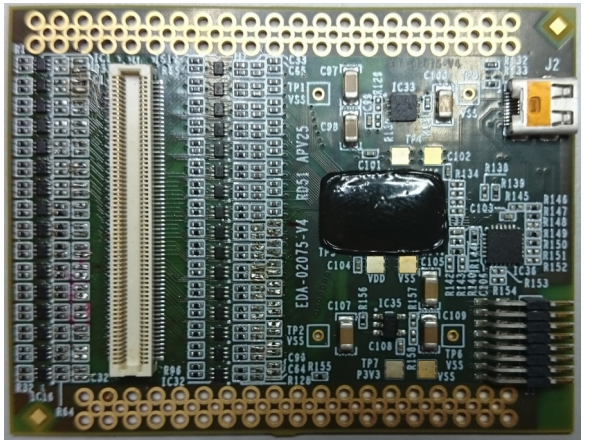
Processing TB2020 SRS raw data

Oleksandr Borysov
DESY

FCAL SW Meeting
February 17, 2021
(April 15, 2020)

APV 25

- 128 channels
- 192 samples analog pipeline
- 40 MHz sampling/RO frequency
- 50ns nominal shaping time
- ~ 2000 ENC @ 50pF
- 100mV / 25000 electrons



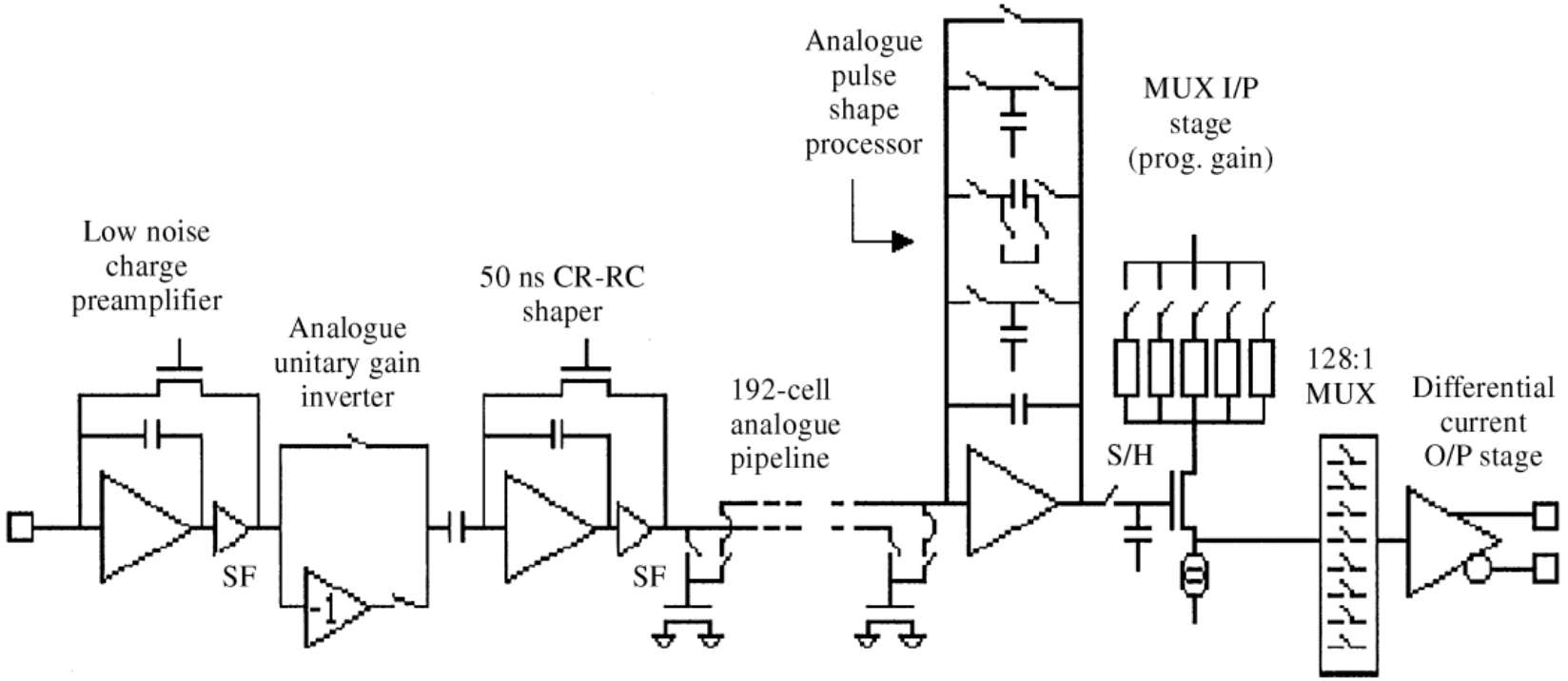
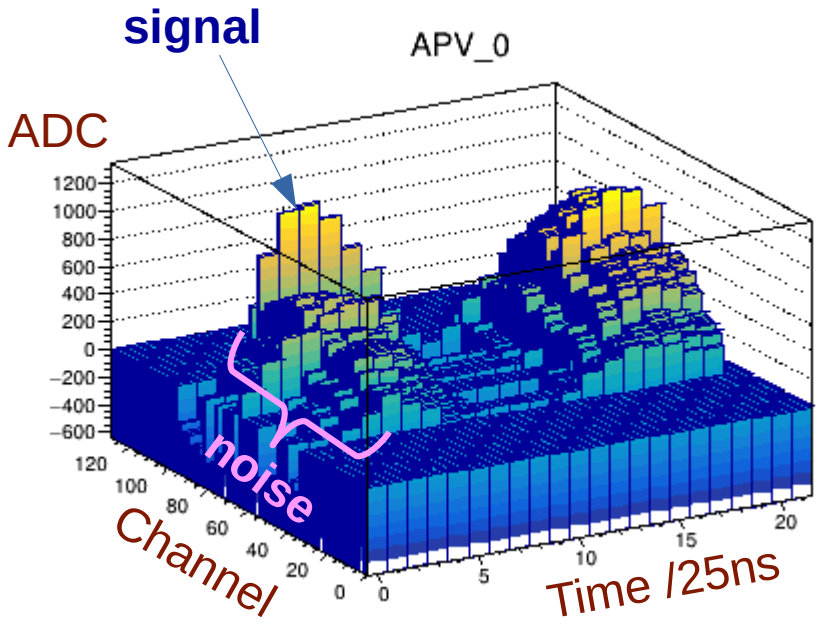


Fig. 1. Block diagram of one channel of the APV25.

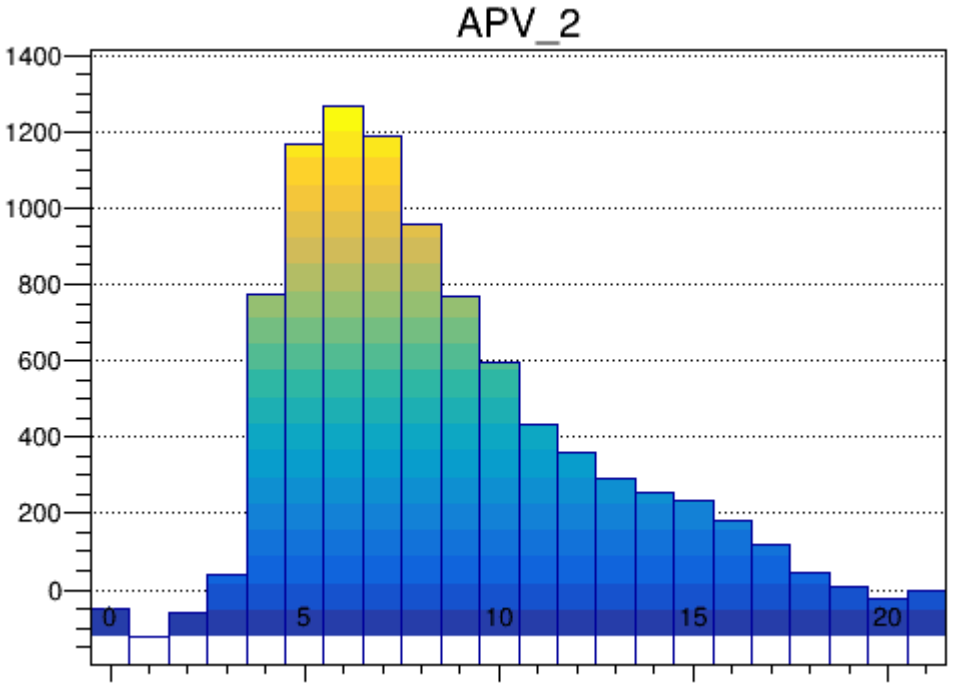
APV 25 sampled signal

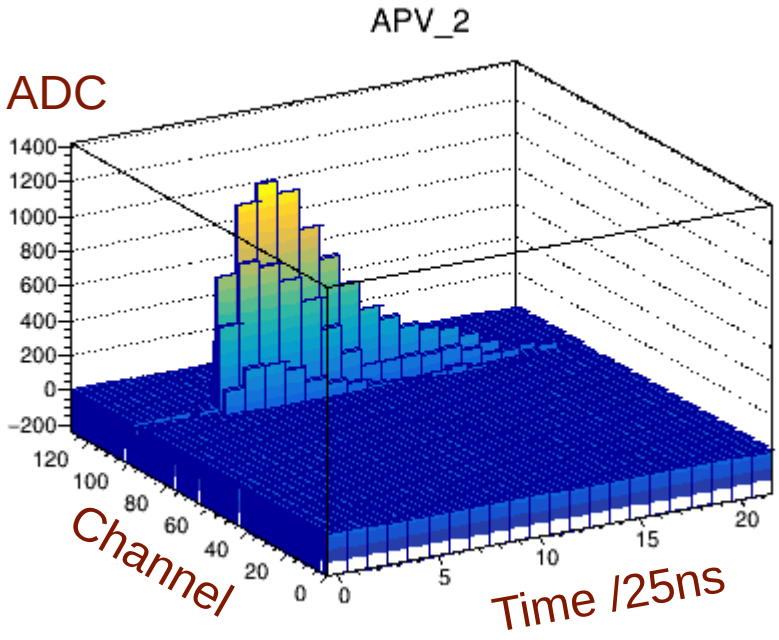
SRS Run 16
5 GeV,
Energy scan study, SRS + Telescope

Example plots:
APV 0,2 (Master)



Signal shape can be used to reject the noise



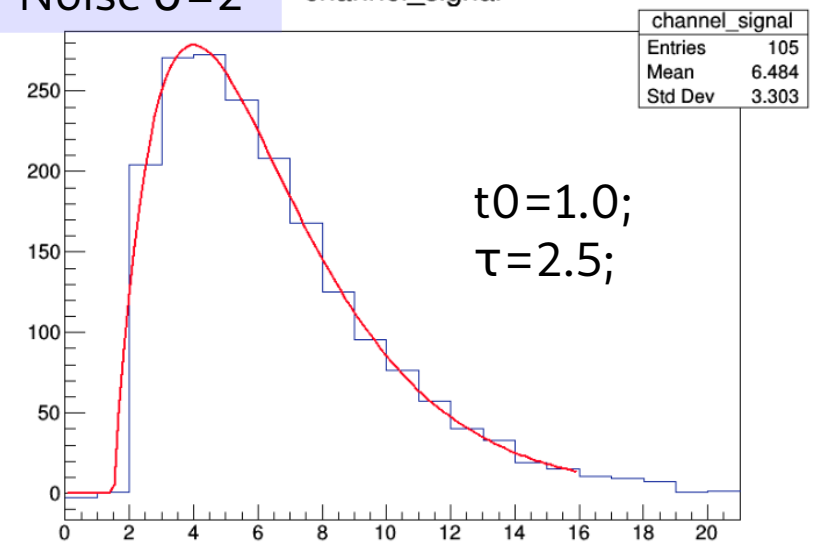


Signal Extraction Optimization

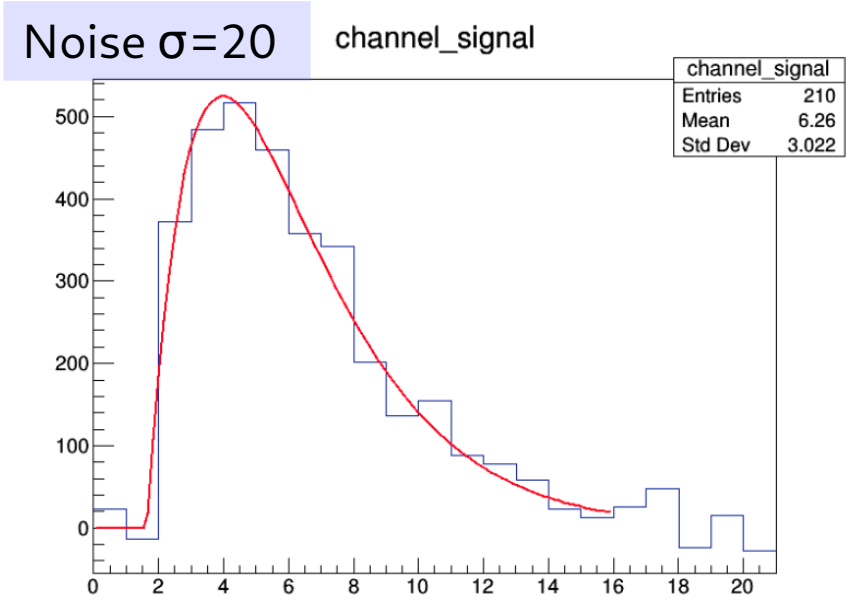
Fit signal with time response function of CR-RC filter:

$$S(t) = A\frac{t - t_0}{\tau}e^{-\frac{t-t_0}{\tau}}\Theta(t - t_0)$$

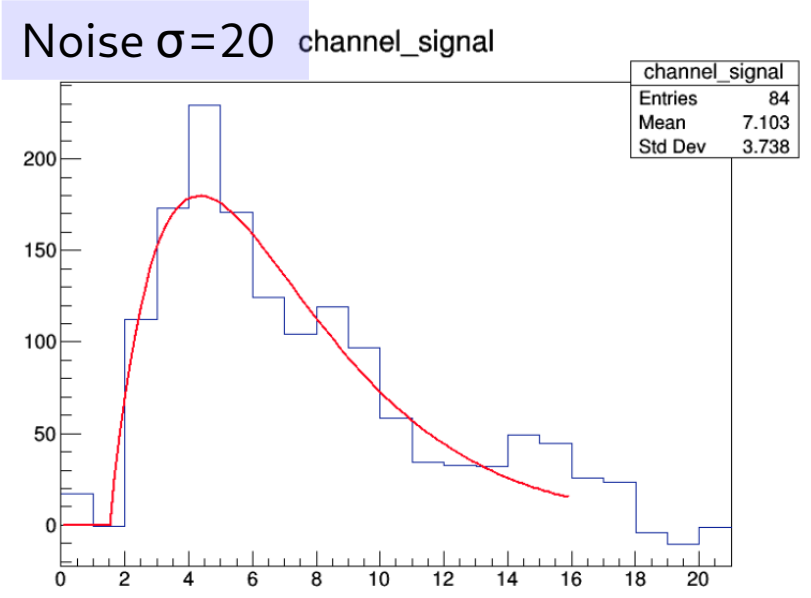
Noise σ=2



Noise σ=20



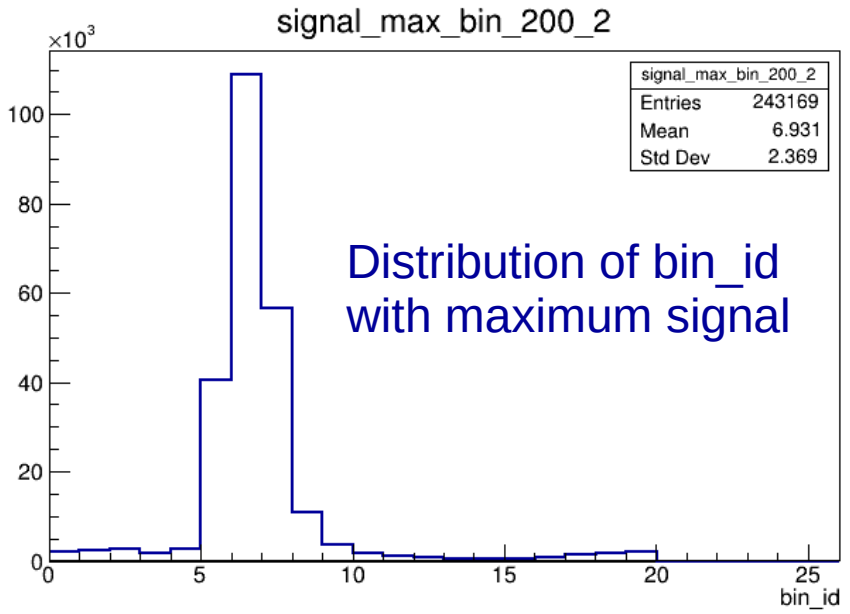
Noise σ=20

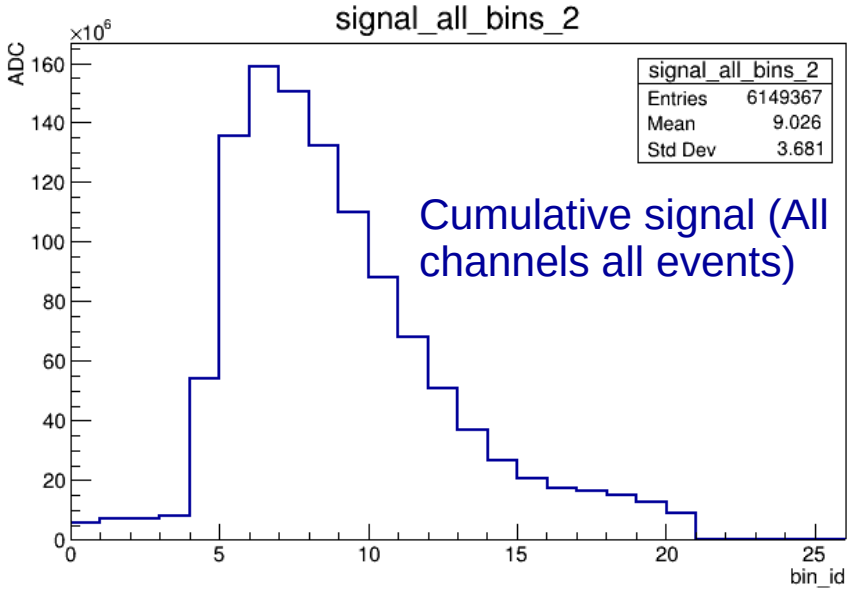


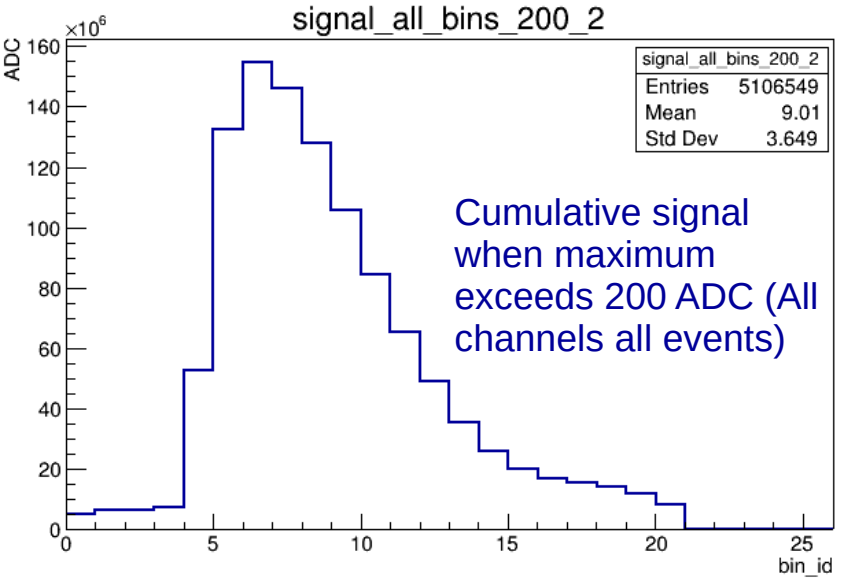
Time bin with signal maximum

SRS Run 16
5 GeV,
Energy scan study, SRS + Telescope

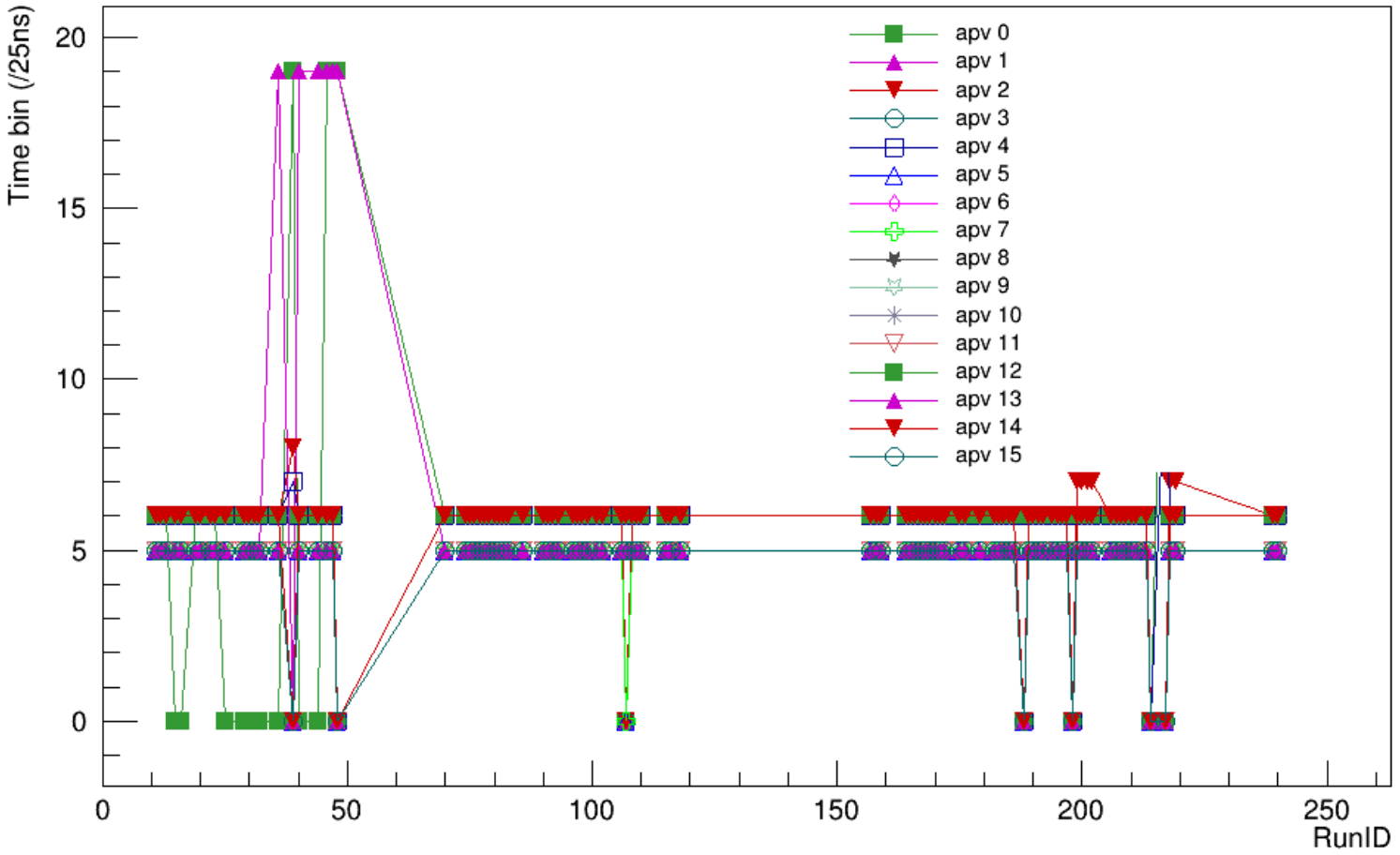
Example plots:
APV 2 (Master)







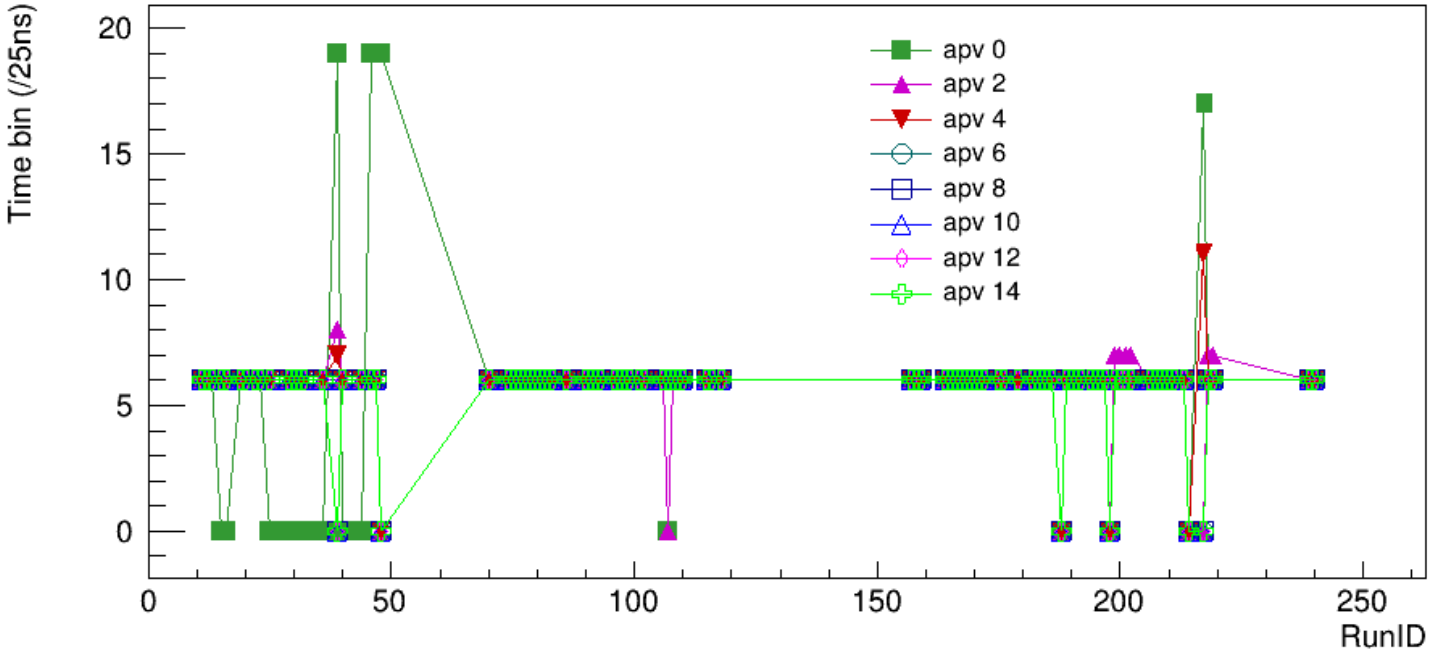
Position of the signal maximum

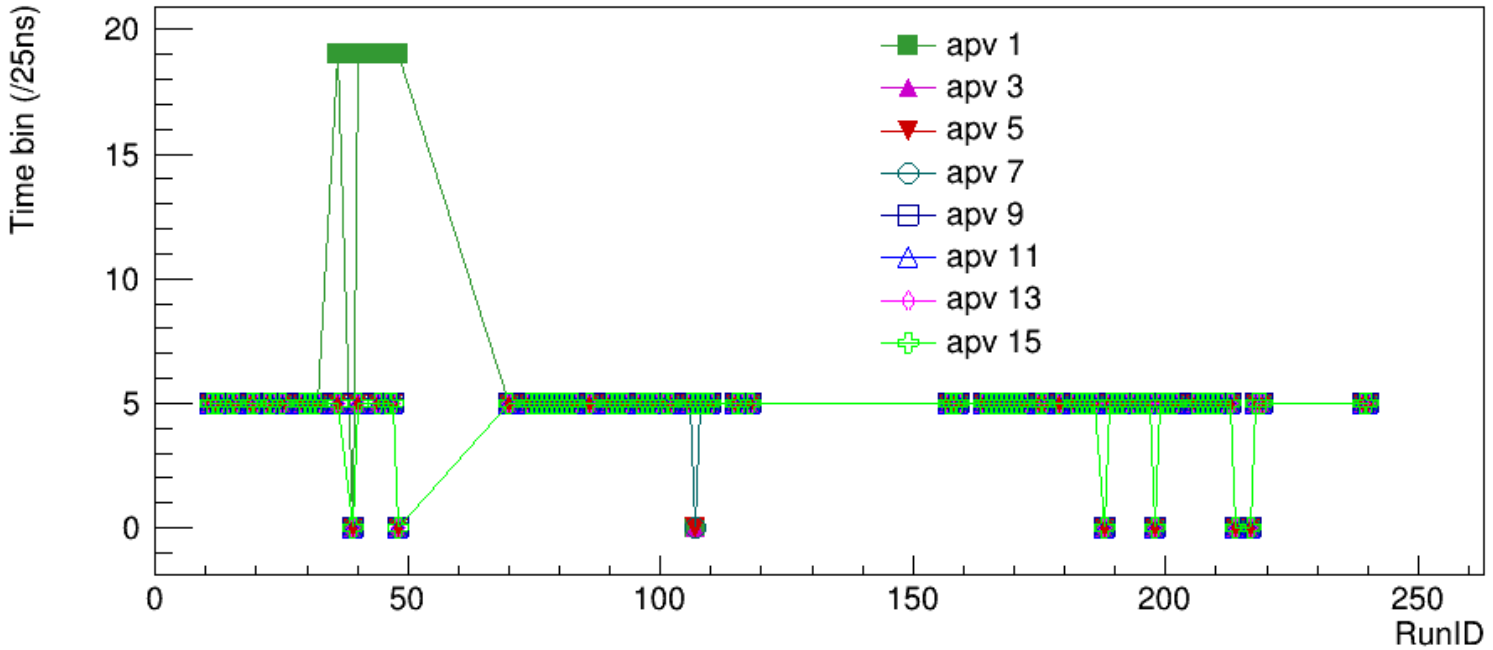


Position of the signal maximum

Max(max_bin(signal_array))

- Even APVs, (masters) have signal maximum in time bin 6;
- Odd APVs, (slaves) have signal maximum in time bin 5;

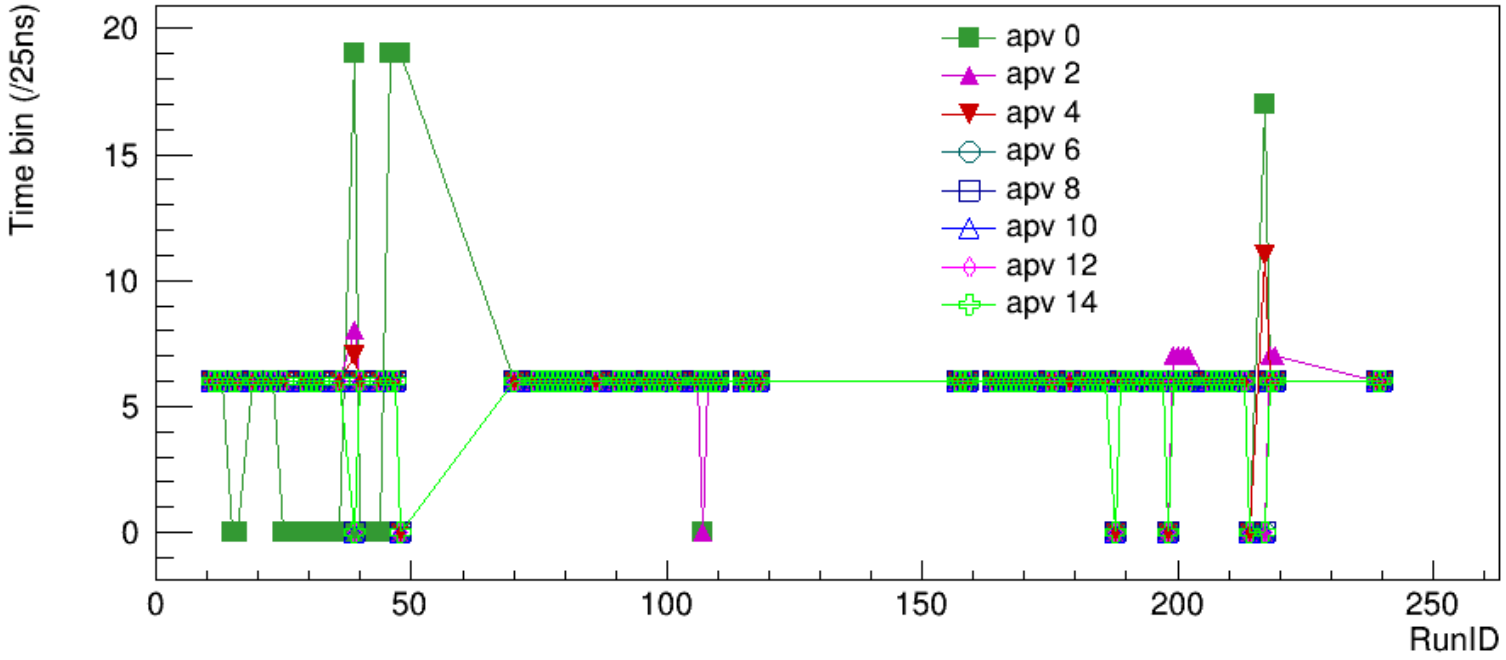


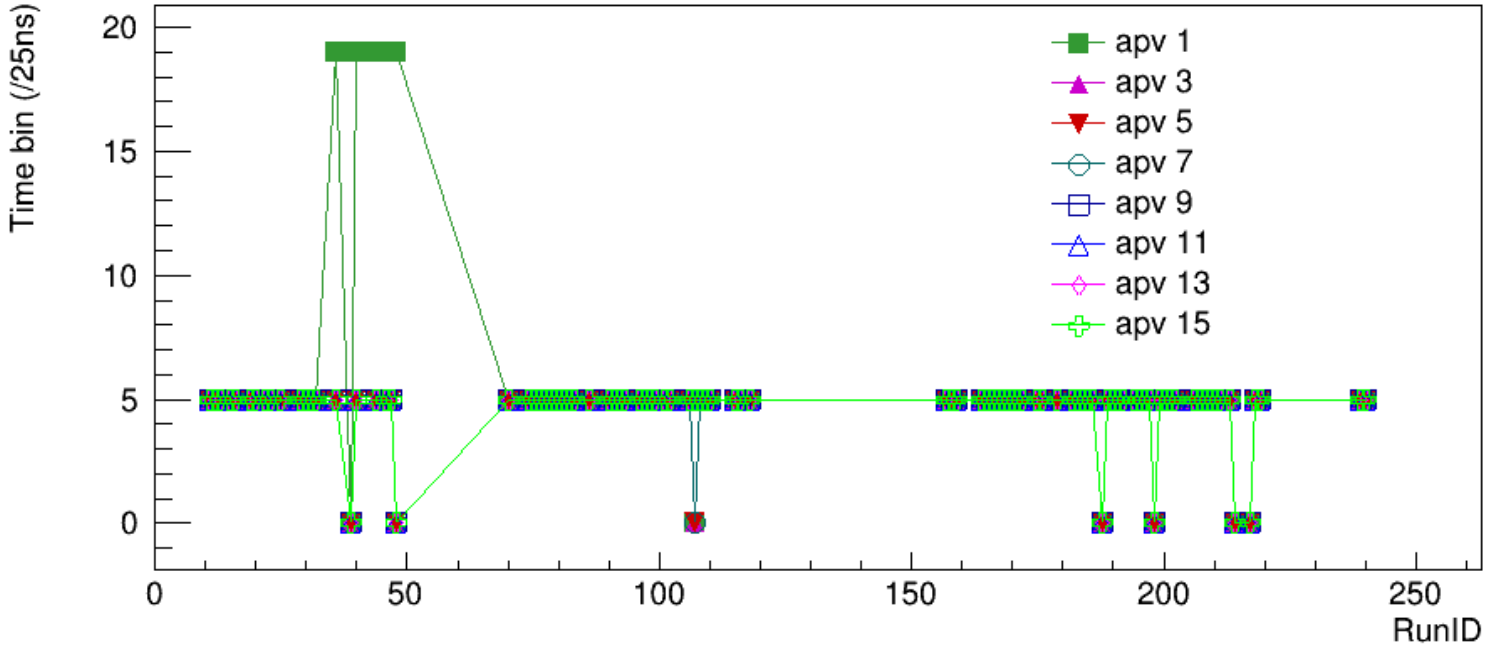


Position of the signal maximum

Max(cumulative_signal_array)

- Even APVs, (masters) have signal maximum in time bin 6;
- Odd APVs, (slaves) have signal maximum in time bin 5;





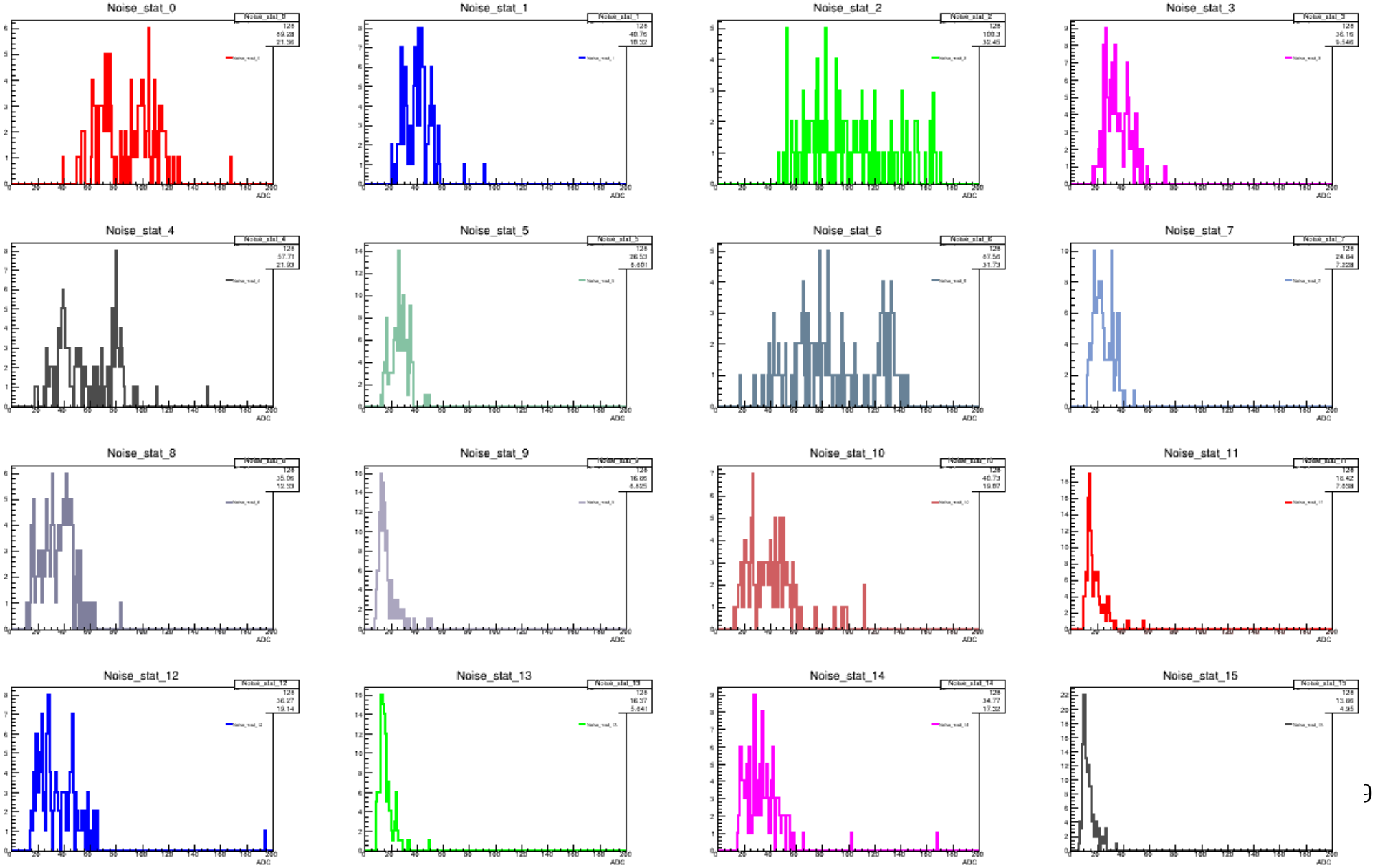
APVs noise run 182

Masters, high gain

Slaves, low gain

Masters, high gain

Slaves, low gain

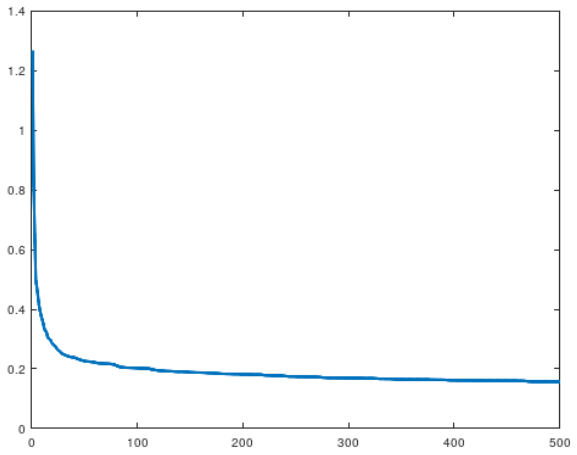


NN Training

- **MPV: 50 ADC**
- **Noise: Mean: 15, sigma: 5**

tarining_sig_sim_land_50_noise_15_t0_3_15k.txt

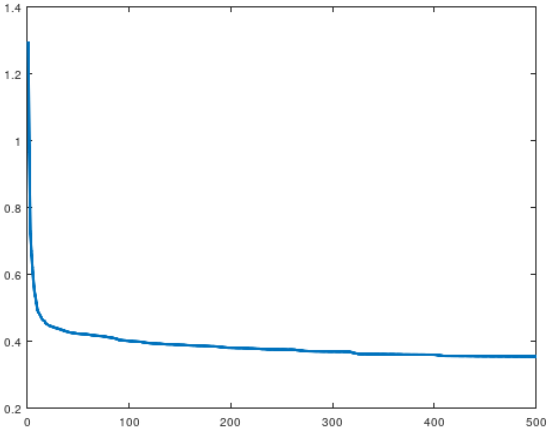
```
Training Neural Network...
Iteration   500 | Cost: 1.564154e-01
Program paused. Press enter to continue.

Visualizing Neural Network...

Training Set Accuracy: 98.614286

Cross validation Set Accuracy: 98.300000

Test Set Accuracy: 97.775000
```



- **MPV: 50 ADC**
- **Noise: Mean: 30, sigma: 10**

tarining_sig_sim_land_50_noise_30_t0_4_15k.txt

```
Training Neural Network...
Iteration   500 | Cost: 3.551756e-01
Program paused. Press enter to continue.

Visualizing Neural Network...

Training Set Accuracy: 94.700000

Cross validation Set Accuracy: 92.825000

Test Set Accuracy: 92.425000
```



- **MPV: 180 ADC**
- **Noise: Mean: 30, sigma: 10**

tarining_sig_sim_land_180_noise_30_t0_4_15k.txt

```
Training Neural Network...
Iteration   500 | Cost: 9.489138e-02
Program paused. Press enter to continue.

Visualizing Neural Network...

Training Set Accuracy: 99.985714

Cross validation Set Accuracy: 100.000000

Test Set Accuracy: 100.000000
```

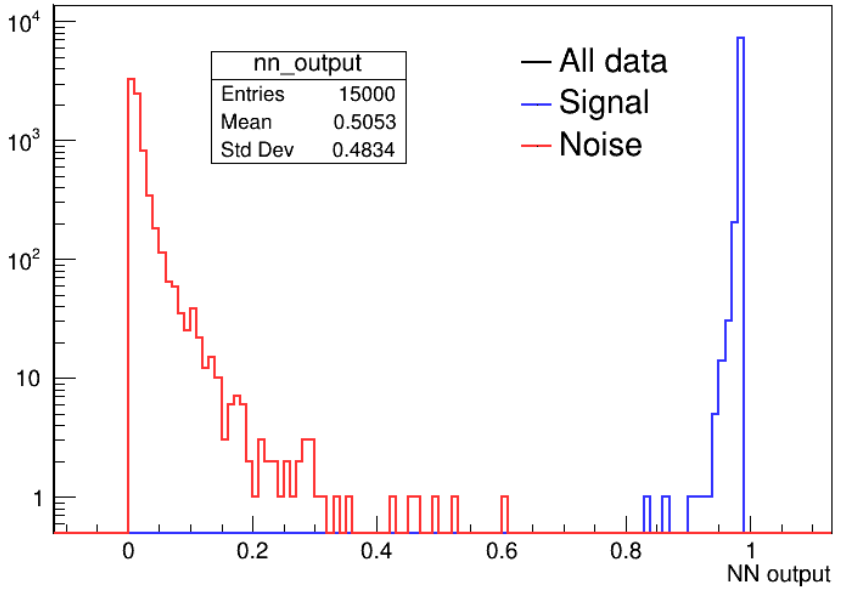
- **MPV: 180 ADC**
- **Noise: Mean: 60, sigma: 20**

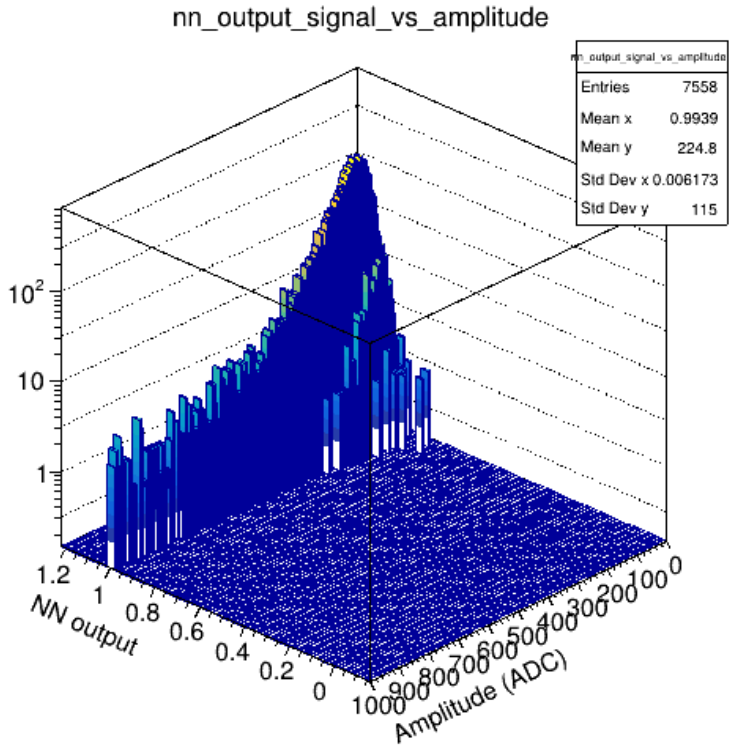
tarining_sig_sim_land_180_noise_60_t0_4_15k.txt

