MAP algorithm for a punctured code

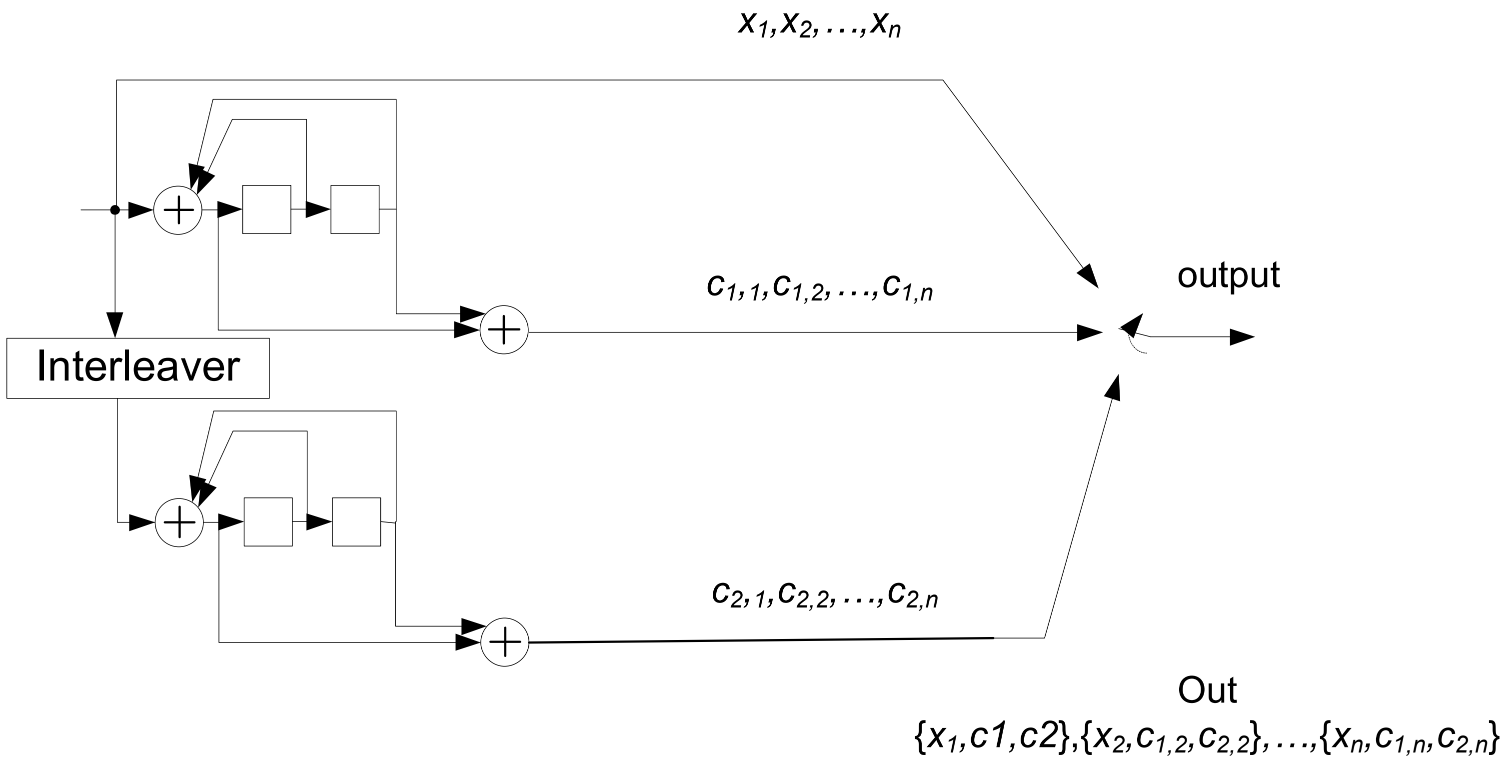


Figure 1: Non punctured encoder

MAP algorithm for a punctured code

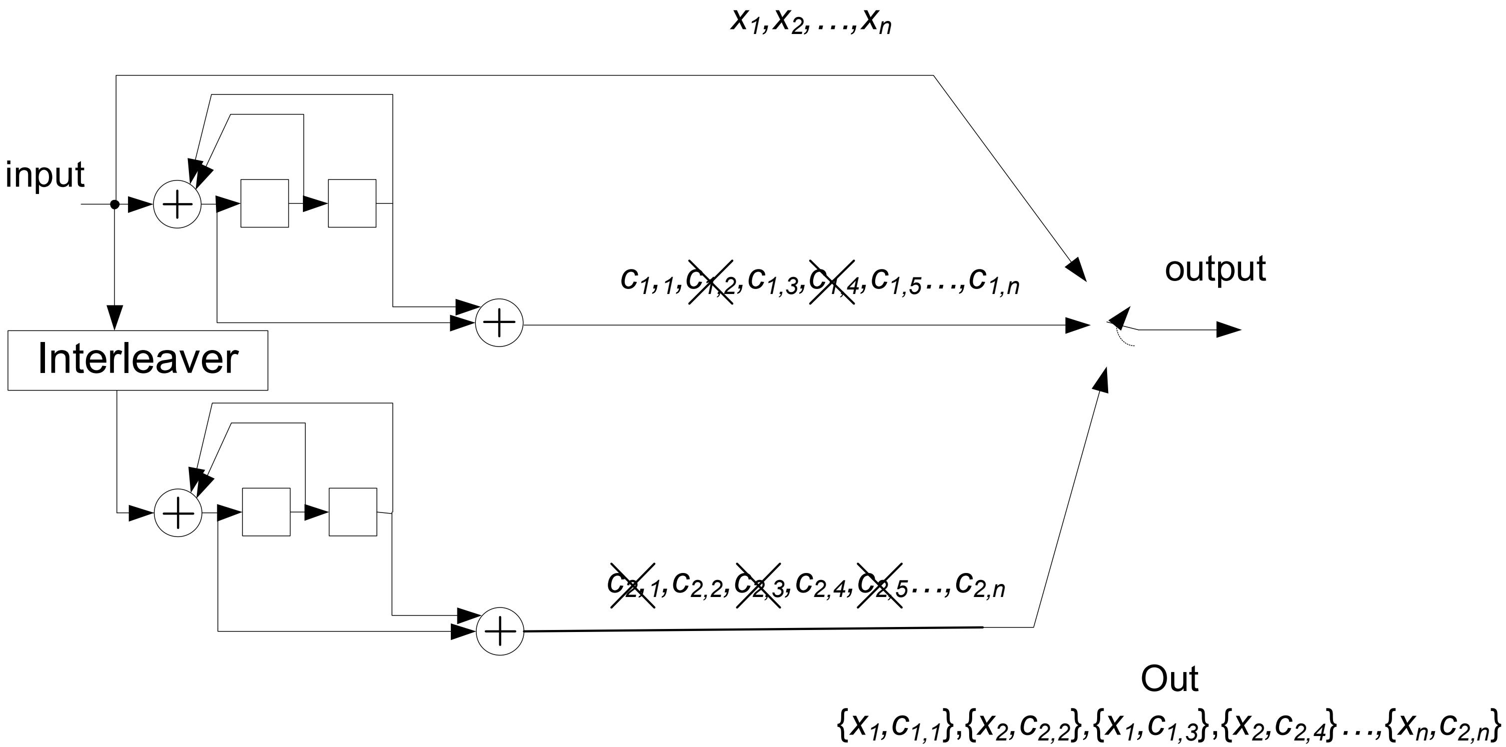


Figure 2: Punctured encoder

MAP trellis for the punctured code

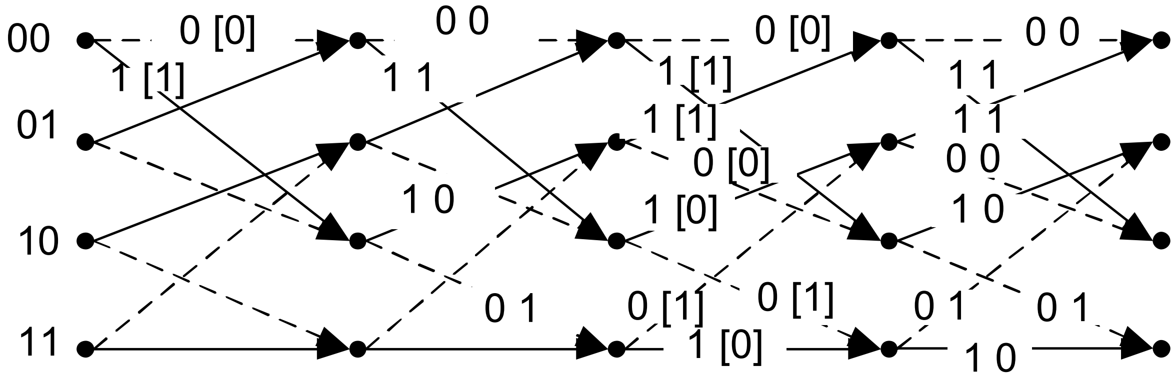


Figure 3: Punctured MAP for decoder 1

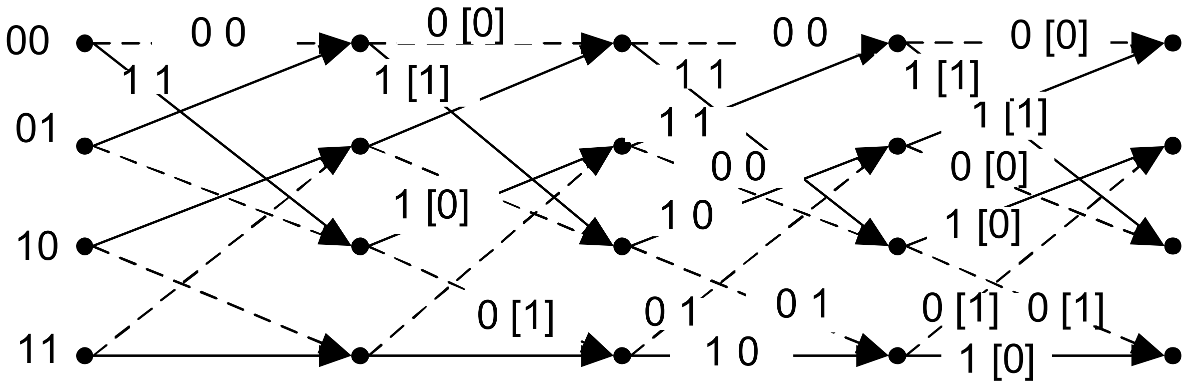


Figure 4: Punctured MAP for decoder 2

Trellis Code Modulation (TCM)

