Subject: GEM 'realistic' track finding
Posted by Radoslaw Karabowicz on Thu, 30 Jul 2009 13:38:42 GMT
View Forum Message <> Reply to Message

Dear pandas,

With this message I would like to announce a working first version of the GEM track finding. It is waiting in the svn/pandaroot/trunk for You to test it.

Below in few words I will try to explain the main idea behind and the implementation.

The class to do the track finding job is PndGemTrackFinderOnHits. As the long name suggests, it uses GEM reco hits to find the tracks. It does it in several steps:

1. As explained in my presentation from Feb 2009 Panda Collaboration Meeting each track crossing GEM station will essentially fire 4 strips in 4 different views (currently: radial, concentric, tilted to left, tilted to right). Geometrically, these fired strips may be grouped in two pairs: radial and concentric strips are fired about 0.6 cm in front of the nominal station position in "z", and the tilted strips are fired about 0.6 cm after the nominal "z_stat". Therefore per one MC track, firing 4 strips, there are two reconstructed hits separated by about 1.2 cm in longitudinal direction.
The first task of the PndGemTrackFinder is to match the two hits on the front and back layers of the station.

2. Combine the matched pairs of hits from each pair of stations into track segments in some reasonable window, where rather energetic (lets say with momentum larger than 0.5 GeV/c) particles emitted at vertex are expected. For these pairs calculate momentum vertex at the vertex.

3. Combine the track segments to form tracks. The track segments are matched, if they share matched hits on one station, and if the momentum is reasonably similar, 30% difference in momentum is allowed.

4. Remove bad tracks. By this I mean:
-too short tracks
-tracks that had too many hits used also by different reconstructed tracks
-the tracks that were build from segments which had momenta too far away from the mean momenta of segments

5. I have also created the PndGemTrackFinderQA task, which, attached after any PndGemTrackFinder should give information about track finding efficiencies, number of ghosts (tracks that could not be matched with any MC) and clones (tracks that seem very similar, both by the hits they were created from and the momentum). For example, when attached after PndGemTrackFinderIdeal task, it produces the following table:

```
------------------- PndGemTrackFinderQA : Summary ------------------
 Events:             1000
 MC Tracks:          1090
 reconstruable:      1018 reconstructed:      1018 >>>> 100%
 primaries   :        984 reconstructed:       984 >>>> 100%
 reference   :        984 reconstructed:       984 >>>> 100%
 secondaries :         34 reconstructed:        34 >>>> 100%
 ghosts      :         69 clones:                0
```

Primaries are tracks with a distance of their start position to interaction vertex smaller than 1 cm, secondaries are all other, and reference are primaries with momentum larger than 0.5GeV/c. As expected, recostruction efficiency is 100%, the only problem are the ghosts, will look later why they appear here at all.

6. When attaching the PndGemTrackFinderQA task after the newly created PndGemTrackFinderOnHits, it gives:
- when one particle per event was generated:
```
------------------- PndGemTrackFinderQA : Summary ----------------
 Events:            10000
 MC Tracks:         10821
 reconstruable:     10154 reconstructed:      9912 >>>> 97.6167%
 primaries    :     9803 reconstructed:      9771 >>>> 99.6736%
 reference    :     9803 reconstructed:      9771 >>>> 99.6736%
 secondaries  :      351 reconstructed:       141 >>>> 40.1709%
 ghosts       :       59 clones:              82
-------------------------------------------------------------
```
- when 5 particles per event were generated:
```
------------------- PndGemTrackFinderQA : Summary ----------------
 Events:            2000
 MC Tracks:         10879
 reconstruable:     10161 reconstructed:      9542 >>>> 93.9081%
 primaries    :     9772 reconstructed:      9399 >>>> 96.183%
 reference    :     9772 reconstructed:      9399 >>>> 96.183%
 secondaries  :      389 reconstructed:       143 >>>> 36.7609%
 ghosts       :      605 clones:             389
```
--- - when ten particles per event were generated:
```
------------------- PndGemTrackFinderQA : Summary ----------------
 Events:            1000
 MC Tracks:         10803
 reconstruable:     10147 reconstructed:      8872 >>>> 87.4347%
 primaries    :     9797 reconstructed:      8765 >>>> 89.4662%
 reference    :     9797 reconstructed:      8765 >>>> 89.4662%
 secondaries  :      350 reconstructed:       107 >>>> 30.5714%
 ghosts       :     1464 clones:             767
-------------------------------------------------------------
```

As expected, the track finding efficiency worsens rather quickly when more particles per event were produced, but:
- it is a first version
- it uses information only from the GEM detector
- it probably have many bugs, so please use the PndGemTrackFinderOnHits, and tell me about your feelings about it

7. I have also created a macro to draw some histograms created by the PndGemTrackFinderQA task. The macro is macro/gem/draw_qa_track_finding.C. It takes as argument the reconstructed file name.
The macro produces one canvas with several plots on it, which looks like this one:

and prints in the terminal a small table, which is identical with the one produced by

PndGemTrackFinderQA task itself.

Sorry for a rather lengthy message, if you have any questions or comments, please do not hesitate to reply to this message.

Thank you for attention,

cheers,
radek

File Attachments

1) GemTrackFinderQACanvas.gif, downloaded 1281 times

Gem Track Finde

