KFParticle

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

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

Second International Workshop for Future Challenges in Tracking and Trigger Concepts

07.07.2011

Content

- Concept of the KFParticle package
- Kalman filter and KFParticle mathematics
- Functionality of the KFParticle package
- SIMDized KFParticle
- Examples of the code with KFParticle
- Implementation of the online particles finder based on the SIMDized KFParticle package
- An approach for multi-vertex event reconstruction

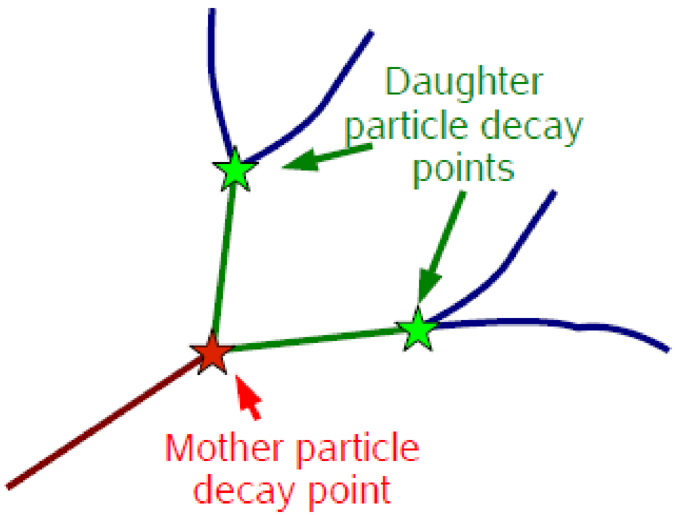
Principal Concept

- 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
- Direct estimation of the parameters of the decayed particle
- 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

State vector

Position, direction and momentum

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



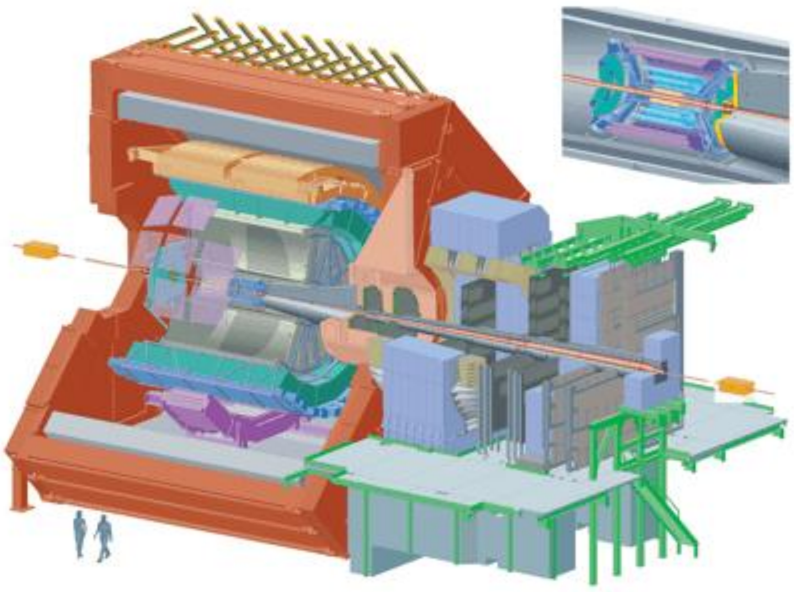
- The state vector does not depend on the number of daughter particles
- All particles are described with the same state vector
- All particles are equivalent
- All functionality is the same for mother and daughter particles

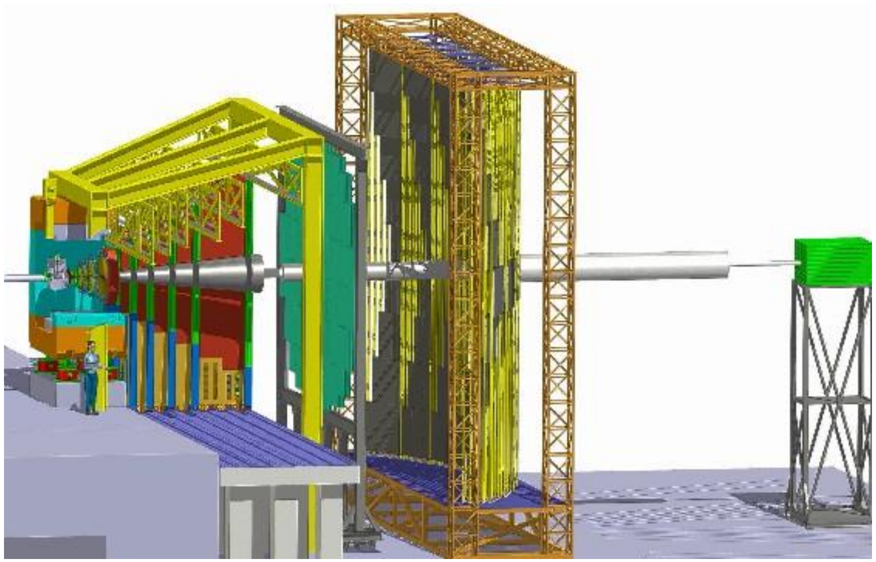
Experiments

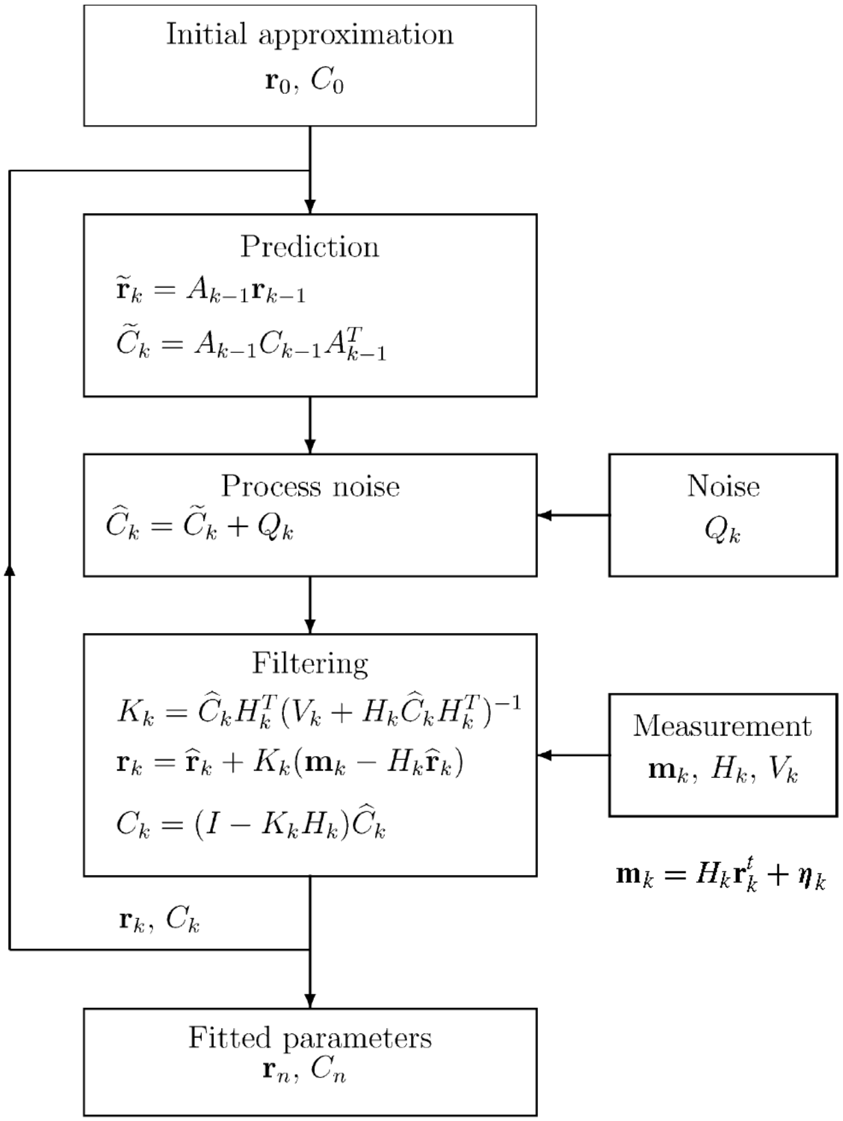
KFParticle is developed based on the ALICE and CBM experiments

