ECE 5654 Lecture 14
Hard Decision Decoding-vs-Soft Decision Decoding

Ravi Tandon

Virginia Tech

March 31, 2014

Learning Objectives

At the end of this lecture, the student should be able to:

- List the major class of block codes
- Describe what is coding gain and the impact of code rate and block length on coding gain
- Understand the differences between hard-decision decoding and soft decision decoding

Major Classes of Block Codes

- Repetition Codes
- Hamming Codes
- Golay Codes
- BCH Codes
- Reed-Solomon Codes
- LDPC Codes
- Turbo product codes (TPC)
- Walsh Codes
- ....others
- BCH and RS are most frequently used but LDPC and TPC are becoming very popular

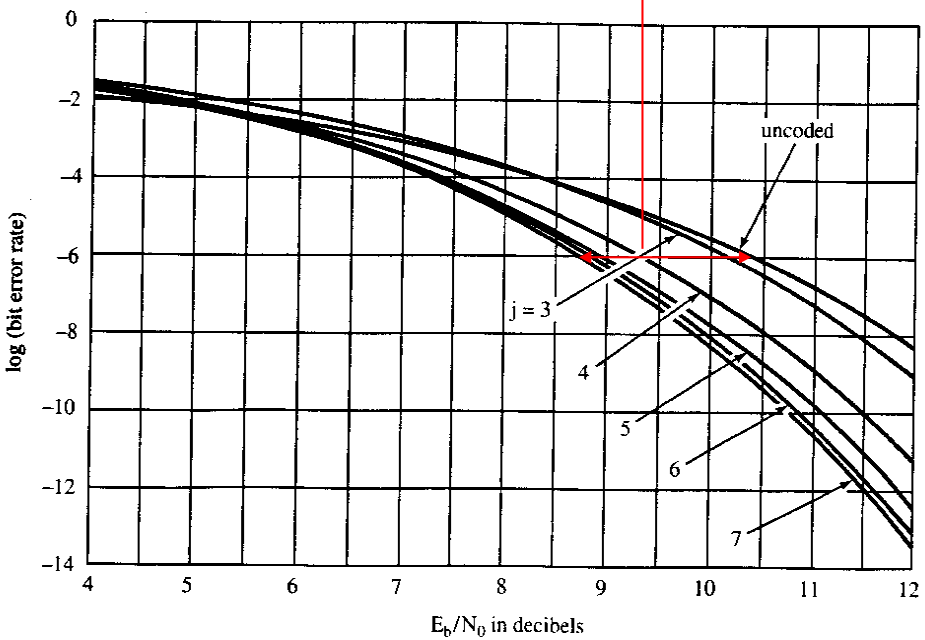
$(n, 1)$ repetition codes

- $r = \frac{k}{n} = \frac{1}{n}$
- $d_{\text{min}} = n$
- $t = \lfloor \frac{n-1}{2} \rfloor$
- $0 \rightarrow 000000$, $1 \rightarrow 111111$
- These codes are simple though wasteful of bandwidth and not widely used
- Not particularly useful in Gaussian noise but can be useful against Jamming

Hamming Codes

- $n = 2^j - 1,\ k = 2^j - 1 - j$
- $r = \frac{2^j - 1 - j}{2^j - 1},\ d_{\min} = 3,\ t = 1$
- $d_{\min}$ stays the same with n, but the rate increases with n and hence the performance improves with n
- Not in widespread practical use

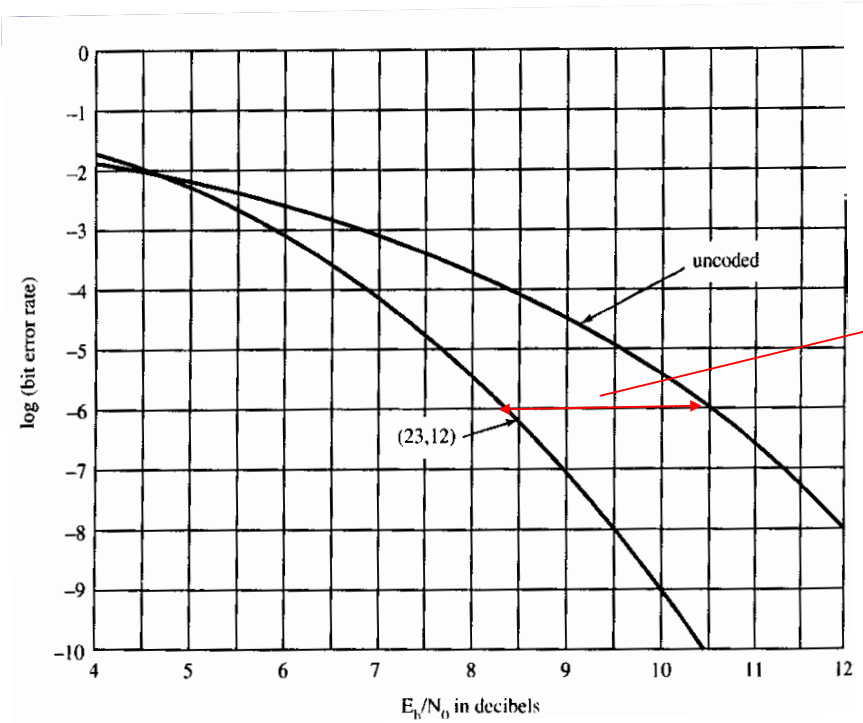
Performance of Hamming Codes



j	k	n	r	Gain (dB)
3	4	7	.57	0.2
4	11	15	.73	1.1
5	26	31	.84	1.4
6	57	63	.90	1.5
7	120	127	.94	1.6

- Decent coding gain at high SNR
- BER worse than uncoded system for low SNR
- Hamming code is not particularly powerful (single error correction capability only)

Golay Code



- One of a kind code (perfect code)
- $n = 23, k = 12, r = \frac{12}{23}$, $d_{\min} = 7$ and $t = 3$
- This code is no longer used much in practice
- Previous practical use: Motorola's old pager system

BCH Codes

- "Bose-Chaudhari-Hocquenghem" - 1959
- $n = 2^j - 1$, k (any value), $t \geq \frac{2^j - 1 - k}{j}$ (guaranteed)
- Decoded with Berlekamp-Massey algorithm (we did not cover this)
- Are a class of cyclic codes
- Widely used in satellite, wireless data links

Reed-Solomon (RS) Codes

- RS - a generalization of BCH Codes (1962)
- $n = 2^j - 1$, k (any value), $d_{\min} = n - k + 1$, $t = \lfloor \frac{n-k}{2} \rfloor$
- RS codes are Maximum Distance Separable (MDS), i.e., have the maximum possible distance
- RS codes are constructed for non-binary symbol sets - frequently used with M-ary FSK
- RS codes used for data communications in power-limited settings:
  - Deep-space communications
  - Compact Disks
  - Military comm. systems
  - Cellular Digital Packet Data

Modifications to Known Codes

- Many known codes can be modified by adding an extra codeword or deleting a symbol
- This process can create codes that approximate almost any desired rate
- Can sometimes create codes with slightly improved performance
- The resulting code can usually be decoded with only slight modification to the decoding algorithm

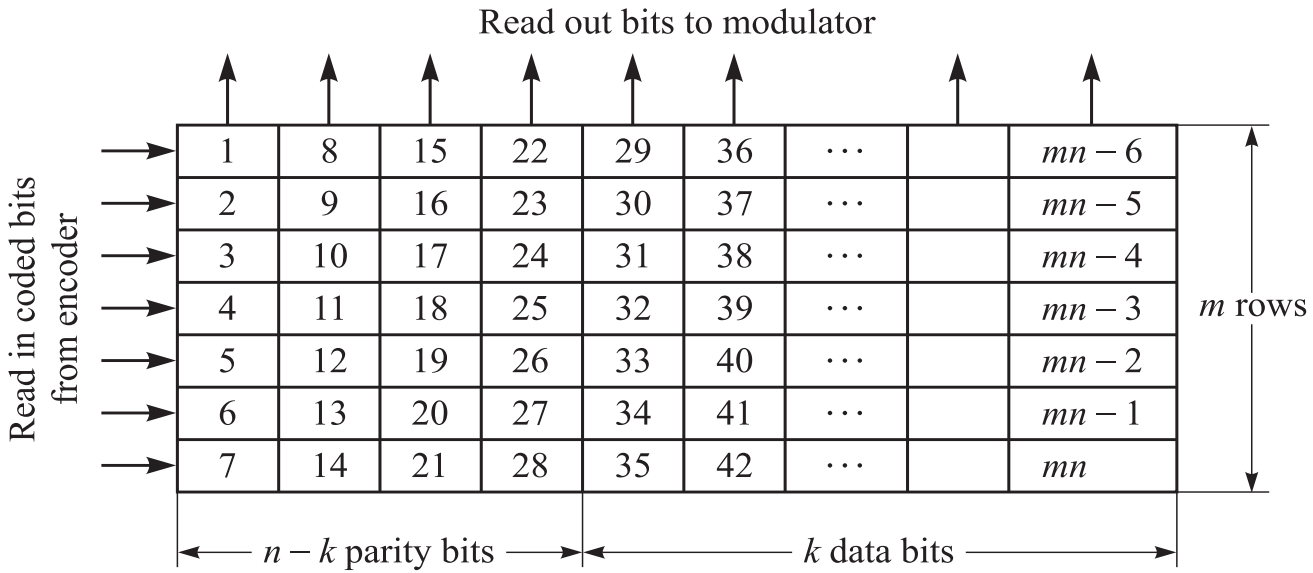
Modifications to Known Codes

- Puncturing: delete a parity symbol
  - (n, k) code $\rightarrow (n-1, k)$ code
  - Code is weakened to some extent
  - This is useful for adaptive coding scenarios or in any setting for which multiple code rates are needed since it allows for a single encoder/decoder to handle multiple rates
- Shortening: deleting a message
  - (n, k) code $\rightarrow (n-1, k-1)$ code
- Expurgating: deleting a codeword
  - (n, k) code $\rightarrow (n, k-1)$ code
- Entending: adding a parity symbol
  - (n, k) code $\rightarrow (n+1, k)$ code
- Lengthening: adding a message
  - (n, k) code $\rightarrow (n+1, k+1)$ code
- Augmenting:
  - (n, k) code $\rightarrow (n, k+1)$ code

Interleaving

- Vast majority of forward error correction (FEC) codes are designed for the AWGN channel with exhibits no memory
- Hence, they do not handle bursts of error well
- Burst errors are common in jamming scenarios as well as in time-varying (e.g., fading) channels
- Interleaving in an operation which randomizes the order of bits going into the channel.
- Burst of errors are then spread out after de-interleaving
- Ideally, this provides the decoder with random independent errors

Block Interleaving

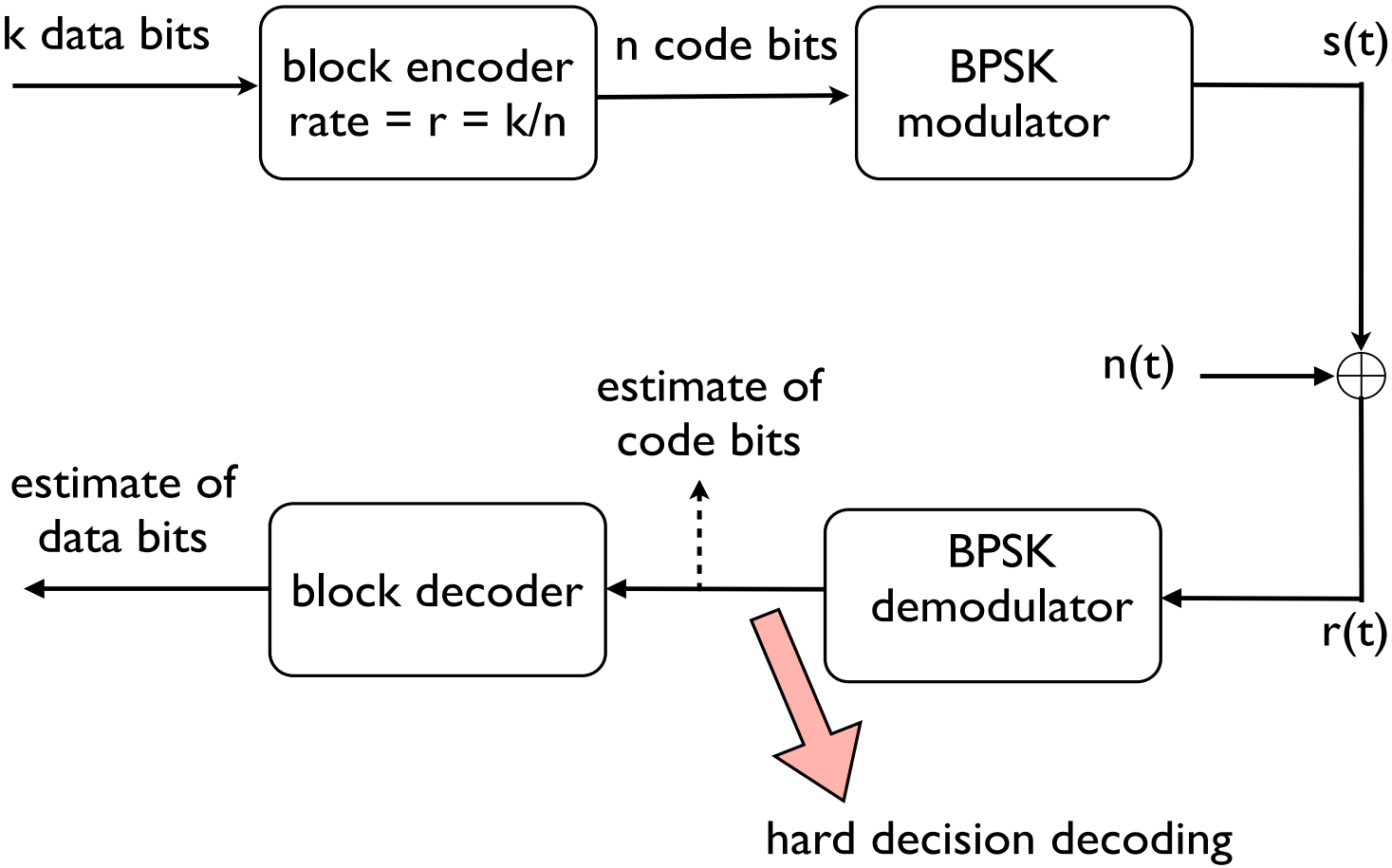


- Take m subsequent codewords (each of length n)
- m is called the degree (depth) of interleaving
- Create a $m \times n$ array, with each row corresponding to a codeword
- The encoder sequentially transmits the columns
- The decoder sequentially decodes the rows
- Longer bursts are essentially broken into smaller ones!

Hard Decisions vs. Soft Decisions

- The natural output of a demodulator is the bits that are associated with a particular symbol
- If we convert the received symbols into bits before decoding we call this Hard Decision Decoding since we make hard decisions on which bits were received before decoding
  - HDD can be thought of as finding the closest possible code sequence (in Hamming distance) to the received bit sequence
- If we instead map bits to soft values (values which represent the probability that a one or zero was sent) at the output of the demodulator, we call this Soft Decision Decoding
  - SDD can be thought of as finding the closest possible coded sequence of modulation symbols (in Euclidean distance) to the received sequence of values
  - SDD maintains more information and is thus generally more accurate but more complex

Hard Decision Decoding



Error Probabiity: Hard Decision Decoding

- Assumptions:
  - Let p be the probability that any single coded bit is in error
  - $p = P_b$ for the type of modulation used
  - Replace E_b/N_0 by rE_b/N_0
  - Errors occur independently
  - All combinations of t or less errors are correctable
  - Most combinations of more than t errors are not correctable
- Then the codeword error probability is:

$$P_c \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Example: performance of (7.4) Hamming code

- Compute the P_c for the $(7, 4)$ Hamming code with BPSK
- Since $d_{\min} = 3$, $t = 1$

$$
\begin{aligned}
P_c &\leq \sum_{i=2}^{7} \binom{7}{i} \left(Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^i \left(1 - Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^{n-i} \\
&= 1 - \sum_{i=0}^{1} \binom{7}{i} \left(Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^i \left(1 - Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^{n-i} \\
&= 1 - \left(1 - Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^7 + 7Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\left(1 - Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)\right)^6
\end{aligned}
$$

$$
\begin{aligned}
p &= Q\left(\sqrt{\frac{2rE_b}{N_o}}\right) \\
&= Q\left(\sqrt{\frac{8E_b}{7N_o}}\right)
\end{aligned}
$$

Codeword Error Probability and Bit Error Probability

- If a code word is received correctly, then all data bits will be correctly decoded.
- If a code word is received incorrectly, then between 1 and k bits will be incorrect at the output of the decoder, so:

$$\frac{1}{k} P_C \leq P_B \leq P_C$$

- We could simply make the pessimistic approximation that half the bits are in error, i.e., $P_B = P_C/2$
- To find the exact BER, we need to know how many bit errors there are at the output of the decoder whenever it makes a codeword error.

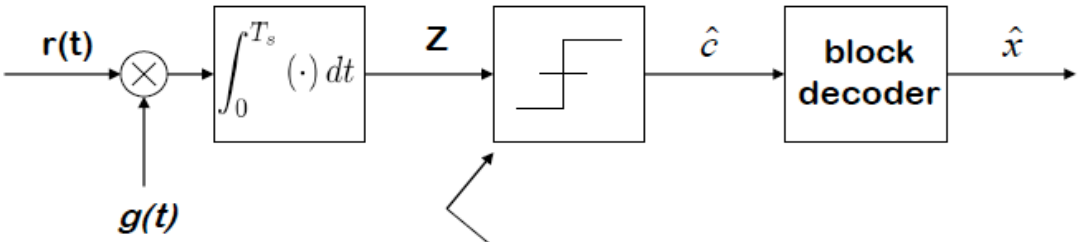
Coding Gain

- The coding gain is the difference between the uncoded and coded E_b/N_0 required to achieve a desired P_B
- Usually, we use a reference of $P_B = 10^{-5}$
- The stronger the code, the higher is the coding gain

Soft Decision Decoding

- With hard-decision decoding, a hard-decision is made on the bits before decoding takes place (input to the decode is hard bit decisions, i.e., $\{0, 1\}$)
- Whenever a hard-decision is made, valuable information is lost
- We are interested in not only if the receiver thinks the received code bit was a 0 or a 1, but how confident it was about that decision
- The decoder should rely more on strong signals, and less on weaker signals
- Any type of decoder that uses soft-information about the confidence of the bit decision is called a soft-decision decoder

