University of Victoria

Preparations for a Simulation Study of the Magnetic Field Requirements of the LC TPC

Christian Hansen
University of Victoria

Jason McGeachie
University of Victoria

Dean Karlen
University of Victoria

June 2006, ILC TPC Brainstorming, DESY

Objectives

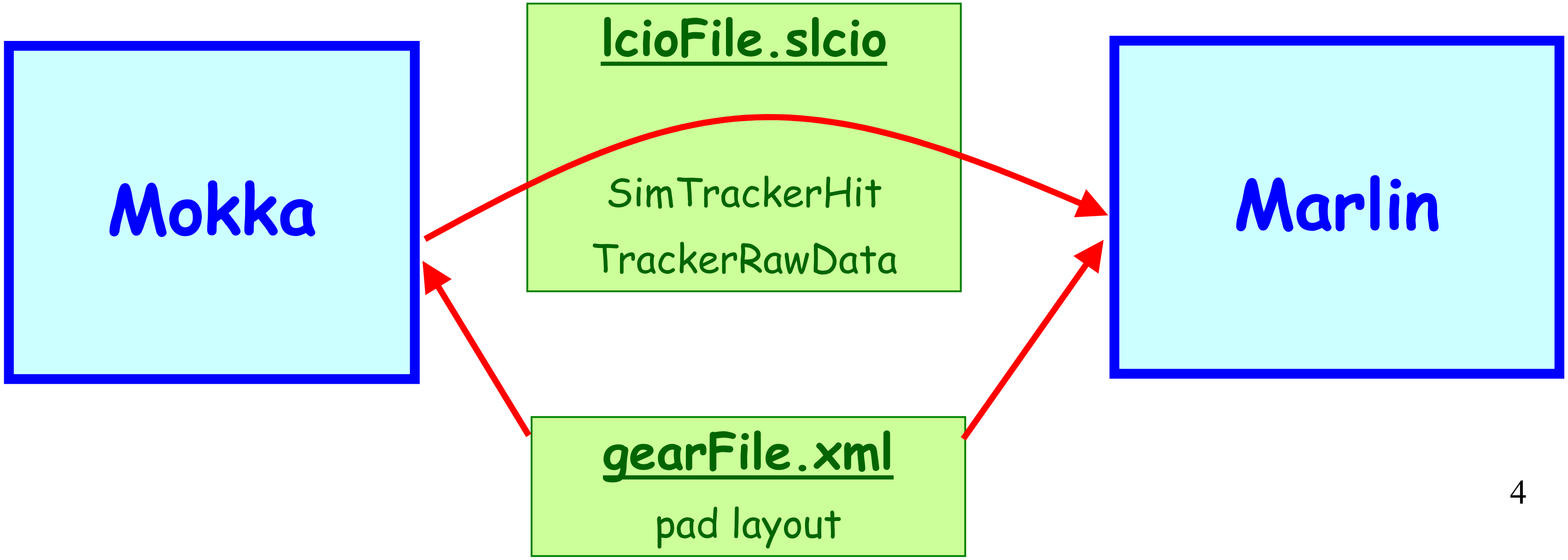
- ILC's crossing angle will cause magnetic field distortions
- Corrections for the inhomogeneous magnetic field will be needed for TPC's required momentum resolution ($d(1/p) \sim 10^{-4} GeV^{-1}$)
- Two ways to correct for magnetic field distortions;
  - a B-field map giving a first approximation
  - external data points (e.g. from the vertex detector)
- Back of the envelope estimations on the possible accuracy of these corrections have been done (see Snowmass 2005 talks by Ron Settles and Don Peterson)
- A simulation study is needed to determine the
  - influence of an inhomogeneous magnetic field on TPC's momentum resolution and the
  - possibility for magnetic field corrections to the reconstruction

Strategy

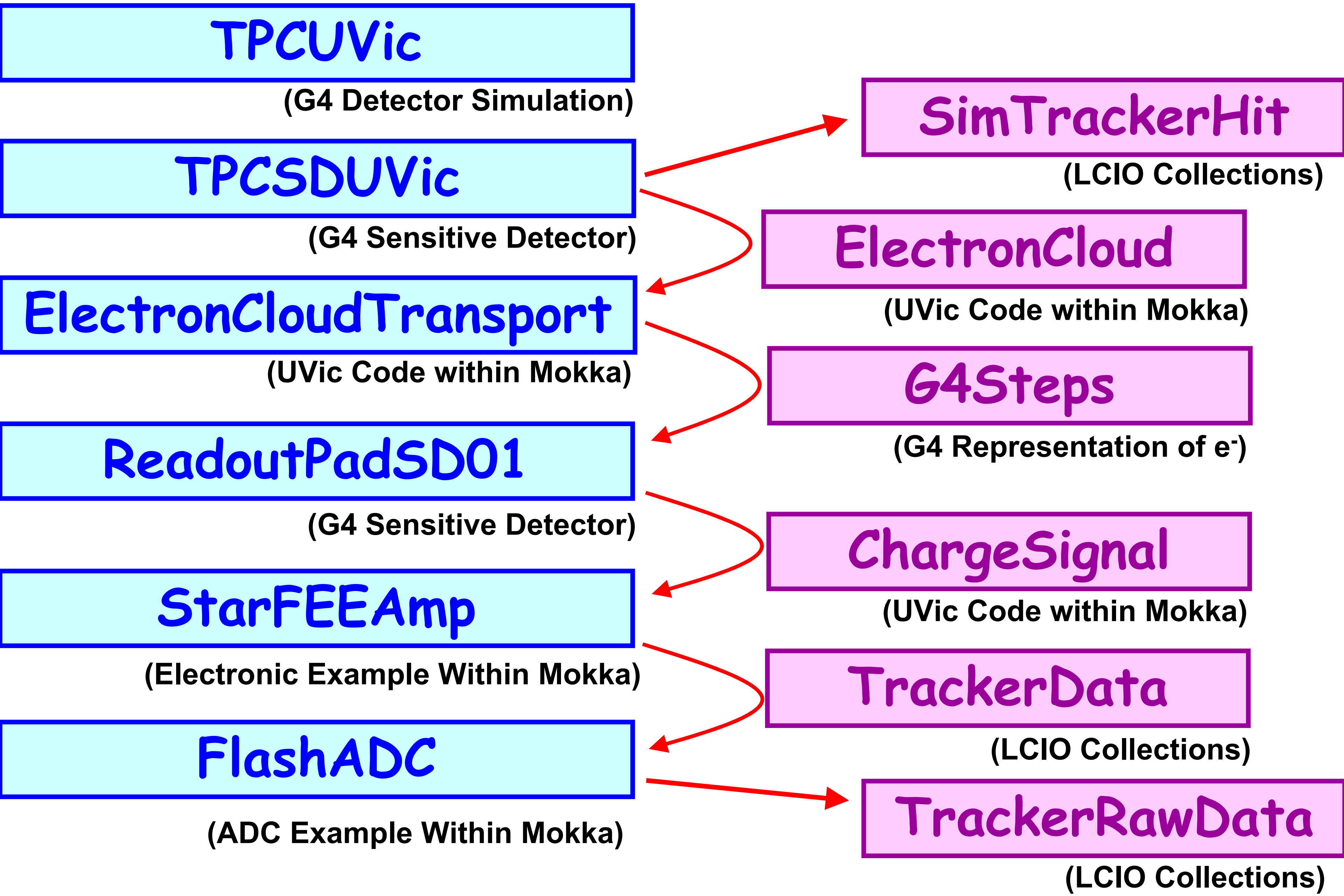
- **To study drift electrons distortion due to inhomogeneous magnetic field we edit Mokka's TPC to include**
  - **Drift electrons**
  - **GEM foils**
  - **Gas gaps**
  - **Endplate pads**
- **To study the effect on reconstruction add Marlin processes to**
  - **Read in and convert raw data**
  - **Make simple seed track fit** **(for now assume pattern recognition done)**
  - **Do the track fit** **(for now only single tracks)**

