CERN SummerStudent 2024

Valerio DI BELLA

Data Driven studies of tracking efficiency with charm-hadron tag-and-probe method

Under the direction of FAGGIN Mattia
and GROSA Fabrizio

1 july 2024 to 14 September 2024

Contents

Chapter 1

Abstract

The aim of this analysis is to demonstrate the effectiveness of the Tag and Probe method to study the matching efficiency between the tracks reconstructed in the Inner Tracking System (ITS) and the Time Projection Chamber (TPC) of the ALICE detector.

Chapter 2

Experimental apparatus

2.1 ALICE

The goal of the ALICE (A Large Ion Collider Experiment) is to study the strongly interacting matter created in in hadronic collisions at ultrarelativistic energies.

ALICE is a large collaboration involving about 40 participating countries and about 2000 people. The detector is designated to study Pb–Pb collisions but can successfully be used for collecting data in proton–proton (pp) and proton–Pb (p–Pb) collisions too. It is composed of 18 sub-systems based on different technologies with the purpose of detecting all kinds of particles in different situations. The whole detector is 26 meters long and 16 meters in diameter, for a total weight of 10 000 tons. We will talk only about the most important detectors for the analysis.

2.1.1 Inner Tracking System

The Inner Tracking System (ITS) is a silicon tracking detector composed of seven cylindrical concentric layers. The internal one are 27.1 cm long and the external one goes to 147.5 cm long for the last one. The layers are paved with chips ALPIDE (ALICE Pixel Detector), that are used for particle detection. An ALPIDE chip is composed of three layers, the first one is a subtract, the second one is an active volume of semiconductor, that's the part sensible to the particles, the last one is composed of electronics, the all is 50 μm thick.

The radius of the inner most layer of the ITS is 22.4 mm, it has a pseudo-rapidity interval of $|\eta| \leq 1.3$, and a material budget of 0.36% X_0 for the internal barrel per layer and 1.10% X_0 for the external one per layer. The material budget is a value expressed with the radiation length X_0, this length correspond to the one needed for a particle to loose its energy by a factor of e^{-1}.

2.1.2 Time Projection Chamber

The Time Projection Chamber (TPC) is the main ALICE tracking detector used to reconstruct charged particle trajectories in order to reconstruct particle's momenta and perform particle identification (PID) via the measurement of ionization energy loss per unit length dE/dx.

The TPC is a cylindrical detector with the length of 500cm and the radius going from 82cm to 250cm, filled with the gas mixture $Ne/CO_2/N_2$ as (90/10/5). For its volume of 90 mm^3, it is a very light detector with the material budget of only 3% of X_0 and got a volume of 90 m^3 fill with active gas, this part is got 552 960 electrodes places on 158 layers to detect the incoming particles.

The readout chambers of the TPC empty an advanced technology called a Gas Electron Multiplier (GEM), that are stacks of foils with micro holes used to multiply the primary ionisation signal by creation of avalanches. With this new technology, the recording of collision data can be performed continuously a rate of 1 Tb$\cdot s^{-1}$ for a Pb–Pb collision at $50kHz$ interaction rate.

Chapter 3

Tag and Probe method

3.1 General method

The tag-and-probe method is used to infer information about the efficiency of detecting particles in data sets where particles are not pre-identified (tagged). Here we are working on the reconstruction of a $D^{\pm}$ mesons with the detection of the decay channel :

$$D^{\pm}\ (c.\tau \approx 309.8\,\mu\text{m}) \rightarrow \pi^{\pm}K^{\mp}\pi^{\pm}\ (B.R. \approx 9.38\%)$$

The tag and probe method adopted in this work, consists in two steps. In the first one, the $D^{\pm}$ decay vertex is reconstructed using only the two pion tracks (tag) and selections to the partially reconstructed decay topology as well as particle identification are applied in order to increase the purity of the selected tags. In the second step, a third track (probe) is combined with the previous two and the D+ mass is computed assuming it to be a kaon, in order to extract the actual D+ signal. No selections related to the decay topology or the particle identification information are applied in order to study only the track-reconstruction efficiency.

In order to study the effect of the track-reconstruction efficiency and ITS-to-TPC matching efficiency, different selection criteria are tested for the probe track. This way we can see reconstructed $D^{\pm}$ with a kaon seen in ITS without knowing if it has been seen in TPC, and we can do the inverse, finally we see if it has been seen in ITS and TPC. The tag will be the same in every case and then this gave us the information needed to compare.

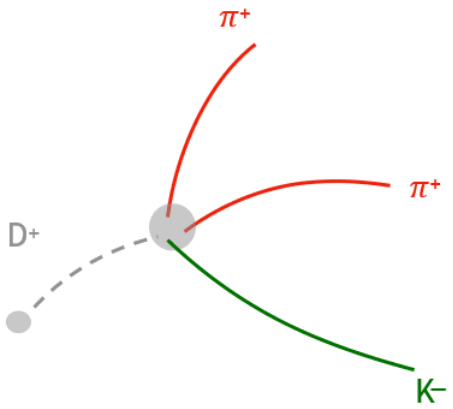

Figure 3.1: Sketch of Tag and Probe method.

Efficiency in particle detection is defined as the ratio of detected particles to the number of particles produced. In real data, unlike simulations, the actual number of produced particles is unknown, making direct efficiency calculations impossible. However, we can still gather useful information by examining relative efficiencies.

Specifically, we compare the D+ signal extracted by requiring the probe track to be reconstructed at least in ITS, at least in TPC, or using both the ITS and TPC detectors. The ratios between the signals extracted in the different cases are then compared to those obtained from MC simulations, to evaluate the level of agreement between data and MC.

Chapter 4

Main analysis

4.1 Tools and Data origin

I am using a software analysis framework named O2Physics, it is a tool developed by the ALICE team in order to analysis data from the LHC. We use it in order to do the analysis of collected data and of MC simulations anchored to them

Our analysis is done on data from the Run 3 of ALICE and compare with data from a MC simulation of the Run 3 of ALICE. The data from the Run 3 is the : LHC22 pass6 minBias APASS6 2022 pp data. The data used for the MC comparison and used in the BDT training is : LHC24d3[a,b] Charm and beauty enriched proton-proton PYTHIA8 simulated events with D mesons decays forced in the golden channels anchored to APASS6 2022 pp data.

In order to be able to treat this data we use Hyperloop because the totality of the data is to huge to treat it locally.

4.1.1 Preselections

After the first test we will see that there is to much background in order to do a proper analysis to resolve this problem we are looking into the distribution of signal and background for different variables. The variables are the length of flight, the transverse length of flight, the transverse length of flight normalised, the cosinus of the pointing angle and the cosinus of pointing angle on the transverse plane. By looking into them we are able to determine a set of selections before the BDT to remove already part of the combinatorial background, trying at the same time to keep as much signal as possible. Each p_{T} bins will represent a different Machine Learning model trained in the end.

p_{T} (GeV/c)	1-2	2-3	3-4	4-6	6-8	8-1000
L (cm) >	0.03	0.02	0.02	0.015	0.015	0.015
$\mathrm{L_{XY}}$ (cm) >	0.03	0.02	0.015	0.015	0.01	0.01
$\mathrm{NormL_{XY}}$ (cm) >	3	3	3	4	5	5
$\cos\theta_p$ >	0.8	0.8	0.8	0.8	0.8	0.8
$\cos\theta_{p_{\mathrm{XY}}}$ >	0.6	0.6	0.6	0.6	0.6	0.6

Table 4.1: Preselection made before the analysis

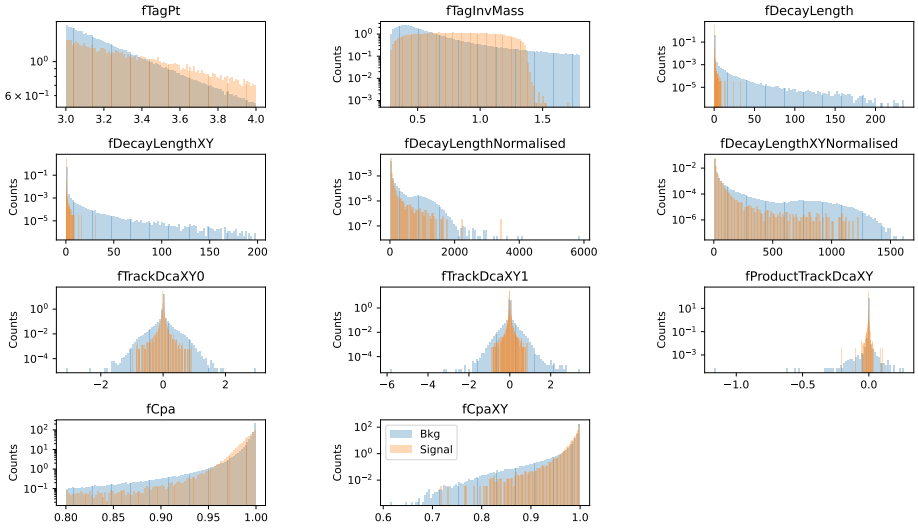

Figure 4.1: Graphics of the different distributions used in the Machine Learning training, in orange is the distribution for the signal and in blue for the background. Here is for the training of the algorithm with a p_{T} between 3-4 GeV/c.

