Table of Contents

RICH mirror alignment and the High Level Trigger

Introduction

Ideas and implementations

For maximal optimization at `Brunel` (i.e. Reconstruction) level [*this* takes up most of the time] we wished to implement an HLT line which selects tracks most likely to populate the outer mirrors of the RICH, in order to fill all of the phi (azimuthal angle) bins of a photon histogram [16 of 20 bins must have 300 photons for the mirror alignment procedure to work] for each mirror combination we use for alignment. Andrew came up with an HLT map to identify these events and potentially select only the tracks that we need for the alignment. This was ideally to be implemented for Run 2.

The HLT map **should still be implemented**, but for now we have leveraged Andrew's work by proceeding with an HLT line that selects whole events, where one track has very low pseudo rapidity and very high momentum. This populates the "outer mirrors" (i.e. least-populated mirrors combinations) quickly enough to ensure the mirror alignment can be performed per run/fill (depending on the allowed rate of the HLT line), but we get many more tracks than we need this way. We could still reduce the number of events we need to populate the mirrors with photons by two orders of magnitude, which would make the mirror alignment lightning-quick.

Return to the Top of this TWIki

Hlt (https://gitlab.cern.ch/lhcb/Hlt ⬚)

We are responsible for our Mirrors HLT Line and the settings. We must communicate with the HLT Operations group if there is any change.

JIRA (HLT specific items)

Most recently active/open/closed tasks should be listed towards the top of each section:

- ◆ **4** HLT Line upgrades for the RICH Mirror Alignment⬚ (RICH-14)
    - ◊ subtask: **4** Reduce the max tracks cut on the mirror HLT lines⬚ (RICH-12)
    - ◊ subtask: **3** Revisit the existing HLT lines to see if it is possible to make it more efficient⬚ (RICH-16)
    - ◊ subtask: **2** Move from the simple HLT line to one based on Andrew's windows (HLT map) or a similar principle
        - · subtask: **2** Make this select individual tracks from the windows, rather than whole events.
    - ◊ subtask: **1** Consider the possibility to use muons to perform the mirror alignment (https://indico.cern.ch/event/477525/⬚)
  - ◆ **4** Add RICH mirror alignment monitor to HLT2 Monitoring⬚ (LBHLT-232)
    - ◊ needs **4** Migrate RichAlignment to new RichFuture sequence in HLT2.⬚ (RICH-11)
    - ◊ needs **4** Investigate possibility of per mirror combination CK angle resolutions from the HLT2 Monitoring⬚ (RICH-17)
  - ◆ **4** Incorporate the required changes to make RichMirrorAlignmentOnline compatible with the newest MirrCombinFit v17r0
    - ◊ see https://gitlab.cern.ch/lhcb/Panoptes/commit/2eefbbe5d188d3317e087183b03f7e01365406f9⬚ for list of changes

- - ♦ **1** Consider running the alignment on MC (Take any MC, say from B Hadron stream; see if it is possible to run HLT line on this and get enough data)
      - ◊ Important for determining best set of mirror combinations for the mirror alignment (and thus also which mirror combinations' magnification factors should be stored in the Online CondDB)

Further information for tasks not yet in JIRA

Move to HLT Map

Things that could be done with regards to speed, that we should be thinking about even if we are already   fast enough":

- Some sort of pre-selection / post-selection related to bin counting in histograms (unlikely, hard to get analyzers to communicate when each histo bin is full)
- Full HLT map selecting single tracks
  - ♦ A HLT map for whole events

Regarding the last point, why not put an HLT map for whole events into the HLT line now. It would to save us such a factor of time [and, the related weighted selection mechanism would be a first step for an eventual HLT map anyway]. The key to doing this well is to perform a study with:

- (A) an unbiased L0-filtered sample   but it may not have enough events
- (B) the existing HLT line sample   have tons of events, just would need to run something over them to figure out the regions and weightings based on the event trigger track. (and of course, remember that our weighted selection gets performed on already HLT line selected data, not on unbiased data)

We may not be able to flatten the distributions but we should be able to cut down events with many central mirror tracks.

One might not ask why not go straight to the map   the only reason is that the map requires isolating and reconstructing individual tracks, and this does not. If along the way isolating tracks becomes possible we can always change gears. So the "whole events" version can possibly be a   simple   extension of the full HLT map.

HLT1 Line

The HLT1 Line for the mirror alignment is absolutely crucial to getting the photons we need, specifically reflecting in the mirror combinations that we use to perform the mirror alignment.

Vava installed the skeleton line, which is based on the one track trigger. Paras provided and refined the selection criteria. Roel and others from the HLT group have been helping us with calibrating the line. Andrew consulted on using ideas from the HLT map work to make this possible.

Our HLT1 line is implemented into the HLT, and is called `Hlt1CalibRICHMirror`. ROUTING BIT 54 provides the data to be accessible Online.

HLT1 Line Details

- The core of our line, currently is:
  - ♦ https://gitlab.cern.ch/lhcb/Hlt/blob/master/Hlt/Hlt1Lines/python/Hlt1Lines/Hlt1CalibRICHMirrorLine
  - ♦ This should match THIRD LINE below, except for the pre-scales.

- The core of our line can be changed by modifying the HltSettings matching the current TCK, the draft for 2017 is:
  - ♦ https://gitlab.cern.ch/lhcb/Hlt/blob/master/Hlt/HltSettings/python/HltSettings/Physics_pp_2017.py ☒

- In early 2016 we ran with THIRD LINE
- Mid-2016 we were asked to reduce our prescales from those in THIRD LINE below. In L0-saturated running, this reduced our rate from ~750 Hz to ~250 Hz
  - ♦ https://gitlab.cern.ch/lhcb/Hlt/blob/master/Hlt/HltSettings/python/HltSettings/Physics_pp_Draft2016.p
- In early 2017 we are asking HLT OPS to raise our rate, to avoid a slow trickle when we start.

Current Line Configuration (THIRD LINE)

▸ Show... ▾ Hide

```
, Hlt1CalibRICHMirrorLinesConf : { 'Prescale' : { 'Hlt1CalibHighPTLowMultTrks'     : 0.0001,
                                                  'Hlt1CalibRICHMirrorRICH1'       : 0.2
                                                      'Hlt1CalibRICHMirrorRICH2'
                                                  'DoTiming' : False
                                                  'R2L_PT'      : 500. * MeV
                                                  'R2L_P'       : 40000. * MeV
                                                  'R2L_MinETA'  : 2.65
                                                  'R2L_MaxETA'  : 2.80
                                                  'R2L_Phis'    : [(-2.59, -2.49), (-0
                                                  'R2L_TrChi2'  : 2.
                                                  'R2L_MinTr'   : 0.5
                                                  'R2L_GEC'     : 'Loose'
                                                  'R1L_PT'      : 500. * MeV
                                                  'R1L_P'       : 20000. * MeV
                                                  'R1L_MinETA'  : 1.6
                                                  'R1L_MaxETA'  : 2.04
                                                  'R1L_Phis'    : [(-2.65, -2.30 ), (-
                                                  'R1L_TrChi2'  : 2.
                                                  'R1L_MinTr'   : 0.5
                                                  'R1L_GEC'     : 'Loose'
                                                  'LM_PT'    : 500. * MeV
                                                  'LM_P'     : 1000. * MeV
                                                  'LM_TrChi2': 2.
                                                  'LM_MinTr' : 1
                                                  'LM_MaxTr' : 40
                                                  'LM_GEC'   : 'Loose'
                                                      }
```

Historical mirror lines

▸ Show... ▾ Hide

- ♦ **SECOND LINE**
  - ◊ Previous line used for RICH2 was not populating RICH1, so RICH2 line stayed the same but for RICH1 a new line was created.

```
, 'R1L_P'       : 20000. * MeV
, 'R1L_MinETA'  : 1.6
, 'R1L_MaxETA'  : 2.04
, 'R1L_Phis'    : [(-2.65, -2.30 ), (-0.80, -0.50), (0.50, 0.80), (2.30, 2.65)]
```

- ♦ **FIRST LINE**
  - ◊ The initial HLT1 line required one track in the event to have track momentum > 40 GeV (eventually, was 20 GeV in very early running) and this selection (where "continue" means the event is **NOT** selected):

```
if( trackEta < 2.59 ) continue; // minimum Eta cut
   if( trackEta > 2.97 ) continue; // maximum Eta cut
```

```
if(!((trackPhi &gt; -2.69 && trackPhi &lt; -2.29 ) || (trackPhi &gt; -0.85 && trackPhi &lt; -0
```

- - - ◊ This selection is designed to focus on the four typically-least populated RICH2 mirrors. If we get enough events producing photons from 40 GeV tracks on those four mirrors, then we will have enough 40 GeV tracks in all of the other mirrors to populate the rest of the mirrors. The thought was that RICH1 would be well-populated from this line alone. That was not the case.

Return to the Top of this TWIki

Future

The RICH mirror alignment using the HLT map does not require many tracks (~50k using HLT map, maybe more with eta cut), but many (5M) have had to be reconstructed in the past, in order to ensure occupancy in all mirrors. For each mirror pair in RICH 2, we need 300 photon hits in at least 16 of 20 bins in phi in order to perform the alignment.

To speed this up, we plan to preselect tracks using HLT1 track information (specifically track location). Aim for constant mirror occupancy, taking efficiencies into account. (weightings)

Andrew knows where to find the tracks we need. His HLT map (on the Bristol SVN, see LHCb-UK talk posted in "Important talks" section above) needs to be implemented into the HLT1 and provide events to our mirror alignment algorithm.

Andrew has shown that the average efficiency (the number of tracks we want for mirror alignment / how many need to be reconstructed) using his method is ~50%

Andrew predicts this means we need on the order of 20k tracks minimum to do the alignment. (50k tracks is a "safer" number we usually quote)

The HLT1 line is **the spine** of our new system. Ideally, this should feed tracks into Brunel to reconstruct photons needed for the mirror alignment, and should force the mirror alignment only to use these tracks.

Ideally, we have the HLT line provide to us the **track** that satisfied the line. We can then just reconstruct that **track** and its photons, which could save O(10) CPU time for Brunel over reconstructing all tracks.

- However, it is likely that the **events** that pass the HLT line are easier to deal with than the tracks, then we reconstruct all tracks in each **event** selected. Perhaps we will find this will be fast enough, but it wastes resources.
    - ♦ We should make it our goal to get the track info though, since speed savings can be made. This savings could be determined by comparing the population of the output histograms.
    - ♦ In the **track** by **track** situation the histograms would be populated evenly. In **event** by **event** the inner mirror pairs will be (unnecessarily) populated more than outer mirror pairs.
    - ♦ Tracks can potentially be stored in a container, then we can refit and reconstruct only the interesting tracks
- Andrew's map provides the x-y position boxes that map track positions to mirror pairs. z information is not provided, but z is basically at the RICH 2 (could use end of T-station z-coordinate as a close-enough estimate, or just look these up).
    - ♦ If this is not easy to implement, can possibly translate the x-y positions into requirements on {P_X, P_Y, P} of the track. However Information about the track state at a specific z should be accesible.

RICH 1 also needs to be aligned, but we should be able to easily use the same tracks collected for RICH2, so the HLT map does not need to be made for RICH 1.

Return to the Top of this TWIki

Details of actual implementation of this future line

Andrew Cook has provided us with a map which is basically contained in two ROOT files [one for each side of the RICH... though given the symmetry of the sides perhaps these files could be merged into one], with these branches (among others): {primary mirror, secondary mirror, xmin, xmax, ymin, ymax, weighting}. We can convert this map into another form if need be (say std::vectors which could be used in C++ code)

for each mirror pair, {xmin, xmax, ymin, ymax} define a box through which the track should pass at a value of z which (as far as we know) is at the plane of the RICH2 primary mirror. We don't know how good an approximation z = end of TT, if we can't get the actual z values, but we could certainly try it. It may make sense also, might be harder to use different z's. "Weighting" is how likely (between 0 and 1) we want this track* to be kept, should it pass through a particular box.

Ideally, a track that passes the TrackCandidates requirements of the HLT line would then be selected in the following way, roughly:

```
r = random(deterministic seed)
for i = 1 to number of mirror pairs
{
   if (track is in x,y box at z)
   {
           if (r < weighting)
           {
           flag the track*
           break;
           }
   }
}
forget the track
```

We may have to implement this differently to fit the constraints of implementing a Tool. Hopefully we can come up with some way of making the seed deterministic [event number, or Gerhard's deterministic prescaler]

Ideally we want to design the Tool so it has a high prob of throwing the track away at an early stage of the Tool.

A JIRA task has been set up for this project: https://its.cern.ch/jira/browse/LHCBPS-1371

Other, unorganized, notes from meeting with Sebastian

- ITrackFilter is a very important class for us. LokiTrackFilter also good too.
- Tools --> class (interface)
- IsMuonTool is a good model to start with
- HLTTracking (Upgrade)
- state is (x,y at z)
- Conditions* in place of ROOT tables --> DBASE? Marco C., Chris.
- simplify parametrization of boxes?
- hard code, but only if number is < 30
- getting a single track is "trivial"
- tool vs. functor
- dont need to clone state
- Roel (heidelberg, now CERN fellow)
- Nightly dashboard
- moore.physics.tracking.hlt1prcheck:
- Tim Head --> Trigger Emulation

- algorithm to run the tool (e.g DumpTracks)

Found elsewhere, Useful info to set up an HLT line...
https://twiki.cern.ch/twiki/bin/view/LHCb/D0ToHHPi0Hlt2DevelopmentPage

Return to the Top of this TWIki

This topic: LHCb > LHCbRichMirrorAlignHLT
Topic revision: r1 - 2017-07-03 - ParasNaik