COM303: Digital Signal Processing

Lecture 22: Digital Communication Systems (I)

overview

- the communication channel
- bandwidth constraints
- power constraints

a comparison of data rates

- Transatlantic cable:
  - 1866: 8 words per minute ($\approx$5 bps)
  - 1956: AT&T, coax, 48 voice channels ($\approx$3Mbps)
  - 2005: Alcatel Tera10, fiber, 8.4 Tbps (8.4×10^{12} bps)
  - 2012: fiber, 60 Tbps
- Voiceband modems
  - 1950s: Bell 202, 1200 bps
  - 1990s: V90, 56Kbps
  - 2008: ADSL2+, 24Mbps

a comparison of data rates

- Transatlantic cable:
  - 1866: 8 words per minute ($\approx$5 bps)
  - 1956: AT&T, coax, 48 voice channels ($\approx$3Mbps)
  - 2005: Alcatel Tera10, fiber, 8.4 Tbps (8.4×10^{12} bps)
  - 2012: fiber, 60 Tbps
- Voiceband modems
  - 1950s: Bell 202, 1200 bps
  - 1990s: V90, 56Kbps
  - 2008: ADSL2+, 24Mbps

Success factors for digital communications

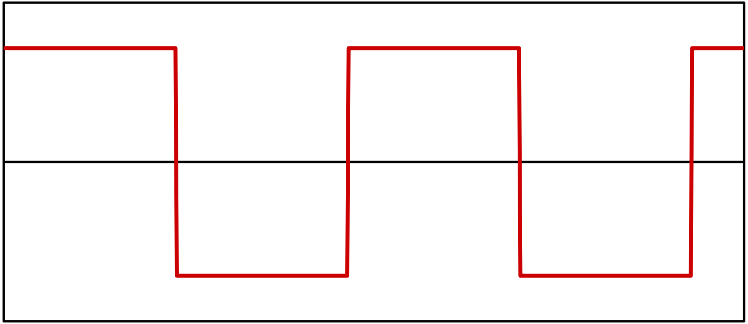
1. power of the digital paradigm
2. natural integration with information theory
3. hardware advancement

Success factors for digital communications

1) power of the DSP paradigm:

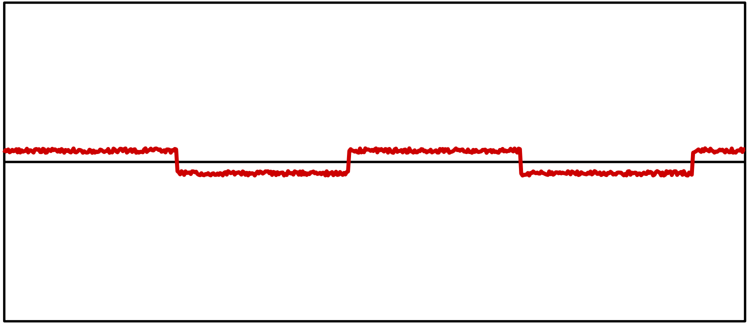
- integers are “easy” to regenerate
- good phase control
- adaptive algorithms

Regenerating signals



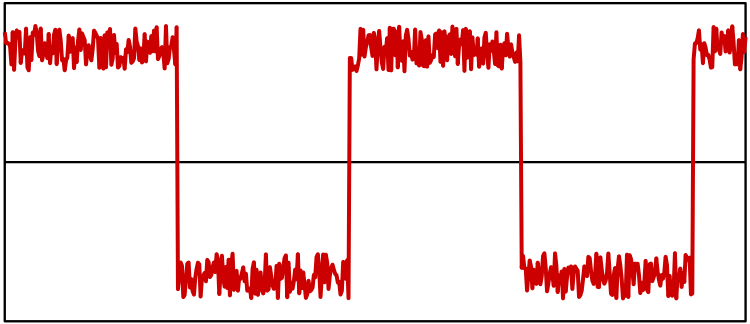
$x(t)$

Regenerating signals



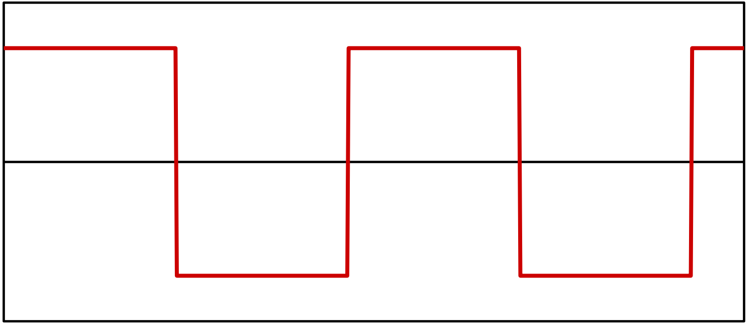
$$x(t)/G + \sigma(t)$$

Regenerating signals



$$G[x(t)/G + \sigma(t)] = x(t) + G\sigma(t)$$

Regenerating signals



$$\hat{x}_1(t) = G\,\mathrm{sgn}[x(t) + \sigma(t)]$$

Success factors for digital communications

2) algorithmic nature of DSP is a perfect match with information theory:

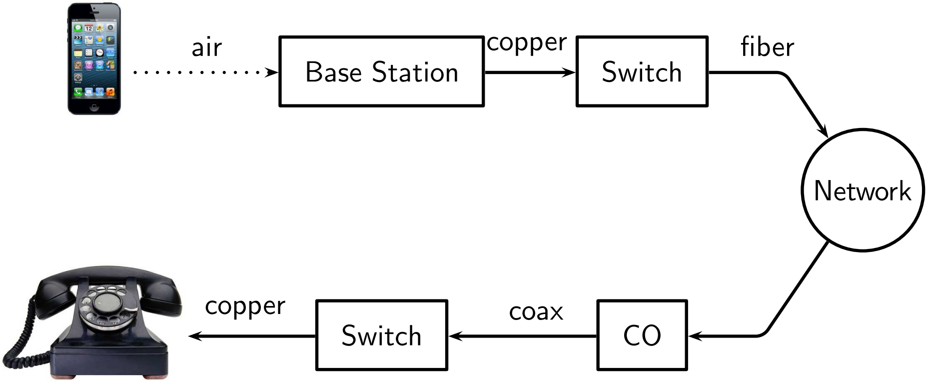
- error correction (CD's and DVD's)
- entropy coding (JPEG)

Success factors for digital communications

3) hardware advancement

- general-purpose platforms
- miniaturization
- power efficiency

The many incarnations of a conversation



The analog channel

unescapable "limits" of physical channels:

- bandwidth constraint
- power constraint

both constraints will affect the final *capacity* of the channel

The analog channel

unescapable "limits" of physical channels:

- bandwidth constraint
- power constraint

both constraints will affect the final *capacity* of the channel

The analog channel

unescapable “limits” of physical channels:

- bandwidth constraint
- power constraint

both constraints will affect the final *capacity* of the channel

The analog channel's capacity

maximum amount of information that can be *reliably* delivered over a channel
(bits per second)

The harsh truth about reliability

we cannot design a perfect (error-free) communication system because of noise

but

we can design a system with arbitrary small error rate (e.g. 10^{-6})

The design problem

- transmitted data is a sequence of digital symbols $a[n] \in \mathcal{A}$
- we will model $a[n]$ as a zero-mean white process
- we need to transform $a[n]$ into an analog signal $s(t)$ that fulfills both bandwidth and power constraints

Bandwidth vs capacity: intuition

- we want to transmit a data sequence $a[n]$ over an analog channel
- we sinc-interpolate $a[n]$ with a period T_s
- if we make T_s small we can send more symbols per unit of time...
- ... but the bandwidth of the signal will grow as $1/T_s$

Bandwidth vs capacity: intuition

- we want to transmit a data sequence $a[n]$ over an analog channel
- we sinc-interpolate $a[n]$ with a period T_s
- if we make T_s small we can send more symbols per unit of time...
- ... but the bandwidth of the signal will grow as $1/T_s$

Bandwidth vs capacity: intuition

- we want to transmit a data sequence $a[n]$ over an analog channel
- we sinc-interpolate $a[n]$ with a period T_s
- if we make T_s small we can send more symbols per unit of time...
- ... but the bandwidth of the signal will grow as $1/T_s$

Bandwidth vs capacity: intuition

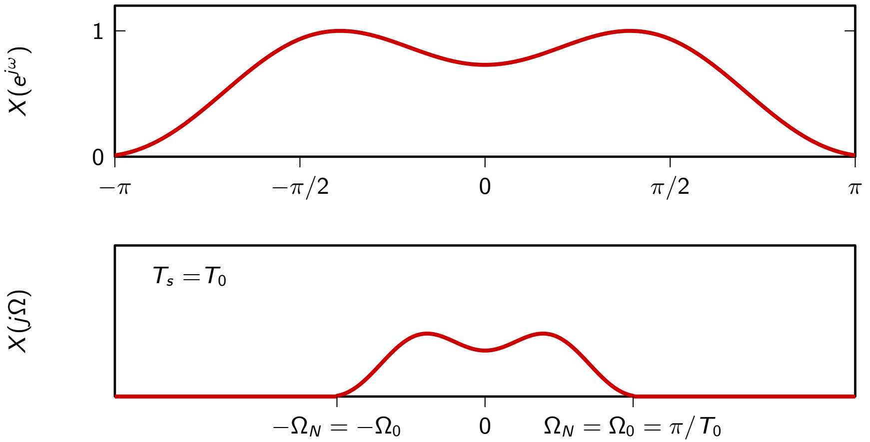
- we want to transmit a data sequence $a[n]$ over an analog channel
- we sinc-interpolate $a[n]$ with a period T_s
- if we make T_s small we can send more symbols per unit of time...
- ... but the bandwidth of the signal will grow as $1/T_s$

Sinc interpolation

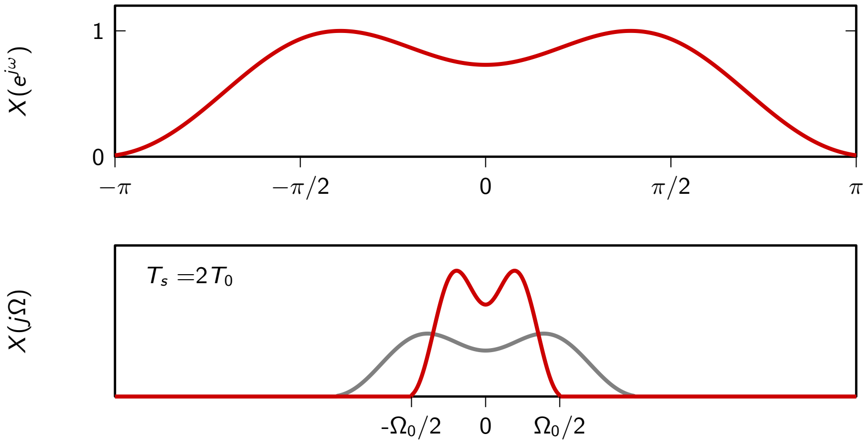
$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

$$X(j\Omega) = \begin{cases} (\pi/\Omega_N) X(e^{j\pi(\Omega/\Omega_N)}) & \text{for } |\Omega| \leq \Omega_N = \dfrac{\pi}{T_s} \\ 0 & \text{otherwise} \end{cases}$$

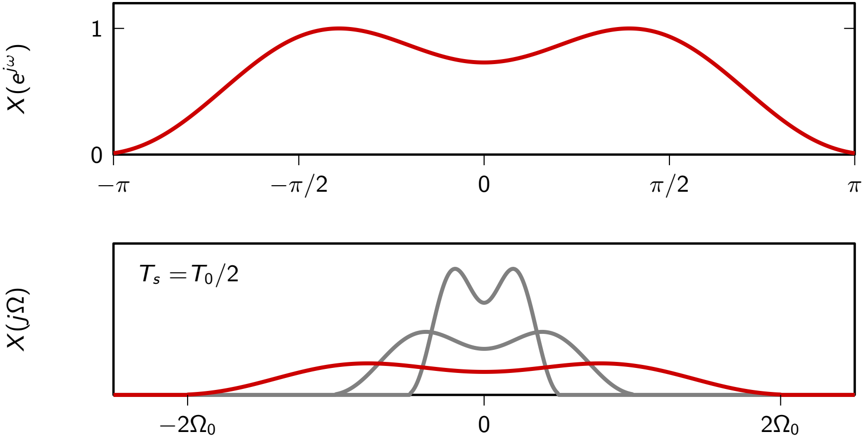
Spectrum of interpolated signals



Spectrum of interpolated signals



Spectrum of interpolated signals



Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to "guess" what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to “guess” what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to "guess" what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to "guess" what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

