~~Alignment and calibration on heterogeneous architectures~~

The future of the RICH alignment (touching on heterogeneous architectures)

Alex Marshall[1]

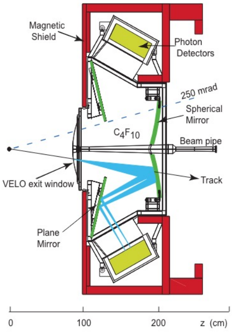
[1]University of Bristol

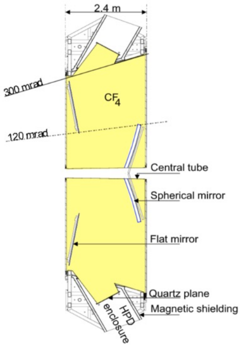
July 9, 2024

Introduction

I will focus on the RICH alignment, I am not sure how much of the conclusions I make apply to other alignments.

- Introduce the RICH alignment procedure
- Layout planned (and un-planned) future improvements to RICH alignment
- Discuss possible uses of *heterogeneous architectures* in the alignment procedure



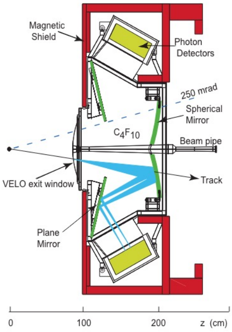


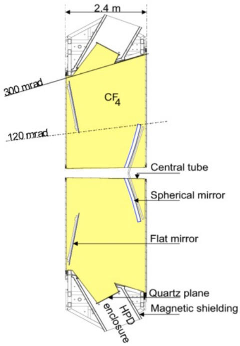
The RICH alignment procedure

Two sets of alignment constants for each RICH:

- Panels
- Mirrors

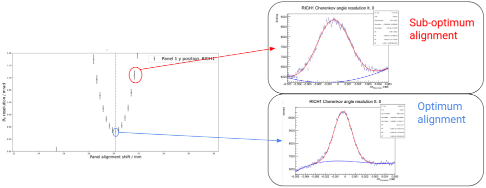
- The constants defining the position of the panels are the most sensitive to the resolution of the Cherenkov angle.
- The panels are therefore aligned first.

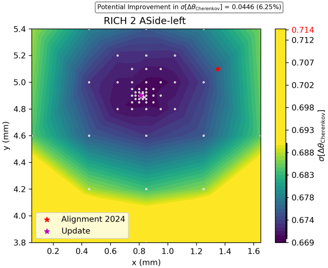




The RICH alignment procedure - Panels

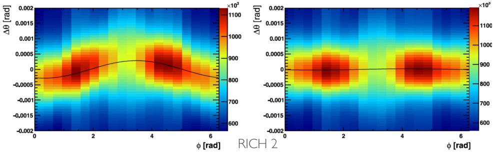
- The alignment of each panel (2 for each RICH) relies on moving two alignment conditions:
  - A translation in x-direction
  - A translation in y-direction
- At each point on a grid of these conditions:
  - Run reconstruction
  - Compute Cherenkov angle resolution - using a fit (Gaussian + polynomial)
- Panel alignment requires only a small number of events, as:
  - It is very obvious when you are in a misaligned region
  - Each optimisation is only of 2 conditions
- Therefore the panel alignment is all done offline, not using the alignment CPU farm
- Each iteration takes $\mathcal{O}(1\ \text{min})$



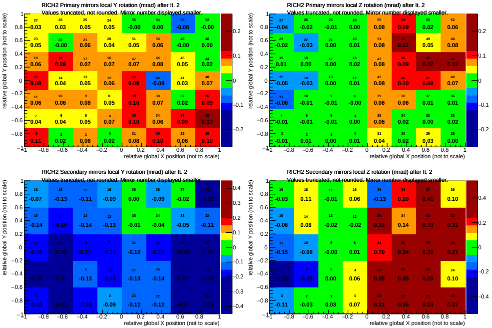


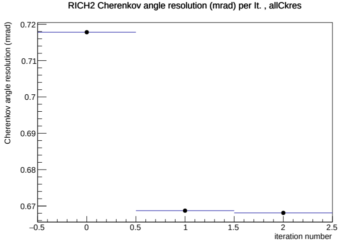
The RICH alignment procedure - Mirrors

- A minimal set of primary and secondary mirror combinations are defined, with this set one can align all mirrors.



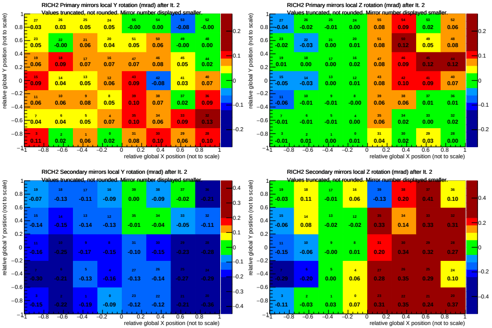
- The alignment is then run as an iterative procedure:
  - Reconstruction is run (with a set of conditions)
  - $\Delta\theta$ vs ϕ histograms for each mirror combination histogram are fit
  - An update to the conditions is suggested and scaled with *magnification factors*
  - Conditions updated
- Convergence is declared when all $\Delta\theta$ vs ϕ fits are consistent with flat to within a defined tolerance.

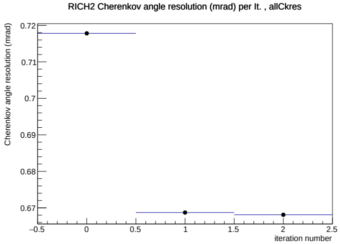




The RICH alignment procedure - Mirrors

- The mirror alignment requires many events $\mathcal{O}$(millions)
  - There are many conditions to optimise.
  - Need to populate outer mirrors.
- Reconstruction is run on the alignment CPU farm
  - The reconstruction for iteration takes $\mathcal{O}$(5 min)
- ROOT histograms are returned
- These histograms are fit and updates to the conditions are computed on a single CPU.
  - Also takes $\mathcal{O}$(few mins)
- The *magnification factors* used are computed once and kept constant.
  - Technically though, one cannot have complete convergence without updating these each iteration.





The RICH alignment - planned updates

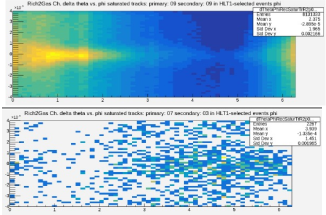
Update 1: **Educated random prescaling based HLT line**

- Currently our alignment HLT lines store every event in our calibration samples
- The populations in our $\Delta\theta$ vs ϕ histograms are extremely imbalanced
- Central mirror combination histograms have 1000x more events than the least populated histograms
- Means a large fraction of our reconstruction time wasted on adding sensitivity to already well-populated regions

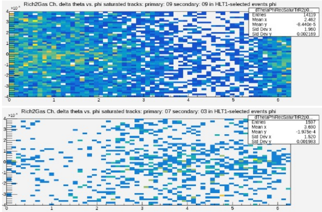
Fix with a random prescaling in our HLT line:

- Each mirror combination given a predetermined probability to keep an events
- Bring populations closer to balanced

INCLUSIVE



RANDOM PRESCALED



Plots and method Anatoly Solomin

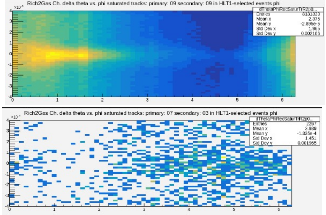
The RICH alignment - planned updates

Update 1: **Educated random prescaling based HLT line**

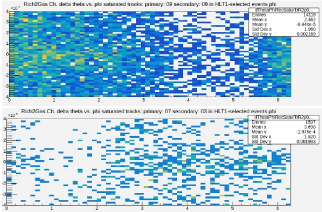
- Result is a ∼x40 faster reconstruction
- The fitting of histograms and computation of updates to conditions will become the bottleneck

- Can look at floating *magnification factors* at each iteration of the mirror alignment:
  - To compute *magnification factors* one must run the reconstruction 9 times for a set of conditions instead of just once.

INCLUSIVE



RANDOM PRESCALED

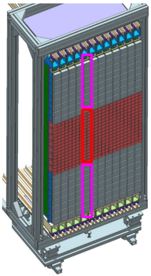


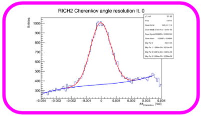
Plots and method Anatoly Solomin

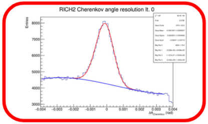
The RICH alignment - planned updates

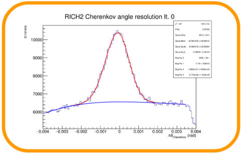
Update 2: **Higher granularity panel alignment**

- ☐ Current each set of panels is aligned as a single unit
- ☐ We could split this up into smaller regions:
  - ▷ Per PMT Column
  - ▷ Large PMTs per PMT column
  - ▷ Small PMTs per PMT column







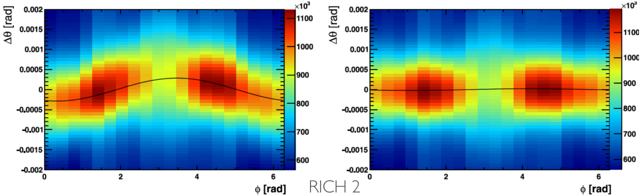


Plots Jake Reich

The RICH alignment - un-planned updates

Update 3 (not yet planned): **Parallelisation of the fitting code**

- When the new HLT lines provide a x40 speed up in reconstruction fitting will be the bottleneck
- Currently the fitting is done on a single CPU
- Would it be easy to split up this work?

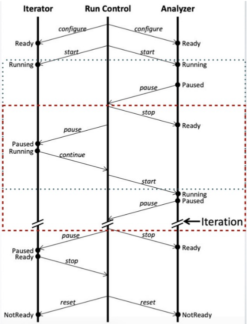


RICH 2

The RICH alignment - heterogeneous architectures

The future of RICH alignment:

- More mirrors will be added → new optimal set of mirror combinations and more alignment conditions to optimise.
- Timing → this will affect how we select tracks in our alignment events
- In the future RICH reconstruction will be done with GPUs
  - However, I am assuming there will always be a CPU version of the same reconstruction available?
- Moving the fitting code to GPUs (for example) would be a very significant amount of work, not sure how much benefit?
- Running the reconstruction with GPUs, with fitting code still on CPU?
  - If that could be done in such a way as: reconstruction is called, runs on GPU, and ROOT histograms returned, then sure.



Notes:

- RICH alignment constants appear to be more stable in Run3 than in Run2
- RICH alignment just needs to be fast enough to take a small fraction of a fill

Backup