SDR Architectures

SDR signal processing generally follows this structure:

Throughput here means, “How fast can the next value be entered”

See [D38]
SDR Architectures

Data flow between SDR blocks:

Control flows from real time (antenna) to not necessarily real time (data sink) processing

Control flows from real time (antenna) to a data source which could have been precompiled in non-real time
Multichannel SDR, simple obvious receiver architecture

RF front end loss = lower receiver sensitivity!
SDR Architectures

DSSS and FEC packet processing using alternate FPGA DR frames:
# GNU Radio, Labview and Simulink

## Overview and comparison of three popular block diagram oriented tools for building SDRs

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNU Radio</td>
<td>Free and open source, accessible to hobby users.</td>
<td>Runs under Linux, only limited support for Windows. Data flow between blocks can make recursive flow graphs a problem. Does not seem to support VHDL code generation, application is limited to running on PC.</td>
</tr>
<tr>
<td>Labview</td>
<td>Runs under both Linux and Windows. Can be event driven or sample driven</td>
<td>Somewhat expensive (about $5000 to get started).</td>
</tr>
<tr>
<td></td>
<td>Makes excellent GUI controls</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Can convert Labview block diagrams to VHDL</td>
<td></td>
</tr>
<tr>
<td>Simulink</td>
<td>Runs under both Linux and Windows. Not event driven. Runs sample by sample or fame by frame, timing is tightly controlled so that recursions around blocks are predictable. Block diagrams directly translate to VHDL, facilities product design</td>
<td>Full featured version is very expensive, requires multiple toolboxes with multiple licenses.</td>
</tr>
</tbody>
</table>

“Software Defined” here means a set of *blocks* can be arranged in a block diagram editor (BDE) any number of ways to implement any radio within the capabilities of the RF front end hardware. Notice the difference with SCA, which depends on an arrangement of *files*. Both approaches require a skilled developer to expand the radio repertoire.
GNU Radio Companion signal flow graph details

- **Hardware Source**: Radio420 GigE: Source Deinterleaved
- **Port (complex data)**: GMSK Demod
- **Port (real data)**: Packet Decoder
- **Hardware Sink**: Audio Sink

**Variable**
- **ID**: data_rate
- **Value**: 4.8M

**Parameter setup for radio that drives port**

**Signal processing block, usually written in C++**

**Parameter, seen by all blocks**

**Software (GUI) Sink**

**Parameter setup for radio that drives port**

**Hardware Source**

**Port (complex data)**

**Port (real data)**

**Hardware Sink**
GNU Radio, Labview and Simulink

Labview Basic Characteristics:

- A programming language that is represented by a block diagram editor.
- Started out as a system for laboratory instrument control and data processing.
- Now has advanced blocks that can implement digital communications systems.
- Labview blocks can run event driven or sample by sample.
Simulink VHDL code generation example

**Key concept:** Simulink BDE, VHDL, and FPGA logic are “Bit and cycle true”

Notice the very precise fixed point attribute (FPA) “sfix24_en12”, a twos complement fixed point signed number with 24 total bits and 12 fractional bits. FPA can be different for every signal to minimize use of FPGA resources. This is a departure from the words and bytes of processor based computing.
A goal of SCA is, from [S2a]:

“Provide for the portability of applications between different SCA compliant implementations.”

SCA seeks to save time and money by enabling deployment of SCA compliant waveform applications on radio hardware manufactured by various companies.

Any qualified company can produce signal processing software for an SCA radio – software becomes a commodity and price to the government falls
Software Communications Architecture (SCA)

The SCA “Big picture”, the framework that sets up SDR applications

Core Framework (CF): Abstracts the underlying hardware

Components (HW & SW)

Assembly Controller (AC): Starts, stops, configures a waveform

Device Manager: keeps track of radio hardware implementations

Domain Profile:
Central repository for information about this radio and its current waveform configurations

File System

© John Reyland, PhD
Is SCA suitable for space flight radio reconfiguration? Space communication can have different constraints:

• Space flight radios tend to use slightly older more “time proven” technology – thus computational resources and memory may be limited.

• Deep space missions can extend over many years - the ability to reconfigure a waveform after deployment can have significant advantages for data collection.
A compelling case for space SDR: The Huygens symbol timing problem [S17]

On Christmas day, 2004, the Cassini spacecraft sent the Huygens space probe coasting down towards the surface of Saturn’s largest moon, Titan.

Several communications challenges caused the receiver symbol timing to fail:

1. Doppler time dilation due to Cassini - Huygens relative velocities.
2. The effect of receiver AGC on symbol timing loop performance
3. Higher than normal link data rates (8 Kbits/sec instead of previously used 2 Kbits/sec)

Unfortunately, ground test coverage was not through enough to pick up the symbol timing problem. The solution was to reconfigure the spacecraft trajectory to reduce Doppler. Would an SDR have provided a more expedient solution?
Space Telecommunications Radio System (STRS)

NASA’s Space Communications and Navigation (SCAN) Test Bed

Launched on July 21, 2012
Tanegashima Space Center of Japan on the launch vehicle H-IIB

http://spaceflightsystems.grc.nasa.gov/SOPO/SCO/SCaNTestbed/
SDR Advantages

SDR facilitates Adaptive Coding and Modulation (ACM):

- **BPSK**
  - $P_{\text{MAX}}$
  - Adequate Received Signal Power
  - $P_{\text{REC}}$
  - EbN0 9.6dB
  - $P_{\text{SENS}}$
  - Inadequate Received Signal Power

- **8PSK**
  - $P_{\text{MAX}}$
  - Adequate Received Signal Power
  - $P_{\text{REC}}$
  - EbN0 >9.6dB
  - $P_{\text{SENS}}$
  - Inadequate Received Signal Power

12/8/2015  
John Reyland, PhD
**SDR Advantages**

**Radio Evolution**

- **1920**
  - **Hardware Defined Radio**
  - No processor at all

- **1980**
  - **Software Designed Radio**
  - Specific defined uses only
  - Generally not upgradable
  - Cannot be future proof
  - Pager, cell phone

- **1990**
  - **Parameter Controlled Radio**
  - Re-configurability based on proprietary design
  - Wide range of parameter settings, tuning, BW, modulation
  - Possibly useful for signals not envisioned by manufacturer
  - Maybe software upgradable
  - Flex 5000 Amateur Radio

- **2000**
  - **Software Defined Radio**
  - Re-configurability based on open source standard
  - Re-configurability only limited by available hardware
  - Future-proof, useful for signals not envisioned by manufacturer
  - Must be software upgradable
  - SCA, STRS

- **2010**
  - **Cognitive Radio**
  - Can re-configure itself based on sensing operating environment
  - Built in intelligence to adapt to quality of service requirements
  - Can take advantage of unused spectrum.
  - IEEE 802.22
SDR Advantages

- Cognitive Radio does not necessarily depend on SDR.
- CR can be implemented with or without the ultimate software defined radio (e.g SCA).
- CR and SDR can have separate technology roadmaps.

From [H5]
Cognitive radio example:

GDD-dependent station sends a channel availability query (CAQ) to a GDD-enabling station (TV station).

GDD-enabling station responds by transmitting a white space map (WSM) that has available radio frequencies and times.

CAQ must be retransmitted after an enabling time out.
SDR Advantages

Optimum signal detection for Wide Sense Stationary (WSS) signals

Optimum signal detection for Cyclostationary (CSS) signals
Orthogonal Frequency Division Multiplexing (OFDM)

Consider two sine waves with

Frequencies of 3 Hz and 5 Hz.

Both are multiples of 1 Hz

Time record = T = 1 second

Sine waves with frequencies that are an integer multiple of a fundamental frequency are orthogonal

\[
0 = \int_{0}^{T} \cos(2\pi 3t) \cos(2\pi 5t) \, dt
\]
Orthogonal Frequency Division Multiplexing (OFDM)

Consider QPSK modulation as a series of complex numbers in the time domain:

\[ s(n) \in \{ s_0, s_1, s_2, s_3 \} \]

Each complex number is called a symbol

Symbol rate = \( F_s = \frac{1}{T_s} \)

Each symbol represents 2 transmit data bits

Bit rate = \( F_B = \frac{1}{T_B} = 2F_s = \frac{2}{T_s} \)

Index \( n \) increments every new symbol time:

\[
\begin{align*}
  n &= 0 & 1 & 2 & 3 & \ldots \\
  0 & T_s & 2T_s & 3T_s & \ldots
\end{align*}
\]

OFDM does not use RC filtering!
SDR Modulation Types

OFDM starts by converting high speed symbols indexed by \( n \) at rate \( 1/T_s \) into parallel blocks indexed by \( k \) at rate \( 1/T = 1/MT_s \).

In this example, \( M=4 \)

Symbols enter at rate \( F_s = 1/T_s \)
Symbols go into frequency domain bins.
Symbols on separate frequency domain channels now exit at \( F_s \). Each block is an OFDM “symbol” at block rate \( F_s/M \)

FDM with channel spacing at the symbol rate
SDR Modulation Types

Here is the output spectrum showing orthogonal channels spaced $1/MT_s$ apart. Note that sampling is at the original symbol rate: $F_s = 1/T_s$

Important: Four time domain symbols in results in four samples out
The four samples out are four samples/symbol on 4 different orthogonal channels

$s(4k) = b(4k)e^{j0} + b(4k+1)e^{j0} + b(4k+2)e^{j0} + b(4k+3)e^{j0}$
$s(4k + 1) = b(4k)e^{j0} + b(4k + 1)e^{jπ/2} + b(4k + 2)e^{jπ} + b(4k + 3)e^{-jπ/2}$
$s(4k + 2) = b(4k)e^{j0} + b(4k + 1)e^{jπ} + b(4k + 2)e^{j2π} + b(4k + 3)e^{-jπ}$
$s(4k + 3) = b(4k)e^{j0} + b(4k + 1)e^{-jπ/2} + b(4k + 2)e^{-jπ} + b(4k + 3)e^{jπ/2}$
SDR Modulation Types

\[ s(4k) = b(4k)e^{j0} + b(4k+1)e^{j0} + b(4k+2)e^{j0} + b(4k+3)e^{j0} \]
\[ s(4k+1) = b(4k)e^{j0} + b(4k+1)e^{j\pi/2} + b(4k+2)e^{j\pi} + b(4k+3)e^{-j\pi/2} \]
\[ s(4k+2) = b(4k)e^{j0} + b(4k+1)e^{j\pi} + b(4k+2)e^{j2\pi} + b(4k+3)e^{-j\pi} \]
\[ s(4k+3) = b(4k)e^{j0} + b(4k+1)e^{-j\pi/2} + b(4k+2)e^{-j\pi} + b(4k+3)e^{j\pi/2} \]