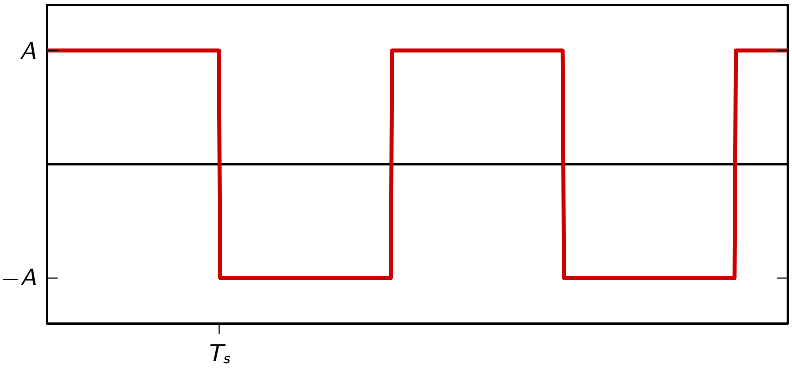
Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to "guess" what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

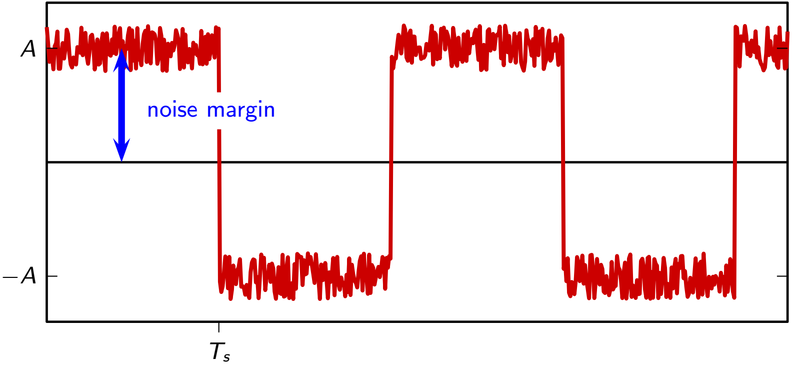
Power vs capacity: intuition

- number of symbols per second is determined by bandwidth constraint
- to increase information rate we must increase the number of possible symbols
- power is proportional to the square of the max symbol
- to keep power limited we need to pack symbol values closer together
- all channels introduce noise; at the receiver we have to “guess” what was transmitted
- closer symbols have less *noise margin*, i.e., we lose reliability

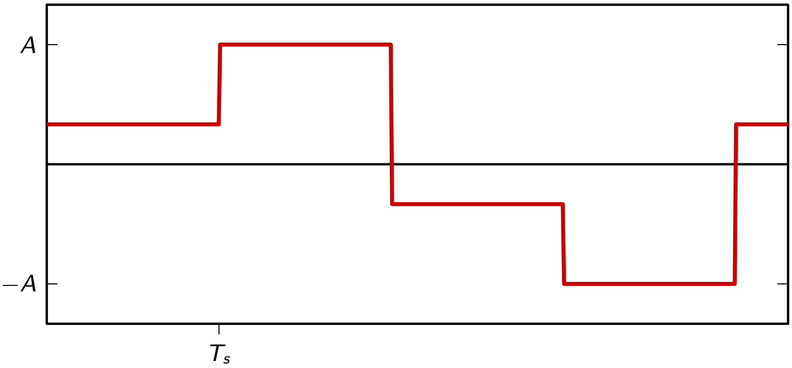
1 bit per symbol



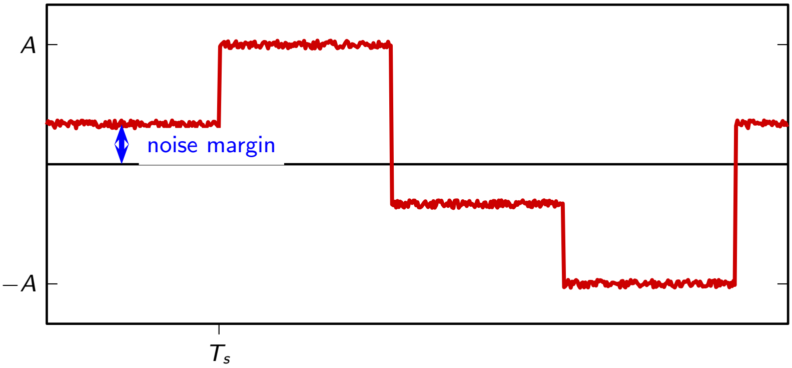
1 bit per symbol



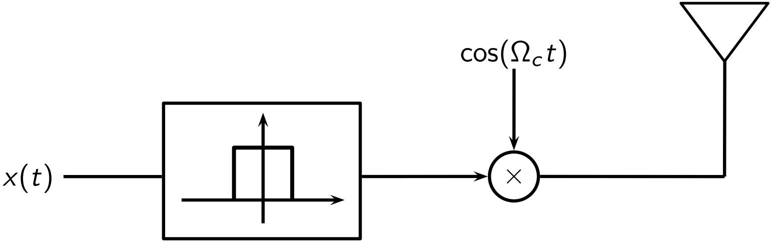
2 bits per symbol



2 bits per symbol



Example: the AM radio channel



Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

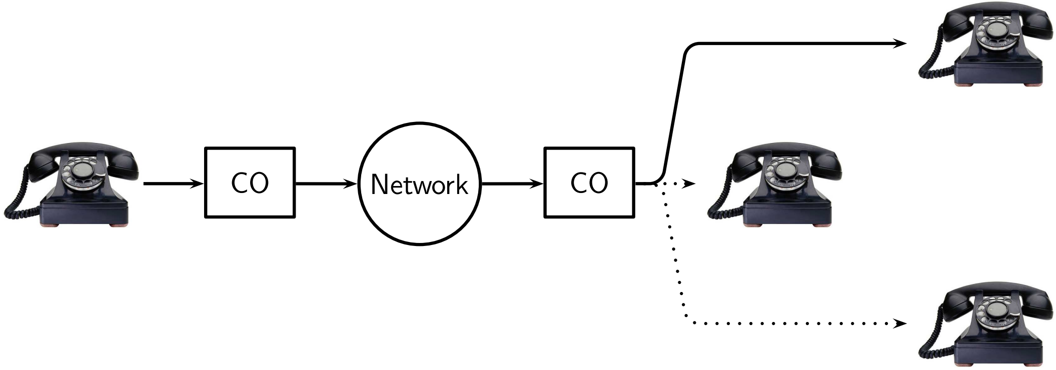
Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

Example: the AM radio channel

- from 530kHz to 1.7MHz
- each channel is 8KHz
- power limited by law:
  - daytime/nighttime
  - interference
  - health hazards

Example: the telephone channel



Example: the telephone channel

- from around 300Hz to around 3500Hz
- power limited by law to 0.2-0.7V rms
- noise is rather low: SNR usually 30dB or more

Example: the telephone channel

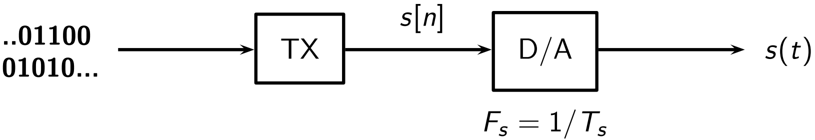
- from around 300Hz to around 3500Hz
- power limited by law to 0.2-0.7V rms
- noise is rather low: SNR usually 30dB or more

Example: the telephone channel

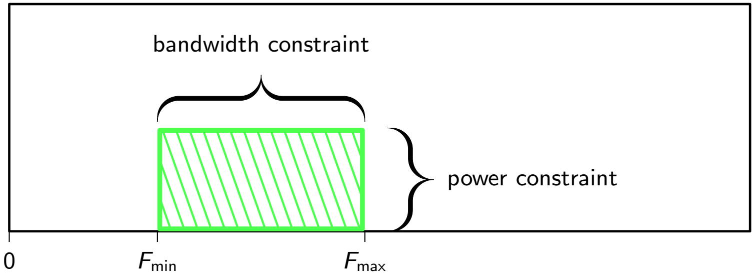
- from around 300Hz to around 3500Hz
- power limited by law to 0.2-0.7V rms
- noise is rather low: SNR usually 30dB or more

The all-digital paradigm

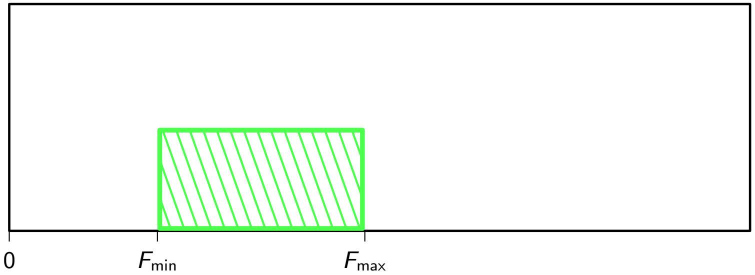
keep everything digital until we hit the physical channel



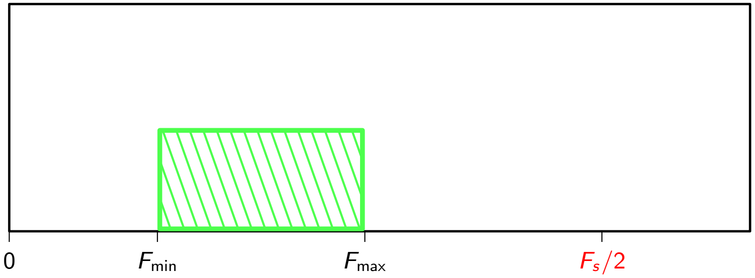
Let's look at the channel constraints



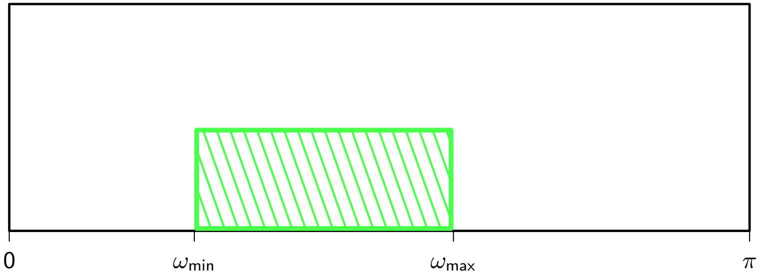
Converting the specs to a digital design



Converting the specs to a digital design



Converting the specs to a digital design



Transmitter design

some working hypotheses:

- convert the bitstream into a sequence of symbols $a[n]$ via a mapper
- model $a[n]$ as a white random sequence (add a scrambler on the bitstream to make sure)
- now we need to convert $a[n]$ into a continuous-time signal within the constraints

Transmitter design

some working hypotheses:

- convert the bitstream into a sequence of symbols $a[n]$ via a mapper
- model $a[n]$ as a white random sequence (add a scrambler on the bitstream to make sure)
- now we need to convert $a[n]$ into a continuous-time signal within the constraints

Transmitter design

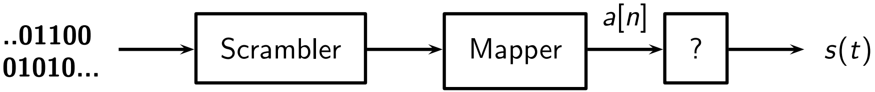
some working hypotheses:

- convert the bitstream into a sequence of symbols $a[n]$ via a mapper
- model $a[n]$ as a white random sequence (add a scrambler on the bitstream to make sure)
- now we need to convert $a[n]$ into a continuous-time signal within the constraints

Transmitter design

some working hypotheses:

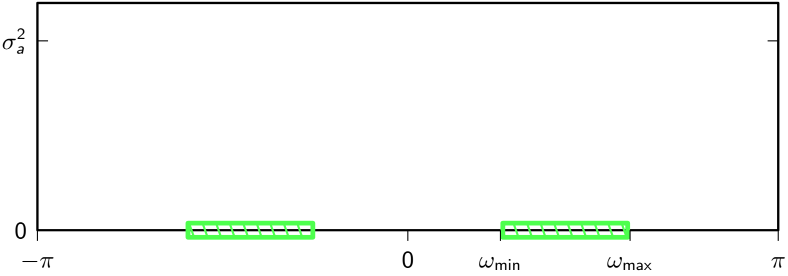
- convert the bitstream into a sequence of symbols $a[n]$ via a mapper
- model $a[n]$ as a white random sequence (add a scrambler on the bitstream to make sure)
- now we need to convert $a[n]$ into a continuous-time signal within the constraints



the bandwidth constraint

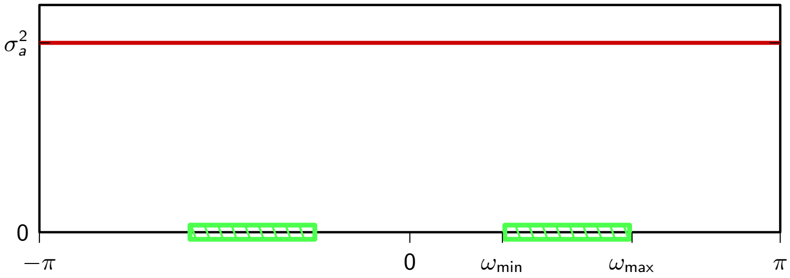
First problem: the bandwidth constraint

$a[n]$ white: $P_a(e^{j\omega}) = \sigma_a^2$



First problem: the bandwidth constraint

$$a[n] \text{ white: } P_a(e^{j\omega}) = \sigma_a^2$$



Shaping the bandwidth

Our problem:

- bandwidth 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
- the answer is *multirate* techniques

Shaping the bandwidth

Our problem:

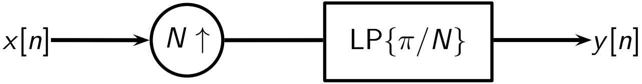
- bandwidth constraint requires us to control the spectral support of a signal
- we need to be able to "shrink" the support of a full-band signal
- the answer is *multirate* techniques

Shaping the bandwidth

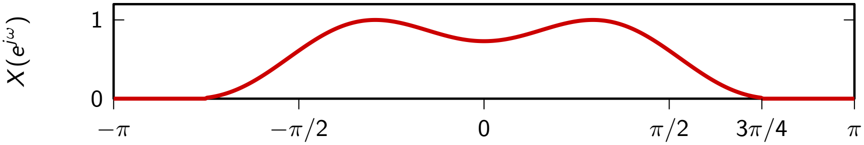
Our problem:

- bandwidth 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
- the answer is *multirate* techniques

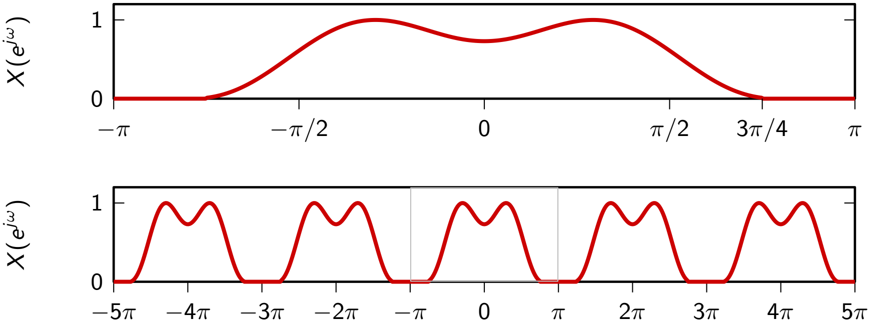
ideal digital interpolator



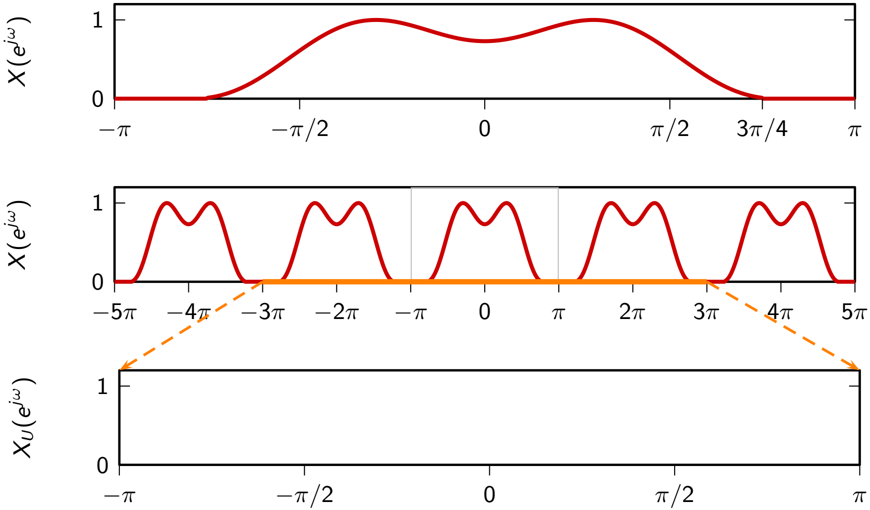
Upsampling by $K = 3$



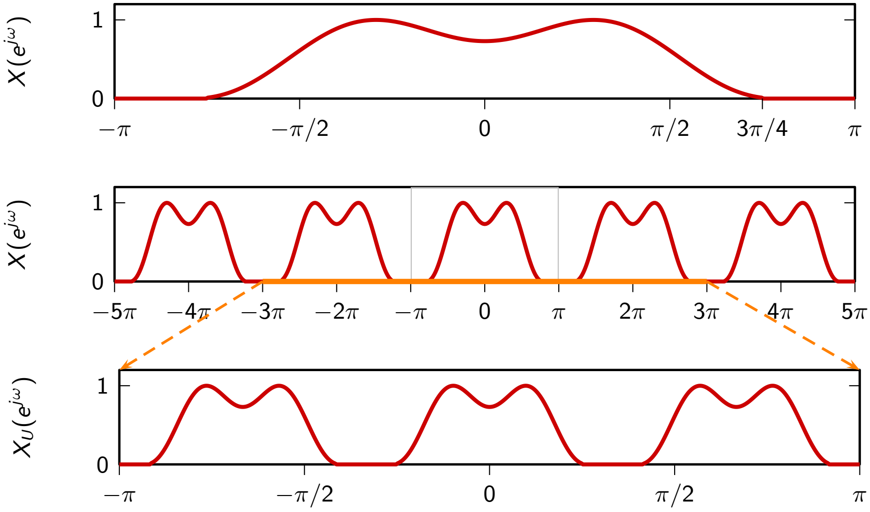
Upsampling by $K = 3$



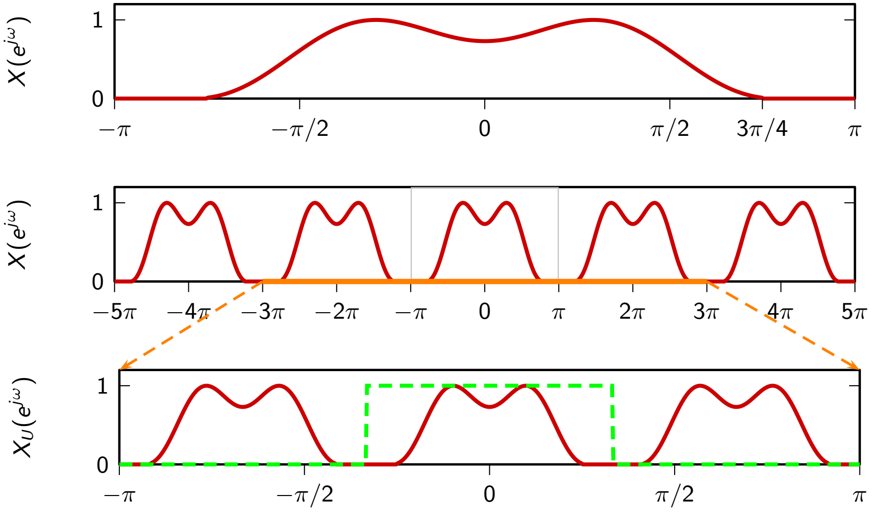
Upsampling by $K = 3$



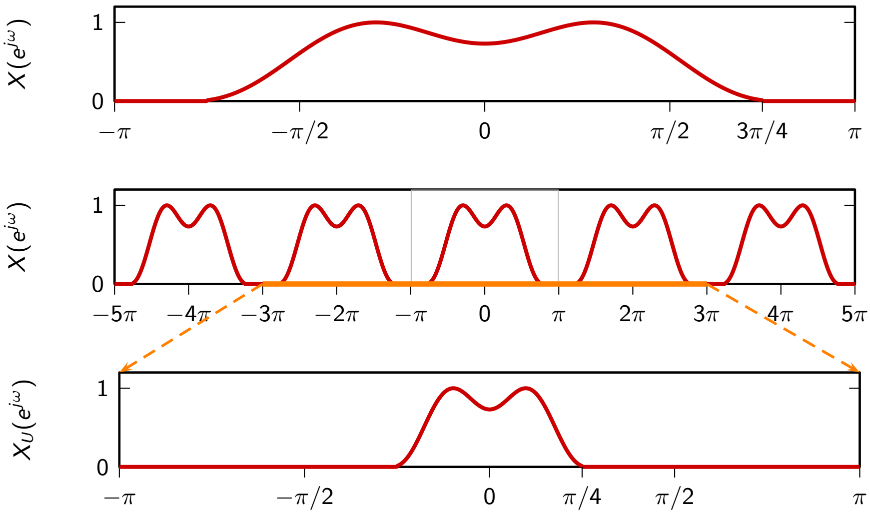
Upsampling by $K = 3$



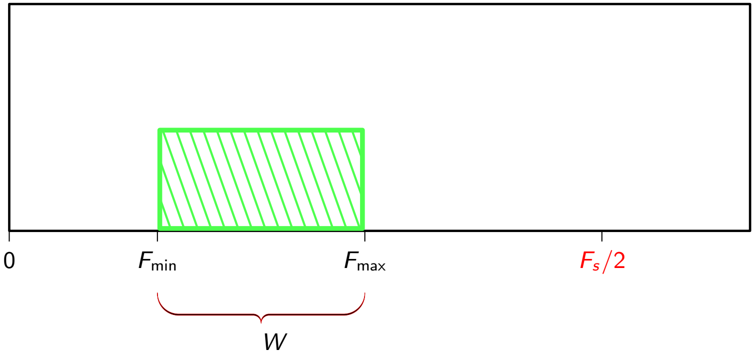
Upsampling by $K = 3$



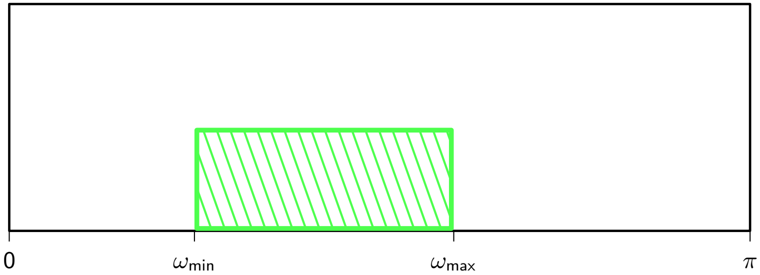
Upsampling by $K = 3$



Fulfilling the bandwidth constraint



Fulfilling the bandwidth constraint



Here's a neat trick

let $W = F_{\max} - F_{\min}$; pick F_s so that:

- $F_s > 2F_{\max}$ (obviously)
- $F_s = KW,\ K \in \mathbb{N}$
- $\omega_{\max} - \omega_{\min} = 2\pi \frac{W}{F_s} = \frac{2\pi}{K}$
- we can simply upsample by K

Here's a neat trick

let $W = F_{\max} - F_{\min}$; pick F_s so that:

- $F_s > 2F_{\max}$ (obviously)
- $F_s = KW$, $K \in \mathbb{N}$
- $\omega_{\max} - \omega_{\min} = 2\pi \dfrac{W}{F_s} = \dfrac{2\pi}{K}$
- we can simply upsample by K

Here's a neat trick

let $W = F_{\max} - F_{\min}$; pick F_s so that:

- $F_s > 2F_{\max}$ (obviously)
- $F_s = KW,\ K \in \mathbb{N}$
- $\omega_{\max} - \omega_{\min} = 2\pi \frac{W}{F_s} = \frac{2\pi}{K}$
- we can simply upsample by K

Here's a neat trick

let $W = F_{\max} - F_{\min}$; pick F_s so that:

- $F_s > 2F_{\max}$ (obviously)
- $F_s = KW$, $K \in \mathbb{N}$

- $\omega_{\max} - \omega_{\min} = 2\pi \dfrac{W}{F_s} = \dfrac{2\pi}{K}$
- we can simply upsample by K

Baud rate

- upsampling does not change the *data* rate, only the sample rate
- we produce (and transmit) W symbols per second
- W is sometimes called the Baud rate of the system and is equal to the available bandwidth

