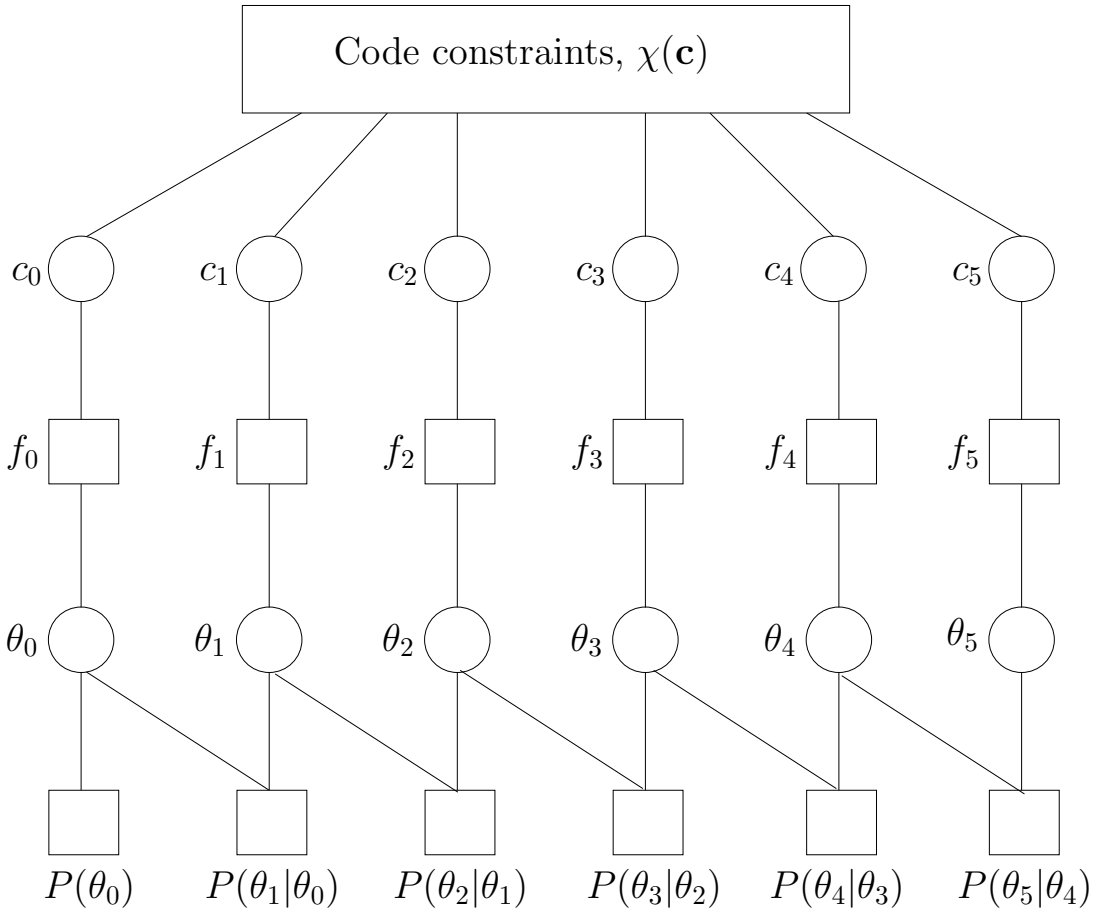
- As an example, for $\sigma_\Delta \ll 1$

$$P_\Delta(\Delta_k)=\begin{cases} 1-\sigma_\Delta^2\left(\frac{L}{2\pi}\right)^2 & \text{for } \Delta_k=0 \\ \frac{\sigma_\Delta^2}{2}\left(\frac{L}{2\pi}\right)^2 & \text{for } \Delta_k=\pm\frac{2\pi}{L} \end{cases}$$

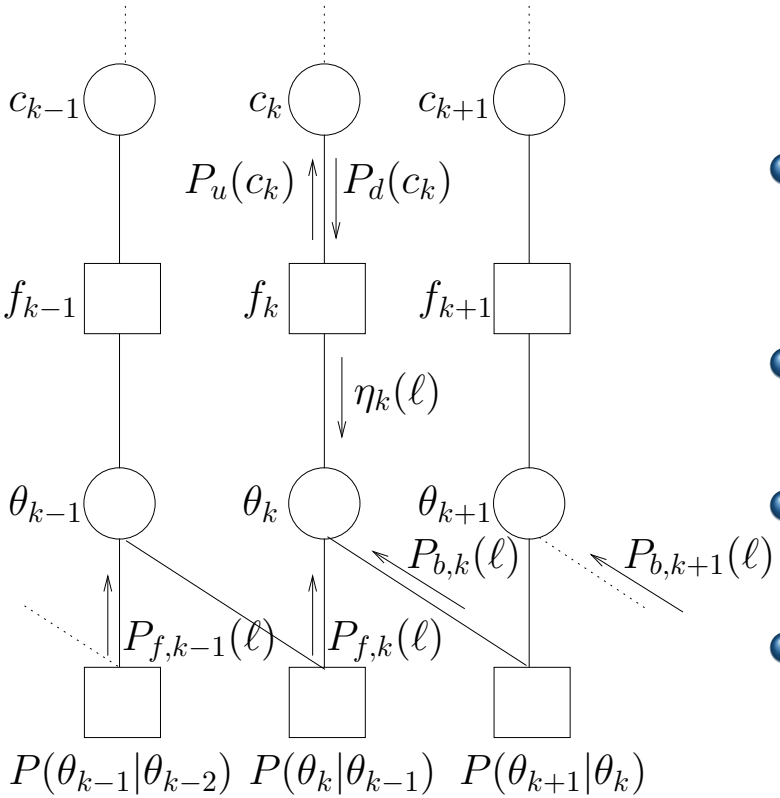
  (discrete random walk approximation)

Bayesian SDD Estimation Using FGs (8/14)

Corresponding FG



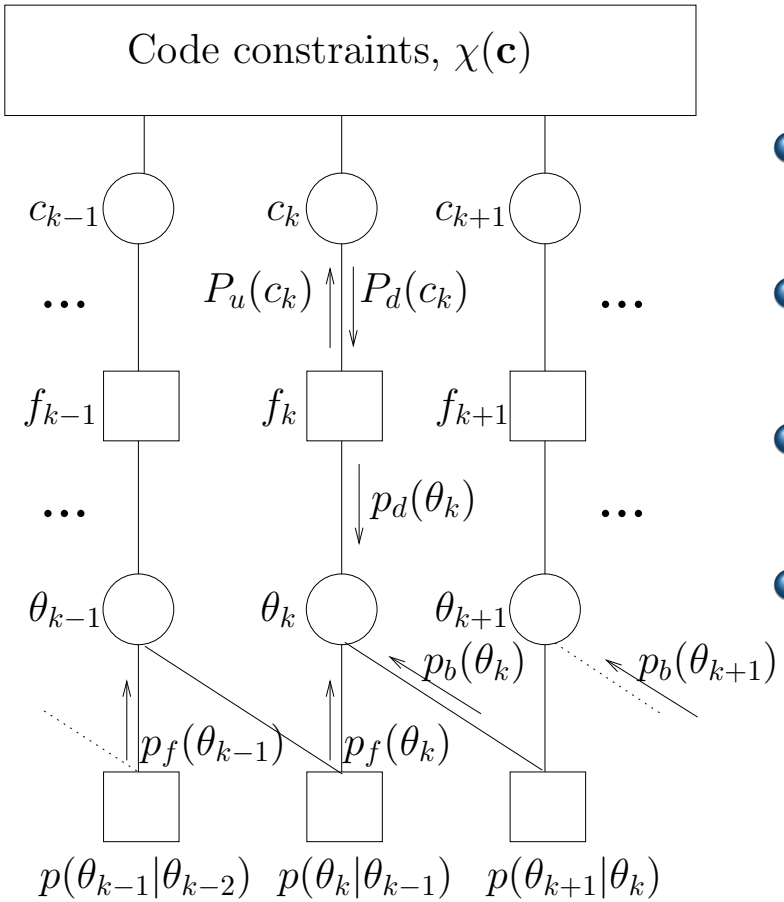
Bayesian SDD Estimation Using FGs (9/14)



- Omitting the esplicit reference to the current iteration
- $\eta_k(\ell) = \sum_{c\in\mathcal{C}} f_k(c_k = c, 2\pi\ell/L) P_d(c_k = c)$
  $\ell = 0, 1, \ldots, L-1$
- $P_{f,k}(\ell) = \sum_{m=0}^{L-1} P_{f,k-1}(m)\eta_{k-1}(m) P_\Delta\left(2\pi\frac{\ell-m}{L}\right)$
- $P_{b,k}(\ell) = \sum_{m=0}^{L-1} P_{b,k+1}(m)\eta_{k+1}(m) P_\Delta\left(2\pi\frac{\ell-m}{L}\right)$
- $P_u(c_k) = \sum_{\ell=0}^{L-1} P_{f,k}(\ell) P_{b,k}(\ell) f_k(c_k, 2\pi\ell/L)$

Bayesian SDD Estimation Using FGs (10/14)

- Is it possible to substantially reduce the complexity?



- $p_d(\theta_k) = \sum_{c\in\mathcal{C}} P_d(c_k = c) f_k(c_k = c, \theta_k)$
- $p_f(\theta_k) = \int_0^{2\pi} p_d(\theta_{k-1}) p_f(\theta_{k-1}) p(\theta_k|\theta_{k-1})\, d\theta_{k-1}$
- $p_b(\theta_k) = \int_0^{2\pi} p_d(\theta_{k+1}) p_b(\theta_{k+1}) p(\theta_{k+1}|\theta_k)\, d\theta_{k+1}$
- $P_u(c_k) = \int_0^{2\pi} p_f(\theta_k) p_b(\theta_k) f_k(c_k, \theta_k)\, d\theta_k$

Bayesian SDD Estimation Using FGs (11/14)

- If the messages $P_d(c_k)$ were the exact a posteriori probabilities of the code symbols, it would be $p_d(\theta_k) = p(r_k|\theta_k)$
- The pdf $p_d(\theta_k)$ is a linear combination of Gaussian pdf
- This pdf will be approximated with a single Gaussian pdf with the same mean and variance (approximation based on the first and second moment matching)
- From direct calculation

$$E\{r_k|\theta_k\} = \gamma_k e^{j\theta_k} \quad , \quad var\{r_k|\theta_k\} = 2\sigma^2 + \beta_k - |\gamma_k|^2$$

where

$$\gamma_k = \sum_{c\in\mathcal{C}} cP_d(c_k = c) \quad , \quad \beta_k = \sum_{c\in\mathcal{C}} |c|^2 P_d(c_k = c)$$

Hence

$$p_d(\theta_k) \simeq \exp\left\{-\frac{|r_k - \gamma_k e^{j\theta_k}|^2}{2\sigma^2 + \beta_k - |\gamma_k|^2}\right\} \propto \exp\left\{2\frac{\mathcal{R}[r_k\gamma_k^* e^{-j\theta_k}]}{2\sigma^2 + \beta_k - |\gamma_k|^2}\right\}$$

Bayesian SDD Estimation Using FGs (12/14)

- Substituting this approximation in the recursive integral equations for $p_f(\theta_k)$ and $p_b(\theta_k)$, we find that these pdfs may be expressed as (Tikhonov distribution)

$$p_f(\theta_k) \propto \exp\{\mathcal{R}[a_{f,k}e^{-j\theta_k}]\} \quad , \quad p_b(\theta_k) \propto \exp\{\mathcal{R}[a_{b,k}e^{-j\theta_k}]\}$$

and

$$a_{f,k} = \frac{a_{f,k-1} + 2\frac{r_{k-1}\gamma_{k-1}^*}{2\sigma^2+\beta_{k-1}-|\gamma_{k-1}|^2}}{1+\sigma_\Delta^2\left|a_{f,k-1} + 2\frac{r_{k-1}\gamma_{k-1}^*}{2\sigma^2+\beta_{k-1}-|\gamma_{k-1}|^2}\right|}$$

$$a_{b,k} = \frac{a_{b,k+1} + 2\frac{r_{k+1}\gamma_{k+1}^*}{2\sigma^2+\beta_{k+1}-|\gamma_{k+1}|^2}}{1+\sigma_\Delta^2\left|a_{b,k+1} + 2\frac{r_{k+1}\gamma_{k+1}^*}{2\sigma^2+\beta_{k+1}-|\gamma_{k+1}|^2}\right|}$$

- The solution of the integral equations is exact for $\sigma_\Delta = 0$ and involves a very good approximation in the case of a time-varying channel phase $\Rightarrow$ in practice, the only approximation is that on $p_d(\theta_k)$

Bayesian SDD Estimation Using FGs (13/14)

- Finally,

$$P_u(c_k) \propto \exp\left\{-\frac{|c_k|^2}{2\sigma^2}\right\} \mathrm{I}_0\left(\left|a_{f,k} + a_{b,k} + \frac{r_k c_k^*}{\sigma^2}\right|\right)$$

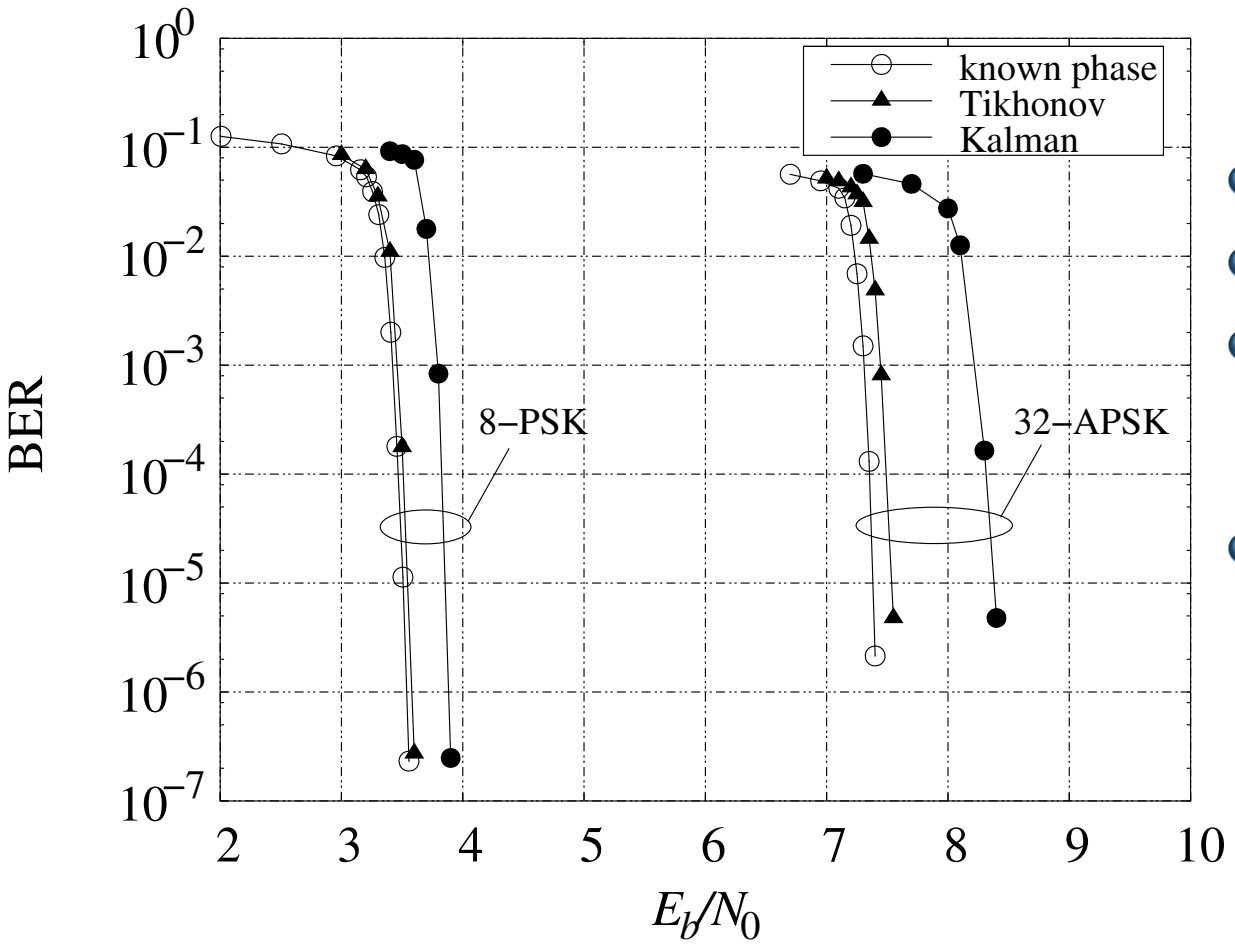
Hence

$$P_d(c_k) \quad \Rightarrow \quad (\gamma_k, \beta_k) \quad \text{(mean and mean square value)}$$

$$(\gamma_k, \beta_k) \quad \Rightarrow \quad a_{f,k}, a_{b,k} \quad \text{(forw. and back. recursions)}$$

$$a_{f,k}, a_{b,k} \quad \Rightarrow \quad P_u(c_k) \quad \text{(probability update)}$$

Bayesian SDD Estimation Using FGs (14/14)



- DVB-S2 phase model
- DVB-S2 LDPC codes
- Maximum of 50 iterations of the SP algorithm on the overall graph
- Pilot distribution of the DVB-S2 standard

Sufficient Statistics (1/2)

