Electronics Simulations for the nuPRISM detector

Marcin Ziembicki

Warsaw University of Technology
Institute of Radioelectronics

nuPRISM meeting, Kashiwa, 01-02-2015

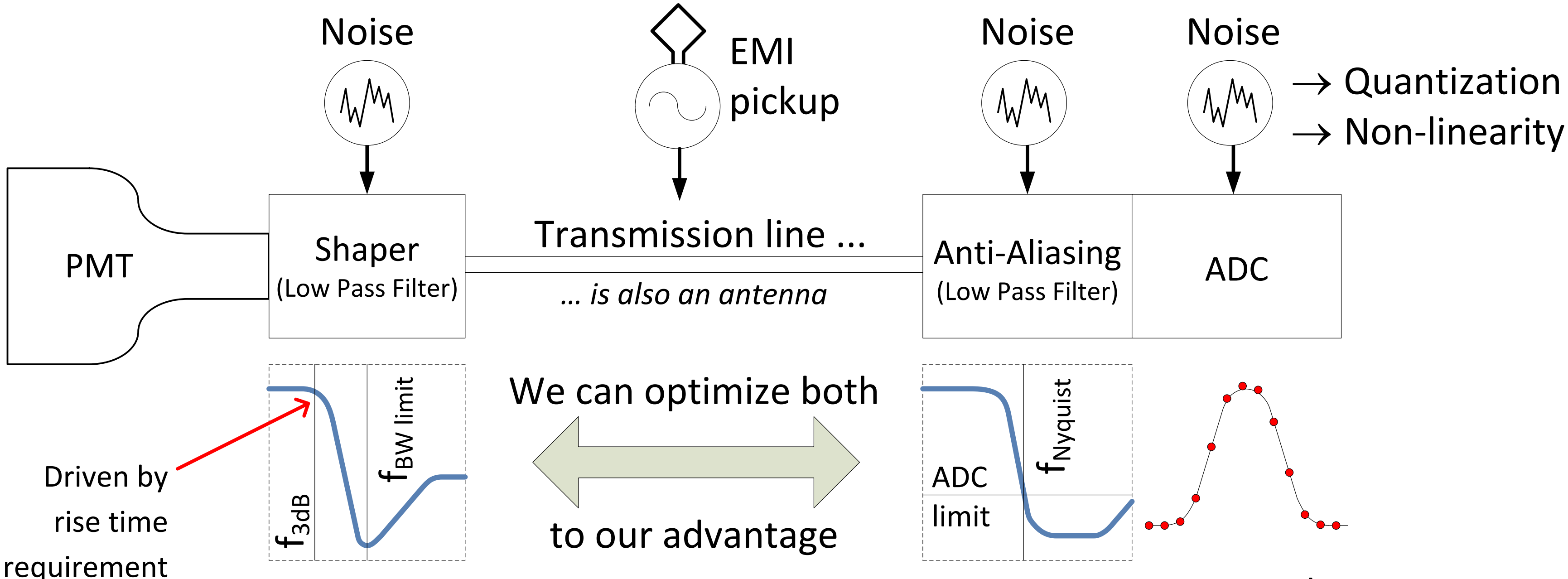
Requirements for the νPRISM

- Electronics with no dead-time.
- Standard commercial ADC + pulse shaping.
- FPGA processing to find PMT hits and calculate pulse time and charge.
- **0.1-1250 p.e. dynamic range (challenging).**
- Distinguish hits that differ by 10s of ns.
- Keep the costs low:
  - Avoid split-gain system.
  - Use slowest & least precise ADC that will do the job.

Simulation Purpose

- Check system performance with various shapers, ADCs (F_S and N_{Bits}) and signal processing algorithms.
- Parameters of interest:
  - **Time resolution**
  - Charge resolution
  - Ability to distinguish piled-up pulsed
- Develop reliable electronics models:
  - Make informed decisions concerning electronics design.
  - Later, include the models into the simulation of the full detector.

