Luminosity detector software

Anastasia Karavdina

on behalf of the luminosity detector group

September 11, 2012

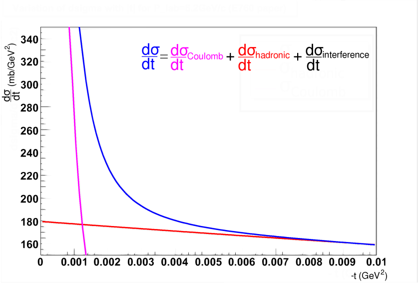
Outline

Outline

Method

$p\bar{p}$ elastic scattering

- coulomb part: can be calculated from QED
- hadronic part: measurement+models



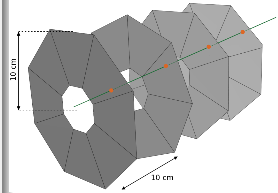
measurement at small momentum transfer
$t = 2p^2_{CM}(1 - cos\theta_{cm}) \rightarrow$ small θ

Design

- measurement at small θ (3-8 mrad)
- position downstream from IP ($\sim$ 11 m)
- placed inside vacuum to minimize $\bar{p}$ scattering

Original design

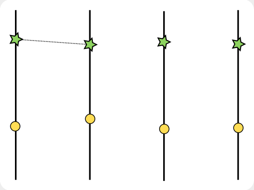
- 4 silicon planes
- distance between planes 10 cm
- each plane contains 8 double sided strip sensors (thickness 150-300 μm)
- strips with 50 μm pitch and 45° stereo angle



Outline

Track search

Track-Following



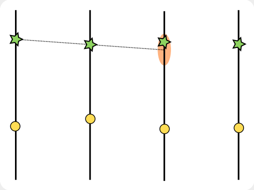
- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

Cellular Automata

- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

Track search

Track-Following



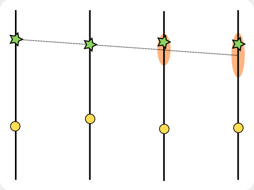
- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

Cellular Automata

- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

Track search

Track-Following



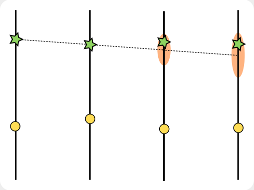
- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

Cellular Automata

- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

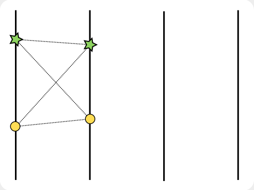
Track search

Track-Following



- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

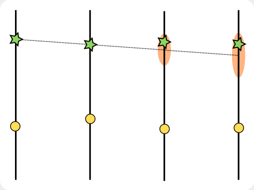
Cellular Automata



- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

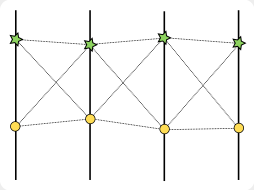
Track search

Track-Following



- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

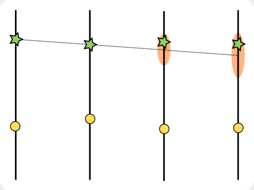
Cellular Automata



- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

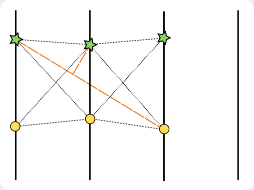
Track search

Track-Following



- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

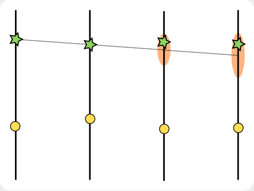
Cellular Automata



- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

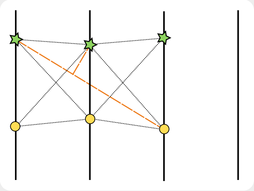
Track search

Track-Following



- build combinations between 1st and 2nd planes
- look for hits on 3rd plane inside corridor

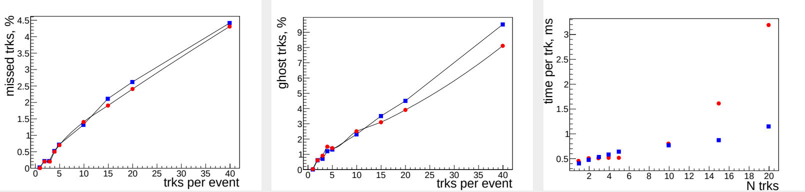
Cellular Automata



- build combinations between hits on pairs of planes (**cells**)
- search for neighboring **cells**
- *missing plane* track search automatically included

Track search

algorithms performance



Cellular Automata vs. Track-Following

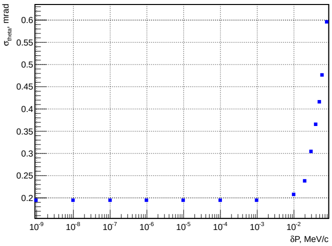
- Track-Following faster for high multiplicity of tracks, but Cellular Automata gives less missing and less ghost tracks
- For low multiplicity of tracks both algorithms show the same performance

Main reason for ghost tracks is fake hits due to double strip structure of sensors

Back-propagation to IP

Calculation of track parameters near interaction point with GEANE

- GEANE helps to calculate track parameters and their errors inside magnetic field
- for all reconstructed tracks $\bar{p}$ assumption with momentum equal P_{beam} is used

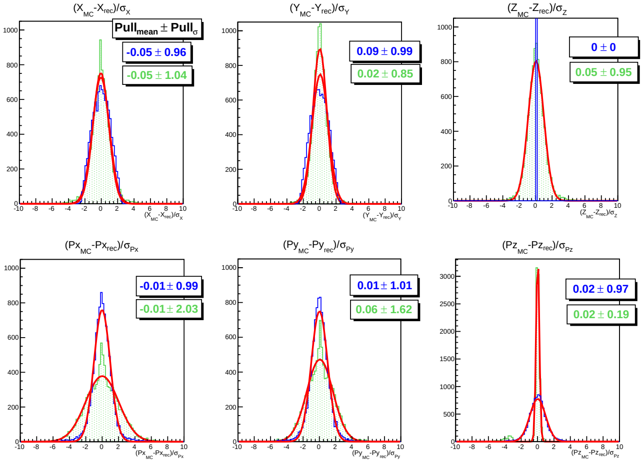


no influence on θ resolution for expected beam uncertainty (10^{-3} GeV/c)

Back-propagation to IP

Problem with pull distributions

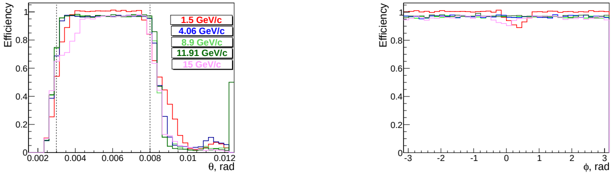
near luminosity detector & near interaction point (P_{beam}=1.5 GeV/c)



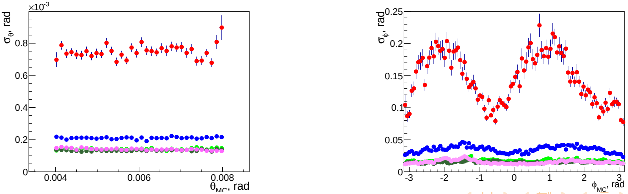
Track reconstruction performance

simulation with beam pipe (v.201112)

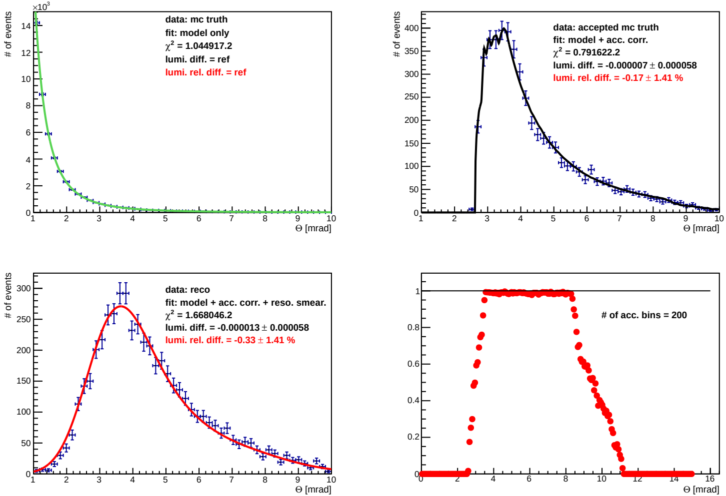
Efficiency



Resolution



Luminosity fit



Outline

Approach

- tracking system without PID and momentum measurement $\rightarrow$ knowledge of signal and background contribution is important

Strategy

- Generation of background with DPM:
  $0.0018 < \theta < 2\pi$ rad, $2 \cdot 10^7$ events
  $P_{beam} = 1.5, 4.06, 8.9, 11.91, 15$ GeV/c
- Track reconstruction in the luminosity detector:
  $\bar{p}$ assumption
- Comparison of reconstructed and MC information
  (for scattered $\bar{p}$ and bkg signal)

Background channels and particles

15 GeV/c

channel	ratio to $\bar{p}^{el}$, %
$p\bar{p}$	100
$p\bar{p}\pi^0$	0.8
$p\bar{p}\pi^-\pi^+$	0.63
$p\bar{p}\pi^-\pi^+\pi^0$	0.55
$n\bar{p}\pi^+\pi^0$	0.46
$n\bar{p}2\pi^+\pi^-\pi^0$	0.19
$n\bar{p}\pi^+2\pi^0$	0.12
$p\bar{p}\pi^-\pi^+\gamma$	0.06
$p\bar{p}\pi^-\pi^0 2\gamma$	0.05
$p\bar{p}\pi^-\pi^+3\pi^0$	0.04
...	
Total	3.72

particle	% of tracks
$\bar{p}$	103.4
π^-	0.17
e^+	0.11
e^-	0.1
π^+	0.01
K^-	0.008
p	0.007
K^+	0.0002

90.4% of bkg = $\bar{p}$ tracks

1.5 GeV/c

channel	ratio to $\bar{p}^{el}$,%
$p\bar{p}$	100
$\pi^+\pi^-\pi^0$	0.005
$\pi^+\pi^-2\pi^0$	0.003
$2\pi^+2\pi^-\pi^0$	0.003
$2\pi^+2\pi^-$	0.002
$\pi^+\pi^-$	0.001
$p\bar{p}\pi^0$	0.001
$p\bar{p}\pi^+\pi^-$	0.001
$2p\bar{p}\pi^-$	0.001
...	
Total	0.025

particle	% of tracks
$\bar{p}$	100.01
π^-	0.02
e^+	0.0002
e^-	0.0003
π^+	0.04
p	0.002

58.6% of bkg = $\bar{p}$ tracks

Background channels and particles

15 GeV/c

channel	ratio to $\bar{p}^{el}$, %
$p\bar{p}$	100
$p\bar{p}\pi^0$	0.8
$p\bar{p}\pi^-\pi^+$	0.63
$p\bar{p}\pi^-\pi^+\pi^0$	0.55
$n\bar{p}\pi^+\pi^0$	0.46
$n\bar{p}2\pi^+\pi^-\pi^0$	0.19
$n\bar{p}\pi^+2\pi^0$	0.12
$p\bar{p}\pi^-\pi^+\gamma$	0.06
$p\bar{p}\pi^-\pi^0 2\gamma$	0.05
$p\bar{p}\pi^-\pi^+3\pi^0$	0.04
...	
Total	3.72

particle	% of tracks
$\bar{p}$	103.4
π^-	0.17
e^+	0.11
e^-	0.1
π^+	0.01
K^-	0.008
p	0.007
K^+	0.0002

90.4% of bkg $= \bar{p}$ tracks

1.5 GeV/c

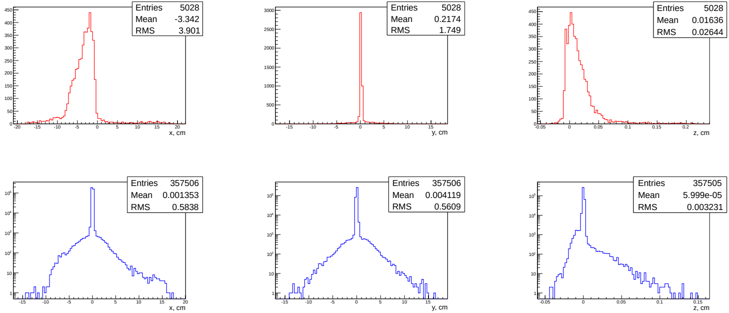
channel	ratio to $\bar{p}^{el}$,%
$p\bar{p}$	100
$\pi^+\pi^-\pi^0$	0.005
$\pi^+\pi^-2\pi^0$	0.003
$2\pi^+2\pi^-\pi^0$	0.003
$2\pi^+2\pi^-$	0.002
$\pi^+\pi^-$	0.001
$p\bar{p}\pi^0$	0.001
$p\bar{p}\pi^+\pi^-$	0.001
$2p\bar{p}\pi^-$	0.001
...	
Total	0.025

particle	% of tracks
$\bar{p}$	100.01
π^-	0.02
e^+	0.0002
e^-	0.0003
π^+	0.04
p	0.002

58.6% of bkg $= \bar{p}$ tracks

Background suppression

How to distinguish background from signal?



Any dedicated cut to reduce background has significant signal inefficiency

Possible solution: fit signal and bkg together
(with Kernel Density Estimation as p.d.f for bkg)

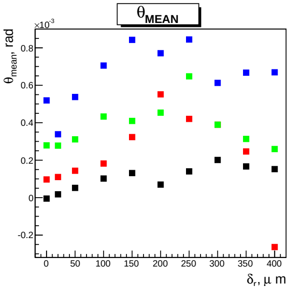
Outline

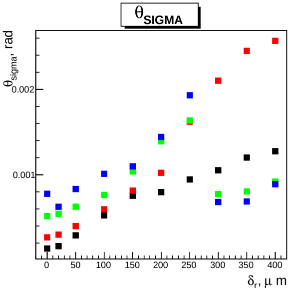
Impact of sensors misalignment
on θ resolution

Each sensors has 6 d.o.f: 3 translation (δ_r) and 3 rotation (δ_α)

δ_r=0, 20, 50, 100, 150, 200, 250, 300, 350, 400 μm

δ_α=0, 3, 6, 9 mrad

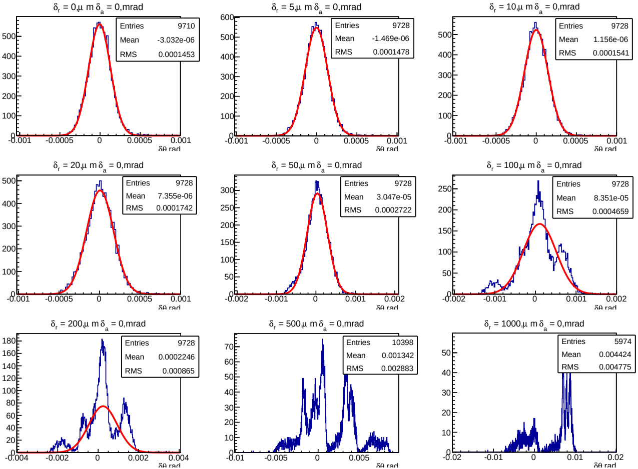




P_{beam}=11.91 GeV/c:

for $\delta_r \sim 100\ \mu m$ ($\delta_\alpha = 0$ mrad) θ resolution become 5 times worse!

Impact of sensors misalignment
on θ resolution (P_{beam}=11.91 GeV/c)



Software alignment

Fitting the residuals (Alignment in one step)

two different types of fitted parameters:

- Related to the track properties determined by the fit. different for each tracks → LOCAL parameters
- Parameters which are depending on the detector position. Alignment parameters → GLOBAL parameters

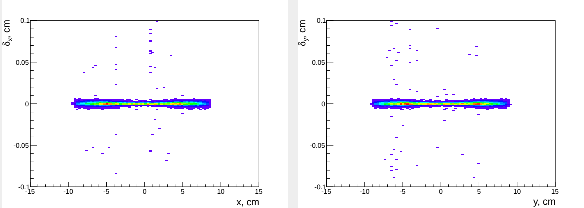
To get a good accuracy significant number of tracks is needed
Matrix size of the system equation: $n_{tot} = n_{loc} \cdot n_{tracks} + n_{gl}$

Millepede algorithm can resolve such problems
And it was used for VELO at LHCb (arXiv:0807.3532)

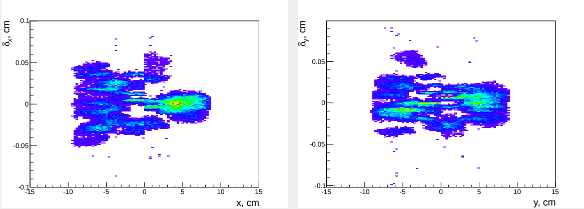
Alignment Results

Residuals between reconstructed hit and track (P_{beam}=11.91 GeV/c)

Ideal case



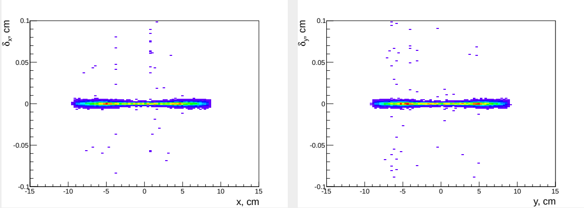
Misaligned sensors



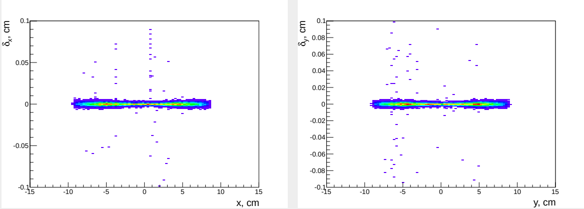
Alignment Results

Residuals between reconstructed hit and track (P_{beam}=11.91 GeV/c)

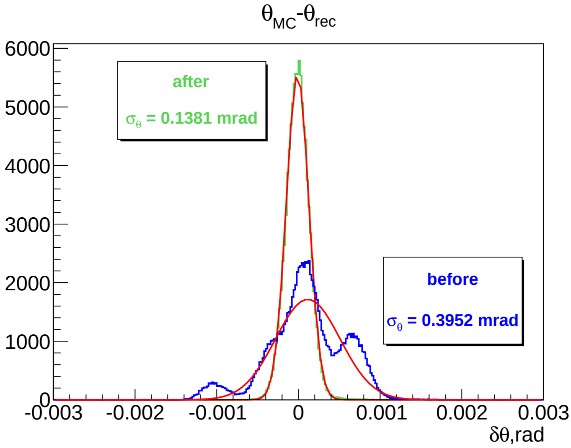
Ideal case



Misaligned sensors



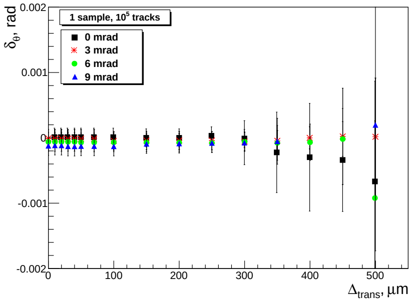
Sensors alignment for translation misalignment $\sim 100\mu m$ (P_{beam}=11.91 GeV/c)



Without sensor misalignment resolution 0.1361 mrad

Sensors alignment

sensitivity to misalignment scale (11.91 GeV/c)



Limits

For translation 300 μm, for rotation 6 mrad

Expectation from mechanical point

translation 200 μm, rotation 3 mrad

Results & Plans

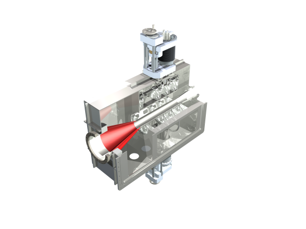
Reconstruction

- Track reconstruction chain is tested on original design
- We have another design with HV-MAPS as sensors
    - 50 μm sensors 2x4 cm^2 and 2x6 cm^2 on both sides
    - 200 μm CVD diamond for heat conduction
    - Length 1m. Halfs diameter 30 cm.

Results & Plans

Reconstruction

- Track reconstruction chain is tested on original design
- We have another design with HV-MAPS as sensors



- 50 μm sensors 2x4 cm^2 and 2x6 cm^2 on both sides
- 200 μm CVD diamond for heat conduction
- Length 1m. Halfs diameter 30 cm.

Results & Plans

Background study

- Background studies are done with DPM generator
- Amount of bkg varies from 0.03% up to 4%
- We are keeping in mind Kernel Density Estimator as a tool for bkg p.d.f description

Alignment

- Software alignment is under development
- Still many things to do here (half planes alignment, studies of required statistics for different beam energies...)

Results & Plans

Background study

- Background studies are done with DPM generator
- Amount of bkg varies from 0.03% up to 4%
- We are keeping in mind Kernel Density Estimator as a tool for bkg p.d.f description

Alignment

- Software alignment is under development
- Still many things to do here (half planes alignment, studies of required statistics for different beam energies...)

Thank you for your attention!

Kernel Density Estimation

Estimating a Signal In the Presence of an Unknown Background, W. Rolke and A. Lopez (arXiv:1112.2299)

"Non-parametric" density estimation

- X_1, .., X_n – observations from some unknown density f.
- K – continuous, non-negative and symmetric function with $K(x) \to 0$ as $x \to \pm\infty$
  (Often K is chosen to be p.d. itself: Gaussian density)

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x-X_i}{h})$$

- h – tuning parameter called bandwidth.
  $\to$ constant or adaptive bandwidth h_x

Semi-parametric fitting

Function for maximum likelihood

$f(x;\alpha,\theta) = (1-\alpha)f_B(x) + \alpha f_S(x;\theta)$

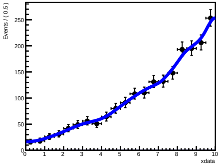
for $f_B(x)$ (estimation of bandwidth and density) one need pure sample of background events
:(

RooFit contains RooKeysPdf class:

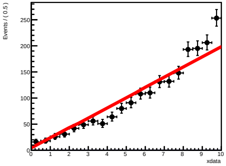
A one-dimensional kernel estimation p.d.f which model the distribution of an arbitrary input dataset as a superposition of Gaussian kernels
:)

Example

Bkg fit by adaptive kernel estimation pdf



Bkg fit by polinom a*x



Sig(x) = Landay(3.5, 1)⊗Gaus(0,0.1)

Bkg(x) = x+0.001x^2+0.01x^3

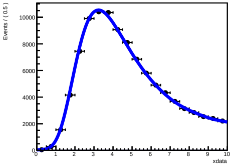
Data(x)= Sig(x) + Bkg(x)

sig/(sig+bkg)=0.95

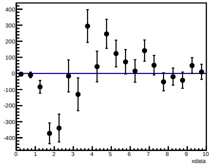
Kernel: 9.4966e-01 ± 2.35e-03 (<σ)

Lin:9.3780e-01 ± 2.84e-03 (4σ)

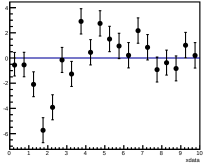
Sig+Bkg data (Kernel estim)



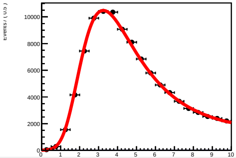
Residual Distribution for Kernel estim



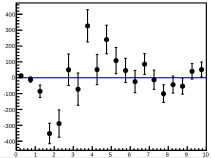
Pull Distribution for Kernel estim



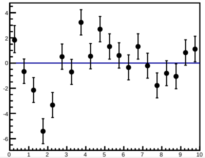
Sig+Bkg data (lin func)



Residual Distribution for function a*x

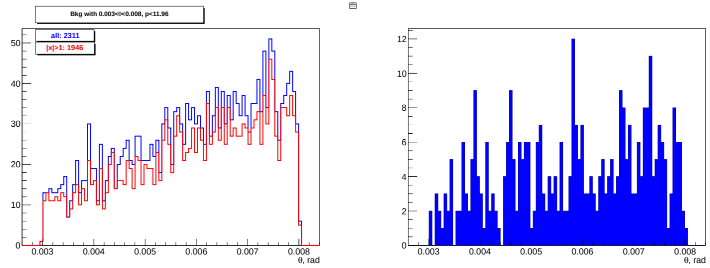


Pull Distribution for function a*x



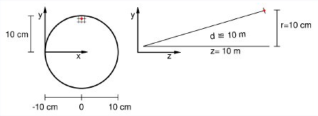
Sample of "pure" background

- Shape of background is not fixed (important for us, because it could differ for different P_{beam})
- But to fit it background sample is needed.



Small signal fraction to bkg sample doesn't change shape too much

$\Delta\phi$ vs. $\Delta\theta$



$$\cos\phi = \frac{x}{r},\ \Delta\cos\phi = \frac{\Delta x}{r}\sqrt{1+\frac{x}{r}}$$

$$\cos\theta = \frac{z}{d},\ \Delta\cos\theta = \frac{\Delta x}{d}\sqrt{1+\frac{z}{d}}$$

$$\Delta y = \Delta x$$

$$r = \sqrt{x^2+y^2},\ \Delta r = \Delta x$$

$$\Delta z \simeq \Delta x$$

$$\Delta\cos\theta / \Delta\cos\phi \simeq \frac{r}{d} = 10^{-2}$$

Elastic Cross Section

$$\frac{d\sigma}{dt} = \frac{d\sigma_C}{dt} + \frac{d\sigma_{int}}{dt} + \frac{d\sigma_H}{dt}$$

with

$$\frac{d\sigma_C}{dt} = \frac{4\pi\alpha_{EM}^2 G^4(t)}{\beta^2 t^2}$$

$$\frac{d\sigma_{int}}{dt} = \frac{\alpha_{EM}\sigma_{Total}}{\beta|t|} G^2(t) e^{\frac{1}{2}Bt}(\rho cos(\delta) + sin(\delta))$$

$$\frac{d\sigma_H}{dt} = A_1 \cdot \left[e^{t/2t_1} - A_2 \cdot e^{t/2t_2}\right]^2 + A_3 \cdot e^{t/t_2}$$

A few words about generated statistic

$2 \cdot 10^7$ events with DPM were generated

$$\frac{dN}{dt} = \sigma \cdot L \rightarrow \tilde{N} = \sigma \cdot \tilde{L} \cdot t$$

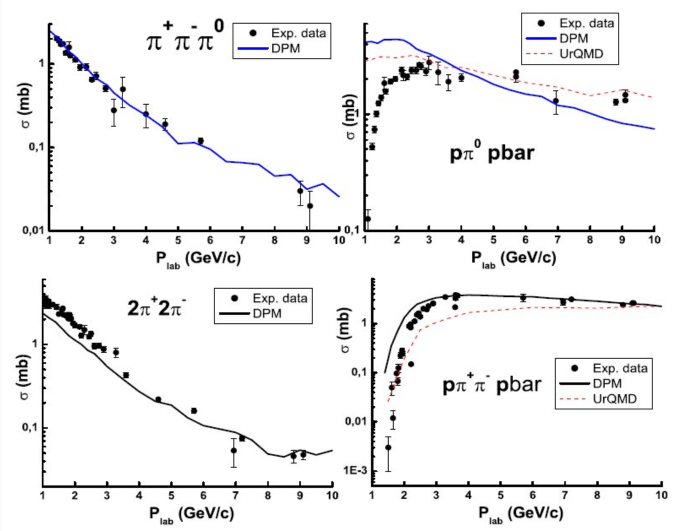
$\tilde{L} = 2 \cdot 10^{32} \frac{1}{cm^2 s}$ (High lumi mode)

$\sigma = (100 - 51)$ mb

$t = \frac{2 \cdot 10^7}{2 \cdot 10^{32} (100-51) \cdot 10^{-27}} = (1 - 2)$ s

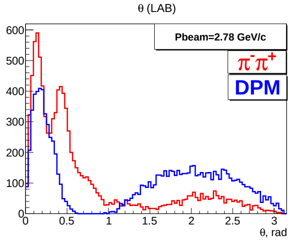
DPM generator

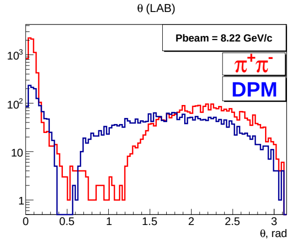
Total cross-sections for different channels



DPM generator

Differential cross-sections

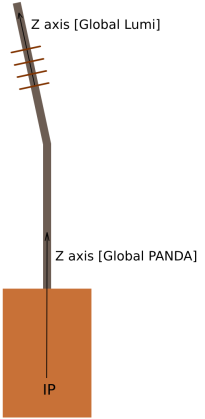




comparison with dedicated $p\bar{p} \rightarrow \pi^+\pi^-$ generator (M.Zambara & D. Khaneft)

Hit reconstruction

Hit position (x,y,z) reconstruction from digital information



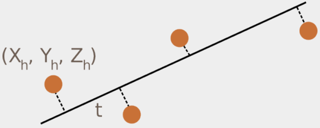
- sensor & strip number
- (x,y,z) local sensor
- Global PANDA frame
- Global Lumi frame

result of sensor alignment will be applied at this stage

Track fit

3D straight line fit by the least-squares method

$$x = p_0 + p_1 t$$
$$y = p_2 + p_3 t$$
$$z = p_4 + p_5 t$$



$$p_4 = Z_0$$
$$p_5 = \sqrt{1 - p_1^2 - p_3^2} \quad t_{min} = p_1(x_h - p_0) + p_3(y_h - p_2) + p_5(z_h - p_4)$$

$$\chi^2 = \frac{(x_h - (p_0 + p_1 t_{min}))}{\sigma_x^2} + \frac{(y_h - (p_2 + p_3 t_{min}))}{\sigma_y^2} + \frac{(z_h - (p_4 + p_5 t_{min}))}{\sigma_z^2}$$

Software alignment

Minimizing the residuals (Iterative)

- Fit a large number of tracks and plot the residuals for each sub-detector
- The mean value of each residual distribution should then be centered on the misalignment value
- Correct the sub-detector coordinates from this value
- Fit the track again, plot the residuals,...
  And so on until the residuals central values are stabilized at 0.

Iterations are based on biased track fit results (fit with misaligned detector). Then, even if the method converges, it could leads to biased misalignment values (offsets to real values).

Software alignment

How to deal with large matrix equation?

Millepede algorithm

- Matrix is divided on sub-matrices of dimensions $n_{loc} * n_{loc}$
- C++ implementation is done for LHCb VELO (Knossos)

Track parametrization

$$\begin{cases} x &= a \cdot z + b \\ y &= c \cdot z + d \end{cases}$$

Residuals parametrization

$$\begin{cases} \epsilon_x &= -\Delta_x + y_{hit} \cdot \Delta_\gamma + a \cdot (\Delta_z + x_{hit} \cdot \Delta_\beta + y_{hit} \cdot \Delta_\alpha) \\ \epsilon_y &= -\Delta_y - x_{hit} \cdot \Delta_\gamma + c \cdot (\Delta_z + x_{hit} \cdot \Delta_\beta + y_{hit} \cdot \Delta_\alpha) \end{cases}$$

3 translations (Δ_x,Δ_y,Δ_z) and 3 rotation (Δ_α,Δ_β,Δ_γ) around x, y, z