

Table of Contents

Proton Taggers	1
What are the proton taggers?.....	1
Implementation in Fast Simulation.....	1
How to use.....	1
Known issues.....	2
Review Status.....	2

Proton Taggers

Complete:

What are the proton taggers?

Proton taggers is a short name for so called near-beam detectors, special tracker devices installed on the beam pipe hundreds metres away from the interaction point and aimed to detect protons which lost a small fraction of their energy or were scattered at small angles (forward protons). This package provides a method to simulate the acceptance of the proton taggers to be installed 420 and 220 m from IP5 (in both beam directions).

Implementation in Fast Simulation

Currently the only module defined in ProtonTaggers is ProtonTaggerFilter. The module is implemented in a form of EDFilter. Basically it can be viewed as a trigger code, which only passes through those events which have some forward protons in the taggers. As an input it accepts Monte Carlo events in the form of HepMCProduct with labels "source" and "famosPileUp". The only output is a boolean value showing whether the event has been accepted. Conditions upon which an event is accepted are steered from .cfg file. You can choose between detector locations (220 and 420 metres or both), ask for 1 or 2 detector arms to be hit, or even ignore one of the arms (e.g. to study acceptance for the clockwise beam only). Apart from a trigger bit the module does not add anything to the event.

The simulation procedure basically follows the procedure from ORCA-based FAMOS (module FastTotem). It does a simple table-based calculation of the probability of detecting a forward proton (given its momentum vector) with a Roman Pot type detector installed in one of the four pre-defined locations (220 or 420m, two beam directions). Note that old FAMOS had a method to simulate the measurement of proton fractional momentum loss. However the detector resolution simulated in FAMOS does not reflect the recent improvements in the near-beam detector design, and therefore has not been ported to CMSSW. In principle, proton taggers also allow for measurement of the t kinematic variable, connected to proton scattering angle, however this measurement has never been implemented in FAMOS. Note also that no dependence on vertex position is simulated. Thus, obviously, there is no reason to apply vertex smearing before running ProtonTaggers.

Starting from version V01-00-03 this package can also process pileup protons simulated with PileUpProducer fast simulation module. Both "signal" and pileup protons are treated on a common basis and equally. For example, a combination of a signal proton in one arm and a pileup proton in the other one would pass the "2 protons" trigger. In case of two or more protons in one arm a message is printed to standard output, but no effect on trigger bit is generated compared to one single proton in the same arm.

A full simulation procedure for the FP420 detectors, based on Geant4 and the Hector proton transport code, shall be available as a separate package(s).

How to use

To use the package you need to include in your python script a piece of text like the following.

```
process.forwardProtonFilter = cms.EDFilter(  
    "ProtonTaggerFilter",  
    # ... choose where you want a proton to be detected for beam 1 (clockwise)  
    #     0 -> ignore this beam  
    #     1 -> only 420 (FP420)  
    #     2 -> only 220 (TOTEM)
```

```
#      3 -> 220 and 420 (region of overlay)
#      4 -> 220 or 420 (combined acceptance)
beam1mode = cms.uint32(1),

# ... and for beam 2 (anti-clockwise)
beam2mode = cms.uint32(1),

# ... choose how the information for the two beam directions should be combined
#      1 -> any of the two protons (clockwise or anti-clockwise) is enough
#      2 -> both protons should be tagged
#      3 -> two protons should be tagged as 220+220 or 420+420 (makes sense with beamXmode=4)
#      4 -> two protons should be tagged as 220+420 or 420+220 (makes sense with beamXmode=4)
beamCombiningMode = cms.uint32(2)
)
```

... and add `forwardProtonFilter` in your path. In this example only events with two protons tagged by the two arms of FP420 will survive after the filter. The outcome of the filter, true or false, will go to `TriggerResults` in the Event stream. The `SelectEvents` parameter of `PoolOutputModule` can then be used to select the "triggerred" events to be written on disk. See also example file

`FastSimulation/ProtonTaggers/test/testProtonTaggers_cfg.py`.

The `ProtonTaggers` package has been included in CMSSW releases starting from 1_7_0 (for 1_6_X: do `cvs co -r V01-00-04 FastSimulation/ProtonTaggers; cd FastSimulation/ProtonTaggers; scarmv1 b`).

Starting from CMSSW 3_0_X the package is moved to `plugins` directory in `FastSimulation/ForwardDetectors`. Example script for 3_1_X can be found in `FastSimulation/ForwardDetectors/test/testProtonTaggers_cfg.py`. For early prereleases of 3_1_X (<= pre6) use `cvs co -r V01-00-06 FastSimulation/ForwardDetectors`.

Known issues

Like other FastSim packages, this package is aimed to be a fast and simple tool for quick physics studies. However it has limitations:

1. The acceptance tables currently in use are quite old and may not provide a very high precision simulation.
2. The module accepts data, signal and pileup, only in the form of `HepMCProduct`. It does not read pileup events simulated as `SimTracks/!SimVertex`'s.

Review Status

PatrickJanot - 12 Sep 2007	Template page
DmitryZaborov - 12 Sep 2007	First documentation for ProtonTaggers
DmitryZaborov - 05 Oct 2007	Add usage example
DmitryZaborov - 28 Oct 2007	Add "Known issues" section, update example
DmitryZaborov - 29 Oct 2007	Revise instructions for use
DmitryZaborov - 4 Nov 2007	Describe the new capability of pileup treatment
DmitryZaborov - 24 Apr 2009	Update of "how to use" section
DmitryZaborov - 28 Apr 2009	Put a notice on CMSSW 3_X series

Responsible: AndreaGiammanco

Last reviewed by: DmitryZaborov

This topic: CMSPublic > SWGuideFastSimProtonTaggers

Topic revision: r10 - 2011-12-13 - AndreaGiammanco



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback