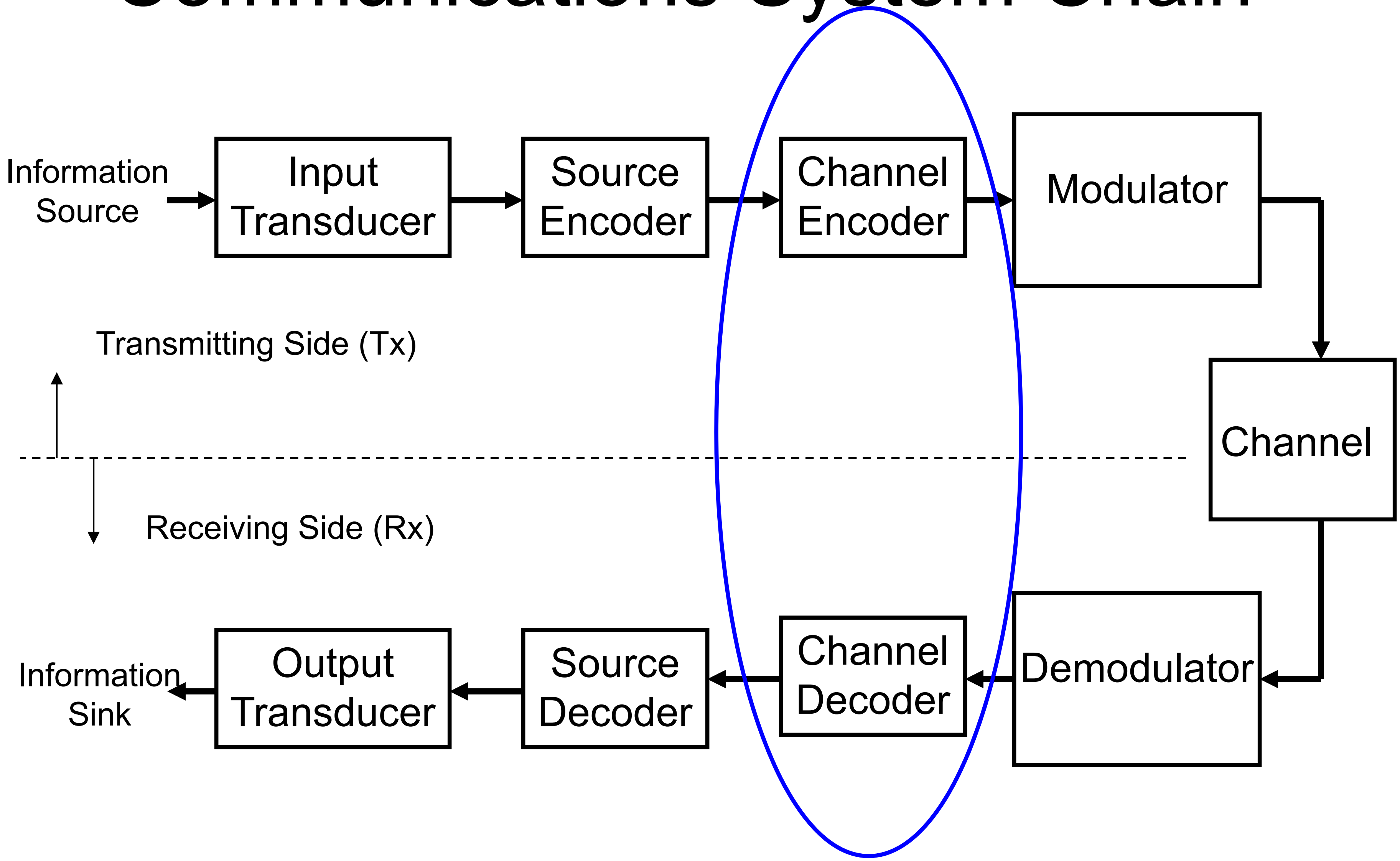
ECE 443:Turbo Codes An Approximation Approach for Probabilistic Modeling

Gaurav Sharma

Lecture Objectives

- So far, exact inference using probabilistic models
  - IID, Markov Models
  - HMMs and SCFGs
- Exact inference sometimes too hard
- Approx. often provides an excellent alternative
  - Example: Turbo Codes
  - Builds upon our example of Convolutional codes and HMM decoding

Channel Coding in the Communications System Chain



Recall: Convolutional Codes

- Systems view
  - k-input, n-output, linear shift invariant system
- State space view
  - Finite state causal system

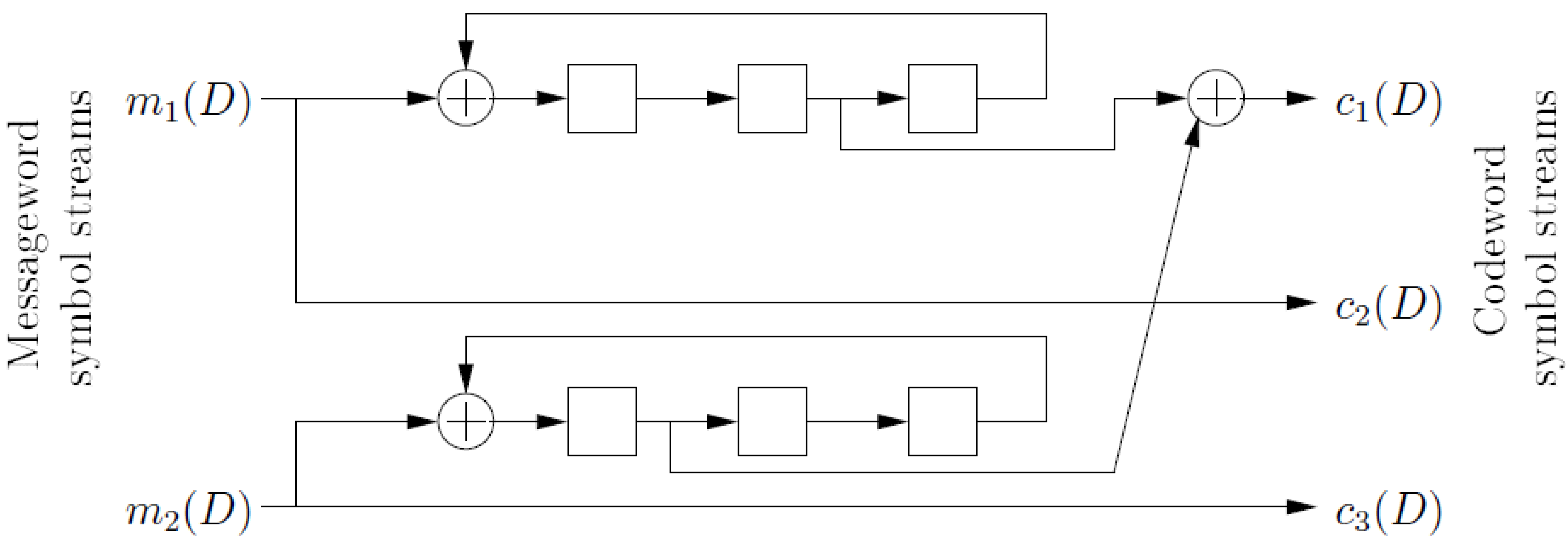
- Prior discussion emphasizes state space view and HMM based decoding
  - Viterbi (most likely state sequence)
  - Symbol-by-symbol MAP

Systematic Convolutional Encoders

- Analogous to systematic encoders for block codes
  - Input bits occur directly in the encoded stream
- Advantages:
  - Decoder can output hard decision symbol values when decoding fails
  - Useful in iterative decoding/concatenated coding (have reliability estimates for the message symbols)

A Recursive Systematic Convolutional Encoder

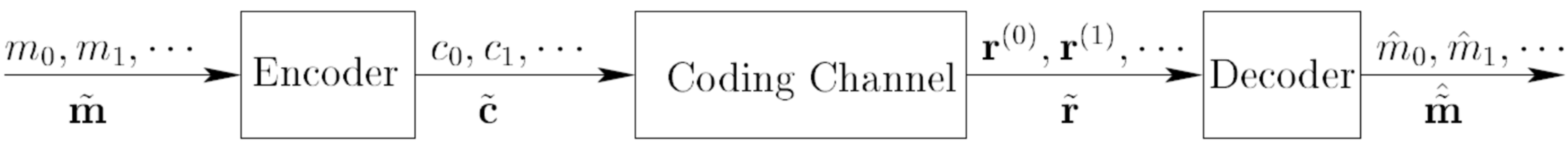
- A (2,3) Recursive Systematic Convolutional Encoder



$$\mathbf{G} = \begin{bmatrix} \frac{D^2}{1+D^3} & 1 & 0 \\ \frac{D}{1+D^3} & 0 & 1 \end{bmatrix}$$

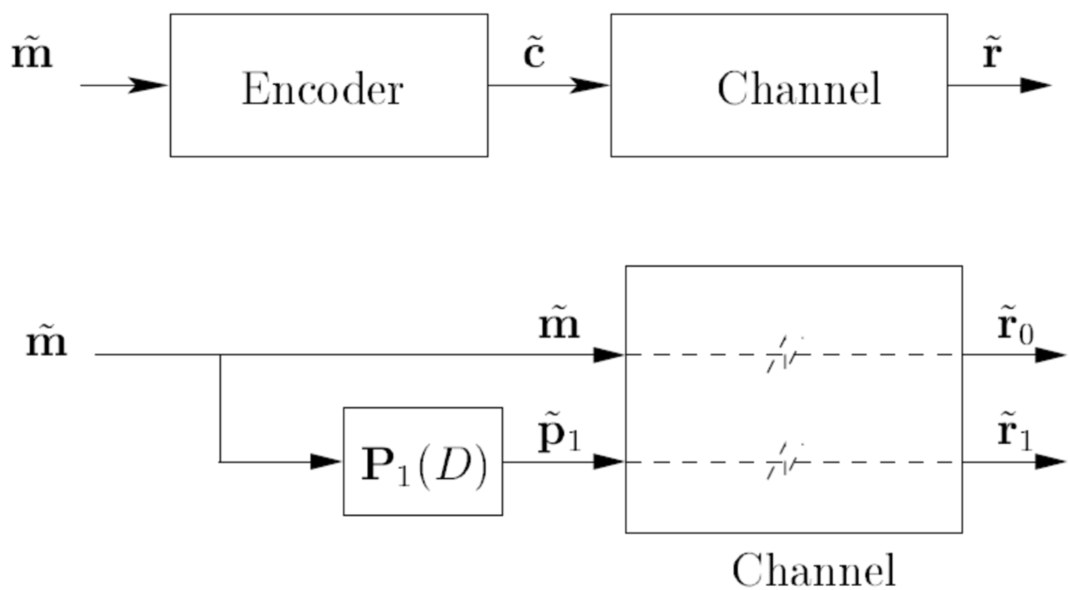
Convolutional Codes: Decoding Context

- Receiver seeks to estimate transmitted message stream based on received stream



Recursive Systematic Convolutional Codes: Decoding Context

- Receiver seeks to estimate transmitted message stream based on received stream



MAP Decoding of Convolutional Codes: Formulation

- Start with initial state 00
- L successive messageword frames

$$= \mathbf{m}^{(0)}, \mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(L-1)}$$

- Corresponding codeword frames

$$= \mathbf{c}^{(0)}, \mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(L-1)}$$

- and receivedword frames

$$\tilde{\mathbf{r}}_{(0)}^{(L-1)} = \mathbf{r}^{(0)}, \mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(L-1)}$$

$$\hat{\mathbf{m}}^{(l)} = \arg\max_{\mathbf{m}} p\left(\mathbf{m}^{(l)} = \mathbf{m} \mid \tilde{\mathbf{r}}_{(0)}^{(L-1)}\right)$$

Recall: MAP Decoding of Convolutional Codes

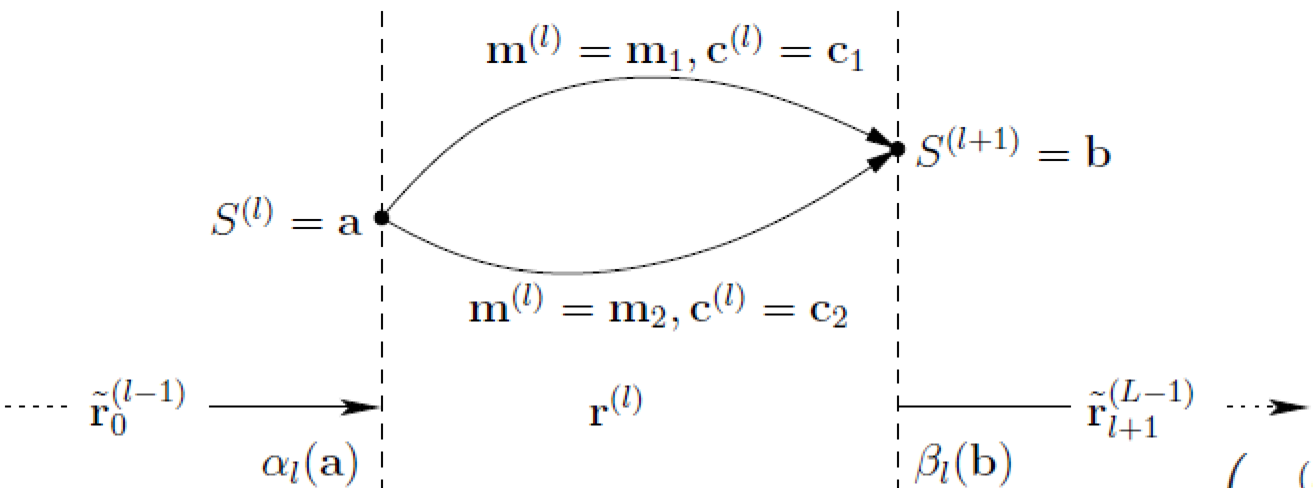
- MAP Decoding requires posterior probabilities

$$p\left(\mathbf{m}^{(l)} = \mathbf{a}, \tilde{\mathbf{r}}_{(0)}^{(L-1)}\right)$$

- Computed efficiently using forward-backward algorithm on code trellis
  – Can also incorporate prior probabilities for independent message words

MAP Decoding

- Dynamic programming analogous to ML decoding: forward-backward/BCJR Algo



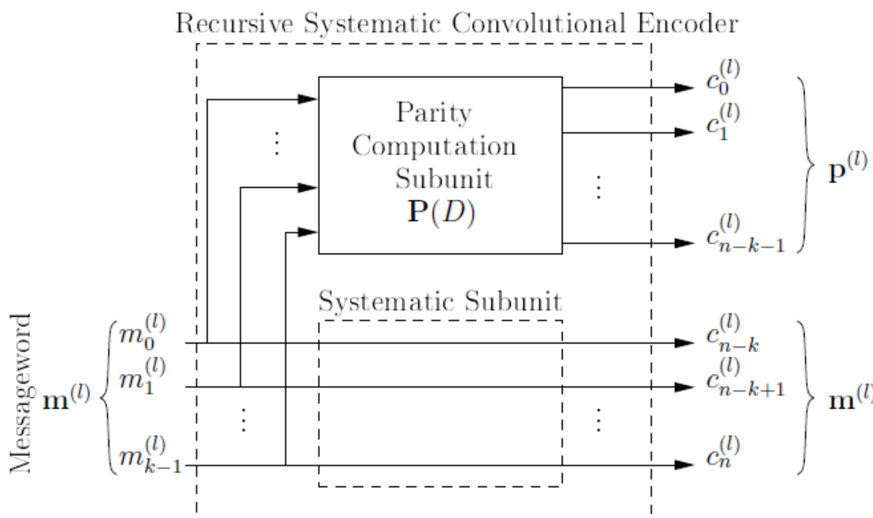
$$\alpha_l(\mathbf{a}) \stackrel{\text{def}}{=} p\left(\tilde{\mathbf{r}}_{(0)}^{(l-1)}, S^{(l)} = \mathbf{a}\right)$$

$$\beta_l(\mathbf{a}) \stackrel{\text{def}}{=} p\left(\tilde{\mathbf{r}}_{(l)}^{(L-1)} \mid S^{(l)} = \mathbf{a}\right)$$

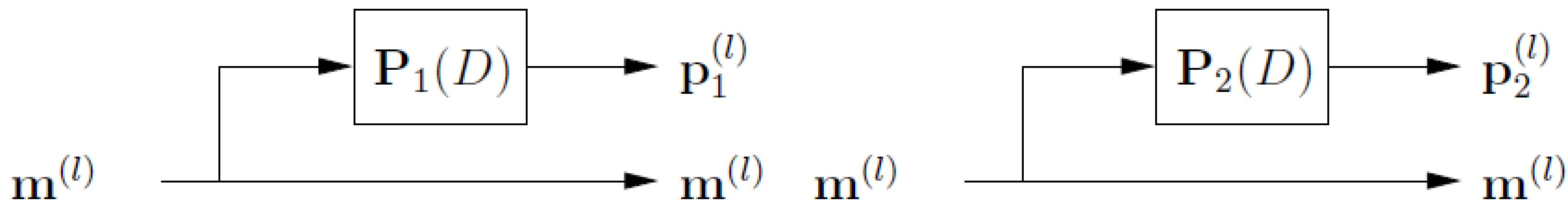
$$p\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}_{(0)}^{(L)}\right) = \sum_{\mathbf{a},\mathbf{b} \in \mathcal{S}} \alpha_l(\mathbf{a}) p\left(\mathbf{r}^{(l)}, \mathbf{m}^{(l)} = \mathbf{m}, S^{(l+1)} = \mathbf{b} \mid S^{(l)} = \mathbf{a}\right) \beta_{l+1}(\mathbf{b})$$

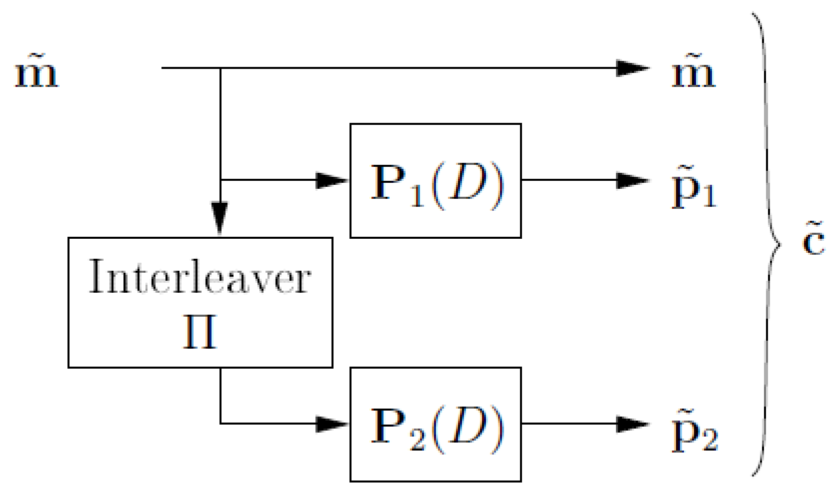
Turbo Codes

- Actually, “Turbo-Decoding” is the innovation rather than the code
- Built out of (parallel) concatenation of convolutional encoders with interleaver



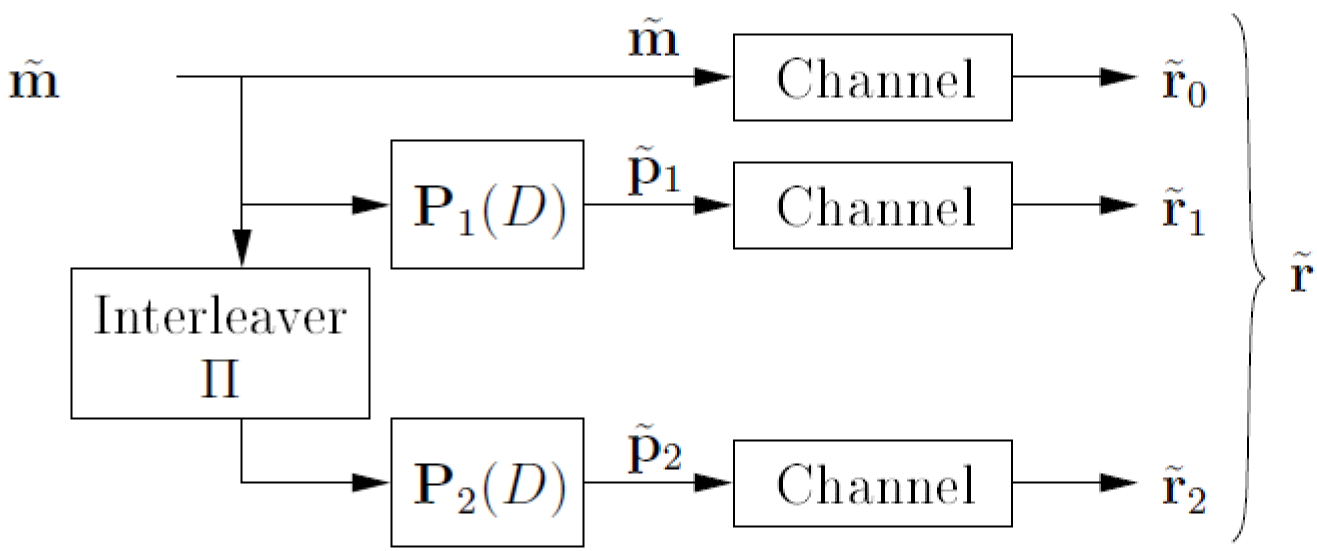
Turbo Codes: Encoding





Turbo Code: Decoding

- Joint decoding gives benefit of amortizing redundancy over large interleaver block length
  - Computationally horrendous



Exact MAP Decoding

- MAP Value of a symbol

$$\hat{\mathbf{m}}^{(l)} = \arg\max_{\mathbf{m}} p\left(\mathbf{m}^{(l)} = \mathbf{m} \mid \tilde{\mathbf{r}}\right)$$

- Baye’s Rule

$$\begin{aligned} \hat{\mathbf{m}}^{(l)} &= \arg\max_{\mathbf{m}} \frac{p\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right)}{p\left(\tilde{\mathbf{r}}\right)} \\ &= \arg\max_{\mathbf{m}} p\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right) \end{aligned}$$

Exact MAP Decoding

- MAP Probabilities
  - Recall Trellis HMM for convolutional codes

$$
\begin{aligned}
p\left(\mathbf{m}^{(l)}=\mathbf{m}, \tilde{\mathbf{r}}\right) &= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{m}}, \tilde{\mathbf{r}}\right) \\
&= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}} \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{m}}\right) \\
&= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{m}}\right) \\
&= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{m}}\right)
\end{aligned}
$$

Exact MAP Decoding

- IID message symbols

$$p\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right) \;=\; \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1} p\left(\mathbf{m}^{(i)}\right)$$

- Two interpretations:
  – First code term $p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1} p\left(\mathbf{m}^{(i)}\right)$
    - Multiplied by additional $p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right)$

  – Vice versa

Approximation I

- For first decoder approximate

$$p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \approx \prod_{j=0}^{L-1} T_j\left(\mathbf{m}^{(j)}\right)$$

- For second decoder approximate

$$p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \approx \prod_{j=0}^{L-1} U_j\left(\mathbf{m}^{(j)}\right)$$

Approximation II

- Two alternate expansions for $p\left(\mathbf{m}^{(l)}=\mathbf{m},\tilde{\mathbf{r}}\right)$

$$
\begin{aligned}
p\left(\mathbf{m}^{(l)}=\mathbf{m},\tilde{\mathbf{r}}\right) &= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{j=0}^{L-1} T_j\left(\mathbf{m}^{(j)}\right) \prod_{i=0}^{L-1} p\left(\mathbf{m}^{(i)}\right) \\
&= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1} \left(T_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
p\left(\mathbf{m}^{(l)}=\mathbf{m},\tilde{\mathbf{r}}\right) &= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{j=0}^{L-1} U_j\left(\mathbf{m}^{(j)}\right) \prod_{i=0}^{L-1} p\left(\mathbf{m}^{(i)}\right) \\
&= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1} \left(U_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right)
\end{aligned}
$$

Approximation III

- Rearrangement of alternate expansions

$$
\begin{aligned}
p\left(\mathbf{m}^{(l)}=\mathbf{m},\tilde{\mathbf{r}}\right) &= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1}\left(p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) T_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right) \\
&= p\left(\mathbf{r}_0^{(l)} \mid \mathbf{m}^{(l)}=\mathbf{m}\right) T_l\left(\mathbf{m}\right) \mathcal{P}\left(\mathbf{m}^{(l)}=\mathbf{m}\right) \times \\
&\quad \left[\sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0\\ i\neq l}}^{L-1}\left(p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) T_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
p\left(\mathbf{m}^{(l)}=\mathbf{m},\tilde{\mathbf{r}}\right) &= \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} \prod_{j=0}^{L-1} p\left(\mathbf{r}_0^{(j)} \mid \mathbf{m}^{(j)}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{i=0}^{L-1}\left(U_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right) \\
&= p\left(\mathbf{r}_0^{(l)} \mid \mathbf{m}^{(l)}=\mathbf{m}\right) U_l\left(\mathbf{m}\right) \mathcal{P}\left(\mathbf{m}^{(l)}=\mathbf{m}\right) \times \\
&\quad \left[\sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0\\ i\neq l}}^{L-1}\left(p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) U_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)\right)\right].
\end{aligned}
$$

Approximation IV

- Two representations coincide when

$$T_l\left(\mathbf{m}\right) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0\\ i\neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) U_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

$$U_l\left(\mathbf{m}\right) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0\\ i\neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(j)}\right) T_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

- Want joint solutions to these systems of equations: Iterative method – Turbo iteration

Iterative Solution on Trellises

- The "extrinsic information" terms

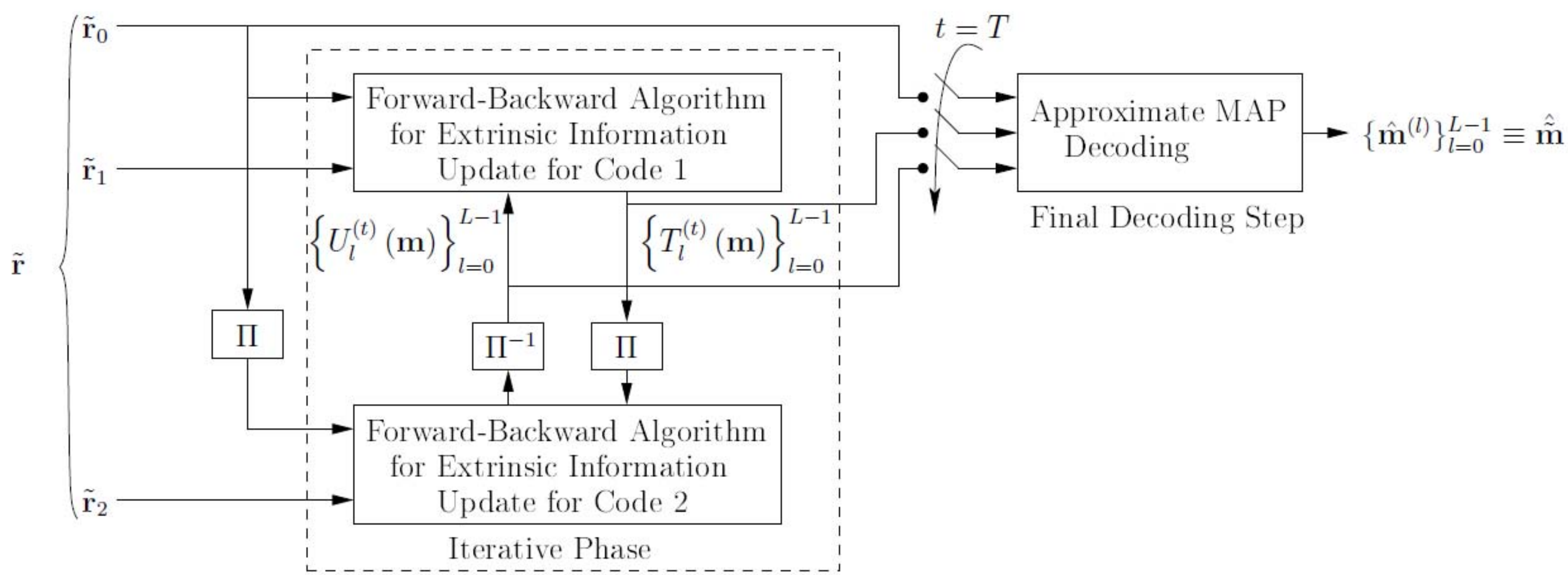
$$T_l\left(\mathbf{m}\right) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0 \\ i\neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) U_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

$$U_l\left(\mathbf{m}\right) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0 \\ i\neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(j)}\right) T_i\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

- Efficiently computed on individual code trellises
  - Modified BCJR (pseudo prior extrinsic information and exclusion of i=l term)

Turbo Code: Decoding

- Approx. MAP decoding: Iterate individual decodings with feedback
  - Per iteration complexity = BCJR
  - Performance close to joint decoding



Turbo Decoding as an Approximation

- A posterior probabilities

$$p\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_0 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) p\left(\tilde{\mathbf{m}}\right)$$

  - Prior probabilities $p\left(\tilde{\mathbf{m}}\right) = \prod_{i=0}^{L-1} p\left(\mathbf{m}^{(i)}\right)$

- Separable approximations (long interleaver)
  - Incorporated in BCJR as "pseudo-prior" probs.

$$p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \approx \prod_{j=0}^{L-1} T_j\left(\mathbf{m}^{(j)}\right) \qquad p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \approx \prod_{j=0}^{L-1} U_j\left(\mathbf{m}^{(j)}\right)$$

Turbo Decoding Algo. Overview

- Use modified BCJR to update extrinsic information for each component convolutional code
  - Use output extrinsic information from other decoder as “pseudo-prior” probabilities
- Iterate between the two decoders
  - Interleave/de-interleave as required
- Once iterations done, compute approximate *a posteriori* probabilities and decode

Turbo Decoding Algorithm

- Inputs: Received-word streams $\tilde{\mathbf{r}} = (\tilde{\mathbf{r}}_0, \tilde{\mathbf{r}}_1, \tilde{\mathbf{r}}_2)$
  - Messageword prior probabilities $\mathcal{P}\left(\mathbf{m}^{(l)} = \mathbf{m}\right)$
- Initialization: $t = 0$

$$T_l^{(t)}(\mathbf{m}) = 1, \quad l = 0, 2, \ldots (L-1), \forall \mathbf{m} \in F^k$$

- Extrinsic information updates (Modif. BCJR)

$$U_l^{(t)}(\mathbf{m}) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_1 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0 \\ i \neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) T_i^{(t-1)}\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

$$T_l^{(t)}(\mathbf{m}) = \sum_{\{\tilde{\mathbf{m}}:\mathbf{m}^{(l)}=\mathbf{m}\}} p\left(\tilde{\mathbf{r}}_2 \mid \tilde{\mathbf{m}}\right) \prod_{\substack{i=0 \\ i \neq l}}^{L-1} p\left(\mathbf{r}_0^{(i)} \mid \mathbf{m}^{(i)}\right) U_i^{(t)}\left(\mathbf{m}^{(i)}\right) p\left(\mathbf{m}^{(i)}\right)$$

Turbo Decoding Algorithm (Contd.)

- Symbol-by-symbol decoding
  - Compute approximate posterior probabilities

$$\hat{p}\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right) = p\left(\mathbf{r}_0^{(l)} \mid \mathbf{m}^{(l)} = \mathbf{m}\right) T_l^{(T)}(\mathbf{m}) \mathcal{P}\left(\mathbf{m}^{(l)} = \mathbf{m}\right) U_l^{(T)}(\mathbf{m})$$

  - Max approx. aposteriori prob decoding

$$\begin{aligned} \hat{\mathbf{m}}^{(l)} &= \arg\max_{\mathbf{m}} \hat{p}\left(\mathbf{m}^{(l)} = \mathbf{m}, \tilde{\mathbf{r}}\right) \\ &= \arg\max_{\mathbf{m}} p\left(\mathbf{r}_0^{(l)} \mid \mathbf{m}^{(l)} = \mathbf{m}\right) \mathcal{P}\left(\mathbf{m}^{(l)} = \mathbf{m}\right) T_l^{(T)}(\mathbf{m}) U_l^{(T)}(\mathbf{m}) \end{aligned}$$

Turbo Decoding: Observations

- Approximate Decomposition of problem of inference on complete data into
  - Two problems that are “loosely” coupled together by the exchange of “extrinsic” information: local probabilistic information
  - Individual problems are tightly coupled (trellis computation)
- Computation of local probabilistic information via (modified) BCJR (forward-backward) is critical
  - Not immediately obvious for Viterbi decoding

Turbo Code Performance Example

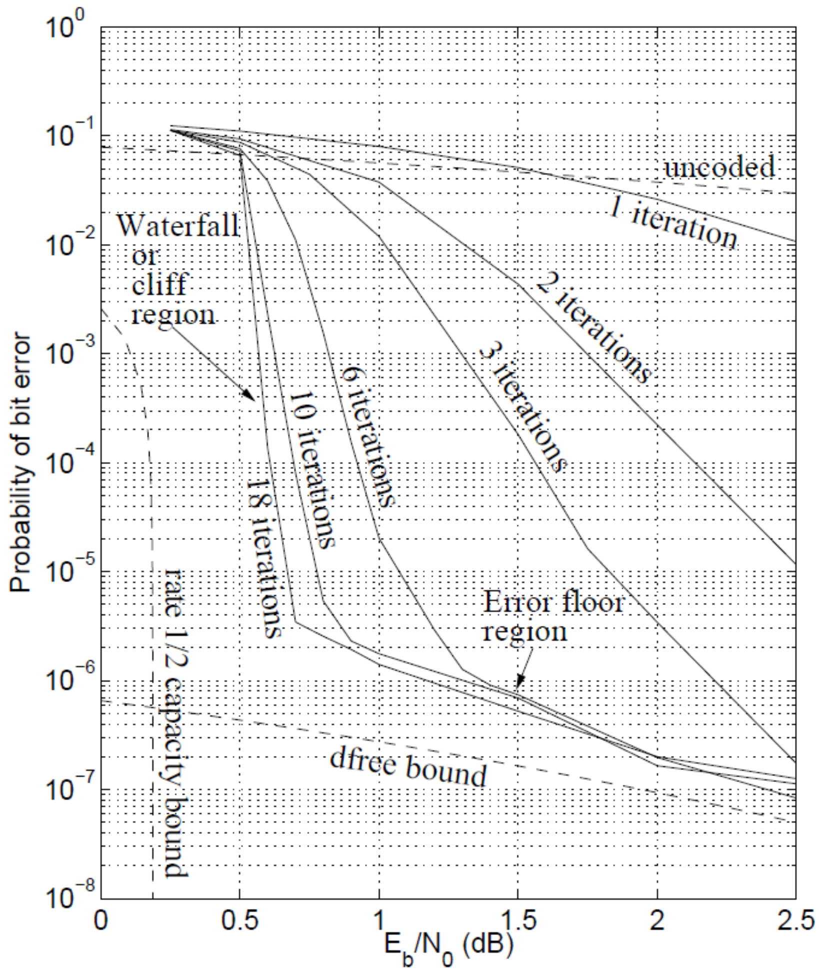


Figure 14.1: Decoding results for a (37,21,65536) code

From ECC, Todd Moon

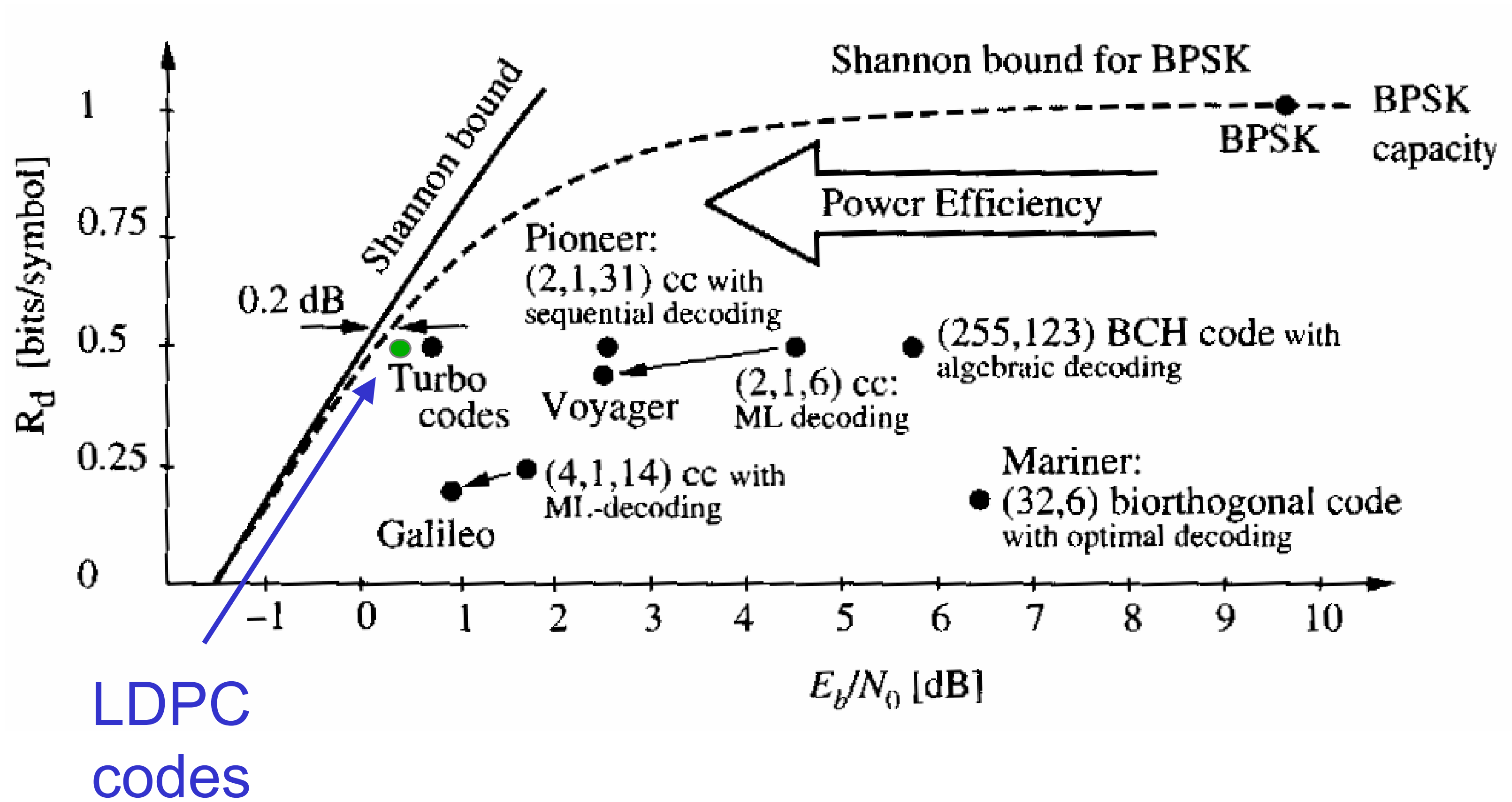
Turbo Code Performance

- Performance within a fraction of a dB of the capacity is achievable using turbo codes
  – Long interleaver lengths
  – Large number of iterations
    - Methods for determining when to terminate (e.g. cross-entropy)
- Practical applications pose restrictions
- Often concatenated with an outer algebraic code to correct residual error floors

Turbo Code Use

- Space
  - Mars Orbiter, Cassini (launched 1997), …
- 3G/4G Mobile Telephony Standards
  - HSPA, EV-DO and LTE
- IEEE Wireless Standards
  - IEEE 802.16 (WiMAX)
- Digital Video Broadcasting - Return Channel via Satellite (DVB-RCS)

Coding in the Power-limited Domain



Source: Trellis and Turbo Coding, Schlegel and Perez, IEEE Press, 2004

Summary

- Example of Approximate Inference using Probabilistic Models
  - Turbo Codes
  - Exact MAP estimation would be exponential in interleaver length and is computationally infeasible
  - Approximation saves the day
    - “Linear complexity”
    - Rather small performance compromise
      - Performance quite close to theoretical limit (capacity)