Defeating Noise in Communication

A Brief Explanation and Demo on Coding

Marcus Greferath

Dept. of Mathematics and Systems Analysis
School of Sciences – Aalto University

Contents of this Presentation

- What we are (not) talking about

Contents of this Presentation

- What we are (not) talking about
- Binary block codes

Contents of this Presentation

- What we are (not) talking about
- Binary block codes
- Encoders and decoders

Contents of this Presentation

- What we are (not) talking about
- Binary block codes
- Encoders and decoders
- Shannon’s promise

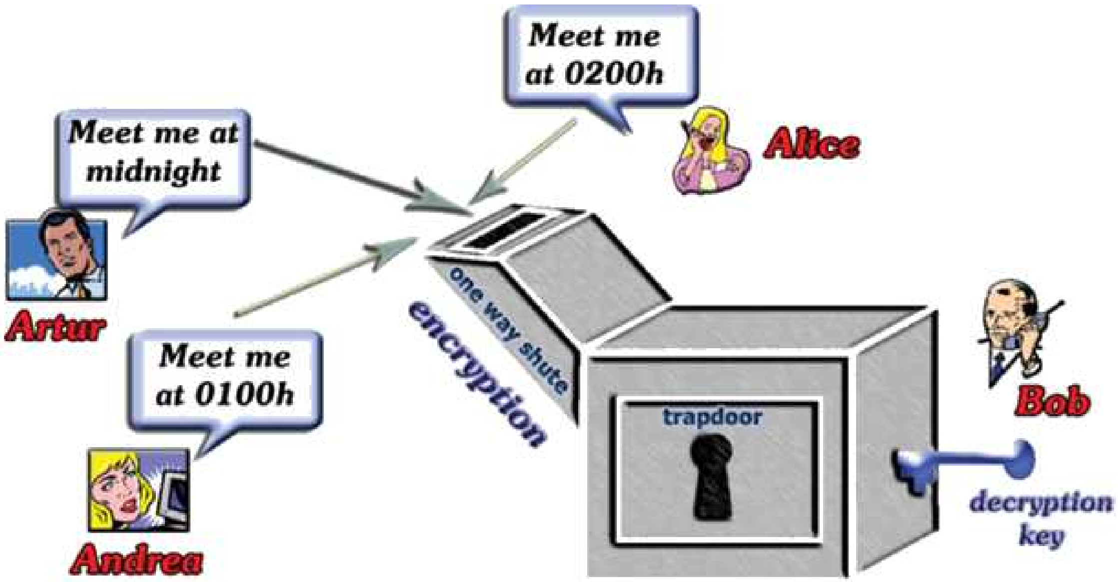
Contents of this Presentation

- What we are (not) talking about
- Binary block codes
- Encoders and decoders
- Shannon’s promise
- Reed-Muller codes

Contents of this Presentation

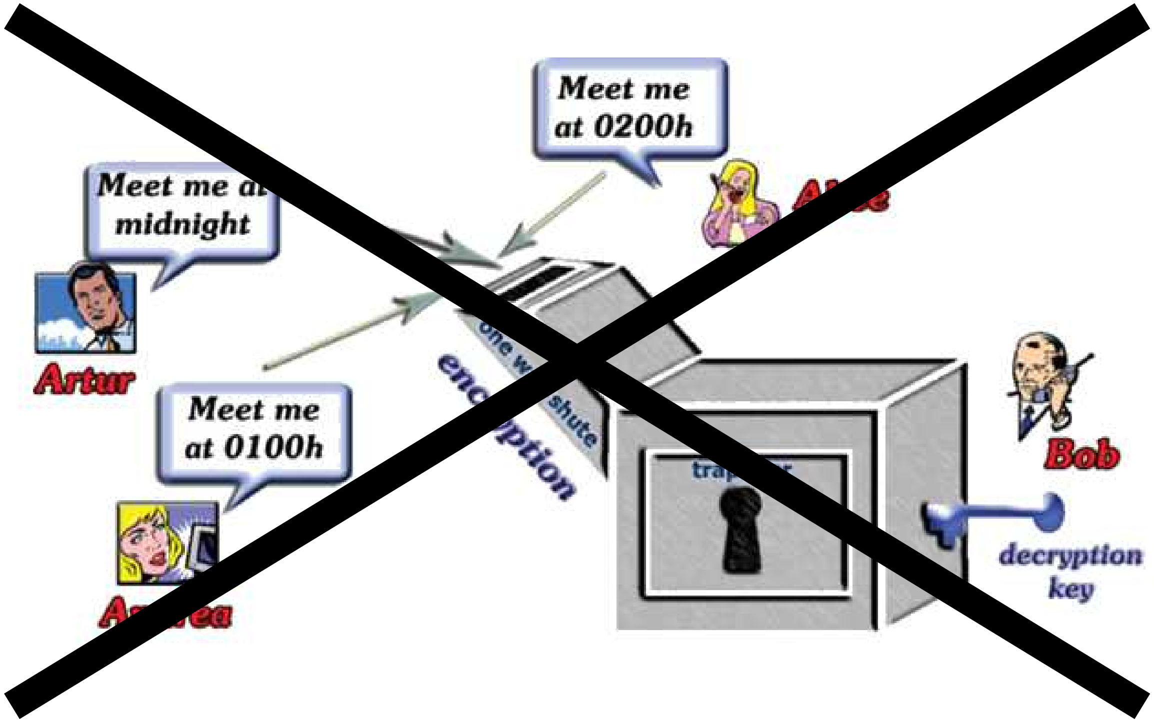
- What we are (not) talking about
- Binary block codes
- Encoders and decoders
- Shannon’s promise
- Reed-Muller codes
- An audible example

What we are **not** *talking about*



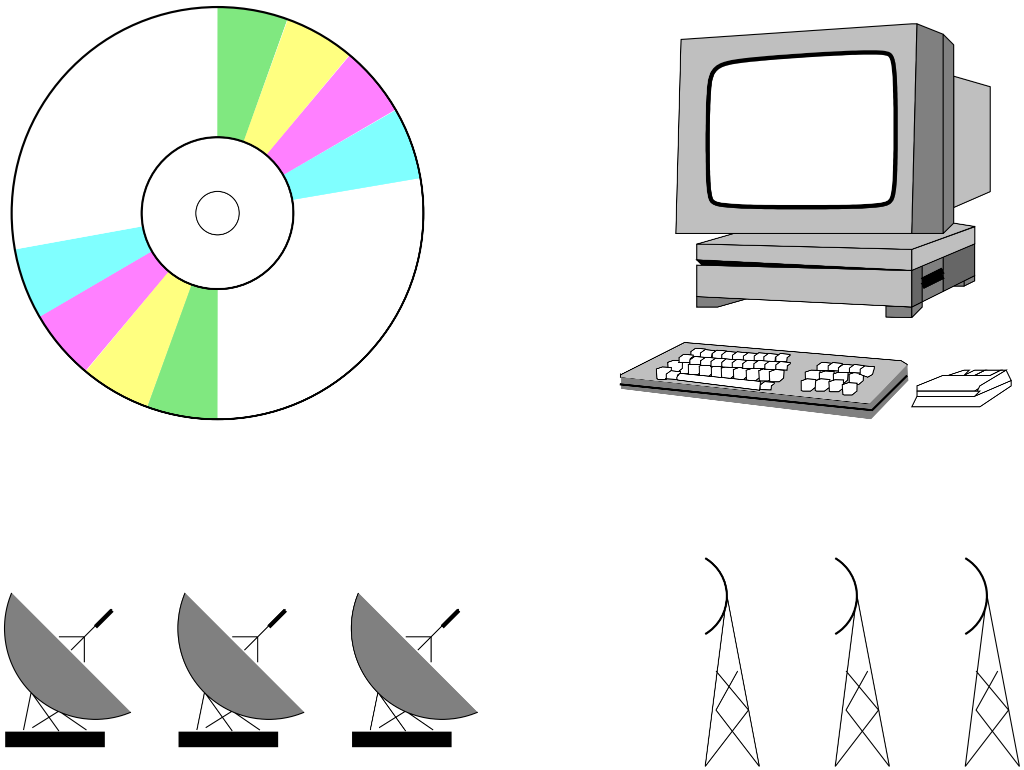
If you wish to protect communications against potential aggressors, you will use *ciphers* rather than *codes*.

What we are ***not*** *talking about*



If you wish to protect communications against potential aggressors, you will use *ciphers* rather than *codes*.

What we are talking about



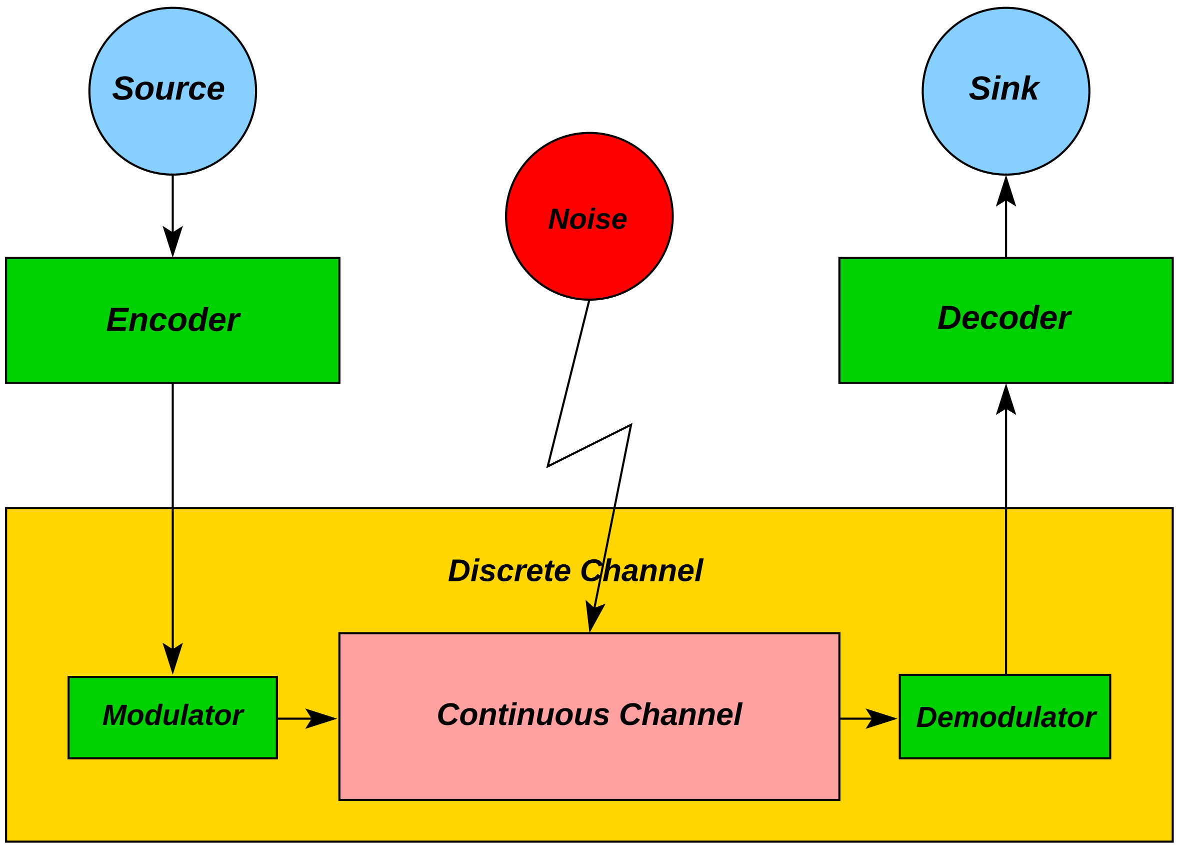
Codes protect against noise. There is no such thing as *cracking a code*.

A Quotation

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

Claude Shannon 1948

Basic Problem: Noisy Transmission



Binary Block Codes

- Assume we have only two letters, 0 and 1, and we wish to form words from these, say of length 8.

Binary Block Codes

- Assume we have only two letters, 0 and 1, and we wish to form words from these, say of length 8.
- Two examples of such words could be

$$11011001 \quad \text{and} \quad 00111110.$$

Binary Block Codes

- Assume we have only two letters, 0 and 1, and we wish to form words from these, say of length 8.
- Two examples of such words could be

$$11011001 \quad \text{and} \quad 00111110.$$

- These two words differ in 6 positions, and hence, we say their *distance* is 6.

Binary Block Codes

- Assume we have only two letters, 0 and 1, and we wish to form words from these, say of length 8.
- Two examples of such words could be

$$11011001 \quad \text{and} \quad 00111110.$$

- These two words differ in 6 positions, and hence, we say their *distance* is 6.
- A componentwise sum (agreeing to $1+1=0$) is defined and yields the new word 11100111.

Binary Block Codes

- A binary block code is a selection

$$C = \{00000000, 11011001, 00111110, 11100111\}$$

of such words of equal length (here 8).

Binary Block Codes

- A binary block code is a selection

$$C = \{00000000, 11011001, 00111110, 11100111\}$$

of such words of equal length (here 8).

- The minimum distance of this code C is the smallest distance that occurs between two of its words. Here it is seen to be 5.

Binary Block Codes

- A binary block code is a selection

$$C = \{00000000, 11011001, 00111110, 11100111\}$$

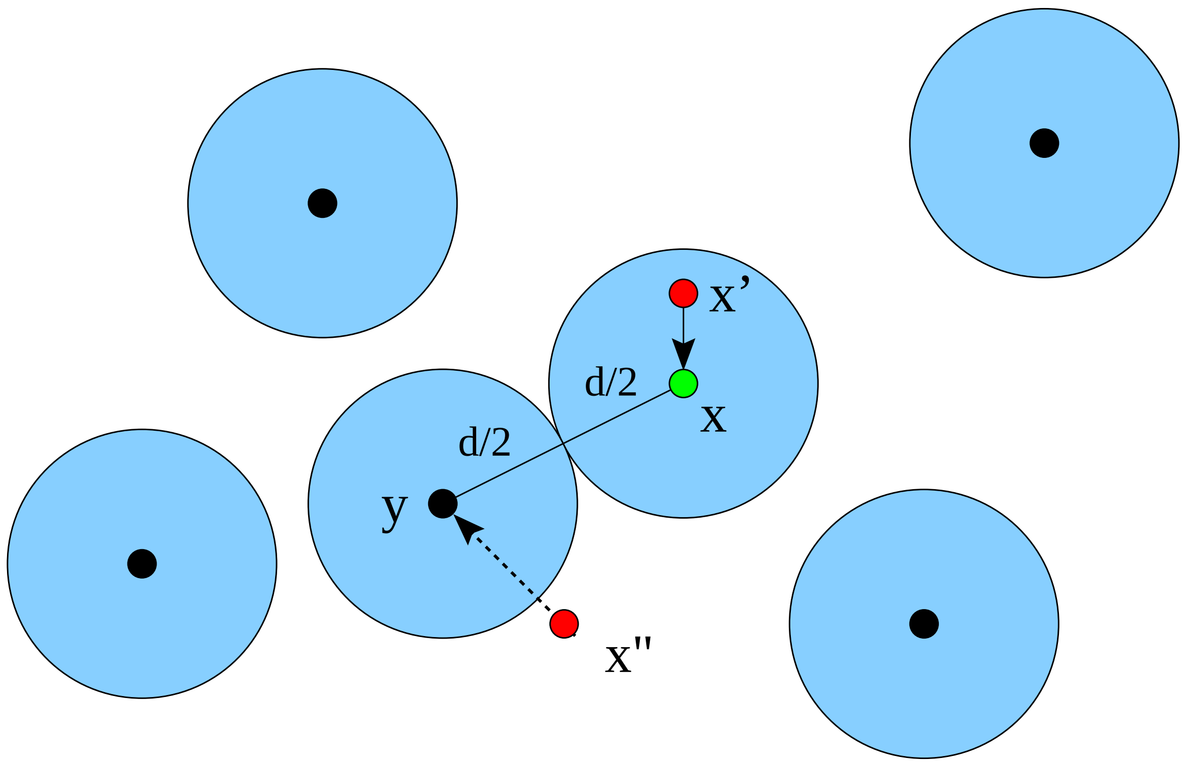
of such words of equal length (here 8).

- The minimum distance of this code C is the smallest distance that occurs between two of its words. Here it is seen to be 5.

- As this code contains 4 words, mathematicians say that C is an $(8, 4, 5)$-code.

Illustration

- **Remark:** A code of minimum distance d can be used to correct errors of weight smaller than $\frac{d}{2}$, and sometimes even more!



What is Encoding?

- Assume we use our $(8,4,5)$-code $C = \{00000000, 11011001, 00111110, 11100111\}$ as introduced above.

What is Encoding?

- Assume we use our $(8,4,5)$-code $C=\{00000000, 11011001, 00111110, 11100111\}$ as introduced above.
- We can use these 4 words to communicate 4 different binary messages, $00, 01, 11$ and 10.

What is Encoding?

- Assume we use our $(8, 4, 5)$-code $C = \{00000000, 11011001, 00111110, 11100111\}$ as introduced above.
- We can use these 4 words to communicate 4 different binary messages, $00, 01, 11$ and 10.
- Then, assigning these four messages the four codewords is what we mean by *encoding*:

$$\begin{array}{lclclcl} 00 & \mapsto & 00000000 & \quad & 01 & \mapsto & 00111110 \\ 10 & \mapsto & 11011001 & \quad & 11 & \mapsto & 11100111 \end{array}$$

What is Encoding?

- Assume we use our $(8,4,5)$-code $C = \{00000000, 11011001, 00111110, 11100111\}$ as introduced above.
- We can use these 4 words to communicate 4 different binary messages, $00, 01, 11$ and 10.
- Then, assigning these four messages the four codewords is what we mean by *encoding*:

$$\begin{array}{lllllll} 00 & \mapsto & 00000000 & \quad & 01 & \mapsto & 00111110 \\ 10 & \mapsto & 11011001 & \quad & 11 & \mapsto & 11100111 \end{array}$$

- As we encode every single bit essentially by 4 bits, we say the *rate* of C is $1/4$.

A Question

- Why did we not use the assignment

$$\begin{array}{rclcrcl} 00 & \mapsto & 00000000 & \quad & 01 & \mapsto & 00001111 \\ 10 & \mapsto & 11110000 & \quad & 11 & \mapsto & 11111111, \end{array}$$

and hence the code

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

A Question

- Why did we not use the assignment

$$\begin{array}{rclcrcl} 00 & \mapsto & 00000000 & \quad & 01 & \mapsto & 00001111 \\ 10 & \mapsto & 11110000 & \quad & 11 & \mapsto & 11111111, \end{array}$$

  and hence the code

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

- Hint: Determine the parameters of D and think of the packing illustration.

A Question

- Why did we not use the assignment

$$\begin{array}{rclcrcl} 00 & \mapsto & 00000000 & \quad & 01 & \mapsto & 00001111 \\ 10 & \mapsto & 11110000 & \quad & 11 & \mapsto & 11111111, \end{array}$$

  and hence the code

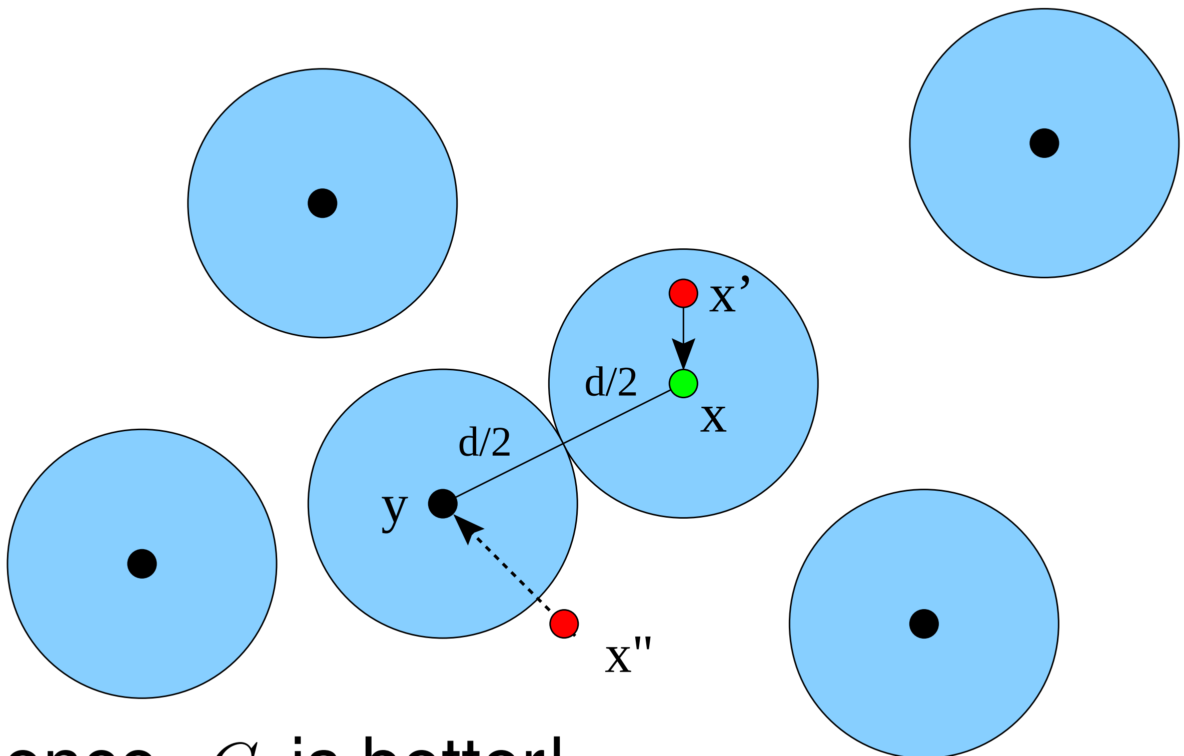
$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

- Hint: Determine the parameters of D and think of the packing illustration.

- Observation: D is an $(8, 4, 4)$ code, hence . . .

Which Code is Better?

- **Recall:** A code of minimum distance d can be used to correct errors of weight smaller than $\frac{d}{2}$, and sometimes even more!



$\ldots$ hence, C is better!

What is Decoding?

- Let us use our $(8, 4, 5)$-code
  $C = \{00000000, 11011001, 00111110, 11100111\}$.

What is Decoding?

- Let us use our $(8,4,5)$-code $C = \{00000000, 11011001, 00111110, 11100111\}$.

- At the receiving end of a noisy channel we are told that the word 11000101 has been received.

What is Decoding?

- Let us use our $(8,4,5)$-code $C = \{00000000, 11011001, 00111110, 11100111\}$.
- At the receiving end of a noisy channel we are told that the word 11000101 has been received.
- This is not a word in C! There must have been errors. Which word has most likely been sent?

What is Decoding?

- Let us use our $(8,4,5)$-code
  $C = \{00000000, 11011001, 00111110, 11100111\}$.

- At the receiving end of a noisy channel we are told that the word 11000101 has been received.

- This is not a word in C! There must have been errors. Which word has most likely been sent?

?

What is Decoding?

- By hand, we find out that among all words in C the word 11100111 is closest to the received word 11000101.

What is Decoding?

- By hand, we find out that among all words in C the word 11100111 is closest to the received word 11000101.
- Assuming that a lower number of errors is more likely than a larger number, we decide that the word 11100111 was the one originally sent.

What is Decoding?

- By hand, we find out that among all words in C the word 11100111 is closest to the received word 11000101.
- Assuming that a lower number of errors is more likely than a larger number, we decide that the word 11100111 was the one originally sent.
- A *decoder* is a device that performs the task of finding the closest codeword to a given received word.

The Other Example

- We have seen that a decoder can correct up to 2 bit changes when C is used. What if we had used

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

The Other Example

- We have seen that a decoder can correct up to 2 bit changes when C is used. What if we had used

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

- Transmit the word 11110000 and assume the channel changes 2 bits, say the 6th and the 8th.

The Other Example

- We have seen that a decoder can correct up to 2 bit changes when C is used. What if we had used

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

- Transmit the word 11110000 and assume the channel changes 2 bits, say the 6th and the 8th.
- We then receive 11110101. How will our decoder react now?

The Other Example

- We have seen that a decoder can correct up to 2 bit changes when C is used. What if we had used

$$D = \{00000000, 11110000, 00001111, 11111111\}?$$

- Transmit the word 11110000 and assume the channel changes 2 bits, say the 6th and the 8th.

- We then receive 11110101. How will our decoder react now?

- The decoder will fail, because it finds two equally likely choices, 11110000 and 11111111.

Shannon's Promise

- Assume a (very long) stream of zeros and ones is being transmitted through a noisy environment.

Shannon's Promise

- Assume a (very long) stream of zeros and ones is being transmitted through a noisy environment.
- We wish to make it robust against that noise.

Shannon's Promise

- Assume a (very long) stream of zeros and ones is being transmitted through a noisy environment.

- We wish to make it robust against that noise.

- **Recipe A:** Divide the stream into pieces of length 4, encode these to length 8 and send them off.

Shannon's Promise

- Assume a (very long) stream of zeros and ones is being transmitted through a noisy environment.
- We wish to make it robust against that noise.
- **Recipe A:** Divide the stream into pieces of length 4, encode these to length 8 and send them off.
- **Recipe B:** Divide the stream into pieces of length 32, encode these to length 64 and send them off.

Shannon's Promise

- Assume a (very long) stream of zeros and ones is being transmitted through a noisy environment.
- We wish to make it robust against that noise.
- **Recipe A:** Divide the stream into pieces of length 4, encode these to length 8 and send them off.
- **Recipe B:** Divide the stream into pieces of length 32, encode these to length 64 and send them off.
- What is better, given the same noise level?

Shannon's Promise

- Suppose we wish to transmit information at a certain constant rate, say $1/2$, over a channel that flips bits with probability ≤ 0.11.

Shannon's Promise

- Suppose we wish to transmit information at a certain constant rate, say $1/2$, over a channel that flips bits with probability ≤ 0.11.
- **Theorem:** Extending the the length (and keeping the rate) of the used codes we can achieve arbitrary reliability of the communication process.

Shannon's Promise

- Suppose we wish to transmit information at a certain constant rate, say $1/2$, over a channel that flips bits with probability ≤ 0.11.
- **Theorem:** Extending the the length (and keeping the rate) of the used codes we can achieve arbitrary reliability of the communication process.
- In other words, recipe B is preferable to recipe A. By going up to higher length, communication errors will become less and less likely.

A Puzzle: Pascal's Triangle

1							
1	1						
1	2	1					
1	3	3	1				
1	4	6	4	1			
1	5	10	10	5	1		
1	6	15	20	15	6	1	
?	?	?	?	?	?	?	?

What are the next entries in this table?

The Plotkin Sum of Codes

- Let C and D be the codes studied earlier. We define

$$C \oplus D \ := \ \{(c, c+d) \mid c \in C,\ d \in D\}$$

which is a code of length 16.

The Plotkin Sum of Codes

- Let C and D be the codes studied earlier. We define

$$C \oplus D \ := \ \{(c, c+d) \mid c \in C,\ d \in D\}$$

which is a code of length 16.

- **Concretely:** For the words 11011001 taken from the code C and 11110000 taken from D we find

$$(11011001, 11011001 + 11110000) = 1101100100101001.$$

The Plotkin Sum of Codes

- Let C and D be the codes studied earlier. We define

$$C \oplus D \;:=\; \{(c, c+d) \mid c \in C,\; d \in D\}$$

which is a code of length 16.

- **Concretely:** For the words 11011001 taken from the code C and 11110000 taken from D we find

$$(11011001, 11011001 + 11110000) = 1101100100101001.$$

- Proceeding in the same way with all choices of words in C and D we see that $C \oplus D$ contains 16 words.

Reed-Muller Codes

- **Definition:** For $m \in \mathbb{N}$ and $0 \leq r \leq m$ we define a family $\mathrm{RM}(r,m)$ of linear codes by:
  - $\star$ $\mathrm{RM}(0,m) = \{000\ldots0, 111\ldots1\}$ of length 2^m.
  - $\star$ $\mathrm{RM}(m,m)$ is the set of all words of length 2^m.
  - $\star$ For all $m \geq 1$ and $1 \leq r \leq m-1$

$$\mathrm{RM}(r,m) \;:=\; \mathrm{RM}(r,m-1) \oplus \mathrm{RM}(r-1,m-1).$$

Reed-Muller Codes

- **Definition:** For $m \in \mathbb{N}$ and $0 \leq r \leq m$ we define a family $\mathrm{RM}(r,m)$ of linear codes by:
  - $\star$ $\mathrm{RM}(0,m) = \{000\ldots0, 111\ldots1\}$ of length 2^m.
  - $\star$ $\mathrm{RM}(m,m)$ is the set of all words of length 2^m.
  - $\star$ For all $m \geq 1$ and $1 \leq r \leq m-1$

$$\mathrm{RM}(r,m) \;:=\; \mathrm{RM}(r,m-1) \oplus \mathrm{RM}(r-1,m-1).$$

- **Observation:** When m is odd, and $r = (m-1)/2$ then $\mathrm{RM}(r,m)$ has rate $1/2$.

Reed-Muller Codes

- **Definition:** For $m \in \mathbb{N}$ and $0 \leq r \leq m$ we define a family $\mathrm{RM}(r,m)$ of linear codes by:
  - $\star$ $\mathrm{RM}(0,m) = \{000\ldots 0, 111\ldots 1\}$ of length 2^m.
  - $\star$ $\mathrm{RM}(m,m)$ is the set of all words of length 2^m.
  - $\star$ For all $m \geq 1$ and $1 \leq r \leq m-1$

$$\mathrm{RM}(r,m) \;:=\; \mathrm{RM}(r,m-1) \oplus \mathrm{RM}(r-1,m-1).$$

- **Observation:** When m is odd, and $r = (m-1)/2$ then $\mathrm{RM}(r,m)$ has rate $1/2$.

Reed-Muller Codes

- **Definition:** For $m \in \mathbb{N}$ and $0 \leq r \leq m$ we define a family $\mathrm{RM}(r,m)$ of linear codes by:
  - $\star$ $\mathrm{RM}(0,m) = \{000\ldots0, 111\ldots1\}$ of length 2^m.
  - $\star$ $\mathrm{RM}(m,m)$ is the set of all words of length 2^m.
  - $\star$ For all $m \geq 1$ and $1 \leq r \leq m-1$

$$\mathrm{RM}(r,m) \;:=\; \mathrm{RM}(r,m-1) \oplus \mathrm{RM}(r-1,m-1).$$

- **Observation:** When m is odd, and $r = (m-1)/2$ then $\mathrm{RM}(r,m)$ has rate $1/2$.

Reed-Muller Codes

- **Definition:** For $m \in \mathbb{N}$ and $0 \le r \le m$ we define a family $\mathrm{RM}(r, m)$ of linear codes by:
  - ⋆ $\mathrm{RM}(0, m) = \{000 \ldots 0, 111 \ldots 1\}$ of length 2^m.
  - ⋆ $\mathrm{RM}(m, m)$ is the set of all words of length 2^m.
  - ⋆ For all $m \ge 1$ and $1 \le r \le m - 1$

$$\mathrm{RM}(r, m) \ := \ \mathrm{RM}(r, m-1) \oplus \mathrm{RM}(r-1, m-1).$$

- **Observation:** When m is odd, and $r = (m-1)/2$ then $\mathrm{RM}(r, m)$ has rate $1/2$.

Another Pascal Triangle

$$\begin{array}{cccccc}
\mathrm{RM}(0,0) & & & & & \\
\mathrm{RM}(0,1) & \mathrm{RM}(1,1) & & & & \\
\mathrm{RM}(0,2) & \mathrm{RM}(1,2) & \mathrm{RM}(2,2) & & & \\
\mathrm{RM}(0,3) & \mathrm{RM}(1,3) & \mathrm{RM}(2,3) & \mathrm{RM}(3,3) & & \\
\mathrm{RM}(0,4) & \mathrm{RM}(1,4) & \mathrm{RM}(2,4) & \mathrm{RM}(3,4) & \mathrm{RM}(4,4) & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots
\end{array}$$

The recursion of Pascal’s triangle is underlying!

- According to the previous slide

$$\begin{aligned}
\mathrm{RM}(1,5) &= \mathrm{RM}(0,4) \oplus \mathrm{RM}(1,4), \\
\mathrm{RM}(1,4) &= \mathrm{RM}(0,3) \oplus \mathrm{RM}(1,3), \\
\mathrm{RM}(1,3) &= \mathrm{RM}(0,2) \oplus \mathrm{RM}(1,2), \\
\mathrm{RM}(1,2) &= \mathrm{RM}(0,1) \oplus \mathrm{RM}(1,1).
\end{aligned}$$

Here, $\mathrm{RM}(0,1) = \{00, 11\}$, $\mathrm{RM}(0,2) = \{0000, 1111\}$, and so forth, and $\mathrm{RM}(1,1) = \{00, 01, 10, 11\}$.

Example $\mathrm{RM}(1,5)$

- According to the previous slide

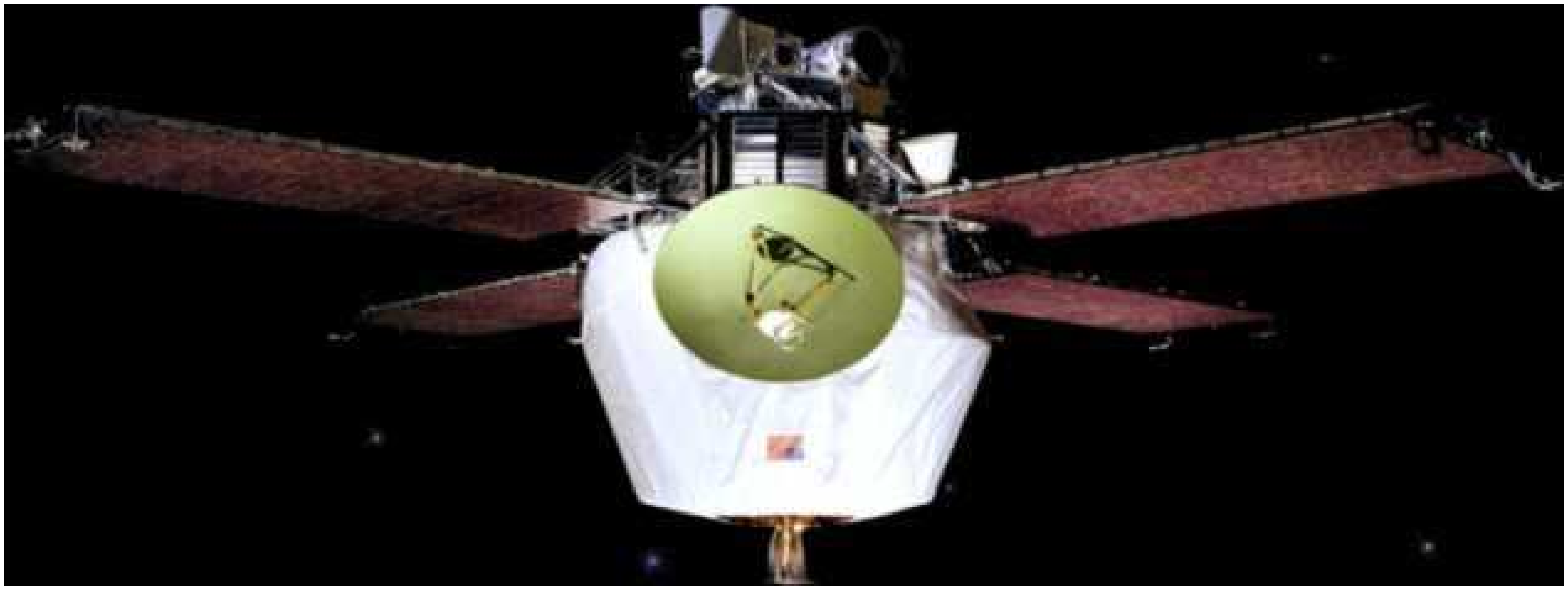
$$\begin{aligned}\mathrm{RM}(1,5) &= \mathrm{RM}(0,4) \oplus \mathrm{RM}(1,4),\\ \mathrm{RM}(1,4) &= \mathrm{RM}(0,3) \oplus \mathrm{RM}(1,3),\\ \mathrm{RM}(1,3) &= \mathrm{RM}(0,2) \oplus \mathrm{RM}(1,2),\\ \mathrm{RM}(1,2) &= \mathrm{RM}(0,1) \oplus \mathrm{RM}(1,1).\end{aligned}$$

  Here, $\mathrm{RM}(0,1) = \{00, 11\}$, $\mathrm{RM}(0,2) = \{0000, 1111\}$, and so forth, and $\mathrm{RM}(1,1) = \{00, 01, 10, 11\}$.

- The entire code consists of 64 words of length 32, has minimum distance 16, hence hence can correct up to 7 errors.

The Mariner-9 Mission

- The code $\mathrm{RM}(1,5)$ was used in the Mariner-9 program of NASA.



The Mariner-9 spacecraft

Performance Questions

- The noise level of a given channel is usually represented by the signal-to-noise ratio λ.

Performance Questions

- The noise level of a given channel is usually represented by the signal-to-noise ratio λ.
- Even if we use an error-correcting code C there will still a probability $P_{BE}(C, \lambda)$ that a transmitted word is decoded wrongly.

Performance Questions

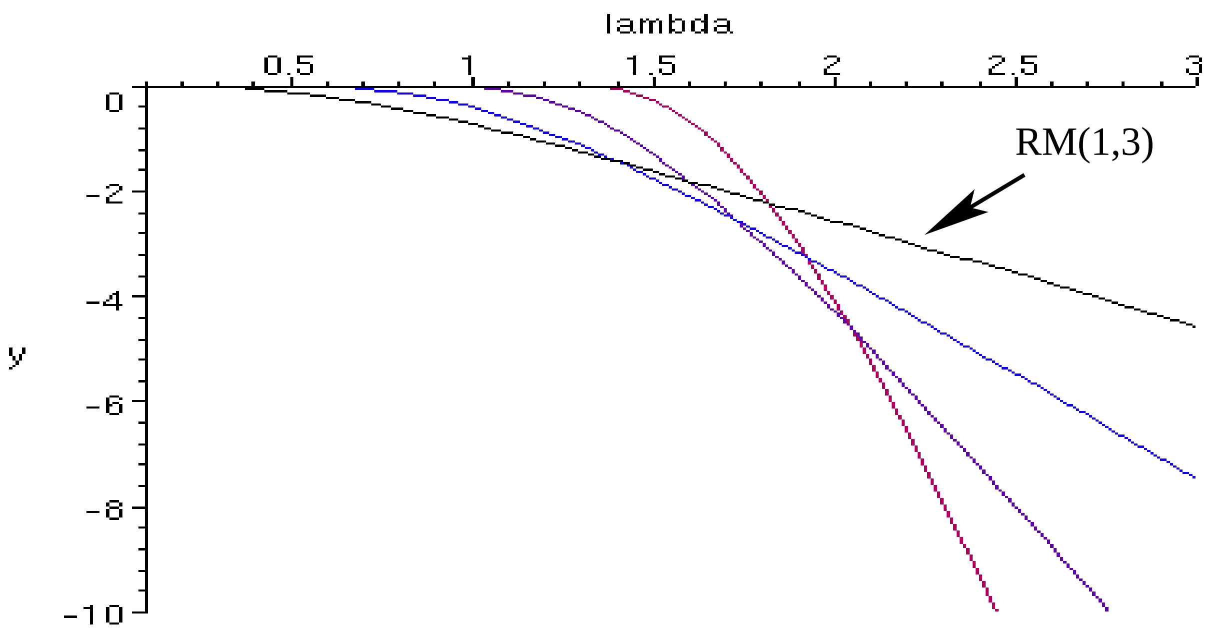
- The noise level of a given channel is usually represented by the signal-to-noise ratio λ.
- Even if we use an error-correcting code C there will still a probability $P_{BE}(C, \lambda)$ that a transmitted word is decoded wrongly.
- Performance comparison of different codes is usually done in a logarithmic plot of $P_{BE}(C, \lambda)$ against λ.

Performance Questions

- The noise level of a given channel is usually represented by the signal-to-noise ratio λ.
- Even if we use an error-correcting code C there will still a probability $P_{BE}(C, \lambda)$ that a transmitted word is decoded wrongly.
- Performance comparison of different codes is usually done in a logarithmic plot of $P_{BE}(C, \lambda)$ against λ.
- We will however not go deeper into the mathematics behind this.

Performance Questions (cntd)

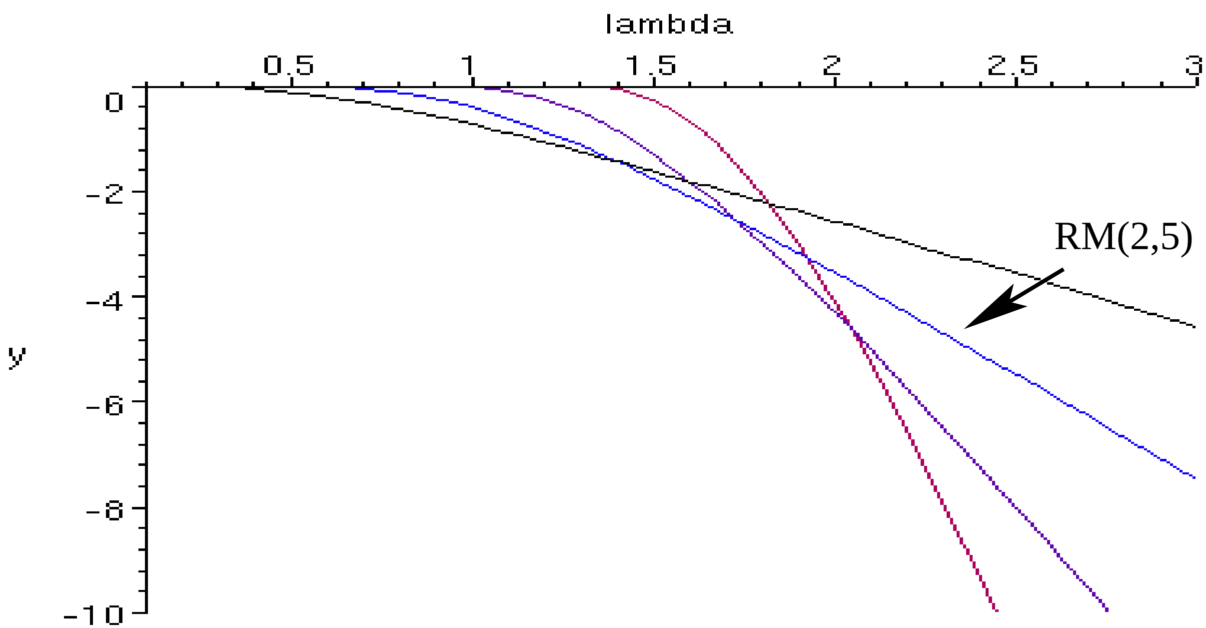
We do this with the codes $\mathrm{RM}(r,m)$ where $(r,m) = (1,3),(2,5),(3,7)$ and $(4,9)$.



Performance Comparison

Performance Questions (cntd)

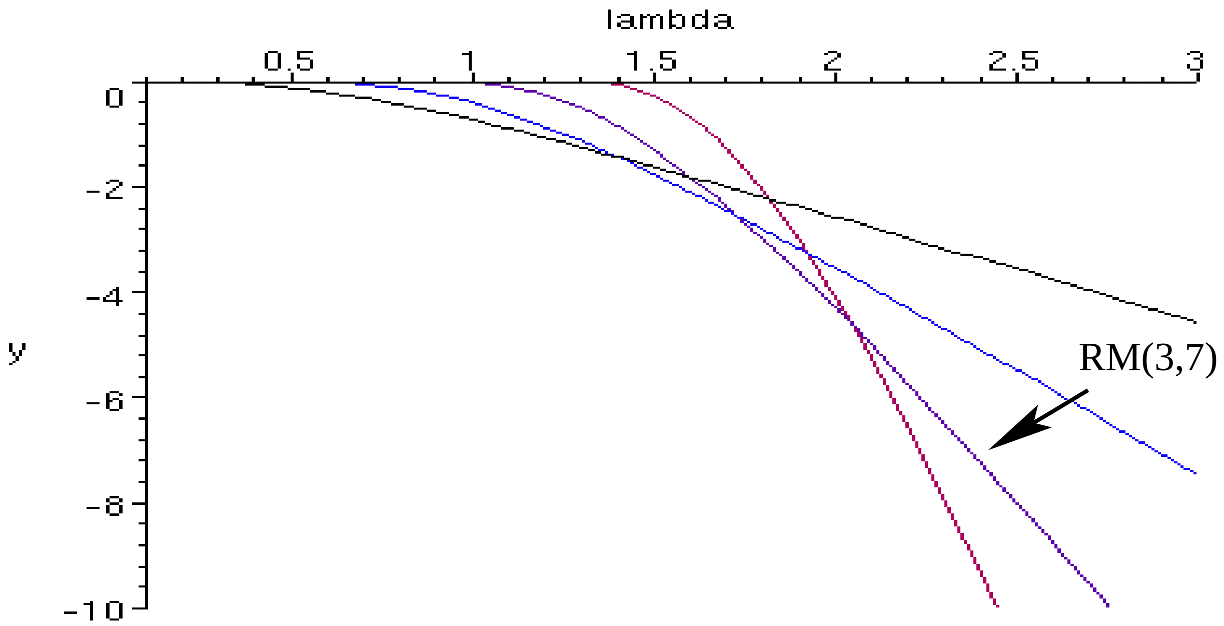
We do this with the codes $\mathrm{RM}(r,m)$ where $(r,m) = (1,3), (2,5), (3,7)$ and $(4,9)$.



Performance Comparison

Performance Questions (cntd)

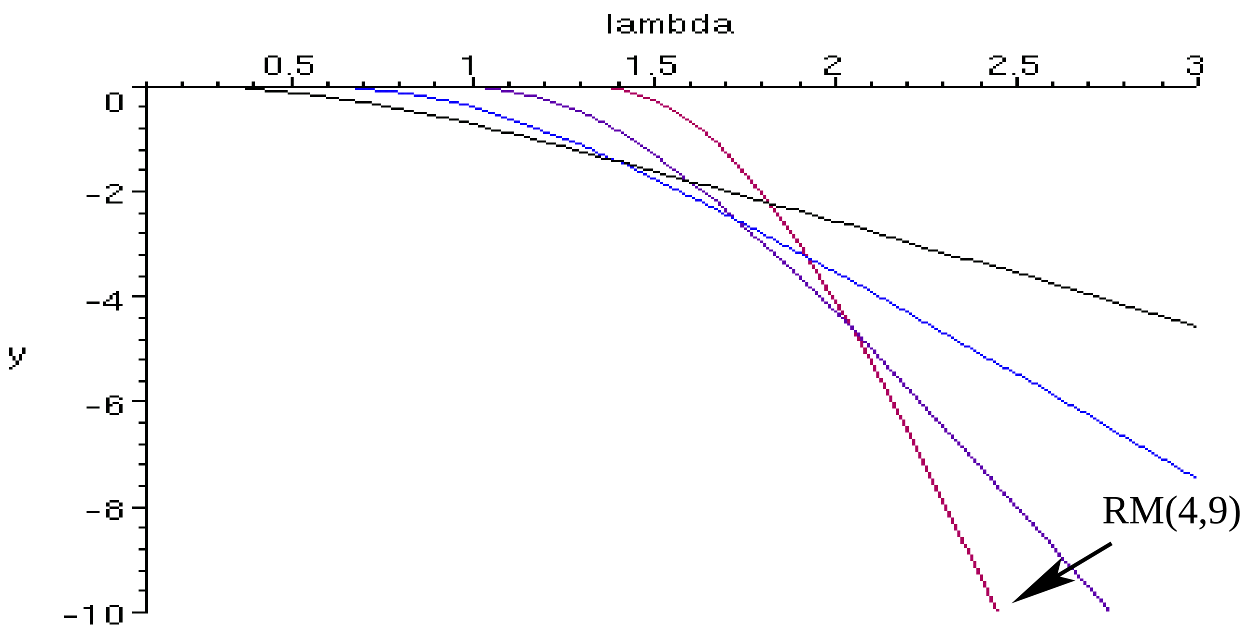
We do this with the codes $\mathrm{RM}(r,m)$ where $(r,m) = (1,3),(2,5),(3,7)$ and $(4,9)$.



Performance Comparison

Performance Questions (cntd)

We do this with the codes $\mathrm{RM}(r,m)$ where $(r,m) = (1,3),(2,5),(3,7)$ and $(4,9)$.



Performance Comparison

- Shannon promises asymptotically excellent benefits of coding.

Conclusions

- Shannon promises asymptotically excellent benefits of coding.
- Using Reed-Muller codes we got an idea of this result.

Conclusions

- Shannon promises asymptotically excellent benefits of coding.
- Using Reed-Muller codes we got an idea of this result.
- An accoustic demonstration helped to physically verify the principle.

Conclusions

- Shannon promises asymptotically excellent benefits of coding.
- Using Reed-Muller codes we got an idea of this result.
- An accoustic demonstration helped to physically verify the principle.
- Thanks for your attention!