EEE 552 – Project 2 – HDD and SDD and Iterative Decoding of LBCs

Consider the (5,2) code given in class. Simulate, using Monte Carlo simulations, a system that uses this code with binary antipodal signaling over AWGN with hard decision decoding. Plot the codeword error rate versus E_b/N_0. Plot also an exact codeword error rate using the expression that depends on α_i from the standard array. Also include the upperbound that we had in class, which depends only on $d_{\min}$.

Consider also optimal soft decision decoding of the same code. Provide the Monte Carlo simulations, and the upper bound given in class that is based on the union bound.

Finally, consider an iterative decoder for the same code. Simulate using Monte Carlo simulations the codeword error rate. We don't have a theoretical bound or exact performance for this code in this project. Compare the performances of all 3 decoders for the same code.

You will turn in a *typed* report of few pages that includes the MATLAB code used in the appendix. It goes without saying that everyone should write their own code. You should implement a table with syndromes and coset leaders and use that for decoding; do not use any specialized functions from any toolbox in Matlab. Excluding the MATLAB code, the report should not run more than 5-6 pages. When preparing your report, please make sure you pay attention to the following:

1. Make sure you consider enough E_b/N_0 points, over a reasonable range of values. Consult the codeword error rate versus E_b/N_0 plots in the text to see what is an appropriate range.

2. Use legends, or some other means to denote clearly which plot is which, and make sure that all the parameters required to generate the plot is provided in the figure caption or written on the figure.

3. When comparing things, it makes sense to put them on the same plot.

4. The document should be self-contained in that any notation should be defined. Also, anyone reading the document should be able to understand it without reference to any other book/document. A good rule of thumb is to ask if you would be able to understand the document if you read it a year from now. This also means that at least a small introduction explaining what is done in the project, and why it is done, and who has done it before etc.

5. The Monte Carlo simulations should be run enough times to make sure that the error rate curves look smooth. The best way to ensure this is to run a "while" loop for each E_b/N_0, until a certain number of errors (say 200) are made at the output of the decoder. You can then count the number of iterations you had to go through to make those 200 errors, and estimate the codeword error rate by taking the ratio. Clearly, this would mean you would have to run more iterations at higher E_b/N_0.

6. Note that the simulations will take time (at least a few hours in some cases depending on how efficient your MATLAB code is and how fast your computer is). So start early.