UCCS

ECE 5965/4965
Introduction to Space Communications

Digital Communications in Space Networks

Channel Coding

Reading: Handouts

Fall 2022

1.0 Administrative

- Quiz 10, question 7 was in error, so I've given everyone credit on this question.
- Quiz 11, question 6, I had a typo in my slides, "given an element spacing of lambda/2 (larger results in grating lobes)."
- I may drop the 2 lowest quiz scores, if I can figure out an easy way to do this in Excel?
- No class next week (Thanksgiving). Extra time to work on the next project. There will be a quiz from this week's work that will be due on Dec. 1.

1.0 Outline

Channel coding to improve performance of satellite communication links over noisy/fading/etc. channel

1.0 Outline:

2.0 Introduction/Motivation

3.0 Review of Communications System Block Diagram

4.0 Detection Theory – BER

5.0 Channel Coding

1.0 Outline

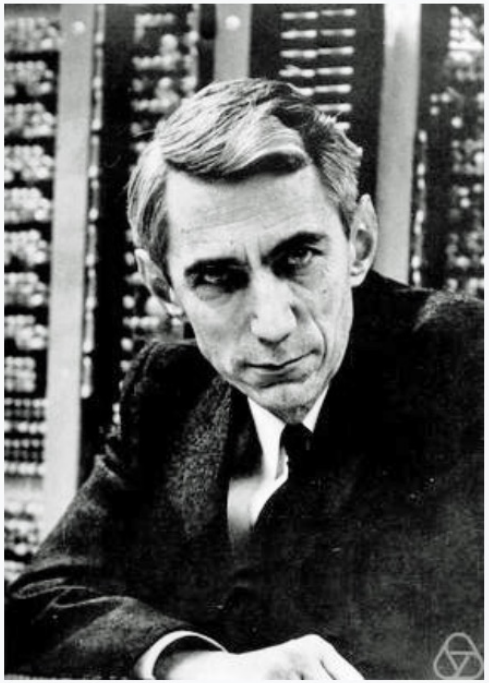
Claude Elwood Shannon (April 30, 1916 – February 24, 2001) was an American mathematician, electrical engineer, and cryptographer known as a "father of information theory".[1][2]

As a 21-year-old master's degree student at the Massachusetts Institute of Technology (MIT), he wrote his thesis demonstrating that electrical applications of Boolean algebra could construct any logical numerical relationship.[3] Shannon contributed to the field of cryptanalysis for national defense of the United States during World War II, including his fundamental work on codebreaking and secure telecommunications.

In 1948, the promised memorandum appeared as "A Mathematical Theory of Communication", an article in two parts in the July and October issues of the *Bell System Technical Journal*. This work focuses on the problem of how best to encode the message a sender wants to transmit. Shannon developed information entropy as a measure of the information content in a message, which is a measure of uncertainty reduced by the message. In so doing, he essentially invented the field of information theory.

The Bit Player tells the story of an overlooked genius who revolutionized the world, but never lost his childlike curiosity.

Claude Shannon



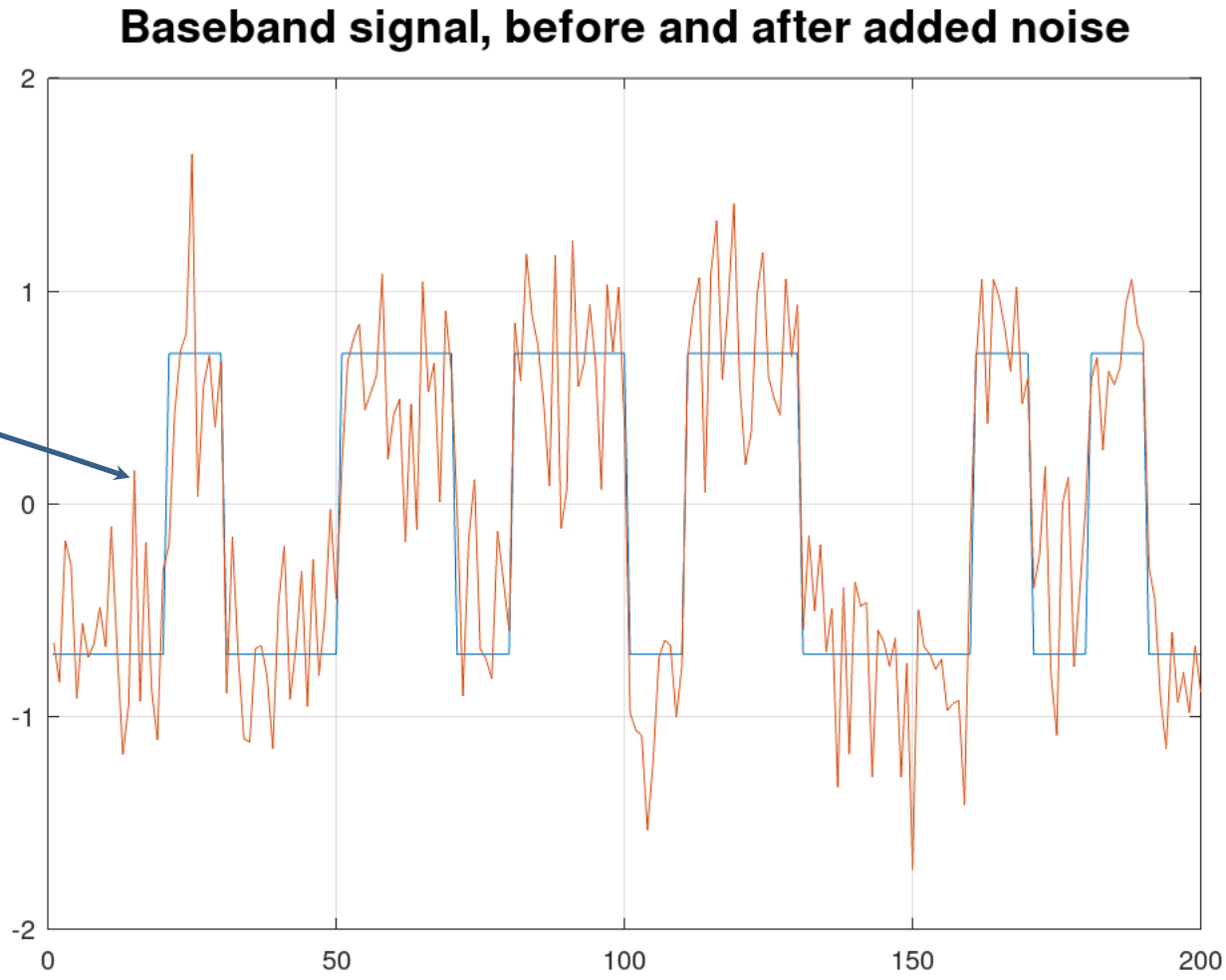
Born	April 30, 1916 Petoskey, Michigan, U.S.
Died	February 24, 2001 (aged 84) Medford, Massachusetts, U.S.
Citizenship	United States
Alma mater	University of Michigan (A.B., B.S.E.E.) Massachusetts Institute of Technology (M.S., Ph.D.)

2.0 Introduction/Motivation: Noisy Channel

bits to be transmitted:
0,0,1,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,1,0

Note that if the receiver just sampled the received waveform at the middle of the bit period, it might make an error here

In general, to improve the bit error rate, the receiver should probably integrate over the bit period, but errors still can be made depending on the amount of noise added to the signal.



2.0 Introduction/Motivation: Add parity Protection

Tack on an extra “even” parity bit:
0,0,1,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,1,0,1

The even parity works as follows:

- Count the number of 1s in the information bits (in this case, there are 9)
- If the number of 1s is odd, then add on a ‘1’ to make the whole word have an even number of 1s.
- Similarly, if the number of 1s is even, then add on a ‘0’ to keep an even number of 1s.
- The receiver then counts the total number of 1s (including the parity bit), and if it is even, no error occurred. Otherwise, one of the bits is in error.

From the previous slide, assuming the second bit is in error, the receiver sees:
0,1,1,0,0,1,1,0,1,1,0,1,1,0,0,0,1,0,1,0,1
and we see that there are 11 1s, so that we know that at least one bit is in error.

Note: that this coding scheme will only detect errors, not correct them. I.e., it doesn’t tell us which bit was in error.

2.0 Introduction/Motivation: Use a block code

For example, the Hamming (7,4) code can correct up to 1 bit errors. The code is defined by a generator matrix, G, and a parity check matrix H:

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Then, for an input set of bits, x, the codeword is found by multiplying x by xG = y. E.g., if x = [1 0 0 1], then y = [0 1 1 1 0 0 1]. Note that the code is systematic in that the input bits show up in the codeword (in red). Now, assume that we receive, z = [0 1 1 0 0 0 1]. If we post-multiply z by H, we get, Hz = [1 1 0] … next page

2.0 Introduction/Motivation: Use a block code

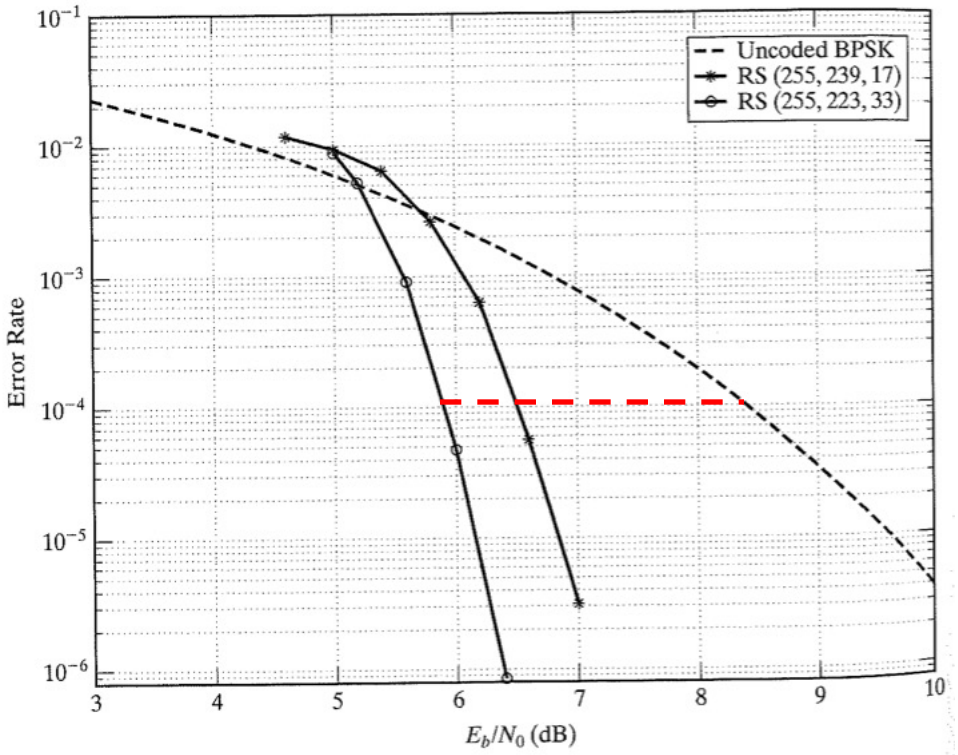
cont'd: We look up the syndrome, Hz = [1 1 0] in the following table:

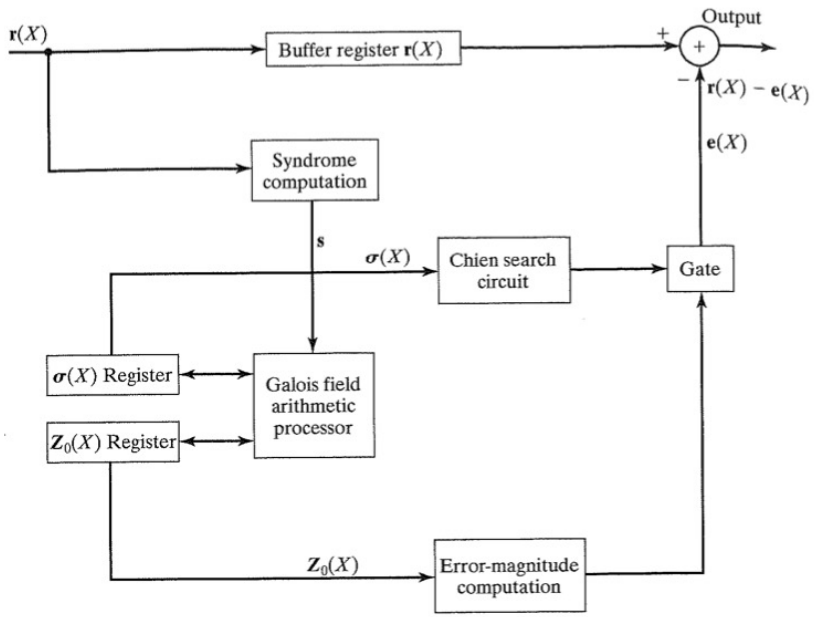
Syndrome	Error Vector
[1 0 0]	[1 0 0 0 0 0 0]
[0 1 0]	[0 1 0 0 0 0 0]
[0 0 1]	[0 0 1 0 0 0 0]
[1 1 0]	[0 0 0 1 0 0 0]
[0 1 1]	[0 0 0 0 1 0 0]
[1 1 1]	[0 0 0 0 0 1 0]
[1 0 1]	[0 0 0 0 0 0 1]

Indicates that the 4th bit is in error, and we can correct by inverting this bit.