4.2 Selection of tags with machine learning methods

4.2.1 Boosted decision trees

A decision tree is a simple algorithm that is defined by a succession of choices. The choices can be dynamically changed and therefore used as a based for a machine learning algorithm, a simple decision tree is considered as a poor learner and can't be enough as an answer for a complete problem like her. Hence we use what's called a Boosted Decision Trees algorithm or BDT. Each ending of a tree will then have a number as an output. All the output obtain for the totality of the trees are then summed in order to get the final value as the output of the Machine Learning algorithm. The output can be analogically considered as a probability of the output to be true or false here more accurately as signal or background. The last thing that we need to do is to considered a threshold in order to have a binary decision made at the end, we should consider that if the output value is above a certain value the output should be considered true. The choice of the threshold will be address later.

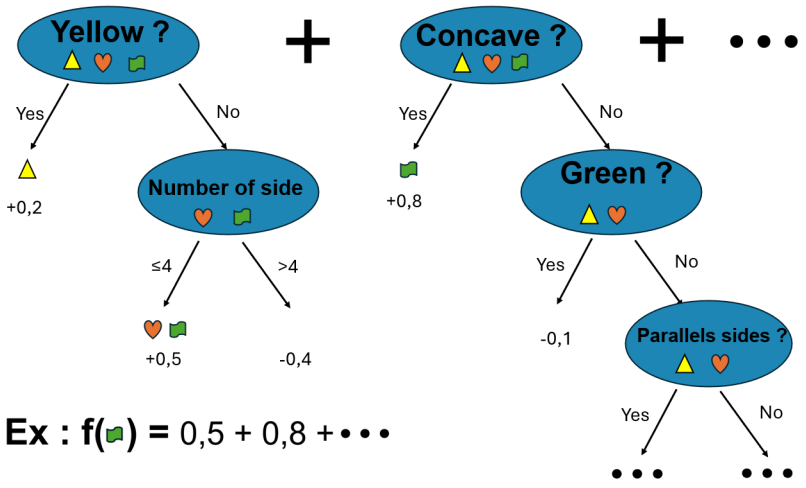

Figure 4.2: Example of a BDT algorithm.

4.2.2 Training

When we use a machine learning algorithm we need to do a training of the model we will use at the end. The training must be performed carefully, in order to avoid two major problems that are the over-training and under-training. The data used in our training are coming from a MC simulation, this way they are flagged with their truth origin. We ask the BDT to classify these data, and we modify the BDT parameters trying to minimize the classification errors. In the series of trees, when a part of a tree is detected as laying false result, the weigh of this part will increase in the next tree, this means that the tree n+1 is worst than the tree n. The meaning of this is to be able to detect and correct the problematic part more easily.

Before the training we need to separate the data set into a training set and a test set, we here use a 70/30%, we need to separate the data set because if the training is too long we could have a so called over-training where the model will fit the fluctuation of the training set. So we need to keep a test set to try after the training so that we can compare the results for the train data and the test data.

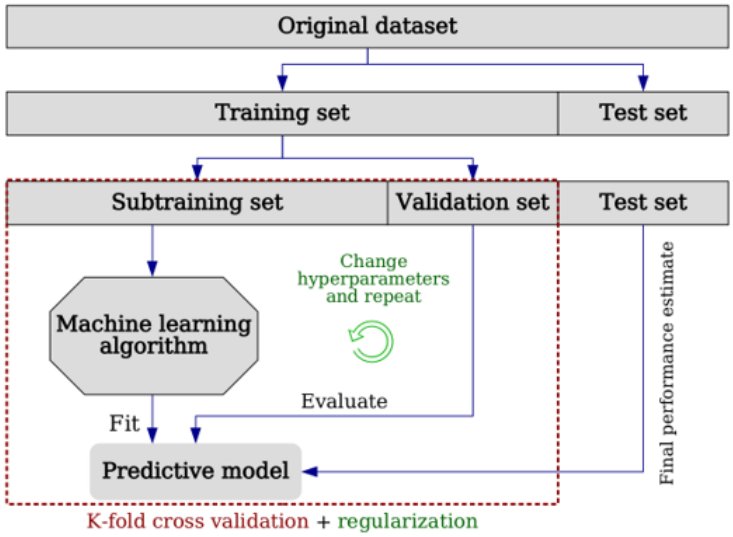

Figure 4.3: Scheme of the training of a BDT algorithm.

The set of background tags corresponds to pion pairs not coming from any heavy-flavour hadron decay from MC simulations. The use of a background coming from real data, by excluding the $\pi\pi$ mass region where we expect to have signal, wasn't giving sufficient results.

There is two types of parameters that are used in the training. The first type is the physics parameters, such as the decay length or the track impact parameters, they are the one that will be used in the method we will have at the end. The other one is what's so called hyper-parameters, they are parameters of the algorithm, they have nothing to do with the physics of our problem and have an interpretation only in the algorithm, for example we have the depth of the tree (the number of binary decisions), or the total number of trees. The hyper-parameters can be trained during the training process just as the other parameters but often they are just fixed at some "reasonable values". Here the difference between a training of hyper-parameter is really negligible and then we will not train hyper-parameter because is takes a way longer time to train the BDT if the hyper parameters are trained too.

4.2.3 ROC function and output of the training

When the model takes in input data and produces a result it can be either true or false. The possible classification outcomes are shown in the table 4.2 :

H1 : observation is signal	Consider H1 (Output Signal)	Consider not H1 (Output Background)
H1 is True (Real Signal)	True Positive	False Negative
H1 is False (Real Background)	False Positive	True Negative

Table 4.2: Table of confusion

Clearly, our aim is to maximise the number of true (positive and negative) classifications, keeping at the same time the number of False (positive and negative) classifications as low as possible. The exact requirements depend on a particular case. But in general, a compromise can be achieved by adjusting the threshold applied to the output of the BDT. The threshold we used is 0.9 for a p_{T} between 1-4 Gev/c and 0.8 for above 4 Gev/c.

Any binary-decision model can be characterised by so called ROC curve, the ROC curve as in abscissa the False Positive rate and in ordinate the True Positive one. The ROC is created by adding a point for every threshold creating at the end a curve. A more successful model would result in ROC curve having a larger area under it, as compared with the same curve for a less successful model. We do a comparison between the ROC curve on the training data set and the test set in order to be able to see if there is over-training, this could be seen if the difference between the two is to high. We can see a comparison of on the figure 4.5

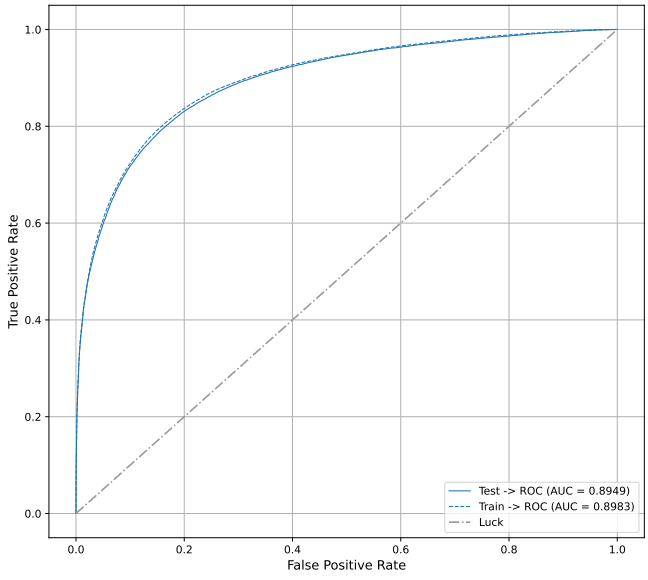

Figure 4.4: Representation of the different ROC curve for a same BDT model, here at a p_{T} between 2-3 Gev/c.

