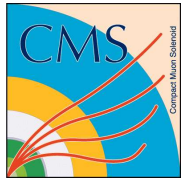
Plans for the implementation of a *Global Track Fitter* in CMS

Daniele Trocino

Università di Torino, INFN Torino

CMS Torino weekly meeting

16 October, 2009



Outline

- Track fitting in CMS
  - Kalman Filter
  - Muon reconstruction: description and performance
  - strengths and weaknesses
- Need for a Global Fitter
  - concepts, working and benefits
  - applications and limits
- Future plans
  - what we have, what we need

Tracking in CMS

In a magnetic field, the trajectory is a helix → **5 parameters**:

x = (charge/momentum, position and direction on a given surface)

Reconstruction of the trajectory of charged particles from *position measurements*

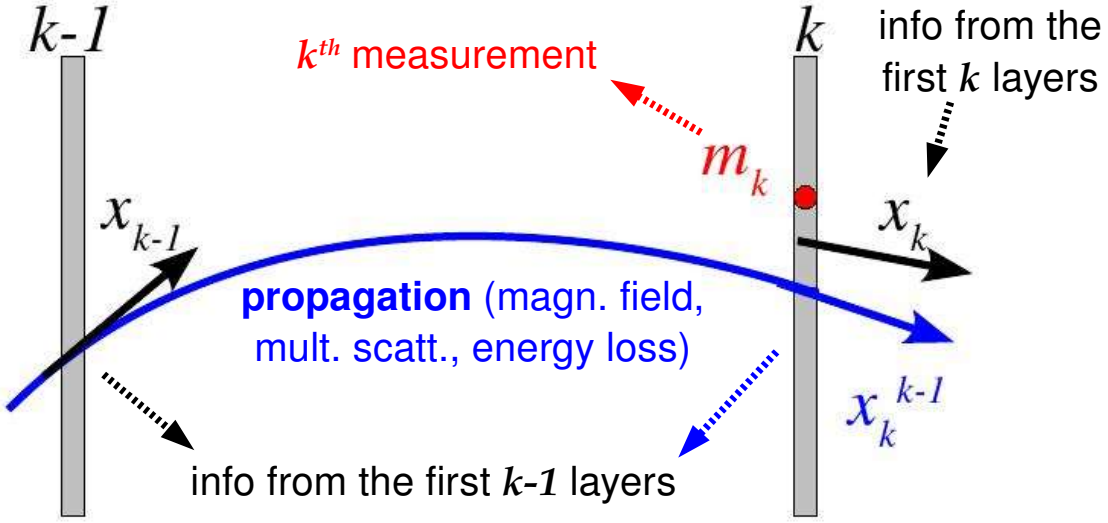
Requirements:

- account for *multiple scattering* and *energy loss*
- used both in Tracker (1/2-dim hits) and Muon System (1/2/3-dim hits/segments)
- it must provide
  - the pattern recognition → collection of hits
  - the best estimation of the track → minimum χ^2
- ... possibly in the fastest way! → well suited for HLT

→ **Kalman Filter**

The Kalman Filter (I)

Based on three basic steps: *prediction* (*propagation*), *filtering*, *smoothing*



Measurement on k^{th} layer:

$$m_k = H_k \, x_k^{true} + \epsilon_k$$

m_k: 1/2/3 dim; $H_k \, x_k^{true}$: 5 dim; ϵ_k: "noise"

Predicted *state* and *covariance matrix*:

$$x_k^{k-1} = F_{k-1} \, x_{k-1}$$

$$C_k^{k-1} = F_{k-1} \, C_{k-1} \, F_{k-1} + M_{k-1}$$

F_{k-1}: magn. field, energy loss

M_{k-1}: multiple scattering, straggling

Filtered *state* and *covariance matrix*:

