

Table of Contents

Combined PID in DC04.....	1
Basic Information on DC04 Combined PID.....	1
Additional Information.....	2

Combined PID in DC04

Basic Information on DC04 Combined PID

The technique used for combining the information from the different subdetectors was presented during the Tracking-PID-physics meeting of the last LHCb week, which you can find at <http://agenda.cern.ch/age?a03997>

The product is made of the likelihoods for each particle type from the different detectors. Particles are then selected by cutting on the ratio of likelihoods between different hypotheses, or equivalently on the difference of log-likelihood (DLL), eg:

$$\text{DLL}(e\text{-}\pi) = \ln L(e) - \ln L(\pi) = \ln [L(e) / L(\pi)]$$

for the electron and pion hypotheses.

The output of the combination is, for each track, a list of the detector information that is available (RICH/CALO/MUON) and the 4 log-likelihood differences for the (e, mu, K, p) hypotheses compared to the pion hypothesis. Any other DLL can be simply determined from these, eg. $\text{DLL}(K\text{-}p) = \text{DLL}(K\text{-}\pi) - \text{DLL}(p\text{-}\pi)$.

To select particles of a given type, one then specifies which detector information is required to be available, and which cut (or cuts) on DLL values are required. For example, to select electrons, one would normally require the Calorimeter information to be available, as this is the most important detector element for electron ID. The most significant background is from pions, so one cuts on the DLL between the electron and pion hypotheses. This DLL is distributed mostly to positive values for electrons, and negative for pions, so a reasonable choice for the cut is $\text{DLL}(e\text{-}\pi) > 0$. As shown in the presentation mentioned above, this gives an efficiency of 95% for electrons that are within the Calorimeter acceptance, with a pion misid rate of about 0.7%. The pions could be even more strongly suppressed by increasing the cut on $\text{DLL}(e\text{-}\pi)$, at the cost of a loss in electron efficiency. Reasonable values for the cut on DLL values are usually in the range -10 to +10, according to the compromise between efficiency and purity.

The combined particle ID is implemented in a DaVinci "Particle Maker" that takes tracks (protoparticles) and assigns a particle ID to them to form particles which will be used in the physics analysis. The particle maker is called "CombinedParticleMaker", and does the job previously done by a number of different particle makers (ElectronParticleMaker, RichParticleMaker etc). Reasonable values are set by default for the particle ID selection cuts, but they can be modified using options, according the efficiency/purity requirements of your analysis. The particle types to be selected are defined by a "Particles" option, with the names of the 5 common charged particle types as arguments. The default is

```
Particles = { "muon", "electron", "kaon", "proton", "pion" }
```

If you wish only to select kaons and pions then you would put:

```
Particles = { "kaon", "pion" }
```

The selection can be exclusive, or not, as set by an ExclusiveSelection option that is set either true or false. If true, as is the case by default, then the selection criteria are applied in the order that the particle types were listed in the Particles option, and if the selection is satisfied for one particle type the track is not considered for subsequent particle types. So for the default Particles option given above, muons are first searched for, then electrons, and so on. In this way each track will have at most one selected particle type. Conversely, if ExclusiveSelection is set false then the selection criteria are applied independently for the different particle types. Tracks can then be selected as more than one particle type, and care should be taken to avoid double

counting in the physics algorithm.

The selections themselves have the form:

```
ElectronSelection = { "det='CALO' e-pi='0.0' }
```

This defines that for selecting electrons, the Calorimeter detector information should be available, and cut is applied on $DLL(e\text{-}\pi) > 0$. To require that both Calorimeter and RICH are available, one would put `det='CALO,RICH'`. `e-pi` is a symbol that has been defined to represent the lower limit on the difference in log-likelihood between electron and pion hypotheses. All other DLLs have similarly defined symbols, constructed using `e,mu,pi,k,p` to represent the particle types. More than one cut can be applied if desired. For example, to select kaons one may wish to reject pions (and hence cut on $DLL(k\text{-}\pi)$) but also reject protons (and hence also cut on $DLL(k\text{-}p)$). This is done in the default selection as follows:

```
KaonSelection = { "det='RICH' k-pi='2.0' k-p='-2.0' }
```

If your algorithm is interested in selecting kaons and doesn't suffer from proton background you could remove the cut on $DLL(k\text{-}p)$: this would emulate what is currently done by the `RichParticleMaker`, identifying as kaons any "heavy" particle (kaon or proton). If you are really keen you can also apply different DLL cuts according to which detectors are available. For example:

```
KaonSelection = { "det='RICH,CALO' k-pi='0.0'", "det='RICH' k-pi='2.0' }
```

would apply a looser cut when both RICH and Calorimeter are available, and a tighter cut when only the RICH is available. The criteria are applied in the order they are listed.

