Advanced lab course for bachelor students in physics

Experiment T7

Gaseous ionisation detectors and statistics

February 2021

Prerequisites

- Passage of charged particles through matter
- Gas-filled particle detectors
- Probability distributions and statistics

Goal of the experiment

- Operation of gaseous ionisation detectors
- Measurement of statistical distributions

Table of Contents

Bibliography

[1] Gerd Otter, Raimund Honecker: Atome – Moleküle – Kerne,
Band I: Atomphysik, Band II: Molekül- und Kernphysik, Bo 184

[2] William R. Leo: Techniques for Nuclear and Particle Physics Experiments, Dr 155

[3] Konrad Kleinknecht: Detektoren für Teilchenstrahlung, Dr 143

[4] Walter Blum, Luigi Rolandi: Particle Detection with Drift Chambers, Dr 181

[5] Claus Grupen: Teilchendetektoren, Dr 189

[6] Hanno Krieger: Strahlenphysik, Dosimetrie und Strahlenschutz, Du 130

[7] Siegmund Brandt: Datenanalyse, mit Beispiel- und Übungsbuch, Cu 202

[8] Strahlenschutzverordnung, `http://www.bfs.de`

[9] The Particle Detector Brief Book, `http://physics.web.cern.ch/Physics/ParticleDetector/`

[10] Detecting Particles: How to "see" without seeing. . . ,
`www.physics.ucdavis.edu/Classes/Physics252b/Lectures/252b_lecture6.ppt`

[11] Properties of argon-ethane/methane mixtures for use in proportional counters, Nuclear Instruments and Methods, vol. 188, issue 3, pp. 521-534, (10/1981)

1 Gaseous Ionisation Detectors

1.1 Principle of the Gaseous Ionization Detector

When the detection of ionizing radiation is required within an experiment, gaseous ionization detectors are often employed. The detectors operate under the principle of gas atom ionization and come in three flavors:

1. Ionisation chamber (not part of this experiment)
2. Proportional counter
3. Geiger-Müller counter

When an ionizing particle passes through the gas contained within the active volume of the detector, the energy transferred from the ionization particle to the gas particle causes an electron to be removed from the gas atom. This leaves an electron and an ionized gas ion. When an electric field, of sufficient strength, is present within the gas volume, the electron is accelerated away from the ion towards the anode. On its way to the anode, the electron can produce additional electrons by colliding with other gas molecules within the volume. When many charges reach the anode. This creates a negative voltage pulse that is forwarded to the readout electronics by a coupling capacitor.

1.2 Operating Regions of a Gaseous Ionization Detector

Figure 1 shows the total charge Q arriving at the anode versus the counter voltage U (logarithmic axes). Curves A and B show the behavior for two incident particles with different ionizing power (e.g. A: α particle, B: β particle). The pulse amplitude at the output of the counter is proportional to Q.

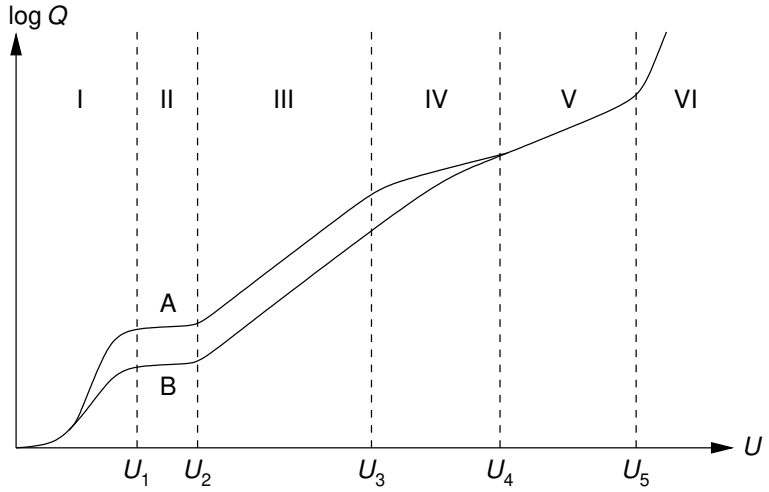

Figure 1: Total charge versus counter voltage.

Region I: The pulse amplitude increases with voltage. As the voltage increases, the probability that the created electrons and ions recombine decreases.

Region II: Above a voltage U_1, recombination no longer occurs. The measured charge stays constant for a certain voltage range and is equal to the sum of the primary ionization charges. The **ionization chamber** operates in this region.

Region III: Starting from a voltage U_2, the electrons are accelerated quite strongly towards the anode. They can ionize other gas atoms in collisions generating charge avalanches. The total charge remains proportional to the number of primary ionization charges. A counter which operates in this region is therefore called a **proportional counter**. The multiplication of the primary ionization charge caused by impact ionization is called gas amplification.

Region IV: Above U_3, the proportion of the total charge to the number of primary ionization charges is limited.

Region V: The Geiger region begins at U_4. The principle property of the Geiger region is that all ionizing particles cause the same voltage pulse, independent of their type and energy. A counter which operates in this region is called a **Geiger-Müller counter**. The gas amplification ($\approx 10^9$) is substantially larger for a Geiger-Müller counter than for a proportional counter, meaning even without additional electronic amplification, pulses with an amplitude of several volts can be generated.

Region VI: The pulses grow larger as voltage increases, until ultimately a continuous discharge occurs, this quite often damages the counter.

1.3 The Ionization Chamber (Example of a Simple Gaseous Ionization Detector)

The ionization chamber is one of the oldest radiation detection devices and is still often used due to its simple setup. A sample schematic is shown in figure 2.

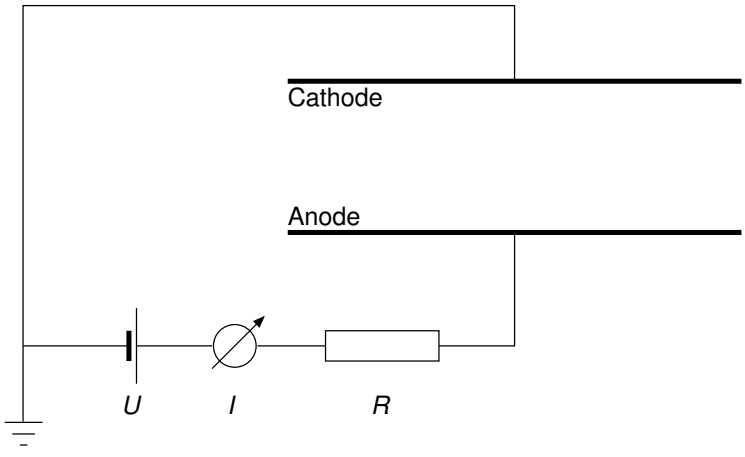

Figure 2: Schematic depiction of an ionization chamber.

The electron-ion pair creation energy in gases is roughly 30 eV, independent of the type and energy of the ionizing particle. If, for example, the incident radiation deposits 1 MeV of energy in the chamber, then roughly $3 \cdot 10^4$ electron-ion pairs are produced.

This corresponds to a charge of roughly $5 \cdot 10^{-15}\,\mathrm{C}$ for each type of charge. Depending on the construction, the charges created by the ionising particle are integrated to a constant current (current mode) or generate a voltage pulse for every individual particle (pulse mode).

For a typical capacitance of $C = 10^{-12}\,\mathrm{F}$ between both electrodes of the ionization chamber, a chamber operated in current mode with a load resistor of e.g. $R = 10^{12}\,\Omega$ has a time constant of 1 s, which is large compared to the time required for the collection of the ions ($\approx 1\,\mathrm{ms}$). Therefore, one does not detect the collection of individual particles, but measures the currents integrated over longer times. Such chambers are used for dose or dose rate measurements for radiation protection and for monitoring, as well as controlling the intensity at particle accelerators. The currents or charges to be measured are generally quite small, so a good electric insulation is required for such a chamber. Another problem is the recombination processes of the gas, which can interfere with the linear relation between absorbed energy and measured current.

For a chamber operating in pulse mode, the resistance is on the order of $R = 10^{6}\,\Omega$ (to be confirmed), so the pulse length for the same capacitance is on the order of $1\,\mu\mathrm{s}$.

1.4 The Proportional Counter

Principle and Operation

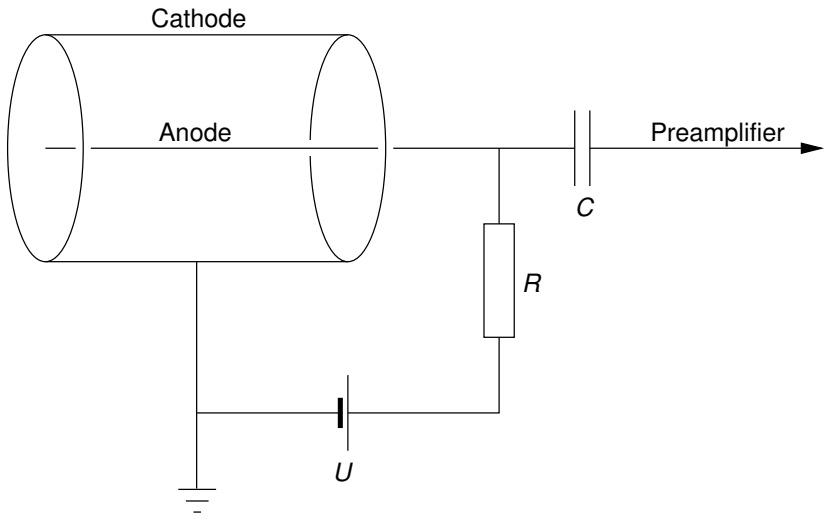

Figure 3: Schematic representation of a proportional counter.

A schematic of a typical proportional counter is shown in figure 3. The cylindrical tube (cathode) is grounded. A large positive voltage ($U \approx 1\,\mathrm{kV}$) is applied to a thin anode wire (20 to 100 μm in diameter) across a resistor ($R \approx 1\,\mathrm{M\Omega}$). Because of its cylindrical geometry, the proportional counter contains an electric field, which increases sharply immediately around the anode ($E \propto 1/r$), with typical field strengths of 10^4 to $10^5\,\mathrm{V/cm}$, such that gas amplification occurs near the anode wire. The amplitude of the measured pulse is proportional to the energy of the particle, so the proportional counter is an ideal detector for energy measurements. The gas amplification factor (10^4 to 10^6) increases exponentially with the applied voltage. In addition to impact ionisation, UV photons are

emitted by excited atoms. These photons cause photoelectrons to be released from the gas or the chamber walls, which can in turn cause new impact ionizations.

If the detector is filled with pure noble gases, the gain increases rapidly with voltage because of the large number of photoelectrons. The proportional region is consequently very small and poorly suited for practical applications. If polyatomic gases, e.g. methane, are used, then the proportional region becomes large. The UV photons are absorbed by the molecules, i.e. the molecules dissociate or vibrations are excited. Quanta emitted by the vibrating states are generally less energetic and can not be used to free photoelectrons. Mixtures of noble gases and polyatomic molecules are often used. Using a proportional counter, resolution times of roughly 10 ns can be achieved.

Gas-flow Proportional Counters

In gas-flow proportional counters, the counting gas is continuously exchanged via a regulating valve. These counters are particularly useful for measuring α, low-energy β particles, as well as soft x-rays because the sample can be put directly inside the counter volume. Counters with a gas-flow must be tightly sealed with gas-tight walls that short-range radiation cannot penetrate. A schematic of a methane-gas-flow proportional counter is shown in figure 4. At atmospheric pressure, an argon-methane mixture (10 % methane) flows through the counter volume, in which a thin wire loop serves as the anode and the walls as the cathode.

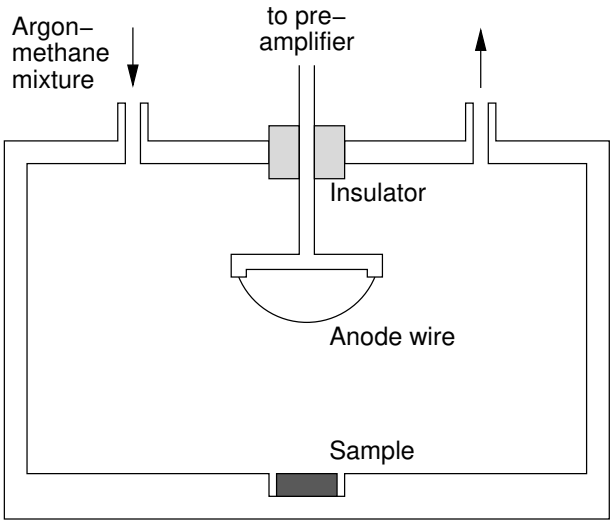

Figure 4: Schematic representation of a methane-gas-flow proportional counter.

Characteristic Curve of the Proportional Counter

The characteristic curve of a counter shows the counter's dependence on the pulse rate (number of pulses per unit of time) as a function the voltage U applied. This curve is shown in figure 5 for the proportional counter for both α and β particles present simultaneously.

For a specific voltage (threshold voltage U_T), the minimum pulse height, which depends on the input sensitivity of the subsequent amplifier, is reached. Because of the larger

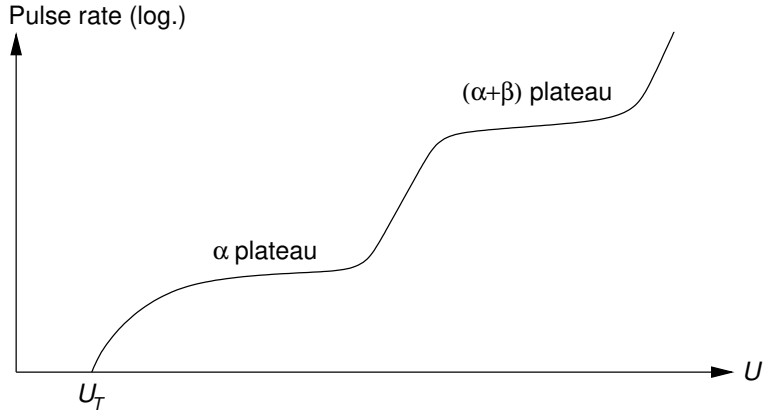

Figure 5: Characteristic curve of a proportional counter in the presence of both α and β particles.

number of produced electron-ion pairs, only α particles are counted at initially. As the voltage increases, β particles begin to be counted. Thus, the characteristic curve first reaches the α plateau and then the $(\alpha + \beta)$ plateau (figure 5). By choosing the voltage appropriately, the proportional counter can therefore distinguish primary ionizations of different energies, i.e. α and β radiation.

1.5 The Geiger-Müller Counter

The Counter Discharge in the Geiger Region

When the field strength in a proportional counter is increased, more and more UV photons are produced in the avalanche. This increases the probability that photons generate electrons through the photoelectric effect at other, more distant locations in the counter. These electrons induce new avalanches, which cause the discharge to propagate along the counter wire. At the end of the avalanche formation, the counter wire is surrounded along its entire length by a plasma. When these conditions are present, the counter operates in the Geiger region.

During the relatively short time (roughly 10 ns) in which the electrons are siphoned off, the ion plasma, which moves slowly, shields the electric field of the counter wire. The spatial charge of the ion plasma reduces the field strength. When this occurs, no new avalanches can be triggered. The positive ions traverse to the cathode, where they are neutralized at the counter tube. This can free secondary electrons from the surface, which restart the discharge process. It should therefore be ensured, that the discharge is „quenched“.

Non-self-quenching counters induce the quench-process by using very large resistors R in the counter circuit. When a particles passes through, the voltage drop across the resistor is so large, that the voltage across the counter falls below the operating voltage. For this to work, the time constant of the counter circuit must be sufficiently large, that the voltage drop persists for long enough for all of the ions to arrive at the counter tube.

In self-quenching counters, a so-called quench gas (e.g. methane or carbon dioxide) is mixed in with the counting gas. These gases have wide absorption bands to absorb UV photons resulting from the gas amplification process. Additionally, there will be impacts

between the positive counting gas ions and the quench gas molecules. The charge of the counting gas ions is transferred to the quench gas due to the different electron properties. Only quench gas ions arrive at the counter tube. Because of the lower ionization potential of the quench gas ions, they are unable to free secondary electrons at the tube, stopping the discharge process.

Characteristic Curve of the Geiger-Müller Counter

If one exposes a Geiger-Müller counter to a constant amount of radiation and records the pulse rate as a function of the applied voltage U, then the resulting curve would look like that shown in figure 6. This is the characteristic curve of the counter. At the threshold voltage U_T, the counter is not yet in the Geiger region. The pulse height depends on the energy of the particles. Only the largest pulses are counted. As the voltage is increased, the pulses grow larger and a larger number of pulses are counted. When the Geiger threshold U_G is reached, all pulses are equally large. At voltages above U_G, the pulse rate is essentially constant (plateau) under ideal conditions. In reality, however, it increases slightly with increasing voltages due to increasingly frequent multiple discharges. The slope of the plateau of good counters does not exceed 1 % pulse rate change per 100 V voltage change. If the voltage is increased further, a continuous discharge occurs, which damages the counter. To provide the proper working point, $U = U_G + 100\,\mathrm{V}$ should be used. This guarantees that the operating voltage is not one the leading edge in the proportional range nor in the discharge range, which could potentially damage the detector. The choice of operating voltage generally depends on the type of counter and the gas mixture. Depending on the gas mixture, it is possible that the plateau is very short or practically non-existent.

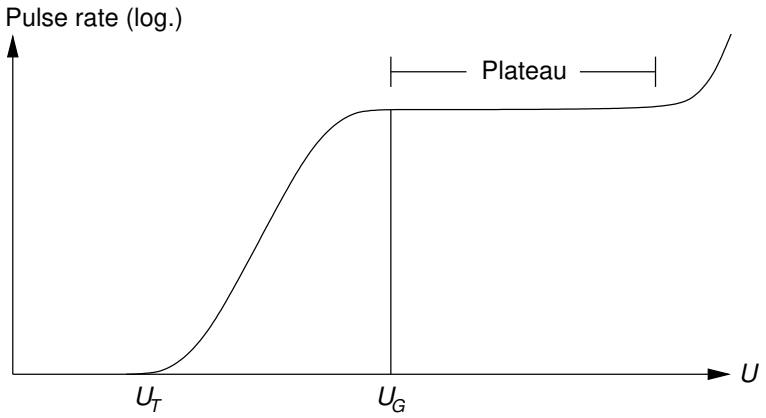

Figure 6: Characteristic curve of a Geiger-Müller counter.

Dead Time and Recovery Time of the Counter

After a particle has caused a discharge in a Geiger-Müller counter, it takes a certain amount of time for the positive ions to wander to the cathode due to their relatively low mobility. During this time, the space charge of the positive ions weakens the field near the counter wire. A particle which passes through the counter during this time either does not cause a pulse or generates only a small pulse height.

A distinction is made between the actual dead time T_d and the recovery time T_r. The dead time is the time required after a discharge for the counter to be able to detect the next particle. After this time, the ion tube has moved far away enough from the counter wire for gas amplification to be able to occur again. The recovery time has passed, when the ion cloud has reached the cathode. After this, the pulses reach their original height again.

The dead time and recovery time of a counter can be visualised and measured with an oscilloscope using a method specified by H. Guyford Stever: the output of the counter is handed over to an oscilloscope and the trigger threshold U_t is set, such that the fully formed pulses are shown at the start of the time axis. Until the end of the dead time, no more pulses appear on the screen. Within the recovery time, pulses occur whose height increases with time and finally reaches the original value. By superimposing multiple images (afterglow of the oscilloscope), one gets a series of pulses, with an envelope from which the dead time and recovery time can be read (see figure 7).

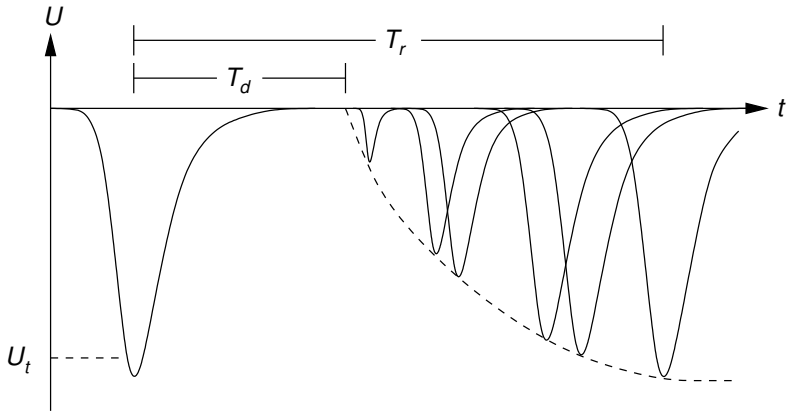

Figure 7: Stever diagram for determining the dead time and recovery time.

Crucial for the measurement is the time, after which the counter can once again deliver a pulse which is large enough to be registered by the pulse counter. This so-called resolution time τ depends on the amplification of the pulse counter and on the chosen threshold of the amplifier and lies between T_d and T_r.

Dead Time Correction

If a counter with resolution time τ measured a pulse rate of n particles per second, then the counter was insensitive for a fraction $n \cdot \tau$ of the time. If a total of N ionising particles per second passed through the counter, then $N \cdot n \cdot \tau$ of those particles were not detected. The measured pulse rate n is the difference between the actual pulse rate N and the number of particles per time $N \cdot n \cdot \tau$ which were not detected:

$$n = N - Nn\tau \quad \text{or} \quad N = \frac{n}{1 - n\tau}$$

In a so-called dead time stage, counter pulses are ignored during an adjustable time interval τ_i after an initial counter pulse passes through. An artificial dead time of known or measurable length τ_i is generated. If one measures the counter rate n_1 for a stage dead

time τ_1, which is considerably larger than the resolution time τ of the counter, then the true pulse rate N is:

$$N = \frac{n_1}{1 - n_1\tau_1} \qquad (\tau_1 \gg \tau)$$

If one repeats this measurement with another dead time τ_2, which is significantly smaller than the resolution time τ of the counter, then the measured counter rate n_2 is determined by the resolution time of the counter:

$$N = \frac{n_2}{1 - n_2\tau} \qquad (\tau_2 \ll \tau)$$

Exercise: Derive a formula for determining the resolution time of a counter by using a dead time stage.

2 PIN photodiode Geiger-Müller Counter

The semiconductor diodes that many of us are most familiar with are a two terminal device used in many electronic circuits to restrict the flow of current in one direction. In the most common type of diodes, called p-n diodes, a crystal semiconductor, typically silicon or germanium, have impurities added to both sides of the crystal. This process is called 'doping'. One side of the crystal is doped in such a way that on one side an excess of negative charge carriers is created, while on the other side an excess of positive charges is created. The negative excess region is often referred to as an 'n-type' semiconductor, while the positve excess region is referred to as 'p-type'. These two semiconductors are brought together which results in a brief transfer of charges between the n- and p-type semiconductors creating a small, third region called the 'depletion' region. One terminal of the diode is attached to the n-type semiconductor and the other terminal connects to the p-type. When an electric field is applied in the direction of p-type to n-type semiconductors, an electric current flows, but not in the reverse direction.

The PIN diode is similar to the p-n diode in that it has a p-type and an n-type semiconductor. However, there is an intermediate region, called the 'intrinsic' or 'i-type' region, sandwiched in between the p-type and n-type semiconductors. The primary difference in operation of the PIN diode is that p-type charge carriers move into the intrinsic region and fill holes within this region before moving to the n-type semiconductor region.

3 Probability Distributions and Statistics

3.1 Binomial, Poisson and Gaussian Distribution

The decay of a radioactive isotope is a typical example of a stochastic process. The atomic nuclei of a radioactive source constitute a very large statistical sample. The observed decay rate is determined by the decay probability of a single atomic nuclei. This probability is a natural constant and is virtually unaffected by external conditions. The decays of different nuclei are independent and are governed by the following probability distributions.

Binomial Distribution

If there are n radioactive atoms, each of which decays with a probability p in a fixed time interval, then $P(k)$ is the probability that exactly k atoms decay within this time interval. Because these n atoms are identical, they obey the following combinatoric relation:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

. Then the probability distribution $P(k)$ is given by the binomial distribution (see figure 8):

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad (k = 0, 1, 2, \ldots, n) \tag{1}$$

The mean of this probability distribution is given by:

$$\overline{k} = \sum_{k=0}^{n} k\, P(k) = np,$$

. The standard deviation is then found to be:

$$\sigma = \sqrt{\overline{k^2} - \overline{k}^2} = \sqrt{\sum_{k=0}^{n} k^2\, P(k) - \sum_{k=0}^{n} \left(k\, P(k)\right)^2} = \sqrt{np(1-p)}.$$

Poisson Distribution

If one starts with the binomial distribution, it can be considered what happens when n is very large, such that k is small compared to n. In this case this one is applying the approximation in the limit $n \to \infty, p \to 0$ with $np = \text{const} = \mu$. This yields the Poisson distribution (see figure 9) as an approximation of the binomial distribution (1):

$$P(k) = \frac{\mu^k e^{-\mu}}{k!} \qquad (k = 0, 1, 2, \ldots) \tag{2}$$

The mean of the Poisson distribution is given by:

$$\overline{k} = \sum_{k=0}^{\infty} k\, P(k) = np = \mu,$$

. The standard deviation is given by:

$$\sigma = \sqrt{np} = \sqrt{\mu}.$$

The Poisson distribution specifies the probability that k events occur, when the expectation value is μ.

Gaussian Distribution

If one considers the case where the Poisson μ becomes large (where $\mu \approx 10$ is sufficient), then the Poisson distribution (2) approximates as the Gaussian or normal distribution (see figure 10):

$$P(k) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right) \tag{3}$$

The Gaussian distribution is, in contrast to the previous distributions, continuous. That is, it is defined for all real numbers k. It is symmetrical around the mean μ with a width σ. Interestingly, in the case that the Gaussian distribution results from approximating the Poisson distribution, μ and σ are not independent. The same relation holds as for the Poisson distribution:

$$\sigma = \sqrt{\mu}. \tag{4}$$

3.2 Spread and Measurement Uncertainty

The fact that the standard deviation is set only by the mean for the Poisson and Gaussian distributions is extremely important. This means that the spread of the measured number of pulses N is known from just a single measurement. This is because, when N is not too small, $N = \mu \approx \overline{N}$ and $\sigma \approx \sqrt{N}$. A larger quantities of measurements N leads to a more precise result. This can be inferred from the relative error:

$$\frac{\Delta N}{\overline{N}} = \frac{\sigma}{\mu} \approx \frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}}$$

Assuming that a series of measurements are Gaussian distributed, the fraction of all data points that lie within a given interval $[\overline{N} - a, \overline{N} + a]$, is given by the integral of the probability distribution (3) over the interval:

$$\int_{\overline{N}-a}^{\overline{N}+a} P(N)\, dN.$$

The probability that a single measurement falls within the given interval is thus 68.3 % for $a = \sigma$, 95.4 % for $a = 2\sigma$ and 99.7 % for $a = 3\sigma$.

3.3 The χ^2 Fit

Often, one wants to understand how well the measured data matches the theoretical model being used to explain, or understand the data. In our case the model under consideration would be either the Poisson or Gaussian distribution. To compare the data with the distribution, one common method of testing the "goodness of fit" is the χ^2 test. The first step is to generate the χ^2 distribution that describes the comparison of the data to the model/distribution.

To do this, consider a set of n measured values x_i. Let these data points be normally distributed data with errors σ_i. These will be compared against a set of values $X_i = X_i(\vec{a})$ that are expected based on the model under consideration and depend on parameters $\vec{a} = (a_1, a_2, \ldots)$. The parameters are to be estimated in such a way that the model

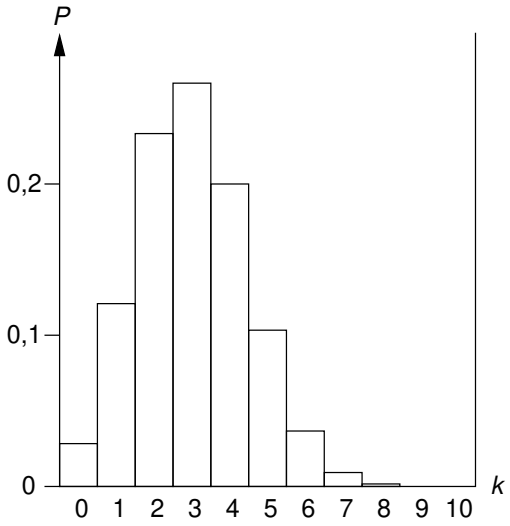

Figure 8: Binomial distribution with $n = 10$, $p = 03$. The probability that an unstable nucleus emits a γ quantum when it decays, is 30 %. How many of ten observed decays show a γ emission?

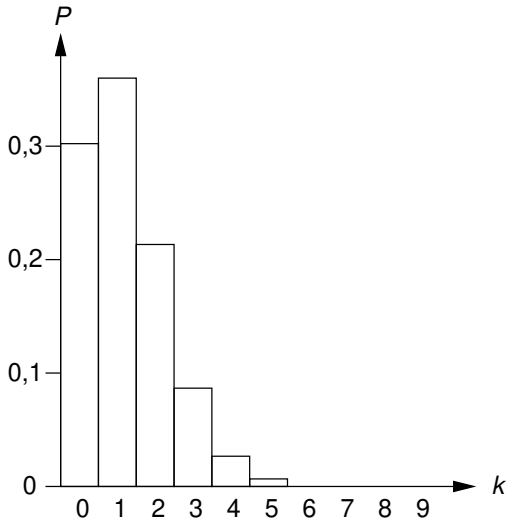

Figure 9: Poisson distribution with $\mu = 1.2$. A radioactive sample contains $3 \cdot 10^{12}$ unstable nuclei, each of which decays with a probability of $4 \cdot 10^{-13}$ within the next second. How many decays are observed per second?

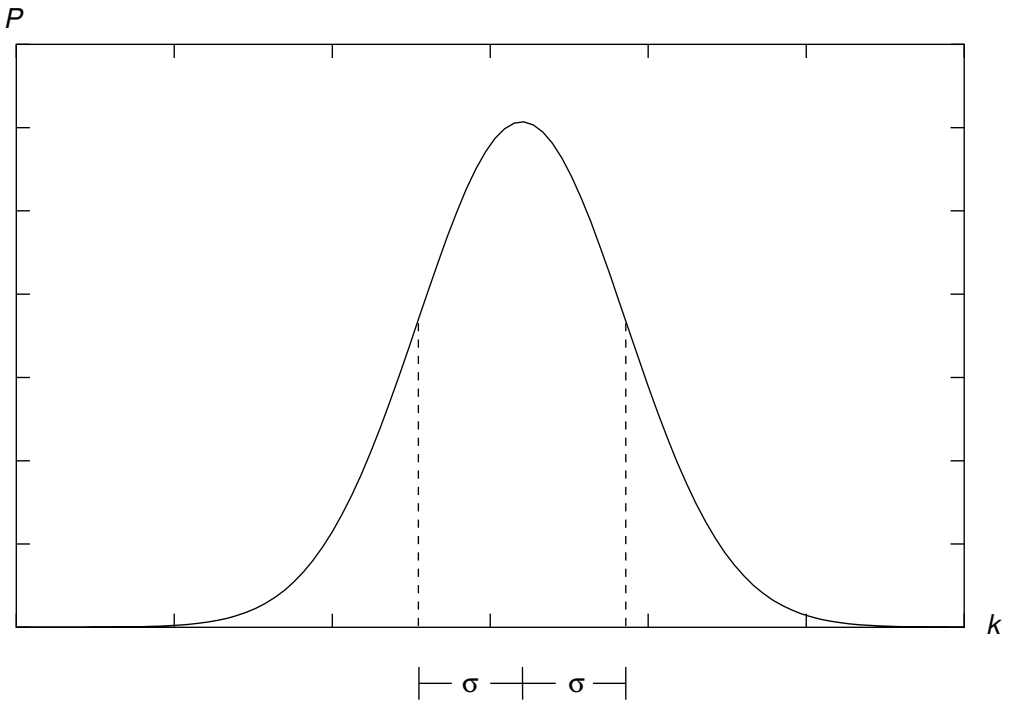

Figure 10: Special Gaussian distribution with $\mu = 4320$, $\sigma = \sqrt{4320}$. The measurement from figure 9 is performed for an hour. How many decays are observed within one hour?

matches the data points as accurately as possible. The function $\chi^2(\vec{a})$ is a measure of the deviation of the measured values from the expected values, where each deviation is weighted using the error on the data point:

$$\chi^2(\vec{a}) = \sum_{i=1}^{n} \frac{(x_i - X_i(\vec{a}))^2}{\sigma_i^2}$$

.

The best match between the data points and the model corresponds to the minimum of $\chi^2(\vec{a})$. Thus, the χ^2 fit consists of varying the parameters $\vec{a}$ of the model, until the minimum of χ^2 has been found. After the optimal estimates of the parameters have been found, the uncertainty on these estimates must be understood as well. The statistical error on the estimated parameter values is obtained from the χ^2 fit by increasing the value of $\chi^2(\vec{a})$ from χ^2_{min} to $\chi^2_{\text{min}} + 1$ (see figure 11).

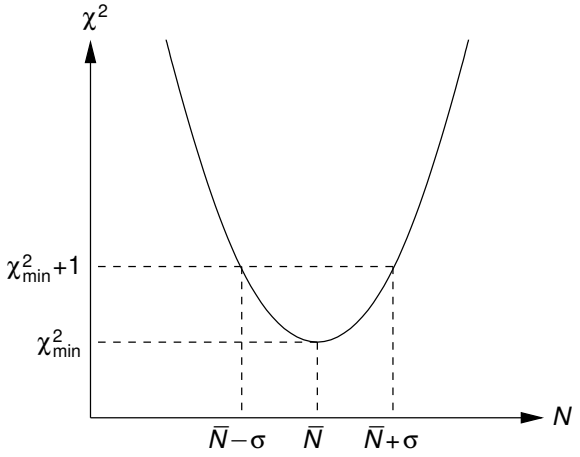

Figure 11: Minimum of the function χ^2

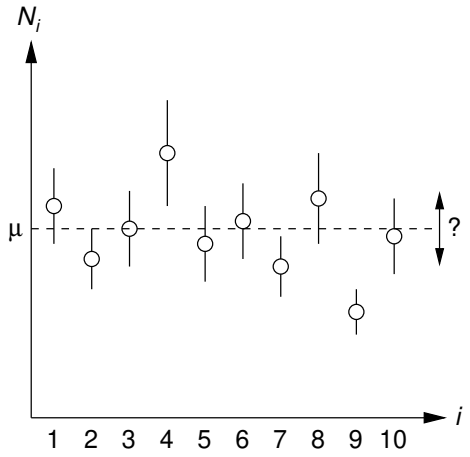

Figure 12: Measurement of ten pulse rates N_i

As an example consider a set of measurements consisting of 10 pulse rates $x_i = N_i$ shown in figure 12. As one expects the same value μ of the pulse rate for every one of the ten measurements, the theoretical expectation is given by $X_i(\mu) = \mu$. In this case, it depends on just a single parameter μ. If $\chi^2(\mu)$ is plotted against μ, the result is a parabolic curve like in figure 11 with a minimum at $\mu = \overline{N}$. The mean $\overline{N}$ is the optimal estimate of the parameter. This example only serves to demonstrate the procedure for finding the χ^2 fit.

3.4 The χ^2 Test

The probability that a repetition of the experiment yields a larger value for χ^2, despite identical experimental conditions is an important consideration of many experimental analyses. In statistics, it can be shown that, for large statistical samples, the probability density associated with χ^2 (the so-called χ^2 distribution) is of the following form:

$$P_n(\chi^2) = \frac{1}{2\,\Gamma\left(\frac{n}{2}\right)} \left(\frac{\chi^2}{2}\right)^{\frac{n-2}{2}} \exp\left(-\frac{\chi^2}{2}\right)$$

The number of degrees of freedom, n, is given by the number of independent equations involving the parameter vector $\vec{a} = (a_1, a_2, \ldots)$ and the observation vector $\vec{x} = (x_1, x_2, \ldots)$. One can determine the number of degrees of freedom by subtracting the number of variable parameters from the number of data points, where in many cases one additional degree of freedom is lost due to the normalization constraint on the theoretical distribution. Shown in figure 13 is the probability distribution $P(\chi^2)$ for $n = 1, 2, 3, 5, 10$ and 20. The curves $P_n(\chi^2)$ take on their maximum slightly before $\chi^2 = n$.

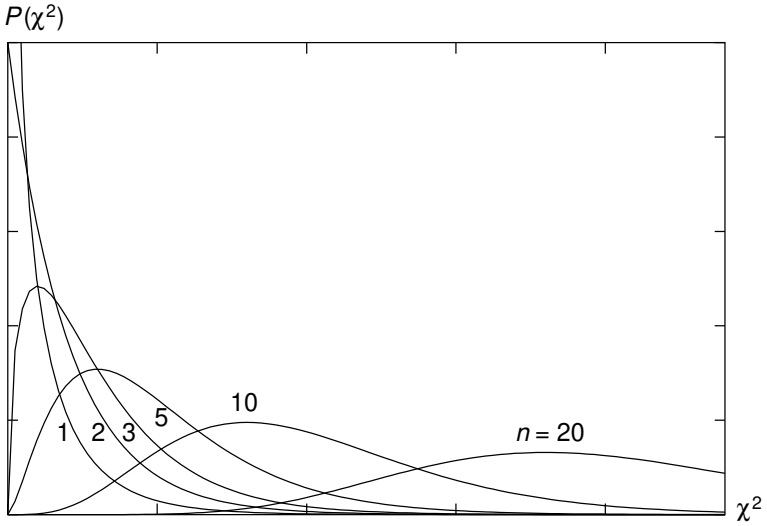

Figure 13: χ^2 distribution $P_n(\chi^2)$ for different values of n.

The integral

$$F_n(\chi^2) = \int_{\chi^2}^{\infty} P_n(\tilde{\chi}^2)\, d\tilde{\chi}^2$$

is the probability, that a larger value of χ^2 is found when the experiment is repeated. Therefore, $F_n(\chi^2)$ is a measure of the goodness of the hypothesis, that the experimental data are correctly described by the existing theory. Very small probabilities ($F < 0{,}05$) indicate a poor agreement between experiment and theory. The function $F_n(\chi^2)$ has been plotted for several values of n in figure 14.

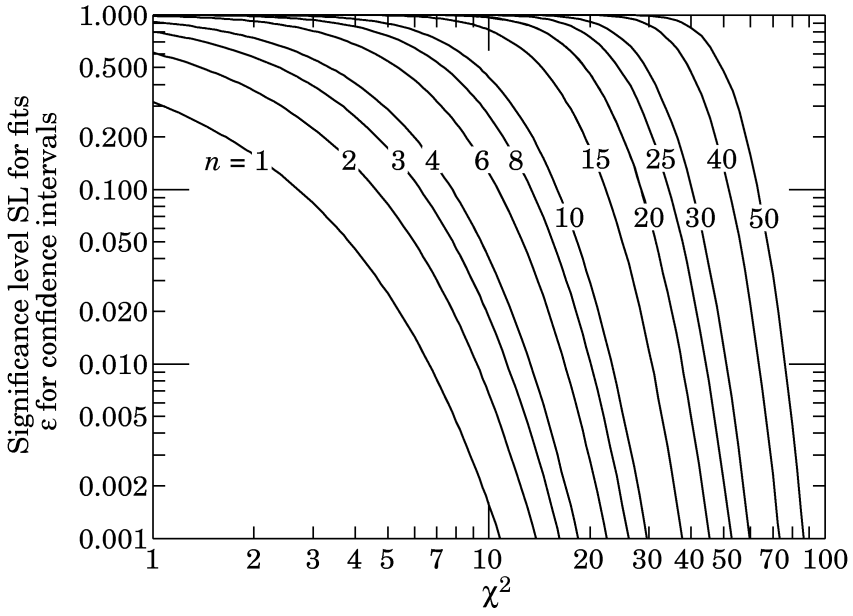

Figure 14: The function $F_n(\chi^2)$ for several values of n.

4 Supporting Hardware: Raspberry Pi, Arduino, and Python

4.1 Raspberry Pi

The Raspberry Pi (often referred to as Pi) is a brand of small single-board computer. The particular version that is used in the lab course is the Raspberry Pi 4. These boards have many uses from becoming replacements for conventional desktop computers to application specific functions such as data collection or controlling experiment hardware. The Raspberry Pi has 2x USB 2.0 ports, 2x USB 3.0 ports, a wired ethernet port, 2x microHDMI ports, a USB C port for connection to the AC power adapter, a 3.5 mm audio ouput, as well as Bluetooth and wifi capabilities. The Raspberry Pi 4 is available with 1, 2, and 4 GB or RAM depending on the amount of computation needed. In lab course experiments, you will find either the 2 GB or 4 GB models. On-board storage is supplied by a microSD card, these come in various sizes, but typically 32 or 64 GB is available for storage. The Raspberry Pi utilizes the Raspbian "Buster" OS which is a diluted Debian Linux based OS similar to Ubuntu. This provides a useful terminal line and GUI to make data collection, processing, and analysis easier. Lastly, the Raspberry has 40 General Purpose Input and Ouput (GPIO) pins. These pins can be used to collect data directly using the Raspberry Pi using I2C, SPI, and other protocols, as well as supplying either 3.3 V or 5 V power to other devices.

For this activity, the Raspberry Pi will be used as miniature computer that interfaces to the Arduino data collection and will process the data received from the Arduino. To start using the Raspberry Pi do the following:

- Ensure that the AC power adapter is plugged into the USB C port on the Raspberry Pi and connect to the power strip.

- Ensure that the Raspberry Pi Keyboard and mouse are plugged into the Raspberry Pi USB ports.
- Make sure that the microHDMI adapter is connected between the microHDMI port on the Pi and the DVI connector on the monitor.
- After powering on the Raspberry Pi, both the red power and green action lights should be on.
- During power up you should see a series of raspberries across the top left of the screen.
- After 1-2 minutes you should arrive at the login screen. If you do not, please contact the tutor immediately.
- To login: username = 'pi' and password = 'labCourseRPi'
- If login is successful, it should take you to the Desktop screen

For those not used to using a UNIX/LINUX system, the following common commands will be quite useful:

- pwd is the command for checking your present working directory, this will display the folder location on the terminal line
- ll -thr is the command to list the files and folders in the current directory, the t, h, and r flags list the folders and files according to time, in human readable, reverse chronological order, respectively.
- cd is the command to change directories. Simply add the folder path after cd to switch to the directory you wish to go to.
- mkdir is the command to make a directory. Simply add the folder name after the command to create it.

4.2 Arduino and PIN Photodiode Circuit

Arduino is a propular brand of microcontroller that is used primarily for data collection. These boards have many different inputs, analog as well as digital. These boards can also supply 3.3 V and 5 V to power devices and can be setup to read in data using many different protocols such as I2C and SPI.

For this experiment, the Arduino Nano is essentially inaccessible, so no additional hardware description will be given. If a problem with the hardware is suspected, please consult with the tutor and they will investigate further. Focus will be given here on setting up the Arduino code and interacting with the PIN photodiode circuit. The Arduino code is written in programs called "sketches" using the Arduino Individual Development Environment (IDE). The code is written in a C-style language, those that have programming experience with C++ will find it extremely similar. To start the arduino IDE, open a terminal and type "arduino &". This will open the IDE and bring up an empty template with two functions: "setup()" and "loop()".

The very first step is to declare and initialize the variables that will be needed:

- Create 2 variables: sigPin and noisePin as:

```
int sigPin = 2;
int noisePin = 5;
```

  These pins are hard wired between the Arduino and the PIN circuit and can not be changed.

- Create the following integer variables:
  - 1 to count the number of signal counts e.g.

```
int sigCount = 0;
```

  - 1 to count the number of noise counts
  - 1 to use as a flag to listen for signal counts from the PIN circuit
  - 1 to use as a flag to listen for noise counts from the PIN circuit
  - 1 to keep track of the current time in ms
  - 1 to keep track of the previous time window in ms
  - 1 to count time in seconds

The next step is to fill the "setup()" function. This function is for initializations or commands that should only be done once upon starting the Arduino. Here you should do the following:

- Begin the serial connection with a baud rate of 9600 e.g.

```
Serial.begin(baudRate);
```

- Set the pin mode for both the signal and noise pins. They need to be set as inputs to accept data and initialized to digital high e.g.

```
pinMode(pin, INPUT);
digitalWrite(pin, HIGH);
```

  This is all the initialization that needs to be done

The last part of the code is to define the actions that need to be taken by the Arduino in the "loop()" function. Make the Arduino do the following:

- Have the Arduino read the signal and noise pins and store them as integers e.g.

```
int sig = digitalRead(sigPin);
```

- The PIN is active on digital LOW. If the flag and the signal are both LOW increase the signal count and set the flag HIGH. It is important to change the flag immediately as the signal stays high for a few tens of ms and leaving this HIGH would give an artificially inflated count. Otherwise, set the flag LOW. Do the same for the noise pin:

```
if(sig == 0 && sigOn == 0){
sOn = 1;
sigCount++;

}

else if(sig == 1 && sigOn == 1){
sigOn = 0;
}
```

- Store the current time in milliseconds e.g.

  ```
  currentTime = millis();
  ```

- If the difference between the current time in ms and the previous time edge is 1000 ms (1 s) increase the number of seconds elapsed by 1 and set the previous time edge equal to the current time.

- Send the results to the Raspberry Pi via the Serial connection e.g.

  ```
  Serial.println(String(sigCount) + ', ' + String(totalSec));
  ```

  Make sure that the comma "," is present, because the analysis program will need a delimeter to split on to properly separate the counts from the time.

If you are happy with the Arduino code, click the checkmark in the upper left corner to compile the code. Address any errors. If no errors, look in the "Tools" tab and check that the "Board" registers the Arduino Nano and the "Port " is not empty. Once this is the case and you confirm that the Arduino is powered correctly, click the arrow next to the compile button to upload to the Arduino. If there are no upload errors, under Tools, click the "Serial Monitor" to bring up the monitor that reads the Serial output from the Arduino to confirm that the Arduino is reading as expected. If you run into problems, don't hesitate to contact your tutor.

4.3 Python

PYTHON is a high-level interpreted programming language that has recently found popularity among the physics and computer science community for, among many things, performing data processing and data analysis. Unlike the C-style language that you used for the Arduino coding section, PYTHON does not end commands in semi-colons ";", encapsulate multiline conditionals within curly braces "{}", or require datatype declarations as part of declaring variables. This can make the flow of the code much faster and easier to read. However, PYTHON handles the first two items by the use of white space and indentation. For example, when making an "if" clause to check something:

```
if(condition):
    command1
```

```
    command2
    ...
    lastCommand

Rest of code
```

A PYTHON program will be used to collect the data from the Arduino and store it as a .txt file. To write the code open an editor, e.g. Emacs or nano, e.g. for Emacs:

```
$ emacs -nw <PythonFileName.py>
```

Here the -nw option opens the terminal (non-windowed) version of Emacs. Now you should write a program that does the following:

- Import the serial and argparse libraries that will allow PYTHON to communicate with the Serial line and take in command line arguments, respectively. e.g.

  ```
  import serial
  ```

- Create an Argument Parser object e.g,

  ```
  parser = argparse.ArgumentParser()
  ```

- Create 3 argparse variables to take in info from the command line for the output file name, data collection time, and number of data points e.g.

  ```
  parser.add_argument('-o', type=str, help="output file name")
  ```

  The add_argument requires the following arguments: the name of the agrument, the type definition, and a help message. The output name should be a string, time a float, and number of points an int, respectively.

- Next parse the arguments given from the command line e.g.

  ```
  args = parser.parse_args()
  ```

- Create a filename object and utilize the input from the command line e.g.

  ```
  outName = args.o + ".txt"
  ```

- Open a text file to store the incoming data e.g.

  ```
  outFile = open(outName, 'w')
  ```

  The 'w' flag tells Python to overwrite the existing file if a file of the same name exists.

- Create a serial connection variable passing the port name from the "Port" variable in the Arduino IDE as the first argument and the baud rate as the second argument e.g.

```
ser = serial.Serial('/dev/ttyACM0',9600)
```

- Create the following variables and initialize them to 0:
  - 1 variable to count the total number of counts
  - 1 variable to store the previous total of counts from the previous time window
  - 1 variable to store the count for the current data point
  - 1 variable to store the total number of elapsed seconds
  - 1 variable to store the number of data points
  - 1 variable that is a temporary time keeper
- Create the "while" loop to keep the Serial connection open e.g.

```
while True:
    command1
    command2
    ...
```

- Read in the time and store it temporarily
- read the Serial line and store it as a list that splits on the comma "," e.g.

```
line = ser.readline()
listOfData = line.split(',')
```

- Create a for loop that gets the count and the elapsed time.
- Check to see if the modulus of elapsed time with the time window is 0. If this is true, set the data point count to be the difference between the total count and previous count.
- increase the number of data point variable by 1
- write these to file
- Update the previous count variable to be the current total count
- Lastly, check to see if the number of data points are equal to the total requested from the command line.

4.4 Running PYTHON From the LINUX Terminal

Running the PYTHON program that you have just made from the LINUX Terminal is straight forward. After you have finished creating your PYTHON program, you can run your program by:

```
$ python <PythonFileName.py> -a <value> -b <text> -z <value>
```

Here the flags 'a', 'b', and 'z' are options that you have specified using your ArgumentParser() within your PYTHON program. The spaces are important, however, the <value> and <text> should match with the expected data type for the flag specified in the PYTHON program. One should note that if the variable/flag is expecting a string, this should be placed in between double quotes, e.g. "FileName.txt". If you have specific question about how to run this PYTHON program please ask your laboratory supervisor.

5 Procedure

5.1 Data collection using the Cobra3 Geiger-Müller (GM) counter

1. Place the Sr-90 source inside the lead enclosure with the exposed opening of the source centered on the active area of the GM counter.

2. Make sure that the Cobra3 measurement aparatus is connected to wall power and the green indicator has a solid green light.

3. On the desktop PC, open the Cobra3 measurement program “Measure” by double clicking the icon “m” as shown in figure 15.

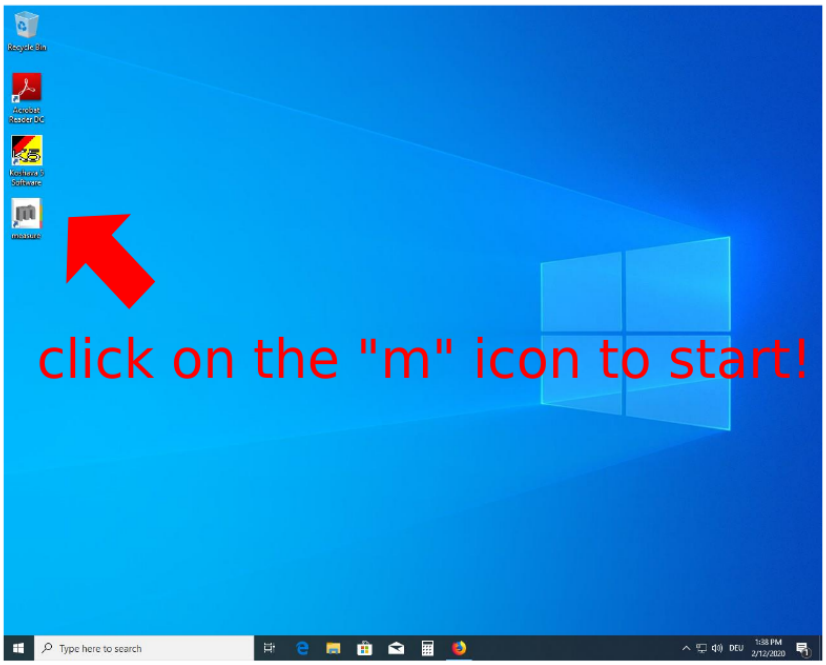

Figure 15: The main Desktop for using the Cobra3 GM. To start the “Measure” program, double click on the “m” icon.

4. To collect data click the red record button on the left side of the menu. This will bring up the measurement menu window as shown in figure 16.

5. In the measurement menu window select the following: for “XData” choose “Time t/s”, for “Unit” select “Impulses”, for “Time inter” select a time window value that will give a mean of (at least) 25 if enough data points are taken to give a Gaussian distribution. The other options are not critical and can be selected or not as you prefer. These settings are shown in figure 17

6. Once satisfied with the settings, submit them. This will bring up the main measurement window.

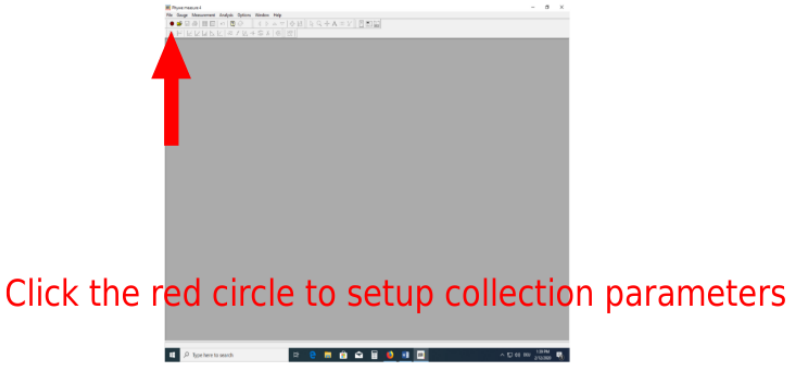

Figure 16: The main opening screen in "Measure" program. Select the red circle to start setting parameters for data collection

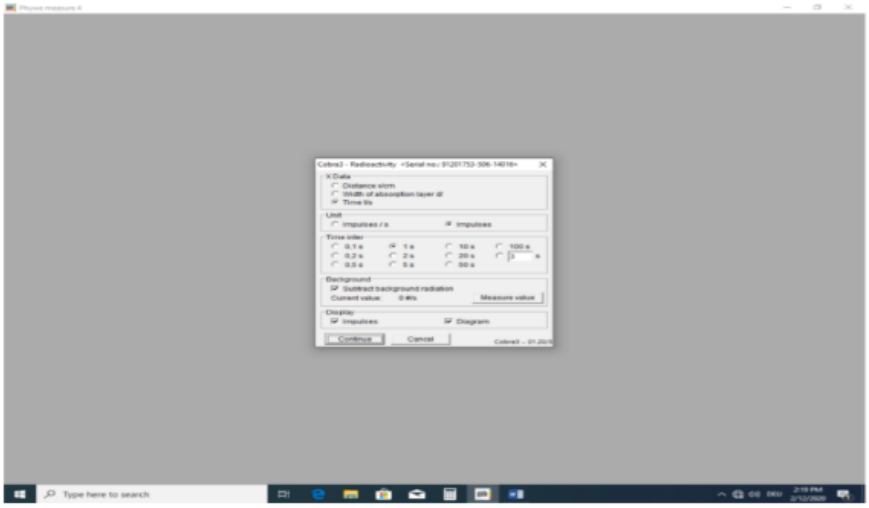

Figure 17: The diaglog box for selecting data collection parameters for the Cobra 3 GM counter in Measure

7. Click "Start measurement" as shown in figure 18 and allow the program to run collecting enough data points to produce a Gaussian distribution.

8. Once the appropriate number of events have been collected, save the measurement.

9. The file is saved as a .msr file. To store this as a file that can be easily read for analysis pruposes, export the data to a .txt file.

10. Remove the Sr-90 source from the lead enclosure and repeat the same process for cosmic ray muons with the following modifcations:

    - Collect enough data points to produce a Poisson distribution.
    - Use a time window that will give a mean no larger than 2.

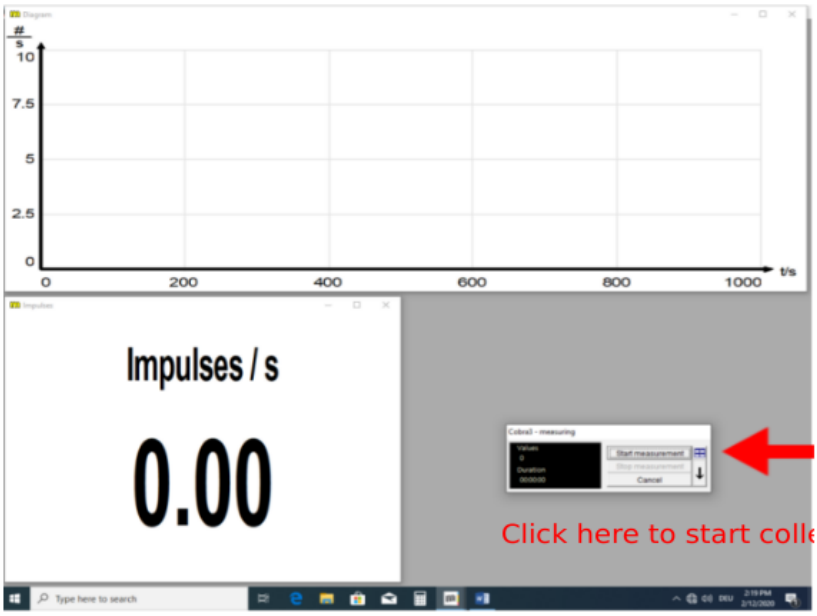

Figure 18: The progress window to track the progress of the measurement. To start a data collection run, click the "Start measurement" button.

5.2 Data collection using the PIN semiconductor counter

1. Power on the Raspberry Pi and open the Arduino Individual Development Environment (IDE)

2. Write a simple Arduino program to collect data from the PIN counter.

3. Once the program compiles without error, upload to the Arduino. If you encounter significant difficulties with coding, please consult your tutor.

4. Check to make sure that the data collectd by the Arduino makes sense with cosmic ray muons by using the Serial Monitor in the IDE.

5. After making any necessary changes to the Arduino program, write a simple Python program to read the data from the Arduino and save it to file. Again, if you run into difficulties with Python coding, please ask your tutor.

6. Once this is ready, collect enough cosmic ray events to produce a Poisson distribution with a mean no larger than 2.

7. After the cosmic ray data has been collected, the Sr-90 source will likely be available to use. Collect data similar to that done with cosmic ray muons except for the following:

    - Collect enough data points to produce a Gaussian distribution.
    - Use a time window that will give a minimum mean of 25.

5.3 Measurement of the Characteristic Curve of a Proportional Counter

The proportional counter must be flushed continuously with a counting gas, argon-methane ($ArCH_4$) in our case. Set the gas flow at approximately 4 divisions on the flow meter. Bubbles should be visble in the viewing glass. **The maximal voltage for this counter is 2 kV, so make sure to pay attention to the switch next to the voltage control; this multiplies the set voltage by the value indicated by the knob.**

1. Insert the Am-241 source into the sample changer and measure the pulse rate as well as the pulse height (measure of gas amplification) of the α particles as a function of voltage. Choose suitable voltage steps. Why are the pulses not all equally large, even though the particles are monoenergetic?

2. Repeat the measurement using the C-14 source.

3. Measure the cosmic ray muon background: measure the pulse rate without a source in the counter. It makes sense to use the same voltage steps used before.

4. Choose an operating voltage of $V = V_G + 100\,\mathrm{V}$. From here on, this setting should not be changed any more.

5. Check the intensity of the background: measure the pulse rate without a source underneath the counter.

6. Produce a Stever diagram using the oscilloscope. For this, use the direct output of the proportional counter. Estimate the dead time and recovery time of the counter.

7. Use the pulses of the 1-MHz-generator, which are available from the backside of the pulse counter, and verify the electronically generated dead times indicated on the dead time stage. To this end, use the pulse counter and write down the respective pulse rates.

8. The pulses from the proportional counter should now pass through the dead time stage before reaching the counter. Measure the count rates for a small and a large dead time of the dead time stage and determine from this the dead time of the counter. Make sure to measure a sufficiently large amount of events per time setting.

5.4 Gain Measurement of a Proportional Counter

We have several different mixtures of gases ($ArCH_4$ and $ArCO_2$). Perform the following test for at least three different gas mixtures. Make a coarse voltage scan similar to before to locate the plateaus for the different gas mixtures. Select three voltage points on the β plateau for Sr-90 and Am-241 and three voltage points on the α plateau for Am-241. For each voltage point collect enough sets of counts in 1s time intervals to get a well-measured Gaussian average for this data point. Then move to the next data point. Once all of the voltage points for the gas mixture have been collected change the gas mixture and allow $\mathcal{O}$(5 minutes) for the gas mixture to flow before collecting data. Repeat this for all gas mixtures.

5.4.1 In Case This Part of the Experiment is Done Before the Geiger-Müller Counter

- If you have some time left, you can already do point 2 from the measurements on statistics using the proportional counter. For this purpose, pick a voltage on the α plateau.
- Set the measurement time such that the **average event count** per measurement is at most 2. Take enough measurements to be able to produce a Poisson distribution later on.

6 Analysis

6.1 Gas GM and Statistics

1. Split the measured values into sensible intervals and present them as histograms.
2. For each of the measurements (Gaussian and Poisson distributions), calculate the mean and the standard deviation. From each of these, calculate the expected distribution and plot it together with the measured distribution.
3. Test the goodness of your prediction of the measured distribution by doing a χ^2 test for each distribution. Calculate the χ^2 between the expected and measured distributions and quantify the probability of the hypothesis using figure 14. Only consider intervals containing more than 4 entries.
4. Using a program (e.g. Origin, Maple, Mathematica, Root, ...), perform χ^2 fits to the measurements. For each distribution, compare the mean, standard deviation and χ^2 with point 2 and 3.
5. Compare and contrast the results obtained by the two different types of GM counters. Do the reults make sense? What could be causing observed differences?

6.2 Solid State GM and Statistics

1. Repeat the steps from section "Gas GM and Statistics"
2. How do the Gaussian plots compare? How do the Poisson measurements compare?
3. Compare and contrast the results obtained by the two different types of GM counters. Do the reults make sense? What could be causing observed differences?

6.3 Measurements with the Proportional Counter

1. Calculate the resolution time using the dead time stage and compare this with the results obtained using the oscilloscope. Plot the dead time correction.
2. Plot the counter's characteristic curve for the α source and again for the β source. Include the correction for the dead time.
3. Plot the measured pulse rates of the three samples versus voltage and discuss the differences.
4. Plot the pulse heights of the samples as a function of the voltage in a suitable scale. Justify your choice. Discuss the curves.
5. With the data points from the gas mixtures, plot the number of counts as a function of voltage.
6. For each voltage point generate a relative gain plot taking the $ArCH_4$ values as the reference.
7. Discuss how the gas composition affects the gain of the proportional counter.

Appendix

A Properties of the Radioactive Sources Used

- $^{241}_{95}$Am: Source is open, placed in recessed and covered enclosure to prevent direct contact. Substance on the front side, 1 mm in diameter, the soiurce is sunk 1.5 mm into the casing; covering foil in front of the radioactive substance: 3 μm of Gold ($\rho = 19.3\,\mathrm{g/cm^3}$). Decay chain:

$$^{241}_{95}\mathrm{Am} \xrightarrow[T_{1/2}\text{: } 433\,\mathrm{y}]{\alpha\,(5{,}65\,\mathrm{MeV})} {}^{237}_{93}\mathrm{Np} \xrightarrow[T_{1/2}\text{: } 2{,}14\cdot10^{6}\,\mathrm{y}]{\alpha\,(4{,}96\,\mathrm{MeV})} {}^{233}_{91}\mathrm{Pa} \xrightarrow[T_{1/2}\text{: } 27{,}0\,\mathrm{d}]{\beta^-\,(0{,}57\,\mathrm{MeV})} {}^{233}_{92}\mathrm{U} \xrightarrow[T_{1/2}\text{: } 1{,}59\cdot10^{5}\,\mathrm{y}]{\alpha\,(4{,}91\,\mathrm{MeV})} {}^{229}_{90}\mathrm{Th}$$

$$^{229}_{90}\mathrm{Th} \xrightarrow[T_{1/2}\text{: } 7932\,\mathrm{y}]{\alpha\,(5{,}17\,\mathrm{MeV})} {}^{225}_{88}\mathrm{Ra} \xrightarrow[T_{1/2}\text{: } 14{,}9\,\mathrm{d}]{\beta^-\,(0{,}36\,\mathrm{MeV})} {}^{225}_{89}\mathrm{Ac} \xrightarrow[T_{1/2}\text{: } 10{,}0\,\mathrm{d}]{\alpha\,(5{,}94\,\mathrm{MeV})} {}^{221}_{87}\mathrm{Fr} \xrightarrow[T_{1/2}\text{: } 286\,\mathrm{s}]{\alpha\,(6{,}46\,\mathrm{MeV})} {}^{217}_{85}\mathrm{At}$$

$$^{217}_{85}\mathrm{At} \xrightarrow[T_{1/2}\text{: } 32{,}3\,\mathrm{ms}]{\alpha\,(7{,}20\,\mathrm{MeV})} {}^{213}_{83}\mathrm{Bi} \xrightarrow[T_{1/2}\text{: } 45{,}6\,\mathrm{m}]{\beta^-\,(1{,}42\,\mathrm{MeV})} {}^{213}_{84}\mathrm{Po} \xrightarrow[T_{1/2}\text{: } 3{,}72\,\mu\mathrm{s}]{\alpha\,(8{,}54\,\mathrm{MeV})} {}^{209}_{82}\mathrm{Pb} \xrightarrow[T_{1/2}\text{: } 3{,}25\,\mathrm{h}]{\beta^-\,(0{,}64\,\mathrm{MeV})} {}^{209}_{83}\mathrm{Bi}$$

- $^{90}_{38}$Sr: Enclosed source; perspex disk mounted in sheet steel. Decay chain:

$$^{90}_{38}\mathrm{Sr} \xrightarrow[T_{1/2}\text{: } 28{,}9\,\mathrm{y}]{\beta^-\,(0{,}55\,\mathrm{MeV})} {}^{90}_{39}\mathrm{Y} \xrightarrow[T_{1/2}\text{: } 64{,}1\,\mathrm{h}]{\beta^-\,(2{,}28\,\mathrm{MeV})} {}^{90}_{40}\mathrm{Zr}$$

- $^{14}_{6}$C: Enclosed source, decay:

$$^{14}_{6}\mathrm{C} \xrightarrow[T_{1/2}\text{: } 5700\,\mathrm{y}]{\beta^-\,(0{,}16\,\mathrm{MeV})} {}^{14}_{7}\mathrm{N}$$