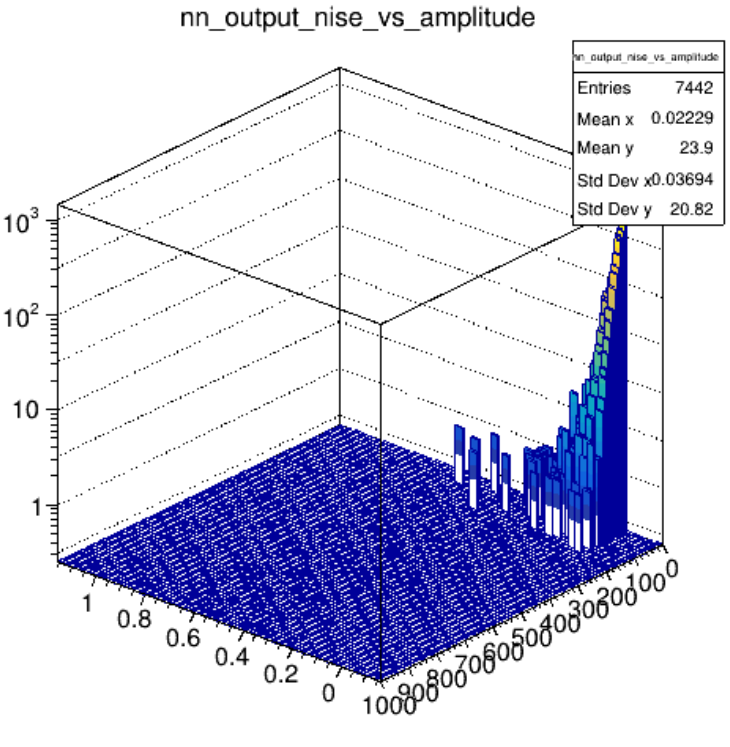
```
Training Neural Network...
Iteration   500 | Cost: 1.998111e-01
Program paused. Press enter to continue.

Visualizing Neural Network...

Training Set Accuracy: 99.428571

Cross validation Set Accuracy: 97.825000

Test Set Accuracy: 98.200000
```

NN performance

tarining_sig_sim_land_180_noise_30_t0_4_15k.txt

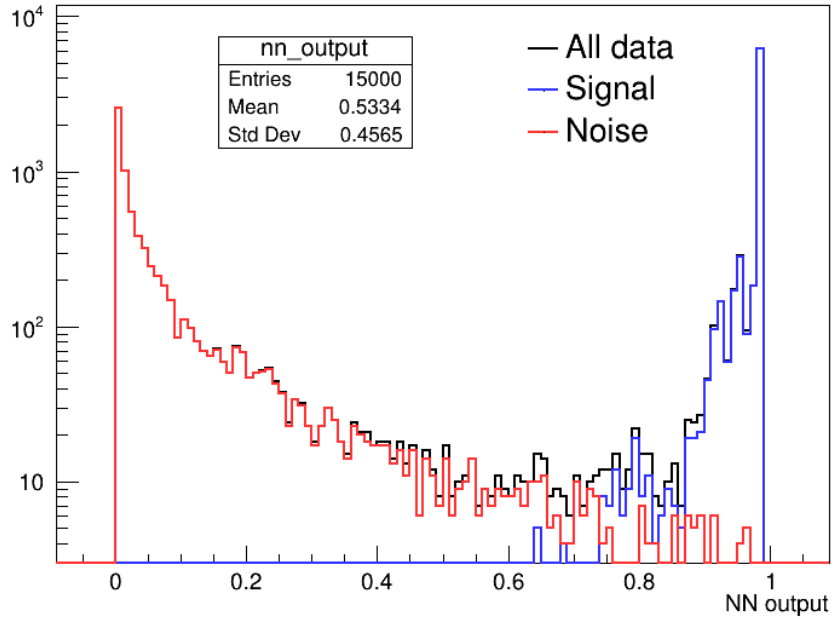






NN performance

tarining_sig_sim_land_180_noise_60_t0_4_15k.txt

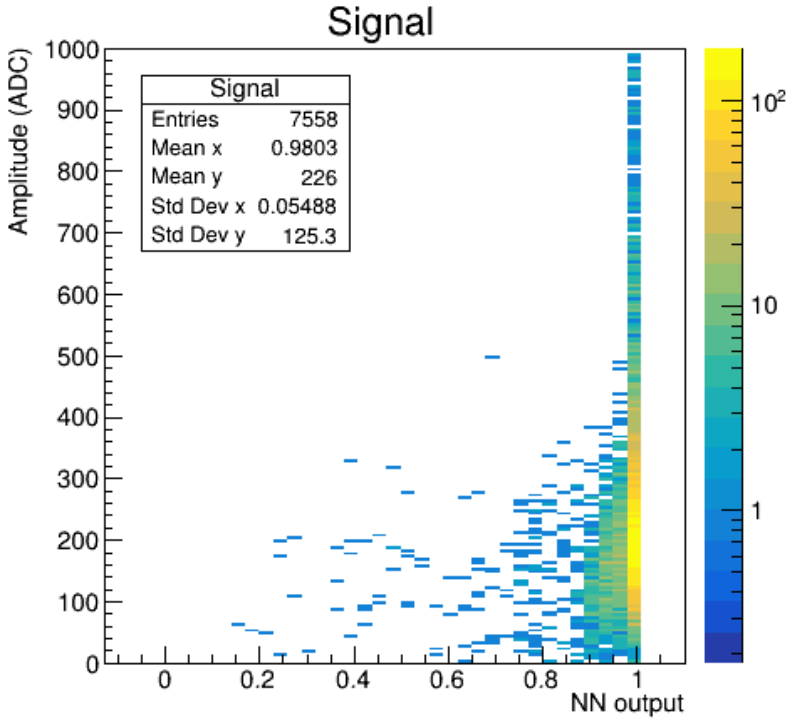


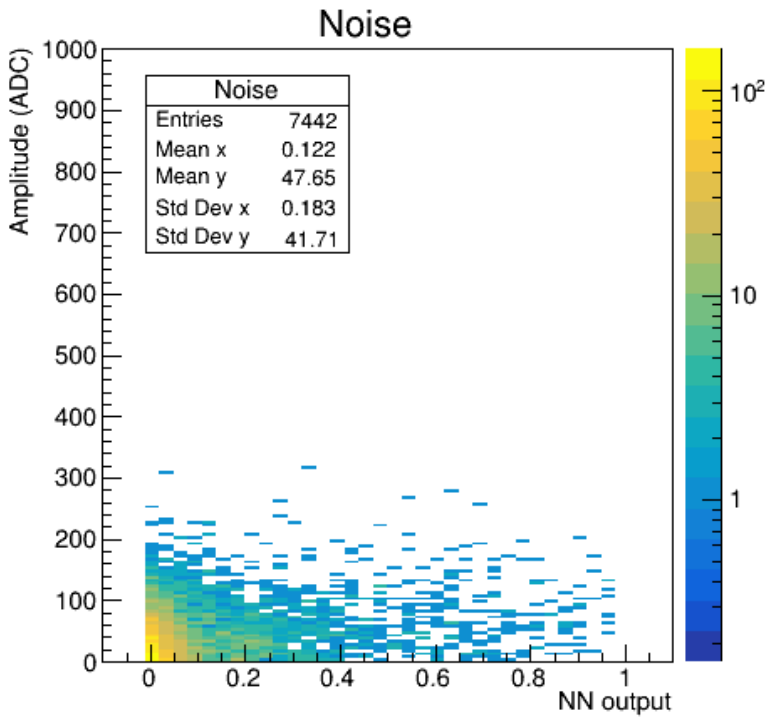
For the threshold at 0.7

Correct predictions:
n/n_data: 14816 / 15000 = 98.7733%

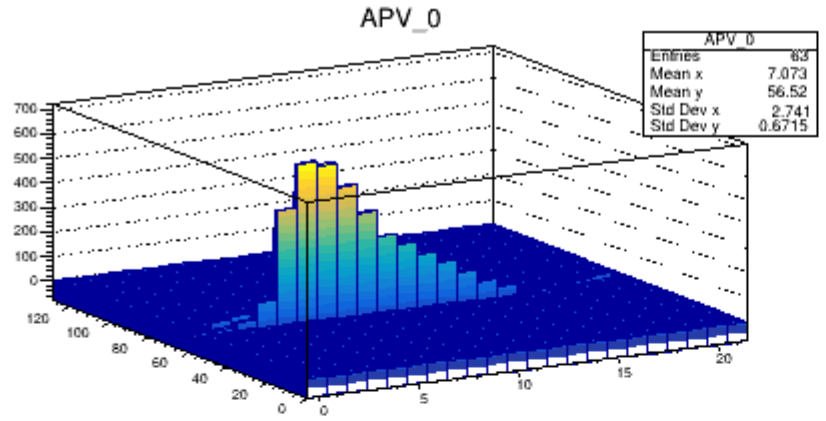
Correct signal:
n/n_data: 7492 / 7558 = 99.1268%

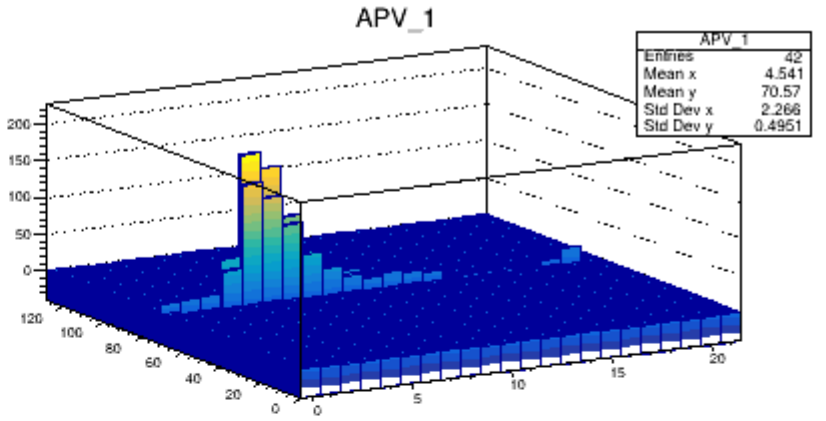
Correct noise:
n/n_data: 7324 / 7442 = 98.4144%

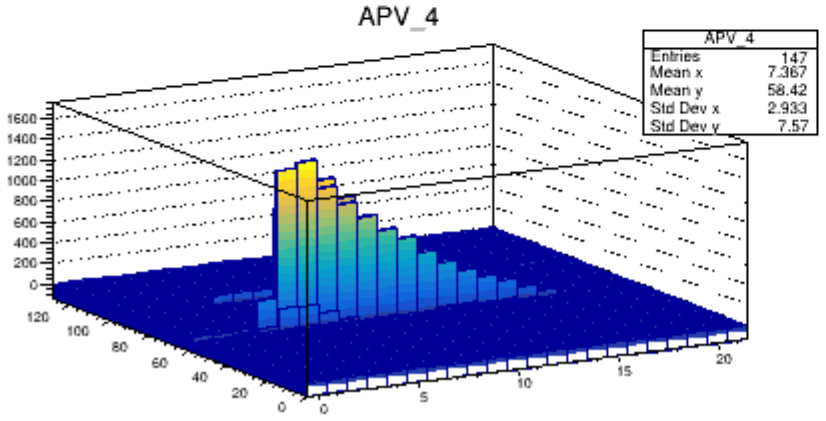


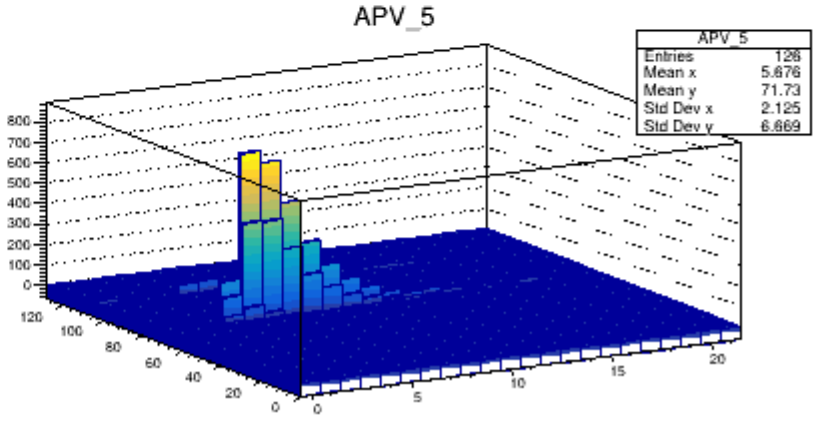


Run 77 event 10

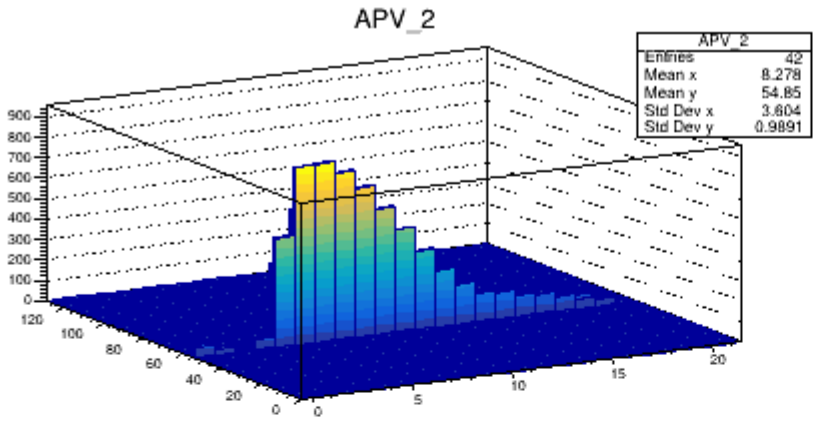


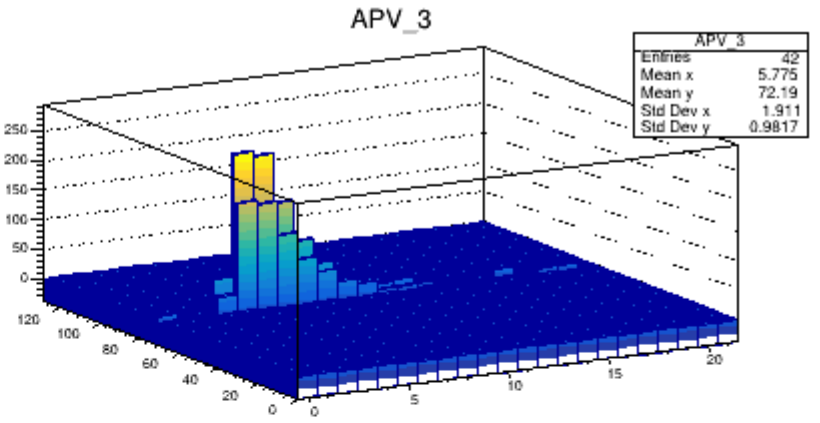


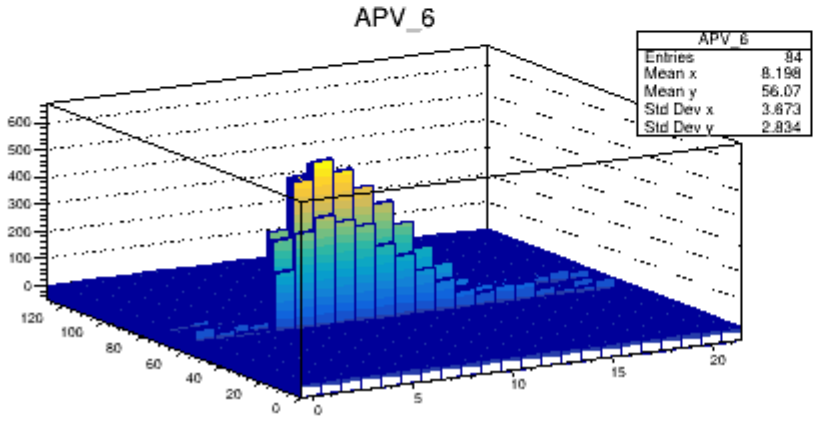


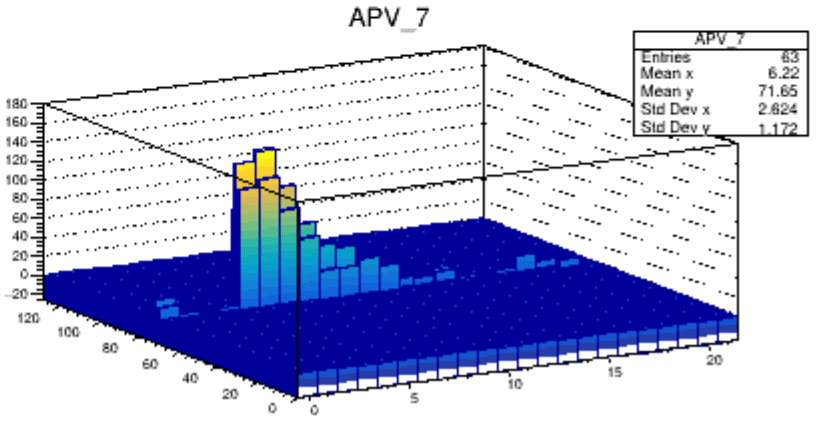


Run 77 event 10

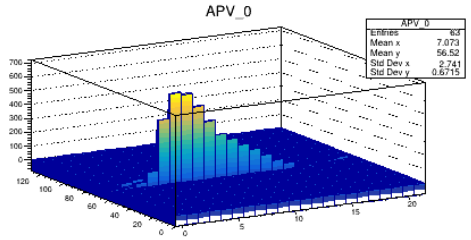


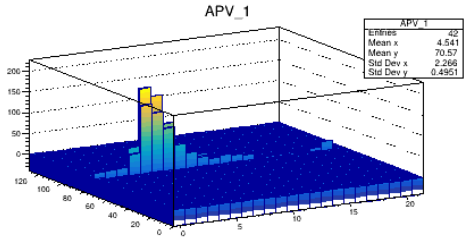


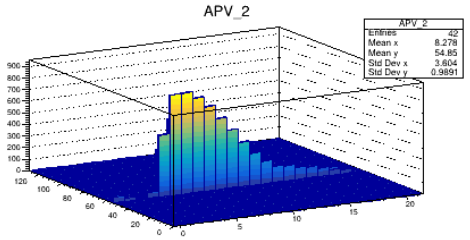


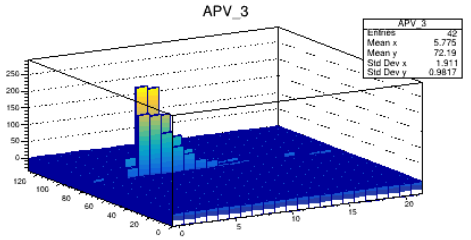


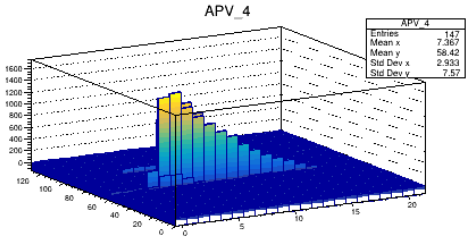
Run 77 event 10

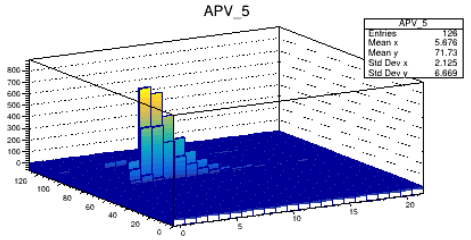


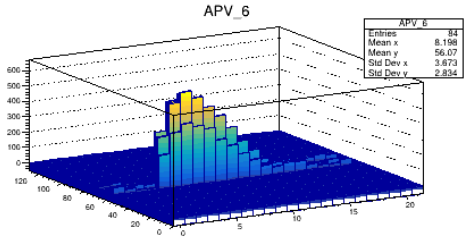


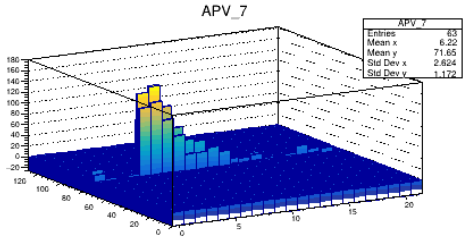


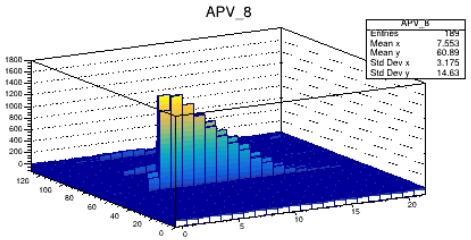


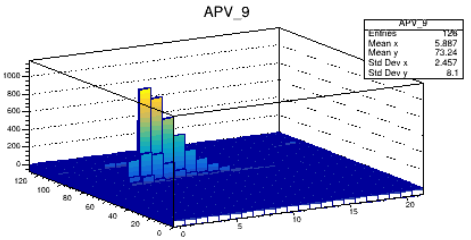


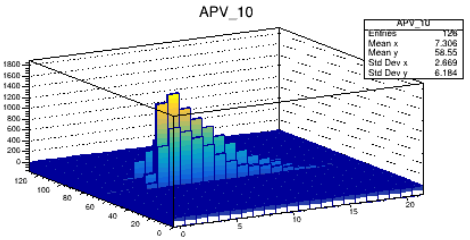


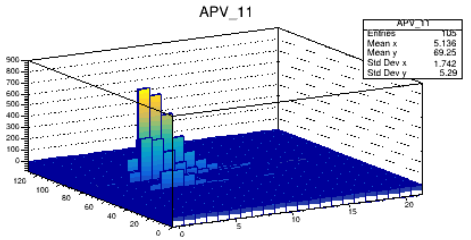


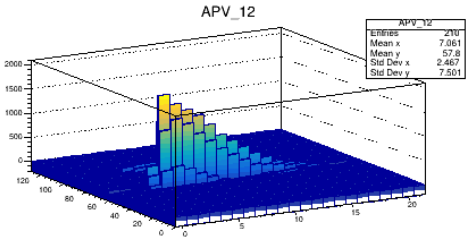


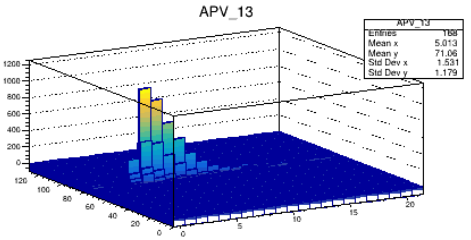


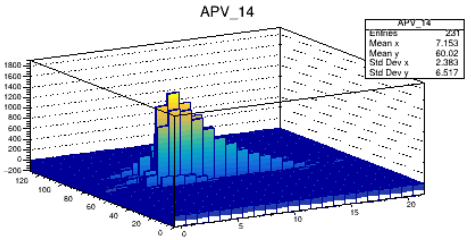


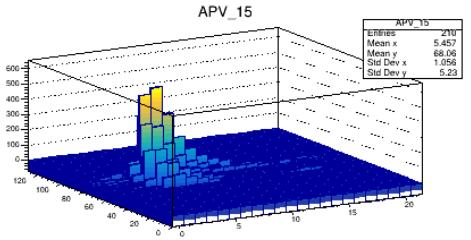












Preprocessed tree

```
p_recotree->Branch("apv_evt",     &m_apv_evt);
p_recotree->Branch("apv_time_s",  &m_apv_time_s);
p_recotree->Branch("apv_time_us", &m_apv_time_us);
p_recotree->Branch("apv_fecNo", &p_m_apv_fec);
p_recotree->Branch("apv_id",    &p_m_apv_id);
p_recotree->Branch("apv_ch",    &p_m_apv_ch);
```

```
…….
std::vector <UInt_t>      m_apv_ch;
std::vector <Double_t>    m_apv_signal_maxbin;
…….
```

```
p_recotree->Branch("apv_signal_maxbin",   &p_m_apv_signal_maxbin);    // maximum value of the bins in signal range
p_recotree->Branch("apv_signal_bint1",    &p_m_apv_signal_bint1);     // value of time bin 10
p_recotree->Branch("apv_signal_bint2",    &p_m_apv_signal_bint2);     // value of time bin 11

p_recotree->Branch("apv_max_bin",  &p_m_apv_maxbin);       // number of the time bin with maximum signal
p_recotree->Branch("apv_bint1",    &p_m_apv_bint1);        // number of the fixed time bin for this apv and this run


p_recotree->Branch("apv_signal_maxfit",   &p_m_apv_signal_maxfit);    // maximum of the fit with RC-CR response function
p_recotree->Branch("apv_fit_t0",          &p_m_apv_fit_t0);           // t0 of the fit with RC-CR response function
p_recotree->Branch("apv_fit_tau",         &p_m_apv_fit_tau);          // tau of the fit with RC-CR response function
p_recotree->Branch("apv_fit_chi2",        &p_m_apv_fit_chi2);         // chi2 of the fit with RC-CR response function
p_recotree->Branch("apv_nn_output",       &p_m_apv_nn_output);        // neural network output

p_recotree->Branch("apv_cm",        &p_m_apv_cm);   // commomn mode noise
```

---

https://www.desy.de/~oborysov/tb2020/index.html

Process_apv_data.h
Process_apv_data.C

} SRS "raw" tree

APV_Reco_Tree.h
APV_Reco_Tree.C
ReadRecoTest.C

} Preprocessed tree

---

info.txt

Short instructions

SRS “raw” tree

```
p_rawtree->SetBranchAddress("apv_evt",   p_m_apv_evt);
p_rawtree->SetBranchAddress("time_s",    p_m_time_s);
p_rawtree->SetBranchAddress("time_us",   p_m_time_us);
p_rawtree->SetBranchAddress("apv_fecNo", &p_m_apv_fec);
p_rawtree->SetBranchAddress("apv_id",    &p_m_apv_id);
p_rawtree->SetBranchAddress("apv_ch",    &p_m_apv_ch);
p_rawtree->SetBranchAddress("mm_id",     &p_m_mm_id);
p_rawtree->SetBranchAddress("mm_readout", &p_m_mm_readout);
p_rawtree->SetBranchAddress("mm_strip",   &p_m_mm_strip);

p_rawtree->SetBranchAddress("apv_q",      &p_m_apv_q);
p_rawtree->SetBranchAddress("apv_presamples",     &p_m_apv_presamples);
```

Summary

- SRS TB2020 and TB2019 data were preprocessed using the procedure similar to TB2016.
- Several signal estimations are available:
  - Value from the expected maximum time bin (6 for master and 5 for slave);
  - Maximum within the range around expected value (time bins 1-15);
  - Amplitude of the fit with CR-RC response function;
- Output of neural network trained separately for different gain and noise level;
- Parameter of CR-RC response function: t_0, τ and Chi^2 .
- There is significant difference in signal shape between modified and non-modified APV. Can be taken into account for NN training when confirmed and understood.
- Deconvolution can be also used for signal estimation.