CKF VXD Hit Recovery for Belle 2

Ian J. Watson

University of Tokyo

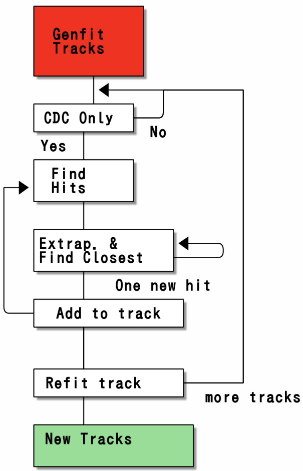
B2 Tracking Meeting
June 24, 2015

Initial Remarks

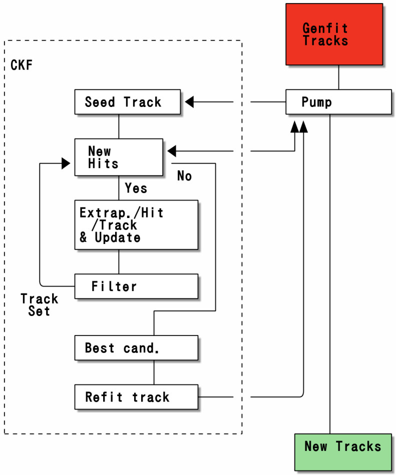
- Thinking on the CKF, produced a simple version following Rudi's points from last week
- Keep the implementation simple, proof of concept to see where the pain points/subtleties are
- As much as possible reusing the code from the VXD extrapolator, ripped out and rearranged
- Code in:
  - tracking/modules/cdcToVXDExtrapolator/{src,include}/CKF.{h,cc}
  - . . . /{src,include}/CKFCdcToVxdModule.{h,cc}

Overview

VXD Extrapolator



CKF

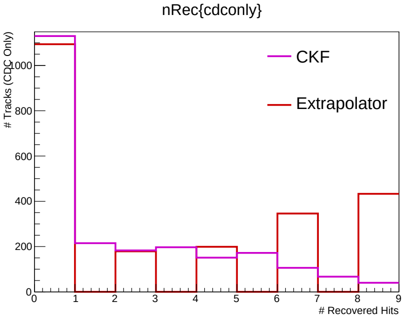


CKF Loop (with details removed)

```
genfit::Track* CKF::processTrack() {
  auto tracks = new std::vector<genfit::Track*>;
  tracks->push_back(new genfit::Track(*seedTrack));
  for (step = 0; true; ++step) {
    // take hits seed track because we assume we have a very wide window.
    std::vector<genfit::AbsMeasurement*> newHits;
    if (!findHits(seedTrack, step, newHits, data)) break;
    std::vector<genfit::Track*>* newtracks = new std::vector<genfit::Track*>;
    for (auto& track : *tracks) {
      // copy the no-added-hit case
      newtracks->push_back(track);
      /// if the track is bad (eg reached nholes limit), don't try to add hits
      if (!passPreUpdateTrim(track, step)) continue;
      for (auto& hit : newHits) {
        genfit::Track* newtrack = new genfit::Track(*track);
        newtrack->insertMeasurement(hit->clone, 0);
        fitter->processTrackPartially(newtrack, newtrack->getCardinalRep(), 1, 0);
        // check track passes quality cuts (e.g. local chi2) after update
        if (passPostUpdateTrim(newtrack, step))
          newtracks->push_back(newtrack);
      }
    }
  }
  // bestTrack currently looks at sum(hit chi2 inc.) / sum(hit ndof)
  genfit::Track* best = bestTrack(tracks);
  if (ret = refitTrack(best)) return ret;
  else return seedTrack;
}
```

- bool (*findHits)(Track*, unsigned, vector<AbsMeasurement*>&, void*) and seedTrack passed into CKF object via constructor

Comparison with Extrapolator



- Number of recovered hits
- Purity of hits about equivalent (96.7% for CKF, 95.7% for Extrapolator)
- No escape hatch for extrapolator, if it finds compatible hits, it adds on that layer, no competition against tracks without added hits
- Probably need a better best track metric
  - Currently $\sum_{hits} \Delta\chi^2 / \sum_{hits} NDOF$ for all hits on the track (not just added ones, otherwise can't compare the no hit case)

Backup

CKF Interface

Leads to something like the following interface

```
class CKF {
public:
  /**
   * Pass in an existing track to be fit, and a way to ask for new
   * hits. findHits should take in a track, and an integer
   * representing the number of steps already processed. data will be
   * passed unmolested to each invocation of findHits
   */
  CKF(genfit::Track* track,
      bool (*findHits)(genfit::Track*, unsigned, std::vector<genfit::AbsMeasurement*>&, void*),
      void* data, double _maxChi2Increment = 20, int _maxHoles = 3);
  /// find hits, run extrapolations, trim outputs, find a best track
  /// candidate
  genfit::Track* processTrack();
private:
  genfit::Track* seedTrack;
  /// refit track
  bool refitTrack(genfit::Track* track);
  /// finds the best track of all the candidates
  genfit::Track* bestTrack(std::vector<genfit::Track*>*);
  /// after adding an additional hit but before updating, check that
  /// the track passes some selections (e.g. check for holes)
  bool passPreUpdateTrim(genfit::Track*, unsigned step);
  /// after adding an additional hit and updating, check that the
  /// track passes some selections (e.g. check the updates chi2
  /// increment)
  bool passPostUpdateTrim(genfit::Track*, unsigned step);
};
```

`findHits` is relatively hacky, how best to represent this?

Timing

VXDExtrapolator			CKF		
Name	Time(s)	Time(ms)/Call	Name	Time(s)	Time(ms)/Call
RootInput	1.00	1.00 ± 2.98	RootInput	1.79	1.79 ± 13.36
Progress	0.02	0.02 ± 0.01	Progress	0.02	0.02 ± 0.01
Gearbox	0.01	0.01 ± 0.00	Gearbox	0.01	0.01 ± 0.00
Gearbox	0.01	0.01 ± 0.00	Gearbox	0.01	0.01 ± 0.00
Geometry	0.01	0.01 ± 0.00	Geometry	0.01	0.01 ± 0.00
SetupGenfitExtrapolation	0.01	0.01 ± 0.00	SetupGenfitExtrapolation	0.01	0.01 ± 0.00
Trasan	49.96	49.96 ± 95.06	Trasan	48.40	48.40 ± 91.69
VXDTF	1.19	1.19 ± 16.05	VXDTF	1.19	1.19 ± 16.10
MCTrackCandCombiner	4.78	4.78 ± 2.03	MCTrackCandCombiner	4.79	4.79 ± 2.02
GenFitter	81.46	81.46 ±110.95	GenFitter	82.27	82.27 ±111.76
CKFCdcToVxd	121.51	121.51 ±200.14	CDCToVXDExtrapolator	40.81	40.81 ± 81.39
TrackBuilder	2.57	2.57 ± 2.21	TrackBuilder	2.70	2.70 ± 8.16
RootOutput	9.53	9.53 ± 47.31	RootOutput	9.53	9.53 ± 47.19
Progress	0.02	0.02 ± 0.01	Progress	0.02	0.02 ± 0.01
Total	272.57	272.30 ±295.53	Total	192.00	191.81 ±227.04

- Timing for K_S particle gun sample (debug mode)
- Currently, CKF takes 3x extrapolator
  - Can't cache extrapolation to detector surface (each track path will be slightly different; done in `processTrackPartially`)
  - Copying entire track for every new path

Timing KsPiPi Events

Name	Time(s)	Time(ms)/Call	Name	Time(s)	Time(ms)/Call
RootInput	92.66	92.57 ±147.39	RootInput	93.23	93.13 ±156.86
Progress	0.02	0.02 ± 0.03	Progress	0.02	0.02 ± 0.01
Gearbox	0.02	0.02 ± 0.00	Gearbox	0.02	0.02 ± 0.00
Geometry	0.01	0.01 ± 0.00	Geometry	0.01	0.01 ± 0.00
SetupGenfitExtrapolation	0.01	0.01 ± 0.00	SetupGenfitExtrapolation	0.01	0.01 ± 0.00
Trasan	865.91	865.91 ±614.56	Trasan	860.24	860.24 ±607.79
VXDTF	149.03	149.03 ±481.93	VXDTF	144.45	144.45 ±470.21
MCTrackCandCombiner	26.83	26.83 ± 4.89	MCTrackCandCombiner	27.15	27.15 ± 4.86
GenFitter	629.35	629.35 ±336.04	GenFitter	611.51	611.51 ±259.79
CKFCdcToVxd	616.33	616.33+-1354.09	CDCToVXDExtrapolator	90.41	90.41 ± 92.80
TrackBuilder	8.86	8.86 ± 3.48	TrackBuilder	8.13	8.13 ± 2.99
V0Finder	31.99	31.99 ± 18.85	V0Finder	30.08	30.08 ± 15.96
Progress	0.02	0.02 ± 0.03	Progress	0.02	0.02 ± 0.01
Total	2421.87	2419.45+-1736.98	Total	1865.99	1864.13 ±935.65

- Timing information for 1000 KsPiPi events
- CKF 6x the CDCToVXDExtrapolator
    - Actually, CDCToVXDExtrapolator has the same timing as before, its doing approx. the same work as in the gun case
    - CKF settings: local $\chi^2 < 10$, 3 holes max; Extrap. settings, 5σ cutoff