Introduction to Per-Survivor Processing

second edition

Riccardo Raheli

Università degli Studi di Parma
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181A
I-43100 Parma - Italia

E-mail: raheli@unipr.it
http://www.tlc.unipr.it/raheli

June 2004

Motivation

Why a course on Per-Survivor Processing (PSP)?

PSP is useful to communication system designers thanks to its *broad applicability* in coping with *hostile transmission environments*, such as those of many current applications

PSP is technically elegant and intellectually appealing. As many interesting ideas, it is *general, intuitive and conceptually straightforward*. It is a nice example of a recent research result which may be worth describing in a structured advanced University course in the area of digital transmission theory and techniques

PSP is intriguing from the scientific and historical viewpoints. Like many other ideas, PSP has been *reinvented* independently by many researchers over the last decades, with different contexts and formulations each time. Its *conceptual roots can be found in earlier general theoretical results*, but this fact was fully understood only after its invention

Foreword

Unfortunately, this course might be unclear (*and likely will !*)

Please, feel free to ask questions. Doing so you will help:*

- ⇝ Yourself understanding what is going on
- ↪ Your colleagues understanding questions they had not even thought of
- ↣ The instructor realizing what is unclear and should be better explained

You will also:

- ⇝ Avoid falling behind (if you do in the first lectures, you will hardly recover)
- ↬ Contribute to make the lectures more stimulating and pleasant

*Arrows by LaTeX and AMSLaTeX

Outline

1. Review of detection techniques
2. Detection under parametric uncertainty
3. Per-Survivor Processing (PSP): concept and historical review
4. Classical applications of PSP
    - 4.1 Complexity reduction
    - 4.2 Linear predictive detection for fading channels
    - 4.3 Adaptive detection
5. Advanced applications of PSP

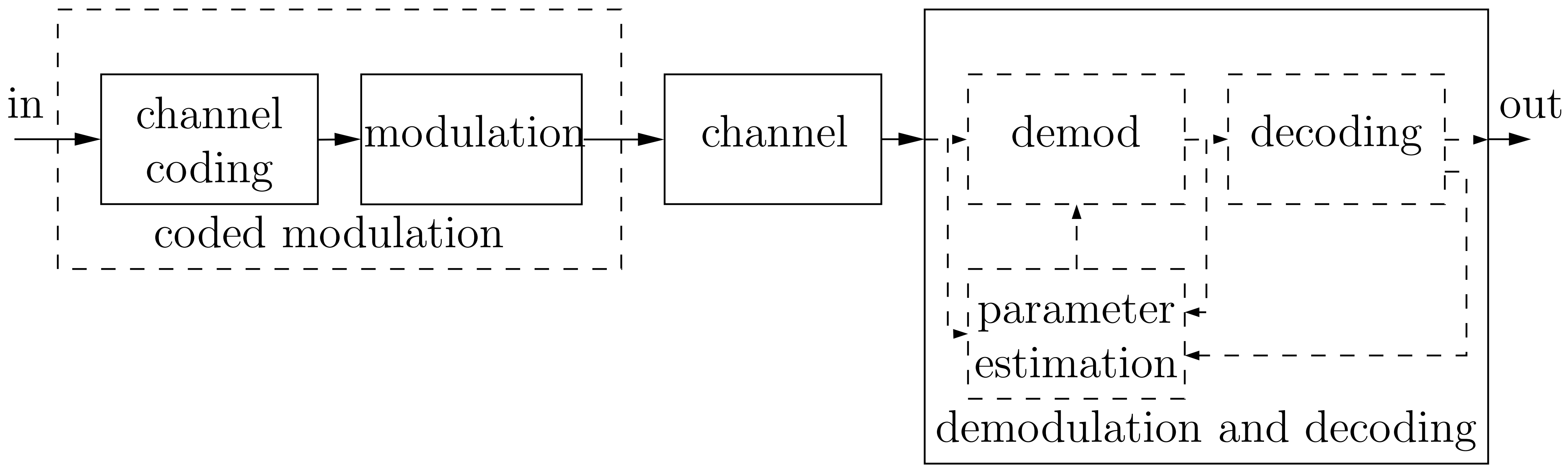
Prerequisite: A course in Digital Transmission Theory

1. Review of detection techniques

Outline

1. Review of detection techniques
2. Detection under parametric uncertainty
3. Per-Survivor Processing (PSP): concept and historical review
4. Classical applications of PSP
    - 4.1 Complexity reduction
    - 4.2 Linear predictive detection for fading channels
    - 4.3 Adaptive detection
5. Advanced applications of PSP

General model of transmission systems
... how about storage systems?



Focus on: demodulation and decoding

Principal channel models

Additive White Gaussian Noise (AWGN) channel

Static dispersive channel

Flat fading channel

Dispersive fading channel

Phase uncertain channel

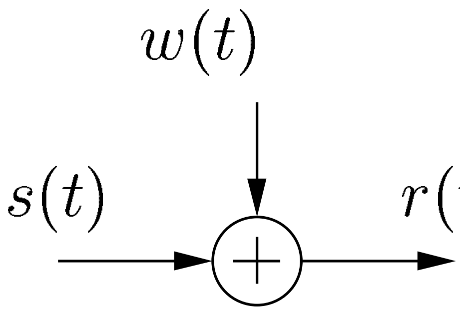
Like-signal (or cochannel) interference channel

Nonlinear channel

Transition noise channel

Combinations

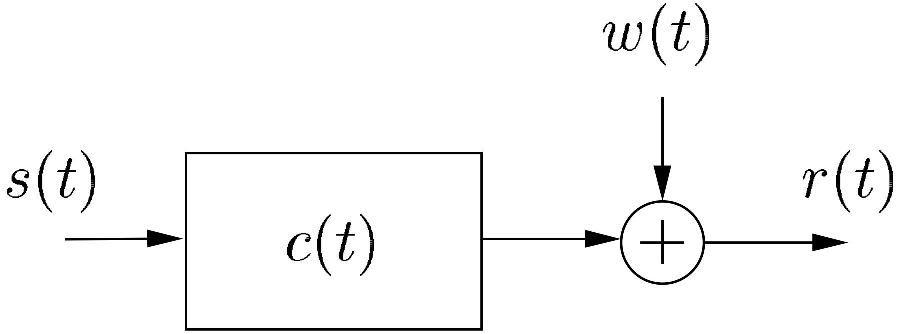
AWGN channel



$$r(t) = s(t) + w(t)$$

$w(t)$: circular complex AWGN

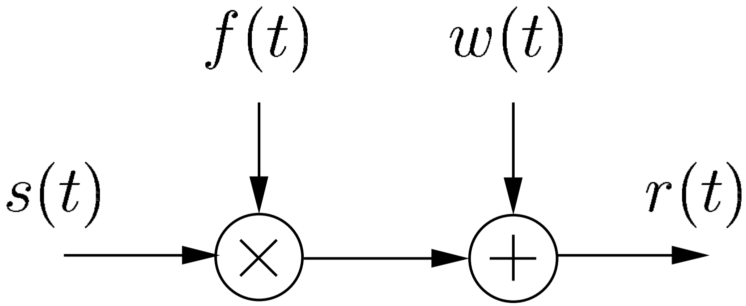
Static dispersive channel



$$r(t) = s(t) \star c(t) + w(t)$$

$c(t)$: channel impulse response

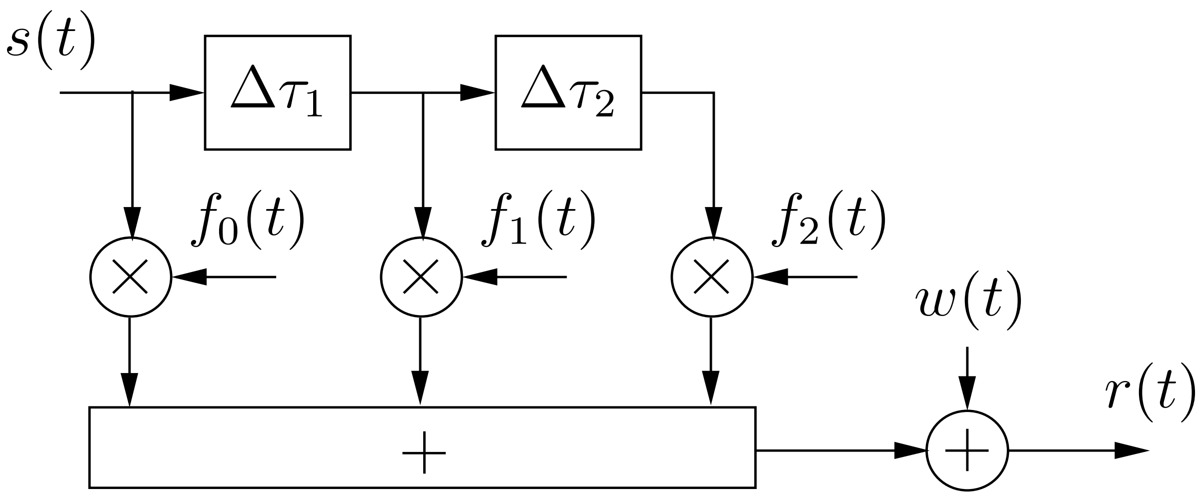
Flat fading channel



$$r(t) = s(t)f(t) + w(t)$$

$f(t)$: circular complex Gaussian random process

Dispersive fading channel

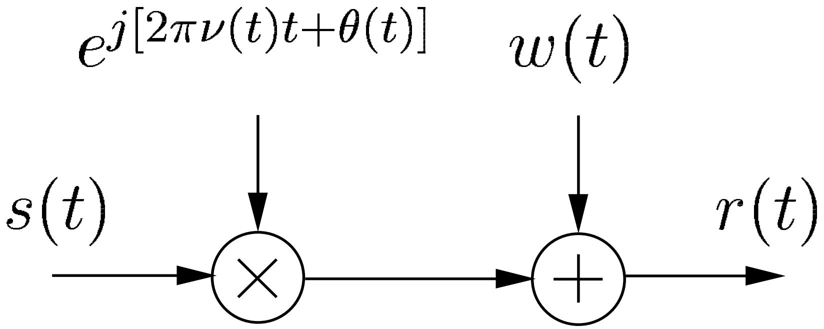


$$r(t) = \sum_{l=0}^{L} f_l(t) s(t - \tau_l) + w(t) \qquad \tau_l = \tau_0 + \sum_{i=1}^{l} \Delta\tau_i$$

$f_l(t)$: independent circular complex Gaussian random processes

The l-th dominant propagation path has delay τ_l

Phase-uncertain channel



$$r(t) = s(t)e^{j[2\pi\nu(t)t+\theta(t)]} + w(t)$$

$\nu(t)$: frequency shift

$\theta(t)$: phase shift

Special cases:

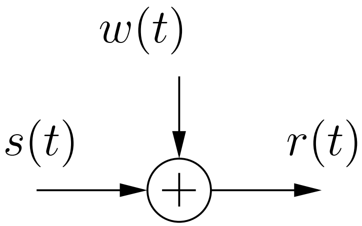
– Phase noncoherent channel ($\nu(t) = 0$, $\theta(t) = \theta$)

– Frequency offset (or Doppler shift) channel ($\nu(t) \neq 0$, $\theta(t) = \theta$)

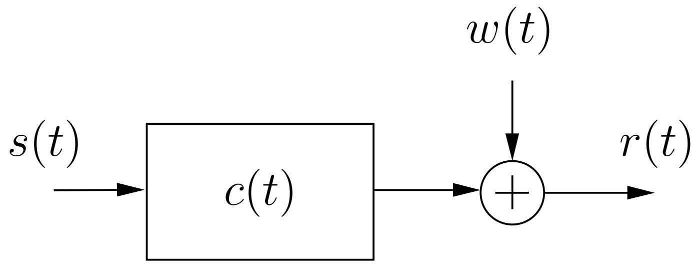
– Phase noisy channel ($\nu(t) = 0$, $\theta(t)$ is a Wiener random process)

Principal channel models

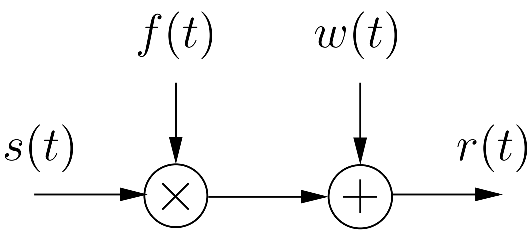
Overview



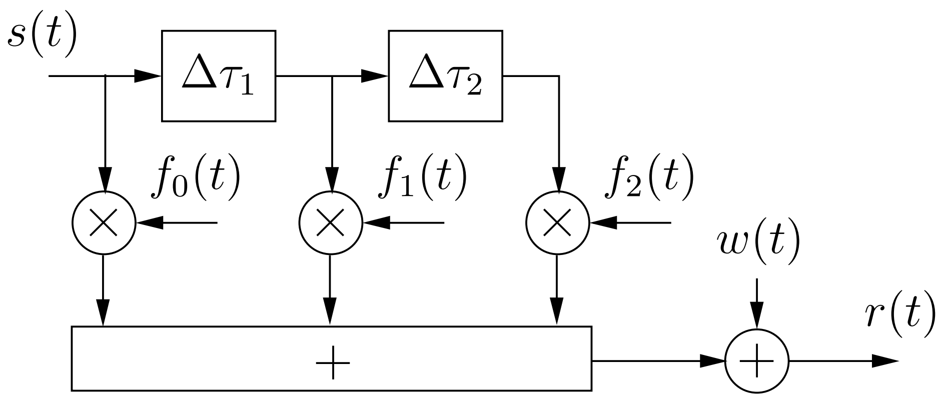
(a)



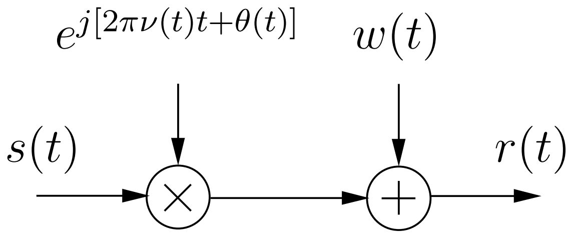
(b)



(c)

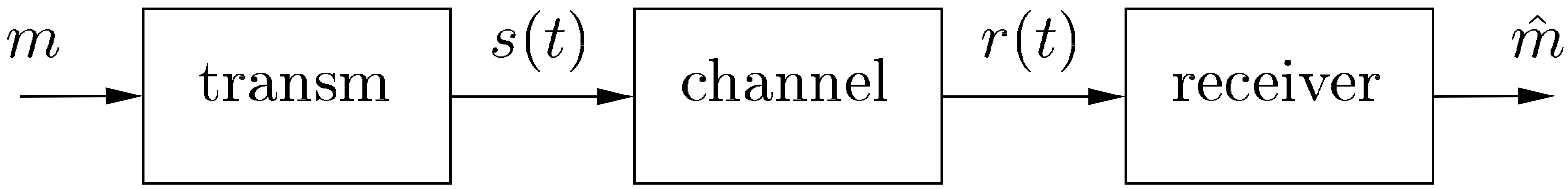


(d)



(e)

Statistical Detection Theory
Optimal detection of M-ary signals



$m \in \{m_i\}_{i=1}^{M}$ $\quad s(t) \in \{s_i(t)\}_{i=1}^{M}$ $\quad r(t)$ $\quad \hat{m} \in \{m_i\}_{i=1}^{M}$

Probabilistic modeling $\Rightarrow$ Optimal decision (detection) *strategy*

Minimize $P(m \neq \hat{m})$:

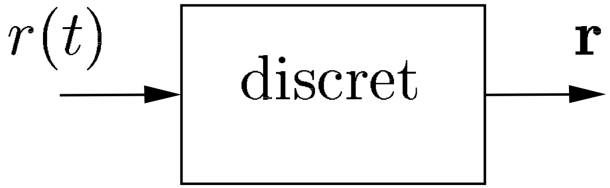
$\Rightarrow$ maximize: $\quad P(m = \hat{m}) = E\left\{P\left[m = \hat{m} | r(t)\right]\right\}$

$\Rightarrow$ maximize: $\quad P\left[m = \hat{m} | r(t)\right] \quad \forall r(t)$ $\quad$ (positive)

For $\hat{m} = m_i$, $P\left[m = \hat{m} | r(t)\right] = P\left[m = m_i | r(t)\right]$ (APP)

$\Rightarrow$ MAP strategy: $\quad \hat{m} = \underset{m_i}{\operatorname{argmax}} \underbrace{P\left[m = m_i | r(t)\right]}_{P[m_i | r(t)]}$

Statistical Detection Theory
Computation of the APPs



Discretization (finite dimensionality) $\Rightarrow$ *Sufficient statistic*

APPs: $$P(m_i|\mathbf{r}) = \frac{p(\mathbf{r}|m_i)P(m_i)}{p(\mathbf{r})} \sim p(\mathbf{r}|m_i)P(m_i)$$

$\sim$: monotonic relationship with respect to the variable of interest

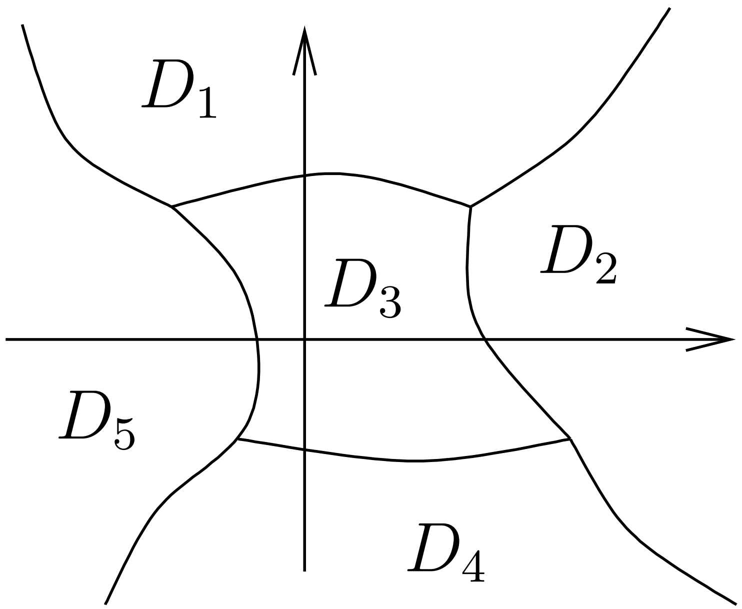
MAP strategy: $$\hat{m} = \operatorname*{argmax}_{m_i} P(m_i|\mathbf{r}) = \operatorname*{argmax}_{m_i} p(\mathbf{r}|m_i)P(m_i)$$

Statistical information:

$P(m_i)$: information source

$p(\mathbf{r}|m_i)$: overall system (transmitter, channel, discretizer)

Statistical Detection Theory
Geometric interpretation



Decision region:

$$D_i = \{\mathbf{r} : P(m_i|\mathbf{r}) = \max_{m_k} P(m_k|\mathbf{r})\}$$

Signal detection is a geometric game

Statistical Detection Theory

Special case: Strategy for the AWGN channel

Discretization: signal space spanned by $\{s_i(t)\}_{i=1}^{M}$ is *relevant*:

$$r_k = \int_0^T r(t)\varphi_k^*(t)dt \quad \text{with} \quad \{\varphi_k(t)\}_{k=1}^{Q} \quad \text{(basis)}$$

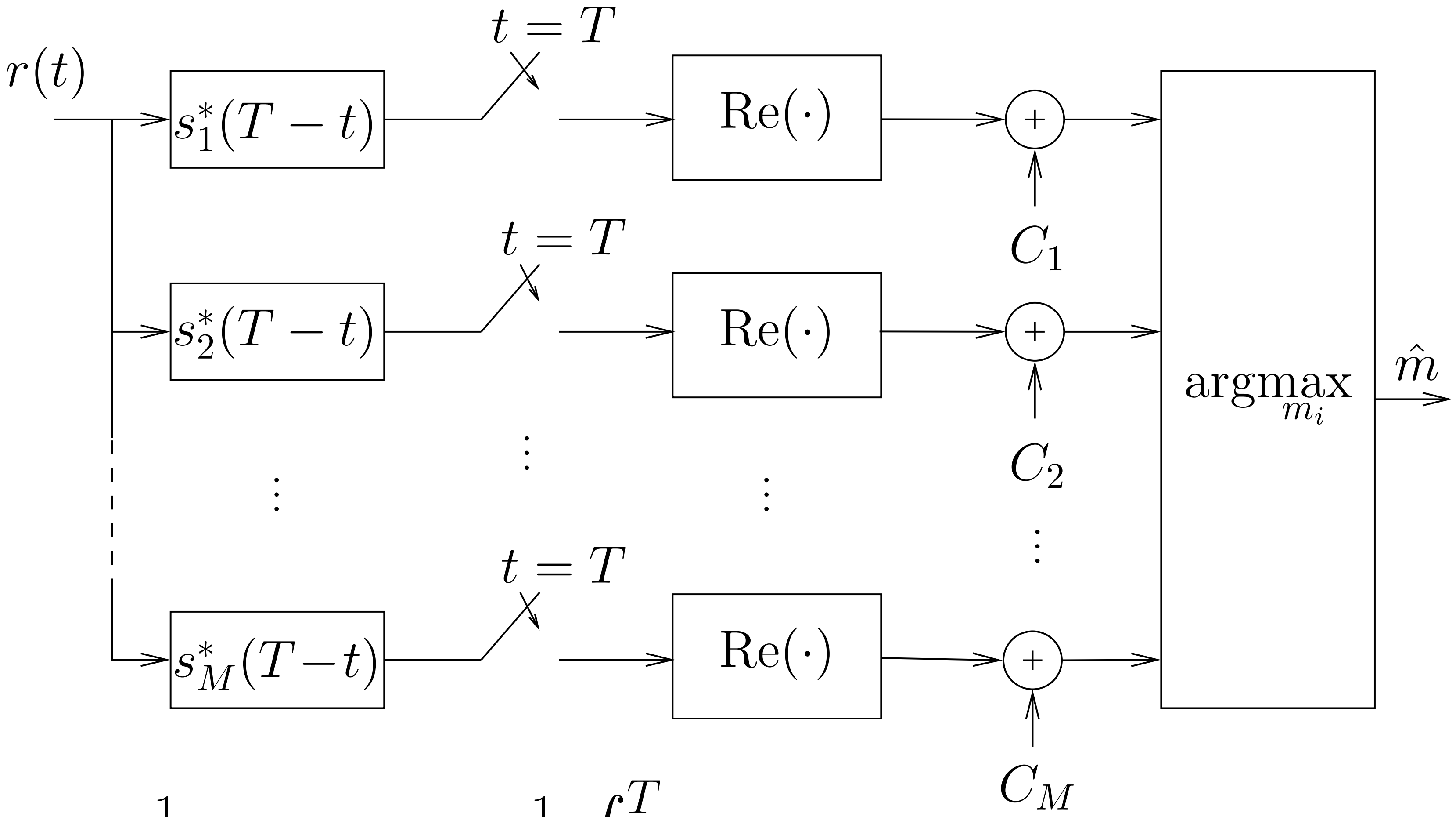
$$\{m = m_i\} \quad \Rightarrow \quad r(t) = s_i(t) + w(t) \quad \Rightarrow \quad \mathbf{r} = \mathbf{s}_i + \mathbf{w}$$

APPs (but for a factor):

$$\begin{aligned} p(\mathbf{r}|m_i)P(m_i) &= \frac{1}{(\pi\sigma^2)^Q} \exp\left[-\frac{1}{\sigma^2}||\mathbf{r}-\mathbf{s}_i||^2\right] P(m_i) \\ &\sim -||\mathbf{r}-\mathbf{s}_i||^2 + \sigma^2 \ln P(m_i) \\ &\sim \text{Re}\left(\mathbf{r}^T\mathbf{s}_i^*\right) - \frac{1}{2}||\mathbf{s}_i||^2 + \frac{1}{2}\sigma^2 \ln P(m_i) \\ &= \text{Re}\left[\int_0^T r(t)s_i^*(t)dt\right] - \frac{1}{2}\int_0^T |s_i(t)|^2 dt + \frac{1}{2}\sigma^2 \ln P(m_i) \end{aligned}$$

Statistical Detection Theory

Special case: Receiver for the AWGN channel

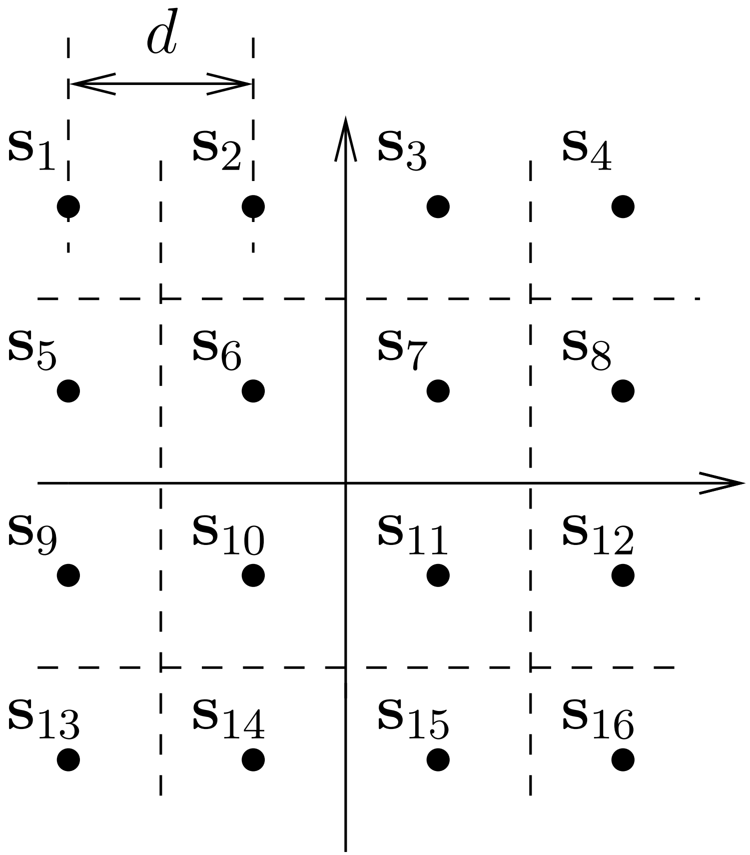


$$C_i = \frac{1}{2}\sigma^2 \ln P(m_i) - \frac{1}{2}\int_0^T |s_i(t)|^2 dt$$

$\{s_i(t)\}_{i=1}^M$ and σ^2 must be known (unless ML)

Statistical Detection Theory

Special case: Decision regions for the AWGN channel



Decision regions are polytopes

2D example: 16QAM (quadrature amplitude modulation)

Statistical Detection Theory
Problem 1

Let the observation vector be the concatenation of two subvectors

$$\mathbf{r}^T = (\mathbf{r}_1^T, \mathbf{r}_2^T)$$

and assume the following condition is satisfied

$$p(\mathbf{r}_2|\mathbf{r}_1, m_i) = p(\mathbf{r}_2|\mathbf{r}_1) \quad \forall m_i$$

Show that vector $\mathbf{r}_2$ is *irrelevant*, given $\mathbf{r}_1$, in the decision problem and can be discarded (Theorem of irrelevance)

Hint: *formulate the MAP detection strategy in terms of the conditional joint pdf of these vectors and use chain factorization*

Statistical Detection Theory
Problem 2

Consider an M-ary signaling scheme with signal set $\{s_i(t)\}_{i=1}^{M}$

Assuming signal $s_i(t)$ is sent, the received signal at the output of an AWGN phase noncoherent channel is

$$r(t) = s_i(t)\, e^{j\theta} + w(t)$$

where θ is uniformly distributed over 2π

A. Determine a discretization process of the received signal which provides a sufficient statistic for MAP detection

Hint: *Extend the results for the simple AWGN channel*

B. Derive the non coherent MAP strategy

C. Give examples of signal sets suitable for non coherent detection

Review of detection techniques
Bibliography

General references:

- J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989, 2nd ed..
- S. Benedetto, E. Biglieri and V. Castellani. *Digital Transmission Theory.* Prentice-Hall, Englewood Cliffs, U.S.A., 1987.

Fading and phase-uncertain channel models:

- G. L. Stüber, *Principles of Mobile Communication*, Kluwer Academic Publishers, 1996.
- U. Mengali and A. N. D'Andrea, *Synchronization techniques for digital receivers.* New York: Plenum, 1997.

Statistical detection theory:

- J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: John Wiley & Sons, 1965.
- H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: John Wiley & Sons, 1968.

Systems with memory
Where does this memory come from?

Any practical system transmits by periodical repetitions of M-ary signaling acts ($\log_2 M$ bits/signaling period or *bits/channel use*)

In *memoryless* systems different signaling acts do not influence each other

In systems *with memory* the detection process may benefit from the observation of the received signal over "present," "past," and possibly "future" signaling periods

Memory arises if (e.g.):

– Channel coding is employed for error control

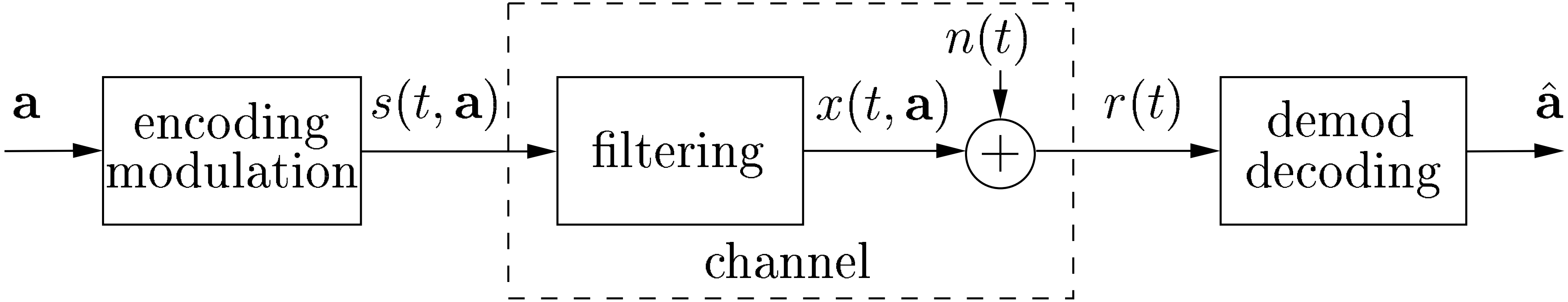
– The transmission channel is dispersive (Inter-Symbol Interference (ISI))

– The transmission channel includes stochastic parameters, such as a phase rotation or a complex fading weight

– The channel additive Gaussian noise is colored, i.e., its power spectral density is not constant

Systems with memory

General system model



Information sequence: $\mathbf{a} = \mathbf{a}_0^{K-1} = (a_{K-1}, \ldots, a_1, a_0)^T$

Transmitted signal: $s(t, \mathbf{a})$

Received signal: $r(t) = x(t, \mathbf{a}) + n(t)$

Notation: $\mathbf{x}_{k_1}^{k_2} = (x_{k_2}, \ldots, x_{k_1+1}, x_{k_1})^T$

Detection strategy? (*What is a message?*)

Sequence and symbol detection
What is a message?

MAP sequence detection

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, P\left[\mathbf{a}|r(t)\right] = \underset{\mathbf{a}}{\operatorname{argmax}}\, P\left(\mathbf{a}|\mathbf{r}\right)$$

MAP symbol detection

$$\hat{a}_k = \underset{a_k}{\operatorname{argmax}}\, P\left[a_k|r(t)\right] = \underset{a_k}{\operatorname{argmax}}\, P\left(a_k|\mathbf{r}\right)$$

$r(t)$ is observed over the entire information bearing interval $T_0 \supset (0, KT)$

Performance is similar and tends to be equal for high SNR

Complexity is different: sequence detection is less complex

Symbol APPs are the route to *iterative detection*

Discretization is the key to the computation of the APPs. One or more discrete observables per information symbol may be used

Causal systems
The viewpoint of detection

A system is causal if:

$$p(\mathbf{r}_0^k|\mathbf{a}) = p(\mathbf{r}_0^k|\mathbf{a}_0^k)$$

This property involves the cascade of encoder, modulator, channel, and signal discretizer

It is formulated in terms of statistical dependence of the discrete observable sequence on the information sequence

Any physical system is causal

MAP sequence detection
Computation of the APPs

Let $\mathbf{a} = \mathbf{a}_0^{K-1}$ and $\mathbf{r} = \mathbf{r}_0^{K-1}$

For a causal system, the APPs are:

$$
\begin{aligned}
P(\mathbf{a}|\mathbf{r}) \sim p(\mathbf{r}|\mathbf{a})P(\mathbf{a}) &= \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a})P(a_k) \\
&= \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^{k})P(a_k) \quad \text{(causality)} \\
&\sim \sum_{k=0}^{K-1} \underbrace{\left[\ln p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^{k}) + \ln P(a_k)\right]}_{\text{branch metrics}}
\end{aligned}
$$

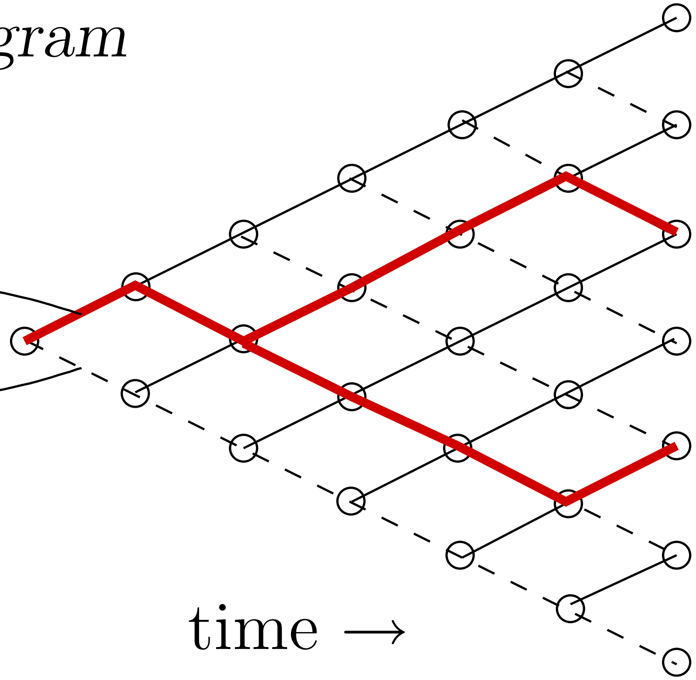
The *sequence metric* can be recursively computed in terms of *branch metrics*

Implementation requires a *tree search* (sequence $\leftrightarrow$ *path*)

Path-search algorithms

On a tree diagram

An example of binary tree:



Branch metric:

$$\ln p(r_k | \mathbf{r}_0^{k-1}, a_k, \mathbf{a}_0^{k-1}) + \ln P(a_k)$$

Branch metrics depend on the entire previous path history:

$\Rightarrow$ unlimited memory (complexity is exponential with K)

Tree reduced-search (approximate) algorithms:

– M-algorithm, T-algorithm (breadth-first)

– Fano-algorithm (single-stack algorithm) (depth-first)

– Jelinek-algorithm (stack algorithm) (metric-first)

Finite-memory causal systems
The viewpoint of detection

A system is causal and finite memory if:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}})$$

$\mathcal{C}$ is a suitable integer (*finite memory parameter*)

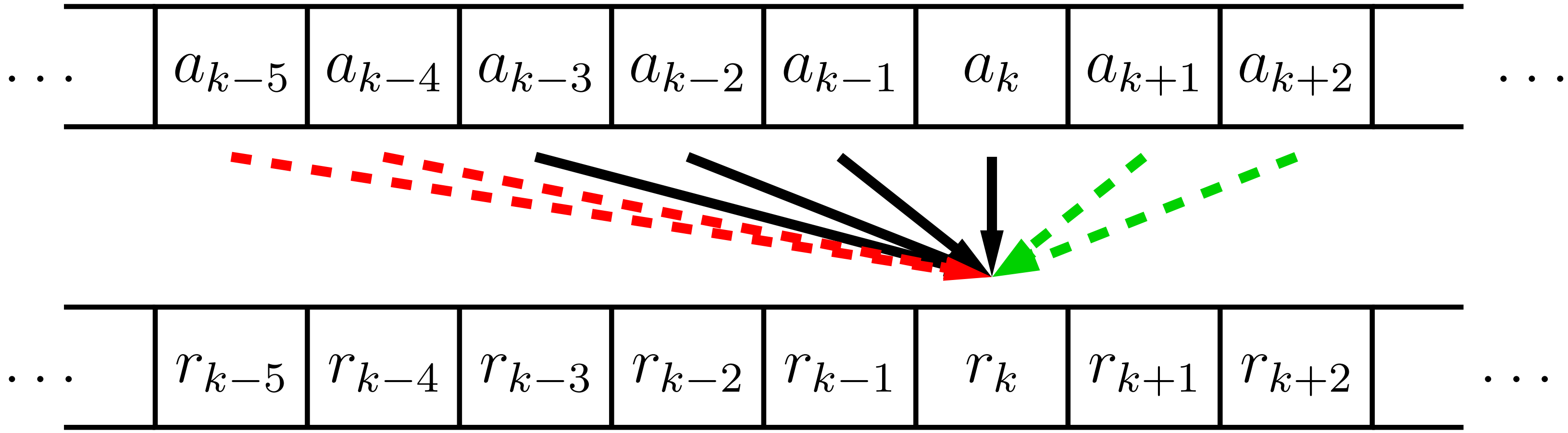
$\mu_{k-\mathcal{C}}$ is a suitable *state*, at epoch $k - \mathcal{C}$, of the encoder/modulator

In the computation of the APPs (or metrics), the system can be modeled as a Finite State Machine (FSM)

Minimal folding condition: the tree folds into a *trellis diagram*

Path search can be implemented efficiently

Causality and finite memory
A pictorial view



$\mathcal{C} = 3$

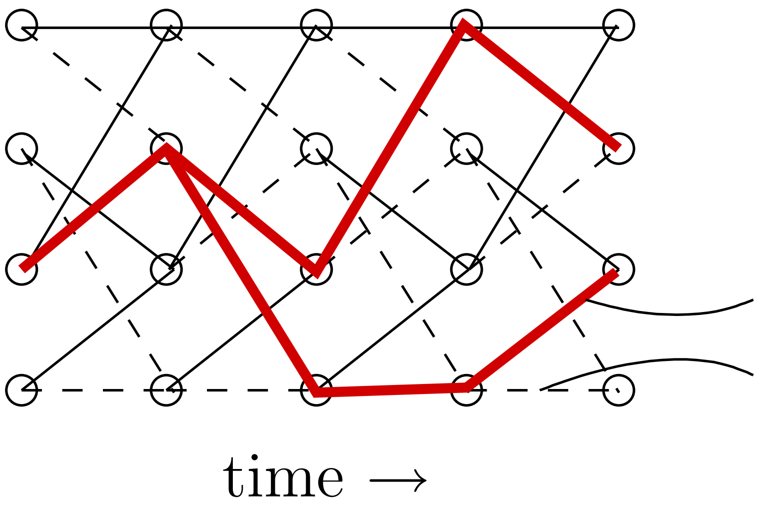
Not allowed because of finite memory

Not allowed because of causality

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-3}^k)$$

Path-search algorithms
On a trellis diagram

An example of binary trellis:



Branch metric:

$$\ln p(r_k|\mathbf{r}_0^{k-1}, a_k, \underbrace{\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}}}_{\sigma_k}) + \ln P(a_k)$$

Augmented trellis state:

$$\sigma_k = (\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})$$

Finite-memory branch metrics:

$$\gamma_k(a_k, \sigma_k) = \ln p(r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k) + \ln P(a_k)$$

Viterbi algorithm
Basic recursions

- Path metric:

$$\Gamma_k(\sigma_k) = \sum_{i=0}^{k} \gamma_i(a_i, \sigma_i) = \sum_{i=0}^{k} \left[\ln p(r_i | \mathbf{r}_0^{i-1}, a_i, \sigma_i) + \ln P(a_i) \right]$$

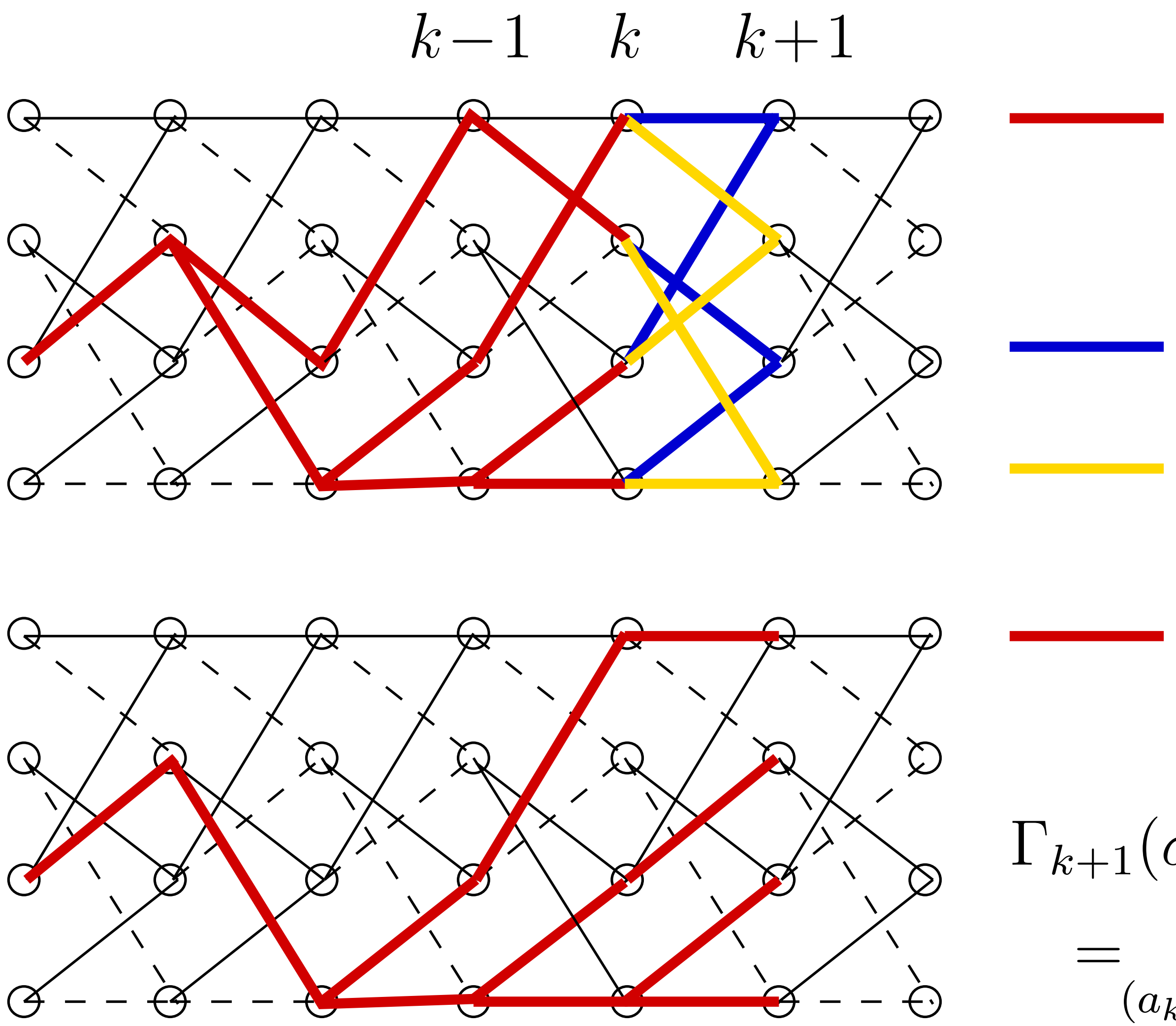
- Path metric update step (*Add-Compare-Select*):

$$\Gamma_{k+1}(\sigma_{k+1}) = \max_{(a_k, \sigma_k):\sigma_{k+1}} \left[\Gamma_k(\sigma_k) + \gamma_k(a_k, \sigma_k) \right]$$

- *Survivor* update step: the survivor of the maximizing state is extended by the label a_k of the *winning* branch

Viterbi algorithm

Add-Compare-Select: a pictorial view



$$\Gamma_{k+1}(\sigma_{k+1}) = \max_{(a_k,\sigma_k):\sigma_{k+1}} [\Gamma_k(\sigma_k) + \gamma_k(a_k,\sigma_k)]$$

MAP symbol detection

Computation of the APPs (1)

By (conditional) *marginalization*:

$$P(a_k|\mathbf{r}) = \sum_{\mathbf{a}_{k-\mathcal{C}}^{k-1}} \sum_{\mu_{k-\mathcal{C}}} P(\underbrace{a_k, \mathbf{a}_{k-\mathcal{C}}^{k-1}}_{\mathbf{a}_{k-\mathcal{C}}^{k}}, \mu_{k-\mathcal{C}}|\mathbf{r})$$

$$\sim \sum_{\mathbf{a}_{k-\mathcal{C}}^{k-1}} \sum_{\mu_{k-\mathcal{C}}} p(\mathbf{r}|\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) P(\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}})$$

By (conditional) *chain factorization*:

$$p(\mathbf{r}|\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) = p(\mathbf{r}_0^{k-1}, r_k, \mathbf{r}_{k+1}^{K-1}|\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}})$$

$$= p(\mathbf{r}_{k+1}^{K-1}|\underbrace{\mathbf{r}_0^{k-1}, r_k}_{\mathbf{r}_0^{k}}, \mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}})\, p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) \cdot p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}})$$

Three factors: *future given the past and present*, *present given the past*, and *past*, respectively

MAP symbol detection
Computation of the APPs (2)

By causality and finite memory, the first and third factors are:

$$p(\mathbf{r}_{k+1}^{K-1}|\mathbf{r}_0^k, \mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) = p(\mathbf{r}_{k+1}^{K-1}|\mathbf{r}_0^k, \mathbf{a}_{k-\mathcal{C}+1}^k, \mu_{k-\mathcal{C}+1})$$

$$p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) = p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})$$

By independence of the information symbols:

$$P(\mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) = P(a_k) P(\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})$$

MAP symbol detection

Computation of the APPs (3)

The APPs can be rearranged as:

$$P(a_k|\mathbf{r}) \sim \sum_{\mathbf{a}_{k-\mathcal{C}}^{k-1}} \sum_{\mu_{k-\mathcal{C}}} \underbrace{p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}}) P(\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})}_{\bar{\alpha}_k(\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})}$$

$$\cdot \underbrace{p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) P(a_k)}_{\bar{\gamma}_k(\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}})}$$

$$\cdot \underbrace{p(\mathbf{r}_{k+1}^{K-1}|\mathbf{r}_0^{k}, \mathbf{a}_{k-\mathcal{C}+1}^{k}, \mu_{k-\mathcal{C}+1})}_{\bar{\beta}_{k+1}(\mathbf{a}_{k-\mathcal{C}+1}^{k}, \mu_{k-\mathcal{C}+1})}$$

$$= \sum_{\mathbf{a}_{k-\mathcal{C}}^{k-1}} \sum_{\mu_{k-\mathcal{C}}} \bar{\alpha}_k(\underbrace{\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}}}_{\sigma_k}) \bar{\gamma}_k(\underbrace{\mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}}_{(a_k, \sigma_k)}) \bar{\beta}_{k+1}(\underbrace{\mathbf{a}_{k-\mathcal{C}+1}^{k}, \mu_{k-\mathcal{C}+1}}_{\sigma_{k+1}(a_k, \sigma_k)})$$

$$= \sum_{\sigma_k} \bar{\alpha}_k(\sigma_k)\, \bar{\gamma}_k(a_k, \sigma_k)\, \bar{\beta}_{k+1}[\sigma_{k+1}(a_k, \sigma_k)]$$

MAP symbol detection
The key quantities

Augmented trellis state:

$$\sigma_k = (\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})$$

Branch metrics (in the *metric* or *logarithmic domain*):

$$\gamma_k(a_k, \sigma_k) = \ln \bar{\gamma}_k(a_k, \sigma_k) = \ln p(r_k | \mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) + \ln P(a_k)$$

Exponential of branch metrics (in the *probability domain*):

$$\bar{\gamma}_k(a_k, \sigma_k) = e^{\gamma_k(a_k, \sigma_k)} = p(r_k | \mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^{k}, \mu_{k-\mathcal{C}}) P(a_k)$$

These are exactly the quantities introduced in MAP sequence detection

What about $\bar{\alpha}_k(\sigma_k)$ and $\bar{\beta}_{k+1}(\sigma_{k+1})$?

MAP symbol detection
Forward recursion

By *averaging*, chain factorization, and causality:

$$
\begin{aligned}
\bar{\alpha}_{k+1}(\sigma_{k+1}) &= p(\mathbf{r}_0^k|\mathbf{a}_{k-\mathcal{C}+1}^k, \mu_{k-\mathcal{C}+1}) P(\mathbf{a}_{k-\mathcal{C}+1}^k, \mu_{k-\mathcal{C}+1}) \\
&= \sum_{(a_{k-\mathcal{C}}, \mu_{k-\mathcal{C}}):\sigma_{k+1}} p(\mathbf{r}_0^k|\mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) P(\mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) \\
&= \sum_{(a_{k-\mathcal{C}}, \mu_{k-\mathcal{C}}):\sigma_{k+1}} \underbrace{p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}}) P(a_k)}_{\bar{\gamma}_k(a_k, \sigma_k)} \\
&\qquad \cdot \underbrace{\underbrace{p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^k, \mu_{k-\mathcal{C}})}_{p(\mathbf{r}_0^{k-1}|\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})} P(\mathbf{a}_{k-\mathcal{C}}^{k-1}, \mu_{k-\mathcal{C}})}_{\bar{\alpha}_k(\sigma_k)} \\
&= \sum_{(a_k, \sigma_k):\sigma_{k+1}} \bar{\gamma}_k(a_k, \sigma_k)\, \bar{\alpha}_k(\sigma_k)
\end{aligned}
$$

MAP symbol detection
Backward recursion

By averaging, independence of the information symbols, chain factorization, and finite memory:

$$\bar{\beta}_k(\sigma_k) = p(\mathbf{r}_k^{K-1}|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-C}^{k-1}, \mu_{k-C})$$

$$= \sum_{a_k} p(\mathbf{r}_k^{K-1}|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-C}^{k}, \mu_{k-C}) \underbrace{P(a_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-C}^{k-1}, \mu_{k-C})}_{P(a_k)}$$

$$= \sum_{a_k} \underbrace{\underbrace{p(\mathbf{r}_{k+1}^{K-1}|\mathbf{r}_0^{k}, \mathbf{a}_{k-C}^{k}, \mu_{k-C})}_{p(\mathbf{r}_{k+1}^{K-1}|\mathbf{r}_0^{k}, \mathbf{a}_{k-C+1}^{k}, \mu_{k-C+1})}}_{\bar{\beta}_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]} \underbrace{p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_{k-C}^{k}, \mu_{k-C})P(a_k)}_{\bar{\gamma}_k(a_k,\sigma_k)}$$

$$= \sum_{a_k} \bar{\beta}_{k+1}[\sigma_{k+1}(a_k, \sigma_k)]\, \bar{\gamma}_k(a_k, \sigma_k)$$

Forward-Backward (BCJR) algorithm
Basic recursions

- APPs:

$$P(a_k|\mathbf{r}) \sim \sum_{\sigma_k} \bar{\alpha}_k(\sigma_k)\,\bar{\gamma}_k(a_k,\sigma_k)\,\bar{\beta}_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]$$

- Forward recursion:

$$\bar{\alpha}_{k+1}(\sigma_{k+1}) = \sum_{(a_k,\sigma_k):\sigma_{k+1}} \bar{\gamma}_k(a_k,\sigma_k)\,\bar{\alpha}_k(\sigma_k)$$

- Backward recursion:

$$\bar{\beta}_k(\sigma_k) = \sum_{a_k} \bar{\beta}_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]\,\bar{\gamma}_k(a_k,\sigma_k)$$

- With suitable initialization

MAP symbol detection
Comparison with MAP sequence detection

Processing the "exponential metrics" $\bar{\gamma}_k(a_k, \sigma_k)$ in the FSM trellis diagram is sufficient (*again!*)

Sum-product algorithm (complexity is much larger than Viterbi)

The entire observation $\mathbf{r}_0^{K-1}$ must be processed before the APPs can be computed

Block processing (or approximations): latency delay

Memoryless systems

Sequence and symbol detection coincide

For *memoryless* systems, $\mathcal{C} = 0$ and the state variable μ_k vanishes:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = p(r_k|\mathbf{r}_0^{k-1}, a_k)$$

Sequence detection:

$$P(\mathbf{a}|\mathbf{r}) \sim p(\mathbf{r}|\mathbf{a})P(\mathbf{a}) = \prod_{k=0}^{K-1} \left[p(r_k|\mathbf{r}_0^{k-1}, a_k)P(a_k)\right]$$

Symbol detection:

$$P(a_k|\mathbf{r}) \sim \sum_{\mathbf{a}_0^{k-1}} \sum_{\mathbf{a}_{k+1}^{K-1}} p(\mathbf{r}|\mathbf{a})P(\mathbf{a}) = \sum_{\mathbf{a}_0^{k-1}} \sum_{\mathbf{a}_{k+1}^{K-1}} \prod_{i=0}^{K-1} p(r_i|\mathbf{r}_0^{i-1}, a_i)P(a_i)$$

$$= p(r_k|\mathbf{r}_0^{k-1}, a_k)P(a_k) \underbrace{\sum_{\mathbf{a}_0^{k-1}} \sum_{\mathbf{a}_{k+1}^{K-1}} \prod_{\substack{i=0 \\ i\neq k}}^{K-1} p(r_i|\mathbf{r}_0^{i-1}, a_i)P(a_i)}_{\text{independent of } a_k}$$

$$\Rightarrow \quad \hat{a}_k = \operatorname*{argmax}_{a_k} p(r_k|\mathbf{r}_0^{k-1}, a_k)P(a_k)$$

Forward-Backward (BCJR) algorithm
Max-log-MAP approximation: APPs

We could equivalently formulate the algorithm in the logarithmic (or metric) domain:

$$\begin{aligned} \ln P(a_k|\mathbf{r}) &\sim \ln \sum_{\sigma_k} \bar{\alpha}_k(\sigma_k)\,\bar{\gamma}_k(a_k,\sigma_k)\,\bar{\beta}_{k+1}[\sigma_{k+1}(a_k,\sigma_k)] \\ &= \ln \sum_{\sigma_k} e^{\alpha_k(\sigma_k)+\gamma_k(a_k,\sigma_k)+\beta_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]} \\ &\simeq \max_{\sigma_k} \{\alpha_k(\sigma_k)+\gamma_k(a_k,\sigma_k)+\beta_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]\} \end{aligned}$$

where $\alpha_k(\sigma_k) = \ln \bar{\alpha}_k(\sigma_k)$ and $\beta_{k+1}(\sigma_{k+1}) = \ln \bar{\beta}_{k+1}(\sigma_{k+1})$

For large $|x-y|$ $\Rightarrow$ $\ln(e^x + e^y) \simeq \max(x,y)$ and by extension:

$$\ln(e^{x_1} + e^{x_2} + \cdots + e^{x_n}) \simeq \max(x_1, x_2, \ldots, x_n)$$

Forward-Backward (BCJR) algorithm
Max-log-MAP approximation: FB recursions

$$\alpha_{k+1}(\sigma_{k+1}) = \ln \bar{\alpha}_{k+1}(\sigma_{k+1}) = \ln \sum_{(a_k,\sigma_k):\sigma_{k+1}} \bar{\gamma}_k(a_k,\sigma_k)\,\bar{\alpha}_k(\sigma_k)$$

$$\simeq \max_{(a_k,\sigma_k):\sigma_{k+1}} [\gamma_k(a_k,\sigma_k) + \alpha_k(\sigma_k)]$$

$$\beta_k(\sigma_k) = \ln \bar{\beta}_k(\sigma_k) = \ln \sum_{a_k} \bar{\beta}_{k+1}[\sigma_{k+1}(a_k,\sigma_k)]\,\bar{\gamma}_k(a_k,\sigma_k)$$

$$\simeq \max_{a_k} \{\beta_{k+1}[\sigma_{k+1}(a_k,\sigma_k)] + \gamma_k(a_k,\sigma_k)\}$$

Forward-Backward (BCJR) algorithm
Max-log-MAP approximation: key features

Forward and backward recursions can be implemented by two Viterbi algorithms running in *direct* and *inverse* time

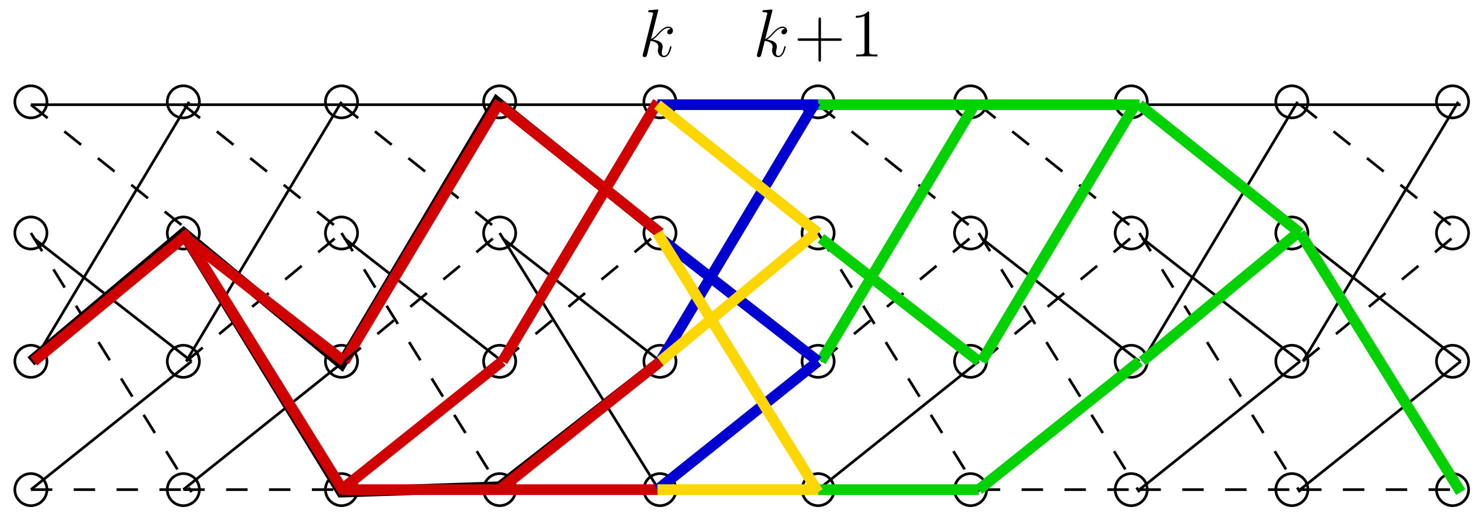
$\alpha_k(\sigma_k)$ and $\beta_k(\sigma_k)$ can be interpreted as forward and backward survivor metrics

The max-log-MAP algorithm is computationally efficient, at the cost of a slight degradation in performance

Various degrees of approximations have been studied (intermediate between the "full-complexity" forward-backward algorithm and the max-log-MAP approximation)

Forward-Backward (BCJR) algorithm

Max-log-MAP approximation: a pictorial view



Forward survivor metrics $\alpha_k(\sigma_k) = \ln \bar{\alpha}_k(\sigma_k)$

Backward survivor metrics $\beta_{k+1}(\sigma_{k+1}) = \ln \bar{\beta}_{k+1}(\sigma_{k+1})$

$$\max_{\sigma_k} \left\{ \alpha_k(\sigma_k) + \gamma_k(a_k, \sigma_k) + \beta_{k+1}[\sigma_{k+1}(a_k, \sigma_k)] \right\} \Big|_{a_k=+1}$$

$$\max_{\sigma_k} \left\{ \alpha_k(\sigma_k) + \gamma_k(a_k, \sigma_k) + \beta_{k+1}[\sigma_{k+1}(a_k, \sigma_k)] \right\} \Big|_{a_k=-1}$$

Summary of MAP detection

- For causal finite-memory systems:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = p(r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k)$$

- In the computation of the APPs, the system can be modeled as a Finite State Machine (FSM) with state σ_k. The underlying FSM model is *identical* for sequence and symbol detection

- Branch metrics (our focus in the following):

$$\gamma_k(a_k, \sigma_k) = \ln p(r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k) + \ln P(a_k)$$

- MAP sequence detection can be implemented efficiently by the Viterbi algorithm

- MAP symbol detection can be implemented by a sum-product forward-backward algorithm (complex)

- The max-log-MAP approximation of the forward-backward algorithm can be implemented efficiently by means of two Viterbi algorithms running in direct and inverse time

MAP sequence and symbol detection
Problem 3

Assuming a system is causal and finite memory:

A. Work out the derivation of the Viterbi algorithm for MAP sequence detection

B. Work out the derivation of the forward-backward algorithm for MAP symbol detection

Rederive the main recursions in each case

Max-log-MAP algorithm
Problem 4

Let

$$\max{*}\,(x_1, x_2, \ldots, x_n) = \ln(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$$

A. Show that

$$\max{*}\,(\max{*}\,(x_1, x_2), x_3) = \max{*}\,(x_1, x_2, x_3)$$

B. Show that

$$\max{*}\,(x_1, x_2) = \max(x_1, x_2) + \ln(1 + e^{-|x_1 - x_2|})$$

C. Show that the *exact* forward-backward algorithm can be formulated replacing the $\max(\cdot, \cdot)$ operator with $\max{*}\,(\cdot, \cdot)$ in the max-log-MAP approximation

D. Comment on the computational complexity of the exact formulation of the forward-backward algorithm in item C.

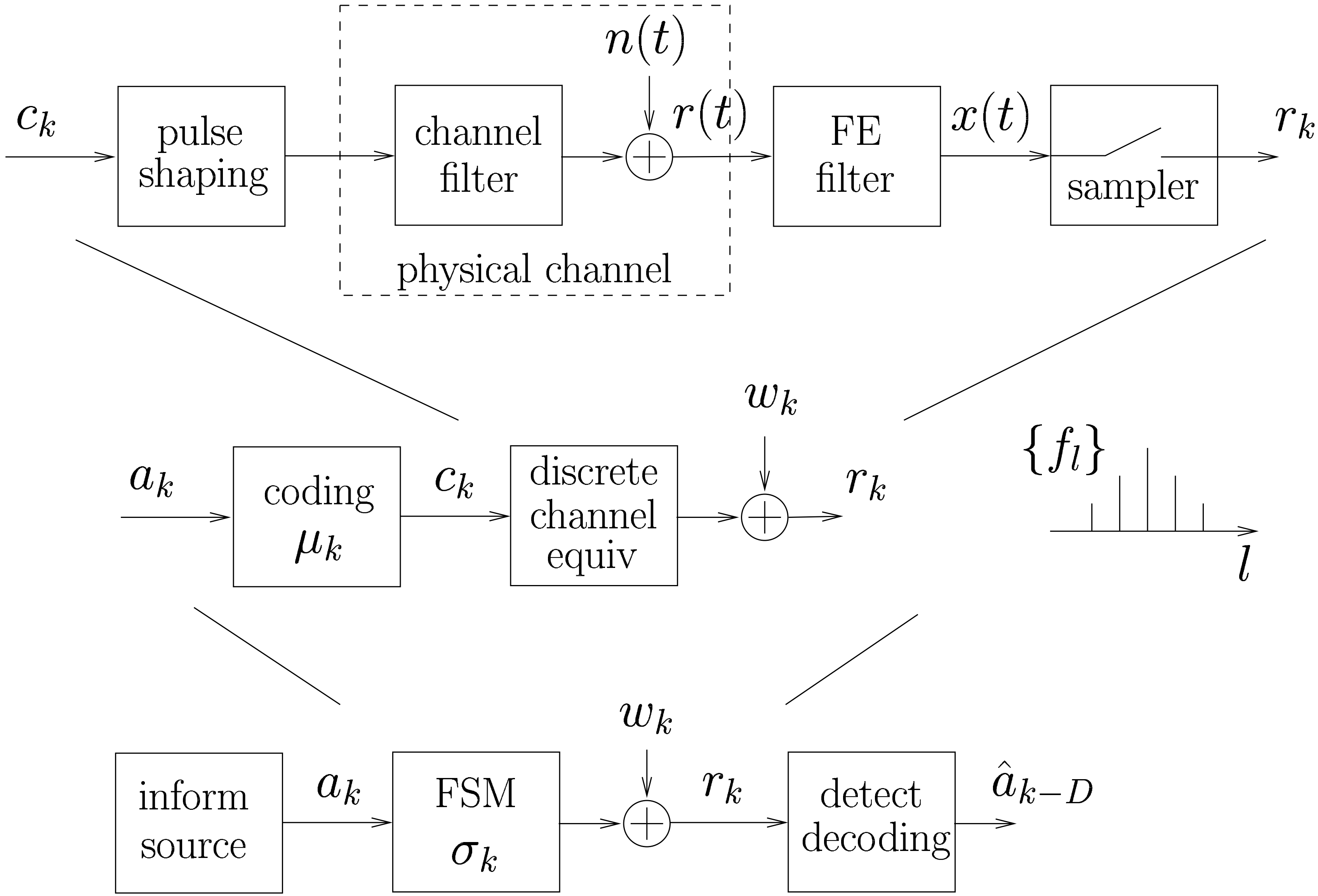
MAP detection for systems with memory
Examples of application

Linear modulation on static dispersive channel

Linear modulation on flat fading channel

Linear modulation on static dispersive channel

System overview



Linear modulation on static dispersive channel
System model

Model of discrete observable:

$$r_k = \sum_{l=0}^{L} f_l c_{k-l} + w_k$$

$\{f_l\}_{l=0}^{L}$: *white-noise discrete equivalent* of the ISI channel
$\{c_k\}$: code sequence
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

Coding rule:

$$\begin{cases} c_k = o(a_k, \mu_k) \\ \mu_{k+1} = t(a_k, \mu_k) \end{cases}$$

μ_k: encoder state

System state:

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

Linear modulation on static dispersive channel
Computation of the branch metrics

Conditional statistics of the observation:

$$\begin{aligned} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) &= p(r_k|\mathbf{a}_0^k) \quad \text{(conditionally independent observations)} \\ &= p(r_k|a_k, \sigma_k) = \frac{1}{\pi\sigma_w^2} \exp\left[-\frac{|r_k - x_k(a_k, \sigma_k)|^2}{\sigma_w^2}\right] \end{aligned}$$

$$x_k(a_k, \sigma_k) = \sum_{l=0}^{L} f_l c_{k-l}$$

Branch metrics:

$$\begin{aligned} \gamma_k(a_k, \sigma_k) &= \ln p(r_k|a_k, \sigma_k) + \ln P(a_k) \\ &\propto -|r_k - x_k(a_k, \sigma_k)|^2 + \sigma_w^2 \ln P(a_k) \\ &\propto \text{Re}\left[r_k x_k^*(a_k, \sigma_k)\right] - \frac{1}{2}|x_k(a_k, \sigma_k)|^2 + \frac{1}{2}\sigma_w^2 \ln P(a_k) \end{aligned}$$

$\propto$: proportionality plus a constant term

Linear modulation on static dispersive channel
Problem 5

Consider uncoded transmission of binary symbols $a_k \in \{\pm 1\}$ through a static dispersive channel with white-noise discrete equivalent

$$(f_0, f_1, f_2) = \frac{1}{\sqrt{6}}(1, 2, 1)$$

A. Define a suitable system state and draw the relevant trellis diagram

B. Express explicitly the branch metrics as a function of the received signal sample r_k for any possible transition

Assume the received sequence is

$$(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7) = (1.7, 1.2, 1.1, 0.3, -0.2, -1.1, 0.7, 0.4)$$

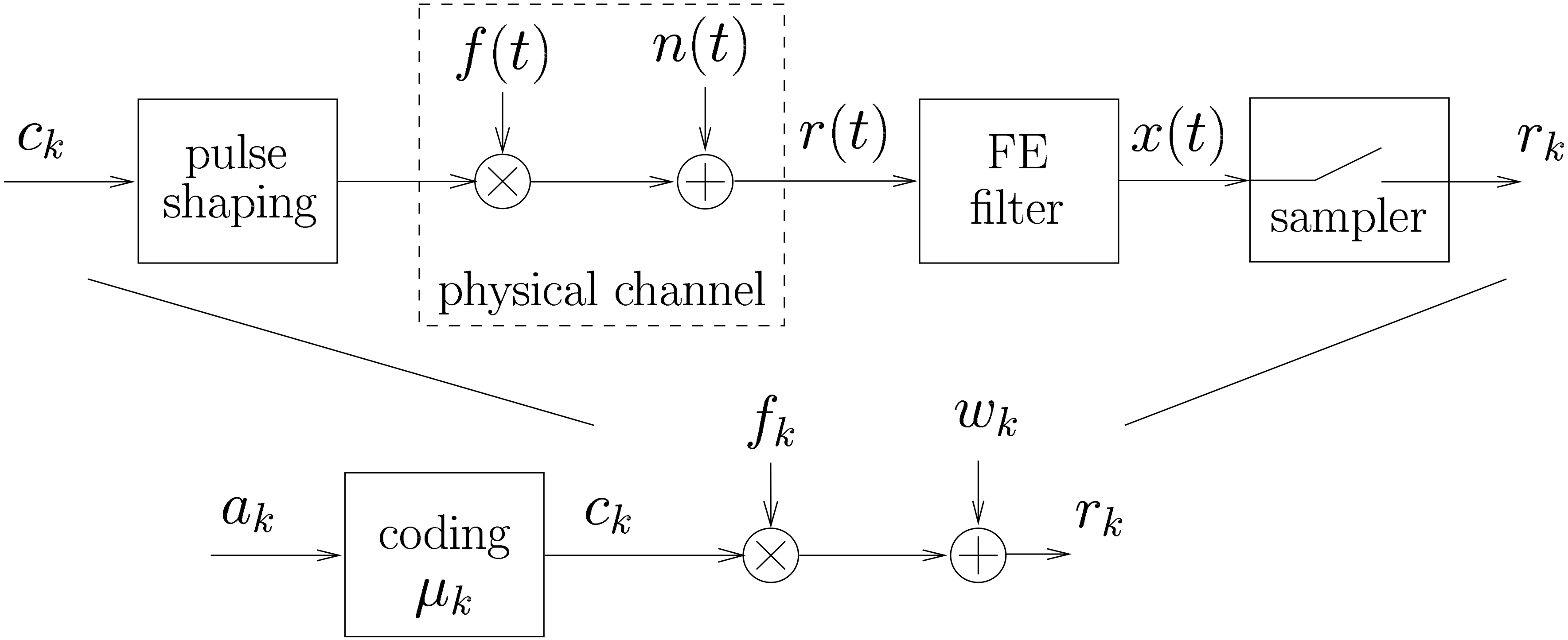
and the initial state is $\sigma_0 = (+1, +1)$

C. Use the Viterbi algorithm to detect the MAP sequence $\{\hat{a}_k^{(VA)}\}_{k=0}^{7}$

D. Use the max-log-MAP algorithm to approximately detect the sequence of MAP symbols $\{\hat{a}_k^{(FB)}\}_{k=0}^{7}$

Linear modulation on flat fading channel

System overview



Discretization provides a sufficient statistic if $f(t)$ is constant (i.e., a random variable). It is a good approximation if $f(t)$ varies very slowly (small Doppler bandwidth)

In general, one sample per signaling interval is not sufficient. *Oversampling*, e.g., two (or more) samples per symbol, provides a sufficient statistic

Linear modulation on flat fading channel
System model

Model of discrete observable:

$$r_k = f_k \, c_k + w_k$$

$\{f_k\}$: circular complex Gaussian random sequence
$\{c_k\}$: code sequence
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

Coding rule:

$$\begin{cases} c_k &= o(a_k, \mu_k) \\ \mu_{k+1} &= t(a_k, \mu_k) \end{cases}$$

μ_k: encoder state

Conditional statistics of the observation are Gaussian

Linear modulation on flat fading channel
Does a FSM model hold? (1)

Conditional statistics of the observation:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = \frac{1}{\pi\bar{\sigma}_k^2(\mathbf{a}_0^k)} \exp\left[-\frac{|r_k - \bar{r}_k(\mathbf{a}_0^k)|^2}{\bar{\sigma}_k^2(\mathbf{a}_0^k)}\right]$$

Conditional mean

$$\bar{r}_k(\mathbf{a}_0^k) = E\{r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\}$$

Conditional variance

$$\bar{\sigma}_k^2(\mathbf{a}_0^k) = E\{|r_k - \bar{r}_k(\mathbf{a}_0^k)|^2 \,|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\}$$

They depend on the fading autocovariance sequence

Linear modulation on flat fading channel
Does a FSM model hold? (2)

For Gaussian random variables, the conditional mean (i.e., the mean square estimate) is linear in the observation

$$\bar{r}_k(\mathbf{a}_0^k) = E\{r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\} = \sum_{i=1}^{k} p_i(\mathbf{a}_0^k)\, r_{k-i} \simeq c_k \sum_{i=1}^{k} p'_i \frac{r_{k-i}}{c_{k-i}}$$

The sequence-dependent linear prediction coefficients of the observation at time k can be approximated as

$$p_i(\mathbf{a}_0^k) \simeq c_k \frac{p'_i}{c_{k-i}} \quad \text{for high SNR}$$

where $\{p'_i\}_{i=1}^k$ are the linear prediction coefficients of the fading process

$$E\{f_k|\mathbf{f}_0^{k-1}\} = \sum_{i=1}^{k} p'_i f_{k-i}$$

The conditional mean depends on all the previous code symbols:

$\Rightarrow$ unlimited memory

Linear modulation on flat fading channel
Special case: slow fading

Constant fading (random variable): $f_k = f$

$$r_k = f\, c_k + w_k$$

Conditional mean:

$$\bar{r}_k(\mathbf{a}_0^k) = E\{r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\} \simeq c_k \frac{1}{k} \sum_{i=1}^{k} \frac{r_{k-i}}{c_{k-i}} \quad \text{for high SNR}$$

depends on all the previous code symbols ⇒ unlimited memory

Linear modulation on static dispersive channel
Problem 6

Consider the flat fading model

$$r_k = f_k\, c_k + w_k$$

with negligible noise power $\sigma_w^2 \simeq 0$

A. Show that the linear prediction coefficients of the observation and fading processes are related by

$$p_i(\mathbf{a}_0^k) \simeq c_k \frac{p'_i}{c_{k-i}} \quad i = 1, 2, \ldots, k$$

B. Show that for slow (constant) fading

$$p'_i \simeq \frac{1}{k}$$

Hint: *Use the fading model in the conditional mean*

Review of detection techniques
Bibliography

MAP sequence and symbol detection:

– G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, pp. 363–378, May 1972.

