From my Microphone to the Ether

An example-based approach to a bit of math

Marcus Müller

Software Defined Radio Academy 2016

Who am I?

- All-purpose SDR nut
- GNU Radio contributor and user
- ...who was a bit overly present on the `discuss-gnuradio@gnu.org` mailing list
- Got hired by Ettus

Who am I?

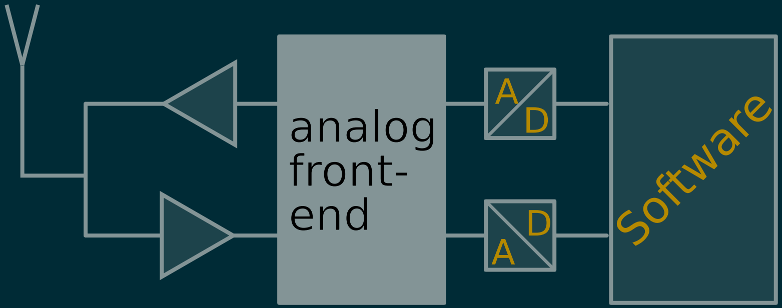
- All-purpose SDR nut
- GNU Radio contributor and user
- ...who was a bit overly present on the `discuss-gnuradio@gnu.org` mailing list
- Got hired by Ettus

...and who is Ettus Research ?

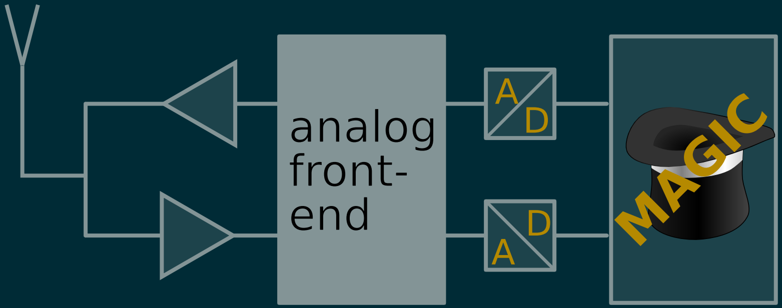
- Producer of the USRP series of SDR frontends
- `gr-uhd` integrates directly in GNU Radio
- `http://www.ettus.com`
- mostly directly mixing complex baseband receivers, but many can be used in Low-IF and direct sampling modes!

A short overview

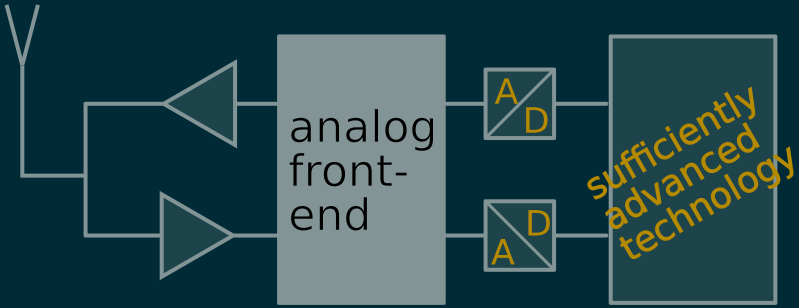
What are we after?



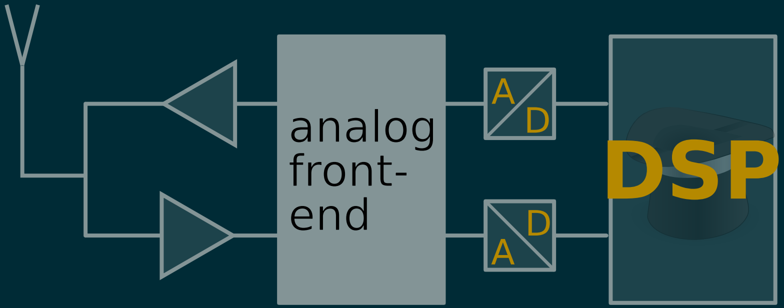
What are we after?



What are we after?



What are we after?



Understanding Signals in the Frequency Domain

Fourier states:

> Every sufficiently well-behaved[1] signal can be reproduced to an arbitrary amount of precision by combining harmonic functions

Bonus: if they are periodic, it's only a discrete set of harmonics!

[1] i.e. the signals we care about

Understanding Signals in the Frequency Domain

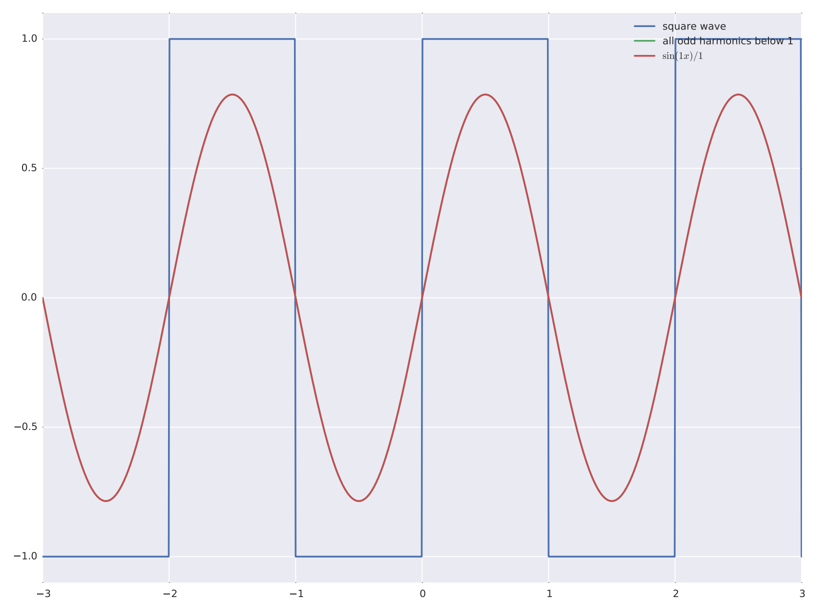
Fourier states:

$\boxed{\text{Every sufficiently well-behaved[1] signal can be reproduced to an arbitrary amount of precision by combining harmonic functions}}$

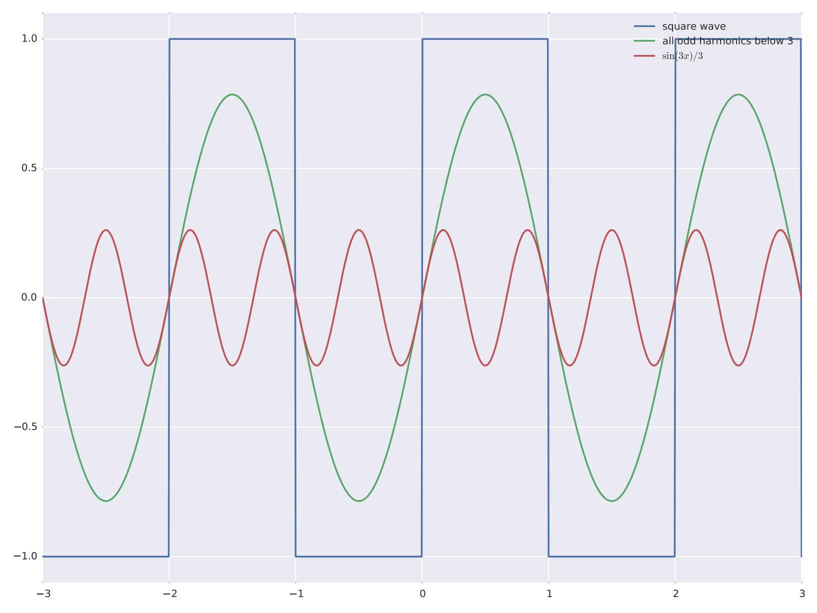
Bonus: if they are periodic, it's only a discrete set of harmonics!
Example: square wave

[1] i.e. the signals we care about

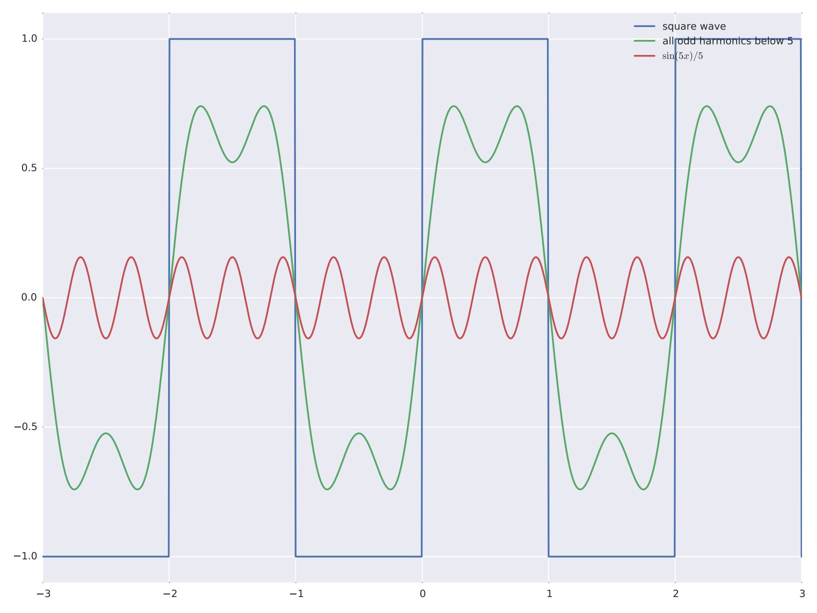
Square Wave reconstruction through harmonic functions