ALICE (CERN, Geneva, Switzerland) – a collider experiment

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







- r, C – optimum estimation of the state vector and its covariance matrix
- 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 k^th 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

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

$$\mathbf{r}_k = \begin{pmatrix} \mathbf{v}_{k-1} + C_{k-1}^v S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right) \\ \underline{\mathbf{p}}_{k-1} + \underline{\mathbf{p}}_k^d + \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right) \end{pmatrix} ,$$

$$C_k = \begin{pmatrix} C_{k-1}^v - C_{k-1}^v S_k C_{k-1}^v & C_{k-1}^{vp\,T} - C_{k-1}^v S_k \left(C_{k-1}^{vp} - V_k^{vp}\right)^T \\ C_{k-1}^{vp} - \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k C_{k-1}^v & C_{k-1}^p + V_k^p - \left(C_{k-1}^{vp} - V_k^{vp}\right) S_k \left(C_{k-1}^{vp} - V_k^{vp}\right)^T \end{pmatrix}$$

$$\chi_k^2 = \chi_{k-1}^2 + \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right)^T S_k \left(\mathbf{v}_k^d - \mathbf{v}_{k-1}\right) ,$$

$$\mathrm{ndf}_k = \mathrm{ndf}_{k-1} + 2 .$$

↓

Fitted parameters after a k^th 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

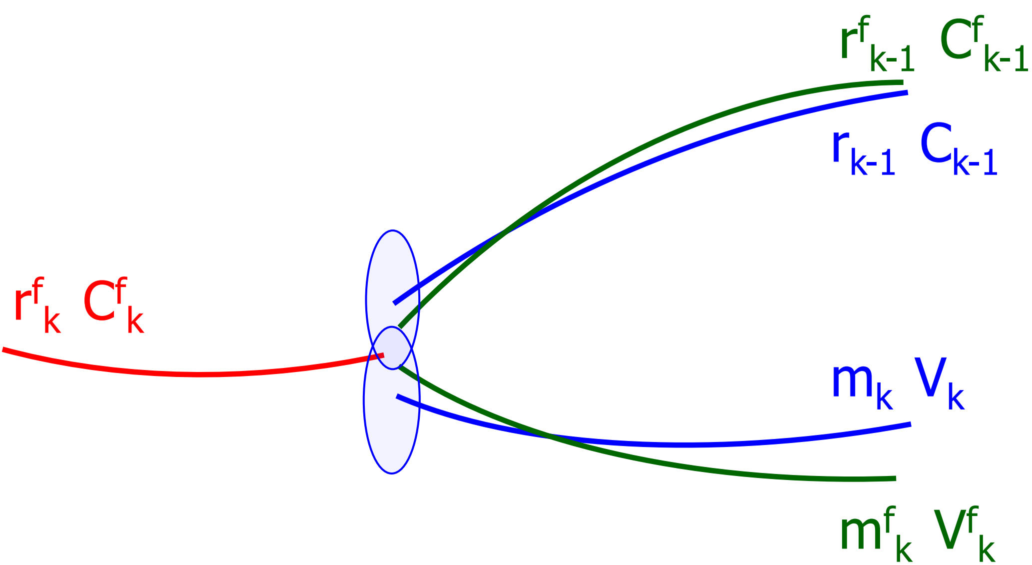
S. Gorbunov and I. Kisel, Reconstruction of decayed particles based on the Kalman filter. CBM-SOFT-note-2007-003, 7 May 2007