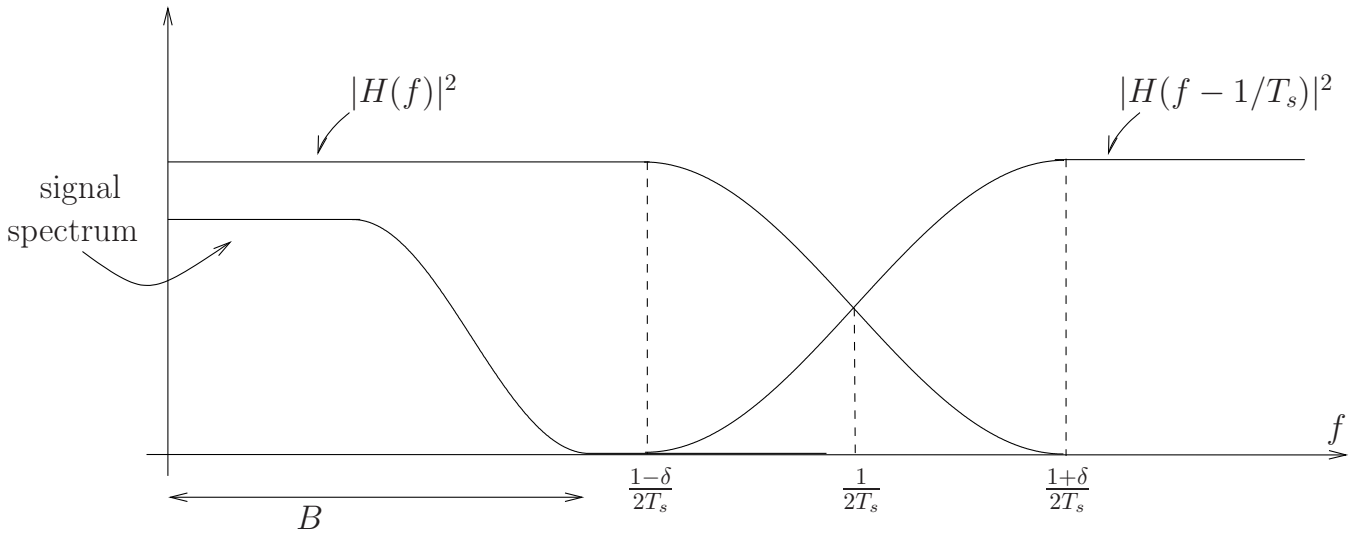
We assumed that one sample per symbol is sufficient to obtain a sufficient statistic. This could be not true when the unknown channel is rapidly time-varying or there is a NL element in the transmission chain.
Let's assume that the signal, after the channel, is bandlimited with banwidth B. In addition, we have AWGN. Let's suppose to filter the complex envelope $r(t)$ with a filter having bandwidth $B_v \geq B$ and frequency response

$$H_{BL}(f) = \begin{cases} 1 & \text{for } |f| \leq B_v \\ 0 & \text{for } |f| > B_v \end{cases}$$

obtaining the signal $r_{BL}(t)$. The useful component of the received signal remains unchanged since the filter has effect on the noise only. The transformation is reversible. Thus, $r_{BL}(t)$ is still a sufficient statistic. Having limited bandwidth, it can be sampled with $f_s = 1/T_s = 2B_v$ and the resulting samples represent a (in general oversampled) sufficient statistic. The additive noise afflicting the samples are white.

Sufficient Statistics (2/2)

The front-end filter $H_{BL}(f)$ is ideal. However, we can easily find an equivalent sufficient statistic by using the filter in the figure, with vestigial symmetry around $1/(2T_s)$

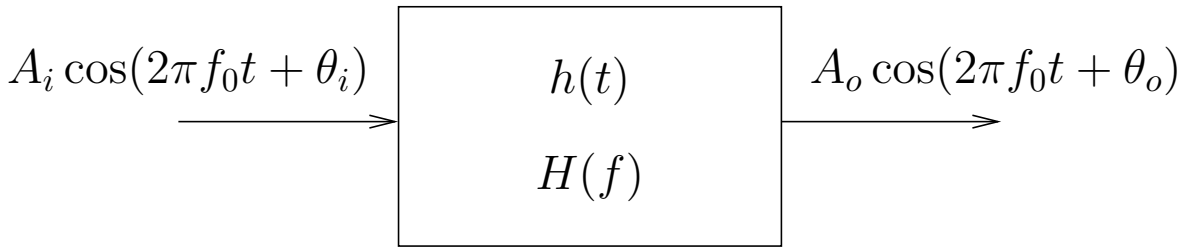


In fact, the effect on the signal is the same and the new filter still gives white noise samples. An example of filter $H(f)$ which satisfies the above mentioned condition is an RRC filter with roll-off δ, bandwidth $\frac{1+\delta}{2T_s}$ and such that $\frac{1-\delta}{2T_s} > B$.

Outline

NL amplifier (1/3)

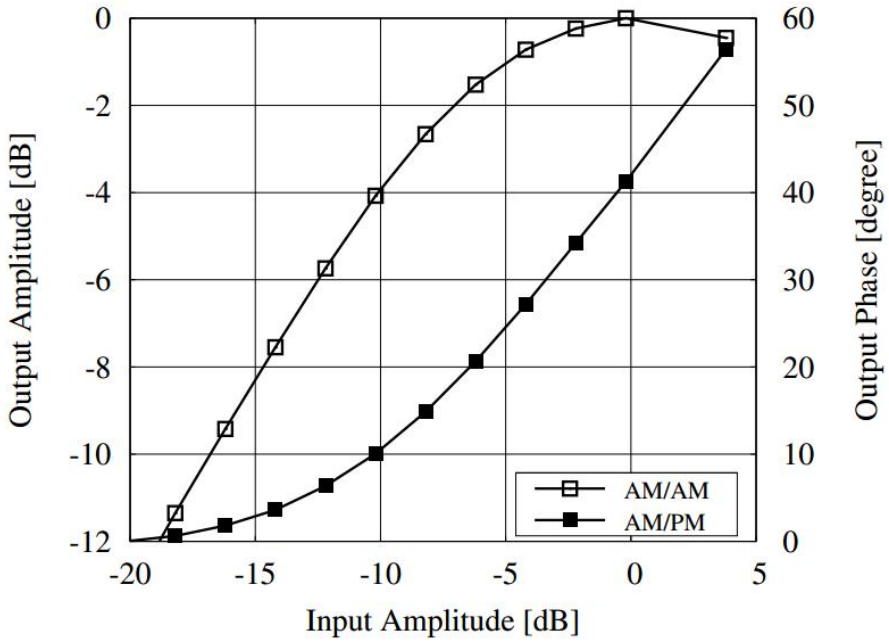
A cosine is an eigenfunction of any linear system:



with $A_o = A_i|H(f_0)|$ and $\theta_o = \theta_i + \arg[H(f_0)]$. Thus A_o vs A_i (the AM/AM characteristic) is a straight line (with slope $|H(f_0)|$) whereas $\theta_o - \theta_i$ vs A_i (the AM/PM characteristic) is a constant.
In addition, a linear amplifier has $H(f)$ constant over a given bandwidth (possibly very large).
HPA amplifiers (e.g., TWTA used onboard of satellites) have a large bandwidth but a non linear behavior,

NL amplifier (2/3)

The HPA can be defined through its AM/AM and AM/PM characteristics, describing the amplitude and phase distortions caused on the signal at its input



The signal at the output of the HPA $y(t)$, is function of the signal $x(t)$ at the input as

$$y(t) = g_{AM}(|x(t)|)\exp\{j(\arg[x(t)] + g_{PM}(|x(t)|))\}$$

where g_{AM} and g_{PM} are the AM/AM and AM/PM characteristics.

NL amplifier (3/3)

In communication systems, nonlinear devices can cause significant performance degradations.

- Nonlinear channel compensation techniques can be essentially classified in techniques applied at the transmitter or at the receiver side.
- Precompensation techniques at the transmitter try to mitigate the nonlinear effects through analog signal predistortion or data predistortion. DVB-S2 systems adopt advanced data predistortion methods to overcome the effect of transponder impairments.
- Techniques at the receiver are in general advanced detection schemes.

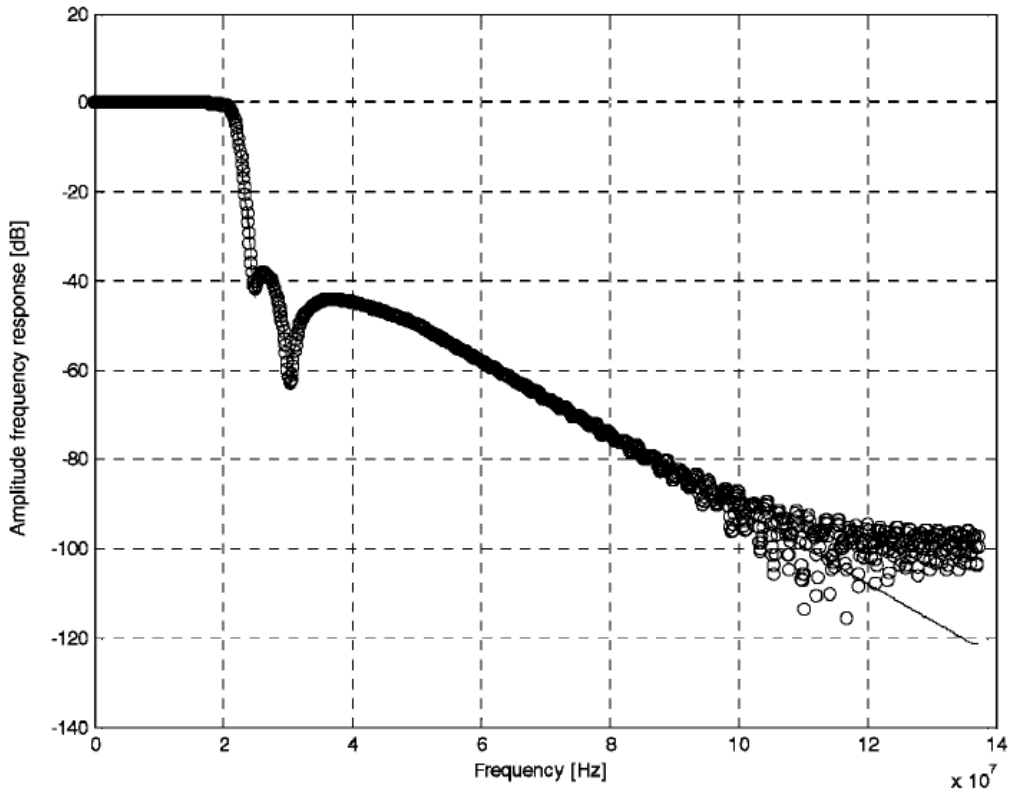
Satellite transponder

Transponder model

- The IMUX filter removes the adjacent channels.
- The OMUX filter reduces the out-of-band power due to the spectral regrowth after nonlinear amplification.
- The HPA is a nonlinear device which can be assumed memoryless.

