Lecture 5: Sampling, Interpolation, and Pulse Code Modulation

John M Pauly

February 3, 2026

Baseband Digital Communications

Digital communications *without* a carrier

Phone lines, ethernet, USB, etc

How do you send digital data over a link?

- Sampling and interpolation
- Practical implementation
- Quantization and bandwidth

The phone system developed Pulse Code Modulation (PCM)

Many variants for efficiency, bandwidth, and reliability

Sampling Theorem

Sampling theorem: a signal $g(t)$ with bandwidth $< B$ can be reconstructed *exactly* from samples taken at any rate $R > 2B$.

Sampling can be achieved mathematically by multiplying by an impulse train. The unit impulse train is defined by

$$\text{III}(t) = \sum_{n=-\infty}^{\infty} \delta(t-k)$$

The unit impulse train is also called the III or comb function.

Sampling a signal $g(t)$ uniformly at intervals T_s yields

$$\overline{g}(t) = g(t)\,\text{III}(t) = \sum_{n=-\infty}^{\infty} g(t)\delta(t-nT_s) = \sum_{n=-\infty}^{\infty} g(nT_s)\delta(t-nT_s)$$

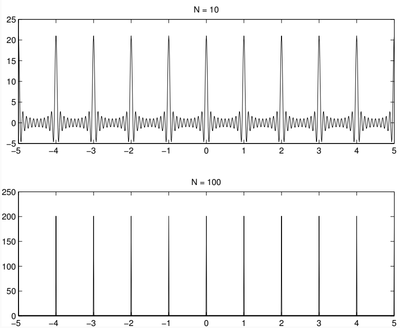
Only information about $g(t)$ at the sample points is retained.

Fourier Transform of $\text{III}(t)$

Fact: the Fourier transform of $\text{III}(t)$ is $\text{III}(f)$.

$$\mathcal{F}\{\text{III}(t)\} = \sum_{n=-\infty}^{\infty} \mathcal{F}\{\delta(t-n)\} = \sum_{n=-\infty}^{\infty} e^{-j2\pi nf} = \sum_{n=-\infty}^{\infty} e^{j2\pi nf} = \text{III}(f)$$

The complex exponentials cancel at noninteger frequencies and add up to an impulse at integer frequencies.



Fourier Transform of Sampled Signal

The impulse train $\text{III}(t/T_s)$ is periodic with period T_s and can be represented as the sum of complex exponentials of all multiples of the fundamental frequency:

$$\text{III}(t/T_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} e^{j2\pi n f_s t} \quad (f_s = \frac{1}{T_s})$$

Thus

$$\overline{g} = g(t)\,\text{III}(t/T_s) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} g(t) e^{j2\pi n f_s t}$$

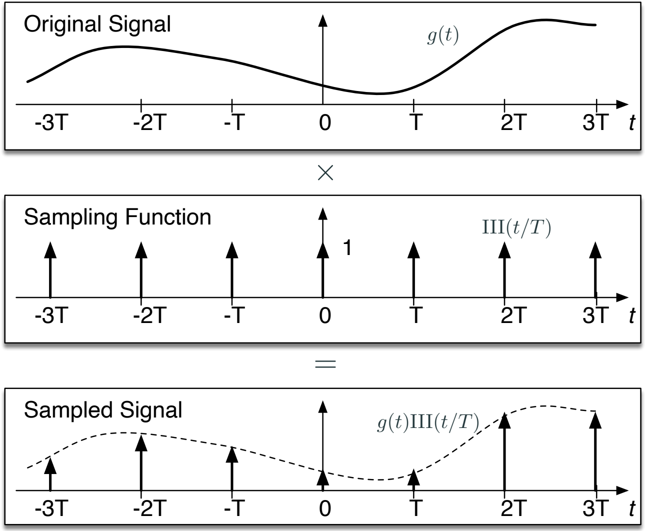
and by the frequency shifting property

$$\overline{G}(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} G(f - n f_s)$$

This sum of shifts of the spectrum can be written as $\text{III}(f/f_s) * G(f)$.

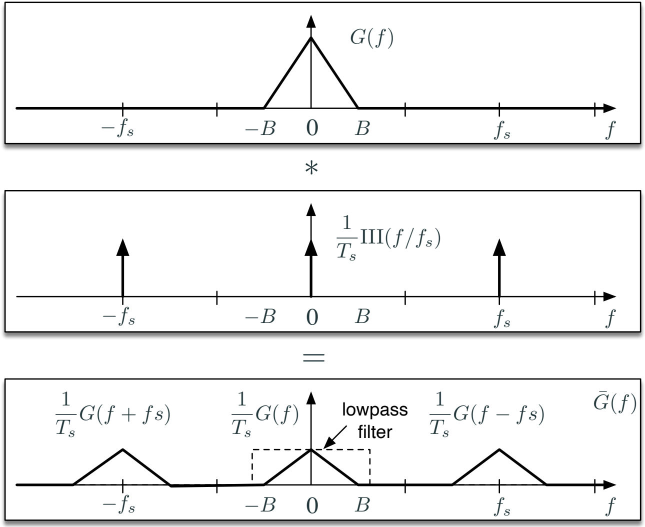
Sampled Signal and Fourier Transform

In the time domain sampling is multiplication by an impulse train



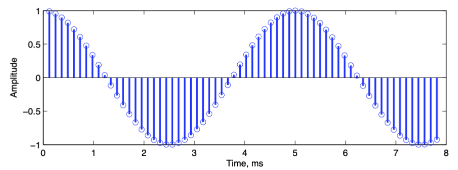
Sampled Signal and Fourier Transform

In the frequency domain sampling is convolution by an impulse train

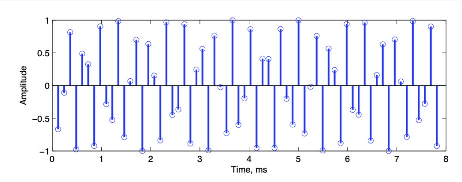


Sampled Cosine Examples

Sometimes it is easy to identify a cosine from its samples

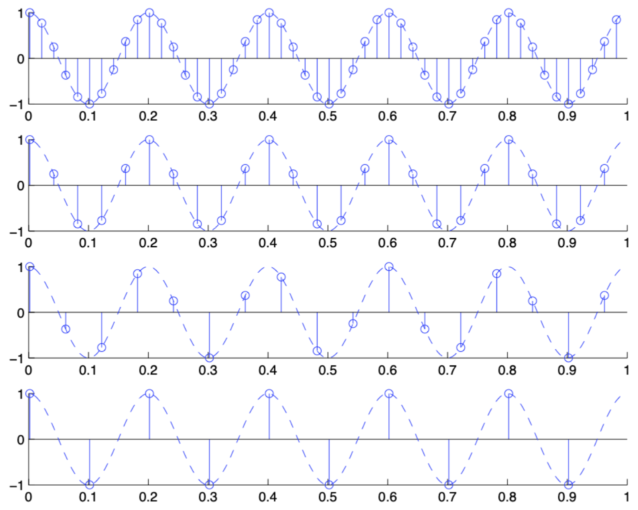


Sometimes it isn't so obvious!



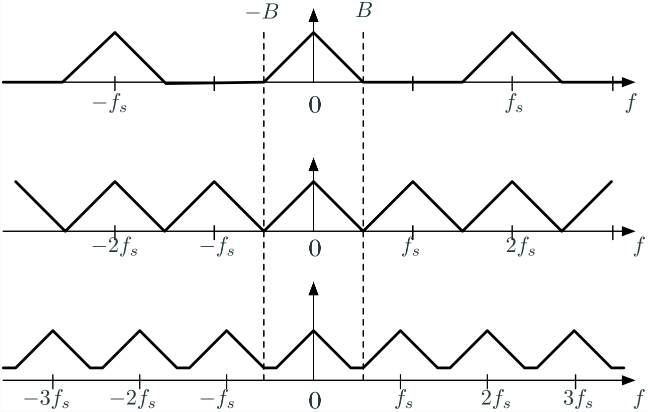
The fact that a signal is bandlimited is a very powerful constraint.

Sampling Examples



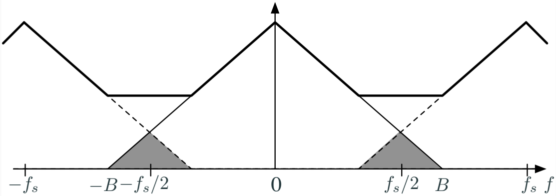
Aliasing and the Minimum Sampling Rate

When the sampling rate is too low, the spectral replicas overlap



This is called aliasing.

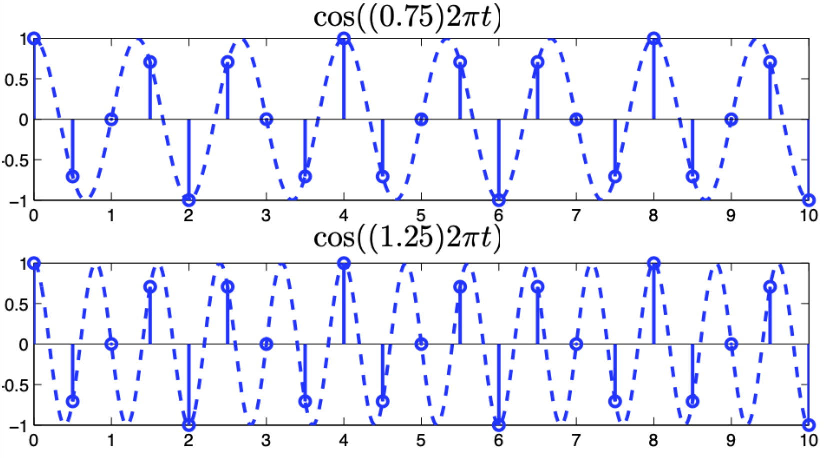
Aliasing



- The shaded frequencies overlap and are ambiguous.
- High positive frequencies wrap around to high negative frequencies
- What signal would you reconstruct if you assumed the signal was actually band limited?

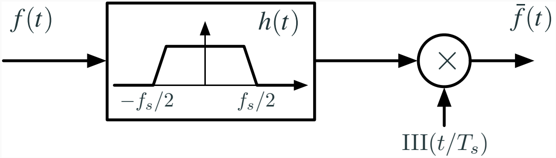
Example of Aliasing

Cosines at frequencies of 0.75 Hz and 1.25 Hz produce exactly the same samples at a sampling rate of 1 Hz



Anti-Aliasing Filter

In practice, a sampler is always preceded by a filter to limit the signal bandwidth to match the sampling rate

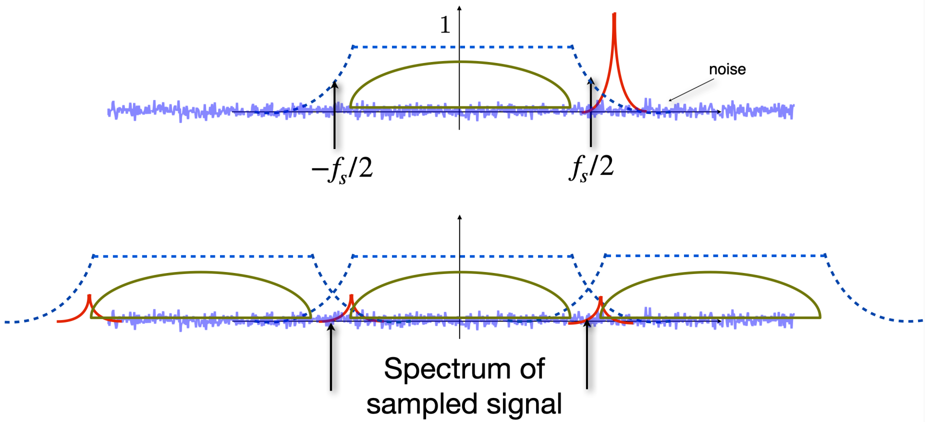


This may delete part of the signal if it isn't bandlimited.

It ensures that the signal that is sampled is bandlimited.

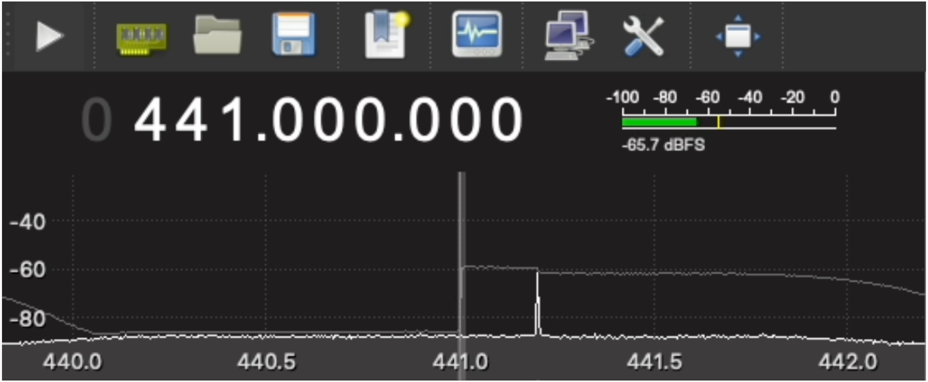
Anti-Aliasing Filter Noise Suppression

The anti-aliasing filter also suppresses out of band noise and interference



Anti-Aliasing Filter on your RTLSDR

Your RTLSDR only has about 10 dB attenuation at the band edge



Reconstruction from Uniform Samples (Ideal)

If sample rate $1/T_s$ is greater than $2B$, shifted copies of spectrum do not overlap, so low pass filtering recovers original signal.

Cutoff frequency of low pass filter should satisfy

$$B \le f_c \le f_s - B$$

Suppose $f_c = B$. A low pass filter with gain T_s has transfer function and impulse response

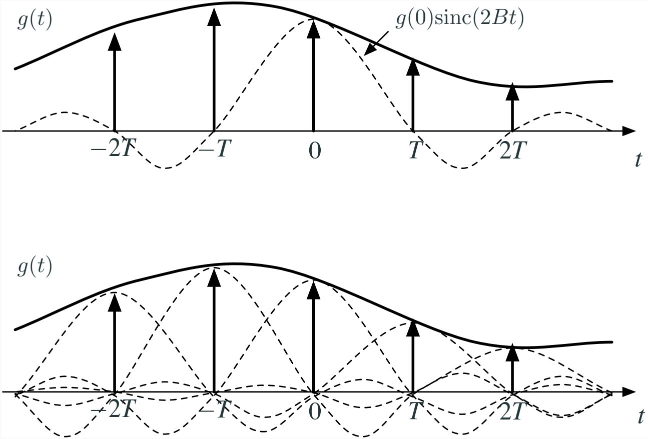
$$H(f) = T_s \Pi\left(\frac{f}{2B}\right), \quad h(t) = 2BT_s \mathrm{sinc}(2Bt)$$

Then if $T_s = 1/2B$

$$\begin{aligned} h(t) * \overline{g}(t) &= \sum_{n=-\infty}^{\infty} h(t) * g(nT_s)\delta(t - nT_s) \\ &= \sum_{n=-\infty}^{\infty} g(nT_s)\mathrm{sinc}(2B(t - nT_s)) \end{aligned}$$

Ideal Interpolation

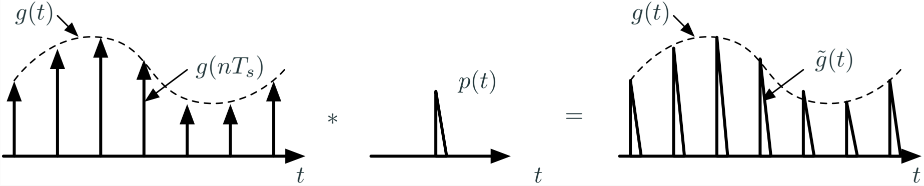
Ideal interpolation represents a signal as sum of shifted sincs.



Practical Interpolation

In practice, the sampled signal is a sum of pulses, not impulses.

$$
\begin{aligned}
\tilde{g}(t) &= \sum_{n=-\infty}^{\infty} g(nT_s)p(t - nT_s) \\
&= p(t) * \sum_{n=-\infty}^{\infty} g(nT_s)\delta(t - nT_s) = p(t) * \overline{g}(t)
\end{aligned}
$$



Practical Interpolation (cont.)

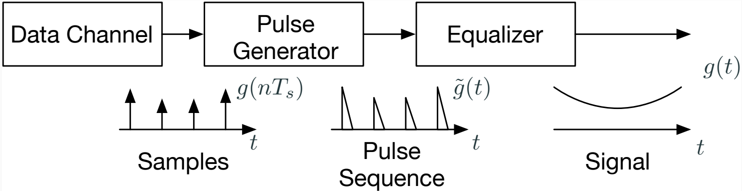
By the convolution theorem,

$$\tilde{G}(f) = P(f) \cdot \frac{1}{T_s} \sum_{n=-\infty}^{\infty} G(f - nf_s)$$

We can recover $G(f)$ from $\tilde{G}(f)$ by low pass filtering to eliminate high frequency shifts and *equalizing* by inverting $P(f)$.

$$E(f) = \begin{cases} T_s/P(f) & |f| < B \\ 0 & |f| > B \end{cases}$$

The transfer function $E(f)$ should not be close to 0 in the pass band.

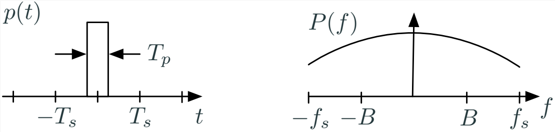


Practical Interpolation (cont.)

Example: rectangular pulses with $T_p < T_s < 1/2B$.

$$p(t) = \Pi\left(\frac{t}{T_p}\right) \implies P(f) = T_p \mathrm{sinc}(T_p f)$$

This looks like:

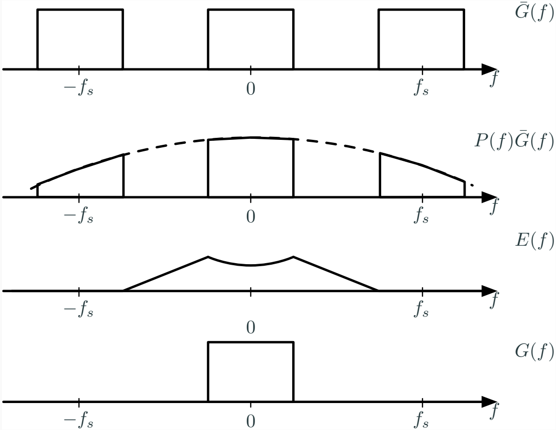


The equalizer should undo the spectral weighting of $P(f)$. The transfer function for the equalizer should satisfy

$$E(f) = \begin{cases} T_s/P(f) & |f| < B \\ \mathrm{don't care} & B < |f| < f_s - B \\ 0 & |f| > f_s - B \end{cases}$$

Practical Interpolation (cont.)

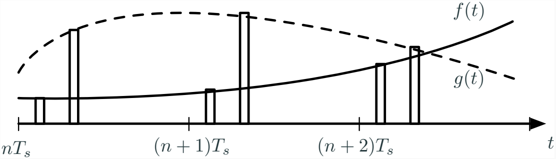
To illustrate, we'll assume that the spectrum $G(f)$ is flat. Then



The equalizer $E(f)$ both corrects for the apodization of $P(f)$, and performs the lowpass filtering for optimal interpolation.

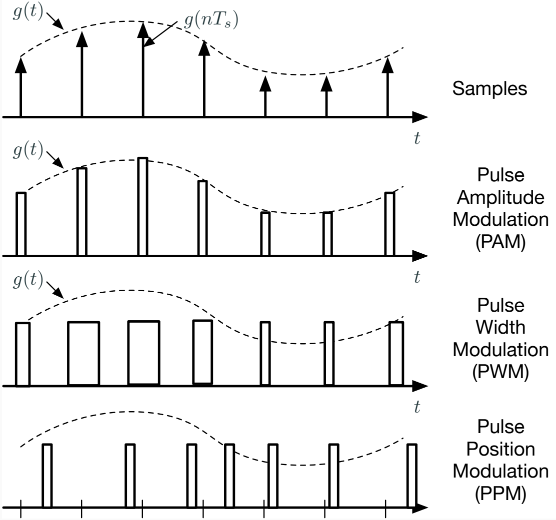
Pulse Modulation of Signals

- In many cases, bandwidth of communication link is much greater than signal bandwidth.
- The signal can be transmitted using short pulses with low duty cycle:
    - Pulse amplitude modulation: width fixed, amplitude varies
    - Pulse width modulation: position fixed, width varies
    - Pulse position modulation: width fixed, position varies
- All three can be time-division multiplexied on a single channel



- Each signal gets its own time slot

PAM, PWM, PPM: Amplitude, Width, Position



Pulse Amplitude Modulation

- The input to a pulse amplitude modulator is the real-world sample of $g(t)$:

  $$g_1(nT_s) = \int_0^{T_s} q(t)g(t - nT_s)\ dt$$

  where $q(t)$ is an integrator function. (Width of $q(t)$ should be $\ll T_s$.)
- Each transmitted pulse is narrow with height (or area) proportional to $g_1(nT_s)$. The pulse is integrated to obtain an analog value.

  $$\tilde{g}(nT_s) = \int_0^{T_p} q_1(t)g_1(t - nT_s)\ dt$$

  where $T_p \ll T_s$
- The original signal $g(t)$ is reconstructed using an equalizer and a low pass filter, as discussed above.