Explanation of the Mathematics

State vector — Position, direction and momentum

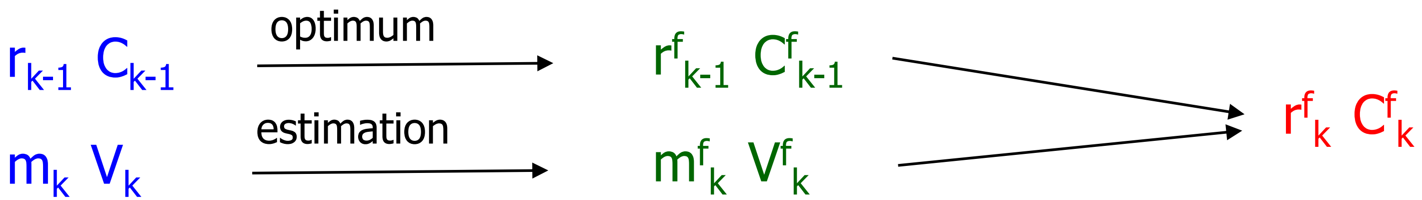
$\mathbf{r} = \{ x, y, z, p_x, p_y, p_z, E \}$

Filtration of the state vector by an optimum measurement is used.



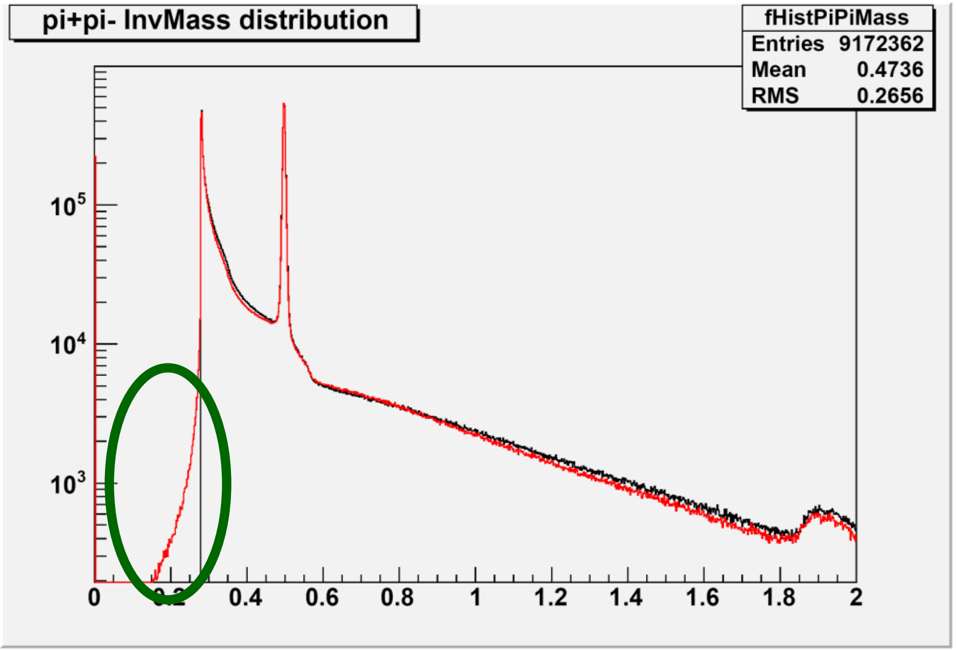
- **p** and E correction on vertex position for daughter particles
- Correction with extended KF (the covariance matrices are taking into account)
- No mass constrain for optimum estimations $\mathbf{r}^f_{k-1}$ and $\mathbf{m}^f_k$ – mass statistically distributed
- **p** and E of the mother particle are statistically distributed, are not 100% correlated - could be, that

$M^2 = E^2 - \mathbf{p}^2 < (M_1 + M_2)^2$

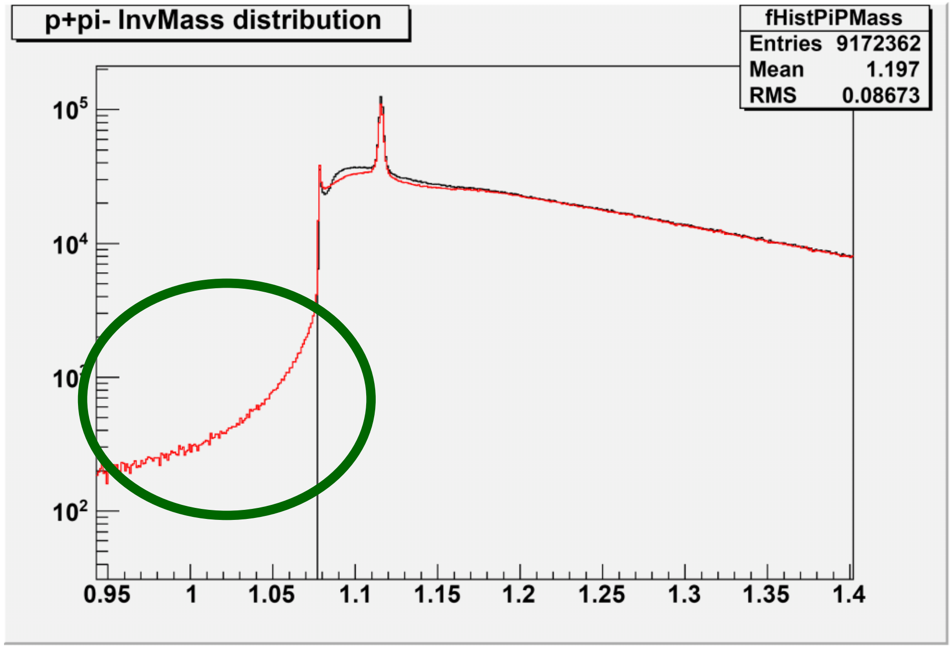


Mass Problem of KFParticle

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

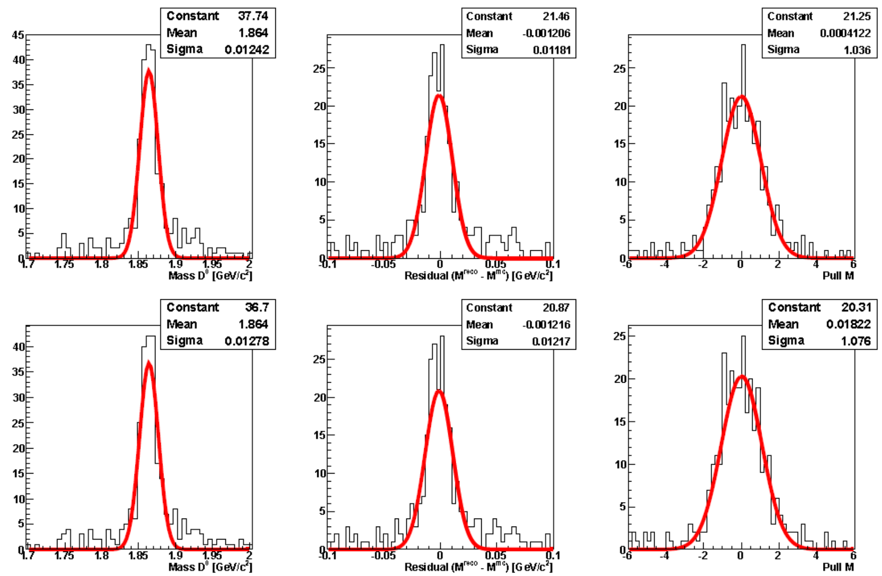
Reconstruction with Mass Calculation

Solution:

- Do not fit an energy as an independent variable
- Recalculate the energy and corresponding elements of the covariance matrix using optimum measurements of the daughter particles
- With this approach only results of E calculation (and, as a result, - M) changes

(Similar approach was implemented in ATLAS, CMS, RAVE package[1], another solution – use constraints)

- The algorithm has been tested on D^0 reconstruction in the ALICE experiment



With mass fit

With mass calculation

- Fit results are similar, tail below sum of masses of the daughter particles disappeared

---

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.

