Alignment of the FoCal prototype detector with cosmic muons

Maxwell Hessey
University of Utrecht

Supervisors:
dr.ir. G.J.L. Nooren
dr. E. Rocco
M. Reicher

26-06-2013

ABSTRACT

The Forward Calorimeter (FoCal) is a proposed subdetector upgrade for the ALICE experiment at CERN. In this research we refer to the FoCal prototype, which is a digital sampling electromagnetic calorimeter. In order to properly evaluate the energy resolution of the FoCal prototype the chip positions of the silicon chip sensors have to be well defined with respect to a fixed reference system. An algorithm that takes into account the strong correlation between sensor position and reconstructed track position is put in place. After the first iteration the chip positions are known to within an accuracy of 0.1 mm.

CONTENTS

1. INTRODUCTION

The Forward Calorimeter (FoCal) is a proposed subdetector upgrade for the ALICE experiment at CERN. A FoCal prototype has been made which is a digital sampling electromagnetic calorimeter. It consists of 96 chips whose positions with relation to a global coordinate system are known to within an accuracy of about 1mm. Using data from cosmic radiation, small corrections to the positions of the chips can be achieved. This in turn increases the energy resolution of the detector. This research will outline the iterative procedure behind the calculation of these small corrections. This procedure is one that takes into account the correlation between the position of the chips (which will be determined by the event position), and the measured position of the events (which are determined by the chips).

2. THEORY

2.1 Cosmic radiation

The muons used to align the FoCal Prototype are formed in the atmosphere, by high-energy particles from space interacting with atmospheric molecules. This interaction may cause a shower, with many particles being created. Some of which will be the muons used. Muons are well suited to detector alignment due to the fact that they do not shower, and should therefore be perceived as straight lines.

2.2 Scintillator

A scintillator is a term for any material in which scintillation occurs. Scintillation is the excitation of an electron by a passing particle and the subsequent emission of a photon as the electron returns to its original state. Scintillator are often combined with photomultipier tubes to detect passing particles. In a photomultiplier tube, incoming photons hit a photocathode upon which serval electrons are emitted. These electrons are accelerated towards a metal electrode (called dynode) where in turn more electrons are created, which are again accelerated towards a dynode, until a detectable electric pulse has been created. This pulse is what signals that a particle has passed through the scintillator.

2.3 The FoCal Prototype

2.3.1 Working Principle of the FoCal Prototype

The FoCal prototype is a digital solid-state sampling electromagnetic calorimeter. A calorimeter measures the energy or of incoming particles, a key number in this is the so-called energy resolution of the detector. In the case of the FoCal prototype, measurements are done by considering electromagnetic particles. Sampling refers to a technique where absorbent layers, in this case made of tungsten, and a signal generating layer, made of silicon, are used. Solid-state to the use of a solid material that generates a signal, in the case of prototype silicon. Digital refers to the output. As a sampling calorimeter, position measurements are possible, which allow for particle identification.[1]

2.3.2 Geometry of the FoCal Prototype

At either end of the prototype is a scintillator, whose sides are 40 mm by 40 mm. The prototype consists of 24 silicon layers, interjected with tungsten layers (see figure 2.1).

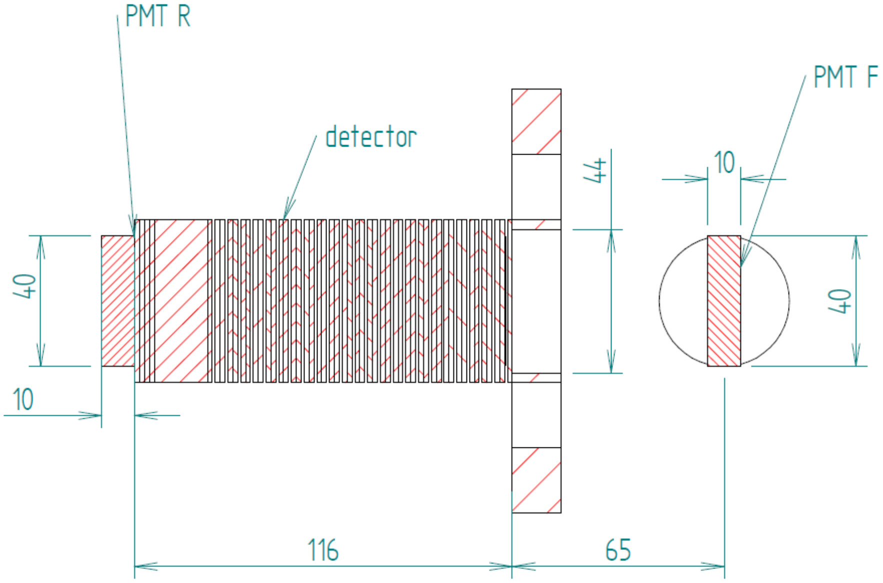

Fig. 2.1: A side on view of the detector. The z direction is from right to left (Front scintilator (F) to rear scintilator (R)). The dark line illustrates a silicon layer. The thicker, dashed, blocks are tungsten layers.

Each layer is made out of two pairs of silicon chips (so there are four chips per layer, see figure 2.2).

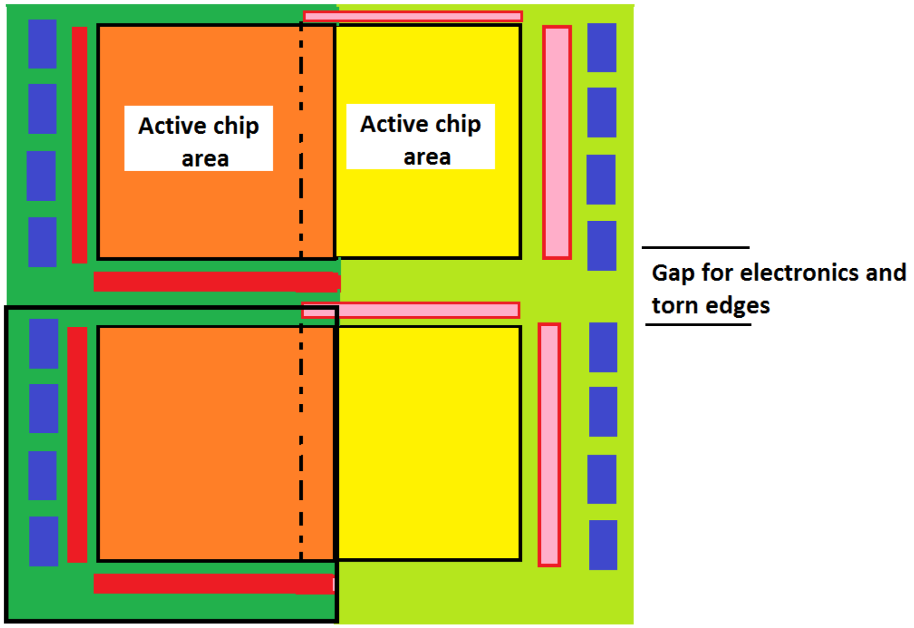

Fig. 2.2: Four chips, the pair on the left (dark colours) overlapping the pair on the right (light colours), denoted by the dotted line. The red and pink boxes are the line logics of the upper and lower chip pairs respectively. The blue boxes are the bit logics.

There is an overlap between the two pairs of chips (again see figure 2.2). Each chip has 640 by 640 pixels, each 30 by 30 μm, as depicted in figure 2.3.

These pixels form the an active area of the chip. The active area is 19.2×19.2 mm^2. The total chip dimensions are 19.65mm by 21.57mm (see figure 2.4).

