

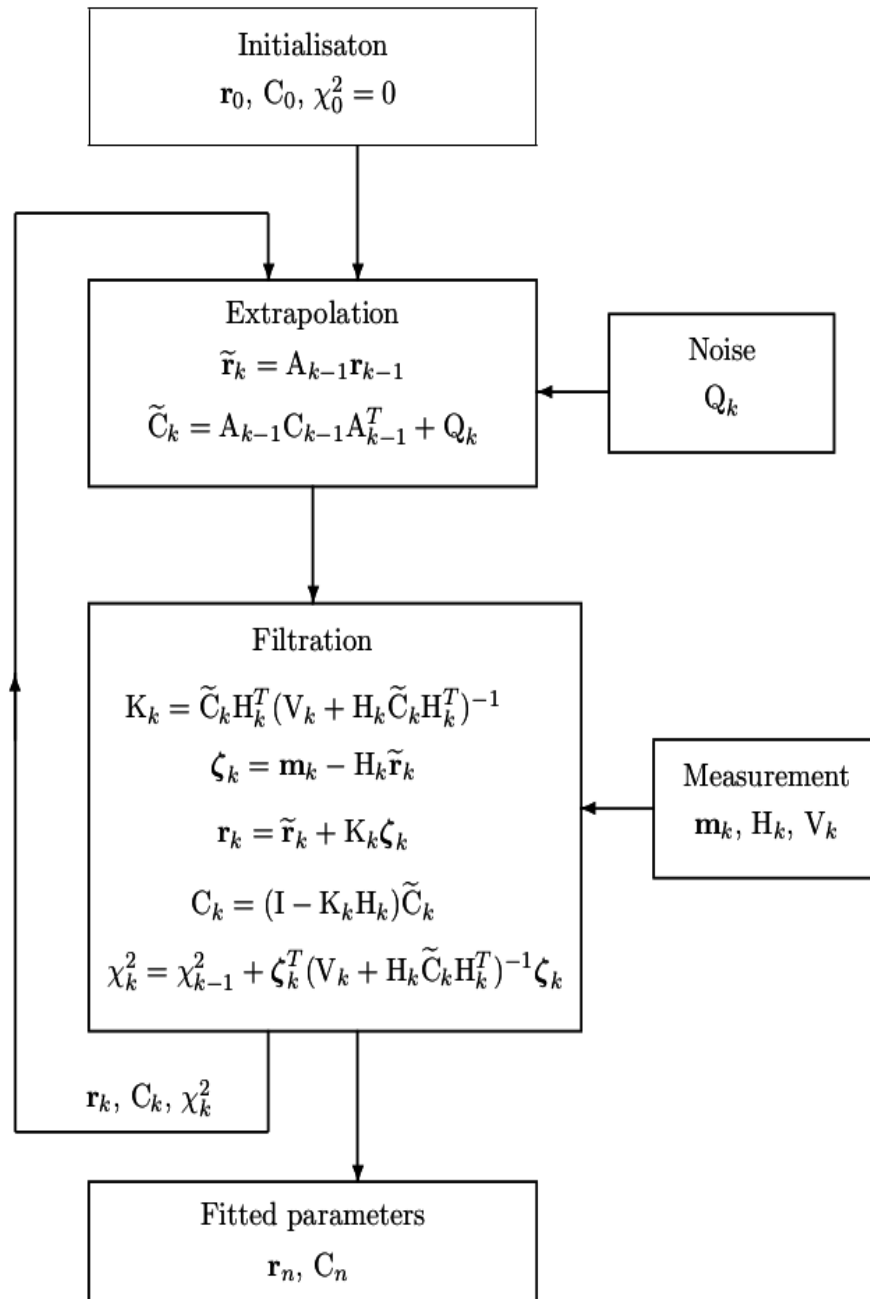


SPD Physics & SW meeting
25 January 2023

Algorithms of vertex reconstruction in SpdRoot

V. Andreev (LPI, Moscow)

Kalman filter algorithm



state vector r^t - vector real numbers that represents the unknown quantities to be estimated

Extrapolation - changes current estimation of vector r_k upon transfer from (k-1)-th measurement to the k-th measurement

$r_k^t = A_k r_{k-1}^t + v_k$, A_k - is a known linear operator, called **extrapolator**; v_k - process noise between (k-1) and k - measurement

Filtration - the measurement information is incorporated into the estimator and its covariance matrix

measurement m_k - a known (measured) quantity with linearly depends on state vector $m = H \cdot r^t + \eta$
 H - is a (known) linearly operator represented as a matrix, called **model of measurement**; η - measurement error

Steps extrapolation and filtration sequentially repeat n times, for each measurement m_k , $k = 1, \dots, n$. After the filtration of the last measurement m_n , the obtained estimator r_n is the desired best estimator with the covariance matrix C_n .

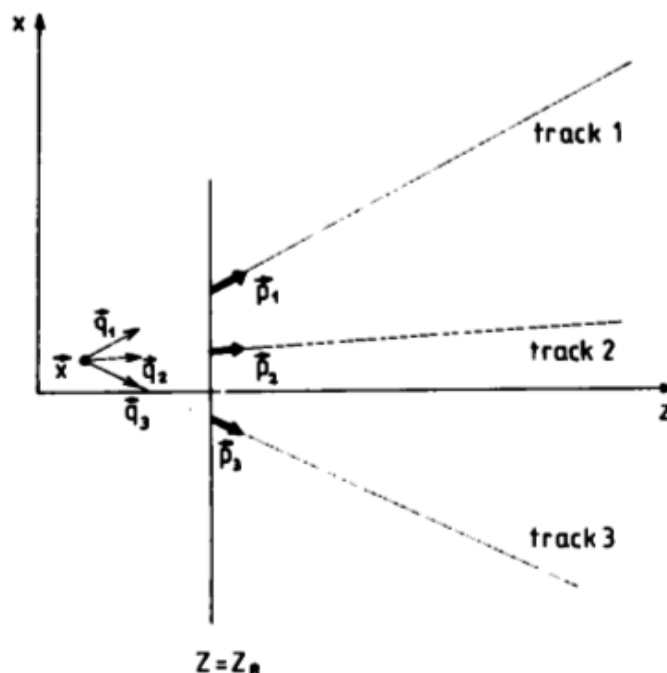
In practice, the transport equation and the measurement model are often nonlinear. To solve the nonlinear fit problem, one should linearise all the equations before applying the fitting algorithm, but the algorithm itself does not change.

General approach to vertex reconstruction

Using the Kalman filter algorithm for vertex reconstruction (fitting) was originally proposed by P.Billoir, R.Fruhwith and M.Regler in “**Track element merging strategy and vertex fitting in complex modular detectors**”, NIM A241 (1985) 115-131 and by R.Fruhwith in “**Application of Kalman filtering to track and vertex fitting**”, NIM A262 (1987) 444-450.

Suppose that there are n -tracks of particles (\mathbf{q}_k) in a magnetic field which are to be fitted to a common vertex (\mathbf{x}). For each reconstructed track, we have **5** fitted track parameters \mathbf{p}_k and the corresponding covariance matrix $\mathbf{V}_k = \mathbf{G}_k^{-1}$, defined at some **reference surface** (Z_0). The fitted track parameters contain the full information from measurements by the tracking detectors, and can be considered as (virtual) measurements for the purpose of the vertex fit.

The aim of the vertex fit is the estimation of the vertex position \mathbf{x} and of the momentum vectors \mathbf{q}_k of all tracks at the common vertex.



Kalman filter for vertex reconstruction

The functional dependence of the track parameters \mathbf{p}_k on the state vector $(\mathbf{x}, \mathbf{q}_k)$ is determined by the equation of motion and is described by the “measurement equation”, which in general is nonlinear

$$\mathbf{p}_k = \mathbf{h}_k(\mathbf{x}, \mathbf{q}_k) + \boldsymbol{\epsilon}_k, \quad \text{cov}(\boldsymbol{\epsilon}_k) = \mathbf{V}_k.$$

If there is multiple scattering between the vertex and the reference surface, it has to be included as additional noise in \mathbf{V}_k . A linear regression model is obtained by approximating \mathbf{h}_k by a first order Taylor expansion at

some point $(\mathbf{x}_e, \mathbf{q}_{k,e})$

$$\mathbf{h}_k(\mathbf{x}, \mathbf{q}_k) \approx \mathbf{h}_k(\mathbf{x}_e, \mathbf{q}_{k,e}) + \mathbf{A}_k(\mathbf{x} - \mathbf{x}_e) + \mathbf{B}_k(\mathbf{q}_k - \mathbf{q}_{k,e}) = \mathbf{c}_{k,e} + \mathbf{A}_k\mathbf{x} + \mathbf{B}_k\mathbf{q}_k,$$

$$\text{with } \mathbf{A}_k = [\partial \mathbf{h}_k / \partial \mathbf{x}]_e \text{ and } \mathbf{B}_k = [\partial \mathbf{h}_k / \partial \mathbf{q}_k]_e$$

And χ^2 can be written as

$$\chi_{k,F}^2 = (\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k-1})^T \mathbf{C}_{k-1}^{-1} (\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_{k-1}) + \mathbf{r}_k^T \mathbf{G}_k \mathbf{r}_k,$$

$$\mathbf{r}_k = \mathbf{p}_k - \bar{\mathbf{p}}_k, \quad \bar{\mathbf{p}}_k = \mathbf{c}_{k,e} + \mathbf{A}_k \bar{\mathbf{x}}_k + \mathbf{B}_k \bar{\mathbf{q}}_k.$$