2.0 Introduction/Motivation: BER Improvement [LIN04]

BER performance improvement using Reed-Solomon code



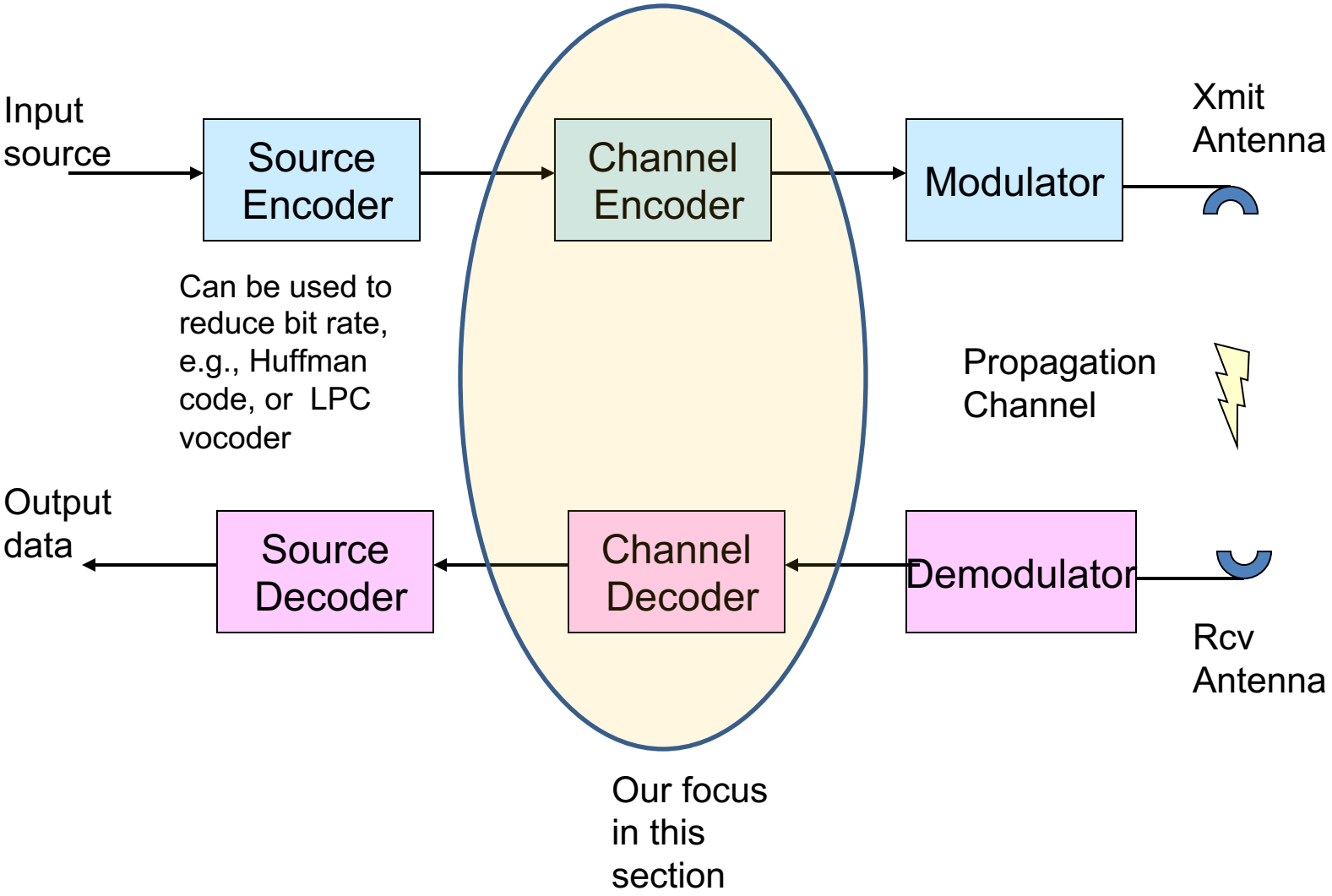


However, note the additional decoding complexity

Note that to achieve a BER of 10^-4, we can use a little over 2 dB less transmit power with a RS code

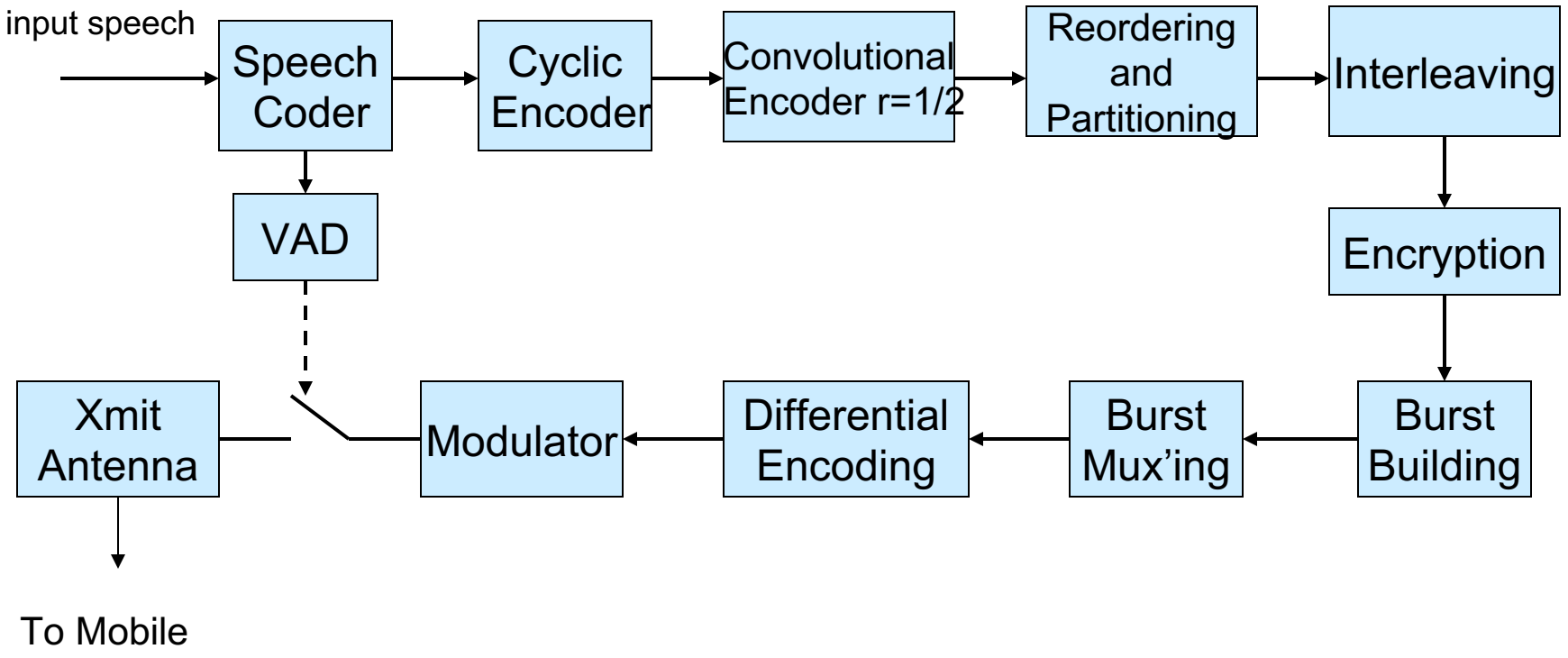
Also, note that the RS(255,223,33) code is a NASA standard for space and satellite communication.

3.0 Review of Communication System Blocks



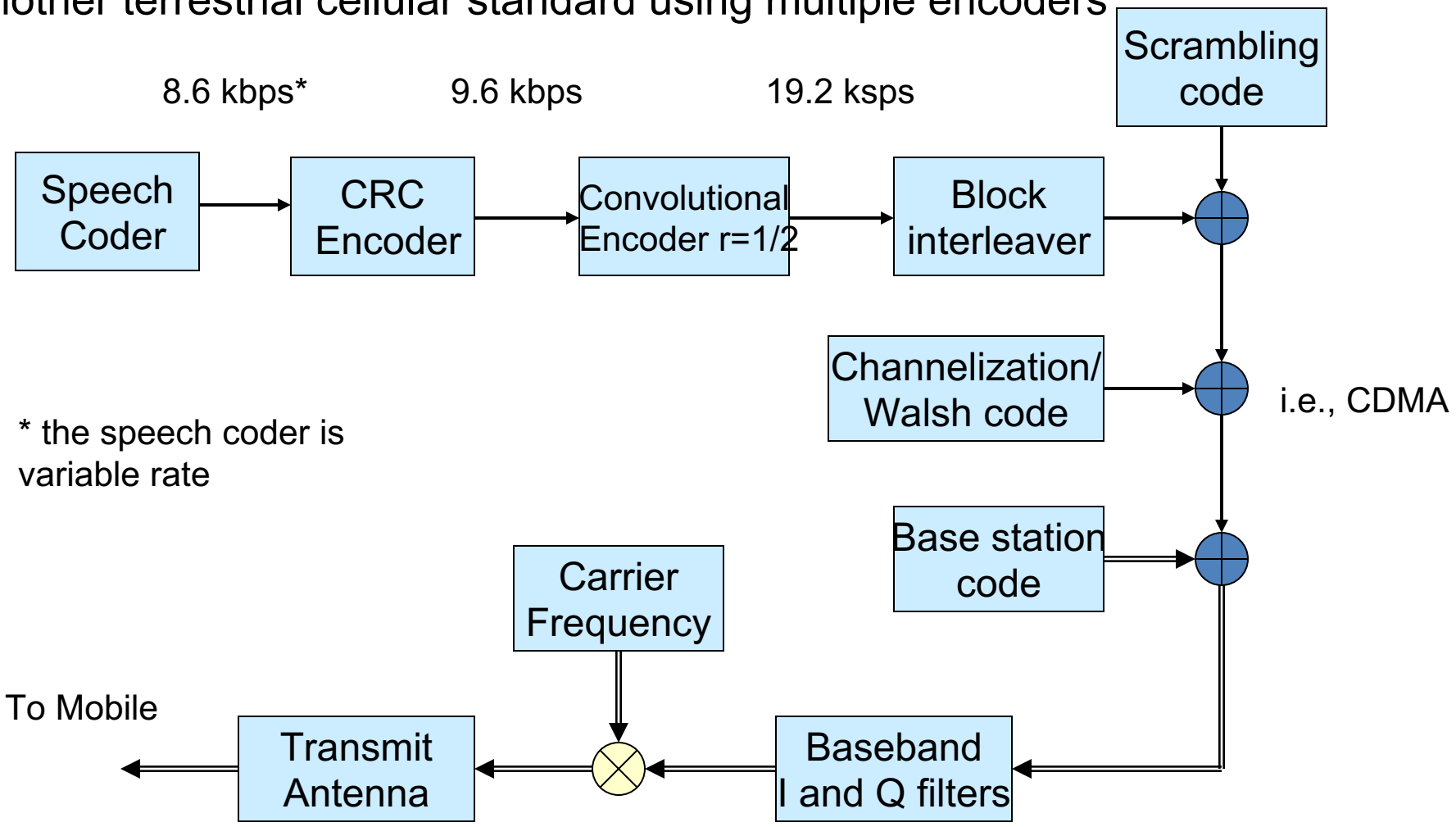
3.1 Example: GSM Forward Traffic Channel

Note the multiple encoders in a terrestrial cellular standard



3.1 Example: IS-95 Forward Traffic Channel

Another terrestrial cellular standard using multiple encoders

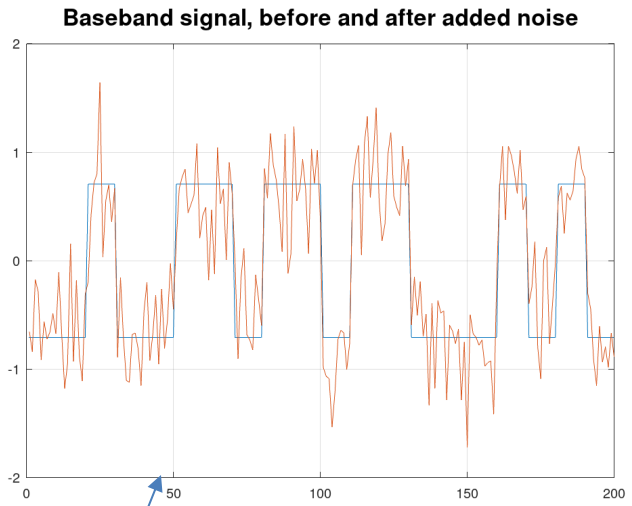


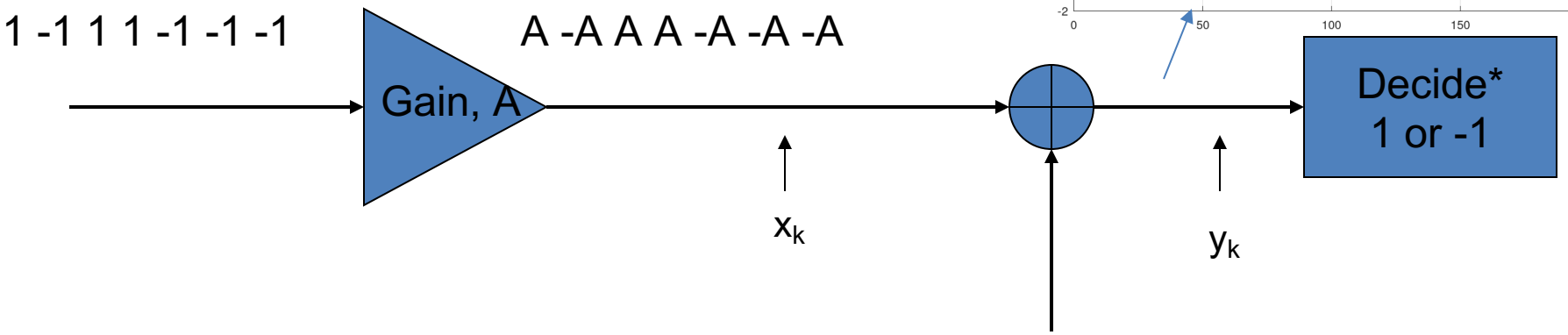
4.0 Detection Theory

In order to better understand how channel coding helps with reliable reception of transmitted signals, we first need to discuss how bit errors can occur.

4.1 Detection Theory - BER

Bit Error Rate for Binary Transmission (e.g., BPSK) with additive white Gaussian noise (AWGN).



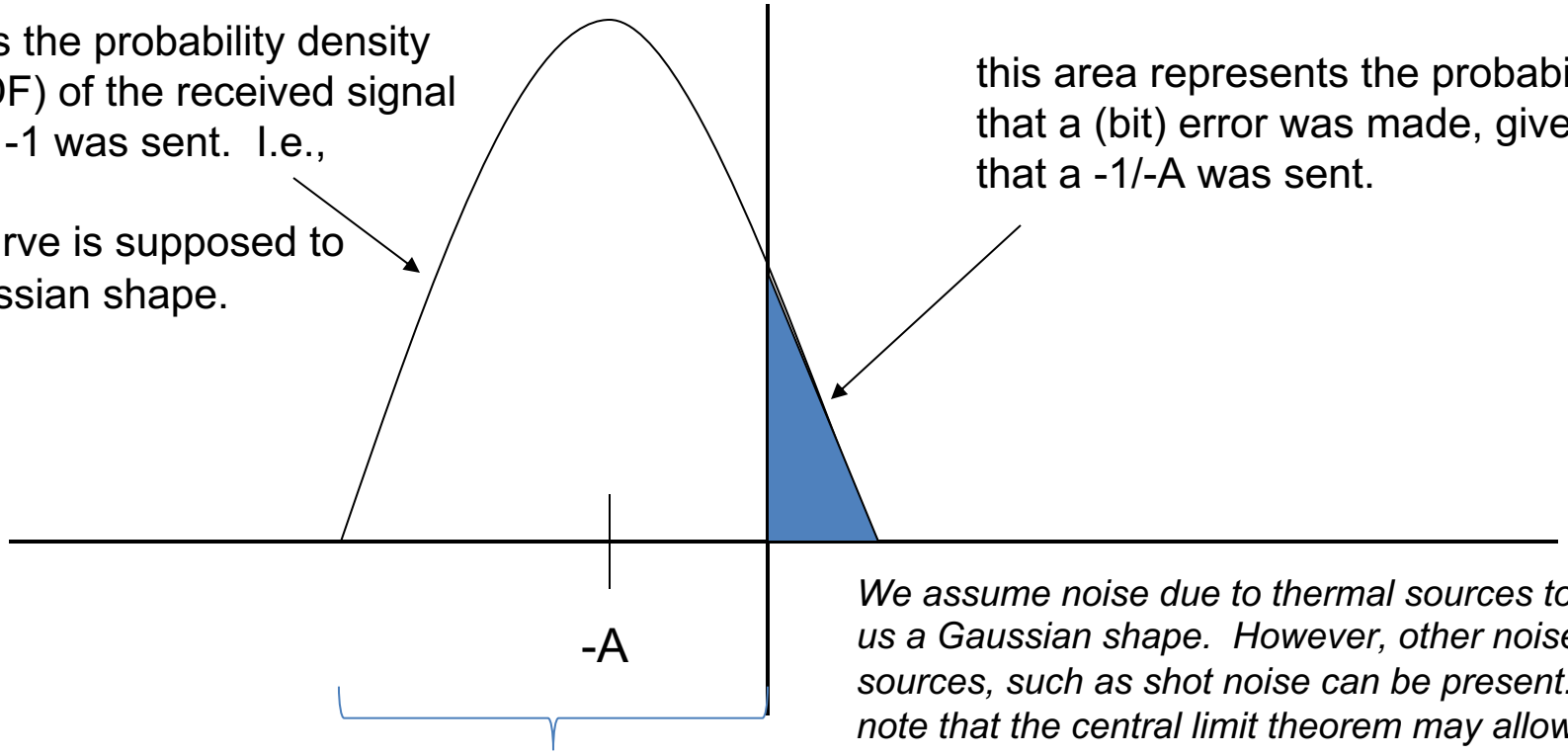


* if $y_k > 0$, 1 was sent, else a zero was sent

4.1 Detection Theory – BER Continued

A theoretical analysis of the probability of a bit error can be obtained by analyzing the distribution of the desired signal + noise

This curve is the probability density function (PDF) of the received signal given that a -1 was sent. I.e., $p(y_k|x_k=-1)$,
Note: the curve is supposed to have a Gaussian shape.



this area represents the probability that a (bit) error was made, given that a -1/-A was sent.

this area represents the probability that a correct decision was made.

We assume noise due to thermal sources to give us a Gaussian shape. However, other noise sources, such as shot noise can be present. Also note that the central limit theorem may allow us to assume a Gaussian PDF.

4.1 Detection Theory – BER: Some Math

$$\mathrm{P}(\text{bit error} \mid \text{-1 was sent}) = \int_0^{\infty} \mathrm{p}(y_k | x_k = -1) dy_k$$

$$\mathrm{P_e}(-1) = \int_0^{\infty} \frac{e^{-(y_k + A)^2 / 2\sigma^2}}{\sigma\sqrt{2\pi}} dy_k, \text{ where } \sigma = \text{noise std dev}$$

$$\mathrm{P_e}(\text{-1}) = \frac{1}{\sqrt{2\pi}} \int_{\frac{A}{\sigma}}^{\infty} e^{-z^2/2} dz, \text{ by change of variable}$$

$$\mathrm{P_e}(\text{-1}) = \mathrm{Q}\left(\frac{\mathrm{A}}{\sigma}\right)$$

Note: The Q() function can be looked up in mathematical tables, or by using qfunc() in Matlab, also note that the PDF for a normal distribution is given by,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

4.1 Detection Theory – BER: Probability of all errors

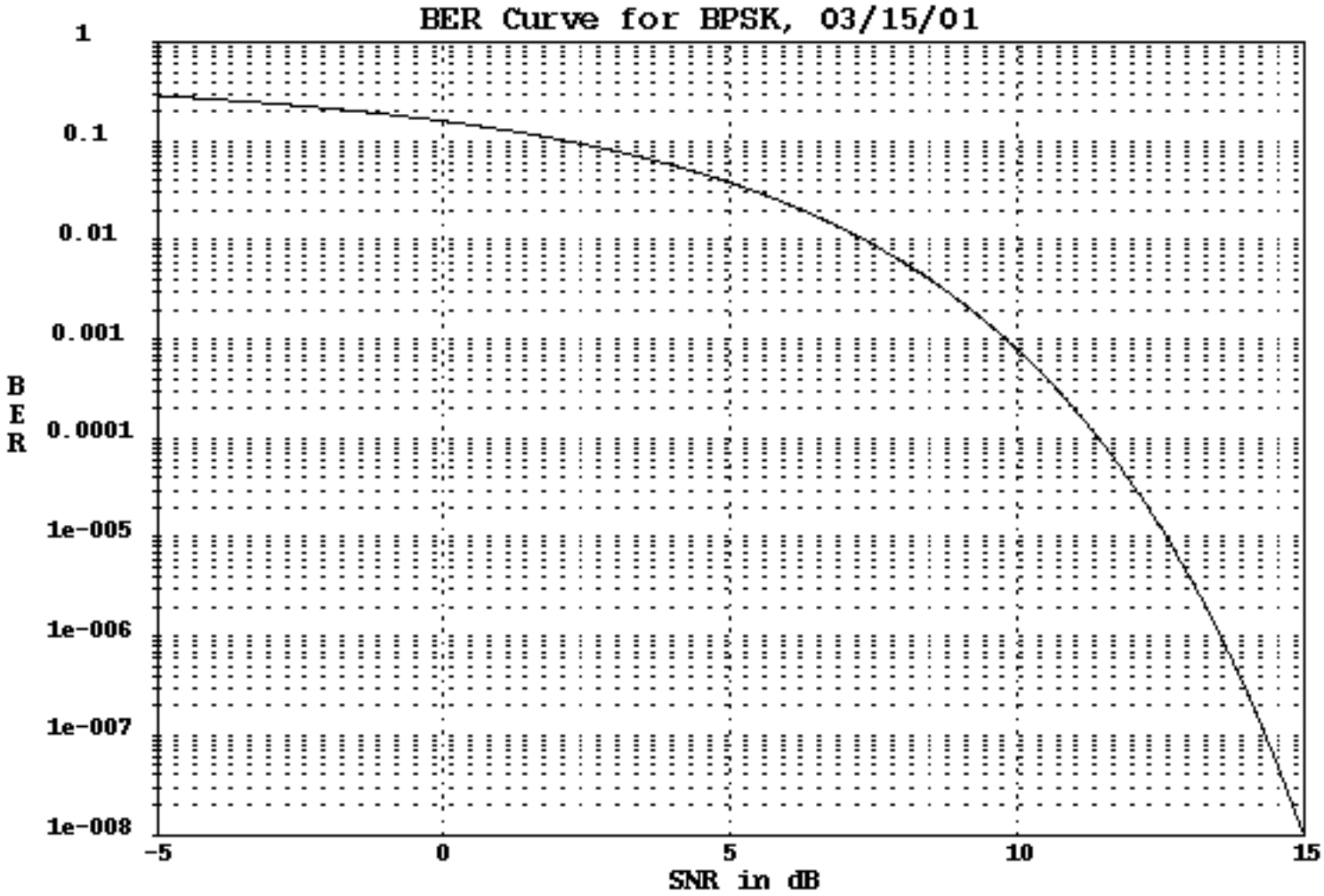
$$P_e = P(\text{-1 sent}) * P_e(-1) + P(1 \text{ sent}) * P_e(1)$$

$$P_e = \frac{1}{2} Q\left(\frac{A}{\sigma}\right) + \frac{1}{2} Q\left(\frac{A}{\sigma}\right)$$

$$P_e = Q\left(\frac{A}{\sigma}\right)$$

This formula gives the probability of a bit error in terms of the signal to noise ratio: A/$\boldsymbol{\sigma}$, where A is the signal amplitude and $\boldsymbol{\sigma}$, is a measure of the noise power.

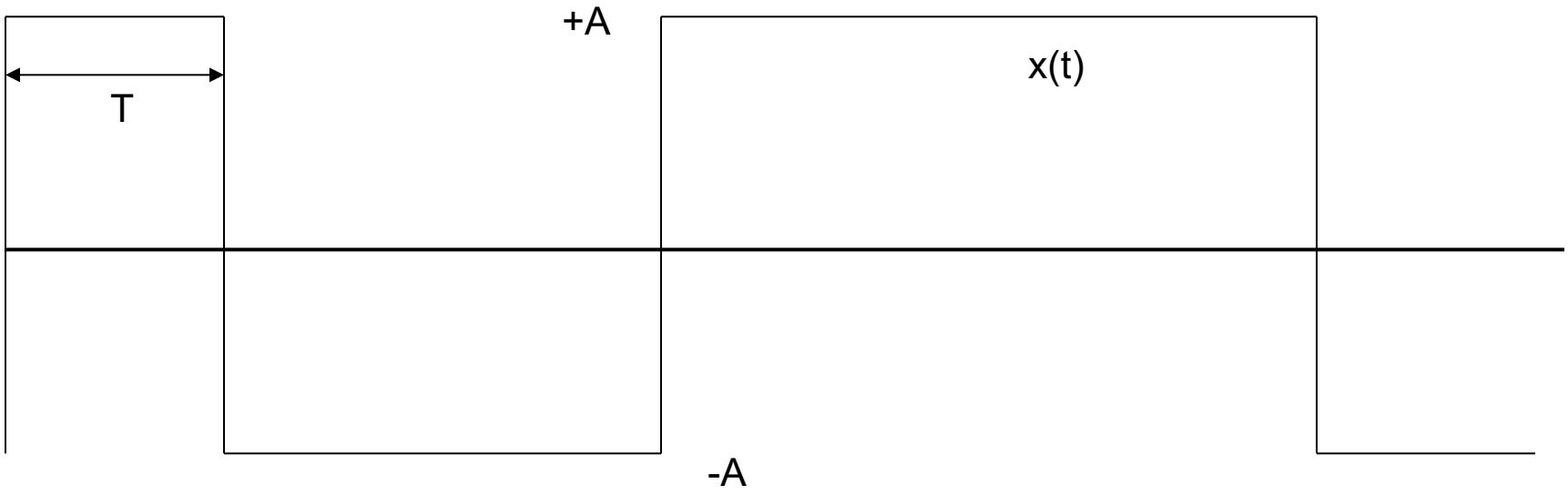
4.1 Detection Theory – BER Plot



We can plot the theoretical bit error rate (BER) as a function of the signal-to-noise ratio using the formula on the previous slide. Note that at a BER of 1e-8, we would only see a bit error every 100 million bits. If we apply channel coding, then we may never see a bit error at this SNR (15 dB).

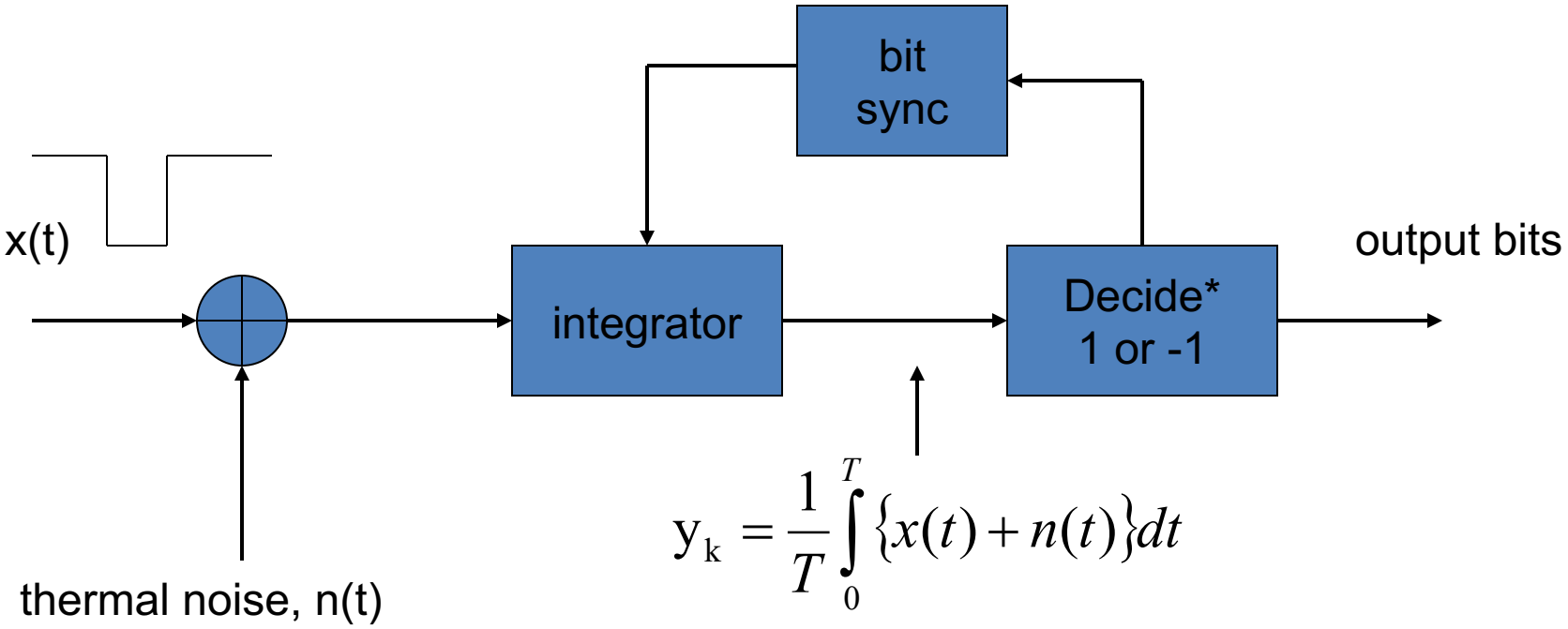
4.2 Detection Theory – Baseband Signal

The BER curves are often expressed in terms of Energy per bit/Noise density (a type of SNR). First, we consider a more realistic transmitted signal. This signal is shown at baseband, that is, before modulation takes place.



4.2 Detection Theory – Reception of Baseband Signal

The purpose of a digital receiver is to determine what bit(s) was sent in the presence of channel degradations (e.g., noise).



Since the receiver doesn't have knowledge of the transmitter timing (i.e., bit start and stop timings), it must use a bit sync to determine this timing. The integrator just sums the samples over the bit period.

4.2 Detection Theory – BER Calculation

BER – E_b/N_0

Assuming a -1 was sent, the kth output signal is,

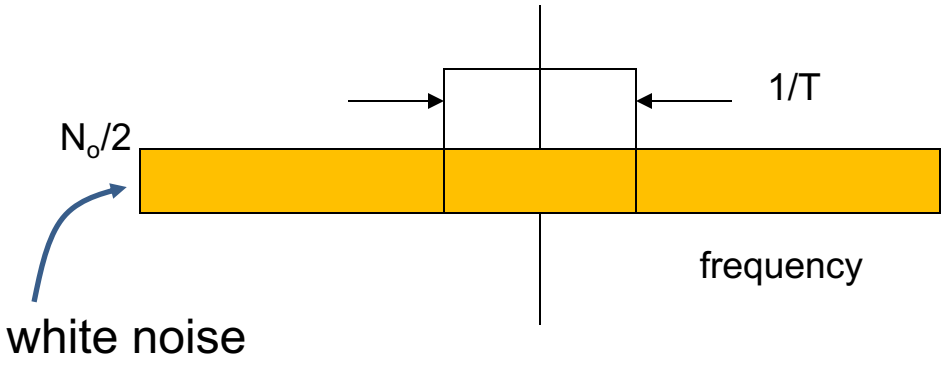
$$y_k = \frac{1}{T}\int_0^T x(t)dt + \frac{1}{T}\int_0^T n(t)dt = -A + n_k$$

The noise variance (power) is given by,

$$\sigma_y^2 = \frac{1}{T^2} E\left\{\int_0^T \int_0^T n(t)n(s)dtds\right\}$$

$$\sigma_y^2 = \frac{N_o}{2T}$$

Using the Wiener-Khintchine theorem:
S(ω) <-> R(τ)



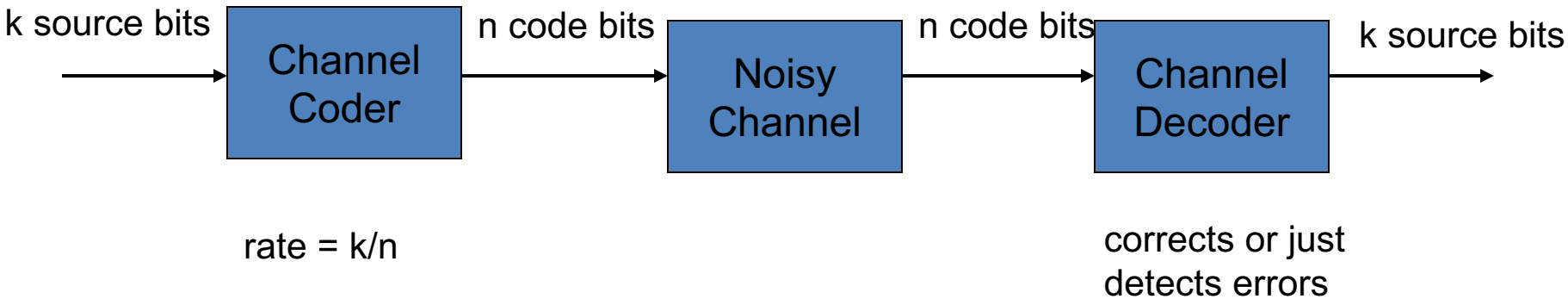
4.3 Channel Coding – Overview

The following topics will be covered in this section:

- Channel coding: Block diagram
- Channel coding: Types of codes
- Channel coding: Code properties
- Channel coding: Coding gain
- Channel coding: Complexity
- Channel coding: Linear block codes
- Channel coding: Linear cyclic codes
- Channel coding: CRC codes
- Channel coding: Convolutional codes
- Channel coding: Interleaving
- Channel coding: Turbo codes
- Channel coding: LDPC codes

4.3 Channel Coding – Block Diagram

Goal: Add parity bits to an input data stream in order to improve the BER performance (at the expense of increased complexity).



4.3 Channel Coding – Types of Codes

Types of Codes

- Linear Block Codes
  - Hamming Codes
  - Cyclic Codes
    - CRC
    - Golay
    - BCH
    - Reed-Solomon
  - Hamming Codes
- Convolutional Codes
- Turbo Codes
- LDPC Codes

4.3.1 Channel Coding – Code Properties

Code Properties:

- Code Rate: information bits/code bits = k/n [(n,k) code]
- Minimum distance: minimum distance between code words
- Error detection capabilities
- Error correction capabilities
- Coding gain
- Complexity (MIPS)

Code Properties - Minimum Distance

The minimum distance, d_{min}, is the minimum Hamming distance between all code words.

Example,

c0 = 1 0 1 0 1 0 1 0 1 0

c1 = 1 1 1 0 1 0 1 0 1 0

c2 = 0 1 0 1 0 1 0 1 0 1

Then minimum distance = 1, between c0 and c1. Obviously, this is a very weak code, since a single bit error could change one code word into another.

For an (n,k) linear block code the Singleton bound states,

$d_{min} <= n - k + 1$, for example Reed-Solomon codes achieve bound.

Recall that k = # info bits, n = #info bits + #parity bits

4.3.1 Channel Coding – Code Properties

Code Properties - Error Detection

Error detection: for a linear block code we can detect all error patterns of weight d_{min} - 1 or less

Error detection coverage: In some environments, the code word can be so badly corrupted, that we want to ensure that a random corrupted word does not look like a code word. The coverage is,

$$\lambda = \frac{\text{invalid codes}}{\text{all codes}} = \frac{2^n - 2^k}{2^n} = 1 - 2^{k-n}$$ k=32, n=33

for example, CRC-32, λ = 0.5

Burst error detection: In some environments, errors come in bursts, therefore, we want the code to perform well in the face of burst errors. For example, a cyclic code is able to detect all burst error patterns of length, r = (n-k), or less.

4.3.1 Channel Coding – Code Properties

Code Properties - Error Detection (Burst)

Burst Error detection: for a linear cyclic code,

A cyclic code with generator polynomial, g(x) of degree r, can detect $1 - 2^{1-r}$ of all burst error patterns of length $(r+1)$

In addition, the code can detect the fraction, $1 - 2^{-r}$ of all burst error patterns of length $b > (r+1)$.

4.3.1 Channel Coding – Code Properties

Code Properties - Error Detection

Example, IS-95 forward link at 9 600 bps

CRC code: n = 184, k = 172, r = 12

$\lambda = 1 - 2^{-12} = 0.999756$

All single bursts of length 12 or less can be detected.

Detects $1 - 2^{1-12} = 99.95\%$ of single bursts of length 13

Detects $1 - \frac{1}{2^{12}}$ = 99.976% of all error bursts of length > 13

That is, it detects 99.976% of all length 14 burst errors, similarly for length 15 burst errors.

4.3.1 Channel Coding – Code Properties

Code Properties - Error Correction

Error correction: for a linear block code there is a number, t, such that,

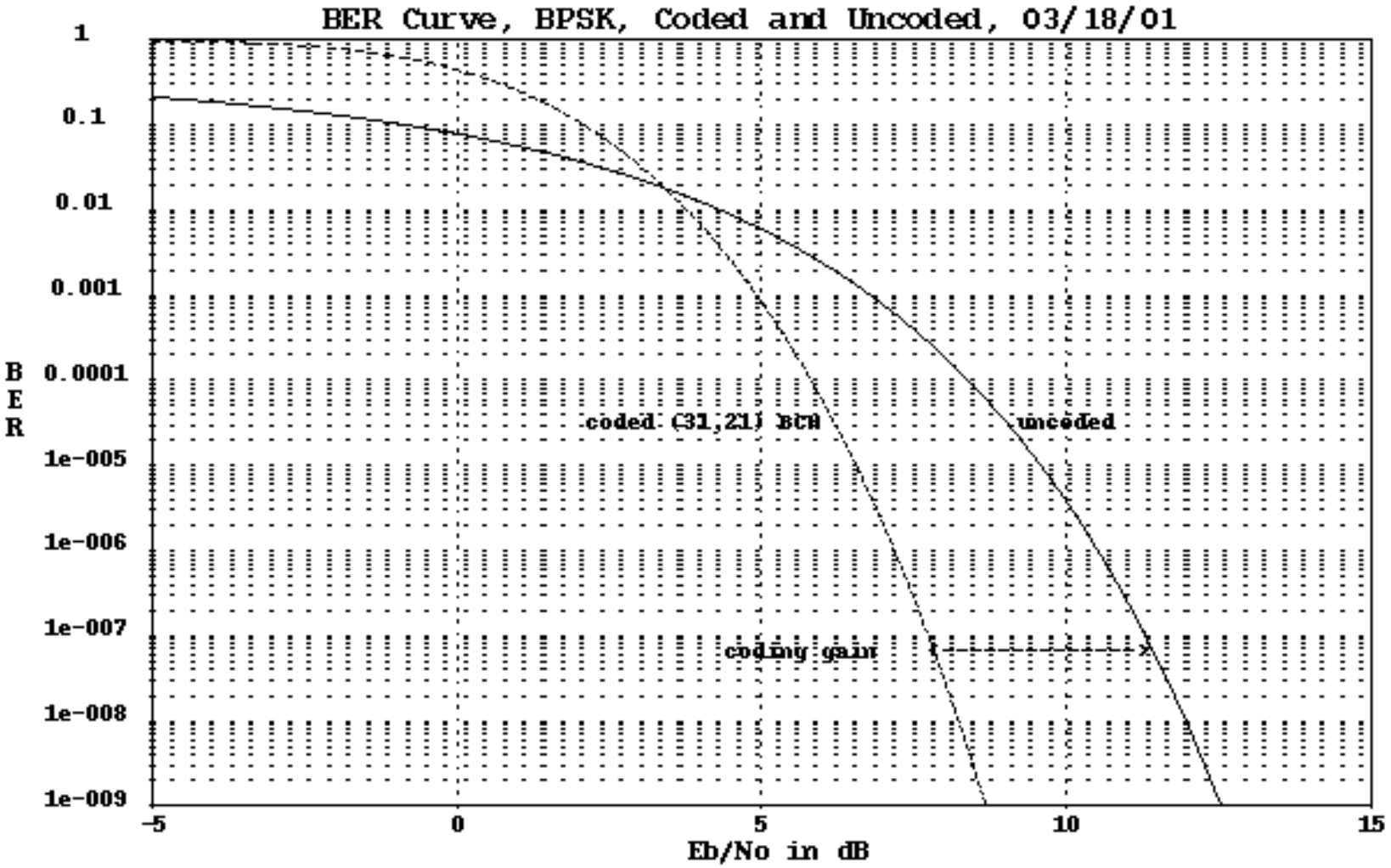
$2t+1 <= d_{min} <= 2t+2$,

Then the code is capable of correcting t or fewer errors. Recall that d_{min} is the minimum Hamming distance of the code. An equivalent way of stating this result is that a code with minimum distance, d_{min} can correct all error patterns having, t, or fewer errors, with t given by,

$t = \text{floor}[(d_{min}-1)/2]$

4.3.2 Channel Coding – Coding Gain

Code Properties - Coding Gain (note: E_b=E/R, where R = code rate, E = total energy, E_b = energy/bit).



4.3.2 Channel Coding – Coding Gain Comparison

Code Properties - Coding Gain Compared to BPSK

Coding	Gain at 10^{-5}	Gain at 10^{-8}	Data Rate
Ideal (Shannon)	11.2 dB	13.6 dB	N/A
Turbo decoding	10.8 dB	13.2 dB?	Moderate
Reed-Solomon and convolutional (Viterbi)	6.5-7.5	8.5-9.5	Moderate
Convolutional with soft Viterbi	6.0-7.0	8.0-9.0	Moderate
Block codes (soft)	5.0-6.0	6.5-7.5	Moderate
Reed-Solomon and short block	4.5-5.5	6.5-7.5	Very high
Convolutional with hard Viterbi	4.0-5.5	5.0-6.5	High
Block Codes (hard)	3.0-4.0	4.5-5.5	High

4.3.3 Channel Coding – Code Properties: Complexity

Code Properties – Complexity:

The complexity of coding and/or decoding may be a function of hardware versus software implementation. In addition, the complexity of a code may determine whether or not it can be implemented in real time.

4.3.4 Channel Coding – Linear Block Codes

Linear Block Codes

A code word can be generated through the use of a generator matrix, as,

c = mG, where m = message bits, and G is given by,

$$G = \begin{bmatrix} g_o \\ \dots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ & \dots & \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

that is, each code word is given as a linear combination of a set of basis code word vectors.

Corresponding to G is a parity check matrix, H, such that for any code word, c, $cH^T = 0$. This parity check can then be used to detect and correct errors.

Linear Cyclic Codes [WIC95]

Linear cyclic codes are block codes with the additional property that every code is a cyclic shift of another code. This property leads to efficient hardware implementations for the encoder and decoder.

Because of the cyclic nature of the codes, we can associate a polynomial with each code word, where multiplication of polynomials corresponds to cyclic shifts.

Using this polynomial representation, each cyclic code can be associated with a generator polynomial {g(x)}, such that the encoder can be implemented as follows (with c(x) the code word to be transmitted):

- Multiply the message polynomial, m(x) by x^{n-k} (note: n = codeword length, k = info bits length)
- Divide the result by g(x), with d(x) the remainder
- Set $c(x) = x^{n-k}m(x) - d(x)$

4.3.5 Channel Coding – Linear Cyclic Codes

Linear Cyclic Codes [WIC95]

Example: (7,3) code with $g(x) = x^4 + x^3 + x^2 + 1$, message = (101) -> $1 + 0x + x^2$:

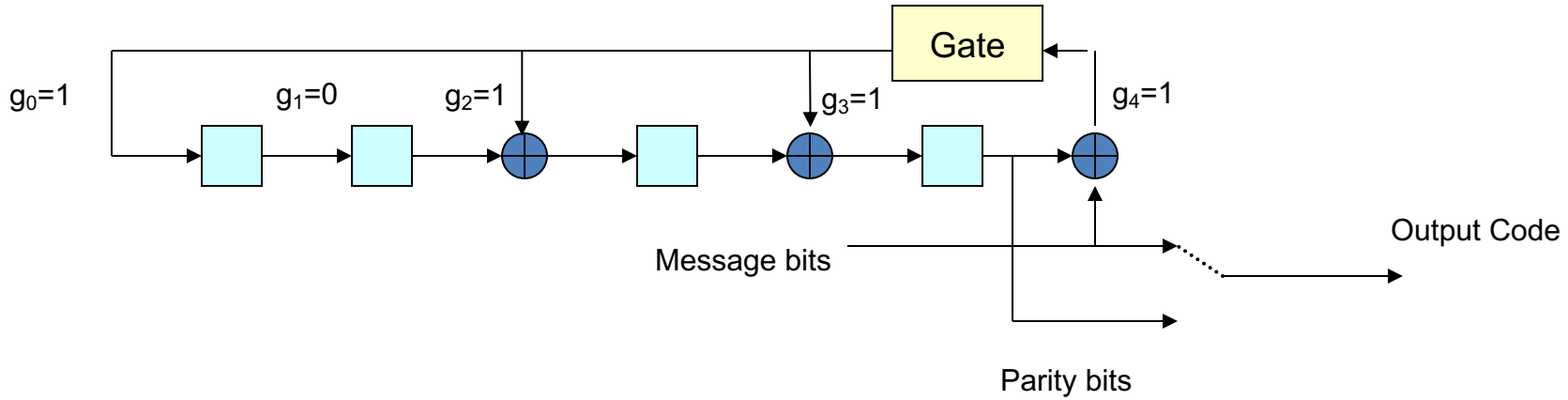
$x^{n-k}m(x) = x^4(x^2+1) = x^6+x^4$

$(x^6 + x^4) / (x^4 + x^3 + x^2 + 1) = x^2 + x + 1$ with $d(x) = x + 1$ (use polynomial division)

$c_m(x) = x^{n-k}m(x) - d(x) = x^6+x^4 - d(x) = 1 + x + x^4 + x^6$ -> (1100101)

(note: operations are over GF(2))

This polynomial representation leads to an efficient hardware implementation using shift registers:

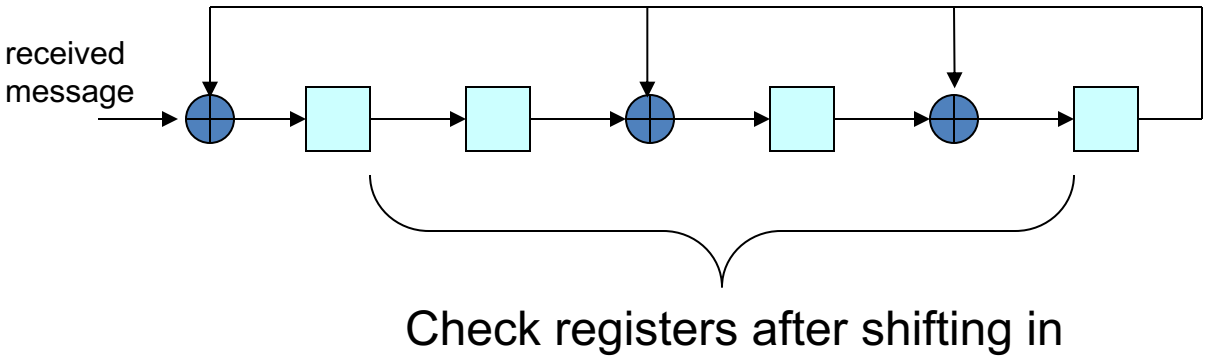


4.3.5 Channel Coding – Linear Cyclic Codes

Linear Cyclic Codes [WIC95]

Error detection with cyclic codes can also be accomplished using polynomial representation.

Again, this polynomial representation leads to an efficient hardware implementation using shift registers. After shifting the message bits in, check the contents of the shift registers, this will be the *syndrome*, and will be zero if the message has no errors.