– L. R. Bahl, J. Cocke, F. Jelinek, and R. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*,, pp. 284-284, March 1974.

– J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989, 2nd ed..

– S. Benedetto, E. Biglieri and V. Castellani. *Digital Transmission Theory*. Prentice-Hall, Englewood Cliffs, U.S.A., 1987.

– K. M. Chugg, A. Anastasopoulos, X. Chen, *Iterative Detection: Adaptivity, Complexity Reduction, and Applications*. Kluwer Academic Publishers, 2001.

– G. Ferrari, G. Colavolpe, R. Raheli, *Detection Algorithms for Wireless Communications, with Applications to Wired and Storage Systems*, John Wiley & Sons, London, (August) 2004.

– G. Ferrari, G. Colavolpe, R. Raheli, "A unified framework for finite-memory detection," March 2004.

2. Detection under parametric uncertainty

Outline

Detection for systems with unlimited memory
Preliminaries

Channel models described in terms of stochastic parameters (even time-invariant) yield systems with *unlimited memory*

Optimal sequence or symbol detection algorithms can be exactly implemented only by resorting to some type of *exhaustive search* accounting for all possible transmission acts

Implementation complexity increases exponentially with the *length of transmission*, i.e., the number of transmitted information symbols K

Optimal detection is implementable only for *very limited* transmission lengths (not of practical interest, even for packet transmissions: e.g., $M^K = 4^8 = 2^{16} = 65536$)

⇒ Design suitable *approximate*, hence *suboptimal*, detection algorithms

Estimation-Detection decomposition
A suboptimal solution

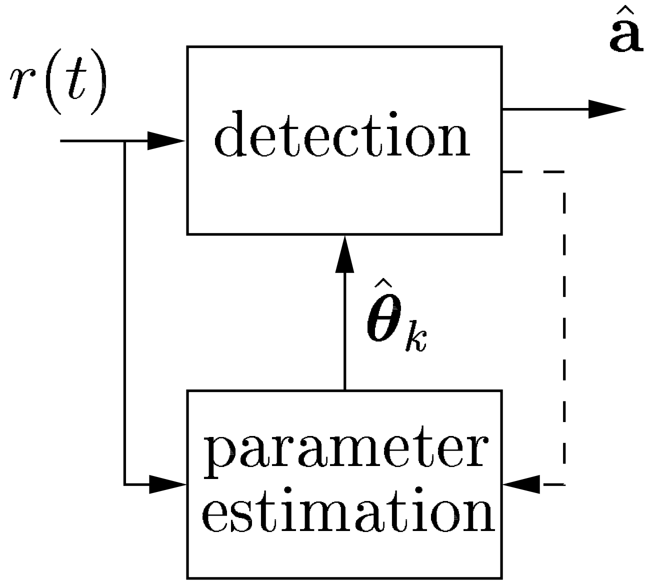
Idea of "decomposing" the functions of data detection and parameter estimation:

1. Derive the detection algorithms under the assumption of knowledge, to a certain degree of accuracy, of some (channel) parameters
2. Devise an estimation algorithm for extracting information about these parameters

This approach is viable alternative if a statistical characterization of the parameter is not available or not usable because of constrained implementation complexity

A statistical characterization is not available if static (or slow varying) parameters are modeled as unknown *deterministic* quantities

Estimation-Detection decomposition
System model



$\hat{\boldsymbol{\theta}}_k$ denotes an estimate of a parameter vector $\boldsymbol{\theta}_k$, at the k-th time-discrete instant

The estimation process observes explicitly the received signal $r(t)$ and possibly the detected data sequence $\hat{\mathbf{a}}$

Estimation-Detection decomposition
Some remarks

Conceptual advantage of decoupling the detection and estimation problems

Implementation advantage of physically simplifying the receiver

This decomposition has been used for decades, e.g., in *synchronization*, i.e., estimation of timing epoch, carrier phase or carrier frequency (of interest in virtually any passband communication system)

Logical ad-hoc solution: no claim of optimality can be made, in general. Optimality, i.e., minimal error probability, can only be attained if the statistical information about the parameter is *known* and *exploited* directly in the detection process.

Time-varying parameters can be viewed as static in the detection process. Their time variations must be *tracked* by the estimation function, provided they are slow. Rate of variation is critical

Parameter-conditional finite memory

A conceptual framework for Estimation-Detection decomposition

Assume information lossless discretization with $\mathbf{a} = \mathbf{a}_0^{K-1}$ and $\mathbf{r} = \mathbf{r}_0^{K-1}$ (for time-varying parameters, more samples per symbol may be necessary)

Let us collect some undesired (or *nuisance*) parameters into a vector $\boldsymbol{\theta}_k$, in general time-varying

The observation conditional statistics obey:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k, \boldsymbol{\theta}_k) = p(r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k, \boldsymbol{\theta}_k)$$

where the system state is

$$\sigma_k = (\mathbf{a}_{k-\mathcal{C}}^{k-1}; \mu_{k-\mathcal{C}})$$

$\mathcal{C}$ is the *residual* channel memory (i.e., assuming knowledge of the parameter)

The system can be modeled as a Finite State Machine (FSM) *conditionally* upon the parameter

Parameter-conditional finite memory

An example: Linear modulation on flat fading channel

Model of discrete observable:

$$r_k = f_k\, c_k + w_k$$

$\{f_k\}$: circular complex Gaussian random sequence
$\{c_k\}$: code sequence
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

The system is not finite memory

$\Rightarrow$ Viewing f_k as a nuisance parameter:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k, f_k) = p(r_k|c_k, f_k) = \frac{1}{\pi\sigma_w^2} \exp\left[-\frac{|r_k - f_k c_k|^2}{\sigma_w^2}\right]$$

The system is conditionally finite memory because the code symbols are the output of a finite state machine

For an uncoded system ($c_k = a_k$), the observation is conditionally memoryless

Parameter-conditional finite memory
Some remarks

- By a *clever* choice of the nuisance parameters, it is possible to transform the transmission system into *conditionally* finite-memory.
- This property holds conditionally on the undesired parameters; hence, only if they are known. It is the route to a decomposed estimation-detection design
- One can assume that some undesired parameters are known in devising the detection algorithms, thus avoiding intractable complexity, and devote some implementation complexity to the estimation of these undesired parameters.
- The parameter-conditional finite memory property suggests to view the presence of stochastic or unknown deterministic parameters as *parametric uncertainty* affecting the detection problem.

Parameter estimation
The "dual" problem

The parameter estimation problem can be viewed as the "dual" of the detection problem.

The "undesired" parameters become parameters of interest, whereas the "parameters" of interest in the detection process, namely the data symbols, are now just nuisance (or undesired) parameters.

Like the knowledge of the undesired parameters simplifies the detection problem, the possible knowledge of the data sequence may facilitate the estimation of the nuisance parameters.

An exact knowledge of the data symbols may reduce the "degree of randomness" of the received signal and facilitate the estimation of the parameters of interest

Parameter estimation

Data-aided parameter estimation

In coded systems the transmitted signal is modulated by the code sequence: knowledge of this sequence may be helpful in parameter estimation

We assume the code sequence is a data sequence *aiding* the parameter estimation process.

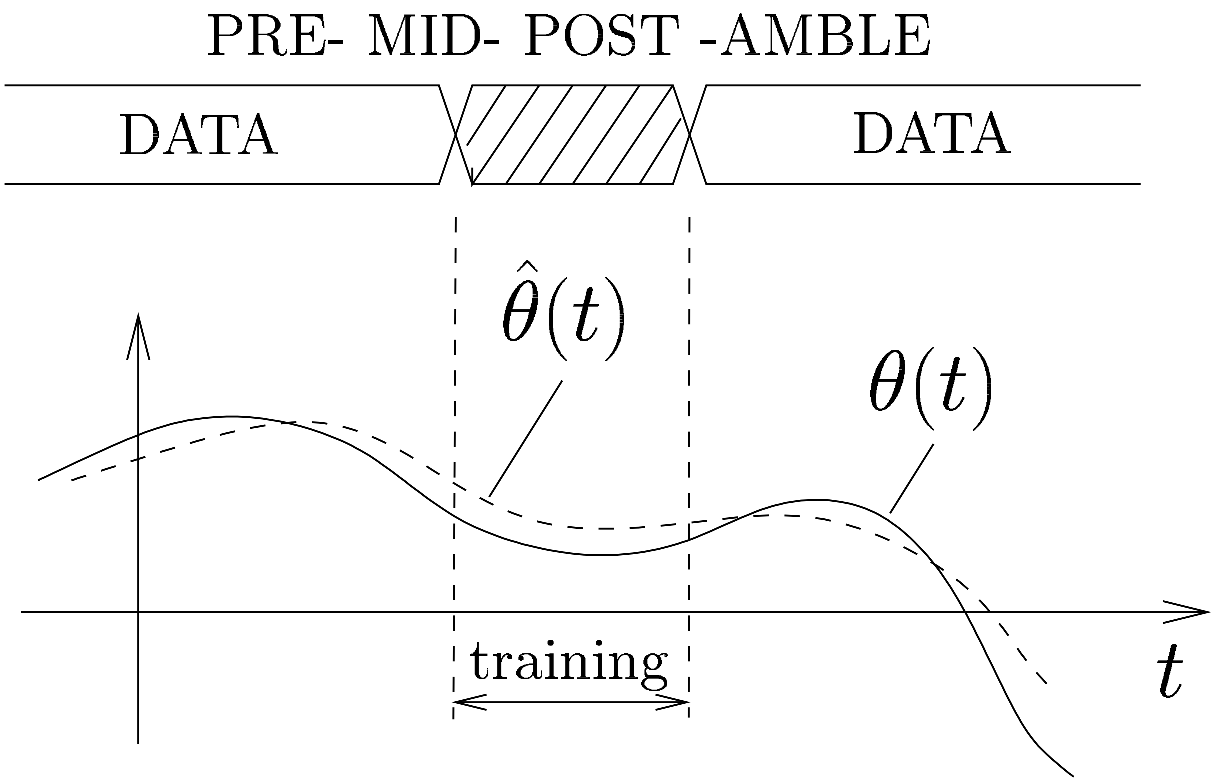
The data sequence is known during the (initial) *training mode*: preamble, midamble, postamble

During the transmission of the training sequence there is no transfer of information: training must be limited

In long term *tracking* of the channel parameters, *detected* data can be used: *decision-directed* estimation

Non-data-aided estimation is more complex: requires averaging over the data

Parameter estimation
A pictorial example



Data-aided parameter estimation
A general formulation

Formal definition of a causal data-aided estimator of parameter $\boldsymbol{\theta}_k$:

$$\hat{\boldsymbol{\theta}}_k = \mathbf{g}_{k-l}\left(\mathbf{r}_0^k, \mathbf{c}_0^{k-d}\right)$$

$\mathbf{g}(\cdot)$ denotes the functional dependence on the observation $\mathbf{r}_0^k$ and the aiding data sequence $\mathbf{c}_0^{k-d}$

Causality upon the observation: at time k, only $\mathbf{r}_0^k$ is observable

Causality upon the data-aiding sequence:

At time k, the most recent available data symbol is c_{k-d}, where d is the decision delay

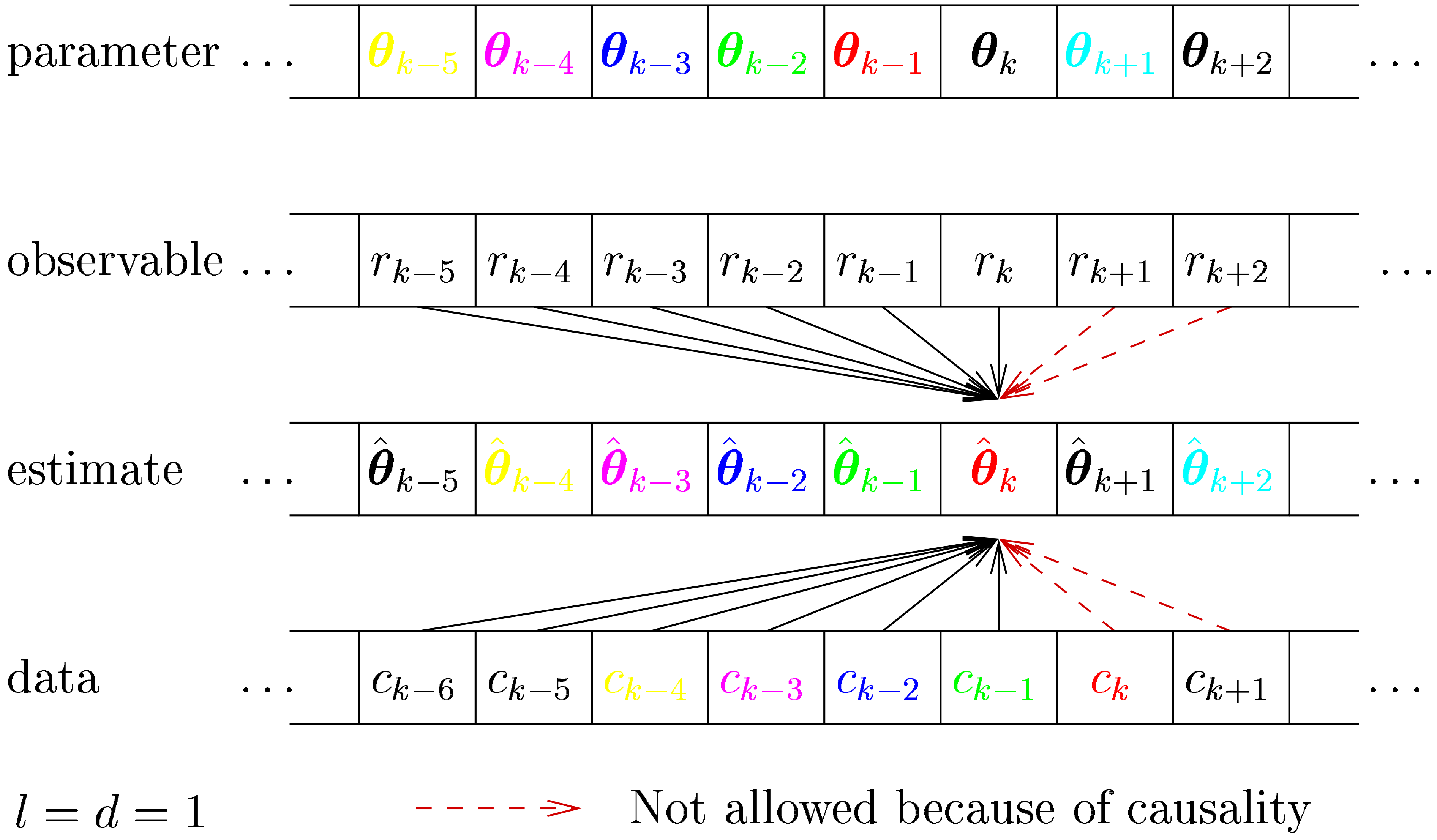
In decision-directed mode causality requires $d \geq 1$

l denotes the estimation delay ($\hat{\boldsymbol{\theta}}_k \simeq \boldsymbol{\theta}_{k-l}$)

$l - d$ is the estimation lag ($l < d$: prediction)

Causal data-aided parameter estimation

A pictorial view



Feedforward parameter estimation
Open-loop processing

The aiding data sequence is assumed ideally known (training)

Feedforward data-aided parameter estimator:

$$\hat{\boldsymbol{\theta}}_k = \mathbf{p}\left(\mathbf{r}_{k-\nu}^{k}, \mathbf{c}_0^{k-d}\right)$$

Explicit function of the $\nu + 1$ most recent signal observations (and the aiding data sequence)

Feedforward processing of the discrete observable (if linear: FIR filter)

The "loop" is "open" because the previous parameter estimates (*not the data!*) are not used in the current estimation

Feedback parameter estimation
Closed-loop processing

The aiding data sequence is assumed ideally known (training)

Feedback data-aided parameter estimator:

$$\hat{\boldsymbol{\theta}}_k = \mathbf{q}\left(\hat{\boldsymbol{\theta}}_{k-\xi}^{k-1}, \mathbf{r}_{k-\nu}^{k}, \mathbf{c}_0^{k-d}\right)$$

Explicit function of ξ previous estimates and $\nu+1$ most recent signal observations (and the aiding data sequence)

Feedback processing of the previous parameter estimates and feedforward processing of the discrete observable (if linear: IIR filter)

The “loop“ is “closed“ because the previous parameter estimates (*not the data!*) are used in the current estimation

$\xi = 1, 2$: first, second order loops (most typical)

Joint detection and estimation

Combination of detection and estimation functions

Define the branch metrics on the basis of the parameter-conditional finite memory p.d.f., with the true parameter vector $\boldsymbol{\theta}_k$ replaced by its estimate $\hat{\boldsymbol{\theta}}_k$:

$$\gamma_k(a_k, \sigma_k) = \ln p(r_k | \mathbf{r}_0^{k-1}, a_k, \sigma_k, \hat{\boldsymbol{\theta}}_k) + \ln P(a_k)$$

Proper definition of trellis state σ_k is necessary

The parameter estimate is obtained using a data-aided estimator:

$$\hat{\boldsymbol{\theta}}_k = \mathbf{g}_{k-l}\left(\mathbf{r}_0^k, \mathbf{c}_0^{k-d}\right)$$

⇒ Which code sequence $\mathbf{c}$ can be used for parameter estimation?

Joint detection and estimation
Final versus preliminary decisions

During *training* the data sequence is readily available

Tracking can be based on previous data decisions: *decision-directed mode*

The detection scheme outputs data decisions with a delay D

E.g., detection delay of the Viterbi algorithm (survivor merge)

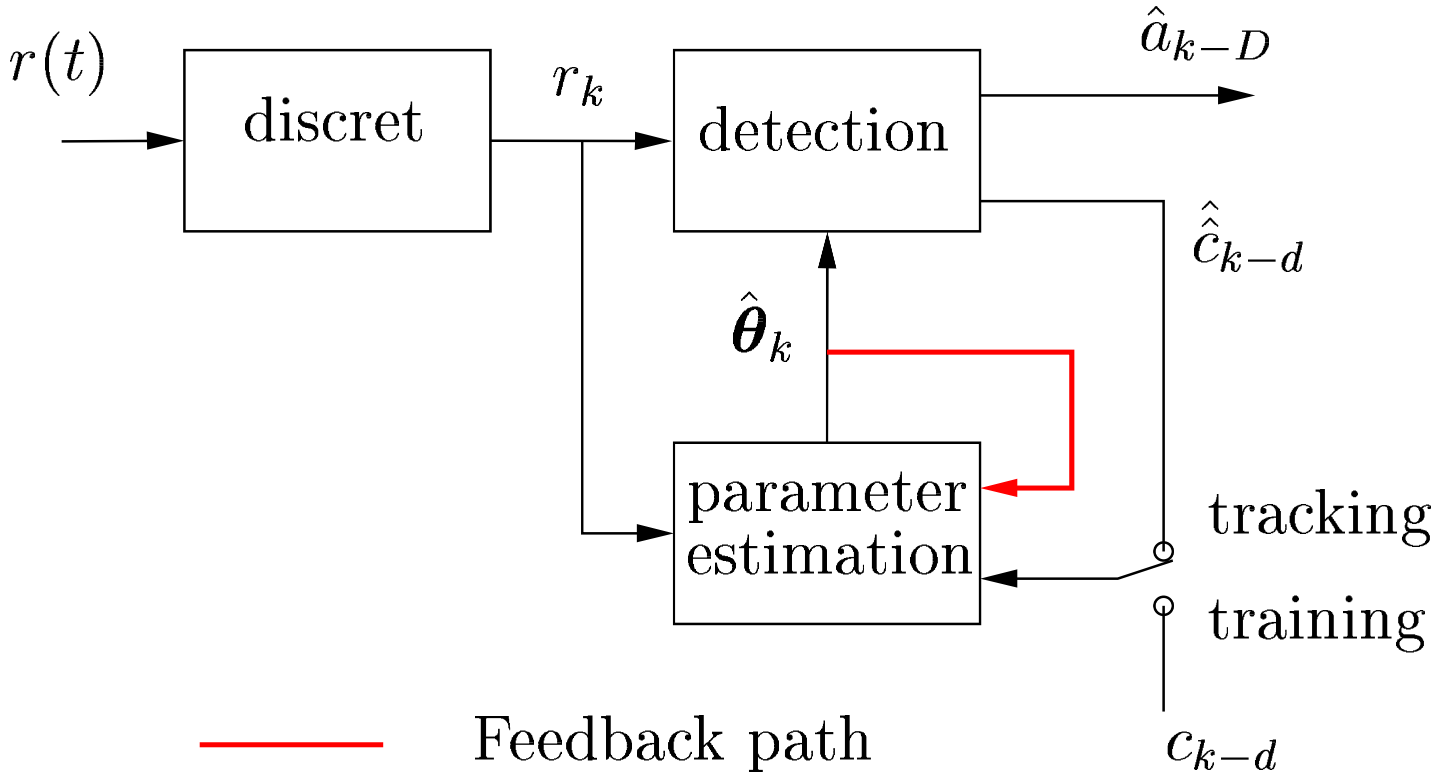
E.g., processing delay of the forward-backward algorithm (possible latency due to the packet duration)

The detection delay of the sequence aiding in parameter estimation should be small because it directly carries over to a delay in the parameter estimate:

Preliminary or *tentative* decisions with delay $d < D$

Joint detection and estimation

System model



D : "final" decision delay

d : "preliminary" or "tentative" decision delay

$\{\hat{\hat{c}}_{k-d}\}$: sequence of tentative decisions

Joint detection and estimation
Summary

- Branch metrics:

$$\gamma_k(a_k, \sigma_k) = \ln p(r_k | \mathbf{r}_0^{k-1}, a_k, \sigma_k, \hat{\boldsymbol{\theta}}_k) + \ln P(a_k)$$

- Data-aided parameter estimator:

$$\hat{\boldsymbol{\theta}}_k = \begin{cases} \mathbf{g}_{k-l}\left(\mathbf{r}_0^k, \mathbf{c}_0^{k-d}\right) & \text{training} \\ \mathbf{g}_{k-l}\left(\mathbf{r}_0^k, \hat{\hat{\mathbf{c}}}_0^{k-d}\right) & \text{tracking} \end{cases}$$

- In the tracking mode, preliminary decisions are used

Detection under parametric uncertainty
Examples of application

Linear modulation on phase-uncertain channel

Linear modulation on dispersive fading channel

Linear modulation on phase-uncertain channel
Synchronization-Detection decomposition

Model of discrete observable (usual notation):

$$r_k = e^{j\theta_k} c_k + w_k$$

θ_k : channel-induced phase rotation
$\{c_k\}$: code sequence with FSM model of state μ_k
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

Unlimited memory (observation not even conditionally Gaussian)

Considering θ_k as undesired, the parameter-conditional finite-memory property is verified:

$$\begin{aligned} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k, \theta_k) &= p(r_k|a_k, \mu_k, \theta_k) \\ &= \frac{1}{\pi\sigma_w^2} \exp\left[-\frac{|r_k - e^{j\theta_k} c_k(a_k, \mu_k)|^2}{\sigma_w^2}\right] \end{aligned}$$

$c_k(a_k, \mu_k)$: code symbol branch label

Linear modulation on phase-uncertain channel
Feedback phase synchronization

A data-aided phase estimate $\hat{\theta}_k$ can be obtained through a first order Phase-Locked Loop (PLL), where η controls the loop bandwidth:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta \, \mathrm{Im}\left\{\bar{r}_{k+1-d}\, c^*_{k+1-d}\right\}$$

$\bar{r}_k = r_k\, e^{-j\hat{\theta}_k}$: phase-synchronized observation

The estimated phase is inherently delayed by d instants

In the *training mode*, d can be chosen arbitrarily, except for the causality condition upon the observation which imposes $d \geq 0$. $d = 0$ is convenient to minimize the estimation delay

In the *decision-directed tracking mode*:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta \, \mathrm{Im}\left\{\bar{r}_{k+1-d}\, \hat{c}^*_{k+1-d}\right\}$$

The tentative decision delay must comply with the causality condition upon the detected data, which implies $d \geq 1$.

Linear modulation on phase-uncertain channel
Joint detection and synchronization

The estimated phase can be used in place of the true unknown phase in the computation of the branch metrics:

$$\begin{aligned}\gamma_k(a_k,\mu_k) &= \ln p(r_k|a_k,\mu_k,\hat{\theta}_k) + \ln P(a_k) \\ &\propto -|r_k - e^{j\hat{\theta}_k} c_k(a_k,\mu_k)|^2 + \sigma_w^2 \ln P(a_k) \\ &= -|\bar{r}_k - c_k(a_k,\mu_k)|^2 + \sigma_w^2 \ln P(a_k)\end{aligned}$$

The detection and synchronization functions can be based on the phase-synchronized observation

$$\bar{r}_k = r_k\, e^{-j\hat{\theta}_k}$$

Linear modulation on phase-uncertain channel
Problem 7

Consider the model of discrete observable

$$r_k = e^{j\theta} c_k + w_k$$

where θ is the overall phase rotation induced by the channel. Let $\hat{\theta}$ be a phase estimate and define the phase-synchronized observation $\bar{r}_k = r_k e^{-j\hat{\theta}}$

A. Derive an explicit expression of the mean square error (MSE) $E\{|\bar{r}_k - c_k|^2\}$ as a function of $\hat{\theta}$

B. Obtain an estimate of θ minimizing the MSE

C. Formulate a data-aided iterative stochastic gradient algorithm for minimizing the MSE

D. Comment on the functional relationship of the obtained synchronization scheme with a first-order PLL

Hint: *Define a stochastic gradient by differentiating the MSE with respect to* $\hat{\theta}$ *and discarding the expectation*

Linear modulation on dispersive fading channel
System model

Model of discrete observable (slow fading):

$$r_k = \sum_{l=0}^{L} f_{l,k}\, c_{k-l} + w_k = \mathbf{f}_k^T \mathbf{c}_k + w_k$$

$\mathbf{f}_k = (f_{0,k}, f_{1,k}, \ldots, f_{L,k})^T$: overall time-varying discrete equivalent impulse response at the k-th instant, circular complex Gaussian random vector
$\mathbf{c}_k = (c_k, c_{k-1}, \ldots, c_{k-L})^T$: code sequence with FSM model of state μ_k

Unlimited memory (observation is conditionally Gaussian)

Linear modulation on dispersive fading channel
Estimation-Detection decomposition

Considering $\mathbf{f}_k$ as undesired, the system is parameter-conditionally finite-memory:

$$\begin{aligned}\gamma_k(a_k,\sigma_k) &= \ln p(r_k|\mathbf{r}_0^{k-1},\mathbf{a}_0^k,\mathbf{f}_k) + \ln P(a_k)\\ &= \ln p(r_k|a_k,\sigma_k,\mathbf{f}_k) + \ln P(a_k)\\ &= \ln\left\{\frac{1}{\pi\sigma_w^2}\exp\left[-\frac{|r_k - \mathbf{f}_k^T\mathbf{c}_k(a_k,\sigma_k)|^2}{\sigma_w^2}\right]\right\} + \ln P(a_k)\\ &\propto -|r_k - \mathbf{f}_k^T\mathbf{c}_k(a_k,\sigma_k)|^2 + \sigma_w^2\ln P(a_k)\end{aligned}$$

$\sigma_k = (a_{k-1},a_{k-2},\ldots,a_{k-L};\mu_{k-L})$: system state

$\mathbf{c}_k(a_k,\sigma_k) = [c_k(a_k,\mu_k),c_{k-1}(a_{k-1},\mu_{k-1}),\ldots,c_{k-L}(a_{k-L},\mu_{k-L})]^T$: code symbol vector uniquely associated with the considered trellis branch (a_k,σ_k), in accordance with the coding rule

Linear modulation on dispersive fading channel
Feedback parameter estimation

Least Mean Squares (LMS) adaptive identification:

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \beta \left(r_{k+1-d} - \hat{\mathbf{f}}_k^T \mathbf{c}_{k+1-d} \right) \mathbf{c}^*_{k+1-d}$$

β compromises between adaptation speed and algorithm stability

During the decision-directed tracking mode:

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \beta \left(r_{k+1-d} - \hat{\mathbf{f}}_k^T \hat{\hat{\mathbf{c}}}_{k+1-d} \right) \hat{\hat{\mathbf{c}}}^*_{k+1-d}$$

$$\hat{\hat{\mathbf{c}}}_{k+1-d} = \left(\hat{\hat{c}}_{k+1-d}, \hat{\hat{c}}_{k-d}, \ldots, \hat{\hat{c}}_{k+1-d-L} \right)^T$$

$d \geq 1$ to comply with the causality condition upon the data

Branch metrics:

$$\gamma_k(a_k, \sigma_k) \propto -|r_k - \hat{\mathbf{f}}_k^T \mathbf{c}_k(a_k, \sigma_k)|^2 + \sigma_w^2 \ln P(a_k)$$

Linear modulation on dispersive fading channel
Problem 8

Consider the model of discrete observable

$$r_k = \mathbf{f}^T \mathbf{c}_k + w_k$$

where $\mathbf{f}$ is the overall discrete equivalent channel impulse response. Let $\hat{\mathbf{f}}$ be an estimate of the channel response. Assume the code symbols are zero-mean and uncorrelated

A. Derive an explicit expression of the mean square error (MSE) $E\{|r_k - \hat{\mathbf{f}}^T \mathbf{c}_k|^2\}$ as a function of $\hat{\mathbf{f}}$

B. Formulate a data-aided iterative stochastic gradient algorithm for minimizing the MSE

C. Comment on the functional relationship of the obtained identification scheme and the LMS algorithm

Hint: *Define a stochastic gradient by differentiating the MSE with respect to* $\hat{\mathbf{f}}$ *and discarding the expectation*

Joint detection and estimation
Error propagation

A decision-feedback mechanism characterizes the decision-directed tracking phase: decisions are used for parameter estimation and, hence, for detecting the successive data

Error propagation may take place, namely wrong data decisions may negatively affect the parameter estimate and cause further decision errors

This effect is usually non catastrophic but it affects the overall performance

Joint detection and estimation

Optimization of the tentative decision delay

Preliminary decisions with delay $d < D$ can be considerably worse than the final decisions. E.g., in the Viterbi algorithm, decisions with reduced delay $d < D$ are affected by the probability of unmerged survivors

⇒ Large values of tentative decision delay d may be best

The delay d of the aiding data sequence yields a delay in the parameter estimate which may affect the detection quality when the true parameter is time-varying

⇒ Small values of d may be best, possibly the minimal value $d = 1$.

Good values of tentative decision delay d must be the result of a trade-off between two conflicting requirements

⇒ In practice, one would have to experiment several values of d and select a good compromise value (minimize error propagation)

Detection under parametric uncertainty
Bibliography

– H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: John Wiley & Sons, 1968.

