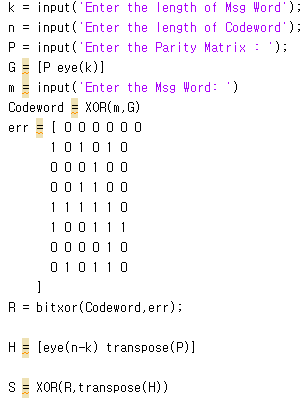
디지털 통신 2차 과제: [시뮬레이터] 채널 코딩

2016707044 김영웅

수업시간에 설명한 채널코딩 과제 소스 코드와 함께 실행 결과 및 분석을 포함한 결과보고서 제출

**index**

1. 6,3 linear block code
2. 7,3 cyclic code
3. Comparison between two code
4. 6,3 linear block code
5. 1 코드 설명

Description 1:

K = length of input msg

n = length of output codeword

p = parity matrix

G = generator matrix

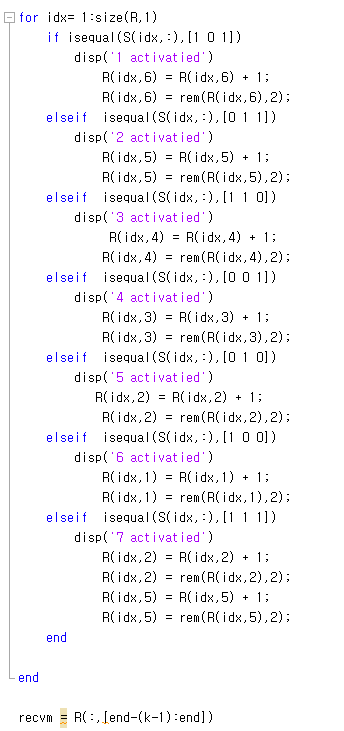
m = message

R = received msg

H = parity check matrix

S = syndrome

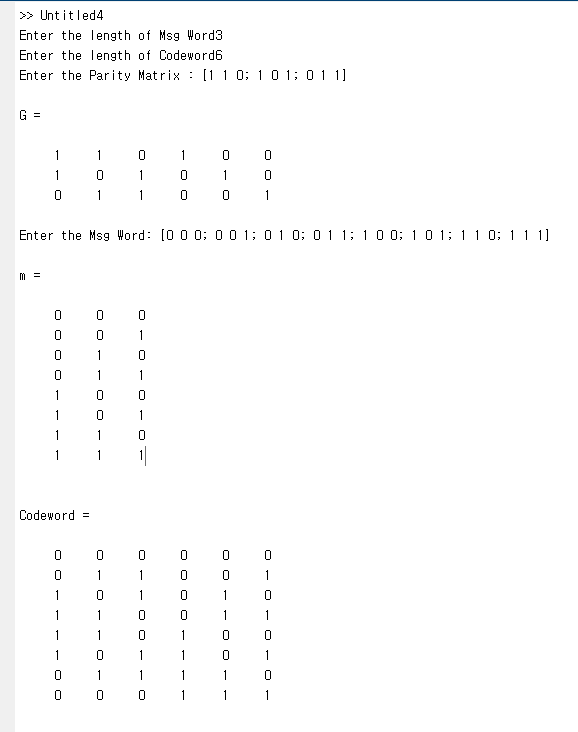
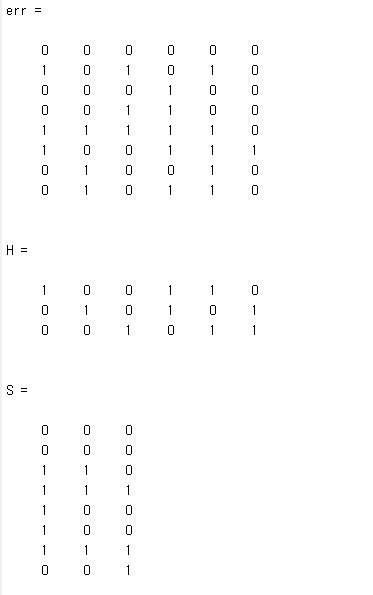
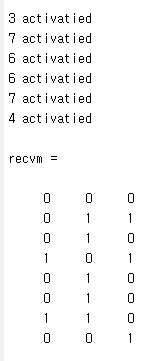
먼저 input 으로 k, n, P, 값을 받은 후에, parity matrix와 k by k 크기의 identity matrix를 결합하여 generator matrix G를 생성한다. 이후 Codeword = m\*G 로, bitxor 함수를 사용하여 codeword를 생성한다 송신부에서 codeword를 보내면 수신부에서는 codeword에 err 가 더해진 값 R로 수신되는 것을 R로 표현하였다. 이후 parity check matrix H를 생성하고, syndrome을 생성하기 위해 수신된 코드 R과 H의 transpose를 곱하였다.



R의 행의 개수만큼 반복하는 for문을 통해 R의 각행에 mapping되는 syndrome 값을 if – else if 문을 통해 구성한다. 각 syndrome은 서로 다른 1bit 에러 6개와 2bit 에러 한 개에 mapping 되어 있다. 에러를 수정하는 방법으로는 에러 bit에 1을 더하고, binary 연산을 하기 위해 2로 나눈 나머지로 해당 bit을 교체한다.

recvm은 수신된 message를 의미하는 변수로 R이 systematic code이기 때문에, R의 오른쪽 끝에서부터 입력 크기 k만큼을 가져와 recvm으로 취한다.

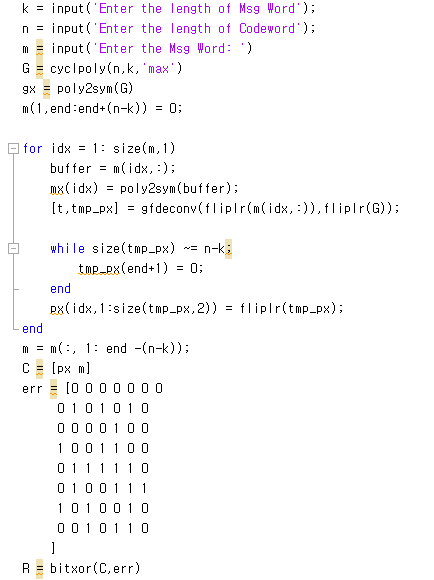
1. 2 실행 결과

Bit error rate = (1+2+2+3+2)/24 = 10/24 =0.416

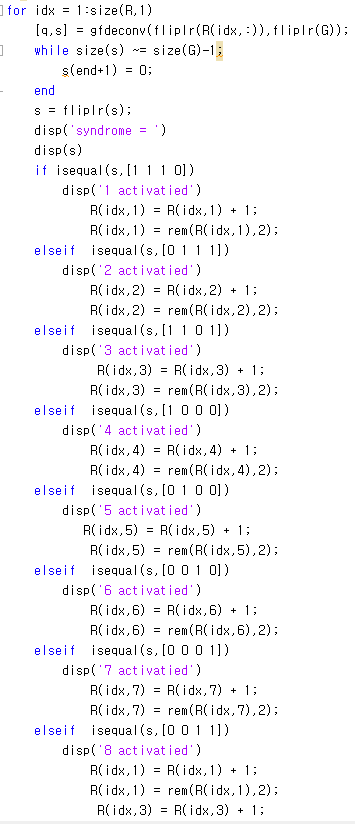
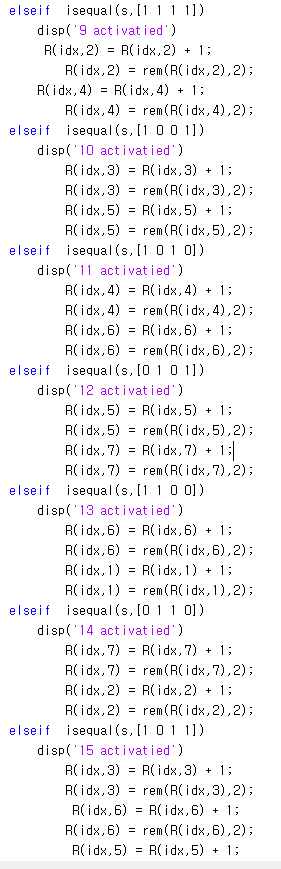
Block error rate =5/8 = 0.625

1. Cyclic code
2. 1 코드 설명

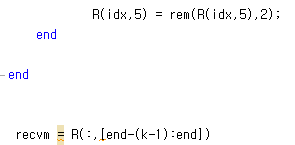


Generator polynomial을 cyclpoly를 통해 생성한다. 이 때 feature를 max로 하여 possible weight이 가장 큰 코드로 선정한다. Message를 나타내는 m을 polynomial에서 X의 n-k승을 곱하는 것을 표현하기 위해 m 행렬에 n-k만큼 행렬의 오른쪽에 0을 더한다. For 문을 통해 행렬 m을 행마다 generator polynomial G와 gfdeconv를 통해 나누고, 나머지를 parity bits로 활용하기 위해 행렬 px에 저장한다. 이후 행렬 m에서 강제로 늘려준 0을 다시 줄이고, parity bits 행렬 px와 결합하여 Codeword C를 만든다. 마지막으로 수신단에서 에러가 더해진 것을 추가하여, 수신된 codeword인 R을 bitxor(C,err)로 만든다.

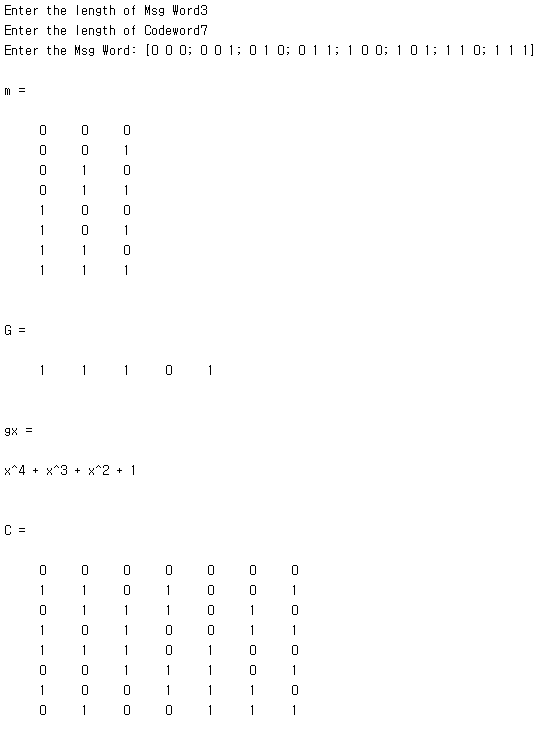
Syndrome bits pattern

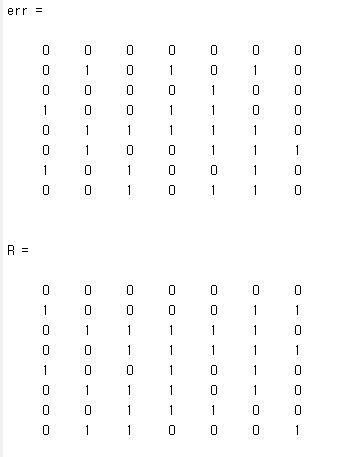
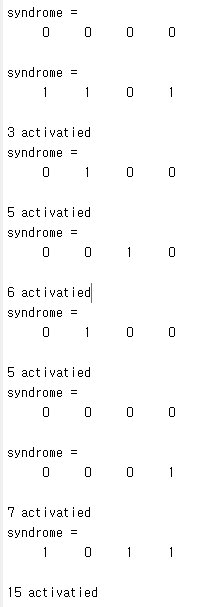
 

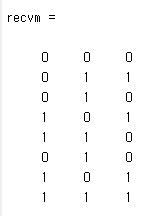
수신한 code R을 G로 나누고, 나머지를 syndrome으로 활용한다. 과정 중에 생략된 0을 패딩 시키고, gfdeconv 에 맞게 좌우 반전 시킨 행렬을 복원한다. Error free를 제외하고 15개의 syndrome bit pattern에 대해서 7개는 1bit error correction, 7개는 2bit error correction, 1개는 3bit error correction으로 mapping 하였다. 이후 for문 탈출 이후에 R에서 message 부분만 추출하여 수신 메시지로 사용한다.



1. 2 실행 결과





Bit error rate = (1+2+1+3+2)/24 = 9/24 =0.375

Block error rate =5/8 = 0.625

1. Comparison between two code

Linear block code의 bit error rate와 block error rate는 각각 41.6%, 62.5%였고

Cyclic code의 경우 bit error rate와 block error rate는 각각 37.5%, 62.5%로 block error rate는 동일 했지만, bit error rate의 경우 cyclic code가 약간 더 에러가 적은 것을 확인할 수 있었다. Cyclic code가 redundancy bit이 4bit으로 syndrome bits이 4bits가 나오기 때문에 6,3 linear block code에 비해 mapping 할 수 있는 error의 경우가 8가지가 더 많이 생긴다. 따라서 이번 과제에서 사용된 임의의 error pattern에서는 소폭의 차이를 보였지만 random error로 여러 회에 걸쳐 실험을 진행 할 경우, 두 코드의 성능 차이가 더 크게 날 것으로 전망한다.