- Spectral efficiency: number of bits per second per 1 Hz of bandwidth $(bits/s/Hz)$.
- The data rate can be increased withouth increasing the bandwidth by transmiting more information per every symbol. (More bits per every channel use).
- The information content of the symbol is increased by increasing amount of possible symbol values. For example for transmitting two bits per symbol we have to have four possible symbol levels.
- If the energy per information bit is kept constant teh higher number of symbol values decrease the average per symbol - increase of error probability for symbol.
- Error probability can be decreased by adding more code bits - the code rate is increased.

- In general the coding maps information bit to higher number of code bits.
- If the coded bits are transmitted bit by bit in order to maintain the data rate we have to increase the channel usage rate.
- More code bits means that spectral efficiency is decreased since more bandwidth is required.
- The trellis coded modulation introduces additional parity bits and does not increase the bandwidth.
- The effective throughput in the channel is maintained by enlarging the number of constellation points.
  - By increasing the amount of constellation points we increase the signal set and more information can be transmitted by each singnal.

- The signalling rate is not increased since each symbol contains more information.
- Since power ber bit is kept constant after increasing the constellation size the distance between the possible constellation points, symbols, is should be decreased
- The positive coding gain is achieved if the increase of the error probability due to smaller distance between constellation points is outweighted by the coding gain of the error correction code.

- For eaxmaple: we have 4 information bits that are tramnsmitted in 4 subsequenct time intervals.
- The information bits are coded with 1/2 rate code. Thecodeword contains 8 coded bits.
- In order to maintain the same information rate we have to send for each infoamtion bit two coded bits. The coded bits can be transmitted either by
  - Using two times shorter pulses. (two times more bandwidth)
  - Using more constellation points per channel use. For example 4 QPSK instead of BPSK.
    In order to keep power bit constant this new constellaion should be scaled with the coding rate.

Coding versus noncoding

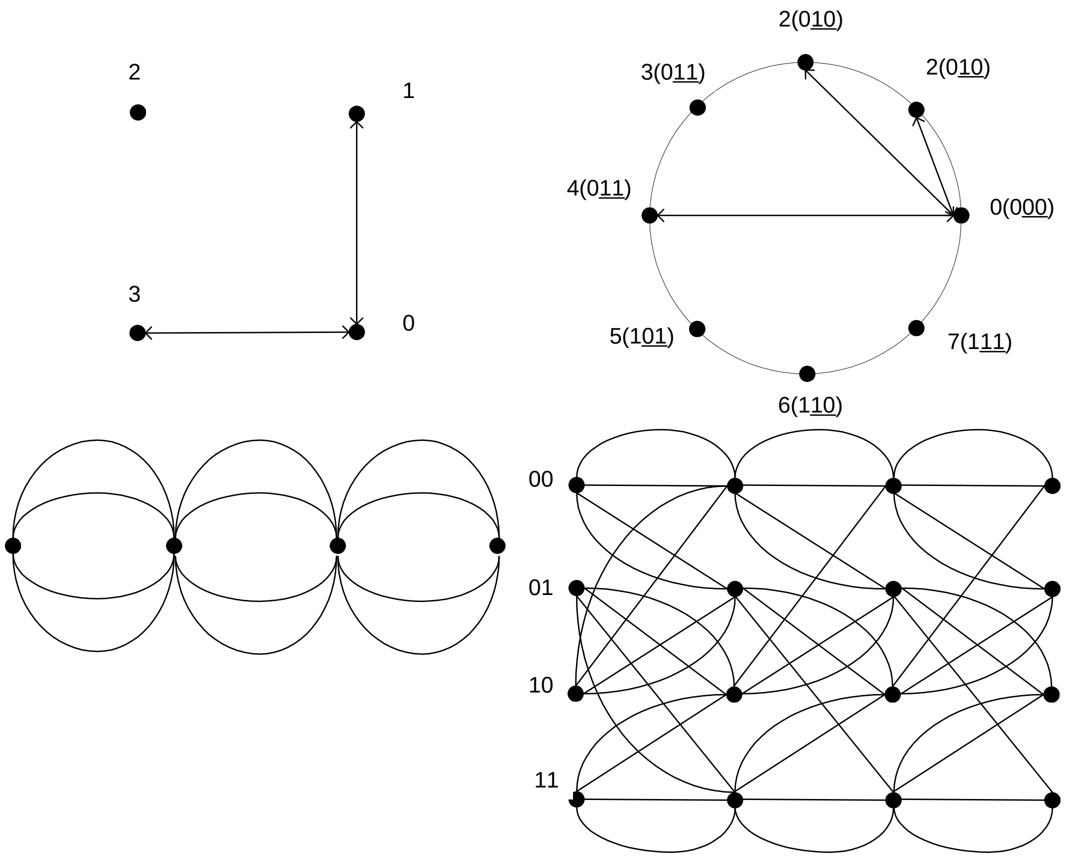


Figure 5: Coded and non coded transmission

4 PSK transmitter

- 4 PSK each of the four bits may be transmitted.
- The sequence that can be transmitted is not restricted.
- The optimum detector makes nearest phasor based decision for each individual received symbol.
- Each phasor is represented by two bit symbol.

8 PSK TCM endoder

- The 8PSK trellis has four states selected accordingly to the state of the shift register.
- At arrival of a new symbol the content of the shift register is changed.
- The encoding allows only certain trajectories trough the trellis.
- Illegitimate sequences can be rejected.

Example of TCM principle

- The extended signal set has smaller distance between the neighbouring constellation points than the initial signal set.
- Subset of the signal set has better distance compared to initial set.
- If we know at the decoder which subset is in the use we have better BER for the bit decision than in the initial code.
- In TCM the selection of the signal set is made accordingly to the state in the trellis of the convolutionally encoded bits.

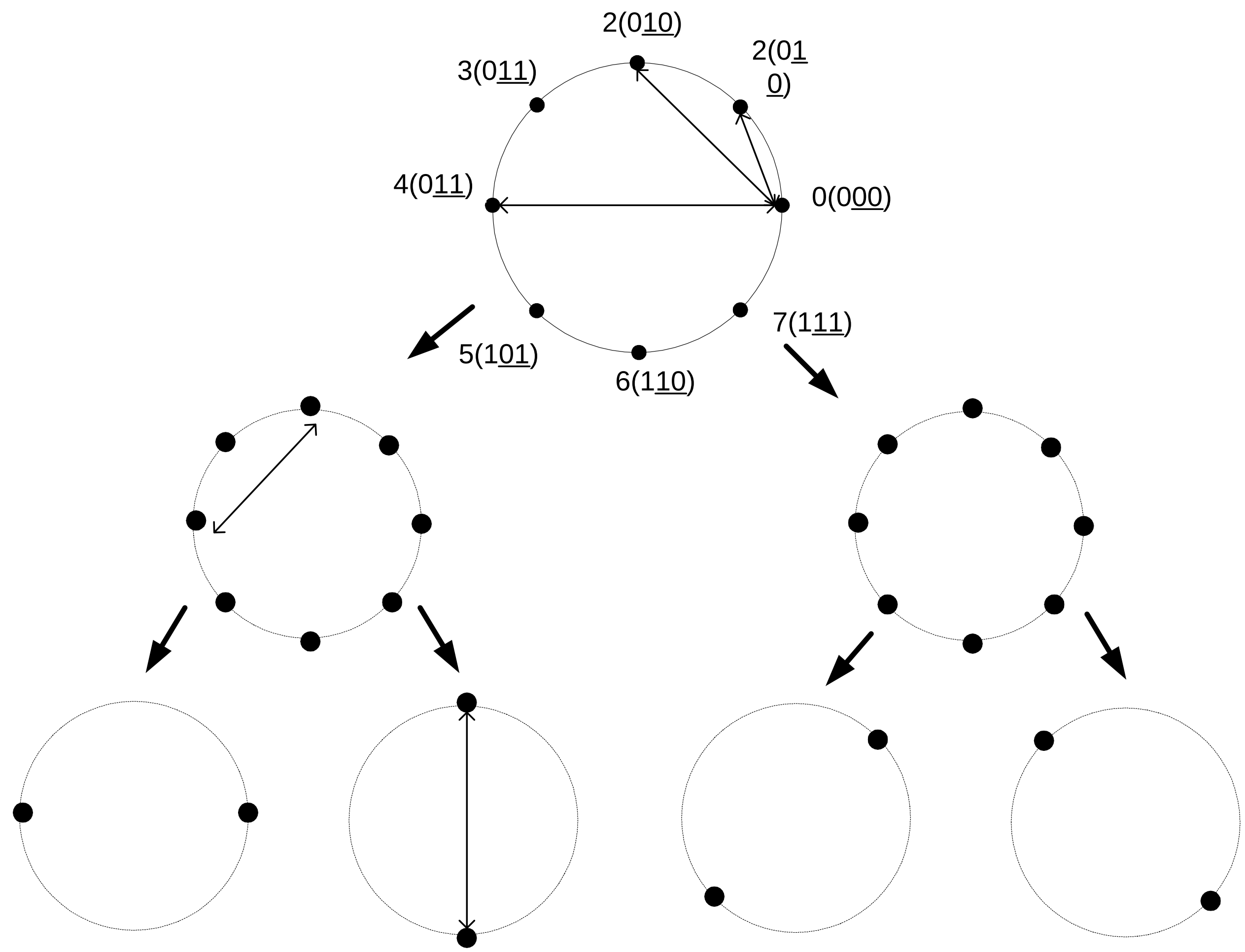


Figure 6: Example of the set partitioning

- The binary phasor identifiers are not Gray encoded.
- The bit assignment is made for achieving high Eucledian distance between the trajectories in the trellis.
- The Eucledian distance amongst constellation points is increased at every partition step.
- Parallel trellis transitions are assigned to phasors with maximum possible distance.
- All the signals are used with equal probability.
- The state transitions have distance of $d_1 = \sqrt{2}$ at least.