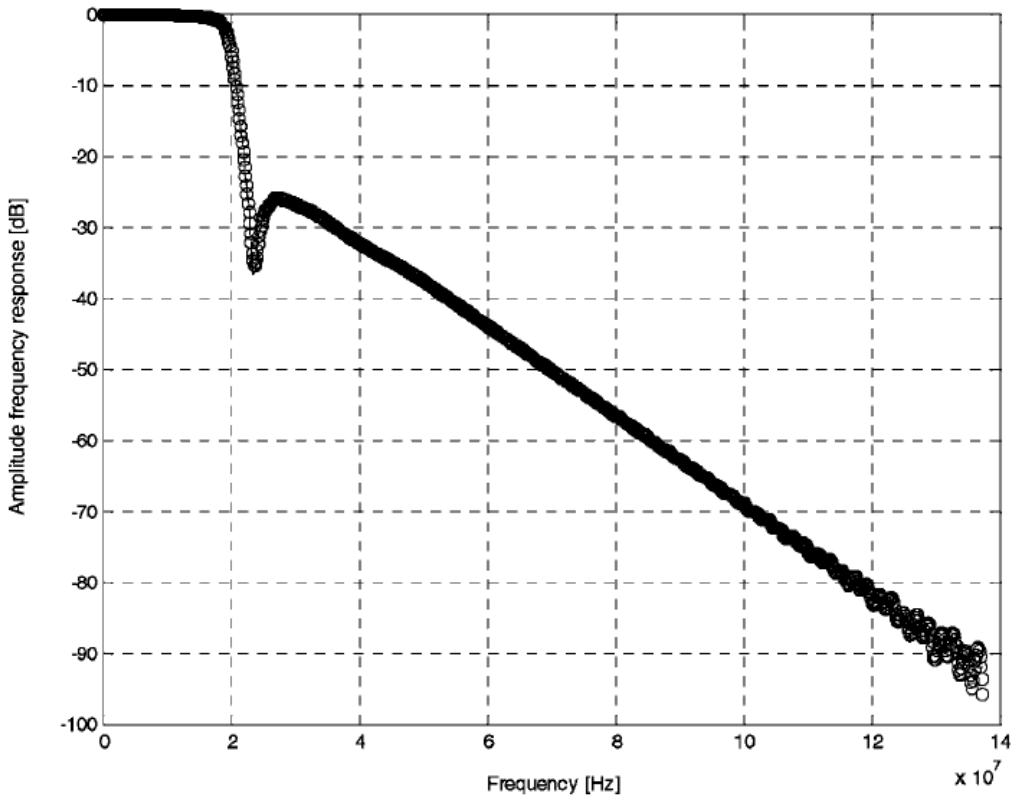
The IMUX filter

IMUX characteristic: amplitude response



The OMUX filter

OMUX characteristic: amplitude response



Effects of a nonlinear amplifier

The effect of a HPA on the transmitted constellation

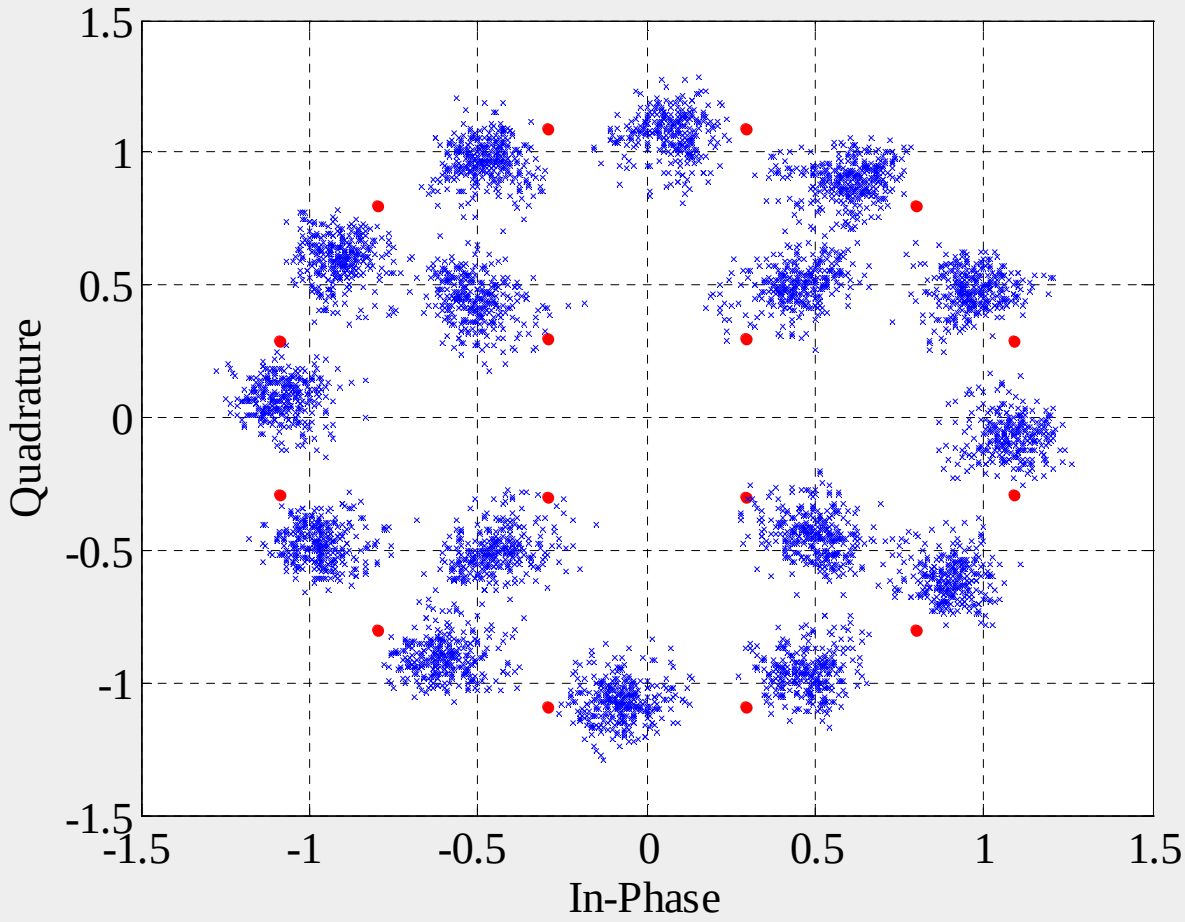
- Warping effect: it is due to the gain compression of the amplifier. In multiple rings modulations (e.g APSK), the points on the inner rings are more amplified than the points on the outer rings (AM/AM distortion). There is also a phase shift among rings (AM/PM distortion).
- Clustering effect: it is equivalent to have ISI in the received signal at the demodulator matched filter output.

The effect of a HPA on the signal spectrum

- The output signal spectrum is affected by the so-called spectral regrowth effect.
  The OMUX filter is used to limit this effect.

