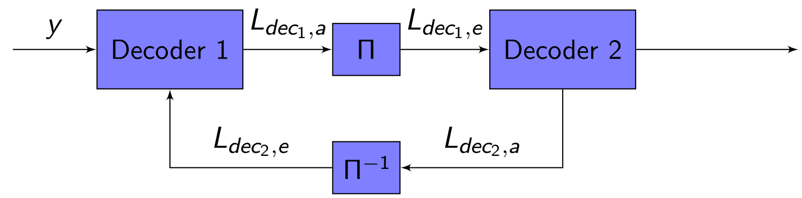
Turbo processing

Kalle Ruttik

Communications Laboratory
Helsinki University of Technology

April 18, 2007

Turbo processing



Intuitive reason for good performance

Each decoders use likelihood generated by other decoder.
At each iteration we have more and more information about the bit.
More information means: better estimates - less noise, less errors.

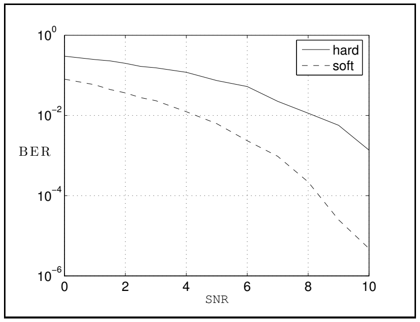
Turbo blocks

- Multiple decoders that can exhange information
- Interleaver to shuffle the bits in the codewords to different decoders.
  - Neighboring bits in code for one decoder are far away in the codeword for the other decoder
  - Errornous paths occure since the neighboring bits are erroneous
  - In other codeword those erroneous bits are not neighbors - no erroneous path in other codeword, we can correct the errors.
- Decoding results of one decoder are feed to other decoder
  - $L_{dec_\#,e}$ interleaved (or deinterleaved) loglikelihood at the input to other decoder
  - $L_{dec_\#,a}$ loglikelihood at the output of a decoder

Providing information back

- How to provide information to other decoder?
  - Need soft output - reliability information
  - MAP algorithm
- How to combine information at the other encoder? How the combination impacts the decoder performance?
  - Optimum combination: conditioning the received bit sequence with information from other decoder
  - Side information
- How to estimate the system performance?
  - Information exchange chart - EXIT chart

Cobining the bits



How to combine the bits in the best possible way:

- Two noisy samples y_1, y_2 describe the same transmitted bit in noise (both are either 1 or 0)
- Combination of hard decided bits gives bad performance
- Combination by using also reliability information allows to improve the performance

Combining hard bits

- Make hard decision on each bit
- Combine the hard decisions accordingly to the rules in table below

bit 1	bit 2	decision
0	0	0
0	1	undecided
1	0	undecided
1	1	1

- Undecided describes decoder inability to decode

Combining soft bits

- The estimation of the bit for one observation

$$P(x|y) = \frac{p(y|x)p(x)}{p(y)}$$
$$L(x|y) = \log\frac{p(y|x=+1)}{p(y|x=-1)} + \log\frac{p(x=1)}{p(x=0)}$$

- In a Gaussian channel we can simplify it to be

$$L(x|y) = L_c x + L(x) = \frac{2a}{\sigma_N^2}x + L(x)$$

where

$L(x)$ is the *a-priory* information about x
$L_c = \frac{2a}{\sigma_N^2} = \frac{4aE_s}{N_0}$ is the channel state information (CSI)

Combining soft bits ...

