Kinematic fit and decay chain reconstruction library for CMS

Kirill Prokofiev, Thomas Speer

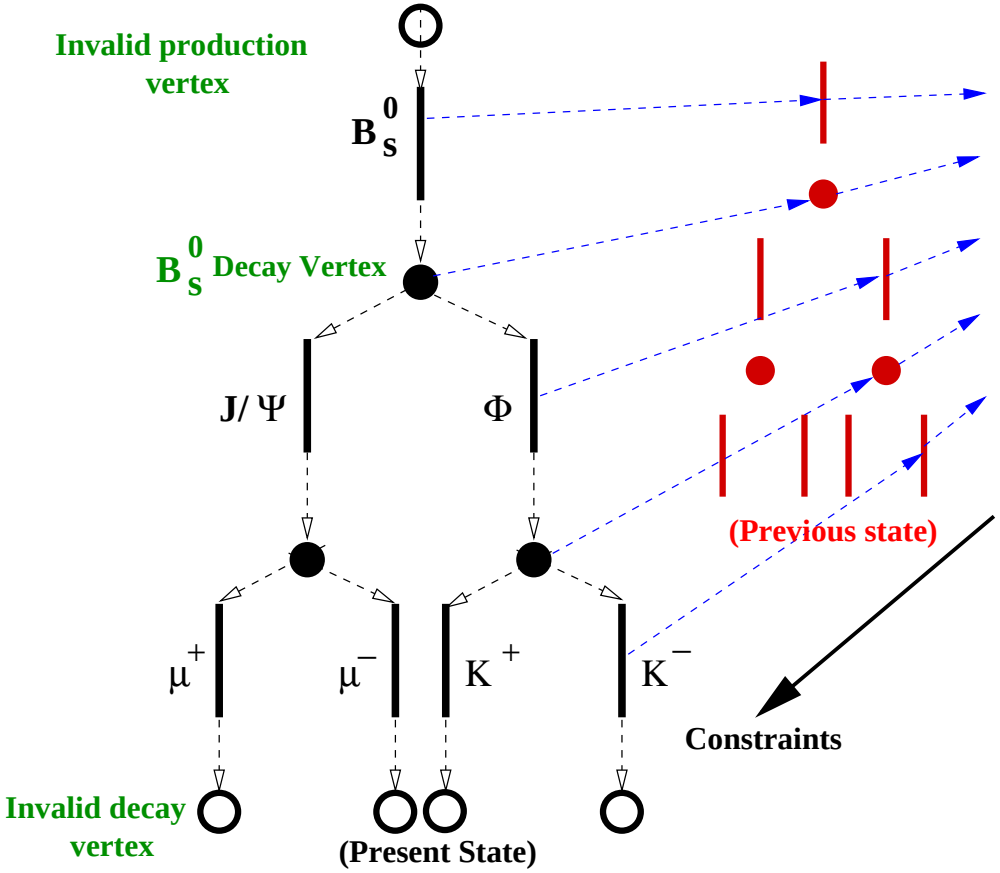
Physik-Institut Universität Zürich-Irchel

Introduction

- Kinematic fit: through constraints derived from physics laws
  - improve the resolution of experimental measurements
  - test hypothesis
  - find unknown parameters
- Aim of the KinematicFit library for the CMS reconstruction framework:
  - Flexible framework with generic minimization algorithms which do not depend on constraints
  - Constraint chosen and implemented by the user.
  - Provide a navigable decay chain, representing the reconstructed physical process
    - stores results of all constraint fits performed during the reconstruction of the current decay
    - access to the constrained and initial unconstrained information

The decay tree: KinematicTree

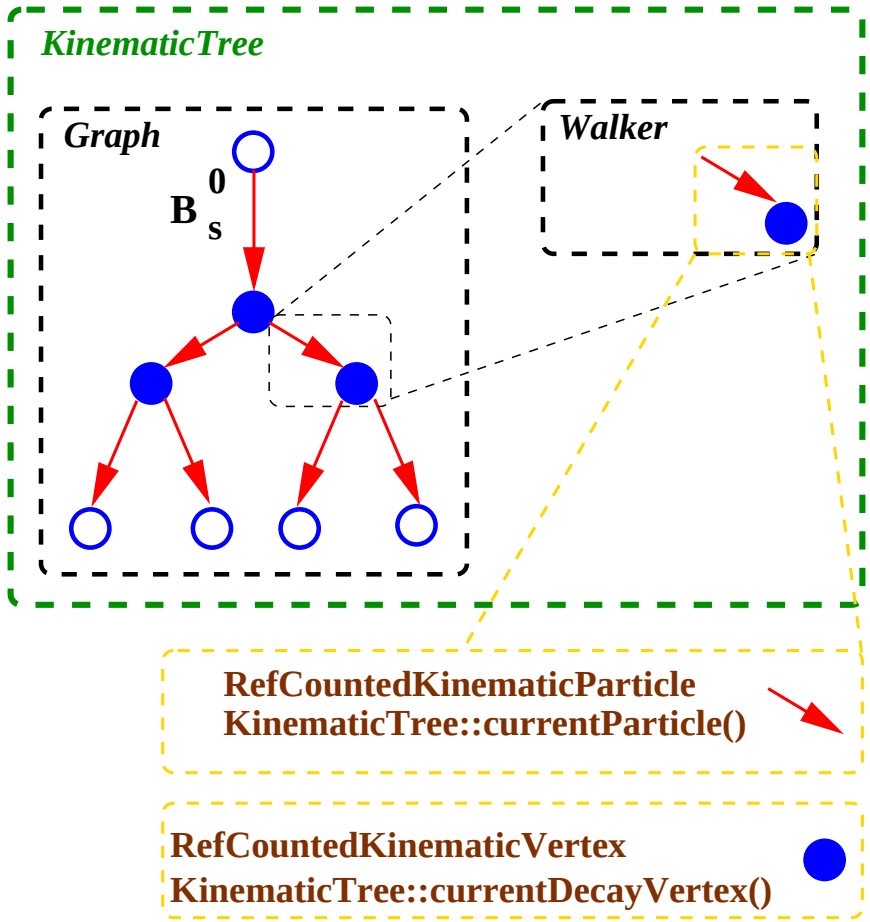
- Decay tree made of independent particles and vertices
- Tree is created from "bottom" to "top", i.e. from final state to the decayed particle
- Result of the reconstruction
  - represents one hypothesis
- Combinatorial search:
  - collection of trees can be created, each representing one possible combination of input objects



- State of any component can be changed (e.g. new constrained kinematic fit):
  - every component "remembers" the last constraint applied and its state before that constraint.

The decay tree: KinematicTree

- *KinematicTree* has a graph-based navigation mechanism
  - Vertices are nodes
  - Particles are edges
  - Particle and vertices are reference counted
- Graph structure of the decay is not seen by user:
  - User can add and modify states, collect the information, navigate up and down the tree through public methods



The decay tree: particles and vertices

- *KinematicParticle*
  - Represents a particle during the kinematic fit.
  - Stores the trajectory state, mass, charge, corresponding covariance matrix, etc
  - Can be created out of any 4-vector based physical object:
    - Reconstructed object (e.g. track with mass hypothesis, jets) by using adequate adapters - contains link to the original object
    - Decayed particle during fit: inferred from its decay products
- *KinematicVertex*
  - Describes a vertex in constraint fit.
    Stores the vertex position, covariance matrix, etc
- Both classes provide the link to the tree they belong to and store their previous states and last constraint applied
- In a *KinematicTree, KinematicParticle* and *KinematicVertex* are reference counted

The kinematic fit: requirements

- Several requirements drove the design of the kinematic fit library:
  - minimization algorithm must be independent of the constraints
  - flexibility to incorporate arbitrary constraints
    - → different physics analysis with their different requirements
  - Addition of new constraint must be easy
    - →Developed and implemented by users and shared
- Minimization: Least Mean Squares (LMS) with Lagrange multipliers
  - analytical solution for linear constraints
  - constraints can be linearized

The kinematic fit: LMS minimization

- χ^2 minimization with the set of additional constraints $H(y_{ref})=0$, linearized (first order Taylor expansion) around some given point $y_{\exp}$

$$\frac{\partial H(y_{\exp})}{\partial y}(y-y_{\exp})+H(y_{\exp})=D\,\delta y+d=0$$

  - *D*: matrix of derivatives, one line per constraint equation *(n_equations x n_parameters)*
  - *d*: vector of values of constraints

- Function to minimize with respect to (y_{ref},λ):

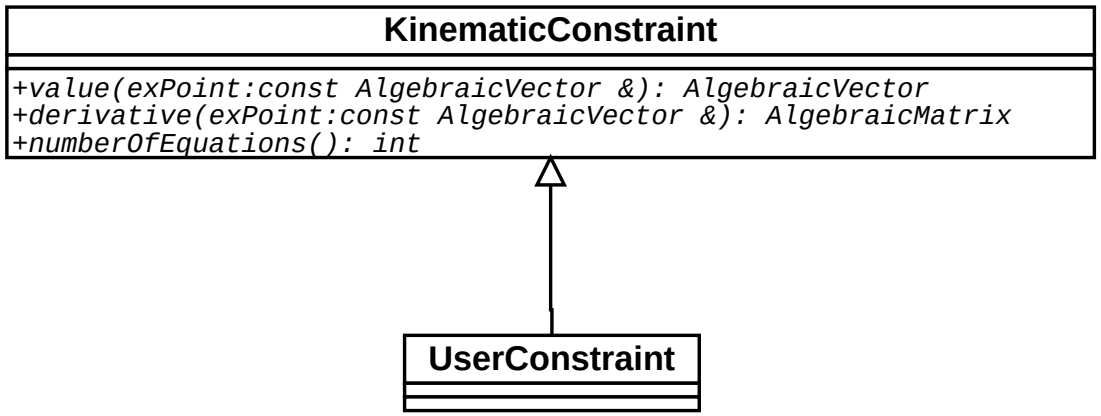
$$\chi^2=(y^{ref}-y)V_y^{-1}(y^{ref}-y)^T+2\lambda^T(D\,\delta y+d)\rightarrow min$$

- Minimization problem has an analytical solution, independent of the constraint equations
- Iterations may have to be performed if initial expansion point is far from minimum

Fitting algorithms: The constraints

- Each constraint equation adds one line into the *D* matrix, and one value into the *d* vector
- Implementation: one class for each constraint (inheriting from the abstract base class *KinematicConstraint)*

  Each constraint class has to return the relevant lines for the *D* matrix and values for the *d* vector for a given set of parameters.



- Several constraints can be used in the same fit:

  Special class collects and assembles the contributions of the individual constraints into the *D* matrix and *d* vector

The kinematic fit: reconstruction strategies

- Global strategy:
  - Constrained fit of several particles with a vertex constraint and any additional number of constraints (e.g. constraints on subset of final state tracks: collinearity, back-to-back, invariant mass)
  - All the constraints are applied together at the same time
  - Vertex constraint adds $2 \cdot N^{tracks}$ to the user-specified constraints
  - Fitter handles construction of D matrix and d vector (vertex+user constraints):

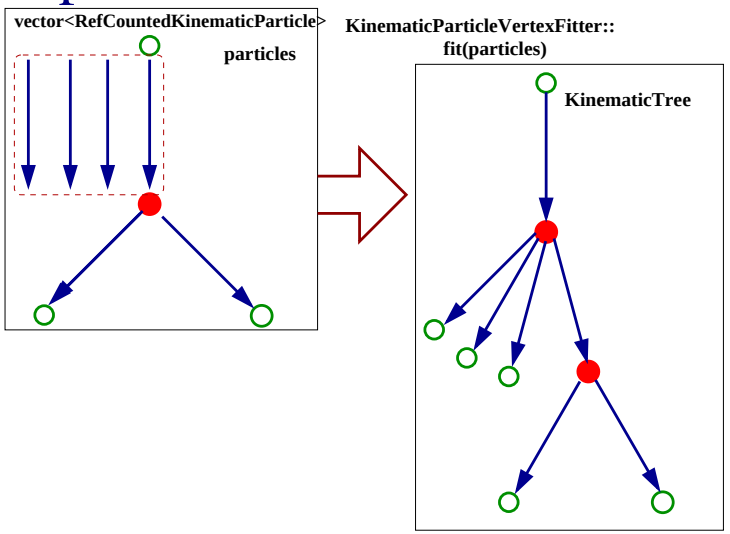
KinematicConstrainedVertexFitter::fit (vector<KinematicParticles>, KinematicConstraint)

The kinematic fit: minimization

- Sequential strategy
  - Constraints are applied sequentially, one after the other, after the vertex fit on the reconstructed mother particle.
  - Sequential fit mathematically equivalent to global fit
  - Unstable particles with significant lifetimes: reconstructed state have to be propagated inside the detector
  - Vertex fit with any vertex fitter already implemented (e.g. Kalman filter, interfaced through *KinematicParticleVertexFitter*)
  - Constrained applied on the mother particle: *KinematicParticleFitter*

The kinematic fit: KinematicTrees

- The “mother” particle is created form its decay products after global fit or vertex fit (sequential strategy)
- A new, fully consistent *KinematicTree* is then produced:
  - Trees of initial particles (if any)
  - Total χ^2 and number of degrees of freedom
  - Fitted vertex
  - Refitted input states
  - New “mother” particle:
    - momentum: sum of momenta of refitted decay products at vertex
    - covariance matrix: calculated from the full particle-to-particle covariance matrix taking all correlations into account



- Sequential strategy: constraint of the “mother particle” modifies the state
  - State of that particle in the tree is updated

The kinematic fit: parametrizations

- The analytical solution of the minimization problem does not depend on the parametrization of the input data
- Constraint equations must be derived in the same parametrization as the fit is performed
  - Global strategy: vertex and additional constraints have to be derived in the same frame - “quasi-Cartesian” parametrization is used
  - Sequential strategy: independent minimization for each constraint
    - Different parametrizations for each fit/constraint
    - Some non-linear constraints may become linear after a change of parametrization.
    - Implementation of the parametrization-independent version in progress
- The particle state is stored in a *KinematicState* class in a “quasi-Cartesian” parametrization: (position, momentum, mass)

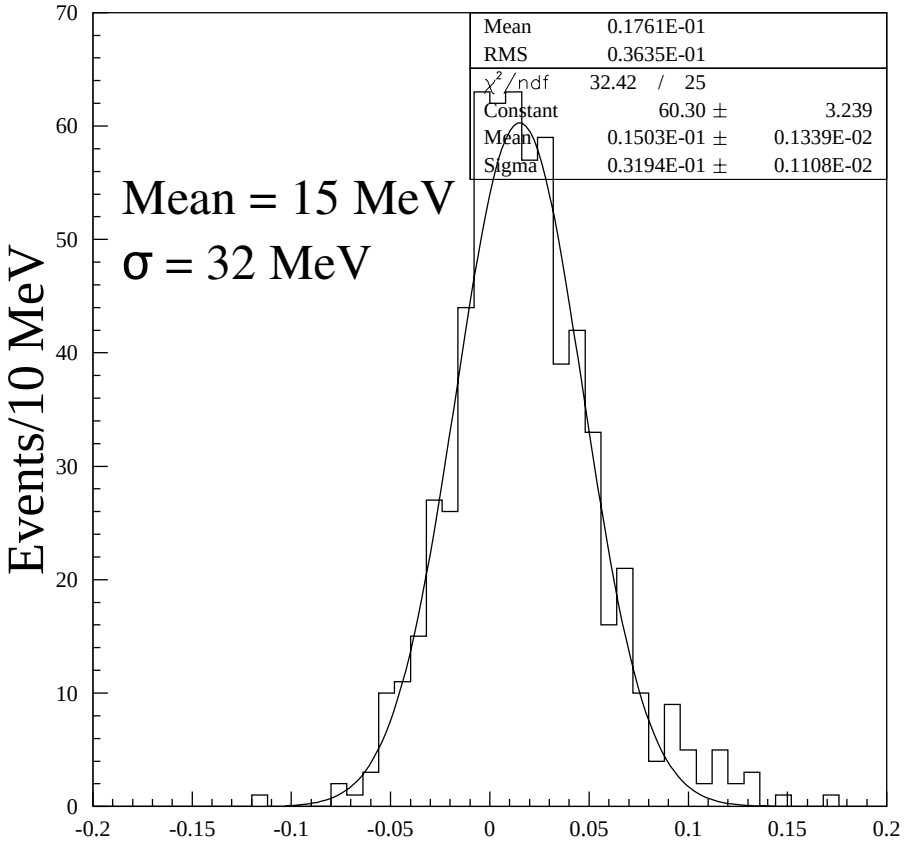
Example: reconstruction of the decay $B_s \rightarrow J/\psi\ \phi$

- Reconstruction of the B_s in the decay $B_s \rightarrow J/\psi\ \phi \rightarrow \mu^+\ \mu^-\ K^+\ K^-$
- Constraints:
  - 4 final state tracks have a common vertex
  - Invariant mass of the muons is equal to the mass of the J/ψ
  - Pointing constraint: reconstructed B_s momentum points toward the primary vertex

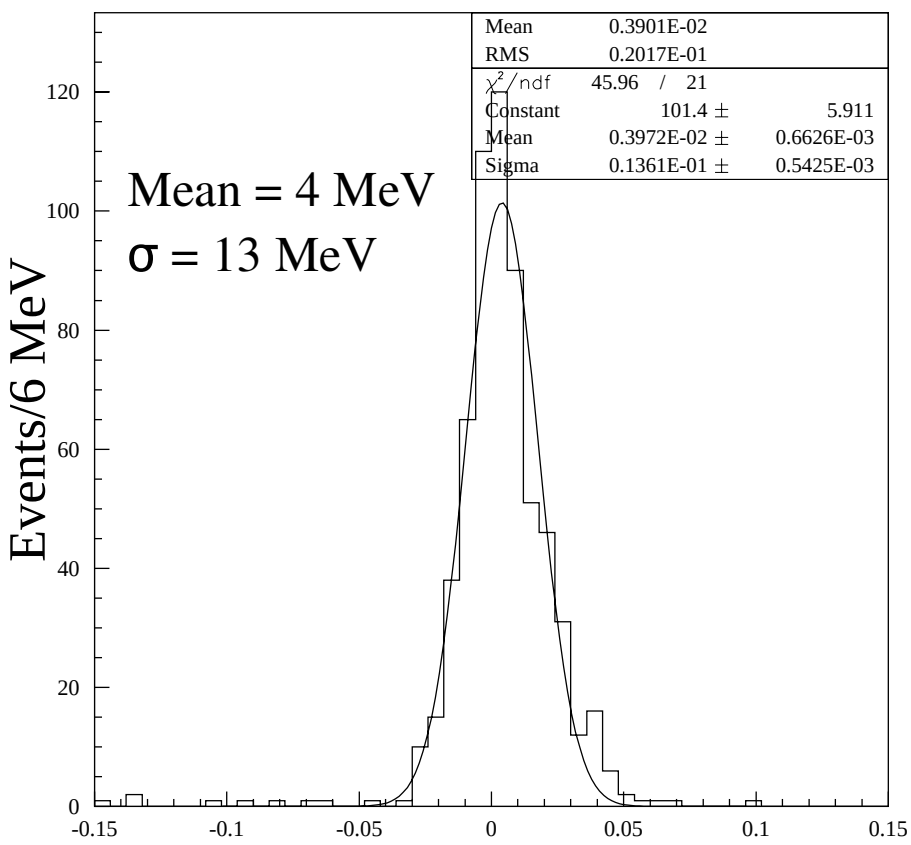
  (momentum vector parallel to the vector from the primary to the secondary vertex)
- Reconstruction without constraint:
  - Vertex reconstruction with Kalman filter, B_s parameters calculated from decay products.
- All test are performed with sample of 1000 simulated signal events

Example: reconstruction of the decay $B_s \to J/\psi\,\phi$

Residual of the $\mu^+\mu^- K^+K^-$ 4-track invariant mass with and without constraints:



Without constraints $M^{rec}_{B_s} - M^{PDG}_{B_s}$

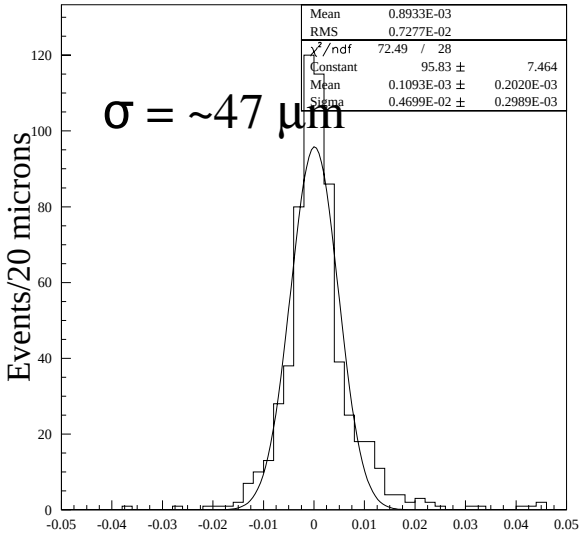


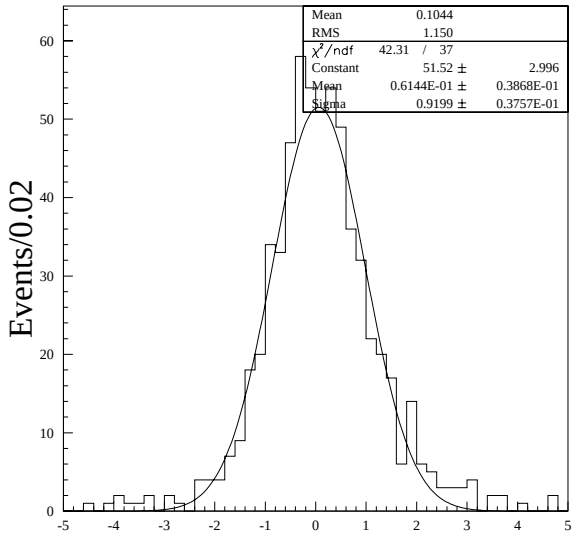
With constraints $M^{rec}_{B_s} - M^{PDG}_{B_s}$

Example: reconstruction of the decay $B_s \to J/\psi\,\phi$

Residual and pulls of the *x*-coordinate of the reconstructed B_s vertex

without constraints:

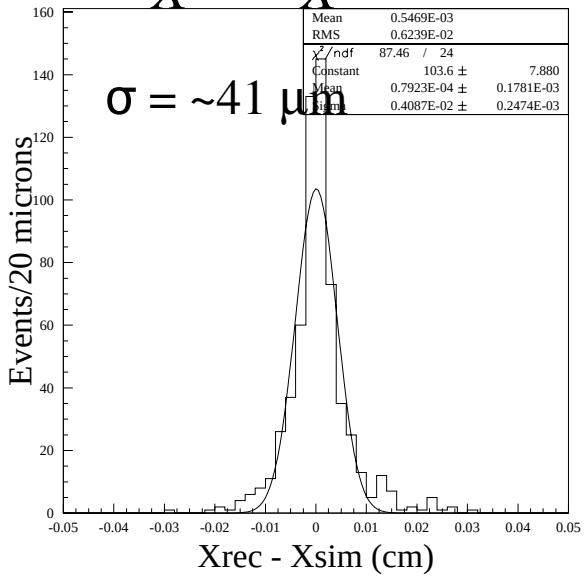


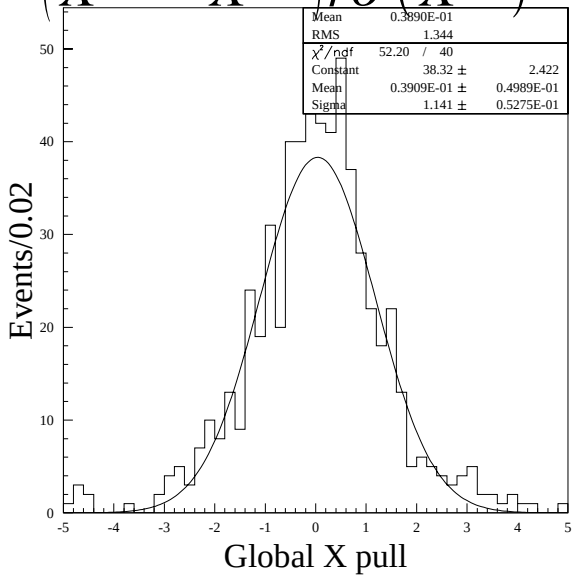


$$X^{Rec} - X^{MC}$$

$$\left(X^{Rec} - X^{MC}\right)/\sigma\left(X^{rec}\right)$$

with constraints:





Further developments

- Parametrization-independent version to be finished in the near future.
- LMS minimization with penalty function could easily be integrated
  - Allows application of soft constraint:
    - parameters are greater/smaller than a given value
    - parameters are distributed according to a given PDF
  - Lagrange multipliers allow only application of "hard constraints" (parameters constrained to a given value).
- Other vertex reconstruction algorithms to be tried out in the sequential strategies: Robust filters, Gaussian-sum filter, etc...

Conclusions

- Flexible kinematic fit algorithm:
  - global and sequential strategies
  - constraints independent and easy to select and implement
  - any 4-vector like objects can be used as input.
  - parametrization independent
- Navigable decay tree to model the decay
- Tests on B_s decays: improvement of all reconstructed parameters
- Different reconstruction strategies tested: results in agreement within numerical precision