VXD Hit Recovery for Belle 2

lan J. Watson

University of Tokyo

B2 Tracking Meeting March 20, 2015

(Reminder) Project Goals



Example: pion from K_S decays between layer 3 & 4, not found by VXD track finder

- Current tracking requires particle can be found stand-alone in VXD for VXD hits to be used
- But, e.g., K_S can decay inside VXD, leaving some VXD hits without enough to find as a standalone track
- Idea for this project is to create a module to take CDC-only tracks and extrapolate back into VXD, create new track adding compatible VXD hits

Intersect with Cone

- Previously, I was extrapolating to cylinders with radius of each detector layer and looking for hits
- Investigated the genfit extrapolation code for the bent modules
- For genfit::RKTrackRep::extrapolateToCylinder, the code finds the straight line intersection from the particle to the cylinder then propagates until the particle intersects with the tangent plane at that point, then reruns this until the particle is on the surface
- Made an extrapolateToCone, taking the extrapolateToCylinder code and changing to intersecting with a cone, adjusting the output planes to work with the conical geometry
- Then, for layers with bent modules, if the particle is past cylindrical modules in z, extrapolate to cone with opening angle = bent layer module angle, positioned along z-axis so that cone radius = cylinder radius at the bent module interface
- From looking through debug statements, this seems to give me good enough results to move on with

Numbers of tracks, $B^{\pm} \rightarrow DK^{\pm}$, $D^0 \rightarrow K_S \pi^+ \pi^-$



- Right gives the fraction of tracks fitted in CDC-only with recoverable VXD hits versus MC particle production radius
 - Defined by having the track with CDC-only hits with a linked to MC particle with VXD hits
- Right gives the fraction of tracks fitted in CDC-only with recoverable VXD hits
- So, roughly 20% of tracks found fall into the "recoverable" category I'm interested in, with a large turn on versus radius
- N.b. no break down in sources, no test of whether the hits are sensibly "recoverable" given the CDC-only track from the standard B2 tracking

Numbers of tracks with recovered VXD hits



- Left shows the number of tracks in the 1000 event B[±] → DK[±] sample versus particle p_T, and the number reconstructed as CDC-only but with MC particles with VXD hits, and the number of such tracks my algorithm recovers
- Right also looks at VXD-recoverable tracks, but also the number of CDC-only tracks without MCParticle VXD hits, and the number of tracks succesfully refit
- Need to further explore why genfit fails to refit half these tracks

Further Breakdown of track numbers



- Looking at the "correctly recovered" tracks, i.e. CDC-only tracks with VXD hits associated to MC particle
- Left shows number, right shows fraction of tracks that have recovered (compared with the information from the matched MC particle)
 - Too few hits
 - Too many hits (N.b. not looking at if they're the "right" hits)
 - Exactly the number of hits
- As production radius increases, more false hits recovered
 - Need to look at tuning the hit search window (module area currently)

From basf2 output at the end of a 100 event run with my module "AddVXDHitsToCDCTracks"

Name	Calls	Mem(MB)	Time(s)	Time(ms)/Call
SetupGenfitExtrapolation	100	0	0.00	0.02 +- 0.01
Trasan	100	0	96.23	962.26 +- 930.49
VXDTF	100	0	1.19	11.93 +- 21.11
MCTrackCandCombiner	100	0	3.36	33.62 +- 7.91
GenFitter	100	2	60.38	603.84 +- 170.40
AddVXDHitsToCDCTracks	100	2	16.80	167.97 +- 121.24
V0Finder	100	2	5.52	55.22 +- 29.78

- Takes a significant fraction of the total tracking time
- But, as shown from first slide, we are extraplating and refitting on ~20% of tracks

- Investigate memory leaks
 - Valgrind or similar possible?
- Tune hit search cuts
 - Improving this can improve refit rate?
- If not, investigate genfit fit failures
- Large time and time variance
 - Any way to cut down on total time taken?

BACKUP

Plots in d0



Plots in p_T

