通訊系統 (II)

國立清華大學電機系暨通訊工程研究所
蔡育仁
台達館 821 室
Tel: 62210
E-mail: yrtsai@ee.nthu.edu.tw

Chapter 9
Error-Control Coding

Introduction

Introduction

- Considering different types of communication channels, such as
  - **AWGN channels**: AWGN is the main source of channel impairment, such as the satellite communication channels
  - **Multipath channels**: multipath interference is the main source of channel impairment, such as the wireless channels
  - **Interference channels**: interference is the main source of channel impairment, such as the random access channels
- These scenarios are naturally quite different from each other
  - But they share a common practical shortcoming: **reliability**
- The use of **error-control coding** is essential for supporting **reliable transmissions**.

Introduction (Cont.)

- From a communication theoretic perspective, the two key resources for reliable transmissions are
  - **Transmitted signal power** P
  - **Channel bandwidth** B
- With the **power spectral density** of the receiver noise, the **signal energy per bit-to-noise power spectral density ratio** is

$$E_b/N_0 = E_s/(N_0 \log_2 M) = PT_s/(N_0 \log_2 M) = P/(N_0 B \log_2 M)$$

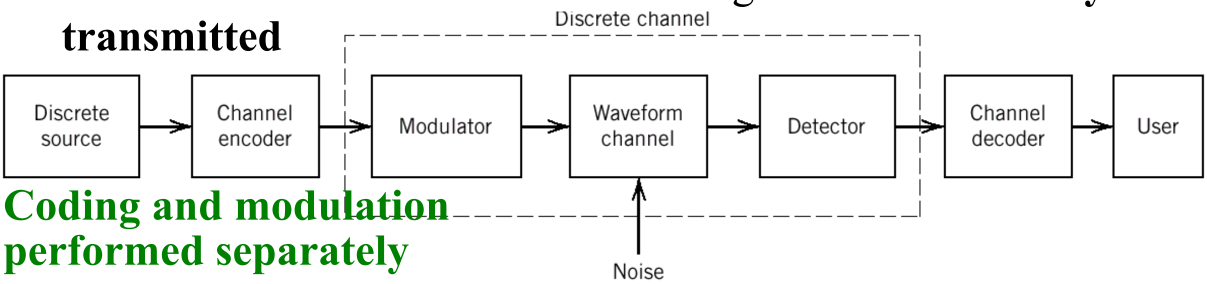
  - E_s: symbol energy; T_s: symbol duration; M-ary modulation
- E_b/N_0 uniquely determines the BER of a particular modulation scheme operating over a **Gaussian noise channel**.
- For a fixed E_b/N_0, the only practical option available for **improving data quality** is to use **error-control coding**

Introduction (Cont.)

- **Error-control coding**: At the transmitter, incorporate a fixed number of **redundant bits** into the structure of a **codeword**
- It is feasible to provide **reliable communication** over a noisy channel
  - Provided that **Shannon's code theorem** is satisfied
- In effect, **channel bandwidth** is traded off for **reliability** in communications.
- Another practical motivation for the use of coding is to **reduce the required E_b/N_0** for a fixed BER. This reduction in E_b/N_0 may, in turn, be exploited to
  - **Reduce** the **required transmitted power**
  - **Reduce** the **hardware costs** by requiring a **smaller antenna size** (antenna gain) in the case of **radio communications**

Forward Error Correction

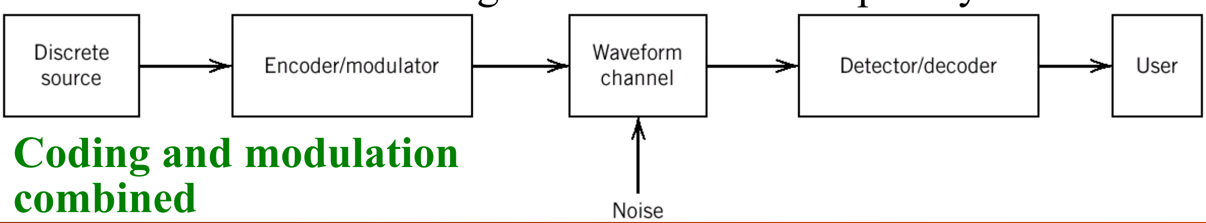
- Error control for data integrity may be achieved by means of **forward error correction** (**FEC**).
- The **discrete source** generates information (binary symbols)
- The **channel encoder** accepts message bits and adds **redundancy** according to a prescribed rule
  - Produce an encoded data stream at a **higher bit rate**
- Based on a **noisy version** of the encoded data stream, the **channel decoder** decide which message bits were **actually transmitted**



Coding and modulation performed separately

Forward Error Correction (Cont.)

- The combined goal of the channel encoder and decoder is to **minimize the effect of channel noise/interference**.
  - The number of errors between the channel encoder input and the channel decoder output (source ⇔ sink) is **minimized**.
- For a fixed **modulation scheme**, the **addition of redundancy** implies the need for
  - **Increasing** in **transmission bandwidth**
  - **Increasing** in **system complexity**
  - **Tradeoff** considering bandwidth and complexity is essential



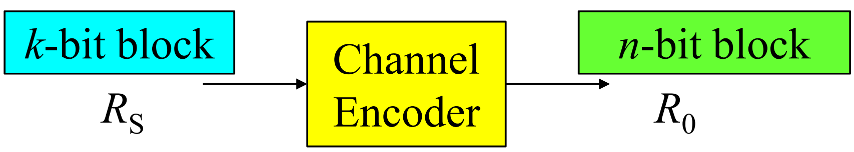
Coding and modulation combined

Types of Error-Correcting Codes

- Historically, error-correcting codes have been classified into **block codes** and **convolutional codes**.
  - The distinguishing feature for this particular classification is the **absence** or **presence** of **memory** in the encoders.
- **Block codes**, **convolutional codes**, and **trellis codes** represent the **classical family of codes**
  - They follow traditional approaches rooted in **algebraic mathematics**
  - **Block codes** and **convolutional codes**: Coding and modulation are designed separately
  - **Trellis codes**: Coding and modulation are designed jointly
- In addition, **turbo codes** and **low-density parity-check (LDPC) codes** are two types of **new generation** coding techniques

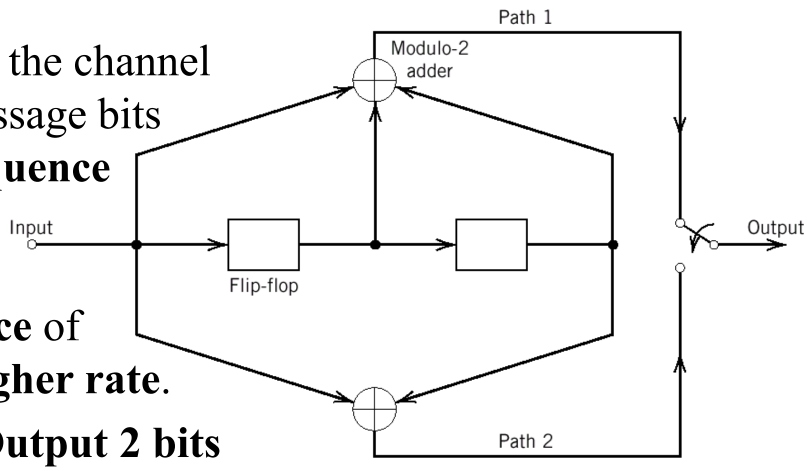
Block Codes

- To generate an (n, k) block code
  - The channel encoder accepts ***k*-bit blocks** successively
  - For each block, the encoder adds $\boldsymbol{n - k}$ **redundant bits**
    - That are **algebraically related to** the k message bits,
    - Thereby producing an encoded block of $\boldsymbol{n}$ **bits**, $n > k$
- **Codeword**: The n-bit block, where n is the **block length**
- The **channel data rate** (at the encoder output) is $R_0 = (n/k)R_S$
  - where R_S is the **bit rate** of the **information source**.
- The ratio $r = k/n$ is called the **code rate**, where $0 < r < 1$.



Convolutional Codes

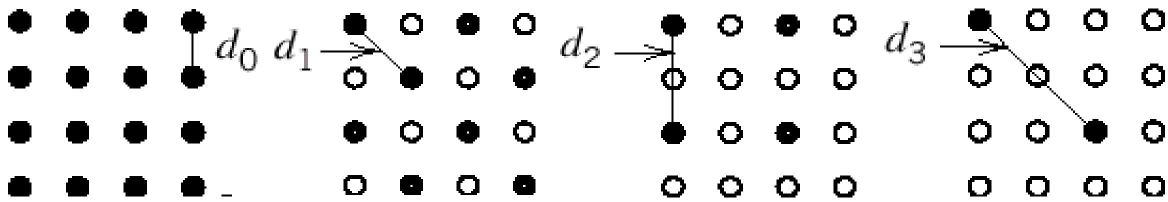
- In a convolutional code, the encoding operation may be viewed as the **discrete-time convolution** of the **input sequence** with the **impulse response of the encoder**.
- The duration of the impulse response equals the **memory** of the encoder.
- Unlike block codes, the channel encoder accepts message bits as a **continuous sequence** and thereby generates a **continuous sequence** of encoded bits at a **higher rate**.
  - **Input 1 bit ⇒ Output 2 bits**



Trellis Codes

- Conventionally, the operations of **channel coding** and **modulation** are **design/performed** separately at the transmitter
- The **most effective** method of implementing forward error correction coding is to **combine** coding with modulation
- Coding is redefined as a process of **imposing certain patterns** (constellation points) on the **transmitted signal**
  - The resulting code is called a **trellis code**
- Based on the concept that different pairs of constellation points have **different error distances**

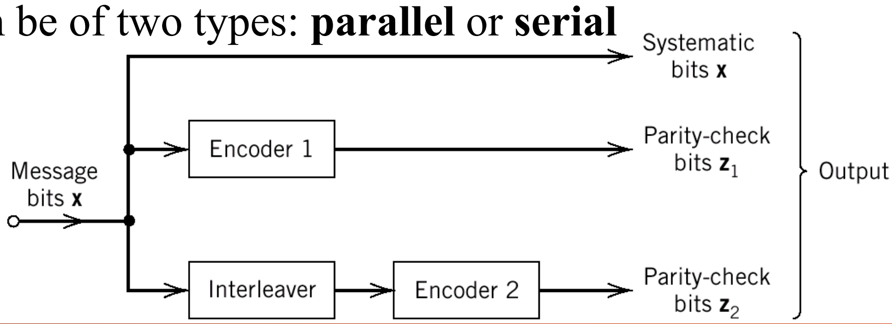
16-QAM



Turbo Codes

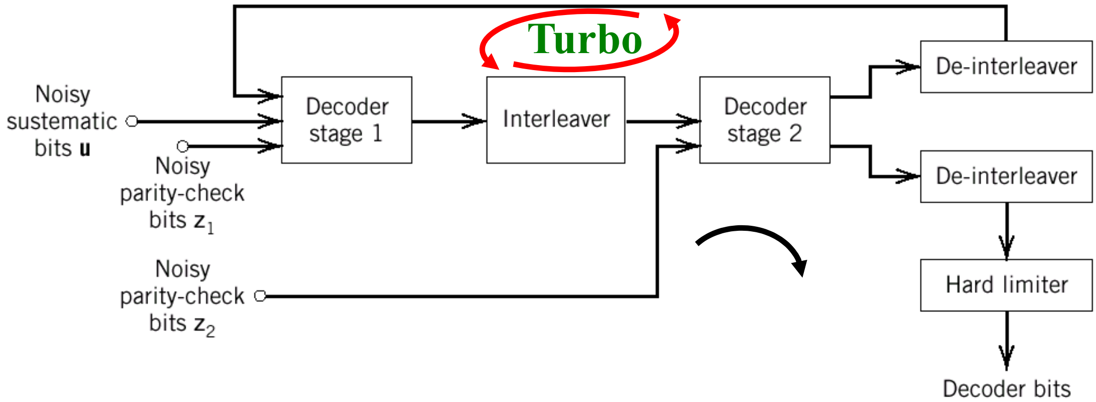
- **Turbo codes** are a class of high-performance forward error correction (FEC) codes
  - The first practical codes to **closely approach** the maximum channel capacity or Shannon limit
  - Turbo codes are used in 3G/4G mobile communications
- The design objective of turbo codes is achieved by using **concatenated codes**
  - which can be of two types: **parallel** or **serial**

Parallel type



Turbo Codes (Cont.)

- The two-stage **turbo decoder** operates on noisy versions of the systematic bits and the two sets of parity-check bits
  - To produce an estimate of the original message bits
- A distinctive feature of the turbo decoder is the use of **feedback**
  - To produce extrinsic information from one decoder to the next in an **iterative manner**



Low-Density Parity-Check (LDPC) Codes

- Low-Density Parity-Check (LDPC) codes are specified by a **parity-check matrix A**, represented as $\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$
  - where $\mathbf{A}_1$ is a square matrix of dimensions $(n-k)\times(n-k)$ and $\mathbf{A}_2$ is a rectangular matrix of dimensions $k\times(n-k)$;
  - **A** is purposely (**randomly** with rules) chosen to be **sparse**; that is, **A** consists mainly of **0**s and a small number of **1**s
- The 1-by-n code vector **c** is partitioned as $\mathbf{c} = [\mathbf{b} \mid \mathbf{m}]$
  - where **m** is the k-by-1 **message vector** and **b** is the $(n-k)$-by-1 **parity-check vector**
- Then, based on the parity-check concept, $\mathbf{c}\,\mathbf{A}^{\mathrm{T}} = [\mathbf{b} \mid \mathbf{m}]\mathbf{A}^{\mathrm{T}} = \mathbf{0}$
- The parity vector **b** is obtained by $\mathbf{b} = \mathbf{mP}$, where $\mathbf{P} = \mathbf{A}_2\mathbf{A}_1^{-1}$

Linear Block Codes

Channel-Coding Theorem (Revisited)

- Consider a discrete memoryless **source** that has the source alphabet $\mathbb{S}$ and entropy $H(S)$ bits per source symbol.
- Assume that the source **emits symbols** once every T_s seconds
  - The **average information rate**: $H(S)/T_s$ bits per second
  - The decoder delivers decoded symbols to the destination at **the same source rate** of one symbol every T_s seconds
- The discrete memoryless **channel** has a **channel capacity** equal to C bits per use of the channel.
- Assume that the channel can be used once every T_c seconds
  - The **channel capacity per unit time**: C/T_c bits per second
  - The **maximum rate** of information transfer over the channel to the destination: C/T_c bits per second

Channel-Coding Theorem (Cont.)

- Shannon's second theorem: the **channel-coding theorem**
- Let a **discrete memoryless source** with an alphabet $\mathbb{S}$ have entropy $H(S)$ for random variable S and produce symbols once every T_s seconds.
- Let a **discrete memoryless channel** have capacity C and be used once every T_c seconds.
- Then, if $$H(S)/T_s \le C/T_c$$ there exists a **coding scheme** for which the source output can be **transmitted** over the channel and be **reconstructed** with an arbitrarily small probability of error.
- The parameter C/T_c is called the **critical rate**.
  - When $H(S)/T_s = C/T_c$, the system is said to be signaling at the critical rate.

Channel-Coding Theorem (Cont.)

- Conversely, if $H(S)/T_s > C/T_c$
  it is **not possible** to transmit information over the channel and reconstruct it **with an arbitrarily small probability of error**.
- The channel-coding theorem is the single **most important** result of information theory.
  - The theorem specifies the channel capacity C as a **fundamental limit** on the rate at which the transmission of reliable **error-free** messages can take place over a discrete memoryless channel.

Error is inevitable — **Information entropy**

Channel capacity

Error-free transmission is possible — **Information entropy**

Binary Arithmetic

- Many of the codes are **binary codes**, for which the alphabet consists only of binary symbols **0** and **1**.
- The encoding and decoding functions involve the binary arithmetic operations of **modulo-2 addition** and **multiplication**.
  - **Modulo-2 addition**: EXCLUSIVE-OR operation
    - $0 + 0 = 0$; $1 + 0 = 1$; $0 + 1 = 1$; $1 + 1 = 0$;
  - **Modulo-2 multiplication**: AND operation
    - $0 \times 0 = 0$; $1 \times 0 = 0$; $0 \times 1 = 0$; $1 \times 1 = 1$;

Linear Block Codes

- Definition of a **linear code**:
  - A code is said to be **linear** if **any two codewords** in the code can be **added in modulo-2 arithmetic** to produce **a third codeword** in the code.
- Consider an (n, k) linear block code, in which k bits of the n code bits are **always identical to the message sequence**.
  - This type of codes are called **systematic codes**.
  - For applications requiring **both error detection** and **error correction**, it simplifies implementation of the **decoder**.
- The $(n - k)$ bits in the remaining portion are computed from the message bits in accordance with a prescribed encoding rule.
  - These $(n - k)$ bits are referred to as **parity-check bits**.

Linear Block Codes (Cont.)

- Let $m_0, m_1, \cdots, m_{k-1}$ constitute a block of k message bits
  - There are 2^k distinct message blocks
- Let this sequence of message bits be applied to a **linear block encoder**, producing an n-bit **codeword**: $c_0, c_1, \cdots, c_{n-1}$
  - The $(n - k)$ **parity-check bits**: $b_0, b_1, \cdots, b_{n-k-1}$
  - For a **systematic code**, a codeword is divided into two parts: the message bits and the parity-check bits
- Assume that the $(n - k)$ **leftmost** bits of a codeword are the corresponding **parity-check bits** and the k **rightmost** bits of the codeword are the message bits.

$$c_i = \begin{cases} b_i, & i = 0, \cdots, n-k-1 \\ m_{i+k-n}, & i = n-k, \cdots, n-1 \end{cases}$$

$$\underbrace{\boxed{b_0, b_1, \cdots, b_{n-k-1}}}_{\text{Parity bits}}\underbrace{\boxed{m_0, m_1, \cdots, m_{k-1}}}_{\text{Message bits}}$$

Linear Block Codes (Cont.)

- The $(n-k)$ parity-check bits are **linear sums** of the k message bits: $b_i = p_{0,i}\, m_0 + p_{1,i}\, m_1 + \cdots + p_{k-1,i}\, m_{k-1}$
  - where $p_{j,i} = 1$, if b_i depends on m_j; and $p_{j,i} = 0$, otherwise
- The coefficients $p_{j,i}$ are chosen in such a way that
  - The rows of the generator matrix are **linearly independent**
  - The parity-check equations are **unique**
- This system can be rewritten in a **matrix form**:
  - The 1-by-k **message** vector $\mathbf{m} = [m_0, m_1, \cdots, m_{k-1}]$
  - The 1-by-$(n-k)$ **parity-check** vector $\mathbf{b} = [b_0, b_1, \cdots, b_{n-k-1}]$
    - $\mathbf{b} = \mathbf{mP}$, where $\mathbf{P}$ is the k-by-$(n-k)$ **coefficient matrix**
  - The 1-by-n **code** vector $\mathbf{c} = [c_0, c_1, \cdots, c_{n-1}]$

Linear Block Codes: Generator Matrix

- The k-by-$(n-k)$ **coefficient matrix** is defined as

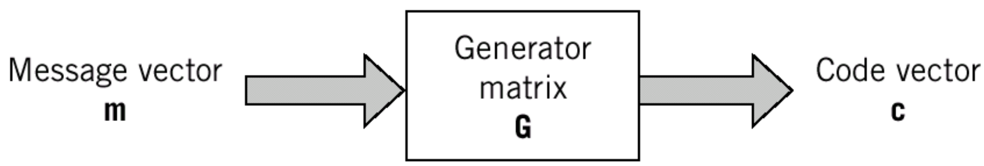
$$\mathbf{P} = \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

- The code vector can be expressed as

$$\mathbf{c} = [\mathbf{b} \vdots \mathbf{m}] = \mathbf{m}[\mathbf{P} \vdots \mathbf{I}_k]$$

  - where $\mathbf{I}_k$ is the k-by-k **identity matrix**
- We then define the k-by-n **generator matrix** as $\mathbf{G}$

$$\mathbf{G} = [\mathbf{P} \vdots \mathbf{I}_k]$$
$$\Rightarrow \mathbf{c} = \mathbf{mG}$$



Linear Block Codes: Generator Matrix (Cont.)

- The full set of **codewords** (the code) is generated by passing the set of possible message vectors **m** into **c** = **mG**
  - The set of all 2^k **binary *k*-tuples** (1-by-k vectors)
- A basic property of linear block codes is **closure**
  - The sum of **any two codewords** in the code is **another codeword**
- Consider a pair of **code vectors** $\mathbf{c}_i$ and $\mathbf{c}_j$ corresponding to a pair of **message vectors** $\mathbf{m}_i$ and $\mathbf{m}_j$, respectively.

$$\mathbf{c}_i + \mathbf{c}_j = \mathbf{m}_i\mathbf{G} + \mathbf{m}_j\mathbf{G} = \left(\mathbf{m}_i + \mathbf{m}_j\right)\mathbf{G}$$

- The modulo-2 sum of $\mathbf{m}_i$ and $\mathbf{m}_j$ is a **new message vector** $\mathbf{m}_k$
  - Correspondingly, the modulo-2 sum of $\mathbf{c}_i$ and $\mathbf{c}_j$ is a **new code vector** $\mathbf{c}_k$

Linear Block Codes: Parity-Check Matrix

- We define the $(n-k)$-by-n **parity-check** matrix as

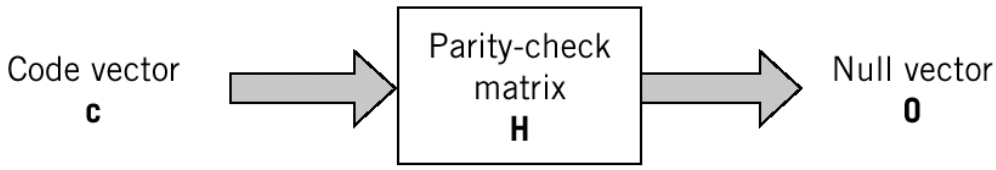
$$\mathbf{H} = \left[\mathbf{I}_{n-k} \vdots \mathbf{P}^{\mathrm{T}}\right]$$

  - where the $(n-k)$-by-k matrix $\mathbf{P}^{\mathrm{T}}$ is the transpose of **P**
- Accordingly, we have

$$\mathbf{H}\mathbf{G}^{\mathrm{T}} = \left[\mathbf{I}_{n-k} \vdots \mathbf{P}^{\mathrm{T}}\right]\begin{bmatrix}\mathbf{P}^{\mathrm{T}} \\ \mathbf{I}_k\end{bmatrix} = \mathbf{P}^{\mathrm{T}} + \mathbf{P}^{\mathrm{T}} = \mathbf{0}; \quad \mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$$

  - In **modulo-2 arithmetic**, the matrix sum $\mathbf{P}^{\mathrm{T}} + \mathbf{P}^{\mathrm{T}}$ is **0**
- The inner product of a **code vector** and the transpose of **H**

$$\mathbf{c}\mathbf{H}^{\mathrm{T}} = \mathbf{m}\mathbf{G}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$$



Linear Block Codes: Syndrome

- The **generator matrix G** is used in the **encoding** operation at the **transmitter**.
- On the other hand, the **parity-check matrix H** is used in the **decoding** operation at the **receiver**.
- Let **r** denote the 1-by-*n* **received vector** that results from sending the code vector **c** over a **noisy binary channel**.
  - The sum of **c** and an **error vector**, or **error pattern**, **e**

$$\mathbf{r} = \mathbf{c} + \mathbf{e}$$

- The *i*-th element of **e** equals **0** (or **1**) if the corresponding element of **r** is **the same as** (or **different from**) that of **c**.

$$e_i = \begin{cases} 1, & \text{if an error has occurred in the } i-\text{th location} \\ 0, & \text{otherwise} \end{cases}$$

Linear Block Codes: Syndrome (Cont.)

- The receiver decodes the code vector **c** from **r**
  - The decoding starts with the computation of a **1-by-(*n* – *k*) vector** called the **error-syndrome vector** or **syndrome**
- The **syndrome** (length $n - k$) corresponding to **r** is defined as

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}}$$

  - Depends only on the **error pattern** and **not** on the transmitted **codeword**

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = (\mathbf{c}+\mathbf{e})\mathbf{H}^{\mathrm{T}} = \mathbf{c}\mathbf{H}^{\mathrm{T}} + \mathbf{e}\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}}$$

  - Equal to the sum of those rows, corresponding to the **errors have occurred**, of the transposed parity-check matrix $\mathbf{H}^{\mathrm{T}}$
  - If errors occur at locations *i* and *j* $\Rightarrow \mathbf{s} = \mathbf{h}_i + \mathbf{h}_j$
    - where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the *i*-th and *j*-th rows of $\mathbf{H}^{\mathrm{T}}$

Linear Block Codes: Syndrome (Cont.)

- For an error pattern **e**, all error patterns that differ to **e** by a codeword are $\mathbf{e}_i$ that satisfy $\mathbf{e}_i - \mathbf{e} = \mathbf{c}_i$
  - There are 2^k distinct code vectors: $\mathbf{c}_i$, $i = 0, 1, \cdots, 2^k - 1$
  - $\mathbf{e}_i = \mathbf{e} + \mathbf{c}_i, \quad \text{for } i = 0, 1, \cdots, 2^k - 1$
  - The set of vectors $\mathbf{e}_i$ is called a **coset** of the code
  - A coset has exactly 2^k elements
  - An (n,k) linear block code has 2^{n-k} possible cosets
    - $2^n / 2^k = 2^{n-k}$
- Each coset of the code is characterized by a unique syndrome

$$\mathbf{e}_i\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}} + \mathbf{c}_i\mathbf{H}^{\mathrm{T}} = \mathbf{e}\mathbf{H}^{\mathrm{T}} + \mathbf{0} = \mathbf{e}\mathbf{H}^{\mathrm{T}}$$

  - **All error patterns** that **differ by a codeword** have **the same syndrome**.

Linear Block Codes: Syndrome (Cont.)

- With the matrix **H**, the $(n - k)$ elements of the syndrome **s** are **linear combinations** of the n elements of the error pattern **e**

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = \mathbf{r}\begin{bmatrix} \mathbf{I}_{n-k} \\ \mathbf{P} \end{bmatrix} = \mathbf{e}\begin{bmatrix} \mathbf{I}_{n-k} \\ \mathbf{P} \end{bmatrix}$$

$$\left.\begin{aligned} s_0 &= e_0 + e_{n-k}p_{0,0} + e_{n-k+1}p_{1,0} + \cdots + e_{n-1}p_{k-1,0} \\ s_1 &= e_1 + e_{n-k}p_{0,1} + e_{n-k+1}p_{1,1} + \cdots + e_{n-1}p_{k-1,1} \\ &\vdots \\ s_{n-k-1} &= e_{n-k-1} + e_{n-k}p_{0,n-k-1} + \cdots + e_{n-1}p_{k-1,n-k-1} \end{aligned}\right\} \text{Linear combinations}$$

- The syndrome (**(*n* – *k*) linear equations**) contains information about the **error pattern** and may be used for **error detection**.
  - There are more unknowns than equations ($(n - k) < n$)
  - The set of equations is **underdetermined**
  - There is **no unique solution** for the error pattern

Minimum Distance Considerations

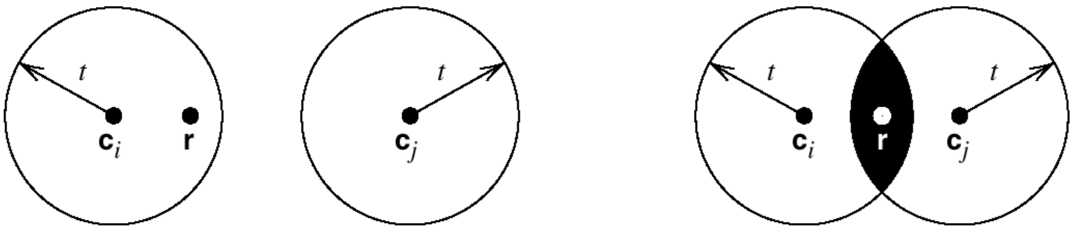
- Consider a pair of code vectors $\mathbf{c}_1$ and $\mathbf{c}_2$ that have the same number of elements.
  - The **Hamming distance**, $d(\mathbf{c}_1, \mathbf{c}_2)$, is defined as the **number of locations** in which their respective elements **differ**.
  - The **Hamming weight**, $w(\mathbf{c})$, of a code vector $\mathbf{c}$ is defined as the **number of nonzero elements** in the code vector.
    - The distance between $\mathbf{c}$ and the **all-zero** code vector.
- The **minimum distance** $d_{\min}$ of a linear block code is the **smallest Hamming distance** between any pair of codewords.
  - $d_{\min}$ is the same as the **smallest Hamming weight** of the **difference** between any pair of code vectors.
  - From the **closure** property, $d_{\min}$ is the **smallest Hamming weight** of the **nonzero code vectors** in the code.

Minimum Distance Considerations (Cont.)

- Specifically, $d_{\min}$ determines the **error-correcting capability** of the code.
- Suppose an (n, k) linear block code is required to **detect** and **correct all error patterns** having a Hamming weight **less than or equal to** t.
  - That is, if a code vector $\mathbf{c}_i$ is transmitted and the received vector is $\mathbf{r} = \mathbf{c}_i + \mathbf{e}$, we require that the **decoder output** $\hat{\mathbf{c}} = \mathbf{c}_i$
    - whenever the error pattern $\mathbf{e}$ has a Hamming weight $w(\mathbf{e}) \le t$
- The **best decoding strategy** is to pick the code vector **closest** to the received vector $\mathbf{r}$
  - **Maximum Likelihood** (ML) decision rule
  - Choose the one with the **smallest Hamming distance** $d(\mathbf{c}_i,\mathbf{r})$

Minimum Distance Considerations (Cont.)

- With the ML strategy, the decoder will be able to detect and correct all error patterns of Hamming weight $w(\mathbf{e})$
  - Provided that the **minimum distance** of the code is **equal to or greater than** $2t + 1$
- **An (n,k) linear block code has the power to correct all error patterns of weight t or less if, and only if,**
  - $d(\mathbf{c}_i, \mathbf{c}_j) \geq 2t + 1$, for all $\mathbf{c}_i$ and $\mathbf{c}_j$



$d(\mathbf{c}_i, \mathbf{c}_j) \geq 2t + 1$ $\qquad$ $d(\mathbf{c}_i, \mathbf{c}_j) < 2t$

Syndrome Decoding

- Consider an (n, k) linear block code with the 2^k **code vectors** $\mathbf{c}_i$ for $1 \leq i \leq 2^k$.
- Let $\mathbf{r}$ denote the **received vector**: one of 2^n possible values
- The receiver partitions the 2^n possible vectors into 2^k **disjoint** subsets D_i
  - The i-th subset D_i corresponds to code vector $\mathbf{c}_i$ for $1 \leq i \leq 2^k$
  - $\mathbf{r}$ is decoded into $\mathbf{c}_i$ if it is in D_i for $1 \leq i \leq 2^k$
- For the decoding to be **correct**, $\mathbf{r}$ must be in the subset that belongs to the code vector $\mathbf{c}_i$ that was actually sent.
- The construction of the 2^k **disjoint** subsets is shown as follows:
  - **Step 1**: The 2^k code vectors are placed in a row with the **all-zero code vector $\mathbf{c}_1$** as the **leftmost** element.

Syndrome Decoding (Cont.)

- **Step 2**: An **error pattern** $\mathbf{e}_2$ is picked and placed under $\mathbf{c}_1$, and a second row is formed by adding $\mathbf{e}_2$ to $\mathbf{c}_i$
- **Step 3**: Repeat Step 2 until **all the possible error patterns** have been accounted for
  - The new error pattern must **not previously appeared**

$$\begin{array}{ccccccc}
D_1\ \mathbf{c}_1 = \mathbf{0} & \mathbf{c}_2 & \mathbf{c}_3 & \cdots & D_i\ \mathbf{c}_i & \cdots & \mathbf{c}_{2^k} \\
\mathbf{e}_2 & \mathbf{c}_2+\mathbf{e}_2 & \mathbf{c}_3+\mathbf{e}_2 & \cdots & \mathbf{c}_i+\mathbf{e}_2 & \cdots & \mathbf{c}_{2^k}+\mathbf{e}_2 \\
\mathbf{e}_3 & \mathbf{c}_2+\mathbf{e}_3 & \mathbf{c}_3+\mathbf{e}_3 & \cdots & \mathbf{c}_i+\mathbf{e}_3 & \cdots & \mathbf{c}_{2^k}+\mathbf{e}_3 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
\mathbf{e}_j & \mathbf{c}_2+\mathbf{e}_j & \mathbf{c}_3+\mathbf{e}_j & \cdots & \mathbf{c}_i+\mathbf{e}_j & \cdots & \mathbf{c}_{2^k}+\mathbf{e}_j \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
\mathbf{e}_{2^{n-k}} & \mathbf{c}_2+\mathbf{e}_{2^{n-k}} & \mathbf{c}_3+\mathbf{e}_{2^{n-k}} & & \mathbf{c}_i+\mathbf{e}_{2^{n-k}} & & \mathbf{c}_{2^k}+\mathbf{e}_{2^{n-k}}
\end{array}$$

Syndrome Decoding (Cont.)

- The 2^k **columns** represent the disjoint subsets D_i
- The 2^{n-k} **rows** represent the **cosets** of the code
  - Their first elements $\mathbf{e}_j$, $j = 2, 3, \cdots, 2^{n-k}$, are **coset leaders**
- The probability of **decoding error** is **minimized** when the **most likely error patterns** are chosen as the **coset leaders**.
  - Those with the **largest** probability of occurrence
- In the case of a **binary symmetric channel**, the **smaller** the **Hamming weight** of an error pattern is, the **more likely** it is for an error to occur.
- The construction should choose the error pattern with the **minimum Hamming weight** in its coset as the **coset leader**

Syndrome Decoding (Cont.)

- The **syndrome decoding** procedure for linear block codes:
- **1.** For the received vector $\mathbf{r}$, compute the syndrome $\mathbf{s} = \mathbf{rH}^T$.
- **2.** Within the coset characterized by the syndrome $\mathbf{s}$, identify the **coset leader**.
  - The error pattern corresponding to the codeword $\mathbf{c}_1$
  - The error pattern is denoted as $\mathbf{e}_0$
- **3.** Compute the code vector $\mathbf{c} = \mathbf{r} + \mathbf{e}_0$ as the decoded output of the received vector $\mathbf{r}$.

$$\mathbf{r} \Rightarrow \mathbf{s} \Rightarrow \mathbf{e}_0 \Rightarrow \mathbf{c} = \mathbf{r} + \mathbf{e}_0$$

Example: Hamming Codes

- **Hamming codes**: a family of (n, k) linear block codes that have the following parameters: $(m \geq 3)$
  - Code length: $n = 2^m - 1$
  - Number of message bits: $k = 2^m - m - 1$
  - Number of parity-check bits: $n - k = m$
- Specifically for $m = 3$, it is the (7, 4) Hamming code with the **error-correcting capability** of $t = 1$ error
- The generator of this code is defined by

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & \vdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & \vdots & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & \vdots & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & \vdots & 0 & 0 & 0 & 1 \end{bmatrix}$$

Example: Hamming Codes (Cont.)

- The corresponding parity-check matrix is given by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & \vdots & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & \vdots & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 1 & 1 & 1 \end{bmatrix}$$

- The columns of **P** consist of all the nonzero m-tuples for $m = 3$
- With $k = 4$, there are $2^k = 16$ distinct message words

Message	Codeword	Weight	Message	Codeword	Weight
0000	0000000	0	1000	1101000	3
0001	1010001	3	1001	0111001	4
0010	1110010	4	1010	0011010	3
0011	0100011	3	1011	1001011	4
0100	0110100	3	1100	1011100	4
0101	1100101	4	1101	0001101	3
0110	1000110	3	1110	0101110	4
0111	0010111	4	1111	1111111	7

Example: Hamming Codes (Cont.)

- The **smallest Hamming weight** of the **nonzero codewords** is 3.
  - It follows that the minimum distance of the code is $d_{\min} = 3$
  - The **error-correcting capability** is $t = 1$ error
- There are 7 error patterns, each of which contains only **1 error**
- The syndrome corresponds to an error pattern: $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}}$
  - If the transmitted codeword is $\mathbf{c}_1$, the received vector $\mathbf{r}$ is the corresponding error pattern of the **coset leader**
- For example: $\mathbf{r} = [0010000]$

$$\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathrm{T}} = [0010000]\begin{bmatrix} 100 \\ 010 \\ 001 \\ 110 \\ 011 \\ 111 \\ 101 \end{bmatrix} = [001]$$

Example: Hamming Codes (Cont.)

- Based on the **syndrome decoding** procedure, the **syndrome** of a received vector shows the **location** of the erroneous bit.
  - If **s** = [001] ⇒ **the third bit** of **r** is **erroneous**
- Thus, **adding** the error pattern $\mathbf{e}_0$ to the received vector **r** yields the **correct code vector** actually sent.
  - $\mathbf{c} = \mathbf{r} + \mathbf{e}_0$

No error → 000

Syndrome	Error Pattern
000	0000000
100	1000000
010	0100000
001	0010000
110	0001000
011	0000100
111	0000010
101	0000001

Homework

- **You must give detailed derivations or explanations, otherwise you get no points.**
- Communication Systems, Simon Haykin **(4^{th} Ed.)**
- 10.4;
- 10.5;
- 10.7;
- 10.8;