2.3.3 Data acquisition process for cosmic muons

When a muon passes through both the front and the back scintillator the detector will be read out. All chips are read out at the same time. Each chip in the detector is read out sequentially by line. The readout of each line takes about one microsecond, since there are 640 lines, the readout of each chip and therefor the whole detector, takes 640μs.

Compared to this timescale the muon may be considered to be everywhere

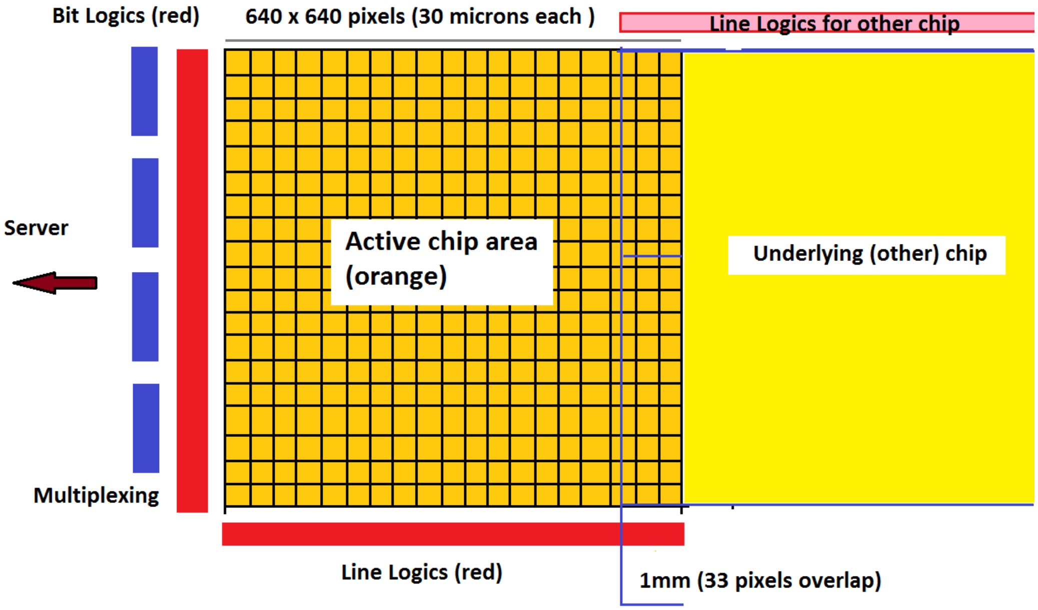

Fig. 2.3: Schematic representation of a chip

along its path at the same time. Due to the readout method, it may occur that a muon may enter at a line that has already been read out (where it is not seen) and exit at a point that it is yet to be read out. To deal with this, each time there is a coincidence in the scintillators, three full detector readouts are done. In this way the full muon track is always present and can be reconstructed by software.

Data is transferred in a compressed (jumbled) state. Before analysis, it is demultiplexed (un-jumbled) and the background corrected for. The background is determined before a series of measurements by always reading out the detector, independent of coincidence between the scintillators. Further analysis is covered in section 4. It is good to note that pixels a muon passes through often leak charge to their neighbours, resulting in several hits close together.

2.4 Motivation for the alignment of the FoCal prototype

In order to properly evaluate the energy resolution of the FoCal prototype, the chip positions must be well defined. A more precise alignment of the

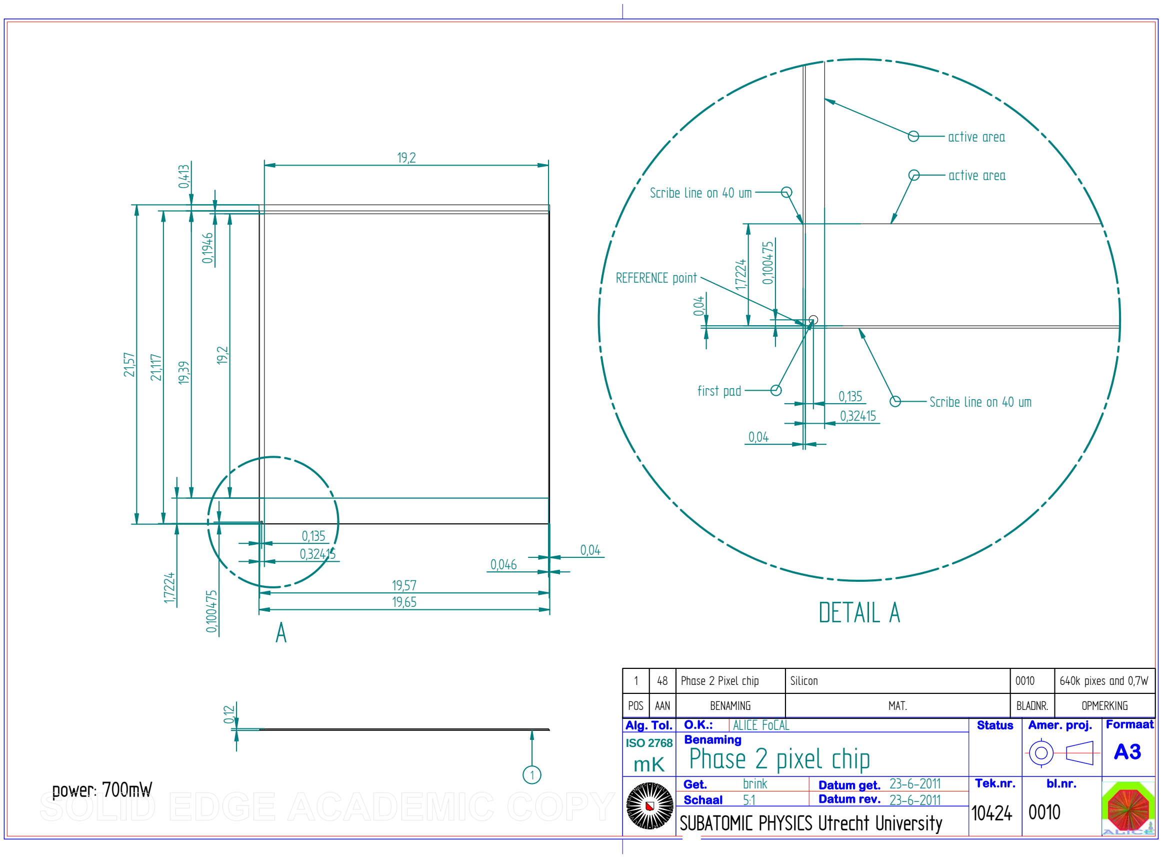

Fig. 2.4: Technical drawing of a single focalchip.

chips also increases the detectors spacial resolution.

3. MATHS AND THEORY BEHIND THE ALIGNMENT ALGORITHM

This section will introduce, step by step, the maths behind the alignment algorithm. The notation for the simple case is a bit more complex than necessary, this is done with consistency of notation in mind. The axes are those such as described in 3.1.

3.1 Defining axes and reference frames

In this text several reference frames will be used. The global reference frame is a right handed axis frame with the z axis pointing from the Front scintillator to the Rear scintillator (as in figure 2.1). (0,0,0) is in the middle of the front scintillator. Local frames on a chip are(x,y) frames with the zero in the center of the chip.