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

Comparison of the 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, +=, -=	+	-

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

  - Under investigation in ALICE now
- Speedup factor of 5 for CBM has been achieved

Examples

Scalar version

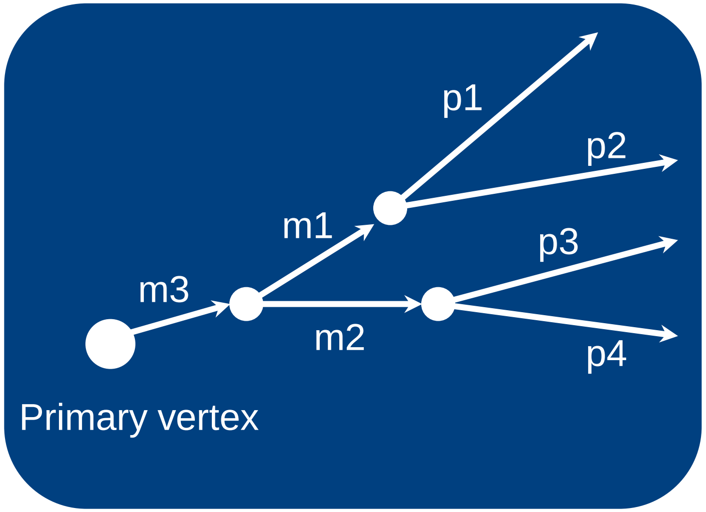
```
AliKFParticle P1, P2;
P1 = AliKFParticle( *pTrack, PDG );
AliKFParticle V0( P1,  P2);

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
```

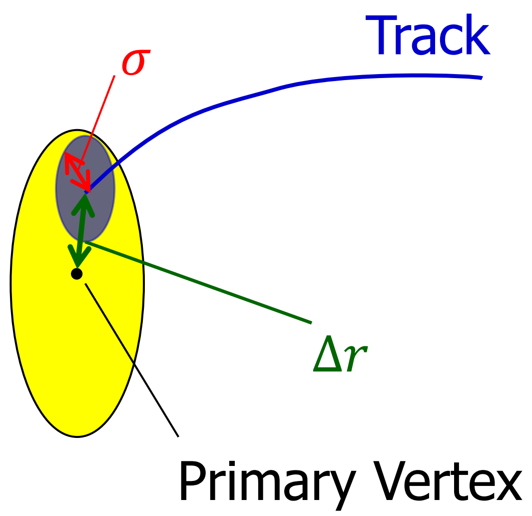


Online Approach for Particles Finding in CBM

- Based on SIMD KFParticle, SIMD KF track fitter
- Online approach for particles finding is implemented within L1 trigger of the CBM experiment
- Algorithm is developed and tested on K_s^0 and Λ^0 ($K_s^0 \rightarrow \pi^+ + \pi^-$, $\Lambda^0 \rightarrow p + \pi^-$)

Tracks and particles selection strategy:

- Track is correctly fitted (parameters and covariance matrix are well defined)
- Track is secondary ($\chi_v < 3$, $\chi_v = \sqrt{\frac{\Delta r^2}{\sigma^2}}$)
- The particle candidate is reconstructed correctly (parameters and covariance matrix are well defined)
- χ^2/NDF < 3 for the reconstructed particle-candidate



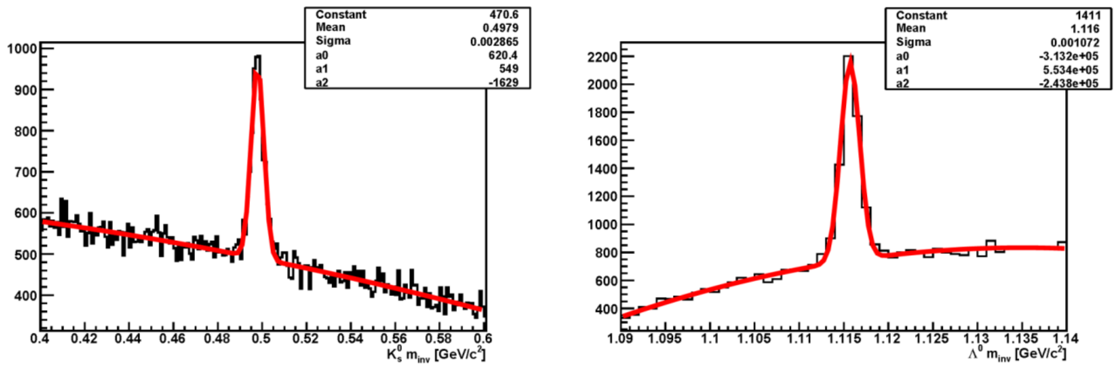
Characteristics of the Online Particles Finder

