A short introduction to Turbo Coding

Introduction to Graphical Models and Inference for Communications

UC3M

March 3, 2018

uc3m

Today

- Turbo Codes
- Encoder structure
- Interleaver
- Decoding
- Introduction to Lab. project 2

Turbo Codes

- High-performance forward error correction (FEC) codes developed around 1990-91 (but first published in 1993).
- First practical known codes to closely approach the channel capacity.
- 3G/4G mobile communications (e.g. in UMTS and LTE) and in (deep space) satellite communications.

Econder structure

- Two convolutional codes in parallel with some kind of interleaving in between.

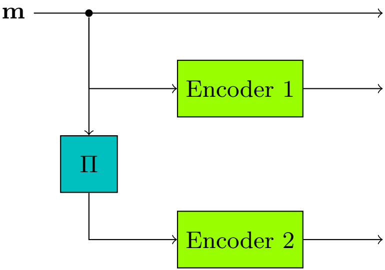


Figure: Turbo code encoder structure

- The frames can be terminated - i.e. the encoders are forced to a known state after the information block. The termination tail is then appended to the encoded information and used in the decoder.

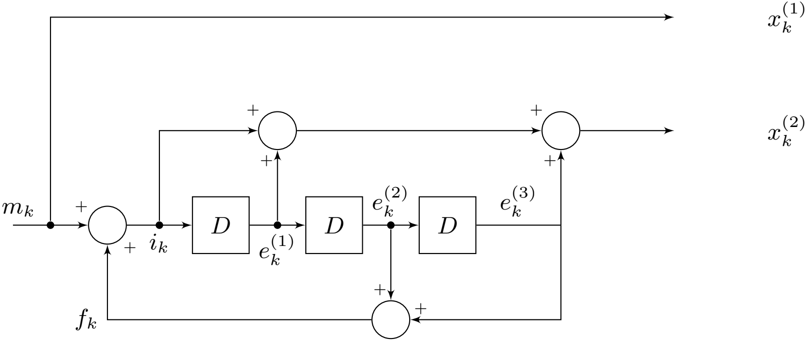
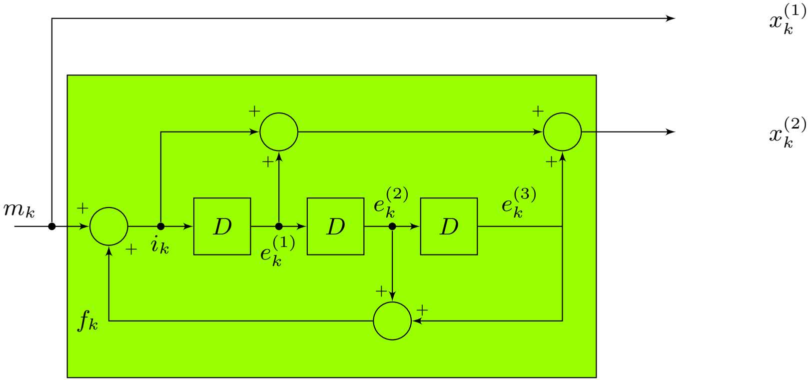
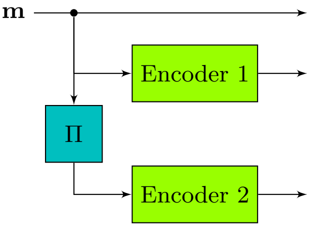


Figure: Rate-1/2 RSC Convolutional Code

3GPP RSC Turbo code





Coding rate $R = 1/3$

Interleaver

- Let L denote the length of the encoded sequence (we have to add the termination bits).
- The interleaver design is a key factor which determines the good performance of a turbo code.
- The input sequence to each of the two component RSC codes look “almost” independent.
- We need $L \to \infty$ to approximate channel capacity.

Row-Column interleaver

- Simplest interleaver. Data is written row-wise and read column-wise.
- Let K, T be positive integers such that $KT = L$.

$$\begin{bmatrix} m_1 & m_2 & m_3 & \dots & m_T \\ m_{T+1} & m_{T+2} & m_{T+3} & \dots & m_{2T} \\ m_{2T+1} & m_{2T+2} & m_{2T+3} & \dots & m_{3T} \\ \dots & \dots & \dots & \dots & \dots \\ m_{KT+1} & m_{KT+2} & m_{KT+3} & \dots & m_L \end{bmatrix}$$

- The input sequence to the encoder 1 is $m_1, m_2, m_3, \dots, m_L$.
- The input sequence to the encoder 2 is $m_1, m_{T+1}, m_{2T+1}, \dots, m_L$.

