FD Pair Generation

Alex Wilkinson

20 October 2023

ND Sim

▶ Starting from random throws + rotations root file

▶ `dumpTree` script (here) converts to HDF5 with datasets:

(fd_deps, segments, fd_vertices, vertices)

▶ Pass through larnd-sim (using tag v0.3.4), adds to the HDF5 file the datasets:

(_header, configs, mc_packets_assn, messages, packets, segments $\rightarrow$ tracks)

ND Sim

► To get 3d positions from packets can use larpixsoft like:

```python
import h5py

from larpixsoft.detector import set_detector_properties
from larpixsoft.geometry import get_geom_map
from larpixsoft.funcs import get_events_no_cuts

detector = set_detector_properties(DET_PROPS_FILE, PIXEL_LAYOUT_FILE, pedestal=74)
geometry = get_geom_map(PIXEL_LAYOUT_FILE)

f = h5py.File(FILE_PATH)

packets = get_events_no_cuts(
    f["packets"], f["mc_packets_assn"], f["tracks"], geometry, detector, no_tracks=True
)

# Event loop
for event_packets in packets:
    # Packet loop
    for p in event_packets:
        x = p.x + p.anode.tpc_x
        y = p.y + p.anode.tpc_y
        z = p.z_global()
        adc = p.adc
```

FD Sim

► Everything done with duneextrapolation larsoft package

► Setting `physics.producers.largeant.NDFDH5FileLoc` in `run_LoadFDDepos.fcl` to the ND sim HDF5 file and running loads "fd_deps" into art-root `sim::SimEnergyDepositions`

► Simulation fcls: ionisation, detector simulation, reconstruction

► Setting `physics.analyzers.addreco.NDH5FileLoc` in `run_AddFDReco.fcl` to the ND sim HDF5 file writes FD reco (CVN scores, energy reconstruction) for each event into the file — adds fd_reco dataset to HDF5 file

► Working on module that writes out to the HDF5 file FD detector response + ND packets projected to FD wires

Plans

▶ Once we have made the new paired data will consolidate this code into a single repository with instructions for setting up and running the different parts

▶ Will include scripts for doing most of these steps automatically (not possible to do all at once since larnd-sim uses GPU)