Chapter 1

Coding for Reliable Digital Transmission and Storage

Dr. Chih-Peng Li (李志鵬)
1.1 Introduction

- A major concern of designing digital data transmission and storage systems is the control of errors so that reliable reproduction of data can be obtained.
- In 1948, Shannon demonstrated that, by proper encoding of the information, errors induced by a noisy channel or storage medium can be reduced to any desired level without sacrificing the rate of information transmission or storage.
- A great deal of effort has been expended on the problem of devising efficient encoding and decoding methods for error control in a noisy environment.
1.1 Introduction

Block diagram of a typical data transmission or storage system
1.2 Types of Codes

There are three different types of codes in common use today:
- Block codes
- Convolutional codes
- Turbo codes

**Block codes**
- The encoder for a block code divides the information sequence into message blocks of $k$ information bits each.
- A message block is represented by the binary $k$-tuple $\mathbf{u}=(u_1, u_2, \ldots, u_k)$ called a message.
- There are a total of $2^k$ different possible messages.
Block Codes

Block codes (cont.)

- The encoder transforms each message \( u \) into an \( n \)-tuple \( v=(v_1, v_2, \ldots, v_n) \) of discrete symbols called a code word.
- Corresponding to the \( 2^k \) different possible messages, there are \( 2^k \) different possible code words at the encoder output.
- This set of \( 2^k \) code words of length \( n \) is called an \((n,k)\) block code.
- The ratio \( R=k/n \) is called the code rate.
- \( n-k \) redundant bits can be added to each message to form a code word.
- Since the \( n \)-symbol output code word depends only on the corresponding \( k \)-bit input message, the encoder is memoryless, and can be implemented with a combinational logic circuit.
**Block Codes**

Binary block code with $k=4$ and $n=7$

<table>
<thead>
<tr>
<th>Messages</th>
<th>Code words</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0 0 0 0)</td>
<td>(0 0 0 0 0 0 0)</td>
</tr>
<tr>
<td>(1 0 0 0)</td>
<td>(1 1 0 1 0 0 0)</td>
</tr>
<tr>
<td>(0 1 0 0)</td>
<td>(0 1 1 0 1 0 0)</td>
</tr>
<tr>
<td>(1 1 0 0)</td>
<td>(1 0 1 1 1 0 0)</td>
</tr>
<tr>
<td>(0 0 1 0)</td>
<td>(1 1 1 0 0 1 0)</td>
</tr>
<tr>
<td>(1 0 1 0)</td>
<td>(0 0 1 1 0 1 0)</td>
</tr>
<tr>
<td>(0 1 1 0)</td>
<td>(1 0 0 0 1 1 0)</td>
</tr>
<tr>
<td>(1 1 1 0)</td>
<td>(0 1 0 1 1 1 0)</td>
</tr>
<tr>
<td>(0 0 0 1)</td>
<td>(1 0 1 0 0 0 1)</td>
</tr>
<tr>
<td>(1 0 0 1)</td>
<td>(0 1 1 1 0 0 1)</td>
</tr>
<tr>
<td>(0 1 0 1)</td>
<td>(1 1 0 0 1 0 1)</td>
</tr>
<tr>
<td>(1 1 0 1)</td>
<td>(0 0 0 1 1 0 1)</td>
</tr>
<tr>
<td>(0 0 1 1)</td>
<td>(0 1 0 0 0 1 1)</td>
</tr>
<tr>
<td>(1 0 1 1)</td>
<td>(1 0 0 1 0 1 1)</td>
</tr>
<tr>
<td>(0 1 1 1)</td>
<td>(0 0 1 0 1 1 1)</td>
</tr>
<tr>
<td>(1 1 1 1)</td>
<td>(1 1 1 1 1 1 1)</td>
</tr>
</tbody>
</table>
Convolutional code

The encoder for a convolutional code also accepts $k$-bit blocks of the information sequence $u$ and produces an encoded sequence (code word) $v$ of $n$-symbol blocks.

Each encoded block depends not only on the corresponding $k$-bit message block at the same time unit, but also on $m$ previous message blocks. Hence, the encoder has a memory order of $m$.

The set of encoded sequences produced by a $k$-input, $n$-output encoder of memory order $m$ is called an $(n, k, m)$ convolutional code.

The ratio $R = k/n$ is called the code rate.

Since the encoder contains memory, it must be implemented with a sequential logic circuit.
Convolutional Codes

- Binary convolutional encoder with $k=1$, $n=2$, and $m=2$
Convolutional Code Structure

The diagram illustrates the structure of a convolutional code. It consists of a series of blocks (denoted as 1, 2, ..., K) each containing k bits. These blocks are combined using addition (denoted by '+') to produce the output. The process involves taking k bits at a time, combining them with a shift, and repeating this process for n-1 and n cycles. The final output is generated after these processes are completed.
Turbo Codes Basic Concepts

- Turbo coding uses parallel or serial concatenation of two recursive systematic convolutional codes joined through an interleaver.
- Information bits are encoded block by block.
- Turbo codes use iterative decoding techniques.
- Soft-output decoder is necessary for iterative decoding.
- Turbo codes can approach the Shannon limit.
Turbo Codes Encoder - An Example

When the switch is placed on the low position, the tail bits are feedback and the trellis will be terminated.
1.5 Types of Errors

- On memoryless channels, the noise affects each transmitted symbol independently.
- Memoryless channels are called random-error channels.

Transition probability diagrams for binary symmetric channel.
1.5 Types of Errors

- On channels with **memory**, the noise is not independent from transmission to transmission.
- Channel with memory are called **burst-error channels**.

Simplified model of a channel with memory.
1.6 Error Control Strategies

- Error control for a one-way system must be accomplished using forward error correction (FEC), that is, by employing error-correcting codes that automatically correct errors detected at the receiver.

- Error control for a two-way system can be accomplished using error detection and retransmission, called automatic repeat request (ARQ).
  - In an ARQ system, when errors are detected at the receiver, a request is sent for the transmitter to repeat the message, and this continues until the message is received correctly.

- The major advantage of ARQ over FEC is that error detection requires much simpler decoding equipment than does error correction.
1.6 Error Control Strategies

- ARQ is adaptive in the sense that information is retransmitted only when errors occur.
- When the channel error rate is high, retransmissions must be sent too frequently, and the system throughput, the rate at which newly generated messages are correctly received, is lowered by ARQ.