In the default selection the pion criteria are left empty, so that all tracks that have not been selected as other particle types are called pions. Obviously this relies on the fact that `ExclusiveSelection` is true, and the pion hypothesis is listed last in the `Particles` option. As an example, the default DaVinci options for the combined particle ID are given below. ("`PreLoadParticles`" should be replaced by the name of your algorithm). The values chosen should be understandable once you have taken a look at the distributions shown in the presentation mentioned above. An example of how to control an algorithm using the `CombinedParticleMaker` is provided with DaVinci v8r1 in:

```
$DAVINCIROOT/options/PreLoadCombinedParticles.opts
```

Note that `DaVinci.opts` provides the possibility of executing such an example algorithm.

```
ParticleMakerType = "CombinedParticleMaker"; CombinedParticleMaker.Particles = { "muon", "electron",
"kaon", "proton", "pion" }; ExclusiveSelection = true; MuonSelection = { "det='MUON' mu-pi='-8.0' };
ElectronSelection = { "det='CALO' e-pi='0.0' }; KaonSelection = { "det='RICH' k-pi='2.0' k-p='-2.0' };
ProtonSelection = { "det='RICH' p-pi='3.0' };
```

Additional Information

Here is some further information to help in the use of the Combined particle ID. In your physics analysis you can get access to the particle type that has been chosen for a given particle using:

```
(*iPart)->particleID().pid()
```

You can get access to the Delta log-likelihood values mentioned in my previous message, by first finding the protoparticle that was used to create the particle ("`pporig`") and then:

```
de = pporig->detPIDvalue( ProtoParticle::LkhPIDe )
```

This is $DLL(e-\pi)$, the difference in log-likelihood between electron and pion hypotheses, and similar expressions give you the other three values $DLL(\mu-\pi)$, $DLL(K-\pi)$ and $DLL(p-\pi)$.

The detector information that was available for use in calculating the DLL values can be accessed as follows:

```
pporig->caloeBit()
```

```
pporig->muonBit()
```

```
pporig->richBit()
```

which are boolean values indicating whether the CALO/MUON/RICH info was available respectively.

Finally there is a "Confidence Level" that has been calculated from the DLL values, to indicate the confidence that the chosen assignment of particle ID is correct. It has a value between 0 and 1 and is accessed as follows: `cl = (*iPart)->confLevel()` It is calculated as the ratio of the likelihood of the chosen hypothesis to the sum of the likelihoods of all hypotheses. So for example, if the chosen particle type is the electron hypothesis, $cl = \exp(de) / (1 + \exp(de) + \exp(dm) + \exp(dk) + \exp(dp))$

All of this is illustrated in a simple UserAlgorithm that I have set up with the help of Gloria. You can find it in directory:

```
/afs/cern.ch/user/r/rforty/newmycmt/Phys/DaVinciUser/v4r1/src
```

It loops over particles, makes the connection to their protoparticles and fills an ntuple with the variables mentioned above. It also finds the true particle type from the Monte Carlo truth. I encourage you to add something similar to your physics analysis, so you can check the source of your background, and adjust the particle ID selection cuts in the CombinedParticleMaker accordingly.

Best regards, Roger

P.S. One could imagine an approach of setting loose selection cuts (on the DLL values) in the CombinedParticleMaker and then cutting on the confidence level variable in your analysis to make the final selection. That would certainly be a simpler arrangement to explain. However, at the moment I find a slightly better performance using the DLL cuts. Perhaps the definition of the confidence level variable can be improved: I offer a bottle of champagne for the best idea that anyone comes up with to improve it.

ChrisRJones - 20 Nov 2006

This topic: LHCb > RogerFortyGPIDEmails

Topic revision: r1 - 2006-11-20 - unknown



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback