TSTE17 System Design, CDIO

- Lecture 4
  - Project hints and deadline suggestions
  - Modulation, cont.
  - Channel coding

General project hints

- Required documents
  - Requirement specification
  - Design specification
  - Project plan
  - Time plan
  - Project report
- 4 versions of the requirement specification
  - Other documents updated as needed (4 times at least)

General project hints, cont.

- Final presentation and demonstration
  - All group members should participate
- Next thing to do
  - Complete the first requirement specification
  - Create project plan and time plan

Behavioral Model

- Describe external behavior of each block in the design
- Used to verify block diagram and function in the complete system
- Internals not of interest

Behaviour Example Filter vs scrambler

- Filter descriptions
  - mathematical expression (transfer function)
  - May use complex blocks in simulink
  - No description of algorithm to use
- Scrambler descriptions
  - shift register with feedback (structure)
  - vector of bits xor:ed once with complete input

Functional Model

- Focus on internal implementation of behaviour
  - May introduce non-ideal effects (noise, crosstalk, aliasing, etc.)
- Used to select functional implementation of a given behaviour
- Functional imperfections influences overall performance

Functional Example Filter vs scrambler

- Filter
  - Filter structure (sequence of operations)
  - Scaling of filter
- Scrambler
  - shift register with feedback
  - multiple bit state machine

Bit-True Model

- Model function as implemented in digital environment
- Include truncation, overflow, latency etc.
- Do not describe limits on clock frequencies

Bit-True Example Filter vs scrambler

- Filter
  - Filter architecture (what hardware units, how they are interconnected, etc.)
- Scrambler
  - Same as functional.
  - May introduce extra pipelining etc.

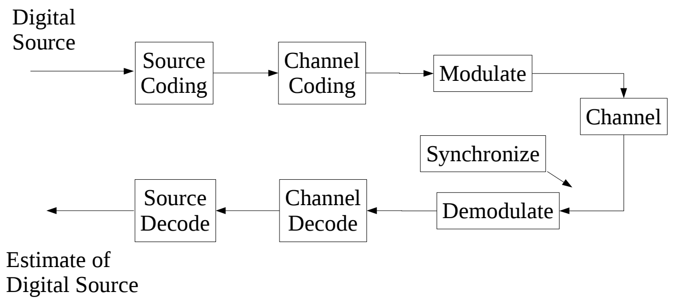
General project hints, cont.

- Models sorted by complexity
  - Model 4: Most complicated
  - Model 1
  - Model 2
  - Model 3: Least complex
- Complexity also dependent on what is included in each model
  - Synchronisation, channel estimation
  - Timing

Deadline suggestions

- Labs
  - Everyone have completed this
- Requirement specification (1st of 4 versions)
  - 1st version Monday 14/9
  - Final v1.0 Thursday 17/9
- Project plan (1st of 4 versions)
  - 1st version Tuesday 22/9
  - Final v1.0 Friday 25/10
- Weekly meetings
  - Start week 38 (18/9 latest)

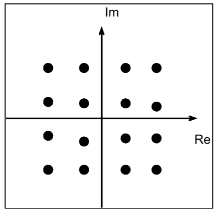
Components of a digital communication system



Quadrature Amplitude Modulation (QAM)

- Modulate both amplitude and phase
- Use equal distance between all points
- Each point represents transmission of one sinusoidal waveform with unique amplitude and phase combination

16-QAM



Coherent and non-coherent modulation

- Coherent modulation
  - requires a phase lock between transmitter and reciever RF carrier waves.
  - Gives higher performance
  - Requires more complex reciever structure
- Non-coherent modulation
  - Simpler reciever structure
  - Can not use QAM, PSK, ASK

Non-Coherent Modulation

- RF carriers not synchronized
  - Amplitude may vary
  - Phase may vary
- ASK, PSK, QAM does not work
- Use Differential modulation
  - Differential PSK (DPSK)
  - Differential APSK (DAPSK)
- Compares previous symbol with current

Differential PSK (DPSK)

