Efficiency and purity determination of the particle identification method in the Central Exclusive Production process for the STAR experiment at RHIC

Project for the Python in the Enterprise by T.Szumlak

Łukasz Fulek, Rafał Sikora

September 13, 2016

Abstract

This is the report explaining basic concepts and features of the program prepared as an assesment project for the Python in the Enterprise, implemented under python in version 2.7. It contains description of all components of the program and short discussion of the results of analysis, which this program was intended to enable.

1 Project objectives

The main (initial) goal of our project, whose name was "Machine learning – particle identification (PID) for the STAR experiment using neural net approach", was to develop an algorithm which would use a machine learning to enhance efficiency and purity of particle identification in the STAR experiment at RHIC in a specific class of particle reactions called "diffractive interactions". In this class of interactions STAR detector does not provide an information about the time of collision, because no particles are produced in the mid-forward region (pseudorapidity η of approximately 4-5), and therefore standard procedure of reconstruction of inversed particle velocity β^{-1} can not be applied (for more detailed explanation see [1]).

Because of a few reasons we decided to partially change the topic of our project. First, we have received recommendations from the physics working group (PWG) in the STAR experiment to postpone development of the machine learning PID, because such approach is not widely used in other STAR analyses and usage of such would be problematic in terms of review of paper proposal in the future. Second, experiment have some difficulties with the detector geometry implementation in the GEANT3 environment and thus production of Monte Carlo (MC) samples with detector response simulation is not possible at the moment, therefore it would be helpful to have some evaluation of currently used PID method.

Given facts above the goal of our project was set to write a toy-MC generator simulating the detector response for the case of exclusive production of two oppositely charged particles, and determine the efficiency and purity of the "nominal" particle identification procedure.

2 Particle identification method

In the case of Central Exclusive Production process, which at high center-of-mass energies occurs mostly via the Double Pomeron Exchange (Fig. 1), in the mid-rapidity region neutral state X is produced, which then decays dominantly into a pair of opposite-charge particles, e.g. $\pi^+\pi^-$. In the analysis of this process we use reconstructed observables from the STAR detector such as track length, track momentum, track energy loss dE/dx (all three from the Time Projection Chamber (TPC)) and time of hit detection in the barrel Time-Of-Flight (TOF) system. The two-particle event is schematically presented in Fig. 2.

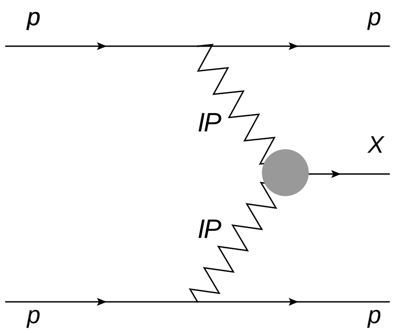

Figure 1: Feynman diagram of Double Pomeron Exchange process.

In the nominal PID method, whose evaluation has been performed with the help of prepared python program, we use the so called nSigma variables, defined as

$$n\sigma_Y = \log \frac{\mathrm{d}E/\mathrm{d}x_{\mathrm{measured}}}{\mathrm{d}E/\mathrm{d}x\;_{Y,\mathrm{Bichsel}}}. \tag{1}$$

Such quantity should be understood as the measure of deviation of measured dE/dx from the dE/dx expected from the particle of given momentum, assuming that this particle is of type Y. One can find exemplary dE/dx distribution as a function particle momentum in Fig. 2. With the solid lines Bichsel parametrisation of dE/dx is presented, which nicely describe data points.

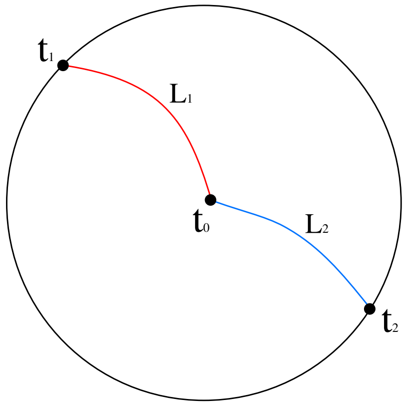

Figure 2: Sketch of the exclusive event with two opposite-sign tracks in the magnetic field of the STAR Time Projection Chamber.

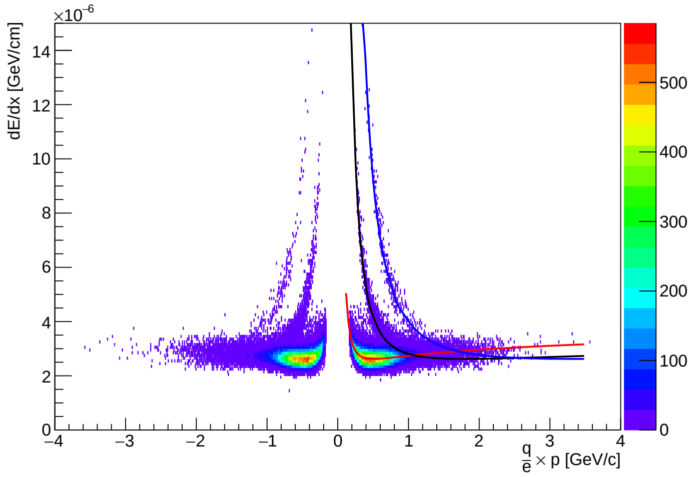

Figure 3: Particle energy loss in TPC, dE/dx, as a function of charged particle momentum (STAR data). Colored lines depict dE/dx parametrisation, which agree well with the data.

Having defined such variables for both tracks, we form the measure of probability that a pair of tracks is of the same type Y:

$$n\sigma_Y^{pair} = \sqrt{\left(n\sigma_Y^{trk1}\right)^2 + \left(n\sigma_Y^{trk2}\right)^2}. \tag{2}$$

In addition to information from the TPC we use time of hit detection in the barrel TOF subsystem. From the simple algebra describing relation between track lengths, momenta and times of hit detection one can derive formula for the squared mass of two particles, assuming that their masses are equal (particles are of the same type):

$$\begin{cases} t_1 - t_0 = L_1\sqrt{1 + \frac{m_1^2}{p_1^2}}, \\ t_2 - t_0 = L_2\sqrt{1 + \frac{m_2^2}{p_2^2}}, \end{cases} \tag{3}$$

$$\Delta t = t_1 - t_2 = L_1\sqrt{1+\frac{m_1^2}{p_1^2}} - L_2\sqrt{1+\frac{m_2^2}{p_2^2}}. \tag{4}$$

Assuming $m_1 = m_2 = m \quad \rightarrow \quad m^2$ from quadratic eq.

Parameters of the quadratic equation whose solution is suqared mass and the final formula for m^2_{TOF} are given below:

$$a = -2\frac{L_1^2 L_2^2}{p_1^2 p_2^2} + \frac{L_1^4}{p_1^4} + \frac{L_2^4}{p_2^4}, \tag{5}$$

$$b = -2L_1^2 L_2^2\left(\frac{1}{p_1^2} + \frac{1}{p_2^2}\right) + \frac{2L_1^4}{p_1^2} + \frac{2L_2^4}{p_2^2} - 2\left(\Delta t\right)^2\left(\frac{L_1^2}{p_1^2} + \frac{L_2^2}{p_2^2}\right), \tag{6}$$

$$c = \left(\Delta t\right)^4 - 2\left(\Delta t\right)^2\left(L_1^2 + L_2^2\right) + L_1^4 + L_2^4 - 2L_1^2 L_2^2, \tag{7}$$

$$m^2_{TOF} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \tag{8}$$

Using variables defined in Eq. 2 and 8 we can now present the logic of the nominal PID algorithm:

```
  if   $n\sigma^{pair}_{pion} > 3$ & $n\sigma^{pair}_{kaon} > 3$ & $n\sigma^{pair}_{proton} < 3$ & $m^2_{TOF} \in (0.7, 1.1)$      $\rightarrow pp$
elif   $n\sigma^{pair}_{pion} > 3$ & $n\sigma^{pair}_{kaon} < 3$ & $n\sigma^{pair}_{proton} > 3$ & $m^2_{TOF} \in (0.2, 0.32)$    $\rightarrow KK$
elif   $|n\sigma^{trk1}_{pion}| < 3$ & $|n\sigma^{trk2}_{pion}| < 3$                                                    $\rightarrow \pi\pi$
else   PID fails
```

3 Implementation

Created software consists of the following files:

1. Main.py - main program file
2. EventGenerator.py - class generating exclusive events
3. EventReconstruction.py - class transforming true-level data to detector-level
4. ExclusiveEvent.py - class representing exclusive event
5. ParticleId.py - class representing type of the particle (of an exclusive pair)
6. StarDetectorAcceptance.py - class containing information about STAR detector acceptance
7. TrackingSimulation.py - class performing track propagation in the STAR magnetic field
8. dEdxParametrisation.py - class providing Bichsel parametrisation of dE/dx in the STAR TPC
9. PlottingHistograms.py - class for histograms plotting