Pulse Width Modulation (PWM)

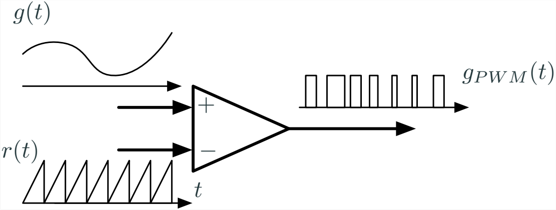
Pulse width modulation is also called pulse duration modulation (PDM).

PWM is more often used for control than for communication

- Motors and servos
- LEDs: output limunosity is proportional to average current.
- Amplifiers

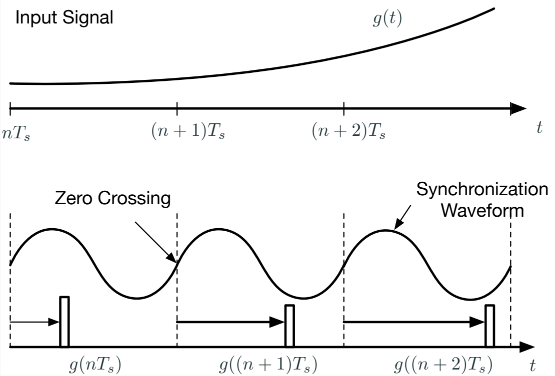
Signal can be recovered exactly if $2B < 0.637 f_s$

Generating the PWM signal can be done simply with a comparator



Pulse Position Modulation (PPM)

The value of the signal determines the delay of the pulse from the clock.



Very common in home automation systems.

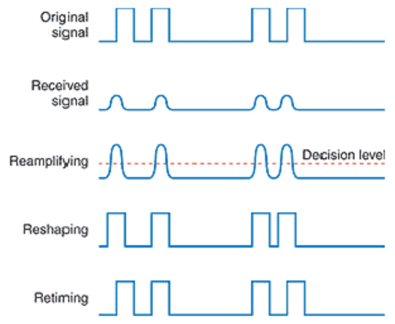
Microcontrollers can generate PPM (and PWM) in software. Doesn't require an D/A.

Simple Baseband Digital Communications

- Pulse Code Modulation (PCM)
- Quantization
    - Uniform Quantization
    - Non-Uniform Quantization
- Quantization Error
- PCM Bandwidth
- PCM SNR

Analog vs. Digital Communication

- Analog communication (baseband and modulated) is subject to noise.
- Pulse modulations (PAM, PWM, PPM) represent analog signals by analog variations in pulses and are also sunbject to noise.
- Long distance communication requires repeaters, which amplify signal and noise. Each link adds noise.
- Digital communication suppresses noise by regenerating signal.

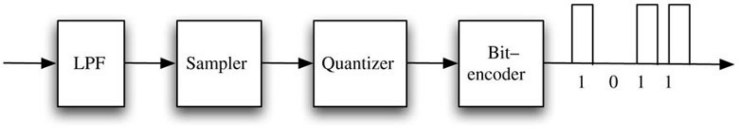


Pulse Code Modulation (PCM)

Pulse Code Modulation (PCM) was developed by the phone companies to transmit digital data

In PCM, signal samples are first quantized to a limited number of bits

These bits are the *serialized* and sent out as a sequence of binary pulses



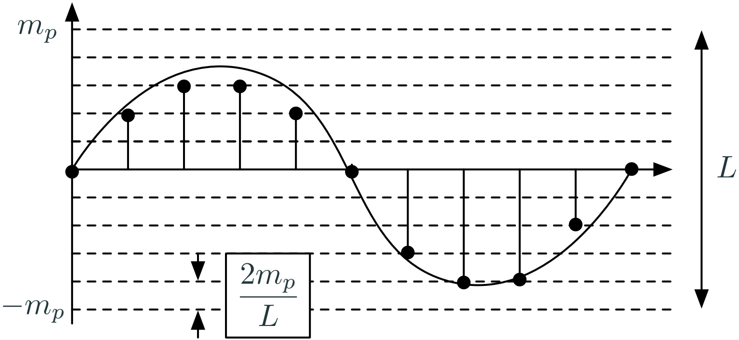
If m bits are used, then 2^m signal values can be represented

- unsigned linear: $0, \Delta\nu, \dots, \Delta\nu(2^m - 1)$
- two's complement: $-\Delta\nu 2^{m-1}, \dots, \Delta\nu(2^{m-1} - 1)$
- A-law and μ-law: approximately logarithmic (more dynamic range)

PCM and Quantization

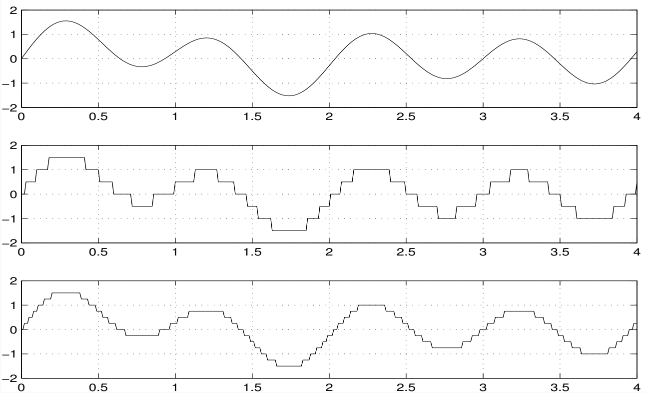
Quantization of a signal produces the closest representable value.

For fixed number of values, spacing between values increases with range.



PCM Tradeoffs

- Signal bandwidth determines minimum sample rate
- Desired signal fidelity determines precision of reproduced signal
- Signals can be quantized using digital-to-analog converter (DAC)

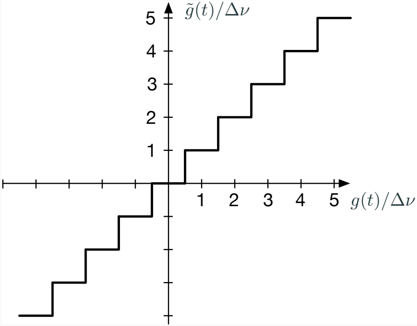


Uniform Quantization

An ideal uniform quantizer is a nonlinear time invariant system.

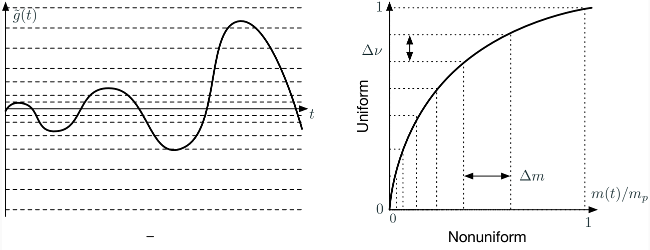
If there are L = 2N+1 levels of width $\Delta\nu$,

$$\widetilde{g}(t) = \begin{cases} -N\Delta\nu & g(t) < -N\Delta\nu \\ n\Delta\nu & (n-\frac{1}{2})\Delta\nu < g(t) < (n+\frac{1}{2})\Delta\nu \\ N\Delta\nu & g(t) > N\Delta\nu \end{cases}$$

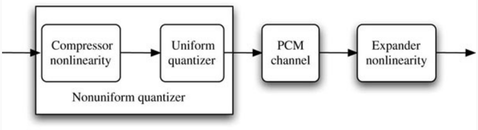


Nonuniform Quantization

Nonuniform quantizers increase quantization intervals as magnitude of value. Interval proportional to value implies logarithmic curve.



An analog compressor (semiconductor diode) can be used.


Compressor nonlinearity

Uniform quantizer

PCM channel

Expander nonlinearity

Nonuniform quantizer

Quantization Error

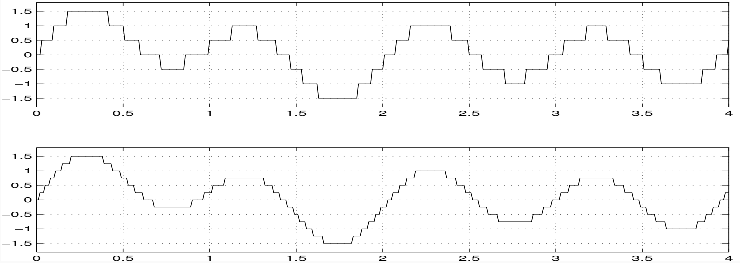
Uniform quantization with L levels of a signal with peak amplitude m_p has maximum quantization error

$$\text{max error} = \frac{m_p}{L},$$

and mean square error

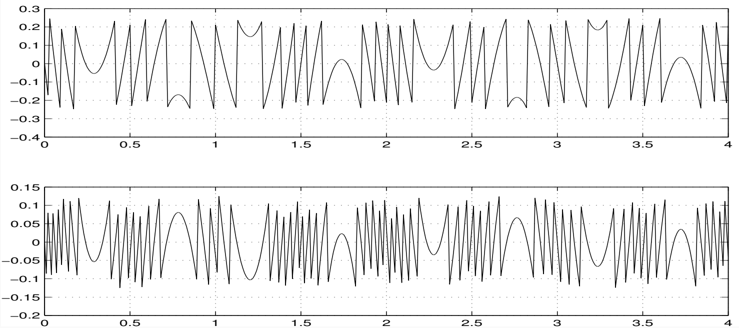
$$\text{average square error} = \frac{m_p^2}{3L^2}$$

Example: signal quantized to 4 and 16 levels.



Quantization Error (cont.)

Quantization error for quantizing to 4 and 16 levels.



Power of quantization error for above example is

$$0.03125 \approx \tfrac{1}{3}\left(\frac{1}{4}\right)^2 \text{ for L} = 4 \text{ and } 0.0013 \approx \tfrac{1}{3}\left(\frac{1}{16}\right)^2 \text{ for L=16}$$

Bandwidth vs. Quantization Error

What bandwidth is needed to transmit a PCM encoded signal?

Example: suppose that we want maximum error $0.5\% m_p$ for a 3 kHz signal.

The maximum error is

$$\frac{\Delta\nu}{2} = \frac{1}{2}\left(\frac{2m_p}{L}\right) = \frac{m_p}{L}$$

We want this to be $0.5\% m_p$,

$$\frac{m_p}{L} = 0.005\ m_p$$

which means $L = \frac{1}{0.005} = 200$.

We would round this up to 256, and use 8 bits per sample.

The number of samples we need per second is given by the Nyquist rate.

With a signal bandwidth of ± 3000 Hz we need a sampling rate of

$$R_N = 2 \cdot 3000 = 6000 \text{ Hz}$$

Each sample requires 8 bits, so we need

$$6000 \cdot 8 = 48000 \text{ bits/sec}$$

- We will see that a bandlimited signal can convey two symbols per Hz.
- For binary PCM, we need $48000/2 = 24000$ Hz.
- For practical reasons, we sample faster than the Nyquist rate. E.g., at rate 4000 Hz, the required bandwidth is 32 kHz.

PCM SNR

The signal-to-noise ratio is

$$SNR = \frac{\text{average signal power}}{\text{average noise power}}$$

The noise could be from many sources. For uniform quantization noise, the *Signal to Quantization Noise (SQNR)* is

$$\text{average signal power} \approx am_p^2 \quad (a \approx \frac{1}{2})$$

$$\text{quantization error} \approx \frac{1}{3}(m_p/L)^2$$

$$\text{SQNR} \approx \frac{3}{2}L^2 = \frac{3}{2}(2^m)^2 = \frac{3}{2}2^{2m}$$

where m is the number of bits in the PCM sample, so $L = 2^m$.

SQNR grows exponentially with the number of bits.

PCM SNR

If we measure SNR in dB,

$$\begin{aligned} SQNR &= 10\log_{10}\left(\frac{3}{2}2^{2m}\right) = 10\log_{10}\left(\frac{3}{2}\right) + 20m\log_{10}(2) \\ &= 1.76 + 6.02m \ \ \mathrm{dB} \end{aligned}$$

Increasing m by one bit improves SNR by 6 dB! One bit quadruples SQNR.

PCM SNR

Consider two cases for a 4 kHz bandwidth signal

- $L = 64$, $m = 6$ bits

$$SNR_{dB} = 1.76 + 36.12 = 37.88 \text{ dB}$$

- $L = 256$, and $m = 8$ bits

$$SQNR = 1.76 + 48.16 = 49.92 \text{ dB}$$

We've gained 12 dB in SNR, while the bandwidth has increased only from

$$(2 \times 4 \text{ kHz})(6 \text{ bits})/2 = 24 \text{ kHz}$$

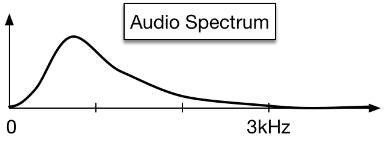
to

$$(2 \times 4 \text{ kHz})(8 \text{ bits})/2 = 32 \text{ kHz}$$

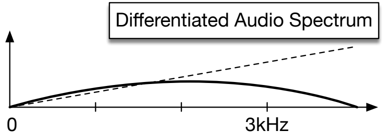
We only need 1/3 greater bandwidth for a 12 dB improvements in SQNR.

Differential PCM

PCM uses a lot of bits per second. Audio has a large dynamic range, and is weighted towards lower frequencies. Samples are highly correlated.



We can improve PCM by transmitting samples of $\frac{d}{dt}m(t)$, and then reconstructing the $m(t)$ at the other end by integration



This is *differential* PCM.

Differential PCM

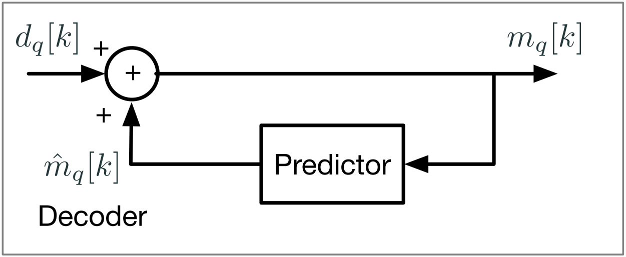
In general, we want to use previous samples to predict the signal.

- $m[k]$ is the sampled unquantized signal we want to transmit, and $m_q[k]$ is the quantized signal
- $\hat{m}[k]$ is the prediction of $m[k]$ based on previous samples $m_q[k]$
- $d[k] = m[k] - \hat{m}[k]$, the unquantized difference between $m[k]$ and the prediction
- $d_q[k]$ is the actual quantized difference that will be transmitted.

A predictor will take previous samples of the signal, and compute the next expected sample

Differential PCM

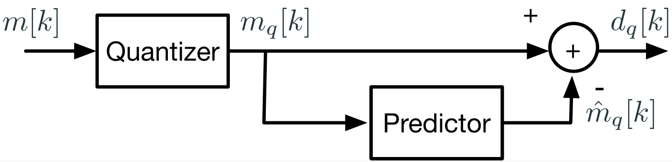
We'll start with the receiver



- We'll assume that the system is working, and has produced previous values of $m_q[k]$
- The predictor computes $\hat{m}_q[k]$, the next predicted sample
- The input is the quantized difference signal $d_q[k]$
- This is added to the predicted signal $\hat{m}_q[k]$ to recover $m_q[k]$.

Differential PCM

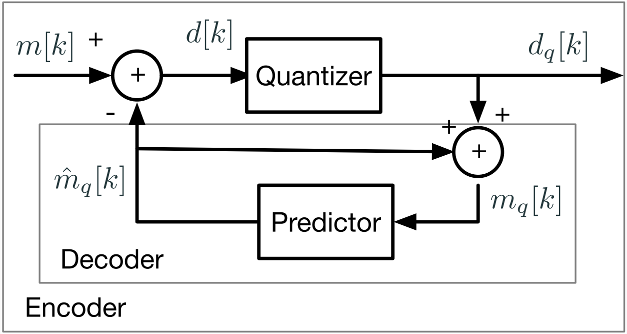
A simple encoder would quantize first, compute the prediction, and then compute the difference signal to transmit



- The problem with this is that quantization errors can build up
- This can lead to larger errors.
- We can do much better!

Differential PCM

However, we can do better with this encoder



- This uses the actual decoder to compute the prediction
- It computes the difference between the *unquantized* signal and the *quantized* prediction.

Differential PCM

In the simplest case, the predictor is just the previous value of the signal

$$\hat{m}[k] = m[k-1]$$

This works pretty well, and can greatly improve the dynamic range. This is called *Delta Modulation*, and we'll come back to this shortly.

The next version uses a locally linear approximation using the first difference as an approximation to the derivative

$$\hat{m}[k] = m[k-1] + (m[k-1] - m[k-2])$$

In general, for any expected signal spectra, we can solve for an FIR filter

$$\hat{m}[k] = \sum_{n=1}^{N} a_n m[k-n]$$

that minimizes the prediction error, and the amount of data we need to transmit.

Adaptive Differential PCM (ADPCM)

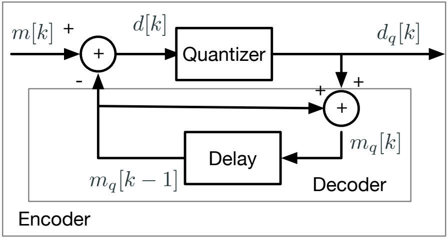
- The DPCM signal will be much lower amplitude if it is a good model
- The number of quantization levels L is fixed
- ADPCM adaptively adjusts the channel gain, so that the quantization levels best represent the signal.

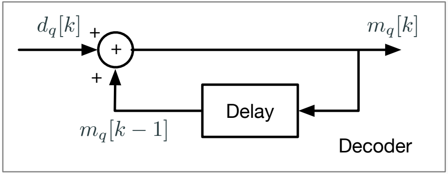
The combination of DPCM and ADPCM can reduce the number of bits required by a factor of two,

This can be used to reduce the bandwidth required by a factor of two, or improve the SNR for a fixed bandwidth.

Delta Modulation

DPCM is much simpler if we just use the previous sample for the predictor. The predictor is just a delay.

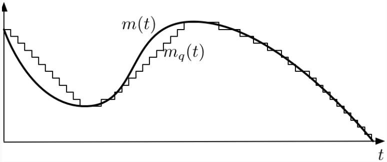




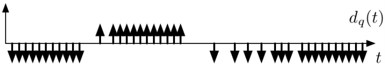
We can make up for the simple predictor by increasing the sampling rate. This increases the similarity between adjacent samples, and reduces the number of bits we need for $d_q[k]$.

Sigma-Delta Modulation

If the predictor is accurate enough we only need one bit for the error.



The waveform we need to transmit 0, 1, and -1,



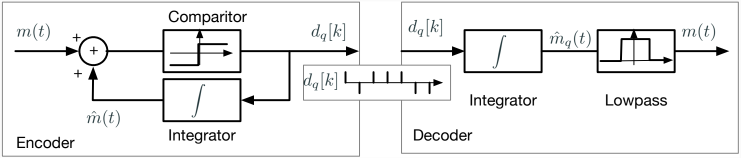
The output waveform is simple the sum of the transmitted errors. Fidelity is limited by the derivative of the signal.

Faster sampling improves fidelity. Adjacent samples are highly correlated.

Typical numbers might be 8 times the Nyquist rate, or more.

Sigma-Delta Modulation

The encoder and decoder are particularly simple



- Result can be very high fidelity if the sampling rate is high enough
- Widely used in consumer electronics, and elsewhere
- This is why you see "1 bit A/D" stickers on products, and it is actually a good thing.
- Similar number of bits per second, but much simpler hardware

PCM Extensions

- PCM does not efficiently use the signal bandwidth or quantization levels.
- DPCM exploits the redundancies in the signal to reduce the amount of information that needs to be transmitted
- ADPCM adjusts the quantization level to best use the quantizer dynamic range.
- If the sampling time is fast enough, the ADPCM signal is only ± 1 bit. This reduces to the very simple sigma-delta modulation,
- More detailed models of the vocal tract (vocoders) further reduce the amount of data we need to transmit. These are currently used in every cell phone and digital radio today.