EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics

Internal NOTE
ALICE Reference Number
ALICE-INT-2009-018
date of last change
October 29, 2009

Vertex reconstruction for proton-proton collisions in ALICE

E. Bruna
Yale University, New Haven CT, USA

A. Dainese
INFN - Sezione di Padova, Padova, Italy

M. Masera, F. Prino
Università degli Studi di Torino and INFN, Torino, Italy

Abstract

Reconstructing the interaction vertex is a challenging task in the low multiplicity environment of pp collisions at the LHC. The two innermost layers of the Inner Tracking System (ITS), made of pixels, allow to obtain a first estimate of the vertex position, which can be provided also in a quasi-online mode, since only the local reconstruction is used. The optimal vertex measurement is obtained after the full event processing, using the tracks reconstructed in the ALICE barrel detectors. We present the methods for primary vertex reconstruction in pp collisions using pixels and tracks reconstructed in the ITS+TPC or in the TPC only. We also show the performance of the vertex finder in reconstructing displaced vertices originated by short-lived particles like charmed mesons.

1 General strategy for primary vertex reconstruction in ALICE

Primary vertex reconstruction is one of the main requirements in the three levels of the data processing in the ALICE experiment: online, reconstruction and analysis.

At the online level, the knowledge of the interaction point is necessary to monitor the beam position and to measure the beam spread along the three coordinates x, y and z. The spread is expected to be of the order of 50–200 μm in the transverse plane (x and y coordinates) and of the order of 5 cm in the longitudinal direction (z), for proton–proton collisions. The Silicon Pixel Detector (SPD), located in the two innermost layers of the Inner Tracking System (ITS), is the ideal detector to perform fast vertex measurements and online monitoring because (a) it gives a fast response, (b) it is the closest detector to the interaction point, and (c) it has an excellent resolution in the transverse plane, due to its high granularity.

In addition, the SPD is used to provide the primary vertex position for events triggered by the Forward Muon Spectrometer, without the need of reading and reconstructing the events in the other barrel detectors.

At the reconstruction level, the position of the primary vertex given by the clusters in the SPD is needed by the Kalman filter algorithm to perform the tracking in the central barrel.

At the analysis level, a good measurement of the primary vertex improves the resolution on the impact parameters of the tracks with respect to the interaction point: this is important for studies of short-lived particles, such as those with open charm and open beauty.

Three algorithms for vertex reconstruction are discussed in this work, which updates and expands a previous note [1]. The algorithms, included in the ALICE software [2], are listed below:

- *VertexerSPDz*: it provides the measurement of the z coordinate of the interaction point by means of the SPD. It requires the knowledge of the x and y coordinates.
- *VertexerSPD3D*: it provides a three-dimensional measurement of the primary vertex by means of the SPD.
- *VertexerTracks*: it provides a three-dimensional measurement of the primary vertex by means of the reconstructed tracks.

The first two algorithms only require local reconstruction in the SPD, whereas *VertexerTracks* can only be used once the reconstructed tracks are available. In the following we summarize the structure of the event reconstruction "loop" (tracking and vertexing), and we describe the use of the three algorithms for primary vertex reconstruction.

Track reconstruction in the ALICE central barrel is performed using three subdetectors (here ordered from the inside to the outside): the Inner Tracking System (ITS) [3], which has an outer radius of $\simeq$ 45 cm, the Time Projection Chamber (TPC) [4], with outer radius $\simeq$ 250 cm, and the Transition Radiation Detector (TRD) [5], with outer radius $\simeq$ 350 cm. These detectors, which are embedded in a large solenoidal magnet providing a magnetic field of 0.5 T, allow the track reconstruction in the pseudorapidity range $-0.9 < \eta < +0.9$. In Table 1 we present the main parameters of the six layers of the ITS, since this is the detector used for vertex reconstruction.

Event reconstruction is performed in the following steps:

1. First estimate of the position of the interaction vertex using the correlation of tracklets in the SPD. The vertex position is reconstructed in the three coordinates using *VertexerSPD3D*; for the events in which this algorithm fails (mostly events with only one SPD tracklet), only the position along the beam line (z) is determined,

Table 1: Parameters of the six detector layers in the ITS [6].

Layer	Type	r[cm]	Thickness [% of X_0]	Spatial precision $r\phi \times z$ [μm^2]
1	pixel	3.9	1.14	12×100
2	pixel	7.6	1.14	12×100
3	drift	15.0	1.13	35×25
4	drift	23.9	1.26	35×25
5	strip	38.0	0.83	20×830
6	strip	43.0	0.86	20×830

using *VertexerSPDz*. The run-by-run information on the position and spread of the interaction region (diamond), if available from the Offline Condition Database (OCDB), is used by the two algorithms, as we will detail in the following. The reconstructed vertex is stored in the ESD (Event Summary Data). The algorithms and their performance are described in Section 3.

2. Track reconstruction in the TPC (inward). Track finding and fitting are performed from outside inward by means of a Kalman filtering algorithm [7]. Track candidates (seeds) are created using the information from the n outermost pad rows ($n \simeq 15$)[1)] and the position of the primary vertex as reconstructed with the SPD. A copy of the set of tracks from the TPC reconstruction is propagated to the primary vertex and stored in the ESD, in order to allow the possibility to perform a TPC-only analysis.
3. Track reconstruction in the ITS (inward). TPC reconstructed tracks are matched to the outermost ITS layer and followed in the ITS down to the innermost pixel layer. Track finding is done in two passes: during the first pass, the position of the primary vertex estimated using the SPD pixels is used to maximize the efficiency for primary tracks; during the second pass, the vertex information is not used, in order to recover the tracks with large displacement from the vertex.
4. Track back-propagation to the outermost layer of the ITS and then to the outermost radius of the TPC. Extrapolation to the TRD and track finding in the six layers of this detector. Extrapolation to outer detectors for particle identification —Time-of-Flight (TOF), High-Momentum Particle Identification Detector (HMPID), Photon Spectrometer (PHOS), Electromagnetic Calorimeter (EMCal)— and matching with hits on these detectors.
5. As a last step, reconstructed tracks are re-fitted inward in TRD, TPC, ITS and are propagated to the primary vertex reconstructed by the SPD.
6. At this stage the set of reconstructed tracks is used to determine the primary vertex position with the optimal resolution, as described in Section 5. Also a TPC-only primary vertex is reconstructed from the set of tracks with TPC-only parameters. These vertices will be used in the subsequent physics analyses. *VertexerTracks* is used in both cases, with different criteria. The two sets of tracks are finally propagated to their respective vertex and stored in the ESD, along with the two vertices.

We mentioned the possibility of using the information on the interaction diamond during vertex reconstruction. This is extremely helpful because: a) it improves the rejection efficiency for secondary tracks that would spoil the vertex resolution; b) it allows to

[1)] The possibility to create the seeds also at inner radial positions in the TPC allows to extend the acceptance to $|\eta| < 1.4$, though with poorer track quality.

determine the position of the vertex in the plane transverse to the beamline (x, y) with a resolution at most as large as the diamond dispersion, for all events; c) it allows to reconstruct the longitudinal (z) position of the vertex also for events with only one primary track. In addition, the usage of the diamond information allows to update dynamically the primary vertex position at analysis time after removing or adding one or more tracks (see Section 5.7).

For a correct estimate of the uncertainties, the diamond information is calculated specifically for each vertexing step (pixel tracklets, TPC-only tracks, ITS+TPC tracks). In the case of pixels, the diamond information is calculated quasi-online by a so-called Detector Algorithm (DA), having also the purpose of monitoring the diamond position and size at run time. This DA, based on *VertexerSPD3D*, is described in Section 4. In the case of tracks, the diamond information is extracted offline by analysing a subsample of the ESD events after a first full event reconstruction pass. The position of the centre of the diamond is obtained from a weighted average of vertex positions from high-multiplicity events. The covariance matrix of the diamond is obtained as the sum of the covariance matrix of the weighted average and the covariance matrix of the beam-beam convolution that is computed from the LHC machine parameters.

The vertexers based on tracklets reconstructed in the SPD detector, i.e. *VertexerSPDz* and *VertexerSPD3D*, provide also the capability of tagging pile-up events on the basis of multiple vertices. For each event, the vertex with the larger number of tracklets, corresponding to the interaction with higher multiplicity, is first found and stored in the ESD as the main vertex reconstructed by the SPD. If other piled-up vertices are found starting from the tracklets which do not point to the main vertex, their positions are stored in a dedicated array in the ESD file and the event is flagged as pile-up.

2 Beams and interaction point at the LHC

The two beams accelerated in the LHC will interact in the so-called intersection point (IP), ideally at (0, 0, 0) in the ALICE global frame. Assuming that the particles in the bunches have Gaussian distributions along the three coordinate axes with dispersions $\sigma^{\rm bunch}_{x,y,z}$, we can define the dispersion of the interaction region as the convolution of the particle distributions in the two intersecting bunches. The interaction vertex lies in a diamond with dimensions:

$$\sigma^{\rm vertex}_{x,y,z} = \sigma^{\rm bunch}_{x,y,z}/\sqrt{2}. \tag{1}$$

The size of the bunches at the IP depends on the *transverse emittance* ϵ (a beam quality parameter) and on the value of the *amplitude function* β at the IP, indicated as β^*, which is determined by the accelerator magnet configuration. We have:

$$\sigma^{\rm bunch}_{x,y,z} = \sqrt{\frac{\epsilon_{x,y,z}\,\beta^*}{\pi}}. \tag{2}$$

In Table 2 we recall the LHC machine nominal parameters at the ALICE IP for pp and Pb–Pb [8].

For pp runs the nominal luminosity of 5×10^{32} cm^{-2}s^{-1} will have to be reduced to $< 3 \times 10^{30}$ cm^{-2}s^{-1}, in order to limit the pile-up in the TPC and in the Silicon Drift Detector (SDD). Such reduction can be achieved in two ways: either by increasing the value of β^* or by displacing the two beams in the transverse plane to make a collision between the tails of the particle distributions. If the first option is chosen, β^* might be increased up to 100 m; this would broaden of a factor $\sqrt{100\ \text{m}/10\ \text{m}} \simeq 3$ the transverse

Table 2: LHC parameters for pp and Pb–Pb runs for ALICE [8].

Parameter		pp			Pb–Pb
$\sqrt{s_{\rm NN}}$	[TeV]	0.9	10	14	5.5
β^*	[m]	10	10	10	0.5
$\sigma_{x,y}^{\rm bunch}$	[μm]	280	84	71	16
$\sigma_{z}^{\rm bunch}$	[cm]	10.5	5.4	7.5	7.5
$\sigma_{x,y}^{\rm vertex}$	[μm]	198	59	50	11
$\sigma_{z}^{\rm vertex}$	[cm]	7.4	3.8	5.3	5.3
Luminosity	[cm^{-2}s^{-1}]	$\sim 10^{27}$	$\sim 10^{29}$	5×10^{32}	5×10^{26}

size of the interaction diamond, up to $\simeq$ 150 μm. If the second option is necessary, the beams might be displaced to a distance of $\simeq 4-5\ \sigma_{x,y}^{\rm bunch}$ and the collisions would occur in the tails at 4-5 σ from the centre of the beams: these tails will most likely be non-Gaussian and the transverse size of the interaction diamond may be even larger than 150 μm.

3 Primary vertex finder with the Silicon Pixel Detector

The two methods for the determination of the primary vertex coordinates implemented in AliRoot, *VertexerSPDz* and *VertexerSPD3D*, are based on "tracklets" built by associating pairs of reconstructed points in the two SPD layers. A description of the two algorithms, with the corresponding results, is given in the following subsections.

3.1 VertexerSPDz

This vertexer provides a measurement of the z-coordinate of the primary vertex assuming that the beam position in the transverse plane is known with an accuracy of the order of 200 μm or better, e.g. from the interaction diamond measured at online level (see Sections 3.3.2 and 4 for details). The algorithm for the estimation of the z-coordinate of the primary vertex consists in the correlation of reconstructed points belonging to the first layer of the SPD with reconstructed points belonging to the second one within a small azimuthal window (by default set to $\Delta\varphi_{12}$=0.01 rad). Matching pairs of points on the two layers define candidate tracklets. In this way, only high momentum tracks, i.e. straight lines in the bending plane, are selected, the combinatorial background is reduced and only the tracks less affected by multiple scattering are considered.

For each candidate tracklet i the intersection point z_i with the beam axis is calculated. First, a Region Of Interest (ROI) around the peak in the distribution of z_i coordinates is defined. Then, a first guess of the position of the interaction point is given by the weighted mean of the z_i coordinates falling in the selected ROI:

$$z_{\rm mean} = \frac{\sum_i^N z_i/\sigma_i}{\sum_i^N 1/\sigma_i} \tag{3}$$

where N is the number of tracklets and σ_i are the errors on the z_i points given by the pixels. The calculation of $z_{\rm mean}$ is iterated re-centering at each step the ROI on the value of $z_{\rm mean}$ from the previous iteration until a symmetric region around $z_{\rm mean}$ is reached. This allows to minimize possible biases due to asymmetries in the tails of the z_i distributions.

The choice of the azimuthal window is especially important for pp collisions with low multiplicity, where a too strict selection would considerably reduce the statistics. There-

fore, to gain efficiency, if the vertex is not found with the default selection $\Delta\varphi_{12}$ <0.01 rad, the vertex-finding procedure is iterated enlarging at each step the azimuthal windows. The number of iterations and the $\Delta\varphi_{12}$ used in each iteration can be configured by the user. By default 3 iterations are performed reaching a maximum $\Delta\varphi_{12}$ of 0.2 rad.

3.2 VertexerSPD3D

This vertexer provides a three-dimensional measurement of the primary vertex based on SPD tracklets. The algorithm can be divided in three main steps, which are repeated two times (first and second pass):

1. *Tracklet finding*: only the tracklets which satisfy the following requirements are selected. The tracklet must cross a cylindrical fiducial region where the interaction point is expected to be located. The size and the centre of this fiducial region, as described in the following, are changed between the first and the second vertex reconstruction pass. The two points in each tracklet must be within a given azimuthal window and the tracklet must not be too displaced with respect to the centre of the fiducial region. The applied cut values on the azimuthal distance between the two points and tracklet displacement are tighter in the second pass.
2. *Tracklet selection*: this selection procedure is applied to pairs of tracklets. After a cut on the distance of closest approach between the two tracklets (DCA< 1 mm), the crossing point $\mathbf{c}_{ij}$ of the pair ij is calculated; only the pairs with the crossing point in the fiducial region are kept. A three-dimensional histogram is filled with all the points $\mathbf{c}_{ij}$ and the peak (i.e. the bin with maximum density of tracklet intersections) is found. Tracklets far from the peak (distance larger than 1 mm along x and y and 8 mm along z) are removed.
3. *Vertex determination*: A first estimate of the primary vertex is calculated with the tracklets passing the selection cuts described above. The coordinates of the vertex are given by finding the point of minimum distance among the tracklets as described in detail later in this section. The coordinates of the primary vertex are re-calculated after a further selection on the tracklets based on their displacement from the first estimation of primary vertex (tracklets with distance $>$ 1 mm from the found vertex are removed).

As stated above, the entire procedure (steps 1, 2 and 3) is repeated twice. In the first iteration a wide fiducial region centred on the average beam position provided by the quasi-online DA (see Section 4) and stored in the OCDB is used for tracklet selection, while in the second iteration a smaller fiducial region centred on the position of the primary vertex found in the previous iteration is defined. Finally, an additional check on the position (the vertex must be located inside the beam pipe) and on the number of contributing tracklets (> 0) is performed. The default values for the parameters used in the tracklet selection in the two vertex reconstruction passes are summarized in Table 3. It should be noted that all these parameters can be configured when launching the reconstruction by means of the AliITSRecoParam object, which defines the configuration for the ITS local reconstruction, SPD-based vertexing and tracking.

The calculation of the vertex coordinate from a sample of selected tracklets is done by minimizing the quantity:

$$D^2 = \sum_i^N d_i^2 \tag{4}$$

Table 3: Default values of the parameters of the *VertexerSPD3D* algorithm.

