Possible Contribution to Electromagnetic Calorimeters

Feature Extraction from Waveforms and Data Reduction

Marcin Ziembicki

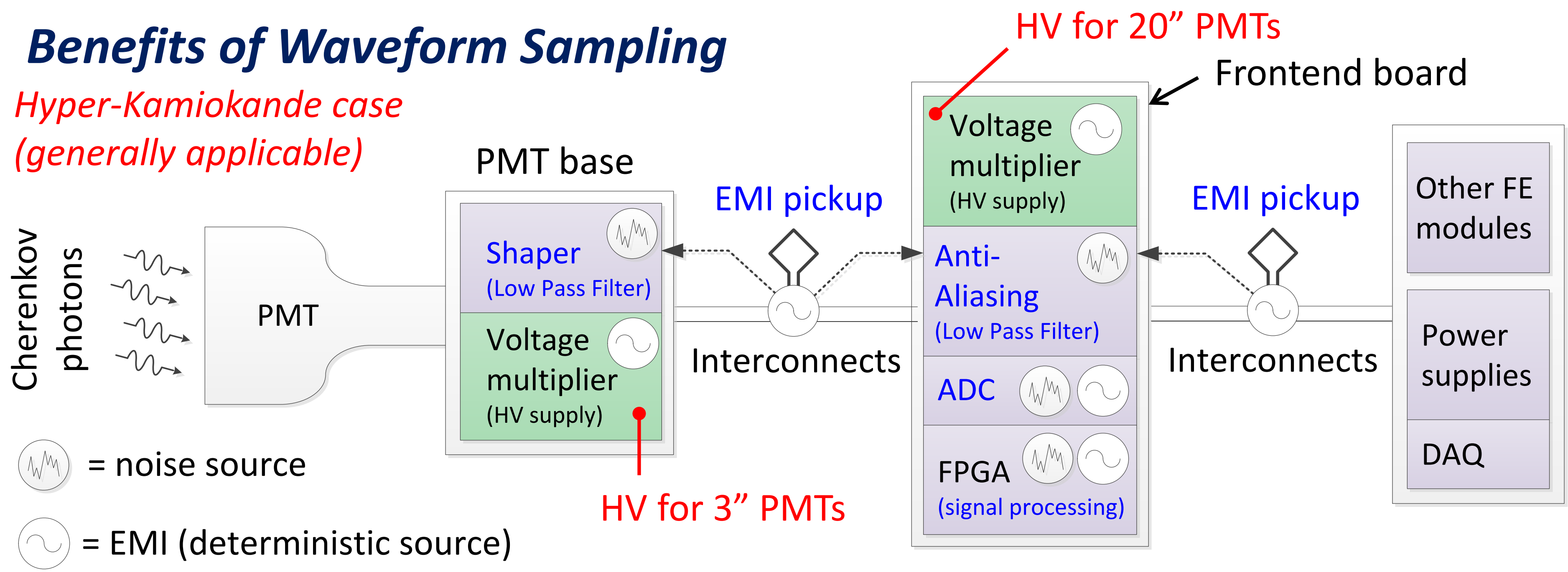
Warsaw University of Technology and AstroCeNT

SPD@NICA

JINR, Dubna, 2019-06-05

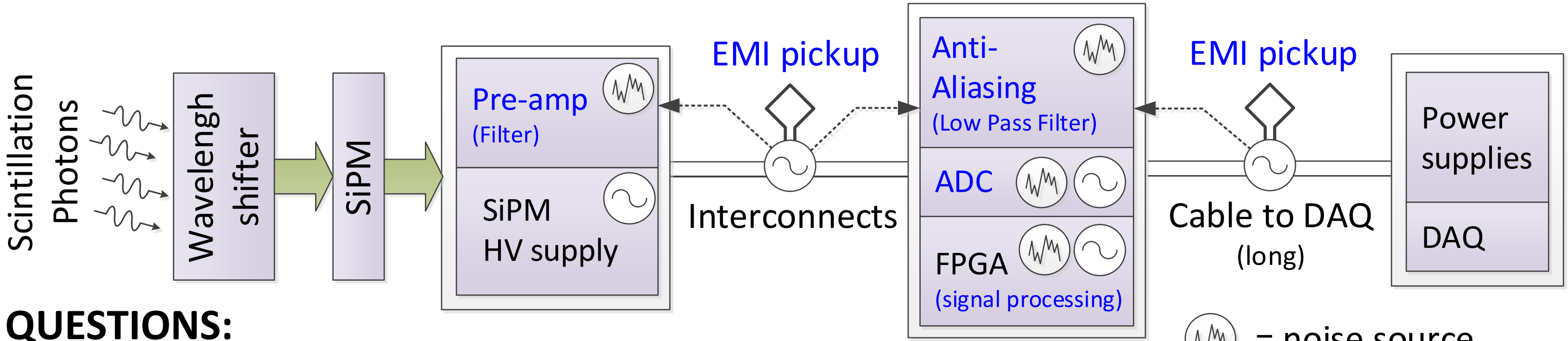
Introduction

Benefits of Waveform Sampling



- Possibility to implement completely dead-time free system.
- Ability to disentangle overlapping pulses (pile-up)
- **Can subtract off periodic EMI by digital filters implemented in FPGA firmware.**
- There is a price to pay: **power consumption, cost, data rate**.
  - Can we reduce the above without affecting the physics performance?

Optimizing the Signal Chain



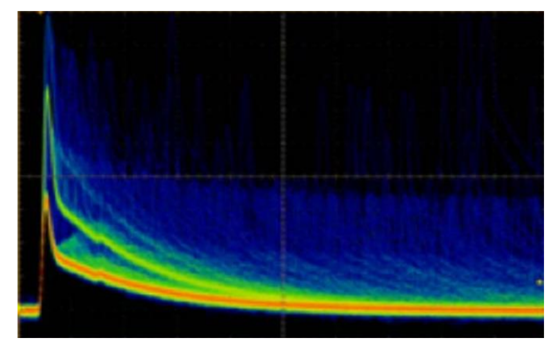
= noise source

= EMI (deterministic source)

QUESTIONS:

- Type and cutoff frequency of analog shaper/anti-aliasing filter?
- Speed and resolution of the ADC?
- Signal processing methods and sharing of signal processing between FPGA and DAQ
- Optimization of resource usage within the FPGA
- Quality of time & charge estimates
- Two independent compression methods:
  - Waveform (potentially lossy)
  - Time/charge (lossless)
- Disentanglement of pulse pile-up

Timing of arriving photons → leading edge



Source: Hamamatsu MPPC note

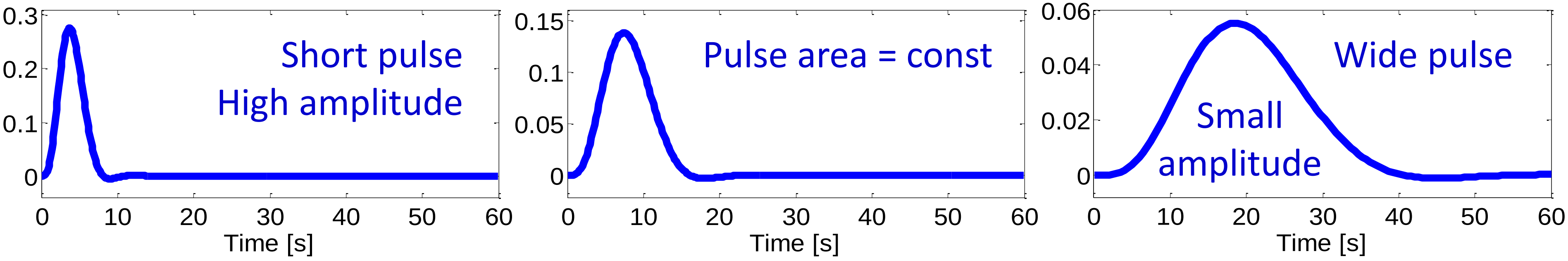
Recovery time → rate tolerance

Need decent model of the full signal chain → having one allows exploration of various variants of shaper/ADC combinations without the need for building prototypes (thus saves labor time)

Study of Sampling Systems

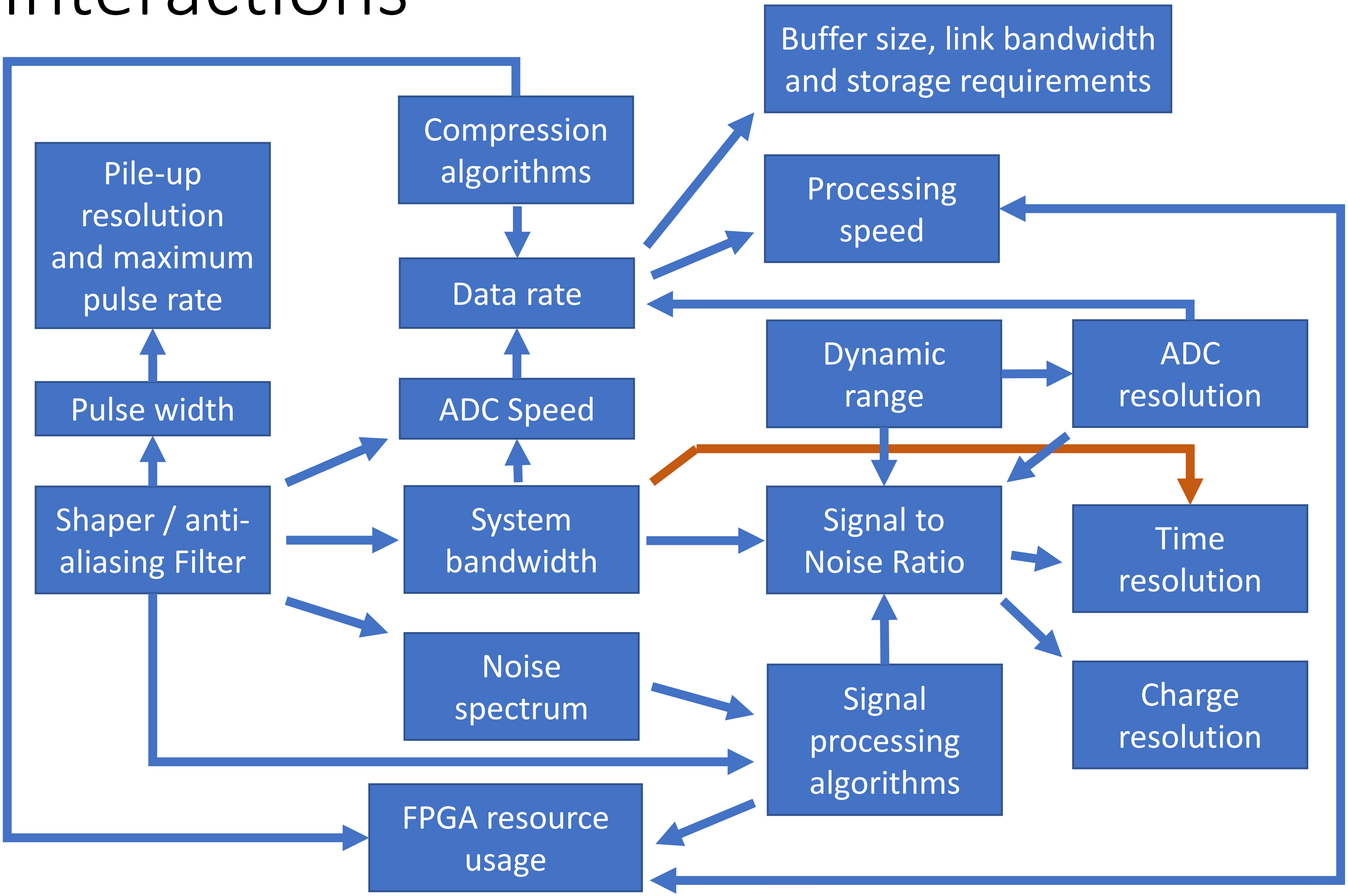
High resolution → Low resolution

How **poor** can the **system specs** be to still be able to tell **when** and how big the **pulse was** with **satisfactory precision**?



High bandwidth → Low bandwidth

Interactions



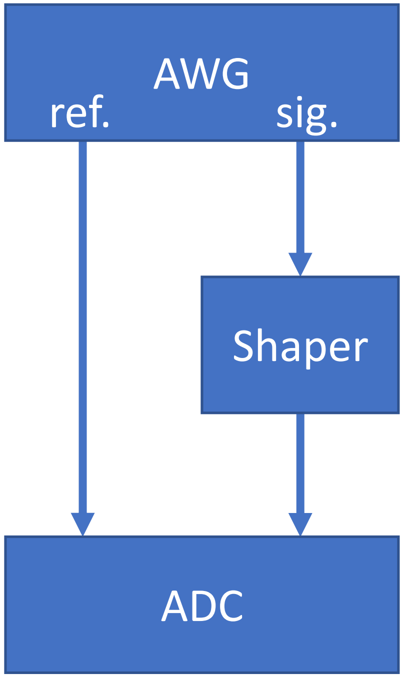
Multi-parameter constrained optimization problem

Timing Resolution of Sampling Digitizers

PURPOSE OF THE STUDY:

Determine how fast and how precise does a system needs to be to achieve given performance specs?