4.3.5 Channel Coding – Linear Cyclic Codes

Linear Cyclic Codes [WIC95]

Error correction with cyclic codes is more complex than detection (for detection we just check for a non-zero syndrome), with the complexity increasing exponentially with code length and the number of errors to be corrected.

Accordingly, subclasses of cyclic codes are used in error correction, which have properties which simplify the decoding operation.

4.3.6 Channel Coding – CRC Codes

Cyclic Redundancy Check (CRC) Codes

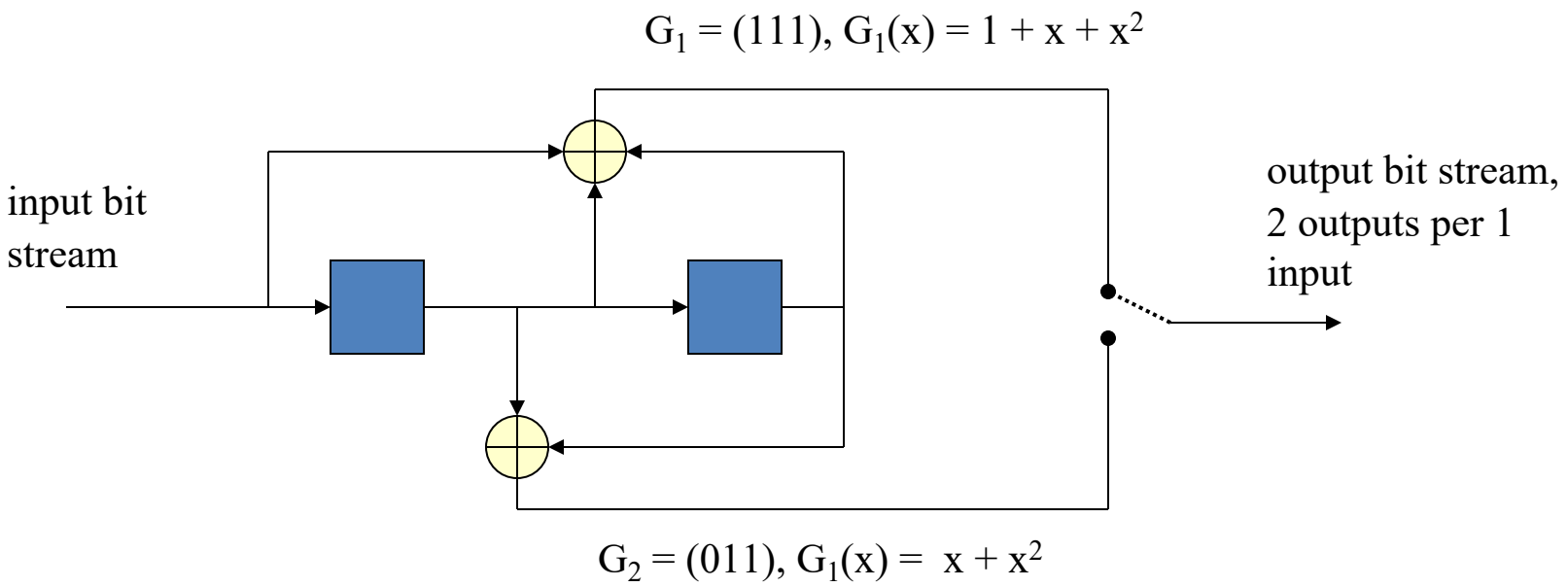
- Cyclic Redundancy check (CRC) codes are shortened cyclic codes, such that (n,k) -> (n-j, k-j). Since the error detection capability of a code is based on d_{min} <= n-k-1, the shortened codes will have detection properties at least as good as the unshortened code (recall that n = code length, k = info bits length).
- The shortened codes will also have encoder and detection circuits that are similar to the unshortened codes. That is, efficient shift register implementations can be used.
- Since these codes have good coverage (many invalid words), these codes are good for detecting high bit error situations. For example, they are used as the outer code for some wireless standards.
- These codes have been developed in order to meet users' requirements on code (n) or information (k) lengths.

Convolutional Codes

In contrast to block coders, convolutional codes encode an entire data stream into a single code word.

This coding is obtained using shift registers.

Example: Rate 1/2, K=3 convolutional coder



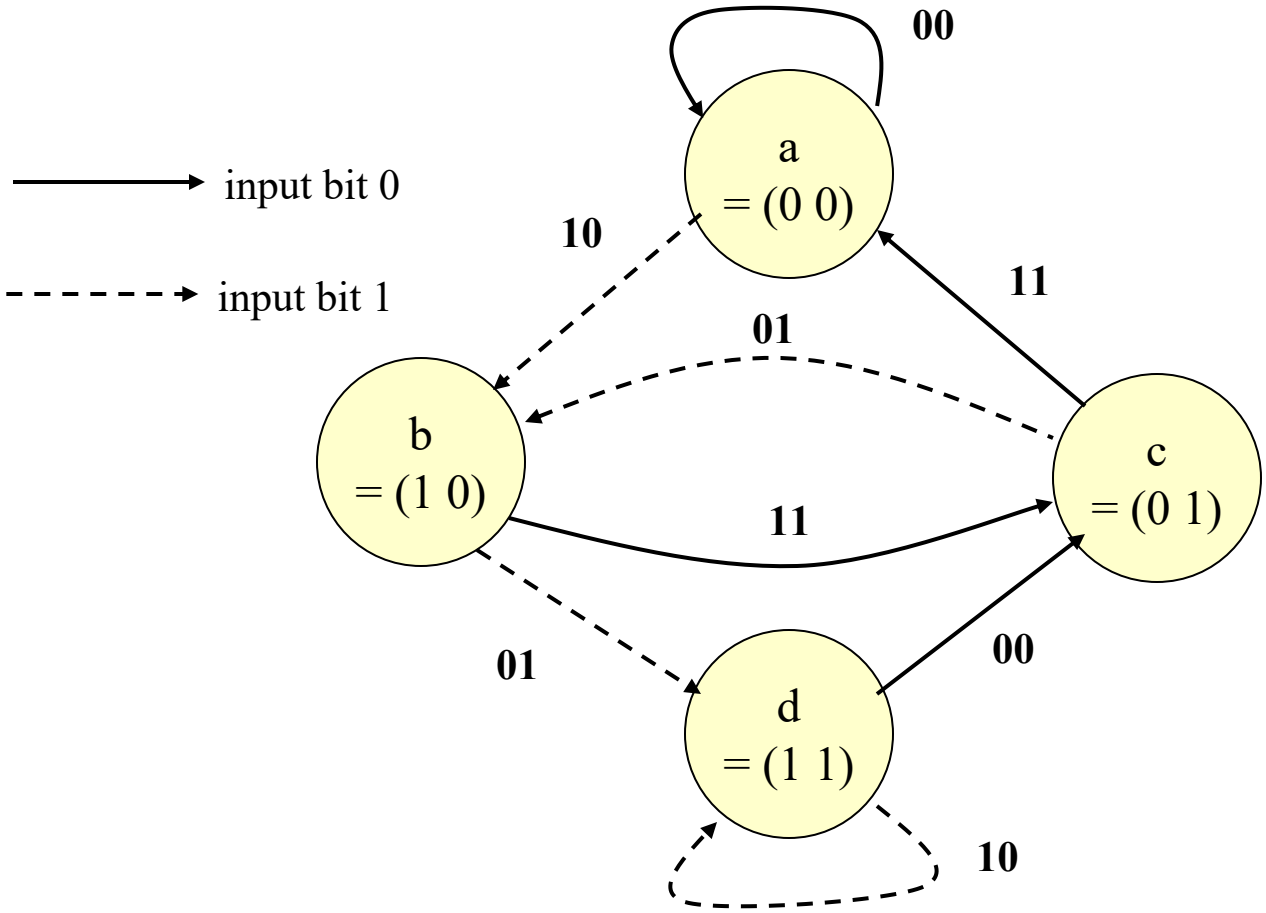
4.3.7 Channel Coding – Convolutional Codes

Convolutional Codes - Properties

- Constraint Length: The maximum number of bits in a single output stream that can be affected by any input bit. In most cases the constraint length is equal to the number of shift register stages + 1, although some authors use a different definition.
- Efficient decoding of convolutional codes can be obtained using maximum likelihood soft decision decoding.
- As opposed to block codes, the block length of convolutional codes can be adjustable.
- Convolutional codes come fairly close to the Shannon limit, only exceeded by turbo codes and LDPC (low-density parity check).

Convolutional Codes - Representations

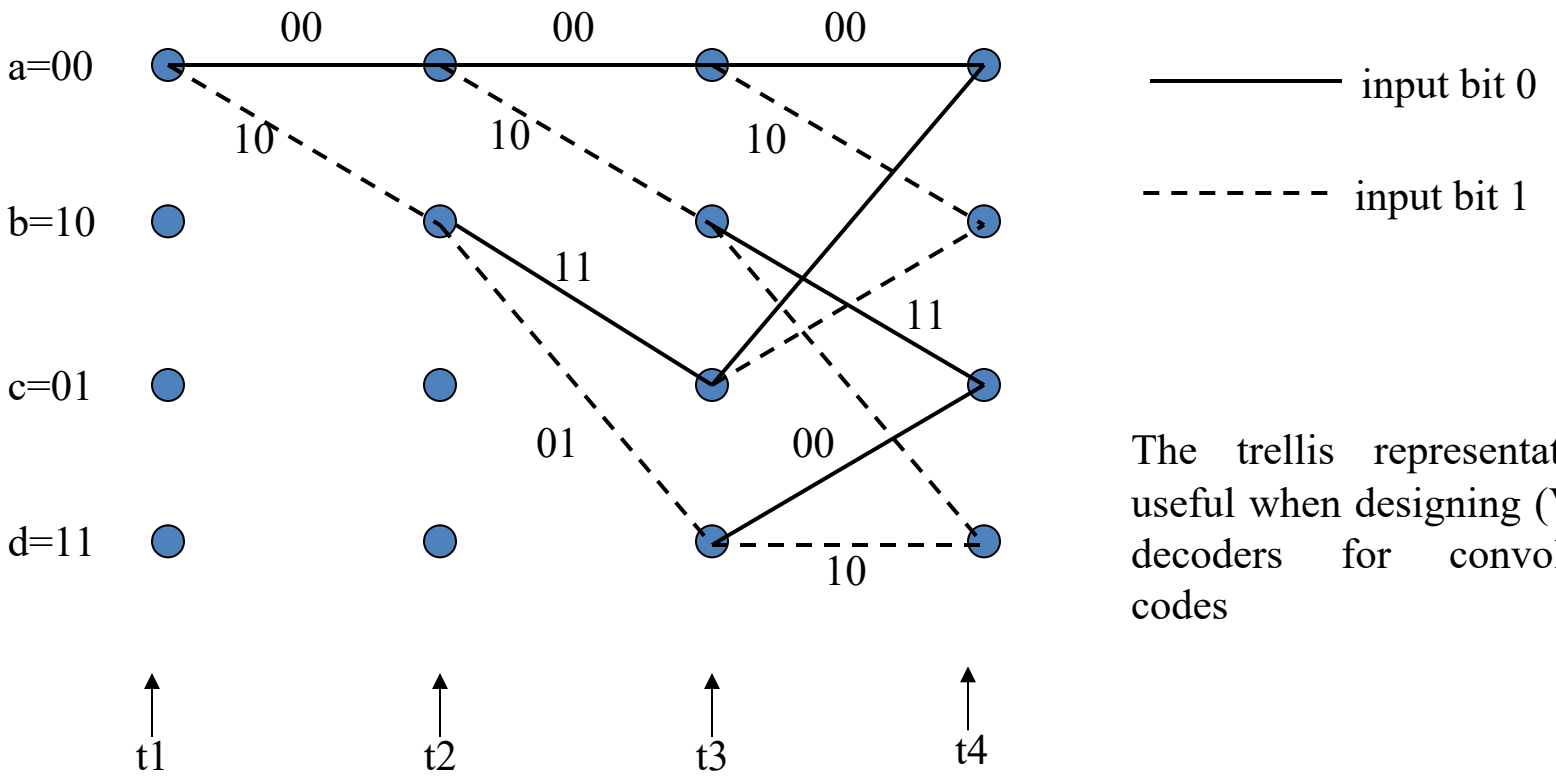
State Representation [LEE98]: The states correspond to shift register contents



Example for 2 stage shift register

Convolutional Codes - Representations

Trellis Representation [LEE98]: The states correspond to shift register contents



The trellis representation is useful when designing (Viterbi) decoders for convolutional codes

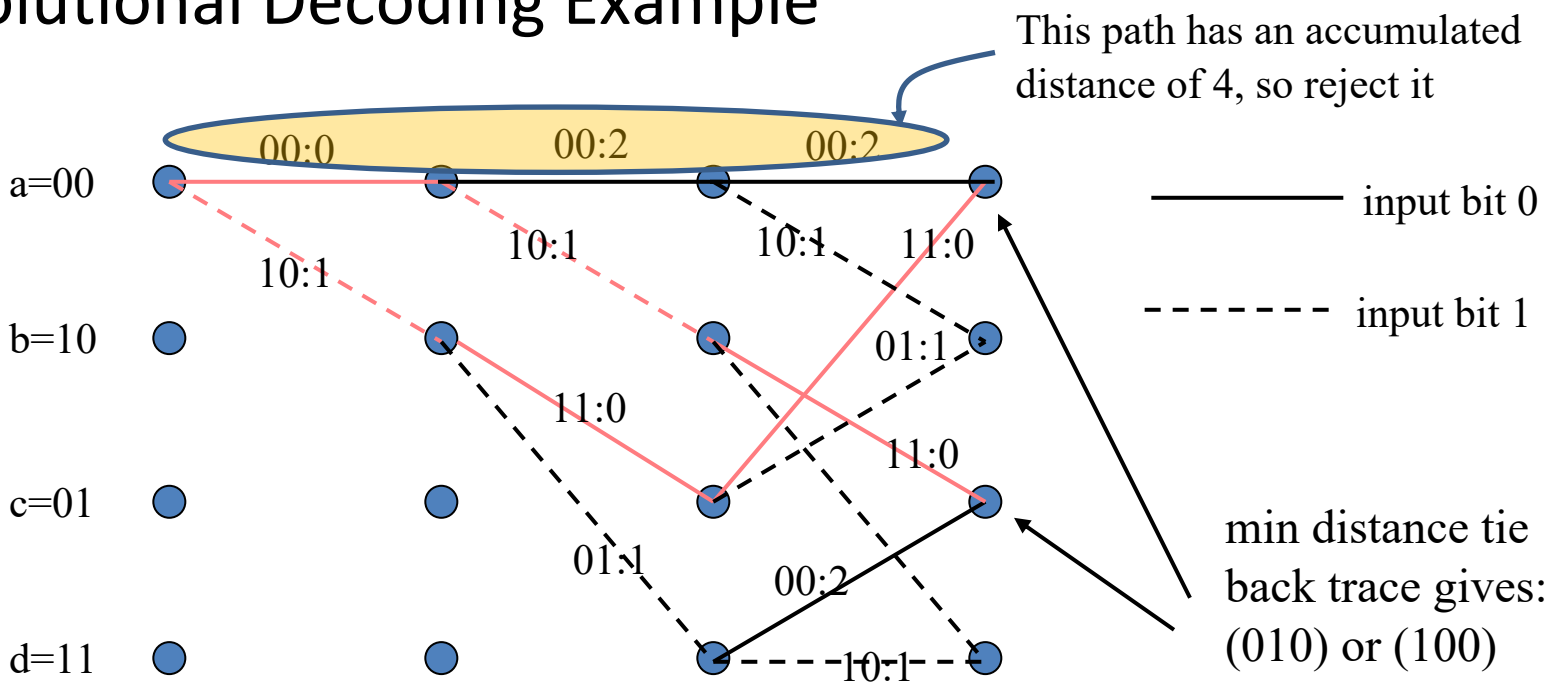
Convolutional Codes - Decoding (Viterbi)

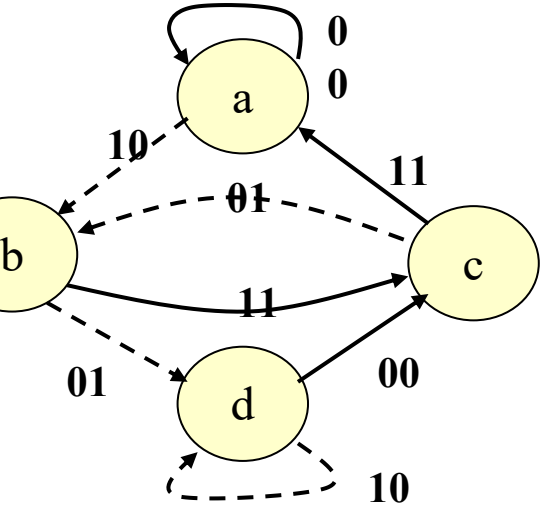
The Viterbi decoder makes us of the trellis diagram (shown on the previous slide) in order to implement the decoding using a minimum distance criterion.