Variable		1st pass	2nd pass
Radius of fiducial region	R_{fid} (cm)	2.5	0.5
Width of fiducial region	Δz_{fid} (cm)	40	0.5
Azimuthal window for tracklet definition	$\Delta\varphi_{12}$ (rad)	0.5	0.025

where N is the number of tracklets used to calculate the vertex, d_i is the distance between the tracklet i and the vertex (x_0, y_0, z_0), weighted by the errors on the tracklets:

$$d_i^2 = \left(\frac{x_i - x_0}{\sigma_{xi}}\right)^2 + \left(\frac{y_i - y_0}{\sigma_{yi}}\right)^2 + \left(\frac{z_i - z_0}{\sigma_{zi}}\right)^2 \tag{5}$$

The errors on the tracklets (σ_{xi}, σ_{yi} and σ_{zi}) are affected by three different contributions:

- Error on the reconstructed points, which accounts for the resolution of the SPD detectors. It is taken from the diagonal elements of covariance matrix of the reconstructed points (c_{xx}, c_{yy}, c_{zz}). These errors are multiplied by a geometrical factor:

$$F = \frac{r_{\mathrm{Lay2}} + r_{\mathrm{Lay1}}}{r_{\mathrm{Lay2}} - r_{\mathrm{Lay1}}} \tag{6}$$

  where r_{Lay2} and r_{Lay1} are the radii of the two SPD layers. This geometrical factor accounts for the error on the prolongation of the tracklet close to the interaction point: due to the above covariance matrices, the tracklet has a larger uncertainty in the proximity of the primary vertex.
- Curvature effect: since the tracklet is a straight line, the curvature of the originating particle inside the magnetic field is neglected. The discrepancy between the bent trajectory and the straight line in the vicinity of the interaction point can be calculated geometrically starting from the radii of the SPD layers and the curvature radius of the particle trajectory. However, the curvature radius depends on the particle momentum that can not be calculated from the SPD reconstructed points. Hence, the mean transverse momentum of particles producing SPD tracklets passing the *VertexerSPD3D* selection cuts described above is used to calculate such correction. At $\sqrt{s}$=14 TeV and B=0.5 T, a value of $\langle p_T^{\mathrm{sel.tracklets}}\rangle$=630 MeV/$c$ is obtained, corresponding to a curvature radius of about 4 m.
- Multiple scattering: this source of error is due to the passage of the charged particle in the beam pipe and in the first layer of the pixel detector. The deflection angle of multiple scattering is calculated as:

$$\delta\theta = \frac{14.1}{\beta p}\sqrt{x/X_0} \tag{7}$$

  where β and p [GeV/c] are the velocity and the particle momentum, and x/X_0 is the material thickness in unit of radiation length (assumed to be 1% for SPD inner layer and 0.2% for the 800 μm Be thickness of the beam pipe). Also in this case, it is necessary to estimate the average momentum p and velocity β of the particles producing tracklets that pass the vertexer selection cuts.

The three sources of error listed above are added in quadrature to obtain the errors σ_{xi} and σ_{yi} on the transverse coordinates of the i-th tracklet in the vicinity of the interaction

point. For the z coordinate, which is not affected by the magnetic field, only the contributions of SPD reconstructed point resolution and multiple scattering are considered in the computation of σ_{zi}. The coordinates of the vertex are given with their errors, calculated from the weight matrix associated to each tracklet:

$$\mathbf{W}_i = \begin{pmatrix} \sigma_{xi}^{-2} & 0 & 0 \\ 0 & \sigma_{yi}^{-2} & 0 \\ 0 & 0 & \sigma_{zi}^{-2} \end{pmatrix} \tag{8}$$

The covariance matrix associated to the vertex is then given by inverting the sum over the N tracklets of the weight matrices:

$$\mathbf{C}_v = \left(\sum_i^N \mathbf{W}_i \right)^{-1} \tag{9}$$

3.3 Results

3.3.1 VertexerSPD3D

The performance of the interaction vertex determination with the SPD depends on a) the event multiplicity, that determines the number of tracklets entering the vertex calculation; b) the position along the z axis of the interaction vertex (z_{vertex} is related with SPD acceptance, while x_{vertex} and y_{vertex} affect the distance of the clusters from the beam axis), c) the value of the magnetic field that determines the track curvature and, consequently, the goodness of the straight line approximation inherent in the usage of tracklets; d) the level of spatial misalignment of the SPD sensitive elements, i.e. the difference between the ideal position of the sensors and their real position. Several pp Monte Carlo samples have been generated to investigate the performance of the SPD-based vertexers when varying these conditions. The events have been simulated with the PYTHIA event generator [9] and track propagation in the ALICE setup has been done with GEANT 3.21 [10]. The SPD digitization and clusterization as well as the calculation of the vertex from the SPD reconstructed points have been done using the official software framework AliRoot [2] (rev. 32456). The list of SPD dead and noisy channels (which has an important effect on the vertexer performance) is taken from the OCDB of the same aliroot revision which contains 15 dead half staves for a total number of ≈ 1.2 millions of bad pixels.

The benchmark study has been performed on a sample of 30'000 pp events generated at $\sqrt{s} = 14$ TeV with ideal SPD geometry (i.e. no misalignment effects), beam centred in the nominal position (0,0) in the transverse plane and magnetic field B=0.5 T. In order to study the performance also for large values of z, the z coordinate of the vertex has been generated according to a Gaussian with r.m.s. of 12 cm (much larger than the nominal r.m.s. of the diamond, 5.3 cm at this energy). The fiducial region for vertex search was set to a cylinder centred on the average (x, y, z) position of the generated events, namely (0,0,0), with 1 cm radius and $\Delta z = \pm 40$ cm. In Fig. 1 (left), the ratio between the number of events with reconstructed SPD vertex and the number of generated PYTHIA events is displayed as a function of the true position of the generated vertex along z (z_{MCvertex}) for both the 3D and the z SPD vertexers. Figure 1 (right) shows the fraction of events with reconstructed (3D or z) SPD vertex normalized to the number of events that pass the loosest ALICE minimum bias trigger selection (MB1: SPD FastOR || V0C || V0A, i.e. signal in the SPD, $-1.96 < \eta < 1.96$, or in C-side V0 detector, $-3.7 < \eta < -1.7$, or in the A-side V0 detector, $2.8 < \eta < 5.1$). It can be seen that, indeed, the vertex is reconstructed for most of the triggered events.

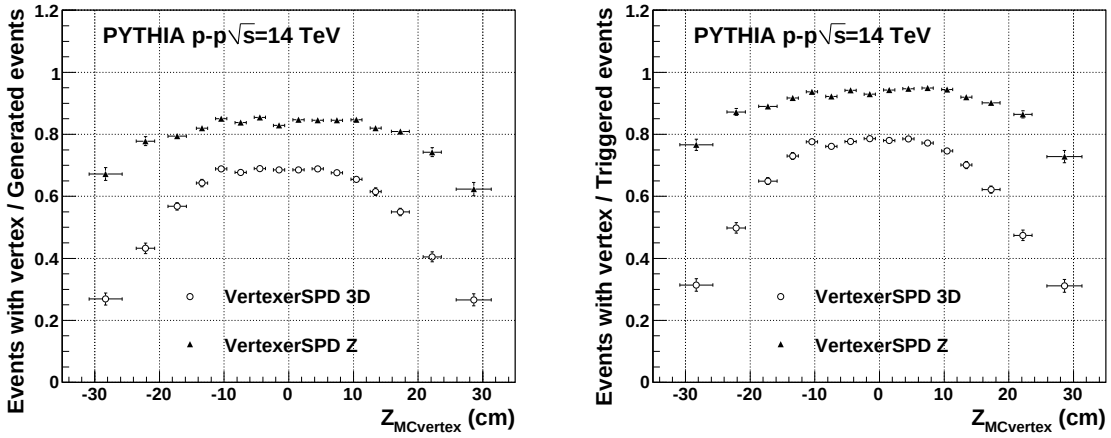

Figure 1: Efficiency of vertex reconstruction for *VertexerSPDz* and *VertexerSPD3D* as a function of the z coordinate of the generated vertex normalized to generated (left) and MB1 triggered (right) events.

The lower efficiency of the *VertexerSPD3D* is due both to the requirement of having at least two tracklets in order to calculate the vertex position (while one tracklet is sufficient in the *VertexerSPDz*) and to the tighter tracklet selection procedure (see Section 3.2). For this reason, the *VertexerSPDz* algorithm is called by the *VertexerSPD3D* when the 3D vertex reconstruction fails. In this way, it is possible to recover the z coordinate of the interaction vertex for about 10–20% (depending on $z_{\rm MCvertex}$) of events. The dependence of vertexing efficiency on $z_{\rm MCvertex}$ is due to the fact that the pseudorapidity coverage of the SPD (and therefore the number of particles crossing the pixel detectors) depends on $z_{\rm MCvertex}$. Notably the decrease of efficiency for the *VertexerSPD3D* starts to occur at about $|z_{\rm MCvertex}| = 14$ cm corresponding to the physical size of the SPD detector.

This acceptance effect is visible in Fig. 2 (left) where the number of contributors (i.e. the number of tracklets used to calculate the vertex position) is shown as a function of $z_{\rm MCvertex}$. In Fig. 2 (right) the distribution of the number of contributors is shown for the case of *VertexerSPD3D* only and for the case when the *VertexerSPDz* is called after the 3D for the events where the 3D algorithm fails. It can be seen that the events recovered by the *VertexerSPDz* are concentrated at low values of number of contributors (i.e. of event multiplicities). The negative values of number of contributors correspond to the cases where also the *VertexerSPDz* algorithm fails either due to the absence of reconstructed points in the SPD (contributors $= -2$) or to the impossibility of building suitable tracklets (contributors $= -1$).

The lowering of the number of contributors with increasing $|z_{\rm MCvertex}|$ is reflected in a reduction of the resolution on the vertex determination (especially for the z coordinate), as it can be seen in Fig. 3 (right) where the r.m.s. of the residuals (i.e. the differences between reconstructed and generated vertex position) is displayed as a function of $z_{\rm MCvertex}$ for the three coordinates of the vertices reconstructed by the 3D algorithm. It has to be pointed out that, since the resolution on the vertex position depends on the number of contributors, in each $z_{\rm MCvertex}$ bin the residual distributions are not Gaussian shaped, but they are rather a sum of Gaussians with different sigmas corresponding to different number of contributors. The better resolution along the z coordinate with respect to the transverse coordinates, despite the larger pixel size along z, is due to the fact that the par-

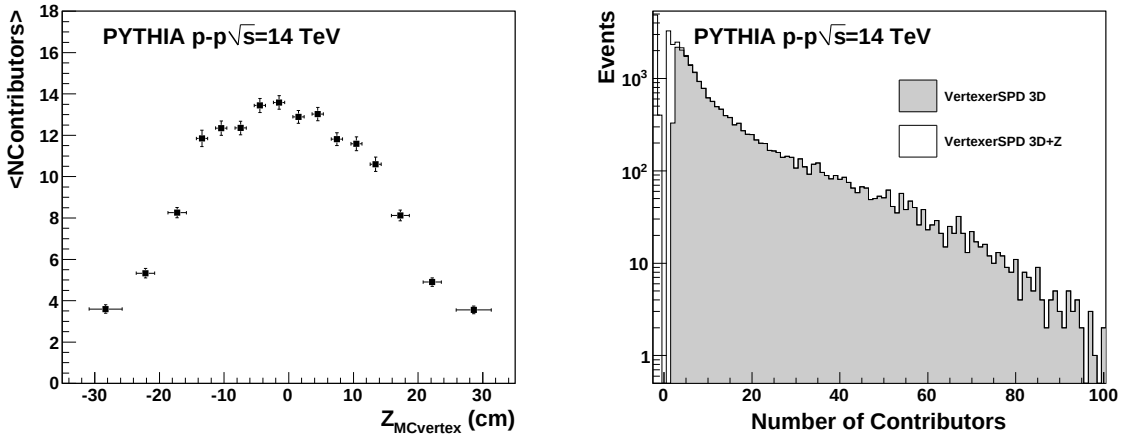

Figure 2: Left: Average number of contributors as a function of the z coordinate of generated vertex. Right: number of contributors for *VertexerSPD3D* and for the case of calling the *VertexerSPDz* after the 3D to recover the events in which the 3D procedure fails. The number of contributors is set to a negative value when also the *VertexerSPDz* fails (see text for details)

ticle trajectory in the transverse plane is not a straight line because of the curvature given by the magnetic field. In the left panel of Fig. 3, the average of the residual distribution is plotted as a function of $z_{\rm MCvertex}$. In the region with optimal resolution ($|z_{\rm MCvertex}| < 14$ cm) the average of residuals is compatible with zero, demonstrating that the vertex determination is unbiased. Deviations from zero start to appear for larger values of $z_{\rm MCvertex}$, in correspondence with the decreasing of number of contributors and the consequent loss of resolution.

In order to limit these acceptance effects, the investigation of the performance (efficiency and resolution) of the *VertexerSPD3D* and the *VertexerSPDz* as a function of generated multiplicity has been done only on the sub-sample of events with $|z_{\rm MCvertex}| = 10$ cm (corresponding to $\approx 2\sigma_z^{vertex}$ of the expected diamond size for pp collisions at 14 TeV). The fraction of events with reconstructed vertex (normalized to the number of generated PYTHIA events) is shown in Fig. 4 (left) as a function of the generated multiplicity (defined as the number of charged and stable primary particles generated in $|\eta| < 1.5$). The efficiency of primary vertex reconstruction increases with the event multiplicity and, in case of *VertexerSPDz*, reaches 100% for events with more than 3 generated particles. In the right panel of Fig. 4 the correlation between the number of tracklets used in vertex calculation (contributors) and the generated multiplicity is displayed. A cut to select tracklets with small $\Delta\phi$ between reconstructed points on SPD layers 1 and 2 is applied in order to improve the precision on vertex determination. This selection excludes the low $p_{\rm t}$ particles (more affected by multiple scattering and bending in the magnetic field) from the vertex calculation and causes the number of contributors to be systematically smaller than the generated multiplicity in the SPD acceptance.

The resolution of the 3D vertexer, expressed both as r.m.s. of the residual distributions and as the sigma of a Gaussian fit to these distributions, is shown in Fig. 5 as a function of the event multiplicity (quantified by the number of reconstructed SPD tracklets) again for events with $|z_{\rm MCvertex}| < 10$ cm. The two methods to evaluate the resolution are in good agreement, indicating that deviations from the normal distribution

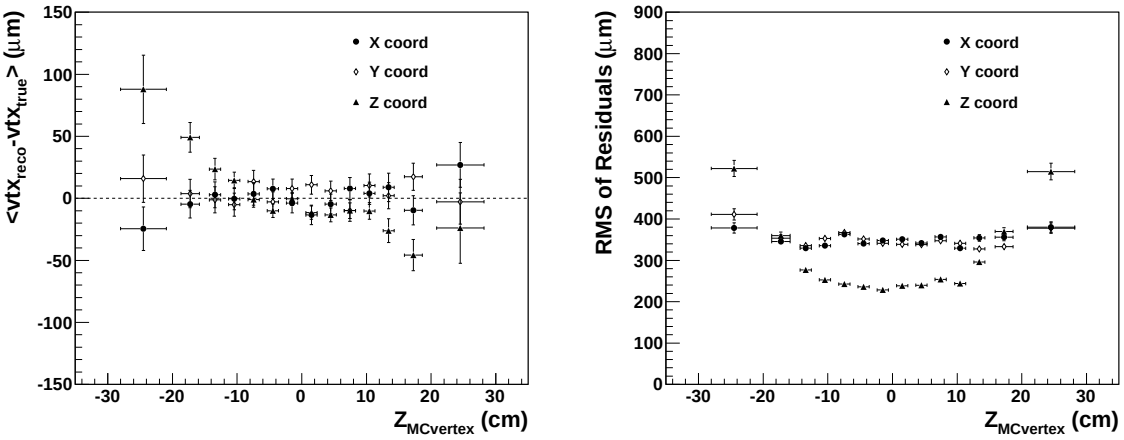

Figure 3: Average (left panel) and r.m.s. (right panel) of residuals of the reconstructed vertex coordinates as a function of the z coordinate of the generated vertex.

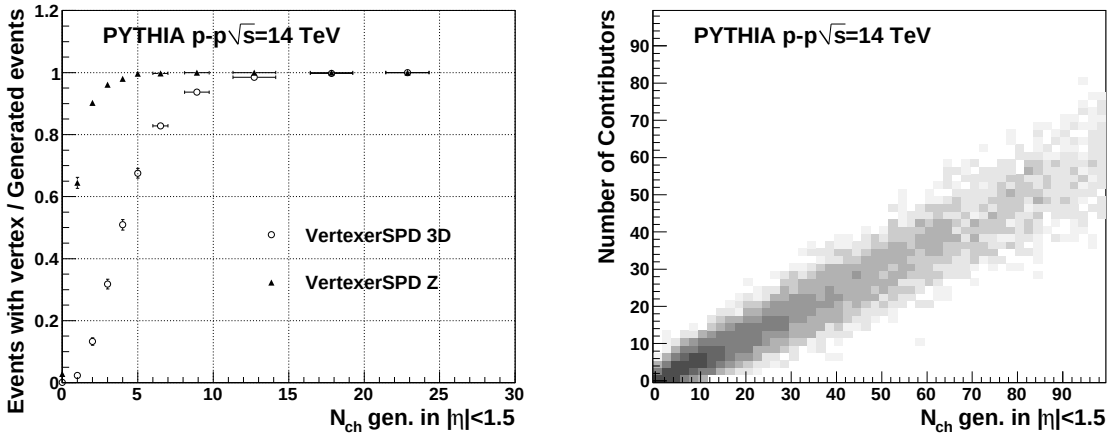

Figure 4: Left: efficiency of vertex reconstruction for *VertexerSPDz* and *VertexerSPD3D* as a function of the generated event multiplicity, for events with $|z_{\mathrm{MCvertex}}| = 10$ cm . Right: Correlation between the number of contributors and the event multiplicity.

in the residuals are negligible. As expected, the resolution improves with increasing multiplicity, reaching a saturation value of about 120 μm for x and y and about 70 μm for z in the largest multiplicity bin.

