Contents

1 Week 8 Module 6:

1.1 Digital Communication Systems

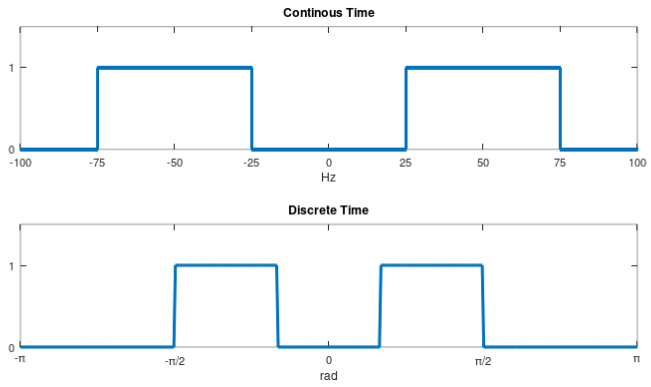
1.1.1 Introduction to digital communications

1. The success factors for digital communications
   (a) Power of the DSP paradigmw
      - integers are easy to **regnerate**
      - good phase control
      - adaptive algorithms
   (b) Algorithmic nature of DSP is a perfect match with information therory:
      - Image Coding: JPEG's entropy coding
      - Encoding of of accustic or video informatiion: CD's and DVD's error correction
      - Communication Systems: trellis-coded modulation and Vierbi coding
   (c) Hardware advancement
      - minituarization
      - general-purpose platforms
      - power efficiency
2. The analog channel constraints
   - unescapable "limits" of physical channels:
     – **Bandwith:** the signal that canbe send over a channel has a limited frequency band

  - **Power:** the signal has limited power over this band, e.g. due to power limit of the equipment
- Both constraints will affect the final capacity of the channel.
- The maximum amound of information that can be reliably delivered over a channel - bits per second -
- Bandwidth vs. capacity:
  - small sampling period $T_s \Rightarrow$ high capacity
  - but the bandwidth signal grows as $\frac{1}{T_s} \Rightarrow \Omega_N = \frac{\pi}{T_s}$

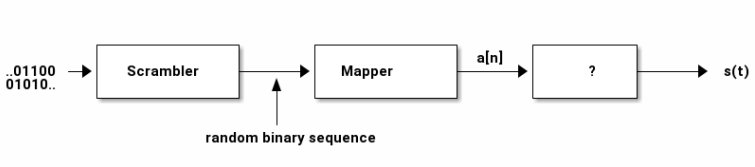
3. The design Problem

- We are going to adapt the all-digital paradigm
- Converting the specs to digital design



- with:
  - Sampling Frequency $F_S \geq 2f_{max}$
  - Continuous Time $F_s/2$: Nyquist frequency
  - Maximum Frequency: $\frac{F_s}{2} \Rightarrow \pi$
  - Bandwidth: $\omega_{min,max} 2\pi \frac{f_{min,max}}{F_s}$
- Transmitter design
  - convert a bitstream into a sequence of symbols a[n] via a mapper
  - model a[n] as white random sequence $\Rightarrow$ add a scrambler

- no we need to convert a[n] into a continuous-time signal within the constraints



If we assume that the data is randomized and therefore the symbol sequence is a white sequence, we know that the power spectral density is simply equal to the variance. And so the power of the signal will be constant over the entire frequency band. But we actually need to fit it into the small band here as specified by the bandwidth constraint. So how do we do this? Well, in order to do that, we need to introduce a new technique called upsampling, and we will see this in the next module.

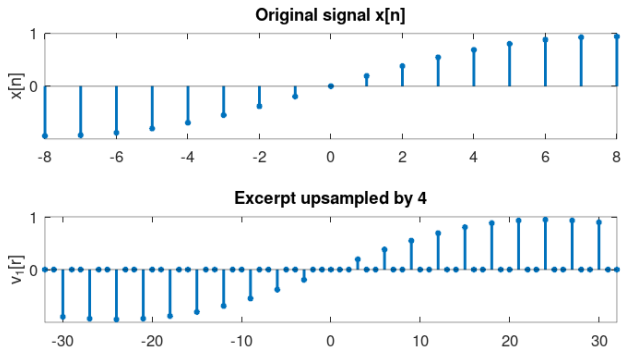
We are talking about digital communication systems and in this lesson we will talk about how to fulfill the bandwidth constraint. The way we are going to do this is by introducing an operation called upsampling and we will see how upsampling allows us to fit the spectrum generated by the transmitter onto the band allowed for by the channel.