\( M=4 \) example has \( m=0,1,2,3 \). With addition of \( 1/M \) scaling, we can write:

\[ s(Mk + m) = \frac{1}{M} \sum_{n=0}^{M-1} b(Mk + n)e^{j2\pi \left( \frac{mn}{M} \right)} \]
\[ m = 0,1,\ldots,M-1 \]

This is the definition of the Inverse Discrete Fourier Transform (IDFT)!
More practical example:

OFDM with $M=12$ BPSK Channels

Input symbol rate: $F_s$

Block Rate = $F_s/M = F_s/12$

Output Sample Rate:

$= 16(F_s/12) = (4/3)F_s$

Frequency bin spacing:

$(1/16)(4/3)F_s = F_s/12$
SDR Modulation Types

Here is the output spectrum showing orthogonal channels spaced $1/12T_s$ apart

Note that sampling is faster than the original symbol rate: $f_{\text{sample}} = 4/3T_s$
Comparison of MSK, SOQPSK and GMSK Power Spectrums

Power Spectrum, 14K Bits/sec

Normalized Frequency fT, shows 25KHz channel

Average Power

-2 -1.5 -1 -0.5 0 0.5 1 1.5 2

-60 -50 -40 -30 -20 -10 0

MSK
SOQPSK
GMSK

© John Reyland, PhD
“Moving charge and time varying electric fields produce magnetic fields”

Andre Ampere
1775 - 1836

“Time varying magnetic fields produce electric fields”

Michael Faraday
1791 - 1867
RF Propagation Channels

\[ f = \frac{c}{\lambda} = \frac{\text{cycles}}{\text{second}} \]

\( c = 3 \times 10^8 \text{ meters/second} \)
\( f = \text{wave frequency} \)

\[ S = E \times H \]

S is called the Poynting vector

S must be perpendicular to both E and H vectors. According to the “right hand rule”, S always points forward
RF Propagation Channels

Line of sight link depends on Fresnel zone radius

\[ R = \sqrt{\frac{\lambda A (D - A)}{D}} \]

Example: \( D = 2000 \text{ m}, A = 200 \text{ m}, \)
\( H_{\text{Tx}} = H_{\text{Rx}} = 20 \text{ m}, \quad \lambda = 0.0556 \text{ m} \)

This means \( R = 3.16 \text{ m}. \) When tree is > 16.84 m high it must be trimmed to avoid knife edge diffraction
RF Propagation Channels

RF signal interaction with a fixed propagation environment:

- **Reflection**, waves hit an object whose dimensions are large compared to the wavelength.

- **Refraction**, waves bend down slightly towards the surface of the earth. A primary example of refraction is bending of waves as they approach the ionosphere.

- **Diffraction**, waves are partially obstructed by an object.

- **Absorption**, waves are attenuated by passing through solid material.

- **Scattering**, waves travel through a medium consisting of a high density of objects that are similar size compared to the wavelength.

These are also called large scale path loss mechanisms.
RF Propagation Channels

Diffraction, reflection, scattering can all contribute to small scale multipath fading

Multipath: Multiple paths from transmitter to receiver. Small receiver or reflector movements can cause variations in received signal power
RF Propagation Channels

Doppler spectrum is brought about by multiple reflectors

Constant velocity, constant wavelength: multiple paths result in multiple Doppler shifts:

\[ f_d = \frac{1}{2\pi} \Delta \phi = \frac{v \cos(\theta)}{\lambda} \]

Note: even without multipaths, Doppler will still change due to changing angle
# RF Propagation Channels

## Simplified summary of small scale fading conditions

<table>
<thead>
<tr>
<th>Multipath Time Dispersion (Frequency selective fading due to fixed or moving receiver)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flat Fading</strong></td>
<td><strong>Frequency Selective Fading</strong></td>
</tr>
<tr>
<td>Signal BW &lt; Channel Coherence BW</td>
<td>Signal BW &gt; Channel Coherence BW</td>
</tr>
<tr>
<td>Delay Spread &lt; Symbol Period</td>
<td>Delay Spread &gt; Symbol Period</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multipath Frequency Dispersion (Time selective fading due to movement)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slow Fading</strong></td>
<td><strong>Fast Fading</strong></td>
</tr>
<tr>
<td>Max Doppler Spread &lt; Signal BW</td>
<td>Max Doppler Spread &gt; Signal BW</td>
</tr>
<tr>
<td>Coherence Time &gt; Symbol Period</td>
<td>Coherence Time &lt; Symbol Period</td>
</tr>
<tr>
<td>Many symbols between fading nulls</td>
<td>Many fading nulls between symbols</td>
</tr>
</tbody>
</table>

Note: these two propagation mechanism are independent of each other!
Wireless channel are so full of hazards!

What can be done??