- Constellation equal to PSK
- Difference is in mapping of bits
  - Binary DPSK

$$b_n = d_n \oplus d_{n-1}$$

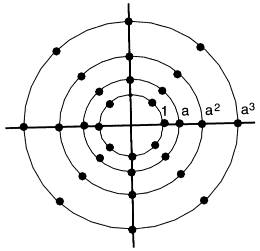
- Used for low data rates systems
- Used if simple receiver structure is needed

Differential Modulation Detection

- Two step procedure
  - Remove differential encoding
  - Use normal demodulation as in coherent modulation
- Two symbols used for each detection
- Double amount of noise per detected symbol

Differential Amplitude Phase Modulation (DAPSK)

- Combine differential phase with differential amplitude
- Amplitude modulation uses nonuniform constellation shape
  - Needed as scaling is unknown



Shannon Channel Limit (1948)

Capacity of an AWGN channel

$$C = W \log_2 (1 + \frac{P}{W N_0})$$

C : Channel capacity

W : Bandwidth

P : Average transmitted power

N_0 : power-spectral density of the additive noise

How to reach high capacity?

- Increase bandwidth
- Increase transmitted power
- Reduce additive noise
  - Noise sources includes physical media, amplifiers, filters, etc.
- Note: Shannon capacity is an upper limit!
  - Most modulation techniques are far from the limit

Modulation Selection

- Affects many properties
  - Bit Error Rate (BER)
  - Peak to Average ratio (PAPR)
  - RF Spectrum shape
- Minimum distance (d_{min})
  - Shortest distance between any two points in a constellation
  - Determines the least amount of noise needed to generate a decision error

Bit Error Rate Calucation

- Q(x) is used in many cases to calculate probability of a bit error P_b
- Q(x) is the area under the tail of the Probability Density Function of a zero mean, unit variance normal random variable.

$$Q(x)=\frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-\frac{t^2}{2}}dt \qquad x\geqslant 0$$

$$P_b \propto Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

Modulation Comparison Criteria

- Compare d_{min} for a given energy per bit or signal symbol
  - Bit energy to noise density ratio E_b/N_0
  - Signal energy to noise density ratio E_s/N_0
- $E_s = kE_b$
  - k bits transmitted in each symbol
- Average power is scaled
  - Equally likely points

$$P_{ave}=\frac{1}{M}\sum_{k=1}^{M}|c_k|^2$$

Amplitude Shift Keying (ASK)

- 2-ASK

$$P_s=Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

- General M-ASK
  - A is minimal distance

$$P_s=2\frac{M-1}{M}Q\left(\sqrt{\frac{A^2}{2N_0}}\right)$$

- SNR increase (required to reach same BER as 2-ASK)
  - 4-ASK 6.99 dB
  - 8-ASK 6.23 dB

Phase Shift Keying (PSK)

- BPSK $$P_s = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$
- QPSK $$P_s = 2Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\left[1 - \frac{1}{2}Q\left(\sqrt{2\frac{E_b}{N_0}}\right)\right]$$
- High order PSK $$P_s = 2Q\left(\sqrt{\frac{E_s}{N_0}}\right)\sin\left(\frac{\pi}{M}\right)$$
- SNR increase to keep BER when adding one more bit
    - QPSK 3.00 dB, 8-PSK 5.33 dB, 16-PSK 5.85 dB

Quadrature Amplitude Modulation (QAM)

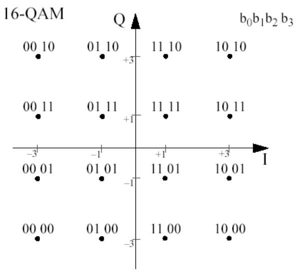
- M-QAM

$$P_s \approx 4\left(1 - \frac{1}{\sqrt{M}}\right)Q\left(\sqrt{\frac{3E_s}{(M-1)N_0}}\right)$$

- SNR increase for each additional bit (initially QPSK)
    - 8-QAM 4.77 dB
    - 16-QAM 2.22 dB
    - 32-QAM 3.01 dB

