Quality Control of the Reconstruction Algorithms in a Tracking Detector

Alonso Guerrero Llorente

Universidad Complutense de Madrid, Spain

Supervised by Yorgos Voutsinas

September 5, 2014

Abstract

The ILC, *International Large Collider*, is going to be the next large particle collider. In order to exploit the potential of this collider a good method to reconstruct the tracks is a requeriment. A good reconstruction of the track of each particle is essential to reach our purposes, so a study of the accuracy of our reconstructed tracks is required. Towards that goal the resolutions of momentum and impact parameter, using reconstructed tracks generated by Marlin Software, are studied in this document. The efficiency and the track parameters pulls have been studied too. These results are obtained for several values of momentum and polar angle.

Contents

1 Introduction

The International Linear Collider is a proposed linear particle collider. It is planned to have an energy of $500GeV$ initially, with an option to upgrade to $1TeV$. Inside this accelerator electrons will collide with positrons.
The length of this new accelerator would be $31km$. It would be ten times bigger than the Standford Linear accelerator, the longest existing linear accelerator.
The International Large Detector (ILD) is one of the two proposed detectors for the ILC. Excellent calorimeter and tracking systems will be combined in order to get the best possible reconstruction. The ILD tracking scheme consists of three independent tracking system. The main tracking detector inside ILD is the TPC (*Time Projection Chamber*). It is complemented by the Vertex Detector, which is specially important for low momentum particles which do not reach the main tracker. Also, the FTD (*Forward Tracking Detector*) plays an important role in tracking, it provides measurement points for very small angles. The result of this combination of tracking devices is a high efficiency for the track reconstruction.
To obtain the reconstructed tracks it is necessary to use reconstruction methods with the signal which is provided by the tracking system.

1.1 What is tracking?

Tracking is concerned with the reconstruction of charged particles trajectories. The reconstruction of particles trajectories using the signals from tracking detectors is an essential part in experimental particle physics. If the trajectory of a particle is known we will be able to understand more things about its behavior. Some magnitudes as momentum or the sign of the charge can be calculated using the characteristics of tracks. For example, in the case we have a charged particle inside an homogeneous magnetic field its trajectory will be an helix and we would be able to calculate the momentum of that particle knowing the radius of that helix. In Figure 1 we can see some helix trajectories.
Reconstruction of charged particles from signals of tracking detectors is not an easy task. There are diverse methods of pattern recognition, which can not easily be ranked against each other. So, nowadays, there are not any accepted standard software for track finding. In new experiments, which contain a high track density, their success depends on the power of the reconstruction methods.

1.2 Aim of this work

The main aim of this work is to develop a tool which allow us to test do a quality control of an tracking software. With this tool we can study some aspect as resolution or pulls of several variables and efficiency. Furthermore, this tool can be used for different studies as well, like studying the performance of the detector versus the value of the magnetic field.

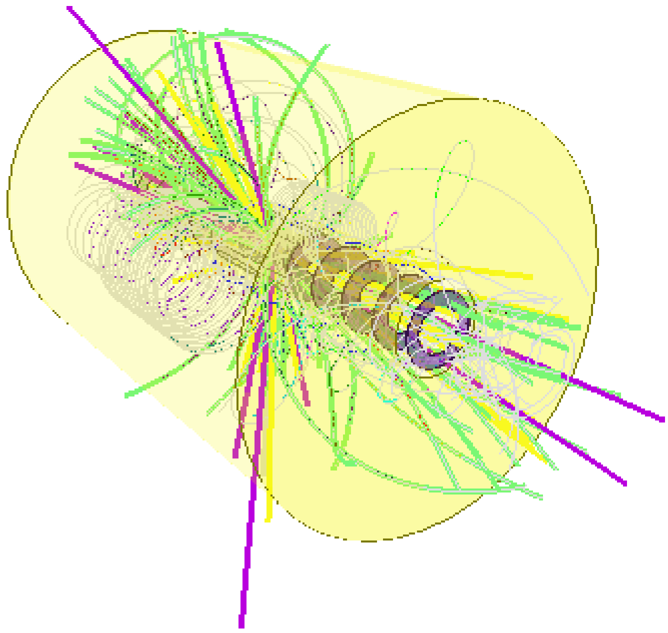

Figure 1: Visualization of MCParticles and MarlinTrkTracks for different kinds of particles in TPC, Si-system and Vertex Detector.

In order to do a quality control our tool needs to compare the reconstructed data with true data. We get the true data using Monte Carlo information.

2 Software and methodology

2.1 Software

In our study we do not have any signal from a detector, because it has not been built yet. So we need to simulate it. The software we use for the simulation is Mokka, which is a GEANT4 based software. Mokka takes as an input a generator file. Additionally, we can create our own single particles, by defining parameters as number of events, initial position, direction, momentum, polar angle, what particles we want to simulate, magnetic field, etc. When Mokka has already been run, we can check the number of hits and in what sub-detector they have been created. Finally, we have an output file which stores several collection of hits, one per sub-detector.
In order to use the simulation data, to reconstruct the tracks, the software we use is Marlin. All the events generated by Mokka are processed in Marlin. Marlin has a set of modules which handle each task in reconstruction stage, for example track finding. When the reconstruction process has finished, we are able to visualize the output file, figure 1, or to do a statistical or physical analysis.

2.2 Methodology

In order to use the data of the reconstruction we need to do three steps. These steps are: simulation[1], reconstruction and analysis.
In order to do it we wrote a script which is able to do all the simulations and reconstructions we want to analyze. So that script needs a *for loop* to change the input parameter for each simulation. A *simulation.slcio* file and one *reconstruction.slcio* will be created for each choice of input parameters. So, using one only script we are able to simulate our events with Mokka and reconstruct them with Marlin.
The next step is the analysis, which can be included inside the script too[2]. For this step we need to write a C++ processor. The goal of this processor is to collect all the reconstruction data we are interested in and store them in a *.root* file. In this file we should save not only track data but Monte Carlo information as well.
After these three steps we are able to use ROOT software to create plots and complete the data analysis.

2.3 Parameters which appear in this report

The quality of a tracking system can be characterized via the study of several parameters of the track and compare them with the particle parameters (Monte Carlo Particle information). The parameters we will use in this report are:

- **Omega** (Ω): is the curvature of the track. It is defined as the inverse of the radius:

$$\Omega = \frac{1}{R} \tag{1}$$

- **Transverse Momentum** (p_t): if we choose the z-axis like the direction of the beam the transverse momentum is $p_t = \sqrt{{p_x}^2 + {p_y}^2}$. Also it can be defined as a function of Ω:

$$p_t(GeV) = 0.3 \cdot 10^{-3} \frac{B(T)}{\Omega(mm^{-1})} \tag{2}$$

- **Impact Parameter** ($d0$): is the distance of the closest approach of the track to the collision point.

- **Polar Angle** (θ): is the angle formed by the direction of the particle we are studying and the z-axis. The values of polar angle are in the range $(0^o, 90^o]$.

[1]In the simulation we will create muons for several values of p_t and θ.
[2]A good idea is to write a script which is able to do the simulation, reconstruction and analysis. However, the analysis should be out of the script, because in the case we want to change something in the analysis processor we would not need to run all the script.

- **Track Finding Efficiency**: is the fraction of tracks which have been found versus the total number of tracks which are related to charged particles. This definition requires a criterion which specifies whether a particle has been found or not. If the particle has been found it means that we know which is the track of that particle. In this way we have a number of found tracks and a number of total tracks:

$$\epsilon = \frac{\#foundTracks}{\#totalTracks} \tag{3}$$

- **Weight:** this parameter describes the fraction hits of the track which belongs to the same MCParticle. If $weight = 1$ all the hits of the track belong to the same particle, in the case that $weight = 0.6$ the 60% of the hits belong to the same MCParticle and the rest belongs to different particles.

3 Results

Before studying how good the tracking software is, we need to simulate the particles whose tracks will be reconstructed. The values we use for p_t are in the range $[1, 500]GeV$ and for θ are 20^o, 40^o and 85^o. And the particles I use are muons.
I use muons because they do not have so many energy losses as other elementary particles. I can study the energy losses by seeing the bremsstrahlung radiation. In an helix the acceleration and velocity of the particle will form a right-angle and the energy of the particule will be $E = \gamma mc^2 \approx \sqrt{p^2c^2}$. So I can write the following expression for the radiation:

$$P_{a\perp v} = \frac{q^2a^2\gamma^4}{6\pi\epsilon_0 c^3} = \frac{q^2a^2}{6\pi\epsilon_0}\frac{p^4}{m^4c^7} \tag{4}$$

So, in order to minimize the losses of energy we should use a heavy elementary particle, for example muons. They are, about, 200 times heavier than electrons.
After having all the reconstructions I should run my C++ processor. In this processor I have included some filters[3]. A filter which deletes that particles which are not muons, because there are some background effects which create other particles. Also I considered that the tracks are related to a muon when the weight of the track is bigger than 0.74, more than 74% of the hits of the track belongs to the same MCParticle, so I made one more filter for these tracks.
In addition, I should mention that the magnetic field in all these results is the same, $3.5T$. The variations of the resolution with the magnetic field are described in the section 3.5.

[3]These filters can be deleted or changed. The processor can work with every kind of event. See section 4.

3.1 How to get a resolution plot

The way to get the resolution of a parameter is to compare the reconstruction information with the true one and create a histogram whose shape is a gaussian-distribution. When we have that distribution we should use a Gauss-fit in order to get σ, the value of sigma is the resolution.
If we want to create a gaussian distribution for $1/p_t$ we should create a histogram with values of

$$\frac{p_{t_{reco}} - p_{t_{true}}}{p_{t_{true}}^2} \tag{5}$$

where $p_{t_{reco}}$ has been calculated using Ω from the reconstructed track and the equation (2).
For $d0$ resolution we will take values from

$$d0_{reco} - d0_{true} \tag{6}$$

To get a good result in our fitting we should choose the limit of the horizontal axis of the histogram and the number of bins properly. We want to have a proper value for σ with a relatively small error. With the aim of selecting the correct limits for the axis for each fitting I created a ROOT macro to study the variation of σ and $\Delta\sigma$ with the axis size and number of bins for some of the data. In this way I was able to see what axis size was the best for each value of p_t and θ and write a small algorithm which can select the limits of the axis in a proper way for each data. In case you do not define properly the fit fails.
After this all we have to do is to plot our results in a figure.

3.2 Resolution

A study of the resolution will probe the sensitivity of our tools, in particular our tracking software.

3.2.1 Momentum Resolution

One of the reasons why we study the resolution of $1/p_t$ instead of the resolution of p_t is that when we create a histogram for values which are got with the expression (5) the result is a gaussian distribution whereas with the expression $p_{t_{reco}} - p_{t_{true}}$ we would get a different distribution.
As we can see in Figure 2 the $1/p_t$ resolution get worse quickly when we have smaller polar angles, for instance we can check that the difference between 20^o and 40^o is bigger than the one between 40^o and 85^o. So for non-low polar angles ($> 40^o$)the resolution does not depend so strongly on θ as for smaller values.
Also, we can check that the resolution is better for high momenta. In fact, for polar angles between 40^o and 85^o our results tend to a fix value about $1.6 \cdot 10^{-5} GeV^{-1}$. For each value of θ that fix value is different, nevertheless for non-low polar angles that value

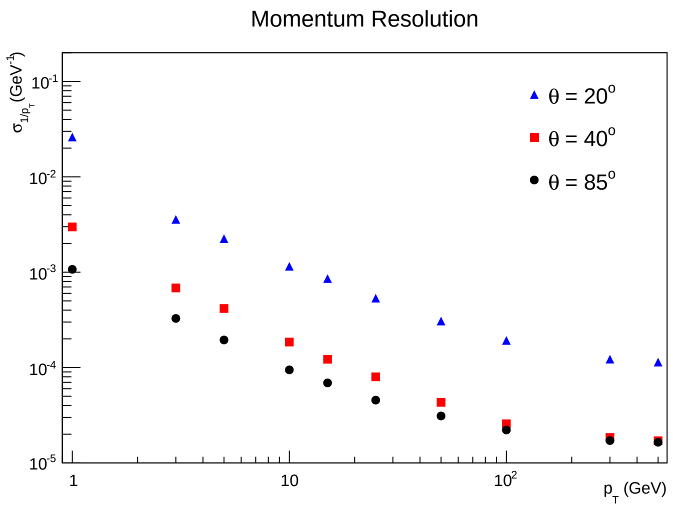

Figure 2: $1/p_t$ resolution as a function of p_t for several values of momentum and polar angle. The magnetic field in this results is $3.5T$.

is similar. In fact for $500GeV$ the resolution for 40^o is $1.7 \cdot 10^{-5}GeV$ while for 85^o it is only $10^{-6}GeV$ smaller .
The better results for non-low polar angles and momentums are due to scattering effects, these effects are less significant for high momentums than for lower ones. So, the resolution depends strongly on multiple scattering effects.

3.2.2 Impact Parameter Resolution

We study the impact parameter resolution with the aim to know how precisely we can identify the origin of the track. In our study, the ideal value of $d0$ should be zero but it is not. So the value of $d0$ describes how good is the reconstruction too, and studying the resolution we can know how precisely the tracking software calculates it.
In Figure 3 we can check that for values of momentum over $50GeV$ our results for all the polar angles are very close to each other. All they are in a range of $1.5\mu m$. If we look at differences in resolution between the different values of θ we see that these differences are huge for low momentums, from 1 to $10GeV$, in comparison with bigger momentums. For high values of momentum there is an asymptotic resolution of $2\mu m$.
If we focus on low momentum we can check that the main factor of variation is the polar angle. Impact parameter resolution will be worse for lower polar angles.
In the same way as we got worse resolution for low p_t and θ in the case of the resolution of $1/p_t$ in the case of $d0$ we have the same problem. The reason because of we have a worse resolution is the same as before, multiple scattering.

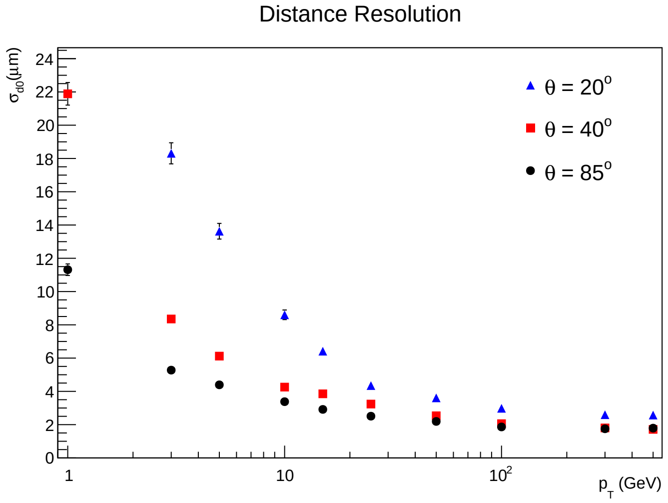

Figure 3: Impact Parameter resolution as a function of p_t for several values of momentum and polar angle. The value of σ for $1GeV$ and 20^o is $61.5\mu m$ and his error 6.2%, they are too big to appear in this figures. The magnetic field in this results is $3.5T$.

3.3 Pulls

The quality of the track fitting can be characterized by the pulls of the track parameters. The quality of a reconstructed track parameter X is given by the residual:

$$R(X_i) = X_i^{reco} - X_i^{true} \tag{7}$$

We can compare the residual with the error of the parameter and we get the pull:

$$P(X_i) = \frac{X_i^{reco} - X_i^{true}}{\Delta X_i} \tag{8}$$

Ideally, all the results which are obtained by this formula, the pull of the parameter X, should follow a Gaussian-distribution whose mean is zero and his standard deviation is one.

In our case we want to study the pulls of Ω and $d0$. We use $d0$ and Ω because they are parameteres we get directly from the track, even we use it to calculate p_t (equation 2). Moreover, we will use the error of the track fitting.

$$P(\Omega) = \frac{\Omega_{reco} - \Omega_{true}}{\Delta\Omega_{reco}},, \qquad P(d0) = \frac{d0_{reco} - d0_{true}}{\Delta d0_{reco}} \tag{9}$$

With the values that I get from this expression I have to build a histogram for each parameter. That histogram will be a gaussian-distribution whose σ would be ≈ 1.

Using table 1 we can check that for low momentum the estimation of $\Delta\Omega_{reco}$ is not the best. The values of $\sigma(\Omega)$ are not very close to 1, whereas it get better for high

