PionLT Luminosity Analysis

Nathan Heinrich
University of Regina
2025 – 09 – 24
heinricn@uregina.ca

Overview

- BCM Calibration and Zero current offset
  - Make Scaler Yields flat
- Carbon Singles
  - Remove Singles rate dependence
- LH2 Elastic Singles
  - Determine Target Boiling Correction
- LH2 Physics Coincidences
  - Make sure physics data do not have any remaining rate dependence

See my previous talk:
https://hallcweb.jlab.org/doc-private/ShowDocument?docid=1264

Carbon Singles - LiveTime

EDTM system is found to be unreliable
(especially with multiple prescales active)

→ See Jacob Murphy's report: DocDB id 1177

Instead use: CPULT x ELT

CPULT from scalers

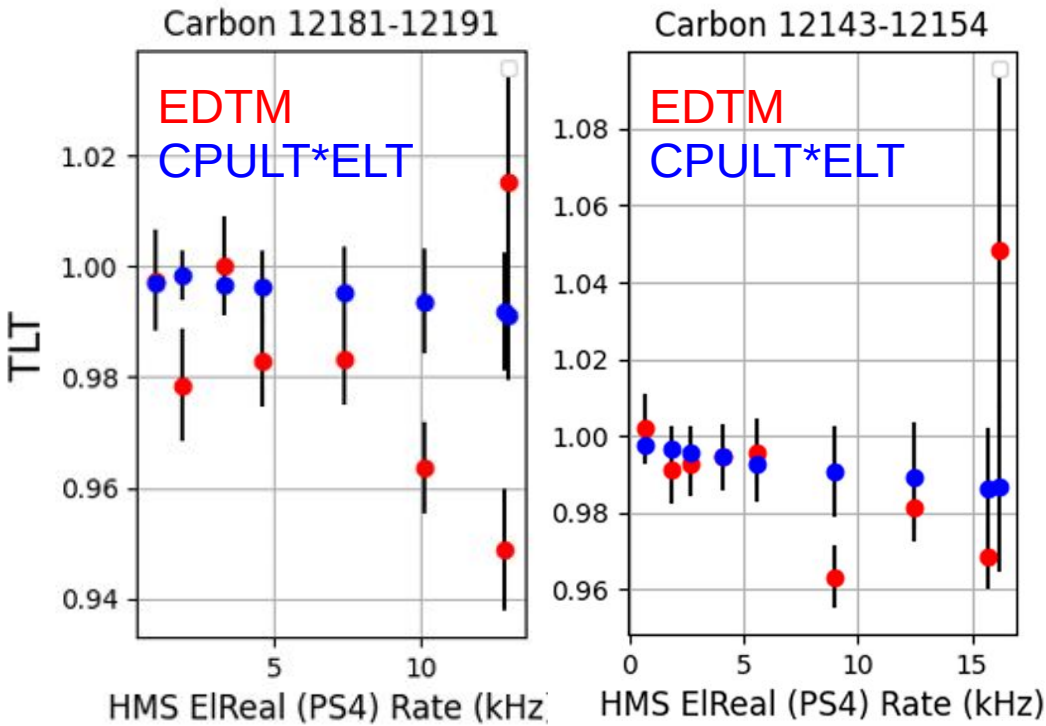
ELT estimated using Hodoscope plane combinatorics (D. Mack's Method)

DT = rate*GateWidth, LT = 1 - DT

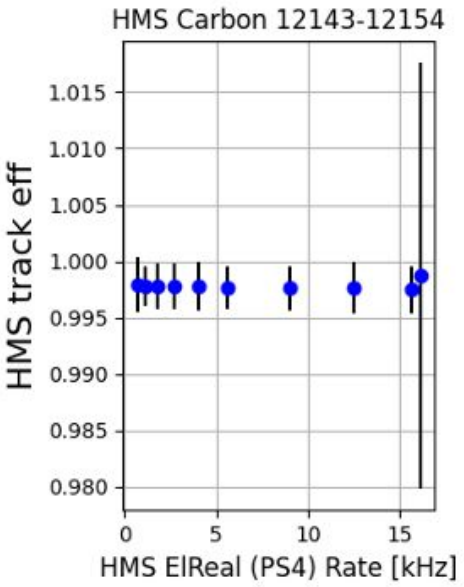
so

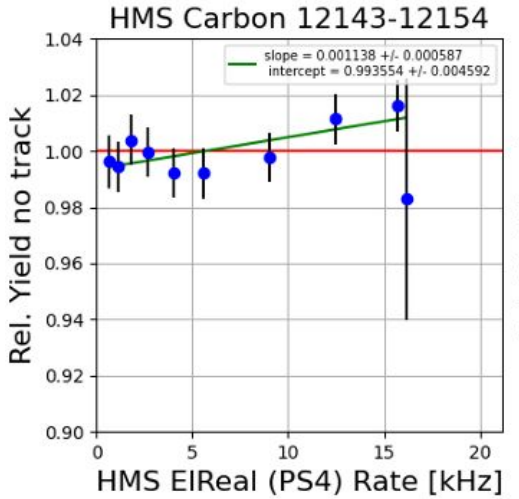
LT3of4 = LT1*LT2*LT3*LT4 + DT1*LT2*LT3*LT4
+ LT1*DT2*LT3*LT4 + LT1*LT2*DT3*LT4
+ LT1*LT2*LT3*DT4

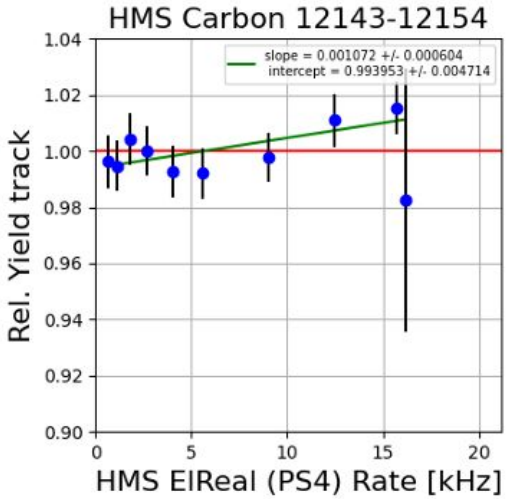
DocDB id: 1063

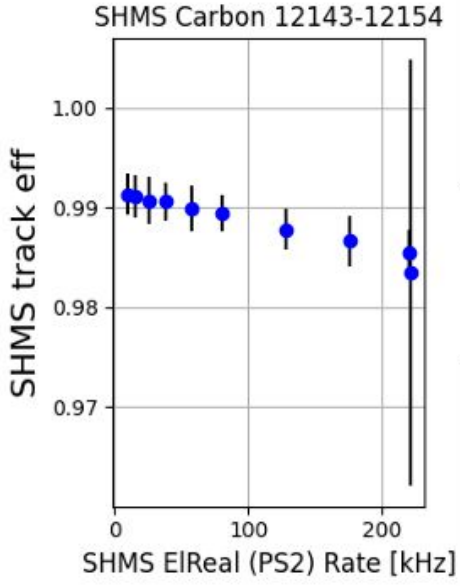


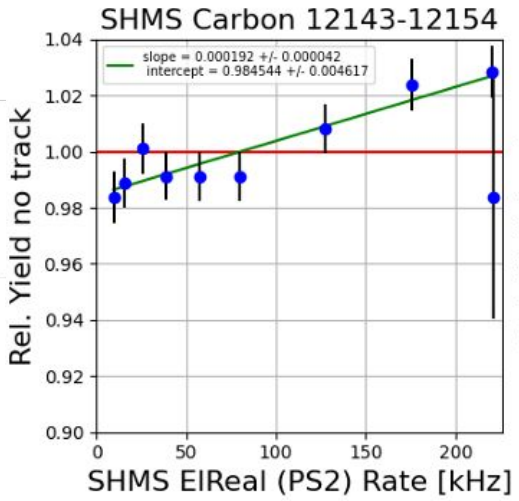
Carbon Singles - Tracking

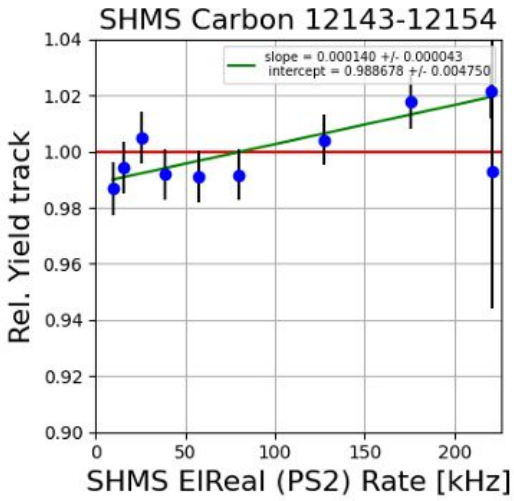








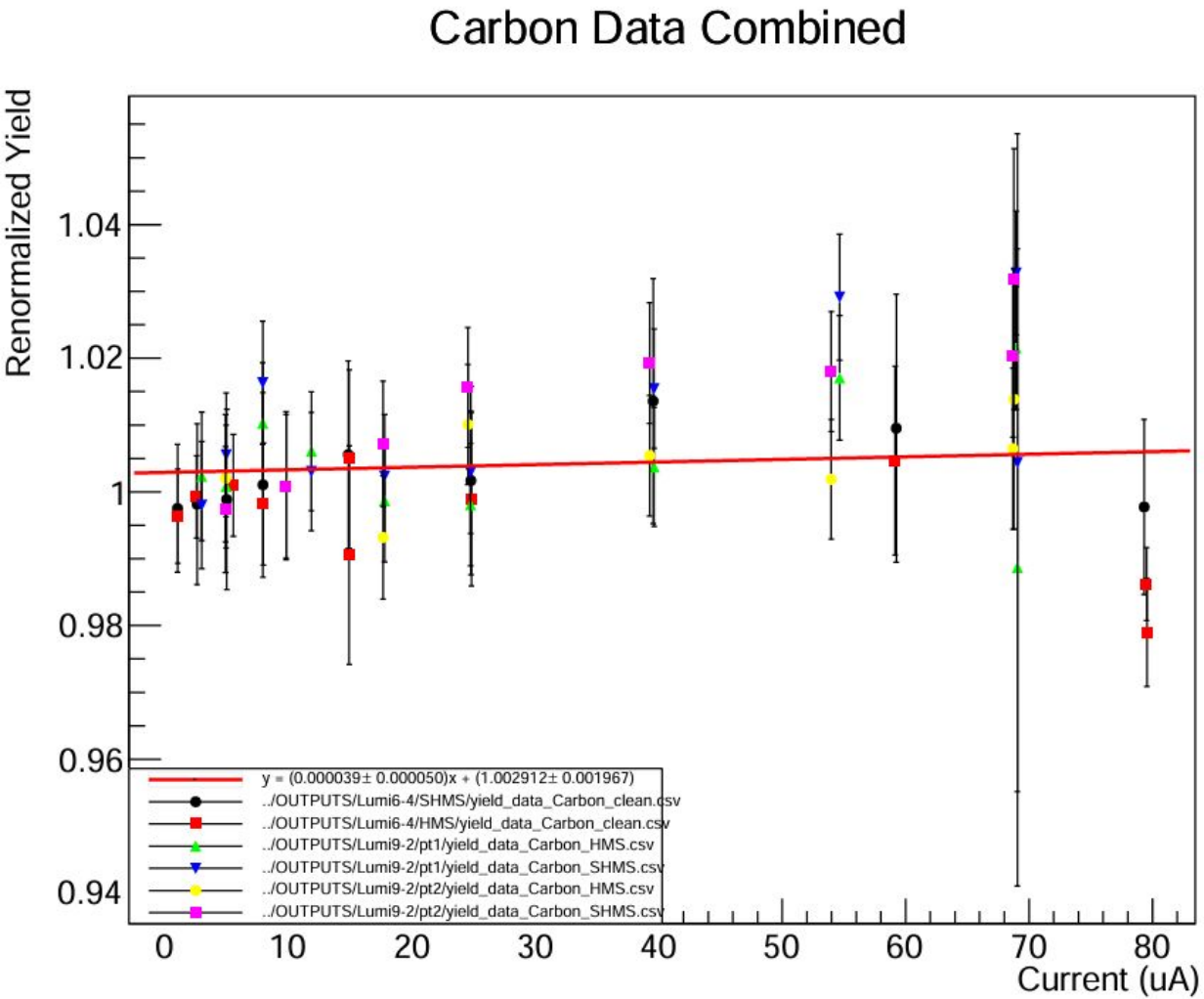




No significant deviation from no-Track to tracked

Tracking efficiency is well understood

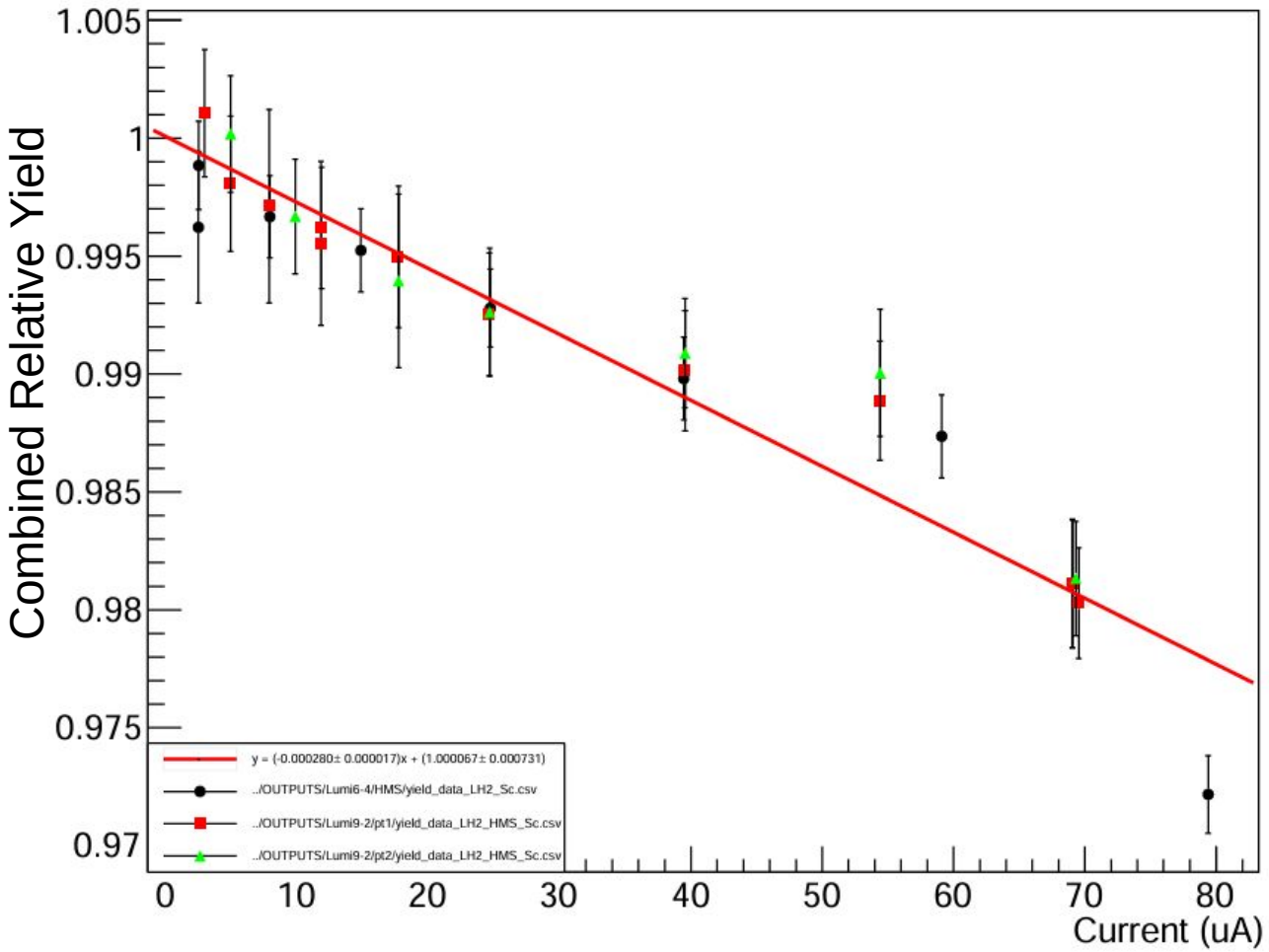
Carbon Singles - Flat



Slope of combined Carbon singles settings:

0.39 ± 0.50 %/100uA

LH2 singles - Boiling



Combine HMS settings determine common Boiling correction,
Get:
2.8 ± 0.17 ± 0.69 %/100uA
10cm target with 60Hz fan speed
Other results:

C. Yero	6.3 ± 0.6 %/100uA
H. Bhatt	3.2 ± 0.4 %/100uA
Deepak	3.2 ± 0.8 %/100uA
R. Trotta	7.9 ± 1.8 %/100uA

Aside:
This plot is generated by fitting each setting individually and off-setting the data so intercept is 1.

LH2 Coincidences

During R-Sidis Running Peter Boosted noticed a potential rate dependence in his Coincidence data.

To properly study this potential dependence PionLT took several sets of coincidence data with varying current using physics data i.e. H(e,e'π)X
We had 2 settings which of Sidis pions, and 5 exclusive πn final state.

With this data I apply all corrections from singles and test for any remaining rate dependence.

First, check the Coin Reference time cuts in tcoin.param to ensure they are appropriate

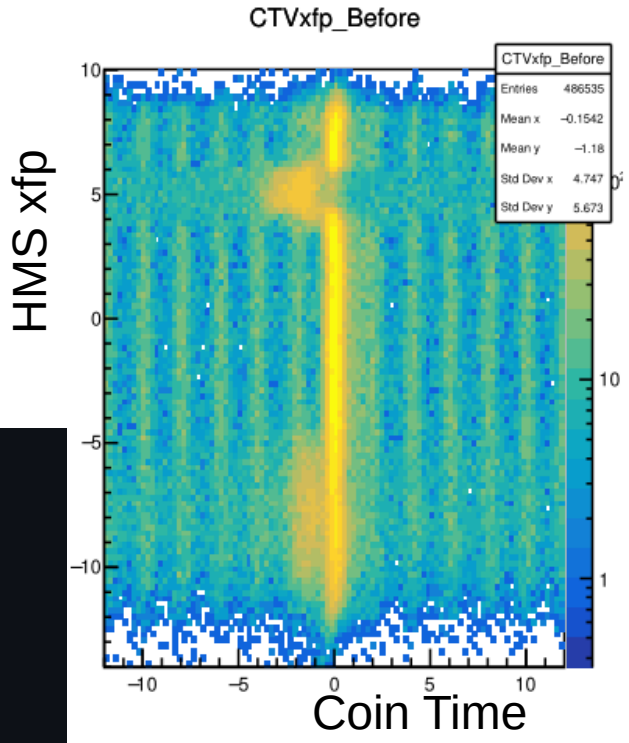
Coin Reference Times

Poor reference time cuts can cause weird behaviours in Coin Time leading to strange rate dependence

Choose good cut values on pTrig tdc time variables corresponding to 3/4 trigger in both spectrometers in tcoin.param file (see talk by C. Yero DocDB id 1209)

```
20
21    t_coin_TdcTimeWindowMin = 0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
22    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
23    0 ,0 ,0, 0, 0, 0, 0, 4800, 0, 4200,
24    0,0 ,0, 0, 0, 0, 0, 0, 0, 0,
25    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
26    0 ,0 ,0, 0, 0, 0, 0, 0, 5500, 0,
27    5100 ,0,0, 0, 0, 0, 0, 0, 0, 0,
28    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
29    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
30    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
31    0 ,0 ,0, 0, 0, 0, 0, 0, 0, 0,
32    0,0,0, 0, 0
33
34    t_coin_TdcTimeWindowMax = 10000 ,10000 ,10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
35    10000 ,10000 ,10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
36    10000 ,10000 ,10000, 5800, 10000, 10000, 10000, 6200, 10000, 5600,
37    10000,10000 ,10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
38    10000 ,10000 ,10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
39    10000 ,10000 ,10000, 10000, 10000, 10000, 10000, 10000, 7000, 10000,
```

tcoin.param File with good time windows

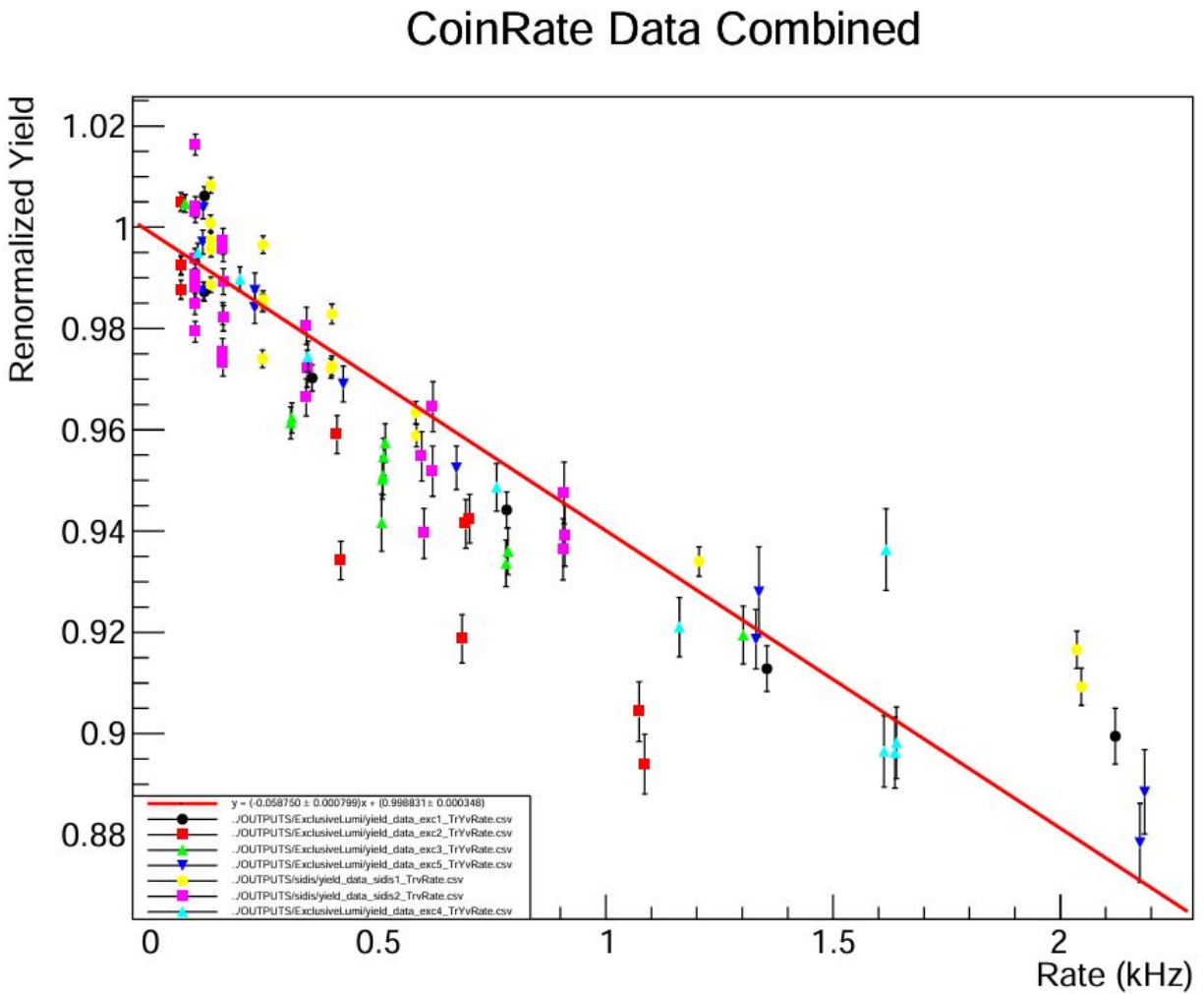


Coin Yields - Mrk1

After applying all the corrections from singles and having appropriate Ref time cuts

Still have a large rate dependence

Idea: maybe it’s Coin Blocking?



Coin Blocking Correction – 6GeV Era

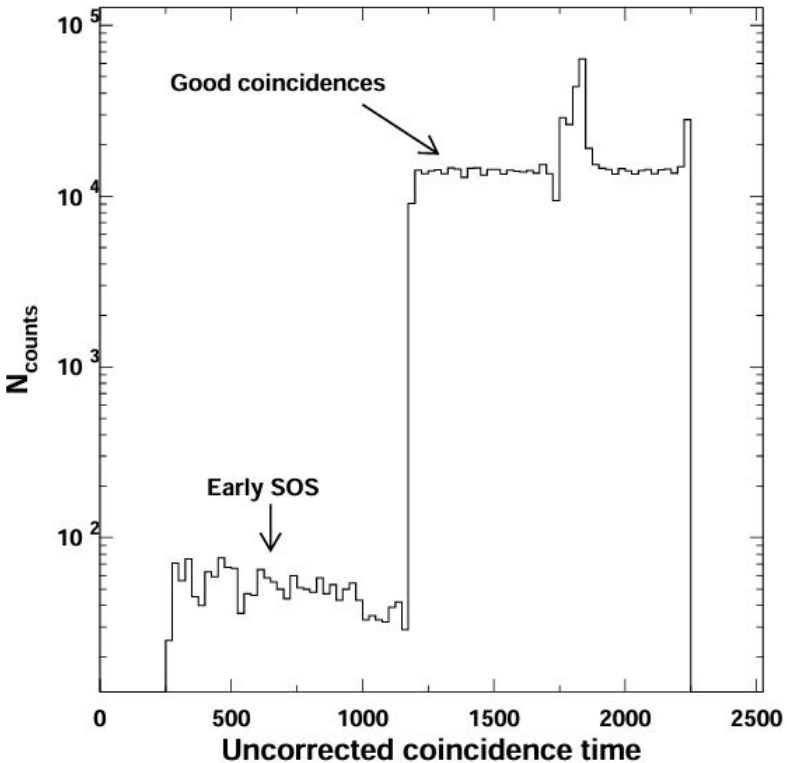
In 6GeV data there was a problem where early singles events would “Steal” the coin time:

3.6 Coincidence Blocking

The coincidence time between the spectrometers is used in the analysis to define good coincidence events. Such a coincidence event will normally be started at the TDC with a delayed HMS trigger (for HMS coincidence time as used in the analysis), and stopped by the SOS. However, due to interference between non-coincident and coincident events a fraction of events are recorded with a value of coincidence time outside the main timing window as defined by the pretrigger signal widths. These “coincidence blocking” events will be lost from the data due to the coincidence time cuts used in the analysis.

Then a correction is calculated as the ratio of good to bad events:

$$Coinblock = \frac{N_{blk}}{N_{tot}},$$

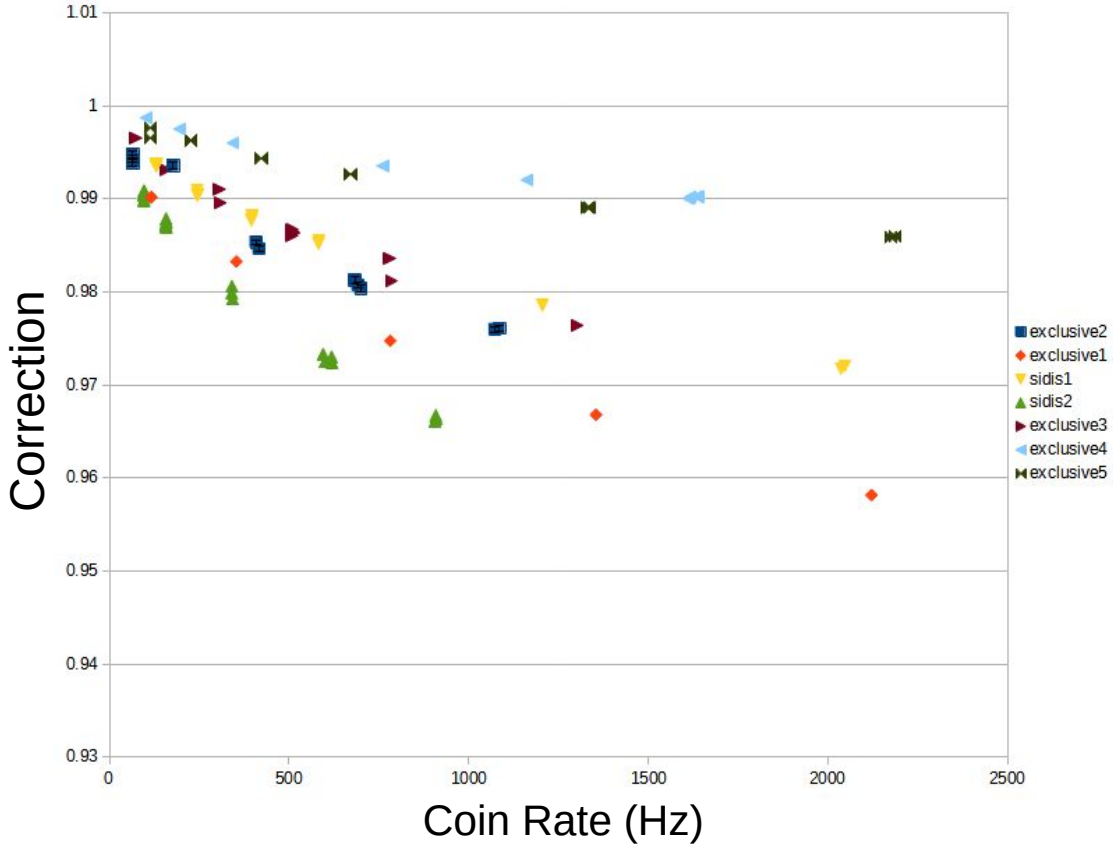


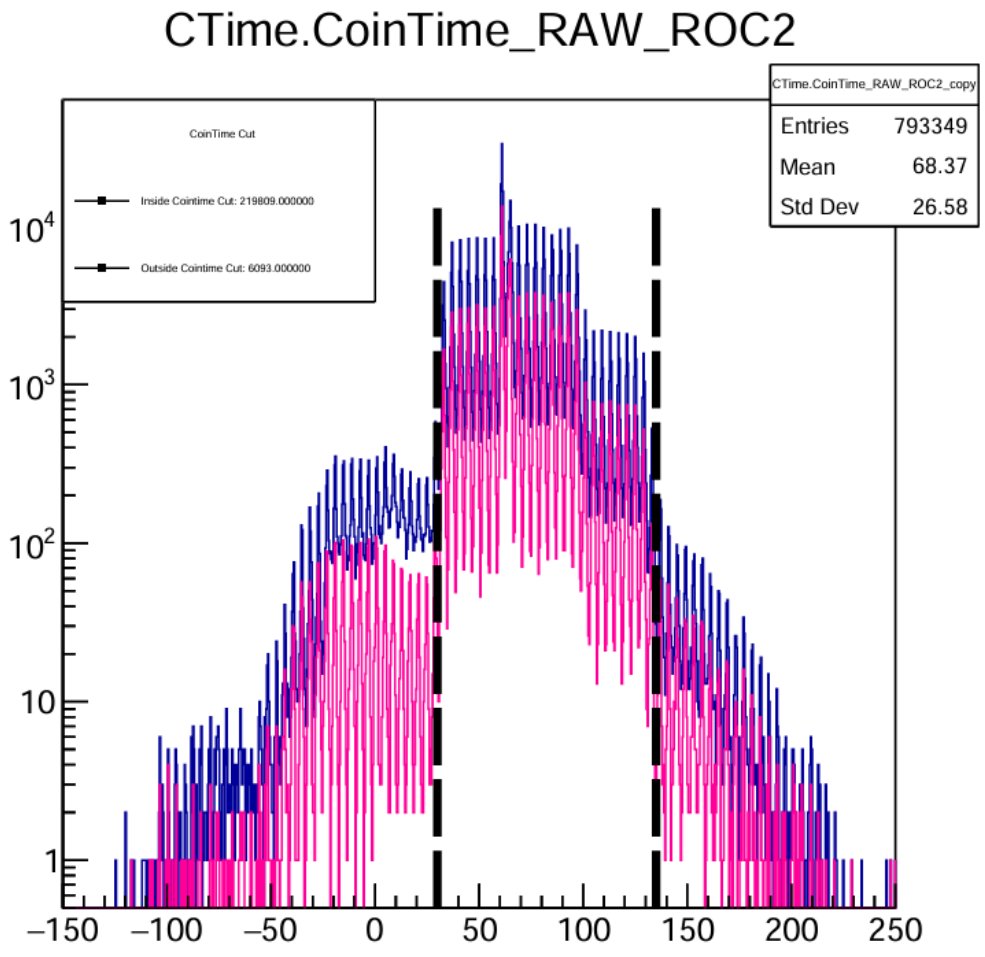
T. Horn PhD Thesis (2006):
https://misportal.jlab.org/sti/publications/7939

Coin Blocking Correction – 12 GeV Era

I find a similar effect in the PionLT data.

This correction can be as large as 4%



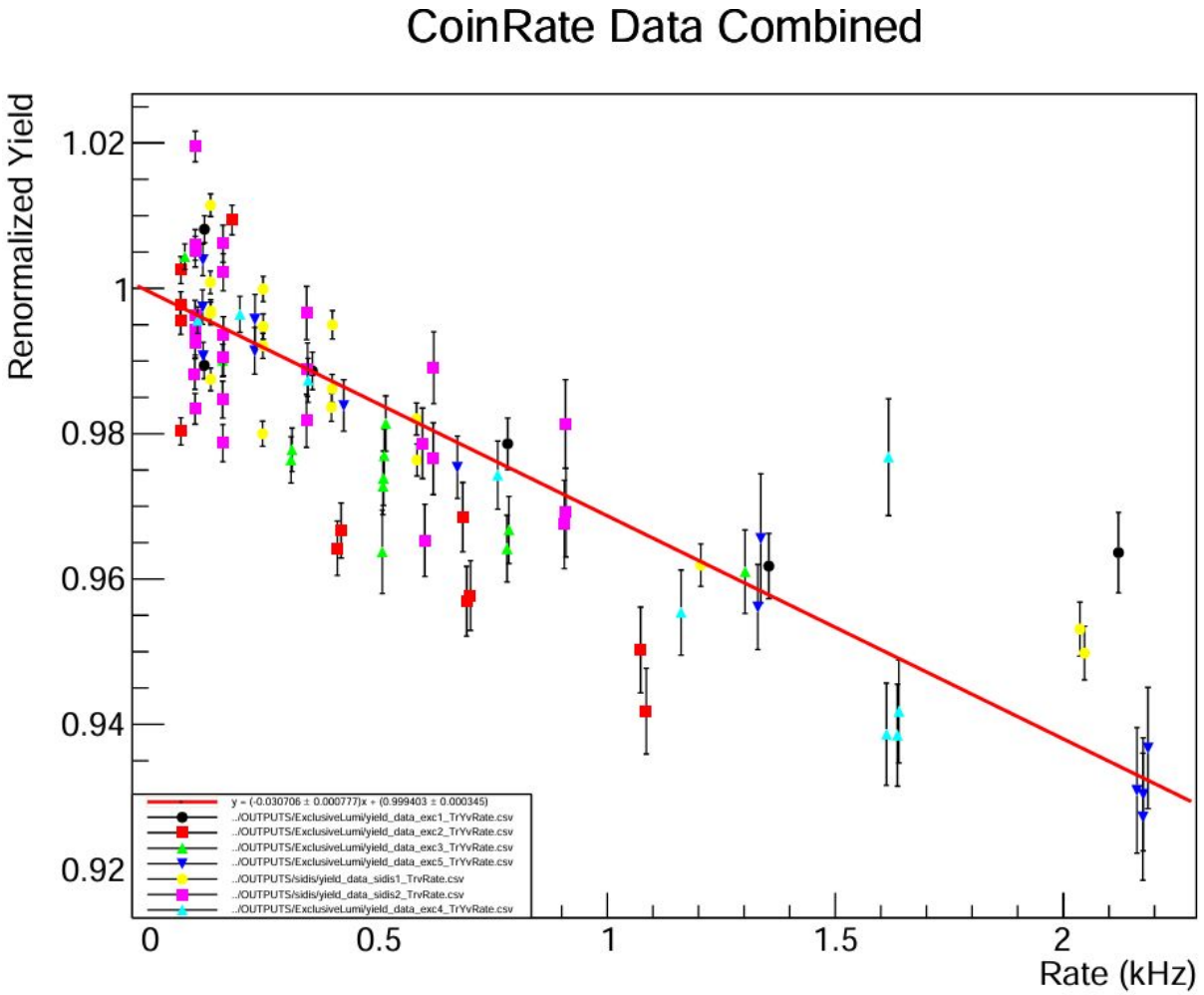


Ad-Hoc Correction?

After all these correction we have a remaining unexplained rate dependence equivalent to

3.1%/kHz pTrig5 rate

Presently the plan is to simply apply an Ad-Hoc Correction to flatten this rate dependence.



PionLT Yield Summary

Calculation of Coin Yield Up to this point, constants inserted when possible.
I've tested these cuts and found no rate dependence

Cuts:
Coin Events Prompt Cut: SHMS_pion & HMS_electron & Current & CoinTime_prompt
Coin Events Random Cut: SHMS_pion & HMS_electron & Current & CoinTime_rand
HMS_electron:
H_cer_npeSum > 1.5 & H_cal_etottracknorm > 0.7 & H_gtr_dp < 8 & H_gtr_dp > -8
SHMS_pion:
P_gtr_dp > -10 & P_gtr_dp > 20 & P_aero_npeSum > 1.5
Current:
|(Avg_current – Set_current)| < Threshold
Coin Time cut expression is long, in backup.

Prescale factor is 1 for coin data

Coin Yield: Y = [(Coin_Prompt – (Coin_Rand/6))*Prescale_factor]/Q_effective

Q_effective = Q_BCM2*Current_correction*eLT_SHMS*eLT_HMS*CPULT*
HMS_track_Efficiency*SHMS_track_Efficiency*Boiling_correction*CoinBlocking

Backup Zone

Run Numbers of Data Used

9-2GeV Carbon Setting 1	12143 12144 12145 12147 12148 12150 12151 12152 12153 12154
9-2GeV Carbon Setting 2	12181 12184 12185 12186 12187 12189 12190 12191
9-2GeV LH2 Setting 1	12126 12128 12129 12130 12131 12132 12133 12135 12137 12140 12141 12142 12156
9-2GeV LH2 Setting 2	12192 12193 12194 12195 12196 12197 12198 12199
6-4GeV HMS Only Carbon	16727 16728 16729 16730 16731 16732 16733 16734 16735 16736 16737
6-4GeV HMS Only LH2	16703 16704 16706 16708 16709 16710 16711 16712
6-4GeV SHMS Only Carbon	16738 16739 16740 16741 16742 16743 16744 16745 16746
6-4GeV SHMS Only LH2	16759 16760 16761 16762 16763 16764
Coin Sidis 1	12200 12201 12202 12203 12204 12205 12206 12207 12208 12209 12210 12211 12212 12213 12214 12215 12216 12217
Coin Sidis 2	12218 12219 12220 12221 12222 12223 12233 12234 12238 12239 12243 12244 12245 12246 12247 12248 12250 12251 12252 12253 12254 12255 12256 12257 12258
Coin Exclusive 1	14601 14602 14603 14604 14605 14606
Coin Exclusive 2	14613 14614 14615 14616 14617 14618 14619 14620 14621 14622 14623 14624 14625 14626 14627
Coin Exclusive 3	15816 15817 15818 15819 15820 15828 15830 15831 15832 15833 15834 15835 15836
Coin Exclusive 4	16220 16224 16225 16226 16236 16237 16238 16239 16242
Coin Exclusive 5	16258 16259 16260 16261 16262 16265 16266 16268 16269 16270 16271 16272 16273 16275 16276

Details of Corrections

Current_correction = (I_nominal-offset)/I
eLT_SHMS – see slide 3, use hodoscope plane rates as rate, and 50ns as gate width
eLT_HMS – same as SHMS
CPULT – accepted_trigger/(sent_triggers/prescale factor)

HMS_track_Efficiency -> see implementation in report files:
SHMS_track_Efficiency
https://github.com/heinricn/hallc_replay_lt/blob/master/TEMPLATES/COIN/PRODUCTION/KaonLT_TEMP/KaonLT_Offline_Luminosity.template

Boiling_correction: 1 – (I_corrected*0.0028)

Coin Time cuts

Coin_Pion_Prompt = ((CTime_ePiCoinTime_ROC2 > (-((CT.Bunch_Spacing)/2.0))) & (CTime_ePiCoinTime_ROC2 < ((CT.Bunch_Spacing)/2.0)))

Coin_Pion_Rand = (((CTime_ePiCoinTime_ROC2 > (((CT.Bunch_Spacing)/2.0)-(2*(CT.Bunch_Spacing))-((6/2.0)*(CT.Bunch_Spacing)))) & (CTime_ePiCoinTime_ROC2 < (-((CT.Bunch_Spacing)/2.0)-(2*(CT.Bunch_Spacing)))))|((CTime_ePiCoinTime_ROC2 > (0+((CT.Bunch_Spacing)/2.0)+(2*(CT.Bunch_Spacing)))) & (CTime_ePiCoinTime_ROC2 < (0+((CT.Bunch_Spacing)/2.0)+(2*(CT.Bunch_Spacing))+((6/2.0)*(CT.Bunch_Spacing))))))