**Report. Stepanova Daria**

**Convolutional Codes**

**Part 1**

Goal of this exercise: to study the effects of various parameters (soft or hard decisions, traceback length, code rate, etc.) on the performance of convolutional codes, we will use the built-in MATLAB functions for convolutional encoding and Viterbi decoding.

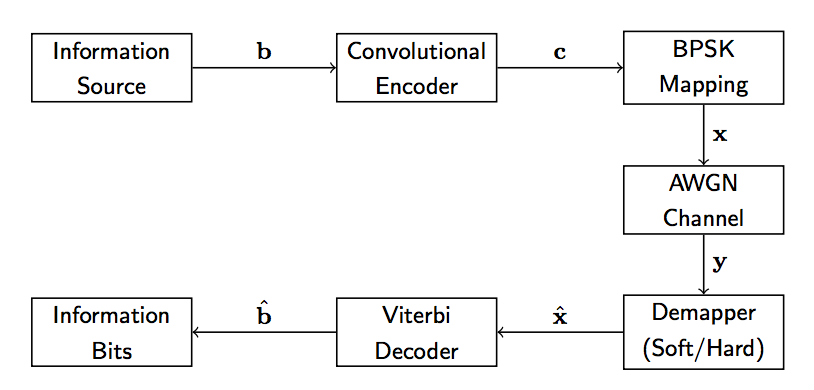


Figure 1 : Convolutional coded system overview.

Tasks:

1. Implement the system depicted in Figure 1 using K = 100 and a hard demapper. Verify that your systems works by making sure that bˆ = b at very high SNR (e.g., 30 dB).

The system was implemented and veryfied. With high SNR bˆ = b. Results of simulation can be seen on Figure2.

1. Study the bit error rate of the above hard input system for an SNR range from −2 to 7 dB (with a 1 dB step) via a Monte-Carlo simulation.

Results of implementation of Monte-Carlo simulation are on Figure2.

1. Extend your Monte-Carlo simulation to study the performance when using a soft demapper. Compare the performance of the hard input Viterbi decoder with the performance of the soft input Viterbi decoder.

There is almost no difference between hard and soft demapper. Hard demapper works better with lower SNR.