The minimum distance decoding works as follows:

- At each time step, for each state compare the input bits with the state transition output.
- The incremental distance is the difference between the input bits and the state transition output. This distance can be “hard” or “soft”. In hard decoding, the input bits are quantized to 0s and 1s. In soft decoding, a Euclidean distance is taken between the input bits, and the state transition output.
- After all the bits have been processed, and the incremental distances have been accumulated, the state with the smallest accumulated distance is selected, and its “backtrace” gives the decoded bits.

Convolutional Decoding Example





input	xmit	rcvd	d_a	d_b	d_c	d_d
0	00	00	0	1	N/A	N/A
1	10	11	2	1	1	2
0	11	11	1	2	1	3

Note that we have a tie, and soft decisions may have helped to break the tie.

4.3.7 Channel Coding – Convolutional Codes

Convolutional Codes - Decoding Issues

- The (back trace) memory requirements grow with the number of states.
- The number of states is equal to $2^{constraint\_length-1}$, with a longer constraint length giving better performance.
- For implementation on a DSP core, it's nice to have a core which includes a compare-select-store operation.
- In practice, it is sometimes good to limit the back trace length. A rule of thumb is to set the decode depth to 5 to 10 times the constraint length.

4.3.7 Channel Coding – Convolutional Codes

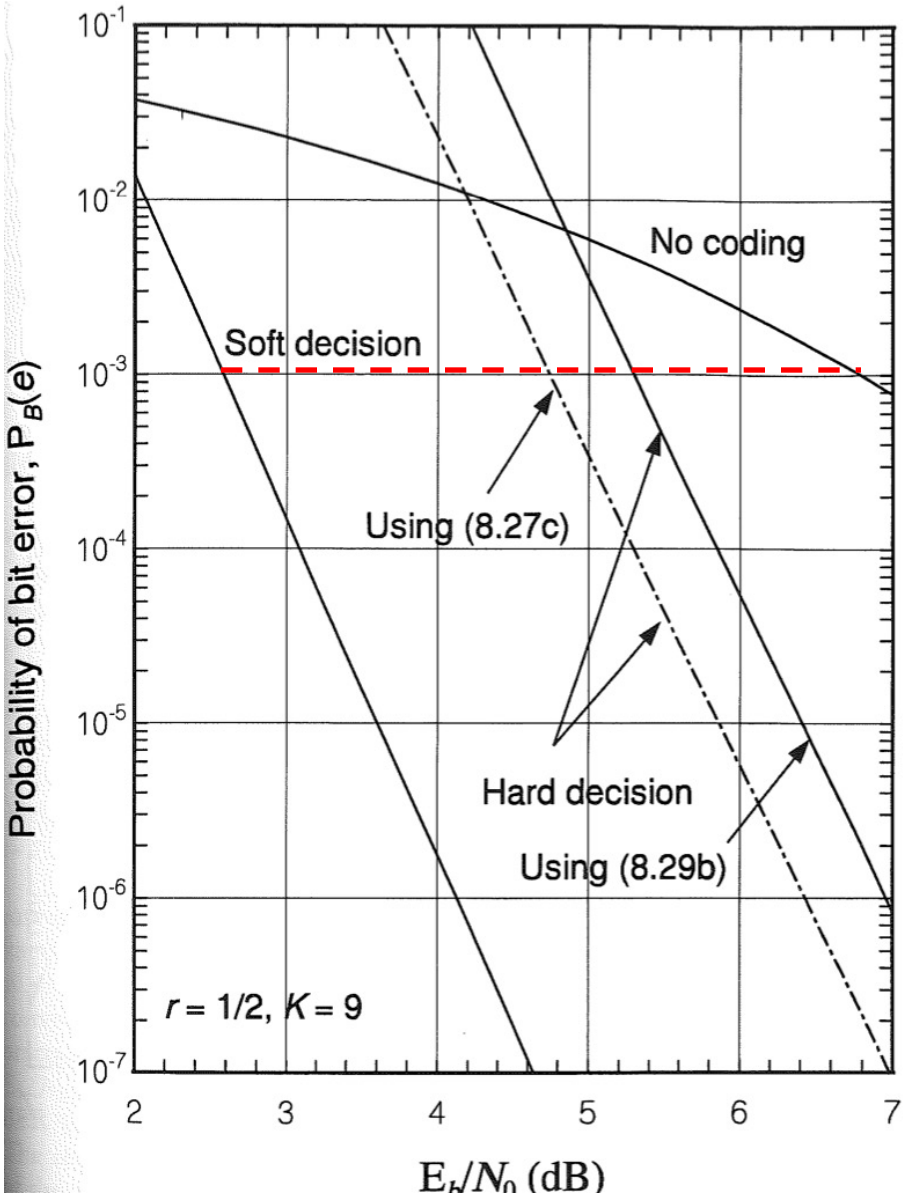
Convolutional Codes - Performance

- The performance of a convolutional code is a function of the minimum free distance of the code.
- This minimum free distance can sometimes only be found through an exhaustive search.
- Bounds on the coding gain are given by [LEE98] (r = code rate, d_{free} = free distance).

$$10 * \log_{10}\left(\frac{rd_{free}}{2}\right) \leq \text{coding gain} \leq 10 * \log_{10}\left(rd_{free}\right)$$

4.3.7 Channel Coding – Convolutional Codes

Coding gain plot [LEE98]
rate ½ code
gain ~ 4 dB at BER = 10^{-3}

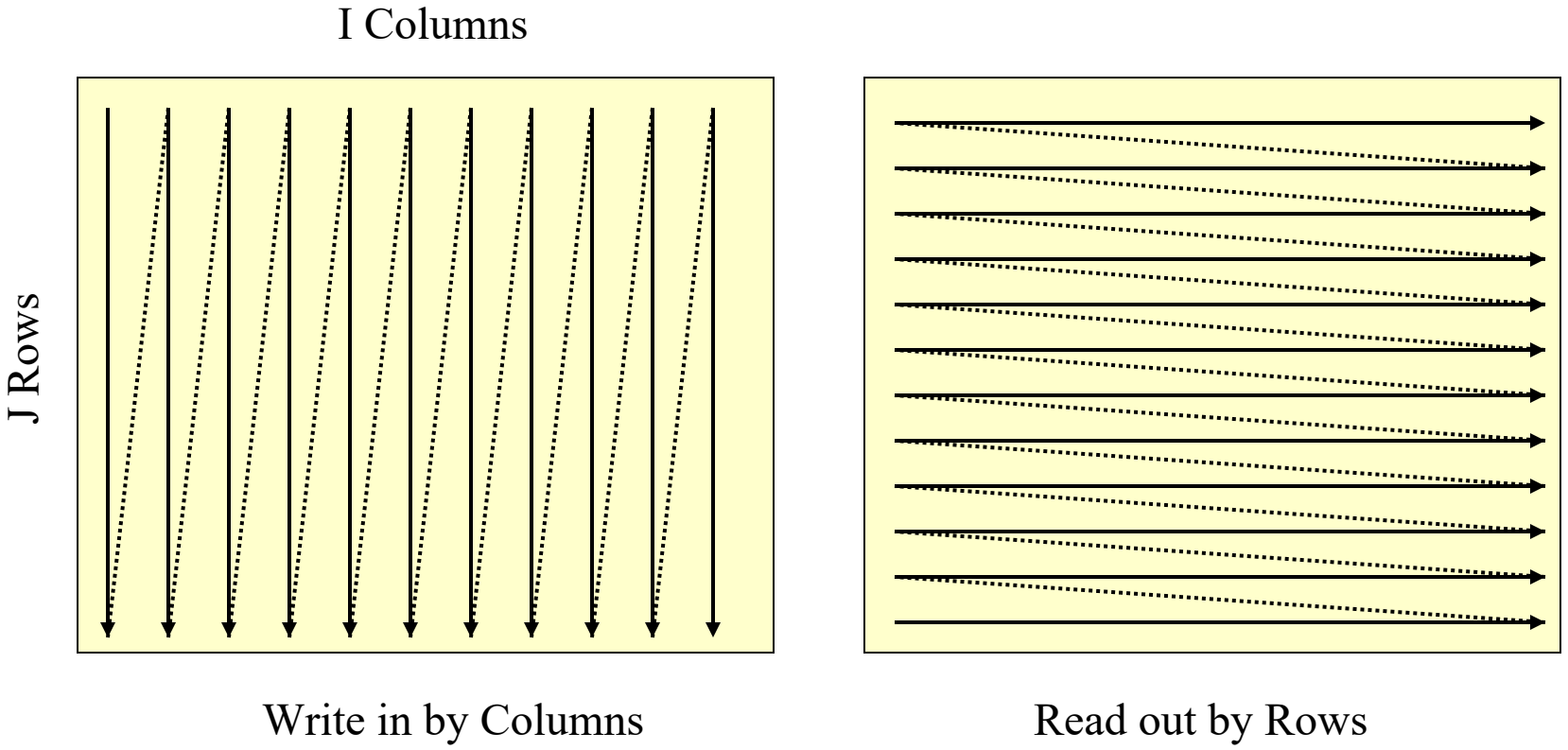


4.3.8 Channel Coding – Interleaving

Interleaving [LEE98]

- Although not a type of channel coding, interleaving is often employed with coding to spread out burst type of error conditions that occur in wireless channels (i.e., fading).
- By spreading out the errors, codes that can't handle burst conditions well (e.g., convolutional codes) achieve better performance.
- In addition, interleaving is used in turbo codes, which are discussed in the next section.
- There are two types of interleaving: block and convolutional. Block is easier to implement, but convolutional has better performance.

Interleaving [LEE98]: Matrix Interleaver



4.3.9 Channel Coding – Turbo Codes

turbo codes (originally in French *Turbocodes*) are a class of high-performance forward error correction (FEC) codes developed around 1990–91, but first published in 1993. They were the first practical codes to closely approach the maximum channel capacity or Shannon limit, a theoretical maximum for the code rate at which reliable communication is still possible given a specific noise level. Turbo codes are used in 3G/4G mobile communications (e.g., in UMTS and LTE) and in (deep space) satellite communications as well as other applications where designers seek to achieve reliable information transfer over bandwidth- or latency-constrained communication links in the presence of data-corrupting noise. Turbo codes compete with low-density parity-check (LDPC) codes, which provide similar performance. The name "turbo code" arose from the feedback loop used during normal turbo code decoding, which was analogized to the exhaust feedback used for engine turbocharging. Hagenauer has argued the term turbo code is a misnomer since there is no feedback involved in the encoding process.

From wikipedia

Claude Berrou was born in Penmarc'h, France, in 1951. In 1978, he joined the Ecole Nationale Superieure des Telecommunications (ENST) de Bretagne (now named Telecom Bretagne), where he is currently a Professor in the Electronics Department. In the early 80's, he initiated the training and research activities in VLSI technology and design to meet the growing demand from industry for microelectronics engineers. Some years later, Prof. Berrou took an active interest in the field of algorithm/silicon interaction for digital communications. In collaboration with Prof. Alain Glavieux, he introduced the concept of probabilistic feedback into error correcting decoders and developed a new family of quasi-optimal error correction codes that he nicknamed turbo codes.

4.3.9 Channel Coding – Turbo Codes

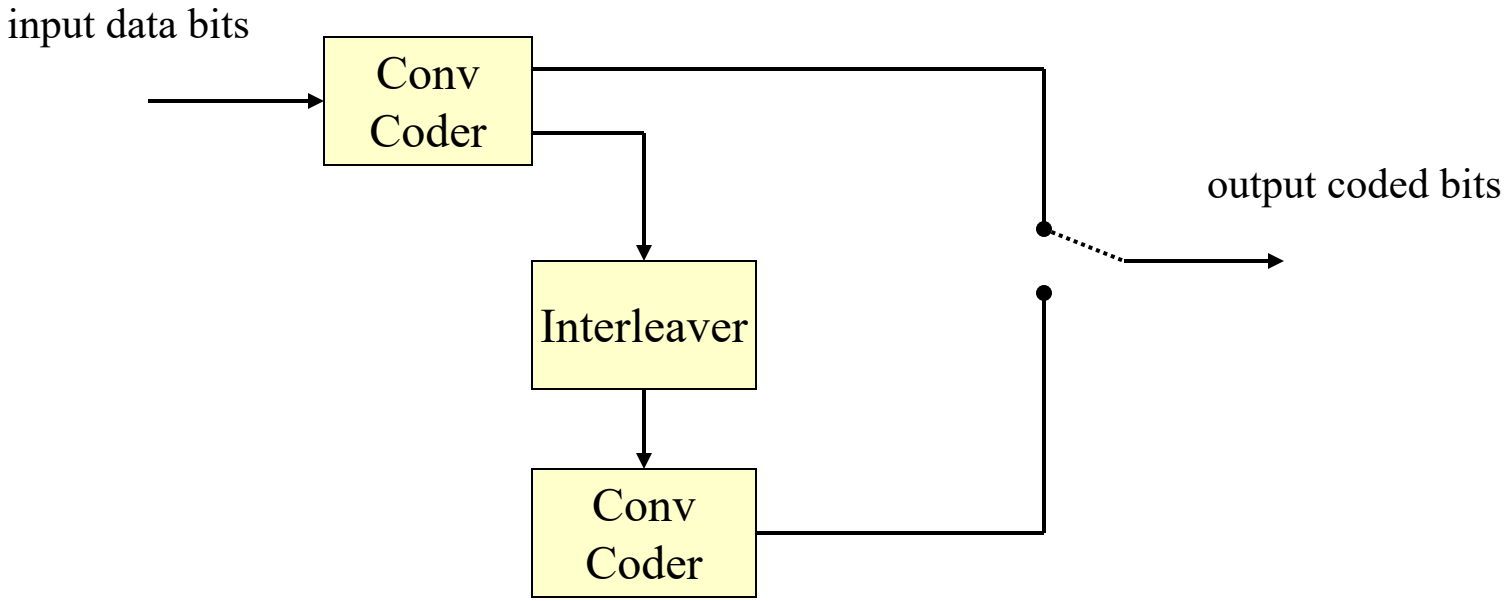
Turbo Coding [HEE99]

- Turbo coding/decoding consists of a parallel concatenated coding scheme with an iterative decoding scheme.
- Turbo coding achieves coding gains within a few tenths of a dB of the ideal (Shannon) limit.

