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?

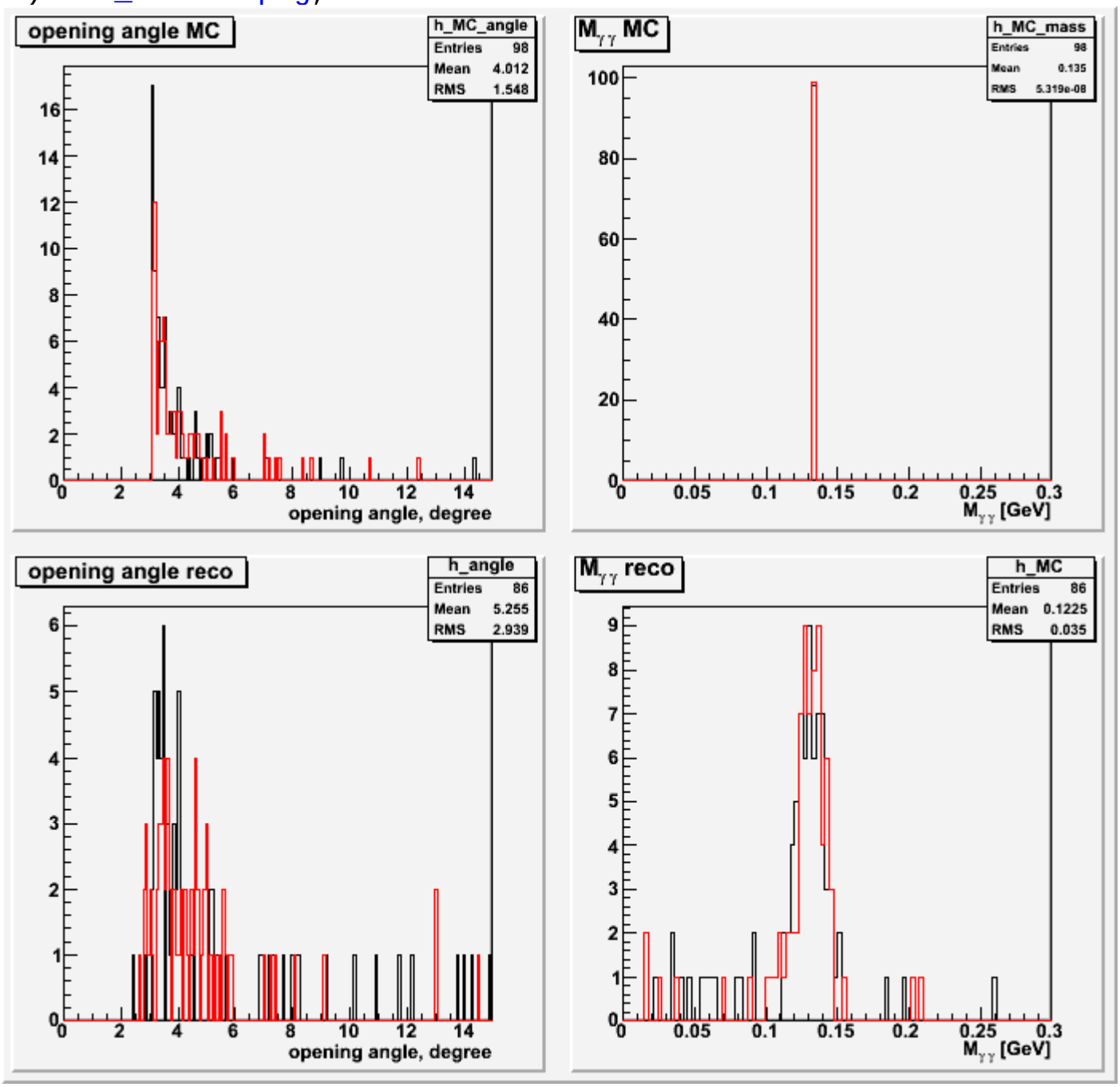
3. Would it make sense to introduce an EMC initialization sequence where everything will be initialized centrally at one place.

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

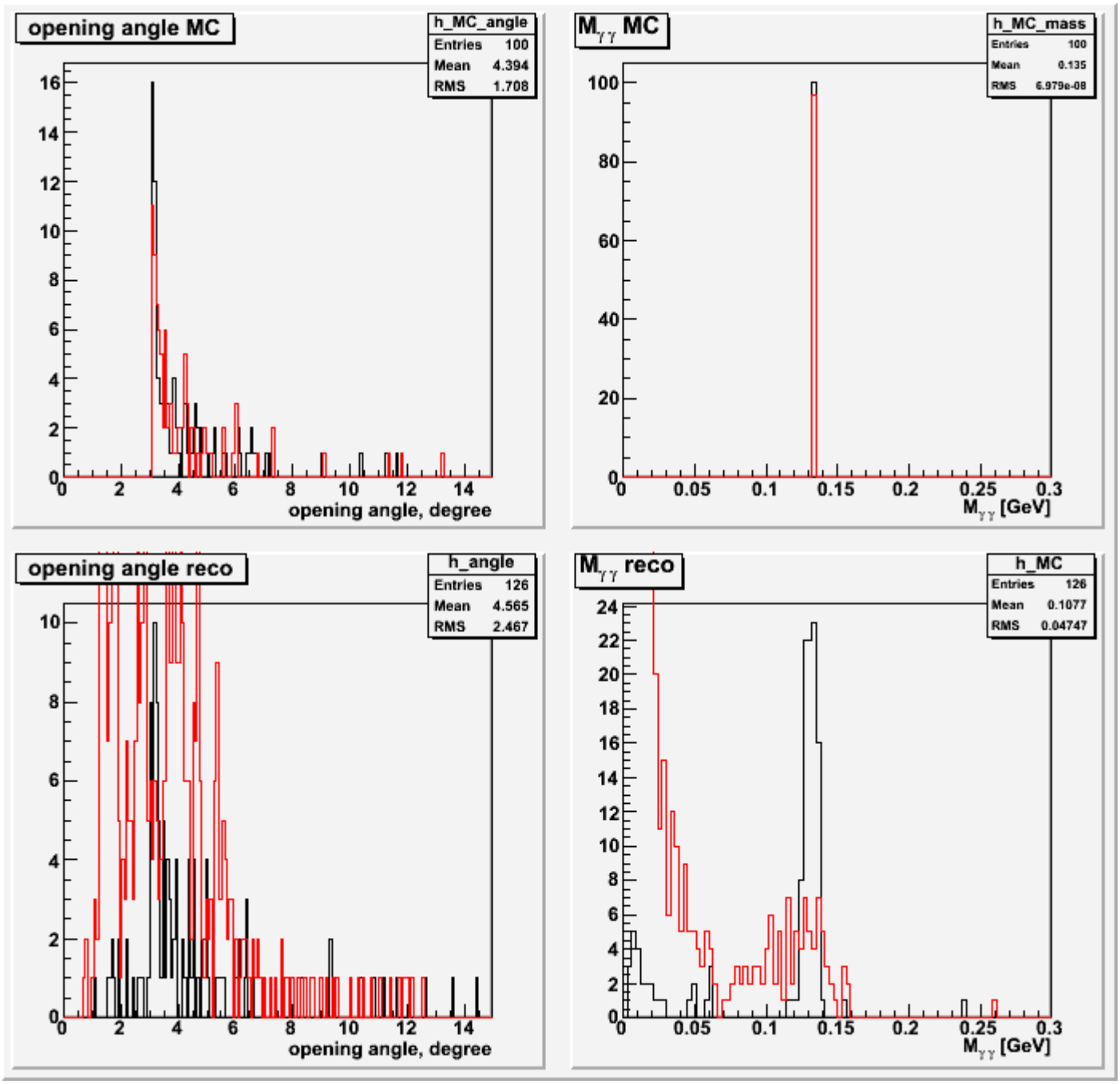
Best regards,
Bertram.

File Attachments

1) emc_barrel.png, downloaded 1717 times



2) emc_endcap.png, downloaded 1760 times


opening angle MC
h_MC_angle
Entries 100
Mean 4.394
RMS 1.708
opening angle, degree
$M_{\gamma\gamma}$ MC
h_MC_mass
Entries 100
Mean 0.135
RMS 6.979e-08
$M_{\gamma\gamma}$ [GeV]
opening angle reco
h_angle
Entries 126
Mean 4.565
RMS 2.467
opening angle, degree
$M_{\gamma\gamma}$ reco
h_MC
Entries 126
Mean 0.1077
RMS 0.04747
$M_{\gamma\gamma}$ [GeV]