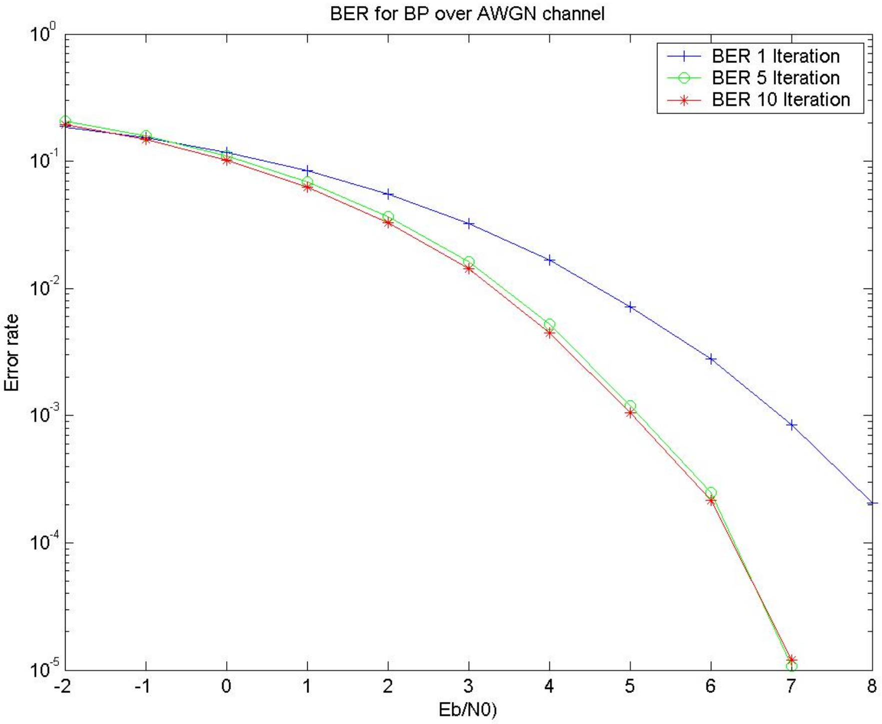
An LDPC Decoder

HANI Mehrpouyan

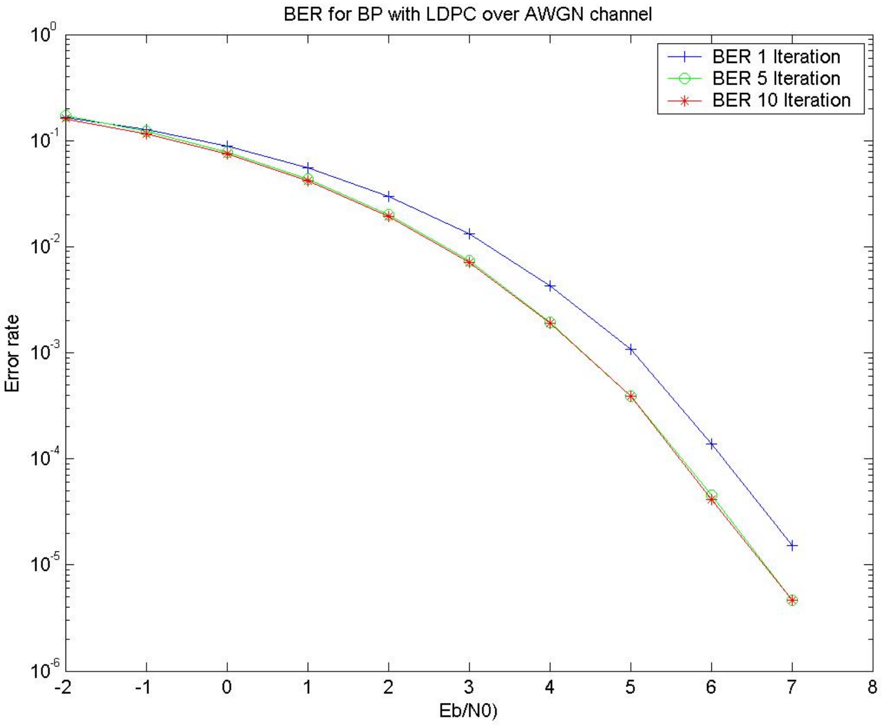
1. syndrom.m Here is the Parity Check Matrix

```
Hb=[    1 1 0 1 0 0 0 1 0 0 0 0 0 0 0;
          0 1 1 0 1 0 0 0 1 0 0 0 0 0 0;
          0 0 1 1 0 1 0 0 0 1 0 0 0 0 0;
          0 0 0 1 1 0 1 0 0 0 1 0 0 0 0;
          0 0 0 0 1 1 0 1 0 0 0 1 0 0 0;
          0 0 0 0 0 1 1 0 1 0 0 0 1 0 0;
          0 0 0 0 0 0 1 1 0 1 0 0 0 1 0;
          0 0 0 0 0 0 0 1 1 0 1 0 0 0 1];
```

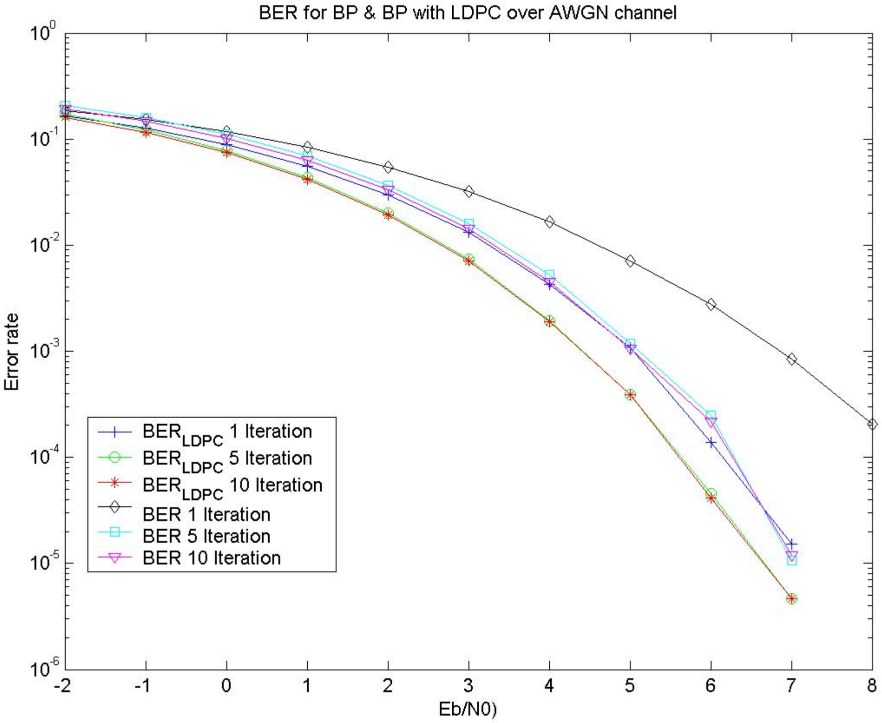
2. The function tanner_gen.m, decoder_spa.m, and BER_FER.m are used for this section
here is the graph



3. tanner_gen_LDPC.m, decoder_spa_LDPC.m, BER_FER_LDPC.m are used in this section.
Here is the resulting Graph



4. here is the resulting picture. To run the whole program use main.m please, it will take a while for the program but the progress is indicated.



a. As expected as the number of iterations is increased the decoding error decreases. This is the due to the fact that more messages are past back and forth, on the other hand it seems like 10 iterations is almost enough since the gain between 5 to 10 iterations is very small.

b. Similar results are observed in the case of the LDPC code.

c. It is interesting to note that the BER curve for the LDPC code performs considerably better in the case of one iteration compared to the BCH code when BP is applied. It is also interesting to note that the overall performance of the LDPC version of the code is almost .3 dB better than the starting BCH code. Thus by adding some redundancy to the parity check of the starting code and modifying the tanner graph we were able to improve the performance of the code and also reduce the number of iterations to achieve the same performance (1iteration version of the LDPC code performs almost as good as the 5 or 10 iteration versions of the BCH code.