Time Resolution of LASPD from Cosmic Test

Danning Di, Ye Tian (SDU), Jixie Zhang

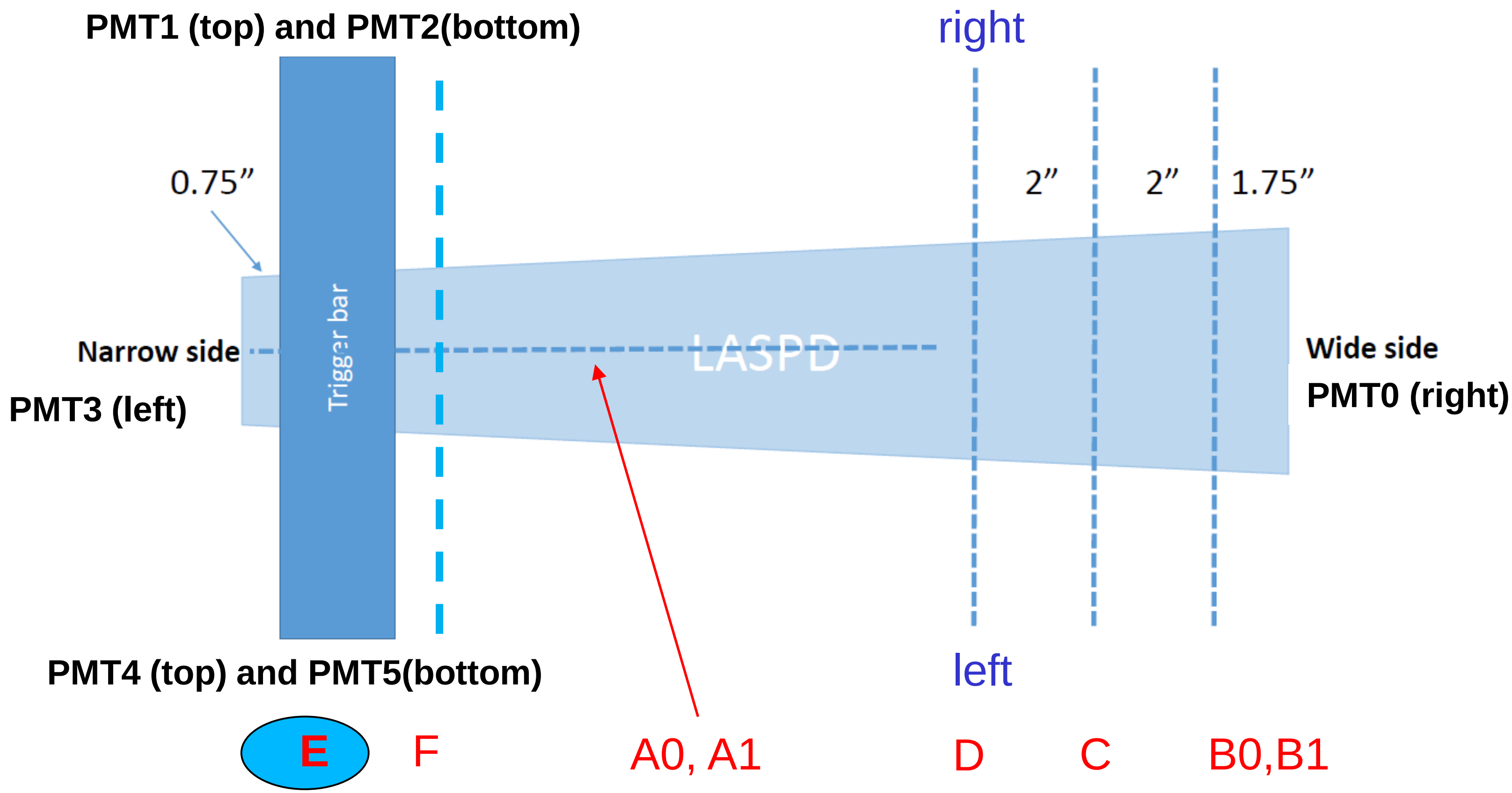
Nov. 30, 2017

Outline

- Previous result.
  - 5^{th} order polynomial fitting to time-walk calibration is over correcting, should fit no more than 2^{nd} order
  - 5^{th} order polynomial calibration result is worse

- What is new?
  - Shift X and Y new LASPD local coordinate, where x=0 is aligned to the narrow edge, y=0 is aligned to the middle point of marrow edge.
  - Try 2^{nd} order polynomial fitting to time-walk calibration
  - New method: X-Y-ADC 3D chi square minimization
- New result.
- Summary and outlook.

Experiment Setup



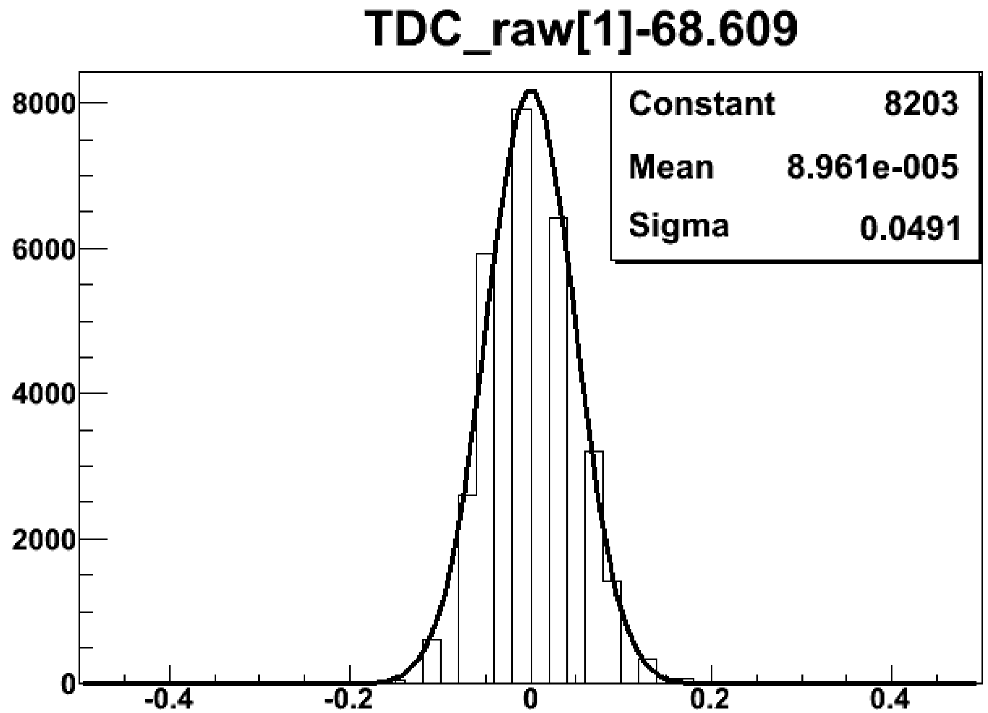
Low HV: A0, B0, C, D
High HV: A1, B1, E, F

Data set F uses only 2 GEM detectors.
All the others use 3 GEM detectors

What is New?

- Check the instinct resolution of trigger time
- Shift X and Y new LASPD local coordinate, where x=0 is aligned to the narrow edge, y=0 is aligned to the middle point of marrow edge.
- Try 2nd order polynomial fitting to time-walk calibration
- New method: X-Y-ADC 3D chi square minimization

Trigger Time Resolution

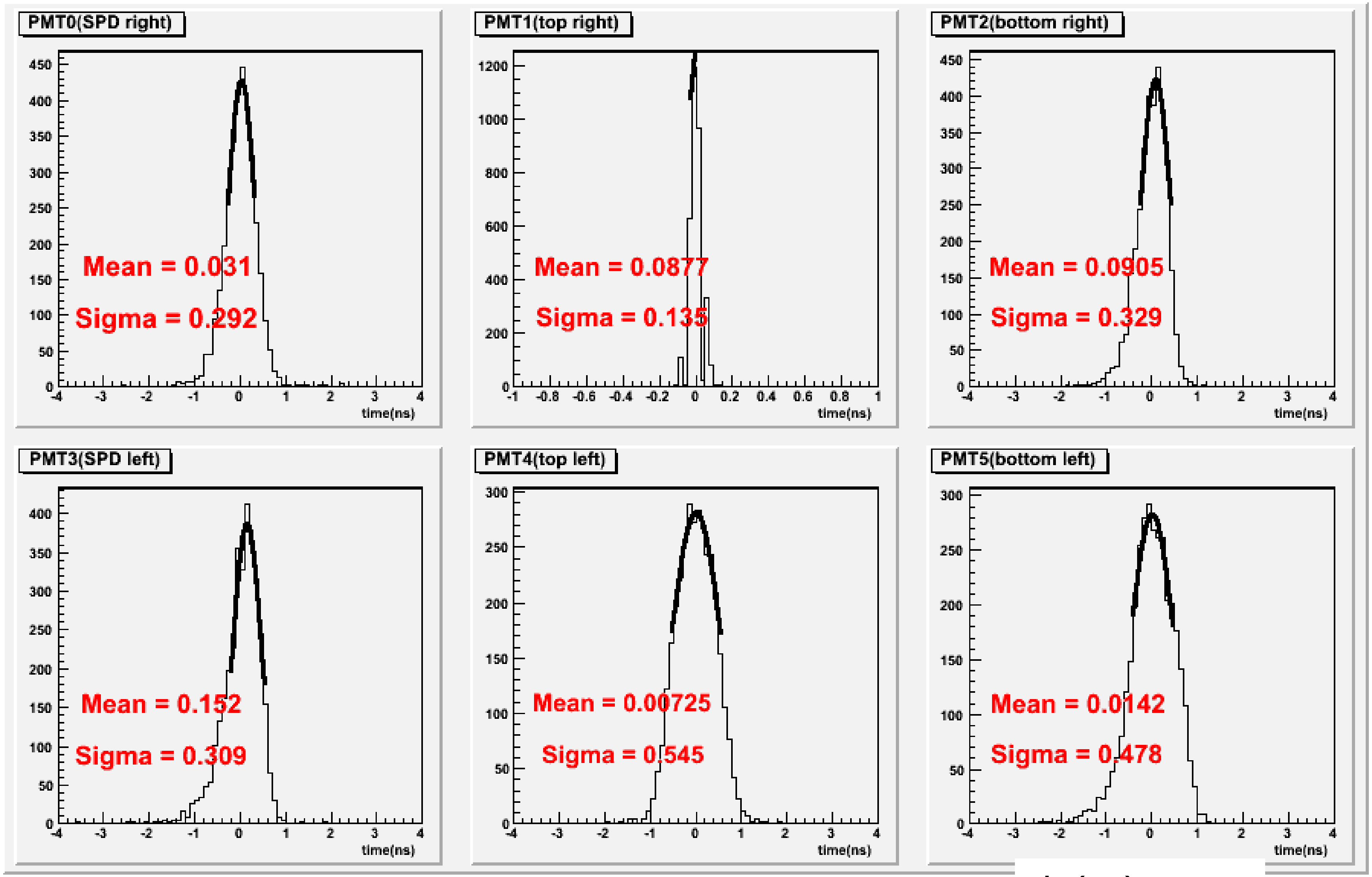


dt (ns)

dt = Time_PMT1 – TriggerTime

- Trigger time is taken from PMT1 and be delayed for ~50ns

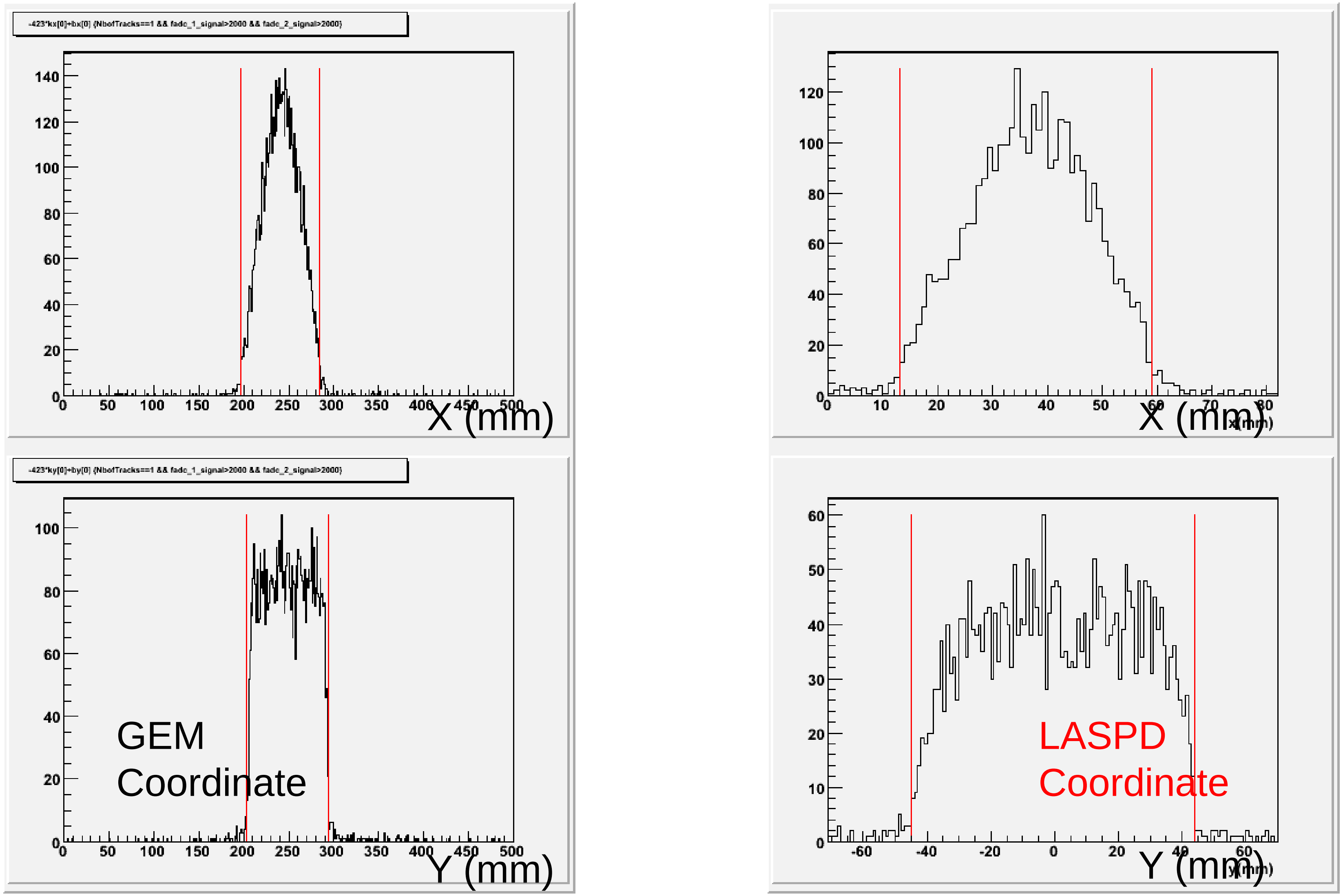
TDC distribution, Calibrated



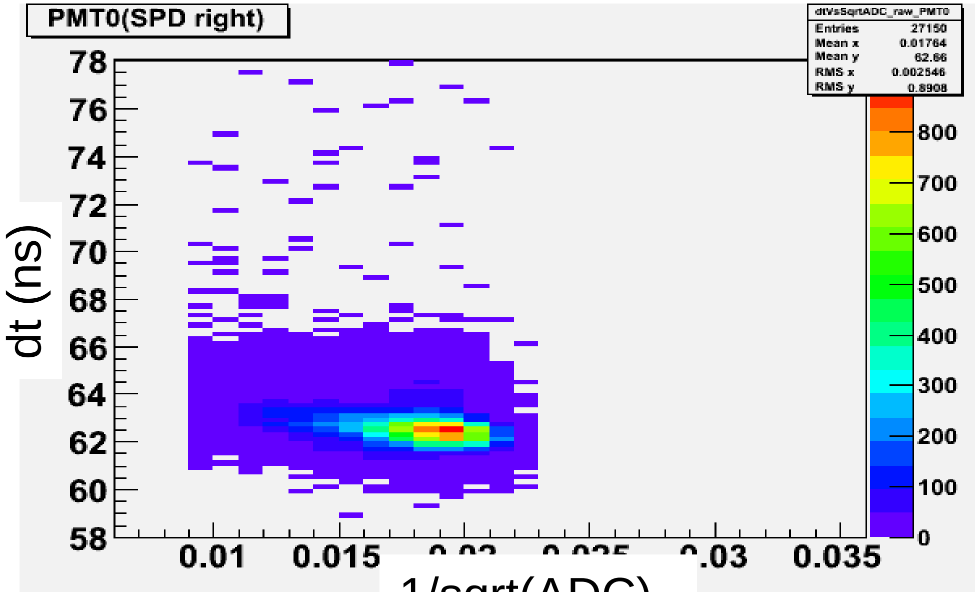
dt (ns)

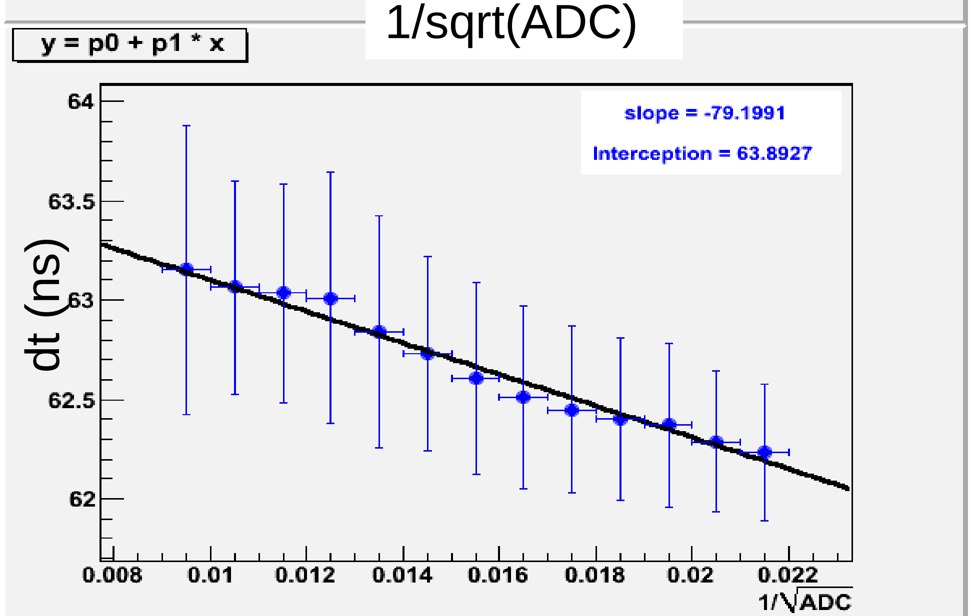
dt = Time_PMT# – TriggerTime

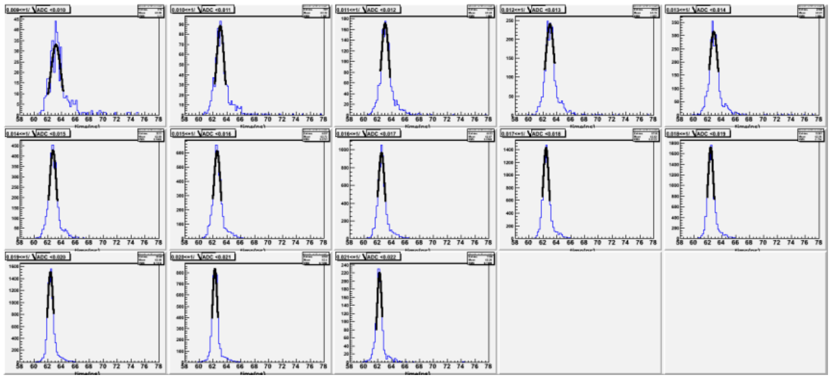
Shift Coordinate



Previous Time-Walk Calibration



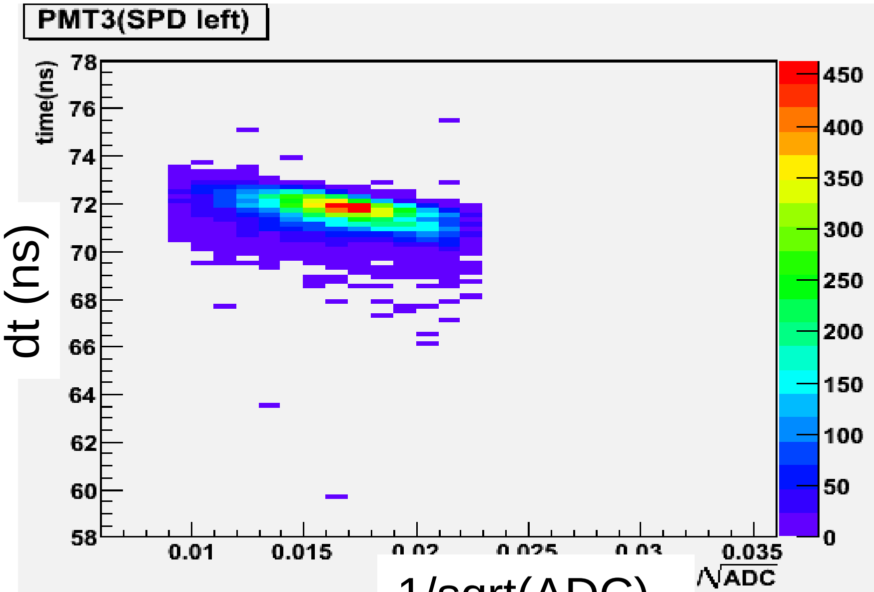


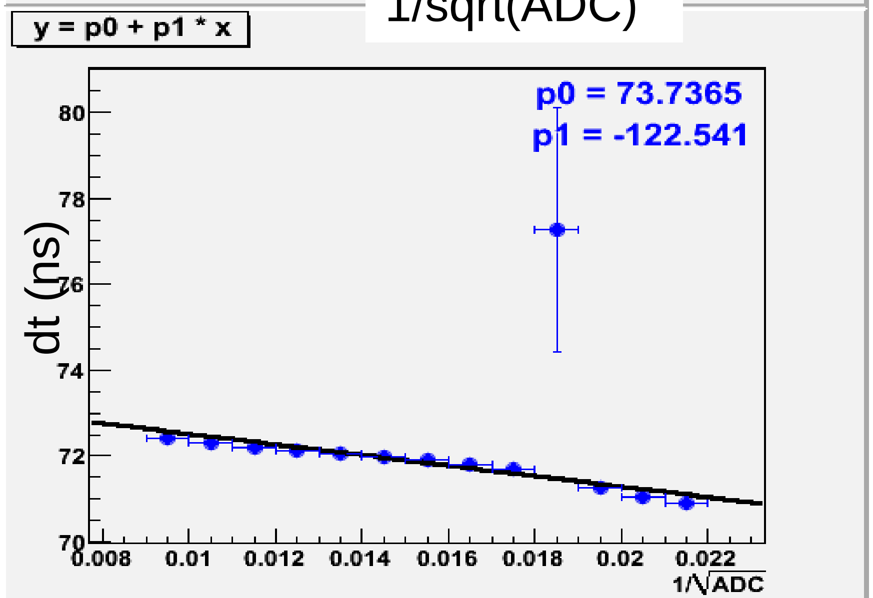


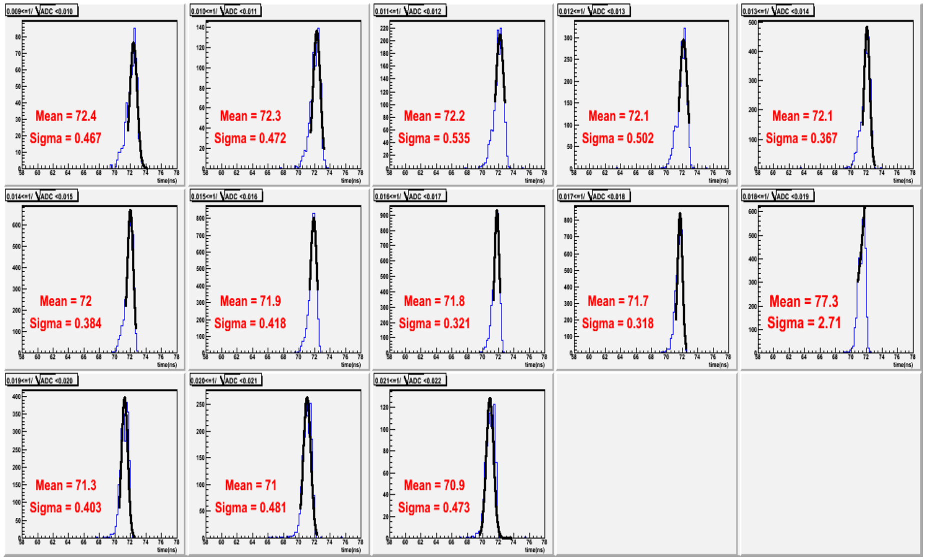
dt = Time_PMT – TriggerTime

- For PMT 0, 2, 3, 5
- Apply trigger cut and ADC cut
- Fit each vertical slices to get the mean and sigma, then fit “mean Vs 1/sqrt(ADC)” by 1st order polynomial, using the sigma as error bar of each mean value.
- It should have x-y dependence, but so far not able to fit it in x-y-ADC 3-D grid yet.

New Time-Walk Calibration, Step 1



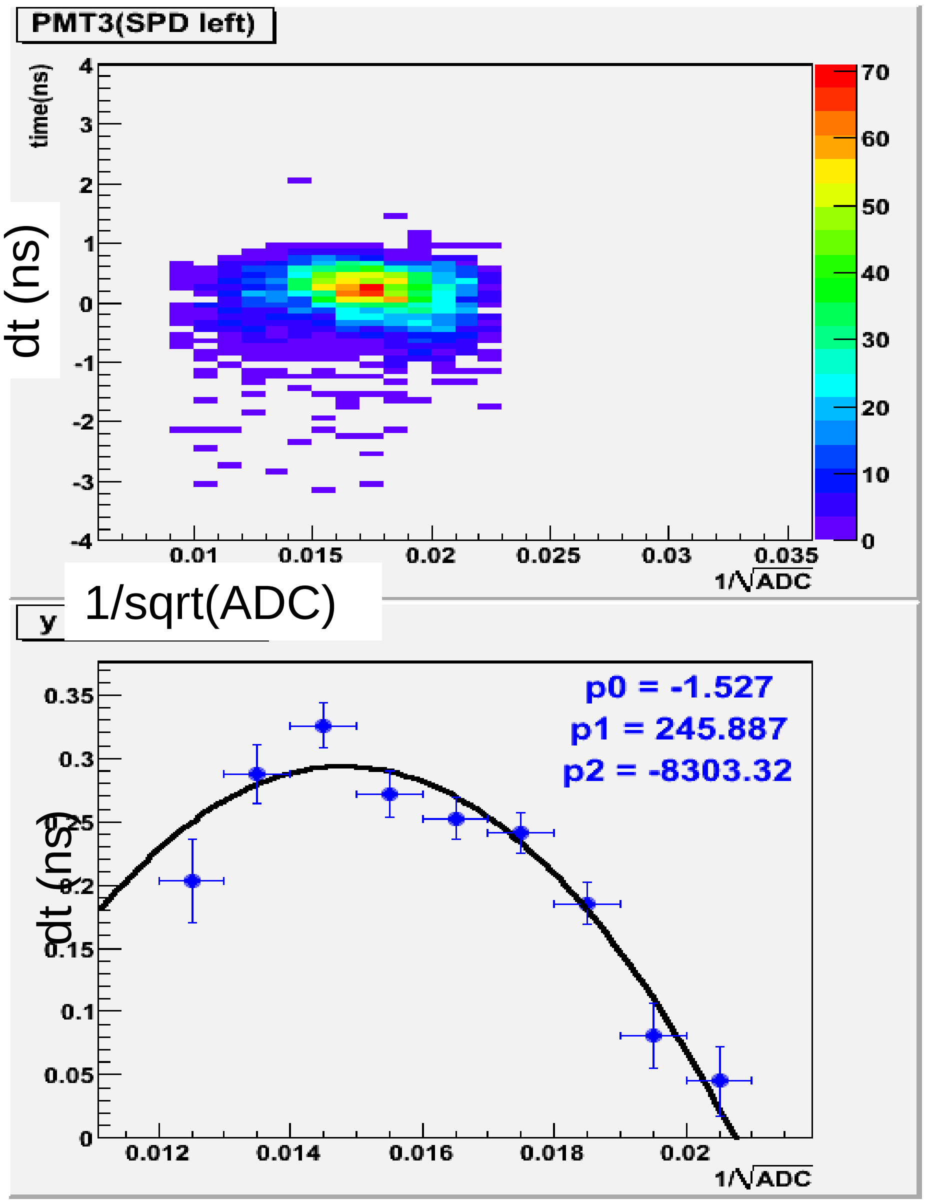


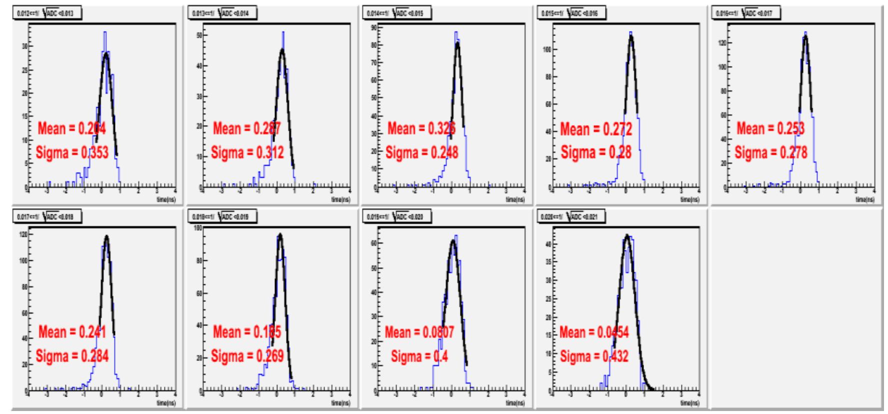


dt = Time_PMT – TriggerTime

- For all PMTs
- Apply trigger cut and ADC cut
- Fit each vertical slices to get the mean and sigma, then fit “mean Vs 1/sqrt(ADC)” by 1st order polynomial, using its uncertainty from the fit as error bar of each mean value.

New Time-Walk Calibration, Step 2





dt = Time_PMT – TriggerTime

- For all PMTs
- Apply trigger cut and ADC cut
- Apply 1st iteration correction
- Fit each vertical slices to get the mean and sigma, then fit “mean Vs 1/sqrt(ADC)” by 2nd order polynomial, using its uncertainty from the fit as error bar of each mean value.
- The fitted range is manually picked

X-Y-Z 3D Calibration

dt = Time_PMT – TriggerTime

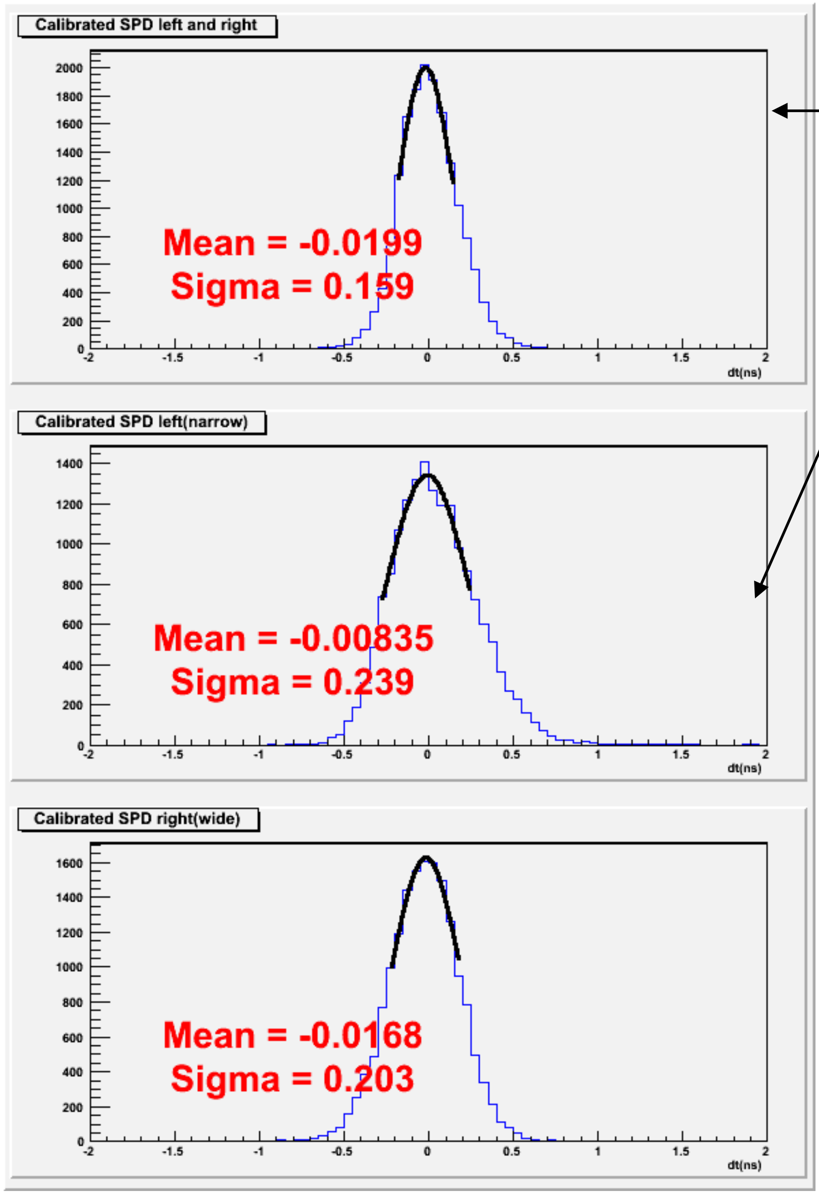
Chi^2 = [dt – f(X,Y,ADC)]^2

F(X,Y,ADC) = p0*x + p1*x*x
+ p2*y + p3*y*y + p4*x*y
+ p5 + p6*pow(ADC,-0.5) + p7*pow(ADC,-1.0)

For each PMT, minimized the chi^2.
Problems: Very hard to fit. Need to give accurate initial values.

Solution: I use the result of 2^{nd}-order-fitting to set initial value of p5, p6, and p7, and set p0 – p4 to zero. Start from here, then do multiple iterations. Each iteration manually set the previous result as new initial values. But still have problems.

New Result of Data Set E



dt_LR = (t1+t4 + t2 + t5)/4 – (t0+t3)/2

dt_L = (t1+t4 + t2 + t5)/4 – t3

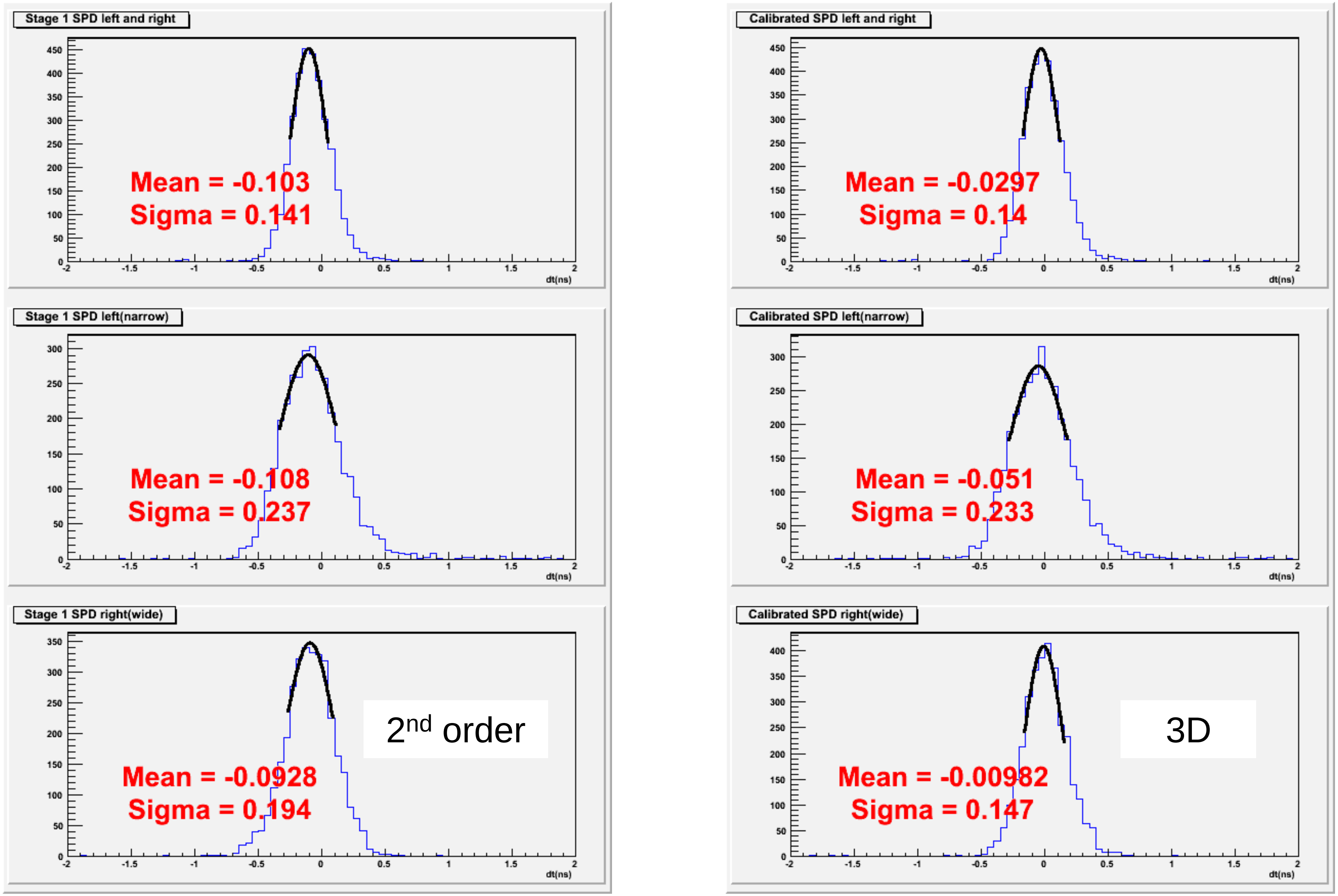
dt_R = (t1+t4 + t2 + t5)/4 – t0

Mean: indicates how good is the calibration

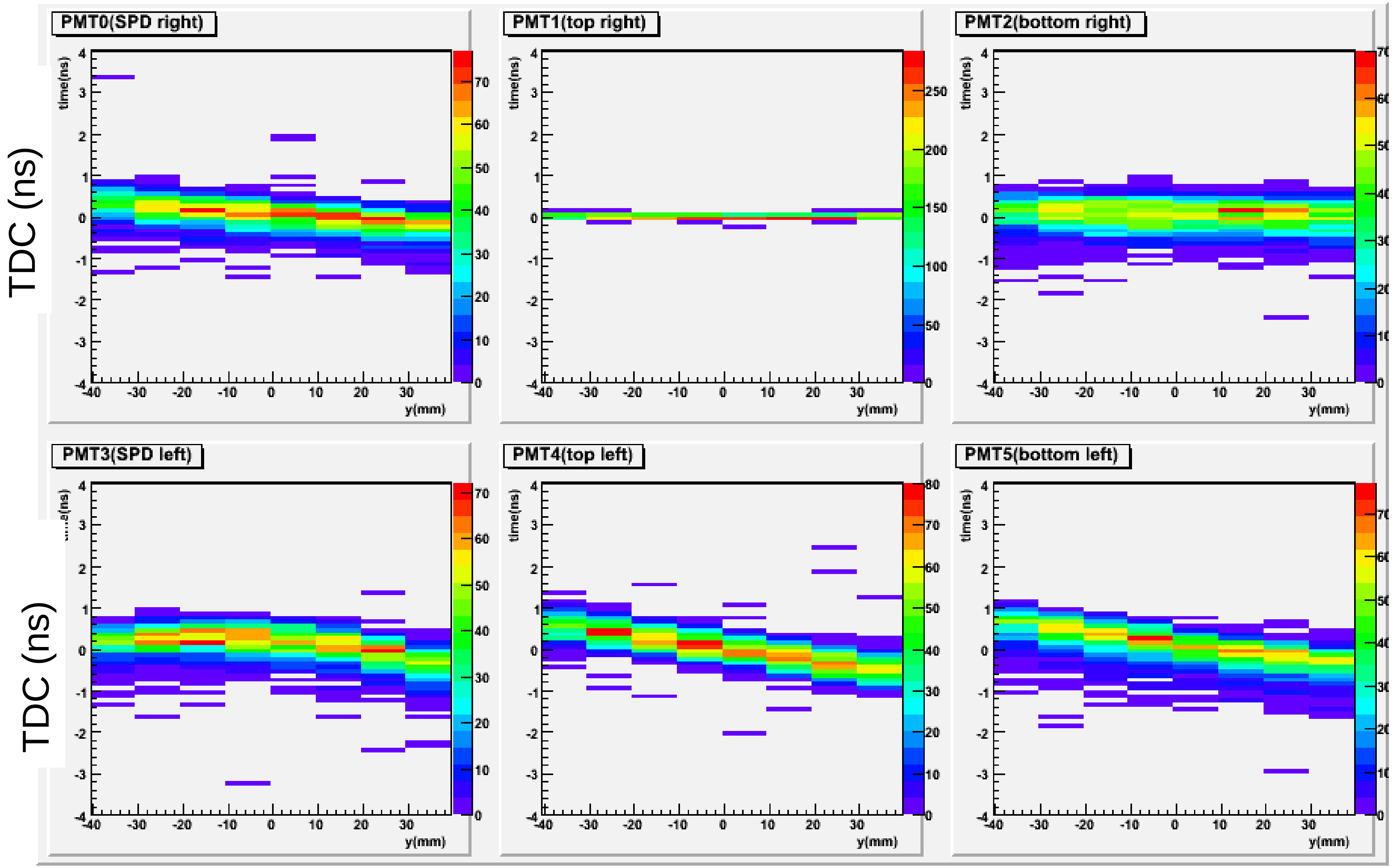
Sigma: time resolution

Overall result after the new calibration. Not require any information from GEM yet.

X-Y-ADC 3D Calibration

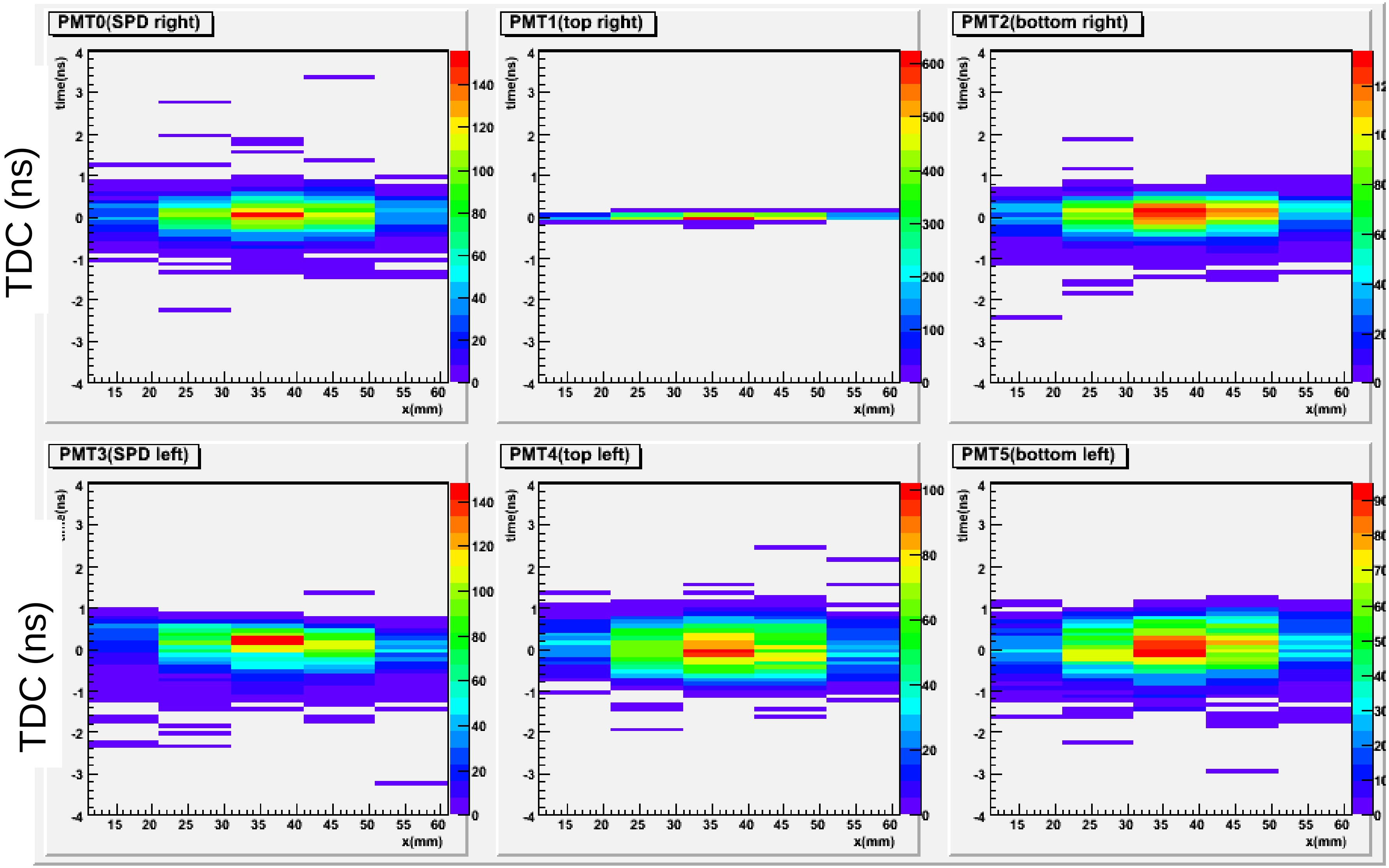


X-Y-ADC 3D Calibration



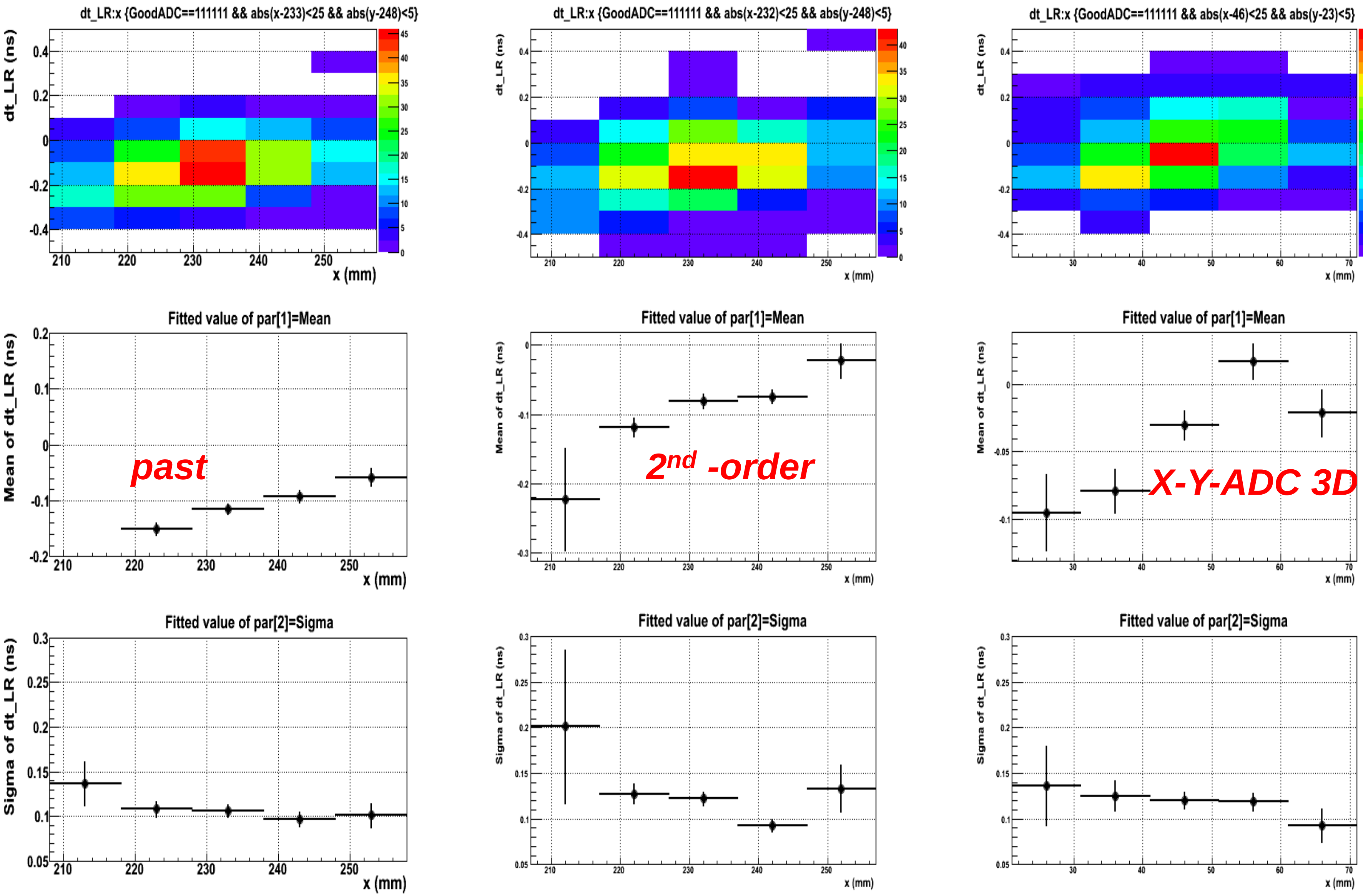
Calibrated TDC, Y dependence if over fitted for some PMTs

X-Y-ADC 3D Calibration



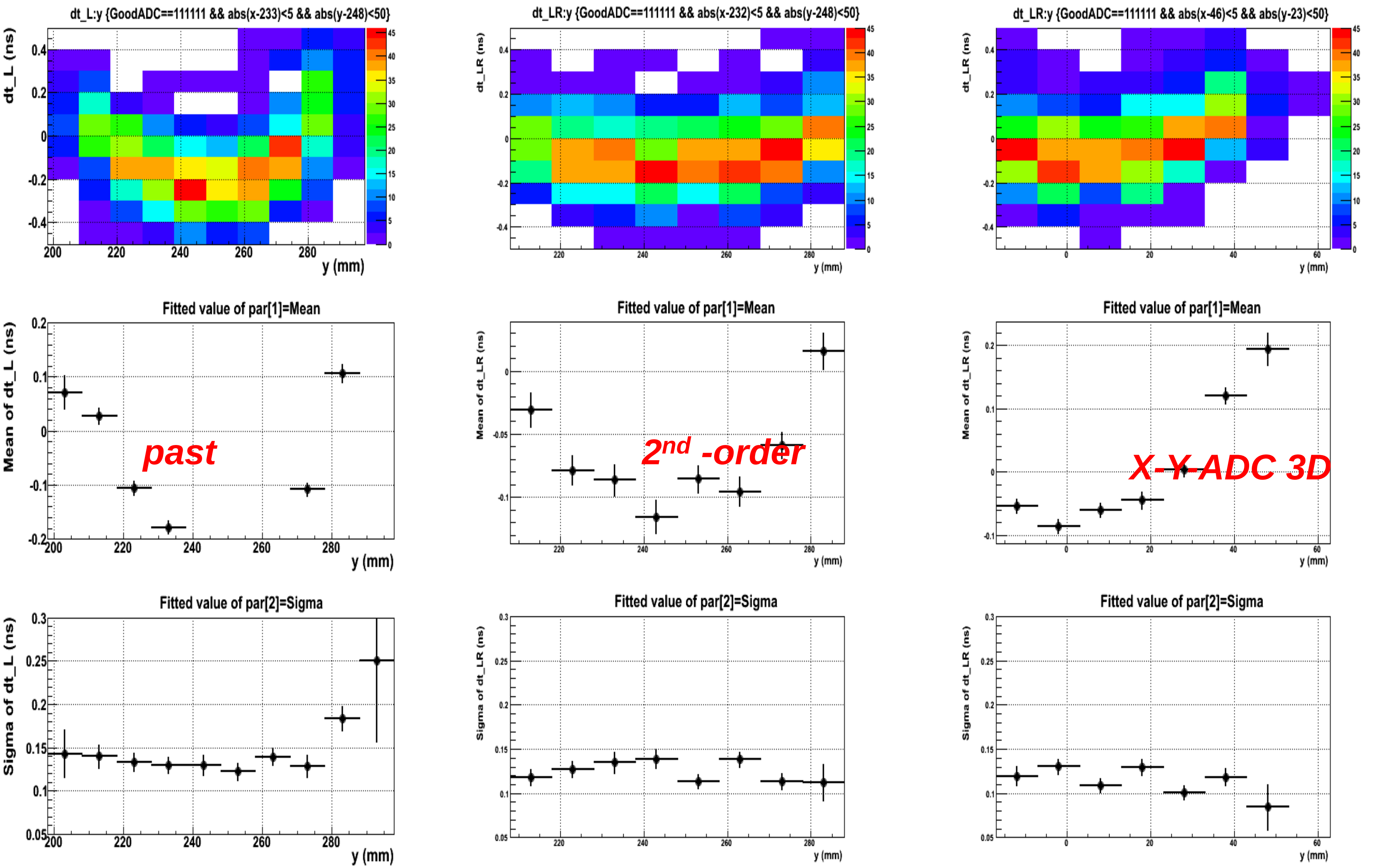
Calibrated TDC, X dependence seems fine.

Data Set E: 1cm x 1cm, dt_LR vs x



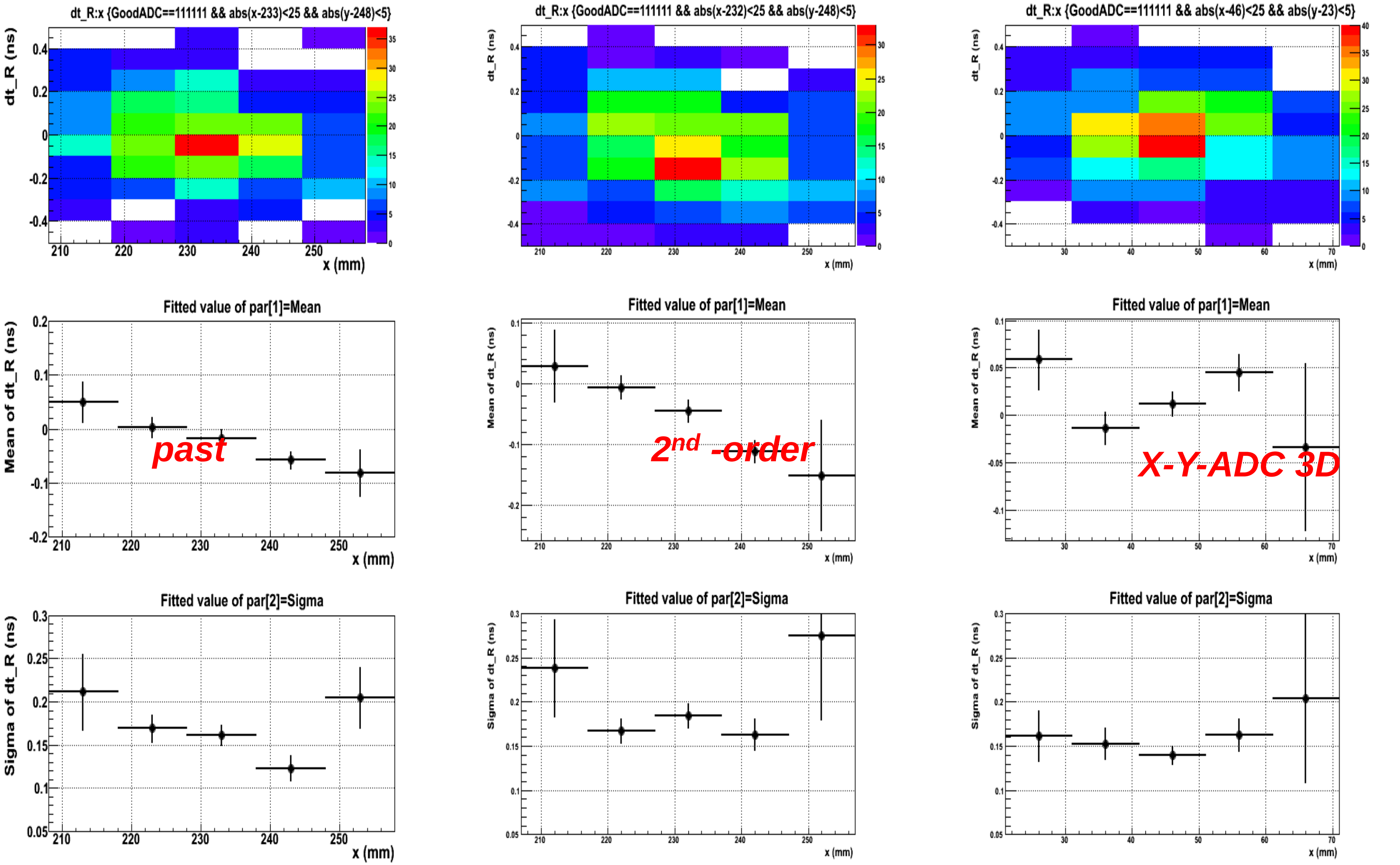
$dt\_LR = (t1+t4 + t2 + t5)/4 - (t0+t3)/2$

Data Set E: 1cm x 1cm, dt_LR vs y



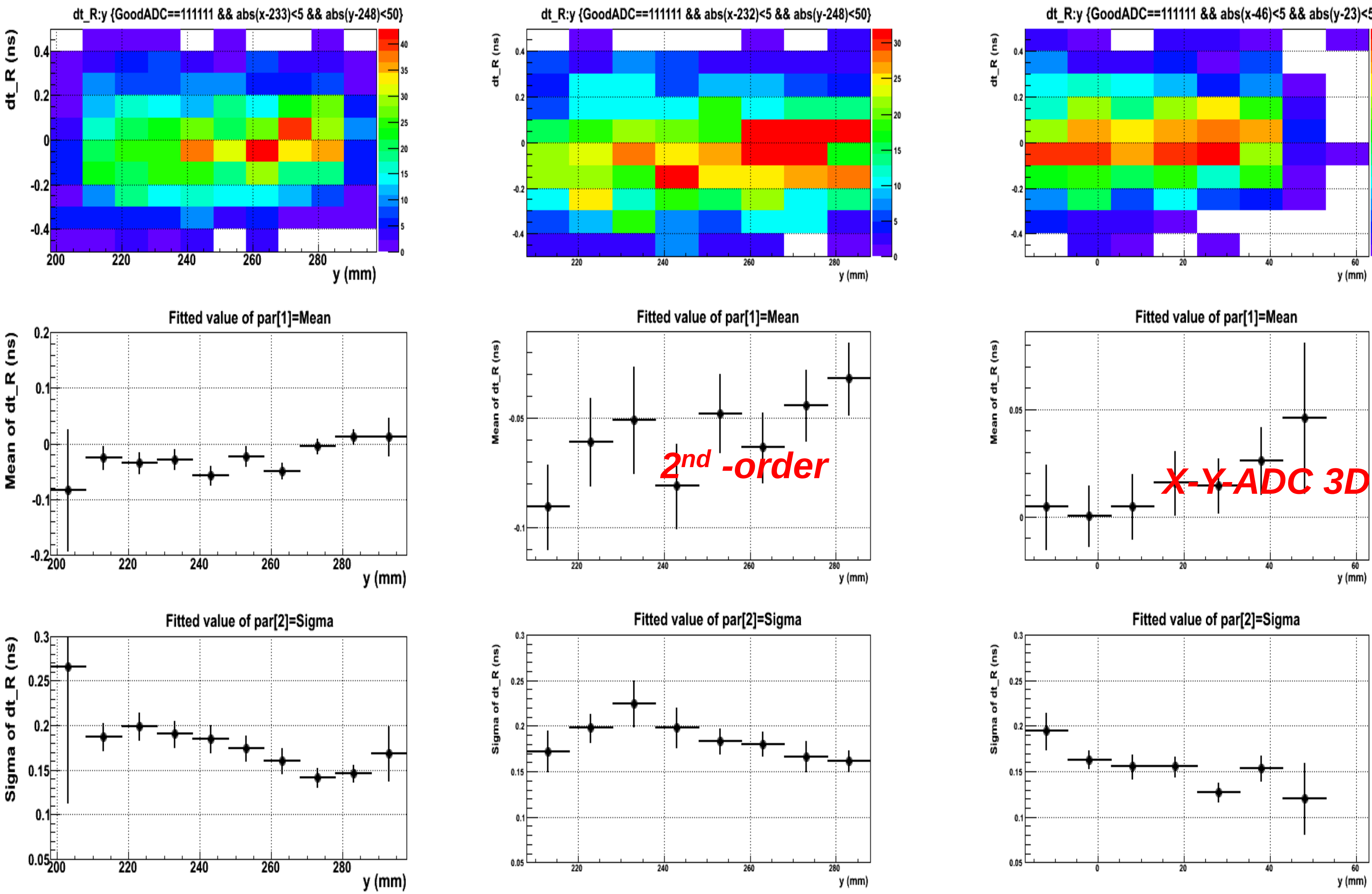
dt_LR = (t1+t4 + t2 + t5)/4 – (t0+t3)/2

Data Set E: 1cm x 1cm, dt_R vs x



$dt\_R = (t1+t4 + t2 + t5)/4 – t0$

Data Set E: 1cm x 1cm, dt_R vs y



dt_R = (t1+t4 + t2 + t5)/4 – t0

Summary and Outlook

- Tried 2nd-order polynomial time-walk calibration. Its result is not better than before.

