Table of Contents

Running Digitization witht the new MuCTPI simulation

Setting up the digitization

Unlike the simulation, the digitization needs the checkout of some additional packages. I've developed and tested the new MuCTPI simulation code with release 10.5.0, but I don't see why it shouldn't work with any release after 10.0.0.

For the following I assume, that the reader knows how to set up and use CMT, and how to compile packages with it.

To get the corect flags for barrel phi overlaps for layout Q, **MuonSpectrometer/MuonCablings/RPCcabling** should be checked out and compiled with tag **RPCcabling-00-03-32**.

The most recent MuCTPI simulation code (**Trigger/TrigT1/TrigT1Muctpi**) has the tag **TrigT1Muctpi-00-01-02**. This should be checked out and compiled as well.

The default MuCTPI job options should be edited. (Found in ~/share/TrigT1Muctpi_jobOptions.py.) The main algorithm of the MuCTPI simulation is called L1Muctpi. It has the following properties:

- **`L1Muctpi.OverlapStrategyName`**: Three strategies can be selected with "NULL", "DEMONSTRATOR" and "LUT".
  - "NULL": If this strategy is selected, all the muon candidates coming from the RPC and TGC simulations will be selected, and sent to the CTP.
  - "DEMONSTRATOR": This is the overlap strategy implemented in the current MuCTPI hardware. The hardware's overlap treatment is actually based on look up tables in the MIOCT modules, but it is's simulation is hard-coded, and is not easily changed.
  - "LUT": This is the strategy I implemented recently. If selected, the simulation reads two LUT files, and selects muon candidates based on them. If this strategy is used, the properties `L1Muctpi.UpperLUTFile` and `L1Muctpi.LowerLUTFile` should also be defined.
- **`L1Muctpi.UpperLUTFile`** and **`L1Muctpi.LowerLUTFile`**: One only has to define these, if the "LUT" overlap strategy is to be used. Currently there is only one file in the ~/data directory, called MioctLUTupper.data. It can be defined for both LUT files. More on the format of the LUT files can be found in the analysis section.

Running the digitization

One either has to check out **Simulation/Digitization**, or get the files `DigitizationConfig.py` and `AtlasDigitization.py` into the run directory. (They are both found in **Digitization/share/**.)

The first one is used to set up the input and output files, and set the detector geometry. Remember to use the same geometry here, that was used for the simulation.

In the latter one, you should disable the Inner Detector and the Calorimeter with: `DetFlags.ID_setOff()` and `DetFlags.Calo_setOff()` before `DetFlags.Print()`. This is of course optional, but it speeds up digitization quite a bit, and decreases the result file size significantly.

Depending on whether you checked out the **Digitization** package, or just copied the two job option files, you can run the digitization with: `#>athena.py Digitization/DigitizationConfig.py Digitization/AtlasDigitization.py` or `#>athena.py DigitizationConfig.py AtlasDigitization.py`. Of course saving the log in a file with `2>&1 | tee log_file.log` (in bash that is)

is good practice here as well.

-- AttilaKrasznahorkay - 23 Jan 2006

This topic: Main > RunningDigWithMuCTPI
Topic revision: r3 - 2006-01-23 - AttilaKrasznahorkaySecondary