VXDTF 2 development

T. Lück, E. Paoloni, V. Bertacchi

December 16, 2016

outline

- renamings
- bugs
- bugs
- more bugs
- outline (future bugs)

- renamed SectorMap.h
- it containes a map to VXDTFFilters for several setups:

```
std::unordered_map< std::string, Belle2::VXDTFFilters
  <point_t>*>;
```

- and some functions to access them renamed it to FiltersContainer
- so renamed it to: tracking/trackFindingVXD/filterMap/map/include/FiltersContainer.h
- that might not be the last renaming e.g. VXDTFFilters is also a bit misleading (as this is the actual sector map or better map of Sector to Filter)

The Problem

- Jonas and Felix reported crashes if the VXDTF2 is run in multi-thread mode ("basf2 -pX" with X¿1)
- segmentation violation while trying to delete a sector map related object
- basf2 will be multi-threaded at least on the HLT
- buy more equipment to run in single thread mode!?!?

The Solution

- we decided not to bye more equipment
- Eugenio proposed that the problem might be related to the fact that we store the sector map in the DataStore which is streamed when the process is forked
- Ansatz: remove the SectorMap from the DataStore and make it a Singleton

solve the segmentation violation for multithreading

- made it a singleton:
- only way to access (and create the one and only object) is through a static function
- deletion of the object is taken care of by the std::unique_ptr

```
/// one and only way to access the singleton object
static FiltersContainer & getInstance()
{
  /// the unique_ptr takes care for the deletion of
  static std::unique_ptr<FiltersContainer> instance
  return * instance;
}
```

- that seems to fix the problem (it compiles, tested up to 30k evt and up to "-p4")

non decaying B - mesons

- in the scripts inherited from Jakob to create root files to train VXDTF2
- if we chose to create Y(4S) events the resulting B - mesons will not decay
- problematic line (I think, still have to test) in: tracking/vxdCaTracking/extendedExamples/scripts/VXDTF/setup_modul

```
if allowDecay:
  uiCommandList.append('/process/inactivate
....
g4sim.param('UICommands', uiCommandList)
```

- allowDecay is true by default, so we trained to track charged B mesons (but we did with almost 100%)
- Felix and Jonas have fixed that in their commit

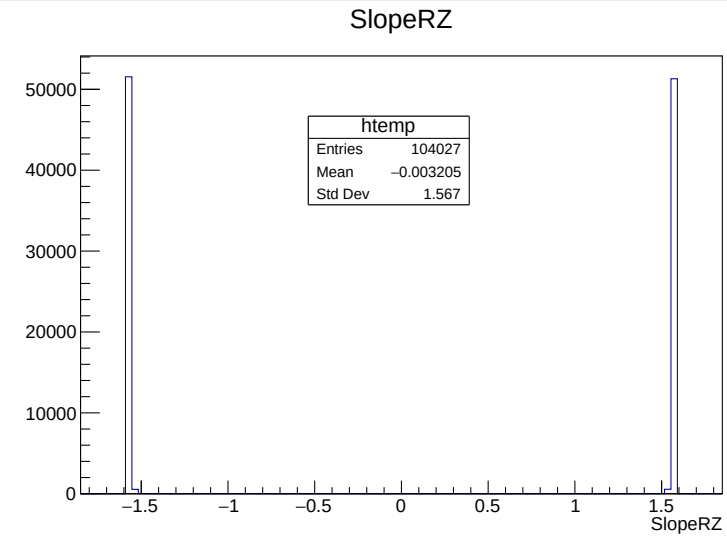
- bug in SlopeRZ.h filter: filter for the slope of a sector wrt. z-axis

```
double result = atan(sqrt(std::pow(double(outerHit.
```

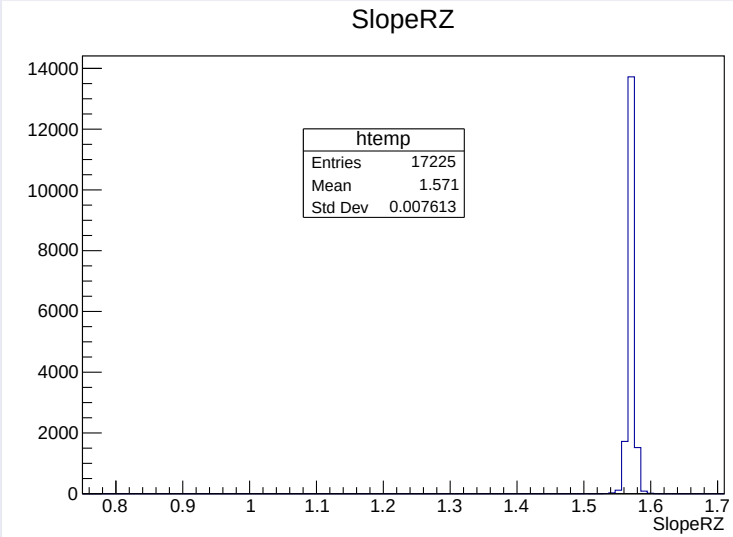
- problem atan also negative angles $\Rightarrow$ around 90 deg produce two peaks at Pi/2 and -Pi/2 which makes it useless for filters as we select $min < X < max$ occured during training
- my solution:

```
if( result < 0.0 ) result += M_PI;
```

with bug



bug fixed



this WILL break old trainings of the Filters!!

- ./tracking/modules/spacePointCreator/src/SPTCRefereeModule.cc
- fixed "bug": there seem to be cases for wich TrackFinderMCTruth or SpacePointCreatorSVD or SpacePointCreatorPXD or SpacePoint2TrueHitConnector or GFTC2SPTCConverter (¡= suspect that one as the trackcand in mc has hits for the event which crashes) creates SpacePointTrackCand with 0 hits
- this was not correctly handled
- added additional checks on that case
- another question not looked into: "why?" (Thomas Madlener)

- tracking/modules/vxdtfRedesign/src/SectorMapBootstrapModule.cc
- so far hard coded number of layers, sensors, and ladders,
- changed to get the list of sensors from the geometry
- this might break old trainings (though should not) as it puts the sector id into the framework

outline

- currently working on letting VXDTF2 run on testbeam (beast) geometry
- further renamings