Track Reconstruction of Time Projection Chamber

Bo LI

Center for High Energy Physics,
Tsinghua Univ.

CEPC Training, IHEP
Oct. 20, 2013

Outline

Outline

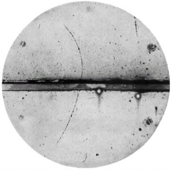
1. Introduction
   - Tracking detector
   - Time Projection Chamber
2. Track recostruction
3. MarlinTPC
4. Summary

History of tracking detector I

- Cloud chamber

(a) Wilson

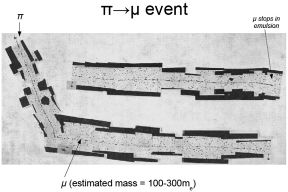
(b) Anderson



(c) Positron

- Nuclear emulsion: very high spatial resolution $\sim 1\ \mu\text{m}$

(a) Powell

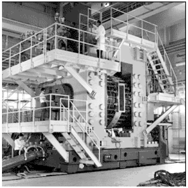


(b) π decay

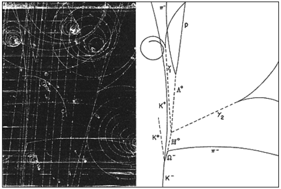
History of tracking detector II

- Bubble chamber

(a) Glaser

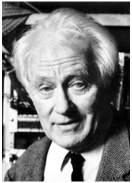


(b) B.C.

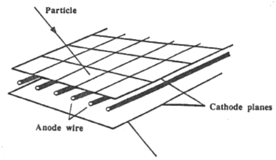


(c) Ω^-

- Multi Wire Proportional Chamber: electronic readout $\Longrightarrow$ fast



(a) Charpak



(b) MWPC

Time Projection Chamber (TPC)

- TPC is a modern tracking detector invented by D. Nygren in 1970s[Kle98]

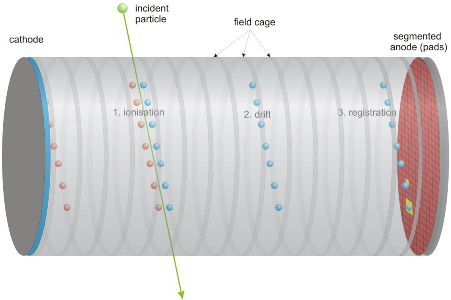


Figure 1 : The principle of TPC

- Advantages: good position and dE/dx resolution; high detection efficiency; low material budget
- Disadvantages: readout relatively slow; Ion feed back

Challenge on LCTPC

- High precision measurement at ILC requires the TPC momentum resolution $\delta(1/p_{\mathrm{t}}) = 9 \times 10^{-5} (\mathrm{GeV}/c)^{-1}$.
- The momentum resolution of a tracking detector is

$$\frac{\sigma_{p_{\mathrm{t}}}}{p_{\mathrm{t}}^2} = \frac{\sigma_{r\phi}}{0.3 L^2 B} \sqrt{\frac{720}{N+4}} \left(\frac{\mathrm{T} \cdot \mathrm{m}}{\mathrm{GeV}/c} \right),$$

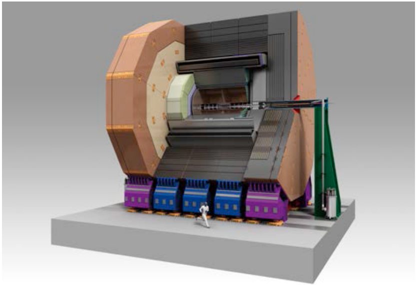
then the spatial resolution should be better than $100\ \mu\mathrm{m}$!.

- Spatial resolution:

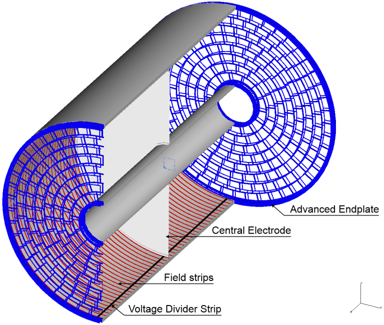
$$\sigma_{r\phi} = \sqrt{\sigma_0^2 + \frac{C_{\mathrm{D}}^2 z}{N_{\mathrm{eff}}}},$$

where σ_0 is related to electronics and readout, diffusion constant C_{D} and effective electron number N_{eff} are related to gas properties.

Design of LCTPC



(a) ILD



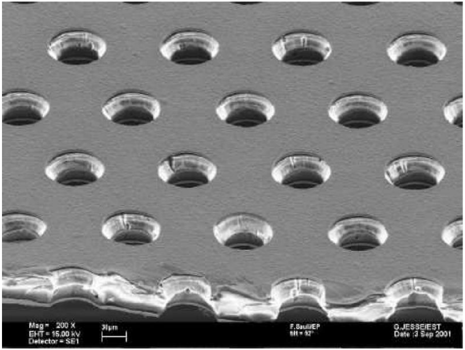
(b) LCTPC

Design parameters of LCTPC

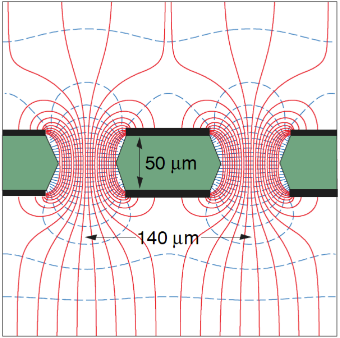
Parameter			
Geometrical parameters	r_{in}	r_{out}	z
	329 mm	1808 mm	± 2350 mm
Solid angle coverage	up to $\cos\theta \simeq 0.98$ (10 pad rows)		
TPC material budget	$\simeq 0.05\ X_0$ including outer fieldcage in r		
	$< 0.25\ X_0$ for readout endcaps in z		
Number of pads/timebuckets	$\simeq$ 1-2 $\times\ 10^6$/1000 per endcap		
Pad pitch/ no.padrows	$\simeq 1\times$ 6 mm^2 for 220 padrows		
σ_{point} in $r\phi$	$\simeq 60\ \mu$m for zero drift, $< 100\ \mu$m overall		
σ_{point} in rz	$\simeq 0.4 - 1.4$ mm (for zero – full drift)		
2-hit resolution in $r\phi$	$\simeq 2$ mm		
2-hit resolution in rz	$\simeq 6$ mm		
dE/dx resolution	$\simeq 5$ %		
Momentum resolution at B=3.5 T	$\delta(1/p_t) \simeq 10^{-4}$/GeV/c (TPC only)		

Readout devices

- MWPC: $E \times B$ effect; ion feed back
- Micro Pattern Gaseous Detector(MPGD): high spatial resolution, small $E \times B$ effect and ion feed back
  - Gas Electron Multiplier(GEM)
  - Micro Mesh Gaseous Detector(Micromegas)

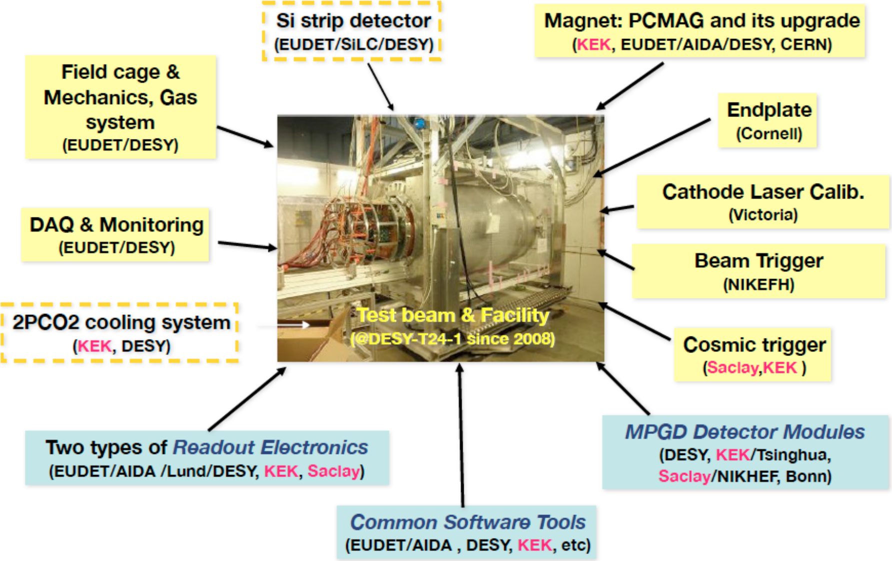


(a) GEM



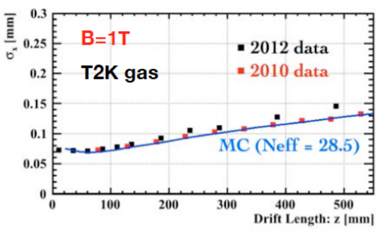
(b) Electric field

Large Prototype (LP)

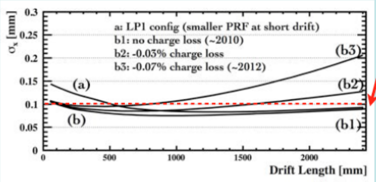


Beam test results of LP

Asian GEM Module

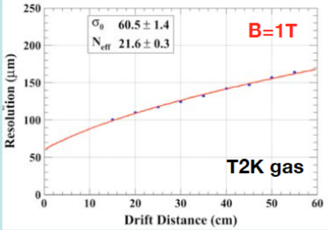


Extrapolation to B=3.5T

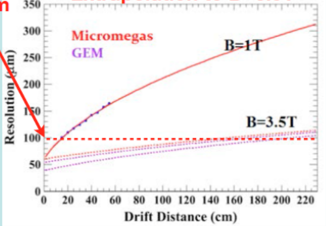


$\sigma_{r\phi}$=100 μm

Saclay-Carleton MM Module



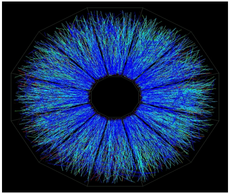
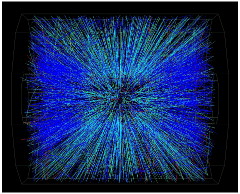
Extrapolation to B=3.5T



Outline

Track reconstruction

- The necessity of track reconstruction by computer program: get track information automatically and precisely.