Signal Chain



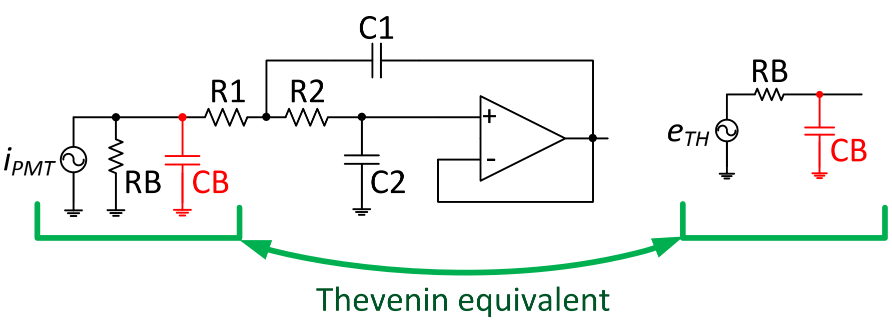
Oversampling

+

Known output from the shaper

Use digital low-pass filtering to remove unwanted signals

Dynamic range issue

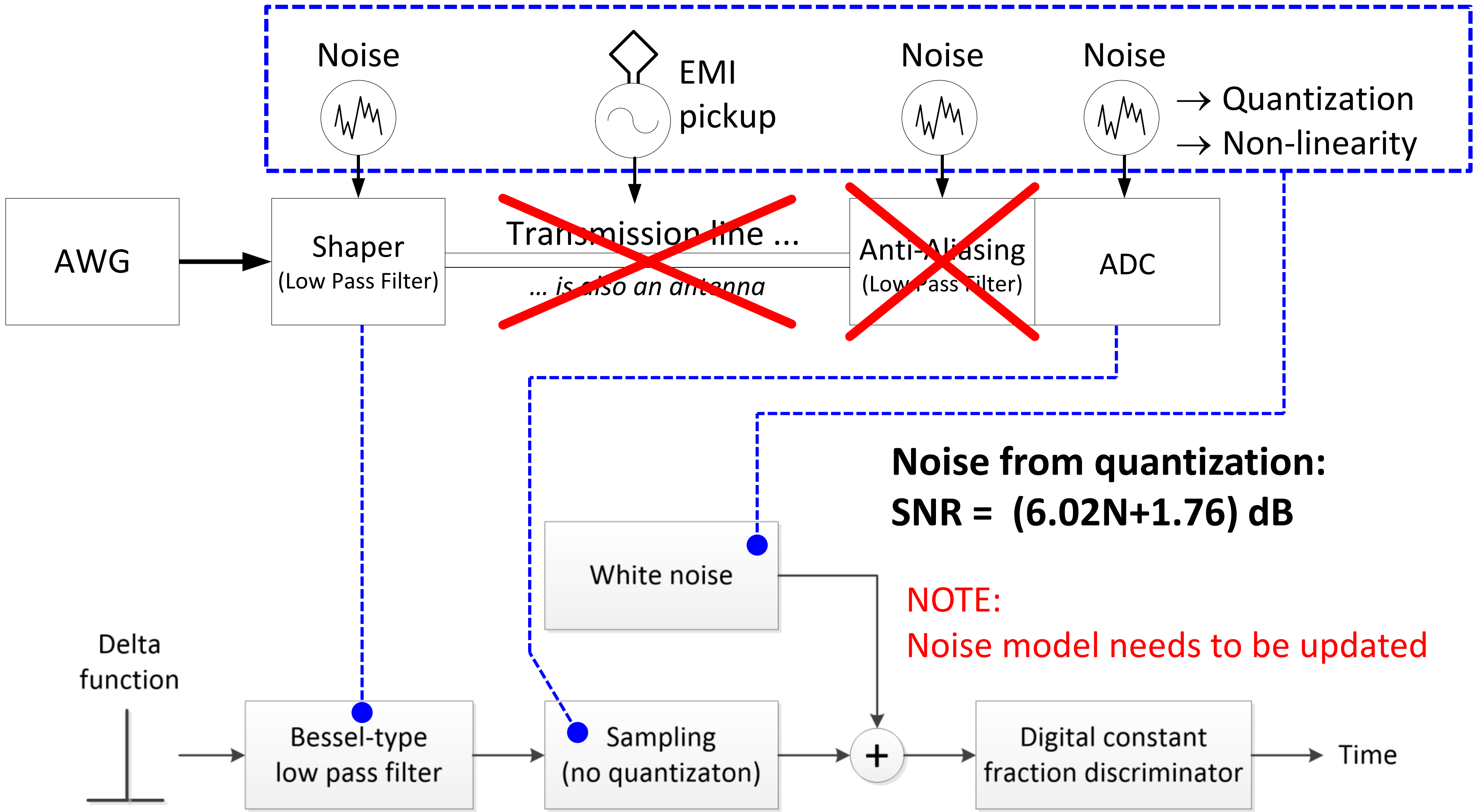


$$v_n = \sqrt{4kTR\Delta f}$$

$$\Delta f = \frac{1}{4RC}$$

$$v_n = \sqrt{\frac{kT}{C}}$$

Simulation Model - Simplified Approach



Noise from quantization:

SNR = (6.02N+1.76) dB

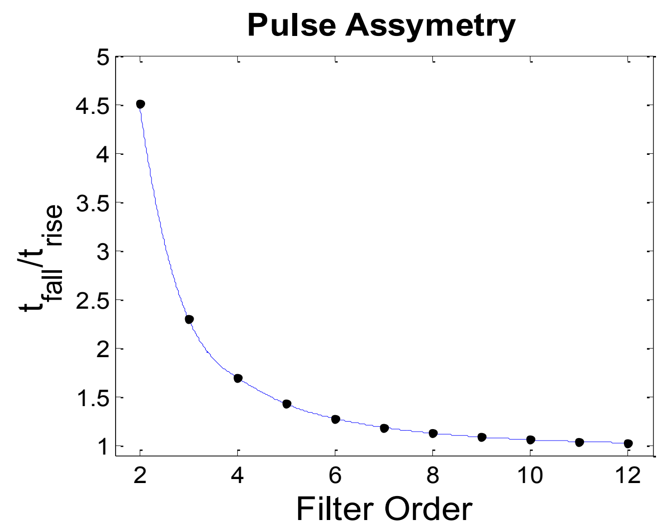
NOTE:

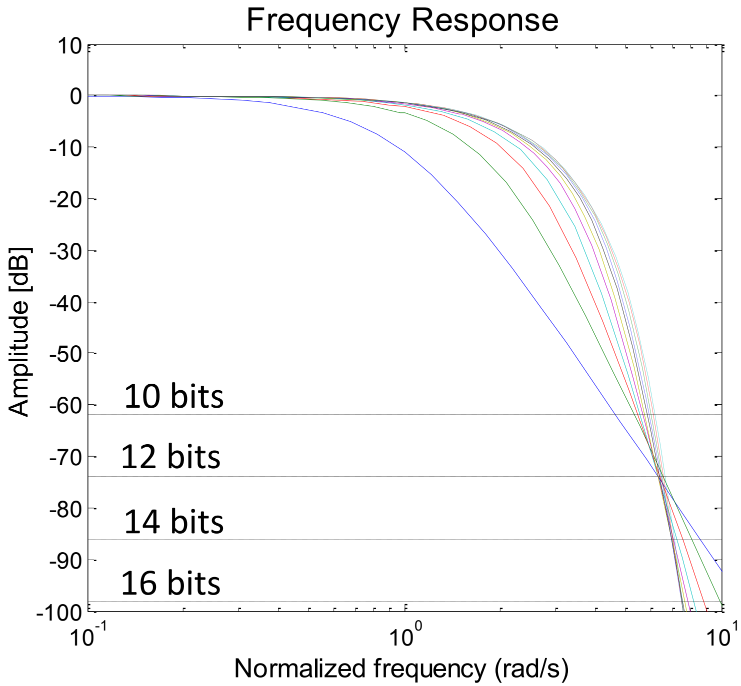
Noise model needs to be updated

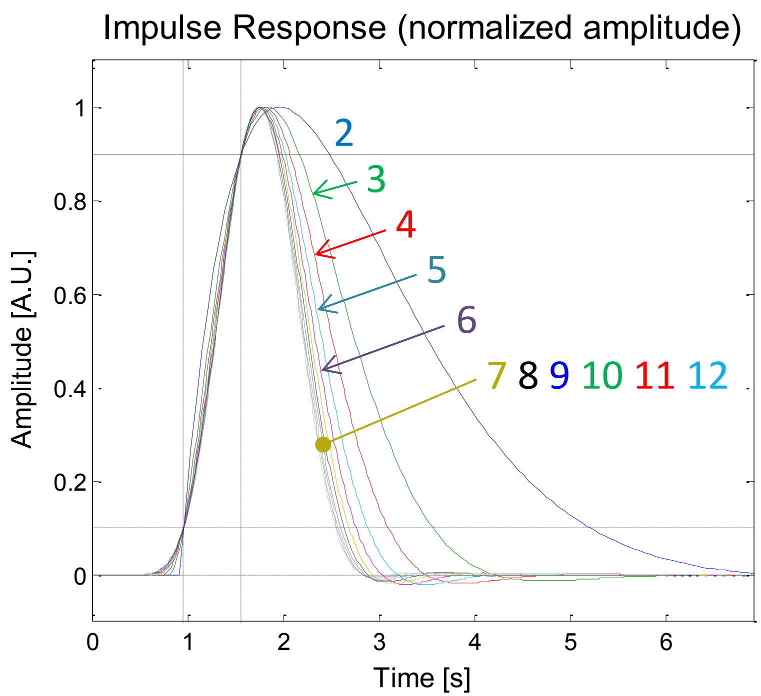
Shaper Design

Design aids (MATLAB code)

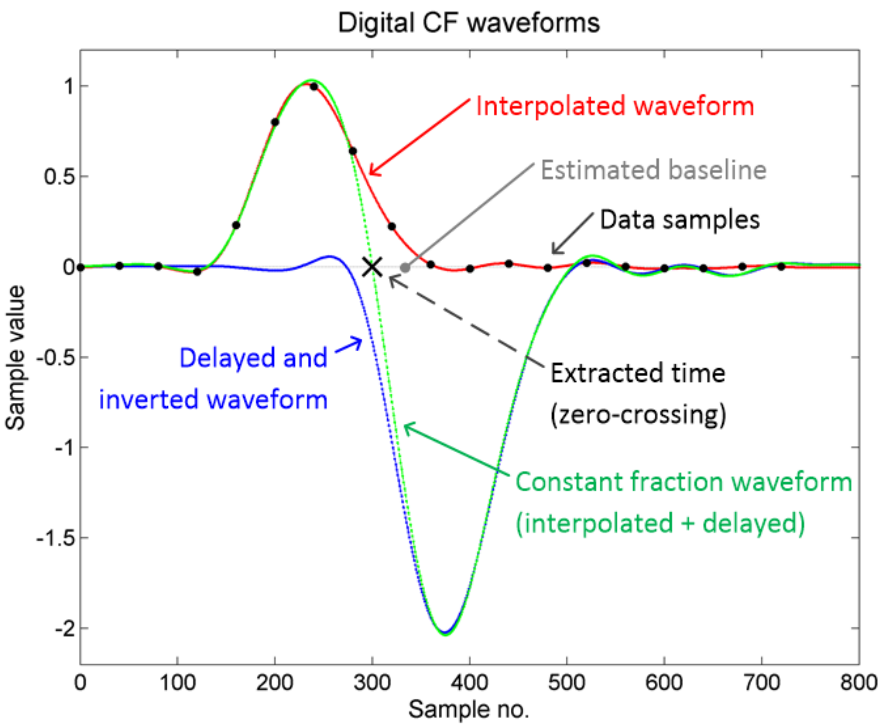
- Calculate filter cutoff frequency:
  - Input parameters: filter order, sampling frequency, rise time (in samples).
- Determine requirements for the amplifier:
  - Bandwidth, slew rate, noise
- Use existing design tools from IC manufacturers to prepare circuit designs.



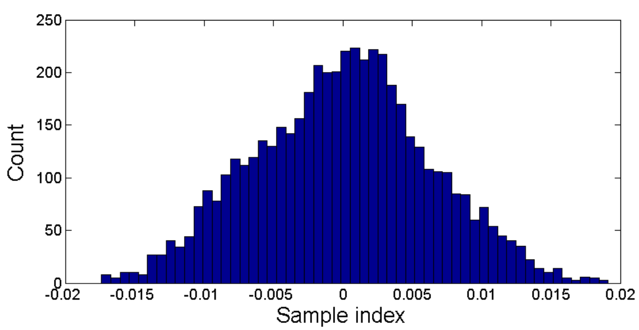




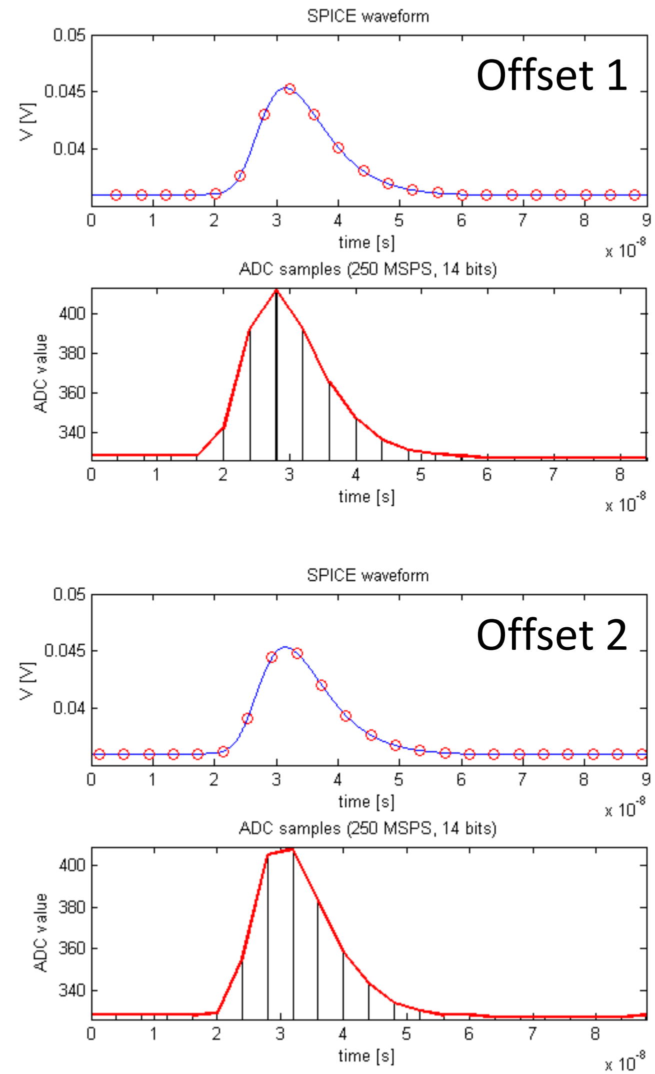
Time Extraction – Digital CFD



Error distribution due to the time offset

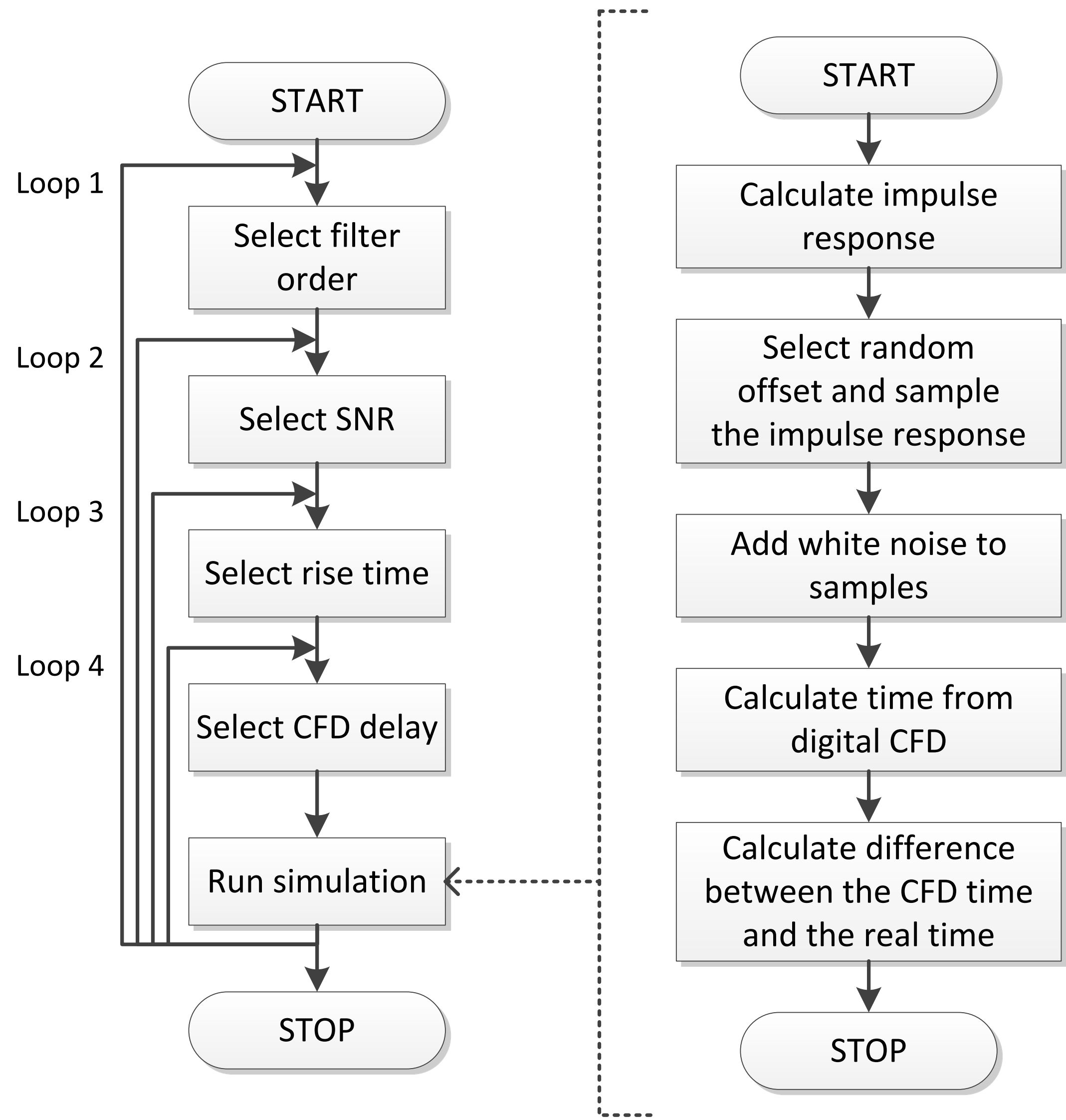


Error due to random time offset between sampling clock and the leading edge



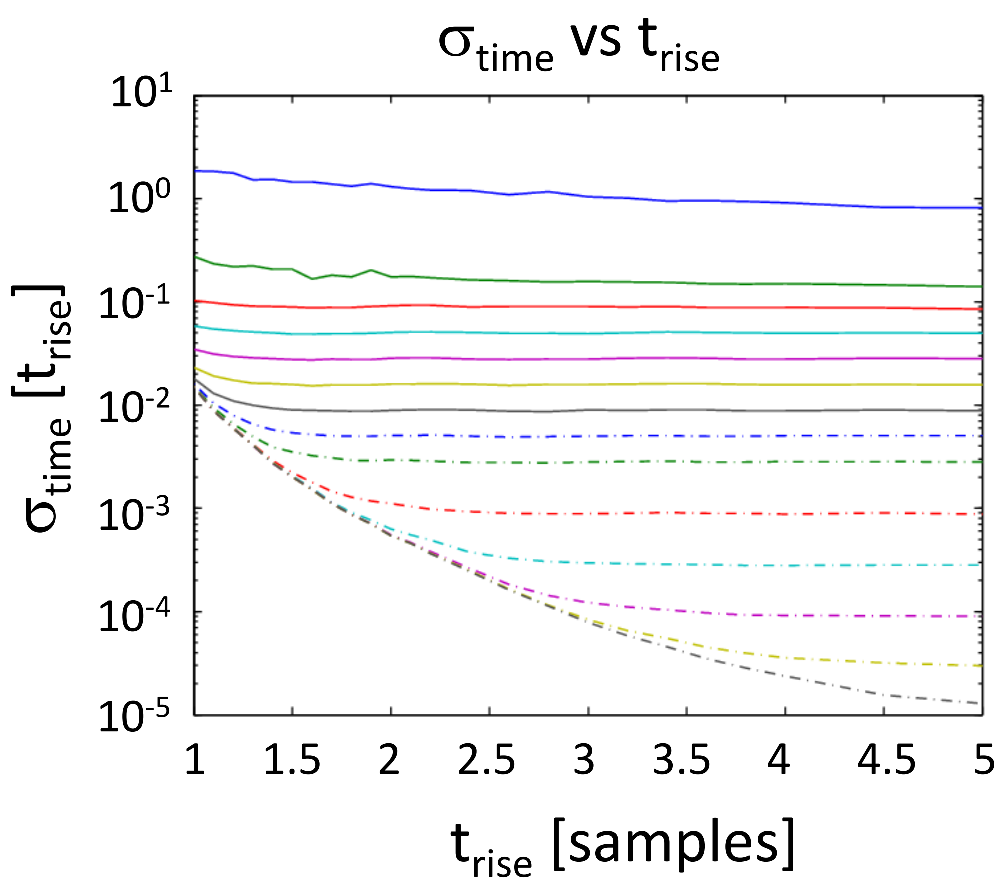
Algorithm

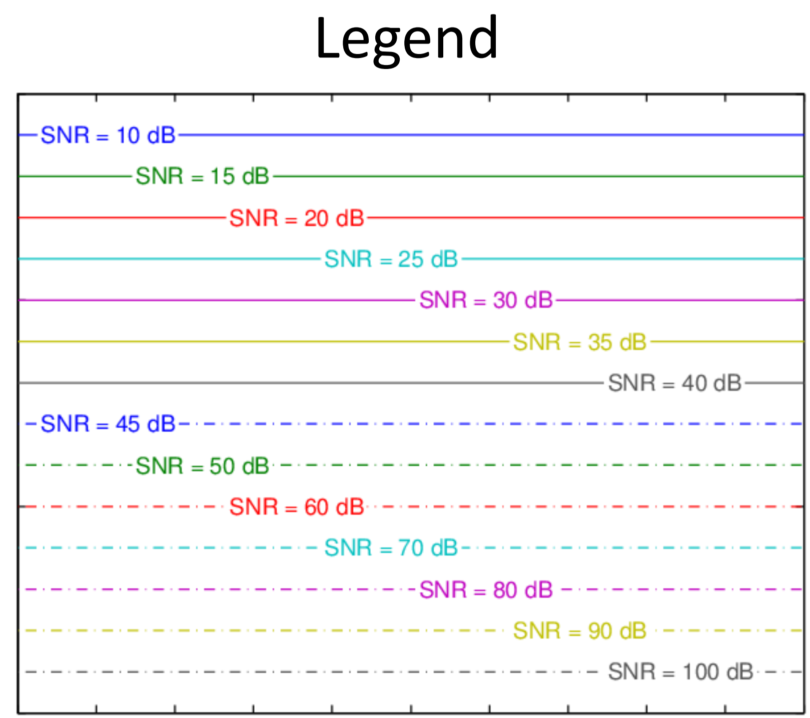
- No PMT randomization
- Signal-level simulation only
  - Savings in computation time
- Generic approach
  - Express results as a function of SNR and shaper's pulse rise time (in samples)
  - Results valid for various ADCs (F_S and N_{BITS})
- Parameter set:
  - Filter orders: 3 to 12
  - Pulse rise time: 1.0 to 5.0 samples
  - SNR: 10 dB to 100 dB
- For a given SNR and rise time, choose CFD delay that allows for the best time resolution



Results (5-th order Bessel filter) - 1/4

If I expect certain SNR and fix rise time of the shaper's pulse, then how would my time resolution depend on the sampling frequency?



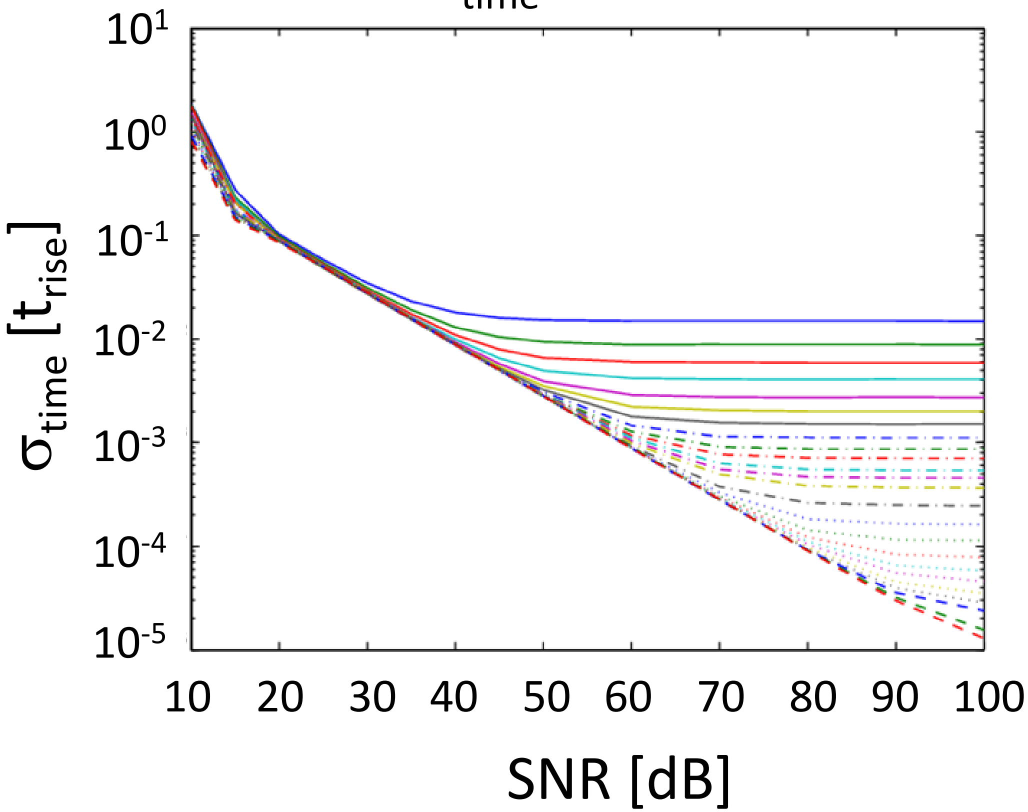


N_{BITS} (real)	10	12	14	16
SNR (1250 p.e.) [dB]	62	74	86	98
noise (p.e.)	1,00	0,25	0,06	0,02
SNR (1 p.e.) [dB]	0	12	24	36
N p.e. for SNR = 20 dB	10,0	2,5	0,6	0,2
N p.e. for SNR = 40 dB	99,7	24,9	6,2	1,6

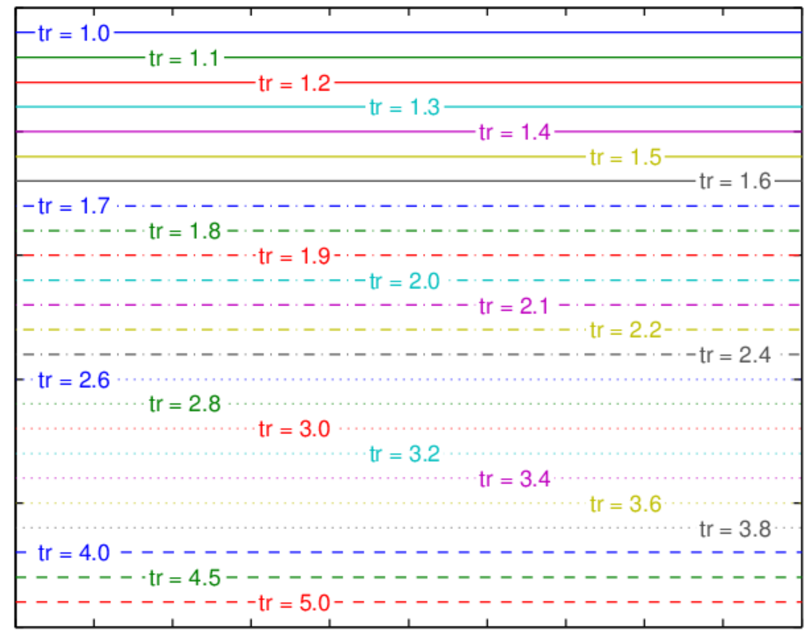
Results (5-th order Bessel filter) - 2/4

If I fix pulse rise time of the shaper's pulse and sampling frequency of the ADC, then how would my time resolution depend on the SNR?