In Fig. 6 (left), the average of the residuals is shown to be compatible with zero for all multiplicities. Finally, in Fig. 6 (right) the r.m.s. of the distributions of standardized residuals (i.e. residual divided by error), the so-called pull, is reported as a function of the number of SPD tracklets. A value of pull ≈ 1 and getting closer to 1 with increasing multiplicity is observed, thus allowing to conclude that the errors on the vertex coordinates are properly evaluated.

One of the design goals of the 3D vertex determination is to have a tool able to reconstruct the vertex for any possible displacement of the transverse position of the beam axis within the beam pipe. For this reason, the performance of the *VertexerSPD3D* has been checked also against shifts of the beam position in the transverse plane, by analyzing samples of pp events generated with the average x position shifted by 1, 5 and

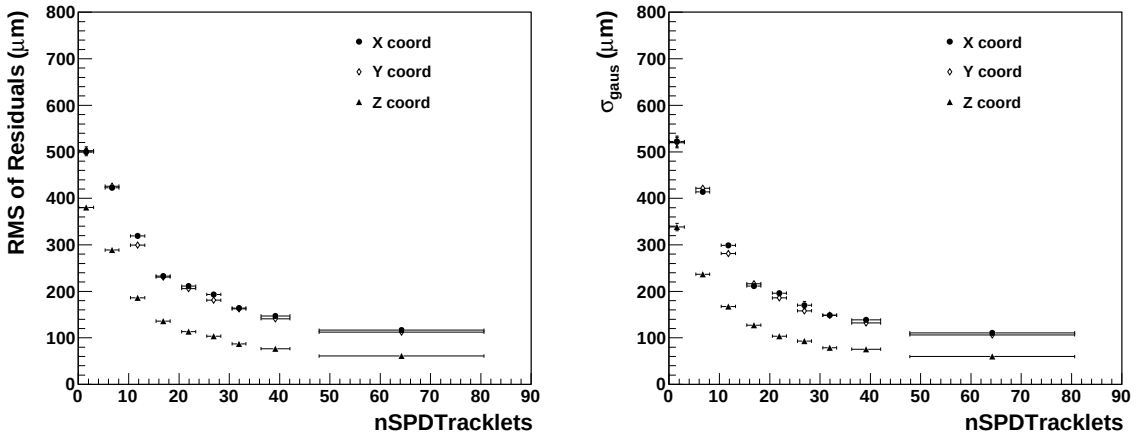

Figure 5: r.m.s. (left) and σ of Gaussian fit (right) of the residuals for the x, y and z coordinates as a function of the number of SPD tracklets (event multiplicity).

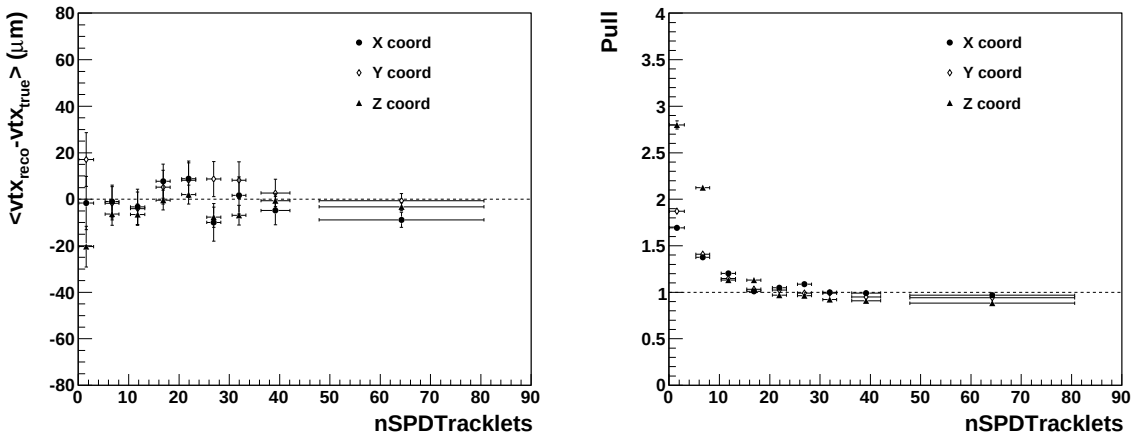

Figure 6: Average (left) and pull (right) of the residual distributions for the three coordinates as a function of the SPD tracklet multiplicity.

10 mm, while keeping $\langle y_{\rm beam}\rangle$=0. For this study, events with $|z_{\rm MCvertex}| < 5.3$ cm have been considered and the radius of the cylindrical fiducial region for vertex search has been extended to 2 cm, in order to accommodate for the largest displacement under test. In Fig. 7 (left) we report the average of the residuals (multiplicity and $z_{\rm MCvertex}$ integrated) as a function of the applied shift to the average x coordinate of the generated beam position. It can be seen that the three coordinates of the vertex are reconstructed without biases in all the generated beam positions. In the right panel of Fig. 7, the sigma of a Gaussian fit to the residuals along the x coordinate is plotted as a function of the event multiplicity (quantified by the number of reconstructed SPD tracklets) for the four studied values of $\langle x_{\rm beam}\rangle$. A similar performance is obtained also for y and z coordinates. We can therefore conclude that the resolution on the vertex determination is not affected by the shift of the transverse position of the beam.

The applied solenoidal field in the central barrel is also affecting the performance of the vertexer since it changes the curvature of the tracks and consequently the effect of the approximation of the helix with a straight line as well as the number of selected tracklets

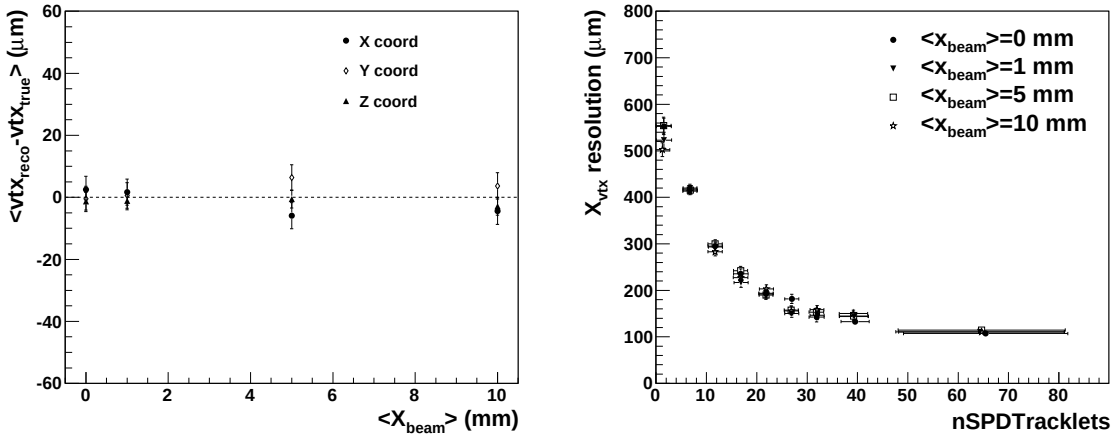

Figure 7: Left: average of residuals as a function of the beam displacement in the transverse plane. Right: x resolution as a function of the number of SPD tracklets for four different x positions of the beam in the transverse plane.

for vertex determination. The comparison of the resolution for x and z coordinates in the cases of magnetic field of 0.5 T (nominal value), 0.2 T and 0 T is presented in Fig. 8. For the x coordinate, we find an improvement of the resolution when lowering the magnetic field which is a consequence of the decreasing track curvature. For the z coordinate one would expect no effect from the magnetic field since particle trajectories are essentially straight lines in the plane (r, z). The small improvement of the resolution that is observed for z coordinate at low multiplicities is actually due to the fact that, for lower values of B, a larger number of tracklets passes the selection cuts (on $\Delta\varphi_{12}$ and R_{fid}) and is used in the vertex calculation, thus improving the resolution on the vertex position also along z. The larger number of tracklets brings also to a slightly larger efficiency of vertex calculation at lower fields, as it can be seen in Table 4.

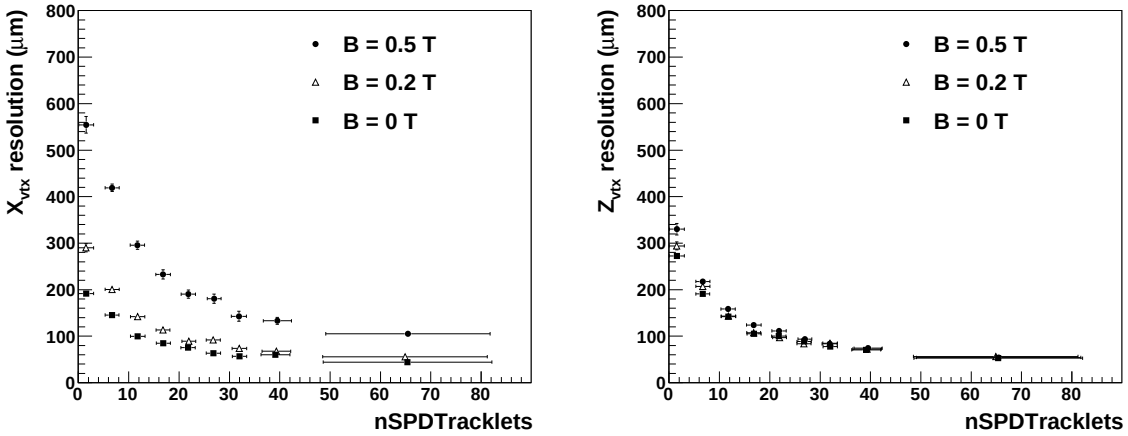

Figure 8: Resolution in x (left) and z (right) as a function of the number of SPD tracklets for three values of the magnetic field.

The performance at different magnetic fields is also a benchmark for the error calculation on the vertex coordinates. The error on the vertex is determined by three

components (as described in Section 3.2): the error on the cluster (due to the SPD intrinsic resolution), the contribution from the track curvature (which strongly depends on the magnetic field) and the contribution from multiple scattering. The last contribution depends on the average momentum of the particles that cross the SPD and pass the selection cuts, which is affected by the intensity of the magnetic field. The relative weight of these contributions is different for the z coordinate, dominated by the error on SPD resolution, and the x and y coordinates, dominated by the curvature contribution at 0.5 T and by the multiple scattering at 0 T. In Fig. 9 the pulls for the x and z coordinates are reported as a function of multiplicity for the three considered values of the magnetic field. The values of pull ($\approx$1) for all fields for events with at least 10 tracklets show that the various error components are properly taken into account. The slighly larger pull obserevd for B=0 T may be due to the fact that the contribution of multiple scattering (which is negligible in case of B$\neq$0) is somewhat underestimated. This is possibily due to the needed hypothesis on the average momentum of the particles corresponding to tracklets.

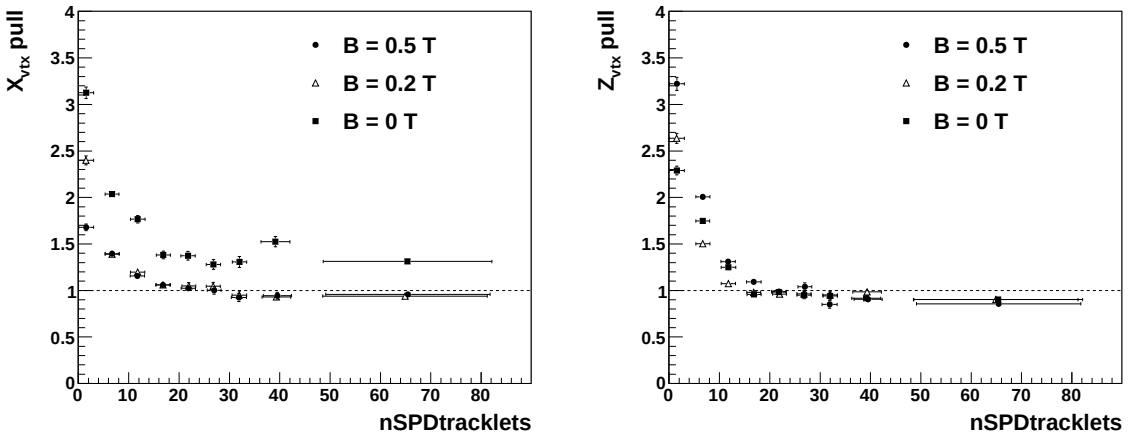

Figure 9: Pulls for x (left) and z (right) as a function of the number of SPD tracklets for three values of the magnetic field.

The performance of the *VertexerSPD3D* at lower LHC collision energies is affected essentially by the lower average multiplicity with respect to the nominal $\sqrt{s}$ = 14 TeV and, to a lower extent, by the different size of the interaction region along the z axis. The performance of the vertexer has been studied at three values of $\sqrt{s}$ = 14 TeV, 10 TeV (the expected energy for the first long data taking) and the injection value of 900 GeV (the energy of the very first collisions). The diamond sizes used in this study are summarized in Table 2.

The lower multiplicity of charged particles produced in collisions at lower $\sqrt{s}$ results in a lower number of tracklets used in vertex calculation (contributors) and consequently in a lower efficiency of the vertexer. This effect can be seen in Fig. 10, which shows the distributions of the number of contributors at the three considered energies (left), as well as the fraction of events with found 3D vertex normalized to the number of generated events as a function of z_{MCvertex} (right). It can also be noticed that the difference between 10 TeV and 14 TeV is negligible.

It should be noted that the dependence of the efficiency on collision energy and magnetic field is different between *VertexerSPDz* and *VertexerSPD3D*. Therefore, the number of vertices found by SPD in three dimensions and only along z changes with the colli-

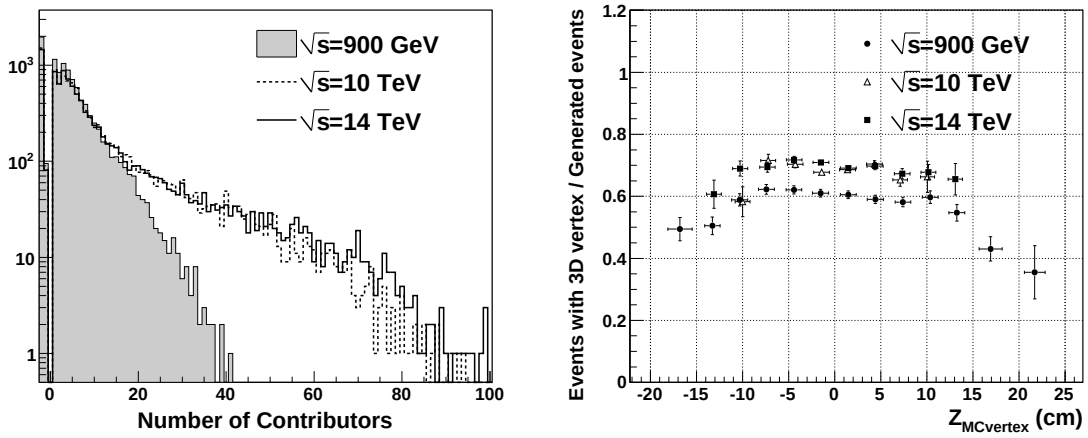

Figure 10: Left: distributions of number of contributors at three LHC energies. Right: fraction of events with vertex as a function of z_{MCvertex}.

sion energy. The average efficiencies of *VertexerSPDz* and *VertexerSPD3D* (normalized to generated PYTHIA events) in the region $z_{\mathrm{MCvertex}} < 10$ cm are summarized in Table 4 together with the fraction of SPD reconstructed vertices where the 3D determination is available. To quantify the effect of SPD dead pixels, also the efficiency obtained with ideal SPD detector is reported.

Table 4: Vertex reconstruction efficiency using SPD tracklets.

