A graph of different colored lines

Description automatically generated

Upon alignment with Windows and Linux, it was seen that Average TDC0 and 1 and 2 were significantly lower for Win and Linux. Michael suggested it was due to lower amount of corruption in data.

A graph of different colored lines

Description automatically generated

One weird event from Windows data. Michael found it to be due to noise during shutting off.

Efficiency studies

A graph of different colored lines

Description automatically generated

This was used to probe the efficiency of the RPCs using reconstruction.

A graph of different colored lines

Description automatically generated

A graph of different colored lines

Description automatically generatedA graph of different colored lines

Description automatically generated

BAD

Skipping channel 0

A graph of different colored lines

Description automatically generated

Surprisingly, TDC2 and 3 were brought back to alignment after skipping the noisy channels, however TDC3 and 4 remained unalignable, however a clear change of state was seen when partial alignment was successful.

A graph of different colored lines

Description automatically generated

Higher overall efficiency

A graph of different colored lines

Description automatically generated

3 orders of magnitude difference (average 100 possible reconstructions, quality factor filters out everything)

A graph of different colored lines

Description automatically generated

This much clearly shows that the shape of this curve determine alignment, which is good. Currently unsure the reason for the unaligned RPC5 enforcing a skipping channel condition.

To analys time, skip chunk was used to quickily walk through regions that needn’t alignment

if we updated consectively, we skip 100 chunks

A graph of a number of squares

Description automatically generated with medium confidenceA screen shot of a grid

Description automatically generatedA green and blue grid with yellow dots

Description automatically generatedA green and blue squares

Description automatically generatedA grid of blue squares

Description automatically generatedA screenshot of a graph

Description automatically generatedA graph of a number of blue squares

Description automatically generated with medium confidenceA screen shot of a graph

Description automatically generatedA graph of a doublet low

Description automatically generatedA green and blue grid

Description automatically generatedA graph with a doublet top

Description automatically generated with medium confidenceA green and blue grid

Description automatically generatedA graph showing different colored lines

Description automatically generatedA graph with colorful lines

Description automatically generated

Made the code cleverer. Now monitors Michael’s other metrics.

A graph of a number of different colored lines

Description automatically generated with medium confidence

We say that if there are more than 4 peaks above a hittime of 300, we name the status of the TDC 3 to be broken, which means we don’t attempt to realign it, instead we use [01,12,24] any other times we use [01,12,23,34]

This reduced the time of processing by 3 fold, but not very useful, since TDC oscillates between dead and alive status, and no clear distinction was seen between the two status in terms of the hit times.

It seems like this behavior is not consistent, especially when you have removed the channel, so really you are hoping the first channel

A graph of a graph

Description automatically generated  
This is not a consistent effect in the broken region, and cannot be used to probe if the TDC is in alive status or notA graph of a broken region

Description automatically generatedA graph with numbers and lines

Description automatically generated

Ok so averaging Is needed, this effect could not be observed using 100 intervals. Hence a larger interval had to be looked at to see if the TDC is dead or alive

Lets try 7000

god region 6704

god region 6724

god region 6472

god region 7190

god region 7060

god region 6954

god region 6938

god region 6648

god region 6283

god region 6116

god region 6110

god region 6488

god region 6323

god region 6602

bad region 7361

bad region 8416

bad region 8363

bad region 9223

bad region 9077

bad region 8695

bad region 8071

bad region 8800

bad region 9099

bad region 8458

bad region 8265

bad region 8274

bad region 8581

bad region 8803

bad region 8644

bad region 8395

bad region 8824

bad region 8131

bad region 8802

A graph with blue and orange bars

Description automatically generated

So 7600 used

Hit time defined by the total amount of time a interval of regions had a event time larger than 700ns

Ahh first hittime instead of hittime

A graph of different colored bars

Description automatically generated

A graph of a graph with numbers and a number of bars

Description automatically generated with medium confidencetry 5000 and 3800