Subject: Bump splitting: new endcap vs. old endcap
Posted by Bertram Kopf on Wed, 04 Feb 2009 18:21:49 GMT
View Forum Message <> Reply to Message

Dear all,
I tested the bump splitting for a couple of different scenarios. 5 GeV/c pi0's have been simulated separately for the forward endcap (10-20 deg) and for the forward barrel part (30-90 deg) for two different geometries. I used in the simulation macro the initialization
Emc->SetGeometryFileName("emc_module12345.dat");
for the old geometry;
and Emc->SetGeometryFileNameDouble("emc_module1245.dat","emc_module3new.root ");
for the new geometry and changed moreover the mapper version number to 1 (old) and to 2 (new) in all.par and emc.par.

I got the following results with the bump analysis qa macro provided by Dima:

1. Everything looks more or less o.k. for the barrel part (30-90 deg) (black=old, red=new)

2.In comparison to this the results for the forward part (10-20 deg) look completely different (black=old, red=new) and it seems that there is only something wrong with the new geometry.

My assumption is that something is not properly initialized in the new EMC geometry setup (e.g. EmcMapper, TwoCoordinateIndex, etc.) and that the bump splitting algorithm works fine.

Due to this I have a few questions / remarks:

1. Is there a tool available with which one can easily check whether the crystal neighbour list, TwoCoordinateIndex, etc. are o.k.?

2. All the geometry information will be handled via the singleton objects "PndEmcMapper" and "PndEmcStructure".
The static function instance has to be called with arguments here: e.g. static PndEmcMapper* Instance(Int_t, TString geoName="");
This suggests that one can get different mappers by steering it via these arguments. But this is not the case: Once initialized, the instance will never be changed afterwards.
In the code the instance of the mapper is sometimes called with fixed arguments: e.g. PndEmcMapper::Instance(0), etc.. Therefore my question: Who takes care of the initialization for such global objects. How is it guaranteed that the instance will be called there the first time in the application?

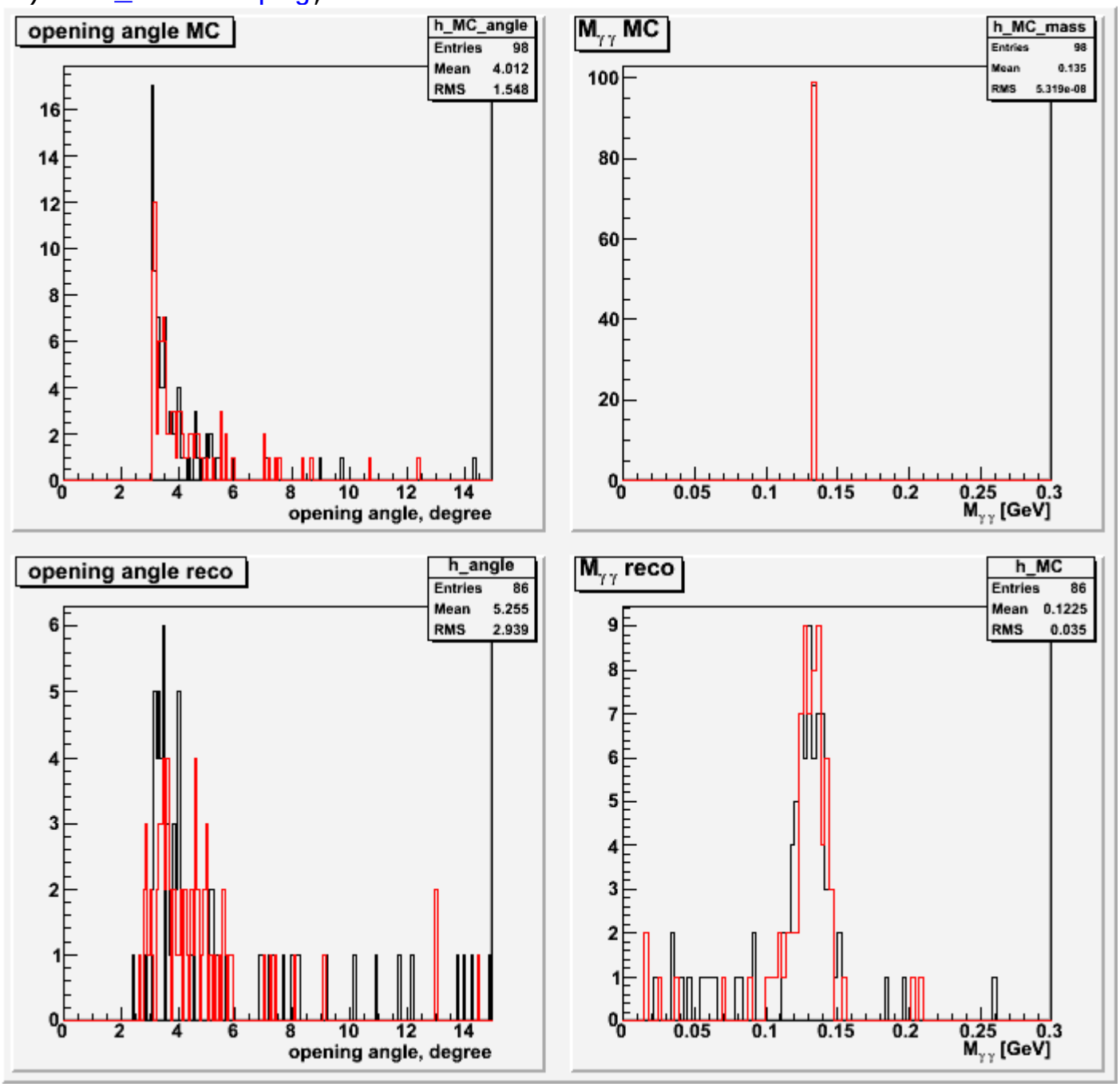
3. Would it make sense to introduce an EMC initialization sequence where everything will be initialized centrally at one place.

4. Does it make sense to introduce a global Emc environment class where all global emc environment objects are collected?

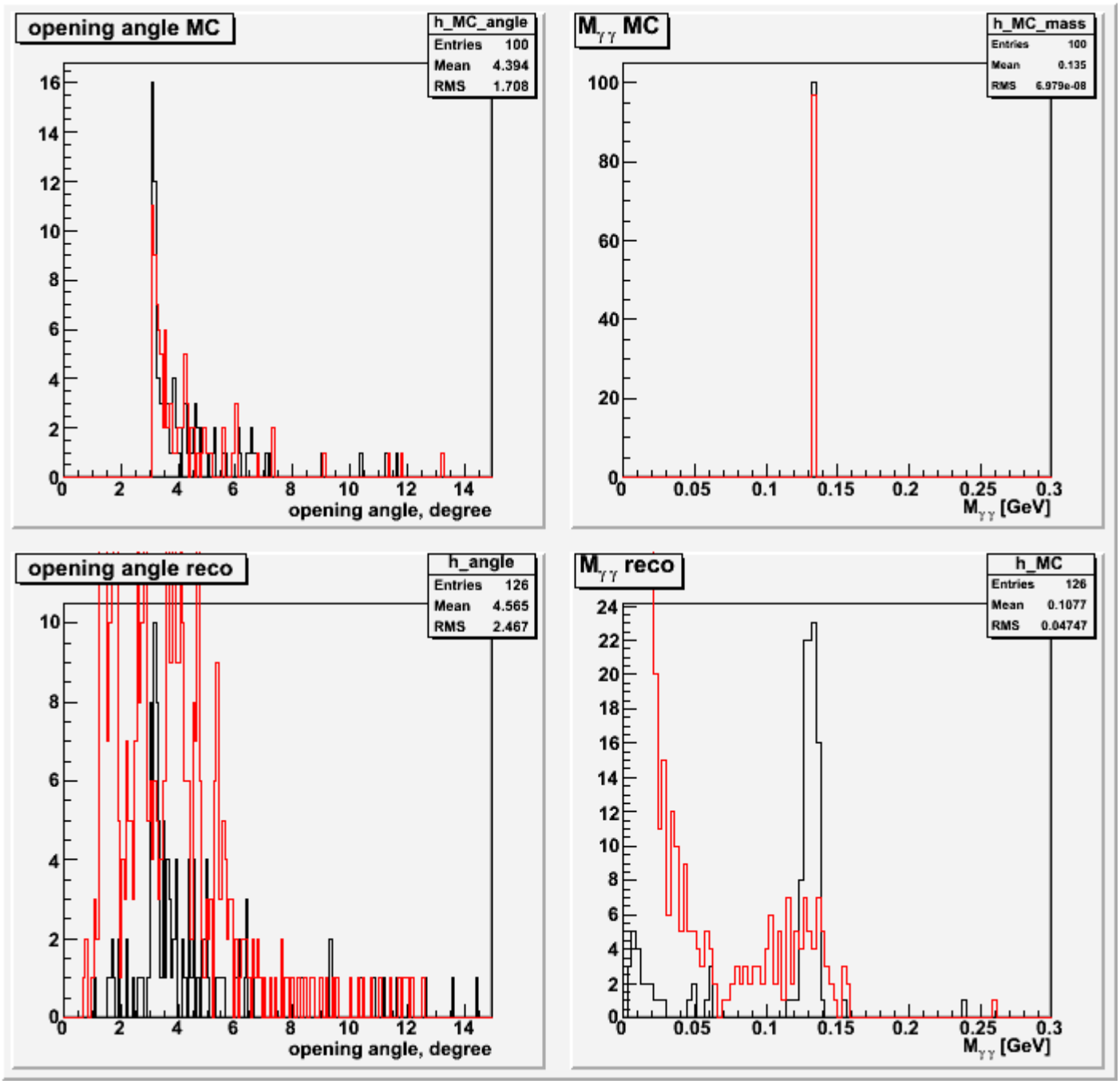
Best regards,
Bertram.

File Attachments

1) emc_barrel.png, downloaded 1716 times



2) emc_endcap.png, downloaded 1759 times



Subject: Re: Bump splitting: new endcap vs. old endcap
Posted by StefanoSpataro on Mon, 09 Feb 2009 12:08:54 GMT
View Forum Message <> Reply to Message

Hi,
only today I can answer to this post.

Quote:I got the following results with the bump analysis qa macro provided by Dima:

First of all, which qa macro have you used to produce those plots? So that I can also try to reproduce the distributions.

Quote:My assumption is that something is not properly initialized in the new EMC geometry

setup (e.g. EmcMapper, TwoCoordinateIndex, etc.) and that the bump splitting algorithm works fine.

When the new geometry for the endcap was put inside svn, Ola has showed plots with cluster residuals for energy and angular distributions, and they looked fine. For this reason I think that the mapping should be ok, considering that the clusterization depends on the map. Maybe there is something just after the cluster and before the bump, one should investigate.

Quote:1. Is there a tool available with which one can easily check whether the crystal neighbour list, TwoCoordinateIndex, etc. are o.k.?

The EMC reconstruction code was substantially ported from the fast simulation framework, thus you could use all your tools to check the mapping in pandaroot, the classes are pratically the same.
Which tools have you developed, so that we can also implement them in pandaroot? In general one can use the GetXPad()/GetYPad() functions and correlate them to the x/y/z. But a real "tool" was never developed by us.

Quote:2. All the geometry information will be handled via the singleton objects...
[cut]

Could you please show an example on how the structure could be messed up? In theory the mapper is needed only for the reconstruction tasks, when looping inside the data after the reconstruction the mapper could only screw the TwoCoordinateIndex objects, but at this stage they are not needed anymore. So maybe I have not well understood which could be the possible problems. And if I remember well, the Instance(0) are called only when no previous mapping is defined and then the code does not know what to use. Maybe Dima can comment on it.

Quote:3. Would it make sense to introduce an EMC initialization sequence where everything will be initialized centrally at one place.

In theory the function static PndEmcMapper* Instance(Int_t, TString geoName=""); is done with this purpose, to setup the environment only one time. Or am I missing something?

Quote:4. Does it make sense to introduce a global Emc environment class where all global emc environment objects are collected?

I have still not understood well what do you call "environment objects". In theory there is just the map and all the correlated classes. I know the structure of the code is a bit complicated, but we have prefered to just port the code and to not modify it a lot.

Which are exaclty your suggestions?

Subject: Re: Bump splitting: new endcap vs. old endcap
Posted by Dima Melnychuk on Mon, 09 Feb 2009 12:34:22 GMT
View Forum Message <> Reply to Message

Hi Stefano,

Actually the emc classes PndEmcMapper and PndEmcStructure
were not ported from Babar framework, but created from scratch because of different representation of geometry. In Babar framework the geometry was based on XML description and these two classes provide similar functionality to what exists in Babar framework but they are not identical.

The macro, which shows the plots on bump splitting is bump_analysis.C.

Concerning the PndEmcMapper::Instance(0) with parameter 0, it does not create the instance of PndEmcMapper, but only get the instance of the PndEmcMapper if it was created previously. Otherwise it gives the warning.

And at the moment PndEmcMapper is instantiated with proper input parameter in the Init() methods of the tasks where it is used, for example PndEmcMakeCluster, PndEmcMakeBump.

But I support the idea of Bertram for emc initialization sequence, let's say the task class PndEmcInit, which would be inside digitization/reconstruction sequence before any other tasks and where PndEmcMapper would be initialised only once.
But it's matter of discussion.

Dima

Subject: Re: Bump splitting: new endcap vs. old endcap
Posted by Aleksandra Biegun on Thu, 19 Feb 2009 14:43:29 GMT
View Forum Message <> Reply to Message

Hi,

../trunk/emc/EmcMC/PndEmc.cxx

and

../trunk/emc/EmcTools/PndEmcStructure.cxx

are posted to svn /pandaroot/trunk/emc/ .