– H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems," *IEEE Trans. Commun.*, pp. 268-280, June 1971.

– F. R. Magee, Jr. and J. G. Proakis, "Adaptive maximum-likelihood sequence estimation for digital signaling in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, pp. 120–124, Jan. 1973.

– S. U. H. Qureshi and E. E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels," *IEEE Trans. Inform. Theory*, pp. 448-457, July 1973.

– G. Ungerboeck, "Adaptive maximum-likelihood receiver for carrier modulated data-transmission systems," *IEEE Trans. Commun.*, pp. 624-636, May 1974.

– G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, pp. 55-67, Jan. 1982.

– S. Benedetto, E. Biglieri and V. Castellani. *Digital Transmission Theory.* Prentice-Hall, Englewood Cliffs, U.S.A., 1987.

– J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989, 2nd ed..

– R. Raheli, A. Polydoros, C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.

3. Per-Survivor Processing: concept

Outline

1. Review of detection techniques
2. Detection under parametric uncertainty
3. Per-Survivor Processing (PSP): concept and historical review
4. Classical applications of PSP:
    4.1 Complexity reduction
    4.2 Linear predictive detection for fading channels
    4.3 Adaptive detection
5. Advanced applications of PSP

Per-Survivor Processing (PSP)

A step toward a unification of estimation and detection

The Estimation-Detection decomposition is a general suboptimal design approach to *force* a finite-memory property and achieve feasible detection complexity

Optimal processing is not compatible with the decomposition approach and would require a *unified* detection function (often of infeasible complexity)

Per-Survivor Processing is an alternative general design approach which still exploits the forced finite-memory property but *reduces the degree of separation* between the detection and estimation functions

In this technique, the code sequences associated with each survivor are used as the aiding data sequence for a *set* of *per-survivor* estimators of the unknown parameters

Per-Survivor Processing
A formal description

Let σ_k be the trellis state descriptive of the overall parameter-conditional finite state machine which models the transmission system

Let $\breve{\mathbf{c}}_0^{k-1}(\sigma_k)$ denote the code sequence associated with the survivor of state σ_k

Per-survivor estimates of the parameter vector $\boldsymbol{\theta}_k$ based on a data-aided estimator can be expressed as

$$\breve{\boldsymbol{\theta}}_k(\sigma_k) = \mathbf{g}_{k-l}\left[\mathbf{r}_0^k, \breve{\mathbf{c}}_0^{k-d}(\sigma_k)\right]$$

These per-survivor estimates can be used in the computation of the branch metrics:

$$\gamma_k(a_k, \sigma_k) = \ln p(r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k, \breve{\boldsymbol{\theta}}_k(\sigma_k)) + \ln P(a_k)$$

Per-Survivor Processing
Some remarks

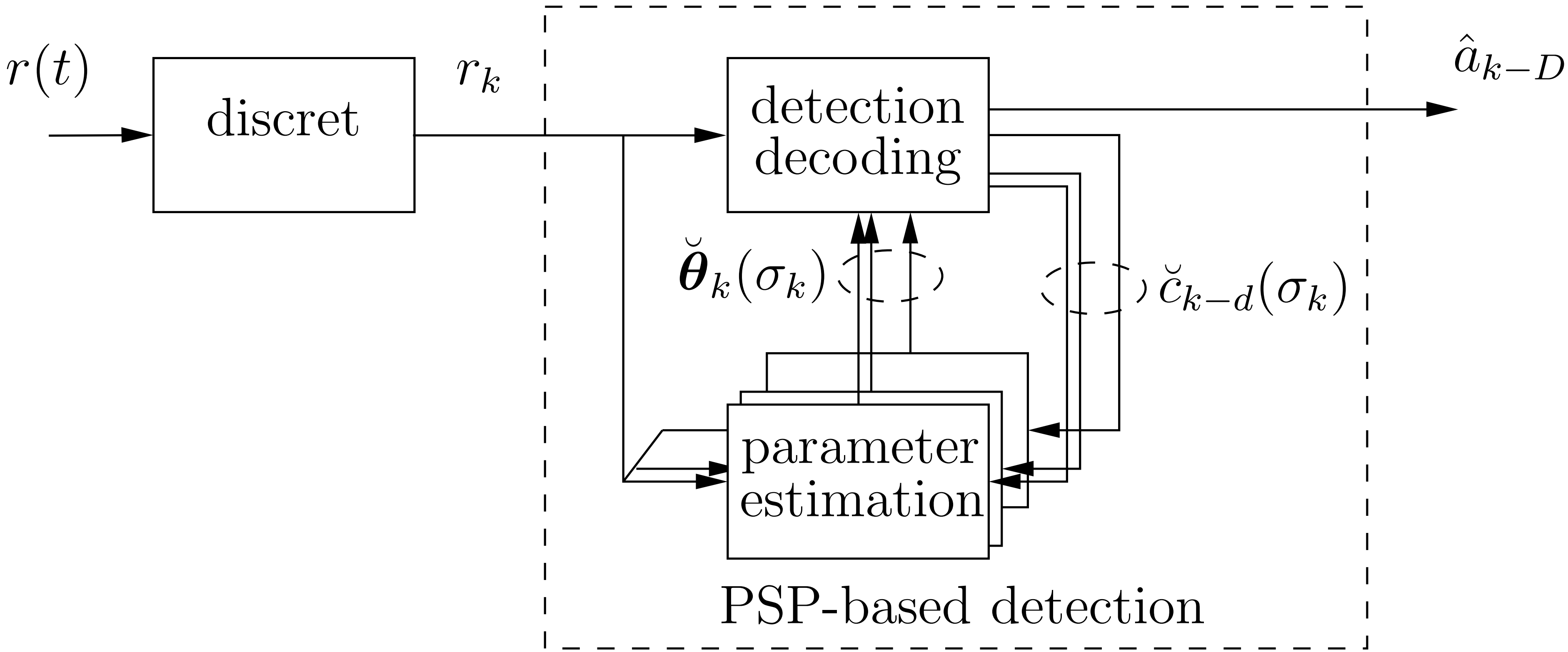
- The structure of the branch metrics is inherently different, with respect to the previous cases, in the fact that it *also* depends on the state σ_k *through the parameter estimate*

- There is now a data-aided parameter estimator *per trellis state*. This estimator uses the data sequence associated with the survivor of this state as aiding sequence. The resulting parameter estimates, one per state, are inherently *associated with the survivor sequences*—hence, the terminology "per-survivor processing"

- Compared to a conventional decomposed estimation-detection scheme based on tentative decisions, the *complexity* of per-survivor processing is *larger*

Per-Survivor Processing

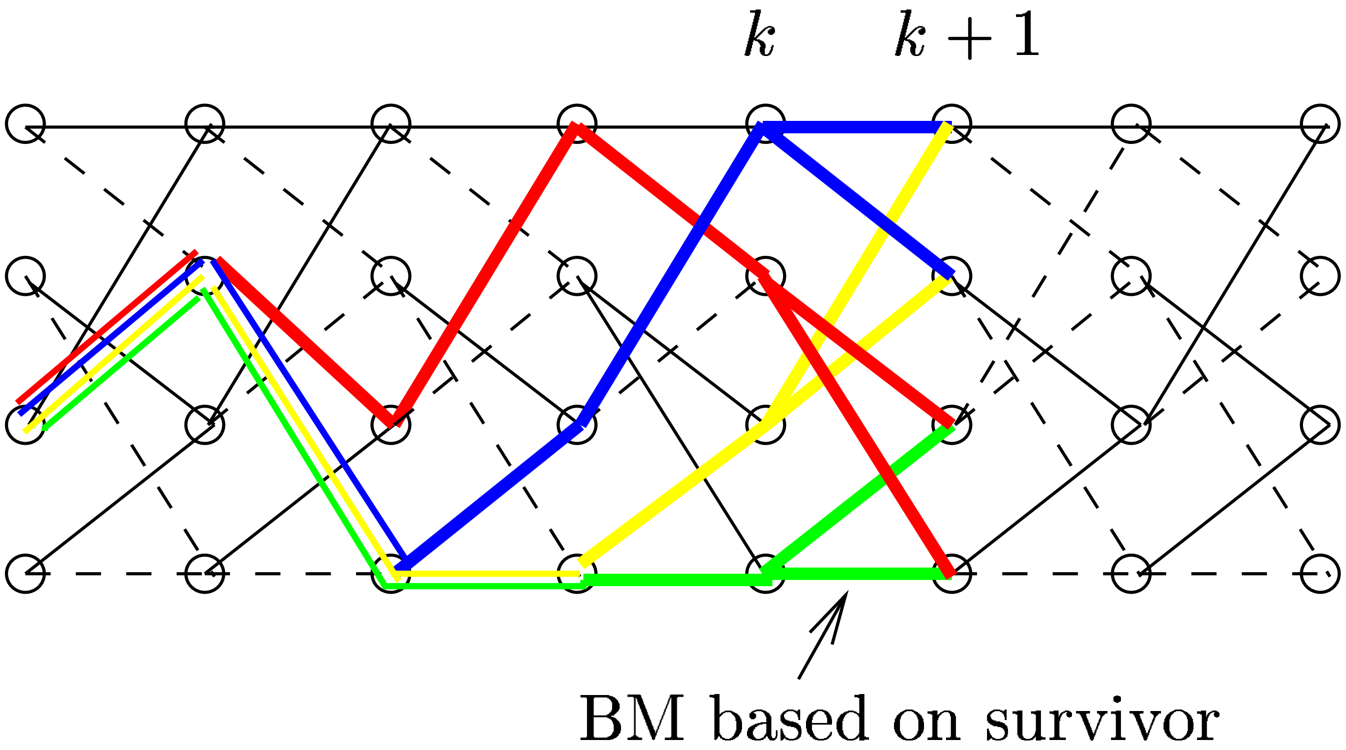
System model



A "tentative" block diagram: a set of parameter estimators observe the received sequence r_k and are aided by the survivor sequences $\breve{\mathbf{c}}_0^{k-d}(\sigma_k)$. A corresponding set of per-survivor parameter estimates $\breve{\boldsymbol{\theta}}_k(\sigma_k)$ are passed to the detection block

Per-Survivor Processing

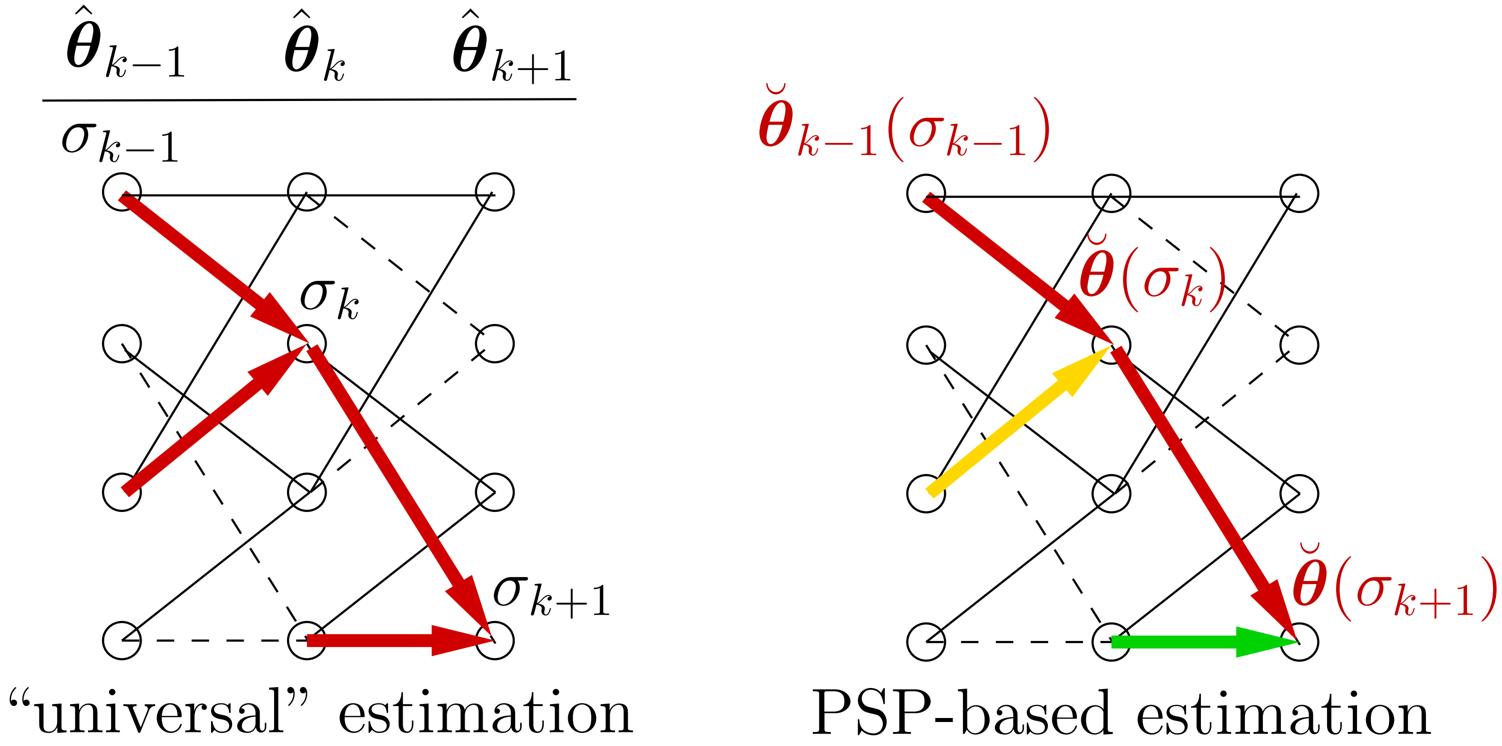
A pictorial description: branch metrics computation



The metrics of branches reaching the same state are computed using different values of the parameter.

Per-Survivor Processing

A pictorial description: evolution of the parameter estimates



In the "universal" scheme, only one parameter estimate is used for computation of all the branch metrics at the k-th step

In the PSP-based scheme the parameter estimates evolve along the survivors

Per-Survivor Processing
An intuitive rationale

Whenever the incomplete knowledge of some quantities prevents us from calculating a particular branch metric in a precise and predictable form, we use estimates of those quantities based on the data sequence associated with the survivor leading to that branch. If any particular survivor is correct (an event of high probability under normal operating conditions), the corresponding estimates are evaluated using the correct data sequence. Since at each stage of decoding we do not know which survivor is correct (or the best), we extend each survivor based on estimates obtained using its associated data sequence.

Roughly speaking, the best survivor is extended using the best data sequence available (which is the sequence associated to it), regardless of our temporary ignorance as to which survivor is the best.

Per-Survivor Processing

Is a delay d necessary?

The best survivor is extended according to its associated data sequence, despite the fact that we do not know which survivor is the best at the current time (we will know the best survivor after D further steps)

There are no reasons for delaying the aiding data sequence of the best survivor beyond the minimal delay $d = 1$ complying with the causality condition

Since all survivors eventually merge, the quality of the data sequences associated to all survivors improves for increasing values of d

$\Rightarrow$ The minimal value $d = 1$ offers the best overall performance because it attains simultaneously good quality of the aiding data sequence and a small delay in the parameter estimate

$\Rightarrow$ PSP allows one to design receivers particularly robust when the undesired parameters are time-varying

Per-Survivor Processing
Error propagation

PSP is a mechanism for virtually using "final" decisions for aiding the parameter estimation (*with no delay!*)

Only errors in the final decisions, the so-called error events, are "fed back" to the parameter estimator of the best survivor

As the aiding data sequence along the best survivor is of best possible quality, the effects of error propagation are reduced (compared with the traditional scheme that uses tentative decisions)

Parameter estimators of other survivors use data sequences of worse quality, but they do not affect future decisions provided these survivors are later discarded

Linear modulation on phase-uncertain channel
PSP-based detection and phase synchronization

Branch metrics:

$$\gamma_k(a_k, \mu_k) \propto -|r_k\, e^{-j\breve{\theta}_k(\mu_k)} - c_k(a_k, \mu_k)|^2 + \sigma_w^2 \ln P(a_k)$$

Phase estimate update recursion:

$$\breve{\theta}_{k+1}(\mu_{k+1}) = \breve{\theta}_k(\mu_k) + \eta\, \mathsf{Im}\left\{ r_{k+1-d}\, e^{-j\breve{\theta}_k(\mu_k)}\, \breve{c}^*_{k+1-d}(\mu_k) \right\}$$

$\breve{c}^*_{k+1-d}(\mu_k)$ is the code symbol at epoch $k+1-d$ in the survivor sequence of state μ_k

The phase estimate update recursions must take place along the branches which extend the survivor of state μ_k, i.e., after the usual add-compare-select step at time k

Remember: $d = 1$ for best performance

Linear modulation on dispersive fading channel
PSP-based detection and channel estimation

Branch metrics:

$$\gamma_k(a_k, \sigma_k) \propto -|r_k - \breve{\mathbf{f}}_k(\sigma_k)^T \mathbf{c}_k(a_k, \sigma_k)|^2 + \sigma_w^2 \ln P(a_k)$$

Channel estimate update recursions:

$$\breve{\mathbf{f}}_{k+1}(\sigma_{k+1}) = \breve{\mathbf{f}}_k(\sigma_k) + \beta \left[r_{k+1-d} - \breve{\mathbf{f}}_k(\sigma_k)^T \breve{\mathbf{c}}_{k+1-d}(\sigma_{k+1}) \right] \breve{\mathbf{c}}^*_{k+1-d}(\sigma_{k+1})$$

$$\breve{\mathbf{c}}_{k+1-d}(\sigma_{k+1}) = [\breve{c}_{k+1-d}(\sigma_{k+1}), \breve{c}_{k-d}(\sigma_{k+1}), \ldots, \breve{c}_{k+1-d-L}(\sigma_{k+1})]^T$$

Channel estimate update recursions must take place over those branches $(\sigma_k \rightarrow \sigma_{k+1})$ which comply with the Viterbi algorithm add-compare-select step at time k

Remember: $d = 1$ for best performance

Per-Survivor Processing
Hybrid version

The survivor merge is normally a few steps backward from current time

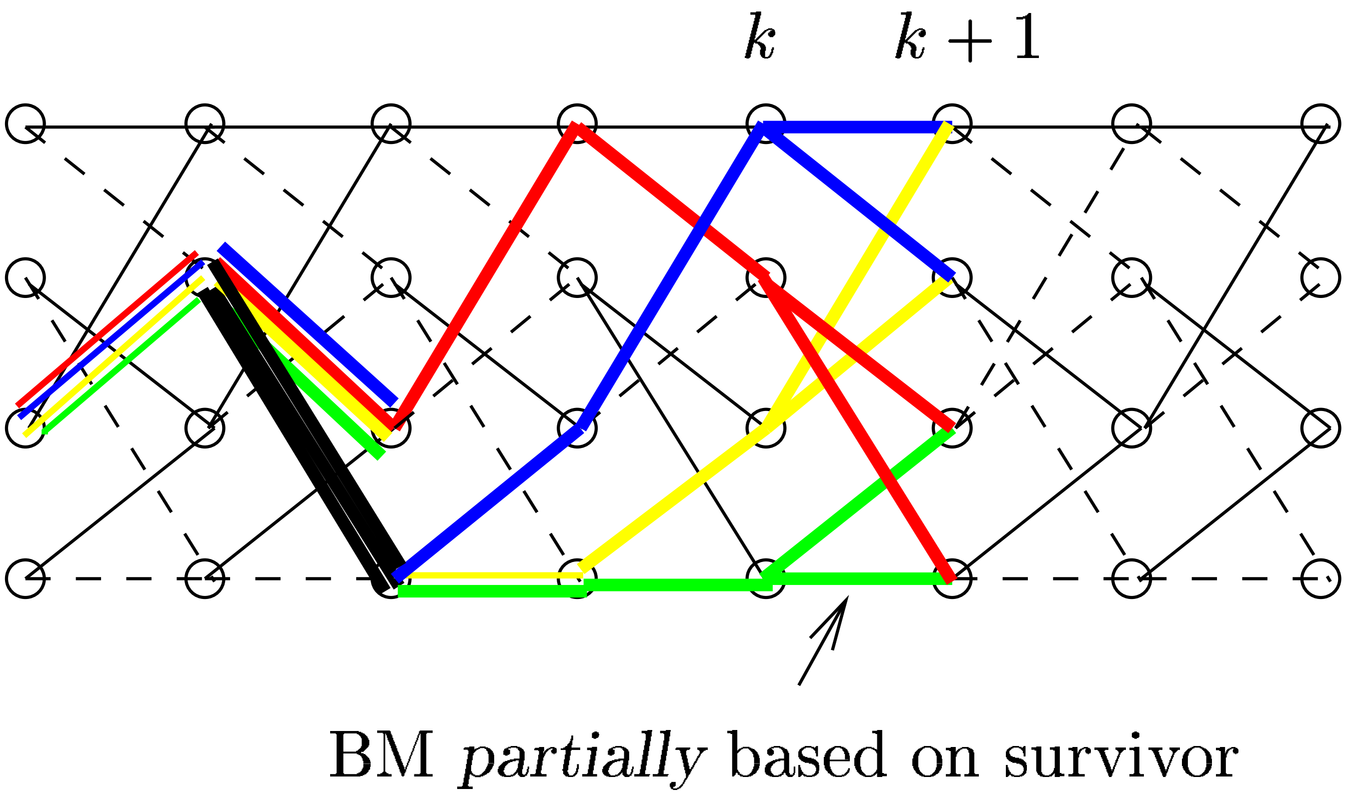
Only the most recent code symbols should be searched for in the survivor history; earlier symbols can be reliably based on preliminary decisions

Formulation:

$$\breve{\boldsymbol{\theta}}_k(\sigma_k) = \mathbf{g}_{k-l}\left[\mathbf{r}_0^k, \hat{\hat{\mathbf{c}}}_0^{k-d}, \breve{\mathbf{c}}_{k-d+1}^{k-1}(\sigma_k)\right]$$

Per-Survivor Processing

A pictorial description: Hybrid version



For $d = 3$, the computation of the branch metrics is based on 2 elements of the survivor sequences and the remaining elements of the tentative decision sequence (red survivor is best at current time)

Per-Survivor Processing
Reduced-estimator version

In PSP, the number of parameter estimators equals the number of survivors

In a conventional decomposed design, there is one parameter estimator

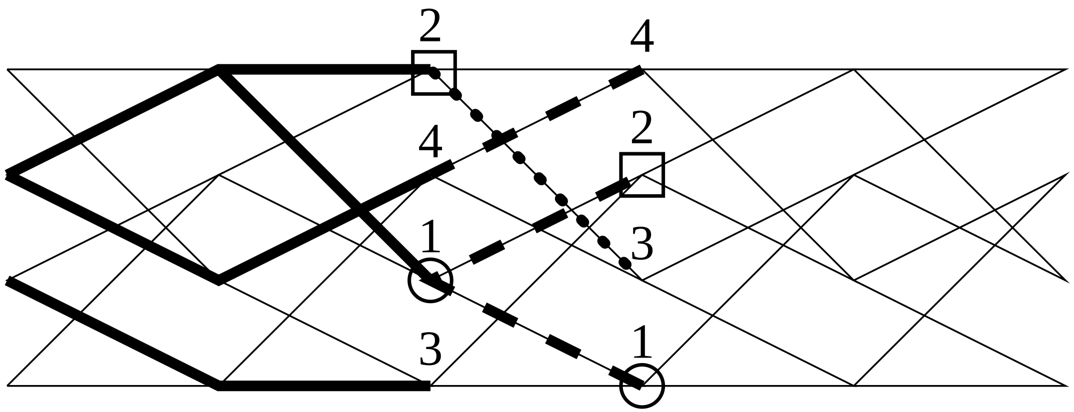
What is in between?

The number N of parameter estimators can be adjusted independently of the number S of survivors: $1 \leq N \leq S$ ($N = 1 \Rightarrow$ tentative decisions; $N = S \Rightarrow$ PSP):

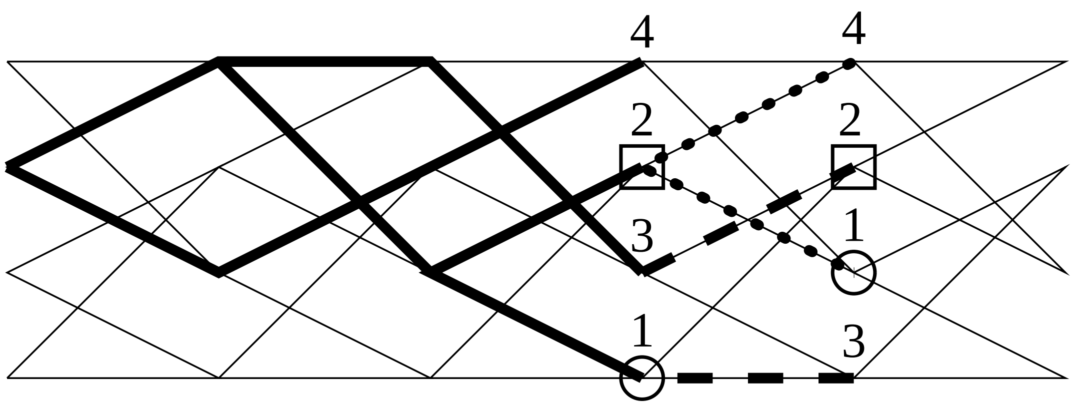
- Select the best survivor and the N best survivors;
- Extend each of the N best survivors using its associated parameter estimate;
- Extend each of the remaining $S - N$ survivors using the parameter estimate associated with the best survivor;
- Update each of the N parameter estimates along the extensions of the N best survivors.

Per-Survivor Processing

A pictorial description: Reduced-estimator version



(a)



(b)

Figure reproduced from:

– R. Raheli, G. Marino, P. Castoldi, "Per-survivor processing and tentative decisions: what is in between?," *IEEE Trans. Commun.*, pp. 127-129, Feb. 1996.

Per-Survivor Processing
Update-first version

The temporal sequence: *ACS step* followed by *parameter update* can be inverted

Update-first version of PSP:

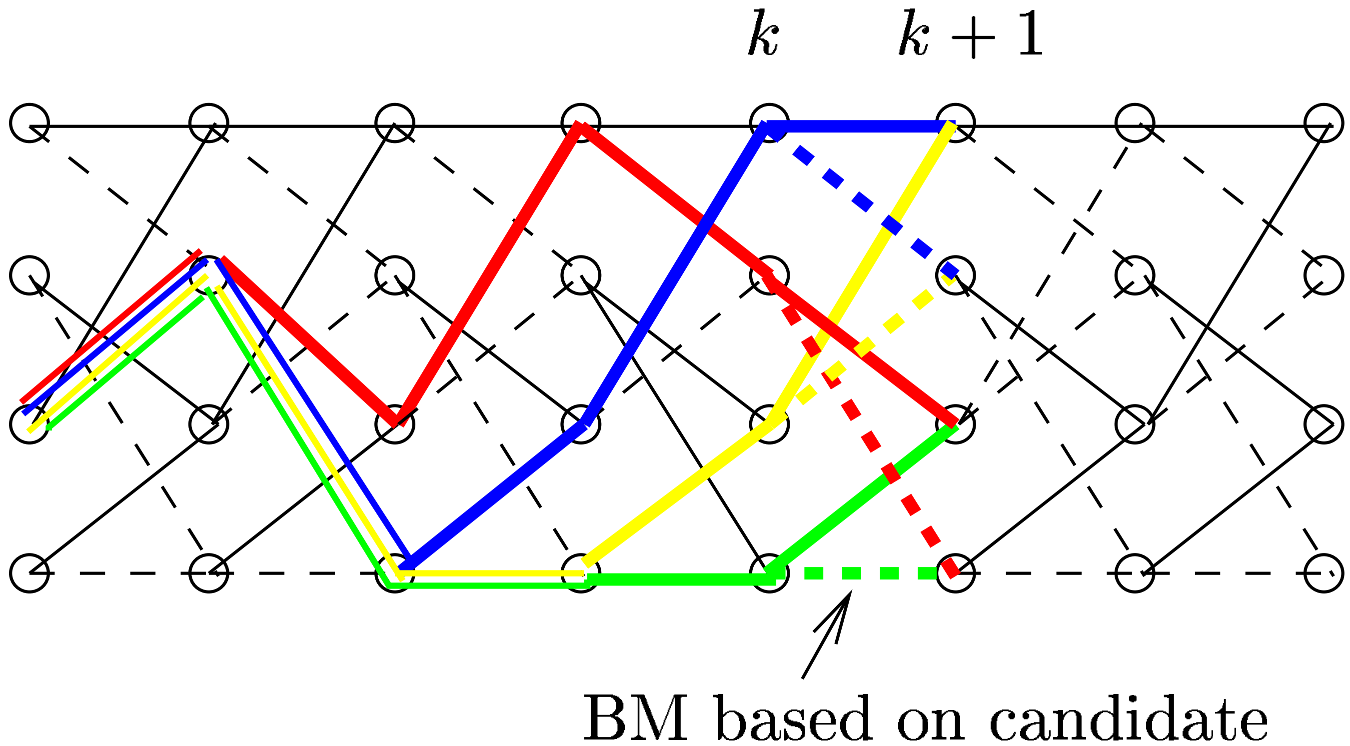
1. The per-survivor parameter update recursions are run along all possible *candidate survivors*
2. The branch metrics are computed using these updated parameter estimates
3. The ACS step is performed

In this version, there is one parameter estimator per candidate (complexity is larger)

PSP allows $d = 0$ in parameter estimation (in a scheme based on tentative decisions this would violate the causality condition)

Per-Survivor Processing

A pictorial description: Update-first version



The computation of the branch metrics is based on the candidate sequences (i.e., survivors plus their possible evolutions)

The ACS step follows on the basis of these branch metrics

Per-Survivor Processing
Application to reduced-search (sequential) algorithms

Per-survivor processing can be directly applied to any tree or trellis reduced-search algorithm, also referred to as *sequential detection* algorithms

Reduced-search algorithms may be used to search a small part of a large FSM trellis diagram or non-FSM tree diagram

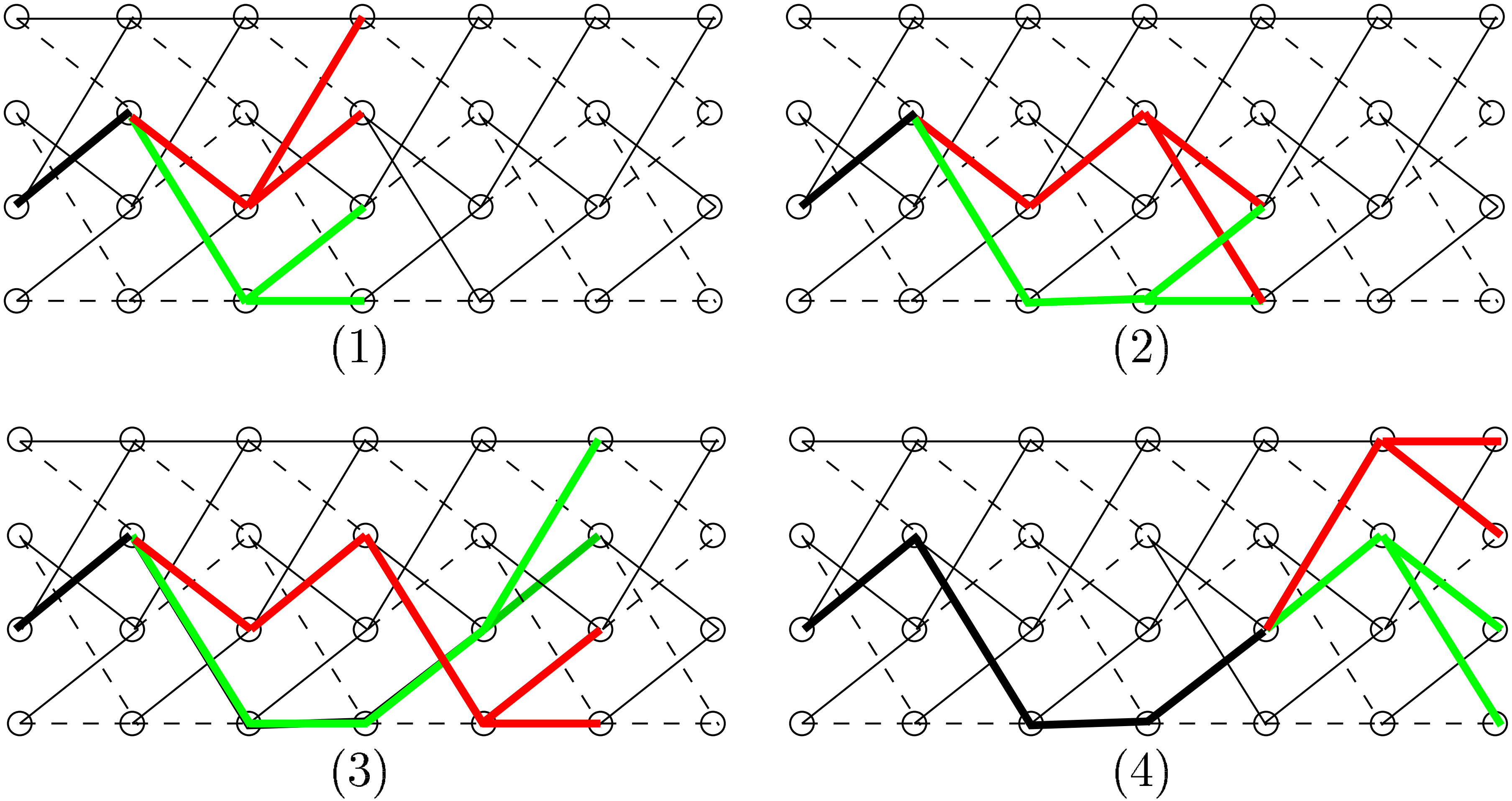
The M-algorithm keeps a list of M best paths: at each step, each path is extended in all possible way, say N; from the resulting list of MN paths, the best M are retained for further extension (*breadth-first*)

Depth-first and *metric-first* algorithms keep one or more paths and *backtrack* whenever the retained paths are judged of insufficient quality, according to some criterion

An alternative terminology could be Per-Path Processing, or P^3

Per-Survivor Processing

A pictorial description: Application to M-algorithm



$M = 2$ Branch metrics based on maintaned paths (survivors)

Per-Survivor Processing
Application to list Viterbi algorithms

The Viterbi algorithm detects the "best" MAP (or ML) path or sequence

Nothing is known about the second, third, etc. best paths

List Viterbi algorithms release the ordered list of V best paths by maintaining V survivors per state

These algorithms may be used in concatenated coding schemes: whenever the outer code detects an error, the second, third, etc. sequence at the output of the inner decoder can be tried out

Per-survivor processing can be readily applied to list Viterbi algorithms by associating a parameter estimator to each survivor

Per-Survivor Processing: concept
Bibliography

– A. Polydoros, R. Raheli, “The Principle of Per-Survivor Processing: A General Approach to Approximate and Adaptive ML Sequence Estimation,” University of Southern California, Technical Report CSI-90-07-05, July 1990. Also presented at the *IEEE Commun. Theory Workshop*, Rhodes, Greece, July 1991.

– R. Raheli, A. Polydoros, C. K. Tzou, “The Principle of Per-Survivor Processing: A General Approach to Approximate and Adaptive MLSE,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM ’91)*, Phoenix, Arizona, USA, Dec. 1991, pp. 1170-1175.

– R. Raheli, A. Polydoros, C. K. Tzou, “Per-survivor processing: a general approach to MLSE in uncertain environments,” *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.

– A. Polydoros, R. Raheli, “System and method for estimating data sequences in digital transmission,” University of Southern California, U.S. Patent No. 5,432,821, July 1995.

– R. Raheli, G. Marino, P. Castoldi, “Per-survivor processing and tentative decisions: what is in between?,” *IEEE Trans. Commun.*, pp. 127-129, Feb. 1996.

– G. Marino, “Hybrid decision feed-back sequence estimation”, in *Proc. Intern. Conf. Telecommun.*, Istanbul, Turkey, Apr. 1996, pp. 132-135.

– K. M. Chugg, A. Polydoros, “MLSE for an unknown channel—Parts I and II,” *IEEE Trans. Commun.*, pp. 836-846 and 949-958, July and Aug. 1996.

3. Per-Survivor Processing: historical review

Outline

1. Review of detection techniques
2. Detection under parametric uncertainty
3. Per-Survivor Processing (PSP): concept and historical review
4. Classical applications of PSP:
   - 4.1 Complexity reduction
   - 4.2 Linear predictive detection for fading channels
   - 4.3 Adaptive detection
5. Advanced applications of PSP

Historical review

The beginning and afterwards ...

The general concept of per-survivor processing was understood and proposed in the early nineties as a generalization of per-survivor DFE-like ISI cancellation techniques of reduced-state sequence detection (RSSD), also known as (delayed) decision-feedback sequence detection (DFSD)

RSSD and DFSD appeared and established in the late eighties, except for isolated seminal contributions which date back to the seventies

In the early nineties, a number of *independent* research results appeared in diverse technical areas which could be interpreted as special cases of the general PSP concept (not yet known)

During the nineties (and currently) PSP has emerged as a broad approach to detection in hostile transmission environments

As we will see, sometimes PSP arises naturally from the analytical development itself, when devising detection algorithms

Reduced-state sequence detection
The main references

– J. W. M. Bergmans, S. A. Rajput, F. A. M. Van De Laar, "On the use of Decision Feedback for Simplifying the Viterbi Decoder," *Philips Journal of Research*, No. 4, 1987.

– K. Wesolowski, "Efficient Digital Receiver Structure for Trellis-Coded Signals Transmitted through Channels with Intersymbol Interference," *Electronics Letters*, Nov. 1987.

– T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," *IEEE Trans. Inform. Theory*, pp. 866-876, Nov. 1987.

– M. V. Eyuboğlu, S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partition and Decision Feedback," *IEEE Trans. Commun.*, pp. 13-20, Jan. 1988.

– A. Duel Hallen, C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. Commun.*, pp. 428-436, May 1989.

– P. R. Chevillat, E. Eleftheriou, "Decoding of Trellis-Encoded Signals in the Presence of Intersymbol Interference and Noise," *IEEE Trans. Commun.*, pp. 669-676, July 1989.

– M. V. Eyuboğlu, S. U. H. Qureshi, "Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels," *IEEE J. Sel. Areas Commun.*, pp. 989-995, Aug. 1989.

– A. Svensson, "Reduced state sequence detection of full response continuous phase modulation,“ *IEE Electronics Letters*, pp. 652 -654, 1 May 1990.

Reduced-state sequence detection
There was earlier work ...

– F. L. Vermeulen and M. E. Hellman, "Reduced state Viterbi decoders for channels with intersymbol interference,' in *Proc. IEEE Int. Conf. Commun. (ICC '74)*, Minneapolis, MN, June 1974, pp. 37B1-37B4.
– F. L. Vermeulen, "Low complexity decoders for channels with intersymbol interference," Ph.D. dissertation, Dep. Elect. Eng., Stanford Univ., Aug. 1975.
– G. J. Foschini, "A reduced state variant of maximum likelihood sequence detection attaining optimum performance for high signal-to-noise ratios," *IEEE Trans. Inform. Theory*, pp. 553-651, Sept. 1977
– A. Polydoros, "Maximum-likelihood sequence estimation in the presence of infinite intersymbol interference," Master's Thesis, Graduate School of State University of New York at Buffalo, Dec. 1978.
– A. Polydoros, D. Kazakos, "Maximum-Likelihood Sequence Estimation in the Presence of Infinite Intersymbol Interference," in *Proc. ICC '79*, June 1979.

Independent results interpretable as PSP

When the time has come ...

Sequence detection for a time-varying statistically known channel:

– J. Lodge, M. Moher, "ML estimation of CPM signals transmitted over Rayleigh flat fading channels," *IEEE Trans. Commun.*, pp. 787-794, June 1990.

– D. Makrakis, P. T. Mathiopoulos, D. P. Bouras, "Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: a combined envelope, multiple differential and coherent detection approach," *IEEE Trans. Commun.*, pp.63-75, Jan. 1994.

Joint ML estimation of a deterministic channel and data detection:

– R. Iltis, "A Bayesian MLSE algorithm for a priori unknown channels and symbol timing," *IEEE J. Sel. Areas Commun.*, April 1992.

– N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Commun.*, Feb.-Apr. 1994.

Independent results interpretable as PSP
When the time has come ... (cntd)

Adaptive sequence detection with tracking of a time-varying deterministic channel:

– Z. Xie, C. Rushforth, R. Short, T. Moon, "Joint signal detection and parameter estimation in multiuser communications," *IEEE Trans. Commun.*, Aug. 1993.
– H. Kubo, K. Murakami, T. Fujino, "An adaptive MLSE for fast time-varying ISI channels," *IEEE Trans. Commun.*, pp, 1872-1880, Feb.-Apr. 1994.

Trellis coded quantization (source encoding):

– M. W. Marcellin, T. R. Fischer, "Trellis coded quantization of memoryless Gauss-Markov sources," *IEEE Trans. Commun.*, Jan. 1990.

Joint sequence detection and carrier phase synchronization:

– A. J. Macdonald and J. B. Anderson, "PLL synchronization for coded modulation," in *Proc. ICC '91*, June 1991.
– A. Reichman and R. A. Scholtz, "Joint phase estimation and data decoding for tcm systems," in *Proc. First Intern. Symp. Commun. Theory and Applications*, Scotland, U.K., Sept. 1991.

Earlier work
... and beforehand

Analog FM demodulation with discrete phase approximation based on the Viterbi algorithm (*there are no data !*)

In an extended-memory version, a procedure similar to PSP was proposed

– C. Cahn, "Phase tracking and demodulation with delay," *IEEE Trans. Inform. Theory*, Jan. 1974.

The roots
Generalized likelihood

Model the parameter as deterministic or random with unknown distribution

Joint ML parameter estimation and sequence detection viewed as a *composite* hypotheses test:

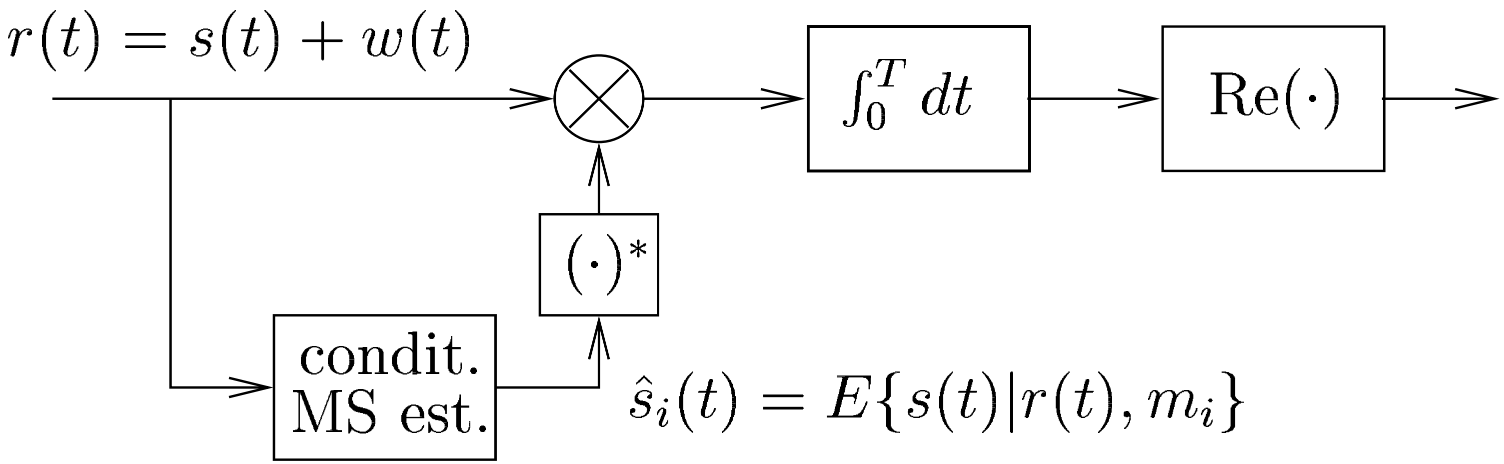
$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} \underbrace{\left\{ \max_{\boldsymbol{\theta}} \; p\left(\mathbf{r} | \mathbf{a}, \boldsymbol{\theta}\right) \right\}}_{\Rightarrow \hat{\boldsymbol{\theta}}\left(\mathbf{a}\right)}$$

⇒ A per-hypothesis parameter estimate is obtained as a side result

– H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: John Wiley & Sons, 1968.

The roots
Estimation-Correlation detection

Detection of M-ary random signals in AWGN: $s(t)$ is conditionally Gaussian, given m_i



$\Rightarrow$ Per-hypothesis conditional mean square estimate of $s(t)$

For deterministic signals: $s(t) \in \{s_i(t)\}_{i=1}^{M} \Rightarrow \hat{s}_i(t) = s_i(t)$

– T. Kailath, "Correlation detection of signals perturbed by a random channel," *IRE Trans. Inform. Theory*, June 1960.
– T. Kailath, "A general likelihood ratio formula for random signals in Gaussian noise," *IEEE Trans. Inform. Theory*, May 1969.

Further references on PSP

... this is not an exhaustive list ...

– A. N. D'Andrea, U. Mengali, and G. M. Vitetta, "Approximate ML decoding of coded PSK with no explicit carrier phase reference," *IEEE Trans. Commun.*, pp. 1033-1039, Feb.-Apr. 1994.

– Q. Dai, E. Shwedyk, "Detection of bandlimited signals over frequency selective Rayleigh fading channels," *IEEE Trans. Commun.*, pp. 941-950, Feb.-Apr. 1994.

– J. Lin, F. Ling, J. Proakis, "Joint data and channel estimation for TDMA mobile channels," *Plenum Intern. J. Wireless Inform. Networks*, vol. 1, no. 4, pp. 229-238, 1994.

– X. Yu, S. Pasupathy, "Innovations-based MLSE for Rayleigh fading channels," *IEEE Trans. Commun.*, pp. 1534-1544, Feb.-Apr. 1995.

– G. M. Vitetta, D. P. Taylor, "Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels," *IEEE Trans. Commun.*, vol. 43, pp. 2750-2758, Nov. 1995

– K. Hamied, G. Stüber, "An adaptive truncated MLSE receiver for Japanese personal digital cellular," *IEEE Trans. Veh. Techn.*, Feb. 1996.

– G. M. Vitetta, D. P. Taylor, U. Mengali, "Double filtering receivers for PSK signals transmitted over Rayleigh frequency-flat fading channels," *IEEE Trans. Commun.*, vol. 44, pp. 686-695, June 1996.

Further references on PSP

... this is not an exhaustive list ... (cntd)

– M. E. Rollins, S. J. Simmons, "Simplified per-survivor Kalman processing in fast frequency-selective fading channels," *IEEE Trans. Commun.*, pp. 544-553, May 1997.
– B. C. Ng, S. N. Diggavi, A. Paulray, "Joint structured channel and data estimation over time-varying channels," in *Proc. IEEE Globecom*, 1997.
– A. Anastasopoulos, A. Polydoros, "Adaptive soft-decision algorithms for mobile fading channels," *European Trans. Telecommun.*., vol. 9, no. 2, pp. 183-190, Mar-Apr. 1998.
– K. M. Chugg, "Blind acquisition characteristics of PSP-based sequence detectors," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 1518-1529, Oct. 1998.
– F. Rice, B. Cowley, M. Rice, B. Moran, "Spectrum analysis using a trellis algorithm," in *Proc. IEEE Intern. Conf. Signal Process. (ICTS '98)*, Oct. 1998.

4. Classical applications of PSP

4.1 Complexity reduction

Reduction of trellis state-complexity
Motivation

Consider a FSM system model based on some finite memory property (strict-sense or conditional)

Detection schemes are based on branch metrics: $\gamma_k(a_k, \sigma_k)$

System state: $\sigma_k = (a_{k-1}, a_{k-2}, \ldots, a_{k-L}; \mu_{k-L}) \qquad (\mathcal{C} = L)$

Encoder state: μ_k

Coding rule: $\begin{cases} c_k = o(a_k, \mu_k) \\ \mu_{k+1} = t(a_k, \mu_k) \end{cases}$

Number of FSM states: $S = S_c M^L$

Example: $S_c = M = L = 4 \quad \Rightarrow \quad S = 4 \times 4^4 = 4 \times 2^8 = 1024$

$\Rightarrow$ State-complexity may be large!

Reduction of trellis state-complexity

An alternative definition of system state

Assume an invertible coding rule

Solving for given (c_{k-1}, μ_k)

$$\begin{cases} o(a_{k-1}, \mu_{k-1}) = c_{k-1} \\ t(a_{k-1}, \mu_{k-1}) = \mu_k \end{cases} \quad \Rightarrow \quad (a_{k-1}, \mu_{k-1})$$

input seq.:	$\ldots$	a_{k-L}	a_{k-L+1}	$\ldots$	a_{k-2}	a_{k-1}	a_k	$\ldots$
state seq.:	$\ldots$	μ_{k-L}	μ_{k-L+1}	$\ldots$	μ_{k-2}	μ_{k-1}	μ_k	$\ldots$
code seq.:	$\ldots$	c_{k-L}	c_{k-L+1}	$\ldots$	c_{k-2}	c_{k-1}	c_k	$\ldots$

System state can be equivalently defined as:

$$\sigma_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-L})$$

State transition and corresponding branch metrics can be equivalently defined as (c_k, σ_k) and $\gamma_k(c_k, \sigma_k)$, respectively

REDUCTION OF TRELLIS STATE-COMPLEXITY
Genie-aided trellis folding

Suppose at each epoch k a *genie* passed a group of correct code symbols $\tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L}$ to the branch metric computer $(Q \leq L)$

Genie-aided branch metrics could be defined as $\gamma_k(c_k, \tilde{\sigma}_k)$ for each state σ_k whose first $Q+1$ entries coincide with those in

$$\tilde{\sigma}_k = (\mu_k; c_{k-1}, \ldots, c_{k-Q}, \tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L})$$

The group of states

$$\sigma_k = (\underbrace{\mu_k; c_{k-1}, \ldots, c_{k-Q}}_{\text{fixed}}, c_{k-Q-1}, \ldots, c_{k-L}) \quad \forall c_{k-Q-1}, \ldots, c_{k-L}$$

would have *identical* path metrics and could be *folded* into a *reduced* state

$$\omega_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-Q})$$

Reduction of trellis state-complexity
Folding by memory truncation

Path search could be equivalently performed in a folded trellis diagram with this reduced (also *partial*, *folded* or *super*) state

$$\omega_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-Q})$$

Genie-aided branch metrics in the reduced-state trellis:

$$\tilde{\gamma}_k(c_k, \omega_k) = \gamma_k(c_k, \tilde{\sigma}_k(\omega_k))$$

A *pseudo state* is defined as

$$\tilde{\sigma}_k(\omega_k) = (\underbrace{\mu_k; c_{k-1}, \ldots, c_{k-Q}}_{\omega_k}, \tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L})$$

where $(\tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L})$ is the genie information

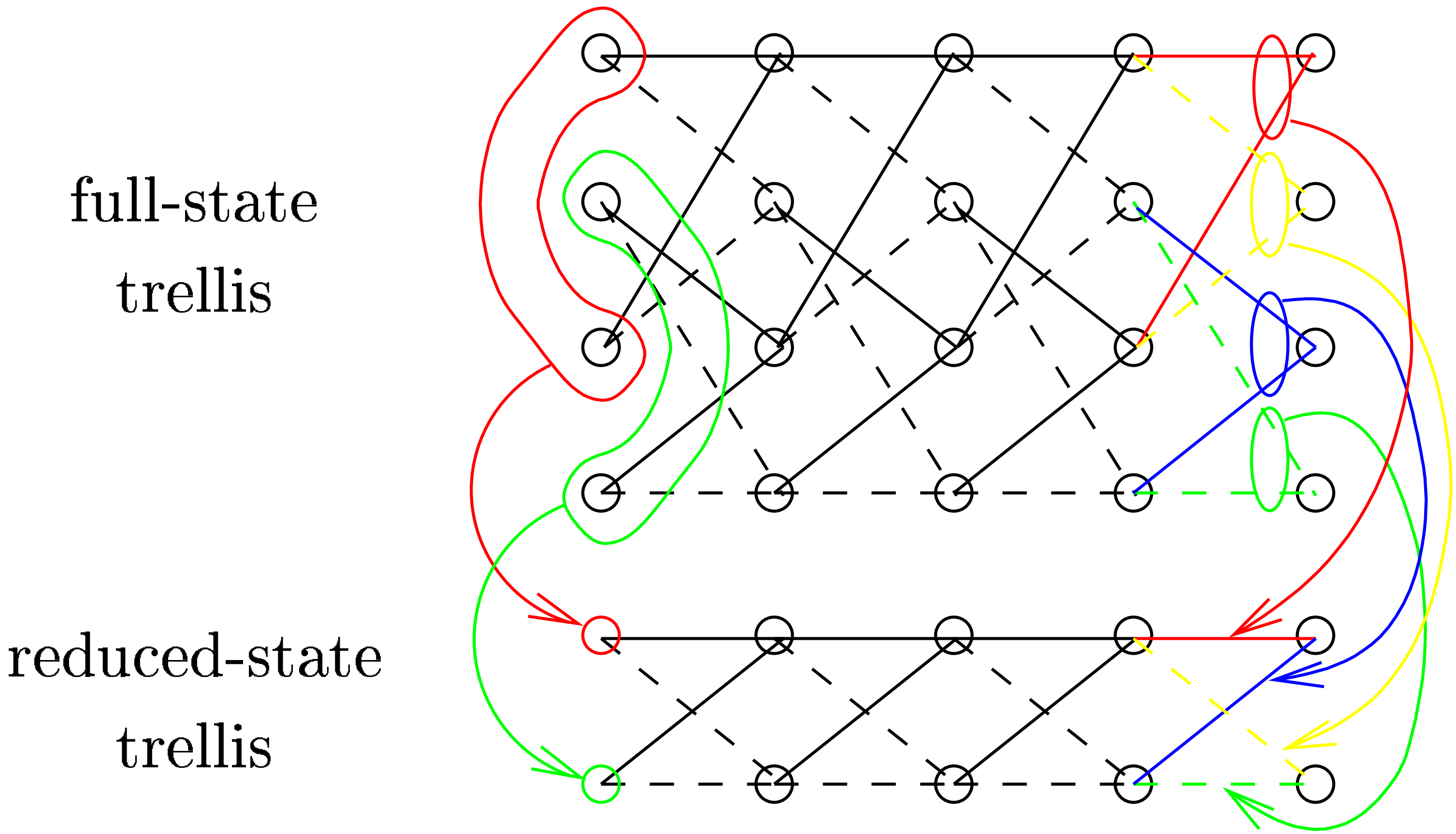
Effective genie-aided *truncation* of the system memory: $Q \leq L$

Reduced number of states: $S' = S_c M^Q \leq S = S_c M^L$

⇒The full-state trellis folds into a reduced-state trellis

Reduction of trellis state-complexity

A pictorial description of trellis folding



Reduction of trellis state-complexity

Interpretation of trellis folding by memory truncation

The code symbols $(\tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L})$ can be viewed as an *undesired set of parameters*

A parameter-conditional *reduced* memory property holds

The Estimation-Detection decomposition can be (again) the route to the approximation of the branch metrics in the presence of this special parametric uncertainty

The genie information $(\tilde{c}_{k-Q-1}, \ldots, \tilde{c}_{k-L})$ must be estimated in order to implement detection schemes with reduced state-complexity

Curiosity: we do not need a data-aided parameter estimator but only the aiding code sequence

We can use tentative decisions or per-survivor processing

Reduction of trellis state-complexity
Branch metrics based on tentative-decision feedback

Branch metrics in the original full-state trellis: $\gamma_k(c_k, \sigma_k)$

Branch metrics in the reduced-state trellis:

$$\tilde{\gamma}_k(c_k, \omega_k) = \gamma_k(c_k, \tilde{\sigma}_k(\omega_k))$$

The pseudo state is defined as

$$\tilde{\sigma}_k(\omega_k) = (\underbrace{\mu_k; c_{k-1}, \ldots, c_{k-Q}}_{\omega_k}, \hat{\hat{c}}_{k-Q-1}, \ldots, \hat{\hat{c}}_{k-L})$$

$\hat{\hat{c}}_{k-Q-1}, \ldots, \hat{\hat{c}}_{k-L}$ are preliminary decisions on the code sequence

Reduction of trellis state-complexity
Branch metrics based on PSP

Branch metrics in the original full-state trellis: $\gamma_k(c_k, \sigma_k)$

Branch metrics in the reduced-state trellis:

$$\tilde{\gamma}_k(c_k, \omega_k) = \gamma_k(c_k, \tilde{\sigma}_k(\omega_k))$$

The pseudo state is defined as

$$\tilde{\sigma}_k(\omega_k) = (\underbrace{\mu_k; c_{k-1}, \ldots, c_{k-Q}}_{\omega_k}, \breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k))$$

$\breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k)$ are the code symbols in the survivor of state ω_k

The pseudo state depends on ω_k through the feedback of survivor symbols as well

Linear modulation on static dispersive channel
Branch metrics in a reduced-state trellis

Branch metrics in the original full-state trellis ($S = S_c M^L$ states)

$$\gamma_k(c_k, \sigma_k) = -\left| r_k - f_k c_k - \sum_{l=1}^{L} f_l c_{k-l}(\sigma_k) \right|^2 + \sigma_w^2 \ln P[a_k(c_k, \sigma_k)]$$

Branch metrics in the reduced-state trellis ($S' = S_c M^Q$ states)

$$\begin{aligned}\tilde{\gamma}_k(c_k, \omega_k) &= \gamma_k(c_k, \tilde{\sigma}_k(\omega_k)) \\ &= -\left| r_k - f_k c_k - \sum_{l=1}^{Q} f_l c_{k-l}(\omega_k) - \sum_{l=Q+1}^{L} f_l \breve{c}_{k-l}(\omega_k) \right|^2 \\ &\quad + \sigma_w^2 \ln P[a_k(c_k, \omega_k)]\end{aligned}$$

$c_{k-1}(\omega_k), \ldots, c_{k-Q}(\omega_k)$ are code symbols uniquely associated with state ω_k

$\breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k)$ are code symbols in the survivor of state ω_k

Reduction of trellis state-complexity
Folding by set partitioning

State-complexity reduction can also be achieved replacing the code symbols c_{k-i} in the "full" state

$$\sigma_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-L})$$

with subsets of the code symbol alphabet (or *constellation*)

Define a reduced state

$$\omega_k = (\mu_k; I_{k-1}(1), I_{k-2}(2), \ldots, I_{k-L}(L))$$

At epoch k, for $i = 1, 2, \ldots, L$:

$I_{k-i}(i) \in \Omega(i)$ are subsets of the code constellation $\mathcal{A}$

$\Omega(i)$ are partitions of the code constellation $\mathcal{A}$

A given reduced state specifies only the constellation subsets $I_{k-i}(i)$

$c_{k-i} \in I_{k-i}(i)$ are code symbols compatible with the given state

Reduction of trellis state-complexity
Folding by set partitioning (cntd)

Let $J_i = \text{card}\{\Omega(i)\}$ and $M' = \text{card}\{\mathcal{A}\}$ $(1 \leq J_i \leq M')$

The reduced state is well-defined if current state ω_k and subset $I_k(1)$ (which the current symbol c_k belongs to) uniquely specify the successive state

$$\omega_{k+1} = (\mu_{k+1}; I_k(1), I_{k-1}(2), \ldots, I_{k-L+1}(L))$$

$\Omega(i)$ must be a further partition of $\Omega(i+1)$

The *partition depths* J_i must satisfy the condition

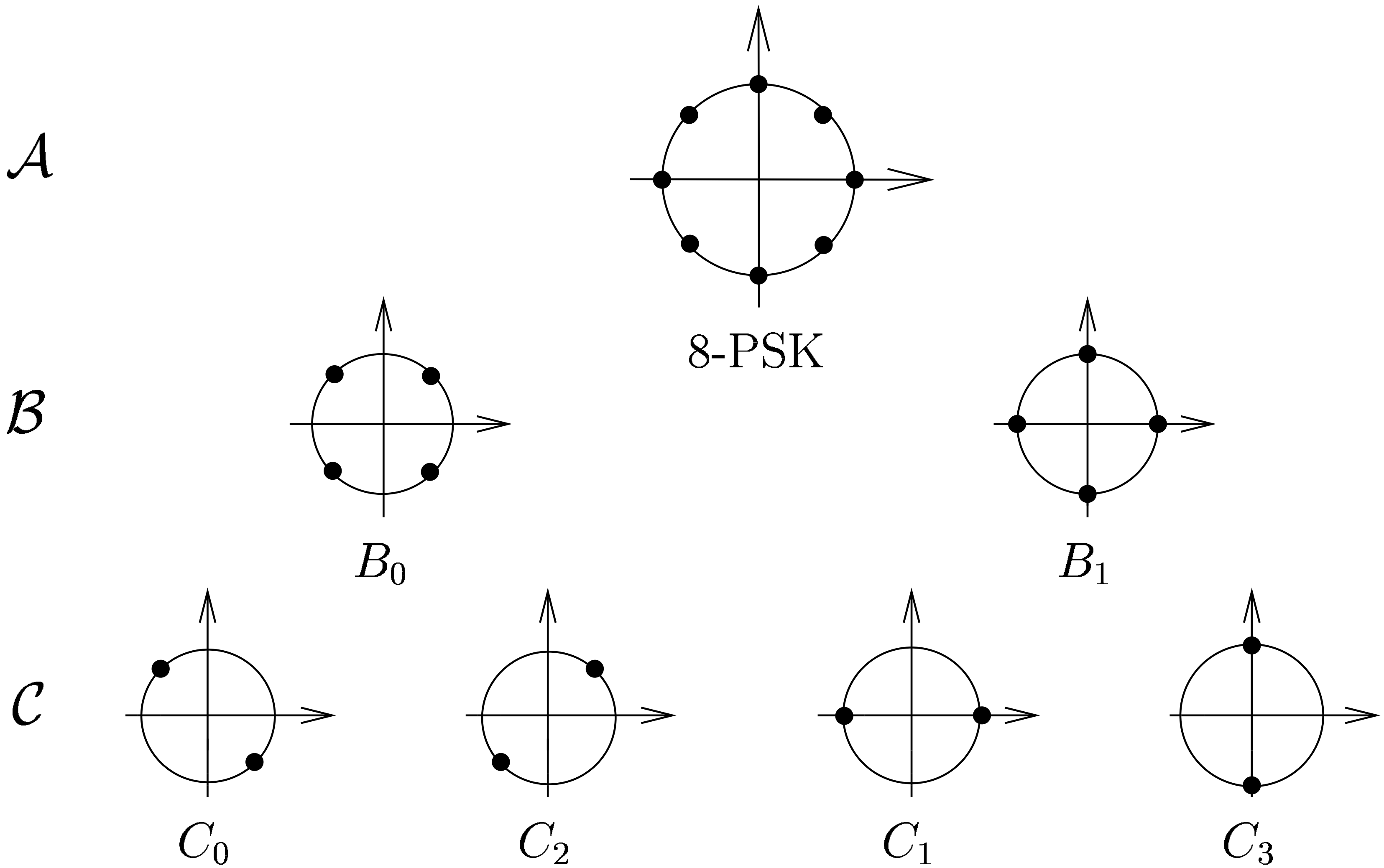
$$J_1 \geq J_2 \geq \cdots \geq J_L$$

If Q is such that $J_Q > 1$ and $J_{Q+1} = \cdots = J_L = 1$, the definition of partial state can be simplified

$$\omega_k = (\mu_k; I_{k-1}(1), I_{k-2}(2), \ldots, I_{k-Q}(Q))$$

For $J_1 = \cdots = J_Q = M'$, memory truncation arises as a special case

Reduction of trellis state-complexity
An example: partition of an 8-PSK constellation



Reduction of trellis state-complexity
An example: definition of partial states

Full-complexity state for uncoded transmission and $L = 2$

$$\sigma_k = (a_{k-1}, a_{k-2}) \qquad S = M^L = 8^2 = 64$$

1. Partial state by memory truncation with $Q = 1$ (also $J_1 = 8,\ J_2 = 1$)

$$\omega'_k = a_{k-1} \qquad S' = M^Q = 8^1 = 8$$

2. Partial state by set partition with

$$\begin{aligned} \Omega(1) &= \mathcal{C} = \{C_0, C_1, C_2, C_3\} \quad (J_1 = 4) \\ \Omega(2) &= \mathcal{B} = \{B_0, B_1\} \quad (J_2 = 2) \\ \omega''_k &= (I_{k-1}(1), I_{k-2}(2)) \qquad S'' = J_1 J_2 = 4 \times 2 = 8 \end{aligned}$$

3. Partial state by set partition with

$$\begin{aligned} \Omega(1) &= \Omega(2) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} \quad (J_1 = J_2 = 4) \\ \omega'''_k &= (I_{k-1}(1), I_{k-2}(2)) \qquad S''' = J_1 J_2 = 4 \times 4 = 16 \end{aligned}$$

Reduction of trellis state-complexity
Folding by set partitioning: some remarks

Set partition should follow the partition rules used in Trellis Coded Modulation (TCM)

If $J_1 < M'$, *parallel transitions* may be present (they are in the uncoded case)

If $J_1 < M'$, state transitions and corresponding branch metrics are defined as $(I_k(1), \omega_k)$ and $\tilde{\gamma}_k(I_k(1), \omega_k)$, respectively

Reduction of trellis state-complexity
Folding by set partitioning: branch metrics

Branch metrics in the original full-state trellis: $\gamma_k(c_k, \sigma_k)$

Branch metrics in the reduced-state trellis:

$$\tilde{\gamma}_k(I_k(1), \omega_k) = \max_{c_k \in I_k(1)} \gamma_k(c_k, \tilde{\sigma}_k(\omega_k))$$

The pseudo state $\tilde{\sigma}_k(\omega_k)$ must be compatible with ω_k: $c_{k-i} \in I_{k-i}(i)$

Missing information can be based on tentative decisions or PSP

PSP-based pseudo state:

$$\tilde{\sigma}_k(\omega_k) = (\mu_k; \tilde{c}_{k-1}(\omega_k), \ldots, \tilde{c}_{k-Q}(\omega_k), \breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k))$$

$\tilde{c}_{k-1}(\omega_k), \ldots, \tilde{c}_{k-Q}(\omega_k)$ are code symbols compatible with state ω_k to be found in the survivor history of state ω_k

$\breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k)$ are code symbols in the survivor of state ω_k

Linear modulation on static dispersive channel
Folding by set partitioning: branch metrics

Branch metrics in the original full-state trellis ($S = S_c M^L$ states)

$$\gamma_k(c_k, \sigma_k) = -\left| r_k - f_k c_k - \sum_{l=1}^{L} f_l c_{k-l}(\sigma_k) \right|^2 + \sigma_w^2 \ln P[a_k(c_k, \sigma_k)]$$

Branch metrics in the reduced-state trellis

$$\begin{aligned}\tilde{\gamma}_k(I_k(1), \omega_k) &= \max_{c_k \in I_k(1)} \gamma_k(c_k, \tilde{\sigma}_k(\omega_k)) \\ &= \max_{c_k \in I_k(1)} -\left| r_k - f_k c_k - \sum_{l=1}^{Q} f_l \tilde{c}_{k-l}(\omega_k) - \sum_{l=Q+1}^{L} f_l \breve{c}_{k-l}(\omega_k) \right|^2 \\ &\quad + \sigma_w^2 \ln P[a_k(c_k, \omega_k)]\end{aligned}$$

$\tilde{c}_{k-1}(\omega_k), \ldots, \tilde{c}_{k-Q}(\omega_k)$ are code symbols compatible with state ω_k to be found in the survivor history of state ω_k

$\breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k)$ are code symbols in the survivor of state ω_k

Reduction of trellis state-complexity
Problem 9

Consider a linear modulation for transmitting uncoded binary symbols $a_k \in \{\pm 1\}$ through the static dispersive channel with white-noise discrete equivalent considered in Problem 5

A. Define a reduced system state by memory truncation and draw the relevant trellis diagram

B. Express explicitly the branch metrics as a function of the received signal sample r_k for any possible transition in the reduced trellis

Assume the received sequence is

$$(r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7) = (1.7, 1.2, 1.1, 0.3, -0.2, -1.1, 0.7, 0.4)$$

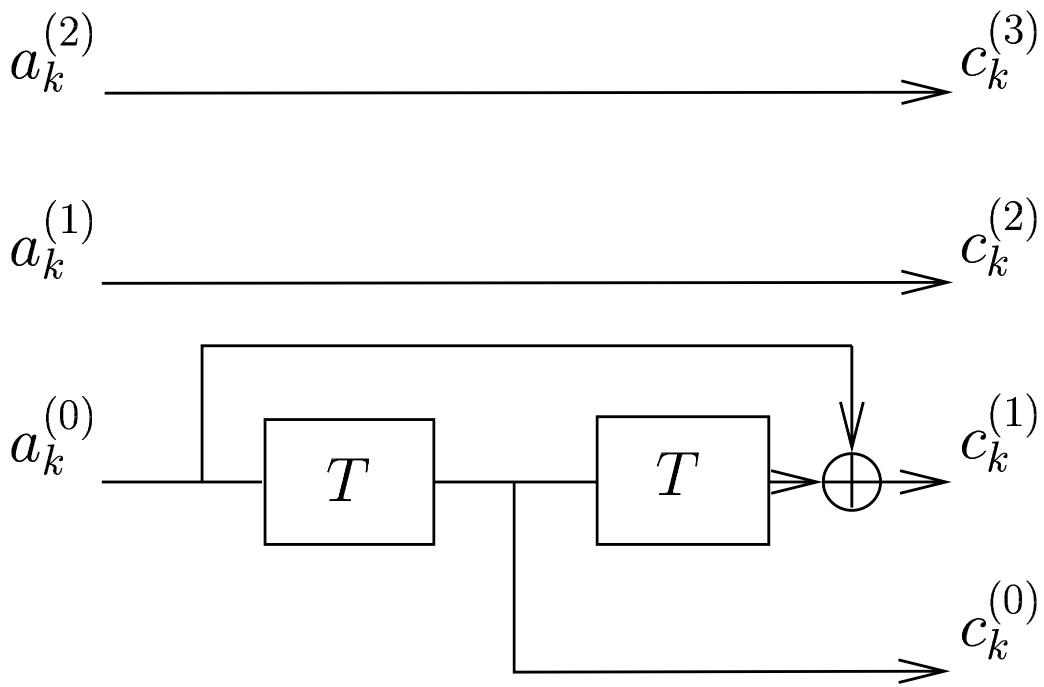
and the initial "full" state is $\sigma_0 = (+1, +1)$

C. Use the Viterbi algorithm to approximately detect the MAP sequence $\{\hat{a}_k\}_{k=0}^{7}$

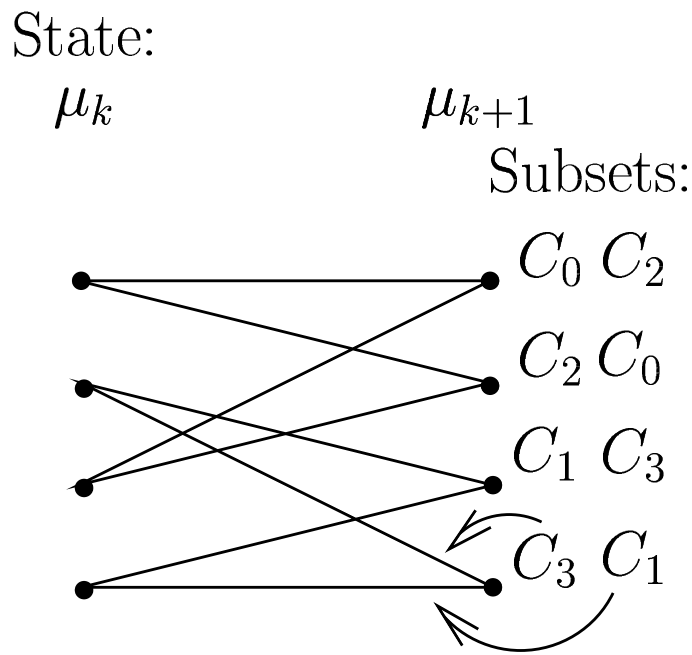
D. Would it be possible to define a different reduced system state by set partitioning?