Helical interleaver

- Data is written row-wise and read data diagonal-wise.

$$\begin{bmatrix} m_1 & m_2 & m_3 & \dots & m_T \\ m_{T+1} & m_{T+2} & m_{T+3} & \dots & m_{2T} \\ m_{2T+1} & m_{2T+2} & m_{2T+3} & \dots & m_{3T} \\ \dots & \dots & \dots & \dots & \dots \\ m_{KT+1} & m_{KT+2} & m_{KT+3} & \dots & m_L \end{bmatrix}$$

- The input sequence to the encoder 1 is $m_1, m_2, m_3, \dots, m_L$.
- The input sequence to the encoder 2 is $m_1, m_{T+2}, m_{2T+3}, \dots$.

There exists a large set of interleaver designs proposed to date (even-odd, smile, pseudo-random, ...)

- $\mathbf{m} = m_0, \ldots, m_{L-1}$ is the input sequence to encoder 1.
- $\mathbf{m} = m'_0, \ldots, m'_{L-1}$ is the input sequence to encoder 2.

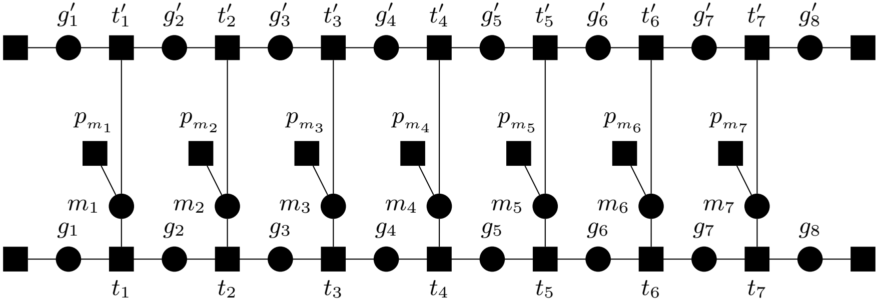
$$m'_i = m_{\pi(i)}, \qquad \pi(i) = \left(f_1 i + f_2 i^2\right) \mod L,$$

where the parameters f_1 and f_2 depend[1] on the block size L.

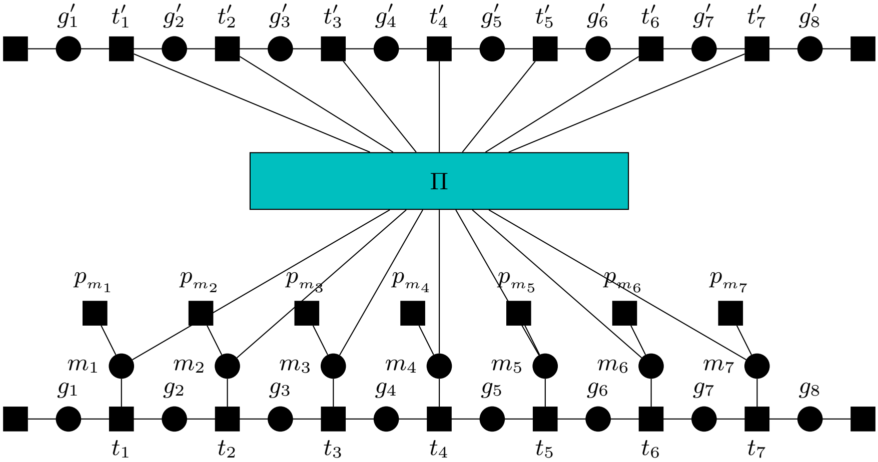
[1] Specification: Group Radio Access Network, Evolved Universal Terrestrial Radio Access, Multiplexing and Channel Coding (Release 8), TS 36.212 v8.3.0, May 2007

$p_{\mathbf{X}|\mathbf{y}}(m)$ factor graph

Without interlever → **very short cycles!**

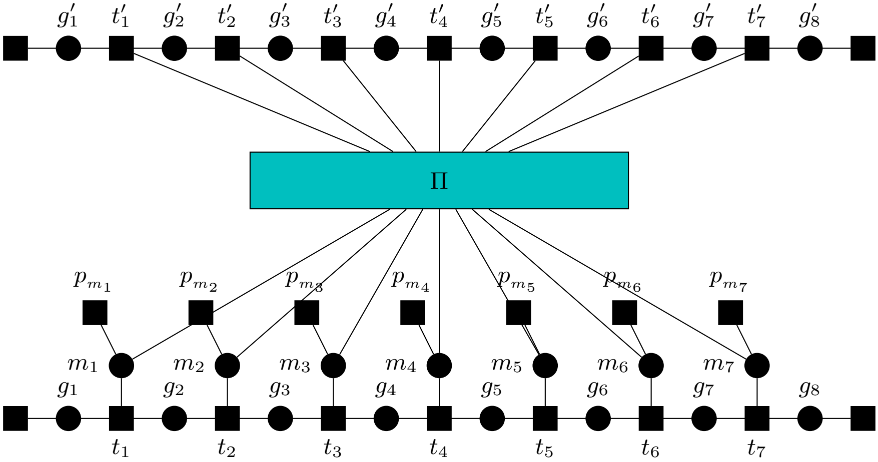


$p_{\mathbf{X}|\mathbf{y}}(m)$ factor graph



The FG still have cycles, but they are very long if $L \to \infty$ and the interleaver is properly designed.

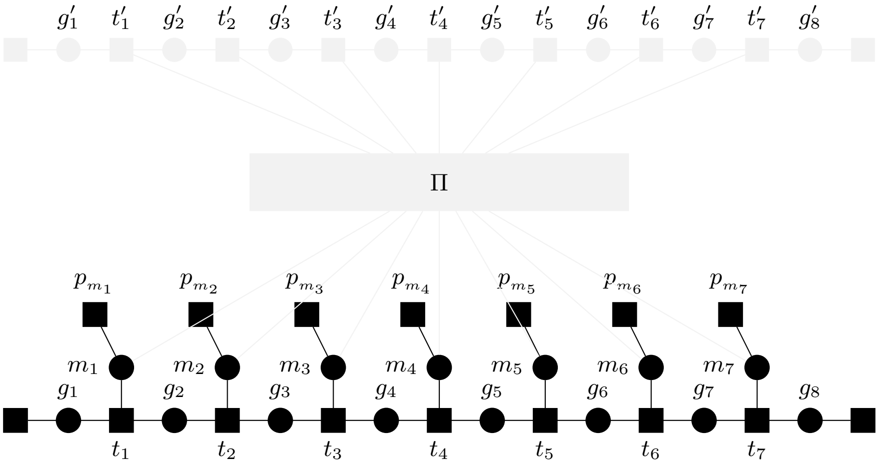
$p_{\mathbf{X}|\mathbf{y}}(m)$ factor graph



BP can be applied directly over the factor graph. However, the resulting algorithm is very complex (lots of state messages computed per iteration)

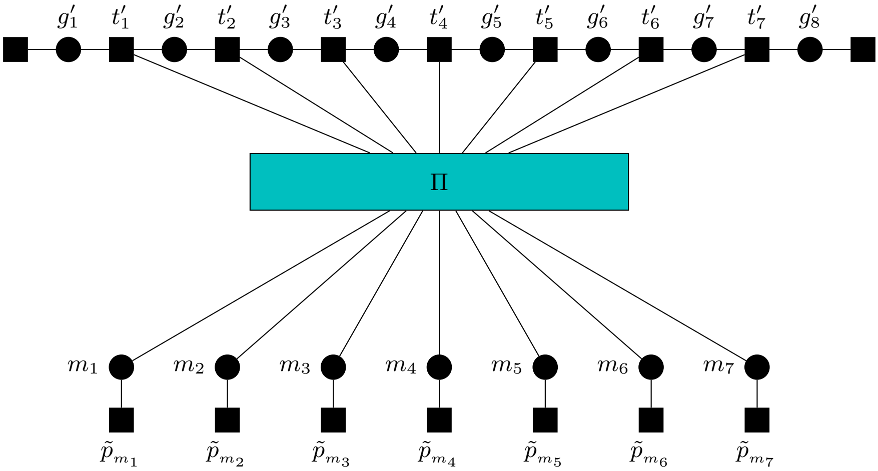
Sequential schedule (I)

We consider one chain at a time. First, BP is run over the chain corresponding to code 1 and uniform priors.



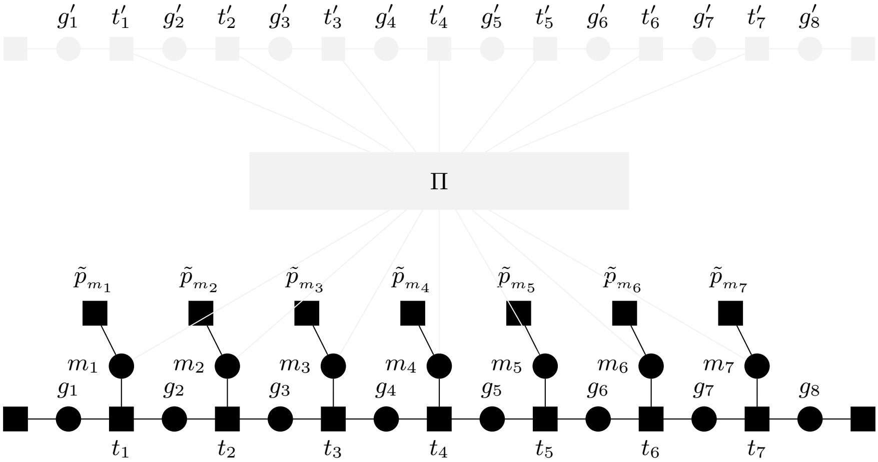
Sequential schedule (II)

Then, BP is run over the second chain using the priors estimated in the previous step.



Sequential schedule (III)

BP is run over the first chain using the priors estimated in the previous step.



We iterate until convergence. Tipically no more than 10-15 iterations.