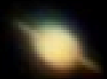
CLAS12 Software Workshop



SOCRAT: a SOftware for Clas12 Reconstruction And Tracking

S. Procureur (CEA-Saclay)

04/25/10

Outline

- Introduction
- The Kalman Filter algorithm
- Central tracking
- Forward tracking
- Miscellaneous (mini-stagger, energy loss)

Introduction

- Goals:

  1) Develop a track reconstruction software for CLAS12, taking into account realistic background (Geant4). This reconstruction is based on the Kalman Filter algorithm

  2) Check that the design reaches the required performance

  3) Optimize the design

- Key issues: charged particle reconstruction in the Forward DC system (~24k sense wires), and the Central Tracker (3x2 Silicon + 3x2 Micromegas)

- Vertex: for accurate vertex reconstruction of forward tracks, e.g. for hyperon and cascades decays, the Forward Vertex Tracker (FVT) is also included

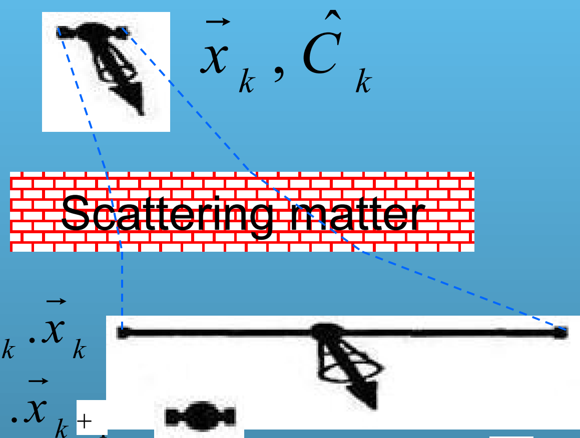
- PID: not yet developed

Kalman Filter algorithm

→ Starting point: a state vector and its covariance matrix

→ Extrapolation to the next measurement:

→ Update of the position (by minimizing the χ^2):



$$\vec{x}_k , \hat{C}_k$$

$$\vec{x}_k^{k+1} = \hat{F}_k . \vec{x}_k \qquad \hat{C}_k^{k+1} = \hat{F}_k . \hat{C}_k . \hat{F}_k^T + \hat{Q}_k$$

$$\vec{m}_{k+1} = \hat{H}_{k+1} . \vec{x}_{k+1}$$

$$\vec{x}_{k+1} = \hat{C}_{k+1} . \left(\left(\hat{C}_k^{k+1}\right)^{-1} . \vec{x}_k^{k+1} + \left(\hat{M}'_{k+1}\right)^{-1} . \vec{m}'_{k+1} \right)$$

$$\hat{C}_{k+1} = \left(\left(\hat{C}_k^{k+1}\right)^{-1} + \left(\hat{M}'_{k+1}\right)^{-1} \right)^{-1}$$

Key issues: → choice of $\vec{x}$ (geometry)

→ initialization of $\vec{x}_0$ and C_0 (helix, torus Bdl param.)

Kalman Filter algorithm

- Central tracking:

  - Measurement at ~constant radius

    $\rightarrow \quad {}^{T} = (\varphi, z, \vec{p})$

  - Homogeneous field (solenoid)

    $\rightarrow$ estimation of $\vec{x}_0$ with helical track

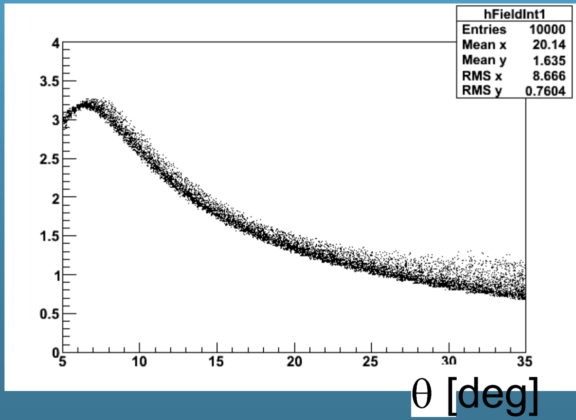
- Forward tracking:

  - Measurement at ~constant z

    $\rightarrow \quad {}^{T} = (x, y, u_x, u_y, q/p)$

  - Toroidal field

    $\rightarrow$ parameterization of $\int Bdl$

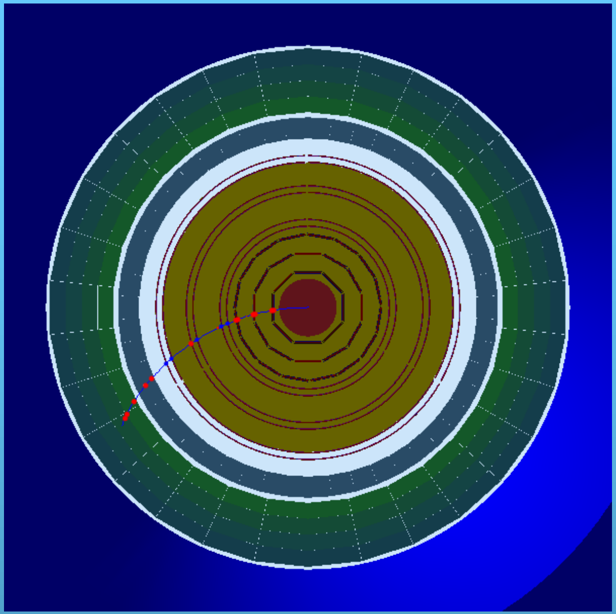


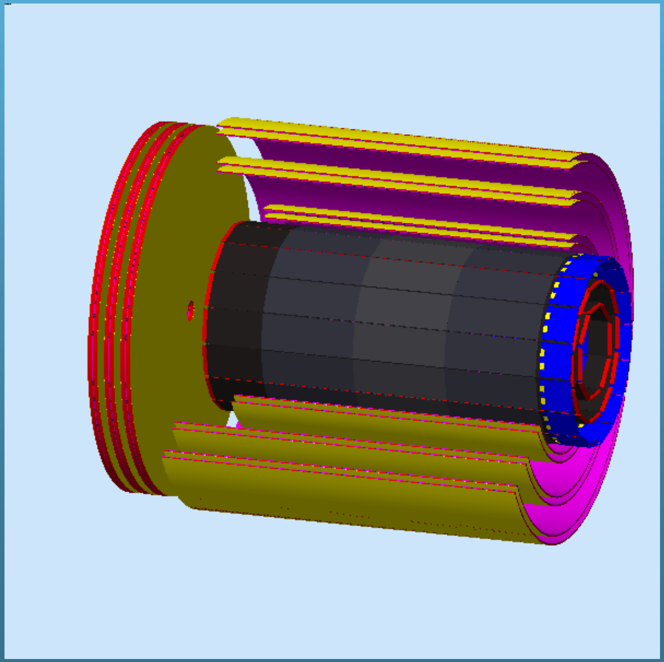
Central tracker (BVT)

Main features (up to now):

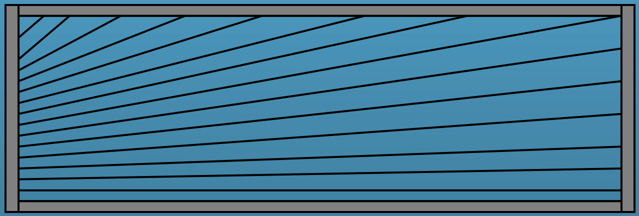
→ 3x2 layers of Silicon
- polygon structure
- graded stereo angle: 0→3°

→ 3x2 layers of Micromegas
- thin, cylindrical detectors
- strip angle: 0 & 90°





Silicon strip layout:



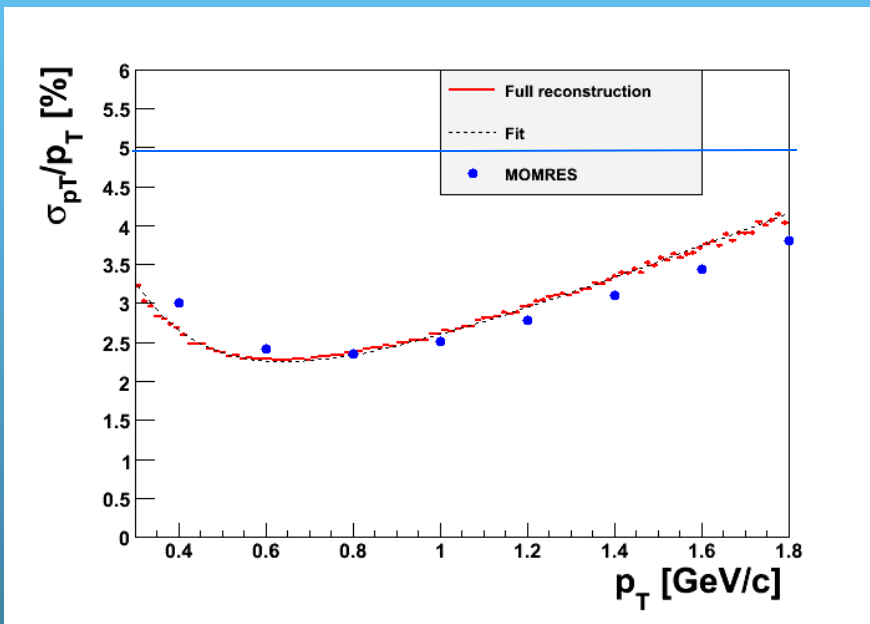
Problems:
- Silicon design is still changing
- Micromegas are not yet fully official

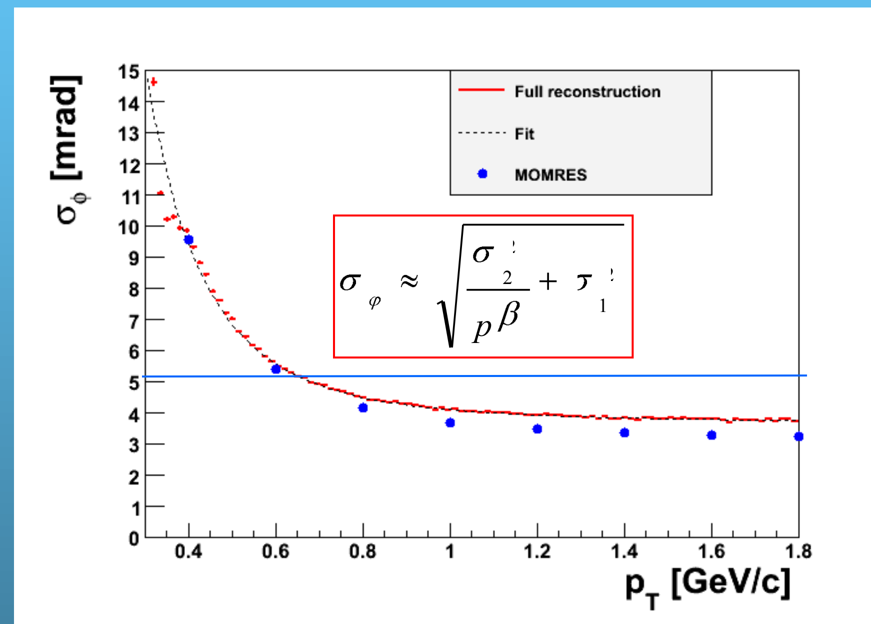
Central tracking - Performance

(using 4x2 layers of Silicon)

Track reconstruction requires at least 3 double layers $\langle \varepsilon \rangle \approx 91\ \%$

Once a combination of hits has been found, it is sent to the KF for the final fit



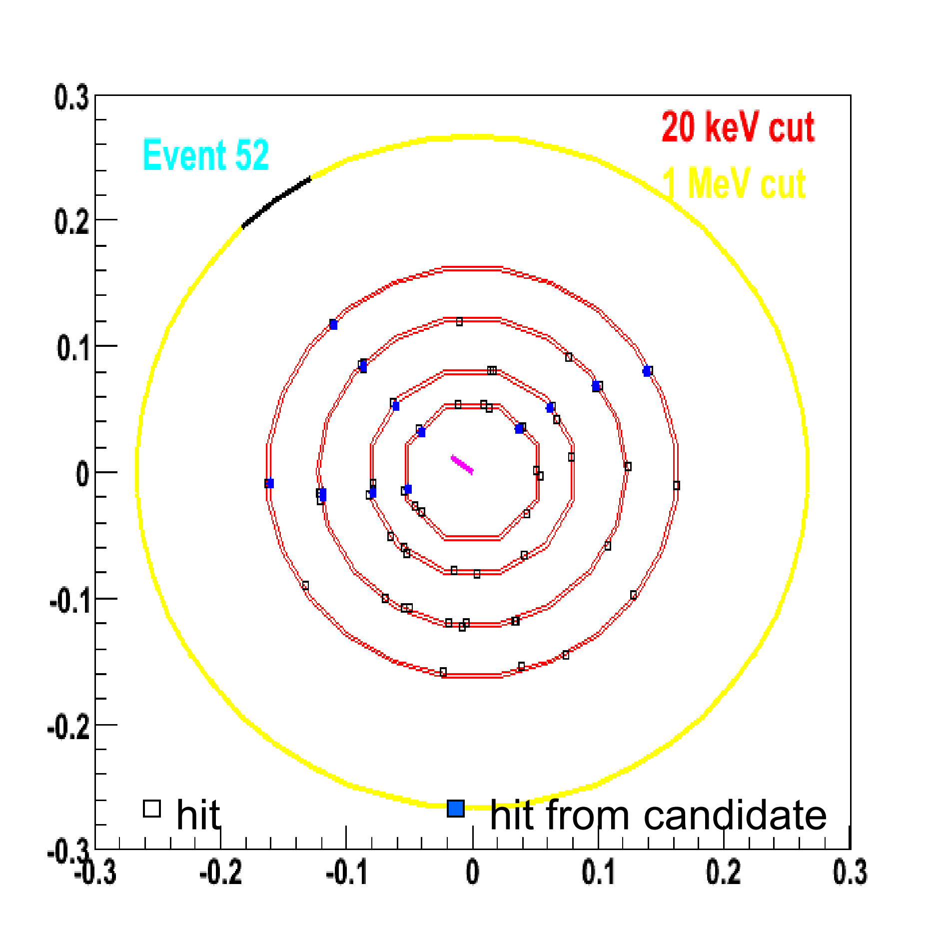


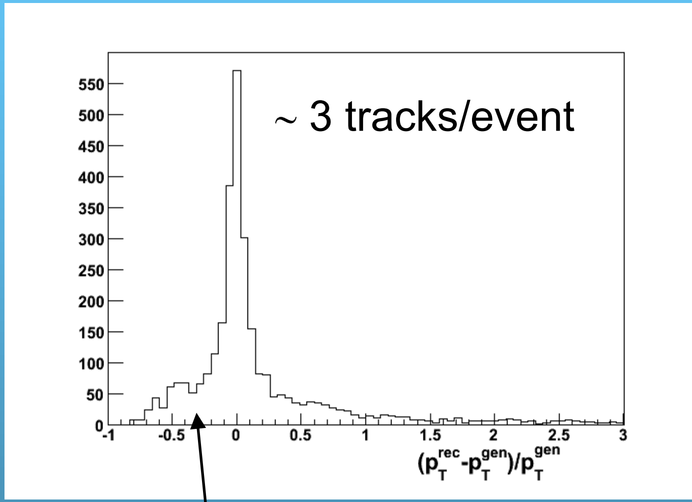
Agreement between Kalman Filter & MOMRES