Constellation Point Labeling

- Shortest d_{min} most likely error
    - Neighbouring points should differ in as few bits as possible
    - Results in a non-natural ordering
    - Gray coding



Detection of Coherent Modulations

- Must correct for channel effects first
- Hard Decision Detection
    - Slicer
    - # Inputs to modulator equal to # outputs
- Soft Decision Detection
    - Outputs both a bit value and reliability information
    - Sign indicates bit value, magnitude indicates reliability
    - Useful information for channel coding

Channel Coding

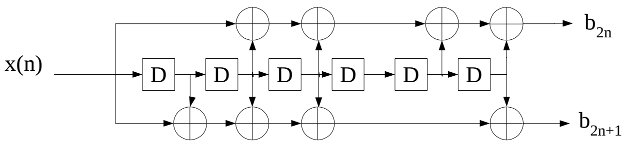
- Add capability to correct data errors
  - Requires additional information to be sent
  - Removing errors makes larger noise power accepted while keeping the same BER.
- Performance measured as coding gain
  - How much can E_b/N_0 be reduced while keeping a given BER
  - Different combinations of coding and modulation may produce equal data rates, but different SNR

Different coding types

- Block codes
  - Map k input symbols into n coded symbols, n>k
  - Example: Reed-Solomon
- Convolutional codes
  - Map k input bits in a continuous stream onto n output bits.
  - Simple structure: Convolving
  - Most commonly used

Convolutional codes

- 802.11a uses a length 7 (constraint length) convolutional encoder ½ rate
  - rate 1/2: 1 input bit => 2 output bits
  - $2^6 = 64$ different states

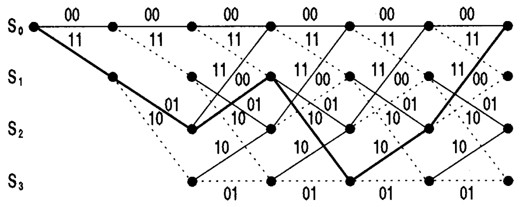


Puncturing Convolutional codes

- Doubling the number of bits in the coder may be to much redundancy
  - Want other rates: 3/4, 9/16, 2/3
- Create other rates by removing (puncture) bits in the bitstream
  - 3/4 by removing 2 out of 6 output bits (3 inputs give 6-2 = 4 outputs)

Trellis description

- The coder has $2^{\text{constrain length}}$ states. Indicate each possible state by a dot. Add a time scale. Connect dots depending on possible input.



solid line = 0 (zero) input
dashed line = 1 (one) input

Decoding

- Channel coding makes some received bit sequences impossible
  - By identifying these can errors be detected and possibly corrected
- Want to estimate the received data by the sequence of bits that gives the smallest distance metric
  - total distance between received and expected constellation points

Viterbi algorithm

- Measure distance between signals at each sampling instant t_i and all paths entering each state or node at time t_i
- Save the path with the lowest distance for each state or node at time t_i. Save the sum of the distances for each saved path.
- Advance deeper in the trellis. The surviving path is then the most likely bitstream.

Interleaving

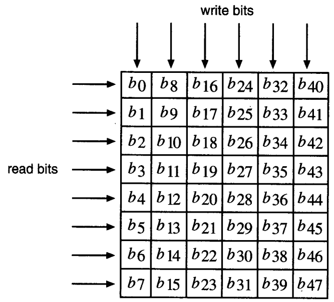
- Want to avoid get bursts of bit errors
  - Helps getting good results in channel coding by changing error distribution
- Interleaving increases delay
  - More efficient with large interleave
  - Acceptable delay often limited
    - phone to phone delay < 20 ms

Block interleaver

- Rearrange bits one block of bits at a time
- Interleaving depth is the size of the block

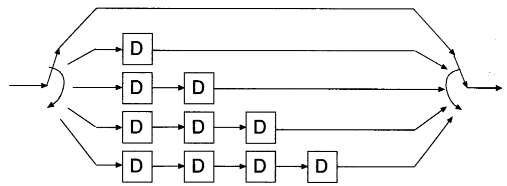
Input
$[b_0, b_1, b_2, b_3, ...]$

Output
$[b_0, b_8, b_{16}, b_{24}, ...]$



Convolutional interleaver

- Best suited for continous streams
- Smaller memory requirements compared to block interleaver
- Identical interleave and deinterleave structures