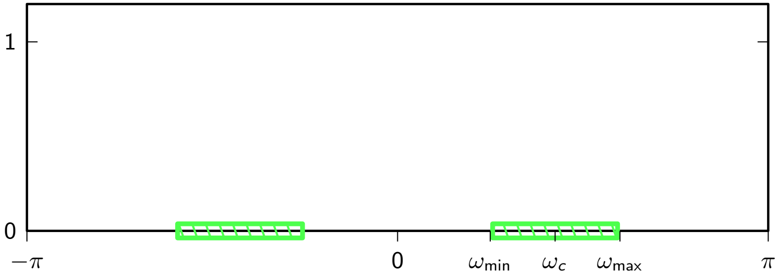
Baud rate

- upsampling does not change the *data* rate, only the sample rate
- we produce (and transmit) W symbols per second
- W is sometimes called the Baud rate of the system and is equal to the available bandwidth

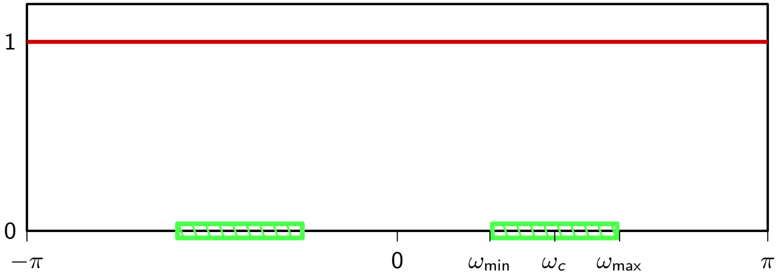
Baud rate

- upsampling does not change the *data* rate, only the sample rate
- we produce (and transmit) W symbols per second
- W is sometimes called the Baud rate of the system and is equal to the available bandwidth

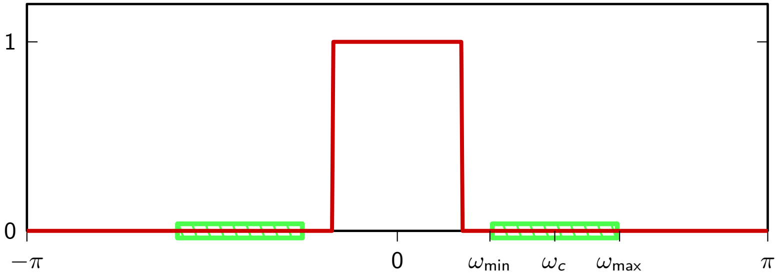
Fulfilling the bandwidth constraint



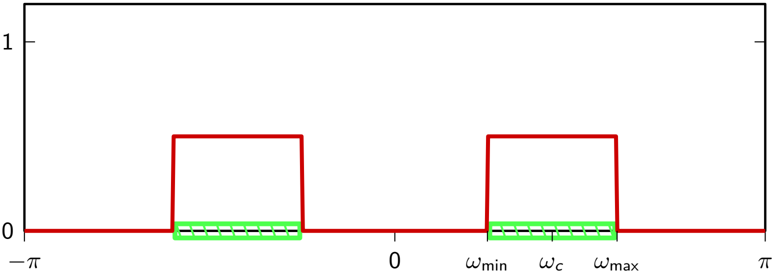
Fulfilling the bandwidth constraint



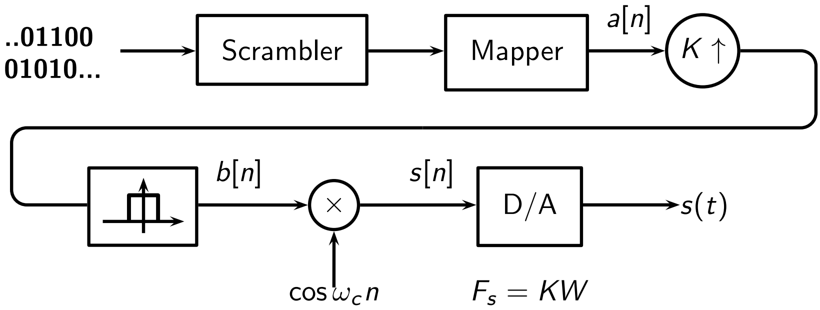
Fulfilling the bandwidth constraint



Fulfilling the bandwidth constraint



Transmitter design, continued

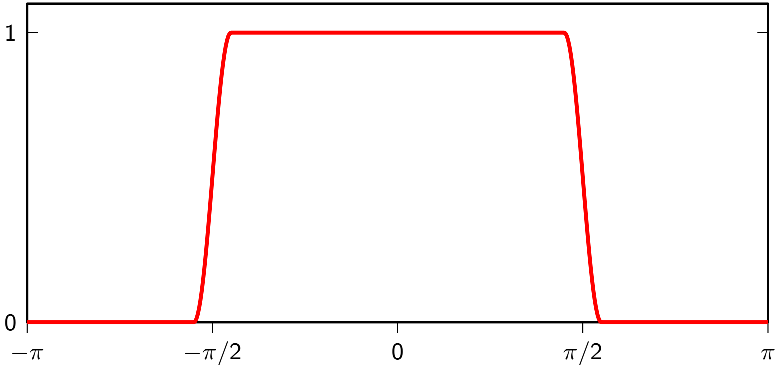


Raised Cosine

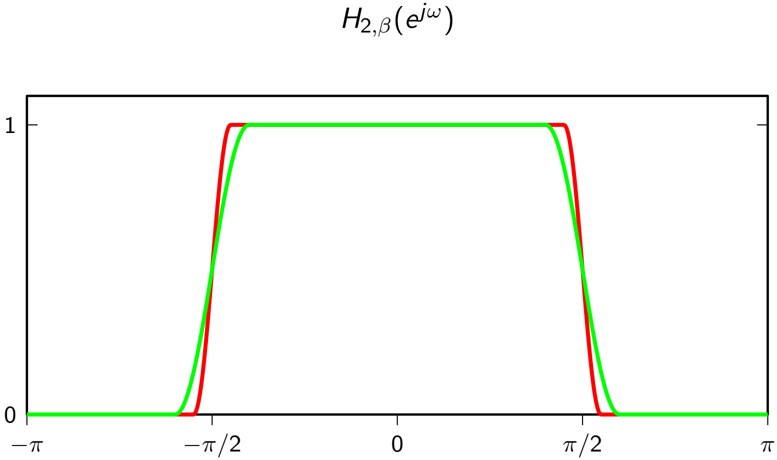
$$H_{K,\beta}(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq \dfrac{\pi(1-\beta)}{K} \\ \frac{1}{2}\left(1 + \cos\left(\dfrac{K|\omega| - (1-\beta)\pi}{\beta}\right)\right) & \dfrac{\pi(1-\beta)}{K} \leq |\omega| \leq \dfrac{\pi(1+\beta)}{K} \\ 0 & |\omega| > \dfrac{\pi(1+\beta)}{K} \end{cases}$$

Raised Cosine

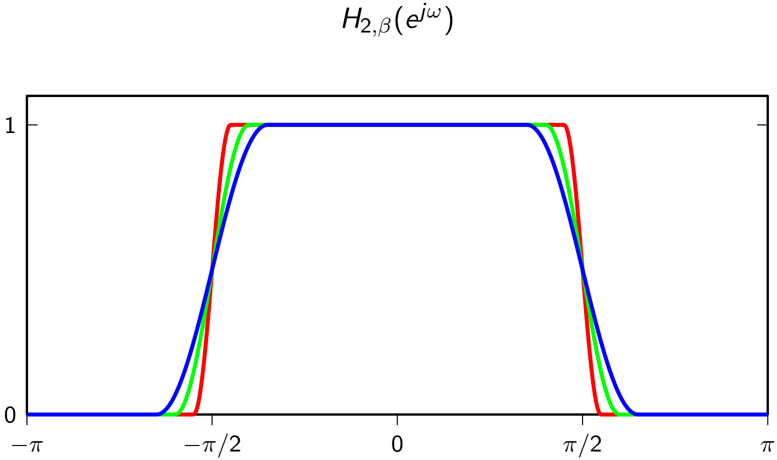
$H_{2,\beta}(e^{j\omega})$



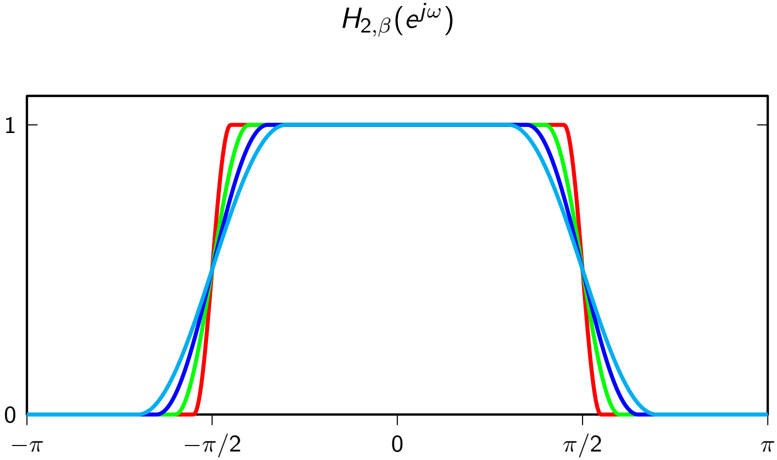
Raised Cosine



Raised Cosine



Raised Cosine

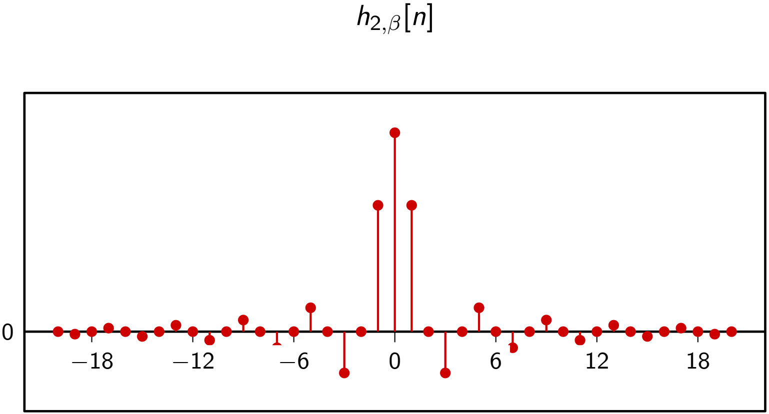


Raised Cosine

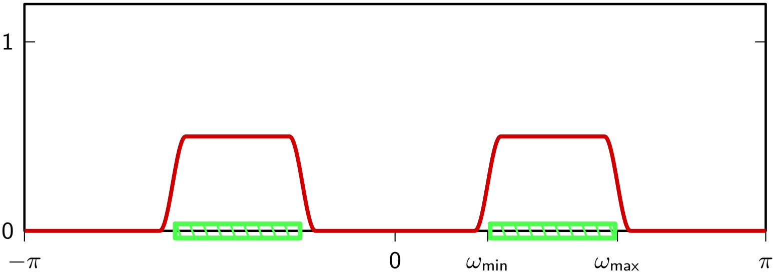
$$h_{K,\beta}[nK] = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h_{K,\beta}[nK] \propto \frac{1}{(\beta n)^2}$$

Raised Cosine: $1/n^2$ decay



Spectral shaping with raised cosine



the power constraint

Overview:

- Noise and probability of error
- Signaling alphabet and power
- QAM signaling

Overview:

- Noise and probability of error
- Signaling alphabet and power
- QAM signaling

Overview:

- Noise and probability of error
- Signaling alphabet and power
- QAM signaling

Transmission reliability

- transmitter sends a sequence of symbols $a[n]$
- receiver obtaines 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

Transmission reliability

- transmitter sends a sequence of symbols $a[n]$
- receiver obtaines 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

Transmission reliability

- transmitter sends a sequence of symbols $a[n]$
- receiver obtaines 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

Transmission reliability

- transmitter sends a sequence of symbols $a[n]$
- receiver obtaines 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 wrt power of the signal
- decoding strategy
- *alphabet* of transmission symbols

Probability of error

depends on:

- power of the noise wrt power of the signal
- decoding strategy
- *alphabet* of transmission symbols

Probability of error

depends on:

- power of the noise wrt power of the signal
- decoding strategy
- *alphabet* of transmission symbols

Signaling alphabets

- we have a (randomized) bitstream coming in
- we want to send some upsampled and interpolated samples over the channel
- how do we go from bitstream to samples?

