4.1 Duplexing: TDD and FDD

Office Hours:
BKD, 6th floor of Sirindhralai building
Tuesday 14:20-15:20
Wednesday 14:20-15:20
Friday 9:15-10:15
Duplexing

- Allow the subscriber to send “simultaneously” information to the base station while receiving information from the base station.
- Talk and listen simultaneously.

Definitions:
- **Forward channel** or **downlink (DL)** is used for communication from the infrastructure to the users/stations.
- **Reverse channel** or **uplink (UL)** is used for communication from users/stations back to the infrastructure.

Two techniques
1. Frequency division duplexing (FDD)
2. Time division duplexing (TDD)

[Rappaport, 2002, Ch 9]

Frequency Division Duplexing (FDD)

- Provide **two distinct bands** of frequencies (simplex channels) for every user.
- The **forward band** provides traffic from the base station to the mobile.
- The **reverse band** provides traffic from the mobile to the base station.
- Any duplex channel actually consists of two simplex channels (a forward and reverse).
- Most commercial cellular systems are based on FDD.
FDD Examples

Q: Why do we need them to be far apart?
A: At an antenna, tx power is much larger than the rx power.

[Karim and Sarraf, 2002, Fig 6-1]

Power of Sinc Function

\[ G(f) = \text{sinc} \left( \frac{\pi}{\lambda_{n}} f \right) \]

\[ G^2(f) = \text{sinc}^2 \left( \frac{\pi}{\lambda_{n}} f \right) \]

\[ 10 \log_{10} \left( \frac{1}{|G(f)|^2} \right) = -50 \text{ dB} \]

\[ f = 105.4 \]
Problems of FDD

- Each transceiver simultaneously transmits and receives radio signals
  - The signals transmitted and received can vary by more than 100 dB.
  - The signals in each direction need to occupy bands that are separated far apart (tens of MHz)
- A device called a duplexer is required to filter out any interference between the two bands.

Even with filter, any practical filter is non-ideal. So, we still see the Tx power leaking into the Rx band.

[Tse and Viswanath, 2005, Ch 4, p 121]

Time Division Duplexing (TDD)

- The UL and DL data are transmitted on the same carrier frequency at different times. (Taking turns)
  - Use time instead of frequency to provide both forward and reverse links.
  - Each duplex channel has both a forward time slot and a reverse time slot.
- “Unpaired spectrum”
**Time Division Duplexing (TDD)**

- If the *time separation* between the forward and reverse time slot is *small*, then the transmission and reception of data *appears* simultaneous to the users at both the subscriber unit and on the base station side.
- Used in Bluetooth and Mobile WiMAX
- LTE can be FDD or TDD.

---

**FDD and TDD LTE frequency bands**

<table>
<thead>
<tr>
<th>FDD LTE frequency band allocations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LTE BAND NUMBER</strong></td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>31</td>
</tr>
</tbody>
</table>

**TDD LTE frequency band allocations**

<table>
<thead>
<tr>
<th><strong>LTE BAND NUMBER</strong></th>
<th><strong>ALLOCATION (MHz)</strong></th>
<th><strong>WIDTH OF BAND (MHz)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>1900 - 1920</td>
<td>20</td>
</tr>
<tr>
<td>34</td>
<td>1910 - 1930</td>
<td>20</td>
</tr>
<tr>
<td>35</td>
<td>1920 - 1940</td>
<td>20</td>
</tr>
<tr>
<td>36</td>
<td>1930 - 1950</td>
<td>20</td>
</tr>
<tr>
<td>37</td>
<td>1940 - 1960</td>
<td>20</td>
</tr>
<tr>
<td>38</td>
<td>1950 - 1970</td>
<td>20</td>
</tr>
<tr>
<td>39</td>
<td>1960 - 1980</td>
<td>20</td>
</tr>
<tr>
<td>40</td>
<td>1970 - 1990</td>
<td>20</td>
</tr>
<tr>
<td>41</td>
<td>1980 - 2000</td>
<td>20</td>
</tr>
<tr>
<td>42</td>
<td>1990 - 2010</td>
<td>20</td>
</tr>
<tr>
<td>43</td>
<td>2000 - 2020</td>
<td>20</td>
</tr>
</tbody>
</table>

- LTE TDD has been commercial since 2011 and is gaining global momentum.
- The initial global unpaired bands include
  - 2.3GHz (B40) used in India and
  - 2.6 GHz (B38) used Europe,
  - with variations (B41) in the U.S. and Japan.
- China is expected to launch LTE TDD in multiple global bands.

Global LTE TDD spectrum

[Qualcomm, “LTE TDD - the global solution for unpaired spectrum”, September, 2014]

Operating bands specified for LTE in 3GPP above 1 GHz

[Dahlman, Parkvall, and Skold, 2016]
Disadvantages of TDD
Advantages of FDD

- TDD frames need to incorporate guard periods equal to the max round trip propagation delay to avoid interference between uplink and downlink under worst-case conditions.

![Diagram showing TDD frames with guard periods (GP)].

- There is a time latency created by TDD due to the fact that communications is not full duplex in the truest sense.

Disadvantages of FDD
Advantages of TDD

- Duplexer is not required.
- Enable adjustment of the downlink/uplink ratio to efficiently support asymmetric DL/UL traffic.
  - With FDD, DL and UL have fixed and generally, equal DL and UL bandwidths.
- Ability to implement in unpaired spectrum
  - FDD requires a pair of channels
  - TDD only requires a single channel for both DL and UL providing greater flexibility for adaptation to varied global spectrum allocations.
- Assure channel reciprocity for better support of link adaptation, MIMO and other closed loop advanced antenna technologies.
Channel Reciprocity

- Usually, for better performance, to choose the coding/modulation scheme and its parameters, the transmitter needs to learn the channel state information (CSI).
- It is relatively easier for a receiver to find CSI.
  - Is the decoded message readable or gibberish?
  - Usually, this is done by the transmitter sending a preamble training sequences or pilot symbols to the receiver.
- How can a transmitter obtain CSI?
  - The corresponding receiver may convey this information via a feedback link.
    - An overhead which reduce the efficiency of the system.
      - Even worse when there are many parameters of the channel to learn.
        - For example, for MIMO, there are many antennas.
    - The information can only be used for a short time duration.
      - The channel changes due to mobility of the Tx, Rx, or objects in the environment.
  - Use channel reciprocity

Channel Reciprocity

- The channel from point A to point B is identical to the channel from B to A if the channel is measured at the same time and same frequency.
- The channel from A to B can be estimated at A using the pilot symbols embedded in the signal sent from B.
- Using the reciprocity principle, this estimate is also an estimate for the channel from A to B.
- In FDD systems, the two directions use different frequencies. Thus, channel reciprocity does not hold.
Multiple Access Techniques

- Allow many mobile users to share simultaneously a finite amount of radio spectrum.
- For high quality communications, this must be done without severe degradation in the performance of the system.
- Important access techniques
  1. Frequency division multiple access (FDMA)
  2. Time division multiple access (TDMA)
  3. Spread spectrum multiple access (SSMA)
     - Frequency Hopped Multiple Access (FHMA)
     - Code division multiple access (CDMA)
  4. Space division multiple access (SDMA)
  5. Random access
     - ALOHA
Frequency division multiple access (FDMA)

- The *oldest* multiple access scheme for wireless communications.
- Used exclusively for multiple access in 1G down to individual resource units or physical channels.
- Assign individual channels to individual users.
  - Different carrier frequency is assigned to each user so that the resulting spectra do not overlap.
  - During the period of the call, no other user can share the same channel.

FDMA (2)

- **Band-pass filtering** (or heterodyning) enables separate demodulation of each channel.
- If an FDMA channel is not in use, then it sits idle and cannot be used by other users to increase or share capacity.
  - It is essentially a *wasted resource*.
- In FDD systems, the users are assigned a channel as a pair of frequencies.

[Rappaport, 2002, Ch 9, p. 449]
Time division multiple access (TDMA)

- Divide the radio spectrum into **time slots**.
- In each slot only one user is allowed to either transmit or receive.
- A channel may be thought of as a particular time slot that reoccurs every frame, where $N$ time slots comprise a frame.

- Transmit data in a **buffer-and-burst method**
  - The transmission for any user is non-continuous.
  - Digital data and digital modulation must be used with TDMA.
  - This results in low battery consumption, since the subscriber transmitter can be turned off when not in use (which is most of the time).
- An obvious choice in the 1980s for digital mobile communications.

FDMA vs. TDMA

- FDMA: Frequency Division Multiple Access
- TDMA: Time Division Multiple Access
- CDMA: Code Division Multiple Access
Tradeoffs

- TDMA transmissions are slotted
  - Require the receivers to be **synchronized** for each data burst.
  - **Guard times** are necessary to separate users. This results in larger overheads.
  - FDMA allows completely **uncoordinated transmission** in the time domain
    - No time synchronization among users is required.
- The complexity of FDMA mobile systems is lower when compared to TDMA systems, though this is changing as digital signal processing methods improve for TDMA.
- Since FDMA is a continuous transmission scheme, fewer bits are needed for **overhead** purposes (such as synchronization and framing bits) as compared to TDMA.
- FDMA needs to use costly **bandpass filters**.
  - For TDMA, no filters are required to separate individual physical channels.

Guard Band vs. Guard Time
GSM utilizes a combination of FDMA and TDMA

- Two-dimensional channel structure
- Each narrowband channel has bandwidth 200 kHz.
- Time is divided into slots of length $T = 577 \text{ ms}$. 

The FDMA/TDMA structure of GSM

- In full-rate configuration, eight time slots (TSs) are mapped on every frequency.

A BS with 6 carriers, as shown here, has 48 (8 times 6) physical channels (in fullrate configuration).
Classifications of Medium Access Control (MAC)

MAC Protocols

- Scheduled Access
  - Static: TDMA, FDMA, CDMA
  - Dynamic: Reservation
- Random Access
  - Static: ALOHA
  - Dynamic: Tree, FCFS

Multiple Access Techniques in Cellular System

<table>
<thead>
<tr>
<th>Cellular System</th>
<th>Multiple Access Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Mobile Phone System (AMPS)</td>
<td>FDMA/FDD</td>
</tr>
<tr>
<td>Global System for Mobile (GSM)</td>
<td>TDMA/FDD</td>
</tr>
<tr>
<td>Interim Standard 95 (IS-95)</td>
<td>CDMA/FDD</td>
</tr>
<tr>
<td>W-CDMA (3GPP)</td>
<td>CDMA/FDD</td>
</tr>
<tr>
<td>CDMA/TDD</td>
<td>CDMA/TDD</td>
</tr>
<tr>
<td>cdma2000 (3GPP2)</td>
<td>CDMA/FDD</td>
</tr>
<tr>
<td></td>
<td>CDMA/TDD</td>
</tr>
</tbody>
</table>
FDMA or TDMA?

Used exclusively for multiple access in 1G down to individual resource units or physical channels

FDMA or TDMA?

Transmit data in buffer-and-burst method
FDMA or TDMA?

Need to use costly bandpass filters

FDMA or TDMA?

Allows completely uncoordinated transmission in the time domain
Spread spectrum (SS)

- Historically spread spectrum was developed for secure communication and military uses.
- Difficult to intercept for an unauthorized person.
- Easily hidden.
  - Can even hide below the noise floor during transmission
  - For an unauthorized person, it is difficult to even detect their presence in many cases.
- Resistant to narrowband jamming and interference.

[Goldsmith, 2005, Ch 13]
Spread spectrum (SS)

- Provide a measure of immunity to distortion due to multipath propagation.
  - In conjunction with a RAKE receiver, can provide coherent combining of different multipath components.
- Asynchronous multiple-access capability.
- Wide bandwidth of spread spectrum signals is useful for location and timing acquisition.
- Applications
  - Cordless phones.
  - The basis for both 2nd and 3rd generation cellular systems as well as 2nd generation wireless LANs (WLAN).

[Goldsmith, 2005, Ch 13]

Spread spectrum: Definition

Spread spectrum refers to any system that satisfies the following conditions [Lathi, 1998, p 406 & Goldsmith, 2005, p. 378]:

1. The spread spectrum may be viewed as a kind of modulation scheme in which the modulated (spread spectrum) signal bandwidth is much greater than the message (baseband) signal bandwidth.

2. The spectral spreading is performed by a code that is independent of the message signal.
   - This same code is also used at the receiver to despread the received signal in order to recover the message signal (from the spread spectrum signal).
   - In secure communication, this code is known only to the person(s) for whom the message is intended.

SS: Processing Gain and BW Sharing

- Increase the bandwidth of the message signal by a factor $N$, called the **processing gain** (or bandwidth **spreading factor**).
  - In practice, $N$ is on the order of **100-1000**. [Goldsmith, 2005, p 379]
    - $N = 128$ for IS-95 [T&V]
  - Wasteful?

- **Bandwidth Sharing**: Although we use much higher BW for a spread spectrum signal,
  - **Multiplexing**: we can also multiplex large numbers of such signals over the same band.
  - **Multiple Access**: many users can share the same spread spectrum bandwidth without interfering with one another.
    - Achieved by assigning **different code** to each user.
    - Frequency bands can be reused without regard to the separation distance of the users.

---

Two forms of spread spectrum

1. **Frequency Hopping** (FH)
   - Hop the modulated data signal over a wide BW by changing its carrier frequency
   - BW is approximately equal to $NB$
     - $N$ is the number of carrier frequencies available for hopping
     - $B$ is the bandwidth of the data signal.
   - The most celebrated invention of frequency hopping was that of actress **Hedy Lamarr** and composer George Antheil in 1942

2. **Direct Sequence** (DS)
Hedy Lamarr

- In 2015, Google honoured Hedy Lamarr with a Google Doodle on her 101st birthday.

FHSS Example: Bluetooth

- The band at 2.4 GHz is divided into 79 channels.
- A Bluetooth device, hops frequency at a rate of 1600 hops per second, randomly selecting a channel of 1 MHz to operate.

DS/SS System
DS/SS System: The Basic

\[ x(t) = m(t)c(t) \]
\[ y(t) = x(t) \]

Compare with ECS332’s modulation:

\[ x(t) = m(t)c(t) \]
\[ y(t) = x(t) \]
DS/SS System (Con’t)

Observe that...

- To be able to perform the despreading operation, the receiver must
  - know the code sequence \( c(t) \) used at the Tx to spread the signal
  - synchronize the codes of the received signal and the locally generated code.
- The process of detection (despreading) is identical to the process of spectral spreading.
  - Recall that for DSB-SC, we have a similar situation in that the modulation and demodulation processes are identical (except for the output filter).

DS/SS System: Signals

- We assume polar signals.
  - In particular, we assume \( c^2(t) = 1 \).

Message signal (data/information signal)

Spreading code (spreading sequence)
DS/SS System: Signals

- During the time that \( m(t) = 1 \), the spreading code is non-inverted in \( x(t) \).
- During the time that \( m(t) = -1 \), the spreading code is inverted (or negated) in \( x(t) \).

\[ y(t) = x(t) = m(t)c(t) \]

DS/SS: Spectral Spreading Signal \( c(t) \)

- The spreading code \( c(t) \) is designed to be pseudorandom
- Appear to be unpredictable
- Can be generated by deterministic recipe (hence, pseudorandom)
  - This will be studied in the next section.

- Each rectangular pulse in \( c(t) \) is called a chip.
- The bit rate of \( c(t) \) is then known as the chip rate.

\[ T_c = \text{duration of } 1 \text{ chip} \]
**DS/SS System: Signals**

- The bit rate of $c(t)$ is chosen to be much higher than the bit rate of $m(t)$.
- In fact, by definition, the spreading factor $N = \frac{T_b}{T_c}$.

$$T_b = \text{duration of 1 data bit}$$

**Message signal**
(data/information signal)

**Spreading code**
(spreading sequence)

$$T_c = \text{duration of 1 chip}$$

---

**Review: Spectrum of PAM signal**

$$p(t) = 1[t \in [0, T_s]]$$

$$x_{PAM}(t) = \sum \limits_n m[n] p(t - nT_s) \xrightarrow{\mathcal{F}} \mathcal{X}_{PAM}(f) = P(f) \sum \limits_n m[n] e^{-j2\pi fnT_s}$$
**Frequency-Domain Analysis**

We will simply draw the main lobes.

**DS/SS: Secure Communication**

- As $N$ increases, the peak of $X(f)$ is reduced.
- Secure communication
  - Signal can be detected only by **authorized** person(s) who **know** the pseudorandom code used at the transmitter.
  - Signal spectrum is spread over a very wide band, the signal’s **spectral level is very small**, which makes it easier to hide the signal within the noise floor.
DS/SS: Jamming Resistance

\[ y(t)c(t) = (x(t) + i(t))c(t) = m(t)c^2(t) + i(t)c(t) = m(t) + i(t)c(t) \]

- Jamming Resistance / Narrowband Interference rejection
  - The decoder despreads the signal \( y(t) \) to yield \( m(t) \).
  - The jamming signal \( i(t) \) is spread to yield \( i(t)c(t) \).
  - Using a LPF, we can recover \( m(t) \) with only a small fraction of the power from \( i(t) \).
- Caution: Channel noise will not spread.

SS Modem

The BW of the first LPF is much wider because it needs to pass the whole spread signal.
Binary Random Sequences

- While DSSS chip sequences must be generated *deterministically*, properties of binary random sequences are useful to gain insight into deterministic sequence design.
- A random binary chip sequences consists of i.i.d. bit values with probability one half for a one or a zero.
  - Also known as *Bernoulli sequences/trials*, “coin-flipping” sequences
- A random sequence of length \( N \) can be generated, for example, by flipping a fair coin \( N \) times and then setting the bit to a one for heads and a zero for tails.
Binary Random Sequence

- These names are simply many versions of the same sequence/process.
- You should be able to convert one version to others easily.
- Some properties are conveniently explained when the sequence is expressed in a particular version.

Properties of Binary Random Sequences:

- Consider the sequence $X_1, X_2, X_3, \ldots, X_n, \ldots$
- Disadvantages
  - Can not further “compress” the sequence
  - Difficult to convey the sequence from the Tx to Rx
  - Require large storage at both Tx and Rx
- Advantages
  - Random = unpredictable
    1. Balanced property
    2. Run length property
    3. Shift property
Properties of Binary Random Sequences: Balanced Property

- $\{0,1\}$ version

$$\text{The proportion of 1s in the sequence} = \frac{1}{N} \sum_{i=1}^{N} X_i \xrightarrow{N \to \infty \text{ LLN}} \mathbb{E}[X_i] = 0 \times \frac{1}{2} + 1 \times \frac{1}{2} = \frac{1}{2}$$

- $\{\pm 1\}$ version

$$\frac{1}{N} \sum_{i=1}^{N} X_i \xrightarrow{N \to \infty \text{ LLN}} \mathbb{E}[X_i] = (-1) \times \frac{1}{2} + 1 \times \frac{1}{2} = 0$$

Runs: An Example

- A run is a subsequence of consecutive identical symbols within the sequence.

- The following sequence contains 16 runs

```
0001111100110100100001010111011
```

- Rel. Freq of Runs

<table>
<thead>
<tr>
<th>Run Length</th>
<th>Rel. Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1/16</td>
</tr>
<tr>
<td>4</td>
<td>1/16</td>
</tr>
<tr>
<td>3</td>
<td>2/16</td>
</tr>
<tr>
<td>2</td>
<td>4/16</td>
</tr>
<tr>
<td>1</td>
<td>8/16</td>
</tr>
</tbody>
</table>

- Rel. Freq of Run Lengths

<table>
<thead>
<tr>
<th>Run Length</th>
<th>Rel. Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>1/16</td>
</tr>
<tr>
<td>00000</td>
<td>1/16</td>
</tr>
<tr>
<td>111</td>
<td>1/16</td>
</tr>
<tr>
<td>000</td>
<td>1/16</td>
</tr>
<tr>
<td>11</td>
<td>2/16</td>
</tr>
<tr>
<td>00</td>
<td>2/16</td>
</tr>
<tr>
<td>1</td>
<td>4/16</td>
</tr>
<tr>
<td>0</td>
<td>4/16</td>
</tr>
</tbody>
</table>
Properties of Binary Random Sequences: Run Length Property

Start of a run of 1s

011001111100001

\[ P[\text{run length} = 1] = \frac{1}{2} \]

\[ P[\text{run length} = 2] = \frac{1}{4} \]

\[ P[\text{run length} = 3] = \frac{1}{8} \]

\[ P[\text{run length} = \ell] = \frac{1}{2^\ell} \quad \text{As } \ell \text{ increases, } P \text{ is reduced.} \]

\[ \text{Small probability of having log runs.} \]

FYI: Run-Length Encoding (RLE)

- A very simple form of lossless data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.
- Most useful on data that contains many such runs.
- Example: Consider a screen containing plain black text on a solid white background.
  A line, with B representing a black pixel and W representing white, might read as follows:

```
WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW
```

With a RLE data compression algorithm applied to the above line, it can be rendered as follows:

```
12W1B12W3B24W1B14W
```
Properties of Binary Random Sequences: Shift Property

Original sequence: $X_1 \ X_2 \ X_3 \ X_4 \ \cdots \ X_N$

Shifted sequence: $X_{-1} \ X_0 \ X_1 \ X_2 \ \cdots \ X_{N-2}$
(amount of shift (delay) = 2)

- When the **shifted amount** = 0, the two sequences are exactly the same.
- When the **shifted amount** = $s$, we want to compare $X_j$ and $X_{j-s}$.
  - What proportion are the same?
  - What proportion are different?
- Recall that the numbers in the sequence are independent results (from several Bernoulli trials)

<table>
<thead>
<tr>
<th>$X_j$</th>
<th>$X_{j-s}$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>¼</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>¼</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>¼</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>¼</td>
</tr>
</tbody>
</table>

Properties of Binary Random Sequences: Shift Property

- $\{0,1\}$ version: The comparison is done via the XOR ($\oplus$) operation
  - $x \oplus y = 0$ iff they are the same
  - $x \oplus y = 1$ iff they are different
- $\{\pm 1\}$ version: The comparison is done via the multiplication operation
  - $x \times y = 1$ iff they are the same
  - $x \times y = -1$ iff they are different
Key randomness properties

[Goldsmith, 2005, p. 387 & Viterbi, p. 12] Binary random sequences with length $N$ asymptotically large have a number of the properties desired in spreading codes

- **Balanced property**: Equal number of ones and zeros.
  - Should have no DC component to avoid a spectral spike at DC or biasing the noise in despreading

- **Run length property**: The run length is generally short.
  - half of all runs are of length 1
  - a fraction $1/2^n$ of all runs are of length $n$ (Geometric)
  - Long runs reduce the BW spreading and its advantages

- **Shift property**: If they are shifted by any nonzero number of elements, the resulting sequence will have half its elements the same as in the original sequence, and half its elements different from the original sequence.

Pseudorandom Sequence

- A deterministic sequence that has the balanced, run length, and shift properties as it grows *asymptotically large* is referred to as a pseudorandom sequence (noiselike or pseudonoise (PN) signal).

- Ideally, one would prefer a random binary sequence as the spreading sequence.

- However, practical synchronization requirements in the receiver force one to use periodic Pseudorandom binary sequences.
  - m-sequences
  - Gold codes
  - Kasami sequences
  - Quaternary sequences
  - Walsh functions
m-Sequences

- **Maximal-length sequences**
- A type of **cyclic code**
  - Generated and characterized by a generator polynomial
  - Properties can be derived using algebraic coding theory
- Simple to generate with **linear feedback shift-register** (LFSR) circuits
  - Automated
- **Approximate a random binary sequence.**
- Disadvantage: Relatively easy to intercept and regenerate by an unintended receiver

(Serial-in/Seral-out) Shift Register

- Accept data serially: one bit at a time on a single line.
- Each clock pulse will move an input bit to the next FF. For example, a 1 is shown as it moves across.
- Example: five-bit serial-in serial-out register.
Linear Feedback Shift-Register (LFSR)

- Binary sequences drawn from the alphabet \{0,1\} are shifted through the shift register in response to clock pulses.
  - Each clock time, the register shifts all its contents to the right.
- The particular 1s and 0s occupying the shift register stages after a clock pulse are called states.

Suppose there are \(r\) FFs. Then a state \(s\) of the SR can be represented by \(r\) bits.
- There are \(2^r\) possible states.
- There are \(2^r - 1\) non-zero states.

GF(2)

- **Galois field** (finite field) of two elements
- Consist of
  - the symbols 0 and 1 and
  - the (binary) operations of
    - modulo-2 addition (XOR) and
    - modulo-2 multiplication.
- The operations are defined by

\[
\begin{array}{c|cc}
+ & 0 & 1 \\
\hline
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array} \quad \begin{array}{c|cc}
\cdot & 0 & 1 \\
\hline
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{array}
\]
Linear Feedback Shift-Register (LFSR)

- All the values are in GF(2) which means they can only be 0 or 1.
- The value of $g_i$ determines whether the output of the $k^{th}$ FF will be in the sum that produce the feedback bit.
  - 1 signifies closed or a connection and
  - 0 signifies open or no connection.
- Ex. Suppose $g_1 = 0, g_2 = 1, g_3 = 1$ in our LFSR.

\[
\begin{align*}
\text{m-sequence generator (1)}
\end{align*}
\]

- Start with a “primitive polynomial”
  - $g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_rx^r$
  - $r = \text{degree of the polynomial}$
- Use $r$ flip-flops.
- The feedback taps in the feedback shift register are selected to correspond to the coefficients of the primitive polynomial.
- Ex. $g(x) = 1 + x^2 + x^3$ is a primitive polynomial.
  \[
  = 1 + 0x + 1x^2 + 1x^3
  \]
  (Degree: $r = 3 \Rightarrow$ use 3 flip-flops)
m-sequence generator (2)

- We start with state 100.
  - You may choose different non-zero state.
  - Note that if we start with 000, we won’t go anywhere.

- Any polynomial generates periodic sequence.
  - The maximum period is $2^r - 1$.
- In this example, the state cycles through all $2^3 - 1 = 7$ non-zero states.

State Diagram

- m-sequence: 001011100101110010111...-periodic with period = 7
- circuit diagram: output

<table>
<thead>
<tr>
<th>Time</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(m-sequence: 001011100101110010111... (repeat))
Primitive Polynomial

- Definition: A LFSR generates an m-sequence if and only if (starting with any nonzero state,) it visits all possible nonzero states (in one cycle).
- Technically, one can define primitive polynomial using concepts from finite field theory.
- Fact: A polynomial generates m-sequence if and only if it is a primitive polynomial.
  - Therefore, we use this fact to define primitive polynomial.
- For us, a polynomial is primitive if the corresponding LFSR circuit generates m-sequence.

Sample Exam Question

Draw the complete state diagrams for linear feedback shift registers (LFSRs) using the following polynomials. Does either LFSR generate an m-sequence?

1. \( g(x) = 1 + x^2 + x^3 \)
2. \( g(x) = 1 + x + x^2 + x^3 \)
**Solution (1)**

Draw the complete **state diagrams** for linear feedback shift registers (LFSRs) using the following polynomials. Does either LFSR generate an m-sequence?

1. \( g(x) = 1 + x^2 + x^3 \)

The corresponding LFSR **generates an m-sequence** because the state diagram contains a cycle that visits all possible nonzero states. We can also conclude that \( g(x) = 1 + x^2 + x^3 \) is a **primitive polynomial**.

**Solution (2)**

\[ g(x) = 1 + x + x^2 + x^3 \]

The corresponding LFSR **does not create m-sequence**. \( g(x) \) is **not primitive**.
m-Sequences: More properties

1. The contents of the shift register will cycle over all possible $2^r-1$ nonzero states before repeating.
2. Contain one more 1 than 0 (Slightly unbalanced)
3. **Shift-and-add property**: Sum of two (cyclic-)shifted m-sequences is another (cyclic-)shift of the same m-sequence
4. If a window of width $r$ is slid along an m-sequence for $N = 2^r-1$ shifts, each $r$-tuple except the all-zeros $r$-tuple will appear exactly once
5. For any m-sequence, there are
   - One run of ones of length $r$
   - One run of zeros of length $r-1$
   - One run of ones and one run of zeroes of length $r-2$
   - Two runs of ones and two runs of zeros of length $r-3$
   - Four runs of ones and four runs of zeros of length $r-4$
   - ...
   - $2^{r-3}$ runs of ones and $2^{r-3}$ runs of zeros of length 1

---

m-Sequences: More Properties

1. The contents of the shift register will cycle over all possible $2^r-1$ nonzero states before repeating.
2. Each cycle contains exactly one more 1s than 0s (Slightly unbalanced)

$$g(x) = 1 + x^2 + x^3$$

period $= 2^r - 1 = 2^3 - 1 = 7$

m-Sequences: More Properties

3. **Shift-and-add property**: Sum of two (cyclic-)shifted m-sequences is another (cyclic-)shift of the same m-sequence

```
00101110010111001011100101110010111001011100101110010111
```

0 phase shift: 0010111
1 phase shift: 0101110
2 phase shift: 1011100
3 phase shift: 0111001
4 phase shift: 1110010
5 phase shift: 1100101
6 phase shift: 1001011

⊕ = 1100101

4. If a window of width $r$ is slid along an m-sequence for $N = 2^r - 1$ shifts, each $r$-tuple except the all-zeros r-tuple will appear exactly once

```
00101110010111001011100101110010111001011100101110010111
```


---

m-Sequences: More Properties

5. For any m-sequence, there are $2^{r-1}$ runs.
   - One run of ones of length $r$
   - One run of zeros of length $r-1$
   - One run of ones and one run of zeroes of length $r-2$
   - Two runs of ones and two runs of zeros of length $r-3$
   - Four runs of ones and four runs of zeros of length $r-4$
   - ... 
   - $2^{r-3}$ runs of ones and $2^{r-3}$ runs of zeros of length 1

In other words, relative frequency for runs of length $\ell$ is

\[
\frac{1}{2^\ell} \quad \text{if } \ell < r, \\
\frac{1}{2^{r-1}} \quad \text{if } \ell = r.
\]

There are 4 runs

\[ \frac{2}{4} = \frac{1}{2} \quad \text{of these are of length 1} \]

m-Sequences: Another Example

- $2^{5} - 1 = 31$-chip m-sequence
- The following sequence contains 16 runs

00011 1110010100 0010 0101 1110 11

- Rel. Freq of Run Lengths

<table>
<thead>
<tr>
<th>Run Length</th>
<th>Rel. Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1/16</td>
</tr>
<tr>
<td>4</td>
<td>1/16</td>
</tr>
<tr>
<td>3</td>
<td>2/16</td>
</tr>
<tr>
<td>2</td>
<td>4/16</td>
</tr>
<tr>
<td>1</td>
<td>8/16</td>
</tr>
</tbody>
</table>

- Rel. Freq of Runs

<table>
<thead>
<tr>
<th>Run</th>
<th>Rel. Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>1/16</td>
</tr>
<tr>
<td>00000</td>
<td>1/16</td>
</tr>
<tr>
<td>111</td>
<td>1/16</td>
</tr>
<tr>
<td>000</td>
<td>1/16</td>
</tr>
<tr>
<td>11</td>
<td>2/16</td>
</tr>
<tr>
<td>00</td>
<td>2/16</td>
</tr>
<tr>
<td>1</td>
<td>4/16</td>
</tr>
<tr>
<td>0</td>
<td>4/16</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
R_x[\tau] &= \sum_{n=-\infty}^{\infty} x[n]x[n - \tau] \\
&= \sum_{n=-\infty}^{\infty} x[n]x[n + \tau]
\end{align*}
\]

(Time) Autocorrelation Function for Energy Sequence

\[ x[n] = (0 \ 2 \ 4 \ 3 \ 2 \ 1 \ 0) \]

\[ R_x[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n-\tau] = \sum_{n=\tau}^{n+\tau} x[n]x[n+\tau] \]

\[ \tau = 1 \]

\[ x[n] = (0 \ 2 \ 4 \ 3 \ 2 \ 1 \ 0) \]

\[ R_x[\tau] = \sum_{n=-\infty}^{\infty} x[n]x[n-\tau] = \sum_{n=\tau}^{n+\tau} x[n]x[n+\tau] \]
MATLAB: \texttt{xcorr}

- \texttt{r = xcorr(x, y)}
  - Return the cross-correlation of two discrete-time sequences, \( x \) and \( y \).
  - If \( x \) and \( y \) have different lengths, the function appends zeros at the end of the shorter vector so it has the same length as the other.
  - The lag (\( \tau \)) is varied from \(- (N - 1)\) to \((N - 1)\) where \( N \) is the longer length of the two sequences.
- \([r, lags] = xcorr(__)\)
  - Also returns vector with the lags (\( \tau \)) at which the correlations are computed.

(Time) Autocorrelation Function for Energy Sequence

close all
x = [0 2 4 3 2 1 0];

% plot the signal
plot(x,'--','LineWidth',1.5)
hold on
plot(x,'o','LineWidth',1.5)
ylabel('x[n]')
xlabel('n')

% plot auto-correlation function
figure
[R lag] = \texttt{xcorr}(x,x);
plot(R,'--','LineWidth',1.5)
hold on
plot(R,'o','LineWidth',1.5)
ylabel('R_x[\tau]')
xlabel('\tau')
(Time) Autocorrelation Function for Power and Periodic Sequence

<table>
<thead>
<tr>
<th>Power Sequence</th>
<th>Time average $\langle x[n] \rangle$</th>
<th>Autocorrelation $R_x[\tau]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lim_{T \to \infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]$</td>
<td>$\langle x[n]x[n-\tau] \rangle = \lim_{T \to \infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]x[n-\tau]$</td>
<td>$\langle x[n]x[n+\tau] \rangle = \lim_{T \to \infty} \frac{1}{2T} \sum_{n=-T}^{T} x[n]x[n+\tau]$</td>
</tr>
</tbody>
</table>

Periodic Sequence with period $T_0$

$\frac{1}{T_0} \sum_{n=T_0}^{x[n]}$ $\frac{1}{T_0} \sum_{n=T_0}^{x[n]} x[n]x[n-\tau] = \frac{1}{T_0} \sum_{n=T_0}^{x[n]} x[n]x[n-\tau]$

Example: (Time) Autocorrelation Function for Periodic Sequence
Example: (Time) Autocorrelation Function for Periodic Sequence

\[ x[n] = 0 2 4 3 2 1 0 2 4 3 2 1 \]
\[ x[n-0] = 0 2 4 3 2 1 0 2 4 3 2 1 \]
\[ x[n] = 0 2 4 3 2 1 0 2 4 3 2 1 \]
\[ x[n-1] = 1 0 2 4 3 2 1 0 2 4 3 2 1 \]
\[ R_x[\tau] = 5.67 \]

\[ \sum_{\tau} 34 \times \frac{1}{6} = 5.67 \]

\[ \sum_{\tau} 28 \times \frac{1}{6} = 4.67 \]

\[ \cdots \]

Back to m-Sequences

\[ c[n]: 00101110010111001011100101110010111001011100101110010111 \]

\[ 001011 \]

\[ \oplus \]

\[ 1001011 \]

In actual transmission, we will map “0 and 1” to “+1 and -1”, respectively.

\[ 0 \oplus 0 = 0 \]
\[ 0 \oplus 1 = 1 \]
\[ 1 \oplus 0 = 1 \]
\[ 1 \oplus 1 = 0 \]

\[ -1 \times 1 = 1 \]
\[ 1 \times -1 = 1 \]
\[ 1 \times 1 = 1 \]
\[ -1 \times -1 = 1 \]

Easy to see that \( * \) is multiplication.
Back to m-Sequences

$c[n]$: 00101110010111001011100101110010111001011100101110010111001011100101110010111

From the previous slide, the mapping that we will use is:

\[
\begin{align*}
0 & \rightarrow +1 \\
1 & \rightarrow -1 \\
\oplus & \rightarrow x
\end{align*}
\]

One more 1s than 0s

1001011

In actual transmission, we will map “0 and 1” to “+1 and -1”, respectively.

Autocorrelation when not aligned:

\[
\begin{align*}
-1 & \quad 1 \quad -1 \quad -1 \quad 1 \quad 1 \\
1 & \quad 1 \quad -1 \quad 1 \quad -1 \quad -1
\end{align*}
\]

\[
-1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1 \quad -1 \rightarrow \sum = -1 \quad \times \quad \frac{1}{7} \quad = \quad -\frac{1}{7}
\]

One more “-1”s than 1s

m-Sequences: Autocorrelation function

\[
R_c[\tau] = \frac{1}{N}\sum_{n=0}^{N-1} c[n]c[n+\tau]
\]

\[
\begin{align*}
c[n] & \quad (N=63) \\
R_c[\tau] & \quad (N=63)
\end{align*}
\]

-1/7
m-Sequences: Autocorrelation function

Autocorrelation Function for Periodic Binary Random Sequence

Consider a periodic sequence whose one period is given by
\[
[-1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ -1 \ -1]
\]

The shift property of binary random sequence implies that
\[
R_x[\tau] = \langle x[n] x[n-\tau] \rangle
\]

\[
\xrightarrow{\text{LLN}} \mathbb{E}[x[n] x[n-\tau]]
\]

\[
= 1 \times \frac{1}{2} + (-1) \times \frac{1}{2} = 0
\]
Consider a periodic sequence whose one period is given by 
$1-2*\text{randi}([0 \ 1],1,100)$

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle$$

$$\xrightarrow{n \to \infty \text{ LLN}} \mathbb{E}[x[n]x[n-\tau]]$$

$$= 1 \times \frac{1}{2} + (-1) \times \frac{1}{2} = 0$$

Consider a periodic sequence whose one period is given by 
$1-2*\text{randi}([0 \ 1],1,1000)$

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle$$

$$\xrightarrow{n \to \infty \text{ LLN}} \mathbb{E}[x[n]x[n-\tau]]$$

$$= 1 \times \frac{1}{2} + (-1) \times \frac{1}{2} = 0$$
Example: Autocorrelation Function for Periodic Binary Random Sequence

Consider a periodic sequence whose one period is given by

$$1 - 2 \cdot \text{randi}([0, 1], 1, 10000)$$

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle$$

$$\xrightarrow{LLN} \mathbb{E}[x[n]x[n-\tau]]$$

$$= 1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2} = 0$$

Autocorrelation Function for Periodic Binary Random Sequence

Consider a periodic sequence whose one period is given by

$$1 - 2 \cdot \text{randi}([0, 1], 1, 100000)$$

The shift property of binary random sequence implies that

$$R_x[\tau] = \langle x[n]x[n-\tau] \rangle$$

$$\xrightarrow{LLN} \mathbb{E}[x[n]x[n-\tau]]$$

$$= 1 \cdot \frac{1}{2} + (-1) \cdot \frac{1}{2} = 0$$
Autocorrelation and PSD

- (Normalized) autocorrelations of maximal sequence and random binary sequence.

\[ R(t) = \sum_{n=-N}^{N} x(t) x(t+n) \]

where the integration is over any period, \( T_0 = NT_c \).

\[ S(f) = \sum_{m=-N}^{N} P_m \delta(f - m f_0) \]

where

\[ P_m = \left\{ \begin{array}{ll}
\frac{\left[(N+1)/N^2\right] \text{sinc}^2(m/N)}{1/N^2, \; m = 0} & \\
1/N^2, & m \neq 0
\end{array} \right. \]

Power spectral density of maximal sequence.

References: m-sequences

  - Page 84-90
  - Chapter 1 and 2
- Goldsmith, *Wireless Communications*, 2005
  - Chapter 13
  - Section 3.4.3
Review: m-sequence

DSSS: $m(t) \times c(t)$

Spectral spreading waveform

Spreading code/sequence

\[
\begin{align*}
c[n] & = 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1
\end{align*}
\]

One important collection of these is the collection of \textbf{m-sequences}.

Generated with LFSR whose connections corresponds to coefficients of primitive polynomials. The resulting sequence achieves the maximum period (length) of \( N = 2^r - 1 \) where \( r \) is the degree of primitive polynomial.
4.4 m-sequence
(Additional Remarks)

Example: (Time) Autocorrelation Function for Periodic Sequence

\[ x[n] = \{0, 2, 4, 3, 2, 1, 0, 2, 4, 3, 2, 1\} \]
\[ x[n-0] = \{0, 2, 4, 3, 2, 1, 0, 2, 4, 3, 2, 1\} \]
\[ x[n] = \{0, 2, 4, 3, 2, 1, 0, 2, 4, 3, 2, 1\} \]
\[ x[n-1] = \{1, 0, 2, 4, 3, 2, 1, 0, 2, 4, 3, 2\} \]

\[ R_x[\tau] = \sum x[n]x[n-\tau] \]

Office Hours:
BKD, 6th floor of Sirindhralai building
Tuesday 14:20-15:20
Wednesday 14:20-15:20
Friday 9:15-10:15
Example: (Time) Autocorrelation Function for Periodic Sequence

\[ x[n] = 0 \ 2 \ 4 \ 3 \ 2 \ 1 \ 0 \ 2 \ 4 \ 3 \ 2 \ 1 \]
\[ x[n-0] = 0 \ 2 \ 4 \ 3 \ 2 \ 1 \]
\[ x[n-1] = 1 \ 0 \ 2 \ 4 \ 3 \ 2 \]

\[ R_x[\tau] \]

\[ \sum \]

\[ 34 \times \frac{1}{6} \]

\[ 5.67 \]

\[ 4.67 \]

\[ \tau \]

\[ \cdots \]
Example: (Time) Autocorrelation Function for Periodic Sequence

Let’s call this the “sumproduct” operation. (This exact name is used in Excel for this kind of operation.) Mathematically, this is simply the dot product between two real-valued vectors.

Let’s call this scaling the “normalization” operation.

So, the combined computation can be called “normalized sumproduct” operation. We may also refer to this as the (sliding computation of) “autocorrelation” operation as well.

Sliding computation of autocorrelation

m-sequence (repeated 5 times)

m-sequence (1 period)
Sliding computation of autocorrelation

m-sequence (repeated 5 times)

\[ c[n] \]

m-sequence (1 period)

autocorrelation
Sliding computation of autocorrelation

m-sequence (repeated 5 times)

\[ c[n] \]

autocorrelation

Sliding computation of autocorrelation

m-sequence (repeated 5 times)

\[ c[n] \]

autocorrelation
Sliding computation of autocorrelation

m-sequence (repeated 5 times)

c[n]

autocorrelation

Sliding computation of autocorrelation

m-sequence (repeated 5 times)

c[n]

autocorrelation
Sliding computation of autocorrelation

\[ c[n] \]

m-sequence (repeated 5 times)

autocorrelation

\[ c[n] \]
The signal received consists of a number of reflected rays, each characterized by a different amount of attenuation and delay.

\[ y(t) = x(t) * h(t) + n(t) = \sum_{i=0}^{V} \beta_i x(t - \tau_i) + n(t) \]

\[ h(t) = \sum_{i=0}^{V} \beta_i \delta(t - \tau_i) \]

\[ h_1(t) = 0.5 \delta(t) + 0.2 \delta(t - 0.2T_s) + 0.3 \delta(t - 0.3T_s) + 0.1 \delta(t - 0.5T_s) \]

\[ h_2(t) = 0.5 \delta(t) + 0.2 \delta(t - 0.7T_s) + 0.3 \delta(t - 1.5T_s) + 0.1 \delta(t - 2.3T_s) \]

Here, let’s consider the discrete-time version of fading:

\[ y[n] = \sum_{i=0}^{V} \beta_i c[n - \tau_i] \]

In particular, let’s try

\[ y[n] = 5c[n] - 3c[n - 4] + c[n - 10] \]
Identifying Parameters of Multipath Fading via Autocorrelation

\[ y[n] = 5c[n] - 3c[n - 4] + c[n - 10] \]

Identifying Parameters of Multipath Fading via Autocorrelation

\[ y[n] = 5c[n - 7] - 3c[n - 11] + c[n - 17] \]
4.5 Cyclic Codes

Note that this topic is not directly related to DSSS nor multiple access. It is a kind of error control codes. However, the technique used are quite similar to the generation of m-sequence and hence we would like to discuss it here.

MATLAB: circcshift

- \( r' = \text{circshift}(r, [0, \Delta]) \)
- \( r' = \text{circshift}(r, \Delta, 2) \)
  circularly shifts the elements in a row vector \( r \) to the right by \( \Delta \) positions.
  - \( \text{circshift}([1 2 3 4 5], [0 3]) = [3 4 5 1 2] \)

- \( \vec{v}' = \text{circshift}(\vec{v}, \Delta) \)
- \( \vec{v}' = \text{circshift}(\vec{v}, [\Delta, 0]) \)
- \( \vec{v}' = \text{circshift}(\vec{v}, \Delta, 1) \)
  circularly shifts the elements in a column vector \( \vec{v} \) down by \( \Delta \) positions.
MATLAB: demo

```
>> r = 1:5
r =
    1    2    3    4    5

>> circshift(r,[0,3])
ans =
    3    4    5    1    2

>> circshift(r,3,2)
ans =
    3    4    5    1    2

>> circshift(r,3)
Warning: CIRCSHIFT(X,K) with scalar K and where size(X,1)==1
will change behavior in future versions. To retain current
behavior, use CIRCSHIFT(X,[K,0]) instead.
ans =
    1    2    3    4    5
```

MATLAB: demo

```
>> v = (1:5)'
v =
    1
    2
    3
    4
    5

>> circshift(v,[3,0])
ans =
    3
    4
    5
    1
    2

>> circshift(v,3,1)
ans =
    3
    4
    5
    1
    2

>> circshift(v,3)
ans =
    3
    4
    5
    1
    2
```
Linear Cyclic Codes

- Definition: A linear code is cyclic if a cyclic shift of any valid codeword is still a valid codeword.
  - Lead to more practical implementation.
  - Allow their encoding and decoding functions to be of much lower complexity than the matrix multiplications.
- Block codes used in FEC systems are almost always cyclic codes [C&C, 2009, p. 611][G, 2005, p. 220].
- CRC = cyclic redundancy check
  - Invented by W. Wesley Peterson in 1961.

Ex. Codebook of a Systematic Cyclic Code

- \( k = 4 \) bits in \( m = 7 \)
- \( n = 7 \) bits in \( c = 7 \)

<table>
<thead>
<tr>
<th>message</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000000</td>
</tr>
<tr>
<td>0011</td>
<td>1010001</td>
</tr>
<tr>
<td>0101</td>
<td>110010</td>
</tr>
<tr>
<td>0110</td>
<td>100011</td>
</tr>
<tr>
<td>0111</td>
<td>100111</td>
</tr>
<tr>
<td>1000</td>
<td>110110</td>
</tr>
<tr>
<td>1001</td>
<td>111000</td>
</tr>
<tr>
<td>1010</td>
<td>100101</td>
</tr>
<tr>
<td>1011</td>
<td>110110</td>
</tr>
<tr>
<td>1100</td>
<td>101100</td>
</tr>
<tr>
<td>1101</td>
<td>100010</td>
</tr>
<tr>
<td>1110</td>
<td>101110</td>
</tr>
<tr>
<td>1111</td>
<td>111111</td>
</tr>
</tbody>
</table>
Associating Vectors with Polynomials

Note that the index starts with 0.

\[ c = (c_0, c_1, c_2, \ldots, c_i, \ldots, c_{n-2}, c_{n-1}) \]

\[ c(x) = c_0 + c_1x + c_2x^2 + \cdots + c_ix^i + \cdots + c_{n-1}x^{n-1} \]

arbitrary variable

Example

\[ c = 1010011 \leftrightarrow c(x) = 1 + 0x + 1x^2 + 0x^3 + 0x^4 + 1x^5 + 1x^6 \]

Similarly,

\[ m = (m_0, m_1, \ldots, m_{k-1}) \leftrightarrow m(x) = m_0 + m_1x + m_2x^2 + \cdots + m_{k-1}x^{k-1} \]

Message polynomial

The powers of \( x \) denote the positions of the bits represented by the corresponding coefficients.

Each codeword has \( n \) bits. So, the degree of \( c(x) \) is \( n - 1 \).

Each message block has \( k \) bits. So, the degree of \( m(x) \) is \( k - 1 \).

Long Division (for numbers)

\[ \begin{array}{c|ccccc|c} \hline \text{divisor} & 6 & \text{dividend} & 83 \\ \hline \text{quotient} & 13 \\ \hline \text{dividend} & 6 & 23 \\ \hline \text{quotient} & 18 \\ \hline \text{remainder} & 5 \\ \hline \end{array} \]

Many way to write equations that describe the results:

- \( 83 = 6 \times 13 + 5 \)
- \( \frac{83}{6} = 13 + \frac{5}{6} \)
- \( 83 \equiv 5 \pmod{6} \)

- Dividing 78 by 6 leaves no remainder
- \( 78 \equiv 0 \pmod{6} \)
- 78 is a multiple of 6
- 6 divides 78
- 6 is a divisor of 78
- 78 is divisible by 6
- 6 is a factor of 78
- 6 \( \mid \) 78
Polynomial (Long) Division

\[ \frac{x^3 - 2x^2 + 0x - 4}{x - 3} = x^2 + x - 1 + \frac{5}{x - 3} \]

Many ways to write equations that describe the results:

\[ x^3 - 2x^2 - 4 = (x - 3)(x^2 + x + 3) + 5 \]

\[ x^3 - 2x^2 - 4 = x^2 + x + 3 + \frac{5}{x - 3} \]

\[ x^3 - 2x^2 - 4 \equiv 5 \pmod{(x - 3)} \]

[https://en.wikipedia.org/wiki/Polynomial_long_division]

---

Polynomial (Long) Division in GF(2)

\[ \frac{x^6 + x^5 + x^2 + 1}{x^3 + x^2 + 1} = x^3 + 1 + \frac{1}{x^3 + x^2 + 1} \]

\[ x^6 + x^5 + x^2 + 1 \equiv 0 \pmod{(x^3 + x^2 + 1)} \]

\[ x^6 + x^5 + x^3 + x^2 + 1 \equiv (x^2 + 1) \pmod{(x^3 + x^2 + 1)} \]

 mod 2 addition and multiplication

Have only 0, 1 (no 2, 3, 4, ... no negative numbers)

\[ 1 + 1 = 0 \]

No remainder

The degree here is already < 3. So, we stop the long division.
Generator Polynomial

- Cyclic codes are generated via a **generator polynomial** instead of a generator matrix.
- \( g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k} \)
- Degree = \( n - k \)
- \( g_0 = g_{n-k} = 1 \)
- Is a divisor of \( x^n - 1 \).
- \( c(x) \) is a valid codeword iff \( g(x) \) divides \( c(x) \) with no remainder.

Two popular ways to generate cyclic code:

1. Non-systematic: \( c(x) = m(x)g(x) \)
2. Systematic: \( c(x) = x^{n-k}m(x) + r(x) \)

They give different codes.

If systematic coding is required, then \( \boxed{2} \) is the method of choice.

Example

- Consider a cyclic code with generator polynomial
  \[ g(x) = 1 + x^2 + x^3. \]
- Determine if the codeword described by each of the following polynomials is a valid codeword for this generator polynomial.
  - \( c_1(x) = 1 + x^2 + x^5 + x^6 \)
    - \( g(x) \) divides \( c(x) \) with no remainder \( \Rightarrow c_1(x) \) corresponds to a codeword.
  - \( c_2(x) = 1 + x^2 + x^3 + x^5 + x^6 \)
    - Look at \( c_2(x) \). There is a remainder of \( x^2 + 1 \) in this division. Therefore, \( c_2(x) \) does not correspond to a valid codeword.
Generation of Systematic Cyclic Code

\[ c(x) = x^{n-k}m(x) \oplus r(x) \]

- Three steps:
  1. Multiply the message polynomial \( m(x) \) by \( x^{n-k} \)
  2. Divide \( x^{n-k}m(x) \) by \( g(x) \) to get the remainder polynomial \( r(x) \).
  - \( r(x) \equiv x^{n-k}m(x) \pmod{g(x)} \)
  3. Subtract (add) \( r(x) \) from (to) \( x^{n-k}m(x) \)
- The polynomial multiplications are straightforward to implement, and the polynomial division is easily implemented with a feedback shift register.
- Thus, codeword generation for systematic cyclic codes has very low cost and low complexity.

Generation of Systematic Cyclic Code

\[ c(x) = x^{n-k}m(x) - r(x) \]

- \( x^{n-k}m(x) \)
  - Shift the message bits to the \( k \) rightmost digits of the codewords
  - The first \( n-k \) bits are “blank”
    - These \( n-k \) bits are to be “filled” by \( r(x) \).
- By construction,
  - \( \deg(r(x)) < \deg(g(x)) = n-k \)
    - \( \deg(r(x)) \leq n - k - 1 \)
    - Correspond to \( n-k \) bits.
  - \( \frac{x^{n-k}m(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)} \)
  - \( x^{n-k}m(x) - r(x) = q(x)g(x) \)
Example

- Consider a systematic cyclic (7,4) code whose generator polynomial is $g(x) = 1 + x + x^3$.
- Suppose the message is 0011. Find the corresponding codeword.

$$m = 0011 \iff m(x) = 0 + 0x + 1x^2 + 1x^3$$
$$= x^2 + x^3$$

$$m(x) = x^3 m(x) = x^5 + x^6$$

$$= 0 + 0x + 0x^2 + 0x^3 + x^4 + x^5 + 1x^6$$

$$c(x) = x^{n-k} m(x) + r(x)$$
$$= 0 + 1x + 0x^2 + 0x^3 + x^4 + 1x^5 + 1x^6$$

$$s_x = c = 0100011$$

References: Cyclic Codes

  - [TK5101 L333 2009]
  - Section 15.4 p. 918-923
  - [TK5102.5 C3 2010]
  - Section 13.2 p. 611-616
- Goldsmith, *Wireless Communications*, 2005
  - Section 8.2.4 p. 220-222
DSSSS and m-sequences

- m-sequences
  - Excellent auto-correlation properties (for ISI rejection)
  - Highly suboptimal for exploiting the multiuser capabilities of spread spectrum.

- There are only a small number of maximal length codes of a given length.

- Moreover, maximal length codes generally have relatively poor cross-correlation properties, at least for some sets of codes.

\[
\text{auto-correlation} = \frac{1}{\text{period}} \sum_{i=-\text{period}}^{\text{period}} x_i x_{i-t}
\]

\[
\text{cross-correlation} = \frac{1}{\text{period}} \sum_{i=-\text{period}}^{\text{period}} x_i y_{i-t}
\]

[Goldsmith, 2005, Ch 13]
Number of primitive polynomials

Number of different primitive polynomials:

- $r$ is the degree of the primitive polynomials and $N_p$ is the number of different primitive polynomials available.

<table>
<thead>
<tr>
<th>$r$</th>
<th>$N_p$</th>
<th>$r$</th>
<th>$N_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>11</td>
<td>176</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>12</td>
<td>144</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>13</td>
<td>630</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>14</td>
<td>756</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>15</td>
<td>1800</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>16</td>
<td>2048</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>17</td>
<td>7710</td>
</tr>
<tr>
<td>9</td>
<td>48</td>
<td>18</td>
<td>8064</td>
</tr>
<tr>
<td>10</td>
<td>60</td>
<td>19</td>
<td>27594</td>
</tr>
</tbody>
</table>

[Chen, 2007, p 145]

SSMA

- For spread spectrum systems with multiple users, codes such as Gold, Kasami, or Walsh codes are used instead of maximal length codes.
- Superior cross-correlation properties.
- Worse auto-correlation than maximal-length codes.
  - The autocorrelation function of the spreading code determines its multipath rejection properties.
Qualcomm

- Founders: Two of the most eminent engineers in the world of mobile radio
- Prof. Irwin Jacobs is the chairman and founder
  - Cornell (undergrad.: Hotel > EE)
  - MIT (grad.)
  - UCSD (Prof.)
- Prof. Andrew J. Viterbi is the co-founder
  - MIT (BS, MS)
  - USC (PhD)
  - UCLA and UCSD (Prof.)
  - Same person that invented the Viterbi algorithm for decoding convolutionally encoded data.

Video: Irwin Jacobs

- Irwin Jacobs: Pioneer of the Wireless Future

- Gallagher’s remark on ideal engineer: 1:46-2:24
- Educational background: 5:00-8:25
- Textbook: 9:03-10:40
- Viterbi: 11:00
- CDMA:
  - 26:14-26:50
  - 28:46-31:20

[http://www.youtube.com/watch?v=EGaG1S4-D6o]
Video: Irwin Jacobs

With Gallager's remark on being an ideal engineer

Video: Irwin Jacobs

Educational background
Video: Irwin Jacobs

Textbook

Video: Irwin Jacobs

Viterbi
At Cornell...

- Toby Berger was the Irwin and Joan Jacobs Professor in Engineering from 1997 to 2005.
- Berger retired from Cornell after the fall 2005 semester.
- From 2006, Lang Tong replaces Toby Berger as the Irwin and Joan Jacobs Professor in Engineering.

Video: Irwin Jacobs
Code Division Multiple Access (CDMA)

- **1991**: Qualcomm announced
  - that it had invented a new cellular system based on CDMA
  - that the capacity of this system was **20 or so times greater** than any other cellular system in existence
- However, not all of the world was particularly pleased by this apparent breakthrough—in particular, GSM manufacturers became concerned that they would start to lose market share to this new system.
  - The result was continual and vociferous argument between Qualcomm and the GSM manufacturers.

CDMA

- One way to achieve SSMA
- May utilize Direct Sequence Spread Spectrum (DS/SS)
  - The narrowband message signal is multiplied (modulated) by the spreading signal which has a very large bandwidth (orders of magnitudes greater than the data rate of the message).
  - Direct sequence is not the only spread-spectrum signaling format suitable for CDMA
- All users use the same carrier frequency and may transmit simultaneously.
- Users are assigned different “signature waveforms” or “code” or “codeword” or “spreading signal”
- Each user’s codeword is approximately orthogonal to all other codewords.
- Should not be confused with the mobile phone standards called cdmaOne (Qualcomm’s IS-95) and CDMA2000 (Qualcomm’s IS-2000) (which are often referred to as simply "CDMA")
  - These standards use CDMA as an underlying channel access method.

Inner Product (Cross Correlation)

- Vector
  - Column vector
  - \[ \langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \mathbf{y}^* = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}^* = \sum_{k=1}^{n} x_k y_k^* \]
- Complex conjugate
- Conjugation is not required when dealing only with real-valued signals.

- Waveform: Time-Domain
  - \[ \langle x, y \rangle = \int_{-\infty}^{\infty} x(t) y^*(t) dt \]

- Waveform: Frequency Domain
  - \[ \langle X, Y \rangle = \int_{-\infty}^{\infty} X(f) Y^*(f) df \]
Review: Orthogonality

- Two signals are said to be **orthogonal** if their inner product is zero.

- The symbol $\langle \cdot, \cdot \rangle$ is used to denote orthogonality.

Vector:
$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a} \cdot \mathbf{b}^* = a_1 b_1^* + \cdots + a_n b_n^* = \sum_{k=1}^{n} a_k b_k^* = 0$$

Example:
$$2t + 3 \quad \text{and} \quad 5t^2 + t - \frac{17}{9} \quad \text{on} \quad [-1,1]$$

Time-domain:
$$\langle a, b \rangle = \int_{-\infty}^{\infty} a(t) b^*(t) dt = 0$$

Example (Fourier Series):
- $\sin \left(2\pi k_1 \frac{t}{T}\right)$ and $\cos \left(2\pi k_2 \frac{t}{T}\right)$ on $[0,T]$
- $e^{j2\pi n \frac{t}{T}}$ on $[0,T]$

Frequency domain:
$$\langle A, B \rangle = \int_{-\infty}^{\infty} A(f) B^*(f) df = 0$$

Ex: Orthogonal Signals

Waveform (cont.-time) version

Vector (discrete-time) version

Check $(\frac{1}{2}) = \mathbf{c}$ pairs

- $\mathbf{c}^{(1)} = [+1 \quad +1 \quad +1 \quad +1]$
- $\mathbf{c}^{(2)} = [+1 \quad +1 \quad -1 \quad -1]$
- $\mathbf{c}^{(3)} = [+1 \quad -1 \quad -1 \quad +1]$
- $\mathbf{c}^{(4)} = [+1 \quad -1 \quad +1 \quad -1]$

Ex. $\langle \mathbf{c}^{(1)}, \mathbf{c}^{(4)} \rangle = 1 - 1 - 1 + 1 = 0$

When $i \neq j$, $\langle c_i(t), c_j(t) \rangle = 0$.

When $i \neq j$, $\langle \mathbf{c}^{(i)}, \mathbf{c}^{(j)} \rangle = 0$.

Ex. $\langle c_2, c_3 \rangle = \int c_2 c_3 dt = \int c_4 dt = 0$
Review: Important Properties

- Parseval’s theorem

\[ \langle x, y \rangle = \int_{-\infty}^{\infty} x(t)y^*(t)dt = \int_{-\infty}^{\infty} X(f)Y^*(f)df \equiv \langle X, Y \rangle \]

1. \( x(t) \perp y(t) \) iff \( X(f) \perp Y(f) \).
2. \( E_x = \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |X(f)|^2 df \).

- Useful observation: If the non-zero regions of two signals do not overlap in time domain or do not overlap in frequency domain, then the two signals are orthogonal (their inner product = 0).
- However, in general, orthogonal signals may overlap both in time and in frequency domain.

Orthogonality-Based MA

- Consider a system with \( L \) users.
- Suppose that the \( k \)th user want to transmit a number \( S_k \).
  - Could be a sample from his/her analog message.
  - Could be -1 or 1, representing message bit 1 or 0.
- We create multiple communication channels (with no inter-channel interference); one for each user.

\[ x(t) = \sum_{k=0}^{K-1} S_k c_k(t) \xrightarrow{\mathcal{F}} X(f) = \sum_{k=0}^{K-1} S_k C_k(f) \]

- The \( k \)th code (signal/waveform) is assigned to (used by) the \( k \)th user.
Orthogonality-Based MA

CDMA
\[ x(t) = \sum_{k=0}^{\ell-1} s_k c_k(t) \rightarrow X(f) = \sum_{k=0}^{\ell-1} s_k C_k(f) \]

(Orthogonal signaling)
where \( c_{k_1} \perp c_{k_2} \)

TDMA
\[ x(t) = \sum_{k=0}^{\ell-1} s_k c(t - kT_s) \rightarrow X(f) = C(f) \sum_{k=0}^{\ell-1} s_k e^{-j2\pi kT_s} \]

where \( c(t) \) is time-limited to \([0, T]\).

This is a special case of CDMA with \( c_k(t) = c(t - kT_s) \)

The \( c_k \) are non-overlapping in time domain.

FDMA
\[ X(f) = \sum_{k=0}^{\ell-1} s_k C(f - k\Delta f) \]

where \( C(f) \) is frequency-limited to \([0, \Delta f]\).

This is a special case of CDMA with \( C_k(f) = C(f - k\Delta f) \)

The \( C_k \) are non-overlapping in freq. domain.

Ex: DS-CDMA

Waveform (cont.-time) version

Vector (discrete-time) version

\[ x(t) = \sum_{k=1}^{4} s_k c_k(t) \]

\[ \mathbf{c}^{(1)} = [+1 \ +1 \ +1 \ +1] \]
\[ \mathbf{c}^{(2)} = [+1 \ +1 \ -1 \ -1] \]
\[ \mathbf{c}^{(3)} = [+1 \ -1 \ -1 \ +1] \]
\[ \mathbf{c}^{(4)} = [+1 \ -1 \ +1 \ -1] \]

\[ \langle \mathbf{c}^{(i)}, \mathbf{c}^{(j)} \rangle = \begin{cases} 4, & i = j, \\ 0, & i \neq j. \end{cases} \]

\[ \mathbf{x} = \sum_{k=1}^{4} s_k \mathbf{c}^{(k)} \]
Ex: DS-CDMA (Uplink)

At the transmitter (mobile phone) of each user:

- User 1’s message is $s_1 = 14$.
  - Transmit $s_1 \times \mathbf{c}^{(1)} = 14 \times [1, 1, 1, 1] = [14, 14, 14, 14]$.
- User 2’s message is $s_2 = 20$.
  - Transmit $s_2 \times \mathbf{c}^{(2)} = 20 \times [1, 1, -1, -1] = [20, 20, -20, -20]$.
- User 3’s message is $s_3 = 26$.
  - Transmit $s_3 \times \mathbf{c}^{(3)} = 26 \times [+1, -1, -1, +1] = [26, -26, -26, 26]$.
- User 4’s message is $s_4 = -5$.
  - Transmit $s_4 \times \mathbf{c}^{(4)} = -5 \times [+1, -1, +1, -1] = [-5, 5, -5, 5]$.

In the air, the signals from all the users are combined to create

$$x = \sum_{k=0}^{\ell-1} s_k \mathbf{c}^{(k)} = [55, 13, -37, 25]$$

Additionally, the signal may be further corrupted by the noise and fading.

- $\mathbf{r} = \mathbf{H}x + \mathbf{n}$
- However, here, we will ignore such corruption for clearer MA calculation.

---

Ex: DS-CDMA (Uplink)

At the receiver (base station),

$$\mathbf{r} = x = \sum_{k=1}^{4} s_k \mathbf{c}^{(k)} = [s_1, s_2, s_3, s_4] = s \mathbf{c}$$

To find $s_3$, note that

$$\langle \mathbf{r}, \mathbf{c}^{(3)} \rangle = \mathbf{r} \cdot \mathbf{c}^{(3)} = \left( \sum_{k=1}^{4} s_k \mathbf{c}^{(k)} \right) \cdot \mathbf{c}^{(3)} = \sum_{k=1}^{4} s_k (\mathbf{c}^{(k)} \cdot \mathbf{c}^{(3)})$$

$$\mathbf{r} \cdot \mathbf{c}^{(3)} = 4 \cdot \mathbf{c}^{(3)} \cdot \mathbf{c}^{(3)} = 4 \mathbf{c}_3$$

$$\hat{s}_3 = \frac{1}{4} \langle \mathbf{r}, \mathbf{c}^{(3)} \rangle$$

In general, for orthogonal codes containing only $\pm 1$,

$$\hat{s}_k = \frac{1}{\text{length of the code}} \langle \mathbf{r}, \mathbf{c}^{(k)} \rangle$$

Observe to recover $s_k$, we only need $\mathbf{c}^{(k)}$; we don’t need to know the codes for other users.
Ex: DS-CDMA (Uplink)

- One can define \( \hat{\mathbf{s}} = [\hat{s}_1 \; \hat{s}_2 \; \hat{s}_3 \; \hat{s}_4] \).
- Then,

\[
\hat{\mathbf{s}} = [\hat{s}_1 \; \hat{s}_2 \; \hat{s}_3 \; \hat{s}_4] = \left[ \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(1)} \; \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(2)} \; \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(3)} \; \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(4)} \right]
\]

\[
= \frac{1}{N} \mathbf{r} \left[ (\mathbf{c}^{(1)})^T \; (\mathbf{c}^{(2)})^T \; (\mathbf{c}^{(3)})^T \; (\mathbf{c}^{(4)})^T \right] = \frac{1}{N} \mathbf{r} \mathbf{C}^T
\]

CDMA’s key equation: \( \mathbf{s} = \frac{1}{N} (\mathbf{sC}) \mathbf{C}^T = (\underline{\mathbf{sC}}) \left( \frac{1}{N} \mathbf{C}^T \right) \)

Key property of \( \mathbf{C} \)

- From the CDMA’s key equation \( \mathbf{s} = \frac{1}{N} (\mathbf{sC}) \mathbf{C}^T \), or from the fact that all the rows of \( \mathbf{C} \) are orthogonal,
- we have the key property of \( \mathbf{C} \): 
  \( \mathbf{CC}^T = N\mathbf{I} \).
- It is tempting to call this an orthogonal matrix.
  - However, in linear algebra, to have an orthogonal matrix, the matrix must satisfy
    1. the rows are orthogonal and
    2. the rows must be unit vectors.

In other words, the rows must be orthonormal vectors. Equivalently, The matrix must satisfy \( \mathbf{AA}^T = \mathbf{A}^T \mathbf{A} = \mathbf{I} \).
DS-CDMA: Uplink vs. Downlink

The BS receives $\mathbf{r} = s \mathbf{c}$. The message from the $k^{th}$ user can be recovered via $\hat{s}_k = \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(k)}$. Alternatively, can recover all messages simultaneously: $\hat{s} = \frac{1}{N} \mathbf{r} \mathbf{c}^T$.

BS wants to transmit $s_k$ to the $k^{th}$ user. $(s_1$ to the 1st user, $s_2$ to the 2nd user, …)
BS transmits $\mathbf{x} = \sum_k s_k \mathbf{c}^{(k)} = \mathbf{s} \mathbf{c}$

The transmitted signals $s_k \mathbf{c}^{(k)}$ are difficult to sync or align. The $k^{th}$ user (MS) wants to send $s_k$. The $k^{th}$ user (MS) transmits $s_k \mathbf{c}^{(k)}$. Each user (MS) receives $\mathbf{r} = s \mathbf{c}$. The $k^{th}$ user (MS) can recover its message from $\hat{s}_k = \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(k)}$.

CDMA: DS/SS

- The receiver performs a time correlation operation to detect only the specific desired codeword.
- All other codewords appear as noise due to decorrelation.
- For detection of the message signal, the receiver needs to know the codeword used by the transmitter.
- Each user operates independently with no knowledge of the other users.
- Unlike TDMA or FDMA, CDMA has a soft capacity limit.
  - Increasing the number of users in a CDMA system raises the noise floor in a linear manner.
  - There is no absolute limit on the number of users in CDMA. Rather, the system performance gradually degrades for all users as the number of users is increased and improves as the number of users is decreased.
An analogy [Tanenbaum, 2003]

- An airport lounge with many pairs of people conversing.
- TDMA is comparable to all the people being in the middle of the room but taking turns speaking.
- FDMA is comparable to the people being in widely separated clumps, each clump holding its own conversation at the same time as, but still independent of, the others.
- CDMA is comparable to everybody being in the middle of the room talking at once, but with each pair in a different language.
  - The French-speaking couple just hones in on the French, rejecting everything that is not French as noise.
  - Thus, the key to CDMA is to be able to extract the desired signal while rejecting everything else as random noise.
CDMA: Near-Far Problem

- At first, CDMA did not appear to be suitable for mobile communication systems because of this problem.
- Occur when many mobile users share the same channel.
- In an uplink, the signals received from each user at the receiver travel through different channels.
- Users that are close to the BS can cause a great deal of interference to user’s farther away.
  - In general, the strongest received mobile signal will capture the demodulator at a base station.
  - Stronger received signal levels raise the noise floor at the base station demodulators for the weaker signals, thereby decreasing the probability that weaker signals will be received.
- Fast power control mechanisms solve this problem.
  - Regulate the transmit power of individual terminals in a manner that received power levels are balanced at the BS.

How many orthogonal signals?

- No signal can be both strictly time-limited and strictly band-limited.
- We adopt a softer definition of bandwidth and/or duration (e.g., the percentage of energy outside the band [-B, B] or outside the time interval [0, T] not exceeding a given bound $\varepsilon$.
- Q: How many mutually orthogonal signals with (approximate) duration T and (approximate) bandwidth B can be constructed?
- A: About $2TB$
  - No explicit answer in terms of T, B, and $\varepsilon$ is known.
  - Unless the product TB is small.
- A K-user orthogonal CDMA system employing antipodal modulation at the rate of R bits per second requires bandwidth approximately equal to

$$B = \frac{1}{2} RK$$

[Verdu, 1998, Ch1, p 7]
4.7 Synchronous CDMA

Synchronous CDMA Model

- Timing is important for orthogonality
- It is not possible to obtain orthogonal codes for asynchronous users.
- Bit epochs are aligned at the receiver
- Require
  - Closed-loop timing control or
  - Providing the transmitters with access to a common clock (such as the Global Positioning System)

[Goldsmith, 2005, Sec. 13.4, p. 425]
[Verdu, 1998, p 21]
Walsh Functions [Walsh, 1923]

- Used in second- (2G) and third-generation (3G) cellular radio systems for providing channelization
- A set of Walsh functions can be ordered according to the number of zero crossing (sign changes)

Walsh Functions of Order $N$: Definition

A set of $N$ functions, denoted, $\{W_j(t); t \in (0,T), j = 0,1, \ldots, N-1\}$, such that

- $W_j(t)$ takes on the values $\{+1,-1\}$
  - Except at the jumps (where it takes the value zero)
- $W_j(0) = 1$ for all $j$.
- $W_j(t)$ has exactly $j$ sign changes (zero crossings) in the interval $(0,T)$.

**Orthogonality:** $\int_0^T W_j(t)W_k(t)\, dt = \begin{cases} 0, & \text{if } j \neq k, \\ T, & \text{if } j = k. \end{cases}$

- Each function $W_j(t)$ is either odd or even with respect to the midpoint of the interval.

Application:

Once we know how to generate these Walsh functions of any order $N$, we can use them in $N$-channel orthogonal multiplexing or multiple access applications.
Walsh Sequences

The Walsh functions, expressed in terms of \{+1,-1\} values, form a group under the multiplication operation (multiplicative group).

The Walsh sequences, expressed in terms of \{0, 1\} values, form a group under modulo-2 addition (additive group).

Closure property:

- \( W_i(t) \cdot W_j(t) = W_{ij}(t) \) for \( i \neq j \)
- \( W_i \oplus W_j = W_{ij} \)

<table>
<thead>
<tr>
<th>Walsh sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>( W_0 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 )</td>
</tr>
<tr>
<td>( W_1 = 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 )</td>
</tr>
<tr>
<td>( W_2 = 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 )</td>
</tr>
<tr>
<td>( W_3 = 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 )</td>
</tr>
<tr>
<td>( W_4 = 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 )</td>
</tr>
<tr>
<td>( W_5 = 0 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 )</td>
</tr>
<tr>
<td>( W_6 = 0 0 1 1 1 1 1 0 0 0 1 1 1 1 0 0 )</td>
</tr>
<tr>
<td>( W_7 = 0 0 1 1 0 0 0 1 1 1 1 1 1 0 0 1 )</td>
</tr>
<tr>
<td>( W_8 = 0 1 1 1 0 0 1 1 0 1 0 0 1 1 0 1 )</td>
</tr>
<tr>
<td>( W_9 = 0 1 1 0 1 1 1 0 0 1 0 1 1 0 1 0 )</td>
</tr>
<tr>
<td>( W_{10} = 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 0 )</td>
</tr>
<tr>
<td>( W_{11} = 0 1 0 1 1 1 0 0 1 0 1 1 0 0 1 1 )</td>
</tr>
<tr>
<td>( W_{12} = 0 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 )</td>
</tr>
<tr>
<td>( W_{13} = 0 1 0 1 1 1 0 0 1 0 1 1 0 0 1 0 )</td>
</tr>
<tr>
<td>( W_{14} = 0 1 0 1 0 1 1 0 1 0 1 1 0 0 1 0 )</td>
</tr>
<tr>
<td>( W_{15} = 0 1 0 1 0 1 1 0 1 0 1 1 0 0 1 0 )</td>
</tr>
</tbody>
</table>

Abstract Algebra

- A **group** is a set of objects \( G \) on which a binary operation “. ” has been defined. "\( \cdot \)": \( G \times G \rightarrow G \) (closure). The operation must also satisfy
  1. Associativity: \( (a \cdot b) \cdot c = a \cdot (b \cdot c) \)
  2. Identity: \( \exists e \in G \) such that \( \forall a \in G \) \( a \cdot e = e \cdot a = a \)
  3. Inverse: \( \forall a \in G \) \( \exists \) a unique element \( a^{-1} \in G \) such that \( a \cdot a^{-1} = a^{-1} \cdot a = e \).

- A group is said to be **commutative** (or abelian) if it also satisfies commutativity:

  \[ \forall a, b \in G \ , \ a \cdot b = b \cdot a. \]

- The group operation for a commutative group is usually represented using the symbol “+”, and the group is sometimes said to be “additive.”
Walsh sequences of order 64

<table>
<thead>
<tr>
<th>Walsh Function Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>We can construct the Walsh functions by:</strong></td>
</tr>
<tr>
<td>1. Using Rademacher functions</td>
</tr>
<tr>
<td>2. Using Hadamard matrices</td>
</tr>
<tr>
<td>3. Exploiting the symmetry properties of Walsh functions themselves</td>
</tr>
</tbody>
</table>

- The **Hadamard matrix** is a square array of “+1” and “−1”, whose rows and columns are mutually orthogonal.

- We can replace “+1” with “0” and “−1” with “1” to express the Hadamard matrix using the logic elements \{0, 1\}.

- The 2×2 Hadamard matrix of order 2 is

\[
H_2 = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} \equiv \begin{bmatrix}
0 & 0 \\
0 & 1
\end{bmatrix}
\]
Hadamard matrix: Properties

Suppose $H_N$ is an $N \times N$ Hadamard matrix.

- $N \geq 1$ is called the order of a Hadamard matrix.
- $N = 1, 2, \text{ or } 4t$ where $t$ is a positive integer.
- $H_N H_N^T = N I_N$
  - $I_N$ is the $N \times N$ identity matrix

Key idea for construction:

If $H_a$ and $H_b$ are Hadamard matrices of order $a$ and $b$, respectively, $H_a \otimes H_b$ is a Hadamard matrix $H_{ab}$ of order $ab$ whose elements are found by substituting $H_b$ for +1 (or logic 0) in $H_a$ and $-H_b$ (or the complement of $H_b$) for -1 (or logic 1) in $H_a$.

Caution: Some textbooks write this symbol as $\times$. It is not the regular matrix multiplication.

Kronecker Product

- An operation on two matrices of arbitrary size
- Named after German mathematician Leopold Kronecker.
- If $A$ is an $m$-by-$n$ matrix and $B$ is a $p$-by-$q$ matrix, then the Kronecker product $A \otimes B$ is the $mp$-by-$nq$ matrix

$$A \otimes B = \begin{bmatrix}
  a_{11}B & \cdots & a_{1n}B \\
  \vdots & \ddots & \vdots \\
  a_{m1}B & \cdots & a_{mn}B
\end{bmatrix}.$$ 

- Example

\[
\begin{bmatrix}
  1 & 2 \\
  3 & 4
\end{bmatrix} \otimes \begin{bmatrix}
  0 & 5 \\
  6 & 7
\end{bmatrix} = \begin{bmatrix}
  1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\
  1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\
  3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\
  3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7
\end{bmatrix} = \begin{bmatrix}
  0 & 5 & 0 & 10 \\
  6 & 7 & 12 & 14 \\
  0 & 15 & 0 & 20 \\
  18 & 21 & 24 & 28
\end{bmatrix}.
\]
Hadamard matrix: Sylvester’s Construction

If \( N \) is a power of two,

start with \( H_1 = [+1] \equiv [0] \),

then \( H_{2^n} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \equiv \begin{bmatrix} H_N & H_N \\ H_N & H_N \end{bmatrix} \).

In MATLAB, use \texttt{hadamard(k)}

Two ways to get \( H_8 \) from \( H_2 \) and \( H_4 \)

\[
H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}
\]

\[
H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix}
\]

\[
H_8 = H_2 \otimes H_4 = \begin{bmatrix} | & | & | & | \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}
\]

\[
H_8 = H_4 \otimes H_2 = \begin{bmatrix} | & | & | & | \\ 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}
\]
Properties

- **Orthogonality**: the rows are orthogonal
  - Geometric interpretation: every two different rows represent two perpendicular vectors
  - Combinatorial interpretation: every two different rows have matching entries in exactly half of their elements and mismatched entries in the remaining elements.

- **Symmetric**
- **Closure property**
- **The elements in the first column and the first row are all 1s. The elements in all the other rows and columns are evenly divided between 1 and -1.**
- **Traceless property**

Walsh–Hadamard (WH) Sequences

- Rows (or columns) of the Hadamard matrix when the order is $N = 2^t$
  - “Same” as Walsh sequences except that
    - they are not indexed according to the number of sign changes.
- **Used in synchronous CDMA**
  - It is possible to synchronize users on the downlink, where all signals originate from the same transmitter.
  - It is more challenging to synchronize users in the uplink, since they are not co-located.
    - Asynchronous CDMA
Hadamard Matrix in MATLAB

- We use the `hadamard` function in MATLAB to generate Hadamard matrix.

```matlab
N = 8; % Length of Walsh (Hadamard) functions
hadamardMatrix = hadamard(N)
```

- The Walsh sequences in the matrix are not arranged in increasing order of their sequencies or number of zero-crossings (i.e. 'sequency order').

Walsh Matrix in MATLAB

- The Walsh matrix, which contains the Walsh functions along the rows or columns in the increasing order of their sequencies, is obtained by changing the index of the `hadamardMatrix` as follows.

```matlab
HadIdx = 0:N-1; % Hadamard index
M = log2(N)+1; % Number of bits to represent the index
binHadIdx = fliplr(dec2bin(HadIdx,M)); % Bit reversing of the binary index
binHadIdx = uint8(binHadIdx)-uint8('0'); % Convert from char to integer array
binSeqIdx = zeros(N,M-1,'uint8'); % Pre-allocate memory
for k = M:-1:2
    % Binary sequency index
    binSeqIdx(:,k) = xor(binHadIdx(:,k),binHadIdx(:,k-1));
end
SeqIdx = bin2dec(int2str(binSeqIdx)); % Binary to integer sequency index
walshMatrix = hadamardMatrix(SeqIdx+1,:) % 1-based indexing
```

- Each column of the sequency index (in binary format) is given by the modulo-2 addition of columns of the bit-reversed Hadamard index (in binary format).
CDMA via Hadamard Matrix

\[ N = 8; \quad \% \ 8 \ \text{Users} \]
\[ H = \text{hadamard}(N); \quad \% \ \text{Hadamard matrix} \]
\[ s = [8 \ 0 \ 12 \ 0 \ 18 \ 0 \ 0 \ 10]; \]
\[ r = s \ast H \]
\[ \% \ r = 8 \ast H(1,:) + 12 \ast H(3,:) + 18 \ast H(5,:) + 10 \ast H(8,:); \]
\[ \% \ \text{Alternatively, use} \]
\[ \% r = \text{ifwht}(s,N,'hadamard') \]
\[ \% \ \text{At Receiver,} \]
\[ s_{\hat{}} = (1/N) \ast r \ast H' \]
\[ \% \ \text{Alternatively, use} \]
\[ \% s_{\hat{}} = \text{fwht}(r,N,'hadamard') \]

Specify the order of the Walsh-Hadamard transform coefficients. ORDERING can be 'sequency', 'hadamard' or 'dyadic'. Default ORDERING type is 'sequency'.

Discrete Walsh-Hadamard transform
4.8 IS-95

Evolution of cellular network

Figure 1.1 Evolution of 2G networks based on TDMA technology.

[Abu-Rgheff, 2007]

Figure 1.2 Evolution of 2G networks based on CDMA technology.
The first CDMA demo

[https://www.youtube.com/watch?v=1X9wWmXzZe]

IS-95 System

- Based on direct sequence CDMA (DS-CDMA)
  - First CDMA-based digital cellular standard.
- The brand name for IS-95 is cdmaOne.
  - Also known as TIA-EIA-95.
  - North America
- Replaced by IS-2000 (CDMA2000)
- 1.25 MHz Channel BW
  - 1.228 Mb/s chip rate
  - WH sequences of order 64 are extensively used in the IS-95 system.
- Remarks
  - IS-95B = cdmaOne
    - Upgrade IS-95A
  - Can carry data at rates up to 14.4 kbps for IS-95A and 115 kbps for IS-95B.
Walsh and WH Sequences of order 64

as indexed in IS-95

[Lee and Miller, 1998, Table 5.8]

WH Sequences in IS-95

- **Forward link (Downlink)**
  - QPSK with a chip rate of 1,228,800 per second.
  - The multiple access scheme is accomplished by the use of 64-bit spreading orthogonal WH sequences (functions).
    - The (coded and interleaved) traffic channel signal symbols are multiplied with distinct repeating WH sequences that are assigned to each channel for the duration of the call.
  - Every base station is synchronized with a GPS receiver so transmissions are tightly controlled in time.

- **Reverse link (Uplink)**
  - The WH sequences are employed as an orthogonal modulation code, which depends only on the data pattern (not channel), forming a 64-ary orthogonal modulation system.
**IS-95 base station transceiver**

![Diagram of IS-95 base station transceiver](image1)

**IS-95 terminal station transceiver**

![Diagram of IS-95 terminal station transceiver](image2)

---

[145] [Fazel & Kaiser, 2008, Fig. 1-13]

[146] [Fazel & Kaiser, 2008, Fig. 1-14]
IS-95

- The reverse link is subject to near-far effects.
- More powerful error correction is employed on the reverse link.
  - A rate 1/2 constraint length 9 convolutional code followed by an interleaver on the forward channel
  - A rate 1/3 constraint length 9 convolutional code followed by an interleaver is used on the reverse link.
    - Also with WH(6,64)
  - Interleaving is utilized to avoid large burst errors, which can be very detrimental to convolutional codes.
- Power control.
  - Use a subchannel on the forward link
  - Every 1.25 ms the base station receiver estimates the signal strength of the mobile unit.
  - If it is too high, the base transmits a 1 on the subchannel. If it is too low, it transmits a 0.
  - In this way, the mobile station adjusts its power every 1.25 ms as necessary so as to reduce interference to other users.

IS-95: Increased Spectral Efficiency

- Improve frequency reuse.
  - Narrow-band systems cannot use the same transmission frequency in adjacent cells because of the potential for interference.
  - CDMA has inherent resistance to interference.
    - Cluster size \( N = 1 \) (theoretically)
    - Although users from adjacent cells will contribute to interference level, their contribution will be significantly less than the interference from the same cell users.
    - Frequency reuse efficiency increases by a factor of 4 to 6.
  - When used to transmit voice signals, CDMA systems may exploit the fact that voice activity typically lies at somewhat less than 40%, thus reducing the amount of interference to 40% of its original value.
References

  - Chapter 4 and 5
  - Chapter 4
UMTS

- **Universal Mobile Telecommunications System (UMTS)**
- The research activity on UMTS started in Europe at the beginning of the 1990s.
  - Even before the earliest 2G systems arrived on the market
- Designed to support wideband services with data rates up to 2Mbit/s.
- Developed from GSM
  - Keep the core network more-or-less intact
  - Change the air interface to use CDMA
- Compatibility between UMTS and GSM:
  - Most UMTS mobiles also implement GSM, and the network can **hand** them **over** from a UMTS base station to a GSM one if they reach the edge of the UMTS coverage area.
  - However, network operators **cannot** implement the two systems in the same frequency band, so they are not fully compatible with each other.
Bandwidth Comparison

<table>
<thead>
<tr>
<th>Generation</th>
<th>Transmission Bandwidth</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1G</td>
<td>25 and 30 kHz</td>
<td></td>
</tr>
<tr>
<td>2G</td>
<td>200 kHz</td>
<td>GSM</td>
</tr>
<tr>
<td></td>
<td>1.25 MHz</td>
<td>IS-95 (CDMA)</td>
</tr>
<tr>
<td>2.5G</td>
<td>1.25 MHz</td>
<td>CDMA2000 1X-RTT</td>
</tr>
<tr>
<td>3G</td>
<td>5 MHz</td>
<td>WCDMA and CDMA2000 3X-RTT</td>
</tr>
<tr>
<td>4G</td>
<td>Up to 20 MHz</td>
<td>LTE</td>
</tr>
<tr>
<td></td>
<td>Up to 60 MHz aggregated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ unlicensed band</td>
<td></td>
</tr>
</tbody>
</table>

Wider and wider radio frequency bands!

[Myung and Goodman, 2008]

UMTS: FDD

- The **chip rate** for spectrum spreading is 3.84 Mc/s.
- The maximum transmitter power of the user equipment is in the range of 21 to 33 dBm (that is, 125 mW to 2 W)

\[
10^{21/10} \text{ mW} \quad 10^{33/10} \text{ mW}
\]

\[
(0 \text{ dBm or dBmW} = 1 \text{ mW})
\]

[Karim and Sarraf, 2002, Fig 6-1]
Review: CDMA

- Two Users. Suppose the code length = \( N \). Taken from \( \mathbf{H}_N \).
- User 1 uses code \( \mathbf{c}^{(1)} \). Want to send messages \( a_1, a_2, a_3, a_4, \ldots \)
  - Send \( \mathbf{x}^{(1)} = \begin{bmatrix} a_1 \mathbf{c}^{(1)} & a_2 \mathbf{c}^{(1)} & a_3 \mathbf{c}^{(1)} & a_4 \mathbf{c}^{(1)} & \ldots \end{bmatrix} \)
- User 2 uses code \( \mathbf{c}^{(2)} \). Want to send messages \( b_1, b_2, b_3, b_4, \ldots \)
  - Send \( \mathbf{x}^{(2)} = \begin{bmatrix} b_1 \mathbf{c}^{(2)} & b_2 \mathbf{c}^{(2)} & b_3 \mathbf{c}^{(2)} & b_4 \mathbf{c}^{(2)} & \ldots \end{bmatrix} \)
- Receiver gets \( \mathbf{r} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} \) MATLAB notation
  - To recover \( a_1 \), calculate \( \frac{1}{N} \langle \mathbf{r}(1: N), \mathbf{c}^{(1)} \rangle \)
  - To recover \( b_1 \), calculate \( \frac{1}{N} \langle \mathbf{r}(1: N), \mathbf{c}^{(2)} \rangle \)
  - To recover \( a_2 \), calculate \( \frac{1}{N} \langle \mathbf{r}((N + 1): (2N)), \mathbf{c}^{(1)} \rangle \)

Observe that, for successful transmission, we need \( \mathbf{c}^{(1)} \perp \mathbf{c}^{(2)} \) or, equivalently, \( \langle \mathbf{c}^{(1)}, \mathbf{c}^{(2)} \rangle = 0 \).

Ex: CDMA

- Two Users. Suppose the code length = 4. Taken from \( \mathbf{H}_4 \).
- User 1 uses code \( \mathbf{c}^{(1)} \). Want to send messages \( a_1, a_2, a_3, a_4, \ldots \)
  - Send \( \mathbf{x}^{(1)} = \begin{bmatrix} a_1 \mathbf{c}^{(1)} & a_1 \mathbf{c}^{(1)} & a_1 \mathbf{c}^{(1)} & a_1 \mathbf{c}^{(1)} & a_2 \mathbf{c}^{(1)} & a_2 \mathbf{c}^{(1)} & a_2 \mathbf{c}^{(1)} & a_2 \mathbf{c}^{(1)} & a_3 \mathbf{c}^{(1)} & a_3 \mathbf{c}^{(1)} & a_3 \mathbf{c}^{(1)} & a_3 \mathbf{c}^{(1)} & a_4 \mathbf{c}^{(1)} & a_4 \mathbf{c}^{(1)} & a_4 \mathbf{c}^{(1)} & a_4 \mathbf{c}^{(1)} \end{bmatrix} \)
- User 2 uses code \( \mathbf{c}^{(2)} \). Want to send messages \( b_1, b_2, b_3, b_4, \ldots \)
  - Send \( \mathbf{x}^{(2)} = \begin{bmatrix} b_1 \mathbf{c}^{(2)} & b_2 \mathbf{c}^{(2)} & b_3 \mathbf{c}^{(2)} & b_4 \mathbf{c}^{(2)} & \ldots \end{bmatrix} \)
- Receiver gets \( \mathbf{r} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} \)
  - To recover \( a_1 \), calculate \( \frac{1}{4} \langle \mathbf{r}(1: 4), \mathbf{c}^{(1)} \rangle \)
  - To recover \( b_1 \), calculate \( \frac{1}{4} \langle \mathbf{r}(1: 4), \mathbf{c}^{(2)} \rangle \)
  - To recover \( a_2 \), calculate \( \frac{1}{4} \langle \mathbf{r}(5: 8), \mathbf{c}^{(1)} \rangle \)

\[ \mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \]
**OVSF (1)**

- Channelization codes used in UMTS W-CDMA and cdma2000 are **variable-length Walsh codes**, also known as **orthogonal variable spreading factor (OVSF)** codes.
- The spreading factors in UMTS may vary from 4 to 256 chips on uplink channels and from 4 to 512 chips on downlink channels.
- In cdma2000, OVSF codes used on traffic channels may vary from 4 to 128 chips.
- **Comparison**: IS-95 uses a set of 64 fixed-length WH codes to spread forward physical channels. In the reverse direction, they are used for orthogonal modulation where every six symbols from the block interleaver output are modulated as one of 64 WH codes.

**OVSF (2): Notation**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$c^{(1,0)} = [1]$</td>
<td>$c^{(2,0)} = [1 \ 1]$</td>
<td>$c^{(4,0)} = [1 \ 1 \ 1 \ 1]$</td>
<td>$c^{(8,0)} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$</td>
<td>$c^{(8,1)} = [1 \ 1 \ 1 \ 1 -1 -1 -1 -1]$</td>
<td>$c^{(8,2)} = [1 \ 1 -1 -1 1 1 -1 -1]$</td>
<td>$c^{(8,3)} = [1 \ 1 -1 -1 -1 -1 1 1]$</td>
<td>$c^{(8,4)} = [1 \ -1 1 -1 1 -1 1 -1]$</td>
</tr>
<tr>
<td>2</td>
<td>$c^{(2,1)} = [1 \ -1]$</td>
<td>$c^{(4,1)} = [1 \ 1 -1 -1]$</td>
<td>$c^{(8,2)} = [1 \ 1 -1 -1 1 1 -1 -1]$</td>
<td>$c^{(8,3)} = [1 \ 1 -1 -1 -1 -1 1 1]$</td>
<td>$c^{(8,4)} = [1 \ -1 1 -1 1 -1 1 -1]$</td>
<td>$c^{(8,5)} = [1 \ -1 1 -1 -1 1 -1 1]$</td>
<td>$c^{(8,6)} = [1 \ -1 -1 1 1 -1 -1 1]$</td>
<td>$c^{(8,7)} = [1 \ -1 -1 1 -1 1 1 -1]$</td>
</tr>
</tbody>
</table>

Extra numbers to indicate the lengths of the codes.

Note that we start with 0.

If the chip rate are the same, then the longer the code (SF) the slower the data rate.
Ex: Multiple Code Lengths

- Two Users. Multiple code lengths: \( \mathbf{c}^{(4,2)} = [1 -1 1 -1] \), \( \mathbf{c}^{(8,7)} = [1 -1 -1 1 -1 1 1 -1] \).
  - User 1 uses code \( \mathbf{c}^{(4,2)} \). Want to send messages \( a_1, a_2, a_3, a_4, \ldots \)
    - Send \( \mathbf{x}^{(1)} = [a_1 \mathbf{c}^{(4,2)}] = \begin{bmatrix} a_1 \mathbf{c}^{(4,2)} & a_2 \mathbf{c}^{(4,2)} & a_3 \mathbf{c}^{(4,2)} & a_4 \mathbf{c}^{(4,2)} & \cdots \end{bmatrix} \).
  - User 2 uses code \( \mathbf{c}^{(8,7)} \). Want to send messages \( b_1, b_2, b_3, b_4, \ldots \)
    - Send \( \mathbf{x}^{(2)} = [b_1 \mathbf{c}^{(8,7)}] = \begin{bmatrix} b_1 \mathbf{c}^{(8,7)} & b_2 \mathbf{c}^{(8,7)} & \cdots \end{bmatrix} \).
  - Receiver gets \( \mathbf{r} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} \).
  - To recover \( a_1 \), calculate \( \frac{1}{4} \langle \mathbf{r}(1:4), \mathbf{c}^{(4,2)} \rangle \).
  - To recover \( a_2 \), calculate \( \frac{1}{4} \langle \mathbf{r}(5:8), \mathbf{c}^{(4,2)} \rangle \).
  - To recover \( b_1 \), calculate \( \frac{1}{8} \langle \mathbf{r}(1:8), \mathbf{c}^{(8,7)} \rangle \).
  - To recover \( b_2 \), calculate \( \frac{1}{8} \langle \mathbf{r}(9:16), \mathbf{c}^{(8,7)} \rangle \).

We still want to make the same CDMA assumption which is that the users (MS) don't have to care about the codes (or the lengths) of other users.

---

Ex: Multiple Code Lengths

- Two Users. Multiple code lengths: \( \mathbf{c}^{(4,2)} = [1 -1 1 -1] \), \( \mathbf{c}^{(8,7)} = [1 -1 -1 1 -1 1 1 -1] \).
  - User 1 uses code \( \mathbf{c}^{(4,2)} \). Want to send messages \( a_1, a_2, a_3, a_4, \ldots \)
    - Send \( \mathbf{x}^{(1)} = [a_1 \mathbf{c}^{(4,2)}] = \begin{bmatrix} a_1 \mathbf{c}^{(4,2)} & a_2 \mathbf{c}^{(4,2)} & a_3 \mathbf{c}^{(4,2)} & a_4 \mathbf{c}^{(4,2)} & \cdots \end{bmatrix} \).
  - User 2 uses code \( \mathbf{c}^{(8,7)} \). Want to send messages \( b_1, b_2, b_3, b_4, \ldots \)
    - Send \( \mathbf{x}^{(2)} = [b_1 \mathbf{c}^{(8,7)}] = \begin{bmatrix} b_1 \mathbf{c}^{(8,7)} & b_2 \mathbf{c}^{(8,7)} & \cdots \end{bmatrix} \).
  - Receiver gets \( \mathbf{r} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} \).
  - To recover \( a_1 \), calculate
    \[
    \frac{1}{4} \langle \mathbf{r}(1:4), \mathbf{c}^{(4,2)} \rangle = \frac{1}{4} \langle \mathbf{x}^{(1)}(1:4) + \mathbf{x}^{(2)}(1:4), \mathbf{c}^{(4,2)} \rangle = \frac{1}{4} \langle a_1 \mathbf{c}^{(4,2)} + b_1 \mathbf{c}^{(8,7)}, \mathbf{c}^{(4,2)} \rangle = a_1 + b_1 \frac{1}{4} \langle \mathbf{c}^{(8,7)}(1:4), \mathbf{c}^{(4,2)} \rangle = 0.
    \]
Ex: Multiple Code Lengths

- Two Users. Multiple code lengths: \( \{ \mathbf{c}^{(4,2)} = [1 -1 1 -1], \mathbf{c}^{(8,7)} = [1 -1 -1 1 -1 1 -1] \} \)
- User 1 uses code \( \mathbf{c}^{(4,2)} \). Want to send messages \( a_1, a_2, a_3, a_4, \ldots \)
  - Send \( \mathbf{x}^{(1)} = \begin{bmatrix} a_1 \mathbf{c}^{(4,2)} & a_2 \mathbf{c}^{(4,2)} & a_3 \mathbf{c}^{(4,2)} & a_4 \mathbf{c}^{(4,2)} \end{bmatrix} \)
- User 2 uses code \( \mathbf{c}^{(8,7)} \). Want to send messages \( b_1, b_2, b_3, b_4, \ldots \)
  - Send \( \mathbf{x}^{(2)} = \begin{bmatrix} b_1 \mathbf{c}^{(8,7)} & b_2 \mathbf{c}^{(8,7)} \end{bmatrix} \)
- Receiver gets \( \mathbf{r} = \mathbf{x}^{(1)} + \mathbf{x}^{(2)} \)
  - To recover \( a_1 \), calculate \( \frac{1}{4} (\mathbf{r}(1:4), \mathbf{c}^{(4,2)}) = a_1 + \frac{b_1}{4} (\mathbf{c}^{(8,7)}(1:4), \mathbf{c}^{(4,2)}) \)
  - To recover \( a_2 \), calculate \( \frac{1}{4} (\mathbf{r}(5:8), \mathbf{c}^{(4,2)}) = a_2 + \frac{b_2}{4} (\mathbf{c}^{(8,7)}(5:8), \mathbf{c}^{(4,2)}) \)
  - To recover \( b_1 \), calculate \( \frac{1}{8} (\mathbf{r}(1:8), \mathbf{c}^{(8,7)}) = \frac{a_1}{8} (\mathbf{c}^{(8,7)}(1:4), \mathbf{c}^{(4,2)}) + \frac{a_2}{8} (\mathbf{c}^{(8,7)}(5:8), \mathbf{c}^{(4,2)}) + b_1 \)
  - To recover \( b_2 \), calculate \( \frac{1}{8} (\mathbf{r}(9:16), \mathbf{c}^{(8,7)}) \)
- Observe that, for successful transmission, we need \( (\mathbf{c}^{(8,7)}(1:4), \mathbf{c}^{(4,2)}) = (\mathbf{c}^{(8,7)}(5:8), \mathbf{c}^{(4,2)}) = 0 \).
- So, we have some idea of how to define orthogonality for codes with different lengths.

OVSF (3)

- Similar to Walsh and WH sequences
  - Arranged and numbered in a different way
- Use a tree structure
- For each spreading factor \( SF = 1, 2, 4, \ldots \), which is a power of 2, there are \( N = SF \) orthogonal codes obtained by the recursion relations:
  \[
  \mathbf{c}^{(2SF,2m)} = \left[ \mathbf{c}(SF,m), \mathbf{c}(SF,m) \right], \quad m = 0,1,2,\ldots, SF - 1
  \]
  \[
  \mathbf{c}^{(2SF,2m+1)} = \left[ \mathbf{c}(SF,m), -\mathbf{c}(SF,m) \right], \quad m = 0,1,2,\ldots, SF - 1
  \]
- Different data rates are supported on a physical channel by simply changing the spreading factor of the associated code.
Code allocation rules

- OVSF codes can be applied to realize connections with different data rates by varying the spreading factor.
  - Smaller SF = Faster data rate
- To have connections with different data rates, need some rules (for selecting the codes) to maintain orthogonality
  - Code blocking property: If a certain code is already used for one connection, neither this code nor a code that is a descendant or an ancestor of this code (on the tree) is allowed to be used for another connection
    - These codes are not orthogonal to the already allocated one.
Code allocation rules: Example

- Suppose code $c^{(4,2)}$ is assigned to a user.

These codes are not allowed to be assigned to other connections.

- If, for example, code $c^{(4,2)}$ is in use, another user (connection) with a different data rate is not allowed to use the encircled codes on the previous slide.
- Other codes can still be used.

- If, for example, the second connection has twice the data rate of the first one, it has to select the code $c^{(2,0)}$.
- Within the period of one data bit of connection 1, connection 2 transmits two data bits.

**Code allocation rules (2)**

- Two OVSF codes are orthogonal if and only if neither code lies on the path from the other code to the root.

- If, for example, code $c^{(4,2)}$ is in use, another user (connection) with a different data rate is not allowed to use the encircled codes on the previous slide.
  - Other codes can still be used.

- If, for example, the second connection has twice the data rate of the first one, it has to select the code $c^{(2,0)}$.
  - Within the period of one data bit of connection 1, connection 2 transmits two data bits.
Code allocation rules: Example

- Suppose code $c^{(4,2)}$ and $c^{(8,7)}$ are currently in use.

\[
\begin{align*}
\mathbf{c}^{(2,0)} &= [1, 1] \\
\mathbf{c}^{(4,0)} &= [1, 1, 1] \\
\mathbf{c}^{(4,1)} &= [1, 1, -1, -1] \\
\mathbf{c}^{(8,0)} &= [1, 1, 1, 1, 1, 1, 1, 1] \\
\mathbf{c}^{(8,1)} &= [1, 1, 1, -1, -1, -1, -1, -1] \\
\mathbf{c}^{(8,2)} &= [1, 1, -1, 1, 1, -1, -1, -1] \\
\mathbf{c}^{(8,3)} &= [1, 1, -1, -1, -1, -1, 1, 1] \\
\mathbf{c}^{(8,4)} &= [1, -1, 1, 1, 1, 1, 1, 1] \\
\mathbf{c}^{(8,5)} &= [1, -1, -1, -1, -1, 1, 1, 1] \\
\mathbf{c}^{(8,6)} &= [1, -1, -1, -1, -1, -1, 1, 1] \\
\mathbf{c}^{(8,7)} &= [1, -1, -1, -1, 1, 1, 1, -1]
\end{align*}
\]
OFDM Applications

- **802.11 Wi-Fi**: a/g/n/ac versions
- **DVB-T** (Digital Video Broadcasting — Terrestrial)
  - terrestrial digital TV broadcast system used in most of the world outside North America
- **DMT** (the standard form of **ADSL** - Asymmetric Digital Subscriber Line)
- **WiMAX, LTE (OFDMA)**

<table>
<thead>
<tr>
<th>Wireless</th>
<th>Wireline</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.11a, g, n (Wi-Fi) Wireless LANs</td>
<td>ADSL and VDSL broadband access via POTS copper wiring</td>
</tr>
<tr>
<td>IEEE 802.15.3a Ultra Wideband (UWB) Wireless PAN</td>
<td>MoCA (Multi-media over Coax Alliance) home networking</td>
</tr>
<tr>
<td>IEEE 802.16d, e (WiMAX), WiBro, and HiperMAN Wireless MANs</td>
<td>PLC (Power Line Communication)</td>
</tr>
<tr>
<td>IEEE 802.20 Mobile Broadband Wireless Access (MBWA)</td>
<td></td>
</tr>
<tr>
<td>DVB (Digital Video Broadcast) terrestrial TV systems: DVB-T, DVB-H, T-DMB, and ISDB-T</td>
<td></td>
</tr>
<tr>
<td>DAB (Digital Audio Broadcast) systems: EUREKA 147, Digital Radio Mondiale, HD Radio, T-DMB, and ISDB-TSB</td>
<td></td>
</tr>
<tr>
<td>Flash-OFDM cellular systems</td>
<td></td>
</tr>
<tr>
<td>3GPP UMTS &amp; 3GPP® LTE (Long-Term Evolution) and 4G</td>
<td></td>
</tr>
</tbody>
</table>
Side Note: Digital TV

**Japan**: Starting July 24, 2011, the analog broadcast has ceased and only digital broadcast is available.

**US**: Since June 12, 2009, full-power television stations nationwide have been broadcasting exclusively in a digital format.

---

**OFDM: Overview**

- Let $\mathbf{S} = (S_1, S_2, \ldots, S_N)$ contains the information symbols.

\[
\begin{align*}
\mathbf{S} \xrightarrow{\text{IFFT}} \mathbf{X} &\xrightarrow{\text{multiplication by a matrix } \mathbf{C} \text{ whose rows are orthogonal}} \mathbf{X} \mathbf{C} \xrightarrow{\text{FFT}} \hat{\mathbf{S}}.
\end{align*}
\]
5.1 Implementation: DFT and FFT

**Review: DS-CDMA**

The BS receives \( \mathbf{r} = \mathbf{sC} \).
The message from the \( k \)th user can be recovered via \( \hat{s}_k = \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(k)} \).
Alternatively, can recover all messages simultaneously: \( \hat{\mathbf{s}} = \frac{1}{N} \mathbf{rC}^T \).

The transmitted signals \( \mathbf{s}_k \mathbf{c}^{(k)} \) are combined in the air
\[ \mathbf{x} = \sum_k \mathbf{s}_k \mathbf{c}^{(k)} = \mathbf{sC}. \]

The \( k \)th user (MS) wants to send \( s_k \).
The \( k \)th user (MS) transmits \( s_k \mathbf{c}^{(k)} \).

Each user (MS) receives \( \mathbf{r} = \mathbf{sC} \).
The \( k \)th user (MS) can recover its message from \( \hat{s}_k = \frac{1}{N} \mathbf{r} \cdot \mathbf{c}^{(k)} \).
OFDM and CDMA

- CDMA’s key equation: \( \bar{s} = \frac{1}{N} (sC)C^T \)
  - All the rows of \( C \) are orthogonal
- Key property of \( C \):
  \[ CC^T = NI. \quad \Rightarrow \quad C^{-1} = \frac{1}{2} C^T \]
- For sync. CDMA, we use the Hadamard matrix \( H_N \).
- For OFDM, we use DFT matrix \( \Psi_N \).
  - The matrix is complex-valued.

Discrete Fourier Transform (DFT)

Here, we work with \( N \)-point signals (finite-length sequences (vectors) of length \( N \)) in both time and frequency domain.

\[
\begin{align*}
\tilde{x} &= \begin{pmatrix}
x[0] \\
x[1] \\
\vdots \\
x[N-1]
\end{pmatrix} \quad \xrightarrow{(N-p+1)\text{DFT}} \quad \tilde{X} = \begin{pmatrix}
X[0] \\
X[1] \\
\vdots \\
X[N-1]
\end{pmatrix} \\
\tilde{X} &= \text{DFT} \{ \tilde{x} \} = \Psi_N \tilde{x}
\end{align*}
\]
DFT matrix $\Psi_N$

$$\Psi_N = \begin{bmatrix}
1 & 1 & 1 & \cdots & 1 \\
1 & \psi_N^{-1} & \psi_N^{-2} & \cdots & \psi_N^{-(N-1)} \\
1 & \psi_N^{-2} & \psi_N^{-4} & \cdots & \psi_N^{-2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \psi_N^{-(N-1)} & \psi_N^{-2(N-1)} & \cdots & \psi_N^{-(N-1)(N-1)}
\end{bmatrix}$$

Element on the $p$th row and $q$th column is given by

$$\psi_N^{-(p-1)(q-1)}$$

where $\psi_N = e^{\frac{2\pi}{N}}$

Note that the “-1” are there because we start from row 1 and column 1 (not from row 0 and column 0).

Key Property:

$$\frac{1}{N} \Psi_N^* \tilde{X} = (\Psi_N^{-1})^{-1} \tilde{X} = \text{IDFT} \{ \tilde{X} \} = \frac{1}{\sqrt{N}} \Psi_N X$$

is a unitary matrix

in time domain

If $A$ is symmetric, suppose $\tilde{Y} = A\tilde{X}$.

Then $\tilde{X} = \tilde{X}^T A^T = \tilde{X} A$

Conclusion: Similar formulas for row vectors.

Example: $N = 2$

- $\psi_2 = e^{\frac{j2\pi}{2}} = e^{j\pi} = -1$

- $\Psi_2 = \begin{bmatrix}
1 & 1 \\
1 & \psi_2^{-1}
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & (-1)
\end{bmatrix} = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} = H_2$

- Suppose $\tilde{X} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$ \rightarrow \text{DFT} \rightarrow $\tilde{X} = \Psi_2 \tilde{X} = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix}$

"Inverse"

$$\tilde{X} = \frac{1}{N} \Psi_N^* \tilde{X} = \frac{1}{2} \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix} \begin{bmatrix} 7 \\ -3 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$
Connection to CDMA

- The rows of $\Psi_N$ are orthogonal. So are the columns.
- Proof: Let $r^{(k)}$ be the $k^{th}$ row of $\Psi_N$.

$$
\left\langle r^{(k_1)}, r^{(k_2)} \right\rangle = \sum_{q=1}^{N} \psi_N^{-(k_1-1)(q-1)}(\psi_N^{-(k_2-1)(q-1)})^* = \sum_{q=1}^{N} \psi_N^{-(k_1-1)(q-1)}\psi_N^{(k_2-1)(q-1)}
$$

$$
= \sum_{q=1}^{N} (\psi_N^{(k_2-k_1)q})^{q-1} = \sum_{q=0}^{N-1} (\psi_N^{(k_2-k_1)})^q
$$

$$
= \frac{1-\psi_N^{(k_2-k_1)N}}{1-\psi_N^{(k_2-k_1)}} = \frac{1-e^{\frac{2\pi}{N}(k_2-k_1)}}{1-\psi_N^{(k_2-k_1)}} = \frac{1-1}{1-\psi_N^{(k_2-k_1)}} = 0, \quad k_1 \neq k_2,
$$

$$
\sum_{q=0}^{N-1} (1)^q = N, \quad k_1 = k_2.
$$

So, $\Psi_N$ “replaces” the role of $H_N$ in CDMA.

Discrete Fourier Transform (DFT)

Matrix form:

$$
\frac{1}{N} \Psi_N^T \bar{X} = \text{IDFT} \{ \bar{X} \} = \bar{x} \quad \xrightarrow{\text{DFT}} \quad \bar{X} = \text{DFT} \{ \bar{x} \} = \Psi_N \bar{x}
$$

Pointwise form:

$$
\frac{1}{N} \sum_{k=0}^{N-1} X[k] \psi_N^{nk} = x[n] \quad \xrightarrow{\text{DFT}} \quad X[k] = \sum_{n=0}^{N-1} x[n] \psi_N^{-nk}
$$

or, equivalently,

$$
\frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{jnk \frac{2\pi}{N}} = x[n] \quad \xrightarrow{\text{DFT}} \quad X[k] = \sum_{n=0}^{N-1} x[n] e^{-jn \frac{2\pi}{N}}
$$

Comparison with Fourier transform

$$
x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad \xrightarrow{\mathcal{F}^{-1}} \quad x(t) = \int_{-\infty}^{\infty} X(t) e^{-j2\pi ft} dt
$$
Efficient Implementation: (I)FFT

An \( N \)-point FFT requires only on the order of \( N \log N \) multiplications, rather than \( N^2 \) as in a straightforward computation.

FFT

- The history of the FFT is complicated.
- As with many discoveries and inventions, it arrived before the (computer) world was ready for it.
- Usually done with \( N \) a power of two.
  - Very efficient in terms of computing time
  - Ideally suited to the binary arithmetic of digital computers.
  - Ex: From the implementation point of view it is better to have, for example, a FFT size of 1024 even if only 600 outputs are used than try to have another length for FFT between 600 and 1024.

OFDM with Memoryless Channel

\[ h(t) = \beta \delta(t) \]  
[should be \( h(t) = \beta \delta(t - \tau) \)]

\[ r(t) = h(t) * s(t) + w(t) = \beta s(t) + w(t) \]

Sample every \( T_s/N \)

\[ r[n] = \beta s[n] + w[n] \]

Additive white Gaussian noise

\[ s[n] = \sqrt{N} \text{IFFT} \{ S \}[n] \]

\[ R_k = \frac{1}{\sqrt{N}} \text{FFT} \{ r \}[n] = \beta S_k + \frac{1}{\sqrt{N}} W_k \]

Sub-channel are independent.  
(No ICI)

OFDM implementation by IFFT/FFT

This form of OFDM is often referred to as **Discrete Multi-Tone (DMT)**.
In a wireless mobile communication system, a transmitted signal propagating through the wireless channel often encounters **multiple reflective paths** until it reaches the receiver.

We refer to this phenomenon as **multipath propagation** and it causes fluctuation of the amplitude and phase of the received signal.

We call this fluctuation **multipath fading**.
Cyclic Prefix: Motivation

- To reduce the ISI, add **guard interval** larger than that of the estimated delay spread.
- If the guard interval is left empty, multipath fading will destroy orthogonality of the sub-carriers, i.e., **ICI** (inter-channel interference) still exists.
- **Solution**: To prevent both the ISI as well as the ICI, OFDM symbol is **cyclically extended** into the guard interval.

Channel with Finite Memory

Discrete time baseband model:

\[ y[n] = \{h * s\}[n] + w[n] = \sum_{m=0}^{\nu} h[m] s[n-m] + w[n] \]

where \( h[n] = 0 \) for \( n < 0 \) and \( n > \nu \)

We will assume that \( \nu \ll N \)

We have \( w[n] \) i.i.d.

\[ w[n] \sim \mathcal{C}\mathcal{N}(0, N_0) \]

Remarks:

- \( Z = X + jY \) is a **complex Gaussian** if \( X \) and \( Y \) are jointly Gaussian.
- If \( X, Y \) is i.i.d. \( \mathcal{N}(0, \sigma^2) \), then \( Z = X + jY \sim \mathcal{CN}(0, \sigma_Z^2) \) where \( \sigma_Z^2 = 2\sigma^2 \) with

\[ f_Z(z) = f_{X,Y}(\text{Re}\{z\}, \text{Im}\{z\}) = \frac{1}{\pi\sigma_Z} e^{-\frac{|z|^2}{\sigma_Z^2}}. \]
Cyclic Prefix

Guard interval, $T_{GI} > \tau_{pr}$
Using empty spaces as guard interval at the beginning of each symbol

End of symbol is prepended to beginning
Guard interval still equals to $T_{GI}$

Recall: Convolution

1. Flip
2. Shift
3. Multiply (pointwise)
4. Add

\[
\{x \ast h\}[n] = \sum_m x[m] h[n - m]
\]
Circular Convolution

Replicate x (now it looks periodic)
Then, perform the usual convolution only from $n = 0$ to $N-1$

Circular Convolution: Examples 1

Find

$\begin{bmatrix} 4, 13, 26, 23, 16 \end{bmatrix} = [1 \ 2 \ 3] * [4 \ 5 \ 6]$

$\gg \text{conv}([1,2,3],[4,5,6])$
$\text{ans} = 4 \ 13 \ 28 \ 27 \ 18$

$\begin{bmatrix} 31, 31, 28 \end{bmatrix} = [1 \ 2 \ 3] \otimes [4 \ 5 \ 6]$

$\gg \text{cconv}([1,2,3],[4,5,6],3)$
$\text{ans} = 31 \ 31 \ 28$

$\begin{bmatrix} 1 \ 2 \ 3 \ 0 \ 0 \end{bmatrix} \otimes [4 \ 5 \ 6 \ 0 \ 0]$

$\gg \text{cconv}([1,2,3,0,0],[4,5,6,0,0],5)$
$\text{ans} = 4.0000 \ 13.0000 \ 28.0000 \ 27.0000 \ 18.0000$
**Discussion**

- *Regular convolution* of an \(N_1\)-point vector and an \(N_2\)-point vector gives \((N_1+N_2-1)\)-point vector.
- *Circular convolution* is performed between two equal-length vectors. The results also has the same length.
- Circular convolution can be used to find regular convolution by **zero-padding**.
  - Zero-pad the vectors so that their length is \(N_1+N_2-1\).
  - Example:
    \[
    \begin{bmatrix}
    1 & 2 & 3 & 0 & 0
    \end{bmatrix} \circledast \begin{bmatrix}
    4 & 5 & 6 & 0 & 0
    \end{bmatrix} = \begin{bmatrix}
    1 & 2 & 3
    \end{bmatrix} \ast \begin{bmatrix}
    4 & 5 & 6
    \end{bmatrix}
    \]

---

**Circular Convolution in Communication**

- We want the receiver to obtain the circular convolution of the signal (channel input) and the channel.
- Q: Why?
- A:
  - **CTFT:** *convolution* in time domain corresponds to *multiplication* in frequency domain.
    - This fact does not hold for DFT.
  - **DFT:** *circular convolution* in (discrete) time domain corresponds to *multiplication* in (discrete) frequency domain.
    - We want to have multiplication in frequency domain.
    - So, we want circular convolution and not the regular convolution.
- Problem: Real channel does regular convolution.
- Solution: With **cyclic prefix**, regular convolution can be used to create circular convolution.
Example

Circular convolution in the time domain is the same as pointwise (element-by-element) multiplication in the frequency domain.

With cyclic prefix, regular convolution can be used to create circular convolution.
Example

\[ s \odot h = [13, 1] \]

\[ s = [2, 5] \]
\[ h = [-1, 3] \]

Cyclic prefix

\[ [5, 2, 5] * h = [-5, 13, 1, 15] \]

Copying only one symbol is enough to get circular convolution inside the result of regular convolution.

Cyclic prefix

\[ [2, 5, 2, 5] * h = [-2, 1, 13, 1, 15] \]

Copying two symbols will also create circular convolution inside the result of regular convolution. However, we are wasting one more symbol space.

26.3 The length of the cyclic prefix is one less than the length of the channel impulse response.

OFDM: One-Shot Transmission

\[ S = [7, -3] \]
\[ \hat{S} = \left[ \frac{14}{2}, \frac{12}{-4} \right] = [7, -3] \]

Pointwise Division by \( H \)

\[ \hat{R} = [14, 12] \]
\[ R = \left[ \frac{13}{2}, \frac{1}{12} \right] \]

\[ y = x * h = \left[ -5\sqrt{2}, 13\sqrt{2}, 1\sqrt{2}, 15\sqrt{2} \right] \]
Example 2

\[ \begin{bmatrix} 1 & -2 & 3 & 1 & 2 \end{bmatrix} \otimes \begin{bmatrix} 3 & 2 & 1 & 0 & 0 \end{bmatrix} = ? \]

Solution:

\[
\begin{array}{ccccc}
1 & -2 & 3 & 1 & 2 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3
\end{array}
\]

\[
\begin{array}{ccccc}
1 & -2 & 3 & 1 & 2 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3
\end{array}
\]

\[
\begin{array}{cccc}
1 & -2 & 3 & 1 & 2 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3
\end{array}
\]

\[
\begin{array}{ccccc}
1 & -2 & 3 & 1 & 2 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3 \\
0 & 0 & 1 & 2 & 3
\end{array}
\]

Let’s look closer at how we carry out the circular convolution operation. Recall that we replicate the \( x \) and then perform the regular convolution (for \( N \) points)

\[
1 \times 1 + 2 \times 2 + 1 \times 3 = 1 + 4 + 3 = 8 \\
2 \times 1 + 1 \times 2 + (-2) \times 3 = 2 + 2 - 6 = -2 \\
1 \times 1 + (-2) \times 2 + 3 \times 3 = 1 - 4 + 9 = 6 \\
(-2) \times 1 + 3 \times 2 + 1 \times 3 = -2 + 6 + 3 = 7 \\
3 \times 1 + 1 \times 2 + 2 \times 3 = 3 + 2 + 6 = 11
\]

Goal: Get these numbers using regular convolution

Example 2

\[ \begin{bmatrix} 1 & -2 & 3 & 1 & 2 \end{bmatrix} \otimes \begin{bmatrix} 3 & 2 & 1 & 0 & 0 \end{bmatrix} = ? \]

Goal: Get these numbers using regular convolution

Observation: We don’t need to replicate the \( x \) indefinitely. Furthermore, when \( h \) is shorter than \( x \), we need only a part of one replica.
Example 2

Copy the last $v$ samples of the symbols to the beginning of the symbol.
This partial replica is called the cyclic prefix.

$$[1 \ 2 \ 1 \ -2 \ 3 \ 1 \ 2] \ast [3 \ 2 \ 1] = ?$$

Try this: use only the necessary part of the replica and then convolute with the channel.
(regular convolution)

We now know that

$$[1 \ 2 \ 1 \ -2 \ 3 \ 1 \ 2] \ast [3 \ 2 \ 1] = [3 \ 8 \ 8 \ -2 \ 6 \ 7 \ 11 \ 5 \ 2]$$

Cyclic Prefix

$$[1 \ -2 \ 3 \ 1 \ 2] \ast [3 \ 2 \ 1 \ 0 \ 0]$$

Similarly, you may check that

$$[-2 \ 1 \ 2 \ 1 \ -3 \ -2 \ 1] \ast [3 \ 2 \ 1] = [-6 \ -1 \ 6 \ 8 \ -5 \ -11 \ -4 \ 0 \ 1]$$

Cyclic Prefix

$$[2 \ 1 \ -3 \ -2 \ 1] \ast [3 \ 2 \ 1 \ 0 \ 0]$$
Example 3

- We know, from Example 2, that
  \[
  \begin{bmatrix}
  1 & 2 & 1 & -2 & 3 & 1 & 2 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  3 & 8 & 8 & -2 & 6 & 7 & 11 & 5 & 2 \\
  \end{bmatrix}
  \]
  And that
  \[
  \begin{bmatrix}
  -2 & 1 & 2 & 1 & -3 & -2 & 1 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  -6 & -1 & 6 & 8 & -5 & -11 & -4 & 0 & 1 \\
  \end{bmatrix}
  \]

- Check that
  \[
  \begin{bmatrix}
  1 & 2 & 1 & -2 & 3 & 1 & 2 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  3 & 8 & 8 & -2 & 6 & 7 & 11 & 5 & 2 \\
  \end{bmatrix}
  \]
  and
  \[
  \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 2 & 1 & -3 & -2 & 1 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & -1 & 6 & 8 & -5 & -11 & -4 & 0 & 1 \\
  \end{bmatrix}
  \]

Example 4

- We know that
  \[
  \begin{bmatrix}
  1 & 2 & 1 & -2 & 3 & 1 & 2 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  3 & 8 & 8 & -2 & 6 & 7 & 11 & 5 & 2 \\
  \end{bmatrix}
  \]
  \[
  \begin{bmatrix}
  -2 & 1 & 2 & 1 & -3 & -2 & 1 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  -6 & -1 & 6 & 8 & -5 & -11 & -4 & 0 & 1 \\
  \end{bmatrix}
  \]

- Using Example 3, we have
  \[
  \begin{bmatrix}
  1 & 2 & 1 & -2 & 3 & 1 & 2 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  3 & 8 & 8 & -2 & 6 & 7 & 11 & 5 & 2 \\
  \end{bmatrix}
  \]
  \[
  +\left(\begin{bmatrix}
  1 & 2 & 1 & -2 & 3 & 1 & 2 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  3 & 8 & 8 & -2 & 6 & 7 & 11 & 5 & 2 \\
  \end{bmatrix}
  \]
  \[
  +\left(\begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 2 & 1 & -3 & -2 & 1 \\
  \end{bmatrix} \begin{bmatrix}
  3 & 2 & 1 \\
  \end{bmatrix} = \begin{bmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 & -6 & -1 & 6 & 8 & -5 & -11 & -4 & 0 & 1 \\
  \end{bmatrix}
  \]
Putting results together...

- Suppose $\mathbf{x}^{(1)} = [1, -2, 3, 1, 2]$ and $\mathbf{x}^{(2)} = [2, 1, -3, -2, 1]$
- Suppose $\mathbf{h} = [3, 2, 1]$
- At the receiver, we want to get
  - $[1, -2, 3, 1, 2] \odot [3, 2, 1, 0, 0] = [8, -2, 6, 7, 11]$
  - $[2, 1, -3, -2, 1] \odot [3, 2, 1, 0, 0] = [6, 8, -5, -11, -4]$
- We transmit $[1, 2, 1, -2, 3, 1, 2, -2, 1, 2, 1, -3, -2, 1]$.

  [Cyclic prefix]

At the receiver, we get

$[1, 2, 1, -2, 3, 1, 2, -2, 1, 2, 1, -3, -2, 1] \odot [3, 2, 1] = [3, 8, 8, -2, 6, 7, 11, -1, 1, 6, 8, -5, -11, -4, 0, 1]$  

Junk! To be thrown away by the receiver.

Circular Convolution: Key Properties

- Consider an $N$-point signal $x[n]$

  **Cyclic Prefix (CP) insertion**: If $x[n]$ is extended by copying the last $v$ samples of the symbols at the beginning of the symbol:

  $\hat{x}[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ x[n+N], & -v \leq n \leq -1 \end{cases}$

- **Key Property 1**: 
  $\{h \otimes x\}[n] = (h \ast \hat{x})[n]$ for $0 \leq n \leq N-1$

- **Key Property 2**:

  $\{h \otimes x\}[n] \xrightarrow{FFT} H_kX_k$
OFDM with CP for Channel w/ Memory

- To send \( N \) samples \( S = (S_0, S_1, \ldots, S_{N-1}) \)
- First apply IFFT with scaling by \( \sqrt{N} \): \( \tilde{s} = \sqrt{N} \text{IFFT}(S) \)
- Then, add cyclic prefix
  \[ x = [\tilde{s}[N-v], \ldots, \tilde{s}[N-1], \tilde{s}[0], \ldots, \tilde{s}[N-1]] \]
- This is inputted to the channel
- The channel output is \( y = x^*h \) which can be viewed as
  \[ y = [p[N-v], \ldots, p[N-1], r[0], \ldots, r[N-1]] \]
- Remove cyclic prefix to get \( r \). (We know that \( r = \tilde{s} \otimes h \).)
- Then apply FFT with scaling by \( 1/\sqrt{N} \): \( \tilde{R} = \frac{1}{\sqrt{N}} \text{FFT}(r) \)
- By circular convolution property of DFT,
  \[ r = \tilde{s} \otimes h \rightarrow R_k = H_k \tilde{S}_k \rightarrow \tilde{R}_k = H_k S_k \rightarrow S_k = \frac{\tilde{R}_k}{H_k} \]

MATLAB Example (1/2)

\[
S = [1 -1 2 4 5 -1 2 -3]; \quad \% \text{data stream} \\
h = [1 0.3 0.1]; \\
\% \text{OFDM transmitter} \\
N = 4; \quad \% \text{Number of data symbols per OFDM symbol} \\
n = \text{length}(S)/N; \quad \% \text{Number of data blocks} \\
St = (\text{reshape}(S,N,n)).'; \quad \% \text{Reshape stream to matrix for} \\
st = (\sqrt{N})*\text{ifft}(St,[\_],2); \quad \% \text{Calculate the IFFT with scaling} \\
\% \text{IFFT w/ scaling} \\
\% \text{(row-wise)} \\
\] 

\[
\begin{align*}
\text{St} &= \\
&= \begin{bmatrix} 1 & -1 & 2 & 4 \\
& 5 & -1 & 2 & -3 \end{bmatrix} \\
\text{st} &= \begin{bmatrix} 3.0 + 0.0i & -0.5 - 2.5i & 0.0 + 0.0i & -0.5 + 2.5i \\
& 1.5 + 0.0i & 1.5 + 1.0i & 5.5 + 0.0i & 1.5 - 1.0i \end{bmatrix} \\
\end{align*}
\]

\[
\begin{align*}
\text{xt} &= \text{reshape}(\text{xt.',}(N+(N+v)*n),1)); \quad \% \text{Reshape back to stream} \\
x &= \begin{bmatrix} 0.0 + 0.0i & -0.5 + 2.5i & 3.0 + 0.0i & -0.5 - 2.5i & 0.0 + 0.0i & -0.5 + 2.5i & 5.5 + 0.0i & 1.5 - 1.0i & 1.5 + 0.0i \\
& 1.5 + 1.0i & 5.5 + 0.0i & 1.5 - 1.0i \end{bmatrix} \\
\end{align*}
\]
MATLAB Example (2/2)

%---------------------------
% Convolve with channel
y = conv(x,h)
H = fft([h zeros(1,N-v-1)])
%---------------------------

OFDM receiver
y = y(1:(N+v)*n));
yt = reshape(y,(N+v),n).';
r = yt(:,v+1:v+N);
Rt = (1/sqrt(N))*fft(r,[],2);

% "Equalization"
S_hatt = zeros(size(Rt));
for i=1:length(H)
    S_hatt(:,i) = Rt(:,i)/H(i);
end
S_hat = reshape(S_hatt.',1,N*n)

OFDM System Design: CP

- A good ratio between the CP interval and symbol duration should be found, so that all multipaths are resolved and not significant amount of energy is lost due to CP.
- As a thumb rule, the CP interval must be two to four times larger than the root mean square (RMS) delay spread.
Summary

- The CP at the beginning of each block has two main functions.
- As guard interval, it prevents contamination of a block by ISI from the previous block.
- It makes the received block appear to be periodic of period N.
  - Turn regular convolution into circular convolution
  - Point-wise multiplication in the frequency domain

OFDM Architecture

[Bahai, 2002, Fig 1.11]
Reference

5.3 OFDM as Multi-Carrier Transmission

Baseband:

\[ s(t) = \sum_{k=0}^{N-1} s_k p(t - kT_s) \]

\[ p(t) = 1_{[0,T_s]}(t) = \begin{cases} 1, & t \in [0,T_s) \\ 0, & \text{otherwise} \end{cases} \]

Passband:

\[ x(t) = \text{Re}\{s(t)e^{j2\pi f_s t}\} \]
Review: Multipath Propagation

- In a wireless mobile communication system, a transmitted signal propagating through the wireless channel often encounters multiple reflective paths until it reaches the receiver.
- We refer to this phenomenon as **multipath propagation** and it causes fluctuation of the amplitude and phase of the received signal.
- We call this fluctuation **multipath fading**.