Mokka and Marlin

- We use a gear.xml file as input to both Mokka and Marlin since the pad layout is needed for both simulation and reconstruction
- The .slcio file from Mokka has both hits from the G4 energy deposition in the TPC (SimTrackerHit) and raw data from the pads (TrackerRawData)



TPCUVic Mokka Simulation

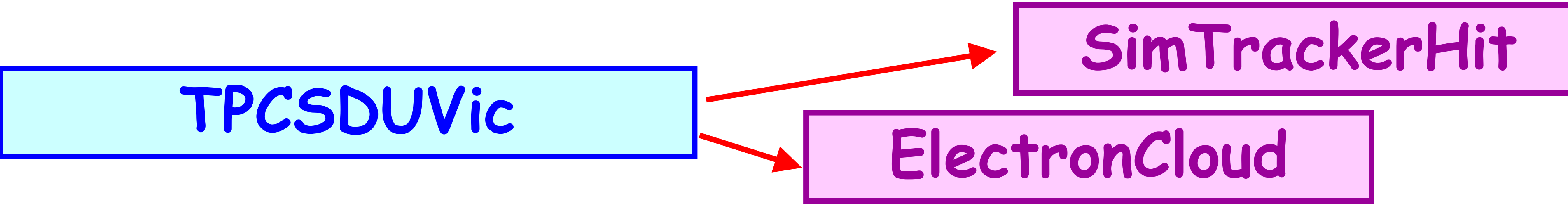


TPC Detector

TPCUVic

- **Geometry and Materials**
- **Edited the Tpc04 driver** (not much, just some lines)
- **User Limits**
  - Maximum step size for ionizing particles (no layers needed)
    - Sometimes new step due to step size and sometimes new step due to EM process. Checked that there is no double counting
  - Maximum amount of time a particle can travel before being stopped
  - Maximum amount of energy it can deposit
- **Sensitive Detector (TPCSDUVic)**
- **Copied and edited the MySQL database table "tpc06" to "tpcuvic" to our local MySQL database**

TPC "Sensitive Detector"



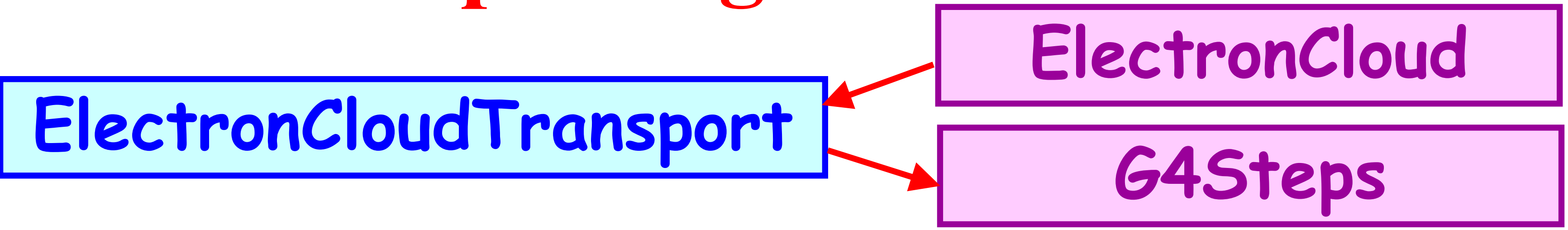
- **ProcessHits() gets the G4Steps**
  - **It stores the "hits" in a G4HitCollection, later saved as SimTrackerHits in a .slcio file**
  - **For each G4Step an ElectronCloud is produced**
    - **Number of electrons = The deposited energy / 26 eV**
  - **Transport the clouds with ElectronCloudTransport**
    - **For now; only primary particles are taken into account**
    - **Change later to deal with multiple tracks**
      - **Then reconstruction and track storage have to be changed as well "Trivial changes"**

A Cloud of Electrons (to save cpu)

ElectronCloud

- **Member variables are**
  - **number of electrons**
  - **x, y, and z coordinates**
  - **standard deviation in xy**
  - **standard deviation in z**
  - **time coordinate**

