LHCf: new analysis library

Alessio

LHCf collaboration meeting

Firenze, 26-27/11/2018

Overview

- Motivation: use the same analysis software within the collaboration
  - easier comparison of results from different people
  - the latest version is immediately available to everyone
- Now hosted on the usual SVN repository (possibility to use a GIT repository in the future)
- The software must have the maximum portability to be able to run on both old (like PC clusters) and new (like personal PC/laptops) operating systems

Main characteristics

- Using CMake to improve portability (automatically generate Makefile depending on system)
- Compatible with both ROOT 5 and ROOT 6
- No other external libraries needed
- Not using C++11 features (compatibility with older compilers)
- Using the LHCfEvent class (from Menjo-san software) for input/output of data
- Using template classes (from Mitsuka-san software)

Template classes

- Each class inherit constant parameters from either Arm1Params or Arm2Params classes (base classes)

Base class definition

```
class Arm2Params
{
protected:

  /* Calorimeter constants */
  static const Int_t cal_ntower_   = 2;
  static const Int_t cal_nlayer_   = 16;
  static const Int_t cal_nrange_   = 2;

  /* Position detector constants */
  static const Int_t pos_ntower_   = 1;
  static const Int_t pos_nlayer_   = 4;
  static const Int_t pos_nview_    = 2;
  static const Int_t pos_nchannel_ = 384;
  static const Int_t pos_nsample_  = 3;
...
};
```

Class definition

```
template<typename armclass>
class Level2Cal: public armclass
{
public:
  Level2Cal();
  ~Level2Cal();
…
};
```

Class instantiation

```
...
Level2<Arm2Params> level2;
...
```

Status

- Reconstruction part (position, energy, PID…) already coded and roughly tested with Arm2 data→ need to test more carefully and with Arm1 data
- Need to implement gain calibration (ADC → GeV)
- This week: discuss with Menjo-san how to share the work