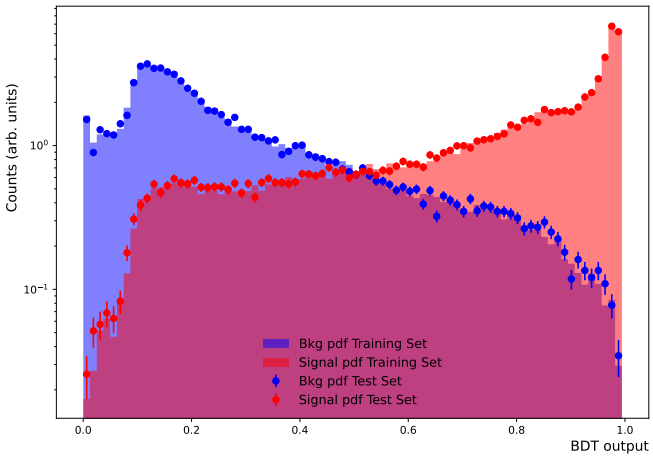

Figure 4.5: Representation of the different value for the output of the training, here at a p_{T} between 3-4 Gev/c.

4.3 Selections of Tag

By using a Machine Learning algorithm we aim aim to select a good sample of same-charge pion tracks for the tags.

The main goal during an analysis like that is to be able to differentiate the background and the particles that are important in our analysis. So one of the things that we could do in order to help the differentiation is to use the properties of the daughter particles detected that reconstructs our candidate and the placement of the vertex of decay and use theses properties in order to recreate new quantities that are really different for the background and the particles. The properties we have at the beginning are different placement of vertex and particles and their direction of movement.

There are two types of selections that we made in our analysis, the first one are the crude one, done without the use of Machine Learning, the second one are more precise selections done with a Machine Learning algorithm, the necessity of the Machine Learning is really evident when we understand that we are trying to minimize a shape in a space of 15 dimensions. I will explain some parameters.

θ_{p} : **pointing angle.** The angle between the direction of propagation seen in the lab frame at the vertex of decay (assumed) and the line create with the primary and secondary vertex. When doing the cosinus of this angle we should get a sharp pic à 0 and a sprawl after that due to the background.

θ_{p}^{xy} : **pointing angle on the transverse plane.** The resolution is better in the transverse plane than the one in the longitudinal plane.

L : **length of flight.** The length of segment between the primary and secondary vertex in the lab frame. This is taken as a straight line so this does not take into account the magnetic field.

$L_{xy}/\sigma_{L_{xy}}$: **the transverse length of flight normalised.** The projection in the transverse space is done for the same reason as before, the transverse space got a way better resolution. The division by the incertitude tends to erase the sprawl of the result due to the background, because when the incertitude is high like for the background the result normalised will then be weaker that is why we bound it from below.

The idea is so have the most discriminating parameters this way the algorithm will be able to cut the phase space on places where the background is high and the density of count from real particles is low.

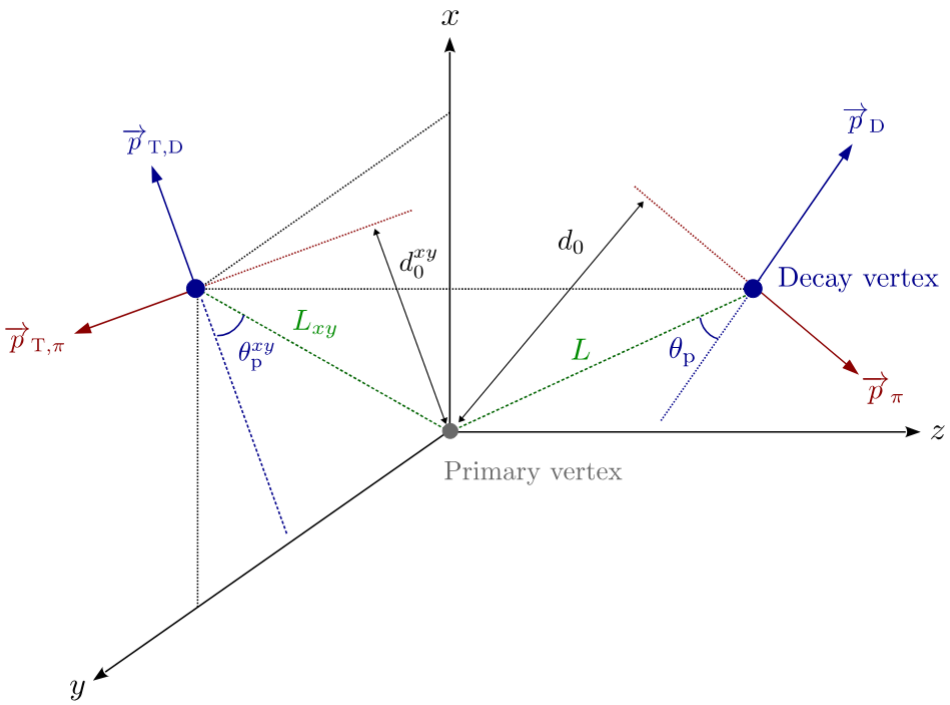

Figure 4.6: Representation of some of the topological variable in the decay of meson $D^{\pm}$ with only one daughter for simplicity.

4.4 Mass fitting

After the analysis we do a multi-dimensional histogram with 10 axis, as a THnSparse object, what we use is the projection on the Invariant mass reconstructed of the D^+ cut on the p_T of the probe, the bins are : [0.5,1,1.5,2,2.5,3,3.5,4,5,6,7,8,9,10,12] GeV/c. The next step will then be fit the invariant mass, in order to integrate the fitting function to get the final yield. In order to do this fitting we are using a python library named Flarefly. The fitting is done with a Gaussian for the signal and for the background an exponential at low p_T and an exponential + power law at higher p_T. We have a significance that goes from 6.3 to 49.6 at 3σ and a S/B that goes from 0.01 to 0.05 at 3σ too.

A control tool to help in the fitting is the raw residuals that is given automatically by Flarefly, this is a good help to see if the results of the fitting are working or not. We stop the fitting at 12 GeV due to limited size of the available dataset.

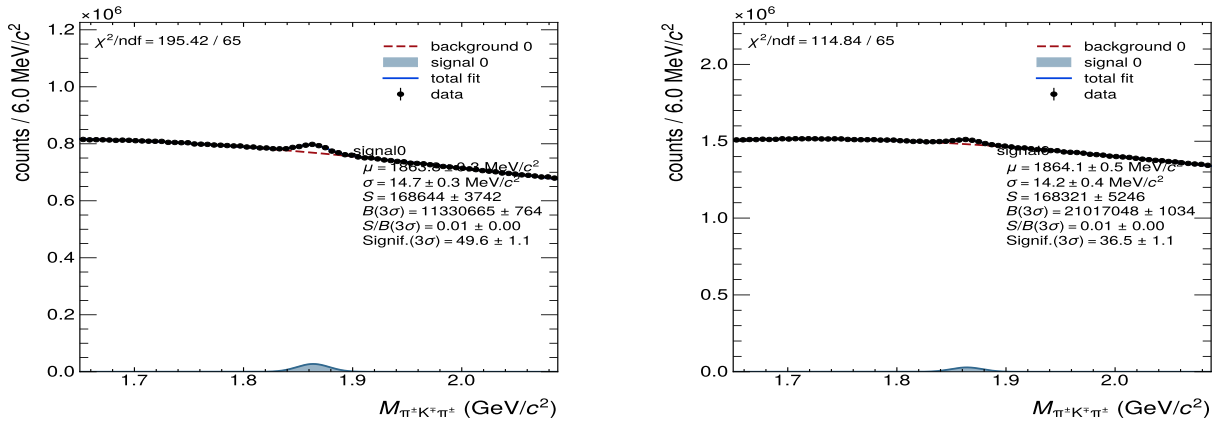

Figure 4.7: A mass invariant plot with a Gaussian in order to fit the peak and an exponential in order to fit the background. The bin is 0.5-1 GeV. At left : the ITS part of the detector. Right : The TPC part of the detector.

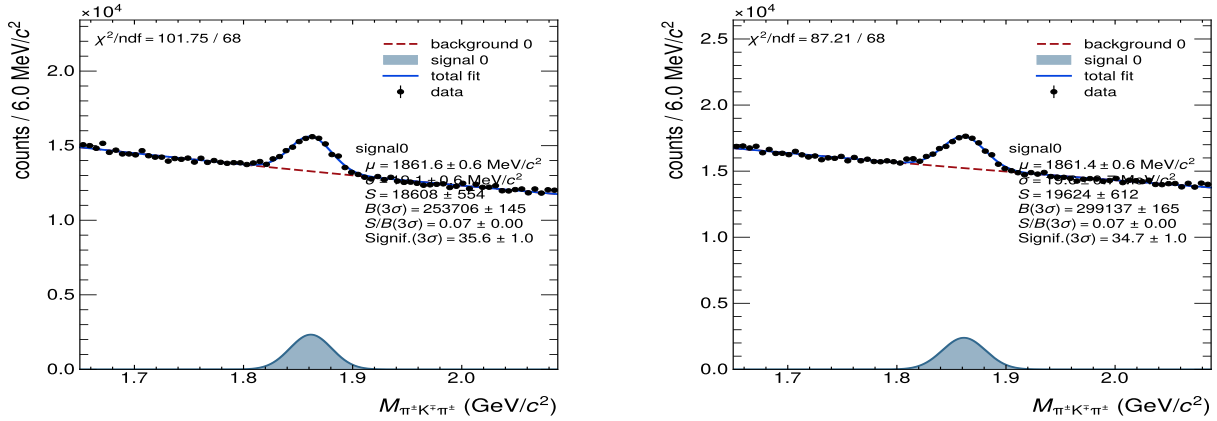

Figure 4.8: A mass invariant plot with a Gaussian in order to fit the peak and an exponential in order to fit the background. The bin is 3-3.5 GeV. At left : the ITS part of the detector. Right : The TPC part of the detector.

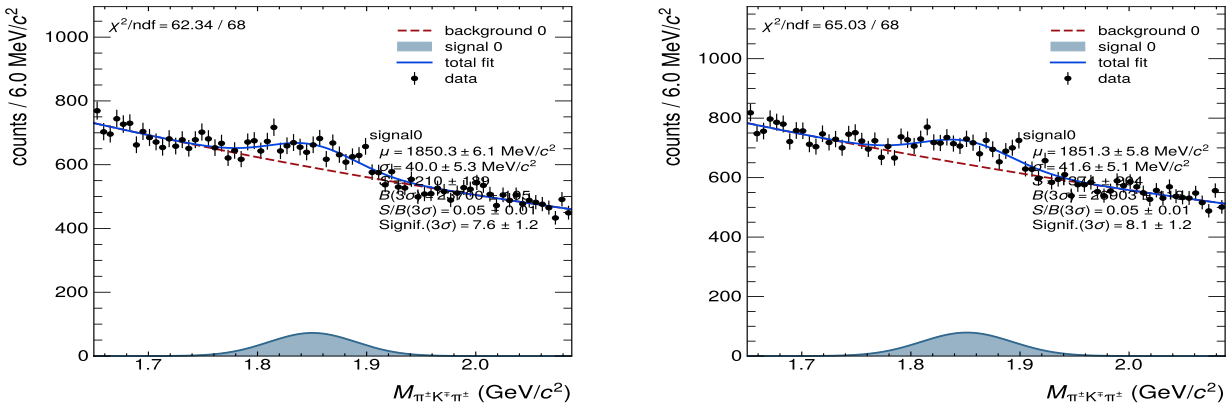

Figure 4.9: A mass invariant plot with a Gaussian in order to fit the peak and an exponential in order to fit the background. The bin is 10-12 GeV. At left : the ITS part of the detector. Right : The TPC part of the detector.

Chapter 5

Results

The aim of this project was to be able to get information on the matching and tracking efficiency on the different part of the detector by a use of a Tag and Probe method, and to compare this results to a MC simulation in order to see if our results are possible or not.

The ratio between the D+ raw yields obtained with the different track selection criteria for the probe track were computed and compared with the ratios obtained from MC simulations. Figure 5.1 shows the ratio between the D+ raw yields obtained by requiring the probe track reconstructed both in ITS and TPC and those obtained with the probe track reconstructed at least in the ITS (left panel) or TPC (right panel). The blue points show the results obtained in data, while the red ones in the MC.