3.2 One-Dimensional (single variable) Case

3.2.1 Simple linear residual minimization

Possibly the most simple example of an optimisation problem, is that of fitting a line through a set of known data points. The principle is to minimize the sum of squared errors, also known as residuals, where the errors are the differences between the line and a measured point. We will first introduce some notation: $x_M(n)$ will denote the x coordinate of the n-th measured point, $x_E(z_n)$ is the corresponding predicted or expected value, the value on the line. The set C_j is the set of all points which have z value z_j. For a line through N measured points we minimise, with respect to whichever variables may influence $x_E(z_n)$, the sum:

$$\sum_{n=0}^{N} \frac{(x_E(z_n) - x_M(n))^2}{\sigma_n^2} \tag{3.1}$$

3.2.2 Introducing offsets

In the example of fitting a line through data points, we have that $x_E(z_n) = az_n + b$, so there are two parameters with respect to which we may optimise the sum. Now consider the more complex case that for each z_n there is some small systematic error, which we will call offset (the reason for which will be apparent later). Note that all points at this z value share that offset. We are no longer only interested in finding the two parameters that define the line, but in the offset for each point z_n. The measured values $x_M(n)$ are now systematically incorrect. We now redefine $x_E(z_n)$ as:

$$x_E(z_n) = az_n + b - x_{off}(z_n) \tag{3.2}$$

This definition has an important consequence. The sign means that, when considering real data, once the offset is known, it should be added to the data before doing any further analysis (see figure 4.1).
Note that there is now an inherent freedom in the parameters defining the residual. If b increases by some value, we can simply subtract the same value from each x_{off} to achieve the same residual. Fitting both the parameters a and b and the values $x_{off}(C_n)$ simultaneously is difficult, but possible [2].

3.2.3 Introducing Multiple Lines

A detector measures many particles. In cosmic muons move in a straight line through the detector without showering. Each particle will have its own small set of corresponding measured points. Consider now the case in which there are several sets of data each corresponding to one line. We will call a set of points corresponding to a line a track (denoted T_i). The set $C(z_n)$ however still includes ALL measured points with z value z_n (so not just points attributed to a certain track, but to all tracks). Equation 3.1 becomes:

$$\sum_{n=0}^{N} \frac{(a(T_i(n))z_n + b(T_i(n)) - x_{off}(C_n) - x_M(n))^2}{\sigma_n^2} \tag{3.3}$$

3.3 Two-Dimensional(three global variables) Case

This is not conceptually more difficult than the one-dimensional case. The same difficulty in fitting the offsets and the track parameters simultaneously is present. Including the offsets in $\vec{r_E}$, the expected position, or in $\vec{r_R}$ is equivalent. It is easier to think of including the offsets and rotation in $\vec{r_R}$. Let $\vec{r_M C}$ be where the track is measured on the chip, $\vec{r_{off}}$ the offset in x and y of the chip and R the rotation of the chips x-axis with relation to the global x-axis (in a counter clockwise direction, see figure 3.1). Using small angle approximation[1], R is given by:

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \approx \begin{pmatrix} 1 & -\theta \\ \theta & 1 \end{pmatrix}$$

We define:

$$\vec{r_R} = R(\vec{r_M C}) + \vec{r_{off}} + \vec{r_{pos}}(C) \tag{3.4}$$

Equation 3.1 becomes:

$$\sum_{n=0}^{N} \frac{(\vec{r_E}(z_n) - \vec{r_R}(n))^T(\vec{r_E}(z_n) - \vec{r_M}(n))}{\sigma_n^2} \tag{3.5}$$

3.4 Three-Dimensional (six global variables) Case

The Three-Dimensional case differs from the two dimensional only in that R is now a 3×3 matrix and r a three dimensional vector. Equation 3.1 is:

$$\sum_{n=0}^{N} \frac{(\vec{r_E}(z_n) - \vec{r_R}(n))^T(\vec{r_E}(z_n) - \vec{r_M}(n))}{\sigma_n^2} \tag{3.6}$$

Again we redefine $\vec{r_R}(n)$ as:

$$\vec{r_R} = R(\vec{r_M}) + \vec{r_{off}} \tag{3.7}$$

Where now:

$$R = R_y R_x R_z \tag{3.8}$$

So R rotates anti-clockwise around the z axis with an angle θ (like in the Flat Plane case),then anti-clockwise around the new local x axis with an angle ϕ and then anti-clockwise around the new local y axis with an angle ψ.

[1] See 3.5 for justification

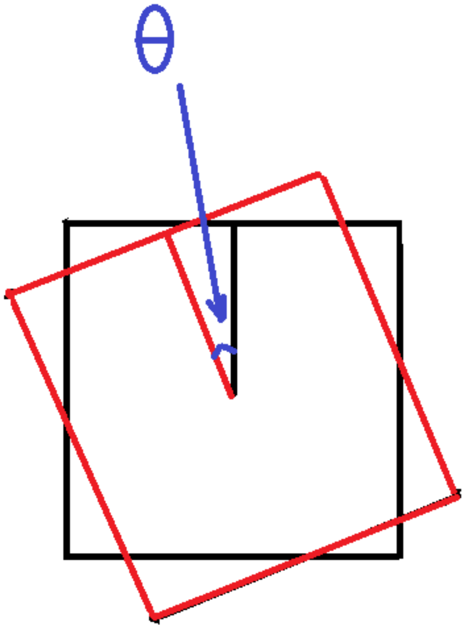

Fig. 3.1: Definition of R: R rotates measured points (black square) anti-clockwise to where they should be measured (red square). In this case over an angle θ.

3.5 Expected misalignments, assumptions on misalignments

Each chip is considered to have six degrees of freedom, and therefor misalignment: $\Delta_x, \Delta_y, \Delta_z, \theta, \phi, \psi$ where θ is the counter clockwise rotation looking along the z axis, ϕ the rotation around the local x axis and ψ the rotation around the local y axis. See figure 3.3. The positions of each chip are measured to within an accuracy of roughly 1mm. Due to the fact that all chips are close to each other, neither large misalignments in position or rotation are possible (figure 3.2). By observation the misalignments in x, y and z should be less than 1mm.

By calculation, θ should be less than $\arctan(\frac{1}{19.2}) = 0.05$radians. This number is derived by assuming we can measure within an accuracy of 1mm, the difference in x and y of both corners of a chip. Small angle approximation $(\cos\theta = 1, \sin\theta = \theta)$ would cause our answers for the misalignments in x and

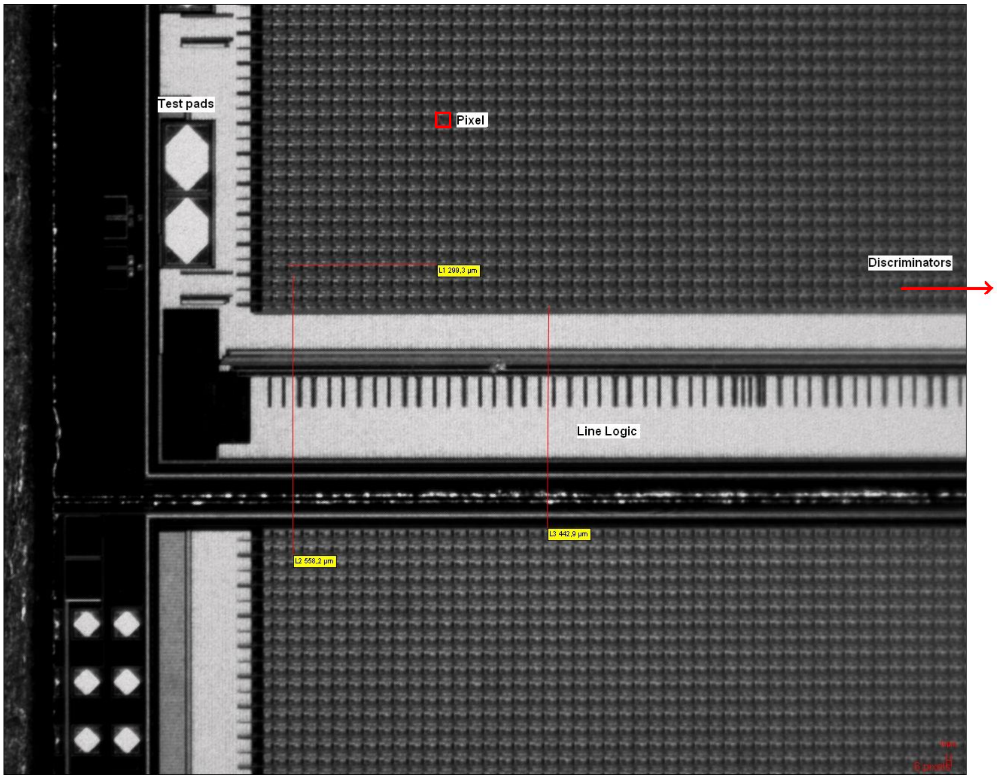

Fig. 3.2: Two chips, about 0.5mm appart. If one of the chips were to be rotated by any great angle they would overlap. Therefore the rotations must be small

y to be misaligned by:

$$(1 - \cos(0.05))x_M - (0.05 - \sin(0.05))y_M \tag{3.9}$$

and

$$(0.05 - \sin(0.05))x_M + (1 - \cos(0.05))y_M \tag{3.10}$$

in x and y respectively. In the worst case scenario that $x_M = y_M = 9.6$, from equations 3.9 and 3.10 this gives a maximum contribution to misalignment of: 0.012mm in x and y. More likely the contribution is about half that, 0.006mm. This means that the small angle approximation is acceptable to use for θ.

Using the same assumption on measuring accuracy as for θ, we obtain for ϕ and ψ a maximum of 0.1radians. The maximum error caused is $9.6(1 -$

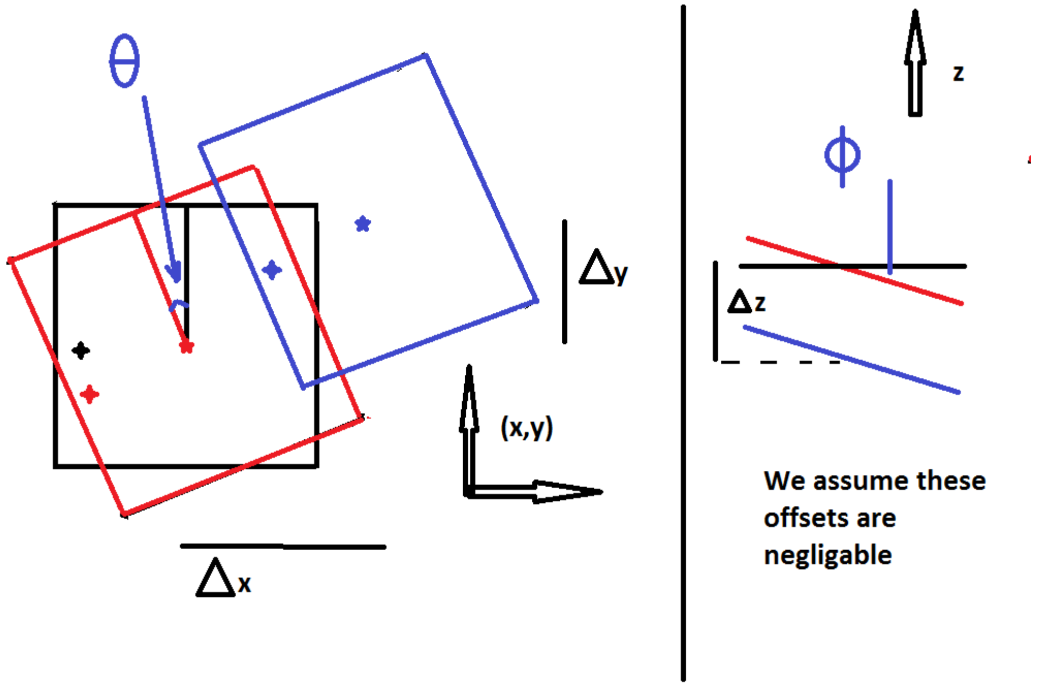

Fig. 3.3: In blue at the top right, the actual position of the chip. In black, edges parallel to those of the paper, where the chip is assumed to be. θ is the rotation from and Δ_x and Δ_y are the displacements from the assumed position, to the actual one. The star denotes the center of the chip. The blue plus is some point on the chip, for further feel.

$\cos(\phi)) = 0.05$mm (distributed over x and y such that the squared sum is 0.05). For the sake of simplicity, it has been assumed that this effect is negligible.
Due to the geometry of the detector, the maximum angle of incidence a muon can have is $\arctan(\frac{40}{196}) = 0.2$radians (see figure 2.1). Solving

$$\frac{\int_{-a}^{a} \cos^2(\alpha)}{\int_{-0.2}^{0.2} \cos^2(\alpha)} = 0.5 \tag{3.11}$$

gives us that half of the muons going through the detector have an incident angle of 0.1radians or less. Due to the distribution shape this is an upper bound for the average incidence angle. Using this boundary we find that a misalignment in z of 1mm will cause a contribution to offset c of 0.1mm. Let β be the rotation around the z-axis of a track, relative to the x-axis, then the contribution to offset in x and y are $\cos(\beta)$ and $\sin(\beta)$ respectively. However, since each track is determined also by β, the contribution to the offset will be in different directions each time and its average will therefore decrease as the number of tracks increases. It will however not decrease to zero, due to the fact that certain chips are more likely to have certain incidence angles. For example, a chip on the left in the top layer is unlikely to have tracks heading to the left go through it, as they would exit the detector (and thus not hit the Front scintilator, meaning they would not be measured at all). For the sake of simplicity, it has been assumed that this effect is negligible.

3.6 Dealing with local(track) variables and global(detector) variables

As previously mentioned, when trying to minimize equation 3.5 one has to deal with both the parameters of the tracks, and of the chips. Since these parameters are strongly correlated, one must be careful when considering them (see figure 3.4). Literature mentions two ways to do this, the Hit and Impact Points Algorithm and the Millipede style algorithm [3].

3.6.1 Biassed, simple algorithm

This is an easy to implement algorithm. First the track parameters are calculated by optimizing:

$$\sum_{n\in T_i} \frac{(a(T_i(n))z_n + b(T_i(n)) - x_{off}(C_j) - x_M(n))^2}{\sigma_n^2} \tag{3.12}$$

With relation to a and b, then all the offsets are found for that track. Since the track parameters are calculated before the offsets, the offsets have a strong bias.

3.6.2 Hit and Impact Points (HIP) Algorithm

To put it simply, this algorithm consists of finding the track parameters, and the parameters of one chip by using the hits in all other chips. This is done for each chip in the track. Then, after correcting for the offsets in each chip, the process is repeated. Although convergence is not guaranteed, reported iterations until convergence are reported in the range from 20 to 100 [4][2]. Mathematically put, consider again equation 3.3. In the following $A \setminus B$ is the set A excluding B. Each offset $x_{off}(C_j)$ is calculated in two steps. First the track parameters are calculated, by minimizing with respect to a and b:

$$\sum_{n \in T_i \setminus C_j} \frac{(a(T_i \setminus C_j)z_n + b(T_i \setminus C_j) - x_{off}(C_j) - x_M(n))^2}{\sigma_n^2} \tag{3.13}$$

Then the offsets are calculated assuming that the track parameters are known, so

$$\sum_{n \in C_j} \frac{(a(T_i \setminus C_j)z_n + b(T_i \setminus C_j) - x_{off}(C_j) - x_M(n))^2}{\sigma_n^2} \tag{3.14}$$

is minimized with respect to $x_{off}(C_j)$. This process is repeated for each set C_j. After one iteration, one corrects the data using the offsets, then repeats tho process by finding first a and b and later $\{x_{off}(C_j)\}$. Figure 3.4 shows a comparison of the methods discussed so far. It considers the case that most hits have negligible offset, except for the third. If linear regression is used and then the hits are moved, each will be moved onto the (exagerated) blue line(Average Position Alignment Algorithm). If we consider the hit and impact point algorithm. At the first iteration, the third point will be moved onto the red line, and the others onto the green line. At the second, the third point will be moved onto the green line and the others onto the dark red line. This will stabilize at some line below the blue line.

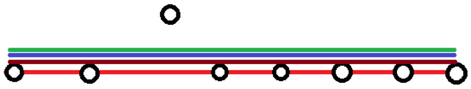

Fig. 3.4: This is an illustration of errors that can arise from the correlation between track parameters and offsets. The circles represent hits on chips. In blue the result of the Biassed simple algorithm. The red line is the line found by excluding the third chip. The green line an example of excluding say the fourth chip. The dark red line is the final result of the HIP algorithm.

4. ALIGNMENT SOFTWARE

This chapter is dedicated to explaining what software is currently in place.

4.1 Detector Simulation

In order to test wether the alignment algorithm is working, software to simulate the FoCal prototype was made. The software builds the FoCal prototype with certain simplifications. The detector is built out of 96 chips, where 'chip' is the active area. These chips are all assumed to be square with a width of 19.2mm. It is assumed (for compatibility with the track finder software) that four chips in a layer all share a z coordinate (see the section on Detector Geometry for the correct layout). Each chip is assumed to be in a quadrant, centered around $(\pm 9.6, \pm 9.6)$ and $(9.6\pm, 9.6\mp)$. This is done in order to be compatible with the tracker software. The AlignmentTester class allows the user to randomly missals each chip in the X and Y direction as well as giving it some small rotation theta. Here the Z axis is from the front scintillator to the back and theta is the angle between the chip X axis to the global X axis.

4.2 Data Simulation

Once the detector has been made, FOCALHit objects (representing active pixels) can be simulated. The FOCALHit objects are stored in FOCALFrame objects, which are stored one to one FOCALTracker objects. At the time of writing, this is the same as in the analysis software. Random entrance and exit points are generated, which we shall call testseeds. The line connecting two such testseeds is a seedtesttrack. Along each seedtesttrack the intersections with the chips are calculated. These take into account the rotations and offsets. The simulation algorithm then converts the (x, y) coordinates on the hit chips to a line and bit number (pixel identifiers). If the line and bit are not on the edge of a chip, five FOCALHits are generated: at the pixel hit, and the four adjacent pixels. We generate several hits because the tracker software requires it. At the time of writing the tracking software requires a track to have at least 24 hits dispersed over twelve layers. These values are somewhat arbitrary. The FOCALHits are made where they would have been measured by the detector (see figure 3.3). Once the FOCALHit objects have been made, we have 'data' to run the alignment algorithm on. From this point onward the alignment process is identical to that which we will apply on the data.

4.3 Track Finding Algorithm

The FOCALTracker class is responsible for finding FOCALTrack objects in a FOCALFrame. A FOCALTrack is a collection of FOCALHit objects that are attributed to the path of a particle. Often several FOCALHit objects will be bunched together. This is cause by charge leaking from the activated pixel into neighbouring pixels. Such a bunch of Hits is called a cluster. There are usually in the range of 2-5 hits in a cluster. If we forget about optimization steps, the FOCALTracker finds tracks as follows:

- Between every pair of FOCALHit objects (from now on I shall refer to these as Hits) a line is drawn.
- The intercepts of this line with the Front and Back scintillators are stored.
- When several Hits are on a line, their intercepts will all be similar. Points $(x_{in}, y_{in}, x_{out}, y_{out})$ near each other (using Euclidian metric) are taken to be the seeds of a 'seedtrack'
- Around the intercepts of the seedtrack with each layer, all Hits within a 1mm are attributed to the track.
- If there are enough Hits, they are grouped together as a FOCALTrack.

4.4 Alignment Algorithm

The short term aim here is to implement the Hit and Inpact Points (HIP) algorithm. However at the time of writing the algorithm is still in a testing phase. Currently the following steps occur:

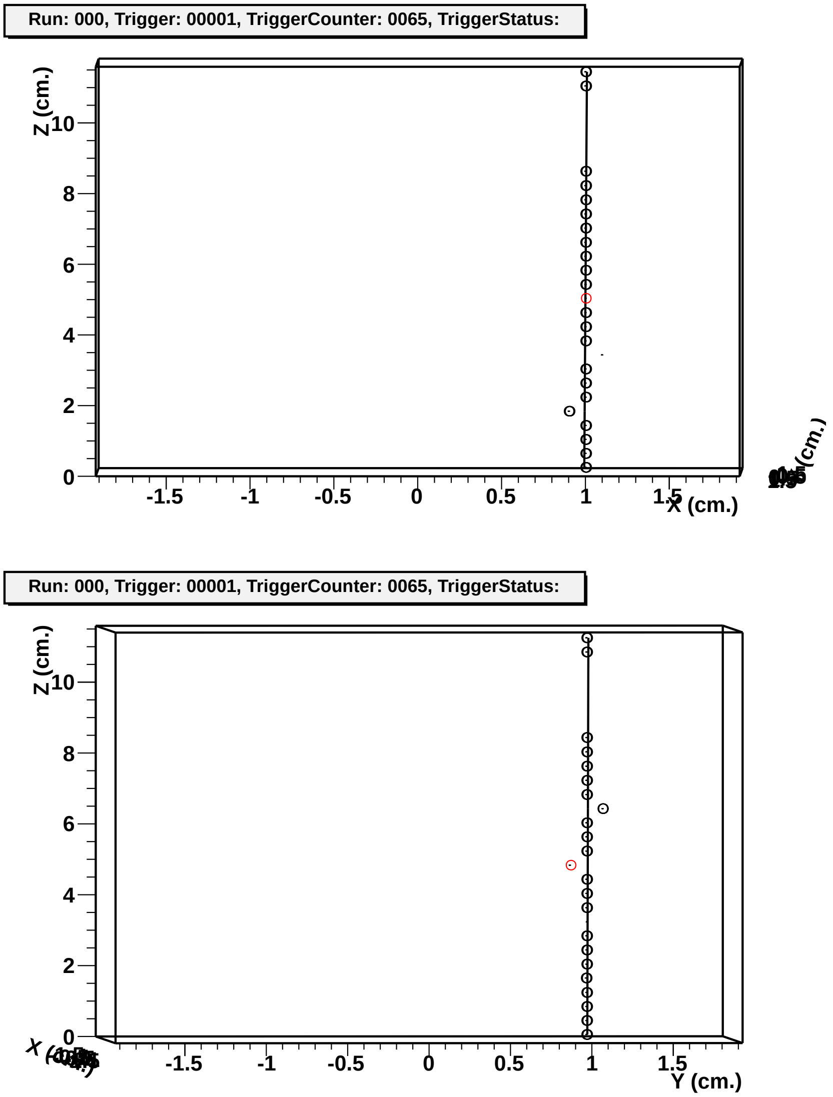

Fig. 4.1: The result of a simulated misalignment. Increassing in z the offsets are 1mm in x, -1mm in x, 1mm in y, -1mm in y. Note that this results in an opposite displacement in the measured point.

- For each track the track parameters are taken from the track finding 'seedtrack'.
- The measured points are found by taking the average position of a

cluster of hits.

- The offsets are then found by optimizing equation 3.5 (one chip at a time).

This method has many serious drawbacks, the most noteworthy of which are the bias from first calculating the track parameters and the fact that the 'seedtrack' is not necessarilly the best value for the seeds. The maths implemented is that of the 2-Dimensional (3 variables case).

4.5 Recapping the Assumptions

At the time of writing, the following implicit assumptions are in place in the algorithm:

- Small angle approximation
- There is no rotation in the ϕ angle, and no offset in the Z direction.
- All chips in a layer share the same z coordinate.
- The alignment algorithm is written to be compatible with tracker software from the third of may, 2013.

4.6 Upgrading the Algorithm

First and foremost the HIP algorithm must be implemented. As soon as this is done the Algorithm should be ready to use on real data. For improved results the algorithm could be expanded to allow three dimensional (6 variables) analysis.

5. RESULTS AND DISCUSSION

5.1 Data / Detector Simulation

The Simulation works well. If a chip is given an offset in X of 15μm to the right, all points measured will be measured 15μm to the left. This can be shown using the residuals between the intercept of the seedtrack with the chip, and the pixel considered to have been activated. Figures 5.1 and 5.2 show that the pixel activated is always within one pixel's size of the track, i.e. that the right pixel is activated. The spread is due to the fact that when a pixel is hit, the track is considered measured in the middle of the pixel. This allows for the pixel-sized (30μm) spread in the hits. The fact that there are never hits outside this box means that all intercepts of the generated track with the chip are reconstructed using the right pixel. In figure 5.3 the chip has been given an offset in X of 15μm, therefore points are measured -15μm compared to where they would be, so that the residuals are all around 15μm. The same effect can be observed in a different way in figure 4.1. Because at a first approximation tracks come in randomly, the flat distribution is to be expected. Therefore we can conclude that the simulation is working.

5.2 Residuals from Trackfinding

Figures 5.4 and 5.4 show ,in x and y respectively, the residuals, defined by the difference between the average position of pixels activated by a muon, and the expected position of that muon calculated using its track parameters. In figure 4.1 the residuals are the difference between the black circled points, and the black line. Due to the accuracy of the track finding software, these residuals have a larger spread than those from the pure simulation.

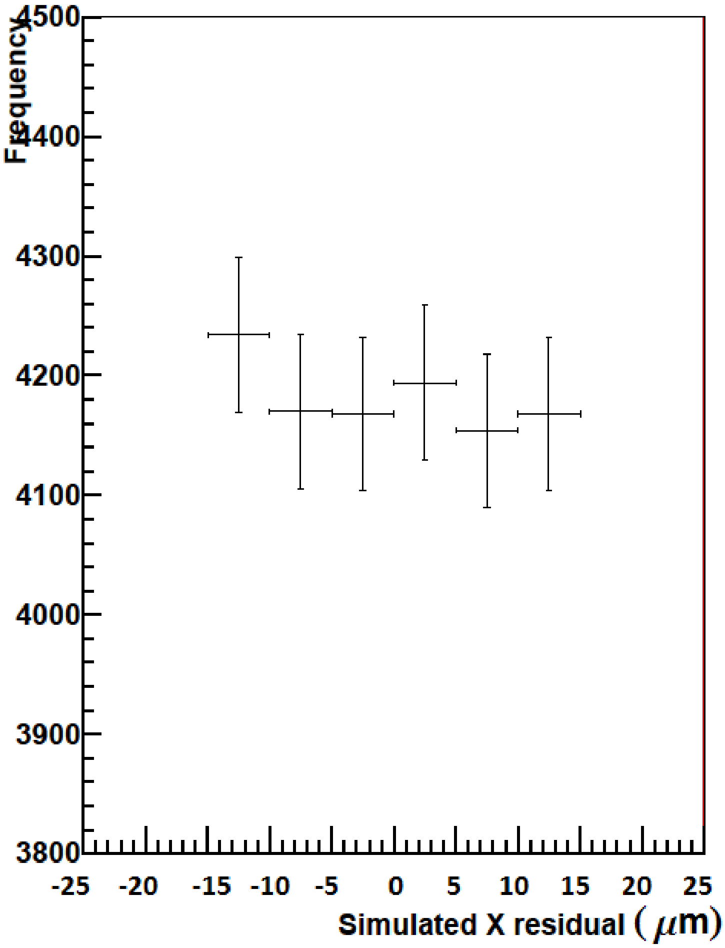

Fig. 5.1: Tracks are correctly generated, the x axis residual between the activated point and the simulated track is always between one pixel width (-15μm and 15μm). This is the desired precision in the simulation.

5.3 Performance of the Alignment Algorithm after one iteration

The Alignment Algorithm after one iteration has an accuracy of around 100μm (see figure 5.6). In order to generate this figure the chips are gener-

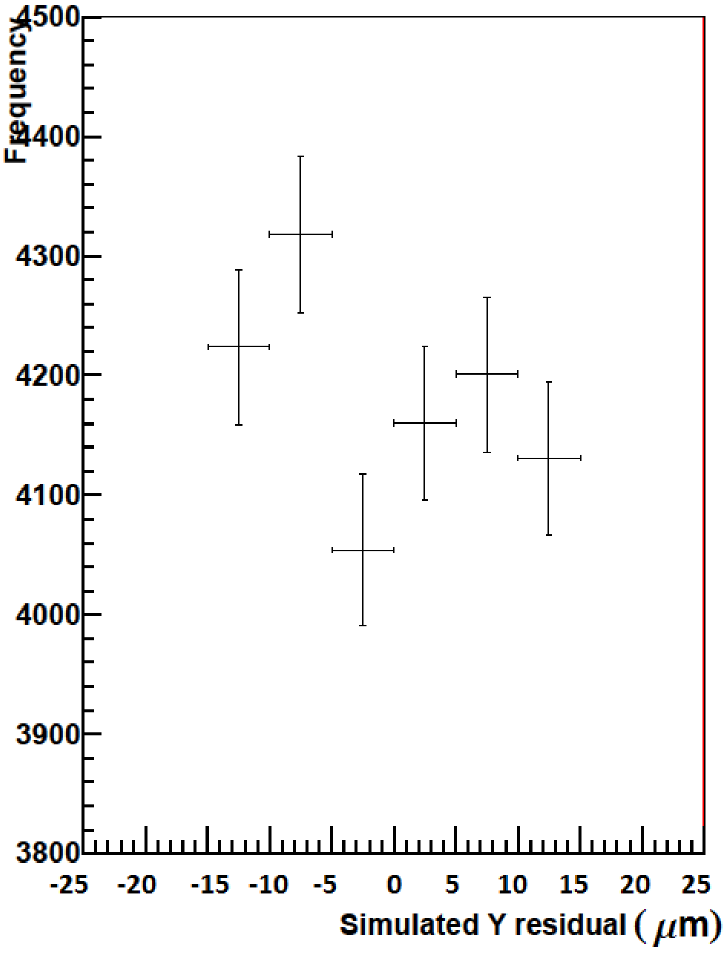

Fig. 5.2: Tracks are correctly generated, the y axis residual between the activated point and the simulated track is always between one pixel width (-15μm and 15μm). This is the desired precision in the simulation.

ated randomly misaligned in the x direction and perfectly aligned in the y direction. 10000 tracks are used to calculate the misalignment. The difference is on the x axis of the graph and the number of chips with that error in

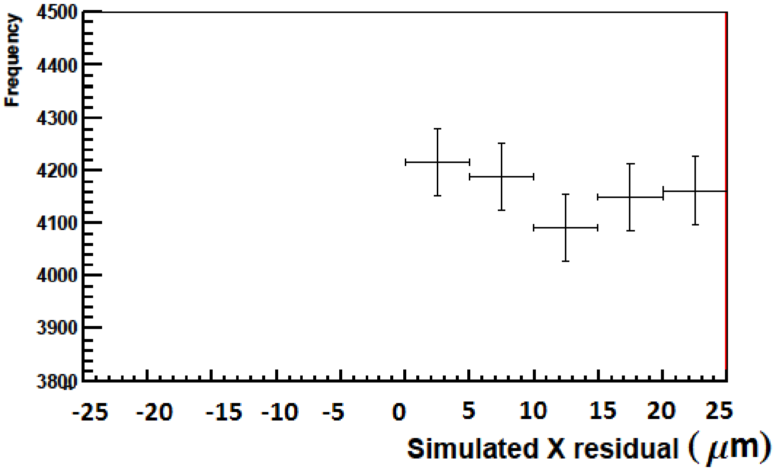

Fig. 5.3: Same image as figure 5.1, now with an offset of 15μm. Tracks will now be generated in one of two pixels (-15μm to 15μm) or (15μm to 45μm), still with one pixel width centered around 15μm.

position reconstruction on the y axis.

5.4 Error in reconstructed position dependant on z coordinate

There is a correlation between the error in the reconstructed position of a chip and its z coordinate (see figure 5.7). This may be due to an underdefined reference frame. Instead of all chips being moved to their correct positions, chips are moved onto a line along the average positions of the incoming tracks. Since chips in the middle of the detector have a wider range of track angles passing through them the effect is not so strong. At the edges, the average track is not going vertically, but has some angle.

However, the average angles of track incidence are independent of chip misalignment, therefore if the reference frame is underdefined one would expect similar behaviour from a perfectly aligned detector. Figure 5.8 does not show this behaviour.

The dependance on generated offset of the chips indicates that it is the correlation between chip parameters and track parameters that causes this effect. If this is the case then the effect should decrease after iterations of the algorithm.

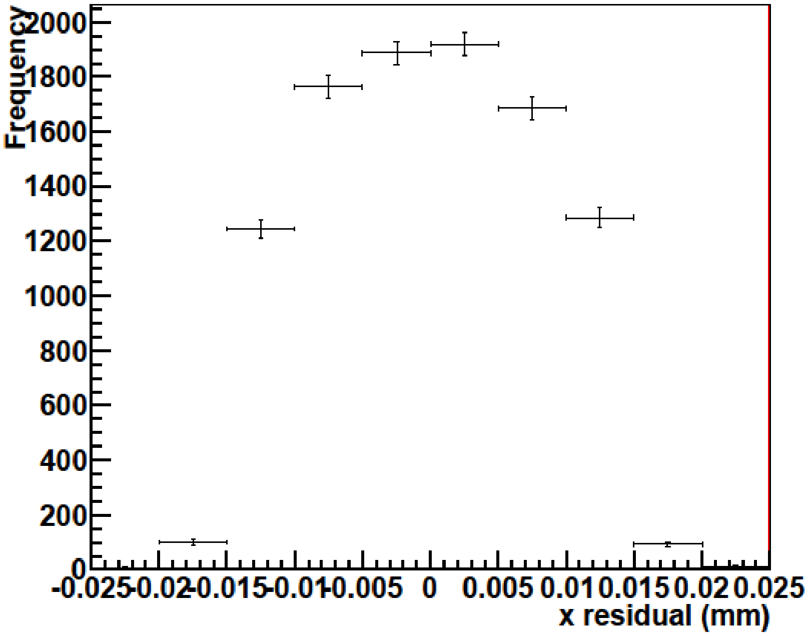

Fig. 5.4: The residuals in X after trackfinding are larger than those from simulation. Note the four low entry bins. Simulated residuals (difference between intercept from seedtrack and measured point) from 10000 tracks on a chip without any offsets.

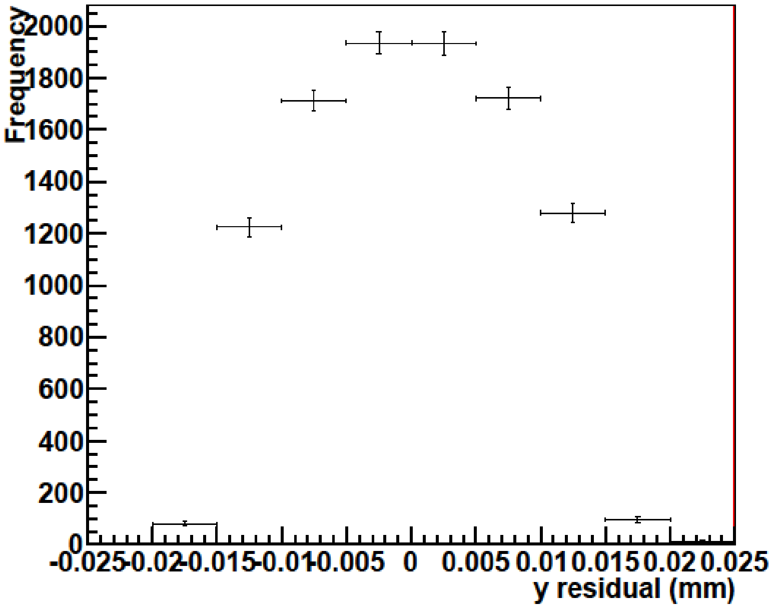

Fig. 5.5: The residuals in Y after trackfinding are larger than those from simulation. Simulated residuals (difference between intercept from seedtrack and measured point) from 10000 tracks on a chip without any offsets.

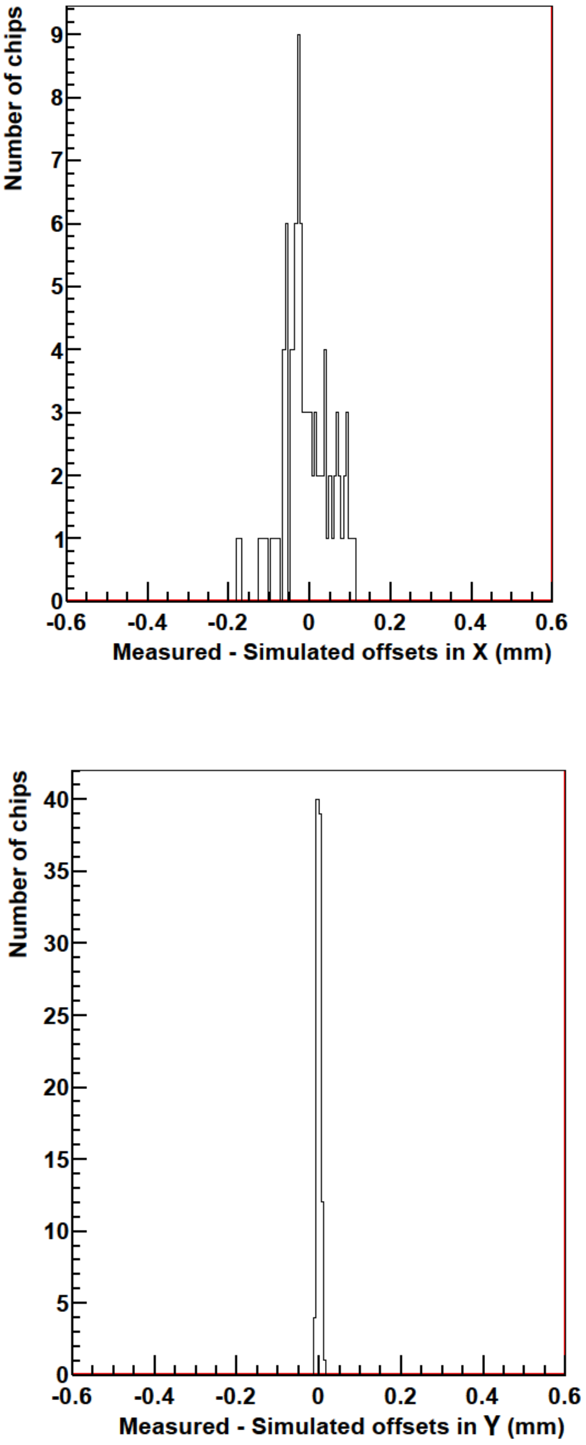

Fig. 5.6: The error in position reconstruction is generally less than 100μm. To aquire these results chips are randomly misaligned in the x direction and perfectly aligned in the y direction. A similar result has been recreated where the misalignment is set in y and the perfect alignment in x. These results are based on 10000 tracks.