All physics requirements met, but upper limit for θ

Effect of the background

Example of DVCS events at full luminosity (from Geant4): e+p→e+p+γ



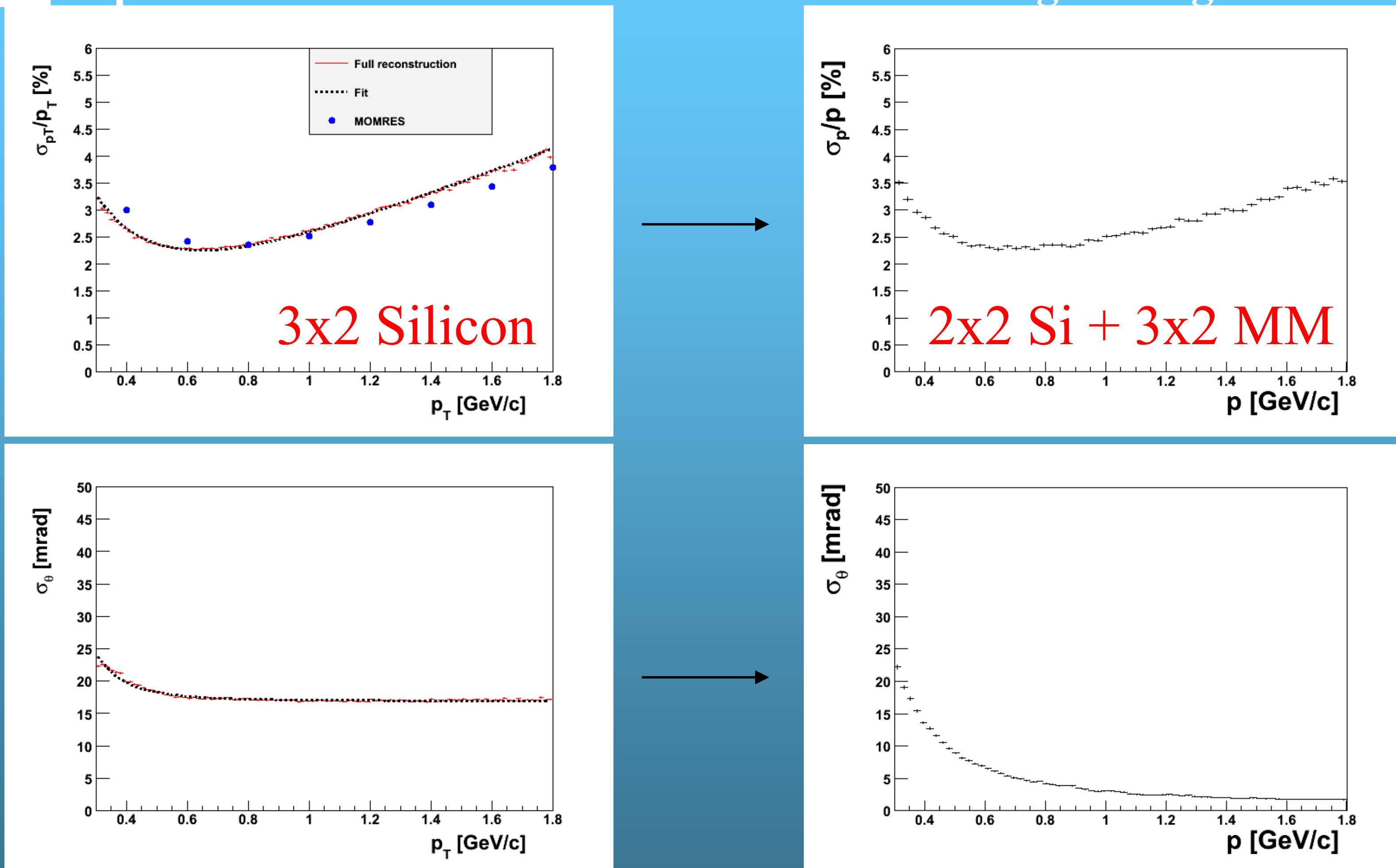


The few remaining fakes can be rejected using the CTOF

Small fraction of fake tracks (mainly « sister » tracks), $\varepsilon_{tracking}$>95%

Effect of the Micromegas

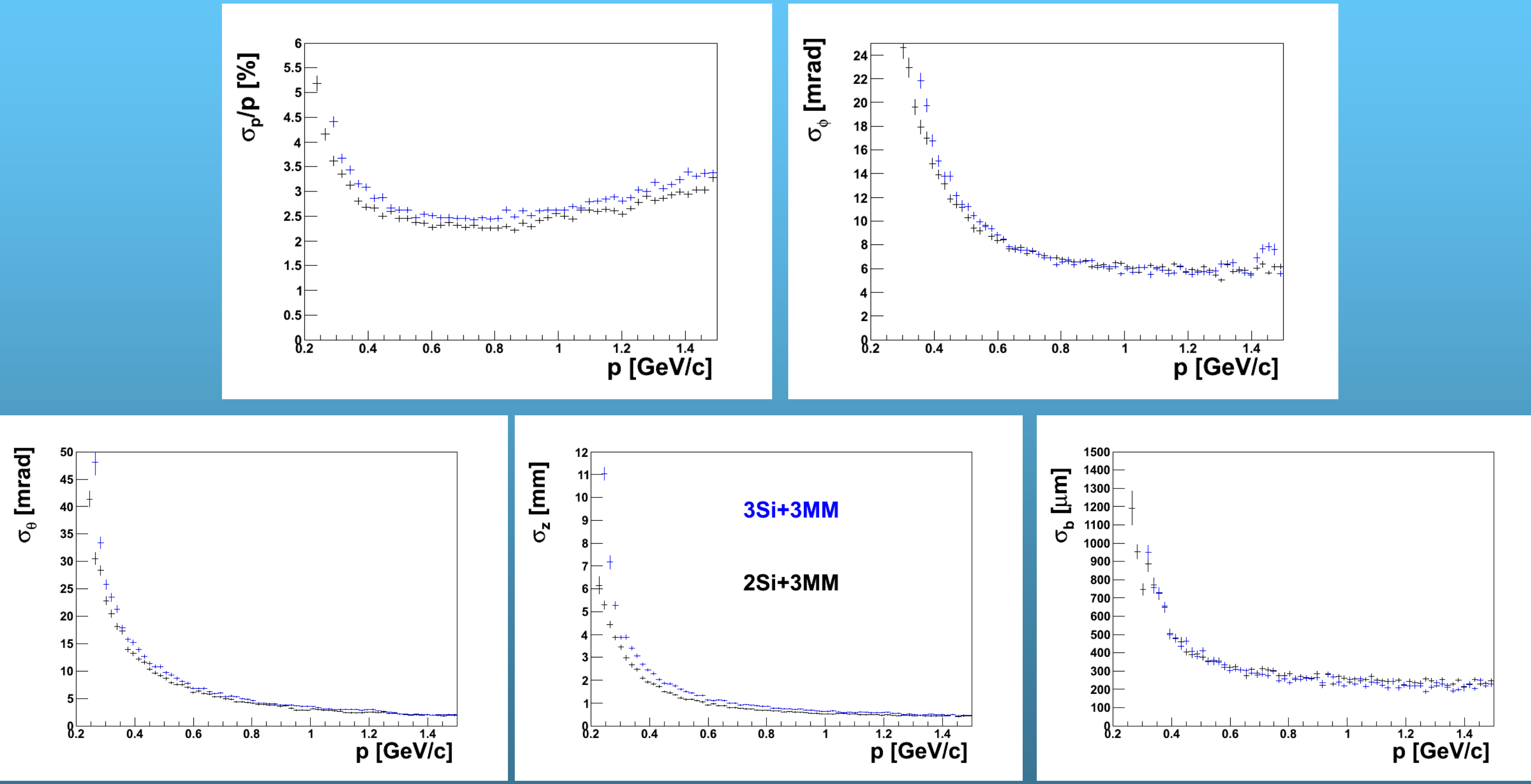
omparison between Silicon and Silicon+Micromegas designs:



⇒ Much better θ resolution, as expected by previous simulations

Central Tracking

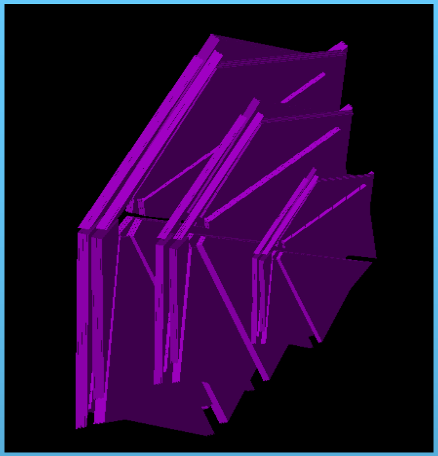
ffect of the number of Silicon layers in the tracking



ore multiple scattering, but limited effect

Forward tracker (DC, FVT)

Main features:

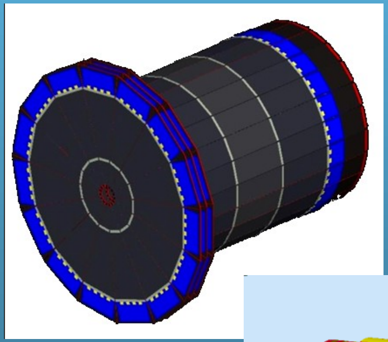


DC

- 3 regions, 3x2 superlayers, 3x2x6 planes
- 6 sectors of 60°
- 6° stereo angle
- plane tilted by 25° wrt the beam axis
- acceptance: 5° °

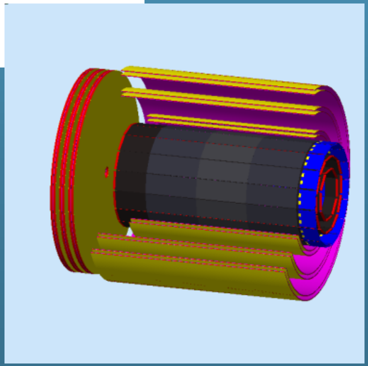
FST

- 3x2 layers
- trapezoidal tiles
- 12° stereo angle
- acceptance: 5° °



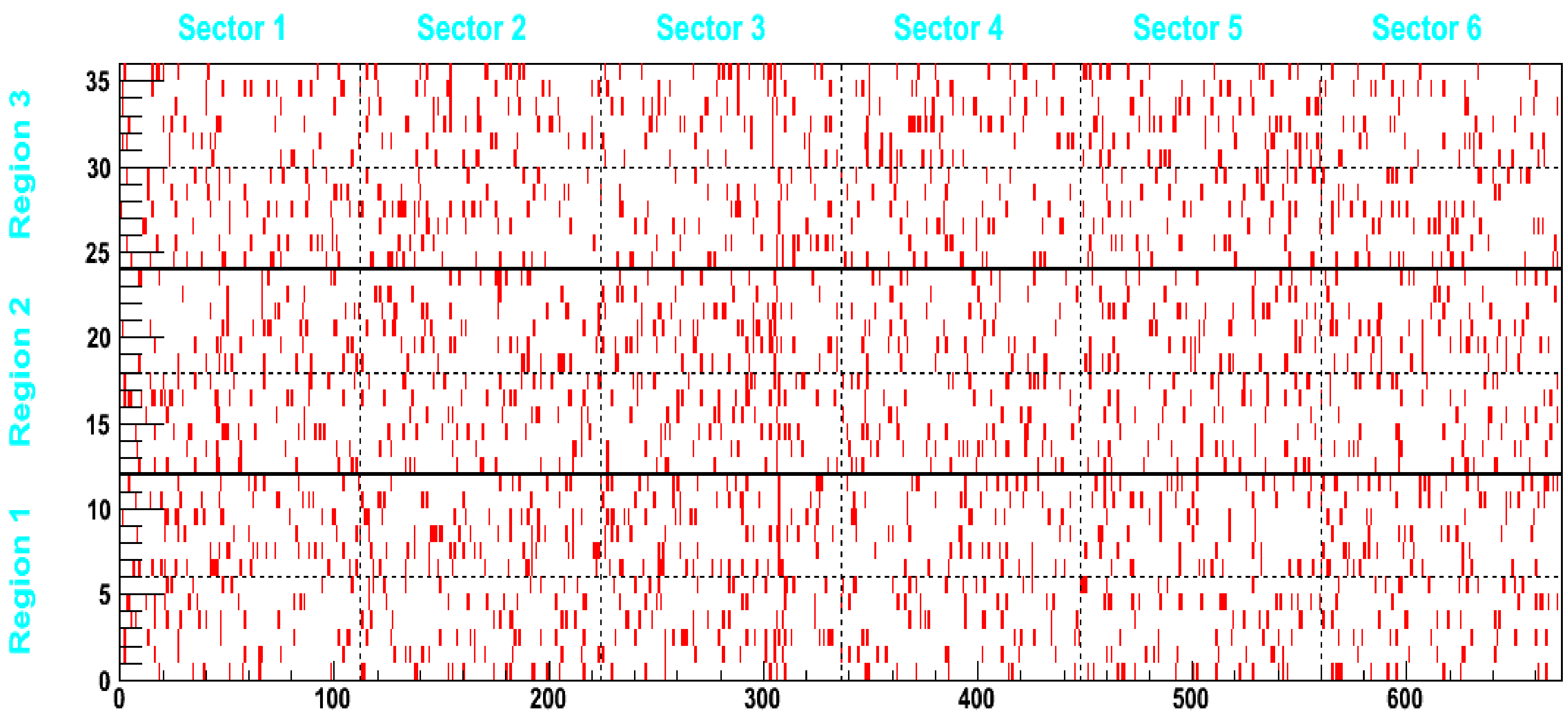
FMT

- 3x2 layers
- circular detectors
- 24 to 45° stereo angle
- acceptance: 5 (2?)° °



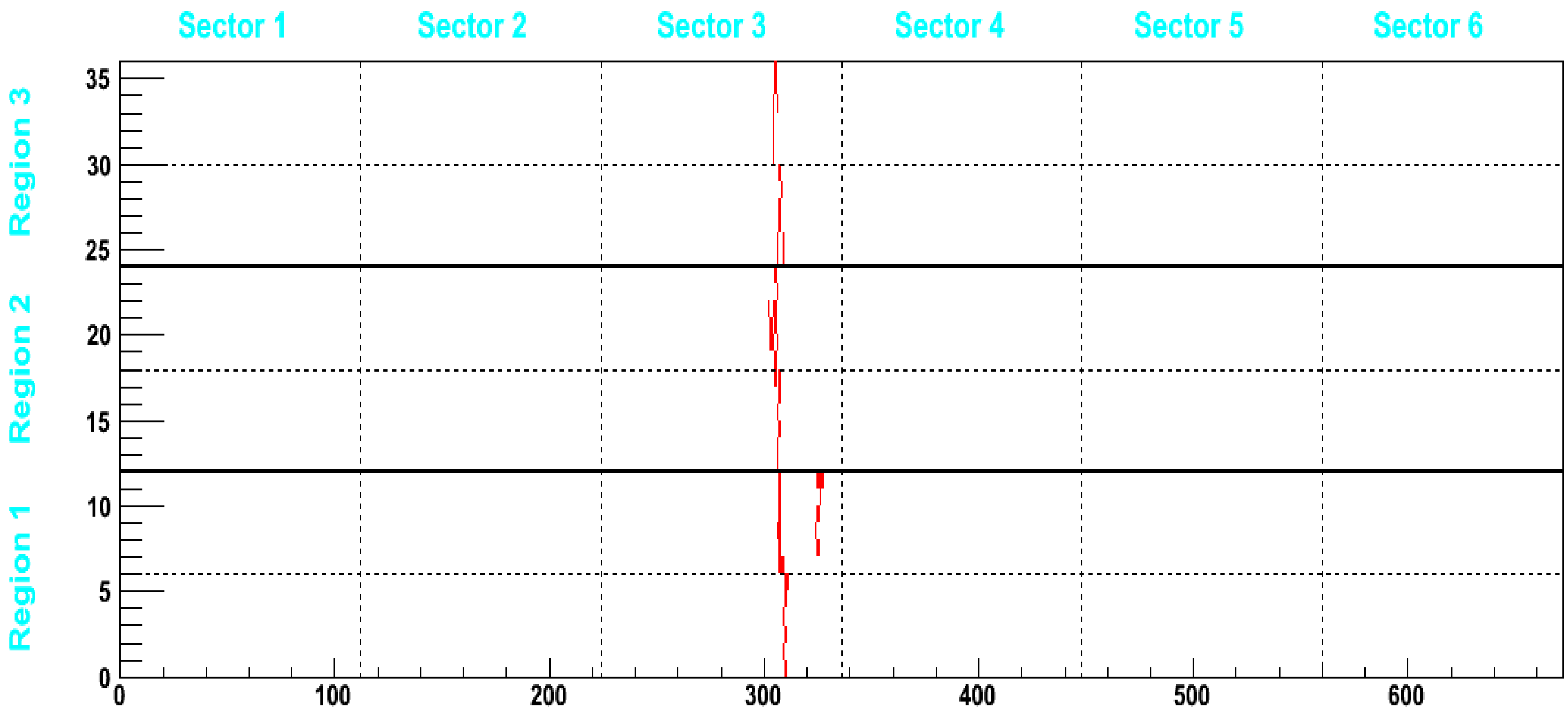
Reconstruction in DC

Starting point (uncorrelated background, just for illustration):



1) Find track candidates (patterns)
2) Find road(s) in each cluster
3) Fit of the track candidates (KF)

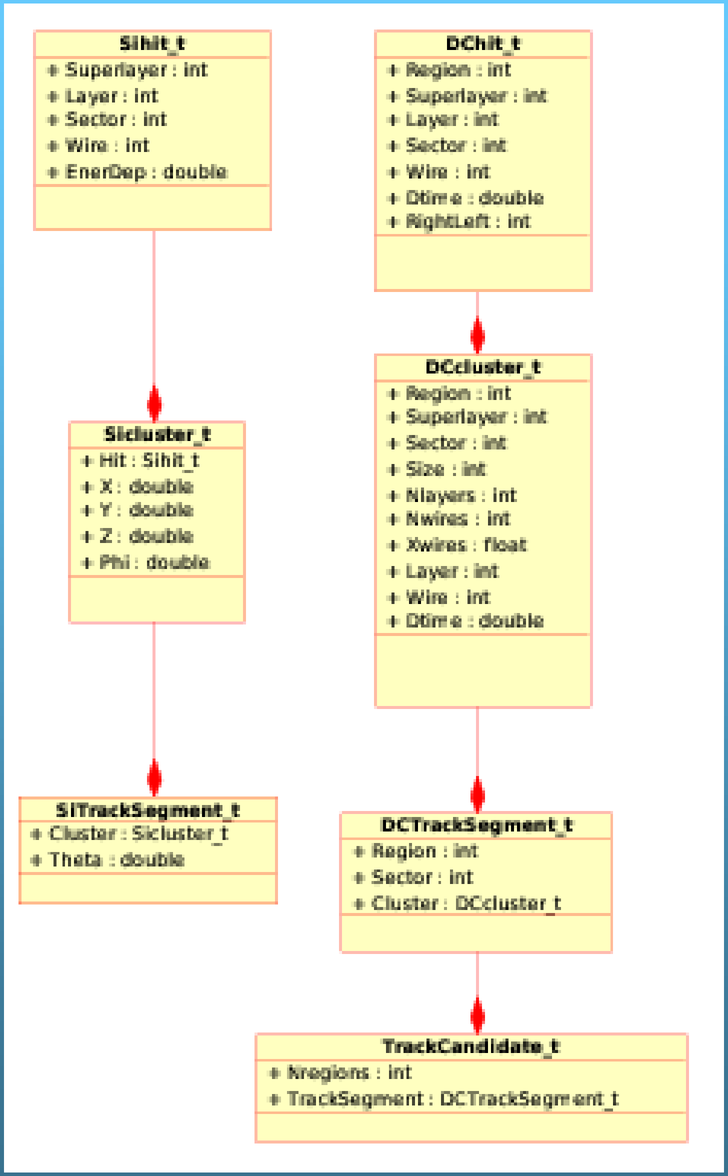
Track finding



1) Find clusters (in each superlayer)
2) Find track segments (in each region)
3) Find track candidates (3 regions)

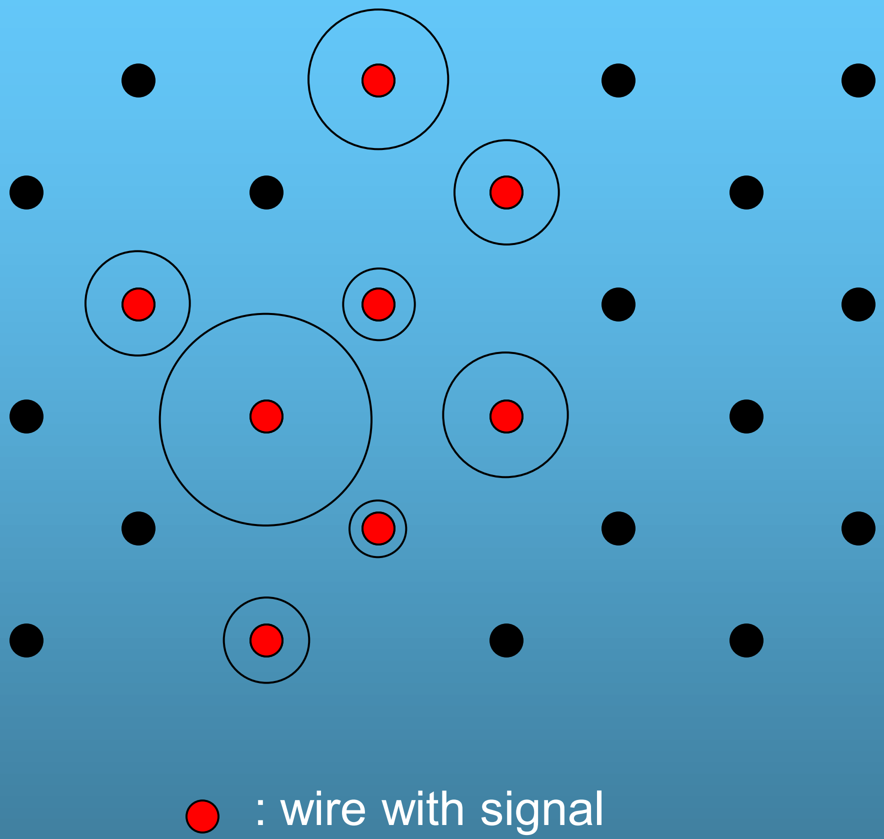
Corresponding structures in Socrat:
DChit, DCcluster, DCTrackSegment, DCTrackCandidate

Structures in Socrat

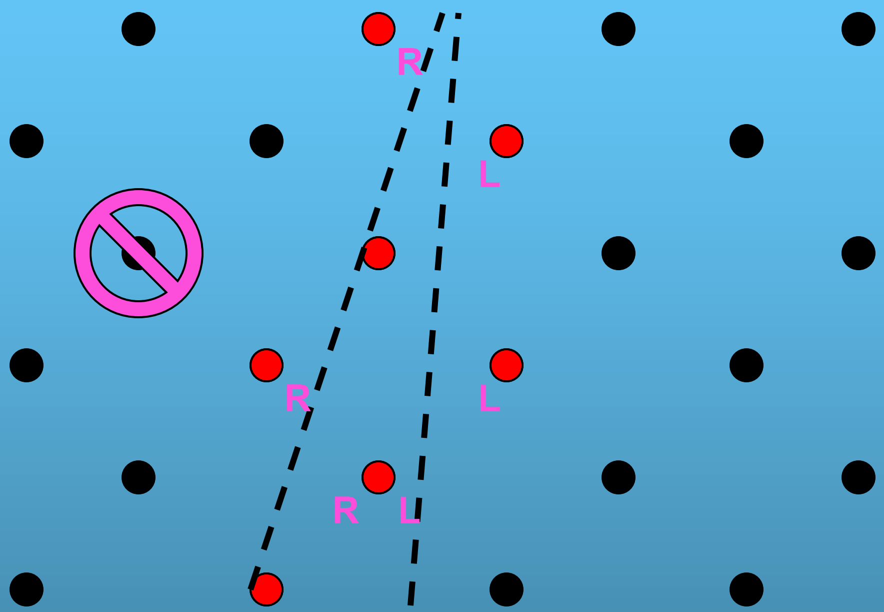


Find roads in clusters

Starting point:



Find roads in clusters

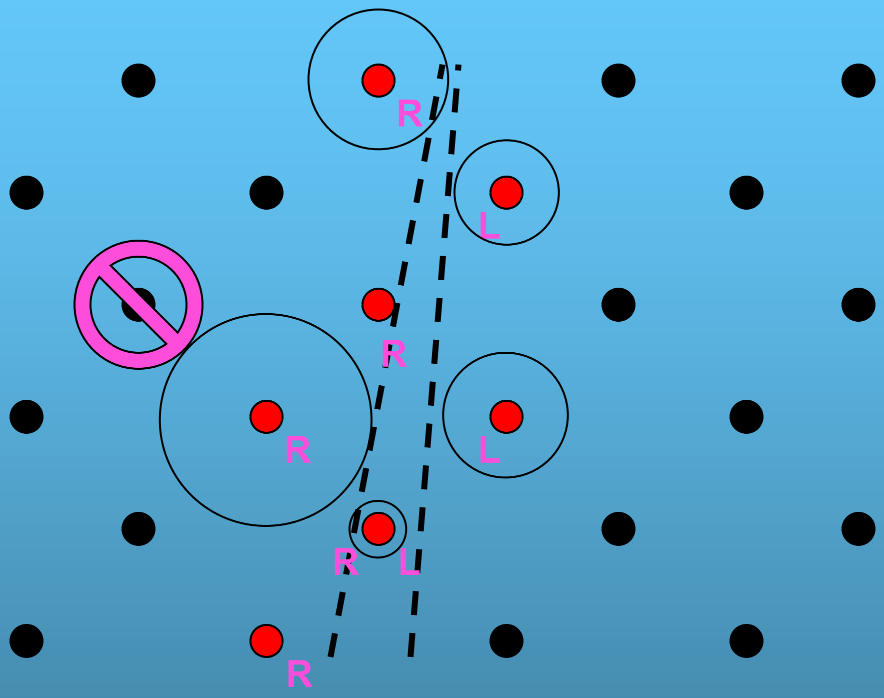


Step 0: fit using wires only
(and large errors)

→ reject some hits
→ solve some L/R ambiguities

: wire with signal

Find roads in clusters



: wire with signal

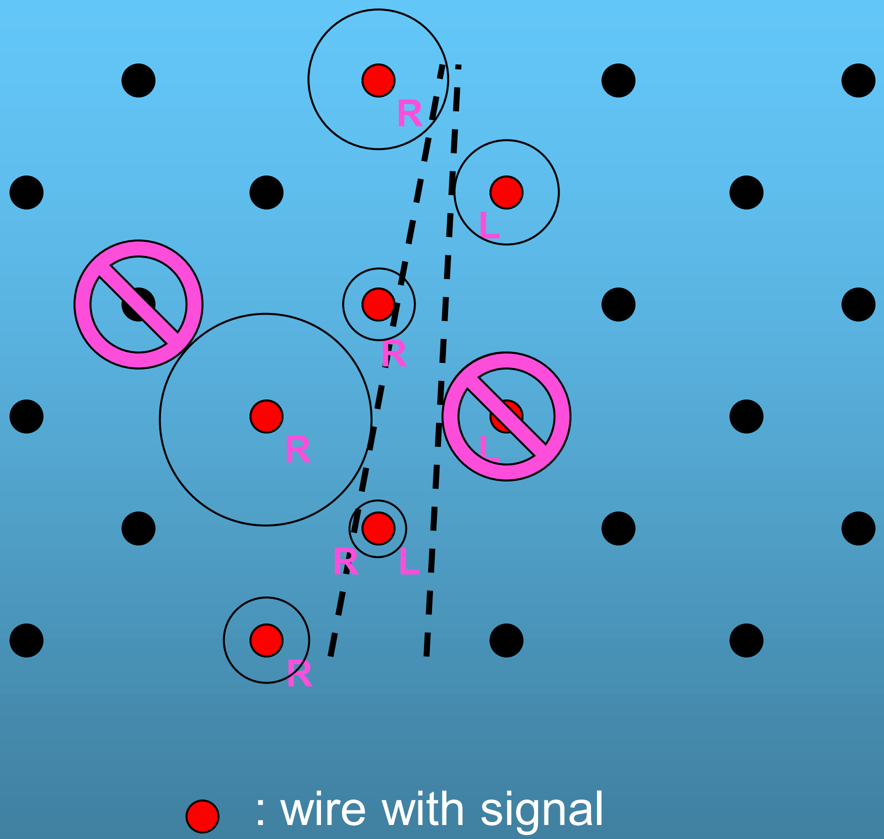
Step 0: fit using wires only (and large errors)

→ reject some hits
→ solve some L/R ambiguities

Step i: fit using wires, or drift dist. if L/R solved

→ solve more L/R ambiguities

Find roads in clusters



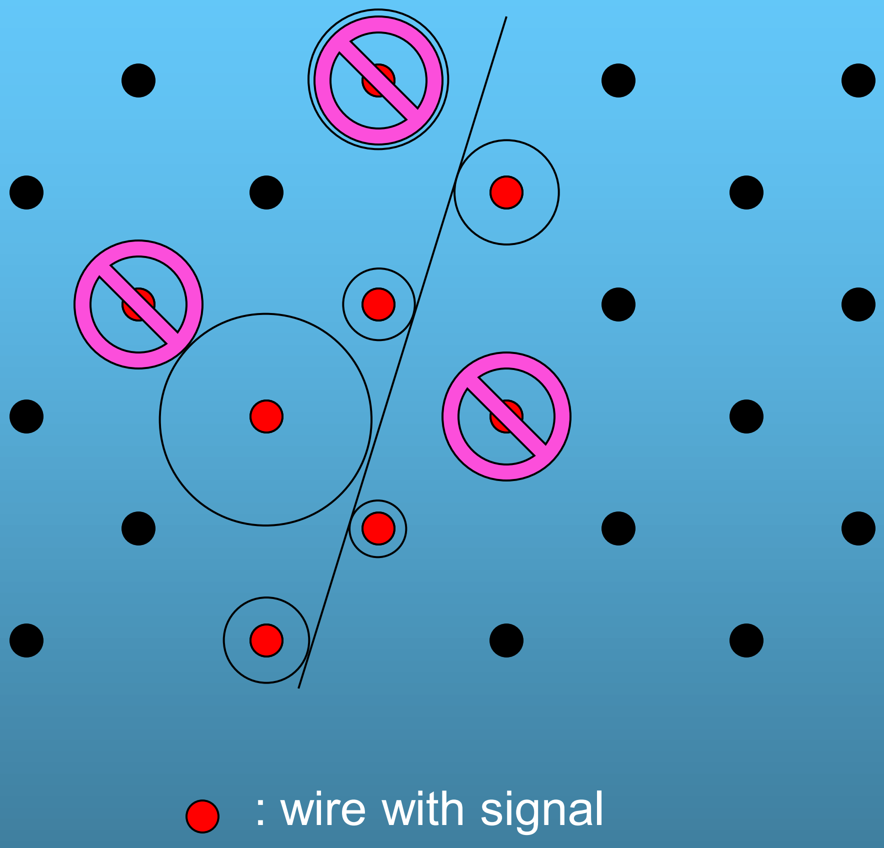
Step 0: fit using wires only (and large errors)

→ reject some hits
→ solve some L/R ambiguities

Step i: fit using wires, or drift dist. if L/R solved

→ solve more L/R ambiguities

Find roads in clusters



Step 0: fit using wires only (and large errors)

→ reject some hits
→ solve some L/R ambiguities

Step i: fit using wires, or drift dist. if L/R solved

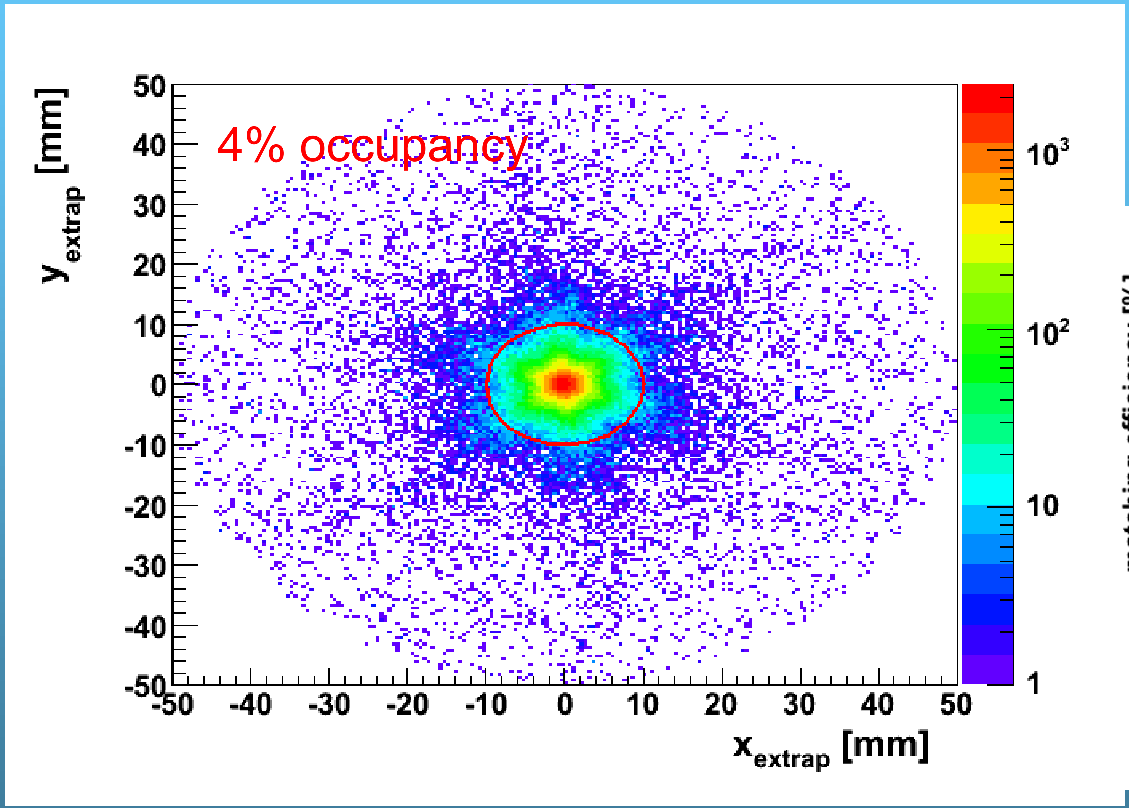
→ solve more L/R ambiguities

Step N(=4): reject bad hits, solve all L/R

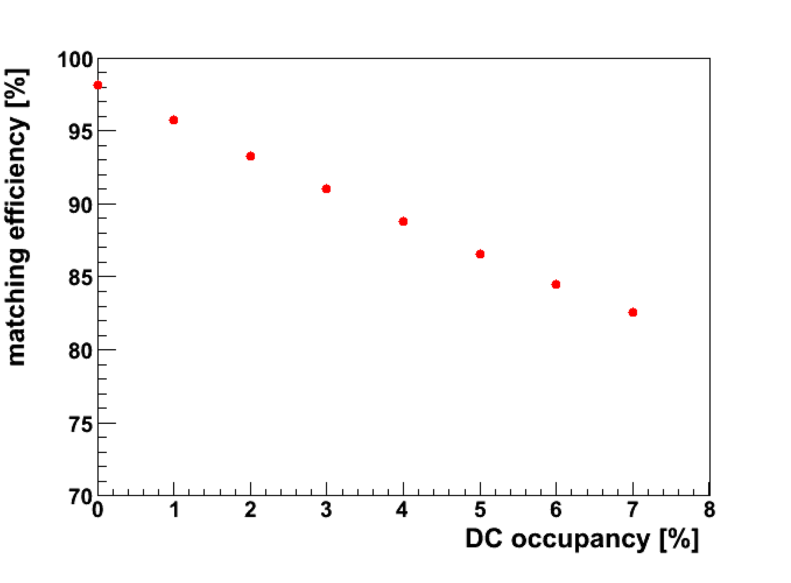
Small probability of mistake (~10%), increase with background

Matching tracks with the FVT

The track obtained in the DC is extrapolated back to the last layer of the FVT. The algorithm then looks for a FVT track segment around this extrap. position (R=1cm)



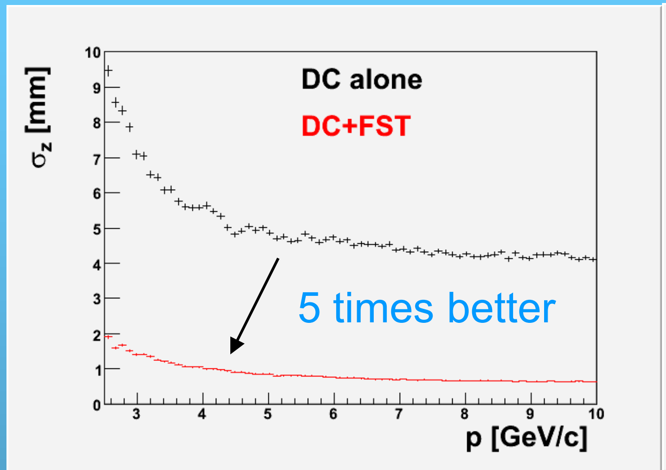
(protons at θ = 15°)

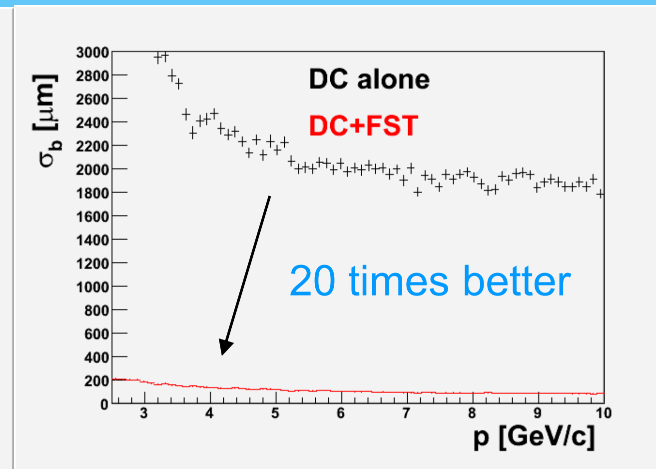


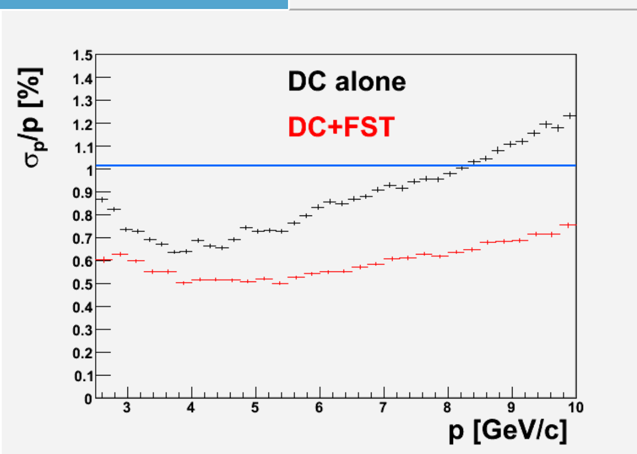
Good DC/FVT matching efficiency, but depends on DC occupancy

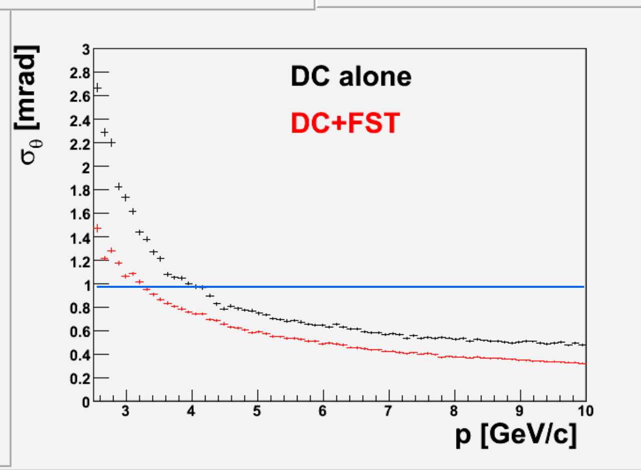
Resolutions with DC+FVT

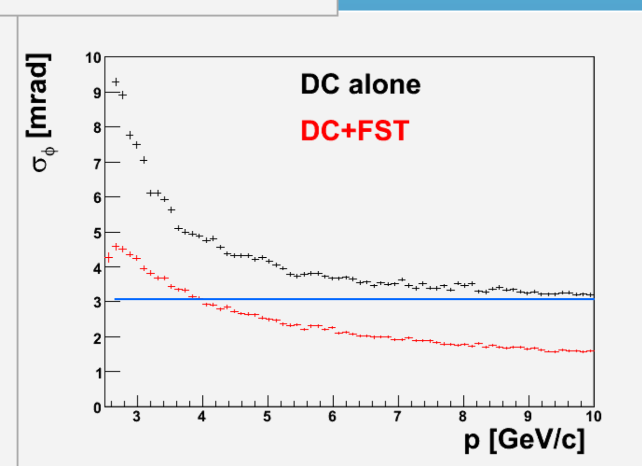
(electrons at θ = 15°, tracks from GEMC):







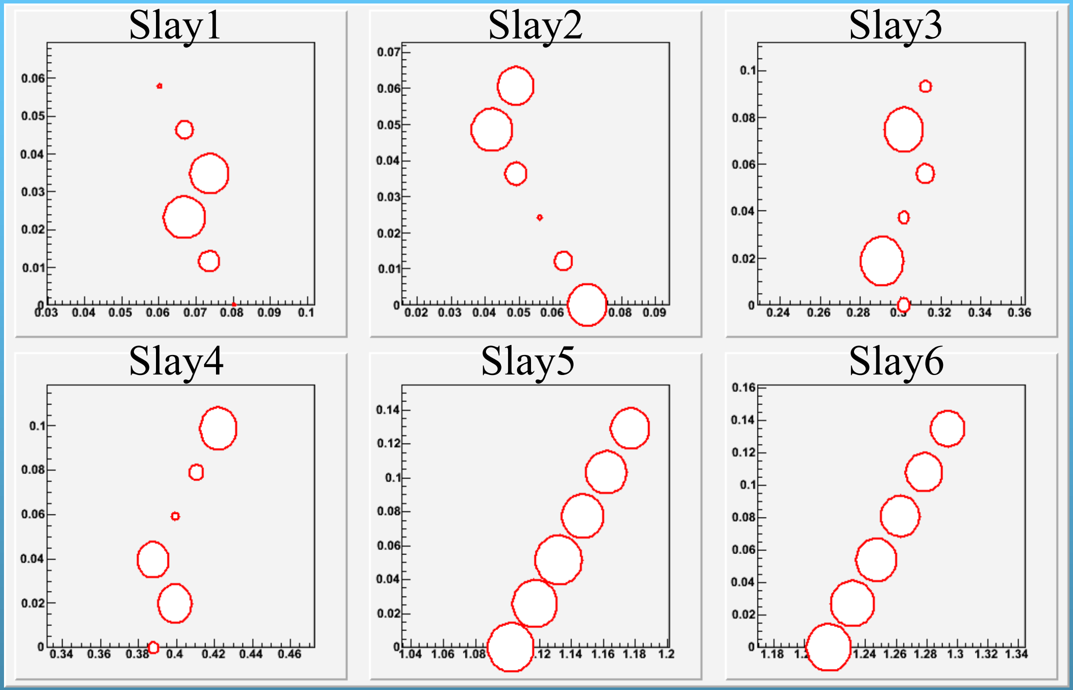


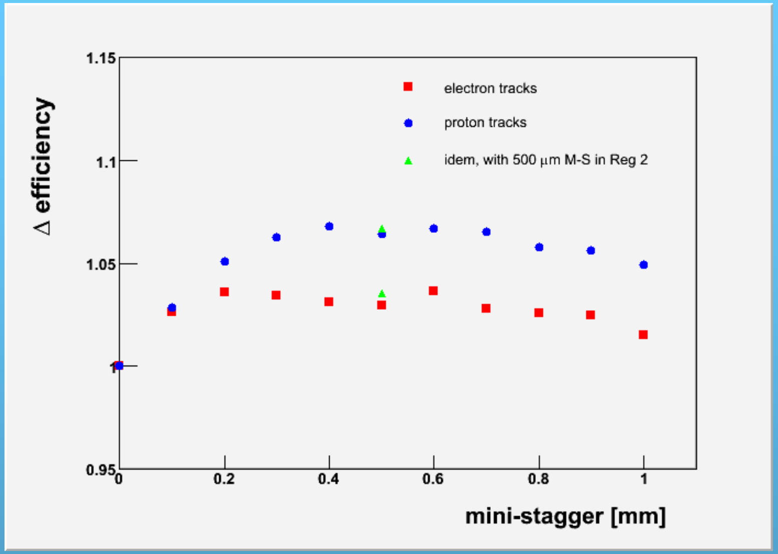


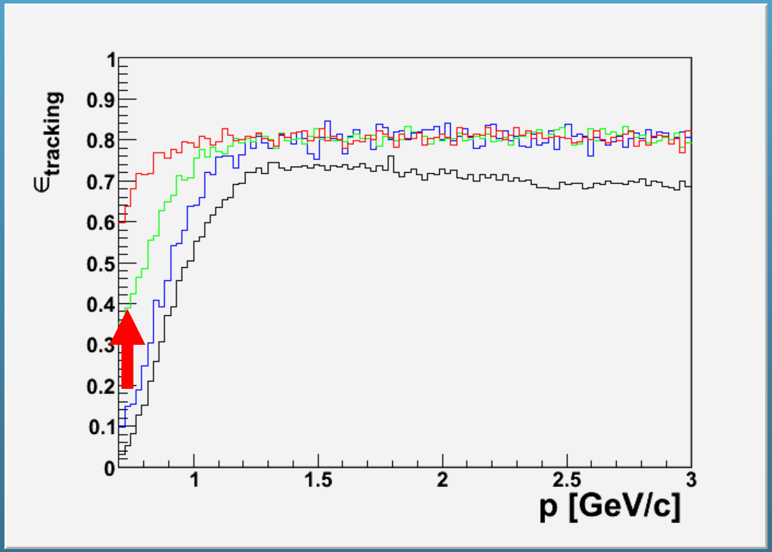
FVT largely improves the vertex resolution, φ, θ and p

Mini-stagger in Region 3

A proton track in DC…







Significant increase of efficiency at small momentum

Momentum bias and energy loss

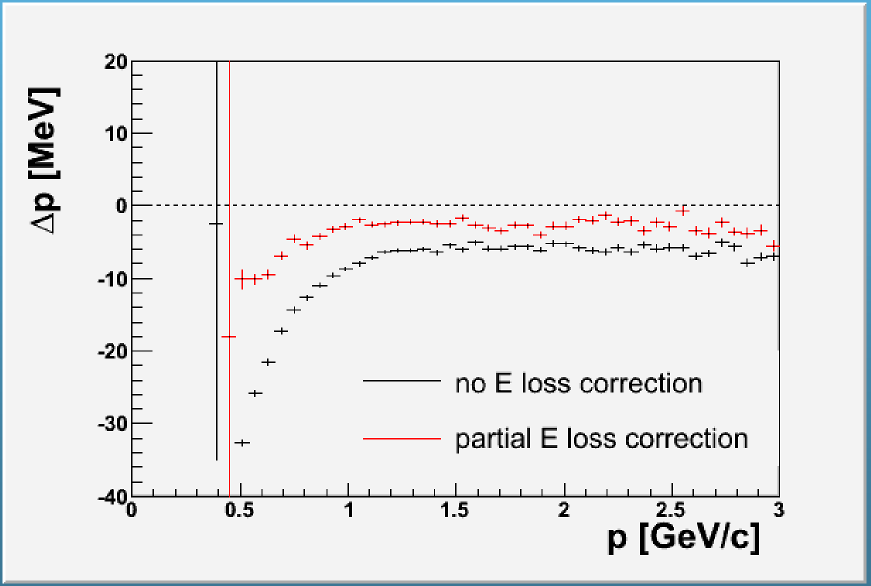
Energy loss through material is well known…

$$-\frac{dE}{dx} = Kz^2 \frac{Z}{A}\frac{1}{\beta}\left[\frac{1}{2}\ln\left(\frac{2m_e c^2 \beta \gamma^2 T_{max}}{I^2}\right) - \beta\right]$$ (Bethe-Bloch)

… and can be easily taken into account step by step in the Kalman Filter.

So far:

- DC (argon gas)
- Air
- FST (silicon+carbon fibers)
- Target (LH2)



Conclusion

elped the design of several detectors:
- strip layout of the BST (zigzag to graded stereo angle)
- mini-stagger in DC Region 3 (300 microns)
- geometry of the Micromegas FVT at small acceptance (in progress)

ode available on svn

ee CLAS note 2008-015 for more details

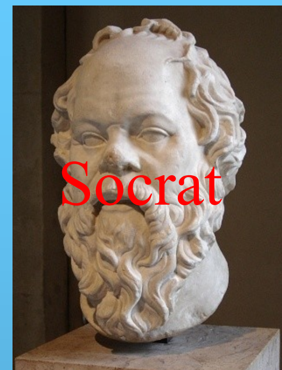
uns with Geant4 events (or internal track generator)

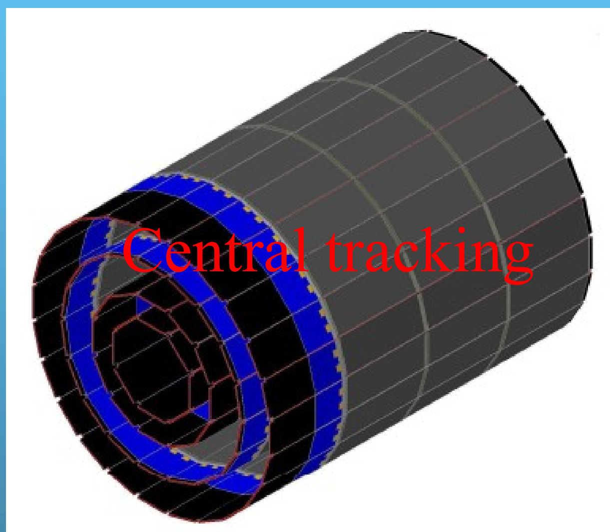
ser friendly script to run Gemc+Socrat on the batch (Simu.pl)

wo (unwanted) sons with the same algorithms translated in different languages:
- JSocrat
- Scrat

. but work is still needed on the algorithms!

Additional slides





Central tracking - backup

Acceptance

Stereo angle

Overlap setup

Acceptance optimization

Uncertainty on B field

Background

Forward tracking - backup

FST alone

FST spacing

FST background

DC – resol & background

DC – L/R ambiguity

FST fake tracks

DC+FST – p resolutions

FST acceptance

Kalman Filter - backup

Initialization

Multiple scaterring

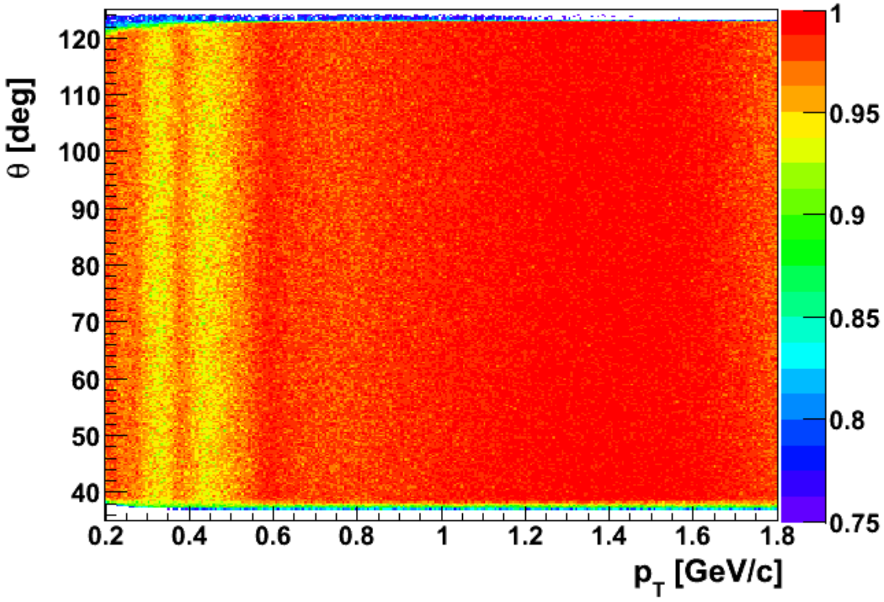
Socrat - backup

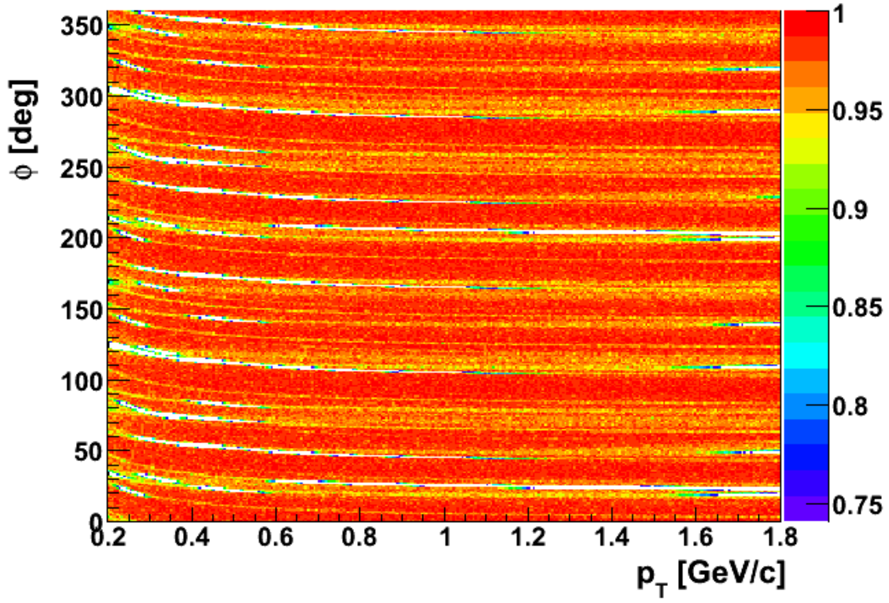
Misalignments

Generation routine

BST - acceptance

Track reconstruction requires at least 3 (out of 4) superlayers

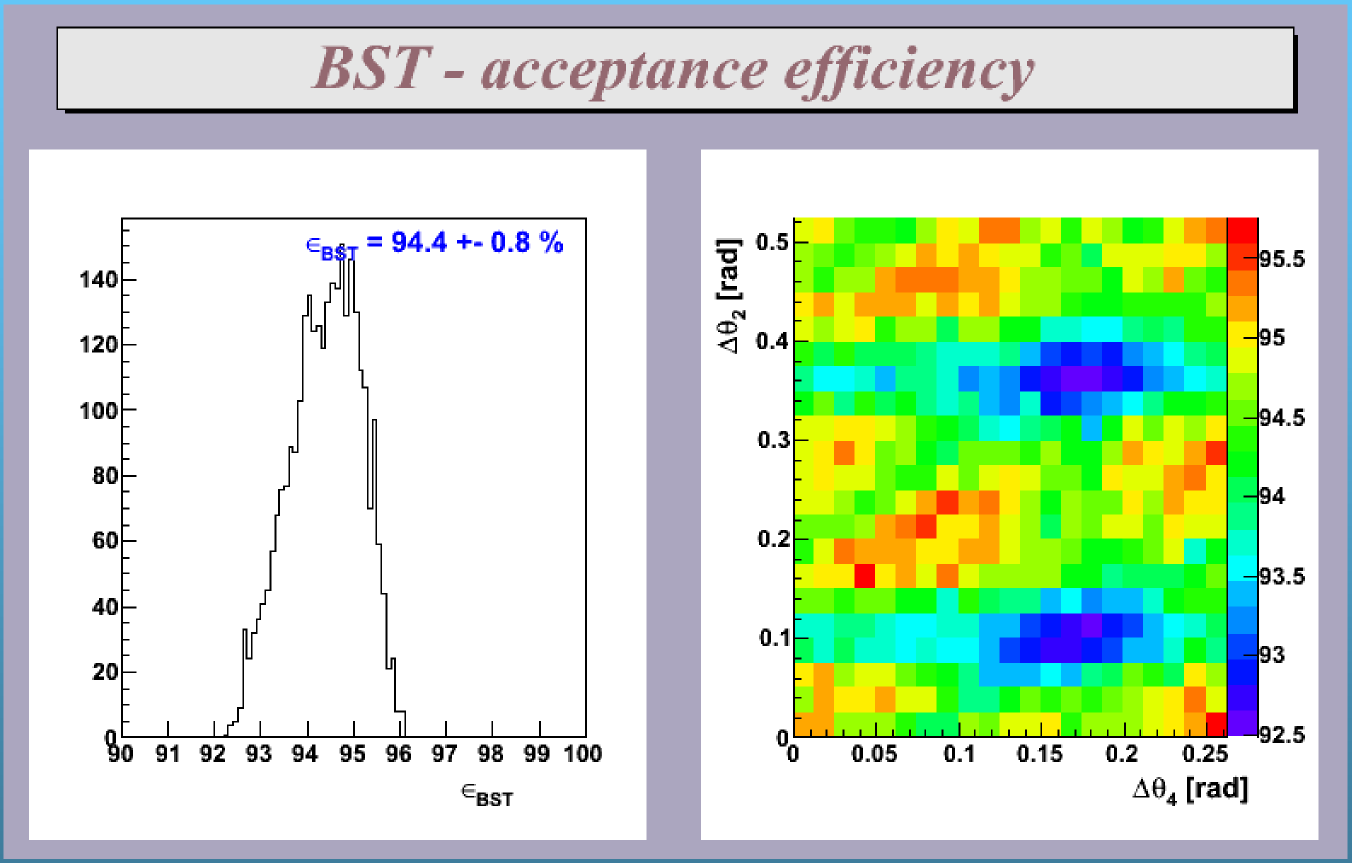


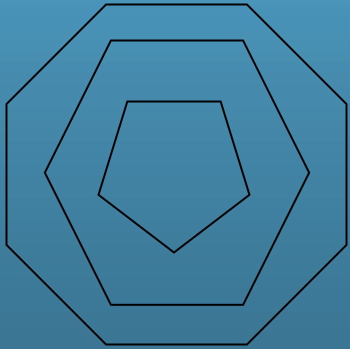


$\langle \varepsilon \rangle \approx 94\ \%$ (thin target)

BST – acceptance optimization

One can increase the acceptance efficiency by rotating the different double layers

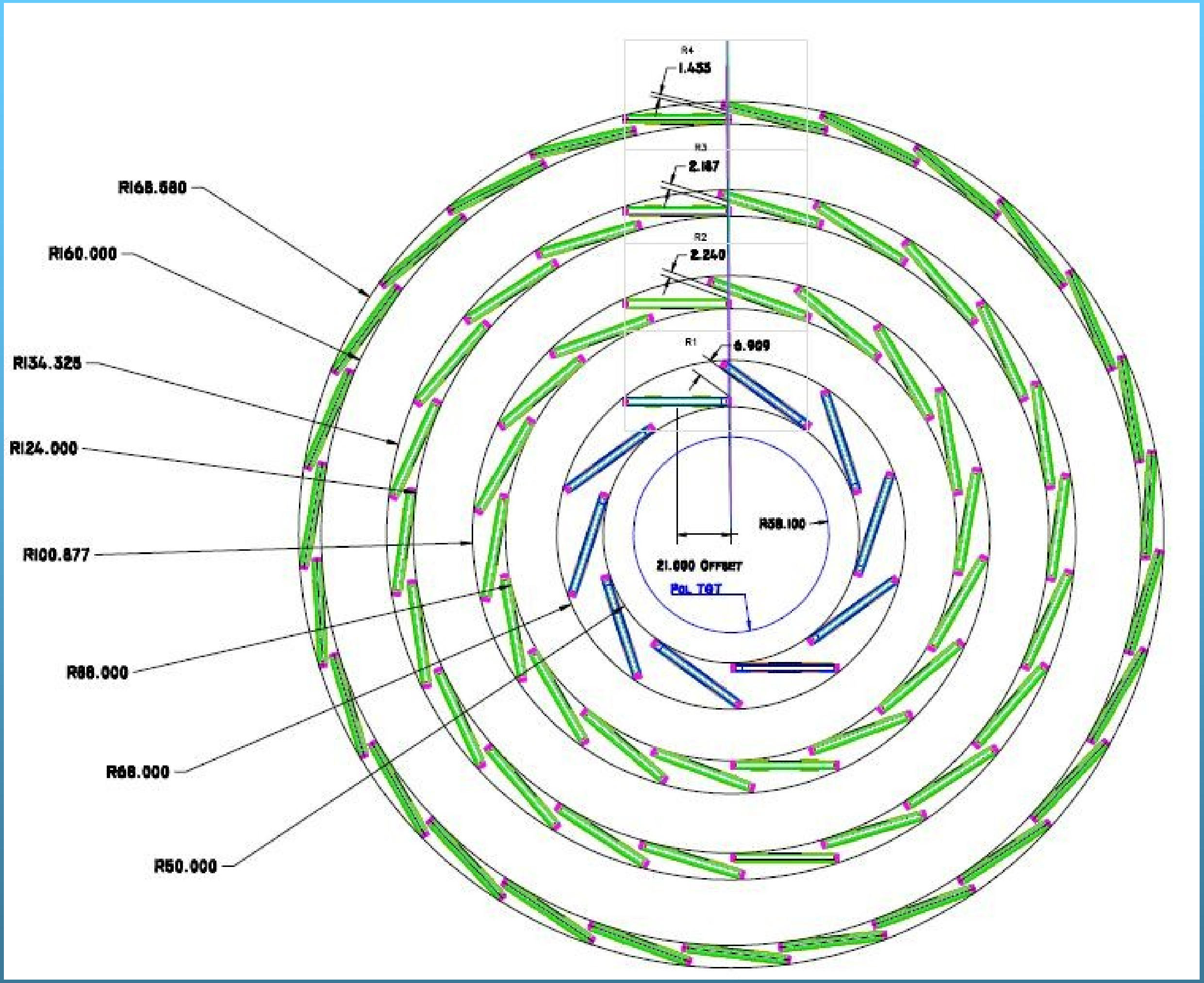




BST – stereo angle

α_S	σ_1 (p_T,%)	σ_1 (φ,mrad)	σ_1 (θ,mrad)	σ_1 (z,mm)	Track cand / event
3°	2.69	4.50	18.9	2.35	2.95
6°	3.04	5.09	12.0	1.43	4.49
9°	3.26	5.48	9.77	1.12	5.85
15°	4.05	6.66	8.74	0.93	6.55

BST – overlap design

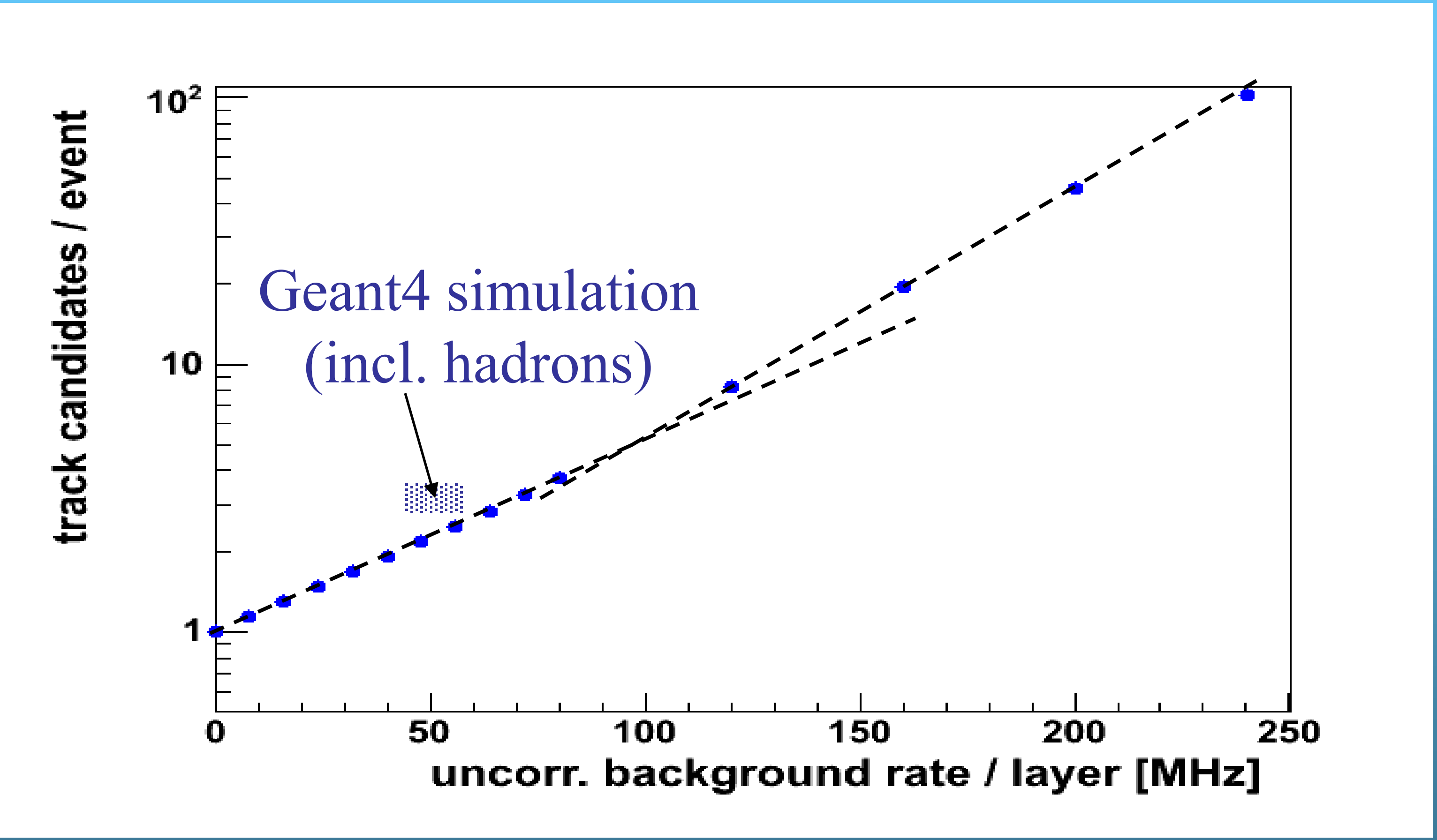


BST – uncertainty on B field

B field	σ_1 (p_T,%)	σ_1 (φ,mrad)	σ_1 (θ,mrad)	σ_1 (z,mm)
0.99%	2.71	4.67	19.1	2.41
1.00%	2.69	4.50	18.9	2.35
1.01%	2.83	4.69	19.1	2.41

BST – background studies

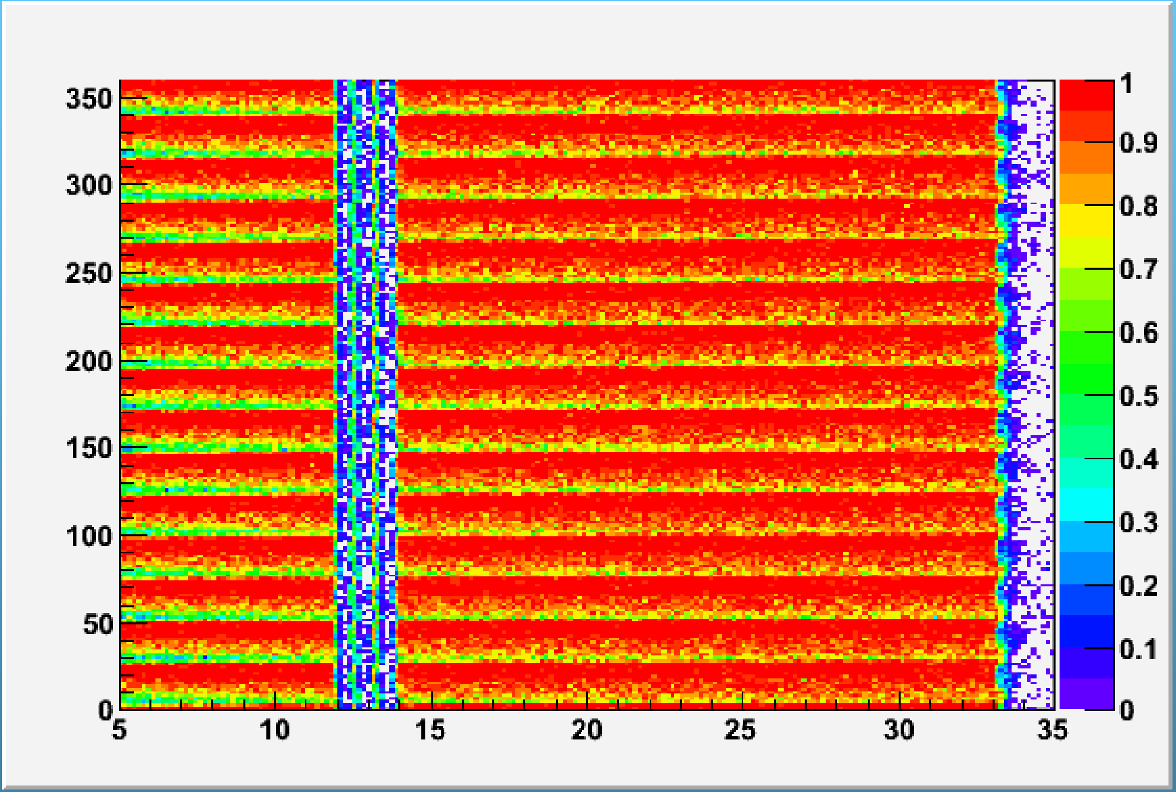
(using the uncorrelated background from the tracking code)



Good agreement with Geant4 background, gives the behaviour with the rate

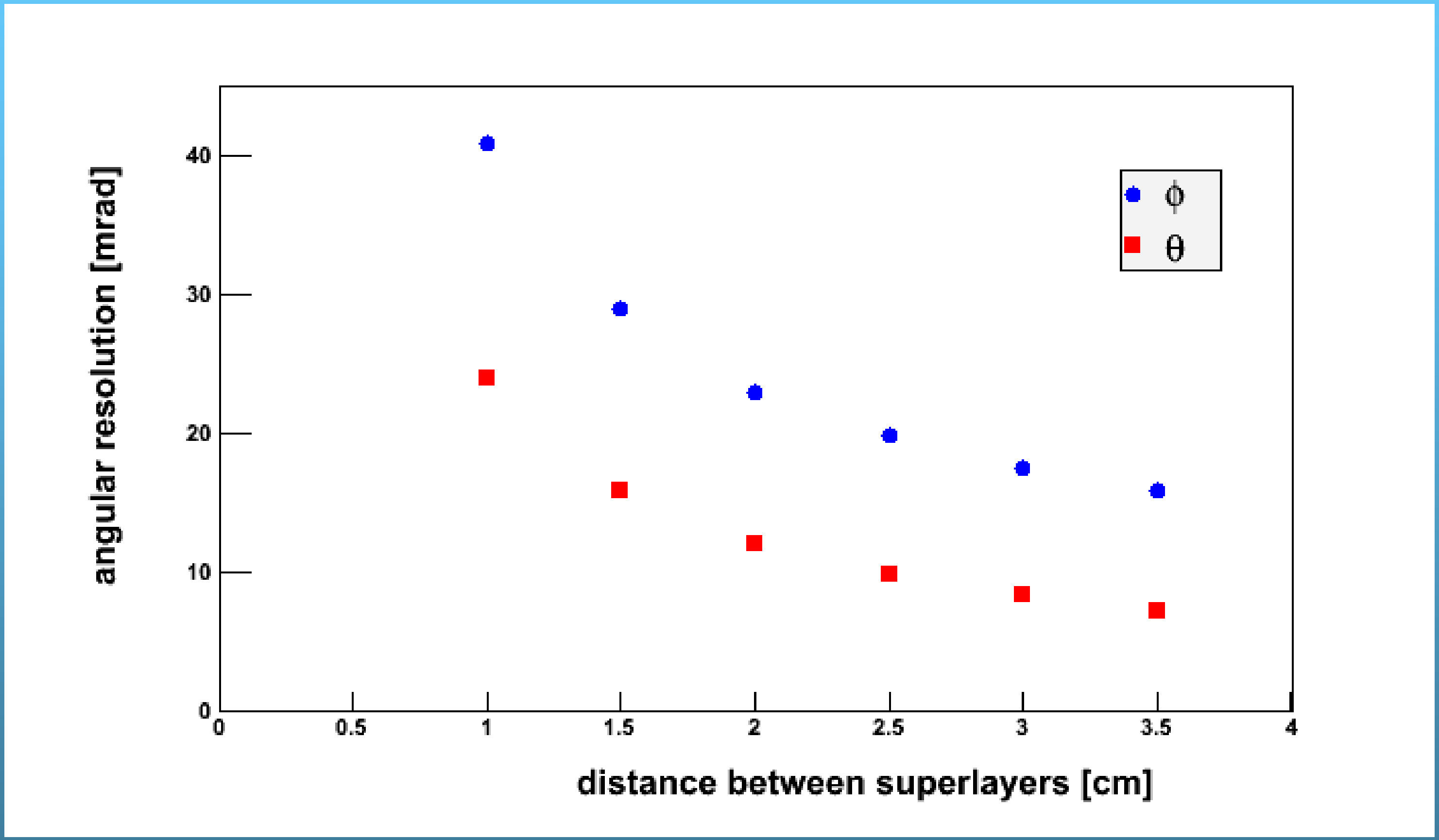
FST - acceptance

Track reconstruction requires 3 out of 3 superlayers



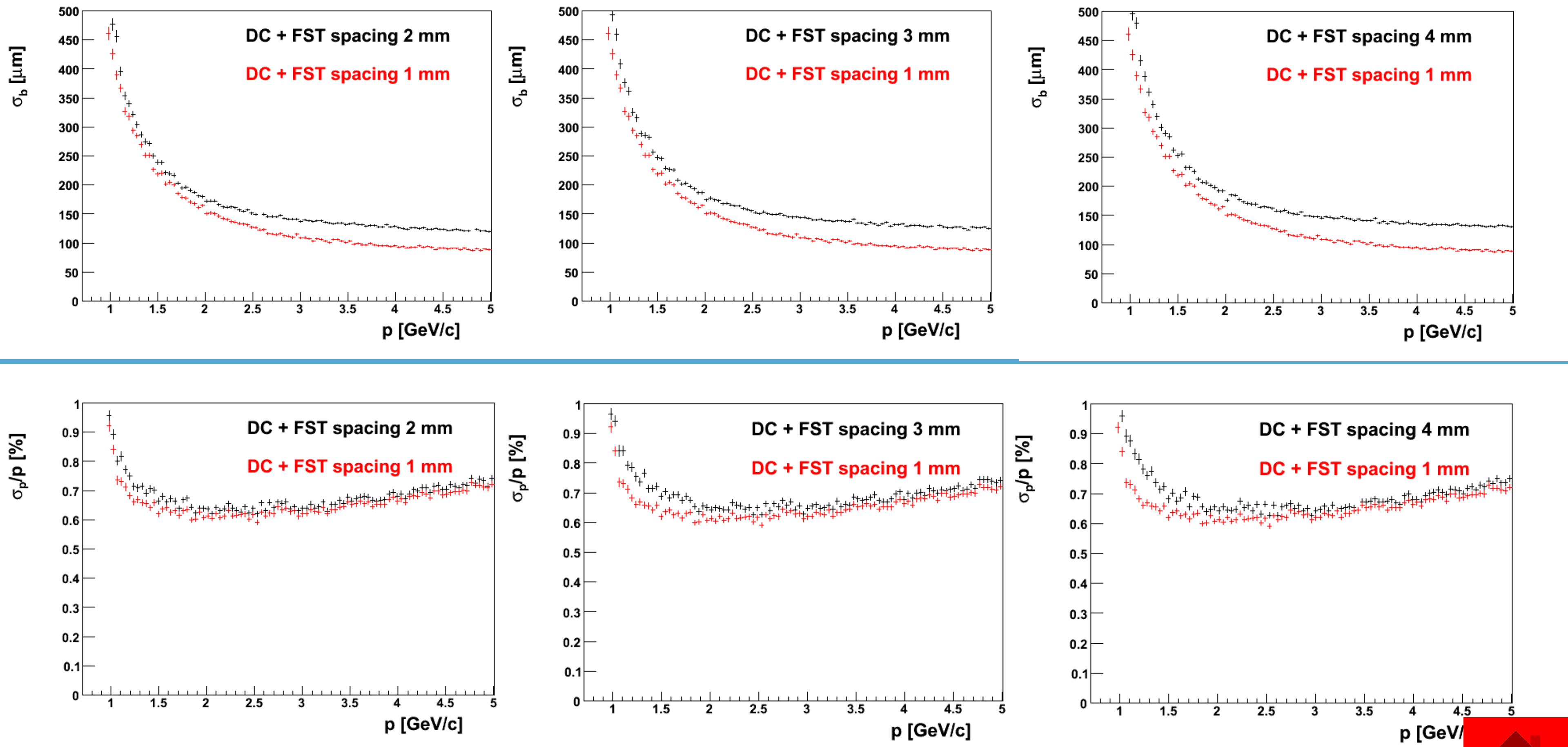
$\langle \varepsilon \rangle \approx 82\ \%$ (thin target)

FST alone



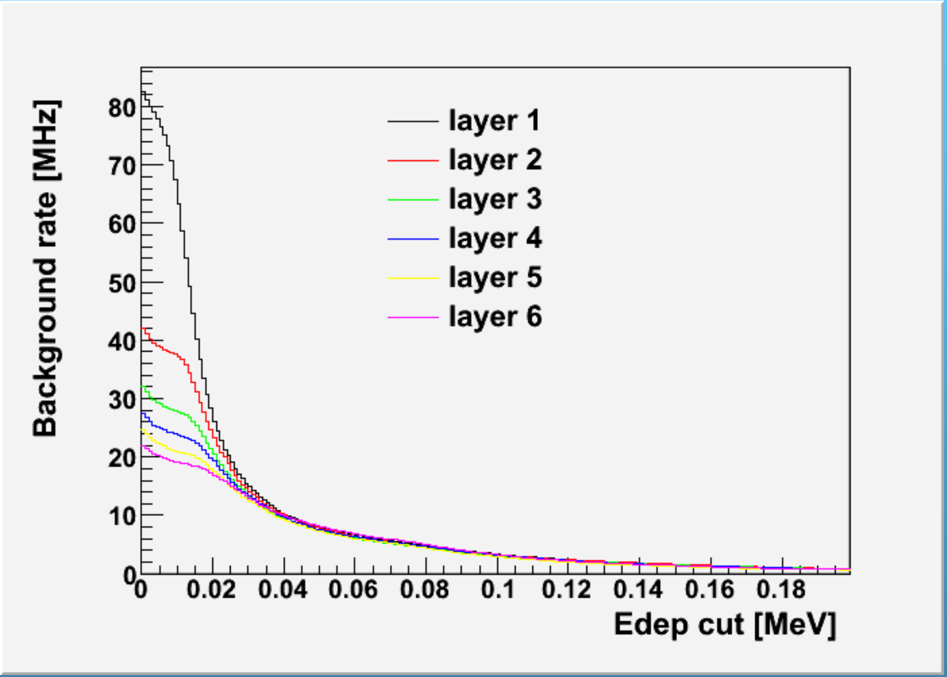
angular resolution [mrad]
40
30
20
10
0
0
0.5
1
1.5
2
2.5
3
3.5
4
distance between superlayers [cm]
φ
θ

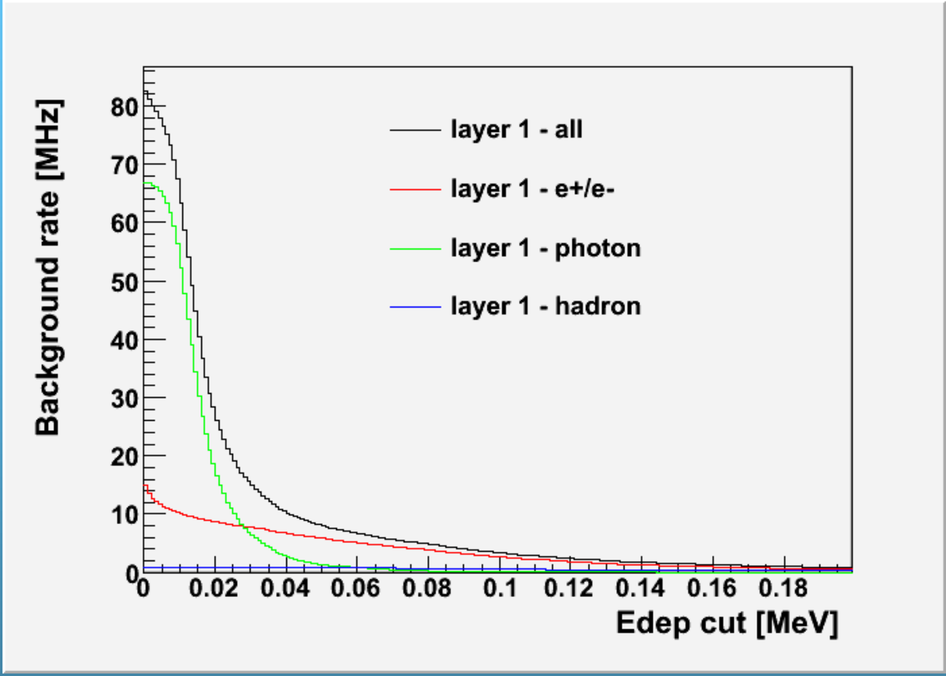
FST – spacing of superlayers



→ Better to have small distance between the superlayers

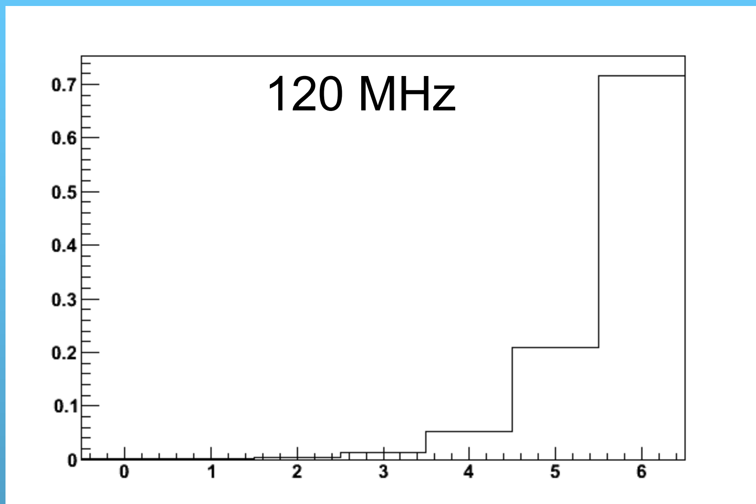
FST – background

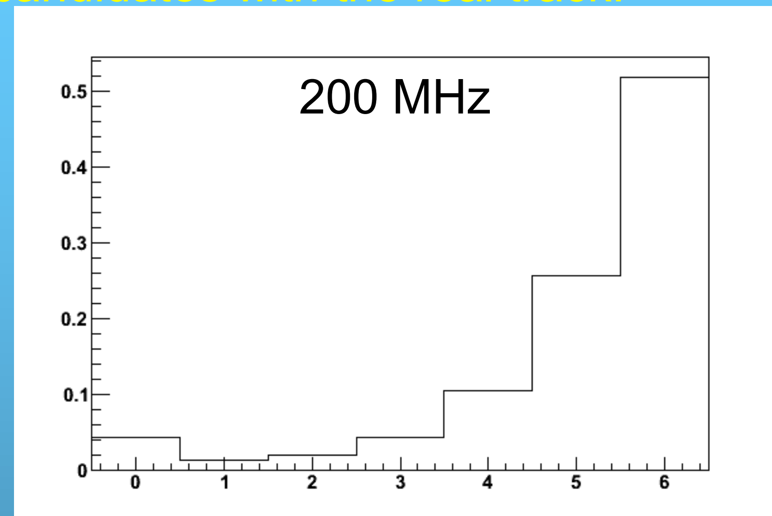


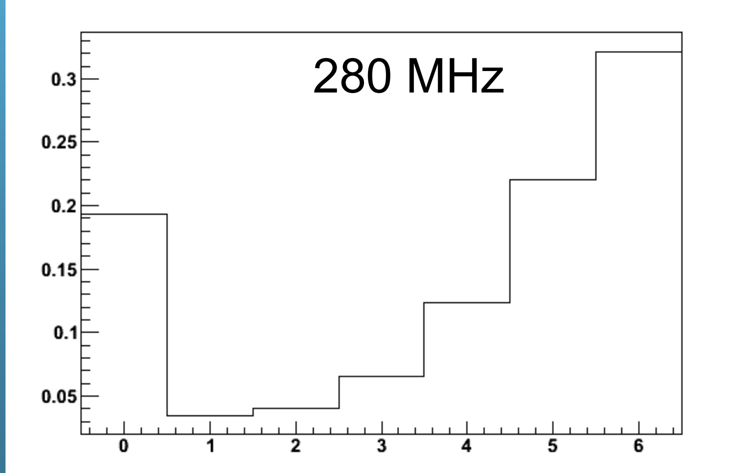


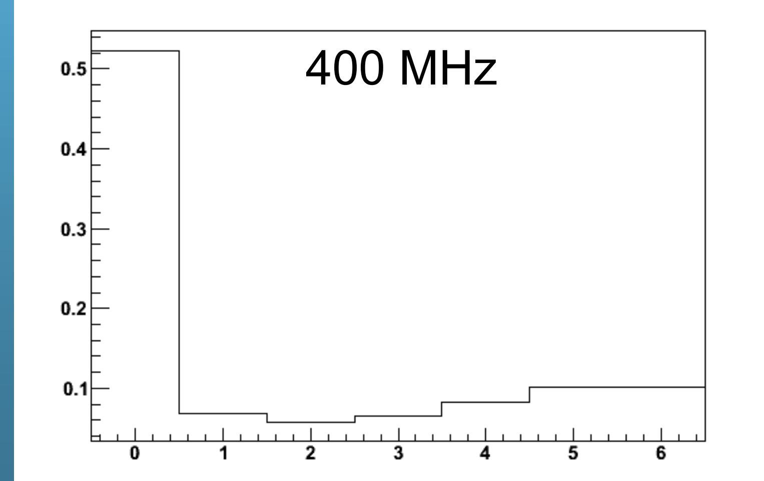
FST - fake tracks

Number of common hits of track candidates with the real track:



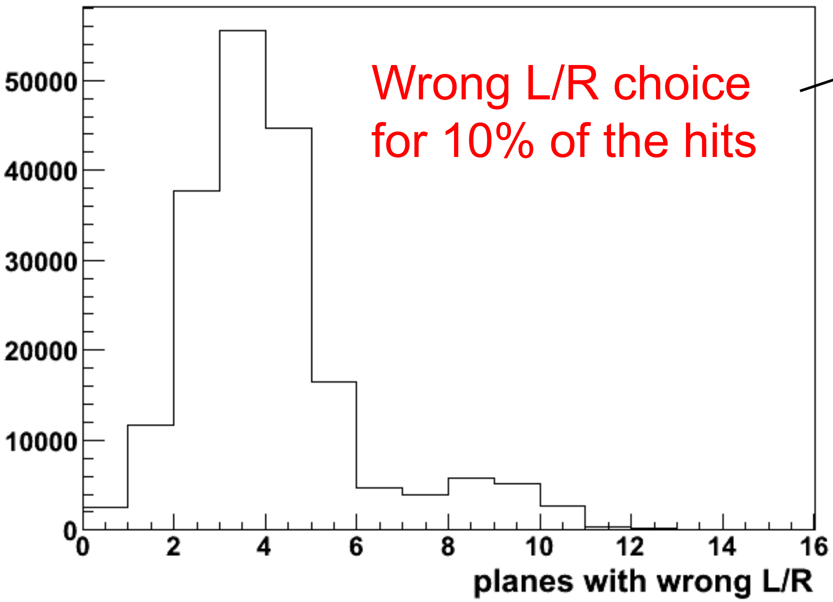




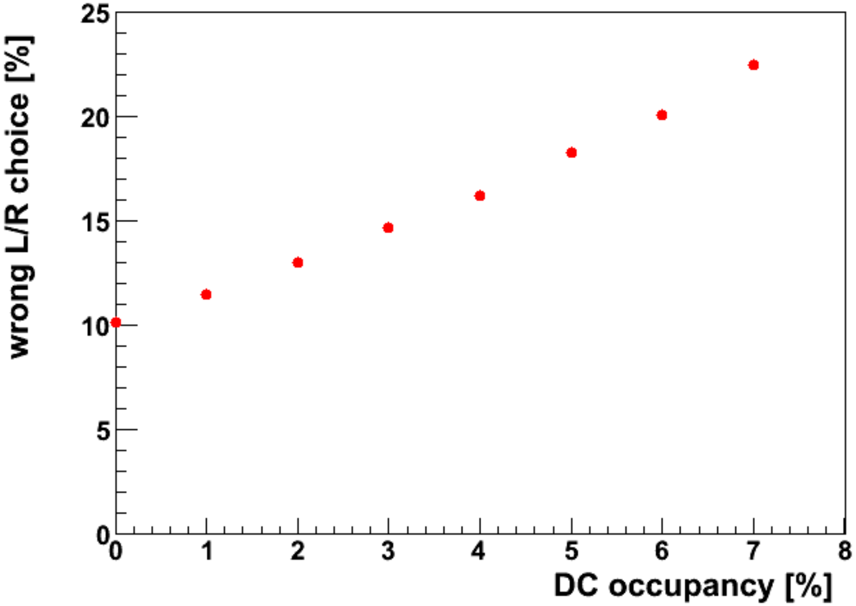


DC – L/R ambiguity

Results (protons at θ = 15°):

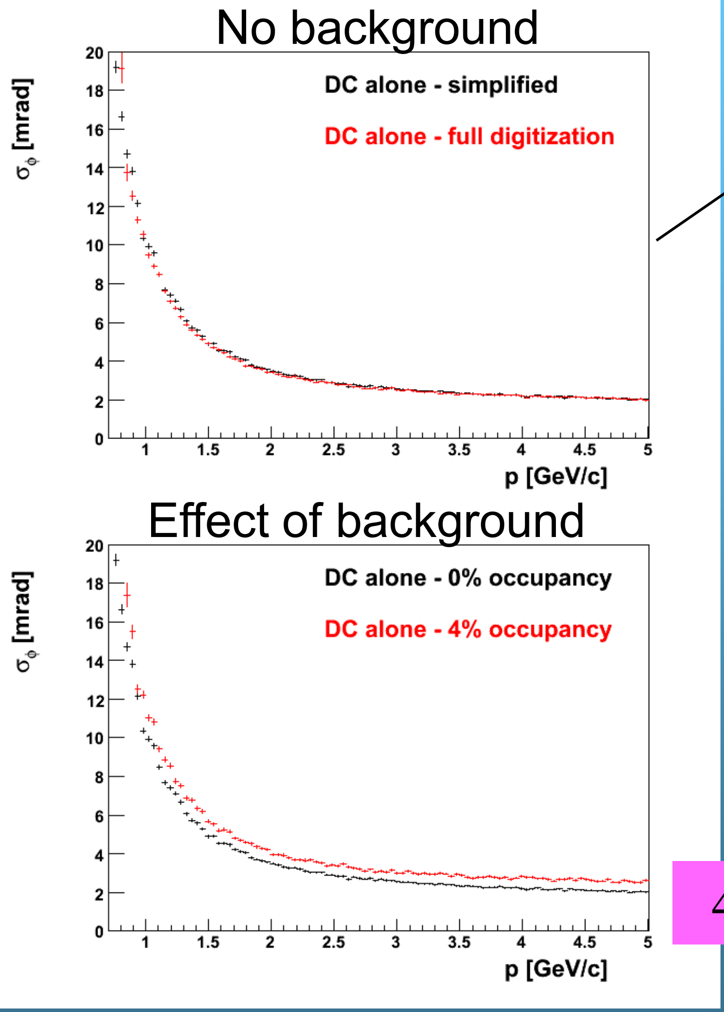


Wrong L/R choice for 10% of the hits → Almost no effect on the resolution

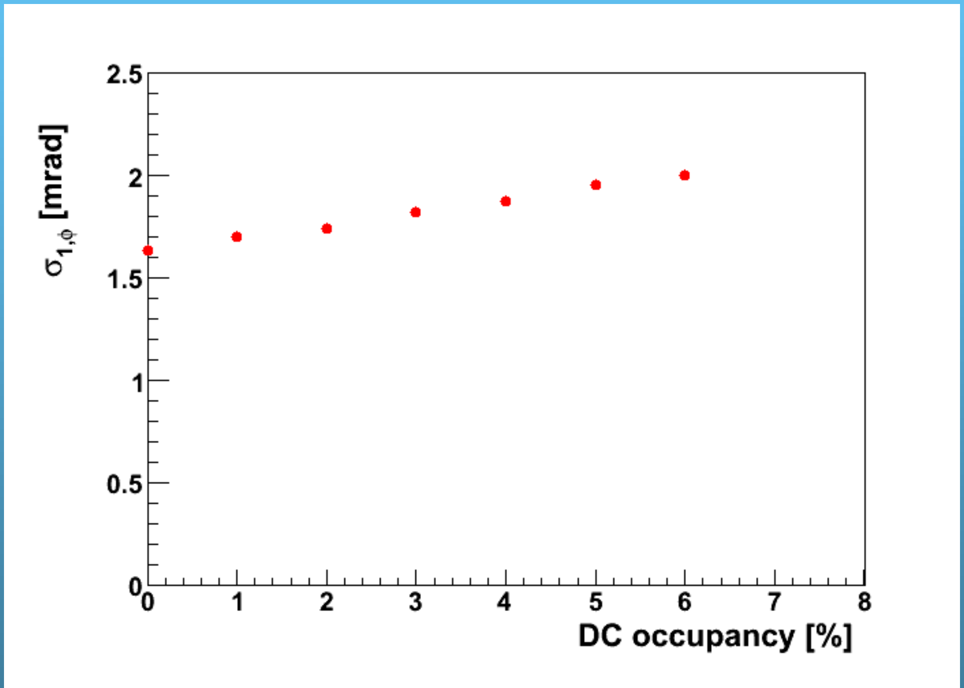


Occupancy degrades the L/R determination (expected!)

DC – resolution & background



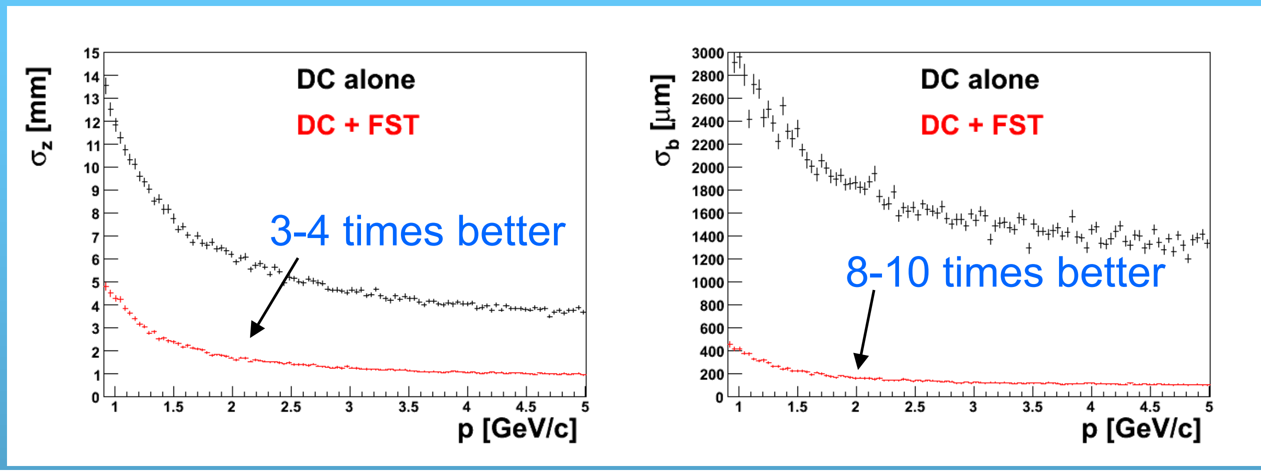
10% mistake on L/R
has indeed no impact

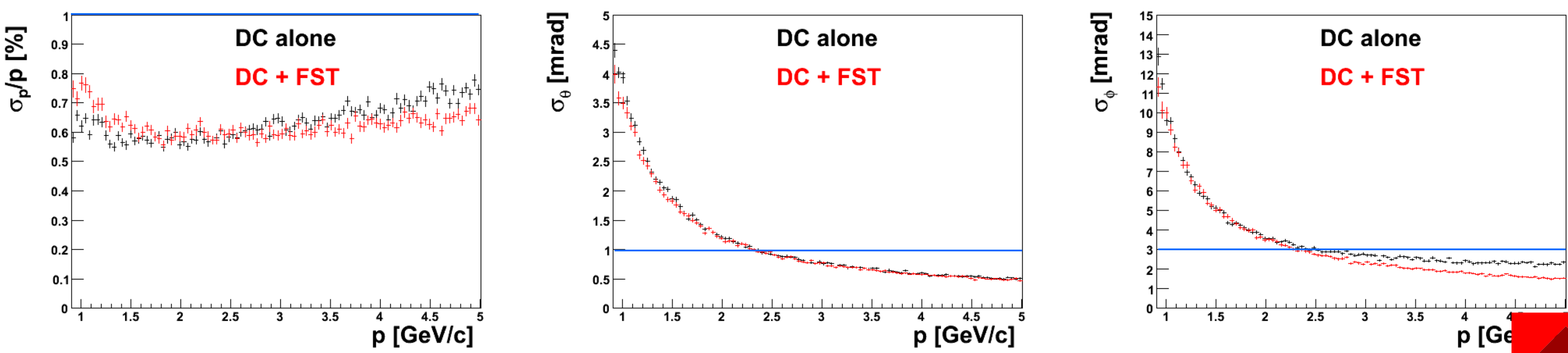


4% occupancy degrades the resolution by ~15%

DC+FST – resolution with protons

(protons at θ = 15°):





Much better vertex resolution with FST (and φ resolution at high p)

KF - initialization

- Central tracking:

  • Measurement at ~constant radius

  $$\rightarrow \quad ^{T} = (\varphi, z, \vec{p})$$

  • Homogeneous field (solenoid)

  $\rightarrow$ estimation of $\vec{x}_0$ with helical track

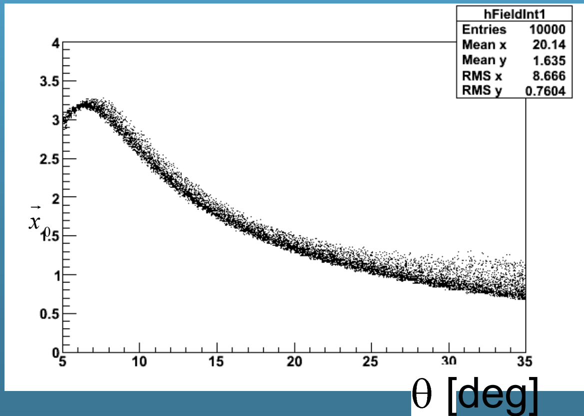
- Forward tracking:

  • Measurement at ~constant z

  $$\rightarrow \quad ^{T} = (x, y, u_x, u_y, q/p)$$

  • Toroidal field

  $\rightarrow$ parameterization of $\int Bdl$



KF – multiple scattering

The multiple scattering can be « easily » taken into account in the KF:

$$\hat{C}_k^{k+1} = \hat{F}_k . \hat{C}_k . \hat{F}_k^T + \hat{Q}_k$$

$$\hat{Q}_{i,j} = \sigma^2(\theta) * \left(\frac{\partial x_i}{\partial \theta_1} \frac{\partial x_j}{\partial \theta_1} + \frac{\partial x_i}{\partial \theta_2} \frac{\partial x_j}{\partial \theta_2} \right)$$

Forward

$$\hat{Q} = \sigma^2(\theta)(1 + x'^2 + y'^2) \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (1 + x'^2) & x'y' & 0 \\ 0 & 0 & x'y' & (1 + y'^2) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Central

$$\hat{Q} = \sigma^2(\theta) \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p_y^2 + p_z^2 & -p_x p_y & -p_x p_z \\ 0 & 0 & -p_x p_y & p_x^2 + p_z^2 & -p_y p_z \\ 0 & 0 & -p_x p_z & -p_y p_z & p_x^2 + p_y^2 \end{pmatrix}$$

Socrat - Misalignments

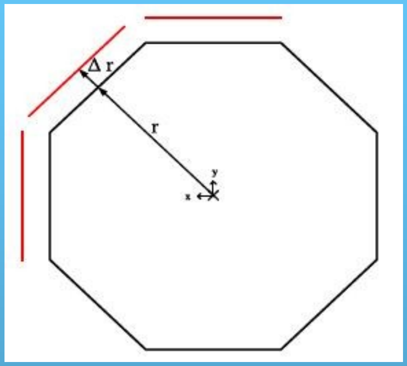
Socrat allows the study of misalignments in various parameters

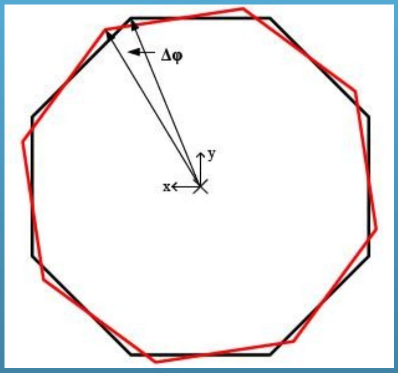
- For each DC region:
  * Distance to the target (~Δz)
  * Tilt around the x_i axis (nominal: 25°)
  * Position of the 1st wire (~Δx)

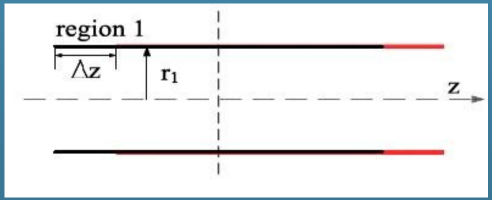
- For each FST superlayer:
  * Azimuthal position (Δφ)
  * Position along the beam axis (Δz)

- For each magnet (solenoid & torus):
  * Position in space (Δx, Δy, Δz)

- For each BST superlayer:
  * Distance to the beam axis (ΔR)
  * Azimuthal position (Δφ)
  * Position along the beam axis (Δz)







Socrat – Generation routine

The tracking code has a routine to generate its own events:

- taking into account the Mult. Scat. (not from air)
- using the available field maps
- with full digitization (for Si and DC)
- with uncorrelated background
- in case of central & forward tracking, tracks are generated at the same vertex position

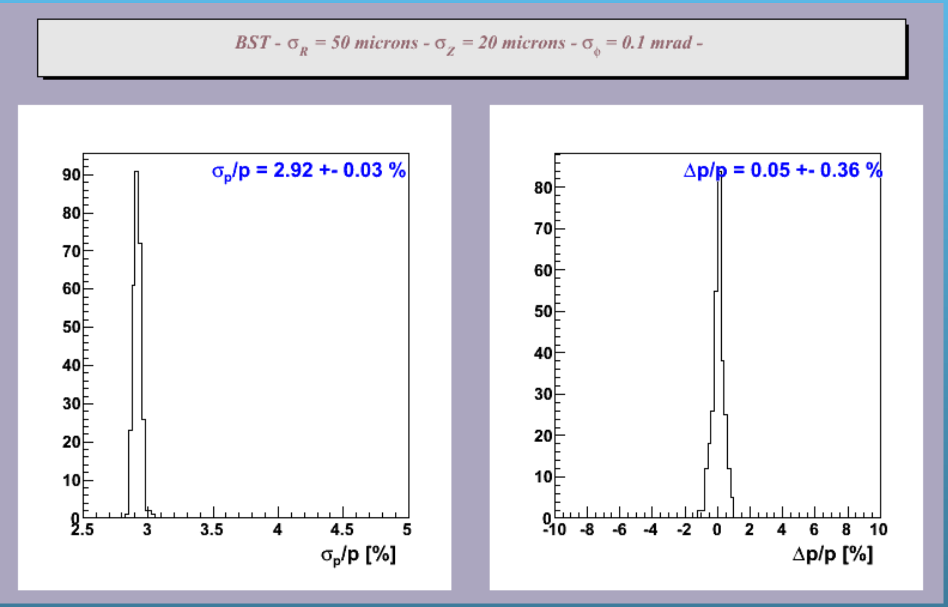
Much faster than Geant4 (no physics!)

Misalignment studies

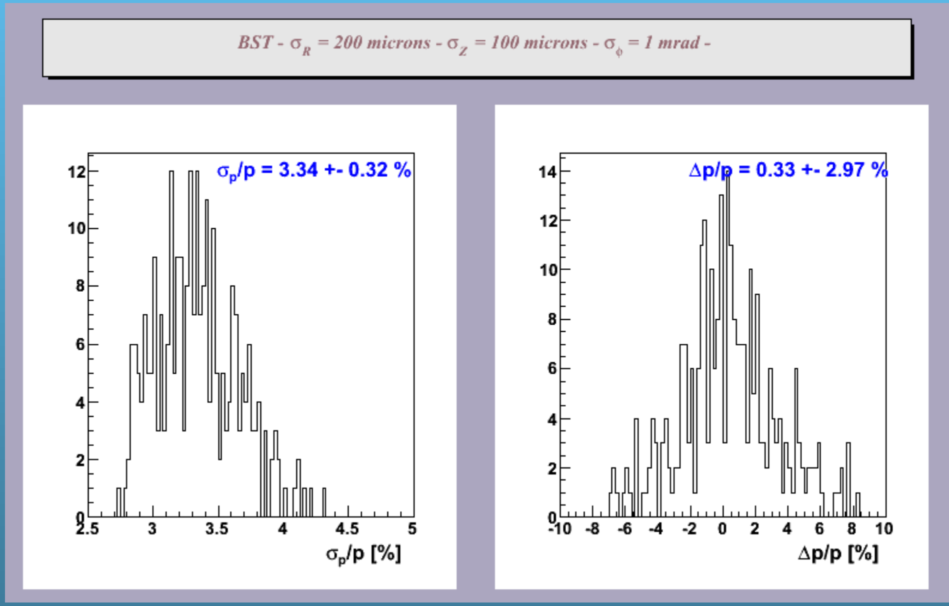
Socrat can generate misalignments in BST, FST, DC, magn. fields

Ex: effect of BST misalignments on momentum

ΔR=50 microns, Δz=20 microns, Δφ=0.1 mrad



ΔR=200 microns, Δz=100 microns, Δφ=1 mrad



⇒ Give constraints on the accuracy of detectors alignment