Wireless Comm. and Multipath Fading

The signal received consists of a number of reflected rays, each characterized by a different amount of attenuation and delay.

\[ y(t) = x(t) * h(t) + n(t) = \sum_{i=0}^{v} \beta_i x(t - \tau_i) + n(t) \]

\[ h(t) = \sum_{i=0}^{v} \beta_i \delta(t - \tau_i) \]

\[ h_1(t) = 0.5\delta(t) + 0.2\delta(t - 0.2T_s) + 0.3\delta(t - 0.3T_s) + 0.1\delta(t - 0.5T_s) \]

\[ h_2(t) = 0.5\delta(t) + 0.2\delta(t - 0.7T_s) + 0.3\delta(t - 1.5T_s) + 0.1\delta(t - 2.3T_s) \]
Frequency Domain

The transmitted signal (envelope)

Channel with weak multipath

Channel with strong multipath

Distortionless Channel

**Definition 3.18.** A channel is called **distortionless** if

\[ y(t) = \beta x(t - \tau) \]

where \( \beta \) and \( \tau \) are constants.

- The channel output has the same “shape” as its input.
- This is the “best” channel we can hope for. Any transmitted signal \( x(t) \) will need to travel over some distance before it reaches the receiver. It will be delayed by the propagation time and its power will be attenuated.

\[ y(t) = \beta x(t - \tau) \]

\[ H(\omega) = e^{-j \omega \tau} \]

From the Fourier transform formula:

\[ H(\omega) = \int_{-\infty}^{\infty} h(t) e^{-j \omega t} \, dt \]

\[ \gamma(t) = \rho x(t) \]

\[ y(t) = \rho e^{-j \omega \tau} x(t) \]

\[ H(\omega) = \gamma(\omega) \]

\[ X(\omega) = X(\omega) \]

\[ \gamma(t) = \rho x(t) \]

\[ y(t) = \rho e^{-j \omega \tau} x(t) \]

\[ H(\omega) = \gamma(\omega) \]

\[ X(\omega) = \rho e^{-j \omega \tau} X(\omega) \]
Distortionless Channel

- A channel is **distortionless** if and only if it satisfies two properties:
  
  (a) "flat" frequency response: constant amplitude response
      \[ |H(f)| = |A| \]
  
  (b) linear phase shift

3.19. Major types of distortion

(a) Linear distortion

(i) **Amplitude distortion** (frequency distortion): \( H(f) \) is not constant with frequency.
\[ |H(f)| \neq |A| \]

(ii) **Phase distortion** (delay distortion): the phase shift is not linear; the various frequency components suffer different amounts of time delay

Observation and a Solution

- **Observation:** Delay spread causes ISI

- A general rule of thumb is that a delay spread of less than 5 or 10 times the symbol width will not be a significant factor for ISI.

- **Solution:** The ISI can be mitigated by reducing the symbol rate and/or including sufficient guard times between symbols.
Multi-Carrier Transmission

- Convert a serial high rate data stream on to **multiple parallel low rate** sub-streams.
- Each sub-stream is modulated on its own **sub-carrier**.
- **Time domain perspective**: Since the symbol rate on each sub-carrier is much less than the initial serial data symbol rate, the effects of delay spread, i.e. ISI, significantly decrease, reducing the complexity of the equalizer.

![Diagram of Multi-Carrier Transmission](image)

\[ T_s = N_c T_d \]

---

Multi-Carrier Modulation

- Old symbol rate: 010011101011
- New symbol rate: 00110101

![Diagram of Multi-Carrier Modulation](image)
Frequency Division Multiplexing

- **Frequency Domain Perspective**: Even though the fast fading is frequency-selective across the entire OFDM signal band, it is effectively flat in the band of each low-speed signal.

  [The flatness assumption is the same one that you used in Riemann approximation of integral.]

Frequency Division Multiplexing (FDM)

- To facilitate separation of the signals at the receiver, the carrier frequencies were *spaced sufficiently far apart* so that the signal spectra did not overlap. Empty spectral regions between the signals assured that they could be separated with readily realizable filters.
- The resulting spectral efficiency was therefore quite low.
Single Carrier vs. Multi-Carrier (FDM)

<table>
<thead>
<tr>
<th>Single Carrier</th>
<th>Multi-Carrier (FDM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single higher rate serial scheme</td>
<td>Parallel scheme. Each of the parallel subchannels can carry a low signaling rate, proportional to its bandwidth.</td>
</tr>
</tbody>
</table>

- **Multipath problem**: Far more susceptible to inter-symbol interference (ISI) due to the short duration of its signal elements and the higher distortion produced by its wider frequency band.
- **Complicated equalization**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Long duration signal elements and narrow bandwidth in sub-channels.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Complexity problem</strong>: If built straightforwardly as several (N) transmitters and receivers, will be more costly to implement.</td>
</tr>
<tr>
<td></td>
<td><strong>BW efficiency problem</strong>: The sum of parallel signalling rates is less than can be carried by a single serial channel of that combined bandwidth because of the unused guard space between the parallel subcarriers.</td>
</tr>
</tbody>
</table>

OFDM

- **OFDM = Orthogonal** frequency division multiplexing
- One of multi-carrier modulation (MCM) techniques
  - Parallel data transmission (of many sequential streams)
  - A broadband is divided into many narrow sub-channels
  - Frequency division multiplexing (FDM)
- High spectral efficiency
  - The sub-channels are made *orthogonal* to each other over the OFDM symbol duration $T_s$.
    - Spacing is carefully selected.
  - Allow the sub-channels to overlap in the frequency domain.
  - Sub-carriers are spaced as close as theoretically possible.
Baseband OFDM Symbol

- Let $\mathbf{S} = (S_1, S_2, \ldots, S_N)$ be the information vector.
- One baseband OFDM modulated symbol can be expressed as

$$s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp\left( j \frac{2\pi kt}{T_s} \right), \quad 0 \leq t \leq T_s$$

$$= \sum_{k=0}^{N-1} S_k \frac{1}{\sqrt{N}} \int_{[0,T_s]} (t) \exp\left( j \frac{2\pi kt}{T_s} \right)$$

Note that:

$$\text{Re}\{s(t)\} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \text{Re}\{S_k\} \cos\left( \frac{2\pi kt}{T_s} \right) - \text{Im}\{S_k\} \sin\left( \frac{2\pi kt}{T_s} \right)$$

| [ECS332 Section 4.6] |

Review: QAM

**Definition 4.76.** In quadrature amplitude modulation (QAM) or quadrature multiplexing, two baseband real-valued signals $m_1(t)$ and $m_2(t)$ are transmitted simultaneously via the corresponding QAM signal:

$$x_{\text{QAM}}(t) = m_1(t) \sqrt{2} \cos(2\pi f_c t) + m_2(t) \sqrt{2} \sin(2\pi f_c t).$$

**Form 1:**

Transmitter (modulator)

$$m_1(t) \rightarrow \sqrt{2} \cos(2\pi f_c t) \rightarrow \frac{\pi}{2} \rightarrow x_{\text{QAM}}(t) \rightarrow h(t) \rightarrow y(t) \rightarrow \pi/2 \rightarrow \sqrt{2} \sin(2\pi f_c t) \rightarrow m_2(t)$$

Receiver (demodulator)

$$v_1(t) \rightarrow H_{v1}(f) \rightarrow \hat{m}_1(t)$$

$$v_2(t) \rightarrow H_{v2}(f) \rightarrow \hat{m}_2(t)$$

Figure 26: QAM Scheme
Review: QAM

4.81. Sinusoidal form (envelope-and-phase description [3, p. 165]):

Form 2: \[ x_{QAM}(t) = \sqrt{2}E(t)\cos(2\pi f_c t + \phi(t)), \]

where

envelope: \[ E(t) = |m_1(t) - jm_2(t)| = \sqrt{m_1^2(t) + m_2^2(t)} \]

phase: \[ \phi(t) = \angle(m_1(t) - jm_2(t)) \]

to calculator

\[ \begin{aligned} 
\text{rectangular} & \quad m_1(t) - jm_2(t) = E(t) \angle \phi(t) \\
\text{polar} & \quad E(t) \angle \phi(t) 
\end{aligned} \]

4.84. Complex form:

Form 3: \[ x_{QAM}(t) = \sqrt{2} \text{Re} \left\{ (m(t)) e^{j2\pi f_c t} \right\} \]

where \[ m(t) = m_1(t) - jm_2(t). \]

OFDM and CDMA: Waveform Version

• Recall: Orthogonality-Based MA (CDMA)

\[ s(t) = \sum_{k=0}^{N-1} S_k c_k(t) \quad \text{where} \quad c_{k_1} \perp c_{k_2} \]

• Baseband OFDM modulated symbol:

\[ s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kt}{T_s} \right), \quad 0 \leq t \leq T_s \]

\[ = \sum_{k=0}^{N-1} S_k \left( \frac{1}{\sqrt{N}} 1_{[0,T_s]}(t) \exp \left( j \frac{2\pi kt}{T_s} \right) \right) c_k(t) \]

Another “special case” of CDMA!
OFDM: Orthogonality

\[
\int c_{k_1}(t)c_{k_2}^*(t)dt = \int_0^{T_s} \exp\left(j \frac{2\pi k_1 t}{T_s}\right) \exp\left(-j \frac{2\pi k_2 t}{T_s}\right) dt
\]
\[
= \int_0^{T_s} \exp\left(j \frac{2\pi (k_1 - k_2) t}{T_s}\right) dt = \begin{cases} T_s, & k_1 = k_2 \\ 0, & k_1 \neq k_2 \end{cases}
\]

When \( k_1 = k_2 \),

\[
\int c_{k_1}(t)c_{k_2}^*(t)dt = \int_0^{T_s} 1 dt = T_s
\]

When \( k_1 \neq k_2 \),

\[
\int c_{k_1}(t)c_{k_2}^*(t)dt = \frac{T_s}{j2\pi (k_1 - k_2)} \left[ \exp\left(\frac{j 2\pi (k_1 - k_2) t}{T_s}\right) \right]_0^{T_s}
\]
\[
= \frac{T_s}{j2\pi (k_1 - k_2)} (1-1) = 0
\]

Frequency Spectrum

\[
s(t) = \sum_{k=0}^{N-1} S_k \frac{1}{\sqrt{N}} 1_{[0,T_s]}(t) \exp\left(j \frac{2\pi k t}{T_s}\right)
\]
\[
c(t) = \frac{1}{\sqrt{N}} 1_{[0,T_s]}(t) \xrightarrow{\mathcal{F}} C(f) = \frac{1}{\sqrt{N}} T_s e^{-j2\pi f \frac{T_s}{2}} \text{sinc}(\pi T_s f)
\]
\[
c_k(t) = c(t) \exp\left(j \frac{2\pi k t}{T_s}\right) \xrightarrow{\mathcal{F}} C_k(f) = C\left(f - \frac{k}{T_s}\right) = C\left(f - k\Delta f\right)
\]
\[
s(t) = \sum_{k=0}^{N-1} S_k c_k(t) \xrightarrow{\mathcal{F}} S(f) = \sum_{k=0}^{N-1} S_k C_k(f)
\]
\[
= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k e^{-j2\pi (f-k\Delta f) \frac{T_s}{2}} T_s \text{sinc}(\pi T_s (f - k\Delta f))
\]

This is the term that makes the technique FDM.
Subcarrier Spacing

\[ \Delta f = \frac{1}{T_s} \]

Each QAM signal carries one of the original input complex numbers.

N separate QAM signals, at N frequencies separated by the signaling rate.

The spectrum of each QAM signal is of the form with nulls at the center of the other sub-carriers.

Normalized Power Density Spectrum

More flat with more sub-carriers

\[ s(t) = \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} |\text{[0,x]}(t)| \exp \left( \frac{2\pi fk}{T_s} \right) \]

\[ S(f) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k e^{-j2\pi k/fM} T_s \frac{\sin(\pi f/A)}{\pi f} \]

[Fazel and Kaiser, 2008, Fig 1-5]
OFDM Carriers: $N = 4$

Recall: Multi-Carrier Modulation
**Time-Domain Signal**

Real and Imaginary components of an OFDM symbol is the superposition of several harmonics modulated by data symbols.

\[
s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kt}{T_s} \right), \quad 0 \leq t \leq T_s
\]

\[
\text{Re}\{s(t)\} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \text{Re}\{S_k\} \cos \left( \frac{2\pi kt}{T_s} \right) - \text{Im}\{S_k\} \sin \left( \frac{2\pi kt}{T_s} \right)
\]

\text{in-phase part} \quad \text{quadrature part}

---

**Summary**

- So, we have a scheme which achieves
  - Large symbol duration \(T_s\) and hence less multipath problem
  - Good spectral efficiency

- One more problem:
  - There are so many carriers!
Discrete Fourier Transform (DFT)

Transmitter produces

\[ s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi k}{T_s} t \right), \quad 0 \leq t < T_s \]

Sample the signal in time domain every \( T_s/N \) gives

\[
s[n] = s \left( n \frac{T_s}{N} \right) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kn}{N} \right) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi k}{N} \right) = \sqrt{N} \text{IDFT}\{S\}[n]
\]

where

\[ \text{IDFT}\{S\}[n] = \frac{1}{N} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kn}{N} \right) \]

\[ \tilde{S} = (S_0, S_1, \ldots, S_{N-1})^T \]

We can implement OFDM in the discrete domain!

DFT Samples

\[ s(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kt}{T_s} \right), \quad 0 \leq t \leq T_s \]

\[
s[n] = s \left( n \frac{T_s}{N} \right) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} S_k \exp \left( j \frac{2\pi kn}{N} \right) = \sqrt{N} \text{IDFT}\{S\}[n] \]

\[ 0 \leq n < N \]
Oversampling

Increase the number of sample points from \( N \) to \( LN \) on the interval \([0, T_s]\).

\( L \) is called the **over-sampling factor**.

\[
s[n] = s\left(n \frac{T_s}{N}\right)
\]

\( 0 \leq n < N \)

\[
s^{(L)}[n] = s\left(n \frac{T_s}{LN}\right)
\]

\( 0 \leq n < LN \)

\[
s^{(L)}[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{S}_k \exp\left(j \frac{2\pi kn}{LN}\right)
\]

\[
= \frac{1}{\sqrt{N}} LN \left( \frac{1}{LN} \sum_{k=0}^{N-1} \tilde{S}_k \exp\left(j \frac{2\pi kn}{LN}\right) + \sum_{k=N}^{LN-1} 0 \exp\left(j \frac{2\pi kn}{LN}\right) \right)
\]

\[
= L \sqrt{N} \left( \frac{1}{LN} \sum_{k=0}^{N-1} \tilde{S}_k \exp\left(j \frac{2\pi kn}{LN}\right) \right) = L \sqrt{N} \text{IDFT}\{\tilde{S}\}[n]
\]

\[
\tilde{S}_k = \begin{cases} 
S_k, & 0 \leq k < N \\
0, & N \leq k < LN
\end{cases}
\]
Oversampling: Summary

\[ s[n] = s\left(n \frac{T_s}{N}\right) = \sqrt{N} \text{IDFT}\{S\}[n] \quad 0 \leq n < N \]

\[ s^{(L)}[n] = s\left(n \frac{T_s}{LN}\right) = L\sqrt{N} \text{IDFT}\{\tilde{S}\}[n] \quad 0 \leq n < LN \]

Zero padding:

\[ \tilde{S}_k = \begin{cases} S_k, & 0 \leq k < N \\ 0, & N \leq k < LN \end{cases} \]

Summary: Three steps towards modern OFDM

1. To mitigate multipath problem
   \( \Rightarrow \) Use multicarrier modulation (FDM)

2. To gain spectral efficiency
   \( \Rightarrow \) Use orthogonality of the carriers

3. To achieve efficient implementation
   \( \Rightarrow \) Use FFT and IFFT
5.4 OFDM in LTE

Dr. Prapun Suksompong
prapun.com/ecs455

---

## Advanced Mobile Wireless Systems

(IEEE) (Ultra Mobile Broadband)

<table>
<thead>
<tr>
<th></th>
<th>Mobile WiMAX</th>
<th>3GPP LTE</th>
<th>3GPP2 UMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel bandwidth</td>
<td>5, 7, 8.75, and 10 MHz</td>
<td>1.4, 3, 5, 10, 15, and 20 MHz</td>
<td>1.25, 2.5, 5, 10, and 20 MHz</td>
</tr>
<tr>
<td>DL multiplex</td>
<td>OFDM</td>
<td>OFDM</td>
<td>OFDM</td>
</tr>
<tr>
<td>UL multiple access</td>
<td>OFDMA</td>
<td>SC-FDMA</td>
<td>OFDMA and CDMA</td>
</tr>
<tr>
<td>Duplexing</td>
<td>TDD</td>
<td>FDD and TDD</td>
<td>FDD and TDD</td>
</tr>
<tr>
<td>Subcarrier mapping</td>
<td>Localized and distributed</td>
<td>Localized</td>
<td>Localized and distributed</td>
</tr>
<tr>
<td>Subcarrier hopping</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data modulation</td>
<td>QPSK, 16-QAM, and 64-QAM</td>
<td>QPSK, 16-QAM, and 64-QAM</td>
<td>QPSK, 8-PSK, 16-QAM, and 64-QAM</td>
</tr>
<tr>
<td>Subcarrier spacing</td>
<td>10.94 kHz</td>
<td>15 kHz</td>
<td>9.6 kHz</td>
</tr>
<tr>
<td>FFT size (5 MHz bandwidth)</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
</tbody>
</table>

[Myung and Goodman, 2008]
LTE: Multiple Access

- Downlink: **OFDMA** (or we can simply say OFDM)

- Uplink: **SC-FDMA**

**LTE: OFDMA**

- 15 kHz subcarrier spacing

**Downlink Resource Assignment in Time and Frequency**

[Holma and Toskala, 2009, Fig 4.4]

[Rysavy, 2007, Fig 37]