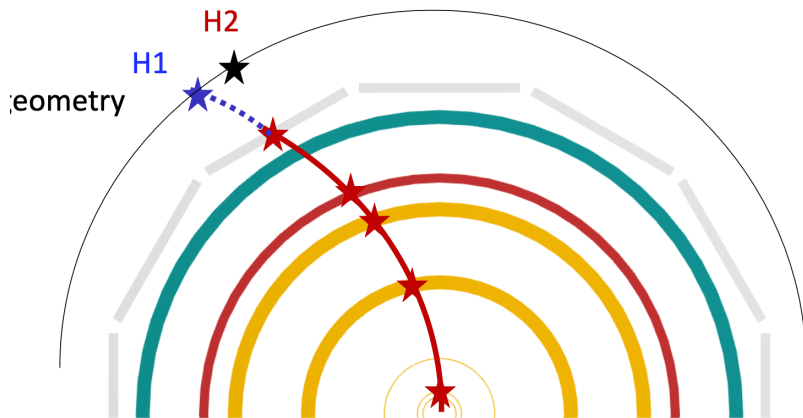


EicRoot tracking-for-PID implementation

EicRoot YR GitHub repository: <https://github.com/eic/EicRoot>

```
MacBook-Pro-105:~ ayk$ docker run -i -p 127.0.0.1:8888:22 -t -v /Users/ayk/DOCKER/eicroot-yr:/scratch ayk1964/eicroot-yr:v06
[root@cldadef2e2c5 /]# cd /container/app/EicRoot/examples/tracking/
[root@cldadef2e2c5 tracking]# ls -l
total 48
-rw-r--r-- 1 eic eic 158 May 21 2020 README
-rw-r--r-- 1 eic eic 3201 May 21 2020 analysis-vtx.C
-rw-r--r-- 1 eic eic 2476 May 21 2020 analysis.C
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.1
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.1a
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.1b
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.2
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.3
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.4
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.77
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.88
drwxr-xr-x 2 eic eic 4096 May 21 2020 config.999
```



From Matt's presentation, to facilitate a discussion

- The one we are interested in does not really work [can be restored if needed, obviously]

EicRoot tracking-for-PID implementation

- Geometry implementation
 - Easy to create (and add objects like profiling sensitive planes, cylinders & such)
 - Tracks will produce fake hits at these locations which automatically enter the Kalman filter fit
 - Truly modular (subdetectors are added one by one in a simulation.C script on the fly)
 - Written as a whole TGeo tree in a simulation.root output file (no need to maintain a separate material database, and no condensing of material at some reference planes is required)
- Digitization
 - Any sensitive volume can be digitized (no reference planes whatsoever in the first place)
 - Several strip, pixel (, 3D) templates available with their own cov.matrix description
- Track reconstruction (*no track finder*; Kalman filter & smoother pass over known hits)
 - Truth and reconstructed (smoothed) state vector parameters can be made available at every node:

```
TVector3 mcpos = par->GetMoCaPosition(), mcmom = par->GetMoCaMomentum();
TVector3 rcpos = par->GetRecoPosition(), rcmom = par->GetRecoMomentum();
printf("Hit-to-plane distance: %7.2f [cm]\n", par->DistanceToPlane(plxx, plnx));
printf("  (REC) -> V: %7.3f %7.3f %7.3f [cm] & P: %7.3f %7.3f %7.3f [GeV/c]\n",
       rcpos.X(), rcpos.Y(), rcpos.Z(), rcmom.X(), rcmom.Y(), rcmom.Z());
printf("  (MC)   %7.3f %7.3f %7.3f      & %7.3f %7.3f %7.3f\n\n",
       mcpos.X(), mcpos.Y(), mcpos.Z(), mcmom.X(), mcmom.Y(), mcmom.Z());
```