Software Infrastructure for the Reconstruction of Events at

the

International Linear Collider

Kudzanayi Munetsi-Mugomba

Lincoln University, Pa

Submitted August 6, 2007

Supervised by Dr. Caroline Milstene, *International Linear Collider*

Summer Internship in Science and Technology (SIST)

Fermi National Accelerator Laboratory

P.O. Box 500 M/S 117

Batavia, IL 60510-0500

Outline

Abstract

This paper describes a way to reconstruct and test the trajectory of charged particles within a detector (in our particular case the Silicon Detector Concept SiD detector) for the linear collider. I worked on the development of an efficient software tool for the reconstruction package and tests in the detector based on the Kalman filtering. In many cases*(International Linear Collider, ILC)* particles are produced as jets, hence the identification, separation and reconstruction of the particles is of paramount importance. The reconstruction I tested takes into account the magnetic field, the material e.g. the dE/dx and the multiple scattering of the particles in the SiD detector. Approximately 80% of the particles are below 30GeV, hence an accurate reconstruction of lower energy particles is important. In the lower range of energy the dE/dx plays an important role, and this effect is emphasized by the High Magnetic Field of the SiD (5 Tesla) which enhances the importance of a good tracking tool and its testing procedure.

Introduction

A team of scientists, engineers, and computer programmers have been simulating and writing software infrastructure in an effort to reconstruct the possible events that would occur in the ILC. My supervisor, Dr. Caroline Milstene is part of this prolific team of scientists, which ardently meant that I would be a part of the team as well. The software that I helped build and test is primarily designed for the Silicon Detector concept (SiD), which has its origins in the United States. It is applied on data generated and designed at SLAC. The SiD concept incorporates Silicon calorimetry and Silicon tracking in a ***linear collider*** detector design which attempts to optimize physics performance while realistically constraining costs.

The main purpose to simulating the possible events that would occur at the ILC is to;

- optimize full detector designs for physics performance.
- establish software infrastructure for the actual detector experiments.
- understand and maximize the best possible conditions and material for the ILC.
- optimizing design structures, and incorporating new technologies.

The main detector concepts are listed as shown;

- The Silicon Detector Concept (SiD) (American origin)
  - Silicon tracker, 5T field
- Large Detector Concept (LDC) (European origin)
  - TPC (+Silicon IT), 4T field
- GLD (Asian origin)
  - TPC (+Large Silicon IT), 3T field
- Fourth concept (Fermilab Design
  - TPC (+Silicon Strips)

The ILC is a proposed global precision machine that is intended on giving physicists a new cosmic doorway in exploring energy regimes beyond the reach of today's accelerators. The ILC is an electron-positron collider, much intended to ment the Large Hadron Collider (CERN), hence working in connaissant to unlocking the deepest mysteries in the universe.

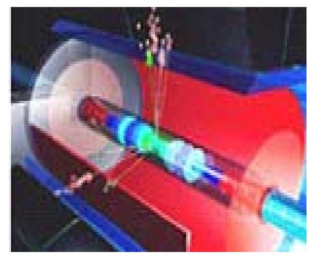
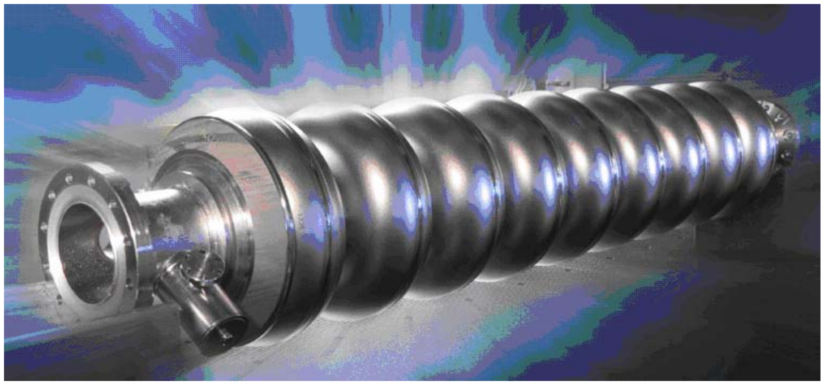


Fig 1

Particle physics has reached an extraordinary moment in the quest to understand the universe and its physical laws. Profound new questions have emerged to capture the human imagination. To address these questions, scientists all over the world are collaborating to design and build the particle accelerator of the future.

The International Linear Collider is a proposed future international particle accelerator. It would create high-energy particle collisions between electrons and positrons, their antimatter counterparts. The ILC would provide a tool for scientists to address many of the most compelling questions of the 21st century-questions about dark matter, dark energy, extra dimensions and the fundamental nature of matter, energy, space and time.



TESLA 9-cell 1.3 GHz SRF cavities from ACCEL Corp. in Germany for ILC. Fig 2

The ILC will consist of 2 linear accelerators facing each other, and hurling approximately 10 billion electrons and their anti-particles, positrons, toward each other at speeds approximated to the speed of light. The superconducting cavities operate at temperatures nearly absolute zero thus giving the particles more and more energy until they smash in a blazing crossfire at the center of the machine. The International Linear Collider will use 16,000 superconducting cavities to accelerate electrons and positrons to 99.9999999998 percent of the speed of light. Superconducting cavities have been proved to operate at higher gradient, lower AC power demand and more favorable beam dynamics conditions than comparable normal conducting resonators. The performance of the best single-cell cavities comes close to the intrinsic limitation of the superconducting material. Complete multicell structures with all auxiliaries (couplers, tuner, etc) lag behind in performance because of their complexity.

The SiD DETECTOR CONCEPT

1) The Vertex Sub detector:

A vertex detector, A detector in collider experiments that is positioned as close as possible to the collision point, and contains sparse material to minimize the energy loss by the particles, It also gives the ***most accurate location of any outgoing charged particles*** as they pass through it. It is typically made of cylindrical layers, positioned at radii of a few centimeters, the innermost layers preferably with pixel readout. The goal of a vertex detector is to measure particle tracks very close to the interaction point (inner radii of a few cm, close to the beam pipe), thus allowing one to identify those tracks that do not come from the vertex (e.g. as a signature for short-lived decaying particles). Most vertex detectors seem to be made of semiconductor detectors, but precise drift chambers have also been used successfully. The vertex detector is composed of a central barrel system with five layers and forward systems composed of 5 disks.

2) The Tracker:

The tracker is composed of five cylindrical barrels with five disk-shaped endplanes. The tracker or reconstructs charged particles' trajectories. Each charged particle that travels through the tracker and leaves hits in the active material, e.g. Silicon for the SiD, and with each hits one can construct the track.

3) The Electromagnetic Calorimeter:

The electromagnetic portion of the calorimeter is the closest calorimeter to the initial high-energy collision point. A calorimeter is a device used for calorimetry, the science of measuring the Energy of a particle through interaction with the medium on its path . This calorimeter is designed to stop/detect electrons and particles with electromagnetic interactions.

4) The Hadronic Calorimeter:

It is calorimeter optimized for incident hadrons, usually placed behind an electromagnetic calorimeter which fully contains electromagnetic showers This calorimeter is just like the EM calorimeter, but it tries to stop/detect hadrons or particles with hadronic interactions.

5) The Muon Detector:

The muon detector fulfills three basic tasks: muon identification, trigger, and momentum measurement. The high field solenoid magnet and its flux-return iron, serves as the absorber for muon identification, planes of strips of active material, e.g. scintillator, are inserted in slits in the Iron ensure the performance of these tasks. Muons tend to go through the entire detector without stopping, leaving only a trail of regular hits as it goes. At the location of the Muon detector most of the particles, besides the muons, have been stopped through interactions with the material, radiation and decay. The muon detector , therefore, detects mostly muons and hadrons which did punch through the hadron calorimeter and the coil into the Muon detector

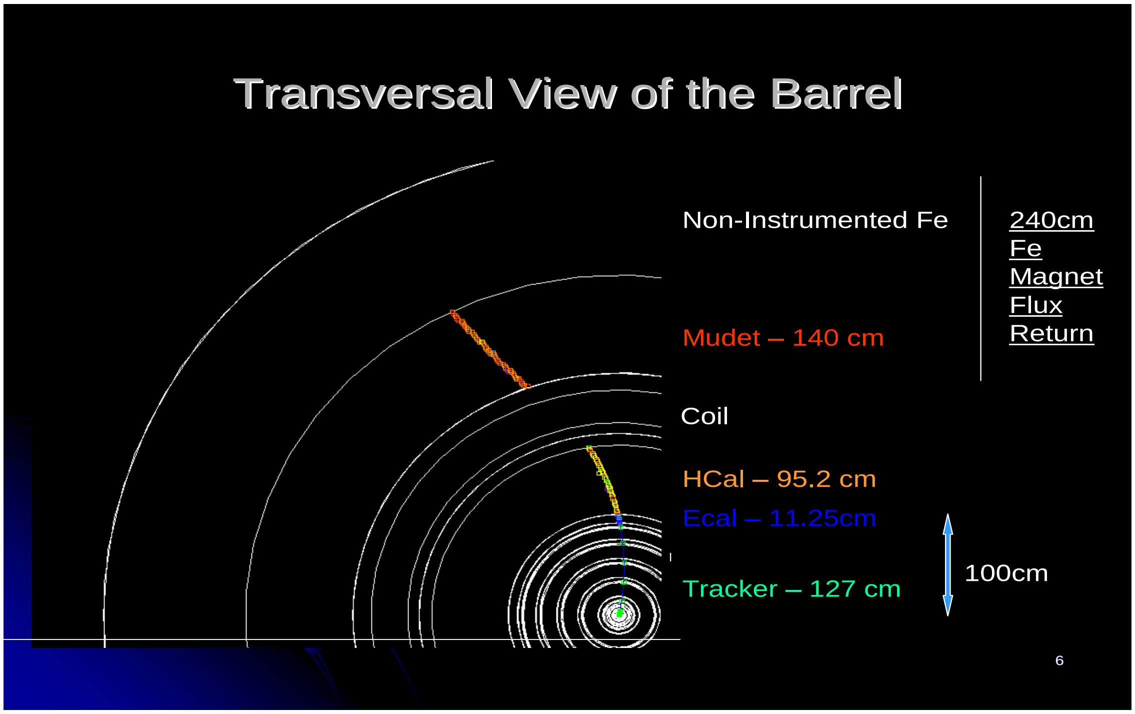


Fig 3

Concepts Learned

Radiation Length

Radiation Length , High-energy electrons predominantly lose energy in matter by **bremsstrahlung**, and high-energy photons by $e^+ e^-$ pair production. The characteristic amount of matter traversed for these related interactions is called the radiation length X_0, usually measured in gcm^{-2}. It is both the mean distance over which a high-energy electron loses all but 1/e of its energy by bremsstrahlung, and 7/9 of the mean free path for pair production by a high-energy photon. It is also the appropriate scale length for describing high-energy electromagnetic cascades.

The radiation length is given, to good approximation, by the expression

$$X_0 = \frac{716 \cdot A}{Z(Z+1)\ln\frac{287}{\sqrt{Z}}}\ g \cdot cm^{-2} \qquad \text{.....................................(1)}$$

A – cross-sectional area ,   Z – atomic number

Interaction Length

The **mean free path** of a particle before undergoing inelastic interactions in a given medium, usually designated by **λ**. The relevant cross-section is

$$\sigma_{tot} - \sigma_{el} - \sigma_{diff}$$

$$\lambda(E) = \left(\sum_i [n_i \cdot \sigma(Z_i, E)]\right)^{-1} \quad g.cm^{-2} \qquad \text{..........(2)}$$

Av	Avogadro's number (6.02486 x10^23 atoms)
Z	atomic number
A	atomic weight
rho ρ	density, g/cm^3
σ	total cross-section for the reaction, cm^2
n_i	proportion by number of the ith element in the material

Bethe-Block Equation

$$-\frac{dE}{dx} = Kz^2 \frac{Z}{A}\frac{1}{\beta^2}[\frac{1}{2}\ln\frac{2m_e c^2\beta^2\gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2}];\ MeV\ g^{-1}\ cm^{-2}$$

$K=4\pi\ m_e r_e^2 c^2$, where m_e and r_e are the mass and radius of the electron
Z,A are the atomic number and the atomic mass of the absorber,
m_e is the electron mass, c the speed of light, I the mean excitation energy(eV)
$\delta(\beta\gamma)$ is the density effect correction

The mean rate of energy-loss by ionization of relativistic charged particles (protons, alpha particles, atomic ions, but not electrons) in material they traverse. Charged particles moving through matter interact with the electrons of atoms in the material, T_{max} is the maximum kinetic energy imparted to a free electron in a single collision when the interaction excites or ionizes the atoms. This leads to an energy loss of the traveling particle which is function of β(v/c). Particles of the same velocity have similar rates of energy loss.

Developing and Testing the Software

My project was mainly centered upon reconstructing the trajectory of charged particles in a medium located in a magnetic field. The charged particle we have chosen to reconstruct is a muon since it interacts very little with matter and looses most of its energy through ionization, whereas hadrons would interact with the material of the detector.

As it passes through matter the charged particle interacts with the electric fields and typically knocks loose some of the loosely bound outer electrons of the atoms.

muons can travel large distances, however, they lose energy proportional to the amount of matter they pass through, and the energy by the charged particle through ionization, and is represented by dE/dx expressed generally in MeV/cm and is given by an empirical formula the *Bethe-Bloch formula shown in equation (3.*

The first couple of days working on this project were coupled with downloading, reading software tutorials, and setting up the required platforms and computer environments to initiate the best working environments. All the software I used is available on the web for free and in the reference section of this paper. Software I mainly used were;

Java

Java is a programming language originally developed by Sun Microsystems and released in 1995. Java applications are typically compiled to byte code, although compilation to native machine code is also possible.

Maven

Maven is a software tool for Java programming language project management and automated software build. It is similar in functionality to the Apache Ant tool (and to a lesser extent, PHP's PEAR and Perl's CPAN), but has a simpler build configuration model, based on an XML[1] format. Maven is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Appendix 1a – errors encountered, and how they were over-come

Netbeans *appendix 1b*

NetBeans refers to both a platform for the development of Java desktop applications, and an integrated development environment (IDE) developed using the NetBeans Platform. advantages of Netbeans;

- User interface management (e.g. menus and toolbars)
- Window management
- Wizard framework (supports step-by-step dialogs)
- applications can be developed from a set of modular software components called modules[2]
- application built in modules can be extended by adding new modules
- applications based on the NetBeans platform can be easily and powerfully extended by third party developers.

JAS3

Java Analysis Studio (JAS) is an object oriented data analysis package developed for the analysis of particle physics data. JAS3 is particularly notable for being a fully **AIDA**[3]-compliant data analysis system.

WIRED 4

WIRED 4 is an extensible experiment independent event display and works as a plug-in module in JAS 3 (Java Analysis Studio), a generic analysis framework. Both WIRED and JAS are written in Java.

