ECE 4965/5965 Satellite Communications – Project 2 – Due December 15, 2022

A. First Option: Channel Coding

1. Overview

This project investigates the design and implementation of a channel coder for SatCom applications. The channel coder is often referred to as an FEC, as in forward error corrector. In this project we will look at the design of both the encoder and decoder.

2. Project Write-Up

The project should be written up as a report similar to the kind of report you would turn in at work. The questions below are a minimum set of questions you should answer. If you can think of more questions to answer, that is good. In addition, don't feel constrained to answer questions directly in a question and answer format. You can work the questions into your project write-up.

3. Convolutional Encoder

The code that will be used in this project is the rate ½, constraint length 9, convolutional code as used in IS-95. A block diagram of the coder is shown in the figure below [LEE98],

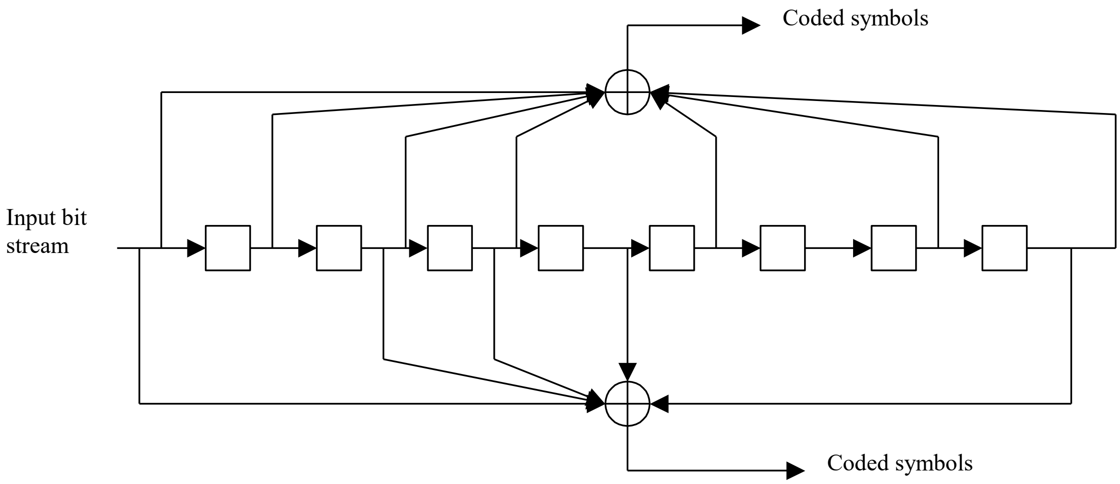

Fig. 1. Block diagram of rate ½, K = 9 convolutional encoder used in IS-95.

Question 1: This code has a constraint length of 9. What is meant by the constraint length of a convolutional code.

Question 2: How many memory elements are used to implement a constraint length, K, code?

Question 3: What is the polynomial representation for this code (show your work)?

Question 4: What is the minimum free distance of this code (see the references)?

Question 5: What are the bounds on the coding gain of this code using the minimum free distance?

Note that in this coder, the + blocks indicate an exclusive-or operation. Also, note that for each input bit, the coder produces 2 output bits, hence, the rate of the code is equal to ½.

For the first part of the project, the encoder needs to be implemented in software using, for example, Matlab, Python, or C++. Include either a copy of your source code, pseudo-code, or a flow chart for the coder in your project report. If you use a simulation package, give pseudo-code.

Question 6: Assume the data rate through the encoder is at 9.6 kBPS. From your software implementation of the encoder, approximately how many MIPS would be needed to implement this coder in software (show how you arrived at this estimate)?

4. Convolutional Decoder

The decoder that will be used in this project is a Viterbi decoder. The Viterbi algorithm is a maximum likelihood decoder for convolutional codes [LEE98], [WIC95]. One way to implement the decoder is to pre-calculate the state transitions. These can be obtained from the coder implementation. For example, in C++ this could be implemented as a structure,

```
struct StateTransition
{
  int nextState[2];   // next state for 0, 1 input bits
  int stateOutputs[2]; // State output bits for 0, 1 input bits
}
```
(1)

Then there would be an array of these state transition structures for each state (where, the number of states is equal to $2^{(\text{constraint length} - 1)}$). Then, a way to implement the states, is to store the states as C++ classes or structures, where each state stores its distance and its history (back trace). For example, each state could be implemented as,

```
class TrellisState
{
  int           _pathBufferPtr;          // The circular buffer pointer for the back trace
  int           _pathLength;             // The size of the back trace buffer
  int           _oldLength;              // Old size to reduce calls to new
  int           _bestInputSymbol;        // Best input symbol for current update
  int           *_backTracePath;         // The back trace of the path
  float         _minimumDistance;        // Minimum distance for current update
  float         _accumulatedDistance;    // The acc distance for the min path to our state
  TrellisState  *_bestInputState;        // The best input state to this one
}
```
(2)

Of course, not all of these variables may be needed for the minimum Viterbi implementation. The state update could then operate as follows,

- Copy the old states into a temporary array of states.
- Update each state as follows:
- For each input state (from the temporary array)
- Calculate the “to” states for each transition (corresponding to a 0 or a 1 input bit),
- For each transition also calculate the distance function between the state transition output (this will be a two bit sequence for a rate ½ encoder), and the actual input bits (you might want to make this distance calculation selectable: soft or hard).
- For each transition “to” state, update the “to” state with the input distance, and the “from” state (for back trace calculation).
- After all states have been cycled through, update the states to only save the best input transition, which has the smallest distance.

After all the input bit pairs have been cycled through the decoder, find the state with the minimum distance, and its back trace should equal the decoded bits.

For the second part of the project, the decoder needs to be implemented in software using, for example, Matlab, Python, or C++. Include a copy of your source code, pseudo-code, or a flow chart for the decoder in your project report.

Question 7: Assume the data rate through the decoder is at 9.6 kBPS. From your software implementation of the decoder, approximately how many MIPS would be needed to implement this coder in software (show how you arrived at this estimate)?

5. Simulation Results

Implement a simulation for testing the BER of a convolutional coder system. A block diagram of the system is shown below.

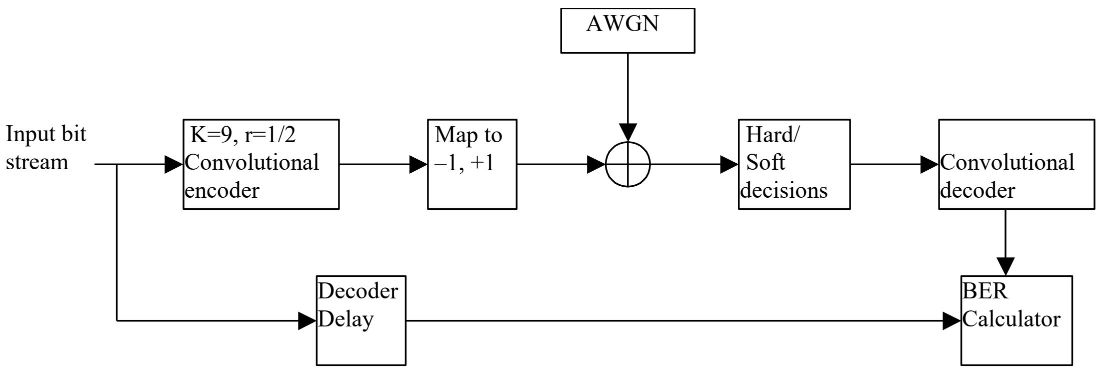

Fig. 2. Block diagram of simulation for assessing BER performance.

5.1 Noise Calibration

We would like to find the coding gain of the convolutional encoder at a BER of 10^{-3}.

Question 8: For the uncoded system, what theoretical SNR is needed to achieve a BER of 10^{-3}? You can use the plots from the lecture.

For the simulated SNR, set the variance of the noise source to achieve this SNR. Note that for the input signal mapped to +/- 1, the input signal power is 1.

Question 9: For the simulated system, what BER was obtained at the theoretical SNR?

Adjust the noise variance to achieve a BER close to 10^{-3}.

Question 10: For the simulated system, what SNR was needed to obtain a BER of 10^{-3}?

Now, include the encoder and decoder, adjust the SNR (noise level) to achieve a BER of 10^{-3}. Record the SNR needed to obtain this BER, and use this SNR to calculate the coding gain of the convolutional coder at 10^{-3}. Note that from [JER94] a rule of thumb for estimating the BER through simulation is to process a number of samples, N, such that,

$$N \approx \frac{10}{p} \tag{3}$$

where p is the probability of a bit error.

Question 11: How many samples were processed in obtaining your BER results. How did you arrive at this number of samples (see above)?

Question 12: What is the coding gain from your simulation results? How does this coding gain compare to the theoretical value? What coding gain is achieved with hard decisions?

B. Second Option – Fading Simulator

1. Overview

This project investigates the design and implementation of a Rayleigh fading channel simulator for SatCom applications. The first part of the project investigates the fading simulator, and the second part of this project uses this fading simulator in testing the performance of a communications system employing an interleaver and a convolutional coder.

2. Project Write-Up

The project should be written up as a report similar to the kind of report you would turn in at work. The questions below are a minimum set of questions you should answer. If you can think of more questions to answer, that is good. In addition, don't feel constrained to answer questions directly in a question and answer format. You can work the questions into your project write-up.

3. Rayleigh Fading Simulator

As discussed in class, one model for simulating a Rayleigh fading channel is based on Clarke's work [CLA68]. In this model, the faded signal is assumed to be composed of a number of signals scattered from obstacles surrounding the mobile user. Due to the motion of the mobile and satellite, the faded signal will have a magnitude spectrum which peaks at plus or minus the Doppler frequency.

Question 1: What is the relationship between the mobile-satellite velocity and the Doppler frequency? In particular, for a mobile-satellite velocity of 10 meters/second, and a carrier frequency of 1.9 GHz, what is the maximum Doppler frequency?

In the first part of the project, you will implement a Rayleigh fading simulator based on Clarke's model. One popular implementation of Clarke's model is Jakes' model, which is described in [DEN93]. In addition, there is an implementation of Clarke's model on page 181 of the text [RAP96]. Discuss which implementation you used, and why. The mobile velocity and the carrier frequency should be inputs to your model. After the model has been implemented, the model will be exercised as described in the questions below.

Question 2: For a sampling frequency of 10 000 Hz, a mobile velocity of 20 meters/second, and a carrier frequency of 1.9 GHz, plot the spectrum at the output of your model. Are the maximum Doppler frequencies at the locations that you expect?

Question 3: Determine the following statistics for your fading simulator using the parameters above:

a) *Level crossing rate: Give the expression for calculating the level crossing rate for –5 dB fades. What is the theoretical level crossing rate for the parameters above? What is the level crossing rate for your simulator given the above parameters?*

b) *Average fade duration: Give the expression for calculating the average fade duration for –5 dB fades. What is the theoretical fade duration given the parameters above? What is the average fade duration for your simulator given the above parameters?*

4. Block Interleaver

In this section you will implement a block interleaver to be used in combating the effects of fading of the signal. The block interleaver is shown in the figure below [LEE98]. Note that this arrangement is not exactly like the one in [LEE98], and that there does not seem to be a standard configuration for block interleavers.

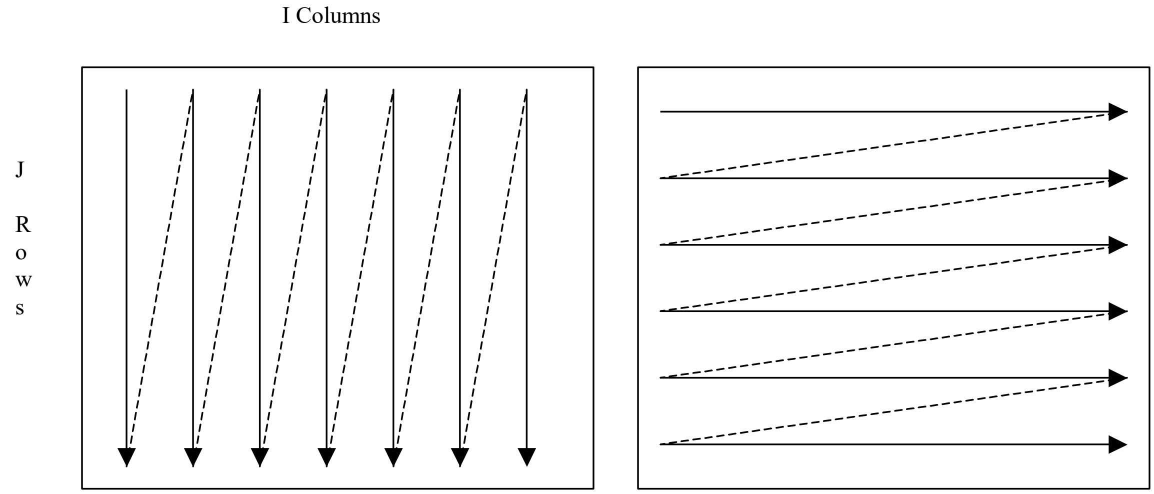

Fig. 3. Block interleaver which writes in by columns, and reads out by rows.

Two common interleaver performance parameters are minimum separation-to-delay S/D and minimum separation-to-memory [LEE98].

Question 1: What is the delay of a block interleaver having I columns and J rows?

Question 2: What is the minimum separation of a block interleaver having I columns and J rows?

Question 3: What is the memory requirements for a block interleaver having I columns and J rows?

Implement the block interleaver as shown above, or another configuration you may prefer.

5. Extra Credit

Try different mobile velocities to check the range of effectiveness of the interleaver.

Try different interleaver sizes.

- etc.

6. References

[LEE98] J.S. Lee and L.E. Miller, *CDMA Systems Engineering Handbook,* Artech House, Boston, MA, 1998.

[RAP96] T.S. Rappaport, *Wireless Communications: Principles & Practice,* Prentice-Hall, Upper Saddle River, NJ, 1996.

[JER94] M.C. Jeruchim, P. Balaban, and K. Shanmugan, *Simulation of Communication Systems,* Plenum Press, New York, NY, 1994.

[DEN93] P. Dent, et. al., "Jakes fading model revisited," Electron. Lett., v. 29, no. 13, pp. 1162-1163, June 1993.

[CLA68] R.H. Clarke, "A statistical theory of mobile radio reception," Bell Syst. Tech. J., vol. 47, July-Aug., 1968, p. 1779.