CLAS12 Drift Chamber Calibration: Updates

Taya Chetry
Mississippi State University
(For the DC Calibration team)

CLAS Collaboration Meeting
06/20/2019

DC Calibration Team

Group Leader

Mac Mestayer (Jlab)

Calibration Suite Optimization/Maintenance

Taya Chetry (MISS)

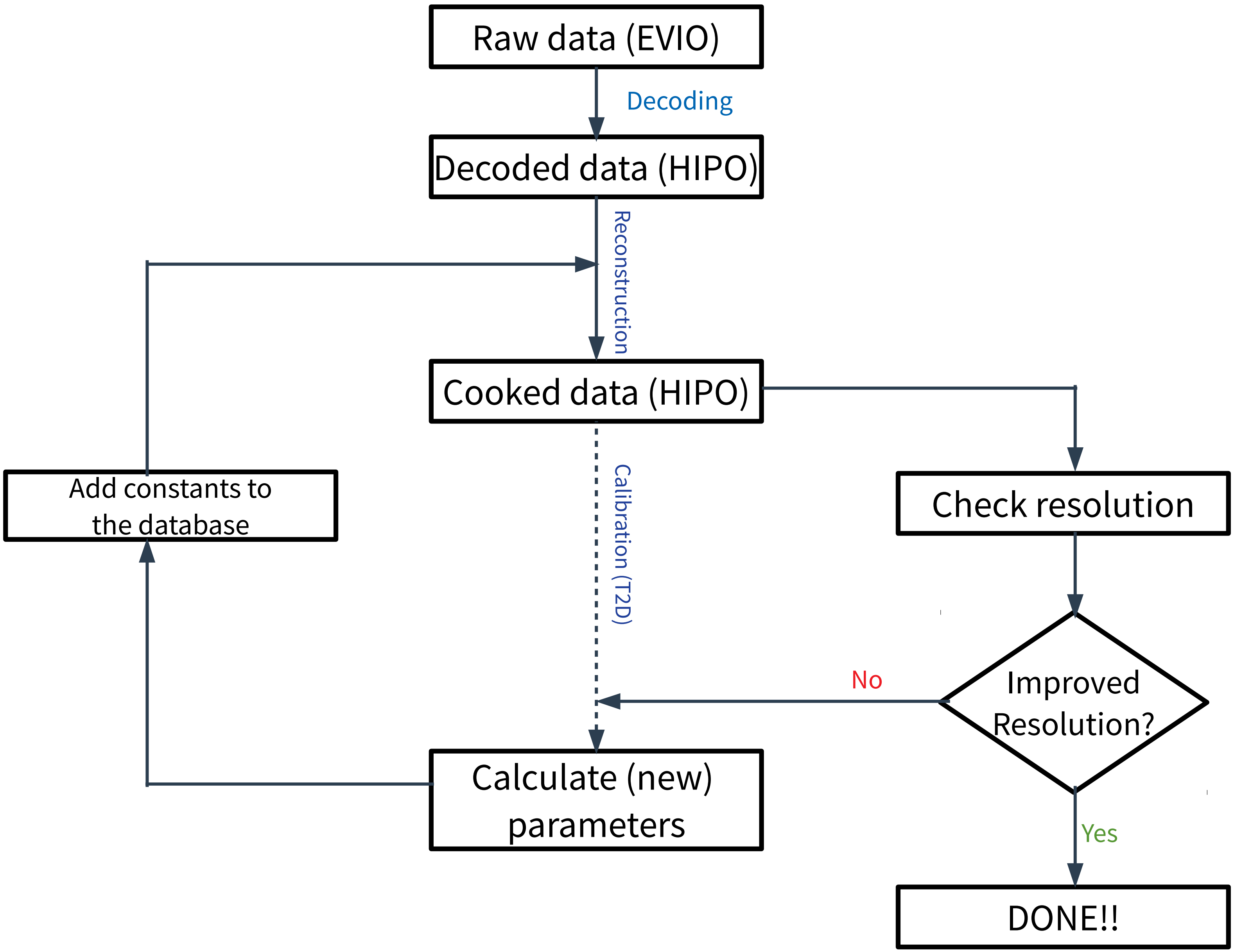
Calibrators

Dilini Bulumulla (ODU), Shirsendu Nanda (MISS)

Reconstruction

Veronique Ziegler (Jlab)

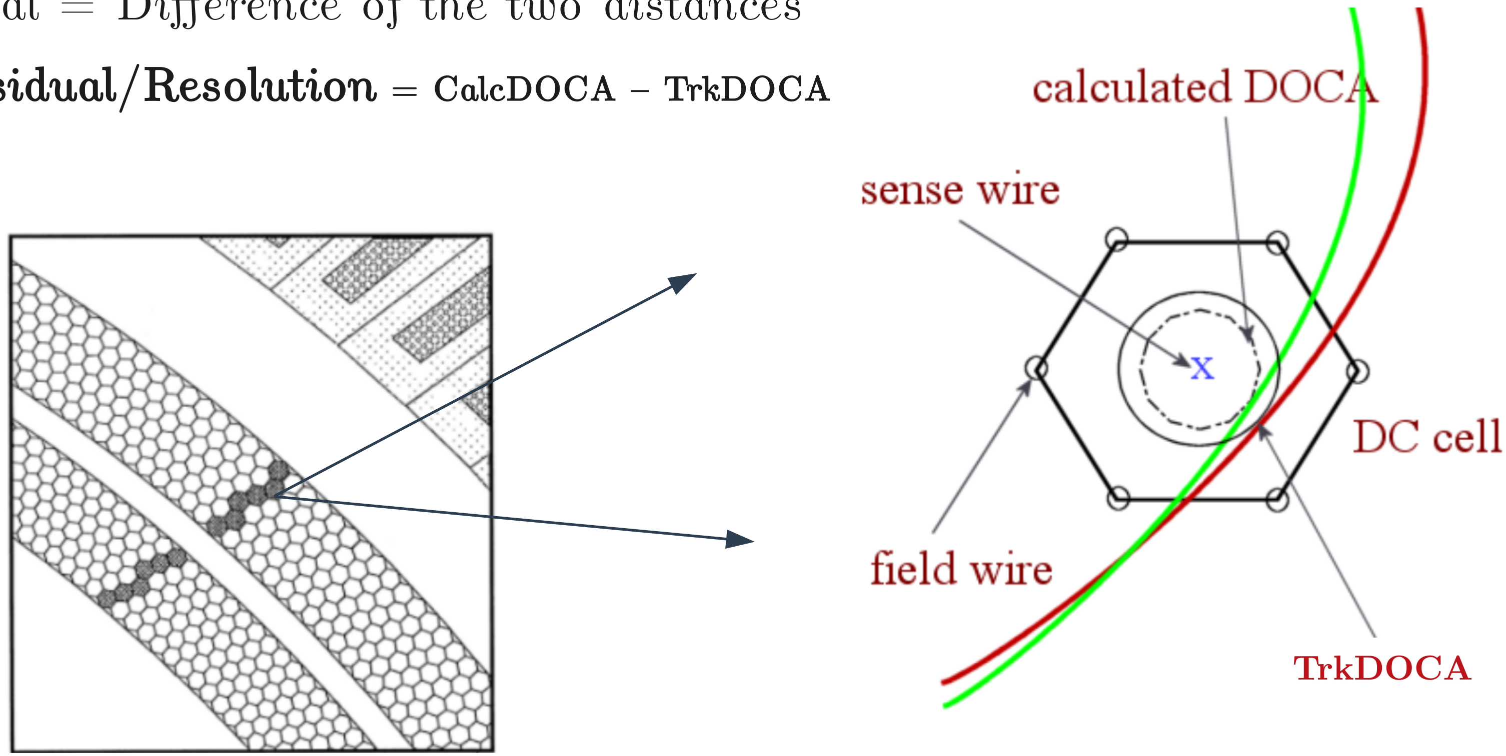
The Calibration process at a glance:



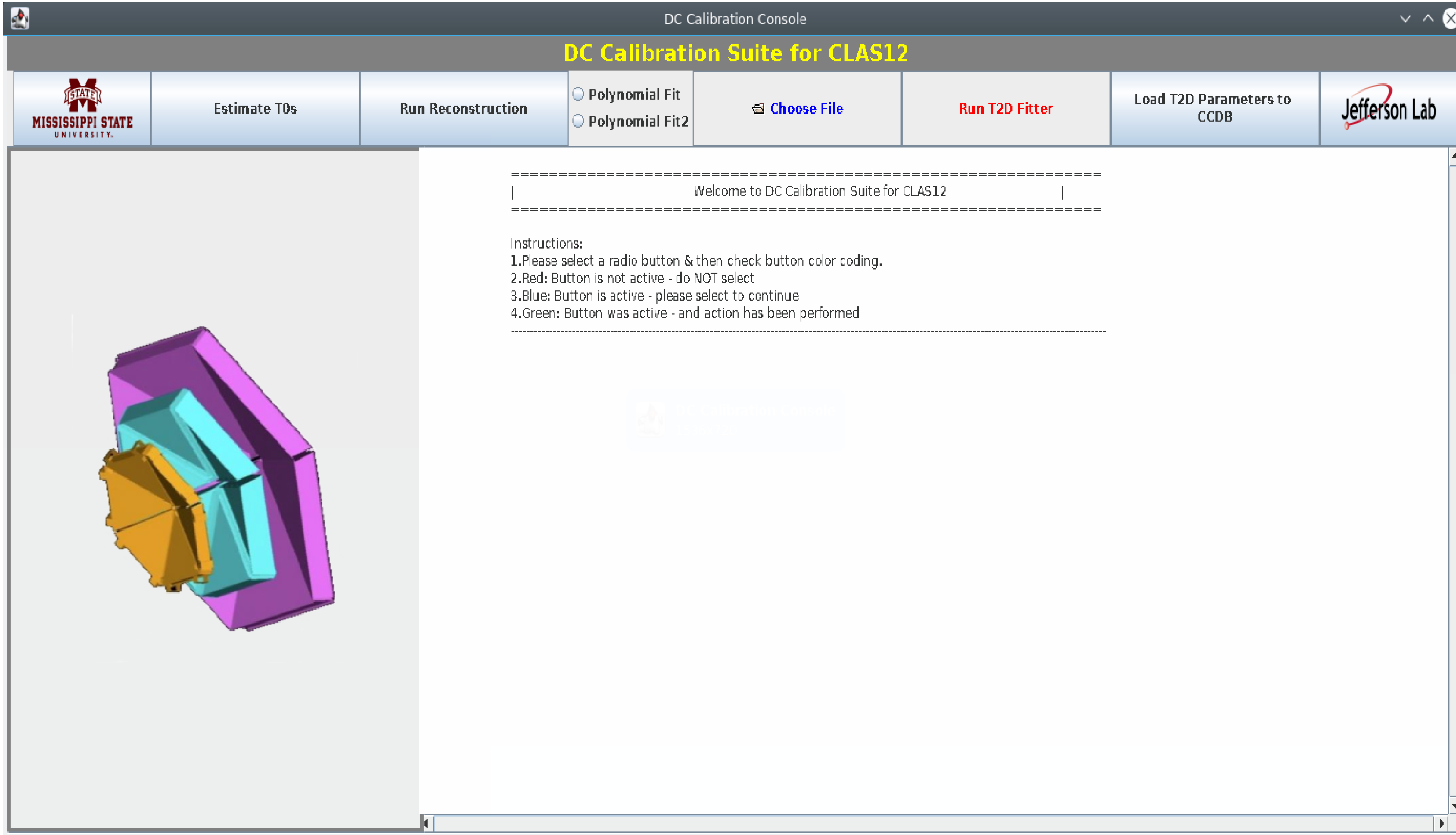
'Jargons' in DC Calibration:

- **Goal: To achieve resolution of ~250-350 μm:**
  - Use Time-Based-Tracking (TBT) along with the Hit-Based-Tracking (HBT).
  - TBT → utilizes the time information.
  - Calculate Distance of Closest Approach (DOCA) from TBT. This distance is known as the CalcDOCA.
  - HBT→ to calculate DOCA based on the hits, known as the Tracked DOCA (TrkDOCA).
  - Time Residual = Difference of the two distances

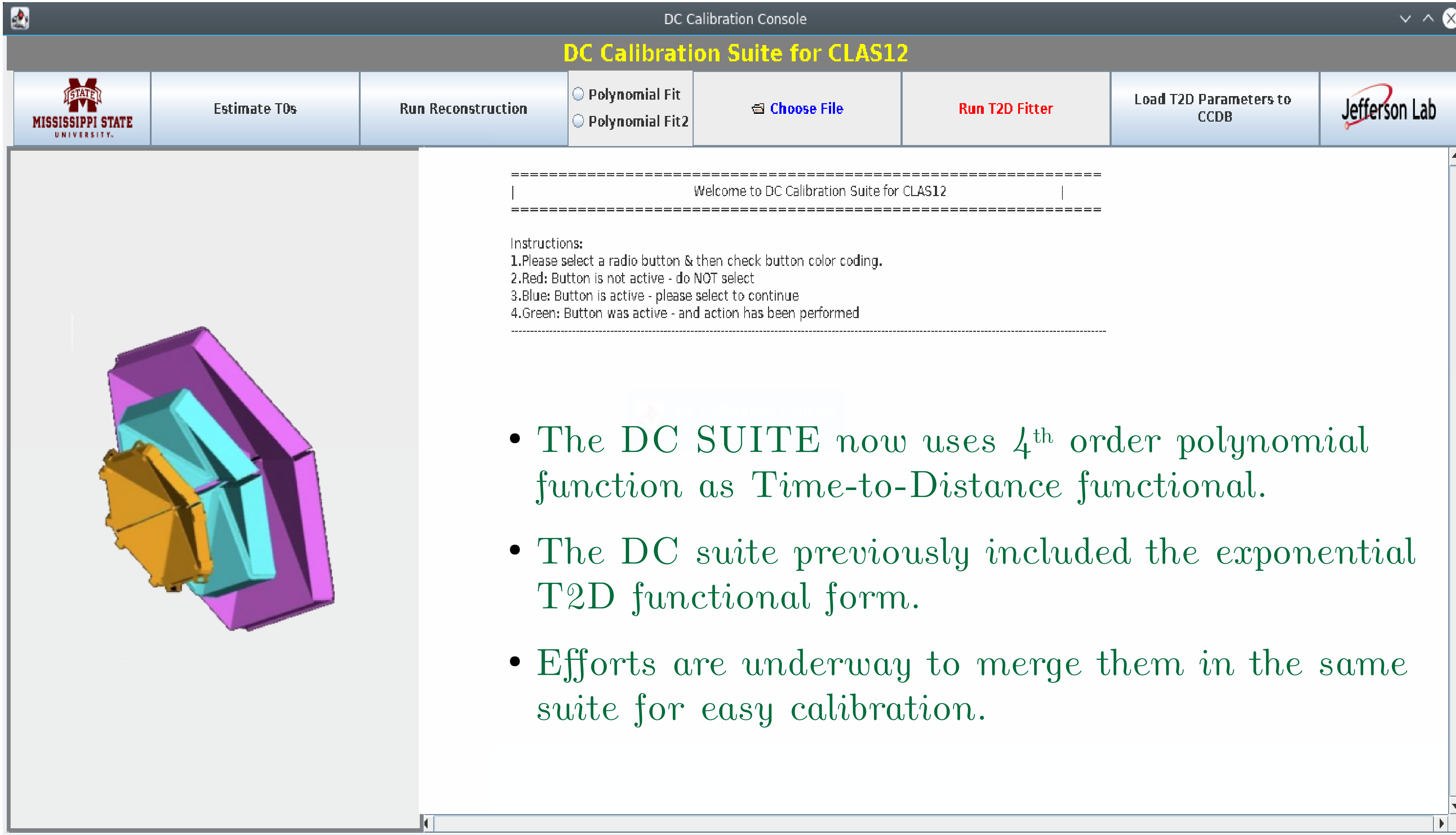
Residual/Resolution = CalcDOCA – TrkDOCA



DC Calibration Console



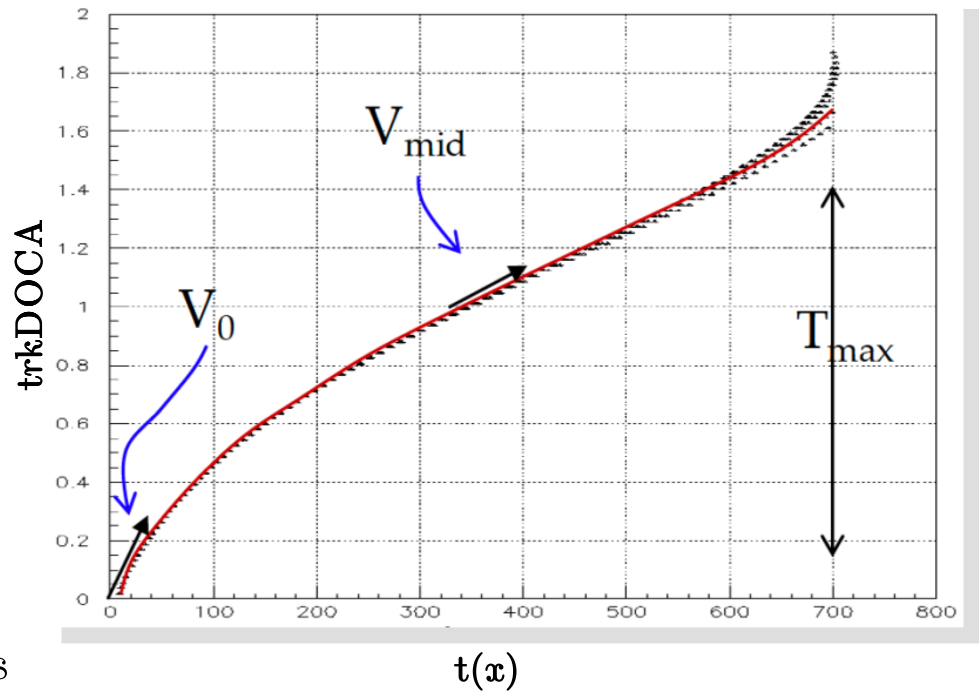
DC Calibration Console



- The DC SUITE now uses 4^{th} order polynomial function as Time-to-Distance functional.
- The DC suite previously included the exponential T2D functional form.
- Efforts are underway to merge them in the same suite for easy calibration.

- Polynomial function:
  - $t(x) = ax + bx^2 + cx^3 + dx^4$

    where, x = trkDOCA
- The equation is solved using 4 constraints:
  - Velocity at $x = 0$ is the saturated drift velocity, **v0**; $v_0 = 1/d$
  - Inflection point at $x = r$ is the parameter **r**. (maximum distance is referred as the dmax)
  - Velocity at the inflection point is the parameter **vmid**.
  - Time at dmax*cos(30-α) is **tmax**, where α is the local angle.

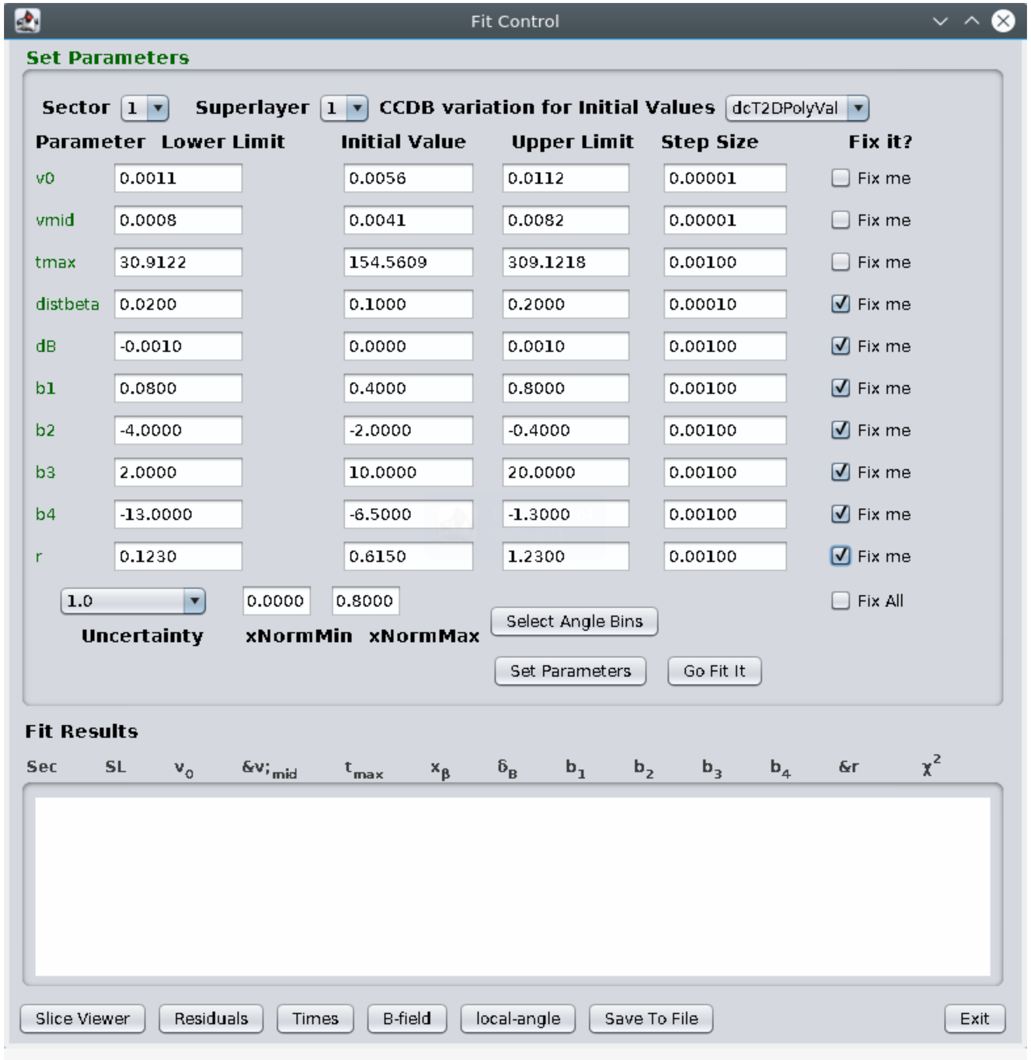


Understanding the parameters

Poly4

- Polynomial function:
  - ▹ $t(x) = ax + bx^2 + cx^3 + dx^4$

    where, x = trkDOCA
- There are 4 constraints used to solve:
  - ▹ Velocity at x = 0 is the saturated drift velocity, **v0**; so that v0 = 1/d
  - ▹ Inflection point at x=r is the parameter **r**. (maximum distance is referred as the dmax)
  - ▹ Velocity at the inflection point is the parameter **vmid**.
  - ▹ Time at dmax*cos(30-α) is **tmax**, where α is the local angle.



- The drift time is given by:

$$t = TDC - T_{start} - T_{flight} - T_{prop} - T_{0} - T_{beta}$$

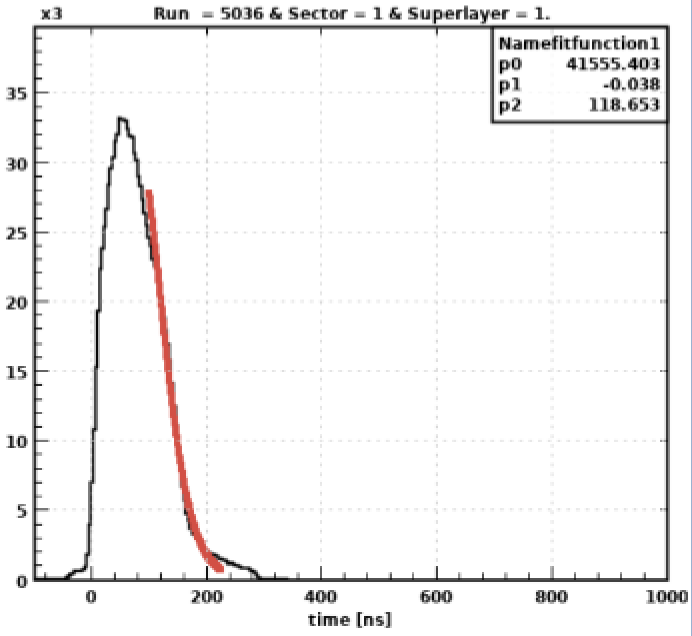
TDC time is corrected for trigger jitter and latency, flight time of the track, time of propagation of the signal along the wire to the readout, cable delay, and beta dependent time-walk correction

Show code →

tmax *Calculation*

- Different approaches to calculate tmax :
  - Fitting the trailing edge with a function (sigmoid).

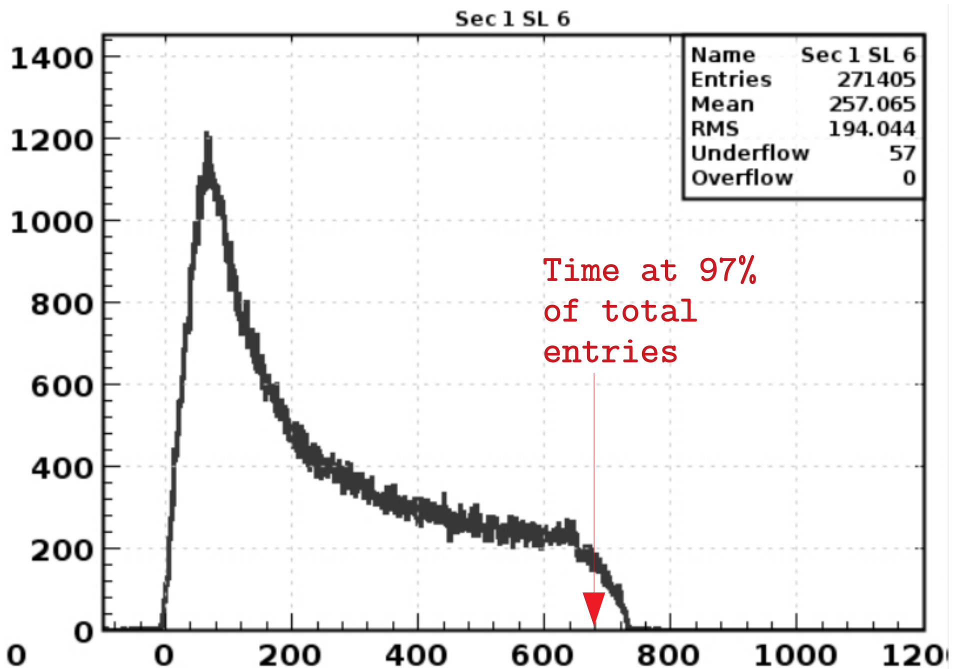
- $y(t) = \frac{p_0}{1+\exp^{-p_1(t-p_2)}}$
- Used the slope $(\frac{dy}{dt})_{t=p_2}$ and $y(t = p_2)$ to extrapolate a straight line
- The interception of that extrapolated straight line on time axis is the value of t_{max}
- Derived $t_{max} = p_2 - (\frac{2}{p_1})$
- An example, for run#5036, sector 1, and superlayer 1 $t_{max} = 118.653 - (\frac{2}{-0.038}) = 171.28$ ns

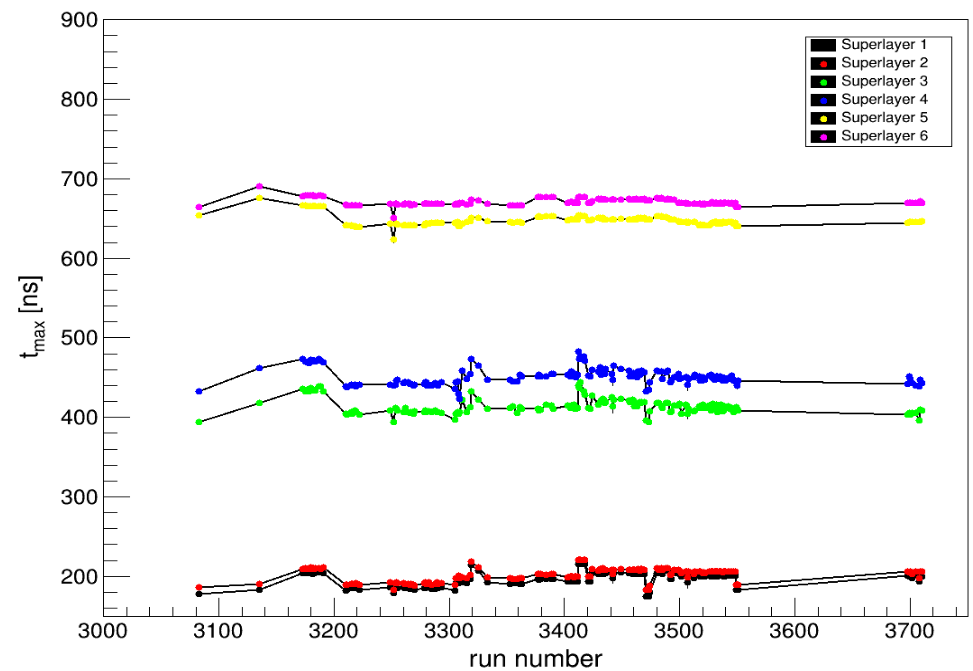


(courtesy: Shirsendu)

tmax *Calculation*

- Different approaches to calculate tmax :
  - Fitting the trailing edge with a function (sigmoid). $y(t) = \frac{p_0}{1+\exp^{-p_1(t-p_2)}}$
  - 97% of total entries
    - The time distribution is integrated bin by bin.
  - When the integrated events reach 97% of the total entries, the bin corresponds to the tmax.
  - A tmax timeline can be utilized as a re-calibration criteria.

