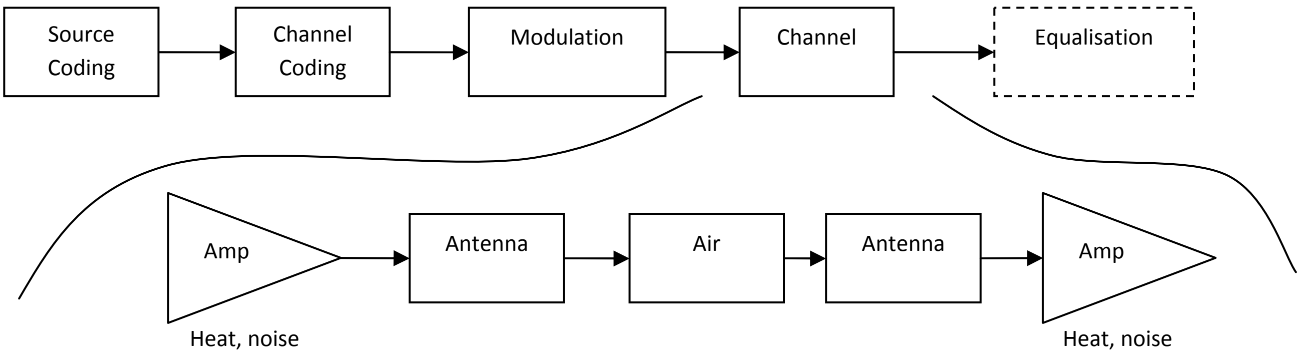
ELEC4505 Digital Communication Systems

Background Knowledge

- Fourier Transforms
    - $F[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt$
- Convolution – an superposition of impulse responses after a signal has been broken into impulses
    - $y = \int s(\tau)h(t-\tau)d\tau$
- Probability
    - PDF
    - CDF
    - Gaussian PDF
    - Q-function
    - Bayes rule

$$P(E_i|A) = \frac{P(E_i)P(A|E_i)}{\sum_{j=1}^{n} P(E_j)P(A|E_j)}$$

Digital Communication System



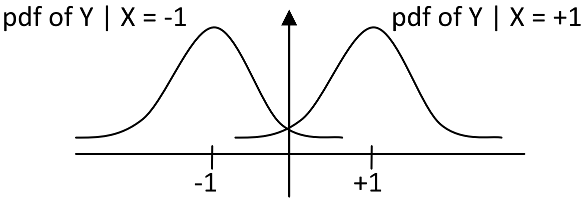
- GSM – speech
- UMTS – mobile data
- WiFi – portable data, no handovers
- BT – short range
- Zigbee – low power, low data rate

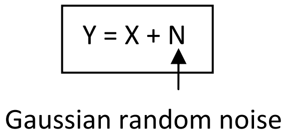
Entropy and Source Coding

Information sources

- Most sources seem to be random, so we model them using rules of probability for its behaviour on average.
- We can model a random source using a probability density function (pdf) that describes a distribution, normalised such that the area under the curve is 1.
- Central Limit Theorem

- If we add up a lot of independent and identically distributed (iid) events, the result is a Gaussian distribution, regardless of the individual pdfs of each event.





-
    - Capital 'Y' – the random variable (RV)
    - Lower case 'y' – individual realisations of the RV
- Def. Expected value:

$$E[Y] = \int_{-\infty}^{\infty} y\, f_Y(y)dy$$

- We are interested in the probability of error. Area under the curve that crosses the decision line:

$$P_E = P(Y < 0|X = +1)$$

- Adding two RV's together = convolving the two pdf's (Hence the two bell curves)

Entropy – a measure of information content

- Determines how much we can compress information
- Continuous decreasing function of probability of source output: $I(p_i)$
- The least probable outcome conveys the most information
- Information adds if events are independent:

$$p_3 = p_1 p_2 \implies I(p_3) = I(p_1) + I(p_2)$$

- The only function that fits all of the above 3 properties is the log function $\log(AB) = \log(A) + \log(B)$
- Taking log of base 2 gives information content in number of bits (base e gives nats)
- $I(p_i) = -\log(p_i)$
- Entropy of a discrete random variable (RV) X is a function of probability mass function (PMF):

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i$$

- Joint entropy of two discrete RVs (X, Y):

$$H(X,Y) = -\sum_{x,y} p(x,y) \log p(x,y)$$

- Conditional entropy for RV X given the RV Y:

$$H(X|Y) = -\sum_{x,y} p(x,y) \log p(x|y)$$

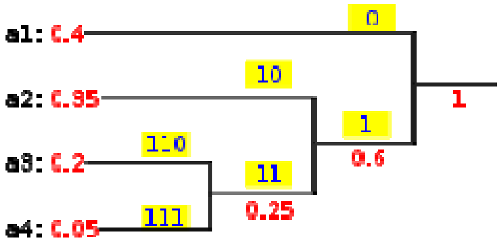
Source coding theorem

- Source with entropy H can be encoded with arbitrarily small error probability at any rate R as long as $R \geq H$
- Fundamental limit of source coding

Source coding algorithms

- **Huffman Coding**
    - Self synchronising
    - Achieves optimal R = H with certain source outputs

- Algorithm
  - Write source outputs in decreasing order or probability
  - Make a tree by iteratively combining the two least probable branches
  - Go backwards on the tree branches, writing 0 on top branch and 1 on the bottom branch
- Performance of Huffman coding
  - $H(X) \leq R_{Huffman} < H(X) + 1$
- Example:



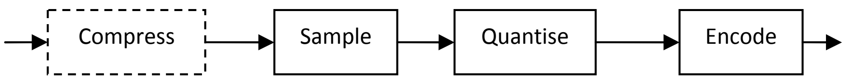
- **Lempel-Ziv Coding**
  - Universal source coding algorithm, independent of source statistics
  - Huffman code relies on source probability statistics, so requires 2 passes
  - Any sequence of source outputs is uniquely parsed into phrases of varying length
  - Parsing – identify phrases of smallest length that have not appeared so far

Quantisation

- Scalar quantisation
  - Operates on single-source outputs
  - Maps each real-numbered source output into a number of levels
  - Used in wireline telephone. Changes an amplitude to a code
- Vector quantisation
  - Operates on blocks of source outputs (converts multiple samples at once)
  - Quantising in ≥ 2 dimensions
  - Less distortion with the same number of bits per source output
- Example: Audio signal
  - I can sample uniformly at the Nyquist rate.
  - Use uniform quantisation (uniform amplitude steps)
  - I can use scalar quantisation (doing it one sample at a time), representing as a mapping between input and output amplitude
  - 8 kilosamples per second × 8 bits per sample → 64 kb/s
  - Or I can use vector quantisation, and do it X samples at a time.

Waveform Coding

- Waveform coding – designed to reproduce the source without paying attention to generation mechanism
- **Pulse-code modulation (PCM)**

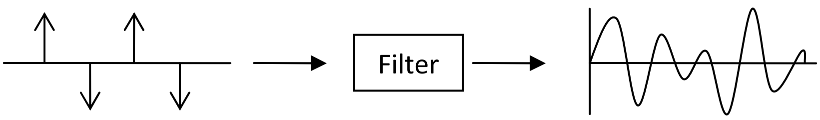


  - Uniform or non-uniform through use of compressor and expander (companding)
- Differential pulse-code modulation (DPCM)
  - Quantising: difference between 2 adjacent samples are quantised
  - In reconstruction, the current PCM value is added to the previous one to give the actual value
- Delta modulation (DM)
  - 1-bit DPCM extreme case

- High quantisation noise – needs much higher sampling rate than Nyquist
- Used in many CD players

Analysis-synthesis coder

- Based on a model for the mechanism that produces the waveform
- **Linear predictive coding (LPC)**
    - Models voice, and only stores parameters to the model which gives the sound
    - Voiced speech = impulse train at certain frequency; unvoiced perhaps white noise
    - Separates speech into milliseconds of sampled phonemes, and adds parameters e.g.
        - 1 bit voiced/unvoiced sound
        - 6 bits frequency
        - 5 bits gain (volume)
        - 10 bits linear filter coefficients (5 time delays × 2 bits each) to model shape of mouth
    - Models vocal chords as an impulse train

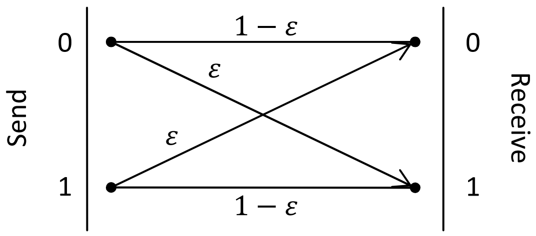


- Typically 2.4 kb/s

Channel Coding

Channel

- Discrete channel – input and output variables are finite
- Binary symmetric channel



- Binary – either 1 or 0 only
- Symmetric – same **crossover probability** $p_{10} = p_{01} = \epsilon$
- To overcome error, need to introduce redundancy (i.e. reduce data rate)

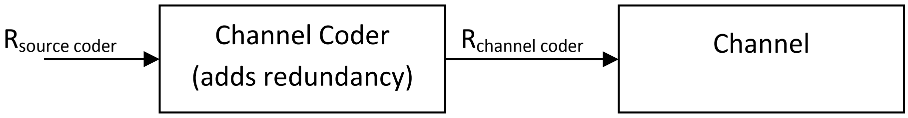
Channel capacity

- Channel capacity is like source entropy – defines a limit
- Shannon's noisy channel-coding theorem (1948)
    - Basic limitation in communication channel is not on the reliability but speed of communication
    - Reliable transmission: Error probability → 0 as block length → ∞
    - The capacity of a discrete memoryless channel is given by:

$$C = \max_{\mathrm{p(x)}} I(X;Y)$$

where $I(X;Y)$ is the mutual information between input X and output Y

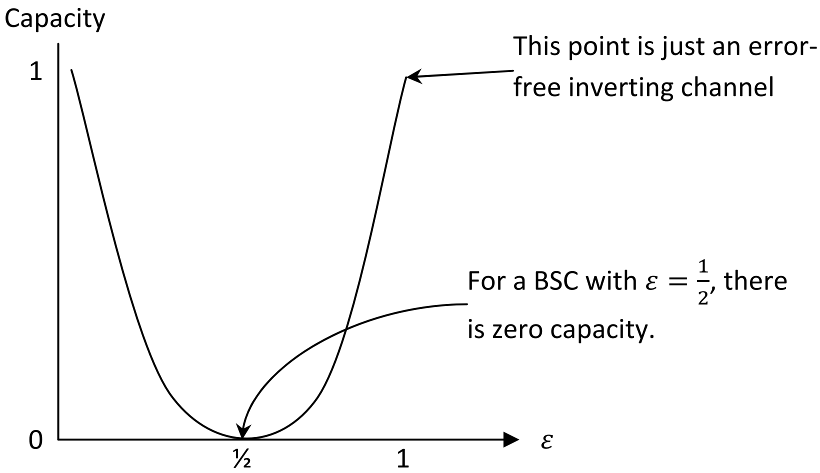
- To overcome errors, we need to introduce redundancy (i.e. reduce data rate) such that $R_{CC} \geq R_{SC}$



- Bounds on communication
  - The capacity of an additive white Gaussian noise channel is given by:

$$C = W \log\left(1 + \frac{P}{N_0 W}\right)$$

  - Capacity can be increased by increasing signal power P, but with diminishing returns
  - Can also be increased by increasing channel bandwidth W, but also increases noise power N_0



Channel coding algorithms

- To transmit messages reliably over a noisy channel without an exponential increase in bandwidth
- **Linear block codes**
  - (n, k) block code separates messages into blocks of k bits.
  - Maps each block of k bits into n bits as an output – by reading the code book.
  - Example: $\frac{1}{2}$ rate parity
    - [1 1 0] → [1 1 0 0 0 1]
    - Append 3 bits of parity 0. First bit takes parity of bits (1,1,1), second (1,1,0), then (0,1,1)
    - To interpret, add up each set of bits: (1,1,1,1,0,0), (1,1,0,0,1,0), (0,1,1,0,0,1) – all should be 0
  - Repetition code
    - Simply repeats the message
  - Evaluation methods
    - Error detection capability
    - Error correction capability
    - Max number of bits in error that the scheme can detect/correct
  - Generator matrix – represents block code book as a matrix
    - So that the code can be found by matrix multiplication

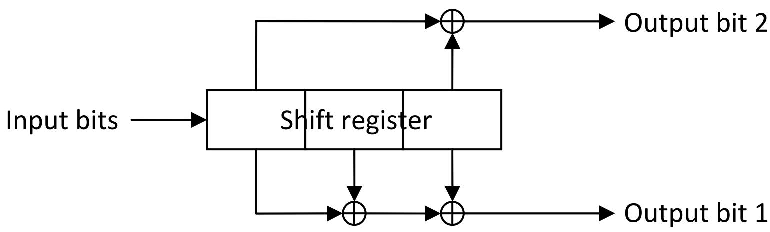
$$\boldsymbol{c} = \boldsymbol{x} \times \boldsymbol{G}$$

    - To decode:

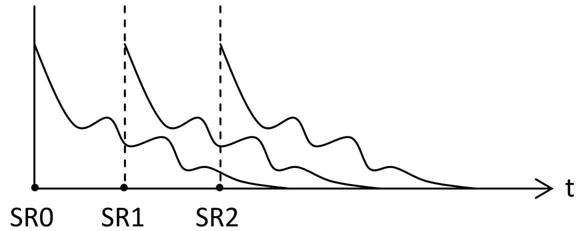
$$\boldsymbol{c} \times \boldsymbol{G}^{-1} = \boldsymbol{x}$$

  - E.g. $(1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1) = (1 \quad 1 \quad 0)\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$
  - What is the optimal generator matrix? Is the rank enough?

- Convolutional codes
  - Streams bit output rather than a block at a time
  - Block codes enforces a delay proportional to block code length
  - Convolutional codes add memory to encoder; each output dependent on adjacent ones.
  - Consists of a shift register and an arrangement of adders
  - For example:



  - Can also be viewed as a finite impulse response filter

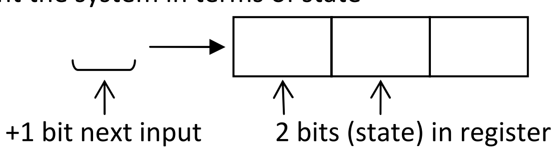


  - Need to equalise the channel since memory is introduced
  - A convention is to have the shift register initialised to all zeros before using
  - Receiver needs to use sequence detection because each input bit affects multiple output bits.
  - Example using above encoder:

0 0 1 1 0 1

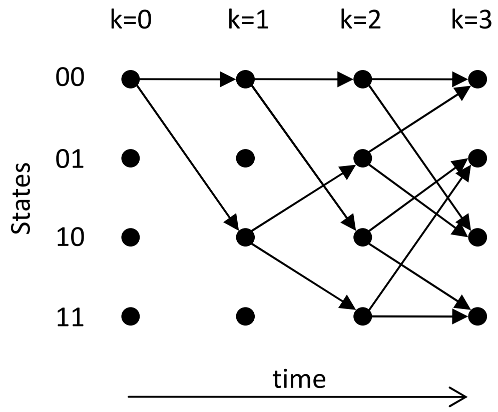
SR	Out	
1 0 0	1 1	
0 1 0	1 0	
1 0 1	0 0	
1 1 0	0 1	Mapping not unique; need to account for memory when decoding
0 1 1	0 1	
0 0 1	1 1	

  - To represent the system in terms of state



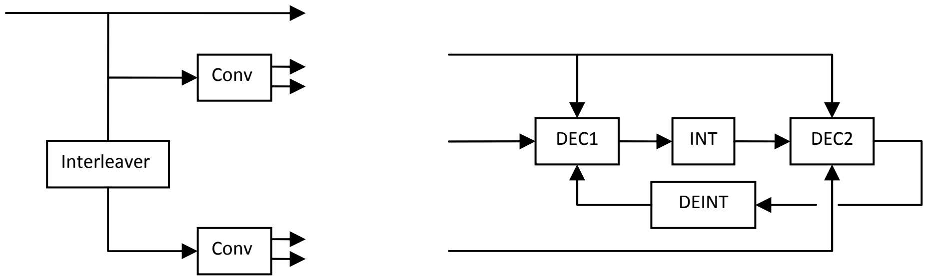
    - Gives 4 states plus the input which defines the transitions
    - Looking at state diagrams, we can see that half of the state transitions cannot be possible – this is the redundancy.

- Trellis diagram



  - To preset the system to 00, add '00' to start of bitstream
  - Shows possible paths of a system
  - Each block of output represents possible state transitions at that time. Match up possible paths of state transitions to decode.
- Viterbi algorithm is a dynamic programming algorithm that keeps probable paths through the trellis

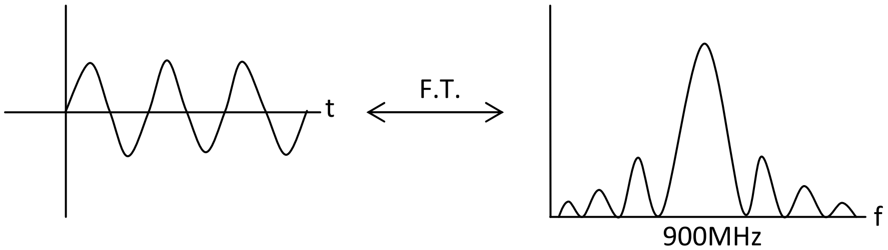
- **Turbo codes**
  - We can use interleaving to spread out bursts of errors
  - In turbo code, we send 2 parallel streams of data



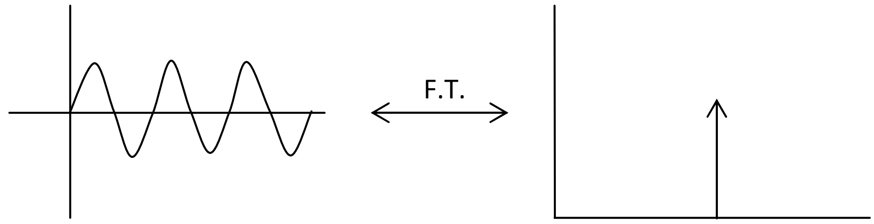
    - DEC1 and DEC2 eventually converges
    - Achieves bit rate close to Shannon's limit

Modulation

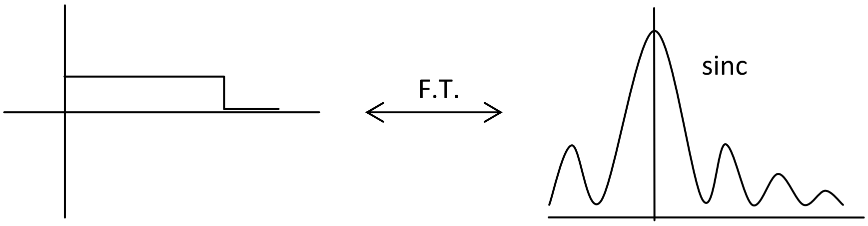
- Degrees of freedom
  - Not frequency (limited to 900MHz for GSM)
  - Amplitude and phase only



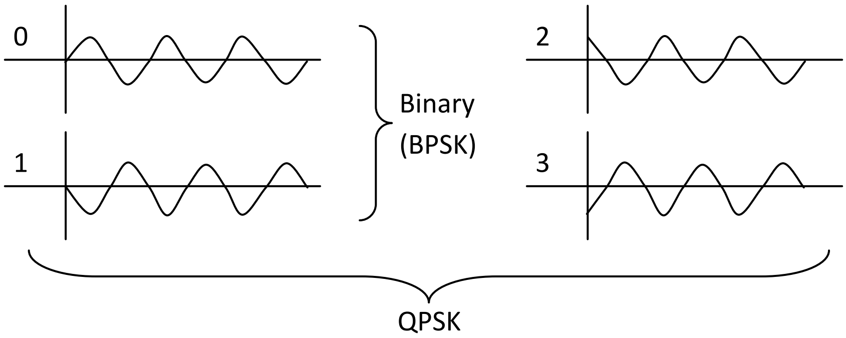
- Oscillator:



- Data:



- Multiplying the signal in time domain ↔ convolving the spectrum

* On Off Keying (OOK)
  - Simplest
  - Used in optics
* Phase shift keying (PSK):
  - 

* Quadrature amplitude modulation (QAM): add amplitude variation as well
* Frequency shift keying (FSK)
  - Change frequency
* Another thing that we can change is the shape of data bits. (i.e. pulse shaping)
  - Changes the bandwidth requirements
* Why are there 2 degrees of freedom?
  - Sinusoids property – at most 2 basis functions.
  - Orthogonality:

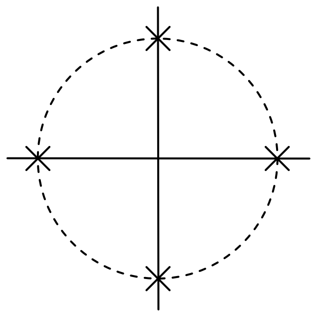
$$\int_0^T x(t)y(t)dt = 0$$

  - Sine is orthogonal to cosine. Orthogonal → receiver can totally separate the combined signals
  - Any waveform sent through the channel can be broken down into the two basis functions.

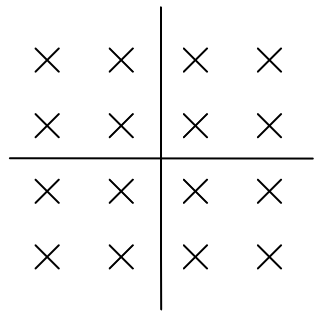
$$\cos(2\pi f t + \theta) = \cos(2\pi f t)\cos(\theta) - \sin(2\pi f t)\sin(\theta)$$

    where $\cos(\theta)$ and $\sin(\theta)$ are constants → constant amplitude difference depending on phase offset

- Orthogonal → can draw in 2D vector space → signal constellation

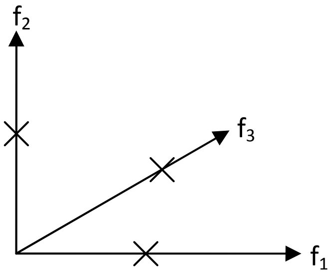


PSK



QAM

Good for wired comms



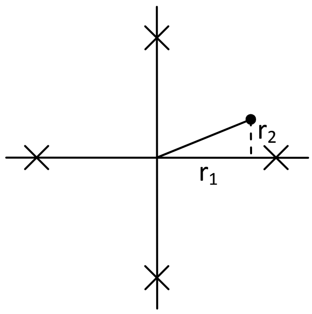
FSK

Good for wireless when amplitude variations can't be handled. E.g. satellite

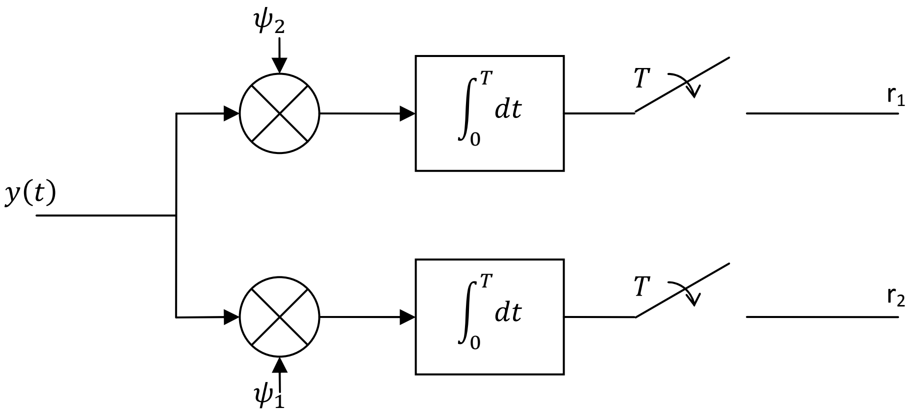
Disadvantage: Bandwidth hungry

Optimal Receiver Design & Error Probability Calculations

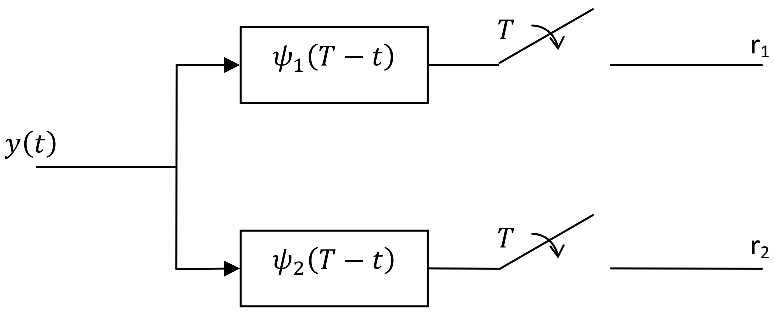
- Demodulation
  - Received vector in signal space:



  - (De) correlator receiver – Project the received signal vector onto the basis functions. i.e. multiply with it and then add up the energy from 0 to T



  - Alternatively, use matched filter receiver (FIR)



    - The filters are modified (delayed) form of the transmit filter.

