Databases and ProtoDUNE

Tingjun Yang

Database Group Meeting
July 6, 2022

Introduction

- At the last meeting Leigh gave a very complete talk on what information we currently retrieve from database for offline reconstruction/analysis
  - Electron lifetime (data)
  - YZ position calibration (data and MC)
  - X calibration (data and MC)
  - Normalisation calibration (data)
- I would like to share my thoughts on database related issues.

Useful information to save in a database

- Per-run information
  - Normalization calibration constant
  - Electron lifetime
  - HV
  - Temperature
  - Dead/bad channel list
- Per-event information
  - BI information
- Per-channel information
  - YZ calibration
  - X calibration
  - TPC electronics gain/shaping time
  - PD electronics gain

Currently implemented in database

Not implemented in database

Time stamp and validity time

- The time stamp can be either the real time (e-lifetime table) or a run number (other tables).
  - Using run number is preferred (easy to cache the database results).
  - Purity monitor runs were not tied to a particular TPC run. Therefore, the lifetime time stamp is the time when the purity monitor toke data.
    - Does it make sense to convert it to a TPC run number?
- How frequent do we need to save information like temperature and HV (validity time)?
  - Given usually a physics run lasts up to a few hours, it makes sense to save information per run.
    - If a run is short (a few minutes), we do not need to save a database entry for that run. One can use the information from the previous stable run.
  - Finer grained HV information is used for a data quality filter.

Dead/noise channels

- A list of dead and noise channels provide useful information for event reconstruction
  - Track reconstruction can skip bad channels and not break a track.
- Currently this information is saved in fcl files
  - [channelstatus_pdsp_2018.fcl]
  - [channelstatus_pdsp_nov2019.fcl]
- Does it make sense to save such information in a database?

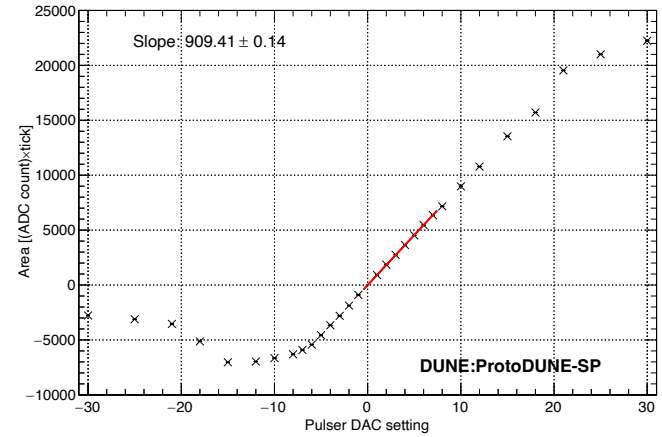
```
# channelstatus_pdsp_2081.fcl
#
# David Adams
# October 2018
#
# Bad and noisy channels for protDUNE single phase.

BEGIN_PROLOG

pdsp_channel_status_2018: {

  # implements IChannelStatusService
  service_provider: SimpleChannelStatusService

  # list of bad channels:
  BadChannels:   [
    # CE group: Inactive FE
     4411,  # femb515x12
     4412,  # femb515x13
     9990,  # femb605x10
    11842,  # femb120x03
    # CE group: Broken connection
        1,  # femb311u39
      400,  # femb301u40
      401,  # femb301u39
      800,  # femb320v01
      801,  # femb320v02
      876,  # femb319v37
     1200,  # femb310v01
     2961,  # femb501u39
     5321,  # femb216u39
     5363,  # femb217u37
     6132,  # femb215v13
     7058,  # femb213x03
     7295,  # femb202x01
```

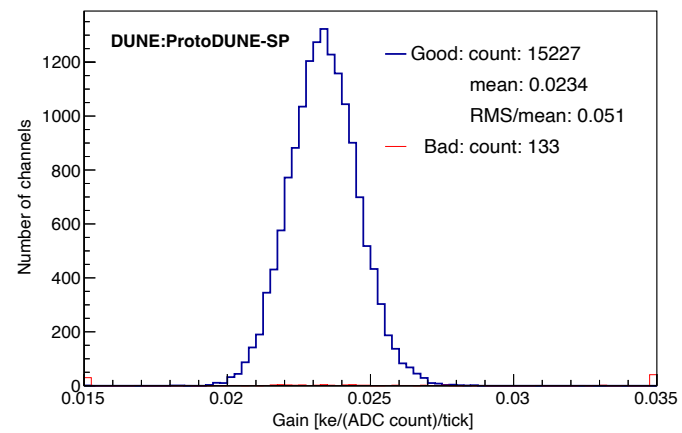
Electronics gain and shaping time

- ProtoDUNE uses a pulser system to measure the electronics gain and shaping time for each channel.
- Currently this information is also saved in fcl files:
  - https://github.com/DUNE/dunesw/tree/develop/fcl/protodune/fcldirs/calib/protodune

ls ./dunesw/fcl/protodune/fcldirs/calib/protodune/
calib20190205 calib20191203 calib20191206_dec18ext
calib20191213_dec18ext cbnov19

- I think it makes sense to save such information in a database.





Other thoughts

- It is hard to predict what information should be saved in a database.
  - Things like photon detector gains and CRT timing offsets are good candidates.
  - Leigh also mentioned new information from the laser system.
- It would be helpful to standardize the database format and interface.
  - Would it be possible to access database both in larsoft and in a standalone application?

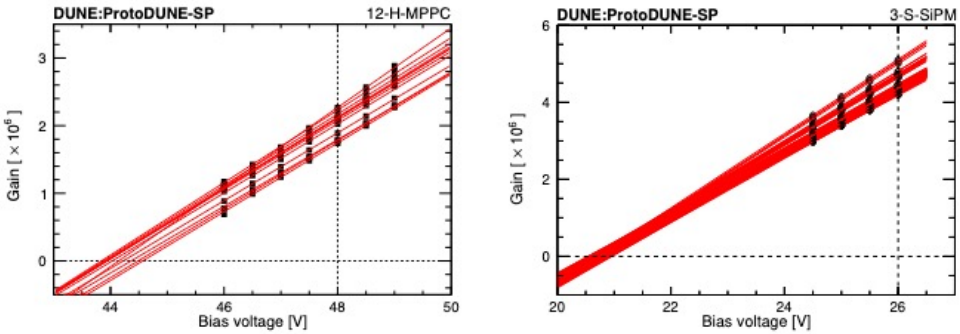


Figure 34. Gain as a function of applied bias voltage for 12-H-MPPC channels (left), and for 3-S-SiPM channels (right). Linearity of individual channel response is shown by the linear fit (red line) across the points at different bias voltage setting. The intercept of the fit line provides a direct evaluation of the breakdown voltage at LAr temperature for each 12-H-MPPC and 3-S-SiPM photosensor.