Minimizing the χ^2 with respect to \mathbf{x}_k and \mathbf{q}_k yields two vector equations which can be solved for the parameters $\mathbf{x}_k, \mathbf{q}_k$ and their covariance matrices. The results are as follows:

$$\bar{\mathbf{x}}_k = \mathbf{C}_k [\mathbf{C}_{k-1}^{-1} \bar{\mathbf{x}}_{k-1} + \mathbf{A}_k^T \mathbf{G}_k^B (\mathbf{p}_k - \mathbf{c}_{k,e})],$$

$$\bar{\mathbf{q}}_k = \mathbf{W}_k \mathbf{B}_k^T \mathbf{G}_k (\mathbf{p}_k - \mathbf{c}_{k,e} - \mathbf{A}_k \bar{\mathbf{x}}_k),$$

$$\text{cov}(\bar{\mathbf{x}}_k) = \mathbf{C}_k = (\mathbf{C}_{k-1}^{-1} + \mathbf{A}_k^T \mathbf{G}_k^B \mathbf{A}_k)^{-1},$$

$$\text{cov}(\bar{\mathbf{q}}_k) = \mathbf{D}_k = \mathbf{W}_k + \mathbf{W}_k \mathbf{B}_k^T \mathbf{G}_k \mathbf{A}_k \mathbf{C}_k \mathbf{A}_k^T \mathbf{G}_k \mathbf{B}_k \mathbf{W}_k = \mathbf{W}_k + \mathbf{E}_k^T \mathbf{C}_k^{-1} \mathbf{E}_k,$$

$$\text{cov}(\bar{\mathbf{x}}_k, \bar{\mathbf{q}}_k) = \mathbf{E}_k = -\mathbf{C}_k \mathbf{A}_k^T \mathbf{G}_k \mathbf{B}_k \mathbf{W}_k,$$

with

$$\mathbf{W}_k = (\mathbf{B}_k^T \mathbf{G}_k \mathbf{B}_k)^{-1},$$

$$\mathbf{G}_k^B = \mathbf{G}_k - \mathbf{G}_k \mathbf{B}_k \mathbf{W}_k \mathbf{B}_k^T \mathbf{G}_k.$$

Possible simplification of general algorithm

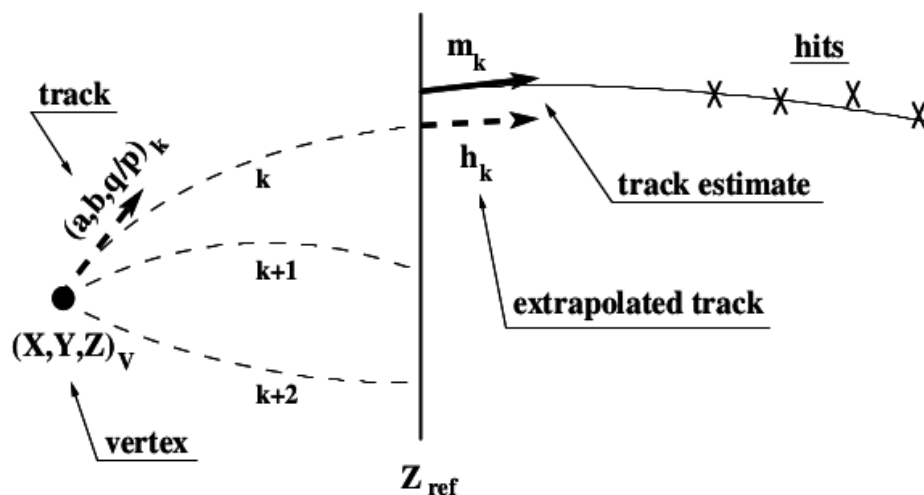
The serious disadvantage of the basic method is that it requires too many calculations. Since the state vector \mathbf{x} has a dimension of 3×3 and each measurement has a dimension of 5×5 , complicated matrix operations must be performed at each step. In particular, at each filtration step it is necessary to invert $(5n+3) \times (5n+3)$ matrices. This is very important when the high track multiplicity are observed.

Various fast algorithms for the vertex fit were considered to simplify the basic method. For speeding up the calculations the following simplifications of the problem are usually applied:

- neglect of the magnetic field in the vertex region when tracks are considered to be straight lines;
- fixation of track directions and momenta neglecting uncertainties of these parameters;
- use the helix parametrization for track with assumption of the constant magnetic field in the vertex region => it is possible to do calculation of matrix members by analytical

CBM vertex reconstruction algorithm (1)

A special feature of the presented algorithm is the extrapolation of the track parameters to the vertex linearisation point \mathbf{r}_0 (S.Gorbunov and I.Kisel, “**Primary vertex fit based on the Kalman filter**”, CMB-SOFT-note-2006-01, 9 January 2006). This approach gives possibility to fit the vertex without including the track parameters into the vertex state vector and so to simplify the calculations. In particular, only two divisions are performed for each track, while in the standard approach inversion of a 5×5 matrix is required.



$\mathbf{r} = (x_v, y_v, z_v)^T$, \mathbf{C}_v — the vertex position and its covariance matrix;

$\mathbf{t}_k = (a_k, b_k, (q/p)_k)^T$, \mathbf{C}^{tk} — the directions and the inverse momentum of the k -th track, originating from the vertex \mathbf{r} , and covariance matrix for these parameters; measurement;

$\mathbf{h}_k(r, a_k, b_k, (q/p)_k)$ — parameters of the k -th track, extrapolated from \mathbf{z}_v to \mathbf{z}_{ref} ;

$\mathbf{m}_k = (x_k, y_k, t_{xk}, t_{yk}, (q/p)_k)^T$ — the k -th track estimation, parametrized at a certain \mathbf{z}_{ref} ;

\mathbf{V}_k — the covariance matrix of the k -th track estimate;

Each track estimation \mathbf{m}_k is considered as measurement of the corresponding track \mathbf{t}_k .

CBM vertex reconstruction algorithm (2)

$\mathbf{h}_k(r, a_k, b_k, (q/p)_k)$ — parameters of the k-th track, extrapolated from \mathbf{z}_v to \mathbf{z}_{ref} .

$$\mathbf{h}_k(\mathbf{r}, a_k, b_k, (q/p)_k) = \begin{pmatrix} x_v + a_k \cdot (z_{ref} - z_v) + O((z_{ref} - z_v)^2) \\ y_v + b_k \cdot (z_{ref} - z_v) + O((z_{ref} - z_v)^2) \\ a_k \\ b_k \\ (q/p)_k \end{pmatrix}.$$

Here the term $O((\mathbf{z}_{ref} - \mathbf{z}_v)^2)$ describes the deviation of the track from a straight line in a magnetic field (see details S. Gorbunov and I. Kisel, “**Analytic formula for track extrapolation in non-homogeneous magnetic field**”. Nucl. Instr. and Meth. A559 (2006)).

The measurement model is:

$$\begin{aligned} \mathbf{m}_k &= \mathbf{h}_k(\mathbf{r}^t, a_k^t, b_k^t, (q/p)_k^t) + \boldsymbol{\eta}_k, \\ \langle \boldsymbol{\eta}_k \cdot \boldsymbol{\eta}_k^T \rangle &= V_k. \end{aligned}$$

The measurement model is nonlinear in z_v, a_k, b_k and, therefore, should be linearized at a certain values (z_v^0, a_k^0, b_k^0)

$$\mathbf{h}_k(\mathbf{v}, \mathbf{t}_k) \approx \mathbf{h}_k(\mathbf{v}^0, \mathbf{t}_k^0) + \left. \frac{\partial \mathbf{h}_k(\mathbf{v}, \mathbf{t}_k)}{\partial (\mathbf{v}, \mathbf{t}_k)} \right|_{\mathbf{v}^0, \mathbf{t}_k^0} (\mathbf{v} - \mathbf{v}^0, \mathbf{t}_k - \mathbf{t}_k^0)$$

CBM vertex reconstruction algorithm (3)

Let us extrapolate all the track estimations to z_v^0 (setting $z_{ref} = z_v^0$). The Jacobian H_k of the vector \mathbf{h}_k at z_v^0 is:

$$H_k = \begin{pmatrix} 1 & 0 & -a_k^0 & 0 & 0 & 0 \\ 0 & 1 & -b_k^0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad \text{where } H_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{v}, \mathbf{t}_k)}{\partial (\mathbf{v}, \mathbf{t}_k)} \right|_{\mathbf{v}^0, \mathbf{t}_k^0} (\mathbf{v} - \mathbf{v}^0, \mathbf{t}_k - \mathbf{t}_k^0)$$

The linearized measurement model is:

$$\mathbf{m}_k(\mathbf{r}) \approx \begin{pmatrix} a_k^0 \cdot z_v^0 \\ b_k^0 \cdot z_v^0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & -a_k^0 & 0 & 0 & 0 \\ 0 & 1 & -b_k^0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_v \\ y_v \\ z_v \\ a_k \\ b_k \\ (q/p)_k \end{pmatrix} + \boldsymbol{\eta}_k.$$

In the linearized model of measurement:

- only the \mathbf{x} and \mathbf{y} components of \mathbf{m}_k depend on the vertex parameters \mathbf{r}_k ;
- these components do not depend on the parameters $(a_k, b_k, (q/p)_k)$ of the vertex track;
- the parameters of the k-th vertex track are measured only by the k-th track estimate \mathbf{m}_k .

Thus, the values $a_k, b_k, (q/p)_k$ do not influence the measurement of the vertex position r_k with the track estimate \mathbf{m}_k and, therefore, there is no need to fit these values at the k-th step of the Kalman filter.

CBM vertex reconstruction algorithm (4)

The vertex fit is performed according to of the Kalman filter algorithm with the state vector $r=(x_v, y_v, z_v)^T$ and the measurement model

$$\begin{aligned} \mathbf{H}_k^{xy} &= \begin{pmatrix} 1 & 0 & -a_k^0 \\ 0 & 1 & -b_k^0 \end{pmatrix} \\ \mathbf{S}_k &= (\mathbf{V}_k^{xy} + \mathbf{H}_k^{xy} \mathbf{C}_k \mathbf{H}_k^{xyT})^{-1} \\ \mathbf{K}_k &= \mathbf{C}_k \mathbf{H}_k^{xyT} \mathbf{S}_k \\ \zeta_k &= \mathbf{m}_k^{xy} - \mathbf{H}_k^{xy} (\mathbf{r}_{k-1} - \begin{pmatrix} 0 \\ 0 \\ z_v^0 \end{pmatrix}) \\ \mathbf{r}_k &= \mathbf{r}_{k-1} + \mathbf{K}_k \zeta_k \\ \mathbf{C}_k &= \mathbf{C}_{k-1} - \mathbf{K}_k \mathbf{H}_k^{xy} \mathbf{C}_k \\ \chi_k^2 &= \chi_{k-1}^2 + \zeta_k^T \mathbf{S}_k \zeta_k \end{aligned}$$

The complete algorithm is following:

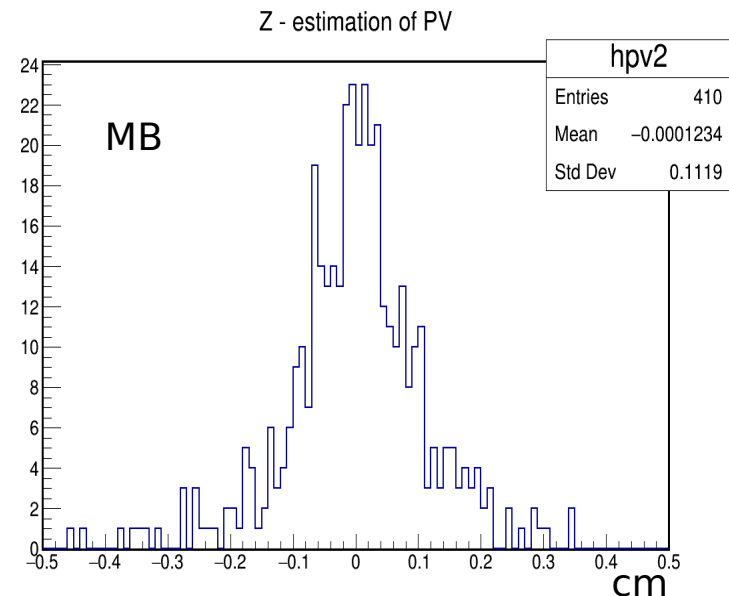
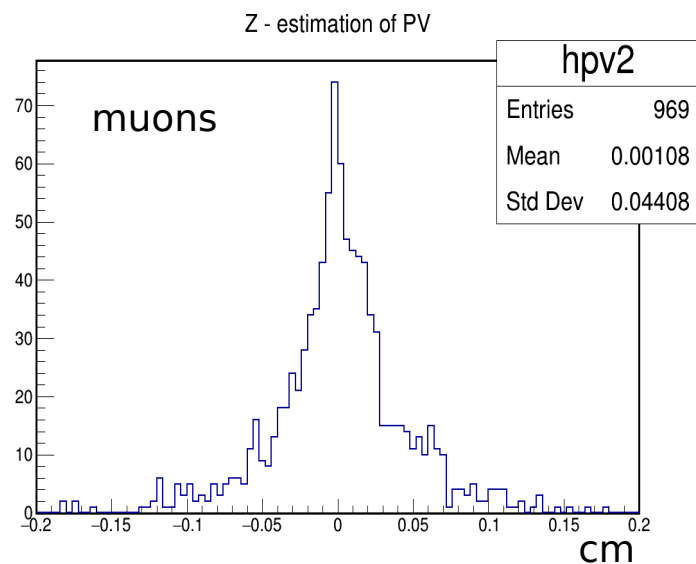
- 1) First approximation of the vertex $\mathbf{r} = (x_v^0, y_v^0, z_v^0)$
- 2) A few iterations of the vertex fit:
 - a) initialization:
 - store vertex from previous iteration: $\mathbf{r}^0 = \mathbf{r}, C^0 = C$;
 - the initial vertex state vector is taken from the previous iteration $\mathbf{r} = \mathbf{r}^0$;
 - the initial covariance matrix is set to $C = I \cdot \text{inf.}$;
 - initialize the vertex χ^2 and number of degrees of freedom.
 - b) filtering (repeats for every track estimate):
 - extrapolate the track estimate \mathbf{m}_k to z_v^0 ;
 - calculation of χ^2 -distance of the k-th track estimate from the vertex \mathbf{r}^0 ;
 - calculation of parameters of the k-th vertex track a_k^0, b_k^0 ;
 - update the state vector \mathbf{r} with the Kalman filter formalism.
 - c) repeat filtering for the next track.
- 3) Make the next iteration of the fitting routine.
- 4) Store r, C .

Implementation of vertex reconstruction in SPD

This vertex reconstruction algorithm was implemented in SPDroot in 2019 and its performance was checked with the standard procedure - comparison with the MC vertex position. The 2 options of vertex detector were considered: DSSD (5 layers) and MAPS (3 layers) + DSSD (2 layers).

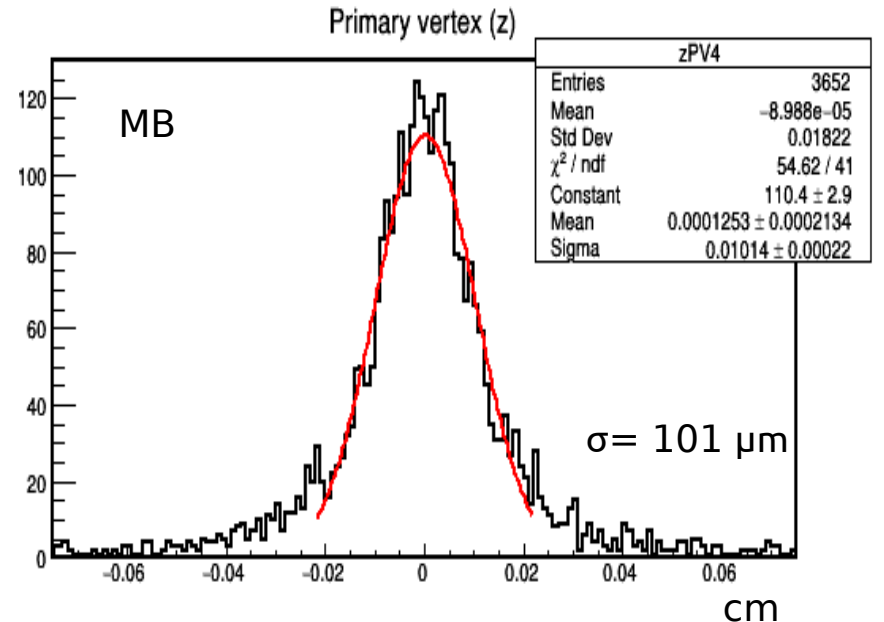
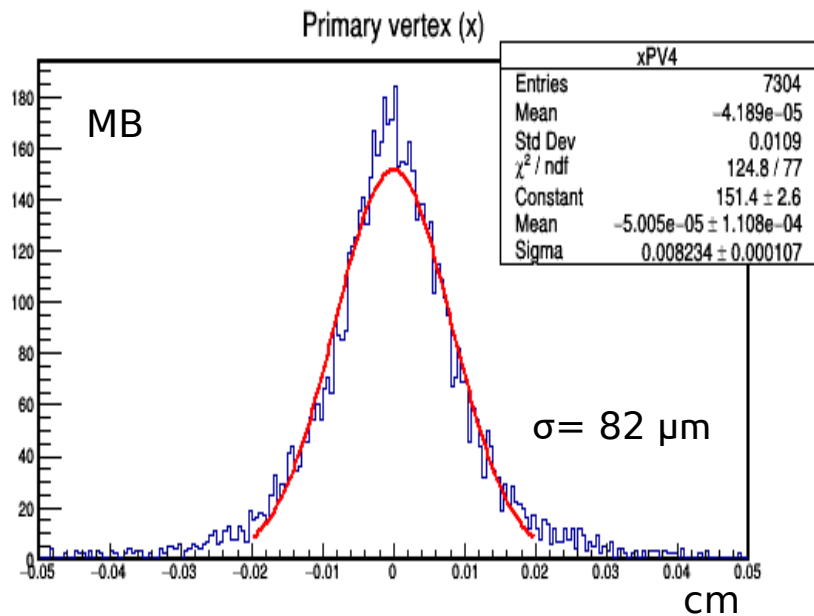
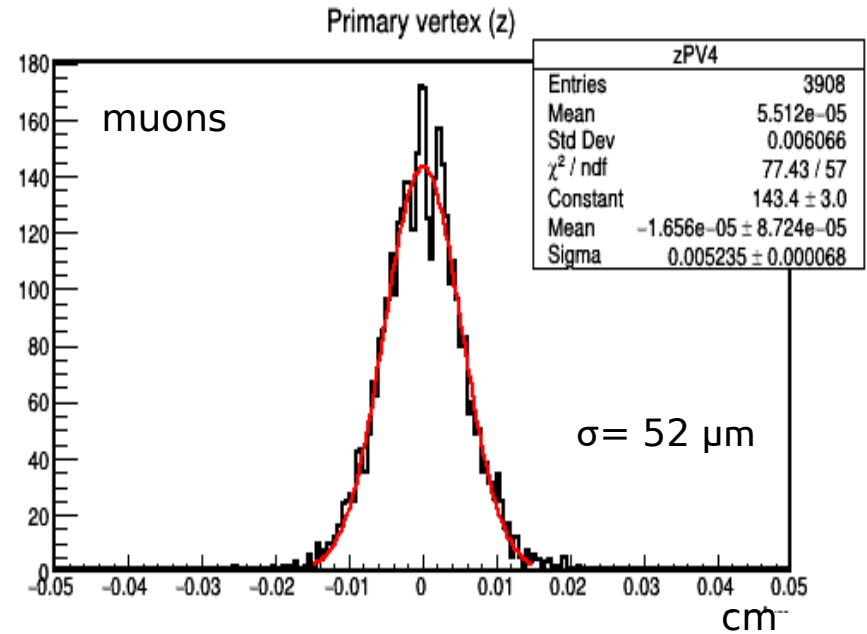
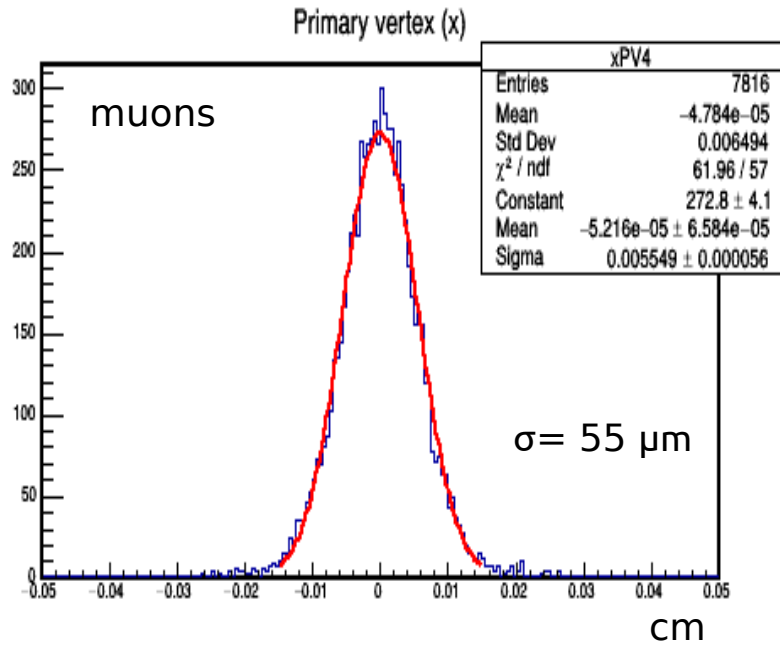
The important point at the vertex reconstruction procedure is the determination of the initial approximation of the primary vertex. In SPD the next procedure is realized for this purpose:

- select tracks;
- extrapolate tracks to beam axis;
- estimate z-coordinate for point of the closest approach to the beam axis;
- apply some clustering algorithm or simplified Kalman filter for estimation of initial vertex position.

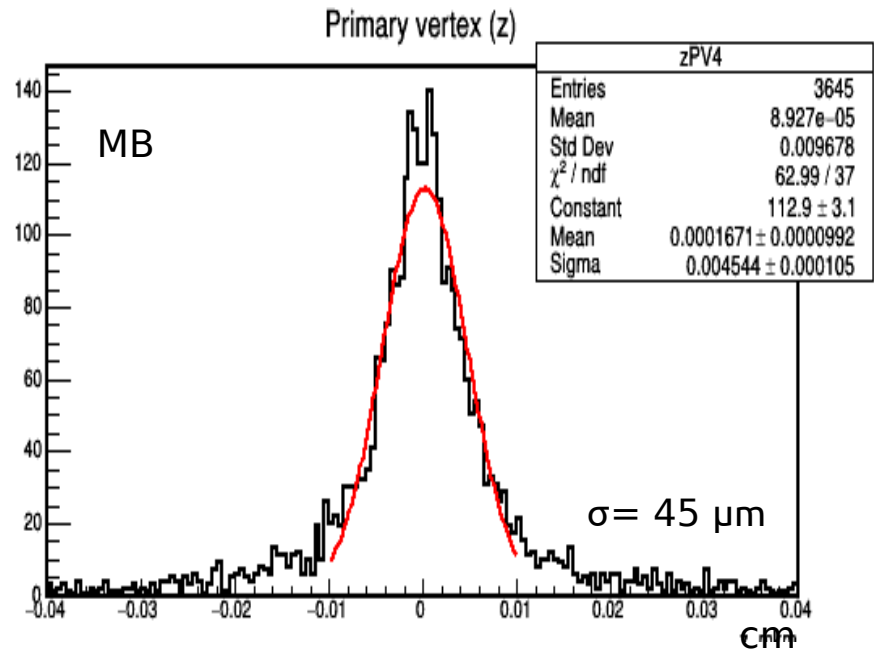
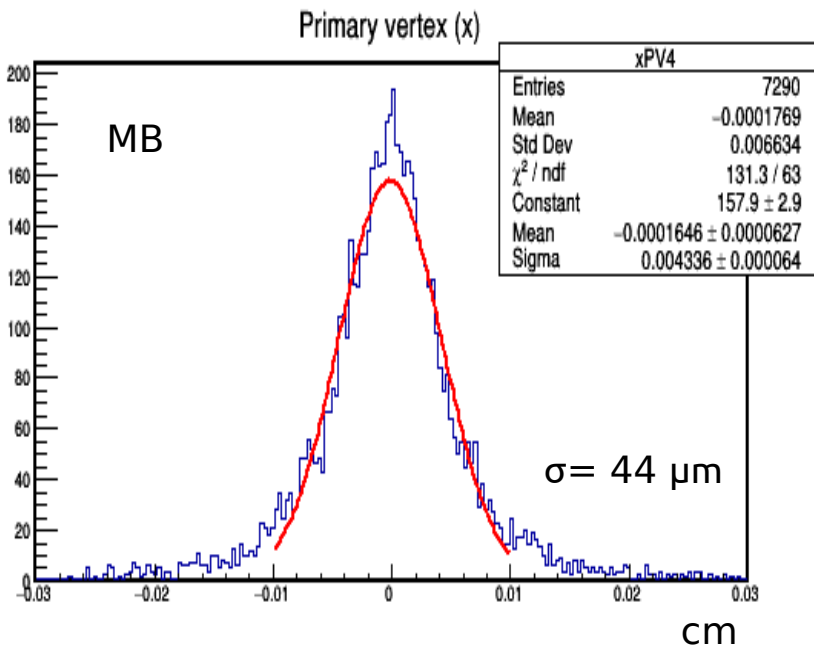
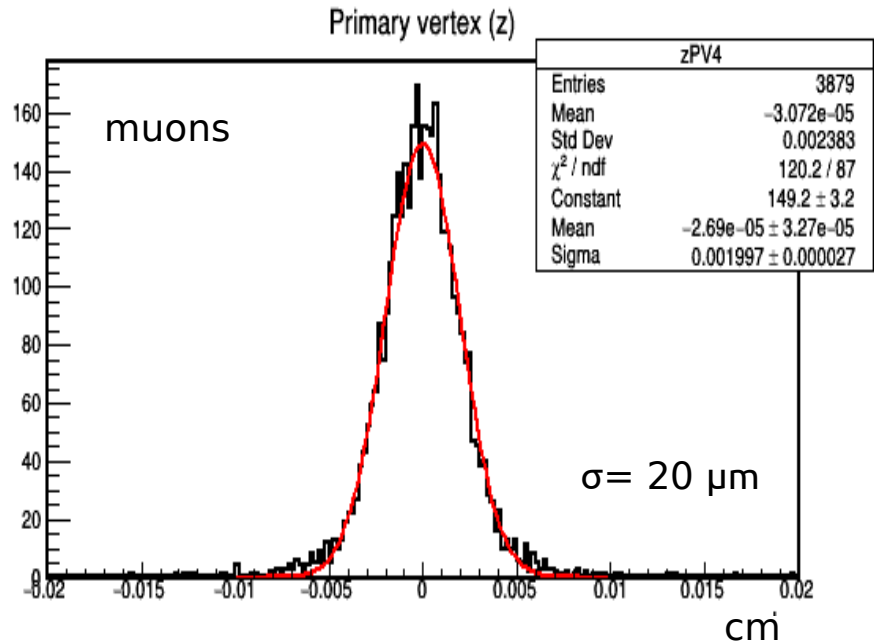
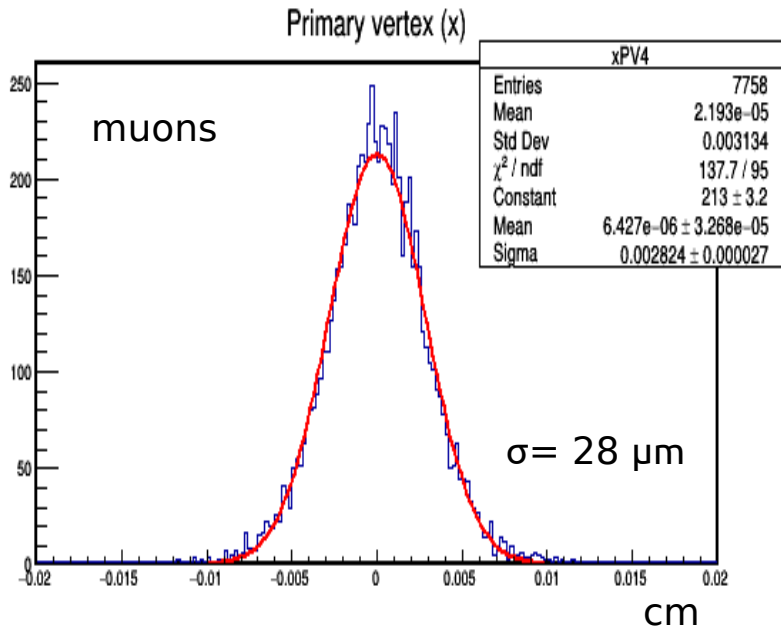


On plots above you see the difference between z-coordinate of the primary vertex position and the initial approximation of vertex for 2 MC samples: 8 muons of 1 GeV/c with uniform distribution of θ and ϕ angles and Minimum Bias events. The primary vertex position was Gaussian distributed with $\sigma_z = 30$ cm and $\sigma_{x,y} = 0.1$ cm. MAPS option of vertex detector is used at this simulation.

Reconstruction precision of primary vertex (DSSD)



Reconstruction precision of primary vertex (MAPS+DSSD)



Cross check for vertex reconstruction algorithm

The additional check for the present SPD primary vertex reconstruction algorithm was done on the base of comparison with the others vertex reconstruction algorithms which used in high energy physics.

The next algorithms were considered:

- classical Kalman vertex filter (KVF);
- trimmed Kalman filter (TVF);
- adaptive vertex filter (AVF);
- primary vertex reconstruction from KFPackage package (KFP);
- SPD vertex reconstruction algorithm.

The KVF, TVF and AVF primary vertex reconstruction algorithms are realized in the special RAVE package (Reconstruction in an Abstract, Versatile Environment) which was developed for vertex reconstruction in CMS experiment. This toolkit uses a set of reconstructed particle tracks as input and produces reconstructed interaction vertices as output. It deals both with finding (pattern recognition) and with fitting of the interaction vertices. This RAVE package is also geometry independent.

It is necessary to emphasize that the considered algorithms in RAVE use the different parameters for track representation and also various algorithms for finding initial approximation of the primary vertex with comparison of SPD reconstruction.

The special interface was written in order to use data from SPD in RAVE package.

Also mentioned algorithms (KVF, TVF, AVF, KFP) are realized in assumption of the constant magnetic field and for this reason the presented comparison was done for the constant magnetic field also for SPD primary vertex reconstruction.

Reconstruction algorithms in HEP (1)

The most commonly used algorithm for reconstruction of vertex is the classical **Kalman vertex filter** (KVF). This algorithm is mathematically equivalent to the least-squares method which is an optimum evaluation in the case where the model is linear and random noise has the Gaussian distribution, as well as there are no outliers. Since this algorithm uses all tracks in the event, it is sensitive enough to tracks from secondary interactions or decays of short-lived particles which can cause biases in estimation of the parameters of the primary vertex.

The **trimmed Kalman filter** (TVF) represents a some extension of the classical Kalman filter in which tracks mismatching with vertex are removed from the vertex reconstruction process. Initially, all tracks are fitted to the vertex, then the track least compatible to this vertex is removed from fit procedure, and vertex fitting is repeated again. The track compatibility with the vertex is defined by the normalized track-vertex distance χ^2 taking into account track and vertex covariance matrices according to the expression

$$\chi^2 = \Delta r^T (C_{\text{track}} + C_{\text{pv}})^{-1} \Delta r$$

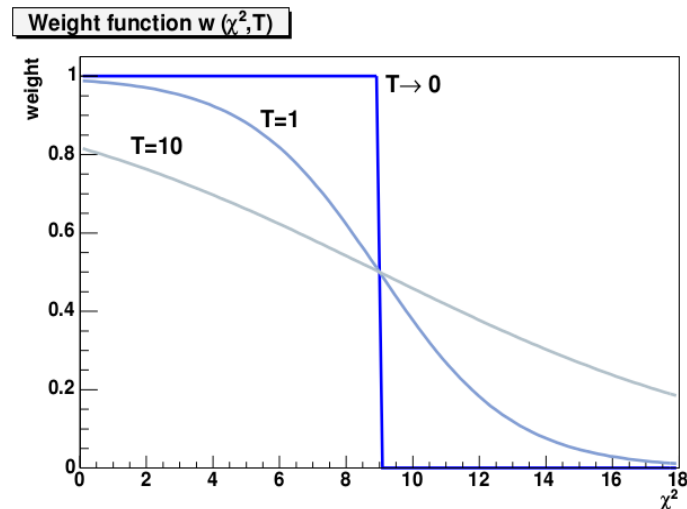
where Δr is the vector of the minimum track-vertex distance, C_{track} and C_{pv} are the covariance matrices of the track and primary interaction vertex. This procedure is repeated until a certain minimum number of tracks related to this vertex will reach or all tracks become correctly related to the vertex.

The **adaptive vertex filter** (AVF) is another algorithm used for fitting the primary interaction vertex in the CMS and ATLAS collaborations at LHC in the CERN. It is an iterative algorithm in which the track weight is a variable and is determined according to its normalized distance χ^2 to the vertex by the formula

$$w(\chi^2) = \frac{1}{\left(1 + \exp \frac{\chi^2 - \chi_{\text{cuto}}^2}{2T}\right)}$$

where χ_{cutoff}^2 is the distance at which the track weight is 0.5. T is the global temperature defining the track weight variation rate as a function of χ^2 (see figure on the next slide).

Reconstruction algorithms in HEP (2)



Also the special procedure (annealing) is applied in this algorithm, according to which, beginning with a certain value, the temperature in the formula for the track weight decreases with each iteration according to a certain scheme until reaching unity.

These KVF, AVF, TVF algorithms for finding the initial approximation of the primary vertex use the special FSMW (Fraction-of Sample Mode with Weights-estimator) algorithm. It forms a set of points defined as a point in space, for which the distance between two tracks is minimum, and a certain weight is attributed to this point. Then, this set of space points is used in a procedure for finding a rough approximation of the vertex.

The next considered algorithm is the algorithm implemented in the **KFParticle package** (KFP). This package was developed for reconstruction of the short-lived particles in the ALICE and CBM experiments and can be also used in the SPD experiment now. The reconstruction procedure is performed according to the standard Kalman filter technique: the estimation of the initial primary vertex approximation and the subsequent iterative fit of the vertex with removal of tracks deflecting from this vertex more than a certain threshold value χ^2_{cut} . In this algorithm, for determination of the initial approximation of the primary vertex, the simplified Kalman filter is used. Then the weight of each track for this initial vertex approximation is determined and then clustering procedure around the track with maximum weight is applied.

Events simulation

To compare the various reconstruction algorithms the next of Monte Carlo events were simulated using the SPDroot simulation and reconstruction package:

- 8 charged muons of 1 GeV/c, uniformly distributed over azimuthal and polar angles;
- Minimum Bias events including diffraction and inelastic interactions;
- deep inelastic events (DIS) with open charm (D^0 -mesons) production.

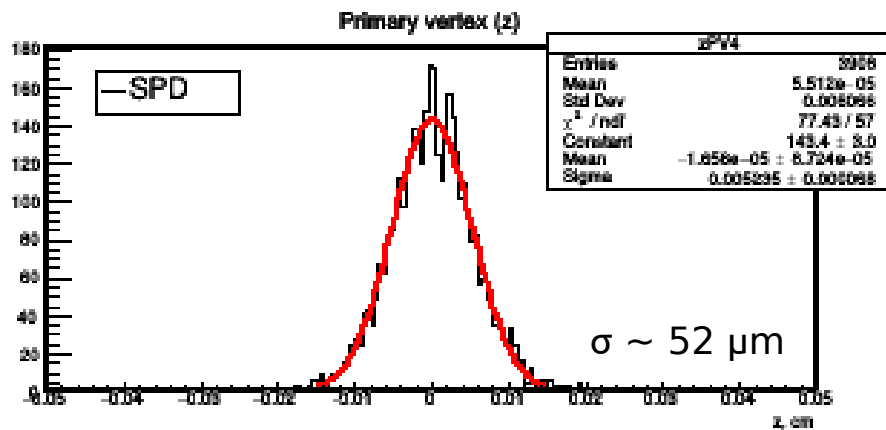
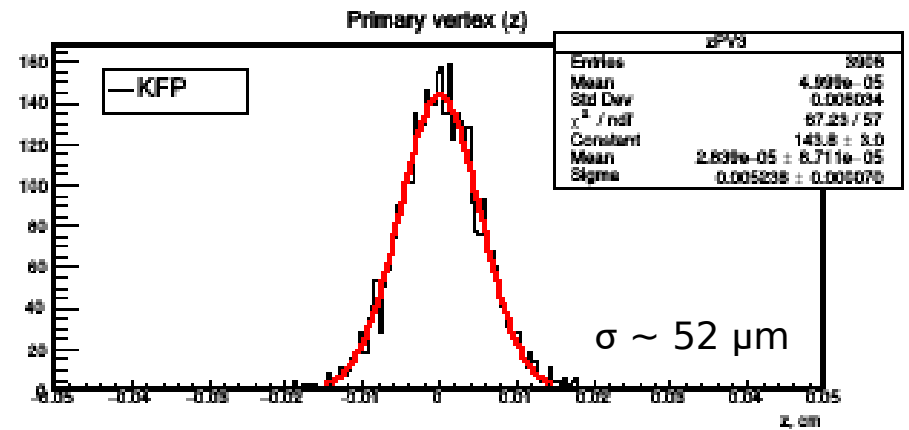
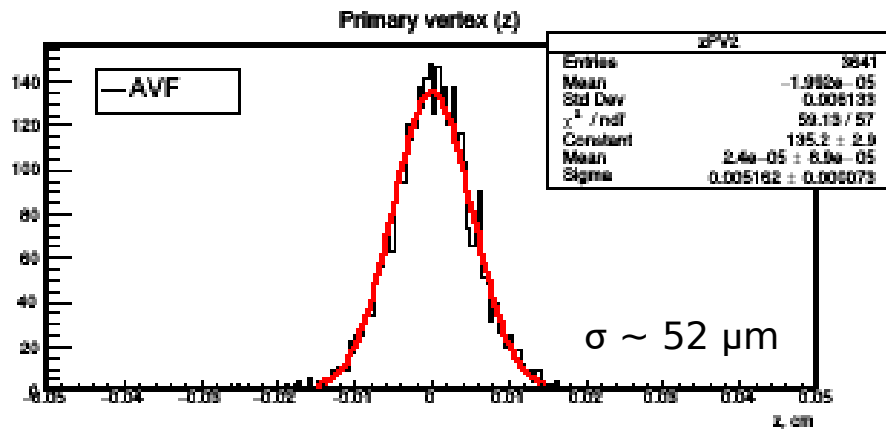
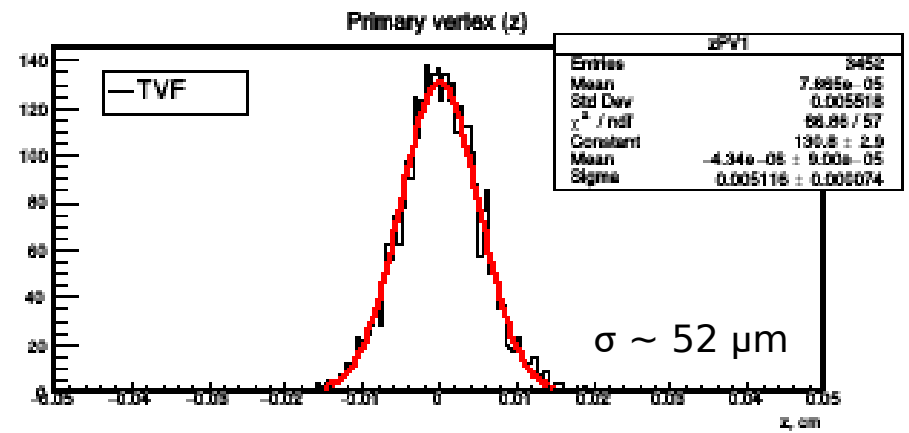
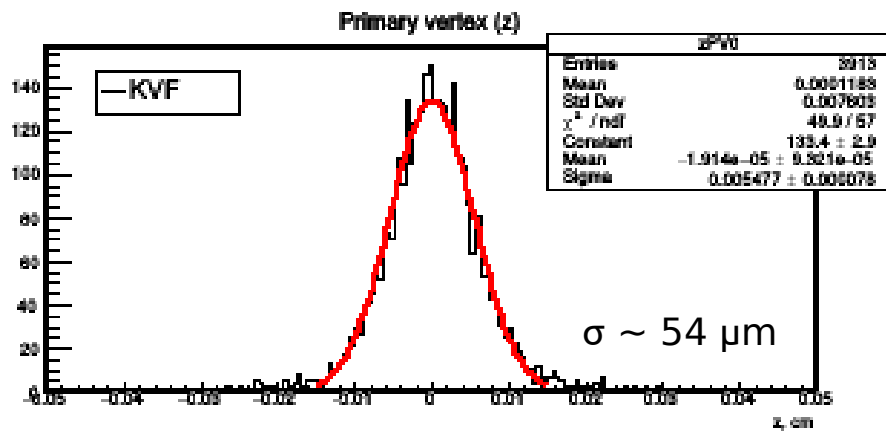
And as was mentioned above some algorithms are used at the constant magnetic field, others can be used for an inhomogeneous magnetic field and consequently for correct comparison this study was performed for a constant magnetic field of 1 T, directed along the z-axis. In the simulation, it was also assumed that the position of primary vertex was $x = 0.0$ cm, $y = 0.0$ cm, $z = 0.0$ cm.

Then, to determine the parameters of the primary vertex, the next procedure was used:

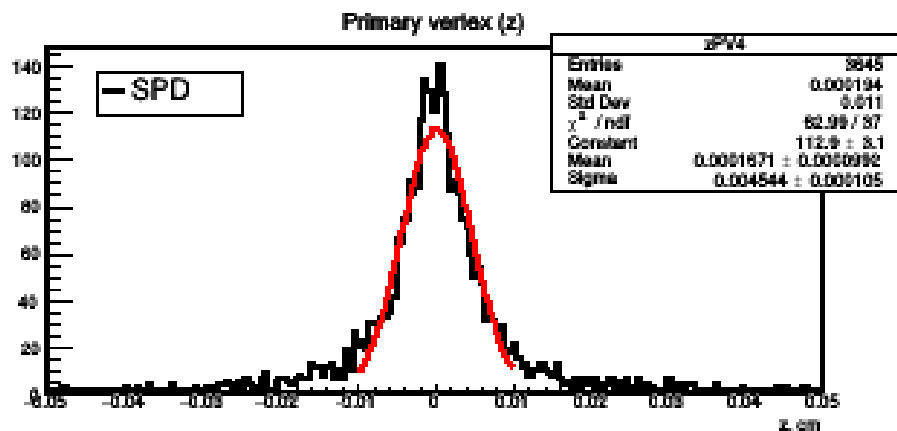
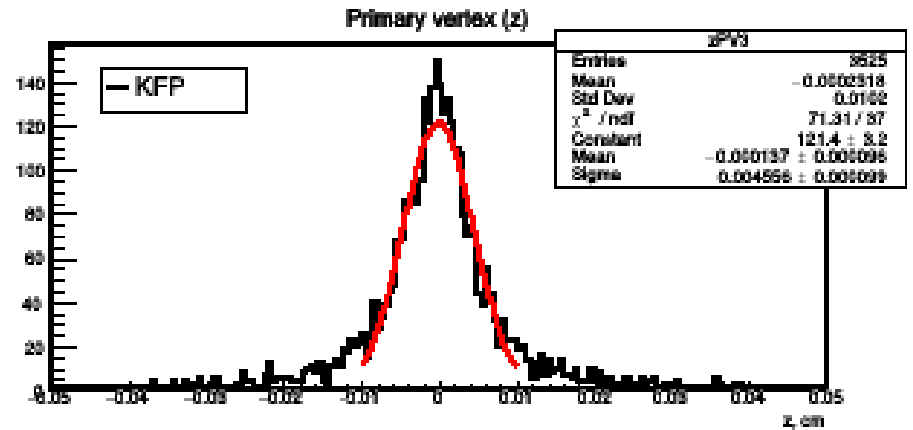
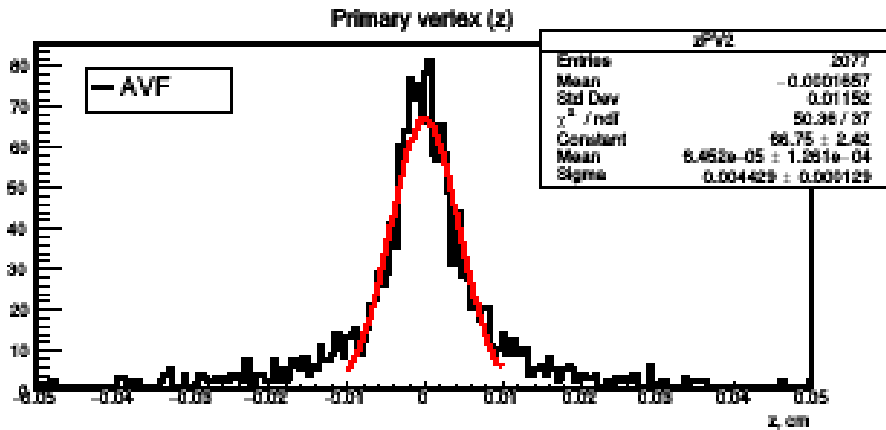
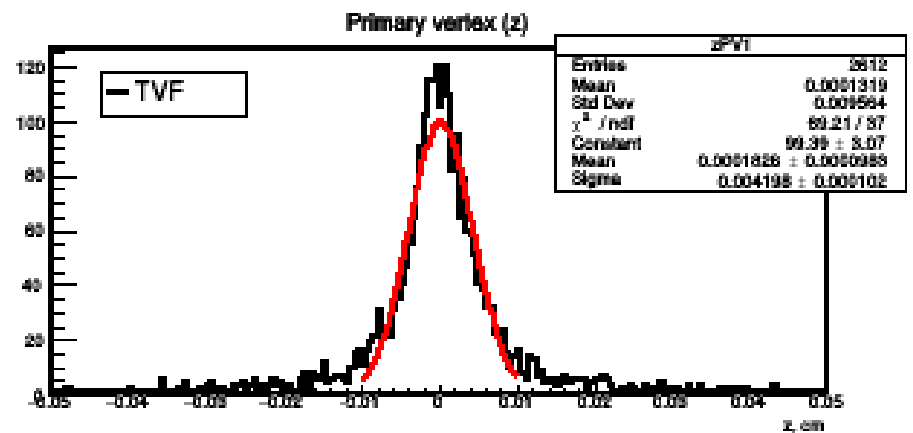
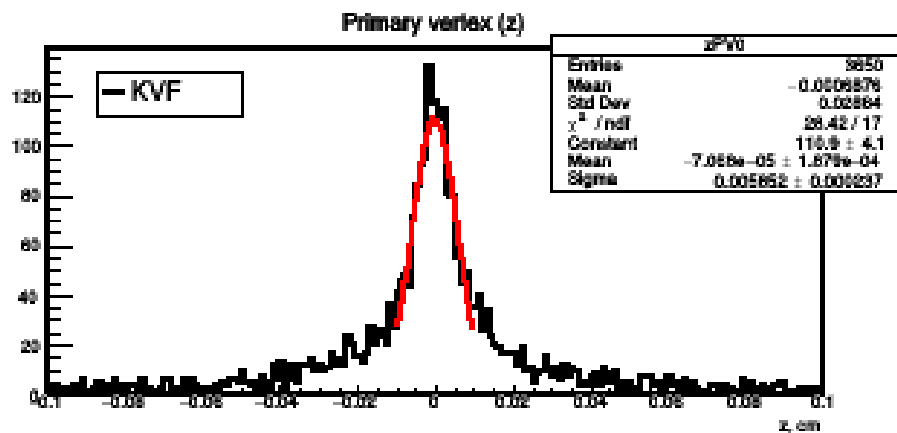
- simulate response in the vertex and tracking detectors with SPDroot;
- determine the charged track parameters on the base of Kalman filter (SPDroot);
- select tracks with 3 or more MC hits in the vertex detector and with 6 or more MC hits in the tracking detector;
- this set of selected tracks was used in the different algorithms for the primary vertex reconstruction.

In this study the following parameters for the adaptive algorithm (AVF) were used: $\chi_{\text{cutoff}} = 9.0$, the initial temperature was set $T = 256$, and the coefficient of temperature decreasing per each subsequent iteration was 0.25. For the trimmed Kalman filter, the threshold of the probability of track compatibility with the interaction vertex was 1% and for the KFP and SPD algorithms, the threshold value of χ_{cut} was 3.5.

Comparison of different algorithms (muons, DSSD)



Comparison of different algorithms (MB, MAPS+DSSD)



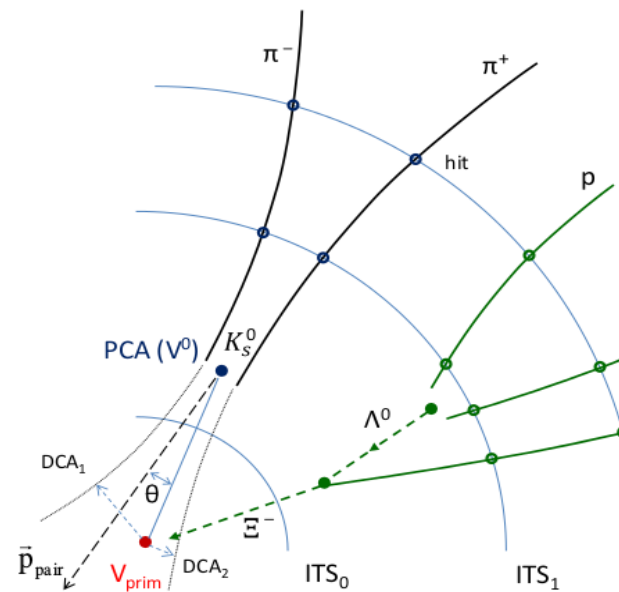
Summary results

		DSSD				MAPS+DSSD			
		σ_x	95%	σ_z	95%	σ_x	95%	σ_z	95%
		[μm]				[μm]			
8 muons	KVF	58.1	198	54.7	144	29.8	81.5	21.6	63.5
	AVF	55.5	169	52.9	119	28.3	71.7	19.7	47.3
	TKF	54.7	127	52.7	113	27.7	60.6	19.1	42.9
	KFP	55.9	122	52.4	111	28.4	61.2	20.1	45.5
	SPD	55.4	121	52.3	110	28.2	59.8	19.9	44.9
MB	KVF	165	638	150	607	88.8	408	73.3	386
	AVF	116	522	122	446	68.2	325	52.0	266
	TVF	106	320	113	384	54.4	176	46.9	213
	KFP	86.6	220	115	411	50.7	140	51.8	220
	SPD	87.3	246	114	410	48.9	161	51.3	250
DIS(D^0)	KVF	116	412	99.4	365	77.0	398	68.3	352
	AVF	88.8	340	85.1	258	58.6	289	51.2	200
	TVF	74.5	197	76.8	206	44.6	117	39.8	129
	KFP	64.1	149	78.0	202	40.7	108	44.8	153
	SPD	64.6	151	79.5	211	39.6	112	45.1	166

Reconstruction of short-lived particles

General scheme of short-lived particles reconstruction:

- fit tracks with Kalman filter or another methods;
- select primary and secondary particles on the base of some criteria;
- reconstruct secondary vertex position using daughter particles;
- finally, reconstruct the mother particle parameters based on parameters of daughter particles and secondary vertex position.



The special **KFParticle package** has been developed for the complete reconstruction of short-lived particles with their momentum, energy, mass, lifetime, decay length, rapidity, etc. KFParticle package is used in ATLAS, STAR, CBM and others experiments.

Now this KFParticle package is also available in SPDroot (thanks to Artur).

KFParticle package

Package has following properties:

- is based on Kalman filter mathematics;
- daughter and mother particles are described with the same set of parameters and are treated in exactly the same way;
- the package is geometry independent;
- daughter particles are added to the mother particle absolutely independently from each other;
- the package allows reconstruction of decay chains;
- the state vector of the particle includes 8 parameters => $\mathbf{r} = (x, y, z, P_x, P_y, P_z, E, s)$, where $s = l/p$ (distance normalized on the momentum, when a production point of particle is known, x,y,z - coordinates of first hit position)

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

State vector

$$\mathbf{C} = \langle \mathbf{r}\mathbf{r}^T \rangle =$$

Covariance matrix

$$\begin{bmatrix} \sigma_x^2 & C_{xy} & C_{xz} & C_{xp_x} & C_{xp_y} & C_{xp_z} & C_{xE} \\ C_{xy} & \sigma_y^2 & C_{yz} & C_{yp_x} & C_{yp_y} & C_{yp_z} & C_{yE} \\ C_{xz} & C_{yz} & \sigma_z^2 & C_{zp_x} & C_{zp_y} & C_{zp_z} & C_{zE} \\ C_{xp_x} & C_{yp_x} & C_{zp_x} & \sigma_{p_x}^2 & C_{p_x p_y} & C_{p_x p_z} & C_{p_x E} \\ C_{xp_y} & C_{yp_y} & C_{zp_y} & C_{p_x p_y} & \sigma_{p_y}^2 & C_{p_y p_z} & C_{p_y E} \\ C_{xp_z} & C_{yp_z} & C_{zp_z} & C_{p_x p_z} & C_{p_y p_z} & \sigma_{p_z}^2 & C_{p_z E} \\ C_{xE} & C_{yE} & C_{zE} & C_{p_x E} & C_{p_y E} & C_{p_z E} & \sigma_E^2 \end{bmatrix}$$

KFParticle algorithm

Algorithm of the short-lived particles reconstruction can be described as follows:

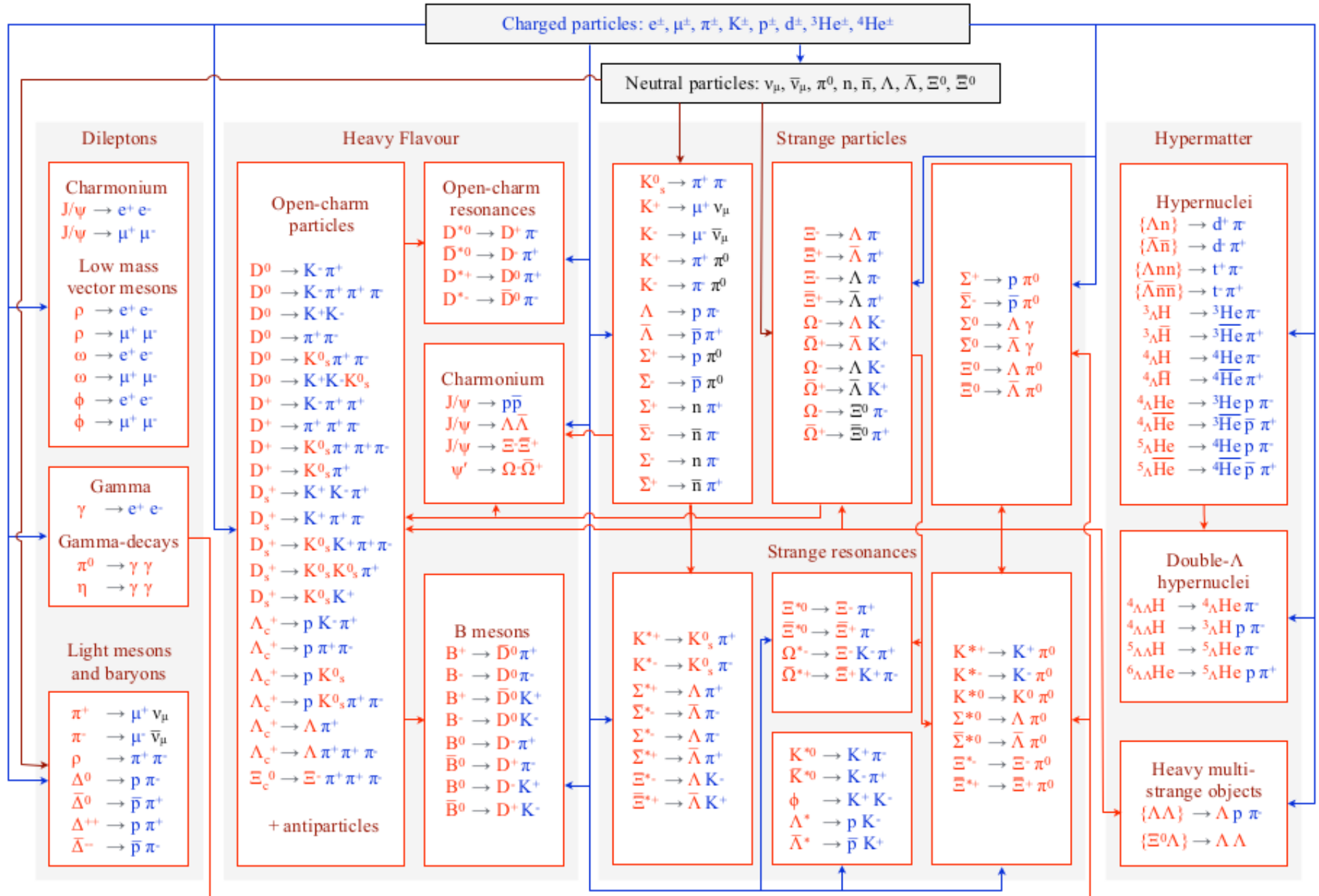
1. fit tracks with Kalman fitter (with PID hypothesis);
2. transform track parameters to KFParticle format $\mathbf{r} = (x, y, z, Px, Py, Pz, E, s)$;
3. reconstruct primary vertex (PV) with KFParticle package or provide PV itself;
4. sort tracks (primary positive, primary negative, secondary positive, secondary negative) on the base of chi2 criteria:

$$\chi_{prim}^2 = \Delta\mathbf{r}^T (C_{track} + C_{PV})^{-1} \Delta\mathbf{r},$$

where $\Delta\mathbf{r}$ – distance between track and the primary vertex position, \mathbf{C}_{track} is covariance matrix of a track and \mathbf{C}_{PV} is a covariance matrix of primary vertex;

5. use for the initial approximation of the secondary vertex the DCA between secondary tracks;
6. transport daughter particles to the secondary vertex approximation;
7. calculate momentum and its covariance matrix of the daughters at the secondary vertex position using Kalman filter algorithm and after reconstruct the mother particle.

List particles with the decay channels (more than 100 decay channels are included)



Main commands in KFParticle

1. KFTrack kfTrack - init KFParticle track and set its parameters
kfTrack.SetParameters(x,y,z, p_x, p_y, p_z)
kfTrack.SetCovarianceMatrix(C)
kfTrack.SetCharge(q)
2. KFParticle p1(kfTrack, pdgTrk) - init KFParticle particle
3. KFParticle pVtx(vert) - set KFParticle vertex
4. KFParticle K0 - init new KFParticle particle
K0.SetProductionVertex(pVtx) - set production vertex

Int_t nDaughters = 2
vDaughters[2] = {&p1, &p2} - init daughters particles
5. K0.Construct(vDaughters,nDaughters,&pVtx,-1) - construct decay particle

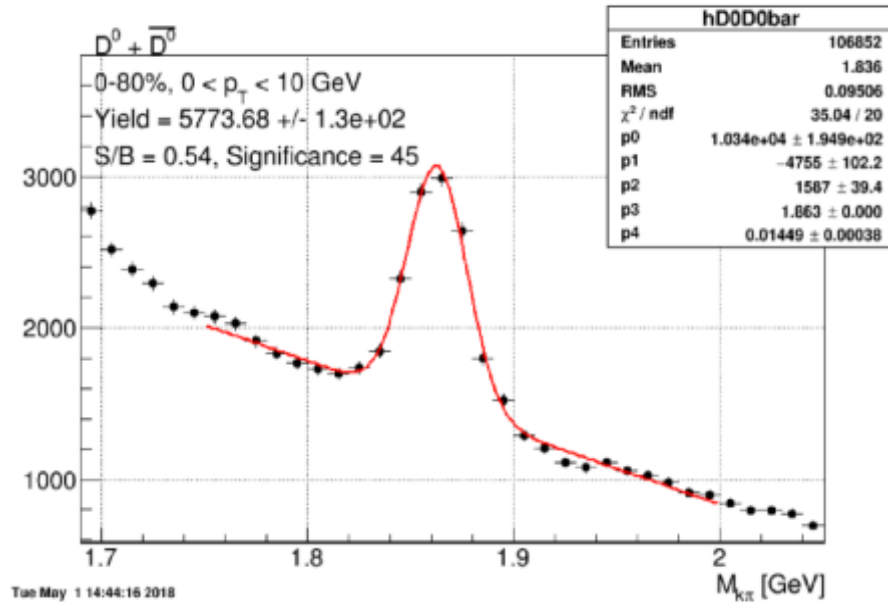
K0.TransportToProductionVertex() - parameters decay particle in production vertex
K0.TransportToDecayVertex() - parameters of particle in decay vertex
6. And also there are getters:

K0.GetMass(), K0_1.GetErrMass()
K0.GetP(), K0_1.GetErrP()
K0.GetDecayLength(), K0_1.GetErrDecayLength()
and some others

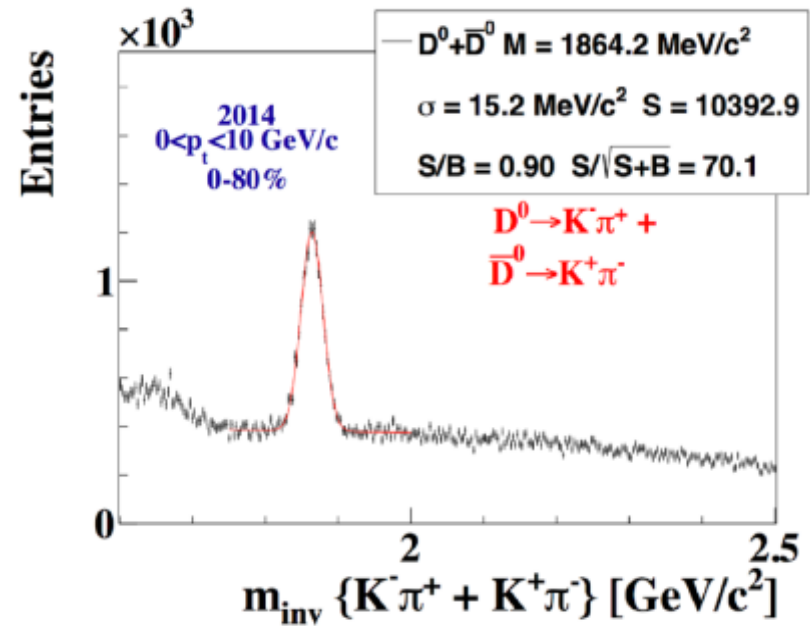
Example of using KFPARTICLE

STAR experiment

Standard reconstruction



KF Particle Finder



Summary

1. Primary vertex reconstruction program is introduced in SPDroot in 2019 on the base of CBM reconstruction algorithm
2. It shows good primary vertex reconstruction precision for the different type of simulated events
3. It also shows the similar vertex reconstruction precision with comparison others reconstruction algorithms used in HEP experiments.
4. KFParticle package is also available in SPDroot for reconstruction of the short-lived particles