Alignment Strategy for the CMS Silicon Tracker

Marco Rovere[1]
for the CMS Collaboration
[1]Università di Milano-Bicocca & INFN

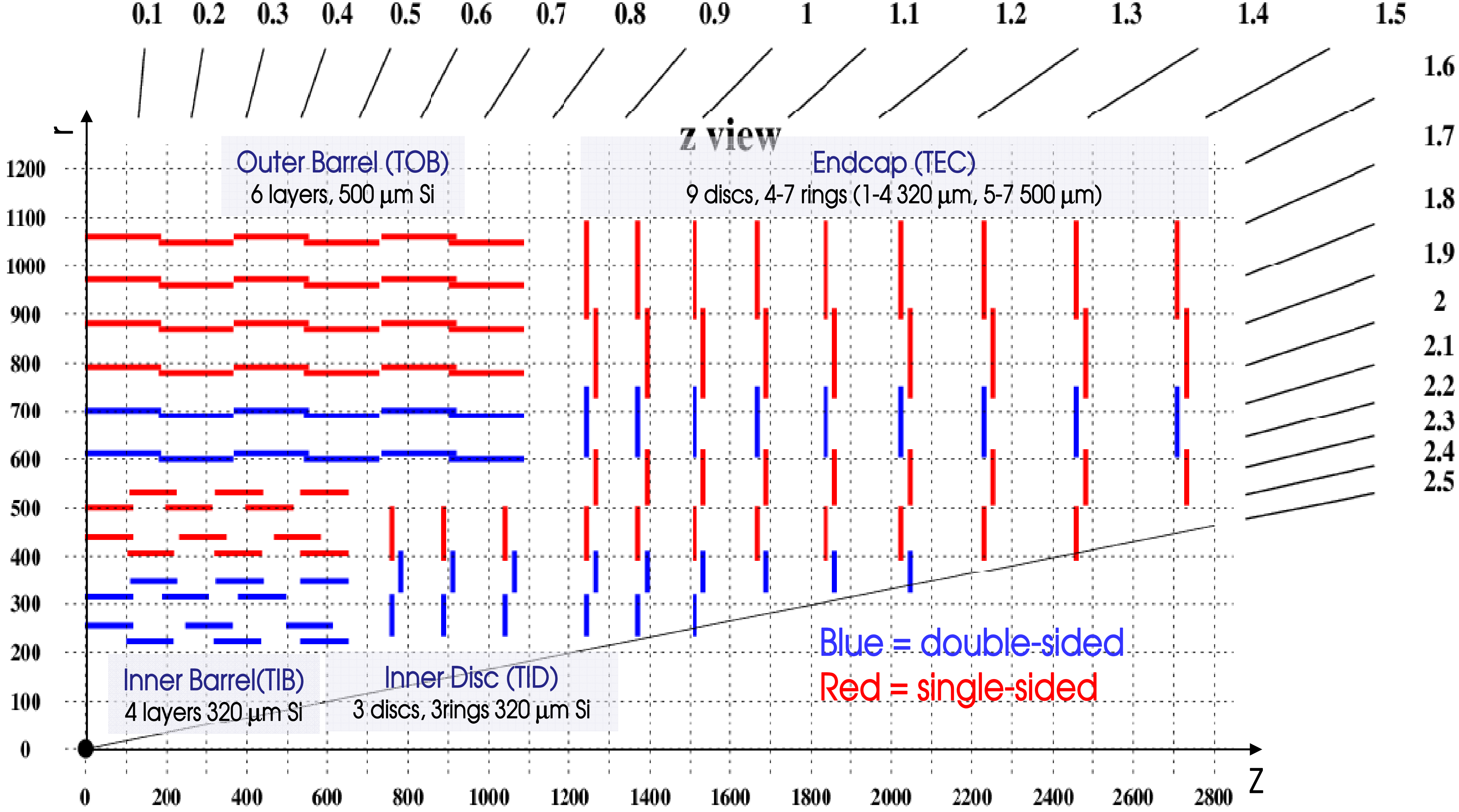
27-29 June 2007, RD07 Firenze

Outline

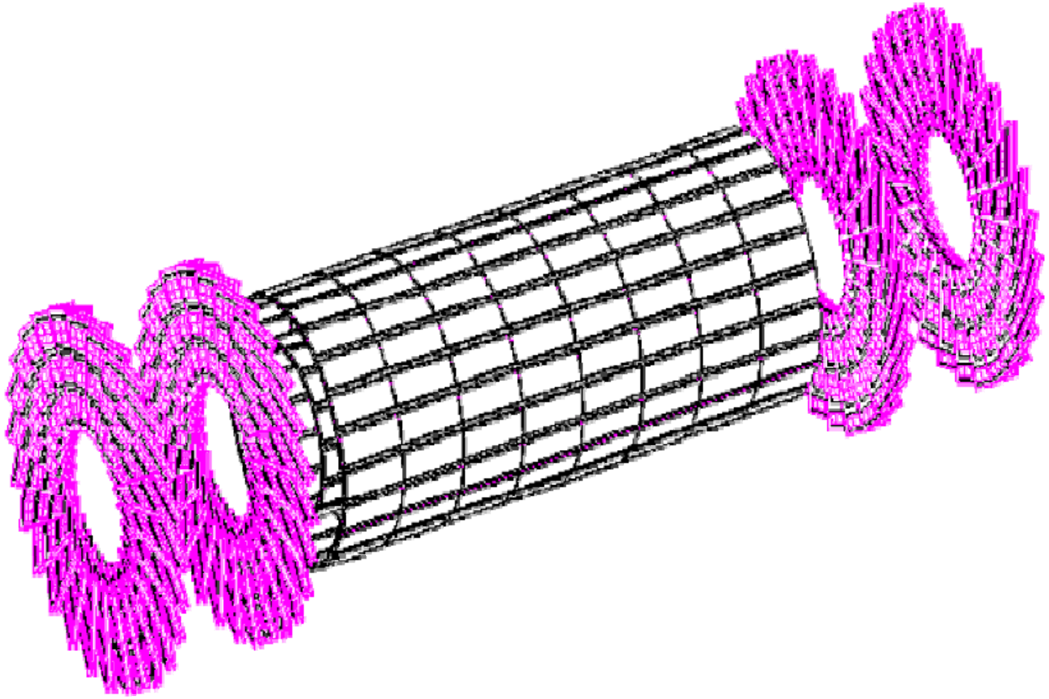
1. Alignment Challanges at CMS
   - Strip Tracker
   - Pixel Tracker
2. Hardware Alignment
3. Software tools for alignment
   - The framework simulation of misalignment
   - Alignment Algorithms:
     - HIP & Results
     - Millepede II &Results
     - Kalman Filter & Results
4. Conclusions

