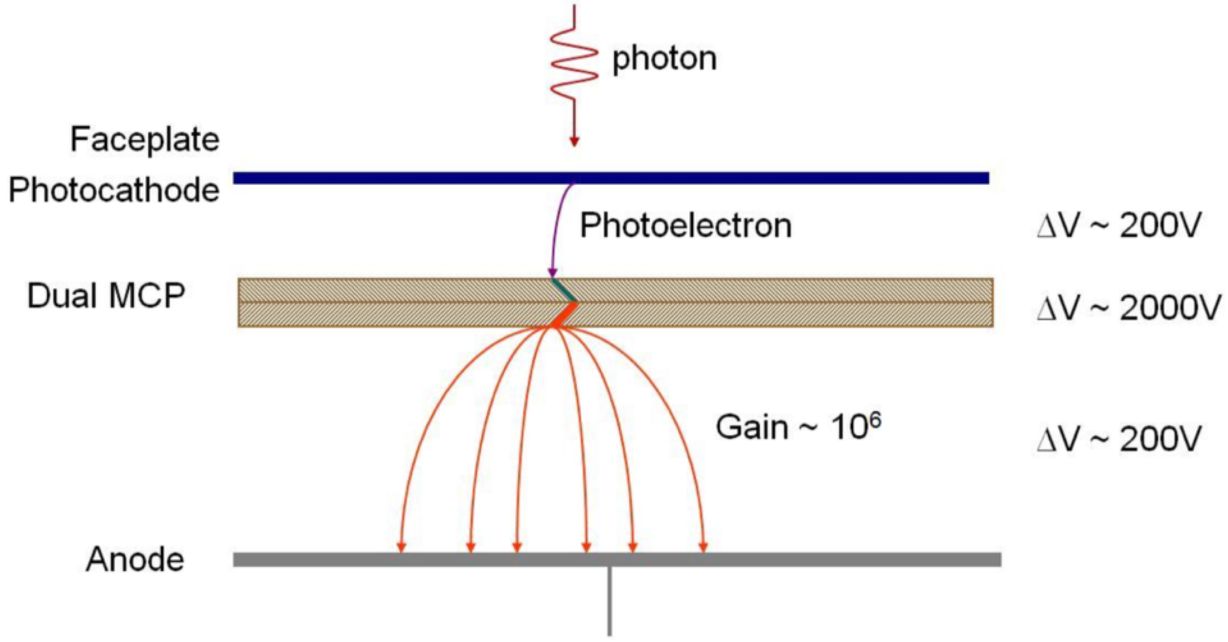
PICOSEC t_0 Reference

Sebastian White, for the PICOSEC Group,
CERN GDD, CEA/Saclay, Princeton, Demokritos, Thessaloniki, Hefei (+LIP)
RD51 mini-week, Dec. 14, 2016

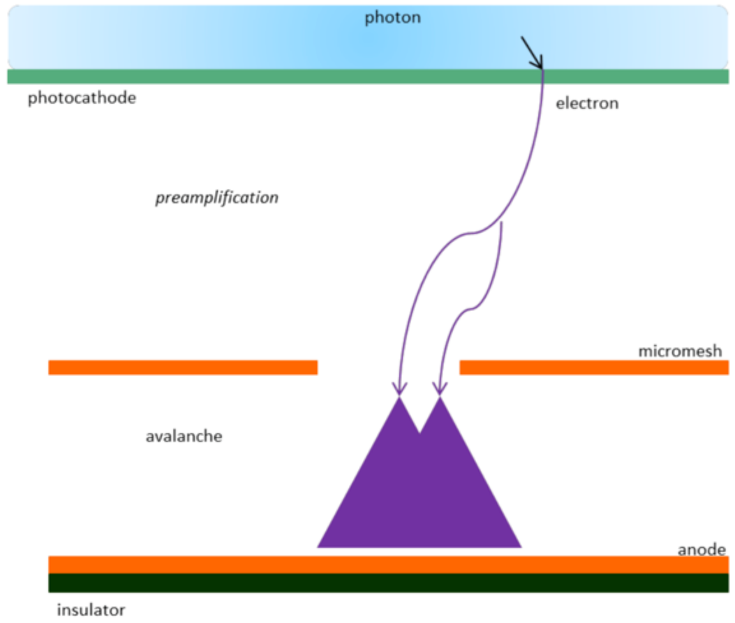
ref.



MCP:

-early record holder
(K. Inami et al. NIM A 560 (2006) p. 303)
-often copied since
-some mysteries relevant here
-what correction for t_0 in PICOSEC data?
-what lessons for signal processing?

DUT

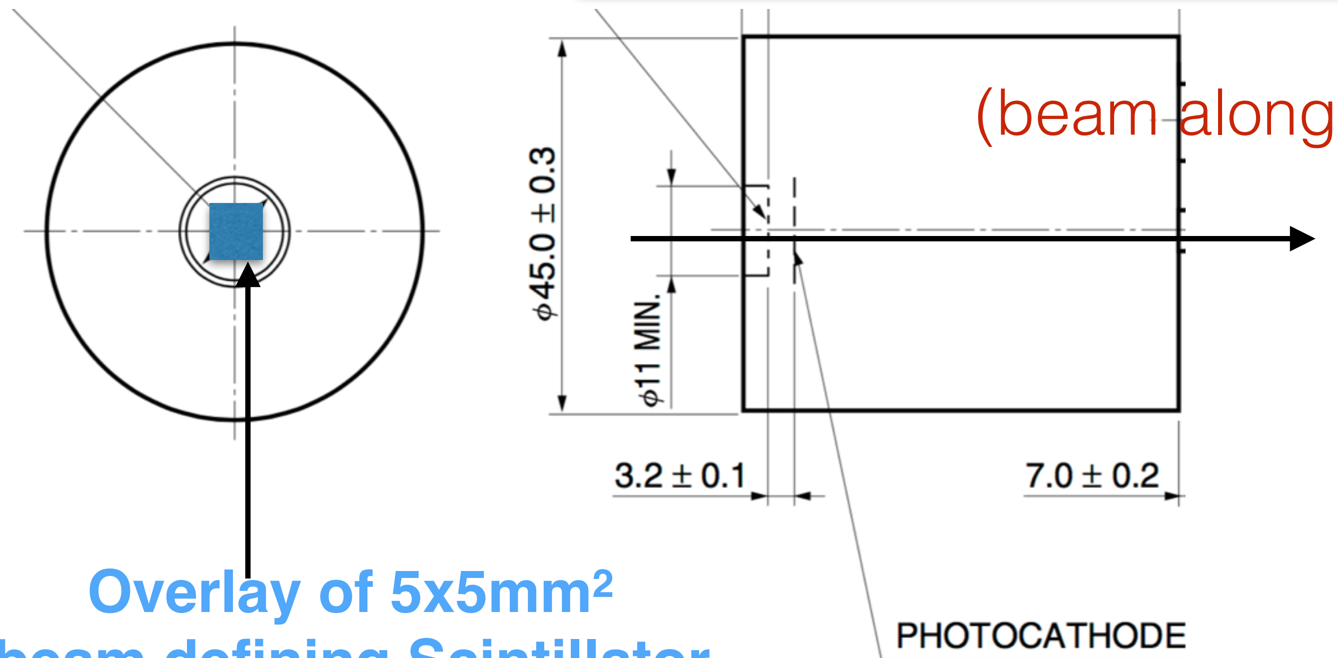


PICOSEC:

-TTS->diffusion term
-same principle of isochronous photoelect.
-similar photo-elect. yield
-in PICOSEC (and HyperFastSilicon)
limitations to CFD techniques
-apply lessons from MCP

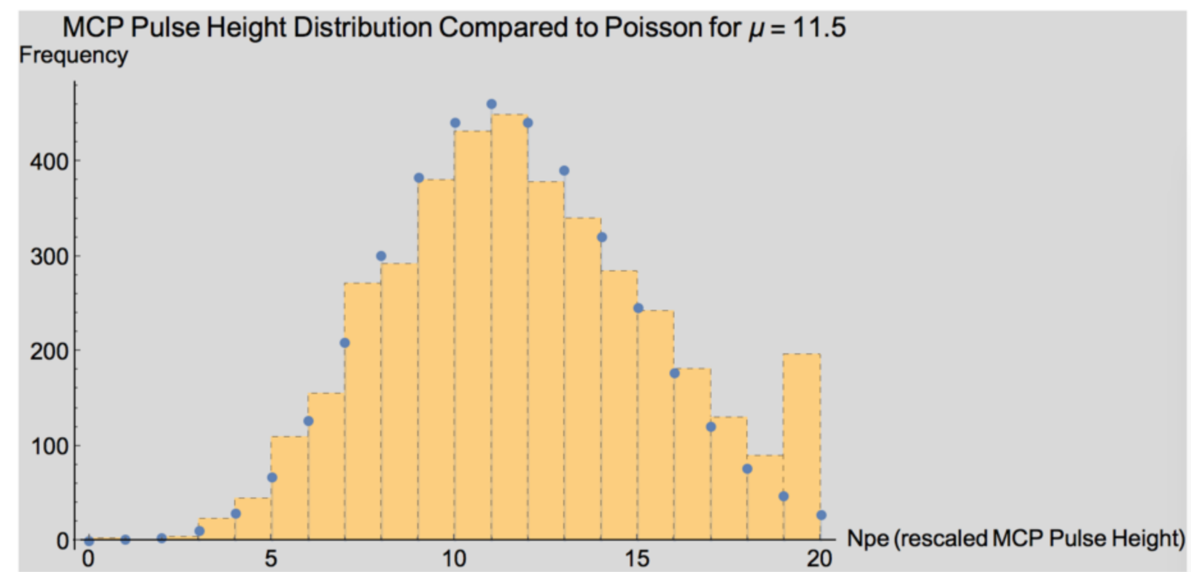
Operating Conditions

Hamamatsu R3809U- 50/52 Bialkali/MultiAlkali (same as Inami et al.)



Overlay of 5x5mm² beam defining Scintillator (trigger for all PICOSEC data) on ~11mm PC Diameter

Npe~11 inferred from our PH Distribution

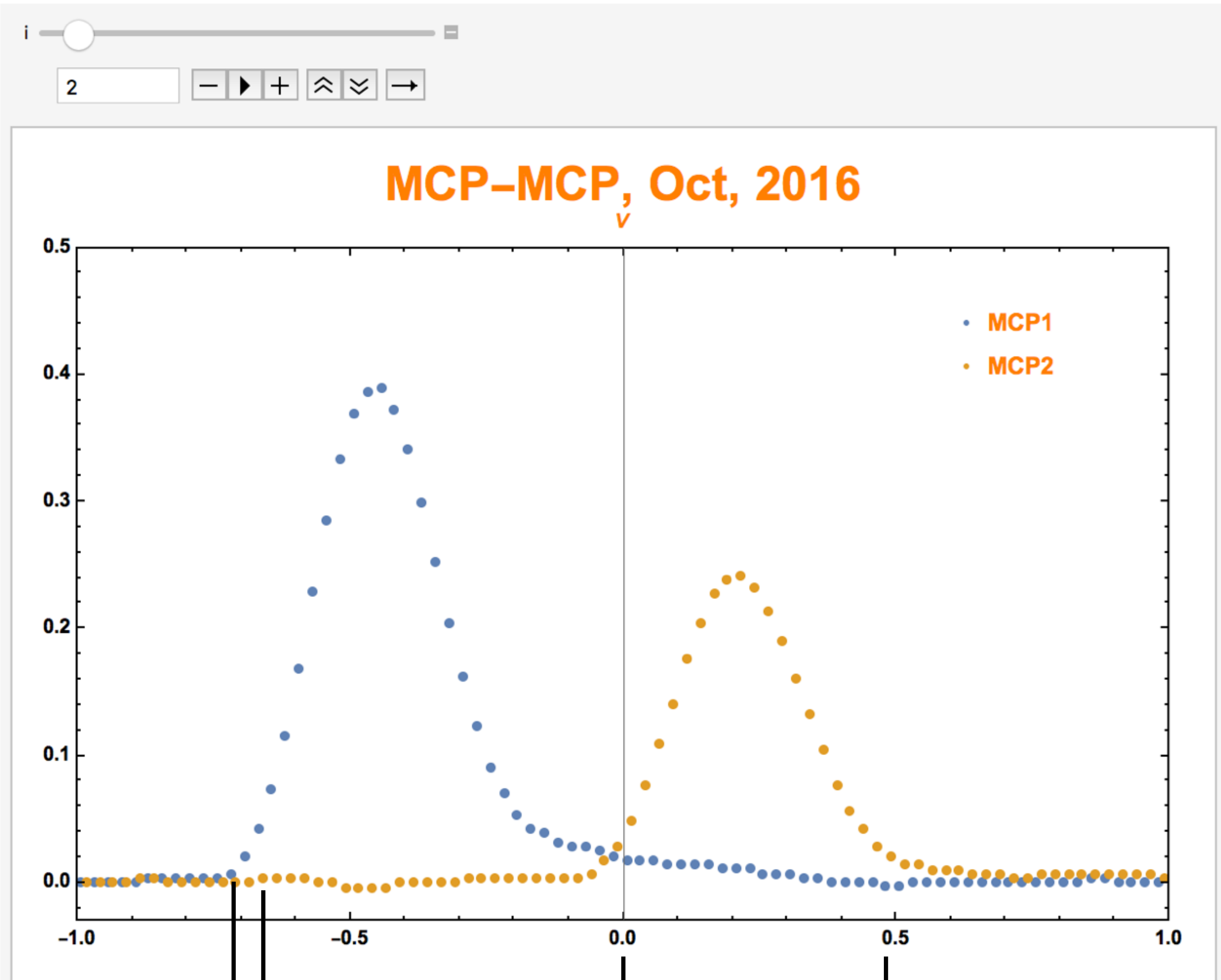


Comparison to Inami et al.

- We use 2.8 kV (nominal G~8x10⁴), they use 3.2-3.6 kV
- we use 3.2mm window as radiator, their optimum at 10mm
- They digitized with Time-Correlated SPC (CFD,TAC), while we used Lecroy 2.5GHz, 8-bit 40GSa/s(20 GSa/s) scope
- we do a bit better

Conditions(cont)

Run 269-Oct. 7,data w. 2-R3809 PMTs to evaluate IRF
(thanks to Stefano Mazzoni for 2nd PMT loan)



40GSa/s

t_R~160 picosec
t_{Base}~500 picosec

CH2 <V_{peak}> =0.4V
CH3 <V_{peak}> =0.23V

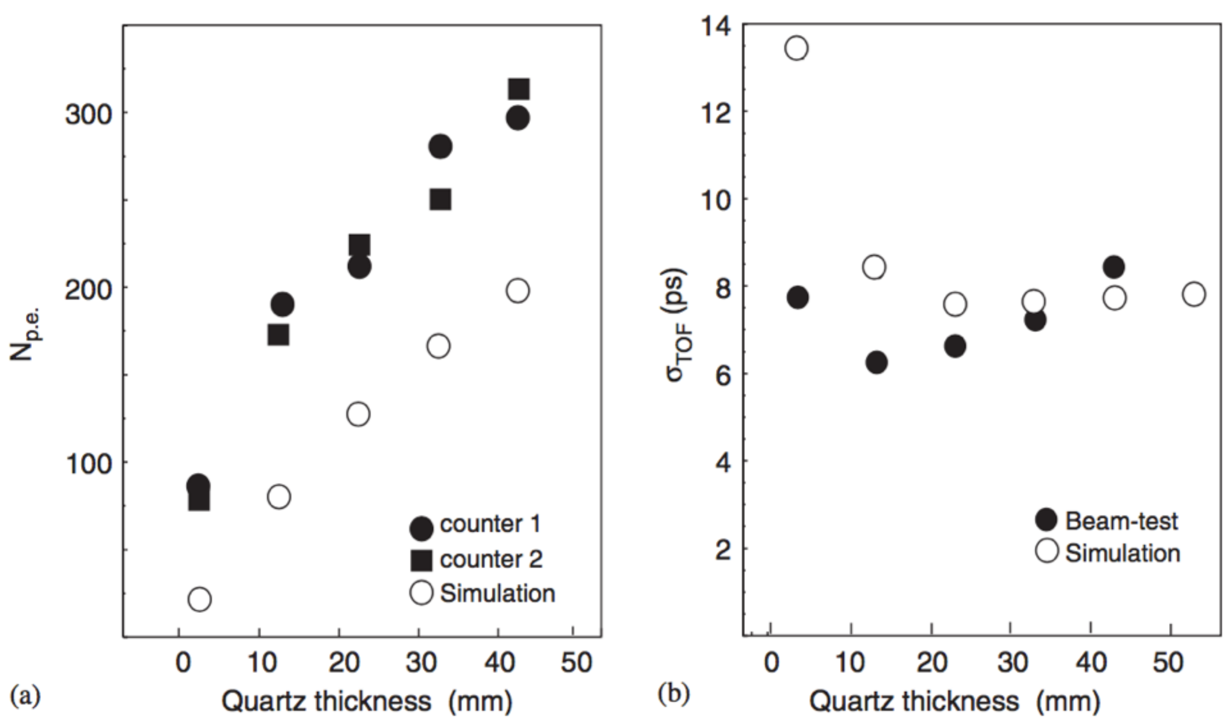
-$N_{pe}1=(V_{peak}/0.4)*11$
-$N_{pe}2=(V_{peak}/0.23)*11$

HF:V_{noise}~1.5 mV (from 4mV least count)
LF: V_{noise}<1 mV (baseline restore)

Q_{out}~0.5-1x10^6*N_{pe}

(aside on MCP Gain)

- Gain estimate on previous slide~ 10* higher than data sheet
- related to data/simulation discrepancy of Inami et al.?
- not really pertinent for us but related to pulse vs. DC gain?



Inami et al.