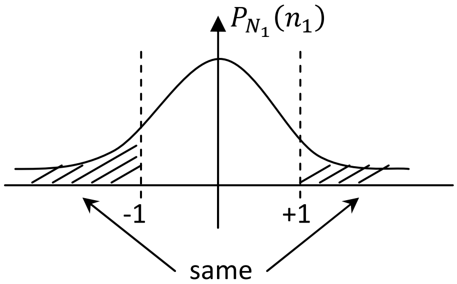
    - Time shift + time reversal
  - Matched filter and correlation type demodulator produces the same output.
  - Matched filter is proven to maximise SNR (and it's easier to prove)
  - But the maths is the same.
  - If the noise is not white (there's memory / correlation), then correlator type doesn't maximise SNR.
- Calculating the probability of error

$$\tilde{r} = \binom{r_1}{r_2} = \binom{x_1}{x_2} + \binom{n_1}{n_2}$$

So

$$P_e = P(n_1 < -1 | x_1 = 1)P(x_1 = 1) + P(n_1 > 1 | x_1 = -1)P(x_1 = -1)$$

- Look at pdf of noise:



Symmetric: $P_e = P(n_1 > 1)$

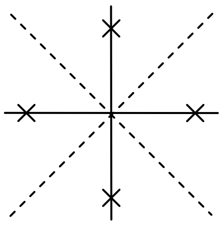
- Q-function:

$$\int_{\psi}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{n^2}{2\sigma^2}\right) dn$$

  - Upper bounds: $Q(x) \leq \frac{1}{2} e^{-\frac{x^2}{2}} \;\; for\ x \geq 0$ $\qquad Q(x) < \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \;\; for\ x \geq 0$
  - Lower bound: $Q(x) < \frac{1}{\sqrt{2\pi}x}\left(1 - \frac{1}{x^2}\right) e^{-\frac{x^2}{2}} \;\; for\ x \geq 0$
  - Used to calculate tail probability for a Gaussian random variable $\mathcal{N}(m, \sigma^2)$:

$$P(X > x) = Q\left(\frac{X - m}{\sigma}\right)$$

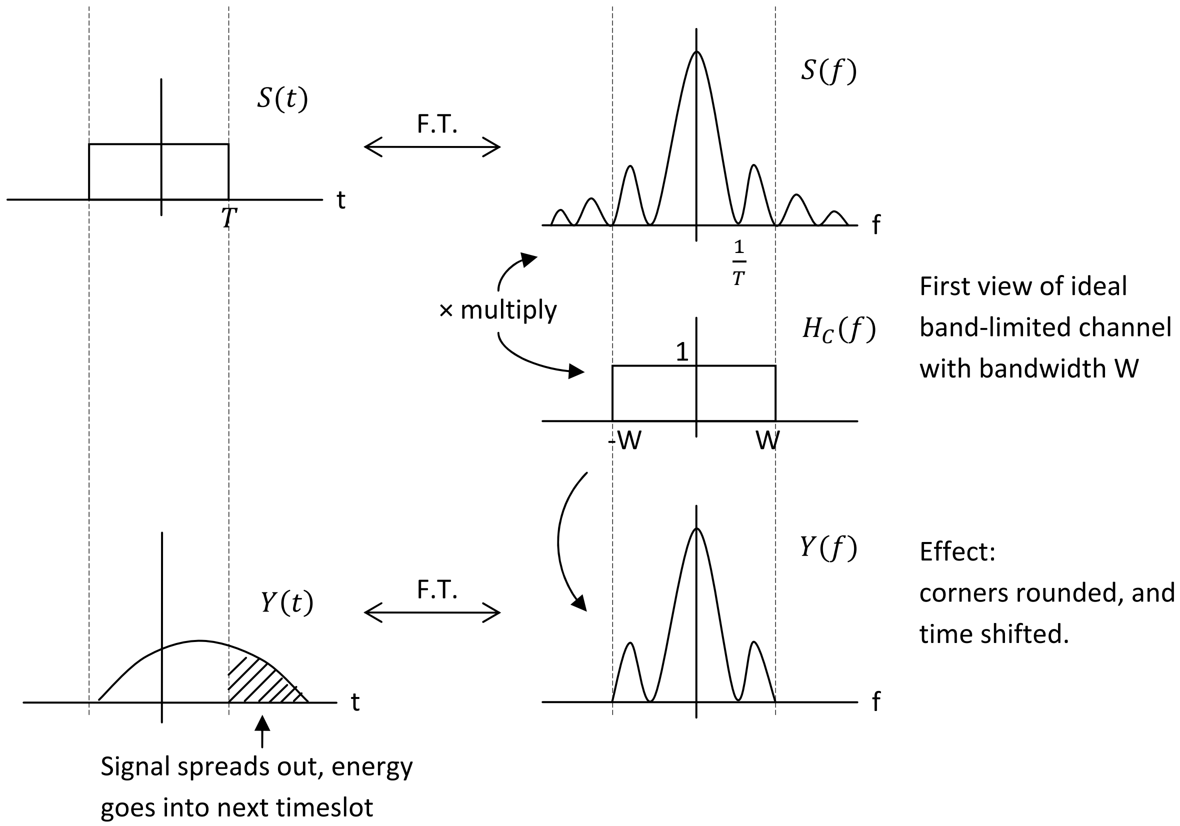
- QPSK: boundaries at ± 45°



- FSK: noise in each frequency is independent → double integral; we need 2 offset dimensions to get error
- All of the above assumes synchronisation between sender and receiver
  - Phase error rotates the received constellation
- Differential phase shift keying (DPSK)
  - Doesn't require exact phase, only takes relative values.
  - Used when it is difficult to get synchronisation between sender and receiver.
  - Disadvantage: taking difference between 2 measurement → 2× noise in received signal
-

Incoherent Systems & ISI

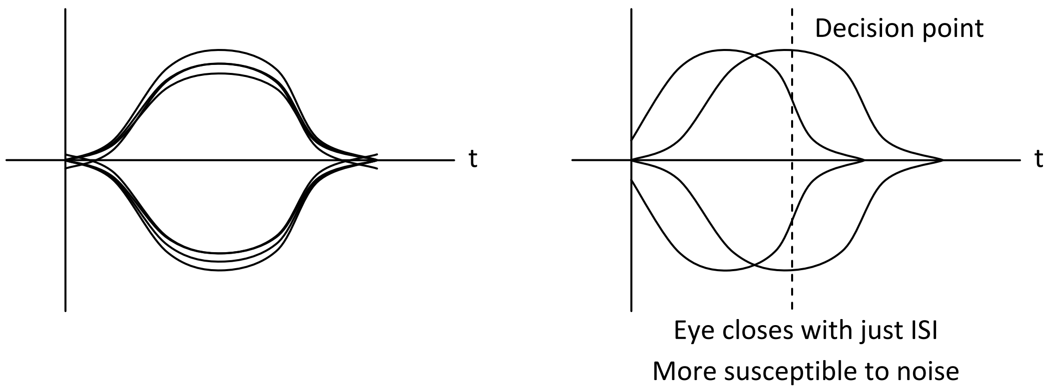
- AWGN Channel: $Y(t) = X(t) + N(t)$
- **ISI Channel** has limited bandwidth – the 'x' component in 'y' is changed.
  - In reality, all channels are band-limited but can be modelled as AWGN if data rates are low.



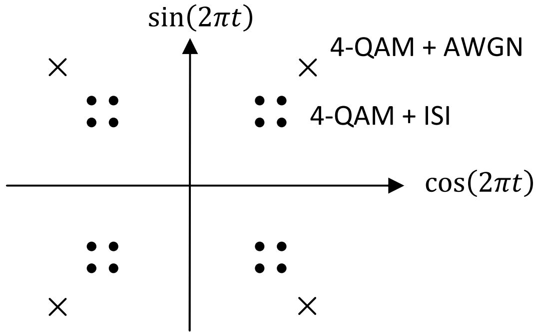
  - To model:

$$y_k = \overbrace{a_0 x_k}^{\text{Direct component}} + \overbrace{a_1 x_{k-1} + a_2 x_{k-2}}^{\text{ISI}} + n_k = \sum_{n=0}^{3} a_n x_{k-n} + n_k$$

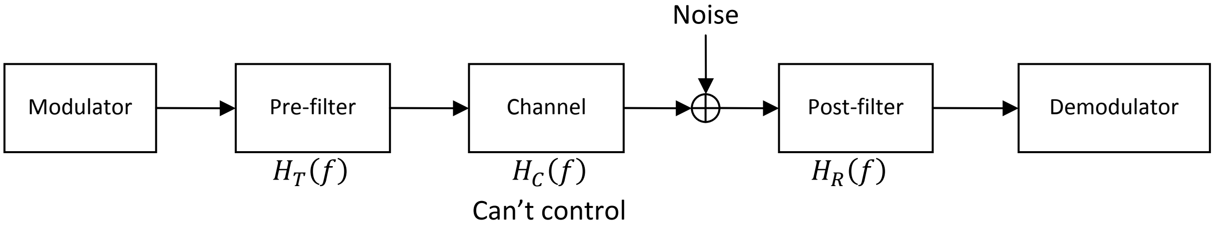
  - Matching filter & decorrelator doesn't work in this case. They're symbol-by-symbol detectors. Need to consider the entire sequence of symbols. → Sequence detection
  - **Eye diagram** – repeated scan of impulse response (different symbols)



- Effect of symbol-by-symbol detection: p_e increases



- Preventing ISI
  - Can't stop channel from spreading out
  - Nyquist sampling theorem: analog → sampled time → analog
    (sampling rate = 2 × highest frequency component) → sample at higher rate
  - Nyquist theorem (ISI): digital → analog → digital → send data at lower rate

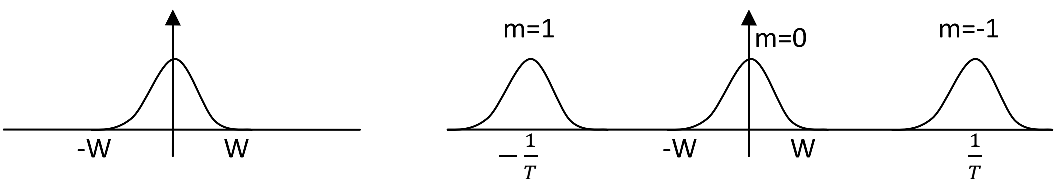


  - So we can represent the channel and filter effects as $X(f) = H_T(f) \times H_C(f) \times H_R(f)$
  - Condition for ISI immunity: We need to set $A(f)$ gives inverse Fourier transform of:

$$a(nT) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

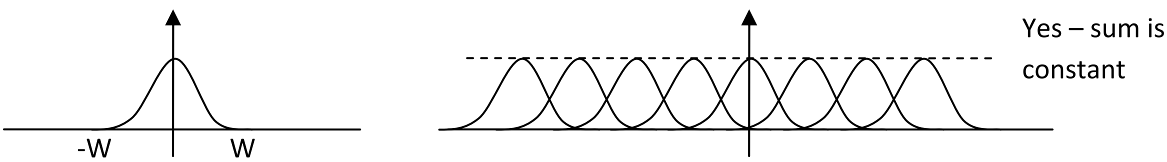
i.e. $$a_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad \text{if} \quad \sum_{m=-\infty}^{\infty} A\left(f + \frac{m}{T}\right) = T \quad \forall f$$

  - 1st design of $H_T(f)$:

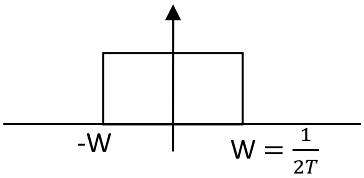


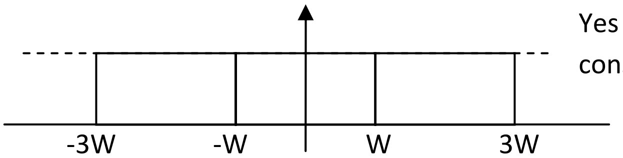
But doesn't satisfy the condition.

  - Need to slow down the rate $\frac{1}{T}$

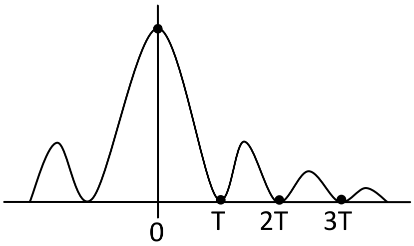


- The highest possible rate while satisfying Nyquist:





Yes – sum is constant

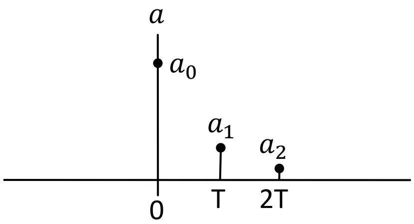


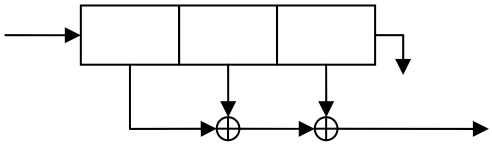
There is ISI in continuous time domain, no ISI after sampling into digital domain

Note: this is a non-causal filter.

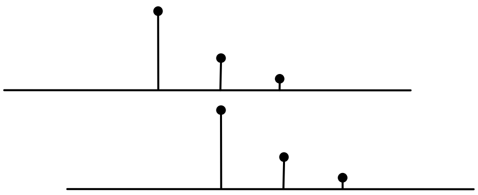
Nyquist & Viterbi

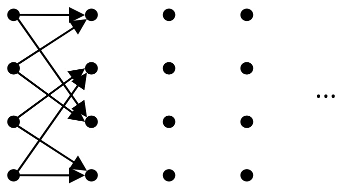
- Sequence detection
  - Just a filter with memory. Like convolutional code, but using real number addition instead of mod-2.



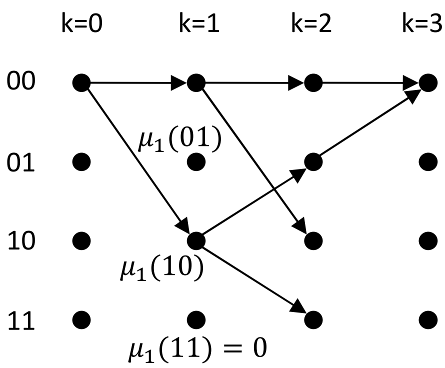


  - But it can also be represented as a trellis with 2^N possible paths





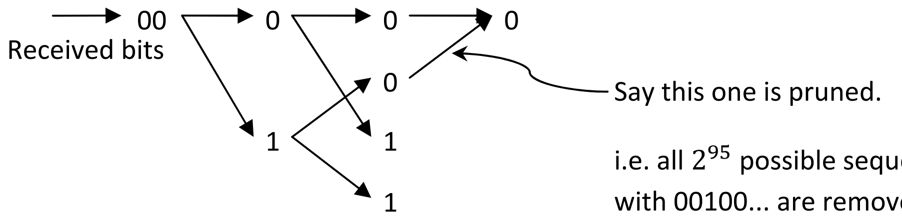
- **Viterbi algorithm**
  - For ISI channel or convolutional decoding
  - For memory length of 100, it is impractical to check all 2^{100} possible combinations of received bits.
  - Relies on the fact that the bits further away in time have less effect than the closer ones.
  - Calculate the path metric (Euclidian distance) between states.
  - Iteratively add path metric to branch metric μ at each k for all states; keeping only the minimum



$$\mu_3(00) = \min \begin{pmatrix} \mu_2(00) + [00 \to 00] \\ \mu_2(01) + [01 \to 00] \end{pmatrix}$$

Pruning – remove less likely paths

- **Complexity reduction** property – removes many possibilities from calculation



i.e. all 2^{95} possible sequences beginning with 00100... are removed.

- **Path merging** property
  - Using the Viterbi algorithm, we don't have to wait until all 100 bits has been received before deciding on the most correct output
  - Reduces receive delay
  - Trace back all surviving paths in the trellis at any time, and we will get to a point where an intermediate state is the same. E.g. for k=5, all surviving paths trace back to state (00) at k=1
    - Then we can be certain that at k=1, the most correct state must be (00).
    - And thus we know all states prior to that.
  - Path merging length ≈ 5 × channel length (number of taps in filter)
    - E.g. for channel length 3, we look back 15 and there is a good chance we can determine the correct path for received sequence so far.

Fading Channels

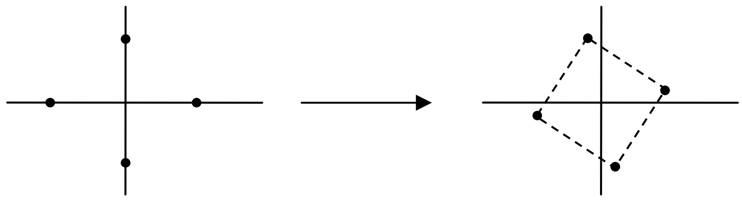
- In a wireless channel even without channel memory, there may still be ISI due to multiple paths of propagation. Each path has different delays so the signal spreads out in time, like channel memory.
  - For ISI and wireless channel:

$$y_k = a_0 x_k + a_1 x_{k-1} + \cdots + n_k$$

- Wireless channels have an added complexity: as receiver moves, the paths vary with time. $(a_1, a_2, \ldots)$
  - Continuous time representation

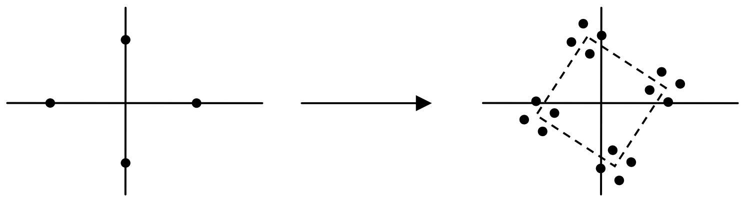
$$A\cos(2\pi f_c t) \to A\sum_n \alpha_n(t)\cos(2\pi f_c[t-\tau_n(t)]) = \sum_n \alpha_n(t) e^{-j2\pi f_c \tau_n(t)}$$

    - $a_n(t)$ – attenuation
    - $\tau_n(t)$ – delay
    - $e^{\tau_n(t)}$ – rotation
  - For a 1-path, not time-varying channel:



Scaled and shifted

  - For 2 paths:



- Back of the envelope calculation for 900MHz GSM: $\lambda = \frac{1}{3}m$. even at walking speed, the phase seems to change by full revolutions quite quickly. But when we compare this speed with the bit rate with which data is sent, it's actually not that fast.
- To track phase variation and perform correction, need training bits (known) as well as real data.
  - E.g. in GSM: 116 data bits, 26 training bits, 3-6 bits header and footer.
  - Training data is sent in the middle of the packet rather at the start/end to give maximum effect, but it means processing can't be started until the training data is received.

Header	58 bits data	26 bits training	58 bits data	Footer

GSM Packet

- Probability of error in a fading channel

$$P_e = Q\left[\sqrt{\frac{2E_0}{N_0}}\right]$$

  - Averaging over density function of the channel:

$$E[P_e] = \int Q\left[\sqrt{\frac{2\varepsilon_b a}{N_0}}\right] P_A(a)da$$

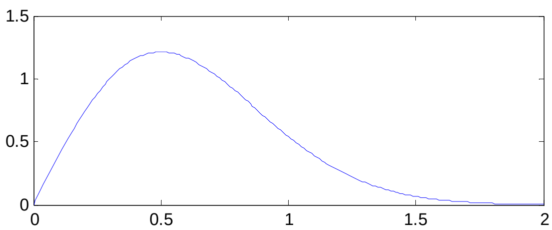
    - a – time varying attenuation
    - $P_A(a)$ – channel pdf
- PDF of channel
  - Continuum of reflection paths
  - Large number of random reflected paths

$$\sum \alpha_n e^{-j2\pi\tau_n(t)}$$

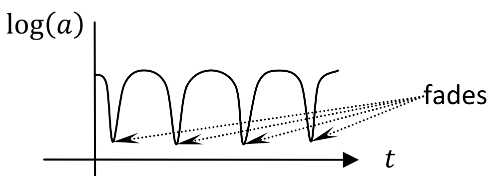
  - Central limit theorem → Gaussian distribution in both cosine and sine dimension

$$f(x,y) = \boldsymbol{J} \times f(r,\theta)$$

    - $\boldsymbol{J}$ – Jacobian matrix
    - r – radii in Rayleigh distribution

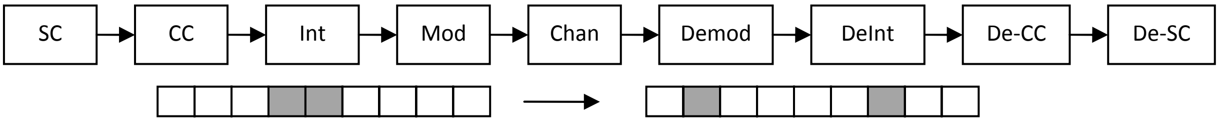


    - θ – angle in uniform distribution from 0 to 2π
    - Gaussian distribution in both x and y gives Rayleigh amplitude and uniform phase.
- In a fading channel, the magnitude changes with time:



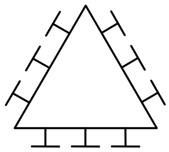
- Overcoming fading
  - Three fundamental resources in a signal: power, bandwidth and frequency.
  - Can increase block length such that fading occurs only in a small part of a block. Using error correction codes (ECC), 0 error is possible as long as block is long enough. Standard convolutional coding works if using long blocks.

- Can apply **interleaving** to spread out the burst of error. Spreading out the errors makes it easier to be handled by Viterbi.



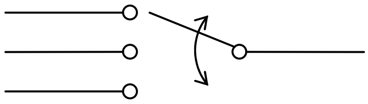
Errors Errors spread out

- **Channel diversity**: use spatial resources. Use multiple antennas, multiple time slots, or multiple channels with different frequencies to reduce the probability of all incoming signals being bad.



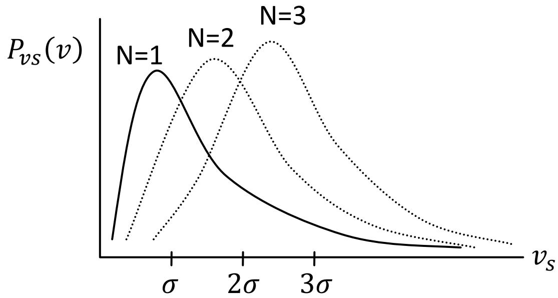
GSM uses sectorised antennas

- **Receive diversity** – switching or combination
  - Signal can be combined anywhere on the receive chain
  - Combining audio streams isn't the best option
  - Combine at the output of the channel
- **Switching diversity**

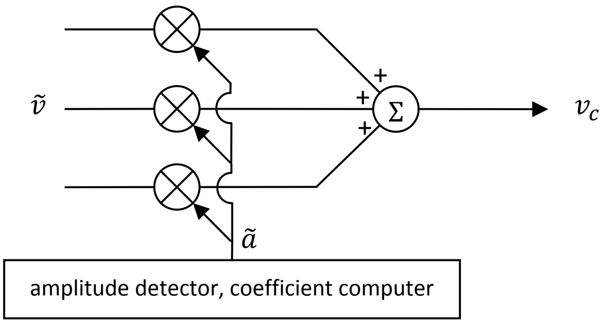
Received signals  Select the strongest signal

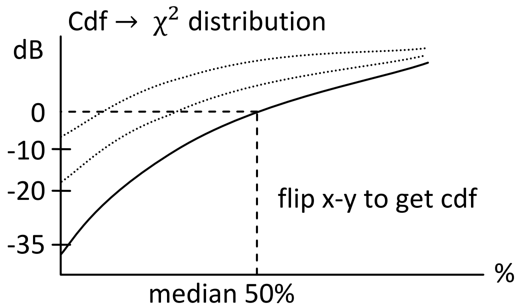
  - Cdf: $P_{vs}(v) = P_r(v_s < v) = P_r(v_1 < v, \dots, v_N < v) = \left[1 - \exp\left(-\frac{\frac{1}{2}v^2}{\sigma^2}\right)\right]^N$

    Assume channels have same variance, and they are all independent.
  - Differentiating gives pdf:



  - Simple to implement because only need input gain control.
- **Combination diversity**



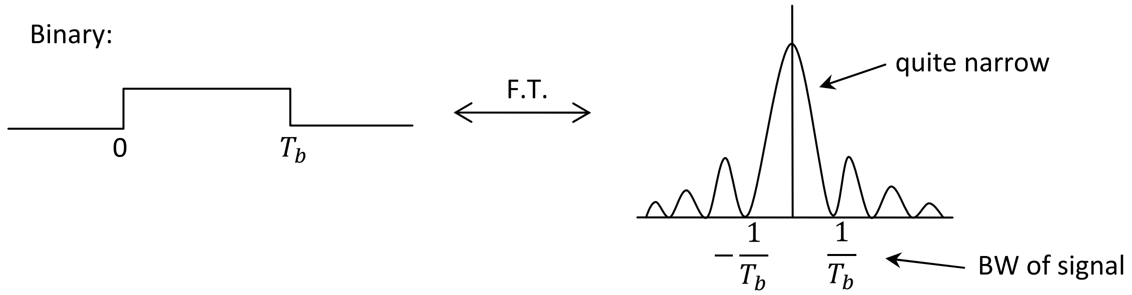


  - Takes a weighted sum of inputs as the signal
  - Outage = % time with errors such that signal is unusable (below a certain SNR)
  - To use the graph, first read the dB value for acceptable SNR, then read % outage.
- Comparing: combination diversity is better with more antennas, but more complex.

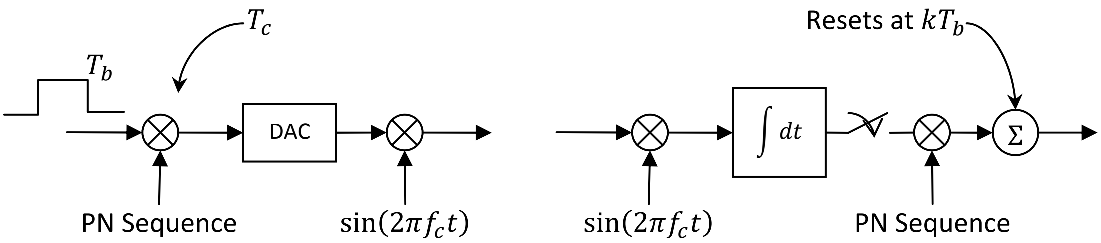
Spread Spectrum

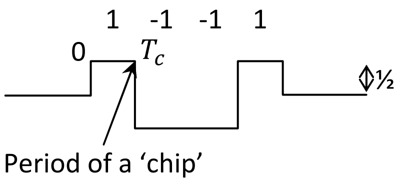
- Can also be used to overcome fading, but has other uses as well.

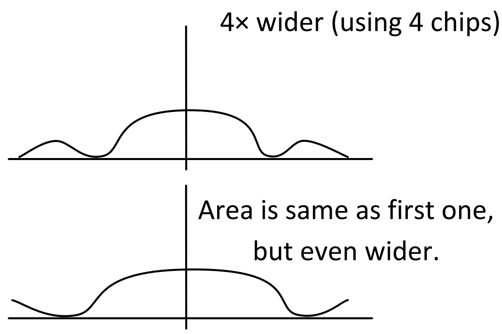
Binary:



  - One way to use more bandwidth than the minimum is to use different carriers – **frequency diversity**
  - Another way is to have the bits change more frequently – apply a spreading code.
- Direct sequence spread spectrum (DSSS)
  - Directly multiply a bit with a spreading code sequence of +1 or -1
  - Random noise / Pseudo noise (PN) as input to modulator

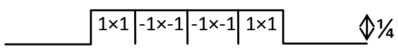






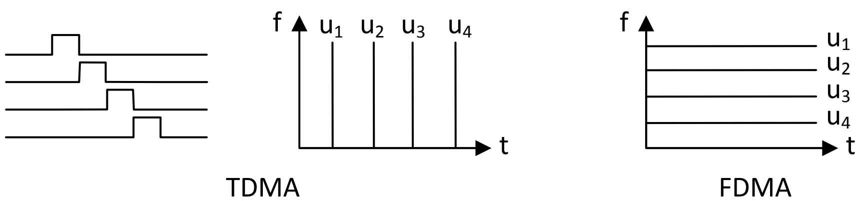
Transmitted signal:

Just use matched filter at the receiver to decode spreading sequence.



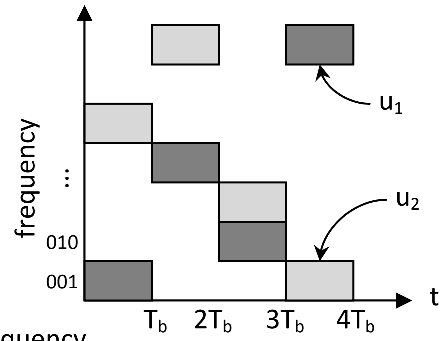
The only difference is scaling.

  - Looking at the SNR of the original input in band-limited channel, it seems pretty good. But since DSSS uses a wider bandwidth, a larger bandwidth of noise is gathered too, so SNR is not as good.
  - This has military/security applications: the transmitted signal is not visible if it is below noise floor. As long as the synchronisation is correct with the same spreading sequence it is possible to recover the signal from below noise floor.
  - Another advantage is code orthogonality → code division multiple access (CDMA)
  - TDMA is a special case of DSSS:



- CDMA phones don't store all possible codes, but the basestation tells it which class of codes to use
  - In real applications, codes used are not exactly orthogonal, but pseudo random instead.

  - So there is some interference with each other, but it is small in general (especially for longer codes)
  - Another effect: spreads out narrowband interference. Spreads out the bandwidth of unwanted signals during the decoding stage.
  - Also suppresses spectrum nulls due to fading
- Frequency hopping (FH)
  - Another way of getting spread spectrum and diversity.



This can be orthogonal: if another user uses a different code. Usually, there is more bandwidth than users, and hope that the pseudo random codes do not collide.

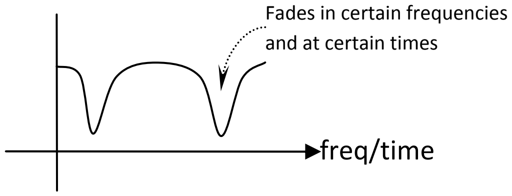
Frequency bits code:	1	4	2	7
PN code:	001	100	010	111

The code actually consists of numbers.

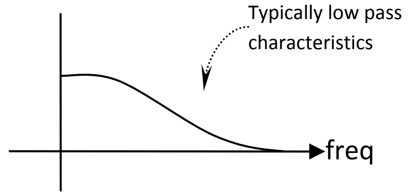
FH can totally avoid a narrowband noise in the channel simply by assigning a code that avoids it.

  - FH = interference avoidance
  - DSSS = interference averaging.
  - We can make a hybrid system, where a DSSS spectrum hops. This is used in WCDMA.
  - Similarities: PN sequence; uses more bandwidth than necessary, ‘hides’ the transmission, multi-user
  - Differences: FH can use multiple bands while DSSS can only use a single band.
- Wireless channel:

  Wireless channel gain:



  Wireline channel gain:

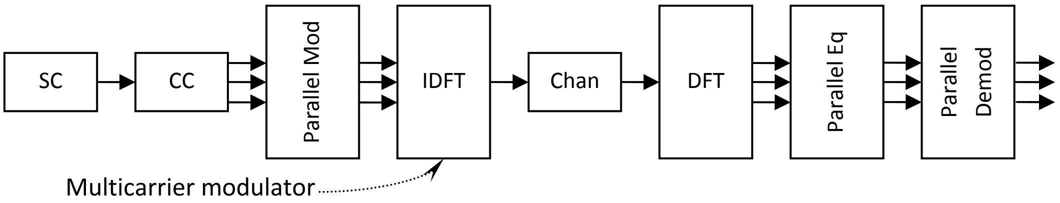


  - In DS, a time-shifted signal due to a different path can cause interference. But, we can design the sequence so that it is both orthogonal to other users and also to a slightly time shifted version of itself. → Combat multipath interference
  - PN sequences are almost orthogonal to time shifts! → low multipath interference.
  - RAKE receiver: parallel receive chains, offset by intervals of T_c; → collect energy from all of the multiple paths.
- GSM and IS-95
  - Issues:
    - Companies want to re-use existing equipment as much as possible
    - Governments have different radio regulations
  - Early tests (Paris 1986)
    - 9 systems tested
    - Field tests and lab simulation tests
  - Criteria: spectrum efficiency, subjective voice quality, mobile cost, hand portable feasibility, base station cost, expansion for new services, coexisting with current systems

- Winner (1989):
  - Source coder (speech encoder): Residual pulse excited RPE-LPC 13kHz 22.8kbps
  - Channel coder: CC(2,1,5) encoding – convolutional code with rate = ½, length k = 5
  - Modulation: GMSK with BT = 0.3

ADSL and OFDM

- Electronics has allowed sending information at a faster rate and with broader bandwidth. But it is still preferable to send in narrow chunks to handle channel characteristics. Just send more in parallel.
- Orthogonal Frequency Division Multiplexing (OFDM)



  - A buffer can be used to convert serial data into parallel data.
  - Uses many orthogonal subchannels, each with its own modulation scheme
  - Able to cope with severe channel conditions
  - Turns out the IDFT can be used to convert the data in multiple frequencies into real valued sequence:

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X'_k e^{j2\pi nk/N} \qquad n = 0,1,\dots,N-1$$

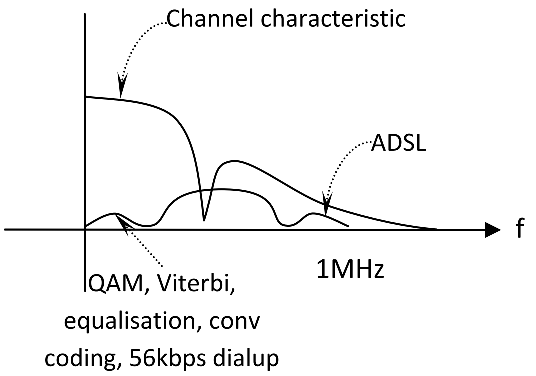
  - $\{x_n, 0 \le n \le N-1\}$ corresponds to samples of multicarrier OFDM signal $x(t)$ with K subcarriers:

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X'_k e^{j2\pi kT/N} \qquad t \in [0,T]$$

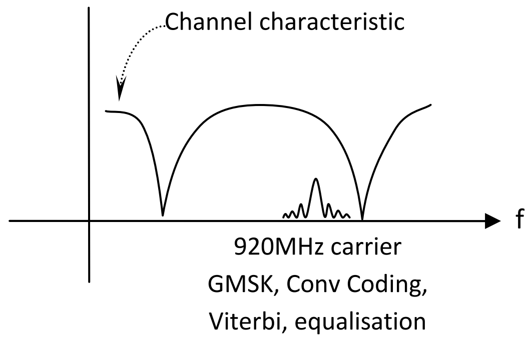
    - T is the signal duration

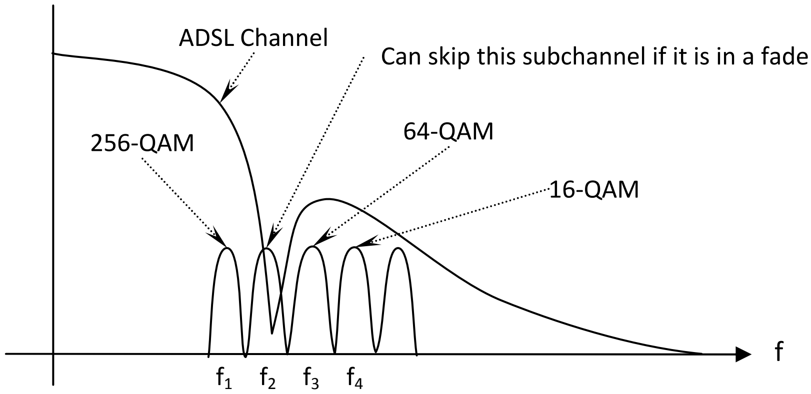
- ADSL has 512 subchannels, uses 1024 samples with 32 samples of guard interval.

Wireline, ADSL

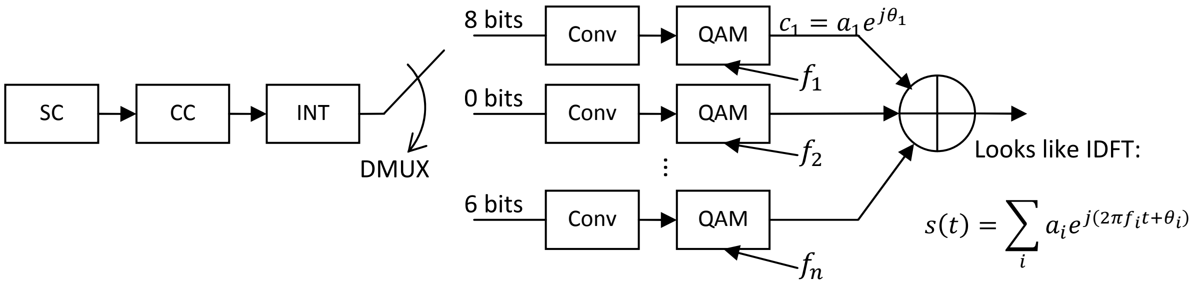


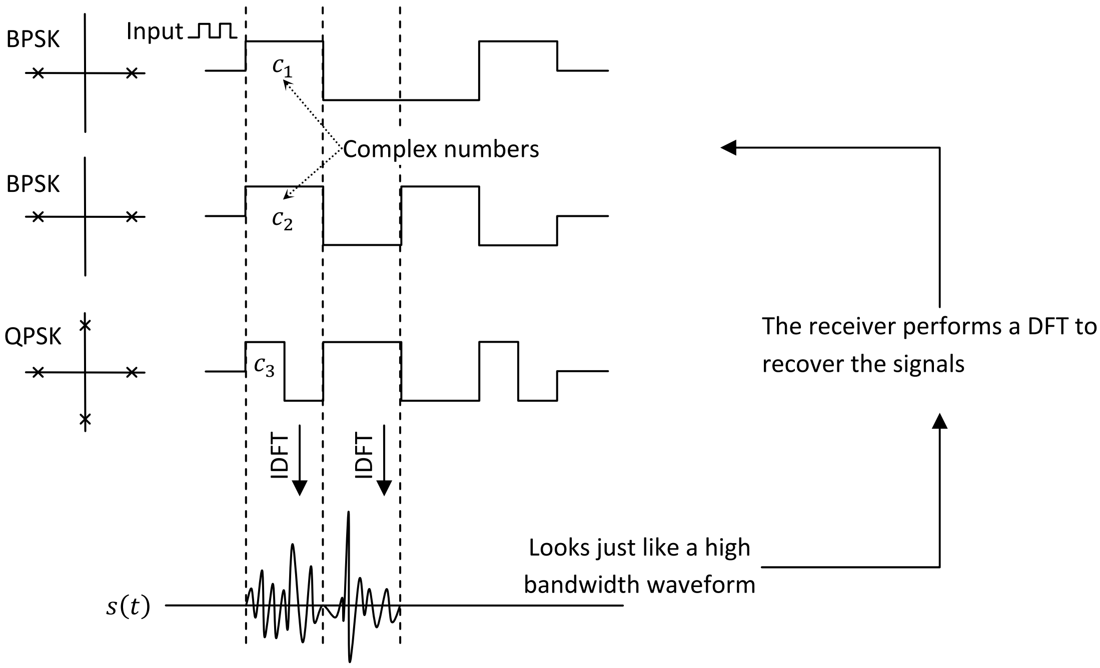
Wireless



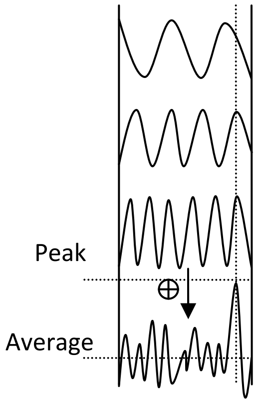


- Bit loading onto multiple frequencies





- Peak-to-average ratio (PAR)

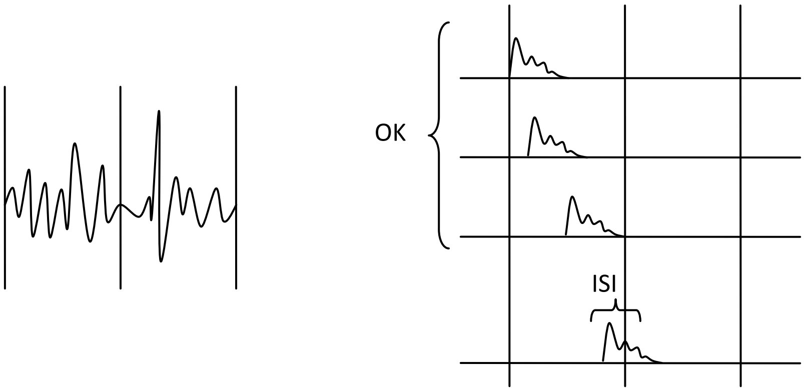


Abnormally big spike or trough is possible when all components add up constructively. It becomes a problem when there is limited dynamic range in the amplifier.

<u>Need to handle</u> this situation: it can cause increased quantisation errors and clipping for signals with high PAR.

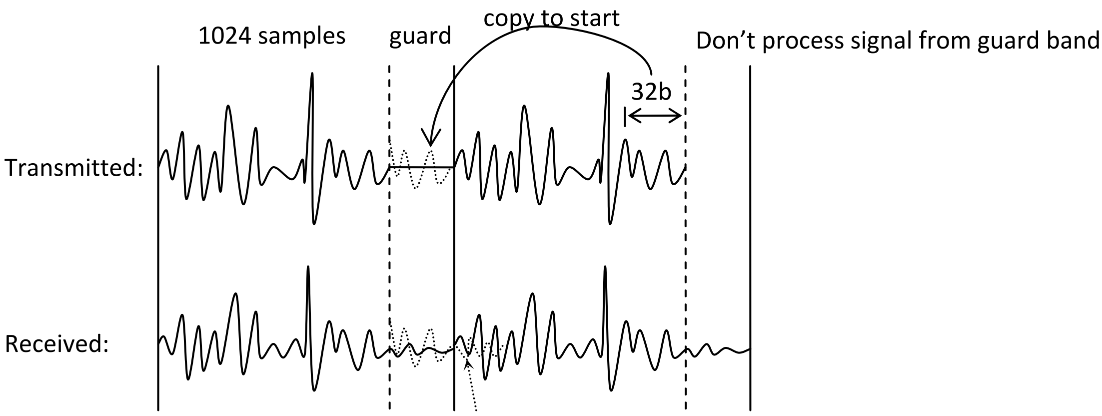
  - Strategy: can apply a phase shift to some component signals so that the peak doesn't happen at that particular time. In this case, we need to reserve a bit in the data stream to indicate this rotation.

- ISI in OFDM signal



  - Interference between samples within the same symbol is not so much of a problem, because each sample belongs to the same combination of complex numbers after receiver DFT.
  - But there is still ISI towards the end, when energy from one symbol spreads into the next.
  - To handle ISI by avoiding the problem, introduce a guard interval with length ≥ impulse response.

- Cyclic prefix
  - Instead of waiting and doing nothing in the guard interval, send a copy of the first 32 samples in the guard interval's 32 samples. (Sending zeros can cause synchronisation problems anyway)
  - E.g. in ADSL's DMT signals



  - With cyclic prefix, get energy from the guard interval smoothed into the signal fed into DFT
  - So that every sample in the time of interest has the same smoothing effect.
  - This energy would otherwise be lost.
  - Doesn't affect calculations, because it comes from the same symbol.