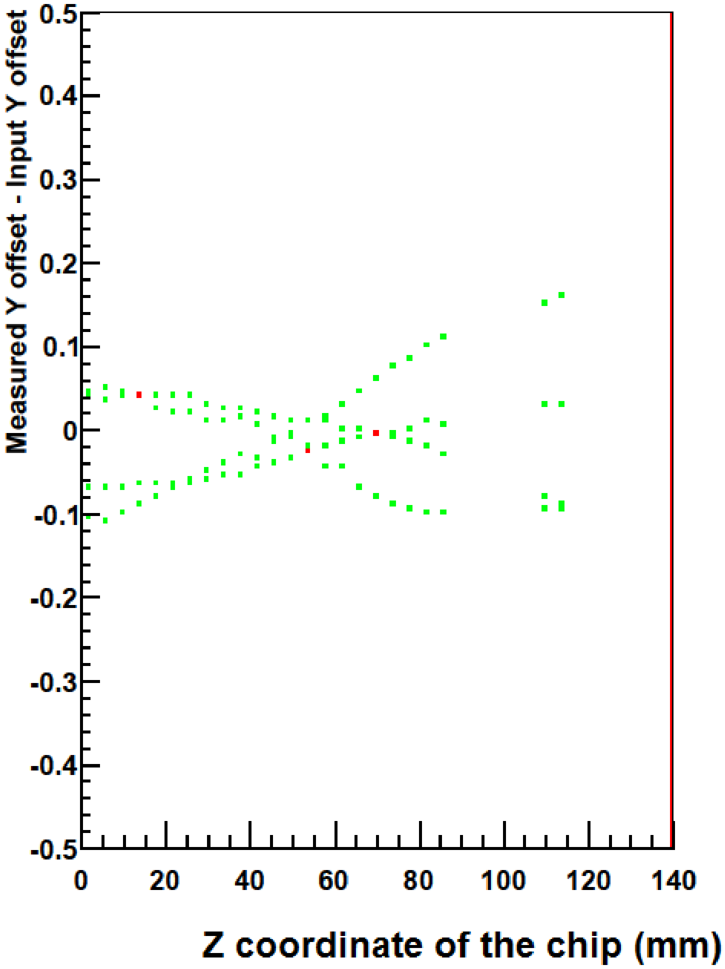

Fig. 5.7: Four lines are visible, one corresponding to each quadrant. This could indicate telescopic misalignment in each of the four quadrants. This figure is based on data from 10000 tracks where all chips have a misalignment in y but not in x.

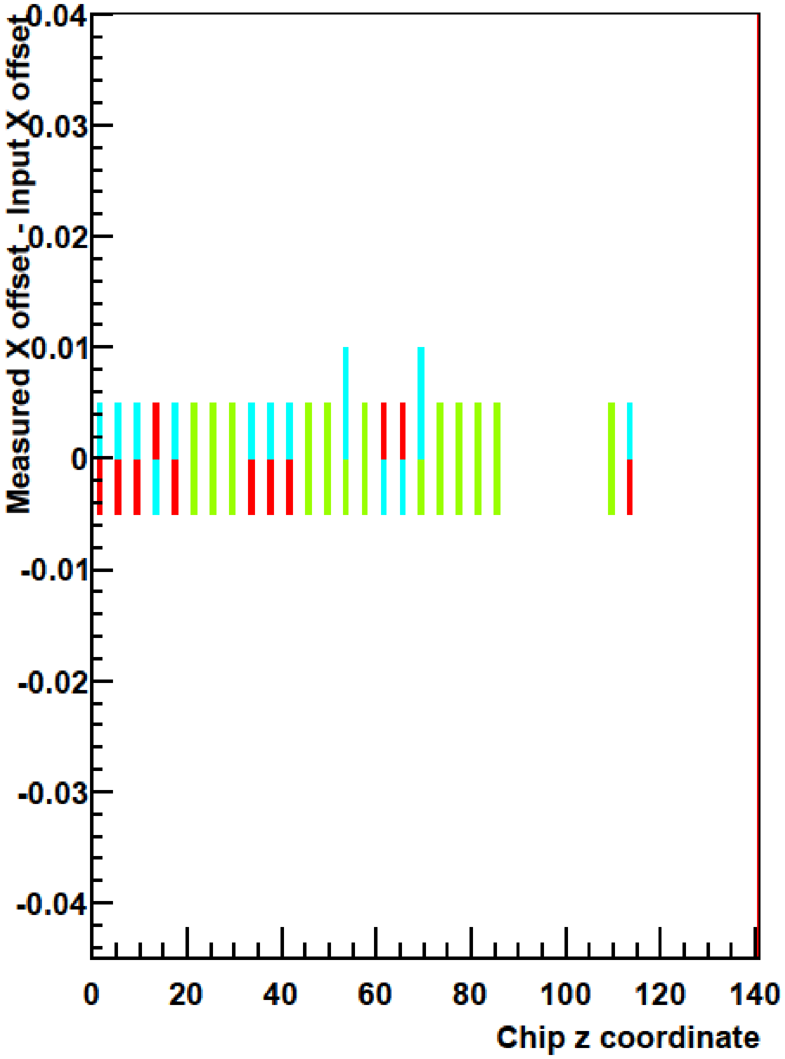

Fig. 5.8: No trend is visible in the Xerrors (the X position is always reconstructed well). This figure is based on data from 10000 tracks where all chips have a misalignment in y but not in x.

6. CONCLUSION AND OUTLOOK

6.1 Conclusions

The simulation software creates an accurate representation of the detector. The Alignment algorithm, in it's first iteration reaches an accuracy of 100μm, which is not the accuracy desired.

6.2 Outlook

Further refinement of the algorithm program is required. By fully implementing the HIP algorithm the z-coordinate dependency (figure 5.7) should diminish, in turn leading to an increase of accuracy. Once this has been achieved the algorithm can be used, without modification, on actual data from the FoCal, in order to accurately determine the chip positions and improve the energy resolution.

BIBLIOGRAPHY

[1] *Calorimetry for Particle Physics.* CERN-EP/2003-075.

[2] V. Blobel. *Alignment algorithms.* http://www.desy.de/ blobel/alirepcern.pdf5.

[3] The CMS Collaboration. *Performance of the CMS Silicon Tracker.* http://pos.sissa.it/archive/conferences/137/013/Vertex

[4] B Mours et al. *The design, construction and performance of the ALEPH silicon vetex detector.* Nucl. Instr. and Meth. A453 (1996) 101-115.