1.1.2 Controlling the bandwidth

- Shaping the bandwidth Remember that our assumption is that the signal generated by the transmitter is a wide sequence and therefore its power spectral density will be constant and full band. In order to meet the bandwith constraint, we need to shrink the support of the power spectral density.
  - the answer is multirate techniques

1. Upsampling
   - Our Problem
   - bandwith constraint requires us to control the spectral support of a signal
     - we need to be able to shrink the support of a full-band signal

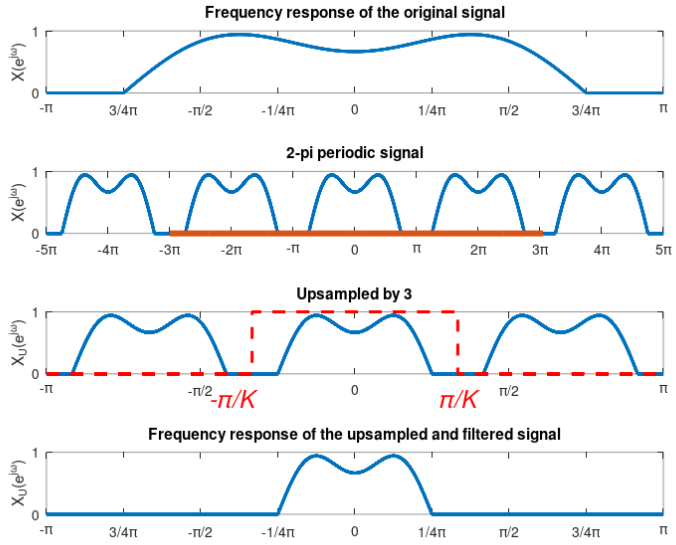
- Upsampling can be obtained by interpolating a discrete time sequence to get a continuous time signal. And resample the signal with a sampling period which is k-times smaller than the original interpolation sample.
- Or we do it entirly digitally.
  (a) we need to "increase" the number of samples by k
  (b) obviously $x_U[m] = x[n] when m multiple of K$
  (c) for lack of better strategy, put zeros elsewhere
- Upsampling in the time domain



- Upsampling in the digital domain: Frequency Domain

$$\begin{aligned} X_U(e^{j\omega}) &= \sum_{m=-\infty}^{\infty} x_U[m]e^{-j\omega m} \text{ with } x_U = 0 \text{ if } m \neq nK \\ &= \sum_{m=-\infty}^{\infty} x[n]e^{-j\omega nK} \\ &= X(e^{j\omega K}) \end{aligned}$$

This is simply a scaling of the frequency axis by a factor of K. Graphical interpretation: since we are multiplying the frequency axis by a factor of K, there will be a shrinkage of the frequency axis.



- $\frac{\pi}{K}$: Filter Cut-Off Frequency
- The bandwidth of the signal was shrinked by factor K=3: from $\frac{3}{4}\pi$ to $\frac{1}{4}\pi$
- Upsampling in the digital domain: Time Domain
  (a) insert K-1 zeros after every sample
  (b) ideal lowpass filtering with $\omega_c => frac\pi K$

$$\begin{aligned} x^{\lfloor}[n] &= x_U(n) * sinc(n/K) \\ &= x_U[i] sinc\left(\frac{n-i}{K}\right) \\ &= x[m] sinc\left(\frac{n}{K} - m\right), \text{ with } i = mK \end{aligned}$$

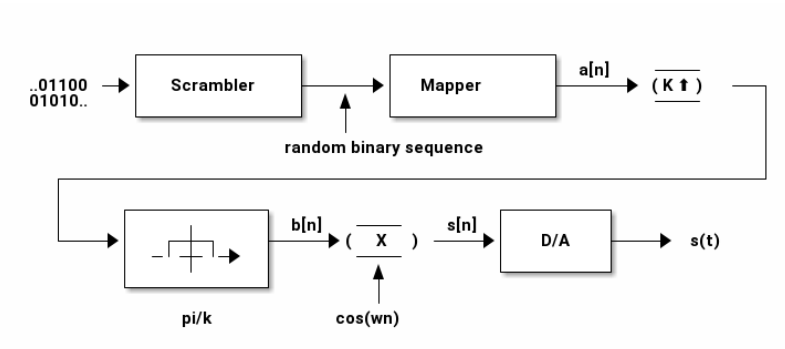
Which is exactely the same formula when using an interpolator and a sampler.

2. Fitting the transmitter spectrum The bandwidth constraint says that only frequencies between F_{min} and F_{max} are allowed. To translate it to the digital domain, follow the preceeding steps:

- let $W = F_{max} - F_{min}$
- pick F_s so that:
  - $F_s > 2F_{max}$
  - $F_s = KW$, $k \in \mathbb{N}$
- $\omega_{max} - \omega_{min} = 2\pi \dfrac{W}{F_s} = \dfrac{2\pi}{K}$
- we can simply upsample by K

Bandwith constrainth

And so we can simply upsample the sample sequence by K, so that its bandwidth will move from 2pi to 2pi/K, and therefore, its width will fit on the band allowed for by the channel.