Signaling alphabets

- we have a (randomized) bitstream coming in
- we want to send some upsampled and interpolated samples over the channel
- how do we go from bitstream to samples?

Signaling alphabets

- we have a (randomized) bitstream coming in
- we want to send some upsampled and interpolated samples over the channel
- how do we go from bitstream to samples?

Mappers and slicers

mapper:

- split incoming bitstream into chunks
- assign a symbol $a[n]$ from a finite alphabet $\mathcal{A}$ to each chunk

slicer:

- receive a value $\hat{a}[n]$
- decide which symbol from $\mathcal{A}$ is "closest" to $\hat{a}[n]$
- piece back together the corresponding bitstream

Example: two-level signaling

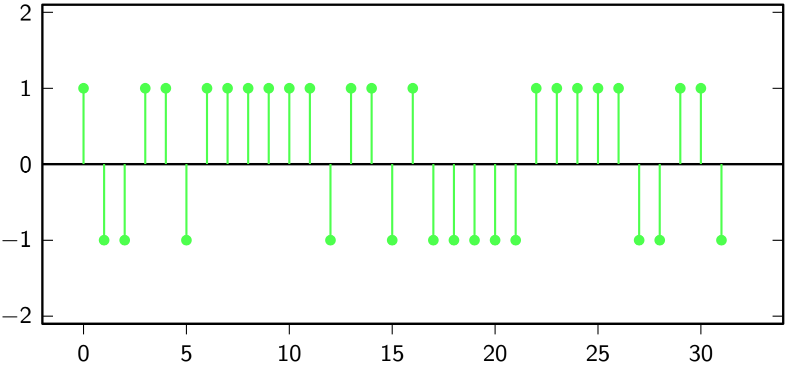
mapper:

- split incoming bitstream into single bits
- $a[n] = G$ if the bit is 1, $a[n] = -G$ if the bit is 0

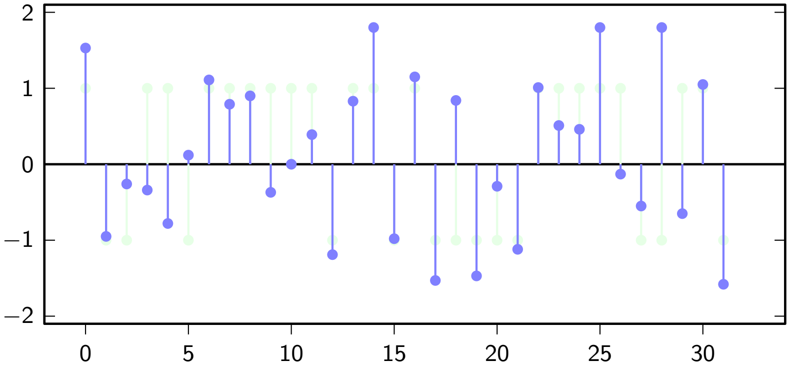
slicer:

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

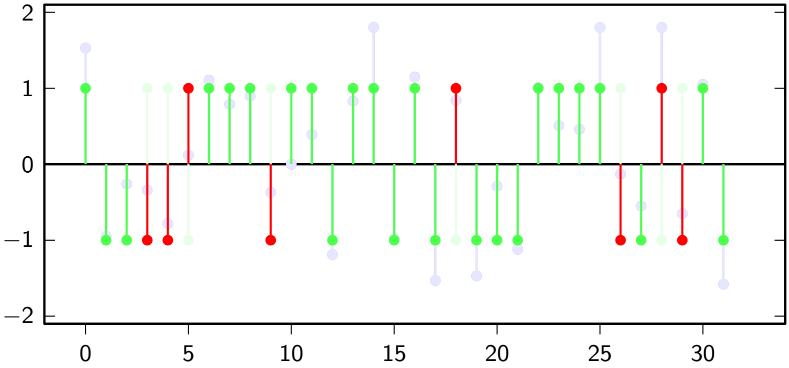
Example: two-level signaling



Example: two-level signaling



Example: two-level signaling



Example: two-level signaling

let's look at the probability of error after making some hypotheses:

- $\hat{a}[n] = a[n] + \eta[n]$
- bits in bitstream are equiprobable
- noise and signal are independent
- noise is additive white Gaussian noise with zero mean and variance σ_0

Example: two-level signaling

$$
\begin{aligned}
P_{\text{err}} &= P[\ \eta[n] < -G \ \mid\ n\text{-th bit is } 1\ ]\, P[\ n\text{-th bit is } 1\ ] + \\
&\quad P[\ \eta[n] > G \ \mid\ n\text{-th bit is } 0\ ]\, P[\ n\text{-th bit is } 0\ ] \\
&= (P[\ \eta[n] < -G\ ] + P[\ \eta[n] > G\ ])/2 \\
&= P[\ \eta[n] > G\ ] \\
&= \int_G^{\infty} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}}\, d\tau \\
&= Q(G/\sigma_0) = \frac{1}{2}\mathrm{erfc}((G/\sigma_0)/\sqrt{2})
\end{aligned}
$$

Example: two-level signaling

$$
\begin{aligned}
P_{\text{err}} &= P[\ \eta[n] < -G \ \mid\ n\text{-th bit is } 1\ ]\, P[\ n\text{-th bit is } 1\ ] + \\
&\qquad P[\ \eta[n] > G \ \mid\ n\text{-th bit is } 0\ ]\, P[\ n\text{-th bit is } 0\ ] \\
&= (P[\ \eta[n] < -G\ ] + P[\ \eta[n] > G\ ])/2 \\
&= P[\ \eta[n] > G\ ] \\
&= \int_G^{\infty} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}}\, d\tau \\
&= Q(G/\sigma_0) = \frac{1}{2}\operatorname{erfc}((G/\sigma_0)/\sqrt{2})
\end{aligned}
$$

Example: two-level signaling

$$
\begin{aligned}
P_{\text{err}} &= P[\ \eta[n] < -G \ \mid\ n\text{-th bit is } 1\ ]\, P[\ n\text{-th bit is } 1\ ] + \\
&\qquad P[\ \eta[n] > G \ \mid\ n\text{-th bit is } 0\ ]\, P[\ n\text{-th bit is } 0\ ] \\
&= (P[\ \eta[n] < -G\ ] + P[\ \eta[n] > G\ ])/2 \\
&= P[\ \eta[n] > G\ ] \\
&= \int_G^{\infty} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}}\, d\tau \\
&= Q(G/\sigma_0) = \frac{1}{2}\operatorname{erfc}((G/\sigma_0)/\sqrt{2})
\end{aligned}
$$

Example: two-level signaling

$$\begin{aligned} P_{\text{err}} &= P[\ \eta[n] < -G \ \mid\ n\text{-th bit is 1}\ ]\, P[\ n\text{-th bit is 1}\ ] + \\ &\quad P[\ \eta[n] > G \ \mid\ n\text{-th bit is 0}\ ]\, P[\ n\text{-th bit is 0}\ ] \\ &= (P[\ \eta[n] < -G\ ] + P[\ \eta[n] > G\ ])/2 \\ &= P[\ \eta[n] > G\ ] \\ &= \int_G^\infty \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}} d\tau \\ &= Q(G/\sigma_0) = \frac{1}{2}\text{erfc}((G/\sigma_0)/\sqrt{2}) \end{aligned}$$

Example: two-level signaling

$$
\begin{aligned}
P_{\text{err}} &= P[\ \eta[n] < -G \ \mid\ n\text{-th bit is 1}\ ]\,P[\ n\text{-th bit is 1}\ ]+ \\
&\qquad P[\ \eta[n] > G \ \mid\ n\text{-th bit is 0}\ ]\,P[\ n\text{-th bit is 0}\ ] \\
&= (P[\ \eta[n] < -G\ ] + P[\ \eta[n] > G\ ])/2 \\
&= P[\ \eta[n] > G\ ] \\
&= \int_G^{\infty} \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{\tau^2}{2\sigma_0^2}}\, d\tau \\
&= Q(G/\sigma_0) = \frac{1}{2}\operatorname{erfc}((G/\sigma_0)/\sqrt{2})
\end{aligned}
$$

Example: two-level signaling

transmitted power

$$\sigma_s^2 = G^2\, P[n\text{-th bit is }1] + G^2\, P[n\text{-th bit is }0]$$

$$= G^2$$

$$P_{\text{err}} = Q(\sigma_s/\sigma_0) = Q(\sqrt{\text{SNR}})$$

Example: two-level signaling

transmitted power

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

$$P_{\text{err}} = Q(\sigma_s/\sigma_0) = Q(\sqrt{\text{SNR}})$$

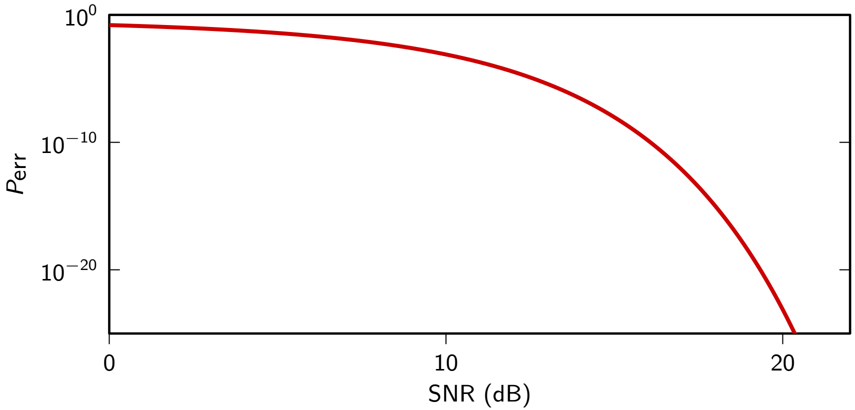
Example: two-level signaling

transmitted power

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

$$P_{\text{err}} = Q(\sigma_s/\sigma_0) = Q(\sqrt{\text{SNR}})$$

Probability of error



Lesson learned:

- to reduce the probability of error increase SNR
- can't control noise so we must increase G
- increasing G increases the power
- we can't go above the channel's power constraint!

Lesson learned:

- to reduce the probability of error increase SNR
- can't control noise so we must increase G
- increasing G increases the power
- we can't go above the channel's power constraint!

Lesson learned:

- to reduce the probability of error increase SNR
- can't control noise so we must increase G
- increasing G increases the power
- we can't go above the channel's power constraint!

Lesson learned:

- to reduce the probability of error increase SNR
- can't control noise so we must increase G
- increasing G increases the power
- we can't go above the channel's power constraint!

Multilevel signaling

- binary signaling is not very efficient (one bit at a time)
- to increase the throughput we can use multilevel signaling
- many ways to do so, we will just scratch the surface

Multilevel signaling

- binary signaling is not very efficient (one bit at a time)
- to increase the throughput we can use multilevel signaling
- many ways to do so, we will just scratch the surface

Multilevel signaling

- binary signaling is not very efficient (one bit at a time)
- to increase the throughput we can use multilevel signaling
- many ways to do so, we will just scratch the surface

PAM

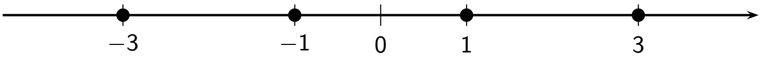
mapper:

- split incoming bitstream into chunks of M bits
- chunks define a sequence of integers $k[n] \in \{0, 1, \ldots, 2^M - 1\}$
- $a[n] = G((-2^M + 1) + 2k[n])$ (odd integers around zero)

slicer:

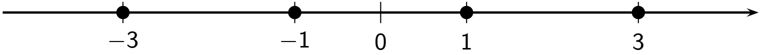
- $a'[n] = \arg\min_{a \in \mathcal{A}} [|\hat{a}[n] - a|]$

PAM, $M = 2, G = 1$



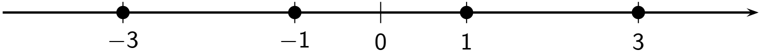
- distance between points is $2G$
- using odd integers creates a zero-mean sequence

PAM, $M = 2, G = 1$



- distance between points is $2G$
- using odd integers creates a zero-mean sequence

PAM, $M = 2, G = 1$



- distance between points is $2G$
- using odd integers creates a zero-mean sequence

From PAM to QAM

- error analysis for PAM along the lines of binary signaling
- can we increase the throughput even further?
- here's a wild idea, let's use complex numbers

From PAM to QAM

- error analysis for PAM along the lines of binary signaling
- can we increase the throughput even further?
- here's a wild idea, let's use complex numbers

From PAM to QAM

- error analysis for PAM along the lines of binary signaling
- can we increase the throughput even further?
- here's a wild idea, let's use complex numbers

QAM

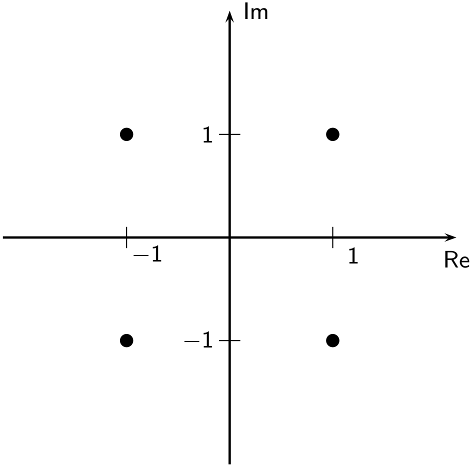
mapper:

- 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 $a_i[n]$
- $a[n] = G(a_r[n] + ja_i[n])$

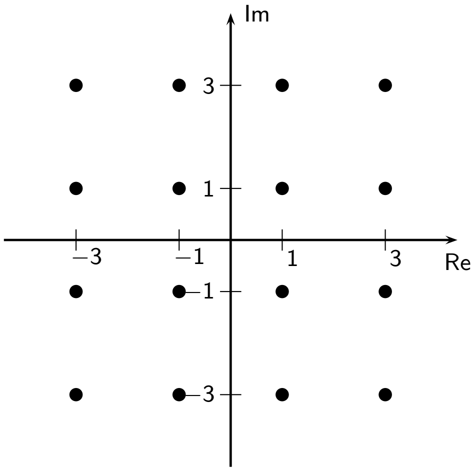
slicer:

- $a'[n] = \arg\min_{a \in \mathcal{A}} [|\hat{a}[n] - a|]$

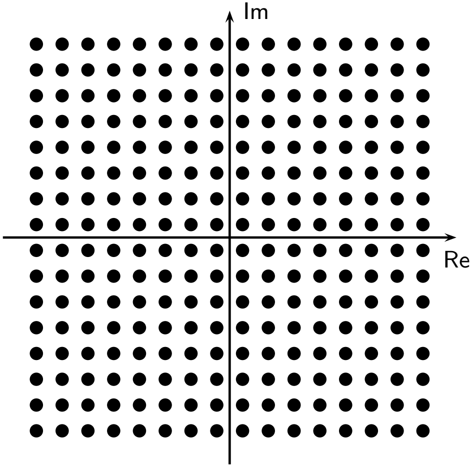
QAM, $M = 2, G = 1$



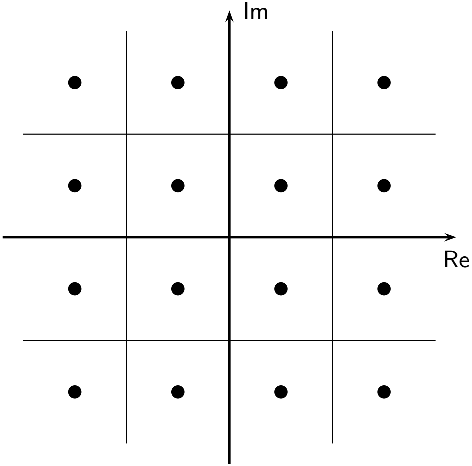
QAM, $M = 4$, $G = 1$



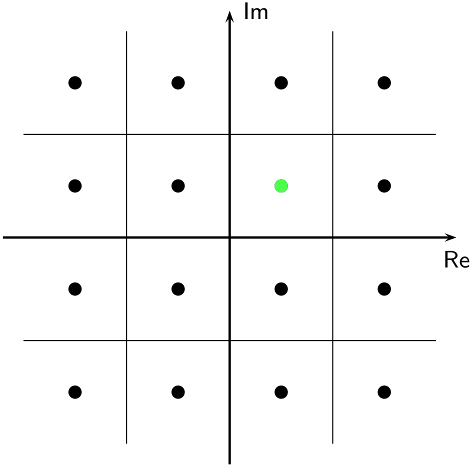
QAM, $M = 8, G = 1$



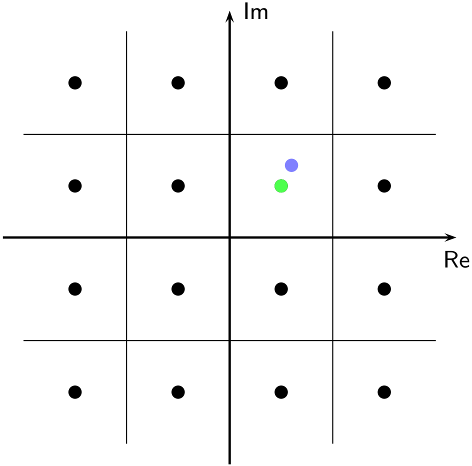
QAM



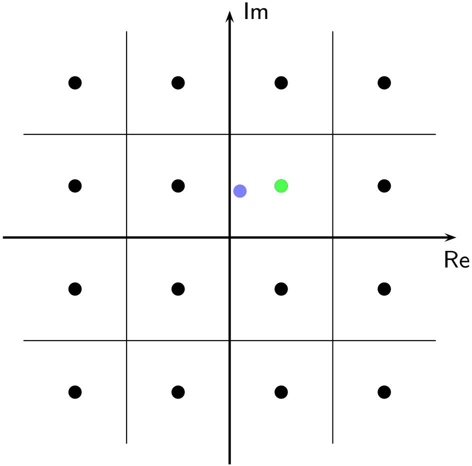
QAM



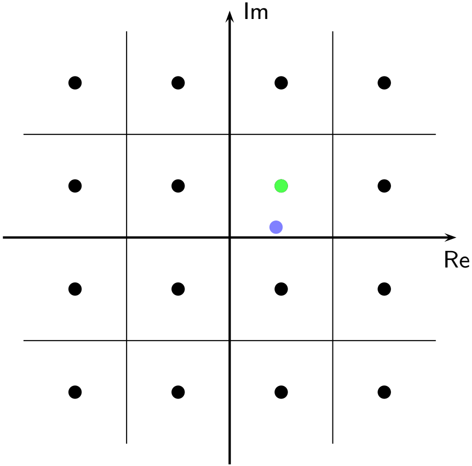
QAM



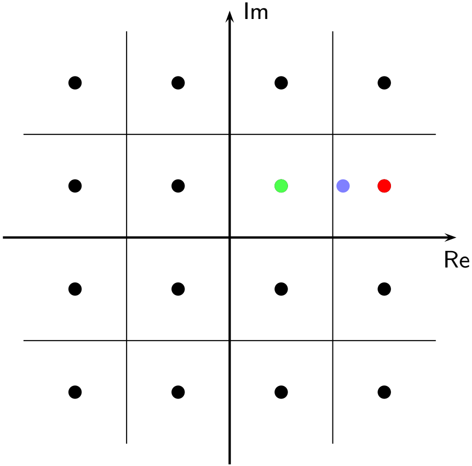
QAM



QAM



QAM

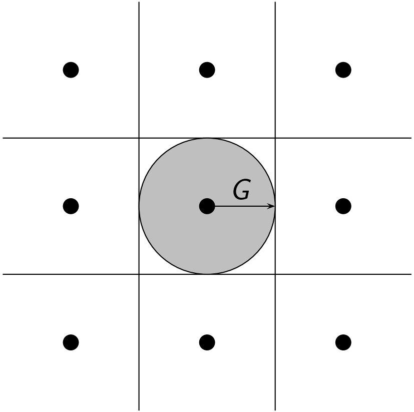


QAM, probability of error

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

$$\begin{aligned} P_{\text{err}} &= 1 - P[|\operatorname{Re}(\eta[n])| < G \wedge |\operatorname{Im}(\eta[n])| < G] \\ &= 1 - \int_D f_\eta(z)\, dz \end{aligned}$$

QAM, probability of error, circular approximation



QAM, probability of error

$$P_{\text{err}} \approx 1 - \int_{|z|<G} f_\eta(z)\, dz \qquad\qquad f_\eta(z) = \frac{1}{\pi\sigma_0^2} e^{-\frac{|z|^2}{\sigma_0^2}}$$

$z \rightarrow \rho e^{j\theta}$

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

$$= e^{-\frac{G^2}{\sigma_0^2}}$$

QAM, probability of error

$$P_{\text{err}} \approx 1 - \int_{|z|<G} f_\eta(z)\, dz \qquad f_\eta(z) = \frac{1}{\pi\sigma_0^2} e^{-\frac{|z|^2}{\sigma_0^2}}$$

$z \to \rho e^{j\theta}$

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

$$= e^{-\frac{G^2}{\sigma_0^2}}$$

QAM, probability of error

$$P_{\text{err}} \approx 1 - \int_{|z|<G} f_\eta(z)\, dz \qquad\qquad f_\eta(z) = \frac{1}{\pi\sigma_0^2} e^{-\frac{|z|^2}{\sigma_0^2}}$$

$z \to \rho e^{j\theta}$

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

$$= e^{-\frac{G^2}{\sigma_0^2}}$$

QAM, probability of error

$$P_{\text{err}} \approx 1 - \int_{|z|<G} f_\eta(z)\, dz \qquad f_\eta(z) = \frac{1}{\pi\sigma_0^2} e^{-\frac{|z|^2}{\sigma_0^2}}$$

$$z \rightarrow \rho e^{j\theta}$$

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

$$= e^{-\frac{G^2}{\sigma_0^2}}$$

QAM, probability of error

transmitted power (all symbols equiprobable and independent):

$$\sigma_s^2 = G^2 \frac{1}{2^M} \sum_{a \in \mathcal{A}} |a|^2$$

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

$$P_{\text{err}} \approx e^{-\frac{G^2}{\sigma_0^2}} \approx e^{-3\,\mathrm{SNR}/2^{(M+1)}}$$

QAM, probability of error

transmitted power (all symbols equiprobable and independent):

$$\sigma_s^2 = G^2 \frac{1}{2^M} \sum_{a \in \mathcal{A}} |a|^2$$

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

$$P_{\text{err}} \approx e^{-\frac{G^2}{\sigma_0^2}} \approx e^{-3\,\text{SNR}/2^{(M+1)}}$$

QAM, probability of error

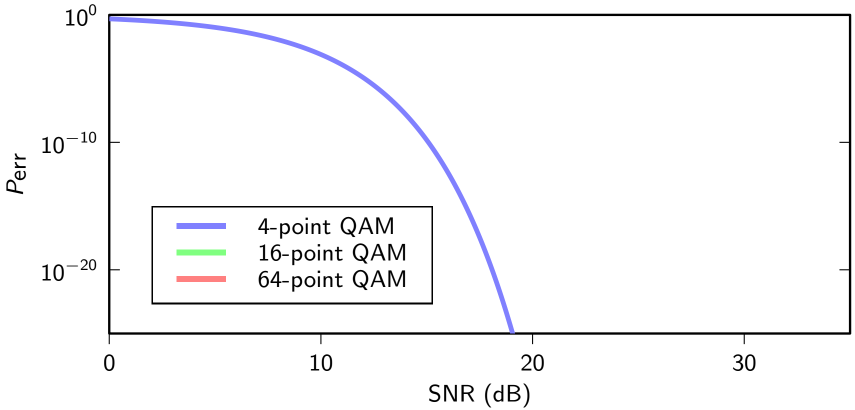
transmitted power (all symbols equiprobable and independent):

$$\sigma_s^2 = G^2 \frac{1}{2^M} \sum_{a \in \mathcal{A}} |a|^2$$

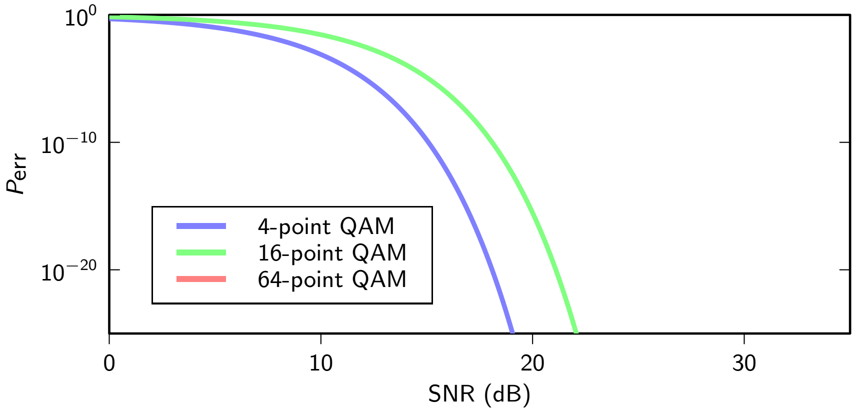
$$= G^2 \frac{2}{3}(2^M - 1)$$

$$P_{\text{err}} \approx e^{-\frac{G^2}{\sigma_0^2}} \approx e^{-3\,\text{SNR}/2^{(M+1)}}$$

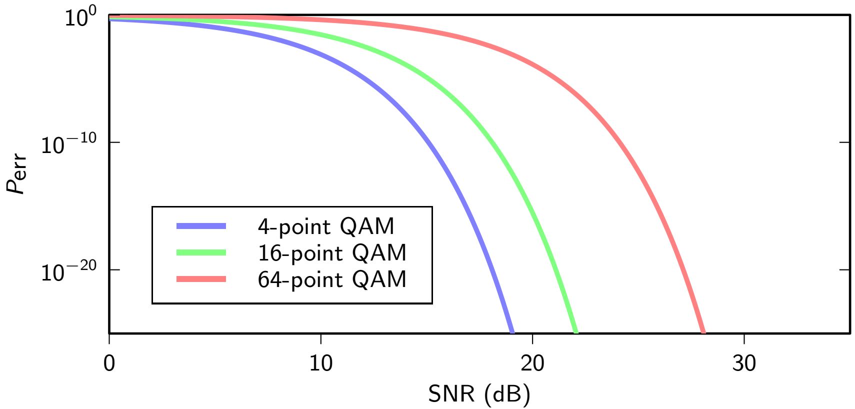
Probability of error



Probability of error



Probability of error



QAM, the recipe

- pick a probability of error you can live with (e.g. 10^{-6})
- find out the SNR imposed by the channel's power constraint
- $M = \log_2\left(1 - \frac{3}{2}\frac{\text{SNR}}{\ln(p_e)}\right)$
- final throughput will be MW

QAM, the recipe

- pick a probability of error you can live with (e.g. 10^{-6})
- find out the SNR imposed by the channel's power constraint
- $M = \log_2 \left(1 - \frac{3}{2} \frac{\text{SNR}}{\ln(p_e)}\right)$
- final throughput will be MW

QAM, the recipe

- pick a probability of error you can live with (e.g. 10^{-6})
- find out the SNR imposed by the channel's power constraint
- $M = \log_2 \left(1 - \frac{3}{2} \frac{\mathrm{SNR}}{\ln(p_e)}\right)$
- final throughput will be MW

QAM, the recipe

- pick a probability of error you can live with (e.g. 10^{-6})
- find out the SNR imposed by the channel's power constraint
- $M = \log_2\left(1 - \frac{3}{2}\frac{\text{SNR}}{\ln(p_e)}\right)$
- final throughput will be MW

QAM

where we stand:

- we know how to fit the bandwidth constraint
- with QAM, we know how many bits per symbol we can use given the power constraint
- we know the theoretical throughput of the transmitter

but how do we transmit *complex* symbols over a real channel?

QAM

where we stand:

- we know how to fit the bandwidth constraint
- with QAM, we know how many bits per symbol we can use given the power constraint
- we know the theoretical throughput of the transmitter

but how do we transmit *complex* symbols over a real channel?

QAM

where we stand:

- we know how to fit the bandwidth constraint
- with QAM, we know how many bits per symbol we can use given the power constraint
- we know the theoretical throughput of the transmitter

but how do we transmit *complex* symbols over a real channel?

QAM

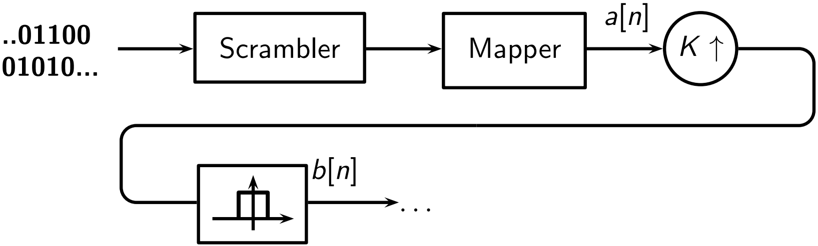
where we stand:

- we know how to fit the bandwidth constraint
- with QAM, we know how many bits per symbol we can use given the power constraint
- we know the theoretical throughput of the transmitter

but how do we transmit *complex* symbols over a real channel?

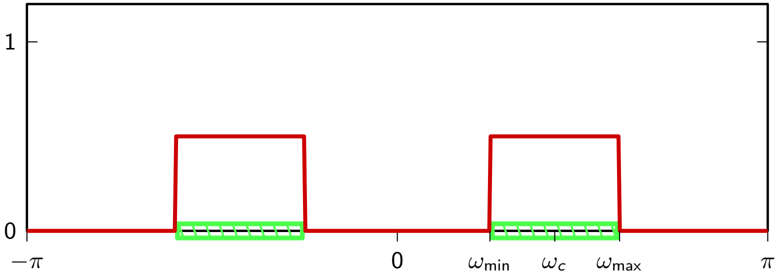
a QAM transmitter

QAM transmitter design

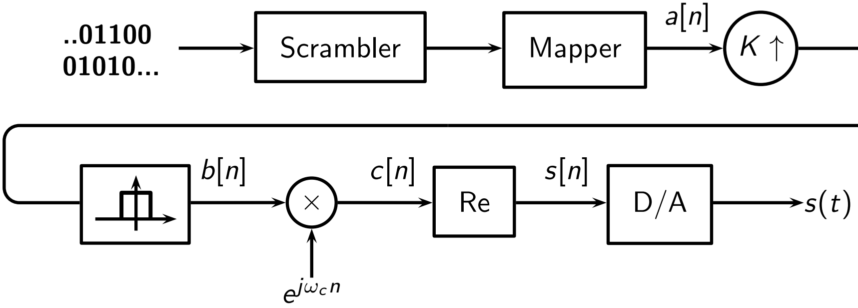


$b[n] = b_r[n] + jb_i[n]$ is a complex-valued baseband signal...

Fulfilling the bandwidth constraint



QAM transmitter, final design



The transmitted passband signal

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

$$\begin{aligned} s[n] &= \mathrm{Re}\{c[n]\} \\ &= \mathrm{Re}\{(b_r[n] + jb_i[n])(\cos\omega_c n + j\sin\omega_c n)\} \\ &= b_r[n]\cos\omega_c n - b_i[n]\sin\omega_c n \end{aligned}$$

The transmitted passband signal

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

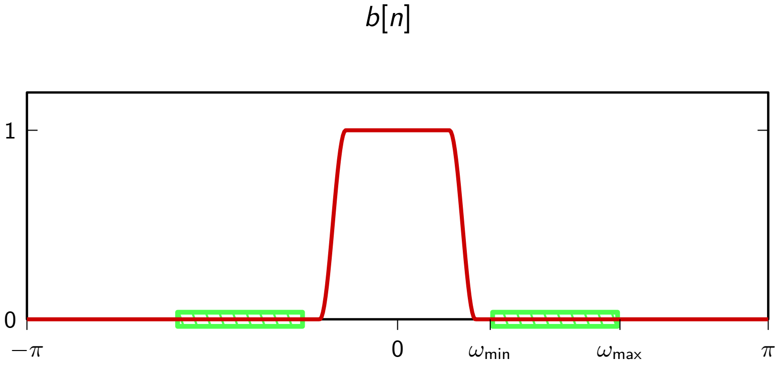
$$\begin{aligned} s[n] &= \mathrm{Re}\{c[n]\} \\ &= \mathrm{Re}\{(b_r[n] + jb_i[n])(\cos\omega_c n + j\sin\omega_c n)\} \\ &= b_r[n]\cos\omega_c n - b_i[n]\sin\omega_c n \end{aligned}$$

The transmitted passband signal

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

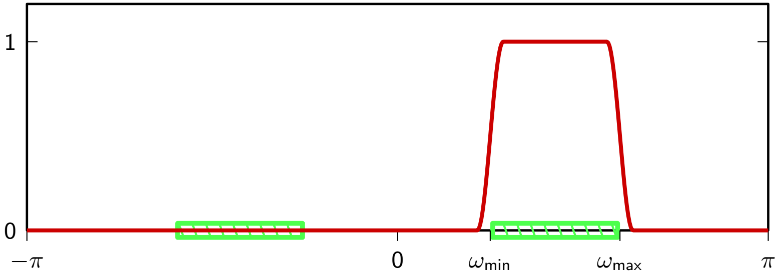
$$\begin{aligned} s[n] &= \mathrm{Re}\{c[n]\} \\ &= \mathrm{Re}\{(b_r[n] + jb_i[n])(\cos\omega_c n + j\sin\omega_c n)\} \\ &= b_r[n]\cos\omega_c n - b_i[n]\sin\omega_c n \end{aligned}$$

From complex baseband to real passband signal



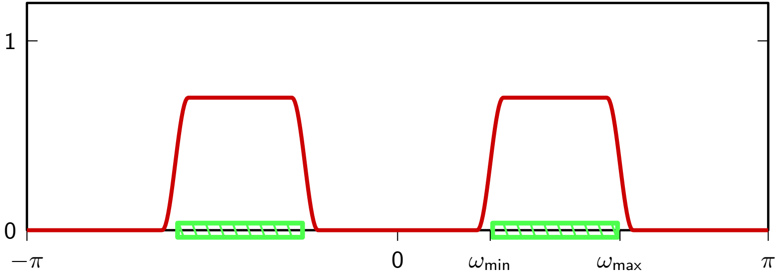
From complex baseband to real passband signal

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



From complex baseband to real passband signal

$$s[n] = \mathrm{Re}\{c[n]\}$$

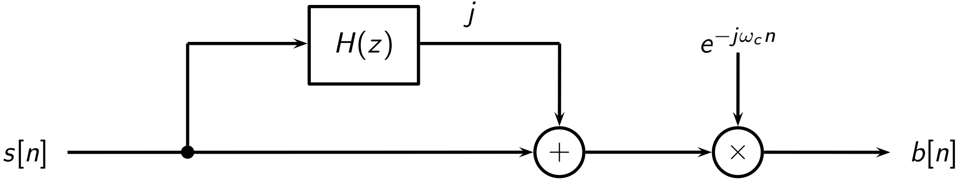


Can we go back?

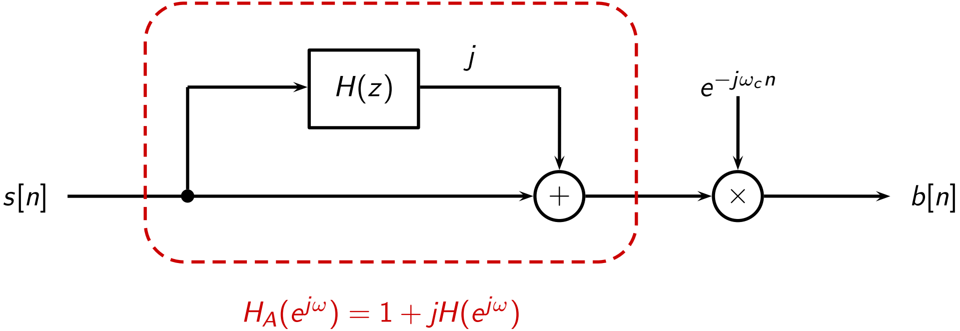
receiver obtains $s[n]$
can we recover the complex baseband $b[n]$ from $s[n]$?

easiest way: Hilbert demodulation

Hilbert demodulation



Hilbert demodulation

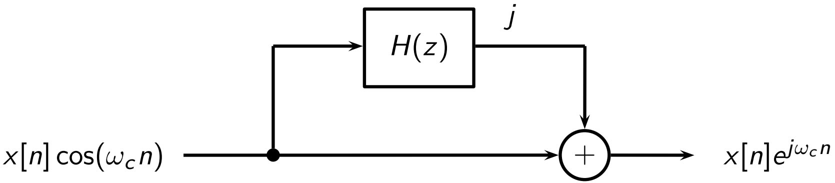


Hilbert demodulation

$$H(e^{j\omega}) = \begin{cases} -j & \omega \geq 0 \\ j & \omega < 0 \end{cases}$$

