A Kinematic fit and a decay chain reconstruction library

K.Prokofiev, Th. Speer, Physik-Institut Universität Zürich, Switzerland

Abstract

A kinematic fit and decay chain reconstruction library has been implemented in the framework of the CMS reconstruction program. The kinematic fit is based on Least Means Squared minimization with Lagrange multipliers and Kalman filter techniques. The structure and the functionality of the library, the mathematical properties and the implementation are discussed. An example of the application of the library to the reconstruction of the $B_s \rightarrow J/\Psi\Phi \rightarrow \mu^+\mu^-K^+K^-$ decay is presented.

INTRODUCTION

The goal of a kinematic fit is to improve the resolution of experimental measurements, test hypothesis and to find unknown parameters by introducing constraints derived from physics laws into the minimization problem. The kinematic fit package should provide a flexible framework with generic minimization algorithms which do not depend on the constraints, such that the constraint can be chosen by the user. It should also provide an easy mechanism to implement a user-defined constraints.

The kinematic fit package implemented in the CMS reconstruction program allows the reconstruction of the full decay chain according to a user-defined model. The full decay tree is reconstructed by fitting all tracks and vertices in the current decay applying user-defined constraints on their parameters.

For instance, the set of constraints to apply during the reconstruction of the decay $B_s \rightarrow J/\Psi\Phi \rightarrow \mu^+\mu^-K^+K^-$ can be the following:

- All four final state tracks come from the B_s decay vertex.
- The invariant mass of two muons should be equal to the mass of the J/Ψ meson.
- The B_s momentum vector reconstructed at the decay vertex should point towards the primary interaction vertex, since we can assume that the B_s meson was produced there.

DECAY CHAIN RECONSTRUCTION

During a kinematic fit, a decay is reconstructed from ”bottom” to ”top”, i.e. from the final state (for example, tracks reconstructed in the tracker) to the mother state (the B_s meson in our example). A decay tree is created as a result of the reconstruction and contains all the relevant information. It describes therefore one hypothesis. In case different competing hypotheses exist for the desired decay channel in an event such as in the case of combinatorial problem, a collection of trees can be created each representing one hypothesis. The total χ^2 of the fit can then be used as a selection criteria.

The decay tree is described by the three following classes: *KinematicParticle*, *KinematicVertex* and *KinematicTree*. The first two store the information about particles and vertices, and the latter one stores the actual position of every particle and vertex in the decay and provides a navigation mechanism between these.

KinematicTree

The *KinematicTree* class describes a decay chain hypothesis. The class contains pointers to all particles and vertices forming the decay chain and allows the user to navigate between them.

Several different trees can be merged by constraining their top particles to a common vertex. A new tree object, containing all available information will then be created.

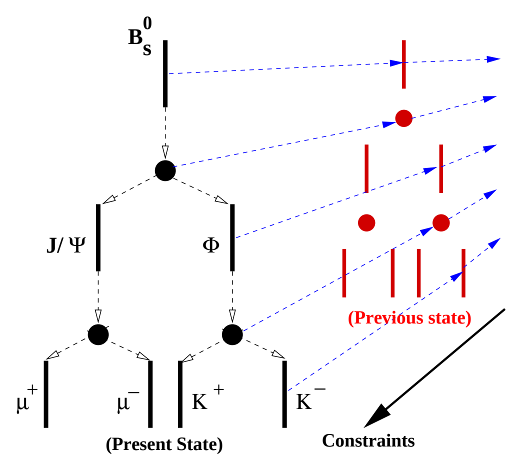

Figure 1: Distribution of the B_s mass residuals for the global fit.

In the fig. 1 the structure of the reconstructed *KinematicTree* for the $B_s \rightarrow J/\Psi\Phi \rightarrow \mu\mu KK$ decay is shown. Every tree component caches pointers to its previous states and constraints applied, as well as the pointer to the current tree the component belongs to.

The internal structure of the tree is based on a graph library. The graph is a private data member of the *KinematicTree* class. It has reference counted *KinematicParticles* as edges and reference counted *KinematicVertices* as nodes.

The navigation in the tree is handled by an internal graph library pointer, which is a private data member of the *KinematicTree* class. The actual graph structure of the class is hidden from the user, and a set of public methods allows the user to navigate in the tree and access the relevant information.

KinematicParticle

This class represents a particle in a decay chain. The class is designed to store the parameters describing the particle (reconstructed trajectory state and assigned or reconstructed mass), the corresponding covariance matrix, the charge, the χ^2 and the number of degrees of freedom assigned to the current particle during previous stages of the reconstruction. It also stores the last constraint applied and the state of the particle before that constraint had been applied.

The initial *KinematicParticles* can be created from any 4-vector-like object with the adequate adapter. For example, reconstructed trajectories can be used by assigning a mass hypothesis, since the CMS detector has no particle identification system. During the kinematic fit, *KinematicParticles* are created by the fitters when a decayed state is reconstructed from its decay products.

In the present version of the library, a particle state is described in a "quasi-cartesian parametrization" by 7 parameters: a reference position in the global frame (x, y, z), the momentum at this point (p_x, p_y, p_z) and the mass of the particle m (calculated as a result of previous fits or hypothesis assigned to the state). This parametrization should be exclusively used when deriving constraint equations and implementing the corresponding classes. The vector of 7 parameters and their joint (7 × 7) covariance matrix are cached in the special *KinematicState* class.

An extended helix "perigee" parametrization, where the first five parameters $(\rho, \theta, \phi, \varepsilon, z_{tr})$ are identical to the usual perigee parameters [3] and the sixth parameter is the particle mass, is also provided.

Both parametrizations can be used for either neutral or charged particles. For neutral particles the trajectory is a straight line. In the extended perigee parametrization, the transverse curvature is replaced with the inverse of the transverse momentum of the particle $(1/p_T)$.

KinematicVertex

This class stores the vertex information needed for the kinematic fit: position, covariance matrix, χ^2 and number of degrees of freedom. As the *KinematicParticle* class, *KinematicVertex* contains references to its previous states. A *KinematicVertex* is always produced as a result of a vertex fit of a set of *KinematicParticles*.

KINEMATIC FITTING

Several requirements drove the design of the kinematic fit library. The minimization algorithm must be independent of the constraints and flexible enough to incorporate arbitrary constraints, such that it can be used in different physical analysis with their different requirements. The development of new constraints has therefore to be simple and independent of the core of the library, such that they can be implemented by a user and shared.

The mathematical approach most widely used in kinematic fitting is the Least Means Squared minimization (LMS) with Lagrange multipliers. This method allows to constrain certain parameters to precise values ("hard" constraint: particle has a given mass, given momentum vector, etc...). Soft, or inequality, constraints (particle mass lies in a certain region, is distributed according to a given *pdf*, etc...), can be imposed using other methods, such as LMS with penalty functions [5]. A complete mathematical description of the methods used in our package can be found in [1],[5] and [6].

One of the advantages of the Lagrange multipliers method is that, when the constraint equations are linear, the minimization problem can be solved analytically. In addition, most non-linear constraints can be linearized (first order Taylor expansion, for example). The minimization algorithm then becomes independent of the particular constraint equations and can be implemented as a standalone software package, where the choice of the constraint equations and linearization points is left to the user. When the set of constraints $H(y^{ref}) = 0$ is linearized around the expansion point y_{exp}, it can be written as:

$$\frac{\partial H(y_{exp})}{\partial y}(y - y_{exp}) + H(y_{exp}) = D\delta y + d = 0, \quad (1)$$

where D is the matrix of derivatives and d the vector of the values of the constraint equations at the expansion point.

The χ^2 function to minimize

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

has then a unique analytical solution (y^{ref}, λ), given in terms of input data y, their covariance matrix V_y, the vector of Lagrange multipliers λ and user-defined quantities D and d.

A constraint can therefore be implemented by writing a single class which inherits from the abstract base class *KinematicConstraint*. This class stores the matrix of derivatives D and the vector of values d for the given constraint. It is easy to see that introducing additional constraint equations in the minimization problem is equivalent to adding lines to the D matrix and elements to the d vector. The matrix of derivatives and the vector of values must be of size $(N^{par} \times N^{eq})$ and (N^{eq}) respectively, where N^{par} is the number of the input parameters, and N^{eq} is the number of constraint equation. A special class (***MultipleKinematicConstraint***) is provided to add constraints together, which collects and assembles the individual contributions into a unique matrix of derivatives and vector of values.

Kinematic fit strategies and factorization of constraints

Two different strategies can be used when reconstructing a decay chain using the Kinematic fit. The most intuitive way is to refit the input data with all the constraints and to find all unknown parameters in a single fit (*Global strategy*). For example, the parameters of the decayed B_s meson can be calculated after a single fit where the two muon and two kaon tracks are constrained to come from a single vertex, and requiring the invariant mass of the two muons to be equal to the mass of the J/Ψ and the sum of the momenta of the four tracks to point towards the primary vertex.

The application of additional constraints during vertex reconstruction is possible using LMS with Lagrange multipliers. The simultaneous application of the constraints introduces $2 \cdot N^{track}$ equations for the vertex constraint (two equations per track: longitudinal and transverse impact parameters), in addition to the user-specified constraints. This algorithm was implemented in the specialized fitter (*KinematicConstrainedVertexFitter*), which handles the extension of the matrix of derivatives D and the vector of values d automatically. The fitter takes the vector of *KinematicParticles* and a *KinematicConstraint* class as input. As it was mentioned before, several constraint can be applied in the same fit.

However, a different strategy can be used. The two final state muons can first be fit to a vertex, reconstructing thus the J/Ψ parameters at this vertex, then constraining the mass of this intermediary state to be equal to the mass of the J/Ψ. After that, the J/Ψ and the two kaons would be fit to a vertex, reconstructing the B_s parameters at this vertex, and the pointing constraint would finally be applied on the B_s. Here the application of the set of constraints in a single fit is substituted with a series of individual fits, applying the constraints one by one on individual, reconstructed particles (*Sequential strategy*). The usage of the sequential fits becomes especially important when working with unstable particles with significant lifetimes, were reconstructed state are to be propagated inside the detector.

This approach is possible due to a remarkable property of the LMS-based algorithms: the result of simultaneous application of the set of constraints is mathematically equivalent to their sequential application. The result of a global fit with a set of constraints is therefore equal to the result of a series of sequential fits, where each constraint is applied individually [1], [7].

The actual implementation of the sequential strategy consists of two classes. The *KinematicParticleVertexFitter* fits a set of *KinematicParticles* to a common vertex with one of the vertex fitters already implemented in the reconstruction code, such as a Kalman filter [4]. The *KinematicParticleFitter* class refits the parameters of the given set of *KinematicParticles* without vertex fit, using the LMS with Lagrange multipliers technique.

Both *KinematicParticleVertexFitter* and *KinematicConstraintVertexFitter* return a fully consistent *KinematicTree* as an output. This tree consists of the refitted input particles, the fitted vertex and the decayed particle, which parameters were calculated after the vertex reconstruction. In case the input particles were already reconstructed as a result of previous fits, their daughter trees are also included in the resulting tree.

The calculation of the momentum of the new particle is done by summing the momenta of the refitted decay products at the vertex position. Its covariance matrix is calculated from the full particle-to-particle covariance matrix taking all correlations into account. A set of special Jacobians is developed to calculate each block of that covariance matrix (position, momentum and position-momentum correlation) and the mathematical technique is in general identical to one used in [2]. In the sequential strategy, the constraint on the mother particle modifies its state which is updated in the tree.

The analytical solution of an LMS minimization does not depend on the parametrization used to describe the input data [1], [4], [5]. Some nonlinear constraint equations may become linear after a change of the parametrization. The constraint equations must however be derived in the same frame as the input data. The global fit requires that the vertex constraint and the additional constraints are derived in the same reference frame. To ensure compatibility between all the constraints, the use of extended the cartesian parametrization is required. In the sequential strategy, since the constraints are applied consecutively, independently of each other, these may be defined in different reference frames. This gives the user the possibility to chose the reference frame where the input data is described and the constraint equations are defined.

EXAMPLE OF B_S DECAY RECONSTRUCTION

The current example is based on the sample of 1000 $B_s^0 \rightarrow J/\Psi\Phi \rightarrow \mu^+\mu^- K^+K^-$ signal events. The final state tracks are selected by associating the reconstructed tracks to the Monte Carlo particles. For the primary vertex, needed for the pointing constraint, the true position is taken from the Monte Carlo information. No selection cuts are applied. The most important parameters for a further analysis are the mass of the reconstructed B_s and the position of its decay vertex.

As a baseline for comparison we use the reconstruction strategy based on the standard Kalman filter: all tracks are fitted to a common vertex using a Kalman filter, and the parameters of the B_s are calculated at the secondary vertex from the refitted final state tracks. No further constraint is applied.

The strategy for the kinematic fitting is the following: the parameters of the decayed B_s meson are calculated after fitting the two muon and two kaon tracks to the same vertex, requiring the invariant mass of the two muons to be equal to the mass of the J/Ψ, in one global fit. The pointing constraint is then applied on the B_s.

In the fig.2 the distribution of the residuals of the invariant mass of the reconstructed B_s meson is shown $(M_{B_s}^{rec} - M_{B_s}^{PDG})$. When fitting this distribution with a Gaussian, the mean is approximately 15 MeV higher than the world-average mass of the B_s, and the standard deviation (which we quote as the resolution), is about 32 MeV. It is believed that the shift of the distribution comes from

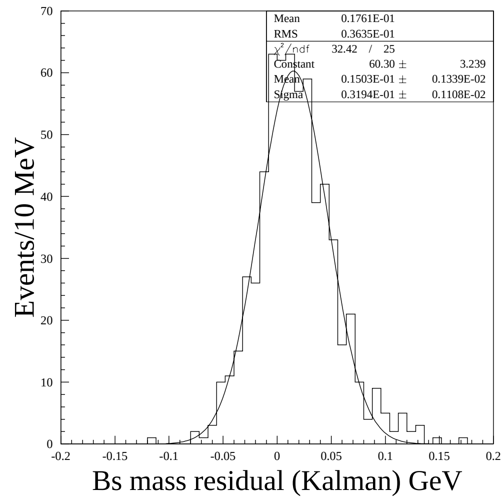

Figure 2: Distribution of the B_s mass residuals for the Kalman filter

inhomogeneities of the magnetic field not properly taken into account during the reconstruction of the muon tracks. The resolutions of the x-coordinate of the secondary vertex is of $47\ \mu m$.

The distribution of the residuals of the invariant mass of the B_s for the global strategy is presented in the fig. 3. The resolution on the reconstructed mass is 13.5 MeV and the mean is displaced for approximately 4 MeV. The

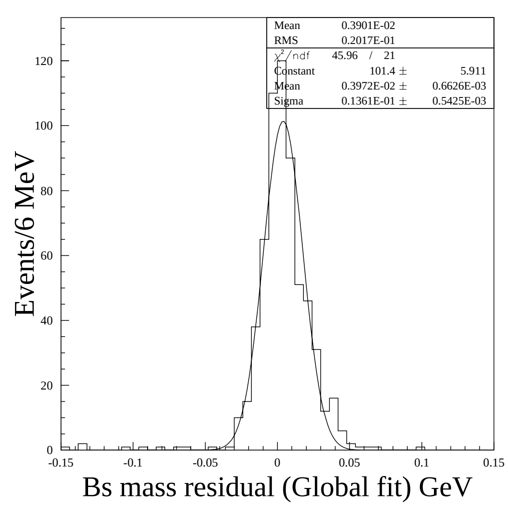

Figure 3: Distribution of the B_s mass residuals for the global fit.

transverse coordinates resolution of the secondary vertex improve with respect to the pure Kalman case. The RMS becomes smaller and the width of the core, when fitted with a Gaussian, improves as well ($\sim 40\ \mu m$).

CONCLUSION

A Kinematic Fit package using LMS with Lagrange multipliers method for the CMS reconstruction framework was implemented. A mechanism to model decay chains by trees was created. The library is flexible enough to incorporate any constraint provided by the user. The constraints are linearized and parametrized with two quantities, and therefore, easy to implement. The possibility of both a global and a sequential application of the constraints is provided.

The package is flexible enough to use any 4-vector like object as an input and to incorporate any vertex fitting algorithm implemented in the CMS reconstruction framework. The structure of the library also foresees the implementation of the new constrained fitting algorithms, for example, the LMS minimization with penalty functions.

The package was tested on the reconstruction of the decay $B_s \rightarrow J/\Psi\Phi \rightarrow \mu^+\mu^-K^+K^-$ and validated. An improvement of all reconstructed parameters is seen with respect to a simple vertex fit with the Kalman filter. Several other reconstruction strategies (sequential with one or several neutral state propagations) were studied, and their results were found to be in agreement with the result of the global strategy within numerical precision.

REFERENCES

[1] P.Avery ”Applied Fitting Theory VI: Formulas for Kinematic Fitting” CBX 98-37, June 9, 1998.

[2] P.Avery ”Applied Fitting Theory VII: Building Virtual Particles” CBX 98-38, June 8, 1998.

[3] P.Billoir, S.Quian ”Fast vertex fitting with a local parametrization of tracks” NIM in PR A311 (1992) 139-150

[4] R.Frühwirth et al. ”Vertex reconstruction and track bundling at the LEP collider using robust algorithms” Computer Physics Communications 96 (1996) 189-208

[5] A.G.Frodsen et al. ”Probability and statistics in particle physics” Universitetsforlaget, 1979

[6] R.Frühwirth et al. ”Data analysis techniques for high-energy physics” Cambridge University Press 2000

[7] P.Avery ”General Least Squares Fitting Theory” CBX 91-72, October 18, 1991