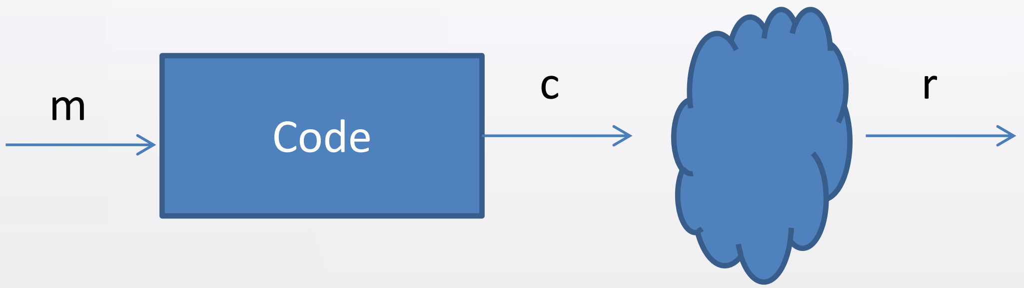
Communication Theory II

Lecture 11: Viterbi Algorithm

How to decode convolution codes



Likelihood function

$$p(\mathbf{r}|\mathbf{c}) = \prod_{i=1}^{N} p(r_i|c_i)$$

Log likelihood function

$$\log p(\mathbf{r}|\mathbf{c}) = \sum_{i=1}^{N} \log p(r_i|c_i)$$

Transition probability

$$p(r_i|c_i) = \begin{cases} p, & \text{if } r_i \neq c_i \\ 1-p, & \text{if } r_i = c_i \end{cases}$$

Hamming distance between received signal is d, now Log likelihood function is given by

$$\begin{aligned} \log p(\mathbf{r}|\mathbf{c}) &= d \log p + (N-d)\log(1-p) \\ &= d \log\left(\frac{p}{1-p}\right) + N \log(1-p) \end{aligned}$$

Choose the estimate $\hat{\mathbf{c}}$ that minimizes the Hamming distance between the received vector $\mathbf{r}$ and the transmitted vector $\mathbf{c}$.

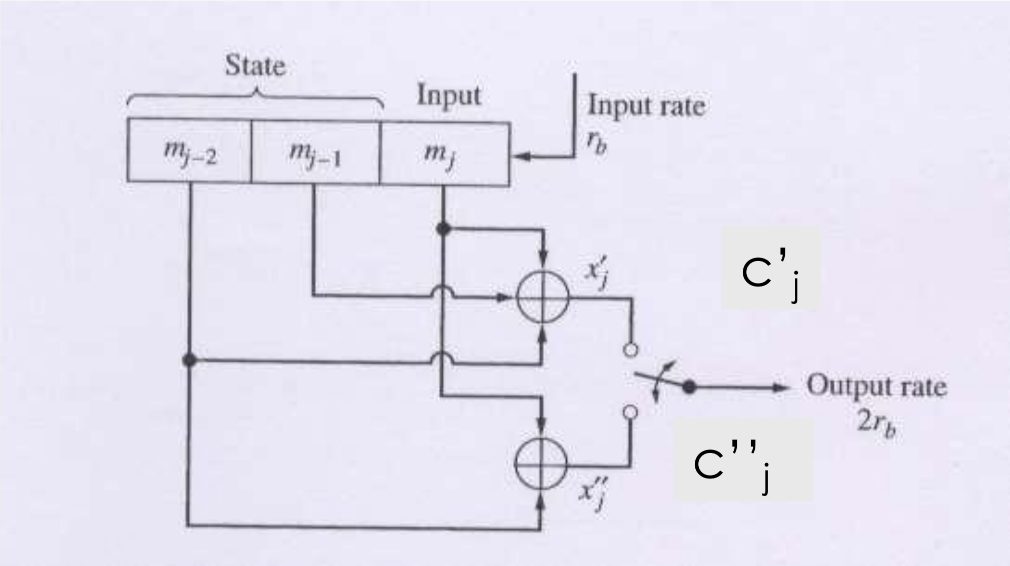
Viterbi Algorithm

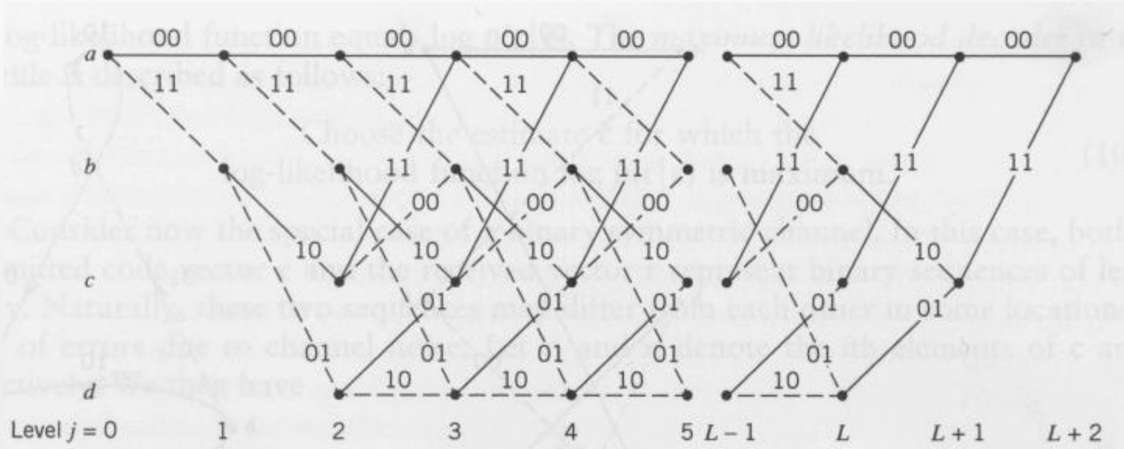
- The Viterbi Decoder is the key to overcoming the decoding challenges posed by Convolutional Codes. By leveraging the Viterbi Algorithm, this decoder identifies the most likely transmitted bit sequence, correcting errors in the process.

Viterbi Algorithm

- Initialization: Start with the first received symbol and initialize state metrics for all possible states.
- Metric Calculation: Compute metrics for all possible state transitions based on received symbols.
- Path Metric Update: Update path metrics for each state, considering incoming transitions and previous metrics.
- Survivor Path Selection: Select the incoming transition that contributes to the highest path metric (lowest hamming distance) for each state.

Viterbi Algorithm





https://www.youtube.com/watch?v=IJE94FhyygM&ab_channel=IainExplainsSignals%2CSystems%2CandDigitalComms

Encoded bits from stream 1, $c'_j = m_{j-2} + m_{j-1} + m_j$

Encoded bits from stream 2, $c''_j = m_{j-2} + m_j$

Decode 0001011010

Additional reading

Example

10.16 Consider the rate $r = 1/2$, constraint length $K = 2$ convolutional encoder of Fig. P10.16. The code is systematic. Find the encoder output produced by the message sequence 10111. . . .

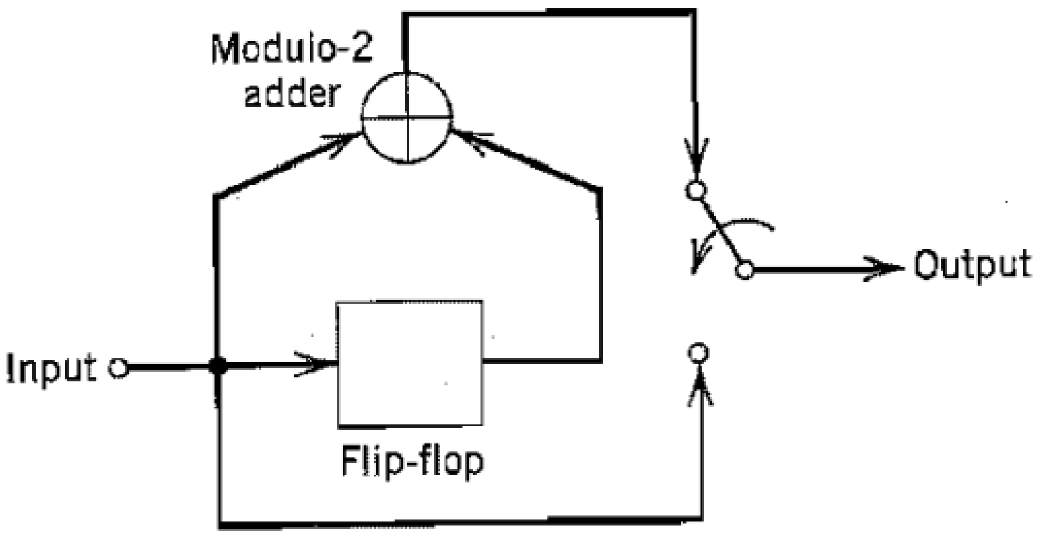


FIGURE P10.16

Message	1	0	1	1	1	1	...
Output	11	10	11	01	01	01	...

Example

10.17 Figure P10.17 shows the encoder for a rate $r = 1/2$, constraint length $K = 4$ convolutional code. Determine the encoder output produced by the message sequence 10111. . . .

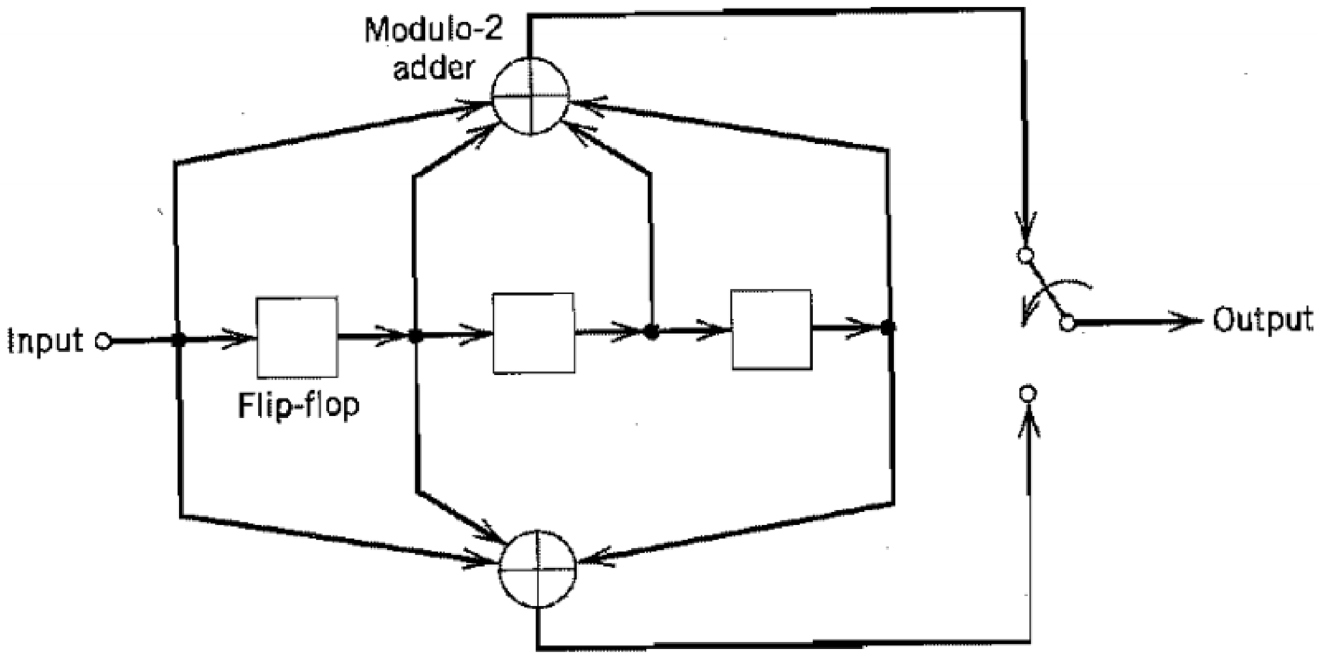


FIGURE P10.17

$$\{c^{(1)}\} = 1,1,0,1,1,1,...$$

$$\{c^{(2)}\} = 1,1,1,1,0,0,...$$

The encoder output is therefore 11, 11, 01, 11, 10, 10.

Example

The trellis diagram of a rate-1/2, constraint length-3 convolutional code is shown in Figure P10.25. The all-zero sequence is transmitted, and the received sequence is 100010000. . . . Using the Viterbi algorithm, compute the decoded sequence.

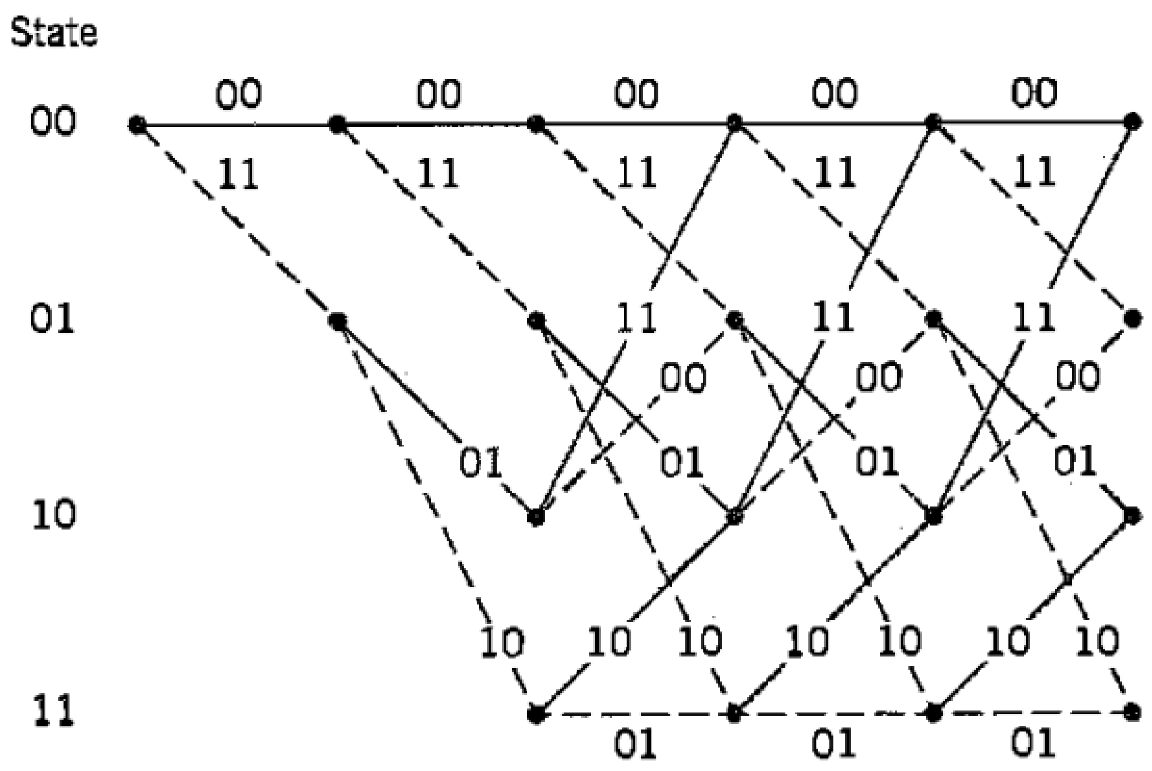
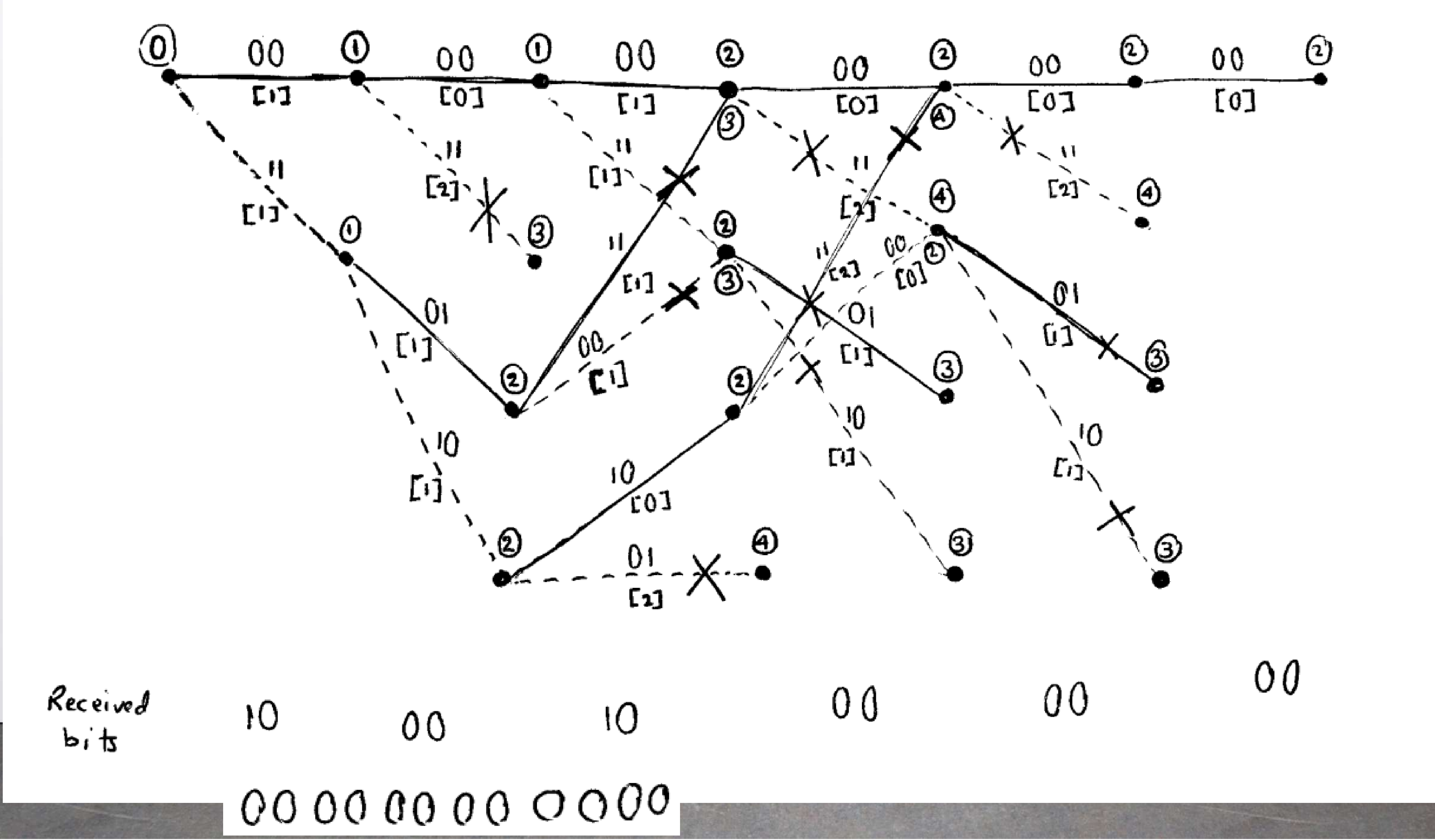


FIGURE P10.25

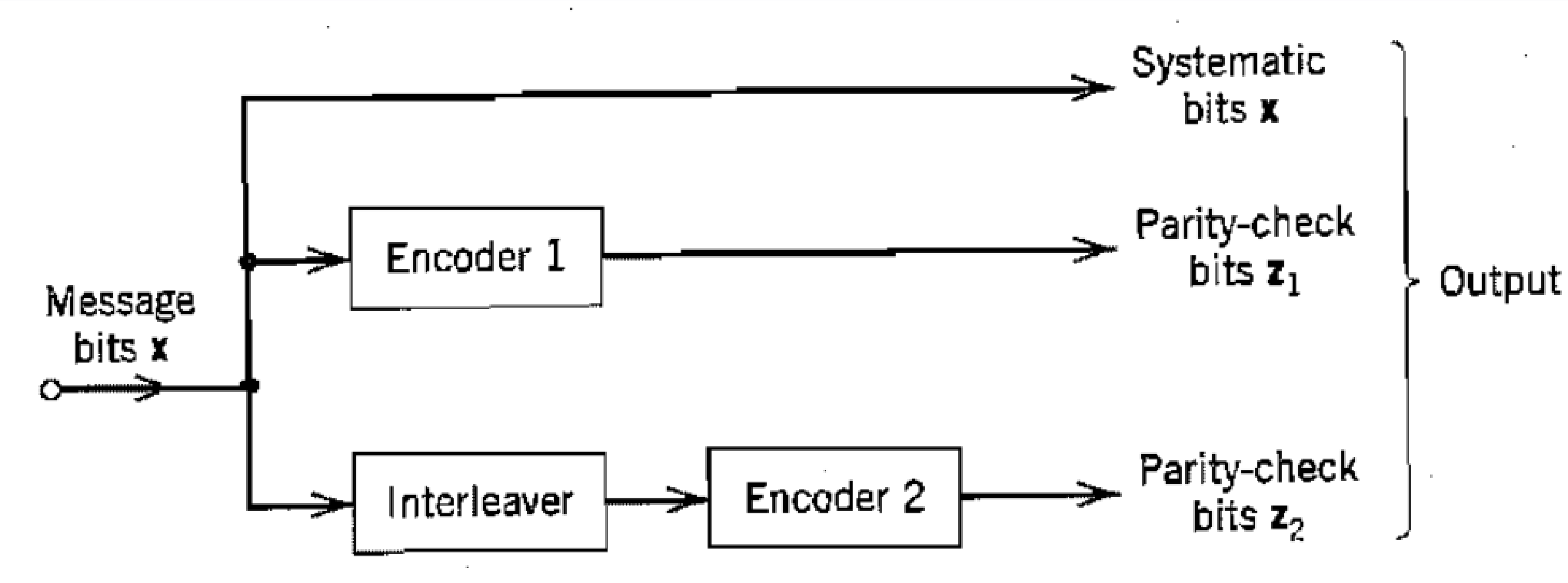
Example



Turbo codes

- Turbo codes are advanced error correction codes used in digital communication systems.
- Introduced by Claude Berrou in the early 1990s.
- Turbo codes consist of two or more recursive convolutional codes.
- Encoding involves passing data through multiple encoders and interleaving the outputs.
- Decoding is an iterative process that refines the estimate of the original data.
- Widely used in wireless communication, satellite links, and other noisy channels.
- Key element in 3G, 4G, and beyond cellular networks

Turbo codes

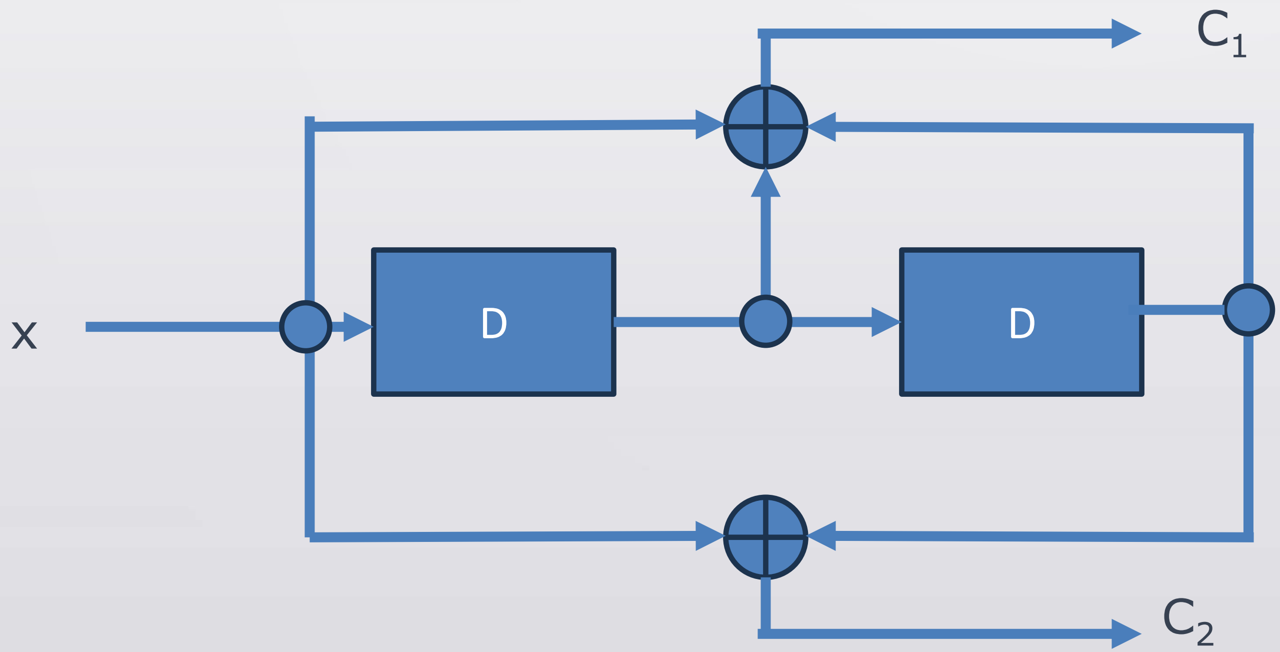


Random Interleaver: This type of interleaver randomly shuffles the bits, which provides good performance in terms of error correction

Recursive Systematic Convolutional (RSC) Encoder

It is a specific class of convolutional encoder

RSC (Recursive Systematic Convolutional) encoder does use feedback in its encoding process. The feedback loop is a key feature of convolutional encoders, including RSC encoders.



g_1=[1 1 1]

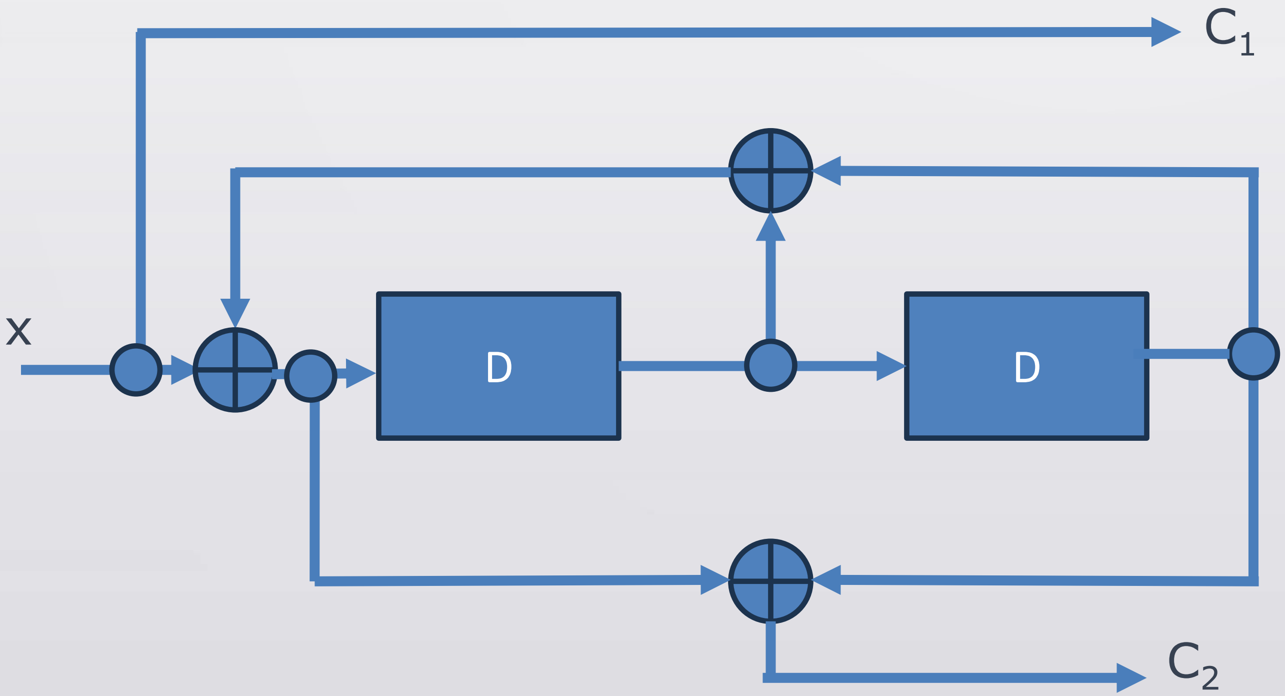
g_2=[1 0 1]

Conventional convolutional encoder

Recursive Systematic Convolutional (RSC) Encoder

Recursive Systematic Convolutional (RSC) encoders are commonly used in the construction of Turbo codes