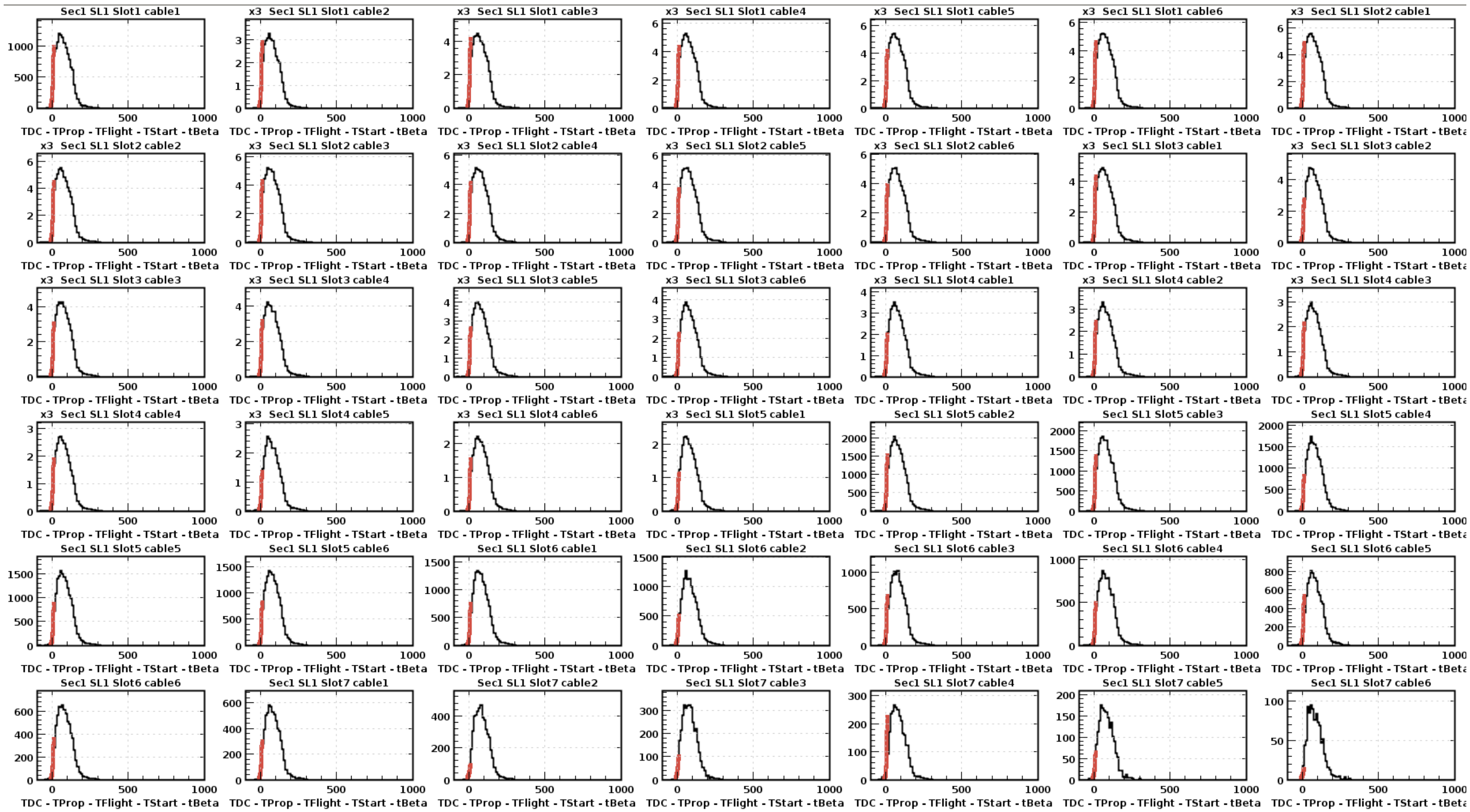




(courtesy: Shirsendu)

t0 *Calculation*

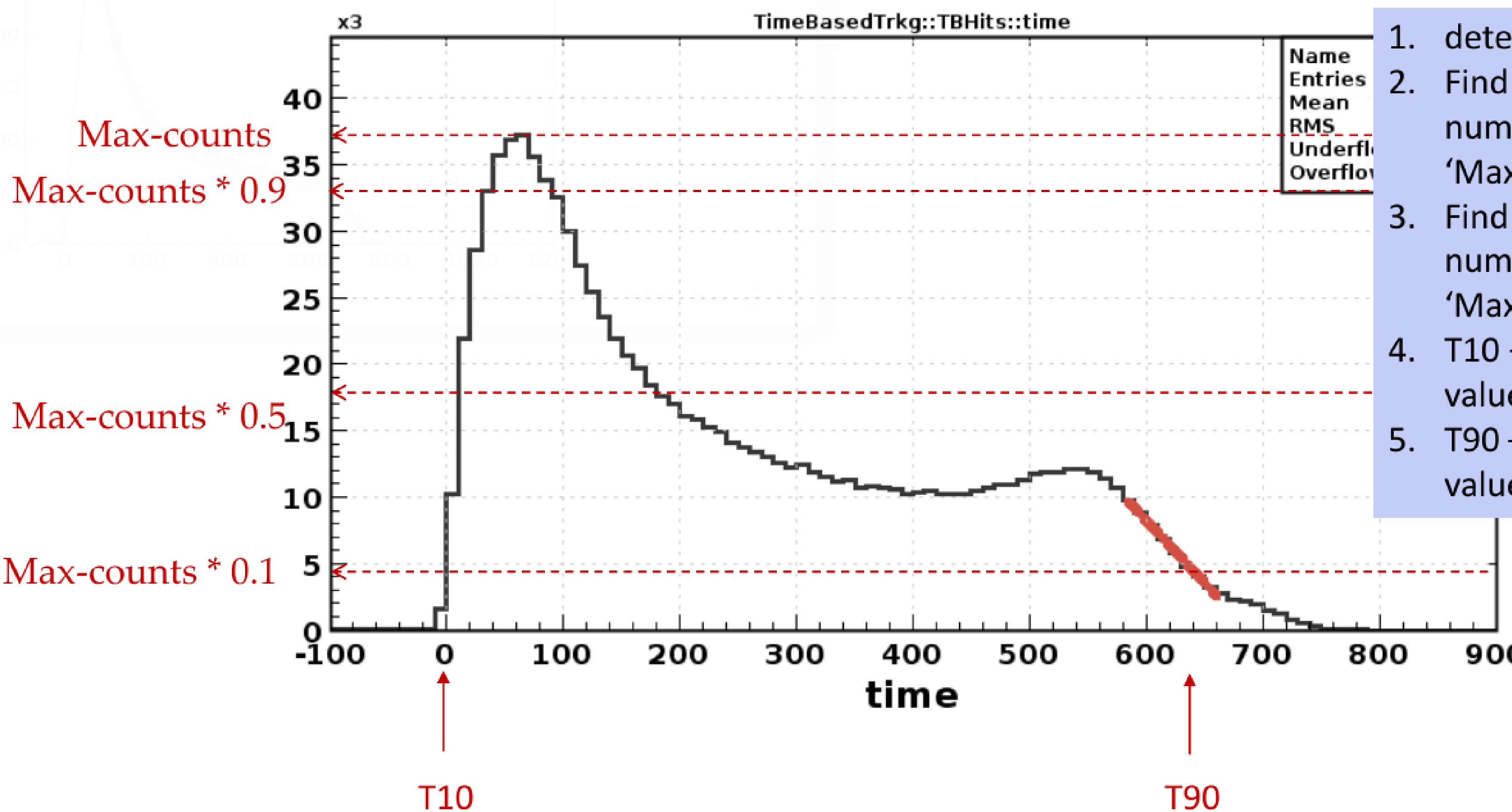
- Fitting the leading edge with a function (sigmoid). $y(t) = \frac{p_0}{1+\exp^{-p_1(t-p_2)}}$
  - time = TDC - T_{Prop} - T_{Flight} - T_{Start} − T_{Beta}



- The plot contains 42 histograms for sector 1 and superlayer 1. Run 5300.

(courtesy: Shirsendu)

Getting starting values for parameters in `t0` and `tmax` fits



1. determine ‘Max-counts’
2. Find ‘T10’, 1st bin with number of counts > 0.1 * ‘Max-counts’
3. Find ‘T90’, 1st bin with number of counts < 0.1 * ‘Max-counts
4. T10 + 20 ns is starting value for `P2` in T0 fit
5. T90 – 40 ns is starting value for `P2` in Tmax fit

Work in progress!

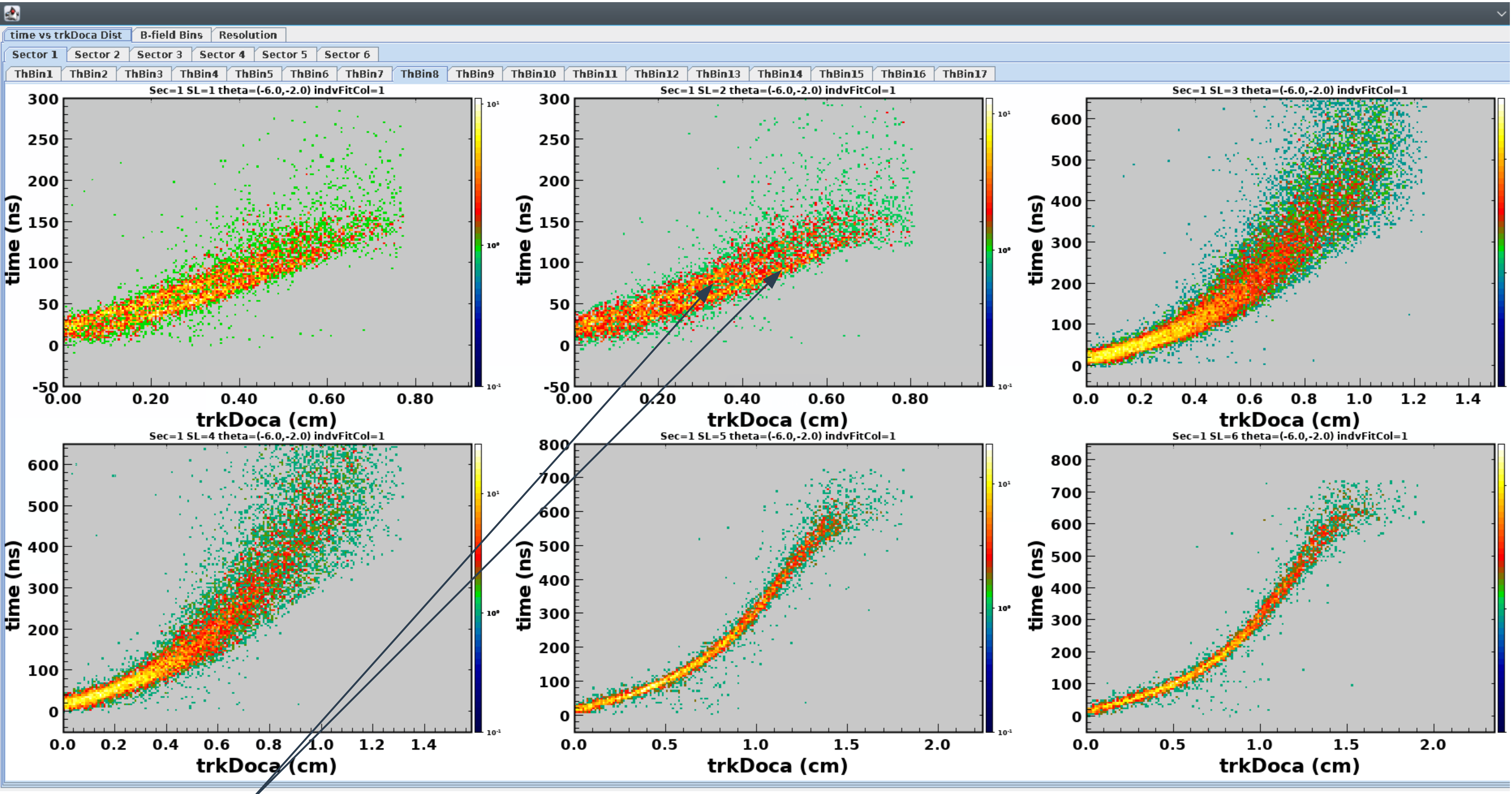
(slide courtesy: Mac)

Cook summary: 5038 (using exponential T2D functional)

Cook Level	μ (mean) [cm]						σ (resolution) [cm]					
V0	-0.011	-0.008	-0.004	0.0	-0.012	-0.014	0.038	0.036	0.036	0.038	0.039	0.039
V1	-0.013	-0.001	-0.004	-0.01	-0.018	-0.021	0.040	0.038	0.044	0.055	0.041	0.041
V2	0.006	-0.003	-0.005	-0.003	-0.022	-0.030	0.038	0.037	0.051	0.047	0.040	0.042
V3	-0.006	-0.006	-0.057	0.043	-0.024	-0.036	0.036	0.036	0.073	0.072	0.039	0.044
V1(Dilini)	-0.013	-0.003	-0.025	-0.029	-0.014	-0.016	0.040	0.037	0.038	0.048	0.035	0.036

Coatjava: 5.9.0; Hipo4

Double banded structures?



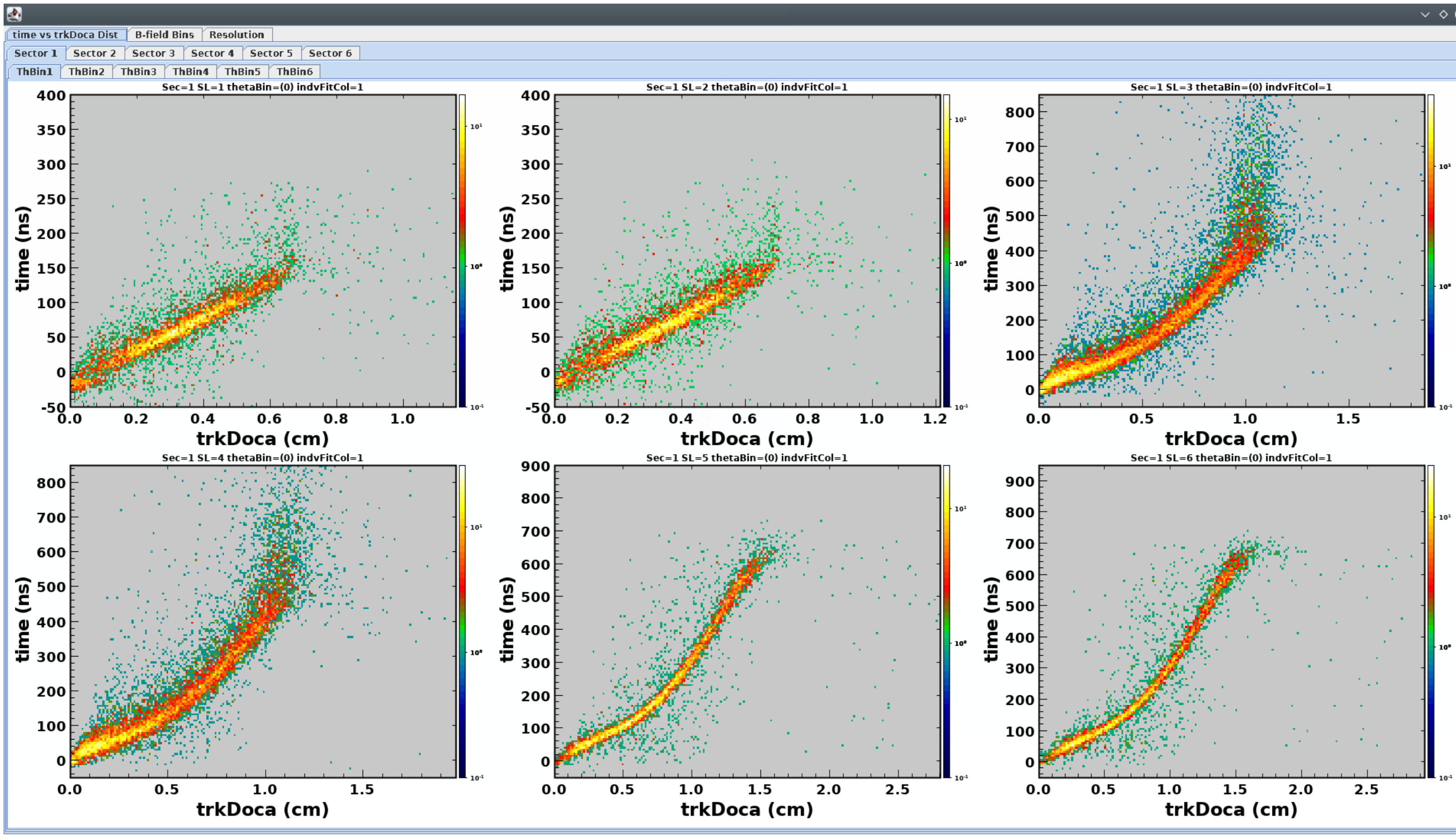
Double band structure??

Coatjava: 5.9.0; Hipo4

Cook 1: Run 5038

With Veronique's new reconstruction software, the double bands are gone!!