Sample	$\frac{\text{VertexSPD}}{\text{Generated}}$	$\frac{\text{Vertex3D}}{\text{Generated}}$	$\frac{\text{Vertex3D}}{\text{VertexSPD}}$
0.9 TeV, B=0.5 T	79.7%	60.3%	75.7%
10 TeV, B=0.5 T	84.2%	68.9%	81.8%
14 TeV, B=0.5 T	84.4%	68.4%	81.0%
14 TeV, B=0.2 T	84.8%	71.3%	84.1%
14 TeV, B=0 T	85.0%	72.6%	85.4%
14 TeV, B=0.5 T, ideal SPD	86.1%	73.9%	85.8%

Concerning the resolution on the vertex determination at different energies, it should be noted that the lower average number of contributors at lower energies leads to a worse resolution on average because the variable governing the resolution is the number of contributors. However, no significant difference on the resolution is expected when comparing the vertexer performance at the three considered energies at fixed number of contributors. This is confirmed by the resolution as a function of event multiplicity (quantified by the number of reconstructed SPD tracklets) shown in Fig. 11 for x (left) and z (right) coordinates.

Finally, let us discuss the effect of SPD modules misalignment on the vertex determination. A misalignment of SPD modules below 50 μm has no effect on the vertex resolution, as expected from the resolution of the 3D vertex position. A worsening of the resolution is observed when random translations and rotations with a Gaussian sigma of 100 μm are applied to the SPD volumes. A level of residual misalignment of ≈ 8 μm is

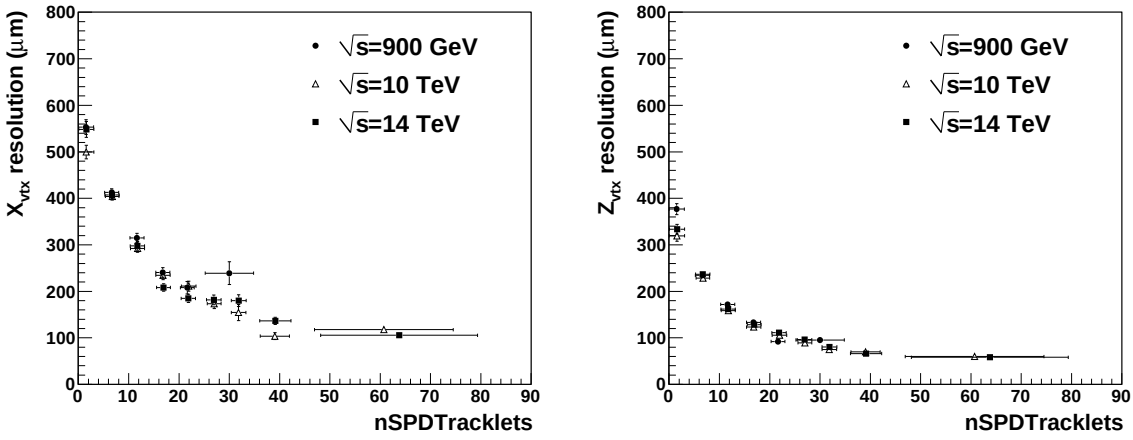

Figure 11: Resolution for the x (left) and z (right) coordinates as a function of event multiplicity at three LHC energies.

expected to be obtained for the $r\phi$ position of the SPD sensor after the re-alignment with tracks from cosmic-ray and p–p events. Therefore, the vertexer performance will not be affected significantly by the residual misalignment. It is also possible to use the *VertexerSPD3D* to check the quality of the alignment. This can be obtained by calculating for each event the vertex position from different sub-samples of SPD modules (for example, the top and bottom hemi-spheres of the SPD barrel) and comparing the results. Detailed studies in this direction are currently ongoing.

3.3.2 VertexerSPDz

The performance of the *VertexerSPDz* has been evaluated on the sample of 30'000 pp events at 14 TeV generated with $\sigma_z^{vertex} = 12$ cm. The efficiency of the *VertexerSPDz* has been already shown in Figs. 1 and 4, when discussing the need of calling the *VertexerSPDz* to recover some of the events for which the 3D determination fails. The higher efficiency of the *VertexerSPDz* is due both to the possibility of reconstructing the z of the vertex with only one tracklet (while at least two are needed for a 3D reconstruction) and to the less severe tracklet selection with respect to the *VertexerSPD3D*.

The average of the residuals is shown in Fig. 12 as a function of $z_{\rm MCvertex}$ for all the generated events (left) and as a function of multiplicity (number of SPD tracklets) for events with $|z_{\rm MCvertex}| < 10$ cm, which are not affected by SPD acceptance limitations (right).

The resolution on the z coordinate (σ of the Gaussian fit to the residual distribution) is shown as a function of multiplicity in Fig. 13 (left) and compared to that obtained with *VertexerSPD3D*. In both cases, a cut $|z_{\rm MCvertex}| < 10$ cm has been applied to select events free from SPD acceptance effects.

In the right panel of Fig. 13, the pull for the z coordinate is shown as a function of the multiplicity, for events with $|z_{\rm MCvertex}| < 10$ cm. The value of the pull is only slightly larger than 1, indicating that the error evaluation is quite satisfactory and only marginally affected by the fact of assuming the vertex to be at $(x = 0, y = 0)$ in the transverse plane.

The slightly better resolution obtained with *VertexerSPDz* is due to the fact that the beam axis (i.e. the x and y coordinates of the vertex) is set at its known average transverse position and the beam smearing in the transverse plane (50 μm) is much smaller than

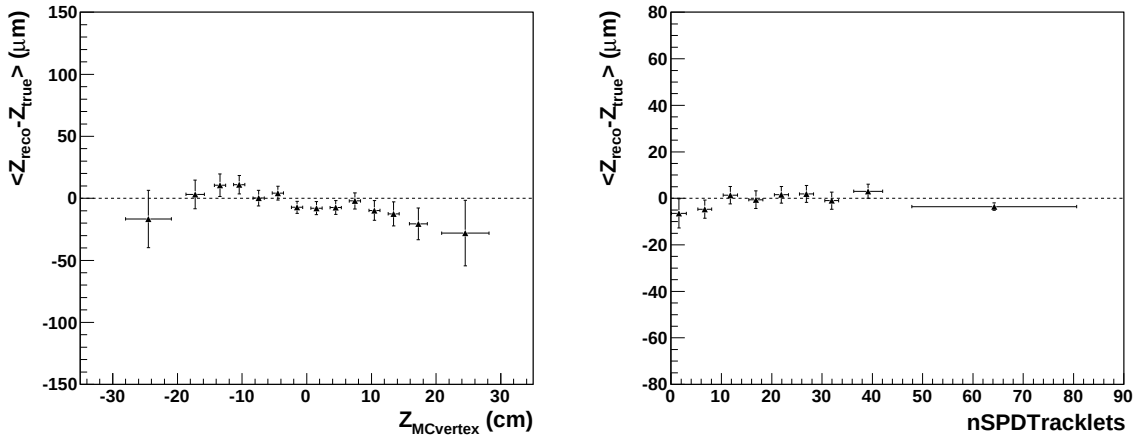

Figure 12: Average of residuals from *VertexerSPDz* as a function of $z_{\rm MCvertex}$ (left) and of the number of SPD tracklets (right). In the right-hand panels, only events with $|z_{\rm MCvertex}| < 10$ cm have been considered.

the *VertexerSPD3D* resolution in x and y. However, it should be pointed out that the performances of *VertexerSPDz* are quite sensitive to the knowledge of the beam centre position in the transverse plane. A dedicated Monte Carlo study has been carried out to make a quantitative statement on this issue: a sample of 500 events, generated with a parametrized generator at a fixed multiplicity of 50 charged particles per event, was used for a simulation in the ALICE apparatus with $x_{MCvertex} = y_{MCvertex} = 0$. These events were then reconstructed, feeding the *VertexerSPDz* with a displaced vertex position in the transverse plane: $x_{vertex} = \Delta x$ and $y_{vertex} = 0$. The reconstruction was repeated several times with Δx varying from 0 up to $\Delta x = 0.3$ cm. The results of this simulation are shown in Fig. 14, where both the bias on the measured z coordinate of the vertex and the resolution are plotted as a function of Δx. The vertexer gives an unbiased estimate with good resolution if $\Delta x \leq 200\,\mu$m, however the resolution degrades exponentially with increasing Δx and the measurement shows a not negligible bias already at $\Delta x \simeq 500\,\mu$m.

The knowledge of the average beam position in the transverse plane will be provided for real data by the quasi-online reconstruction of the interaction vertex with the *VertexerSPD3D*, which will allow to store in the OCDB the beam position in the transverse plane and use it during the offline event reconstruction, as described in Section 4.

4 Quasi-online reconstruction of the interaction diamond

The SPD based vertexing algorithms provide also a tool to monitor the beam position in the transverse plane and the interaction diamond profile quasi-online. This is achieved by running the local reconstruction (i.e. the cluster finding) on the SPD detectors and successively the vertex finding algorithm quasi-online on a sub-sample of events picked-up by the normal data flow and processed on dedicated machines. This possibility is provided by the ALICE DAQ framework within the context of the so-called detector algorithms (DA), which are calibration tasks running quasi-online (either during the run or immediately after the end of the run) to extract the calibration parameters that are needed in the reconstruction phase. These calibration quantities provided by the DAs are stored in the OCDB and are then accessed during the reconstruction. In particular, in the case of the SPD-Vertex-Diamond DA, the mean and r.m.s. of the reconstructed vertex

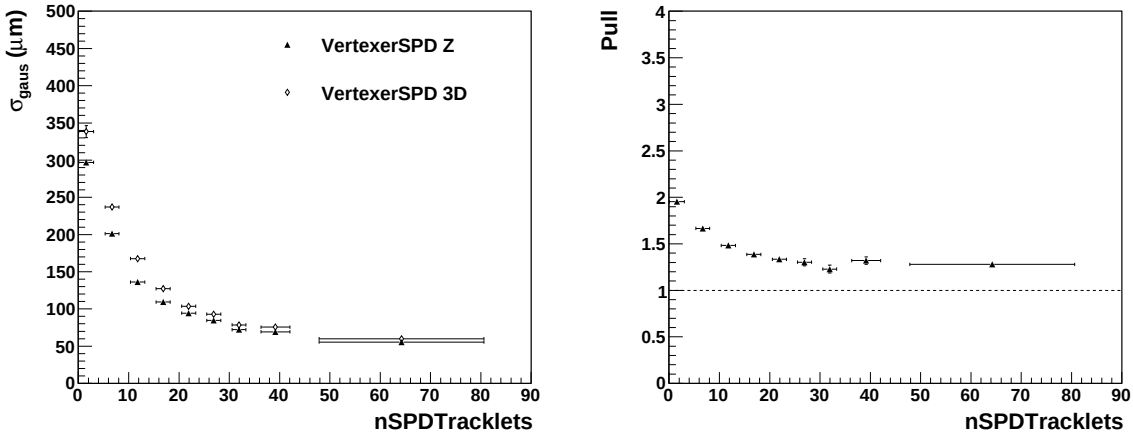

Figure 13: Resolution in z coordinate (left) and pull (right) as a function of number of SPD tracklets for events with $|z_{\mathrm{MCvertex}}| < 10$ cm. In the left plot, the resolution of *VertexerSPD3D* on the z coordinate is superimposed for comparison.

coordinates (x, y and z) of the sub-sample of analyzed events are stored in the OCDB and they are then used during the reconstruction phase to account for possible beam shifts in the transverse plane. The knowledge of the transverse position of the beam is needed in case of reconstruction with the *VertexerSPDz* and it is also important to centre properly the fiducial region in the first step of the *VertexerSPD3D*. Moreover, the online determination of the beam spread in the transverse plane can also be used as a tool to monitor the LHC luminosity. It should be pointed out that, to obtain the beam profile from the reconstructed vertex r.m.s., a deconvolution procedure is needed to correct for the enlargement of the distribution due to the resolution on the vertex coordinates. Since the resolution on vertex position depends on the event multiplicity (see e.g. Fig. 5), high multiplicity events are best suited for the determination of the beam diamond profile. These events can be selected during the DA execution by applying a cut on the number of contributors of the reconstructed vertex or on correlated quantities such as the number of reconstructed points in the SPD detector or the number of fired pixels, which have the advantage of being available at an earlier stage of the DA execution and therefore allow to spare computing time.

An example of distribution of reconstructed vertex coordinates is shown in Fig. 15, for two values of the applied cut on the minimum number of contributors for accepting a vertex. They are extracted from a sample of 30'000 PYTHIA pp events generated at $\sqrt{s} = 14$ TeV with a beam profile centred in $\langle x \rangle = 500$ μm and $\langle y \rangle = -200$ μm and a beam spread in the transverse plane of 50 μm (see Table 2). The two chosen cuts on the number of contributors are respectively 10 and 30. The looser selection (> 10 contributors) is aimed at excluding the low multiplicity events with much worse resolution while keeping a large fraction ($\approx 30\%$) of pp events. The tighter cut (> 30 contributors) selects the 10% of events with better resolution. The distributions of the generated y coordinates for the same sub-samples of events are superimposed as dashed histograms.

It can be seen that the tighter the selection on the number of contributors, the smaller the r.m.s. of the reconstructed vertex distribution, due to the better resolution on vertex position with increasing multiplicity. Hence, as expected, in order to achieve better precision in the extraction of the beam spread via a deconvolution procedure, a tight

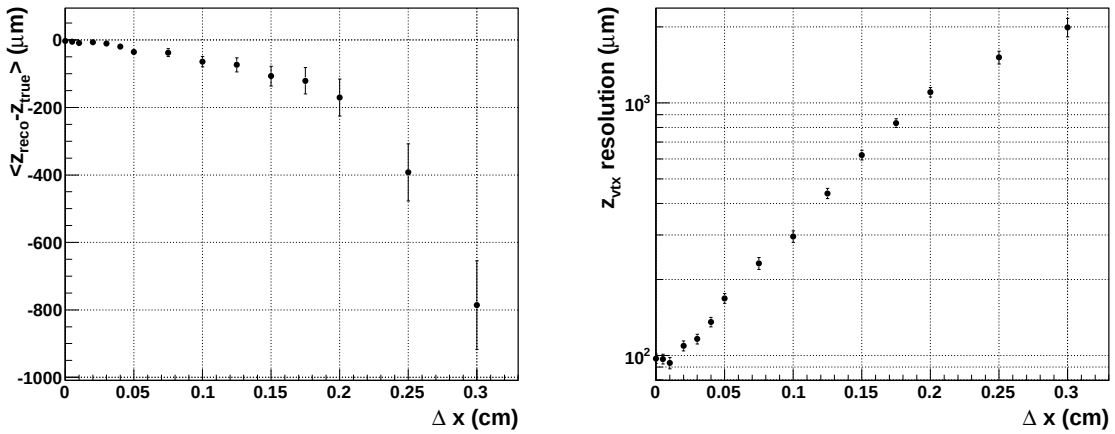

Figure 14: Average of residuals from *VertexerSPDz* (left) and resolution (right) as a function of Δx, which is the distance between the true beam position in the transverse plane and the position used by the vertexer

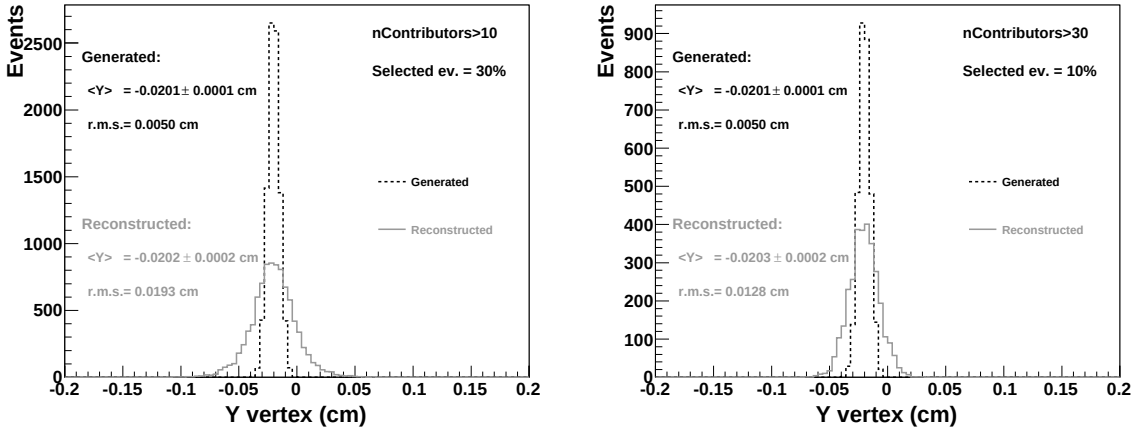

Figure 15: Distributions of reconstructed (solid) and generated (dashed) vertex coordinates for two values of selection on number of contributors.

selection on event multiplicity is needed. The fraction of selected events and the r.m.s. of the vertex distribution along y are shown in Fig. 16 as a function of the applied cut on the number of contributors. Having in mind that the beam spread used in this simulation study is 50 μm, it can be seen that also for a selection as tight as 100 contributors (i.e. about 1 event every 2000) the reconstructed vertex distribution is still dominated by the resolution on the vertex position which has therefore to be known with good precision in order to obtain a good estimation of the beam spread. Different deconvolution techniques are presently under study. Also the possibility of using the beam spread information from the LHC beam monitoring is taken in consideration. It should be finally noted that the selection of high multiplicity events increases, in the sample used for beam profile determination, the contribution from pile-up events, in which more than one pp collision occurs in the SPD strobe. Given the small beam spread in the transverse plane, pile-up events are expected to affect marginally the determination of the transverse beam profile. In any case, they can be rejected by means of the pile-up flag provided by the

VertexerSPD3D, as described in Section 1.

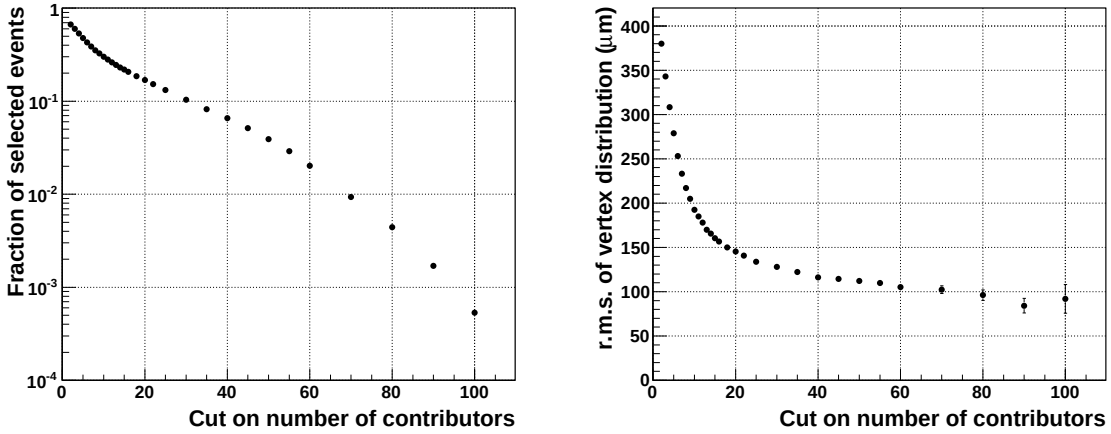

Figure 16: Fraction of selected events (left) and r.m.s. of the vertex distribution along y as a function of the cut value on the number of contributors.

5 Reconstruction of the vertex position using tracks

The reconstruction of the primary vertex is affected by the low multiplicity created in pp collisions. The resolution can be improved using tracks instead of tracklets. In the following, a general description of the procedure is presented and both the algorithms for vertex finding and fitting are described in detail.

5.1 Notation

After track reconstruction in the central barrel, for each track, which is locally parameterized as an helix, all the information is summarized in the state vector of the 5 helix parameters and in their 5×5 covariance matrix.

The track parameterization is defined in the local reference frame obtained from the global one with a rotation about the z axis:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R}(\alpha) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{10}$$

where α is the azimuthal angle w.r.t. the x axis and identifies the TPC sector in which the track lies at the inner radius of this detector. The angle α can take one of the 18 values $\alpha_i = 10^\circ + i \cdot 20^\circ$ ($i = 0, 1, \ldots, 17$). Thus, x' is essentially a radial coordinate. The track parameters are given at a reference plane at $x' = x'_{\rm ref}$.

The five track parameters are:

- y': local y coordinate of the track at the reference plane $x' = x'_{\rm ref}$;
- z': local z coordinate of the track at the reference plane $x' = x'_{\rm ref}$;
- $\sin\phi$: ϕ is the azimuthal angle formed by the track momentum with the x' axis;
- $\tan\lambda$: tangent of the track angle with the bending plane ($\tan\lambda = p_z/p_{\rm t}$);
- k: track curvature. $k = 1/R$, where R is the radius of the circle obtained projecting the track on the bending plane.

If the track is propagated to the reference position $x'_{\rm ref}$, the current position $\mathbf{r} = (x, y, z)^T$ in the global frame is obtained as $\mathbf{r} = \mathbf{R}^{-1}(\alpha)\,\mathbf{r'}$, where $\mathbf{r'} = (x'_{\rm ref}, y', z')^T$ (note that α does not change with $x'_{\rm ref}$). Vice-versa, the two track position parameters are obtained from the current global position as:

$$\begin{pmatrix} y' \\ z' \end{pmatrix} = \mathbf{Q}(\alpha) \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{11}$$

where $\mathbf{Q}(\alpha)$ is a 2×3 matrix formed by the second and third row of the matrix $\mathbf{R}(\alpha)$.

We will use the notation $\mathbf{F}$ for the complete track covariance matrix and $\mathbf{U}$ for the covariance matrix of (y', z'). The vertex coordinates in the global frame will be indicated with $\mathbf{r}_v$.

5.2 Outline of the method

The scheme used for vertex reconstruction is reported in Fig 17: the procedure is performed in three passes. The first consists in a preselection of the tracks and in a rough preliminary vertex estimate ($vertex_0$) used as a starting point for the reconstruction. Tracks with insufficient number of associated clusters or not pointing to a fiducial cylinder with $r < 3$ cm (beam pipe radius) and $|z| < 30$ cm, are rejected at preselection stage.

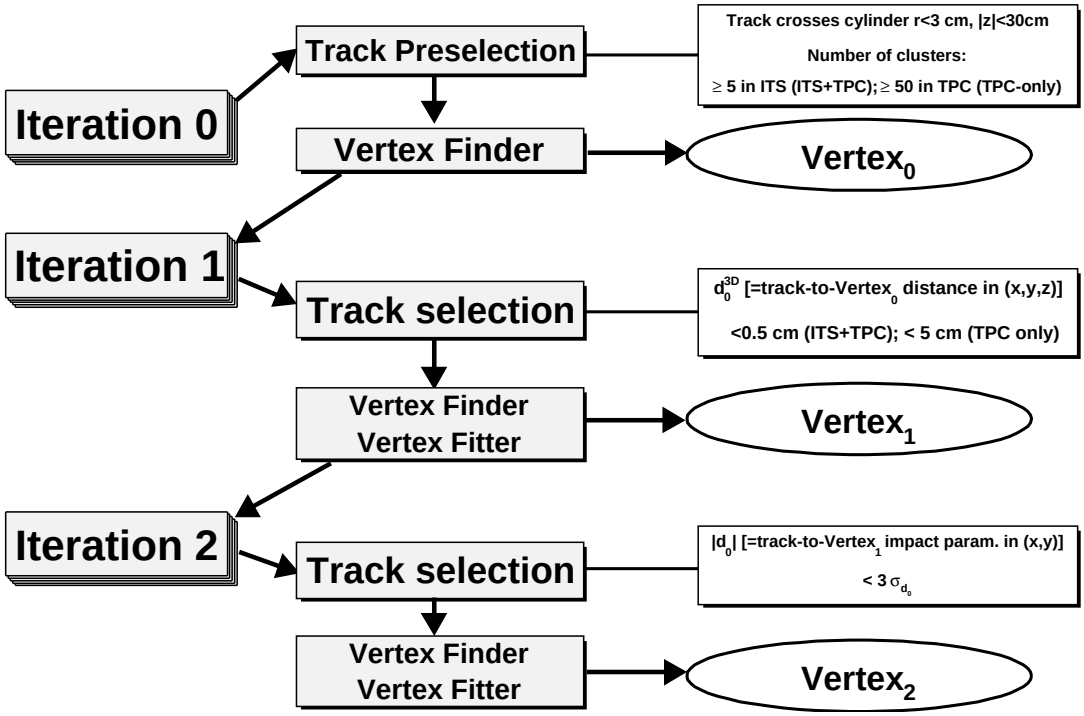

Figure 17: Scheme adopted for vertex reconstruction with tracks reconstructed in both TPC and ITS (ITS+TPC), and only in TPC (TPC-only).

Both subsequent iteration passes consist of three steps:

1. TRACK SELECTION: the goal of this step is to reject displaced tracks.
2. VERTEX FINDING: a first estimate of the vertex position is obtained. This method does not require an accurate *a priori* vertex information. A specific vertex finding algorithm, described in Section 5.4, is also used at the track preselection stage.

3. VERTEX FITTING: tracks are propagated to the position estimated in the previous step and the optimal estimate of the vertex position, as well as the vertex covariance matrix and χ^2 are obtained via a fast fitting algorithm. The interaction diamond information can be used as a constraint in the fit.

5.3 Track selection

The selection criteria are looser in the first iteration pass, when the vertex position is only approximately known (in order to avoid possible biases, we do not use the information on the SPD-reconstructed vertex at this stage), and tighter in the second pass, when the estimated vertex position from the first pass can be used. The loose criteria (first iteration), meant to reject tracks with large displacements (far secondaries from strangeness decays and possible fake tracks), are cuts in absolute distance from the current estimate of the vertex. These cuts are different for TPC-only and for ITS+TPC. The tight criteria (second iteration), meant to reject tracks with small displacement (close secondaries and particles that undergo large scatterings in the material), are cuts in normalized distance, i.e. number of standard deviations, from the current vertex estimate. In this case the same cuts are used for TPC-only and ITS+TPC tracks. The values of the cuts are reported in Fig 17.

5.4 Vertex finding algorithms

Two vertex finding algorithms are implemented, with slightly different features in terms of robustness and precision, in order to comply with the requirements at the various stages of the vertex reconstruction procedure. The first algorithm, more robust against outliers, is used at the preselection stage in order to start the whole vertex reconstruction procedure (Iteration 0 in Fig. 17), while the second algorithm, more precise, is used in Iterations 1 and 2.

In the first algorithm, the reconstructed tracks are propagated to the reference plane $x'_{\rm ref} = x_{\rm nom}\cos\alpha + y_{\rm nom}\sin\alpha$, where $(x_{\rm nom}, y_{\rm nom})$ are the coordinates of the beam position in the tranverse plane (see Section 4). The algorithm is based on the straight line approximation of the tracks, i.e the tangent line at $x'_{\rm ref}$ to the helix representing the reconstructed trajectory of the particle is used for the calculation. The expected spread of the vertex position in the transverse plane with respect to $(x_{\rm nom}, y_{\rm nom})$ is of the order of 50 μm (See Table 2) for pp at 14 TeV, hence the adoption of the straight line approximation computed at the beam average positions is appropriate, given the relatively low magnetic field adopted in the ALICE barrel. As a matter of fact, the method turned out to be poorly sensitive to the values used for the coordinates of the beam position, giving good results when they were arbitrarily chosen in a range of few millimetres with respect to the true position. All the possible track pairs ij are considered and for each pair the coordinates of the points $\mathbf{p}_i \equiv (x_i, y_i, z_i)$ and $\mathbf{p}_j \equiv (x_j, y_j, z_j)$ on the two lines defining the segment of closest approach are used to determine the *intersection point* of the two lines as:

$$\mathbf{c}_{ij} = \begin{pmatrix} x_{ij} = \frac{\sigma^2_{x_j}}{\sigma^2_{x_i}+\sigma^2_{x_j}} x_i + \frac{\sigma^2_{x_i}}{\sigma^2_{x_i}+\sigma^2_{x_j}} x_j \\ y_{ij} = \frac{\sigma^2_{y_j}}{\sigma^2_{y_i}+\sigma^2_{y_j}} y_i + \frac{\sigma^2_{y_i}}{\sigma^2_{y_i}+\sigma^2_{y_j}} y_j \\ z_{ij} = \frac{\sigma^2_{z_j}}{\sigma^2_{z_i}+\sigma^2_{z_j}} z_i + \frac{\sigma^2_{z_i}}{\sigma^2_{z_i}+\sigma^2_{z_j}} z_j \end{pmatrix} \tag{12}$$

where the weights are based on the errors of the $\mathbf{p}_i$ and $\mathbf{p}_j$ coordinates taken from the reconstructed tracks covariance matrices and propagated to the master reference system. As a result, the *intersection point* is closer to the best reconstructed track. If the tracks

were reconstructed with the same errors, the point would simply be the centre of the closest approach segment between the two lines. The coordinates of the primary vertex are then determined as:

$$Vertex_0 = \begin{pmatrix} \frac{1}{N_{\rm pairs}} \sum_{i,j} x_{ij} \\ \frac{1}{N_{\rm pairs}} \sum_{i,j} y_{ij} \\ \frac{1}{N_{\rm pairs}} \sum_{i,j} z_{ij} \end{pmatrix} \tag{13}$$

where $N_{\rm pairs}$ is the number of track pairs. The contamination from secondary particles can be easily reduced with this algorithm: pairs which have a distance of closest approach exceeding a fiducial cut (1 mm) are not used in Eq. (13). This method does not provide an estimate for the errors on the vertex, since it is intended only to seed a more refined selection of the tracks in the subsequent reconstruction pass.

Also the second algorithm is based on the straight line approximation of the tracks. In the first iteration the tangent to the track at the nominal beam position is used, while in the second iteration the track is prolonged to the vertex found at the end of the first iteration. The coordinates of the primary vertex are given by finding the point of minimum distance among the tracks. This is done by minimizing analytically the quantity defined in Eq. (4), but in this case the errors on the track parameters (σ_{xi}, σ_{yi} and σ_{zi}) are obtained from the 5×5 covariance matrix of the track. The advantage of the straight line approximation in this case is that the minimization procedure is fast because it simply consists in a set of linear equations. This second method for vertex finding allows also a determination of the errors on the vertex position, but it relies on an appropriate selection of the tracks used for the computation, so it cannot be used at preselection stage.

5.5 Vertex fitting

The vertex finding algorithm provides, as described, a first estimate of vertex position and propagates track parameters to this position. The task of the vertex fitting algorithm is to determine the best fit coordinates of the vertex and the vertex covariance matrix. We have implemented this step on the basis of the fast vertex fitting method described in Ref. [11].

Since the measurements of different tracks are independent of each other, the χ^2 function to be minimized can be written as a sum over tracks. In Ref. [11] it is shown that, if the tracks can be approximated to straight lines ($\phi = constant$, $\lambda = constant$ and $k = 0$) in the vicinity of the vertex position, the χ^2 is:

$$\chi^2(\mathbf{r}_v) = \sum_i (\mathbf{r}_v - \mathbf{r}_i)^T \, \mathbf{V}_i^{-1} (\mathbf{r}_v - \mathbf{r}_i). \tag{14}$$

In this expression, $\mathbf{r}_i$ is the current global position of the track i (i.e. the position given by the vertex finder) and $\mathbf{V}_i$ is the covariance matrix of the vector $\mathbf{r}_i$:

$$\begin{aligned} \mathbf{V} &= \mathbf{Q}(\alpha)^T \, \mathbf{U} \, \mathbf{Q}(\alpha) \\ &= \begin{pmatrix} -\sin\alpha & 0 \\ \cos\alpha & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \langle y', y' \rangle & \langle y', z' \rangle \\ \langle y', z' \rangle & \langle z', z' \rangle \end{pmatrix} \begin{pmatrix} -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \sin^2\alpha \, \langle y', y' \rangle & -\sin\alpha \cos\alpha \, \langle y', y' \rangle & -\sin\alpha \, \langle y', z' \rangle \\ -\sin\alpha \cos\alpha \, \langle y', y' \rangle & \cos^2\alpha \, \langle y', y' \rangle & \cos\alpha \, \langle y', z' \rangle \\ -\sin\alpha \, \langle y', z' \rangle & \cos\alpha \, \langle y', z' \rangle & \langle z', z' \rangle \end{pmatrix} \end{aligned} \tag{15}$$

The approximation of the track to a straight line allows to neglect, in the covariance matrix of $\mathbf{r}$, the contribution of the elements of the track covariance matrix relative to the curvature and direction parameters.

We will now verify that this approximation holds in our particular case. Since the tracks are propagated by the vertex finder to the first estimate of the vertex position, which is determined with a resolution $\sigma \sim 100$ μm in the bending plane, the length over which we neglect the curvature and the changes in the direction parameters is of the same order of magnitude; however we consider a safety factor of 10 and we estimate the effects of the linear approximation over a length $\ell \sim 1$ mm. The sagitta of the arc with this length and with radius of curvature $R = 1$ m is $\ell^2/8R \simeq 0.125$ μm. In a magnetic field of 0.2 T (the minimum value that will be used in ALICE) a radius of curvature of 1 m corresponds to $p_{\rm t} = 60$ MeV/c, which is lower than the minimum transverse momentum of the reconstructed tracks.

