LHRS Analysis for d_2^n

VDC Tracking Efficiency, Trigger Efficiency & Scintillator Studies

D. Flay

Hadronic & Nuclear Physics Group
Temple University Physics Department

7/1/10

Outline

VDC Tracking Efficiency (1)

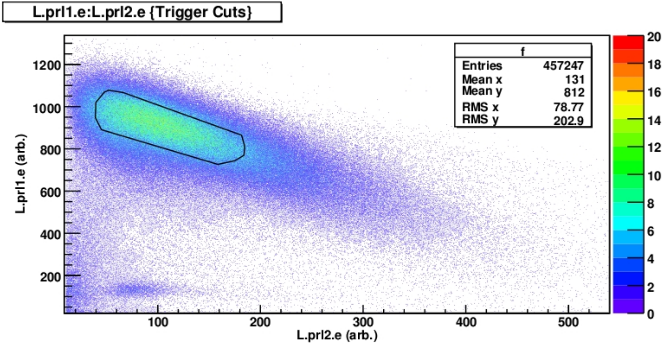
Method: Selection of Events

- We need to first select good events for our study. We do this by considering only those events which generate a T3 trigger and pass our PID cuts:
  - *Trigger Cuts*: (DL.edtpl==0)&&((DL.evtypebits&(1<<3))==(1<<3))
  - *PID Cuts*: (L.cer.asum_c>300)&&(Graphical Cut on 2D Pion Rejector Energy Plot) &&(TDC Cuts on the Gas Čerenkov)

VDC Tracking Efficiency (2)

Method: Selection of Events

- We cannot use the E/p cut here – it relies on tracking! Therefore, we use a graphical cut on the PR 2D energy plot:



VDC Tracking Efficiency (3)

Method: Calculating the Tracking Efficiency

- For our physics studies, we usually choose one-track events ⇒ there's an efficiency tied to this cut – what is this value?
- Firstly, the inefficiency of the VDC is attributed to no-track events and multi-track events
  - The inefficiency is dominated by the latter case, arising from many particles traversing the planes of the VDC, which results in a large number of possible trajectories that could be reconstructed by the software
- Therefore, the efficiency of the one-track cut is given as:

$$\varepsilon_1 = \frac{N_1}{\sum_{i=0}^{4} N_i}$$

Note: The VDC software reconstructs up to four tracks per event

VDC Tracking Efficiency (4)

Results: Elastic Runs 1229, 1230

VDC Tracking Efficiency		
# of Tracks	# of Events	ε (%)
0	16	0.003 ± 0.001
1	526460	98.981 ± 0.192
2	4946	0.929 ± 0.013
3	425	0.079 ± 0.004
4	33	0.006 ± 0.001

- Two-track events tend to dominate the inefficiency – typical across the whole kinematic range

VDC Tracking Efficiency (5)

Full Results

- The full results of the study for each major kinematic setting in the LHRS
- $\varepsilon_1 \sim 99\%$, total inefficiency $\leq \sim 1\%$
- Consistent across the whole kinematic range

VDC One-Track Efficiency		
p (GeV)	E (GeV)	ε_1 (%)
1.23	1.23	98.981 ± 0.192
0.60	4.73	99.282 ± 0.592
0.60	5.89	99.339 ± 0.430
0.80	4.73	99.209 ± 0.843
0.90	5.89	99.293 ± 0.796
1.13	5.89	99.228 ± 1.037
1.20	5.89	99.213 ± 1.307
1.27	5.89	99.172 ± 0.967
1.42	4.73	99.189 ± 1.235
1.42	5.89	98.810 ± 1.176
1.51	4.73	99.104 ± 1.149
1.51	5.89	99.172 ± 1.326
1.60	4.73	98.953 ± 1.421
1.60	5.89	98.832 ± 1.413
1.70	5.89	98.620 ± 1.924

Trigger Efficiency (1)

Method

- For the main (T3) trigger efficiency, one usually considers the T4 trigger – as this variable gives the inefficiency of the T3 trigger:

$$\varepsilon_{T3} = \frac{N_{T3}}{N_{T3} + N_{T4}}$$

  where the N_j are the number of events for either the T3 or T4 trigger

- There are a few points at which we can lose track of a T3 trigger:

  1. We generate a T3, and does not pass the prescale condition (depsite the fact that $ps = 1$ for production) at the Trigger Supervisor (TS)
  2. We generate a T3, it passes the prescale condition, but does not pass the L1A – a T4 beat it there

Trigger Efficiency (2)

Method

- Variables to consider:
  - The '`DL.LTN`' variable corresponds to those events that generated a trigger of type N ('raw trigger')
  - The '`DL.bitN`' variable corresponds to those events that pass the prescale condition, and we set the corresponding bit for trigger N ($\Rightarrow$ set the bit pattern)
  - The '`DL.L_l1a`' variable is the L1A – whichever trigger gets to the L1A first 'takes the timing' and the CODA event is set as type N
- This brings us to the main question: **which ratio do we consider for the T3 efficiency?**

$$\varepsilon_{T3} = \frac{\mathrm{L1A}}{\mathrm{bit3} + \mathrm{bit4}}$$

$$\varepsilon_{T3} = \frac{\mathrm{bit3}}{\mathrm{LT3} + \mathrm{LT4}}$$

In general, we send all of our possible triggers at a fictional detector, and see how many T3's 'stick' $\Rightarrow$ utilize evtypebits. . .

S2m Left-Right Time Difference (1)

Method

1. Before we take a look at adjusting the time averages for each paddle, we first minimize the left-right time difference for each paddle – using the raw TDC variables
   - Effectively start from scratch here
2. Plot the L (blue) and R (red) TDCs (see next slide) and shift the L TDCs to coincide with those of the R TDCs – this corresponds to the L-R time difference $\delta^{LR} = t^L - t^R$
3. Insert an arbitrary (global) offset of 700 channels for both L and R TDCs (for paddle n):

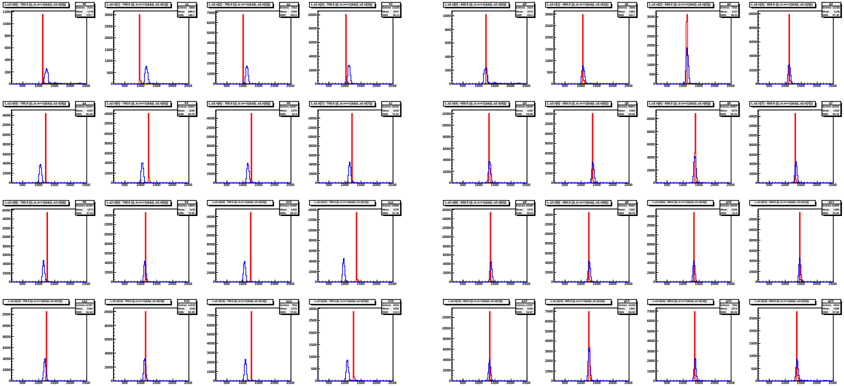
$$
\begin{aligned}
t_n'^L &= t_n^L - 700 - \delta_n^{LR} = t_n^L + \delta_n^L \\
t_n'^R &= t_n^R - 700 = t_n^R + \delta_n^R
\end{aligned}
$$

   - $\delta_n^{L,R}$ are inserted into the DB

S2m Left-Right Time Difference (2)

Results: Before & After

- Check the offsets before putting in the DB – left shows TDCs with no offsets; right shows TDCs after inserting the offsets
- Both cases use the raw variables

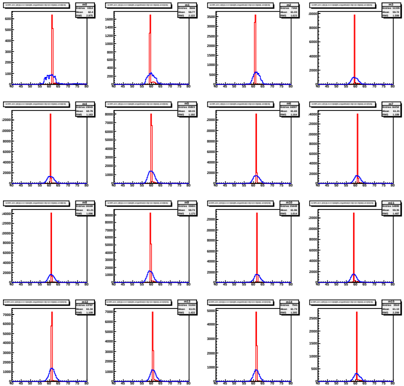


S2m Left-Right Time Difference (3)

Results: Before & After

- Looking at the corrected variables
- The width in the L (blue) TDCs is probably due to the way the corrected variable is built in the source code:

```
fLT_c[i] = (fLT[i] - fLOff[i])*fTdc2T
           - TimeWalkCorrection(i,kLeft)
```



S2m Time Average Calibration (1)

Method

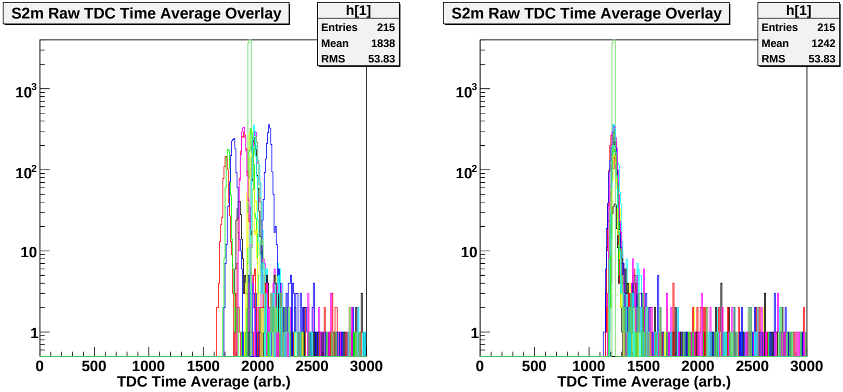
- We look to align the time averages of each paddle in the following way
  1. Choose a reference paddle (I chose paddle 7)
  2. Align the $i-1/i+1$ paddles to the reference paddle
  3. Walk this alignment down to paddle 0 and up to paddle 15
     - Require both the i^{th} and $(i-1)^{\mathrm{th}}$ [$(i+1)^{\mathrm{th}}$] paddle to have a TDC for $i>7$ [$i<7$]
     - Require that for such events, the $(i-1)^{\mathrm{th}}$ [$(i+1)^{\mathrm{th}}$] paddle takes the timing when correcting the i^{th} paddle $\Rightarrow$ cut on the $(i-1)^{\mathrm{th}}$ [$(i+1)^{\mathrm{th}}$] peak
  4. Calculate the peak position for each paddle, relative to its overlap $=\delta_{i,i\pm1}$
  5. Calculate the offsets (to insert into the DB):

$$\Delta_n^{L,R} = \begin{cases} \delta_n^{L,R} + \sum_{i=n}^{7} \delta_{i,i+1} & \text{for} \quad n<7 \\ \delta_n^{L,R} + \sum_{i=7}^{n} \delta_{i,i-1} & \text{for} \quad n>7 \end{cases}$$

Note: All calculations are done with raw variables

S2m Time Average Calibration (2)

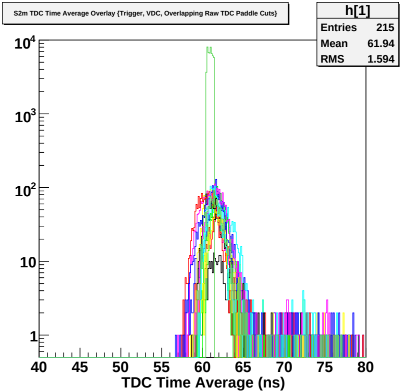
Results: Before & After



S2m Time Average Calibration (3)

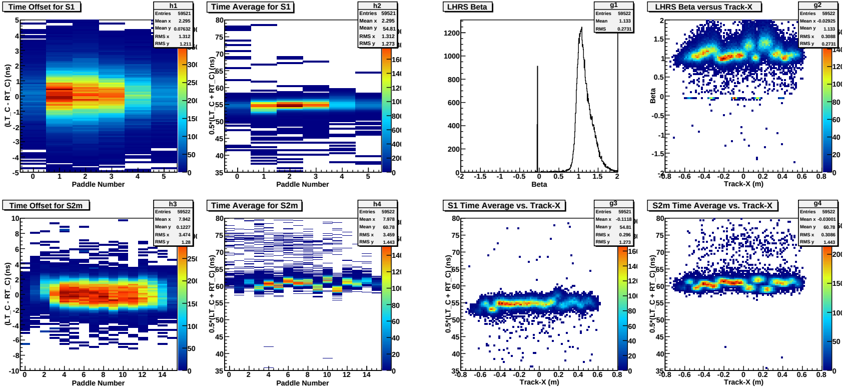
Results: Before & After

- Check how the corrected varibles look
- Again, widening of the peaks is most likely due to how the corrected variable is constructed



S2m Time Average Calibration (4)

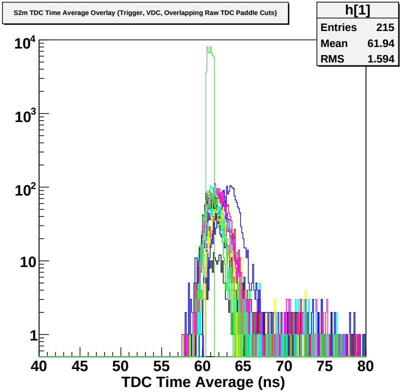
Results: Before & After



S2m Time Average Calibration (5)

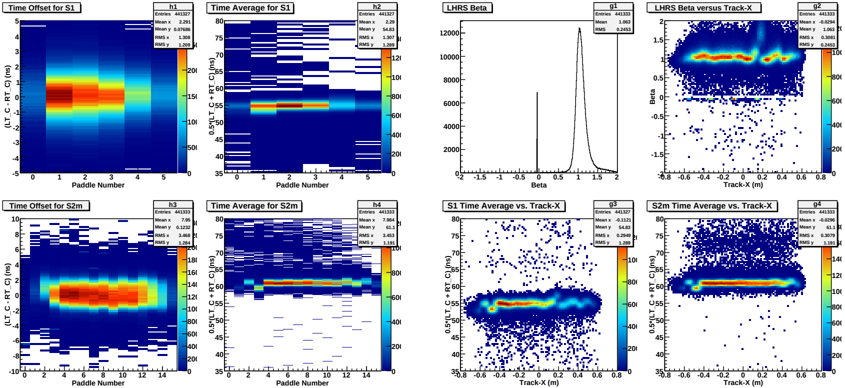
Results: Before & After (Round 2)

- We can clean up the S2m time averages one-by-one
- Overlay of time averages



S2m Time Average Calibration (6)

Results: Before & After (Round 2)



Summary

- VDC Efficiency Study:
  - $\varepsilon_1 \sim 99\%$ across the whole kinematic range
  - Inefficiency is dominated by two-track events
    - Total inefficiency $\leq \sim 1\%$
  - The full details of the study may be found on my website
- Trigger Efficiency Study:
  - Just getting started – thinking about proper ratios to consider
- Scintillators:
  - This ‘by hand’ method yields decent results on the level of L-R offsets for each paddle
  - Inconsistent results for the time average
    - Further ‘one-by-one’ iterations clean it up

What's Next?

- VDC Study:
  - Check on the t_0 calibration (?)
- Trigger Efficiency Study:
  - Debug & write in proper code for the calculation of the T3 efficiency
- Scintillators:
  - Finish cleaning up the S2m time averages and try to get S1 to level out (?)
    - Vince mentions mentions something about the trigger time from S2m being the reason for the S1 jitter (we know that) but does this mean we're not looking at the right variables to tweak?