Cyclic Code

Set the code parameters.

n = 15;                % Code length

k = 5;                 % Message length

Create a binary message having length k.

data = randi([0 1],k,1);

Create a generator polynomial for a cyclic code. Create a parity-check matrix by using the generator polynomial.

gpol = cyclpoly(n,k);

parmat = cyclgen(n,gpol);

Create a syndrome decoding tabled by using the parity-check matrix.

trt = syndtable(parmat);

Single-error patterns loaded in decoding table.  1008 rows remaining.

2-error patterns loaded.  918 rows remaining.

3-error patterns loaded.  648 rows remaining.

4-error patterns loaded.  243 rows remaining.

5-error patterns loaded.  0 rows remaining.

Encode the data by using the generator polynomial.

encData = encode(data,n,k,'cyclic/binary',gpol);

Introduce errors in the 4th and 7th bits of the encoded sequence.

encData(4) = ~encData(4);

encData(7) = ~encData(7);

Decode the corrupted sequence. Observe that the decoder has correctly recovered the message.

decData = decode(encData,n,k,'cyclic/binary',gpol,trt);

numerr = biterr(data,decData)

numerr = 0