RSC encoder



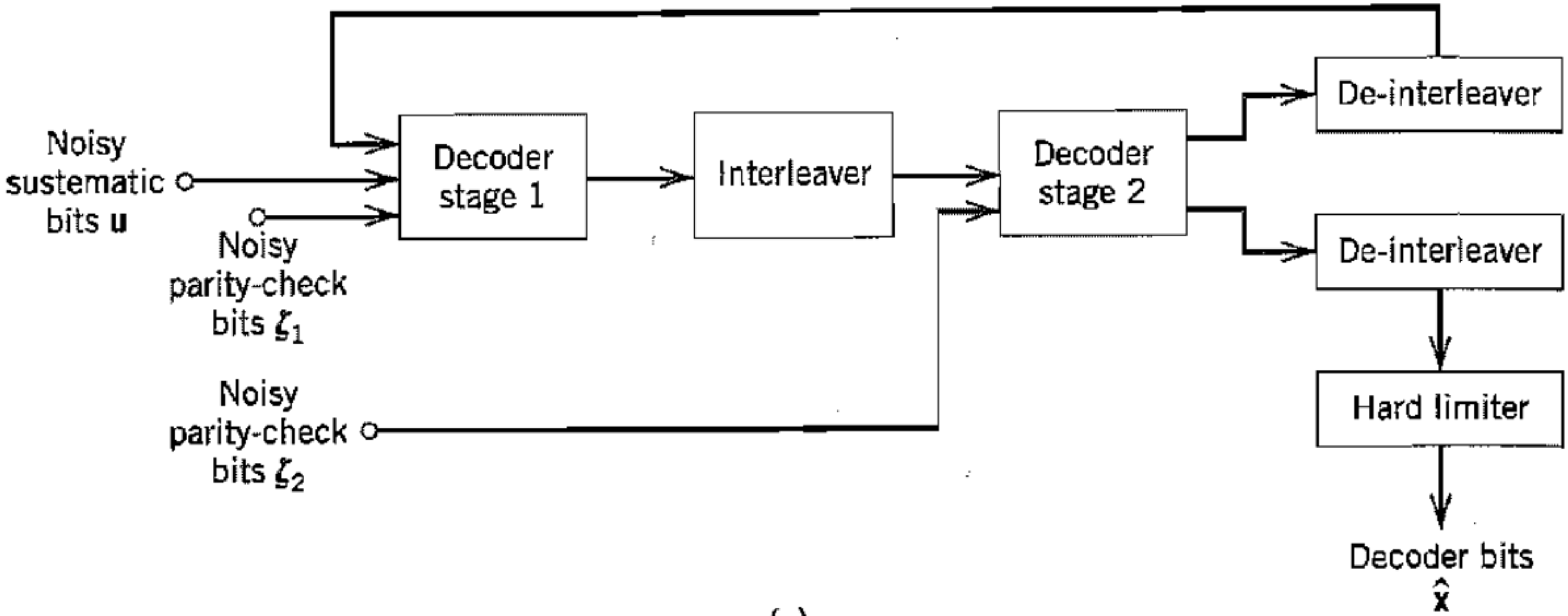
g_1=[1 1 1] feedback to the input

g_2=[1 0 1] feedforward output

$G=[1\ ,\ g_2\ /g_1\ ]$

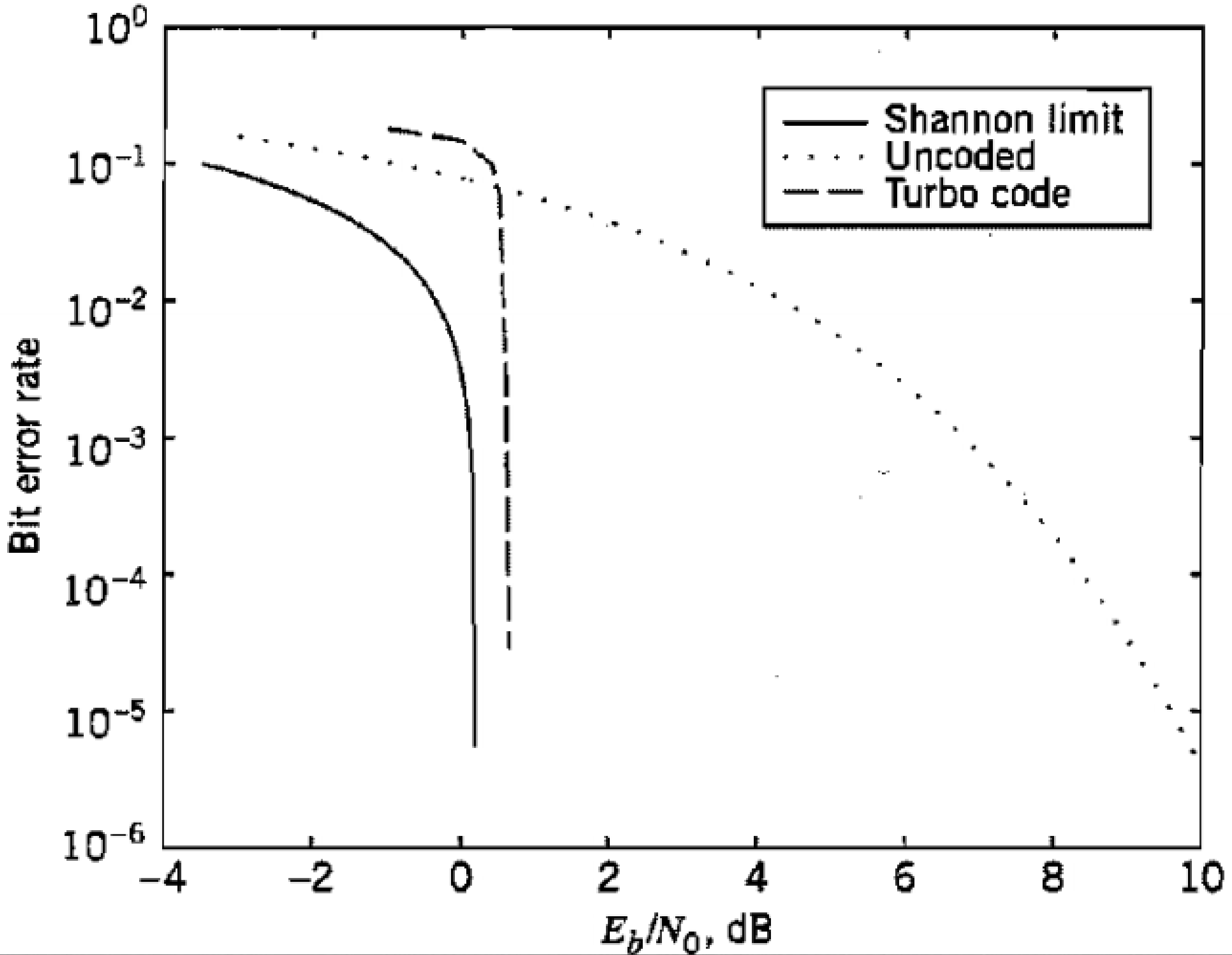
1 denotes the systematic output

Turbo codes- decorder



(a)

Performance of turbo codes



C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: turbo codes, ” Proc. ICC’ 93, pp. 1064-1047, Geneva, May 1993