4.3.9 Channel Coding – Turbo Coder

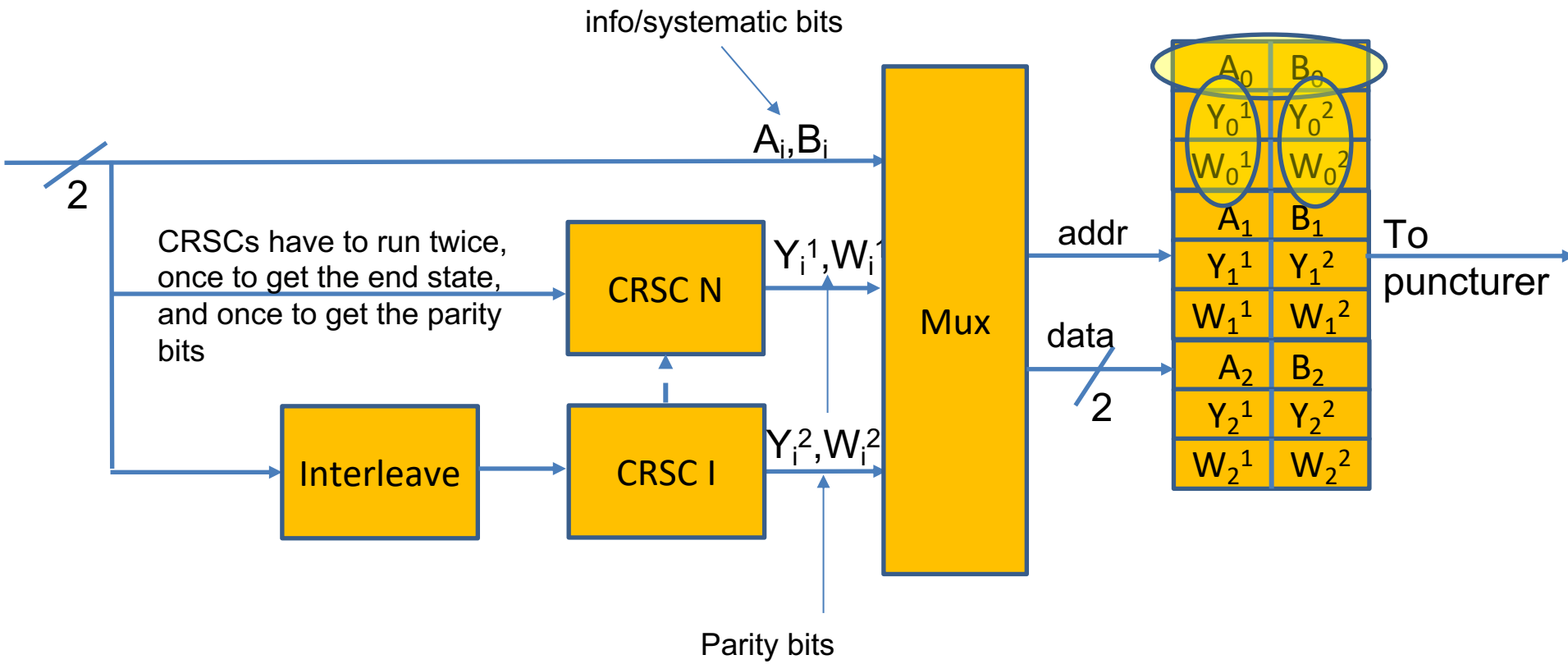
Turbo Coder [HEE99]

- The parallel structure allows the implementation of an iterative decoding scheme, whereby the output of one decoder aids the other decoder in recovering the data bits.
- The interleaver de-correlates the bit errors for the iterative decoding.



4.3.9 Channel Coding – Turbo Coder: DVB-RCS1

Note that with no puncturing (described later), the rate of the code is 2 info bits/(2+2y1w1+2y2w2) = 1/3

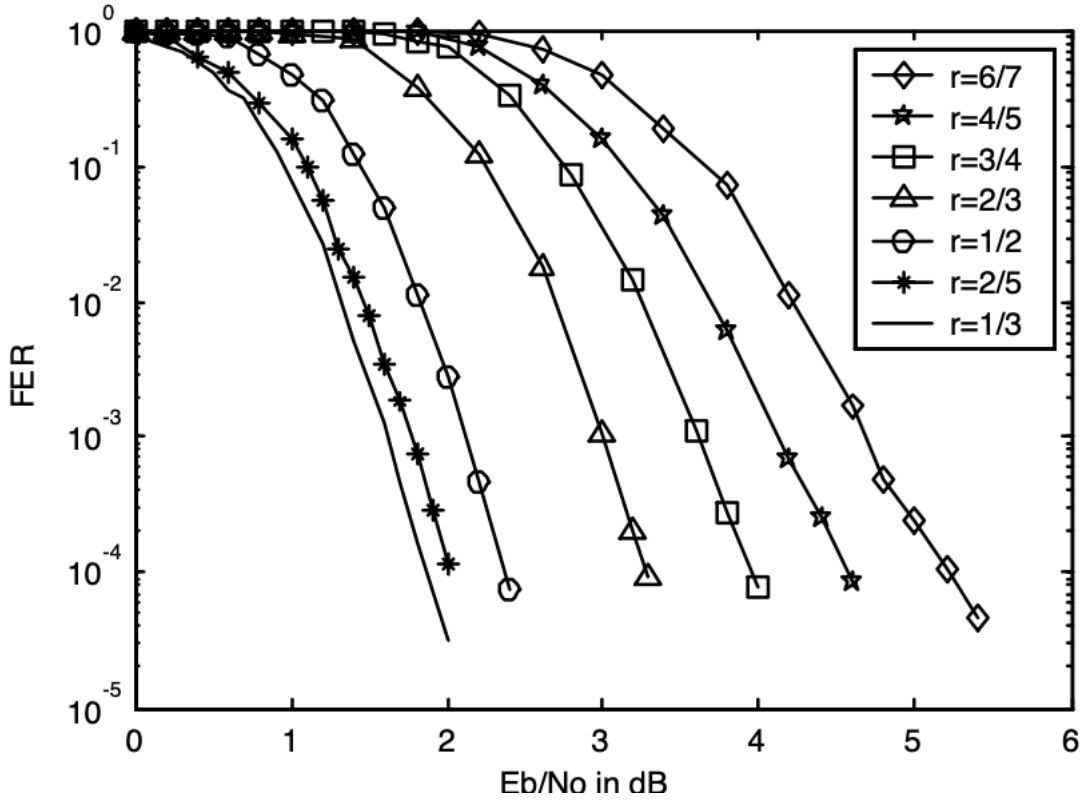


CRSC = Circular Recursive Systematic Convolutional Code

4.3.9 Channel Coding – DVB-RCS1: Puncturing

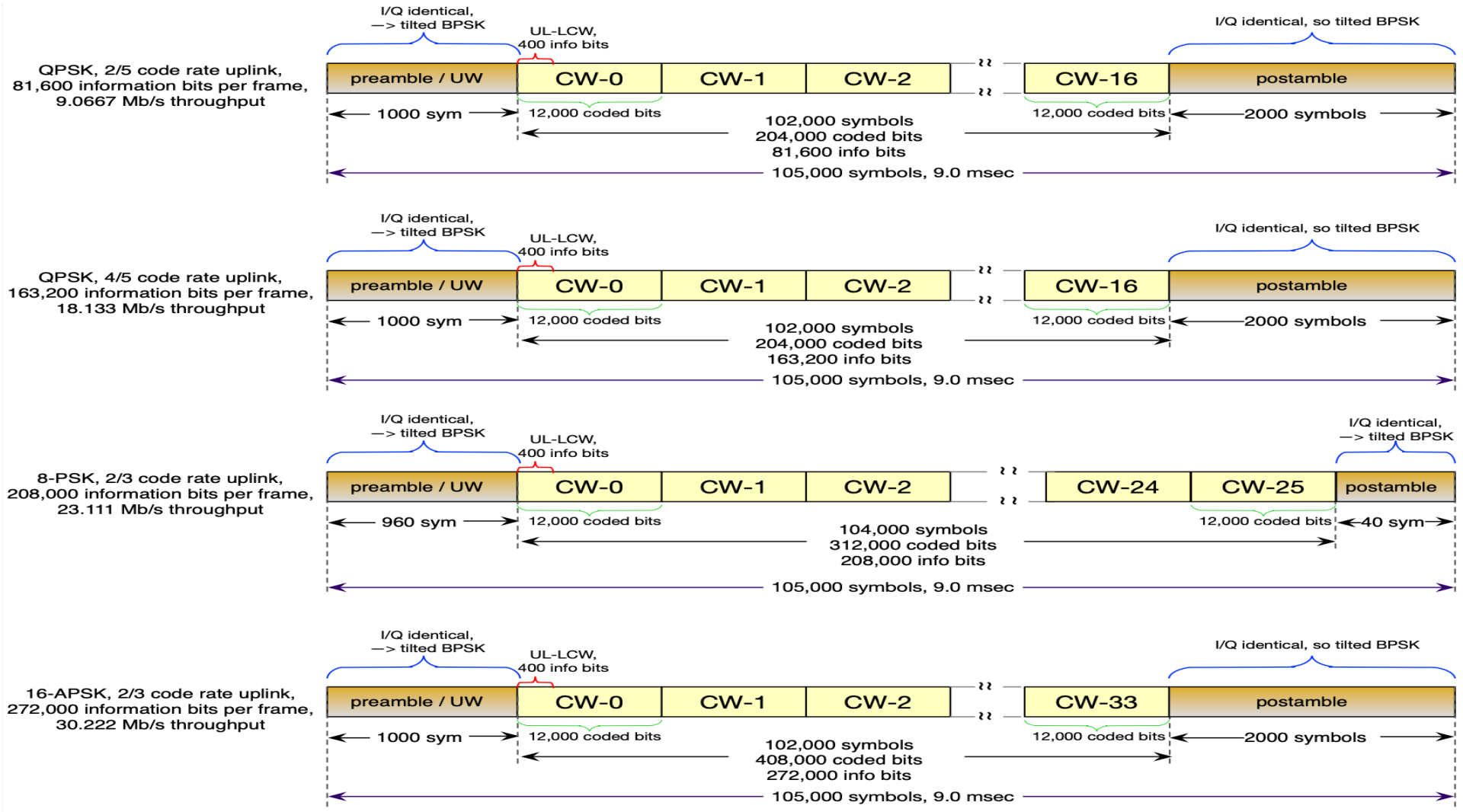
Turbo Coder - Puncturing

If we can afford to lose a few dB in our link margin, we can send data at a higher rate by puncturing/removing some of the parity bits, as seen in the plot below. E.g., in going from rate 1/3 to rate 2/3, we get twice as much throughput at the expense of 1.5 dB.



An example implementation may base the puncturing on the channel conditions. If there is a lot of fading, then send more parity bits, at reduced throughput rate. Similarly, if the channel is “clean”, send fewer parity bits, with a increased throughput rate.

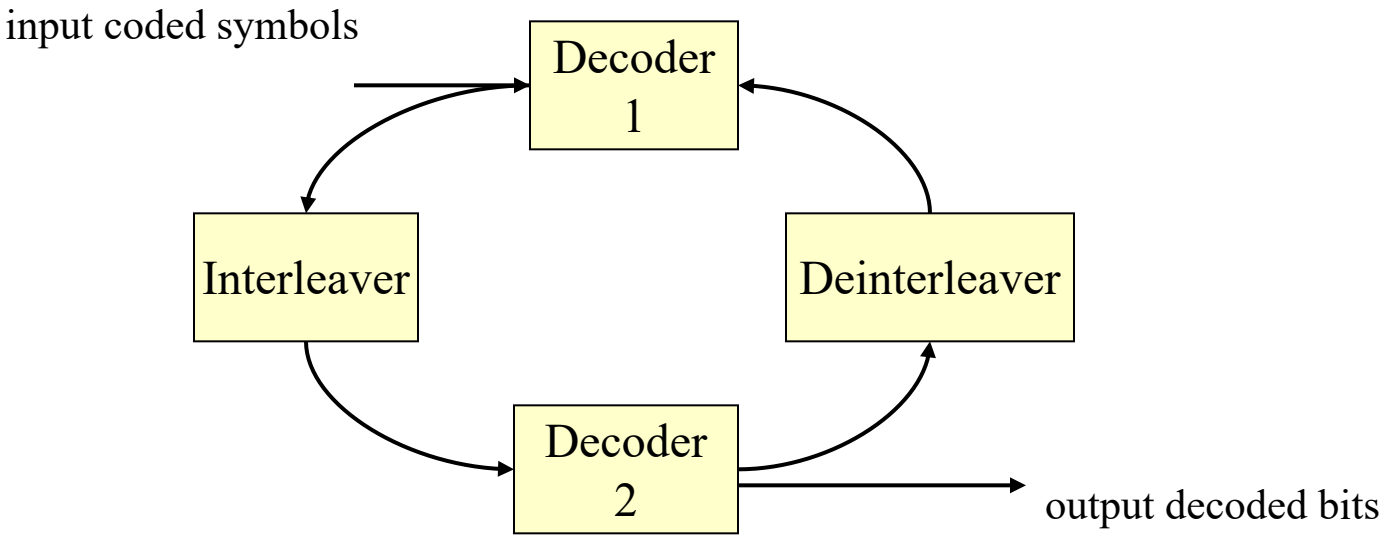
4.3.9 Channel Coding – DVB-RCS1 Framing



4.3.9 Channel Coding – Turbo Decoder

Turbo Decoder [HEE99]

The iterative decoding architecture allows the algorithm to generate improved estimates of the transmitted data.
"Soft" information is passed between the two decoders.
After about 3 iterations of the cycle below, little improvement in performance is seen.



4.3.10 Channel Coding – LDPC Codes

LDPC codes were discovered by Gallager in the early 1960s, and were ignored for 20 years, until Tanner showed a new description of LDPC codes in terms of (Tanner) graphs. Even still, LDPC codes weren't used in any systems for another 14 years until researchers found that these codes, like Turbo codes approach the Shannon limit within a few tenths of a dB.

LDPC codes are currently being used in the following systems:

- DVB-S2 (satellite communications)
- ITU-T G.hnn (home networking)
- 10GBASE-T (ethernet)
- DVB-T2 (terrestrial TV)
- 802.11n (WiFi)
- 5G (terrestrial cellular)
- DOCSIS 3.1 (cable modem)

Robert Gray Gallager

Born	May 29, 1931 (age 91) Philadelphia, Pennsylvania
Nationality	American
Alma mater	University of Pennsylvania MIT
Awards	Claude E. Shannon Award (1983) IEEE Centennial Medal (1984) IEEE Medal of Honor (1990) Harvey Prize (1999) Marconi Prize (2003) Dijkstra Prize (2004) Japan Prize (2020)
Scientific career	
Fields	Information theory
Doctoral advisor	Peter Elias
Doctoral students	Muriel Médard Elwyn Berlekamp David Tse Erdal Arıkan

4.3.10 Channel Coding – LDPC Codes

LDPC codes have a number of advantages over Turbo codes [LIN04]:

- They do not require a long interleaver (delay) to achieve good error performance.
- They have better block error performance than turbo codes.
- Their error floor occurs at a much lower BER.
- Their decoding is not trellis based (lower complexity).

4.3.10 Channel Coding – LDPC Codes

[LIN04] "An LDPC code is defined as the null space of a parity check matrix, H, that has the following structural properties:

- Each row consists of ρ 1s
- Each column consists of γ 1s
- The number of 1s in common between any two columns is <= 1
- Both ρ and γ are small in comparison with the length of the code and the number of rows in H (low density)

The density of the code, r, is given by,

$$r = \frac{\rho}{n} = \frac{\gamma}{J}, \text{where n} = \text{number of columns in H, J} = \text{number rows}$$

4.3 Channel Coding – LDPC Codes

Example [LIN04] (15,7) LDPC code

$$H = \begin{bmatrix} 0&0&0&0&0&0&0&1&1&0&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0&1&1&0&0&1&0&0 \\ 0&1&0&0&0&0&0&0&0&1&1&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&1&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&1&1&0&1 \\ 1&0&0&0&1&0&0&0&0&0&0&0&1&1&0 \\ 0&1&0&0&0&1&0&0&0&0&0&0&0&1&1 \\ 1&0&1&0&0&0&1&0&0&0&0&0&0&0&1 \\ 1&1&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&1&0&0&0&1&0&0&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&1&0&1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&1&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&1&0&1&0&0&0&1&0 \end{bmatrix}$$

$$\rho = 4$$

$$\gamma = 4$$

$$r = \frac{\rho}{n} = \frac{4}{15} = 0.2667\ (low\ density)$$

Gallagher gave a construction method for generating H from permutations of smaller matrices. In general, a computer search is used to find good parity check matrices, H

4.3 Channel Coding – LDPC Codes

Example [LIN04] (15,7) LDPC code

$$H = \begin{bmatrix} 0&0&0&0&0&0&0&1&1&0&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0&1&1&0&0&1&0&0 \\ 0&1&0&0&0&0&0&0&0&1&1&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&1&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&1&1&0&1 \\ 1&0&0&0&1&0&0&0&0&0&0&0&1&1&0 \\ 0&1&0&0&0&1&0&0&0&0&0&0&0&1&1 \\ 1&0&1&0&0&0&1&0&0&0&0&0&0&0&1 \\ 1&1&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&1&0&1&0&0&0&1&0&0&0&0&0 \\ 0&0&0&1&1&0&1&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&1&0&1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&1&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&1&0&1&0&0&0&1&0 \end{bmatrix}$$

$$\rho = 4$$

$$\gamma = 4$$

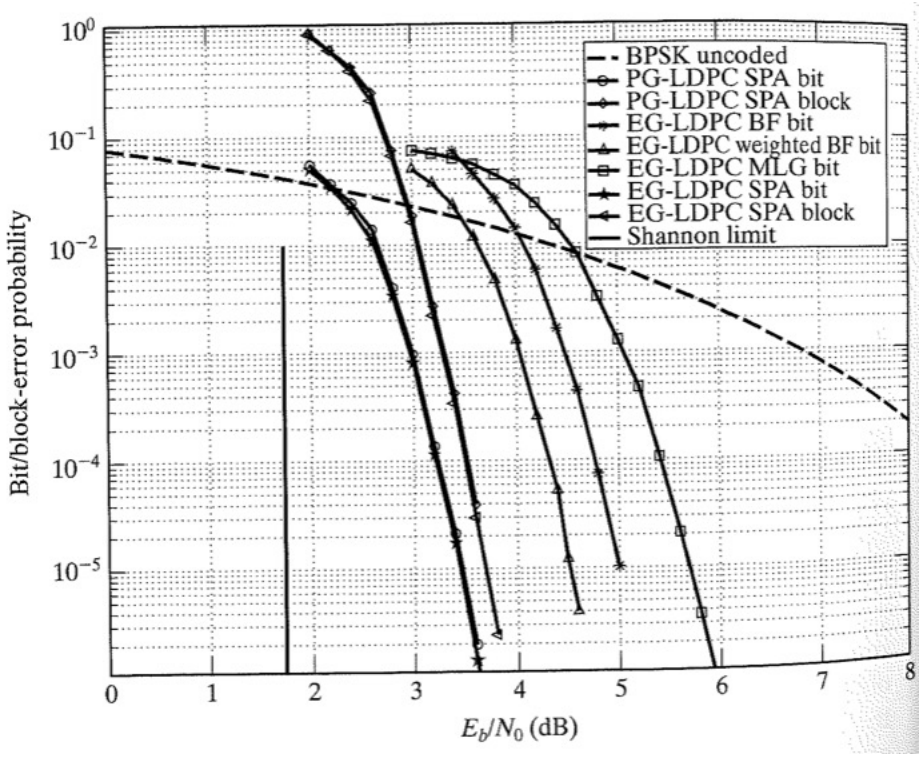
$$r = \frac{\rho}{n} = \frac{4}{15} = 0.2667\ (low\ density)$$

Gallager gave a construction method for generating H from permutations of smaller matrices. In general, a computer search is used to find good parity check matrices, H

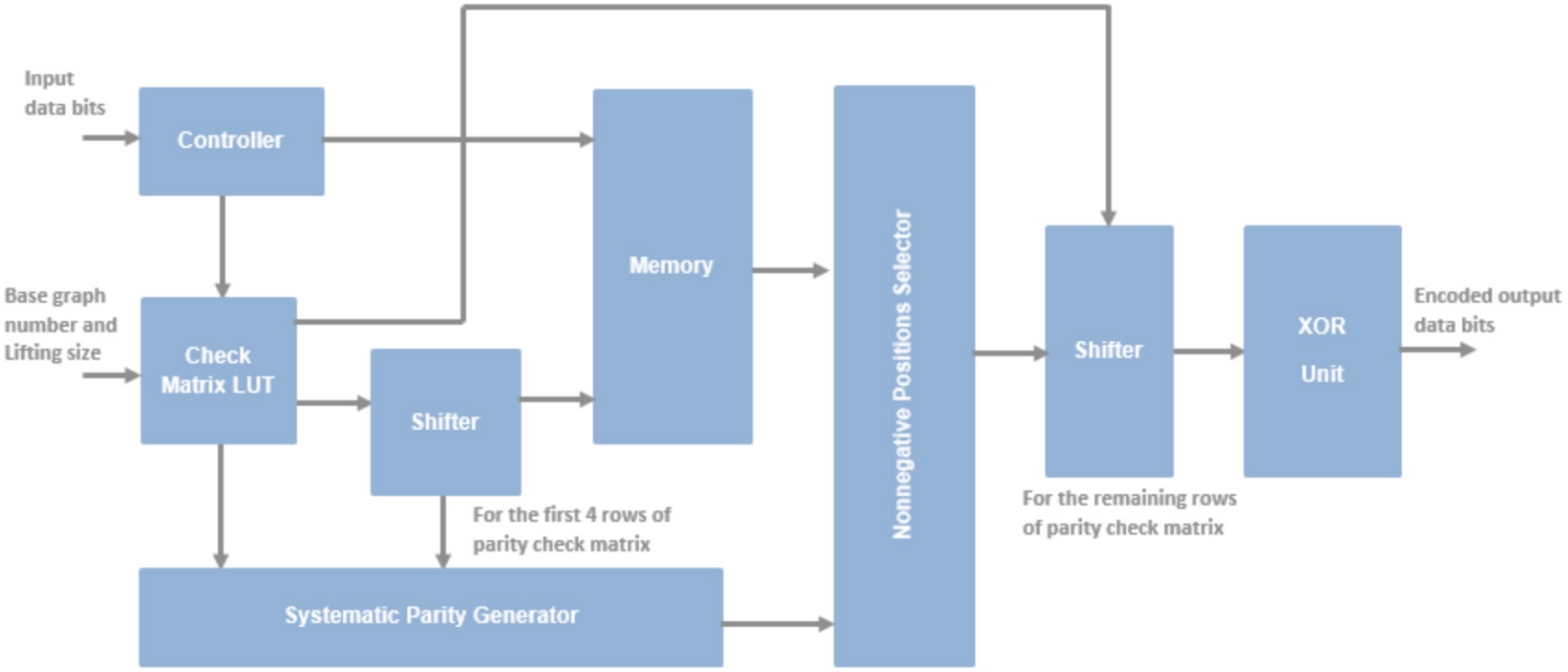
4.3 Channel Coding – LDPC Decoder

There are a variety of methods for decoding LDPC (in order of increasing complexity):

- Majority logic (MLG) – Hard decisions
- Bit flipping (BF) – Hard decisions
- Weighted BF – Soft decisions
- Iterative decoding based on belief propagation (IDBP, SPA) – Soft decisions

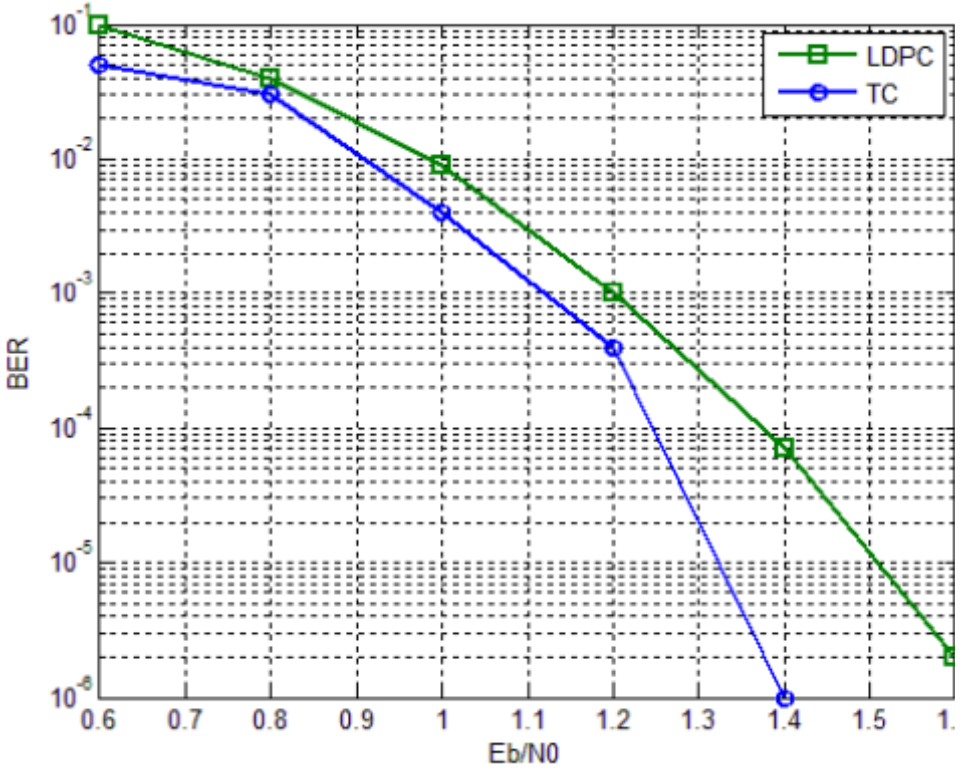


4.3 Channel Coding – LDPC Decoder Block Diagram

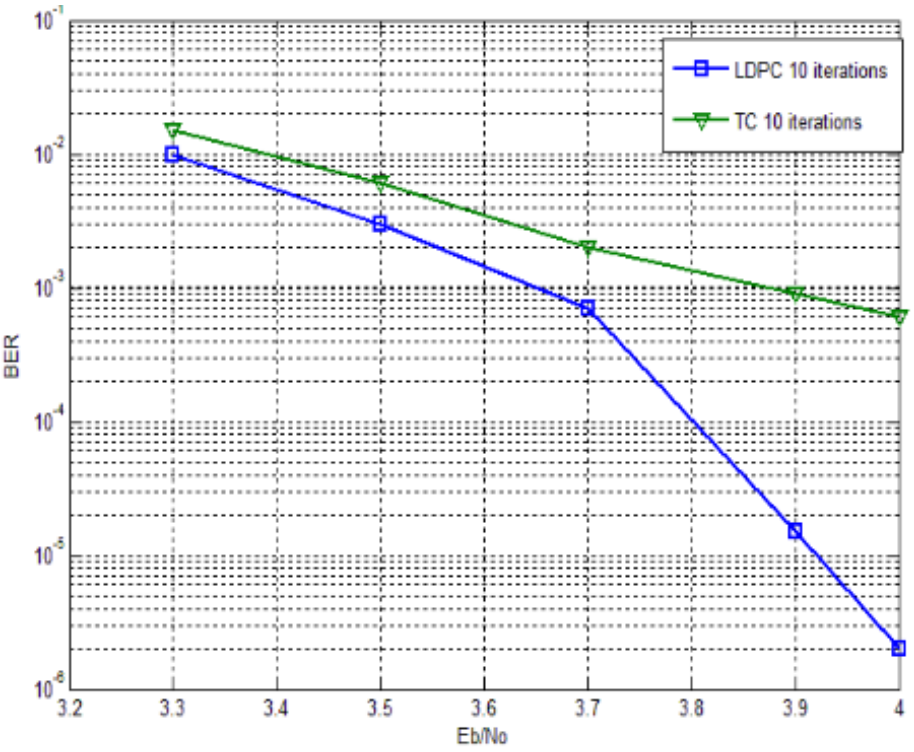


4.4 Channel Coding – Turbo vs LDPC

Comparison of Turbo vs LDPC [HAS12]



rate 7/8 codes



rate 1/2 codes

"So, it is concluded that from the point of view of performance, the LDPC is recommended for higher code rates for communication systems applications, while in the low code rates it is better to use the turbo code. In the next section, a complexity comparison for turbo code and LDPC for different code rates will take place."

Summary

- What we discussed:
  - Channel Coding
  - Error Detection
  - Error Correction
  - Block Codes
  - Turbo Codes
  - Low Density Parity Check (LDPC)

- Next …
  - Spectrum Management

References

- [RAP95] T.S. Rappaport, *Wireless Communications: Principles and Practice.*
- [MEH96] A. Mehrotra, *GSM System Engineering*
- [LEE98] J.S. Lee and L.E. Miller, *CDMA Systems Engineering Handbook.*
- [WIC95] S.B. Wicker, *Error Control Systems: for Digital Communication and Storage.*
- [STE94] R. Steele (Ed.), *Mobile Radio Communications.*
- [COU90] L.W. Couch, *Digital and Analog Communication Systems.*
- [HEE99] C. Heegard and S.B. Wicker, *Turbo Coding.*
- [JPL01] www331.jpl.nasa.gov/public/tcodes-bib.html
- [LIN04] S. Lin, D.J. Costello, *Error Control Coding*
- [HAS12] A.E.S. Hassan, M. Dessouky, A.A. Elazm, and M. Shokair, *Evaluation of Complexity Versus Performanac for Turbo Code and LDPC Under Different Code Rates.*

Acronyms

Acronym	Meaning	Notes
AWGN	Additive white Gaussian noise	e.g., thermal noise
BER	Bit error rate	Number of bits in error
CRC	Cyclic Redundancy Check (Code)	Type of FEC
FEC	Forward Error Correcting (Code)	Power measurement
LDPC	Low Density Parity Check (Code)	Shannon approaching code
PSD	Power spectral Density	
PLL	Phase-locked loop	For clock generation
SNR	Signal-to-noise ratio	Measure of signal quality