Program executes a loop with a customizable number of iterations. In each loop one event is simulated, which starts from determination of the type of particles in a pair ($\pi\pi$, KK or pp) according to the particle yields extracted from the STAR data. Then momenta of particles are read from the input files, which contain output of the GenEx generator [3] which provide momenta of exclusively produced particle pairs according to the Lebiedowicz-Szczurek model [4]. Particles are checked whether they fit in the STAR acceptance - if not, new set of momenta is read until they fulfill acceptance limitations. Particles are then tracked in the magnetic field of the STAR main detector using Newton's method. From this one obtains positions of the TOF module hit by the particles and thus one is able to recontruct the length of the tracks from the vertex to the TOF modules. Next, time of hits detection in TOF are reconstructed, and appropriate timing resolution of 100 ps is accounted. After all necessary quantities are known, dE/dx of both tracks are simulated using Bichsel parametrisation of the TPC response, and finally nSigma variables and m^2_{TOF} are calculated in the very same way as the real data. For every event PID algorithm is used to determine particle types, which are compared with true-level PID. Also histograms of many other quantities are filled and draw at the end of program execution.

4 Program usage

1. Download code from the GitHub repository [1]
2. Set number of events to simulate and analyse in Main.py
3. Start program with a standard command
   :~$ python Main.py
4. Once program execution is done you shall find output files in the Output directory.

5 Results

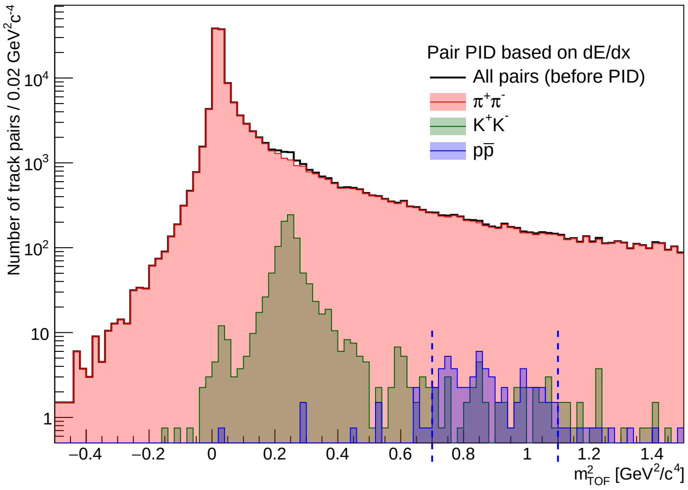

Figure 4: Squared mass m^2_{TOF} (using Eq. 8) in the desceribed PID procedure.

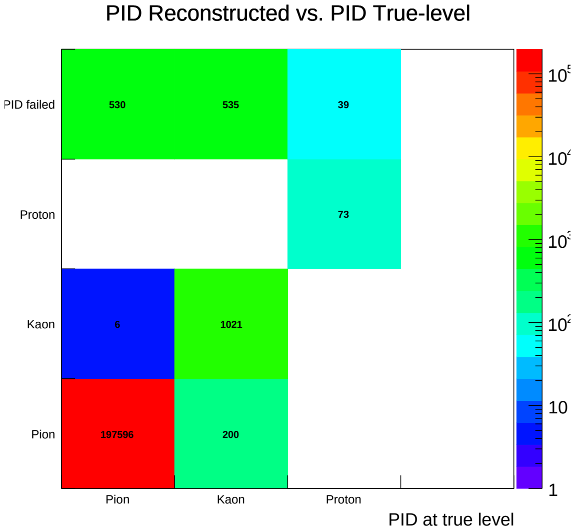

Figure 5: Response matrix of the PID algorithm described in the text.

$$\text{Efficiency} = \epsilon = \frac{\text{number of XX pairs at true-level identified as XX}}{\text{number of XX pairs at true-level}} \tag{9}$$

$$\text{Purity} = \psi = \frac{\text{number of XX pairs at true-level identified as XX}}{\text{number of pairs identified as XX}} \tag{10}$$

	Efficiency ϵ	Purity ψ
$\pi^+\pi^-$	99.7%	99.9%
K^+K^-	58.1%	99.4%
$p\bar{p}$	65.2%	100.0%

Table 1: Efficiency and purity calculated based on simulation results depicted in Fig. 5 using Eqs. 9 and 10.

References

[1] M.Shao *et al.*, *Extensive particle identification with TPC and TOF at the STAR experiment*, Nucl. Instrum. Meth. **A558** (2006), 419-429, arXiv:nucl-ex/0505026.

[2] `https://github.com/rafalsikora/PITE_PidForSTAR`

[3] R.A. Kycia, J. Chwastowski, R. Staszewski and J. Turnau, *GenEx: A simple generator structure for exclusive processes in high energy collisions*, arXiv:1411.6035 [hep-ph].

[4] P.Lebiedowicz and A.Szczurek, *Revised model of absorption corrections for the $pp \to pp\pi^+\pi^-$ process*, Phys.Rev. **D92** (2015) no.5, 054001, arXiv:1504.07560 [hep-ph].