Guna sai kiran
BT20ECE075
LAB-05

I. Aim:

Error Correction and Detection using Linear block codes Using Matlab.

II. Software Required:

Matlab;

III. Theory:

Linear Block code: k message bits encoded to n code bits i.e., each of 2 k messages encoded into a unique n-bit codeword via a linear transformation. Key property: Sum of any two codewords is also a codeword necessary and sufficient for code to be linear.

(n,k) code has rate k/n. Sometime written as (n,k,d), where d is the minimum Hamming Distance of the code.

Error-control coding can be used for a number of different applications. Codes can be used to achieve reliable communication in presence of interference. In military applications error control codes are used to protect information from intentional enemy interference. In case of satellite communication, there are severe limitations on transmitter power. So with the help of error control coding we can correctly recover very weak messages. Even when the received signal power is close to thermal noise power, error control coding is used to achieve reliable communication. The deep-space communications application has been the arena in which most of the most powerful coding schemes for the power-limited AWGN channel have been first deployed, because the only noise is AWGN in the receiver front end; bandwidth is effectively unlimited; power fractions have huge scientific and economic value; and receiver (decoding) complexity is effectively unlimited.

Generator Matrix of Linear Block Code Linear transformation: C=D.G C is an n-element row vector containing the codeword D is a k-element row vector containing the message G is the kxn generator matrix
If d is the minimum Hamming distance between codewords, we can:

- detect all patterns of up to t bit errors if and only if d : t+1
- correct all patterns of up to t bit errors if and only if d : 2t+1
- detect all patterns of up to tD bit errors while correcting all patterns of tC (¡tD) errors if and only if d : tC+tD+1

IV. Encoding:

A linear block code takes k-bit message blocks and converts each such block all the into n-bit coded blocks. The rate of the code is k/n. The conversion in a linear block code involves only linear operations over the message bits to produce codewords. For concreteness, let's restrict ourselves to codes over F2, so all the linear operations are additive parity computations.
If the code is in systematic form, each codeword consists of the k message bits

D1,D2, ... Dk followed by the n k parity bits P1P2 ... Pnk, where each Pi is some linear combination of the Di's.
Because the transformation from message bits to codewords is linear, one can represent each message-to-codeword transformation succinctly using matrix notation:
$D \cdot G : C$,
where D is a $k \times 1$ matrix of message bits D1D2 ... Dk, C is the n-bit codeword C1C2 ...Cn, G is the $k \times n$ generator matrix that completely characterizes the linear block code, and $\cdot$ is the standard matrix multiplication operation. For a code over F2, each element of the three matrices in the above equation is 0 or 1, and all additions are modulo 2.

If the code is in systematic form, C has the form D1D2 ... DkP1P2 ... Pnk. Substituting this form into Equation 6.1, we see that G is decomposed into a $k \times k$ identity matrix "concatenated" horizontally with a $k \times (n\ \ k)$ matrix of values that defines the code.

The encoding procedure for any linear block code is straightforward: given the generator matrix G, which completely characterizes the code, and a sequence of k message bits D, use Equation 6.1 to produce the desired n-bit codeword. The straightforward way of doing this matrix multiplication involves k multiplications and k 1 additions for each codeword bit, but for a code in systematic form, the first k codeword bits are simply the message bits themselves and can be produced with no work. Hence, we need O(k) operations for each of n k parity bits in C, giving an overall encoding complexity of O(nk) operations.

V. Syndrome Decoding:

Syndrome decoding is an efficient way to decode linear block codes. We will study it in the context of decoding single-bit errors; specifically, providing the following semantics:

- If the received word has 0 or 1 errors, then the decoder will return the correct transmitted message.
- If the received word has more than 0 or 1 errors, then the decoder may return the correct message, but it may also not do so (i.e., we make no guarantees).

It is not difficult to extend the method described below to both provide ML decoding (i.e., to return the message corresponding to the codeword with smallest Hamming distance to the received word), and to handle block codes that can correct a greater number of errors.

VI. Conclusion:

We have successfully find the error correction and detection using linear block code method Using Matlab.