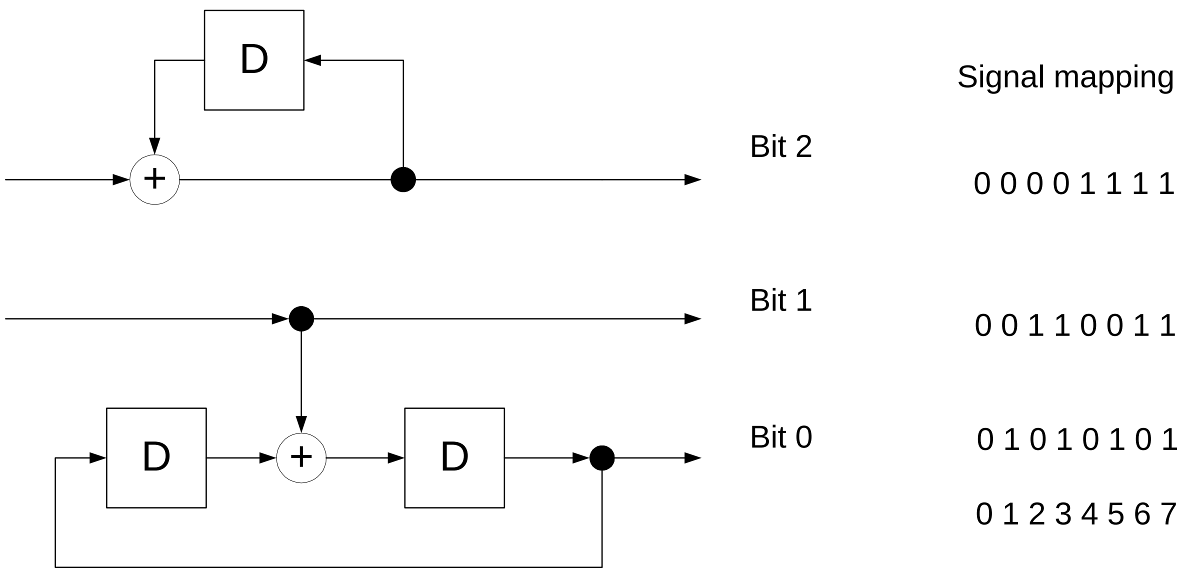


Figure 7: Example of a TCM encoder

- Last two bits are used for identifying the used set.
- The unprotected bit 2 is used for selecting the point in each partitioned set.

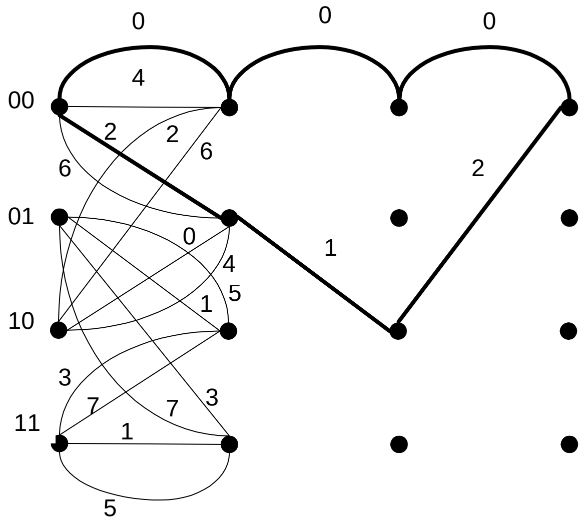


Figure 8: Example of a TCM encoder, symbol based trellis

- The error is divergence from the correct trellis path.
- The minimum distance is the minimum from :
  - the distance between the phasors labelling the parallel branches
  - distance between the trellis paths.
- Free distance in 4PSK

$$d_{free} = \sqrt{2}$$

- Free distance in TCM 8PSK

$$d_{free} = \min\left\{d_2; \sqrt{d_1^2 + d_0^2 + d_1^2}\right\} = 2$$

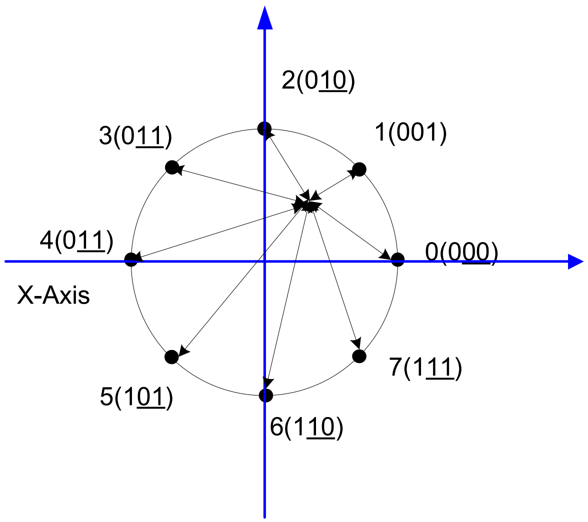
Probability calculation for the symbols

We have to evaluate the *aposteriori* probability for each possible constellation point given the observed value from the channel.

$$p(R|H)$$

H is defined by the possible constellation point.
R is the observed value

Example: 8 PSK



- possible 8 positions in the complex plain.
- Can be evaluated as multiplication of two independent gaussian probabilities or as a complex gaussian probability.
- For a symbol $s_1 = 1$

$m_{1,real} = 1$ $m_{1,imag} = 0$

$$p(r|s_1) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(r_{real}-m_{1,real})^2}{2\sigma}} \cdot \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(r_{imag}-m_{1,imag})^2}{2\sigma}}$$

$$= \frac{1}{2\pi\sigma_c^2} e^{-\frac{|r-m_1|^2}{2\sigma_c}}$$

If the received point is $r = 0.7 + 0.6 * j$

Const. Point c_i	$P(r\|c_i)$	$\ln(P(r\|c_i))$	$\frac{\ln(P(r\|c_i))}{\sum_j \ln(P(r\|c_j))}$
1.0000	0.13	-2.026	-2.34
0.7071 + 0.7071i	1.19	0.1768	-0.1408
0 + 1.0000i	0.048	-3.0307	-3.34
-0.7071 + 0.7071i	0.000057	-9.7697	-10.09
-1.0000	0.000...	-16.0925	-16.41
-0.7071 - 0.7071i	0.000...	-18.2953	-18.61
0 - 1.0000i	0.000...	-15.0878	-15.41
0.7071 - 0.7071i	0.000237	-8.3488	-8.67

Turbo Pinciple on Trellis codes

- The probability is calculated for the whole symbol s_1.
- The *aposteriori* probabilities can not be calculated for the bits separately.
- In general the bits probabilities are not independent.

$$p(r|s_1) = p(r|x_1, x_2, x_3) \neq p(r_1|x_1)p(r_1|x_2)p(r_1|x_3)$$

Possible Solutions

- Symbol based coding → Symbol Based MAP.
- Marginalisation → BICM.
- Use multiple interleavers and clever puncturing.

Symbol based turbo codes

- We employ multiple trellis encoders in parallel.
- The puncturing is made at the output of the encoders.
  - Puncturing is made on the symbols.
  - In case of two encoders every other symbol is selected from different encoder output.
- The symbol stream to different encoders is interleaved.

  In order not to puncuture systematic bits:
  - The symbol stream is split into two sets.
  - Interleaving is made inside of the sets.
  - During the puncturing: from one encoder is punctured away symbols in one set, from the other encoder symbols from the ohter set.

Turbo Trellis Code Modulator

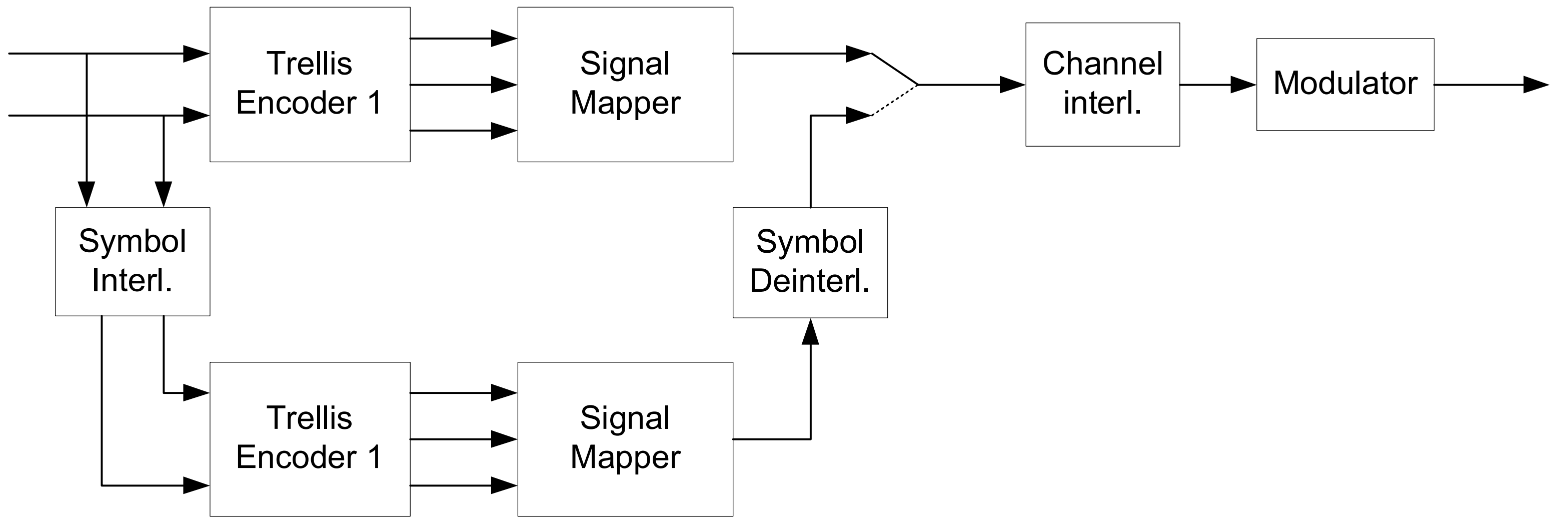


Figure 9: Example of a TTCM encoder

Symbol based Turbo Decoding

The symbol based MAP algorithm differ from its bit based counterpart:

- The probabilities are calculated for the symbols.
- In general the logarithms are evaluated for the probabilities not for the likelihood ratios.
- The extrinsic infomation is calculated for the symbols.

In TTCM encoder the full symbol is punctured away.

- In the trellis all the probabilities for all the possible values the punctured symbol can take are set to be equal.
- The MAP decoder evaluates the *aposteriori* probability also for the symbols that are punctured away and use these as the extrinsic information for the other decoder.
- Between the decoders is transmitted the extrisic information for the symbols.

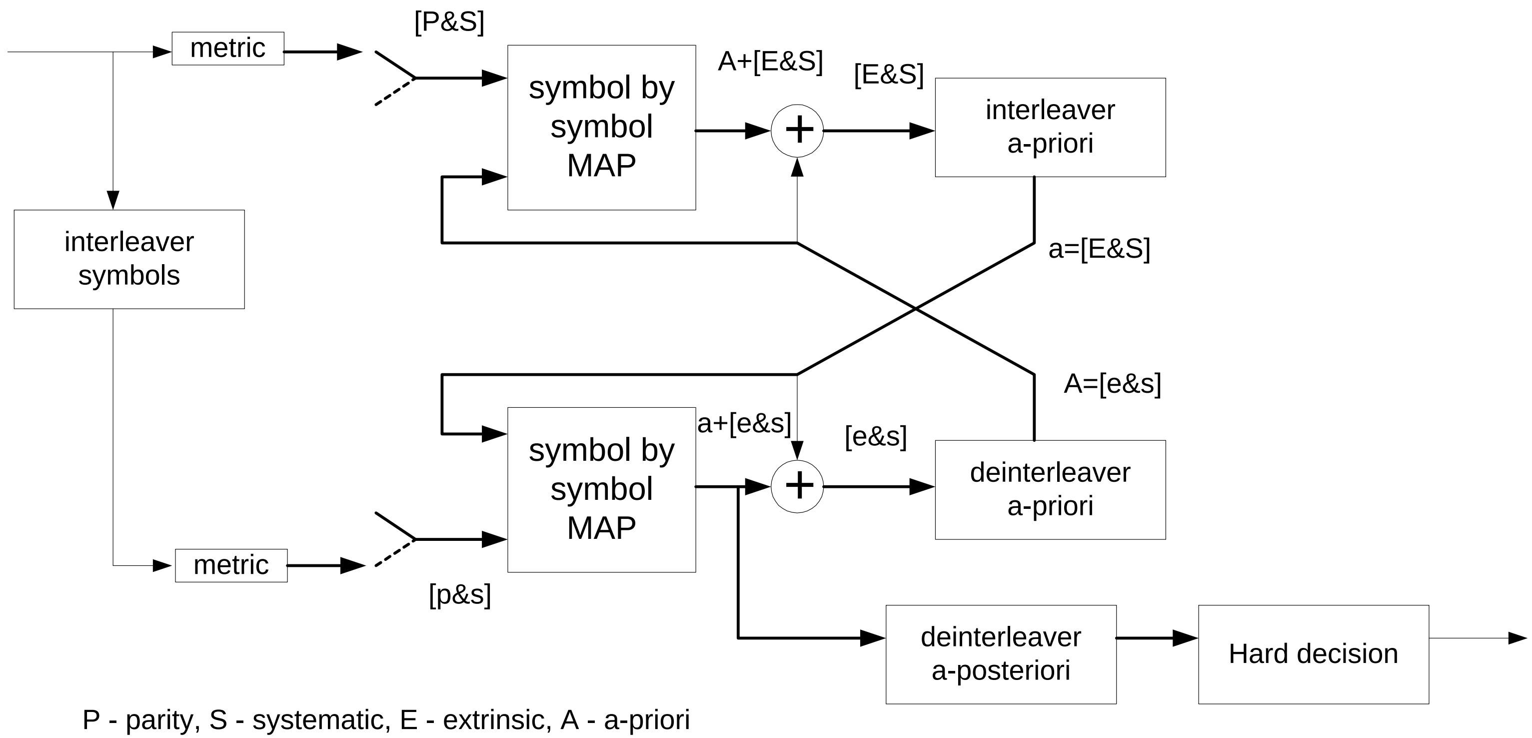
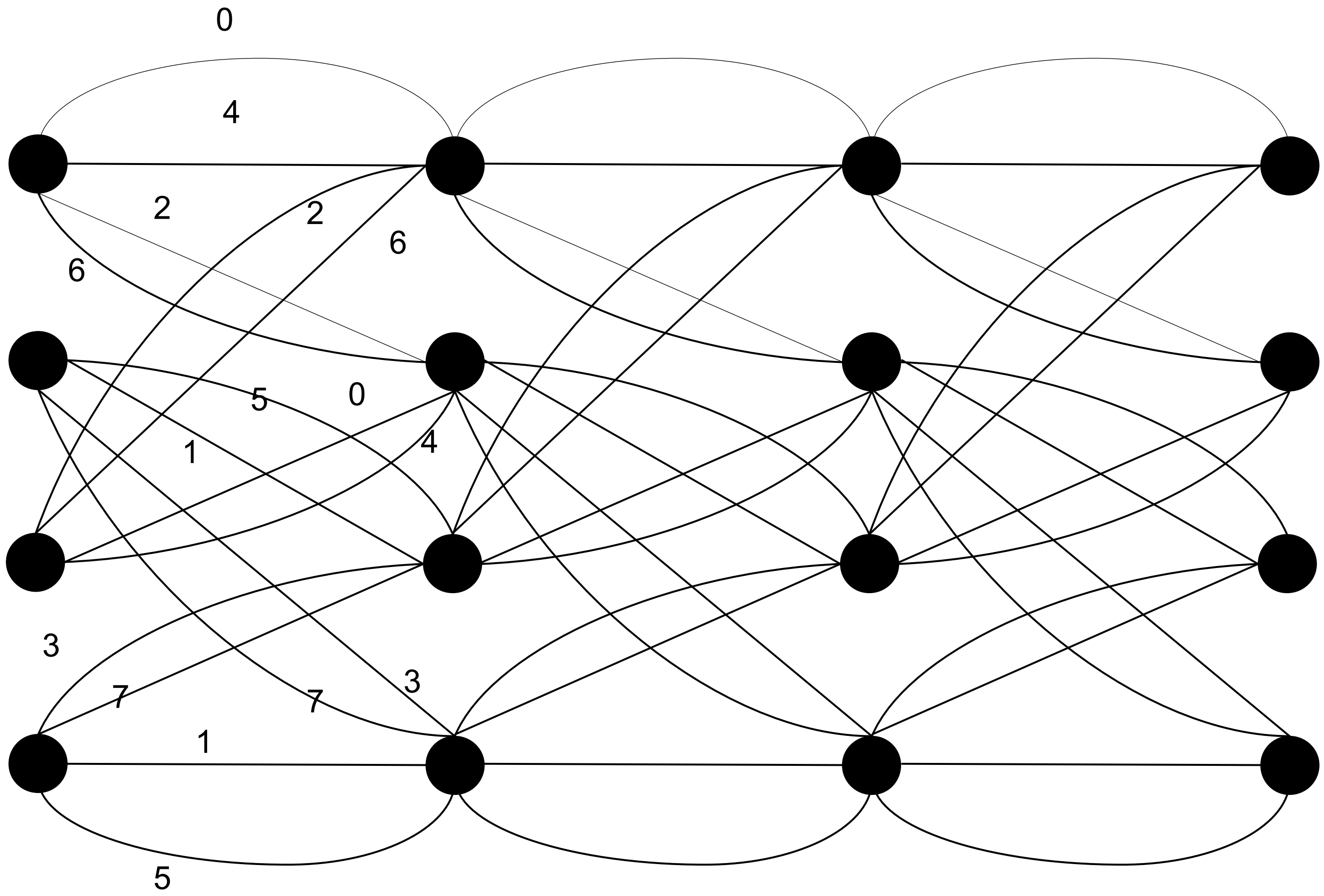


Figure 10: Example of a TCM decoder



$P(r_1|s_{\#,2})=P(r_1|b_{\#,11},b_{\#,21},b_{\#,31})$

$P(r_2|s_{\#,2})=P(r_2|b_{\#,12},b_{\#,22},b_{\#,32})$

$P(r_3|s_{\#,3})=P(r_2|b_{\#,13},b_{\#,23},b_{\#,33})$

Punctured

Nonbinary or symbol based MAP algorithm

- Calculation of the probability value for the symbol.
- Each information symbol u_k can have M different values and contain $m \log_2 (M)$ bits of information.
- The decoder computes aposteriori Probability (APP) $p^A_{k,m}$ for each of the possible 2^m symbols.
- $p^A_{k,m}$ is the probability that the received symbol at time k was $s_k = m$.
- The received symbol is decided to be the one with the highest $p^A_{k,m}$ probability.

The algorithm for calculating $p^A_{k,m} = P\left(s_k = m|\underline{y}\right)$

- Calculation of the $p^A_{k,m}$ is similar to calculation of a bit aposteriori probablity in the trellis of a binary code.
- We sum over all the paths trough the trellis where given symbol at time k was $s_k = m$.

$$p^A_{k,m} = \frac{1}{P\left(\underline{y}\right)} \sum_{\substack{\left(s',s\right)\Rightarrow \\ s_k=m}} A_{k-1}\left(st\right) \cdot M_k\left(st',st\right) \cdot B_k\left(s'\right) = C_k \cdot \overline{p}^A_{k,m}$$

- In decision process we do not need the absolute values but ratios and therefore can drop from multiplication with the common constants.

- The computation of $M_k(st', st)$ is different in the symbol based map compared to the binary MAP.

$$M_k(s', s) = P(\{y \wedge S_k = st\} | S_{k-1} = st')$$
$$= P(y_k | \{st', st\}) \cdot P(st | st')$$
$$= P(y_k | \{st', st\}) \cdot P(m)$$

- $s_k = m$ is the input symbol necessary to cause the transition from state st' to state st.
- $P(m)$ is the a-priori probability of m. (Typically all the symbols are equiprobable).
- The first term is probability that the symbol received is y_k and the symbol transmitted is x_k (x_k is phasor corresponding to s_k).
- $y_k = x_k + n_k$. n_k is complex AWGN.

$$P\left(y_k \middle| \{st', st\}\right) = \frac{1}{2\pi\sigma^2} e^{-\frac{|y_k - x_k|^2}{2\sigma^2}} = C_k^2 \eta_k\left(st', st\right)$$

$$A_k\left(st\right) = \sum_{all\ \ st'} A_{k-1}\left(st\right) M_k\left(st', st\right)$$

$$B_k\left(st'\right) = \sum_{all\ \ st} B_{k+1}\left(st\right) M_k\left(st', st\right)$$

- In the result of the algorithm we get *aposteriori* estimation for each symbol.
- Since in TTCM the codes have the common symbols this *aposteriori* information can be used in decoding of the other code as the extrinsic information of the symbols.

Complexity of the MAP symbol decoder

- N information symbols.
- Each information symbol contains M possible values
- Number of encoder states is S.
- The trellis code double the original signal constellation. There are $\overline{M} = 2M$ possible transmitted symbols.
- The forward and backward computation requires
  - $2 \cdot N \cdot M \cdot S$ multiplications
  - $N \cdot M \cdot S$ additions
- The terms in $p^{A}_{k,m}$ require three multipliations
- there are total $N \cdot M$ terms with S terms to be summed.

$7 \cdot N \cdot M \cdot S$	multiplications
$3 \cdot N \cdot M \cdot S$	summations
$3 \cdot N \cdot M$	exponentials

Performance of TTCM

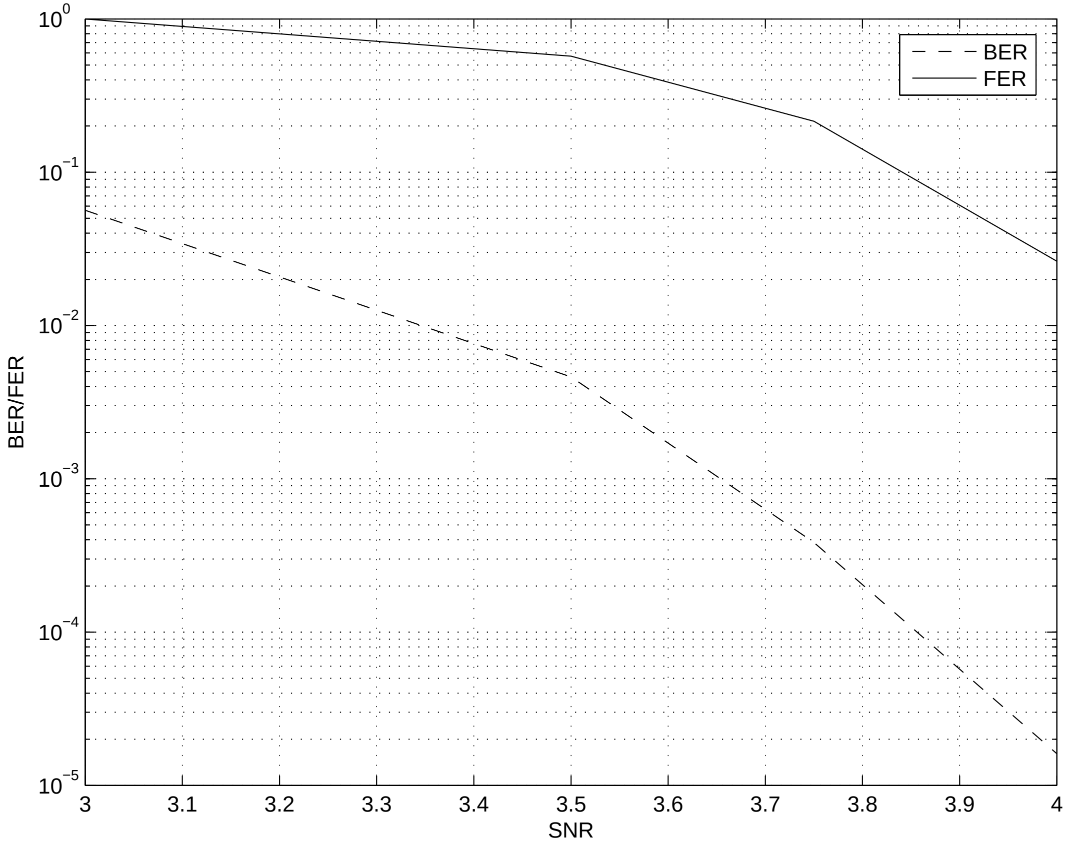


Figure 12: BER and FER for 8PSK encoded TTCM, datarate 2 bits per symbol

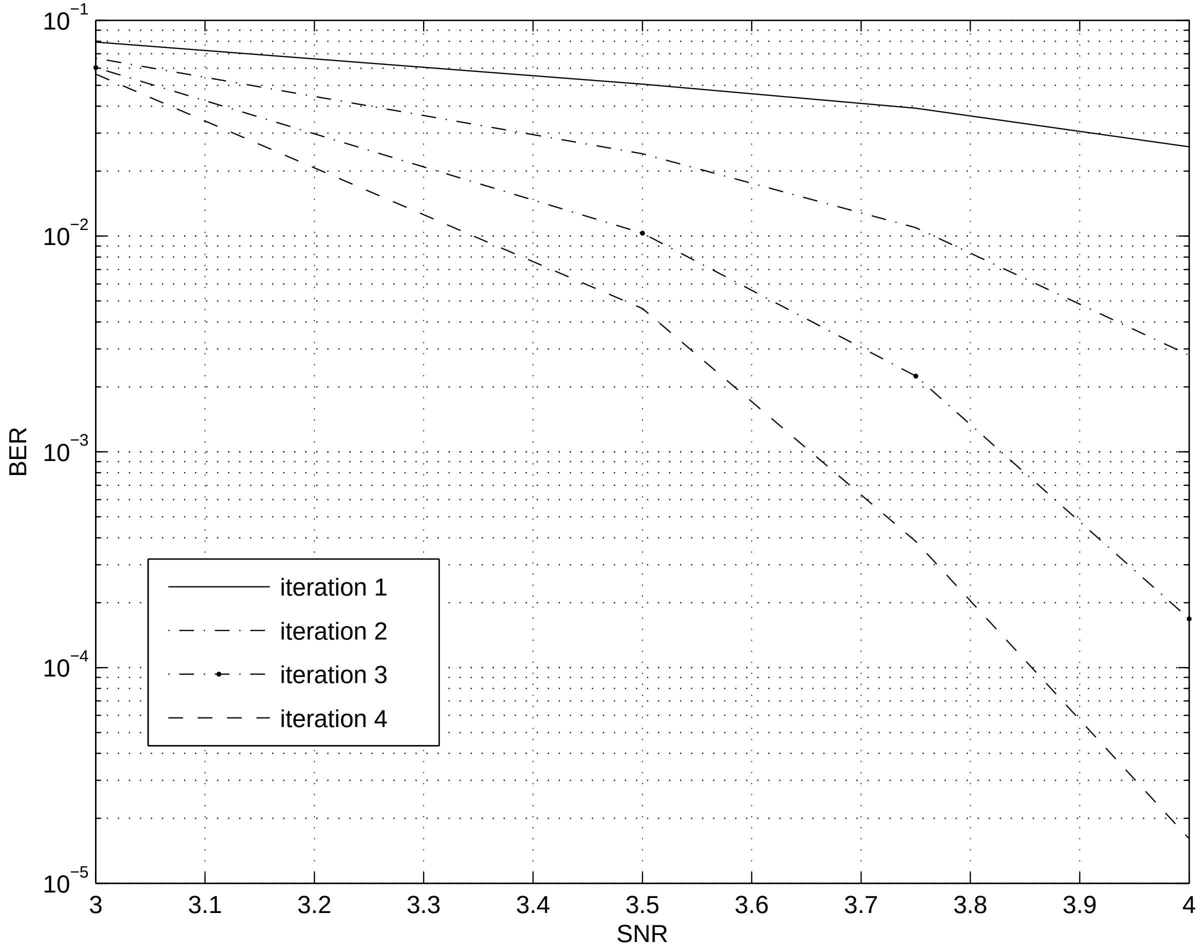


Figure 13: TTCM performance dependency on interations

Bit Interleaved Coded Modulation (BICM)

- Purpose of the BICM is to increase the diversity order of TCM schemes.
- Diversity order of TCM is the minimum number of different symbols along the shortest error event path between the correct and error event path.
- If there is no parallel branch the degree of diversity is increased when constraint length of the code is increased.

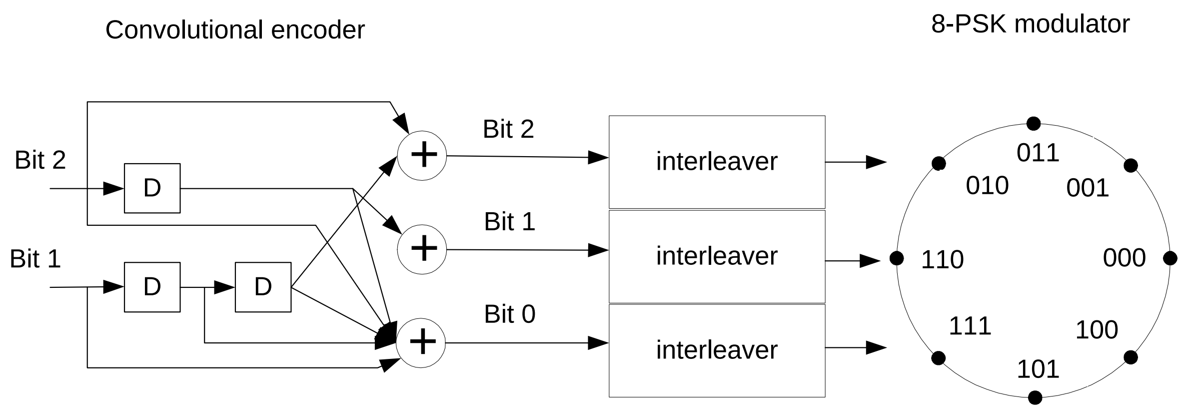


Figure 14: Example of a BICM encoder

- The symbols are generated by a non-systematic convolutional encoder.
- BICM uses bit interleavers for all the bits of a symbol.
- The number of bitinterleavers equals to the number of bits assigned to one non-binary codeword.
- Purpose of the bit interleaver:
  - Disperse the burst errors and maximize the diversity order of the system.
  - Uncorrelate the bits associated with the given transmitted symbol.
- The interleaved bits are collected into non-binary symbols.
- The symbols are Gray labeled.

Turbo BICM

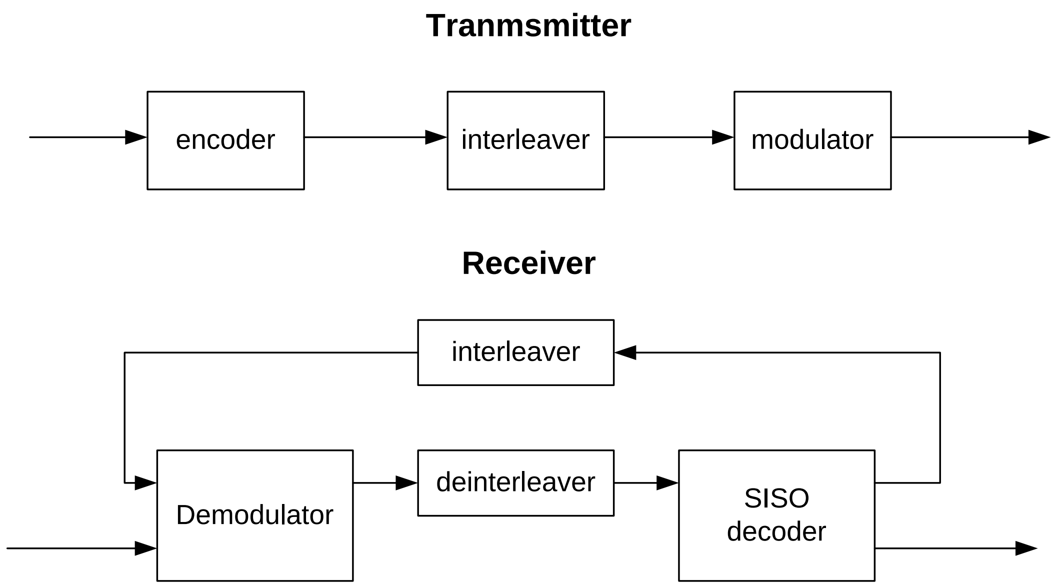


Figure 15: BICM combined with forward error correction

The iterative turbo BICM improves BICM performance in gaussian channel by increasing free eucledial distance of the code. Two new ideas:

calcualtion of the marginal probability for each bit

set partitioning

- The deocder treats each bit stream as independent.
- Form the symbols are calcualted the marginal probability for each bit.
- The decoder treats marginal probablities as they would be independent.

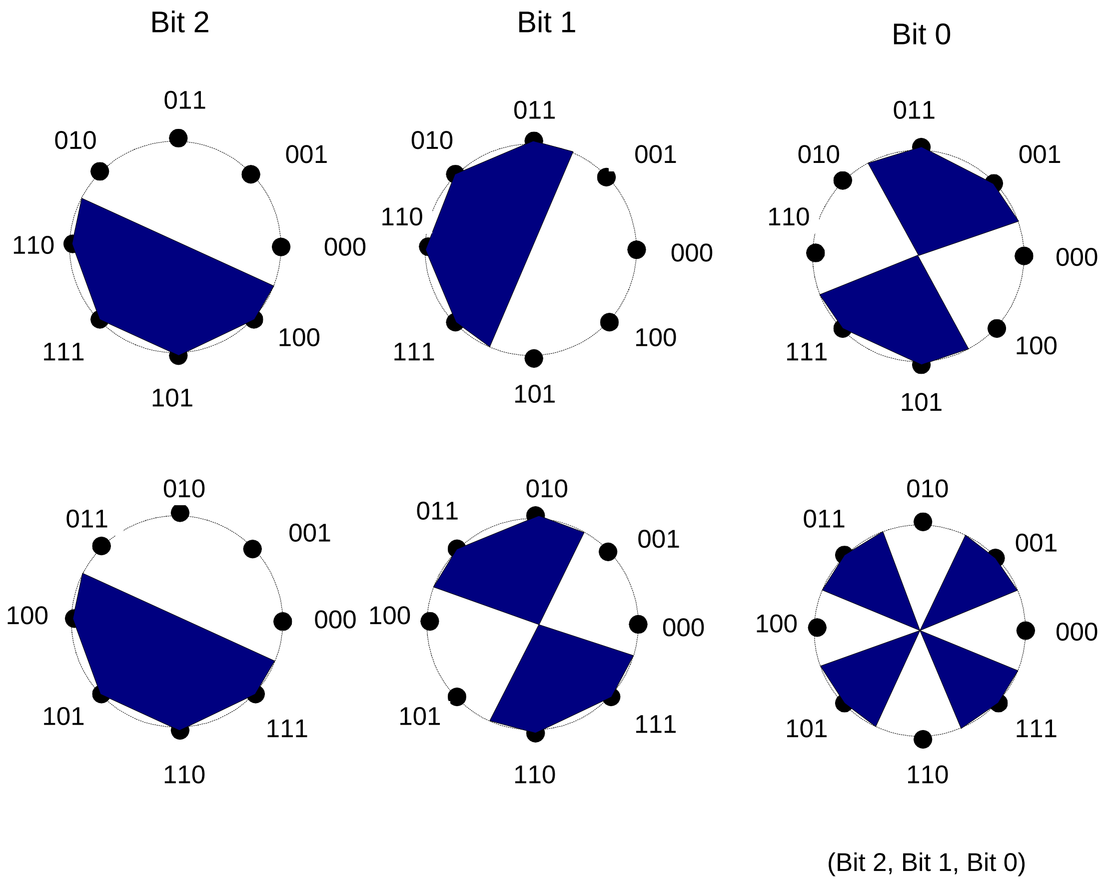


Figure 16: Example of a set partitioning compared to Gray labelling

BICM-ID converts a $2^{\overline{m}}$ ary signalling scheme into $\overline{m}$ independent parallel binary schemes.

- First iteration - the Gray labelling optimal.
  - Gray labelling has a lower number of nearest neighbours compared to SP - based labelling.
  - The higher the number of nearest neighbour the higher the chances for a bit to be decoded into wrong region.
- Second iteration
  - The soft information allows to confine the decision region into a pair of constellation points.
  - We want to maximise the minimum Eucledian distance between any two points in the possible phasor pairs for all the bits.

Turbo channel equaliser

MLSE Equaliser

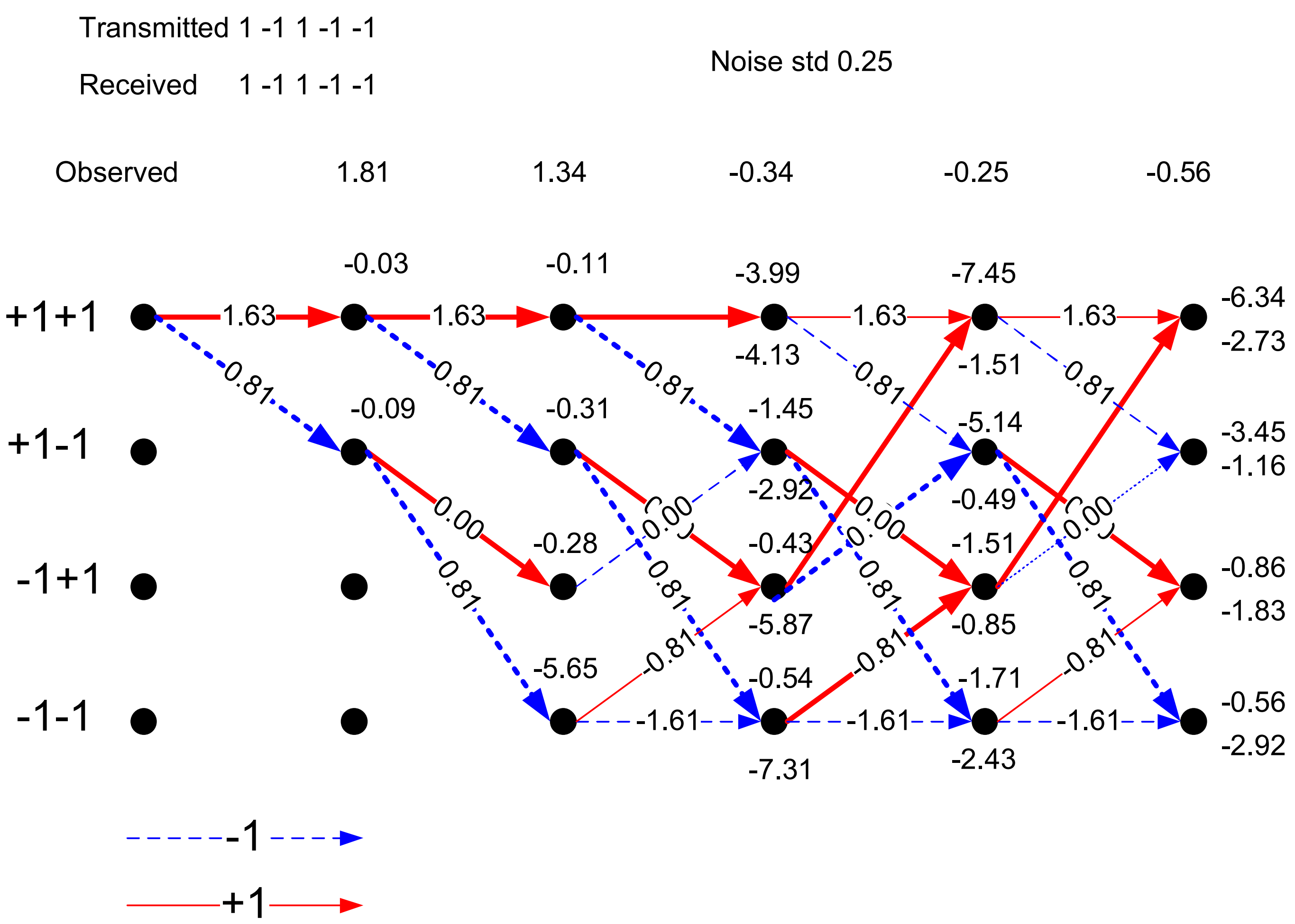
- For the channel with $L+1$ taps and the signal constellation 2^n there are 2^{nL} states.
- The state at epoch k $s_k = (u_{k-1}, u_{k-2}, \ldots, u_{k-L})$.
- ML receiver chooses the sequence with maximal likelihood function $P(y_k, \ldots, y_1 | u_k, \ldots, u_1)$ or log-likelihood function

$$\ln P(y_k, \ldots, y_1 | u_k, \ldots, u_1) =$$
$$= \ln P(y_k | u_k, \ldots, u_{k-L}) + \ln P(y_{k-1}, \ldots, y_1 | u_{k-1}, \ldots, u_1)$$

- For Gaussian noise the first term yields the branch metric $\Gamma_k = -(y_k - \hat{y}_k)^2$,
  where the channel estimate $\hat{y}_k = \sum_{i=0}^{L} g_i u_{k-i}$.
  For Viterbi algorithm the path metric is $\sum_k \Gamma_k$.
- The branch metric for the transition $s_i \to s_j$ at epoch k is

$$\Gamma(_i, s_j) = -\left(y_k - g_0 u_k(_i, s_j) - \sum_{m=1}^{L} g_m u_{k-m}(_i, s_j) \right)^2 .$$

Example



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

Adaptive MLSE Equaliser

With the LMS algorithm the tap coefficients are updated

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

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

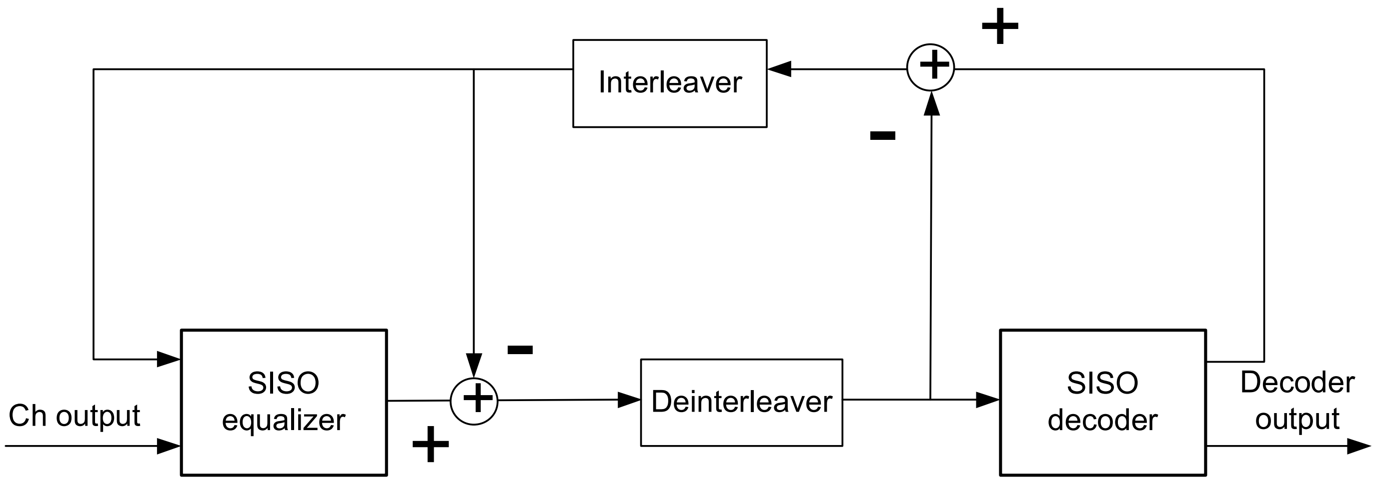
$$\epsilon_{k-Q} = y_{k-Q} - \sum_{i=0}^{L} \hat{g}_i(k)\hat{u}_{k-i-Q}$$

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

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

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

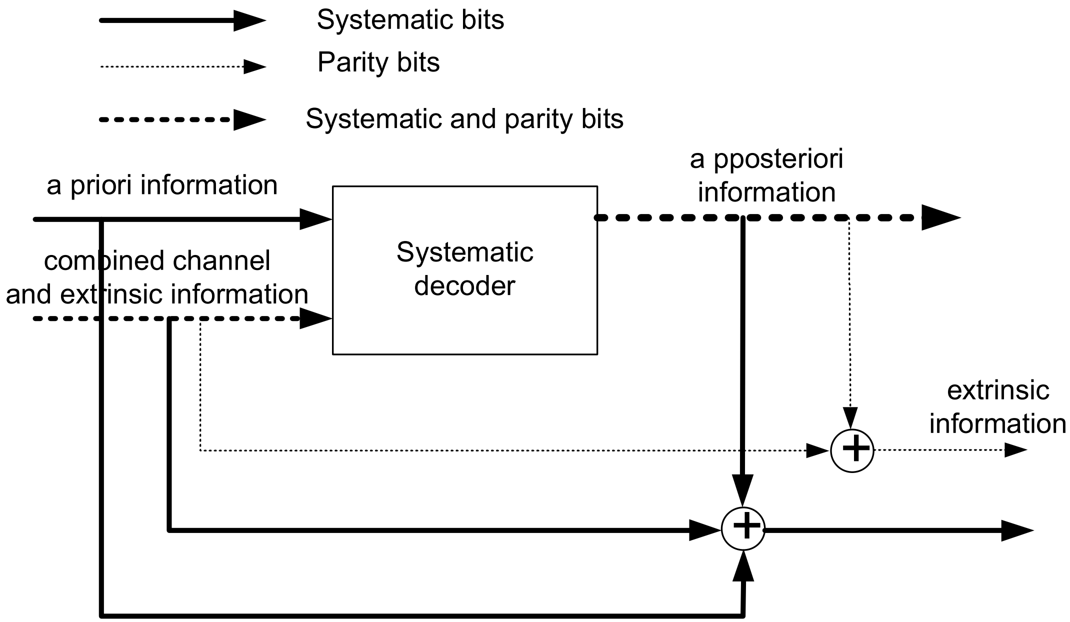
Turbo Equaliser



Structure of original Turbo Equaliser introduced by Douillard et al. (1995)

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

Information flow

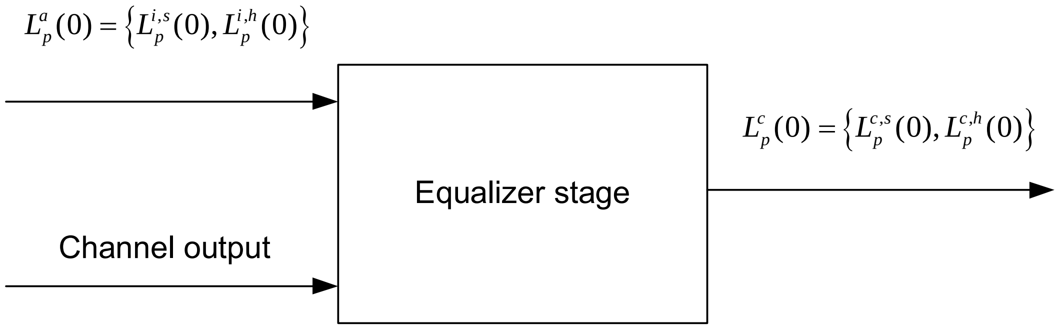


A-priori (intrinsic) - information, known before decoding (equalisation), from other sources than received sequence or code constraints

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

Equaliser



L^c: composite a-posteriori LLR

$L^{c;s}$: a-posteriori information of the source bit

$L^{c;h}$: a-posteriori information of the parity bit

L^a: composite a-priory LLR

$L^{a;s}$: a-priory information of the source bit

$L^{a;h}$: a-priory information of the parity bit

Equaliser information

for p-th iteration and N_d number of decoders

$$L_p^a(0) = \left\{ \sum_{j=1}^{N_d} L_{p-1}^e(j); L_{p-1}^t(1); \ldots ; L_{p-1}^t(N_d) \right\}.$$

$$\begin{aligned} L_p^c(0) &= L_p^i + L_p^a(0) \\ &= \left\{ L_p^{i;s} + \textstyle\sum_{j=1}^{N_d} L_{p-1}^e(j); [L_p^{i;h} + L_{p-1}^t(1); \ldots ; L_p^{i;h} + L_{p-1}^t(N_d)] \right\}. \end{aligned}$$

L^i: combined channel and extrinsic information, unseparable since channel response is a non-systematic code

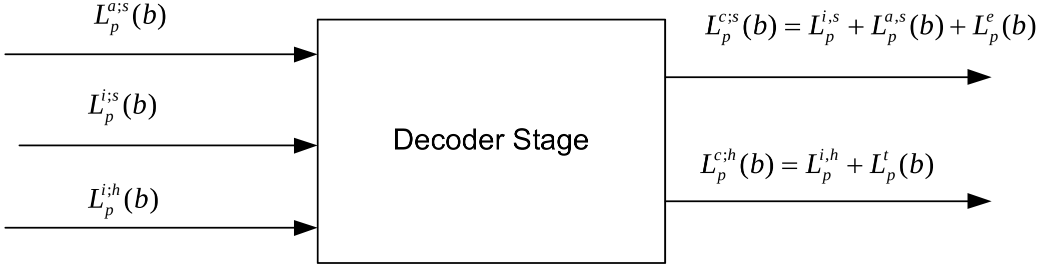
$L^{i;s}$: combined channel and extrinsic information for the source bits

$L^{i;h}$: combined channel and extrinsic information for the parity bits

L^e: extrinsic information of the source bit

L^t: extrinsic information of the parity bit

Decoder



$$L_p^{a;s}(b) = \sum_{j=1}^{b-1} L_p^e(j) + \sum_{j=b+1}^{N_d} L_{p-1}^e(j).$$

$$\begin{aligned} L_p^{c;s}(b) &= \sum_{j=1}^{b-1} L_p^e(j) + \sum_{j=b+1}^{N_d} L_{p-1}^e(j) + L_p^e(b) + L_p^{i;s} \\ &= \sum_{j=1}^{b} L_p^e(j) + \sum_{j=b+1}^{N_d} L_{p-1}^e(j) + L_p^{i;s}. \end{aligned}$$

MAP Conditional LLR for transmitted bit at epoch m

$$
\begin{aligned}
L(u_m|y) &\triangleq \ln\left(\frac{P(u_m=+1|y)}{P(u_m=-1|y)}\right)\\
&= \ln\left(\frac{P(u_m=+1\wedge y)}{P(u_m=-1\wedge y)}\right)\\
&= \ln\left(\frac{+1P(\grave{s}\wedge s\wedge y)}{-1P(\grave{s}\wedge s\wedge y)}\right)\\
&= \ln\left(\frac{+1P(y_{j<m}\wedge \grave{s})P(y_m\wedge s|\grave{s})P(y_{j>m}|s)}{-1P(y_{j<m}\wedge \grave{s})P(y_m\wedge s|\grave{s})P(y_{j>m}|s)}\right)\\
&= \ln\left(\frac{+1\alpha_{m-1}(\grave{s})\gamma_m(,s)\beta_m(s)}{-1\alpha_{m-1}(\grave{s})\gamma_m(,s)\beta_m(s)}\right).
\end{aligned}
$$

Log MAP based on

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

The second term can be stored as a look-up table.

Complexity reduced by transforming to logarithmic domain:

$\Gamma_m(, s) \triangleq \ln(\gamma_m(, s))$

$$\begin{aligned} A_m(s) &\triangleq \ln(\alpha_m(s)), \\ &= \ln\left(\sum_{\forall \grave{s}} \alpha_{m-1}(\grave{s})\gamma_m(, s)\right) \\ &= \ln\left(\sum_{\forall \grave{s}} \exp[A_{m-1}(\grave{s}) + \Gamma_m(, s)]\right). \end{aligned}$$

$$\begin{aligned} B_{m-1}(\grave{s}) &\triangleq \ln(\beta_{m-1}()), \\ &= \ln\left(\sum_{\forall s} \beta_m(s)\gamma_m(, s)\right) \\ &= \ln\left(\sum_{\forall s} \exp[B_m(s) + \Gamma_m(, s)]\right). \end{aligned}$$

Branch metric in equaliser

by Bayes rule

$$\begin{aligned}\gamma_m(\grave{s},s) &= P(y_m \wedge s|\grave{s}) \\ &= P(y_m|s\wedge)P(s\wedge) \\ &= P(y_m|s\wedge)P(u_m)\end{aligned}$$

for equaliser given Gaussian channel

$$P(y_m|s\wedge) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(y_m-\hat{y}_m)^2}{2\sigma^2}\right].$$

$$\begin{aligned}\Gamma_m(\grave{s},s) &= \ln\left\{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(y_m-\hat{y}_m)^2}{2\sigma^2}\right]P(u_m)\right\} \\ &= \ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right)-\frac{(y_m-\hat{y}_m)^2}{2\sigma^2}+\ln(P(u_m)).\end{aligned}$$

The first term is constant and independent of u_m and can be neglected

$$\Gamma_m(\grave{s},s) = -\frac{(y_m-\hat{y}_m)^2}{2\sigma^2}+\ln(P(u_m))$$

$$P(u_m) = \left(\frac{e^{-L(u_m)/2}}{1+e^{-L(u_m)}}\right) e^{u_m L(u_m)/2}$$
$$= C_{L(u_m)} e^{u_m L(u_m)/2}$$

$C_{L(u_m)}$ doesn't depend on the sign of u_m and can be omitted

$$\ln(P(u_m)) = \frac{u_m L(u_m)}{2}$$

$L(u_m)$ is taken from $L_p^a(0)$ for the epoch m

$$\Gamma_m(,s) = -\frac{1}{2\sigma^2}(y_m^2 - 2y_m\hat{y}_m + \hat{y}_m^2) + \ln(P(u_m))$$

For decoder $\hat{y}_m$ is unity, y_m is the same for each transition and can be neglected.

Both source and parity bits must be considered, denote coded bits $c_{l,d}$ and the source bit u_d

$$\Gamma_d(,s) = \frac{1}{\sigma^2}\sum_{l=1}^{n=2} y_{l,d}c_{l,d} + \ln(P(u_d))$$

Branch metric in decoder

For turbo decoder

$$\Gamma_d(,s) = \frac{L_c}{2}\sum_{l=1}^{n=2} y_{l,d}c_{l,d} + \frac{u_d L(u_d)}{2}$$

where L_c is channel reliability for transmitted energy per bit E_b and fading amplitude a

$$L_c = 4a\frac{E_b}{2\sigma^2}$$

For decoder in turbo equaliser channel reliability is included in L_p^i

$$\frac{y_m}{\sigma^2} = \frac{L_p^{i;m}}{2}$$

$$\Gamma_d(,s) = \frac{1}{2}\sum_{l=1}^{n=2} L_p^{i;l}c_{l,d} + \frac{u_d L(u_d)}{2},$$

where $L(u_d)$ is taken from $L_p^{a;s}$

LogLR

for equaliser

$$L(u_m|y) = \ln\left(\frac{+1 \exp(A_{m-1}(\grave{s}) + \Gamma_m(\grave{s}, s) + B_m(s))}{-1 \exp(A_{m-1}(\grave{s}) + \Gamma_m(\grave{s}, s) + B_m(s))}\right).$$

for decoder both source and parity bits must be considered

$$L(c_{l,d}|L_p^i) = \ln\left(\frac{l{+}1 \exp(A_{d-1}(\grave{s}) + \Gamma_d(, s) + B_d(s))}{l{-}1 \exp(A_{d-1}(\grave{s}) + \Gamma_d(\grave{s}, s) + B_d(s))}\right).$$

Let $x(\grave{s}_i, s_j)$ represent

$$P(\grave{s}_i \wedge s_j \wedge L_p^i) = \exp(A_{d-1}(\grave{s}_i) + \Gamma_d(\grave{s}_i, s_j) + B_d(s_j))$$