MAP algorithm for a punctured code

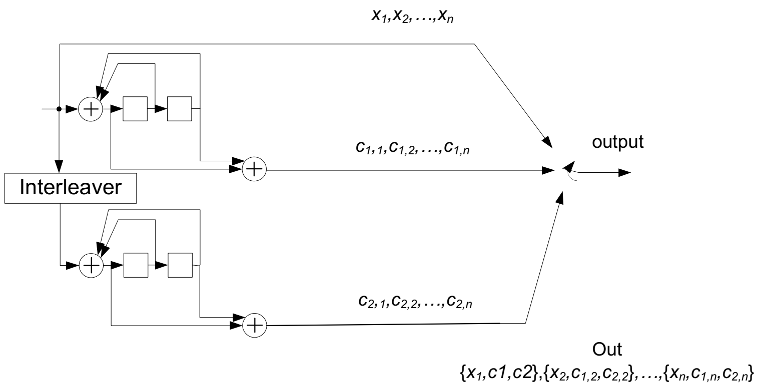


Figure: Non punctured encoder

MAP algorithm for a punctured code

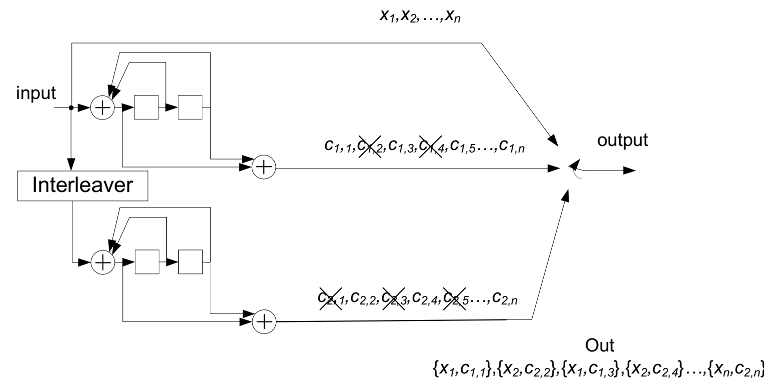


Figure: Punctured encoder

MAP trellis for a punctured code

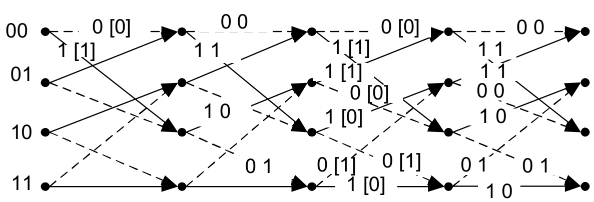


Figure: Punctured MAP for decoder 1

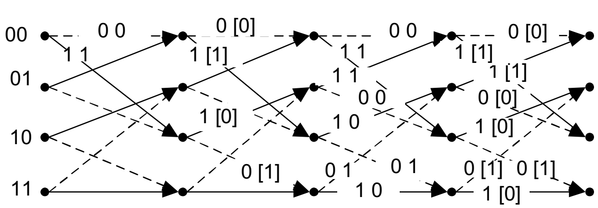


Figure: Punctured MAP for decoder 2

Turbo equalization

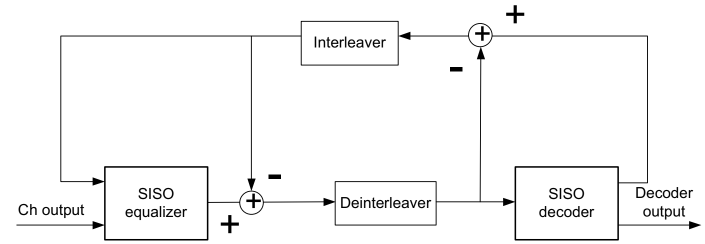
- Making the equalization based only on the cahnnel knowledge and obseserved values is suboptimal
- The equalization does not utilize the knowledge about the code structure
- Multipath channel can be interpreted as a spreading code
  - The spreading code can be described by a trellis structure
  - The ML equalizer on this trellis is the Viterbi algorithm
  - For one bit the optimal algorithm is marginalization by APP calculation
- Together the channel and the error correction code give a serially concatenated code
  - Serically concatenated codes can be decoded by applying turbo processing
- Turbo equalizer is an iterative equalization/decoding process

Soft equalizer

- For turbo equalization the equaliziation operation should
  - provide a soft values to the decoding block
  - be able to incorporate a priori values from the channel code decoder block to the equalization operation

- Soft trellis based equalizer
  - APP equalizer
  - SOVA
- Soft interference cancellation
  - Soft DFE
  - Soft sphere decoder

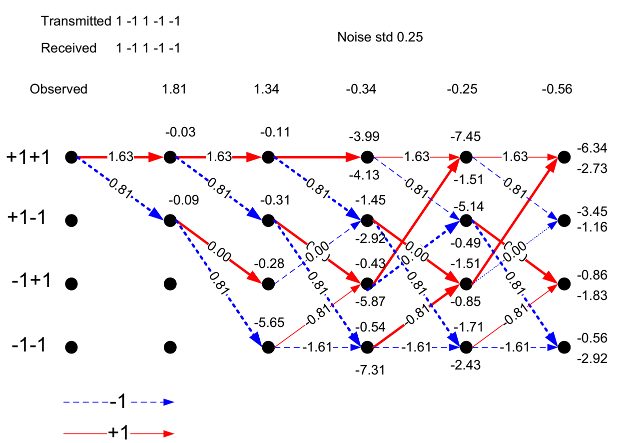
Turbo Equalizer



Turbo Equalizer [Douillard et al. 1995]

SISO: MAP, Log-MAP, Max-Log-MAP, SOVA, Soft DFE

Example



Trellis for the 3-path symbol spaced channel [0.407 0.815 0.407]

How to make an APP equalizer

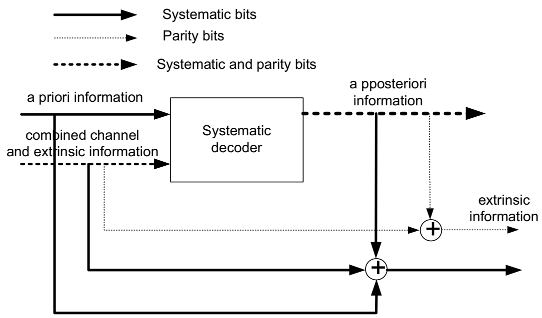
- For turbo decoding purposes we have to change the normal equalizer to an equalizer with Soft Input Soft Output.
- For creating the soft output we can use a soft trellis decoding
- For accepting soft input we have to consider that

  The bit impacts multiple stages in the trellis.

  Question: At which stage to combine the prior knowledge?

  It turns out that the prior information can be incorporated at whatever stage where this particular input bit is present in the trellis

Information flow for a soft equalizer



- A-priori (intrinsic) - information, known before decoding (equalisation), from other sources than received sequence or code constraints
- Extrinsic - information provided by decoder(equaliser), based on the received sequence and a-priori information of other bits
- A-posteriori - information generated by a SISO algorithm

Making MAP equalizer to Soft In eqaulizer

- In a stage k the edge metric is calculated based on the bits in the stage channel bit and channel coefficients

$$\prod_k p(y_k|x_k, s_k, h_{ch}) = \sum_k (y_k - f(x_k, s_k, h_{ch}))^2$$

Where

$$f(x_k, s_k, h_{ch}) = h_1 \cdot x_k + h_2 \cdot x_{k-1} + \cdots + h_L \cdot x_{k-(L-1)}$$

- For given extrinsic information we weight this metrics with the symbol probability

$$\sum_k (y_k - f(x_k, s_k, h_{ch}))^2 \cdot p(x_k = X)$$

where X is the symbol value corresponding to the particular edge in the trellis

Adaptive MLSE Equalizer

With the LMS algorithm the tap coefficients are updated

$$\hat{h}_i(k+1) = \hat{h}_i(k) + \alpha\epsilon_{k-Q}\hat{x}_{k-i-Q}$$

where α is the adaptation state size, Q is the decision delay, $Q > 5L$ for minimal performance degradatation, and error at epoch $k - Q$ is

$$\epsilon_{k-Q} = y_{k-Q} - \sum_{i=0}^{L} \hat{h}_i(k)\hat{x}_{k-i-Q}$$

Channel variations over Q degrade the tracking performance
Reducing Q reduce reliability of $\hat{x}_{k-i-Q}$
Solution - per-survivor processing

$$\hat{h}_i(k+1) = \hat{h}_i(k) + \alpha\epsilon_k\tilde{x}_{k-i}$$

where $\tilde{\mathbf{x}}$ is the surviving sequence for a state.
Each state uses individual channel estimator.

Soft Feedback Equalizer

- In order to reduce the complexity of MLSE equalizer we can reduce the number of states.
- Alternatively if the bits are known we can compensate the ISI from the neighbouring bits.
- If the bits are partly known we can compensate the interference partially.
- The amount knowledge about a bit can be expressed as a **soft bit**.
  - For a binary symbol the soft bit is a MMSE estimate of the bit.
- The equalization is made by substracting interference described by the soft bit and scaled by the corresponding channel tap.

System model

- We assume that at the front end the channel matched filter is applied.
  - The corresponind channel response h_l becomes symmetric with maximum at the center.
  - The lenght of such channel response is $2L-1$, where L corresponds to the amount of channel taps.

The received singal is modeled at the output of the matched filter

$$y_k = \sum_{l=-(L-1)}^{L-1} h_l x_{k-l} + n_k$$

Where x corresponds to the transmitted symbol
n is the additive Gaussian noise
y_k is the received symbol at the moment k

Probability of the bit

The probability of a sparticular bit in a mutlipath channel depends on the neighbouring symbol values and on the channel tap values. For one symbol k we have

$$\log P(y_1, \ldots, y_K | x_1, \ldots, x_K) =$$
$$= \log P(y_k | x_k, \ldots, x_{k-(L-1)}) + \log P(y_1, \ldots, y_{k-1}, y_{k+1}, \ldots y_K | x_1, \ldots, x$$

The second part in this equation decribes the state probability at the stage k and the first part in this equation impacts the transition in the trellis section k.
We can calcualte probability of each symbol separately.

$$\log P(y_K, \ldots, y_1 | x_K, \ldots, x_1) =$$
$$= \sum_{k=1}^{K} \ln P(y_k | x_1, \ldots, x_K) \approx \sum_{k=1}^{K} \log P(y_k | x_{k-(L-1)}, \ldots, x_{k+(L-1)})$$

In Gaussian channel the approximated symbol probability is

$$p(y_k | x_{k+L-1}, \ldots, x_{k-(L-1)}) =$$
$$\frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(- \frac{\left(y_k - \sum_{l=-(L-1)}^{L-1} h_l x_{k-l} \right)^2}{2\sigma_n^2} \right) \prod_{l=-(L-1)}^{L-1} p(x_l)$$

Difference compared to the full trellis based calcualtion is that we consider only one trellis section.

The previous state is defined by the symbol sequence
$x_{k-(L-1)} \cdots x_{k-1}$
The next state is defined by the symbol sequence $x_{k+1} \cdots x_{k+(L-1)}$
If we know these bit probablities we can calculate the states probabilities.
For example for a state [0 0]

$$p\left\{ \ x_{k-1} = 0, \quad x_{k-2} = 0 \ \right\} = p\left\{x_{k-1} = 0\right\} p\left\{x_{k-2} = 0\right\}$$

Here we assumed that the symbols x are independent.

Soft DFE

An suboptimal but well working method is to average first for each bit x and then calculate the posterior probability for y.
The averaging result is a soft bit $\hat{x}$.
The posterior probability for the y is

$$p(y_k|x_{k+L-1},\ldots,x_{k-(L-1)}) \approx p(y_k|\hat{x}_{k-(L-1)},\ldots,\hat{x}_{k+(L-1)}) =$$
$$\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left(-\frac{\left(y_k-\sum_{l=-(L-1)}^{L-1}h_l\hat{x}_{k-l}\right)^2}{2\sigma_n^2}\right)$$

In log domain

$$\log p(y_k|\hat{x}_{k-(L-1)},\ldots,\hat{x}_{k+(L-1)}) = \hat{y}_k^i = \hat{y}_k - \sum_{\substack{l=-(L-1),\\ l\neq 0}}^{L-1} h_l\hat{x}_{k-1}$$

$\hat{y}_k^i$ equalized symbol k at the iteration i.
$\hat{y}$ scaled received symbol value.
$\hat{x}$ soft symbol values (for a binary symbol value between -1 and 1).
After equalization we can treat the output as a new equalized observed value.

- At iteration i we denote a new channel *observed* value as $\hat{y}_k^i$.
- This observation is provided to the soft channed decoder.

If all the bits are known exactly the soft values are precise bit values and the Soft DFE becomes a classical DFE.

Soft bit

The soft bit describe the mean amplitude of the symbol For a binary case

$$\hat{s} = x(0)\cdot p\,(b=0) - x(1)\cdot p\,(b=1)$$

Where $x(0)$ and $x(1)$ stand for mapped value of input bit 0 and 1
In what follows we use a simple mapping
$b(0) \Rightarrow x(0) = 1 \qquad , b(1) \Rightarrow x(1) = -1$

In a Gaussian channel assume that we know the liglikelihood of the bit x_k a posterior probability at the iteration step i

$$llr_k^i = llr(x_k^i) = \log\frac{p(b=1)}{p(b=0)}$$

The soft bit is calculated as

$$\begin{aligned}\hat{x}_k^{i+1} &= \frac{e^{llr_k^i}}{1+e^{llr_k^i}} - \frac{1}{1+e^{llr_k^i}} = -\frac{1-e^{llr_k^i}}{1+e^{llr_k^i}}\\ &= \tanh\left(\frac{llr_k^i}{2}\right)\end{aligned}$$

SDF equalization algorithm

1. Based on the *llr* from the decoder output calculate the soft bit values.

   in first iteration $llr = 0$
2. Substract the interfering soft bits from the received symbol value

   This operation reduces ISI.
3. Calculate the probability for the bit value.

Calculation of output soft value

- The soft equalizer output should provide soft values to the next decoding operation.
- The soft equalized bit values contain remaining interference and noise.
  - We approximate this by Gaussian distribution.
- For calculating the output soft value we need the variance and mean of the output value.

The noisy value of y_k^i

$$y_k^i = h_0 x_k + \sum_{\substack{l=-(L-1) \\ l \neq 0}}^{L-1} h_l \left(x_{k-l} - \hat{x}_{k-l}\right) + n$$

Estimation of the mean of $\hat{y}^i$
Mean of y_k^i is $h_0 \cdot x_k$, since $x_{k-1} \in \{\pm 1\}$ the amplitude of the symbol is h_0.

Estimation of the variance
By assuming that all the bits are independent the variances of the terms

$$h_l \left(x_{k-l} - \hat{x}_{k-l}\right)$$

can be evaluated indenpendently.
Total varinace is the sum of the remaining interfernece variances

$$\hat{\sigma}^2 = \sigma_n^2 + \sum_{\substack{l=-(L-1) \\ l \neq 0}}^{L-1} h_l^2 \left(1 - \hat{x}_{k-l}^2\right)$$

$$p\left(y_k^i x_k\right) \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{\left(y_k^i - h_0 x_k\right)^2}{2\hat{\sigma}^2}\right)$$

$$\ln p\left(y_k^i | x_k\right) = \frac{h_0}{\hat{\sigma}^2} x_k y_k^i$$

The loglikelihood of the output bit is

$$\ln\left(y_k^i\right) = 2\frac{h_0}{\hat{\sigma}^2} y_k^i$$