New STAR tracker(Stv)

V. Perevoztchikov
Brookhaven National Laboratory,USA

STAR

Why new tracker?

STAR has used ITTF (Sti) as its main tracker for over 6 years, enabling a successful physics program. However, there are important limitations which constrain us moving forward:

- Geometry Description is too simple and not detailed;
- Geometry ordering does not fit well, even for TPC, and will be nearly impossible for new detectors
- ITTF requires all geometrical objects to be oriented along the z-axis, which precludes forward tracking detectors such as the FGT;
- Complicated task to integrate a new detector
- Magnetic field must be constant and oriented along z

Stv tracker. Main features.

Implemented:

- ROOT/TGeo geometry package is used. Ensures identical geometry in simulation and reconstruction;
- GEANT is used to propagate tracks through material;
- Detector orientation is arbitrary;
- Kalman fit is modified to be more precise for forward detectors;
- Support for running multiple seed finders;

Planned

- Track extentions to other detectors, e.g. BEMC;
- Arbitrary magnetic field;

Stv tracker, additional features.

- Automatic recognition of detector orientation;
- Automatic hits <-> detector elements relationship. No hand made lookup tables;
- All additional information, like detector orientation, hits collections, error functors, activity status etc… contains in TGeo extension classes;
- In a result, flexibility to add and modify new detectors. Practically only Geometry description is needed;

Integrating a new Detector with Stv

Stv was designed to streamline the process of integrating a new detector into the tracker. In order to integrate a new detector, developers need to:

- Define the detector geometry in AgML
  http://drupal.star.bnl.gov/STAR/comp/simu/geometry0

- Create the detector's hit class and hit containers in StEvent.
  - If hits are space points, done.
  - Otherwise, space points should be provided *or* a custom seed finder will need to be implemented.

- Implement an iterator over the hit container in StEventUtilities

- Provide hit errors, either through the hit class or through StvHitErrorCalculator

Stv Seed Finders

By design Stv uses several seed finders. The first one is the default seed finder and others specialized ones.

- Stv allows list of seed finders called one after another;
- Each seed finder could be called several times;
- At the end default seed finder is called repetitively up to no more tracks is founded.

Three seed finders right now are implemented.

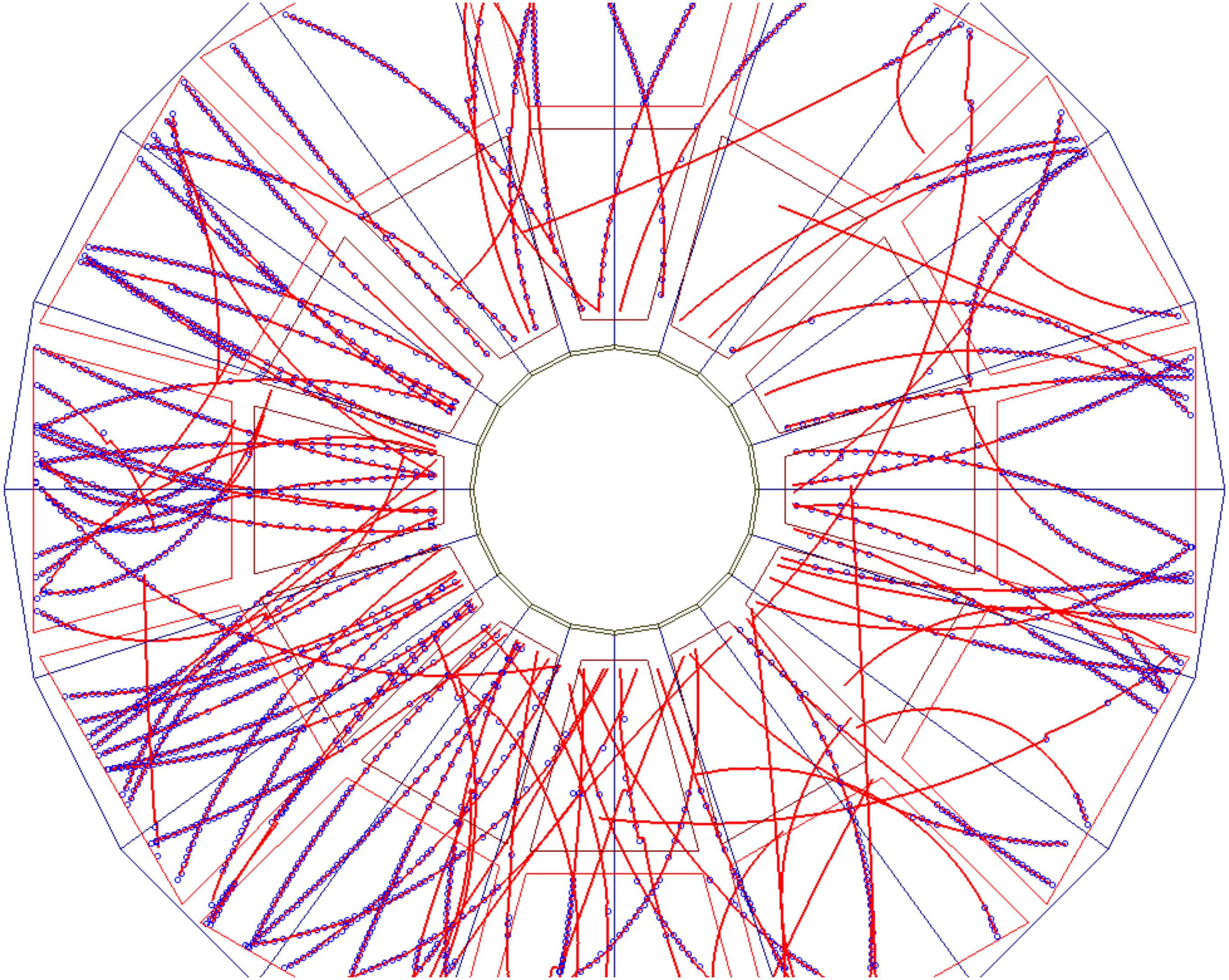
Modified Kalman fit

Stv used modified Kalman fit. More precise, not a Kalman fit was modified but the system of coordinates is different than usual.

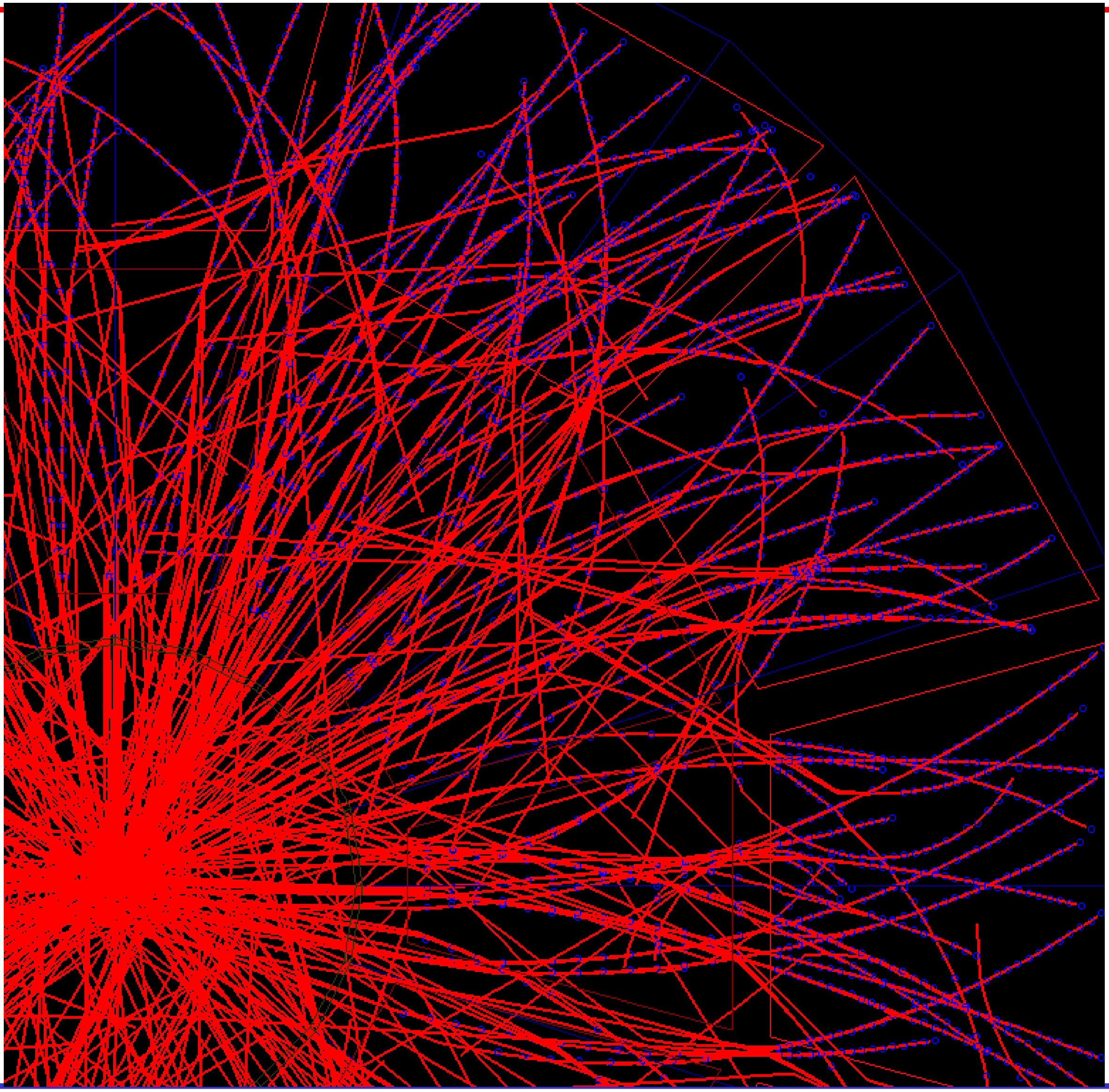
Traditionally the detector plane is used for the fit. But when track is crossing this plane with an angle far from orthogonal, the result is unstable and inaccurate. In Sti this angle < 1 radian;

In Stv we use the DCA plane. This plane is orthogonal to the track and crosses the hit. Hit errors are projected into this plane; Due to orthoganality, accuracy is better for forward tracks.

Just illustration that Stv is working

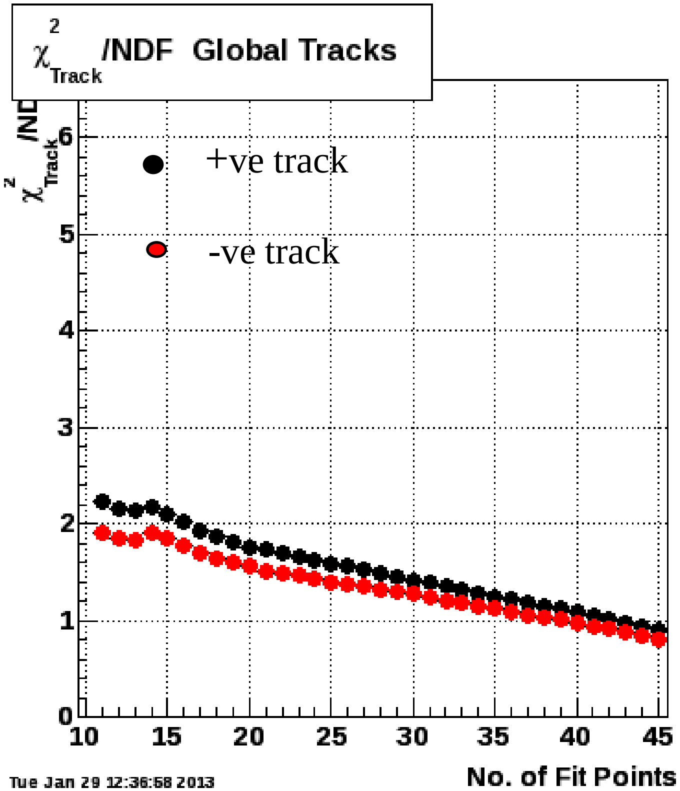


Another one

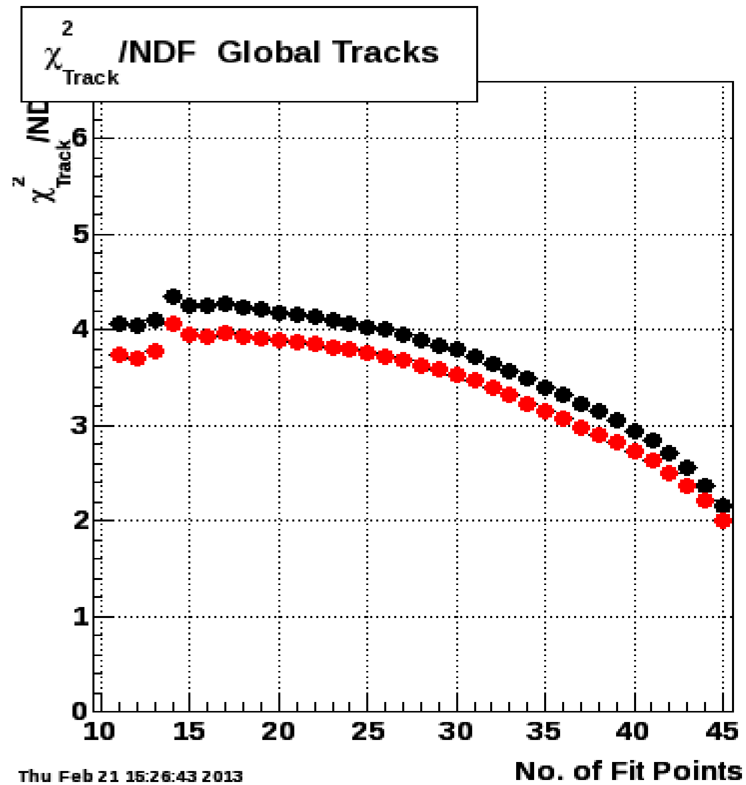


Xi2(nHits) comparison

Sti13



Stv13

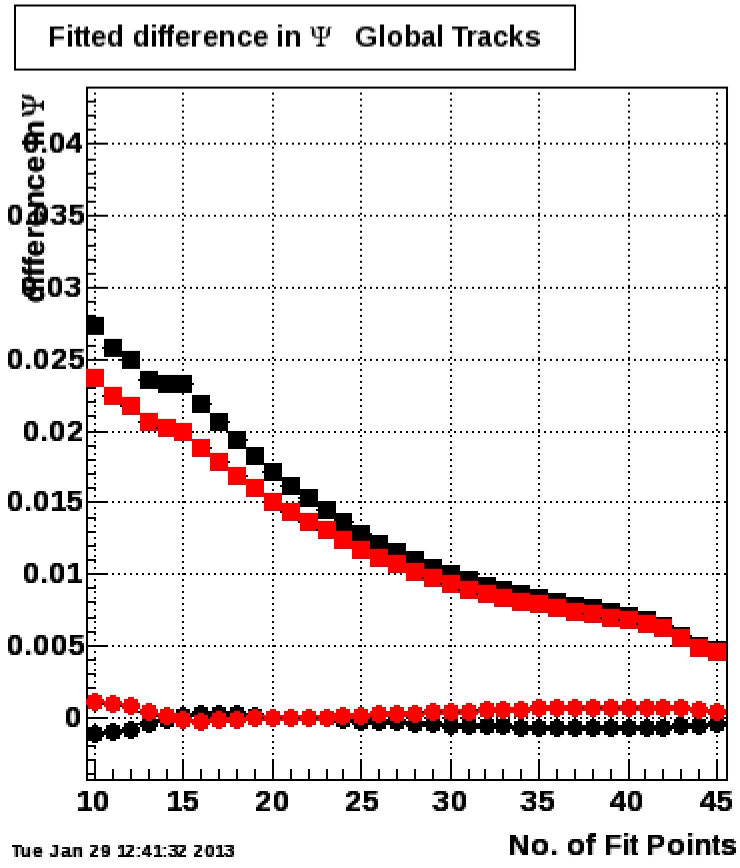


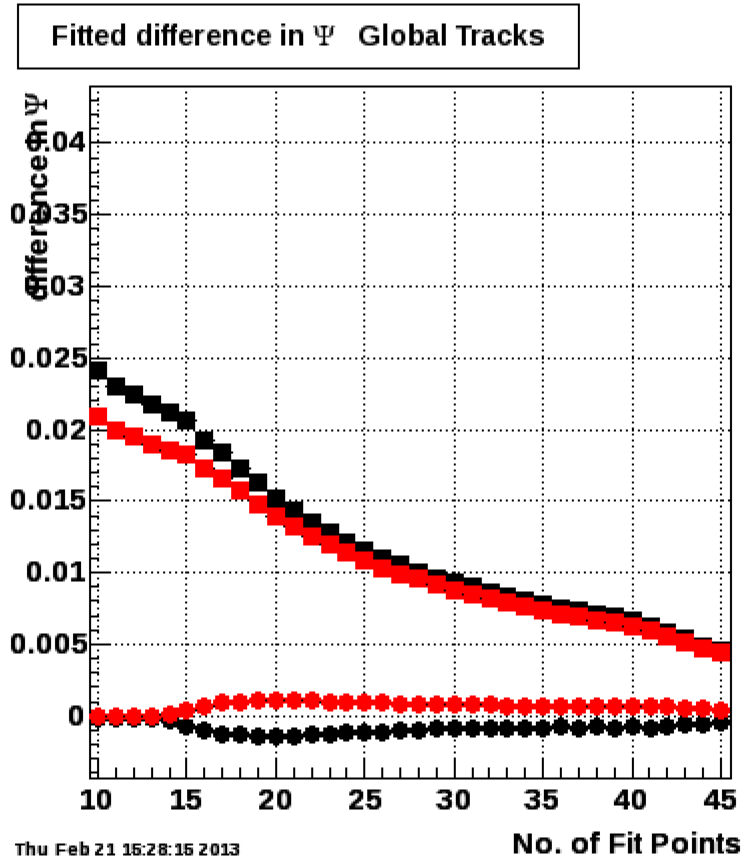
<Xi2> must be 1. Stv Errors 1.4 times less than needed

Track azimuth angle Psi(nHits) deviation comparison

Sti13

Stv13

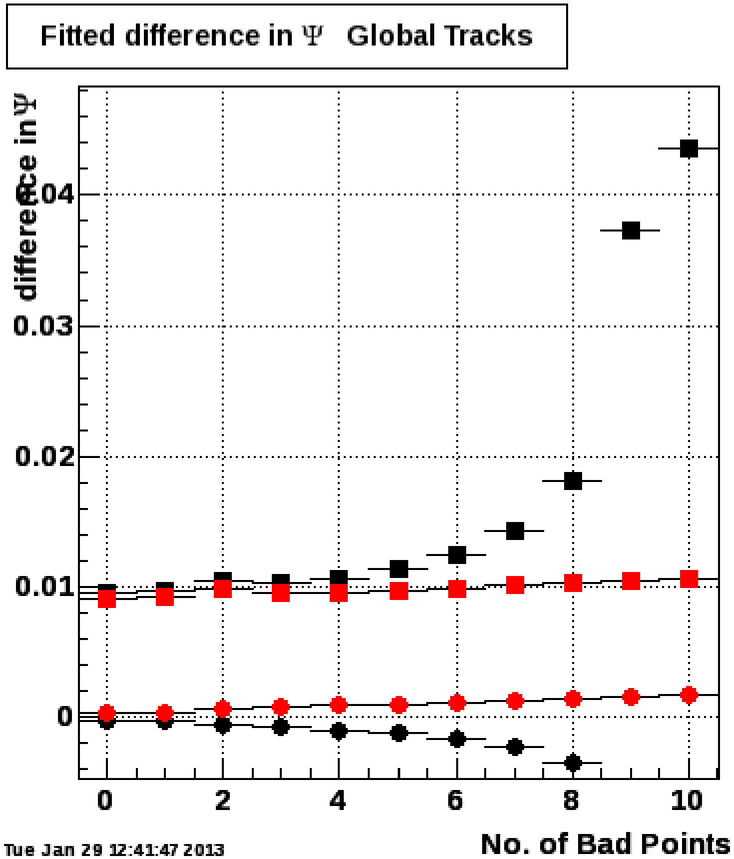




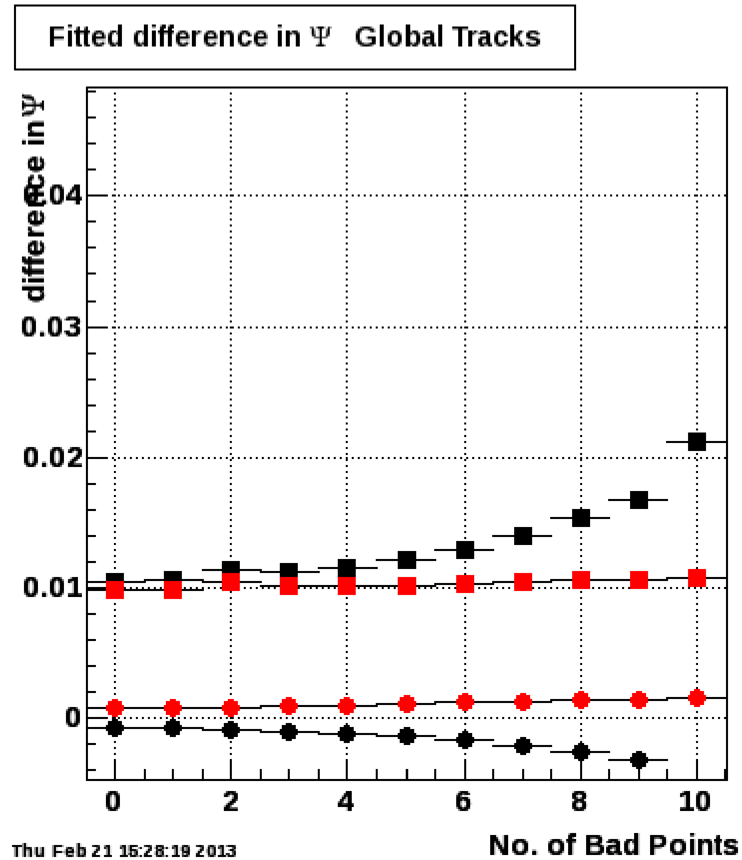
Stv and Sti comparable angle

Track azimuth angle Psi(nBadHits) deviation comparison

Sti13



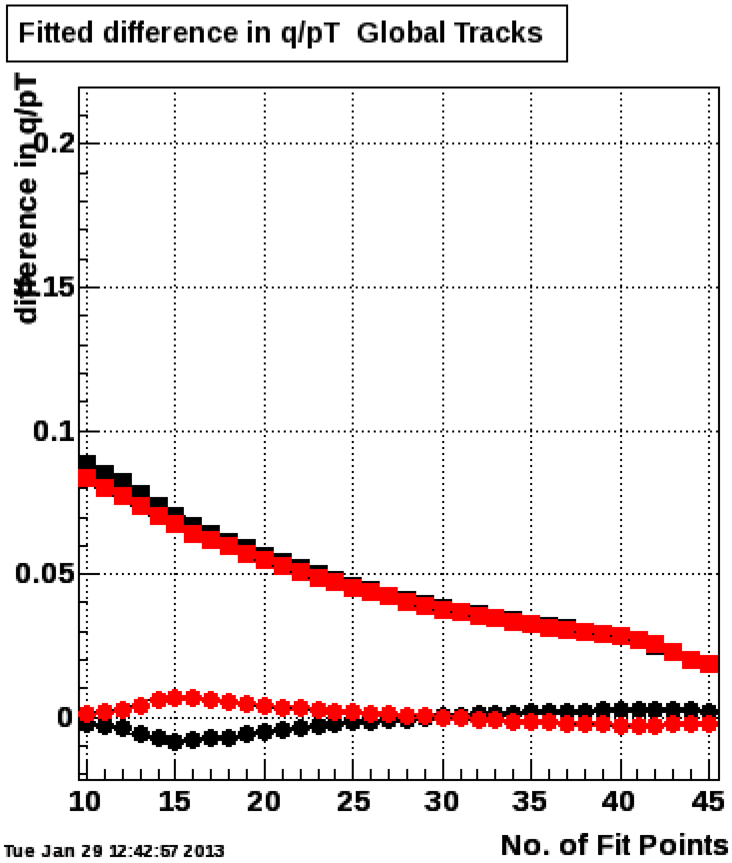
Stv13



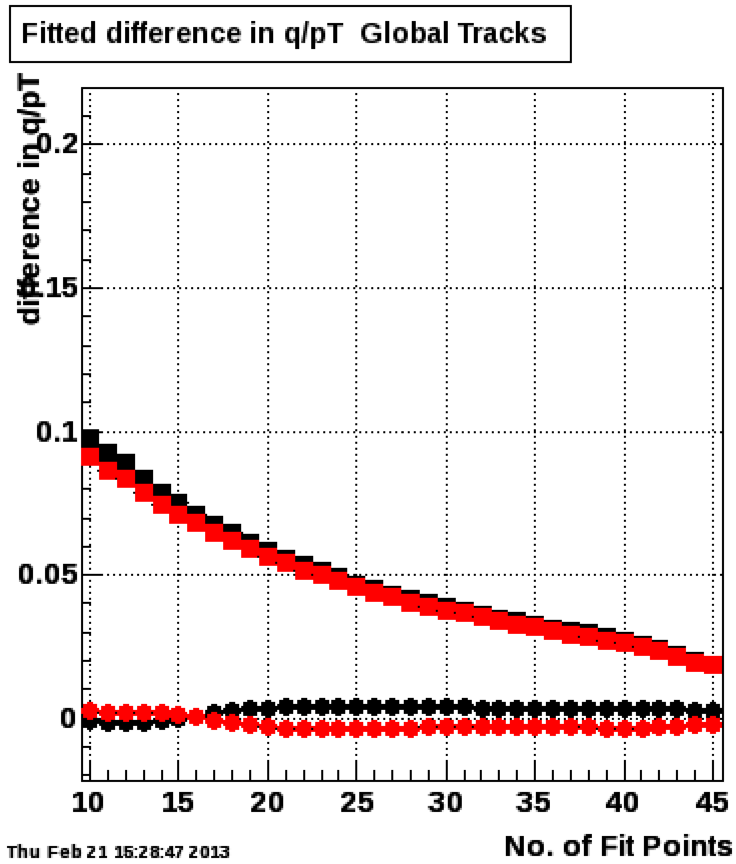
Stv appears more stable with respect to bad points

.q/pt deviation(nHits)

Sti13



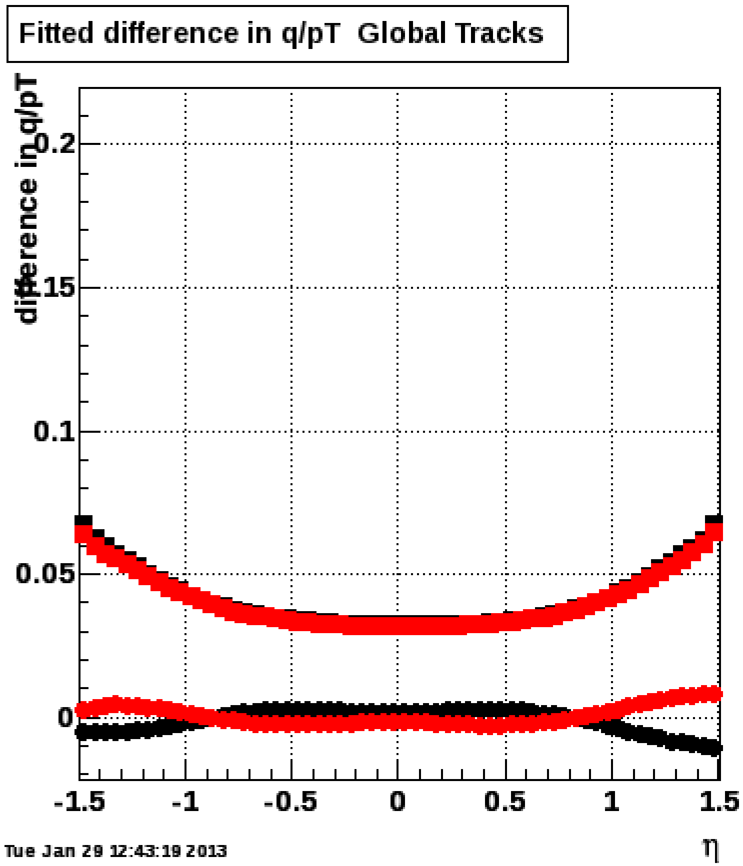
Stv13



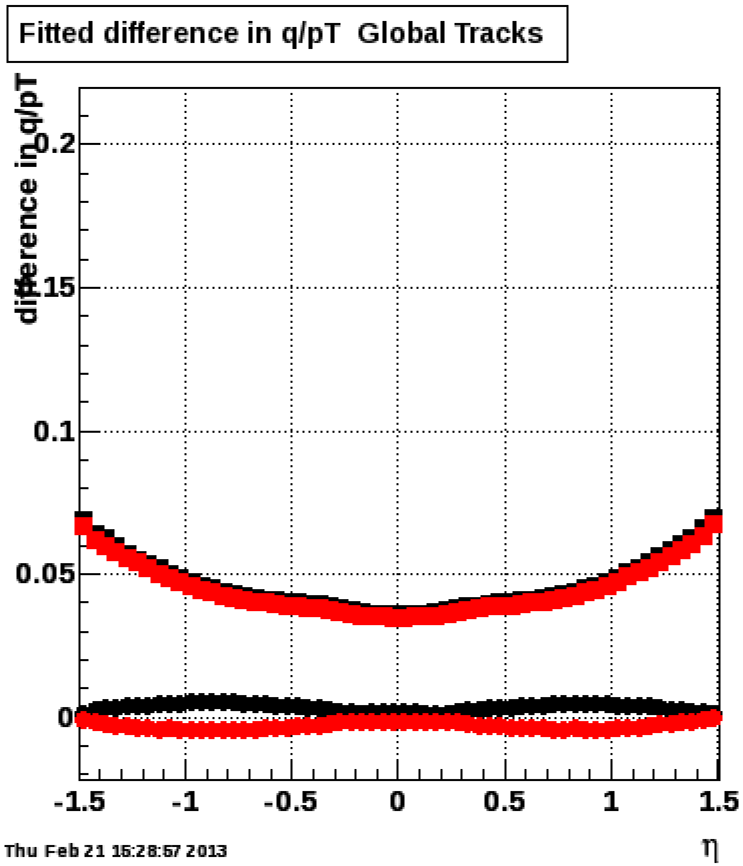
Stv has comparable charge-sign discrimination to Sti

q/pt deviation(Eta)

Sti13



Stv13

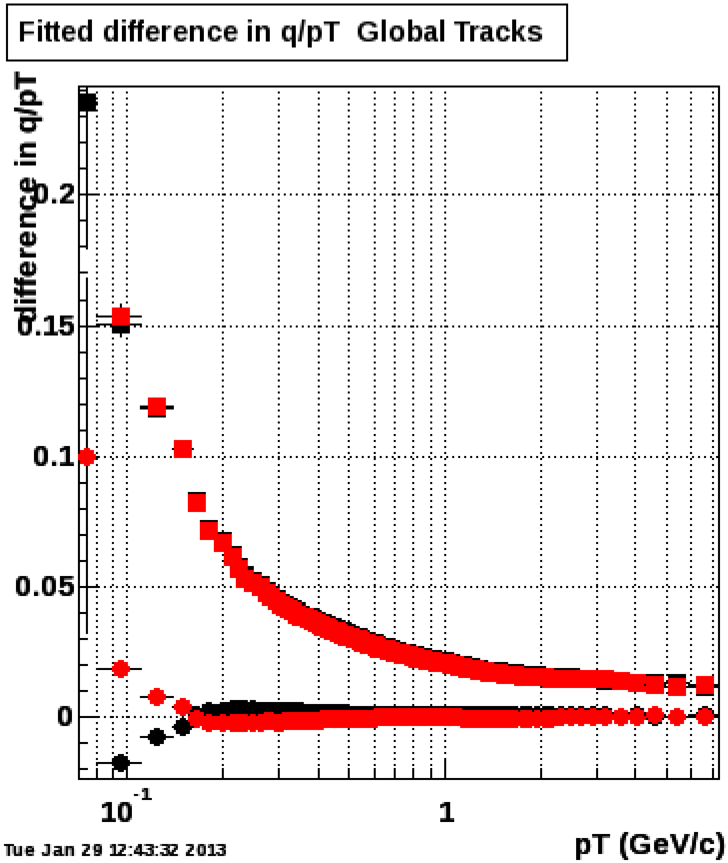


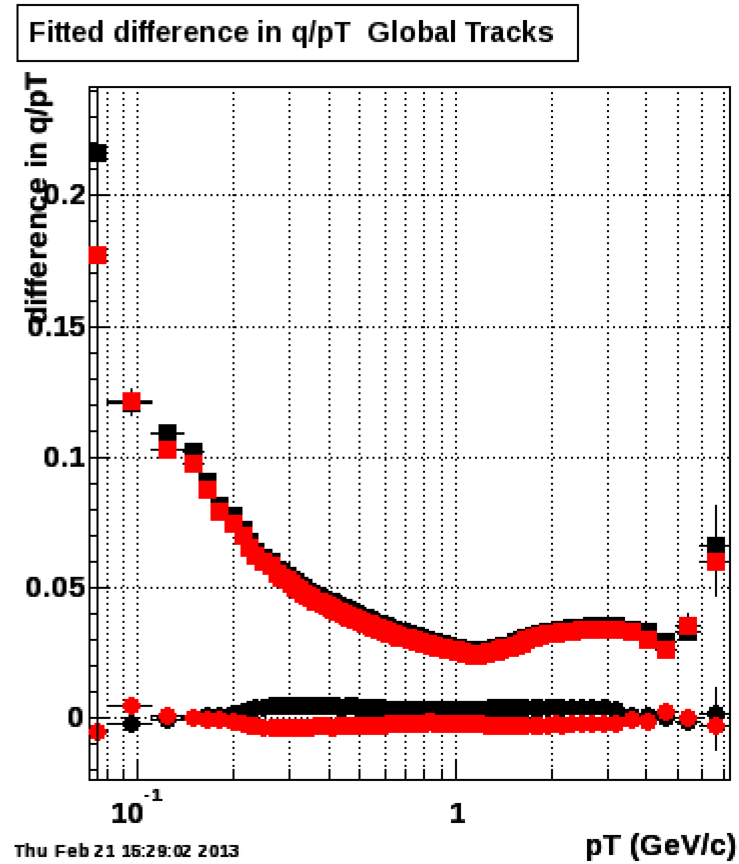
Charge discrimination as function of Eta comparable

STAR

q/pt deviation(pT)

Sti13 | Stv13

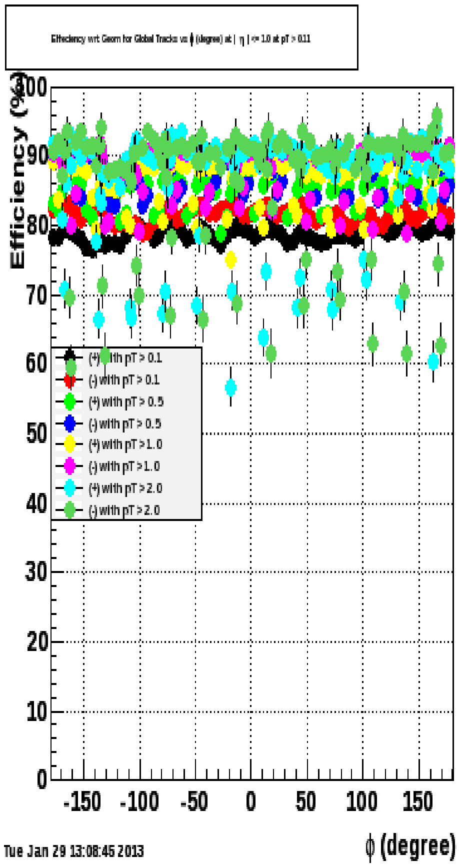




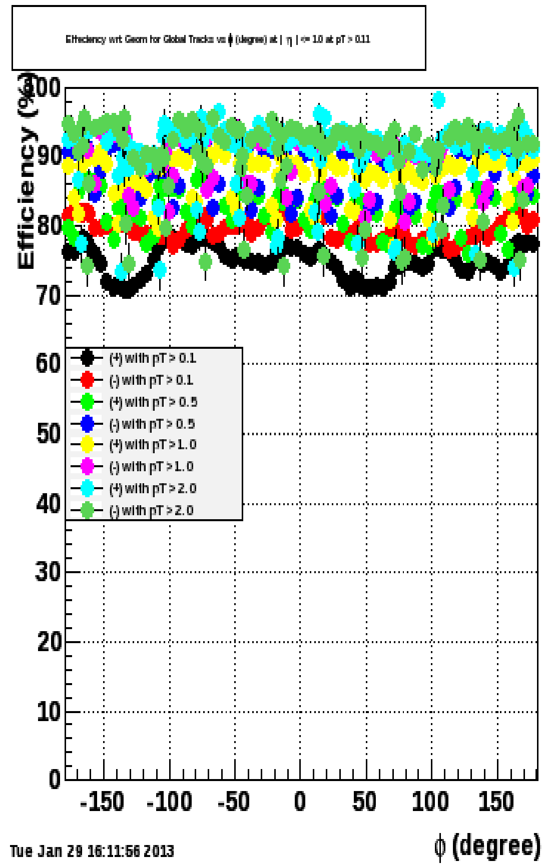
Stv has a problem in high pT, Sti in low pT

Efficiency(Phi)

Sti13



Stv13-

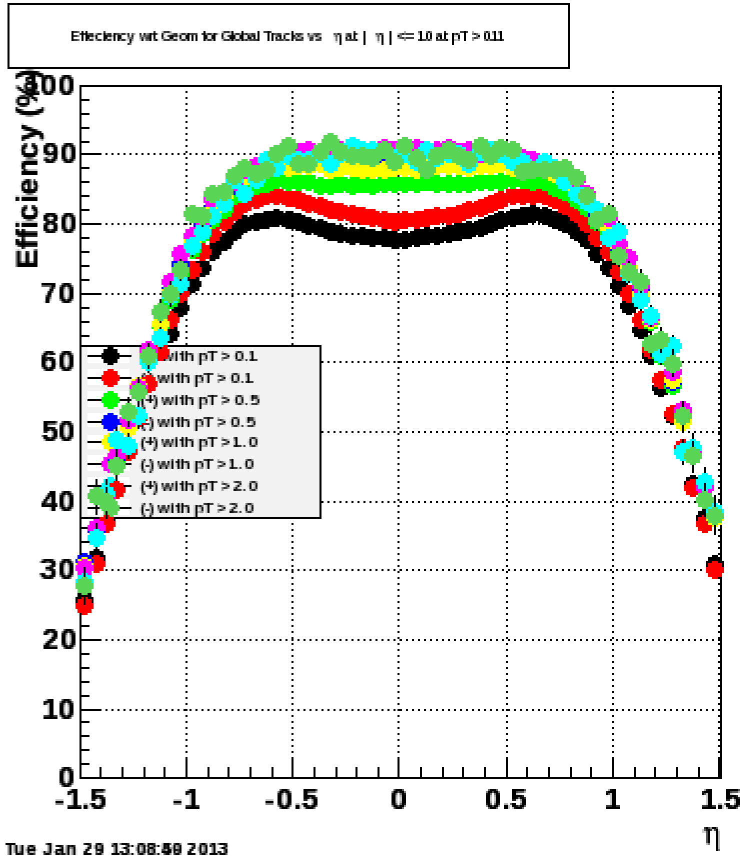


- +pt> 0.1
- - pt> 0.1
- +pt> 0.5
- -pt> 0.5
- +pt> 1.0
- -pt> 1.0
- +pt> 2.0
- -pt> 2.0

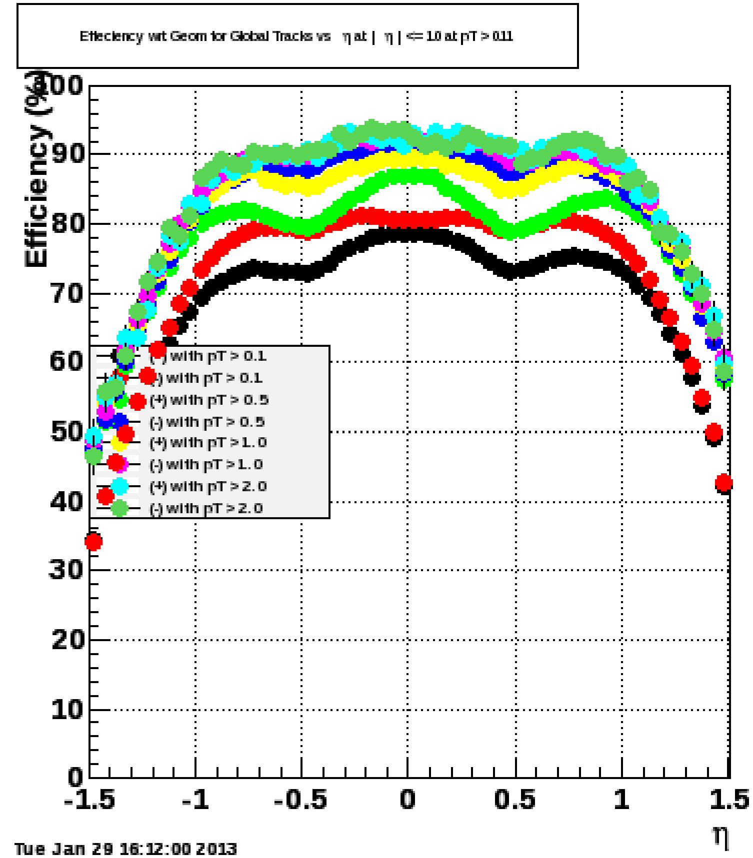
Sti has outsiders, Stv not. But Sti~80 Stv~75. Sti better

Efficiency(Eta)

Sti13



Stv13-



Sti in average better, but high pT in Stv is better.

Something wrong

You noticed that in previous slides was typed Stv-. Why (-)?

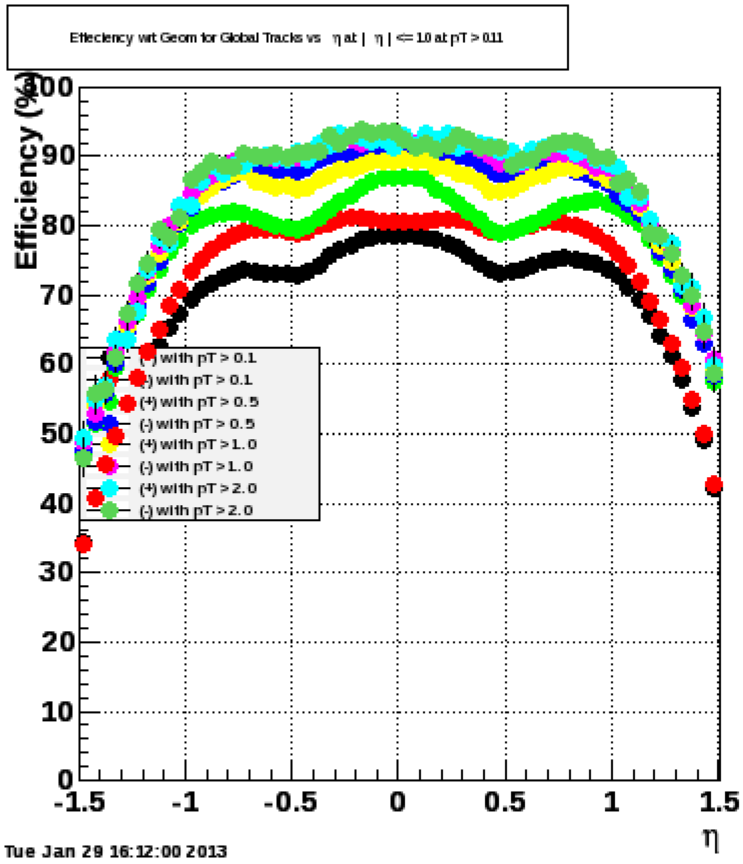
Three weeks ago some modifications was made, mostly result become better, but efficiency…

List of modifications:

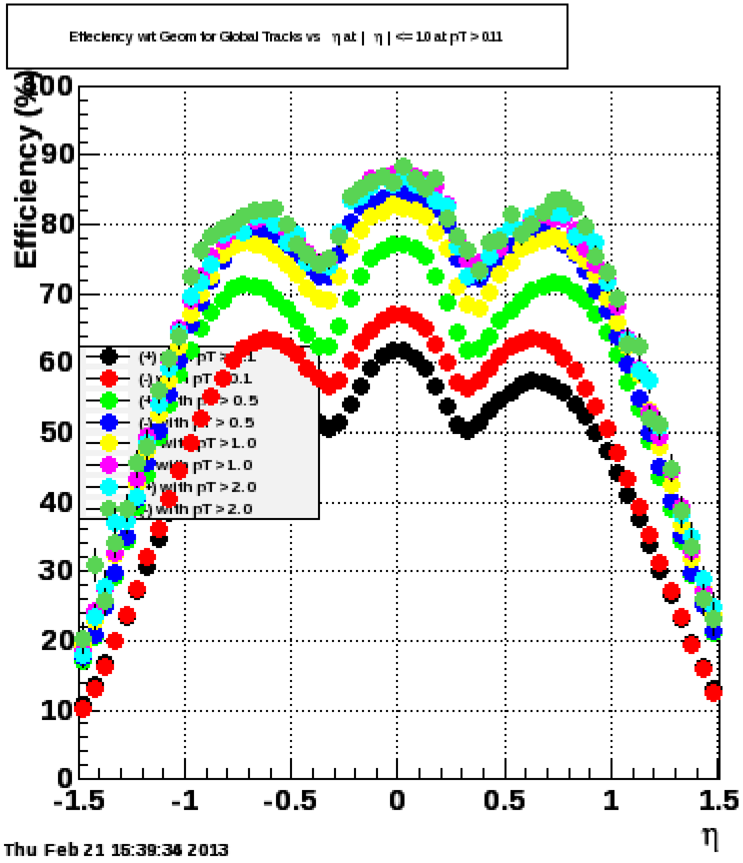
- Kalman Track initialization;
- Hit errors refitted;
- …

Efficiency Stv 3 weeks ago and current one

Stv13-

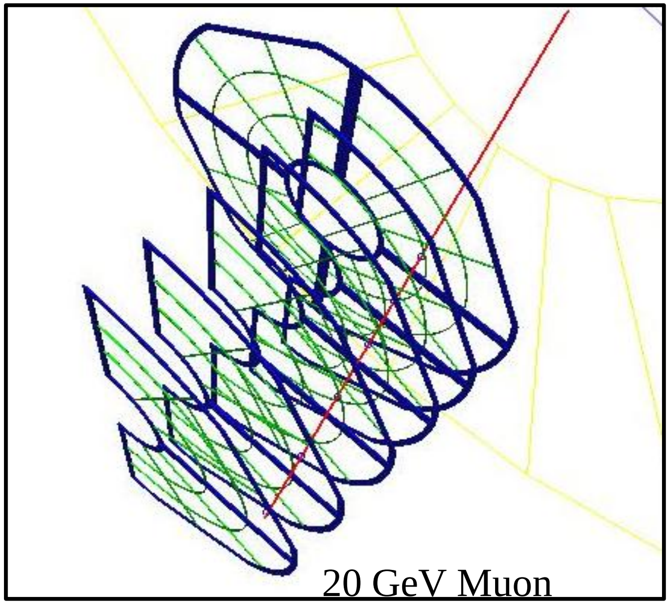


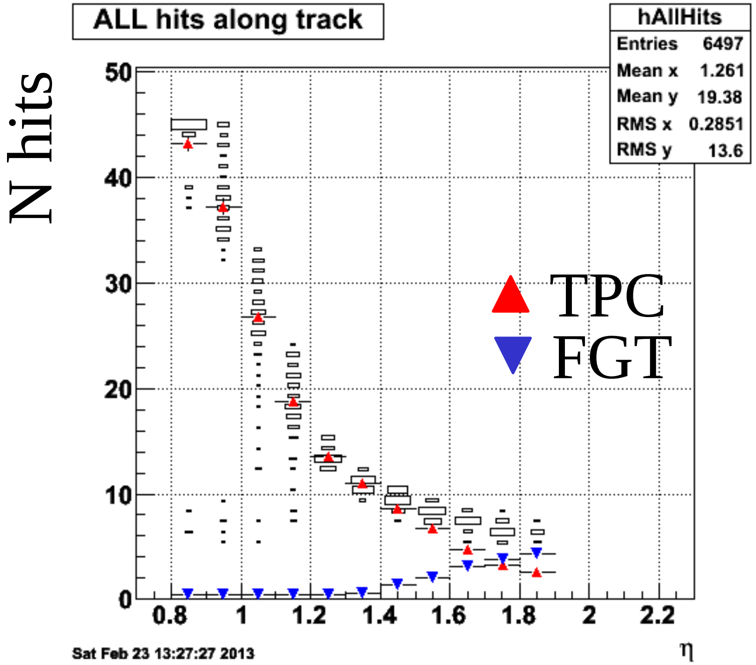
Stv13



Something happened. Current Stv efficiency mach worse.

Proof of Principle FGT Tracking with Stv



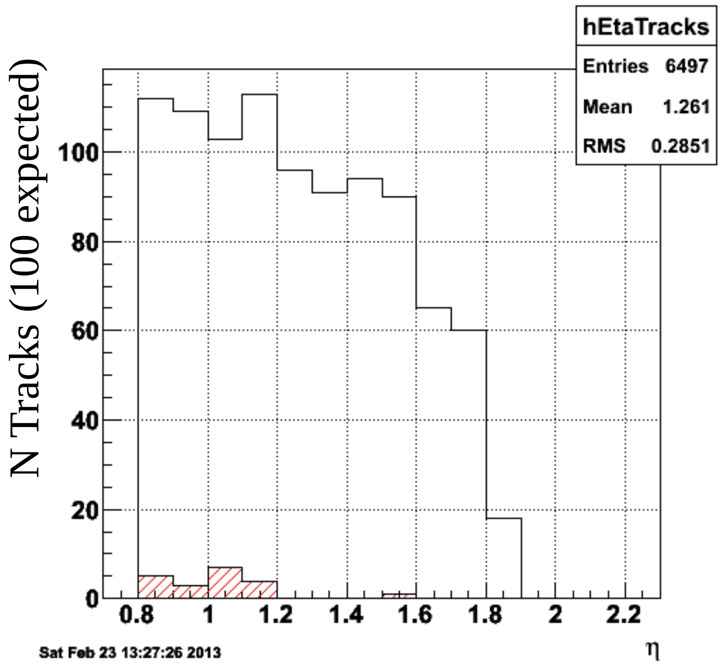


Stv is capable of tracking with the FGT in the forward direction, reconstructing tracks with hits in both TPC and FGT.

- MC sample: 1 mu- with pT > 15 GeV in FGT acceptance on each event
- FGT clusters are combined into space points
- Space points passed to customized seed finder
- Seeds are extended and fit as tracks using Stv
- Default seed finder was used

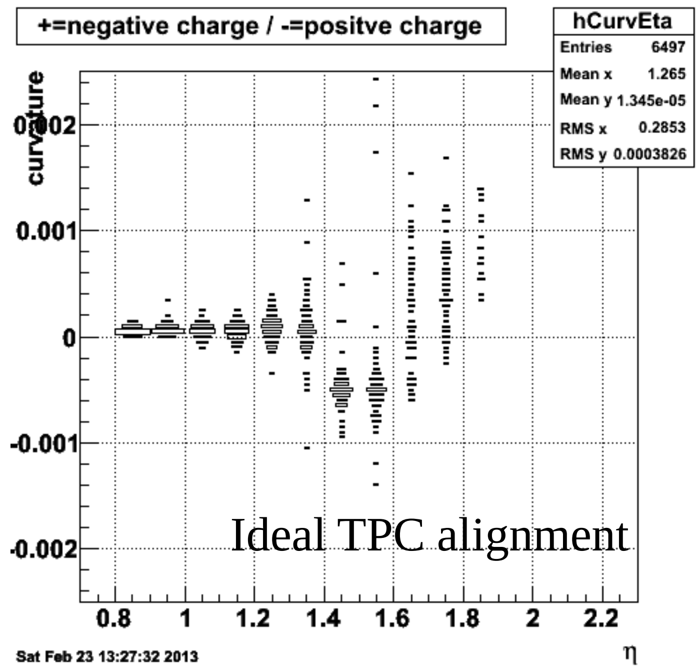
Efficiency and Charge Sign

The good news…



Reconstruction efficiency for *global* tracks is reasonable out to eta = 1.6. Clone tracks in red.

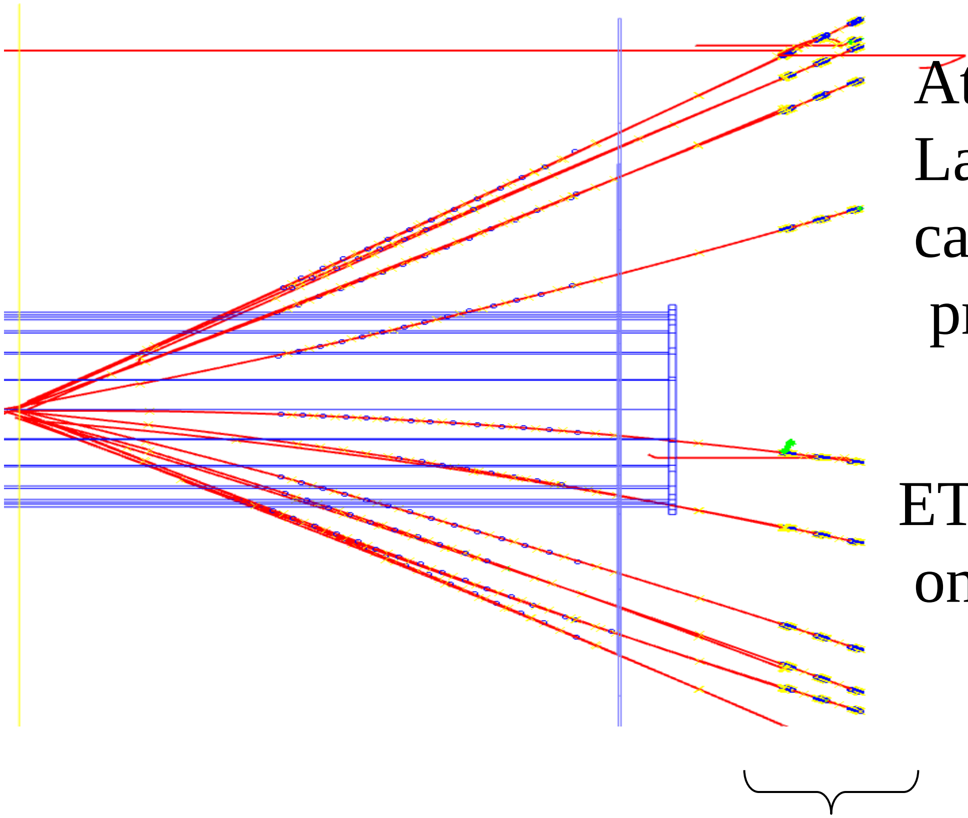
The bad news…



Charge is pulled to the wrong value in the region where tracks meet the FGT acceptance.

The take away: more work needs to be done to understand why tracking falls off at eta of 1.6, and why charge sign gets pulled in wrong direction. FGT expertise is needed here.

Forward Tracking in the ETTR



At the STAR upgrades workshop Last year, Stv was shown to be capable of forward tracking in the proposed ETTR upgrade detector.

ETTR provides 3 x 30 hit points on the track.

ETTR

Stv tracker status

What is implemented:

- Track propagator based on Geant geometry and Geant propagator;
- Arbitrary orientation of sensitive detectors;
- Forward oriented Kalman fit and filter;
- Multiple Seed Finders;
- Automatic recognition of detector orientatation and types;
- Automatic relationship between hits and sensitive detectors, without hand make lookup tables;

Stv status continue

What was tested and tuned:

- Stv & Sti give mostly comparable results;
- Two forward detectors was introduced and successfully tested. Some additional tuning is needed;

What improvement is needed :

- Tpc Hit errors is still too small. Reason is probably understood. New results expected in a week;
- Efficiency is about 5% less than in Sti. Probably related to underestimated hit errors.
- Performance was not yet investigated;
- Expected time scale for tuning is 2-3 weeks with the help of workforce team assigned;
- Common work with FGT group is expected.

The End

Arbitrary magnetic field orientation

We use Geant as a tracking engine. Geant allows tracking in arbitrary magnetic field. But tracking only is not enough. To do fitting the error propagation is also needed. Right now error propagation is implemented for Z oriented magnetic field only. But Stv design allows such implementation with arbitrary magnetic field.

Stv Seed Finders continue

Non standard seed finders are needed in two cases:

1. Increase performance for concrete detector;
2. Non standard detector, for which default seed finder does not work;

The typical example of point 2. is Fgt. Fgt hits are elongated.

Some hits define only Z and Rxy, others Z and Phi. In addition, there are amplitudes which could be used as well.

Right now we have:

- Default, :following nose", seed finder, for any detector ;
- KNN (K-Nearest Neighbor) seed finder, for any detector ;
- CA seed finder oriented for TPC only;
- 2d hit generic seed finder in development

2D hit Seed finder.

It is based on the following ideas:

- All 2d hits are joined into 3D hits, with a many fake ones. Amplitudes could be used to decrease amount of fake hits.
- 1st hit selected. Its coordinates considered as zero.
  - All hits in neighborhood projected onto 2d plane, using their φ,λ angles. Real track on this plane represented by spot.
  - Using well known "K neighbor" method we find the spot.
  - If there is no spot, the first hit is fake. Select the next.
  - When spot is found, the seed is created from all related hits

This is not yet ready, only part of code is written.

What is Kalman

The simple example. We have 4 measurements of the same parameter .

To estimate the value we search the minimum of:

$$\chi^2 = \frac{(x_1 - m)^2}{\sigma_1^{\,2}} + \frac{(x_2 - m)^2}{\sigma_2^{\,2}} + \frac{(x_3 - m)^2}{\sigma_3^{\,2}} + \frac{(x_4 - m)^2}{\sigma_4^{\,2}}$$

The solution is trivial, this is a "***global fit***":

$$m = \frac{\frac{x_1}{\sigma_1^{\,2}} + \frac{x_2}{\sigma_2^{\,2}} + \frac{x_3}{\sigma_3^{\,2}} + \frac{x_4}{\sigma_4^{\,2}}}{\frac{1}{\sigma_1^{\,2}} + \frac{1}{\sigma_2^{\,2}} + \frac{1}{\sigma_3^{\,2}} + \frac{1}{\sigma_4^{\,2}}}$$

Kalman continue

It was a ***global*** solution. The ***Kalman fit*** is:

$$m_1 == x_1$$

$$\sigma_{m1} = \sigma_1$$

$$m_2 = \frac{\dfrac{m_1}{\sigma_{m1}^2} + \dfrac{x_2}{\sigma_2^2}}{\dfrac{1}{\sigma_{m1}^2} + \dfrac{1}{\sigma_2^2}}$$

$$\frac{1}{\sigma_{m2}^2} = \frac{1}{\sigma_{m1}^2} + \frac{1}{\sigma_2^2}$$

Kalman continue

Kalman fit features:

- The result of ***Kalman fit*** is exactly the same as for a ***global*** one. No mystics;
- In global fit, size of matrix to invert, depends on number of measurements. In Kalman fit not. As a result, machine accuracy for Kalman fit are more critical;
- In Kalman fit an estimation is updated with each new measurement. This allows to make a decision: is this new measurement belong to our object, or not? So we can use not only ***Kalman fit*** but ***Kalman filter***. Not possible in ***global;***
- By the points above, Kalman fit and filter are so popular.

Modified Kalman fit

Stv used modified Kalman fit. More precise, not a Kalman fit was modified but the system of coordinates is different than usual.

Standard Kalman fit:

Let consider simplified, 2d case. Hitting plane along Y axis, track crossed Y axis with angle α wrt X axis. So :

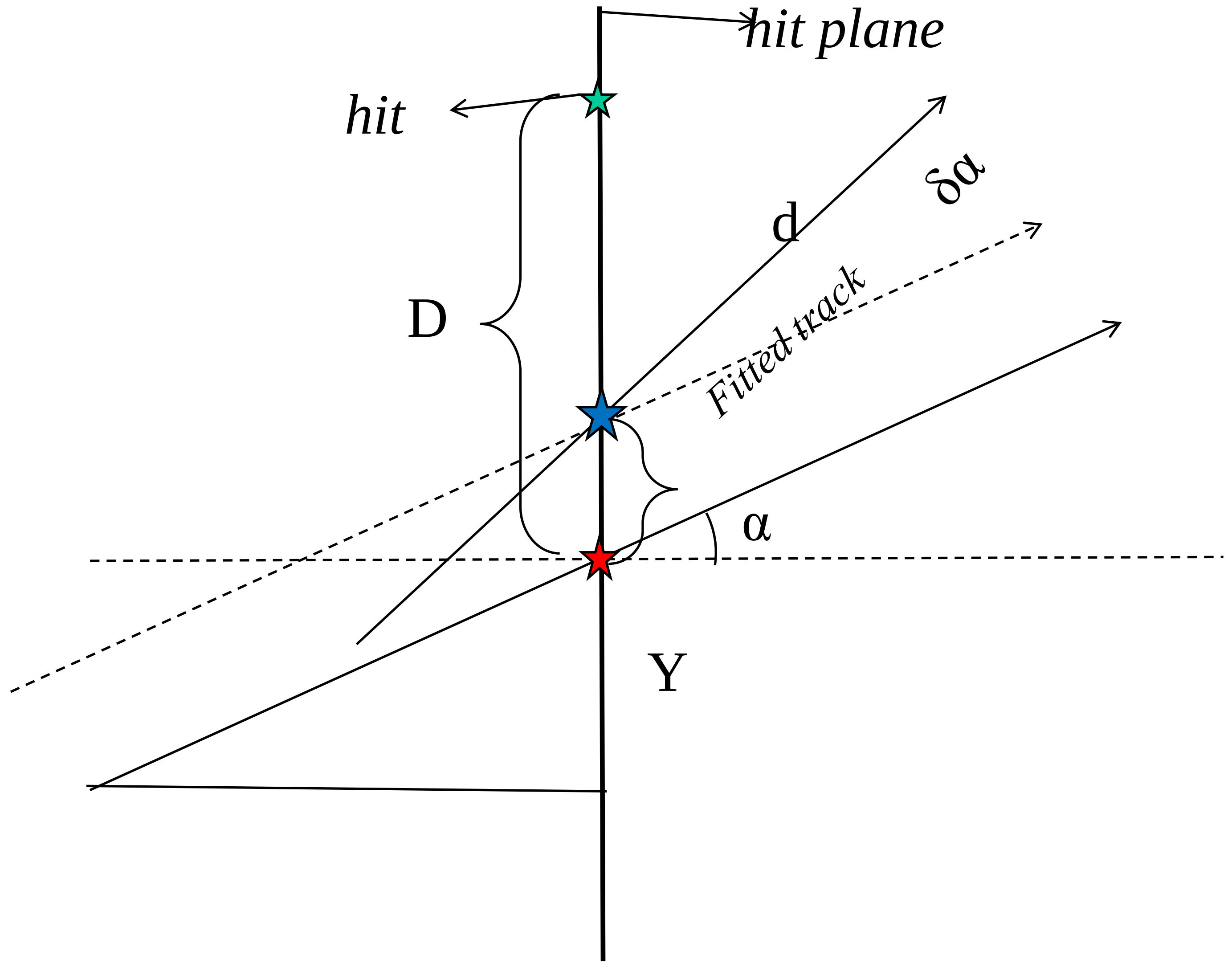
- global track parameter α projected into local Y and proportional to tan(α).
- then δY ~ δ α *(1+ tan(α) * (δα) /2)/
- when α << 1, then cos(α) =1, tan(α) = α, second term is very small and projection from global to local system is linear.
- Projections of error matrix is also linear.

In local frame fit is linear. Transformation to global of fitted parameters and errors is linear too. So life is good.

But when α >1 , cos(α) ~0, life is bad. Linearization is wrong, linear fit is wrong,

Backward transformation into global is also wrong. All times, when I saw unstable fit in Sti, it was α >1

Detector Frame



Fit continue

Could we do something with bad fit when $\alpha > 1$?

Yes, we can!

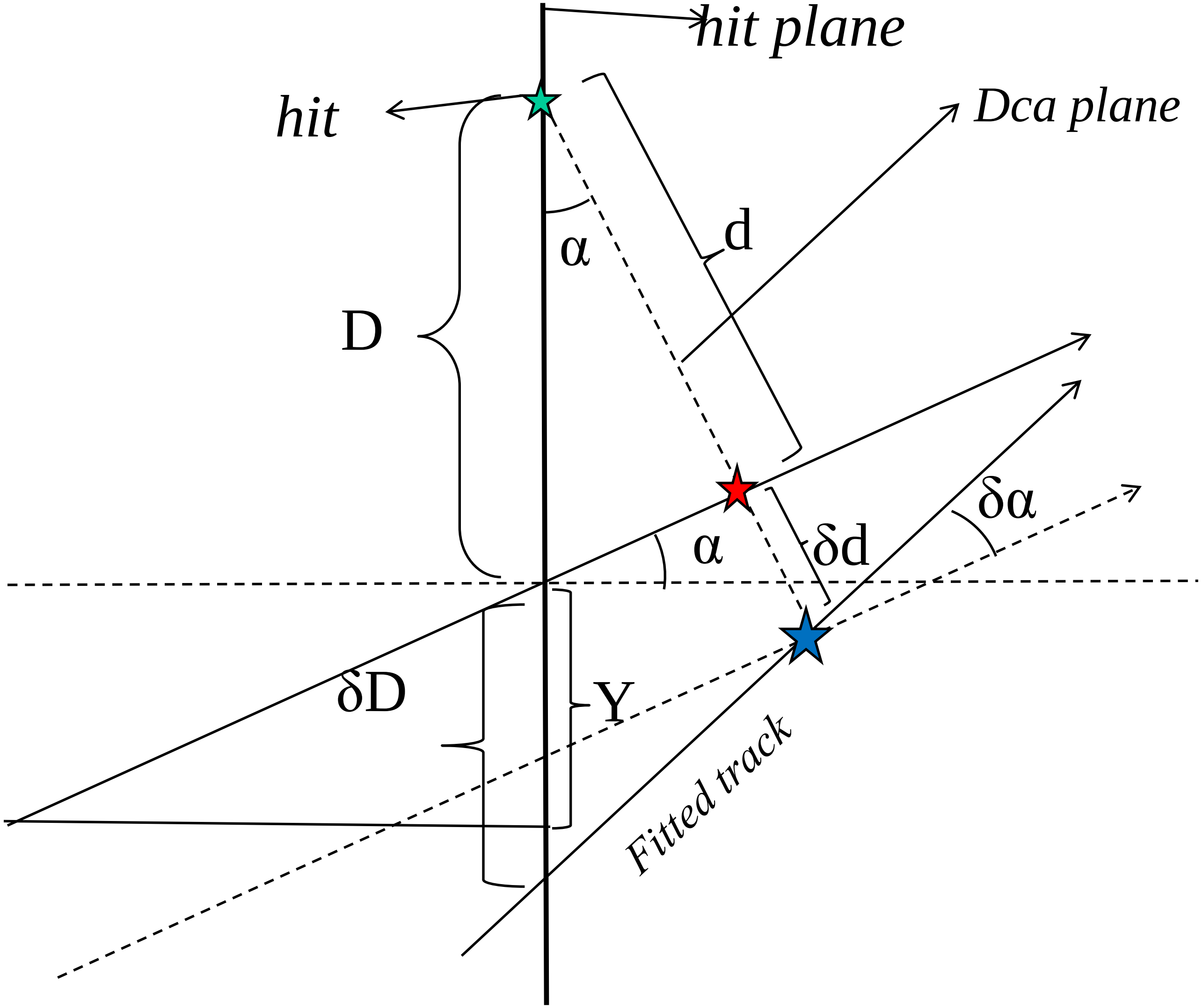
Why we fit in local frame? There are only two reasons:

- We know that track is crossing our hit plane;
- We know the errors in this frame;

Let invent another local frame, where linearization is always working .

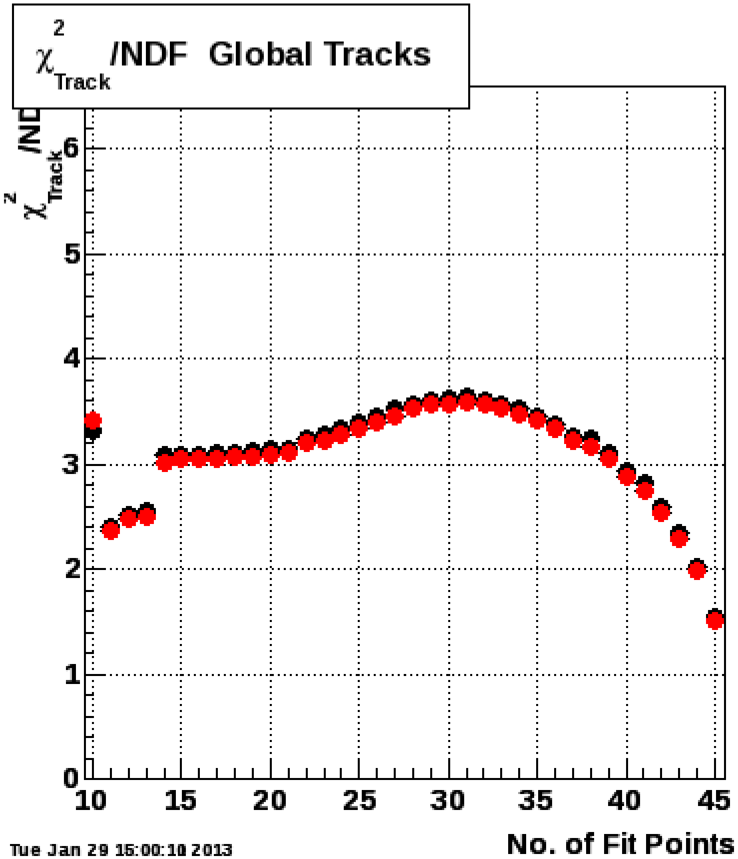
The evident candidate is Dca frame. Dca frame is a track coordinate system where origin is in Dca point to hit. In this frame plane perpendicular to the track and crossing the hit point is a Dca plane. Look the following picture.

Dca Frame

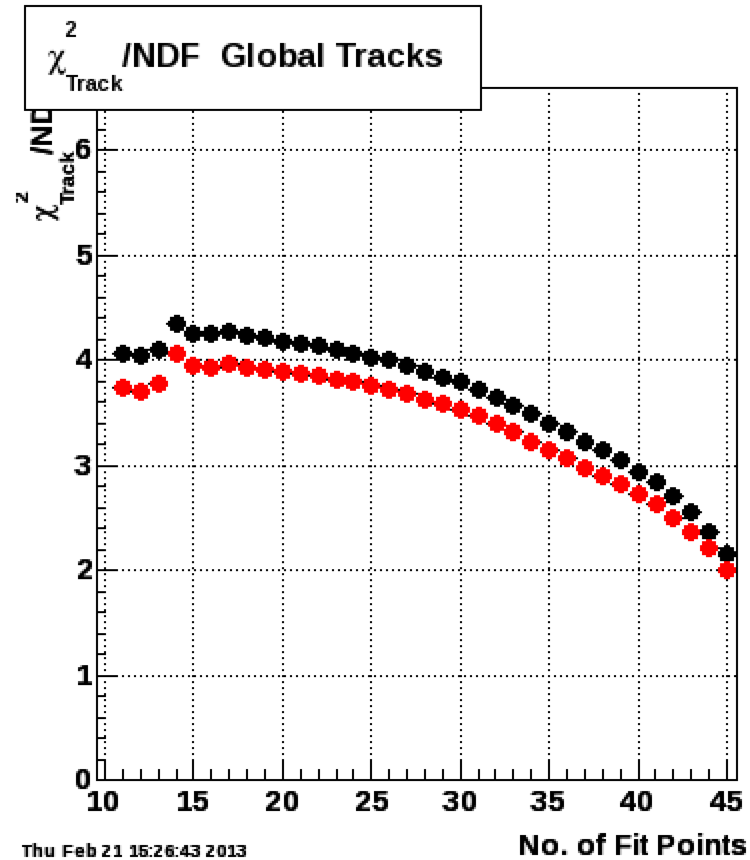


Stv12 & Stv13 comparison.

Stv12



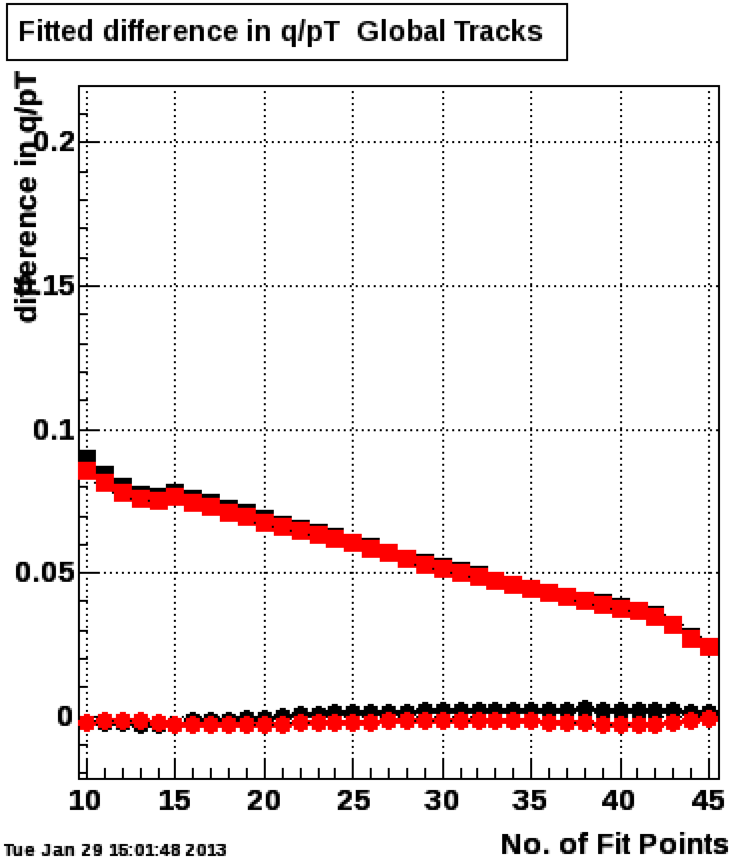
Stv13



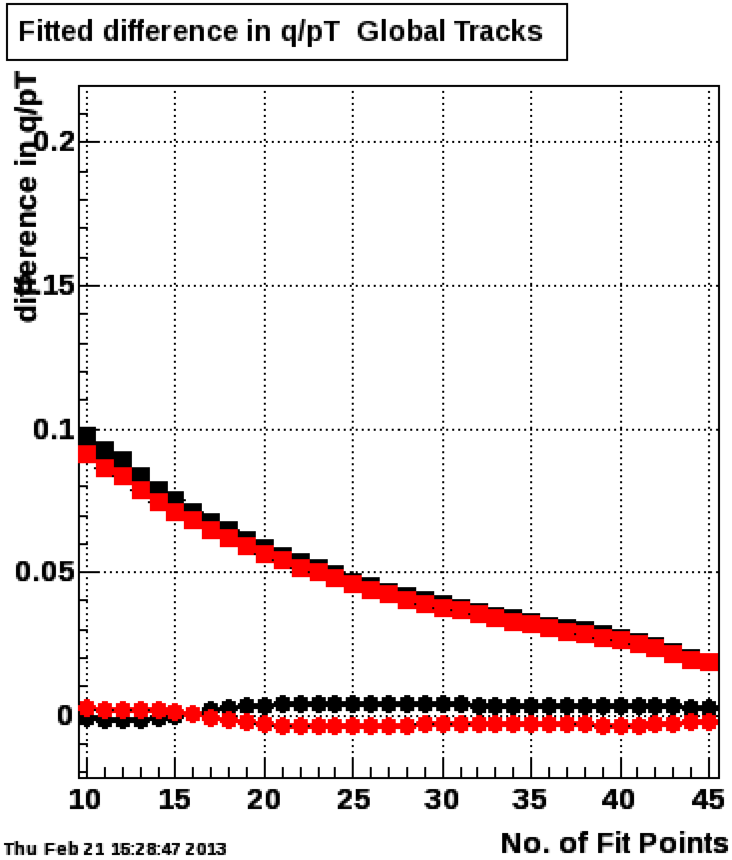
Xi2 still high, but shape now is correct

q/pt(nHits)

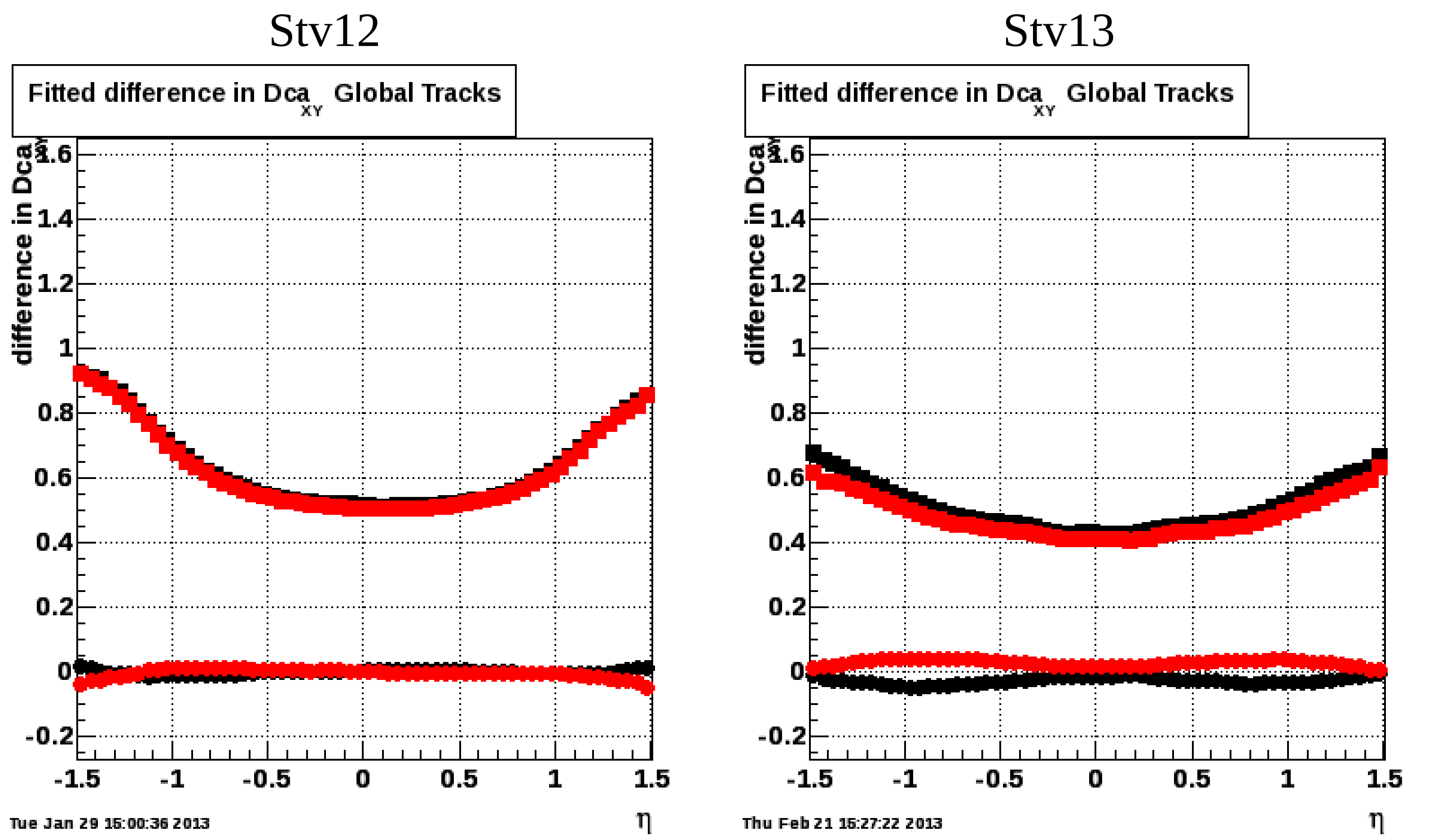
Stv12



Stv13



Dca(xy)


Stv12
Fitted difference in Dca XY Global Tracks
difference in Dca xy
η
Tue Jan 29 15:00:36 2013
Stv13
Fitted difference in Dca XY Global Tracks
difference in Dca xy
η
Thu Feb 21 16:27:22 2013

Summary for Sti/Stv comparison.

- It is clear that hit errors are too small. Factor is about 1.4 . Right now we know the reason. FitHitError application will be modified, and on next week we will see result.
- Efficiency in average is still better in Sti. It could be explained again, by too small estimations of hit errors.
- In other parameters, Sti/Stv are comparable.
- Tuning is still needed, hopefully 3-4 week will be enough.