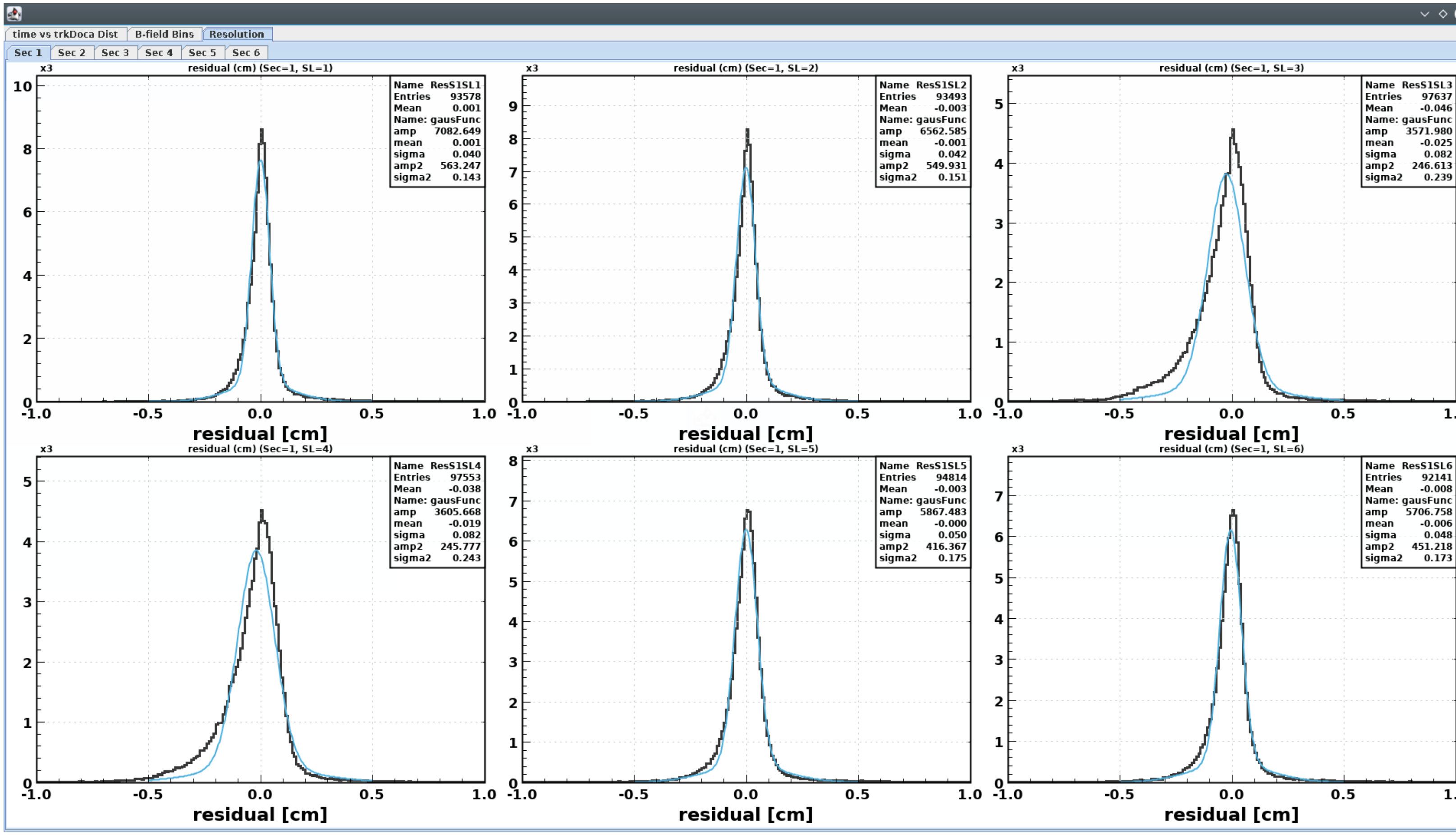
clas-offline-software	hipo4-Time2Dist4Cal
variation	dcT2DpolyVal
timestamp	
Run #	5038 (~15 k events)
Double band ?	No
Resolution	--



Cook 1: Run 5038

- Parameters were guessed to begin with.

clas-offline-software	hipo4-Time2Dist4Cal
variation	dcT2DpolyVal
timestamp	
Run #	5038 (~15 k events)
Double band ?	No
Resolution	--

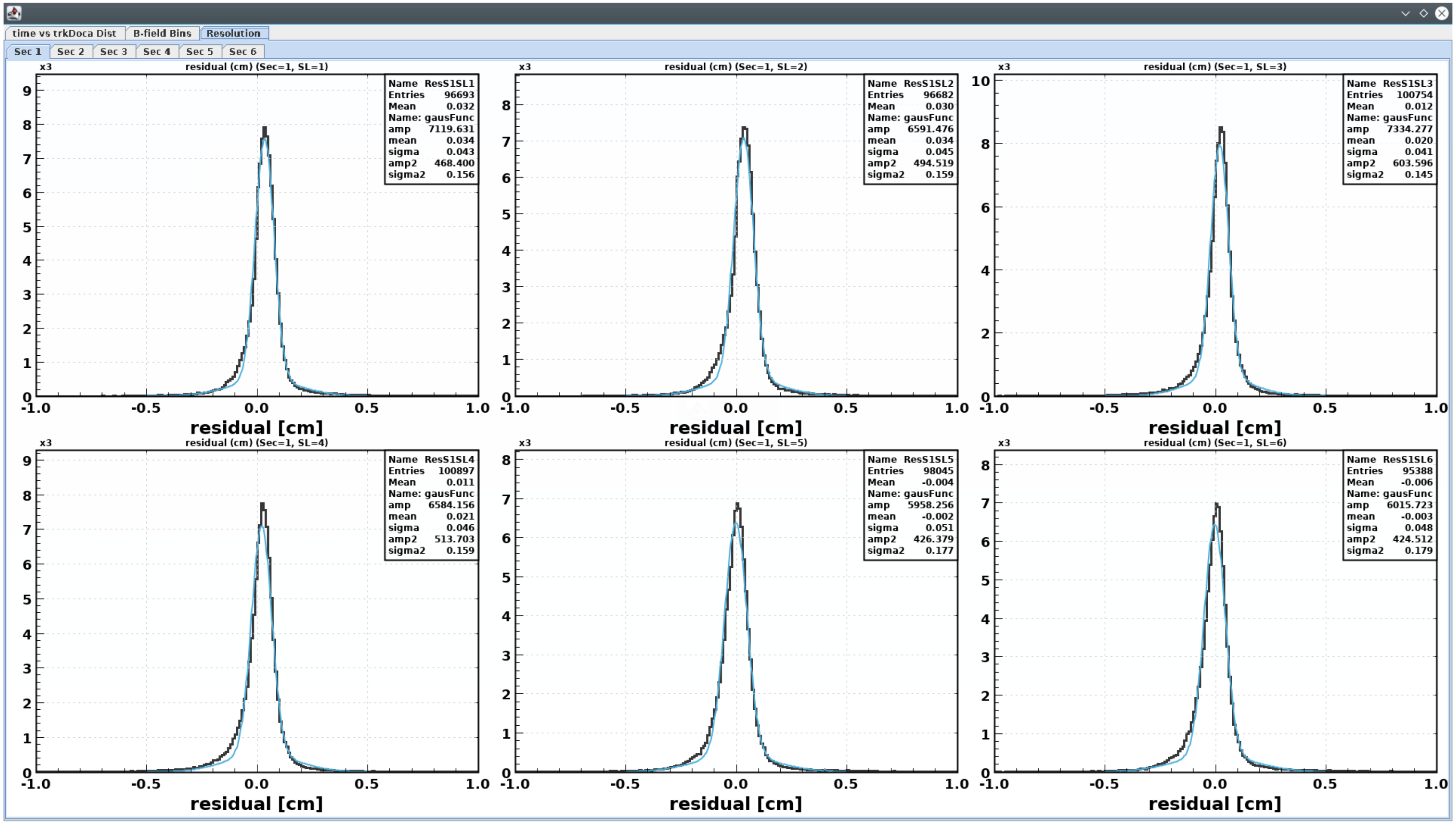


- Double gaussian fits. Width of the inner Gaussian gives the resolution.

Cook 2: Run 5038

- Learned from previous cook.

clas-offline-software	hipo4-Time2Dist4Cal
variation	dcT2DpolyVal
timestamp	06/17/2019
Run #	5038 (~15 k events)
Double band ?	No
Resolution	Slight improvement

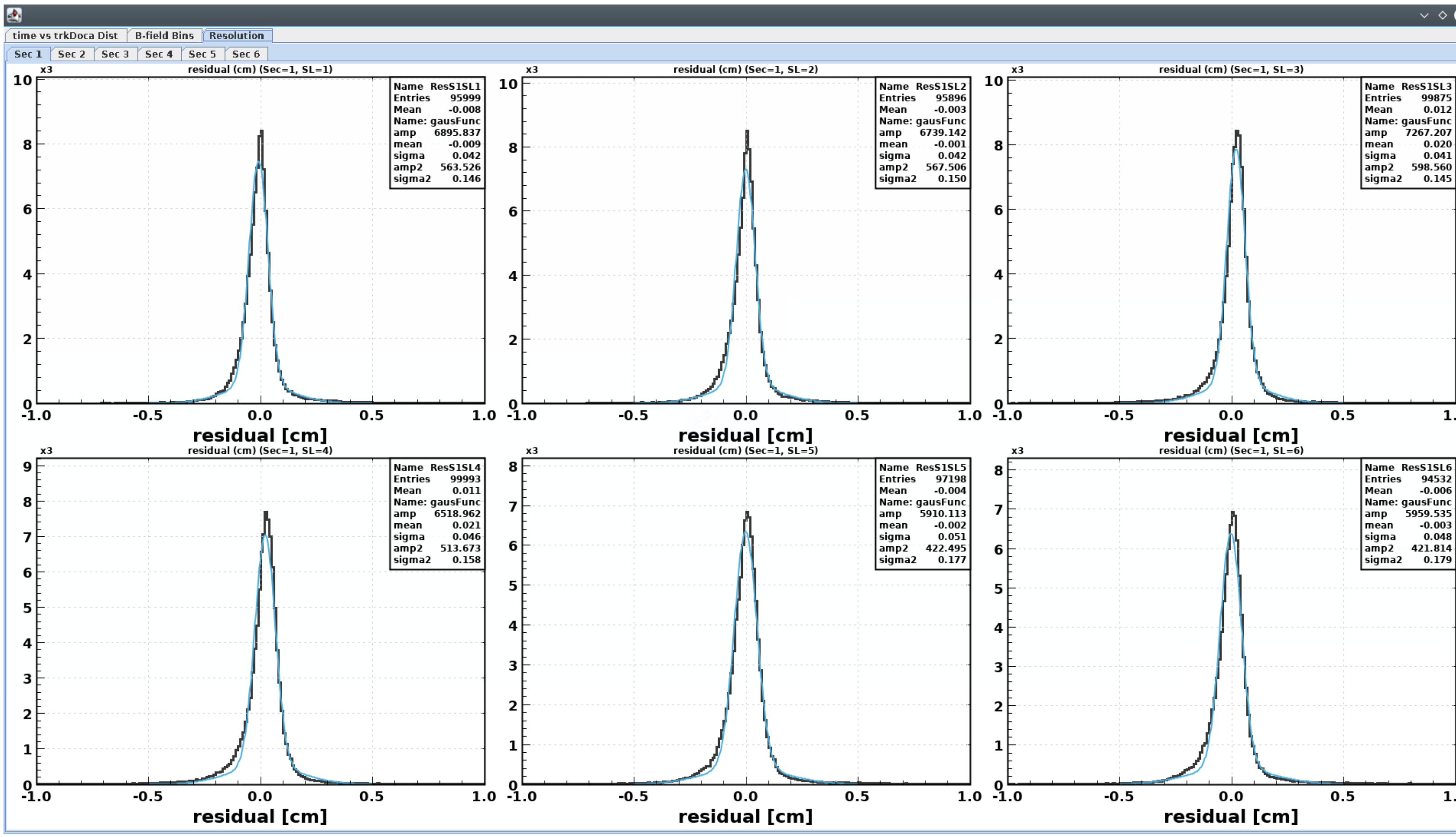


- Slight improvement in the resolution.

Cook 2: Run 5038

- Learned from previous cook.

clas-offline-software	hipo4-Time2Dist4Cal
variation	dcT2DpolyVal
timestamp	06/17/2019
Run #	5038 (~15 k events)
Double band ?	No
Resolution	Slight improvement



- Slight improvement in the resolution compared to previous cook.

Cook summary: 5038 (using 4th order poly T2D function)

Cook Level	μ (mean) [cm]						σ (resolution) [cm]					
1	0.001	-0.001	-0.025	-0.019	-0.018	-0.000	0.042	0.046	0.082	0.082	0.050	0.048
2	0.034	0.034	0.020	0.021	-0.002	-0.003	0.043	0.045	0.041	0.046	0.051	0.048
3	-0.009	-0.001	0.020	0.021	-0.002	-0.003	0.042	0.042	0.041	0.046	0.051	0.048

- Slight improvement in the resolution (not really – compared to what we expect!)
- It is important to iterate (and learn as we repeat) and extract the best practices for the calibration using the new functional: in progress!

Summary

- New 4th order polynomial time-to-distance function to better describe the data has been implemented in the calibration GUI.
- `t0` and `tmax` timelines for determining "When to calibrate".
- Sanity checks/iterations in progress for the optimal calibration parameters: "How to calibrate?"

Thank you!!

Extras

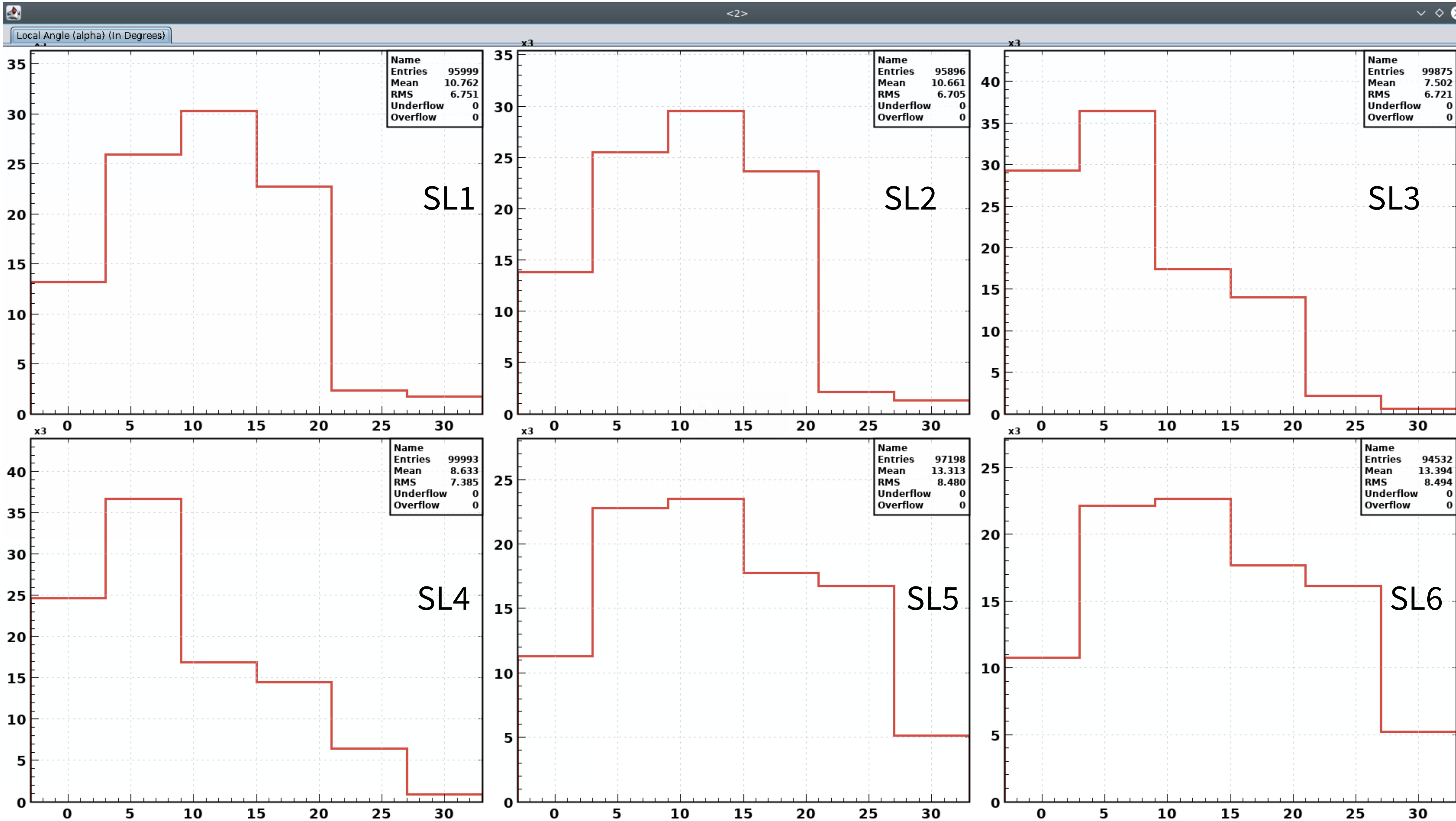
Poly 4: code snippet

```
double cosA = DMAX * TMath::Cos((30.0-ang)*PI/180.0);

double dmaxalpha = DMAX*cosA;
double xhatalpha = (double)x[0]/dmaxalpha;

double denom = cosA * cosA * DMAX * DMAX * (3.0 * cosA * cosA - 8.0 * cosA * r + 6.0 * r * r) * v0 *vmid;

a = (cosA * cosA * cosA * DMAX * (v0-vmid))
        - 3.0 * cosA * DMAX * r * r * vmid
        + 3.0 * cosA * cosA * DMAX * r * (vmid - v0) / (DMAX * DMAX * r * r * denom);

b = (6.0 * cosA * cosA * DMAX * r * r * (v0 - vmid)
    + 8.0 * cosA * DMAX * r * r * r * vmid
    - 8.0 * r * r * r * tmax * v0 *vmid
    + cosA * cosA * cosA * cosA * DMAX * (vmid - v0))/(DMAX * r * r * denom);

c = (3.0 * cosA * cosA * cosA * cosA *DMAX * (v0 - vmid)
    - 6.0 * cosA * DMAX * r * r * r * vmid
    + 6.0 * r * r * r * tmax * v0 * vmid
    + 6.0 * cosA * cosA * cosA * DMAX * r * (vmid - v0))/(r * denom);

d = 1.0/v0;

double tt = a*x[0]*x[0]*x[0]*x[0] + b*x[0]*x[0]*x[0] + c*x[0]*x[0] + d*x[0];


double deltatime_bfield = delBf * pow(bfield,2) * tmax *
                          ( b1 * xhatalpha
                          + b2 * pow(xhatalpha, 2)
                          + b3 * pow(xhatalpha, 3)
                          + b4 * pow(xhatalpha, 4) );

tt += deltatime_bfield;
return tt * pow(10.0, -9);
```

← Go back

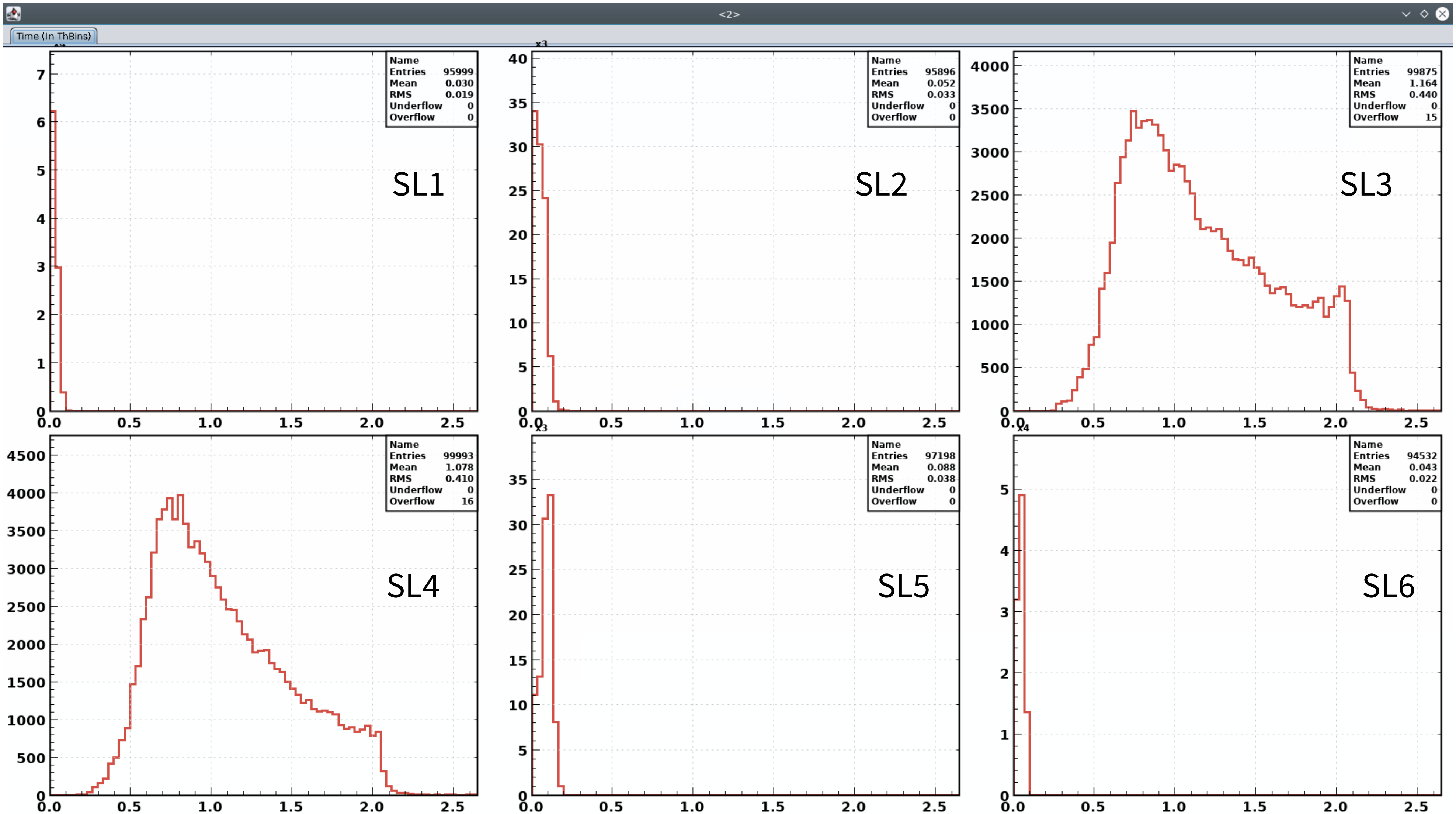
Local Angle (α)

Sector 1: Run 5038.00001



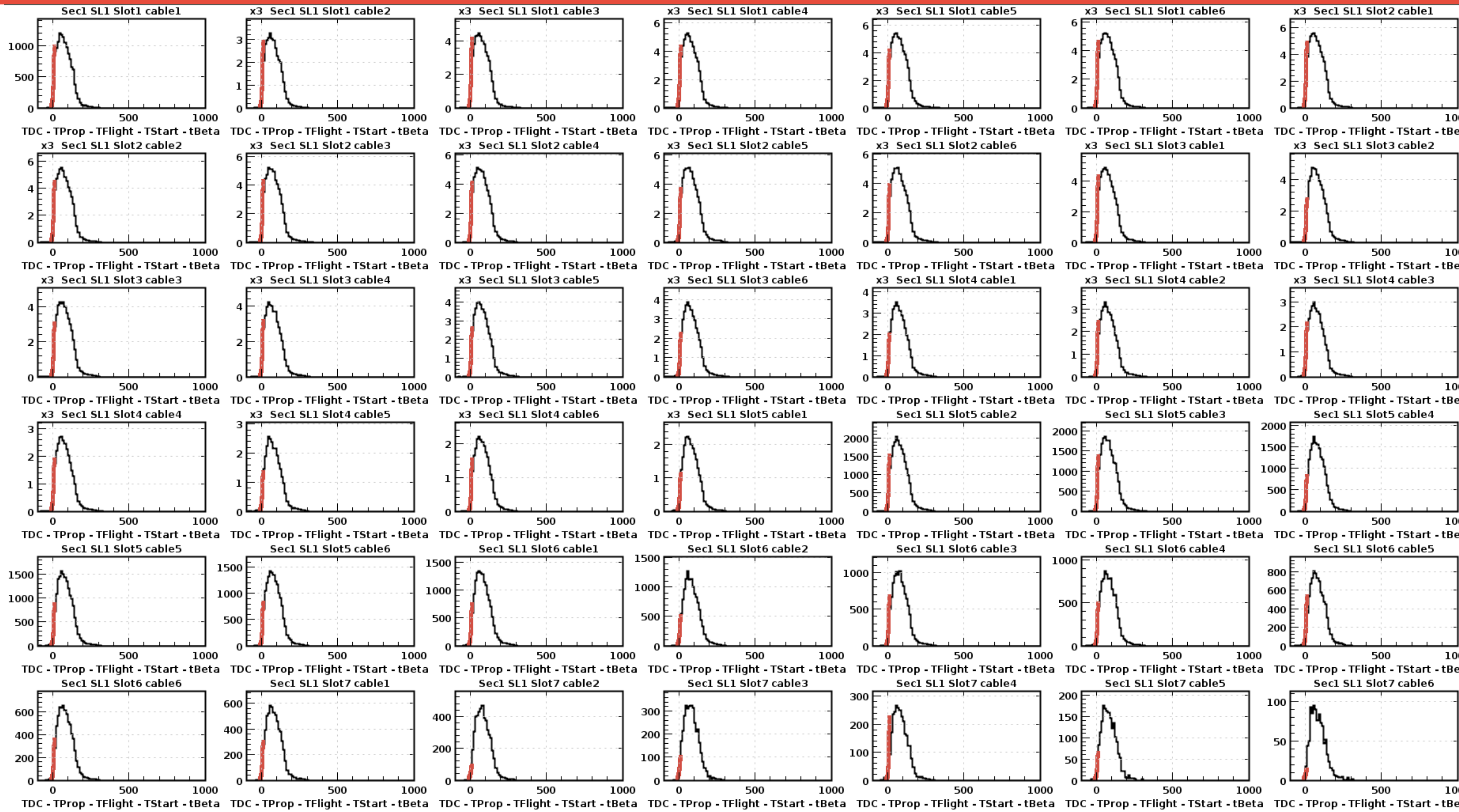
B-field (dB)

Sector 1: Run 5038.00001



hipo-utils -dump

HTCC::rec	21500	22	1	59562	32
HitBasedTrkg::HBClusters	20600	22	42	38076	2310
HitBasedTrkg::HBCrosses	20600	25	14	43103	812
HitBasedTrkg::HBHits	20600	21	279	23281	14787
HitBasedTrkg::HBSegments	20600	23	37	40394	2701
HitBasedTrkg::HBTracks	20600	26	2	43923	276
LTCC::adc	21600	11	2	6380	30
LTCC::clusters	21600	22	1	59422	44
LTCC::tdc	21600	12	1	16909	9
RAW::tdc	20000	12	48	17309	432
RAW::vtp	20000	14	945	17749	4725
REC::Calorimeter	300	32	5	71574	485
REC::Cherenkov	300	33	2	72228	88
REC::CovMat	300	38	2	73974	128
REC::Event	300	30	1	71524	42
REC::ForwardTagger	300	34	2	72324	100
REC::Particle	300	31	3	71399	117
REC::Scintillator	300	35	3	72067	153
REC::Track	300	36	2	72432	30
REC::Traj	300	40	44	72470	1496
RECHB::Calorimeter	300	12	5	59755	485
RECHB::Cherenkov	300	13	1	60409	44
RECHB::Event	300	10	1	59705	42
RECHB::ForwardTagger	300	14	2	60461	100
RECHB::Particle	300	11	3	59580	117
RECHB::Scintillator	300	15	3	60248	153
RECHB::Track	300	16	2	60569	30
RF::adc	21700	11	2	9881	30
RF::tdc	21700	12	24	17023	216
RICH::hits	22000	21	3	59474	78
RICH::tdc	21800	12	6	17247	54
RUN::config	10000	11	1	22482	38
RUN::rf	10000	12	2	59560	12
RUN::trigger	10000	13	3	22528	24
TimeBasedTrkg::TBClusters	20600	32	12	66447	660
TimeBasedTrkg::TBCovMat	20600	37	2	44207	204
TimeBasedTrkg::TBCrosses	20600	35	6	68095	348
TimeBasedTrkg::TBHits	20600	31	72	60607	5832
TimeBasedTrkg::TBSegments	20600	33	12	67115	972
TimeBasedTrkg::TBTracks	20600	36	2	68451	236
TimeBasedTrkg::Trajectory	20600	53	44	68695	1628



- X-axis, time = TDC - TProp - TFlight - TStart – tBeta
- (run 5300).
- The plot contains the 42 (7 slots * 6 cables) histograms for sector 1 and superlayer 1. Fits are using a sigmoid function

(courtesy: Shirsendu)