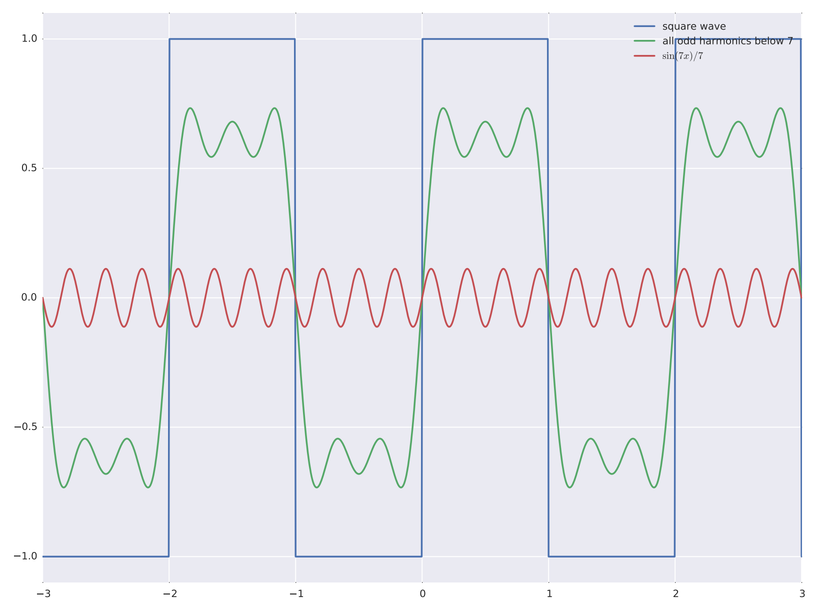
Square Wave reconstruction through harmonic functions



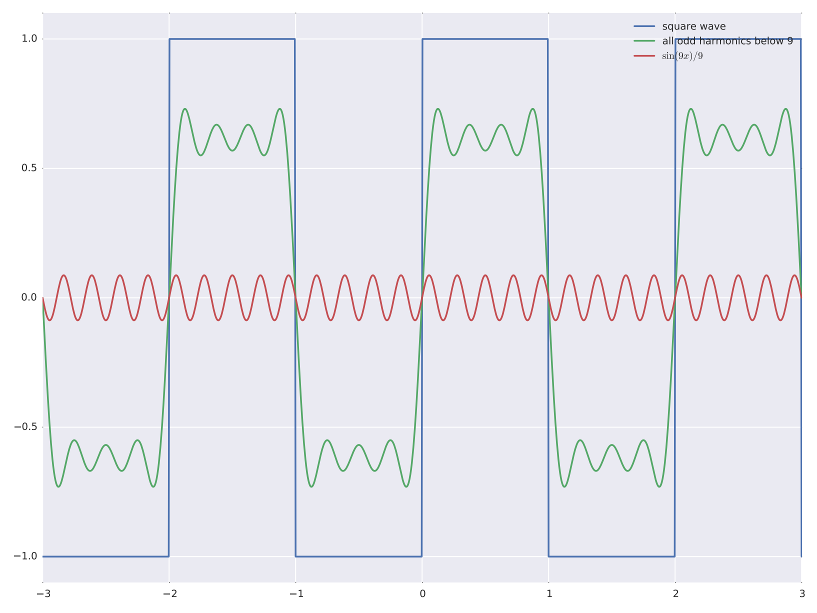
Square Wave reconstruction through harmonic functions



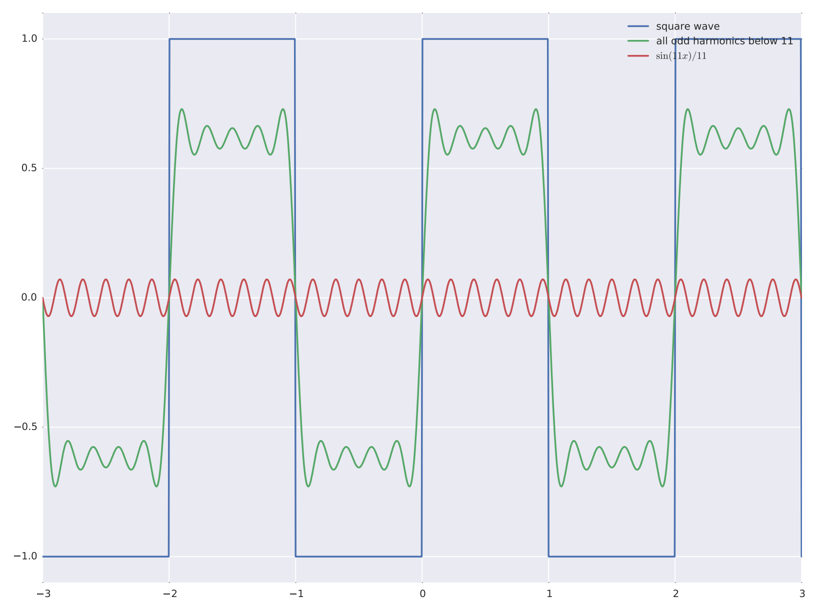
Square Wave reconstruction through harmonic functions



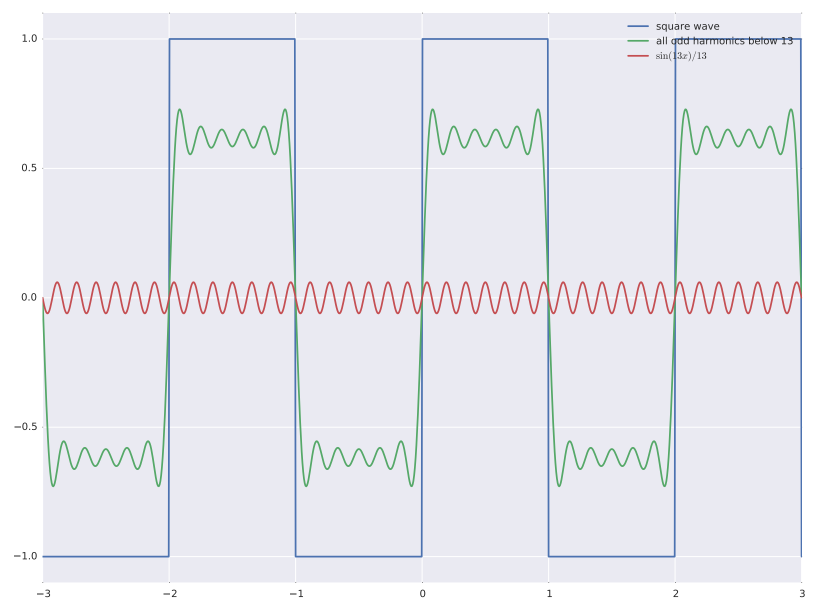
Square Wave reconstruction through harmonic functions



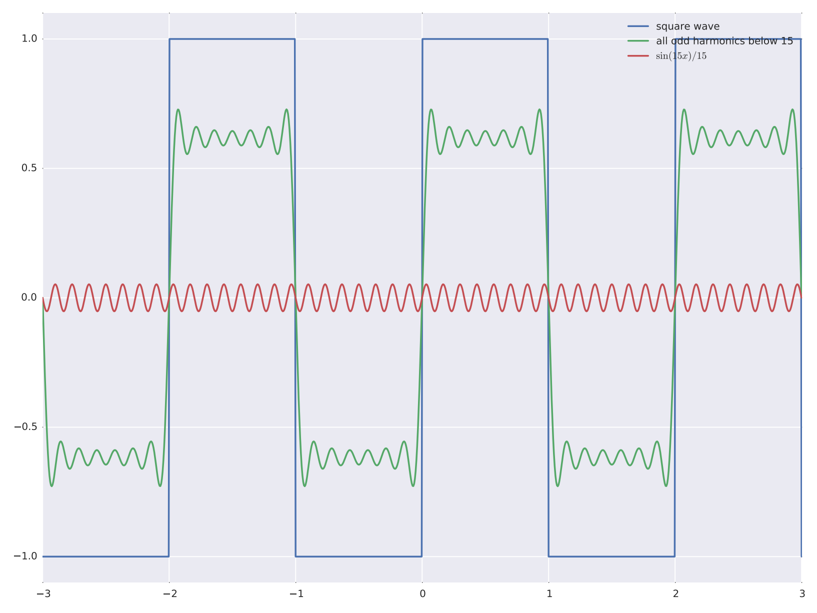
Square Wave reconstruction through harmonic functions



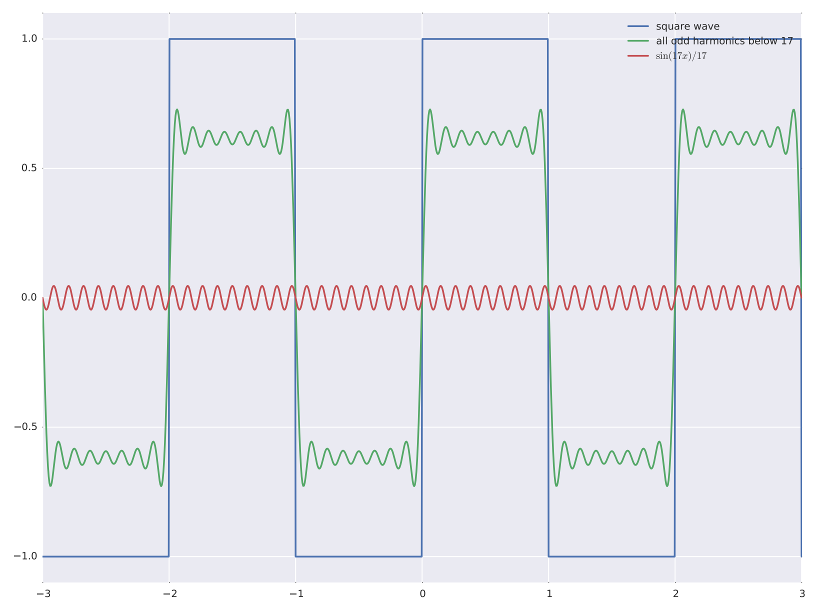
Square Wave reconstruction through harmonic functions



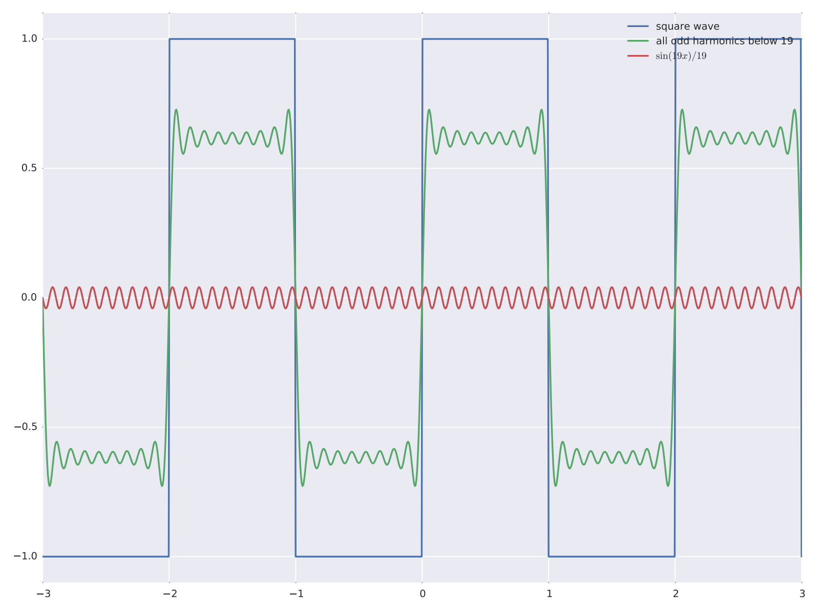
Square Wave reconstruction through harmonic functions



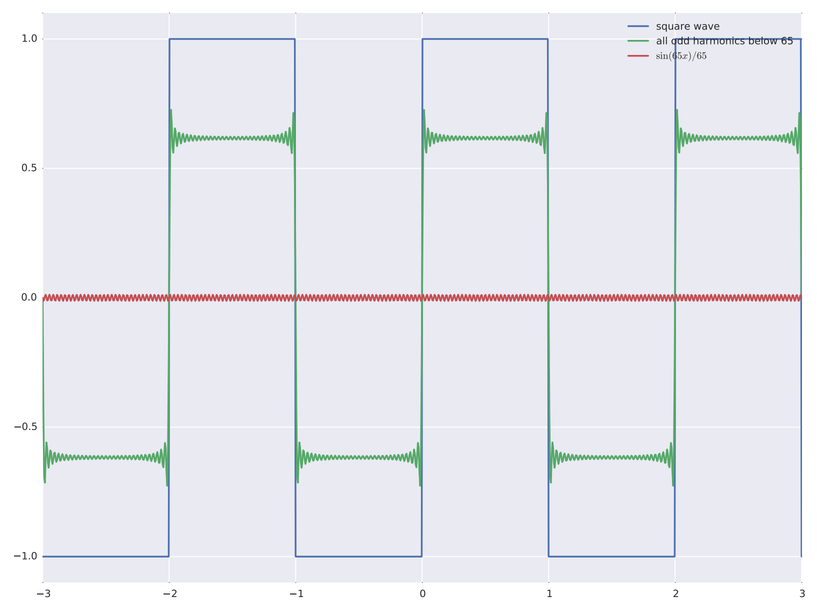
Square Wave reconstruction through harmonic functions



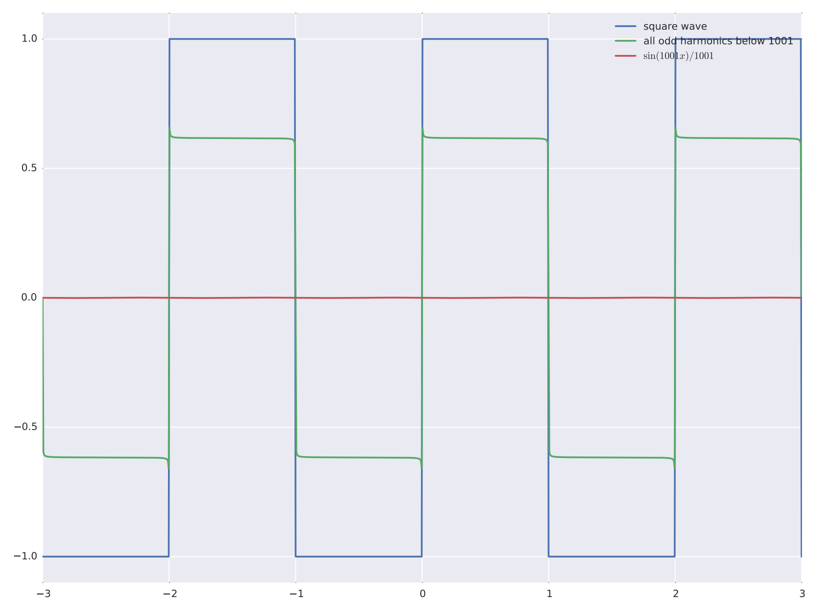
Square Wave reconstruction through harmonic functions



Square Wave reconstruction through harmonic functions

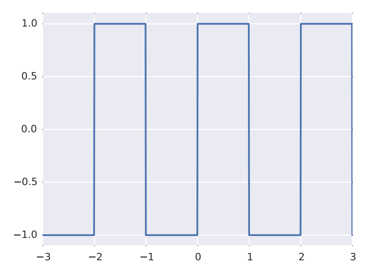


Square Wave reconstruction through harmonic functions

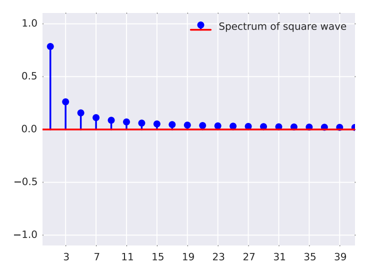


Taking a look at the spectrum

Intuitively, we know that all our sines are just single tones, and will leave a simple line in the spectrum

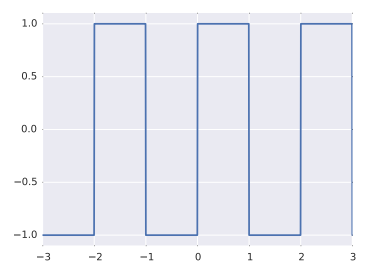


$\leftrightarrow$

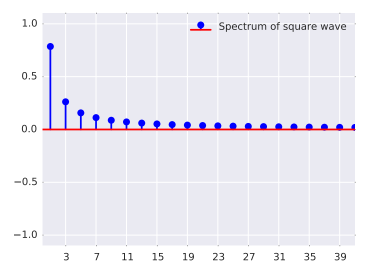


Taking a look at the spectrum

Intuitively, we know that all our sines are just single tones, and will leave a simple line in the spectrum

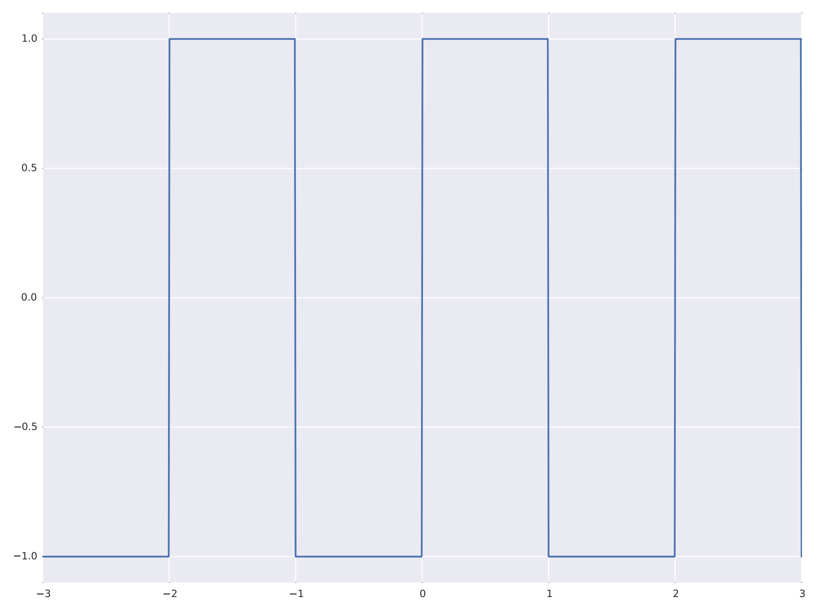


$\leftrightarrow$

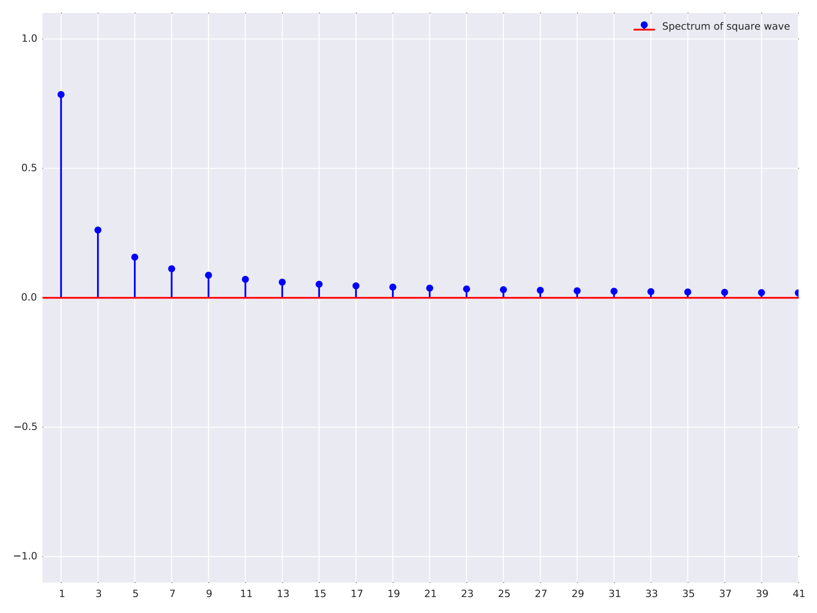


- The **Fourier Transform** actually does exactly that: Converting between time domain and frequency domain.
- Allows for negative frequencies and complex signals/spectra; a bit much math for 30 minutes

Taking a look at the spectrum



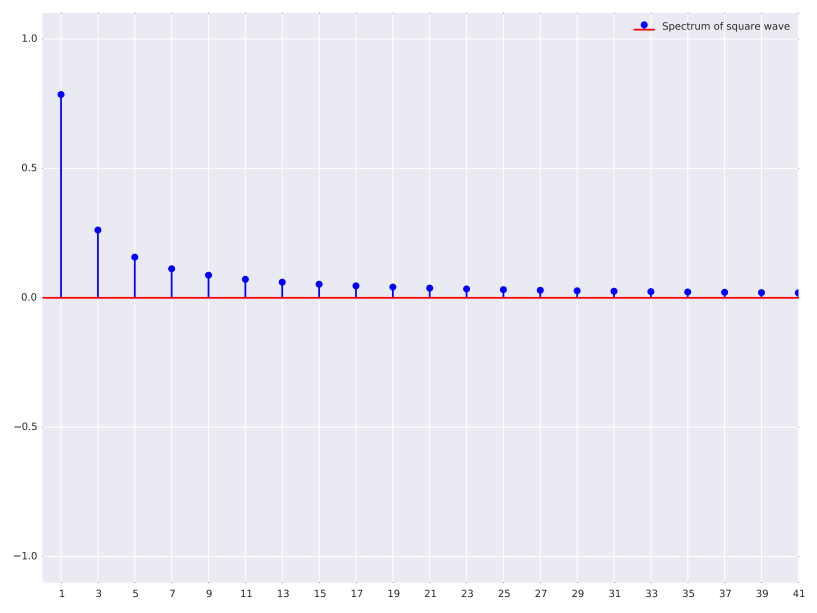
Taking a look at the spectrum



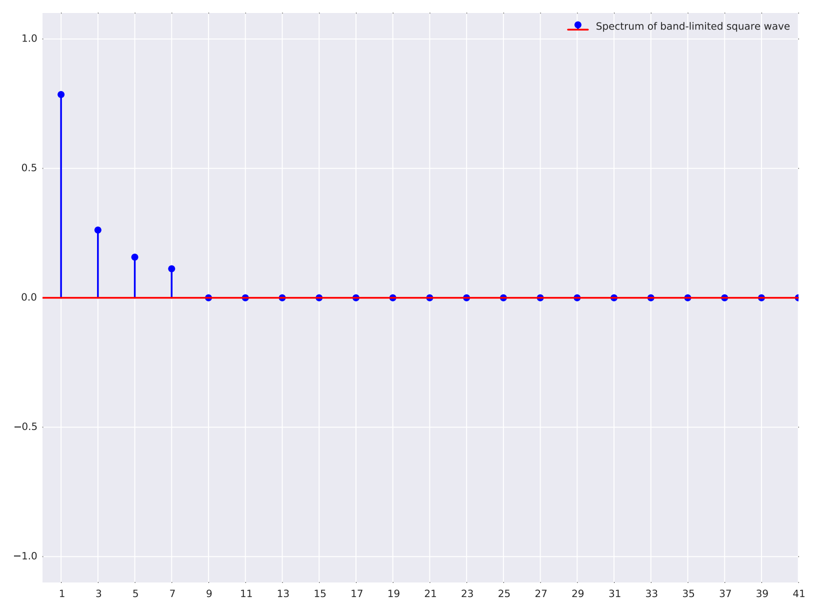
Why is this important?

- real-world systems always have a bandwidth-limiting, usually low-pass, behaviour
- that leaves us with only a limited amount of spectrum to reproduce the original signal

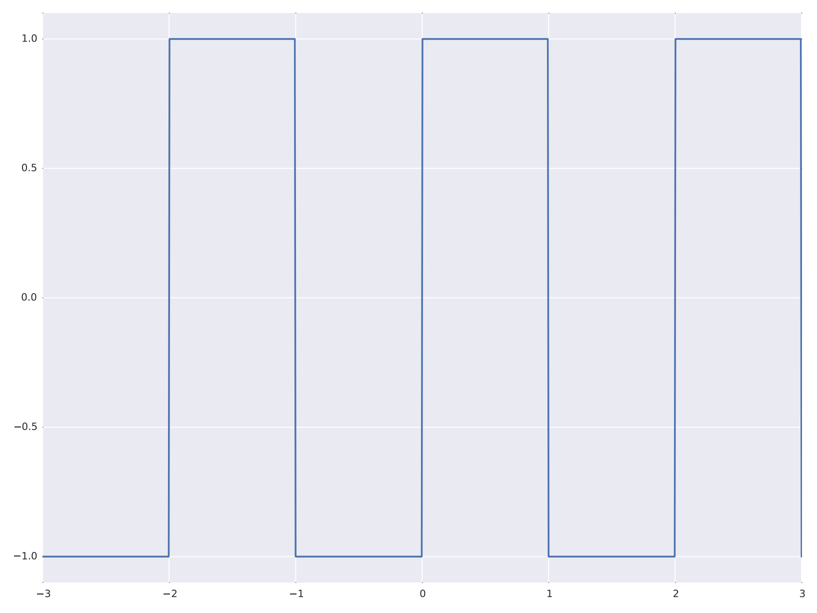
Why is this important?



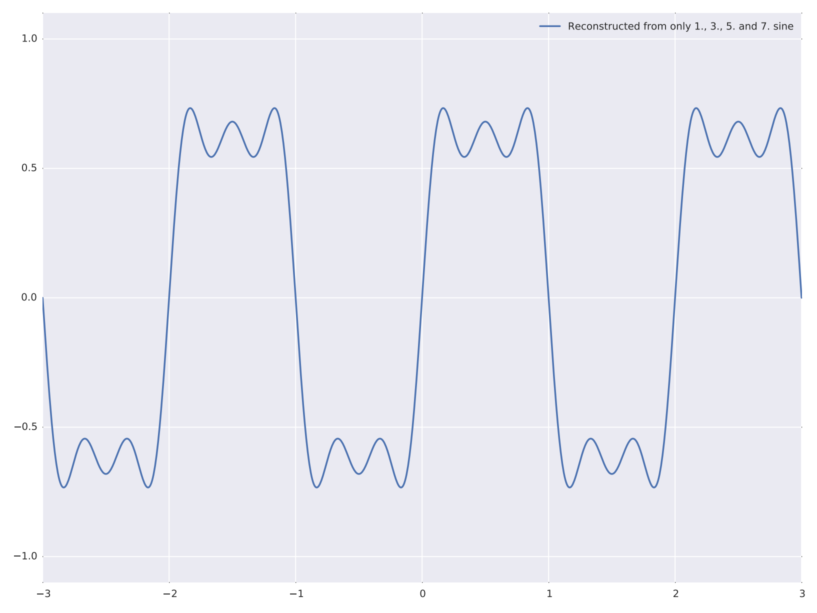
Why is this important?



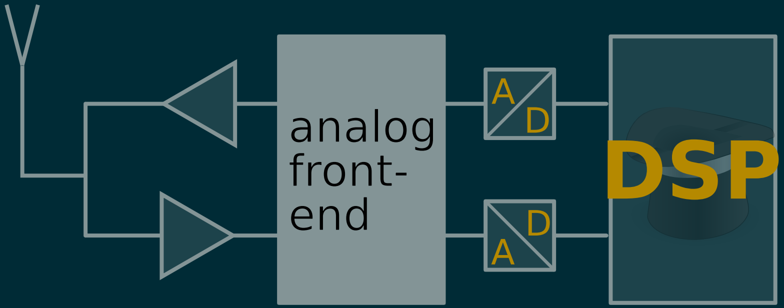
Why is this important?



Why is this important?



Remember this slide?



Digital Signal Processing

- works with a digital signal instead of analog things like voltages, currents
- typically uses software running on processors or specific hardware to implement all kinds of functionality
  - computers are cheap
  - good filters are expensive
  - software can much easier be written than implementing e.g. a cell phone in hardware alone

What *is* a Digital Signal?

formally

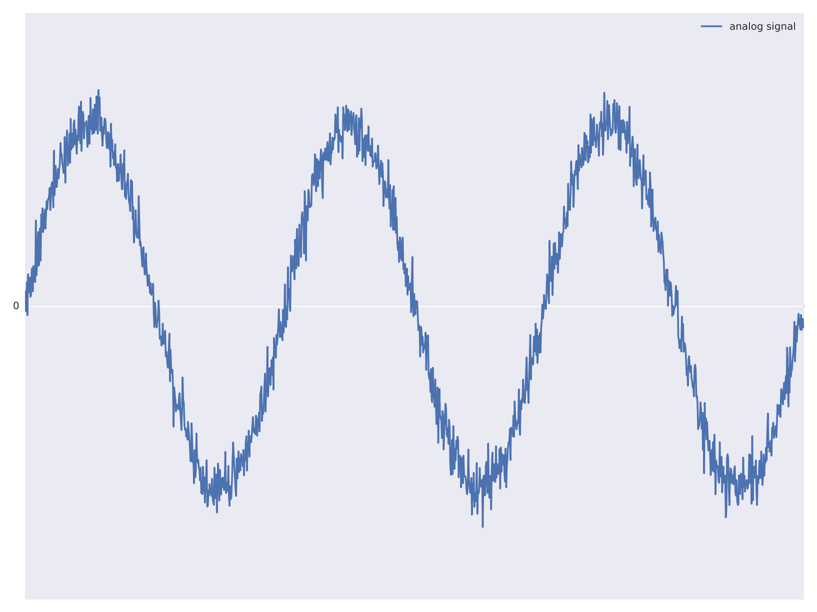
Definition:

A digital signal is a signal that

- only takes discrete values, and
- only exists for discrete times.

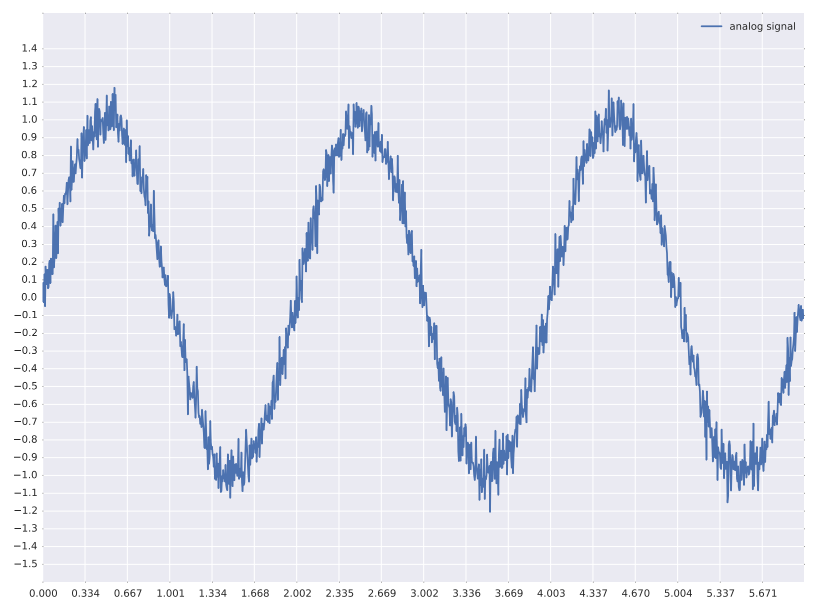
What *is* a Digital Signal?

visually



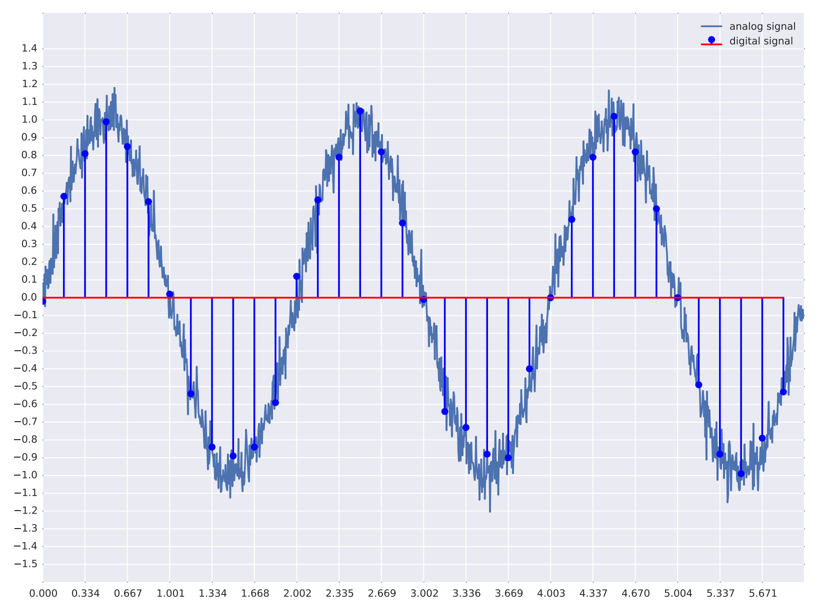
What *is* a Digital Signal?

visually



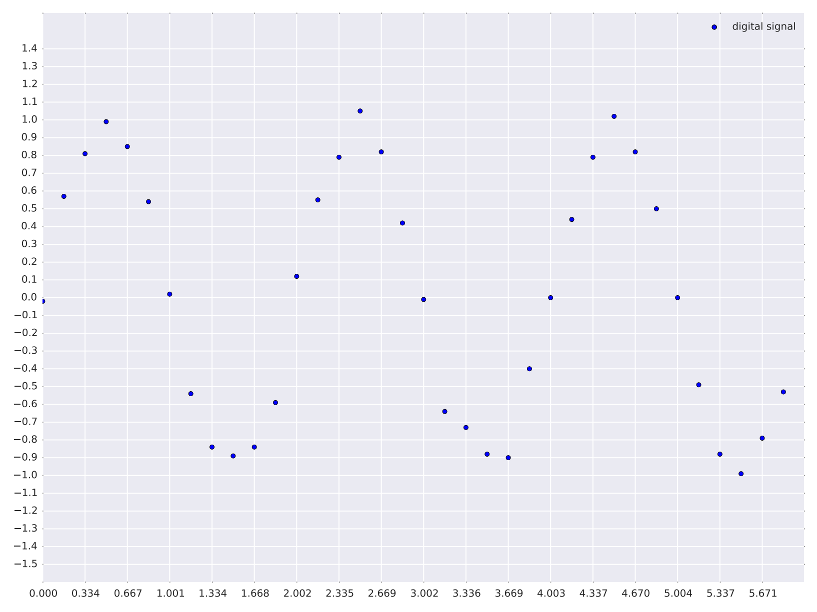
What *is* a Digital Signal?

visually



What *is* a Digital Signal?

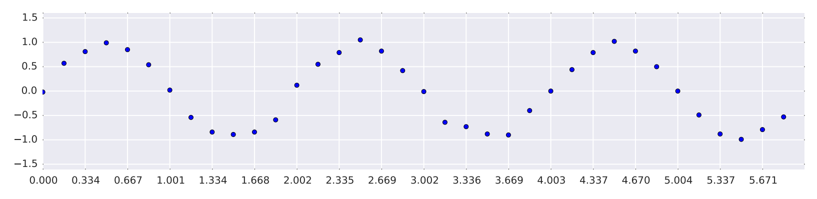
visually



Digital Signals

are great!

- can just be represented as series of numbers



Digital Signals

are great!

- can just be represented as series of numbers
  `-0.02, 0.57, 0.81, 0.99, 0.85, 0.54, 0.02, -0.54, -0.84, -0.89, -0.84, -0.59, 0.12, 0.55, 0.79, 1.05, 0.82, 0.42, -0.01, -0.64, -0.73, -0.88, -0.90, -0.40, 0.00, 0.44, 0.79, 1.02, 0.82, 0.50, -0.00, -0.49, -0.88, -0.99, -0.79, -0.53`

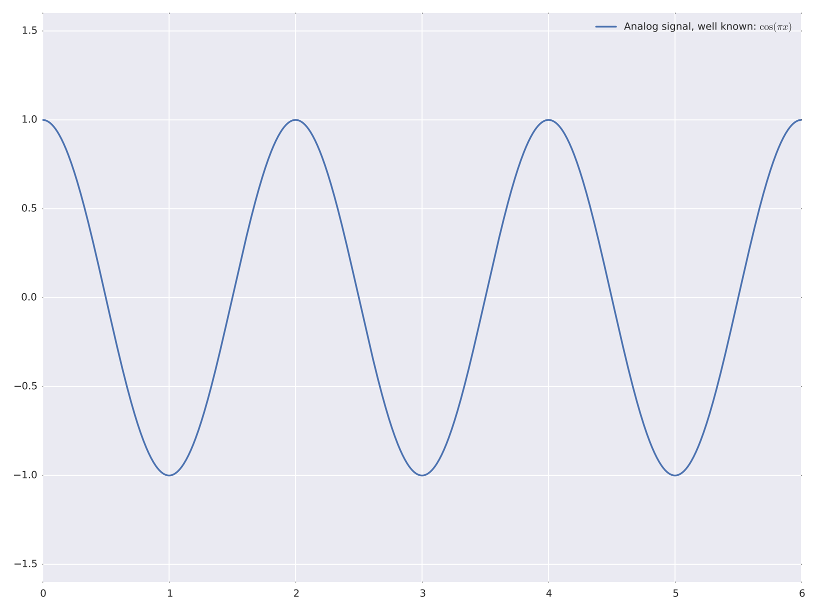
Digital Signals

are great!

- can just be represented as series of numbers
  -0.02, 0.57, 0.81, 0.99, 0.85, 0.54, 0.02, -0.54, -0.84, -0.89, -0.84, -0.59, 0.12, 0.55, 0.79, 1.05, 0.82, 0.42, -0.01, -0.64, -0.73, -0.88, -0.90, -0.40, 0.00, 0.44, 0.79, 1.02, 0.82, 0.50, -0.00, -0.49, -0.88, -0.99, -0.79, -0.53
- which is actually something that computers can work with!

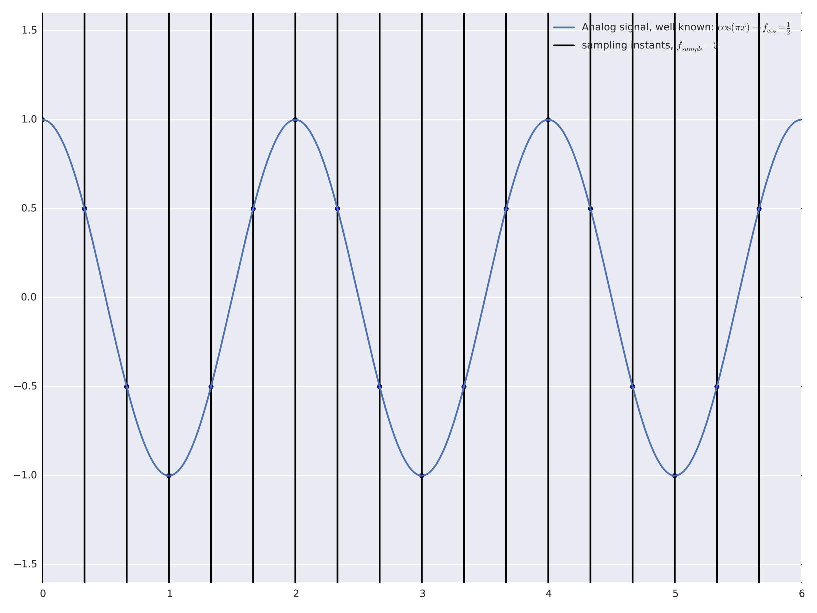
Sampling

Going from Analog to Digital in N discrete Steps



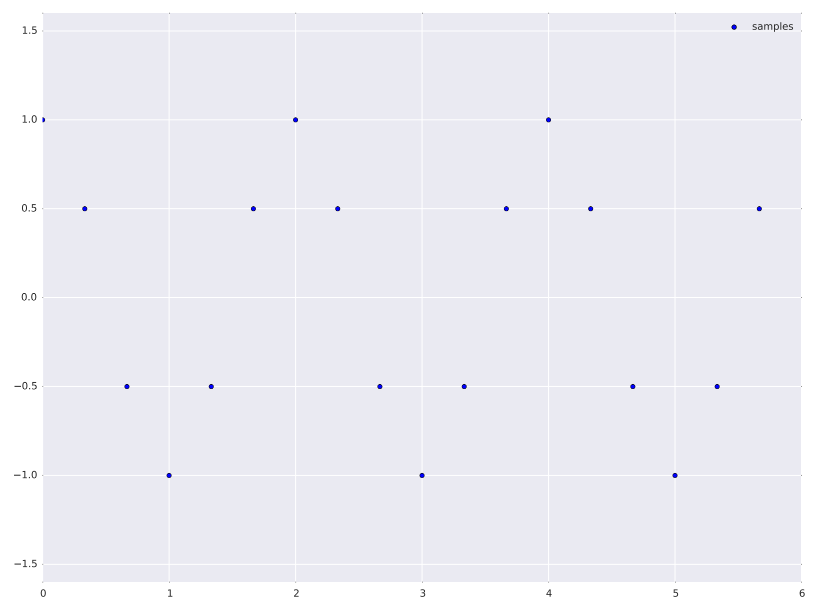
Sampling

Going from Analog to Digital in N discrete Steps



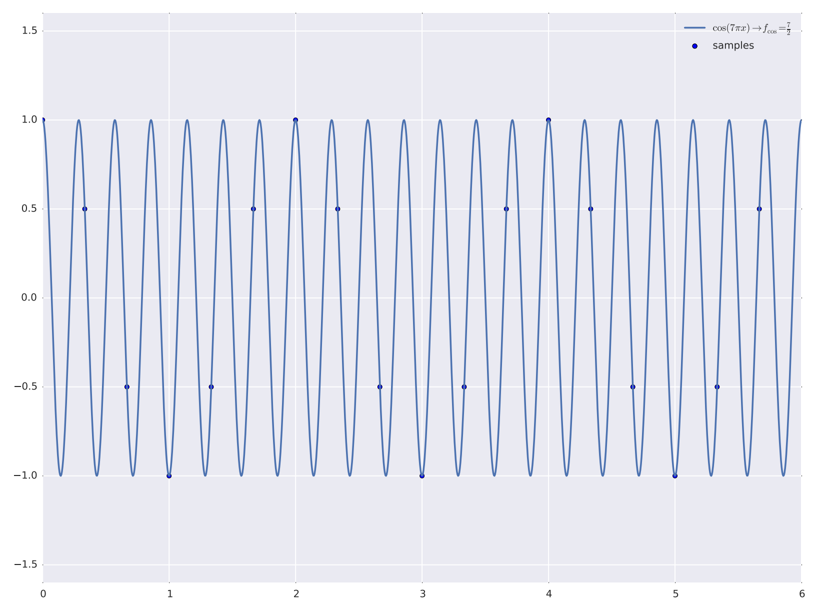
Sampling

Reconstructing is easy!



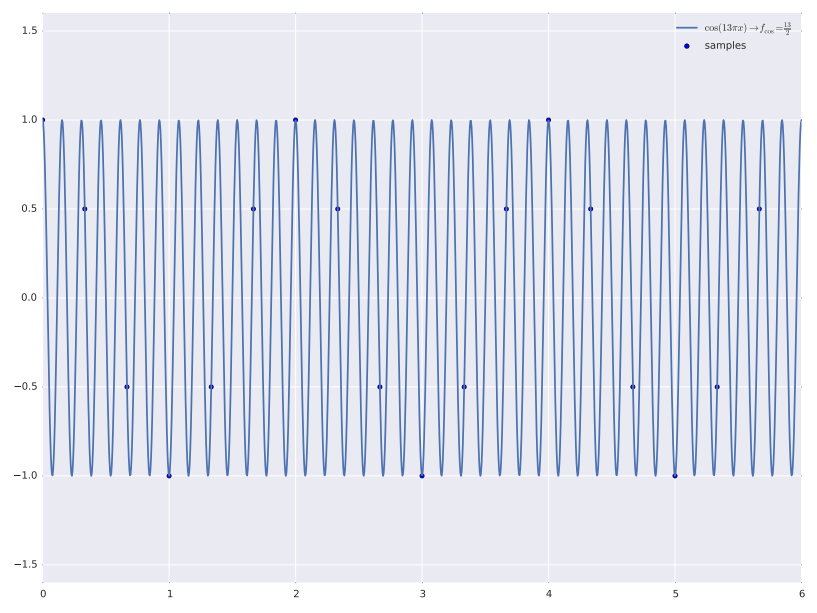
Sampling

Ouch.



Sampling

Ouch.



Sampling – Periodicity in Frequency Domain

- There's *always* ambiguity whether the original periodic signal had frequency f ...

Sampling – Periodicity in Frequency Domain

- There's *always* ambiguity whether the original periodic signal had frequency f . . .
- or $f + f_{sample}$ . . .

Sampling – Periodicity in Frequency Domain

- There's *always* ambiguity whether the original periodic signal had frequency f ...
- or $f + f_{sample}$ ...
- or $f + 2f_{sample}$ ...

Sampling – Periodicity in Frequency Domain

- There's *always* ambiguity whether the original periodic signal had frequency f ...
- or $f + f_{sample}$ ...
- or $f + 2f_{sample}$ ...
- or, in fact, $f + Nf_{sample}$ for any $N \in \mathbb{N}$.

Sampling – Periodicity in Frequency Domain

when considering a digital signal, its spectrum

- can only meaningfully be defined for a bandwidth of f_{sample},
- repeats every f_{sample}.

Hence: Sampling analog signals demands:

bandwidth limited sufficiently (filtered)!

- frequencies at $f + Nf_{sample}$ ending up at f is called **aliasing**.
- Anti-Aliasing Filter: typically low pass filter

Works the other way around, too – **Images** every f_{sample} when DAC'ing $\rightarrow$ *Reconstruction* filter

Sampling – Periodicity in Frequency Domain

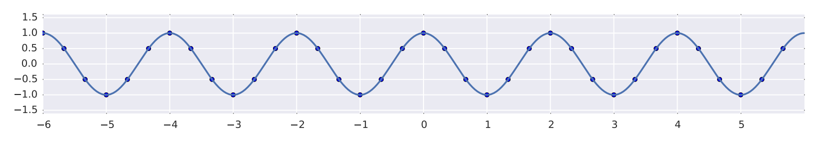
Undersampling

Alternatively *use* aliasing

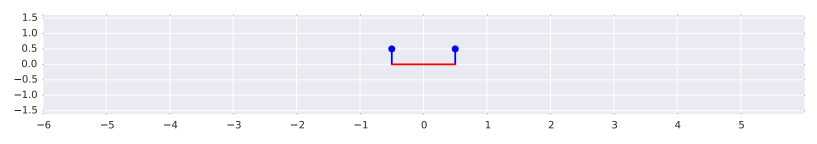
- alias a higher range of spectrum into baseband – **Undersampling**
- works well for relatively low frequencies (filters are easy/affordable/can be build adjustably)
- good high-frequency analog filters expensive/hard to make
- building a receiver working 0–25 MHz just as well as 5.000–5.025 GHz is physically hard
- . . . and expensive: imagine the loads of filters!
- typical frequency agility seen with Ettus USRPs is achieved by first mixing to baseband, and then digitizing

Spectra of Digital Signals

- Example: impossible to tell whether your signal is $\cos(\pi x)$ or $\cos(-\pi x)$:

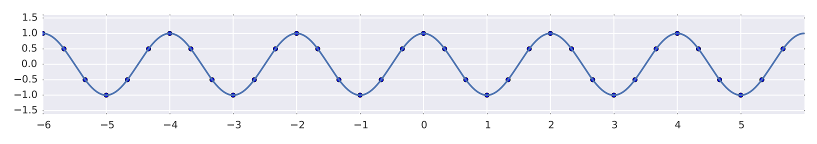


- spectrum of $\cos(\pi x)$ has a isolated value at both $\pm\frac{1}{2}$

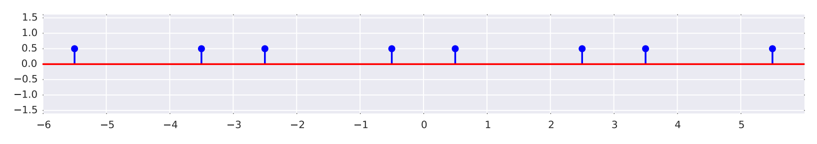


Spectra of Digital Signals

- Example: impossible to tell whether your signal is $\cos(\pi x)$ or $\cos(-\pi x)$:



- spectrum of $\cos(\pi x)$ has a isolated value at both $\pm\frac{1}{2}$
- and because we've sampled it, it's f_{sample}-periodic:



Spectra of Digital Signals

Real Signals: Spectrum is hermitian symmetric

- spectra can be complex (otherwise, we couldn't represent phase of a signal)
- $\Re\{S(f)\} = \Re\{S(-f)\}$
- $\Im\{S(f)\} = -\Im\{S(-f)\}$
- Spectrum Analyzer only shows magnitude of spectrum: Can't tell sign of $\Im\{S\}$

Spectra of Digital Signals

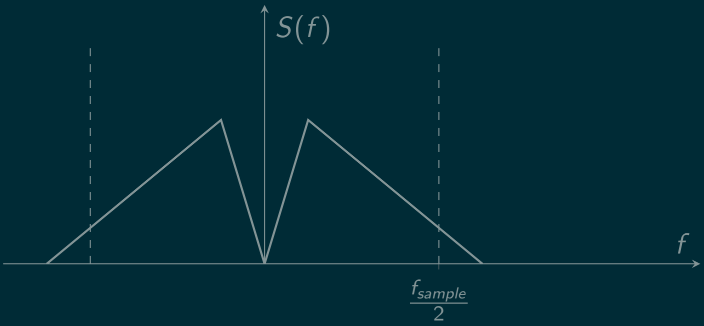
Real Signals: Spectrum is hermitian symmetric

- Positive half of spectrum fully defines negative half
- if symmetric and f_{sample} periodic...
- only half of f_{sample} "usable"

$\rightarrow$ **Sampling Theorem** for real-valued signals

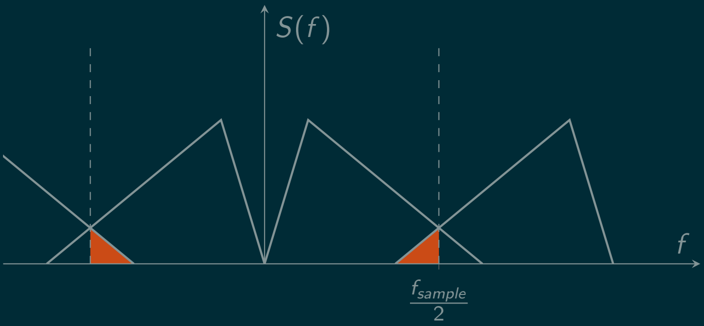
The Shannon-Nyquist Sampling Theorem

For real-valued sampling, the observed bandwidth of the analog signal must be limited to less than half the sampling rate.



The Shannon-Nyquist Sampling Theorem

For real-valued sampling, the observed bandwidth of the analog signal must be limited to less than half the sampling rate.



The Shannon-Nyquist Sampling Theorem

For real-valued sampling, the observed bandwidth of the analog signal must be limited to less than half the sampling rate.

Example:

- Sound cards sample with 44.1 kHz, 48 kHz or 96 kHz. Human perception reaches roughly from 10 Hz to 16 kHz.

The Shannon-Nyquist Sampling Theorem

For real-valued sampling, the observed bandwidth of the analog signal must be limited to less than half the sampling rate.

Example:

- Sound cards sample with 44.1 kHz, 48 kHz or 96 kHz. Human perception reaches roughly from 10 Hz to 16 kHz.
- Understanding voice possible using lower bandwidths – can find sampling rates of 16 kHz and below in standards.

The Shannon-Nyquist Sampling Theorem

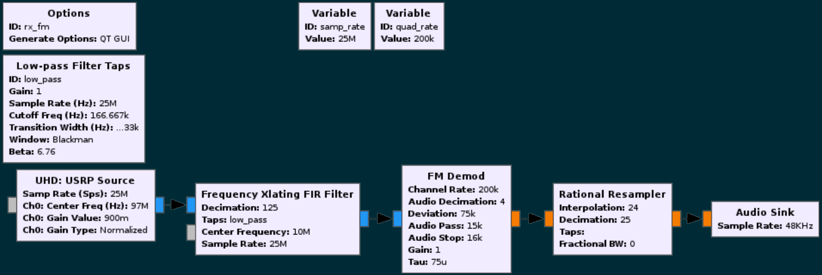
For real-valued sampling, the observed bandwidth of the analog signal must be limited to less than half the sampling rate.

Example:

- superheterodyne receiver:mix 1 MHz of signal from 465.5 – 4.665 MHz to 69.5 – 70.5 kHz ($f_c = 70$ MHz).
  - when considering 0 Hz – 70.5 MHz, one would need a sampling rate of at least 141 MHz
    - high, but far from impossible (USRPs currently do up to 200 MS/s)
    - totally unnecessary!
  - when undersampling, sampling rate of 2 MHz is sufficient
    - high requirement for quality of band-pass signal
    - good SAW filters exist for specific frequencies (reason Superhet is popular, even analog!)

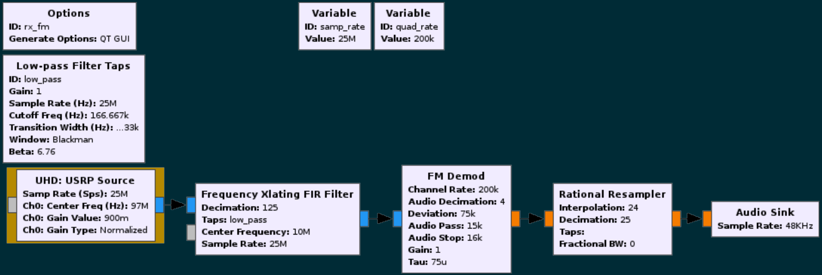
Looking at a complete system

FM Receiver in GNU Radio



Looking at a complete system

FM Receiver in GNU Radio

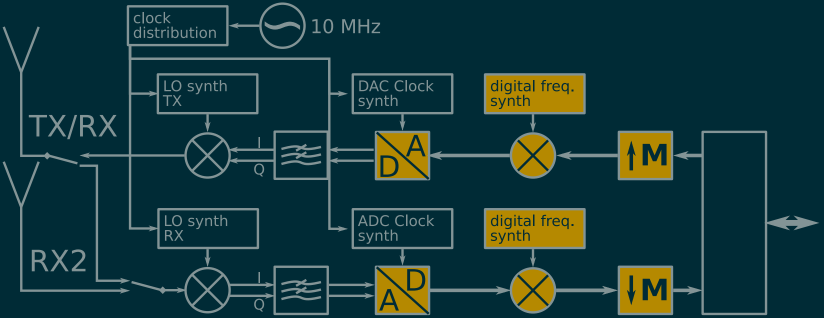


USRP Source

- interface to the USRP
- talks to the driver
- configures all analog and DSP aspects of the USRP
- receives samples

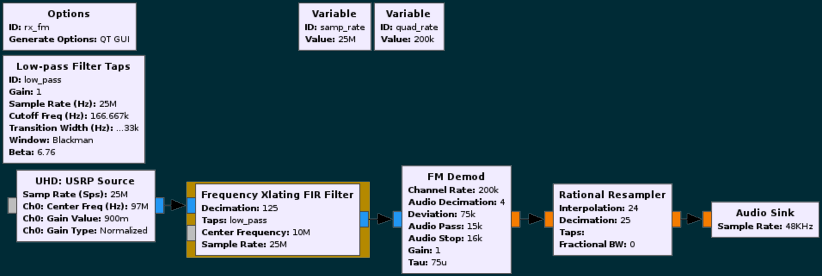
Looking at a complete system

DSP in the USRP



Looking at a complete system

FM Receiver in GNU Radio

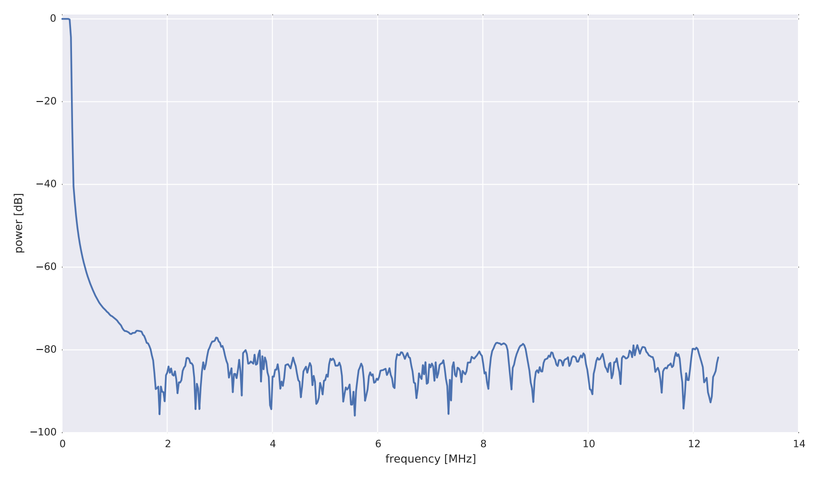


Frequency Translating FIR filter

- shifts the desired frequency to 0 Hz
- then applies filter
- and decimates on the go

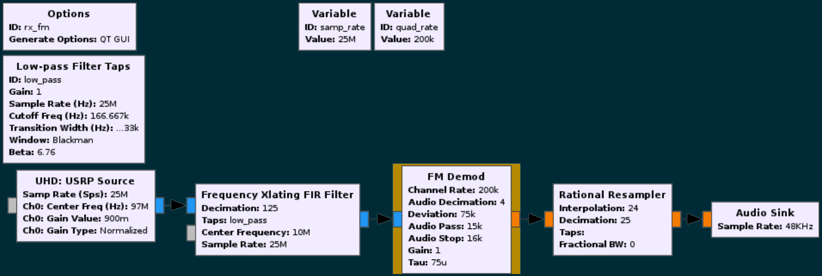
Looking at a complete system

- 2045 tap monster of filter
- . . . runs in real time on old laptop at up to 10 MS/s
- attenuation far above necessity



Looking at a complete system

FM Receiver in GNU Radio

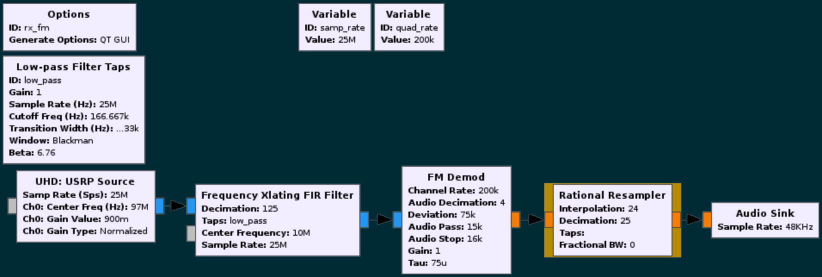


FM Demodulator

- Calculates instantaneous frequency of signal
- integrates and scales the result
- and decimates to an audio-typical rate on the go

Looking at a complete system

FM Receiver in GNU Radio

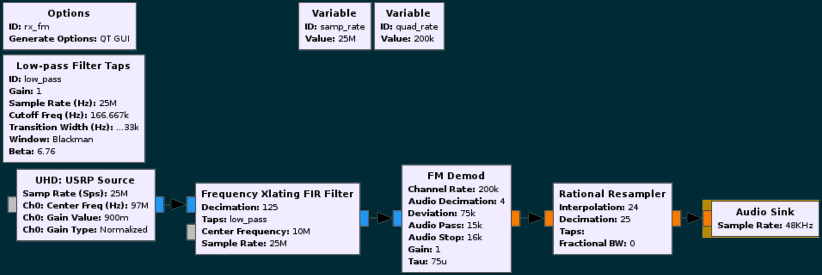


Rational Resampler

- No sound card can do 50 kS/s, but they do 48 kS/s
  - Interpolate to 48x input rate
  - suppress spectral images
  - filter and decimate by 50
- In fact, there's tricks to do the downsampling, filtering and upsampling without going to 50x input rate

Looking at a complete system

FM Receiver in GNU Radio



Audio Sink

- sends samples to the sound card, which Digital-to-Analog converts them

Conclusion

- any signal is representable in digital form . . .
- . . . as long as it's band-limited
- Aliasing can lead to out-of-band overlaying wanted signal
  - Anti-Alias Filtering necessary
  - Aliasing can be used for good
- DSP is a rich toolbox that allows construction of incredible filters at very low cost
- SDR hardware gives access to the raw digital signal – great flexibility
- Toolboxes like GNU Radio make it very easy to build extremely capable SDR applications

Wrapping things up

- Presentation can be found under

  `http://marcus.hostalia.de/sdra16.pdf`
- Always open for mail! marcus@hostalia.de