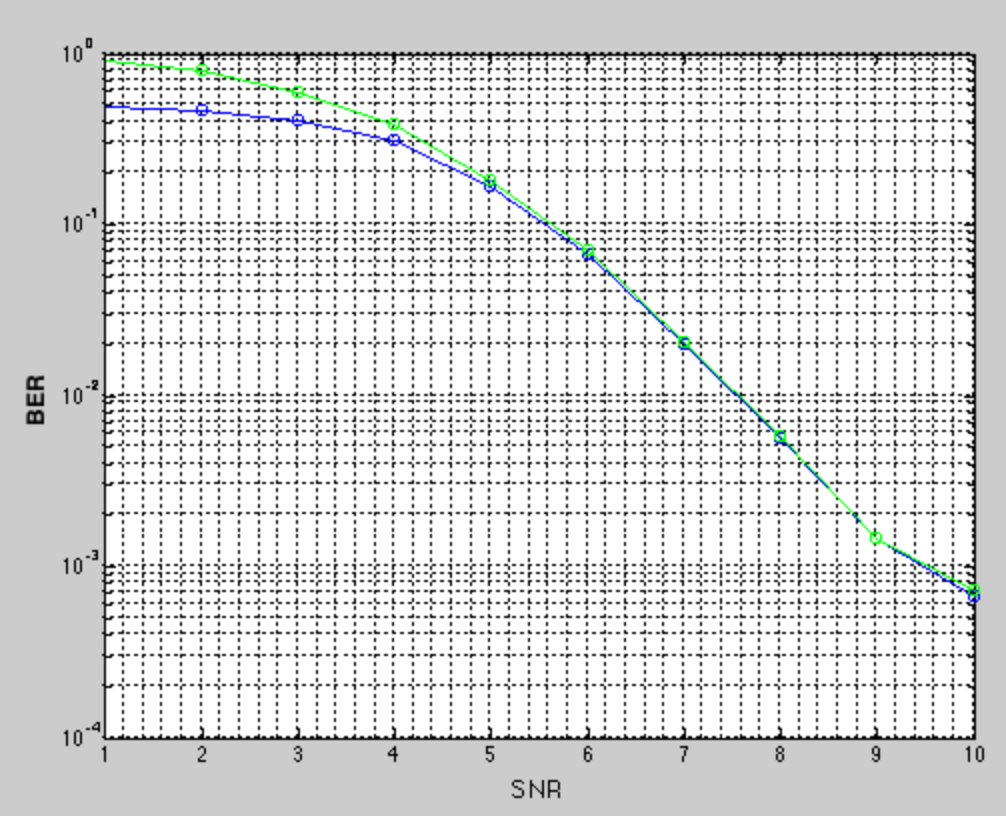


Figure 2: BER for K = 100

blue – hard detector

green – soft detector

1. Increase the number of information bits to K = 1000. What is the effect on the bit error rate? Can you explain this?

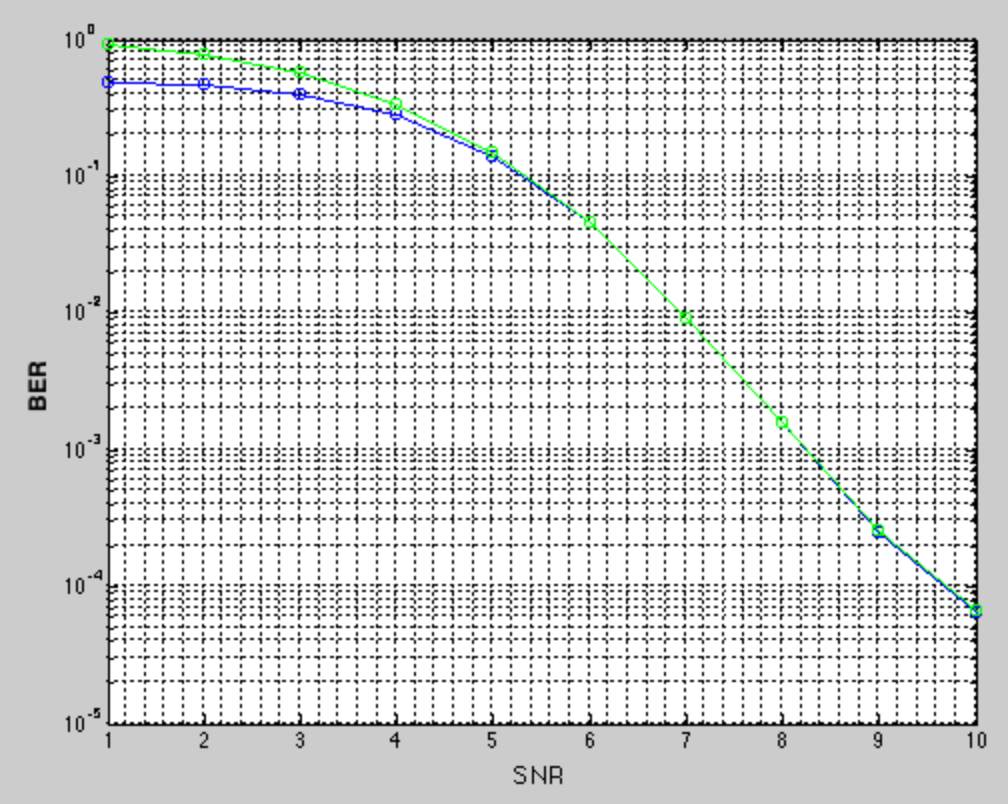


Figure 3: BER for K = 1000

Difference is not significante, because BER doesn't depend on the lenght of vector.

Conclusion: In Section 1 of this exercisу use MATLAB to encode and decode convolutional codes is shown, the effect of using hard and soft decoder input on the performance of the Viterbi decoder sre studied, the effect of the blocklength on the performance of the Viterbi decoder are also studied.

**Part 2**

Goal: to study the effect of some additional parameters on the performance of the Viterbi decoder (the effect of the traceback depth and the memory on the decoding performance and on the decoding complexity; compare the performance of convolutional codes with different rates)

**1 Convolutional Codes with Different Traceback Depth L**

Tasks:

1. Rewrite the simulation framework of Part 1 to perform Viterbi decoding with traceback depths L = 13, 34, 55, n. The case with L = n corresponds to the full Viterbi decoder which first performs the entire forward step before proceeding with the backward step. Use k = 200 information bits and soft decoder inputs.

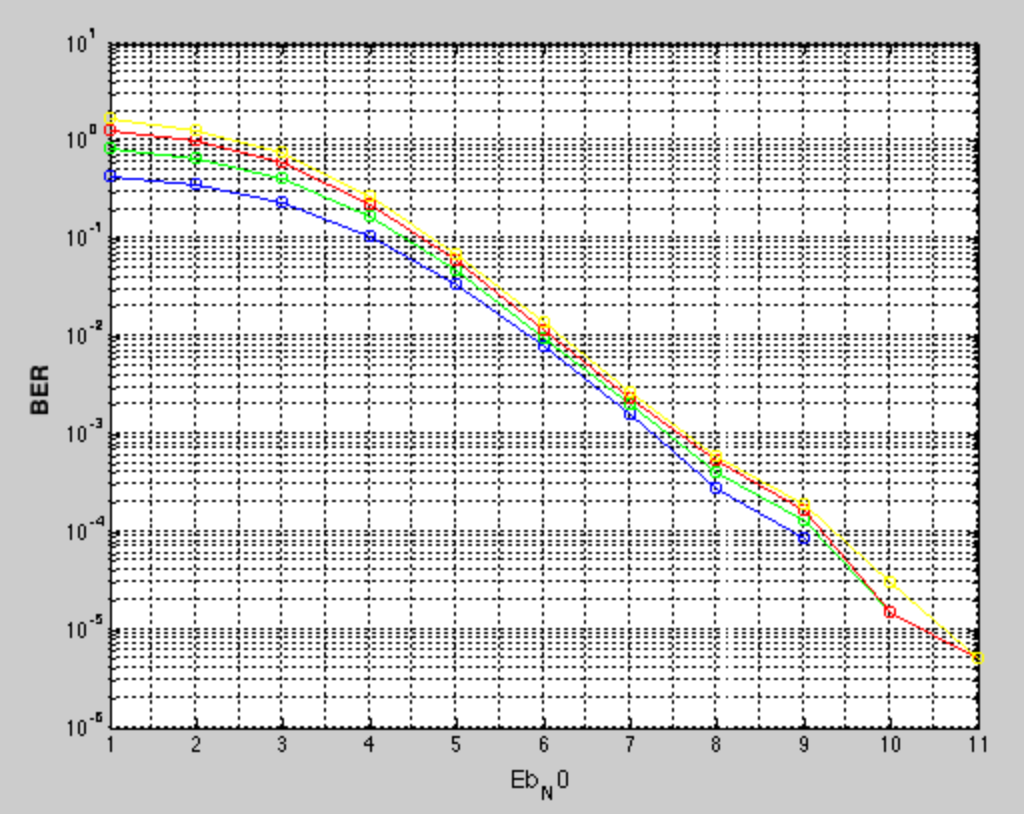


Figure4. BER of different traceback depth

L = 13 – blue

L = 34 – green

L = 55 – red

L = k = 200 – yellow

1. Compare the BER performance of the Viterbi decoder with different traceback depths. What do you observe?

Difference between BER of L=13 to L=k is not considerable. With larger L performance is decreasing. The edge L, when performance will significantly decrease, should be L = 5. Traceback depth is L = 5\*K. Then with K = 1 BER is critical.

1. Measure the average running time of the Viterbi decoder with different traceback depths.

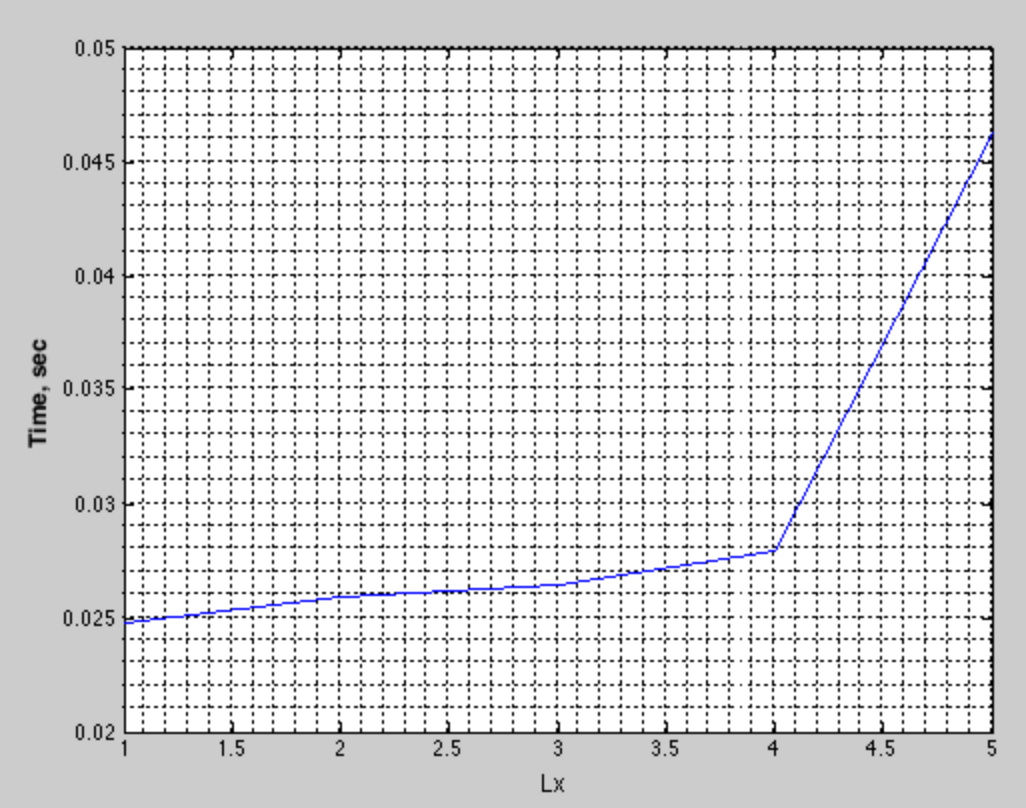


Figure5. Timing of algorithm

L1 = 13

L2 = 34

L3 = 55

L4 = 200

The more steps program makes back, the more time consumin algorithm becomes.

**2 Convolutional Codes with Different Memory K**

Tasks:

1. Rewrite the simulation framework of Part 1 to compare the BER performance of convolutional codes with the following constraint lengths and generator polynomials: (a) K = 3, poly = [7 5] (b) K = 7, poly = [155 117] (c) K = 9, poly = [753 561] Use k = 200 information bits and soft decoder inputs.

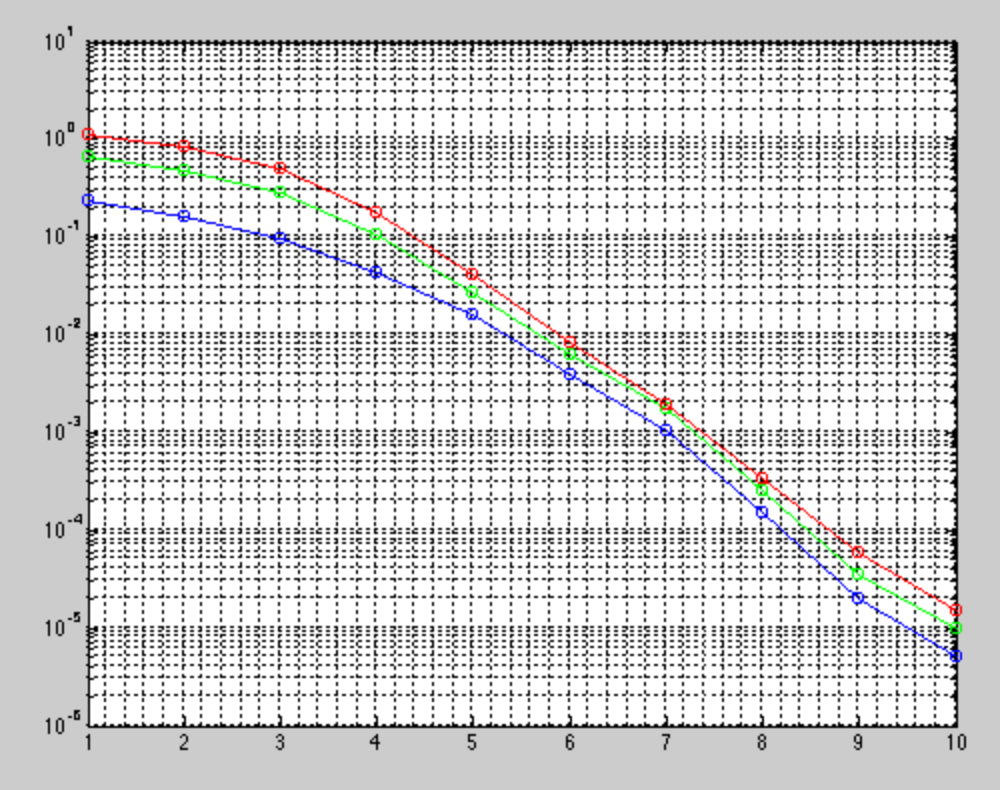


Figure6. BER performance of different constraint lengths and generator polynomials

a) K = 3, poly = [7 5] - red

b) K = 7, poly = [155 117] - green

c) K = 9, poly = [753 561] – blue

The difference between BERs is not considerable. Higher K and poly just increasing the complexity of algorithm, but do not increase much perfomance.

1. Measure the average running time of the Viterbi decoder with different constraint lengths (as in the previous task). How does the running time increase with K?

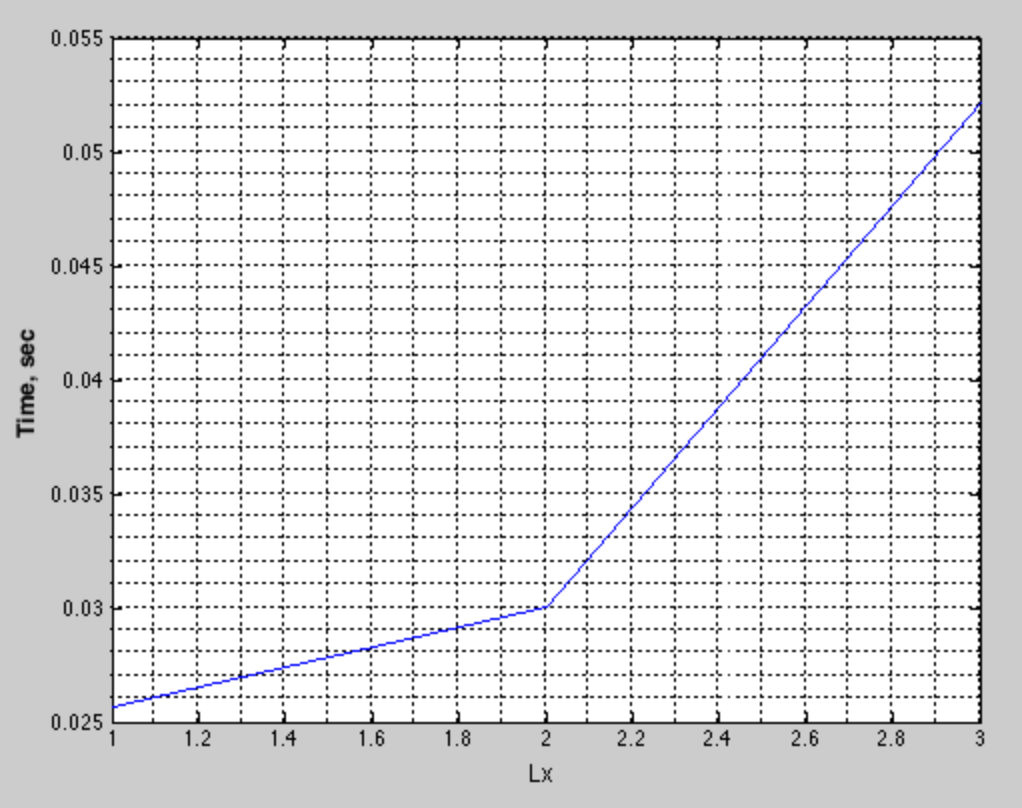


Figure7. Timing of algorithm

L1: poly [7 5] k =3

L2: poly [155 117] k =7

L3: poly [753 561] k =9

The same situation as in previous case – with increasing of K and poly, complexity of calculations and time spent on calculations increases.

**3 Convolutional Codes of Different Rates**

Tasks:

1. For a given Eb/N0 value (given in dB) and code rate R, derive the formula to calculate the complex normal noise variance σ 2 (this is very similar to the SNR-based formula, but you must include the code rate R).

sigma\_kvadr = (P/R)./(10.^(Eb\_No/10)); for Rate = 1.

Later sigma\_kvadr is used for adding noise to signal

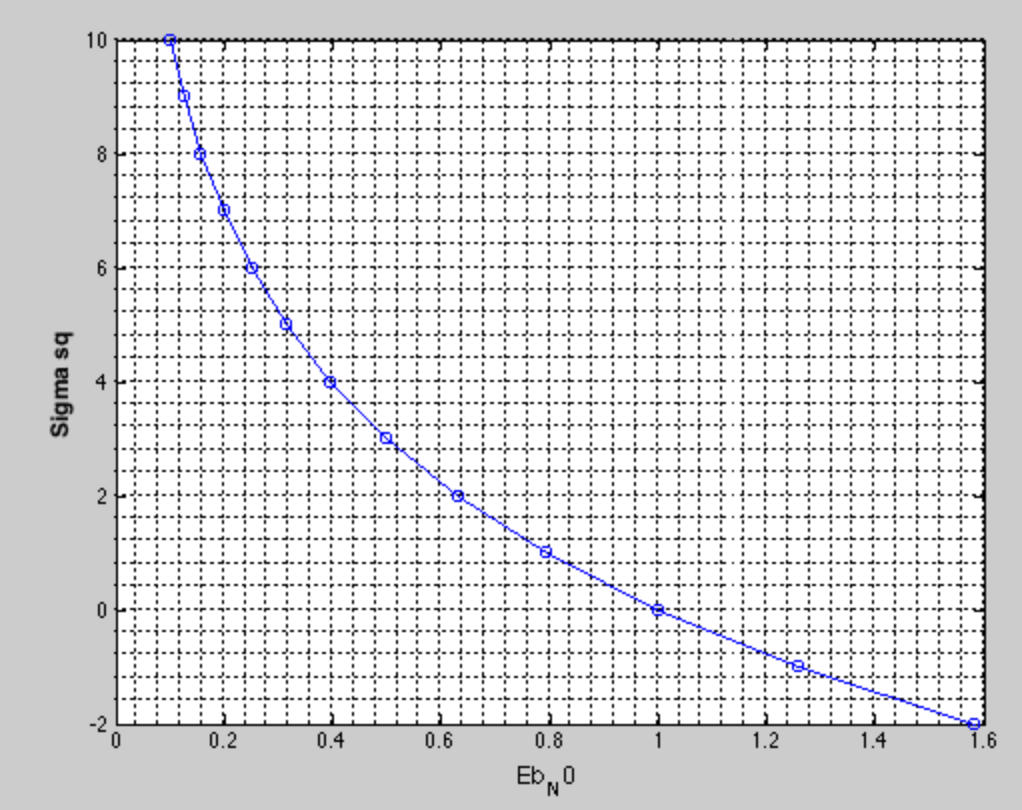


Figure8. sigma\_kvadr

1. Compare fairly the performance of uncoded BPSK transmission (i.e. R = 1) with the performance of a system employing a convolutional code of rate 1/2. Use puncturing of a code of rate 1/2 to also compare it (fairly) with the performance of systems employing convolutional codes of rate 2/3 and 3/4. Use the following parameters:

• Number of information bits per block: k = 1200.

• Eb/N0 range: from −2 dB to 10 dB.

• Use soft decoder inputs.

For R = 1/2 convEnc.PuncturePattern = [1 ;1]; - blue

For R = 2/3 convEnc.PuncturePattern = [1 ;1 ;0 ;1]; - green

For R = 3/4 convEnc.PuncturePattern = [1 ;1 ;0 ;1 ;0 ;1]; - red

Results of simulation are below.

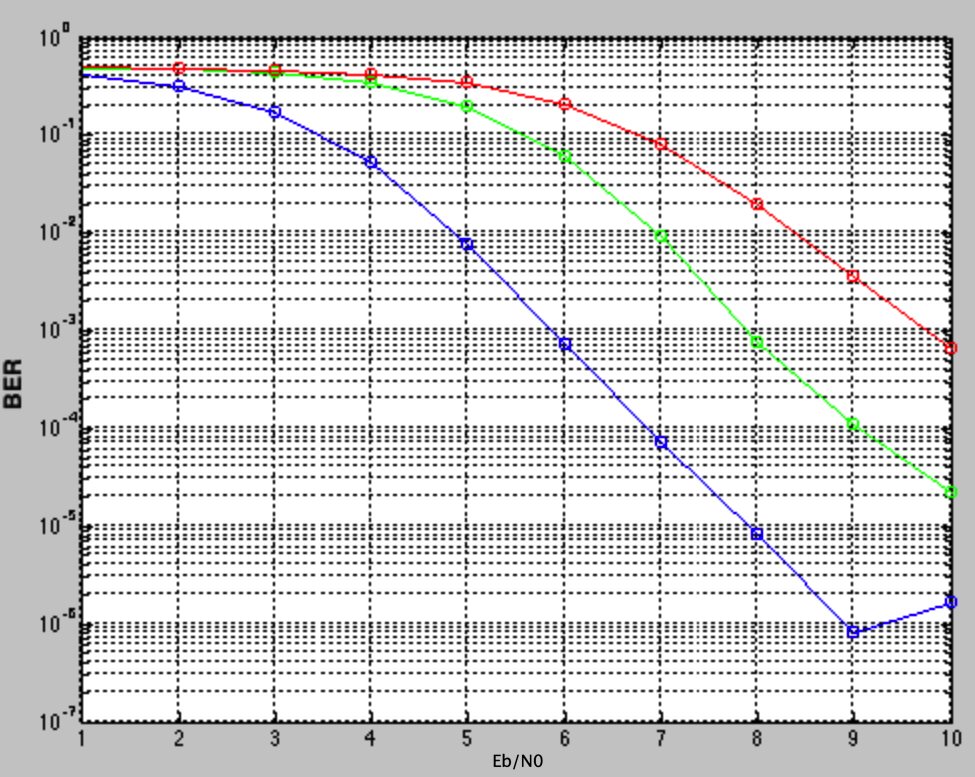


Figure9. BER of different Rates

We can observe, that with BER becomes worse with higher rates.

When comparing codes of different rates, the amount of information bits contained in the transmitted signal x varies. The slower rate – less information could be transmitted in the same time.