- Reconstruction algorithm[FR00]
  - **Track finding** identifies whether a hit belongs to a candidate track, i.e. Pattern Recognition. Algorithms: Hough transformation and neural networks, track following etc.
  - **Track fitting** calculates the track parameter from the found hits. A fitting algorithm used often is the method of Least Square:

$$\chi^2 = \sum_{i=1}^{N} \frac{(\boldsymbol{f}(\boldsymbol{a};\boldsymbol{x}_i) - \boldsymbol{m}_i)^2}{\boldsymbol{\sigma}_i^2} \equiv min \implies \boldsymbol{a} \quad (1)$$

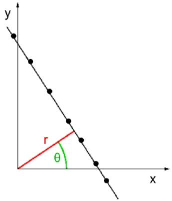
Hough transformation

- Parameterization of straight lines:

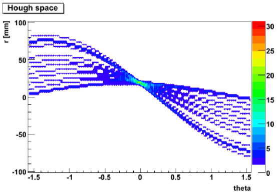
$$y(x) = -\frac{\cos\theta}{\sin\theta}x + \frac{r}{\sin\theta}$$

- Hough transformation:

$$r(\theta) = \cos\theta \cdot x + \sin\theta \cdot y$$



(a) Hits

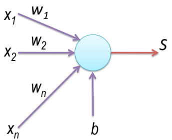


(b) Hough space

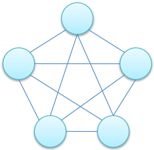
- Pathfinder in ILCSoft: for straight line and helix reconstruction

Neural network I

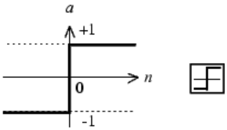
- Perceptron: $S = f\left(\sum w_i x_i + b\right)$



- Hopfield network
  - The neurons are interconnected, $w_{ij} = w_{ji}$
  - the state of a neuron is -1 or 1, $S_i = \mathrm{sgn}\left(\sum_j w_{ij} S_j\right)$



(a) Network



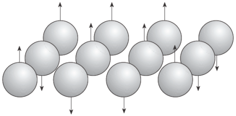
(b) Transaction function

Neural network II

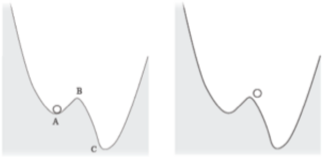
- System energy: $E_i = -\frac{1}{2}\sum_{ij} w_{ij} S_i S_j$
- The energy function evolves into a local minimum under interaction

$$\begin{aligned} \Delta E =& E' - E = -\frac{1}{2}\left(\sum_j w_{ij} S'_i S_j - \sum_j w_{ij} S_i S_j\right) \\ =& -\frac{1}{2}(S'_i - S_i)\sum_j w_{ij} S_j \implies \Delta E \le 0 \end{aligned}$$

- To reach a global minimum, simulated annealing (or Mean Field Theory approximation) is introduced.



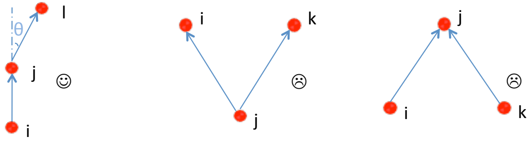
(a) Energy



(b) Annealing

Track finding with Hopfield network

- Typical connections in network[PR92]



- Definition of neuron
  - Connected from i to j: $S_{ij} = 1$
  - Otherwise $S_{ij} = 0$
- Enegy equation

$$E = -\frac{1}{2}\sum_{ijkl}\delta_{jk}\frac{\cos^m\theta_{ijl}}{d_{ij}+d_{kl}}S_{ij}S_{kl} + \frac{1}{2}\alpha\left(\sum_{i\neq k}S_{ji}S_{jk} + \sum_{i\neq k}S_{ij}S_{kj}\right) + \frac{1}{2}\beta\left(\sum_{kl}S_{kl} - N\right)^2$$

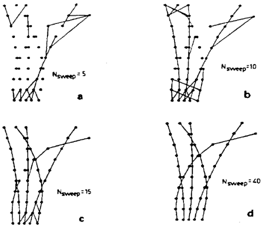
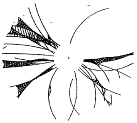
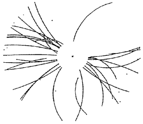


Figure 2 : Track segments at different evolution stages



(a) Network



(b) Tracks

Figure 3 : $Z^0 \rightarrow hadron$ event in ALEPH TPC

Track following

- Tracking following is a layer-by-layer tracking algorithm:

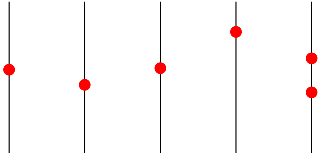


Figure 4 : A progress to demonstrate track following

Track following

- Tracking following is a layer-by-layer tracking algorithm:

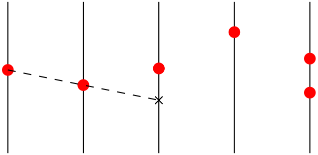


Figure 4 : A progress to demonstrate track following

Track following

- Tracking following is a layer-by-layer tracking algorithm:

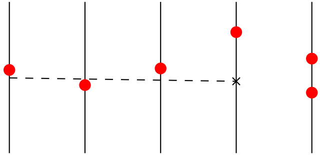


Figure 4 : A progress to demonstrate track following

Track following

- Tracking following is a layer-by-layer tracking algorithm:

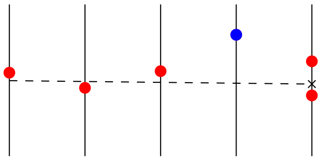


Figure 4 : A progress to demonstrate track following

Track following

- Tracking following is a layer-by-layer tracking algorithm:

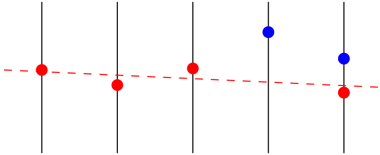


Figure 4 : A progress to demonstrate track following

Track following

- Tracking following is a layer-by-layer tracking algorithm:

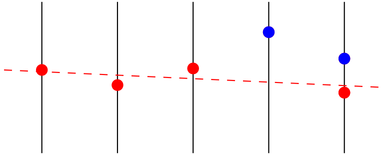


Figure 4 : A progress to demonstrate track following

- Three elements of track following[Man04]:

Track following

- Tracking following is a layer-by-layer tracking algorithm:

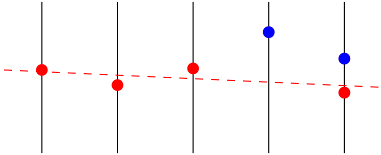


Figure 4 : A progress to demonstrate track following

- Three elements of track following[Man04]:
  - track seed: combination of hits at nearby layers

Track following

- Tracking following is a layer-by-layer tracking algorithm:

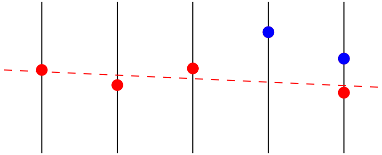


Figure 4 : A progress to demonstrate track following

- Three elements of track following[Man04]:
  - track seed: combination of hits at nearby layers
  - track model: straight line; **helix**

Track following

- Tracking following is a layer-by-layer tracking algorithm:

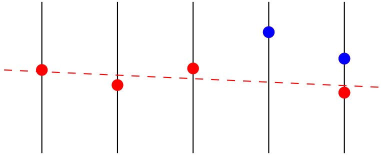


Figure 4 : A progress to demonstrate track following

- Three elements of track following[Man04]:
  - track seed: combination of hits at nearby layers
  - track model: straight line; **helix**
  - track quality criterion: hit number N and $\chi^2 \implies \mathrm{Prob}(\chi^2, N)$

Track following

- Tracking following is a layer-by-layer tracking algorithm:

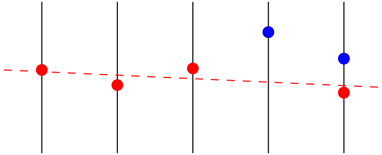


Figure 4 : A progress to demonstrate track following

- Three elements of track following[Man04]:
  - track seed: combination of hits at nearby layers
  - track model: straight line; **helix**
  - track quality criterion: hit number N and $\chi^2 \implies \mathrm{Prob}(\chi^2, N)$
- Track parameters can be updated dynamically by **Kalman filter**. The algorithm can also give $\delta\chi^2$ for selecting hit.

Assume the z axis of coordinate is parallel with the uniform magnetic field, helix in xy plane can be plot by the following figure[Fuj]:

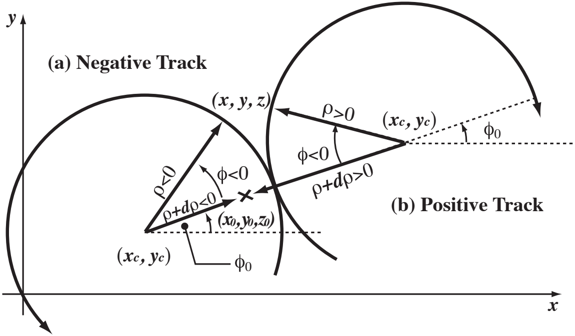


Figure 5 : Track model

Parametrization of helix

- Then the parametrization of helix is

$$\begin{cases} x = x_0 + d_\rho \cos\phi_0 + \frac{\alpha}{\kappa}\left[\cos\phi_0 - \cos(\phi_0+\phi)\right] \\ y = y_0 + d_\rho \sin\phi_0 + \frac{\alpha}{\kappa}\left[\sin\phi_0 - \sin(\phi_0+\phi)\right] \\ z = z_0 + d_z - \frac{\alpha}{\kappa}\tan\lambda\cdot\phi \end{cases} . \quad (2)$$

- The **state vector** of track can be defined by

$$\boldsymbol{a}_k = \left(\; d_\rho, \phi_0, \kappa, d_z, \tan\lambda \; \right)^T . \quad (3)$$

From Eq.(2), it means if the pivot and state vector are given, a helix is determined.

- For non-uniform magnetic field, a numerical track model (e.g. Runge-Kutta) or **segment-wise helical track** may take account into the non-uniformity.

Kalman Filter

For each site, Kalman filter algorithm has two steps:

- Prediction:

$$\boldsymbol{a}_k^{k-1} = \boldsymbol{f}_{k-1}(\boldsymbol{a}_{k-1}), \quad (4)$$

in which, $\boldsymbol{f}_k$ is propagation function[1]. And the corresponding **propagation matrix** is defined by

$$\boldsymbol{F}_{k-1} = \frac{\partial \boldsymbol{f}_{k-1}}{\partial \boldsymbol{a}_{k-1}}. \quad (5)$$

- Filtering:

$$\boldsymbol{a}_k = \boldsymbol{a}_k^{k-1} + \boldsymbol{K}_k \left(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{a}_k^{k-1}) \right), \quad (6)$$

where $\boldsymbol{K}_k$ is the gain matrix, $\boldsymbol{h}_k$ is the measurement function. It means the state vector is adjusted according to the predicted and real measurement.

[1] Don't confuse it with the function $\boldsymbol{f}$ in Eq.(1)

Algorithm for non-uniform magnetic field

To use the helical track model of KalTest in the non-uniform magnetic field, we have to:

- assume the magnetic field between two nearby layers is uniform;
- transform the frame to make the z axis point to the direction of magnetic field.

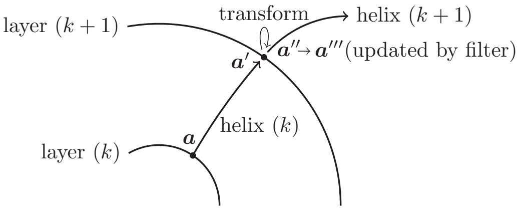


Figure 6 : The updated track propagation procedure.

Therefore we now have a **segment-wise helical track model**.

Transforming the frame

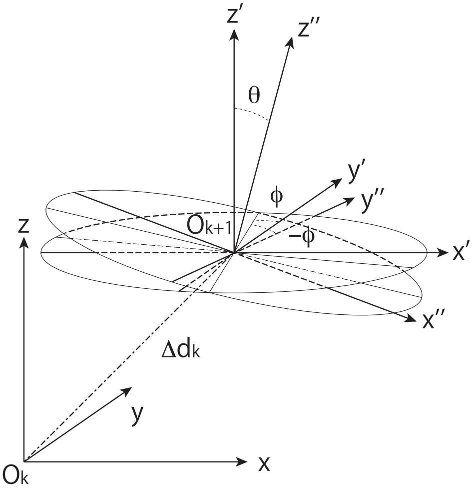


Figure 7 : Transformation

Rotation matrix

Define the transforming operation is to make the z axis rotate θ angle in the $z\prime Oz\prime\prime$ plane, then rotation matrix is equivalent to the product of three successive rotations:

$$\boldsymbol{\Delta R} = \boldsymbol{\Delta R}_{z''}(-\phi)\boldsymbol{\Delta R}_{y''}(\theta)\boldsymbol{\Delta R}_{z'}(\phi). \tag{7}$$

The rotation is passive, and the rotation matrices are

$$\boldsymbol{\Delta R}_{z'}(\phi) = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$\boldsymbol{\Delta R}_{y''}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}.$$

Modified propagator

- The propagation procedure can be represented by four equations:

$$\begin{cases} \boldsymbol{a}' &=& \boldsymbol{f}_k(\boldsymbol{a}_k) \\ \boldsymbol{p} &=& \boldsymbol{c}(\boldsymbol{a}') \\ \boldsymbol{p}' &=& \boldsymbol{t}(\boldsymbol{p}) \\ \boldsymbol{a}'' &=& \boldsymbol{c}^{-1}(\boldsymbol{p}') \end{cases} . \tag{8}$$

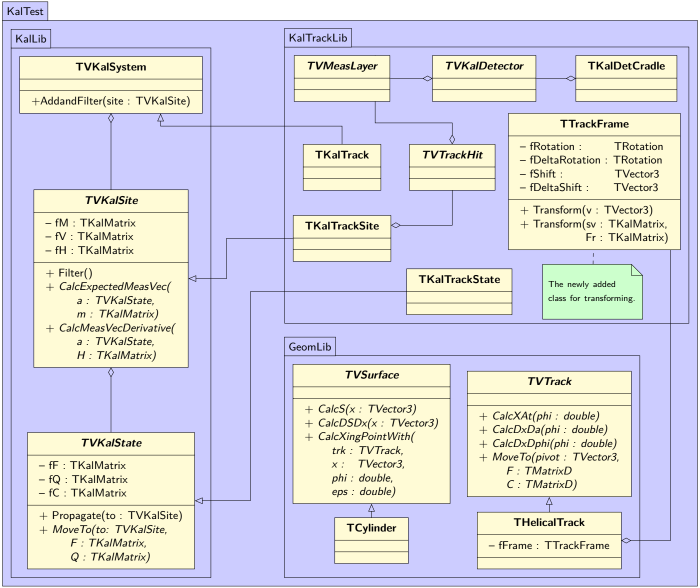
In Eq.(8), $\boldsymbol{f}_k$ in known in the original KalTest; $\boldsymbol{c}$ is the function to convert state vector to momentum, and $\boldsymbol{c}^{-1}$ is its inverse function; The transformation function $\boldsymbol{t}$ is actually the rotation matrix.

- Therefore, the propagation matrix should be modified accordingly[2]:

$$\boldsymbol{F}^m_{k-1} = \frac{\partial \boldsymbol{a}''}{\partial \boldsymbol{p}'} \frac{\partial \boldsymbol{p}'}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{a}'} \frac{\partial \boldsymbol{a}'}{\partial \boldsymbol{a}} = \boldsymbol{F}^r_{k-1} \boldsymbol{F}_{k-1}.$$

[2] Concrete matrix form is in the backup slides.

Implementation: class diagram of KalTest

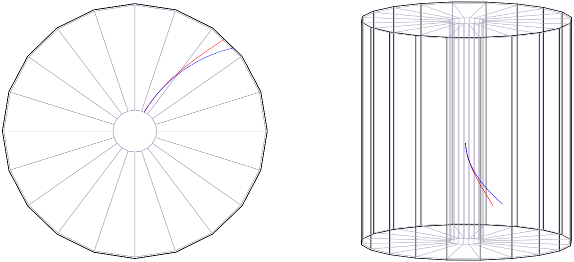


Simulation conditions

- Suppose the non-uniform magnetic field is

$$\begin{cases} B_x &= B_0 kxz \\ B_y &= B_0 kyz \\ B_z &= B_0(1 - kz^2) \end{cases},$$

  in which, $k = \frac{k_0}{z_m r_m}$, $B_0 = 3$ T, $z_m = r_m = 3000$ mm;
- Runge-Kutta track generator;
- Track parameters: dip angle $\lambda \in [0, 0.5]$, azimuth angle $\phi \in [0, 2\pi]$;
- Detector: 251 layers, distance between two nearby layers is 6 mm.



(a) xy view

(b) 3D view

Figure 8 : Event display. $2\ \mathrm{GeV}$ tracks generated in uniform magnetic field (blue curve), and non-uniform magnetic field (red curve, $k_0 = 5$).

Momentum resolution

- $k_0 = 1,\ p = 10\ \mathrm{GeV};$
- Tracks are reconstructed in **uniform** magnetic field.

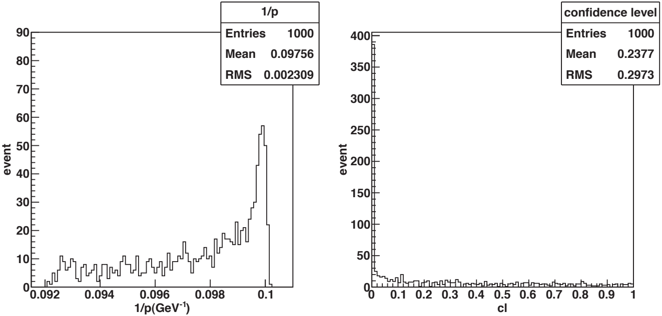


Figure 9 : Momentum and confidence level with uniform magnetic field.

Momentum resolution

- $k_0 = 1,\ p = 10$ GeV;
- Tracks are reconstructed in **non-uniform** magnetic field.

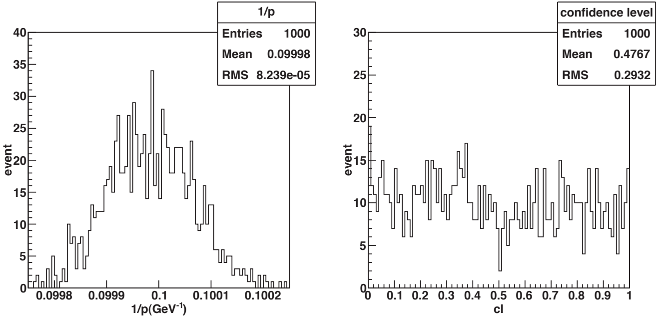


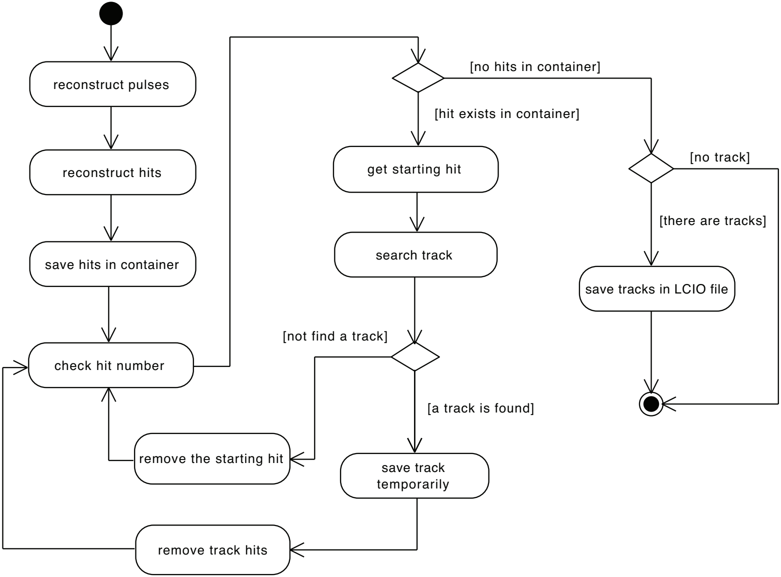
Figure 10 : Momentum and confidence level with non-uniform magnetic field.

Outline

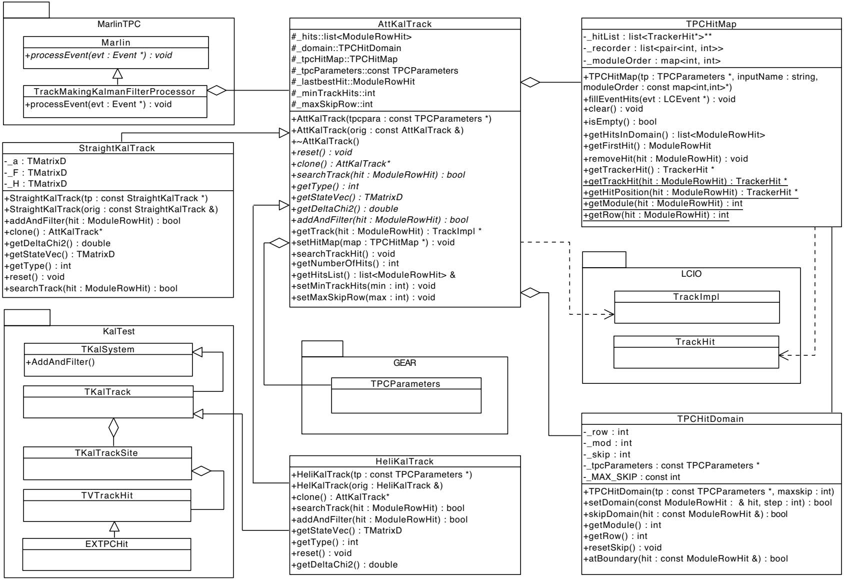
ILCSoft

- ILCSoft: a collection of software packages for ILC research.
- The following packages are related to MarlinTPC, which is for TPC reconstruction and simulation.
  - Marlin: a C++ software framework for ILC software.
    - ★ Every computing task is implemented as a processor that analyzes data in an event;
    - ★ In a analysis/reconstruction task, used processors with their parameters are defined in steering file;
    - ★ The author of processor deal with processors with defined callbacks, i.e. init(), processRunHeader(), processEvent(), and end().
  - LCIO: a framework that defines a data model (e.g. pulse, hit, track) for linear collider detector studies.
  - GEAR: geometry description toolkit for ILC reconstruction software. it uses XML files for the definition of the detector's geometry parameters.
  - LCCD: a conditions data framework for the ILC - based on LCIO.

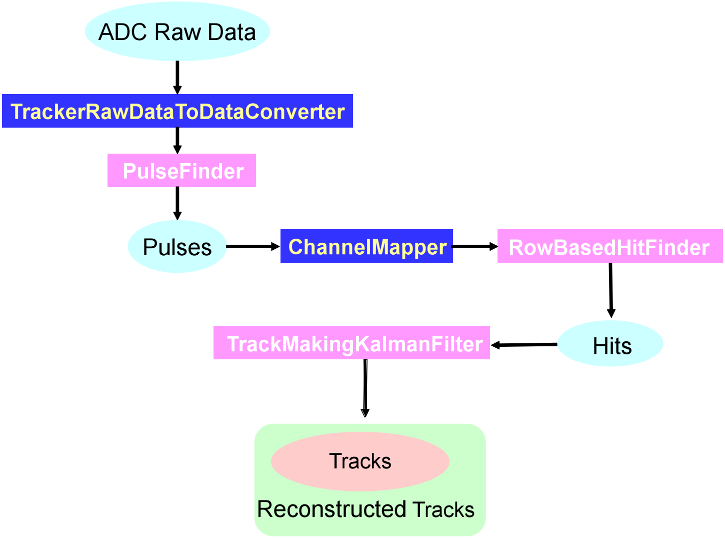
MarlinTPC tracking processor: flow chart



MarlinTPC tracking processor: class diagram



MarlinTPC reconstruction chain

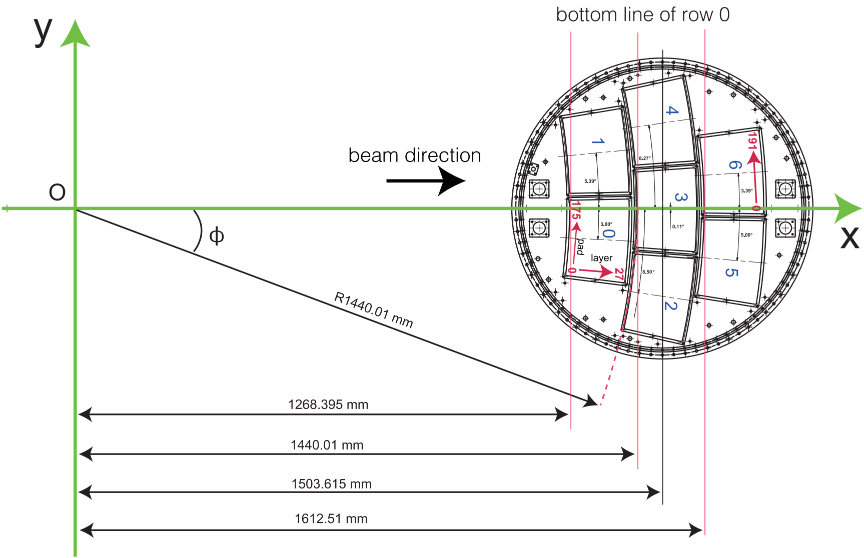


Steering file

```
<execute>
     <processor name="MyConditionsProcessor"/>
     <processor name="MyTrackerRawDataToDataConverterProcessor"/>
     <processor name="MyADCPulseConverterProcessor"/>
     <processor name="MyChannelMappingProcessor"/>
     <processor name="MyRowBasedHitFinderProcessor"/>
     <processor name="MyTrackMakingKalmanFilterProcessor"/>
     <processor name="MyLCIOOutputProcessor"/>
</execute>


<processor name="MyTrackMakingKalmanFilterProcessor" type="TrackMakingKalmanFilterProcessor">
     <!--Name of the Input TrackerHits collection-->
     <parameter name="InputTrackerHits" type="string" lcioInType="TrackerHit"> TPCHits </parameter>
     <!--Name of the output Tracks collection-->
     <parameter name="OutputTracks" type="string" lcioOutType="Track"> TPCTracks </parameter>
     <!--Maximum number of chi2 increment (default: 30.)-->
     <parameter name="MaxDeltaChi2" type="double" value="30"/>
     <!--Maximum number of subsequnetly missing hits (default: 5)-->
     <parameter name="MaxSkipRows" type="int" value="5"/>
     <!--Minimum number of hits on track (default: 15)-->
     <parameter name="MinTrackHits" type="int" value="60"/>
     <!--Track type (default: 1, 0=straight line 1=helix)-->
     <parameter name="TrackType" type="int" value="1"/>
</processor>
```

