Data Analysis and Reconstruction Techniques

DRD1 Detector School 2024; 27.11 – 06.12.2024,CERN

Theo Alexopoulos, National Technical University of Athens

November 29, 2024

Lecture Menu

- **Event Reconstruction**:
  Trigger & Data Acquisition; Track Reconstruction; Vertex Reconstruction.

- **Statistics & Numerical Methods**: Function Minimization; Statistical Models and Estimation.

- **Track Reconstruction**:
  Track Models: Equations of Motion; Track parametrization; Track propagation.
  Track Finding Techniques: Basic-techniques, Conformal mapping transformation, Artificial Retina, Hough/Radon transform, Legendre transform, Neural Networks, Kalman Filter.
  Track Fitting: Least-Square Fitting, Adaptive Fitting, Circle & Helix Fitting.

- **Vertex Reconstruction**:
  Vertex Finding: Primary Vertex Finding in 1D.

Event Reconstruction

The event reconstruction chain of a typical experiment spans from the trigger to the physics objects reconstruction. The main components of the event reconstruction cover the following:

1. **Trigger and Data Acquition**
2. **Track Reconstruction**
3. **Vertex Reconstruction**
4. **Physics Objects Reconstruction**

1. Trigger and Data Acquisition

- Trigger and Data Acquisition: A selection mechanism is required that tags the physically interesting events and activates the next steps of the data recording (data acquisition - DAQ). Trigger systems of the experiments have been deployed for many decades, from bubble chambers to electronic tracking detectors and calorimeters.

- Trigger systems are vital for fixed target as well as for colliding beam experiments due to limitations in data rates, storage capacity and computing resources.

- Trigger systems are implemented in several levels/stages with increasing computational complexity and decision latency to minimize the dead time of the trigger.

- A typical example of a Trigger & DAQ (TDAQ) system based on the ATLAS experiment follows.

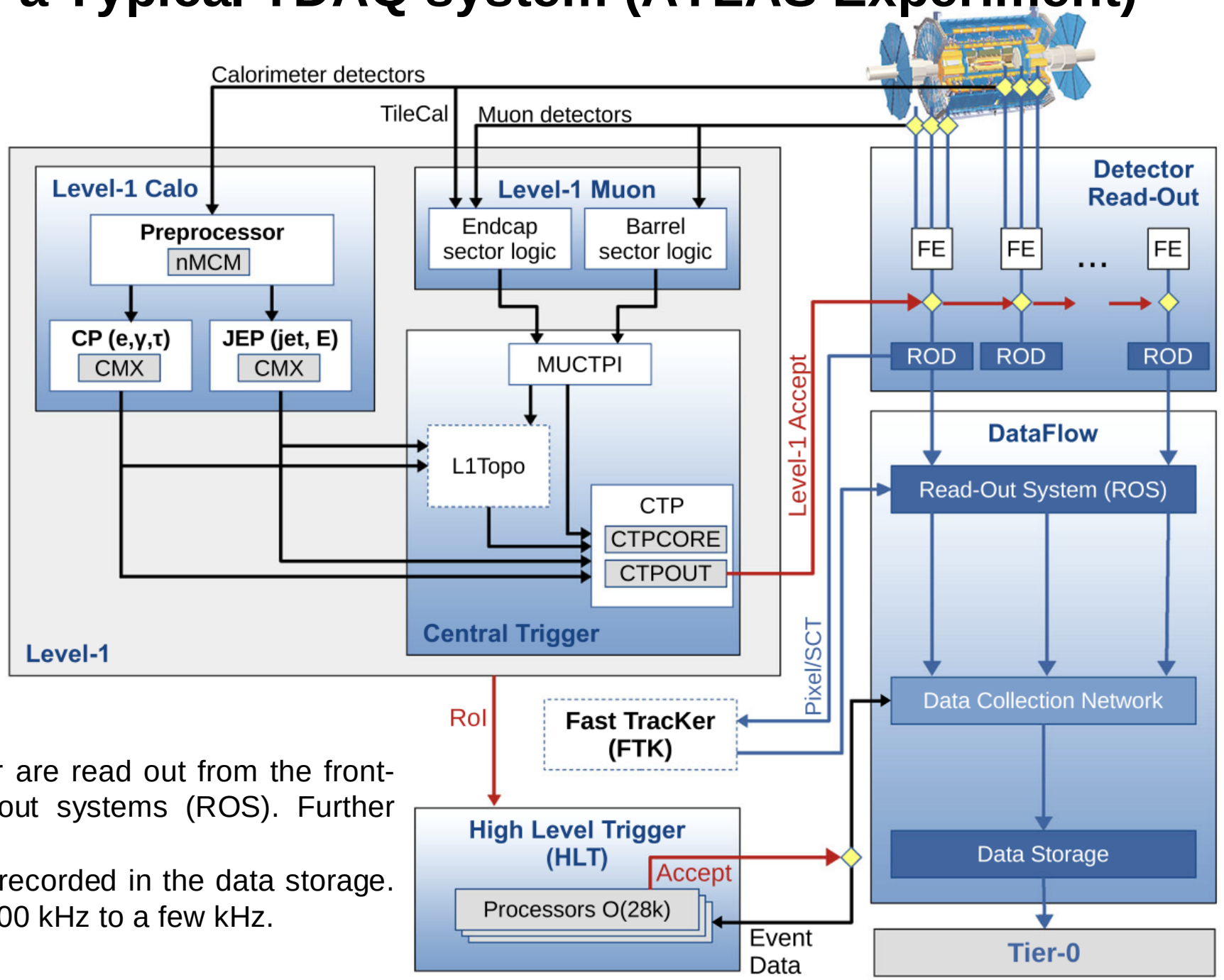
Continue...

from previous page…

Example of a Typical TDAQ system (ATLAS Experiment)

ATLAS trigger and DAQ scheme:

- Level-1 trigger makes an initial selection based on the huge number of electronics modules. There are two main L1 trigger systems: L1 calorimeter trigger (Level1-Calo) and the L1 muon trigger (Level-1 Muon)
- Selected objects are sent to L1 central trigger processor (Central Trigger)
- The L1 trigger provides “region-of-interest (RoI)” information including position (η and φ) and p_T range of candidate objects for the input of HLT.
- The L1 trigger makes a trigger decision within about 2.5 μs and reduces the event rate from 40 MHz to 100 kHz.
- Only events selected by the L1 trigger are read out from the front-end electronics systems to the readout systems (ROS). Further trigger selections are done by the HLT.
- Only events accepted by the HLT are recorded in the data storage. The HLT reduces the event rate from 100 kHz to a few kHz.



2. Track Reconstruction

- Track reconstruction is a main task in the analysis of the event data.

- It provides estimates of the track parameters, including the position, the direction, and the momentum of charged particles at one or several specific points or surfaces.

- The tracks of stable or sufficiently long-lived charged particles are visible in the tracking detectors. The short-lived particles (i.e. *B* hadrons or *J/ψ* mesons) are reconstructed from their decay products.

The reconstruction of charged particles can be divided into four steps (1. **Hit generation, 2. Local track reconstruction, 3. Global track reconstruction, 4. Assesment of track quality**):

1. **Hit generation –** the electronic signals from the various tracking system detectors are converted to spatial coordinates either 2D or 3D, using the detector-specific calibration constants. The coordinates are called *measurements or observations or hits*.

2. **Local track reconstruction –** tracks are reconstructed in each tracking system**.** This process can be analysed into three steps: *a) Track segment reconstruction, b) Track finding, and c) track fitting*.

Continue...

from previous page…

Track Reconstruction

a) *Track segment reconstruction*: this step is relevant only for tracking systems that are composed of several independent devices capable of giving at least the position and the direction of the particle and possibly the momentum. An example is the muon MDT/ATLAS chambers or the micromegas modules of the ATLAS News Small Wheel detector system which consist of eight layers where each layer can provide enough hits to estimate the parameters of a straight line track segment (tracklet).

b) *Track finding*: in this step, hits or track segments are clustered to track candidates. Track finding can be done iteratively, i.e., in the very-high multiplicity events recorded by the LHC experiments. In this case, "easy/obvious" tracks with high momentum and small material effects are found first, while more "difficult" tracks are extracted in the subsquent passes.

c) *Track fitting*: for each track candidate, a track model is fitted to the hits in order to get the best estimtes of the track parameters. The fit gives an indication of the quality of the fit, by examing the chi-square χ^2 parameter. An abnormal large value of the chi-square indicates either a random combination of hits or the presence of outliers in the track candidate. Outliers can either be remove from the track or down-weighted.

3. Global track reconstruction – after the local track reconstruction, the tracks found in the individual tracking systems must be combined to global track candidates. The track candidates accepted by the track fit in the main tracking system are extrapolated to the other tracking systems and checked for compatibility with the tracks reconstructed there. The successful combination of local tracks to a global candidate is followed by a track fit of the global candidate.

4. Assessment of track quality – Not every track candidate generated by the track finding is a valid track. Testing the track hypothesis and assessing the track quality after the track fit is therefore mandatory.

3. Vertex Reconstruction

- A point where particles are produced in a collision, or a decay is called a vertex.
- *Primary vertex:* The point of collision of two beam particles in a collider or of a beam particle with a target particle in a fixed-target experiment.
- In high-luminosity colliders, such as the LHC, many collisions occur in a single bunch crossing; consequently, there are many primary vertices. It is, however, statistically almost certain that at most one of the collisions generates a pattern recognized by the trigger as being of potential physical interest. The vertex of this collision is called the *signal vertex*.
- *Secondary vertex:* many of the particles produced at a primary vertex, including the *signal vertex*, are unstable and decay at a secondary vertex.
- The aim of of vertex reconstruction is to find sets of particles that have been produced at the same vertex, to estimate the vertex position, test whether the assignment of the particles to the vertex is correct and improve the estimates of the track parameters by imposing the vertex constraint. Alternatively, the vertex can be estimated from the hits in a global method without benefit of tracking.

4. Physics Objects Reconstruction

- Both the trigger and the physics analysis require not just tracks, but objects that represent physical entities, i.e. electrons, photons, muons, τ leptons, jets, missing energy, etc.

- Object identification can be obtained by two complementary approaches: 1. dedicated detectors for particle identification (PID), and **2.** combining information from different sub-detectors.

1. Dedicated detectors for particle identification (PID) **–** Charged particles can be identified by dedicated detectors in various ways, like:

- **Measurement of the velocity:** Given the momentum as determined by the tracking system, the mass can be estimated. Velocity can be measured directly by time-of-flight detectors, or indirectly by measuring the emission angle of Cherenkov radiation in Cherenkov detectors.
- **Energy loss by ionization:** In a large range of velocity, the expected energy loss by ionization is proportional to $(m/p)^{2,}$ where m is the unknown mass and p is the momentum of the particle. In practice, the most probable energy loss is estimated from several measurements. In a silicon tracker, energy loss is measured in each sensor; in a drift or time projection chamber (TPC), the energy loss is measured for each wire hit or for each cluster in the endplates, respectively.
- **Transition radiation:** Transition radiation (TR) is electromagnetic radiation in the X-ray band. It is emitted when an ultra relativistic particle crosses the boundary between two media with different dielectric constants. The radiator is combined with a gaseous detectors that measures the TR signal and the position of the particle.

Continue...

from previous page…

2. *Particle and Object ID by tracking and Calorimetry* **–** PID in dedicated detectors is complemented by combining information from the tracking systems and the calorimeters.

- **Electrons:** Electrons and positrons are identified as such by the fact that they have a reconstructed track and a cluster in the electromagnetic calorimeter that matches the track in energy and position.
- **Photons:** Clusters in the electromagnetic calorimeter that are not matched to a track or a cluster in the hadronic calorimeters are candidates for photons.
- **Muons:** Global tracks with hits in both the central tracking system and the muon tracking system.
- **Jets:** Jets are narrow bundles of charged and neutral particles produced by the hadronization of a quark or a gluon. Jet reconstruction algorithms are based on clustering the charged tracks but should also provide a good correspondence between the energy deposits in the calorimeters and the reconstructed tracks. This is the aim of the particle flow method, which originated in the ALEPH experiment at the LEP collider and is now employed by LHC experiments as well.
- **Tau leptons:** Tau leptons must be reconstructed from their decay products. In 2/3 of the cases, τ leptons decay into hadrons, typically into one or three charged mesons (mainly π), often accompanied by neutral π*'s* decaying into photons and an invisible neutrino.
- **Missing energy:** Missing transverse energy is a signature for invisible particles such as neutrinos, dark matter, etc.

Track Reconstruction

The track reconstruction includes three main items:

1. **Track Models**
2. **Track Parametrization**
3. **Track Finding Techniques**

1. Track Models

- Let's examine how the equations of motion for charged particles in a homogeneous or inhomogeneous magnetic field are solved. Various types of parametrizations are presented, and formulas for track propagation are given.

Consider a charged particle with mass m and charge $Q = qe$, where e is the elementary charge and q is an integer, usually $q = \pm 1$. Its trajectory or position $\boldsymbol{r}(t) = (x(t), y(t), z(t))^{\mathrm{T}}$ in a magnetic field $\boldsymbol{B}(\boldsymbol{r})$, as a function of time, is determined by the equations of motion given by the Lorentz force $\boldsymbol{F} \propto q\boldsymbol{v} \times \boldsymbol{B}$, where $\boldsymbol{v} = \mathrm{d}\boldsymbol{r}/\mathrm{d}t$ is the velocity of the particle. In vacuum, Newton's second law gives:

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = kq\boldsymbol{v}(t) \times \boldsymbol{B}(t), \quad \boldsymbol{p} = m\gamma\boldsymbol{v},\ \gamma = \left(1 - \boldsymbol{v}^2/c^2\right)^{-1/2}$$

where the parameter $k = 0.3$ (GeV/c) T^{-1} m^{-1}

The trajectory is uniquely defined by the initial conditions, i.e., the six degrees of freedom specified for instance by the initial position and the initial momentum. If these are tied to a reference surface, five degrees of freedom are necessary and sufficient. Geometrical quantities other than position and velocity can also be used to specify the initial conditions. The collection $\boldsymbol{q} = (q_1, \ldots, q_m)$ of these quantities is called the *initial track parameter vector* or the *initial state vector*.

continue...

from previous page…

Track Models

Equation of motion can be written in terms of the path length *s*(*t*) along the trajectory instead of *t*:

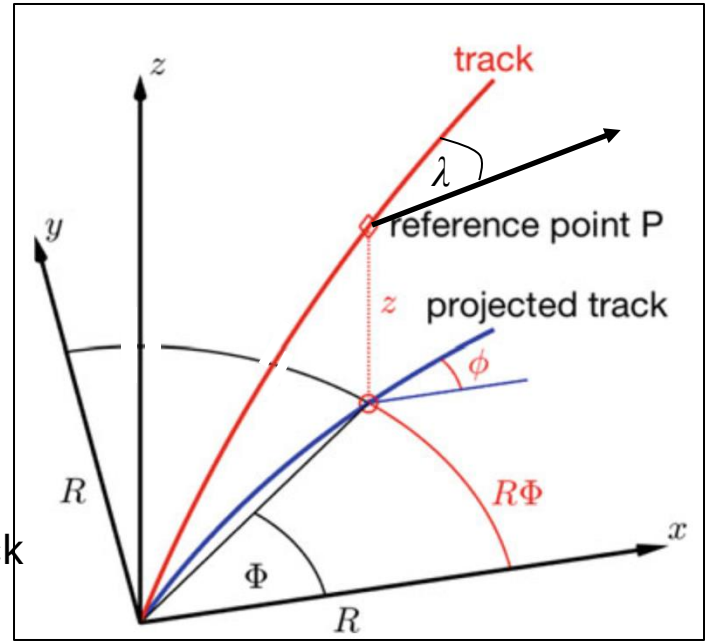
$$\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}s^2} = k\frac{q}{|\boldsymbol{p}|}\underbrace{\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s}}_{\dot{\boldsymbol{r}}(s)} \times \boldsymbol{B}(\boldsymbol{r}(s)) = \boldsymbol{F}(s, \boldsymbol{r}(s), \dot{\boldsymbol{r}}(s))$$

In a homogeneous magnetic field, the solution is a helix; it reduces to a straight line in the limit of $\boldsymbol{B} = \boldsymbol{0}$. In the general case of an inhomogeneous magnetic field, one must resort to numerical methods such as Runge–Kutta integration of the equations of motion.

The above Equation of motion can be expressed in terms of other independent variables. For example, if the equations of motion are integrated in a cylindrical detector geometry, the radius *R* is a natural integration variable. In a planar detector geometry, the position coordinate *z* could be the variable of choice.

continue...

2. Track Parametrization



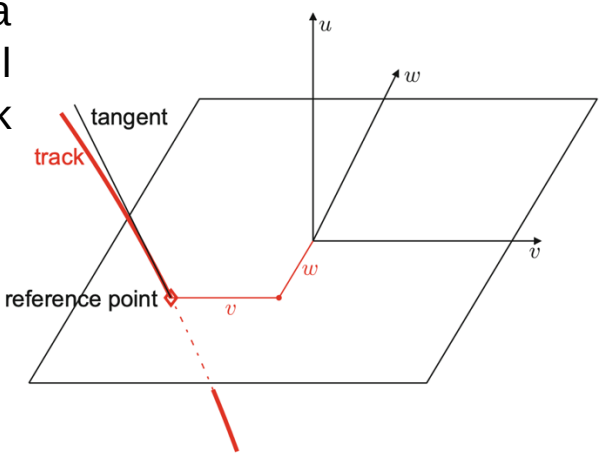
- *Different detector geometries often lead to different choices of the track parameters.* In a barrel-type detector system typical for the central part of collider experiments, a natural reference surface of the track parameters is a cylinder with radius *R*, centred around the global z-axis, which usually coincides with the beam line. The track parameters are, in this case, defined at the point of intersection P between the track and the reference cylinder. In such a system, one possible choice of track parametrization is the following:

$$q_1 = q/p_{\mathrm{T}},\ q_2 = \phi,\ q_3 = \tan\lambda,\ q_4 = R\Phi,\ q_5 = z$$

where $p_T = p \cos\lambda$ is the transverse momentum, φ is the azimuth angle of the tangent of the track at P, λ is the dip angle (complement of the polar angle) of the tangent at P, and $R\Phi$ and z are the cylindrical coordinates of P in the global coordinate system.

- In a detector system based on planar detector elements, the natural reference surface is a plane. Such a surface is uniquely determined by a normal vector of the plane and the position of a reference point inside the plane. A local coordinate system is defined such that the *u*-axis is parallel to the normal vector and the *v*- and *w*-axes are inside the plane. A natural choice of track parameters is:

$$q_1 = \psi,\ q_2 = \mathrm{d}v/\mathrm{d}u,\ q_3 = \mathrm{d}w/\mathrm{d}u,\ q_4 = v,\ q_5 = w$$



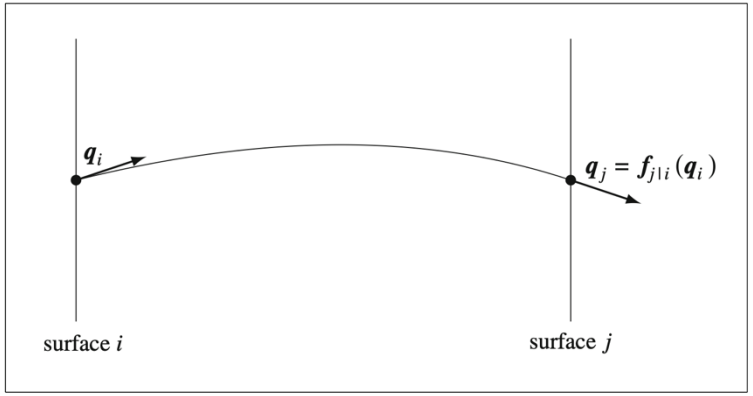
where $\psi = q/p$, d*v*/d*u* is the tangent of the angle between the projection of the track tangent into the (*u*,*v*)-plane and the *u*-axis, d*w*/d*u* is the tangent of the angle between the projection of the track tangent into the (*u*,*w*)-plane and the *u*-axis, and *v* and w are the local coordinates of the intersection point of the track with the plane.

Track Propagation

- The track model, given by the solution of the equations of motion, describes the functional dependence of the state vector $\boldsymbol{q_j}$ at a surface *j* on the state vector $\boldsymbol{q_i}$ at a different surface *i*:

$$\boldsymbol{q}_j = \boldsymbol{f}_{j|i}\left(\boldsymbol{q}_i\right)$$

The function $f_{j|i}$ is called the *track propagator* from surface *i* to surface *j*. When closed-form solutions of the equations of motion exist, e.g., in the two situations of $\boldsymbol{B}$ = 0 and homogeneous magnetic field, the track propagator can be written as an explicit function of the path length. For the helical solution in a homogeneous magnetic field, however, such an analytical formula exists only for propagation to cylinders with symmetry axis parallel to the field direction or to planes orthogonal to the field direction.



continue...

from previous page…

Track Propagation - *Homogeneous Magnetic Fields*

The helical track propagator takes the solution of the Equation of motion as a starting point. The solution can be written in the form:

$$\boldsymbol{r}(s) = \boldsymbol{r}_0 + \frac{\delta}{K}(\theta - \sin\theta)\hat{\boldsymbol{h}} + \frac{\sin\theta}{K}\hat{\boldsymbol{t}}_0 + \frac{\alpha}{K}(1 - \cos\theta)\hat{\boldsymbol{n}}_0$$

$$\boldsymbol{r}_0 = \boldsymbol{r}(s=0),\ \hat{\boldsymbol{h}} = \boldsymbol{B}/|\boldsymbol{B}|,\ \hat{\boldsymbol{t}} = \boldsymbol{p}/|\boldsymbol{p}|,\ \hat{\boldsymbol{n}} = (\hat{\boldsymbol{h}} \times \hat{\boldsymbol{t}})/\alpha,$$

$$\alpha = |\hat{\boldsymbol{h}} \times \hat{\boldsymbol{t}}|,\ \delta = \hat{\boldsymbol{h}} \cdot \hat{\boldsymbol{t}},\ K = -k\psi|\boldsymbol{B}|,\ \psi = q/p,\ \theta = Ks$$

Any point along the trajectory can be specified by a corresponding value of *s*. The equation of the unit tangent vector ***t*** is found by differentiating the above Equation with respect to *s*:

$$\hat{\boldsymbol{t}} = \frac{\mathrm{d}\boldsymbol{r}(s)}{\mathrm{d}s} = \delta(1 - \cos\theta)\hat{\boldsymbol{h}} + \cos\theta\hat{\boldsymbol{t}}_0 + \alpha\sin\theta\hat{\boldsymbol{n}}_0$$

For a given value of *s*, any desired set of track parameters can be calculated from above Equations for the positions and for the directions. In the helical track model, the momentum *p* is constant.

continue...

from previous page…

Track Propagation - *Inhomogeneous Magnetic Fields*

In an inhomogeneous magnetic field, the equations of motion have no exact closed-form solutions, and one must resort to either *a)* ***approximate solutions or b) numerical Runge–Kutta Method.***

a) Approximate Analytical Formula:

The magnetic field $\boldsymbol{B}(z) = (B_x, B_y, B_z)$ is assumed to depend only on the z-coordinate. The particle is assumed to move along the z-axis, and the track parameters are x, y, t_x, t_y, ψ, where t_x, t_y are the direction tangents. In this parametrization, the equations of motion are:

$$\frac{\mathrm{d}x}{\mathrm{d}z} = t_x, \quad \frac{\mathrm{d}y}{\mathrm{d}z} = t_y$$

$$\frac{\mathrm{d}t_x}{\mathrm{d}z} = h\left[t_x t_y B_x - (1+t_x^2)B_y + t_y B_z\right] = \boldsymbol{a}(z)\cdot\boldsymbol{B}(z) = \sum_{i_1=x,y,z} a_{i_1} B_{i_1}(z)$$

$$\frac{\mathrm{d}t_y}{\mathrm{d}z} = h\left[(1+t_y^2)B_x - t_x t_y B_y - t_x B_z\right] = \boldsymbol{b}(z)\cdot\boldsymbol{B}(z) = \sum_{i_1=x,y,z} b_{i_1} B_{i_1}(z)$$

$$\boldsymbol{a}(z) = h(t_x t_y,\ -1-t_x^2,\ t_y),\ \boldsymbol{b}(z) = h(1+t_y^2,\ -t_x t_y,\ -t_x)$$

$$\frac{\mathrm{d}\psi}{\mathrm{d}z} = 0, \quad h = k\psi\sqrt{(1+t_x^2+t_y^2)},\ \psi = q/p$$

continue...

from previous page…

Approximate Analytical Formula:

The aim is to find formulas for the extrapolation of (t_x, t_y) from z_0 to z_f; the extrapolation of x and y can then be performed by integration of the track directions. It will be convenient to obtain first a general formula for extrapolation of any function $T(t_x(z), t_y(z))$ from z_0 to z_f and only then to substitute $T = t_x$ and $T = t_y$ into the final formula.

Let a function T is given by $T(z) = T(t_x(z), t_y(z))$, then:

$$\frac{\mathrm{d}T(z)}{\mathrm{d}z} = \left(\frac{\partial T}{\partial t_x}\frac{\mathrm{d}t_x(z)}{\mathrm{d}z} + \frac{\partial T}{\partial t_x}\frac{\mathrm{d}t_y(z)}{\mathrm{d}z}\right) = \sum_{i_1=x,y,z} \overbrace{\left(\frac{\partial T}{\partial t_x}a_{i_1} + \frac{\partial T}{\partial t_x}b_{i_1}\right)}^{T_{i_1(z)}} B_{i_1}(z) = \sum_{i_1=x,y,z} T_{i_1}(z)B_{i_1}(z)$$

The derivatives of the new functions $T_{i1}(z)$ can also be expanded in the same way as the $T(z)$ derivative by introducing new functions $T_{i1i2}(z)$:

$$\frac{\mathrm{d}T_{i_1}(z)}{\mathrm{d}z} = \sum_{i_2=x,y,z} \overbrace{\left(\frac{\partial T_{i_1}}{\partial t_x}a_{i_2} + \frac{\partial T_{i_1}}{\partial t_y}b_{i_2}\right)}^{T_{i_1 i_2(z)}} B_{i_2}(z) = \sum_{i_2=x,y,z} T_{i_1 i_2}(z)B_{i_2}(z)$$

$$\Rightarrow \frac{\mathrm{d}T_{i_1\ldots i_{k-1}}(z)}{\mathrm{d}z} = \sum_{i_k=x,y,z} T_{i_1\ldots i_k}(z)B_{ik}(z), \text{ where } T_{i_1\ldots i_k}(z) \equiv \frac{\partial T_{i_1\ldots i_{k-1}}(z)}{\partial t_x}a_{ik} + \frac{\partial T_{i_1\ldots i_{k-1}}(z)}{\partial t_y}b_{ik}$$

Using Equation the function $T(z_f)$ can be written as:

$$T(z_f) = T(z_0) + \int_{z_0}^{z_f}\frac{\mathrm{d}T(z_1)}{\mathrm{d}z_1}\mathrm{d}z_1 = T(z_0) + \sum_{i_1}\int_{z_0}^{z_f} T_{i_1}(z_1)B_{i1}(z_1)\mathrm{d}z_1$$

$$= T(z_0) + \sum_{i_1}\int_{z_0}^{z_f}\left(T_{i_1}(z_0) + \int_{z_0}^{z_1}\frac{\mathrm{d}T_{i_1}(z_2)}{\mathrm{d}z_2}\mathrm{d}z_2\right)B_{i_1}(z_1)\mathrm{d}z_1$$

$$= T(z_0) + \sum_{i_1} T_{i_1}(z_0)\int_{z_0}^{z_f} B_{i_1}(z_1)\mathrm{d}z_1 + \sum_{i_1}\int_{z_0}^{z_f} B_{i_1}(z_1)\int_{z_0}^{z_1}\sum_{i_2} B_{i_2}(z_2)T_{i_1 i_2}(z_2)\mathrm{d}z_2\mathrm{d}z_1 = \ldots$$

$$T(z) = T(z_0) + \sum_{k=1}^{n}\sum_{i_1,\ldots,i_k} T_{i_1,\ldots,i_k}(z_0)$$
$$\times\left(\int_{z_0}^{z_f} B_{i_1}(z_1)\ldots\int_{z_0}^{z_{k-1}} B_{i_k}(z_k)\mathrm{d}z_k\ldots\mathrm{d}z_1\right)$$
$$+O\left(\frac{(kB(q/p)(z_f - z_0))^{n+1}}{(n+1)!}\right)$$

Substituting $T=t_x$ and t_y into the formula one can obtain the extrapolated track parameters. Coefficients $t_{xi1..ik}$, $t_{yi1..ik}$ are calculated using the recursive formula.

from previous page…

b) numerical methods (Runge–Kutta),

- Runge–Kutta methods are iterative algorithms for the approximate numerical solutions of ordinary differential equations, given initial values.
- Runge–Kutta–Nyström methods are specialized Runge–Kutta methods that are optimized for second-order differential equations such as the one:

$$\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}s^2} = k\frac{q}{|\boldsymbol{p}|}\underbrace{\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s}}_{\dot{\boldsymbol{r}}(s)} \times \boldsymbol{B}(\boldsymbol{r}(s)) = \boldsymbol{F}(s, \boldsymbol{r}(s), \dot{\boldsymbol{r}}(s))$$

- In the fourth-order version a step of length h, starting at $s = s_n$, is computed by:

$$\boldsymbol{r}_{n+1} = \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + h^2\left(\boldsymbol{k}_1 + \boldsymbol{k}_2 + \boldsymbol{k}_3\right)/6, \quad \dot{\boldsymbol{r}}_{n+1} = \dot{\boldsymbol{r}}_n + h\left(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4\right)/6,$$

$$\boldsymbol{k}_1 = \boldsymbol{F}(s_n, \boldsymbol{r}_n, \dot{\boldsymbol{r}}_n)$$

$$\boldsymbol{k}_2 = \boldsymbol{F}\left(s_n + h/2, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n/2 + h^2\boldsymbol{k}_1/8, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_1/2\right)$$

$$\boldsymbol{k}_3 = \boldsymbol{F}\left(s_n + h/2, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n/2 + h^2\boldsymbol{k}_1/8, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_2/2\right)$$

$$\boldsymbol{k}_4 = \boldsymbol{F}\left(s_n + h, \boldsymbol{r}_n + h\dot{\boldsymbol{r}}_n + h^2\boldsymbol{k}_3/2, \dot{\boldsymbol{r}}_n + h\boldsymbol{k}_3\right)$$

where $\boldsymbol{r}_n$ is the position of the particle at $s = s_n$, $\boldsymbol{r}^*_n$ is the unit tangent vector. The magnetic field needs to be looked up three times per step, at the positions $\boldsymbol{r}_n$, $\boldsymbol{r}_n + h\,\boldsymbol{r}^*_n/2 + h^2\boldsymbol{k}_1/8$, and $\boldsymbol{r}_n + \boldsymbol{r}_n + h\,\boldsymbol{r}^*_n + h^2\boldsymbol{k}_3/2$.

3. Track Finding Techniques

There is no systematic theory of track finding yet. We will present some of an extensive list of basic techniques which have been successfully used, stand-alone or in combination, in past and present experiments. Among them are the

- ***Conformal Mapping Transformation***
- ***Artificial Retina***
- ***Hough Transform***
- ***Legendre Transform***
- ***Neural Networks***
- ***Kalman Filter***

As track finding in most cases delivers some candidates that do not correspond to actual particle tracks, we will discuss some methods for an *efficient selection of valid track candidates*.

Conformal Transformation

In the conformal algorithm, point coordinates in global Euclidean space (*x, y*) are translated into the conformal space (*u, υ*). The idea behind this coordinate transformation is that circles passing through the origin of a coordinate system (*x, y*) can be transformed into straight lines in a new coordinate system (*u, υ*). The circle equation in global coordinates (*x, y*):

$$(x-a)^2 + (y-b)^2 = R^2$$

is equivalent to a straight line in the (*u, υ*) plane:

$$v = -\frac{a}{b}u + \frac{1}{2b}$$

if the circle is passing through the origin such that *R* is fixed to $R^2 = a^2 + b^2$ and if the following transformations are applied:

$$u = \frac{x}{x^2+y^2}, \qquad v = \frac{y}{x^2+y^2}$$

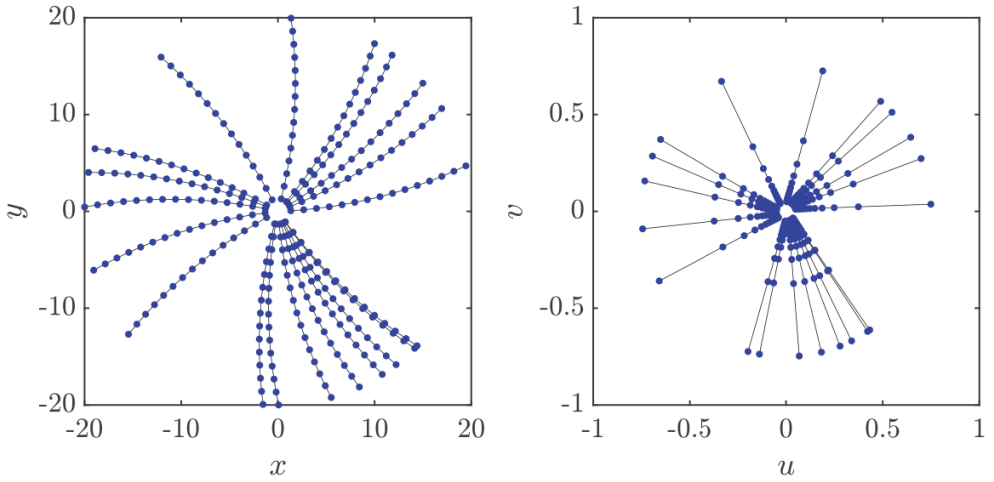
Through the application of the conformal mapping, finding tracks of charged particles bent by a homogeneous magnetic field can be reduced to a search for straight lines. The radial order of the hit positions is inverted in the conformal space with respect to the global space: hits on the innermost part of the detector are mapped to outer regions in the (*u, υ*) plane and vice versa.

from previous page…

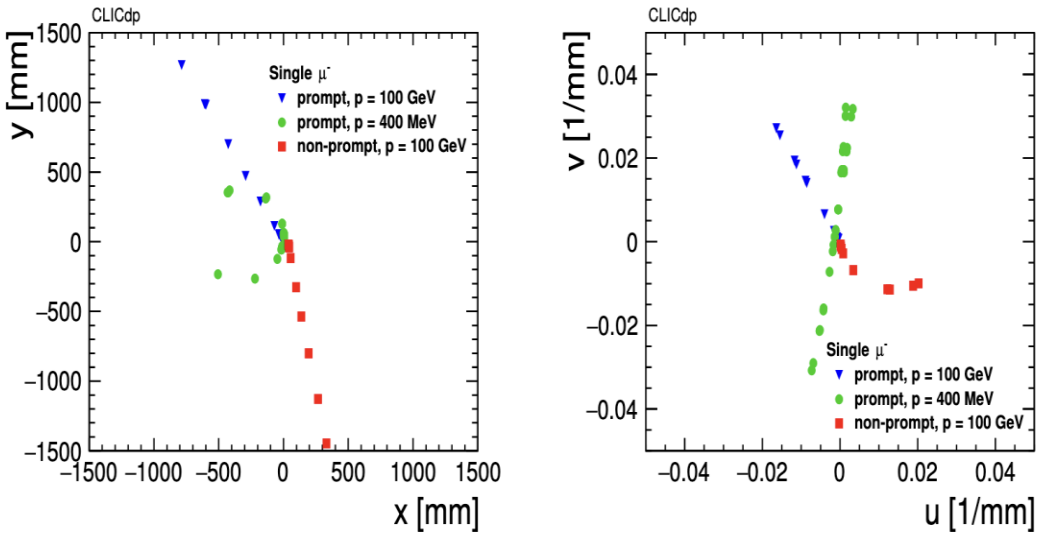
Conformal Transformation

$$2au + 2bv = 1$$

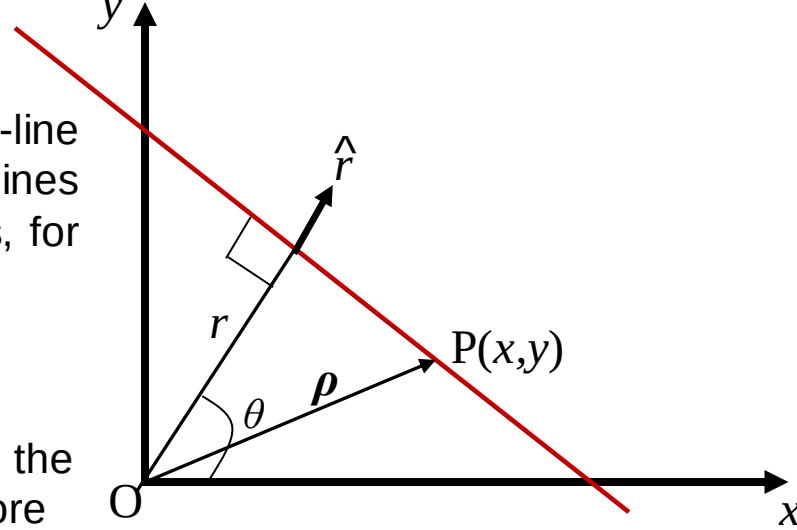
This is the equation of a straight line in the (*u,u*) plane with distance *d*=1/(2*R*) from the origin. A circle with a large radius *R* or small curvature is therefore transformed into a line that passes very close to the origin. In the limit of zero curvature, the circle becomes a line transformed into itself by the conformal mapping. Both circle finding and circle fitting can be simplified by this transformation from circles to straight lines.



The simple pattern recognition technique suggested in original applications of conformal mapping consists of grouping hits aligned in the same direction in the (*u, u*) plane, by searching for peaks in the angular hit distribution in conformal space. However, this method does not consider deviations from the straight-line path, which can arise in real measurements. These deviations come either from multiple scattering or from the mathematical approximations introduced in the conformal mapping formulas, as is the case of particles not produced at the origin of the (*x, y*) plane, also known as displaced or non-prompt particles.



Hough Transform

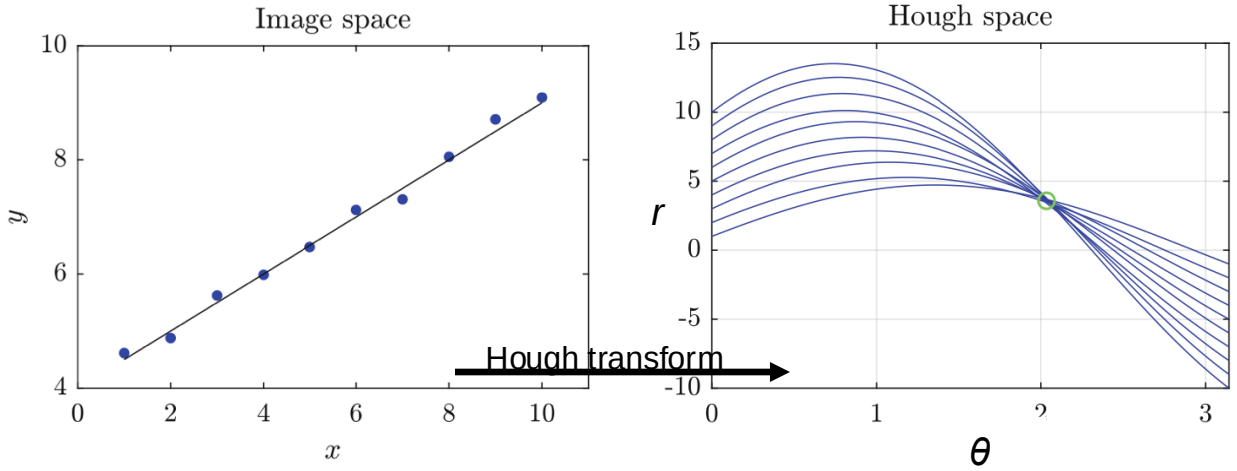


The Hough transform is used to detect straight lines can be detected. In general, the straight-line $y = ax + b$ can be represented as a point (b, a) in the parameter space. However, vertical lines pose a problem. They would give rise to unbounded values of the slope parameter a. Thus, for computational reasons, Duda and Hart proposed the use of the Hesse normal form of a line

$$\boldsymbol{\rho} \cdot \hat{\boldsymbol{r}} - r = 0 \Rightarrow x\cos\theta + y\sin\theta = r$$

where r is the distance from the origin to the closest point on the straight line, and θ is the angle between the axis and the line connecting the origin with that closest point. It is therefore possible to associate with each line of the image a pair (r,θ). The (r,θ) plane is sometimes referred to as *Hough space* for the set of straight lines in two dimensions. The Hough transform is a case of the general Radon transform.

Given a *single point* in the plane, the set of *all* straight lines going through that point corresponds to a sinusoidal curve in the (r, θ) plane, which is unique to that point. A set of two or more points that form a straight line will produce sinusoids crossing at the (r, θ) for that line. Thus, the problem of detecting collinear points can be converted to the problem of finding concurrent curves, i.e. intersection points in the Hough space.



In practice, the measured points do not lie exactly on a straight line, and the lines in the Hough space do not intersect exactly in a single point. The usual approach is to define a binning in the Hough space and count the number of lines crossing each bin. Peaks in the 2D histogram correspond to lines that are close to many points in the image space. The size of the bins depends on the distribution of the measurement errors and can be tuned on simulated tracks.

Continue...

from previous page…

Hough Transform

If the curve to be found in the image space is a circle (as in the case of detecting Cerenkov rings) in general position with the equation:

$$(x - x_0)^2 + (y - y_0)^2 = R^2$$

the constraint that the circle passes through the point (x_i, y_i) defines a second order surface in the 3D Hough space of (x_0, y_0, z):

$$z = (x_i - x_0)^2 + (y_i - y_0)^2, \quad \text{with } z = R^2$$

It follows that finding circles requires finding intersection points of surfaces in a 3D histogram, which is computationally much more expensive than the same problem in 2D.

An alternative is the *randomized Hough transform*, that randomly selects triplets of points. The centre of the circle passing through the triplet (that defines a triangle), and its radius are stored in a 1D histogram. Peak finding can be done in 3D or in the 2D histogram of the circle centres.

Continue...

from previous page…

Application of Hough Transform

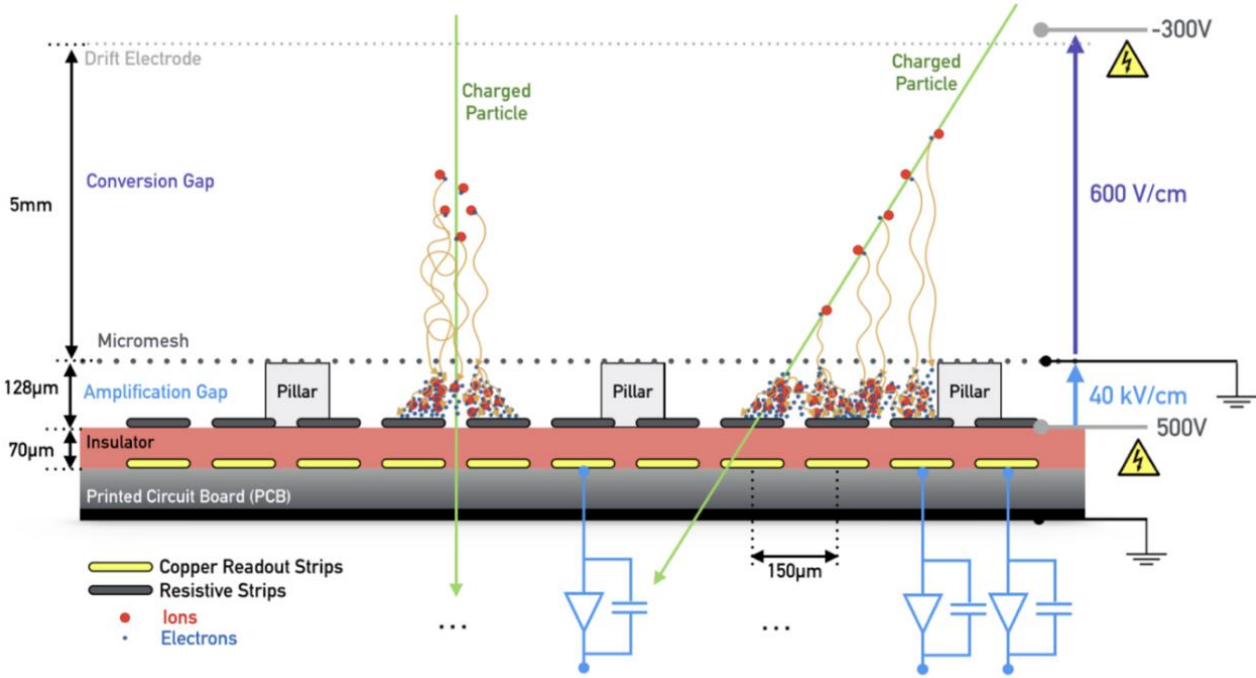
Application of Hough transform in micromegas detector (example of a MPGD-Micro Pattern Gaseous Detectors).

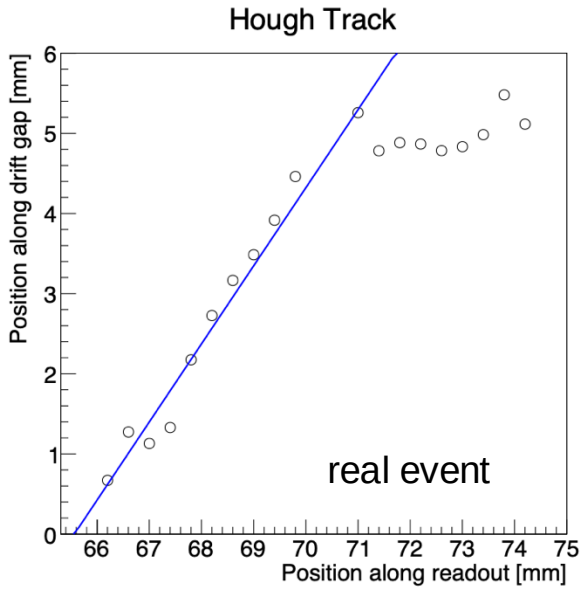
- To exploit the timing information per strip the μTPC reconstruction method has been developed based on Hough transform that can provide a very precise track segment in a single detector gap.
- The clustering algorithm distinguishes multiple tracks and/or hits due to noise or delta rays.

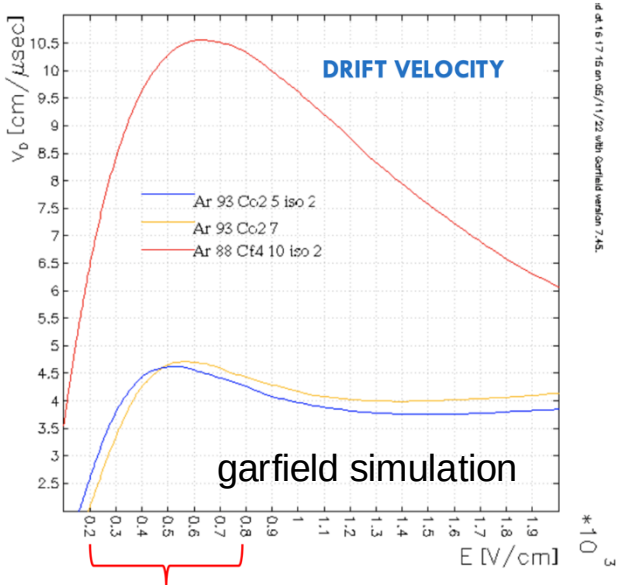
- For normal tracks, a cluster building algorithm can be used allowing for a single missing strip between two strips per cluster. Cluster building is performed in ascending strip numbers. The charge weighted track position is

$$x_p = \frac{\sum_i Q_i x_i}{\sum_i Q_i},$$

$$x_i = \text{strip coordinate},\ Q_i = \text{strip charge}$$







Artificial Retina

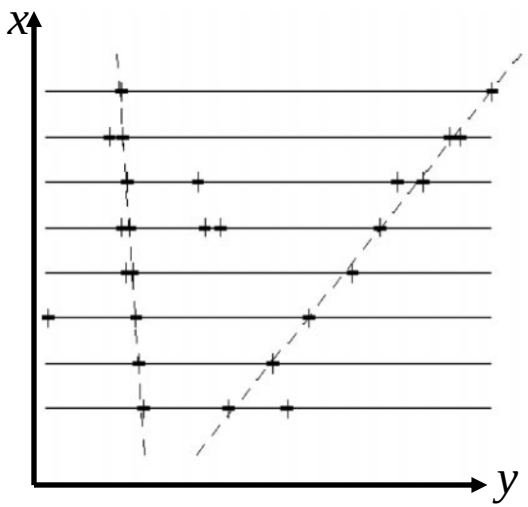
The concept of the "Artificial Retina" was introduced like the Hough transform, it relies on a partition of the track parameter space into cells.

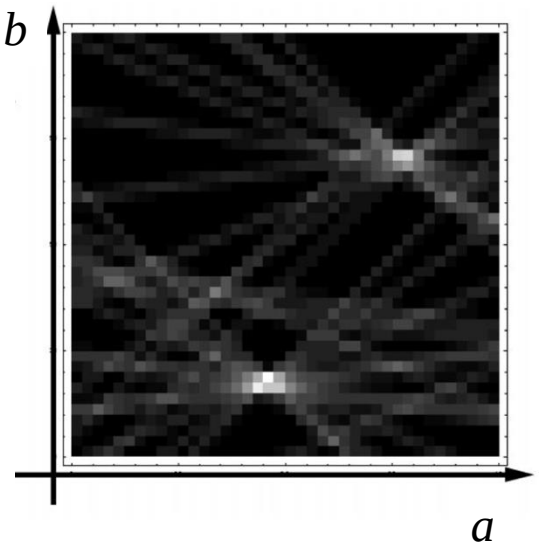
Consider an events with two tracks passing through several parallel detector layers. Given the coordinates of the hits, we want to estimate the parameters of all the tracks that generated them. Each track is identified by two parameters (*a,b*) and the coordinates of the intersections of the track with the detector layers are $y_i(a,b) = ax_i + b$ where *i* is the layer number. We build a bidimensional grid in parameter space where each grid element is identified by a pair of values (*a,b*). For each event we then compute the response function, Gaussian type like:

$$R(a,b) = \sum_{i=1}^{n}\sum_{j=1}^{m} e^{-s_{ij}^2/2\sigma_i^2}, \qquad s_{ij} = y_{ij} - (ax_i + b)$$

where $s_{ij} = y_{ij} - (ax_i + b)$ is the distance of hit *j* in layer *i* from the ideal track position in layer *i*, and σ_i is a scale parameter that regulates the width of the receptive field in layer *i*. The sum is extended to all hits present in all the layers and computed for all elements in the grid.

Other response functions are of course possible (like a Lorentzian), and their shape and width can be adjusted for optimal performance. As with the Hough transform, track candidates correspond to the local maxima of intensity in parameter space.





- The artificial retina is eminently suitable for high-speed track finding, as it can be highly parallelized and implemented on commercial FPGAs.

Geometrical Legendre Transform

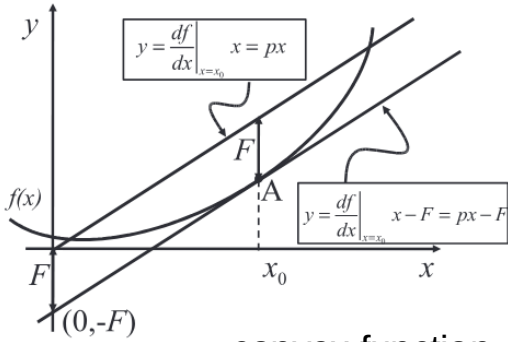
- The Legendre transform is an extension of the Hough transform, used to find common tangent lines to a set of circles.

- The Legendre transform is a well-known mathematical tool in Thermodynamics and Analytical Mechanics. Consider a convex function and a straight line of the form $y = px + a$, where p and a are the slope and intercept, respectively. For a value p of the slope the Legendre transform of the function $f(x)$ is defined as follows:

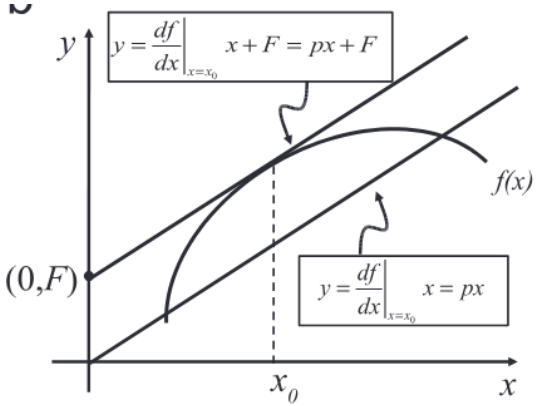
$$F(p) = \sup_x [px - f(x)] = -\inf_x [f(x) - px]$$

The notation sup_*x* indicates the maximization of the function *px* – *f*(*x*) with respect to *x* for constant *p*, while inf_*x* indicates the minimization of *f*(*x*) - *px* with respect to *x* for constant *p*.

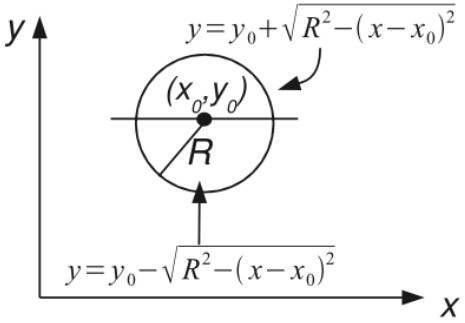
As it is demonstrated, for a given value *p* of the slope, this transform finds the point of *f*(*x*), where the tangent line has a slope *p*. The intersection of the straight line with the y-axis is given by –*F*(*p*) (in convex functions). Thus, each point (*p*,*F*(*p*)) in Legendre space represents a line, tangent to the curve *f*(*x*). The Legendre transform can also be applied to a concave function.

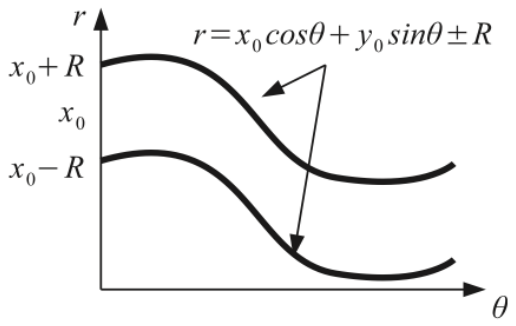


convex function



concave function





Legendre transform of a circle:

$$f(x) = \begin{cases} f_1(x) = y_0 + \sqrt{R^2 - (x - x_0)^2} \\ f_2(x) = y_0 - \sqrt{R^2 - (x - x_0)^2} \end{cases}$$

$$f(x) \overset{\mathcal{L}}{\longleftrightarrow} \begin{cases} r = x_0 \cos\theta + y_0 \sin\theta + R & \text{for concave} \\ r = x_0 \cos\theta + y_0 \sin\theta - R & \text{for convex} \end{cases}$$

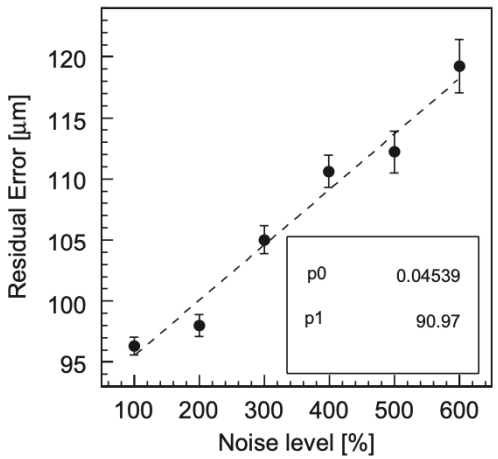
Representation of the circle in Legendre transformation space. The circle corresponds to two sinograms in the Legendre transformation space.

Continue...

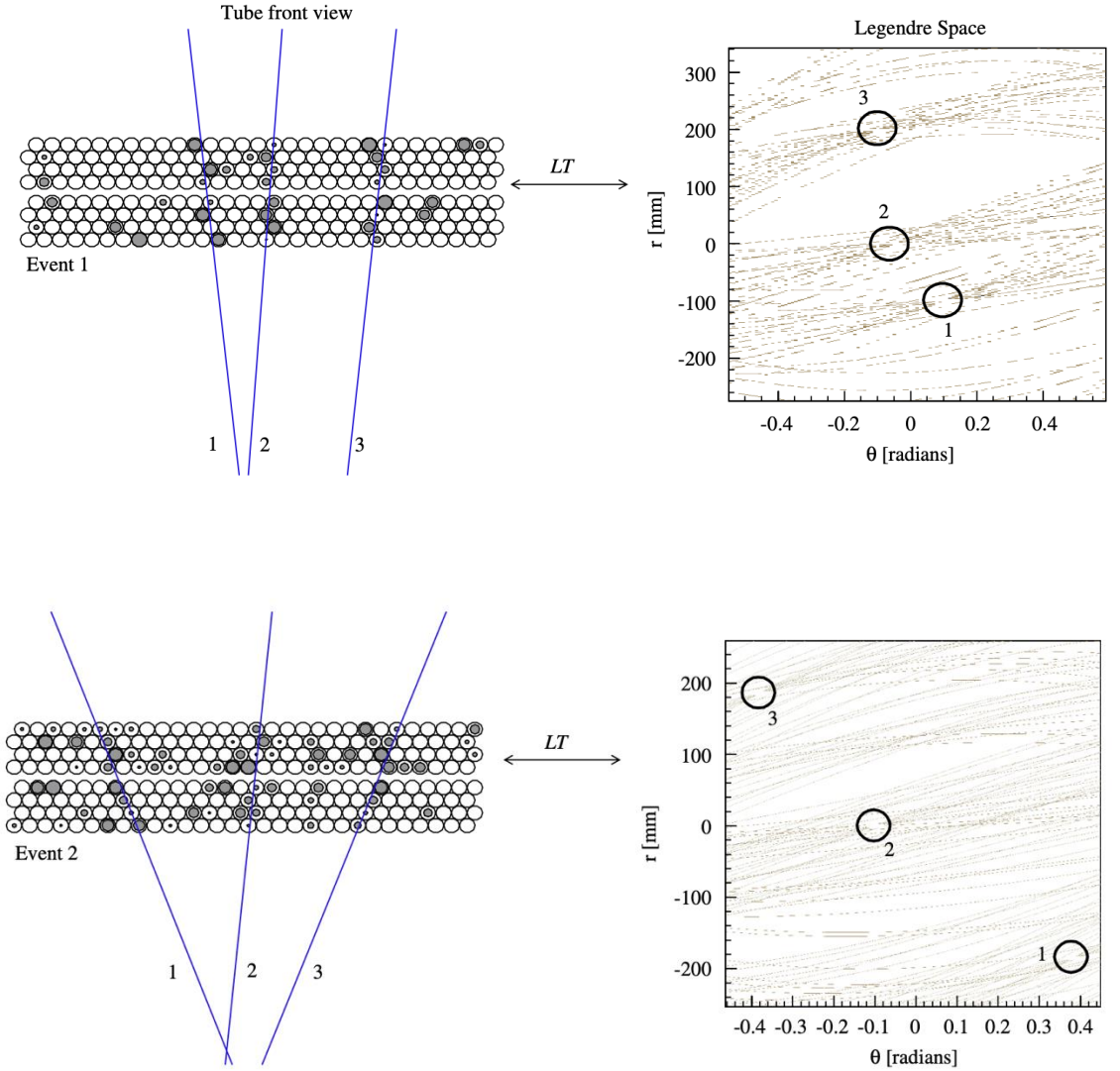
from previous page…

Legendre Transform

Study the Legendre transform method using a Monte Carlo algorithm to produce random lines and create the hits for each tube. As an example, the algorithm is tested in the Monitored Drift Tube detector of the ATLAS experiment, a straw type chamber. In this study, a Drift Chamber of eight tube layers including 36 tubes in each layer, is used. The diameter of each tube is 30 mm. After calculating the hits, a Gaussian measurement error is applied to each hit. Moreover, random hits are generated to simulate random noise hits in the detector. The study is performed for single and multi-track events. In each case, the reconstructed line parameters are calculated.



resolution versus noise using hits with a standard deviation of 100 μm. The data are simulated with noise up to 600%.



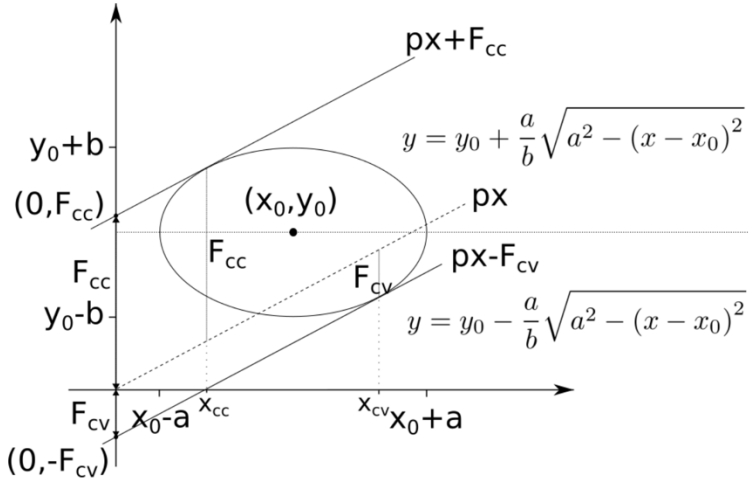
Drift chamber with two multi-track events with noise level of 100% and 200%, for Events 1 and 2, respectively. Each one of the events were reconstructed using the Legendre transform method with their corresponding Legendre transforms. The circles in Legendre space graphs denote the points with the highest height, corresponding to the three tracks.

Continue...

from previous page…

Legendre Transform

- Application of Legendre transform in pattern recognition method that identifies the common tangent lines of a set of **ellipses**:

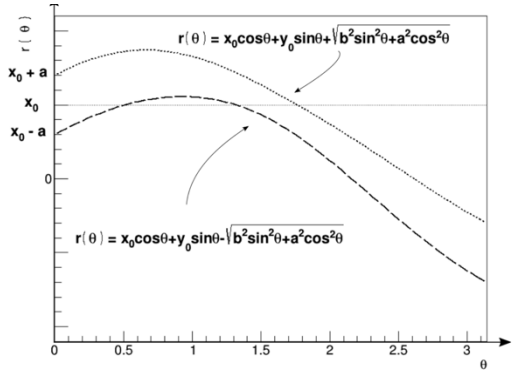


Elipse:

$$f(x) = \begin{cases} f_1(x) = y_0 + \frac{b}{a}\sqrt{a^2 - (x - x_0)^2} & \text{for the concave part} \\ f_2(x) = y_0 - \frac{b}{a}\sqrt{a^2 - (x - x_0)^2} & \text{for the convex part} \end{cases}$$

Legendre Trasform:

$$f(x) \leftrightarrow r(\theta) = \begin{cases} x_0 \cos\theta + y_0 \sin\theta + \sqrt{b^2 \sin^2\theta + a^2 \cos^2\theta} & \text{for the concave part} \\ x_0 \cos\theta + y_0 \sin\theta - \sqrt{b^2 \sin^2\theta + a^2 \cos^2\theta} & \text{for the convex part} \end{cases}$$



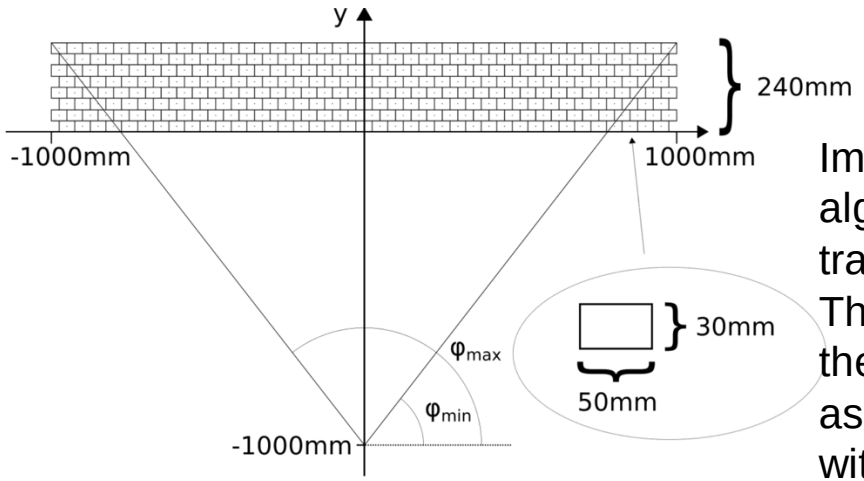


Figure 8: Specifications of the detector and the chambers.

Implementation of a Monte Carlo algorithm that produces random two track events and assigning noise hits. These tracks pass through the cell of the different layers of the detector, assigning ellipses that are co-eccentric within the different cells of the chamber.

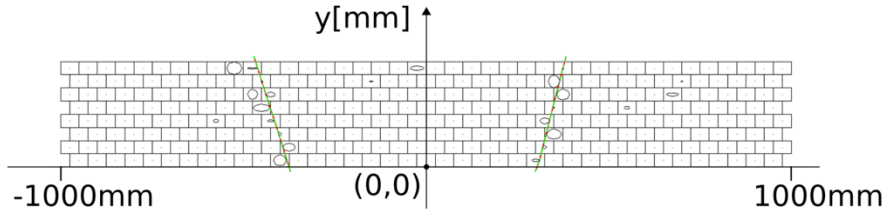
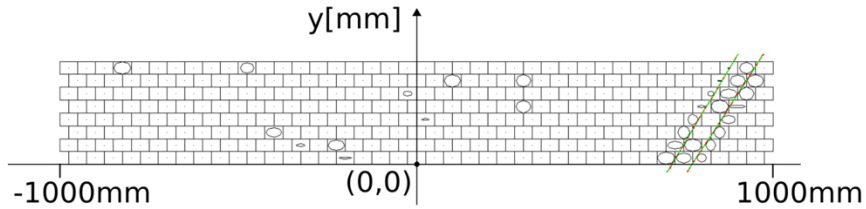


Figure 11: A dual track event with 50% noise. The green lines are the reconstructed tracks. The red dashed lines are the initial lines for reference. From this set of ellipses, the algorithm created the histograms in figures 4 and 5



Continue...

from previous page…

Legendre Transform

- A method of reconstructing the **circle** parameters from a set of datapoints on a plane based on the geometrical Legendre transform. This method can be used to identify Cerenkov rings.
- For three datapoints on a plane a circle is constructed, and then its Legendre transform:

$$x_{\text{est}} = \frac{m_t m_r (y_3 - y_1) + m_r (x_2 + x_3) - m_t (x_1 + x_2)}{2(m_r - m_t)} = x_0$$

$$y_{\text{est}} = -\frac{1}{m_r}\left(x_{\text{est}} - \frac{x_1 + x_2}{2}\right) + \frac{y_1 + y_2}{2} = y_0$$

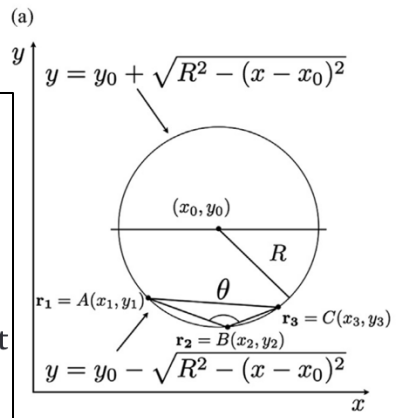
$$R_{\text{est}} = \sqrt{(x_{\text{est}} - x_1)^2 + (y_{\text{est}} - y_1)^2}$$

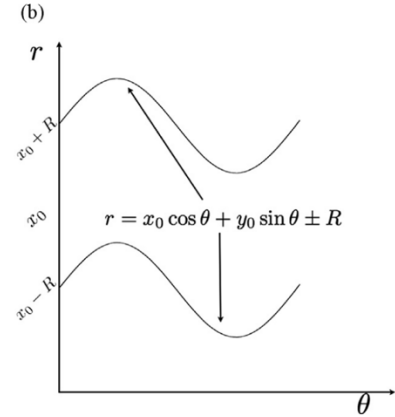
$$m_r = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{and} \quad m_t = \frac{y_3 - y_2}{x_3 - x_2}$$

circle:
$$f(x) = \begin{cases} f_1(x) = y_0 + \sqrt{R^2 - (x - x_0)^2} & \text{concave part} \\ f_2(x) = y_0 - \sqrt{R^2 - (x - x_0)^2} & \text{convex part} \end{cases}$$

$$f(x) \leftrightarrow \mathcal{L} \begin{cases} r_1 = x_0 \cos\theta + y_0 \sin\theta + R_0, & \text{concave part} \\ r_2 = x_0 \cos\theta + y_0 \sin\theta - R_0, & \text{convex part} \end{cases}$$

Legendre Transform of the circle:





- Considering *n* given datapoints, all the possible circles for each triplet of datapoints are constructed. The Legendre transform, of all reconstructed circles, will be given by the sinograms:

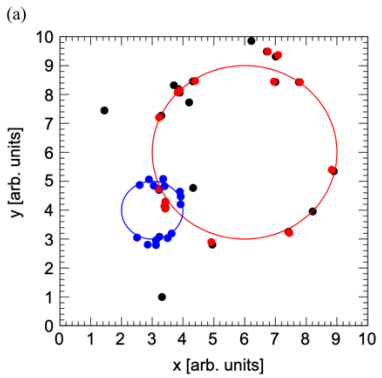
$$r_{1,2} = \sum_{i=1}^{k} (x^i_{\text{est}} \cos\theta + y^i_{\text{est}} \sin\theta \pm R^i_{\text{est}})$$

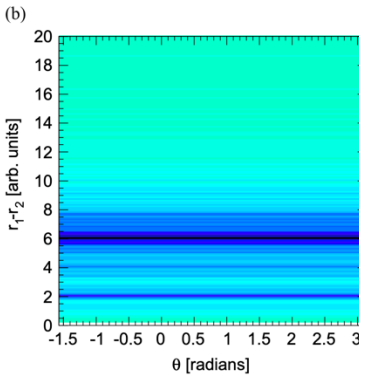
- Also consider the difference and sum of r1 and r2:

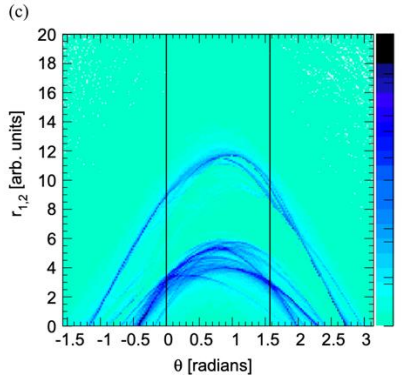
$$r_1 - r_2 = 2 \sum^{k} R^i_{\text{est}}$$

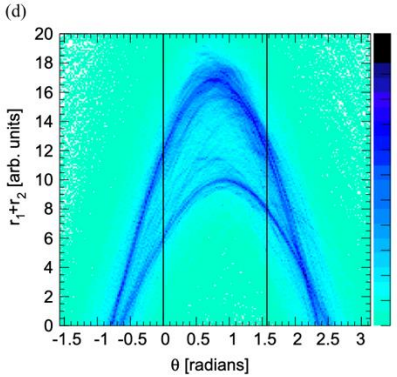
$$r_1 + r_2 = 2 \sum^{k} (x^i_{\text{est}} \cos\theta + y^i_{\text{est}} \sin\theta)$$

(a) The red and blue datapoints originate from the two circles (red and blue lines) having received a smearing of 10%. The outliers/noise hits (black datapoints) are on a 50% percentage of the circle's datapoints. (b) The Legendre space of r_1-r_2 from the datapoints of (a). (c) Concave and convex representations of the circle's datapoints. (d) The Legendre space of r_1+r_2 from the datapoints of (a).
Estimation of the radius of the circles by searching for maxima in r_1-r_2 while the centre of circles by searching in r_1+r_2 distributions.





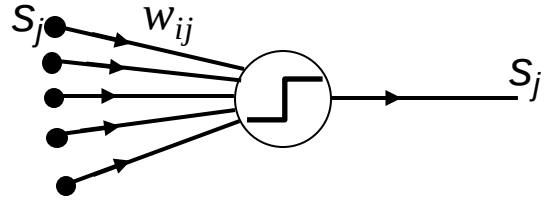




- The Legendre Transform eminently suitable for high-speed track finding, as it can be highly parallelized and implemented on FPGAs

Neural Networks

- The application of neural networks to tracking is of the Hopfield type, the neurons being track segments that connect observations in adjacent or nearby layers of the detector.



- A Hopfield network is a fully connected network with a single layer of neurons. In the simplest case, the neurons are binary with two states: $s_i = \pm 1$, $i = 1, \ldots, n$. Each pair (i,j) of neurons has a fixed connection weight/synapses w_{ij} with $w_{ij} = w_{ji}$ and $w_{ii} = 0$. The states of the neurons evolve in discrete time steps according to the rule:

$$s_i(t) = \text{sign}\left[\sum_{j=1}^{n} w_{ij} s_j(t-1)\right]$$

The network has an associated "Energy" function:

$$E(\boldsymbol{s}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} s_i s_j, \quad \boldsymbol{s} = (s_1, s_2, \ldots, s_n), \quad \text{state of network}$$

- The aim is to find the global minimum. To this end, thermal noise is introduced in the network. At temperature T, the state $\boldsymbol{s}$ is Boltzmann distributed with the probability function and partition function Z:

$$P(s) = \frac{1}{Z} e^{-E(s)/T}, \qquad Z = \sum_{s} e^{-E(s)/T}$$

Continue...

from previous page…

Neural Networks

- As the number of possible states rises exponentially with the number of neurons, the partition function Z is computed in the mean-field approximation, and the thermal average u_i of s_i is given by:

$$v_i = < s_i >_T = \tanh\left(-\frac{1}{T}\frac{\partial E}{\partial v_i}\right)$$

where the states $\boldsymbol{u}$ = (u_1 ,.., u_n) are now continuous in the interval (−1,1). The definition of the new energy function $E(\boldsymbol{u})$ is analogous to $E(\boldsymbol{s})$:

$$E(\boldsymbol{v}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} v_i v_j$$

and the update is modified accordingly:

$$v_i(t) = \tanh\left[\frac{1}{T}\sum_{j=1}^{n} w_{ij} v_i v_j\right]$$

- Finding the global minimum of the energy function is facilitated by deterministic simulating annealing algorithm. First, the energy function is minimized at high temperature; the temperature is then lowered according to a predefined cooling or annealing schedule. At low temperature, the states of the network are close to either +1(active) or −1(inactive).

- To keep the number of neurons manageable, geometric cuts ensure that only segments that can be part of an actual track in the momentum range of interest are used as neurons.

Track Candidate Selection

- After track finding, track candidates may share hits.
- If two candidates share more hits than is deemed acceptable, for instance more than one, the track candidates are called *incompatible*.
- The *incompatibility relation* can be represented by a *graph* (*V,E*), where the *n* vertices $u_i \in V$, $i = 1, \ldots, n$ are the track candidates.
- Two incompatible track candidates u_i, and u_j are connected by the edge $e_{ij} = e_{ji}$, which is defined as the unordered pair (u_i, u_j).
- The number of compatible track candidates can be maximized by finding an independent set of vertices of maximal size, i.e., a subset $V_1 \subseteq V$ of vertices, no two of which are connected by an edge.
- Finding the largest independent set is not necessarily the best approach for finding an “optimal” set of track candidates, as the quality of the track candidates should be considered, too. If the quality of the candidate u_i is quantified by a positive weight w_i, the problem is to find an independent set that maximizes the $\Sigma_i w_i$.

Linear Approaches to Circle and Helix Fitting

- Will present a couple of linearized fits of space points to circles and helices.

1. *Conformal Mapping Method*:

- The conformal transformation described in track finding analysis can be generalized to deal with circles passing close the origin. The values obtained however, are only approximate, since equation $R^2 = a^2+b^2$ forces the circle to pass through the origin and the important third parameter determining a track, the impact parameter ε, is lost.
- The situation can be resolved by allowing for a small difference between R^2 and a^2+b^2 , which we call it δ:
  $\delta = R^2 - (a^2 + b^2)$
- For $\delta << R^2$ we then have, instead of a straight line, a parabola with a very small curvature:

$$v = \frac{1}{2b}\left(1 - \frac{\delta}{4b^2}\right) - u\frac{a}{b}\left(1 - \frac{\delta}{2b^2}\right) - u^2\delta\frac{R^2}{2b^3}$$

here terms of the order of δ^2 and higher have been neglected. Using the approximations:

$$\left(1 - \frac{\delta}{4b^2}\right) \approx 1, \quad \left(1 - \frac{\delta}{2b^2}\right) \approx 1, \text{ for } \frac{\delta}{2b^2} \ll 1$$

the equation for the parabola then becomes:

$$v = \frac{1}{2b} - \frac{a}{b}u - \epsilon\left(\frac{R}{b}\right)^3 u^2, \quad \epsilon = R - \sqrt{a^2+b^2} \approx \frac{\delta}{2R}$$

this equation is identical to a linear equation except for the term representing the curvature, which is proportional to the impact parameter. An ordinary parabola fit in (*u, v*) space will therefore yield the three circle parameters *a*, *b*, ε in (*x, y*) space.

Continue...

from previous page…

2. *Chernov and Ososkov's Method*:

- The task of fitting a circular track to a set of measurements is equal to minimizing the function:

$$\chi^2 = \sum_{i=1}^{n} d_i^2$$

where d_i are measurement residuals orthogonal to the particle trajectory:

$$d_i = \pm \left[\sqrt{(x_i - a)^2 + (y_i - b)^2} - R \right], \quad i = 1, \ldots, n$$

where *a, b*, and *R* are the coordinates of the circle centre and the radius. The approach of Chernov and Ososkov is to simplify this non-linear minimization problem by introducing an approximate expression for the residuals d_i:

$$d_i \approx \pm \frac{\left[(x_i - a)^2 + (y_i - b)^2 - R^2 \right]}{2R}$$

if the residuals are small compared to the circle radius. The equations obtained by differentiating χ^2 with respect to the circle parameters and setting these to zero are quartic (polynomial equations of 4th degree) and can be solved efficiently by a standard Newton iteration procedure.

Continue...

Vertex Reconstruction

- Vertex finding is the search for clusters of tracks that originate at the same point in space.
- Vertex finding is the process of dividing all or a subset of the reconstructed tracks in an event into classes such that presumably all tracks in a class are produced at the same vertex. Vertices in an event can be classified as primary vertices or secondary vertices.
- In a fixed target experiment, a primary vertex is the point where a beam particle collides with a target particle; in a collider experiment, a primary vertex is the point where two beam particles collide.
- A secondary vertex is the point where an unstable particle decays, or where a particle interacts with the material of the detector. The search for secondary vertices is often based on a well-reconstructed primary vertex or vertices.

Continue...

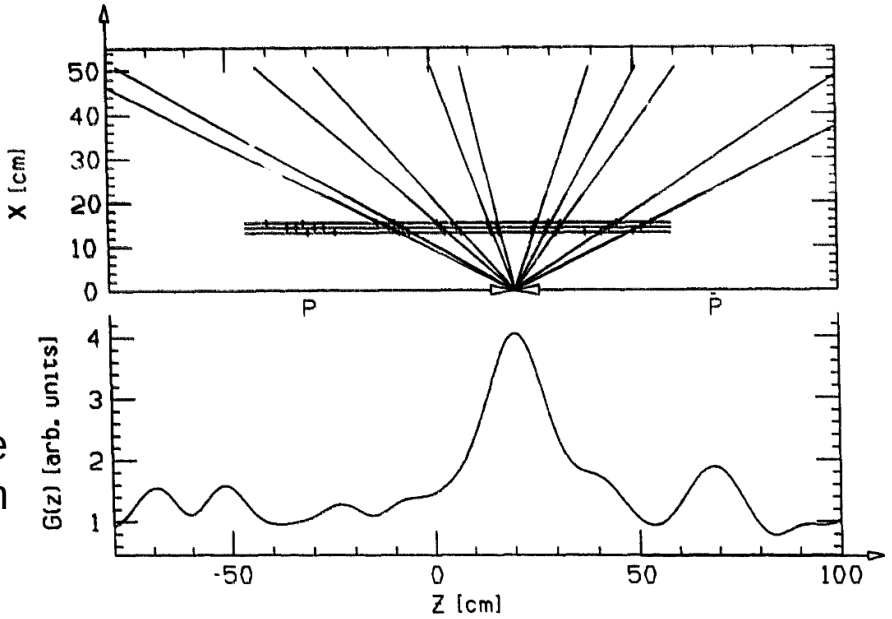
from previous page…

Vertex Reconstruction

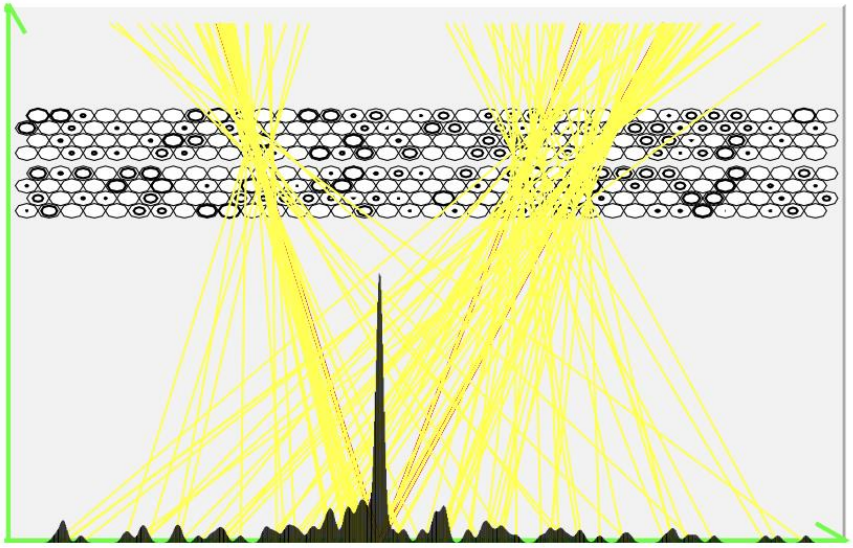
- ***Hits from a vertex chamber:*** The vertex along the beam line, z_u, is estimated from the hits in a global method without benefit of tracking. It maximizes the following summation vertex function over all hits in the vertex chamber:

$$G(z) = \sum_{k,m,n} \exp\left(-\left[(z - z_{k,3})(x_3 - x_n)/x_3 + z_{k,3} - z_{m,n}\right]^2 / 2\sigma^2\right)$$

The subscript m (m = 1,…,m(n)) is the number of hits in plane n (n = 1, 2, 3). The coordinate of plane n is x_n cm from the beam line at x = 0. The index k steps through the hits of plane of the outer plane.



- ***Hits from straw chambers:*** Apply the Legendre transform and clustering algorithms to the set of hits, in a fast and coarse way and extract the lines and respective associated hits per line. Then the associated hits form the vertex function $G(z)$ as before. After maximizing the vertex function $G(z)$ a vertex point is extracted.



References

- R. Fruhwirth, A. Strandlie, “Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors”, 2020, Springer. (This lecture is mostly based on materials from this and the references within it),
- R. Fruhwirth, M. Regler, R.K. Bock, H.Grote, D. Notz, “Data Analysis Techniques for High Energy Physics”, 200 Cambridge University Books.
- C. Fabjan, H. Schopper, “Particle Physics Reference Library, Volume 2: Detector for Particles and Radiation, 2020 Springer. Chapter 13: “Pattern Recognition and Reconstruction” by R. Frühwirth, E. Brondolin, and A. Strandlie.
- W. Blum, W. Riegler, L. Rolandi, “Particle Detection with Drift Chambers”, 2008, Springer.
- K. Hanagaki, J. Tanaka, M. Tomato, Y. Yamazaki, “Experimental Techniques in Modern High Energy Physics, 2021, Springer.

Backups

Statistics & Numerical Methods

Will discuss some statistical and numerical methods in data Analysis and Reconstruction techniques.

1. Analysis of **Functions Minimization:** Gradient-based methods and a popular non-gradient method will be presented.

2. Discussion of **Statistical Models and the Estimation of Model Parameters**. The basics of linear and nonlinear regression models and state space models are presented, including least-squares estimation and the (extended) Kalman filter. *(in Backups)*

1. Function Minimization

- The minimization (or maximization) of a multivariate function $f(\boldsymbol{x})$ is a task in solving non-linear systems of equations, clustering, maximum-likelihood estimation, function and model fitting, supervised learning, etc.

- A basic classification of minimization methods distinguishes between methods that require the computation of the gradient or even the Hessian matrix of the function and gradient-free methods. ”Hessian matrix describes the local curvature of a function of many variables. It is a square matrix of second-order partial derivates of a scalar-valued function, or scalar field”

1. ***Methods require gradient and/or Hessian martrix***

2. ***Gradient-free methods (in Backups).***

Continue...

from previous page…

Function Minimization

1. *Methods require gradient and/or Hessian martrix*

Newton-Raphson Method: If $f(\boldsymbol{x})$ is at least twice continuously differentiable in its domain, it can be approximated by its second order Taylor expansion $\hat{f}(\boldsymbol{x})$ at the starting point $\boldsymbol{x}_0$:

$$f(\boldsymbol{x_0}+\boldsymbol{h}) \approx \hat{f}(\boldsymbol{x}) = f(\boldsymbol{x_0}) + \boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_0}}\, \boldsymbol{h} + \frac{1}{2!}\boldsymbol{h}^{\mathrm{T}} \underbrace{\overbrace{\boldsymbol{\nabla}^2 f(\boldsymbol{x})\big|_{\boldsymbol{x_0}}}^{\boldsymbol{H}(\boldsymbol{x_0})}}_{\text{Hessian matrix}} \boldsymbol{h}$$

The step $\boldsymbol{h}$ is determined such that $\hat{f}(\boldsymbol{x})$ has a stationary point at $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{h}$:

$$\boldsymbol{\nabla}\hat{f}(\boldsymbol{x_1}) = \boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_0}} + \boldsymbol{h}^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x_0}) = 0 \quad \Rightarrow \quad \boldsymbol{h} = -\left[\boldsymbol{H}(\boldsymbol{x_0})\right]^{-1}\left[\boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_0}}\right]^{\mathrm{T}}$$

To ensure that the Wolfe conditions are satisfied, above relation is often relaxed to:

$$\boldsymbol{h} = -\eta\left|\boldsymbol{H}(\boldsymbol{x_0})\right|^{-1}\left[\boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_0}}\right]^{\mathrm{T}}, \qquad \text{with a learning parameter } \eta \in (0,1)$$

Inverting the Hessian matrix can be computationally expensive; in this case, $\boldsymbol{h}$ can be computed by finding an approximate solution to the linear system:

$$\boldsymbol{H}(\boldsymbol{x_0})\boldsymbol{h} = -\left[\boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_0}}\right]^{\mathrm{T}}$$

This procedure is iterated to produce a sequence of values according to:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta\left[\boldsymbol{H}(\boldsymbol{x_k})\right]^{-1}\left[\boldsymbol{\nabla} f(\boldsymbol{x})\big|_{\boldsymbol{x_k}}\right]^{\mathrm{T}}$$

Continue...

In practice, the iteration is stopped as soon as the norm of the gradient of $f(\boldsymbol{x})$ of falls below some predefined bound ε

from previous page…

Function Minimization

Descent Methods: As the computation of the Hessian matrix is computationally costly, various methods that do not require it have been devised, for instance, descent methods. A descent method is an iterative algorithm that searches for an approximate minimum of ***f(*x*)*** by decreasing the value of *f* in every iteration. The iteration has the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \eta_k \boldsymbol{d}_k, \quad \text{where } \boldsymbol{d}_k \text{ is the search direction and } \eta_k \text{ is the step-size parameter}$$

As with the Newton–Raphson method, when a (local) minimum is reached, it cannot be left anymore.

Line search: a search direction ***d*** is called a descent direction at the point

$$\boldsymbol{x} \in \mathbb{R}^n \text{ if } \boldsymbol{g}(\boldsymbol{x}) \cdot \boldsymbol{d} < 0, \text{ where } \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{\nabla} f(\boldsymbol{x}) \text{ is the gradient.}$$

$$\text{If } \eta \text{ is sufficiently small, then } f(\boldsymbol{x} + \eta \boldsymbol{d}) < f(\boldsymbol{x}).$$

$$\text{Once a search direction } \boldsymbol{d}_k \text{ has been choosen at the point}$$

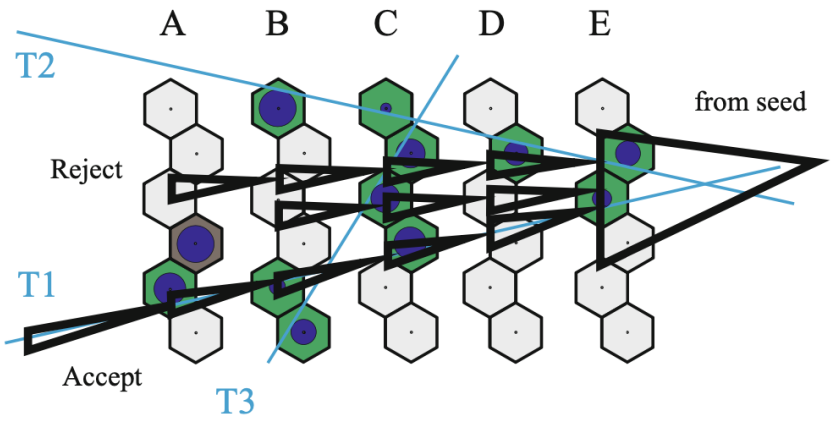
$$\boldsymbol{x}_k, \text{ line search implies that the line } \boldsymbol{x} = \boldsymbol{x}_k + \lambda \boldsymbol{d}_k$$

$$\text{is followed to its closest local minimum.}$$

Continue...

Track following by the Combinatorial Kalman Filter

- This method finds tracks locally or sequentially, one after the other.
- In track following, a track candidate starts from a “seed”, i.e., a short track segment. This seed can in principle be anywhere in the tracking detector. Generating seeds in the outer layers of the trackers has the advantage of smaller occupancy and less background from low-momentum tracks; generating seeds in the inner layers, which are frequently pixel layers, has the advantage of using 3D hits with higher resolution both in the bending plane anhits anddinal direction.
- The generation of seeds is often a simple combinatorial search for compatible triplets or quadruplets of hits, and includes information about the size and position of the beam spot.
- Once the seeds have been found, each seed is then followed through the tracker by extrapolating it toward the outside of the tracker or toward the production region, depending on where the seed is situated. After each extrapolation step, compatible hits are searched for and attached to the track candidate.
- The progressive track recognition described using the combinatorial Kalman filter (CKF). First, each seed is fitted. The parameters and the covariance matrix of the seed are then extrapolated to the nearest tracker layer, considering interactions with the detector material. The hits in the sensor in which the extrapolated trajectory intersects with the layer are tested for compatibility with the predicted track parameters using a chi-square statistic.
- If *n* compatible hits are found, *n* copies of the predicted state, i.e., its track parameters and its covariance matrix, are generated, and each one of them is updated with one of the *n* hits according to the Kalman filter. The original state is also kept and marked as having a missing hit, giving a total of *n* + 1 track candidates. This procedure is iterated on each track candidate until the last layer of the tracker is reached or the count of missing hits in a candidate exceeds a preset threshold, typically one or two.
- In the course of the combinatorial Kalman filter, it may be necessary to limit the number of active candidates for reasons of memory and/or speed. In this case, the “worst” track candidates are discarded and not followed anymore. The quality of a track candidate can be measured by a combination of its total number of hits, its number of missing hits, and its total chi-square χ^2.



Track Fitting

- Track fitting is an application of established statistical estimation procedures with well-known properties.
- Estimators based on the least-squares principle were the principal methods for track fitting. Robust and adaptive methods have been used.
- Will present a couple of methods:

  - **Least-squares regression,**
  - **the extended Kalman filter**

Least-squares regression

- Assume that track finding has produced a track candidate, i.e., a collection of n measurements $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n$ in different layers of the tracking detector, along with their respective covariance matrices $\boldsymbol{V}_1, \ldots, V_n$. The measurements may have different dimensions m_i and usually have different covariance matrices. The initial parameters of the track to be fitted to the measurements are denoted by $\boldsymbol{p}$. They are assumed to be tied to a reference surface (layer 0). The regression model has the following form:

$$\boldsymbol{m} = \boldsymbol{q}\boldsymbol{f}(\boldsymbol{p}) + \epsilon, \quad \mathrm{E}[\epsilon] = \mathbf{0}, \quad \mathrm{Var}[\epsilon] = \boldsymbol{V}$$

where $\boldsymbol{m} = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_n)^{\mathsf{T}}$ and $\boldsymbol{f} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n)^{\mathsf{T}}$. The function $\boldsymbol{f}_k$ maps the initial parameters $\boldsymbol{p}$ to the measurement $\boldsymbol{m}_k$ in layer k. It is the composition of the track propagators up to layer k and the function that maps the track state to the measurement:

$$\boldsymbol{f}_k = \boldsymbol{h}_k \circ \boldsymbol{f}_{k|k-1} \circ \boldsymbol{f}_{k-1|k-2} \circ \cdots \circ \boldsymbol{f}_{2|1} \circ \boldsymbol{f}_{1|0}$$

- Its Jacobian $\boldsymbol{F}_k$ is given by the product of the respective Jacobians:

$$\boldsymbol{F}_k = \boldsymbol{H}_k \boldsymbol{F}_{k|k-1} \boldsymbol{F}_{k-1|k-2} \cdots \boldsymbol{F}_{1|0}$$

- The covariance matrix V is the sum of two parts, $\boldsymbol{V} = \boldsymbol{V}_M + \boldsymbol{V}_S$. The first part is the joint covariance matrix of all measurement errors. These can virtually always be assumed to be uncorrelated across different layers, so that VM is block-diagonal:

$$\boldsymbol{V}_M = \mathrm{blkdiag}\left(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n\right), \quad \boldsymbol{V}_i = \mathrm{Var}[\boldsymbol{\epsilon}_i], \; i = 1, \ldots, n$$

Continue...

from previous page…

Least-squares regression

- The second part $\boldsymbol{V}_S$ is the joint covariance matrix of the process noise caused by material effects, mainly multiple Coulomb scattering. The process noise encountered during the propagation from layer $k-1$ to layer k is denoted by $\boldsymbol{\gamma}_k$ and its covariance matrix by $\boldsymbol{Q}_k$. The integrated process noise up to layer k is denoted by $\boldsymbol{\Gamma}_k$. Linearized error propagation along the track gives the Linearized error propagation along the track gives the following expression for the covariance matrix of $\boldsymbol{\Gamma}_k$:

$$\mathrm{Var}\left[\boldsymbol{\Gamma}_k\right] = \sum_{i=1}^{k} \boldsymbol{F}_{k|i} \boldsymbol{Q}_i \boldsymbol{F}_{k|i}^{\mathrm{T}}$$

with $$\boldsymbol{F}_{k|i} = \boldsymbol{F}_{k|k-1} \boldsymbol{F}_{k-1|k-2} \ldots \boldsymbol{F}_{i+1|i} \quad \text{for } i < k \quad \text{and} \quad \boldsymbol{F}_{k|k} = \boldsymbol{I}$$

If $i < k$, $\boldsymbol{\Gamma}_i$ and $\boldsymbol{\Gamma}_k$ are correlated with the following cross-covariance matrix:

$$\mathrm{Cov}\left[\boldsymbol{\Gamma}_i, \boldsymbol{\Gamma}_k\right] = \sum_{j=1}^{i} \boldsymbol{F}_{i|j} \boldsymbol{Q}_j \boldsymbol{F}_{k|j}^{\mathrm{T}}$$

- Error propagation from the track states to the measurements gives the final block structure of $\boldsymbol{V}_S$:

Continue...

Function Minimization

2. gradient-free methods

A popular gradient-free method is the downhill-simplex or Nelder–Mead algorithm. It can be applied to functions whose derivatives are unknown, do not exist everywhere or are too costly or difficult to compute. In n dimensions, the method stores n+1 test points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1}$ at every iteration, ordered by increasing function values, and the centroid $\boldsymbol{x}_0$ of all points but the last one. The simplex generated by the test points is then modified according to the function values in the test points. The allowed modifications are reflection, expansion, contraction and shrinking. The iteration is terminated when the function value of the best point does not change significantly anymore. The size of the initial simplex is important; choosing it too small can lead to a very localized search. On the other hand, it is possible to escape from a local minimum by restarting the search with a sufficiently large simplex.

Statistical Models and the Estimation of Model Parameters

- A statistical model is defined as a functional dependence of observed quantities (observations or measurements) on unknown quantities of interest (parameters or state vectors). The parameters cannot be observed directly, and the observations are subject to stochastic uncertainties. The aim is to estimate the parameters from the observations according to some criterion of optimality. There are three Models:

❑ ***Linear Regression Model***

❑ ***Nonlinear Regression Models***

❑ ***State Space Models***

Linear Regression Model

- A linear regression model has the following general form

$$\boldsymbol{m} = \boldsymbol{F}\boldsymbol{p} + \boldsymbol{c} + \boldsymbol{\epsilon}, \quad \mathrm{E}[\boldsymbol{\epsilon}] = 0, \quad \mathrm{Var}[\boldsymbol{\epsilon}] = \boldsymbol{V} = \boldsymbol{G}^{-1}$$

where ***m*** is the ($n \times 1$)-vector of observations, ***F*** is the known ($n \times m$) model matrix with $m \leq n$ and assumed to be of full rank, ***p*** is the ($m \times 1$) vector of model parameters, ***c*** is a known constant offset, and ***ε*** is the ($n \times 1$) vector of observation errors with zero expectation and ($n \times n$) covariance matrix ***V*** , assumed to be known.

- Least square (LS) estimation of ***p*** requires the minimization of the following objective function:

$$S(\boldsymbol{p}) = (\boldsymbol{m} - \boldsymbol{F}\boldsymbol{p} - \boldsymbol{c})^{\mathrm{T}} \boldsymbol{G} (\boldsymbol{m} - \boldsymbol{F}\boldsymbol{p} - \boldsymbol{c})$$

The least-squares estimator ***p*** and its covariance matrix ***C*** are given by:

$$\hat{\boldsymbol{p}} = \left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{F}\right)^{-1} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{m} - \boldsymbol{c}), \quad \boldsymbol{C} = \left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{F}\right)^{-1}$$

- The estimator ***p*** is unbiased and the estimator with the smallest covariance matrix among all estimators that are linear functions of the observations. If the distribution of ε is a multivariate normal distribution, the estimator is efficient, i.e., has the smallest possible covariance matrix among all unbiased estimators.

Continue...

from previous page…

Linear Regression Model

- The residuals ***r*** of the regression are defined by:

$$\boldsymbol{r} = \boldsymbol{m} - \boldsymbol{c} - \boldsymbol{F}\hat{\boldsymbol{p}}, \quad \boldsymbol{R} = \mathrm{Var}[\boldsymbol{r}] = \boldsymbol{V} - \boldsymbol{F}\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{F}\right)^{-1}\boldsymbol{F}^{\mathrm{T}}$$

- The standardized residuals s, also called the “pulls” in high-energy physics, are given by:

$$s_i = \frac{r_i}{\sqrt{\boldsymbol{R}_{ii}}}, \quad i = 1, \ldots n$$

- If the model is correctly specified, the pulls have mean 0 and standard deviation 1. The chi-square statistic of the regression is defined as:

$$\chi^2 = \boldsymbol{r}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{r}, \quad \text{with} \quad \mathrm{E}\left[\chi^2\right] = n - m$$

- If the observation errors are normally distributed, $\chi 2$ is $\chi 2$-distributed with $d = n - m$ degrees of freedom; its expectation is d and its variance is $2d$. Its p-value p is defined by the following probability transform:

$$p = 1 - G_d(\chi^2) = \int_{\chi^2}^{+\infty} g_d(x)\mathrm{d}x$$

where $G_k(x)$ is the cumulative distribution function of the $\chi 2$-distribution with k degrees of freedom and $g_k(x)$ is its probability density function. Large values of $\chi 2$ correspond to small p-values. If the model is correctly specified, p is uniformly distributed in the unit interval. A very small p-value indicates a misspecification of the model or of the covariance matrix ***V*** , or both.

Nonlinear Regression Models

- The linear regression model can be generalized to a nonlinear model:

$$\boldsymbol{m} = \boldsymbol{f}(\boldsymbol{p}) + \boldsymbol{\epsilon}, \quad \mathrm{E}[\boldsymbol{\epsilon}] = 0, \quad \mathrm{Var}[\boldsymbol{\epsilon}] = \boldsymbol{V} = \boldsymbol{G}^{-1}$$

where ***f*** is a (n × 1)-vector of smooth functions of m variables. LS estimation of ***p*** requires the minimization of the following objective function:

$$S(\boldsymbol{p}) = [\boldsymbol{m} - \boldsymbol{f}(\boldsymbol{p})]^{\mathrm{T}} \boldsymbol{G} [\boldsymbol{m} - \boldsymbol{f}(\boldsymbol{p})]$$

- The function S(***p***) can be minimized with the Gauss-Newton method, based on the first-order Taylor expansion of ***f*** and resulting in the following iteration:

$$\hat{\boldsymbol{p}}_{k+1} = \hat{\boldsymbol{p}}_k + \left(\boldsymbol{F}_k^{\mathrm{T}} \boldsymbol{G} \boldsymbol{F}_k \right)^{-1} \boldsymbol{F}_k^{\mathrm{T}} \boldsymbol{G} [\boldsymbol{m} - \boldsymbol{f}(\hat{\boldsymbol{p}}_k)], \quad \boldsymbol{F}_k = \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} \right|_{\hat{\boldsymbol{p}}_k}$$

- At each step, the covariance matrix $\boldsymbol{C}_{k+1}$ of $\hat{\boldsymbol{p}}_{k+1}$ is approximately given by:

$$\boldsymbol{C}_{k+1} = \left(\boldsymbol{F}_k^{\mathrm{T}} \boldsymbol{G} \boldsymbol{F}_k \right)^{-1}$$

- In general, the covariance matrix of the final estimate ***p*** can be approximated by the inverse of the Hessian of S(***p***) at ***p***^. The final chi-square statistic $\chi 2$ is given by:

$$\chi^2 = [\boldsymbol{m} - \boldsymbol{f}(\hat{\boldsymbol{p}})]^{\mathrm{T}} \boldsymbol{G} [\boldsymbol{m} - \boldsymbol{f}(\hat{\boldsymbol{p}})]$$

- In the case of Gaussian observation errors, the chi-square statistic is approximately $\chi 2$-distributed, and its p-value is approximately uniformly distributed. The iterationis stopped when the chi-square statistic does not change significantly anymore.

State Space Models

- A dynamic or state space model describes the state of an object in space or time, such as a rocket or a charged particle.
- The state usually changes continuously but is assumed to be of interest only at discrete instances in the present context. These instances are labelled with indices from 0, the initial state, to *n*, the final state.
- The state at instance *k* is specified by the state vector $\boldsymbol{q}_k$. The spatial or temporal evolution of the state is described by the system equation, which is

$$\boldsymbol{q}_k = \boldsymbol{F}_{k|k-1}\boldsymbol{q}_{k-1} + \boldsymbol{d}_k + \boldsymbol{\gamma}_k, \quad \mathrm{E}[\boldsymbol{\gamma}_k] = \boldsymbol{g}_k, \quad \mathrm{Var}[\boldsymbol{\gamma}_k] = \boldsymbol{Q}_k$$

$$\boldsymbol{\gamma}_k = \text{process noise}, \ \boldsymbol{Q}_k = \text{process noise covariance matrix}$$

in the linear state space models case & based on Kalman filter, and

$$\boldsymbol{q}_k = \boldsymbol{f}_{k|k-1}(\boldsymbol{q}_{k-1}) + \boldsymbol{\gamma}_k, \quad \mathrm{E}[\boldsymbol{\gamma}_k] = \boldsymbol{g}_k, \quad \mathrm{Var}[\boldsymbol{\gamma}_k] = \boldsymbol{Q}_k$$

in the nonlinear state space models & based on Kalman filter case.

3. *Karimaki's Method*:

Karimäki's approach starts from the simplified expression of the residuals d_i introduced by Chernov and Ososkov and considers a χ^2 with weighted residuals:

$$\chi^2 = \sum_{i=1}^{n} w_i d_i^2$$

The weights can for instance contain measurement uncertainties if they are not the same for all measurements (x_i, y_i). The χ^2 function is minimized with respect to a set of circle parameters with Gaussian behaviour: the curvature $\kappa = 1/R$, the impact parameter ϵ (the distance from the origin to the point of closest approach of the fitted circle), and the direction φ of the tangent of the circle at the point of closest approach. Using this set of parameters, the simplified residuals are expressed as:

$$d_i = \frac{1}{2} k r_i^2 - (1 + k\epsilon) r_i \sin(\phi + \phi_i) + \frac{1}{2} k \epsilon^2 + \epsilon$$

where r_i and φ_i are the polar coordinates of measurement i. The residuals can be written as:

$$d_i = (1 + k\epsilon)\eta_i, \quad \eta_i = \gamma r_i^2 - r_i \sin(\phi - \phi_i) + \delta$$

$$\gamma = \frac{k}{2(1 + k\epsilon)}, \qquad \delta = \frac{(1 + k\epsilon)/2}{(1 + k\epsilon)} \epsilon$$

from previous page…

***Karimaki's Method*:**

Using these definitions, the χ^2 can be written as:

$$\chi^2 = (1 + k\epsilon)^2 \hat{\chi}^2, \quad \hat{\chi}^2 = \sum_i w_i \eta_i^2$$

with the approximation $1 + \kappa\epsilon \approx 1$, $\hat{\chi}^2$ can be minimized instead of χ^2, leading to a set of equations with solutions:

$$\phi = \frac{1}{2} \arctan(2q_1/q_2)$$

$$\gamma = (\sin\phi X_{xz} - \cos\phi C_{yz}) / C_{zz}$$

$$\delta = -\gamma < z > + \sin\phi < x > - \cos\phi < y >$$

$$q_1 = C_{zz} C_{xy} - C_{xz} C_{yz}$$

$$q_2 = C_{zz}(C_{xx} - C_{yy}) - C_{xz}^2 + C_{yz}^2$$

$$< x >= \sum_i w_i x_i / \sum_i w_i \quad \ldots$$

The variances and covariances of the measurements x, y, $z=x^2+y^2$ are given by:

$$C_{xx} = < x^2 > - < x >^2$$

$$C_{xy} = < xy > - < x >< y >$$

$$C_{yy} = < y^2 > - < y >^2$$

$$C_{xz} = < xz > - < x >< z >$$

$$C_{yz} = < yz > - < y >< z >$$

$$C_{zz} = < z^2 > - < z >^2$$

The curvature k and impact parameter ϵ are given by: $k = \frac{2\gamma}{\sqrt{1 - 4\delta\gamma}}, \; \epsilon = \frac{2\delta}{1 + \sqrt{1 - 4\delta\gamma}}$

Continue...

from previous page…

***Karimaki's Method*:**

Using these definitions, the χ^2 can be written as:

$$\chi^2 = (1 + k\epsilon)^2 \hat{\chi}^2, \quad \hat{\chi}^2 = \sum_i w_i \eta_i^2$$

with the approximation $1 + \kappa\epsilon \approx 1$, $\hat{\chi}^2$ can be minimized instead of χ^2, leading to a set of equations with solutions:

$$\phi = \frac{1}{2} \arctan(2q_1/q_2)$$

$$\gamma = \left(\sin\phi C_{xz} - \cos\phi C_{yz}\right)/C_{zz}$$

$$\delta = -\gamma < z > + \sin\phi < x > - \cos\phi < y >$$

$$q_1 = C_{zz}C_{xy} - C_{xz}C_{yz}$$

$$q_2 = C_{zz}(C_{xx} - C_{yy}) - C_{xz}^2 + C_{yz}^2$$

$$< x >= \sum_i w_i x_i / \sum_i w_i \quad \ldots$$

The variances and covariances of the measurements x, y, $z=x^2+y^2$ are given by:

$$C_{xx} = < x^2 > - < x >^2$$

$$C_{xy} = < xy > - < x >< y >$$

$$C_{yy} = < y^2 > - < y >^2$$

$$C_{xz} = < xz > - < x >< z >$$

$$C_{yz} = < yz > - < y >< z >$$

$$C_{zz} = < z^2 > - < z >^2$$

The curvature k and impact parameter ϵ are given by: $k = \frac{2\gamma}{\sqrt{1 - 4\delta\gamma}}, \ \epsilon = \frac{2\delta}{1 + \sqrt{1 - 4\delta\gamma}}$

from previous page…

Kalman Filter

- **Tracking method in 2-D using Kalman filter**

1. Define a track with slope *b* and intercept *a* where are unknown, track model $y = ax + b$; x plays the role of the time variable.
2. State vector defined as $\boldsymbol{s} = (b,a)$; $\boldsymbol{b}_k$ is the y-position of the track at detector position x_k.
3. No process noise $\boldsymbol{W} = 0$, no control-input matrix $\boldsymbol{B} = 0$.
4. State update $a \rightarrow a$, $b \rightarrow b + a\,\Delta x$ with $\Delta x = x_k - x_{k-1}$

$$\boldsymbol{F}_k = \begin{pmatrix} 1 & \Delta x_k \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{s}_k = \boldsymbol{F}_k \boldsymbol{s}_{k-1}, \quad \boldsymbol{S}_k = \boldsymbol{F}_k \boldsymbol{S}_{k-1} \boldsymbol{F}_k^{\mathrm{T}}$$

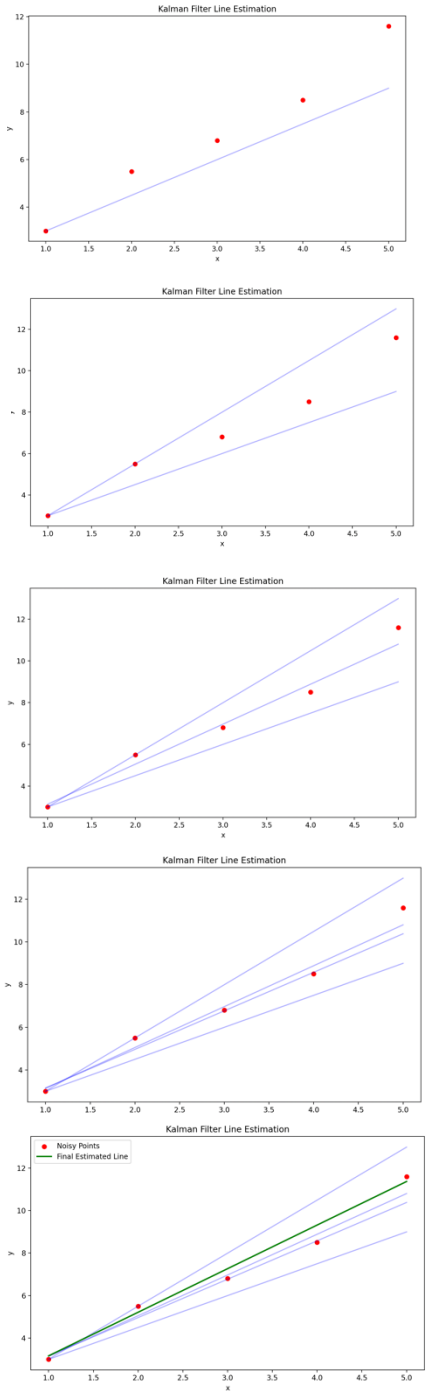
5. For the initial guess make a χ^2 fit through the first two points (the track seed).
6. Result is the parameter for $x = 0$, can then propagate to second point.
7. First step of Kalman filter is then to propagate to the third point – the first new measurement.
8. Measurement is the y – position. $\boldsymbol{H}_k\boldsymbol{s}_k$ gives the expected measurement.
9. For $s = (b,a)$, thus $H_k = (1\ 0)$ projects out of position.
10. Define Kalman gain:

$$\boldsymbol{K}_k = \boldsymbol{S}_k \boldsymbol{H}_k^{\mathrm{T}} \left(\boldsymbol{R}_k + \boldsymbol{H}_k \boldsymbol{S}_k \boldsymbol{H}_k^{\mathrm{T}}\right)^{-1}$$

$$\boldsymbol{K}_k = \left(\boldsymbol{S}_{b,b} \boldsymbol{S}_{b,a}\right) / \left(\sigma^2_{\mathrm{meas},k} + \boldsymbol{S}_{b,b}\right)$$

$$\boldsymbol{s}_k = \boldsymbol{s}_{k-1} + \boldsymbol{K}_k \left(y_{\mathrm{meas},k} - b_{k-1}\right)$$

$$\boldsymbol{S}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{S}_{k-1}$$



Continue...

