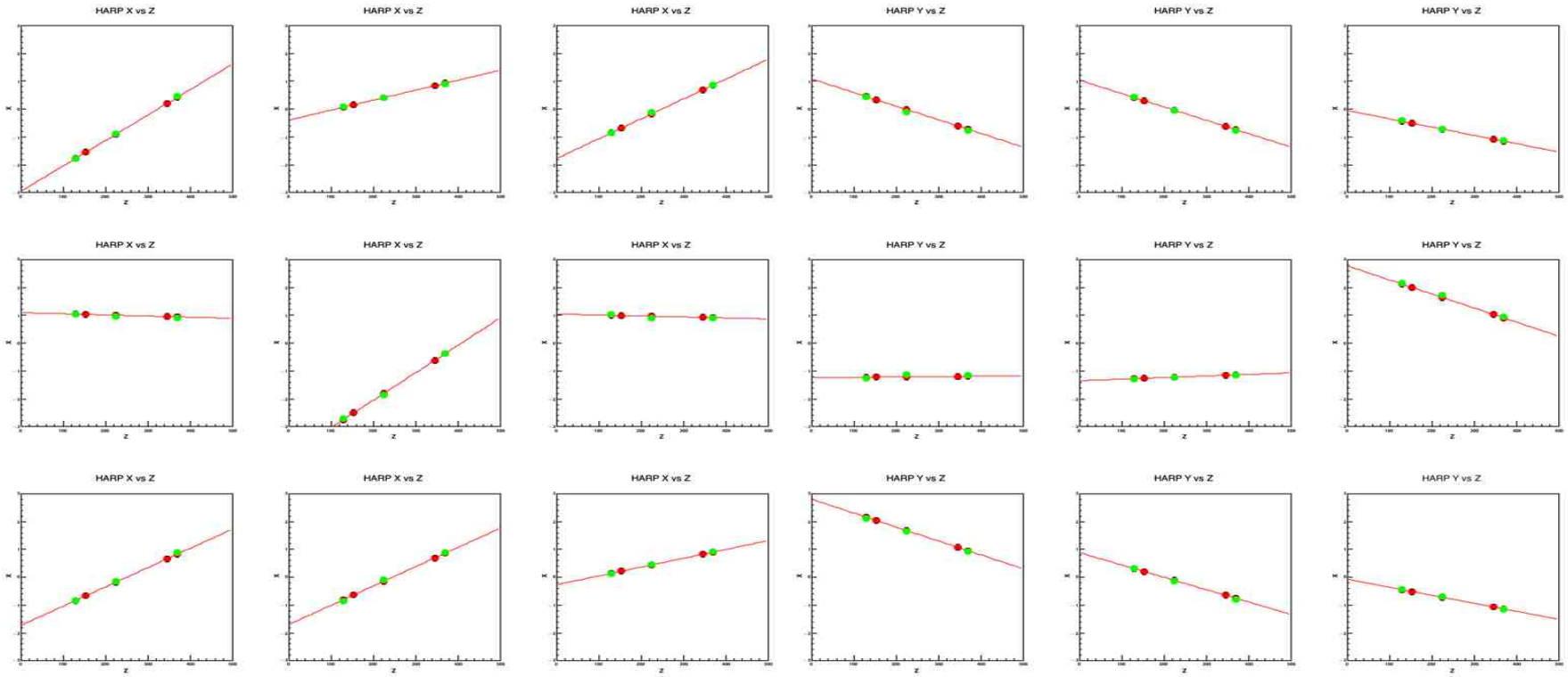
BPM Calibration Update

E. Brash

August 9, 2018

Sanity Checks for Internal Consistency



Gains and Offsets

- Projected X/Y Position = slope * Raw EPICS BPM X/Y Position + offset
- A calibration script now exists in hallc_replay to calculate these parameters, which can then be added to ~/PARAM/GEN/gbeam.param

```
BPM Ax:  Slope = -1.00111 +/- 0.278043
BPM Ax:  Constant = -0.123099 +/- 0.255359

BPM Bx:  Slope = -1.24023 +/- 0.126607
BPM Bx:  Constant = -0.061674 +/- 0.0863716

BPM Cx:  Slope = -0.940987 +/- 0.058928
BPM Cx:  Constant = -1.00727 +/- 0.0792616

BPM Ay:  Slope = 0.957734 +/- 0.133041
BPM Ay:  Constant = -0.44177 +/- 0.114019

BPM By:  Slope = 1.19394 +/- 0.100589
BPM By:  Constant = 0.190897 +/- 0.0863716

BPM Cy:  Slope = 0.842772 +/- 0.052127
BPM Cy:  Constant = 0.549773 +/- 0.076011
```

Beam Position at Target (z = 0)

$X(z) = m_x*z + b_x$ and $Y(z) – m_y*z + b_y$

Calculate slopes from A and C BPM's (longest lever arm):

$m_x = (A_x – C_x)/(A_z – C_z)$ and $m_y = (A_y – C_y)/(A_z – C_z)$

Then calculate b_x and b_y (positions at target):

$b_x = A_x – m_x*A_z$ and $b_y = A_y – m_y*A_y$

Finally, add raster X and Y values to these offsets to get rastered beam position at the target.

EPICS data in detector classes

- The previous algorithm requires the BPM information event by event (as the raster information is event by event).
- Need the calibrated BPM information from (last) EPICS event in the THcRaster class.
- Steve Wood has added the capability to get the EPICS event handler to the THaAnalyzer class
- Requires creating an instance of the THcAnalyzer object in the detector class, and then one can use the various THaAnalyzer Get* methods to get the THaEpicsEvtHandler event handler object, and in turn one can use then the various THaEpicsEvtHandler Get* methods to get the EPICS data of interest.

THcRaster.h

```
#include "THaEpicsEvtHandler.h"
protected:
         THaEpicsEvtHandler* fEpicsHandler;
```

THcRaster.cxx

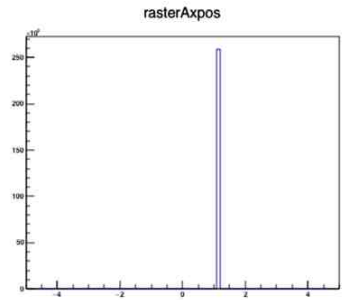
(in somewhere like the ReadDatabase method …)

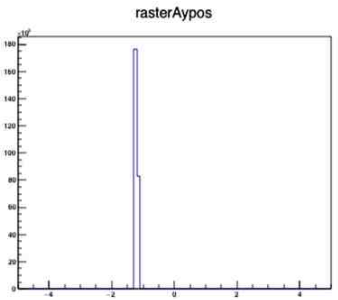
```
         THcAnalyzer *analyzer = dynamic_cast<THcAnalyzer*>(THcAnalyzer::GetInstance());
         fEpicsHandler = analyzer->GetEpicsEvtHandler();
```

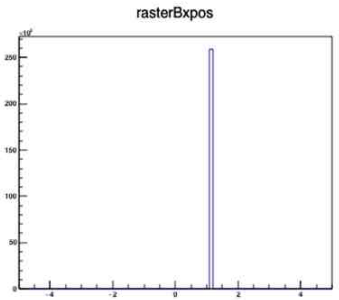
(in the Decode method …)

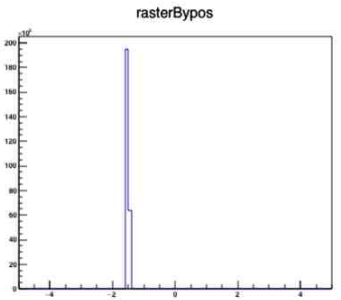
```
         if (fEpicsHandler) {
           if (fEpicsHandler->IsLoaded("IPM3H07A.XRAW")){
                 BPMXA_raw = atof(fEpicsHandler->GetString("IPM3H07A.XRAW"));
           }
         }
```

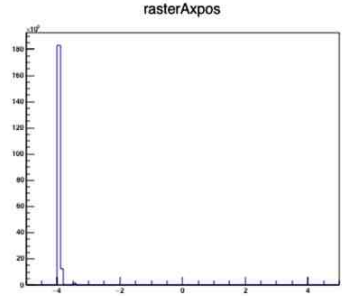
N.B. Need to be careful with units!!! BPM calibrations are in mm, raster is in cm currently!!

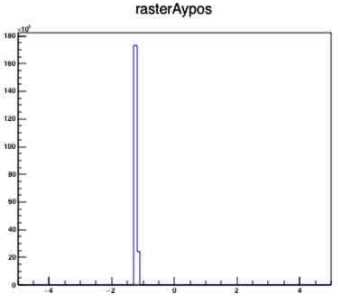

rasterAxpos

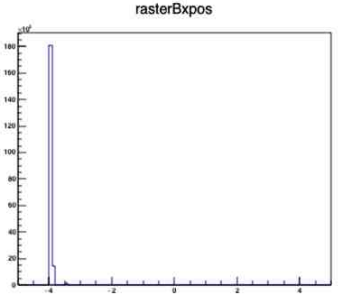

rasterAypos

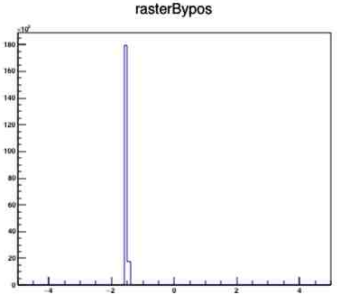

rasterBxpos

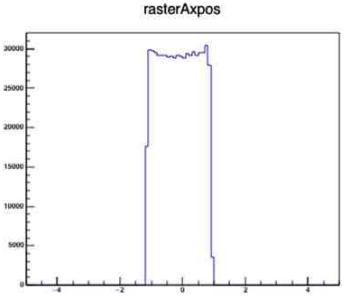

rasterBypos

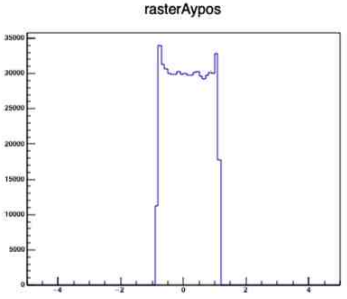

rasterAxpos

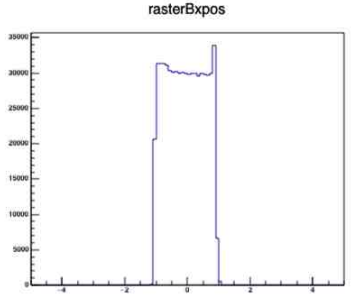

rasterAypos

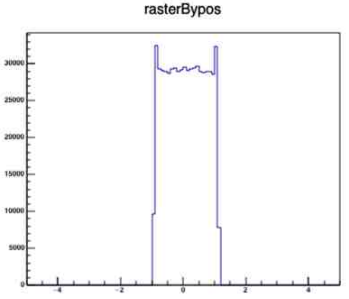

rasterBxpos

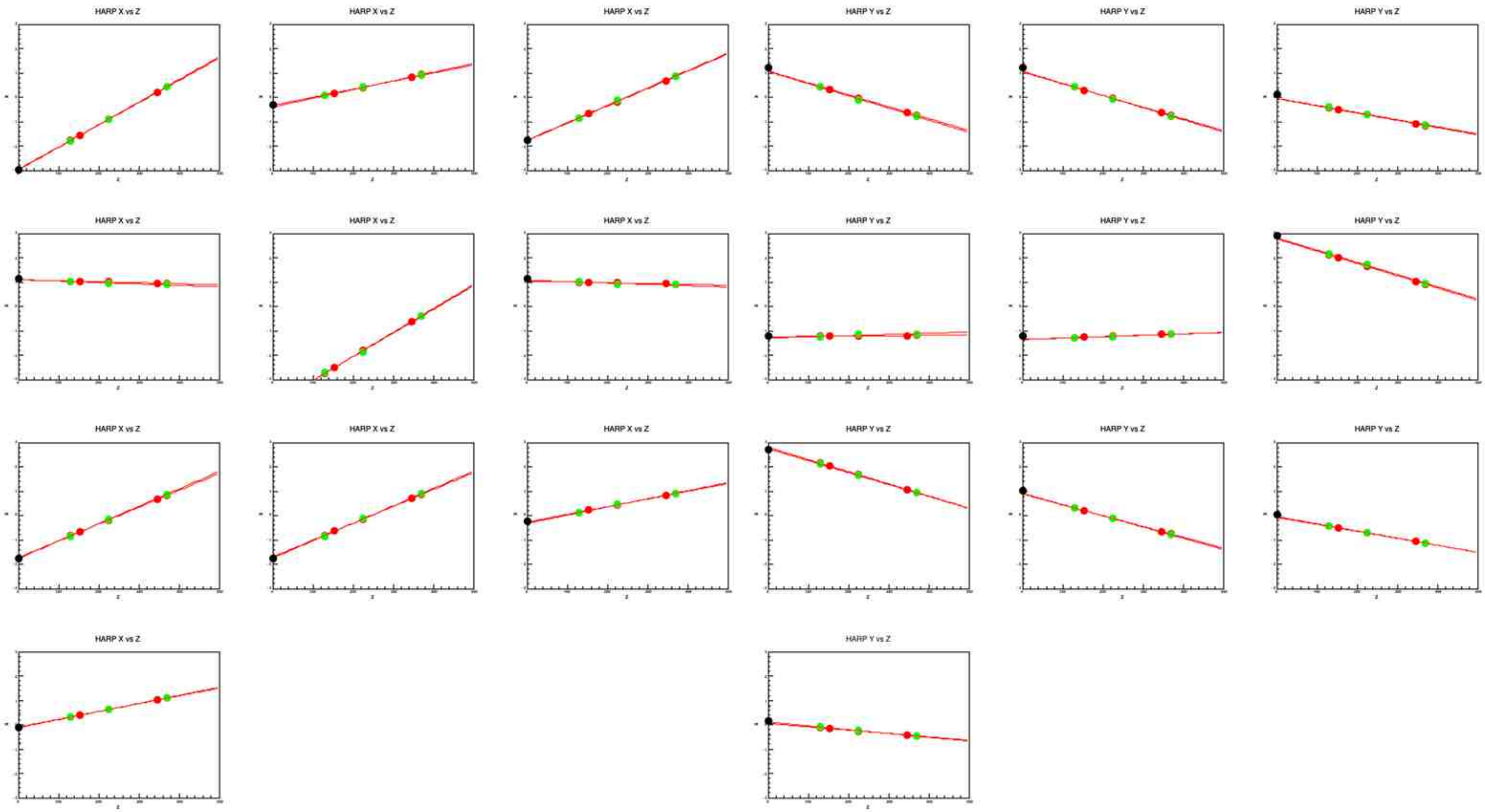

rasterBypos


rasterAxpos


rasterAypos


rasterBxpos


rasterBypos


HARP X vs Z
HARP X vs Z
HARP X vs Z
HARP Y vs Z
HARP Y vs Z
HARP Y vs Z
HARP X vs Z
HARP X vs Z
HARP X vs Z
HARP Y vs Z
HARP Y vs Z
HARP Y vs Z
HARP X vs Z
HARP X vs Z
HARP X vs Z
HARP Y vs Z
HARP Y vs Z
HARP Y vs Z
HARP X vs Z
HARP Y vs Z