Scrambler	Randomizes the data
Mapper	Segments the bit-stream into consecutive groups of M bits. And this bits select one of 2^M possible signaling values. The set of all possible signaling values is called the alphabet.
a[n]	The actual discrete-time signal. The sequence of symbols to be transmitted.
K	The uppsampler narrows the spectral occupancy of the symbol sequence. The following low pass filter is known as the shaper, since it determines the time domain shape of the transmitted symbols.

b[n] The baseband signal. Produced

s[n] The passand signal. $s[n] = Re\{c[n]\} = Re\{b[n]e^{j\omega_c n}\}$ The signal which is fed to the D/A converter is simply the real part of the complex bandpass signal. With $\omega_c = \frac{\omega_m ax - \omega_m in}{2}$

Data Rates

- up-sampling does not change the data rate
- we produce (and transmitt) W symbols per seconds
- W is sometimes called the Baud Rate of the system and is equal to the available bandwidth.

Raised Cosine

1.1.3 Controlling the power

1. Noise and probability of error

   - Transmitter reliability
     - transmitter sends a sequence of symbols a[n]
     - receiver obtains a sequence $\hat{a}[n]$
     - even if no distortion we can't avoid noise: $\hat{a}[n] = a[n] + \eta[n]$
     - when noise is large, we make an error
   - Probability of error depends on:
     - power of the noise with respect to the power of the signal
     - decoding strategy
     - alphabet of transmission symbols

   (a) Signaling alphabets

   - we have a (randomized) bitstream coming in
   - we want to send some up-sampled and interpolated samples over the channel
   - how do we get from bit-stream to samples: How does the mapper works
   - mapper:
     - split incoming bitstream into chunks
     - assign a symbol a[n] from a finite alphabet A to each chunk.

- slicer:
  - receive a value $\hat{a}$[n]
  - decide which symbol from A is "closest" to $\hat{a}[n]$

(b) Example: two-level signaling

- mapper:
  - split incoming bitstream into **single bits**
  - a[n] = G of bit is 1, a[n] = -G if bit is 0
- slicer:

$$n-th \text{ bit} = \begin{cases} 1, \text{ if } \hat{a}[n] > 0 \\ 0, \text{ otherwise} \end{cases}$$

- Porbability Error

$$\begin{aligned} P_{err} &= P\left[\eta[n] < -G|\text{ n-th bit is 1}\right] + P\left[\eta[n] > G|\text{ n-th bit is 0}\right] \\ &= \left(P\left[\eta[n] < -G\right] + P\left[\eta[n] > G\right]\right)/2 \\ &= P\left[\eta[n] > G\right] \\ &= \int_G^\infty \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}} d\tau, \text{ with the PDF for the Gaussian Distribution with the kno} \\ &= erfc(G/\sigma_0), \text{ Numerical Packages: The Error Function} \end{aligned}$$

Probability Error

Is some function of the ratio between the amplitude of the signal and the standard deviation of the noise.

- transmitted power

$$\begin{aligned} \sigma^2 &= G^2 P\left[\text{ n-th bit is 1}\right] + G^2 P\left[\text{ n-th bit is 0}\right] \\ &= G^2 \end{aligned}$$

Probability of Error

$$P_{err} = erfc(\sigma_s/\sigma_0) = erfc(sqrtSNR)$$

And since we are in a log log scale, we can see that the probability of error decays exponentially with the signal to noise ratio.

Absolute rate of decay might change, in terms of the linear constants involved in the curve. The trend will stay the same, even for more complex signaling schemes

(c) Lesson learned:

- to reduce the probability of error increase G
- increasing G increases the power
- we can't go above the channel's power constraint.

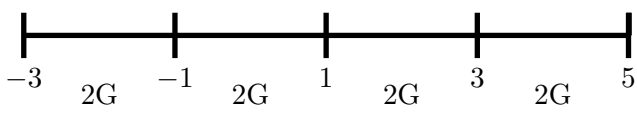
2. Multilevel signaling

- binary signaling is not very efficient (one bit at a time)
- to increase the throughput we can use multilevel signaling

the general idea We take now larger chunks of bits, and therefore, we have alphabets, that have a higher cardinality. So more values in the alphabet, means more bits per symbol, and therefore a higher data rate. But not to give the ending away, we will see that the power of the signal will also be dependent of the size of the alphabet, and so in order not to exceed the certain probability of error, given the channel's power of constraint. We will not be able to grow the alphabet indefinitely, but we can be smart in the way we build this alphabet. And so we will look at some examples.