1. The **Extensible Markup Language** (**XML**) is a general-purpose markup language
2. A module is a Java archive file that contains Java classes written to interact with the NetBeans Open APIs and a manifest that identifies it as a module.
3.Abstract Interfaces for Data Analysis (AIDA) is a set of defined interfaces and formats for representing common data analysis objects.

The Helix

Describing the dependence of the trajectory on the charged particles in the magnetic field.

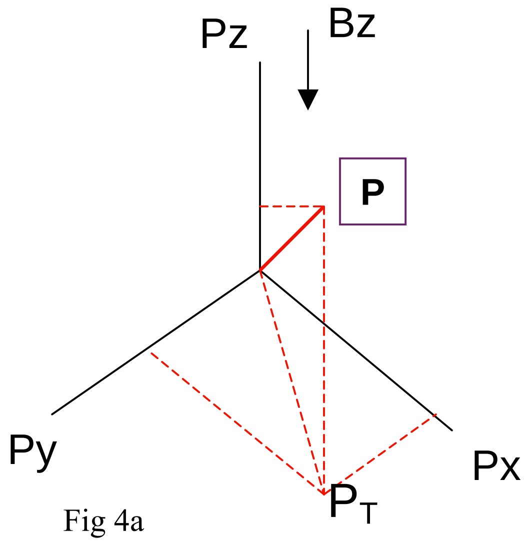


Fig 4a

For a particle with a momentum, P, the components x,y, and z are Px, Py and Pz. A magnetic field Bz, is parallel to the z-axis and acts on the Px-Py plane.

Step A

Assuming that we are in a vacuum, there are no interactions that occur within the detector, our particle only changes in the direction, but the absolute value of the velocity remains constant, this result in a circle in the Px-Py plane. But since also the particle has a momentum in the z-direction our circle then transforms itself into a helix.

Step B

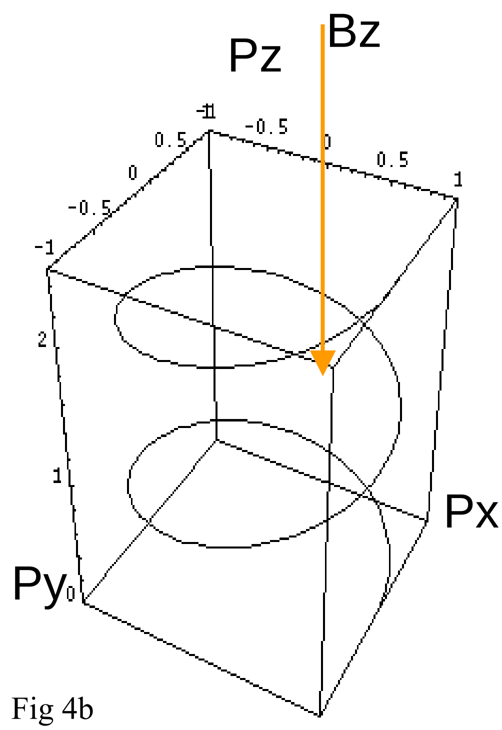


Fig 4b

The Kalman Filter – Part 1

Since the detector contains material a more elaborate algorithm is needed to represent the motion of the particle in the detector.

The Kalman Filter[1], contains a theoretical part (transport matrix), and is dependent upon Bz (magnetic field), and the dEdx[2] *(loss of energy due to the ionization of the medium by the particle)*

Transport Model

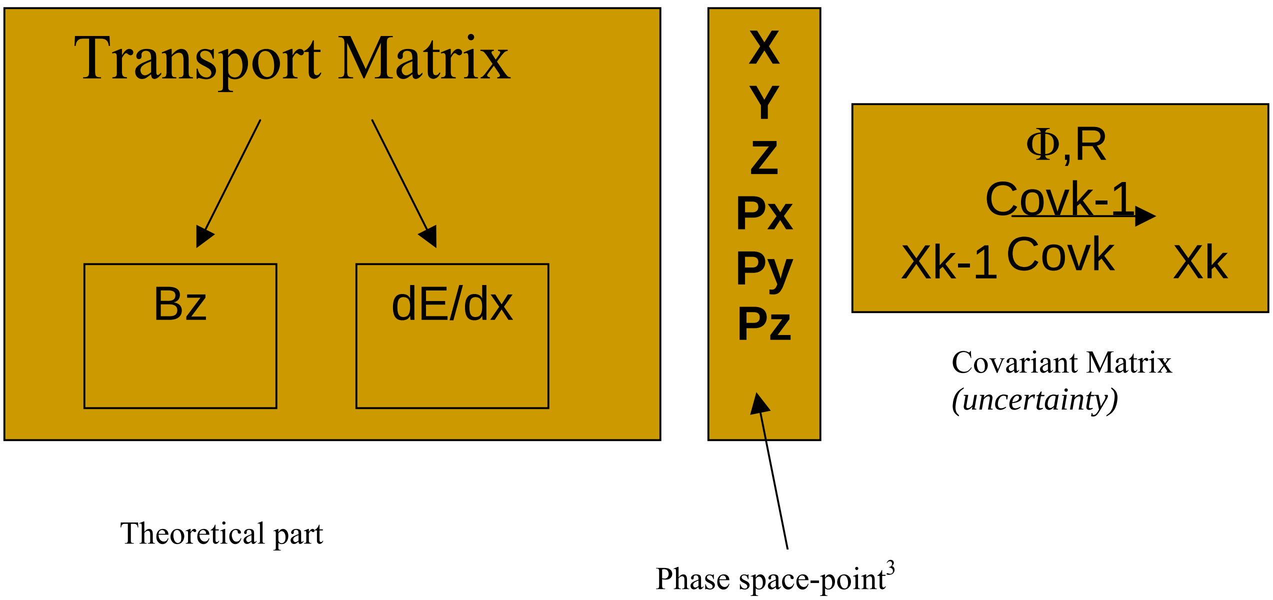


Fig 5

1. The Kalman Filter –The **Kalman** filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error.
2. Equation 3
3. Phase space-point refers to the x, y, x, Px, Py, Pz components

RESULTS

The Trajectory of a 0.5 GeV Muon - Comparing the Particle Motion to the Helix, namely dE/dx(Fe) versus dE/dx~0

