SYSTEM ARCHITECTURE
ADVANCED SYSTEM ARCHITECTURE
BATEMAN
Chapter7: Coding theory and practice

2013/Fall-Winter Term
Monday 12:50

Room# 1-322 or 5F Meeting Room

Instructor: Fire Tom Wada, Professor

Chapter7: Coding theory and practice

7.1 Source coding

7.2 Channel coding

7.3 Block coding

7.4 Advanced block coding

7.5 Convolution coding

7.6 Combined coding and modulation

7.1 Source coding

The purpose of source coding is to transform the information type in the source to a form best suited to the transmission process. Often this involves converting an analogue signal such as voice or light intensity in an image to a digital binary representation for transmission using a modem.

- **Analog to Digital conversion**
  - ✓ **Pulse code modulation**
- **Nyquist sampling**
  - ✓ **Sample without loss of imformation = 2x bandwidth of inputwaveform**
  - ✓ **Aliasing**

Example 7.1

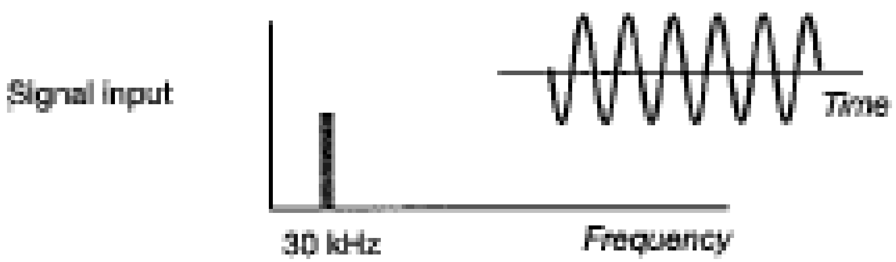
A broadcast audio source signal contains frequencies in the range from 50 Hz to 18 kHz. What is the minimum sampling rate required for an A/D converter in order to ensure that there will be no aliasing? What is a practical sampling rate to choose for this application?

If by accident a high frequency tone at 30 kHz is added into the audio source, at what frequency will this signal appear in the sampled waveform if the sampling rate is set at 40 000 samples per second?

Example 7.1 : solution

The Nyquist sampling rate for perfect signal reconstruction (no aliasing) must be twice the highest frequency component of a baseband signal, that is, 2×18 kHz $= 36\,000$ samples per second. This sampling rate assumes that a 'brick wall' low pass filter can be used to remove alias components. In practice, a sampling rate of 44 100 samples per second is commonly used in the hifi industry as the standard sampling rate for high quality audio signals with frequencies up to 20 kHz.

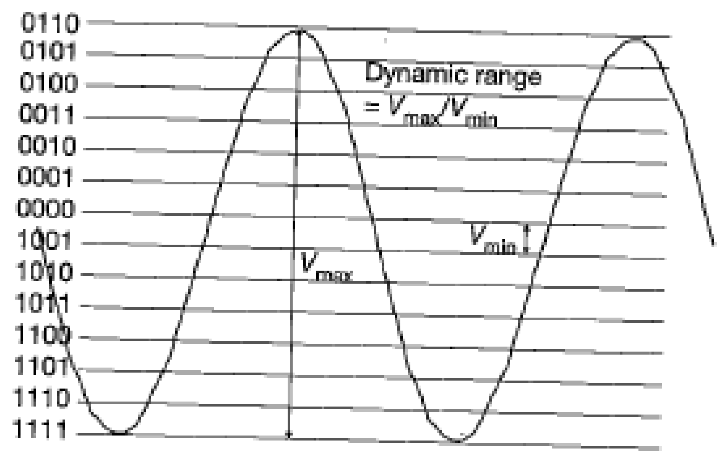
If a sampling rate of 40 000 is used to digitize a tone with a frequency of 30 kHz, the sampling process will produce a difference frequency component at $(40\,000 - 30\,000) = 10\,000$ Hz, which falls well within the half Nyquist bandwidth of 20 kHz for a 40 kHz sampled system. It would thus appear that a 10 kHz tone had been applied at the input to the A/D converter.



Dynamic range

Dynamic range

The ability of an A/D converter to cope with both large and small signals is an important factor in waveform encoding, and the ratio of V_{max} to V_{min} over which a converter will operate is called the *dynamic range*. This parameter depends heavily on the resolution of the A/D, that is, the number of bits available to represent any given sample. The more bits in the converter, the more *quantization* levels the converter is using to match to any given waveform sample.



A/D converter dynamic range

It is not difficult to see that an n-bit converter can differentiate between $2^n = M$ discrete signal levels and that the minimum signal variation that it can detect and represent is V_{max}/M volts. This is often called the *quantization step size*.

> The dynamic range for an A/D converter is thus given by:
>
> $$\text{Dynamic range} = V_{max}/V_{min} = V_{max}/(V_{max}/M) = M \text{ or } 2^n$$
>
> Expressed in dB, this gives us the well-known formula: the dynamic range of a linear A/D is $n \times 6.02$ dB.

An 8-bit converter thus has a dynamic range of approximately 48 dB.

Quantization Noise

IN DEPTH

Calculation of signal to quantization noise ratio

The maximum quantization error q that can occur in the sampled output of an A/D converter is equal to half the minimum resolution of the converter which is given by $q = V_{max}/M$, where $M = 2^n$ is the number of quantization levels for the converter.

Making the assumption that all values of quantization error within the range $+q/2$ to $-q/2$ are equally likely, the mean-squared value of the quantization error, E_q, can be calculated from,

$$E_q = \frac{1}{q}\int_{-q/2}^{q/2} (error)^2 \cdot d \cdot error = \frac{q^2}{12}$$

The rms error is thus $q/\sqrt{12}$.

The peak signal voltage for a sinewave input will be $V_{max} = M \cdot q/2$ (allowing for positive and negative excursions of the sinewave), and the rms value is thus $M \cdot q/2\sqrt{2}$. The signal to quantization noise power ratio can now be determined as:

$$S/N_q = \frac{(Mq/2\sqrt{2})}{(q/\sqrt{12})^2} = \frac{3M^2}{2}$$

Thus, for an 8-bit converter, the peak signal to quantization noise ratio is approximately $3 \times (2^8)^2 \times 0.5 = 98.304$ or 50 dB.

Example 7.2

A telephone system uses filters to band limit the signal from each user to a maximum of 4 kHz. The maximum signal generated by each user is 1 V, and the minimum signal that must be supported is 72 dB below this value. What is the minimum bit rate that must be supported by the telephone system for each user on the network?

If companding is used within the A/D conversion process to achieve equivalent dynamic range, but with only eight bits, what will be the bandwidth required of a binary baseband data link for a single digitized voice channel? If 32 channels are multiplexed into an E1 frame, what is the number of bits per frame and the data rate on the E1 link?

Example 7.2: solution

A 4 kHz maximum input signal requires a minimum sampling rate of 8000 samples per second. In order to maintain a 72 dB dynamic range, the A/D converters must use 72/6 = 12 bits, giving a total bit rate per telephone channel of 8000 × 12 = 96 000 bps.

With the use of companding, the bit rate per channel reduces to 8 × 8000 = 64 kbps. For a baseband channel, the maximum bandwidth efficiency is 2 bits/second/Hz for binary signalling, hence 32 kHz of bandwidth is needed per voice band.

For 32 channels per frame, the frame length of the E1 line is 32 × 8 = 256 bits. The frame rate must equal the sampling rate of any one of the channels, that is, 8000 frames per second, hence the E1 data rate = 256 × 8000 = 2048 Mbps.

A description of the E1 and T1 channel types is given in Section 1.3.

Voice coding

world. At the telephone exchange or switching centre, each analogue signal from the domestic phone is converted using an 8-bit μ-law or A-law codec, with a standardized sampling rate of 8000 times per second. (The maximum voice frequency is limited to 3400 Hz, hence the Nyquist criterion is met.) This results in a data rate of 64 kbps for each voice link.

7.2 Channel coding

Channel coding is most often applied to communications links in order to improve the reliability of the information being transferred. By adding additional bits to the transmitted data stream (which of course increases the amount of data to be sent), it is possible to detect and even correct for errors in the receiver.

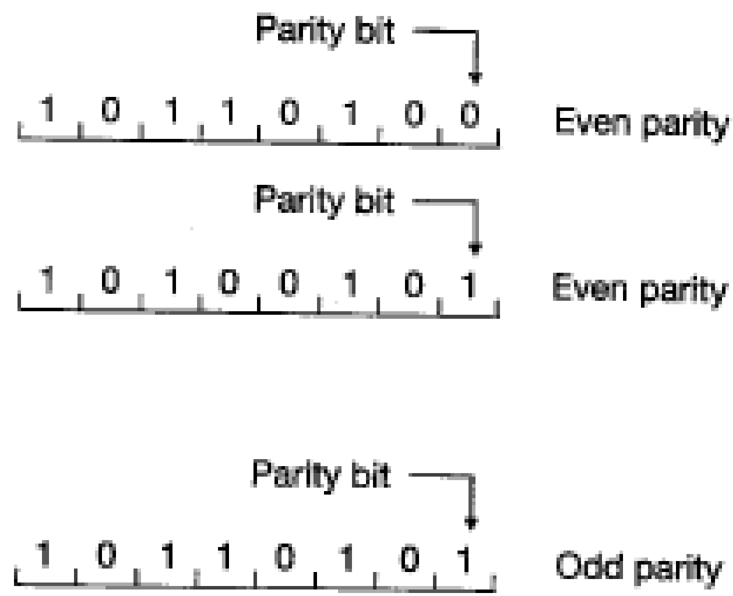
- *Error detection* – In its most elementary form this involves recognizing that part of the received information is in error and if appropriate or permissible requesting a repeat transmission – *ARQ* (*Automatic Repeat Request Systems*).
- *Error detection and correction* – With added complexity, it is possible not only to detect errors, but also to build in the ability to correct errors without recourse to retransmission. This is particularly useful where there is no feedback path to the transmission source with which to request a resend. This process is known as *FEC* (*Forward Error Correction*).

Parity

Parity

A basic requirement of the ARQ system is for the receiving equipment to be able to detect the presence of errors in the received data. One of the simplest yet most frequently used techniques for detecting errors is the *parity check bit*. Most people who have set up a modem for a computer link will have found a setting for *odd* or *even parity* alongside the number of 'stop bits' and so on.

The parity check bit is usually a single bit (1 or 0) appended to the end of a data word such that the number of 1s in the new data word is *even* for even parity, or *odd* for odd parity.



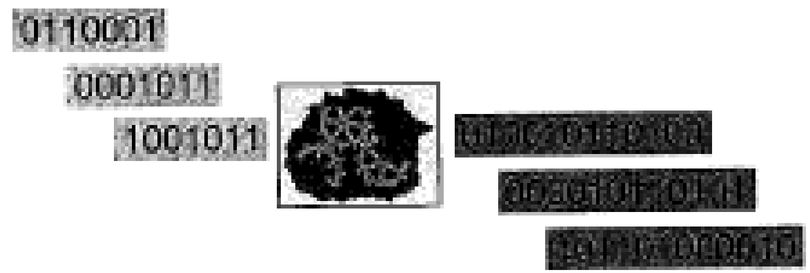
Parity error checking

Thus for the first example data word shown here, a 0 must be added as the parity bit for an even parity design because the number of 1s in the word is already even. For the second word, there is an odd number of 1s so a logic 1 parity bit is added to make this number even.

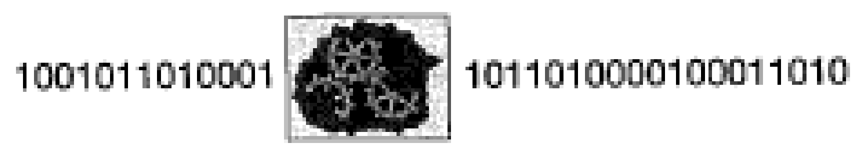
Type of FEC coding

There are two main types of forward error correction coding scheme:

- *Block coding* – where a group (block) of bits is processed as a whole in order to generate a new (longer) coded block for transmission. A complementary block decoder is used in the receiver. Block coding is described in detail in below.
- *Convolutional coding* – which operates on the incoming serial bit stream generating a real-time encoded serial output stream. A complementary serial (convolutional) decoder is used in the receiver. Convolutional coding is described in detail in Section 7.5.



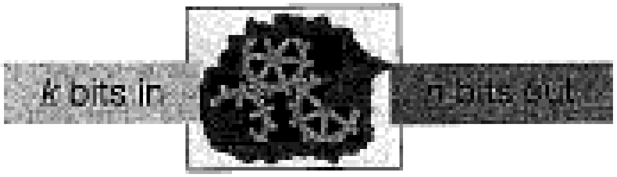
Block coding



Convolutional coding

7.4 Block coding

The terminology for block coding is that an input block of k bits gives rise to an output block of n bits and this is called an (n, k) *code*. This increase in block length means that the useful data rate (the information transfer rate) is reduced by a factor k/n. This is called the *rate of the code*.



Code rate $R = k/n$

The additional data bits are carefully chosen such that they help differentiate one pattern of k bits in a block from a different pattern of k input bits. The factor $1 - k/n$ is usually termed the *redundancy* of the block code.

Hamming codes

A rate $R = 4/7$ Hamming code is illustrated here, with each of the 16 possible four-bit input blocks, coded as seven-bit output blocks. This set of 16 output blocks has been selected from the $2^7 = 128$ possible seven-bit patterns as being most dissimilar. In this case, each output block can be seen to differ from all the other blocks by at least three bits. Hence, if one or two errors occur in the transmission of a block, the decoder will realize that this is not a valid block and flag an error. In the case of only a single bit error occurring, it is also possible for the receiver to match the received block to the closest valid block and thereby *correct* the single error. If three errors occur in a block, the original block may be transformed into a new valid block and all the errors go undetected.

Hamming R = 4/7 code

Block number	Input block data	Output block data
0	0000	000 + 0000
1	1000	110 + 1000
2	0100	011 + 0100
3	1100	101 + 1100
4	0010	111 + 0010
5	1010	001 + 1010
6	0110	100 + 0110
7	1110	010 + 1110
8	0001	101 + 0001
9	1001	011 + 1001
10	0101	110 + 0101
11	1101	000 + 1101
12	0011	010 + 0011
13	1011	100 + 1011
14	0111	001 + 0111
15	1111	111 + 1111

Hamming distance

The number of bits difference between pairs of coded blocks (code words) is a very important property of the code, and is known as the *Hamming distance*. The greater the Hamming distance, the more dissimilar the code words or blocks and the better the chance of detecting or correcting errors.

A block code with a Hamming distance of p can detect up to $p - 1$ errors, and correct $(p - 1)/2$ errors.

Shown here is a subset of code words from a (4, 7) code which can be seen to have a Hamming distance of 3 between each code word. If this were the minimum distance between all code words in the set then this code would be classed as having a *minimum Hamming distance* $d_{min} = 3$. The code could thus detect up to two errors and correct one error.

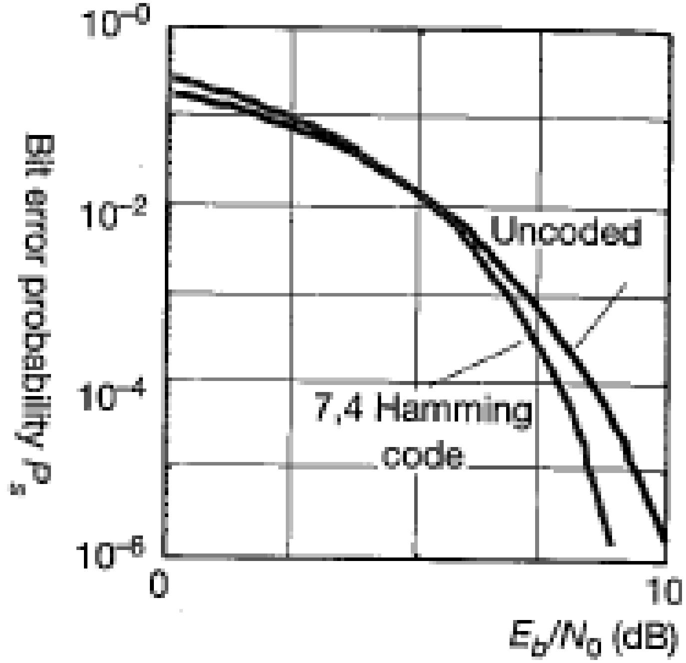
4	0010	111 + 0010
5	1010	001 + 1010
6	0110	100 + 0110
7	1110	010 + 1110
8	0001	101 + 0001

BER performance of (7,4) Hamming

The BER curves for uncoded and coded BPSK are shown here for the $k = 4$, $n = 7$ Hamming code. The curves have been normalized for equal energy per information bit (pre-coding), bearing in mind that the energy per transmitted bit is less than the energy per information bit by a factor equal to the *code rate R*. The mathematics required to work out the performance of some of these codes is very complex and often it is only possible to derive approximate results or run simulations over very long periods of time.

The improvement in E_b/N_0 performance of the coded vs uncoded systems, at a specified BER, is termed the *coding gain*.

Note that:



$$E_b(\text{precoded}) = E_b(\text{postcoded})/R$$

7.5 Advanced block coding

- BCH code :

Hamming codes are in fact a subset of a more general code family called *BCH* (*Bose–Chaudhuri–Hocquenghem*) codes discovered in 1959 and 1960.

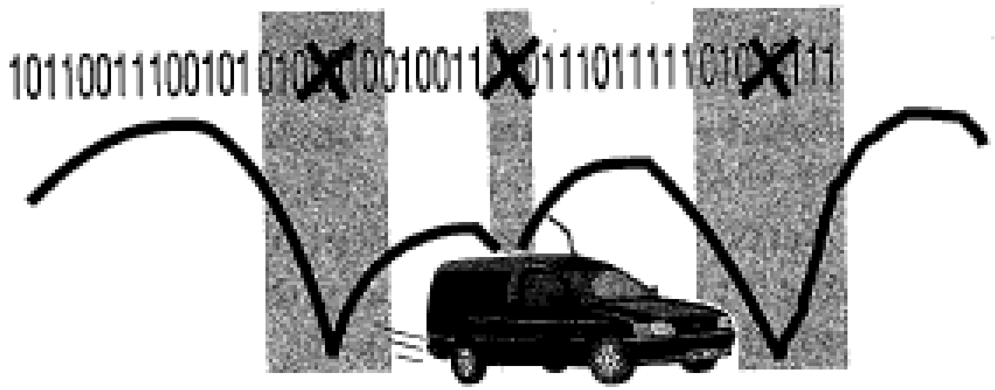
Whereas the Hamming codes can *only* detect up to two errors *or* correct one, the general BCH code family can detect and correct *any number of errors* if the code word used is long enough.

For example, the Hamming (4, 7) code corrects only one error, while the BCH (64, 127) code corrects 10 errors. For real error correcting power, the (11, 1023) code can correct a staggering 255 errors but with a very high coding overhead indeed. This would be used where reliability of transmission is key and data throughput is less important.

k	n	Code rate $R = k/n$	No. of bits corrected
4	7	0.57	1
5	15	0.33	3
24	63	0.38	7
64	127	0.5	10
247	255	0.97	1
171	255	0.67	11
11	1023	0.01	255

To be explained detail in coming lecture.

Interleaving



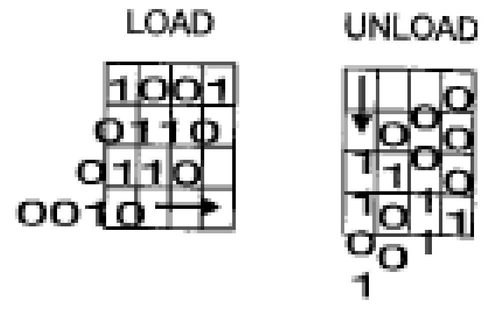
The block codes described thus far work best when errors are distributed evenly and randomly between incoming blocks. This is usually the case for channels corrupted primarily by AWGN, such as a land-line telephone link.

In a mobile radio environment, however, errors often occur in bursts as the received signal fades in and out due to the multipath propagation and the user's motion. In order to distribute these errors more evenly between coded blocks, a process known as *interleaving* is used.

Interleaving (2)

One way to accomplish interleaving is to read the encoded data blocks as rows into a matrix. Once the matrix is full (incurring a time delay penalty), the data can be read out in columns, redistributing the data for transmission.

At the receiver, a *de-interleaving* process is performed using a similar matrix filling and emptying process, reconstituting the original blocks. At the same time the burst errors are uniformly redistributed across the blocks.



Interleaving Matrix

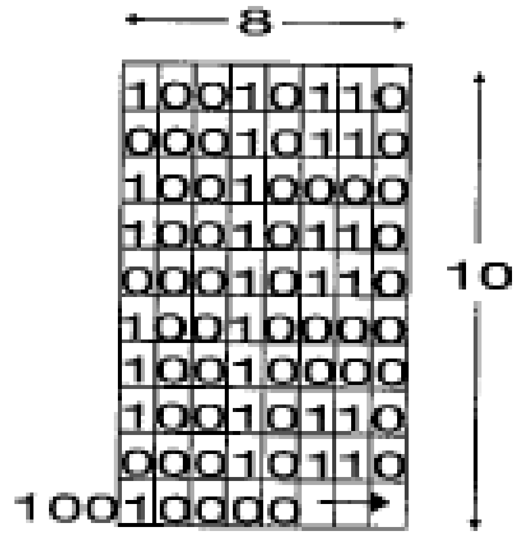
The number of rows or columns in the matrix is sometimes referred to as the *interleaving depth*. The greater the interleaving depth, the greater resistance to long fades, but also the greater the *latency* in the decoding process as both the TX and RX matrix must be full before encoding or decoding can occur.

Example 7.3

A mobile radio data link uses interleaving to spread the data errors on reception. If the interleaving depth used is a 10 × 8 matrix, and the bit rate for the signal is 9600 bps, what is the latency introduced by the interleaving process?

Solution

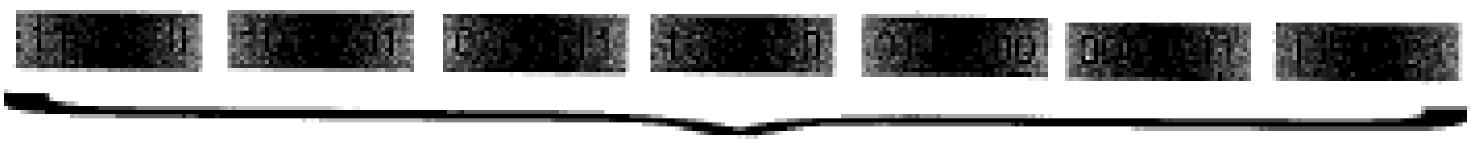
The de-interleaving process requires that the matrix rows are full up before the data can be read out from the columns, thus 8 × 10 bits = 80 bits must be read into the matrix before the de-interleaved data can be extracted. The data rate is 9600 bps, thus the time taken to load the matrix is 80/9600 = 8.3 ms. A further 8.3 ms will be used loading the interleaving matrix in the transmitter, giving a total latency of 16.6 ms.



10 x 8 Interleaving

RS coding

RS codes are a subset of BCH codes that operate at the block level rather than the bit level. In other words, the incoming data stream is first packaged into small blocks, and these blocks are then treated as a new set of *k* symbols to be packaged into a *super-coded* block of *n* symbols. The result is that the decoder is



Reed–Solomon coded block

To be explained detail in coming lecture.

7.5 Convolutional coding

To be explained detail in coming lecture.

Need two presenter

- Each presenter shall chose two problems from 7.1 to 7.10. The one for even and the other for odd problems.
- Show your answer at the beginning of next lecture. Each has 5 minutes.
- Next lecture will be Jan/20 (mon)