A case-study: TCM on ISI channel

System model: encoder



4-state TCM encoder



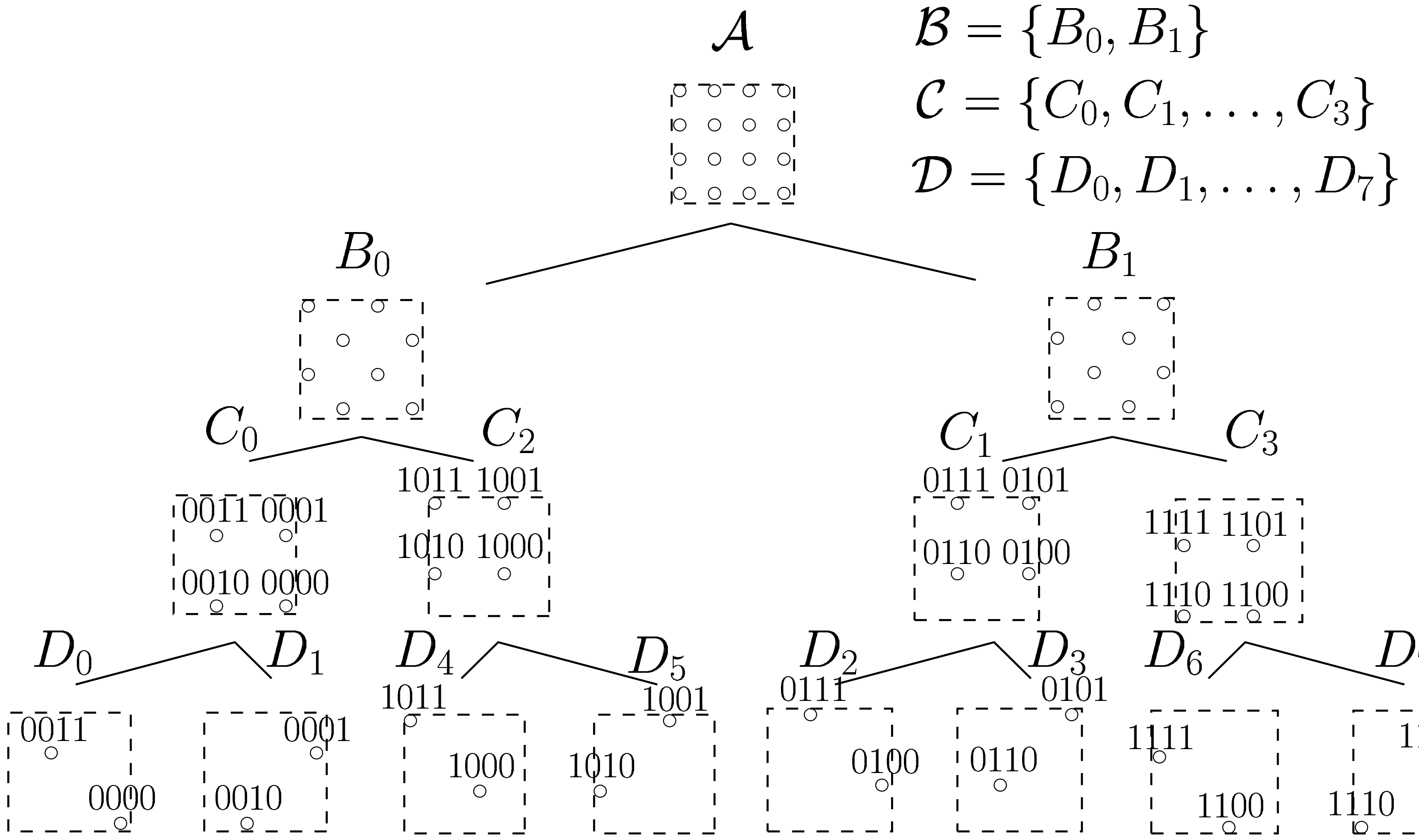
4-state code trellis

Information and code *bits*: $a_k^{(i)}$ and $c_k^{(j)}$

Gross spectral efficiency: 3 bit/s/Hz (to be reduced by the bandwidth expansion factor $1 + \alpha = 1.3 \Rightarrow$ net spectral efficiency is 2.3 bit/s/Hz)

TCM on ISI channel

System model: set partition and mapping rule



2 bits per parallel transition (code trellis)

TCM on ISI channel

System model: channel response

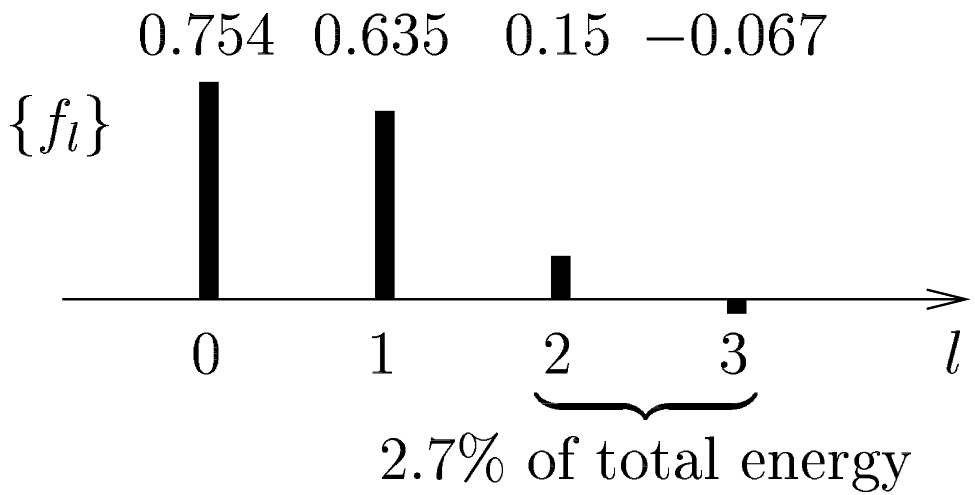
Model of discrete observable:

$$r_k = \sum_{l=0}^{L} f_l c_{k-l} + w_k$$

$\{f_l\}_{l=0}^{L}$: *white-noise discrete equivalent* of the ISI channel

$\{w_k\}$: i.i.d. Gaussian noise sequence.

$\{c_k\}$: code sequence



L must be large enough to accommodate the significant pulses

$L = 3$ may be sufficient for the considered channel

TCM on ISI channel
Definition of partial states

0. Full-complexity state for $L = 3$

$$\sigma_k = (\mu_k; c_{k-1}, c_{k-2}, c_{k-3}) \qquad S = S_c M^L = 4 \times 8^3 = 4 \times 2^9 = 2048$$

1. Partial state by memory truncation with $Q = 1$ ($J_1 = 16,\ J_2 = J_3 = 1$)

$$\omega'_k = (\mu_k; c_{k-1}) \qquad S' = S_c M^Q = 4 \times 8^1 = 32$$

2. Partial state by set partition with

$$\begin{aligned} \Omega(1) &= \mathcal{D} = \{D_0, D_1, \ldots, D_7\} && (J_1 = 8) \\ \Omega(2) &= \Omega(3) = \mathcal{A} && (J_2 = J_3 = 1) \end{aligned}$$

$$\omega''_k = (\mu_k; I_{k-1}(1)) \qquad S'' = S_c \frac{J_1}{2} = 4 \times 4 = 16$$

TCM on ISI channel
Definition of partial states (cntd)

3. Partial state by set partition with

$$\Omega(1) = \Omega(2) = \mathcal{C} = \{C_0, C_1, C_2, C_3\} \quad (J_1 = J_2 = 4)$$
$$\Omega(3) = \mathcal{A} \quad (J_3 = 1)$$
$$\omega_k''' = (\mu_k; I_{k-1}(1), I_{k-2}(2)) \qquad S''' = S_c \frac{J_1}{2} \frac{J_2}{2} = 4 \times 2 \times 2 = 16$$

4. Partial state by set partition with

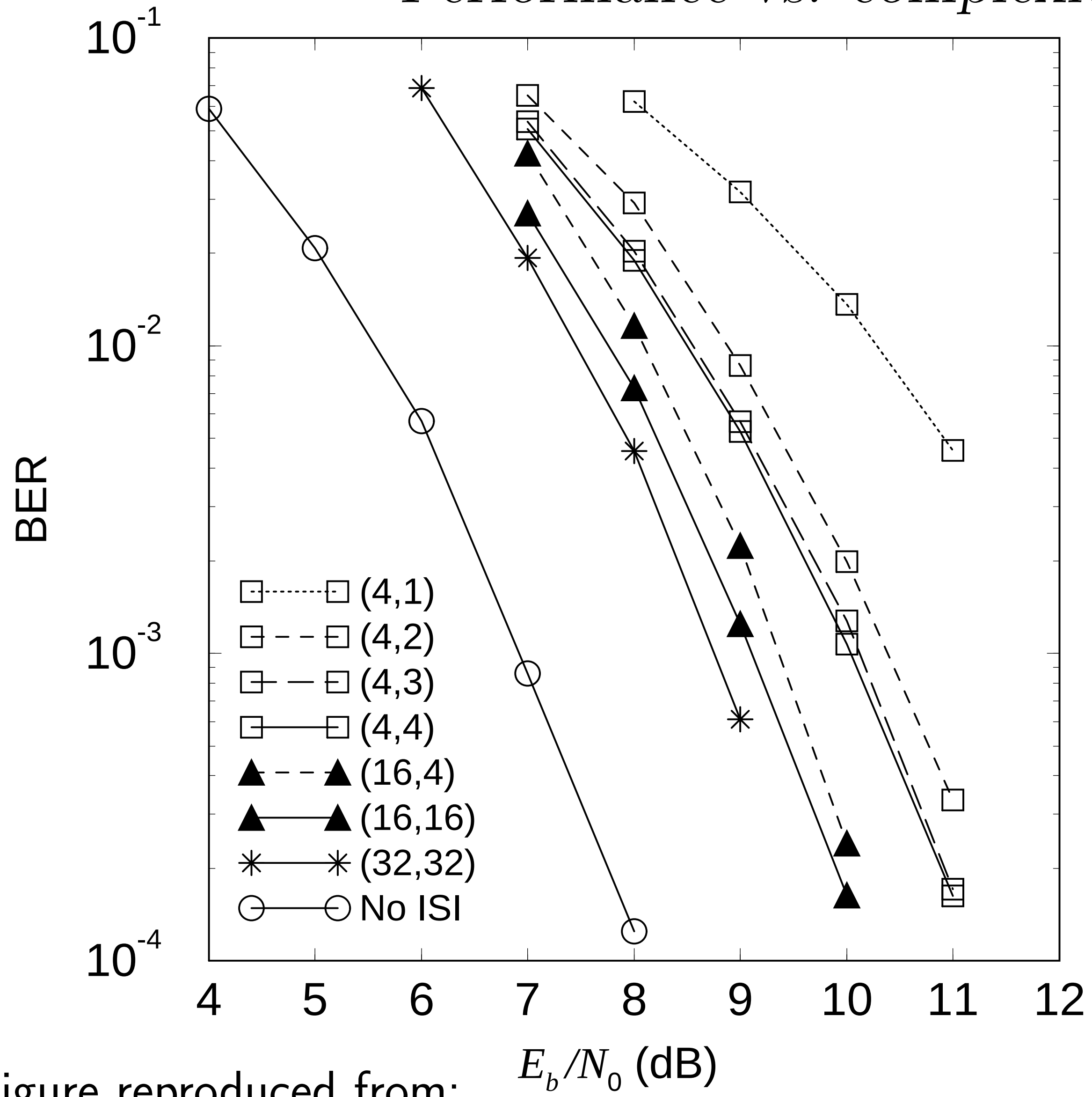
$$\Omega(1) = = \mathcal{C} = \{C_0, C_1, C_2, C_3\} \quad (J_1 = 4)$$
$$\Omega(2) = \Omega(3) = \mathcal{A} \quad (J_2 = J_3 = 1)$$
$$\omega_k'''' = (\mu_k; I_{k-1}(1)) \qquad S'''' = S_c \frac{J_1}{2} = 4 \times 2 = 8$$

5. Partial state by memory truncation with $Q = 0$ (code trellis)

$$\omega_k''''' = \mu_k \qquad S''''' = S_c = 4$$

TCM on ISI channel

Performance vs. complexity for RSSD



- TC-16QAM
- 4-tap channel
- reduced-estimator PSP with (S, N)
- $S = 2048$: full combined code/ISI trellis
- $S = 32$: reduced combined code/ISI trellis (case 1)
- $S = 16$: reduced combined code/ISI trellis (case 2)
- $S = 4$: code trellis (case 5)
- Reference curve for no ISI

Figure reproduced from:

– R. Raheli, G. Marino, P. Castoldi, "Per-survivor processing and tentative decisions: what is in between?," *IEEE Trans. Commun.*, pp. 127-129, Feb. 1996.

Reduced-search algorithms
Motivation

Reduced-search (or sequential) algorithms may be used to search a small part of a large FSM trellis diagram or non-FSM tree diagram

As opposed to state-complexity reduction, the original full-complexity trellis (or tree) diagram is searched in a partial fashion

These algorithms date back to the pre-Viterbi algorithm era. They were first proposed for decoding convolutional codes. The denomination “sequential” emphasizes the “novelty” compared to the then-established algebraic decoding of block codes

These algorithms can be applied to any system characterized by large memory or state complexity (if a FSM model hold)

If optimal processing is infeasible, any type of suboptimal processing may deserve our attention. Ranking of suboptimal solutions is difficult because of lacking of reference criteria

RSSD must be considered but an alternative among many others

Reduced-search algorithms
A general formulation of breadth-first detection

Assume a FSM model hold and let S be the number of states (full-size trellis)

Partition the S states into C (disjoint) classes

Mantain B paths per class selecting those which maximize the APPs under the constraint imposed by the partition rule and class structure

The resulting search algorithm may be denoted as SA(B,C)

Special cases:

$B > 1$ and $C = S$ $\Rightarrow$ list Viterbi algorithms with B survivors per state
$B = 1$ and $C = S$ $\Rightarrow$ classical Viterbi algorithm
$B = 1$ and $C < S$ $\Rightarrow$ RSSD with C states
$B > 1$ and $C < S$ $\Rightarrow$ list RSSD with B survivors per state and C states
$B = M$ and $C = 1$ $\Rightarrow$ M-algorithm

Reduced-search algorithms
A general formulation of breadth-first detection (cntd)

Whenever $C < S$, PSP allows the branch metrics to be defined

PSP also allows the above formalization to be applied when an FSM model does not hold ($S \to \infty$)

Define the complexity level as the total number of paths being traced, i.e., BC

Imposing a constraint on complexity, i.e., $BC \leq \eta$, constrained optimality can be defined

According to this criterion, the M-algorithm is considered the constrained optimal search algorithm

– T. Aulin, "Breadth-first maximum likelihood sequence detection: basics," *IEEE Trans. Commun.*, pp. 208-216, Feb. 1999.

Complexity reduction
Bibliography

– J. B. Anderson, S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. 32, pp. 169-176, Feb. 1984.
– J. W. M. Bergmans, S. A. Rajput, F. A. M. Van De Laar, "On the use of Decision Feedback for Simplifying the Viterbi Decoder," *Philips Journal of Research*, No. 4, 1987.
– K. Wesolowski, "Efficient Digital Receiver Structure for Trellis-Coded Signals Transmitted through Channels with Intersymbol Interference," *Electronics Letters*, Nov. 1987.
– T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," *IEEE Trans. Inform. Theory*, pp. 866-876, Nov. 1987.
– M. V. Eyuboğlu, S. U. H. Qureshi, "Reduced-State Sequence Estimation with Set Partition and Decision Feedback," *IEEE Trans. Commun.*, pp. 13-20, Jan. 1988.
– A. Duel Hallen, C. Heegard, "Delayed Decision-Feedback Sequence Estimation," *IEEE Trans. Commun.*, pp. 428-436, May 1989.
– P. R. Chevillat, E. Eleftheriou, "Decoding of Trellis-Encoded Signals in the Presence of Intersymbol Interference and Noise," *IEEE Trans. Commun.*, pp. 669-676, July 1989.
– M. V. Eyuboğlu, S. U. H. Qureshi, "Reduced-State Sequence Estimation for Coded Modulation on Intersymbol Interference Channels," *IEEE J. Sel. Areas Commun.*, pp. 989-995, Aug. 1989.

Complexity reduction
Bibliography (cntd)

– J. B. Anderson, "Limited search trellis decoding of convolutional codes," *IEEE Trans. Inform. Theory*, pp. 944-955, Sept. 1989.

– S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. Commun.*, vol. 38, pp. 3-12, Jan. 1990.

– A. Svensson, "Reduced state sequence detection of full response continuous phase modulation," *IEE Electronics Letters*, pp. 652 -654, 1 May 1990.

– J. B. Anderson, E. Offer, "Reduced-state sequence detection with convolutional codes," *IEEE Trans. Inform. Theory*, pp. 965-972, May 1994.

– R. Raheli, A. Polydoros, C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.

– R. Raheli, G. Marino, P. Castoldi, "Per-survivor processing and tentative decisions: what is in between?," *IEEE Trans. Commun.*, pp. 127-129, Feb. 1996.

– T. Aulin, "Breadth-first maximum likelihood sequence detection: basics," *IEEE Trans. Commun.*, pp. 208-216, Feb. 1999.

4. Classical applications of PSP

4.2 Linear predictive detection for fading channels

Linear modulation on flat fading channel
System model

Model of discrete observable:

$$r_k = f_k \, c_k + w_k$$

$\{f_k\}$: circular complex Gaussian random sequence
$\{c_k\}$: code sequence
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

Coding rule:

$$\begin{cases} c_k = o(a_k, \mu_k) \\ \mu_{k+1} = t(a_k, \mu_k) \end{cases}$$

μ_k: encoder state

Conditional statistics of the observation are Gaussian

Linear modulation on flat fading channel

Does a FSM model hold?

Conditional statistics of the observation:

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = \frac{1}{\pi\bar{\sigma}_k^2(\mathbf{a}_0^k)} \exp\left[-\frac{|r_k - \bar{r}_k(\mathbf{a}_0^k)|^2}{\bar{\sigma}_k^2(\mathbf{a}_0^k)}\right]$$

Conditional mean

$$\bar{r}_k(\mathbf{a}_0^k) = E\{r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\}$$

Conditional variance

$$\bar{\sigma}_k^2(\mathbf{a}_0^k) = E\{|r_k - \bar{r}_k(\mathbf{a}_0^k)|^2\,|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k\}$$

The conditional mean and variance depend on the entire previous code sequence:

$\Rightarrow$ unlimited memory

Linear predictive detection
Markov assumption

These receivers are based on the approximation

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) \simeq p(r_k|\mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k)$$

where integer $\nu > 0$ must be sufficiently large

Intuitive motivation: "old" observations do not add up much information to the current observation, given the immediately preceding ones

If this condition were strictly met, the random sequence r_k would be Markov of order ν, conditionally upon $\mathbf{a}_0^k$

This Markov assumption is never verified in an exact sense for realistic fading models. Even assuming a Markov fading model, thermal noise destroys the Markovianity in the observation.

The quality of this approximation depends on the autocovariance sequence of the fading process f_k and the value of ν, which is an important design parameter

Linear predictive detection
Problem 10

Assume a first-order autoregressive fading model

$$f_{k+1} = \sqrt{1-\rho^2}\, f_k + \rho\, v_k$$

$0 \leq \rho \leq 1$ is a constant

$\{v_k\}$ is an i.i.d. zero-mean Gaussian sequence with variance σ_v^2

A. Show that the fading sequence is Markov of first order

Assume f_0 is Gaussian with variance σ_v^2

B. Show that $\{f_k\}$ is a stationary Gaussian sequence

C. Check if the conditional observation $\{r_k\}$ satisfies a Markov property.

Linear predictive detection
Conditional observation

For Markovian observation, we may concentrate on

$$p(r_k|\mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k) = \frac{1}{\pi \hat{\sigma}_k^2(\mathbf{a}_0^k)} \exp\left[-\frac{|r_k - \hat{r}_k(\mathbf{a}_0^k)|^2}{\hat{\sigma}_k^2(\mathbf{a}_0^k)}\right]$$

The conditional mean and variance

$$\begin{aligned} \hat{r}_k(\mathbf{a}_0^k) &= E\{r_k|\mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k\} \\ \hat{\sigma}_k^2(\mathbf{a}_0^k) &= E\{|r_k - \hat{r}_k(\mathbf{a}_0^k)|^2 \, |\mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k\} \end{aligned}$$

are the ν-th order mean-square prediction of current observation r_k, given the previous ν observations and the information sequence, and the relevant prediction error, respectively

Note the difference with respect to the previously introduced notation $\bar{r}_k(\mathbf{a}_0^k)$ and $\bar{\sigma}_k^2(\mathbf{a}_0^k)$, which denoted similar quantities given the entire previous observation history $\mathbf{r}_0^{k-1}$ (k-th order prediction at time k)

Linear predictive detection

Linear prediction

For zero-mean Gaussian random sequences, the conditional mean (i.e., the mean-square prediction) is linear in the observation

$$\hat{r}_k(\mathbf{a}_0^k) = E\{r_k | \mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k\} = \sum_{i=1}^{\nu} p_{i,k}(\mathbf{a}_0^k)\, r_{k-i}$$

At time k, the linear prediction coefficients $p_{i,k}$ and the mean-square prediction error $\hat{\sigma}_k^2$ are the solution of the (linear) matrix equation (Wiener-Hopf)

$$\mathbf{R}_k(\mathbf{a}_0^k)\, \mathbf{p} = \mathbf{q}$$

where

$$\begin{aligned} \mathbf{R}_k(\mathbf{a}_0^k) &= E\left\{ \mathbf{r}_{k-\nu}^k (\mathbf{r}_{k-\nu}^k)^H \middle| \mathbf{a}_0^k \right\} \\ \mathbf{p} &= [1, -p_1, -p_2, \ldots, -p_\nu]^T \\ \mathbf{q} &= (\hat{\sigma}^2, \underbrace{0, \ldots, 0}_{\nu \text{ zeros}})^T \end{aligned}$$

The observation correlation matrix $\mathbf{R}_k(\mathbf{a}_0^k)$ incorporates the dependence on the data sequence $\mathbf{a}_0^k$ and vectors $\mathbf{p}$ and $\mathbf{q}$ include the unknowns

Linear predictive detection

Finite-memory condition

Given the flat fading model, the observation vector can be expressed as

$$\mathbf{r}_{k-\nu}^{k} = \mathbf{C}_k \, \mathbf{f}_{k-\nu}^{k} + \mathbf{w}_{k-\nu}^{k}$$

where $\mathbf{C}_k = \text{diag}(\mathbf{c}_{k-\nu}^{k})$

$$\begin{aligned}
\mathbf{R}_k(\mathbf{a}_0^k) &= E\left\{ \mathbf{r}_{k-\nu}^{k} (\mathbf{r}_{k-\nu}^{k})^H \middle| \, \mathbf{a}_0^k \right\} \\
&= E\left\{ \left[\mathbf{C}_k \, \mathbf{f}_{k-\nu}^{k} + \mathbf{w}_{k-\nu}^{k} \right] \left[(\mathbf{f}_{k-\nu}^{k})^H \, \mathbf{C}_k^H + (\mathbf{w}_{k-\nu}^{k})^H \right] \middle| \, \mathbf{a}_0^k \right\} \\
&= \mathbf{C}_k \, \mathbf{F} \, \mathbf{C}_k^H + \sigma_w^2 \, \mathbf{I} \; = \; \mathbf{R}(c_k, \zeta_k)
\end{aligned}$$

and $\mathbf{F} = E\left\{ \mathbf{f}_{k-\nu}^{k} (\mathbf{f}_{k-\nu}^{k})^H \right\}$ is the fading correlation matrix, which does not depend on k assuming *stationary fading*

The dependence of $\mathbf{R}_k(\mathbf{a}_0^k)$ on the information sequence can be compacted into the code sequence $\mathbf{c}_{k-\nu}^{k}$, hence in a suitably defined transition with state

$$\zeta_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-\nu})$$

Linear predictive detection

Finite-memory condition (cntd)

Since

$$\mathbf{R}_k(\mathbf{a}_0^k) = \mathbf{R}(c_k, \zeta_k)$$

a similar dependence characterizes the prediction coefficients, the conditional mean and variance, and the entire conditional statistics of the observation

$$\begin{aligned} p_i(\mathbf{a}_0^k) &= p_i(c_k, \zeta_k) \\ \hat{r}_k(\mathbf{a}_0^k) &= \hat{r}_k(c_k, \zeta_k) = \sum_{i=1}^{\nu} p_i(c_k, \zeta_k)\, r_{k-i} \\ \hat{\sigma}_k^2(\mathbf{a}_0^k) &= \hat{\sigma}^2(c_k, \zeta_k) \\ p(r_k|\mathbf{r}_{k-\nu}^{k-1}, \mathbf{a}_0^k) &= p(r_k|\mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k) \end{aligned}$$

where unnecessary time indexes can be dropped assuming a stationary fading regime

We may conclude that

Markov assumption $\Rightarrow$ finite memory

Linear predictive detection

Problem 11

Assume $\{x_k\}$ is a stationary time-discrete random process with autocorrelation sequence $E\{x_{k+m}x_k^*\} = \rho_m$

Let $\hat{x}_k = \sum_{i=1}^{\nu} p_i x_{k-i}$ denote the linear prediction of x_k, given the previous ν samples $x_{k-1}, x_{k-2}, \ldots, x_{k-\nu}$. The prediction coefficients $\{p_i\}_{i=1}^{\nu}$ minimize the mean-square prediction error $\hat{\sigma}^2 = E\{|x_k - \hat{x}_k|^2\}$

A. Show that the prediction coefficients $\{p_i\}_{i=1}^{\nu}$ and the minimum prediction error $\hat{\sigma}^2$ are the solution of the Wiener-Hopf equation

$$\mathbf{X}\,\mathbf{p} = \mathbf{q}$$

where

$$\begin{aligned} \mathbf{X} &= E\left\{\mathbf{x}_{k-\nu}^{k}\left(\mathbf{x}_{k-\nu}^{k}\right)^{H}\right\} \\ \mathbf{p} &= [1, -p_1, -p_2, \ldots, -p_\nu]^T \\ \mathbf{q} &= (\hat{\sigma}^2, \underbrace{0, \ldots, 0}_{\nu \text{ zeros}})^T \end{aligned}$$

B. Show that the linear prediction coincides with the mean-square prediction when $\{x_k\}$ is a zero-mean Gaussian sequence

Linear predictive detection

Branch metrics

The resulting branch metrics are

$$
\begin{aligned}
\gamma_k(c_k, \zeta_k) &= \ln p(r_k | \mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k) + \ln P[a_k(c_k, \zeta_k)] \\
&\propto -\frac{|r_k - \hat{r}_k(c_k, \zeta_k)|^2}{\hat{\sigma}^2(c_k, \zeta_k)} - \ln \hat{\sigma}^2(c_k, \zeta_k) + \ln P[a_k(c_k, \zeta_k)] \\
&= -\frac{|r_k - \sum_{i=1}^{\nu} p_i(c_k, \zeta_k)\, r_{k-i}|^2}{\hat{\sigma}^2(c_k, \zeta_k)} - \ln \hat{\sigma}^2(c_k, \zeta_k) + \ln P[a_k(c_k, \zeta_k)]
\end{aligned}
$$

They are based on linear predictions $\hat{r}_k(c_k, \zeta_k)$ of the current observation r_k based on the previous observations and *path-dependent* prediction coefficients

Linear predictive detection
An interpretation

Based on the conditional Gaussianity of the observation and the Markov assumption, we can concentrate on the Gaussian p.d.f. $p(r_k|\mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k)$

The conditional mean $\hat{r}_k(c_k, \zeta_k)$ and variance $\hat{\sigma}^2(c_k, \zeta_k)$ can be viewed as *system parameters* to be estimated

1. Adopt a linear feedforward data-aided parameter estimator of order ν (see Section 2)
2. Use a set of estimators by associating one estimator to each trellis path
3. Compute the estimation coefficients in order to minimize the mean-square estimation error with respect to the random variable r_k, conditionally on the path data sequence

⇒ The resulting estimator is the described *path-dependent* linear predictor

Linear prediction of r_k based on the previous observations is a form of PSP-based *feedforward* parameter estimation

We obtained it *naturally* in the derivation of the detection algorithm

Linear predictive detection
Alternative formulation of the branch metrics

The observation prediction can be expressed as

$$\begin{aligned}\hat{r}_k(c_k,\zeta_k) &= E\{r_k|\mathbf{r}_{k-\nu}^{k-1},c_k,\zeta_k\} = E\{f_k c_k + w_k|\mathbf{r}_{k-\nu}^{k-1},c_k,\zeta_k\} \\ &= c_k E\{f_k|\mathbf{r}_{k-\nu}^{k-1},c_k,\zeta_k\} = c_k\hat{f}_k(c_k,\zeta_k) \\ \hat{f}_k(c_k,\zeta_k) &= \sum_{i=1}^{\nu} p_i''(c_k,\zeta_k)\,\frac{r_{k-i}}{c_{k-i}(\zeta_k)}\end{aligned}$$

$\hat{f}_k(c_k,\zeta_k)$ denote *path-dependent* linear predictions of the fading coefficient at time k, based on previous observations

$p_i''(c_k,\zeta_k)$ are *path-dependent* linear prediction coefficients of the fading process based on previous observations of noisy *fading-like path-dependent* sequences $\{r_i/c_i(\zeta_k)\}_{i=k-\nu}^{k-1}$

$$\Rightarrow \quad p_i(c_k,\zeta_k) = p_i''(c_k,\zeta_k)\frac{c_k}{c_{k-i}(\zeta_k)}$$

Linear predictive detection

Alternative formulation of the branch metrics (cntd)

The mean-square prediction error of observation and fading are similarly related by

$$\begin{aligned}\sigma^2(c_k,\zeta_k) &= E\{|r_k - \hat{r}_k(c_k,\zeta_k)|^2 | \mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k\} \\ &= E\{|f_k c_k + w_k - c_k \hat{f}_k(c_k,\zeta_k)|^2 | \mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k\} \\ &= |c_k|^2 \epsilon^2(c_k,\zeta_k) + \sigma_w^2 \\ \epsilon^2(c_k,\zeta_k) &= E\{|f_k - \hat{f}_k(c_k,\zeta_k)|^2 | \mathbf{r}_{k-\nu}^{k-1}, c_k, \zeta_k\}\end{aligned}$$

The branch metrics can be expressed as

$$\begin{aligned}\gamma_k(c_k,\zeta_k) &= -\frac{|r_k - c_k \hat{f}_k(c_k,\zeta_k)|^2}{|c_k|^2\epsilon^2(c_k,\zeta_k)+\sigma_w^2} - \ln\left[|c_k|^2\epsilon^2(c_k,\zeta_k)+\sigma_w^2\right] \\ &+ \ln P[a_k(c_k,\zeta_k)]\end{aligned}$$

Linear predictive detection
An interpretation of the alternative formulation

The observation model $r_k = f_k c_k + w_k$ satisfies a parameter-conditional finite memory property by viewing f_k as an undesired parameter (see Section 2)

For estimating this parameter we could:

1. Adopt a linear feedforward data-aided parameter estimator of order ν (see Section 2)
2. Use a set of estimators by associating one estimator to each trellis path
3. Compute the estimation coefficients in order to minimize the mean-square estimation error with respect to the random variable r_k/c_k, conditionally on the path data sequence

$\Rightarrow$ The resulting estimator is exactly the described *path-dependent* linear predictor

Linear prediction of f_k based on the previous observations is a form of PSP-based *feedforward* parameter estimation

Linear predictive detection

Computation of the fading prediction coefficients

The fading prediction coefficients $p''_i(c_k, \zeta_k)$ and mean-square prediction error $\epsilon^2(c_k, \zeta_k)$ are the solution of the following Wiener-Hopf equation

$$\left(\mathbf{F} + \sigma_w^2 \tilde{\mathbf{C}}_k\right) \mathbf{p}'' = \mathbf{q}''$$

where

$$\begin{aligned}
\mathbf{F} &= E\left\{\mathbf{f}_{k-\nu}^{k} \left(\mathbf{f}_{k-\nu}^{k}\right)^H\right\} \\
\tilde{\mathbf{C}}_k &= \text{diag}\left(\frac{1}{|c_k|^2}, \frac{1}{|c_{k-1}|^2}, \ldots, \frac{1}{|c_{k-\nu}|^2}\right) \\
\mathbf{p}'' &= \left[1, -p''_1, -p''_2, \ldots, -p''_\nu\right]^T \\
\mathbf{q}'' &= (\epsilon^2 + \frac{\sigma_w^2}{|c_k|^2}, \underbrace{0, \ldots, 0}_{\nu \text{ zeros}})^T
\end{aligned}$$

The dependence of the solution on the hypothetical sequence is only through the moduli of the code symbols

For code symbols with constant modulus, the solution is *path-independent*

Linear predictive detection
Special case: Coded PSK

For code symbols with constant modulus $|c_k| = 1$ (e.g., PSK), the fading prediction coefficients p_i'' and mean-squared prediction error ϵ^2 are path-independent

The branch metrics simplify as

$$\begin{aligned}\gamma_k(c_k, \zeta_k) &= -\left|r_k - c_k \hat{f}_k(\zeta_k)\right|^2 + \ (\epsilon^2 + \sigma_w^2) \ln P[a_k(c_k, \zeta_k)] \\ &= -\left|r_k - c_k \sum_{i=1}^{\nu} p_i'' \frac{r_{k-i}}{c_{k-i}(\zeta_k)}\right|^2 + \ (\epsilon^2 + \sigma_w^2) \ln P[a_k(c_k, \zeta_k)]\end{aligned}$$

This solution is remarkably similar to what we would obtain in a decomposed estimation-detection design by estimating the "undesired" parameter f_k according to PSP

The (Gaussian) prediction error variance ϵ^2 affects the "overall" thermal noise power

Linear predictive detection
Problem 12

Consider the mean-square prediciton of f_k given the previous ν fading samples $\mathbf{f}_{k-\nu}^{k-1}$ and let $\{p'_i\}_{i=1}^{\nu}$ denote the linear prediction coefficients for this problem

A. Show that the fading prediction coefficients $\{p''_i(c_k, \zeta_k)\}_{i=1}^{\nu}$ equal $\{p'_i\}_{i=1}^{\nu}$ in the limit of vanishing noise power

Consider the first-order autoregressive fading model described in Problem 10 in a stationary regime and a constellation of unit-modulus code symbols

B. Show that the ν-th order prediction coefficients $\{p'_i\}_{i=1}^{\nu}$ satisfy $p'_1 = \rho$, $p'_2 = \cdots = p'_\nu = 0$

C. Show that the ν-th order prediction coefficients $\{p''_i\}_{i=1}^{\nu}$ are not zero for $\sigma_w^2 \neq 0$

Linear predictive detection
State-complexity reduction

The state-complexity of a linear predicition receiver can be naturally decoupled from the prediction order ν by means of state reduction techniques

For simplicity we consider folding by memory truncation, but set partitioning could be used as well

Let $Q < \nu$ denote the memory parameter to be taken into account in the definition of reduced trellis state

$$\omega_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-Q})$$

The branch metrics can be obtained by defining a pseudo state

$$\tilde{\zeta}_k(\omega_k) = (\underbrace{\mu_k; c_{k-1}, \ldots, c_{k-Q}}_{\omega_k}, \breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-\nu}(\omega_k))$$

$\breve{c}_{k-Q-1}(\omega_k), \ldots, \breve{c}_{k-L}(\omega_k)$ are the code symbols in the survivor of state ω_k

Linear predictive detection
State-complexity reduction: branch metrics

The branch metrics in the reduced-state trellis can be defined as usual according to

$$\tilde{\gamma}_k(c_k, \omega_k) = \gamma_k(c_k, \tilde{\zeta}_k(\omega_k))$$

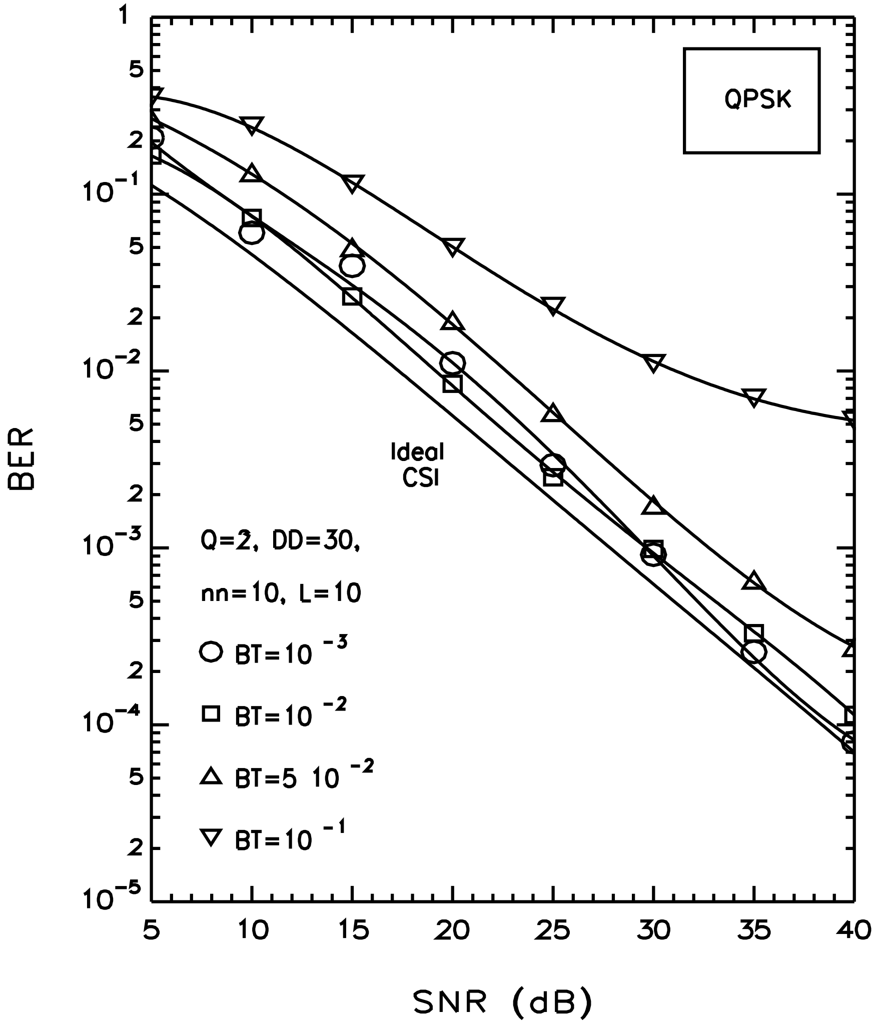
For *coded PSK* the branch metrics are

$$\begin{aligned}\tilde{\gamma}_k(c_k, \omega_k) &= -\left|r_k - c_k \hat{f}_k(\tilde{\zeta}_k(\omega_k))\right|^2 + (\epsilon^2 + \sigma_w^2) \ln P[a_k(c_k, \omega_k)] \\ &= -\left|r_k - c_k \sum_{i=1}^{Q} p_i'' \frac{r_{k-i}}{c_{k-i}(\omega_k)} - c_k \sum_{i=Q+1}^{\nu} p_i'' \frac{r_{k-i}}{\breve{c}_{k-i}(\omega_k)}\right|^2 \\ &\quad + (\epsilon^2 + \sigma_w^2) \ln P[a_k(c_k, \omega_k)]\end{aligned}$$

The prediction order ν and assumed memory Q are design parameters to be jointly optimized by experiment to yield a good compromise between performance and complexity

Linear predictive detection

Performance vs. ideal CSI



- QPSK ($M = 4$)
- time-varying flat Rayleigh fading
- BT : max Doppler rate
- $\nu = 10$, $Q = 2$ (16 states)
- Periodically inserted pilot symbols (one every 9 data symbols)
- Reference curve for ideal channel state information (CSI)

Figure reproduced from:

– G. M. Vitetta, D. P. Taylor, "Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels," *IEEE Trans. Commun.*, vol. 43, pp. 2750-2758, Nov. 1995

Linear predictive detection
Bibliography

– J. Lodge, M. Moher, “ML estimation of CPM signals transmitted over Rayleigh flat fading channels,” *IEEE Trans. Commun.*, pp. 787-794, June 1990.

– D. Makrakis, P. T. Mathiopoulos, D. P. Bouras, “Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: a combined envelope, multiple differential and coherent detection approach,” *IEEE Trans. Commun.*, pp.63-75, Jan. 1994.

– Q. Dai, E. Shwedyk, “Detection of bandlimited signals over frequency selective Rayleigh fading channels,” *IEEE Trans. Commun.*, pp. 941-950, Feb.-Apr. 1994.

– X. Yu, S. Pasupathy, “Innovations-based MLSE for Rayleigh fading channels,” *IEEE Trans. Commun.*, pp. 1534-Feb.-Apr. 1995.

– G. M. Vitetta, D. P. Taylor, “Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels,” *IEEE Trans. Commun.*, vol. 43, pp. 2750-2758, Nov. 1995

– G. M. Vitetta, D. P. Taylor, U. Mengali, “Double filtering receivers for PSK signals transmitted over Rayleigh frequency-flat fading channels,” *IEEE Trans. Commun.*, vol. 44, pp. 686-695, June 1996.

Linear predictive detection
Bibliography (cntd)

– D. M. Matolak, S. G. Wilson, "Detection for a statistically known, time-varying dispersive channel," vol. 44, pp. 1673-1683, Dec. 1996.

– M. E. Rollins, S. J. Simmons, "Simplified per-survivor Kalman processing in fast frequency-selective fading channels," *IEEE Trans. Commun.*, pp. 544-553, May 1997.

– P. Castoldi, R. Raheli, "On recursive optimal detection of linear modulations in the presence of random fading", *European Trans. Telecommun. (ETT)*, vol. 9, no. 2, pp. 209-220, March-April 1998.

– G. Colavolpe, P. Castoldi, R. Raheli, "Linear predictive receivers for fading channels", *IEE Electronics Letters*, vol. 34, no. 13, pp. 1289-1290, 25th June 1998.

4. Classical applications of PSP

4.3 Adaptive detection

Outline

Adaptive detection
Motivation

Channel model parameters can be time-varying (e.g., carrier phase, timing epoch, and channel impulse response)

A receiver based on the estimation-detection decomposition must be able to *track* these time variations, *provided they are not too fast*

The receiver must *adapt* itself to the time-varying channel conditions

PSP may be useful in *adaptive receivers*:

a) The per-survivor estimator associated with the best survivor is derived from data information which can be perceived as high-quality zero-delay decisions

$\Rightarrow$ Useful in *fast* time-varying channels

b) Many hypothetical data sequences are simultaneously considered in the parameter estimation process

$\Rightarrow$ Acquisition without training (*blind*) may be facilitated

Adaptive detection

PSP-based feedforward parameter estimation

Assume a parameter-conditional FSM model with state σ_k

PSP-based feedforward data-aided parameter estimator at time k

$$\breve{\boldsymbol{\theta}}_k(\sigma_k) = \mathbf{p}\left[\mathbf{r}_{k-\nu}^{k-1}, \breve{\mathbf{c}}_0^{k-1}(\sigma_k)\right]$$

Function of the ν most recent signal observations $\mathbf{r}_{k-\nu}^{k-1}$ and the per-survivor aiding data sequence $\breve{\mathbf{c}}_0^{k-1}(\sigma_k)$

Branch metrics at time k

$$\gamma_k(a_k, \sigma_k) = \ln p\left[r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k, \breve{\boldsymbol{\theta}}_k(\sigma_k)\right] + \ln P(a_k)$$

Update of the parameter estimator at time $k+1$

$$\breve{\boldsymbol{\theta}}_{k+1}(\sigma_{k+1}) = \mathbf{p}\left[\mathbf{r}_{k-\nu+1}^{k}, \breve{\mathbf{c}}_0^{k}(\sigma_{k+1})\right]$$

These estimates are simply recomputed for the new observation vector $\mathbf{r}_{k-\nu+1}^{k}$ and each new survivor sequence $\breve{\mathbf{c}}_0^{k}(\sigma_{k+1})$

An example is the linear predictive receiver for fading channels

Adaptive detection
PSP-based feedback parameter estimation

Assume a parameter-conditional FSM model with state σ_k

PSP-based feedback data-aided parameter estimator $\breve{\boldsymbol{\theta}}_k(\sigma_k)$

Branch metrics

$$\gamma_k(a_k, \sigma_k) = \ln p\left[r_k|\mathbf{r}_0^{k-1}, a_k, \sigma_k, \breve{\boldsymbol{\theta}}_k(\sigma_k)\right] + \ln P(a_k)$$

Update of the parameter estimator

$$\breve{\boldsymbol{\theta}}_{k+1}(\sigma_{k+1}) = \mathbf{q}\left[\breve{\boldsymbol{\theta}}_{k-\xi+1}^{k}(\sigma_k), \mathbf{r}_{k-\nu+1}^{k}, \breve{\mathbf{c}}_0^k(\sigma_{k+1})\right]$$

These estimates are computed for the ν most recent observations $\mathbf{r}_{k-\nu+1}^{k}$ and each new survivor sequence $\breve{\mathbf{c}}_0^k(\sigma_{k+1})$

The previous ξ parameter values $\breve{\boldsymbol{\theta}}_{k-\xi+1}^{k}(\sigma_k)$ are those associated with the survivors of states σ_k in the transitions $(\sigma_k \longrightarrow \sigma_{k+1})$ selected during the ACS step

Feedback parameter estimation is usually implied in adaptive receivers

Adaptive detection
Tentative decisions can be used

In feedforward and feedback parameter estimation, *tentative decisions* $\hat{\hat{\mathbf{c}}}_0^k$ can be used in place of the survivor data sequences $\breve{\mathbf{c}}_0^k(\sigma_{k+1})$ for updating the parameter estimate

The parameter estimate becomes *universal*, i.e., identical for all survivors

Formally, the updating recursions yield identical estimates for all survivors

The parameter estimator becomes *external* to the detection block

During *training* the correct data sequence would be used

Tracking of a dispersive time-varying channel
System model and notation

Model of linearly modulated discrete observable (slow variation)

$$r_k = \sum_{l=0}^{L} f_{l,k}\, c_{k-l} + w_k = \mathbf{f}_k^T \mathbf{c}_k + w_k$$

$\mathbf{f}_k = (f_{0,k}, f_{1,k}, \ldots, f_{L,k})^T$: overall time-varying discrete equivalent impulse response at the k-th instant

$\mathbf{c}_k = (c_k, c_{k-1}, \ldots, c_{k-L})^T$: code sequence with FSM model of state μ_k

$\sigma_k = (a_{k-1}, a_{k-2}, \ldots, a_{k-L}; \mu_{k-L})$: system state

$\mathbf{c}_k(a_k, \sigma_k) = [c_k(a_k, \mu_k), c_{k-1}(a_{k-1}, \mu_{k-1}), \ldots, c_{k-L}(a_{k-L}, \mu_{k-L})]^T$: code symbol vector uniquely associated with the considered trellis branch (a_k, σ_k), in accordance with the coding rule

Tracking of a dispersive time-varying channel
LMS adaptive identification

Least Mean Squares (LMS) adaptive identification

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \beta\,(r_{k+1-d} - \hat{\mathbf{f}}_k^T \mathbf{c}_{k+1-d})\,\mathbf{c}^*_{k+1-d}$$

β compromises between adaptation speed and algorithm stability

Branch metrics

$$\gamma_k(a_k, \sigma_k) = -|r_k - \hat{\mathbf{f}}_k^T \mathbf{c}_k(a_k, \sigma_k)|^2 + \sigma_w^2 \ln P(a_k)$$

In the (tentative) decision-directed tracking mode

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \beta\,(r_{k+1-d} - \hat{\mathbf{f}}_k^T \hat{\hat{\mathbf{c}}}_{k+1-d})\,\hat{\hat{\mathbf{c}}}^*_{k+1-d}$$

$$\hat{\hat{\mathbf{c}}}_{k+1-d} = (\hat{\hat{c}}_{k+1-d}, \hat{\hat{c}}_{k-d}, \ldots, \hat{\hat{c}}_{k+1-d-L})^T$$

$d \geq 1$ to comply with the causality condition upon the data

Tracking of a dispersive time-varying channel
PSP-based LMS adaptive identification

Branch metrics

$$\gamma_k(a_k, \sigma_k) = -|r_k - \breve{\mathbf{f}}_k(\sigma_k)^T \mathbf{c}_k(a_k, \sigma_k)|^2 + \sigma_w^2 \ln P(a_k)$$

Channel estimate update recursions

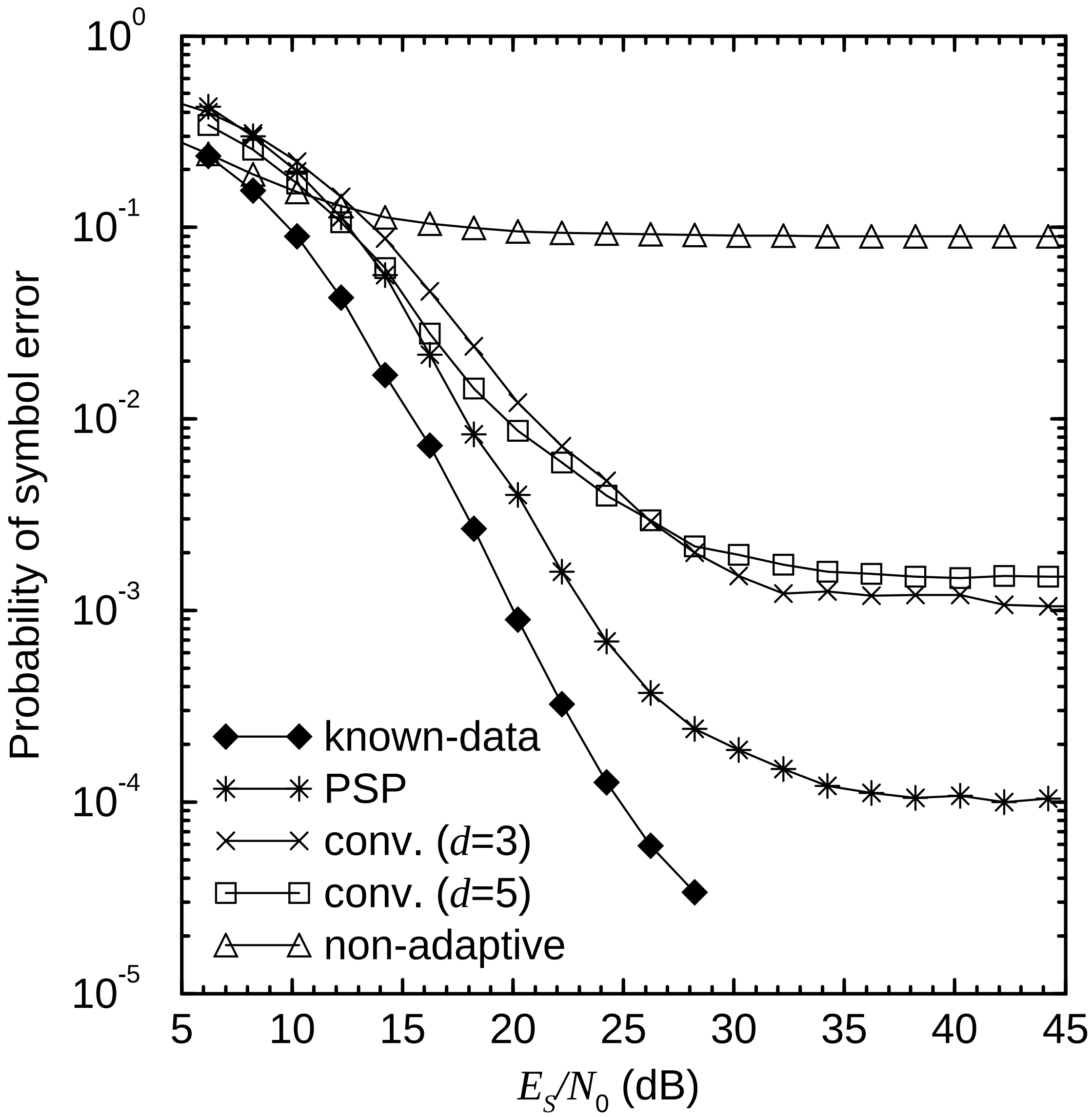
$$\breve{\mathbf{f}}_{k+1}(\sigma_{k+1}) = \breve{\mathbf{f}}_k(\sigma_k) + \beta \left[r_k - \breve{\mathbf{f}}_k(\sigma_k)^T \breve{\mathbf{c}}_k(a_k, \sigma_k) \right] \breve{\mathbf{c}}_k^*(a_k, \sigma_k)$$

$$\breve{\mathbf{c}}_k(a_k, \sigma_k) = [\breve{c}_k(a_k, \sigma_k), \breve{c}_{k-1}(\sigma_k), \ldots, \breve{c}_{k-L}(\sigma_k)]^T$$

The parameter estimate update recursions must take place along the transitions $(\sigma_k \to \sigma_{k+1})$ which extend the survivors of states σ_k, i.e., those selected during the ACS step at time k

ADAPTIVE DETECTION

LMS tracking of a dispersive fading channel



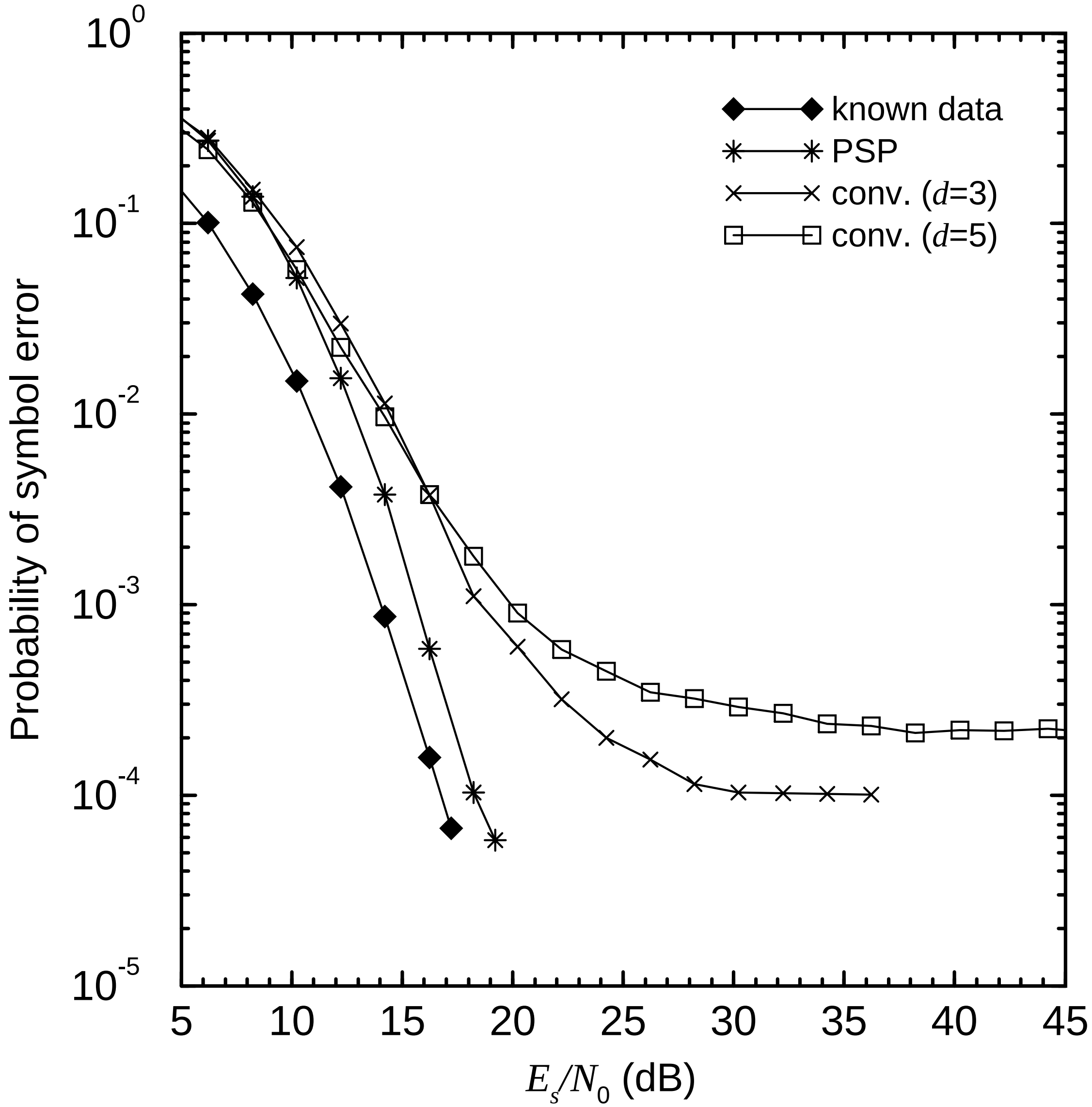
- QPSK ($M = 4$)
- Data blocks of 60 symbols
- Training preamble and tail
- Rayleigh fading channel with 3 independent tap weights
- Power delay profile (standard dev. of tap gains): $\frac{1}{\sqrt{6}}(1, 2, 1)$
- Doppler rate: $f_D T = 1.85 \times 10^{-3}$
  In the 1.8 GHz band:
  32.5 km/h with $1/T = 24.3$ kHz
  300 km/h with $1/T = 270.8$ kHz
- Full-state sequence detection: $Q = L = 2$ (16 states)

Figure reproduced from:

– R. Raheli, A. Polydoros, C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.

Adaptive detection

LMS tracking of a dispersive fading diversity channel



- QPSK ($M = 4$)
- Data blocks of 60 symbols
- Training preamble and tail
- Rayleigh fading channel with 3 independent tap weights
- Power delay profile (standard dev. of tap gains): $\frac{1}{\sqrt{6}}(1, 2, 1)$
- Doppler rate: $f_D T = 3.69 \times 10^{-3}$
  In the 1.8 GHz band:
  65 km/h with $1/T = 24.3$ kHz
  600 km/h with $1/T = 270.8$ kHz
- Dual diversity
- Full-state sequence detection: $Q = L = 2$ (16 states)

Joint detection and phase synchronization

System model

Model of linearly modulated discrete observable (slow variation)

$$r_k = e^{j\theta_k} c_k + w_k$$

θ_k : channel-induced phase rotation
$\{c_k\}$: code sequence with FSM model of state μ_k
$\{w_k\}$: i.i.d. Gaussian noise sequence with variance σ_w^2

First order data-aided Phase-Locked Loop (PLL)

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta \, \mathrm{Im} \left\{ r_{k+1-d} \, e^{-j\hat{\theta}_k} \, c^*_{k+1-d} \right\}$$

η controls the loop bandwidth

Joint detection and phase synchronization
Decision-directed phase tracking

Branch metrics

$$\gamma_k(a_k, \mu_k) = -|r_k e^{-j\hat{\theta}_k} - c_k(a_k, \mu_k)|^2 + \sigma_w^2 \ln P(a_k)$$

$c_k(a_k, \sigma_k)$: code symbol branch label

PLL phase-update (feedback) recursion

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \eta \, \mathrm{Im} \left\{ r_{k+1-d} \, e^{-j\hat{\theta}_k} \, \hat{\hat{c}}^*_{k+1-d} \right\}$$

The tentative decision delay must comply with the causality condition upon the detected data, which implies $d \geq 1$.

Joint detection and phase synchronization
PSP-based phase tracking

Branch metrics:

$$\gamma_k(a_k, \mu_k) = -\left|r_k\, e^{-j\breve{\theta}_k(\mu_k)} - c_k(a_k, \mu_k)\right|^2 + \sigma_w^2 \ln P(a_k)$$

Phase estimate update recursion:

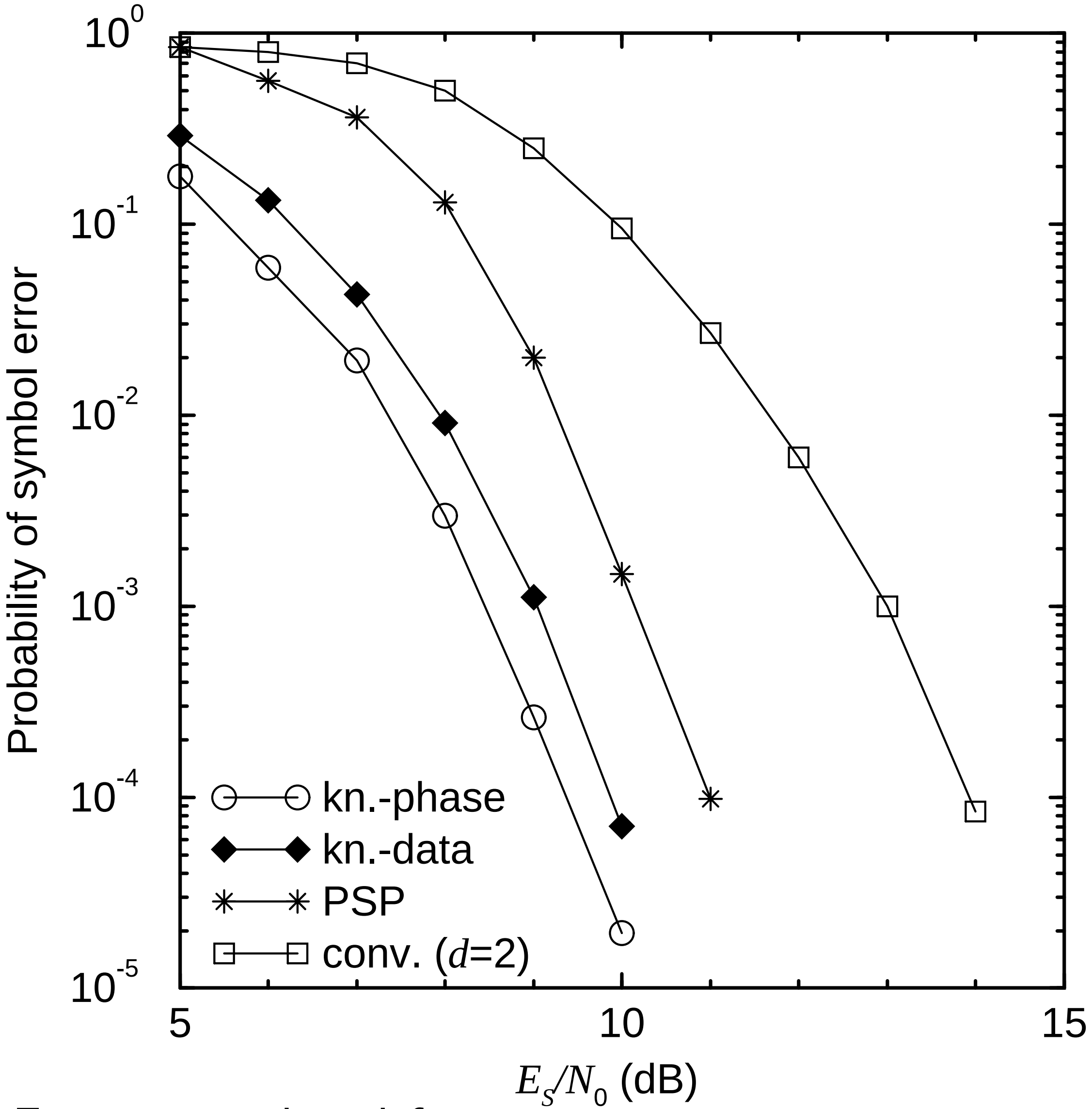
$$\breve{\theta}_{k+1}(\mu_{k+1}) = \breve{\theta}_k(\mu_k) + \eta\ \mathsf{Im}\left\{r_k\, e^{-j\breve{\theta}_k(\mu_k)}\, \breve{c}_k^*(a_k, \mu_k)\right\}$$

$\breve{c}_k^*(a_k \mu_k)$ is the code symbol associated with the transition (a_k, μ_k)

The phase estimate update recursions must take place along the transitions $(\mu_k \rightarrow \mu_{k+1})$ which extend the survivors of states μ_k, i.e., those selected during the ACS step at time k

Adaptive detection

Joint TCM decoding and phase synchronization



- TC-8PSK (4 states)
- Phase noise with Wiener model:

$$\theta_{k+1} = \theta_k + \Delta_k$$

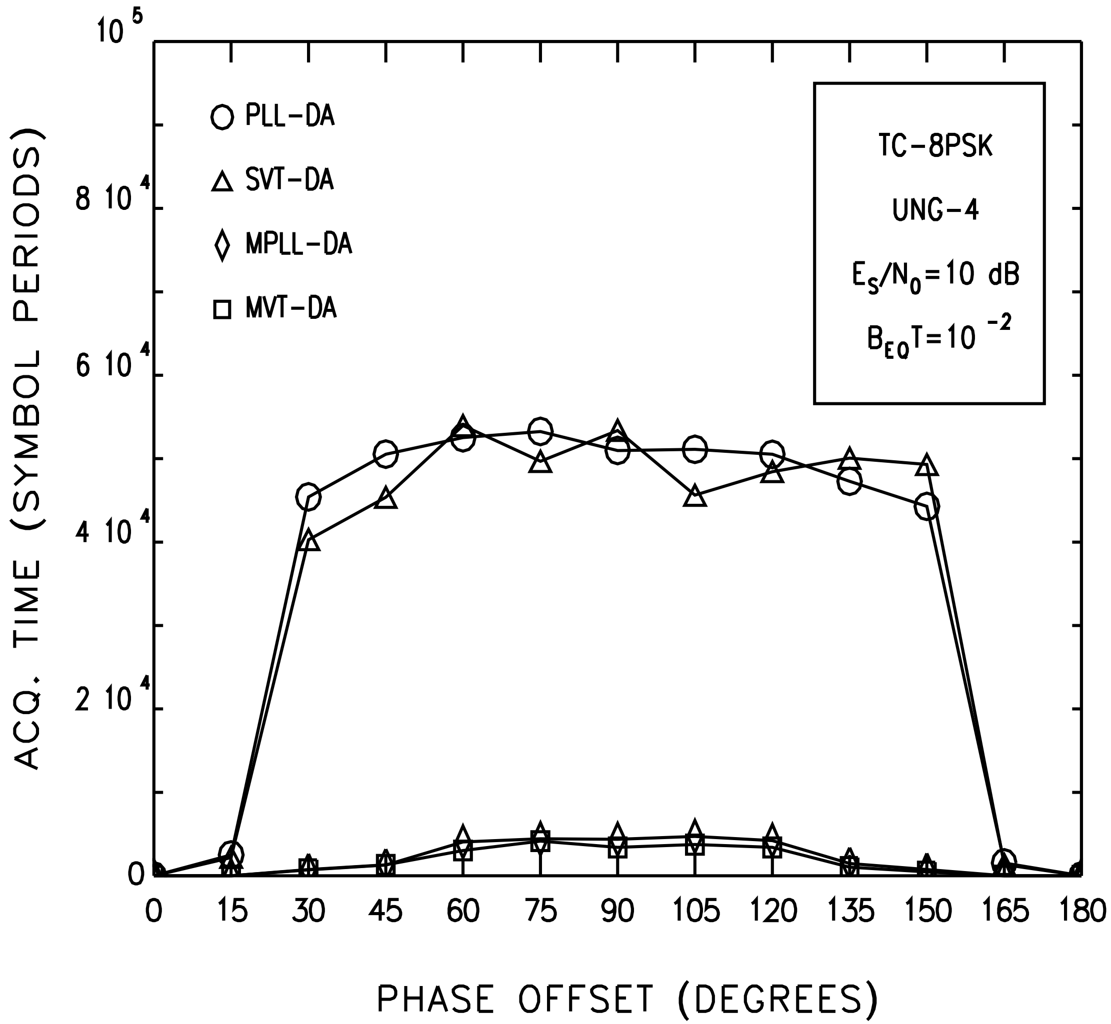
Δ_k are Gaussian, i.i.d. with standard deviation 2°

Figure reproduced from:

– R. Raheli, A. Polydoros, C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.

Adaptive detection

Phase acquisition of PLL and VT



- TC-8PSK (4 states)
- 1st order PLL
- 1st order vector tracker (VT)
- With system in lock, a phase step $\Delta\phi$ is applied at time zero
- Phase evolution is monitored until the phase error reduces to $\pm 10°$
- Acquisition time in symbol periods vs. $\Delta\phi$
- $E_s/N_0 = 10$ dB
- $B_{EQ}T = 10^{-2}$

Figure reproduced from:

– A. N. D'Andrea, U. Mengali, and G. M. Vitetta, "Approximate ML decoding of coded PSK with no explicit carrier phase reference," *IEEE Trans. Commun.*, pp. 1033-1039, Feb.-Apr. 1994.

Adaptive detection
Bibliography

– A. N. D'Andrea, U. Mengali, and G. M. Vitetta, "Approximate ML decoding of coded PSK with no explicit carrier phase reference," *IEEE Trans. Commun.*, pp. 1033-1039, Feb.-Apr. 1994.
– H. Kubo, K. Murakami, T. Fujino, "An adaptive MLSE for fast time-varying ISI channels," *IEEE Trans. Commun.*, pp, 1872-1880, Feb.-Apr. 1994.
– H. Kubo, K. Murakami, T. Fujino, "Adaptive MLSE by means of combined equalization and decoding in fading environments," *IEEE J. Sel. Areas Commun.*, pp, 102-109, vol. 13, Jan. 1995.
– R. Raheli, A. Polydoros, C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, pp. 354-364, Feb.-Apr. 1995.
– R. Raheli, G. Marino, P. Castoldi, "Per-survivor processing and tentative decisions: what is in between?," *IEEE Trans. Commun.*, pp. 127-129, Feb. 1996.
– K. M. Chugg, A. Polydoros, "MLSE for an unknown channel—Parts I and II," *IEEE Trans. Commun.*, pp. 836-846 and 949-958, July and Aug. 1996.
– K. M. Chugg, "Blind acquisition characteristics of PSP-based sequence detectors," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 1518-1529, Oct. 1998.

5. Advanced applications of PSP

Iterative detection

Motivation

Iterative, or *turbo*, detection/decoding was first proposed as a suboptimal algorithm for decoding special very powerful channel codes, widely known as *turbo codes*

Turbo codes are a parallel concatenation of simple component recursive convolutional codes through a long permuter, or *interleaver*

The principle of iterative detection/decoding can be applied to any parallel or serial concatenation of FSM models:

a) Each FSM model is detected/decoded by means of a suitable *soft-input soft-output* (SISO) module accounting for that model
b) The soft-outputs of the various modules are passed to other modules, which *refine* the detection/decoding process in a next iteration
c) The process can be iterated several times and usually converges in a few steps

Since the channel can be typically modeled as a FSM, exactly or approximately, joint iterative detection of the received signal and decoding of a possible channel code can be performed

Iterative detection
Soft-input soft-output (SISO) modules

A SISO module processes the soft-information received from other modules and combines it with the possible observation of the channel output

The input soft-information can be accounted for by assigning proper values to the *a priori* probabilities of the information or code symbols

⇒ This is the reason for having so diligently accounted for these probabilities in the various branch-metric expressions

In non-iterative detection, we are allowed to eliminate the a priori symbol probabilities from the very beginning, on the basis of the reasonable assumption that they have equal values (hence, they are irrelevant)

The output soft-information is computed on the basis of the APPs of the possible information or code symbols

Iterative detection
PSP-based SISO modules

A SISO module computes the APPs of the information symbols by means of a forward-backward (FB) or soft-output Viterbi algorithm

Soft-output Viterbi algorithms estimate a *reliability* value of any decision by comparing the metrics of best paths to those of their competitors

The max-log approximation of the FB algorithm allows a direct application of PSP to the two counter-running Viterbi algorithms (in direct and inverse time)

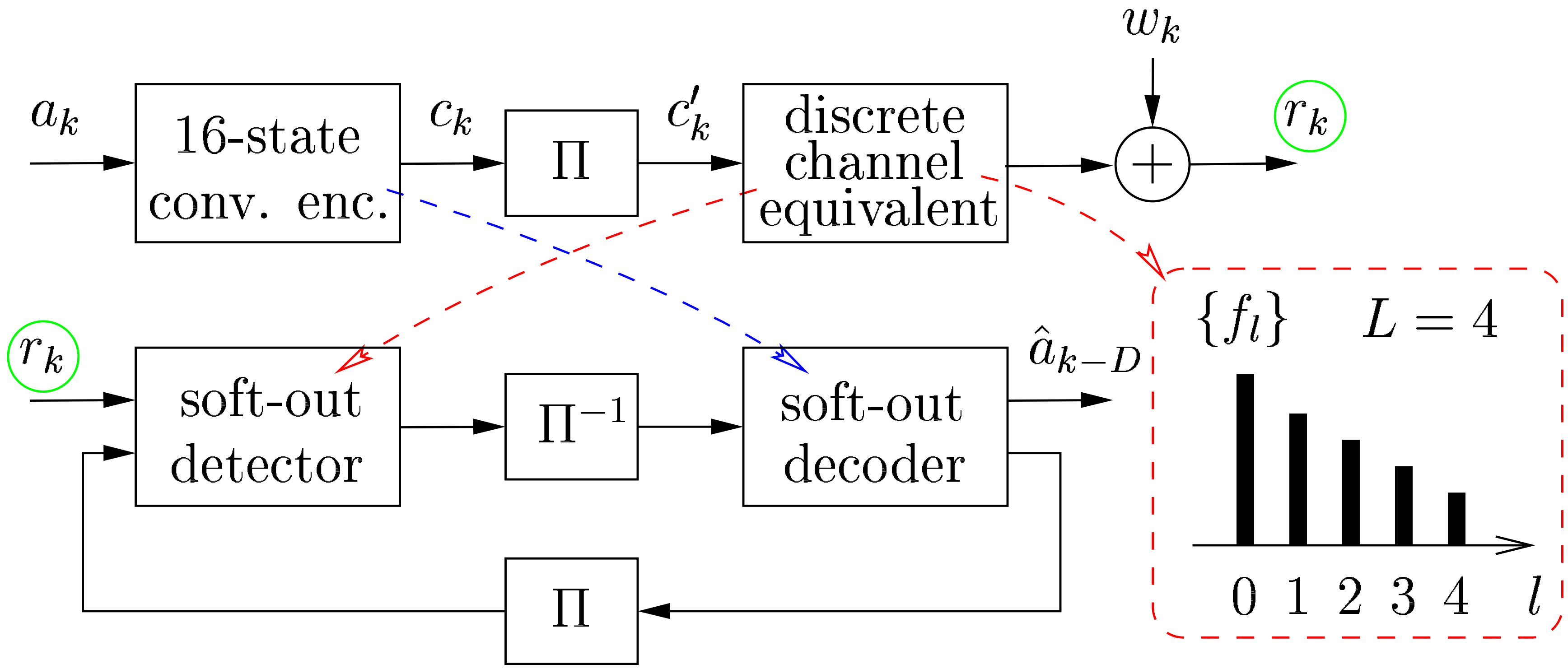
Soft-output Viterbi algorithms can be readily augmented with PSP

These remarks entitle us to exploit any possible application of PSP in the soft-output modules used in iterative decoding, e.g. for:

- Complexity reduction
- Linear predictive detection
- Adaptive detection

Concatenation of code and ISI channel

Reduced-state iterative detection/decoding



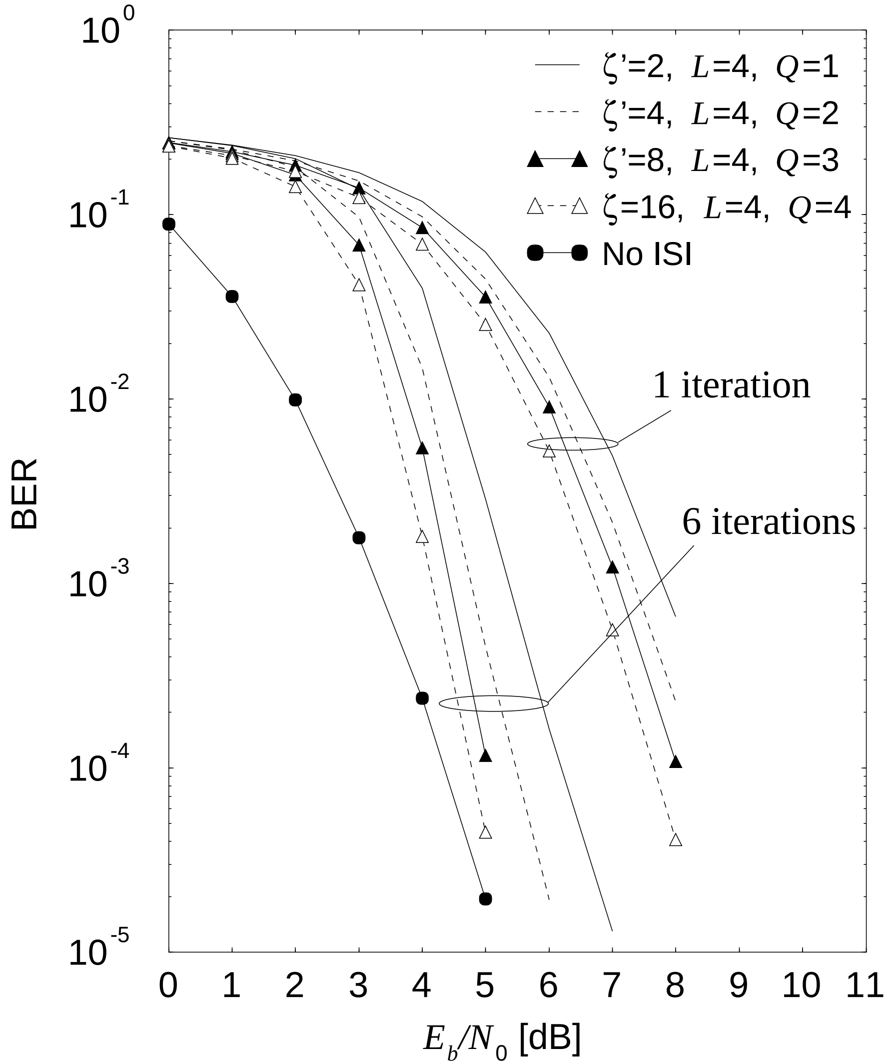
Soft-output detector based on FB with max-log approximation in the forward and backward recursions (not in the APP computation)

Exact FB decoder

Pseudo-random interleaver Π

Concatenation of code and ISI channel

Reduced-state iterative detection/decoding



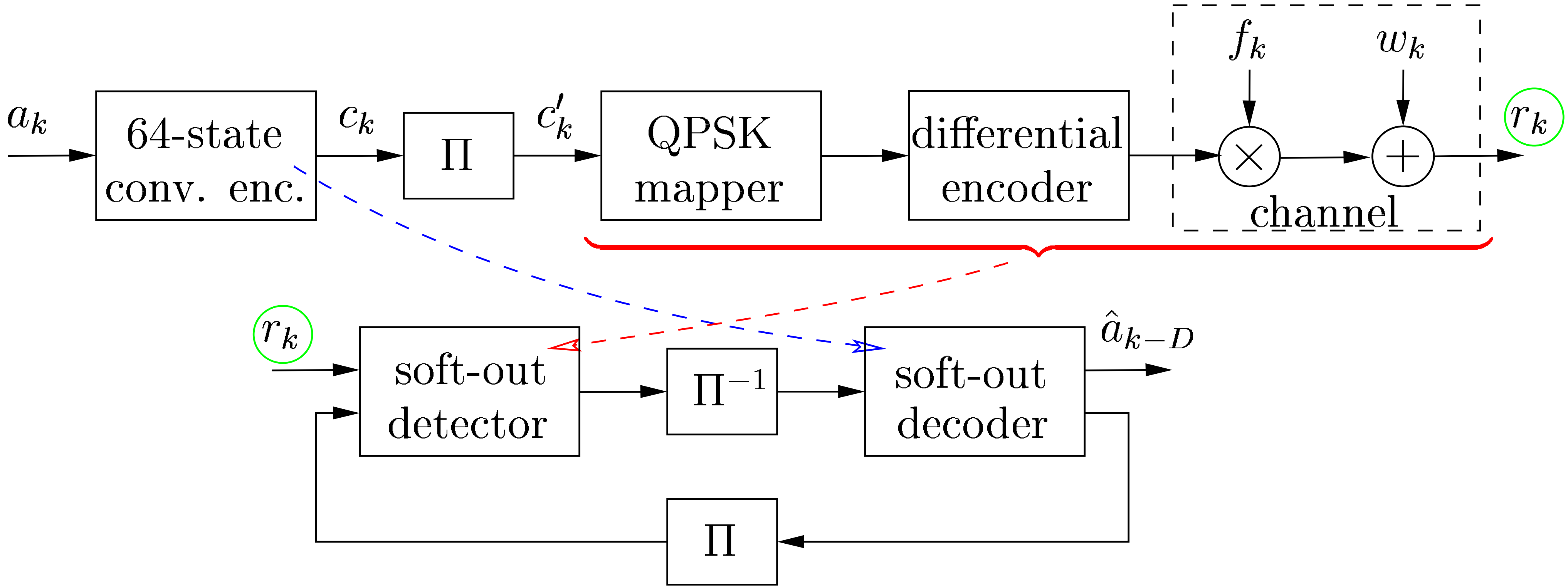
- Outer rate-1/2 16-state recursive systematic convolutional code (RSC)
- Code generators: $G_1 = (23)_8$ and $G_2 = (35)_8$
- 64×64 pseudo-random interleaver
- BPSK
- Known static ISI
- $\zeta' =$ number of states in reduced-state trellis
- 1 and 6 iterations
- Reference curve for ideal channel

Figure reproduced from:

– G. Colavolpe, G. Ferrari, R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 848-859, May 2001.

Concatenation of code and fading channel

Linear-predictive iterative detection/decoding



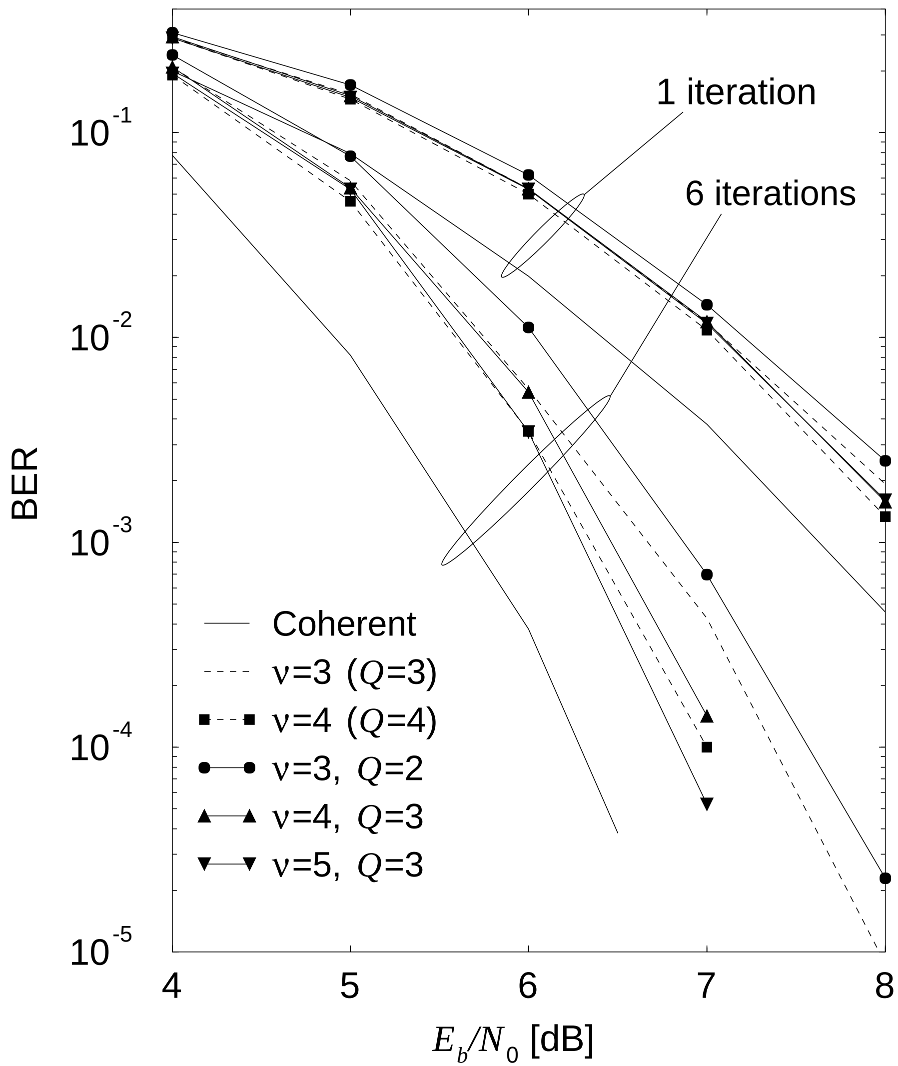
Soft-output detector based on FB with linear prediction and state reduction (bit APPs)

Exact FB decoder

Pseudo-random *bit* interleaver Π

Concatenation of code and fading channel

Linear-predictive iterative detection/decoding



- Outer rate-1/2 64-state non recursive non systematic convolutional code
- Code generators: $G_1 = (133)_8$ and $G_2 = (171)_8$
- 64×64 pseudo-random bit interleaver
- DQPSK
- Flat Rayleih fading
- Isotropic scattering model with normalized fading rate: $f_D T = 0.01$
- Various levels of complexity (ν, Q)
- Number of detector trellis states: $S = 4^{Q-1}$ (accounting for diff. enc.)

Figure reproduced from:

– G. Colavolpe, G. Ferrari, R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 848-859, May 2001.

Noncoherent detection
Motivation

In virtually any bandpass transmission system, the carrier phase reference is not known by the receiver

In coherent detection this phase reference must be recovered by the receiver, provided it is sufficiently stable, according to the synchronization-detection decomposition

Noncoherent detection assumes complete absence of knowledge about the phase reference—an effective approach if the phase is unstable

A nonchoerent channel introduces unlimited memory in the signal—suboptimal detection algorithms are in order

Noncoherent detection

Unlimited memory

Discrete channel model

$$r_k = c_k e^{j\theta} + w_k$$

The conditional p.d.f. of the observation is

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{a}_0^k) = \frac{p(\mathbf{r}_0^k|\mathbf{a}_0^k)}{p(\mathbf{r}_0^{k-1}|\mathbf{a}_0^{k-1})} = \frac{e^{-(|r_k|^2+|c_k|^2)/\sigma_w^2}}{\pi\sigma_w^2} \frac{I_0\left[\frac{2}{\sigma_w^2}\left|(\mathbf{r}_0^k)^H \mathbf{c}_0^k\right|\right]}{I_0\left[\frac{2}{\sigma_w^2}\left|(\mathbf{r}_0^{k-1})^H \mathbf{c}_0^{k-1}\right|\right]}$$

In fact

$$\begin{aligned} p(\mathbf{r}_0^k|\mathbf{a}_0^k) &= \frac{1}{2\pi}\int_0^{2\pi} p(\mathbf{r}_0^k|\mathbf{a}_0^k,\theta)d\theta \\ &= \frac{1}{2\pi(\pi\sigma_w^2)^{k+1}}\int_0^{2\pi} e^{-(||\mathbf{r}_0^k-\mathbf{c}_0^k e^{j\theta}||^2)/\sigma_w^2}\, d\theta \\ &= \frac{1}{2\pi(\pi\sigma_w^2)^{k+1}} e^{-(||\mathbf{r}_0^k||^2+||\mathbf{c}_0^k||^2)/\sigma_w^2} \int_0^{2\pi} e^{(2\mathrm{Re}\{(\mathbf{r}_0^k)^H \mathbf{c}_0^k e^{j\theta}\})/\sigma_w^2}\, d\theta \\ &= \frac{1}{(\pi\sigma_w^2)^{k+1}} e^{-(||\mathbf{r}_0^k||^2+||\mathbf{c}_0^k||^2)/\sigma_w^2}\, I_0\left[\frac{2}{\sigma_w^2}\left|(\mathbf{r}_0^k)^H \mathbf{c}_0^k\right|\right] \end{aligned}$$

Noncoherent detection
Feedforward PSP-based phase estimation

Data-aided mean-square phase estimate based on N most recent observations $\mathbf{r}_{k-N}^{k-1}$

$$e^{-j\hat{\theta}} = \frac{(\mathbf{r}_{k-N}^{k-1})^H \mathbf{c}_{k-N}^{k-1}}{\left|(\mathbf{r}_{k-N}^{k-1})^H \mathbf{c}_{k-N}^{k-1}\right|}$$

Branch metrics:

$$\begin{aligned}
\gamma_k(c_k, \mu_k) &= -\left| r_k\, e^{-j\breve{\theta}_k(\mu_k)} - c_k \right|^2 + \sigma_w^2 \ln P[a_k(c_k, \mu_k)] \\
&\propto \mathrm{Re}\left\{ r_k c_k^*\, e^{-j\breve{\theta}_k(\mu_k)} \right\} - \frac{|c_k|^2}{2} + \frac{1}{2}\sigma_w^2 \ln P[a_k(c_k, \mu_k)] \\
&= \frac{\mathrm{Re}\left\{ r_k c_k^* \cdot (\mathbf{r}_{k-N}^{k-1})^H \breve{\mathbf{c}}_{k-N}^{k-1}(\mu_k) \right\}}{\left|(\mathbf{r}_{k-N}^{k-1})^H \breve{\mathbf{c}}_{k-N}^{k-1}(\mu_k)\right|} - \frac{|c_k|^2}{2} + \frac{\sigma_w^2}{2} \ln P[a_k(a_k, \mu_k)]
\end{aligned}$$

The trellis state μ_k can be augmented to include part of the phase memory

$$\omega_k = (\mu_k; c_{k-1}, c_{k-2}, \ldots, c_{k-Q}) \quad Q \le N$$

Feedforward phase estimation
Problem 13

Consider the random-phase discrete channel model

$$r_k = c_k e^{j\theta} + w_k$$

Define a feedforward data-aided phase estimate $\hat{\theta}$ based on the previous N observations by minimizing the mean-square error

$$E\left\{ \left\| \mathbf{r}_{k-N}^{k-1} - \mathbf{c}_{k-N}^{k-1} e^{j\hat{\theta}} \right\|^2 \middle| \mathbf{c}_{k-N}^{k-1} \right\}$$

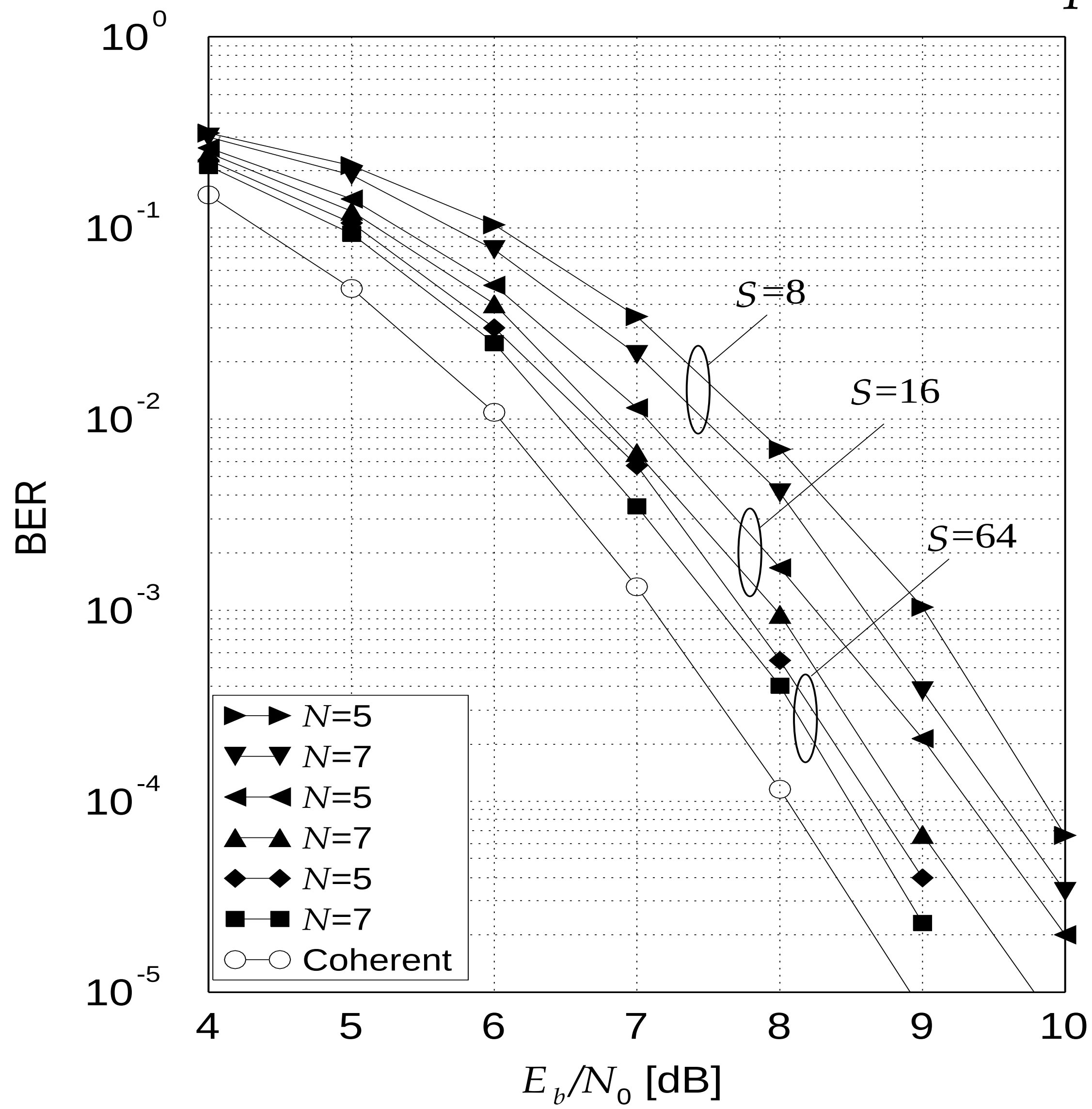
A. Show that this estimate must verify the condition

$$e^{-j\hat{\theta}} = \frac{\left(\mathbf{r}_{k-N}^{k-1}\right)^H \mathbf{c}_{k-N}^{k-1}}{\left| \left(\mathbf{r}_{k-N}^{k-1}\right)^H \mathbf{c}_{k-N}^{k-1} \right|}$$

B. Show that the result in part A coincides with the data-aided maximum-likelihood phase estimate based on the previous N observations

Noncoherent sequence detection

Feedforward PSP-based phase estimation



- Performance vs. ideal coherent detection (stable phase)
- 8-state TC-16QAM
- 90° rotational invariance with differential encoding
- Robust when phase is unstable

Figure reproduced from:

– G. Colavolpe, R. Raheli, "On noncoherent sequence detection of coded QAM", *IEEE Commun. Lett.*, vol. 2, pp. 211-213, August 1998.

Detection in MIMO systems
Motivation

Multiple-input multiple-output (MIMO) systems arise in a number of current scenarios:

a) Multiuser detection, or *code division multiple access* (CDMA), when the user of interest is interfered by other users due to non-orthogonal or non-synchrounous codes

b) Receive- and transmit-diversity systems, e.g., the well known *space-time* coded systems for fading channels

c) *Orthogonal frequency division multiplexing* (OFDM) currently used as a *modulation* scheme in many systems (xDSL, DAB, DVB, WLAN, . . .), just to mention a few

d) Information storage, such as magnetic or optical memories, e.g., due to the multitrack interference in magnetic recording systems

Detection in MIMO systems
The basic approach

MIMO systems are multidimensional versions of transmission (or storage) systems and can be described by a suitable vector or matrix notation

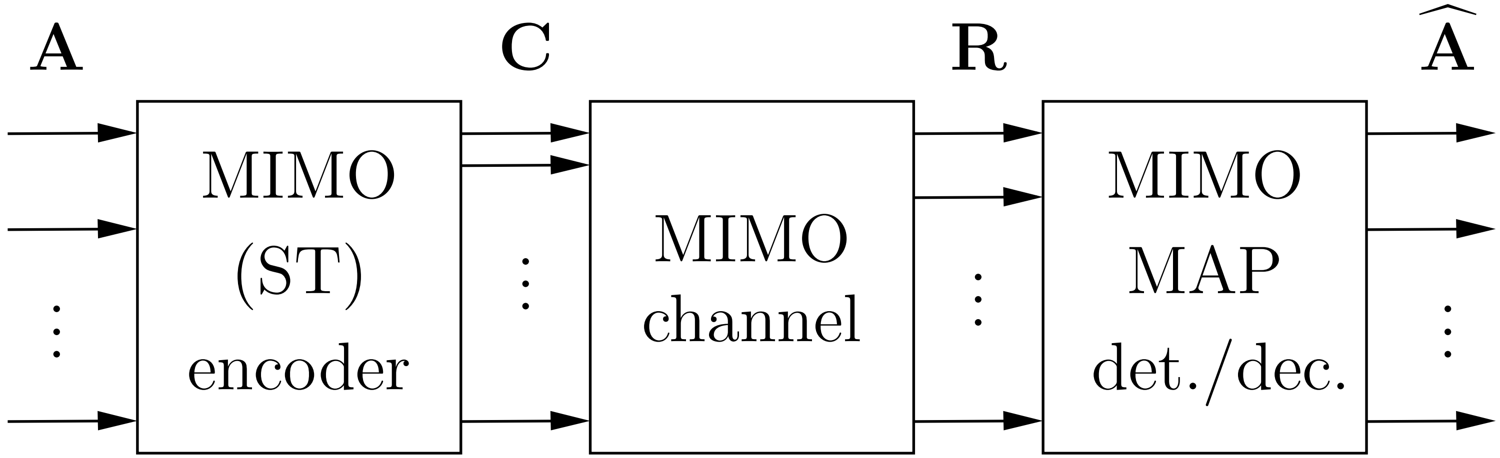
In some cases the increased system dimensionality can be exploited, e.g. in space-time coding

Most detection techniques can be applied to MIMO systems in a conceptually straightforward manner by a proper notational extension

The complexity may be an issue due to the increased dimensionality (not only notationwise)

Detection in MIMO systems

General system model



$\mathbf{A}$: input information matrix $(N \times K)$ (or $N' \times K$)

$\mathbf{C}$: code matrix $(N \times K)$

$\mathbf{R}$: received matrix $(M \times K)$

$\widehat{\mathbf{A}}$: detected information matrix $(N \times K)$ (or $N' \times K$)

Rows and columns may represent the "space" and time dimensions, respectively

Notation: for an $(N \times K)$ matrix $\mathbf{X}$, ${}_n\mathbf{X}_k$ is element (n, k), ${}_n\mathbf{X}$ is the n-th row, $\mathbf{X}_k$ is the k-th column, and ${}_{n_1}^{n_2}\mathbf{X}_{k_1}^{k_2}$ is an $(n_2{-}n_1) \times (k_2{-}k_1)$ submatrix

Detection in MIMO systems
MAP strategies

MAP *block* detection

$$\widehat{\mathbf{A}} = \operatorname*{argmax}_{\mathbf{A}} P(\mathbf{A}|\mathbf{R})$$

MAP *sequence* detection

$${}_n\widehat{\mathbf{A}}_0^{K-1} = \operatorname*{argmax}_{{}_n\mathbf{A}_0^{K-1}} P({}_n\mathbf{A}_0^{K-1}|\mathbf{R})$$

MAP *symbol* detection

$${}_n\widehat{\mathbf{A}}_k = \operatorname*{argmax}_{{}_n\mathbf{A}_k} P({}_n\mathbf{A}_k|\mathbf{R})$$

Finite memory systems

$$p(\mathbf{R}_k|\mathbf{R}_0^{k-1}, \mathbf{A}_0^k) = p(\mathbf{R}_k|\mathbf{R}_0^{k-1}, \mathbf{A}_k, \sigma_k)$$

where vectors $\mathbf{R}_k$ and $\mathbf{A}_k$ are the signal received and the information transmitted at time k (i.e., over "space"), respectively, and σ_k is a suitably defined system state

Advanced applications of PSP

Bibliography

Iterative detection:

– V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Selet. Areas Commun.*, vol. 16, pp. 186-195, February 1998.

– K. R. Narayanan, G. L. Stüber, "List decoding of turbo codes", *IEEE Trans. Commun.*, vol. 46, pp.754-762, June 1998.

– P. Hoeher, J. Lodge, "'Turbo DPSK': iterative differential PSK demodulation and channel decoding", *IEEE Trans. Commun.*, vol. 47, pp. 837-843, June 1999.

– G. Colavolpe, G. Ferrari, R. Raheli, "Noncoherent iterative (turbo) decoding", *IEEE Trans. Commun.*, vol. 48, pp. 1488-1498, September 2000.

– G. Colavolpe, G. Ferrari, R. Raheli, "Reduced-state BCJR-type algorithms," *IEEE J. Select. Areas Commun.* Special Issue-The Turbo Principle: from Theory to Practice, vol. 19, pp. 848-859, May 2001.

Noncoherent detection:

– G. Colavolpe, R. Raheli, "On noncoherent sequence detection of coded QAM", *IEEE Commun. Lett.*, vol. 2, pp. 211-213, August 1998.

– G. Colavolpe, R. Raheli, "Noncoherent sequence detection", *IEEE Trans. Commun.*, vol. 47, pp. 1376-1385, September 1999.

– G. Colavolpe, R. Raheli, "Noncoherent sequence detection of continuous phase modulations", *IEEE Trans. Commun.*, vol. 47, pp. 1303-1307, September 1999.

ADVANCED APPLICATIONS OF PSP
Bibliography (cntd)

Detection in MIMO systems:

– G. Paparisto, K. M. Chugg, A. Polydoros, "PSP array processing for multipath fading channels", *IEEE Trans. Commun.*, vol. 47, pp. 504-507, April 1999.

– G. Caire, G. Colavolpe, "On low-complexity space-time coding for quasi-static channels", in *IEEE Trans. Inform. Theory*, vol. 49, pp. 1400-1416, June 2003.

– E. Chiavaccini, G. M. Vitetta, "Further results on differential space-time modulations", *IEEE Trans. Commun.*, vol. 51 , pp. 1093-1101, July 2003.

Free-space optical communications:

– Xiaoming Zhu, J. M. Kahn, "Free-space optical communication through atmospheric turbulence channels", *IEEE Trans. Commun.*, vol. 50, pp. 1293-1300, Aug. 2002.

– Xiaoming Zhu, J. M. Kahn, "Markov chain model in maximum-likelihood sequence detection for free-space optical communication through atmospheric turbulence channels", *IEEE Trans. Commun.*, vol. 51, pp. 509-516, March 2003.

– Wei Mao, J. M. Kahn, "Free-space heterochronous imaging reception of multiple optical signals", *IEEE Trans. Commun.*, vol. 52, pp. 269-279, Feb. 2004.