- Use AWG instead of PMT.
- Use large reference pulse (timing accuracy $\sigma \approx 10$ ps) and small, shaped signal pulse (1 mV ~ 100 mV).
- Apply signal processing methods and calculate time difference Δt between ref. and sig. channels.
- Repeat multiple times and compute RMS of Δt values.
- Two shapers:
  - 15 ns and 30 ns rise time (10% to 90%), 5-th order **Bessel-type** low-pass filters.
- Shared project WUT/TRIUMF



Agilent 33600A (1 GSPS/80 MHz)

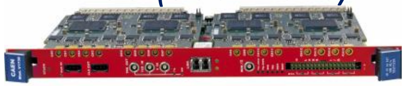
Custom shapers

Commercial ADCs (CAEN)

DT5724 (100 MSPS/14b)

V1720 (250 MSPS/12b)

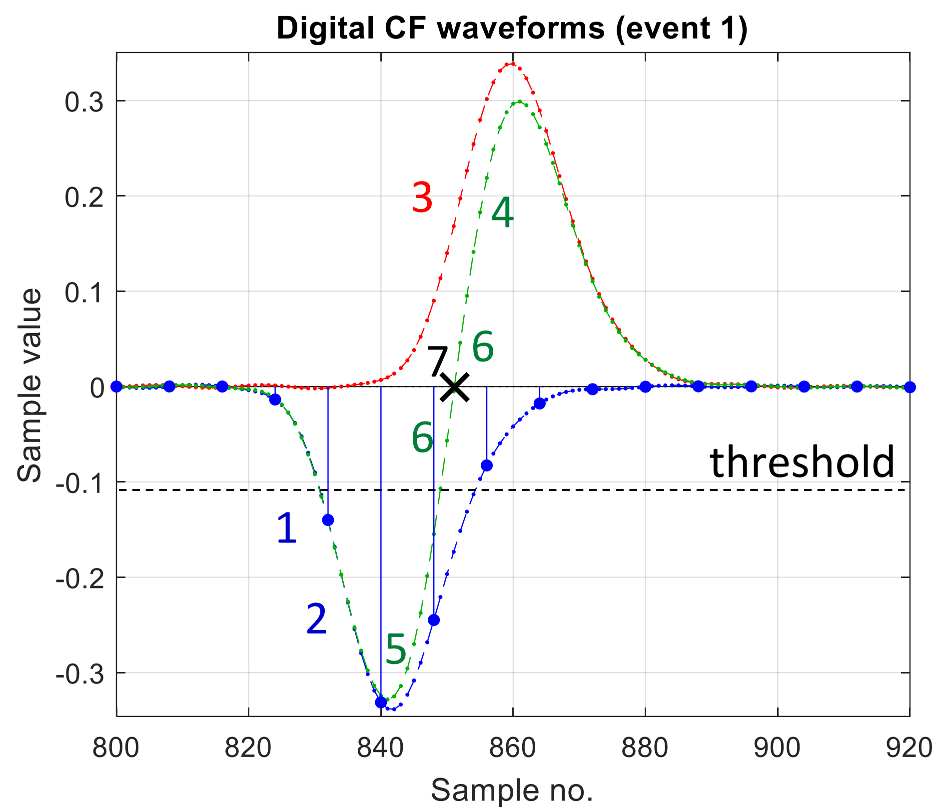
V1730 (500 MSPS/14b)



Signal Processing Methods

Digital Constant Fraction Discriminator:

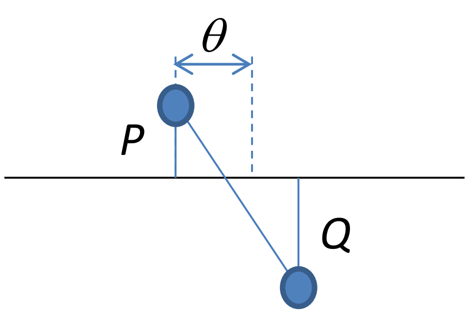
- Simple processing → needs little FPGA resources
- Does not make any assumption as to the pulse shape
- Favors high sampling rate, but some improvements are possible for low sampling rates if pulse shape is invariant
- Poor performance in low SNR conditions



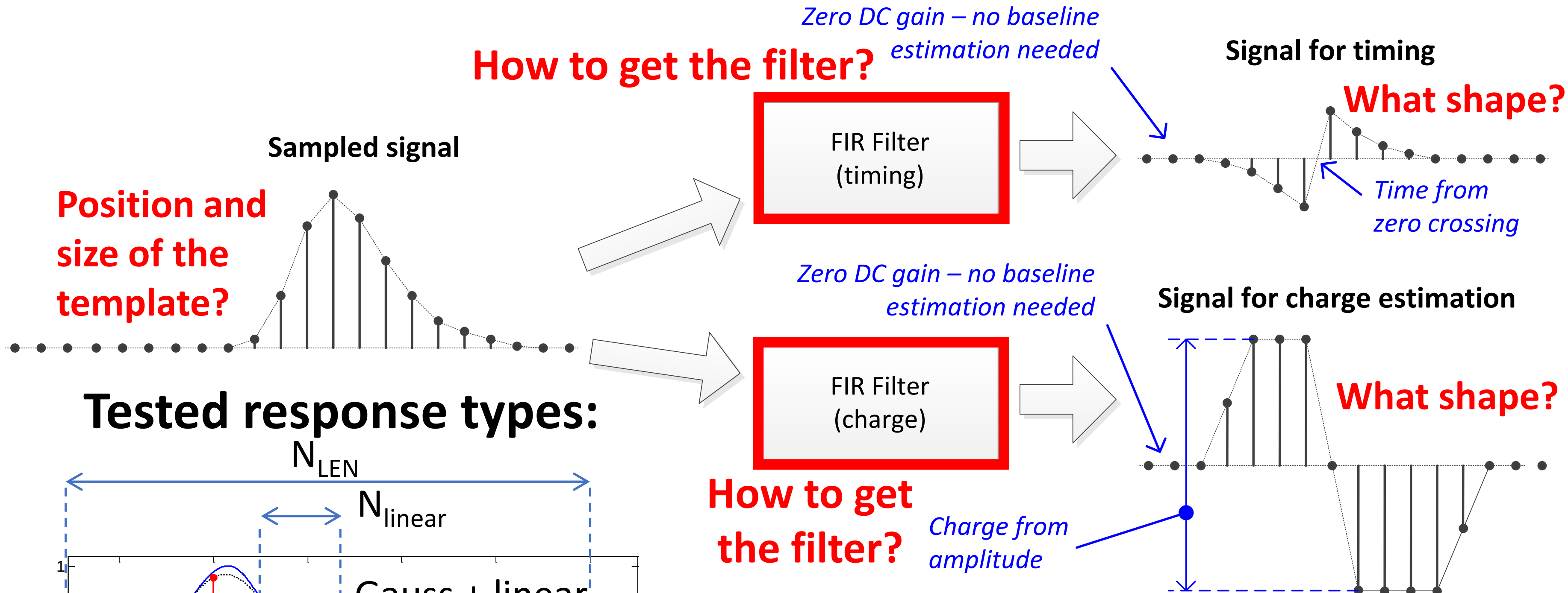
Time errors and possible correction →

θ - actual sub-sample shift

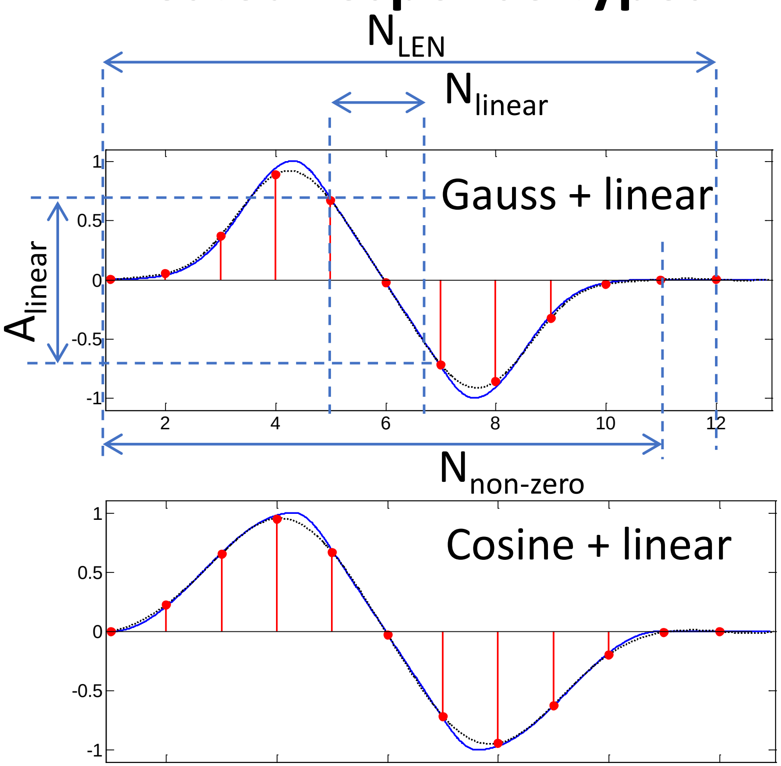
$$CR = \frac{P}{P - Q}$$



Signal Processing – FIR DPLMS



Tested response types:

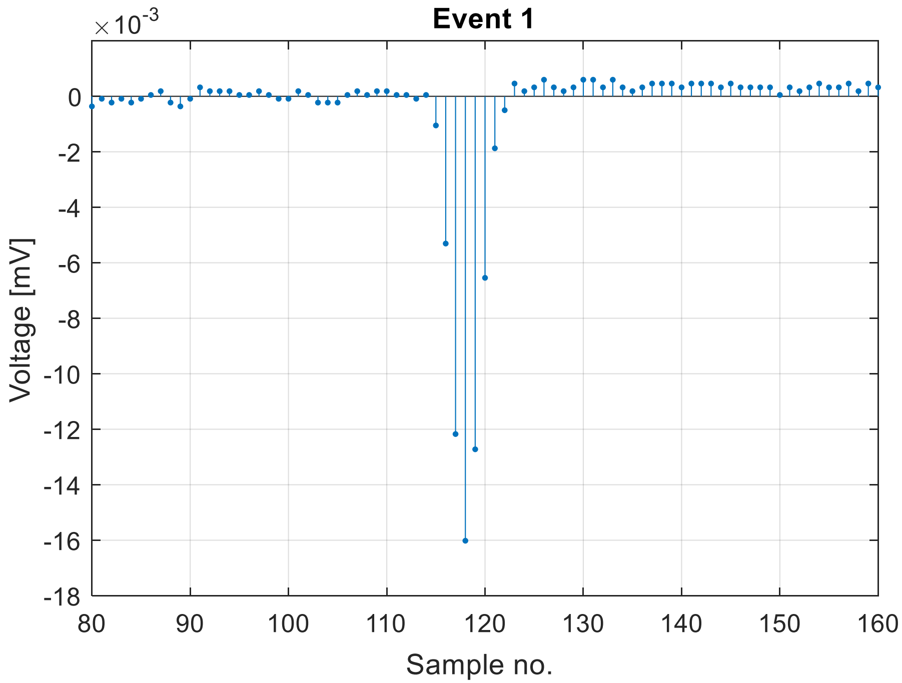


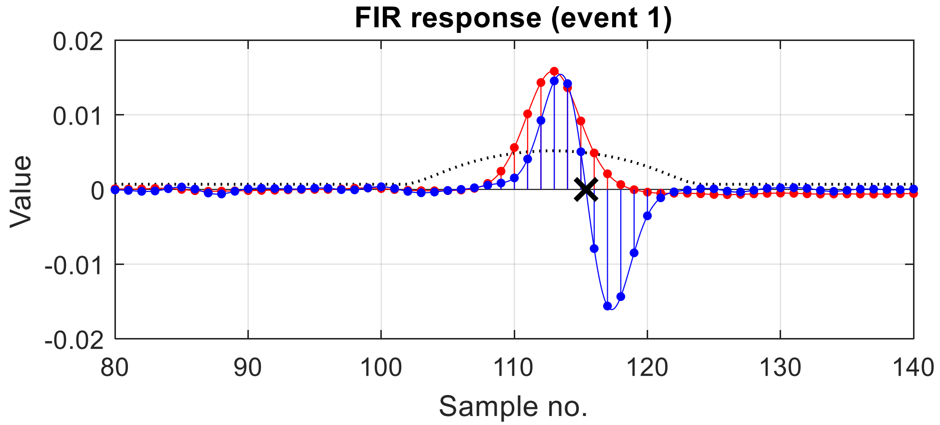
… or simply subtract pedestal and integrate.

