

Table of Contents

Welcome to the NP04/protoDUNE-SP Online Monitoring TWiki Home page.....	1
List of Metrics.....	1
Priority:.....	1
Desirable:.....	1
Online Monitoring on.....	1
Simulated Scenarios.....	2
Access and copy of the simulated data.....	2
Scenario Description.....	2
Performance monitoring via profiling tool.....	4
Output Root Files.....	5
Location.....	5
Convention.....	5
Web Tool Monitor (Marco).....	6

Welcome to the NP04/protoDUNE-SP Online Monitoring TWiki Home page

List of Metrics

Priority:

- Event Display
- Mean and RMS
- FFT
- Check for Stuck Bits
- display of beam region vs non-beam region
- Trigger ?
- Photon Detector ?

Desirable:

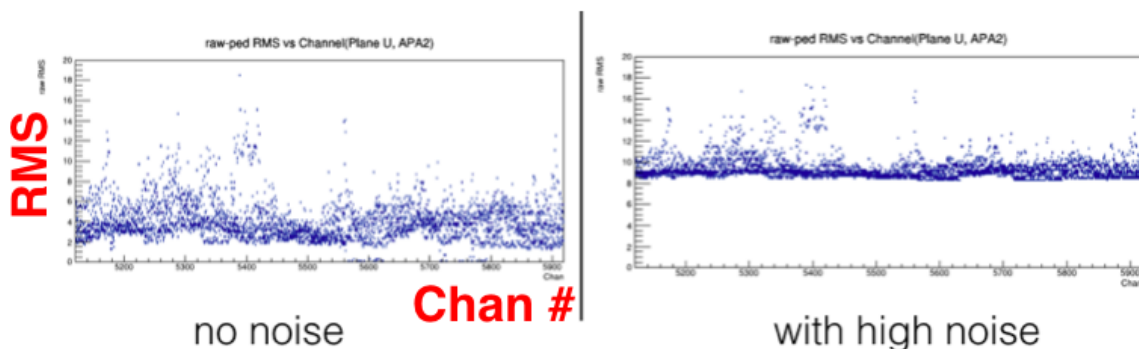
- Purity check
- Muon monitors

Online Monitoring on

Disclaimer: The source codes were inspired by the work stored in dune35t found in various modules in LArSoft.

- The source code is stored under `_/srcs/dunetpc/dune/Protodune/OnlineMonitor` and it contains multiple modules:
 - ◆ **DONE** `OnlineMonitor_module.cc`:
 - ◇ ⚠ the ranges are hardwired --> needs improvements
 - ◆ **DONE** `OnlineMonitorWithPedestal_module.cc`:
 - ◇ ⚠ --> same as above

Over 5 events:



RMS metrics for non-noisy vs noisy conditions

- ◆ **DONE** `RawEventDisplay_module.cc`
- ◆ **DONE** `FFT_module.cc`
- ◆ **DONE** `StuckBits_module.cc`
- ◆ ⚠ `PurityAna_module.cc`

- Channel Mapping:
 - ◆ loop through channels in the first APA to find the channel boundaries for each view
- Filling:
 - ◆ loop over all raw digits (entire channels) , look in each view (per chan) for each APA
 - ◆ Histos stored for each APA, each View

Simulated Scenarios

The OnlineMonitor metrics are tested and validated under various simulated scenarios. These simulated data are based on following chain: **gen** (particle ID, input geometry)-> **g4** (full MC truth, energy deposits) -> **detsim** (scintillation light and ionization electrons from energy deposits, recombination of ionization electrons, attenuation of electrons number, due to LAr impurities at given drift distance, longitudinal and transverse diffusion of electrons at given drift distance, added readout info: electronics response, noise, ADC waveforms on readout channels)

Various conditions of the noise were produced (by Dorota and Robert) and they are stored here: https://dune-data.fnal.gov/mc/mcc8/protodune_request1/index.html

Access and copy of the simulated data

For new people: follow up the instructions at:

<https://web.fnal.gov/collaboration/DUNE/SitePages/How%20to%20Map%20Your%20CERN%20Grid%20Cert%20to>

For those who have already certificate installed and want to copy from fermilab to neutplatform - example to use copy from

https://dune-data.fnal.gov/mc/mcc8/protodune_request1/xml/protoDune/ProtoDUNE_beam_2GeVc_cosmics_3ms_noise

- voms-proxy-init --voms dune:/dune/Role=Analysis -vomses ~/dunevomses
- xrscp
 - xroot://fndca1.fnal.gov/pnfs/fnal.gov/usr/dune/persistent/dunepro/v06_18_01_01/detsim/ProtoDUNE_beam_2/mnt/nas01/users/radescu/Feb2017_v22/inputs/.