We will look at:

1. Fade Margin
2. Diversity
3. Rake Receiver
Time Diversity:

After receiver reassembles, all errors can be corrected. See [R5] and [R6].
Space-time coding [R12]:

Two base station antennas transmit the same symbols but with different coding. Receiver has only one antenna. Performance is equivalent to two receive antenna diversity combining using maximum ratio combining. This is using in LTE [R10] and wireless LANs [R11].

In this example, transmit signal is: $s(t) \in [s_0, s_1]$

<table>
<thead>
<tr>
<th>Symbol Time:</th>
<th>Symbols ready to transmit</th>
<th>Transmit antenna 0</th>
<th>Transmit antenna 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s(0), s(1)</td>
<td>s(0)</td>
<td>s(1)</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>-s(1)*</td>
<td>s(0)*</td>
</tr>
<tr>
<td>2</td>
<td>s(2), s(3)</td>
<td>s(2)</td>
<td>s(3)</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>-s(3)*</td>
<td>s(2)*</td>
</tr>
<tr>
<td>...</td>
<td>....</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
RF Propagation Channels

Example:

\[ f_c = 1 \text{ GHz} = 1 \times 10^9 \text{ Hz} \]

\[ v(t) = v = 350 \text{ meters/second (constant, approx. Mach 1)} \]

\[ \theta(t) = 0 \text{ (constant, worst case for Doppler shift. This is a simplifying assumption)} \]

\[ v_r = v = \text{Velocity of receiver relative to transmitter} \]

\[ f_d(t) = f_d = f_c \left( \frac{v}{c} \right) = \left(1 \times 10^9\right) \frac{350}{3 \times 10^8} = \frac{10(350)}{3} = 1167 \text{ Hz} = \text{Doppler carrier frequency shift at receiver} \]

\[ T_t(t) = T_t = \frac{1}{1e + 6} = 1e - 6 = \text{Transmit symbol time} \]

\[ T_r(t) = T_r = T_t + \frac{v T_t}{c} = (1e - 6) \left(1 + \frac{350}{3 \times 10^8}\right) = (1e - 6)(1.000001167) = \text{Receive symbol time} \]

This means receive symbol time increases by 0.0001167%. - called time dilation
Some basic transmit or receive antenna characteristics

\[ \lambda = \text{carrier wavelength} = \frac{c}{f} = \frac{(3 \times 10^8 \text{ meters/second})}{(\text{cycles/second})} \]

\[ \theta = \frac{72 \lambda}{D} \text{ degrees} = \text{half-power beamwidth angle} \]

\[ G = \frac{4\pi A_e}{\lambda^2} = \text{antenna gain as a function of antenna aperture and wavelength} \]

\[ A_e = \eta A_p = \eta \left( \pi \left( \frac{D}{2} \right)^2 \right) \quad \eta \approx 0.6 \quad \text{for parabolic dish} \]
RF Propagation Channels

Link Margin: Will the radio system work or not?
# RF Propagation Channels

## Link Budget: Considering the entire radio system

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx power output</td>
<td>+30 dBm</td>
<td>Rx signal power</td>
<td>-85 dBm</td>
</tr>
<tr>
<td>Tx cable loss</td>
<td>-2 dB</td>
<td>Rx system noise temp</td>
<td>4000 Kelvin</td>
</tr>
<tr>
<td>Tx antenna gain</td>
<td>+40 dB</td>
<td>Boltzmann Constant</td>
<td>-198.6 dBm/(K*Hz)</td>
</tr>
<tr>
<td>Pointing Loss</td>
<td>- 1 dB</td>
<td>Noise Spectral Density</td>
<td>-198.6+10*log_{10}(4000) = -161 dBm/Hz</td>
</tr>
<tr>
<td>Tx EIRP</td>
<td>+30-2+40-1 = 67 dBm</td>
<td>Data rate</td>
<td>2 Mbit/sec</td>
</tr>
<tr>
<td>Propagation Loss</td>
<td>-180 dB</td>
<td>10*log10(data rate)</td>
<td>63 dB(bits/sec)</td>
</tr>
<tr>
<td>Fade allowance</td>
<td>-4 dB</td>
<td>Eb = Energy/bit</td>
<td>-85 -63 = -148 dBm/Hz</td>
</tr>
<tr>
<td>Rx isotropic power</td>
<td>67-180-4 = -117 dBm</td>
<td>(rec pwr)(bit time)</td>
<td>(rec pwr)/(bit rate)</td>
</tr>
<tr>
<td>Rx antenna gain</td>
<td>+35 dB</td>
<td>Received Eb/N0</td>
<td>-148 –(-161) = 13dB</td>
</tr>
<tr>
<td>Pointing Loss</td>
<td>- 1 dB</td>
<td>Required Eb/N0</td>
<td>10 dB</td>
</tr>
<tr>
<td>Cable Loss</td>
<td>-2 dB</td>
<td>Link Margin</td>
<td>3 dB</td>
</tr>
<tr>
<td>Rx signal power</td>
<td>-117+35-1-2 = -85 dBm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Raised cosine pulses have an extremely important attribute: at the ideal sampling points, they don’t interfere with each other.

Over an ideal channel, delayed transmit signal will be observed at the receiver.

Ideal channel: \( s_{received}(t) = s_{transmit}(t - \delta) \)
Channel Equalization Techniques

A practical LMS adaptive equalizer:

This is a decision directed equalizer because training is based on the slicer decisions about the transmitted symbols.