HDD vs. SDD

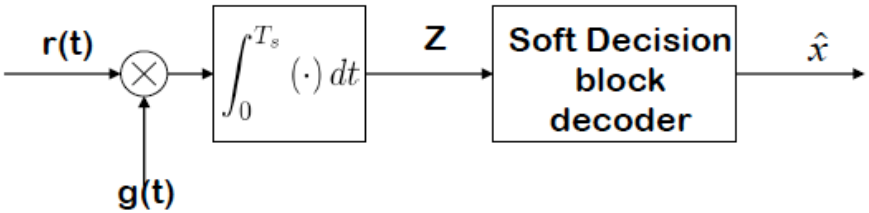


HDD

This is where the hard decision is made:

$$\hat{c} = \begin{cases} 0 & Z < 0 \\ 1 & Z > 0 \end{cases}$$

Information is lost!
Essentially a 1-bit (2 level) quantizer



SDD

The vector of **Z** values contains more information than $\hat{\mathbf{c}}$

- The magnitude of Z tells us how confident we are in the value
- Due to the "data processing theorem"
- Therefore we can obtain better performance with soft-decision decoding

Example: $(3, 1)$ repetition code

- We encode bits as $0 \rightarrow 000$, $1 \rightarrow 111$
- BPSK modulation: $0 \rightarrow g(t)$, $1 \rightarrow -g(t)$
- Example: suppose that we transmit 0 (i.e., 000)
- Receiver (after correlation), observes: $Z = -0.1, -0.05, 1.25$
- Hard Decision Decoding
  - decoded bits passed to the block decoder: $\rightarrow 110$
  - Hamming distance: $d(110, 111) = 1$
  - Hamming distance: $d(110, 000) = 2$
  - HDD decoder decides that message 0 was sent (closest in Hamming distance)
- Soft Decision Decoding
  - the vector $Z = (-0.1, -0.05, 1.25)$ is passed to the decoder
  - Decoder finds the Euclidean distance
  - $d_E(Z, 1, 1, 1) = 2.4$
  - $d_E(Z, -1, -1, -1) = 4.1$
  - It decodes in favor of the message 0 (correct decision!)

Comments on Soft Decision Decoding

- Hard decision decoding chooses the code word with smallest Hamming distance from the received code word
- Soft decision decoding chooses the code word with the smallest Euclidian distance from the received code word
- For block codes, soft-decision decoders are usually much more complex than hard-decision decoders
- However, soft-decision decoding is easy for convolutional codes

Performance of Soft Decision Decoding

- Calculate the pairwise error probability between any pair of codewords $i \neq j$:

$$P_2 = Pr(\hat{c} = c_j | c = c_i) = Q\left(\frac{d(c_i, c_j)}{\sqrt{2N_0}}\right)$$

- Where $d(c_i, c_j)$ is the Euclidean distance between the modulated codewords c_i and c_j
- Euclidean distance is related to the Hamming distance
- Depends on the type of modulation used
- E.g., for BPSK:

$$d(c_i, c_j) = 2\sqrt{E_b \times r \times d_H(c_i, c_j)}$$

where $d_H(c_i, c_j)$ is the Hamming distance

Performance of Soft Decision Decoding

- We can apply Union Bound to compute code word error probability

$$
\begin{aligned}
P_C &\leq \sum_{i=1}^{2^k} \frac{1}{2^k} \sum_{j=1, j\neq i}^{2^k} P_2(c_j, c_i) \\
&= \sum_{i=1}^{2^k} \frac{1}{2^k} \sum_{j=1, j\neq i}^{2^k} Q\left(\sqrt{\frac{2E_b r d_H(c_i, c_j)}{N_0}}\right) \\
&= \sum_{j=1, w_j\neq 0}^{2^k} Q\left(\sqrt{\frac{2E_b r w_j}{N_0}}\right) = \sum_{d=d_{\min}}^{n} Q\left(\sqrt{\frac{2E_b r d}{N_0}}\right) \\
&\approx a_{d_{\min}} Q\left(\sqrt{\frac{2E_b r d_{\min}}{N_0}}\right)
\end{aligned}
$$

- a_d is the number of codewords with weight d
- For high SNR, performance is dominated by codewords of weight $w = d_{\min}$