σ_{time} vs SNR



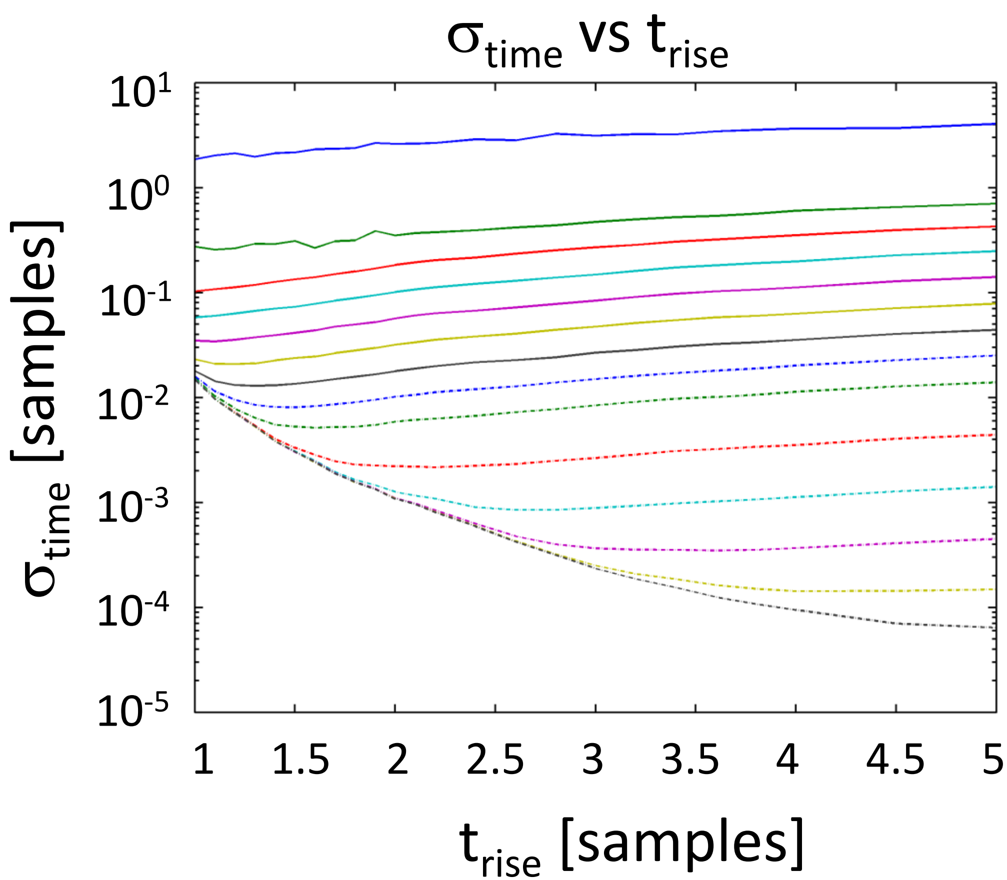
Legend

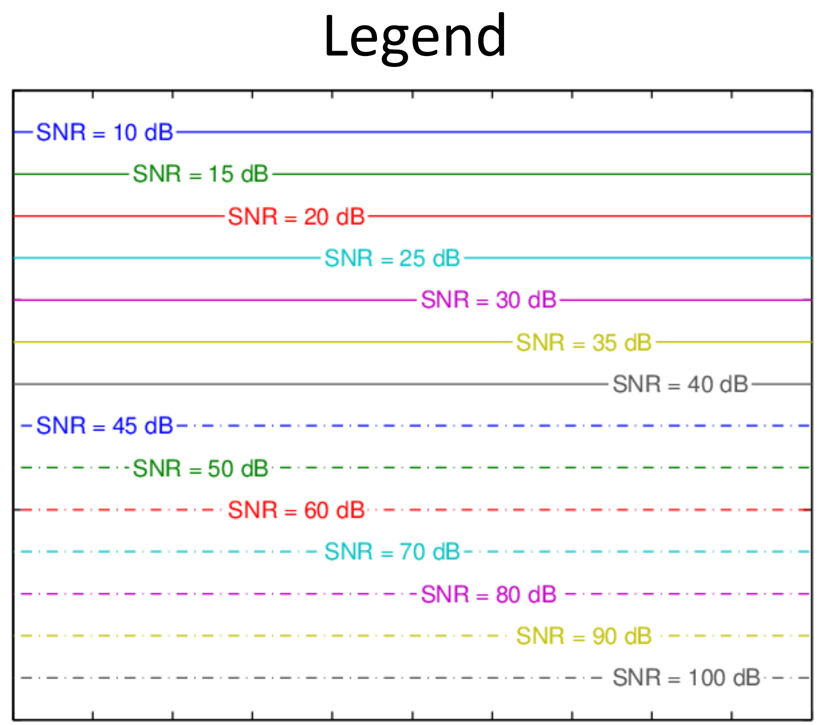


N_{BITS} (real)	10	12	14	16
SNR (1250 p.e.) [dB]	62	74	86	98
noise (p.e.)	1,00	0,25	0,06	0,02
SNR (1 p.e.) [dB]	0	12	24	36
N p.e. for SNR = 20 dB	10,0	2,5	0,6	0,2
N p.e. for SNR = 40 dB	99,7	24,9	6,2	1,6

Results (5-th order Bessel filter) - 3/4

If I expect certain SNR and already have an ADC, then how would the time resolution depend on the rise time of the shaper's pulse?

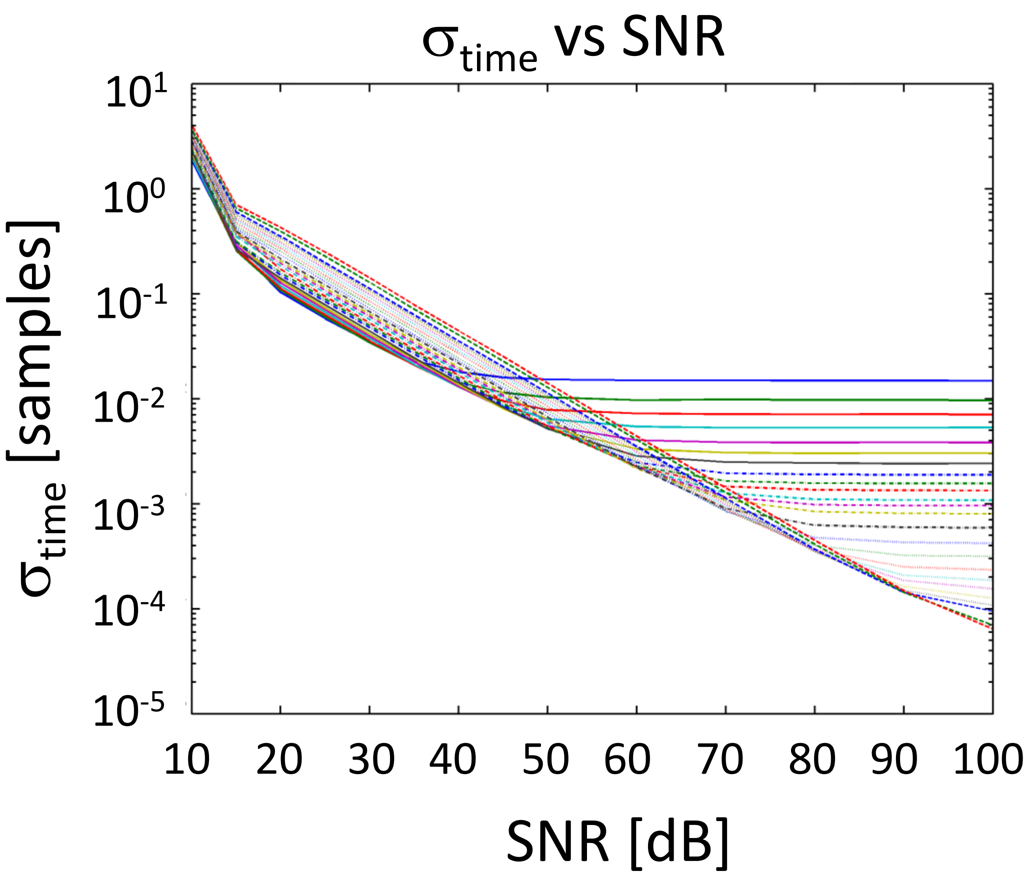




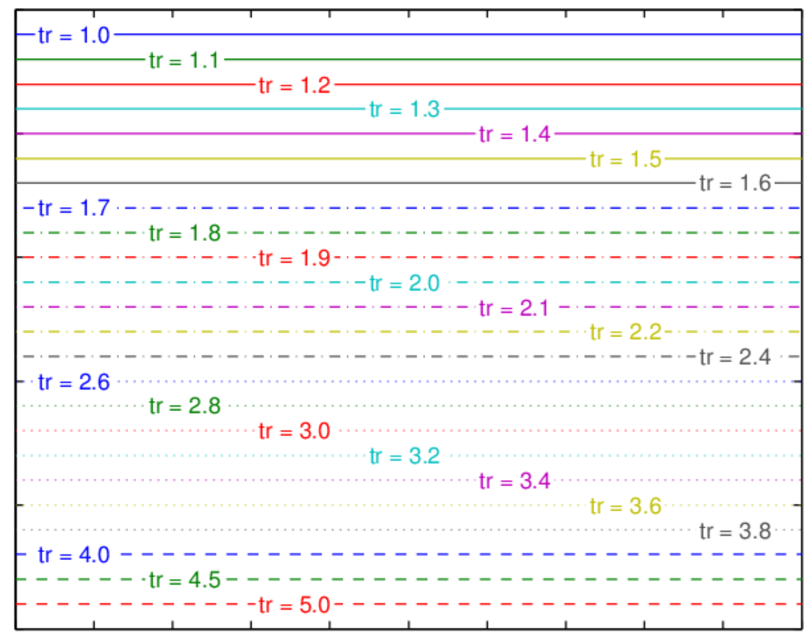
N_{BITS} (real)	10	12	14	16
SNR (1250 p.e.) [dB]	62	74	86	98
noise (p.e.)	1,00	0,25	0,06	0,02
SNR (1 p.e.) [dB]	0	12	24	36
N p.e. for SNR = 20 dB	10,0	2,5	0,6	0,2
N p.e. for SNR = 40 dB	99,7	24,9	6,2	1,6

Results (5-th order Bessel filter) - 4/4

If I already have an ADC, how would my time resolution depend on the SNR and the rise time of the shaper's pulse?

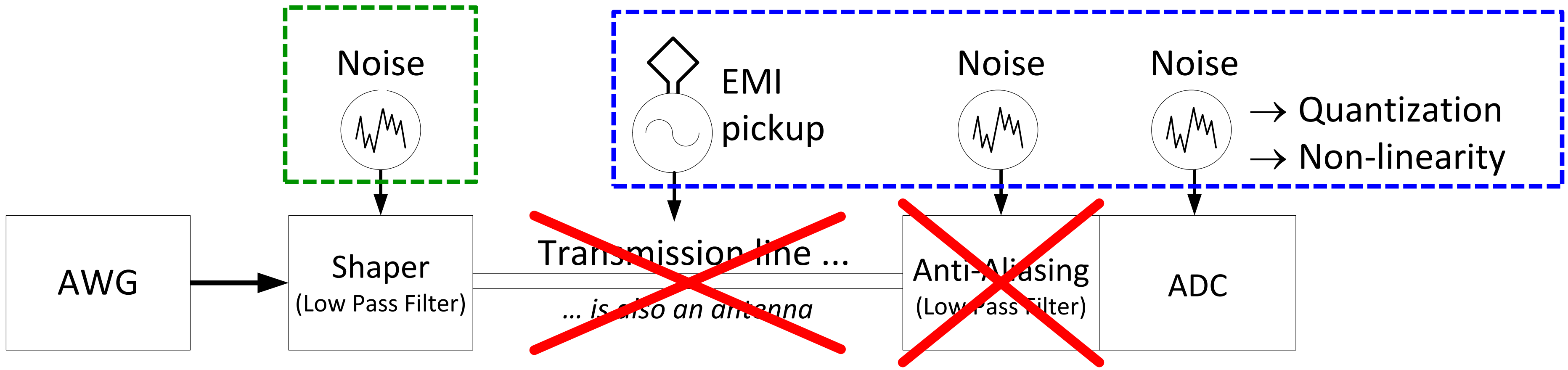


Legend

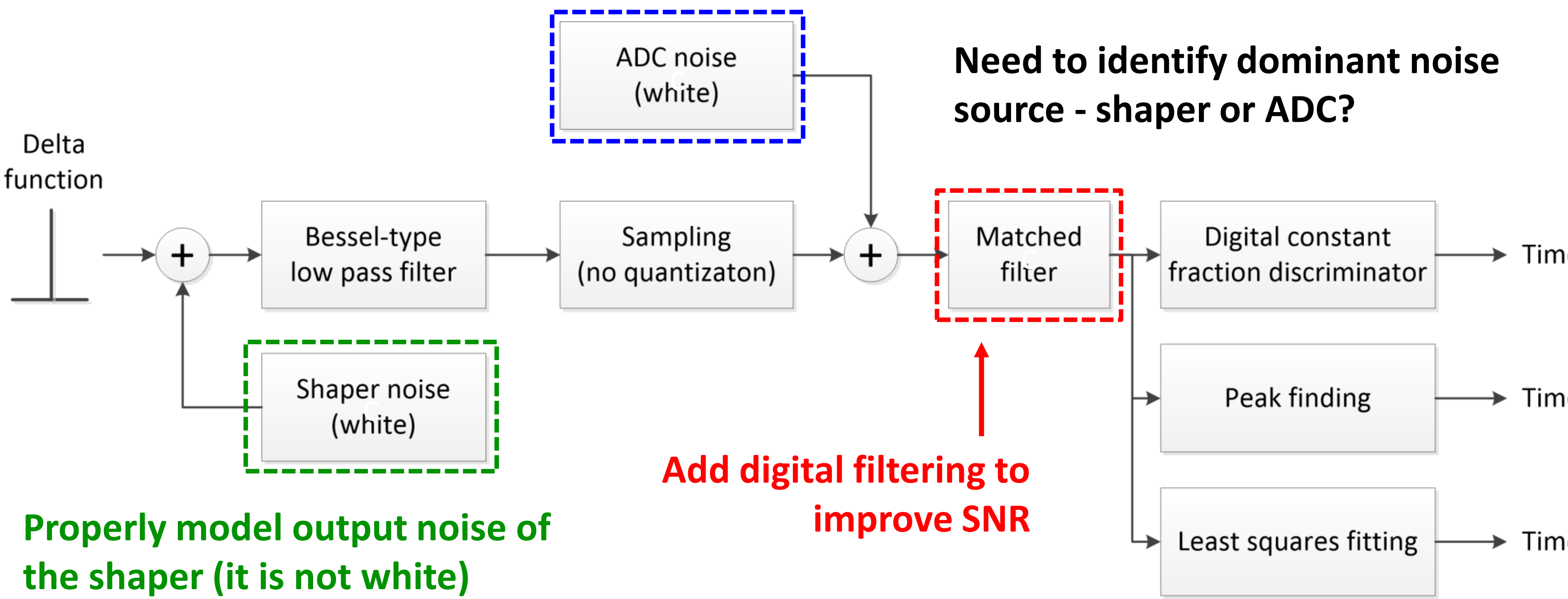


N_{BITS} (real)	10	12	14	16
SNR (1250 p.e.) [dB]	62	74	86	98
noise (p.e.)	1,00	0,25	0,06	0,02
SNR (1 p.e.) [dB]	0	12	24	36
N p.e. for SNR = 20 dB	10,0	2,5	0,6	0,2
N p.e. for SNR = 40 dB	99,7	24,9	6,2	1,6

Next Steps – Revised Model



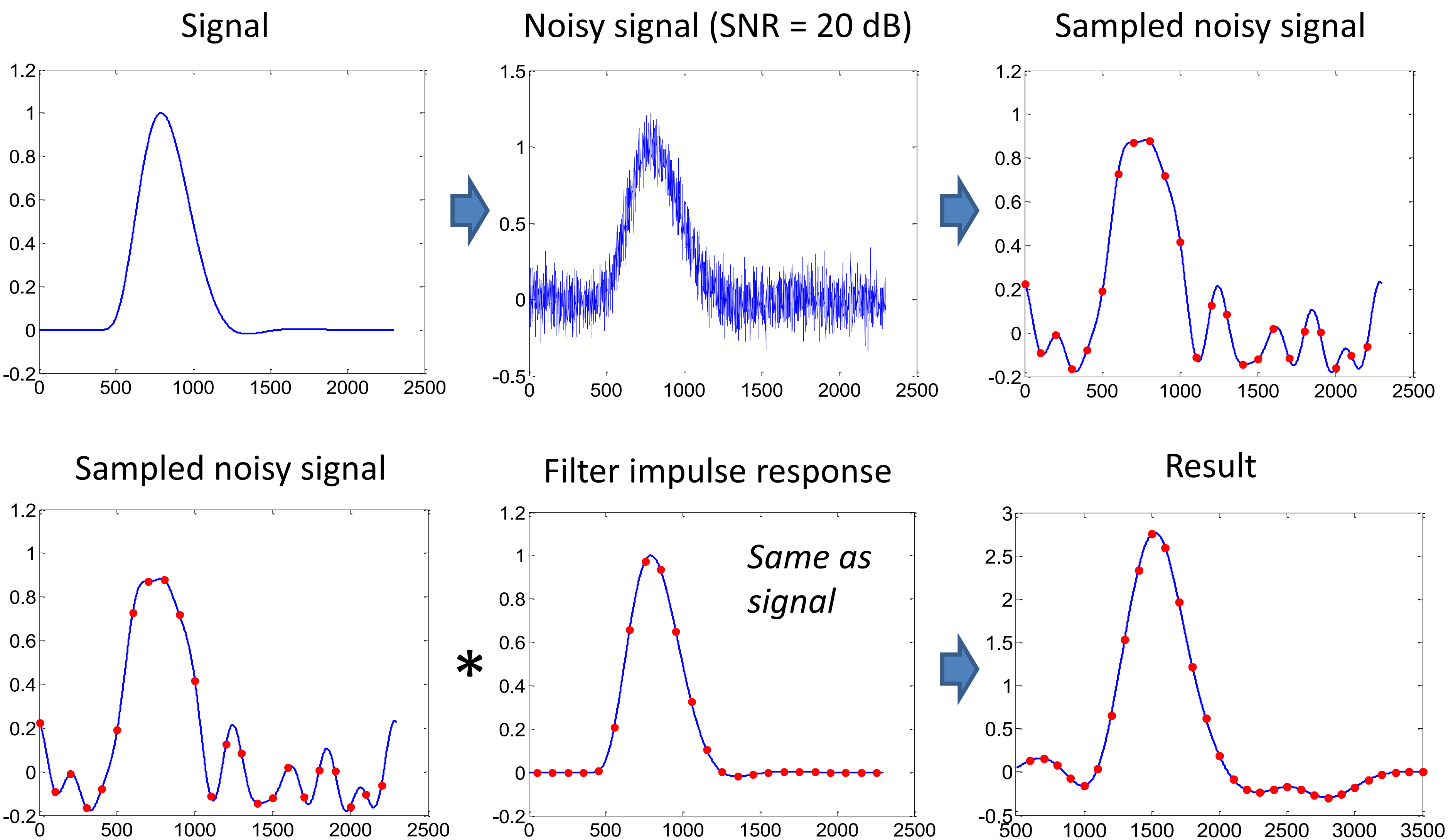
Need to identify dominant noise source - shaper or ADC?



Add digital filtering to improve SNR

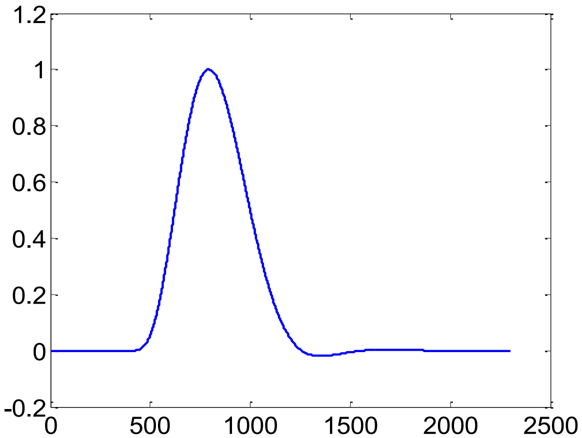
Properly model output noise of the shaper (it is not white)

Next Steps - Matched Filtering (1/2)

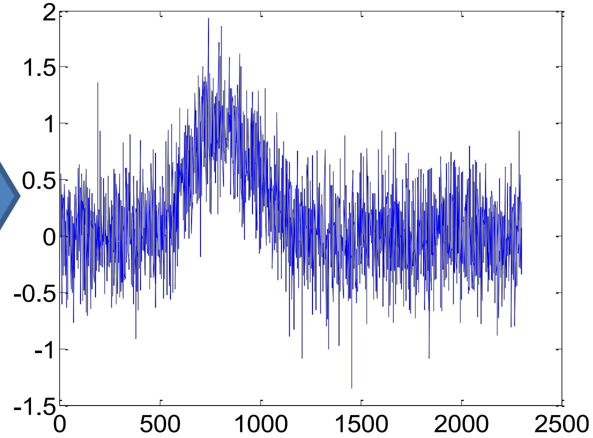


Next Steps - Matched Filtering (2/2)

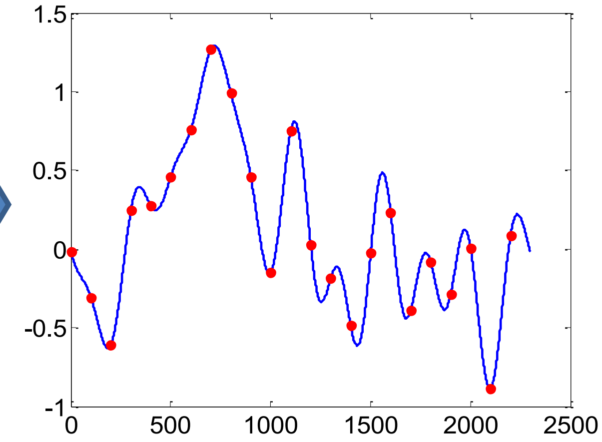
Signal



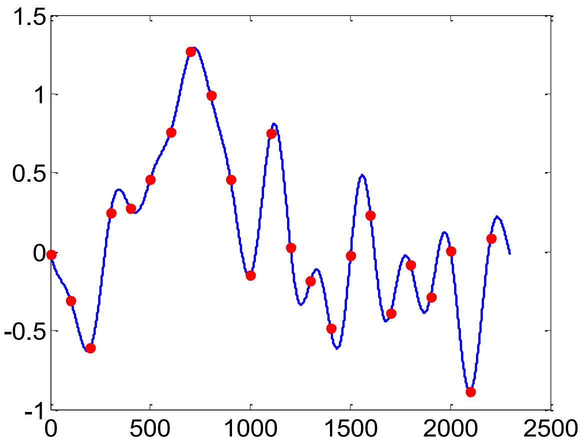
Noisy signal (SNR = 10 dB)



Sampled noisy signal

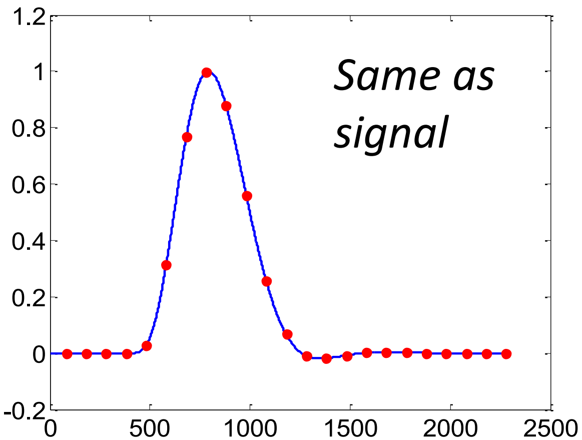


Sampled noisy signal

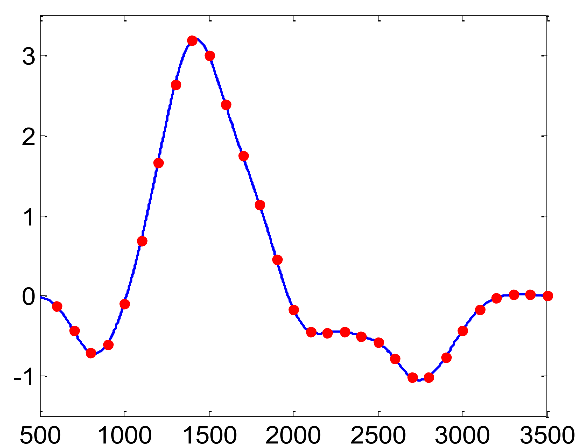


*

Filter impulse response



Result



Summary

- Design aids for shaper construction are ready
  - 4th or 5th order Bessel low-pass filter should be OK – it can be implemented using two amplifiers.
- Finalized analysis of digital constant fraction approach
  - Algorithm is extremely sensitive to SNR.
  - Relatively short pulses (1.5 sample at the rising edge) give better results at low SNR.
  - Algorithm is nice (because it is simple), but with pure CFD approach we will not achieve the required dynamic range and time resolution.
- New shaper designs done – for 100 MHz, 250 MHz and 500 MHz; 1.5, 2.0 and 3.0 samples at the rising edge.
- Prototypes assembled by TRIUMF, first data taken.
- Nest steps
  - Update noise models
  - Implement matched filtering into the simulation
  - Tune model to match experimental data taken at TRIUMF
- Need waveform-level modeling of the PMT response
  - Already have an algorithm (tested with H8711-10 & published)
  - Needs some tweaking