University of Klagenfurt
Mobile Systems Group

Mobile Communications
Univ.-Prof. Dr.-Ing. Christian Bettstetter

GROUP PROJECT: CONVOLUTIONAL CODING

Your group works for an electronics company and is involved in the development of a new technology for indoor wireless communication in factories. It was decided that convolutional coding would be used to add error correction capabilities to the system. It is now the task of your group to find a convolutional code that has good error correction capabilities and is suitable for the new technology. To do this, write a computer program that implements the encoding, the channel with a simple error model, and the decoding. With this program you simulate the encoding of bit sequences, their transmission, and decoding as well as evaluate the results.

System model

Code rate and encoder memory. In first discussions with the hardware engineers in your company, you decide to use low-complex convolutional encoding: First, the code rate should be $R = 1/2$, which means that, at each discrete time instance $i \in \mathbb{N}_0$, a single information bit u_i fed into the encoder yields two code bits $v_i^{(1)}$ and $v_i^{(2)}$ at the output of the encoder. Second, the number of shift registers required for implementation (i.e., the memory M) should be low, there should be $M = 2$ registers at maximum. It must be decided which convolutional code of this type should be used.

Encoder generator matrix. Doing some literature research, you find a scientific publication from a university group that runs a similar development project. This group uses a convolutional encoder that can be described in terms of the generator matrix $\mathbf{G}_1(D) = (1,\ 1 + D^2)$ with the delay operator D. The publication explains that the encoding process can be described as follows: A sequence of information bits $u(D) = u_0 + u_1 D + u_2 D^2 + \ldots + u_i D^i + \ldots$ is encoded in a way that the sequence of code bits is $\boldsymbol{v}(D) = u(D)\,\mathbf{G}_1(D)$ with $\boldsymbol{v}(D) = \big(v^{(1)}(D),\ v^{(2)}(D)\big)$. Additional literature search reveals a publication that uses a different encoder, specified by $\mathbf{G}_2(D) = (1 + D + D^2,\ 1 + D^2)$. Your group is now to find out which of these two codes provides better error correction over a given channel.

Channel. Each code bit is transmitted independently from other code bits over a channel, which is called binary symmetric channel. This channel causes bit errors: Each bit is "flipped" (from 0 to 1, or vice versa) with a *crossover probability* p, and otherwise remains correct. This means: A transmitted code bit $v = 0$ is detected at the receiver as a bit $w = 0$ with probability $1 - p$ and as $w = 1$ with probability p. In analogy, a bit $v = 1$ is detected as $w = 1$ with probability $1 - p$ and as $w = 0$ with probability p. The channel decoder will ideally correct these bit errors.

Decoder. The channel decoder should use the Viterbi algorithm for decoding, since well-tested and inexpensive chipsets are available. For decoding of the received bit sequence $\boldsymbol{w}(D)$ the Viterbi algorithm is applied at the trellis of the code. The decoded sequence of code bits is denoted by $\hat{\boldsymbol{v}}(D)$. The resulting sequence of information bits is called $\hat{u}(D) = \hat{u}_0 + \hat{u}_1 D + \hat{u}_2 D^2 + \ldots + \hat{u}_i D^i + \ldots$. The communication of a bit u_i is correct if $\hat{u}_i = u_i$.

Project task

Your task is to use computer simulations to compare the error correction capabilities of the two convolutional codes in this arrangement. Specifically, you should investigate the *bit error ratio* (BER), i.e., the fraction of bits that are wrong ($\hat{u}_i \neq u_i$) at the receiver after decoding. The BER is the simulated approximation of the bit error probability $\mathrm{P}\{\hat{u}_i \neq u_i\}$. Study the BER as a function of the crossover probability p. Use a BER range from 0.1 (10^{-1}) down to 0.00001 (10^{-5}). Any programming language can be used. Consider the following subtasks:

- Information bits: Generate a bit sequence using a random generator. Use a bit sequence length of $k = 30$ bits, i.e., $u(D) = u_0 + u_1 D + \ldots + u_{29} D^{29}$ for both codes.

- Encoding: Apply the generator matrix and/or use existing libraries for error-correction coding.

- Channel: Apply the given channel model with random bit flipping (probability p) on the bit sequences.

- Decoding: Implement the Viterbi algorithm on the trellis or use existing libraries for Viterbi decoding. Map $(\hat{v}_i^{(1)}, \hat{v}_i^{(2)})$ to $\hat{u}_i$.

Repeat this experiment for a sufficiently large number of bit sequences to obtain a value for the BER for a given crossover probability p, i.e., BER(p). Then repeat everything for different values of p. As in all simulation-based studies, the statistical confidence of the simulation results is of high importance. For a given set of input parameters, the simulated estimate of $\mathrm{P}\{\hat{u}_i \neq u_i\}$ is accurate if the number of simulated bit errors is large.

Expected results and slides

Each group is expected to summarize the work in a set of five to eight slides describing the approach and main results. The mandatory content is as follows:

- List of active participants and their contributions
- Approach
- Diagram(s) showing the BER(s) as a function of p for both codes. Hint: Use logarithmic scale(s) on the axis/axes. Plot for BER from 10^{-5} to 10^{-1}
- Interpretation of results
- Sample program code

Send your slides to professor Bettstetter as a single PDF file. Use filename `project-convolutional-coding-`*yourname*`.pdf`. All groups will present their work in class (10-minute presentation).