COMM1001

Modulation and Coding

Dr. Wassim Alexan
Spring 2018

Lecture 4
Memory and Memoryless Channels

- A communication channel that has memory is one that exhibits mutually dependent signal transmission impairments.

- A channel that exhibits multipath fading, where signals arrive at the receiver over two or more paths of different lengths, is an example of a channel with memory.

- The effect is that the signals can arrive out of phase with each other, and the cumulative received signal is distorted.

- Wireless mobile communication channels, as well as ionospheric and tropospheric propagation channels suffer from such phenomena.

- Telephone line channels and channels disturbed by pulse jamming suffer from switching noise and other burst noise.
Memory and Memoryless Channels

- Such time–correlated impairments result in statistical dependence among successive symbol transmissions

- These disturbances cause errors that occur in bursts, instead of isolated events

- Assuming that a channel has memory, the errors can no longer be characterized as single randomly distributed bit errors whose occurrence is independent from bit to bit

- Most block or convolutional codes are designed to combat random independent errors

- The result of a channel having memory on such coded signals is to cause degradation in error performance

So, what sort of coding techniques can be proposed to deal with such memory channels?
Interleaving

- Typically, for a memory channel with burst errors, rearranging the symbols can cause the bursts of channel errors to be spread in time.

- Doing so allows the decoder to deal with the errors as if they are independently random errors.

- In practice, the channel memory decreases with time separation.

- The idea behind interleaving is to separate the codeword symbols in time, such that the intervening times are similarly filled with symbols of other codewords.

- This codeword symbol–separation in time effectively transforms a channel with memory into a memoryless one.

- This allows random–error–correcting codes to be useful in burst–noise channels.
Interleaving

- An interleaver shuffles the code symbols over a span of several block lengths (for block codes) or several constraint lengths (for convolutional codes)
- The span required is determined by the burst duration
- There are three types of interleavers that are commonly used in communications:
  1. Block
  2. Convolutional
  3. Pseudorandom
Fig. 1. Interleaving example. (a) Original uninterleaved codewords, each comprised of seven coded symbols. (b) Interleaved code symbols. (Sklar, Digital Communications, 2nd edition)
In Fig. 1, there are seven uninterleaved codewords, A through G.

Each codeword is comprised of seven code symbols.

Assume that the code has a single error–correcting capability within each seven–symbol sequence.

If the memory span of the channel is one codeword in duration, such a seven–symbol–time noise burst could destroy the information contained in a complete codeword.

However, by interleaving the codewords, each codeword is now separated from its preinterleaved neighbors by a span of seven symbol times.

Now, assume that a noise channel burst that occupies the seven symbol times occurs.
Interleaving Example

- No complete codeword would be damaged!

- After deinterleaving and decoding the bit stream, since each codeword possesses a single-error-correcting capability, the burst noise has no degrading effect on the final sequence

- Note that the details of the bit redistribution pattern must be known to the receiver in order for the symbol stream to be *deinterleaved* before being decoded
Block Interleaving

- A block interleaver accepts the coded symbols in blocks from the encoder, shuffles the symbols, and then feeds the rearranged symbols to the modulator.

- The shuffling of the block is accomplished by filling the columns of an $M \times N$ array with the encoded sequence.

- After the array is filled, the symbols are then fed to the modulator one row at a time and transmitted over the channel.

- At the receiver, the deinterleaver performs the inverse operation, the symbols are entered by rows and removed one column at a time.
Characteristics of Block Interleaving

1. Any burst of less than \( N \) consecutive symbol errors results in isolated errors at the deinterleaver output that are separated from each other by at least \( M \) symbols.

2. Any \( bN \) burst of errors, where \( b > 1 \), results in output bursts from the deinterleaver no more than \( \lceil b \rceil \) symbol errors. Each output burst is separated from the other bursts by no less than \( M - \lfloor b \rfloor \) symbols.

3. A periodic sequence of single errors spaced \( N \) symbols apart results in a single burst of errors of length \( M \) at the deinterleaver output.

4. The interleaver/deinterleaver end-to-end delay is approximately \( 2MN \) symbol times. To be precise, only \( M(N - 1) + 1 \) memory cells need to be filled before transmission can begin (as soon as the first symbol of the last column of the \( M \times N \) array is filled). A corresponding number needs to be filled at the receiver before decoding begins. Thus, the end-to-end delay is \( (2MN - 2M + 2) \) symbol times.
5. The memory requirement is $M N$ symbols for each location (interleaver and deinterleaver). However, since the $M \times N$ array needs to be (mostly) filled before it can be read out, a memory of $2 M N$ is generally implemented at each location to allow the emptying of one $M \times N$ array while the other is being filled, and vice versa.
Block Interleaving

- Input to the interleaver is column-wise, while output is row-wise. So the interleaver output sequence is

\[1 \ 5 \ 9 \ 13 \ 17 \ 21 \ 2 \ 6 \ 10 \ 14 \ ...

\]

\[N = 6 \text{ columns} \]
\[M = 4 \text{ rows} \]

\[
\begin{array}{cccccc}
1 & 5 & 9 & 13 & 17 & 21 \\
2 & 6 & 10 & 14 & 18 & 22 \\
3 & 7 & 11 & 15 & 19 & 23 \\
4 & 8 & 12 & 16 & 20 & 24 \\
\end{array}
\]

\textbf{Fig. 2.} An } M \times N \text{ Block interleaver example. (Sklar, Digital Communications, 2nd edition)
Exercise 1

- Using the $M = 4$, $N = 6$ structure of Fig. 2, verify each of the block interleaver characteristics described earlier in the lecture
Exercise 1 Solutions

1. Let there be a noise burst of five symbol times, such that the symbols shown encircled in Fig. 3 experience errors in transmission. After deinterleaving at the receiver, the sequence is

\[ 1 \ 2 \ (3) \ 4 \ 5 \ 6 \ (7) \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ (14) \ 15 \ 16 \ 17 \ (18) \ 19 \ 20 \ 21 \ (22) \ 23 \ 24 \]

where the encircled symbols are in error. Observe that the smallest separation between symbols in error is \( M = 4 \)

![Fig. 3. A five-symbol error burst.](Sklar, Digital Communications, 2nd edition)
2. Let $b = 1.5$, so that $bN = 9$. Fig. 4 illustrates an example of a nine-symbol error burst. After deinterleaving at the receiver, the sequence is

1 2 (3) 4 5 6 (7) 8 9 10 (11) 12 13 (14) (15) 16 17 (18) (19) 20 21 (22) (23) 24

Again, the encircled symbols are in error. It is also seen that the bursts consist of no more than $\lceil 1.5 \rceil = 2$ successive symbols and that they are separated by at least $M - \lfloor 1.5 \rfloor = 4 - 1 = 3$ symbols.

*Fig. 4.* A nine-symbol error burst.
(Sklar, *Digital Communications*, 2nd edition)
3. Fig. 5 illustrates a sequence of single errors spaced by $N = 6$ symbols apart. After deinterleaving at the receiver, the sequence is

$$1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ (9) \ (10) \ (11) \ (12) \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20 \ 21 \ 22 \ 23 \ 24$$

It is seen that the interleaved sequence has a single error burst of length $M = 4$ symbols.

![Fig. 5. Periodic single-error sequence space $N = 6$ symbols apart.](Sklar, Digital Communications, 2nd edition)
4. End-to-end delay: The minimum end-to-end delay due to the interleaver and deinterleaver is $(2MN - 2M + 2) = 42$ symbol times.

5. Memory requirement: The interleaver and the deinterleaver arrays are each of size $M \times N$. Therefore, storage for $MN = 24$ symbols is required at each end of the channel. As pointed out earlier, in practice, storage for $2MN = 48$ symbols would generally be implemented.
Exercise 2

- The following sequence is the input to a $4 \times 6$ block interleaver. What is the output sequence?

1101 0101 1111 0000 1010 1011
Exercise 2 Solutions

- We setup the interleaving matrix with 4 rows and 6 columns

- We then input the bit stream column by column

\[
\begin{array}{cccccc}
1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 \\
\end{array}
\]

- The output of the interleaver is then read row by row, resulting in the following bit stream

101011 111000 001011 11001
Block Interleaver Design Notes

- For use with a single-error-correcting code, the interleaver parameters are selected such that the number of columns $N$ overbounds the expected burst length.

- The choice of the number of rows $M$ is dependent on the coding scheme used.

- For block codes, $M$ should be larger than the code block length, thus a burst of length $N$ can cause at most a single error in any block codeword.

- For convolutional codes, $M$ should be larger than the constraint length $K$, thus a burst of length $N$ can cause at most a single error in any decoding constraint length.
Convolutional Interleaving

- Convolutional interleavers were first proposed by Forney and Ramsey in the early 1970s, with a structure as in Fig. 6.

- The code symbols are sequentially shifted into the bank of $N$ registers.

![Convolutional Interleaving Diagram](https://via.placeholder.com/150)

**Fig. 6.** Shift register implementation of a convolutional interleaver/deinterleaver. (Sklar, *Digital Communications*, 2nd edition)
Convolutional Interleaving

- Each successive register provides $J$ symbols more storage than did the preceding one.

- The zeroth register provides no storage → The symbol is transmitted immediately.

- With each new code symbol, the commutator switches to a new register, and the new code symbol is shifted in while the oldest code symbol in that register is shifted out to the modulator/transmitter.

- After the $(N - 1)$th register, the commutator returns to the zeroth register and starts again.

- The deinterleaver performs the inverse operation, and the input and output commutators for both interleaving and deinterleaving must be synchronized.
Convolutional Interleaving Example

- Fig. 7 shows an example of a simple convolutional four–register \((J = 1)\) interleaver being loaded by a sequence of code symbols.

- The synchronized deinterleaver is shown simultaneously feeding the interleaved symbols to the decoder.

- Symbols 1 to 4 are being loaded; the \(\times\)s represent unknown states.

![Convolutional Interleaver/Deinterleaver Diagram](image)

**Fig. 7.** A convolutional interleaver/deinterleaver. (Sklar, *Digital Communications*, 2nd edition)
Convolutional Interleaving Example

- Fig. 8 shows the first four symbols shifted within the registers and the entry of symbols 5 to 8 to the interleaver input.

![Convolutional Interleaver Diagram](image)

**Fig. 8.** A convolutional interleaver/deinterleaver.
(Sklar, *Digital Communications*, 2nd edition)
Convolutional Interleaving Example

- Fig. 9 shows symbols 9 to 12 entering the interleaver

- The interleaver is now filled with message symbols, but nothing useful is being fed to the decoder yet

Fig. 9. A convolutional interleaver/deinterleaver.
(Sklar, *Digital Communications*, 2nd edition)
Convolutional Interleaving Example

- Finally, Fig. 10 shows symbols 13 to 16 entering the interleaver, and at the output of the deinterleaver, symbols 1 to 4 are being passed to the decoder.

- This process continues in the same manner, until the entire codeword sequence, in its preinterleaved form, is presented to the decoder.

Fig. 10. A convolutional interleaver/deinterleaver. (Sklar, *Digital Communications*, 2nd edition)
Convolutional Interleaving Performance

- The performance of a convolutional interleaver is very similar to that of a block interleaver

- One advantage here is that the end–to–end delay is only $M(N - 1)$ symbols, where $M = NJ$

- Another advantage is that the memory requirement is $M(N - 1)/2$ at both ends of the channel

- Thus, there is a reduction of about one–half in delay and memory over the block interleaving requirements
Performance Comparison

<table>
<thead>
<tr>
<th>Interleaver Type</th>
<th>Delay</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>$2M \times N - 2M + 2$</td>
<td>In theory: $M \times N$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In practice: $2M \times N$</td>
</tr>
<tr>
<td>Convolutional</td>
<td>$M(N-1)/2$</td>
<td>In theory: $M(N-1)/2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In practice: $M(N-1)$</td>
</tr>
</tbody>
</table>
Exercise 3

Assuming a couple of interleaver/deinterleaver pairs are designed such that $M = 10$ and $N = 4$. One of them is a block interleaver/deinterleaver and the other is a convolutional interleaver/deinterleaver. Let the symbol duration be $3.6 \mu s$

- (a) Calculate the end–to–end delay for each of the interleaver/deinterleaver pairs

- (b) What is the minimum memory requirement for each of the interleaver/deinterleaver pair?

- (c) What is the practical memory requirement for each of the interleaver/deinterleaver pair?
Exercise 3 Solutions

Assuming a couple of interleavers/deinterleaver pairs are designed such that \( M = 10 \) and \( N = 4 \). One of them is a block interleaver/deinterleaver and the other is a convolutional interleaver/deinterleaver. Let the symbol duration be 3.6 \( \mu s \).

For the **block** interleaver:

- (a) The minimum end-to-end delay due to the interleaver and deinterleaver is
  \[
  (2MN - 2M + 2) = 2 \times 10 \times 4 - 2 \times 10 + 2
  = 62 \text{ symbol durations} = 62 \times 3.6 \times 10^{-6} = 223.2 \mu s
  \]

- (b) The minimum memory requirement for the interleaver/deinterleaver pair is twice
  \( MN = 2 \times MN = 2 \times 10 \times 4 = 80 \text{ memory elements} \)

- (c) The practical memory requirement for the interleaver/deinterleaver pair is twice
  \( 2 \times MN = 4 \times MN = 4 \times 10 \times 4 = 160 \text{ memory elements} \)
Exercise 3 Solutions

Assuming a couple of interleavers/deinterleaver pairs are designed such that $M = 10$ and $N = 4$. One of them is a block interleaver/deinterleaver and the other is a convolutional interleaver/deinterleaver. Let the symbol duration be $3.6 \mu s$

For the **convolutional** interleaver:

- (a) The minimum end–to–end delay due to the interleaver and deinterleaver is $M(N - 1) = 10 (4 - 1)$ symbol durations
  
  $= 30$ symbol durations $= 30 \times 3.6 \times 10^{-6} = 108 \mu s$

- (b) The minimum memory requirement for the interleaver/deinterleaver pair is $M(N - 1) / 2 = 10 (4 - 1) / 2 = 15$ memory elements

- (c) The practical memory requirement for the interleaver/deinterleaver pair is twice $M(N - 1) / 2 = 10 (4 - 1) = 30$ memory elements
Exercise 4

The following sequence is applied as the input to a convolutional interleaver, as shown in Fig. 11. What is the output of this interleaver?

1101 0101 1111 0000 1010 1011

Fig. 11. A convolutional interleaver.
(Sklar, *Digital Communications*, 2nd edition)
Exercise 4 Solutions

- We setup the interleaving matrix, entering the bits column–wise

- We then add columns of x's to the left and right of the columns containing the bits

- The output of the interleaver is then read diagonally from top–left of the message bits to bottom–left of the x's as

\[
\begin{array}{cccccccc}
\times & \times & \times & 1 & 0 & 1 & 0 & 1 & 1 \\
\times & \times & \times & 1 & 1 & 1 & 0 & 0 & 0 \\
\times & \times & \times & 0 & 0 & 1 & 0 & 1 & 1 \\
\times & \times & \times & 1 & 1 & 1 & 0 & 0 & 1 \\
\end{array}
\]

\[1 \times \times x 01 \times x 110 x 0101 1011 1001 x 010 x \times 10 x \times x 1\]
Pseudorandom Interleaving

- A pseudorandom interleaver is a block interleaver which takes a block of channel symbols after encoding and reorders, or permutes, them in a pseudorandom fashion.

- This can be implemented by writing the symbols into a random access memory (RAM) and then reading them out pseudorandomly.

- This interleaving technique provides a high degree of robustness to variability in the burst parameters, but at the cost of more complexity than a block or convolutional interleaver of the same size.

Fig. 12. A couple of RAM chips.
Concatenated Codes

- A concatenated code is one that uses two levels of coding, an inner code and an outer code, to achieve the desired performance.

- The inner code is configured to correct most of the channel errors.

- The outer code, usually a higher-rate (lower redundancy) code, then reduces the probability of error to the specified level.

- The main reason for using a concatenated code is to achieve a low error rate with an overall reduced implementation complexity (as compared to the use of a single code).

- Fig. 13 shows an interleaver between the two coding steps. This is usually required to spread out any error bursts that may appear at the output of the inner coding operation.
Concatenated Codes

- One of the most popular concatenated coding systems uses a Viterbi–decoded convolutional inner code and a Reed–Solomon outer code, with interleaving between the two steps.

- Operation of such a system with $E_b/N_0$ in the range of 2.0 to 2.5 dB to achieve a BER of $10^{-5}$ is possible with practical hardware.

![Block diagram of a concatenated coding system.](sklar_diagram.png)

Fig. 13. Block diagram of a concatenated coding system. (Sklar, *Digital Communications*, 2nd edition)
Fig. 14. BER performance for various concatenated codes, with $K = 8$, $R = 1/3$ convolutional inner code and a $2^k$–symbol $E$–error–correcting outer code.
The earliest research on concatenated codes was carried out at NASA in the 1970s.

NASA’s deep space communication systems are among the most efficient anywhere.

NASA’s deep space network can detect signals so weak that would have to be integrated for ten trillion years to power a fridge light bulb for a single second.

Communications for deep space missions operate close to the theoretical efficiency limit (typically, within 1 dB).

Example: If a spacecraft, designed to work with a 70m parabolic antenna, lost 1 dB of performance, it would take an additional 32m antenna to make up the difference.
The approximate cost of three 32m parabolic antennas is $100 million!

Fig. 15. One of the newest parabolic antennas at NASA’s Deep Space Communication Complex in Canberra, Australia.
Fig. 16 shows one such example of concatenated codes for deep space missions communication.

Butman et al., Performance of concatenated codes for deep space missions, NASA Technical Reports
Space Communications Research in Egypt

- How does all that relate to us as future communications engineers?

Fig. 17. The Egyptian National Authority for Remote Sensing & Space Sciences (NARSS) website. (http://www.narss.sci.eg)
Expectations for the Practical Assignment