

tpass2 Recon QA

Norman Graf (SLAC)

HPS Software Meeting, December 4, 2018

“it’s been a long week”

Me, in the middle of Monday

tpass2 Recon / Analysis

- Most significant issue with the tpass2 recon (noted so far) is the shift in vertex z position.
- My presentation at the collaboration meeting was meant to convince you that the target z position, as determined with FEEs, Møllers and V0s, was at $z=-3.9\text{mm}$.
- Analysis of tpass2 output gives $z=-8!$
- What gives?

Alignment!?

- Results I had shown were based on the survey geometry plus some global tweaks to account for the SVT opening angle.
- HPS-PhysicsRun2016-Pass2 was based on a further millepede alignment.
- Hard to understand how the millepede alignment could produce such a significant shift in z .
- Worked my way back through most of the millepede alignment steps, arriving back at my starting point.
 - Z shift, and degradation in resolution still there.

Code Changes

- There were a lot of code changes introduced right before tpass2.
- Rewind to earlier this year, before any of the vertex code changes.
- Reconstruct using old code base and new detector.

Code Rewind

```
> git log
```

```
...
```

```
commit 9aa1186d25a16ff8918c8ea304e0ca6f60cffb3e
```

```
Author: Matt Graham <mgraham@slac.stanford.edu>
```

```
Date: Mon May 14 17:51:26 2018 -0600
```

```
really add vertexing stuff
```

```
commit 346389b0df995bd619292f87b9b40f9122681b4d
```

```
Author: Matt Graham <mgraham@slac.stanford.edu>
```

```
Date: Mon May 14 07:24:52 2018 -0600
```

```
add V0 momentum and target pointing position and errors; kludge for GBL track finding with no hits bug
```

```
commit 2b99ba366ff09ac3c745297e75ceaa64eea45547
```

```
Author: Matthew Solt <mrsolt@stanford.edu>
```

```
Date: Tue May 8 14:39:31 2018 -0700
```

```
adding 1.5 mm detector
```

```
...
```

```
so May 8 looked good. Check out a version of the git master from that date.
```

```
> git checkout `git rev-list -1 --before="May 8 2018" master`
```

```
That version doesn't have the latest detector, so fetch that from the master,
```

```
> git checkout master -- detector-data/detectors/HPS-PhysicsRun2016-Pass2
```

```
then build the jar file,
```

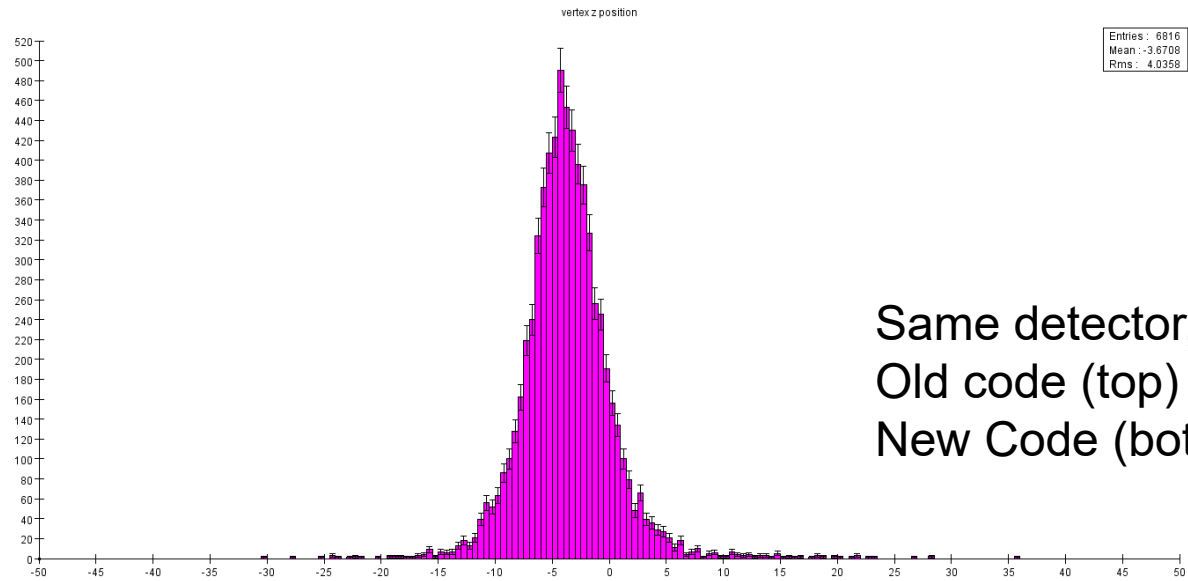
```
> mvn clean install -DskipTests
```

```
Run over a 2016 V0 skim file:
```

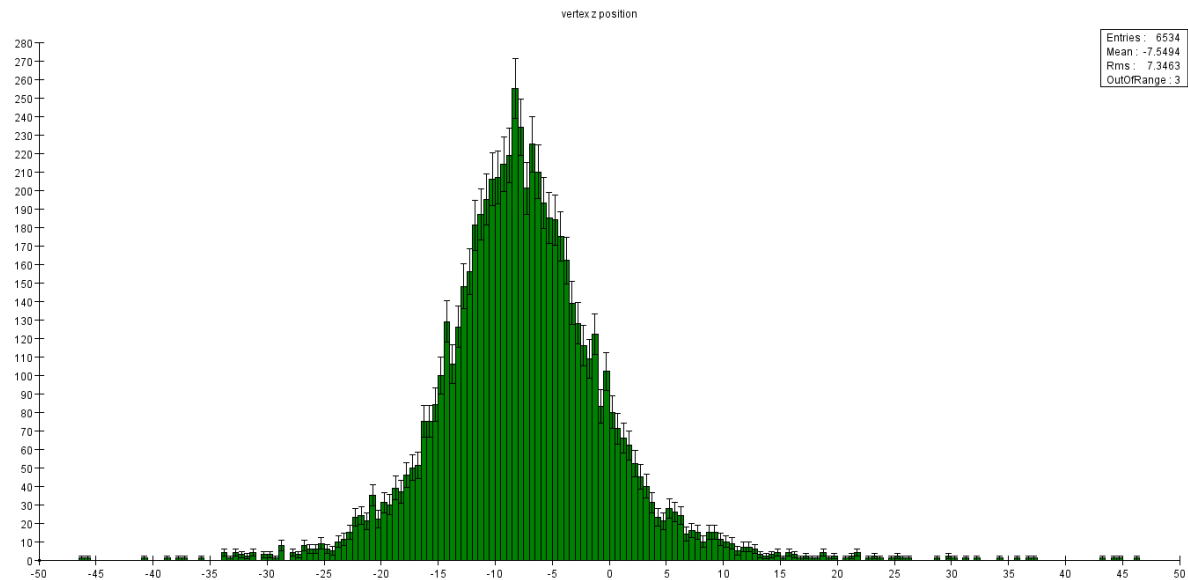
```
> java -cp distribution/target/hps-distribution-4.1-SNAPSHOT-bin.jar org.hps.evio.EvioToLcio -x /org/hps/steering/recon/PhysicsRun2016FullRecon.lcsim -r -d HPS-PhysicsRun
```

```
Then repeat with the jar file from today's git master, run analysis over the V0 collection and compare the unconstrained vertex z position. (
```

Results



Same detector,
Old code (top)
New Code (bottom)



Analysis

- So it appears that somewhere along the line an error in the vertexing code might have been introduced which leads to both the shift in position and degradation in resolution.
- Analysis of MC, where we know both the detector and the target position exactly, should confirm this.
- Will have to sort through git commits to do a traceback.