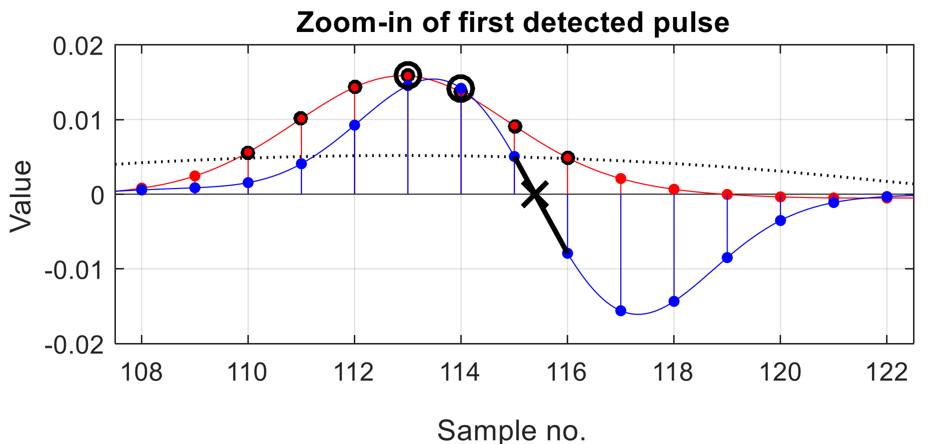
- FIR = Finite Impulse Response
- 'Black-box' approach → transform **known** input into desired output, don't care how.
- Arbitrary filter characteristic possible.
- Filter should be 'optimal' → **minimize certain cost function (constrained optimization)**.

Gatti E., et al., "*Digital Penalized LMS method for filter synthesis with arbitrary constraints and noise*", NIM A523, 167-185, 2004

Signal Processing - FIR Filters







- Trigger on matched filter response (red)
- Use adaptive threshold to prevent false positives (dotted black line)
  - Average signal to get the threshold and delay FIR processing to check for pulses and their timing
- Get time using the ‘timing’ filter (blue)
- Apply correction to counteract non-linear shape of the waveform near zero-crossing.

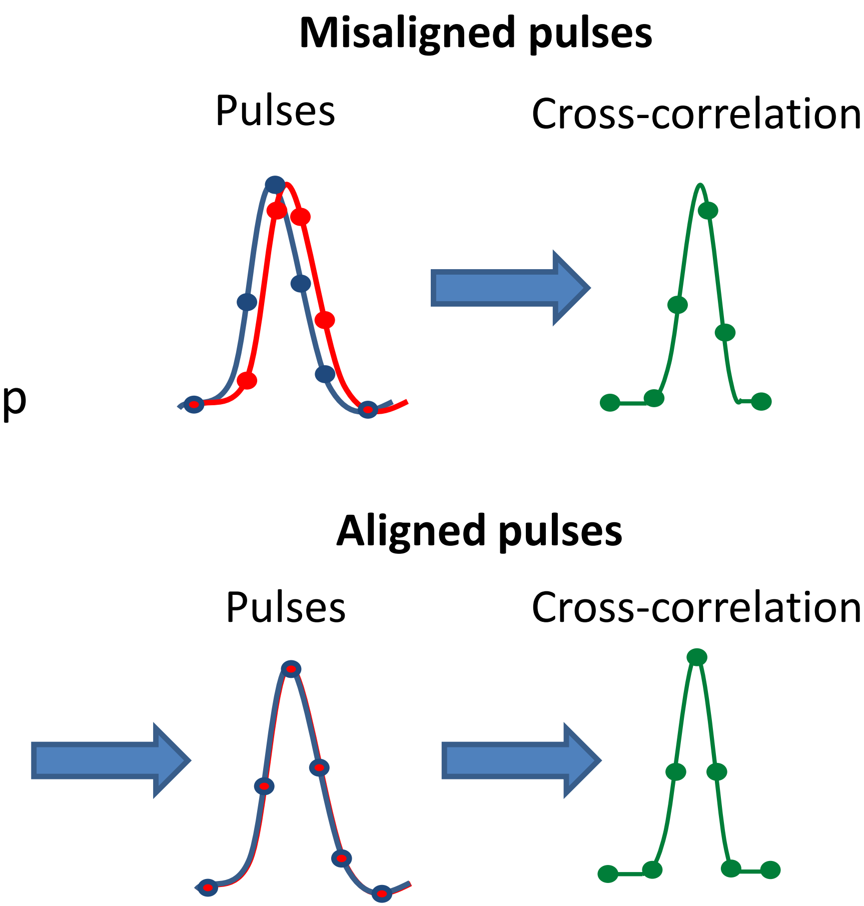
Method assumes that shape is constant

Need on-line Quality Factor to judge accuracy of estimation

Signal Processing – Continued

Matched FIR Filter and Cross-Correlation Processing:

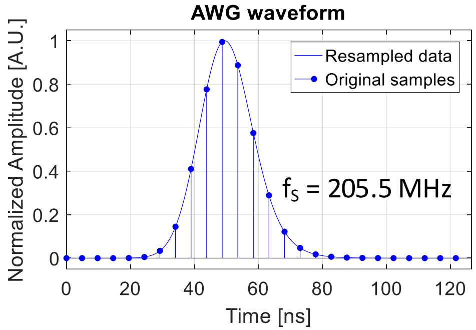
- Much more complex processing
  - Works well with filter orders of 9-12
- **Assumes that shape is constant**
- Similar timing performance to zero-average FIR filter
- Relatively easy to disentangle piled-up pulses



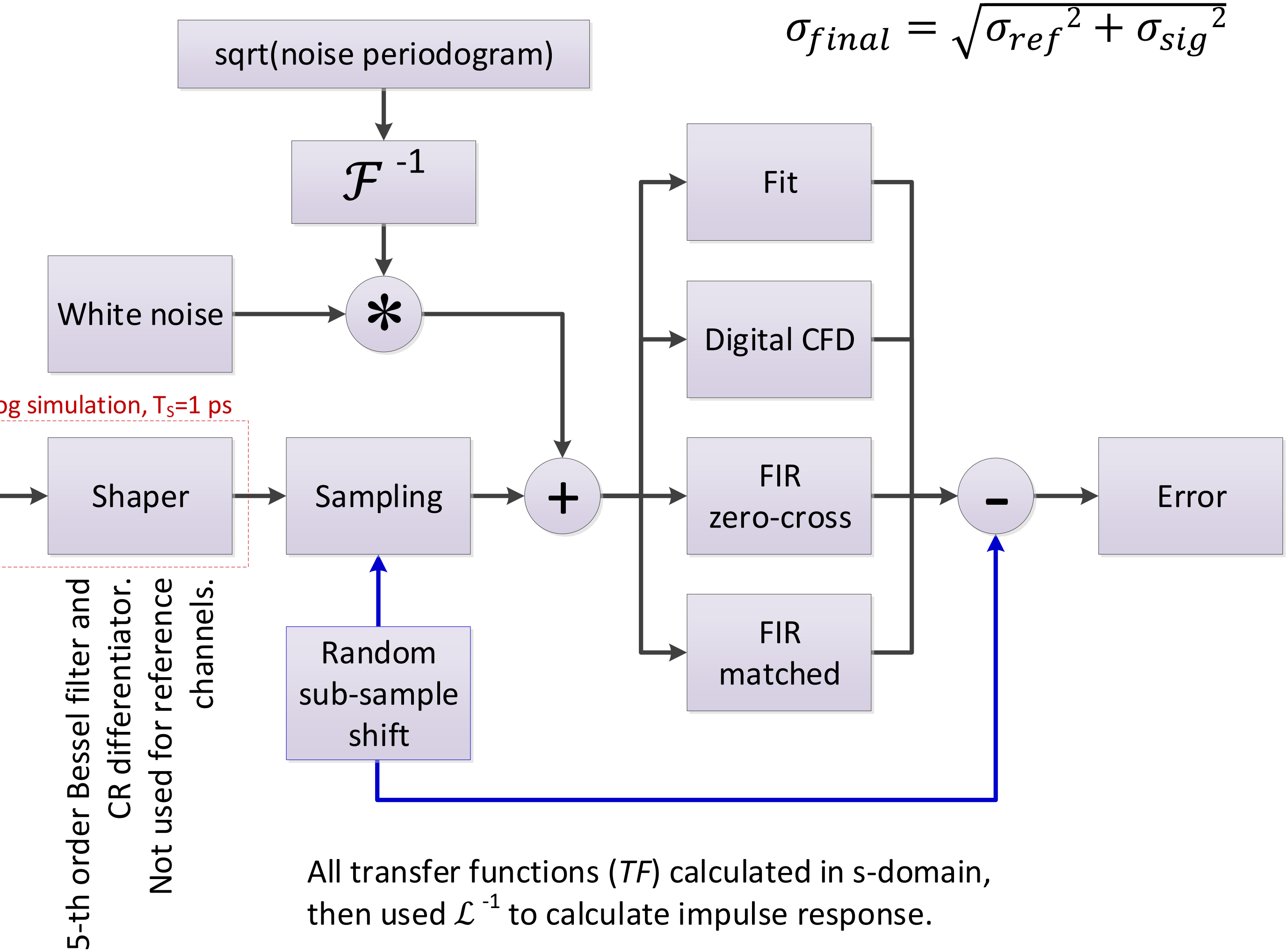
Sub-sample shifts done using windowed sinc interpolation (Blackman window). FFT interpolation also possible if shifting impulse response.

System Model (each channel)

Used 250 MHz data to determine actual AWG f_S

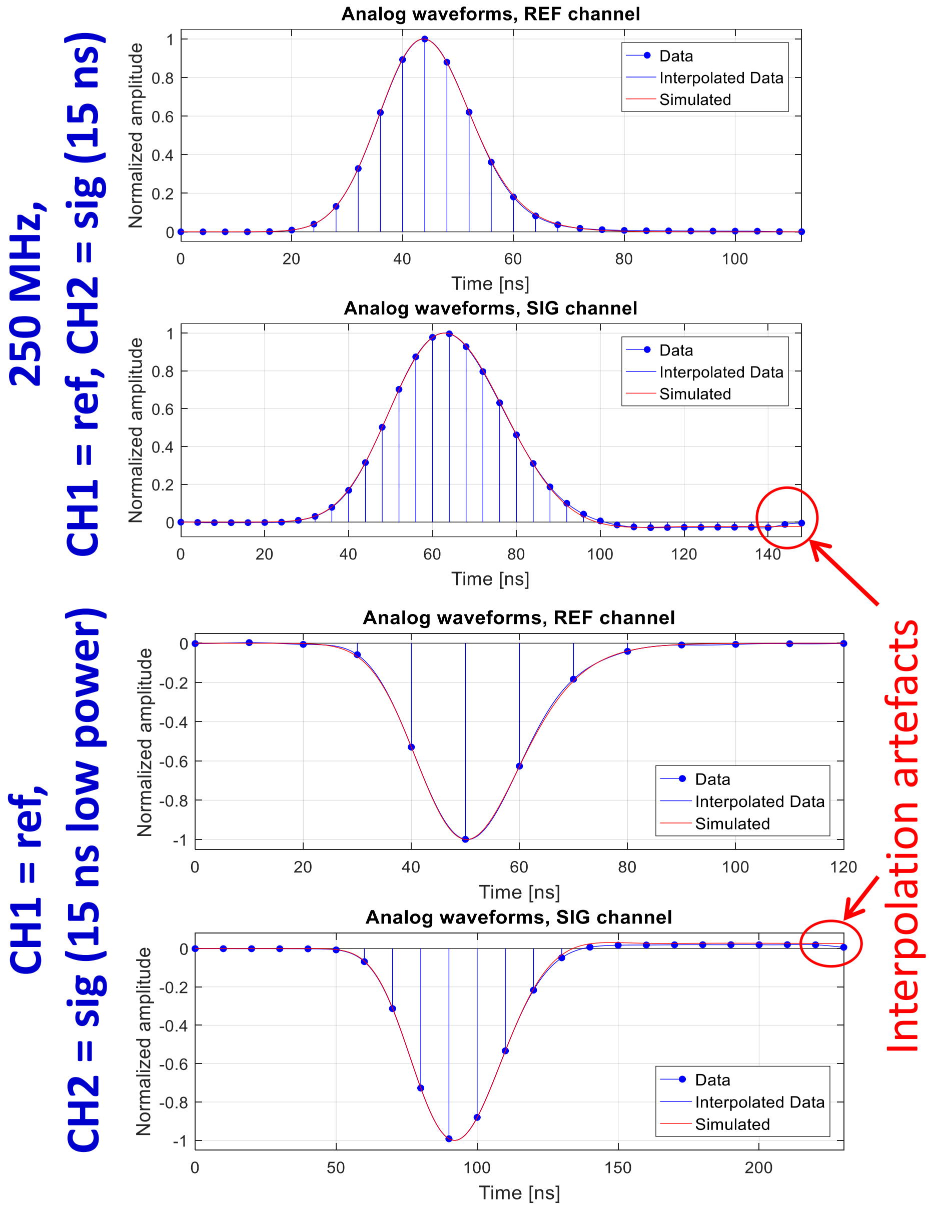


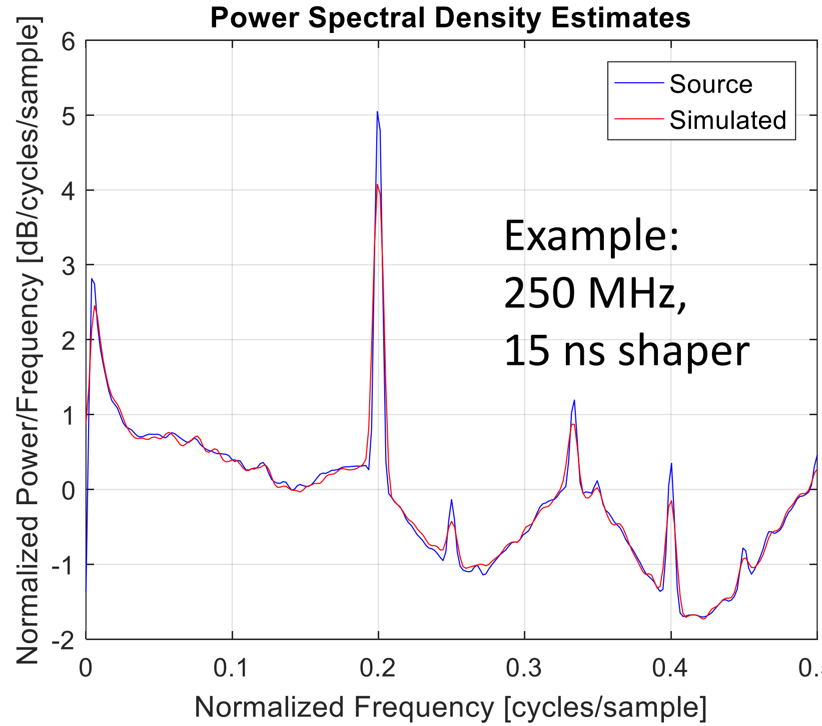
$$\sigma_{final} = \sqrt{\sigma_{ref}^2 + \sigma_{sig}^2}$$



All transfer functions (*TF*) calculated in s-domain, then used $\mathcal{L}^{-1}$ to calculate impulse response.

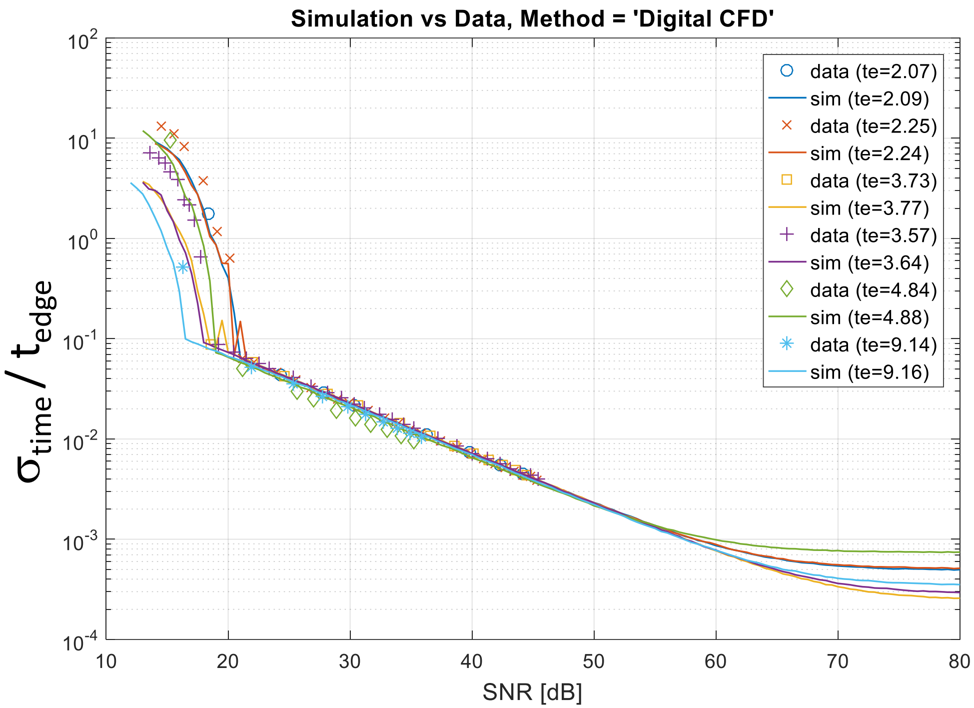
Signal and Noise Models

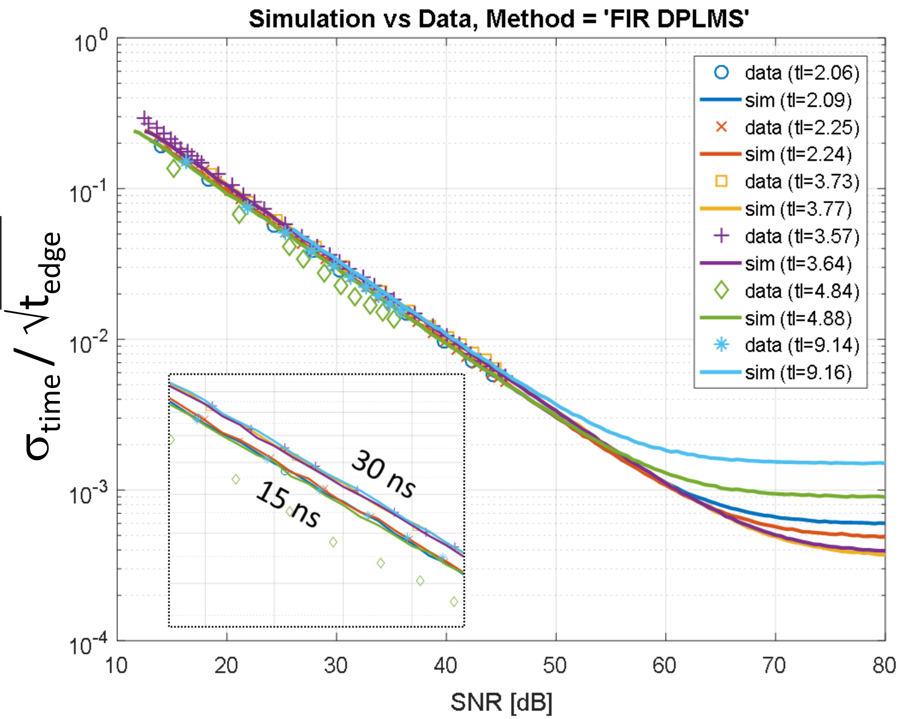




- Good match of simulated periodogram with an experimental one.
- Potential problem:
  - Some of the deterministic components (peaks in spectrum) do not have random phase, but are correlated to the sampling clock.

Digital CFD / FIR DPLMS – Normalized





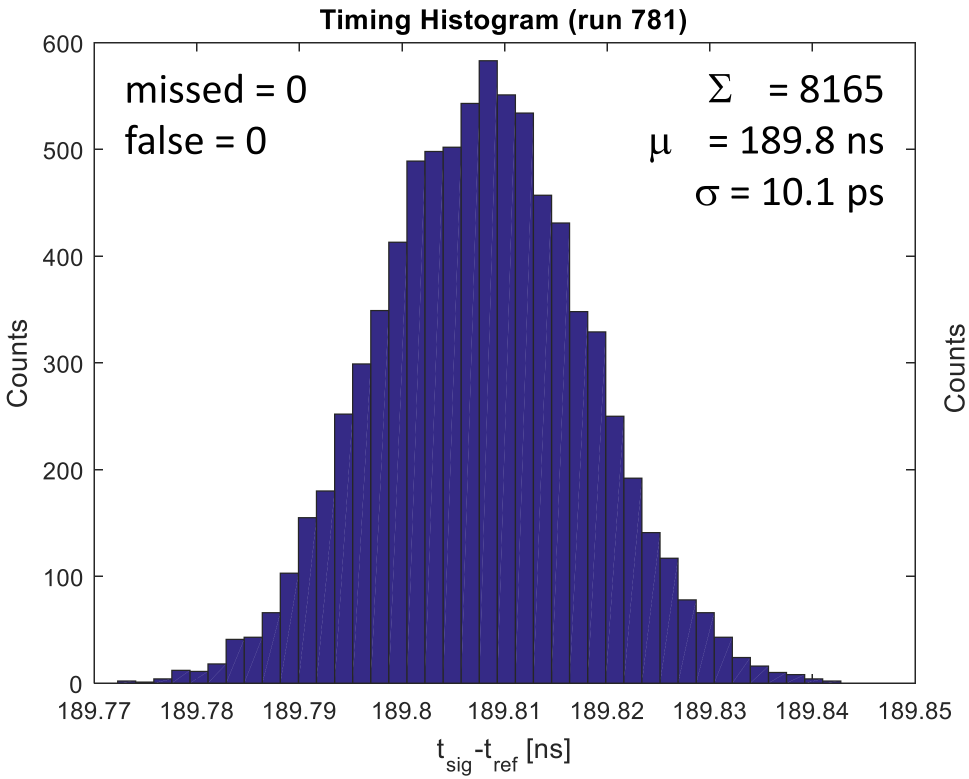
- Don't need extremely high sampling rates to maintain good timing resolution, as long as SNR is sufficient
- It seems that it is better to maintain sharp edge → logical, as we don't cut bandwidth of the signal that still has valid information
  - Sharp edges help in pile-up resolution
- Oversampling help only in case of FIR-based algorithms → SNR gets better

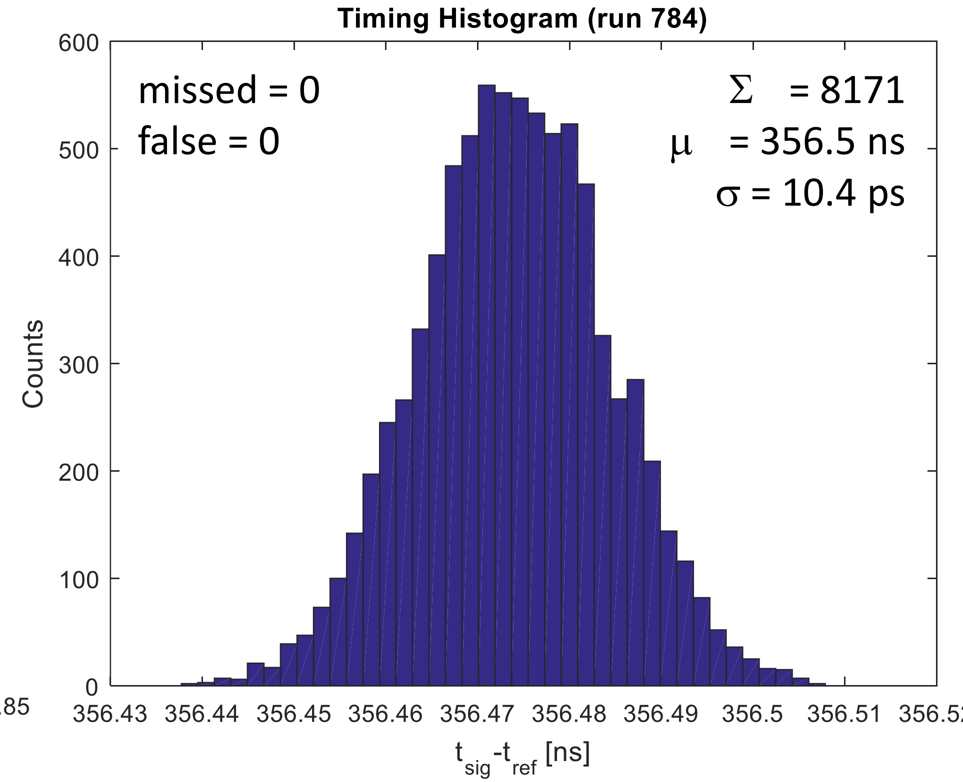
Optimization criteria:

$$\frac{\boldsymbol{t_{rise}}}{\boldsymbol{SNR}}$$

Example Histograms – FIR Timing

Large SNR case (approx. 60 dB)





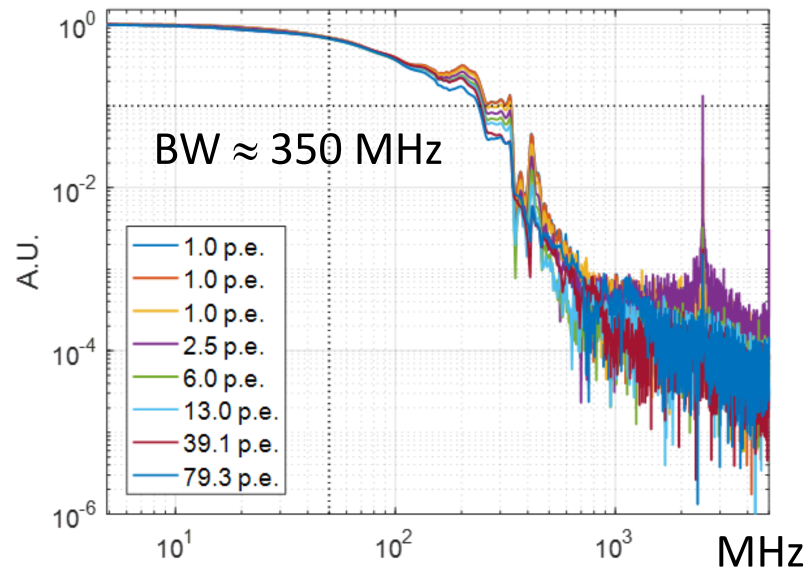
100 MSPS ADC, 14-bit, no shaper (left), 15 ns shaper (right)

10 ps resolution from a system with 10 ns sampling

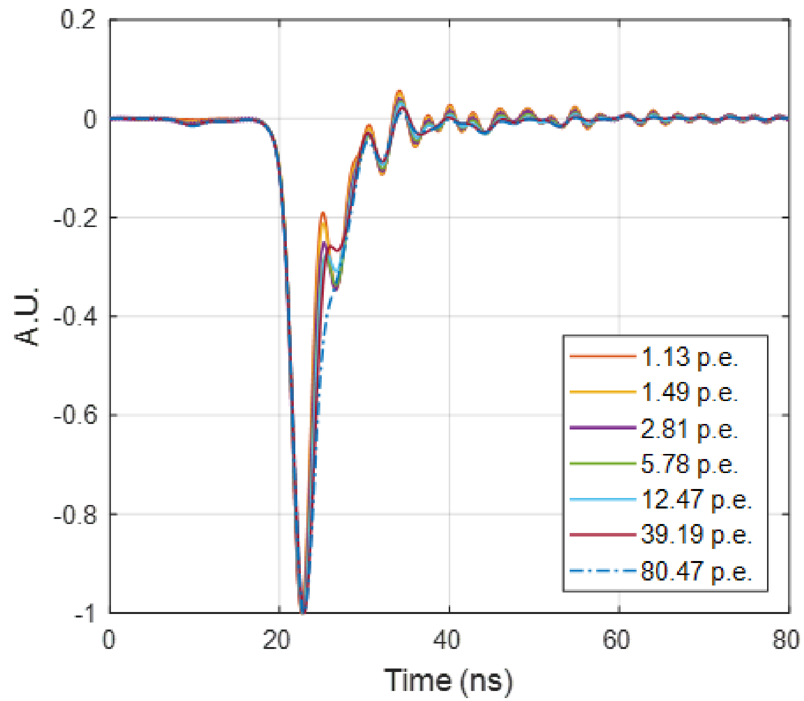
Photosensor - R14374

- Visible dependence of waveform shape on position of the light source on the photocathode
- $t_{rise} \in$ (1.9 ns, 3.0 ns), FWHM $\in$ (3.0 ns, 4.7 ns); both increase with PE level (expected)
- Not a lot of change in spectra density in the 'recorded' bandwidth → good news!

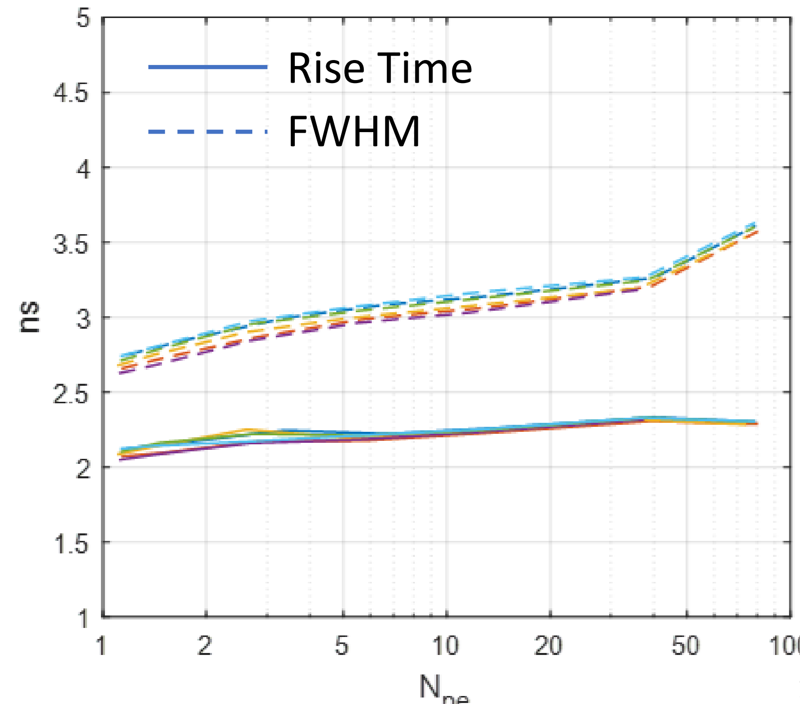
Normalized Amplitude Spectrum



Normalized templates

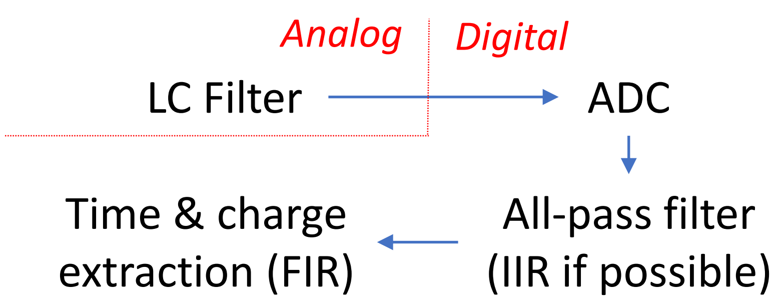


Rise Time & FWHM

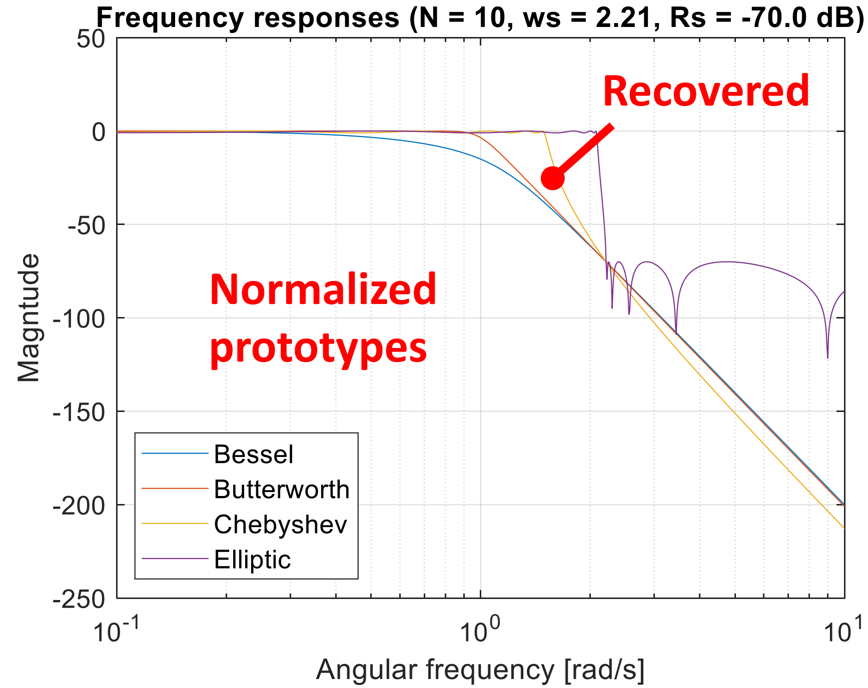


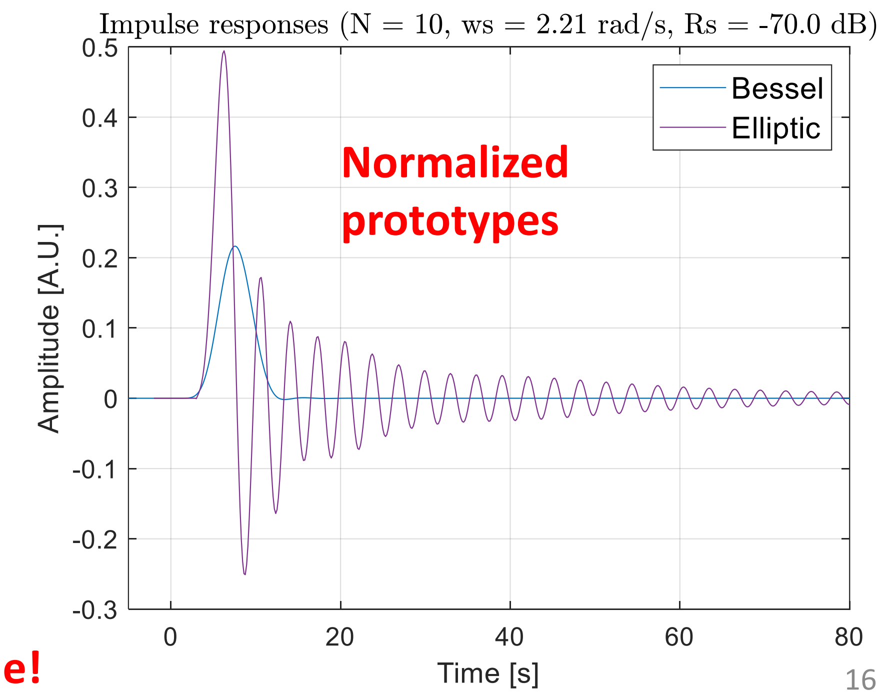
Where are we now?

- Prototype in July
- Re-designed the shaper
  - Old shaper used for tests was too noisy, had too low cutoff frequency
  - Decided to switch to fully passive design (LC-ladder) – still need one amplifier to separate LC circuit from the twisted pair
  - Investigating possibility to switch from Bessel to a filter with a sharper roll-off
- Need additional digital all-pass filter to correct passband ripple and phase



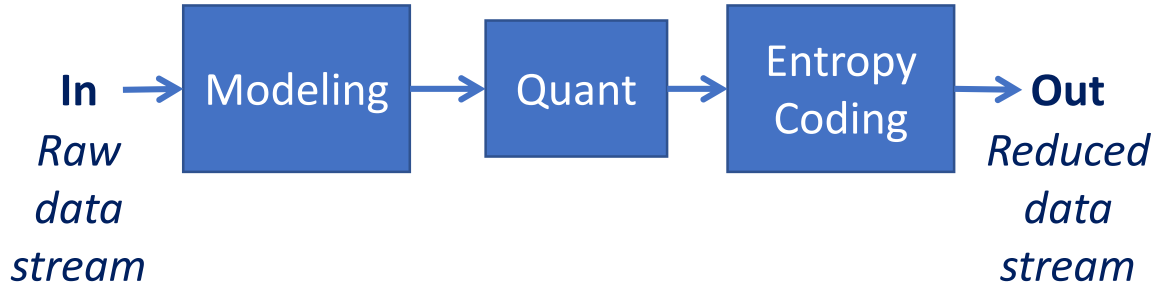
Check needed if this is possible!





Compression Studies

- Modeling
  - Linear Prediction
  - Signal Models
  - Transforms
- Quantization
  - Scalar quantization
  - Vector quantization – using signal models
- Entropy Coding
  - Variable length coding
  - Arithmetic coding – more complex and better compression



- Lossless Coding of waveforms
  - Compression ratio: about 2-6
  - Depends on SNR, sampling frequency, signal dynamics
- Lossy Coding of waveforms
  - Compression ratio: more than 3, e.g. 10, 20 …
  - Distortion (D) and bit rate (R) depend on quantization step
  - RD Tradeoff
  - Allowable losses should be lower than signal noise

TWO INDEPENDENT COMPRESSIONS

- Time/charge data (lossless)
- Waveforms (lossless or lossy)

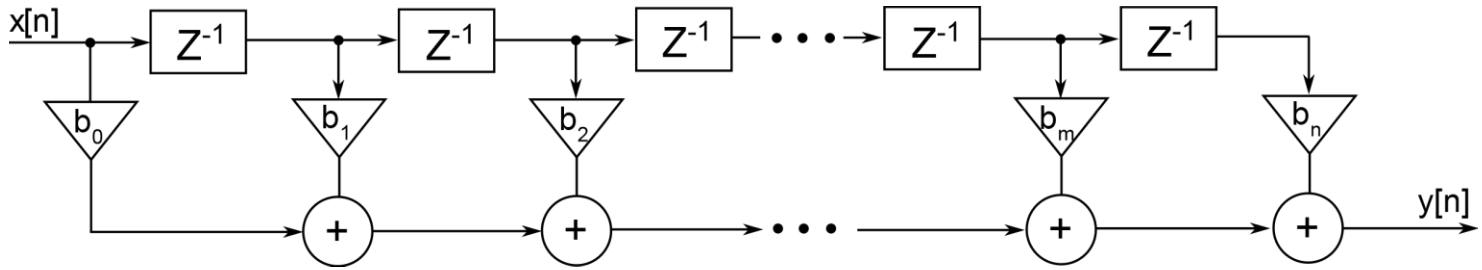
Preliminary results on Super-Kamiokande data

→ Time/charge data

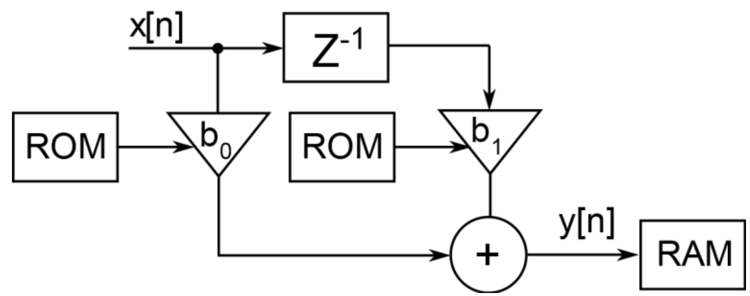
→ **1:1.6** reduction (**1:2** reduction within reach)

Filter Implementation

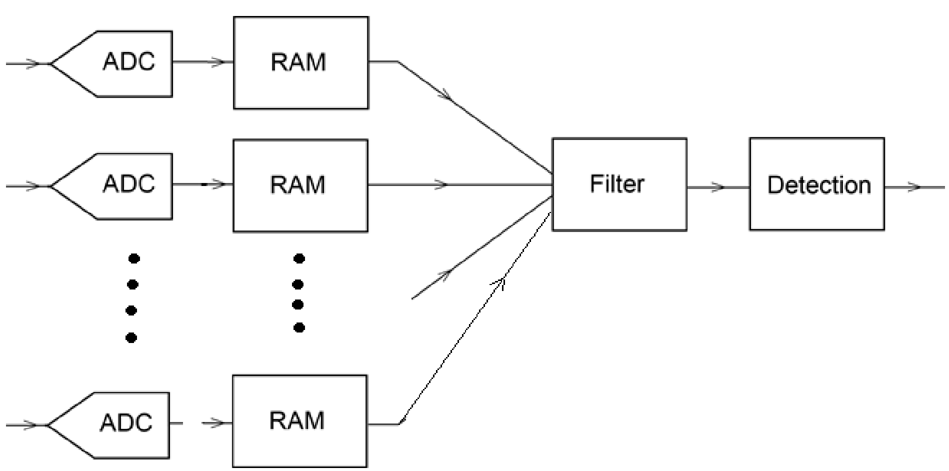
Direct imple-mentation



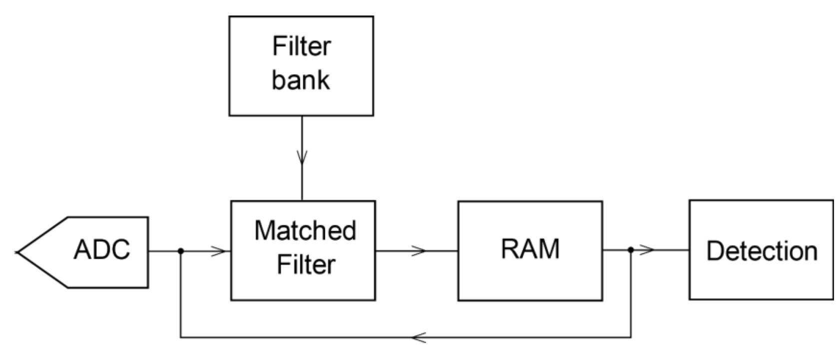
FPGA runs with a faster clock than ADC, so multiple cycles possible for one sampling period → multiplex FIR processing in time



Filter sharing among channels



Filter sharing for different coefficients

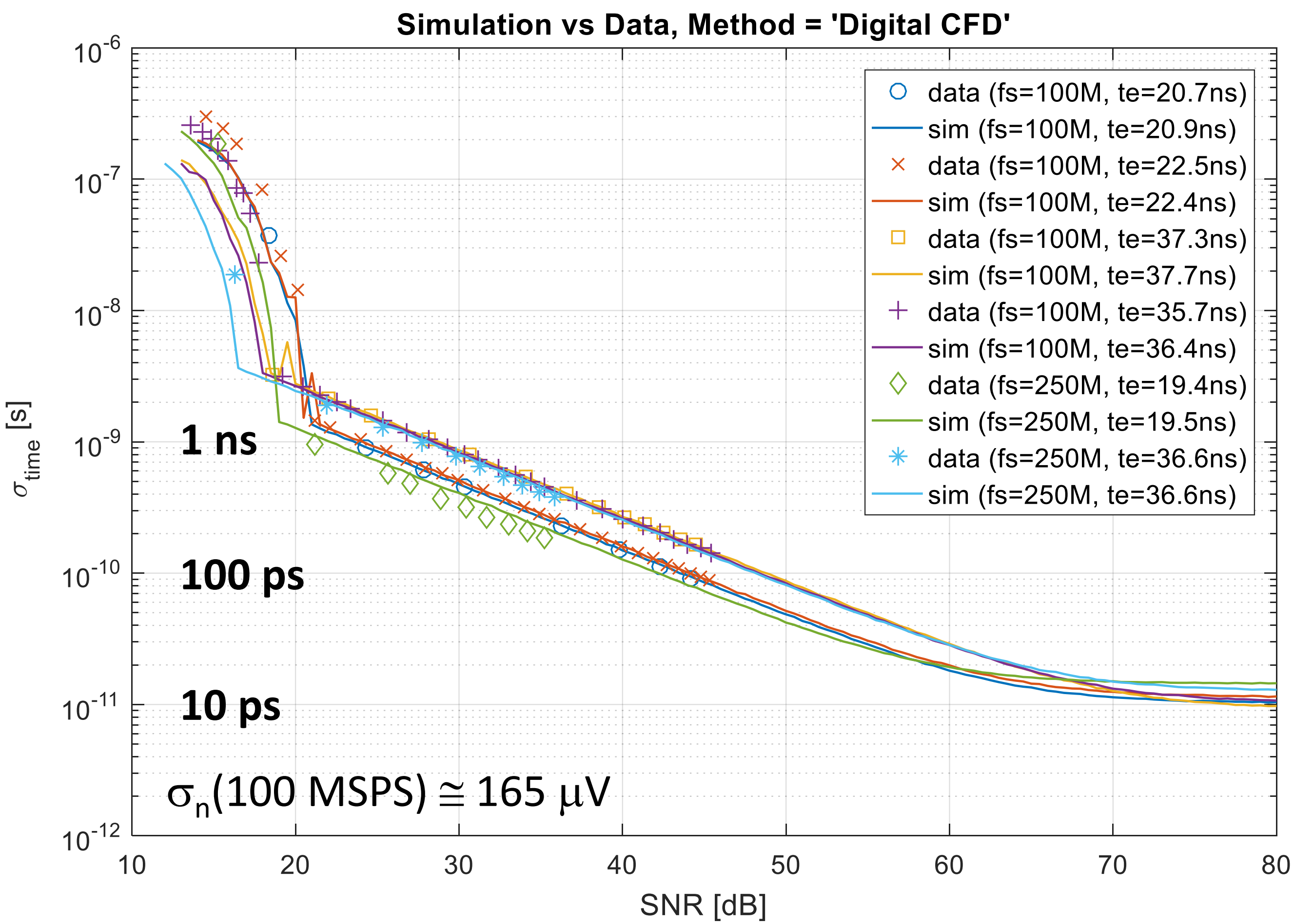


Summary

- Much work already done, even more still to do
  - ‘Attacking’ problem from multiple angles
- Prototype foreseen in July
- Need to foresee that in FIR-based methods the estimate may be completely wrong in case of non-standard shape (for ex. pile-up)
  - Need quality factor for each time/charge estimate
  - Should send full waveform for off-line processing
- We’re also involved in photosensor characterization
  - Can’t design good electronics without understanding signal source
- Closely working with the TRIUMF laboratory and TU Munich
- Recently teamed up with INFN Trieste
  - They made spectroscopy system using the same filtering approach, but optimizing amplitude resolution $\rightarrow$ complementary our efforts so far
- Planning beam test sometime in November

BACKUP

Results – Digital CFD



mV → 0.5 1.7 5.2 16.5 52.3 165.3 523 1653

SNR ≥ 20 dB

Good match of model and data for 100 MHz ADC, slightly worse for 250 MHz ADC

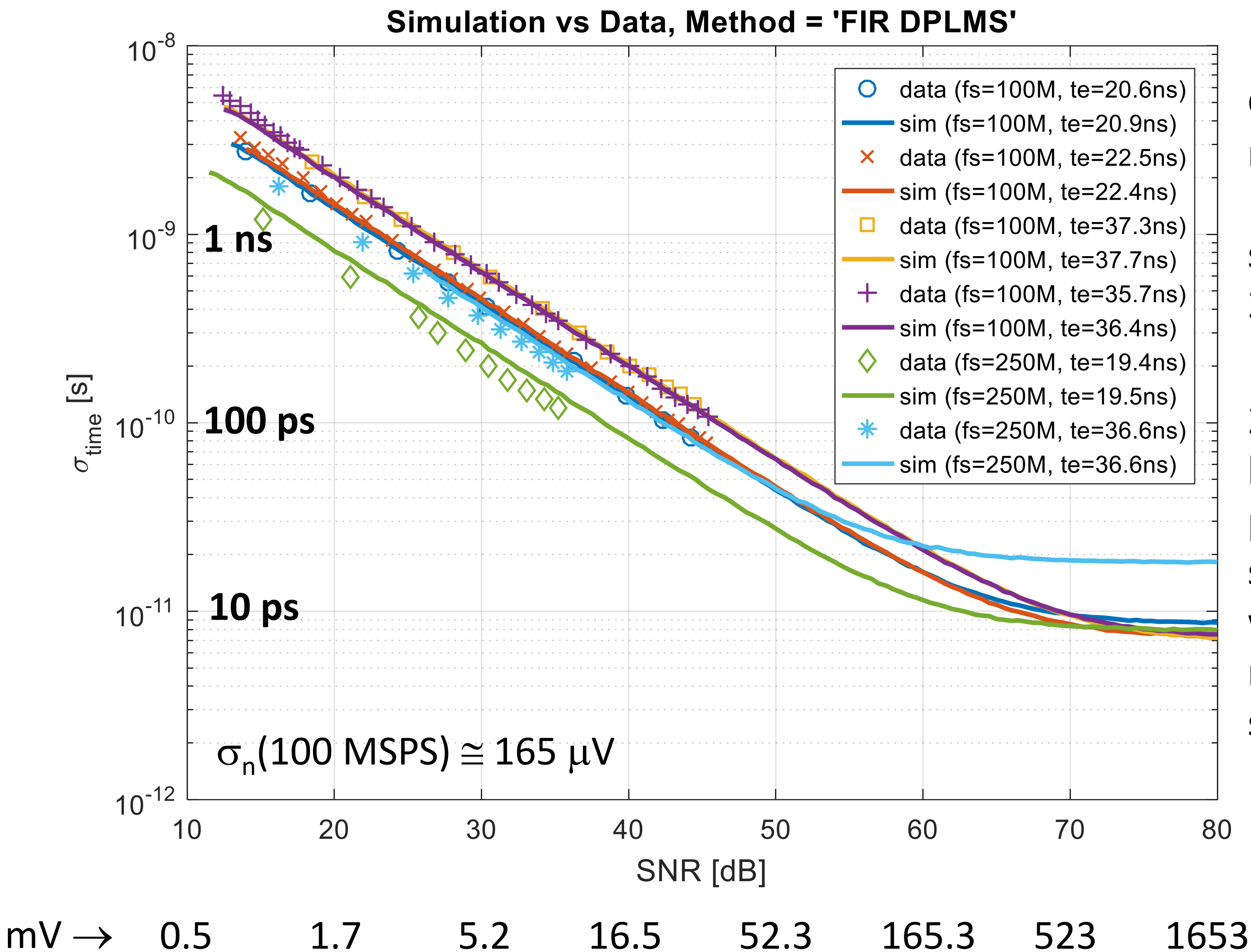
SNR < 20 dB

Poor match, data worse than model. Not a useful range anyway, as we need σ_{time} < 1 ns.

Timing resolution is proportional to

$$\frac{t_{rise}}{SNR}$$

Results – FIR DPLMS

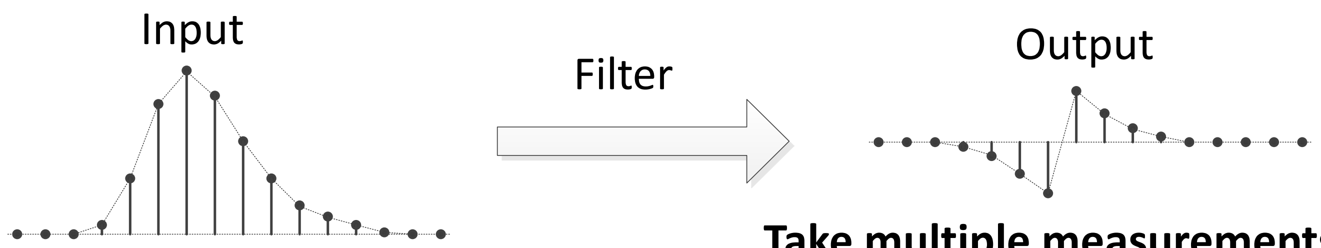


Good match of model and data for 100 MHz ADC, slightly worse for 250 MHz ADC

250 MHz data better than model – possibly due to some correlation which is not reflected by simulation.

Synthesizing FIR filter – Method 1

Digital Penalized LMS Method



input signal — *noiseless signal (our template)* — *stationary noise*

$$x[n] = x'[n] + x"[n]$$

number of filter taps — *impulse response of the filter*

Filter is **linear**, so the output signal is:

$$y[n] = \sum_{l=0}^{N-1} h[l] \cdot x'[n-l] + \sum_{l=0}^{N-1} h[l] \cdot x"[n-l]$$

Therefore, we can deal with noise and signal components separately

Take multiple measurements, then:

Minimize overall variance of the response:

Sought filter

$$Var(y) = \boldsymbol{h}^{1,N} \cdot \boldsymbol{R}^{N,N} \cdot \boldsymbol{h}^{N,1}$$

Noise auto-covariance matrix

Minimize difference between filter response and our desired response

$$(E(y[k] - v_k))^2 = (\boldsymbol{h}^{1,N} \cdot \boldsymbol{x}'(k)^{N,1} - v_k)^2$$

Value of k-th sample of the response to x'

N past samples of x', starting from k

Gatti E., et al., "*Digital Penalized LMS method for filter synthesis with arbitrary constraints and noise*", NIM A523, 167-185, 2004

Synthesizing FIR filter – Method 1 (cont.)

Digital Penalized LMS Method

Add additional constraints for frequency response, including gain at DC ...

Add constraints related to bit-gain (i.e. how well we are supposed to reject quantization noise) ...

Finally, build the error functional and minimize it:

$$Area(FIR) = \frac{Area(y)}{Area(x)}$$

Constraint for variance

Constraints for shape of response to pulse template

Frequency constraints

$$\varepsilon^2 = Var(y) + \sum_{k=1}^{N} \alpha_k \cdot (E(y[k]) - v_k)^2 + \sum_{l=1}^{L} \beta_l \cdot \left(|F\{\boldsymbol{h}\}|_{\omega_l}\right)^2$$

$$+ \varphi \cdot (F\{\boldsymbol{h}\}_{\omega=0} - Area(FIR))^2 + \gamma \cdot \sum_{n} (h[n])^2$$

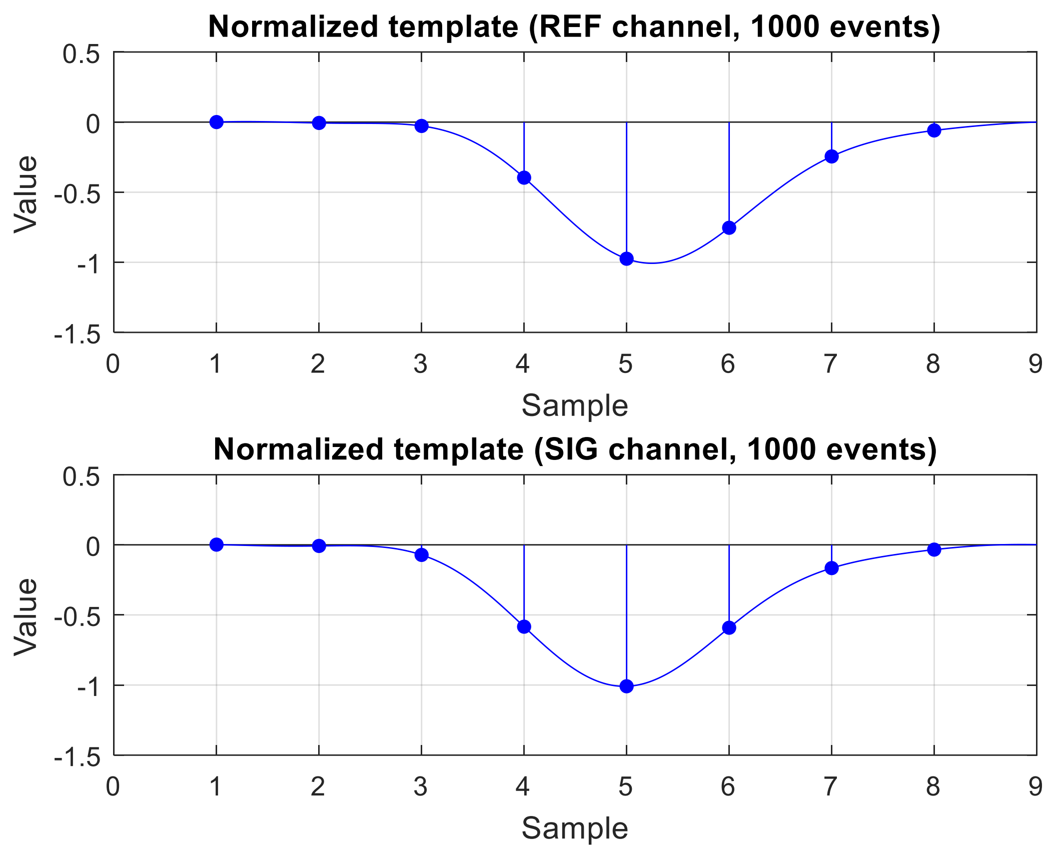
DC gain (i.e. area) constraint

Bit-gain constraint

All components are square functions, so there exists a global minimum – just need to properly choose $\boldsymbol{N}$, $\vec{v}$, $\vec{\alpha}$, $\vec{\beta}$, φ and γ → papers don't say much about that

FIR synthesis

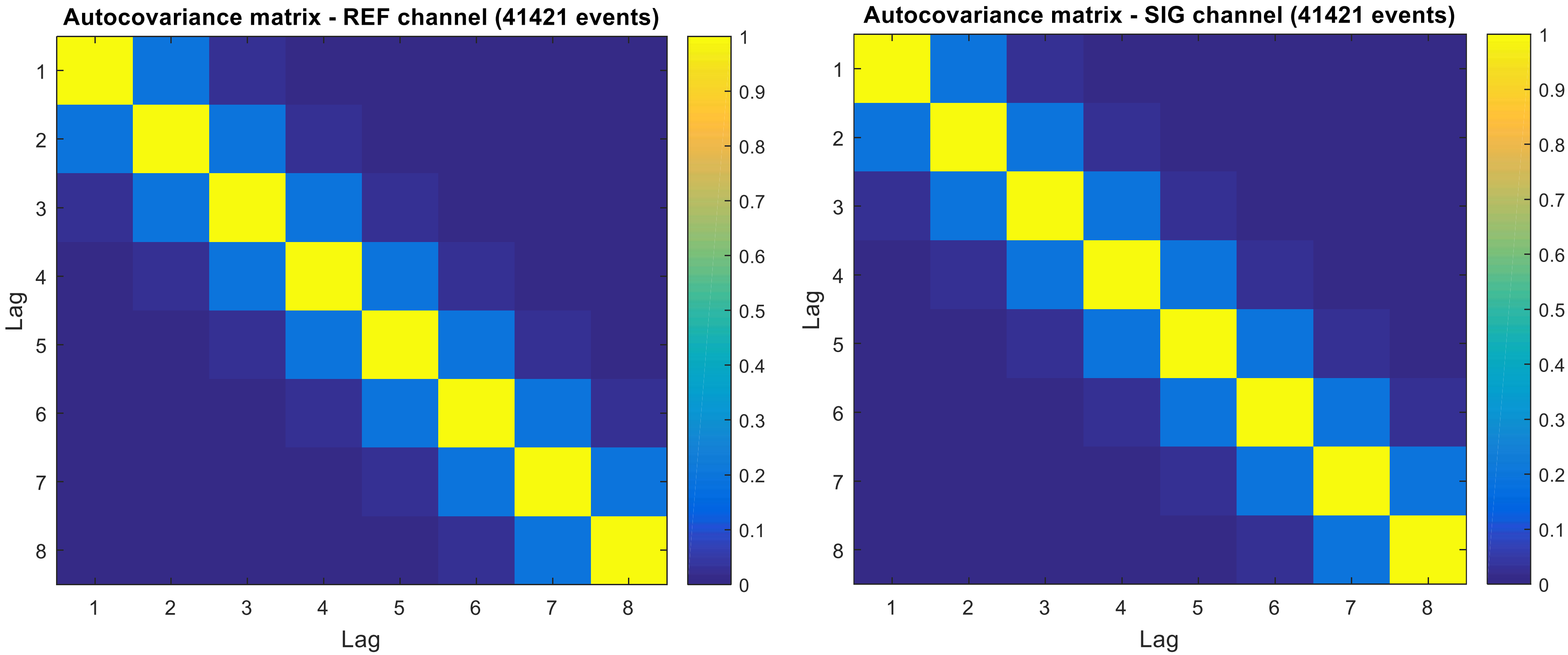
STEP 1: Detect template



- Compute cross-correlation between two events.
- Align pulses using sinc interpolation – resample 2nd event to maximize cross-correlation.
- Average events.
- Take next event and resample it to maximize cross-correlation with the averaged event.
- Repeat last step for desired amount of events.

FIR synthesis

STEP 2: Calculate noise autocovariance matrix

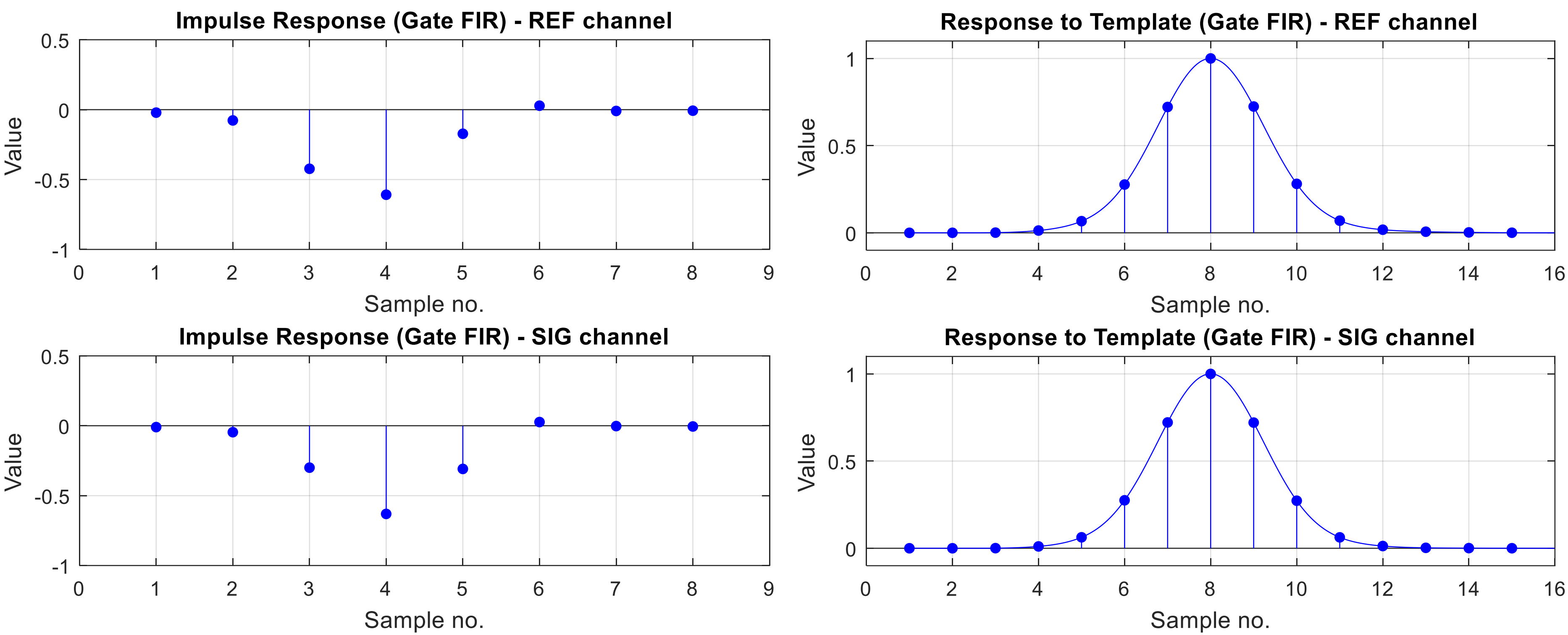


If the images are smeared, then it is PDF's image compression rather than strange covariance matrix.

FIR synthesis

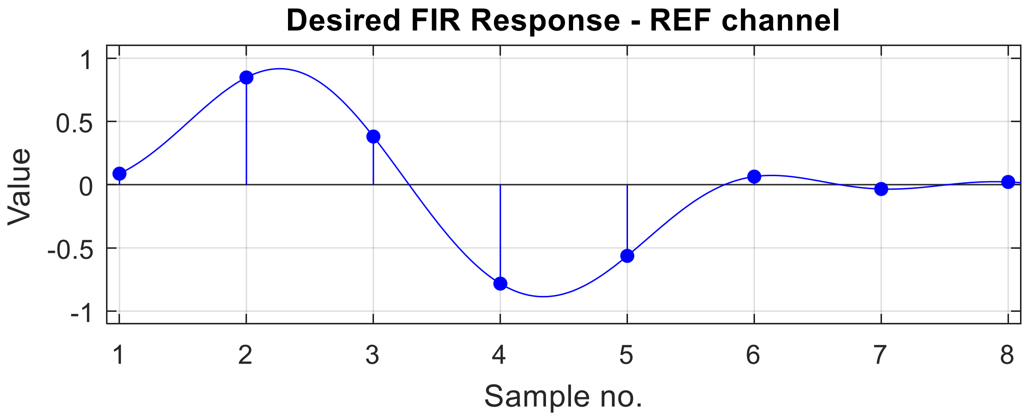
STEP 3: Calculate ‘gate’ filter

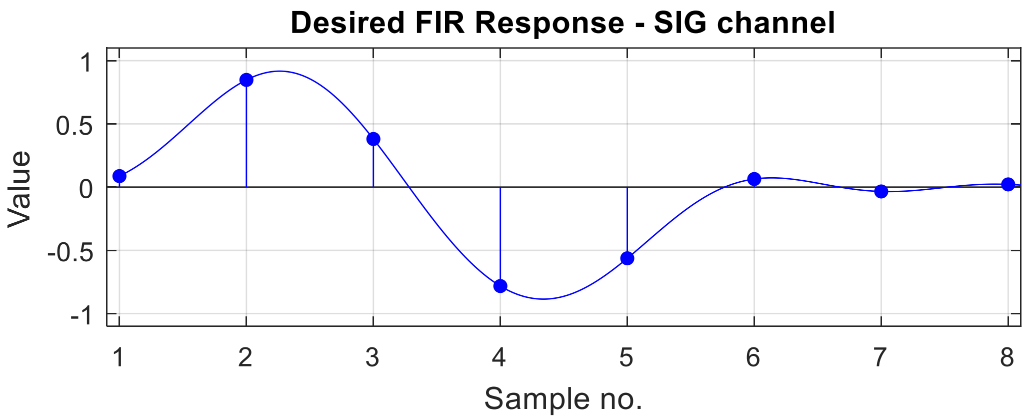
The ‘gate’ filter will be used to detect pulse. It is a standard matched filter that maximizes SNR.



FIR synthesis

STEP 4: Calculate desired FIR response



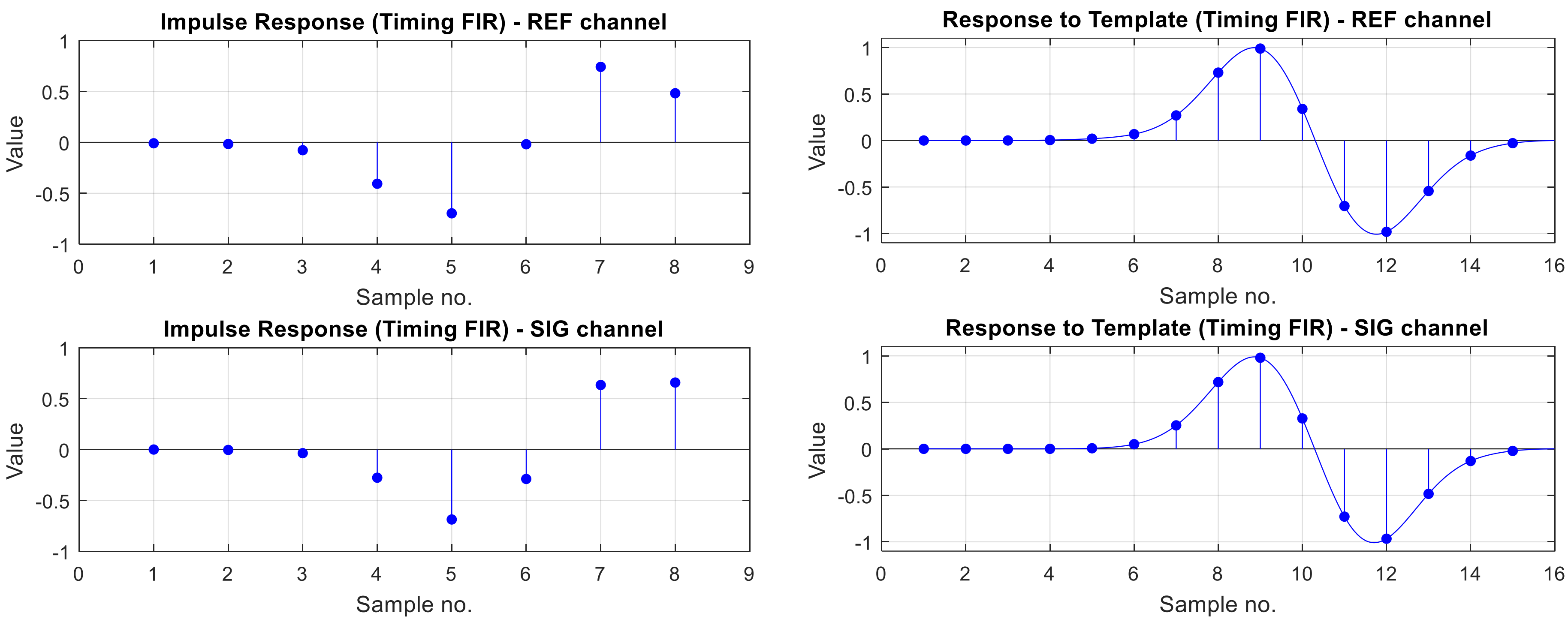


- Use solver and compute waveform shape that meets desired shape, length and linear edge requirements.
- Downsample resulting waveform so that Nyquist criteria is met.
- Figures show downsampled responses.

FIR synthesis

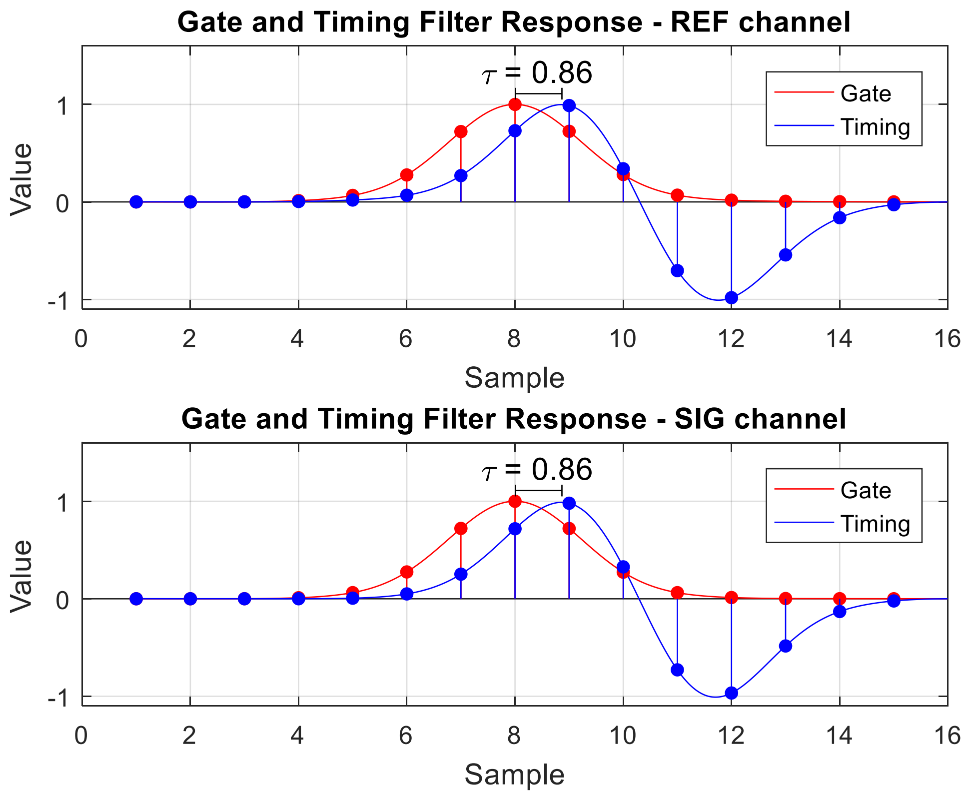
STEP 5: Calculate ‘timing’ FIR

- Use DPLMS method to calculate FIR filter based on pulse template, desired response and noise autocovariance matrix.



FIR synthesis

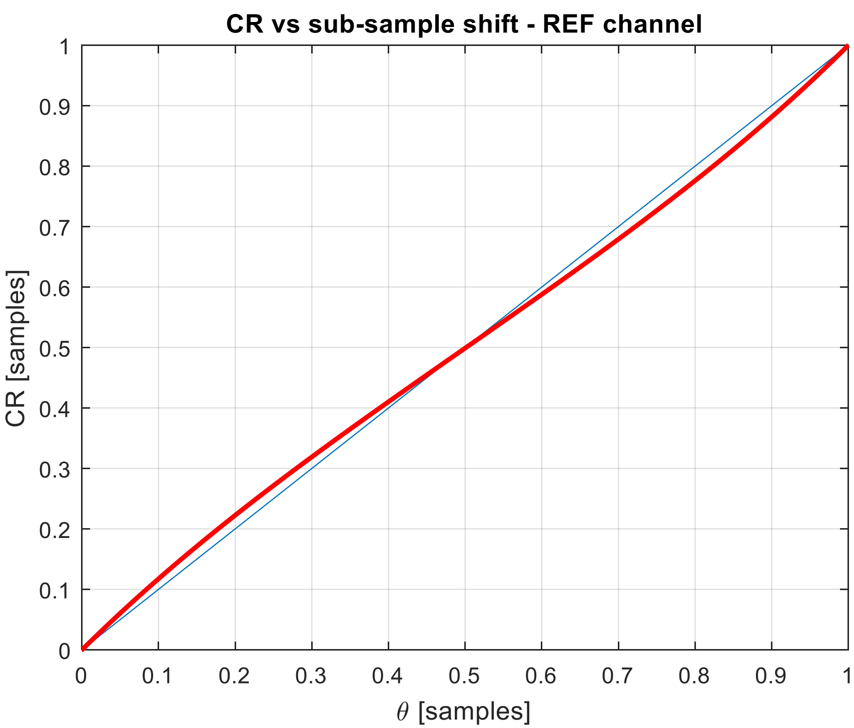
STEP 6: Calculate shift between maximums of 'gate' and 'timing' filter response

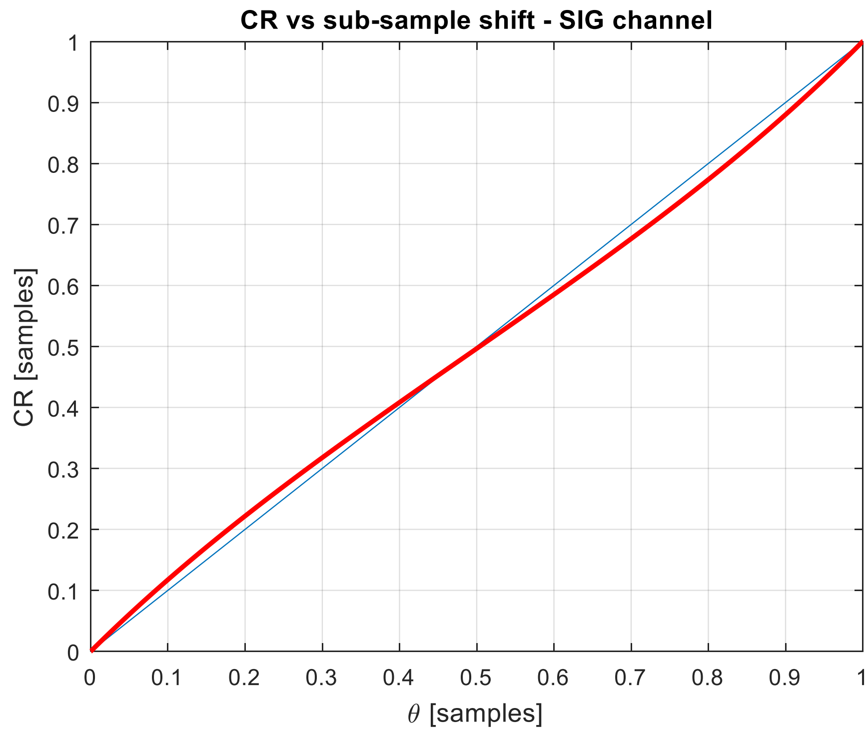


- Make separate calculation for 'reference' and 'signal' channels
- This value will later be used to start searching for zero-crossing of 'timing' filter response.

FIR synthesis

STEP 6: Calculate correction function to account for non-linear shape near zero crossing of ‘timing’ filter response





θ - actual sub-sample shift

$$CR = \frac{P}{P - Q}$$