Alignment Challanges at CMS – Strip Tracker

- 15148 modules
- Strip pitch: 80-205 µm
- σ ≈ 23-60 µm



Alignment Challanges at CMS – Pixel Tracker



General Layout
- active area ~ 1m^2
- dimensions: 100 cm x 30 cm
- 66*10^6 channels
- pixel size: 100 μm (r) x 150 μm (z)

Hit Resolution
- resolution: 10 μm (r) x 15 μm (z)

Barrel layers
- r = (4.4;7.3;10.2)cm
- 1200 modules

Endcap disks
- r = 6cm-15cm
- 700 modules

Alignment Challenges at CMS – Summary

- The large number of independent silicon sensors (~15K) and their excellent resolution make the alignment of the CMS strip and pixel trackers a challenging task.
- Knowledge of detector positions should be known at the level of 10 μm in the r-ϕ plane. This level of accuracy can only be reached with a track-based alignment procedure. But…
- … a more realistic procedure would be:

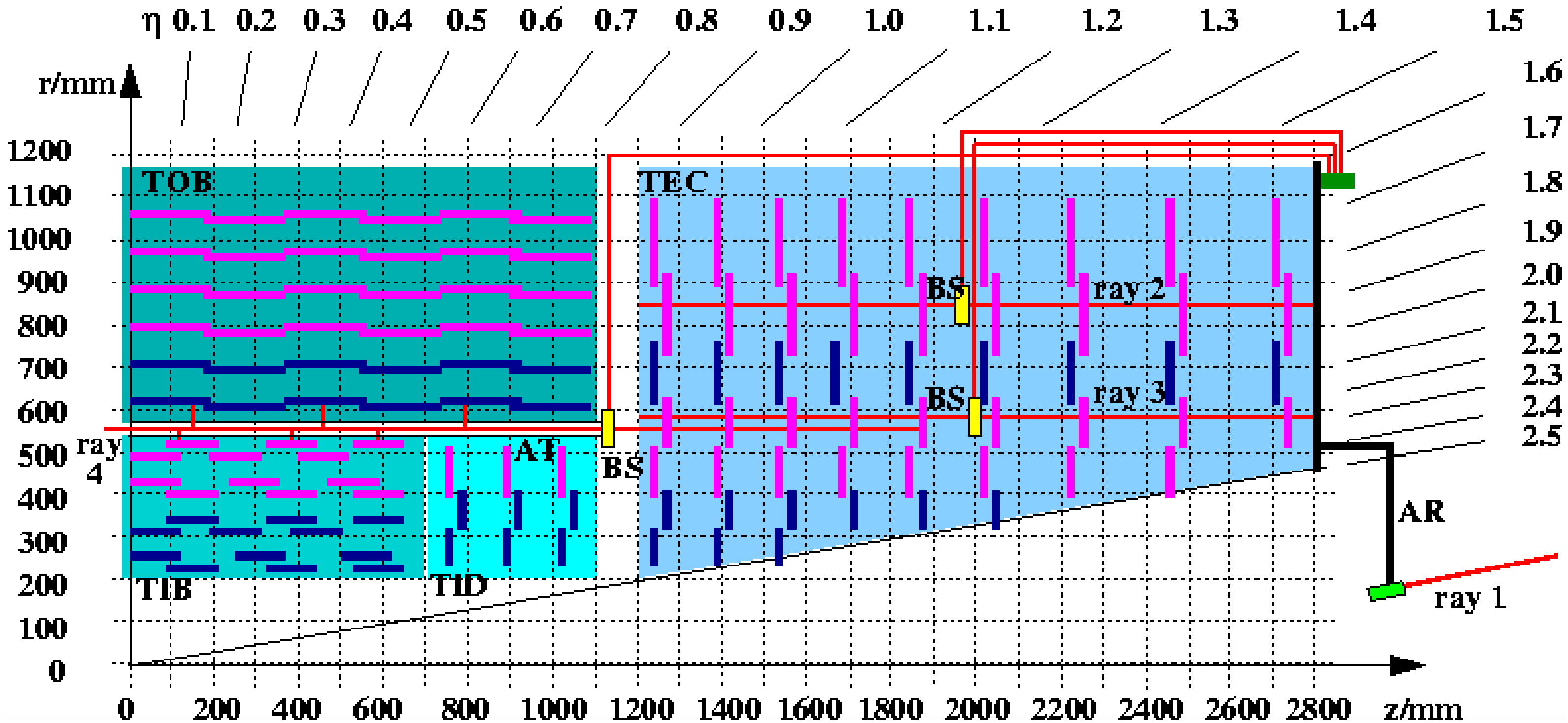
Hardware alignment

1. measurement of placement and its precision during assembly of tracking devices, e.g., from photogrammetry and detector position survey measurements
2. measurement of relative positions of sub-detectors using the Laser Alignment System (LAS)

T.B. alignment

3. track-based alignment

Hardware Alignment: LAS & Survey



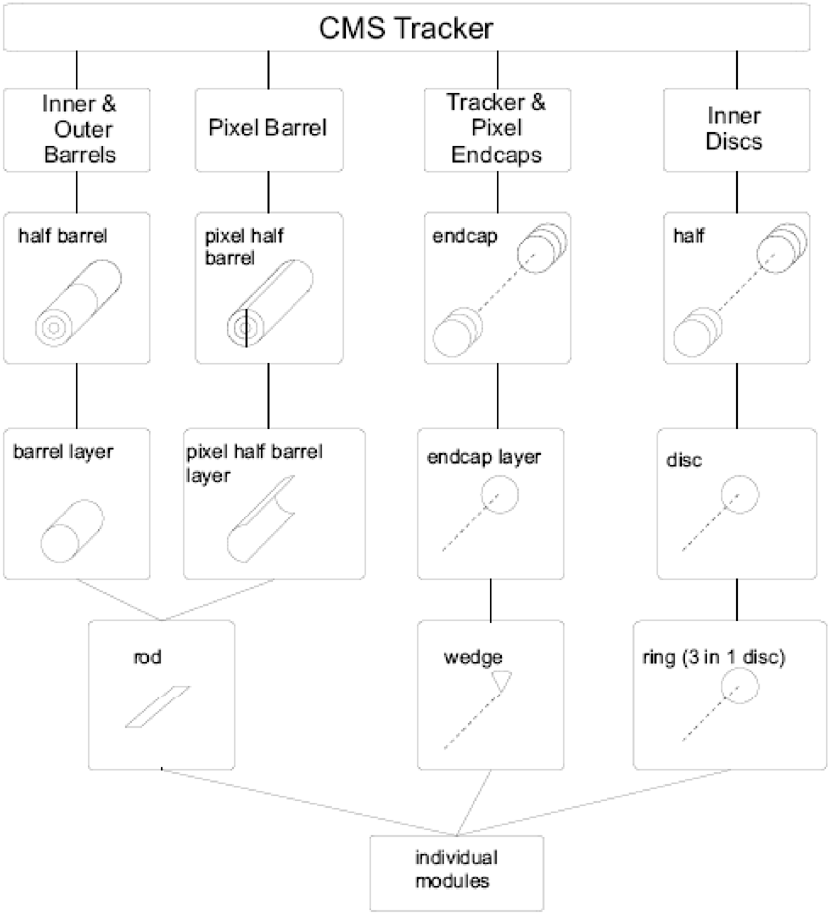
BS: Beam Splitter AT: Alignment tube AR: Alignment ring

- Survey: will provide an initial correction to assumed ideal Tracker geometry. If no complete measurement ⇒ an estimate of the placement uncertainty is added to the error of the track hit position leading to an improved efficiency during initial track reconstruction.
- Survey&LAS: In order to make efficient pattern recognition for the track reconstruction possible at CMS start-up, it is sufficient that the individual positions of the silicon sensors are known to about 100 μm. This can be achieved with a combination of survey and LAS measurements.

Software Tools: Simulation of misalignment

To study the impact of Tracker misalignment on track and vertex reconstruction in concrete physics analysis channels, as well as to study track-based alignment algorithms, a realistic model of misalignment effects has been implemented within the standard CMS reconstruction software (CMSSW).

- (Mis)alignment implemented at reconstruction level:
  - "Misalignment tools":
  - Implemented as a hierarchical structure
  - Ability to move and rotate modules or higher level structures



Software Tools: Simulation of misalignment

To study the impact of Tracker misalignment on track and vertex reconstruction in concrete physics analysis channels, as well as to study track-based alignment algorithms, a realistic model of misalignment effects has been implemented within the standard CMS reconstruction software (CMSSW).

- (Mis)alignment implemented at reconstruction level:
  - "Misalignment tools":
  - Implemented as a hierarchical structure
  - Ability to move and rotate modules or higher level structures
- Dedicated "Misalignment Scenarios"
  - Short term scenario
    - First data taking (few 100 pb^{-1})
    - Pixel already aligned
    - Strip tracker misaligned, only survey and laser alignment
  - Long term scenario
    - Few fb^{-1} accumulated
    - Full alignment performed, residual misalignments ~20μm
- Fast track refit (without redoing pattern recognition)
- implemented in standard CMS reconstruction software using a common layer of general functionality
  - Management of parameters and covariances
  - Derivatives wrt track and alignment parameters
  - I/O, Database connection

		Δx (μm)	Δy (μm)	Δz (μm)	R_Z (μrad)	LAS available
TPB	Dets	13	13	13	0	No
	Rods	5	5	5	0	
	Layers	10	10	10	10	
TPE	Dets	5	5	5	0	No
	Petals	2.5	2.5	2.5	2.5	
	Layers	5	5	5	5	
TIB	Dets	200	200	200	0	Yes
	Rods	200	200	200	0	
	Layers	100	100	500	50	
TOB	Dets	100	100	100	0	Yes
	Rods	100	100	100	0	
	Layers	70	70	500	90	
TID	Dets	100	100	100	0	No
	Rings	300	300	300	0	
	Layers	400	400	400	100	
TEC	Dets	50	50	50	0	Yes
	Petals	100	100	100	0	
	Layers	60	60	500	45	

Software Tools: the Alignment Algorithms

The CMS Collaboration has developed 3 independent (complementary) algorithms to align the tracker

1. HIP
2. Millepede I&II
3. Kalman Filter

- Algorithms are implemented in standard CMS reconstruction software using a common layer of general functionality
  - Management of parameters and covariances
  - Derivatives wrt track and alignment parameters
  - I/O, Database connection