- At the input of the combiner we assume to have two independent noisy observations

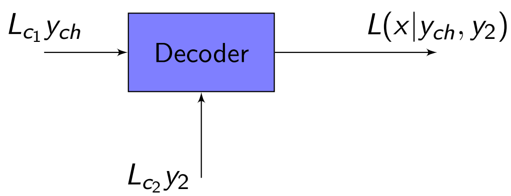
$$p(x|y_1, y_2) = p(x|y_1)\cdot(x|y_2)$$

- Loglikelihood of aposteriori probability

$$L(x|y_1, y_2) = \log\frac{p(y_1|x_1=1)p(y_2|x_2=1)}{p(y_1|x_1=0)p(y_2|x_2=0)} + L(x)$$
$$= L_{c_1}x + L_{c_2}x + L(x)$$

- both of these observations are derived from the same input bit value x
  - They both have the same x value: both are either 0 or 1
- the combined likelihood give a new likelihood

Combining soft bits ...



- Question: How much the knowledge about the bit is improved when the likelihoods are combined?
  - Whether it is improved?
  - How to describe the improvement? - reduced BER

Amount of the information in the soft bit

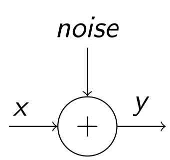
- What is changing when we combine two soft bits?
- How to measure the change? Are we improving or are we making the system worse?
- We have a feedback system: is the system going to be stable or not?
- The change can be measured as:
  - Change of the variance of the additive noise
  - Change of the BER
  - Change of the mutual infomation $\rightarrow$ EXIT chart

Bit with reliability information

- The observed bit is described as the exact bit value observed in the noise

$$y = x + n$$

  - x the bit value (either ± 1)
  - n noise with the two side power spectral density $\frac{N_0}{2}$



Expressions for noisy bit

- In binary modulation can be expressed in multiple ways
  - Probability

$$p(y|x=1) = 1 - p(y|x=-1)$$

  - Soft bit

$$\lambda(x) = +1 \cdot p\left(y|x=1\right) - (-1) \cdot p\left(y|x=0\right) = \tanh\left(\frac{L_c \cdot x}{2}\right)$$

  - Loglikelihood ratio (llr)

$$L_c \cdot x = \frac{p(y|x=1)}{p(y|x=0)}$$

- All of them contain sufficient information for making optimal decision

Bit in a Gaussian channel

Assuming AWGN with $n \in N(0, \sigma^2)$

$$\begin{aligned} y &= x + n & p(y|x=X) &= \frac{1}{\sqrt{2\pi\sigma_N^2}} e^{-\frac{(y-X)^2}{2\sigma_N^2}} \\ L &= \log \frac{p(y|x=1)}{p(y|x=0)} & L &= \frac{2}{\sigma_N^2} \cdot (x+n) \end{aligned}$$

$$\begin{aligned} L(x|y) &= \log \frac{p(y|x=1)}{p(y|x=-1)} \\ &= \log \left(\frac{\exp\left(-\frac{E_b}{2\sigma_N^2}(y-1)^2\right)}{\exp\left(-\frac{E_b}{2\sigma_N^2}(y-(-1))^2\right)} \right) + \log \frac{p(x=+1)}{p(x=-1)} \\ &= \left(\left(-\frac{E_b}{2\sigma_N^2}(y-1)^2\right) - \left(-\frac{E_b}{2\sigma_N^2}(y+1)^2\right) \right) + \ln \frac{p(x=+1)}{p(x=-1)} \\ &= -\frac{E_s}{2\sigma_N^2} \cdot 4 \cdot y + \ln \frac{p(x=+1)}{p(x=-1)} \\ &= -L_c y + L(x) \end{aligned}$$

Amount of information in a noisy bit

Mutual information

$$I(y,x) = H(x) - H(x|y) = H(y) - H(y|x)$$

In a binary input and equally probable symbols

$$H(x) = p(x=1)\log_2(p(x=1)) + p(x=-1)\log_2(p(x=-1)) = 1$$

$$I(y,x) = 1 - \int_{-\infty}^{\infty} p(x,y) \cdot \log(p(x,y))dy$$

Mutual information between the transmitted value x and llr value $L = L_c y$

$$I = \frac{1}{2} \sum_{x=-1,1} \int_{-\infty}^{\infty} p(L|x=X) \times \log \frac{2 \cdot p(L|x=X)}{p(L|x=1) + p(L|x=-1)}$$