- Did x-y-ADC 3-D calibration in data set E. It is very tricky to do the fit. Although the fitted result is obvious over

- corrected in y dependence, but the time resolution is better that the result in the past.

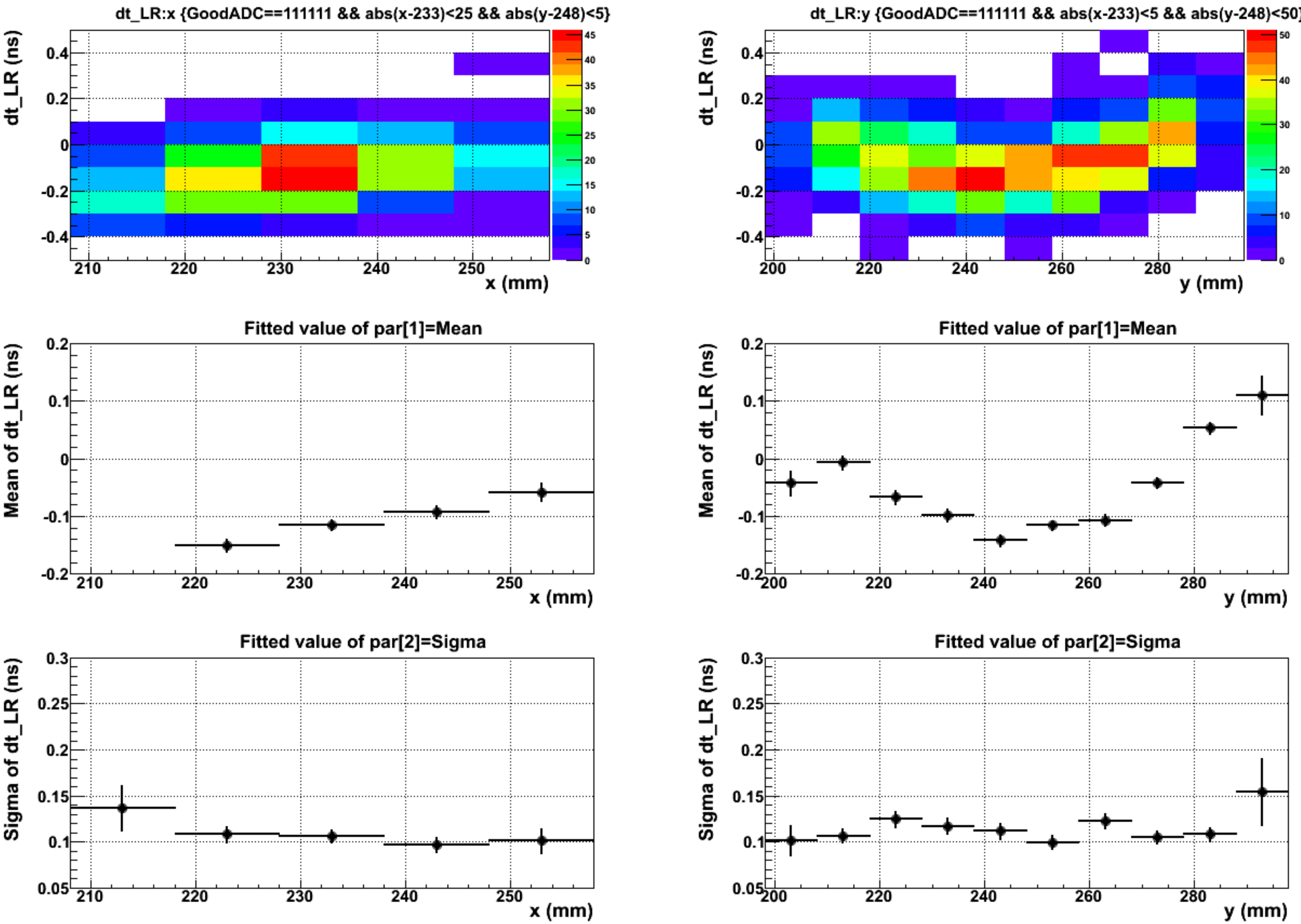
- Need more fine tuning on 3-D fitting.

- Other data sets is still pending.

Available Data

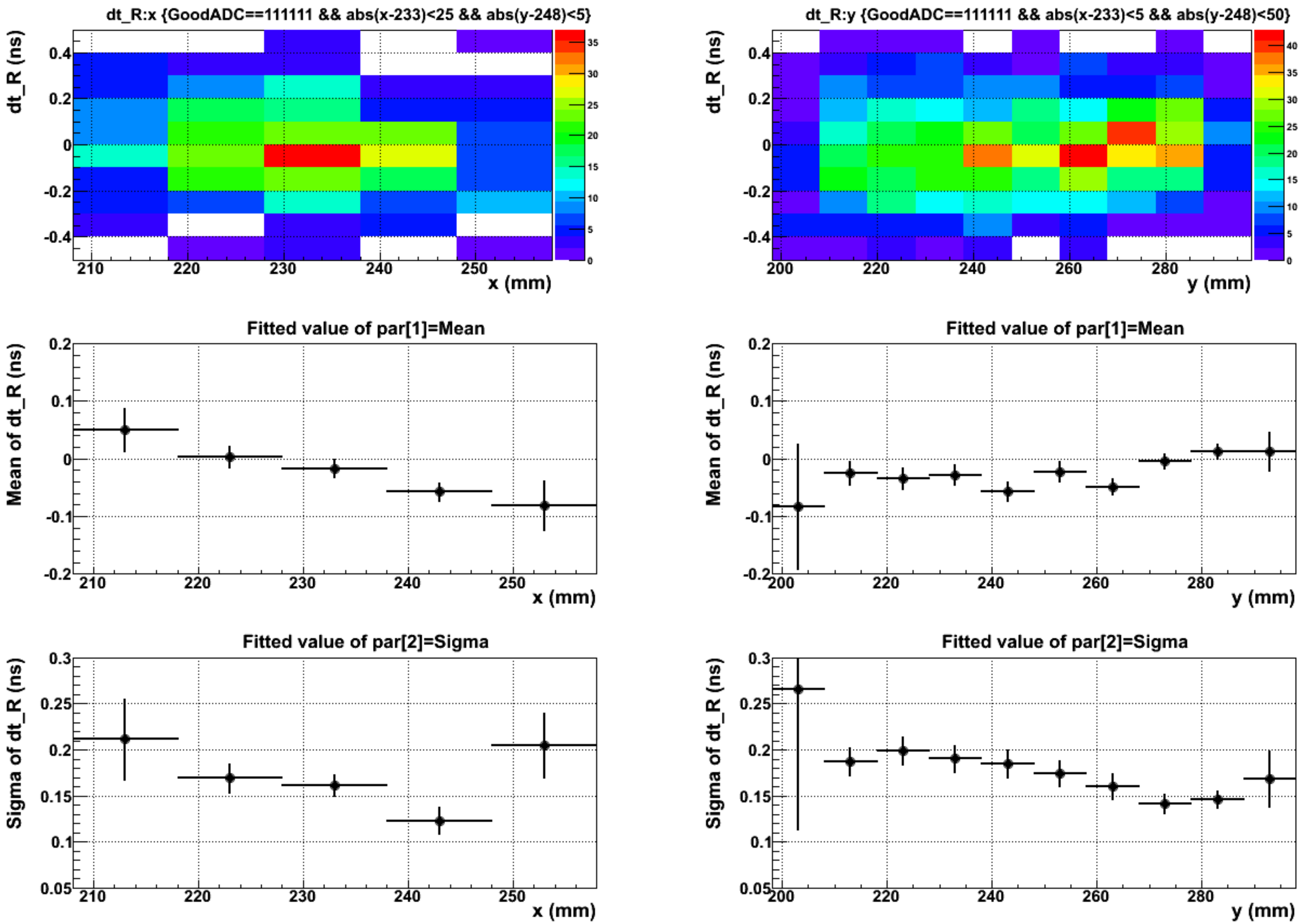
Data-set	A0+A1	B0+B1	C	D	E	F
Trigger#	363.1k	26.3k	33.0k	25.3k	28.6k	95.9k
+ADC	219.1k	16.7k	21.7k	17.0k	17.1k	53.9k
+GEM	55.8k	2828	3660	2859	2908	7770
#/cm^2	372	57	73	57	58	155

Best Result: dt_LR



dt_LR = (t1+t4 + t2 + t5)/4 – (t0+t3)/2

Best Result: dt_R



dt_R = (t1+t4 + t2 + t5)/4 – t0