CSE4214 Digital Communications

Chapter 5 Part 2

Channel Coding

Structured Sequence

Structured Sequence

- Structured Sequence
  - Transforming data sequence into “better sequence”, having structured redundancy. The redundant bits can be used for the detection and correction of errors.
- Three types of structure sequence
  - Block
  - Convolutional
  - Turbo

Binary Symmetric Channel

- Binary Symmetric Channel (hard-decision decoding)
  - The input and output alphabet sets consist of the binary elements (0 and 1) and the conditional probabilities *P* are symmetric.

$$P(0|1) = P(1|0) = p \text{ and } P(1|0) = P(0|0) = 1 - p$$

Given that a channel symbol was transmitted, the probability that it is received in error is *p,* the probability of received correctly is (1-*p*)

- The channel symbol error probability

$$P = Q\left(\sqrt{\frac{2E_c}{N_0}}\right)$$

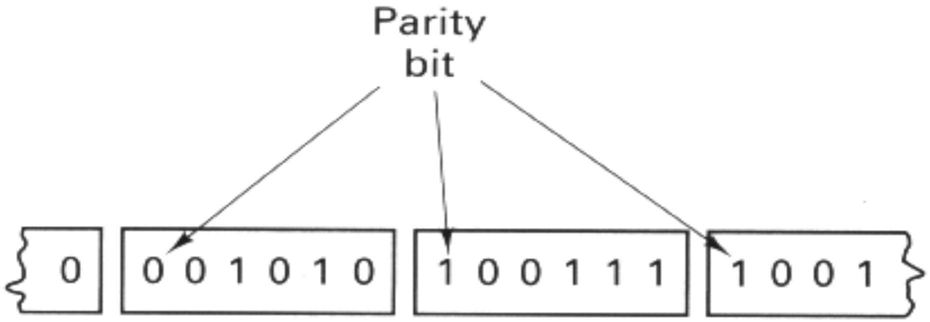
Where $\frac{E_c}{N_0}$ is the channel symbol energy per noise density

Code Rate and Redundancy

- (n, k) block code
  - Source data segmented into blocks of k data bits (message bits)
  - Each block can represent any one of 2^k distinct messages.
  - Encoder transform each k-bit data block into a larger block of n bits called *code bits* or *channel symbols*.
  - The (n-k) bits are called redundant bits, parity bits, or check bits.
- *Code rate* is k/n
- *Redundancy of the code is* (n-k)/k.

Parity-Check Code

- Single-Parity-Check Code
  - A single-parity-check code is constructed by adding a single-parity bit to a block of data bits.
  - The parity bit takes on the value of 1 or 0 as needed to ensure the summation of all the bits in the codeword yields an even or odd result.
  - Even parity – added parity yields an even result
  - Odd parity – added parity yields an odd result



Parity-Check Code

- Probability of *j* errors occurring in a block of *n* symbols

$$P(j,n) = \binom{n}{j} p^j (1-p)^{n-j}, \quad \binom{n}{j} = \frac{n!}{j!(n-j)!}$$

where *p* is probability that a channel symbol is received in error.

- The probability of an undetected error P_{nd} with a block of *n* bits is:

$$P_{nd} = \sum_{j=1}^{\substack{n/2 \text{ (for n even)} \\ (n-1)/2 \text{ (for n odd)}}} \binom{n}{2j} p^{2j} (1-p)^{n-2j}$$

Activity 1

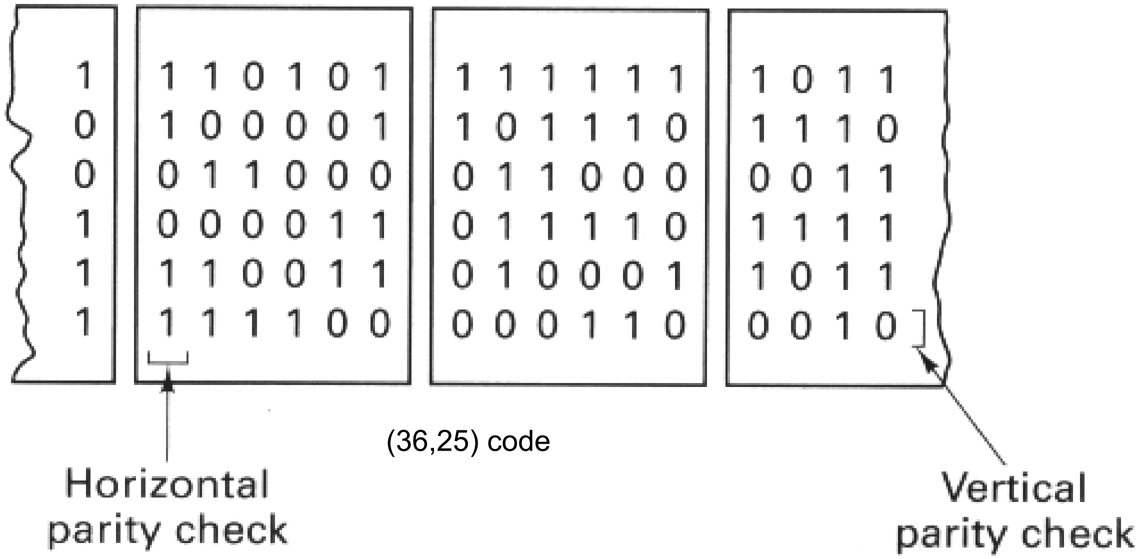
Configure a (4,3) even-parity error-detection code such that the parity symbol appears as the leftmost symbol of the codeword. Which error patterns can the code detect?

Compute the probability of an undetected message error, assume that all symbol error are independent events and that the probability of a channel symbol error is $p = 10^{-3}$.

Rectangular Code

- A Rectangular code (product code) can be thought of as a parallel code structure
  - The message bits are formed by *M* rows and *N* columns
  - A horizontal parity check (row) and a vertical parity check (column), resulting an augmented array of dimension (*M* +1)x(*N*+1), with the rate of rectangular code as

$$\frac{k}{n} = \frac{MN}{(M+1)(N+1)}$$

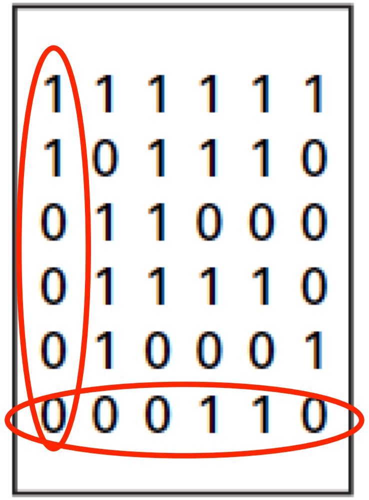


1	1	1	0	1	0	1	1	1	1	1	1	1	1	0	1	1
0	1	0	0	0	0	1	1	0	1	1	1	0	1	1	1	0
0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1
1	0	0	0	0	1	1	0	1	1	1	1	0	1	1	1	1
1	1	1	0	0	1	1	0	1	0	0	0	1	1	0	1	1
1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0

(36,25) code

Rectangular Code (2)

- Rectangular code is capable of correcting a single error located anywhere in block.
- The probability of message error (block error) for a code that can correct all *t* and fewer error patterns :



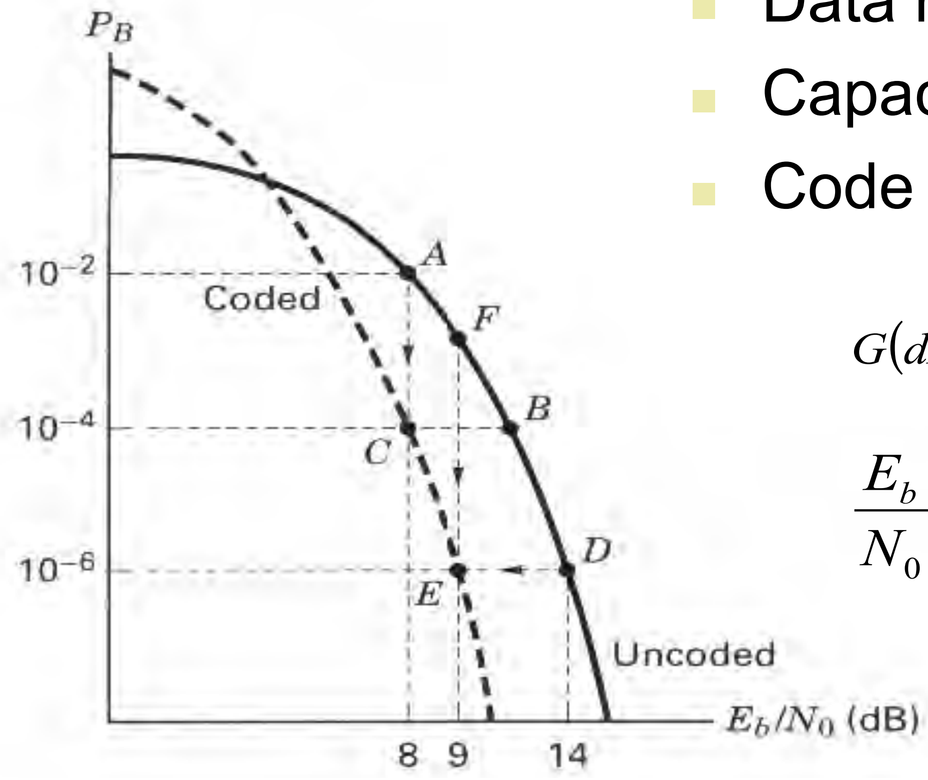
1	1	1	1	1	1
1	0	1	1	1	0
0	1	1	0	0	0
0	1	1	1	1	0
0	1	0	0	0	1
0	0	0	1	1	0

(36,25)

$$P_M = \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

Why Use Error-Correction Coding?

- Error Performance vs Bandwidth
- Power vs Bandwidth
- Coding Gain
- Data rate vs Bandwidth
- Capacity vs Bandwidth
- Code Performance at Low Values of E_B/N_0



$$G(dB) = \left(\frac{E_b}{N_0}\right)_u (dB) - \left(\frac{E_b}{N_0}\right)_c (dB)$$ → Coding gain

$$\frac{E_b}{N_0} = \frac{P_r}{N_0}\left(\frac{1}{R}\right)$$

where R is data rate, P_r is received power, N_0 is noise power in a 1-Hz BW

Linear Block Codes

Vector Space

- The set of all binary *n*-tuples, V_n, is called a vector space over the binary field of two elements (0 and 1).
- The binary field has two operation: addition and multiplication, and the results are in the same set of two elements.

Addition	Multiplication
0+0=0	0 . 0 = 0
0+1=1	0 . 1 = 0
1+0=1	1 . 0 = 0
1+1=0	1 . 1 = 1

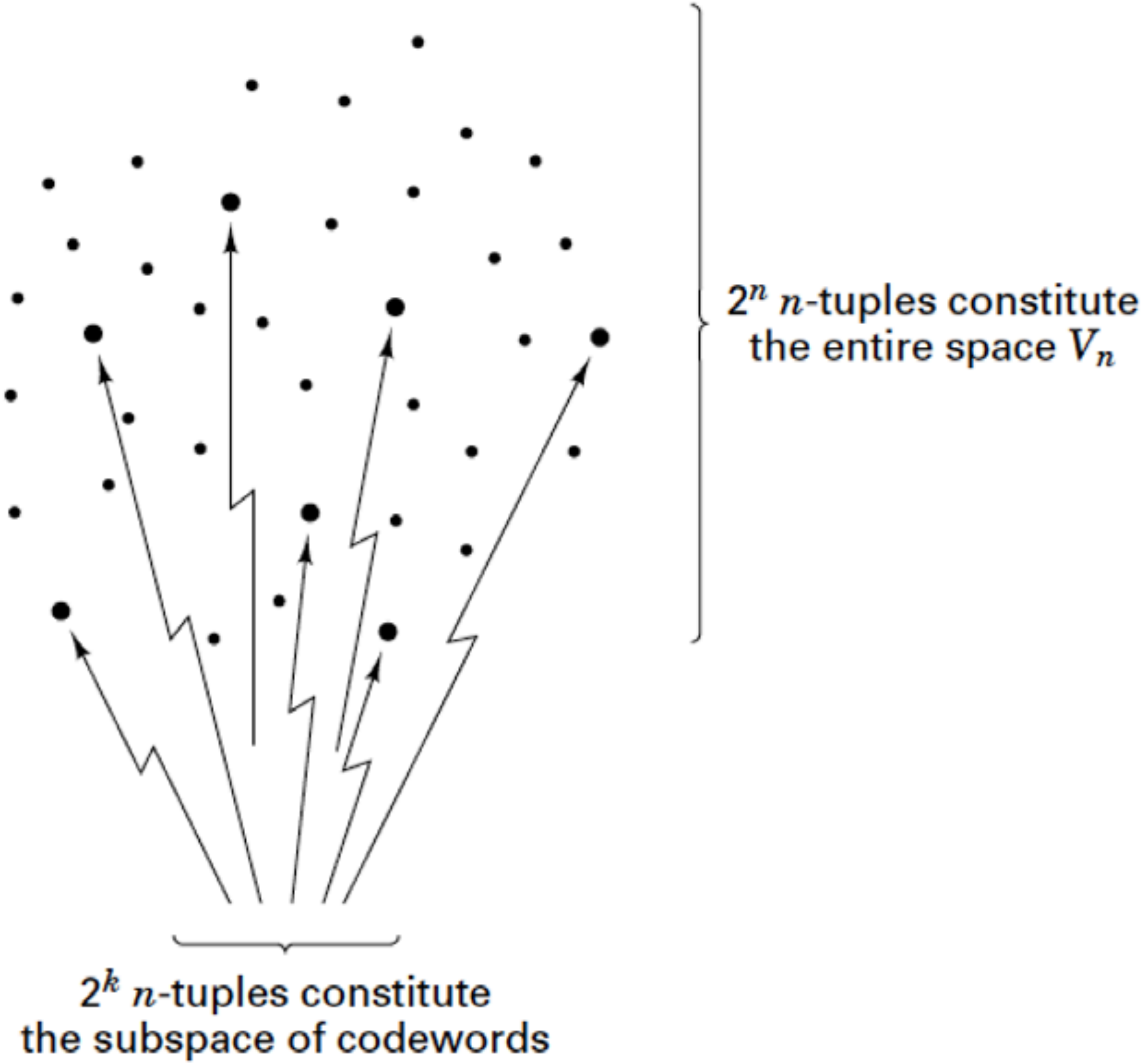
Vector Subspaces

- A subset *S* of the vector space V_n is called a subspace if the following two conditions are met:
  - The all-zeros vector is in *S*
  - The sum of any two vectors in *S* is also in *S* (known as the closure property).
- Example: If there is vector space V_4 that is populated by the following 2^4= sixteen 4-tuples

0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1011 1100 1101 1110 1111

Subset V_4 that forms a subspace is
0000 0101 1010 1111

Vector Subspaces



2^n n-tuples constitute
the entire space V_n

2^k n-tuples constitute
the subspace of codewords

Linear Block Code Example

- (6, 3) code → n=6, k=3
- Message vector: $2^k=2^3=8$
- 6-tuples: $2^n=2^6=64$ in the V_6 vector space.

Message vector	Codeword
000	000000
100	110100
010	011010
110	101110
001	101001
101	011101
011	110011
111	000111

Generating Codeword

- If *k* is large, a table look-up implementation of the encoder becomes prohibitive.
  - Example: (127,92) code have approximately $5X10^{27}$ code vectors.
- Since a set of codewords that forms a linear block code is a *k*-dimensional subspace of the *n*-dimensional binary vector space, it is always possible to find a set of *n*-tuple that can generate all the 2^k codeword of the subspace.
- The smallest linearly independent set that spans the subspace is called a basis of the subspace.

Generating Codeword (2)

- Each of set of 2^k codewords (U) can be generated by

$$U = mG$$

where $\boldsymbol{m}=[m_{1,}\ m_2, \ldots, m_k]$ is a sequence of k message bits.

- ***G*** is a generator matrix

$$G = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & & & \\ V_{k1} & V_{k2} & \cdots & V_{kn} \end{bmatrix}$$

Activity 2

Generate a codeword for message vector [1,1,0] (U_4) in a (6,3) code if the generator matrix G is given by

$$G = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Systematic Linear Block Code

- A systematic (*n*, *k*) linear block is a mapping from a *k*-dimensional message vector to an *n*-dimensional codeword in such a way that part of the sequence generated coincides with the *k* message digits and the remaining (*n*-*k*) digits are parity digits.
- The generator matrix of a systematic linear code is:

$$G = \left[P \mid I_k \right] = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & \vdots & \\ p_{k1} & p_{k2} & \cdots & p_{k(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

where ***P*** is the parity array portion of the generator matrix and ***I*** is the *k* X *k* identity matrix.

Systematic Linear Block Code (2)

- Each codeword is expressed as:

$$u_1, u_2, \ldots, u_n = \left[m_1, \ldots, m_k \right] \times \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1(n-k)} & 1 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{2(n-k)} & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & \vdots & \\ p_{k1} & p_{k2} & \cdots & p_{k(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$U = \underbrace{p_1, p_2, \ldots, p_{n-k}}_{\text{parity bits}}, \underbrace{\mathrm{m}_1, \mathrm{m}_2, \ldots, \mathrm{m}_\mathrm{k}}_{\text{message bits}}$$

Activity 3

For the (6,3) code, the codewords are described as follows:

$$U = \left[m_1, m_2 m_3 \right] \times \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Find u_1, u_2, …, u_6.

Parity Check Matrix

- Parity-check matrix (**H** matrix) will enable to decode the received vectors. The components of H matrix are written as

$$H = \begin{bmatrix} I_{n-k} & \mid & P^T \end{bmatrix}$$

$$H^T = \begin{bmatrix} I_{n-k} \\ -- \\ P \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ p_{11} & p_{12} & \cdots & p_{1,(n-k)} \\ p_{21} & p_{22} & \cdots & p_{2,(n-k)} \\ \vdots & & & \\ p_{k1} & p_{k2} & \cdots & p_{k,(n-k)} \end{bmatrix}$$

Parity Check Matrix

- The product **UH**^{T} of each codeword **U**, generated by G and **H**^{T} matrix, yields:

$$UH^T = p_1 + p_1, p_2 + p_2, \ldots, p_{n-k} + p_{n-k} = 0$$

where the parity bits p_1, p_2, …, p_{n-k} are defined by

$$p_1 = m_1 p_{11} + m_2 p_{21} + \ldots + m_k p_{k1}$$

$$p_2 = m_1 p_{12} + m_2 p_{22} + \ldots + m_k p_{k2}$$

$$\ldots$$

$$p_{n-k} = m_1 p_{1(n-k)} + m_2 p_{2(n-k)} + \ldots + m_k p_{k(n-k)}$$

Syndrome Testing

- Let $\boldsymbol{r}=r_1,r_2,\ldots,r_n$ be a received vector resulting of transmitting of $\boldsymbol{U}=u_1,u_2,\ldots,u_n$ → $\boldsymbol{r}=\boldsymbol{U}+\mathbf{e}$

  e is an error vector introduced by the channel.
- The syndrome of ***r*** is defined as

$$\mathrm{S} = \mathrm{r}\mathrm{H}^{T}$$

- The syndrome is the result of a parity check performed on ***r*** to determine whether ***r*** is a valid member of the codeword set
  - If ***r*** is a member, the syndrome **S** has zero value.
  - If ***r*** contains detectable errors, **S** has some nonzero value.
  - If ***r*** contains correctable errors, **S** has some nonzero value that can earmark the particular error pattern.

Syndrome Testing

- Given $S = rH^T$ and $r = U + e$, we have

  $$S = UH^T + eH^T$$

  Since $\mathbf{UH}^T$=0, → $S = eH^T$

- Note the following two required properties of the parity check matrix:
  - No column of **H** can be all zeros, or else an error in the corresponding codeword position would not affect the syndrome and would be undetectable.
  - All columns of **H** must be unique. If two columns of **H** were identical, errors in these two corresponding codeword positions would be indistinguishable.

Activity 4

Suppose that codeword U=101110 from the (6,3) code in Activity 3 is transmitted and the vector **r**=001110 is received; i.e. the leftmost bit is received in error. Find the syndrome vector $S = \mathbf{r}\mathbf{H}^T$ value and verify that it is equal to $\mathbf{eH}^{T}$.

Error Correction

- Standard array – 2^n n-tuples that represent possible received vectors in an array
  - The first row contain all the codewords starting with the all-zeros codeword, and the first column contains all the correctable error patterns.
  - Each row is called *coset*
  - Row consists of an error pattern in the first column is called *coset leader*.
  - Coset is short for “a set of numbers having a common feature”.

Error Correction (2)

- The standard array format as follows

$$
\begin{array}{cccccc}
\mathrm{U}_1 & \mathrm{U}_2 & \cdots & \mathrm{U}_i & \cdots & \mathrm{U}_{2^k} \\
\mathrm{e}_2 & \mathrm{U}_2+\mathrm{e}_2 & \cdots & \mathrm{U}_i+\mathrm{e}_2 & \cdots & \mathrm{U}_{2^k}+\mathrm{e}_2 \\
\mathrm{e}_3 & \mathrm{U}_2+\mathrm{e}_3 & \cdots & \mathrm{U}_i+\mathrm{e}_3 & \cdots & \mathrm{U}_{2^k}+\mathrm{e}_3 \\
\vdots & \vdots & & & & \\
\mathrm{e}_j & \mathrm{U}_2+\mathrm{e}_j & \cdots & \mathrm{U}_i+\mathrm{e}_j & \cdots & \mathrm{U}_{2^k}+\mathrm{e}_j \\
\vdots & \vdots & & & & \\
\mathrm{e}_{2^{n-k}} & \mathrm{U}_2+\mathrm{e}_{2^{n-k}} & \cdots & \mathrm{U}_i+\mathrm{e}_{2^{n-k}} & \cdots & \mathrm{U}_{2^k}+\mathrm{e}_{2^{n-k}}
\end{array}
$$

- Each coset consists of 2^k n-tuples, therefore there are $(2^n/2^k)=2^{n-k}$ cosets
- Codeword U_i ($i=1,\ldots,2^k$) is transmitted over a noisy channel, resulting a corrupted vector U_i+e_j.
- If the error pattern e_j caused by the channel is a coset leader, the received vector will be decoded correctly into the transmitted codeword U_i.

Error Correction (3)

- Syndrome of a coset
  - The syndrome of this n-tuple can be written as

$$\mathrm{S} = (\mathrm{U}_i + \mathrm{e}_j)\mathrm{H}^T = \mathrm{U}_i\mathrm{H}^T + \mathrm{e}_j\mathrm{H}^T = \mathrm{e}_j\mathrm{H}^T$$

- Error Correction Decoding procedure
  - Calculate the syndrome of r using $S=rH^T$
  - Locate the coset leader (error pattern) e_j, whose syndrome equals rH^T
  - This error pattern is assumed to be the corruption caused by the channel
  - The corrected received vector, or codeword, is identified as $U=r+e_j$. We retrieve the valid codeword by subtracting out the identified error.

Error Correction (4)

- Locating the error pattern
  - Standard array for a (6,3) code

000000	**110100**	**011010**	**101110**	**101001**	**011101**	**110011**	**000111**
000001	110101	011011	101111	101000	011100	110010	000110
000010	110110	011000	101100	101011	011111	110001	000101
000100	110000	011110	101010	101101	011001	110111	000011
001000	111100	010010	100110	100001	010101	111011	001111
010000	100100	001010	111110	111001	001101	100011	010111
100000	010100	111010	001110	001001	111101	010011	100111
010001	100101	001011	111111	111000	001100	100010	010110

$$
\boxed{\begin{array}{cccccc}
U_1 & U_2 & \cdots & U_i & \cdots & U_{2^k} \\
e_2 & U_2 + e_2 & \cdots & U_i + e_2 & \cdots & U_{2^k} + e_2 \\
e_3 & U_2 + e_3 & \cdots & U_i + e_3 & \cdots & U_{2^k} + e_3 \\
\vdots & \vdots & & & & \\
e_j & U_2 + e_j & \cdots & U_i + e_j & \cdots & U_{2^k} + e_j \\
\vdots & \vdots & & & & \\
e_{2^{n-k}} & U_2 + e_{2^{n-k}} & \cdots & U_i + e_{2^{n-k}} & \cdots & U_{2^k} + e_{2^{n-k}}
\end{array}}
$$

$$
S = e_j \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}
$$

Syndrome look-up table

Error Pattern	**Syndrome**
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100
010001	111

Error Correction (5)

- Determine the syndrome corresponding to each of the correctable error sequence by computing $e_j H^T$ for each coset leader:

$$S = e_j \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow$$

Syndrome look-up table

Error Pattern	**Syndrome**
000000	000
000001	101
000010	011
000100	110
001000	001
010000	010
100000	100
010001	111

Activity 5

Assume that codeword **U**=1 0 1 1 1 0, from (6,3) code in Activity 3, is transmitted, and the vector **r**= 0 0 1 1 1 0 is received. Show how a decoder can correct the error (by using syndrome look-up table)

Decoder Implementation

$$S = rH^T$$

$$S = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$s_1 = r_1 + r_4 + r_6$$

$$s_2 = r_2 + r_4 + r_5$$

$$s_3 = r_3 + r_5 + r_6$$