- 1000 Au+Au central events has been used
- 8 STS stations, digitization and clasterization
- Tested on lxir039 (two Xeon X5550 processors at 2.7 GHz and 8 MB L3), 1 core is used

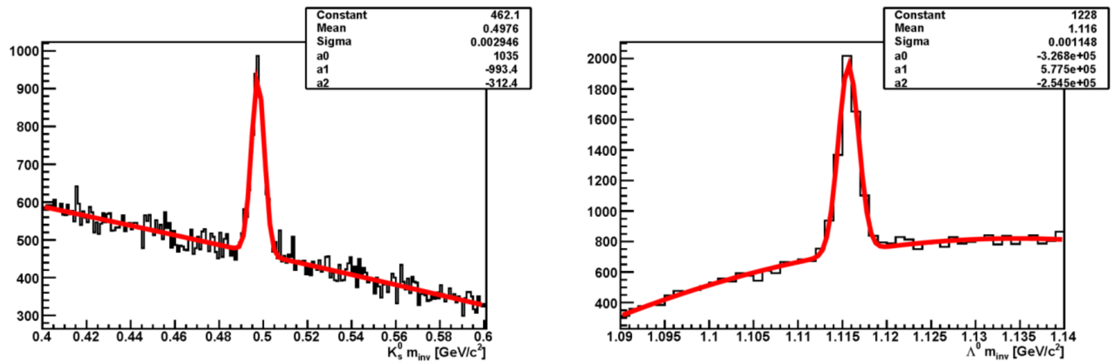
Efficiencies of the algorithm		
Particle	Efficiency, % (normalized on all particles, produced in the event)	Efficiency, % (normalized on particles, which has both daughters reconstructed in STS)
K_s^0	13.9	68.0
Λ^0	12.1	57.8

Efficiency is not 100%, because part of the signal we could not separate from background.

Comparison of the Online and Offline Approaches for Particles Finding



Offline approach
(by I. Vassiliev)



Online approach

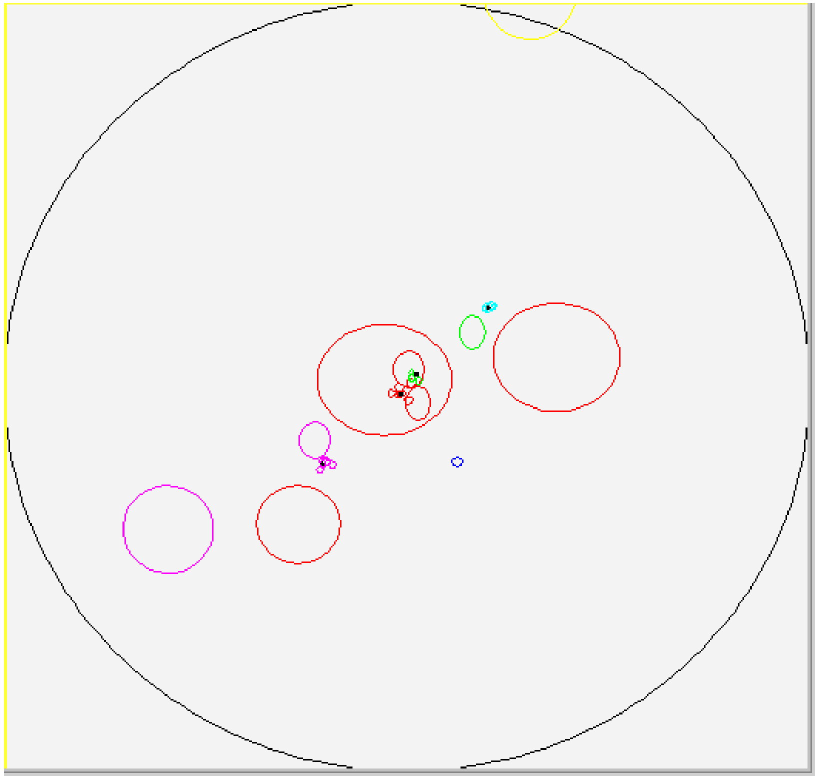
Comparison with offline approach					
		Signal	S/B	Mass, MeV/c^2	Sigma, MeV/c^2
K_s^0	Offline	3225	0.57	497.9	2.9
	Online	3258	0.60	497.6	2.9
Λ^0	Offline	3619	1.14	1115.7	1.1
	Online	3373	1.01	1115.7	1.1

Results are practically the same

An Approach for Multi-Vertex Events Reconstruction

An algorithm for the multi-vertex events reconstruction based on the CBM experiment has been developed:

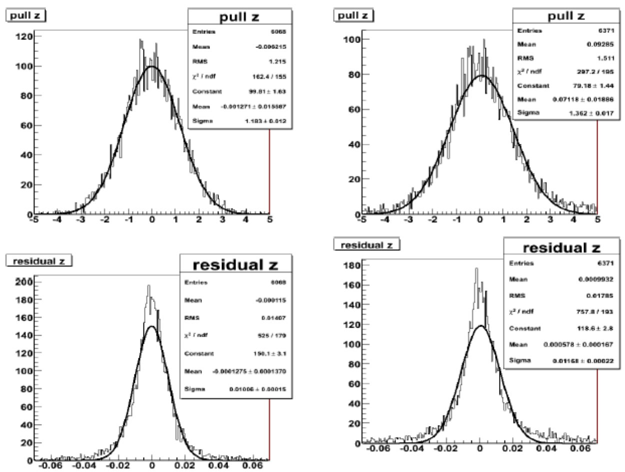
- The weights are calculated for each track according to the number of the number of the closest tracks
- The cluster is formed around the track with the maximum weight
- The procedure is continued with remaining tracks



The algorithm has been tested with p+C events at 30 GeV.

Quality of Multi-Vertex Events Reconstruction

Event with 1 vertex

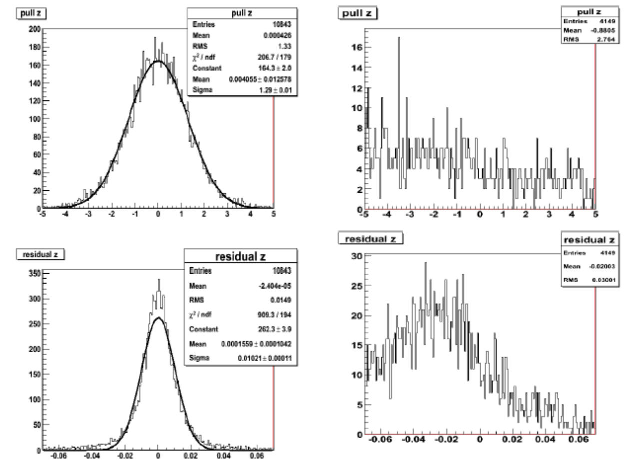


With cluster finding — Without cluster finding

		X	Y	Z
With cluster finder	Residual, µm	19	19	101
	Pull	1.1	1.1	1.2
Without cluster finder	Residual, µm	21	21	117
	Pull	1.3	1.3	1.4

Improvement about 10-13%
(secondary tracks rejection)

Event with 4 vertices



With cluster finding — Without cluster finding

	X	Y	Z
Residual, µm	20	20	102
Pull	1.27	1.26	1.29

Summary

- The KFParticle package is a particle reconstruction package with a rich functionality
- KFParticle has been SIMDized. Simdized version shows the same results
- The online particles finder has been developed based on the SIMDized KFParticle package

Future Plans

- 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
- Multi vertex events reconstruction with KFParticle
- Add adaptive methods (DAF, PDAF, etc.) to the KFParticle package
- Use KFParticle for the 4D tracking developing