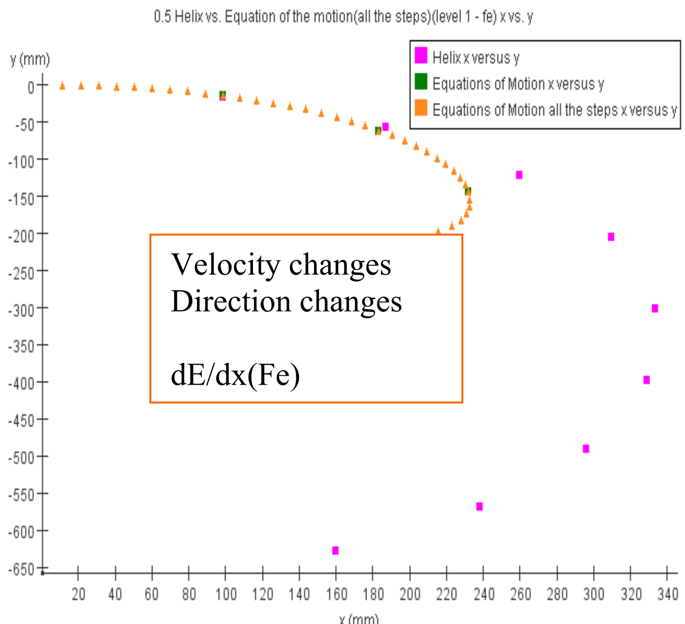


Fig 6a

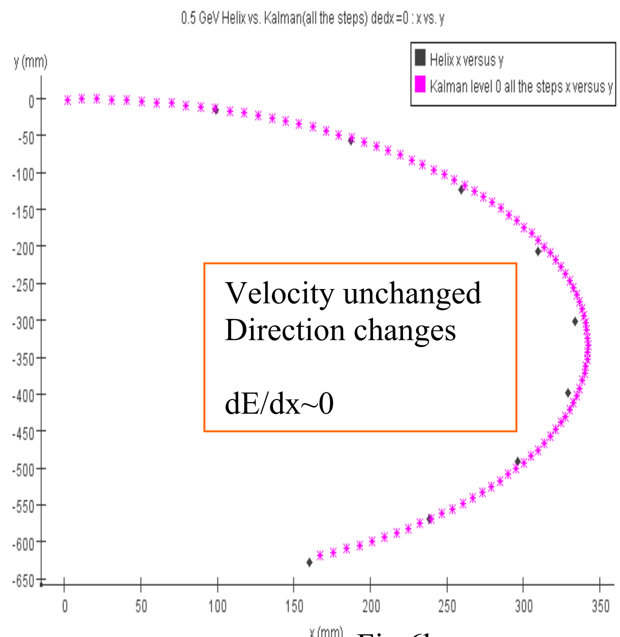


Fig 6b

Helix accounts only on the effect of the magnetic field on a charged particle.

We want to find/ measure the trajectory of a particle in the Detector hence; we account for the energy loss in the material. We realize that the Transport Model is better than the Helix, but in the material there are random processes which still are not accounted for, e.g. Bremstrahlung, Multiple Scattering, decays …

The diagram on the right , Fig 6b is a representation of the helix and our transport matrix, when we do not have any material. The Helix and our transport matrix continuously go on because we are assuming that there are no energy losses in the material.

For Fig 6a, the brown dots represent the hits of the particle within our absorber until their eventual loss in energy.

The diagrams represent a comparison when dE/dx for our material is approximately 0, and when the dE/dx is Iron (Fe). Here, we see that at higher energies of 5GeV our Helix, and our transport matrix results are directly above/ on top of each other.

The Trajectory of a 5 GeV Muon: Motion Comparison with the Helix for dE/dx=Fe and dE/dx~0

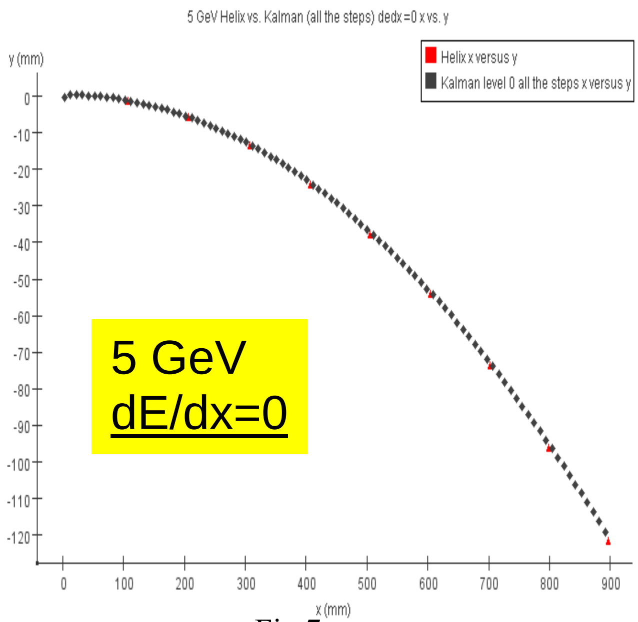


Fig 7a

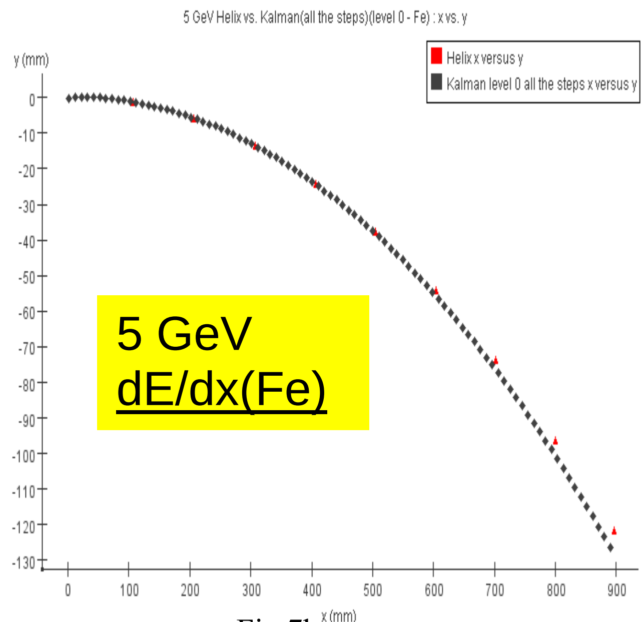


Fig 7b

For the diagrams that follow below using 2 different materials, the graph represented when our dE/dx is Aluminum, Al is much smoother as opposed to the graph represented when our dE/dx is Iron, Fe because of the different values in the dE/dx. We see that the dE/dx, (loss of energy due to the ionization of the material) for Al is almost half that of the dE/dx of Fe, hence smoother curve. We also see that the continuous dots representing our Helix continue on because for our Helix model we do not assume any energy losses.

Representations of the Equations of motion used are available in **Appendix C**

The Trajectory of a 0.5 GeV Muon: Comparison for 2 different Materials dE/dx(Al) versus dE/dx(Fe)