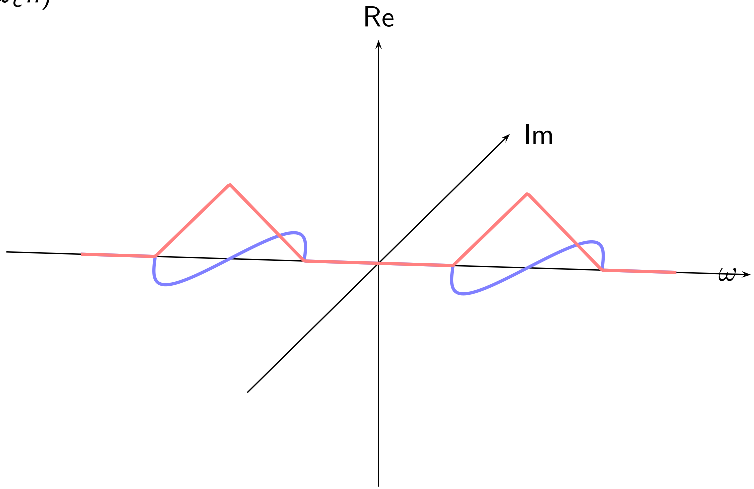
$$H_A(e^{j\omega}) = 1 + jH(e^{j\omega}) = \begin{cases} 2 & \omega \geq 0 \\ 0 & \omega < 0 \end{cases}$$

Hilbert demodulation (recap)



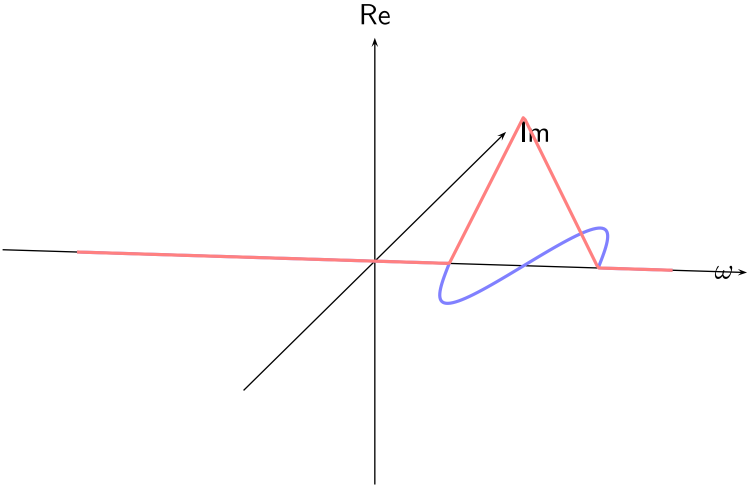
Hilbert demodulation

$y[n] = x[n]\cos(\omega_c n)$

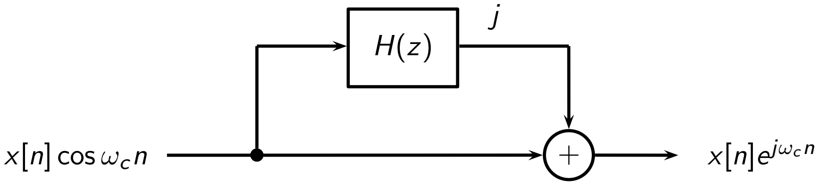


Hilbert demodulation

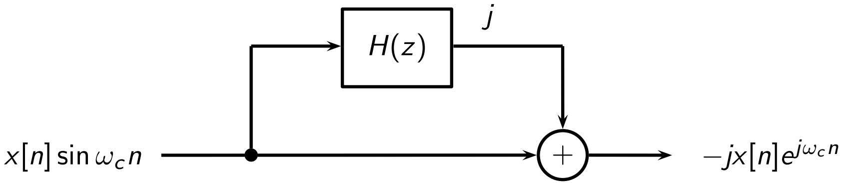
$jy[n] * h[n] + y[n]$



Hilbert demodulation



Hilbert demodulation

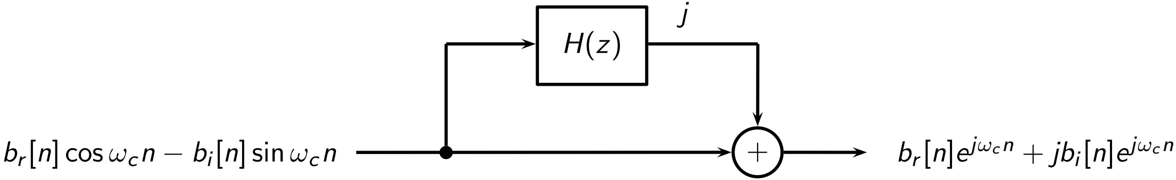


The transmitted passband signal

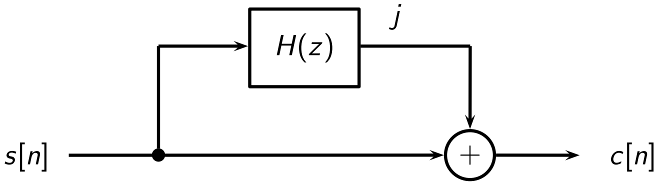
$$
\begin{aligned}
c[n] &= b[n]\, e^{j\omega_c n} \\
&= (b_r[n] + j b_i[n]) e^{j\omega_c n}
\end{aligned}
$$

$$
\begin{aligned}
s[n] &= \mathrm{Re}\{c[n]\} \\
&= b_r[n] \cos \omega_c n - b_i[n] \sin \omega_c n
\end{aligned}
$$

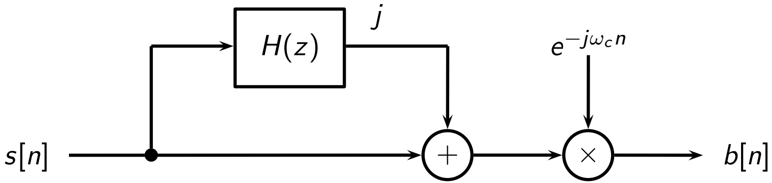
Hilbert demodulation



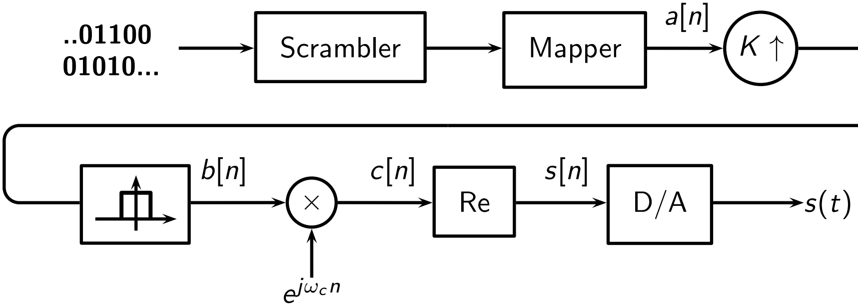
Hilbert demodulation



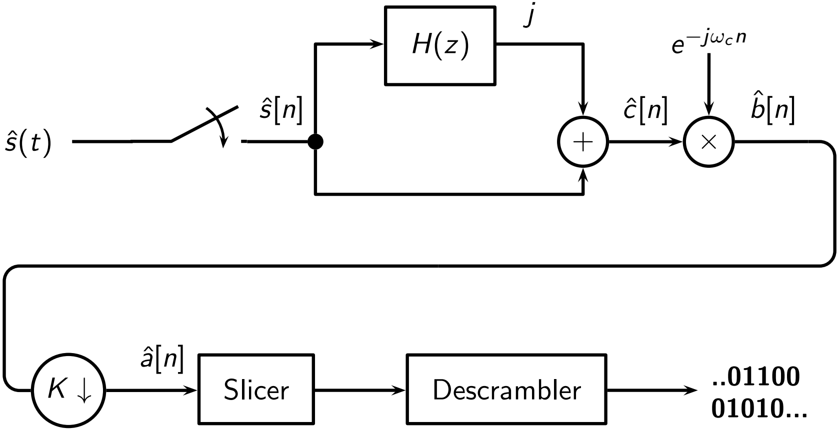
Hilbert demodulation



QAM transmitter, final design



QAM receiver, idealized design



Example: the V.32 voiceband modem

- analog telephone channel: $F_{\text{min}} = 450\text{Hz}$, $F_{\text{max}} = 2850\text{Hz}$
- usable bandwidth: $W = 2400\text{Hz}$, center frequency $F_c = 1650\text{Hz}$
- pick $F_s = 3 \cdot 2400 = 7200\text{Hz}$, so that $K = 3$
- $\omega_c = 0.458\pi$

Example: the V.32 voiceband modem

- analog telephone channel: $F_{\text{min}} = 450\text{Hz}$, $F_{\text{max}} = 2850\text{Hz}$
- usable bandwidth: $W = 2400\text{Hz}$, center frequency $F_c = 1650\text{Hz}$
- pick $F_s = 3 \cdot 2400 = 7200\text{Hz}$, so that $K = 3$
- $\omega_c = 0.458\pi$

Example: the V.32 voiceband modem

- analog telephone channel: $F_{\text{min}} = 450\text{Hz}$, $F_{\text{max}} = 2850\text{Hz}$
- usable bandwidth: $W = 2400\text{Hz}$, center frequency $F_c = 1650\text{Hz}$
- pick $F_s = 3 \cdot 2400 = 7200\text{Hz}$, so that $K = 3$
- $\omega_c = 0.458\pi$

Example: the V.32 voiceband modem

- analog telephone channel: $F_{\text{min}} = 450\text{Hz}$, $F_{\text{max}} = 2850\text{Hz}$
- usable bandwidth: $W = 2400\text{Hz}$, center frequency $F_c = 1650\text{Hz}$
- pick $F_s = 3 \cdot 2400 = 7200\text{Hz}$, so that $K = 3$
- $\omega_c = 0.458\pi$

Example: the V.32 voiceband modem

- maximum SNR: 22dB
- pick $P_{\text{err}} = 10^{-6}$
- using QAM, we find

$$M = \log_2\left(1 - \frac{3}{2}\frac{10^{22/10}}{\ln(10^{-6})}\right) \approx 4.1865$$

  so we pick $M = 4$ and use a 16-point constellation
- final data rate is $WM = 9600$ bits per second

Example: the V.32 voiceband modem

- maximum SNR: 22dB
- pick $P_{\text{err}} = 10^{-6}$
- using QAM, we find

$$M = \log_2\left(1 - \frac{3}{2}\frac{10^{22/10}}{\ln(10^{-6})}\right) \approx 4.1865$$

so we pick $M = 4$ and use a 16-point constellation

- final data rate is $WM = 9600$ bits per second

Example: the V.32 voiceband modem

- maximum SNR: 22dB
- pick $P_{\text{err}} = 10^{-6}$
- using QAM, we find

$$M = \log_2 \left(1 - \frac{3}{2} \frac{10^{22/10}}{\ln(10^{-6})} \right) \approx 4.1865$$

  so we pick $M = 4$ and use a 16-point constellation

- final data rate is $WM = 9600$ bits per second

Example: the V.32 voiceband modem

- maximum SNR: 22dB
- pick $P_{err} = 10^{-6}$
- using QAM, we find

$$M = \log_2\left(1 - \frac{3}{2}\frac{10^{22/10}}{\ln(10^{-6})}\right) \approx 4.1865$$

  so we pick $M = 4$ and use a 16-point constellation
- final data rate is $WM = 9600$ bits per second

Theoretical channel capacity

- we used very specific design choices to derive the throughput
- what is the best one can do?
- Shannon's capacity formula is the upper bound

$$C = W \log_2 (1 + SNR)$$

- for instance, for the previous example $C \approx 17500$ bps
- the gap can be narrowed by more advanced coding techniques

Theoretical channel capacity

- we used very specific design choices to derive the throughput
- what is the best one can do?
- Shannon's capacity formula is the upper bound

$$C = W \log_2(1 + SNR)$$

- for instance, for the previous example $C \approx 17500$ bps
- the gap can be narrowed by more advanced coding techniques

Theoretical channel capacity

- we used very specific design choices to derive the throughput
- what is the best one can do?
- Shannon's capacity formula is the upper bound

$$C = W \log_2 (1 + SNR)$$

- for instance, for the previous example $C \approx 17500$ bps
- the gap can be narrowed by more advanced coding techniques

Theoretical channel capacity

- we used very specific design choices to derive the throughput
- what is the best one can do?
- Shannon's capacity formula is the upper bound

$$C = W \log_2 (1 + SNR)$$

- for instance, for the previous example $C \approx 17500$ bps
- the gap can be narrowed by more advanced coding techniques

Theoretical channel capacity

- we used very specific design choices to derive the throughput
- what is the best one can do?
- Shannon's capacity formula is the upper bound

$$C = W \log_2 (1 + SNR)$$

- for instance, for the previous example $C \approx 17500$ bps
- the gap can be narrowed by more advanced coding techniques