EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics

Internal Note/PHOS
ALICE reference number

ALICE-INT-2006-014 version 1.0

Date of last change
29.08.2006

Investigation of Time and Amplitude Resolution of the Photon Spectrometer PHOS FEE with Simulation

Authors:

Yu. A. Kucheryaev [a], A. V. Kuryakin [b], Yu. G. Sibiriak [a]

[a] *Russian Research Centre Kurchatov Institute, Moscow, Russia*
[b] *RFNC VNIIEF, Sarov, Russia*

Abstract:

PHOS is the photon spectrometer for the ALICE experiment at the Large Hadron Collider (LHC) at CERN. The PHOS consists of 17920 detection channels, segmented in 5 modules. For identification of heavy particles PHOS will use a time of flight (TOF) method. It is proposed to use the same electronics for measurement of particle's energy and TOF. This paper is described the measurement electronics simulation aimed at estimate of the amplitude and time resolutions and their dependence on the electronics parameters.

Investigation of Time and Amplitude Resolution of the Photon Spectrometer PHOS FEE with Simulation

Yu. A. Kucheryaev [a], A. V. Kuryakin [b], Yu. G. Sibiriak [a]

[a] *Russian Research Centre Kurchatov Institute, Moscow, Russia*
[b] *RFNC VNIIEF, Sarov, Russia*

Abstract

PHOS is the photon spectrometer for the ALICE experiment at the Large Hadron Collider (LHC) at CERN. The PHOS consists of 17920 detection channels, segmented in 5 modules. For identification of heavy particles PHOS will use a time of flight (TOF) method. It is proposed to use the same electronics for measurement of particle's energy and TOF. This paper is described the measurement electronics simulation aimed at estimate of the amplitude and time resolutions and their dependence on the electronics parameters.

---

1 Introduction

The main task of PHOS [1] is an identification and measurement of the energy of direct photons emitted by quark-gluon plasma cloud. Expected signal of the direct photons is a small difference between two distributions, namely, measured spectrum of the particles identified as photons and calculated spectrum of photons generated in decays of other particles. To achieve statistical reliability of results the both spectra have to be measured as precisely as possible. The process $\pi^0 \to 2\gamma$ gives main contribution to the decay spectrum, but η-meson decay and antineutron annihilation are also significant. For identification of π^0 and η-mesons the two photons invariant mass method can be used. For antineutrons identification the time of flight (TOF) method may appear more reliable. The distance between the interaction point and the spectrometer front surface is 4.6 m, so one needs to have the time resolution around 1 ns for 2 GeV particles.

PHOS measuring channel starts from $PbWO_4$ crystal which converts the incident particle energy into photons of visible light. The light is converted into a voltage pulse by avalanche photodiode (APD) connected to a charge sensitive preamplifier (CSP) input. CSP output signal comes to a shaper that amplifies it and increases signal/noise ratio due to its specific frequency response. The shaper output signal is converted into a digital code by fast 10-bit ADC that performs continuous sampling of its input voltage with period T_s (typically 100 ns). The codes are stored in a buffer memory [2]. Useful information is marked by a trigger signal.

For TOF measurements a time (fast) channel can be used, which includes a shaper and a time-to-amplitude converter. But in measurement systems consisting of several thousands of channels, like PHOS, such approach results in a considerable increase of hardware costs. Another approach consists in a digital processing of the stored sample codes. This way allows to determine the signal time parameters and, in particular, to measure the particle TOF.

In electron beam tests performed with a measuring channel prototype, its relative energy resolution has been found [1] in the form of

$$\frac{\sigma_E}{E} = \sqrt{\frac{a^2}{E^2} + \frac{b^2}{E} + c^2} \ .$$

Here a = 0.013 GeV, b = 0.0358$\sqrt{GeV}$, c = 0.0112, energy E is in GeV. The 1st term under root sign represents electronics noise contribution, the 2nd one – statistics of photoelectrons, the 3rd one is system constant. At E = 1 GeV the sum of the last two terms is equal to 0.0014. So the measuring electronics will not worsen the PHOS energy resolution if its relative amplitude resolution σ_A / A at the same energy is better than 1%.

The present paper summarizes the first part of our work devoted to optimal choice of PHOS front-end electronics (FEE) parameters and to elaboration of suitable methods of data processing. At this initial stage our main objective was to resolve a principal question: is it possible to achieve the time and amplitude resolutions which can satisfy PHOS physics requirements at least at somewhat idealized conditions? If an answer is negative, a prospects of above-mentioned work would become strongly problematic because at real conditions the resolutions will be only worse. To get an estimate of PHOS FEE time and amplitude resolution we have used FEE signal simulation within the bounds of simplified model.

Several versions of the model have been worked out. They use different functions for creation of the pulse histogram, different ways of the pulse parameters determination, and different sets of these parameters. The versions using program package MINUIT [3] for search of model pulse parameters are described in Section 2. The main results of the simulation are presented in Section 3. Section 4 contains a brief discussion of the results.

2 Model

The computer simulation of the ADC output codes consists in following. It is specified a function which in some approximation describes the analog signal on the shaper output (model function). Then a Gaussian noise with zero mean and given sigma is added to this function to simulate the thermal noise of the shaper output signal. ADC performs sampling of the analog signal in time and digitizes these samples. The digitization procedure creates additional stochastic errors, so called digital noise. The same rounding off is performed when the model pulse histogram is created. The abscissa represents time in the sampling periods (clock bins) and the ordinate – the signal in ADC bits. For determination of the model pulse parameters the histogram must be processed with some method. In this paper the fit method using MINUIT package is described. Multiple repetition of the foregoing procedure creates a set of the model histograms which differ only by noise pattern. The fit results are accumulated in the parameters' histograms. An analysis of these distributions allows to estimate the time and amplitude resolutions at given conditions.

2.1 Model function

The shaper is a filter with active elements. The schematic diagram of the n- order filter is shown in Fig. 1 where n is a number of the integration stages.

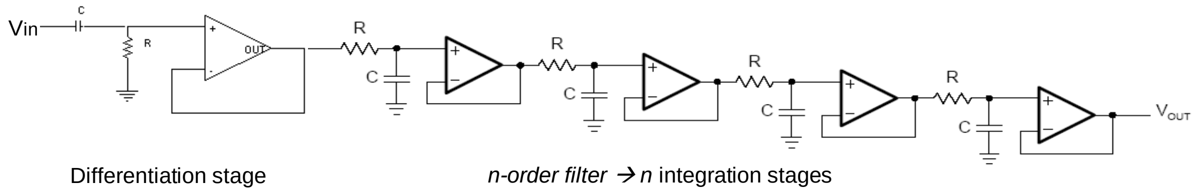

Fig. 1. The schematic diagram of the n-order filter.

If RC-values of the differentiating and integrating circuits are equal and a step signal comes to the filter input, then the filter output signal is

$$\gamma(z) = A_0 \left(\frac{z}{N} \right)^N \exp(N - z), \tag{1}$$

where $z = t/\tau$, t is time counted from the beginning of the pulse, $\tau = RC$, N is the filter order, A_0 is output signal amplitude. Maximum of $\gamma(z)$ is achieved at $z_{\max} = N$; therefore, the signal rise time (sometimes called "peaking time") is equal to $N\tau$.

The pulse model takes into account the following PHOS FEE features:

1) in absence of a signal on the shaper input, some approximately constant voltage level (pedestal P) exists on its output;
2) the beginning of the pulse is delayed relative to the beginning of the histogram by some value $Delay0$ to ensure pedestal measurements.

So the model function can be taken in the form of

$$f(z) = \begin{cases} P & \text{at } z < 0, \\ P + \gamma(z) & \text{at } z \geq 0, \end{cases} \tag{2}$$

where

$$z = (x - Delay0)/\tau , \tag{3}$$

x is time counted from the beginning of the histogram. One more FEE feature is that the pulse appearance moment is known with accuracy up to 1 sampling clock bin because only integer number $delayBins$ of clock bins preceding the pulse beginning is defined in the course of the pulse sampling, while the pulse appears at arbitrary moment of the next bin. So an initial delay of the pulse in the model is described as

$$Delay0 = delayBins + del, \tag{4}$$

where del is a random value uniformly distributed from 0 to 1 clock bin. An example of $Delay0$ distribution is shown in Fig. 2. A typical ADC output pulse histogram created with model function (1) – (3) and Gauss noise with zero mean and $\sigma_{Gauss} = 1$ ADC bin is presented in Fig. 3.

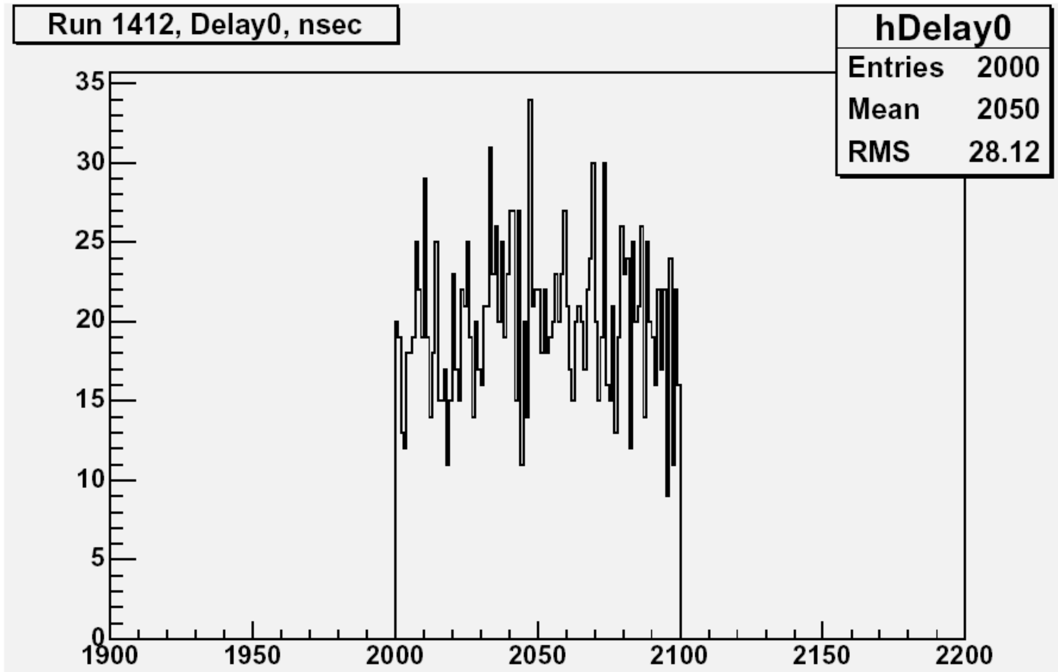

Fig. 2. Typical distribution of initial delay of the pulse beginning relative to the histogram beginning in the computer run of 2000 model pulses. Sampling frequency *Freq* = 10 MHz (1 clock bin = 100 ns), *delayBins* = 20.

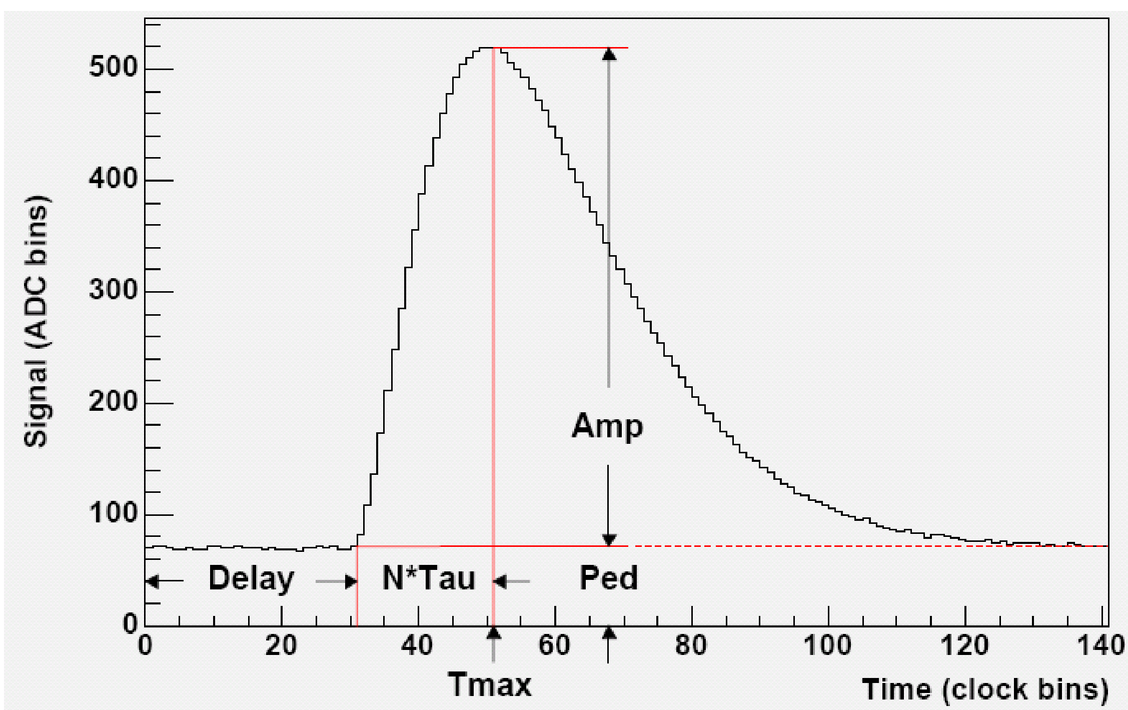

Fig. 3. The model pulse. Its parameters are shown in computer notations: N is filter order, Tau = τ = RC, Ped = P is pedestal, Amp = A_0 is amplitude.

2.2 Pulse histogram fit

The function defined by equations (1) – (3) where all or several pulse parameters are considered as free (i.e. must be found as a result of the fit) is fitted to the model histogram. In present paper three versions of fitting program are used. In the 1st version 4 parameters are sought: Ped, Amp, Tau, and $Delay$. In the 2nd one Tau is given and fixed during the fit, in the 3rd one Tau and Ped are fixed.

Least squares fit uses MINUIT package [3] implemented in the form of ROOT TH1::Fit method. The values entered into model function for pulse histogram creation are used then as initial values of the sought parameters.

In each computer run 2000 model histograms are created. Each histogram is fitted, and the results are accumulated in the parameters' histograms which, in turn, are fitted by Gaussian. The shaper relative amplitude resolution is defined as

$$\delta A = sigmaAmp / Amp, \tag{5}$$

where $sigmaAmp$ is a standard deviation of the Amp distribution.

The time resolution of the shaper is defined by two different ways. One can accept as a "time stamp" the beginning of the pulse and calculate a difference

$$Tdiff = Delay - Delay0 \tag{6}$$

between fitted value $Delay$ and given value $Delay0$ (4). Then the shaper time resolution may be characterized by $sigmaTdiff$ value. In [4] it has been proposed to use another "time stamp", namely a moment x_* when the shaper output signal 1st derivative reaches maximum. It follows from (2) and (3) that $x_* = Delay + (N - \sqrt{N})\tau$. A width $sigmaRiseTdiff$ of the histogram

$$RiseTdiff = x_* - Delay0 = Tdiff + (N - \sqrt{N}) * Tau \tag{7}$$

represents another definition of the shaper time resolution. It is clear that $sigmaRiseTdiff$ coincides with $sigmaTdiff$ at N = 1 or in the case of fixed Tau.

A value

$$Disp = chi2/NDF \tag{8}$$

gives the fit dispersion estimate and is used for the model performance check-up. Here $chi2$ is sum of squares of differences between the model histogram points and fit function points, NDF is number of degrees of freedom, i.e. number of fitted points minus number of sought parameters. The dependence of mean value $< Disp >$ on Gauss noise dispersion is shown for example in Fig. 4. If the model contains no errors except for noises, this result allows to estimate digital noise dispersion from equation

$$< Disp > = \sigma^2_{Gauss} + \sigma^2_{digital}. \tag{9}$$

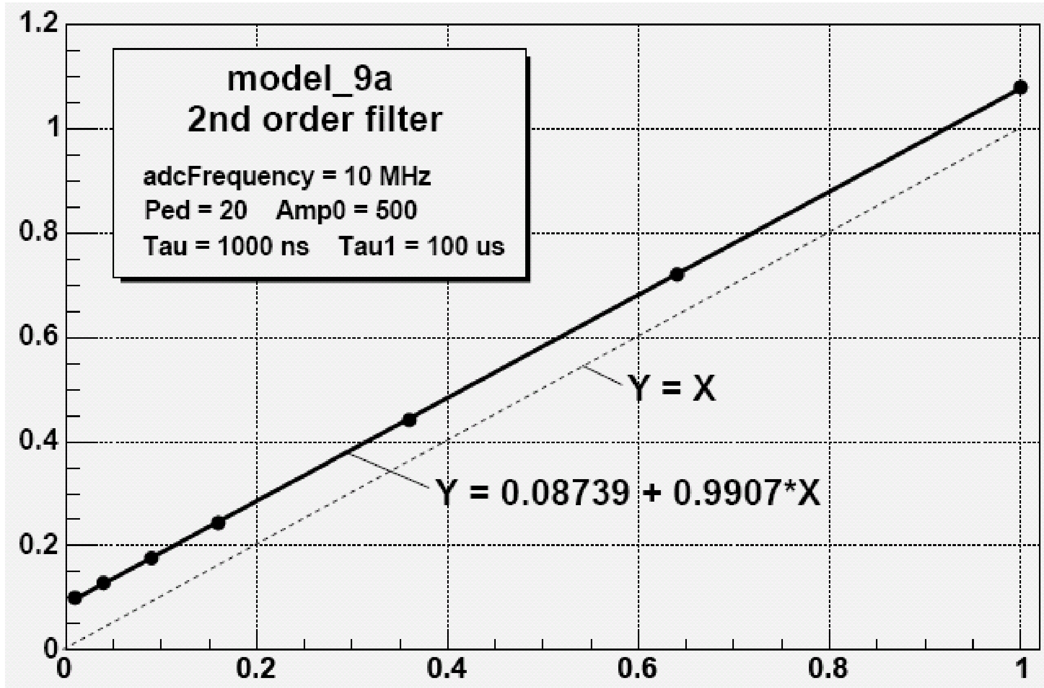

Fig. 4. The dependence of mean fit dispersion $< Disp > = chi2/NDF$ on Gauss noise dispersion (solid line). The dashed line shows the same dependence in absence of digital noise.

It follows from Fig. 4 that $\sigma_{digital}$ = 0.29 ADC bins (compare with expected mean value of rounding-off error which is equal to 0.5 ADC bins).

In practice another fit criterion is more convenient, namely

$$Ratio = Disp / \sigma^2_{Gauss} = 1 + \sigma^2_{digital} / \sigma^2_{Gauss} . \qquad (10)$$

If the program works properly, then at $\sigma_{Gauss} \sim 1$ ADC bins $< Ratio >$ must be close to 1.

Before an investigation of time and amplitude resolution dependencies on the pulse parameters one has to establish an optimal range of the model histogram fitting: at small number of the fitted points the errors in parameter determination may appear too big, at redundant number of the points the duration of the computer runs is increasing uselessly. The dependence of time and amplitude resolution on M value related to fit range width is presented in Fig. 5 and Fig. 6. Fit range is defined as follows: its left edge always is equal to zero in order to include the pedestal region, its right edge is chosen in the form of

$$xmax = \min(delayBins + M * N * Tau,\ xMax), \qquad (11)$$

where $xMax = 30\ \mu$s is right edge of the model pulse histogram. Because $N * Tau$ is the pulse rise time, the M value in Fig. 5 and Fig. 6 shows how many of such intervals are contained in the fit range.

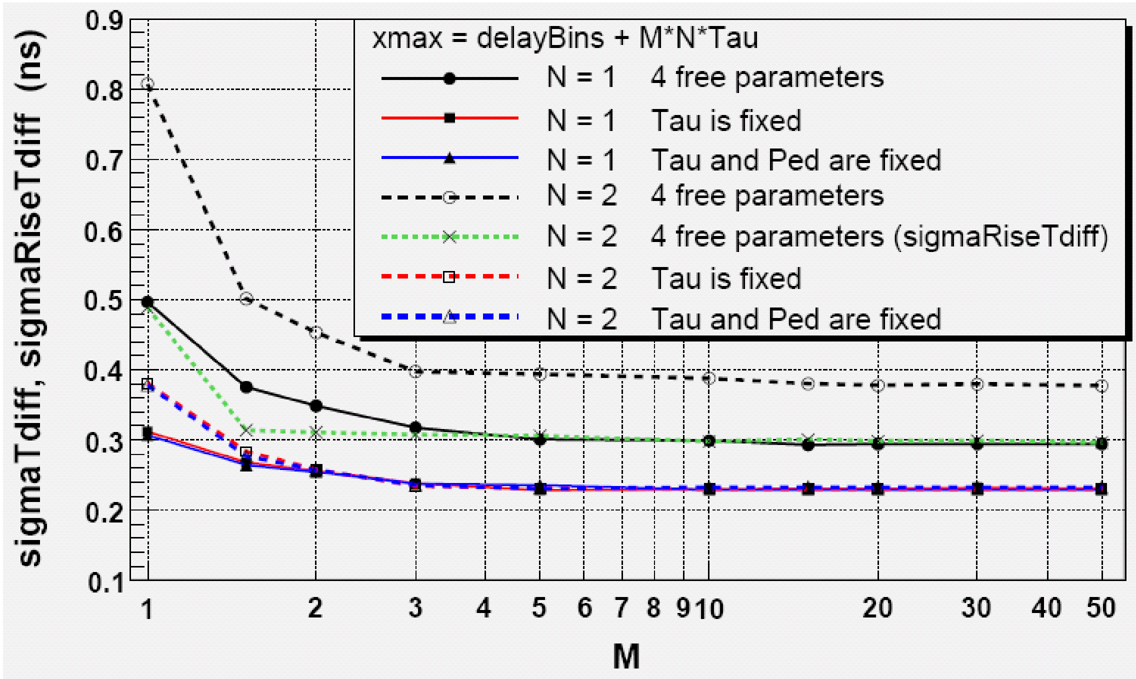

Fig. 5. The dependence of time resolutions *sigmaTdiff* and *sigmaRiseTdiff* on fit range width for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, $N*Tau$ = 500 ns, $Amp0$ = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

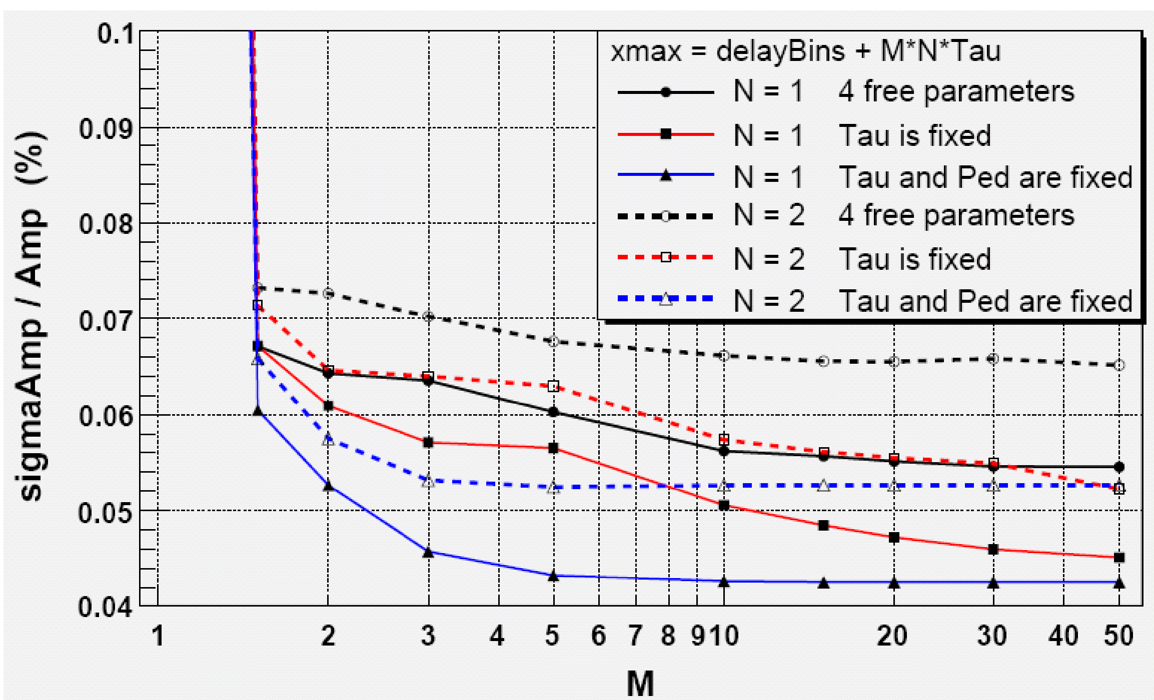

Fig. 6. The dependence of relative amplitude resolution $sigmaAmp / Amp$ on fit range width for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, $N*Tau$ = 500 ns, $Amp0$ = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

It is seen that the time resolution is almost constant at $M \geq 10$ independently of fit method. The amplitude resolution at fixed Tau continues to improve with increase of M up to 50, but this improvement is limited by value $\sim 0.01\%$. In what follows the $xmax$ value from (11) at M = 15 is accepted as an optimal fit range value.

3 Main results

The dependencies of time and relative amplitude resolution on the pulse rise time $N*Tau$, ADC sampling frequency $Freq$, pulse amplitude Amp, and Gauss noise sigma σ_{Gauss} are presented in Fig. 7...14. The shapers with 1st order and 2nd order filters were investigated. Each shaper time and amplitude resolution are obtained by three different fits of the model histogram (see Section 2.2).

4 Discussion

4.1 Time resolution

In the parameters' region that we have studied, $sigmaTdiff$ and $sigmaRiseTdiff$ are approximately proportional to $\sqrt{N*Tau}$ and σ_{Gauss}, and they are approximately inversely proportional to Amp and $\sqrt{Freq}$. This result does not depend on the model pulse histogram fitting method.

If all 4 pulse parameters (Ped, Amp, Tau and $Delay$) are considered as free, and the time resolution is defined by $sigmaRiseTdiff$ value, then the 1st order filter and the 2nd order filter time resolutions are equal at equal rise time values $N*Tau$ and at the same other conditions. This resolution is improved by approximately 30% at fixed Tau. If Tau has been fixed, then additional fixing of the pedestal Ped does not change the time resolution, but slightly decrease the program run time.

It is worthy of notice that the pulse histogram fit at 4 free parameters of the 2nd order filter gives $sigmaRiseTdiff$ value notably less than $sigmaTdiff$. Though as can be seen from (7) $RiseTdiff$ contains another fitted parameter Tau in addition to $Tdiff$. A possible explanation of this result is $Tdiff$ and Tau are partly correlated (see Fig. 15).

4.2 Amplitude resolution

In the investigated area of the parameters, the relative amplitude resolution $sigmaAmp / Amp$ is approximately proportional to σ_{Gauss} and is approximately inversely proportional to $\sqrt{N*Tau}$, Amp, and $\sqrt{Freq}$. As in the case of the time resolution, this result is independent of the model pulse histogram fitting method. In contrast to the time resolution, the amplitude resolution is notably improved when number of fixed parameters increases. At the same fitting method, the amplitude resolution of the 1st order filter is about 20% better than one of the 2nd order filter .

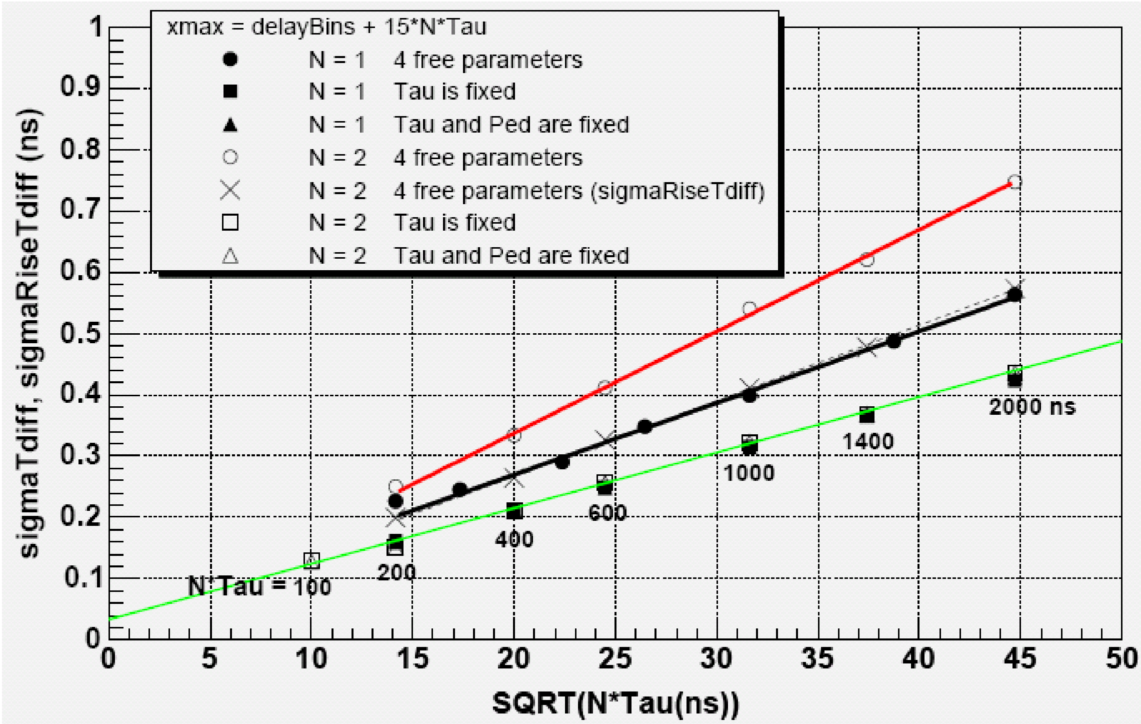

Fig. 7. The dependence of time resolutions *sigmaTdiff* and *sigmaRiseTdiff* on $\sqrt{N*Tau}$ for 1st order (N = 1) and 2nd order (N = 2) filters. *Freq* = 10 MHz, *Amp*0 = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

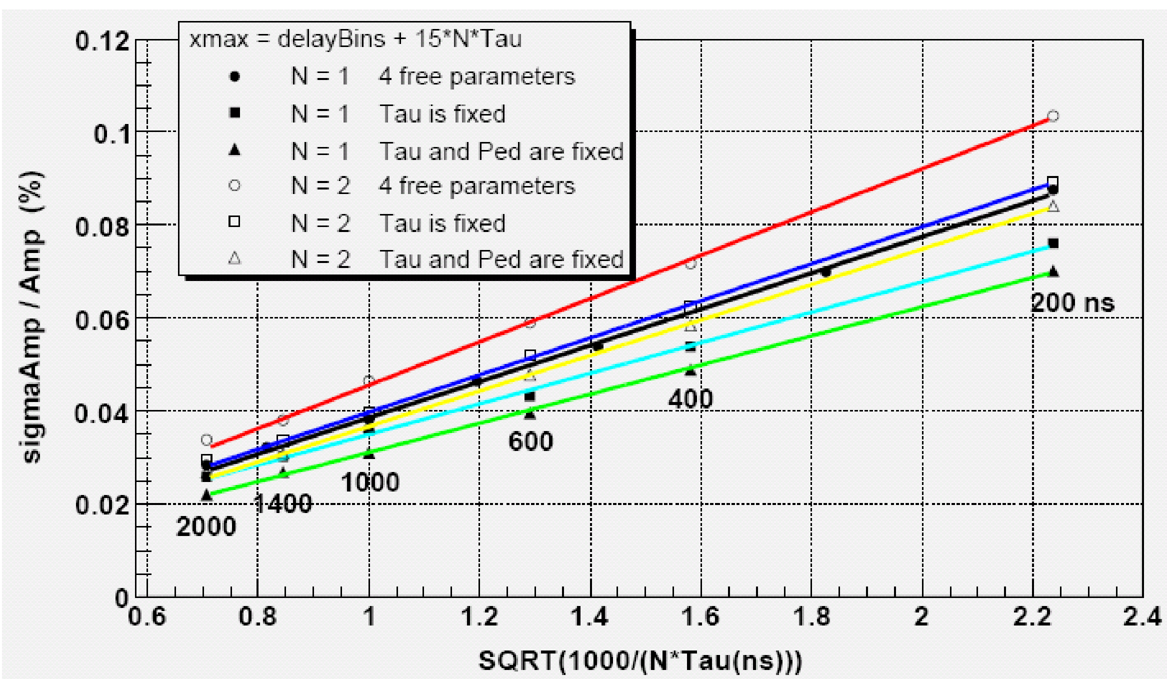

Fig. 8. The dependence of relative amplitude resolution *sigmaAmp / Amp* on $\sqrt{1000/(N*Tau)}$ for 1st order (N = 1) and 2nd order (N = 2) filters. *Freq* = 10 MHz, *Amp*0 = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

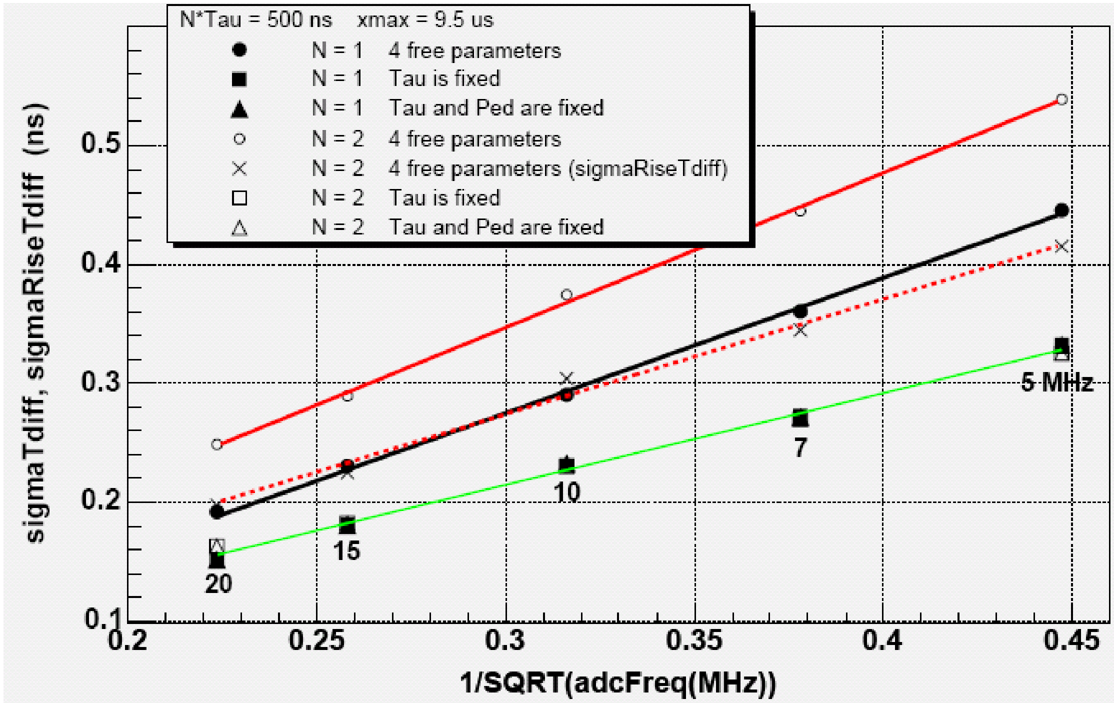

Fig. 9. The dependence of time resolutions *sigmaTdiff* and *sigmaRiseTdiff* on $1/\sqrt{Freq}$ for 1st order (N = 1) and 2nd order (N = 2) filters. $N*Tau$ = 500 ns, $Amp0$ = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

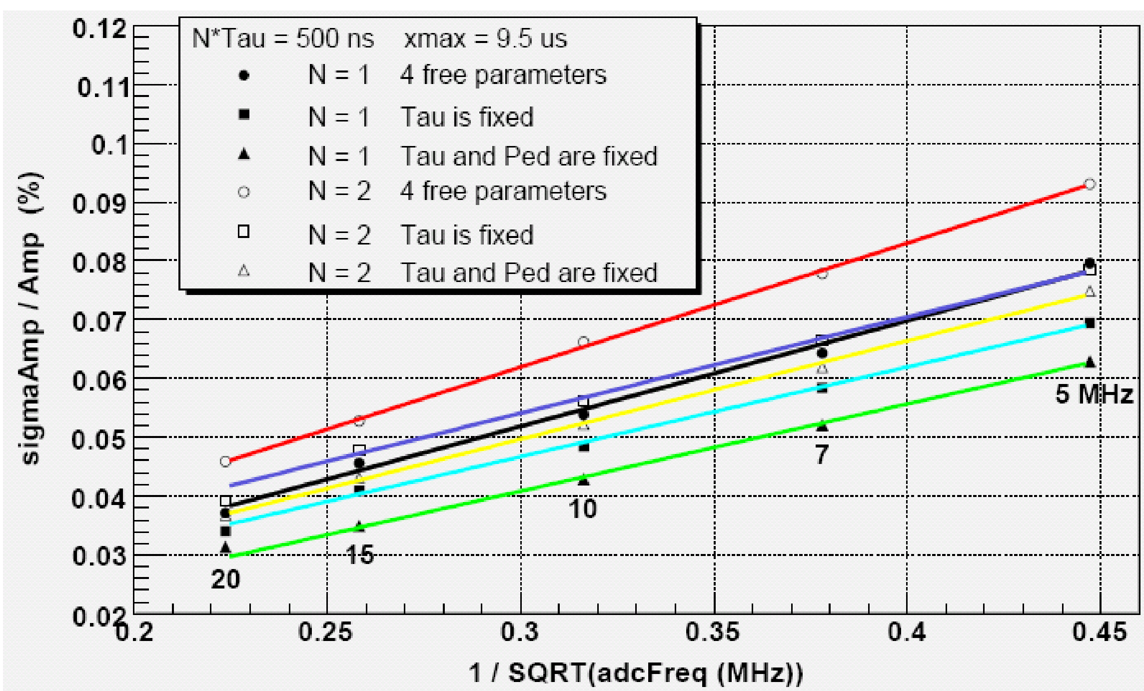

Fig. 10. The dependence of relative amplitude resolution $sigmaAmp / Amp$ on $1/\sqrt{Freq}$ for 1st order (N = 1) and 2nd order (N = 2) filters. $N*Tau$ = 500 ns, $Amp0$ = 500 ADC bins, σ_{Gauss} = 0.6 ADC bins.

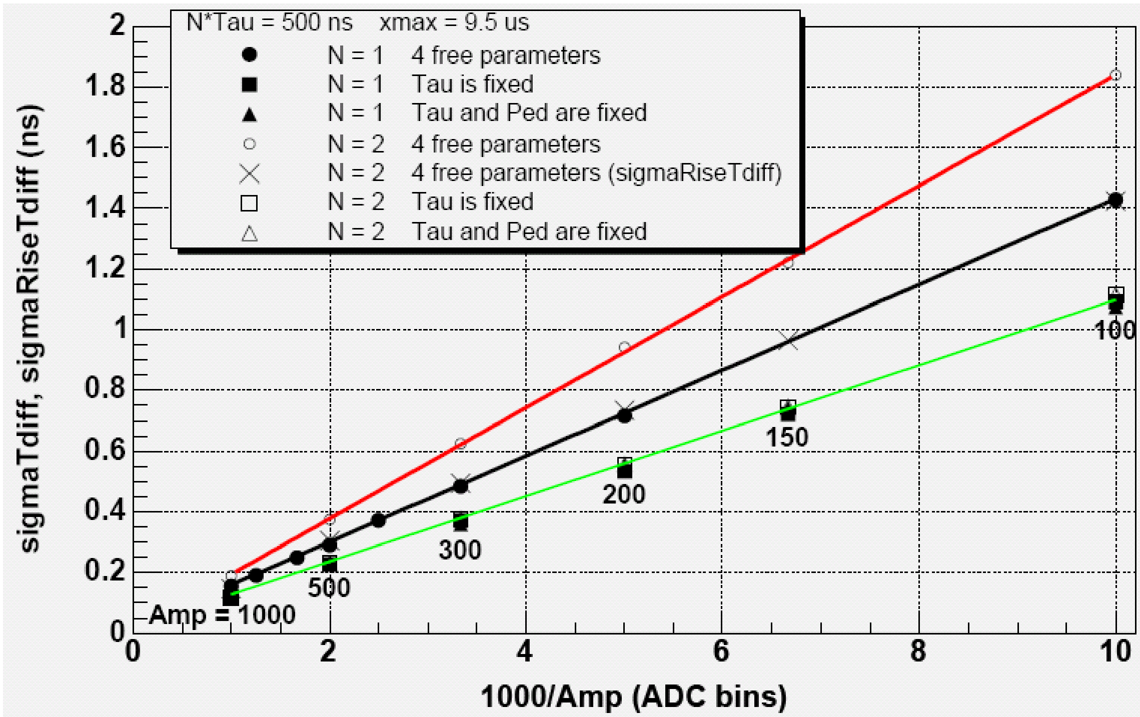

Fig. 11. The dependence of time resolutions $sigmaTdiff$ and $sigmaRiseTdiff$ on $1000 / Amp0$ for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, , $N*Tau$ = 500 ns , σ_{Gauss} = 0.6 ADC bins.

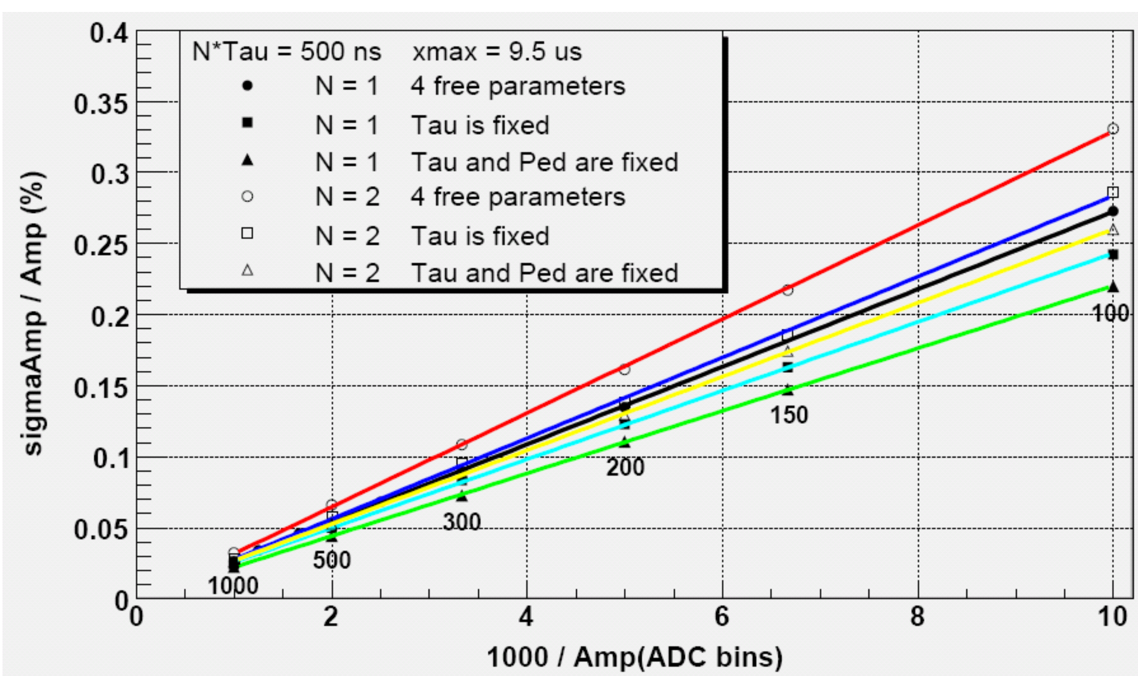

Fig. 12. The dependence of relative amplitude resolution $sigmaAmp / Amp$ on $1000 / Amp0$ for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, $N * Tau$ = 500 ns, σ_{Gauss} = 0.6 ADC bins.

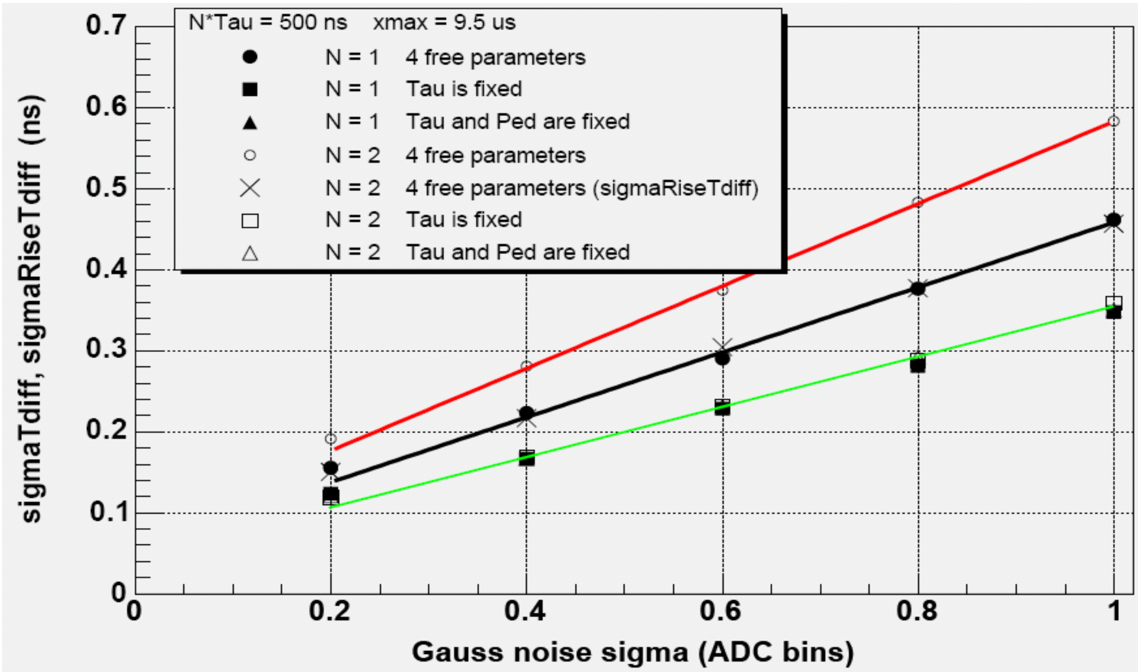

Fig. 13. The dependence of time resolutions $sigmaTdiff$ and $sigmaRiseTdiff$ on Gauss noise sigma σ_{Gauss} for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, $N * Tau$ = 500 ns, $Amp0$ = 500 ADC bins.

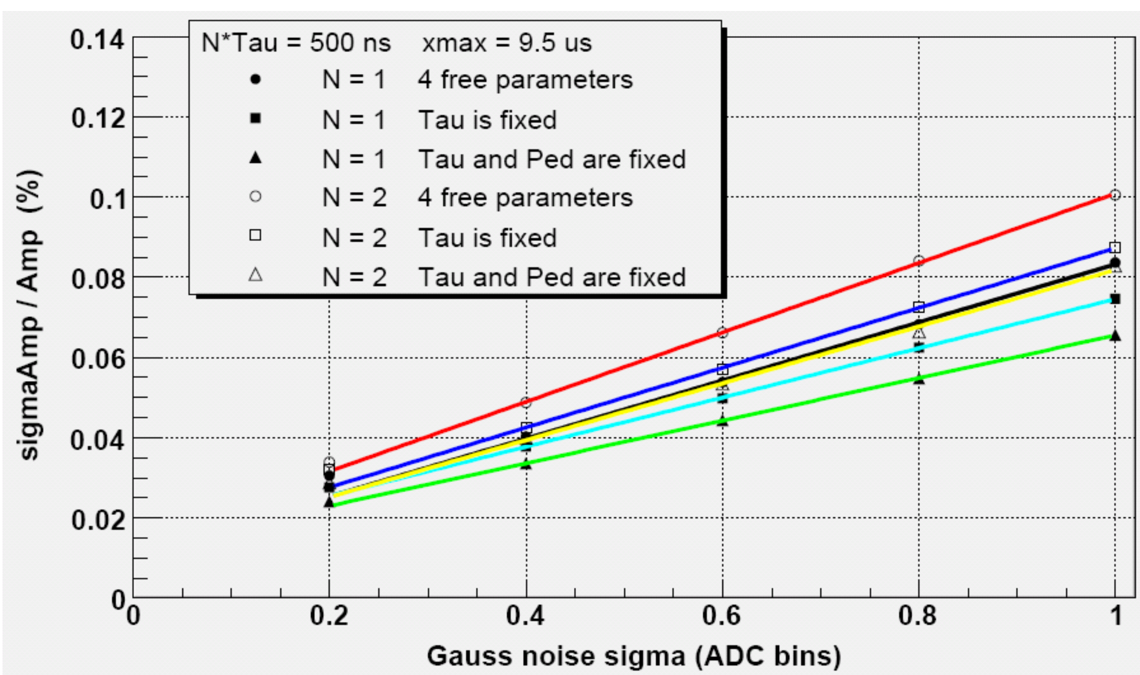

Fig. 14. The dependence of relative amplitude resolution $sigmaAmp / Amp$ on Gauss noise sigma σ_{Gauss} for 1st order (N = 1) and 2nd order (N = 2) filters. $Freq$ = 10 MHz, $N * Tau$ = 500 ns, $Amp0$ = 500 ADC bins.

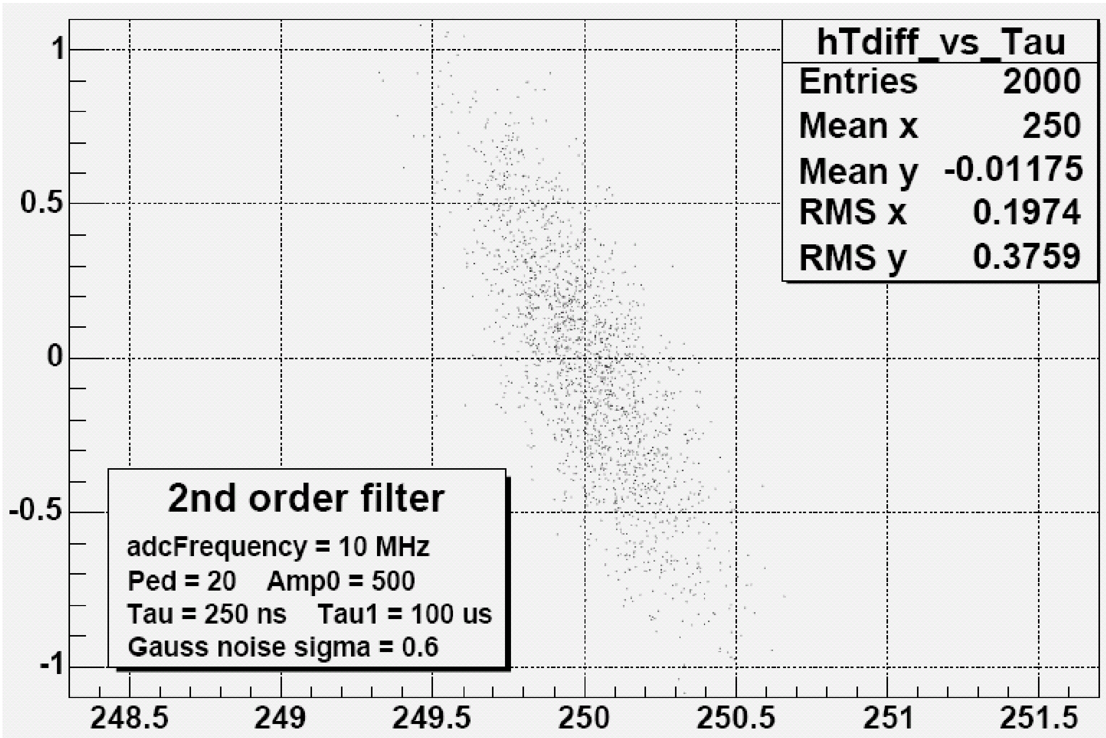

Fig. 15. The $Tdiff$ vs Tau distribution. The both values are in nanoseconds.

4.3 Empiric formulae

All dependencies shown in Fig. 7 … Fig. 14 were fitted by simple functions of the investigated parameters. Based on the functions that have been obtained in the model with fixed Tau and Ped, two empiric formulae are derived. They allow to estimate the shaper time resolution ($sigmaTdiff$) and the amplitude resolution ($sigmaAmp$) at given the sampling frequency $Freq$, the time constant Tau, the pulse amplitude Amp, and Gauss noise sigma σ_{Gauss} without use of relatively slowly working simulation program. An inaccuracy of the empiric formulae in the investigated parameter range does not exceed value of a few percents. The calculations are performed by small program sigmaProbe_1.cxx presented in Attachment.

5 Conclusions

The dependence of PHOS FEE channel time and amplitude resolutions on the pulse parameters and ADC sampling frequency has been studied using an idealized shaper model with the 1st and 2nd order filters. The program package MINUIT with three different sets of sought parameters has been used for the model pulse fitting. The best results have been obtained at Tau and Ped fixed simultaneously. In this case the 1st and 2nd order filter time resolutions are the same at equal pulse rise time $N*Tau$ and at the same other conditions, and the 1st order filter amplitude resolution is about 20% better than the 2nd order filter one.

The time and amplitude resolutions improve with decrease of thermal noise level and with increase of the pulse amplitude and signal sampling frequency, but depend in different ways on filter time constant Tau. Choice of Tau depends on specific goal: to improve time resolution one has to reduce Tau, while to improve amplitude resolution (for example, at small pulse amplitudes) one has to increase it. The optimal shaper design for PHOS is the 1st order filter with time constant Tau (peaking time) in the range of 0.5 – 1 μs. At sampling frequency 10 MHz one may use 30 – 60 samples to fit the pulse histogram. Then it is seen from the plots in Fig. 5, 6 that the time resolution is equal to 0.24 ns and the amplitude resolution is equal to 0.045% at energy 2.5 GeV that satisfied the PHOS requirements.

The described results give positive answer to the principal question formulated in Introduction. But it does not mean that FEE resolutions will be so excellent at real conditions. Moreover, the method used in this work for processing of the model pulse cannot be used in real measurements: 1) precise MINUIT fit is too slow for on-line data processing; 2) even if the samples will be available off-line, a real pulse shape could significantly and in non-predictable way differ from ideal one. It is evident that we need to invent another method which must be fast enough, less sensitive to the pulse shape, and at the same time must ensure fulfillment of PHOS resolution requirements. A new approach with good prospects have been proposed last year by one of the authors. We are working now on some details of this so called “K-level” method [5]. Nevertheless, we believe that presented results may be useful for an optimal FEE design and as a “reference point” for comparison with results of future data processing developments.

This work was supported by Russian Federal Agency of Atomic Energy, Russian Federal Agency of Science and Innovations, INTAS contract No 03-52-5747.

Acknowledgments

We would like to thank Dr. V. Manko from “Kurchatov Institute” for attention to this work and useful discussions.

References

1. A high resolution electromagnetic calorimeter based on lead-tungstate crystals.
   Nuclear Instruments and Methods in Physics Research A550 (2005) 169-184

2. Trigger electronics for the Alice PHOS detector.
   Nuclear Instruments and Methods in Physics Research A518 (2004) 525-528

3. MINUIT, Reference Manual. CERN Program Library Long Writeup D506, 1992
   See also ROOT Users Guide (Fitting Histograms)
   URL: http://root.cern.ch/root/doc/RootDoc.html

4. Zhong-Bao Yin, Results and Methods of the PHOS FEE Card Evaluation, Sept. 26, 2005

5. A. Kuryakin, PHOS-2005-ALTRO.ppt
URL: http://cmsdoc.cern.ch/~akuryaki/2005/ALTRO/

Attachment

```
// sigmaProbe_1.cxx

// CALCULATE sigmaAmp AND sigmaTdiff FOR GIVEN SET OF PARAMETERS
// Tau AND Ped ARE FIXED

double sigmaAmp, sigmaTdiff;

void probe(int N, double Freq, double Tau, double Amp, double Sigma){

// N      = order of filter
// Freq   = ADC frequency (in MHz)
// Tau    = filter RC (in ns)
// Amp    = pulse amplitude (in ADC bins)
// Sigma = sigma of Gauss noise (in ADC bins)
// pedestal Ped may be any

  if(N!=1 && N!=2){
        cout << "\nN must be 1 or 2!\n\n"; return;
  }

  if(N==1){

        double sigmaAmp0_1 = 0.22126;

        double aFreq_1  = (-0.01808 + 0.74103/TMath::Sqrt(Freq))/sigmaAmp0_1;
        double aTau_1   = (-0.00094 + 0.15657*TMath::Sqrt(1000/(N*Tau)))/sigmaAmp0_1;
        double aAmp_1   =   0.22090/sigmaAmp0_1;
        double aSigma_1 =  (0.06151 + 0.26581*Sigma)/sigmaAmp0_1;

        double sigmaTdiff0_1 = 0.22852;

        double tFreq_1  = (-0.02022 + 0.78469/TMath::Sqrt(Freq))/sigmaTdiff0_1;
        double tTau_1   =  (0.03311 + 0.008863*TMath::Sqrt(N*Tau))/sigmaTdiff0_1;
        double tAmp_1   =  (0.01179 + 107.12/Amp)/sigmaTdiff0_1;
        double tSigma_1 =  (0.04788 + 0.29771*Sigma)/sigmaTdiff0_1;

        sigmaAmp   = sigmaAmp0_1 * aFreq_1 * aTau_1 * aAmp_1 * aSigma_1;
        sigmaTdiff = sigmaTdiff0_1 * tFreq_1 * tTau_1 * tAmp_1 * tSigma_1;
  }

  if(N==2){

        double sigmaAmp0_2 = 0.26261;

        double aFreq_2  = (-0.00224 + 0.83552/TMath::Sqrt(Freq))/sigmaAmp0_2;
        double aTau_2   = (-0.00706 + 0.19062*TMath::Sqrt(1000/(N*Tau)))/sigmaAmp0_2;
        double aAmp_2   =   0.26138/sigmaAmp0_2;
        double aSigma_2 =  (0.05482 + 0.35430*Sigma)/sigmaAmp0_2;

        double sigmaTdiff0_2 = 0.23053;

        double tFreq_2  = (-0.00869 + 0.74515/TMath::Sqrt(Freq))/sigmaTdiff0_2;
        double tTau_2   =  (0.03106 + 0.009070*TMath::Sqrt(N*Tau))/sigmaTdiff0_2;
```

```
        double tAmp_2   =  (0.00764 + 110.66/Amp)/sigmaTdiff0_2;
        double tSigma_2 =  (0.04617 + 0.30564*Sigma)/sigmaTdiff0_2;

        sigmaAmp   = sigmaAmp0_2   * aFreq_2 * aTau_2 * aAmp_2 * aSigma_2;
        sigmaTdiff = sigmaTdiff0_2 * tFreq_2 * tTau_2 * tAmp_2 * tSigma_2;
    }
    cout << "\nsigmaAmp   = " << sigmaAmp;
    cout << "\nsigmaTdiff = " << sigmaTdiff << "\n\n";
}
```