Particle Identification (PID) in FairShip

Behzad Hosseini
For INFN-Cagliari
7th SHiP Collaboration Meeting
10 February 2016
Outline

1) PID criteria and selections.
2) Explaining the structure of the PID code.
3) How to use the PID information in data analyzing (for all users).
4) As an example I’ll show the application of the particle identification using “simple” cuts for Muons, Hadrons and Electrons.
Introduction

• The aim is to explain how PID is implemented in FairShip.

• I’ve adopted a step by step approach; therefore, any modification and using specific cuts easily can be implemented to the code.

• However, in order to have a unique PID information to be used by other users, please send your commendeds and modifications to us to implement them into the code.
PID criteria and selections
PID criteria and selections

• **Selection variables for the muonid:**
  – Difference in position of extrapolated tracks and the position of hits (inside each pad).
  – Depth in 4 layers of the muon detector due to the momentum of tracks.
  – Number of related hits.

• **Field of interests for mounid:**
  – Each layer digitized with the pads, each of them having a size of 5x5 cm².
  – MuonPoint hits related to each pad, are merged together. Therefore, there is one single hit per each pad.
  – PID depends on momentum and for each momentum is interval (which is customizable); then, I ask for hits to be present at a given depth around a number of pad area depending on the momentum of the track. If the number of hits around these number of pad area are 1 or 2, the track is identified as a muon.
PID criteria and selections

• Selection variables for electron:
  – Difference in position of extrapolated tracks and the clusters in Ecal detector.
  – Value of E/P.

• Hadron detector:
  – The Hcal information also will be add to this work when this information will be ready.
PID criteria and selections

• The main strategy is:

1) First extrapolate tracks up to the end of the detector.

2) Using the information of muon detector check if the track is a muon.

3) If not, Check the electronid by using the Ecal information.

4) If it is not muon and electron, it is a hadron.
The muon identification using some simple cuts

1000 events with a single muon are generated using Pgun with momenta in the range of (1-50 GeV).
Difference in position of Muon detector hits and extrapolated tracks (layer 1)

Each layer digitized with 5x5 cm² pads

\[ \Delta X = X_{\text{extrapolated}} - X_{\text{hit}} \]
\[ \Delta Y = Y_{\text{extrapolated}} - Y_{\text{hit}} \]
\[ E = \text{sum of the energy of hits in one pad} \]
Difference in position of Muon detector hits and extrapolated tracks (layer 2)

\[
\Delta X = X_{\text{extrapolated}} - X_{\text{hit}} \\
\Delta Y = Y_{\text{extrapolated}} - Y_{\text{hit}} \\
E = \text{sum of the energy of hits in one pad}
\]
Difference in position of Muon detector hits and extrapolated tracks (layer 3)

\[ \Delta X_3 = X_{\text{extrapolated}} - X_{\text{hit}} \]
\[ \Delta Y_3 = Y_{\text{extrapolated}} - Y_{\text{hit}} \]
\[ E = \text{sum of the energy of hits in one pad} \]
The hadron identification using some simple cuts

1000 events with a single proton are generated using Pgun with momenta in the range of (1-50 GeV).
Difference in position of proton detector hits and extrapolated tracks

$\Delta X_1$  $\Delta Y_1$

$\delta X_1$  $\delta Y_1$

Entries 129
Mean 0.07692
Std Dev 3.849

Entries 129
Mean -0.2262
Std Dev 5.328

$E_1$

Entries 129
Mean 1.264
Std Dev 0.933
Difference in position of proton detector hits and extrapolated tracks

\[ \Delta X_2 \quad \text{delta } X2 \]

\[ \Delta Y_2 \quad \text{delta } Y2 \]

\[ E_2 \]
The electron-pion identification using some simple cuts

1000 events with a pion and electron in their final state are generated with momenta in the range of (1-50 GeV).
Difference in position of cluster and extrapolated tracks

ΔX position

ΔY position
Energy over Momentum

<table>
<thead>
<tr>
<th>EP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Entries</td>
<td>1103</td>
</tr>
<tr>
<td>Mean</td>
<td>0.3276</td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.418</td>
</tr>
</tbody>
</table>
Structure of the Code
The structure of our code

- A Pid folder is created under Fairship and the Pid Class is written under C++

- The Particle identification code is written under “python folder” using python program which includes different parts such as executing part and analysis part that it will be executed in ShipReco

- Then it brings us to have a Pid container in “ship.10.0.Pythia8-TGeant4_rec.root” file

Commends used in ShipReco:

```python
import shipPid

SHiP.Pid.execute()
```

Many thanks to Thomas for his support and his infinite help.
Pid container
Is it an Muon?

• As a first step, the code checks if the track is a Muon using the information given by Muon detector.
• This has been done based on:
  – Difference in position of extrapolated tracks and the position of hits (inside each pad).
  – Depth in 4 layers of the Muon detector due to the momentum of tracks.
  – Number of related hits.
• Is it a muon?
  – Yes => MuonID = 1
  – No => MuonID = 0 (and check the Electron)
  – If the information is not enough => MuonID = -1
  – If it is out of the Muon detector acceptance => MuonID = -2
Electron and Hadron

• Check if it is an Electron using the Ecal information:
  • If it is electron => ElectronID = 1
  • If it is not muon and not electron => HadronID = 1
Using the PID information
Using Pid in ShipAna

The data analysis experts who do the physics analysis regarding the HNL events, such as Nico, now easily can extract the PID information in the ShipAna.

```python
for apid in sTree.Pid:
    trackid=apid.TrackID()
    if apid.MuonID():
        track = sTree.FitTracks[trackid]
        ...
    if apid.ElectronID():
        track = sTree.FitTracks[trackid]
        ...
    if apid.HadronID():
        track = sTree.FitTracks[trackid]
        ...
```
Add versioning

• We are planning to add a versioning to this work. In this way there will be no confusion to which PID version people refers in her/his analysis. It will be printed when running ShipAna so that users will know which version are they using; as example “PIDv1.0”.

• Also an “history file” will be created in order to store all changes that applied to the PID code during the time.
The application of muon, electron and hadron identification by applying simple cuts
Electron, Hadron and Muon identification

1) 1000 events with a single muon are generated using Pgun with momenta in the range of (1-50 GeV).

- 996 muon tracks have been reconstructed.
- Using our code, 960 tracks are selected correctly as Muon.
- 36 tracks misidentified as hadron which they mostly have momentum lesser than 3 GeV.
Electron, Hadron and Muon identification

2) 1000 events with a single proton are generated using Pgun with momenta in the range of (1-50 GeV).

- 997 protons have been reconstructed.
- 994 protons are selected correctly as Hadron.
- 3 tracks misidentified; 2 of them as electron and one as muon.
3) 1000 events with a single electron are generated using Pgun with momenta in the range of (1-50 GeV).

- 995 electrons have been reconstructed.
- 978 tracks are selected correctly as Electron.
- 17 tracks misidentified as hadron.
Conclusion and Future Plans

• Particle Identification works very well and it is in the level to be committed in fairShip and be checked and used by the collaboration.

• In addition, Hcal information could be also added to this work and for that we will need the help of Hcal experts.

• The next step will be to improve our work by using the more advanced analysis methods such as BDT, Likelihoods or NN.

• We also plan to add the related histograms to ShipAna.