Given that the matrix $\mathbf{V}$ is independent of $\mathbf{r}_v$, the expression (14) is a linear function of $\mathbf{r}_v$. The solution for the vertex coordinates which minimize (14) reads then:

$$\mathbf{r}_v = \left(\sum_i \mathbf{W}_i \right)^{-1} \sum_i \mathbf{W}_i \, \mathbf{r}_i \tag{16}$$

with $\mathbf{W}_i = \mathbf{V}_i^{-1}$ and the covariance matrix of $\mathbf{r}_v$ is

$$\mathbf{C}_v = \left(\sum_i \mathbf{W}_i \right)^{-1} \quad . \tag{17}$$

It is straight-forward to include the diamond information as a vertex constraint. If the diamond is defined by the (x, y, z) position of its centre, $\mathbf{r}_d$, and by the covariance matrix $\mathbf{C}_d$, which describes the spread of the interaction region, the vertex coordinates and covariance matrix become (with $\mathbf{W}_d = \mathbf{C}_d^{-1}$):

$$\begin{aligned} \mathbf{r}_v &= (\mathbf{W}_d + \textstyle\sum_i \mathbf{W}_i)^{-1} \, (\mathbf{W}_d \, \mathbf{r}_d + \textstyle\sum_i \mathbf{W}_i \, \mathbf{r}_i) \ , \\ \mathbf{C}_v &= (\mathbf{W}_d + \textstyle\sum_i \mathbf{W}_i)^{-1} \ . \end{aligned} \tag{18}$$

We remind that, as discussed in Section 1, the diamond information to be used with ITS+TPC (TPC-only) tracks has to be extracted using the distribution of vertices reconstructed from ITS+TPC (TPC-only) tracks.

5.6 Results

5.6.1 Results for tracks reconstructed in the TPC and in the ITS

We tested the algorithm on two samples of 10^4 proton–proton minimum bias events produced with the PYTHIA event generator [9], at $\sqrt{s} = 0.9$ and 14 TeV, with the interaction diamond centred at $(0, 0, 0)$ and smeared according to the values reported in Table 2.

The minimum number of tracks required in the vertex fit is 2 when the diamond constraint is not used, and 1 when the diamond constraint is used. The vertex reconstruction efficiency, integrated over the event multiplicity, is reported in Table 5. We considered three possible normalizations: a) all generated events (PYTHIA minimum bias); b) events that pass the loosest ALICE minimum bias trigger selection (MB1: SPD FastOR || V0C || V0A, i.e. signal in the SPD, $-1.96 < \eta < 1.96$, or in C-side V0 detector, $-3.7 < \eta < -1.7$,

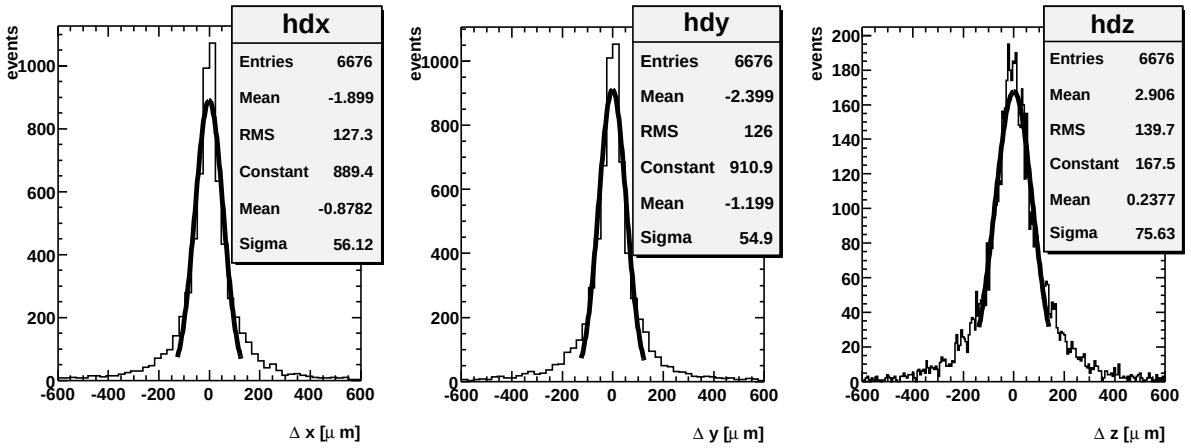

Figure 18: Distributions of the residuals for the vertices reconstructed with ITS+TPC tracks, in pp at 14 TeV without using the diamond constraint: $\Delta q = q_{\rm measured} - q_{\rm true}$, for $q = x$, y, z.

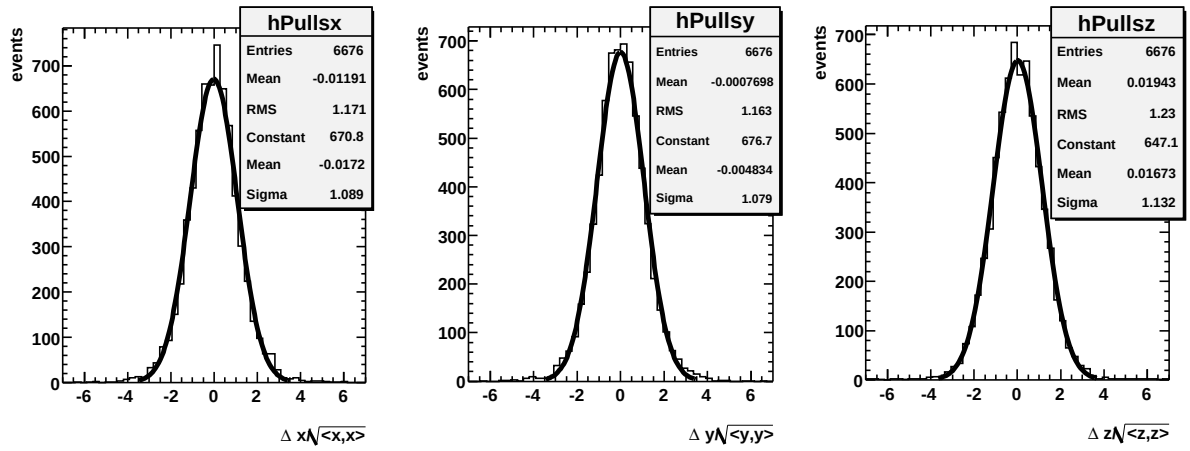

Figure 19: Same as Fig. 18 for the standardized residuals.

or in the A-side V0 detector, $2.8 < \eta < 5.1$); c) triggered events, as defined in b), in which at least one track is reconstructed in ITS+TPC with 5 or 6 clusters associated in the ITS. The efficiency is larger when the diamond constraint is used, and it is close to perfect (98.1% at 14 TeV and 96.4% at 0.9 TeV) when at least one "usable" track is present.

Table 5: Vertex reconstruction efficiency using ITS+TPC tracks.

$\sqrt{s}$	diamond constraint	PYTHIA min. bias	triggered (MB1)	triggered (MB1) & nTrks(nCls$_{\rm ITS} \geq 5$)≥ 1
14 TeV	w/o constraint	69.0%	76.8%	85.2%
14 TeV	with constraint	79.5%	88.5%	98.1%
0.9 TeV	w/o constraint	55.8%	62.5%	75.9%
0.9 TeV	with constraint	70.9%	79.5%	96.4%

The efficiency for vertex reconstruction as a function of event multiplicity will be

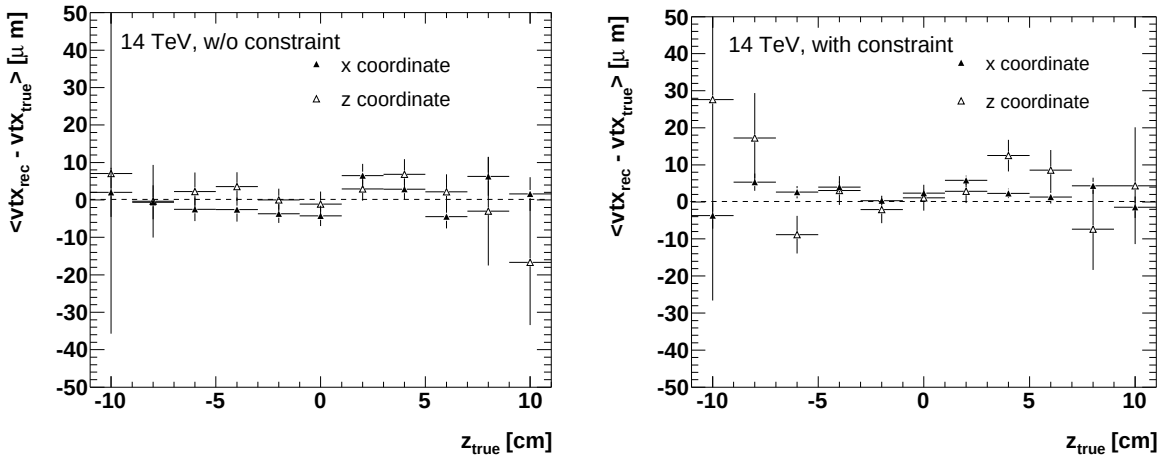

Figure 20: Mean values, as obtained from a Gaussian fit, of the x and z residuals for vertices reconstructed with ITS+TPC tracks, as a function of the generated vertex z, for pp at 14 TeV. Diamond constraint not used for the left-hand figure and used for the right-hand figure.

detailed in the following.

Figure 18 presents the residuals Δq for the three vertex coordinates ($q = x,\ y,\ z$) for pp at 14 TeV without diamond constraint, integrated over all the statistics. The distributions are clearly non-Gaussian, since they are the convolution of many Gaussian distributions with different dispersions, depending on the number of tracks used in each event for the fit. However, we fitted with a Gaussian the central part of the distributions, in order to quantify the global resolution. We obtain $\sigma_x \simeq \sigma_y \simeq 55$ μm and $\sigma_z \simeq 75$ μm. We checked the reliability of the estimated errors on the vertex position with the test of the pulls. The distribution of the standardized residuals, $\Delta q/\sqrt{\langle q, q\rangle}$ for $q = x,\ y,\ z$, are normal (Fig. 19), indicating that the errors given in the vertex covariance matrix describe the resolution on the vertex estimate. The resolutions and pulls as a function of event multiplicity will be presented in the following.

Figure 20 shows the mean value, as obtained from a Gaussian fit, of the distribution of the residuals for the x and z coordinates, as a function of the generated z position of the vertex. The mean value is compatible with zero, indicating that the reconstructed vertex position is unbiased both when the diamond constraint is not used and when it is used.

As for the reconstruction of the vertex position with the SPD, the performance of the algorithm was evaluated as a function of the event multiplicity, using the number of reconstructed tracklets in the SPD as an estimator of the multiplicity. The efficiency for vertex reconstruction for triggered events (defined as the ratio of the number of events with vertex to the total number of triggered events, in each bin) is reported in Fig. 21. At low multiplicity, the use of the diamond constraint allows to increase the vertex reconstruction efficiency.

In Figs. 22 and 23 we present the resolutions and the pulls as a function of the event multiplicity (number of tracklets in SPD), without and with diamond constraint, for pp events at 14 TeV. At lower values of $\sqrt{s}$, the resolutions are approximately the same as at 14 TeV, for the same multiplicity. The resolutions, without diamond constraint, have

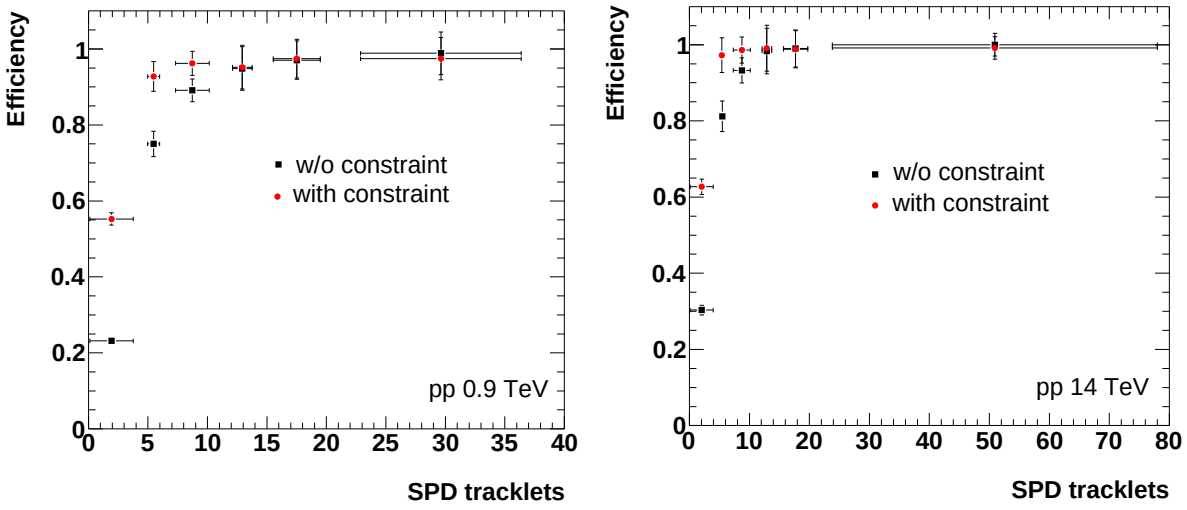

Figure 21: Vertex reconstruction efficiency with ITS+TPC tracks for triggered pp events at 0.9 (left) and 14 TeV (right), without and with the diamond constraint.

been fitted to the expression:

$$\sigma(nSPDtracklets) = a + \frac{b}{\sqrt{nSPDtracklets}}. \tag{19}$$

The results of the fit are reported in Table 6. When the diamond constraint is used, the resolutions in x and y improve drastically for low multiplicities and, as expected, they are always smaller than the diamond spread (50 μm for $\sqrt{s} = 14$ TeV). The resolution for the z coordinate is unaffected by the constraint (in the first multiplicity bin, the resolution is slightly worse with the constraint because a larger number of events with very few tracks enter now the sample of events with reconstructed vertex, see Fig. 21). The pulls are within 1.00 ± 0.15 for all event multiplicities, both when the diamond constraint is not used and when it is used. Therefore, the vertex position uncertainties are properly estimated.

Table 6: Results of the fit of the multiplicity dependence of the resolutions. The fit function is $a + b/\sqrt{nSPDtracklets}$.

Parameter	x	z
a [μm]	-20 ± 1	-13 ± 2
b [μm]	377 ± 9	466 ± 11

The vertex reconstruction resolution with ITS+TPC tracks is determined by the single-track precision in space, which is affected by ITS misalignments (difference between real and ideal position in space of the detector sensors). Therefore, we have investigated the effect of ITS spatial misalignment on the vertex reconstruction in pp collisions at $\sqrt{s} = 14$ TeV. The exercise was done simulating pp events with a misaligned ITS geometry (random Gaussian shifts of the positions of the sensors) and reconstructing them with the ideal, perfectly aligned, geometry. We have considered three (mis)alignment scenarios: "null" misalignment, i.e the ideal geometry; "residual" misalignment, which represents the expected misalignment after the alignment procedures have been completed and was

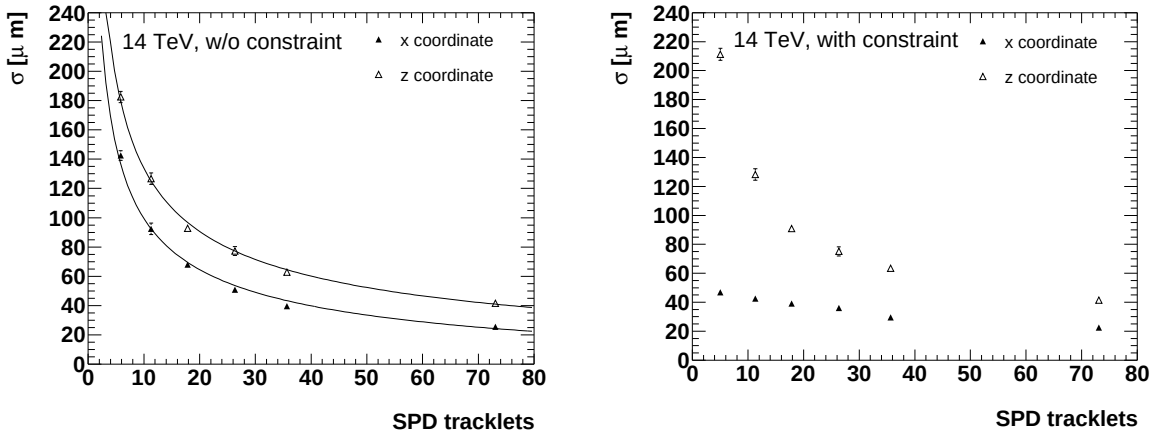

Figure 22: Resolutions on vertex position reconstructed with ITS+TPC tracks in x and z as a function of event multiplicity, without and with diamond constraint. The trends have been fitted to the expression $a + b/\sqrt{nSPDtracklets}$ and the fitted parameters are reported in Table 6.

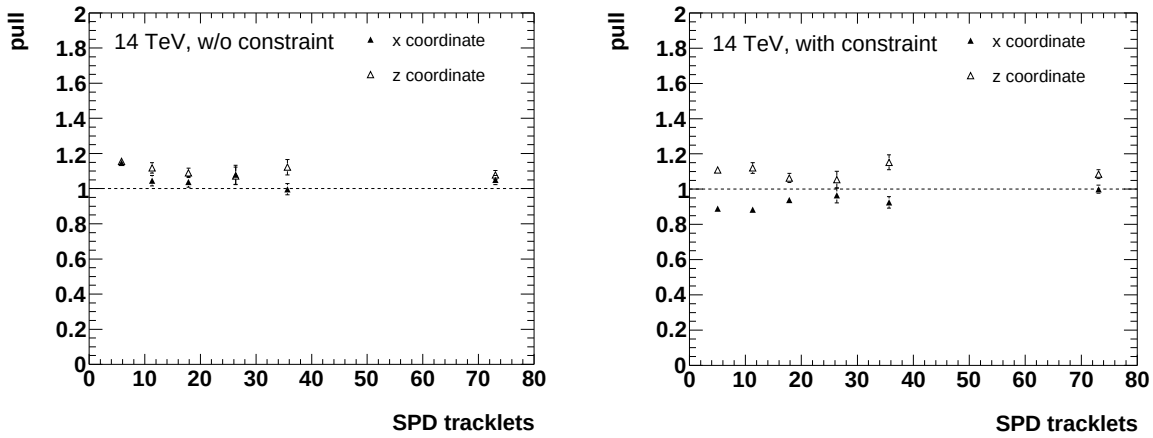

Figure 23: Pulls for the x and z vertex coordinates reconstructed with ITS+TPC tracks, as a function of event multiplicity, without and with diamond constraint.

set to a random misalignment with r.m.s. of ≈ 10 μm (to be compared to the SPD intrinsic spatial resolution of ≈ 11 μm in the most precise coordinate, $r\phi$); "full" misalignment, a larger random misalignment with r.m.s. of ≈ 20 μm. The results, shown in Fig. 24, indicate that misalignments at this level have a negligible effect on the primary vertex reconstruction performance.

5.6.2 Results for tracks reconstructed only in the TPC

The vertex reconstruction efficiency using TPC-only tracks, integrated over the event multiplicity, is reported in Table 7. Again, we considered three possible normalizations: a) PYTHIA minimum bias; b) ALICE minimum bias trigger selection (MB1); c) triggered events, as defined in b), in which at least one track reconstructed in the TPC crosses the beam pipe ($r = 3$ cm). The efficiency is larger when the diamond constraint is used, and it is above 98% in events with at least one TPC track pointing to the beam

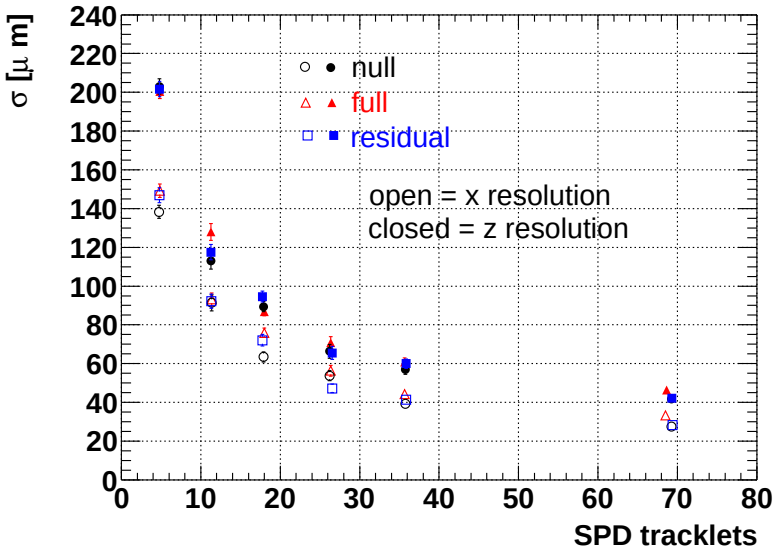

Figure 24: Effect of ITS misalignment on the primary vertex position resolution in pp collisions at $\sqrt{s} = 14$ TeV. .

pipe.

Table 7: Vertex reconstruction efficiency using TPC-only tracks.

$\sqrt{s}$	diamond constraint	PYTHIA min. bias	triggered (MB1)	triggered (MB1) & $\mathrm{nTrks_{TPC}}(r < 3\ \mathrm{cm}) \geq 1$
14 TeV	w/o constraint	74.3%	82.7%	91.0%
14 TeV	with constraint	81.9%	91.1%	98.3%
0.9 TeV	w/o constraint	61.9%	69.3%	82.7%
0.9 TeV	with constraint	75.6%	84.7%	98.1%

The efficiency for vertex reconstruction with TPC-only tracks in triggered pp events is reported in Fig. 25 as a function of event multiplicity (number of tracklets in the SPD). Also in this case, at low multiplicity, the use of the diamond constraint allows to increase the vertex reconstruction efficiency. Note that, when the constraint is not used, the efficiency for the case of ITS+TPC tracks is a bit lower (Fig. 21) than for the case of TPC-only tracks (Fig. 25). This is due to the fact that ITS+TPC tracks are more precise and, therefore, the algorithm is more effective in rejecting secondary tracks. As a result, in low multiplicity events, there is a higher probability that an insufficient number of tracks (less than two) are selected.

In Figs. 26 and 27 we present the resolutions and the pulls as a function of the event multiplicity (number of tracklets in SPD), without and with diamond constraint. When the constraint is used, since the size of diamond ($\sim 100\ \mu$m) is much smaller than the resolution of TPC tracks ($\sim$ few mm), the centre of the diamond is used to define the (x, y) position of the vertex, while the z coordinate is determined by the TPC tracks. The pulls without constraint tend to be larger than unity, especially for z. This is probably due the residual contamination of secondary TPC tracks in the sample used to reconstruct

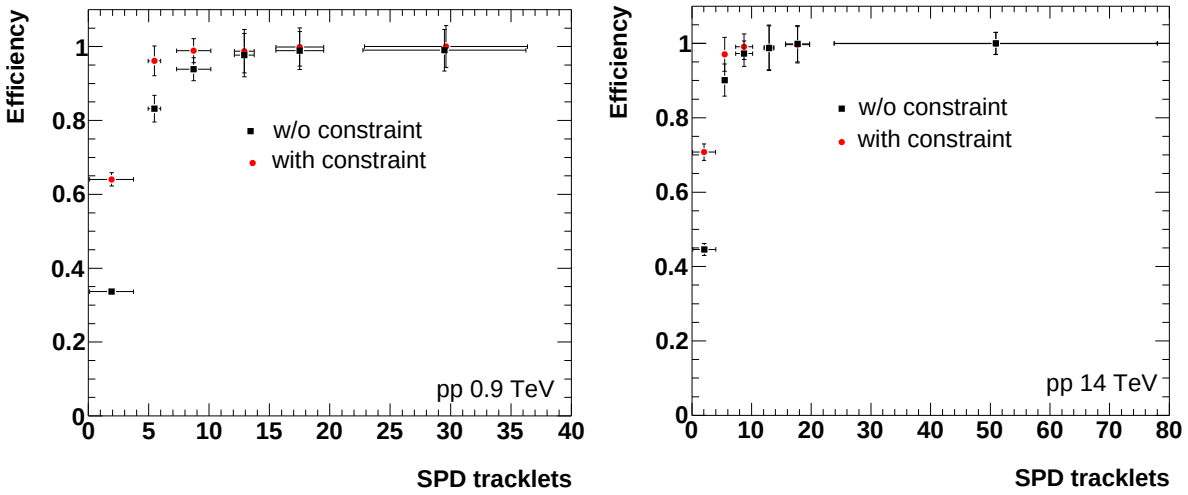

Figure 25: Vertex reconstruction efficiency with TPC-only tracks for triggered pp events at 0.9 (left) and 14 TeV (right), without and with the diamond constraint.

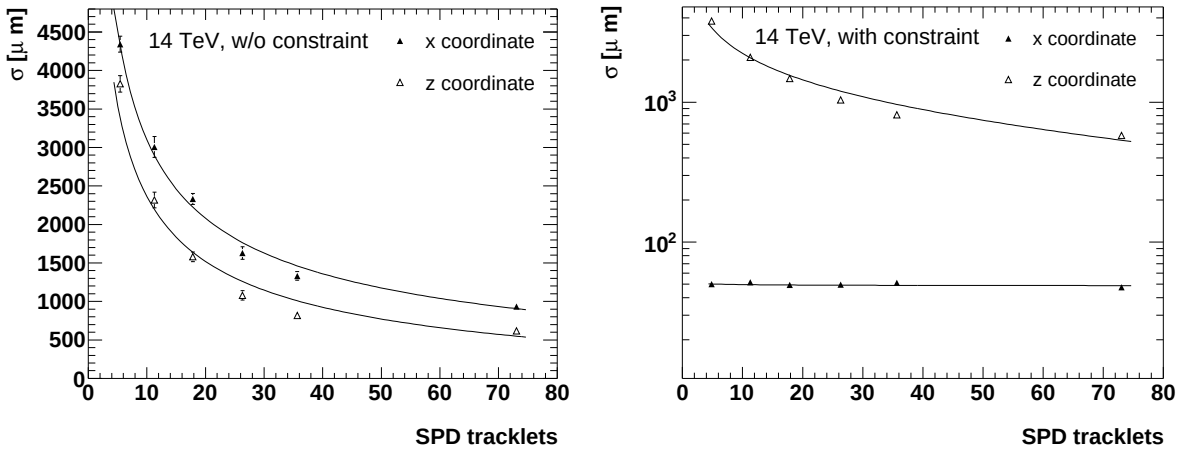

Figure 26: Resolutions on vertex position, reconstructed with TPC-only tracks, in x and z as a function of event multiplicity, without (left) and with (right) diamond constraint. To guide the eye, the trends have been fitted to the expression $a + b/\sqrt{nSPDtracklets}$.

the primary vertex.

5.7 Fast re-evaluation of the vertex after removal or addition of tracks

The possibility to evaluate quickly the effect of removing a few tracks from the set of tracks used to reconstruct the primary vertex is very useful during the physics analysis. In particular, this is a common practice in the analyses for the measurement of heavy-flavour (charm and beauty) production. Due to the short mean decay length of D and B mesons ($c\tau \simeq$ 100–300 μm and $c\tau \simeq 500$ μm, respectively), the tracks produced by their decay particles are often not rejected by the primary vertex reconstruction algorithms and are, therefore, included in the set of tracks used to define the vertex. This causes a bias in the measurement of their impact parameter, which is underestimated. A way to reduce this bias is to calculate the impact parameter of a given track with respect to a primary vertex position obtained after removing that track from the set of tracks used to define

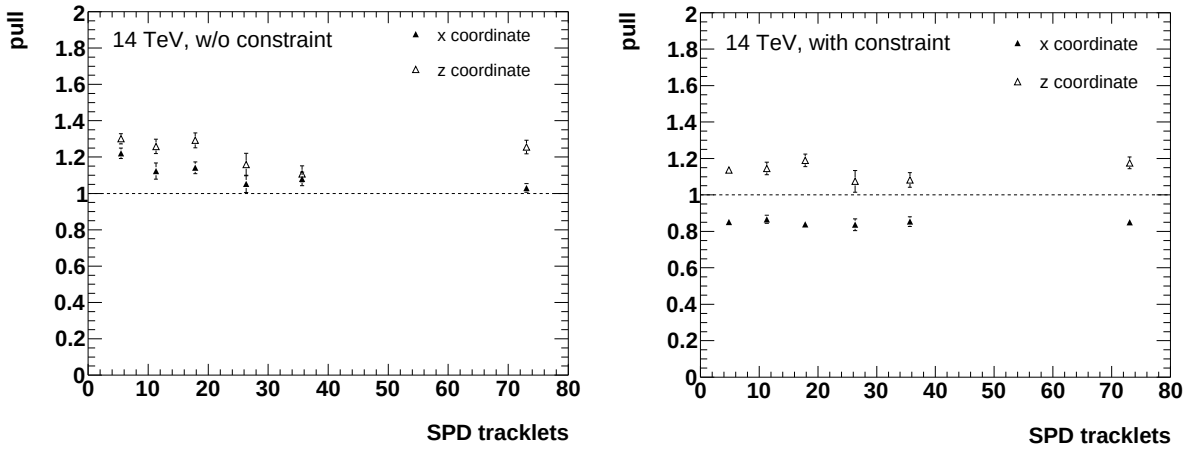

Figure 27: Pulls for the x and z vertex coordinates, reconstructed with TPC-only tracks, as a function of event multiplicity, without (left) and with (right) diamond constraint.

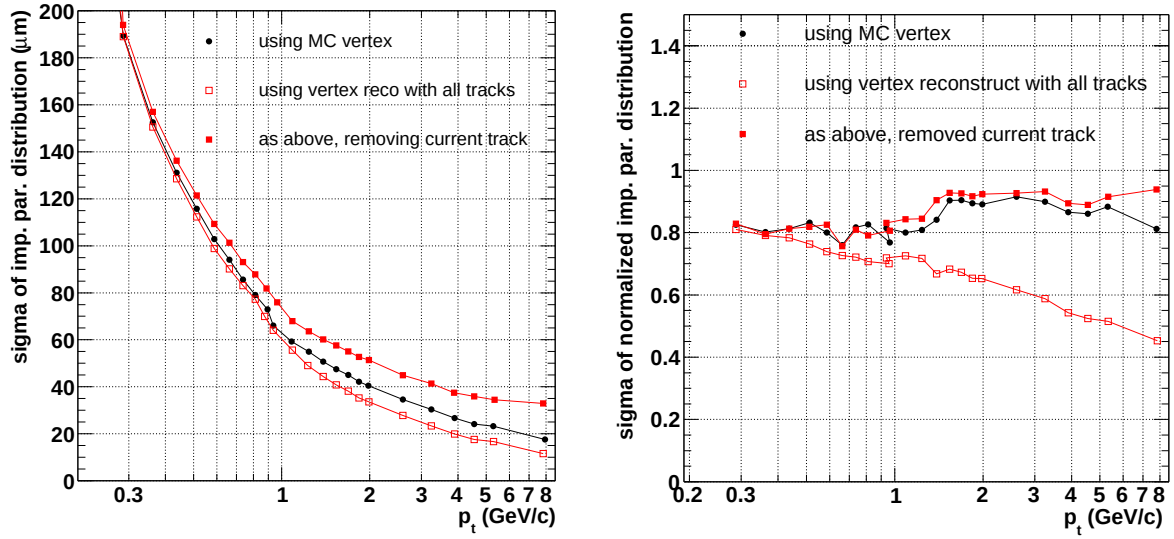

Figure 28: Sigma (Gaussian fit) of the distribution of the impact parameters (left) and normalized impact parameters (right) for reconstructed primary $\pi^{\pm}$ tracks, as a function of p_{t} [12], using the true Monte Carlo primary vertex, the vertex reconstructed using all tracks, or the vertex reconstructed using all tracks except the track under study.

the vertex. Figure 28, from Ref. [12], shows the sigma (Gaussian fit) of the distribution of the impact parameters (left) and normalized impact parameters (right) for reconstructed primary $\pi^{\pm}$ tracks, as a function of p_{t}. When using the vertex reconstructed from all tracks, including the track under study, the impact parameter is biased to smaller values (the distribution is narrower than in the case of using the true Monte Carlo vertex, and the pulls are much smaller than one). When the track under study is not used in the primary vertex fit, the distribution becomes broader than in the case of the true vertex, as it should, and pulls get closer to one. The procedure would require to re-evaluate the position of the vertex several times per event, i.e. for each of the heavy-flavour particle candidates (for example, $\mathrm{D}^0 \rightarrow \mathrm{K}^-\pi^+$ or $\mathrm{D}^+ \rightarrow \mathrm{K}^-\pi^+\pi^+$) that can be constructed from

the tracks of the event.

Since, in the vertex fitting algorithm, the vertex is defined as a sum of contributions from the single tracks (Eq. (16)), it is in principle possible to remove one or more tracks from the vertex by subtracting their contributions. However, the contribution of a given track depends on the position at which the track has been propagated. In the algorithm we described, this is the position of the vertex given by the vertex finder and, therefore, it depends on the set of tracks that have been used. In order to make the subtraction procedure mathematically *exact*, the contribution of a given track should be independent of the rest of the tracks. This could be achieved by recalculating the primary vertex position, once per event, at the analysis level, if the diamond position estimated from ITS+TPC tracks is available. At this stage, the tracks could be propagated to the position of the centre of the diamond, rather than to the position given by the vertex finder. Then, starting from this vertex ($\mathbf{r}_v$, $\mathbf{C}_v$), given by a set of tracks $1, \ldots, n$, the vertex ($\mathbf{r}'_v$, $\mathbf{C}'_v$) given by the same set of tracks except the track j can be obtained as:

$$\begin{aligned} \mathbf{r}'_v &= \left(\mathbf{C}_v^{-1} - \mathbf{W}_j\right)^{-1} \left(\mathbf{C}_v^{-1}\mathbf{r}_v - \mathbf{W}_j\mathbf{r}_j\right) \\ \mathbf{C}'_v &= \left(\mathbf{C}_v^{-1} - \mathbf{W}_j\right)^{-1} . \end{aligned} \tag{20}$$

This can be done, quickly, for all tracks (or pairs, or triplets of tracks) in the event. Similarly, a track can be added to the vertex, replacing the subtractions with additions in Eq. (20).

6 Reconstruction of secondary vertices of charmed particles

The algorithm for the reconstruction of secondary vertices originated by n-prong decays ($n \geq 2$) is the same as the one described in Section 5 to find the interaction vertex with tracks. Tracks are approximated as straight lines in the vicinity of the primary vertex. A sketch of the linear approximation is shown in Fig. 29 for the three-body decay $\mathrm{D}^+ \to \mathrm{K}^-\pi^+\pi^+$ meson. The straight line approximation of the track near the primary vertex may produce a significant worsening in the measurement of a displaced vertex. For this purpose, we consider the distance d between the secondary vertex and the straight line (reported in Fig. 29) as a geometrical measurement of the discrepancy between the helix and the tangent line. In Fig. 30 the distance d defined above is plotted as a function of the transverse momentum of the particle originating from the D^+ decay, for two different choices of the D^+ decay length: 300 μm, of the order of $c\tau$, and 3000 μm, less probable since it corresponds to $\approx 10\ c\tau$.

As expected, d decreases for increasing p_t, because high-momentum tracks are less bent; in addition, as it appears from the comparison of the two plots, d is larger when the secondary vertex is more displaced from the interaction point, confirming that the farther the origin of the particle from the primary vertex, the less precise is the approximation of the helix with the tangent line calculated in proximity of the interaction point. However, even in the worst case of a distance between primary and secondary vertices of the order of 3 mm, the discrepancy between the helix and the tangent line is less than 1 μm for $B = 0.5$ T and $p_\mathrm{t} \approx 0.5$ GeV/c, and thus much smaller than the track position resolution.

In Fig. 31 we report the secondary vertex position resolution and pulls as a function of the p_t of the D^+. At low p_t the daughter tracks are softer and therefore they suffer multiple scattering, thus the resolution degrades toward small p_t. As the p_t of the D^+ increases one would expect an improvement of the resolution since the decay tracks have

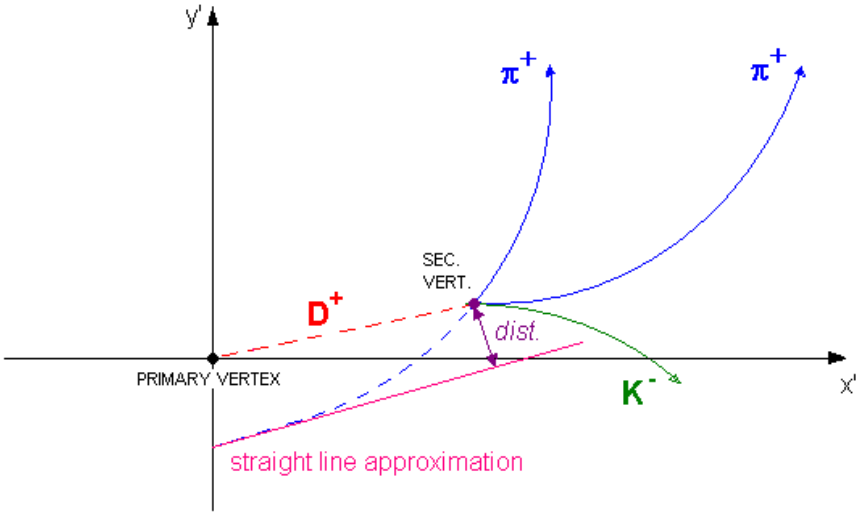

Figure 29: Sketch of the three-body decay $\mathrm{D}^+ \to \mathrm{K}^-\pi^+\pi^+$ with the illustration of the straight line approximation for one of the decay products. d is the distance between the secondary vertex and the tangent line. The reference system (x', y') represents the local coordinates of the tracking algorithm.

larger momentum and, thus, are less affected by multiple scattering. Instead, we observe that in the bending plane the resolution is worsening at high p_{t} of the D^+. This can be understood taking into account the fact that, as the D^+ transverse momentum increases, its decay products get more and more collinear with a direction given by the transverse momentum of the D^+. Hence, along this direction a worsening of the resolution of the secondary vertex finder should be observed. This behaviour is illustrated in Fig. 32, where the resolution is now calculated in a rotated reference system in the bending plane: x' is the direction of the p_{t} of the D^+. Along the x' coordinate the resolution increases at high p_{t}, while in the y' coordinate the r.m.s. goes down to ≈ 40 μm.

In Fig. 33 we report the secondary vertex position resolution and the pulls for the two-body decay $\mathrm{D}^0 \to \mathrm{K}^-\pi^+$. The resolution is very similar to that of the D^+, except at low p_{t} (below 1 GeV/c), where the two decay particles of the D^0 are essentially back-to-back (the Lorentz boost is small) and therefore the vertex is determined with worse precision along the direction of the decay particle momenta. This issue occurs only for two-body decays.

7 Conclusions

We have described the implementation and the performance of the three algorithms that are used for the reconstruction of the interaction vertex position in proton–proton collisions in ALICE. The first two algorithms use only the information from the two silicon pixel layers to provide the vertex position in either the three coordinates or only along the beam direction, thus they can be executed before track reconstruction. Their combined use allows to reconstruct the vertex in more than 95% of the events selected by the loosest minimum-bias trigger condition. Such a large efficiency will keep the correction for events with undefined vertex reasonably small. This is particularly relevant in the "first Physics" analyses on the charged particle multiplicity, which are based essentially on the SPD local reconstruction. The resolution in the z coordinate ranges from ~ 400 μm for events with

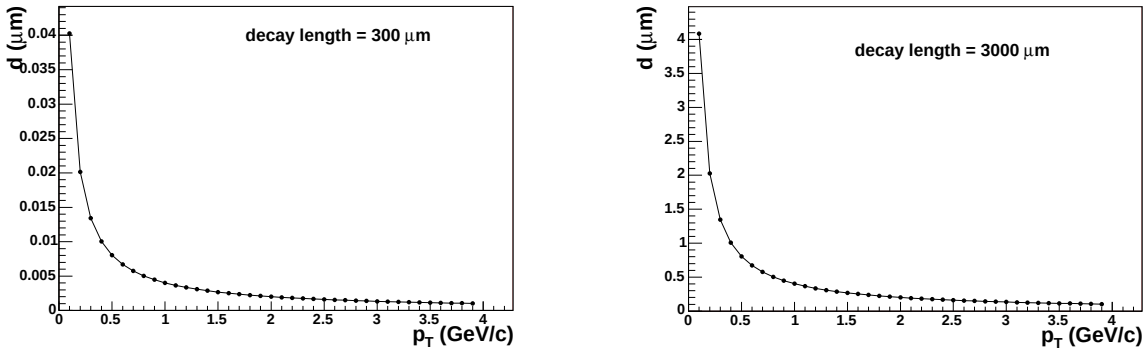

Figure 30: $D^+ \to K^-\pi^+\pi^+$ decay: distance between secondary vertex and the tangent line as a function of the transverse momentum of the particle, for decay lengths of 300 μm (left panel) and 3000 μm (right panel), with $B = 0.5$ T. See text for more details.

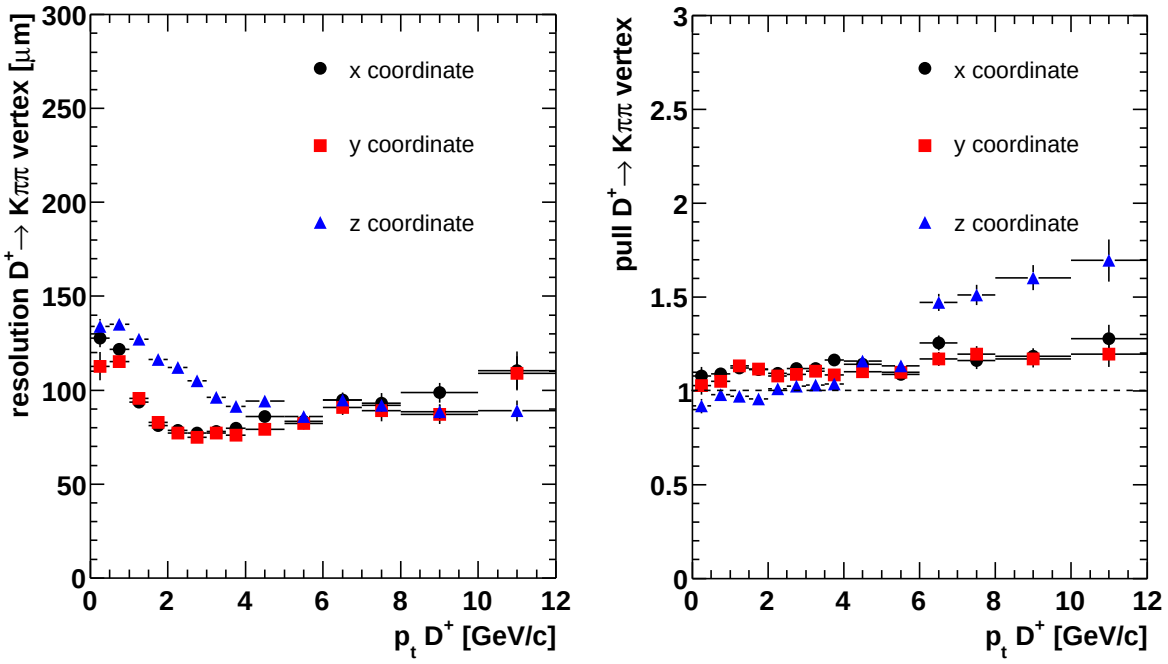

Figure 31: $D^+ \to K^-\pi^+\pi^+$ secondary vertex resolution (left) and pulls (right) for the x, y and z coordinates as a function of the p_t of the D^+.

few tracklets in the SPD and saturates at $\sim 70\,\mu$m at high multiplicity. In the transverse plane the figures are $\sim 500\,\mu$m and $\sim 120\,\mu$m respectively. The same algorithms are also used for the online monitoring of the interaction diamond position in the transverse plane. The third algorithm uses reconstructed tracks to provide the optimal measurement of the primary vertex position for more advanced analyses, with a resolution that at high multiplicities is $\sim 25\,\mu$m in the transverse plane and $\sim 40\,\mu$m in the z coordinate. This algorithm can be applied for tracks reconstructed in both the ITS and the TPC or for tracks reconstructed only in the TPC and it can use the information on the interaction diamond position, which allows to reach a large efficiency also for low-multiplicity events. The same algorithm is also used to reconstruct the decay vertices of charmed hadrons,

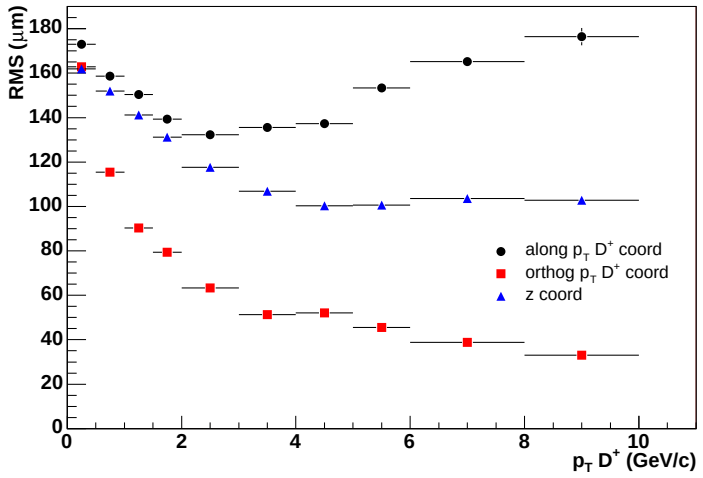

Figure 32: $D^+ \to K^-\pi^+\pi^+$ secondary vertex resolution for the x' (along the p_t of the D^+), y' (orthogonal to the p_t of the D^+), and z coordinates as a function of the p_t of the D^+.

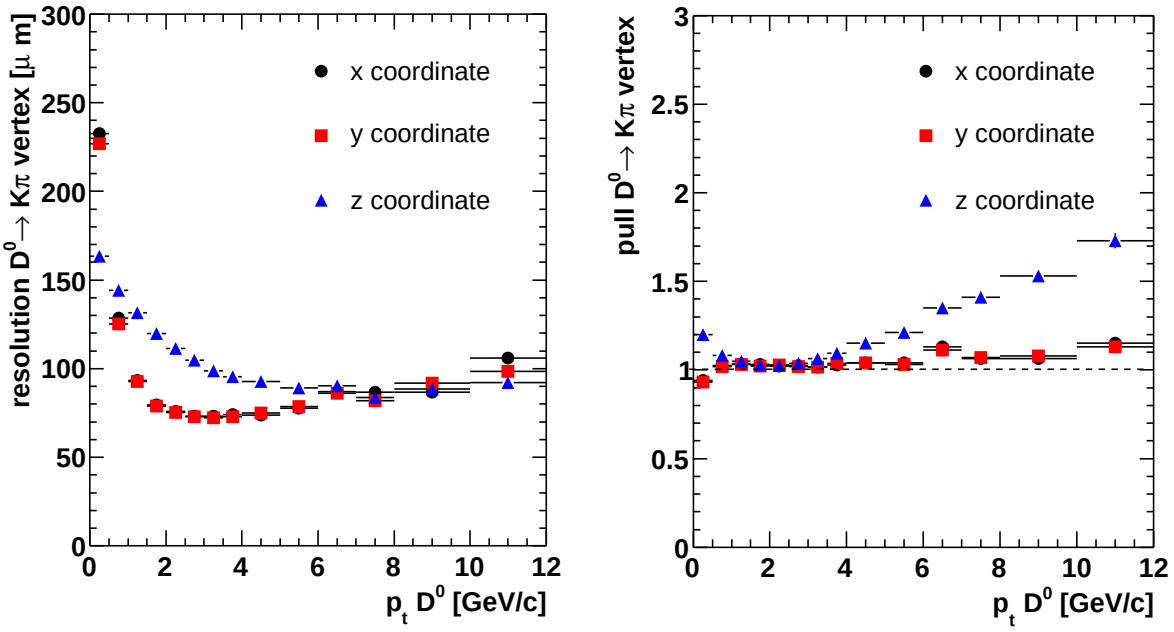

Figure 33: $D^0 \to K^-\pi^+$ secondary vertex resolution (left) and pulls (right) for the x, y and z coordinates as a function of the p_t of the D^0.

for which it provides a resolution of the order of 100 μm and accurately-estimated uncertainties, thus allowing to envisage good results for the measurement of D^0, D^+ and D_s^+ production [13, 12, 14, 15].

References

[1] A. Dainese and M. Masera, ALICE Internal Note 2003-27 (2003).

[2] http://aliceinfo.cern.ch/Offline

[3] ALICE Collaboration, Technical Design Report of the Inner Tracking System, CERN/LHCC/1999-12.
[4] ALICE Collaboration, Technical Design Report of the Time Projection Chamber, CERN/LHCC/2000-01.
[5] ALICE Collaboration, Technical Design Report of the Transition-Radiation Detector, CERN/LHCC/2001-21.
[6] K. Aamodt et al. [ALICE Collaboration], JINST 3 (2008) S08002.
[7] B. Batyunya, Yu. Belikov, K. Šafařík, ALICE Internal Note /PAT ALICE/97-24 (1997).
[8] F. Carminati et al. [ALICE Collaboration], Physics Performance Report Vol I, J. Phys. G30 (2004) 1517.
[9] T. Sjostrand et al., arXiv:hep-ph/0308153.
[10] http://wwwinfo.cern.ch/asdoc/geantold/GEANTMAIN.html
[11] V. Karimäki, CMS Note 1997/051 (1997).
[12] A. Rossi, Diploma Thesis (in Italian), Padova University (2006).
[13] A. Dainese, PhD Thesis, Padova University (2003), arXiv:nucl-ex/0311004.
[14] E. Bruna, PhD Thesis, Torino University (2007).
[15] S. Senyukov, PhD Thesis, Torino University (2009).