KFParticle

Ivan Kisel[1,2], Igor Kulakov [1,3],
Iourii Vassiliev [1,2], Maksym Zyzak[1,3]

1 – Goethe-Universität Frankfurt, Frankfurt am Main, Germany
2 – GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany
3 – Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Third International Workshop for Future Challenges in Tracking and Trigger Concepts

29.02.2012

Content

- Concept of the KFParticle package
- Kalman filter and KFParticle mathematics
- Functionality of the KFParticle package
- SIMDized KFParticle
- Examples of the code with KFParticle
- Particles finding with the SIMDized KFParticle package

Principal Concept

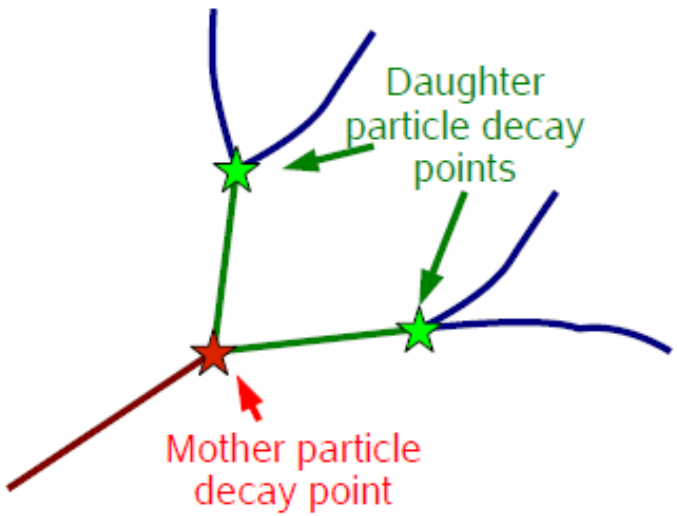
- Direct estimation of the parameters of the decayed particle
- The KFParticle object has the parameters as the physical particle – x, y, z coordinates and p_x, p_y, p_z components of the momentum
- The parameters do not depend on the track models and the experiment geometry
- Simple access to the most interesting parameters of the particle: mass, momentum, lifetime, decay length
- Construction mother particles from tracks or other particles
- Reconstruction of decay chains

Reconstruction of decay chains

State vector

Position, momentum and energy

$$\mathbf{r} = \{ x, y, z, p_x, p_y, p_z, E \}$$

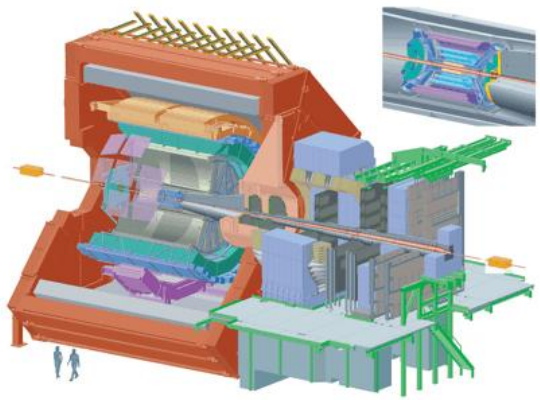


- The state vector does not depend on the number of daughter particles
- All particles are described with the same state vector
- All particles are equivalent and functionality is the same for mother and daughter particles

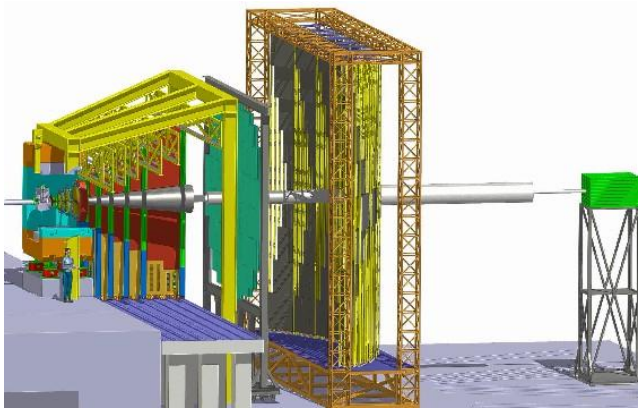
Experiments

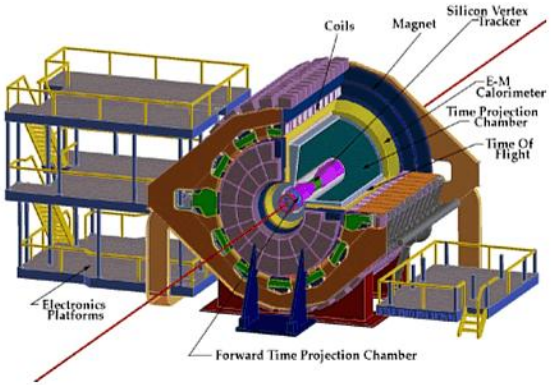
KFParticle is developed based on the ALICE and CBM experiments. In progress in the STAR experiment

ALICE (CERN, Switzerland) – a collider experiment



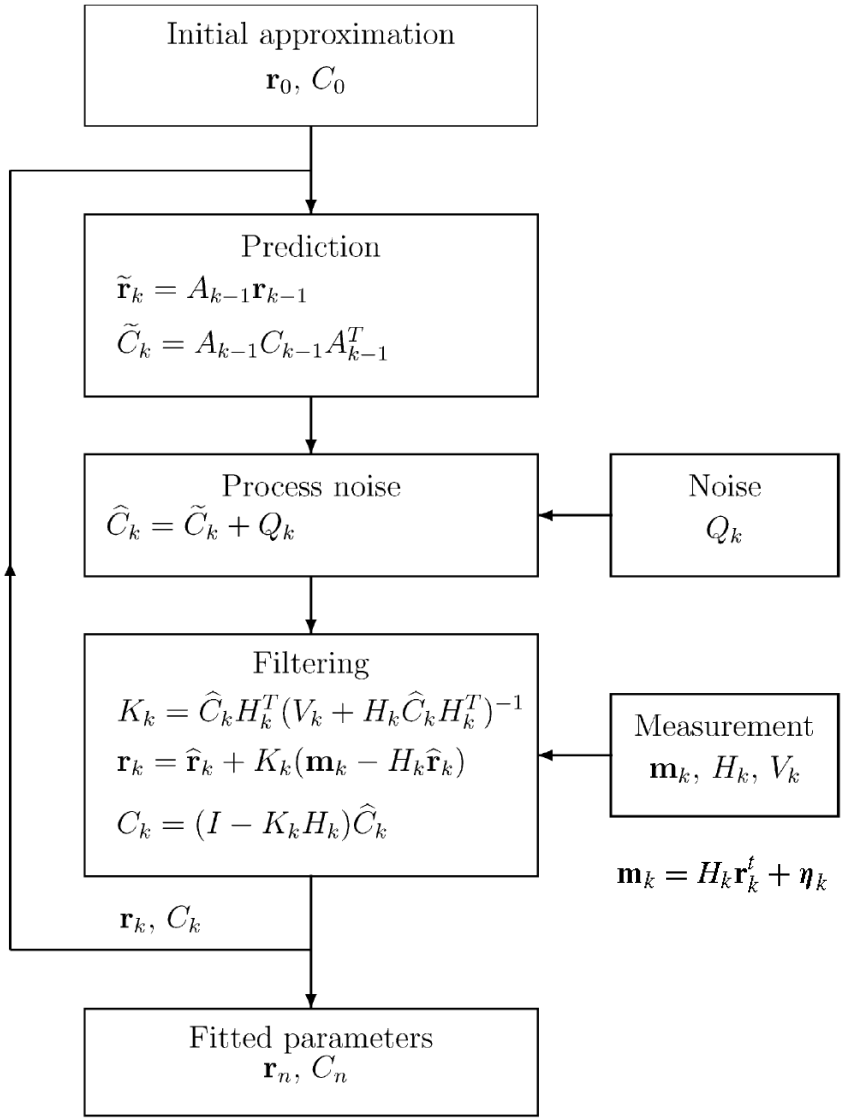
CBM (FAIR, Germany) – a fixed-target experiment





STAR (RHIC, USA) – a collider experiment

Kalman Filter Mathematics



- r, C – optimum estimation of the state vector and its covariance matrix
- A – propagation operator
- m, V – measurement and its covariance matrix
- H – measurement model
- K – gain matrix
- Assumptions:
  - Noise and errors are unbiased and uncorrelated
  - A and H – linear operators

KFParticle Mathematics

Initial approximation
$\mathbf{r_0}$, C_0

Fitted parameters before a kth iteration
$\mathbf{r_{k-1}}$, C_{k-1}

Transport of the daughter to the decay point, construct parameters

$$\mathbf{m}_k \equiv \begin{pmatrix} \mathbf{v}_k^d \\ \underline{\mathbf{p}}_k^d \end{pmatrix} = \mathbf{r}_k^d, \qquad \text{(already transported)}$$

$$V_k \equiv \begin{pmatrix} V_k^v & V_k^{vp\,T} \\ V_k^{vp} & V_k^p \end{pmatrix} = C_k^d + \begin{pmatrix} \mathbf{p}_k^d \\ \mathbf{p}_k^d \times B \cdot q_k \\ 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_k^d \\ \mathbf{p}_k^d \times B \cdot q_k \\ 0 \end{pmatrix}^T \cdot \sigma_s^2$$

Filtering a daughter particle

$$\begin{aligned}
S_k &= \left(C_{k-1}^v + V_k^v\right)^{-1}. \\
\mathbf{r}_k &= \begin{pmatrix} \mathbf{v}_{k-1} + C_{k-1}^v S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right) \\ \underline{\mathbf{p}}_{k-1} + \underline{\mathbf{p}}_k^d + \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right) \end{pmatrix}, \\
C_k &= \begin{pmatrix} C_{k-1}^v - C_{k-1}^v S_k C_{k-1}^v & C_{k-1}^{vp\,T} - C_{k-1}^v S_k \left(C_{k-1}^{vp} - V_k^{vp}\right)^T \\ C_{k-1}^{vp} - \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k C_{k-1}^v & C_{k-1}^p + V_k^p - \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k \left(C_{k-1}^{vp} - V_k^{vp}\right)^T \end{pmatrix} \\
\chi_k^2 &= \chi_{k-1}^2 + \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right)^T S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right), \\
\mathrm{ndf}_k &= \mathrm{ndf}_{k-1} + 2.
\end{aligned}$$

Fitted parameters after a kth iteration
$\mathbf{r_k}$, C_k

Fitted parameters
$\mathbf{r_n}$, C_n

$$C_{k-1} \equiv \begin{pmatrix} C_{k-1}^v & C_{k-1}^{vp\,T} \\ C_{k-1}^{vp} & C_{k-1}^p \end{pmatrix}$$

$$V_k \equiv \begin{pmatrix} V_k^v & V_k^{vp\,T} \\ V_k^{vp} & V_k^p \end{pmatrix}$$

$$S_k = \left(C_{k-1}^v + V_k^v\right)^{-1}$$

$$\mathbf{r}_k \equiv \begin{pmatrix} \mathbf{v}_k \\ \underline{\mathbf{p}}_k \end{pmatrix}$$

$$\mathbf{r}_k^d \equiv \begin{pmatrix} \mathbf{v}_k^d \\ \underline{\mathbf{p}}_k^d \end{pmatrix}$$

v – coordinates
p – 4D momentum

The mathematics is described in

S. Gorbunov and I. Kisel, Reconstruction of decayed particles based on the Kalman filter. CBM-SOFT-note-2007-003, 7 May 2007

Structure of KFParticle

- Low level (for developers)
  - Transport functions
  - KF mathematics
  - Constraints
- Intermediate level (for advanced users)
  - Feasibility studies
  - Reconstruction of particles
  - Reconstruction of decay chains
- High level or KFParticle-Light (for users and triggering)
  - Reconstruction of standard decays (K^0_s, hyperons, D^0-mesons,…)
  - Reconstruction of event topology
  - On-line selection of events

Functionality of the package:

- Construction of the particles from tracks or another particles
- Decay chains reconstruction
- Transport of the particles (on the distance, to a point, to another particle, to vertex)
- Calculation of the distance to point, vertex or another particle
- Calculation of the deviation from point, vertex or another particle
- Simple access to the particle parameters and their errors

Functionality in ALICE and CBM

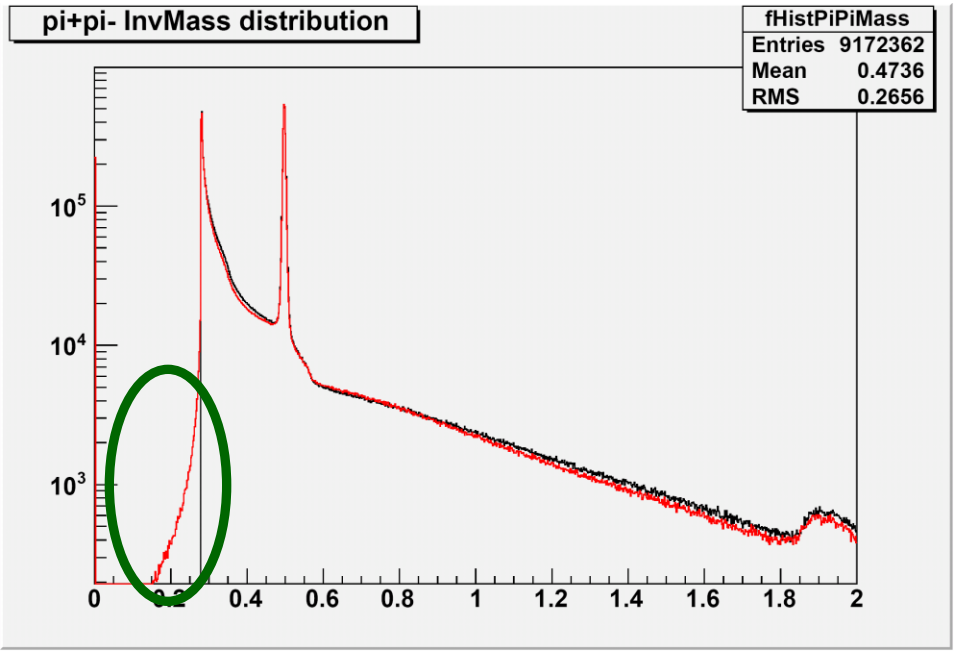
Functions	ALICE	CBM
Construct, SetMassConstraint, SetProductionVertex, SetVtxGuess	+	+
GetMass, GetMomentum, GetDecayLength, GetLifeTime	+	+
GetDecayLengthXY, GetPhi, GetR	+	-
Extrapolate, TransportToProductionVertex(), TransportToDecayVertex()	+	+
TransportToPoint, TransportToVertex, TransportToParticle, TransportToDS,	+	-
GetDStoPoint	+	+
GetDStoParticle, GetDStoParticleXY, GetDistanceFromVertex, GetDistanceFromVertexXY, GetDistanceFromParticle, GetDistanceFromParticleXY, GetDeviationFromVertex, GetDeviationFromVertexXY, GetDeviationFromParticle, GetDeviationFromParticleXY	+	-
GetAngle, GetAngleXY, GetAngleRZ	+	-
SubtractFromVertex, ConstructGamma	+	-
SetNoDecayLength, +=, -=	+	-
Particles finder	-	+

Functionality become more and more advanced

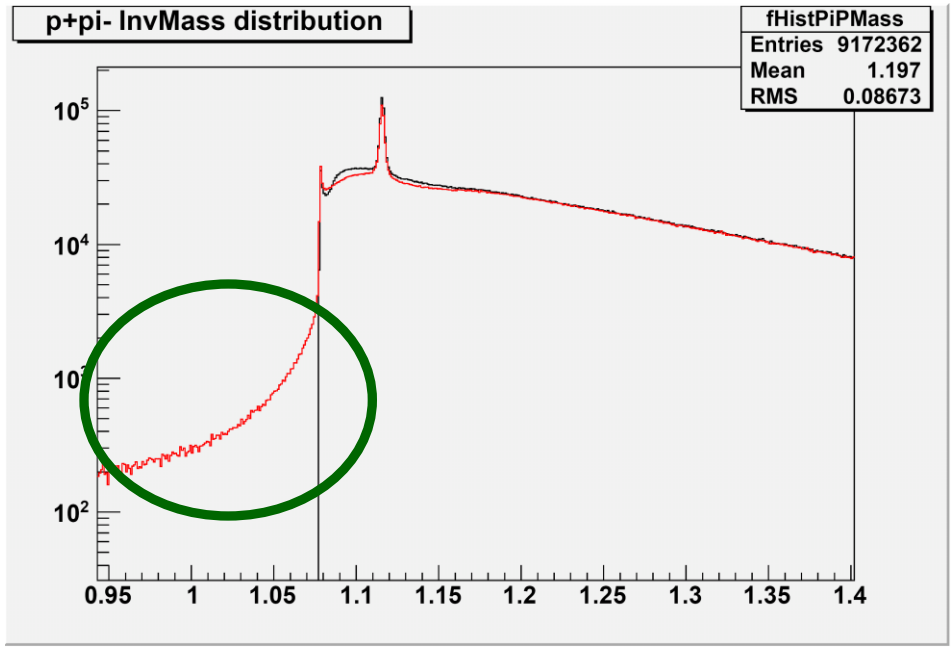
Improvement: Mass Problem of KFParticle

Problem: fit of a particle with the KFParticle package sometimes gives a mass less, then total mass of daughter particles.

Problem is reported by ALICE group.



K^0_s



Λ^0

Pictures are provided by Ana Marin

Improvement: Solution of the Mass Problem

Solutions:

- Energy considered as dependent variable, recalculated from fitted momentum and mass hypothesis:

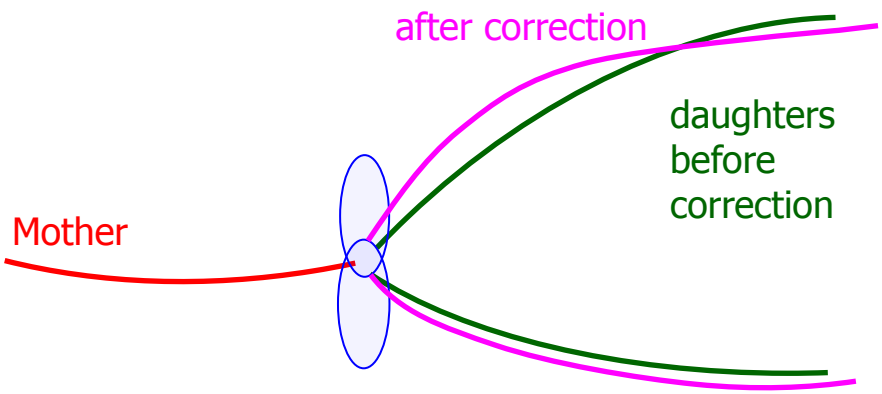
  (Similar approach was implemented in ATLAS, CMS reconstruction and the RAVE package[1])

Fit with KF

Calculate from fitted parameters

$$r = \{ x, y, z, p_x, p_y, p_z, E \}$$

- Set nonlinear mass constraint on the daughter particles before momentum calculation:

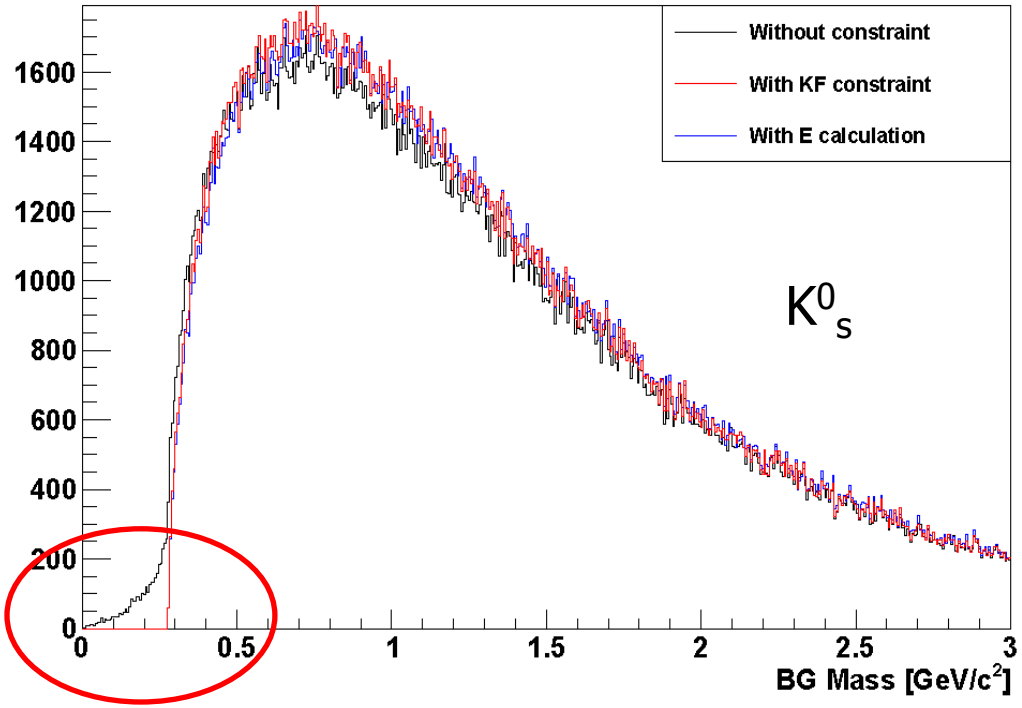


Problem is reported by Ana Marin

R. Fruehwirth, P. Kubinec, W. Mitaroff and M. Regler, Vertex reconstruction and track bundling at the LEP collider using robust algorithms. Comp. Phys. Commun. 96 (1996) 189-208.

Improvement: Comparison of the Solutions for the Mass Problem

- Mass of the background particles has been analyzed. The tail in unphysical region disappeared in both cases
- Both solutions show similar quality (residuals and pulls) for the mass reconstruction of the signal particles
- The solution with E calculation is less stable
- Both solutions are added to the repository. Second solution is a default one



Stability Test			
Reconstruction method	Initial	With KF Constraint	With E calculation
N_{part}, $C_{EE} < 0$	~ 10^{-3} %	~ 10^{-3} %	~ 3 %

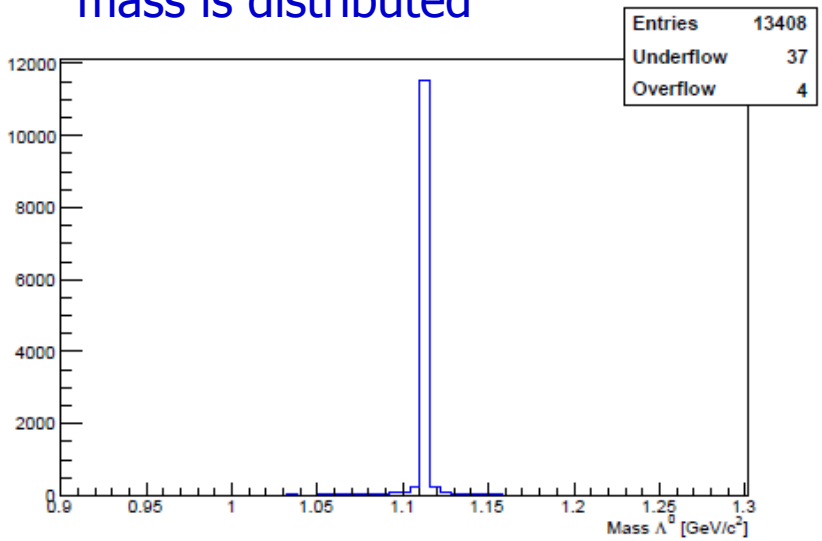
Tests have been performed using ALICE simulated data

Problem is reported by Ana Marin

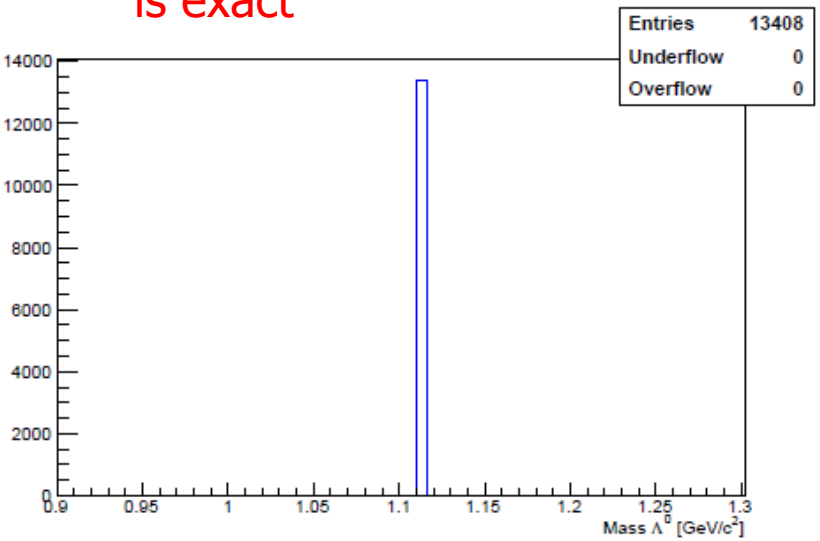
Improvement: Nonlinear Mass Constraint

- Nonlinearities in the Kalman filter and in the extrapolation formulas brake constraints. Before addition of the daughters it is needed to set the correct mass to them
- Linearized constraint on mass is not exact (was in AliKFParticle before)
- Nonlinear mass constrained has been implemented

Linear mass constraint on Λ^0 – mass is distributed



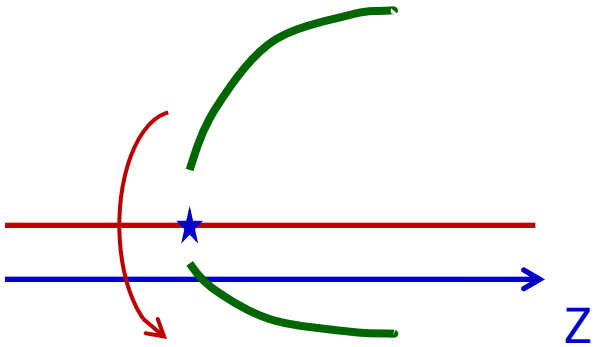
Nonlinear mass constraint is exact



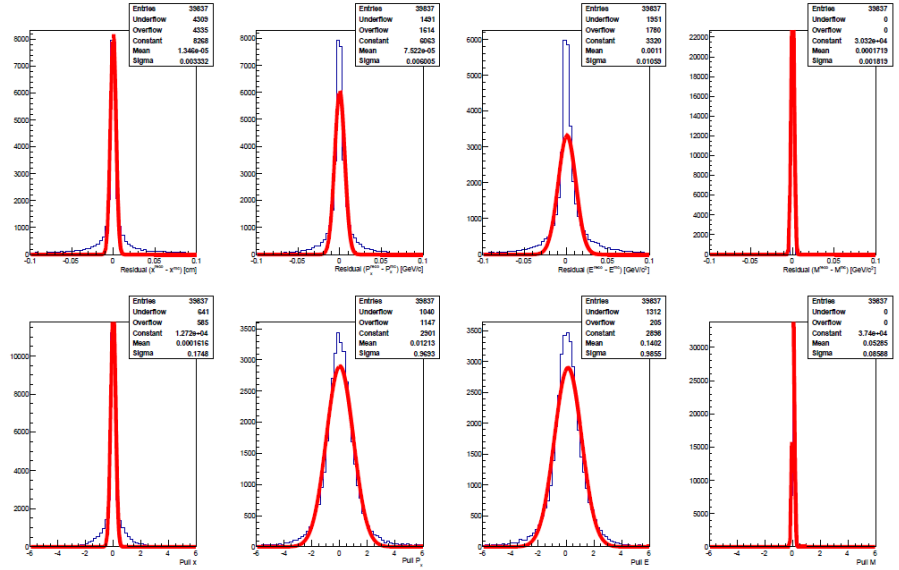
- Nonlinear mass constraint has been added to the AliKFParticle (in the repository now)

Improvement: Rotation of the AliKFParticle

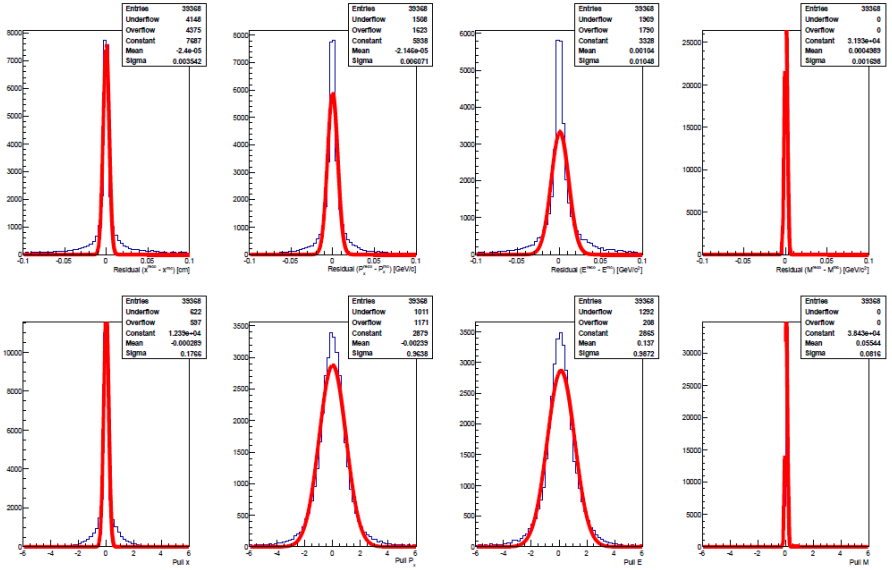
- Is required for background study for J/Ψ and gamma
- A function for rotation of the particle around vertex in XY plane has been added to AliKFParticle (in the repository now)



Before rotation



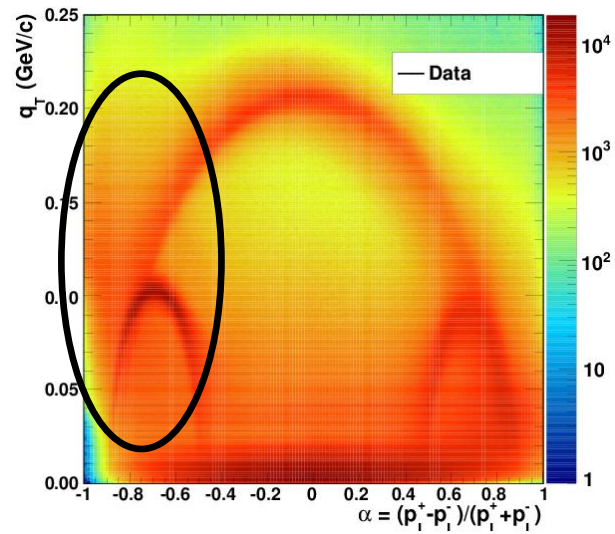
After rotation

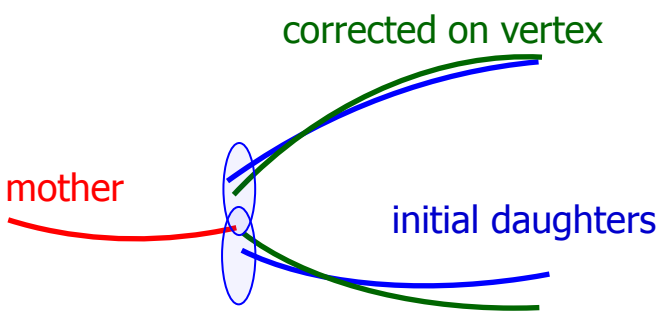


On request of Jens Wiechula

Improvement: Problems with Reconstruction of Gammas

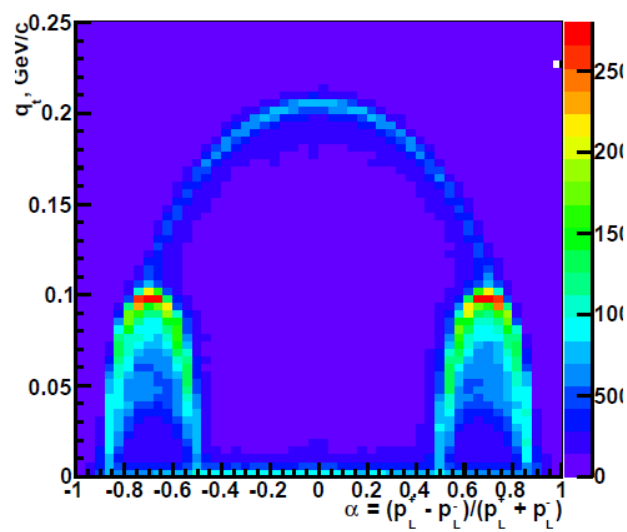
- The plot allows to distinguish different types of V^0 particles
- Used by gamma-conversion group to suppress the background
- Asymmetric Armenteros-Podolanski plot, 2 reasons:
  - from V^0 reconstruction
  - projection on the reconstructed KFParticle object is incorrect: E and P doesn't conserve





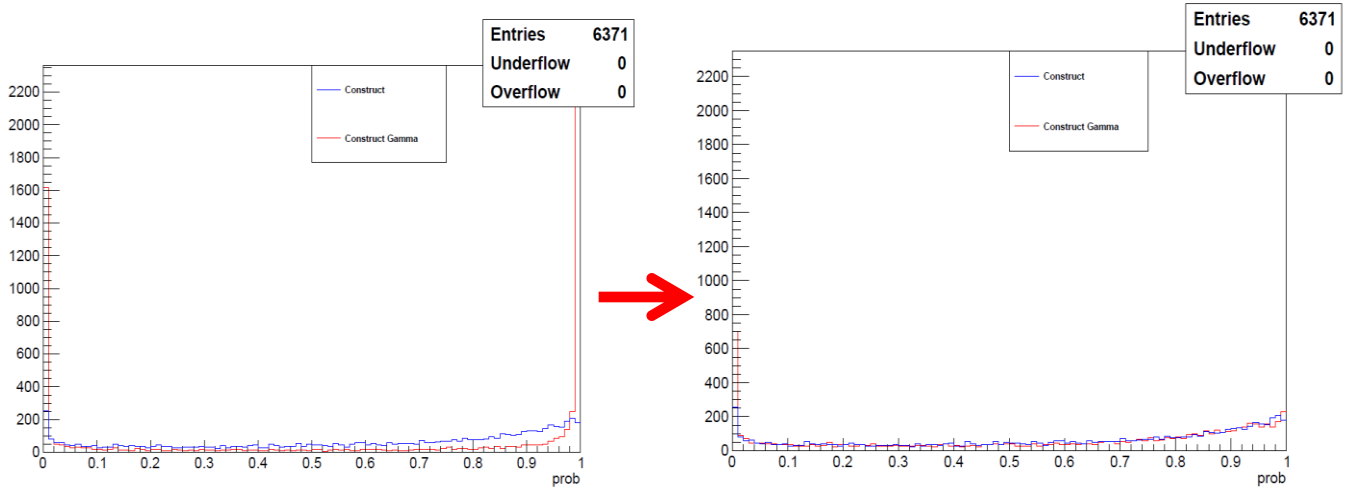
- New function is added to the repository: daughters are projected on the sum of the momentums

On request from Ana Marin, Martin Wilde, Friederike Bock and Daniel Lohner



Improvement: Problems with χ^2 Distribution of ConstructGamma()

- The χ^2 distributions of the usual "Construct()" and "ConstructGamma()" are different
- Prob – probability that the observed χ^2 should be less than an obtained value of χ^2 for the given NDF; prob distribution is flat
- The first correction has been done (not yet in the repository)
  - More accurate initial estimation of the vertex position
  - Corrections in the χ^2 calculation
- Tests are done with MC gammas



- Further improvement are foreseen

On request from Ana Marin, Martin Wilde, Friederike Bock and Daniel Lohner

SIMDized KFParticle

- KFParticle has been SIMDized
- The reconstruction quality is the same for the scalar version and the SIMD version:
  - Λ reconstruction in CBM

	Resolution				Pull			
	M, MeV/c2	X, cm	Y, cm	Z, cm	M	X	Y	Z
Scalar	1.2	0.011	0.015	0.18	1.54	1.50	1.42	1.63
SIMD	1.2	0.013	0.015	0.18	1.54	1.51	1.50	1.69

  - D^0 reconstruction in ALICE (using MC data)

	Resolution				Pull			
	M, MeV/c2	X, cm	Y, cm	Z, cm	M	X	Y	Z
Scalar	18.4	0.012	0.011	0.016	1.16	1.15	1.12	1.12
SIMD	18.5	0.012	0.012	0.016	1.19	1.16	1.15	1.11

- Speedup factor of 5 for CBM and 3 for ALICE has been achieved

Examples

Scalar version

```
AliKFParticle P1, P2;
P1 = AliKFParticle( *pTrack, PDG );
AliKFParticle V0( P1,  P2);
```

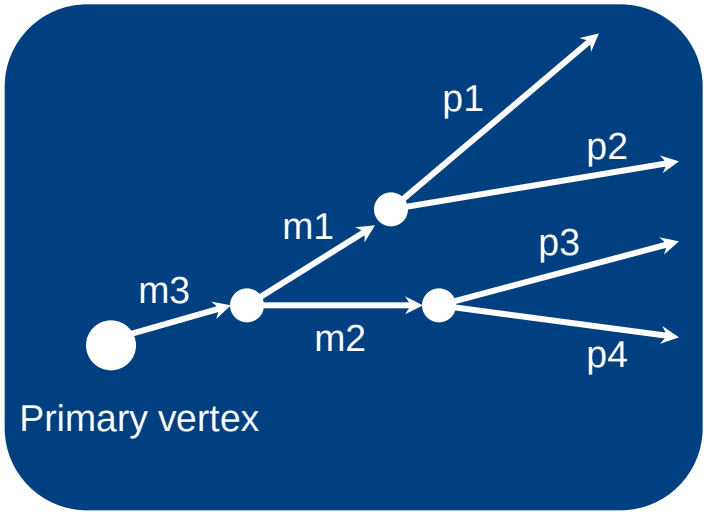
```
Double_t length, sigmaLength;
V0.GetDecayLength( length, sigmaLength ) ;
Double_t mass, sigmaMass;
V0.GetMass(mass, sigmaMass ) ;

TH1F *MassDistribution;
…
MassDistribution->Fill(mass[i]);
```

SIMD version

```
AliKFParticle P1[fvecLen], P2[fvecLen];
for(int i=0; i<fvecLen; i++)
{
   P1[i] = AliKFParticle( *pTrack, PDG );
}
AliKFParticleSIMD PartPos( P1, PDG );
AliKFParticleSIMD PartNeg( P2, PDG2 );
AliKFParticleSIMD V0( PartPos, PartNeg );

fvec length, sigmaLength;
V0.GetDecayLength( length, sigmaLength ) ;
fvec mass, sigmaMass;
V0.GetMass( mass, sigmaMass );

TH1F *MassDistribution;
…
for(int i=0;i<fvecLen; i++)
{
   MassDistribution->Fill( mass[i] );
}
```

Example of Decay Chain reconstruction

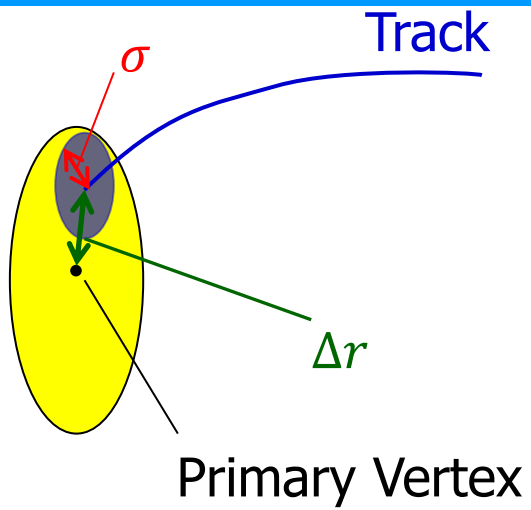
```
AliKFVertex PrimVtx( ESDPrimVtx );                          // Set primary vertex
AliKFParticle p1( ESDp1, pdg1 ), p2( ESDp2, pdg2 ),         // Set daughters
              p3( ESDp3, pdg3 ), p4( ESDp4, pdg4 ) ;

AliKFParticle  m1( p1, p2 ), m2( p3, p4 ), m3( m1, m2 ); // Construct all mothers
PrimVtx += m3;                          // Improve the primary vertex.

m3.SetProductionVertex( PrimVtx ); // m3 is fully fitted
m1.SetProductionVertex( m3 );      // m1 is fully fitted
m2.SetProductionVertex( m3 );      // m2 is fully fitted
p1.SetProductionVertex( m1 );      // p1 is fully fitted
p2.SetProductionVertex( m1 );      // p2 is fully fitted
p3.SetProductionVertex( m2 );      // p3 is fully fitted
p4.SetProductionVertex( m2 );      // p4 is fully fitted
```



Fast Particles Finding with KFParticle

- Implemented for the CBM experiment
- Is planed to be used in the online selection – should be fast
- Based on SIMD KFParticle, SIMD KF track fitter
- Algorithm is developed and tested on decays:
  - $K^0_s \rightarrow \pi^+ + \pi^-$,
  - $\Lambda \rightarrow p + \pi^-$,
  - $\Xi^- \rightarrow \pi^- + \Lambda$,
  - $\Omega^- \rightarrow K^- + \Lambda$,
  - H-dibaryon $\rightarrow \Lambda + \Lambda$.
- All particles are found by one function: only one loop over tracks
- No PID is applied
- Parameters and their errors of particles are mathematically correct
- The user gets also a set of histograms in the end of the task

Reconstruction Algorithm



- Tracks selection strategy:
  - Track is correctly fitted (parameters and covariance matrix are well defined)
  - Track is secondary ($\chi_v > 3$, $\chi_v = \sqrt{\frac{\Delta r^2}{\sigma^2}}$)
- Reconstruction of K^0_s and Λ:
  - All selected negative tracks are combined with positive
  - Parameters are finite
  - $\chi^2/ndf < 3$ of a particle-candidate
- Reconstruction of Ξ^-, Ω^- and H-dibaryon:
  - Secondary Λ are selected: $\chi^2_{topo}/ndf > 5$, $z > 4$, $|M - M_{MC}| < 3\sigma$
  - Particles are reconstructed from selected Λ and tracks
  - Reconstructed particles are selected using criteria:

	χ^2/ndf	χ^2_{topo}/ndf	z
Ξ^-	<6	<5	>3
Ω^-	<3	<3	>3
H-dibaryon	<3	<3	>3

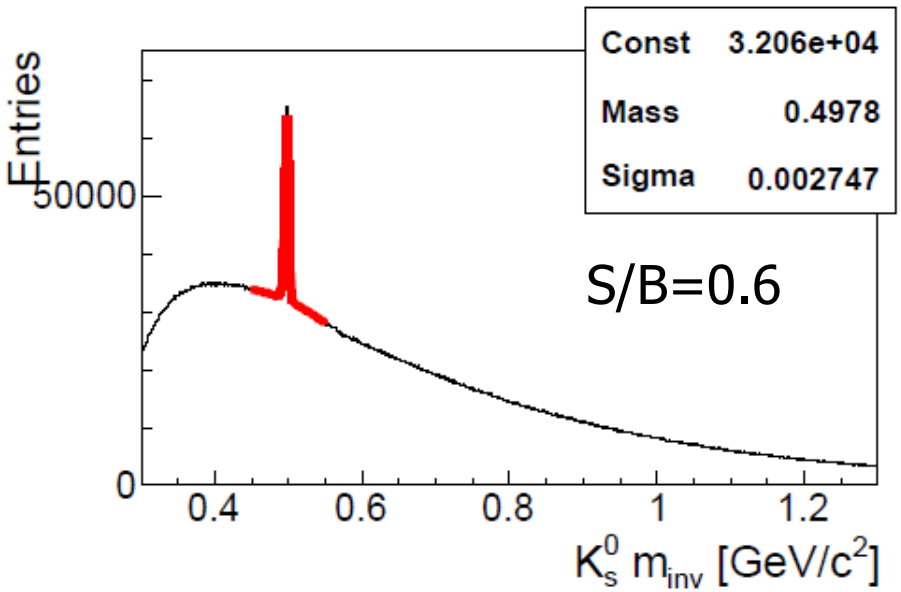
Characteristics of Particles Finding 1

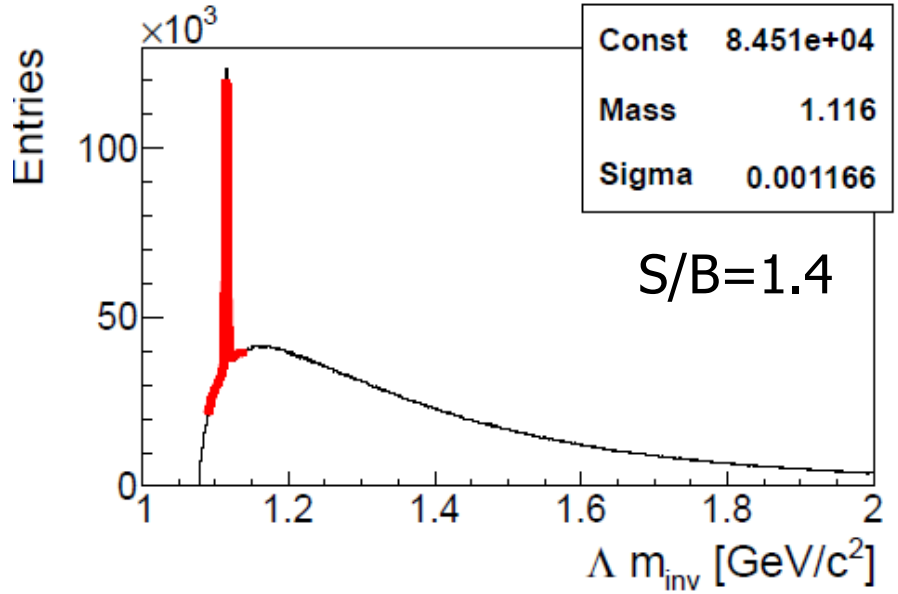
		Efficiency, % (normalized on all MC particles)	Efficiency, % (both daughters reconstructable)	Efficiency, % (both daughters reconstructed)
K_s^0	Minbias	26.5	44.0	64.5
	Central	24.8	39.9	64.7
Λ^0	Minbias	21.2	35.2	59.2
	Central	20.1	32.0	57.5
Ξ^-	Signal	1.1	-	-
Ω^-	Signal	1.9	-	-

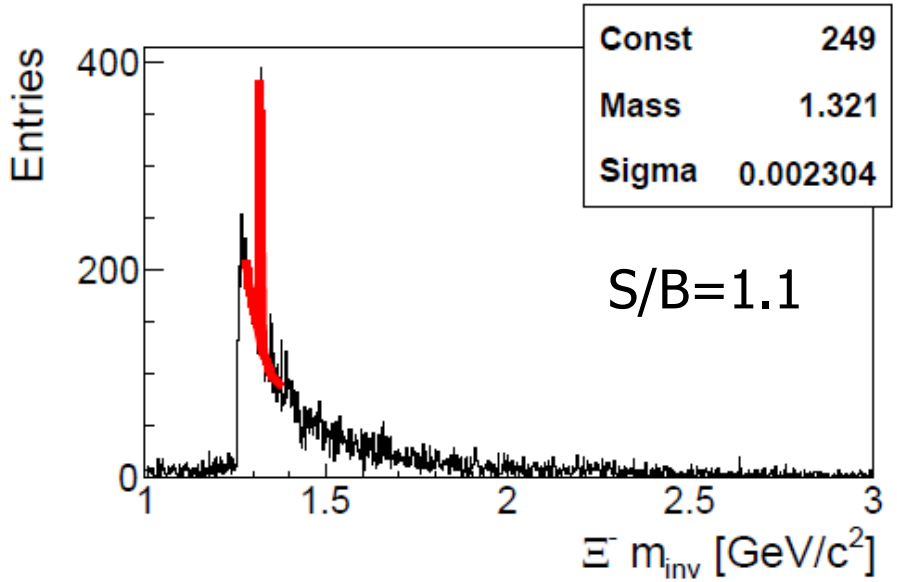
Execution time per event		
	Intel E7-4860 @ 2.27GHz	Intel X5550 @ 2.67GHz
Minbias	4 ms	3 ms
Central	33 ms	23 ms

Characteristics of Particles Finding 2

80 central kEvents Au+Au at 25AGeV

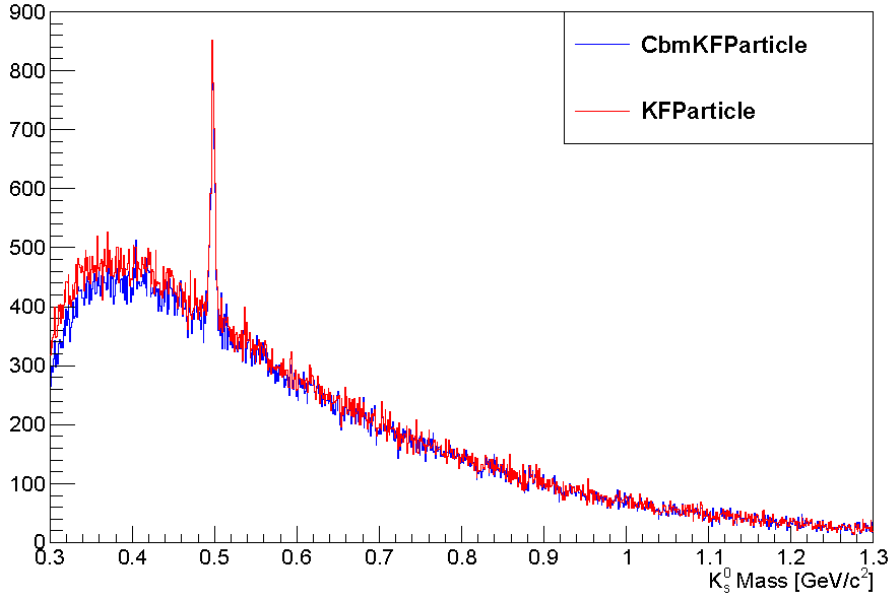


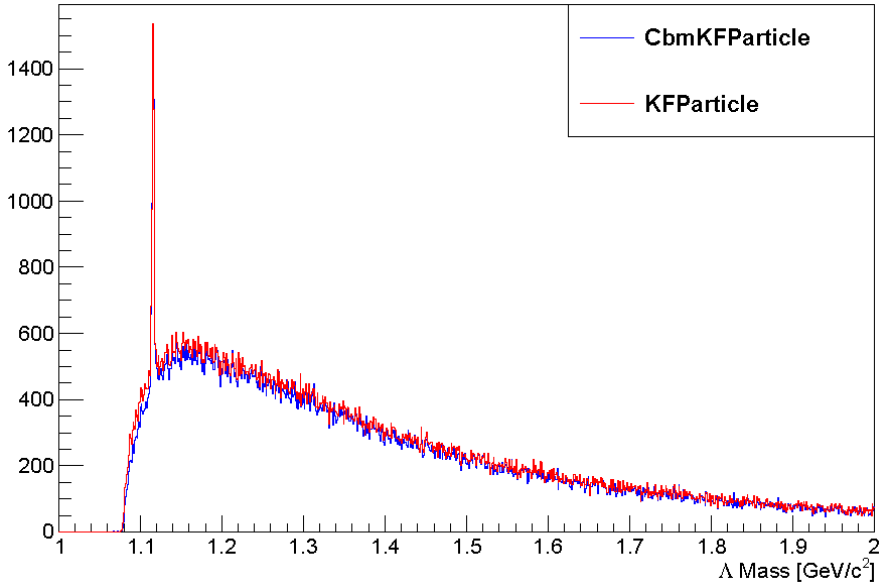




Unification of ALICE and CBM KFParticle

- The unified KFParticle package has been created and tested within the CBMRoot framework
- The unified package allows to add ALICE functionality to the CBM
- The first tests has been done using CbmV0Analysis





- The unified KFParticle and CbmKFParticle show similar results
- Further tests of the package functionality will be done

Summary

- The KFParticle package is a particle reconstruction package with a rich functionality. The functionality become more and more advanced.
- KFParticle has been SIMDized. SIMDized version shows the same results.
- The first steps for a unification of the KFParticle have been done.
- The particles finder has been developed based on the SIMDized KFParticle package. The algorithm shows high speed and efficiency.

Future Plans

- Increase the functionality of the package, create the KFParticle library.
- Add KFParticle to the STAR experiment and STAR HLT.
- Further development of the KFParticle package using real data from the ALICE (CERN) and STAR (RHIC) experiments.
- Implement statistical methods for the particle reconstruction and selection based on KFParticle.
- Add user code for the physics analysis to KFParticle and help to speed up it.
- Reconstruct the full topology of the event within KFParticle.
- Multi vertex events reconstruction with KFParticle.
- Add adaptive methods (DAF, PDAF, etc.) to the KFParticle package.
- Use KFParticle for the 4D tracking developing.