$$x_k = x_k^{k-1} + K_k \, (\, m_k - H_k \, x_k^{k-1} \,)$$

$$C_k = (\, 1 - K_k \, H_k \,) \, C_k^{k-1}$$

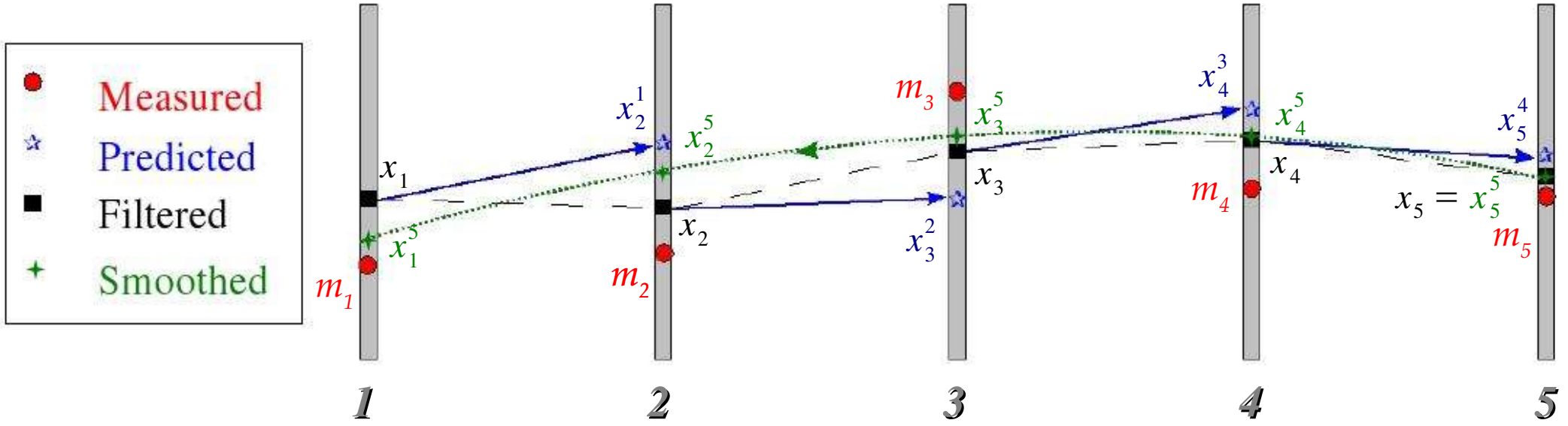
K_k: Kalman gain matrix

The Kalman Filter (II)

On the k^{th} layer, the *filtered state* x_k contains information only from the first k hits

Once all the n hits have been collected, the ***smoothing*** is performed

→ each x_k is updated with the information from the last $n - k$ layers



Smoothed *state* and *covariance matrix*:

$$x_k = x_k^{k-1} + A_k \left(x_{k+1}^n - x_{k+1}^k \right)$$

smoother gain matrix

$$C_k^n = C_k + A_k \left(C_{k+1}^n - C_{k+1}^k \right) A_k^T$$

The *smoothed* trajectory represents the best estimate (minimum χ^2) for the given set of hits

The Propagators

Muon transport in the detector accounting for *magnetic field* and *material effects*

Magnetic field, *mean* energy loss (Bethe-Block) → **state vector**

Multiple scattering, energy loss *fluctuations* → **covariance matrix**

1. `Analytic`
(used in the Tracker)

assumes a *uniform magnetic field* (perfect helix)

The state vector is propagated from layer to layer, material effects are introduced in the end-points

2. `Runge-Kutta`
(used in the Tracker)

4th order Runge-Kutta method → *non-uniform magnetic field (radial component)*

The state vector is propagated from layer to layer, material effects are introduced in the end-points

3. `Stepping-Helix`
(mainly outside the Tracker)

2nd order Runge-Kutta method → *uniform magnetic field*

The CMS volume is mapped in *cells* with uniform magnetic field

→ inside each cell the trajectory is a helix
→ at each step, the magnetic field is updated and the material effects are introduced

Stand-alone reconstruction

The **track reconstruction** in the **Muon System** takes place in four steps:

1. Estimation of the initial state (*seed*)
   - *on-line*: input from Level 1 trigger
   - *off-line*: built from one or more track segments
     p_T parametrized as a function of ϕ (or $\Delta\phi$) of segments: $\boldsymbol{p_T = A - B/\Delta\phi}$

2. Pre-filter or forward filter (inside-out)
   - starts from the *seed state* (extrapolated at the *innermost* layer)
   - segments (1D hits for RPC) are used for *pattern recognition*, on a χ^2 basis
   - segments (1D hits for RPC) are used for *update* (*filtering*) of the trajectory
   - needed to avoid possible *biases* from the seed

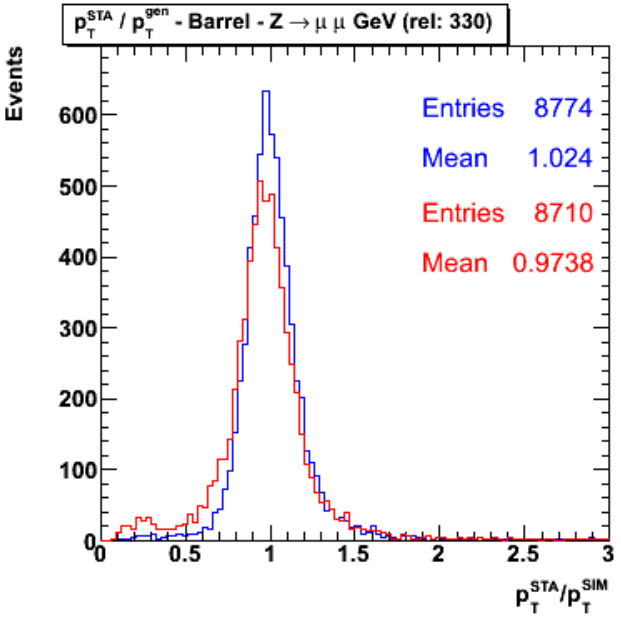
No smoothing!

3. Filter or backward filter (outside-in)
   - starts from the *outermost* pre-filtered state
   - segments (1D hits for RPC) are used for *pattern recognition*, on a χ^2 basis
   - single hits (1D for DT/RPC, 2D for CSC) are used for the *update*

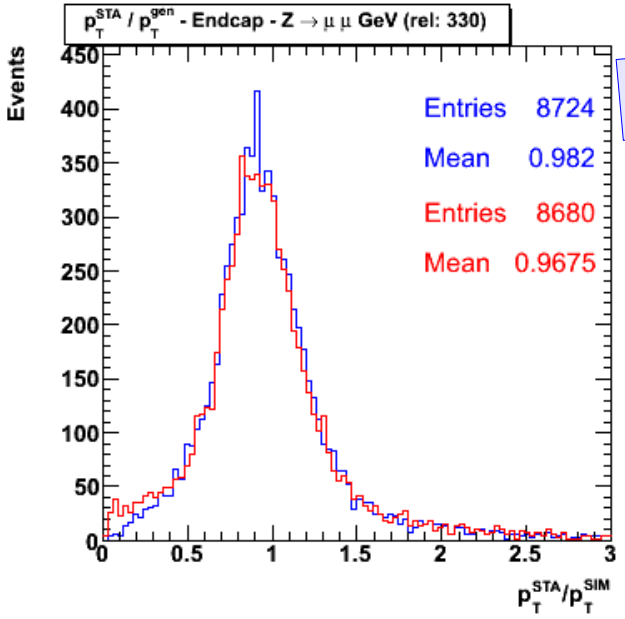
4. Vertex constraint
   After the ghost suppression, the trajectory is extrapolated to the *point of closest approach* to the *beam line* and the *beam spot* is constrained to be a point of the track, to improve the p_T resolution