Alignment Algorithms: HIP – Hit and Impact Point

- Minimization of track impact point (x) - hit (m) residuals in local sensor plane as function of alignment parameters
- χ^2 function to be minimized on each sensor (after many tracks per sensor accumulated)
  - V: covariance matrix of measurement
- Linearized χ^2 solution:
  - δp is the vector of alignment parameters, namely $\delta p=(\delta u, \delta v, \delta w, \delta\alpha, \delta\beta, \delta\gamma)$
  - J_i : derivative of residuals w.r.t. alignment parameters
- Local solution on each "alignable object"
  - Only inversion of small (6x6) matrices
  - computationally light
- Correlations between modules not included explicitely but …
- … implicitely included through iterations

$$\epsilon = \begin{pmatrix} \epsilon_u \\ \epsilon_v \end{pmatrix} = \begin{pmatrix} u_x - u_m \\ v_x - v_m \end{pmatrix}$$

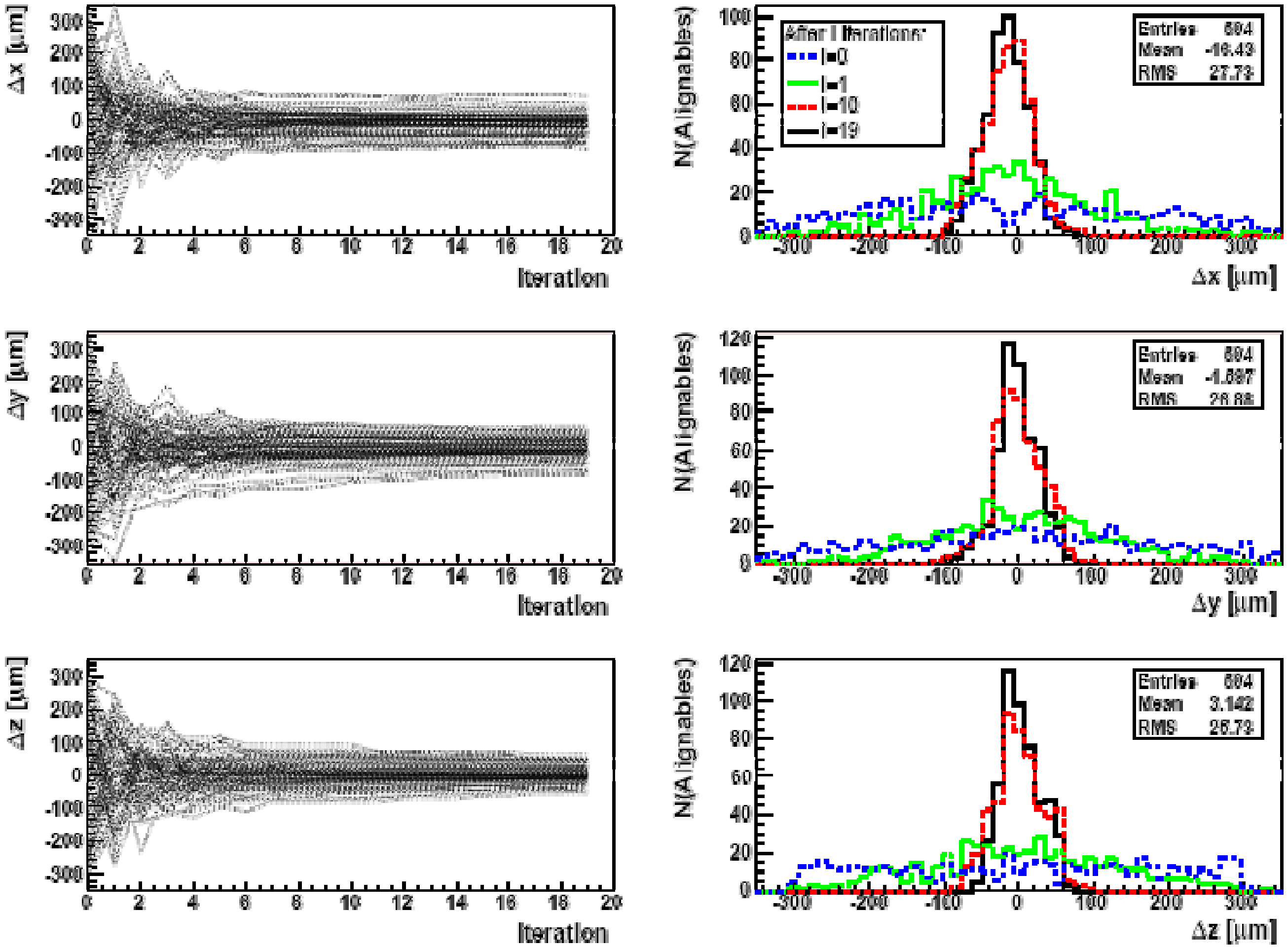
$$\chi^2 = \sum_{i=1}^{\mathrm{N_{hits}}} \epsilon_i^T V_i^{-1} \epsilon_i$$

$$J_i = \nabla_{\overline{p}} \epsilon_i (\overline{p})$$

$$\delta \mathrm{p} = \left[\sum_{i=1}^{\mathrm{N_{hits}}} J_i^T V_i^{-1} J_i \right]^{-1} \left[\sum_{i=1}^{\mathrm{N_{hits}}} J_i^T V_i^{-1} \epsilon_i \right]$$

Alignment Algorithms: HIP Results

- Standalone alignment of pixel modules
- Minimize influence of misaligned strip detector:
  - refitting only pixel hits of the tracks
  - use momentum constraint from full track (significantly improves convergence)
- Two muons from $Z^0 \to \mu^+ \mu^-$ are fitted to common vertex
- Flat misalignment 300μm in x,y,z
- 500K events, 19 iterations
- Resonable convergence, RMS ~25μ m in all coordinates



Alignment Algorithms: Millepede I …

- Millepede is a linear least square method
- The global (alignment) parameters and the local (track) parameters are treated simultaneously
- Unique solution, no iterations.
- Constraints can be implemented via Lagrangian multipliers.
- Initial "knowledge" can be implemented via χ^2 penalties.
- Millepede I algorithm decouples global (alignment) and local (track) parameters.
  - linear equation system with only N (N = number of alignment parameters) needs to be solved!
- Millepede I determines a by inversion of C'
  - The CPU times for inversion scales with N^3, memory with N^2.
- Millepede I is limited to ~10^4 parameters.

$$\begin{pmatrix} \sum C_i & \cdots & G_i & \cdots \\ \vdots & \ddots & 0 & 0 \\ G_i^T & 0 & \Gamma_i & 0 \\ \vdots & 0 & 0 & \ddots \end{pmatrix} \cdot \begin{pmatrix} a \\ \vdots \\ \alpha_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \sum b_i \\ \vdots \\ \beta_i \\ \vdots \end{pmatrix}$$

$$\begin{pmatrix} C' \end{pmatrix} \begin{pmatrix} a \end{pmatrix} = \begin{pmatrix} b' \end{pmatrix}$$

$$C' = \sum_i C_i - \sum_i G_i \Gamma_i^{-1} G_i^T \qquad b' = \sum_i b_i - \sum_i G_i \left(\Gamma_i^{-1} \beta_i\right)$$

Alignment Algorithms: … Millepede II

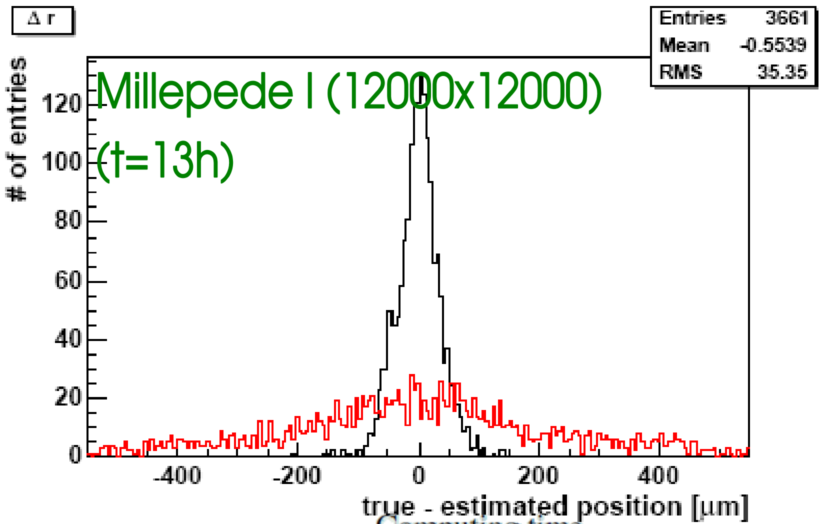
- Millepede II especially developed by Volker Blobel to handle next generations detector needs.
- Millepede II has a new method to solve the matrix equation: it numerically minimises | C′a-b′ |.
- The numerical (iterative) method uses the fact that the matrix is sparse: only nonzero elements are stored in double precision.
- Millepede II is faster and can handle a higher number of parameters.

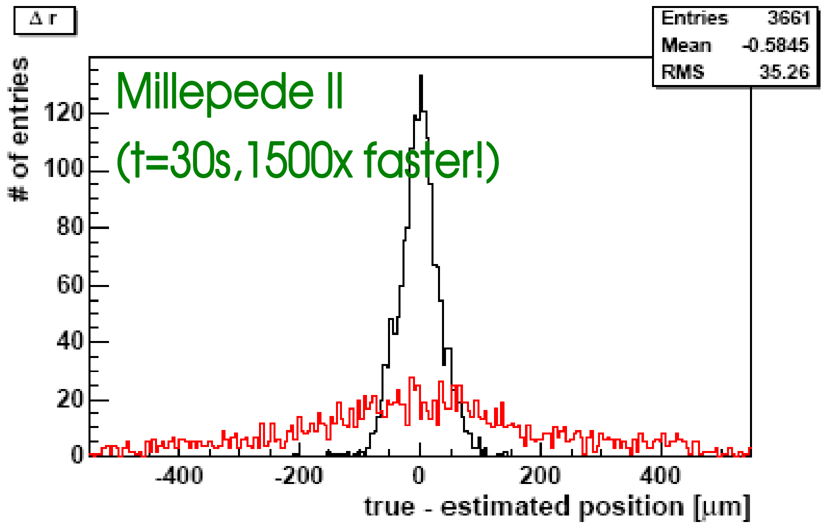
$$\begin{pmatrix} \sum C_i & \cdots & G_i & \cdots \\ \vdots & \ddots & 0 & 0 \\ G_i^T & 0 & \Gamma_i & 0 \\ \vdots & 0 & 0 & \ddots \end{pmatrix} \cdot \begin{pmatrix} a \\ \vdots \\ \alpha_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \sum b_i \\ \vdots \\ \beta_i \\ \vdots \end{pmatrix}$$

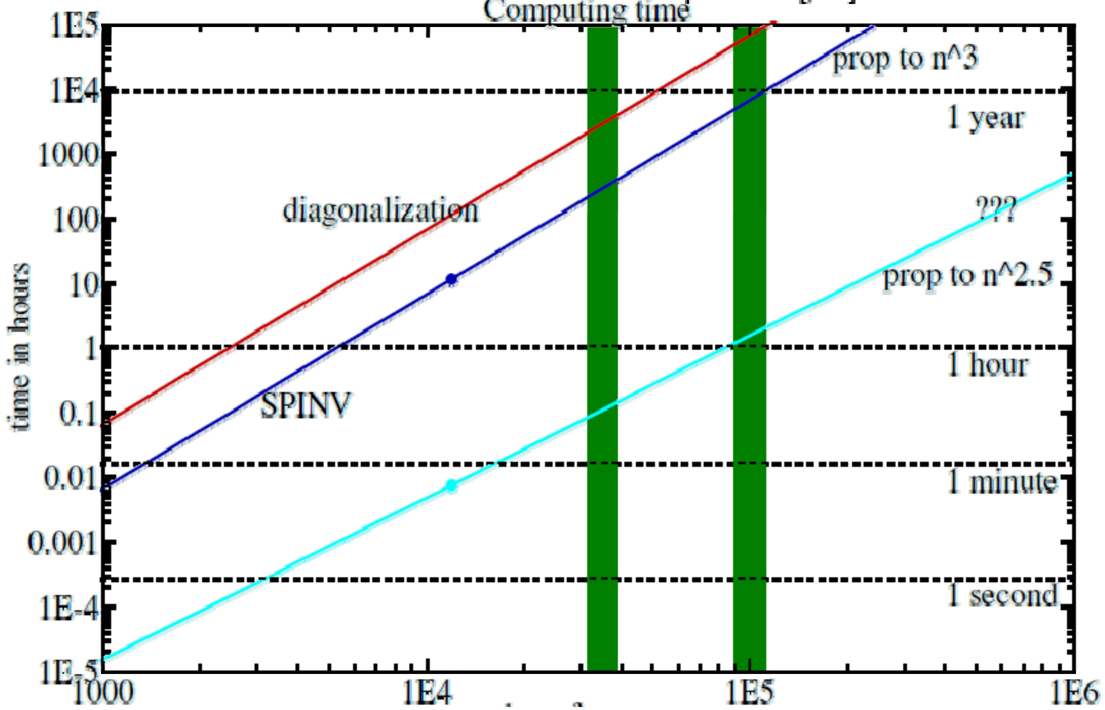
$$\begin{pmatrix} C' \end{pmatrix} \begin{pmatrix} a \end{pmatrix} = \begin{pmatrix} b' \end{pmatrix}$$

$$C' = \sum_i C_i - \sum_i G_i \Gamma_i^{-1} G_i^T \qquad b' = \sum_i b_i - \sum_i G_i \left(\Gamma_i^{-1} \beta_i\right)$$

Alignment Algorithms: Millepede I & II Comparison







CPU Time for CMS (100k parameters):

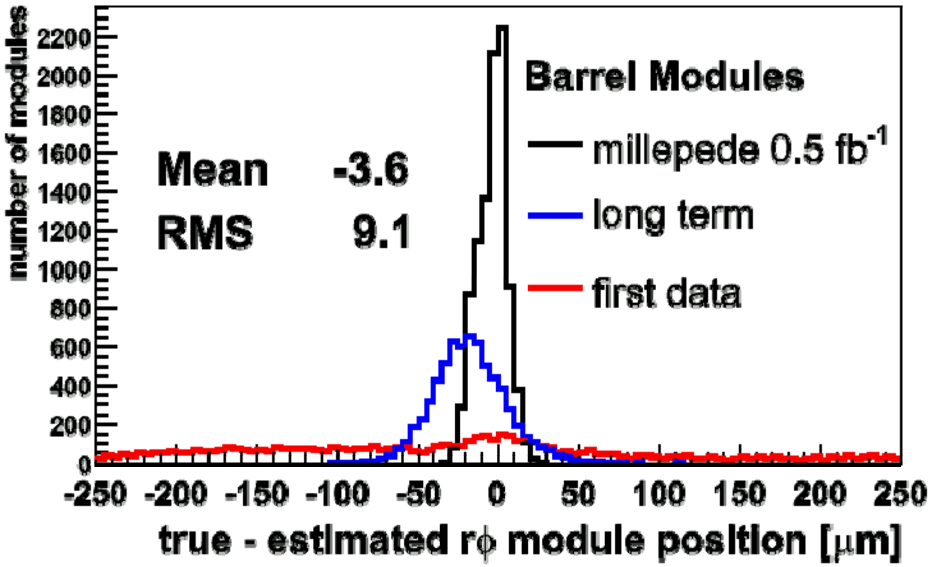
- Diagonalization *~ 10 year* @1CPU
- Inversion *~1 year* @1CPU
- Iteration *~ 1 h* @1CPU

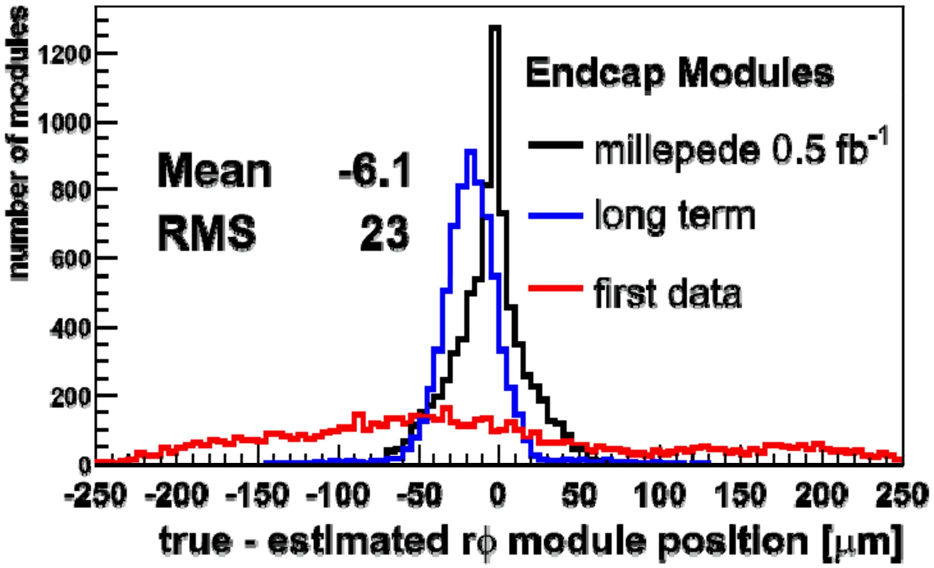
Alignment Algorithms: Millepede II Results

- Misalignment: Default first data scenario.
- Data sets:
  - 0.5 mio. Z (0.5 fb-1) mass and vertex constraint
  - 25 k cosmics with momentum > 50 GeV
  - Single muons of 1.5 mio. Z
  - ~ 3 mio W (0.5 fb-1) events
- Alignment:
  - All silicon modules (PB,PE,TIB,TID,TOB,TEC)
  - translation and the rotation around normal of sensor.

- Align the full strip and pixel tracker!
- Number of aligned parameter ~ 50K
- CPU time total: 1h:40min
- Use of complementary data sets.
- Utilizing initial knowledge.

Full alignment procedure tested!





Alignment Algorithms: Kalman Filter

- Method for global alignment derived from Kalman Filter
- How it works:
  - measurements *m* depend via track model *f* not only on track parameters *x*, but also on alignment parameters *d*:

$$m = f(x, d) + \epsilon \qquad \mathrm{Cov}(\epsilon) = V$$

  - Update equation of Kalman Filter:

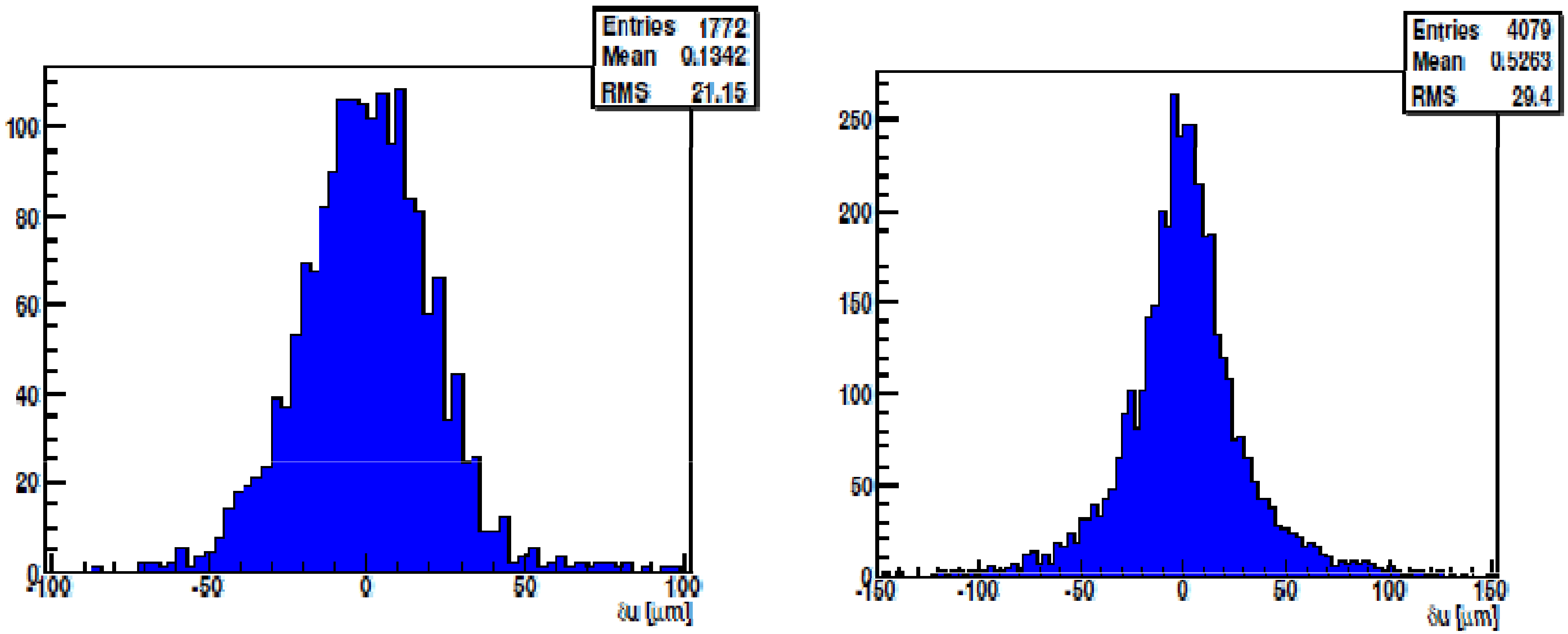
$$\begin{pmatrix} \hat{d} \\ \hat{x} \end{pmatrix} = \begin{pmatrix} d \\ x \end{pmatrix} + K(m - c - Ad - Bx)$$

- Iterative: Alignment Parameters updated after each track
- Global: Update not restricted to modules crossed by track
  - Update can be limited to those modules having significant correlations with the ones in current trajectory
  - Requires some bookkeeping
  - No large matrices to be inverted!
- Possibility to use prior information (e.g. survey data, laser al.)
- Can add mass / vertex constraints

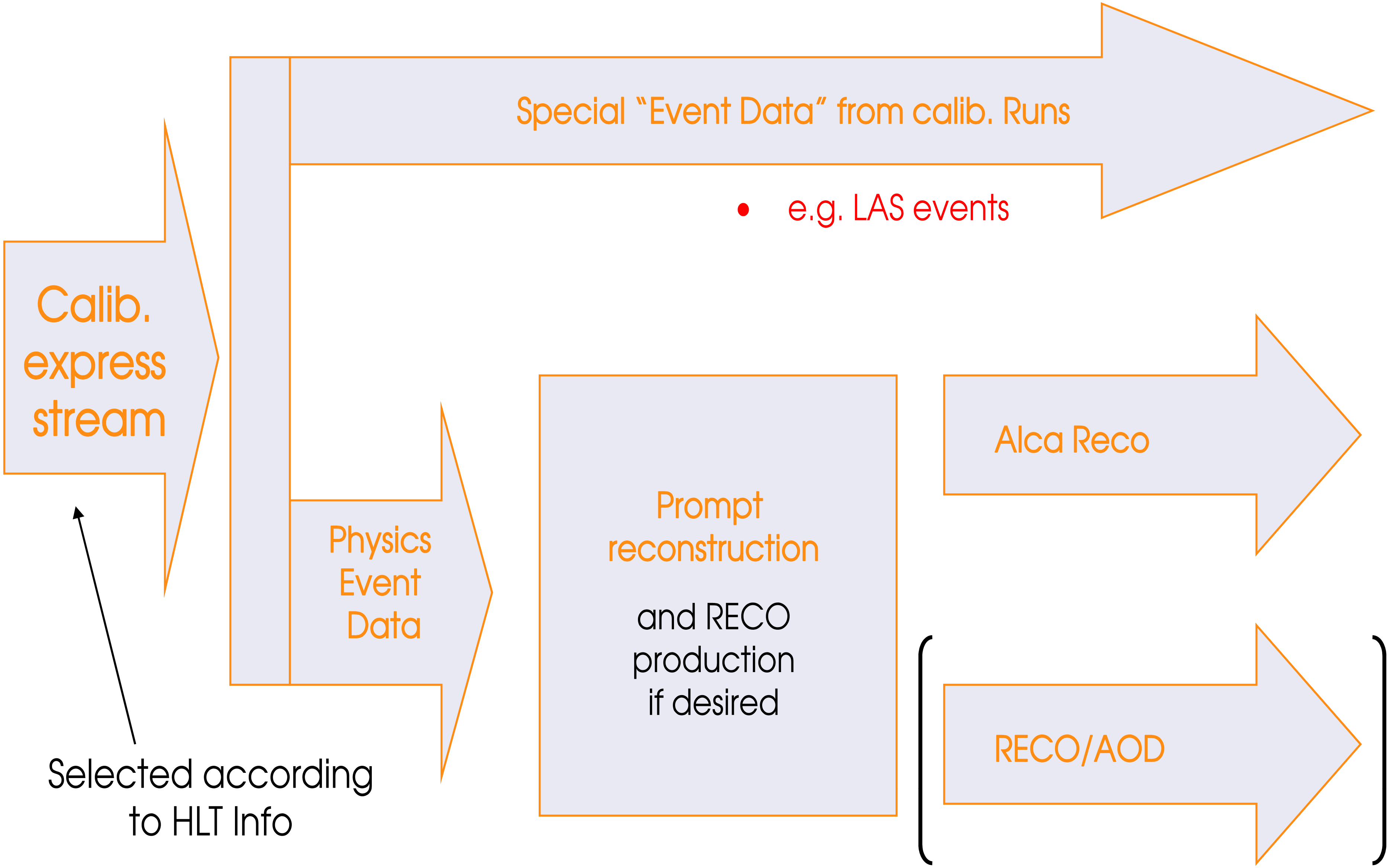
Alignment Algorithms: Kalman Filter Results

Example of TIB(left) and TOB(right) alignment

- use ~75K $Z^0 \rightarrow \mu^+\mu^-$ tracks (no mass-constrain applied)
- cpu time ~50 min(left) ~ 90min(right)



Data Flow for Alignment&Calibration



AlCaReco Format

- AlCaReco for tracker alignment
  - Reduced data format containing only tracks used for alignment (plus associated hits for refitting)
    - Very little disk space (local disk storage)
    - Fast processing (important especially for iterating algorithms)
- AlCaReco producers
  - Run during prompt reconstruction at Tier-0
  - Read express stream written by HLT
  - Write AlCaReco files
  - Functionality:
    - Select appropriate events (e.g. $Z \rightarrow \mu\mu$)
    - Write out reduced information (e.g. only two muon tracks)

Data Samples

Luminosity	10^{32} cm^{-2} s^{-1}		2×10^{33} cm^{-2} s^{-1}		
Time int. Luminosity	Few weeks	6 months	1 day	Few weeks	One year
	100 pb^{-1}	1fb^{-1}		1fb^{-1}	10fb^{-1}
$W^\pm \rightarrow \mu^\pm \nu$	700K	7M	100K	7M	70M
$Z^0 \rightarrow \mu^+ \mu^-$	100K	1M	20K	1M	10M

- Collision events
  - High Pt isolated muons from W,Z decays
  - Isolatedhigh pt tracks in min. bias / QCD jet events (at startup)
  - Muons from J/Psi / Upsilon
- Non-collision events
  - Cosmic Muons
  - Beam Halo Muons
- Special events
  - Laser alignment system

Conclusions

- Alignment of the CMS tracker and muon system is a challenge
  - Large number of parameters (~100,000 in tracker)
  - High intrinsic resolution of devices
- A lot of work on track based alignment already done
  - Implementation and further development of 3 different algorithms
  - Alignment studies using various MC data sets
  - Dedicated HLT alignment stream
  - Use of mass, vertex constraints and survey information