Mutual information for binary bit in noise

In Gaussian noise the intergral can be expressed as

$$I(\sigma^2) = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(L-1)^2}{2\sigma^2}} \cdot \log\left(1 + e^{-L}\right) dL$$

- Amount of information depends only from from the noise variance.
- One parameter is sufficient to describe the amount of mutual information in llr value

Calculation of the mutual information

The mutual information can be calculated from a sequence of noisy binary values y_k as an average.
For stationary signal average over sequence converges to the same as the integral over distribution

$$I(x,y) = 1 - \frac{1}{N} \sum_{k=1}^{N} \log(1 + e^{-L_c \cdot y_k})dx$$

hint: A simple way to evaluate the extrinsic information is

$$I(x,y) = \frac{1}{N} \sum_k \left(\begin{array}{l} \frac{1}{1+e^{|L_c \cdot y_k|}} \cdot \log\left(\frac{2}{1+e^{|L_c \cdot y_k|}}\right) \\ + \frac{1}{1+e^{-|L_c \cdot y_k|}} \cdot \log\left(\frac{2}{1+e^{-|L_c \cdot y_k|}}\right) \end{array} \right)$$

This equation does not require knowledge of the exact bit values x

Exchanging information

- Idea of turbo processing is to improve iteratively the knowledge about the received bits
- For that in one decoder is combined channel observation and the output of the other decoder
- The decoding result is used by other decoder as an additional input
- The decoder improves BER since by combining information and calculating the MAP estimate of the bit the variance of the noise is reduced (we hope at least)
- To quarantee the convergence of the decoding process we have to calculate how the information is changed at output of each decoder in each iteration
- The decoding result is improved if from the decoder comes out more information than went into it in extrinsic information

Incorporation of side information

- Information coming from the decoder and provided to the equalizer is called side information or extrinsic information
- Extrinsic iformation is the outside information (not measured directly from the channel)
- For taking advantage of the extrinsic iformation we have to modify our receiver
- The modification was proposed 1992 by Berrou et.al. in the form of turbo decoder
- The transition between the states are defined by the symbol
- We know additional information about the symbol and combine it into transition

Practical calculation of mutual information

Unfortunately there is no known closed from equation for calculating mutual information at an ouput of decoder since we do not know how to calculate (by a closed from equation) llr distribution at the output of a decoder.
Usually for the analusis is used semianalytical methods.

1. For that we simulate the decoder and calculate the llr at its output
2. Calculate amount of information in the llr values
3. Repeate the simulation by providing the decoder with different amount of extrinsic information
    - The extrinsic information with given amount of information is generated by adding to the correct bit value a Gaussian random variable. Variance of the random variable is defined by the reguired mutual infomation

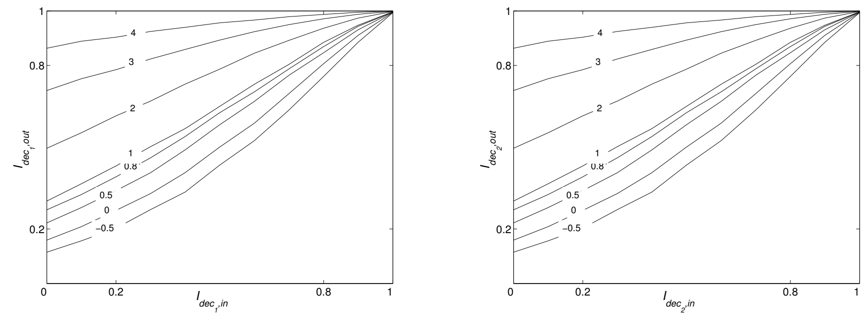
Algorithm for generating extrinsic information with required amount of mutual information

- We fix the mutual information
- Calculate how big variance correspond to this amount of mutual information
- Generate noise samples from Gaussian distribution with mean zero and given variance
- Add the noise values to the exact bit values and provide it to the decoder

Using extrinc information change for convergence analysis

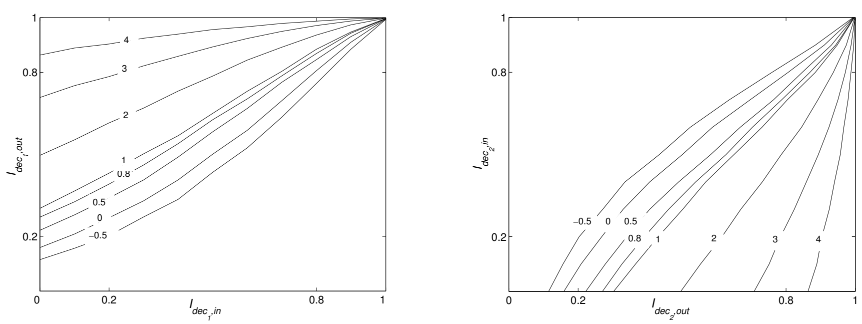
- The amount of the information in the output bit on a graph
  - On x axis is the amount of etrinsic information provided to the decoder
  - On y axis is amount of information at the output of the decoder
- The decoder is improved if at each iteration at the output of the decoder is more information than went into it
- The convergence can be checked by combining the EXIT charts of the decoders

Figure of EXIT



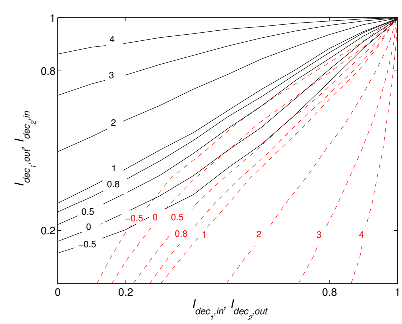
The information transfer fucntion for all the decoders can be generated by simulations.
A curve on the chart is parameterized by *SNR* in the channel

Figure of EXIT



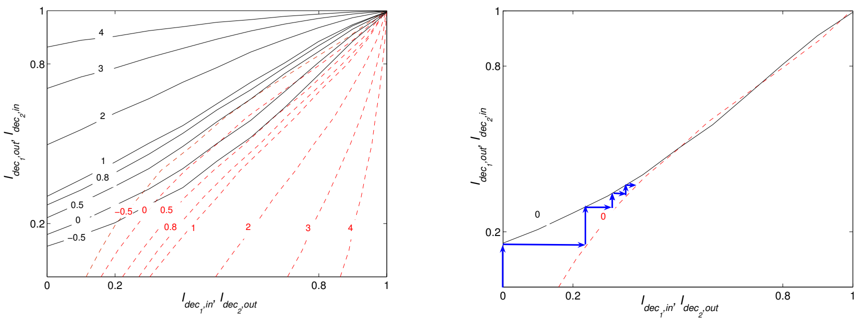
The transfer function of the second decoder can be converted x and y axes are changed

Figure of EXIT



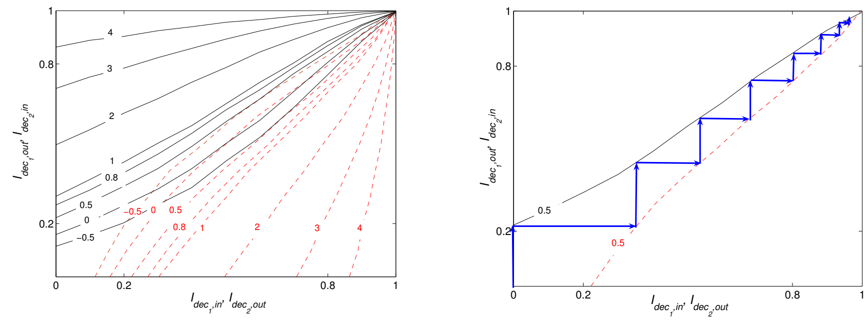
The resulting transfer fucntins can be drawn on common plot
Frow this plot can be seen at what input *SNR* the decoding process converges

Figure of EXIT



For given decoders at input $SNR = 0$ decoding does not converge

Figure of EXIT

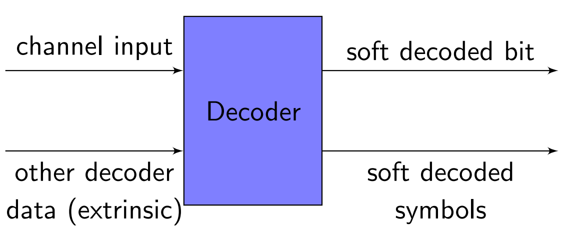


For given decoders at input $SNR = 0.5$ decoding does converges

SISO Decoder

In order to be able to apply turbo processing the decoder of one code should

- give out soft values
- be able to take in information from other decoder and combine it with values from the channel



Turbo Decoding

- Turbo codes are constructed by applying two or more component codes to different interleaved versions of the same information sequence.
- The idea of turbo decoding is to pass information from output of one decoder to the input of the other decoder and to iterate this process several times to produce better decision.
- Decoding algorithm should exchange soft decisions rather than hard decisions.
- The optimal soft output should be the a posteriori probability (APP), the probability of the received bits or sequence conditioned on the received signal.

Turbo Principle

Iterative exchange of soft information between different blocks in a receiver nowadays is successfully applied (beyond channel coding) for a wide range of communications problems:

- Combined equalization/estimation and error correction decoding
- Iteratively improved synchronization
- Combined multiuser detection and error correction decoding
- (Spatial) diversity combining for coded systems in the presence of multiple-access interference (MAI) or inter-symbol interference (ISI)

$\Rightarrow$ Iterative Receiver Concept

Turbo Equalization

ISI channel is the inner code in the serial concatenation.
ISI channel may be seen as a rate 1 code defined over the field of real numbers

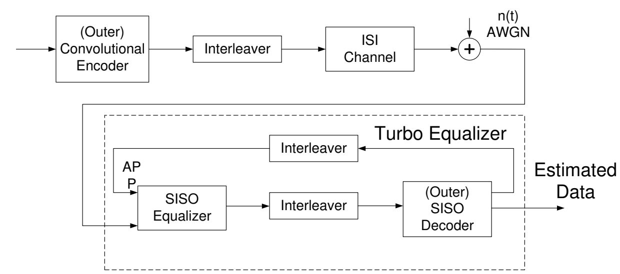


Figure: Turbo equalisation.

Turbo Multiuser Detection (1)

Multiple-access interference (MAI) channel is the inner code of a serial concatenation
MAI channel may be seen as a time-varying ISI channel.
MAI channel is a rate 1 code with time-varying coefficients over the field of real numbers.
The input to the MAI channel consists of the encoded and interleaved sequences of all users in the system.

Turbo Multiuser Detection (2)

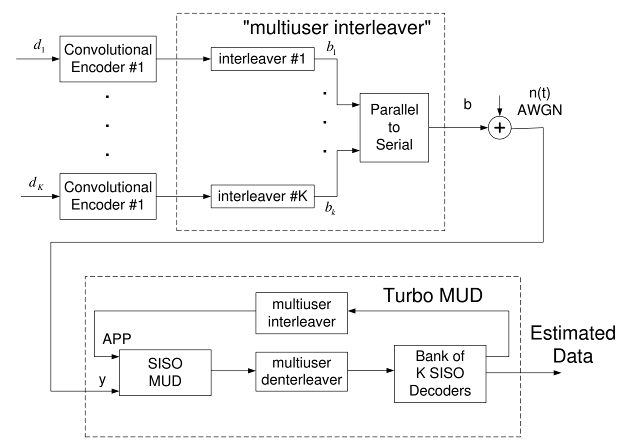


Figure: Turbo multi user detection.