(a) Pulse Ampltitude Modulation PAM

- mapper:
  - split incoming bitstream into chunks of M bits
  - chunks define a sequence of integers k[n] $\in \{0,1,2..2^{M}\text{-}1\}$
  - a[n] = G((-2^{M} +1) + 2k[n]) odd integers around zero
    - with M=2 and G=1: a[n]=-3.-1, 1, 3
- slicer:
  - $a'[n] = argmin[|\hat{a}[n] - a|]$
- PAM with M=2, G=1

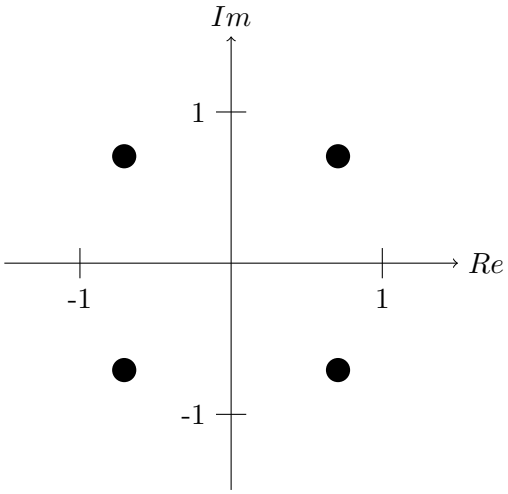


- distance between points is 2G
- using odd integers creates a zero-mean sequence
- probability error analysis for PAM is analog the lines of binary signaling
- can we increase the throughput of PAM even further
- here's a wild idea, let's use complex numbers

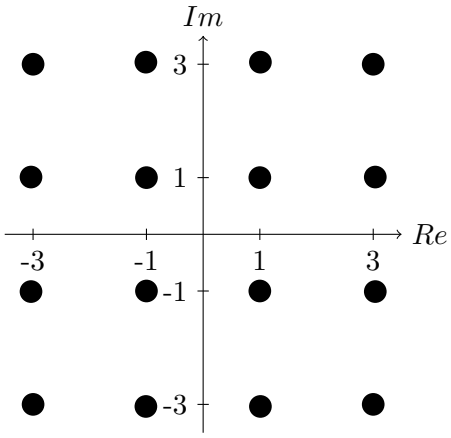
(b) Quadratur Amplitude Modulation QAM

- mapper:
  - split incoming bitstream into chunks of M bits, M even
  - use M/2 bits to define a PAM sequence $a_r[n]$
  - use the remaining M/2 bits to define an independent PAM sequence $a_i[n]$
  - a[n] = G(a_r[n] Aj$_i$[n])
- slicer:
  - $a'[n] = arg min[$|\hat{a}[n] - a|$]

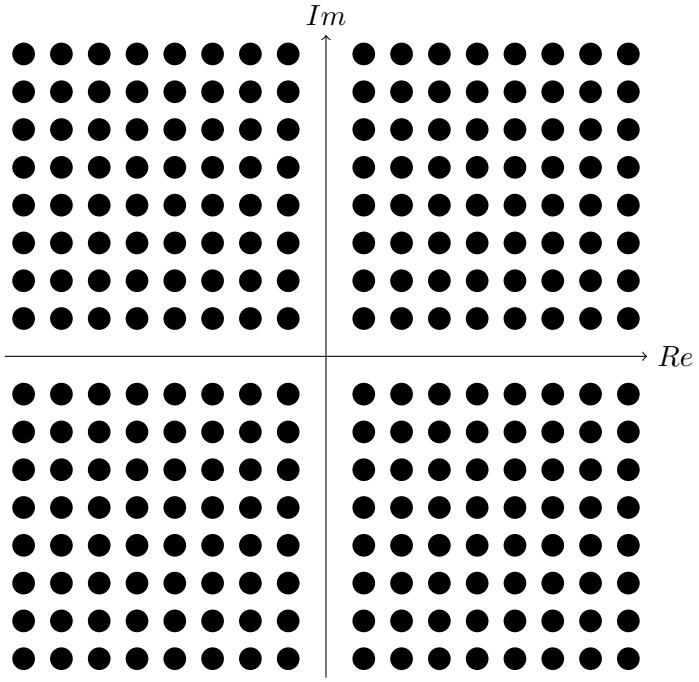
So the transition alphabet a, is given by points in the complex plane with odd valued coordinates around the origins. The receiver deslicer works by finding the symbol in the alphabet which is closest in Euclidian distance to the relieved symbol.



- QAM, M=2, G=1
- QAM, M=4, G=1



- QAM, M=8, G=1



2 Week 8 Module 7:

2.1 Image Processing