Transporting the drift electrons



- transportToEnd() gets a cloud
  - transportThroughGap(driftGap, aCloud)
  - transportThroughGEM(gem1, aCloud)
  - transportThroughGap(transferGap, someClouds)
  - transportThroughGEM(gem2, someClouds)
  - transportThroughGap(inductionGap, someClouds)
  - hitReadoutPads(someClouds)
    - Extracts electrons and gives G4Steps (represent e^-) to ReadoutPadSD01

Transporting the drift electrons (cont)

Drift Volume

implemented

- First approach; homogenous fields in drift volume
  - The clouds are transported with one step to the first GEM
  - Parameterized diffusion of the e^-

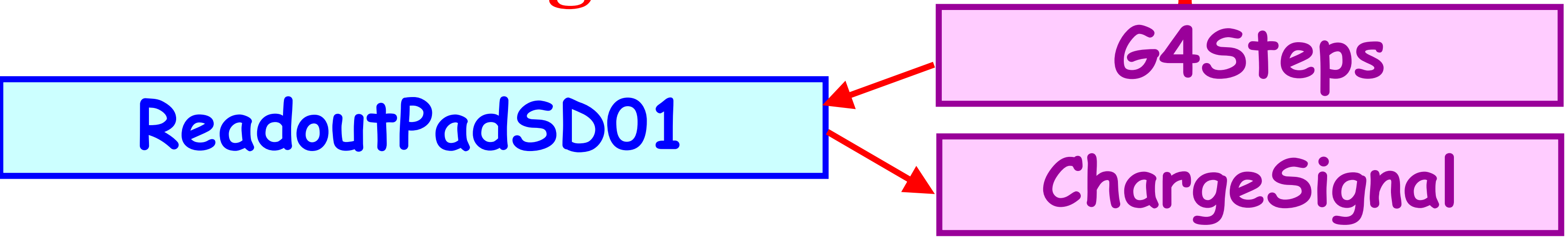
future work

- Later approach; inhomogeneous magnetic field
  - The clouds will be transported with several steps to the GEM
  - In each step a field map (FieldX00Map) will be used to get the Lorentz angle

End Plate

- Parameters for the GEMs (Gain, Collection and Extraction efficiency) is hard coded (in Gear file in future (?))
- Will this narrow area be effected by inhomogeneous magnetic field?
- "TPCEndPlate" is a hard coded name; don't change!

Assign electrons to pads



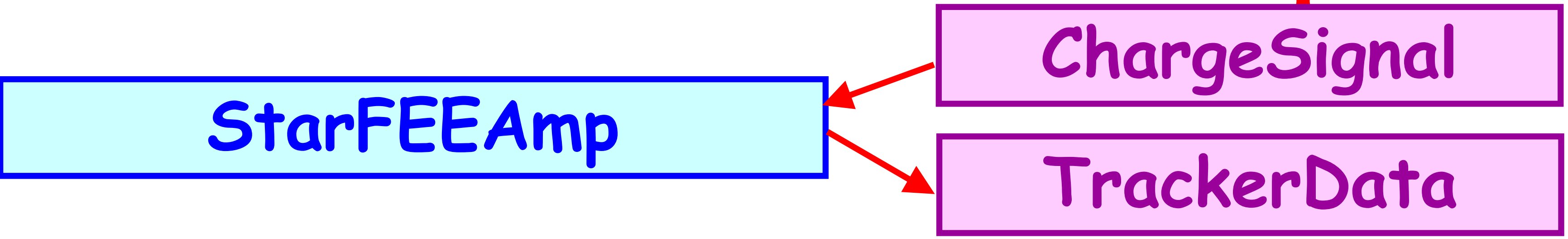
- **The function ProcessHit()**
  - **uses Gear functions to get the PadLayout and checks if electrons hit a pad**
  - **saves it to charge_signals which is a std::map of PadIndex (Gear) to ChargeSignals**
    - **i.e. only pads that are hit are saved in the map**

Electron Saved on a Pad

ChargeSignal

- **Member variables are**
  - number of electrons
  - mean time the electrons hit the pad (t_m)
  - standard deviation from the mean time
  - induction time (t_i)
    - the time the pad induce signal before the electron hit, normally the time the electron spent in the induction gap

Front End Electronics Amplifier



- **As an example of front-end electronics we have implemented Star's amplifier** (from Karlen's JTPC)
- **Shaping according to equation, A(t), that gives #e^- per time bin**
  - Width of time bins from "readoutFrequency" in the Gear file
- **Then convert it to mV per time bin**
  - VoltagePerTimeBin = gain * 10mV/fC * A(t) * CoulumbPerElectron
- **Reads in parameters from Gear file**
  - Gain
    - 3.3
  - Voltage threshold (indirect)
    - 10 mV ⇨ zero suppression, saves storage space
  - Rise and fall time
    - 30 ns and 2000 ns
  - mV per fC
    - 10

Star's FEE Amplifier

For the time shaping equation these time constants $\tau_1 = 729.878$ ns, $\tau_2 = 89.0123$ ns, $\tau_3 = 88.9504$ ns as well as the current rise time (τ_4) and fall time (τ_5) are used. The "star constants" are

$$s_0 = 58 \cdot 4$$

$$s_1 = k_1 \frac{\tau_1^3 \tau_3}{(\tau_1 - \tau_2)(\tau_1 - \tau_3)}$$

$$s_2 = \frac{\tau_1 \tau_2^2 \tau_3}{(\tau_2 - \tau_1)(\tau_2 - \tau_3)}$$

$$s_3 = \frac{\tau_1 \tau_3^3}{(\tau_3 - \tau_1)(\tau_3 - \tau_2)}$$

where $k_1 = 0.32851$.

Star's FEE Amplifier (cont)

The amplitude we then get from

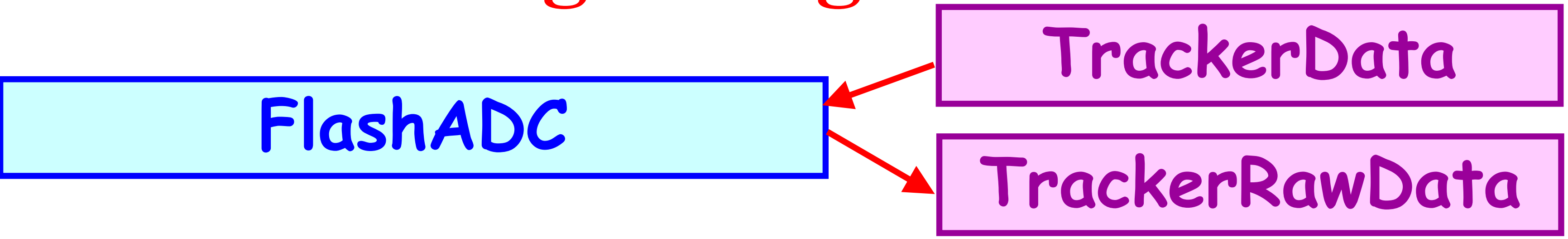
$$A = \begin{cases} 0, & t < 0 \\ \Sigma_{i=1}^{3} \frac{s_i}{\tau_4 s_0} \left(e^{-t/\tau_i} + \frac{t}{\tau_i} - 1 \right), & t \in [0, \tau_4] \\ \Sigma_{i=1}^{3} \frac{s_i}{s_0} \left(\frac{\tau_i + \tau_4 + \tau_5 - t}{\tau_i \tau_5} + \frac{1}{\tau_4} e^{-t/\tau_i} - \left(\frac{1}{\tau_4} + \frac{1}{\tau_5} \right) e^{(\tau_4 - t)/\tau_i} \right), & t \in [\tau_4, \tau_4 + \tau_5] \\ \Sigma_{i=1}^{3} \frac{s_i}{s_0} e^{-t/\tau_i} \left(\frac{1}{\tau_4} + \frac{1}{\tau_5} e^{(\tau_4 + \tau_5)/\tau_i} - \left(\frac{1}{\tau_4} + \frac{1}{\tau_5} \right) e^{\tau_4/\tau_i} \right), & t > \tau_4 + \tau_5 \end{cases}$$

Where 0 = meantime - induction time = $t_0 = t_m - t_i$

This is a semi-emperical fitted equation to front-end electronics used in the Star Experiment

Implemented into Mokka just as an example

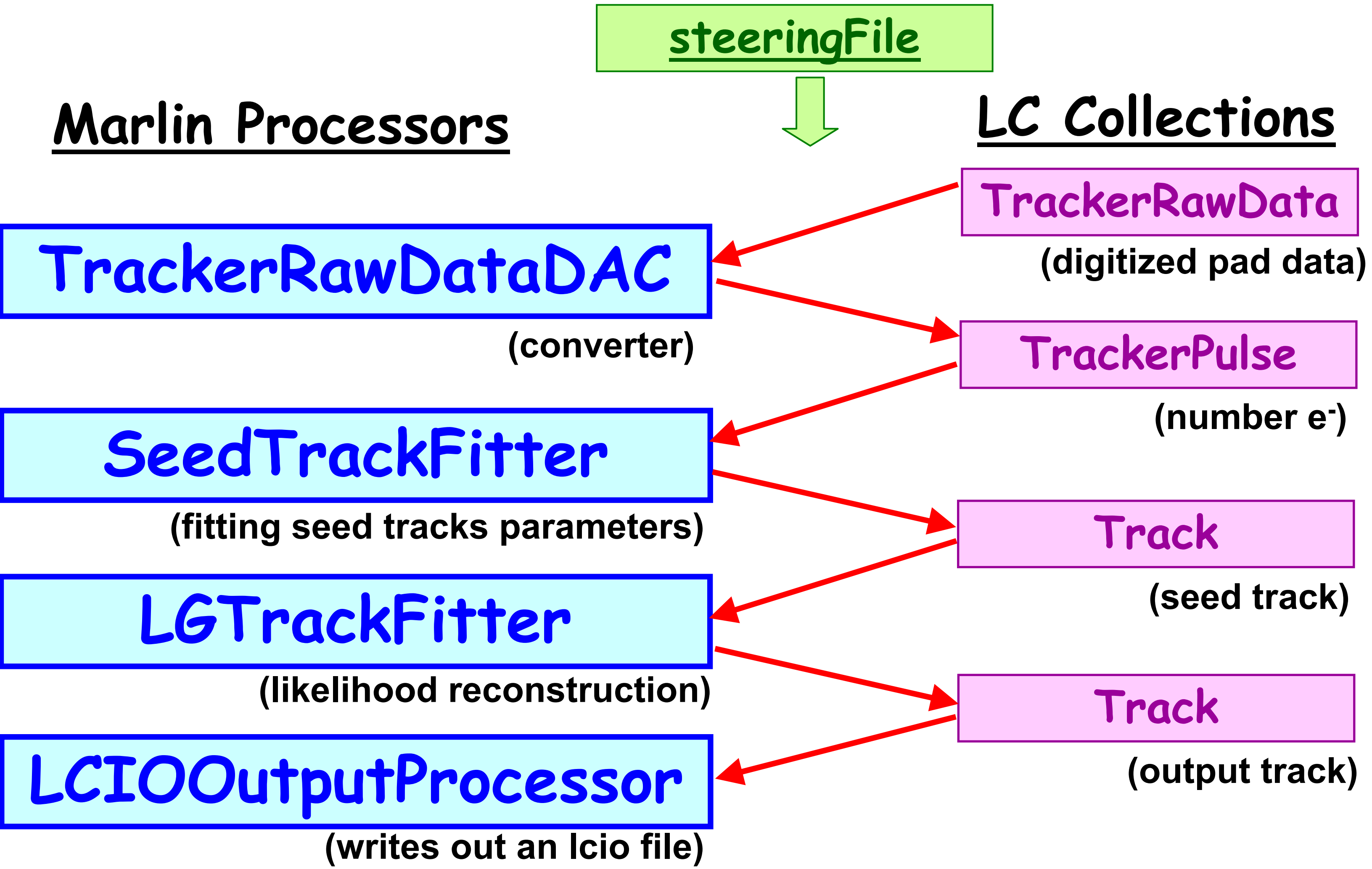
Analog to Digital Converter



- **Calculates the pedestal** (hard coded for now ... Gear?)
- **Converts time bins into channels**
  - Floating numbers → integers
  - Add pedestal → unsigned integers
    - if pedestal not big enough some values are truncated
- **Reads in parameters from Gear file**
  - gain 1
  - bits per channel 10
  - mV per channel 0.1

