Project: LDPC Implementation

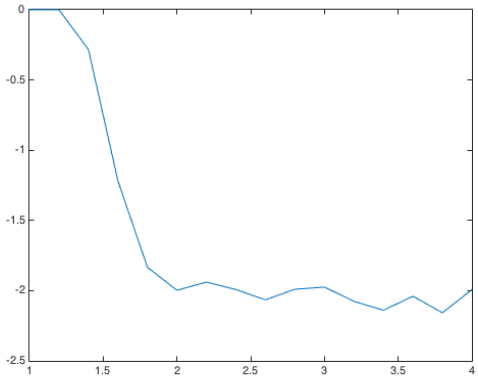
20110727 이준영

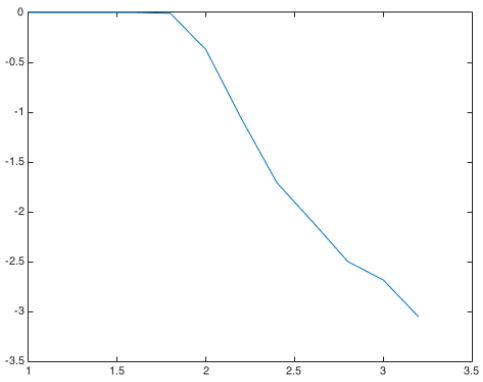
generate_matrix_G_and_H.m 의 스크립트를 통해 G와 H 를 만들어 냅니다. make_matrix_H 함수를 통해, 문제에서 요구한 대로 H 행렬을 만들어 냅니다. 이후, make_matrix_G 함수에서는 Gaussian - Jordan Elemination을 통해 G 행렬을 찾습니다. 이 함수에 해당하는 matlab 내장 함수를 찾았으나, KAIST 라이선스 밖에 있어 웹에서 제공하는 코드를 이용했습니다. G*H' = O이 될 때 까지, iteration이 돌면서 적절한 G와 H를 찾아냅니다. 이 프로젝트에서는 다음 케이스들 중

1. n=100, r=0.5, dv=3
2. n=100, r=0.8, dv=3
3. n=1000, r=0.5, dv=5
4. n=1000, r=0.8, dv=5

여기서 1. 과 2.에 관한 내용만 수행하였습니다. iteration이 너무 많이 돌아서 3. 과 4. 에 대한 내용은 수행하지 못했습니다. 1. 과 2. 에서 얻은 행렬 데이터를 1.mat 과 2.mat 에 각각 저장하였습니다.

이 저장한 행렬을 이용하여, test.m script에서 실제 실험을 수행하였습니다. SNR과 WER의 관계를 알아보는 것을 해보았습니다. generate_codeword 함수를 통해 x를 만들어내었고, get_bipolar_codeword 함수를 통해 0과 1이던 input을 -1 과 1의 bipolar form으로 바꾸었습니다. get_channel_codeword 함수를 통해, 노이즈를 추가하였습니다. ldpc_decode를 통해, ldpc에서 제시한 방법대로 iteration을 돌아 찾아냈습니다. 문제에서 요구한 대로, WER을 계산하였습니다.





왼쪽 그림이 1. 에 해당하는 log10(WER) - SNR 그래프 이고, 오른쪽 그림이 2.에 해당하는 그래프 입니다. Code Rate이 작은 1.에 해당하는 그림이 훨씬 더 빨리 Error Rate이 감소함을 확인할 수 있습니다. 즉, 1.이 더 노이즈에 안정적입니다.