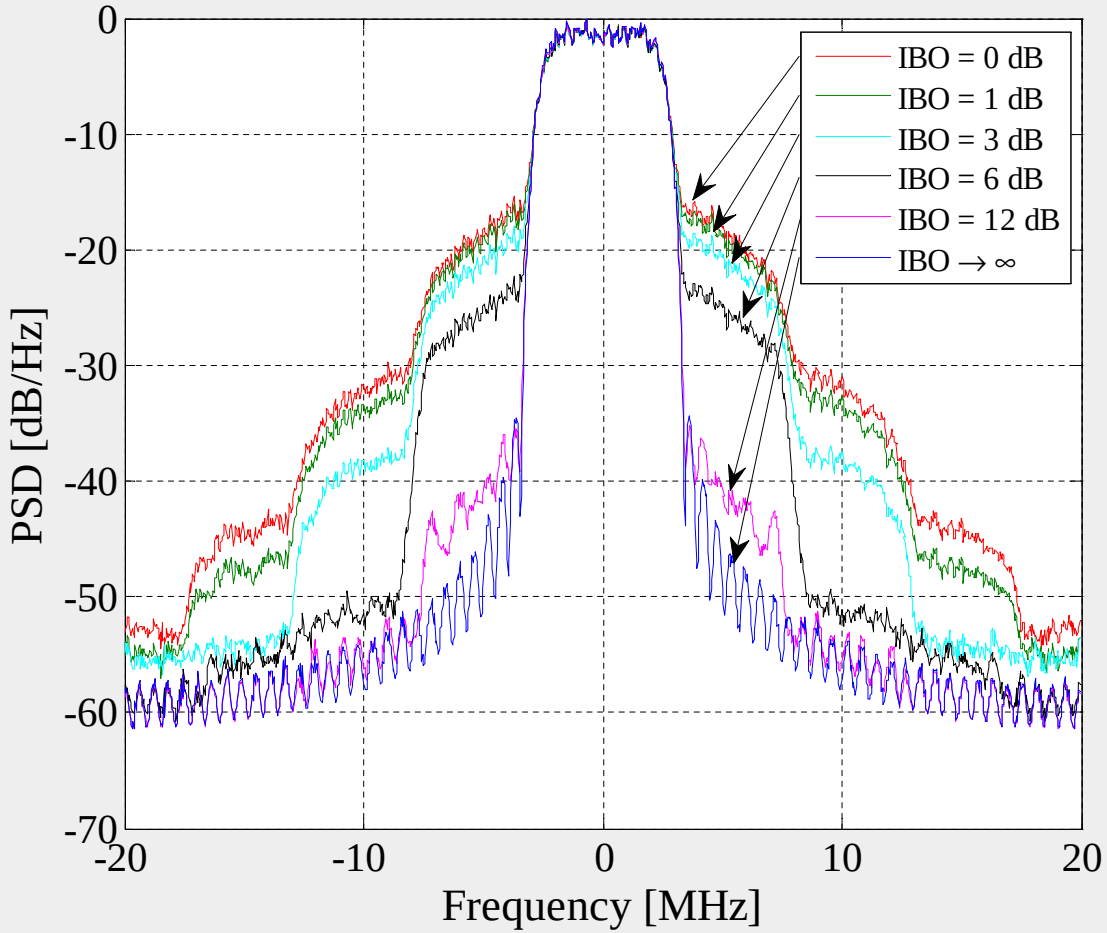
Effects of a nonlinear amplifier

Warping and clustering phenomena



Effects of a nonlinear amplifier

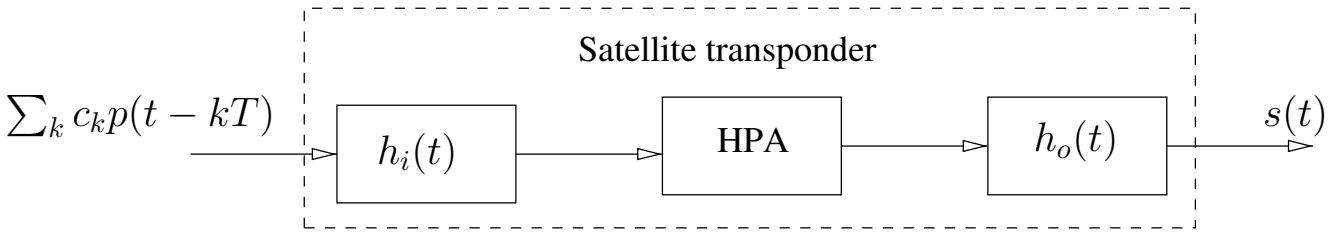
The spectral regrowth effect



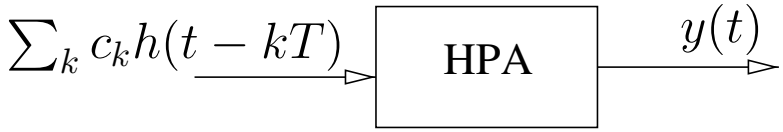
Volterra-series expansion (1/5)

The Volterra-series expansion of the satellite channel is largely adopted for modeling digital satellite channels and for designing equalizers and predistorters.

- Let us consider a linear modulation.
- The transponder model is composed of an IMUX filter $h_i(t)$ and an OMUX filter $h_o(t)$.
- The HPA is a nonlinear memoryless device.



Volterra-series expansion (2/5)



We define the impulse response of the system at the HPA input as $h(t) = p(t) \otimes h_i(t) \rightarrow$ the signal at the HPA input is

$$x(t) = \sum_k c_k h(t - kT) \, .$$

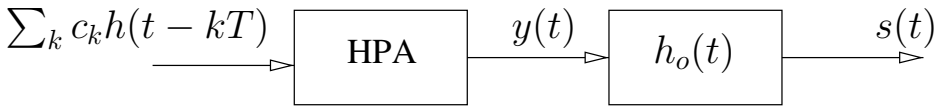
The signal at the HPA output can be expressed as a function of $x(t)$ by using the following polynomial expansion

$$y(t) = \left[\sum_m \gamma_{2m+1} \frac{1}{2^{2m}} \binom{2m+1}{m} |x(t)|^{2m} \right] x(t) \, .$$

With good approximation, the HPA can be modeled as a cubic nonlinearity:

$$y(t) \simeq \gamma_1 \sum_k c_k h(t - kT) + \frac{3}{4} \gamma_3 \sum_i \sum_j \sum_\ell c_i c_j c_\ell^* h(t - iT) h(t - jT) h^*(t - \ell T) \, .$$

Volterra-series expansion (3/5)



Signal $y(t)$ is then filtered through the OMUX filter

$$s(t) \simeq \gamma_1 \sum_k c_k h^{(1)}(t-kT) + \frac{3}{4}\gamma_3 \sum_i \sum_j \sum_\ell c_i c_j c_\ell^* h^{(3)}(t-iT, t-jT, t-\ell T) \quad (14)$$

where

$$h^{(1)}(t) = h(t) \otimes h_o(t)$$

$$h^{(3)}(t_1, t_2, t_3) = \int_{-\infty}^{\infty} h_o(\tau) h(t_1-\tau) h(t_2-\tau) h^*(t_3-\tau) d\tau$$

When multiplied by γ_1 and $\frac{3}{4}\gamma_3$, $h^{(1)}(t)$ and $h^{(3)}(t_1, t_2, t_3)$ take the form of the so-called Volterra kernels of first and third order, respectively

Volterra-series expansion (4/5)

An approximate Volterra-series expansion guarantees a simpler but still accurate expression. In the triple summation in (14), we keep only the the terms for $i = \ell$ or $j = \ell$ and obtain

$$s(t) \simeq \sum_k c_k \left[\gamma_1 h^{(1)}(t - kT) + \frac{3}{4}\gamma_3 \sum_i |c_i|^2 \overline{h}^{(3)}(t - iT, t - kT) \right]$$

where

$$\overline{h}^{(3)}(t_1, t_2) = h^{(3)}(t_2, t_1, t_1) + h^{(3)}(t_1, t_2, t_1) - I(t_1 - t_2) h^{(3)}(t_2, t_2, t_2)$$

where $I(t)$ is an indicator function, equal to one if $t = 0$ and to zero otherwise. PSK: $|c_i|^2$ and

$$s(t) \simeq \sum_k c_k \overline{h}(t - kT)$$

where $\overline{h}(t) = \gamma_1 h^{(1)}(t) + \frac{3}{4}\gamma_3 \sum_i \overline{h}^{(3)}(t - iT, t)$. The optimal receiver is simply designed. Note that we need to use a filter matched to $\overline{h}(t)$.

Volterra-series expansion (5/5)

APSK or QAM: We use the approximation $|c_i|^2 = E\{|c_i|^2\}$ only for $i \neq k$ and preserve the term $|c_k|^2$, which helps to take into account of the warping effect suffered by the constellation points. The new signal model becomes

$$s(t) = \sum_k c_k \left[g^{(1)}(t - kT) + |c_k|^2 g^{(3)}(t - kT) \right]$$

where

$$g^{(1)}(t) = \gamma_1 h^{(1)}(t) + \frac{3}{4}\gamma_3 \sum_{i \neq k} \bar{h}^{(3)}(t - iT, t)$$

$$g^{(3)}(t) = \frac{3}{4}\gamma_3 \bar{h}^{(3)}(t, t) \, .$$

We can build the optimal receiver by using two matched filters and proper branch mentrics.

Total Degradation

- The performance in the presence of nonlinear effects are generally evaluated by using is the total degradation

$$D_{\text{TOT}} = [E_s/N_0]^{\text{NL}} - [E_s/N_0]^{\text{LIN}} + \text{OBO [dB]}$$

where

  - $[E_s/N_0]^{\text{NL}}$ is the SNR required to achieve the target frame error rate in the nonlinear channel
  - $[E_s/N_0]^{\text{LIN}}$ is the SNR required to achieve the target frame error rate in the linear channel
  - OBO is defined as the power ratio (in dBs) between the unmodulated carrier at saturation and the modulated carrier at the output of the amplifier

Data Predistortion (1/8)

When the HPA AM/AM and AM/PM characteristics are properly estimated and fed back to the transmitter, the sequence of symbols $\{c_k\}$ can be properly predistorted to form the sequence $\{d_k\}$ that is transmitted instead, in order to compensate for the effect of the non-linearity and possibly to reduce the ISI. The dynamic data predistortion technique with the best performance is that suggested for the application in DVB-S2 systems. Symbol d_k transmitted at time k is a function of a sequence of $2L_p + 1$ input symbols, i.e., $d_k = f(c_{k-L_p}, \ldots, c_k, \ldots, c_{k+L_p})$. The mapping f at the transmitter is implemented through a look-up table (LUT), which is computed through an iterative procedure performed off-line for each modulation format, setting of the system parameters, and SNR value. This procedure searches the best trade-off between the interference reduction and the increase of the OBO. The complexity at the transmitter depends on the number of symbols accounted for through the parameter L_p. The transmitted signal is thus

$$s_T(t) = \sum_k d_k p(t - kT) \tag{15}$$

Data Predistortion (2/8)

- $L_p = 0 \rightarrow$ static predistortion. $L_p > 0 \rightarrow$ dynamic predistortion. We wil see how to set function f by considering the case of a static predistorter 8the extension is trivial).
- The static predistortion technique consists in modifying the constellation points to minimize centroids distance of samples at the output of the front end filter from the reference constellation
- Function f is set off-line in the absence of AWGN, generating S blocks of W symbols over which the centroids are computed and evaluating the error signal at the end of each block.
- The constellation points are thus updated through an iterative LMS-like algorithm

Data Predistortion (3/8)

Let $\mathcal{D} = \{d(i)\}_{i=1}^{M}$ be the predistorted constellation, and $\mathcal{C} = \{c(i)\}_{i=1}^{M}$ the original constellation (e.g., 16APSK). Each $d(i)$ is related one-to-one to $c(i)$.

The algorithm

1. Set $\mathcal{D} = \mathcal{C}$.
2. Draw a sequence of W symbols $\{c_k\}_{k=1}^{W}$ from $\mathcal{C}$.
2. For each $c_k = c(i)$ transmit $d(i)$ and consider the received samples $\{z_k\}$.
3. For each $i = 1, \ldots, M$ consider the values of the absolute error and phase error

$$e_r(i) = |c(i)| - \frac{1}{N(i)} \sum_{k:d_k=d(i)} |z_k| \tag{16}$$

$$e_\theta(i) = \angle c(i) - \frac{1}{N(i)} \sum_{k:d_k=d(i)} \angle z_k \tag{17}$$

where $N(i)$ is the number of times for which $d(i)$ was transmitted.

Data Predistortion (4/8)

The algorithm (cont'd)

4 Update the predistorted constellation as

$$|d(i)| = |d(i)| + \gamma_r e_r(i) \quad (18)$$

$$\angle d(i) = \angle d(i) + \gamma_\theta e_\theta(i) \quad (19)$$

where γ_r and γ_θ are two adaptive step sizes.

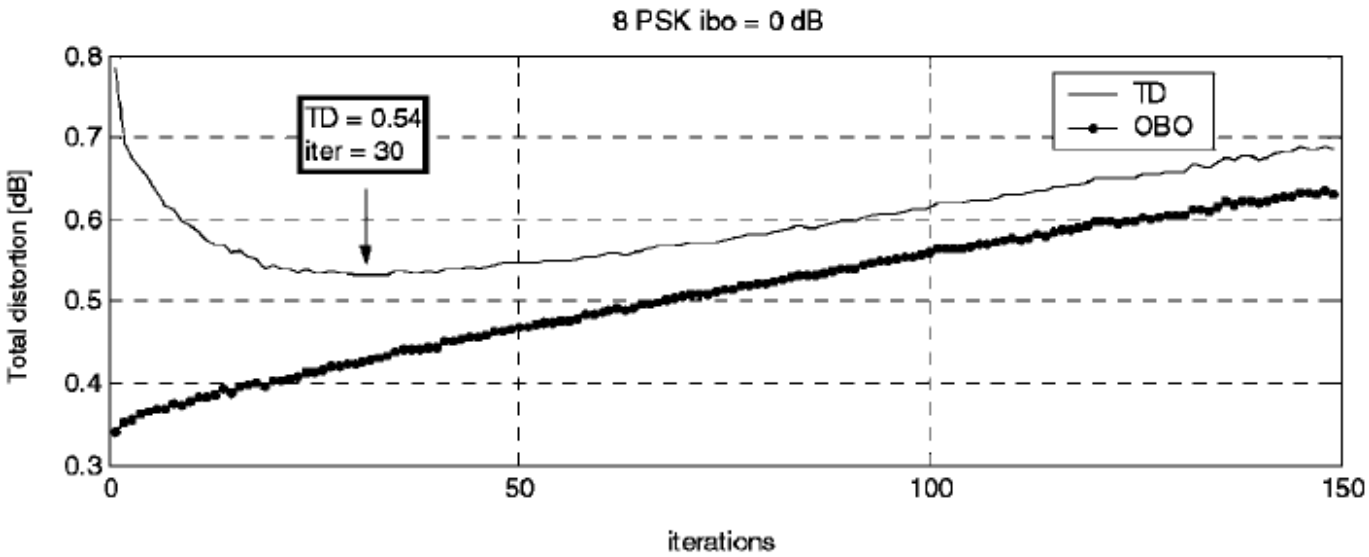
5 Go to 2 and repeat (for a total number S of iterations).

How S should be set? If S is too low, there is no predistortion. If S is too high, the PAPR increases, and also the OBO.

S must be set as the one who offers the best trade-off.

Data Predistortion (5/8)

Example: Total degradation (TD) for 8PSK as function of S



- $S = 30$ seems to be an almost optimal value.
- This procedure searches the best trade-off between the interference reduction and the increase of the OBO.

Data Predistortion (6/8)

- Since it considers only the current symbol, it does not compensate for clustering but only for warping
- It has lower computational complexity with respect to methods with memory (dynamic)
- Let L_p be the memory of the predistorter. The predistorted constellation $\mathcal{D} = \{d(i)\}_{i=1}^{C}$ will now have cardinality $C = M^{2L_p+1}$.
- Each symbol $d(i)$ is in a one-to-one correspondence with a set of $2L_p + 1$ symbols beloging to the original constellation $\mathcal{C} = \{c(i)\}_{i=1}^{M}$.
- In other words, each predistorted symbols d_k is obtained from a LUT: $d_k = f(c_{k-L_p}, \ldots, c_{k+L_p})$.
- The computation of $\mathcal{S}$ is done through the same algorithm as for the static predistortion, with the difference that each time we need to update C symbols (complexity is exponential with L_p).

Data Predistortion (7/8)

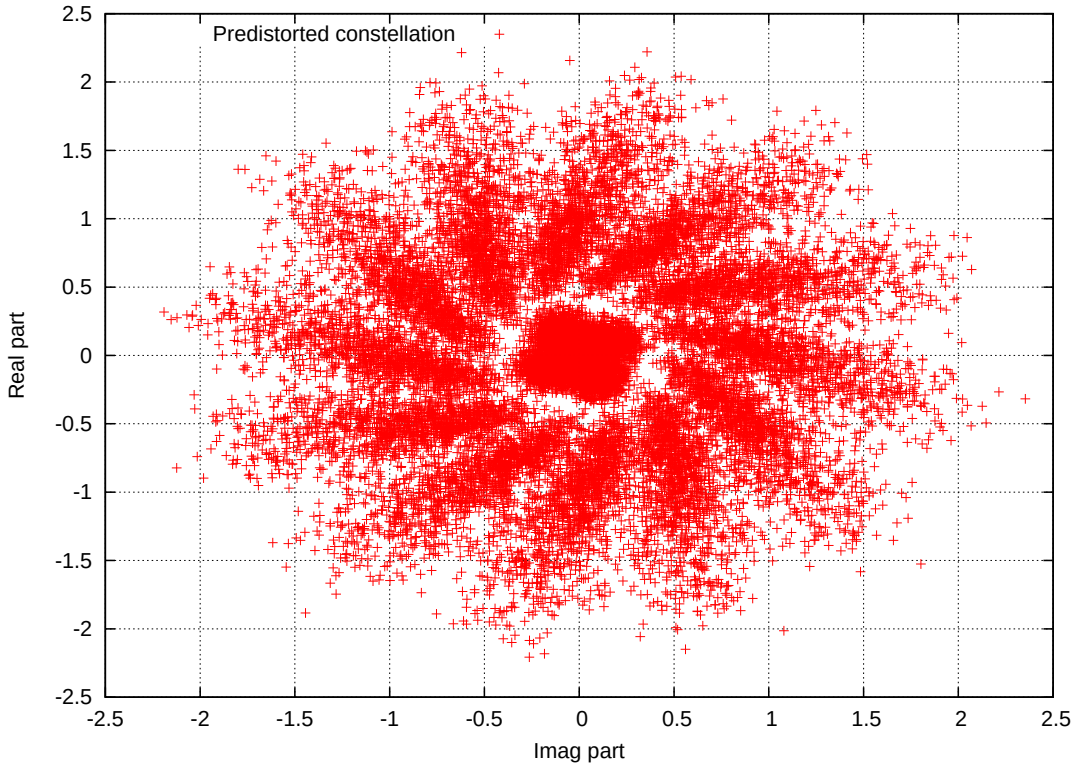


Figure: Transmitted constellation for 16APSK, $L_p = 2$

Data Predistortion (8/8)

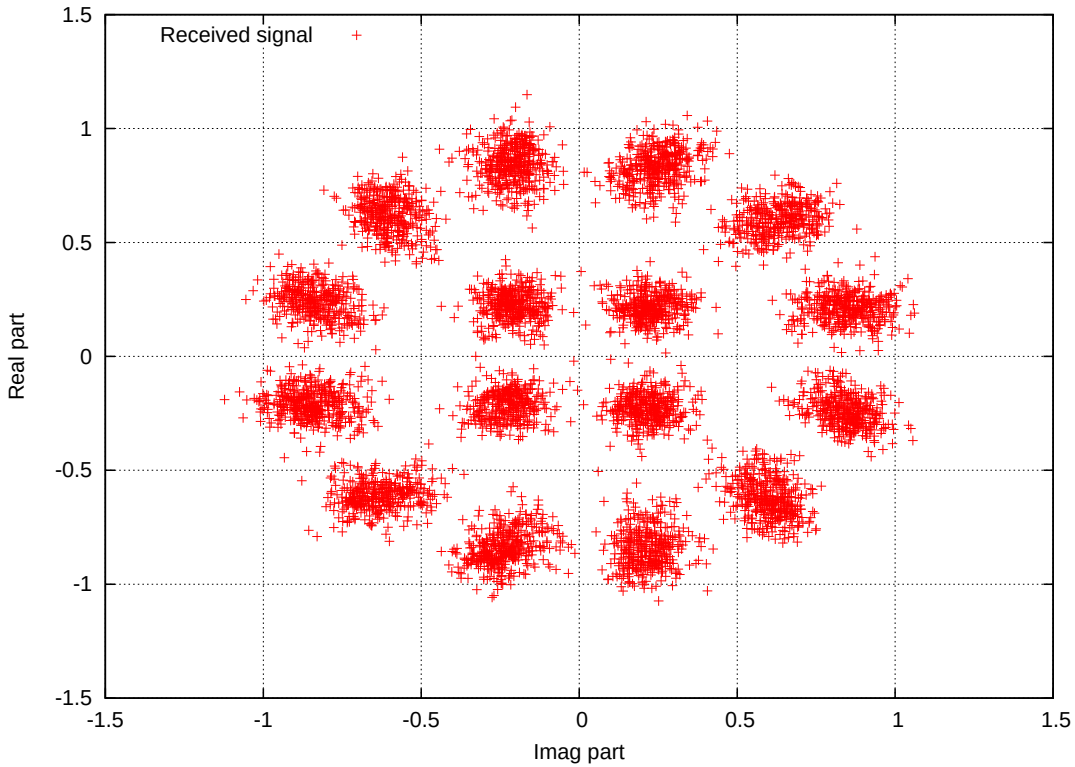


Figure: Received constellation (noiseless) for 16APSK, $L = 2$

Detection algorithm (1/3)

We can model the encoder, modulator, IMUX and OMUX filters, and the HPA as a finite-state machine (FSM), whose input is the symbol sequence $\boldsymbol{a}$ and whose output $s(t, \boldsymbol{a})$ can be expressed as

$$s(t, \boldsymbol{a}) = \sum_{k=0}^{K-1} \bar{s}(t - kT, a_k, \sigma_k) \,, \tag{20}$$

where signal $\bar{s}(t - kT, a_k, \sigma_k)$ is assumed to have support in the interval $[kT, (k+1)T)$ and is thus a chip (a slice) of useful signal. It can be computed from the Volterra-series expansion. The state σ_k of the FSM contains the previous L channel inputs, L being the memory length of the channel:

$$\sigma_k = (c_{k-1}, c_{k-2}, \ldots, c_{k-L}) = (a_{k-1}, a_{k-2}, \ldots, a_{k-L}, \mu_{k-L}) \,.$$

Therefore, the optimal maximum a posteriori (MAP) sequence detector consists of a bank of filters matched to all possible $M^{(L+1)}$ waveforms $\bar{s}(t - kT, a_k, \sigma_k)$, followed by a VA with branch metric

$$\lambda(a_k, \sigma_k) = \mathcal{R}\left[\int_{kT}^{(k+1)T} r(t)\bar{s}(t - kT, a_k, \sigma_k)\mathrm{d}t\right] - \frac{1}{2}\int_{kT}^{(k+1)T} |\bar{s}(t - kT, a_k, \sigma_k)|^2 \mathrm{d}t$$

Detection algorithm (2/3)

The implementation of the bank of matched filters can be performed by working in the digital domain. Let us denote by B_s the bandwidth of signal $s(t)$. Clearly $B_s \geq B$ (B is the bandwidth of the transmitted signal) due to the spectral broadening caused by the nonlinear transponder. A sufficient statistic can be obtained through the technique already described. It is assumed $B_s < \eta/2T$, where η is a proper integer. A sufficient statistics is thus obtained by extracting η samples per symbol interval from the received signal prefiltered by means of an analog lowpass filter which leaves unmodified the useful signal and has a vestigial symmetry around $\eta/2T$. The obtained sufficient statistic, reads

$$\boldsymbol{r} = \boldsymbol{s} + \boldsymbol{w}$$

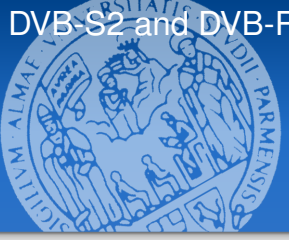
where $\boldsymbol{s} = [s(0), s(T/\eta), s(2T/\eta), \ldots, s((\eta-1)NT/\eta)]^T$ and $\boldsymbol{w}$ is a vector collecting the noise samples after filtering.

Detection algorithm (3/3)

The condition on the vestigial symmetry of the analog prefilter ensures that the components of the noise vector $\boldsymbol{w}$ are i.i.d. complex Gaussian random variables with independent real and imaginary components, having mean zero and variance $N_0\eta/T$. The bank of matched filters can thus be implemented by using the samples $\boldsymbol{r}$ and the samples of waveforms $\{\bar{s}(t-kT,a,\sigma)\}$. In fact, by denoting $\overline{\boldsymbol{s}}(a,\sigma)=(\bar{s}(0,a,\sigma),\bar{s}(T/\eta,a,\sigma),\ldots,\bar{s}((\eta-1)T/\eta,a,\sigma))^T$ and $\boldsymbol{r}_k=(r(kT),r(kT+T/\eta),\ldots,r(kT+(\eta-1)T/\eta))^T$, it is

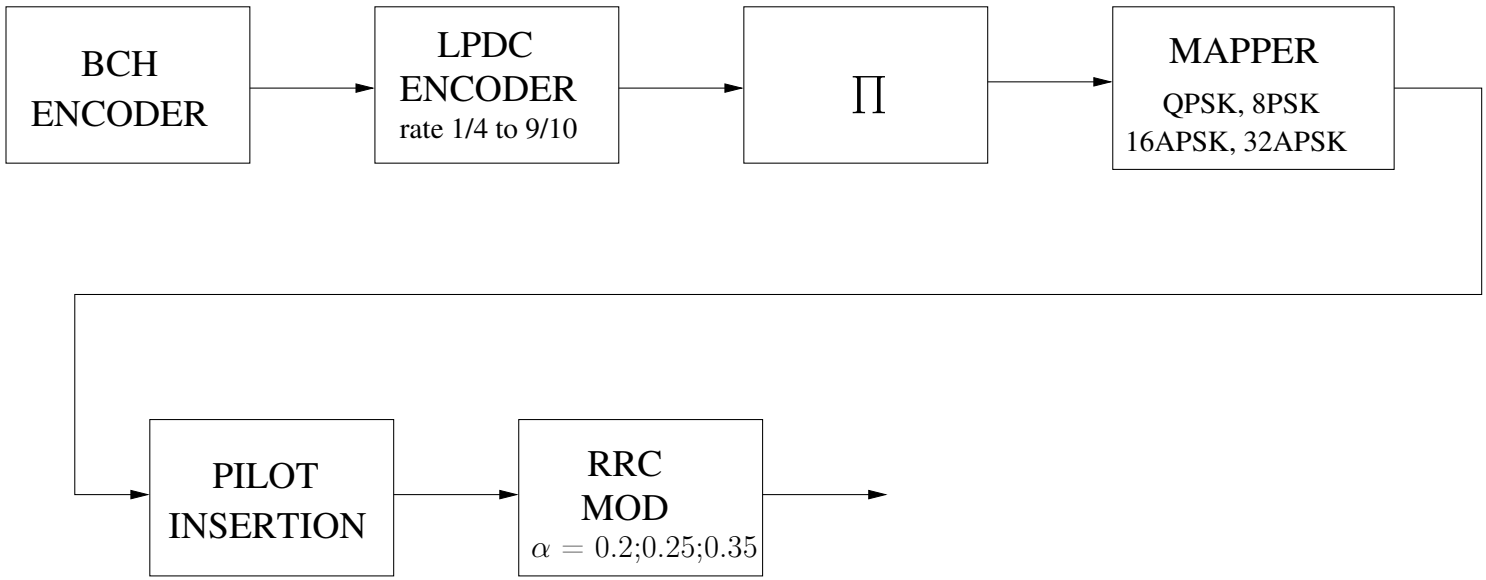
$$\int_{kT}^{(k+1)T} r(t)\bar{s}^*(t-kT,a,\sigma)\,dt=\overline{\boldsymbol{s}}^H(a,\sigma)\boldsymbol{r}_k\,.$$

Outline



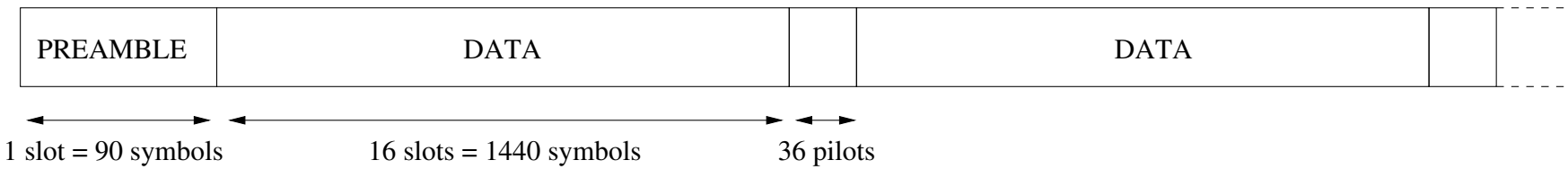
DVB-S2 Transmitter (1/2)

Digital Video Broadcasting - Satellite 2nd generation (DVB-S2) standard: transmitter



DVB-S2 Transmitter (2/2)

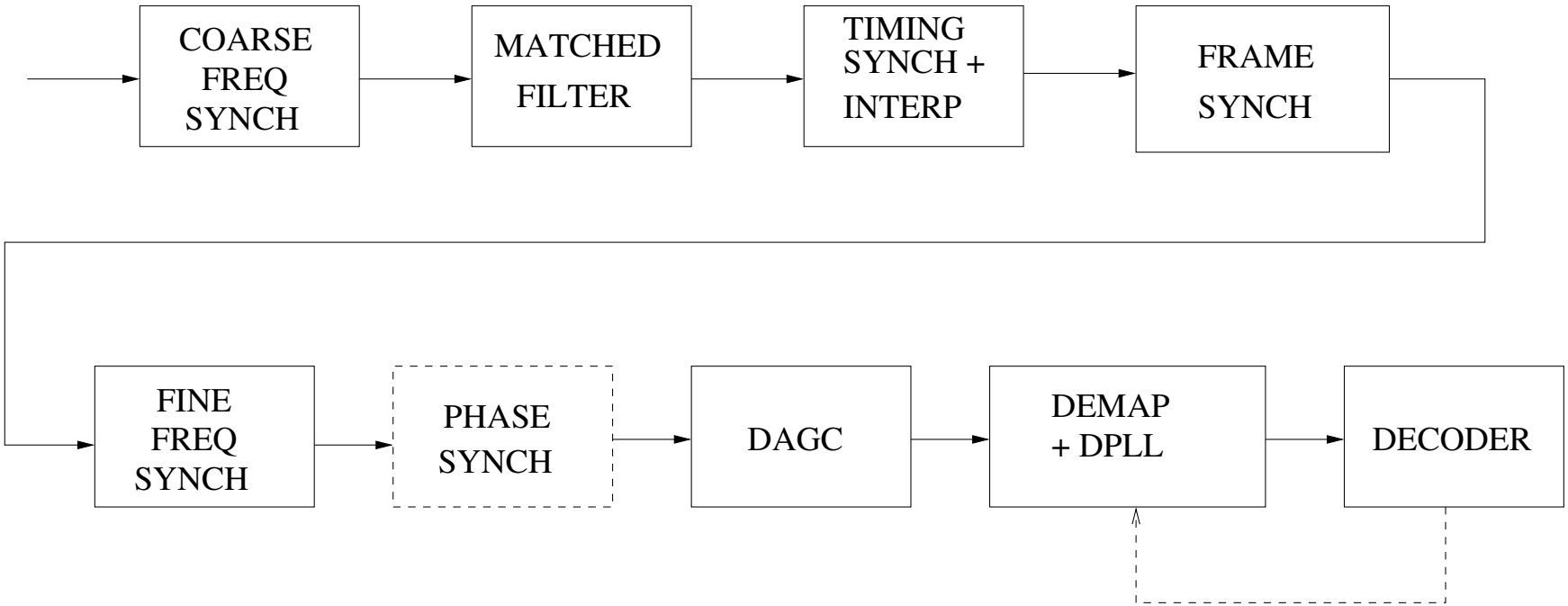
Known data symbols to be exploited for synchronization purposes are placed in a preamble and in pilot fields equally spaced all along the frame. Namely, 36 pilot symbols are inserted every 1440 information symbols, whereas 90 pilot symbols are placed at the beginning of the frame as preamble.



- At the transmitter the base pulse is an RRC-shaped pulse with roll-off factor equal to 0.2, 0.3, or 0.35
- The following modulation formats are considered: QPSK, 8PSK, 16APSK, and 32APSK

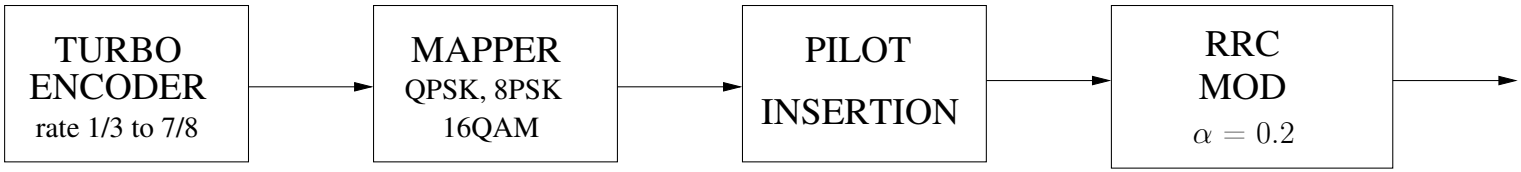
DVB-S2 Receiver

A possible block scheme for a DVB-S2 receiver:

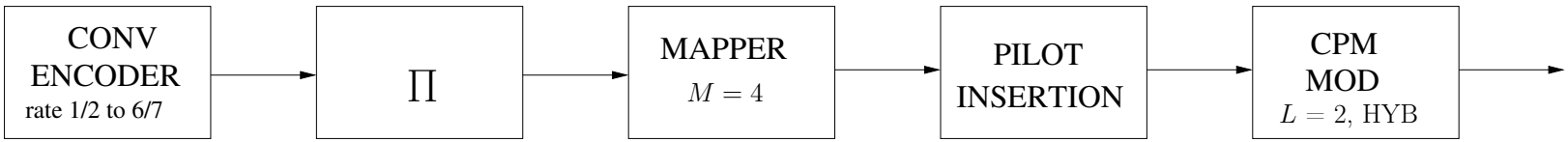


DVB-RCS2 Transmitter (1/2)

Digital Video Broadcasting - Return Channel Satellite 2nd generation (DVB-RCS2) standard: two possible transmitters. When linear modulations are adopted, the transmitter scheme is

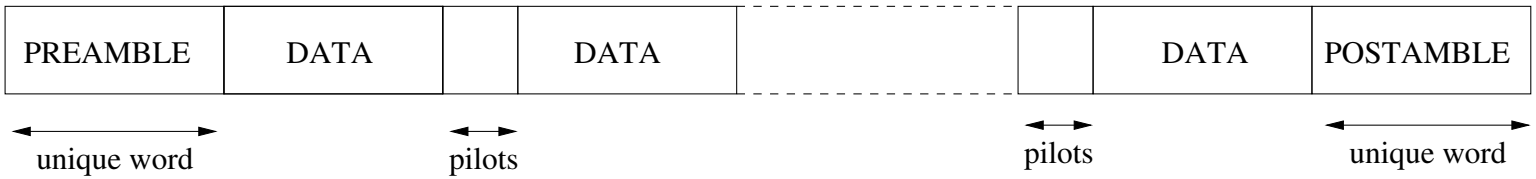


Otherwise, when continuous phase modulations are employed, the transmitter scheme is

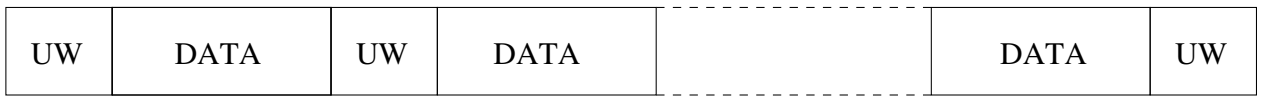


DVB-RCS2 Transmitter (2/2)

DVB-RCS2 standard considers different pilot insertions. For linear modulations, preamble and pilot field length and position can be customized.



For CPM, the unique word (UW) is decomposed in pilot fields placed all along the frame.



DVB-RCS2 Receiver

A possible block scheme for a DVB-RCS2 receiver is: