Manual

Pulse Recorder & Analyser

PRA

Version 18.0.0

Marek Dolleiser, Sydney 2018.10.01

Table of Contents

General information

Introduction

The **Pulse Recorder and Analyser** (PRA) application uses computer's sound adaptor to analyse in real time signal received from different detectors, as Geiger counters, proportional gas counters, scintillation counters, semiconductor detectors etc. By using this software you can convert your computer in the multi-channel analyser (MCA), scaler, or simple counter, without buying expensive specialised hardware. Obviously, there are some limitations to the functionality of such instrument, mainly because of sampling rate usually limited to 48000 samples/s. However, if counting rate is reasonably low (less than 1000 pulses/s), the results are very good. PRA was designed for use in experiments with weak radioactive sources which are common in teaching laboratories.

Changes in version 18.0.0

1. Changed: Memory needed by the program to handle pulses is allocated dynamically and its amount can be adjusted by the user. This version is backward compatible with versions 16 and 17 and makes them obsolete. See Data Acquisition and Analysis for more information.

Requirements

PRA can run on any computer with sound adapter and Windows (XP or higher) operating system. It also runs on Linux boxes under Wine but it depends on how well your audio hardware is supported by Linux. Memory requirement depends mainly on the maximum number of pulses and number of audio channels.

Installation

PRA does not need any special installation, just copy “PRA.exe” and “PRA.pdf” files to a folder of your choice and run the application. It is purely portable application and can be run from any location. Sometimes you may need administrator rights to install PRA in some special Windows folders as for example: “Program Files”. For convenience, you can make a link to PRA on your desktop or in the Windows menu, but you will be responsible for removing it if you decide to uninstall the program. To uninstall PRA simply delete “PRA.exe”, “PRA.pdf”, and “PRA.ini” files. The PRA program itself does not write any information to the Windows Registry. PRA comes in two versions: 32 bits and 64 bits. The 64 bits version works only on 64 bits operating system. The 32 bits version works on both: 32 bits and 64 bits operating systems. To read “PRA.pdf” file you need PDF reader application available on your system. If you like to open PRA program automatically when you mouse click over ".pls" or ".wav" file, you have to associate those type of files with PRA program. To do so just click right mouse button over document file and from the menu choose *Open With→Choose Program...* and then *Browse...* and select PRA program file. To make your association permanent check "Always use the selected program to open this kind of file" and confirm your choice.

Users who like typical Windows installation procedure please use PRA installer.

Contact

Should you have any questions, comments or suggestions regarding PRA please e-mail

mdolleiser@optusnet.com.au

New version of the PRA application (if any) can be found at

http://www.gammaspectacular.com/marek/pra/index.html

Hardware

PRA analyses signal from the sound adapter which can be selected from the menu: *Settings → Audio Interface*. Additionally, you may also need to change audio interface settings in your Windows operating system by using *Audio Control Panel* which comes with your audio interface or to use *Windows Recording Control*. To access *Recording Control* use *Settings → Control Panel* from the main Windows menu and then choose *Sound and Audio Devices* controller. Alternatively, you can click over loudspeaker icon with right mouse button and choose *Adjust Audio Property*. I usually use *CD Player* or *Line In* input. Pulses produced by most detectors are quite narrow and hardly can be sampled accurately by a sound adaptor. To fix this problem I use a simple low-pass RC-filter at the input. This solution works well with Geiger counter and NaI scintillation detector. For gas proportional counter used to detect X-rays, I built simple amplifier based on OPA129 mainly to match high output impedance of the detector to low input impedance of the audio input, and also to amplify the signal to the line level ≈ 0.3 V.

Menu

File

The **File** menu contains following items:

1. **Open** – opens data previously saved to a file. Default extension of the file is ".pls". You can also open ".wav" sound file and it will be immediately analysed using the current settings, the same way as if the signal is connected to the sound card input. This mode of data acquisition is only useful when you are adjusting trigger settings because usually, the ".wav" file is larger than the ".pls" file. The advantage of using ".wav" file is that you can open the same file with different settings and to compare the results of analysis. The ".wav" file can be recorded using InTune application. To record stereo files use any good sound recording program as for example Audacity and save it as Windows PCM format (.wav).
2. **Load Data** – opens data file without changing current audio settings, pulse shape, and regions of interest.
3. **Obtain Pulse Shape from Sound File** – uses an existing sound file to determine pulse shape. It is an alternative to the Pulse Shape Acquisition method.
4. **Save** – saves all acquired data and settings to a file, which later can be opened by PRA.
5. **Export Counting Rate vs Time** – saves counting rate vs time histogram as a text file which can be imported into other applications for further data analysis. The column separator is the "tab" symbol. Exported data depends on the Settings.
6. **Export Counting Rate Histogram** – as above but a different histogram.
7. **Export Interval Histogram** – as above.
8. **Export Pulse Width Histogram** – as above.
9. **Export Pulse Height Histogram** – as above.
10. **Export Pulse Height Histogram vs Time** – saves a matrix of pulse height histograms as text file.
11. **Export Regions of Interest** – saves as text file ROIs selected in the Pulse Height Histogram.
12. **Export Coincident Pulses Height** – saves as text file heights of simultaneous pulses from the left and the right channels.
13. **Exit** – quits the application, equivalent to closing the main window.

Action

The **Action** menu contains following items:

1. **Start Data Acquisition** (keyboard shortcut **A**) – starts data acquisition. All current data is discarded.
2. **Continue Data Acquisition** (keyboard shortcut **C**) – starts data acquisition appending

results to existing data. This option should not be used if time dependence consistency is important.
3. **Stop Data Acquisition** (keyboard shortcut **S**) – when you start data acquisition, all data previously acquired is discarded, so always save your data to the file after data acquisition stops. Data acquisition stops automatically if application reaches any of data acquisition limits (see Settings). This menu entry works also if data acquisition was started from Start Pulse Shape Acquisition. In this case, Edit Pulse Shape window will open with applied new pulse shape parameters.
4. **Start Pulse Shape Acquisition** – starts data acquisition with the goal to produce automatically averaged shape of the pulse. You can use this option if you like to average many pulses. In this mode, the program always uses baseline triggering and the pulse shape central point is a top of the pulse. When using this option and your pulse threshold is smaller than 10 arb.u, higher settings of the trigger threshold are automatically applied (10 arb.u. to 90 arb.u.). Please, ensure that polarity of pulses is indicated correctly by the pulse threshold sign.
5. **Sound On/Off** (keyboard shortcut **Space**) – switches on and off playing in real time sound representing arriving pulses. Larger pulses are played with a higher pitch. This implemented in PRA v7 feature, is primarily for visually impaired people, but also can be quite entertaining for others. It is absolutely possible to recognise many of the simple radioactive sources by listening to the sound. I can recognise ^{54}Mn, ^{57}Co, ^{60}Co, ^{133}Ba, ^{109}Cd, and ^{137}Cs using NaI detector.

View

The **View** menu contains following items:

1. **Counting Rate vs Time** – opens the Counting Rate vs Time window.
2. **Counting Rate Histogram** – opens the Counting Rate Histogram window.
3. **Interval Histogram** – opens the Interval Histogram window.
4. **Pulse Width Histogram** – opens the Pulse Width Histogram window.
5. **Pulse Height Histogram** – opens the Pulse Height Histogram window.
6. **Audio Input** – opens the Audio Input window.

Settings

The **Settings** menu contains following items:

1. **Data Acquisition and Analysis** – opens the Data Acquisition and Analysis window.
2. **Calibration** – opens the Calibration window.
3. **Audio Interface** – opens the Audio Interface Settings window.
4. **Pulse Shape** – opens the Pulse Shape Editor window.
5. **Advanced Filter** – opens the Advanced Pulse Filter window.
6. **Load Settings …** – opens and applies saved program settings.
7. **Save Settings** – saves current program settings to the default configuration file “PRA.ini”. This file is saved in the PRA program directory and the user needs to have rights to write to this folder.
8. **Save Settings As ...** – saves current program settings to file at chosen location which you can apply later by “Load Settings ...” command.

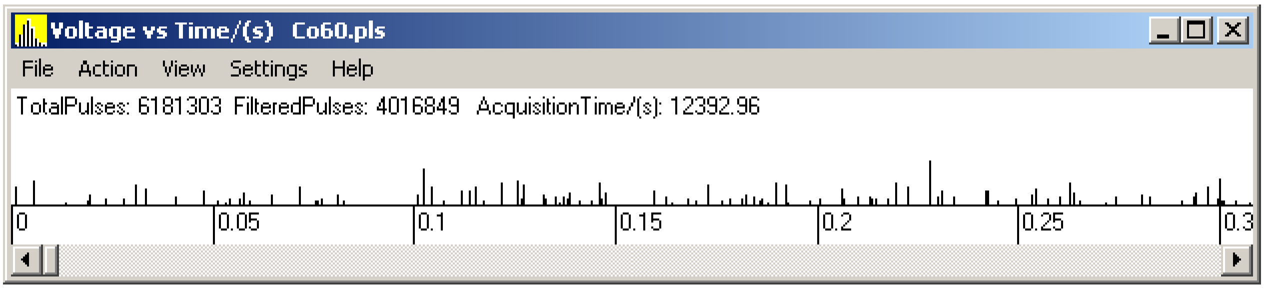
Help

The **Help** menu contains the following items:

1. **Help Topics** – opens “PRA.pdf” help file.
2. **About** – displays version and copyright information about the PRA application.

Windows

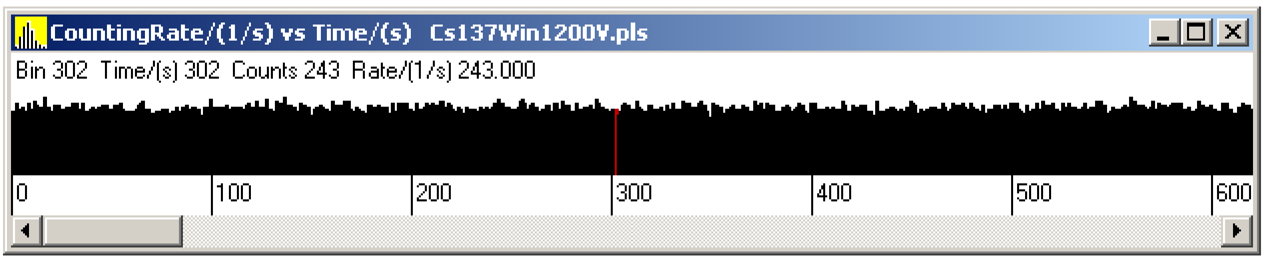
Main



This window shows simplified pulses (only width and height) on the time scale. The most important elements of the window are: the window's title which contains also opened file name (as all other histogram windows), the application's menu, the time scale which can be change by using the ↑ and ↓ keys on the keyboard (when the window is active), and the information bar which displays a total number of recorded pulses, a number of filtered pulses (which is affected by using *Analysed data selection* in the *Data Acquisition and Analysis* window), and the total data acquisition time.

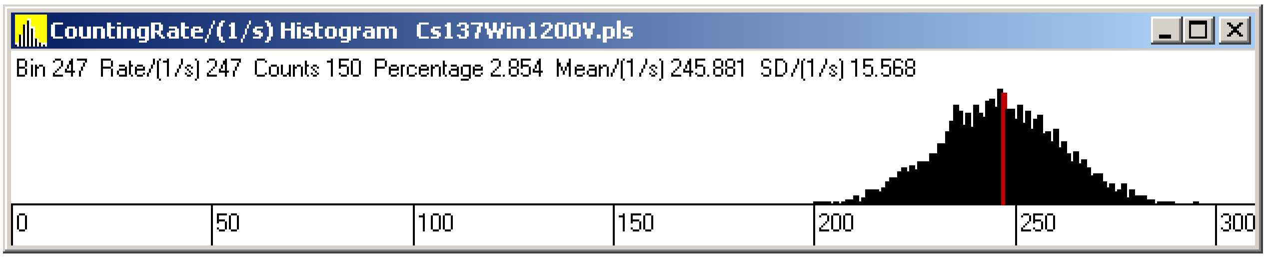
Closing this windows quits the application.

Counting Rate vs Time



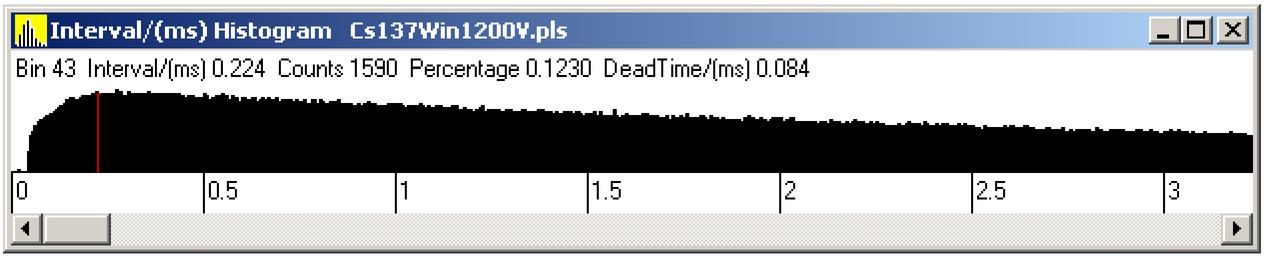
This window displays counting rate as a function of time. Graphically it looks like histogram but it is not. In histogram on the horizontal axis (abscissa) are bins with values of the measured variable and on vertical axis (ordinate) frequency (occurrence) of events (number of counts) in each bin. On this display, the abscissa represents bins (steps) in time and the ordinate shows frequency of pulses (counting rate). *Counting rate vs Time* is very important when analysing decay of radioactive materials. In this case, the *Time step* parameter should be set smaller than a radioactive isotope's half lifetime, but large enough to avoid bins with zero counts if possible. Information about selected bin (indicated by the red colour) is displayed under the title bar.

Counting Rate Histogram



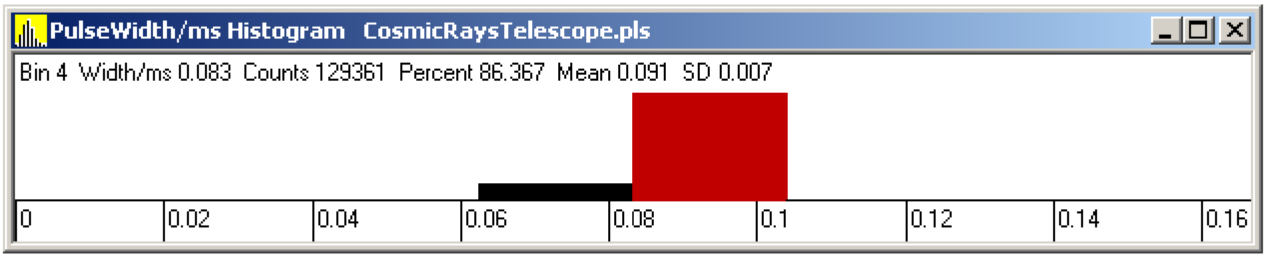
This histogram shows the distribution of the data displayed in the *Counting rate vs Time* window. It is important to set proper parameters for its bin size and *Time step* in *Counting rate vs Time* to build histogram which best represents the data. PRA calculates minimum size of the bin in this histogram and bin size can be the only multiplicity of this value. The displayed information is related to the selected bin (red colour) and the mean value and the standard deviation of the counting rate distribution.

Interval Histogram



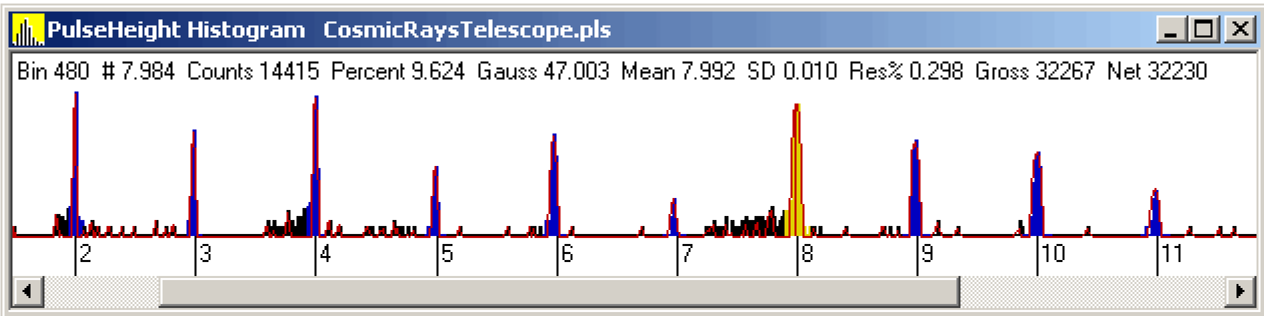
This window displays a histogram of the time interval of pulses. If the S-fold parameter is set to one, it represents the distribution of an interval between consecutive pulses. In this case, if there is not dead time present (which is not true in any real case at least because of a finite pulse width) and average counting rate is constant, the histogram should be of an exponential decay shape. The biggest value should be in the first bin. For real data usually, we observe fewer pulses in a few first bins than expected. PRA uses interval distribution to calculate the dead time of the data acquisition system. The displayed information is related to the selected bin indicated by the red colour. Also, if *S-fold* equals to one, the approximate value of the dead time is displayed.

Pulse Width Histogram



This window shows a histogram of the pulse width. The pulse width is recognised in PRA during data acquisition as the period from a first sample when the voltage exceeds the threshold to the first sample when the voltage drops below the threshold. With such definition, the pulse width depends on all parameters for pulse triggering. Even if it does not tell much about real pulse width (especially if pulse passes through by low-pass filter on the input of the sound interface), it can be still useful. When the Shape tolerance method option is enabled, a pulse width is also affected by the shape threshold value (listed under Shape tolerance method check box) and usually leads to a very short pulse width, which has no expected physical meaning of the pulse width. In such a case pulse width is a good indication of the period when left and right channel pulses are considered to be in coincidence. The displayed information is about selected bin indicated by the red colour.

Pulse Height Histogram



This window displays pulse height histogram. For many detectors, pulse height is proportional to the energy deposited in the detector[1] by an elementary particle such as electron, proton, alpha etc. For this reason, it is very useful to produce spectra of radioactive sources. Pulse height is recognised by PRA during data acquisition. The triggering method and triggering parameters can be set in the Settings window. The pulse height is measured in arbitrary units (100 arb.u corresponds to the maximum value which a sound adapter can sample) and it needs calibration before using it as a multi-channel analyser. The displayed information is about selected bin indicated by the red colour.

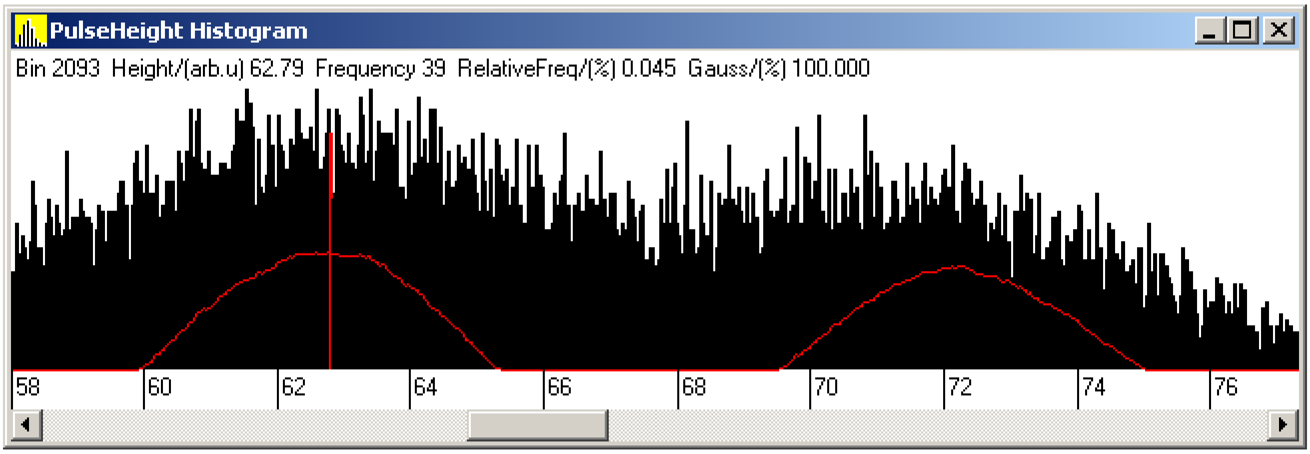
The user can also define up to 32 'Regions Of Interest' (ROI) in each channel. To do this set the current bin where you like ROI to start and then press key **B** and then move the cursor to a position where you like ROI to end and then press key **E**. Alternatively you can select ROI with the mouse. Depress **Ctrl** key and then click and hold left mouse button at the position which you like ROI to begin, drag the cursor where ROI should end and release the mouse button. If a new ROI overlaps any existing ROI(s), the last one is automatically deleted. You can also delete ROI by selecting

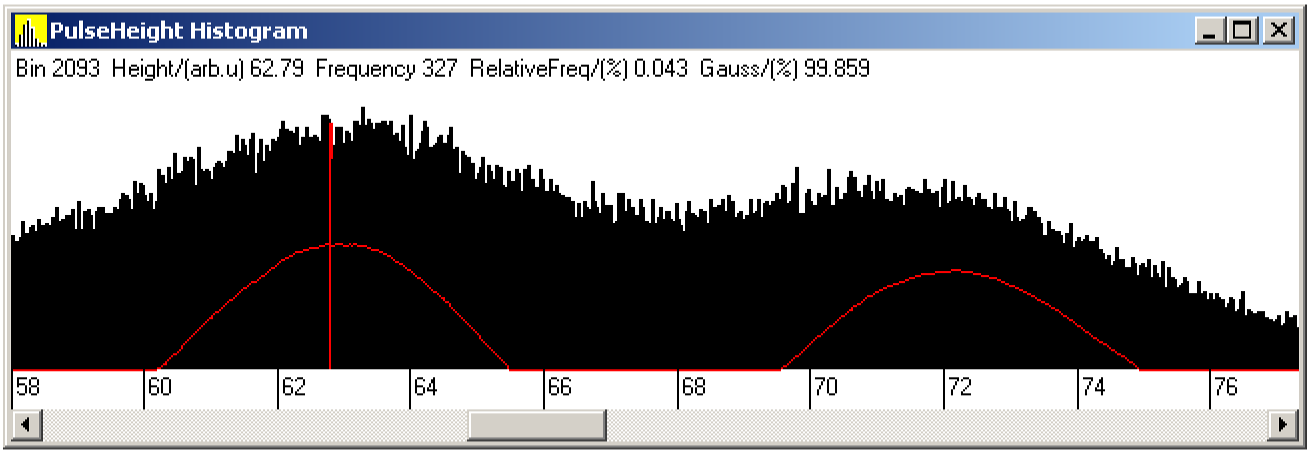
1 The height of pulse from detector depends but is not a function of the energy of detected particle. For example, 662 keV photons from ^{137}Cs source can produce a wide distribution of pulses including Gaussian shaped photo-peak and large Compton effect part.

current bin inside an ROI and pressing **Delete** key. ROIs are indicated by the blue colour. When the current bin is inside ROI the colour changes to dark yellow. Additionally, an information about the peak's statistics is displayed under the title bar. This can be useful to obtain more accurate parameters of selected peak (or deep). Parameters are calculated using standard statistical analysis on the net counts. Net counts are obtained by subtracting the background from the peak. The background is estimated as a straight line connecting first and last sample in ROI.

You can export all ROIs as a text file by using *Export Regions of Interest* from the *File* menu.

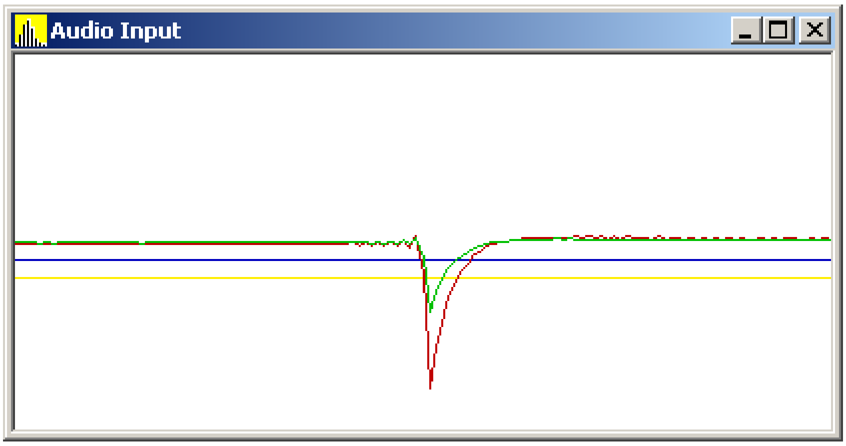
If *Show Gaussian correlation* check box in *Data Acquisition and Analysis* dialogue is checked, then also 'Gaussian correlation' (actually it is a correlation between normalised discrete Gaussian function in range (-2σ, 2σ) and pulse height histogram) is calculated and displayed at pulse height histogram window. Most of the nuclear detectors exhibit a simple dependence of standard deviation (the spread) of the peak and its mean value. The standard deviation is proportional to square root of the mean. Figures below shows pulse height histograms respectively for the small and large number of counts. If current bin is at a position where the Gaussian correlation value is larger than zero, then you can quickly select ROI by pressing **D**, **F**, **G**, or **H** key (each one selects different width of the ROI around the peak).





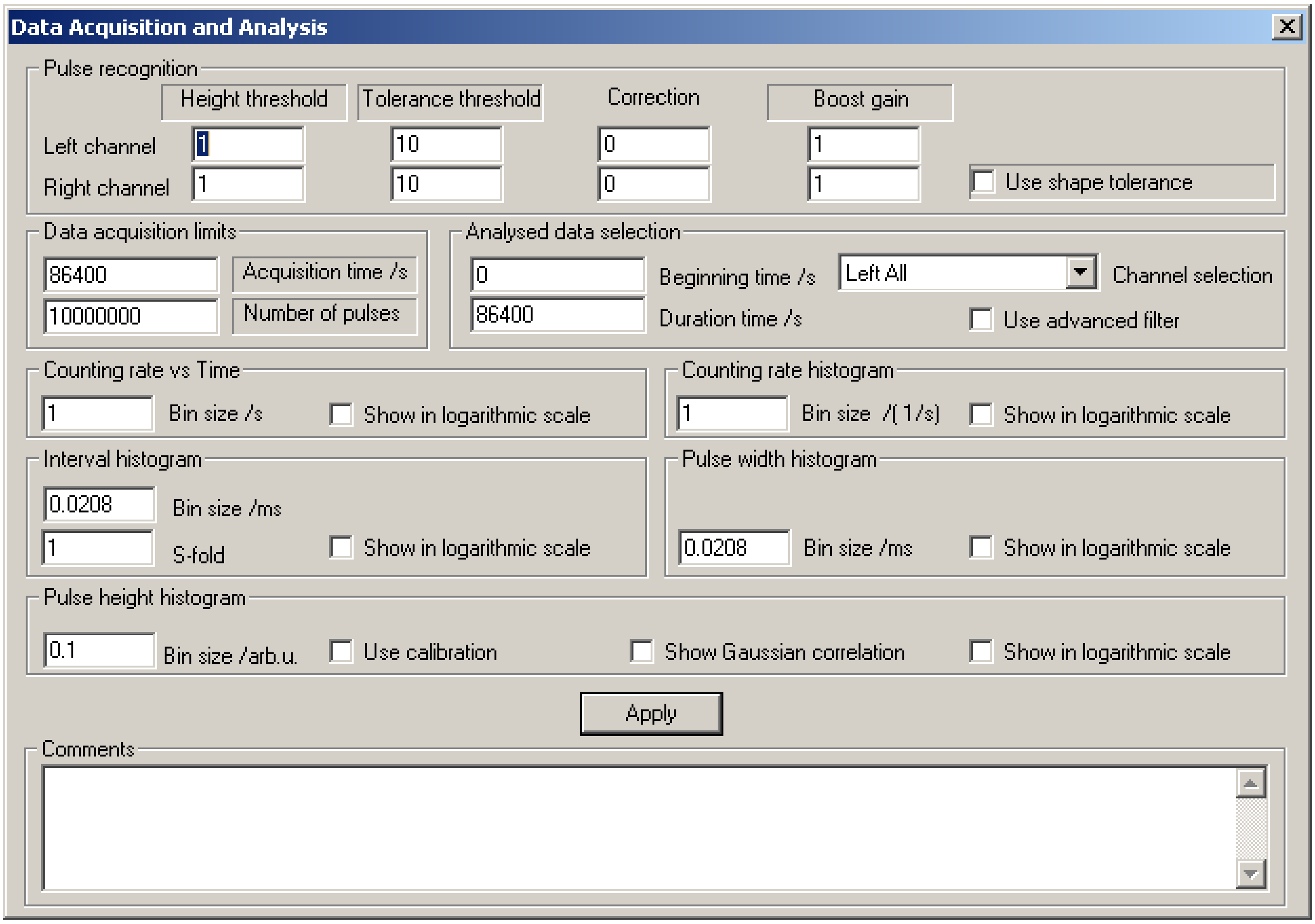
In both cases 'Gaussian correlation' curve (red colour) is quite smooth even so the first histogram is built with many times smaller number of counts than the second one. You can use positions of local maxima of those curves to approximate quite accurately mean values of histogram peaks without using ROI (or to find which part of histogram should be selected).

Audio Input



This window shows a signal at the inputs of the audio interface. Red and green lines show left and right channels respectively. Yellow and blue horizontal lines correspond to the trigger levels. Full height of the display corresponds to the full range of ADC of your sound interface. The picture shows the same signal from BGO scintillation detector applied to both, left and right channels, where boost gain was set equal 2 and 1 on left and right channels respectively. By using “T” key user can toggle between normal (default when starting PRA) auto trigger modes which work the same way as on the standard oscilloscope.

Data Acquisition and Analysis



Important: Controls in this dialogue box whose names are in the static frame are only important during data acquisition. Changing their values after collecting the data have no influence on analysed data. Their values should be set before starting data acquisition. However, for testing only they can be changed during data acquisition without stopping the action. In such a case the resulted data will be not consistent.
The **Data Acquisition and Analysis** window is divided into 9 groups:

1. **Pulse recognition** – you can choose one of two different pulse recognition methods based on a shape analysis or on an absolute value. When the *Use shape tolerance method* is selected, all eight parameters in this group are important. In this case, the application calculates pulse height as a value of cross-correlation function (the sum of products) of sampled values and values defined by the pulse shape. If it exceeds the *Height threshold* value then the current pulse shape and defined pulse shape are compared by means of the sum of squares of differences between the two shapes. If the sum of differences is smaller than the *Tolerance threshold* value, a pulse is counted. The value of the shape tolerance should be set between 0 and 100. The value of 0 corresponds to the shape of the current pulse to be identical with the predefined one. The value of 100 corresponds to the very bad correlation between the current pulse and predefined shape. Proper settings of the shape tolerance value depend on the defined pulse shape, but the value of 10 as the starting point is

a good choice. To obtain very good pulse height resolution the value of tolerance should be very small, say 0.2. Unfortunately, this small tolerance value filters out a lot of the pulses. Version 9 of PRA try to solve this problem by introducing correction parameters which correct pulse height according to the accuracy of the pulse shape. By using these parameters one can obtain almost the same good resolution using a threshold of 10 as with threshold 0.1 without correction. It gives a boost to the number of counts especially at lower pulse heights (lower energy) where pulses are rejected frequently because of a small signal-to-noise ratio. I found that correction parameters in range 50 to 100 are giving the best results, but certainly, it depends on the pulse shape and should be optimised in every case. Version 10 of PRA let you collect the data with a high value of the tolerance threshold where most of the pulses are accepted and then use "Advanced Pulse Filter" to select required range later. If you do not like pulse height correction, set it to 0.
If the *Use shape tolerance method* is not checked, only pulse height threshold parameters are taken into account to trigger the pulse. In this case, the value of the baseline is calculated over many samples and a difference between the top of the pulse and the baseline becomes a pulse height.
The sign of the threshold decides about the polarity of triggered pulses: plus for positive and minus for negative. As an example, a typical pulse shape taken from the gas proportional counter is shown below. The shape is mostly dependent on the amplifier. To produce quickly pulse shape you can use *Action → Start Pulse Shape Acquisition* command**.**
New in version 11 are *Boost gain* parameters. Their values can be set in range 1 to 100. It is an additional adjustment of the sound interface gain which does not change the gain set in the sound recording control panel. This feature can be useful if the observed pulses are smaller than 10 arb.u. and the gain control is already set to maximum. Also, fine adjustment of the gain in the left and the right channel can be handy.

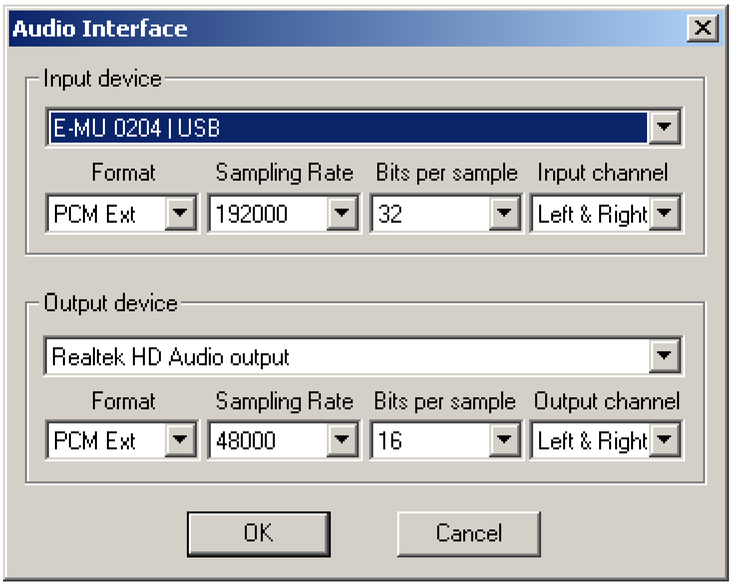
2. **Data acquisition limits** – you can set a maximum value for acquisition time (not larger than 3 155 692 600 s which is 100 years) or a number of pulses (maximum 36 000 000 in 32 bit Windows or 100 000 000 in 64 bit version). If your data acquisition process exceeds any of set limits it will stop immediately. When you are setting limit of number of pulses consider to choose number of pulses which you really need to be recorded and number of audio channels which you are going to use, save settings and quit PRA. When you start program again only memory which is necessary to handle that number of pulses will be allocated. For example total memory used by PRA for 1 000 000 pulses limit and single audio channel is 83 MB and for 100 000 000 pulses is and two audio channels is 4.54 GB. 100 000 000 pulses is enough to keep recording at rate 1000 pulses per second for almost 28 hours.
3. **Analysed data selection** – you can set beginning and duration of the data that you like to select for analysis. It is usually good practice to start data acquisition earlier than some interesting event is going to happen (for example a sudden increase in counting rate) and after stopping data acquisition select an only interesting section of the acquired data. Also, you can select the audio channel from which histograms are built and displayed. For example, if you like to see only those pulses from the left channel which occurs at the same time as another pulse in the right channel (time coincidence), choose *Left if Right*. Advanced events filtering including pulse height, pulse distortion, and the pulse width is possible by checking "Use advanced filter" and editing entries in "Advanced Pulse Filter" dialogue box window.
4. **Counting rate vs Time** – there is only one parameter which you can change: the time period that you used to calculate counting rate. There is also checkbox which you can use to display statistics in logarithmic scale.
5. **Counting rate histogram** – as for all other following histograms there is *Bin size* parameter

and *Show in logarithmic* scale check box which you can edit.
This histogram uses the *Counting rate vs Time* data and it is affected by its *Time step* parameter. For radioactive source which does not change its intensity, the counting rate can be described as Gaussian or Poisson distribution, depending on chosen bins' size and intensity of the source.

6. **Interval histogram** – you can change additionally *S-fold* parameter. If the *S-fold* parameter is one then the interval between every two consecutive pulses is counted, if it is two, then the interval between every first and third pulse is counted etc. Possibly the smallest bin size is selected automatically after opening ".pls" or ".wav" file and after starting data acquisition with a new sampling rate.
7. **Pulse width histogram** – you can change two settings as with other histograms. The bin size is set automatically the same way as for *Interval histogram*. If you are using pulse shape analysis the width of the pulse is very short (usually one or two sampling periods) and it does not correspond to the real pulse's width.
8. **Pulse height histogram** – the settings are identical as for *Pulse width histogram* but pulse height is analysed rather than pulse width. Additionally, there are two check boxes: *Use energy calibration* and *Show Gaussian correlation*. If the *Use energy calibration* check box is checked the *Pulse height histogram* uses parameters from *Energy Calibration* dialogue box where you can specify parameters for the left and the right audio channel. This check box also is affecting data which is exported using *Export Pulse Height Histogram* menu. If *Show Gaussian correlation* is checked, then the result correlation curve of predicted Gaussian distribution and pulse height histogram is also shown in the *PulseHeight Histogram* window.
9. **Comments** – contain a text which is saved with the data. PRA automatically is including the beginning and the end of all data acquisition sessions, for example: <2017/07/23 16:45:46 0, 2017/07/23 16:45:51 5.5><2017/07/23 16:46:20 5.5, 2017/07/23 16:46:28 13.5>. This indicates that data acquisition was done in two sessions, one started in 23th of July 2017 at 4:45:46 pm and ending the same day at 4:45:51 pm and second data acquisition session started the same day but at 4:46:20 pm and finished at 4:46:28 pm. The first session was 5.5 s long and the second session extended total acquisition time to 13.5 s. Of course you can edit comments to your needs before saving the data.

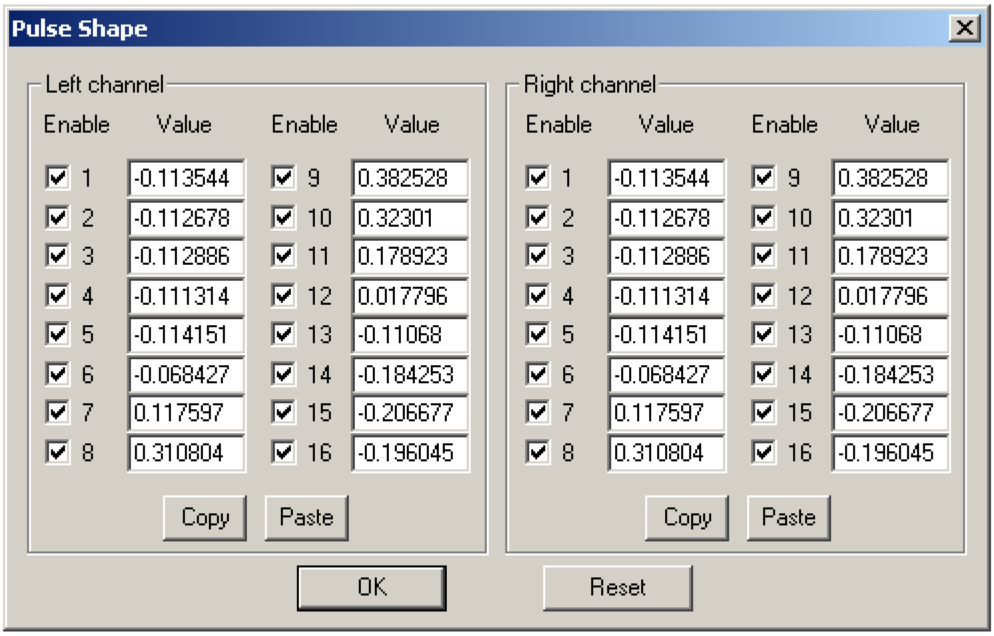
The *Apply* button applies settings and the *Save* button additionally saves them as default in the "PRA.ini" file. The *Save As* button let you save PRA settings under a different name. This file can be later loaded by using the *Load* button. From version 4.2 settings files contains also ROIs, from version 5.0 also audio settings, and from version 6.0 calibration data points and comments. This can sometimes produce a problem if your audio device has been changed. In such a case just delete a "PRA.ini" file and start the program again.

Audio Interface Settings



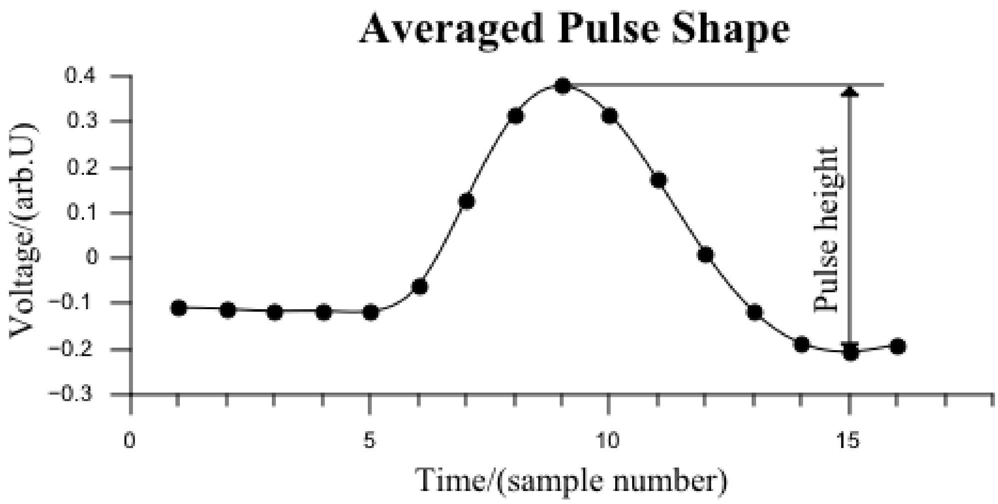
Use this dialogue box to select audio device and its parameters for data acquisition and playback. Higher sampling rates at the input are preferred for better height pulse resolution and higher counting rates. If your sound device does not truly support higher sampling rates, do not use this option[2]. It is interesting that some settings are not supported as expected from hardware specification. For example, when I tried my "E-MU 0204" interface, it supported well all options when bits per sample was set to 16, but only stereo mode was working for 24 bits and 32 bits. For output, I recommend to use 48000 sampling rate and preferably on the different output device to avoid sound distortion when your hardware is not able to play smoothly. Popular sound interfaces build in motherboard perform well with both input and output set to 48000 samples per second.

Pulse Shape Editor



2 Windows drivers sometimes artificially support higher sampling rate by interpolating additional samples.

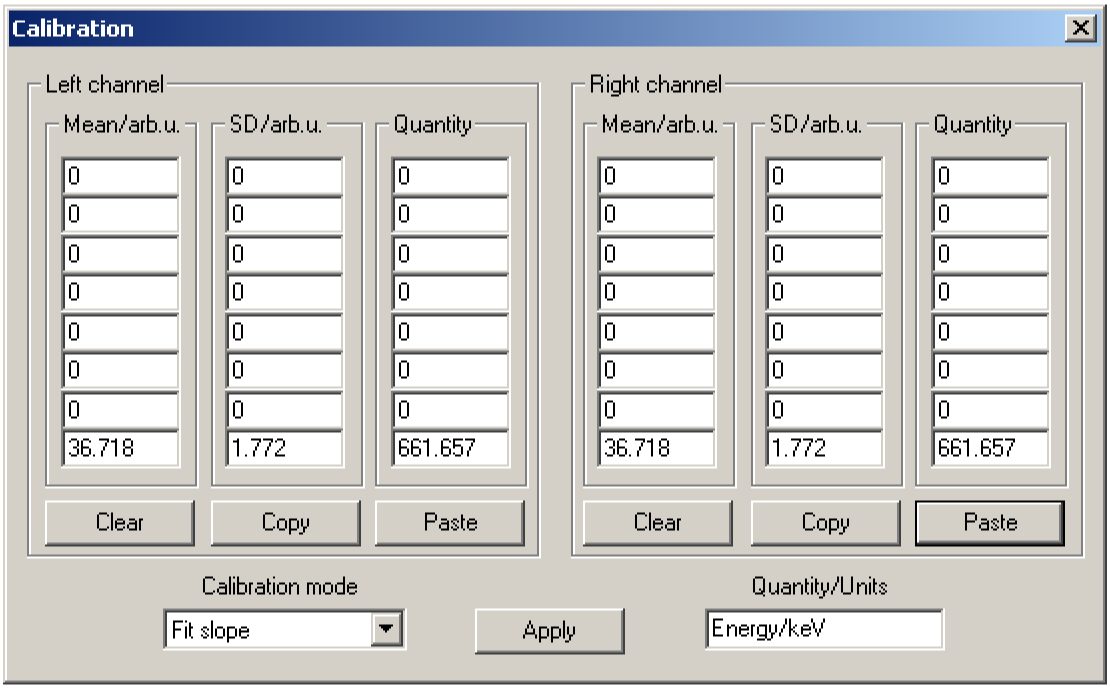
The Edit Pulse Shape window allows editing shape of the pulse which can be used to analyse the waveform. For each channel, there are 16 consecutive sample values of which at least two different values should be enabled. To copy or paste all data in one channel use *Copy* or *Paste* buttons. The pasted data is limited to 1×16 matrix if the clipboard contains more columns or rows. The values are automatically normalised[3] when you click the *OK* button. The current pulse shape is saved in "PRA.ini" file by means of the *Save* (or *Save As*) command from the *Data Acquisition and Analysis* window and will be automatically loaded next time you open the application. From version 5.0 of PRA also data file ".pls" includes pulse shape and it becomes current pulse shape when you open previously saved file. The *Reset* button discards all changes and sets the last accepted values.
Pulse shape shown in the above image was taken from my NaI detector with the left channel connected through low pass filter and right channel straight to the audio input. Comparing first five samples in each channel shape one can notice small oscillations[4] in the right channel are almost totally suppressed in the left channel.



The pulse shape presented by the above image was constructed by averaging a few pulses from the waveform recorded with the InTune application. One can use *Action → Start Pulse Shape Acquisition* menu to achieve a similar effect. The shape can be set to look for leading or falling edge of the pulse. The purpose of the pulse shape analysis is to increase spectrum resolution by rejecting badly formed pulses (as it is in a case with partially overlapping pulses) from statistics. Also, the noise (usually masking a low amplitude pulses) can be significantly reduced.

3 The mean value of the data points to be 0 and sum of its squares to be a difference between the maximum and minimum value of the data points.
4 Those oscillations are introduced by digital anti-aliasing filter in the audio interface to remove frequencies above half of the sampling rate (Nyquist frequency)

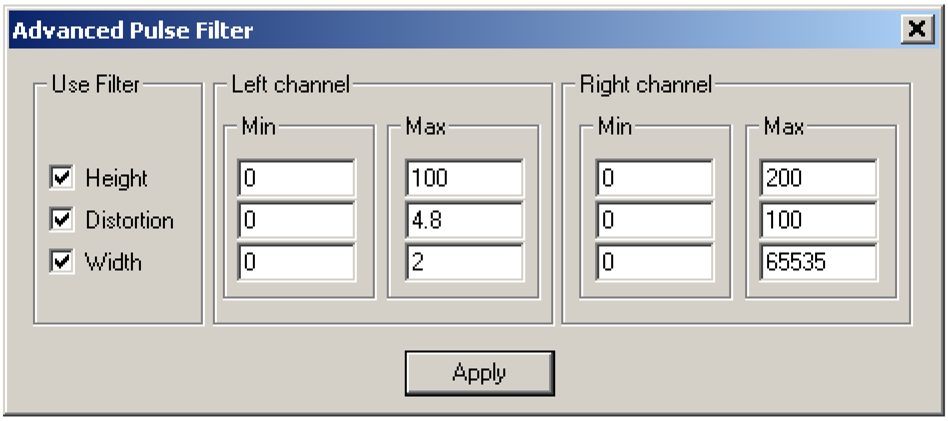
Calibration



In this dialogue box, you can specify up to eight calibration data points for each channel. *Clear* buttons set all values in the corresponding channel to zero. To copy or paste all data in one channel use *Copy* or *Paste* buttons. When you copy, data on the clipboard is in text form which consists of three tab-separated columns. Paste function works with a clipboard containing text with column separated by tabs, spaces, and semicolons. The number of columns and rows pasted from the clipboard is limited to 3×8 matrix. Also, you can choose calibration mode from *Fit slope*, *Fit linear,* and *Interpolate* options. The popular and recommended for most of the detectors is a linear fit. Similar to linear is slope fit, where calibration line is forced to go through (0,0) point. This can be useful for example if you calibrate your system using a source which has only peaks at higher energy and no low energy peaks are available. PRA uses also standard deviation of calibration peaks (column *SD*) for weighting calibration points and for Gaussian correlation parameters. *Interpolate* option is for calibrating detector systems where energy versus pulse height dependence is nonlinear. PRA uses simple linear interpolation between calibration points. Energy beyond the last calibration point is extrapolated from the two largest data. If you have just one calibration point, all three method gives you the same result. Text edit box *Quantity/Units* lets you assign a different name (maximum 31 characters) for quantity and units which you are using. PRA is mainly used for acquiring energy spectra, but you have freedom to assign other quantity and units. For example, if you measure automatically sizes of small particles by using an electronic device which converts their diameter to pulses whose height depends on particle size, you can set *Quantity/Units* text to Diameter/µm. When you click *Apply* button, all entries are checked for consistency[5] and accepted values are propagated to the pulse height histogram.

5 All three values in any row can be simultaneously zeroes or non-zeroes and larger pulse height must correspond larger energy. At least one row for each channel must contain non-zeroes data.

Advanced Pulse Filter



This dialogue box provides PRA users with opportunities to select additional parameters of the pulses which can be used to produce all histograms. This is a filter which does not change any information contained in *.pls file. Especially selecting ranges of pulse width and pulse distortion can be useful to find proper values for "Tolerance correction" to increase the resolution of the pulse height histogram. This filter, when used during data acquisition affects also 'played' pulses.

Mathematics in PRA

In this chapter I would like to show[6] mathematical formulas working behind the scenes in PRA. Some of them are commonly well-known and some are designed by myself specifically for use in PRA. I think this could be useful for PRA users to better understand how the program works and for programmers to use similar solutions in their own software.

Pulse Recognition

Baseline method

There are two methods implemented in PRA to recognise and analyse pulses. The first method is standard. First, we are looking for the baseline of the input signal. In PRA the baseline is calculated as an average of all samples in a buffer (usually few hundred samples) which are smaller than zero (assuming that the pulses are positive).

$$base=\frac{\sum_{i=1}^{i=max} y_i \cdot \varepsilon_i}{\sum_{i=1}^{i=max} \varepsilon_i}$$

where:

max is the number of samples in the buffer,

y_i is value of i^{th} sample,

ε_i equals to 1 if $y_i < 0$ and 0 otherwise.

Then PRA checks if $y_i - base$ exceeds the pulse threshold if so, corresponding ***i*** is marked as the beginning of the pulse. Now PRA is looking for the maximum value of $y_i - base$ until it becomes smaller than the pulse threshold. Then the corresponding ***i*** become the end of the pulse and the maximum value of $y_i - base$ its height.
This method of pulse recognition is simple, fast, and it works fine for pulse counting when good

6 It could be pleasure and challenge for students and other mathematically minded PRA users to proof some of those formulas.

pulse height resolution is not important (Geiger-Müller counter). Poor pulse height resolution is the result of changes in the baseline value and the fact that only one sample decides about pulse height.

Pulse shape method

The second method of pulse analysis was designed to achieve the best possible pulse height resolution. I call it a 'pulse shape method'. This method relies on comparison of a specially prepared 'standard pulse shape function' with a small part of consecutive signal samples. The comparison is a simple dot multiplication of standard pulse shape vector and signal samples. PRA uses up to 16 points long[7] discrete standard pulse shape function, which can be edited by hand or obtained automatically from the signal input. By editing pulse shape by hand the user can achieve similar effects of pulse recognition as in the previous (baseline subtracting) method or commonly used in commercial MCAs pulse integration method. However, the most powerful is the procedure where pulse shape representation is obtained automatically from the signal. In this case, PRA is looking for local maximum in input samples and saves it together with 8 samples before maximum and 7 samples after. All those samples are averaged and then 'normalised' to make useful standard pulse shape representation. In PRA normalisation is a linear function applied to the pulse shape to achieve two special properties:

1. Sum of all 16 pulse shape numbers must be equal to zero. This is done by subtracting the average. This feature of pulse shape function makes the result of the dot product to independent of any constant shift of the baseline.
2. The length of pulse shape vector (sum of squares of all pulse shape values) must be equal to the difference between its maximum and minimum components[8]. This makes pulse height to be of similar value as pulse height obtained using baseline method.

In a more formal mathematical way this can be represented by formulas:

$$\bar{s}=\frac{\sum_{k=0}^{k=15} s_k}{16}$$

$$S_k=\frac{(s_k-\bar{s})\cdot h}{\sum_{k=0}^{k=15}(s_k-\bar{s})^2}$$

where:

s_k is k^{th} value of pulse shape before normalisation,

S_k is k^{th} value of pulse shape after normalisation,

h is a difference between the maximum and minimum value of pulse shape (the height of pulse shape itself). The reader can check that after this normalisation procedure $h=\sum_{k=0}^{k=15} S_k^2$ become invariant under normalisation (it does not change after next normalisation) and is equal to the difference between the maximum and minimum components of the standard pulse shape vector.

7 The number 16 is arbitrary and chosen to achieve good pulse height resolution and relatively small calculation load.

8 Usually vectors are normalised to one.

Now we can use our pulse shape to find dot product of our standard pulse shape and vector consisting of 16 consecutive samples:

$$h_i = \sum_{k=0}^{k=15} y_{i+k} \cdot S_k$$

where:

i is a running index along samples in the input buffer,

h_i is a dot product of standard pulse shape vector and 16 consecutive samples vector.

PRA uses h_i as a new value of the sample and is looking if it is larger than pulse threshold to mark the beginning of a pulse and then identify pulse height and pulse width the same way as in the baseline method (base=0).

Pulse height calculated using pulse shape method is more accurate than when using baseline method, but still, it can count pulses which shape is quite distorted. To rectify this problem PRA uses pulse shape filter to count only those pulses whose shape is similar enough to the standard pulse shape. The procedure is very simple. Each time when h_i has been found to exceed pulse threshold PRA normalises 16 samples starting from y_i the same way as standard pulse shape. Then these 16 samples are subtracted from the standard shape, each difference squared and added together. The result is a good measure how both, standard and current pulse shapes are similar. When it is zero, the shapes are identical. If this value is larger than the pulse shape tolerance threshold, the pulse is rejected.

$$t_i = \sum_{k=0}^{k=15} (Y_{i+k} - S_k)^2$$

where:

Y_{i+k} is normalised 16 consecutive samples vector (starting from the i^{th} sample),

t_i 'tolerance' is a measure of the difference between the current samples and the standard shapes.

Calibration

Signal samples from the sound adapter which are analysed by PRA usually come as 16 bits, 24 bits, and 32 bits integer numbers. PRA converts all of them to a real number from (-100, +100) interval. Those numbers do not have any physical units (they are proportional to the input voltage) and PRA uses for them so-called arbitrary units (arb.u.). Pulse height is also expressed in arbitrary units. The height of pulse coming from detector usually depends on energy deposited by elementary particle inside the detector.[9] This dependence is not very strict and for certain energy deposited in the detector, there is possible the whole range of output pulse heights which produce a peak in pulse height histogram. The shape of this peak can be described by so-called normal or Gaussian distribution. By calibration, we understand the dependence of the position of this peak and corresponding energy deposited in the detector (which very often is a full kinetic energy of the detected particle). For example, well-known source ^{137}Cs produces gammas with the energy of 662 keV. If this gamma interacts with the detector and all its energy is deposited, then pulses corresponding to this event will form a Gaussian peak on pulse height histogram. If the mean value of this peak is said 20 arb.u. then our calibration function will assign 662 keV to 20 arb.u. and PRA

9 This is the most common use of PRA, but you can use a different transducer which produces pulses dependent on different physical quantity, not necessary energy.

will display abscissa on pulse height histogram in keV rather than arbitrary units. PRA uses three methods of calibration.

Fit slope

This is the simplest (and very good) method to calibrate the detector. Most of the detectors show quite good proportionality between energy deposited in the detector and pulse height. In this case function $y = a\,x$ will be a good choice. PRA uses weighted least squares method to find parameter ***a***. From *Calibration* dialogue box we have up to 8 calibration entries for each channel:(*Mean* of the peak in arbitrary units, Standard Deviation *SD* in arbitrary units, *Quantity* in quantity units).

Let's name it as (x_i, σ_i, y_i) .

Using least squares method we can find

$$a = \frac{\sum_i \frac{x_i y_i}{\sigma_i^2}}{\sum_i \frac{x_i^2}{\sigma_i^2}}$$

where ***i*** is running over all (non-zero) calibration data points.

Fit linear

In this case fitting function is linear: $y = a\,x + b$. This is more general form than fitting slope and should be used if you know well some low energy calibration points.

The parameters ***a*** and ***b*** are calculated using formulas:

$$a = \frac{\sum_i \frac{x_i}{\sigma_i^2} \sum_i \frac{y_i}{\sigma_i^2} - \sum_i \frac{x_i y_i}{\sigma_i^2} \sum_i \frac{1}{\sigma_i^2}}{\left(\sum_i \frac{x_i}{\sigma_i^2}\right)^2 - \sum_i \frac{x_i^2}{\sigma_i^2} \sum_i \frac{1}{\sigma_i^2}}$$

$$b = \frac{\sum_i \frac{x_i}{\sigma_i^2} \sum_i \frac{x_i y_i}{\sigma_i^2} - \sum_i \frac{x_i^2}{\sigma_i^2} \sum_i \frac{y_i}{\sigma_i^2}}{\left(\sum_i \frac{x_i}{\sigma_i^2}\right)^2 - \sum_i \frac{x_i^2}{\sigma_i^2} \sum_i \frac{1}{\sigma_i^2}}$$

Interpolation

This calibration method should be used if your detector is not linear e.g. calibration function slope is changing from low energies to high energies. In this case, PRA uses simple linear interpolation between calibration data points including the point (0,0) at the beginning and extrapolating the last pair of calibration data points for higher energies. Linear interpolation for *x* lying between two calibration data points $x_i \le x \le x_{i+1}$ can be expressed as: $y = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} x + \frac{x_{i+1} y_i - x_i y_{i+1}}{x_{i+1} - x_i}$.

Dead time calculations

Dead time is an unavoidable property of any detector and data acquisition system. Dead time is a minimum time between two consecutive pulses to be recognised by data acquisition system. The dead time of the system can be affected by detector itself (as recovery time in GM counter, scintillation time in NaI crystal or plastic scintillator), analogue electronics (amplifiers, filters, and shaping devices), and digital electronics (including ADC, data storage time etc.).

In the case of our "PRA" program, the mainspring to the dead time is the width of the pulse which can't be too small to be properly sampled by the computer's sound card. PRA uses time interval statistics to find the dead time of the system. A manifestation of dead time phenomenon in time interval histogram are empty and less populated than expected bins at the beginning. If counting rate is constant, we are expecting that distribution of time interval will follow exponential decay curve with the maximum at first bin.

How PRA is analysing this histogram to find a dead time of the apparatus?

1. We are looking for the bin with a maximum number of counts. Let's name corresponding time interval ***T***.
2. Next we are counting all events in regions $[0,T)$, $[T,2T)$, and $[2T,3T)$. Let's call them $s_{0,}$ $s_{1,}$ and s_2 correspondingly.
3. Now we can calculate dead time ***W*** using a formula:

$$W \approx \left(1-\frac{s_0 s_2}{s_1^2}\right)T$$

Region Of Interest statistics

Region Of Interest (ROI) is any range of consecutive bins in pulse height histogram. Usually, we use it to select a peak. The most interesting parameters calculated by PRA are the mean value, standard deviation, gross number of counts, and net numbers of counts. The gross number of counts in ROI is just a sum of all counts in selected region. Usually, the peak is not built with the base at zero but is sitting on the top of what we call a background (not necessary coming from a background radiation). By background, we understand a slowly changing profile of the histogram. The net number of counts is calculated by subtracting background counts from gross counts. PRA assumes that background under the peak is changing linearly. To calculate this line's parameters firstly we take a few (10% of the width of ROI) samples around the beginning and calculate their average value and do the same at the end of ROI. Next, we subtract the area of the trapezoid which is built on those new points from the gross number of counts to get the net number of counts. The mean value and standard deviation are calculated using net counts which are more meaningful that the raw counts. More precisely we can describe the whole process using formulas:

$$Gross=\sum_{i=R_b}^{i=R_e} c_i$$

$$\overline{a_b}=\frac{\sum_{i=R_b-0.1(R_e-R_b)}^{i=R_b+0.1(R_e-R_b)} c_i}{0.2(R_e-R_b)+1}$$

$$\overline{a_e}=\frac{\sum_{i=R_e-0.1(R_e-R_b)}^{i=R_e+0.1(R_e-R_b)} c_i}{0.2(R_e-R_b)+1}$$

$$Net = Gross - \frac{(\overline{a}_b + \overline{a}_e)(R_e - R_b + 1)}{2}$$

$$n_i = c_i - (\overline{a}_b + \frac{(i - R_b)(\overline{a}_e - \overline{a}_b)}{R_e - R_b})$$

$$Mean = \sum_{i=R_b}^{i=R_e} \frac{n_i}{Net} i$$

$$\sigma = SD = \sqrt{\sum_{i=R_b}^{i=R_e} \frac{n_i}{Net} (i - Mean)^2}$$

for $Net \neq 0$

where:

R_b is the beginning bin of ROI

R_e is the end bin of ROI

c_i is a number of counts in the i^{th} bin

n_i is a net number of counts in the i^{th} bin

σ is a standard deviation of the peak in the ROI

Gaussian correlation filter

In many cases, the only purpose of pulse height histogram is to find characteristic photoelectric peaks which can help to identify radioactive isotopes in your source. The best method is to collect data for such long time that pulse height histogram becomes a very smooth curve and all peaks become clearly visible. The problem using this method is obvious; usually, you have to collect your data for a very long time. This produces another problem with long term stability and temperature dependency of your detector. Some people try to solve this problem by smoothing data using simple averaging consecutive bins. Although it can sometimes help to see peaks which can be hidden under normal statistical data fluctuation, but usually it also leads to the identification of false peaks and smears the data. This method is equivalent to just larger bin size. A new method used by PRA is filtration of the pulse height histogram data in the more scientific way. Most of the nuclear detectors are working according to simple principle; the number of detectable particles produced in the detector is proportional to the kinetic energy of the primary particle which enters the detector. In the case of scintillation detectors, it will be a number of photons produced during scintillation, in the case of gas proportional counter it will be a number of positive ions and electrons or in the case of semiconductor a number of electron-hole pairs. A transducer attached to the detector (photomultiplier, thin wire anode or electrode) produce a signal which is proportional to the number of detectable particles. This number undergoes certain statistical fluctuation which can be described as Poisson distribution. This is a distribution which has a very simple and useful property; its standard deviation is a square root of the mean value. We will use this property to identify the peaks in our pulse height histogram. If the number of detected particles in one pulse is large, then Poisson distribution can be simply replaced with a normal (Gaussian) distribution which is more convenient for calculations. We still assume that peak standard deviation is proportional to the square root of the mean value (mean bin of the peak or energy of detected particle). PRA does all calculations automatically in real time, you have just to tell what is the standard deviation for peaks that you use for calibration (this is the main reason for a new column in the calibration dialogue box). For filtering peaks PRA uses a similar technique as for filtering pulses from the sound signal, it performs just dot product operation of pulse height histogram data and especially normalised Gaussian shape.[10] We use the discrete Gaussian shape in range $(-2\sigma, 2\sigma)$ which covers 95% of the peak. The Gaussian shape is normalised this way that the sum of all its values is zero (subtracted average value) and the sum of all squares of the (new) values is one. Because of this normalisation and the fact that Gaussian is symmetrical with respect to the mean, the dot product does not depend on data shift up or down as well as when a peak is lying on the slope or not. This makes our operation result not sensitive to the background counts. The dot product of Gaussian shape (with standard deviation dependent on the bin number) is performed along all data in the pulse height histogram. The result is called here Gaussian correlation and its value is printed on the same pulse height histogram display as red colour points. All negative values of the Gaussian correlation are plotted as zero. Gaussian correlation curve is usually very smooth even with not so large number of counts in the histogram bins and its local maximum quite accurately indicate mean values of the peak. PRA uses scaling to display Gaussian correlation curve to be well visible together with pulse height histogram data.[11] You can use page up and page down keys to change vertical scale or just use a logarithmic scale for your convenience to make the Gaussian curve better visible. PRA uses standard deviation which is calculated from all non-zero SD values in the calibration dialogue box. The distance between points where Gaussian correlation curve under the peak crosses zero approximately equals to 2.6 standard deviations (a little larger than FWHM). All this can be more precisely described using mathematical formulas:

$$\text{for } k \in (-2\sigma_i, 2\sigma_i) \quad Gs_k = \exp\left(\frac{-k^2}{2\sigma_i^2}\right)$$

$$Gs_k = Gs_k - \overline{Gs}$$

$$Gs_k = \frac{Gs_k}{\sum Gs_k^2}$$

$$Gc_i = \sum_{k=-2\sigma}^{k=2\sigma} c_{i+k} Gs_k$$

where:

σ_i is an expected standard deviation of the peak with mean value at i^{th} bin,

Gs_k is the k^{th} element in Gaussian shape,

Gc_i is the i^{th} element of the Gaussian correlation curve,

c_i is number of counts in the i^{th} pulse height histogram bin.

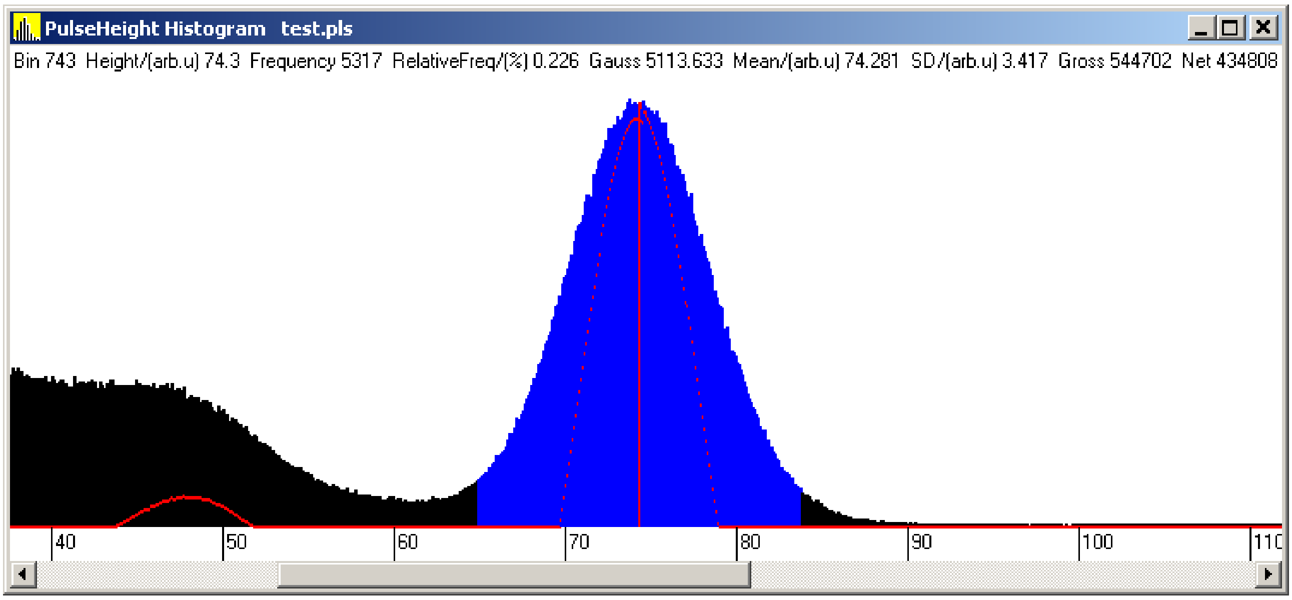
Tolerance correction

Many PRA users have a problem how to find and use “Tolerance correction” parameter. To rectify

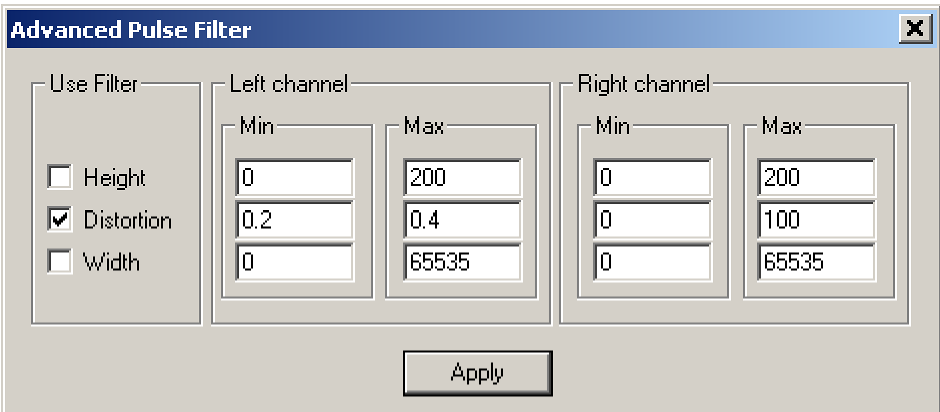
10 This operation can be also named as calculating the discrete correlation of the histogram and Gaussian filter.

11 If the standard deviation is chosen properly, then Gaussian correlation value at the centre of the peak should be in range (or slightly below) of the number of counts (subtracting the background). However, it is not crucial to use the right value of the standard deviation. If SD is too small, then Gaussian correlation curve shows more irregularities. This can be even beneficial to recognise two close peaks.

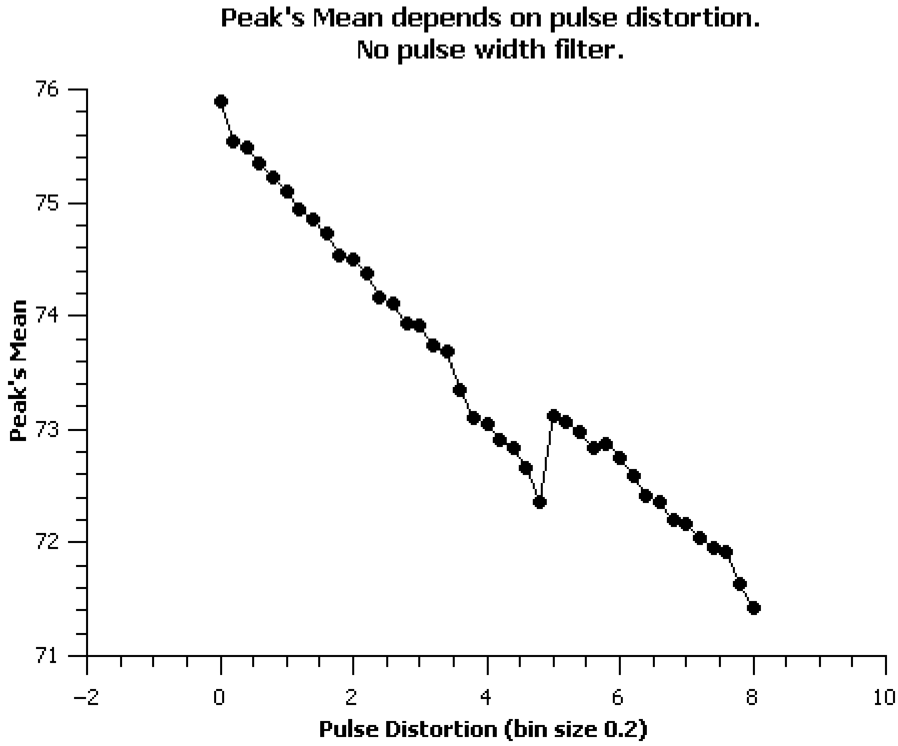
the problem I will shortly show how to do it in PRA version 10 using “Advanced pulse filter”.



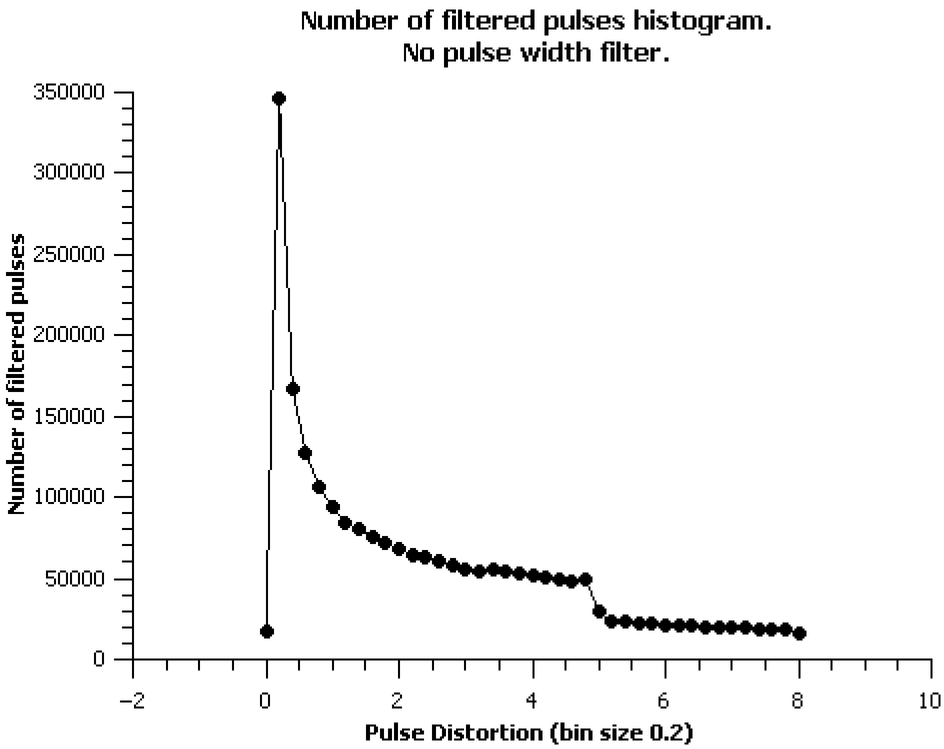
1. After establishing pulse shape, I started recording pulses from NaI detector using the Cs-137 source. By looking at “Counting Rate vs Time” I found that setting 10 for “Tolerance threshold” does not change counting rate. I recorded 10000000 pulses. Pulse height histogram without using any filter or correction is shown below. My NaI detector has around 10% resolution.
2. Then I used to select ranges of pulse distortion: 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, etc. “Advanced Pulse Filter” for range 0.2 to 0.4 is shown below.



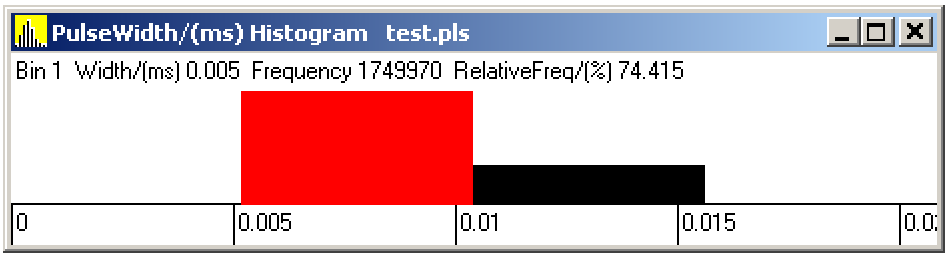
3. For each of the selected distortion range, I read “Mean” and “Gross” values of the 662 keV photo-peak and make a scatter graphs using QtiPlot (just advertising the good program, you can use MS Excel, LibreOffice Calc, Gnumeric or similar).



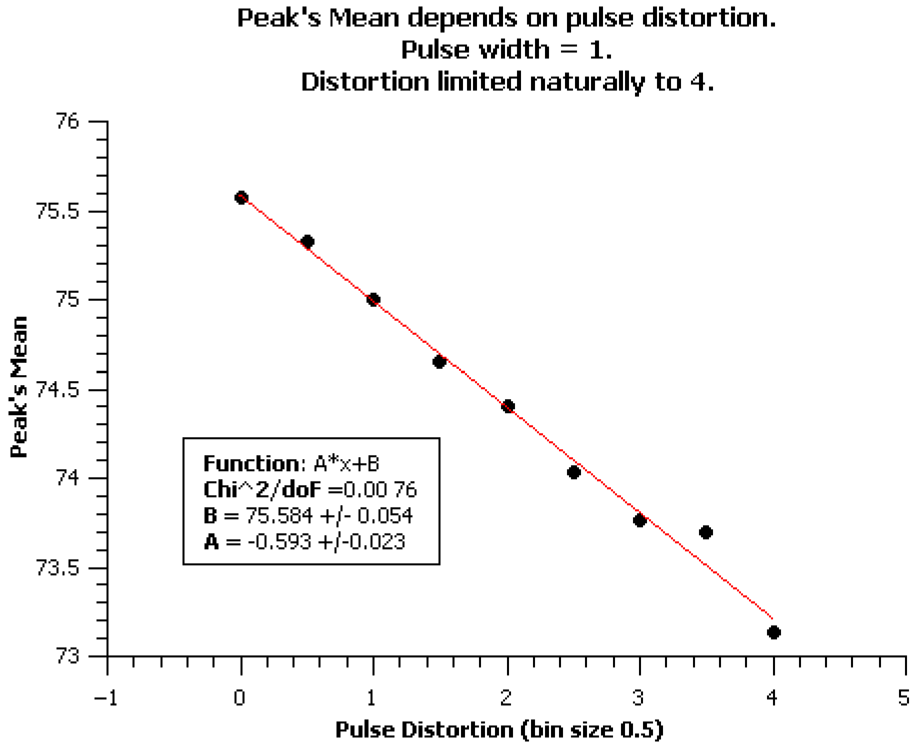
4. As one can see, peak's mean value is changing approximately linearly, except a small discontinuity around pulse distortion at 5.



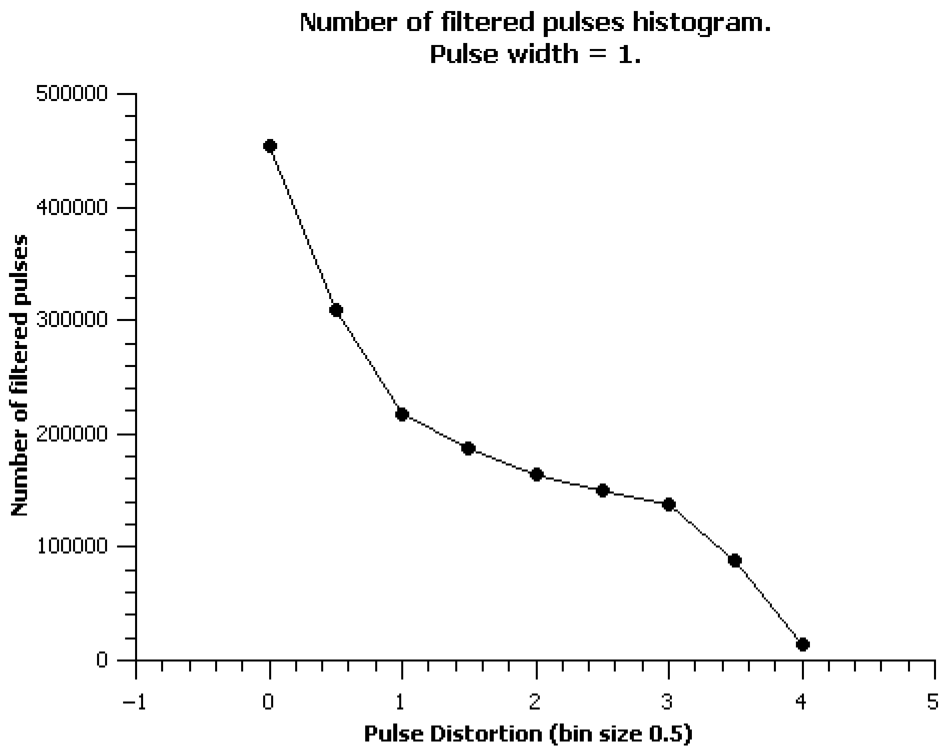
5. Similar behaviour represents a number of filtered pulses. The sudden change in a number of counts at distortion = 5 is due to different pulse width. Most of the events are of width = 1, but a small amount has width = 2.



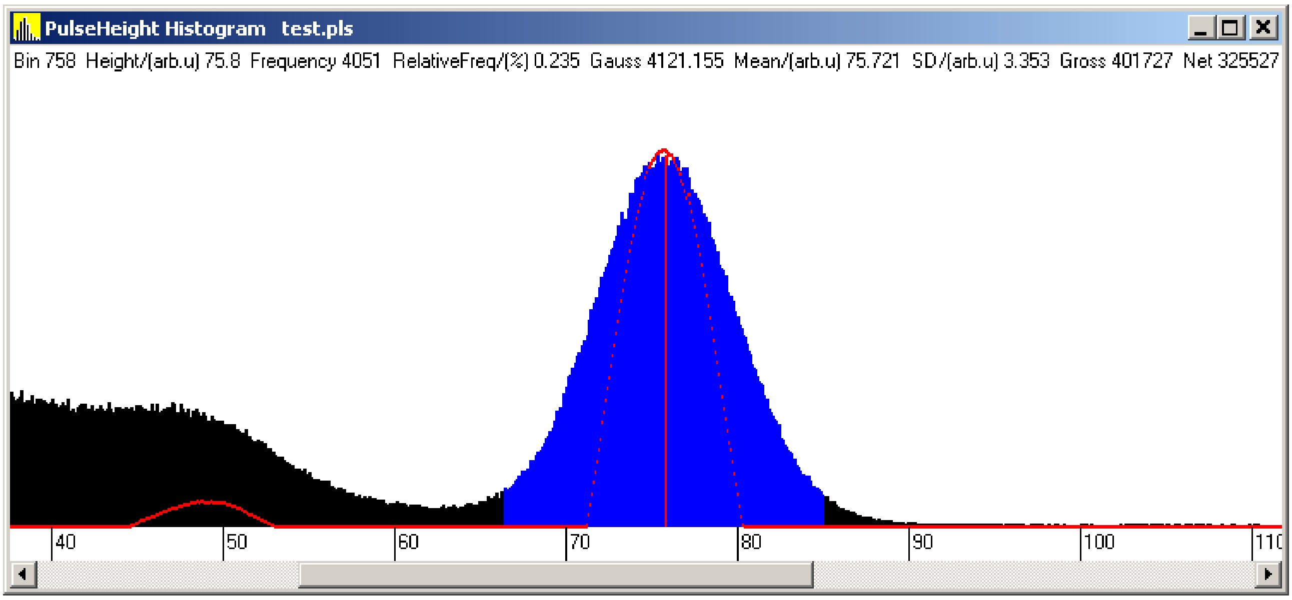
6. When I selected only events with pulse width = 1, I had this plots for mean and number of counts.



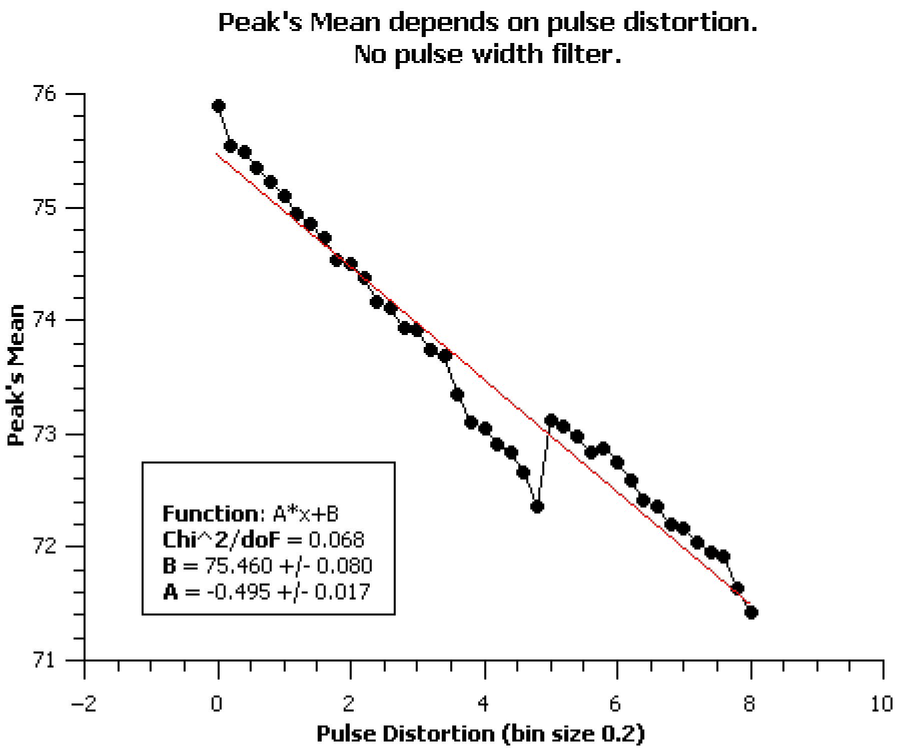
7. This time I used bin size 0.5 just to save time.



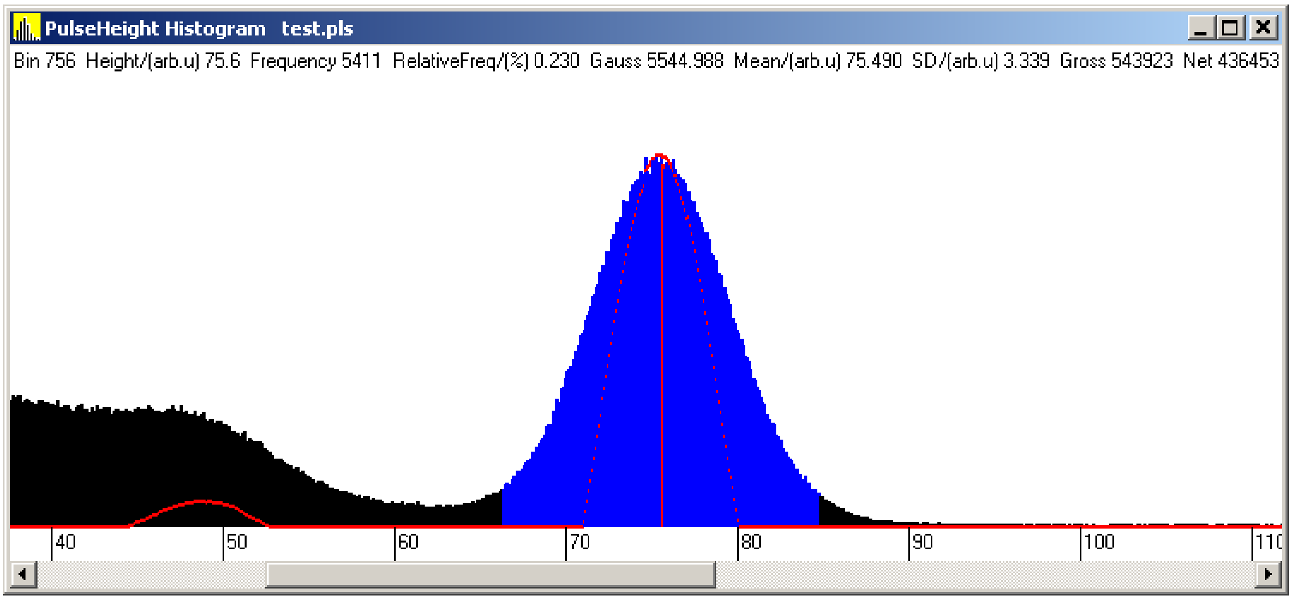
8. A number of counts drops to zero above distortion = 4.
9. From linear fit to mean vs distortion plot, I found tolerance correction by using the equation, tolerance correction = -A*10000/B = 78.46. The factor 10000 comes from increasing both, pulse distortion (tolerance) and tolerance correction 100 times.



10. As we can see the standard deviation of the peak is improved from 3.417 to 3.353, but we lost some counts; 544702 vs 401727. Those lost pulses were painful for me and I went back to the full range of pulse width and fit a straight line to the mean vs distortion graph despite discontinuity.



11. Correction factor calculated from this data comes as 0.495*10000/75.460 = 65.60. After applying this tolerance correction (without any additional pulse filtering), I saw the result.



12. As you can see standard deviation is even smaller than with pulse width filter (3.339 vs 3.353) and the gross number of counts in the peak is almost the same as without any correction (543923 vs 544702). The small difference in a number of counts in the peak comes just from more narrow peak (see Gaussian correlation).

Keyboard Shortcuts

PRA uses following commands which can be accessed by keyboard shortcut:

1. → **(right arrow) –** increase current position in an active histogram by one.
2. ← **(left arrow) –** decrease current position in an active histogram by one.
3. ↑ **(up arrow) – horizontal zoom out**.
4. ↓ **(down arrow) –** horizontal zoom in.
5. **Page Up** – vertical zoom in. Can be also done with the Mouse Wheel rotation.
6. **Page Down** – vertical zoom out.
7. **A –** start data acquisition.
8. **C –** continue data acquisition.
9. **S –** stop data acquisition.
10. **Space** – Start or stop 'playing' pulses.
11. **.** (period) – Rises the pitch of the played sound.
12. > – Increases the rate of the "played pulses".
13. **,** (comma) – Falls the pitch of the played sound.
14. < – Decreases the rate of the "played pulses".
15. **B** – mark the beginning of 'Region Of Interest' in Pulse Height Histogram.
16. **E** – mark the end of 'Region Of Interest'.
17. **K** – mark all ROIs in the right channel identical to the left channel ROIs.
18. **D** – mark automatically very wide ROI around Gaussian correlation peak (D for double).
19. **F** – mark automatically wide ROI around Gaussian correlation peak (F for full).
20. **G** – mark automatically medium ROI around Gaussian correlation peak (G for Gaussian).
21. **H** – mark automatically narrow ROI around Gaussian correlation peak (H for half).
22. **L** – toggles automatic selection of last entry in "Counting Rate vs Time" window.
23. **T** – toggles normal and auto trigger modes in "Audio Input" window.
24. **Delete** – deletes ROI at the current position in pulse height histogram.
25. **Delete** + **Shift** – deletes all ROIs in pulse height histogram.
26. **Q** – opens *Information* dialogue box which displays active window's information as text which can be read by "JAWS" program. To quit this dialogue box press *Enter* key.
27. **F1** – Activates all "Save" menu items during data acquisition without stopping. This status will be reset to 'normal' when you restart data acquisition.

File Structures

All files exported from PRA are in tab separated text format (ASCII).
Other files are in binary formats as follows:

Main data file *.pls starts from the header:

```
typedef struct PlsHeader {
        unsigned long       PulseTag; //"Puls"
        unsigned long       NumberOfPulses[2];
        long long           AcquisitionTime; //numberOfTimeFrames
        unsigned long       PRA version;
        unsigned long       DataOffset;
        unsigned long       Filler;
        SETTINGS            AllSettings; //5888 bytes
}PLSHEADER; //5920 bytes
```

following by 'numberOfPulses' pulses:

```
typedef struct Pulse {

        long long           Begin;
        unsigned long       Width;
        unsigned long       Height;
        unsigned long       Distortion;
        unsigned long       Coincidence;
  }PULSE; //24 bytes
```

Settings file (*.ini):

```
typedef struct Settings {
        AUDIOSETTINGS   CurAudioSettings;//2*16 bytes
        double          SettingsData[18];//18*8 bytes
        char            CheckBox[9];//9 bytes
        char            CoincidenceMode;//1 byte
        char            FilterCheckBox[3];//3bytes
        char            Comment[1007];//1007bytes
        WINDOWPLACEMENT     WinPlacement[7];//7*44=308
        double          Filter[2][16];//32*8 bytes
        double          PulseShape[2][16];//32*8 bytes
        char            ShapeCheckBox[2][16];//32 bytes
        double          CalibrationData[2][8][3];//2*8*3*8 bytes
        char            EnergyUnits[31];//31 bytes
        char            CalibrationMode;//1 byte
         ROI            Roi[2][32];//64*56 bytes
 }SETTINGS;//5888 bytes
```