DSP Coursera IV 11.8.24

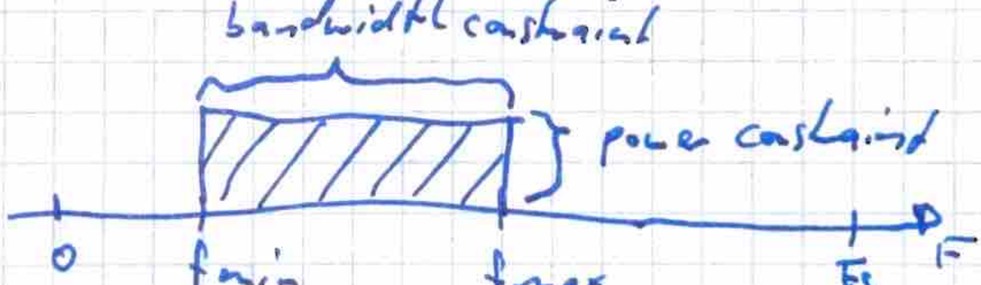
Ch. 12 Design of a Digital Communication System

12.2 Modem Design: The Transmitter

- Physical Medium: Analog
- Modern communication systems place all processing into digital domain
  - ↳ only interface D/A converter

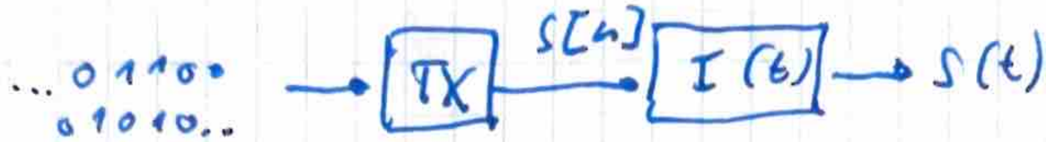
12.2.1. Digital Modulation and the Bandwidth constraint

- Bandwidth constraint imposed by channel



- D/A converter: $F_s \geq 2 \cdot f_{max}$

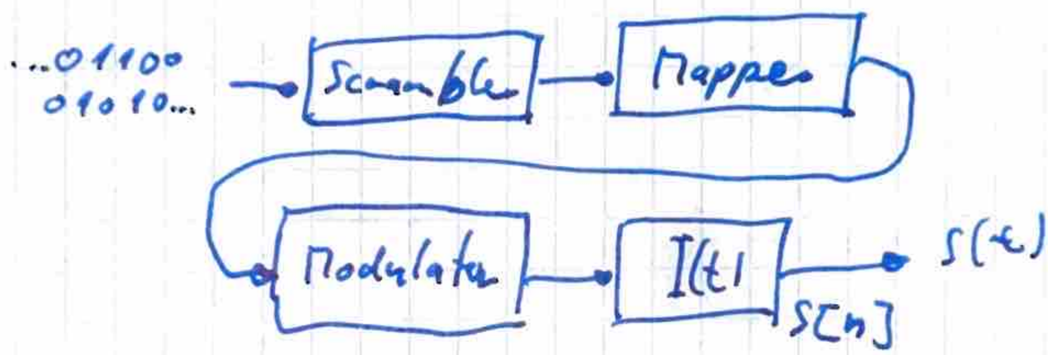
· Digital transmitter (abstract view):



- choose interpolation freq (see below)
  ↳ bandwidth requirement:

$$\omega_{min,max} = 2\pi \frac{f_{min,max}}{F_S}$$

· input side of the transmitter



· build suitable s[n]

· Scrambler: randomize data
  · pseudo random
  · needs to be undone by receiver (algorithmically)

· Mapper: · multilevel signaling
  · efficient utilization of available bandwidth
  · bitstream segmented into M bit chunks
  · chunks (sets) select one of 2^M possible values (can be complex)
  ↳ alphabet

DSP Coursera IV 11.8.24

- Spectral Properties of the Symbol Sequence
  - mapper produces sequence $a[n]$
    ↳ the ~~distr~~ discrete-time signal that needs to be transmitted
  - if initial binary bitstream is a maximum information sequence (distribution of zeros and ones is equal 50%-50%)
  - and scrambler randomizing input bitstream

↳ sequence $a[n]$: stochastic i.i.d process distributed over the alphabet

⇒ power spectral density of $a[n]$:

$$P_A(e^{j\omega}) = \sigma_A^2$$

where σ_A depends on the design of the alphabet and on its distribution.

12.8.24

- Choice of Interpolation Rate
  - determine suitable rate F_S for final interpolator
  - signal $a[n]$:
    - baseband, fullband
    - centered around zero
    - power spectral density: non-zero over entire $[-\Omega, \Omega]$ interval

↳ interpolated $a[n] \Rightarrow$ analog signal (at F_S)

- nonzero spectral power over entire $[-F_S/2, F_S/2]$ interval (including DC level)

- to fulfill bandwidth constraint, produce signal with
  - bandwidth $\omega_w = \omega_{max} - \omega_{min}$
  - centered at $\omega_c = \pm(\omega_{max} + \omega_{min})/2$

↳ trick: upsample and interpolate $a[n]$ to narrow spectral support

DSP Coursera IV 12.8.24

- ideal discrete-time interpolators
  - upsampling by factor 2
    ↳ half-band signal
  - upsampling by factor 3
    ↳ signal with support of 1/3 of total band
  - general case: upsampling factor K

$$\frac{2\pi}{K} \leq \omega_w$$

  - maximum efficiency:
    - occupy entire bandwidth with signal
  - $\Rightarrow K = 2\pi / \omega_w$
    - analog bandwidth requirements

$$K = \frac{F_s}{f_w}$$

where $f_w = f_{max} - f_{min}$ (effective bandwidth of signal)

- K must be integer:

$$\begin{cases} F_s \geq 2 f_{max} \\ F_s = K \cdot f_w \quad K \in \mathbb{N} \end{cases}$$

↳ optimal signaling

- The Baseband Signal
  - upsampling by K:
    - narrow spectral occupacy of symbol sequence
    - prescribed bandwidth needs to be fulfilled
  - lowpass filter the upsampled signal
    - remove the multiple copies of upsampled spectrum
  - lowpass filter is called „shaper"
    - determines the time-domain shape of transmitted symbols
  - ideally use a sinc filter
    - not possible (ideal filter)
  - alternative:
    - baseband signal $b[n]$:

$$b[n] = \sum_m a_{KU}[m]\, g[n-m]$$

- where: $a_{KU}[n]$: upsampled symbol sequence
  - $g[n]$: impulse response of the lowpass filter

DSP Coursera IV 12.8.24

- since $a_{K\uparrow}[n] = 0$ for n not multiple of K

↳ $b[n] = \sum_i a[i]\, g[n - iK]$

- at multiples of K the upsampled sequence $b[n]$ takes on the exact symbol value

↳ $b[mK] = a[m]$

⇒ requirements for lowpass filter

$$g[mK] = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases}$$

↳ classical interpolation property

- Realizable filters: minimum frequency support of $G(e^{j\omega})$ cannot be smaller than $[-\pi/K, \pi/K]$

↳ allways a (controllable) frequency-leakage outside of prescribed band

- Need to use a FIR lowpass filter
  - sinc approximations (FIR) are very poor

- raised cosine:
  - much „friendlier" lowpassfilter
  - raised cosine with nominal bandwidth ω_w (and nominal cutoff $\omega_b = \frac{\omega_w}{2}$)

$$G(e^{j\omega}) = \begin{cases} 1 & \text{if } 0 < \omega < (1-\beta)\omega_b \\ 0 & \text{if } (1+\beta)\omega_b < \omega < \pi \\ \frac{1}{2} + \frac{1}{2}\cos\left(\pi \frac{\omega - (1-\beta)\omega_b}{2\beta\omega_b}\right) & \text{if } (1-\beta)\omega_b < \omega < (1+\beta)\omega_b \end{cases}$$

  - defined over positive frequency axis
  - symmetric around origin
  - β with $0 < \beta < 1$ exactly defines leakage as percentage of Passband
  - β closer to $1 \rightarrow$ sharper magnitude response
  - still an ideal filter
  - impulse response decays as $1/n^3$
  - good FIR approximations
  - number of needed taps increases as β approaches one
    - rarely exceeds 50

DSP Coursera IV | 12.8.24

- The Bandpass Signal
  - filtered signal $b[n] = g[n] * a_{KU}[n]$
    ↳ baseband signal with bandwidth ω_w
  - modulate $b[n]$ to shift into required frequency band
  - sinusoid carrier:
    $$c[n] = b[n]\, e^{j\omega_c n}$$
    $c[n]$: complex bandpass signal
  - modulation frequency: center-band freq.
    $$\omega_c = \frac{\omega_{min} + \omega_{max}}{2}$$
  - spectral support of modulated signal: positive interval $[\omega_{min}, \omega_{max}]$
    ↳ analytical signal: complex signal with one-sided spectral occupancy e.g. $[\omega_{min}, \omega_{max}]$
  - Signal fed to D/A: real part
    $$s[n] = \mathrm{Re}\{c[n]\}$$
  - baseband signal $b[n]$ complex:
    ↳ bandpass signal is combination of sine and cosine modulation

- Baud Rate vs. Bit Rate
  - Baud Rate: number of symbols which can be transmitted in one second
  - interpolator at F_s
  - exactly K samples per symbol (upsampling)

  ↳ baudrate: $B = \frac{F_s}{K} = f_w$

  - general rule: baudrate is always smaller or equal to positive passband of channel

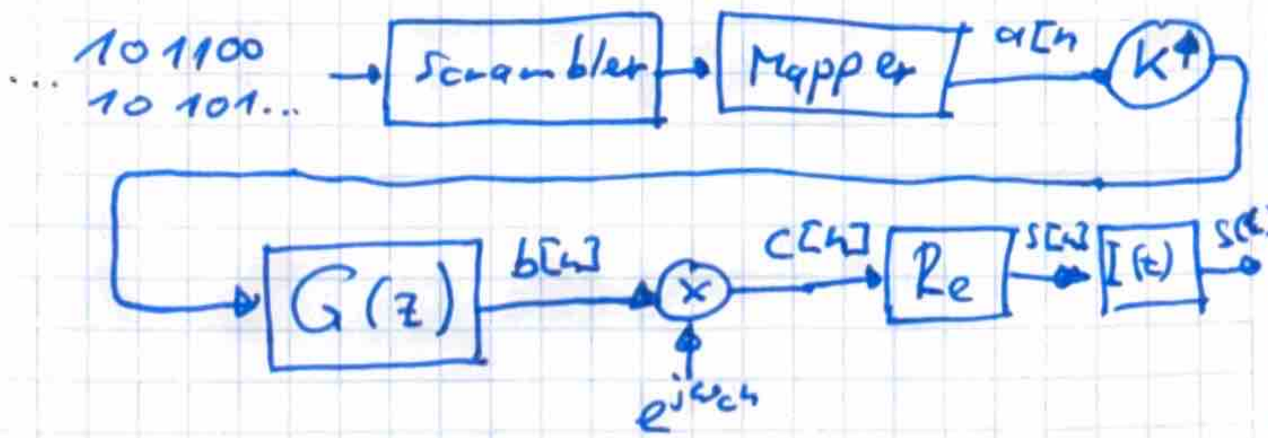
  - effective bandwidth f_w depends on modulation scheme and on frequency leakage of shaper

  - total bit rate of transmission system
    - at most baud rate times $\log_2$ of number of symbols in alphabeth
    - for mapper operating on M bits per symbol:

$$R = M \cdot B$$

DSP Coursera IV 12.8.24

· Complete digital transmitter



DSP Coursera IV 13.8.24

12.2.2. Signaling Alphabets and the Power Constraint

- Mapper: associate each group of M bits a value α from given alphabet $\mathcal{A}$.

- Mapper includes multiplicative factor (gain) G_0 → set final gain of generated signal

- symbol sequence:

$$a[n] = G_0 \alpha[n], \quad \alpha[n] \in \mathcal{A}$$

- Transmitted power
  - binary sequence: i.i.d., uniformly distributed input sequence
    → each group of M bits is equally probable
  - memory-less mapper: no dependency between symbols introduced
  - mapper: source of random process $a[n]$ which is also i.i.d.

- power of output sequence:

$$\sigma_a^2 = E|a[n]|^2$$
$$= G_0^2 \sum_{\alpha \in \mathcal{A}} |\alpha|^2 p_a(\alpha)$$
$$= G_0^2 \sigma_\alpha^2$$

where $p_a(\alpha)$ probability assigned to symbol $\alpha \in \mathcal{A}$ (by mapper)

- distribution over alphabet $\mathcal{A}$ is one design parameter of the mapper
  - not necessarily uniform

- variance σ_α^2: intrinsic power of the alphabet. Depends on:
  - alphabet size (increases exponentially with M)
  - alphabet structure
  - probability distribution of symbols in alphabet
- to avoid wasting transmission energy:
  - mapper is balanced (no DC component)

$$E[\alpha[n]] = \sum_{\alpha \in \mathcal{A}} \alpha \, p_a(\alpha) = 0$$

DSP Coursera IV 13.8.24

- power of signal after upsampling and modulation

$$\sigma_s^2 = \frac{1}{\pi} \int_{\omega_{min}}^{\omega_{max}} \frac{1}{2} |G(e^{j\omega})|^2 G_0^2 \sigma_\alpha^2$$

- shaper is designed such that overall energy over passband:

$$G^2 = 2\pi$$

$$\hookrightarrow \sigma_s^2 = G_0^2 \sigma_\alpha^2$$

- respect power constraint
  - → choose value for G_0
  - → design alphabet $\mathcal{A}$

so that

$$\sigma_s^2 \leq P_{max}$$

where P_{max} : maximum power allowed to be transmitted over channel

- goal: maximize the reliable troughput
  - but: conflicting priorities:
    - σ_α^2
    - G_0
  - if we boost transmitted bitrate ($R = MB$)

    by increasing $M \Rightarrow \sigma_\alpha^2$ grows

    → we must reduce Gain G_0
    - but we reduce reliability of transmission

DSP Coursera IV 13.8.24

- PAM (pulse amplitude modulation
  - simplest mapping strategies
  - correspondence between binary values and signal values
  - symbol sequence is uniformly distributed with $p_a(\alpha) = 2^{-M}$ for all $\alpha \in A$
  - e.g. we can assign to each group of M bits $(b_0, ..., b_{M-1})$ the signed binary number $b_0 b_1 b_2 ... b_{M-1}$ which is a value between -2^{M-1} and 2^{M-1} (b_0 is sign bit)
  - amplitude of each transmitted symbol determined by binary input values
  - PAM alphabet is balanced (no DC component)
  - power of mappers output:
    $$\sigma_\alpha^2 = \sum_{\alpha = 1}^{2^{M-1}} 2^{-M} \alpha^2 = \frac{2^M(2^M+3)+2}{12}$$
  - PAM signal is baseband (with bandwidth ω_0) before modulation
    - total spectral support: $2\omega_0$
  - ↳ after modulation spectral support doubles
    - redundancy of modulated signal ⇒ underutilization of available bandwidth

- QAM (quadrature amplitude modulation)
  - symbols in alphabet are complex values
    ↳ 2 real values are transmitted simultaneously at each symbol interval

- complex symbol sequence

$$a[n] = G_0 (a_I[n] + j a_Q[n])$$
$$= a_I[n] + j a_Q[n]$$

- Shaper: real-value filter
  ↳ $b[n] = (a_{I,ku} * g[n]) + (a_{Q,ku} * g[n])$

$$= b_I[n] + j b_Q[n]$$

$$\Rightarrow s[n] = Re\{b[n] e^{j\omega_c n}\}$$
$$= b_I[n] \cos(\omega_c n) - b_Q[n] \sin(\omega_c n)$$

- QAM signal is linear combination of two pulse amplitude modulated (PAM) signals
  - cosine carrier modulated by real part
  - sine carrier modulated by imag. part

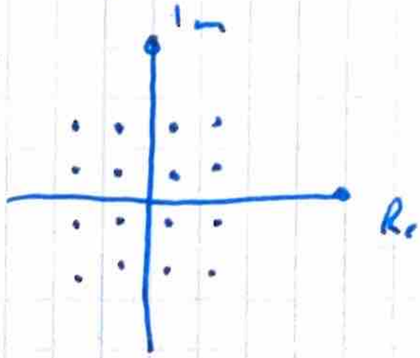
  of symbol sequence

DSP Coursera IV 13.8.24

- QAM: linear combination of two PAM signals
  - the two signals (their carrier) are orthogonal to each other
  - ↳ $\underbrace{b_I[n]}_{\text{cosine carrier (in-phase)}}$ and $\underbrace{b_Q[n]}_{\text{sine carrier (quadrature/orthogonal)}}$ can be separated at the receiver via subspace projection operation

- complex symbols: abstraction that simplifies notation (and calculation)

- Constellations
  - the 2^M symbols of the alphabet can be represented as points on the complex plane
  - geometric arrangement of the points: signaling constelation



  - simplest constellations: upright square lattices with points on odd integers
  - for M even: the 2^M constellation points α_{hk} form a square shape with $2^{M/2}$ points per side

$$\alpha_{hk} = (2h-1)+j(2k-1), \quad -2^{M/2-1} < h,k < 2^{M/2-1}$$

  - square constellations: called regular
  - nominal power of regular, uniformly distributed constellation:

$$\sigma_\alpha^2 = 4\sum_{h=1}^{2^{M/2-1}}\sum_{k=1}^{2^{M/2-1}} 2^{-M}\left[(2h-1)^2+(2k-1)^2\right]$$

$$= \frac{2}{3}\left(2^M-1\right)$$

  - constellations can be defined on other lattices, either irregular or regular

DSP Coursera IV 14.8.24

- Transmission Reliability
  - receiver eliminates all "fixable" distortions (this is an assumption here)
  - "almost exact" copy of sent symbol sequence available for decoding
  - $\hat{a}[n]$: "almost exact" symbol seq.
  - additive noise introduced by channel

    $\hat{a}[n] = a[n] + \eta[n]$

    where $\eta[n]$: complex white Gaussian noise
  - complex representation of noise:
    - convenient abstraction
    - simplifies math. analysis of decoding process
  - real-valued zero-mean Gaussian noise introduced by channel:
    - variance σ_0^2
    - transformed by receiver into complex Gaussian noise
    - real and imaginary parts are independent zero-mean Gaussian variables with variance $\sigma_0^2/2$

- each complex noise sample $\eta[n]$ is distributed

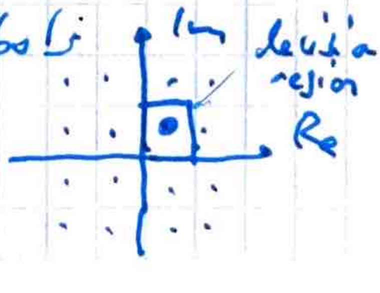
$$f_\eta(z) = \frac{1}{\pi \sigma_0^2} e^{\frac{-|z|^2}{\sigma_0^2}}$$

- magnitude of noise samples introduce shifts in the complex plane for demodulated symbols $\hat{a}[n]$
  - if shift is too big → decoding error

- probability of error (bound)
  - QAM decoding techniques: hard slicing
  - associate $\hat{a}[n]$ with most probable symbol $\alpha \in \mathcal{A}$
    - ↳ choose alphabet symbol at minimum Euclidean distance

$$\mathcal{D}\{\hat{a}[n]\} = \underset{\alpha \in \mathcal{A}}{\arg\min} \left\{ |\hat{a}[n] - G_0 \alpha|^2 \right\}$$

- hard slicer partitions complex plane into decision regions centred around alphabet symbols



DSP Coursera IV 14.P.24

- when error sample $\eta[n]$ moves received symbol outside of the right decision region
  ↳ decoding error
- happens for square-lattice constellation when either
  - real or
  - imaginary

  noise part of ~~error~~ sample is larger than min. distance between ~~samples~~ symbols and the decision region boundary
- probability of error:

$$p_e = 1 - P\left[\left(Re\{\eta[n]\} < G_0 \right) \wedge \left(Im\{\eta[n]\} < G_0 \right) \right]$$

$$= 1 - \int_D f_\eta(z) dz$$

where f_η : pdf of additive complex noise

D square on complex plane centered at origin and $2d_{min}$ wide

- closed form for probability of error:
  - approximate decision region D by circle of radius d_{min}

$$P_e = 1 - \int_{|z|<a_0} f_\eta(z)\,dz$$

$$= 1 - \int_0^{2\pi} d\theta \int_0^{a_0} \frac{\rho}{\pi\sigma_0^2} e^{\frac{-\rho^2}{\sigma_0^2}}$$

where $z = \rho e^{j\theta}$

- the probability error decreases exponentially with the gain (and with the power of the transmitted)

DSP Coursera IV 14.8.24

- concept of "reliability"
  - quantified by probability of tolerable error
  - probability of error cannot be zero
  - can be made arbitrarily low
    - reference: $p_e = 10^{-6}$
- Assume: transmission at max. permissible power
  ↳ SNR is maximized
  - $SNR = \frac{\sigma_s^2}{\sigma_0^2} = G_0^2 \frac{\sigma_\alpha^2}{\sigma_0^2}$
  ↳ $SNR \approx -\ln(p_e)\,\sigma_\alpha^2$
- For regular square-lattice constellation
  - maximum number of bits per symbol

$$M = \log_2\left(1 - \frac{3}{2} \frac{SNR}{\ln(p_e)}\right)$$

↳ this is how the power constraint affects the max. achievable bitrate.

- upper bound of achievable bitrate on channel
  - Shannon's capacity formula

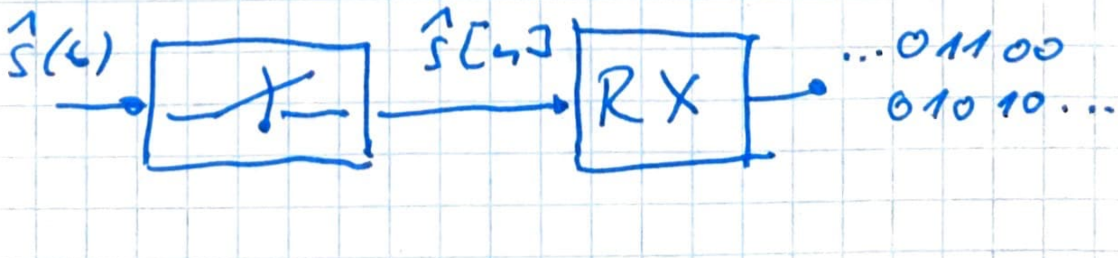
$$C = B \log_2\left(1 + \frac{S}{N}\right)$$

where: C: absolute max capacity in bits per second
- B: available bandwidth in Hz
- S/N: signal to noise ratio

DSP Coursera IV 14.8.24

12.3. Modem Design: The Receiver

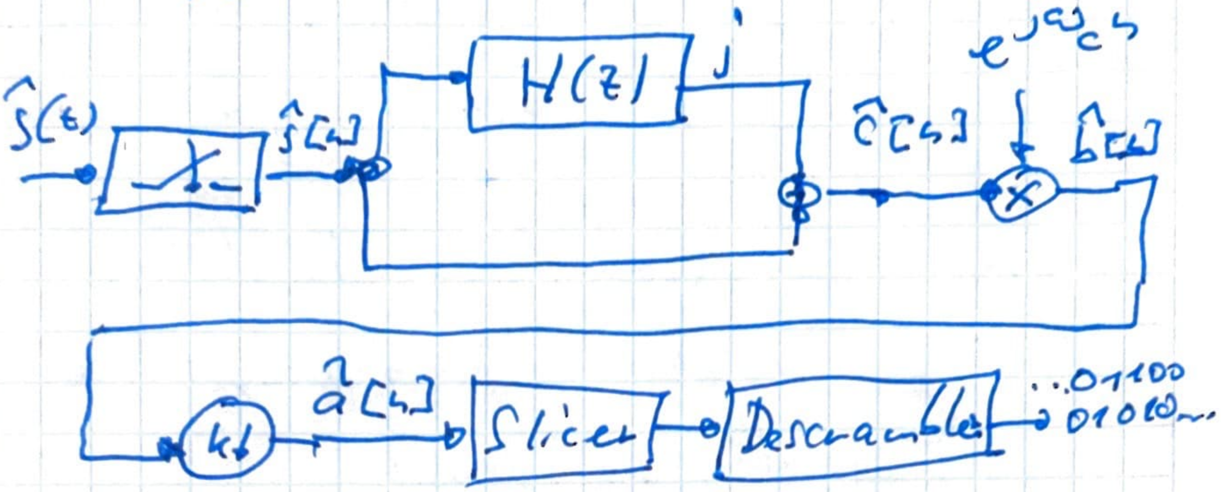
- Transmitter sends signal $s(t)$ over channel
- Receiver receives distorted and noise-corrupted signal $\hat{s}(t)$
- Input interface of receiver is A/D converter operating at same frequency F_s as transmitter's D/A converter
- Receiver tries to undo impairments introduced by channel and demodulate signal
  - ↳ output: binary sequence which is identical to the one injected into transmitter (if no decoding errors occur)



15.8.24

12.3.1. Hilbert demodulation

- neglect effects of channel
  - ↳ $\hat{s}(t) = s(t)$ and after A/D $\hat{s}[n] = s[n]$



- first operation:
  - retrieve complex bandpass signal $\hat{c}[n]$ from the real signal $\hat{s}[n]$
- efficient way:
  - original signal $c[n]$ is analytical signal
    - ↳ imaginary part is completely determined by real part
  - for complex analytical signal $x[n]$
    - $X(e^{j\omega}) = 0$ over $[-\pi, 0]$, 2π periodicity
    - split $x[n]$ into real and imaginary signal

$$x[n] = x_r[n] + j x_i[n]$$

↳ $$x_r[n] = \frac{x[n] + x^*[n]}{2}$$

$$x_i[n] = \frac{x[n] - x^*[n]}{2j}$$

DSP Coursera IV 16.8.24

- frequency domain:

$$X_r(e^{j\omega}) = \frac{X(e^{j\omega}) + X^*(e^{-j\omega})}{2}$$

$$X_i(e^{j\omega}) = \frac{X(e^{j\omega}) - X^*(e^{-j\omega})}{2j}$$

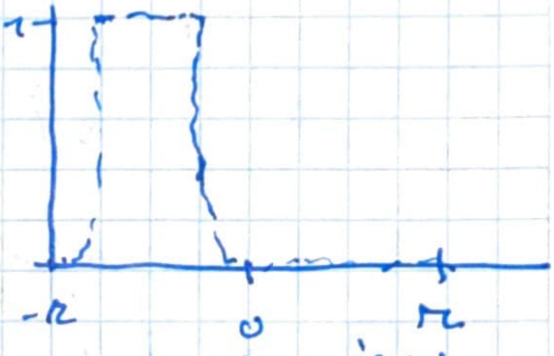
- since $x[n]$ is analytic (by definition):

$X(e^{j\omega}) = 0$ for $-\pi \leq \omega < 0$

$X^*(e^{-j\omega}) = 0$ for $0 < \omega \leq \pi$

$X(e^{j\omega})$ does not overlap with $X^*(e^{-j\omega})$

$|X(e^{j\omega})|$



$|X^*(e^{-j\omega})|$

$$X(e^{j\omega}) = \begin{cases} 2X_r(e^{j\omega}) & \text{for } 0 \leq \omega \leq \pi \\ 0 & \text{for } -\pi < \omega < 0 \end{cases}$$

$x_r[n]$ is real $\Rightarrow$ Fourier transform is complex conjugate symmetric

$X_r(e^{j\omega}) = X^*(e^{-j\omega})$

$$X^*(e^{-j\omega}) = \begin{cases} 0 & \text{for } 0 \leq \omega \leq \pi \\ 2X_r(e^{j\omega}) & \text{for } -\pi < \omega < 0 \end{cases}$$

$\Rightarrow$ finally:

$$X_i(e^{j\omega}) = \begin{cases} -j\,X_r(e^{j\omega}) & \text{for } 0 \le \omega \le \pi \\ +j\,X_r(e^{j\omega}) & \text{for } -\pi < \omega < 0 \end{cases}$$

↳ product of $X_r(e^{j\omega})$ with frequency response of a Hilbert filter

- time domain:
  - imaginary part can be retrieved from real part by convolution

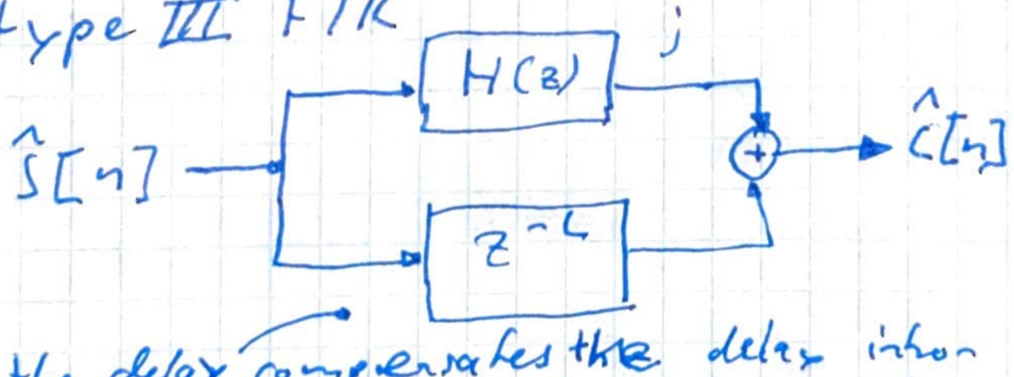
$$x_i[n] = h[n] * x_r[n]$$

- at the demodulator $\hat{s}[n] = s[n]$ is the real part of $c[n]$

↳ analytic bandpass signal is

$$\hat{c}[n] = \hat{s}[n] + j\,(h[n] * \hat{s}[n])$$

In practice the Hilbert filter is approximated with a casual $2L+1$ tap type III FIR



the delay compensates the delay introduced by the casual filter (sync the branches)

$$\hat{c}[n] = \hat{s}[n-L] + j\,(h[n] * \hat{s}[n])$$

DSP Coursera IV 16.8.24

- once the analytic bandpass signal is reconstructed
  - ↳ bring it back to baseband (complex demodulation ~~with~~ with carrier $-\omega_c$)

$$\hat{b}[n] = \hat{c}[n]\, e^{-j\omega_c n}$$

- interpolation property:
  - downsampling by K

$$\hat{a}[n] = \hat{b}[nK]$$

- slicer associates a group of M b.l to each received symbol

- descrambler reconstructs original binary stream

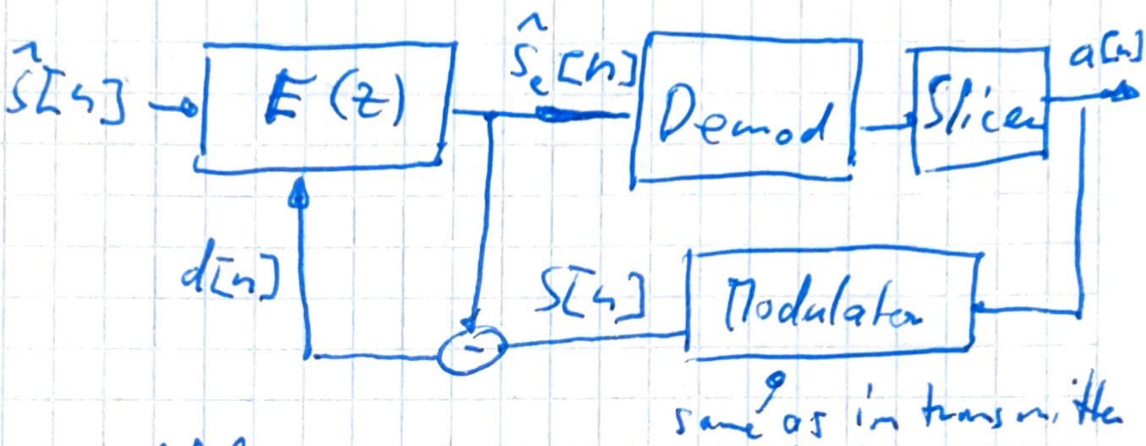
12.3.2. The Effects of the Channel

- channel affects received signal in three fundamental ways:
  - noise: SNR cannot exeed max. limit
  - distortion (acting as linear filter)
  - delays (~~xxx~~ linear)
- distortion and delay are linear
- lack of synchronization between transmitter and receiver

- Noise (see above)
  - analog noise is transformed to discrete-time noise by sample
  - as noise level increases (SNR decreases)
    - shape of constellation progressively looses tightness around alphabet values
    - decoding errors happen if noise level too high

- Equalization
  - passband of channel : region over which channel introduces only linear distortion
  - channel can be modeled as continuous-time linear filter $D_c(j\Omega)$
    - impulse response unknown and (potentially) time-varying
  - received signal (neglecting noise)
    $$\hat{S}(j\Omega) = D_c(j\Omega)\, S(j\Omega)$$
    - after sampler
    $$\hat{S}(e^{j\omega}) = D(e^{j\omega})\, S(e^{j\omega})$$
    - where $D(e^{j\omega})$ combined effect of
      - original channel and
      - anti-aliasing filter at D/A
  - counteract channel distortion
    - adaptive equalizer after A/D

- FIR filter
  - modified on the fly so that
    $$E(z) \approx 1/D(z)$$
  - demodulator contains exact copy of modulator



- The difference between equalized signal and the reconstructed original signal is used to adapt the taps of the equalizer

$$d[n] = \hat{s}_e[n] - s[n] \to 0$$

- initial estimate for $D(e^{j\omega})$ is missing
  ↳ transmitter sends a known trainings-sequence
- trainings sequence and other synchronization signals are sent when connection is established
  ↳ handshaking protocoll

DSP Coursera IV 16.8.24

· Delay

· continuous-time arrived at receiver:

$$\hat{s}(t) = (s * v)(t - t_d) + \eta(t)$$

where · $v(t)$: continuous-time impulse response of chanel

· $\eta(t)$: continuous-time noise process

· t_d: propagation delay

· after the sampler

· demodulation $\hat{s}[n] = \hat{s}(nT_s)$

· if we neglect noise and distortion:

$$\hat{s}[n] = s(nT_S - t_d) = s((n - n_d)T_S - \tau T_S)$$

↳ split delay as $t_d = (n_d + \tau) T_S$

· n_d: bulk delay $n_d \in \mathbb{N}$

$|\tau| \leq 1/2$

· bulk delay can be estimated by handshaking protocol (see book)