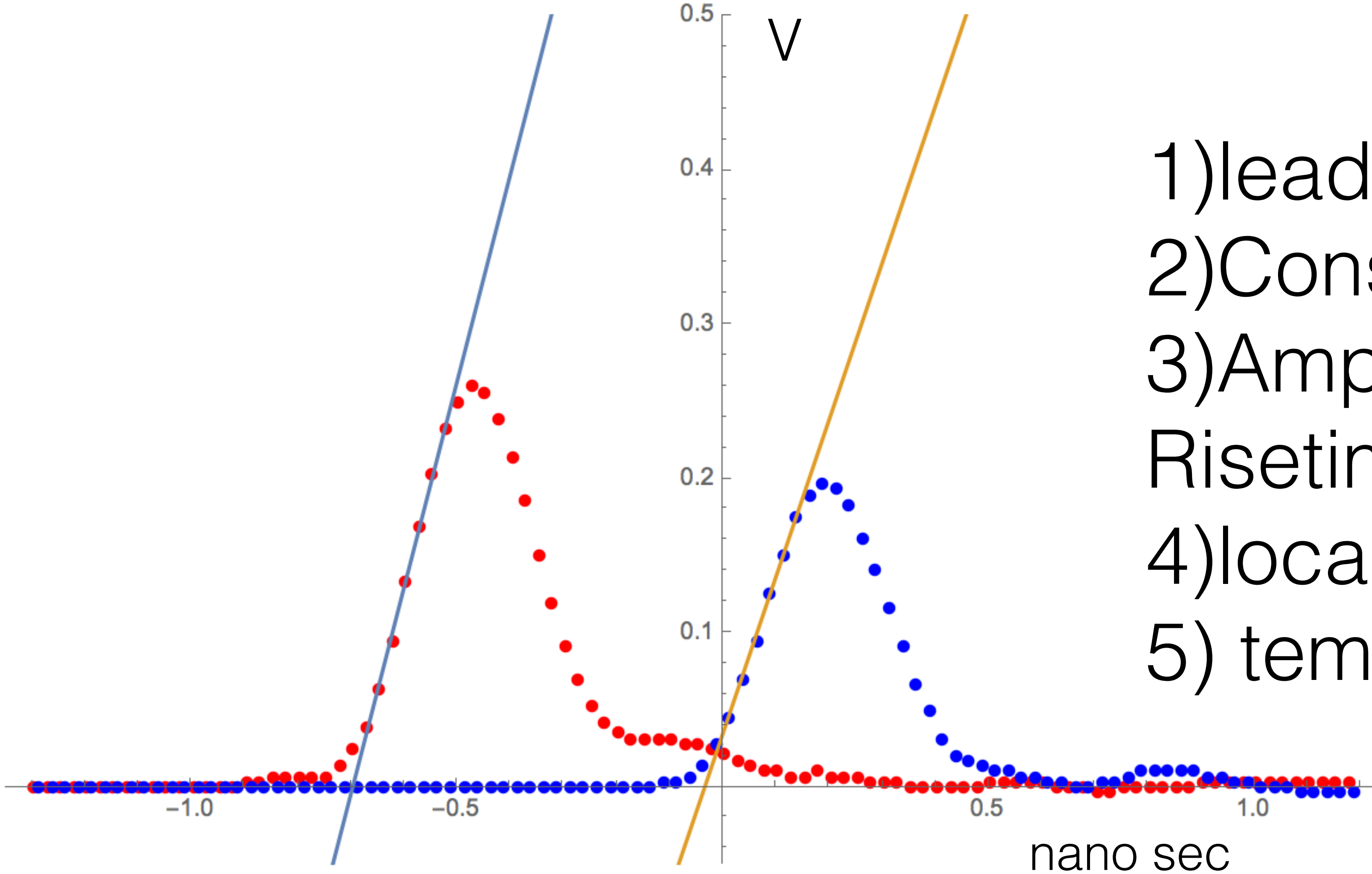
(note their best result

$\sigma_{TOF} = 6.2\,\mathrm{ps} \longrightarrow$ Sqrt[2]*6.2

for MCP1-MCP2 when comparing to our results below)

timing Algorithms

ch2 risetime=0.116459 channel 3 risetime= 0.115825 (20-80%)



Options:

1)leading edge (LED)
2)Constant Fraction (CFD)
3)Amplitude and Risetime Compensated(ARC)
4)local Constant Fraction (lCF)
5) template-see next (Spyros)

1)->optimum at low threshold-> fit reduces noise
2)->CFD good for MCP because t_{Rise} stable
3) mostly used for slow signals i.e. large Ge detectors
4) interesting for PICOSEC and HFS, evaluate here.

time jitter contributions

- **$dt = TTS/Sqrt[N_{pe}] + t_{Rise}/SNR$ + Slew term + scope**

1 2 3 4

1) useful to identify amplitude dependence->non-Gaussian

2) negligible (ie~ 160picosec/200)

3) minimize with choice of algorithm

4) small (but beware of i.e. Lecroy scope event-dependent channel offsets:

Hello Yi and Sebastian,

All input channels in the scope has it's own signal path. Even if we tried to match the each input signal path length to ADC equal, but it's not possible to make it perfect.
So the actual time of the first sample to the trigger point for each input channels will be different.

I believe that matlab code is converting each channel trace file, so it will not correlate other channels. It will just convert the waveform data to ASCII.
If you have to align with other channels, then you have to read out each channels horiz_offset descriptor header information to align.

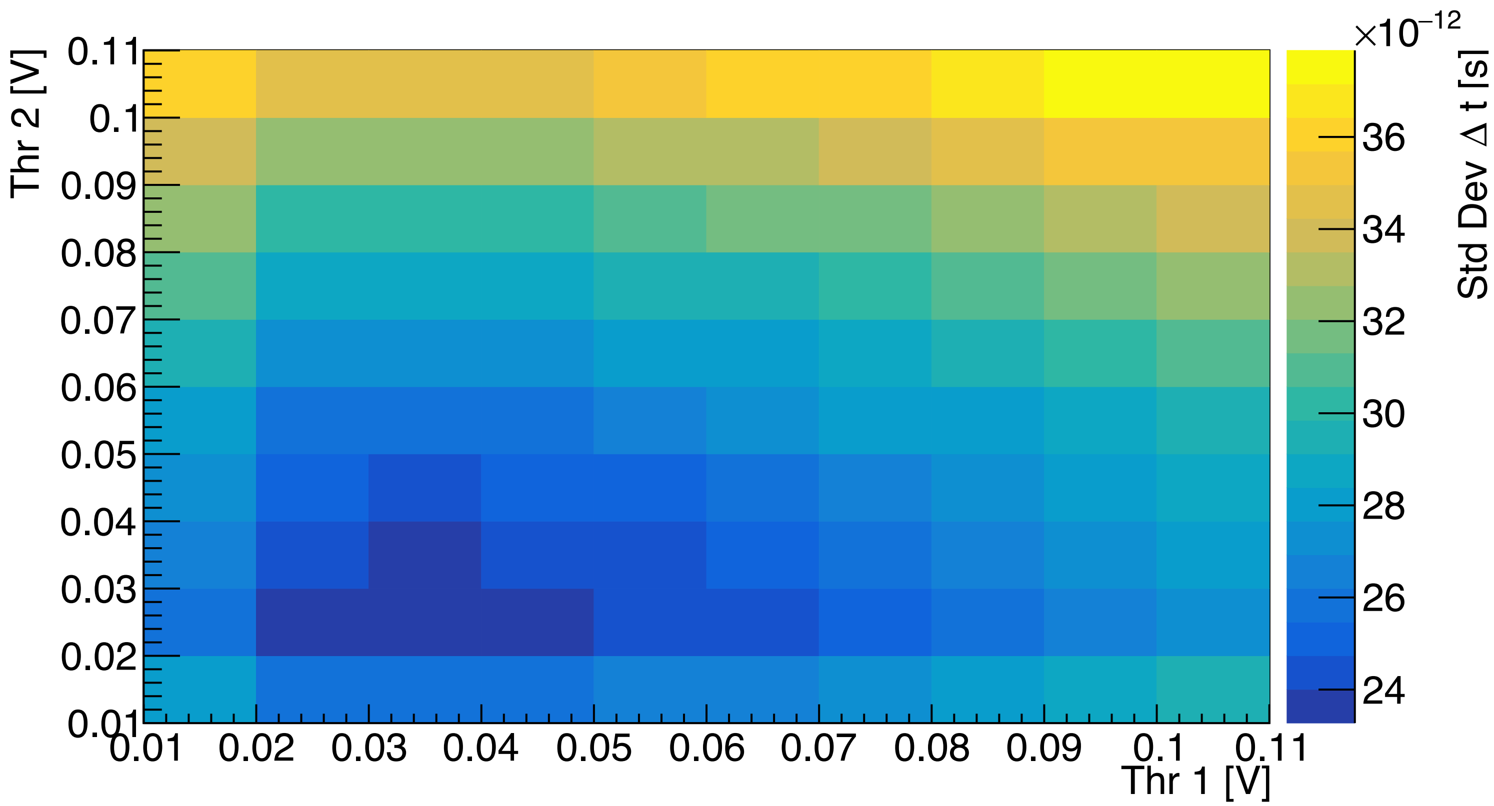
Thanks and Best Regards,

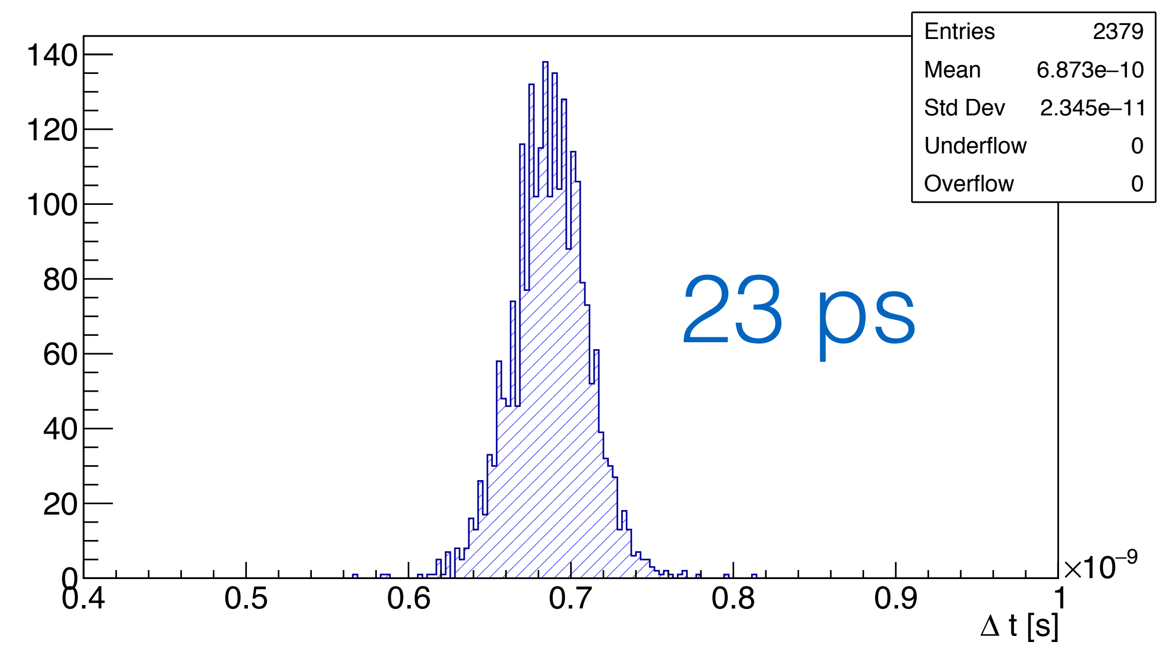
Honam Kwak
Application Engineer

try LED

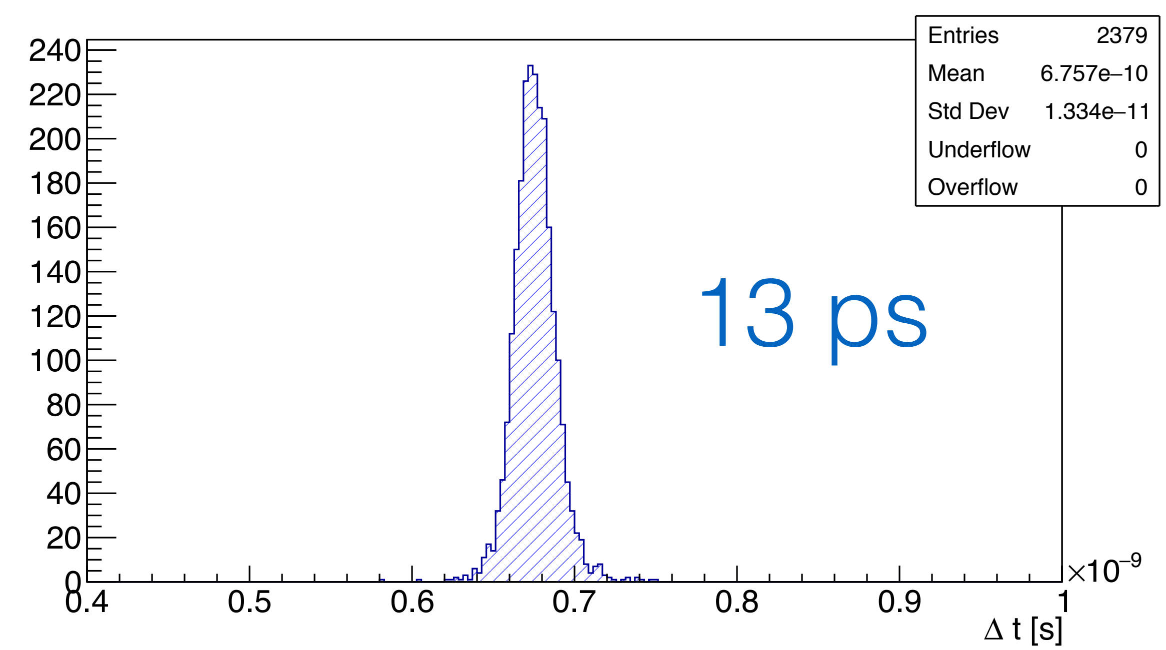
(LED plots below from M. Vignali, CERN SSD group)

in rest of slides always reporting rms on time difference





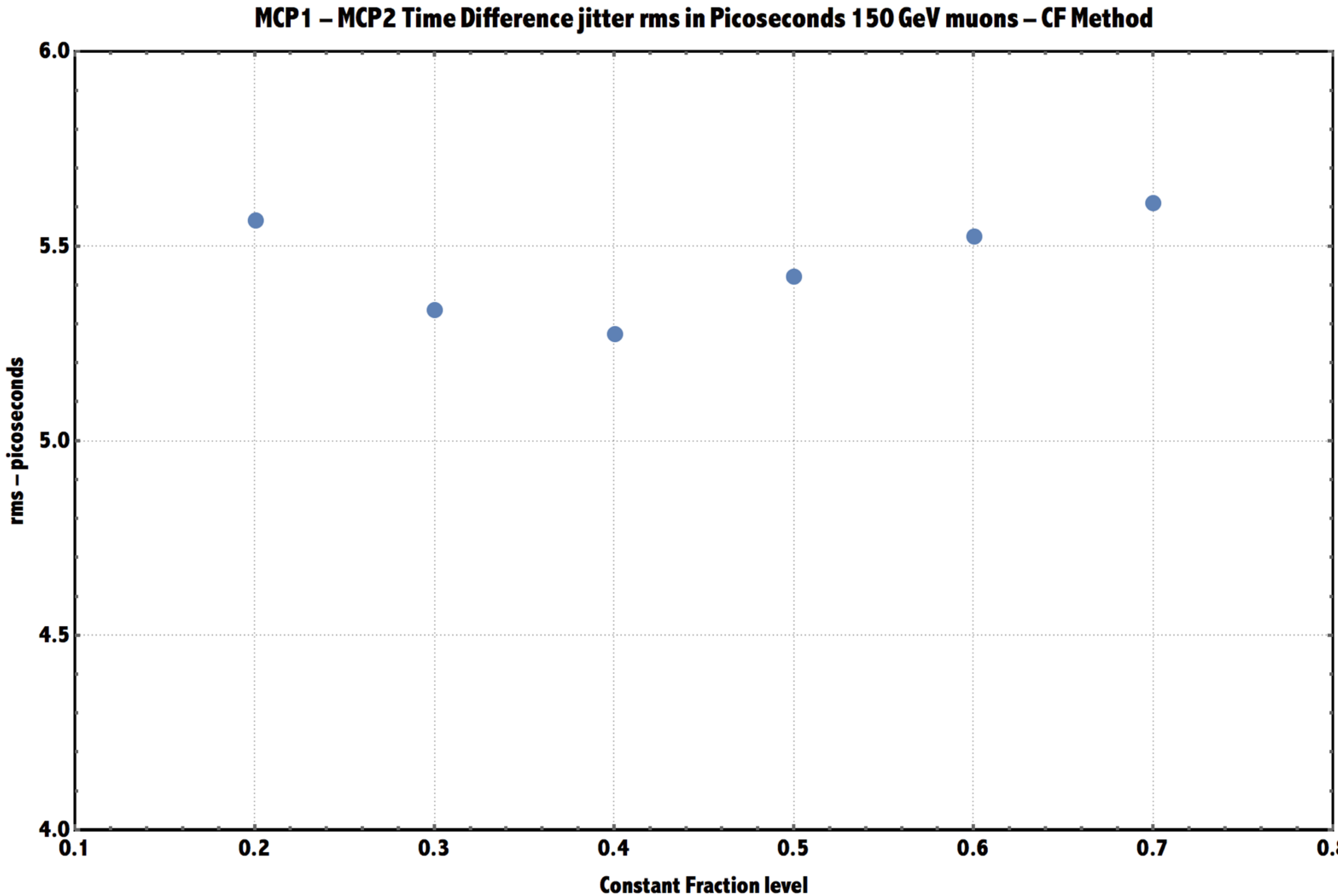
apply threshold on interpolation (upper) or local fitted (lower)



find smallest jitter when V_{thr}~30 mV since close to noise-> some benefit from using fit rather than interp.

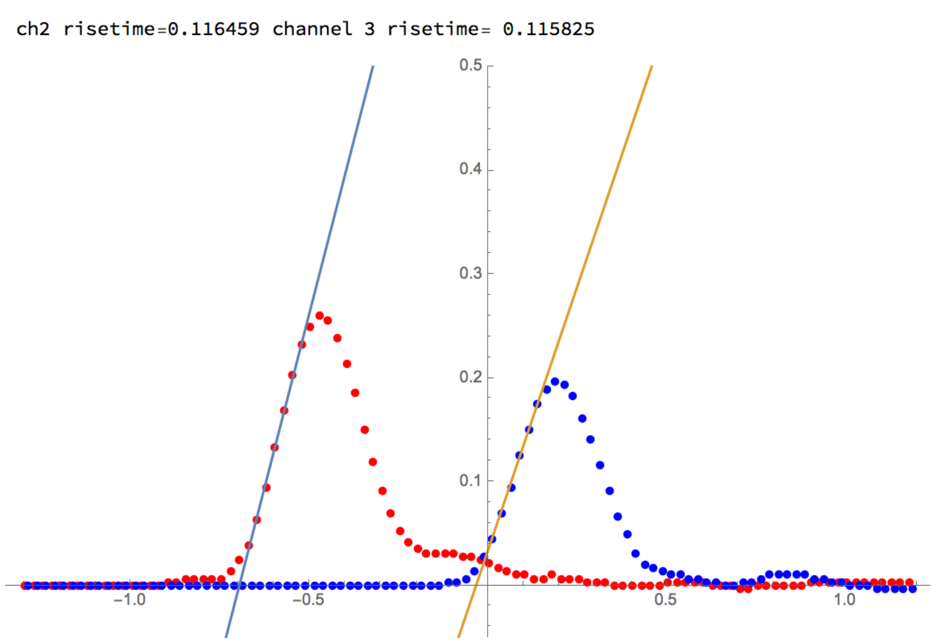
Constant Fraction

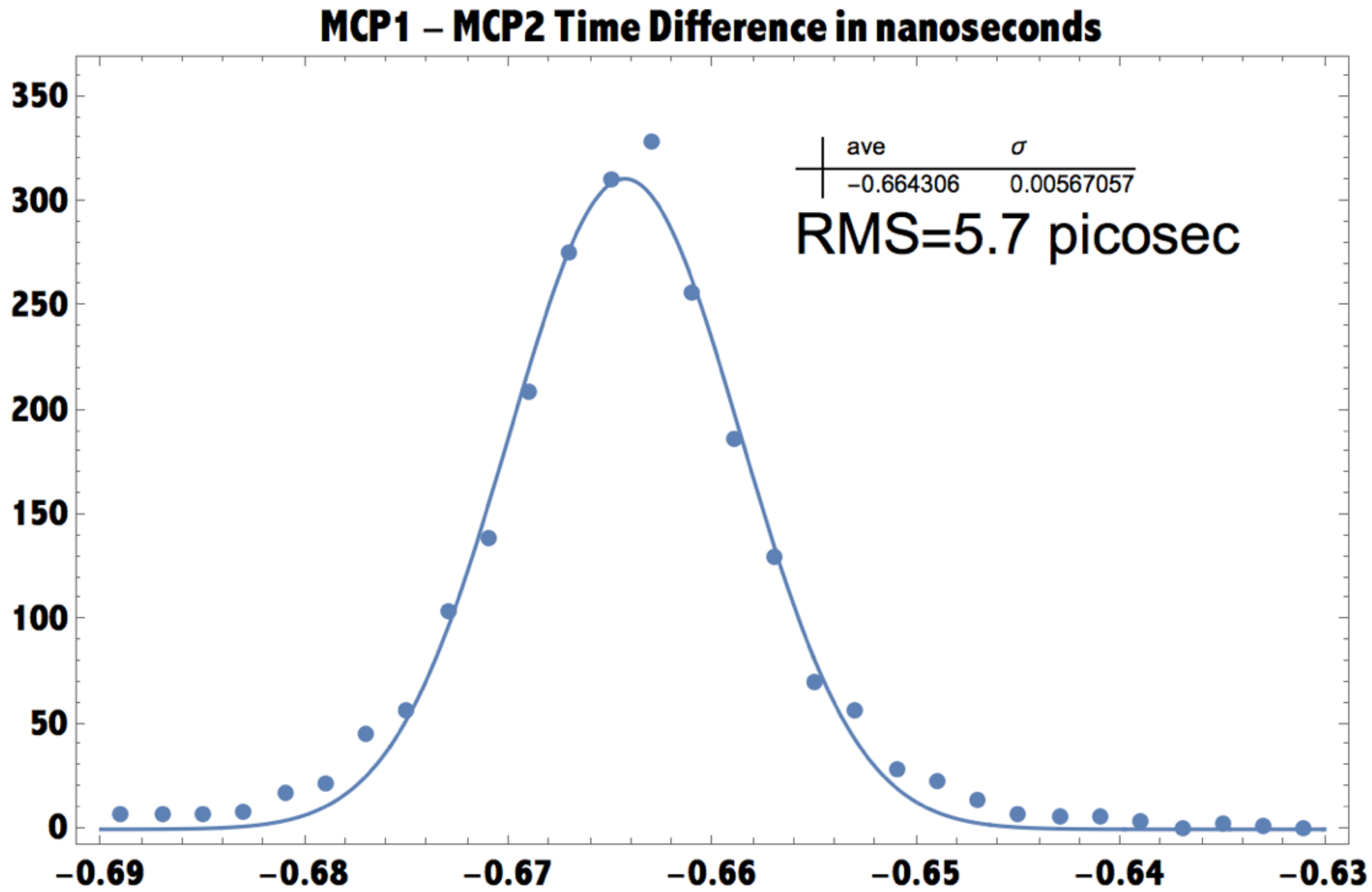
- in below plot I discriminate with the same value (fraction) on MCP1 and MCP2 and report fitted sigma



local Constant Fraction

- this technique could be useful when information is lost by using peak pulse amplitude (ie when ion and electron peaks not well resolved in PICOSEC)
- I use a simple linear fit of points in 20-80% range-> extrapolate to baseline





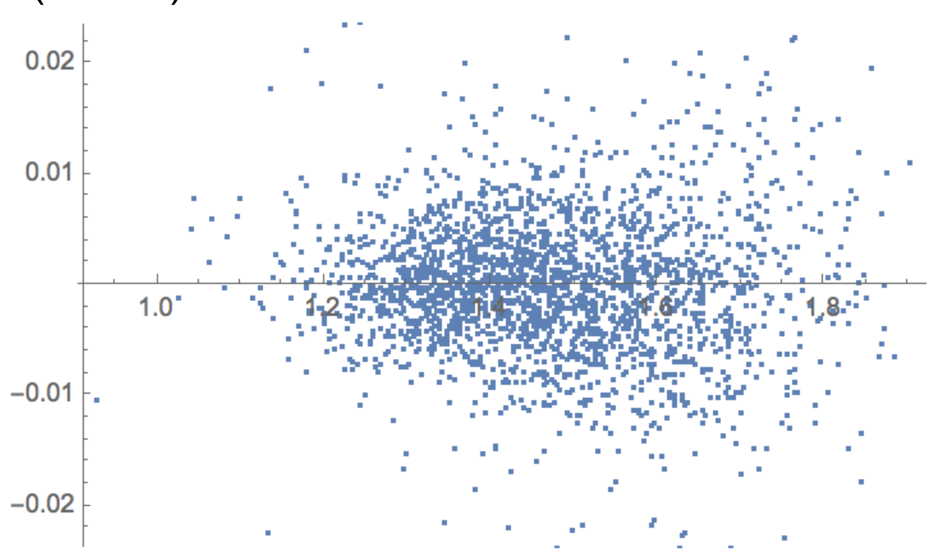
-> similar performance but may be more robust for PICOSEC and HFS

Gaussian fit improvement

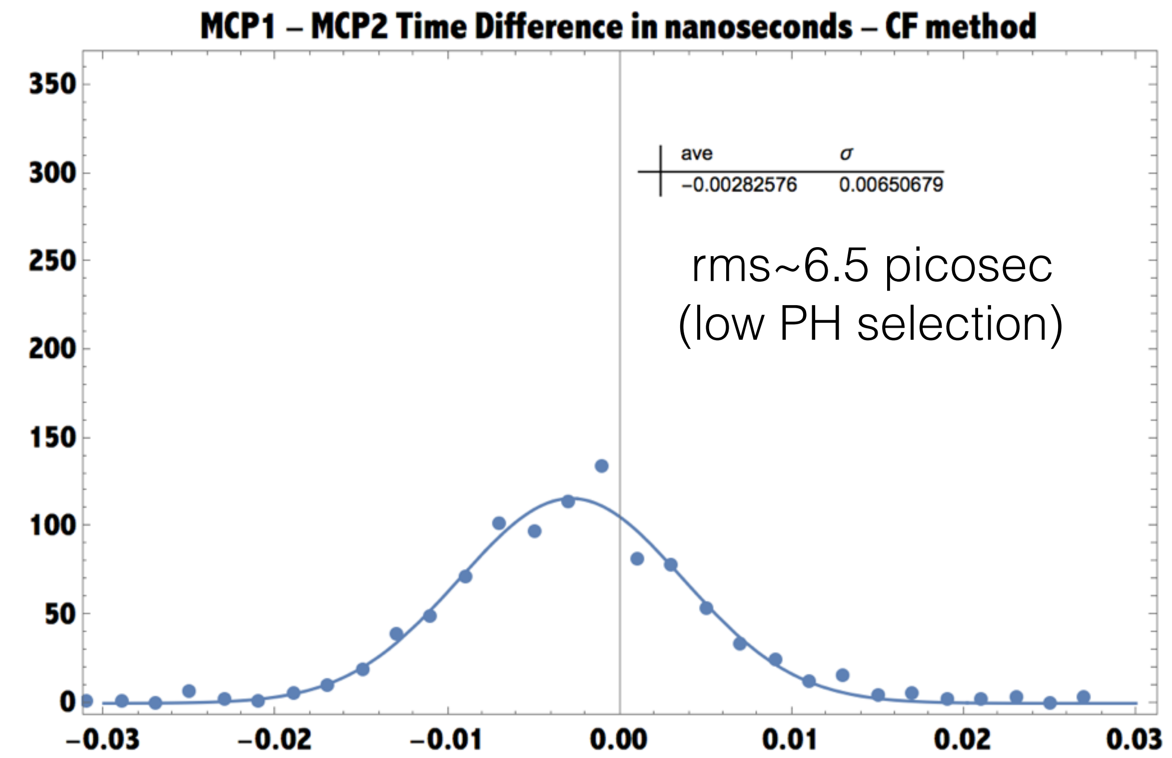
- dt(MCP1-MCP2)~Sqrt[1/N_{pe}1+ 1/N_{pe}2]

can we see this dependence in the data? yes.

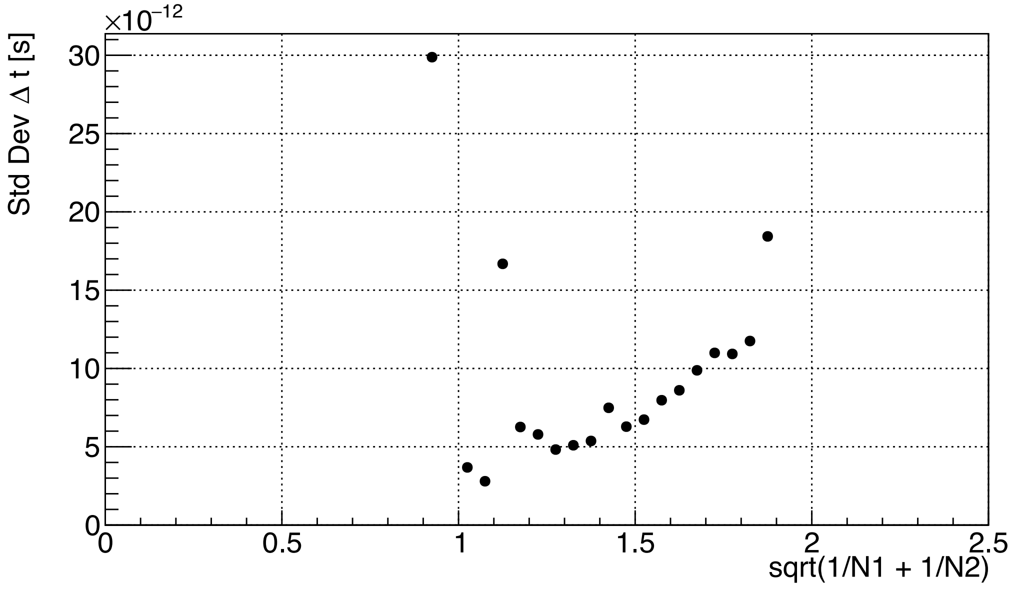
dt(nsec)



rhs



rms~6.5 picosec
(low PH selection)



by choosing lowest half of pulse height range
we get “bad” time jitter of 6.5 picosec
but good fit to Gaussian
upper half gives better jitter w. good Gaussian fit