Detector geometry of LP



GEAR

The GEAR file for describing LP:

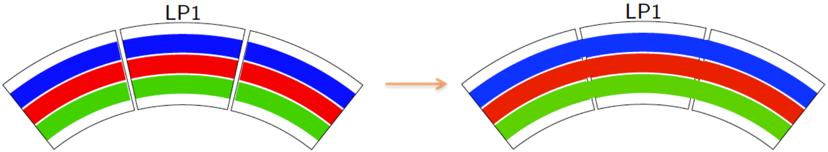
```
<modules moduleIDStartCount="0">
<default>
   <PadRowLayout2D type="VersatileDiskRowLayout"   rMin="1439.96">
     <!-- Set nPad, rowHeight, padPitch, offset, and repeat of row-->
     <!-- row 0 -->
     <row nPad="176" padPitch="1.18511" rowHeight="5.36"
            padWidth="1.085106" padHeight="5.26" />
     <!-- row 1 -->
     <row nPad="176" padPitch="1.18951" rowHeight="5.36"
            padWidth="1.08951" padHeight="5.26"  offset="0.594754"/>
      ...
     </PadRowLayout2D>
</default>

<!-- module No.0 -->
<module>
     <angle value="-0.12555026" />
     <offset x_r="-171.615" y_phi="0" />
</module>
```

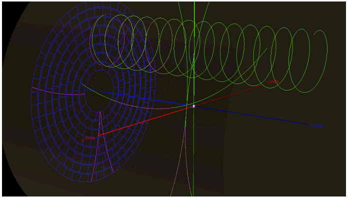
The current status of tracking processor

Updated for 7-module readout of LPTPC(Micromegas) in 2012:

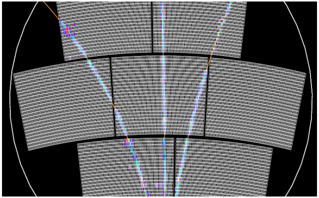
- GEAR



- MarlinTPC can reconstruct tracks in multi-module:



(a) Full endplate (MC)



(b) Micromegas (beam test)

MarlinTPC: installation

- OS: Ubuntu 12.04, 32-bit
- Prerequisites:
  1. ROOT[3], gsl, and CLHEP: install from source
  2. Java, MySQL, cmake, QT4[4]:
     $ sudo apt-get install openjdk-6-jdk mysql-server \
     cmake libqt4-opengl-dev
- ILCSoft[5]
  1. $ svn co https://svnsrv.desy.de/public/ilctools/ilcinstall/tags/v01-15
  2. Get the installation script, and configure software package paths according to your computer environment.
  3. $ ./ilcsoft-install -i v01-15.cfg
  4. Probably some libraries or binary files can not be found by ilcsoft-install (but they actually exist), then make a soft link.
  5. I had a linking problem when building MarlinTPC on Ubuntu, therefore I modified CMakeLists.txt a little bit to make it compiled. Good luck :-)

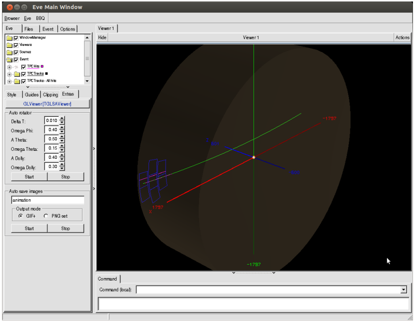
[3] enable miniut2

[4] optional

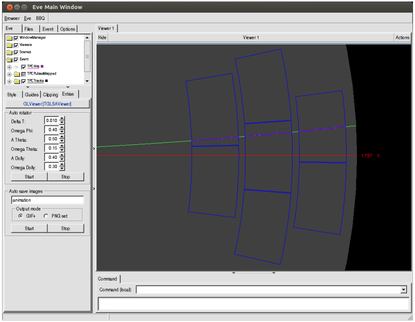
[5] A more detailed installation guide can be found at MarlinTPC Wiki

MarlinTPC, run!

- How to run:
  1. $ source init_ilcsoft.sh
  2. $ cd examples/reconstruction_multi_module
  3. $ generateHelix gear_LP_TPC_GEM_7module.xml
  4. $ Marlin reconstruction_tracks.xml
- Use BBQ to display to the reconstructed events:



(c) 3D



(d) xy

Outline

Summary

- Time Projection Chamber is under research for the future collider experiments.
- MarlinTPC is used for track reconstruction in LPTPC study. The tracking algorithm combines track following and Kalman filter.
- A segment-wise helical track model is built for tracking in non-uniform magnetic field, and source code is available.
- Plans:
  - Implement Runge-Kutta propagator in KalTest;
  - Study the magnetic field of LPTPC and LCTPC by simulation in MarlinTPC.

References

R. Frühwirth and M. Regler, *Data analysis techniques for high-energy physics*, 2nd ed., Cambridge University Press, 2000.

Keisuke Fujii, *Extended kalman filter*, http://www-jlc.kek.jp/subg/offl/kaltest.

Konrad Kleinknecht, *Detectors for particle radiation*, 2nd ed., Cambridge University Press, 1998.

Rainer Mankel, *Pattern recognition and event reconstruction in particle physics experiments*, Rept.Prog.Phys. **67** (2004), 553.

C. Peterson and T. Rögnvaldsson, *An introduction to artificial neural networks*, Proc. 1991 CERN Summer School of Computing, CERN Yellow Report 92-02 (1992), 113–170.