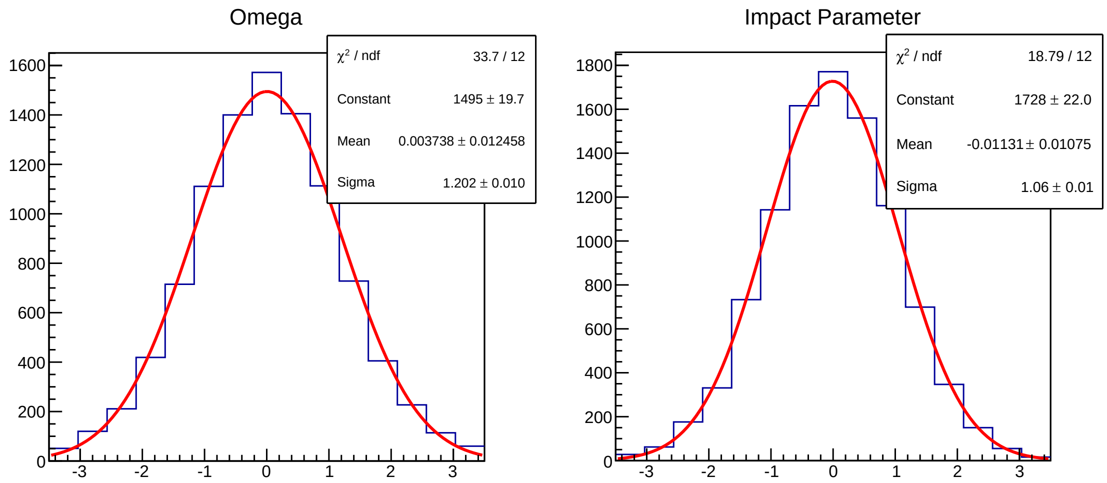

Figure 4: Gaussian-distributions created with the expressions 9 using the data of Ω and $d0$ for all the values of p_t and $\theta = 85^o$, $B = 3.5T$ and single muons.

p_t (GeV)	σ (Ω)	σ $(d0)$
1	1.67 ± 0.06	1.02 ± 0.03
3	1.34 ± 0.04	1.08 ± 0.03
5	1.29 ± 0.04	1.09 ± 0.02
10	1.18 ± 0.03	1.09 ± 0.03
15	1.12 ± 0.03	1.07 ± 0.03
25	1.12 ± 0.03	1.05 ± 0.03
50	1.12 ± 0.03	1.06 ± 0.03
100	1.10 ± 0.03	1.02 ± 0.03
300	1.01 ± 0.02	1.00 ± 0.02
500	1.00 ± 0.02	1.01 ± 0.02

Table 1: Values of σ for the gaussian-distributions created from the expressions 9 for each value of p_t. All these data are for $\theta = 85^o$, $B = 3.5T$ and single muons.

momentums. A possible explanation of these values which are not close to one can be an underestimation of the material effects during the reconstruction. However, in the case of $d0$ all the results are much more homogeneous and closer to 1.

3.4 Efficiency

In this study we have needed tracks with some requirements, because we want to use *good* ones. So, in our processor we have introduced a filter for those tracks whose weight is worse than 0.74. This means that more than 74% of the hits have been originated by

the same particle, in our case a muon, and we will considered that the track has been found. So, studying the efficiency we test the performance of our pattern recognition.
A high efficiency is a requirement for a tracking software, it means that the software is able to find the track for each particle. It is necessary to know the track of as many particle as it is possible, so we expect to have a efficiency very close to 100%.

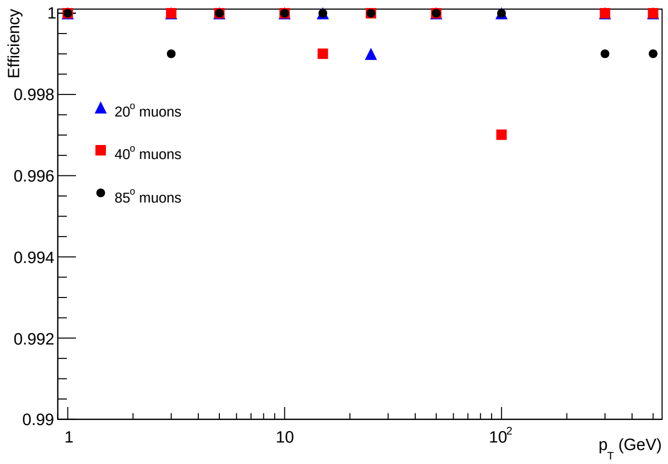

Figure 5: Fraction of tracks whose weight is bigger than 0.74. In some points the twenty-degrees-signal is behind the square.

In our case the efficiency is always 1, or almost (see figure 5). So all tracks have been found.

3.5 How significant would be a change in magnetic field?

Using the same script and ROOT macro than in the section 3.2 we can study the momentum resolution for different values of magnetic field. We only need to input the values of the BFactor ($B = BFactor \cdot 3.5T$) and momentum we want to use and run the script[4].
Figure 6 shows the resolution of $1/p_t$ for three different values of magnetic field. With these values of B we can check that the resolution is worse if we use a smaller magnetic field. Nevertheless, we can improve the resolution using a bigger detector, but the price would be higher too. So a requirement in the design of a detector is to choose the correct magnetic field and radius in order to carry out the experiments we want to do successfully and minimize the cost.

[4]This is the same script than the one which was described in the section 2.2.

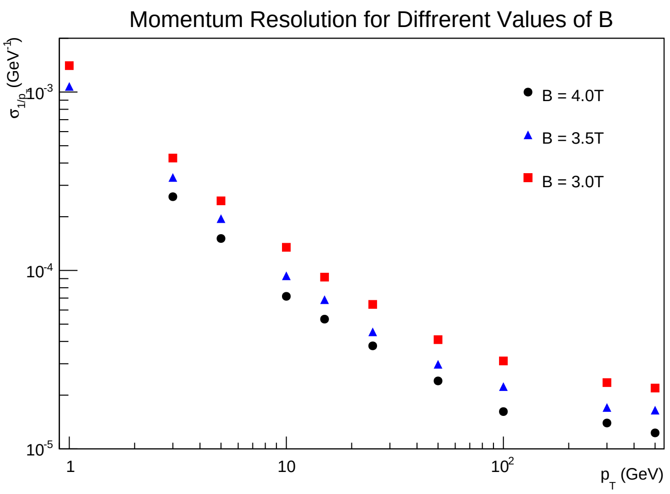

Figure 6: $1/p_t$ resolution for three different values of the magnetic field. All the errors are between 2.1% and 3.2%. The polar angle in this results is 85^o.

4 Comments on my processor

All the results in this work have been obtained for muons. Nevertheless, my processor can work for every physical event. Also, it can be used for different values of magnetic field (as in the section 3.5), momentum, direction or initial position.
Furthermore, it creates an *efficiency-vector*, this vector stores in each component the number of found tracks for a different range of momentum. A PDG-dictionary has been included too, it is useful when you do not have a list of the PDG codes close to you and you are working with different particles.
In this work we only have studied some parameters of our tracks. However, the processor I have written creates *trees*[5] for other parameters as ϕ, $z0$ or $tan\lambda$.

5 Conclusions

In this project I have created a tool which allows us to test our tracking software in an easy and fast way. This tool is formed by a C++ processor and some ROOT macros which are able to plot the resolution, pulls and efficiency. Also it has been created one script which simulates the particles we want to study and reconstructs their tracks, so this script gives us all the data that we will analyze with the processor.
So, in order to test the tracking software what we have to do is, firstly, to execute the script and the processor and then we should choose the ROOT macro we want to use. There are ROOT macros for resolution, efficiency and pulls. With these plots we can start to test the tracking method.

[5] Collections of data that can be read using ROOT.

We have studied each characteristic of the track to test a different module of the tracking software Marlin. We have studied resolution to know the accuracy, efficiency to test the pattern recognition module and pulls to test the fitting.
Using this tool, we have observed the resolution of $1/p_t$ and $d0$ using several values of momentum and polar angle. We have checked that for low values of both, p_t and θ, the resolution is worse than for high values. For the resolution of $1/p_t$, as well as for $d0$, the variation of the results for high momentums do not depend, strongly, on θ. The resolution is worse due to multiple scattering. The best results have been obtained for high values of p_t and θ, for those values the scattering effects are not so important as for smaller values of p_t or θ so we get a better resolution, efficiency and pull.
On the other hand, we have commented the changes that the resolution suffer when we use a different magnetic field. When the magnetic field is smaller the resolution is worse, one option to solve this would be to build a bigger detector in order to improve the resolution. So the best combination for an accuracy reconstruction is to have a detector with a big radius and a strong magnetic field.

References

[1] The International Large Detector, Letter of Intent

[2] Pattern recognition and event reconstruction in particle physics experiments, *R. Mankel*

[3] Tracking Performace and Momentum Resolution of the CLIC_ILD Detector for Single Muons, *Martin Killenberg, Jacopo Nardulli*

[4] Tracking Performace in High Multiplicity Environment for the CLIC_ILD Detector, *Martin Killenberg, Jacopo Nardulli*

[5] The International Linear Collider, Gateway to the Quantum Universe