The part corresponding the creation of detectorID
for the forward endcap is changed in both cases.

Best regards,
Ola.

Subject: Re: Bump splitting: new endcap vs. old endcap
Posted by Dima Melnychuk on Fri, 20 Feb 2009 15:32:04 GMT
View Forum Message <> Reply to Message

Hi all,

With the new changes, made by Ola in creation of emc forward endcap geometry, the bump splitting gives reasonable results.

For the 500 events 5 GeV pi0 generated in theta [10;20] degree an phi in range [0;360] degree.

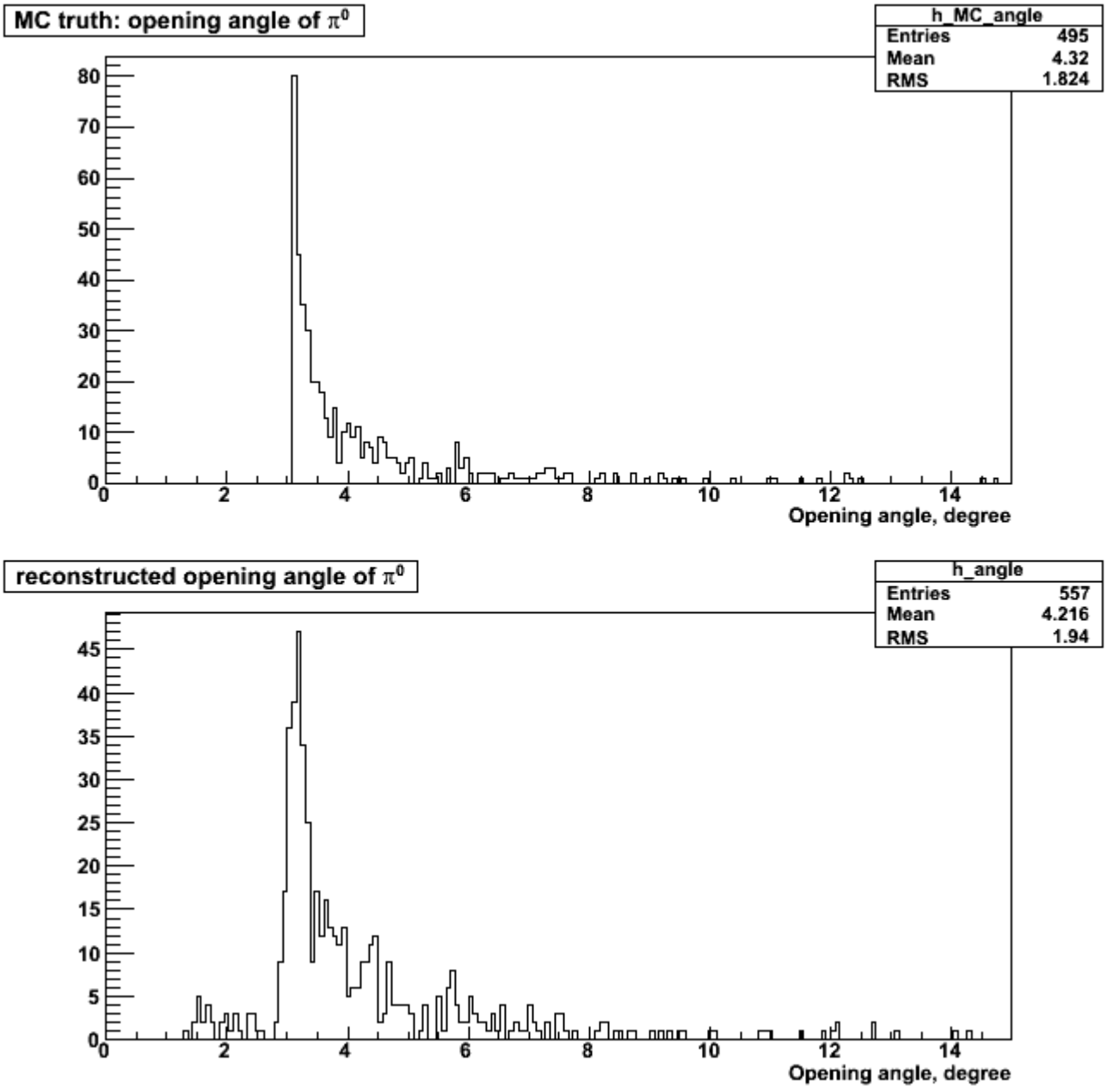
pi0 opening angle:

and reconstructed invariant mass:

Dima

File Attachments

1) pi0_angle.png, downloaded 1424 times



2) pi0_mass.png, downloaded 1356 times

