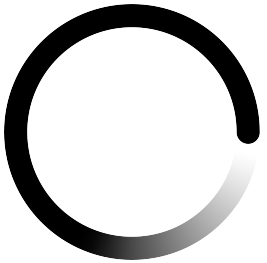
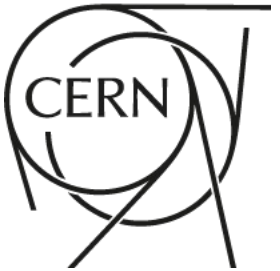
IDEA Drift Chamber in DD4hep

Readout implementation

Brieuc François (CERN)
March. 20th, 2023



FUTURE
CIRCULAR
COLLIDER



Detector Segmentation

- Detector segmentation: assigns an ID to G4Steps
  - Defines the detector 'cells': CellID uniquely identifies a sensitive wire
  - Currently not used in DriftChamber.cpp
  - Example from SimplifiedDriftChamber

```
<!--  Definition of the readout segmentation/definition  -->
<readouts>
  <readout name="SimplifiedDriftChamberCollection">
    <segmentation type="GridSimplifiedDriftChamber" inner_radius="innerRadius" cell_size="12*mm" detector_length="detectorLength" offset_phi="0" identifier_phi="phi"/
    <id>${GlobalTrackerReadoutID_DCH}</id>
  </readout>
</readouts>
```

```
<constant name="GlobalTrackerReadoutID_DCH" type="string" value="system:1,layer:16,phi:16"/>
```

  - CLD Silicon tracker:

```
<constant name="GlobalTrackerReadoutID" type="string" value="system:5,side:-2,layer:6,module:11,sensor:8"/>
```

```
<readouts>
  <readout name="CDCHHits">
    <id>innmodule:8,</id>
  </readout>
</readouts>
```

- **Task: implement a reliable segmentation for the DD4hep Idea Drift Chamber**
  - Requires a good knowledge of the detector geometry
  - Where to develop: DriftChamber.xml, DriftChamber.cpp
  - Potentially useful codebase: parametrised_DriftChamber.cpp, GridSimplifiedDriftChamber

Detector Hits

Task: implement a new Sensitive detector action (ProcessHit) for drift chambers

- Each G4Step has to be 'processed' to keep relevant information
- Key4hep similar example: DetSensitive/SimpleDriftChamber::ProcessHits

Task: these hits have to be stored in the final collection (edm4hep::SimTrackerHit) at the end of each event with the 'saveOutput' method

  - Key4hep similar examples: SimG4SaveTrackerHits.cpp

Task: implement a digitizer converting edm4hep::SimTrackerHit into physical edm4hep::TrackerHit

- This is where smearing, time window selection, etc, should occur (in my opinion)
- Strategy: implement first what is done in the standalone drift chamber simulation
  - Codebase: GMCTReadMCDataCDCH.cpp
- No Key4hep example available that I could find...
  - ILCSoft example (silicon tracker)
- This part should live in a new key4hep repository dedicated to tracker reconstruction that we will set up

Vertex Detector Reconstruction

- Two solutions for the vertex detector reconstruction code
  - Use what is available in ILCSoft through k4MarlinWrapper
    - Data conversion can be quite painful
  - Implement an edm4hep native solution
    - Preferred choice
- Possibility to host the Vertex detector reconstruction code in the same key4hep repository as the one for the drift chamber
  - Could also host tracking related tools
  - Name: k4RecTracker (too generic?), k4RecIdeaTracker (too specific?),

Summary

- Tasks (quite sequential)
  - Implement detector segmentation
  - Implement G4Step processing into detector hits (per hit)
  - Implement the method saving the output into edm4hep collection (per event)
  - Write a digitizer transforming simTrackerHit into TrackerHit
- How to share the load?
  - One task ↔ one person or common effort? (most tasks require both Key4hep and Drift Chamber expertise)
- Open questions
  - A single wire signal has ambiguity (right/left), how/where to lift this ambiguity?
  - Detector provide wire position and shortest distance to the wires?
  - How to determine the z position of the hit?

Additional material