MAUS - Feature #1389

Global PID framework

17 December 2013 12:07 - Pidcott, Celeste

Status:	Open	**Start date:**	17 December 2013
Priority:	Normal	**Due date:**	01 January 2014
Assignee:	Pidcott, Celeste	**% Done:**	0%
Category:	Global Reconstruction	**Estimated time:**	0.00 hour
Target version:	Future MAUS release		
Workflow:	Under Review		

Description

Inclusion of Global PID framework. Code has been prepared and is ready for testing.

History

#1 - 08 January 2014 11:02 - Pidcott, Celeste

- Workflow changed from Under Test to Under Review

Code in branch lp:~c-e-pidcott/maus/1389 now ready for code review.
Files to be reviewed:
All under directories src/map/MapCppGlobalPID , src/map/MapCppGlobalReconImport , src/reduce/ReduceCppGlobalPID
Within src/common_cpp/Recon/Global : ImportTOFRecon, TrackMatching, PIDBase, PIDVarA.

The python scripts that run the mappers and reducers above are GlobalPID.py, GlobalReconImport.py and pid_pdf_generator.py, all contained within bin/Global

#2 - 10 January 2014 16:42 - Lane, Peter

Yesterday I committed (lp:maus/merge), among other things, a MapCppGlobalRawTracks mapper that creates raw (pre-global reconstruction) Tracks from detector reconstruction data and adds them to the Recon/Global section of the data structure. This is probably what you want to use/modify instead of your ImportTOFRecon. If you end up playing with it, let me know if you need help or have issues with it.

A couple of notes:

1) Currently the mapper uses the reference particle to set the PID (Track::set_pid()) for the tracks, so that will obviously have to change. I'm only concerned with single muon tracks at the moment, so this was convenient for me.

2) Until the detector groups provide measurement errors themselves, the configuration needs to be customized. It's probably not something you are concerned with, but it needs to be there since the mapper expects it so it can set the space point uncertainties. You'll have to add a section as follows to your configuration:

```
global_recon_detector_attributes = [
  {
    "id":2,  # MAUS::DataStructure::Global::kTOF0
    "uncertainties":[
      [4.0, 0.0, 0.0, 0.0, 0.0, 0.0],
      [0.0, 1.44e7, 0.0, 0.0, 0.0, 0.0],
      [0.0, 0.0, 133.3, 0.0, 0.0, 0.0],
      [0.0, 0.0, 0.0, 1.0e7, 0.0, 0.0],
      [0.0, 0.0, 0.0, 0.0, 133.3, 0.0],
      [0.0, 0.0, 0.0, 0.0, 0.0, 1.0e7]
    ]
  },
  {
    "id":6,  # MAUS::DataStructure::Global::kTOF1
    "uncertainties":[
      [4.0, 0.0, 0.0, 0.0, 0.0, 0.0],
      [0.0, 1.44e7, 0.0, 0.0, 0.0, 0.0],
      [0.0, 0.0, 300.0, 0.0, 0.0, 0.0],
      [0.0, 0.0, 0.0, 1.0e7, 0.0, 0.0],
      [0.0, 0.0, 0.0, 0.0, 300.0, 0.0],
      [0.0, 0.0, 0.0, 0.0, 0.0, 1.0e7]
    ]
  },
  ...
}
```

#3 - 06 February 2014 10:24 - Pidcott, Celeste

PID framework that is inline with current version of lp:maus/merge is now in branch lp:~c-e-pidcott/maus/1389a, and has been successfully built on Jenkins

#4 - 14 February 2014 07:40 - Rajaram, Durga

Hi Celeste,

A few things

1. The file PIDhists.root is currently in the MAUS_ROOT directory. This should go into its own appropriate area -- probably inside the map/MapCppGlobalPID/ directory -- just so we don't clutter the root directory.
2. line 53 of src/map/MapCppGlobalPID/MapCppGlobalPID.cc -- the file name is hard-coded, might be better to have this as a configuration flag, especially if you anticipate the possibility of running with different different inputs
3. lines 45,46 of reduce/ReduceCppGlobalPID/ReduceCppGlobalPID.cc look for

   ```
   global_pid_hypothesis, unique_identifier
   ```

   flags in ConfigurationDefaults.py. However there are no defaults for these flags in src/common_py/ConfigurationDefaults.py, so the reducer as-is will never run -- should add a default and some comment on the meaning of the flag and what possible values it may take
   1. also add documentation to the user guide
4. in bin/Global/ -- can you add some documentation on how to run the examples? i.e. run simulate_mice_json.py with a datacard for eg datacard_200MeV_mu_plus.py --> run global recon --> run global PID mapper/reducer.
   1. also, can't all these steps be chained in one file? i.e. add the GlobalReconImport and GlobalPID mappers to the simulate_ script?

#5 - 17 February 2014 16:49 - Pidcott, Celeste

Starting with point 4, I do have some documentation that I have written for the PID that I forgot to include when I did my push, so that is there now. I currently perform the simulation without my mappers just to make the simulation take less time, and so that I have the original json output to re-use should my mappers change, that way I don't have to re-run the simulation each time. As for keeping GlobalReconImport as a separate step, my reason for doing this is just because once I've incorporated in Peter's work on the Global Reconstruction, GlobalReconImport will then no longer be needed, so I thought it best to keep it separate from GlobalPID.

Onto point 3, I've done as you said and added the flags to ConfigurationDefaults.py, with the default for the unique_identifier ~~set to the current date and time~~ set as an empty string, because setting it to the current date and time, which is what I suggest doing in the accompanying comment in ConfigurationDefaults, would mean importing the datetime module and I wasn't sure if that was the kind of thing you'd want adding in to the defaults, so hopefully the comment will be sufficient guidance for people using the flag, and I've left global_pid_hypothesis as an empty string, because, as it says in the documentation that I so helpfully didn't upload before, it should reflect the MC being run at the time, and so if someone hasn't defined what particle hypothesis they are producing PDFs for in their datacard, I want the reducer to not run for them and instead tell them that they are doing something wrong. Again, apologies for the previously missing documentation, it would have been far more useful for it to already be there.

And points 1 and 2 were fine, I've now made those changes.