TPC Marlin Reconstruction

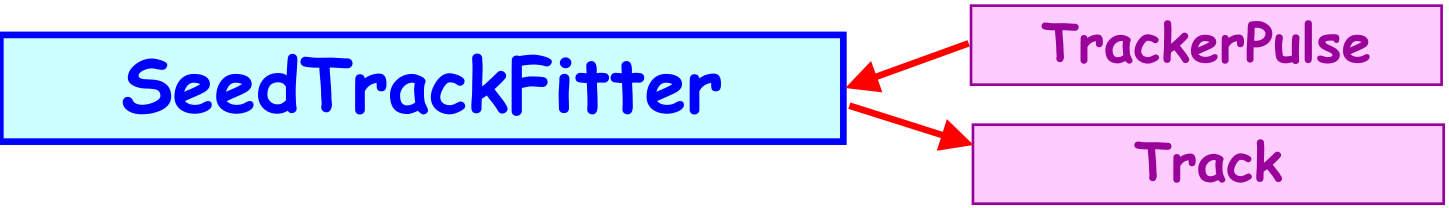
- **A steering file will control the LC Collections flow between the Marlin Processors**



Digital to "Analog" (#e^-) Converter

- The processEvent() function gets an LCEvent
  - Opens TrackerRawData collection from the LCEvent
  - Checks that all parameters are in the LCEvent
  - Converts all ADC values in the TrackerRawData to
    - TrackerData = Number electrons per time bin
    - TrackerPulseData = TrackerData and total number of electrons in the pad

Fitting the Seed Track



- **The SeedTrackFitter gives a rough estimate of a Track to LGTrackFitter**
- **Assumes pattern recognition already done, i.e. all hit pads from one track**
  - **will add established pattern recognition code later**

future work

Parameters of an LCIO Track

Track

(LCIO)

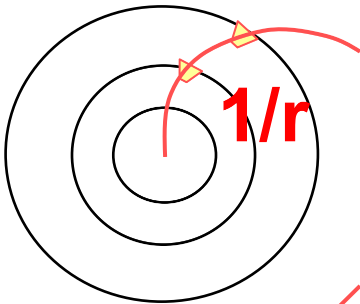
- To define a Track it's needed
  - Reference point
    - default; PCA ... to what? To origo or to collision point?
  - Ω
    - curvature, 1/r, sign = charge of particle
  - φ
    - phi angle for the track at the reference point
  - tan(λ)
    - lambda is the dip angle at the reference point
  - D_0
    - impact parameter in the r-φ plane (= 0 if PCA is **0**)
  - Z_0
    - signed impact parameter in the r-z plane (= 0 if PCA is **0**)
      - Z_0 = (z @ r = 0)
- Is there a documentation of LCIO's Track parameters?

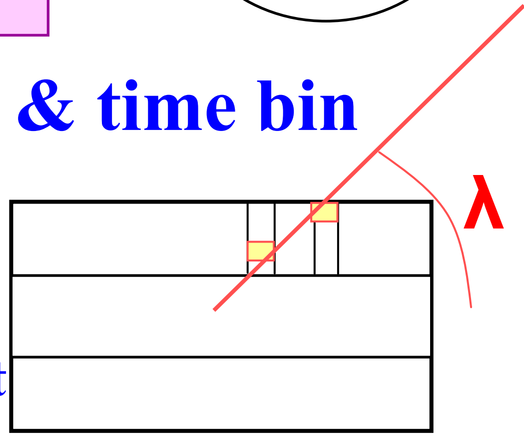
implementing

Fitting the Seed Track

SeedTrackFinder ← TrackerPulse

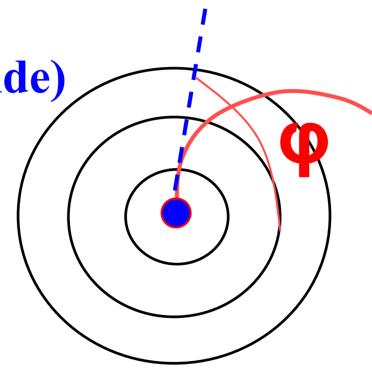
SeedTrackFinder → Track





- SeedTrackFinder gets #e⁻ in each pad & time bin

Ω
- Find two seed rows
  - First seed row: the first hit from outside
  - Second seed row: mid between origo and first
- Find seed pad with maximum # e⁻ for each seed row
- Radius from seed pads and origo (eq on next slide)
- Assume reference point = 0



λ
- Get lambda using seed pads (more later)

φ
- Get phi using seed pads (more later)

D_0, Z_0
- Assume both impact parameters zero

Curvature

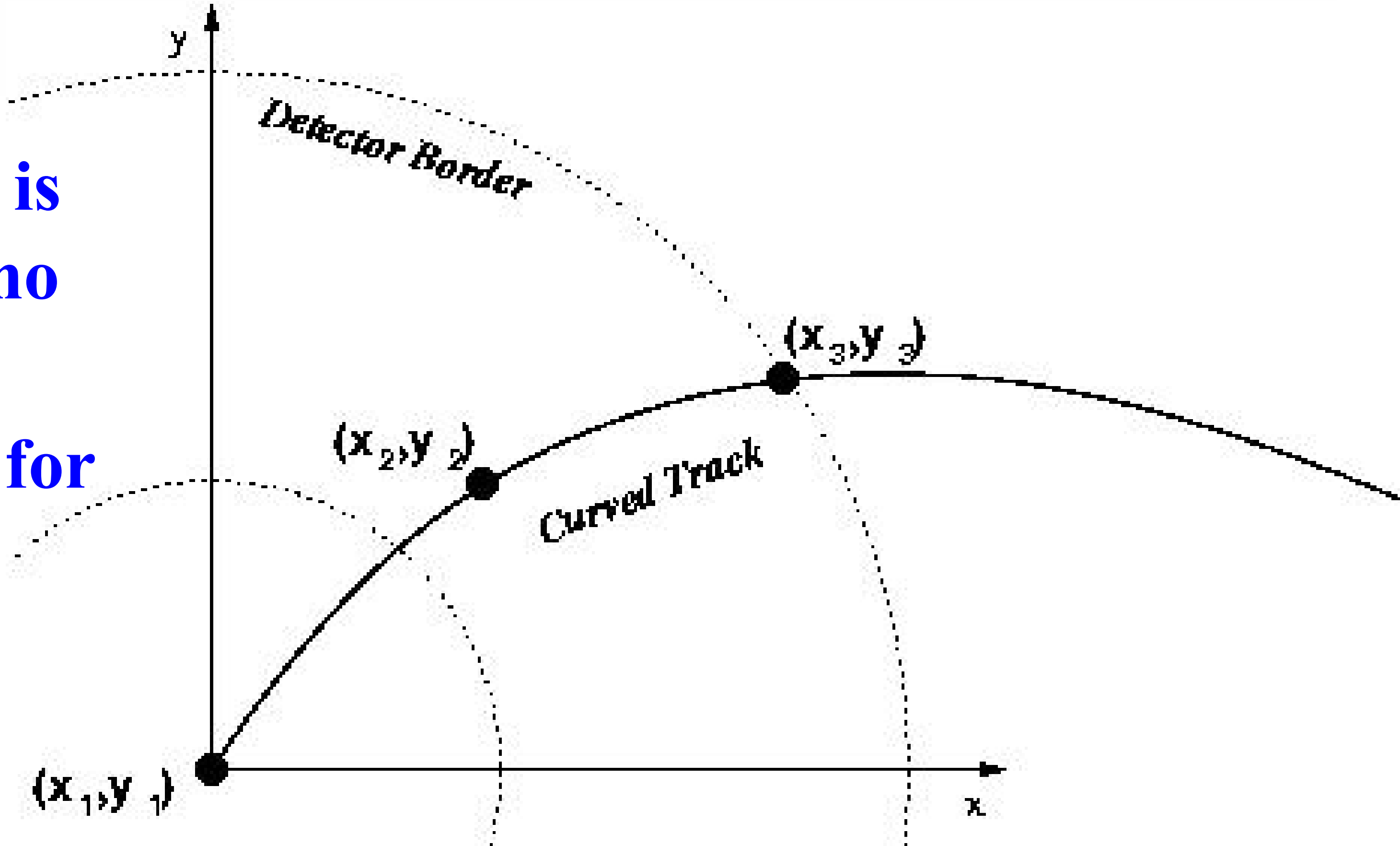
Three points on a circle $([x_1, y_1], [x_2, y_2], [x_3, y_3])$ give the radius:

$$r = \sqrt{\frac{(y_2(x_3^2 + y_3^2) - y_3(x_2^2 + y_2^2))^2 + (x_3(x_2^2 + y_2^2) - x_2(x_3^2 + y_3^2))^2}{4(x_2 y_3 - x_3 y_2)^2}}$$

if $x_1 = y_1 = 0$.

Perfect circle is assumed, i.e. no energy loss

Good approx for Seed Track



Lambda and Phi

Three points on a circle $([x_1, y_1], [x_2, y_2], [x_3, y_3])$, assuming $x_1 = y_1 = 0$, give the center of the circle:

$$x_c = -\frac{y_2(x_3^2 + y_3^2) - y_3(x_2^2 + y_2^2)}{2(x_2 y_3 - x_3 y_2)}$$

$$y_c = -\frac{x_3(x_2^2 + y_2^2) - x_2(x_3^2 + y_3^2)}{2(x_2 y_3 - x_3 y_2)}$$

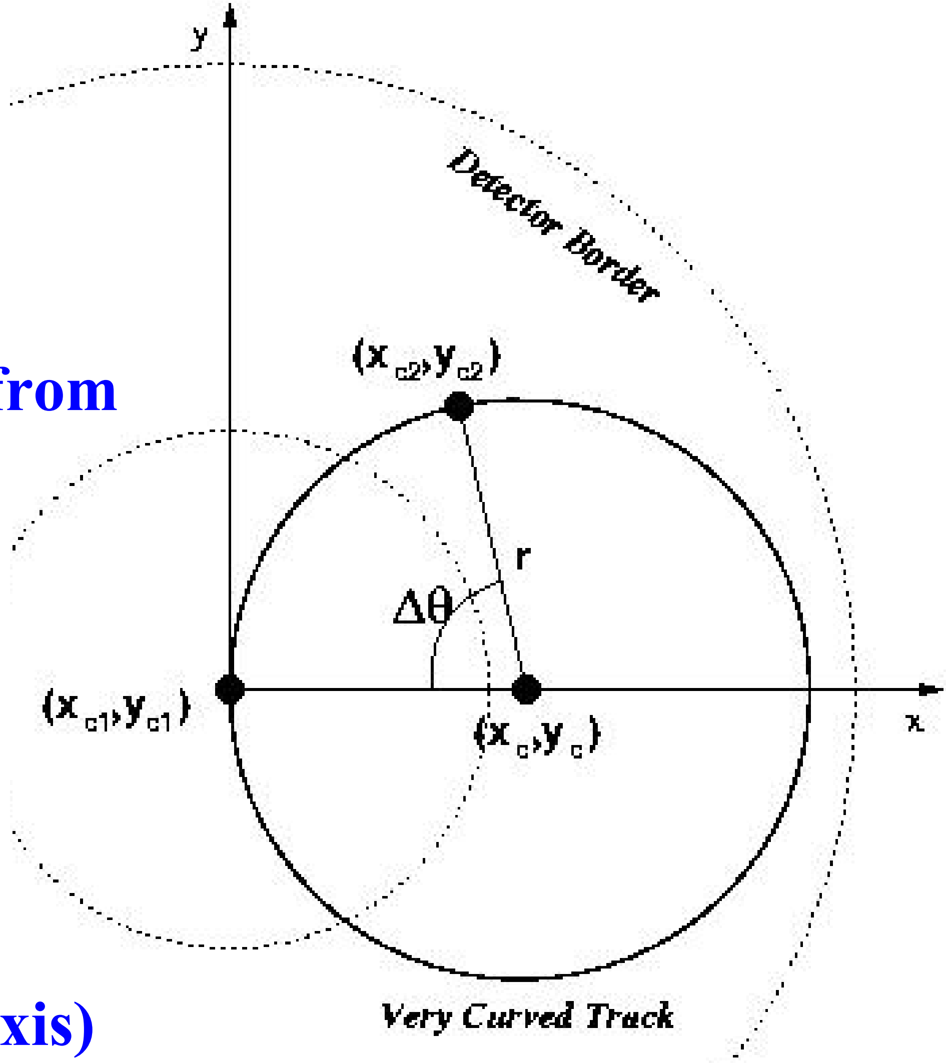
Changing origo to $[x_c, y_c]$ we get λ from

$$\tan\lambda = \frac{p_z}{p_t} = \frac{\Delta z/\Delta t}{\Delta\theta r/\Delta t} = \frac{\Delta z}{\Delta\theta r}$$

where

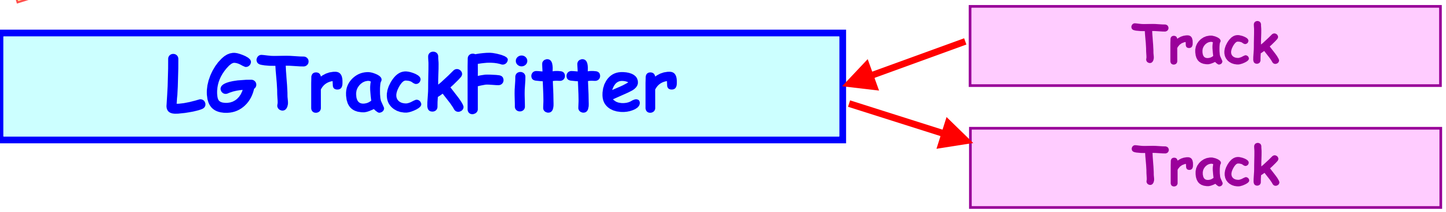
$$\Delta\theta = \left|\arctan\frac{y_{c1}}{x_{c1}} - \arctan\frac{y_{c2}}{x_{c2}}\right|$$

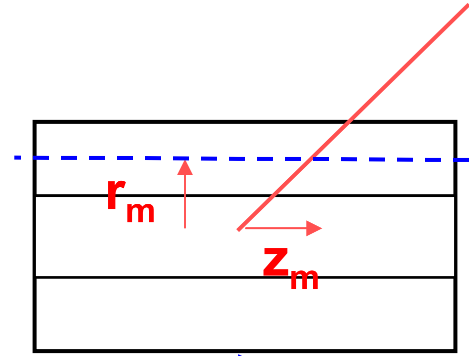
And φ from using Hep3Vector::deltaPhi(tangent, x-axis)



future work

Fitting the Track





- The processEvent() gets the seed track
  - Initilize Minuit(LogLikelihoodFunction)
  - Call Minuit(seed track)
- The LogLikelihoodFunction(track parameters, σ_m)
  - $\sigma^2 = \sigma_m^2 + (z - z_m)\, D^2$
    - D = diffusion constant (in Gear file ?)
    - σ_m = characteristic diffusion for that track ($\sigma_m = \sigma$ @ $r=r_m$)
    - σ_m is good to add to the minimization parameters since it
      - is uncorrelated to the others → does not increase errors in the other parameters
      - can be used to check gas properties from experimental data
      - indicates the fit quality (a σ_m way off → bad fit)
  - Impact parameters not zero due to
    - reconstruction imperfections
    - b-hadron → new vertex

Suggestions of use of GEAR

- We use Gear as input both to Mokka and Marlin
  (e.g. PadLayout is needed by both simulation and reconstruction)
- In Gear now; Drift velocity and readout frequency, so not only geometry... (?)
- Could Gear be edited to take care of dimensions?
- Organize the Gear file in sections
  - Gap Sections
  - Gem Sections
  - ... and so on ...

```
<!-- build the TPC inward from the readout electronics -->
<gap name="induction_gap" width="5mm" eField="0 0 1000V"/>
<gem name="gem2" gain="40" width="0.5mm" eField="0 0 1000V"/>
<gap name="transfer_gap" width="0.5mm" eField="0 0 1000V"/>
<gem name="gem1" gain="40" width="0.5mm" eField="0 0 1000V"/>
<gap name="drift_gap" width="*" eField="0 0 1000V"/>
```

Upgrade of CED

- **Can now draw curved tracks in 3D**
- **Can now draw keystones (for pads)**
- **Much faster**
  - **changed so that “DisplayList” is used to redraw objects and store geometry**
  - **and more ...**

UVic's "Wish List"

- **Standardized description of <u>whole</u> of the TPC** (Gear or MySql ...?)
  - **GEM's** (positions, widths, gains ...)
  - **Micromegas** (positions, widths, gains ...)
  - **GasVolume** (drift velocity, widths, gains ...)
  - **and more ...**
- **More structured Gear file**
  - **Sections**
  - **Dimensions**

```
<!-- build the TPC inward from the readout electronics -->
<gap name="induction_gap" width="5mm" eField="0 0 1000V"/>
<gem name="gem2" gain="40" width="0.5mm" eField="0 0 1000V"/>
<gap name="transfer_gap" width="0.5mm" eField="0 0 1000V"/>
<gem name="gem1" gain="40" width="0.5mm" eField="0 0 1000V"/>
<gap name="drift_gap" width="*" eField="0 0 1000V"/>
tpcParemeters.getGem("gem1").getGain();
theGems = tpcParameters.getGems();
```

- **Better documentation for the LCIO Track Parameters**
  - **Reference point is the PCA ...to what? Collision point or origo?**
  - **Signed impact parameters?**
  - **Diagram for angles**
    - **Phi and Lambda are respect of which axis?**
- **Can phpMyAdmin handle Mokka's MySQL database**

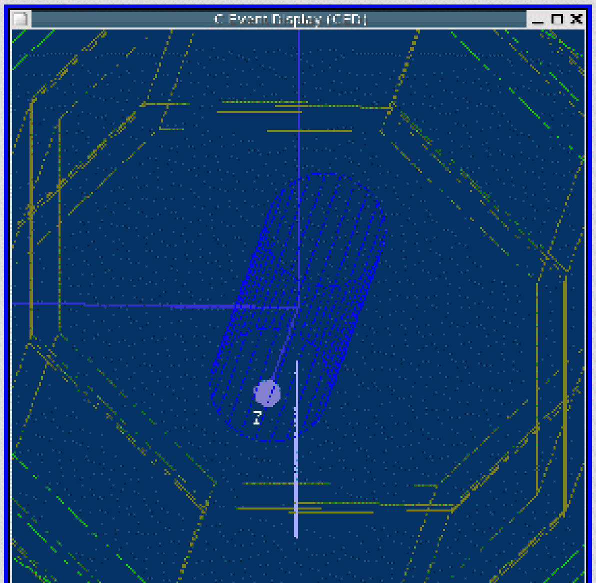
Home Pages

- **Installation Home Page:**

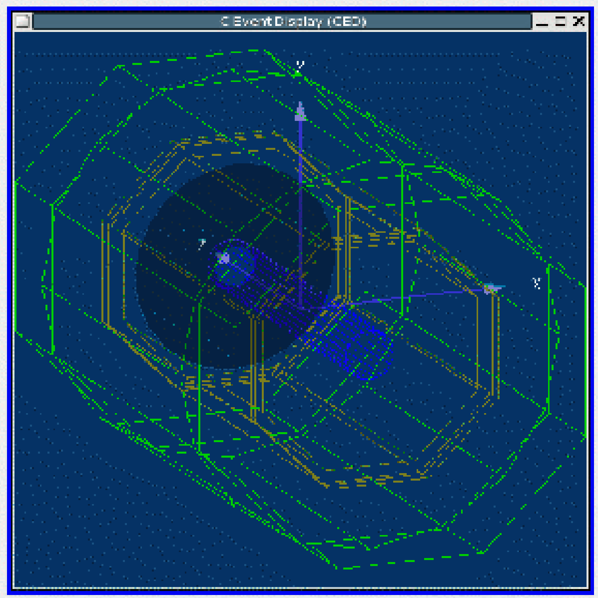
http://particle.phys.uvic.ca/~hansen/ILC/MOKKA/installation.html

- **Results and Download Home Page:**

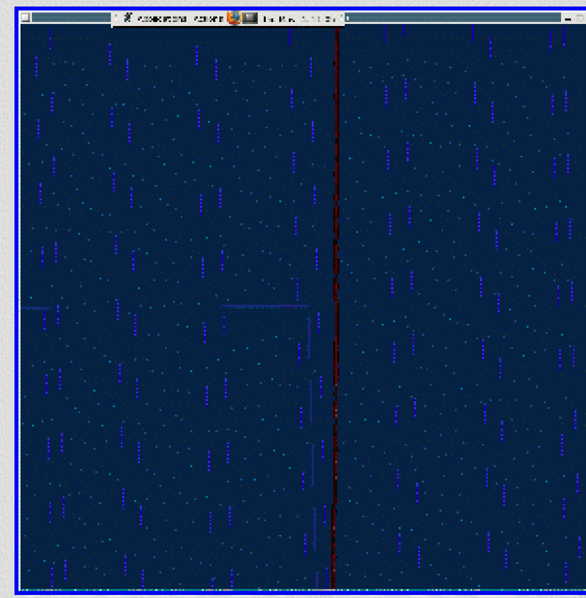
http://particle.phys.uvic.ca/~mcgeac00



Raw track data generated from Mokka *single colour pads - about 2000 were hit in this picture*



The TPC viewed in CED with Marlin. The semi-transparent circle is the pad layout.



A close-up of the representation of the TrackerPulse data reproduced in Marlin

Conclusions and Future Work

Installation

- Installation of Mokka, LCIO, Gear and Marlin has been done on SL3 and FC4

Objective

- Simulation is needed to understand the LC TPC's magnetic field requirements and possible corrections for the magnetic field distortions

Strategy

- Make a more detailed TPC simulation in Mokka
- Study what effect inhomogeneous magnetic field has on the momentum resolution of the LDC concept
- Study possibility to correct for field distortions with either a B-field map or external track points

Web

- More info here:

http://particle.phys.uvic.ca/~mcgeac00

http://particle.phys.uvic.ca/~hansen/ILC/MOKKA/installation.html

Extra Slides

Momentum Resolution from a Toy MC

- Plan to study momentum resolution from SeedTrackFinder with a Toy Monte Carlo
- Send out many charged tracks with random (reasonable) momentum
- Get the reconstructed momentum, p, from the reconstructed curvature
- Histogram the resolution, $(p - p_0) / p_0$
  - where p_0 is the true momentum

future work

Mokka installation

- **Installations done on two different system**
  Scientific Linux 3 (SL3) and Fedora Core 4 (FC4)
- **Installed needed software**
  CLHEP, Wired, OpenGL, Geant4, MySQL...
- **Installed LCIO**
  Had to add an include statement due to new gcc version in FC4
- **Installed Mokka**
  Had to create shared libraries and edit files for compile and link
- **Created Local Mokka MySQL Database**
  To be able to define our own Sub Detector
- **Installed Gear and linked it inside Mokka**
  Upgraded the implementation of Gear to define the TPC pads
- **Installed Marlin and CED** **(C Event Display)**

Details described step by step:

http://particle.phys.uvic.ca/~hansen/ILC/MOKKA/installation.html

Ionization in Mokka's TPC

- **Control stepsize for ionizating particles in the TPC gas**
  E.g. 1mm
- **G4 gives the energy loss for that step**
  100-400 eV/mm
- **One electron cloud per step**
- **Number electrons per cloud is given by the energy loss from G4 (26 eV/e^-)**

ElectronCloud.hh/cc

TPCSDUVic.hh/cc