Fit in stand-alone reconstruction

Barrel

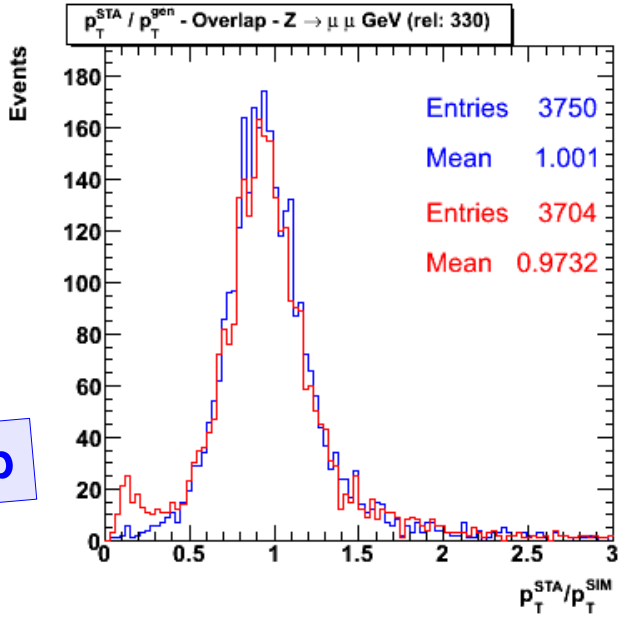


Endcap

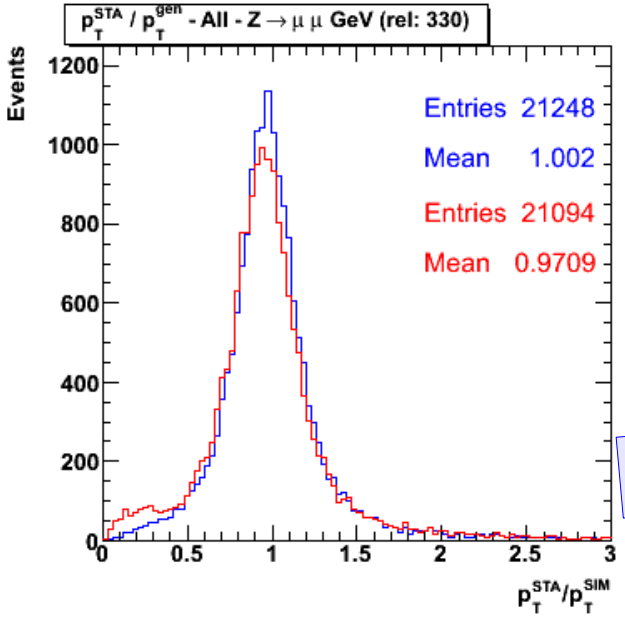


CMSSW_3_3_0

Overlap



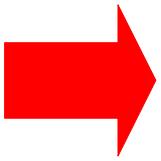
Total



— standard STA
— STA + smoothing

Weaknesses of Kalman Filter

- Crucial point: initialization of the covariance matrix
  - too large initial errors → too sensitive to possible low quality hits
  - too small initial errors → bias
- recursive method proves more unstable
  a single bad hit can irreparably damage the fit



efficiency loss

worsening of p_T estimation (in some cases)

Besides (not directly related to KF)

- calorimeters information is not exploited in the estimation of the energy loss (only simulated)

Global χ^2 Fit (I)

Non iterative method: all the collected measuremets are fitted *simultaneously*

Function to be minimized:

$$\chi^2 = \sum_{meas} \frac{r^2_{meas}}{\sigma^2_{meas}} + \sum_{scatt} \left(\frac{\theta^2_{scatt}}{\sigma^2_{scatt}} + \frac{\sin^2\theta_{loc}\ \phi^2_{scatt}}{\sigma^2_{scatt}} \right) + \sum_{E_{loss}} \frac{(\Delta E - \overline{\Delta E})^2}{\sigma^2_{E_{loss}}}$$

residuals — *multiple scattering* — *energy loss*

Fit parameters:

- *track parameters at vertex:* $\boldsymbol{\alpha} = (q/p_{T,0},\ D_0,\ z_0,\ \phi_0,\ \lambda_0 = \mathrm{cotg}\,\theta_0)$
- *scattering angles* (ϕ_{scatt}, θ_{scatt}) for each layer
- *energy loss* ΔE ($\overline{\Delta E}$ from *material description* or *measured* in calorimeters)

Global χ^2 Fit (II)

Starting from the *initial state* of parameters $\boldsymbol{\alpha}_0$, we propagate the trajectory layer by layer. So on each layer we have

- a *measurement* y_k
- a *propagated state* $y_k^{prop} = f_k(\boldsymbol{\alpha}_0)$

The residuals are

$$
\begin{aligned}
r_k &= y_k - y_k^{fit} \\
&= y_k - f_k(\boldsymbol{\alpha}) \\
&= \underbrace{y_k - f_k(\boldsymbol{\alpha}_0)}_{\Delta y_k} - \sum_i \underbrace{\frac{\partial f_k}{\partial \alpha_i}(\alpha_{i,0})}_{A_{k,i}} \cdot \underbrace{(\alpha_i - \alpha_{i,0})}_{\eta_i} \\
&= \Delta y_k - \sum_i A_{k,i} \cdot \eta_i
\end{aligned}
$$

Thus on each layer we have to pick up

- the (approximated) *residual* Δy_k
- its *derivatives w.r.t. the fit parameters* $A_{k,i}$

Global χ^2 Fit (III)

In matrix form: $\mathbf{r} = \Delta\mathbf{y} - \mathbf{A}\cdot\boldsymbol{\eta}$

$$\chi^2 = \left(\Delta\mathbf{y} - \mathbf{A}\cdot\eta\right)^t \mathbf{V}_\mathbf{y}^{-1} \left(\Delta\mathbf{y} - \mathbf{A}\cdot\eta\right) = \mathbf{r}^t\,\mathbf{V}^{-1}\,\mathbf{r}$$

with $\mathbf{V}_\mathbf{y}^{-1}$ being the *weight matrix*, inverse of the *covariance matrix* of the measurements y_i.

Minimizing, one gets: $\boldsymbol{\eta} = \mathbf{V}_\mathbf{A}\,\mathbf{A}^t\,\mathbf{V}_\mathbf{y}^{-1}\,\Delta\mathbf{y}$

with $\mathbf{V}_\mathbf{A} = \left(\mathbf{A}^t\,\mathbf{V}_\mathbf{y}^{-1}\,\mathbf{A}\right)^{-1} = \mathbf{V}_\eta$

Finally: $\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \boldsymbol{\eta}$ and $\mathbf{V}_\alpha = \mathbf{V}_\eta = \mathbf{V}_\mathbf{A}$

Notice that by replacing $\mathbf{V}_\mathbf{y} \rightarrow \mathbf{V} = \mathbf{V}_\mathbf{y} + \mathbf{V}_\mathbf{MS}$, the *multiple scattering* contribution can be taken into account as well.

Fitting the *scattering angles* and the *energy loss* is more complicated.

Virtues of the Global χ^2 Fitter

- Only *initial state* is needed, no need to *initialize* the *covariance matrix* of $\boldsymbol{\alpha}_0$
- *Non-iterative* procedure

→ *more stable*
(no efficiency loss, no resolution degrade)

Moreover...

… it yields the *scattering angles* and *residual derivatives* on each layer (e.g. used for alignment)

… it can be used (optionally) to resolve the *left-right ambiguity* and compute the *time pedestal* in the DT's

… it makes it easier to include *calorimeter measurements*

Vices of the Global χ^2 Fitter

- Inversion of large matrices: for *n* measurements, $\mathbf{V}_y$ is $n \times n$
  - if measurements are *all* uncorrelated, time for inversion ~ $O(n)$
  - if measurements are correlated (e.g. *multiple scattering*), time ~ $O(n^3)$
- Including scattering angles (2 per layer), many fit parameters
  - also $\mathbf{V}_A$ is very large

→ Need for *fast matrix inversion* algorithms

e.g. the *Bunch-Kaufman* method (in `CLHEP`) is used in ATLAS and requires ~ twice as much CPU time as the Kalman Filter

Using approximated inverse matrices, the estimate of the parameters is still *unbiased*, but with larger errors (*non-optimal least squares* fitting)

Future Plans

- A Global Fitter is not yet available in CMS, but some tools are already there
  - `ReferenceTrajectory` code by G. Flucke *et al.*
    used for Tracker alignment
  - specific for Tracker, not immediately usable outside
    e.g. uses the Analytic Propagator
- Need to figure out how to
  - handle the energy loss and multiple scattering in the Muon Chambers
  - feed the calorimeter deposits in the track fitting
- Possible applications
  - fitter for the stand-alone tracks
  - additional fitter for tracker and global tracks

Back-up slides

Global reconstruction

The stand-alone track defines a *region of interest* (ROI) in the Tracker.
Then two different strategies are available.

- Prompt reconstruction
  Tracker tracks are reconstructed *only* inside the ROI.
  It is used in the on-line trigger reconstruction for its rapidity

- Track matching
  Tracker tracks are built independently in the *whole* Tracker and the ones in the ROI are selected
  It is used in the off-line reconstruction

Reconstruction in the inner Tracker

Two different algorithms can be used to resolve the hit pattern in the Tracker

- **Kalman Filter**, with *seeds* built from 2 or 3 consecutive hits
- **Road Search:** starting from one hit in the innermost layer and one in the outermost layer, the hits are selected using pre-set roads

After the selection, the hits are fitted and the track is built.

The most compatible Tracker track is chosen and the whole set of hits is fitted (Tracker + Muon hits)