A maximum-likelihood decoding algorithm for parallel concatenated convolutional code

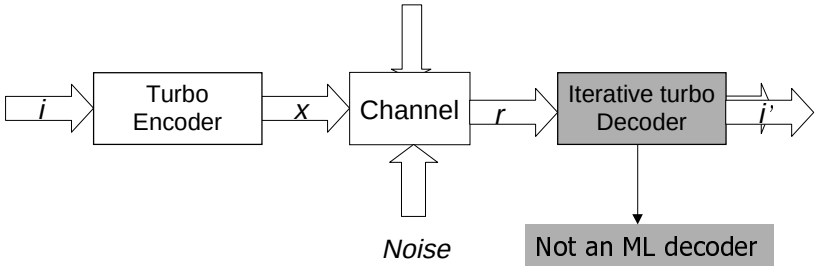
Presenter: Hsin-Chi Hsu (徐信琪)
Directed by Prof. Po-Ning Chen
Dept. of Communication Engineering
National Chiao-Tung University

Outline

- Introduction and background
- Tree-based MLSDA
- The structure of turbo encoder
- MLSDA for turbo codes
- Simulation Result
- Conclusions

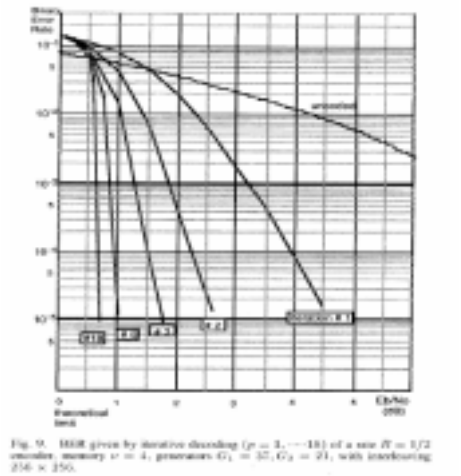
Introduction and background

- The original turbo decoder uses the **iterative decoding algorithm** [Berrou ,Glavieux, 1993] .
- It is not an ML decoding algorithm.



Introduction and background

- Near optimum error correcting coding and decoding (iteration=18). (Berrou, Glavieux)



Introduction and background

- The **error floor** observed in simulations of Turbo codes is a manifestation of the **free-distance asymptote**. (Perez, Seghers, and Costello in 1996)

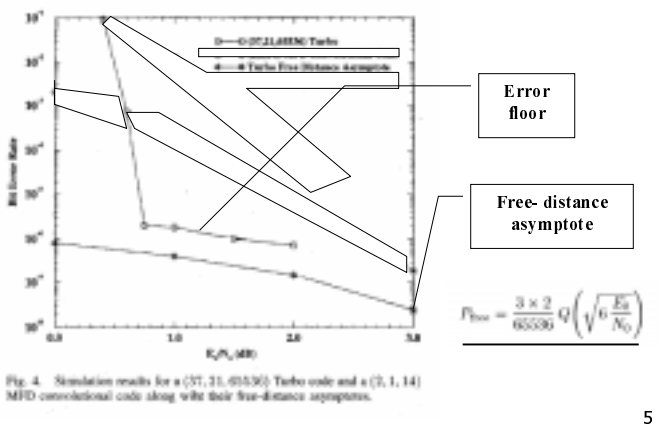
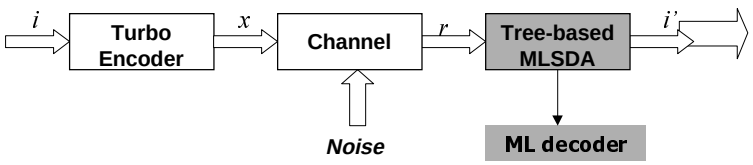


Fig. 4. Simulation results for a (37, 21, 65536) Turbo code and a (2, 1, 14) MFD convolutional code along with their free-distance asymptotes.

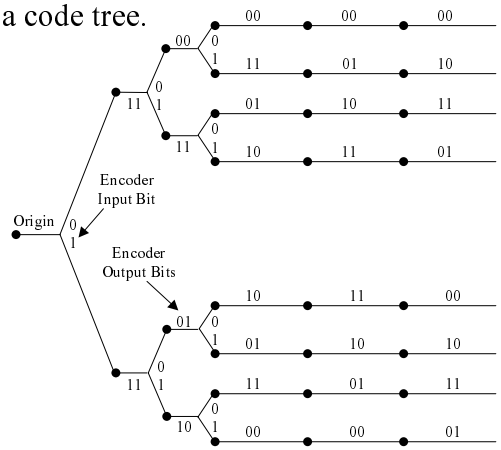
Introduction and background

- Tree-based *Maximum-Likelihood* Soft-decision sequential Decoding Algorithm (MLSDA) is an **ML decoding algorithm** for convolutional codes (Han and Chen 1998).
- In this thesis,
  - We apply the algorithm(MLSDA) for turbo encoder(PCCC).
  - We try to resolve the error floor phenomenon based on the new ML decoding algorithm.



Tree-based MLSDA

- The stack decoding algorithm with Fano metric, operating over a code tree.



Tree-based MLSDA

- Tree-based *Maximum-likelihood* Soft-decision Sequential Decoding Algorithm (Han and Chen 1998)
  - The conventional sequential decoding algorithm with a new metric

$$M\left(r_i^{(j)}\middle|x_i^{(j)}\right) = \left(y_i^{(j)} \oplus x_i^{(j)}\right)\left|\log_2 \frac{P\left(r_i^{(j)}\middle|x_i^{(j)}=0\right)}{P\left(r_i^{(j)}\middle|x_i^{(j)}=1\right)}\right|$$

where $x_i^{(j)}$ *and* $r_i^{(j)}$ are respectively the i[th] bit of the j[th] transmitted block and i[th] received bit at j[th] received block, and

$$y_i^{(j)} = \begin{cases} 1, & \text{if } \log_2\left[\dfrac{P\left(r_i^{(j)}\middle|x_i^{(j)}=0\right)}{P\left(r_i^{(j)}\middle|x_i^{(j)}=1\right)}\right] < 0; \\ 0, & \text{otherwise.} \end{cases}$$

Tree-based MLSDA

- Tree-based *Maximum-likelihood* Soft-decision Sequential Decoding Algorithm on **AWGN** channels (Han and Chen 1998)
  - The conventional sequential decoding algorithm with a new metric

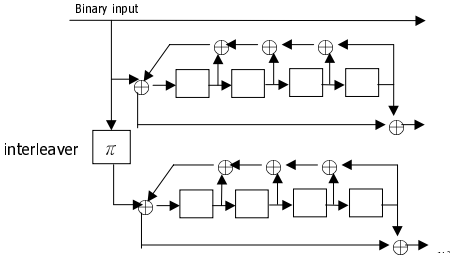
$$M\left(r_i^{(j)}\middle|x_i^{(j)}\right)=\left(y_i^{(j)}\oplus x_i^{(j)}\right)\left|r_i^{(j)}\right|$$

where $x_i^{(j)}$ *and* $r_i^{(j)}$ are respectively the i[th] bit of the j[th] transmitted block and i[th] received bit at j[th] received block, and

$$y_i^{(j)}=\begin{cases}1, & if\ r_i^{(j)}<0;\\ 0, & otherwise.\end{cases}$$

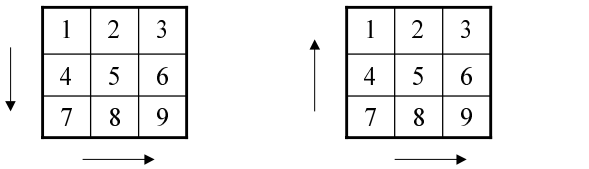
Turbo Encoder Structure

- PCCC with G_1=37,G_2=21 as its component code.
  - Interleaver is a reordering device.
  - Two identical component code.
  - Code rate(R) = 1/3.



Turbo Encoder Structure

- Block interleaver(BI) and Reverse block interleaver(RBI) with N=9 (Herzberg, 1996)



- Data write in rows , read out in columns.

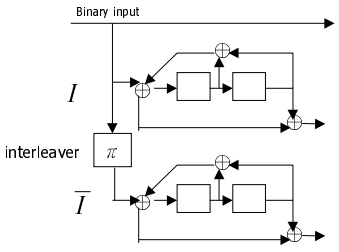
Generate code tree

- Combine the information bits before and after interleaving.

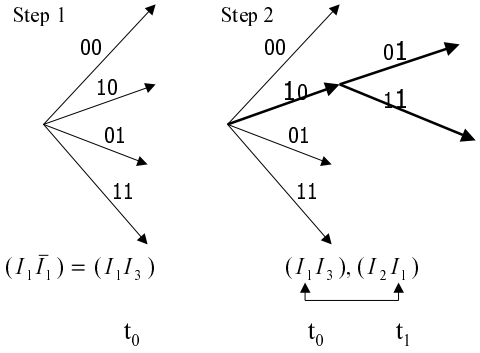
$$I = I_1 I_2 I_3 I_4 I_5 \Rightarrow (I_1 I_3),(I_2 I_1),(I_3 I_4),(I_4 I_5),(I_5 I_2)$$
$$\bar{I} = I_3 I_1 I_4 I_5 I_2 \qquad t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4$$

- π function (interleaver)

$$\bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4 \bar{I}_5 = I_3 I_1 I_4 I_5 I_2$$



Generate code tree



How many valid successors?

◆1

Both new information bits are deterministic.

◆2

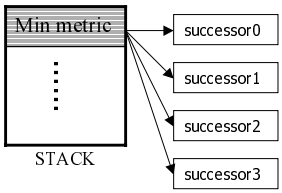
Only one of the two new info. Bits is deterministic.

◆4

None of the two new info. Bits is deterministic.

MLSDA for turbo codes

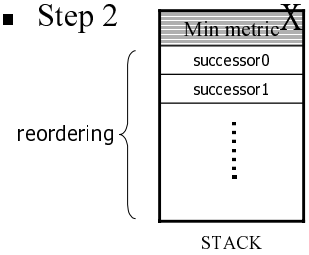
- Step1



◆ find the successors of the top path in the STACK

◆ There could be 1 or 2 or 4 successors.

- Step 2



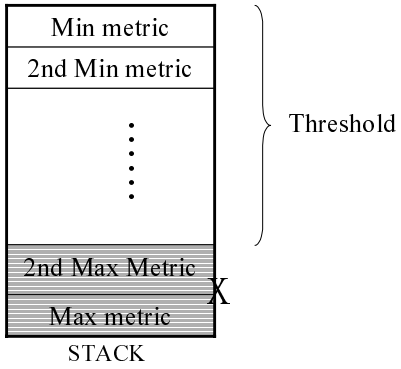
◆Delete the top path.

◆Insert the successors into the STACK.

◆Reordering the STACK according to the metric.

MLSDA for turbo codes

- Step 3
  If the length of STACK exceeds the Threshold, delete the path with Max metric in the stack.



Upper bound

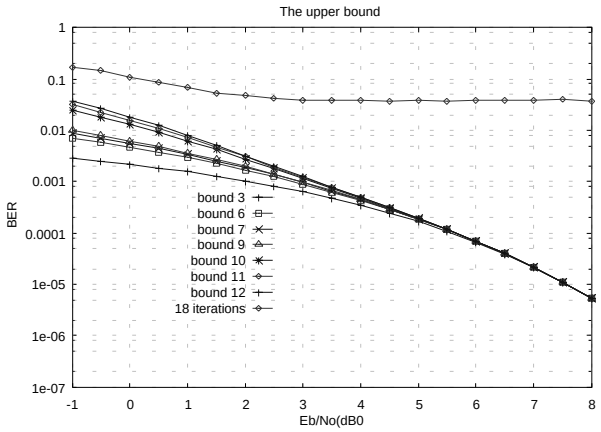
- The BER with ML decoding over AWGN channel can be upper-bounded by

$$P_b \le \sum_{i=1}^{2^L} \frac{w_i}{L} Q\left(\sqrt{d_i \frac{2RE_b}{N_0}}\right) \qquad P_b \le \sum_{d=d_{free}}^{3L} \frac{W_d}{L} Q\left(\sqrt{d \frac{2RE_b}{N_0}}\right)$$

L : informatio n length.
R : code rate.
w_i : weight of informatio n word.
d_i : weight of codeword.

Simulation Result

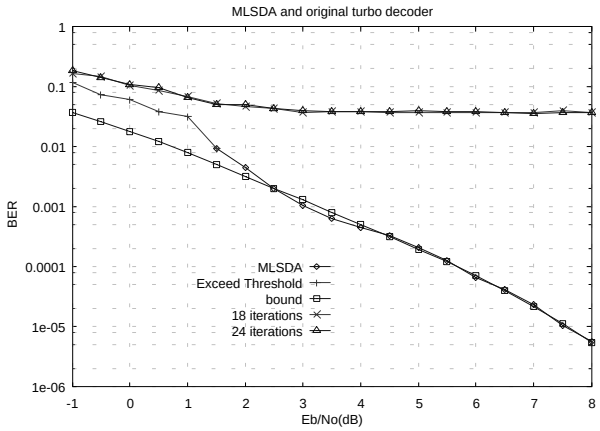
- L=36, G_1=37,G_2=21,R=1/3,BI.



D	W_m
3	1
6	4
7	2
9	3
10	50
11	31
12	34

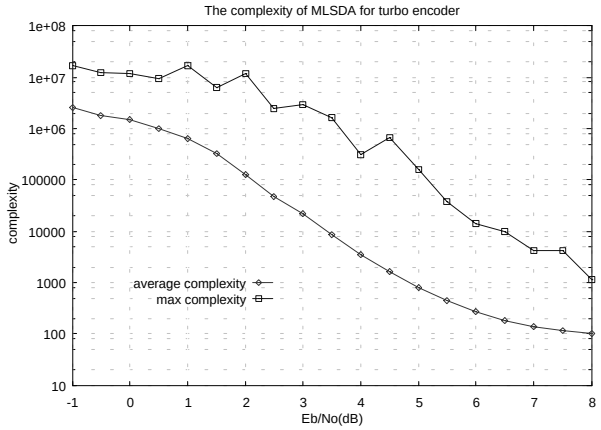
Simulation Result

- L=36, G_1=37,G_2=21, R=1/3,BI.



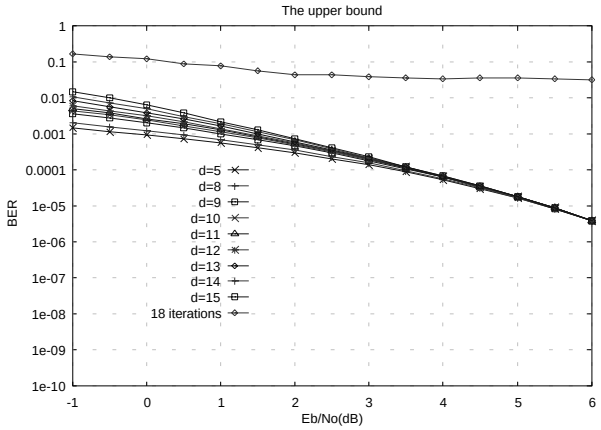
Simulation Result

- L=36, G_1=37,G_2=21, R=1/3,BI.



Simulation Result

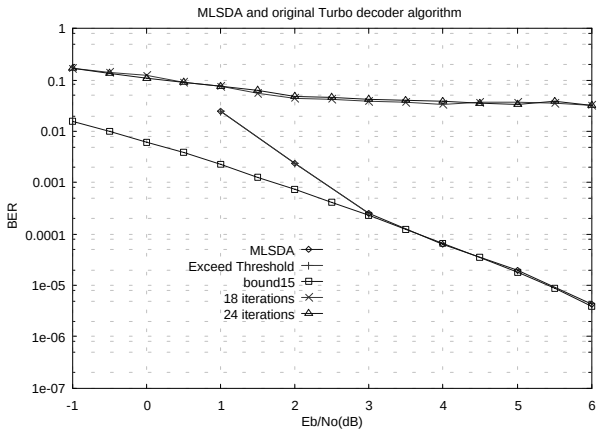
- L=36, G_1=37,G_2=21, R=1/3,RBI.



D	W_m
5	1
8	1
9	4
10	3
11	3
12	6
13	17
14	31
15	65

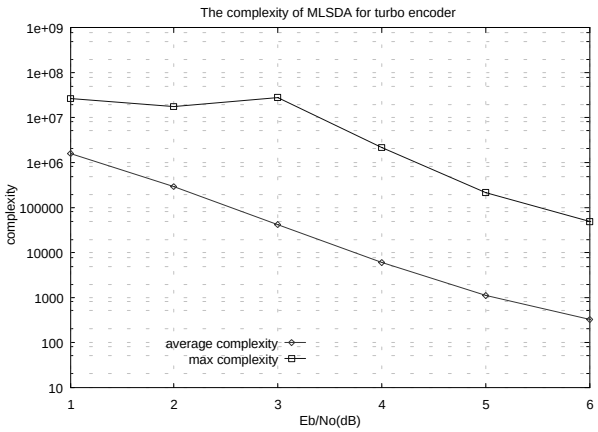
Simulation Result

- L=36, G_1=37,G_2=21, R=1/3,RBI.



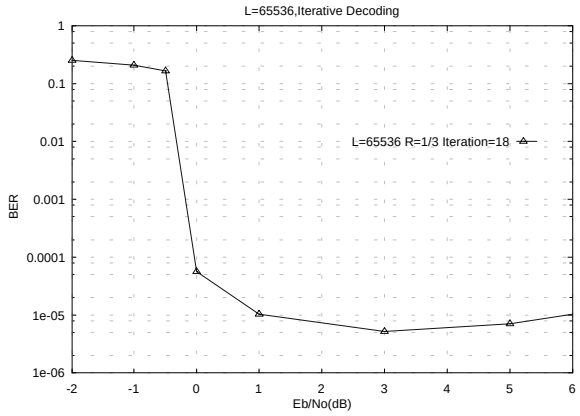
Simulation Result

- L=36, G_1=37,G_2=21, R=1/3,RBI.



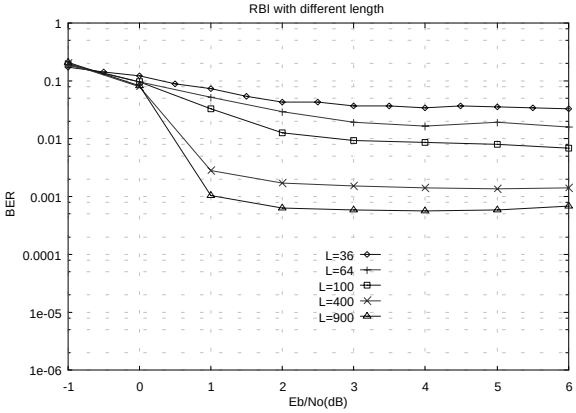
Simulation Result

- L=65536,R=1/3, G_1=37,G_2=21.



Simulation Result

- Change the information length L.



Observation

- When SNR is high, BER of MLSDA almost meet the ML decoding bound.
- If the length of the STACK exceeds Threshold, BER of MLSDA will be larger than the ML decoder.
- The complexity grows exponentially as SNR decreases.
- The reverse block interleaver has better performance.

Conclusions

- We confirm that the MLSDA for turbo encoder is an ML decoding algorithm.
- The high complexity of MLSDA at very low SNR make the algorithm infeasible.
- In certain situations, error floor phenomenon is induced by the decode, instead of the encoder.