Scenario Description

- standard detsim makes low, white noise, equivalent to 1.6 parameter for white noise in exp.noise version
- desim_noise has exponential noise on top of white, with still reasonable amplitudes
- detsim_noise_high is ~worst case.

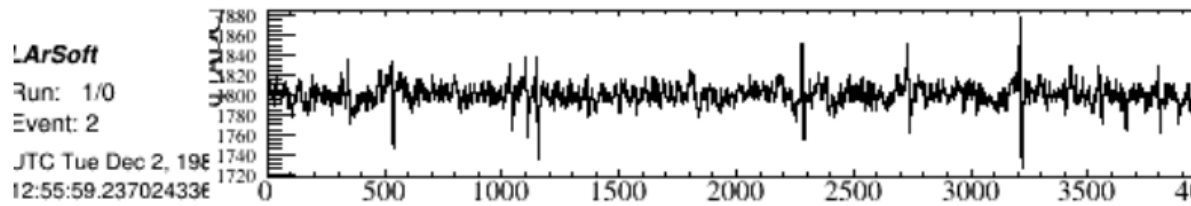
- - ◆ **low** noise (using protoDUNE_detsim.fcl)
 - ◆ **medium** noise (using protoDUNE_detsim_noise.fcl)

```
Channel noise service:
  ExponentialChannelNoiseService:
    NoiseNormZ: 10
    NoiseWidthZ: 150
    LowCutoffZ: 0
    NoiseNormU: 15
    NoiseWidthU: 150
    LowCutoffU: 0
    NoiseNormV: 15
    NoiseWidthV: 150
    LowCutoffV: 0
    NoiseArrayPoints: 1000
    OldNoiseIndex: 0
    WhiteNoiseZ: 1.6
    WhiteNoiseU: 1.6
    WhiteNoiseV: 1.6
    RandomSeed: 0
    LogLevel: 1
    Actual random seed: 324020508
```

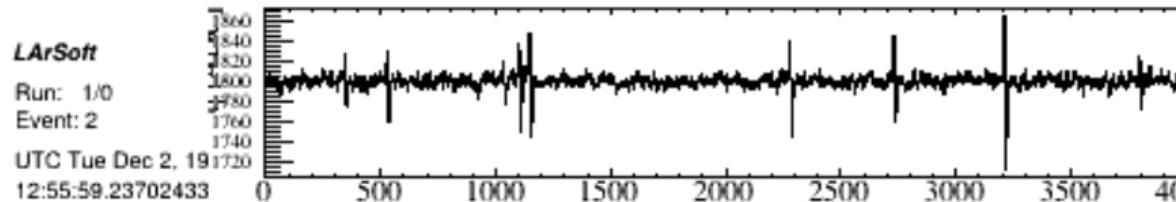
Medium Noise Scenario

- ◆ **high** noise - a la dune35t (using protoDUNE_detsim_noise_high.fcl)

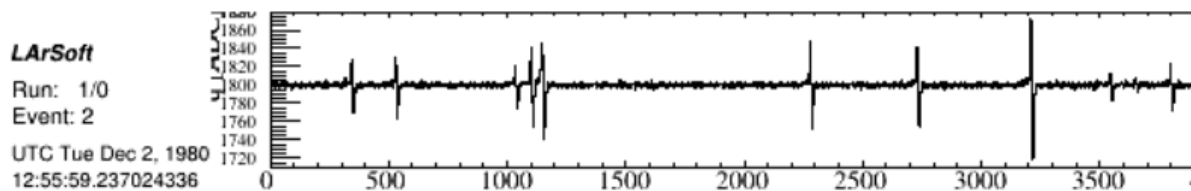
- **high noise:**



- **medium noise**



- **zero exp. noise**



Simulated Noise Scenarios

These noise definitions are based on the parameters used for the WhiteNoise (incoherent - flat noise) and for the NoiseNorm parameters (exponential noise). These parameters are defined in *.fcl* files in *srcs/dunetpc/fcl/protodune/detsim/*.

Moreover, since the Online Monitoring aims to address the sanity of the data quality, raw data is simulated with various assumptions of electronics: These flags exist in the default *.fcl* files: *srcs/dunetpc/fcl/protodune/detsim/protoDUNE_detsim.fcl*

- physics.producers.daq.NoiseOn:
- physics.producers.daq.PedestalOn:
- physics.producers.daq.DistortOn:
- physics.producers.daq.SuppressOn:

Which correspond to:

- with or without distortion applied
- with or without Suppression applied

Following parameters were used for these conditions if applied:

```

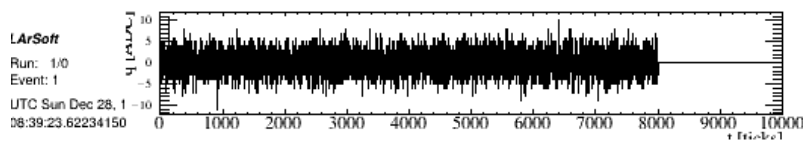
SimWireDUNE::reconfigure: Accessed services:
SimWireDUNE::reconfigure:   SimChannel extraction service:
SimWireDUNE::reconfigure:     GenericSimChannelExtractService
SimWireDUNE::reconfigure:   Channel noise service:
SimWireDUNE::reconfigure:     ExponentialChannelNoiseService:
SimWireDUNE::reconfigure:       NoiseNormZ: 3.16
SimWireDUNE::reconfigure:       NoiseWidthZ: 2000
SimWireDUNE::reconfigure:       LowCutoffZ: 7.5
SimWireDUNE::reconfigure:       NoiseNormU: 3.16
SimWireDUNE::reconfigure:       NoiseWidthU: 2000
SimWireDUNE::reconfigure:       LowCutoffU: 7.5
SimWireDUNE::reconfigure:       NoiseNormV: 3.16
SimWireDUNE::reconfigure:       NoiseWidthV: 2000
SimWireDUNE::reconfigure:       LowCutoffV: 7.5
SimWireDUNE::reconfigure:       NoiseArrayPoints: 1000
SimWireDUNE::reconfigure:       OldNoiseIndex: 1
SimWireDUNE::reconfigure:       WhiteNoiseZ: 0
SimWireDUNE::reconfigure:       WhiteNoiseU: 0
SimWireDUNE::reconfigure:       WhiteNoiseV: 0
SimWireDUNE::reconfigure:       RandomSeed: 0
SimWireDUNE::reconfigure:       LogLevel: 1
SimWireDUNE::reconfigure:       Actual random seed: 777512808
SimWireDUNE::reconfigure:   Pedestal addition service:
SimWireDUNE::reconfigure:     ProvidedPedestalAdditionService:
SimWireDUNE::reconfigure:       Noise scale: 0
SimWireDUNE::reconfigure:   ADC distortion service:
SimWireDUNE::reconfigure:     StuckBitAdcDistortionService:
SimWireDUNE::reconfigure:       StuckBitsProbabilitiesFname: ADCStuckCodeProbabilities35t/output_produceDcScanSummaryPlots_201
SimWireDUNE::reconfigure:       fStuckBitsOverflowProbHistoName: pCorrFracOverflowsInputLsbCell
SimWireDUNE::reconfigure:       fStuckBitsUnderflowProbHistoName: pCorrFracUnderflowsInputLsbCell

SimWireDUNE::reconfigure:   ADC suppression is off.
SimWireDUNE::reconfigure:   Compression service:

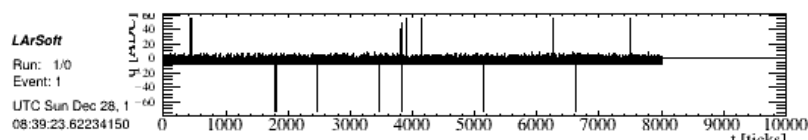
SimWireDUNE::reconfigure:     LarsoftHuffmanCompressServiceSimWireDUNE::reconfigure:   KeepEmptyChannels:0

```

Sample with Distortion and Noise as viewed by lar event display



No Distortion, no Suppression



Sample with Distortion, no Suppression as viewed by lar event display

Performance monitoring via profiling tool

A first attempt to monitor the profiling was performed used igprof tool. It has the advantage that it does not require re-configuration and it **should** work with the default LArSoft optimisation. This is now installed centrally on the neut platform cluster. To use it just run a setup:

- ♦ source /mnt/nas01/software/products/setups
- ♦ setup igprof v5_9_16 -q e10

However, the demangle does not really work and then it's hard to digest the performance

- - for performance:
 - ♦ igprof -d -pp -t lar -o igprof_lar.pp.gz lar -c ..
 - ♦ igprof-analyse -d -v -g igprof_lar.pp.gz > igreport_perf.res

- - for memory:
 - ◆ igprof -d -mp -t lar -o igprof_lar.gz lar -c ...
 - ◆ igprof-analyse -d -v -g -r MEM_TOTAL igprof_lar.gz > igreport_total.res

using debug:e10

Sorted by cumulative cost

(Sort by self cost)

Rank	Total %	Cumulative	Symbol name
1	100.00	0.47	<spontaneous>
6	96.81	0.46	TGDMLParse::GDMLReadFile(char const*)
5	96.81	0.46	@?0x351be0e7{<dynamically-generated>}
4	96.81	0.46	@?0x351be035{<dynamically-generated>}
3	96.81	0.46	cling::Interpreter::RunFunction(clang::FunctionDecl con
2	96.81	0.46	@?0x51a78e0{<dynamically-generated>}
9	76.60	0.36	TGDMLParse::ParseGDML(TXMLEngine*, void*)'3
8	76.60	0.36	TGDMLParse::ParseGDML(TXMLEngine*, void*)'2
7	76.60	0.36	TGDMLParse::ParseGDML(TXMLEngine*, void*)
10	61.70	0.29	TGDMLParse::VolProcess(TXMLEngine*, void*)
11	19.15	0.09	TGDMLParse::PosProcess(TXMLEngine*, void*, void*)
12	18.09	0.09	TXMLEngine::ParseFile(char const*, int)
17	17.02	0.08	std::Rb_tree<std::basic_string<char, std::char_traits<
16	17.02	0.08	TXMLEngine::ReadNode(void*, TXMLInputStream*, int&)'3
15	17.02	0.08	TXMLEngine::ReadNode(void*, TXMLInputStream*, int&)'2
14	17.02	0.08	TXMLEngine::ReadNode(void*, TXMLInputStream*, int&)

Counter: PERF_TICKS

Rank	% total	Counts		Paths		Symbol name
		to / from this	Total	Including child / parent	Total	
	96.81	0.46	0.46	1	1	@?0x51a78e0{<dynamically-generated>}
[3]	96.81	0.00	0.46	1	1	cling::Interpreter::RunFunction(clang::FunctionDecl
	96.81	0.46	0.46	1	1	@?0x351be035{<dynamically-generated>}

Igprof summary table in SQL web tool format

Output Root Files

The working platform for the online monitor is now on the daq machine: pddaq-gen05.

Location

The files will be stored there: /daq/onlinemonitor/data_histos/simulation .

Convention

The name convention of files is run_#_ev_#_time_#.root The time # is ...

⚠ The test outputs are stored for now in the public lplus area till Marco settles his accounts on neutrino cluster:

- ~radescu/public/forMarco/

Web Tool Monitor (Marco)

[Back to protoDUNE-SP Data Quality Twiki Main Page](#)

[Back to protoDUNE-SP Data Preparation Twiki Main Page](#)

Major updates:

-- NectarB - 2017-03-02 - created web site

-- VoicaAnaMariaRadescu - 2017-03-16 - created structure and content

This topic: CENF > DUNEProtSPOM

Topic revision: r9 - 2017-05-10 - VoicaAnaMariaRadescu



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback