Analysis of Mille output

CMS Tracker Alignment with Millepede

DESY Summerstudent Program 2008

Michael Prim

Universität Karlsruhe

14th September 2008

Abstract

One of the inner subdetectors at the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC) consists of silicon strip modules. It is called the silicon tracker and it is essential for its resolution to determine the exact position of every module during the alignment process. Otherwise all measurements are biased due to misalignment.

One method to align the detector is called Millepede[1]. It allows to take all sources of information into account to perform a single linear least square fit and all correlations between these information are taken into account.

Millepede is separated into two steps: Mille and Pede. The output of the first step was analyzed in dependence on different track models and material effects.

Contents

1 Introduction

A brief introdcution to the idea of Millepede is given in the following (more details found in [2] chapter 5). A particle track in the CMS tracker typically has more then 20 independent measurements of its position. The five parameters of a helix to describe the track together with a reference plane should be therefore overdetermined. The recorded measurements u_m can be compared to the predictions of the track model u_p. These predictions depend on the track parameters $\boldsymbol{\tau}_j$ and parameters $\boldsymbol{p}$ which describe position, orientation and deformation of the detectors. The normalized residual of track j r_{ij} between the predicted hit and the measured hit i is given by:

$$r_{ij} = \frac{u_{im} - u_{ip}\left(\boldsymbol{\tau}_j, \boldsymbol{p}\right)}{\sigma_i}$$

where σ_i is the uncertainty of the measurement. To achieve the best agreement between the track model and the measured hits one tries to minimize these residuals for all tracks. This leads to a χ^2 function:

$$\chi^2\left(\boldsymbol{\tau}, \boldsymbol{p}\right) = \sum_j \sum_i r_{ij}^2\left(\boldsymbol{\tau}_j, \boldsymbol{p}\right)$$

where $\boldsymbol{\tau}$ is the vector of all track parameters. To minimize this function it is first linearized to simplify the problem:

$$\chi^2 = \sum_j \sum_i r_{ij}^2\left(\boldsymbol{\tau}_j, \boldsymbol{p}\right) \simeq \sum_j \sum_i \frac{1}{\sigma_i^2}\left(u_{im} - u_{ip}\left(\boldsymbol{\tau}_{j0}, \boldsymbol{p}_0\right) + \frac{\partial u_{ip}}{\partial \boldsymbol{p}}\delta\boldsymbol{p} + \frac{\partial u_{ip}}{\partial \boldsymbol{\tau}_j}\delta\boldsymbol{\tau}_j\right)^2$$

where $\boldsymbol{p}_0$ are initially assumed geometry parameters and $\boldsymbol{\tau}_{j0}$ are initially assumed track parameters. The geometric correction parameters $\delta\boldsymbol{p}$ are the alignment parameters, which are also called global parameters, since they are not related with a single track. The track correction parameters $\delta\boldsymbol{\tau}_j$ are specific only for a single track and are also called local parameters.

The minimization of this linearized function is done during the Mille step. As output one obtains the residuals r_i for hit i and their corresponding errors σ_{r_i}. Assuming ideal geometry, perfect understanding of material and other effects and ideal minimization the distribution of the scaled residual $\frac{r_i}{\sigma_{r_i}}$ should follow a Gaussian with mean 0 and width 1. This is of course not the case and this analysis might help to understand systematic errors.

The Pede step then uses the Mille output to determine the global parameters and so determine the true position of every module within the tracker.

2 Analysis Setup

In this chapter the used setup in the CMS software framework (CMSSW) and a short description of a self developed tool `mille2root` will be given.

2.1 CMSSW

2.1.1 Software framework

The version 2-0-11 of the CMSSW was used. Additionally there have been used the following packages in a newer version as at the time of the 2-0-11 release:

```
cvs co -r V03-01-06    Alignment/CommonAlignment
cvs co -r V00-27-14-17 Alignment/CommonAlignmentProducer
cvs co -r V00-16-01    Alignment/MillePedeAlignmentAlgorithm
cvs co -r V00-11-01    Alignment/ReferenceTrajectories
```

They include some bugfixes and new features such as the new `DualTrajectory` for magnetic field on.

To find systematic errors there are further information on the track needed as only the local parameters. These so called special data is usually not written into the Mille output because it is not needed for the Pede step. To write these special data there were made some small changes to the

```
Alignment/MillePedeAlignmentAlgorithm/src/MillePedeAlignmentAlgorithm.cc
```

which are documented in the Appendix A and should be also found in future releases of the MillePedeAlignmentAlgorithm package.

2.1.2 Data samples

All studies have used MC simulated data for cosmic muons with magnetic field on at $3.8T$. The sample was called `CSA08 - TkCosmicBON`. There was a cut set to only select cosmics with a transverse momentum of at least $1GeV$. And only the Tracker Inner Barrel (TIB) and the Tracker Outer Barrel (TOB) have been studied.

The data was simulated for the 2-0-X CMSSW version and due to improvement and changes in the simulation software the results may differ in newer versions or the data might be even incompatible to newer CMSSW releases.

2.2 mille2root

The Mille output is written into a binary file. The ROOT macro `mille2root` provides easy access to the data by converting it into a ROOT file. In a second step the macro creates all plots to analyze the output. The whole `mille2root` package consists of three files:

```
mille2root.cc
TrackerHit.cc
TrackerHit.h
```

Both TrackerHit files provide a class called `TrackerHit` which is able to save all data of a single hit and can be stored in an ROOT tree. The macro creates a shared library out of these files which can be also included in other projects if necessary in the future. The mille2root macro file consists of six methods:

```
Float_t calc_mean(vector<Float_t> &res)
Float_t calc_weighted_mean(vector<Float_t> &res, vector<Float_t> &reserr)
Float_t calc_median(vector<Float_t> &res)
void mille2root_conversion(string &filename, const Int_t &max_records)
void mille2root_plots(string &filename)
void mille2root()
```

The first three methods calculate mean, weighted mean and median of a given vector. The forth method converts a Mille binary file into a ROOT file. The fifth method creates plots to analyze the output. The sixth method is the automatically started main method which calls the conversion and plot method. It is the only one the user has to make changes. There the user can set the location and filename of the Mille binary file and can comment out for example the conversion if one only wants to create plots of an allready existing ROOT file.

The macro is called from the shell prompt via `root -l -b mille2root.cc` and the author recommends to run it in batch mode (`-l -b` parameter). For further questions the author suggests to have a view at the hopefully well commented code.

3 Analysis

As mentioned in the introduction Millepede is a linear model which means there are made some approximations compared to the real world. Outgoing from the reconstructed track in a Kalman fit which includes all material effects and other effects there is made a first approximation by assuming the track to be a helix. There exist several trajectory models to describe the helix in different cases as magnetic field on/off or two body decays.

Two models with magnetic field on have been studied in detail. They are called `ReferenceTrajectory` and `DualTrajectory`. The first one starts the development of the helix at the first hit (for cosmics usually the highest hit) and then goes hit by hit through the tracker. The second one starts at a hit in the middle of the tracker and then develops two helices in opposite directions to the outside and combines them afterwards to a single helix. The starting point is called the reference plane of the helix and together with it's five parameters describes the helix.

Both trajectorys allow to change the development direction from along particle momentum to opposite particle momentum. For the `ReferenceTrajectory` this means beginning at the last hit and developing to the first one. For the `DualTrajectory` this means beginning outside and developing to a common hit in the middle.

The residuals r_i are the difference between measured hit and hit prediction. To simplify and linearize the problem (as shown in the introduction) the prediction is a helix propagation with the initial track parameters from the reference plane. Corrected by the local derivate at every hit. This corresponds to a first order Taylor expansion of the helix at the hit.

Material effects may either influence the track model and therefore the helix or the errors on the hit prediction. While energy loss due to ionization changes the curvature it changes the predicted hit position and therefore influences the residual. Whereas multiple scattering influences the prediction error and is correlated to the inverse of the particle's momentum. The error due to multiple scattering is also effected by the amount of material between hit and reference plane. So it might be useful to change the reference plane.

The scaled residual $\frac{r_i}{\sigma_{r_i}}$ of hit i will be called the pull $pull_i$ in the following. σ_{r_i} mainly consists of the intrinsic module resolution and the prediction error. There are other contributions that are neglected: If there is a particle interaction at the first hit this might lead to a correlation with all following hits. Also the hit is used for the prediction which means a correlation between measured position and predicted position.

As physically expected there where no systematic differences found between development along and opposite momentum. Therefore this report focuses on the differences between the two trajectorys and the influence of material effects.

3.1 Pulls versus particle momentum

The pull $pull_i = \frac{r_i}{\sigma_{r_i}}$ of all hits i has been plotted versus the charged sign particle's transverse momentum p_t. To spread the statistic of the used sample over a wide momentum scale the logarithm of the momentum was taken. This is shown in figure 1 for the `DualTrajectory`.

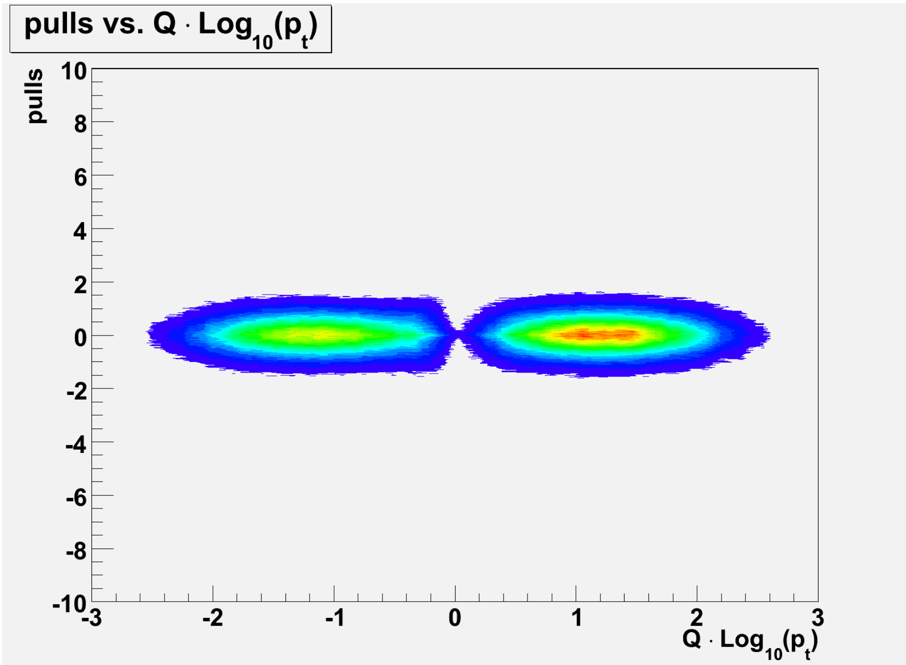

Figure 1: $pull_i$ versus $Q \cdot Log_{10}(p_t)$ for `DualTrajectory`

Afterwards for every bin in x, there was a fit performed to determine mean m_n and sigma σ_n of bin n. The ROOT method `TH2::FitSlicesY()` was used therefore. Additionally the normalization constant of the fit c_n and the χ^2_n was determined. The results are illustrated in figure 2.

The upper left plot shows where the most statistic is available. The lower right plot describes the quality of the fit. As mentioned in the introduction we expect the pulls to have a mean of $m_n \approx 0$ and a sigma of $\sigma_n \approx 1$. The upper right plot shows that this is the case for the mean apart of regions with very low statistics and therefore high statistical fluctuations. Whereas sigma $\sigma_n < 1$ over several orders of magnitude in p_t. It's nearly constant and show only a small dependence on the momentum. $\sigma_n < 1$ means that the error of the residual is overestimated. Only for low momentum the residuals and their errors seam to be well described.

For the `ReferenceTrajectory` the results are shown in figure 3. One can see that σ_n shows a small increase for high momentum and is not that constant as the `DualTrajectory`.

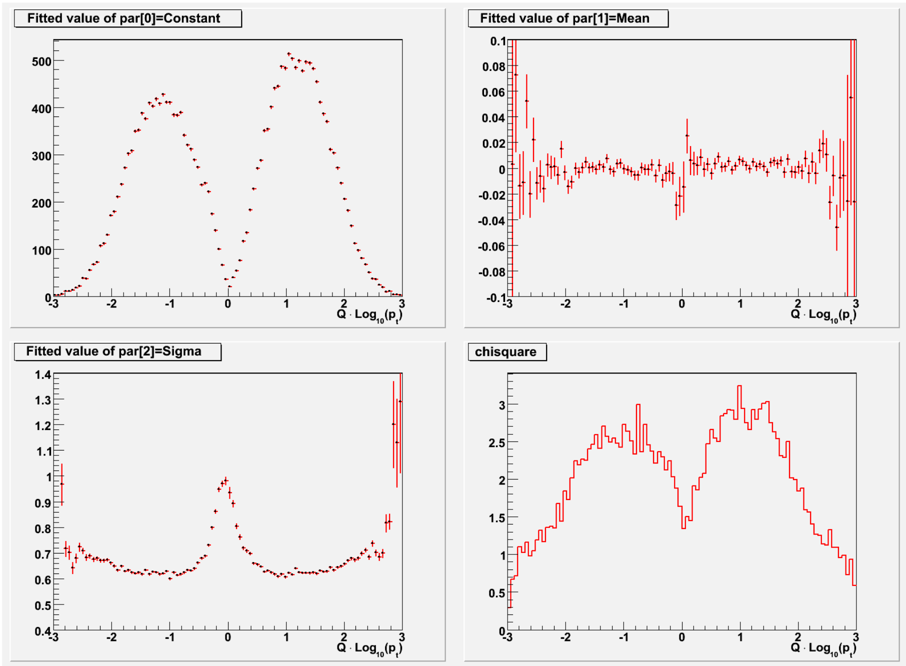

Figure 2: Fitted slices for `DualTrajectory` with normalization constant c_n, mean m_n, sigma σ_n and χ^2_n

It has shown that the mean is for all studied cases around $m_n \approx 0$ and there seem to be no systematics. So in the following chapters only σ_n, which shows a strong dependence on the chosen model, will be plotted.

There is also a strong dependence of the results on the fit. This problem occurs mainly when regarding results with no material effects concerned. In this case the pulls in every single slice where a fit is performed may have a very non Gaussian shape. This means long tails which strongly influence the results in σ_n if they are simply cut off. Because of this it was tried to take these tails into account as far as possible. In general the results show the same shape for different tail cut offs but may differ in the absolute value.

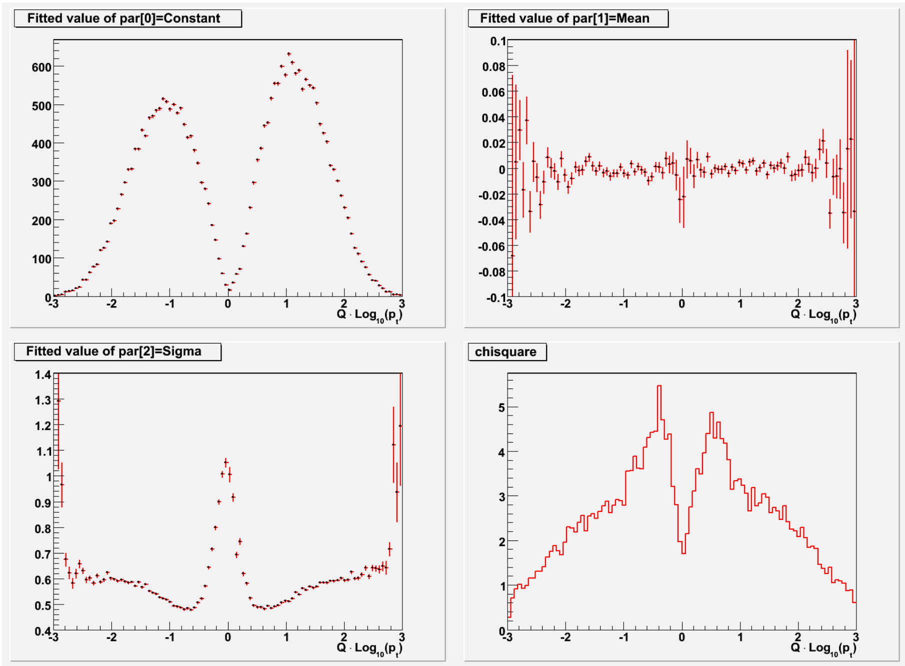

Figure 3: Fitted slices for `ReferenceTrajectory` with normalization constant c_n, mean m_n, sigma σ_n and χ^2_n

3.2 Differences between u and v measured hits

As mentioned in the abstract the tracker consists of silicon strip modules. Some layers in the tracker can not only measure one dimension but two. This is done by installing two modules adjoined and rotating them against each other by a small angle of $100mrad$. These combined two dimensional modules can now measure the v direction along the strips and not only the u dimension that basically measures which strip was hit. The intrinsic resolution of the v direction is about one order of magnitude lower $\mathcal{O}\,(230 - 530\mu m)$ compared to the one in u direction $\mathcal{O}\,(30 - 55\mu m)$

Figures 6 and 7 show that there is a big difference in the characteristics for sigma between u and v measured hits in the case of the `ReferenceTrajectory`. Where u shows only low momentum dependence, v shows a high momentum dependence. For the `DualTrajectory` (Figure 4 and 5) the difference between u and v is not that significant. Both show a low dependence on the momentum. In v there is no peak for low momentum which could be understood as the momentum independence (in first order) of the measurement along the magnetic field.

It is important to mention that the ratio between u and v hits is about $4:1$ and so the increase for high momentum in v (as expected from multiple scattering) is suppressed in the combined plots (Figure 2 and 3).

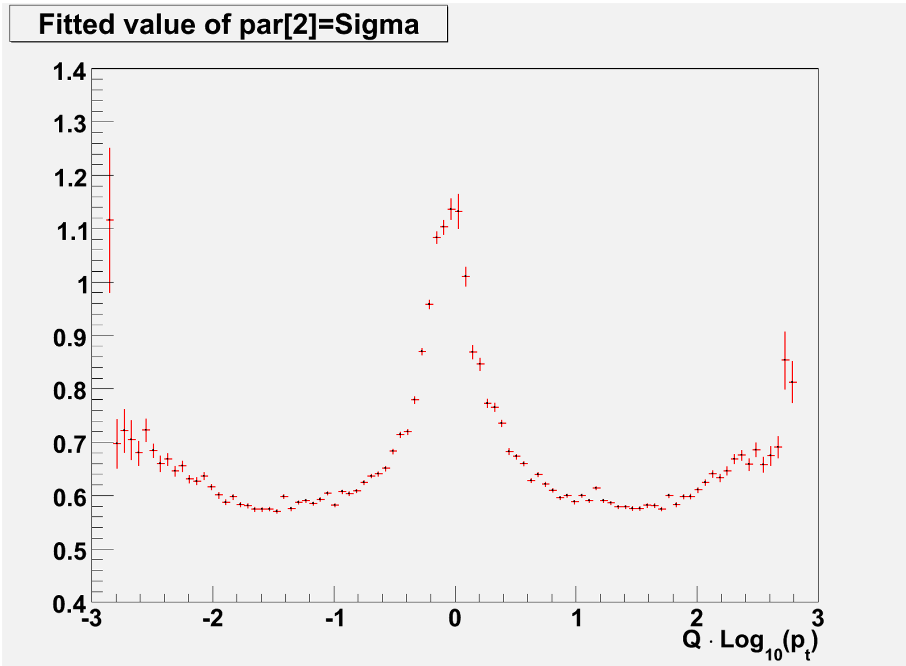

Figure 4: σ_n for `DualTrajectory` in u direction

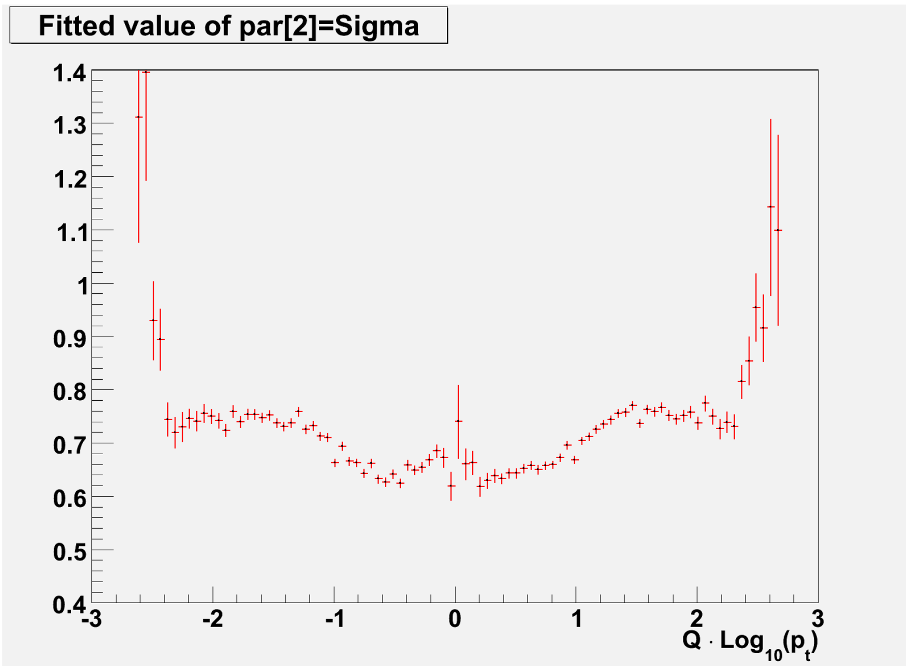

Figure 5: σ_n for `DualTrajectory` in v direction

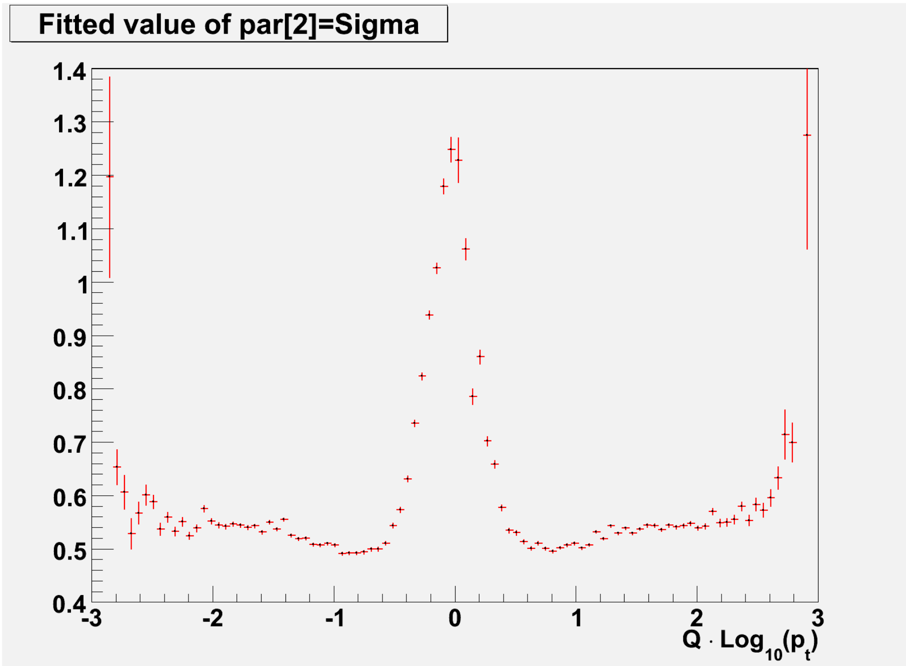

Figure 6: σ_n for `ReferenceTrajectory` in u direction

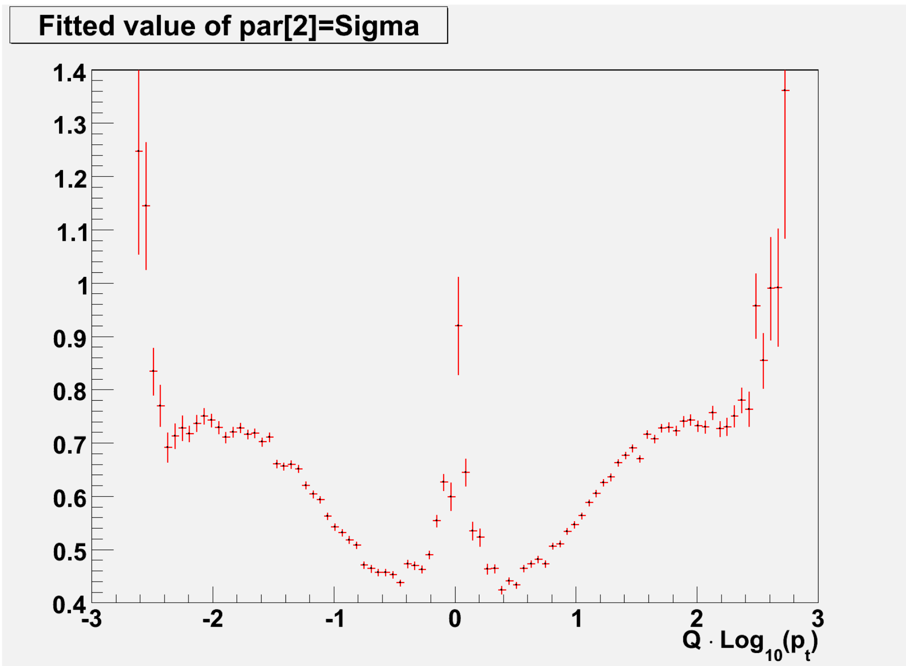

Figure 7: σ_n for `ReferenceTrajectory` in v direction

3.3 Influence of material effects

It is possible to individually switch on/off the consideration of material effects in the trajectory. One can switch off all material effects or take into account either energy loss or multiple scattering. The default mode is a combination of both effects.

As mentioned before the fits get long tails without material effects. Figures 8 and 9 illustrates this problem in the bin at $10GeV$. One can see how the tails get suppressed when using multiple scattering.

If material effects are switched off $\sigma_n > 1$ for `DualTrajectory` (Figure 10) and `ReferenceTrajectory` (Figure 11) as one expects. The material effects are not taken into account and so the error of the residual is estimated to small and σ_n gets big. Only for extreme high momentum $p_t > 100GeV$ is σ_n below 1. For very low momentum σ_n shows some fluctuations and no continuous shape.

As mentioned energy loss has no direct influence on the residual's error but on the residual. If one concerns energy loss σ_n gets a continuous shape for low momentum and is typically bigger then with no material effects (Figure 12 (`DualTrajectory`) and 13 (`ReferenceTrajectory`)). With no change in the errors this means for low momentum that the residuals for the helix with adjusted curvature are typically bigger then with constant curvature. There is no significant change for high momentum.

One expects σ_n to be better when concerning multiple scattering. The error σ_{r_i} gets bigger and therefore σ_n should decrease. Since multiple scattering is the dominating material effect one expects a significant change in the pulls. As shown in figure 14 (`DualTrajectory`) and 15 (`ReferenceTrajectory`) one sees a big improvement towards $\sigma_n \approx 1$ over the complete momentum scale compared to not concerning material effects. In the `ReferenceTrajectory` there seems to be a underestimation of the error for low momentum.

Combining the effects of energy loss and multiple scattering the error of the residuals is influenced by multiple scattering and the residual might change due to energy loss. As shown in figure 17 $\sigma_n \approx 1$ for low momentum in the case of the `ReferenceTrajectory`. Also for the `DualTrajectory` $\sigma_n \approx 1$ for low momentum as shown in figure 16. There is no change for high momentum compared to multiple scattering only when combining both effects.

Therefore the combination of energy loss and multiple scattering increases σ_n for low momentum in the case of the `DualTrajectory` whereas it decreases σ_n for the `ReferenceTrajectory`.

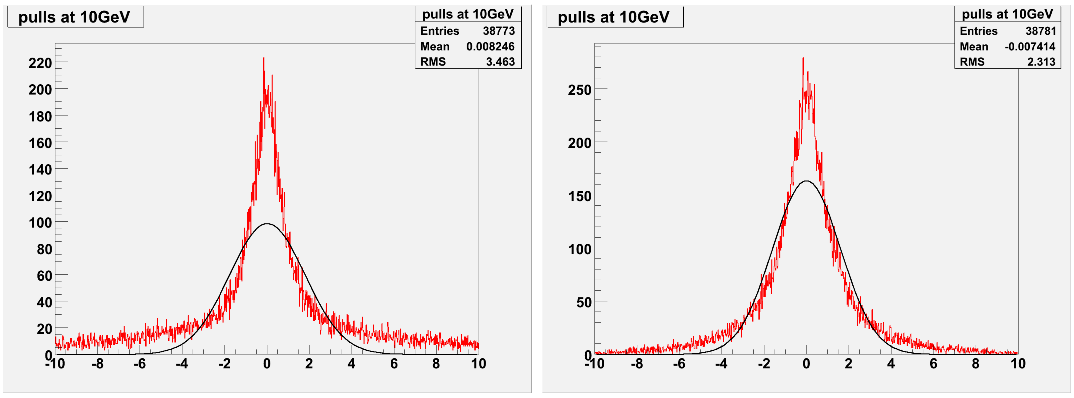

Figure 8: Fitted slice in bin at $10GeV$ for no material effects (left) and with energy loss (right)

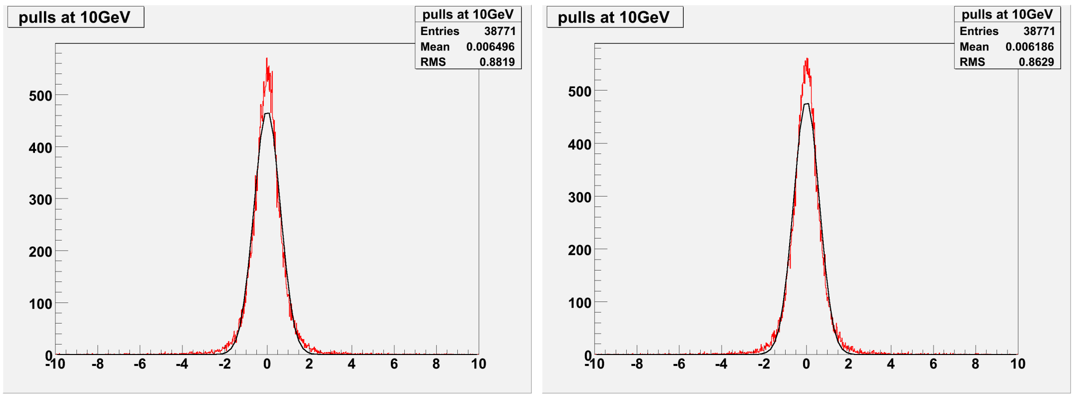

Figure 9: Fitted slice in bin at $10GeV$ for multiple scattering (left) and with combinded material effects (right)

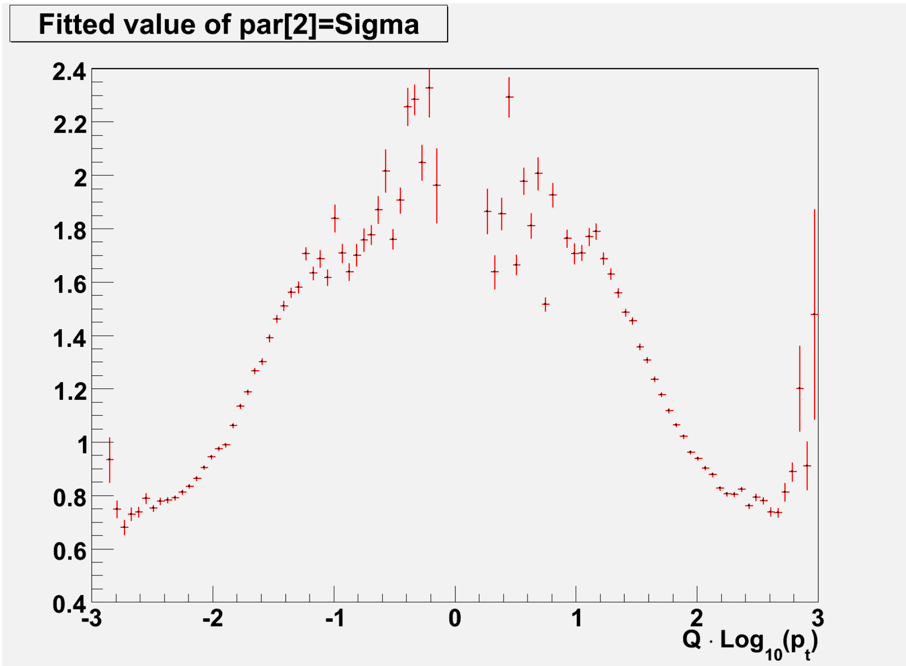

Figure 10: σ_n for `DualTrajectory` with no material effects

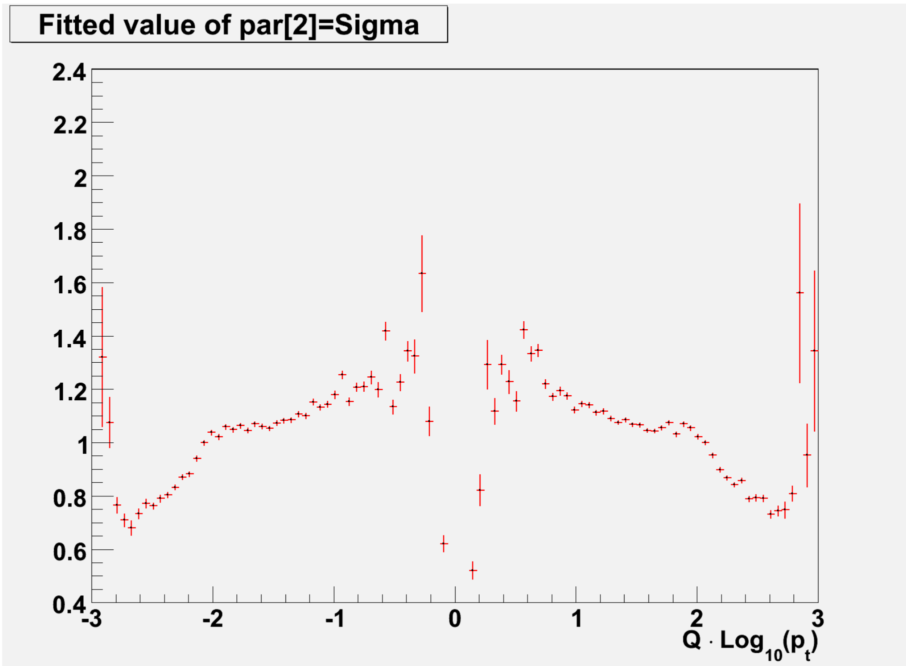

Figure 11: σ_n for `ReferenceTrajectory` with no material effects

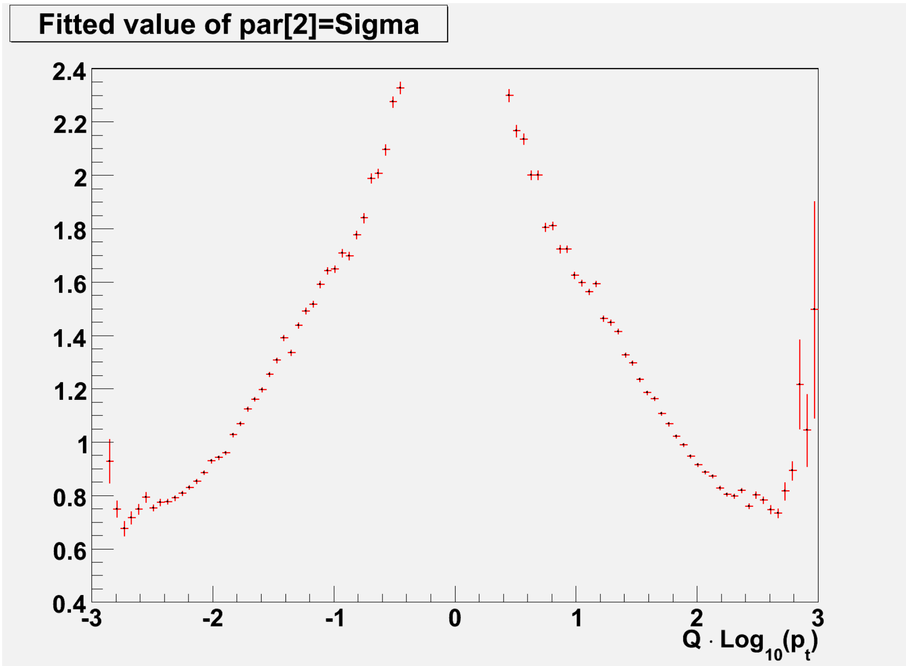

Figure 12: σ_n for `DualTrajectory` with energy loss

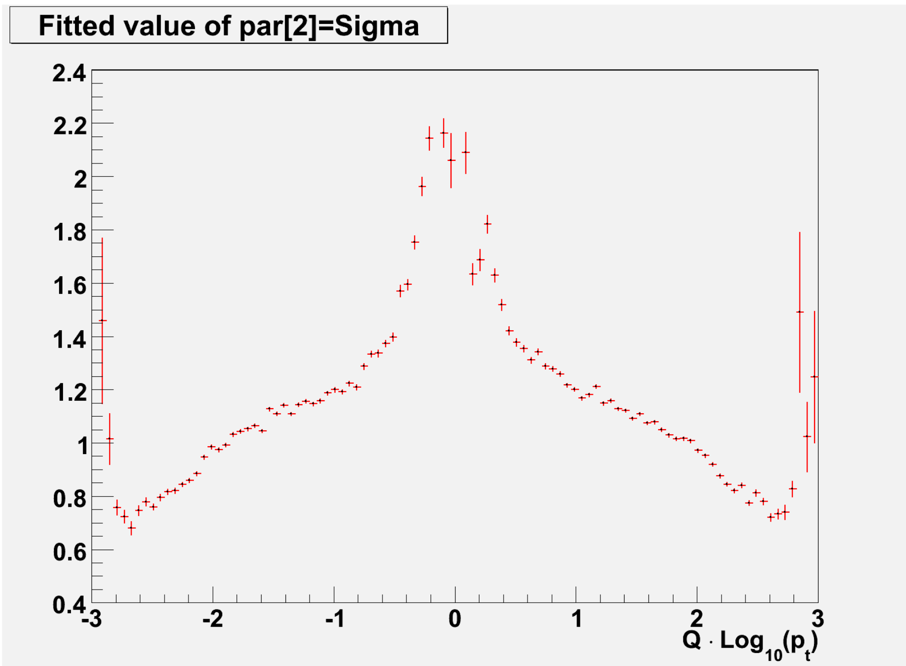

Figure 13: σ_n for `ReferenceTrajectory` with energy loss

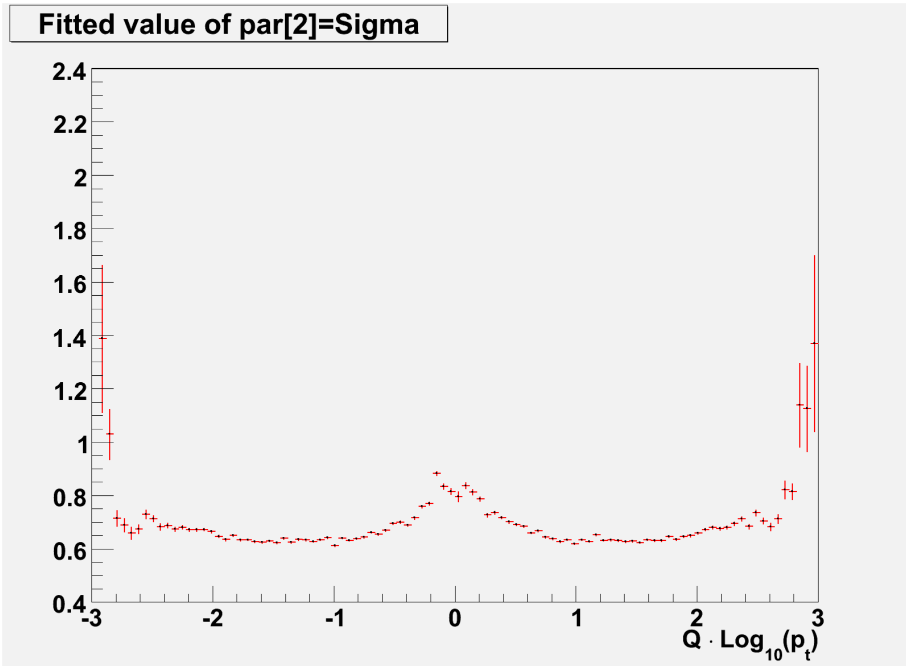

Figure 14: σ_n for `DualTrajectory` with multiple scattering

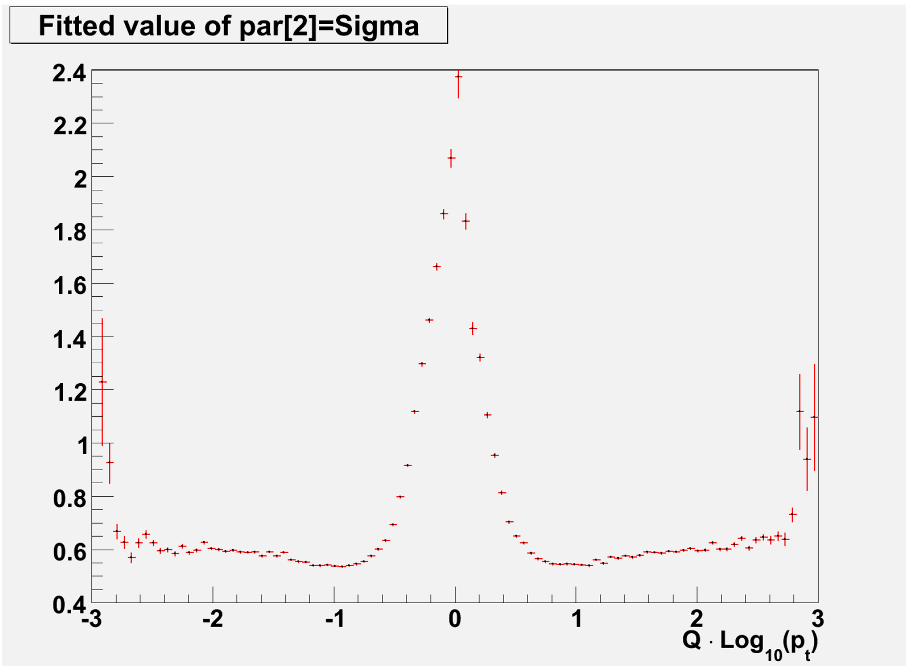

Figure 15: σ_n for `ReferenceTrajectory` with multiple scattering

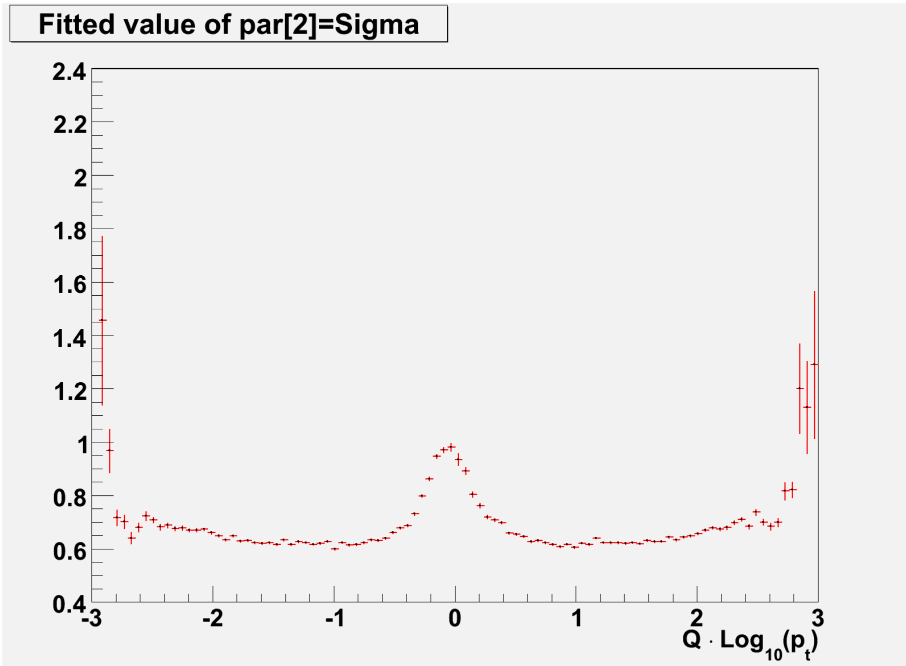

Figure 16: σ_n for `DualTrajectory` with combined material effects

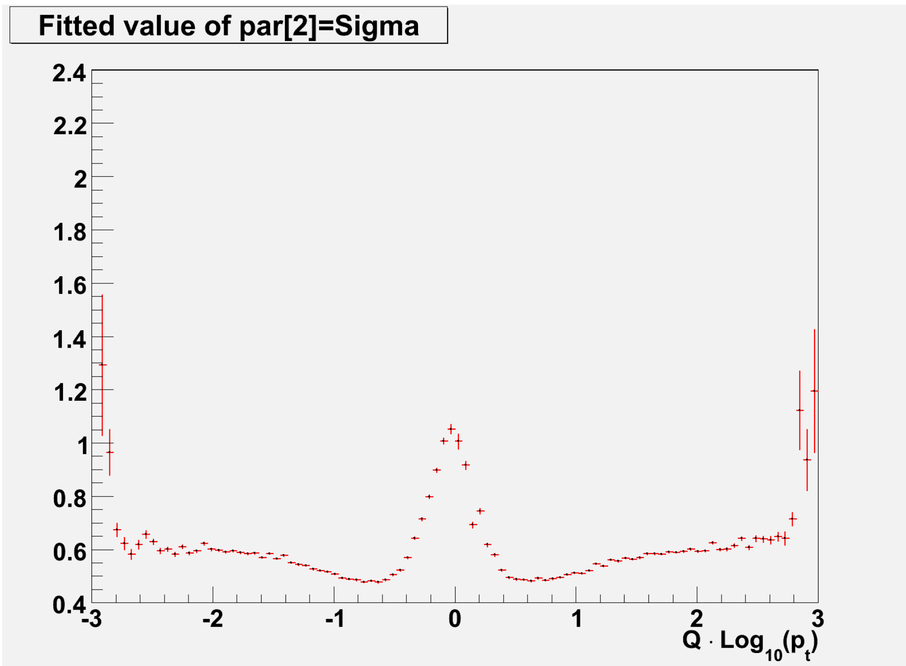

Figure 17: σ_n for `ReferenceTrajectory` with combined material effects

3.4 Track estimated residuals

Due to assembly and other reasons not all modules are oriented in the same direction. So when one module measures in u direction the next module might measure the same track in $-u$ direction. The same for v modules. One can unfold this effect by using the sign of the 4th and 5th local parameter which are the partial derivate with respect to u and v.

$$r'_i = r_i \cdot sign(local_4 \textbf{ or } local_5)$$

Figure 18 shows the weighted mean of all transformed residuals r'_i in a track versus the momentum. There is no systematic trend to either positive or negative residuals as shown in figure 19 that shows the mean of every single bin.

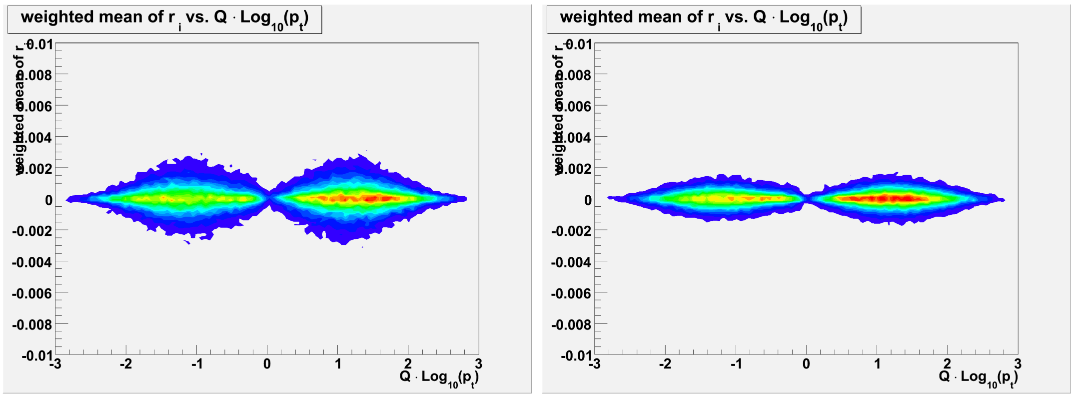

Figure 18: Weighted mean of all r'_i of a track versus $Q \cdot Log_{10}(p_t)$ for `DualTrajectory` (left) and `ReferenceTrajectory` (right)

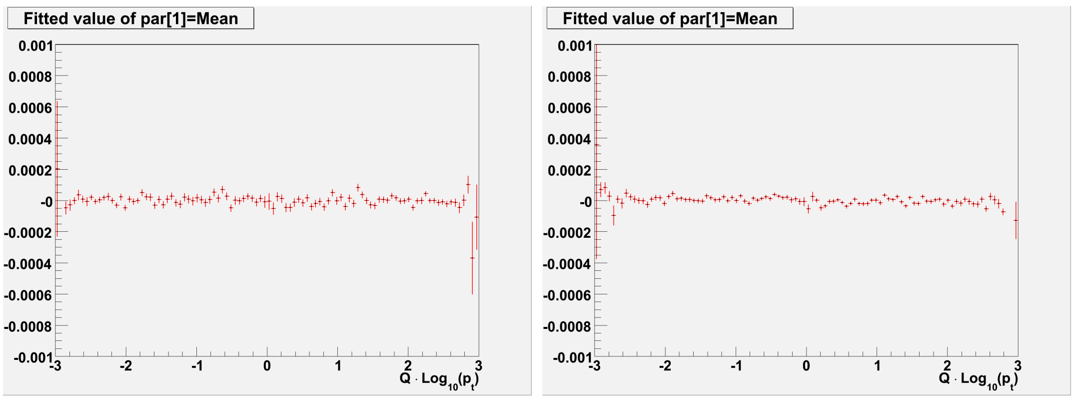

Figure 19: Mean per bin as shown in figure 18 for `DualTrajectory` (left) and `ReferenceTrajectory` (right)

4 Conclusion

Finally there are some results that are worth to mention again:

- The pulls are nearly constant over more then one order of magnitude in the momentum. So the proportionality to the inverse momentum might not be well described. One expects the error due to multiple scattering at $100 GeV$ to be only a tenth of the error at $10 GeV$. As the errors are add squared to the intrinsic error one would expect a bigger influence. Execpt for a dominating intrinsic error but this is not the case. Taken multiple scattering into account the shape of σ_n changes completly and so the intrinsic error can not dominate.
- Energy loss effects only the low momentum range below $10 GeV$
- For low momentum the combined material effects have different effects as compared to multiple scattering only. Adding energy loss either increases σ_n or decreases it. In both cases it improves the results but in two different ways.

In general the `DualTrajectory` seems to show the better results. The pulls width σ_n is closer to 1 and it one can explain the behavior in u and v direction.

A Changes to MillePedeAlignmentAlgorithm.cc

The below mentioned code was added to the MillePedeAlignmentAlgorithm.cc in the method `int MillePedeAlignmentAlgorithm::callMille2D([...])` after the first call of `theMille->mille([...]);` to write the special data of every record after the first hit of a record.

```
//write special data to the milleBinary.dat
if (iTrajHit == 0) {
    const AlgebraicVector &pars = refTrajPtr->parameters();
    int nPar = pars.num_row();
    std::vector<int> integers(nPar); // filled with 0.
    std::vector<float> floats(nPar);
    for (int i = 0; i < nPar; ++i) {
      floats[i] = pars[i];
    }
    theMille->special(nPar, &(floats[0]), &(integers[0]));
}
```

References

[1] Volker Blobel, “Millepede II - Linear Least Square Fits with a Large Number of Parameters”, http://www.desy.de/∼blobel/

[2] Markus Stoye, “Calibration and Alignment of the CMS Silicon Tracking Detector”, Dep. Physik - Universität Hamburg, Signature N4/3529

Acknowledgements

I would like to thank DESY for providing the chance to spend a great time here on site. Special thanks to my supervisor Claus Kleinwort, who had always time for my questions and taught me a lot. Thanks also to Gero Flucke who provided assistance with the CMSSW when I got stuck.