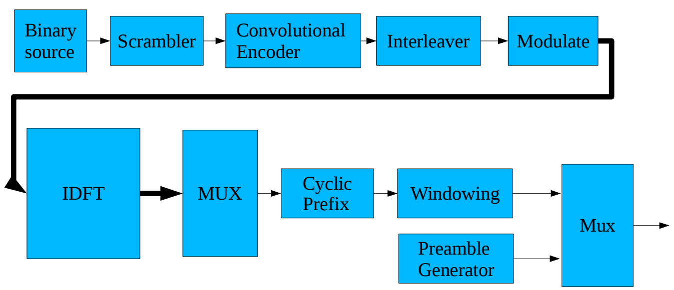


802.11a OFDM Parameters

- Bit rate 6, 9, 12, 18, 24, 36, 48, and 54 Mbps
- Modulation BPSK, QPSK, 16-QAM, 64-QAM
- Coding rates 1/2, 2/3, 3/4
- Number of subcarriers 52 (4 pilots)
- OFDM symbol duration 4 µs (800 ns guard interval)
- Signal bandwidth 16.66 Mhz
- Subcarrier spacing 312.5 kHz

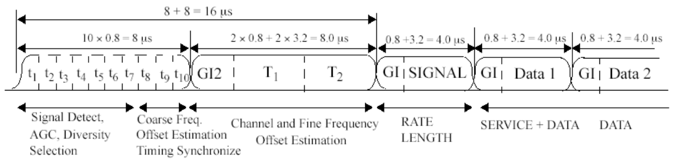
802.11a & HiperLAN/2 Transmitter Details

- Excluding interpolation, A/D, and RF circuits



802.11a Preamble

- Used to detect start of packet
- Used to synchronize receiver
- 10 short symbols + 2 long symbols



Synchronization

- Coherent modulation => Must synchronize carrier frequency
- OFDM works with frames => Must detect start of frame
- Channel is slowly changing => Must correct for changes

Packet synchronization

- Use only in packet sending applications
  - Broadcasting system does not need them
- Task: Find start of the preamble of an incoming packet
- Two possible values
  - H_0 packet not present
  - H_1 packet present

Packet Detection

- Usual test
  - $H_0 : m_n < Th$ => Packet not present
  - $H_1 : m_n \geq Th$ => Packet present
  - m_n is a decision variable
  - Th is a threshold

Packet detection performance

- Probability of detection P_D, should be as large as possible
- Probability of false alarm P_{FA}, should be as low as possible
- Want high P_D and low P_{FA}, but increasing P_D generally increases P_{FA}
- Generally worse with low P_D

Packet detection algorithms

- Received Signal Energy Detection
- Double Sliding Window Packet Detection
- Using the preamble structure

Packet detection algorithms

- Received Signal Energy Detection

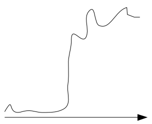
$$m_n = \sum_{k=0}^{L-1} r_{n-k} r^*_{n-k} = \sum_{k=0}^{L-1} |r_{n-k}|^2$$

- L samples added to reduce influence of noise
- The change of noise indicates start of packet

Received Signal Energy Detection

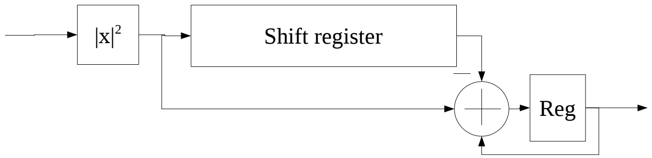
- Moving sum of signal energy

$$m_{n+1} = m_n + |r_{n+1}|^2 - |r_{n-L+1}|^2$$



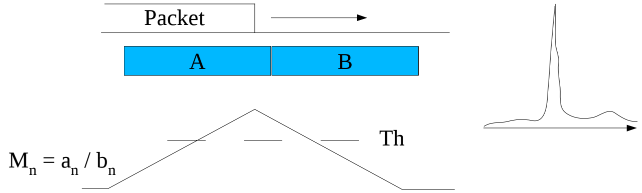
- One complex multiplication/sample, L samples stored in memory
- Drawback: Threshold depends on signal energy!

Moving Sum Implementation



Double Sliding Window Packet Detection

- Compute m_n as ratio between two consecutive sliding windows



Double Sliding Window Packet Detection

- Two sliding windows
  - One complex multiplication, one division, storage for all values

$$m_n = \frac{a_n}{b_n} = \frac{\sum_{m=0}^{M-1} r_{n-m} r_{n-m}^*}{\sum_{l=1}^{L} r_{n+l} r_{n+l}^*} = \frac{\sum_{m=0}^{M-1} |r_{n-m}|^2}{\sum_{l=0}^{L} |r_{n+l}|^2}$$

Double Sliding Window Packet Detection

- Can be used to estimate the received SNR

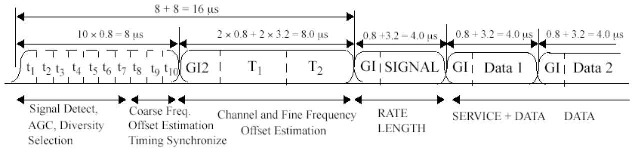
$$m_{peak} = \frac{a_{peak}}{b_{peak}} = \frac{S+N}{N} = \frac{S}{N} + 1$$

$$\widehat{SNR} = m_{peak} - 1$$

- Does not use known information about expected format of the preamble

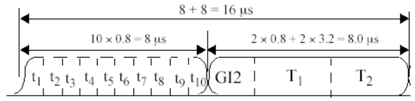
Using the Structure of the Preamble

- Use as much information as possible
- Preambles in IEEE802.11a and HIPERLAN/2 have been designed to ease detection



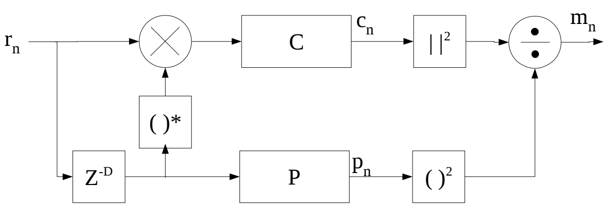
Preamble components

- t1 to t10 are short training symbols
  - Identical 16 samples long
- G12 is a cyclic prefix
  - 32 samples long
- T_1 and T_2 are long training symbols
  - Identical 64 samples long

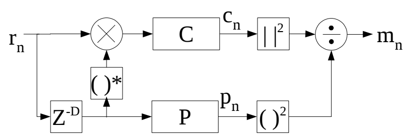


Delay and Correlate Algorithm

- Take advantage of periodicity of the short training symbols
  - Correlate two consecutive short symbols (c_n)
  - Normalize with signal power (p_n)



Delay and Correlate Algorithm



$$c_n = \sum_{k=0}^{L-1} r_{n-k}\, r^*_{n+k+D}$$

$$p_n = \sum_{k=0}^{L-1} r_{n+k+D}\, r^*_{n+k+D} = \sum_{k=0}^{L-1} |r_{n+k+D}|^2 \qquad m_n = \frac{|c_n|^2}{(p_n)^2}$$

HiperLAN/2 Preambles

- Multiple preambles, different lengths
- General structure
  - Two waveforms A and B
  - Inverted versions of the waveforms IA and IB
- Broadcast packet preamble

A	IA	A	IA	IA	B	B	B	B	IB	CP	C	C

Generates a zigzag detection output

HiperLAN/2 Preambles

- Downlink packet preamble (stations already synchronized)

CP	C	C

- General uplink preamble

B	B	B	B	IB	CP	C	C

- Long uplink preamble (antenna diversity)

B	B	B	B	B	B	B	B	B	IB	CP	C	C