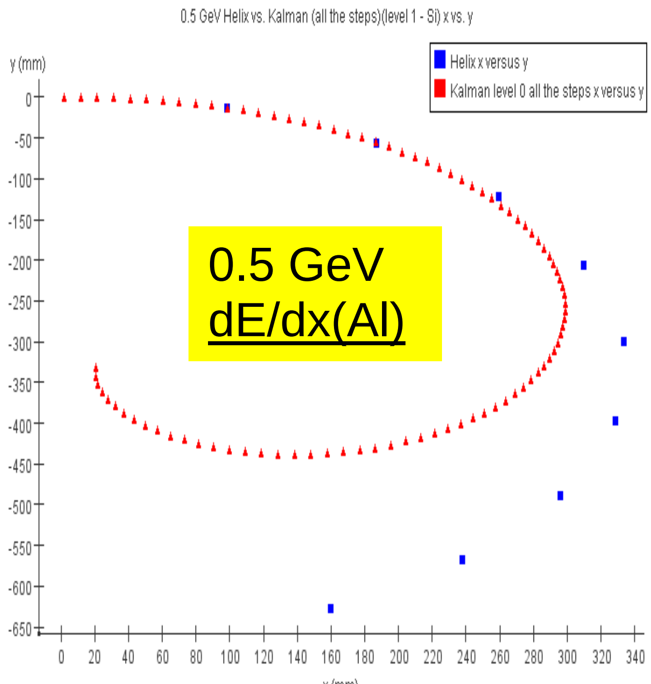


Fig 8a

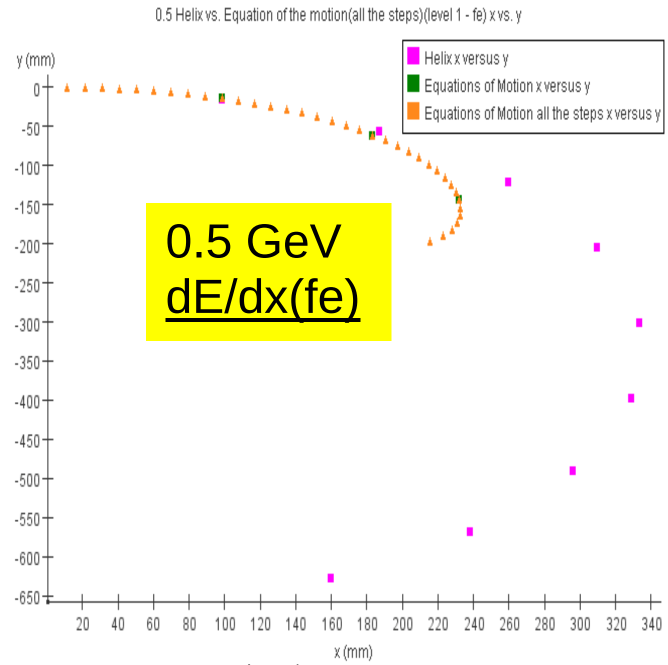


Fig 8b

Updated Transport Model

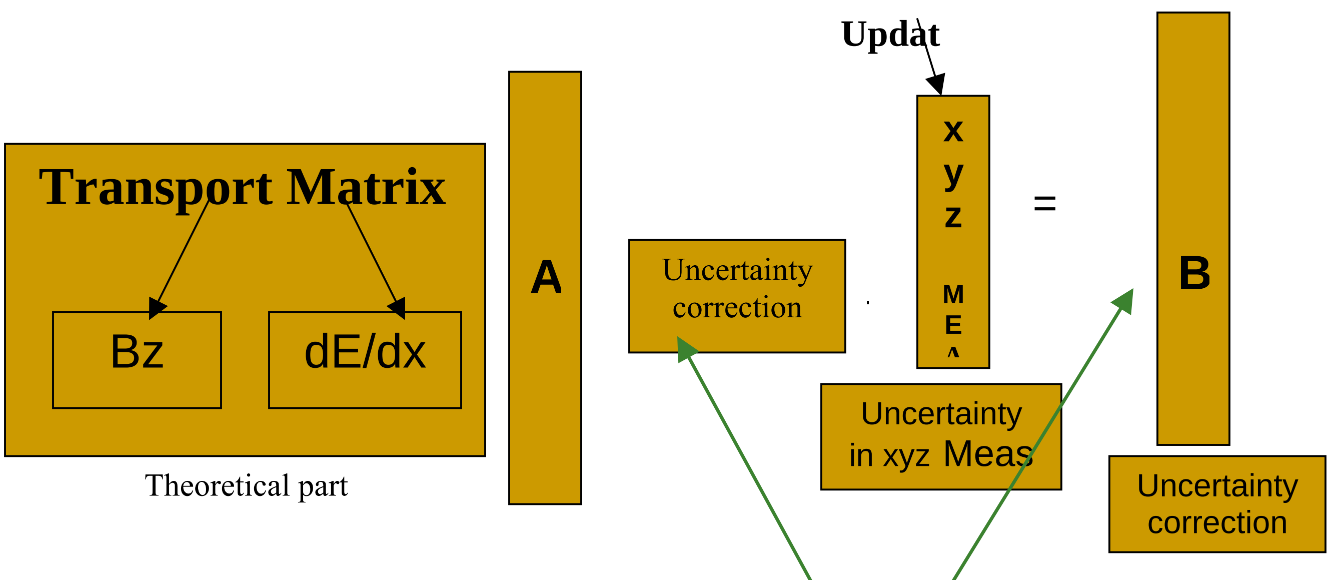


Fig 8c

Using the actual data obtained from the Active Material (measured point xyz) we are now able to combine the data obtained from our Transport Matrix with that of the measured space point of to obtain a new phase-space point.

Explaining the Transport of the Precision Information

To improve the Model by correcting the trajectory obtained using our measurement points.

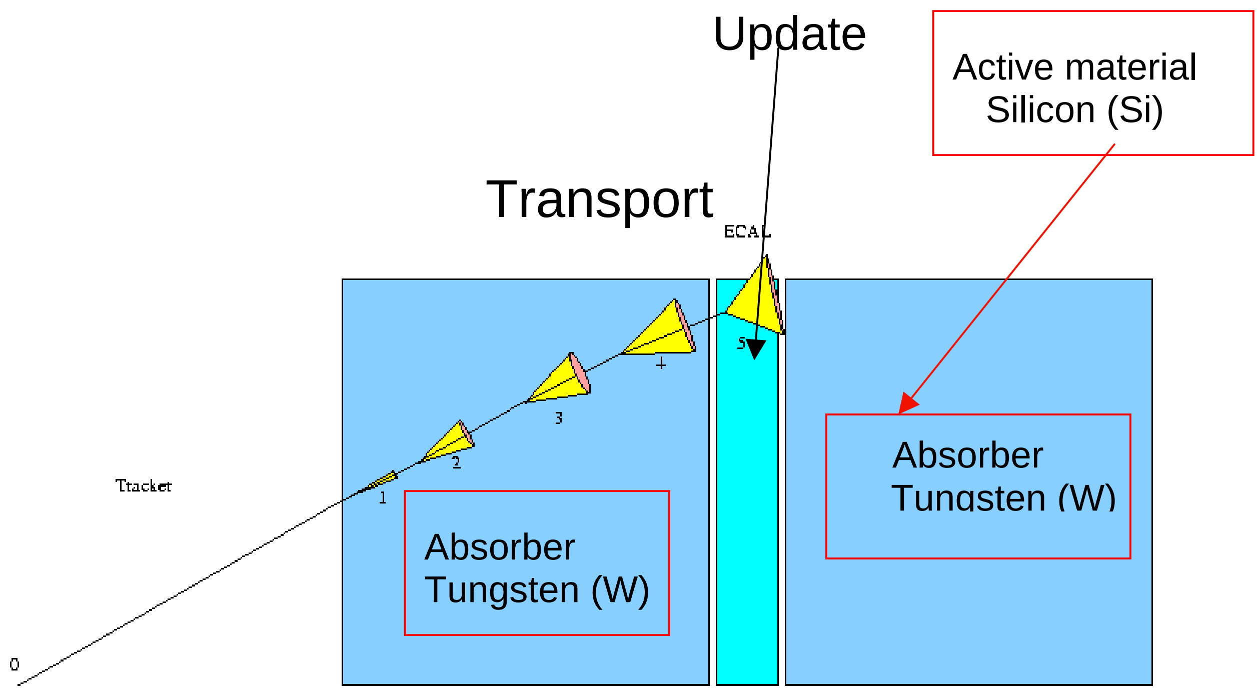


Fig 9

The yellow cones are a representation of the uncertainty (covariant matrix), as our particle which is a muon traverses from the interaction point through the tracker and the absorber leaving hits in the material which I connect to reconstruct the track of the particle.

Understanding the Results

In order to be able to test the software for various detector setups while insuring the stability of the software, I developed a set of tests for the Kalman Filter. To make it simple and without any external input, from now on, I will be using the Helix as data on which to correct the particle trajectory at update.

Hadron Calorimeter HCal setup;

- assuming that the absorber thickness of iron(Fe)=2cm
- the number of steps in the absorber =4
- number of tracks =1

0.5 GeV Muon Trajectory- Kalman Filter (Material -Fe) using the HELIX as DATA

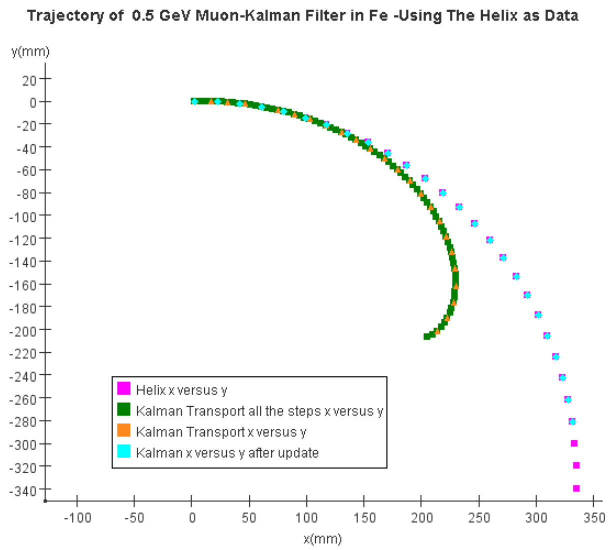


Fig 10a

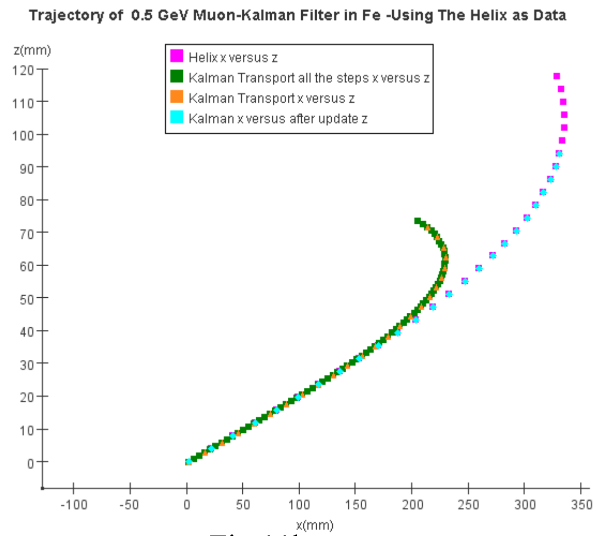


Fig 11b

Muon Detector, MuDet setup;

- the absorber is Iron(Fe) – 10 cm thick
- the number of steps = 10
- the number of tracks = 1

The Trajectory of a 5GeV Muon- Kalman Stepper updated on the Helix as Data

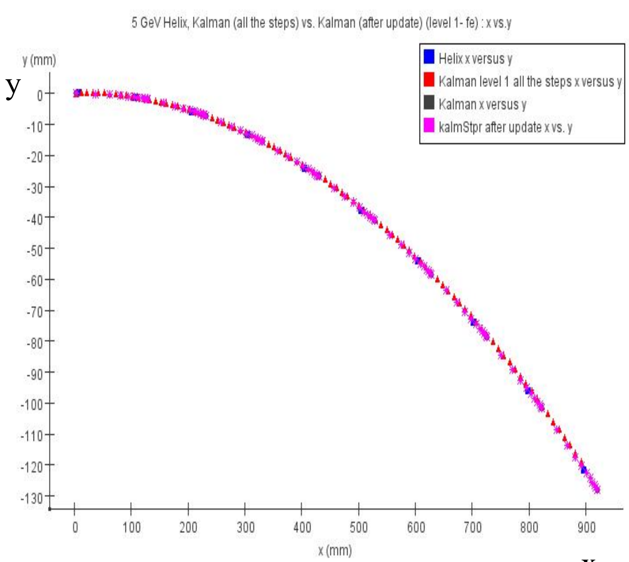


Fig 12a

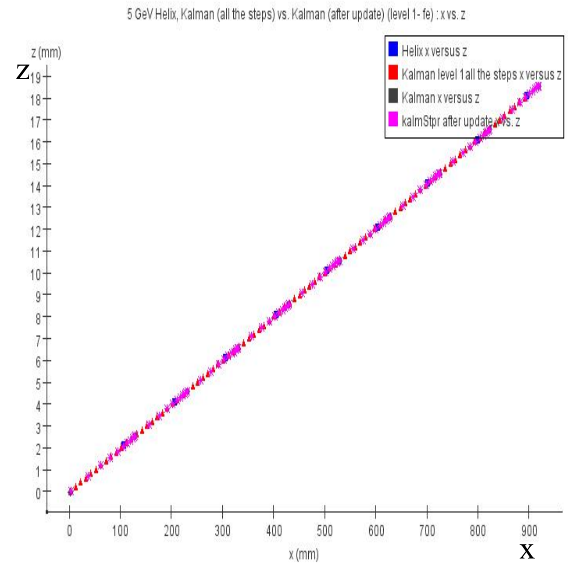


Fig 12b

Showing the Kalman Filter's Dependence on Energy

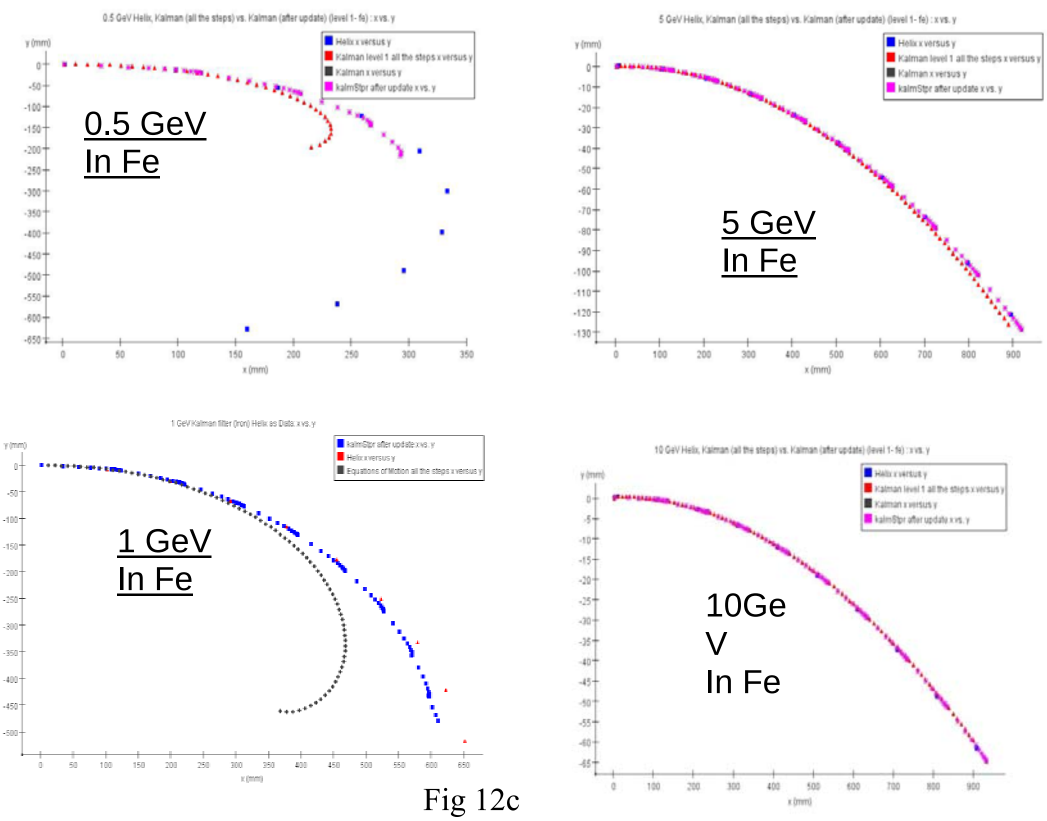


Fig 12c

Muon-x versus y: Kalman Stepper updated on the Helix as Data with dE/dx(Fe). At 0.5 GeV the track stopped. Using 10cm thick Fe absorber as in MuDet

The Following Results are for the Kalman Filter on 20 tracks at 2 different energies;

-the 20 tracks were obtained by smearing the hits of the Helix in a Gaussian.

- the trajectories of the particles are starting from an interaction point smeared in a Gaussian

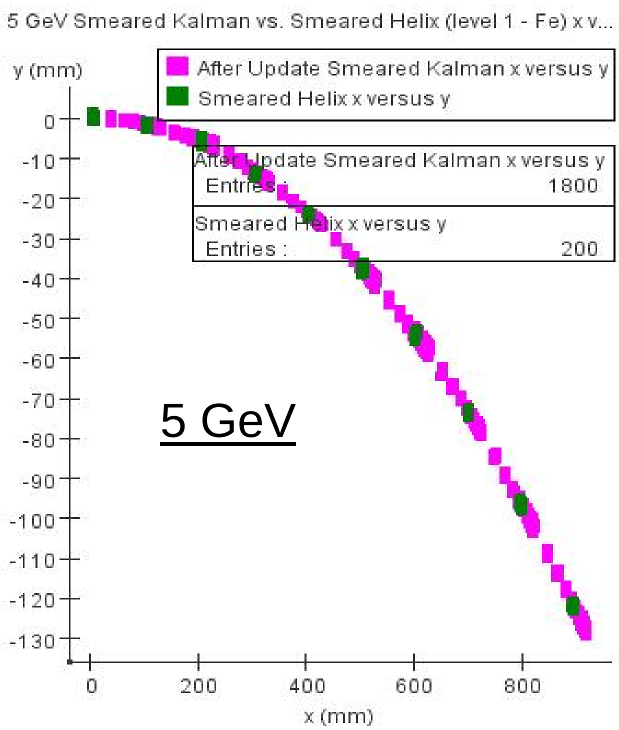


Fig 14a

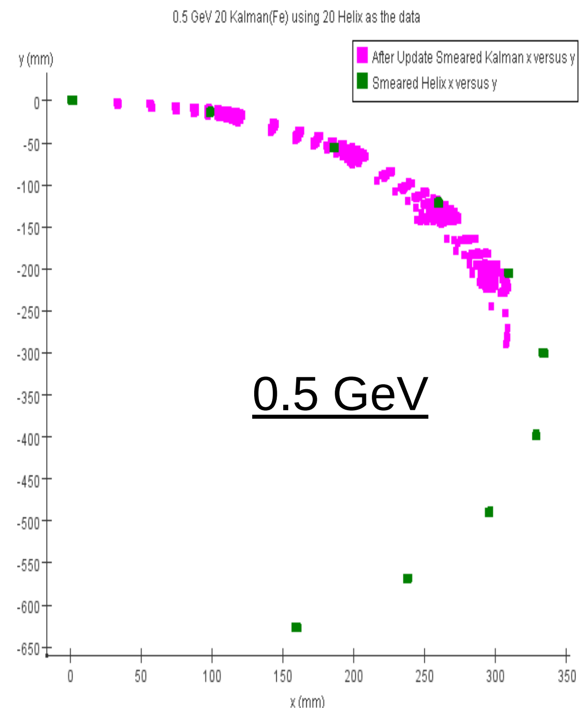


Fig 14b

Kalman Filter Application

The nonlinear problem of tracking and correcting the trajectory of a satellite over a time is difficult with the recognition of modeling errors and ground site radar tracking errors.
An accurate modeling program with the fidelity to correct for any errors in orbital motion and predict the most accurate position at some future time is required.
Here we see the correction of the trajectory of a satellite during its motion in space using the Kalman Filter

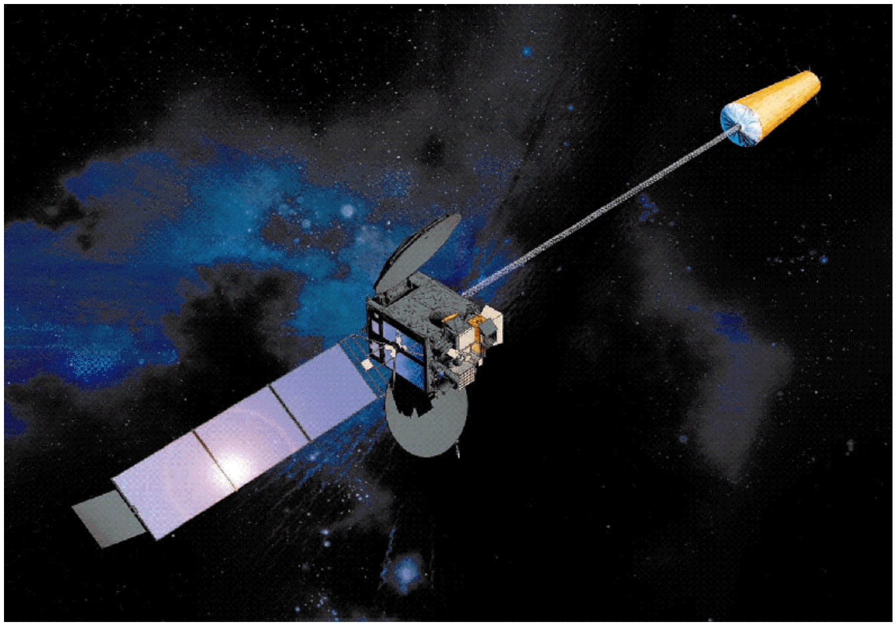


Fig 14

Conclusion

The helix was a good approximation for the trajectory of high momenta particles or low dE/dx materials. Applying the transport matrix alone, did take are of both the magnetic field effects and material effects through dE/dx and is a better approximation of the trajectory than the Helix. However other random effects are not accounted for, e.g. multiple-scattering, Bremsstrahlung, etc. But we have shown that after the inclusion of the update one is able to correct the trajectory of the reconstructed track to fit the actual data and account for all the random processes. Successfully developed an efficient algorithm that helps build and test the reconstruction of charged particle trajectories. This algorithm can be used to reconstruct particles trajectories inside jets.

I now understand the interplay between the properties of the particles we detect and the material in the detector used to detect them. I have Understood the method used to reconstruct particle trajectory in various sub-detectors i.e. Kalman Filtering and its applications

Acknowledgements

- Dr. Caroline Milstene (my supervisor, and mentor for this project)
- Dr. Marcel Demarteau (for having made it possible)
- Dianne Engram (Head, Equal Opportunity Office)
- Dr. Elliot Mcrory (program director)
- Dr. Jamieson Olsen (secondary supervisor)
- Dr. Mayling Wong (secondary supervisor)
- Dr. James Davenport (program coordinator)
- Lincoln University Physics Department (for giving me support and encouragement)
- A special thanks to Fermilab for granting me this opportunity to conduct research in the dynamic world of High Energy Physics.

References:

Milstene, Fisk, et al. Para. Muon ID at the ILC. FNAL: Fermilab, 2006
http://www.linearcollider.org/cms/
http://rkb.home.cern.ch/rkb/PH14pp/node202.html
http://www2.slac.stanford.edu/vvc/theory/calorimeter.html
http://www.javacoffeebreak.com/articles/toptenerrors.html
http://aspectj-netbeans.sourceforge.net/doc/netbeans-building.gif
http://www.fnal.gov/pub/about/contributions/index.html
http://www.linearcollider.org/cms/?pid=1000000
http://ilc.fnal.gov/about/index.html
https://confluence.slac.stanford.edu/display/ilc/ILC+Detector+Simulation+FAQ
http://www.interactions.org
http://www.stormingmedia.us/75/7501/A750134.html

Appendices

Appendix 1a

Accessing non-static member variables from static methods (such as main)

Many programmers, particularly when first introduced to Java, have problems with accessing member variables from their *main* method. The method signature for main is marked static - meaning that we don't need to create an instance of the class to invoke the main method. For example, a Java Virtual Machine (JVM) could call the class MyApplication like this :-

MyApplication.main (command_line_args);

This means, however, that there isn't an instance of MyApplication - it doesn't have any member variables to access! Take for example the following application, which will generate a compiler error message.

```
public class StaticDemo
{
        public String my_member_variable = "somedata";
        public static void main (String args[])
        {
                                // Access a non-static member from static method
                System.out.println ("This generates a compiler error" +
                                        my_member_variable );
        }
}
```

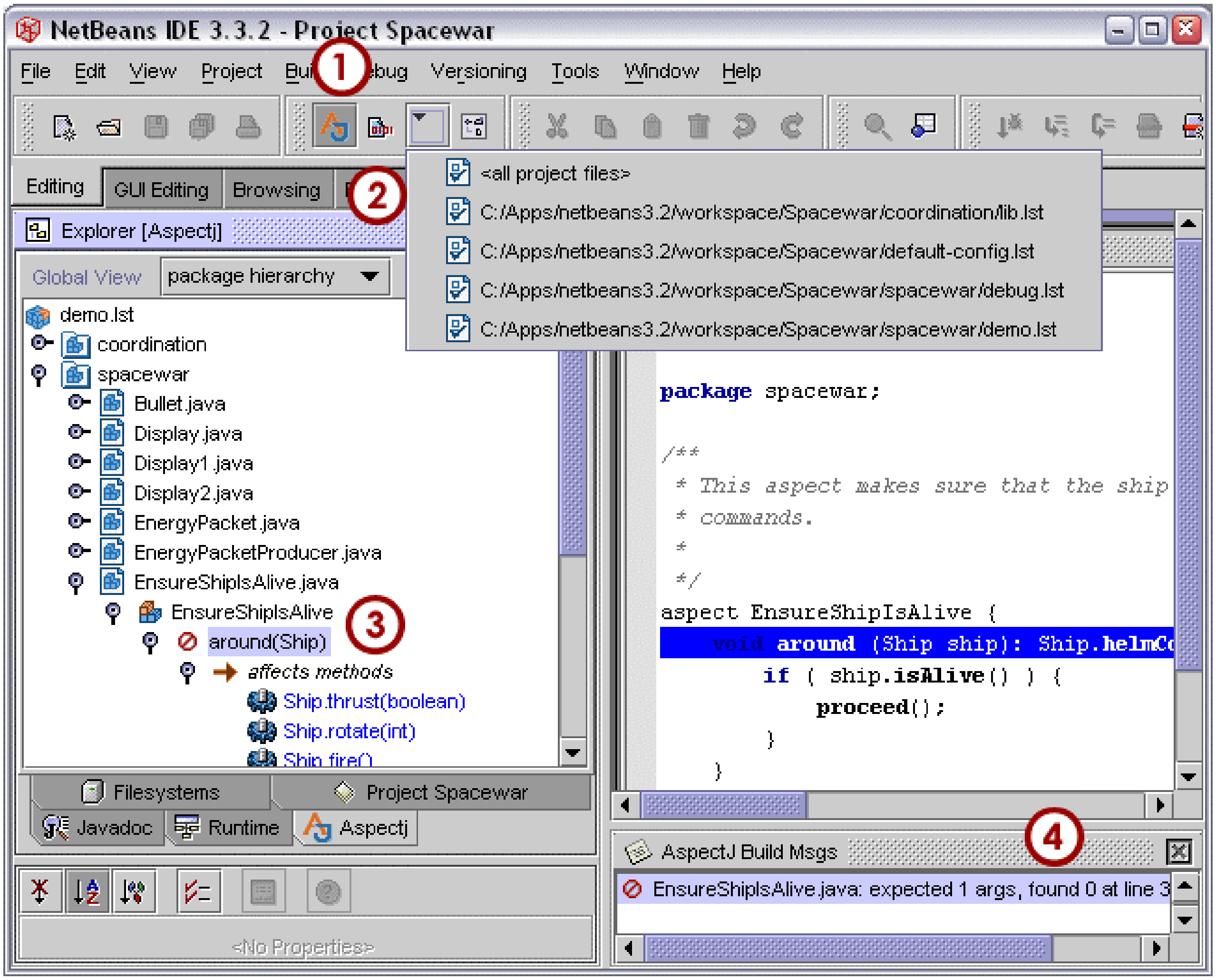
If you want to access its member variables from a non-static method (like *main*), you must create an instance of the object. Here's a simple example of how to correctly write code to access non-static member variables, by first creating an instance of the object.

```
public class NonStaticDemo
{
        public String my_member_variable = "somedata";

        public static void main (String args[])
        {
                NonStaticDemo demo = new NonStaticDemo();

                                // Access member variable of demo
                System.out.println ("This WON'T generate an error" +
                        demo.my_member_variable );
        }
}
```

Appendix 1b


NetBeans IDE 3.3.2 - Project Spacewar
File Edit View Project Build Debug Versioning Tools Window Help
1
2
3
4
Editing
GUI Editing
Browsing
Explorer [Aspectj]
Global View
package hierarchy
<all project files>
C:/Apps/netbeans3.2/workspace/Spacewar/coordination/lib.lst
C:/Apps/netbeans3.2/workspace/Spacewar/default-config.lst
C:/Apps/netbeans3.2/workspace/Spacewar/spacewar/debug.lst
C:/Apps/netbeans3.2/workspace/Spacewar/spacewar/demo.lst
demo.lst
coordination
spacewar
Bullet.java
Display.java
Display1.java
Display2.java
EnergyPacket.java
EnergyPacketProducer.java
EnsureShipIsAlive.java
EnsureShipIsAlive
around(Ship)
affects methods
Ship.thrust(boolean)
Ship.rotate(int)
Ship.fire()
package spacewar;
/**
* This aspect makes sure that the ship
* commands.
*
*/
aspect EnsureShipIsAlive {
void around (Ship ship): Ship.helmC
if (ship.isAlive()) {
proceed();
}
}
Filesystems
Project Spacewar
Javadoc
Runtime
Aspectj
AspectJ Build Msgs
EnsureShipIsAlive.java: expected 1 args, found 0 at line 3
<No Properties>

Appendix 1c

Equations Of Motions –Function **(Bz dE/dx)**
arXiv:physics.inst.det/0604197-C. Milstene, G. Fisk, A. Para

$$p_x(n+1) = p_x(n) - 0.3*q*\frac{p_y}{E(n)}*c_{light}*B_z*\delta t(n) - \gamma_x(n)\ ;$$

$$p_y(n+1) = p_y(n) + 0.3*q*\frac{p_x}{E(n)}*c_{light}*B_z*\delta t(n) - \gamma_y(n);$$

$$p_z(n+1) = p_z(n) - \gamma_z(n);$$

$$\gamma_i(n) = \frac{dE}{dx}*\frac{E(n)}{|p(n)|}*\frac{p_i(n)}{|p(n)|}*\delta s;\ i = x, y, z.$$

The point (x(n+1),y(n+1),z(n+1)) is the position at step n+1, after the momentum change to px,y,z(n+1) at step n.

$$x(n+1) = x(n) + \frac{p_x(n+1)}{E(n+1)}*c_{light}*\delta t(n)\ ;$$

$$y(n+1) = y(n) + \frac{p_y(n+1)}{E(n+1)}*c_{light}*\delta t(n)\ ;$$

$$z(n+1) = z(n) + \frac{p_z(n+1)}{E(n+1)}*c_{light}*\delta t(n)\ .$$