Application: Lorentz angle in micromegas

- At the steady state, the general expression of the drift velocity $\boldsymbol{u}_D = \langle \boldsymbol{u} \rangle$ can be derived by the equation of motion of an electron under the influence of electric and magnetic fields and a friction due to multiple scattering of the drift electron with the gas:

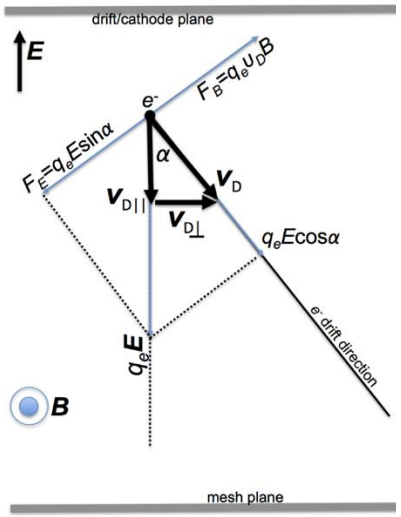
$$m\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -e(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) - m\frac{\boldsymbol{v}}{\tau}$$

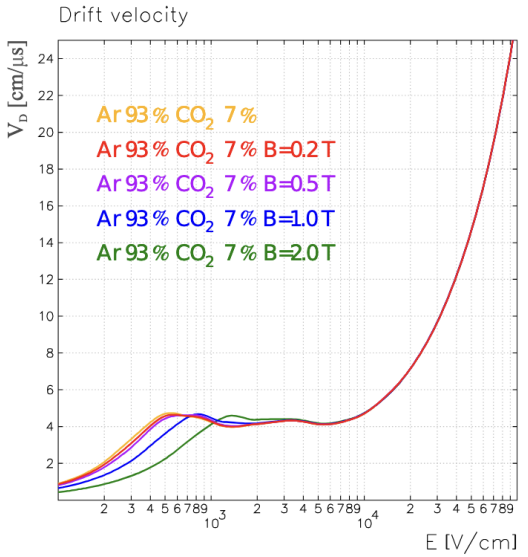
$$v_{D_\parallel} = \frac{\mu E}{1+(\omega\tau)^2}\left[1 + (\omega\tau)^2\left(\hat{\boldsymbol{E}}\cdot\hat{\boldsymbol{B}}\right)^2\right]$$

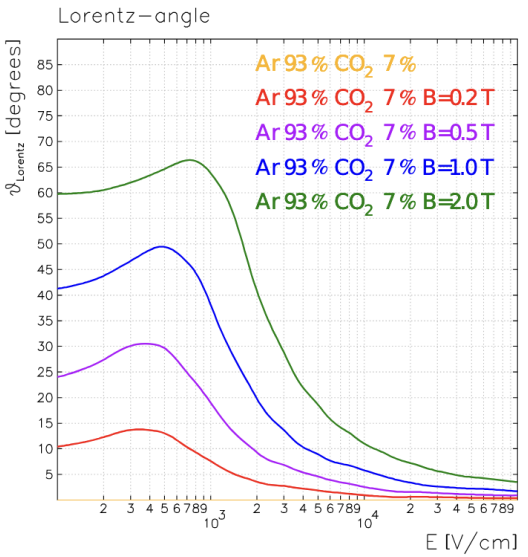
$$v_{D_\perp} = \mu E(\omega\tau)\frac{\left|\hat{\boldsymbol{E}}\times\hat{\boldsymbol{B}}\right|\sqrt{1+(\omega\tau)^2\left(\hat{\boldsymbol{E}}\cdot\hat{\boldsymbol{B}}\right)^2}}{1+(\omega\tau)^2}$$

$$\tan\alpha = \frac{v_{D_\perp}}{v_{D_\parallel}} = \omega\tau\frac{\left|\hat{\boldsymbol{E}}\times\hat{\boldsymbol{B}}\right|}{\sqrt{1+(\omega\tau)^2\left(\hat{\boldsymbol{E}}\cdot\hat{\boldsymbol{B}}\right)^2}}$$

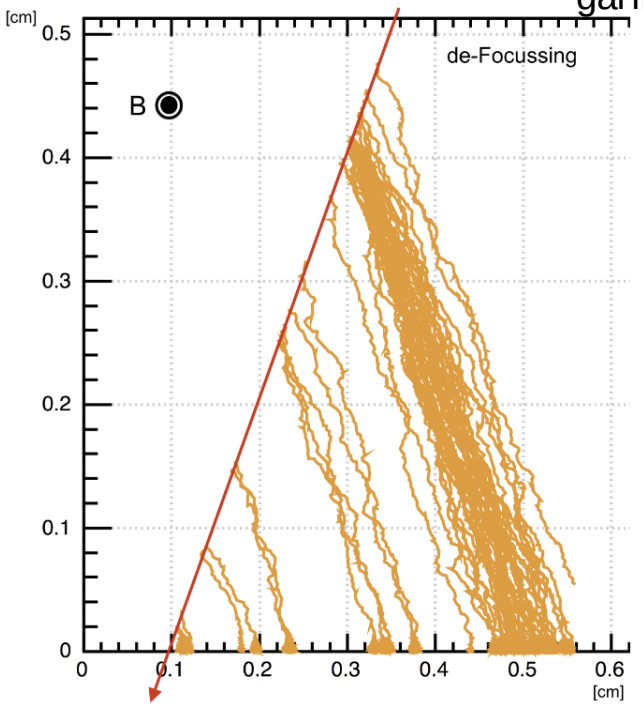
mobility: $\mu = \frac{e}{m}\tau$, cyclotron frequency: $\omega = \frac{e}{m}B$

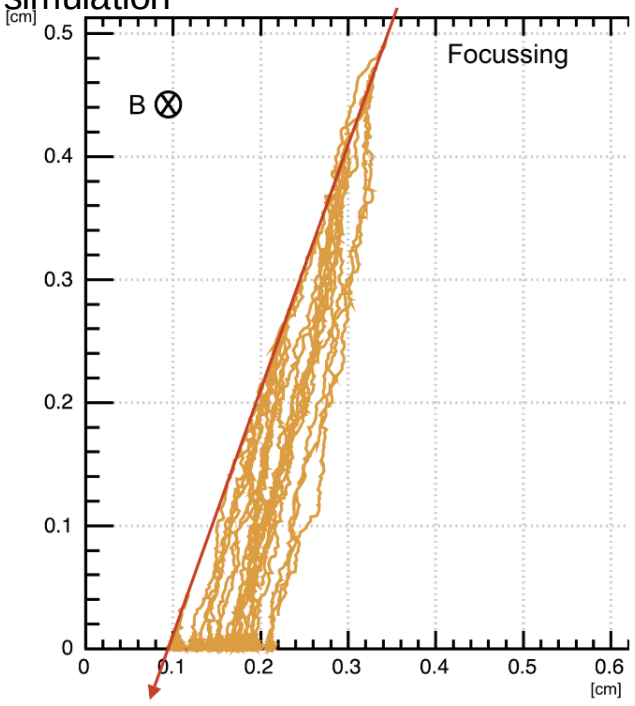






garfield simulation





from previous page…

Kalman Filter

- **Problem definition**

Kalman filters are used to estimate states based on linear dynamical systems in state space format. The process model defines the evolution of the state from time $k-1$ to time k as:

$$\boldsymbol{s}_k = \boldsymbol{F}\boldsymbol{s}_{k-1} + \boldsymbol{B}\boldsymbol{u}_{k-1} + \boldsymbol{w}_{k-1}$$

where $\boldsymbol{F}$ is the state transition matrix applied to the previous state vector s_{k-1}, $\boldsymbol{B}$ is the control-input matrix applied to the control vector $\boldsymbol{u}_{k-1}$, and $\boldsymbol{w}_{k-1}$ is the process noise vector that is assumed to be zero-mean Gaussian with the covariance $\boldsymbol{W}$, i.e., $\boldsymbol{w}_{k-1} \sim \mathcal{N}(0,\boldsymbol{W})$. The process model is paired with the measurement model that describes the relationship between the state and the measurement at the current time step k as:

$$\boldsymbol{m}_k = \boldsymbol{H}\boldsymbol{s}_k + \boldsymbol{v}_k$$

where $\boldsymbol{m}_k$ is the measurement vector, $\boldsymbol{H}$ is the measurement matrix, and $\boldsymbol{u}_k$ is the measurement noise vector that is assumed to be zero-mean Gaussian with the covariance $\boldsymbol{R}$, i.e., $\boldsymbol{u}_k \sim \mathcal{N}(0,\boldsymbol{R})$.

The role of the Kalman filter is to provide estimate of $\boldsymbol{s}_k$ at time k, given the initial estimate of $\boldsymbol{s}_0$, the series of measurement, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$,…,$\boldsymbol{m}_k$, and the information of the system described by ***F, B, H*** F, W, and $\boldsymbol{R}$.

Continue...

from previous page…

Kalman Filter

- **Kalman filter algorithm**

Kalman filter algorithm consists of two stages: prediction and update. Note that the terms 'prediction' and 'update' are often called 'propagation' and 'correction', respectively. The Kalman filter algorithm is summarized as follows:

Prediction:

$$\text{Predicted state estimate} \qquad \hat{\boldsymbol{s}}_k^- = \boldsymbol{F}\hat{\boldsymbol{s}}_{k-1}^+ + \boldsymbol{B}\boldsymbol{u}_{k-1}$$

$$\text{Predicted error covariance} \qquad \boldsymbol{S}_k^- = \boldsymbol{F}\boldsymbol{S}_{k-1}^+\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{W}$$

Update:

$$\text{Measurement residual} \qquad \tilde{\boldsymbol{y}}_k = \boldsymbol{m}_k - \boldsymbol{H}\hat{\boldsymbol{s}}_k^-$$

$$\text{Kalman gain} \qquad \boldsymbol{K}_k = \boldsymbol{S}_k^-\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{S}_k^-\boldsymbol{H}^{\mathrm{T}}\right)^{-1}$$

$$\text{Update estimate} \qquad \hat{\boldsymbol{s}}_k^+ = \hat{\boldsymbol{s}}_k^- + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k$$

$$\text{Update error covariance} \qquad \boldsymbol{S}_k^+ = (1 - \boldsymbol{K}_k\boldsymbol{H})\,\boldsymbol{S}_k^-$$

Kalman filters are derived based on the assumption that the process and measurement models are linear, i.e., they can be expressed with the matrices ***F, B,*** and ***H***, and the process and measurement noise are additive Gaussian.

In the above equations, the hat operator, ˆ, means an estimate of a variable. That is, $\boldsymbol{x}$^, is an estimate of $\boldsymbol{x}$. The superscripts – and + denote predicted (prior) and updated (posterior) estimates, respectively. The predicted state estimate is evolved from the updated previous updated state estimate. The term ***S*** is called state error covariance.

Continue...

from previous page…

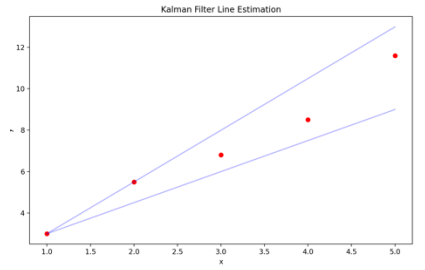
Kalman Filter

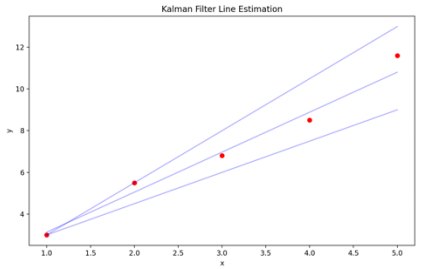
- **Tracking method in 2-D using Kalman filter**

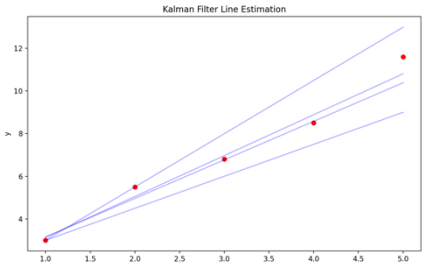
1. Define a track with slope *b* and intercept *a* where are unknown, track model *y* = *ax* + *b*; *x* plays the role of the time variable.
2. State vector defined as $\boldsymbol{s}$ = (*b*,*a*); $\boldsymbol{b}_k$ is the *y*-position of the track at detector position x_k.
3. No process noise $\boldsymbol{W}$ = 0, no control-input matrix $\boldsymbol{B}$ = 0.
4. State update *a* → *a*, *b* → *b* + *a* Δ*x* with Δ*x* = x_k - x_{k-1}

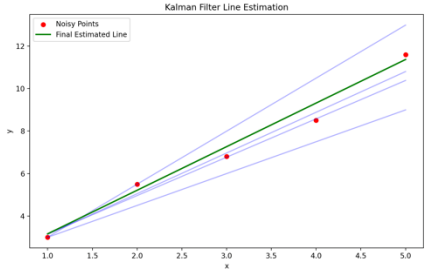
$$\boldsymbol{F}_k = \begin{pmatrix} 1 & \Delta x_k \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{s}_k = \boldsymbol{F}_k \boldsymbol{s}_{k-1}, \quad \boldsymbol{S}_k = \boldsymbol{F}_k \boldsymbol{S}_{k-1} \boldsymbol{F}_k^{\mathrm{T}}$$

5. For the initial guess make a χ^2 fit through the first two points (the track seed).
6. Result is the parameter for *x* = 0, can then propagate to second point.
7. First step of Kalman filter is then to propagate to the third point – the first new measurement.
8. Measurement is the y – position. $\boldsymbol{H}_k\boldsymbol{s}_k$ gives the expected measurement.
9. For *s* = (*b*,*a*), thus H_k = (1 0) projects out of position.









Continue...