Sometimes the slicer output is replaced by a training symbol sequence known a priori at the receiver.
Channel Equalization Techniques

The effect of decision delay on the LMS adaptive equalizer:

![Diagram of a seven symbol LMS equalizer with a four symbol delay]

<table>
<thead>
<tr>
<th>Symbol Times:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taps:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T(0)</td>
<td>X(0)</td>
<td>X(1)</td>
<td>X(2)</td>
<td>X(3)</td>
<td>X(4)</td>
</tr>
<tr>
<td>T(1)</td>
<td>X(-1)</td>
<td>X(0)</td>
<td>X(1)</td>
<td>X(2)</td>
<td>X(3)</td>
</tr>
<tr>
<td>T(2)</td>
<td>X(-2)</td>
<td>X(-1)</td>
<td>X(0)</td>
<td>X(1)</td>
<td>X(2)</td>
</tr>
<tr>
<td>T(3)</td>
<td>X(-3)</td>
<td>X(-2)</td>
<td>X(-1)</td>
<td>X(0)</td>
<td>X(1)</td>
</tr>
<tr>
<td>T(4)</td>
<td>X(-4)</td>
<td>X(-3)</td>
<td>X(-2)</td>
<td>X(-1)</td>
<td>X(0)</td>
</tr>
<tr>
<td>T(5)</td>
<td>X(-5)</td>
<td>X(-4)</td>
<td>X(-3)</td>
<td>X(-2)</td>
<td>X(-1)</td>
</tr>
<tr>
<td>T(6)</td>
<td>X(-6)</td>
<td>X(-5)</td>
<td>X(-4)</td>
<td>X(-3)</td>
<td>X(-2)</td>
</tr>
<tr>
<td>4 Symbol Dly</td>
<td>Y(0)</td>
<td>Y(1)</td>
<td>Y(2)</td>
<td>Y(3)</td>
<td>Y(4)</td>
</tr>
<tr>
<td></td>
<td>Y(0)</td>
<td>Y(1)</td>
<td>Y(2)</td>
<td>Y(3)</td>
<td>Y(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Y(0)</td>
<td>Y(1)</td>
<td>Y(2)</td>
<td>Y(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y(0)</td>
<td>Y(1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y(0)</td>
</tr>
<tr>
<td>Error Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E(0)=D(0)-Y(0)</td>
</tr>
</tbody>
</table>
Channel Equalization Techniques

Group delay variation = 0.0154194 (nsec), Percent of symbol time = 0.770968

Green trace is the composite group delay of the Cheby II analog LPF and the digital equalizer.
Channel Equalization Techniques

\[ M F F_{sym} < B_{coh} \]

\[ B_{coh} \approx \frac{1}{\text{RMS Delay Spread}} \]

\[ F_{sym} = N_{OFDM} \]
\[ T_{sample} = \text{OFDM symbol rate} \]

\[ M F F_{sym} = \text{Frequency domain spacing, Hertz} \]

\[ B_{coh} = \text{RF channel coherence bandwidth} \]

\[ N_{OFDM} = \text{Number of OFDM channels} \]

\[ T_{sample} = \text{Sample rate of each input signal} \]
IEEE 802.11a/g transmits rectangular QAM on data subcarriers and BPSK on pilot subcarriers.

Pilot symbols are known at the receiver. Pilot symbols can be constant vectors or some repeating pattern.

Trade-off: Too many pilots reduce throughput and lower SNR because less power is available for information channels. Too few pilots may reduce channel estimation accuracy.
Channel Equalization Techniques

\( N_p \) pilot subcarrier transmit complex signals written as a diagonal matrix:

\[
X_p = \begin{bmatrix}
 x_p(0) & 0 & 0 \\
 0 & \ddots & 0 \\
 0 & 0 & x_p(N_p - 1)
\end{bmatrix}
\]

A vector of \( N_p \) pilot subcarrier complex channel responses:

\[
H_p = \begin{bmatrix}
 h_p(0) & h_p(0) & \cdots & h_p(N_p - 1)
\end{bmatrix}^T
\]

Receive pilots are now: \( Y_p = X_p H_p \)

Unique least squares solution is:

\[
H_p = X_p^{-1} Y_p = \begin{bmatrix}
 y_p(0) & y_p(1) & \cdots & y_p(N_p - 1) \\
 x_p(0) & x_p(1) & \cdots & x_p(N_p - 1)
\end{bmatrix}
\]

See [E13,14,15] for more details
De-Spreading results in $10\log(4) = 6$ dB noise reduction, called Processing Gain:

$$P_{\text{noise}} = \text{Total Noise Power}$$

$$N_{\text{noise}} = \frac{P_{\text{noise}}}{2F_{\text{chip}}} = \text{Noise Power Density, Watts/Hz}$$

$$P'_{\text{noise}} = 2F_{\text{bit}}N_{\text{noise}} = \frac{2F_{\text{bit}}P_{\text{noise}}}{2F_{\text{chip}}} = \frac{P_{\text{noise}}}{4}$$
Multiple Access Techniques

Spread Spectrum is used for Multiple Access:

Correlating the composite received signal with spreading code 1 will only despread signal 1. Signal 2 will remain spread and look like noise.

Within the information bandwidth the desired signal power will be much larger than the interfering power.
0 = extra distance the wavefront must travel to antenna element E1
L_2 = extra distance the wavefront must travel to antenna element E2
L_2 + L_3 = extra distance the wavefront must travel to antenna element E3

For a given carrier frequency \( f \) and speed of light \( C \), extra distance corresponds to phase shift in radians, for example: 
\[ \phi_3 = 2\pi f C (L_2 + L_3) \]
Element receivers compensate this phase to create a main lobe at any angle
Source and Channel Coding

There is *information* in the outcome of a random event. Consider a random variable $X$ with sample space $S_X = \{A, B\}$. Outcomes $A$ and $B$ represent the only two messages a weather station in Death Valley, CA can send.

Message A: “It is sunny” $P(A) = 0.9$
Message B: “It is raining” $P(B) = 0.1$

Self information in these two messages is:

$$h(A) = - \log_2 \left( p_x (A) \right) = 0.152$$
$$h(B) = - \log_2 \left( p_x (B) \right) = 3.322$$

Notice that the self information in a rare event is greater, kind of makes sense!

The average (expected value) of self information is called the *entropy*:

$$H(X) = - \left( p_x (A) \log_2 (p_x (A)) + p_x (B) \log_2 (p_x (B)) \right)$$

$$= -0.9(-0.152) - 0.1(-3.322) = 0.47\text{bits}$$

$H(X)$ is the minimum *average* number of bits to encode this source.
Source and Channel Coding

Entropy coding removes redundancy, but not necessarily in real time. Voice coding removes redundancy from, generally, real time sampled voice.

The voice coder works continuously to reduce the data rate needed to represent the sampled voice. Low latency is important for phone conversations.

Let’s look at efficient ways to obtain the voice samples...
Start with a BER requirement, say $10^{-5}$. Plot the required EbN0 points for 4PSK, 8PSK with no channel coding.

For the same C/W, we can operate at lower BER with more complicated coding.

See [A19] for a more detailed version of this chart.
We select 8PSK for modulation, symbol rate = 9600/(log₂M)= 3600 symbols/sec, for M = 8. This signal BW is less than our channel BW of 4000 Hz, so the signal ‘fits” However, still need BER < 10⁻⁹ . Let’s see were we are now. From [D30] for PSK, M>2 we have:

\[
Pr_{symerr}(M) = 2Q \left[ \sqrt{\frac{2E_s}{N_0}} \sin \left( \frac{\pi}{M \sqrt{2}} \right) \right], \quad Q(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-u^2} du \approx \frac{1}{x\sqrt{2\pi}} e^{-\frac{x^2}{2}}
\]

\[
\frac{E_s}{N_0} = \frac{E_b}{N_0} \frac{R}{R_{sym}} = \left( \log_2 M \right) \frac{E_b}{N_0}, \quad Pr_{biterr} = \frac{Pr_{symerr}}{\log_2 M}
\]

For 8PSK, EbN0 = 13.0 dB, these equations result in BER = 1.47889e⁻⁵, not good enough. To meet the BER < 1e⁻⁸ requirement, we turn to block coding.

The uncoded minimum BW requirement is 3200 Hz. We have 4000 Hz available. Thus we can utilize a block code that improves BER by increasing signal BW by 25%.
8PSK BER vs EbN0 (dB)

Uncoded BER
Coded Demod Output
Decoder Output
Step 7: Traceback from largest final PM

Note traceback bit sequence: 1 1 1 0 1, reversed = 1 0 1 1 1 = transmit bit sequence
Source and Channel Coding

Turbo Decoder diagram for this example:

The maximum a posteriori (MAP) log likelihood ratio:

\[
L_{MAP}^{col} = \log \left( \frac{p(S_1 | r)}{p(S_0 | r)} \right) = \log \left( \frac{p(r | S_1)}{p(r | S_0)} \right) + \log \left( \frac{p(S_1)}{p(S_0)} \right) = L_{ML} + L_{AP}^{col} = L_{ML} + L_{EXT}^{row}
\]
Source and Channel Coding

Column decode generates new extrinsic information

Transmit Data
Row parity
Column parity

Receive Log Likelihood Ratios

\[ L_{ML}(r) = \log \left( \frac{P(r \mid S_1)}{P(r \mid S_0)} \right) \]

MAP Log Likelihood ratios from column decoding

\[ L_{MAP}^{col} \]

Extrinsic Information derived from column decoding

\[ L_{EXT}^{col}(0,0) = L_{MAP}^{col}(1,0) \odot L(p_{c0}) = (3.0) \odot (2.0) = 2.0 \\
L_{EXT}^{col}(0,1) = L_{MAP}^{col}(1,1) \odot L(p_{c1}) = (-0.5) \odot (-2.5) = 0.5 \\
L_{EXT}^{col}(1,0) = L_{MAP}^{col}(0,0) \odot L(p_{c0}) = (1.5) \odot (2.0) = 1.5 \\
L_{EXT}^{col}(1,1) = L_{MAP}^{col}(0,1) \odot L(p_{c1}) = (2.0) \odot (-2.5) = -2.0 \]

12/8/2015 © John Reyland, PhD
Now, let’s review RF mixer characteristics:

Note: in this example the spectrum is inverted; RF low-high goes to IF high-low.
Analog Signal Processing

NZIF: Single Conversion to a non-zero intermediate frequency
We will start with this. First consideration – frequency planning
Typical response for preselector bandpass filter, Bandwidth = 45 MHz (~1.9% of center frequency), Insertion loss < 1dB

This is an example of an ISM band channelizing filter
For gain planning, receiver has to cope with *contradicting* requirements

**ADC Input:** Only one optimum power level for best performance

\[ \text{Max ADC input} = \text{received signal peak to average power ratio (PAPPR)} \]

**Antenna Input:** Needs to handle wide range of inputs from -100 dBm or less to 0 dBm or more

\[
P_{dBm} = 10 \log_{10} \left( \frac{P_{watts}}{0.001} \right) \quad P_{watts} = 0.001 \left( \frac{10^{P_{dBm}/10}}{} \right)
\]

\[-100 \text{ dBm} \Rightarrow 10^{-10} \times 10^{-3} = 10^{-13} = 0.1 \text{ picowatt} \]

\[0 \text{ dBm} \Rightarrow 1 \text{ milli watt} \]

We start at the ADC input and plan backwards.
Continuing on, we can calculate the receiver noise at the ADC input.

Composite Noise Figure up to the VGA output:

\[
NF_{RxdB} = 10 \log_{10} \left[ \frac{F_1 - 1}{G_0} + \frac{F_2 - 1}{G_0 G_1} + \frac{F_3 - 1}{G_0 G_1 G_2} + \frac{F_4 - 1}{G_0 G_1 G_2 G_3} \right]
\]

Composite Gain (max AGC gain):

\[
G_{RxdB} = 10 \log_{10} \left[ G_0 G_1 G_2 G_3 G_4 \right]
\]

\[
P_{RXnoisedBm} = P_{NdBm} + NF_{RxdB} + G_{RxdB}
\]
ADC dynamic range details for one in-band signal:

Signal power must be backed off by peak to average power ratio (PAPR). For OFDM:

\[
20 \log_{10} \left( \frac{V_{FS}}{V_{ADC_{max}}} \right) \approx 8 \text{dB}
\]

Minimum SNR required by the modulation type:

\[
SNR_{Req} = 20 \log_{10} \left( \frac{V_{ADC_{min}}}{V_{NoiseFloor_{ADC}}} \right)
\]

Note: ADC driver amplifier must also have sufficient dynamic range.
Analog Signal Processing

ADC Integral Nonlinearity (DNL)

Independent of signal amplitude
Generated by encoding process
Spread across ADC range to minimize average effect for full scale inputs

ADC Differential Nonlinearity (DNL)

Similar to amplifier nonlinearity
Generated by ADC sample and hold process and/or input analog circuits

© John Reyland, PhD
ADCs do not have the predictable spurious responses of amplifiers!

**Amplifier:** Increase power out and you get a corresponding threefold increase in third order IMD. Design requirement: Amplifier chain must have high enough IP3 so the IMD is kept below the ADC noise floor.

**ADC:** Considered linear for signals corresponding to correct input voltage range; IP3 is infinite. However, the ADC transfer curve contains small random quantizing errors called differential nonlinearity (DNL) as well as only all quantizing nonlinearity called (INL).

DNL is fixed relative to the converter’s full scale range. Good ADC design can spread out DNL effect over the entire input range. This means that for a full scale input, INL generates most of the distortion. For a smaller input, ADC spurs start to be more influenced by DNL.

ADC SFDR is the ratio of the RMS signal amplitude (either full scale or input signal level) to the RMS value of the peak spurious spectral content.
ADC frequency domain analysis shows noise advantage

\[ H(s) = \frac{1}{s}, \text{ lowpass integrator (could also be highpass, centered at the signal IF).} \]

\[ U(s) = H(s) \left( X(s) - W(s) \right) \]

\[ Y(s) = U(s) + E_q(s) \]

\[ Y(s) = H(s) \left( X(s) - Y(s) \right) + E_q(s) \]

\[ Y(s) = \left( \frac{H(s)}{1 + H(s)} \right) X(s) + \left( \frac{1}{1 + H(s)} \right) E_q(s) \]

\[ Y(s) = S(s) X(s) + N(s) E_q(s) \]
One bit ADC: small, low power, cheap and flexible

Commonly used for satellite and space craft communications because of the power savings and also because it does not need to tolerate adjacent channel interference in space.
ADC performance is critical for Software Defined Radio:

UMTS Advanced Wireless Service (AWS) Uplink specifies both GSM and WCDMA channels

A basestation SDR designed for both receives in-band WCDMA or *multiple* in-band GSM signals
A complex representation is required at baseband because the modulation will cause the instantaneous phase to go positive or negation:

\[ e^{j\theta_{BB}(t)} = \cos(\theta_{BB}(t)) + j \sin(\theta_{BB}(t)) \]

After up-conversion, phase is always positive and complex exponential terms are redundant

\[ e^{j(\omega_{RF}t + \theta_{BB}(t))} = \cos(\omega_{RF}t + \theta_{BB}(t)) + j \sin(\omega_{RF}t + \theta_{BB}(t)) \]

Signal now can be real: \( \cos(\omega_{RF}t + \theta_{BB}(t)) = e^{j(\omega_{RF}t + \theta_{BB}(t))} + e^{-j(\omega_{RF}t + \theta_{BB}(t))} \)

This forces the existence of a negative RF image (ignored for most analog processing):
Let’s build a direct conversion ZIF receiver for:
802.11g Wireless LAN, Center frequency = 2412 MHz, Bandwidth = 20 MHz
Downconvert this channel to 0 MHz using complex LO = 2412 MHz
As before, we pre-filter the RF with a 45 MHz bandpass filter

Note that there is no image frequency for ZIF
Also, because the image frequency is not moving with tuning frequency, there is a potential for wide tuning range
ZIF introduces new problems, now for the bad news:

1. DC offset due to on channel local oscillator feeding back to RF input
2. DC offset due to second order harmonic distortion
3. DC offset due to IF amplifiers
4. 1/f noise at output of IF amplifiers
5. IQ imbalance of quadrature downconverting mixer

We will look at each of these drawbacks ....
# Analog Signal Processing

## Power amplifier class information

<table>
<thead>
<tr>
<th>Class</th>
<th>DC Bias</th>
<th>Signal</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>large</td>
<td>small</td>
<td>Best linearity, worst efficiency</td>
</tr>
<tr>
<td>B</td>
<td>zero</td>
<td>large</td>
<td>Still linear if output is filtered around carrier frequency. Useful for amplitude modulation. Better efficiency because negative input excursions result in zero collector current.</td>
</tr>
<tr>
<td>C</td>
<td>negative</td>
<td>large</td>
<td>Not linear, Input amplitude modulation not accurate on output. Useful for phase modulated, constant envelope signals only. Best efficiency</td>
</tr>
</tbody>
</table>
Some signals, for example CDMA, have amplitude modulation peaks that must be reproduced at the PA output.

Class A,B PA will produce AM-AM distortion if driven hard enough.

As shown, AM-AM can be prevented by operating at lower input power, however this solution wastes power.
Analog Signal Processing

AM-AM distortion applied to the CDMA signal below causes intermodulation distortion and adjacent channel interference due to spectral regrowth.
Another example is the IEEE 802.11 preamble

$B_k$ blocks are 16 sample identical known sequences; 16 samples at $F_s = 20\text{MHz}$. A delay and correlate technique provides initial packet detection:

Adapted from [D39]
Generalized bandpass sampling:

Floor function even: No spectral inversion, $F_{IF}$ offset from 0 Hz

$$\left[ \frac{F_c}{0.5F_s} \right] = 4$$

Floor function odd: Spectral inversion, $F_{IF}$ offset from 0.5$F_s$ Hz

$$\left[ \frac{F_c}{0.5F_s} \right] = 5$$

Additional constraint

$$0 < F_{IF} - 0.5B$$

$$F_{IF} + 0.5B < 0.5F_s$$
Let's take another look at digital receiver color coded sampling rates:

- **$F_S$** = ADC sampling rate. $F_S$ is related to intermediate frequency location.
- **$F_S/K$** = Complex baseband sample rate, $K = 2$ or $4$ to reduce processing load.
- **$MF_{sym}$** = Fixed number of samples/symbol. Often $M = 2$, $4$ or $8$.
- **$F_{sym}$** = One sample/symbol. Final subsampling before decoding payload data.

Generally there is no integer relation between $F_S$ and $F_{sym}$, also $F_S/K > MF_{sym}$.

The FIFO effects a transition between real time and non-real time processing.
A DSP receiver is basically a complicated downsampler!
The symbol rate resampler needs an accurate estimate of the intersample time.

Intersample time represents the actual number of input samples in a symbol (11.5, above) divided by the desired number of samples/symbol (11.5/4 = 2.875, above).

We will look at several DSP symbol timing recovery techniques:
1. Open loop correlator
2. Open loop timing tone
3. Transition tracking
4. Gardner
5. Band edge
Delay Transition Tracking Loop is an example of closed loop symbol timing.
Performance of some symbol timing algorithms is based on excess pulse shaping bandwidth. Spectral power at +/-F_s/2 band-edges generally improves timing recover performance.
Let's match filter receive sequence $x(k)$

$$y(k) = \sum_{n=0}^{3} h(n)x(k-n)$$

$h = \{1 \ 1 \ 1 \ 1\}$

$x(k)$

Match Filter
To achieve fast response, this carrier tracking loop has low closed loop delay. We achieve that by not feeding back the carrier phase correction to the equalizer input. Instead we de-spin at the equalizer output. To make the equalizer work we must re-spin the remodulated signal that is feedback to calculate the tap update. See [D12], page 182
Typical AGC action at receive packet start:

- **Received signal envelope at ADC input**
- **Received signal envelope at DSP AGC gain output**

- **ADC Saturation**
- **Attack Threshold**
- **Hang Threshold**
- **Desired Range**
- **Envelope Detect**
- **RF Gain**
- **DSP Gain**
A better solution is to filter 4 parallel symbol samples simultaneously:

This filter design processes four sample vectors (i.e. one symbol per vector) at 200 MHz instead of just a single sample stream at 800 MHz.

Current FPGA technology will not support 800 MHz sample by sample internal DSP.
Parallel processing works for IIR filters also, consider this Direct Form II second order section:

\[
\frac{y(z)}{x(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}
\]