

Contents

1	Current monitoring	
	<i>Real-time particle identification</i>	2
1.1	Motivation	3
1.2	Requirements	3
1.3	Device specifications	4
1.4	Pulse parameters	4
1.5	Applications	5
1.6	Description of the firmware	6
1.6.1	Analysis module	7
1.6.2	Vector cleaning for area and width measurements	8
1.7	Control and data interface	11
1.7.1	Software	11
1.7.2	Data readout	11
1.8	Performance results	13
1.8.1	Comparison between the charge- and current-sensitive spectroscopy	14
1.9	Source calibration	14
1.9.1	Source measurements - scatter plots	18
1.9.2	Discussion	26
1.9.3	Space charge build-up	27
1.10	Neutron reactor measurements	29
1.10.1	Thermal neutron spectroscopy - TRIGA reactor	29
1.10.2	Fast neutron spectroscopy - CROCUS reactor	30
1.11	Conclusion	33

Chapter 1

Current monitoring

Real-time particle identification

Diamond sensors have a very fast signal response due to their low capacitance. The electrical signal created by drifting charge carriers retains its shape without significant distortion. When the sensor is used together with a fast current amplifier with a high broadband limit (~ 2 GHz) and a readout device with a similar limit, the information about the drifting charges is retained. For instance, a proton creates the free e-h pairs along its trajectory. The electrons and holes start drifting immediately. Those closest to the electrodes recombine quickly whereas those at the opposite side contribute to the induced signal for longer. The resulting signal is therefore a triangular pulse with a steep rising edge and a gentle falling edge. It is possible to determine the drift velocity of the charge carriers by measuring the width of the pulse, as was done in chapter ???. Furthermore, it is possible to determine with a certain probability what is the type of incident radiation, judging by the shape of the induced pulse. This, however, only applies to sCVD diamond material. Its uniform carbon lattice allows the ionisation profiles to retain their shape, unlike in pCVD material, laden with grain boundaries, or in even in silicon where the shape is deformed due to p-n junction non-uniformities.

This chapter describes an application that carries out particle identification by means of the pulse shape analysis. It was developed for measuring activity of a neutron reactor. In this case the device has to be able to filter out the photon background with a rate several orders of magnitude higher than the neutron rate. Overall detected rate in a neutron reactor can easily exceed 10^8 particles $\text{cm}^{-2}\text{s}^{-1}$, depending on the distance of the detector from the reactor core. The device has to be able to cope with such high rates. It also needs to be dead time free or at least close to that, to minimise the counting error. At these rates, it still has to be able to identify the types of pulse. This type of online analysis cannot be done in software. It has to be implemented in an FPGA.

1.1 Motivation

Pulse shape analysis (PSA) is a common software tool for analysing sensor response to incident particles. It is usually done by means of software that runs over big amounts of data that have been acquired and saved to storage. This offline analysis can be repeated and improved. However, the saved data take up a lot of storage space. In addition, saving raw waveform data requires a system capable of a high data throughput and fast data storing. For instance, an oscilloscope can save up to 100 signal waveforms per second. This means that there is a high measurement dead time. To avoid the high dead times, the software algorithms can be ported to the FPGA where they analyse the incoming signal in real time. The signal is then discarded and only the analysis results are saved, decreasing the storage space significantly.

The offline pulse shape analysis has already been used for particle identification with a diamond sensor [1]. An effort has been made to implement an online and real time application for this analysis by porting the algorithms into an FPGA. This section first describes the device specifications Then it describes in detail the PSA algorithms and the structure of the code. Afterwards it discusses the performance results, which showcase the limitations of the device. Finally it describes the data acquired with radioactive sources and in neutron reactors.

1.2 Requirements

Chapter ?? shows that the shape is heavily dependent on several factors, such as environmental temperature and received irradiation dose. At temperatures lower than 150 K the signal from an α starts deteriorating due to recombination of charges in the charge cloud. Sensor irradiation, on the other hand, introduces charge traps, which cause the signal to decay exponentially. These two factors are a significant limitation for particle identification. Priming can improve the charge collection and longterm stability of the pulse shapes. To improve the measurement further, a high bias voltage has to be applied, increasing the measurement SNR.

Factor	Operating range
Sensor material	sCVD diamond
Sensor thickness	500 μm
Temperature	150 K – 400 K
Radiation dose	$1 \times 10^{13} \text{ neq cm}^{-2} \text{ s}^{-1}$
Charge carriers	holes
Bias voltage	$\sim 1 \text{ V } \mu\text{m}^{-1}$
Signal-to-noise	5

Table 1.1: Limitations to particle identification

1.3 Device specifications

The ROSY box has a single BNC input with the termination $50\ \Omega$ or $1\ M\Omega$ with a DC or AC coupling. The analog chain has a 250 MHz bandwidth limit. The input range can be set from $\pm 50\ \text{mV}$ up to $\pm 5\ \text{V}$. The signal offset can be set to any value within this range. The ADC samples this signal with an 8-bit precision at a rate of up to 5 GSPS. The PSA uses the highest sampling to achieve width measurement resolution of 0.2 ns. The spectroscopic application does not need such a fine timing resolution and therefore operates at a reduced sampling rate of 0.8 ns. The amplitude resolution depends on the chosen input range, but at 256 ADC counts per sample, it can be as low as $0.39\ \text{mV s}^{-1}$ at the range of $\pm 50\ \text{mV}$ and as high as $39\ \text{mV s}^{-1}$ at the range of $\pm 5\ \text{V}$.

The logic structure of the PSA is designed using VHDL and runs on Xilinx Virtex 5. The PSA is capable of a maximum counting rate of 1.56×10^8 pulses per second, yielding a 6.4 ns double pulse resolution. The analysis is more time consuming; the maximum throughput rate of the pulse shape analysis is $\sim 5 \times 10^6$ pulses per second. This means that after every pulse, the device has a dead time of approximately (200 ± 15) ns, depending on the width of the pulse being analysed. Any pulse arriving during the analysis of the previous one will be counted, but not analysed. Any two pulses with the distance between the rising edges lower than 6.4 ns will be counted as a single pulse.

The device is very sensitive to noise pick-up. Therefore the setup must be designed to minimise the pick-up by means of proper shielding, use of high-quality cables etc. The relatively low bandwidth limit filters out some high-frequency noise, but not the ringing or higher noise spikes. That is the task for the PSA.

1.4 Pulse parameters

A signal pulse on the input is parametrised during the analysis process. The PSA measures its amplitude, area, width and the slope of its falling edge. The amplitude is the difference between the baseline and the highest sample in the pulse and is given in ADC counts as an 8-bit value. The area is defined as the sum of amplitudes of all samples between two defined boundaries within the pulse. The width is defined as the number of samples with a value higher than a set amplitude threshold. If the threshold is at half the maximum amplitude, the resulting width is *full width at half maximum* (FWHM). The falling slope is the maximum negative difference between values of two samples and is given in ADC counts per sample. These parameters can also be used as *qualifiers* for accepting or discarding a pulse. All four parameters limited by the low and high limit are called a *qualifier set*. For instance, a rectangular pulse by an α particle will always have the same FWHM and a very steep slope. In comparison, a photon will have a lower falling slope value and a narrower FWHM. Therefore the low and high cut on these two qualifiers will make it possible to discriminate between the two pulses. Another qualifier is a *form factor* and is defined as the

ratio between the measured area and the amplitude multiplied by the width. By comparing the measured and the calculated area the difference between a triangular and a rectangular pulse can be inferred.

1.5 Applications

The FPGA firmware is designed for systems instrumented with CIVIDEC amplifiers and CIVIDEC sVCD diamond detectors. Three applications are available: *Spectroscopy*, *Pulse Shape Analysis* and *Counter*, each optimised for a specific task. Their capabilities are described below. The firmware runs in ROSY, a readout system produced by CIVIDEC.

Spectroscopy is a tool for measuring energy spectra of radioactive sources. It is used in combination with the CIVIDEC Cx spectroscopic charge amplifier. The signal from the charge amplifier is analysed in real time. The FPGA measures the maximum amplitude of the signal. The amplitude value is ready at the end of the pulse and is stored in the amplitude histogram. Immediately after, the analysis is reset and the system is ready for a new acquisition. Upon request from the software, the histogram is read out, during which the analysis is paused. In addition to the histogram building, the firmware can also store raw pulse waveforms, which can be then read out by the software. The maximum allowed throughput is 1 million counts per second.

Pulse Shape Analysis is a tool for measuring energy spectra of radioactive sources, with an additional feature. It can identify the type of radiation detected by the diamond detector. By means of the pulse analysis it can subtract the background radiation and only measure the signals from the defined radiation source. It is used in combination with the CIVIDEC C2 fast current amplifier. The firmware receives a current pulse from the detector and digitises it. The pulse is then analysed and parametrised. The analysis module measures its maximum amplitude, full width at half maximum (FWHM), baseline amplitude, falling slope and its area. Then it compares the obtained pulse parameters with the qualifiers set by the software and determines what type of radiation hit the diamond detector. Depending on the qualifiers, the pulse can either be *accepted* or *rejected*. The firmware then stores the parameters of the analysed pulse into histograms. Two histograms exist for each parameter: one for all pulses and one for accepted pulses. In addition, there is one 2D histogram (a scatter plot), which can plot two parameters one with respect to the other. Upon request from the software, all histograms are read out, during which the analysis is paused. The maximum allowed throughput is 1 million counts per second.

Counter is a tool that measures the count rate and the mean time during counts. It is used in combination with the CIVIDEC Cx, C6 or C2 amplifier. It contains one histogram which holds the information about the mean time during counts. The

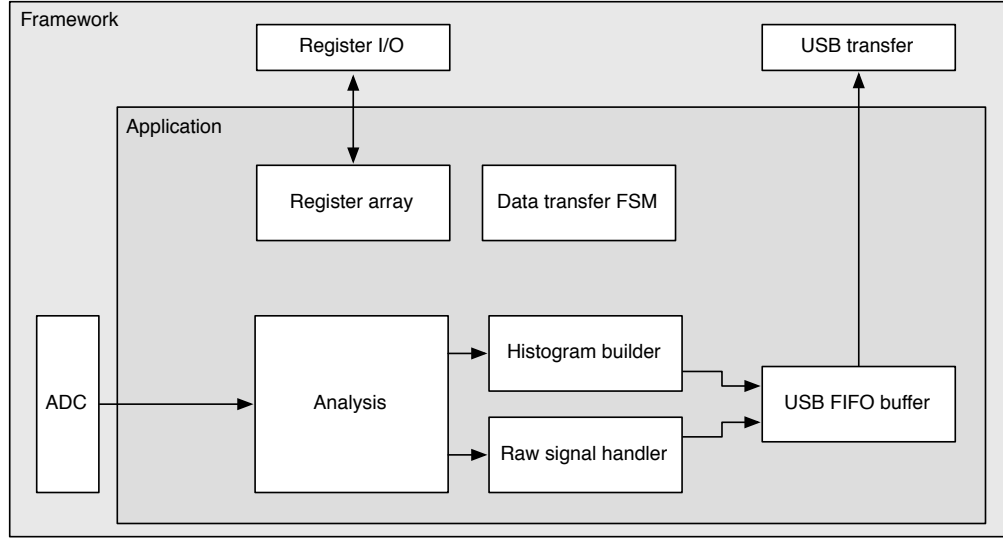


Figure 1.1: Firmware design structure

counter is operational also during the readout of the histogram. The highest counting rate with enabled histogram writing is $3 \times 10^7 \text{ s}^{-1}$.

1.6 Description of the firmware

The applications are built on top of the Picotech platform. The base code handles the communication between the software and the hardware. Furthermore, it provides the interface to the ADC data, the input/output registers and the USB data transfer. The applications have a set of modules that handle the data input and output and a module for signal analysis (see figure 1.1).. The data handling modules are very similar in all the applications to ensure compatibility of the communication between software and firmware and the readout data format. The analysis module, however, is different from one application to the other. The data handling layer is the same for all applications and consists of the final state machine (FSM), the histogram builder, the raw signal handler, the USB FIFO buffer and the register array.

The firmware is written entirely in VHDL. The diagram in figure 1.1 shows the module architecture. The ADC provides the module with 32 digitised signal samples every clock cycle (6.4 ns). The signal is routed directly to the pulse analyser and into the raw signal handler. The analyser outputs are connected to the I/O registers and to histogram buffers. Both the histogram buffers and raw signal buffers are connected to the USB FIFO through a multiplexor. The firmware communication to the controller is done via input/output (I/O) registers (control and status registers, counters) and serially via USB (histogram data, waveforms).

1.6.1 Analysis module

This module is different for different applications. The Pulse Shape Analysis (PSA) application has the most complex module design. The spectroscopy application only uses a small part of that design and the Counter application an even smaller one.

The analysis (or parametrisation) is carried out in several steps. The triggering block starts the readout upon signal threshold crossing. The maximum slope of the falling edge is observed. The Amplitude block calculates the pulse height and retains the maximum amplitude while pushing the signal into the pulse buffer. Then the whole pulse is clocked out of the buffer while its FWHM, baseline width and area are measured. Finally, the Form factor is calculated. At the end the Evaluation block takes in all the parametrised information and classifies the pulse according to user-defined cuts.

Triggering module handles signal polarity swapping, triggering on threshold and defining the trigger window. The real-time processing algorithm allows for a positive or an inverted input signal. However, the PSA only handles positive-polarity pulses. Therefore a negative signal is swapped in the *triggering* block. Signal analysis and readout are then triggered when the signal crosses a user-defined threshold. In addition, the signal has to be over the threshold for a defined number of samples. This is to avoid triggering on noise spikes. A double clock cycle delay is used on the signal to make sure that the recorded signal window will include the rising edge of the pulse as well as some baseline before it. A *trigger active* signal marks a window that contains the whole pulse including some baseline signal before and after it. The trigger can be vetoed by three signals: if the pulse analysis is still taking place, if the input signal exceeds the maximum voltage range or if the data transfer FSM is pausing the analysis due to data transfer to the controller.

Amplitude block calculates the pulse height from the difference between the pulse and the baseline. It also finds the position of the maximum amplitude within the clock cycle. It receives 32 8-bit samples from the triggering block every clock cycle. Time delays in the logic prevent it to find the maximum value of the 32 samples within one clock cycle (6.4 ns). Therefore the decision logic has been pipelined in three stages, which means that the final maximum value is ready three clock cycles after the end of the pulse.

Pulse buffer is a FIFO that stores the signal while its amplitude is being measured. At the end of the pulse the FIFO is read out so that the remaining measurements can take place.

FWHM block uses the maximum amplitude to determine the *half-maximum* and to measure the FWHM. To do so, it counts the samples that are above the half-maximum amplitude. However, this method might also count high enough noise spikes before or after the pulse. Hence an improved method, which “cleans” the measurement of unintentional additional noise, has been implemented. It is described in section 1.6.2.

Baseline width block is the same as the FWHM block, but it measures the width either at 50 %, 25 %, 12.5 % or 6.25 %, depending on the setting in the register. It also makes use of the special method described in 1.6.2 to avoid overestimations due to including noise in the measurement.

Area block measures the pulse area by summing up the amplitude values of the samples in the pulse. The boundaries of the summation are defined with the crossing of the amplitude above a certain threshold. Only the samples between those boundaries are summed up. The boundaries can be set at 50 %, 25 %, 12.5 % or 6.25 % of the maximum amplitude of the pulse. The area measurement makes use of the same routine as the FWHM and Baseline width block to remove the potential outlying samples.

Falling slope block measures the highest negative difference between amplitudes of two adjacent samples, thus getting the maximum negative slope of the pulse. It is an experimental routine, only used for academic purposes.

Form factor block is used as a special qualifier for particle identification. It compares the weighted measured area of the pulse with its weighted calculated “form”, which is defined as the multiplication of the measured amplitude and baseline width. The equation is as follows:

$$x \cdot a - y \cdot A \cdot BW \geq 0, \quad (1.1)$$

where a is the measured area, A is the amplitude, BW is the baseline width and x and y the weighting factors for the measured and calculated area, respectively. The output of the block is the boolean result of this equation.

Evaluation block takes in all the parameters from the analysis blocks and compares them against the user-defined qualifiers. If the parameters are within the bounds, the pulse is accepted, otherwise it is rejected. The corresponding counters within the block are incremented.

1.6.2 Vector cleaning for area and width measurements

The routine for measuring pulse area and width was designed to improve the measurements with respect to the previous implementation. The core difference is that the new routine precisely defines the edges of a pulse. It does so by means of *vector cleaning*, presented in figure 1.2. An important input, beside the ADC data and the measurement threshold, is the position of the sample with the highest amplitude.

The signal arrives from the ADC as a set of 32 8-bit samples every clock cycle with a period of 6.4 ns. All 32 samples are compared against the width measurement threshold. If a sample value is equal or higher than this threshold, a binary 1 is set in a 32-bit *vector* on the position corresponding to the position of the sample in the incoming ADC data set. The resulting vector might also include some noise at the

edges of the pulse, depending on the height of the width measurement threshold. The old routine simply counts the binary ones in this vector to get the pulse width. This works well for measuring the FWHM because the threshold was high. However, for width measurements at 25 %, 12.5 % or 6.25 % of the pulse height this might already become a problem, because the noise might be counted in as well. This is why the new routine cleans the outliers in this vector before counting the remaining ones in the clean vector.

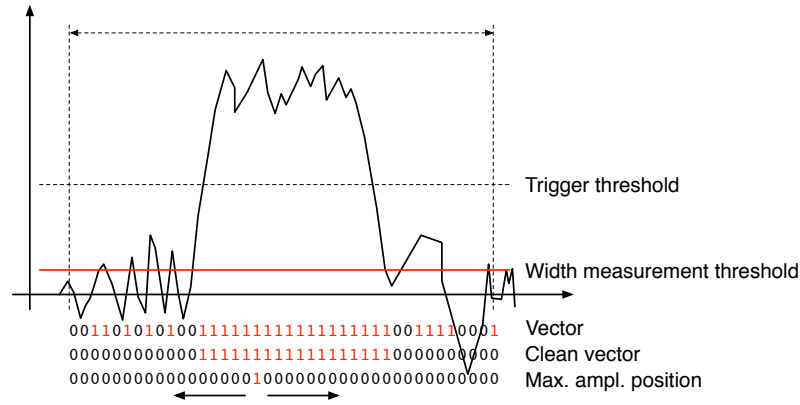


Figure 1.2: A sample pulse. The first vector shows which samples are above the width measurement height. The second vector is a clean vector. The third line shows the position of the maximum amplitude. The vector cleaning algorithm starts from the maximum amplitude and continues in both ways along the vector.

The routine starts from the position of the maximum height. It follows the vector in both ways and finds the first falling edge (0 at this position and 1 at the previous one). From there on it rewrites any binary 1 with a binary 0. The resulting clean vector only has one bunched set of binary ones which are summed, yielding a precise pulse width. The area measurement is similar - it only integrates over the samples marked in the clean vector. Both measurement routines, for area and for width, are implemented separately so that the area routine can have a different threshold set.

This section explains how the algorithm is designed. First, the idea for it was tested using Excel and was only afterwards ported to the VHDL. The underlying algorithm first cleans the vector. Then it passes the cleaned vector either to the width or area measurement (see figures 1.3a, 1.3b, 1.4a and 1.4b). The width measurement module only sums the ones in the vector whereas the area measurement module sums the data samples marked by the cleaned vector. Both modules issue a *valid* signal when they finish the measurement.

Vector cleaning

This is the most important block. Its inputs are: *vector*, *parsing active*, *position of the max. amplitude (PA)* and *its delay (DA)*. PA is a 32-bit binary number that shows the position of the sample with the maximum amplitude within the data block whereas

1.6. DESCRIPTION OF THE FIRMWARE

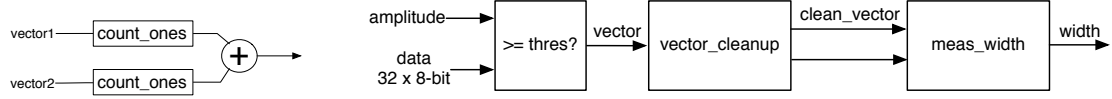


Figure 1.3: This block counts the remaining binary ones in the clean vectors and outputs this value as the pulse width.

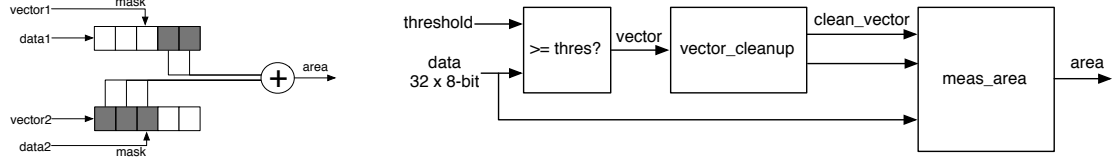


Figure 1.4: This block masks the input data with the clean vector and sums the remaining samples.

the DA tells us how many clock cycles after the start of the parsing this PA block is. The vector cleaning module is designed as a final state machine (FSM) with the

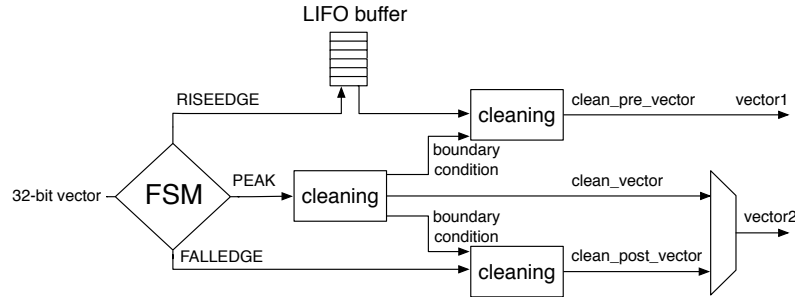


Figure 1.5: Vector cleaning routine outputs two vectors - one forward in time and one back in time from the peak of the pulse.

states IDLE, RISEEDGE, PEAK, FALLEDGE and READY. The FSM is idle until it receives the *active* signal from the external module, marking that the vector parsing has commenced. It switches to RISEEDGE, which starts two procedures: 1) it fills the vector of the pulse's rising edge into a last-in-first-out (LIFO) buffer and 2) counts down from the DA value. When this counter reaches 0, the FSM changes its state to PEAK because the current vector on the input is the one containing the maximum amplitude. This data block is sent through the *peak algorithm*, which cleans the vector. The FSM switches to FALLEDGE state. Now both the previously buffered vector of the rising edge and current vector of the falling edge go through the *pre- and post- algorithm* where they are cleaned, but they get their boundary conditions from the *peak algorithm*. The output of this module is therefore two cleaned vectors in parallel – one forward in time and the other backwards.

Algorithm

The underlying algorithm is sequential - it carries out a logic operation on vector bit 0, uses the output of this operation for the operation on bit 1 and so on. This means that it has to carry out 32 logic operations per clock cycle. With each operation taking approximately 0.3 ns, the whole logic chain takes approximately 10 ns to complete. With only 6.4 ns per clock cycle, this means timing errors would occur. To fix the problem, a more complicated *decimated algorithm* was invented. It consists of two parallel logic chains which only take every second bit into account and are at the end merged together. This effectively reduces the number of sequential logic operations to around 18, which is within the timing constraints.

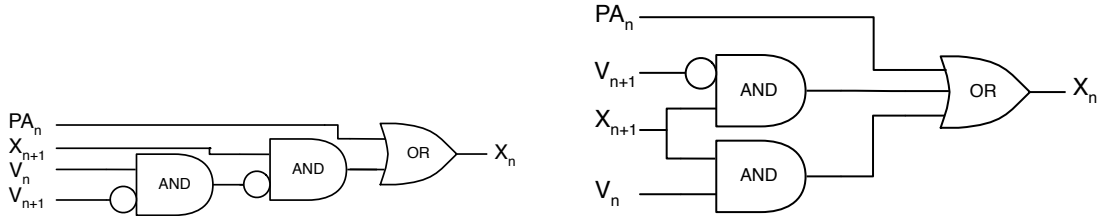


Figure 1.6: One logic step in the algorithm chain before and after Karnaugh minimisation.

1.7 Control and data interface

Communication between the Picoscope and the controller PC is done via the Picotech API functions. These are described in detail in [1]. In addition to the functions provided to the Picoscope users, the API used by CIVIDEC has access to several extra functions. These allow the user to download a customised bitfile to the FPGA, access the I/O registers and use the USB data transfer.

1.7.1 Software

The software has been designed in C++ in several levels of abstraction. Figure 1.7 shows the structure of the classes. The classes Picoscope, PSA and RawSignalHandler are there to make it easier to read and understand the controller code. In principle the PSfunctions can also be accessed directly by the controller, but for this the instruction sequences must be well known and understood.

1.7.2 Data readout

The device records the data in two forms - as signal waveforms and as histograms of analysed pulse parameters. Both are available upon request from the controller. Only one of the two can be transferred via the USB line at a time.

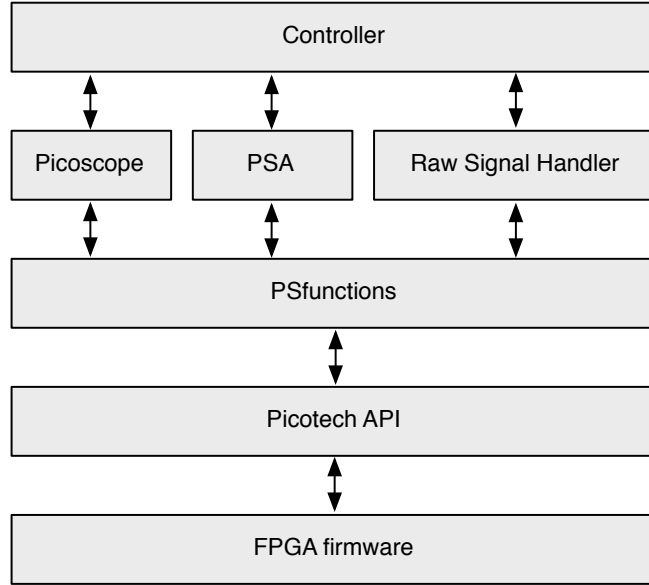


Figure 1.7: Abstraction levels of the controller software

The waveforms are saved into a FIFO buffer, which can hold up to 64 pulses of the length of ~ 500 samples. The data format for each pulse is such that it starts with a header containing the pulse timestamp and the sequential number, continues with the data samples and ends with a header containing all the measured parameters (width, amplitude, area, falling slope and form factor). When the FIFO is full, it issues a flag, which tells the controller that the data buffer is ready for readout.

The histograms are implemented into the FPGA's Block RAM. Their size ranges from 256 to 4096 bins (an 8-bit or a 12-bit histogram, respectively), depending on the required histogram resolution. For instance, the width parameter is measured with a 0.2 ns resolution; the expected maximum pulse width is less than 20 ns. This yields the maximum range of 100 bins, making an 8-bit histogram sufficiently large. The same reasoning is applied to the amplitude measurement. In this case the maximum range is defined by the 8-bit resolution of the ADC. The area measurement, however, yields higher values and can therefore have a more refined binning (12-bit). Finally, a single 12-bit 2D histogram is included, with 6 bits for every axis. It is used as an online scatter plot for comparing one measured parameter to another. An example for it is a comparison of the width against the area, which can help the user determine the cuts that need to be applied to the measurement.

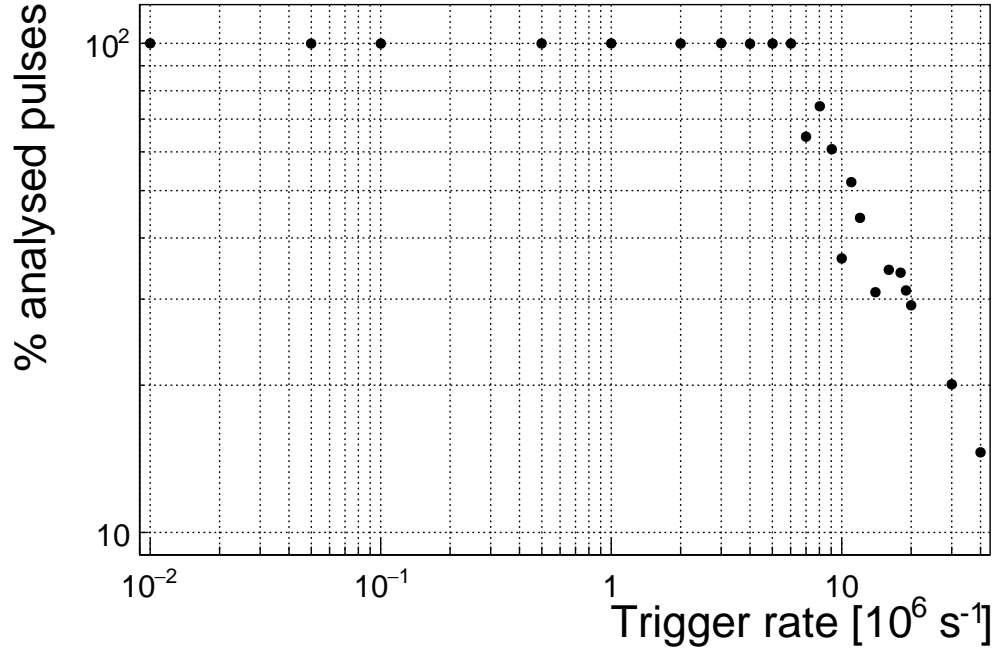


Figure 1.8: This figure shows the capability of the device to analyse all arriving pulses for a range of input frequencies. The highest achievable rate with zero lost pulses is $5 \times 10^6 \text{ s}^{-1}$.

1.8 Performance results

The device has been tested in the lab using a pulse generator as well as several radioactive sources. The results show that: 1) the amplitude, area and width measurement are linear across all input ranges, 2) the highest rate of the PSA algorithm is $\sim 5 \times 10^6$ pulses per second and 3) the lowest SNR where the algorithm still functions is ~ 5 .

Trigger rate A pulse generator was used to verify the highest achievable rate at which the PSA still analyses every incoming pulse. The final state machine implemented in the pulse analysis module prevents the triggering block from issuing a trigger due to an incoming pulse if the previous analysis is still in ongoing. Given that all the pulses were of the same length, the analysis duration was always the same. When the time between the incoming pulses was shorter than the time of the analysis, the pulses were not analysed. Figure 1.8 shows the sharp decline in the percentage of the analysed pulses when reaching the rate of 5 MHz. Therefore the overall analysis duration for a 10 ns pulse is approximately 200 ns.

Linearity A pulse generator was used to verify the linearity of the measurements across all input ranges. Pulse width and amplitude were varied and measured both with the oscilloscope and the PSA to check for non-linearities or inconsistencies in the PSA measurements. The resulting plots in figures ?? and 1.9c show that the PSA measurements agree well with those from the oscilloscope. The major inconsistency

is observed in the lower range of the plots. It stems from the fact that the bandwidth limit of the PSA is lower than that of the oscilloscope, which affects the pulse shape. Effectively, the PSA cannot measure the rectangular pulses of the width smaller than 2 ns.

Stability The input pulse signal was superimposed with white noise generated by a noise generator with a variable gain. The mixed signal yielded pulses with an SNR ranging from 5 (very noisy) to 100 (noise negligible). The PSA then performed the pulse parametrisation at different SNRs. The resulting plots in figures 1.9b, 1.9d and 1.9e show that the pulse width measurement is stable even for low SNR whereas the amplitude measurement is affected significantly. This stems from the analysis taking the highest sample as the pulse's amplitude. The area measurement, being effectively the integrated amplitude across the pulse, is also affected by the faulty amplitude measurement. Nevertheless, the mean area remained the same. This means that the added noise only affects the resolution of the spectrum, not its position.

1.8.1 Comparison between the charge- and current-sensitive spectroscopy

The calibration was done using a $^{148}\text{Gd}^{239}\text{Pu}^{241}\text{Am}^{244}\text{Cm}$ source which emits α particles with four different energies. The PSA in combination with the current amplifier was compared against the 8-bit spectroscopic application in combination with the charge amplifier and a commercial 14-bit spectroscopic readout.

The ^{241}Am peak measured by the Cx amplifier has an RMS of 0.8 ADC, which corresponds to a 32 keV energy resolution. For comparison, the C2 amplifier measures this peak with an RMS of 1.9 ADC, which corresponds to a 75 keV energy resolution. Therefore the energy spectrum measured by the current amplifier has a lower energy resolution.

1.9 Source calibration

The operation of the pulse shape analysis was tested using several radioactive sources. In particular, an α , a β , a γ and a n source were used. Each source was placed on top of the diamond detector and left for a predefined time depending on its activity. Table 1.2 shows the sources used, the time of exposure and their rate during this time. The data for the α source were taken for both polarities. In addition, a long run with an α source with a sheet of paper in between the source and the diamond was taken. The paper stopped the α particles but let through the photons, which helped to estimate the background photon radiation of the source.

1.9. SOURCE CALIBRATION

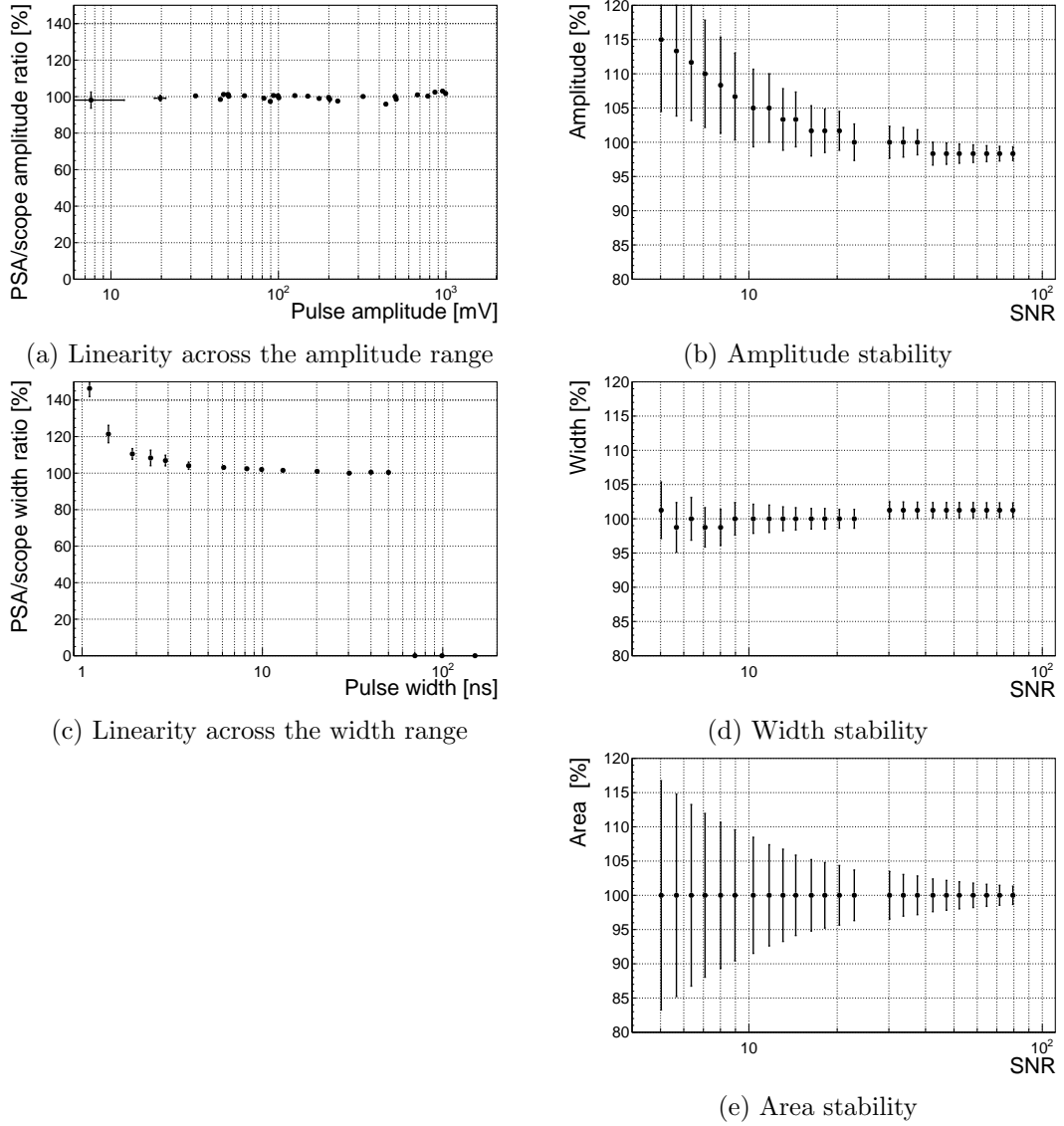


Figure 1.9: These diagrams show the linearity of the measurements and their stability with respect to analog noise.

Run	Source	Radiation	Energy [MeV]	Time [h]	Triggers	Rate [s ⁻¹]	Bias [V]
1	²⁴¹ Am*	α	5.5	60	958	4.4e-3	500
2	²⁴¹ Am	α	5.5	17	10558	0.17	500
3	²⁴¹ Am	α	5.5	18	11454	0.18	-500
4	⁹⁰ Sr	β	2.3	0.42	1.07e6	1000	500
5	⁶⁰ Co	γ	1.3	0.28	1.34e6	3300	500
6	²³⁹ Pu Be	n	1–10	2.5	1.5e6	230	500

Table 1.2: Measurements carried out at Atominstitut

The pulses acquired during the data taking are shown in persistence plots in figures 1.11. Figure 1.11a showing the ²⁴¹Am source background reveals that the diamond detector had been contaminated. This stems from the fact that α pulses

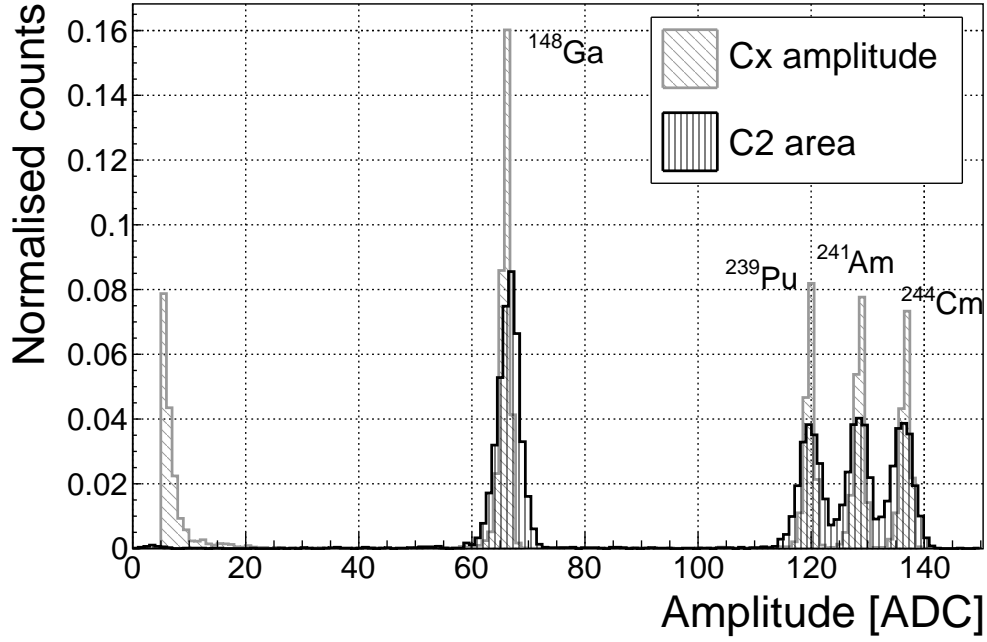


Figure 1.10: Spectrum of a $^{148}\text{Gd}^{239}\text{Pu}^{241}\text{Am}^{244}\text{Cm}$ source using a Cx and a C2 amplifier

are recorded despite having a sheet of paper, which stops all the particles emitted by the source. However, the number of α hits due to contamination is negligible - an estimated 1 h^{-1} . Another point worth noting is the falling slope of the rectangular pulse in figure 1.11c. This stems from the space charge that had built up during the neutron irradiation and is discussed in section 1.9.3. Finally, figure 1.11f shows that the neutron source causes the widest variety of pulse shapes - triangular and rectangular as well as those in between. Pulse shapes caused by neutrons are described in detail in [1].

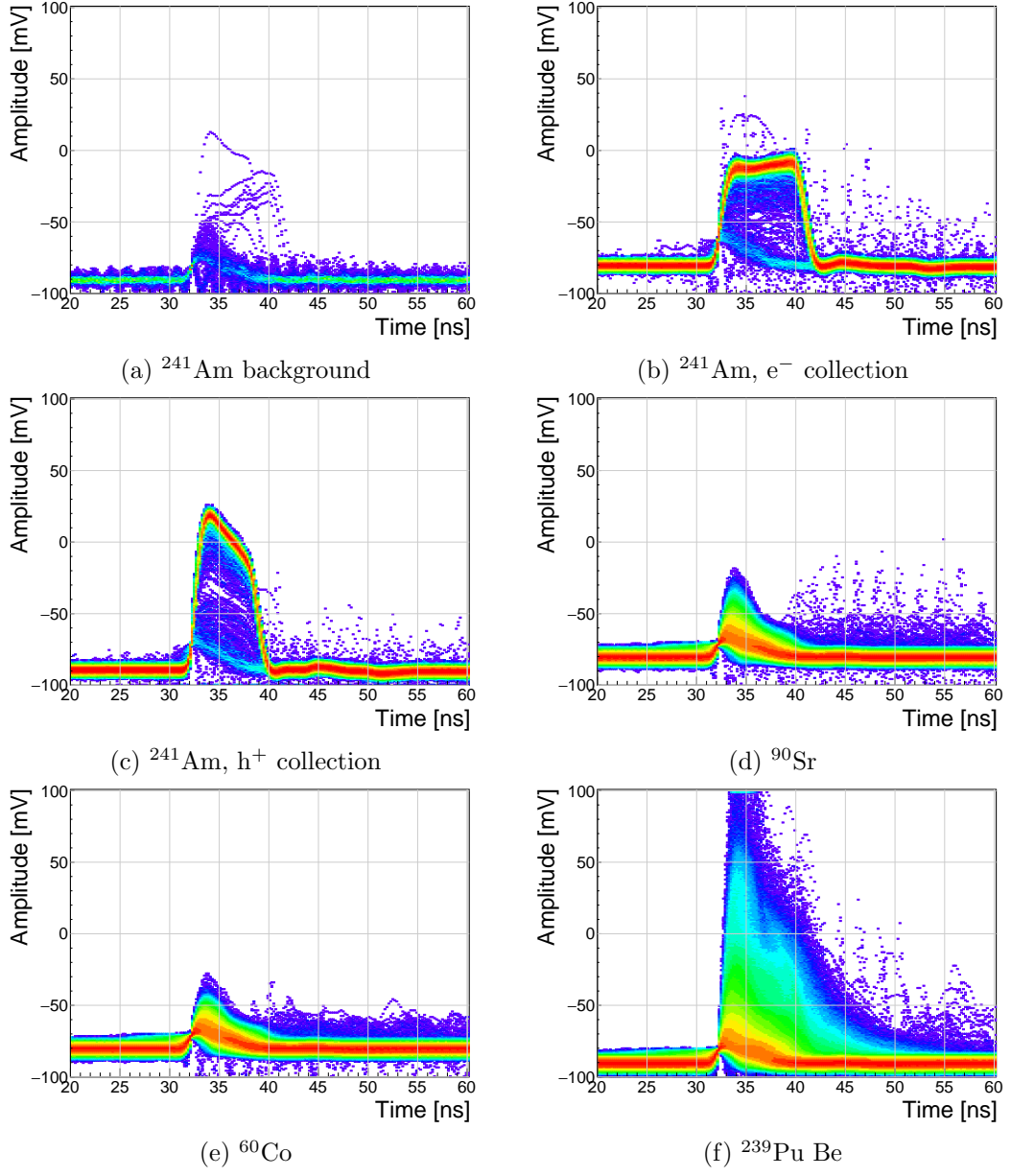


Figure 1.11: Accumulated pulses for all runs

1.9.1 Source measurements - scatter plots

An online pulse shape analysis was run on all the above mentioned data sets. The parameters of the pulses were plotted in 2D histograms - in form of scatter plots. The aim was to find a way to distinguish between the various types of radiation in order to only select the spectrum of a single type of particles from a spectrum of a mixed source. The energy spectrum is directly proportional to the measured area of the current pulses, therefore all the parameters were plotted against the pulse area. The parameters used were:

- FWHM
- Amplitude
- Amplitude \times Base width
- Base width - FWHM
- Form factor
- Slope

The sets of plots in figures 1.12, 1.13, 1.14, 1.15, 1.16 and 1.17 show the above listed parameters plotted against the pulse area for ^{241}Am background, electrons and holes, ^{90}Sr , ^{64}Co and ^{239}Pu -Be source, respectively. Any distinguishable difference between the plots of two sources would suggest that that particular parameter can be used to distinguish one type radiation from the other. In these figures the photons are considered the rejected pulses (greyscale colour palette) whereas α particles or neutrons are accepted (yellow colour palette).

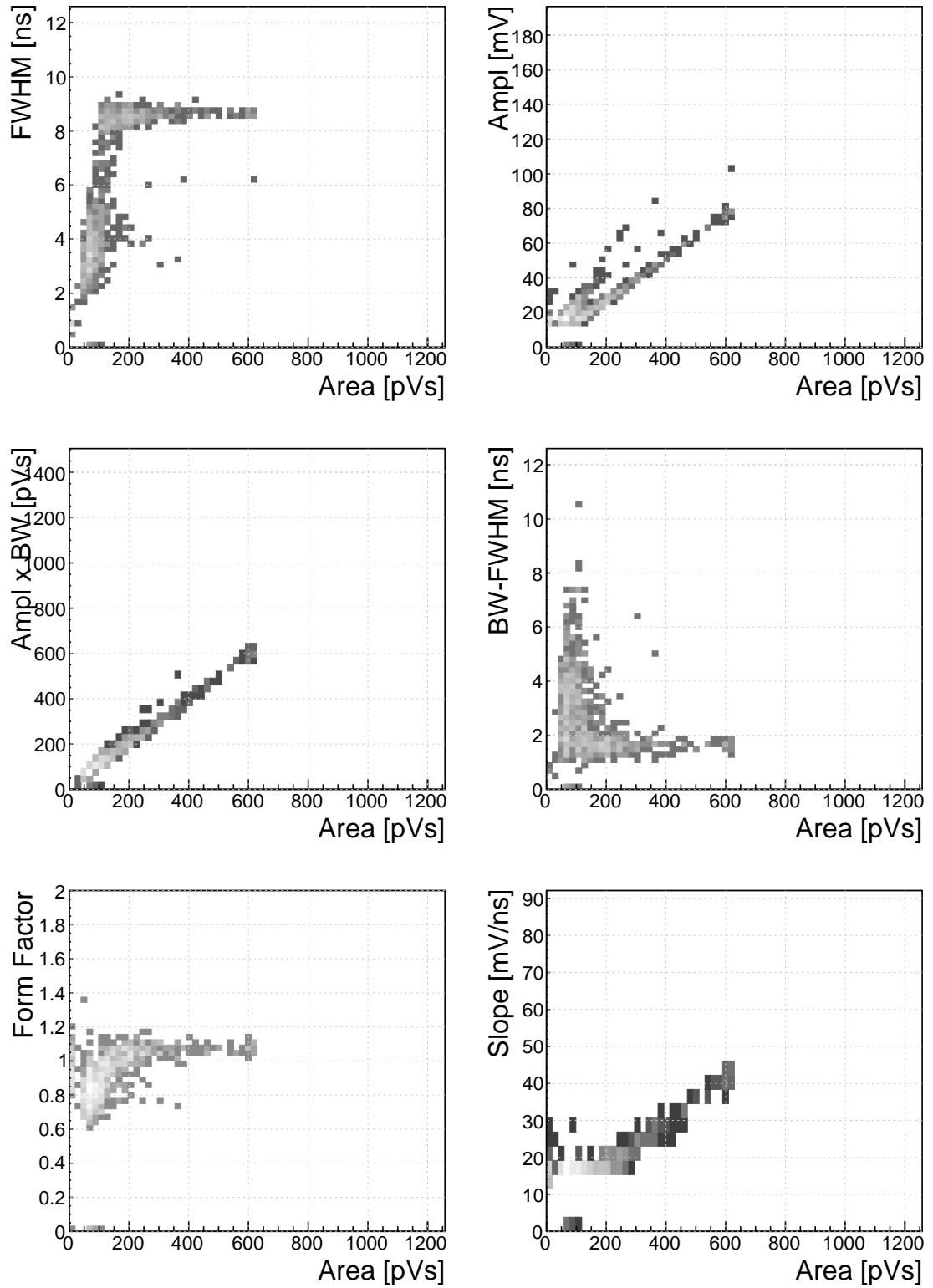


Figure 1.12: Background measurements

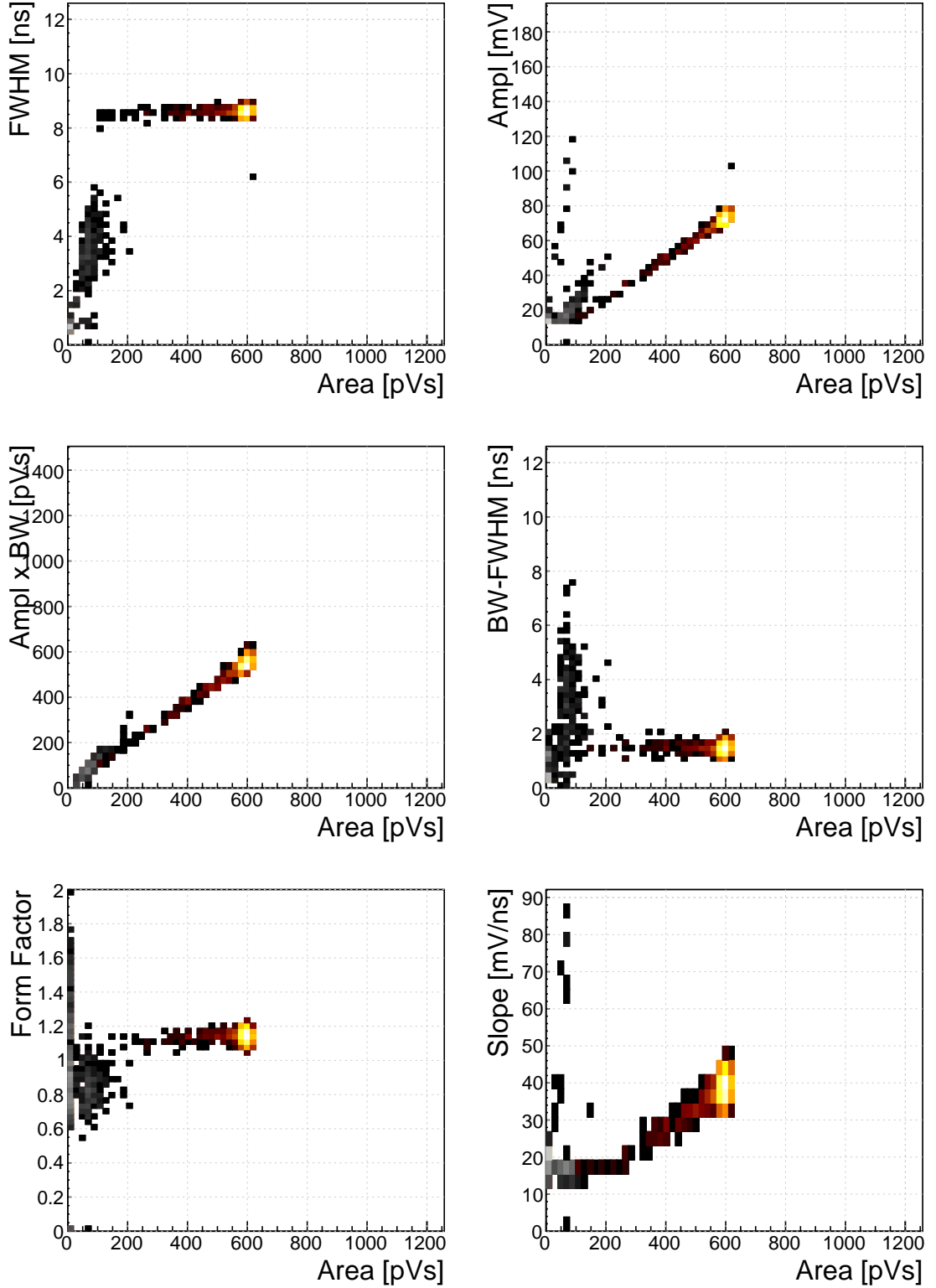


Figure 1.13: ^{241}Am , e^- collection. Qualifier: FWHM 7–10 ns.

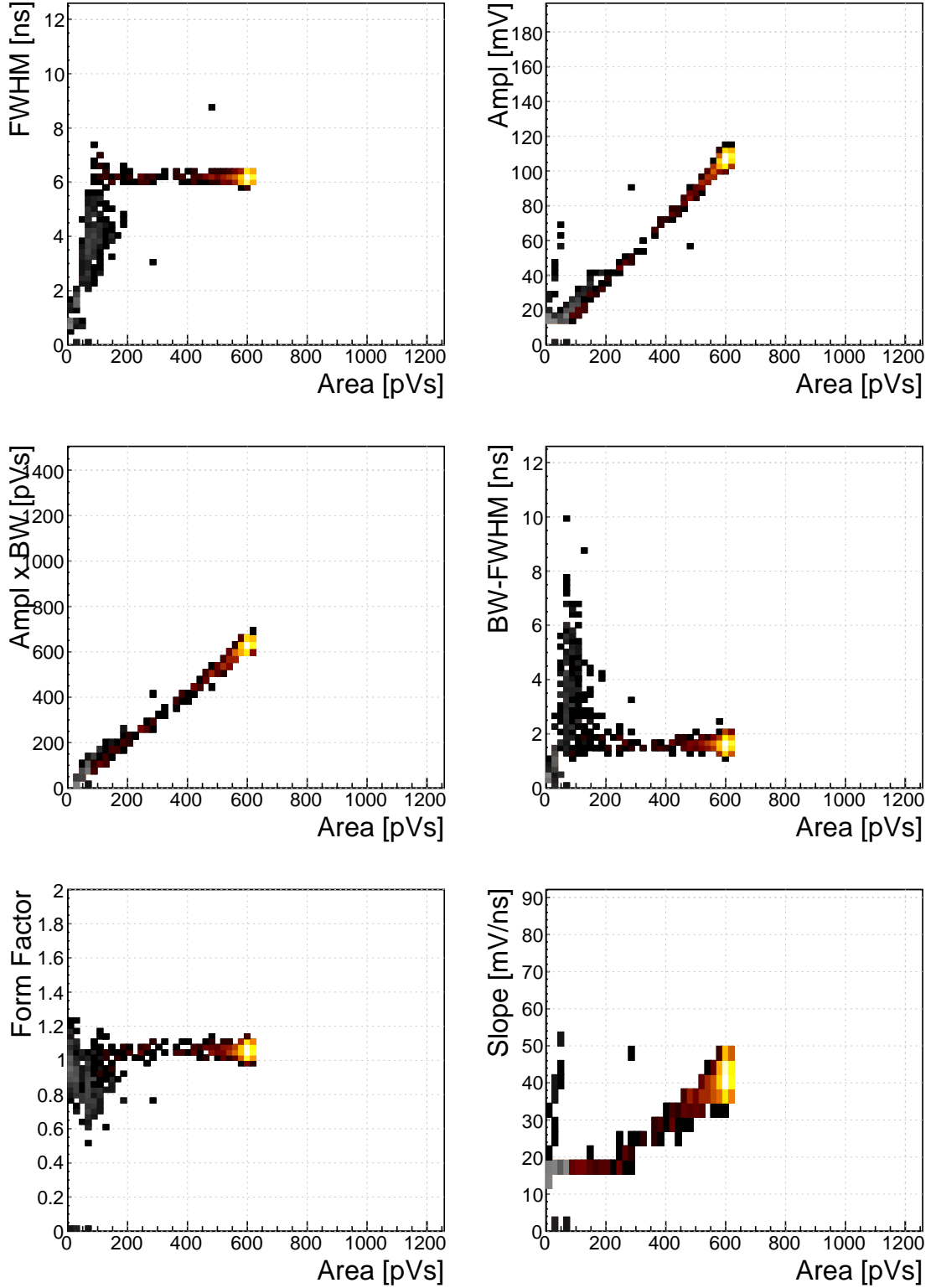


Figure 1.14: ^{241}Am , h^+ collection. Qualifier: FWHM 5.5–8 ns.

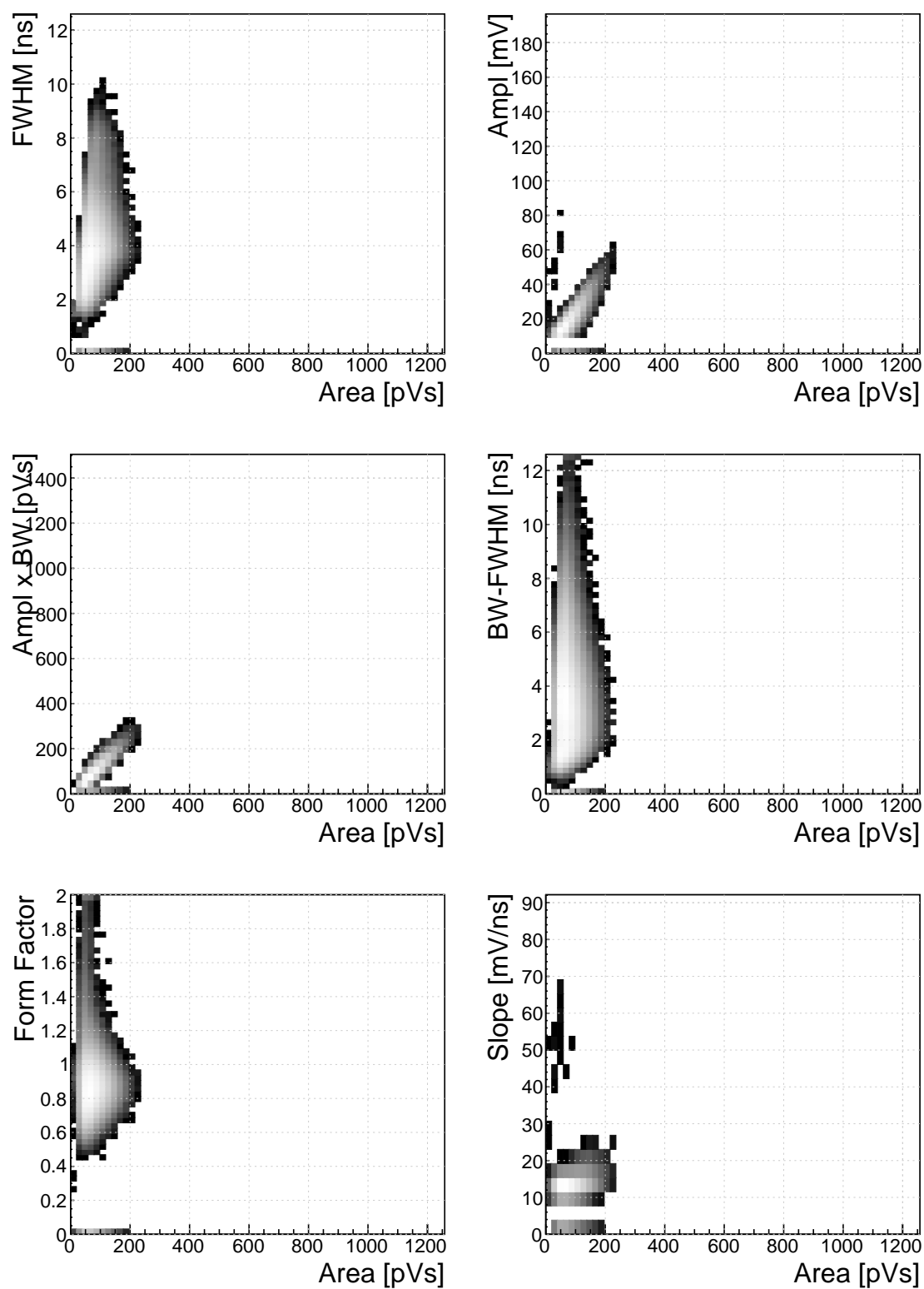


Figure 1.15: ^{90}Sr

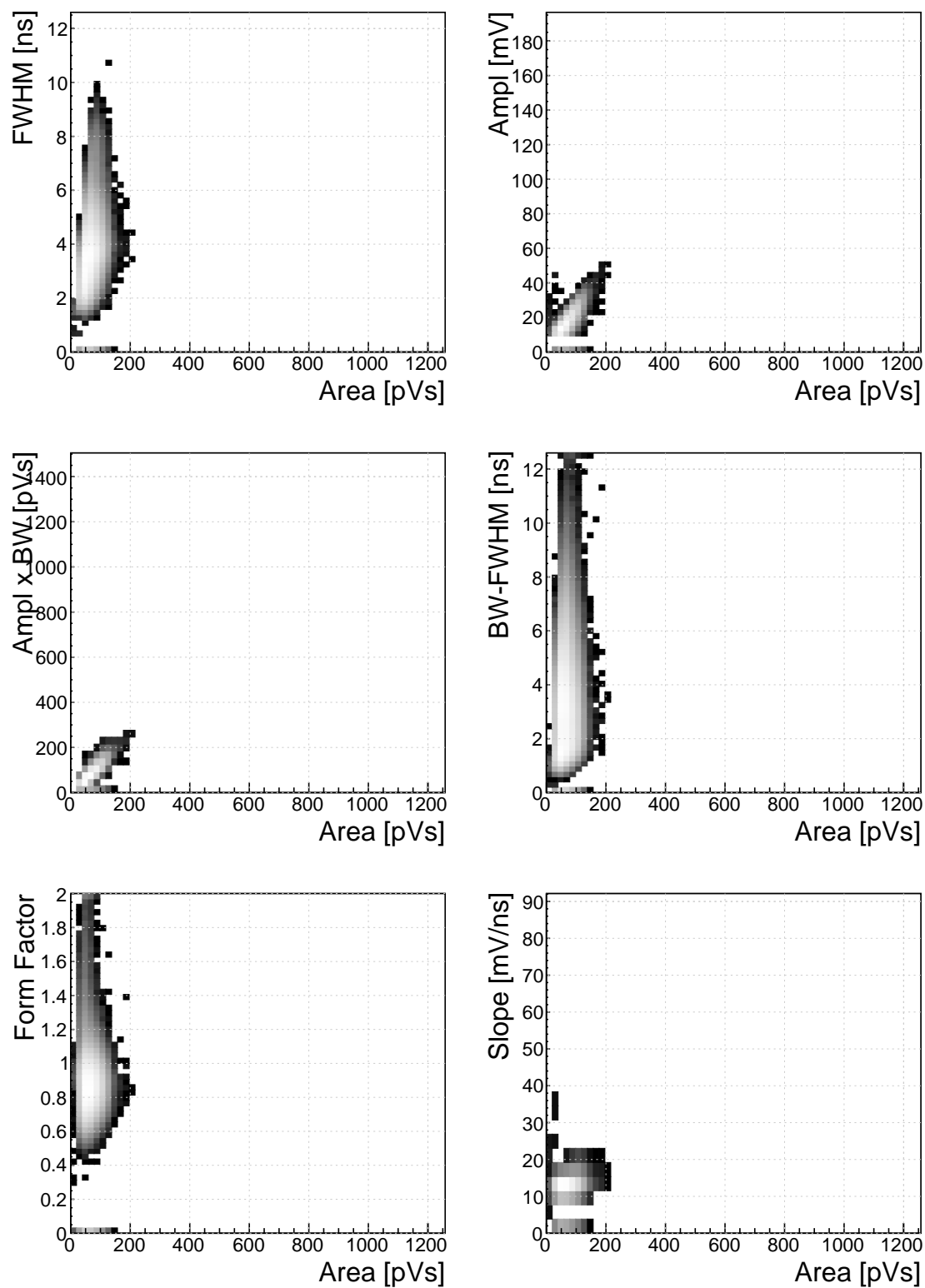


Figure 1.16: ^{60}Co

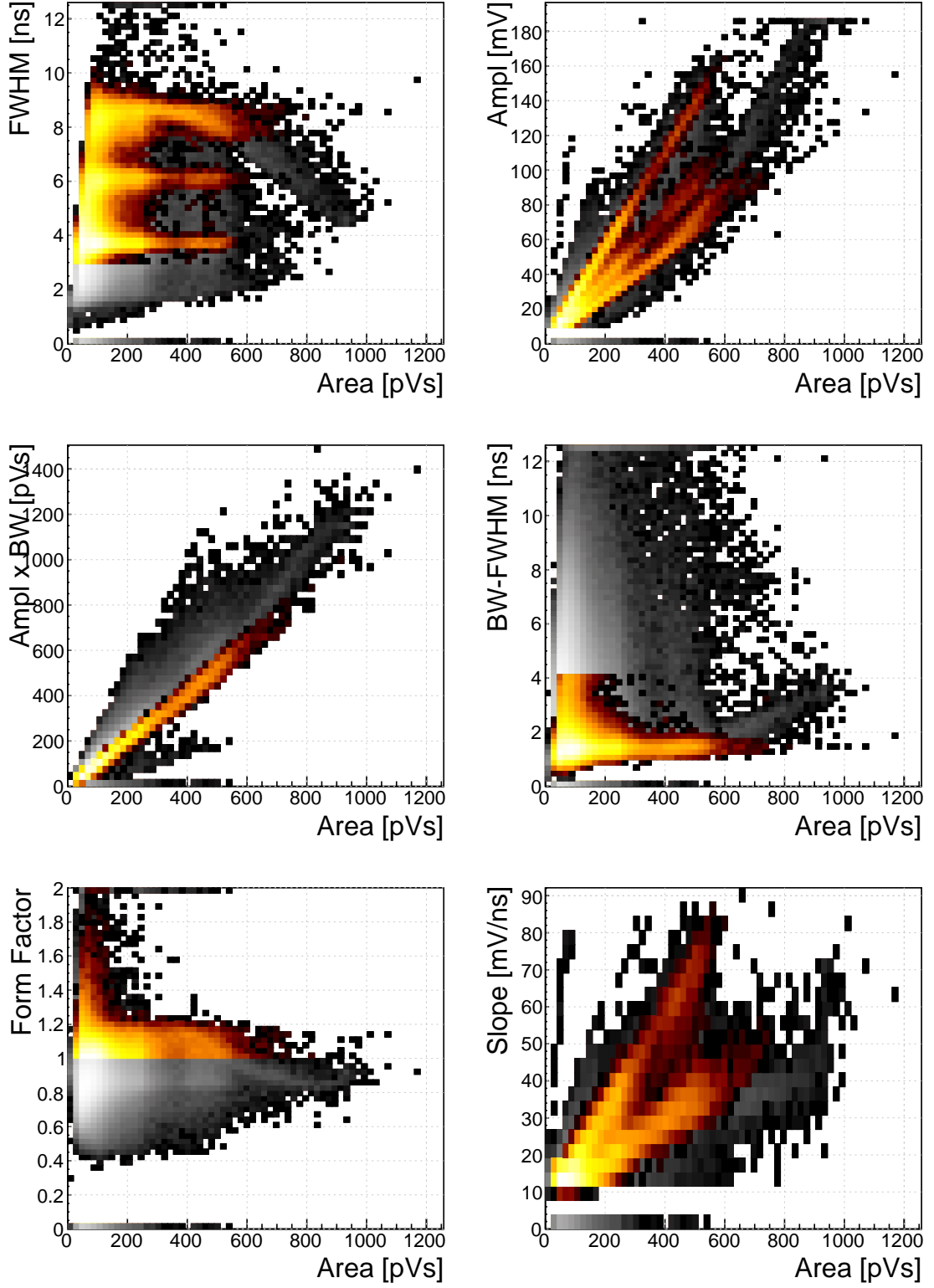


Figure 1.17: ^{239}Pu Be. Qualifiers: BW-FWHM 0.2–4 ns, FWHM 3–12 ns, Form factor 1.45

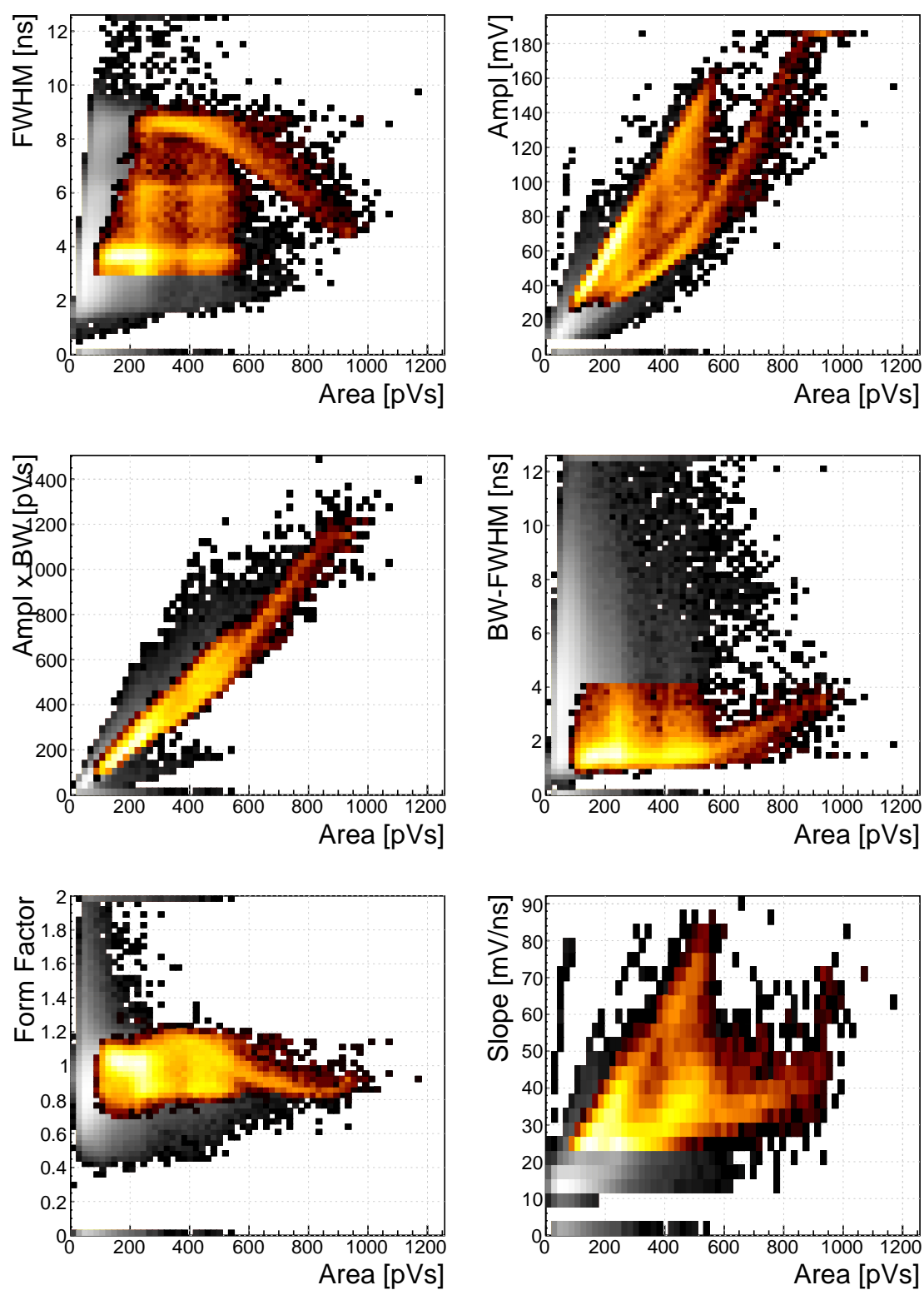


Figure 1.18: ^{239}Pu Be. Qualifiers: BW-FWHM 0.2–4 ns, FWHM 3–12 ns, Slope 25–104 mV/ns

1.9.2 Discussion

The ^{241}Am background measurement (see figure 1.12) revealed that the diamond detector has been contaminated, probably with chipped-off grains of the source. These emit α particles, which are detected by the diamond detector. The resulting plots are almost identical to those taken during the ^{241}Am run (figure 1.13), with a difference that there are much fewer α entries in the former. Due to the losses in the air they vary in energy – from ~ 5 MeV down to 1 MeV. Therefore in the top left plot a distinctive horizontal stripe appears at a width of 9 ns, ranging from 100 up to 630 pVs. The shape of the pulse at smaller energies retains the width, only its amplitude is decreased. The other cluster in the FWHM–area phase space comes from the background photons. The two clusters are far apart from one another with no overlap. It is therefore straightforward to define a cut in the FWHM to distinguish between the α and γ entries. The situation is similar when inverting the bias voltage and collecting holes (see figure 1.14), but here the two clusters are much closer together, making it more difficult to define a clear border between the two. The other five qualifiers are in this case less important than the FWHM. Nevertheless, it can be deduced from the plots that the difference BW-FWHM must be below 4 ns and that the Form Factor (bottom left plots) has to be set to 1.45 to push the α entries over 1, which is the threshold for accepting entries. The slope is dependent of the amplitude, which can be seen in the bottom right plot, making it an unreliable qualifier in the lower area range. The amplitude, scaling with area, makes a distinguishable straight line in the bottom right plot. For the electron collection this line has a different slope coefficient than the photon related line. For the holes, however, these two lines are overlapping, rendering it impossible to use this as a means of identifying the α from a γ entry.

The phase space of the ^{90}Sr source overlaps entirely with that of the ^{60}Co source (see figures 1.15 and 1.16). This renders it virtually impossible to distinguish between photons and electrons (MIPs). Comparing the FWHM phase space of the photons and alphas and the high reach of the former, the electron collection of the alphas would need to be used to distinguish between the two types of particles.

The neutron source emits a mixed field of neutrons and photons. The neutrons are rarely detected – the interactions happen mostly in the electrodes on either side of the detector. The α particles created by the interactions are detected by the diamond. Depending on the side of the interaction, the created pulse is either due to hole- or electron collection. They make the two distinct lines in the FWHM phase space (see figure 1.17, top left plot) at 9 ns and 6 ns. The third line at 4 ns is created when a neutron interacts in the ballistic centre \square of the diamond – the position where it takes the holes and the electrons the same time to drift to the opposite electrodes. The entries in between are created by neutron interactions at random positions in the diamond, which produce pulses of various shapes. By making use of the Form Factor set to 1.45, the three-line structure becomes more pronounced. However, a part of the cluster belonging to the 9 ns line is omitted. This part with the area ranging between 600 and 1000 pVs starts decreasing in FWHM due to an unexplained phenomenon, probably detector related. Consequently its Form Factor falls below the threshold

of 1, rejecting these entries as background. In an alternative case (see figure 1.18), a Slope qualifier is used instead. By cutting the pulses whose falling slope is not steep enough, there is a big chance that neutrons with a low energy deposition (small amplitudes) are cut away as well. Nevertheless, the entries with the large area, which definitely pertain to the neutron interactions, are accepted in this case. The three distinct lines in the FWHM phase space are still visible.

1.9.3 Space charge build-up

Space charge built up during the neutron irradiation ($\sim 10^{10}$ n), which reflected on the pulse shapes. Instead of a flat top, they developed a slope. This was of opposite signs for the two polarities, as can be seen in figures 1.19a and 1.19b. The figures contain a set of 64 superimposed pulses taken at ± 500 V bias. The shape persisted until the space charge was neutralised by means of a β source irradiation at a 0 V bias. Figure 1.20 shows the decreasing of the slope coefficient as a function of received β dose.

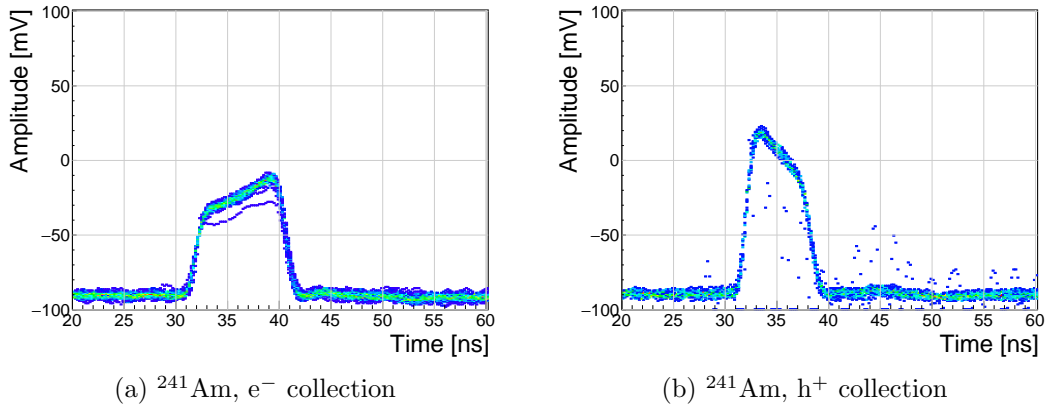


Figure 1.19: Built up space charge causes a slope which has the opposite slope for electrons and holes

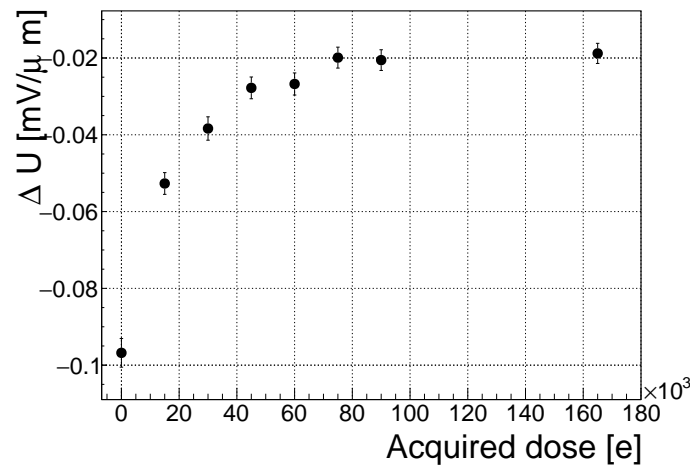


Figure 1.20: Using β radiation to reduce space charge in diamond

1.10 Neutron reactor measurements

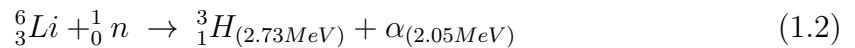
Semiconductor-based neutron detectors provide a compact technology for neutron detection. However, the cross section of a neutron with the diamond bulk is very low, since it only interacts with the core of the atom. Diamond is mainly used to detect charged particles and photons.

Research neutron reactors radiate a mix of particles, apart from neutrons also photons. The photons are considered a background radiation, concealing the neutron spectrum. When measured with diamond, the signal from neutrons is difficult to distinguish from the photon spectrum. In addition, low energy neutrons do not cause nuclear reactions in the bulk. All in all, the neutron measurements in a reactor present a challenge with diamond. However, by means of the PSA, the neutron signal can be discriminated from the photon background to some extent. The following two cases show how measurements of fast (n^+) and thermal (n^-) neutrons were carried out by making use of the PSA.

Note the changed scale on the X axis with respect to the figures in section 1.9.

1.10.1 Thermal neutron spectroscopy - TRIGA reactor

Thermal neutrons do not interact with the diamond bulk due to their low kinetic energy (of the order of 0.012eV). Hence a converter foil has to be added to produce second order effects. Incoming neutrons interact with the foil, producing a set of secondary particles. These can then be detected upon hitting the detector bulk. Common neutron interactions that are used in thermal neutron detection are $^{10}\text{B}(n,\alpha)^7\text{Li}$ reaction and $^6\text{Li}(n,\alpha)^3\text{H}$ reaction (α stands for ^4_2He , see equation 1.2). The focus in this section is on the latter. With a foil installed, there are several possibilities for neutrons to interact with the detector system. Each of these interactions ionises the diamond bulk in its own way, resulting in a specific shape of the current pulse. A neutron can: 1) interact with the foil, producing an α and a ^3H , 2) interact with a carbon atom, producing an α and a γ or even three α . Focusing on case (1), the equation for this reaction is the following:



The particles in the first case will be produced outside the diamond and will get stopped immediately upon hitting the sensor. The resulting pulses for both particles will have a rectangular shape of the same width, because the carriers will drift with the same speed in both cases. The difference is in the number of free carriers produced - triton will release more of them, which will in turn induce a higher pulse. TRIGA MARK II neutron reactor emits large amounts of γ radiation in the energy range up to 3 MeV. This already affects the measurements of α particles, the energy of which peaks at 2.05 MeV in the case of ^6Li converter foil. However, γ background radiation can be suppressed by discriminating current pulses of photons and α particles. This idea has already been implemented in offline analysis in []. The results show that the

background photons can be subtracted successfully. In order to improve the speed of the readout system and pulse analysis, the algorithm has been ported to FPGA where it discriminates particles in real time.

Measurements

ROSY readout device with the implemented Pulse Shape Analysis was put to a test at Atominstut in Vienna. Their TRIGA2 neutron reactor is capable of delivering thermal neutrons with the energy 0.012 eV at a rate of $10^3 \text{ n cm}^{-2} \text{ s}^{-1}$, with a considerable γ background.

First, the device was calibrated using an unsealed monochromatic α -source with the energy $E_\alpha = 5.12 \text{ MeV}$. Then the diamond detector was exposed to the beam. Secondary reaction products (α and ^3H particles), created by neutrons hitting the converter foil, were detected by the diamond sensor, together with a significant photon background. Then the pulse identification algorithm was applied to discriminate between the reaction products and the photons. The main parts of the detector were an scCVD diamond sensor sized $4.7 \times 4.7 \text{ cm}^2$ and a $1.8 \text{ }\mu\text{m}$ thick LiF converter foil, both embedded in an RF-tight PCB. The diamond sensor was biased with high voltage and capacitively coupled to CIVIDEC's C2 40 dB wide bandwidth signal preamplifier. A 5 m long BNC cable connected the preamplifier to CIVIDEC ROSY box. The detector assembly together with the preamplifier was fixed in front of an exit hole of the reactor.

Results

The data collected by the PSA show a high flux of photons, which covers a wide area range. The ^3He peak is clearly visible and has almost no overlap with the photon cluster. The α cluster has a much lower energy and is in the same energy range as the photons. However, if a FWHM parameter is observed, a distinction between the photons and the α can be seen. By setting a qualifier to the right value, the photon background is cut away, leaving only the thermal neutron decay products in the data set (see figure 1.21).

Note: this data set was taken with an older version of the firmware, which only measured a limited number of pulse parameters.

1.10.2 Fast neutron spectroscopy - CROCUS reactor

The CROCUS reactor at EPFL [] is a research neutron reactor. Its highest output power is 100 W. Currently there are fission chambers that carry out the neutron counting, which is a measure of the activity of the reactor. The new goal is to measure both neutrons and photons, but separately. The pulse shape analysis is a good solution for this task. Therefore a $400 \text{ }\mu\text{m}$ thick diamond detector with a specially designed casing was added to measure the activity. The LiF foil was added

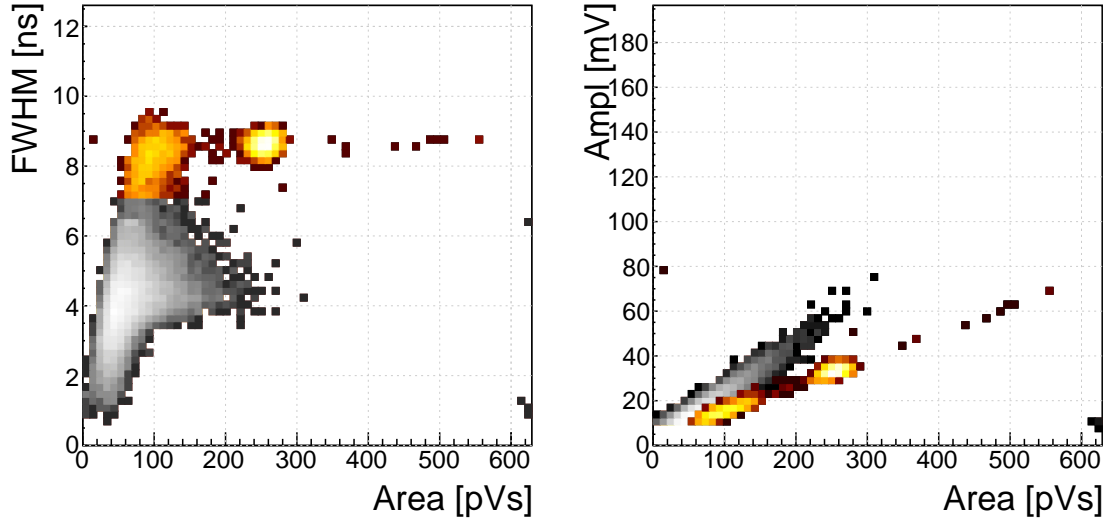


Figure 1.21: Thermal neutrons, photons. Qualifier: FWHM 7–12 ns

for conversion of thermal neutrons. The ROSY box with the integrated PSA routine was used for signal analysis.

The results from a test run at 10 W output power are shown in figure 1.22. The data include a mixed field consisting of fast neutrons, photons and of α and ^3He particles as products of thermal neutron decay. The data after the applied cuts again show a distinct ^3He cluster, as well as the three-lane neutron signature. The α signal is buried under the signals stemming from fast neutrons and photons.

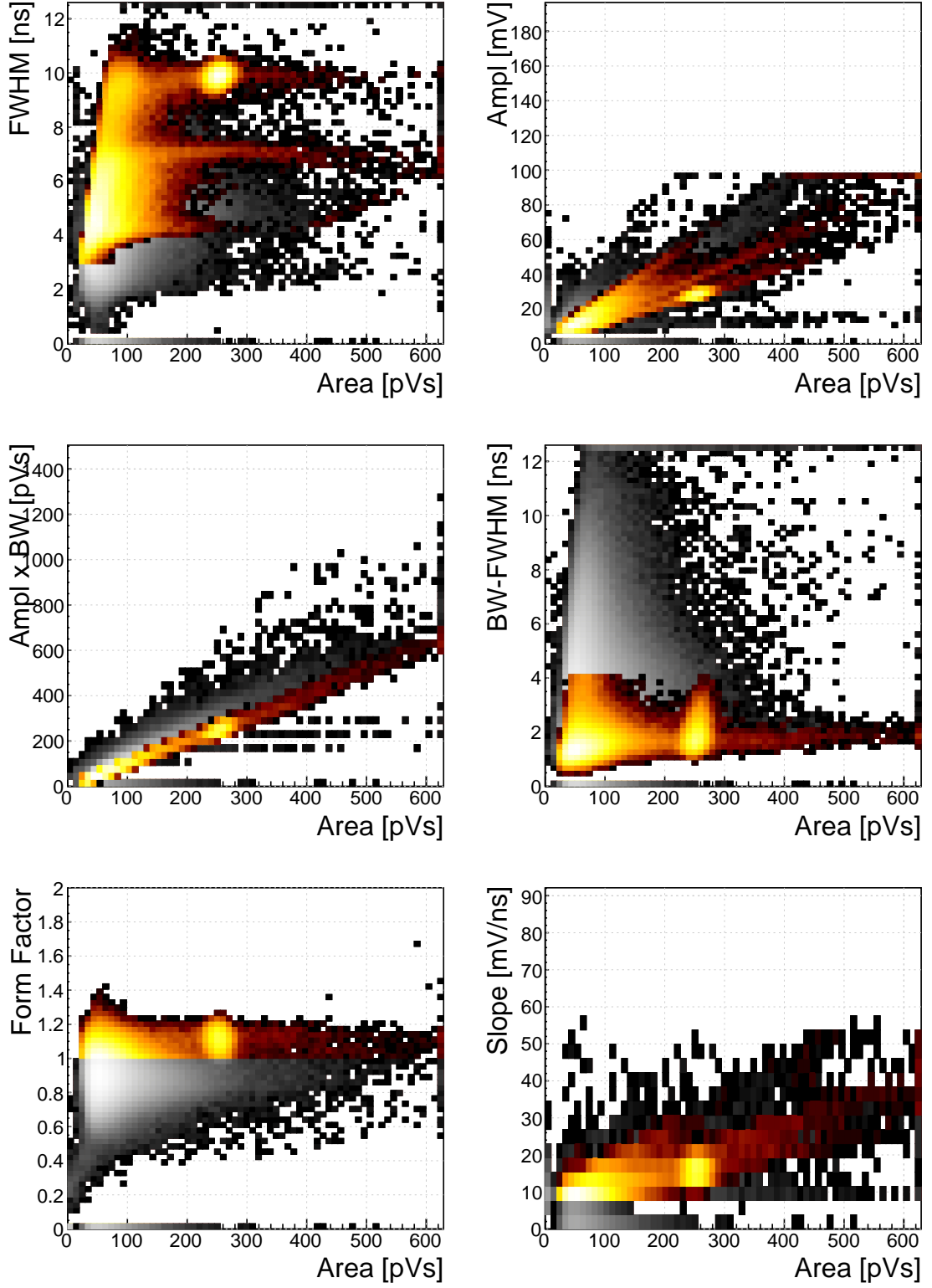


Figure 1.22: Fast neutrons, thermal neutrons, photons. Qualifiers: BW-FWHM 0–4 ns, FWHM 3–12 ns, Form factor 1.45, Slope 10–108 mV/ns

1.11 Conclusion

This chapter describes a system that can identify the type of radiation in real time. The system is implemented on an FPGA in a CIVIDEC ROSY box and is used with diamond detectors. The signal from the diamond sensor is read in and analysed in the firmware. First the shape of the pulse is parametrised. Then the logic determines the type of particle according to the user defined cuts. Finally the parameters are written into a histogram, which is read out by the user. The firmware is designed to carry out the pulse shape analysis of a single pulse in ~ 200 ns, yielding a maximum pulse rate of 5×10^6 particles per second. The rate as well as the linearity the measurement stability with respect to noise have been verified using a pulse generator. Then several radioactive sources were used to calibrate the device. Finally the system was installed in two neutron reactors to test the operation in a mixed field containing thermal neutrons, fast neutrons and photons. Various cuts were tested to optimise the identification.