Fading Mitigation Techniques

- **Equalization**: mitigates frequency-selective fading channels
- **Diversity**: mitigates flat fading
- **Channel Coding**: mitigates errors due to fading and noise, provide time diversity
- **Adaptive Modulation**: mitigates time-selective fading
Causes of ISI

- Channel Distortion

- **Multipath Fading:** Frequency selective channels act as an FIR filter causing channel-induced ISI

- Pulse shapes not designed for zero ISI (GMSK used in GSM)

---

Problems in FS Fading Channels

- Multi-path Channel model causes ISI at receiver

\[ y'_n = x_0 I_n + \sum_{k=1}^{L} x_k I_{n-k} + w_n \]

- As signal power $S \uparrow$, ISI power $\uparrow$

- Hence, $S/(I+N)$ decreases slowly with increasing $S/N$, independent of the signal power
An equalizer is a filter that equalizes the effect of the nonideal (frequency-selective) channel.

- $C(f) H(f) = 1$ (Equalizer inverses the effect of the channel)
- For time-varying channels the equalizer has to be adaptive.
Equalizer Modes

- **Training:**
  - The equalizer must periodically learn the channel by transmitting a training sequence.
  - The training sequence is used at the receiver to choose the equalizer parameters.

- **Tracking:**
  - The equalizer parameters are adjusted based on the difference between the equalizer output and the output of the decision device.

Training & Decision Modes
Classification of Equalizers

- **Linearity:**
  - Linear
  - Nonlinear (DFE, MLSE)

- **Structure:**
  - Transversal
  - Lattice

- **Algorithm:**
  - Zero Forcing
  - MMSE
Linear Transversal Equalizer

Lattice Implementation
Linear Equalizer

- Design $C(z)$ or $c_n$ such that the output is as close to $I_n$ as possible
- **Zero-Forcing:**
  - Forces the samples of combined channel/equalizer impulse response to be zero at all but one of the $NT_s$ spaced samples
  - It is an impulse in time domain:
    \[ F(z)C(z) = 1 \]
  - Simple, but may enhance the noise

**Minimum Mean Square Error (MMSE):**

\[
MSE = E\left[ e_k^2 \right] = E\left[ (x_k - \hat{x}_k)^2 \right] \\
= E\left[ x_k^2 \right] - 2p^Tc + c^TRc
\]

\[
y_k = \begin{bmatrix} y_k & y_{k-1} & \cdots & y_{k-N} \end{bmatrix}^T
\]

\[
c = \begin{bmatrix} c_0 & c_1 & \cdots & c_N \end{bmatrix}^T
\]
To minimize MSE,

\[
\frac{\partial E}{\partial \mathbf{c}} \mathbb{E}[e_k^2] = -2\mathbf{p} + 2\mathbf{R}\mathbf{c} = 0
\]

\[\Rightarrow \mathbf{c}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}\]

To estimate \(\mathbf{R}\) and \(\mathbf{p}\), the transmitter can transmit a training sequence that is known by the receiver.

Equalizer requires periodic retraining in order to maintain effective ISI cancellation.

Decision Feedback Equalizer (DFE)

- DFE attempts to subtract from the current symbol the ISI created by previously detected symbols.
- It performs better than linear equalizer always.
- Suitable to channels with deep frequency nulls.

\[V_n \xrightarrow{\text{Feedforward Filter } c_K \ldots c_0} \hat{I}_n \xrightarrow{\text{Symbol Detection}} \tilde{I}_n \xrightarrow{\text{Feedback Filter } c_1 \ldots c_K}\]
Consists of a feed-forward filter followed by a feedback filter with the bit decisions as input.

DFE does not suffer from noise enhancements.

Coefficients can be updated using LMS, RLS or MMSE algorithms.
Performance Measures

- **Rate of convergence:**
  
  # iterations required for the equalizer to converge to the correct solution

- **Misadjustment:**

  Measure of the amount of deviation from the correct solution

- **Computational complexity:**

  # operations required to make one iteration in the equalizer

Adaptive Algorithms

- **Least Mean Square (LMS):**

  - To minimize:  \[ \text{MSE} = E \left[ e_k^2 \right] = E \left[ (x_k - \hat{x}_k)^2 \right] \]

  - The algorithm is:
    \[ C^{(k+1)} = C^{(k)} - \mu \nabla^{(k)} \]

  where
    \[ \nabla^{(k)} = \frac{\partial E \left[ e_k^2 \right]}{\partial c} = -2p + 2Rc = -2E[e_k y_k] \]

  - So,
    \[ C^{(k+1)} = C^{(k)} + \mu e_k y_k \]
**MSE in LMS Algorithm**

- **Total MSE**
- **MSE due to lag**
- **MSE due to gradient noise**

**Step Size $\mu$**

---

**Training Algorithms Tradeoffs**

- **LMS**: $2N+1$ multiply operations, low complexity, slow convergence, poor tracking
- **MMSE**: $N^2 - N^3$ multiply operations, very high complexity, very fast convergence, good tracking
- **RLS**: $2.5N^2 + 4.5N$ multiply operations, high complexity, fast convergence, good tracking
Maximum Likelihood Sequence Equalizer (MLSE)

- ISI introduces some form of memory (relation between adjacent samples over the span of ISI)
- Instead of detecting the received stream symbol-by-symbol like in previously discussed equalizers
- MLSE observes a sequence of received symbols and searches for the most likely transmitted sequence

MLSE

- Comparing the received sequence to all possible transmitted sequences is a very computational complex task
- An efficient algorithm of finding the most likely sequence without the need for comprehensive search is known as the Viterbi Algorithm
- MLSE is used in GSM
Fractionally-Spaced Equalizers (FSE)

- In previous equalizers, taps are separated by symbol duration $T_s$
- However, the pulse often extends to more than a symbol duration (such as in RC pulses)
- In this case FSE performs better
- In FSE, the taps are separated in time by the reciprocal of Nyquist rate ($<T_s$)
- FSE has better performance

Equalizer Design

- Complexity has to be justified by SNR gain and battery savings
- Coherence time should be greater than equalizer convergence time
- Maximum number of resolvable multipath components in the channel dictates the number of taps in the equalizer
- An equalizer can equalize a channel with a maximum delay spread less than or equal the maximum delay in the equalizer
Performance of Linear Equalizers

- Probability of error vs. SNR, dB

- Channels A and B
- No interference

Performance of DFE

- Probability of error vs. SNR, dB

- Channels A and B
- No interference
- Detected symbols feed back
- Current symbols feed back
- \( N_c = 15 \)
- \( N_f = 15 \)
The GSM System

- Slot duration = 0.577 ms
- Bit duration: 3.69 µs
- Carrier freq. = 900 MHz
- $W = 200$ kHz
- $T_m = 16$ µs
- Assumption:
  Speed 100 km/hr (55.56 m/s)

Case Study: The GSM System
The GSM System

- Channel parameters:
  - $T_c \approx 3\ ms \implies$ slowly fading
  - $B_c \approx 62.5\ kHz \implies$ frequency selective

- Equalization is required
- Observation interval = 4-6 bits

EE 577 - Wireless and Personal Communications

Lecture 16: Diversity
Diversity

- **Basic idea:** send the same information over independently fading paths, then combine the paths
- If diversity branches are uncorrelated, the probability of deeply faded received signal is reduced

- **Macro Diversity:** provides a method to mitigate the effects of shadowing
- **Micro Diversity:** provides a method to mitigate the effects of multi-path fading

Diversity Approaches

- **Space Diversity:**
  - Using antennas spaced enough (at Tx or Rx)
- **Polarization Diversity:**
  - Using antennas with different polarizations
- **Frequency Diversity:**
  - Using frequency channels separated in frequency more than the channel coherence BW
- **Time Diversity:**
  - Using time slots separated in time more than the channel coherence time
- **Multi-path Diversity:**
  - Utilized efficiently in CDMA using RAKE receiver
Time Diversity

Interleave the repeated bits over a duration longer than the coherence time $T_c$

Block Interleaving

**Example:**

- If bit stream is 1 2 3 4 ….12
- After block interleaving,

- Less affected by burst errors
Frequency Diversity

- Send the signal over multiple carriers separated in frequency by more than the Coherence Bandwidth.

- FH-SS is a special case of frequency diversity.

Space (Antenna) Diversity

- Use more than one antenna to receive the signal.
- The distance between two antennas should:
  - exceed $\lambda/2$ at MS due to large amount of scatterers
  - around $10\lambda$ at BS due to less number of scatterers
- Very suitable for base station implementation.
Polarization Diversity

- Use the same antenna to receive the signal
- Orthogonal polarizations (vertical and horizontal) are used to provide two diversity paths
- Reflection coefficients for vertical and horizontal polarized waves are different
- This causes orthogonally polarized waves to undergo uncorrelated fading
- Very suitable for fixed wireless links such as microwave links

Combining Techniques

- Selection Combining (SC)
- Equal Gain Combining (EGC)
- Maximal Ratio Combining (MRC)
- Generalized SC (GSC)

MRC, EGC and GSD require a coherent phase reference to co-phase the different branch signals
**Selection Combining**

**Key idea:** Monitor **ALL** $M$ branches at a time and select the branch with highest SNR to receive the signal.

![Selection Combining Diagram]

Output best one of the $M$ Receivers

---

Let the average SNR in branch $i$ be:

$$\bar{\gamma}_i = \bar{\gamma}$$

The average SNR for Selection Combining is:

$$\bar{\gamma}_S = \bar{\gamma} \sum_{i=1}^{M} \frac{1}{i}$$

The incremental gain becomes extremely small for large value of $M$.
Scanning Diversity (Switched) Combining

- Monitor **ONE** branch at a time.
- If signal quality of monitored branch falls below a threshold, the receiver scans other branches for better signal quality.

![Diagram of Scanning Diversity Combining]

- If threshold is large, the scanning process will be activated often.
- If threshold is small, no improvement in diversity combining.

Feedback Diversity Combining

- Antenna switching is done at the BS.
- Transmitter antennas are switched instead of the receiver antennas.

![Diagram of Feedback Diversity Combining]

- **Advantage:** Simplify the circuit complexity of mobile unit.
- **Disadvantage:** It is not an optimal diversity technique.
Maximal Ratio Combining (MRC)

- The signal from all branches are weighted and then summed together

\[ r_i = a_i s + n_i \]

- where \( s \) is the transmitted signal
- \( n_i \) is the \( i \)-th noise process and \( g_i \)'s are the weights

MRC

- The signals from each of the \( M \) branches are co-phased

- The resultant signal is represented as:

\[ r_T = \sum_{i=1}^{M} g_i r_i \]

- Let the average SNR in branch \( i \) be:

\[ \gamma_i = \frac{r_i^2}{2N_i} \]

- The SNR can be written as:

\[ \gamma = \frac{1}{2} \left( \sum_{i=1}^{M} g_i r_i \right)^2 \]

- Maximization gives:

\[ g_i = \frac{r_i}{N_i} \]
MRC

- **Advantage:** Produce an output with an acceptable SNR even when none of the individual branches are themselves acceptable.
- **Disadvantage:** Channel estimation is required for each diversity gain.

Thus, the SNR becomes:

$$\gamma_{MR} = \sum_{i=1}^{M} \gamma_i$$

The mean SNR is given by:

$$\overline{\gamma}_{MR} = M \overline{\gamma}$$

Equal Gain Combining (EGC)

- Similar to MRC with the weights $a_i$'s are all equal to 1.
- No need to estimate the channel gains for each diversity branch.
- The average SNR is given by:

$$\overline{\gamma}_E = \overline{\gamma} \left( 1 + \frac{(M - 1)\pi}{4} \right)$$
Diversity SNR Gains

Diversity Performance
Generalized Selection Combining (GSC)

- Select the $L$ diversity branches with the largest receive signal level (including noise and interference) among the $M$ branches.
- Combine the selected branches using MRC.
- Provides a tradeoff between SC and MRC:
  - Performs better than SC.
  - Less complexity than MRC.
- Avoid noisy branches with small SNR values.

RAKE Receiver

- Used with DSSS systems.
- Provides a means to combine resolvable multipath components as diversity branches.
- The RAKE receiver works as follows:
  - Correlate the received signal with the PN sequence.
  - Correlate with a delayed version of the PN sequence to capture the first delayed multipath finger.
  - Repeat delay-and-correlate process until all multipath figures are captured.
  - Combine the outputs of the correlators using MRC.
- **Advantage:** Takes advantage of multipath.
- **Disadvantage:** Needs several correlators.
Multi-Input Multi-Output (MIMO)

TX power: \( x \) dB

RX capacity: 6 bits/symbol

TX power: \( x - 3 \) dB

Combine

RX capacity: 6 bits/symbol

3 dB increased sensitivity

Total TX power: \( x \) dB (\( x - 3 \) dB per antenna)

Enables 2-channel simultaneous transmission = Double Rate

RX capacity: 12 bits/channel use

MIMO encode

MIMO decode
Capacity vs. SNR, Flat fading Channel, Open-Loop

$$\text{SNR (dB)}$$

$$\text{bps/Hz}$$

Capacity vs. SNR, Flat fading Channel, Open-Loop

$$\text{SNR (dB)}$$

$$\text{bps/Hz}$$

MIMO

$$H$$ is $$N_t \times N_r$$ matrix

$$x = s H + z$$
Advantages of MIMO

- Order of magnitude increased rate, range and robustness
- Good cost-performance trade-off and scalability
- Same bandwidth and higher rates
  
  => more efficient use of spectrum

- Increase downlink capacity
- Combats multipath fading
- Initial application in 3G WCDMA standard and 802.16 broadband wireless
MIMO Techniques

- **Open-Loop MIMO:**
  - Multiple coded data streams across multiple parallel transmitters
  - Achieve diversity gain by space-time coding and/or interference cancellation at the receiver
  - Linear increase in data rates with number of antennas

- **Closed-Loop MIMO:**
  - Waterfilling to achieve higher data rates
  - Transmitter requires channel knowledge
  - Data rates and range/throughput gains

Adaptive Modulation

- **Basic idea:**
  - Measure the channel at the receiver
  - Feed the measurement back to the transmitter
  - Adapt the transmission scheme relative to the channel estimate to maximize the data rate, minimize transmit power or minimize BER

- **What to adapt?**
  - Constellation size/power
  - Symbol time
  - Coding rate/scheme
Bit Rates in IEEE 802.16a

- Bit rate shifting is achieved using adaptive modulation.
- When you are near to the BS => offered high speed,
- When you are far, reliability decreases => offered lower speed.

HSDPA Features

- Adaptive Modulation and Coding
  - Data rate adapted to radio conditions
  - 2 ms time basis
Multi-User Diversity

- Fast Scheduler
  - 2 ms time basis
  - Round Robin, Proportional Fair or Max-C/I

- Since users are independent with each other, let the users with good channel condition send at any given time ➔ Multiuser Diversity.

- Fairness is also an important attribute

Channel Coding
Basic Channel Coding Concepts

Example: Binary Repetition Codes
- (3,1) code: 0 => 000, 1 => 111
- (3,1) repetition code can correct single errors
- Block error probability:

\[ P_E = \frac{3}{2}(1-p)^2 + \frac{3}{3}p^3 \]

Gain: For a BSC with \( p = 10^{-2} \), \( P_E = 3 \times 10^{-4} \).
Cost: Expansion in bandwidth by 3 times
Error Control Coding

- The function of the encoder is to introduce redundancy in the binary information sequence.
- Such redundancy is used in the receiver to overcome the effects of noise, interference and (fading) encountered through the channel.

- **Encoding** is the process of mapping \(k\)-bit information into a unique \(n\)-bit sequence called the “**codeword**”
- The code rate is defined as \(R = \frac{k}{n}\)

Shannon’s Channel Capacity

- Shannon derived the capacity formula in 1948:

\[
C = W \log_2 \left(1 + \frac{S}{N}\right)
\]

- \(W\) is the bandwidth in Hz
- \(S\) is the signal power in watts
- \(N\) is the total noise power

- The bandwidth efficiency can be found as:

\[
\eta = \frac{\text{Transmission Rate}}{\text{Signal Bandwidth} W} \quad \text{[bits/s/Hz]}
\]

\[
\eta_{\text{max}} = \log_2 \left(1 + \frac{S}{N}\right) \quad \text{[bits/s/Hz]}
\]
Shannon’s Channel Capacity

- The average signal power: \( S = \frac{kE_b}{T} = RE_b \)
- \( \eta_{\text{max}} = \log_2 \left( 1 + \frac{RE_b}{N_0W} \right) \)
  - \( E_b \) is energy per bit
  - \( k \) is the number of bits transmitted per symbol
  - \( T \) is the duration of a symbol
  - \( R = k/T \) is the transmission rate in bits/s
  - \( N = N_0W \) is the total noise power
  - \( N_0 \) is the one-sided noise power spectral density

Shannon’s Channel Capacity

- The minimum bit energy required for reliable transmission (Shannon bound):
  \[ \frac{E_b}{N_0} \geq \frac{2^{\eta_{\text{max}}} - 1}{\eta_{\text{max}}} \]
- In the case of infinite bandwidth, i.e., \( \eta_{\text{max}} \rightarrow 0 \),
  \[ \frac{E_b}{N_0} \geq \lim_{\eta_{\text{max}} \rightarrow 0} \frac{2^{\eta_{\text{max}}} - 1}{\eta_{\text{max}}} = \ln(2) = -1.59 \text{ dB} \]
- This is the **minimum signal-to-noise ratio** required to reliably transmit one bit of information
History of Channel Coding

Black Codes
- Shannon code [17] (1949)
- Random coding [28]
- Hamming code [39] (1950)
- Convolutional codes [29]

Convolutional Codes
- BCH codes [17]
- Reed Solomon codes [17]
- 1000 GUS algorithms [13]
- Braggman shuffle algorithm [17]
- RRNS codes [17, 46]
- Viterbi algorithm [34]
- 1000 Chase algorithm [6]
- 1000 Bald, MAP algorithm [97]

Walsh, cellular block codes [44]

1990
- Viterbo-Ochiai, TCM [88, 89]
- MacKay, VLSI algorithm [88, 97]
- Bahl, 1000/1000/1000/1000 algorithms [84]
- Berrou, turbo codes [83, 84]
- Robertson, Log-MAP algorithm [78]
- Robertson, TCM/CM [83]
- Viterbi, 1000/1000/1000/1000 cellular block code [84]
- Bald, cellular block code [97]

2020
- Punctured, layer space-time block codes [41, 42]
Hamming Distance

- The **Hamming distance** between two codewords $c_i$ and $c_j$, denoted by $d_H(c_i, c_j)$, is the number of elements at which they differ.

**Examples:**

$d_H(011,000) = 2$

$d_H(011,111) = 1$

Error Correction and Detection

- Consider a code consisting of two codewords with Hamming distance $d$. How many errors can be detected? Corrected?

- # of errors that can be detected = $\lambda = d - 1$

- # of errors that can be corrected = $t = \left\lfloor \frac{d - 1}{2} \right\rfloor$

- In other words, for $t$-error correction: $d = 2t + 1$
Minimum Distance of a Code

- **Def.:** The **minimum distance** of a code $C$ is the minimum Hamming distance between any two different codewords.

$$d_{\text{min}} = \min_{i \neq j} d(c_i, c_j), \ \forall c_i, c_j \in C$$

- A code with minimum distance $d_{\text{min}}$ can correct all error patterns up to and including $t$-error patterns, where

$$d_{\text{min}} = 2t + 1$$

- It may be able to correct some higher weight error patterns, but not all.

Linear Block Codes

- A binary information vector ($X$) of $k$ bits is mapped onto a binary vector $C$ with $n > k$ bits.
- The transformation is defined by a generator matrix $G$ which is $k \times n$ matrix.
- The message is segmented into blocks of $k$ bits.
- There are $2^k$ codewords (one for each $2^k$ possible information vectors).
- A binary block code is **linear** if and only if the modulo-2 sum of two codewords is also a codeword.
Generator Matrix

- Message vector: $X = [x_{m1} \ x_{m2} \ x_{m3} \ \ldots \ x_{mk}]$
- Codeword vector: $C = [c_{m1} \ c_{m2} \ c_{m3} \ \ldots \ c_{mn}]$
- Generator matrix of the code:

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

- Encoding is performed by using:

$$C_m = X_m \cdot G$$

(6,3) Linear Block Codes
Example

<table>
<thead>
<tr>
<th>Messages</th>
<th>Codewords</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000000</td>
</tr>
<tr>
<td>100</td>
<td>110100</td>
</tr>
<tr>
<td>010</td>
<td>011010</td>
</tr>
<tr>
<td>110</td>
<td>101110</td>
</tr>
<tr>
<td>001</td>
<td>101001</td>
</tr>
<tr>
<td>101</td>
<td>011101</td>
</tr>
<tr>
<td>011</td>
<td>110011</td>
</tr>
<tr>
<td>111</td>
<td>000111</td>
</tr>
</tbody>
</table>
Systematic Property

- A codeword is divided into two parts:
  - Message (systematic) bits \( k \)
  - Parity-check bits \( n-k \)
- A linear block code with this structure is referred to as a linear **systematic block code**
- The generator matrix for such a code is given by:

\[
G = \left[ I_k | P \right] = \begin{bmatrix}
1 & 0 & \ldots & 0 & p_{11} & p_{12} & p_{13} & \ldots & p_{1n-k} \\
0 & 1 & \ldots & 0 & p_{21} & p_{22} & p_{23} & \ldots & p_{2n-k} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \ldots & \vdots \\
0 & 0 & \ldots & 1 & p_{k1} & p_{k2} & p_{k3} & \ldots & p_{kn-k}
\end{bmatrix}
\]

Encoding Example

- The \((7,4)\) linear code has the following matrix as generator matrix

\[
G = \begin{bmatrix}
g_1 \\
g_2 \\
g_3 \\
g_4
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

- If the message \( x = (1101) \), its corresponding codeword \( C \) is given by:

\[
C = X \cdot G = 1 \times g_1 + 1 \times g_2 + 0 \times g_3 + 1 \times g_4 = (1101000) + (0110100) + (1010001) = (0001101)
\]
**Encoding Example**

- The following matrix $G$ is in the systematic form:
  
  $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

- The codeword $C$ for a message $X$ is given by:
  
  $C = X \cdot G$

  - $c_1 = x_1$
  - $c_2 = x_2$
  - $c_3 = x_3$
  - $c_4 = x_4$
  - $c_5 = x_1 + x_2 + x_3$
  - $c_6 = x_2 + x_3 + x_4$
  - $c_7 = x_4 + x_2 + x_4$

**Parity-Check Matrix**

- An $(n-k) \times n$ parity check matrix $H$ has its rows orthogonal to all codewords generated by $G$.

- Thus, a vector $C$ is a codeword in the code generated by $G$ if and only if $C \times H^T = 0$.

- The matrix $H$ is called a parity check matrix.

- For a generator matrix $G = [I_k \ P]$ => $H = [P^T \ I_{n-k}]$.

- For the last example:

  $H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
Hamming Weight & Distance

- The weight of the codeword is the number of non-zero element it contains
- Hamming distance $d_{ij}$ is a measure of the difference between $C_i$ and $C_j$ in any $(n,k)$ block code
- The smallest value of the hamming weight is called the “minimum distance” $d_{\text{min}}$
- The error detection capability of the code is: $d_{\text{min}} - 1$
- The error correction capability of the code is: $\left\lfloor \frac{1}{2}(d_{\text{min}} - 1) \right\rfloor$

Example: (7,4) Hamming Code

<table>
<thead>
<tr>
<th>No.</th>
<th>Message</th>
<th>Codeword</th>
<th>No.</th>
<th>Message</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0000000</td>
<td>8</td>
<td>0001</td>
<td>1010001</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>1101000</td>
<td>9</td>
<td>1001</td>
<td>0111001</td>
</tr>
<tr>
<td>2</td>
<td>0100</td>
<td>0110100</td>
<td>10</td>
<td>0101</td>
<td>1100101</td>
</tr>
<tr>
<td>3</td>
<td>1100</td>
<td>1011100</td>
<td>11</td>
<td>1101</td>
<td>0001101</td>
</tr>
<tr>
<td>4</td>
<td>0010</td>
<td>1110010</td>
<td>12</td>
<td>0011</td>
<td>0100011</td>
</tr>
<tr>
<td>5</td>
<td>1010</td>
<td>0011010</td>
<td>13</td>
<td>1011</td>
<td>1001011</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>1000110</td>
<td>14</td>
<td>0111</td>
<td>0010111</td>
</tr>
<tr>
<td>7</td>
<td>1110</td>
<td>0101110</td>
<td>15</td>
<td>1111</td>
<td>1111111</td>
</tr>
</tbody>
</table>
Linear Block Codes

- **Linear code**: The sum of any two codewords is a codeword.
- **Hamming codes** constitute a class of single-error correcting codes defined as:
  \[ n = 2^m - 1, \quad k = n - m, \quad m > 2 \]
- The **minimum distance** of the code \( d_{\text{min}} = 3 \)
- Hamming codes are perfect codes.

CYCLIC CODES

- An \((n,k)\) linear code \(C\) is cyclic if every cyclic shift of a codeword in \(C\) is also a codeword in \(C\).
- If \(c_0 \ c_1 \ c_2 \ \ldots \ c_{n-2} \ c_{n-1}\) is a codeword, then the following sequences:
  \[c_{n-1} \ c_0 \ c_1 \ \ldots \ c_{n-3} \ c_{n-2}\]
  \[c_{n-2} \ c_{n-1} \ c_0 \ \ldots \ c_{n-4} \ c_{n-3}\]
  \[\vdots \ \vdots \ \vdots \ \vdots \ \vdots \]
  \[c_1 \ c_2 \ c_3 \ \ldots \ c_{n-1} \ c_0\]
  are all valid codewords.
Example

- The (7,4) Hamming code discussed before is cyclic:
  
  1010001 1110010 0000000 1111111
  1101000 0111001
  0110100 1011100
  0011010 0101110
  0001101 0010111
  1000110 1001011
  0100011 1100101

Code Polynomial

- Let $\mathbf{c} = c_0 \ c_1 \ c_2 \ \ldots \ c_{n-1}$
- The code polynomial of $\mathbf{c}$ is:
  
  $c(X) = c_0 + c_1X + c_2X^2 + \ldots + c_{n-1}X^{n-1}$

  where the power of $X$ corresponds to the bit position, and the coefficients are 0’s and 1’s.

- Example:
  
  1010001 $1+X^2+X^6$
  0101110 $X+X^3+X^4+X^5$
Generator Polynomial

- **All code polynomials** are generated from one polynomial, the *generator polynomial*, using
  \[ c(X) = a(X)g(X) \]

- The *generator polynomial* completely defines the code

- The (7,4) Hamming code can be generated from the generator polynomial \(1+X+X^3\)

BCH Codes

- **Definition of BCH codes:**
  For any positive integers \(m \geq 2\) and \(t_0 \) \((t_0 < n/2)\), there is a BCH binary code of length \(n = 2^m - 1\) which corrects all combinations of \(t_0\) or fewer errors and has no more than \(mt_0\) parity-check bits.

  - Codeword length: \(2^m - 1\)
  - Number of parity-check bits: \(n - k \leq mt_0\)
  - Minimum distance: \(d_{min} \geq 2t_0 + 1\)
Table of Some BCH Codes

<table>
<thead>
<tr>
<th>$n$</th>
<th>$k$</th>
<th>$d$ (designed)</th>
<th>$d$ (actual)</th>
<th>$g(X)$ *</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>3</td>
<td>3</td>
<td>23</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>721</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>7</td>
<td>7</td>
<td>2463</td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>3</td>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>31</td>
<td>16</td>
<td>5</td>
<td>7</td>
<td>107657</td>
</tr>
<tr>
<td>31</td>
<td>11</td>
<td>7</td>
<td>11</td>
<td>5423325</td>
</tr>
</tbody>
</table>

* Octal representation with highest order at the left.

721 is 111 010 001 representing $1+X^4+X^6+X^7+X^8$

Reed-Solomon (RS) Codes

- A class of non-binary BCH codes.
- The codeword consists of $n$ $m$-bit symbols.
- The parameters of the code are related by: $n-k = 2t$

**Example:** $m = 4$, $n = 15$, $k = 11$,
- Codeword length is 15 symbols, or $15*4 = 60$ bits.
- It is a double-error correcting code ($t = 2$).
- It can correct any burst of 8 or less bit errors.
Convolutional Codes

- Convolutional codes differ from block codes in:
  - The encoder has memory
  - The \( n \)-bit output codeword depends on the \( k \)-bit input message and the previous input bits

- Convolutional coding is suitable for long messages such as streaming data (e.g., voice)

- The encoder consists of linear finite-state shift registers of \( K \) stages,

- Input bits are shifted \( k \) at a time to give \( n \) coded bits

- \( K \) is called the constraint length of the code.

- The rate of the code is \( R = \frac{k}{n} \).

Convolutional Code Example 1

- This code has \( K = 3 \), \( k = 1 \), \( n = 3 \), \( \Rightarrow \) rate = 1/3

- The generator functions for the code are:
  \[ g_1 = [100], \quad g_2 = [101], \quad g_3 = [111]. \]

- The generator functions can be represented either in octal form or by generator polynomials:
  - Octal form: \((4, 5, 7)_8\)
  - Generator polynomials:
    \[ g_1(x) = 1 \]
    \[ g_2(x) = 1 + x^2 \]
    \[ g_3(x) = 1 + x + x^2 \]
Convolutional Code Example 2

- This code has $K = 2$, $k = 2$, $n = 3$, and a rate of $2/3$.
- The generator functions for this code are $g_1 = [1011]$, $g_2 = [1101]$, $g_3 = [1010]$ and in octal form, these generators are: $(13, 15, 12)_8$.
- Generator polynomials can be written as:
  
  $g_1(x) = 1 + x^2 + x^3$
  $g_2(x) = 1 + x + x^3$
  $g_3(x) = 1 + x^2$

Encoding Example

- Find the state changes and the resulting output codeword sequence for the message $m = 1101100$. Assume that the initial contents of the encoder are all zero.

<table>
<thead>
<tr>
<th>Input bit</th>
<th>Register Content</th>
<th>State at time $t_i$</th>
<th>State at time $t_{i+1}$</th>
<th>output1</th>
<th>Output2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>000</td>
<td>00</td>
<td>00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>00</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>110</td>
<td>10</td>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>011</td>
<td>11</td>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>101</td>
<td>01</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>110</td>
<td>10</td>
<td>11</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>011</td>
<td>11</td>
<td>01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>001</td>
<td>01</td>
<td>00</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Output sequence: $C = 11010100010111$
Convolutional Codes Representation

- Three methods to represent a convolutional code:
  - Tree diagram
  - Trellis diagram
  - State diagram
- A tree diagram for the rate-1/3, \( K = 3 \) code is shown:

State Diagram

A state diagram for the rate-1/3, \( K = 3 \) code is shown:

- Transition due to 0
- Transition due to 1
Decoding of Convolutional Codes

- **Hard Decoding:**
  The received symbols at the output of the demodulator are quantized into two levels; zero and one, and fed to the decoder.

- **Soft Decoding:**
  - The received symbols at the output of the demodulator are unquantized value (analog value) is used and fed to the decoder.
  - Yields a gain of 2 - 2.2 dB compared to hard decoding.
Decoding of Convolutional Codes

- **Maximum Likelihood Sequence Decoding:**
  - A convolutional code is converted into a block code of length \( L \) by feeding zeros at the end of the input message to force the encoder back to the zero state.
  - Find the codeword with closest Hamming distance (hard decoding) or Euclidean distance (soft decoding) from the possible \( 2^L \) codewords.

- **Viterbi Algorithm:**
  - Computes the distance between the received sequence and all the potential trellis paths.
  - At each stage, keeps one “most likely” (surviving) path for each state.

### Viterbi Decoding Example

- For the convolutional encoder, assume that the received sequence \( z = 11 \ 01 \ 01 \ 10 \ 01 \).
- Find the input message sequence \( m \).

![Trellis diagram illustrating the Viterbi decoding process for a convolutional code.](image-url)
Viterbi Decoding Example

(a) $a = 00, b = 10, h_a = 2, h_b = 0$

(b) $a = 00, b = 10, c = 01, d = 11, h_a = 3, h_b = 3, h_c = 2, h_d = 0$

(c) $a = 00, b = 10, c = 01, d = 11, h_a = 3, h_b = 3, h_c = 2, h_d = 0$
Viterbi Decoding Example

(d)

Viterbi Decoding Example

(e)
Viterbi Decoding Example

(f)

Viterbi Decoding Example

(g)
Viterbi Decoding Example

Distance Properties of Convolutional Codes

- Convolutional codes are linear codes
- The free distance of convolutional codes is associated with the path that starts and ends in the all zero state and does not return in between.
- So given the all-zero transmission an error occurs whenever the all-zero path does not survive.
- The minimum distance is found by exhaustively searching every path from the all-zero state to the all-zero state.
Distance Properties of Convolutional Codes

Examining the previous trellis diagram it is clear that the free distance of the code is 5

This means that this code can correct up to:

\[
\left\lfloor \frac{1}{2} (d_{\text{min}} - 1) \right\rfloor = 2 \quad \text{errors}
\]

A more closed form expression can be obtained by finding the transfer function of the code.
Trellis Coded Modulation (TCM)

- Convolutional encoder with a signal output (instead of binary bits)
- Encoding is done to maximize some distance criterion in the signal constellation

Turbo Codes

- Achieves performance close to channel capacity over AWGN and flat fading channels
- Information is encoded by two encoders after being interleaved
**Turbo Decoder**

- **Iterative decoding** is composed of:
  - Two soft-input soft-output decoders for codes 1 and 2
  - Interleaver – DeInterleaver pair
  - Soft information about massage bits are exchanged between the SISO decoders

![Diagram of Turbo Decoder]

**Coding and Interleaving**

- Codes designed for the AWGN channel do not work well in fading channels due to burst errors

- This can be compensated for by using standard AWGN channel combined with an interleaver to spread burst errors at the decoder
Coding and Interleaving

- Channel coding is a form of time diversity
- Independent fading is needed on each bit in a codeword to get the diversity gain
- Interleaving breaks the memory of the channel and provides independent fading for each bit
- The cost of interleaving is increased complexity and delay