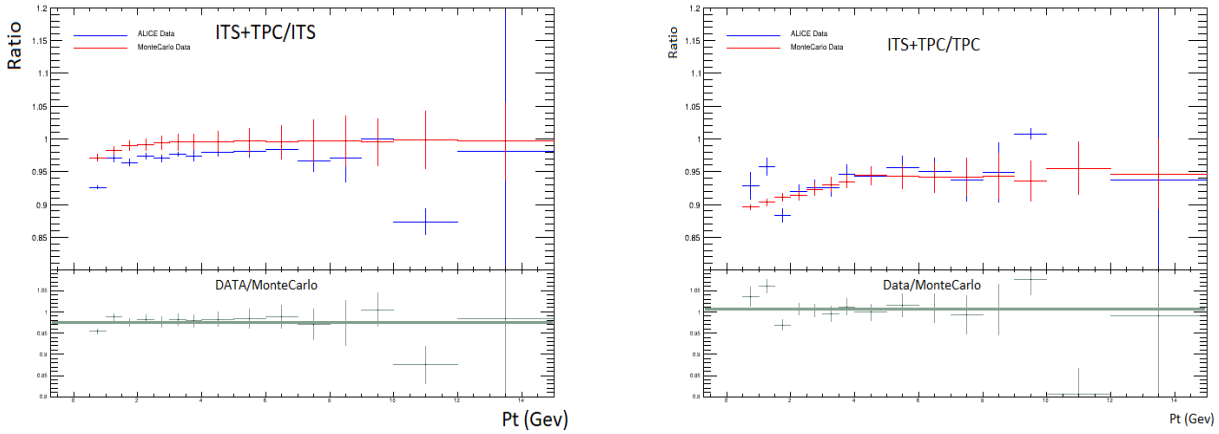

Figure 5.1: Ratio of the efficiency of all the mesons reconstructed and the one with a probe detected in Left : ITS, Right : TPC. Below is the ratio of the value for the Data divided by the value for a MC simulation.

By looking into this different results we are able to gain information about the probability that an ITS track is matched in TPC and vice versa. We can see that the matching in the ITS in comparison to the one for the ITS and TPC is close in real data and in MC, in fact it is higher in MC by about 3%. For the TPC we can see that the value are similar between Monte Carlo simulation and in real data.

If we do the ratio between the yield of particles in ITS and in TPC we see that the result is higher in real data than in MC, this could be explained because the efficiency is higher in

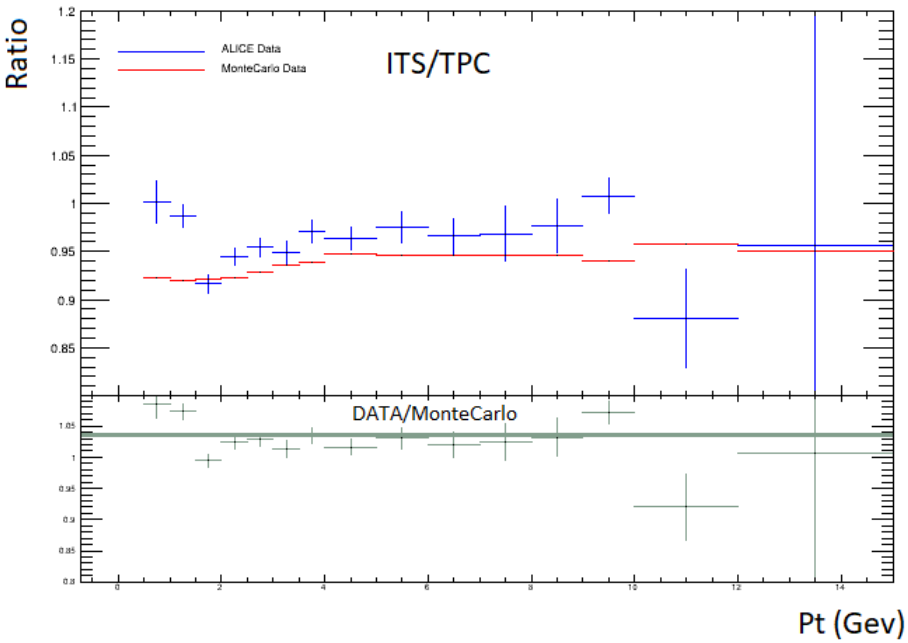

Figure 5.2: Ratio of the efficiency of all the mesons reconstructed in ITS with the one reconstructed in TPC. Below is the ratio of the value for the Data divided by the value for a MC simulation.

real data for ITS or higher in Monte Carlo for TPC, we could speculate that it is because the efficiency is smaller in Real data than in MC so we assume that it is due to the less part in TPC.

Chapter 6

Conclusions and perspectives

In the end we examine a new data-driven method to study the matching efficiency between ITS and TPC by exploiting 3-prong HF hadron decays and see that the result are encouraging in the use of this method in the futur. The analysis was successful for p_{T} being inside a interval of [0.5,12] GeV we need to stop because there was a lack of statistics above. This is really interesting and could help us a lot to have information of different matching efficiency that are usually only calculated with simulation and then can only be trust as much as we trust the simulation.

We could continue the work above the 12GeV p_{T} if we could have more statistics. An other thing that we could do is to exploit theses studies to evaluate systematics uncertainties for tracking efficiency of primary tracks. The next step would then to extend the study to different decays channels, track selections and collisions systems.