Burst Error Correction

- Bursty Channel
- Interleaving
- Concatenated Codes and Deep Space Applications
- Erasure decoding
- Cross-Interleave RS Code and CD/DVD Applications
Bursty Channel

So far, only memoryless channels which cause independent random errors are discussed. Many physical channels generate error bursts instead, i.e. Bursty Channel.

Common causes for error bursts include:

- **Fading** in signal levels (e.g. in mobile comm.)
- **Interference** (e.g. bandlimited/multi-access channels).
- **Contaminated** physical media (e.g. a dirt in magnetic disk, a scratch in Compact Disc)
- Characteristics of inner decoder/equalizer output.
Burst Error Correction

Channel codes discussed so far are mainly designed for **random error correction**.

A random-error-correcting code could perform much worse in a bursty channel than in a memoryless channel, even though the channel BER of the former is much lower than that of the latter.

e.g. A **single-error-correcting code** of length 5 bits can correct all 4 random 1-bit errors below, but fails to correct the 2-bit error burst.

**Bursty error seq:** 00000; 00011; 00000; 00000

**Random error seq:** 00010; 00100; 01000; 10000
Interleaving Techniques

Coding for correcting specified error bursts is a sophisticated branch of coding theory.

A simple but effective technique is to transform bursty channel into short-term memoryless by interleaving.

A channel is short-term memoryless if at any time instant, all symbol errors inside a short window are mutually independent. i.e. adjacent highly correlated symbol errors are separated after interleaving.

If window length > codeword length, the errors experienced by the code is random. In this way, interleaving randomizes the burst error.
Interleaving Techniques

Interleaving is realized by a interleaver/de-interleaver pair, which acts like an innermost rate-1 code.

The idea is to use an interleaver to permute the channel input sequence before transmission, and to undo the permutation after transmission by an de-interleaver.

Short-term Memoryless Channel

Int. → Bursty Channel → De-int.
Block Interleaver

An $n \times m$ block interleaver permutes a block of $nm$ symbols by first writing in an $n \times m$ memory array from right to left & from top to bottom; then reading out from bottom to top & from right to left.

e.g. 3x3 block interleaver from [Wicker].
An $n \times m$ block de-interleaver permutes a block of $nm$ symbols by first writing in an $n \times m$ memory array from top to bottom & from right to left and then reading out from right to left & from bottom to top.

e.g. 3x3 block de-interleaver from [Wicker].
Properties of Block Interleaving

- **Memory requirement** for both interleaver & deinterleaver are $nm$ symbols.
- Interleaver-deinterleaver end-to-end delay is $2(m(n-1) +1) \approx 2nm$ symbols.
- Any 2 adjacent symbols are separated by $(n-1)$ others.
- Typically, each row of the memory array corresponds to an $m$-symbol codeword.
- Using a $t$-error-correcting code per row, length of the shortest uncorrectable error burst is $(nt+1)$ symbols.
Interleaver Permutation Matrix

Let \( \mathbf{x} \) be the input vector of length \( L \) & the \( L \times L \) permutation matrix \( \mathbf{P} \) representing the interleaver. A permutation matrix has a single 1 in each row and each column.

e.g. the previous 3x3 block interleaver is represented by

\[
\mathbf{P} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
\mathbf{x} = \begin{pmatrix}
x_0 \\
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
\end{pmatrix}
\]

\[
\mathbf{x} \mathbf{P} = \begin{pmatrix}
x_6 \\
x_3 \\
x_0 \\
x_7 \\
x_4 \\
x_1 \\
x_8 \\
x_5 \\
x_2 \\
\end{pmatrix} = \tilde{\mathbf{x}}
\]
De-Interleaver Permutation Matrix

Clearly, the de-interleaver performs the inverse permutation $P^{-1}$ s.t. $P^{-1}P = I$, the identity matrix. For permutation matrix, $P^{-1} = P^T$, i.e. transpose of $P$.

For the previous example 3x3 de-interleaver,

$$P^{-1}\bar{x} = P^T\bar{x} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_6 \\ x_3 \\ x_0 \\ x_7 \\ x_4 \\ x_1 \\ x_8 \\ x_5 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = x$$
Interleaver-Deinterleaver Pair

**Theorem:** If \((P, P^{-1})\) is an interleaver-deinterleaver pair (IDP), so is \((P^{-1}, P)\).

**Proof:** Both \(P\) & \(P^{-1}\) are permutation matrices and 
\[ P^{-1}P = I \implies PP^{-1} = I. \]
\(Q.E.D.\)

**Theorem:** If \((P_1, P_1^{-1})\) (\(P_2, P_2^{-1}\)) are both IDP of the same size, so is \((P_1 P_2, P_2^{-1}P_1^{-1})\).

**Proof:** \((P_2^{-1}P_1^{-1})(P_1P_2) = P_2^{-1}(P_1^{-1}P_1)P_2 = P_2^{-1}IP_2 = I, \)

as \(P_1^{-1}P_1 = P_2^{-1}P_2 = I. \)
\(Q.E.D.\)

**Remarks:**
- Concatenation of interleavers are also inteleavers.
- In general, \(P_2^{-1}P_1^{-1} \neq P_1^{-1}P_2^{-1}. \)
Convolutional Interleaving

Block interleaving is convenient only for packetized data transmission with matched block sizes.

Ramsey (1970) proposed convolutional interleaving for continuous data streams over bursty channels s.t. it is unnecessary to packetize stream-oriented data.

Convolutional interleaver/deinterleaver consists of:

- $m$ delay lines.
- the $k^{th}$ delay line has $(k-1)$ $D$-symbol delay elements, for $k=1,2,...,m$.
- multiplexer & demultiplexer at both ends.
Convolutional Interleaver

Convolutional De-Interleaver

e.g. m=4. Taken from [Wicker].
Properties of Conv. Interleaving

- Typically, an $m$-symbol codeword is used so that each of its symbols is on a different delay line.
- Any 2 adjacent symbols are separated by $mD$ symbols from $mD$ different codewords.
- Using a $t$-error-correcting code, length of the shortest uncorrectable error burst is $(mD+1)t+1$ symbols.
- Memory requirement for both interleaver & deinterleaver are $[0+1+2+\ldots+(m-1)]D = m(m-1)D/2$ symbols.
- Interleaver-deinterleaver end-to-end delay is $m(m-1)D$ symbols.

$\Rightarrow$ For similar $(n \approx (m-1)D)$ burst-error-correcting ability, convolutional interleaving has only one-half of the memory requirement and the end-to-end delay, relative to block interleaving.
NASA Planetary Standard Code

For deep space & satellite communications, on-board batteries & solar cells are heavy and contribute significantly to launching costs. A strong code enables a BER of $10^{-5}$ at extremely low SNR is desirable.

Convolutional code with sequential decoding were first used in the *Pioneer* mission.

Later, a **NASA Planetary Standard** includes a rate-1/2 binary convolutional code with constraint length 7 & free distance 10, which was used for the *Voyager* mission.
NASA/ESA Concatenated Codes

In 1987, NASA/ESA Standard adopted the rate 0.437 concatenated code, which joins the rate-1/2 Planetary Standard convolutional code with a (255,223) RS code.

Remark: The codes have been used in Galileo mission to Jupiter & Giotto mission to Halley’s Comet.
Role of Block Interleaving

Viterbi decoder improves the effective channel quality to the point where RS code can be used efficiently.

Though space communications mainly suffer from random noise, errors at the output of the inner Viterbi decoder are bursty in nature.

The outer RS code are $2^8$-ary s.t. burst errors within a symbol, whether 1 bit or 8 bits, have similar effect.

Block interleavers of sizes 2 to 8 times 255-byte codewords were used to randomize long burst of symbol errors before RS decoding is performed.

Remark: Interleaving operates at symbol (not bit) level.
Remark: Concatenated code is ~4dB from Shannon limit -1.6dB.

Taken from [Wicker, Fig. 16-6].
Erasure Decoding

- Demodulator produces an erasure to indicate the received signal sample, whose corresponding symbol value is in doubt.

- **Erasure decoding** is the simplest form of soft-decision decoding.

- Binary Erasure Channel (BEC)

\[
p_e = \text{Erasure Probability}
\]
Erasure Decoding

- **Erasure decoding** provides substantial improvement over fading and bursty channels, though not for AWGN channels.
- BCH and RS codes allow for a very efficient means of erasure decoding.
- Given $f$ erased coordinates, the codewords will have an **effective min distance** of $(d_{\text{min}} - f)$, or an **effective error correcting capability** of $\left\lfloor \frac{d_{\text{min}} - f - 1}{2} \right\rfloor$ over the unerased coordinates.
- In general, we can correct $e$ errors and $f$ erasures if $2e + f < d_{\text{min}}$. 
Erasure Decoding

• Intuitively, we can correct twice as many erasures as errors because the positions of erasures, but not those of errors, are known to begin with.

**Binary Erasure Decoding Algorithm**

**Step 1:** Replace all erasures by 0’s and decode normally (via MLD) to, say, the codeword $C_0$.

**Step 2:** Replace all erasures by 1’s and decode normally to, say, the codeword $C_1$.

**Step 3:** Select the output codeword as the one between $C_0$ and $C_1$ that is closer in Hamming distance to the received word, ignoring erased coordinates.
Erasure Decoding

• Assume that the erasures-replaced-by-0’s codeword contains $e$ and $f_0$ errors in the unerased and erased coordinates respectively. Then the erasures-replaced-by-1’s codeword must contain $e+f-f_0$ errors. One of them must contain at most $e+f/2$ errors and is thus correctable assuming $2e + f < d_{\min}$.

• This justifies the correctness of the previous erasure decoding algorithm.

• For codes with $n - k < d_{\min}$ such as RS codes, an erasure decoder can also be used as the encoder. E.g. a single device suffices for a transceiver in the half-duplex mode modem.
Compact Disc Digital Audio System

The CD Digital Audio System Standard was jointly defined by Philips & Sony in 1979. A recorded bandwidth of 20 KHz provided by a sampling frequency of 44.1 KSamples/s. A plastic disc, 120 mm in diameter with playing time ~70min, stores about $10^{10}$ bits in minute pits which are optically scanned by a laser.

Main sources of channel errors:

- small unwanted particles or air bubbles on disc surface, or pit inaccuracy in manufacturing.
- fingerprints or scratches during handling.
Compact Disc Digital Audio System

Though accurate channel model is missing, errors are mostly bursty in nature. A powerful concatenated code called *Cross-Interleave Reed-Solomon Code* (CIRC) is adopted in the standard.

CIRC provides a hierarchy of error control techniques:
- Perform *error correction*.
- If error correction capability is exceeded, perform *erasure correction*.
- If erasure correction capability is exceeded, perform *error concealment* by interpolating reliable samples.
- If interpolation capability is exceeded, *mute the system* for the duration of unreliable samples.

*Remark:* Cross-interleaving = convolutional interleaving.
## Specification for CIRC

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longest completely correctable burst</td>
<td>~4,000 data bits</td>
</tr>
<tr>
<td></td>
<td>(2.5mm track length)</td>
</tr>
<tr>
<td>Longest interpolatable burst</td>
<td>~12,300 data bits</td>
</tr>
<tr>
<td></td>
<td>(7.7mm track length)</td>
</tr>
<tr>
<td>Sample interpolation rate</td>
<td>1 sample every 10 hours at BER=10⁻⁴, 1000 samples per min at BER=10⁻³.</td>
</tr>
<tr>
<td>Undetected error rate</td>
<td>&lt; 1 every 750 hours at BER=10⁻³, negligible at BER=10⁻⁴</td>
</tr>
<tr>
<td>Implementation</td>
<td>1 LSI chip &amp; 2048 bytes of RAM</td>
</tr>
</tbody>
</table>
CIRC consists of 2 RS encoders & 3 cross-interleavers. It is *byte-oriented*. This takes full advantages of the “burst-trapping” capability of the component *shortened RS codes* defined on GF(256).

It is easier to understand the coding system via the decoder (shown next).
CIRC Decoder

\textbf{D deinterleaver}: \( m=2, \, D=1 \) cross-interleave the even bytes of a frame with the odd bytes of the next frame.

\textbf{(32,28) C_1 decoder}: correct single-byte error. On detecting multi-byte errors, 28 systematic bytes are passed on unchanged as erasure symbols.

\textbf{D* deinterleaver}: \( m=28, \, D=4 \) cross-interleave the erasure symbols over many codewords.

\textbf{(28,24) C_2 decoder}: correct up to 4-byte erasures. On detecting more than 4-byte erasures, 24 systematic bytes are passed on unchanged as erasures.

\textbf{\Delta deinterleaver}: \( m=2, \, D=2 \) cross-interleave uncorrectable 2-byte (left-/right-audio) samples to facilitate interpolation between neighboring samples. If a burst length of 48 frames or consecutive unreliable samples occur, the system is muted (discernible to human ear if the muting time is < a few msec).
Summary

- **Interleaving** techniques effectively convert bursty channels into *(short-term) memoryless* channels allowing the use of *random-error-correcting* codes.

- **Block & convolutional interleaving** effective for *packetized & stream-oriented* data communications respectively were introduced. The latter has only half the memory requirement & end-to-end delay of the former.

- **Erasure decoding** provides the *simplest form of soft-decision decoding* and its complexity is only twice that of hard-decision decoding.

- Important roles of **interleaving, erasure decoding & concatenated coding** techniques in *space communications & CD standards* were discussed.
Suggested Readings
