Texas A&M University

Department of Electrical and Computer Engineering

ECEN 604 Final project

[December 13 2012]

Author:
Shravya Kunamalla
Marwa Ali

Submitted to:
Prof. Pfister

I Introduction

In this report, a concatenated encoder - decoder is simulated system with an outer Reed-Solomon code and an inner convolutional code. The report is organized as follows: in section II the system model is presented with details for each block of the system and the parameters for the implemented codes. In section III, simulation results using Matlab is presented. last, in appendix I, a detailed description for each function is included.

II System Model

From Fig.[1], the system is described as follows:

- *RS encoder*:
  A Randomly generated message is encoded using (255,239,17) RS code.

- *Interleaver*:
  This message is interleaved on two levels.

- *Convolutional Encoder*:
  The output of the RS encoder is converted from GF(256)to GF(2) and is given as input to the Convolution encoder. Convolution encoding is done by a rate $\frac{1}{2}$. We have attempted different constraint length of 3 and 7. For constraint length of 7, the generator functions

$$\begin{aligned} g^0(x) &= 1 + x + x^2 + x^3 + x^6 \\ &\text{and} \\ g^1(x) &= 1 + x^2 + x^3 + x^5 + x^6, \end{aligned} \tag{1}$$

  see figure in Fig.[2]. This gives a $(133, 171)_8$ generator matrix. For constraint length 3, the generator functions

$$\begin{aligned} g^0(x) &= 1 + x^2 \\ &\text{and} \\ g^1(x) &= 1 + x + x^2 \end{aligned} \tag{2}$$

  , see figure in Fig.[3]. This gives a $(7, 5)_8$ generator matrix.

- *Channel*:
  Transmitted signal pass through the channel. Additive White Gaussian noise is added to the transmitted signal.

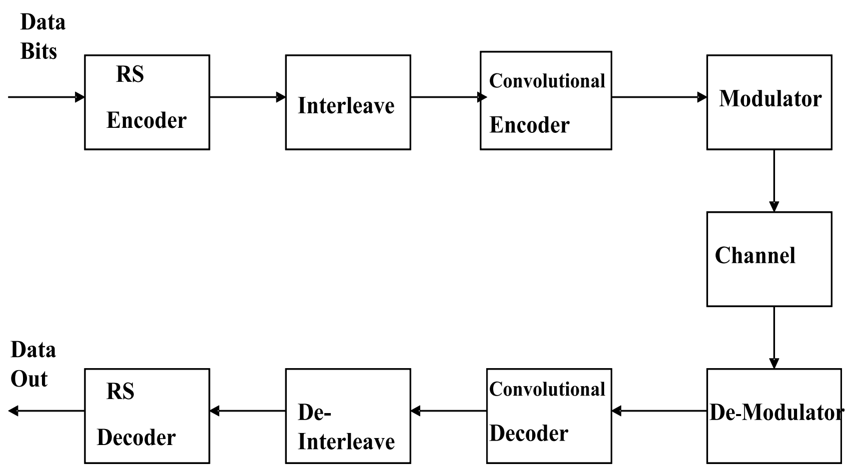

Figure 1: Block diagram of the Implemented coding system

- *Convolutional decoder*:
  The noisy received signal is first decoded by the Viterbi/decoder decoder.

- *Deinterleaver*:
  The output of the Viterbi is deinterleaved. Then the message is changed back from GF(2) to GF(256).

- *RS decoder*:
  The output of the Viterbi goes finally through the RS decoder. The Rs decoder runs Berleykamp Masey algorithm and estimates the message.

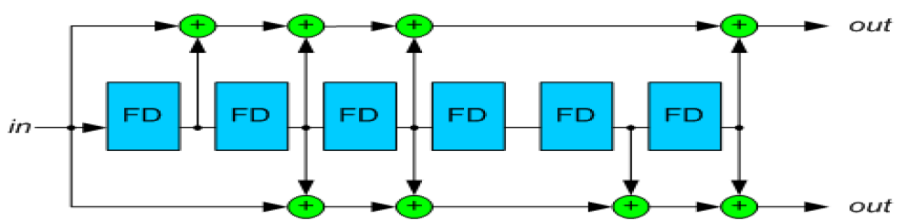

Figure 2: Block diagram of the convolutional code, constraint length=7

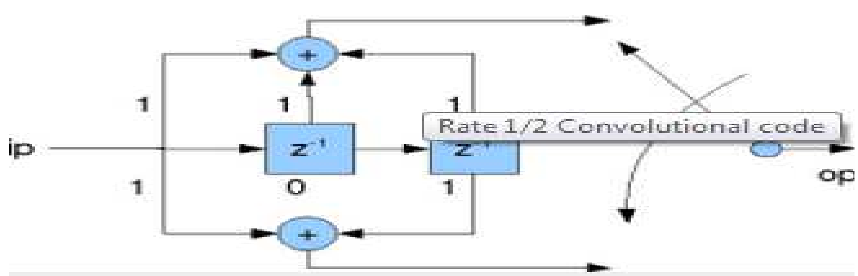

Figure 3: Block diagram of the convolutional code, constraint length=3

[0]Block diagrams of figures 2 and!3 are from: `http://www.1-core.com/library/comm/viterbi/` and `http://www.dsplog.com/2009/01/04/convolutional-code/`

III Simulation and Results

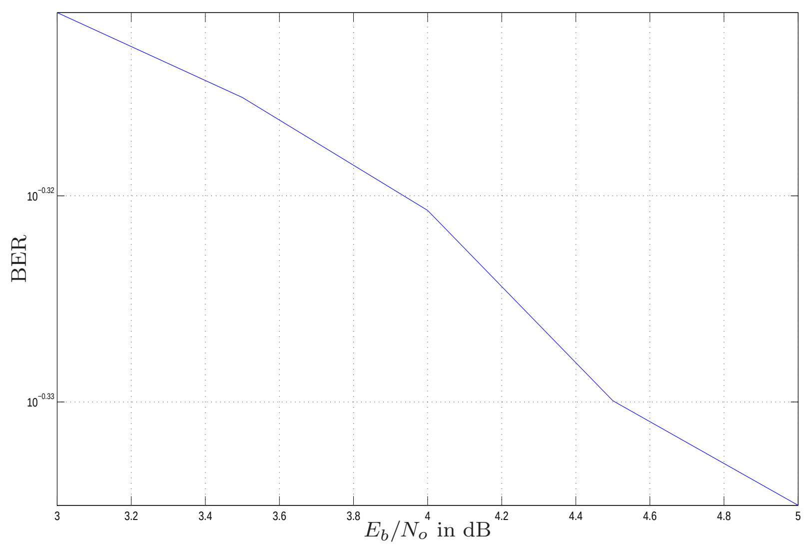

Figure 4: Bit error rate versus E_b/N_o for convolutional encoder-decoder , constraint length=3

In this section, simulation results are proposed and their parameters. In Fig.[4], bit error rate (BER) is plotted versus energy per bit over variance of white noise ($\frac{E_b}{N_o}$) for the outer encoder- decoder system, the convolutional encoder decoder. Fig.[4] is a result of 100,000 realization for a convolutional code of (2,1,3). In Fig.[5], The overall system of concateneted RS-convolutional encoder decoder. This curve obviously needs to be averaged over higher number of realizations for a more precise and smoother look. Montocarlo Prònciple is shown by comparing Fig.[5] and Fig.[7], where Fig.[5] is for a higher number of realizations. The constraint length for the convolutional code in this case is 3. A significant improvement in the BER is shown comparing the overall system - see Fig.[5]- to just the convolutional encoder decoder -see Fig.[4]. The BER has dropped from the order of $10^{-0.2}$ to 10^{-2}.

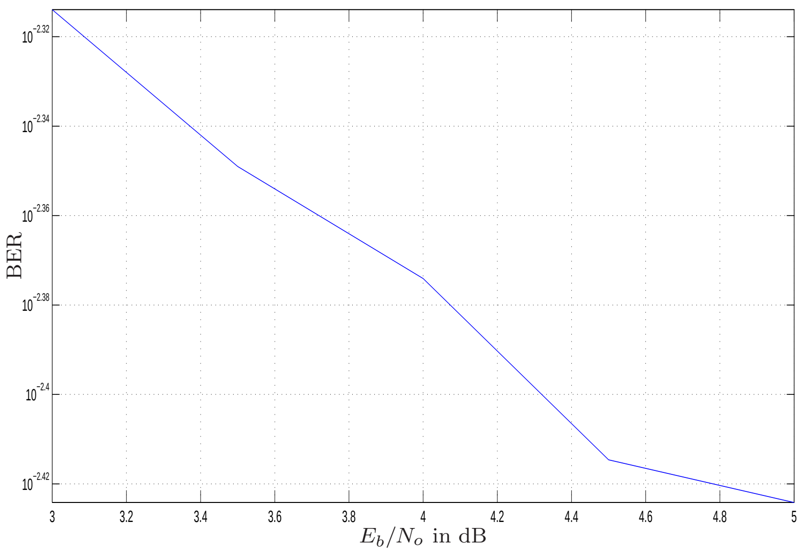

Figure 5: Bit error rate versus E_b/N_o for concatenated RS and convolutional encoder-decoder , constraint length=3

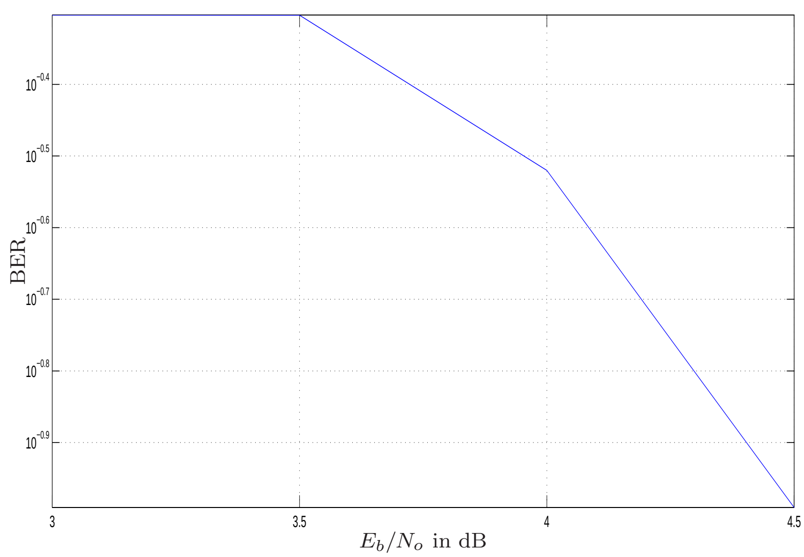

Figure 6: Bit error rate versus E_b/N_o for concatenated RS and convolutional encoder-decoder , constraint length=7

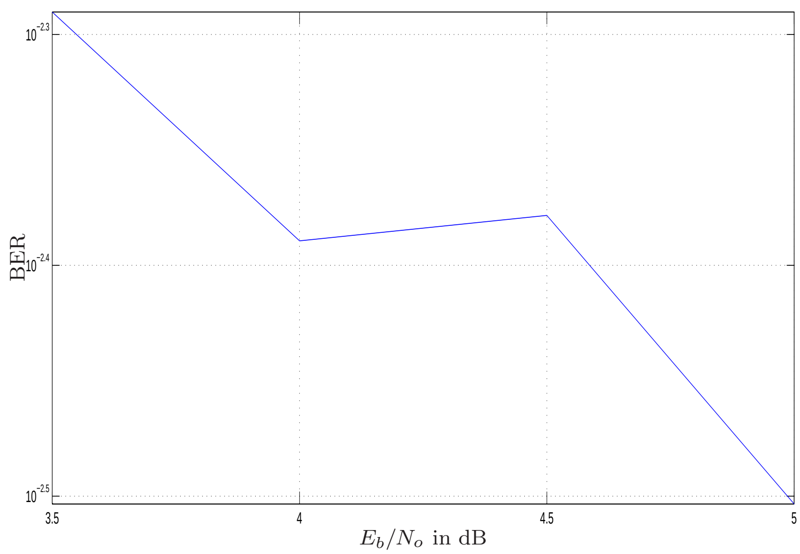

Figure 7: Bit error rate versus E_b/N_o for concatenated RS and convolutional encoder-decoder , constraint length=3

I Appendix

The main file to run the whole system is *main_encoder.m*. . It is worth mentioning that both decoders and encoders are generic for different code rates and parameters and not just the set of paramerters mentioned in this report. A full description of the functions for the RS Encoder and Decoder is as follows:

- Function *gfpower* generates the Galois field matrix. It takes the primitive element a, primitive polynomial qx and the prime number p. Size of the Galois Field matrix is (p^m, m), where m is an integer. In the implemented system, $p = 2$, $a = [10]$, $qx = [100011101]$ and $[Y] = gfpower(a, qx, p)$.

- $A = add(a1, a2, Y)$: $a1$ and $a2$ are the powers of α which we want to add in the field Y.

- $[M] = mult(a, b, Y)$: a, b are the powers of α which we want to multiply in the field Y.

- $P = polymult(a1, a2, Y)$: performs the multiplication $(X + a1)(X + a2)$. Here, *polymult* is used for the calculation of generator polynomial.

- $G = GP(t, Y)$: generator polynomial for Rs encoding. GP takes the error correcting capability t and Y as the input and gives the generator polynomial as the output; in increasing powers of indeterminate variable x).

- $rsencode(mx, gx, Y)$: once the Generator polynomial is found, systematic rs encoding. It takes the message mx, generator polynomial gx and field Y as the input and gives the code polynomial as the output.

- $[qx, rx] = polydiv(ax, bx, Y)$: takes x^{n-k} mx as ax and bx as gx and gives the quotient and remainder when ax is divided by bx.

- RS decoding is done using the Berleykamp Masey algorithm.

- $s = syndrome(rx, t, Y)$: it takes the received vector (output of convolution decoder), number of errors t and the field as the inputs and gives the syndrome as the output. The syndrome is given as the input to the function BKM which gives the error locater polynomial as the output.

- $roots = errloc(lamda, t, Y)$: it takes $lamda$ as the input and gives the exact locations of the errors.

A full description of the functions for the convolutional Encoder and Decoder is as follows:

- $conv\_encoder(G, K, c3)$: encodes the binary message through convolutional encoder. The function takes an arbitrary generating matrix G and K input bits entering the encoder at each clock cycle. The output is in binary form. It checks on the dimensions of the generator matrix G and make sure that the number of column of G is a multiple of K.

- $vit\_decoder(G, K, detected)$: performs the Viterbi algorithm on detected to get the decoded sequence in binary form. The parameters G is the generator polynomial matrix and K is the number of encoder input bits.