



**Neutron capture reactions
for nuclear astrophysics:
Development & characterization
of an innovative detection setup
based on trans-Stilbene organic scintillators**

M.Sc. Thesis

Papanikolaou Dimitrios

Academic Supervisor: prof. Patronis Nikolaos

MSc in Physics with Specializations in Theoretical and Experimental Physics

Registration Number: 828

Ioannina (Greece)

22/07/2024



**Αντιδράσεις σύλληψης νετρονίων-
για πυρηνική αστροφυσική:
Ανάπτυξη και χαρακτηρισμός
μίας καινοτόμου ανιχνευτικής διάταξης
βασισμένης σε οργανικούς σπινθηριστές trans-Στιλβένης**

Μεταπτυχιακή Διπλωματική Εργασία

Παπανικολάου Δημήτριος

Ακαδημαϊκός επιβλέπων: **Πατρώνης Νικόλαος**

ΠΜΣ στη Φυσική με Ειδικεύσεις στη Θεωρητική και στην Πειραματική Φυσική

Αριθμός Μητρώου: **828**

Ιωάννινα (Ελλάδα)

22/07/2024

I) Acknowledgments

I would like to express my deepest gratitude to my supervisor Prof. Nikolaos Patronis for his invaluable guidance, support and encouragement throughout the course of my research. His expertise and insights have been instrumental in shaping this work and I am deeply grateful for his mentorship.

I also extend my appreciation to Associate Prof. Agatino Musumarra and Dr. Pellegriti Maria Grazia for their significant contributions and support. Their collaboration and insights have greatly enriched my research. I am particularly thankful for their thoughtful feedback and the many stimulating discussions we have had.

Moreover, I wish to acknowledge the fruitful and long-standing collaboration between the University of Ioannina and the INFN-CT (Istituto Nazionale di Fisica Nucleare - Sezione di Catania) and the Department of Physics in Catania (DFA). This partnership has provided an enriching environment for research which has remarkably contributed to the success of this project.

Finally, my sincere thanks also goes to my colleagues and the administrative staff at the University of Ioannina and our partner institutions for their unwavering support and assistance. This work would not have been possible without their collective effort and dedication.

II) Ευχαριστίες

Θα ήθελα να εκφράσω τις βαθύτερες ευχαριστίες μου στον επιβλέποντα καθηγητή μου Καθ. Νικόλαο Πατρώνη για την ανεκτίμητη καθοδήγηση, υποστήριξη και ενθάρρυνση καθ' όλη τη διάρκεια της έρευνάς μου. Η τεχνογνωσία και οι γνώσεις του συνέβαλαν καθοριστικά στη διαμόρφωση αυτής της δουλειάς, και είμαι βαθιά ευγνώμων για την καθοδήγησή του.

Εκφράζω επίσης την εκτίμησή μου στον Αναπληρωτή Καθ. Agatino Musumarra και την Dr. Pellegriti Maria Grazia για τη σημαντική συνεισφορά και την υποστήριξή τους. Η συνεργασία και οι γνώσεις τους έχουν εμπλουτίσει πολύ την έρευνά μου. Είμαι ιδιαίτερα ευγνώμων για τα στοχαστικά σχόλιά τους και τις πολλές ενδιαφέρουσες συζητήσεις που είχαμε.

Επιπροσθέτως, θα ήθελα να αναγνωρίσω τη γόνιμη και μακροχρόνια συνεργασία μεταξύ του Πανεπιστημίου Ιωαννίνων και του INFN-CT (Istituto Nazionale di Fisica Nucleare - Sezione di Catania) και του Τμήματος Φυσικής στην Κατάνια (DFA). Αυτή η συνεργασία παρείχε ένα εμπλουτιστικό περιβάλλον για έρευνα που συνέβαλε σημαντικά στην επιτυχία αυτού του έργου.

Τέλος, θα ήθελα να ευχαριστήσω επίσης τους συναδέλφους μου και το διοικητικό προσωπικό του Πανεπιστημίου Ιωαννίνων και τα συνεργαζόμενα ιδρύματά μας για την αμέριστη υποστήριξη και βοήθειά τους. Αυτό το έργο δεν θα ήταν δυνατό χωρίς τη συλλογική προσπάθεια και την αφοσίωσή τους.

III) Abstract

Neutron capture reactions are crucial for understanding stellar nucleosynthesis beyond iron and the formation of elements in the universe. This dissertation presents the development and characterization of an innovative detection setup using organic scintillators made of trans-Stilbene, aiming to improve neutron capture measurements for nuclear astrophysics.

For this dissertation, trans-Stilbene detectors enclosed in carbon fiber housings, coupled with Hamamatsu photomultipliers, were characterized. These detectors were developed at the DFA (Dipartimento di Fisica e Astronomia "Ettore Majorana", Catania) and the INFN-CT facilities.

Beyond the characterization of each individual detector, a fully symmetrical multi-detector arrangement was developed and characterized, placing the trans-Stilbene units in a regular tetrahedral pyramid for optimal geometric configuration. The detectors were tested for different operation modes (pulse height/charge) and their ability to distinguish neutron/photon pulses (PSD – pulse shape discrimination). Moreover, a pioneering approach using multivariate analysis through Principal Component Analysis (PCA), facilitated a method of handling multiple characteristic parameters of a pulse.

The aim of this work is to study these detectors and investigate their potential to replace the existing C_6D_6 detectors at n_TOF, as well as to explore possible applications in the study of elastic and inelastic neutron scattering reactions.

IV) Περίληψη

Οι αντιδράσεις σύλληψης νετρονίων είναι ζωτικής σημασίας για την κατανόηση της αστρικής πυρηνοσύνθεσης πέρα από τον σίδηρο και το σχηματισμό στοιχείων στο σύμπαν. Αυτή η διατριβή παρουσιάζει την ανάπτυξη και τον χαρακτηρισμό μιας καινοτόμου διάταξης ανίχνευσης χρησιμοποιώντας οργανικούς σπινθηριστές *trans-Stilbene*, με στόχο τη βελτίωση των μετρήσεων σύλληψης νετρονίων για την πυρηνική αστροφυσική.

Στην παρούσα διατριβή έγινε ο χαρακτηρισμός ανιχνευτών *trans-Stilbene* σε περιβλήματα από ανθρακονήματα, συζευγμένους με φωτο-πολλαπλασιαστές Hamamatsu οι οποίοι αναπτύχθηκαν τις εγκαταστάσεις του DFA (Dipartimento di Fisica e Astronomia "Ettore Majorana", Κατάνια) και στο INFN-CT.

Πέρα από τον χαρακτηρισμό του κάθε ανιχνευτή ξεχωριστά αναπτύχθηκε και χαρακτηρίστηκε μια πλήρως συμμετρική διάταξη πολλαπλών ανιχνευτών, τοποθετώντας τις μονάδες *trans-Stilbene* σε μία κανονική τετραεδρική πυραμίδα για βέλτιστη γεωμετρική διαμόρφωση. Οι ανιχνευτές ελέγχθηκαν για διαφορετικούς τρόπους λειτουργίας (*pulse height/charge*) αλλά και ως προς την δυνατότητα διαχωρισμού παλμών νετρονίων/φωτονίων (*PSD – pulse shape discrimination*). Επιπροσθέτως, μια πρωτοποριακή προσέγγιση χρησιμοποιώντας πολυμεταβλητή ανάλυση μέσω *Principal Component Analysis (PCA)*, διευκόλυνε την μέθοδο χειρισμού πολλαπλών χαρακτηριστικών παραμέτρων ενός παλμού.

Στόχος της παρούσας εργασίας είναι η μελέτη των ανιχνευτών αυτών και η διερεύνηση των δυνατοτήτων που παρέχουν ώστε να αντικαταστήσουν τους ήδη υπάρχοντες ανιχνευτές C_6D_6 του *n_TOF* αλλά και να διερευνηθούν πιθανές εφαρμογές στην μελέτη αντιδράσεων ελαστικής και ανελαστικής σκέδασης νετρονίων.

V) Abbreviations list

α	alpha particle (^4He nucleus)
γ	gamma particle
n	neutron
p	proton
D.....	Deuterium (^2H)
Fe	Iron
Ge	Germanium
Co	Cobalt
Cs	Cesium
pStil	protonated Stilbene ($\text{C}_{14}\text{H}_{12}$)
CERN	Conseil Européen pour la Recherche Nucléaire
EAR1	Experimental Area 1
EAR2	Experimental Area 2
INFN-CT	Istituto Nazionale di Fisica Nucleare - Sezione di Catania
DFA	Dipartimento di Fisica e Astronomia "Ettore Majorana", Catania
eV	electron Volt
MeVee	Mega electron Volt electron-equivalent
GEANT	Geometry And Tracking
n_TOF	neutron Time Of Flight
PS	Proton Synchrotron
PMT	Photo Multiplier Tube
DAQ	Data AcQuisition
CFD	Constant Fraction Discriminator
PSD	Pulse Shape Discrimination
PCA	Principal Component Analysis
PC	Principal Component
Savgol	Savitzky Golay

Contents

<i>Acknowledgments</i>	<i>(I)</i>
<i>Ευχαριστίες</i>	<i>(II)</i>
<i>Abstract</i>	<i>(III)</i>
<i>Περίληψη</i>	<i>(IV)</i>
<i>Abbreviations list</i>	<i>(V)</i>

1. Introduction

1.1 Motivation	2
1.2 Nuclear Astrophysics – The Physics Case	3
1.3 Nuclear Reactions	6
1.4 Interaction of particles with matter	8
1.4.1 . . . Interaction of Photons with Matter	8
1.4.2 . . . Interaction of Neutrons with Matter	10
1.5 Scintillation detector principles	15
1.5.1 . . . Scintillation Mechanism in Organics	15
1.5.2 . . . Types of Organic Scintillators	17
1.5.3 . . . Response of Organic Scintillators	18

2. Experimental Set-Up & Tool-kits

2.1 The n_TOF facility & Neutron Capture Reactions	22
2.1.1 . . . General Overview	22
2.1.2 . . . During LS2 (Long Shutdown 2) and Afterwards	23
2.2 Experimental Set-up	25
2.2.1 . . . Crystal Characteristics & p-Stil modules	25
2.2.2 . . . Multi-Detector Array	26
2.2.3 . . . GEANT4 Simulation	28
2.3 Analysis Tools	29
2.3.1 . . . GEANT4	29
2.3.2 . . . ROOT	31
2.3.3 . . . Python	31

3. Detector Characterization

3.1	Detection Efficiency	33
3.2	Pulse-height/Charge Spectra & Energy Calibration	38
3.3	Rise/Fall time measurements	41
3.4	γ - γ Time Coincidence measurements	44
3.4.1	Time Pick-Off Methods	44
3.4.2	Time Resolution Determination	48

4. Pulse Shape Analysis (PSA)

4.1	Pulse Shape Discrimination (PSD)	50
4.1.1	PSD Properties in Organic Scintillators	50
4.1.2	FOM (Figure Of Merit)	51
4.2	Principal Component Analysis (PCA)	52

5. Results

5.1	PSD - Results	56
5.1.1	PSD Method & Useful Parameters	56
5.1.2	PSD Distributions	58
5.2	PCA - Results	62
5.2.1	PCA Method & Useful Parameters	62
5.2.2	Implementations	62

6. Conclusions

6.1	Conclusions	69
------------	-------------	-----------

7. Bibliography

7.1	Bibliography	71
------------	--------------	-----------

8. Appendices

8.1	Appendix A	77
8.2	Appendix B	81
8.3	Appendix C	92

Chapter 1 - Introduction

- *Motivation*
- *Nuclear Astrophysics – The Physics Case*
- *Nuclear Reactions*
- *Interaction of Particles with Matter*
- *Scintillation Detector Principles*

1.1 – Motivation

The motivation for this research stems from the need to improve neutron capture measurements, which are essential for understanding the processes of stellar nucleosynthesis. Traditional detectors, like C_6D_6 , face challenges related to toxicity and flammability, prompting the exploration of safer and more efficient alternatives. Trans-Stilbene organic scintillators from both INRAD and PROTEUS offer excellent neutron-gamma discrimination, good timing performance and non-toxic properties, making them ideal candidates for this application. By developing and characterizing a new detection setup based on these scintillators, this work aims to provide a robust and innovative solution for high-precision neutron capture studies in nuclear astrophysics, particularly aimed at (n,γ) experiments at the n_TOF facility at CERN.

s-Process:

The s-process occurs in relatively low neutron flux environments, typically found in asymptotic giant branch (AGB) stars. During this process, a nucleus captures a neutron and, after a series of beta decays, transforms into a heavier element. The s-process is characterized by slow neutron capture rates relative to beta decay rates, allowing nuclei to move along the valley of stability in the nuclear chart. This process is responsible for the synthesis of many elements in the mass range of $A=60$ to $A=90$, as well as heavier nuclei up to bismuth ($Z=83$) [5, 19]. To estimate the impact of nucleosynthetic processes on a nuclear species A we can solve the differential equation that describes the time-dependent variation of its abundance.

$$\frac{dN_A}{dt} = -N_A N_B \langle \sigma v \rangle_{AB} + N_C N_D \langle \sigma v \rangle_{CD} \quad (1)$$

Where, N_i ($i : A - D$) are the abundances - number density - of species i , $\langle \sigma v \rangle_{ij}$, the reaction rate per unit pair for the interaction between species i and j or is the thermally averaged neutron-capture cross section for the stable isobar for the interaction between species i and j .

For the s-process, the time variation of the abundance of a heavy element with atomic mass A , if Φ is the neutron flux integrated on the duration of the process we have:

$$d\Phi = n_n(t) v(t) dt \quad (2)$$

Then

$$\frac{dN_A}{d\Phi} = -\sigma_A N_A + \sigma_{A-1} N_{A-1} \quad (3)$$

Where n_n is the neutron density, v is the relative velocity, N_A and N_{A-1} are the abundances of the elements (species) A and $A-1$ and σ_A and σ_{A-1} the cross sections of $A(n,\gamma)(A+1)$ and $(A-1)(n,\gamma)A$ [42].

r-Process:

On the contrary, the r-process takes place in environments with high neutron densities, such as those found in supernovae or neutron star mergers. In these scenarios, nuclei rapidly capture neutrons before they have a chance to beta decay, resulting in the creation of highly neutron-rich, unstable isotopes. These isotopes subsequently undergo a series of beta decays, moving back toward stability and forming many of the heaviest elements in the periodic table, including those beyond Lead ($Z=82$) and up to Uranium ($Z=92$).

The r-process nucleosynthetic path varies a lot from the s-process path. It covers an area in the Z, N plane where the very neutron-rich nuclei are generally unstable. Neutron capture only stops when the probability becomes larger for the (γ, n) reaction than the (n, γ) reaction, as aforementioned. This occurs when the binding energy of the last neutron in the $(Z, A+1)$ nucleus is so weak that the (Z, A) nucleus cannot absorb it. This point on the (Z, N) plane is denoted as the “waiting point”. The nucleus undergoes no reaction until a beta decay occurs, which makes its atomic number increase by one unit from Z to $Z+1$. Then, neutron absorption proceeds on the atomic number $(Z+1)$ and so forth.

Under these conditions, the relative abundance of each sequence of nuclei Z is given by the following equation:

$$\frac{dn_Z(t)}{dt} = \lambda_{Z-1}n_{Z-1}(t) - \lambda_Z n_Z(t) \quad (4)$$

where λ_{Z-1} and λ_Z are respectively the rates of β decay of the waiting points corresponding to the charges $Z-1$ and Z [42].

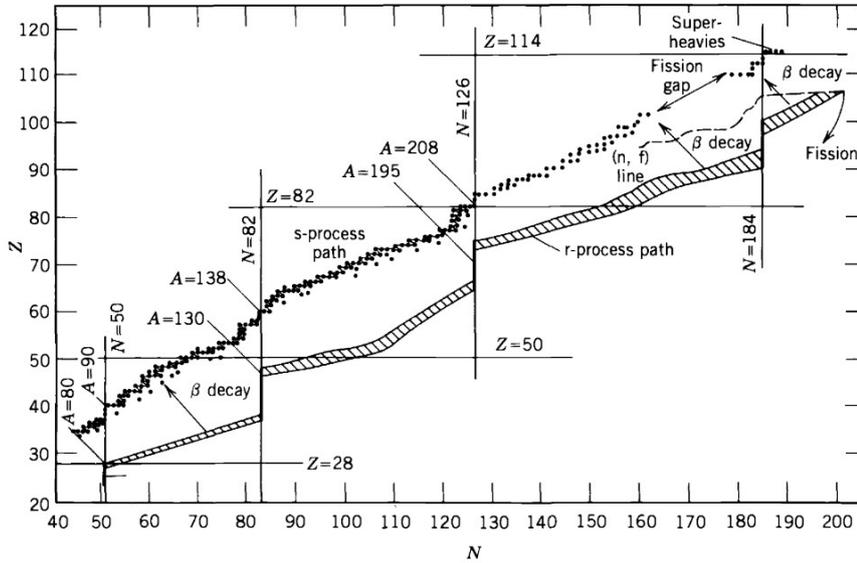


Figure 1.3: Neutron capture paths for r- and s-processes [1]

To summarise, the s- and r-process are two pathways of nucleosynthesis that produce stable isotopes. The above figure shows that the s-process moves in a “zigzag”-like pattern through stable isotopes, ending at ^{209}Bi because no stable isotopes exist beyond this point. In contrast, the r-process can continue until fission half-lives match the r-process capture times, potentially leading to the creation of superheavy nuclei. Near nuclear magic numbers, the r-process path shows vertical rises due to rapid neutron decay to protons, leading to over-abundances of stable nuclei around $A = 80, 130$ and 195 .

As a concluding note, this sub-chapter aims to provide a general introductory overview of nuclear astrophysics, mainly for the sake of comprehensiveness. Future sections will dive a bit deeper into nuclear astrophysics, emphasizing on practical implementations and ongoing research at the n_TOF facility at CERN.

1.3 – Nuclear Reactions

Basic Classification of Nuclear Reactions:

To understand the nature of nuclear reactions, a classification should be introduced based on two important factors, the reaction time and the number of intranuclear collisions which are critical in determining the reaction mechanism. Therefore, all types of nuclear reactions can be divided into three broad categories: *direct reactions*, *pre-equilibrium reactions* and *compound nucleus reactions*.

- **Direct Reactions:** The incident particle mainly interacts with a few, usually with the outermost nucleons of the target nucleus (one or two intranuclear collisions), with the emission of the reaction products occurring in a very short time ($\sim 10^{-22}$ s). As the energy of the incident particle increases, the de Broglie wavelength decreases, until it becomes more likely to interact with a nucleon-sized object than with a nucleus-sized object ($\lambda \approx 1$ fm) [1].

- **Compound Nucleus Reactions:** Take much longer ($\sim 10^{-16}$ to 10^{-18} s) as the statistical nature of the reaction has increased and the projectile interacts with the target nucleus with many intranuclear collisions. This increase in the statistical nature of the reaction causes the coupling between the incident and outgoing channels to decrease, leading to Bohr's Independence Hypothesis, which states that the "memory" of the incident channel is lost, i.e. that the probability of decay into a specific set of products is assumed to be independent of the formation of the compound nucleus [1].

- **Pre-Equilibrium Reactions:** A particle can also be emitted on intermediate time scales that do not fit into either of the two aforementioned categories, thus implying the existence of a mechanism incorporating both features of direct and compound nucleus reactions. As the projectile has time to interact with some of the target's nucleons, a complex system begins to form while the energy of the projectile is shared among neighboring nucleons. If during this process a particle gains enough energy to be emitted (before equilibrium is reached in the system), it then leads to a pre-equilibrium reaction [1-3].

From a projectile energy perspective, the compound nucleus mechanism is dominant in the lower energy range, while direct reactions predominate for higher projectile energies. Pre-equilibrium emission occurs in an intermediate range, typically above 10 MeV per nucleon. The cross section of each mechanism according to the reaction time and energy is summarized in Figure 1.4.

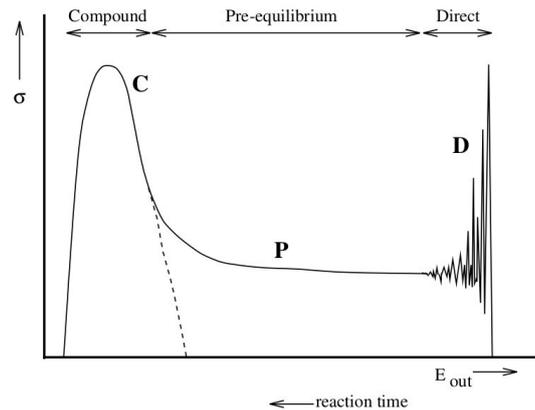


Figure 1.4: Schematic representation of reaction cross section and dominant mechanisms according to time and energy [6]

Formalism:

Nuclear reaction is the phenomenon of interaction between nuclei. This is a process which is accompanied by mass/energy exchange resulting in energy production or absorption depending of the masses of the involved nuclei. In its general form, it is described formalistically as follows:



or



where a is the projectile, X is the target nucleus, b is the ejected particle and Y is the residual nucleus.

Alternative classification:

We can classify reactions in many ways and not only by the mechanism that governs the process. When the incident and outgoing particles remain the same, it constitutes a **scattering process**. If Y and b are in their ground states, it's termed as "**elastic**", while if Y or b is in an excited state, it's referred to as "**inelastic**", often leading to rapid decay via gamma emission. Occasionally, both a and b are the same particle, but the reaction can eject another nucleon separately, resulting in three particles in the final state, termed as a "**knockout reaction**". Last but not least, in "**transfer reactions**", one or two nucleons are exchanged between the projectile and target. Note that inelastic scattering could occur either through a direct process or through a compound nucleus process, largely depending on the energy of the incident particle [1].

Another category of reactions is **capture reactions**, wherein the incoming projectile is assimilated by the target nucleus, leading to the creation of a distinct nucleus often in an excited state. Subsequently, this excited nucleus undergoes a process of de-excitation to transition to its ground state. Capture reactions are significant in understanding nuclear processes and they play a crucial role in elucidating various phenomena, including nuclear structure, astrophysical nucleosynthesis, and reactor design. Other types of reactions may include capture and stripping reactions, fission, (n, xn) , etc [1].

1.4 – Interaction of Particles with Matter

1.4.1 - Interaction of Photons with Matter:

There are three primary mechanisms through which photons interact with matter in radiation measurements: photoelectric absorption, Compton scattering and pair production. In each of these interactions, photons transfer all or a portion of their energy to electrons.

- **Photoelectric Absorption:** In the process of photoelectric absorption, a photon is captured by an atom and in its place an energetic photoelectron is ejected from the tightly bound shells of the atom. For γ -rays of sufficient energy, the most likely origin of the photoelectron is the most strongly bound shell (K-shell) of the atom. The photoelectron appears with energy given by:

$$E_e = h \cdot \nu - E_b \quad (7)$$

Where E_b represents the binding energy of the photoelectron in its original shell. In addition to the photoelectron, the interaction leaves behind an ionized atom with a hole in one of its bound shells. This gap is quickly filled by the capture of a free electron from the medium and/or by the electron rearrangement of the other shells of the atom which, therefore, leads to the emission of one or more characteristic X-rays [2].

- **Compton Scattering:** The interaction process takes place between the incident photon and an electron in the absorbing medium. The photon transfers part of its energy to the electron (assumed to be initially at rest), which is then denoted as the recoil electron. Because all scattering angles are possible, the energy transferred to the electron can vary from zero to a large fraction of the γ -ray energy. A schematic representation of the Compton scattering process, together with the formula used to calculate the energy of the scattered photon, is given in Figure 1.5 [2].

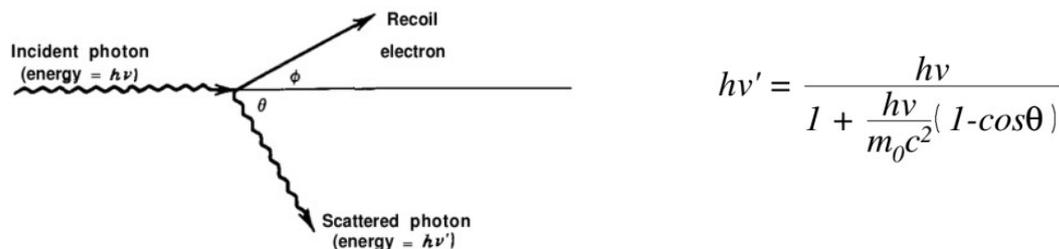


Figure 1.5: A schematic representation of the Compton scattering process (left) together with the formula giving the energy of the scattered photon (right)

where m_0c^2 is the rest energy of the electron (0.511 MeV)

- **Pair Production:** The process of pair production is energetically possible if the γ -ray energy exceeds twice the rest energy of an electron (1.02 MeV), however the probability of this interaction remains very low until the γ -ray energy reaches above several MeV. Therefore pair production is mainly limited to high-energy γ -rays. It is a reaction that can only take place in the Coulomb field of a nucleus and leads to the creation of an electron-positron pair in place of the original photon. All excess energy carried by the photon, above the threshold of 1.02 MeV required to create the pair, is converted into kinetic energy which is shared by the positron and the electron. Very soon, the positron will interact with an electron in the absorbing medium and annihilate, leaving behind two photons (Figure 1.6) [2-4].

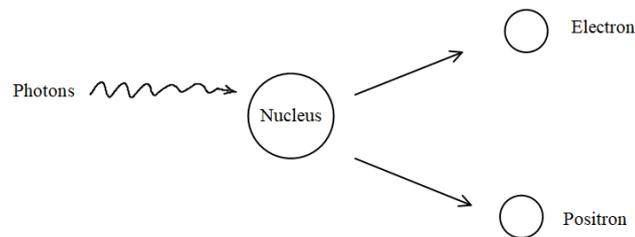


Figure 1.6: A schematic representation of the pair production process

It should be noted that Rayleigh scattering would normally be added to the above mechanisms of the interaction of photons with matter. The Rayleigh scattering is defined as the type of scattering where despite the fact that the direction of the photon changes, its energy remains the same. But because it is a phenomenon that mainly refers to the propagation of low energy γ radiation (a few hundreds of keV), so it will not be considered.

In our upcoming discussion on scintillators and their properties, we will observe that Compton scattering predominates.

The energy range over which each mechanism dominates is given in Figure 1.7

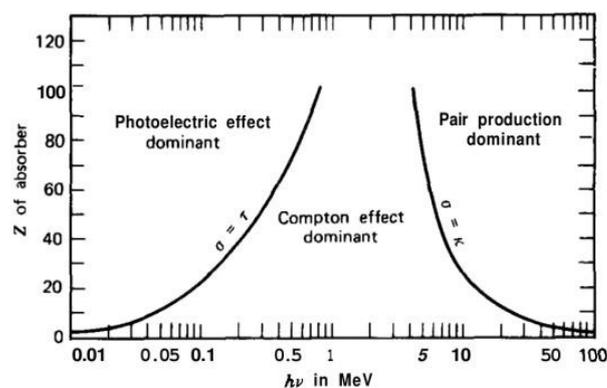


Figure 1.7: The dominant photon interaction mechanisms in relation to their energy

1.4.2 - Interaction of Neutrons with Matter:

Neutrons lack charge, thus they don't interact with matter via the Coulomb force, which mainly affects energy loss in charged particles. Consequently, they can traverse several centimeters of material without interaction, rendering them invisible to conventional detectors. When they do interact, it's typically with nuclei in the medium, leading to scenarios where they either are absorbed completely, giving rise to secondary radiations, or undergo substantial alterations in energy and direction [2].

For low-energy neutrons, known as thermal neutrons, neutron capture is the most probable reaction. As the energy of the incident neutron increases, the possibility of scattering and neutron-induced reactions such as (n, α) , (n, xn) , etc., also increases [2]. Slow neutrons primarily interact via elastic scattering with absorber nuclei and a plethora of neutron-induced reactions. In regards to their low kinetic energy, elastic scattering transfers minimal energy to the nucleus, rendering it inadequate for slow neutron detection. However, elastic collisions often lead to thermal equilibrium with the absorber medium. Most interactions with slow neutrons occur when they are in their thermal state, with an average energy of ~ 0.025 eV, at room temperature [2].

Neutron-induced reactions are crucial, generating secondary radiations detectable directly. These reactions necessitate a positive Q-value due to the low energy of the incoming neutrons. Generally speaking, while radiative capture reactions are common and important for neutron shielding, they are not extensively utilized in active neutron detectors due to challenges in detecting gamma rays. Reactions such as (n, α) , (n, p) , and $(n, \text{fission})$ are preferred due to their secondary radiations consisting of charged particles, making them more suitable for detection.

As neutron energy increases, the likelihood of neutron-induced reactions useful in detectors decreases rapidly. However, scattering becomes more significant because neutrons can transfer substantial energy in a single collision. In this scope, recoil nuclei, carrying detectable energy from neutron collisions, result from scattering. With each scattering event, neutrons are losing energy, being moderated or slowed down, especially effectively by hydrogen. Inelastic scattering with nuclei at high neutron energies can lead to recoil nuclei being elevated to excited states, emitting gamma rays upon de-excitation. This process is crucial for shielding high-energy neutrons but complicates the response of fast neutron detectors based on elastic scattering [2].

The cross-sections governing neutron interactions differ significantly from those of photons. They not only fluctuate with the energy of the incoming neutron but also exhibit a variance across different elements and even among isotopes of the same element. Describing neutron-nucleus interactions involves intricate and complex interactions between all nucleons within the nucleus and the incoming neutron. Consequently, fundamental theories capable of precisely predicting neutron cross-section variations are still elusive. Hence, all cross-sectional data are essentially empirical, derived from experimental observations [19].

As mentioned above, a simple way of detecting a neutron requires transferring its energy to a charged particle, which can be achieved through various methods. A straightforward approach is elastic scattering on light nuclei. Light nuclei like hydrogen offer the advantage of absorbing the entire neutron energy in one go, whereas heavier nuclei can only absorb a small portion of it.

Based on the aforementioned neutron interaction with matter principles, we are going to describe below three predominant mechanisms of neutron interactions with the absorber nuclei.

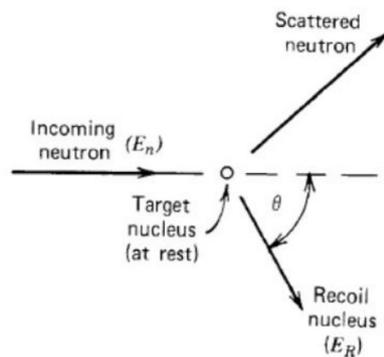
- **Elastic Scattering:**

Neutron elastic scattering is the process in which a neutron collides with a nucleus, leading to the recoil of the nucleus, while preserving the total kinetic energy and momentum before and after scattering thus enabling neutron detection based on the energy of recoiled nuclei [27].

Elastic scattering is the main mechanism for slowing down and consequently detecting fast neutrons (energies $\sim 1\text{-}20$ MeV) using low-Z materials. Specifically, materials like hydrogen, deuterium or helium, where the entire neutron energy can be transferred to the target nucleus, resulting in recoil protons. The recoil protons will traverse a small distance within the absorbing medium, transferring their kinetic energy to the absorber via Coulomb interactions with the nuclei and orbital electrons.

However, distinguishing fast neutrons from background radiation, such as gamma rays, becomes increasingly challenging as the energy of the incoming neutron decreases. Special detectors utilizing pulse shape discrimination techniques could possibly allow the identification of sub-MeV neutrons, but it is extremely difficult. The mechanics of elastic collisions between nucleons and nuclei are governed by two-body mechanics, described in both laboratory and center-of-mass systems. This understanding forms the basis for interpreting neutron interactions and designing effective neutron detection systems [27].

For a neutron with mass m_n and initial kinetic energy $(E_K)_i$, the kinetic energy ΔE_K transferred to the nucleus of mass M that recoils at an angle θ with respect to the neutron initial direction of motion, is given as (Figure 1.8):



$$E_R = E_n \frac{4 m_n M}{(m_n + M)^2} \cos^2 \theta$$

Figure 1.8: A schematic representation of neutron **elastic scattering** (left) together with the formula giving the energy of the recoiled nucleus (right)

- **Inelastic Scattering:**

In the case of inelastic scattering, the neutron is initially captured by the nucleus and then re-emitted with lower energy and in a different direction from its original path. Meanwhile, the nucleus is left in an excited state, which it subsequently de-excites from by emitting high-energy gamma rays. This process is exemplified by the following relationship (Eq. 8).



where,

${}^A_Z X$ the stable target nucleus,

${}^{A+1}_Z X^*$ an unstable compound nucleus and

${}^A_Z X^*$ an excited target nucleus.

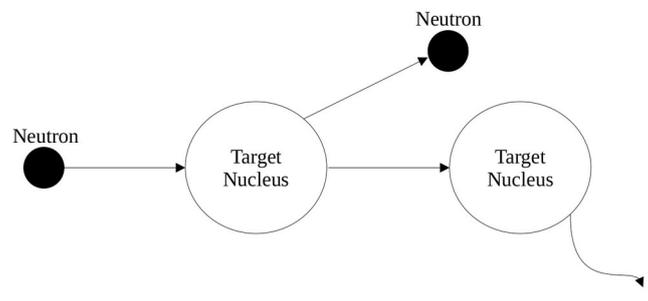


Figure 1.9: A schematic representation of neutron *inelastic scattering*

Inelastic scattering is the main mechanism for slowing down fast neutrons using high-Z materials, with an energy threshold close to 5 MeV.

- **Neutron radiative capture:**

When a neutron collides with a target nucleus, it can be absorbed, forming a compound nucleus that may become excited and emit gamma radiation. This reaction is quite prevalent, especially since thermal neutrons have the ability to induce this process in almost all nuclides. Typically, the excitation energy of the target nucleus is released in the form of one or several photons with each neutron capture event leading to the emission of energy ranging from about 6 to 10 MeV [28].

Neutron capture is an exoergic interaction in which Q is positive. The lack of dependence on the momentum of the emitted γ -ray and on the angle of emission, indicates that it is mainly isotropic at low energies. In the context of nuclear astrophysics, neutron radiative capture plays a crucial role in several important astrophysical phenomena with one notable example being nucleosynthesis (the process by which elements are synthesized in stellar environments) [28].

During nucleosynthesis, neutrons can be captured by nuclei to form heavier elements through series of reactions known as the s-process (slow neutron capture) and the r-process (rapid neutron capture), as discussed in Chap. 1.2.

Overall, neutron radiative capture plays a fundamental role in shaping the elemental abundance distribution observed in the universe, providing insights into the astrophysical processes occurring in different stellar environments.

To establish the quantitative details of these processes, precise energy-averaged neutron-capture cross sections at about 30 keV are vital for determining a detailed view of the mechanisms, durations, and the temperatures involved. This information also aids in identifying neutron sources, necessary flux levels, and potential locations for these processes to occur.

Neutron Capture Cross Sections:

Neutrons generated within stellar interiors rapidly become thermalized via elastic scattering, a process that takes approximately 10^{-11} seconds. Following this thermalization, their velocities adhere to a Maxwell-Boltzmann distribution. Given this context, the expected energy dependence of the neutron-capture cross section follows the form:

$$\sigma_{ny} \propto 1/v \propto 1/E^{1/2} \quad (9)$$

The most probable energy for the process to occur is around $E_0 = kT$ (see Fig. 1.10). Correspondingly, the most probable thermal velocity is $v_T = (2kT/m)^{1/2}$ where m represents the reduced mass. Neutrons participating in the s-process are typically produced during the helium-burning phase in red giants. During these reactions, the temperatures range from $T = 0.1 - 0.6 \times 10^9$ K, resulting in $E_0 = 30$ keV, thus obtaining [19]:

$$\sigma v = \text{const.} = \sigma_T v_T \quad (10)$$

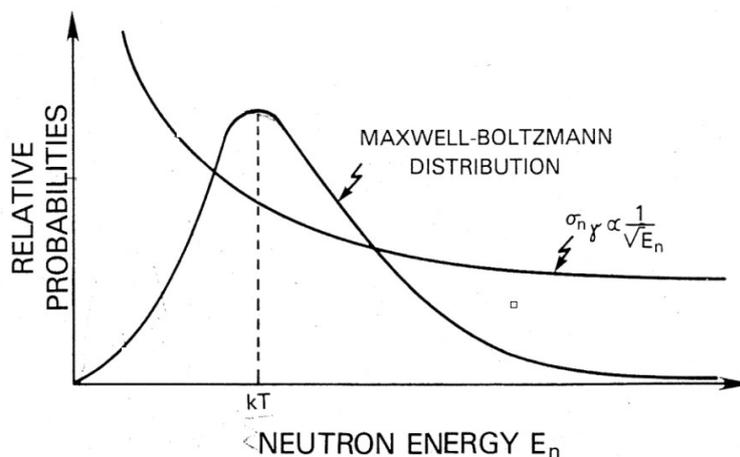


Figure 1.10: A schematic diagram of the Maxwell-Boltzmann energy distribution and the expected energy dependence of the neutron-capture cross section [19]

The reaction rate per particle pair in constant:

$$\langle \sigma v \rangle = \text{const.} = \langle \sigma \rangle v_T \quad (11)$$

In this context, an averaged cross section, $\langle \sigma \rangle$, is defined so that its product with the thermal velocity, v_T , yields $\langle \sigma v \rangle$. Generally, $\langle \sigma \rangle$ approximates the cross section at $v = v_T$, often making $\langle \sigma \rangle = \sigma_T$. For most cross-section dependencies, such as $\sigma \propto \text{const.}$ or $\sigma \propto 1/v$ this holds nearly true, with $\langle \sigma \rangle = 2\sigma_T/\sqrt{\pi}$. Therefore, measurements near this value provide a good estimate of $\langle \sigma \rangle$. Despite some uncertainties in the actual stellar temperatures, Maxwellian-averaged capture cross sections, $\langle \sigma \rangle$, remain relatively stable for most nuclides between 10 and 100 keV. Consequently, it is practical to standardize measurements at 30 keV, extrapolating nearby data to this energy with minimal uncertainty.

In cases where the capture cross section is influenced by a few narrow resonances or spans a broader temperature range (e.g., in the r-process), the cross section must be measured over a wider energy range and numerically integrated with the Maxwellian distribution [19]:

$$\langle \sigma \rangle = \frac{\langle \sigma v \rangle}{v_T} = \frac{2}{\sqrt{\pi}} \frac{1}{(kT)^2} \int_0^{\infty} \sigma(E) \cdot E \cdot e^{-\frac{E}{kT}} dE \quad (12)$$

In the neutron energy range of importance for the s-process nucleosynthesis ($E_n \simeq 1 - 300 \text{ keV}$), capture cross sections can be measured utilizing several different methods and a variety of neutron sources.

1.5 – Scintillation Detector Principles

General Description:

Scintillation is an age-old technique that remains essential in detecting and studying various radiations due to its ability of converting charged particle energy into detectable light. However, finding the appropriate scintillator material is difficult as it necessitates a balance between multiple properties. While an ideal material should boast high scintillation efficiency, linearity in light yield, and a short decay time, no single material meets all these criteria. Consequently, the selection of scintillator materials often involves compromises based on specific application requirements and inherent material characteristics. Commonly used scintillator materials range from inorganic alkali halide crystals like sodium iodide to organic-based liquids, solids, crystals and plastics, each offering unique advantages and drawbacks [3].

In scintillation processes, fluorescence emerges as a key mechanism, promptly emitting light upon excitation. Other processes such as phosphorescence and delayed fluorescence also play a role in the emission of visible light. Pulse mode operation of scintillators predominantly relies on prompt fluorescence, facilitating precise measurements with rapid time constants. Conversely, current mode operation yields a steady-state signal current proportional to the total light yield but may underestimate light yield compared to pulse mode. Moreover, memory effects, or "afterglow," could arise in scintillation detectors under swiftly changing radiation intensities, specifically if long-lived decay components are prominent. Thus, understanding the complexities of scintillator behavior and properties becomes imperative for effective radiation detection applications.

Despite the challenges and complexities associated with scintillation, a complete understanding of its behavior and properties is crucial for practical applications in radiation detection. Numerous materials and mechanisms contribute to the scintillation process, each with its unique advantages and limitations. By closely examining these trade-offs we can develop solutions for a wide array of radiation detection and spectroscopy needs, spanning from fundamental research in nuclear physics to practical applications in medical imaging [3].

1.5.1 - Scintillation Mechanism in Organics:

Fluorescence in organic materials originates from transitions within the energy levels of individual molecules, enabling fluorescence regardless of the material's physical state. In contrast to crystalline inorganic scintillators, which depend on a regular lattice structure, organic scintillators rely on molecules possessing specific symmetry properties, giving rise to what's termed as a π -electron structure (Fig. 1.11).

In organic scintillator molecules, energy can be absorbed by exciting the electron configuration into any one of a number of excited states. We can distinguish between singlet (S_0, S_1, S_2, \dots) and triplet states (T_1, T_2, T_3, \dots). The energy difference between the ground state (S_0) and the first excited state (S_1) usually falls within the range of 3 - 4 (eV), while the energy gaps between higher energy states tend to be smaller [3]. These electronic configurations are then further divided into more refined levels, which correspond to various vibrational states of the molecule, typically with a spacing of around 0.15 eV. In order to assort these vibrational states, a second subscript is often introduced, with " S_{00} " denoting the lowest vibrational state of the ground electronic state. Since the energy difference between the ground state and the lowest

vibrational state is much higher than the average thermal energy of around 0.025 eV, nearly all electrons will be in the S_{00} -state [3].

When a molecule absorbs energy, as indicated by upward arrows, likewise in the case of a scintillator, it signifies the absorption of kinetic energy from nearby charged particles. The higher singlet electronic states excited during this process swiftly transition to the S_1 electron state through radiation-less internal conversion, typically occurring within picoseconds. Moreover, states with surplus vibrational energy, such as S_{11} or S_{12} , do not remain in thermal equilibrium with adjacent states and promptly release this excess energy. Consequently, in a simple organic crystal, the excitation process promptly generates a group of excited molecules in the S_{10} state after a very short period.

In organic scintillators, the primary scintillation light, also known as prompt fluorescence, occurs in transitions between the S_1 state and one of the vibrational states of the ground electronic state, as depicted by the downward arrows in Figure 1.11. If τ_1 denotes the fluorescence decay time for the S_1 level, then the intensity of prompt fluorescence at a given time t after excitation can be determined accordingly. Typically, in most organic scintillators, the decay time τ is a few nanoseconds, resulting in a relatively rapid prompt scintillation component (Eq. 13).

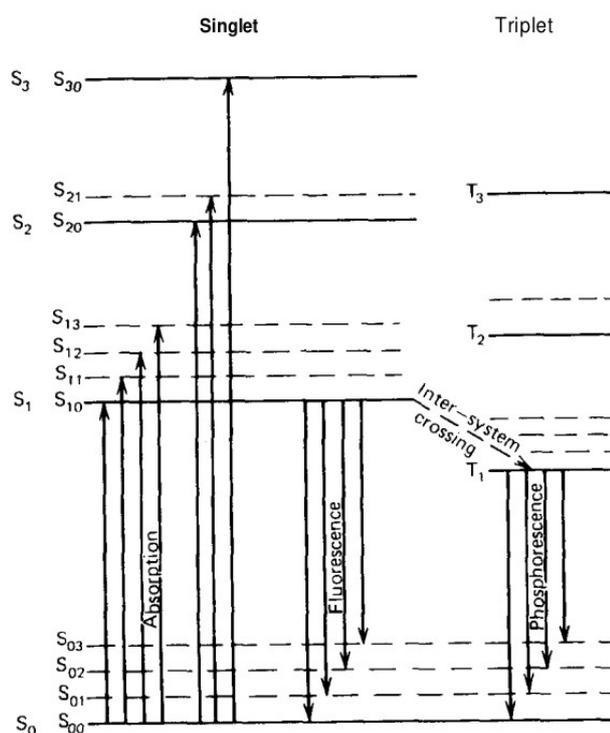


Figure 1.11: Transition scheme for the excitations and de-excitations in an organic scintillator [29].

$$I = I_0 \cdot e^{-t/\tau} \quad (13)$$

The first triplet state T_1 has characteristically much longer lifetime than the singlet state S_1 . Through a process called inter-system crossing, certain excited singlet states can be converted into triplet states with the lifetime of T_1 being as long as 10^{-3} sec., and the radiation emitted during the transition from T_1 to the ground singlet state (S_0) termed as phosphorescence, characterized by delayed light emission.

Figure 1.11 also offers insight into why organic scintillators allow their own fluorescence emission to pass through without absorption. The length of the upward arrows in the figure indicates photon energies that are absorbed within the material. Because fluorescence transitions represented by downward arrows, excluding S_{10} to S_{00} , possess lower energy than the threshold for excitation, there exists minimal overlap between the optical absorption and emission spectra, thus leading to negligible self-absorption of fluorescence. This is known as the Stokes shift and is exemplified by the spectra of a typical organic scintillator in Figure 1.12.

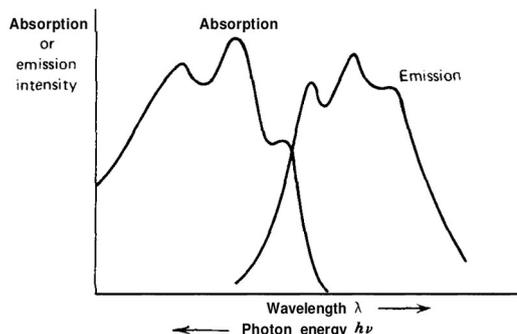


Figure 1.12: The optical absorption and emission spectra for a typical organic scintillator with the level structure shown in Fig. 1.6. [3].

Scintillation Efficiency and Quenching:

Scintillation efficiency in a scintillator denotes the ratio of incident particle energy converted into visible light. While it's preferable for this efficiency to be high, there exist alternative de-excitation pathways for excited molecules, which don't involve light emission but instead degrade excitation into heat. These processes of radiation-less de-excitation, are known as quenching and can hinder the light output. Hence, in the production and application of organic scintillators, it's vital to remove impurities, such as dissolved oxygen in liquid scintillators, to prevent quenching mechanisms and uphold optimal light output.

1.5.2 - Types of Organic Scintillators:

We will focus mainly on discussing the types of scintillator detectors relevant to this work, highlighting their unique characteristics and properties, while also addressing potential weaknesses. Therefore, the following sub-chapters will primarily cover organic liquids and, to a large extent, crystals with a specific emphasis on stilbene.

Pure Organic Crystals:

Among organic crystalline scintillators, anthracene and stilbene stand out as the most commonly used, though both materials pose challenges due to their fragility and limited availability in large sizes. Trans-stilbene (diphenyl-ethylene, $C_{14}H_{12}$) is one of the most useful organic crystal scintillators due to the fact that it combines relatively high scintillation efficiency with a short scintillation decay time. Its density is 1.16 g/cm^3 and its melting point 124°C , with a self-absorption being less than anthracene. While anthracene boasts the highest scintillation efficiency, stilbene, with lower efficiency, is particularly suitable for pulse shape discrimination purposes (see Chap. 5), which is one of the reasons that it was chosen for this work.

One problem, that will be extensively discussed later, is that the scintillation efficiency of these materials is influenced by the orientation of the ionizing particle relative to the crystal axis, resulting in directional discrepancies of up to 20-30%. This discrepancy undermines the achievable energy resolution in these crystals when incident radiation creates tracks in diverse directions within the crystal, a phenomenon called *response anisotropy* [3, 29].

Liquid Organic Solutions:

Liquid scintillators, created by dissolving organic scintillators in solvents, offer practical solutions with customization options. They may include only the organic scintillator and solvent or incorporate additional components like wavelength shifters for tailored emission spectra. Liquid scintillators are preferred for large-volume detectors due to cost considerations. Their absence of solid structure makes them more resilient to radiation damage, as confirmed by measurements showing durability even at high exposures. Liquid scintillators are extensively used in radioactivity counting, ensuring nearly complete counting efficiency, specifically for low-level beta activity [3].

Despite the above benefits, a number of problems arise when dealing with liquid scintillators. Firstly, dissolved oxygen in liquids can reduce fluorescence efficiency, requiring sealed containers for oxygen purging. Additionally, they expand with increasing temperature usually requiring special solutions like external expansion bellows, so-called optical Dip-in windows and/or an expansion reservoir outside of the interaction volume. Lastly, there is always a finite chance of leakage of toxic and low flash point materials, like the commonly used Benzene-based type detectors (e.g. C₆H₆, C₆D₆, EJ301, etc.) [36].

1.5.3 - Response of Organic Scintillators:

Light Output

When a charged particle interacts with a scintillator, a portion of its kinetic energy is transformed into fluorescent energy, while the rest dissipates as heat. The scintillation efficiency, which indicates the fraction of energy converted, varies based on the particle type and its energy. Occasionally, the scintillation efficiency remains constant regardless of energy, resulting in a linear correlation between light yield and initial energy. Organic scintillators like anthracene, stilbene, and various liquid and plastic types demonstrate linear responses for electron energies above roughly 125 keV. However, the response to heavier charged particles like protons or alpha particles is consistently lower for equivalent energies and exhibits nonlinearity at much higher initial energies [3].

Around energies in the few hundred keV range, the light yield in response to protons is approximately one-tenth of the light yield generated by electrons with equivalent energy. Although this difference decreases at higher energies, the response to protons consistently falls below that to electrons. To standardize the measurement of light yield, a specific term, "MeV electron equivalent" (MeV_{ee}), is utilized, considering the varying light yield of organic materials based on the particle type. In this context, the concept of MeV_{ee} denotes the energy of a particle needed to produce 1 MeV_{ee} of light, set at 1 MeV for fast electrons but higher for heavy charged particles due to their lower light yield per unit energy [3].

Birks [29] proposed a relationship to describe how organic scintillators respond to charged particles, linking emitted fluorescent energy per unit path length dL/dx with the specific energy loss of the particle dE/dx . According to the following relationship, increased ionization density along the particle's path leads to quenching from damaged molecules, reducing scintillation efficiency. The equation assumes that the density of damaged molecules is proportional to ionization density and that a fraction of these molecules undergo quenching. Additionally, it assumes that, in the absence of quenching, light yield is proportional to energy loss:

$$\frac{dL}{dx} = S \frac{dE}{dx} \quad (14)$$

where S is the normal scintillation efficiency. By adding a probability for quenching, we get:

$$\frac{dL}{dx} = \frac{S \frac{dE}{dx}}{1 + kB \frac{dE}{dx}} \quad (15)$$

Where L is the light yield, S is the scintillation efficiency, dE/dx is the specific energy loss of the particle per path length, k is the probability of quenching, and B is a constant of proportionality linking the local density of ionized molecules at a point along the particle's path to the specific energy loss. kB is treated as an adjustable parameter to fit experimental data. The above equation is commonly referred to as Birks' formula. We can modify the formula according to the type of incident particle making semi-empirical ones using additional fitting parameters [3, 31].

At this point it must be addressed, that a study about the light output response function to different types of particles for this specific trans-stilbene detectors was out of the scope of this work. In order to advance the results, several existing studies were used to assess this problem [30-32]. In this regard, the following figure and equations below, provide the light output response function for Stilbene- d_{12} to deuterium-ions and alpha particles using a modified Birks fit.

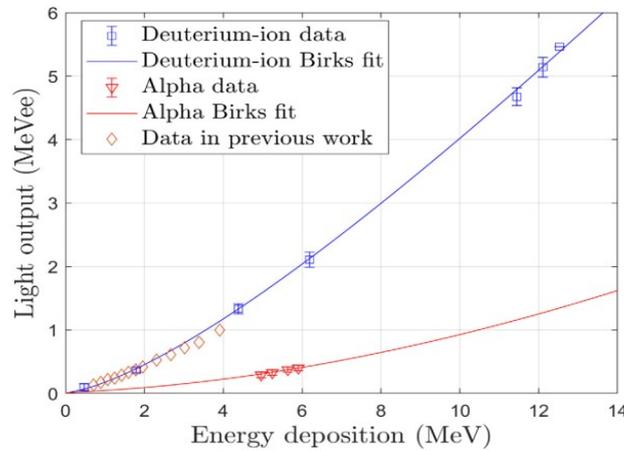


Figure 1.13: Stilbene- d_{12} light output responses to deuterium-ions and alpha particles.[30].

For deuterium-ions:

$$f(E) = -0.000583965 \cdot E^3 + 0.0252841 \cdot E^2 + 0.206746 \cdot E - 0.00303 \quad (16)$$

For alpha particles:

$$f(E) = 1.5783 \cdot 10^{-5} \cdot E^3 + 0.0051299 \cdot E^2 + 0.039266 \cdot E + 0.00742424 \quad (17)$$

The above equations were used in order to roughly be able to convert *energy*, of the incident particle, to light output units, i.e. *energy-electron-equivalent* units.

Response Anisotropy:

Scintillation anisotropy in single-crystal organics refers to the variation in the light emitted by heavy charged particles based on their trajectory relative to the crystal axes. For instance, a heavy charged particle of the same energy will generate differing amounts of scintillation light depending on whether its path aligns with the a-axis or b-axis of a stilbene crystal (Figure 1.14).

The crystal structure dictates the probability of singlet quenching and triplet–triplet annihilation, impacting the amount of prompt and delayed light emitted by the scintillator. Trajectories along directions of high probability for singlet or triplet transport can affect the amplitude of prompt or delayed components, respectively, influencing the recorded pulse in scintillator detectors [33].

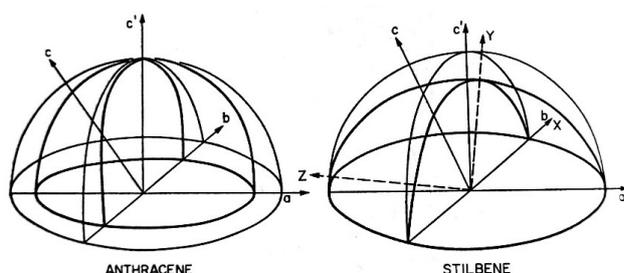


Figure 1.14: Scintillation response anisotropy of anthracene and stilbene crystals. Polar diagrams of relative scintillation response to alpha-particles for different directions of incidence relative to the crystal axes [29]

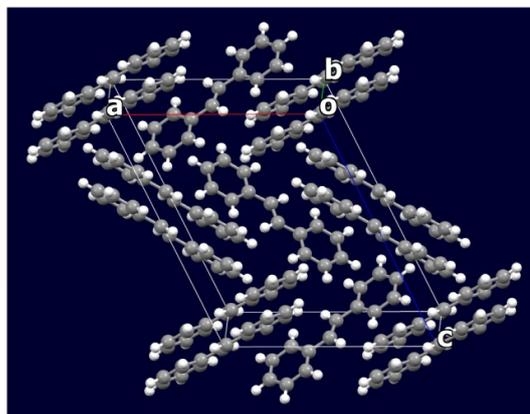


Figure 1.15: Stilbene unit cell with labeled crystal axes (o is the origin of the unit cell). This image was made with the Mercury CSD software package [34, 33].

Conclusively, in single-crystal organics with fixed lattice structures the transport of excited singlet and triplet states between molecules can lead to scintillation anisotropy.

Time Response:

The time profile of the light pulse in organic molecules typically exhibits a fast leading edge followed by a simple exponential decay if only prompt fluorescence is considered (Eq. 18). A more detailed model must also consider the finite time required to populate luminescent states and slower components like delayed fluorescence and phosphorescence. It takes approximately half a nanosecond to populate the levels responsible for prompt fluorescence light, and for very fast scintillators, the decay time from these levels is only slightly longer. Therefore, a thorough description of the pulse shape must include the finite rise time, assuming that the population of optical levels follows an exponential distribution [3].

$$I = I_0 (e^{-t/\tau} - e^{-t/\tau_1}) \quad (18)$$

where τ_1 is the time constant describing the population of the optical levels and τ is the time constant describing their decay. Other studies have shown that the population step is better represented by a Gaussian function $f(t)$ with a standard deviation σ . The overall light versus time profile is then described by:

$$\frac{I}{I_0} = f(t) e^{-t/\tau} \quad (19)$$

Chapter 2 - Experimental Set-Up & Tool-kits

- *The n_TOF facility & Neutron Capture Reactions*
- *Experimental Set-up*
- *Analysis Tools*

2.1 – The n_TOF facility & Neutron Capture Reactions

2.1.1 – General Overview:

The n_TOF facility:

The n_TOF facility at CERN is explicitly designed for investigating neutron-induced reactions, which hold significance across various research domains including stellar nucleosynthesis and nuclear technology applications. The facility operates by employing spallation reactions on a nitrogen-cooled lead target, induced by 20 GeV proton pulses from the CERN Proton Synchrotron (PS) accelerator. Each proton pulse carries a nominal intensity of $8.5 \cdot 10^{12}$ protons, with each collision generating approximately 300 neutrons per proton. The proton pulses have a maximum repetition rate of 0.8 Hz, with each pulse lasting 7 ns (rms). This setup enables excellent energy resolution of the resulting neutron beam, even in the GeV energy range. Consequently, the neutron spectrum produced spans a wide range, from meV to GeV. The facility utilizes the time-of-flight technique to precisely determine the kinetic energy of neutrons, a method from which the name "n_TOF" is derived [13].

n_TOF comprises two flight paths, leading to two experimental areas known as EAR1 and EAR2. The first flight path, nearly horizontal and 185m in length, directs neutrons to EAR1, while the second path, vertical and 20m long, leads to EAR2. These flight paths, depicted in Figure 2.1 along with the graphical representation of the top-down view, guide the neutron beams. Prior to reaching the experimental areas, the neutron beam undergoes collimation. Two types of collimators are utilized by n_TOF: capture collimators, approximately 2cm in diameter, and fission collimators, approximately 8cm in diameter, depending on the requirements of each experiment in flux.

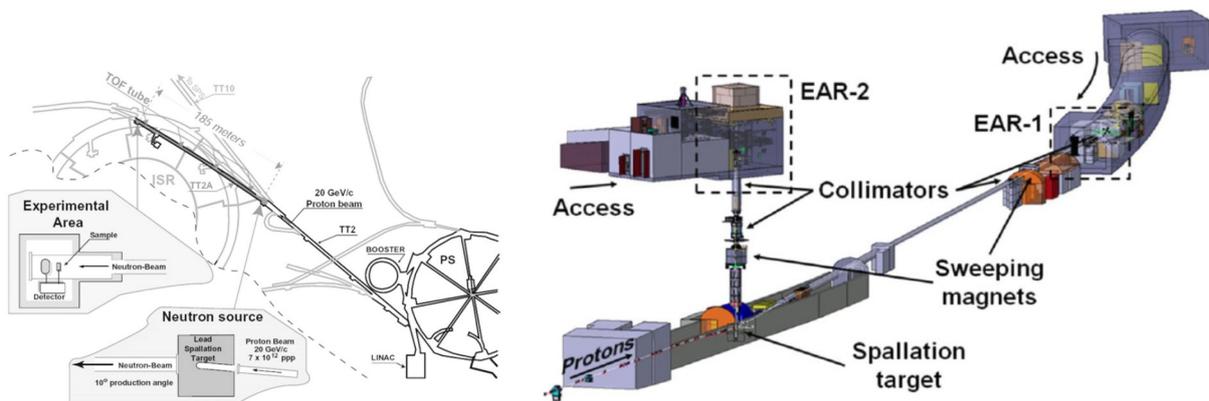


Figure 2.1: Neutron time-of-flight facility at CERN n_TOF, top-down view (left) [37]
The layout of the n_TOF facility at CERN related to EAR1 and EAR2 (right) [35].

2.1.2 – During LS2 (Long Shutdown 2) and Afterwards:

NEAR Station:

During LS2, a significant development at the n_TOF facility was the creation of a new experimental area called NEAR Station, situated approximately 3 meters away from the lead spallation target (hence the name [38, 39]). NEAR Station boasts a remarkably higher neutron flux compared to EAR1 and EAR2, making it ideal for irradiation and activation studies, particularly when dealing with radioactive samples that have mass limitations. In the activation zone of NEAR Station, nuclear astrophysics experiments are planned following the appropriate moderation and filtering of the neutron beam to achieve quasi-Maxwellian energy distributions. Additionally, NEAR Station includes a dedicated irradiation area where studies on material irradiation hardness have been conducted since 2022. For these studies, samples are placed in specially designed air-tight containers and installed near the spallation target, with handling performed by a robotic system due to the harsh radiation conditions. A schematic representation of the station can be seen below (Figure 2.2):

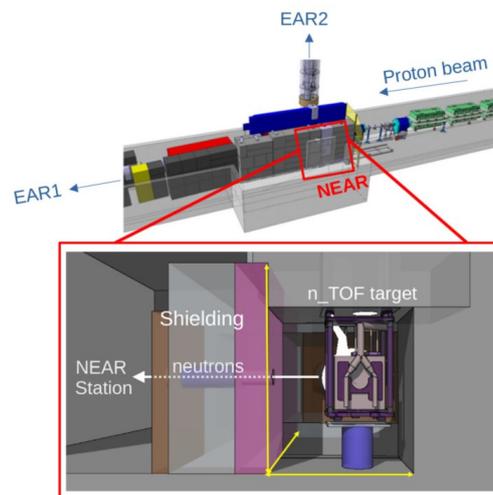


Figure 2.2: A schematic of the NEAR station [35].

EAR2 & Neutron Capture Reactions:

During LS2, the n_TOF teams not only enhanced the neutron beam characteristics but also developed innovative detection setups to enable new measurements and explore previously unexplored physics scenarios. One notable development is the i-TED setup [40], a gamma-ray detection system utilizing Compton imaging technology. This setup facilitates the identification and selection of emitted gamma rays from capture events within the sample volume, significantly improving the signal-to-background ratio and enabling measurements with minimal sample masses [41]. Additionally, to address challenges associated with neutron capture reaction studies using low-mass or radioactive samples, the high instantaneous flux of EAR2 proved invaluable.

However, this high flux also led to issues like high counting rates and strong pile-up events in the detection systems. To mitigate these challenges, small-volume segmented total-energy detectors (sTED) were implemented in a compact configuration around the capture sample. The high segmentation of these detectors allowed for a shorter distance between the sample and detector, resulting in a better signal-to-background ratio while maintaining manageable counting rates [35].

Liquid scintillation C_6D_6 detectors have been integral to neutron-capture cross-section measurements at CERN n_TOF since the initial experiments. However, optimizing these detectors has been crucial due to neutron sensitivity and background issues within the detector material itself, which affected data accuracy. Efforts focused on improving neutron sensitivity and safety features, such as encapsulations to prevent leaks and ensure stability during experiments. Despite these advancements, and as mentioned before, C_6D_6 remains a hazardous material due to its flammability, toxicity, and other risks. Hence, there's a proposal to explore trans-stilbene organic scintillators as potential replacements for C_6D_6 . Stilbene's chemical composition offers advantages like higher neutron sensitivity and density compared to C_6D_6 , making it safer and more efficient. Additionally, solid-state stilbene eliminates the need for a quartz-crystal window, reducing neutron sensitivity further [36].

Furthermore, the proposal suggests replacing bulky PMTs (PhotoMultiplier Tubes) with lightweight Silicon PhotoMultipliers (SiPM). This switch could reduce neutron sensitivity and enable more compact setups with lower voltage requirements.

At this point it should be noted that, the aforementioned description is based on the Letter of Intent (LoI) to the ISOLDE and Neutron Time-of-Flight Committee [36], in which the above information is stated. This work is focused, within the reasonable extend for a M.Sc. thesis, on fully characterizing two types of stilbene crystals (INRAD and PROTEUS) and the development and characterization of a multi-detector array (see next section) using the above crystals.

Total Absorption Calorimeters:

The goal of energy-differential TOF methods is to measure neutron capture cross sections over a broad neutron energy range, allowing for the calculation of Maxwellian averaged cross sections (MACS) for any relevant stellar temperature. Recent advancements in pulsed neutron sources and detection techniques have remarkably improved the accuracy of (n, γ) cross section measurements, often reducing uncertainties to just a few percent. This enhanced precision is crucial for accurately determining s-process abundances, which in turn helps infer the physical conditions of stellar environments by analyzing the abundance patterns of s-process branchings in solar material or presolar grains [5].

The energy sum of the γ -ray cascade emitted during the decay of a compound nucleus equals the binding energy of the captured neutron, making this neutron separation energy a clear indicator of a capture event. Thus, 4π detectors with near-100% efficiency are ideal for accurately identifying (n, γ) reactions and measuring capture cross sections. Initially, large liquid scintillator tanks were used for this calorimetric approach, but they have been replaced by arrays of small-volume organic scintillators aimed at exploiting the high instantaneous neutron-flux of EAR2 [5, 10].

2.2 – Experimental Set-Up

At this point, I ought to address the fact that this work serves as a crucial example of the longstanding collaboration between the University of Ioannina and Istituto Nazionale di Fisica Nucleare (INFN-CT, Sezione di Catania [44]), both deeply involved through countless contributions in the n_TOF collaboration at CERN. In this framework, the assistance of the above institutions was indispensable. The experiments, that will be thoroughly discussed and analyzed in the following chapters, were performed at the facilities of INFN-CT, with all the equipment used for the “on-line” measurements and analysis being generously provided.

2.2.1 – Crystal Characteristics & p-Stil modules:

In this work, three INRAD, developed at INFN-CT [47], produced using a proprietary low temperature solution growth technology, and one PROTEUS [48, 49] trans-stilbene ($C_{14}H_{12}$) crystals were used (Figure 2.4). Both crystals are solid, light-weight, non-hygroscopic, non flammable and non hazardous. Their melting point is 124 °C, with a density of 1.15 g/cm³ (INRAD). Moreover, the same dimensions of Ø1”x1” were used, in order to properly draw a comparison between their characteristics.

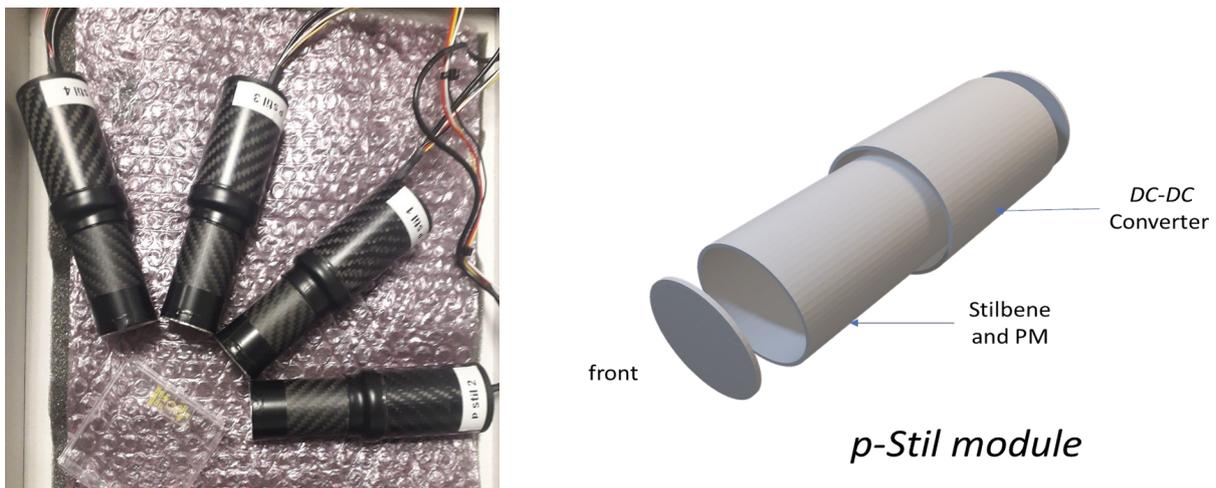


Figure 2.3: A picture of the four trans-stilbene modules, labeled as “p-Stil #” (left). A schematic of the module (right).

The crystals were assembled in a cylindrical carbon fiber housing, with a thin aluminium cover in the front window, as shown in the above schematic. They were then coupled with a R7378A–Hamamatsu PMT (Photomultiplier Tube), with the power base provided by a PS1807-Sens-Tech [51], Active base DC-DC converter. This ultra compact power base was tested for a self-powering device and a high counting rate with no HV (High Voltage) needed.

The aforementioned detector modules were thoroughly characterized and compared using various techniques, which will be extensively analyzed in the following chapters. This work aims to extrapolate useful data for future applications and serve as a foundation for innovation in the field of nuclear (astro-)physics. Through the use of the CAEN-DT5743, 8 Channel 12bit 3.2 GS/s Switched Capacitor Digitizer [52], every aspect of the analysis was based on the “off-line” capability of the digitized pulses recorded. As a final note on the topic, the emphasis on the significance of digitization will be also a key-point to extract from this work.

Rise/Fall time measurements and Pulse-Height/Charge spectra were extracted from the digitized data that was collected and compared for the two types of crystals using off-line analysis scripts. Along with the use of different radioactive sources, the energy calibration of the detectors was also implemented. As we will later see, both performed similarly and as expected.

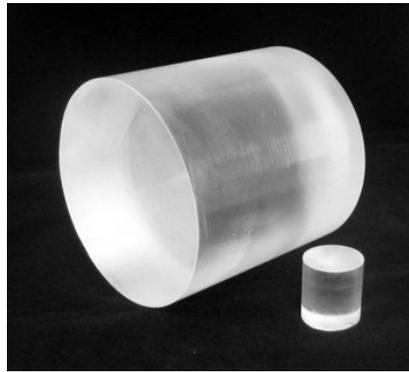


Figure 2.4: A picture of the trans-stilbene crystal [47]

2.2.2 – Multi-Detector Array:

Based on the idea of small-volume segmented total-energy detectors (sTED), described in the above chapter (Chap. 2.1.2), this work was aimed also at the development and characterization of an innovative fully symmetric multi-detector array (Fig. 2.5), using the aforementioned p-Stil modules. The focus is on developing a high-performance solid-state alternative for a segmented neutron capture setup, surpassing the capabilities of traditional C_6D_6 detectors. Even though new innovations with compact arrays of small C_6D_6 detectors have mitigated most problems related to low signal-to-background (S/B) ratio, (n, γ) efficiency, and counting rate capabilities, the issue of using highly chemically hazardous and flammable materials still persists.

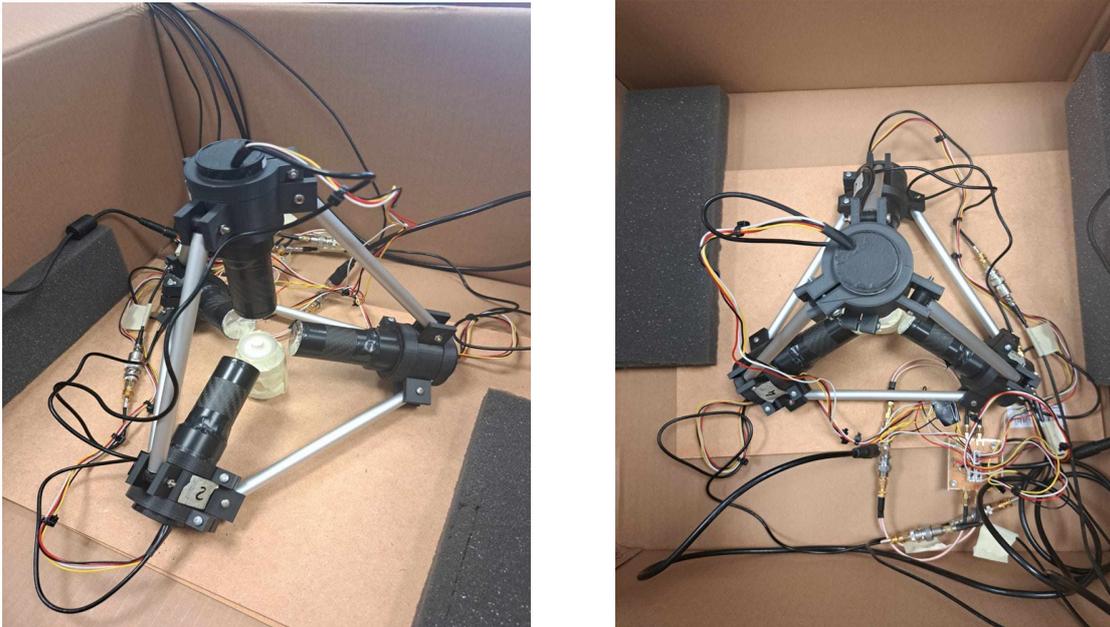


Figure 2.5: A picture of the multi-detector array. Side view (left). Top-down view (right)

As shown in the above Figure 2.5 the p-Stil modules are assembled in each corner of a regular tetrahedron with the active window of each module facing the centroid in which a radioactive source can be easily placed. Each module is also fixed in place using lightweight 3D-printed PLA holders, easy to produce and replicate. Thin hollow aluminum rods connect each corner, making it rigid and stable. At last, the circuitry involved is simple and reliable, with a few components. It consists of a power supply that can be adjusted to the full dynamics of the detector (if needed) and a voltage divider, for the four channels (can easily be extended to more).

Time resolution through time coincidences and γ/n Pulse Shape Discrimination (PSD) measurements were performed using this setup. The easy placement of any required radioactive source for each experiment, along with the full symmetry and a fast, reliable digitizer, made the array extremely efficient.

Moreover, future implications of the above apparatus can be focused on the characterization of neutron flux, enabling the usage of this set-up in neutron detection and imaging in space environment, as reliable, low-power, and adaptive detection systems become increasingly demanding. Notably, the introduction of mechanical adaptive features in the above set-up, will enable real-time optimization of the trade-off between efficiency and tracking angular resolution.

2.2.3 – GEANT4 Simulation:

Moreover, a complementary simulation, using the GEANT4 simulation toolkit mentioned below (Chap. 2.3.1), was performed in order to characterize the intrinsic and absolute detection efficiency of the stilbene crystals to gammas and neutrons. The Figure (Figure 2.6 - left) below shows a simple simulation of an isotropic “point-source” with an implemented randomization in the starting positions of the primaries (see Chap. 2.3.1), gammas and neutrons respectively, and a pure stilbene crystal placed 10cm away on the z axis.

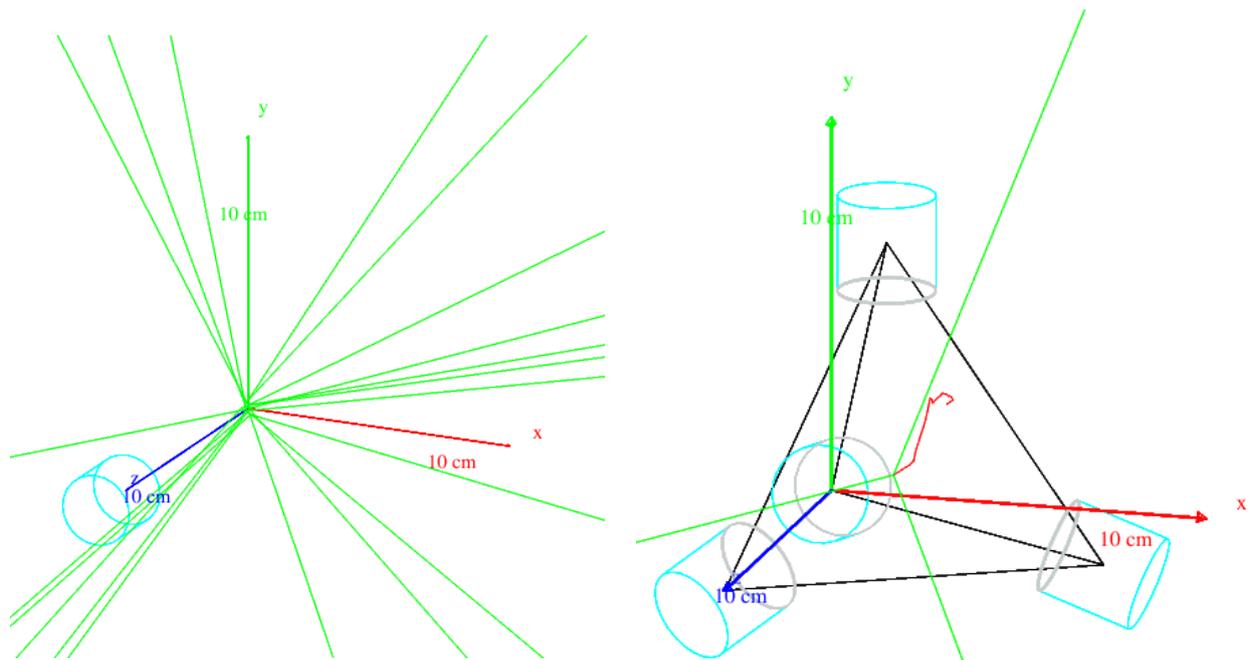


Figure 2.6: A visual representation of the stilbene crystal placed 10cm away of the point source (left).
The multi-detector array with source placed at the centroid (right)

The right side of Figure 2.6, which is the simulation of the tetrahedron with the four detectors placed at each corner (see Fig. 2.5 above), though not used in any way for this work, it serves as a proof-of-concept for the future implementations mentioned above. More on neutron/gamma detection efficiency calculations can be found on the respective Chapter below.

2.3 – Analysis Tools

2.3.1 – GEANT4:

GEANT4 [12] is a widely utilized simulation toolkit designed to simulate particle interactions with matter across various physical processes, including electromagnetic, hadronic, and optical phenomena. Employed in fields such as nuclear physics, high energy physics, astrophysics, and biology, GEANT4 incorporates extensive physical models and experimental data. Utilizing object-oriented technology and implemented in C++, it also offers visualization options [11].

To execute simulations with GEANT4, users define the geometry of their physical setup and specify primary particles. The toolkit constructs simulations using different volumes, each described by its shape and physical characteristics, arranged within a containing volume. This is achieved through concepts like "logical volume," "physical volume," and "solid." Solid objects in GEANT4 have specific shapes and dimensions, while logical volumes encompass both geometric properties and physical characteristics like material composition. Physical volumes represent instances of logical volumes placed within a containing volume, often referred to as the "mother volume," including coordinates and rotation details. The largest volume, containing all others, is called the "World." Rotating a mother volume also rotates its "daughter" volumes, as their placement details are relative to the mother volume. GEANT4 utilizes the Constructed Solid Geometry (CSG) technique to model primitive solids, offering primitives like Boxes, Tubes, Cones, Spheres, Wedges, and Toruses. Users can create complex solids by applying boolean operations (union, intersection, subtraction) to these primitives. A collection of such intricate shapes, made using boolean operations, can be seen in Figure 2.7.

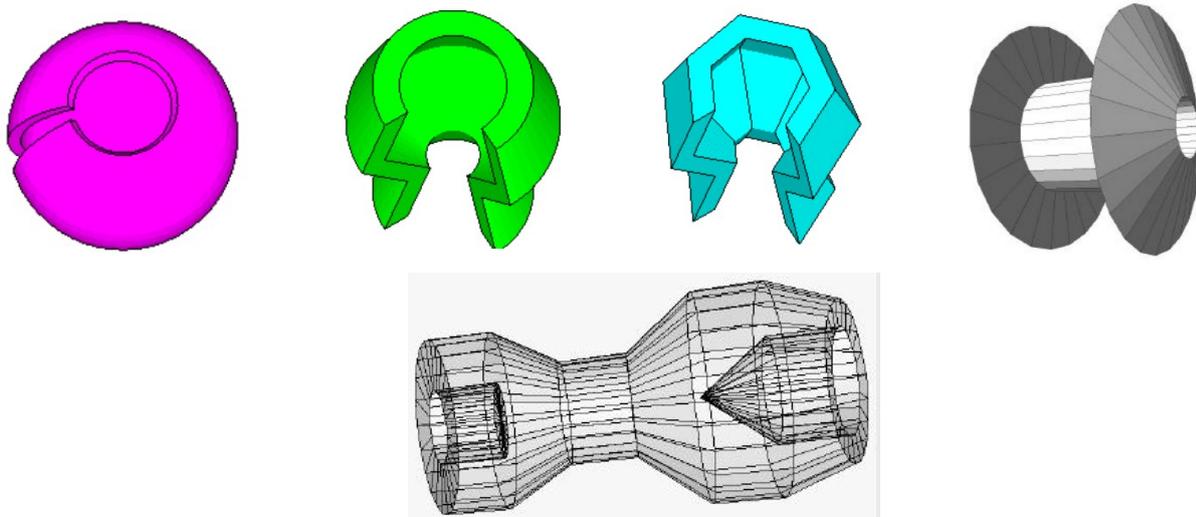


Figure 2.7: Examples of complex shapes that can be made using GEANT4

In GEANT4 simulations, users define primary particles' properties using classes like G4PrimaryGenerationAction [12]. The G4ParticleGun class is commonly used for primary particle generation, allowing users to specify particle type, position, energy, and momentum. While G4ParticleGun does not include built-in randomization, users can utilize C++ methods to create desired distributions.

Once generated, GEANT4 tracks primary particles until their kinetic energy drops below a threshold or they leave the simulation volume or decay into other particles. Information about particle tracks and interactions is stored within an "event," which can be written to an output file. A collection of events sharing the same configuration is referred to as a "Run".

To accurately calculate energy losses and reaction cross-sections, GEANT4 employs physics lists, offering a variety of options containing relevant physical processes. Users can choose the appropriate physics list to optimize simulation efficiency.

GEANT4 also provides visualization options, enabling users to graphically depict simulation data such as detector components, particle trajectories, and hits. Visualization parameters like color can be specified, and users can choose the type of output file and its contents [12].

In this work particularly, GEANT4 was only used in order to determine the simulated efficiency of the stilbene crystal. A simple simulation using two different physics lists was deployed in order to extent and “solidify” the characterization of the crystals. Further analysis and discussion on the topic can be seen on Chapter 3.

2.3.2 - ROOT:

ROOT [8] is an object-oriented framework designed primarily for analyzing High Energy Physics data but is versatile enough to address challenges across various scientific domains, including industry. Built on the C++ programming language, ROOT also integrates with other languages like Python. Its object-oriented architecture simplifies code complexity and allows for easy modifications and extensions through classes and inheritance.

One notable advantage of ROOT is its extensive library of existing code, enabling developers to utilize pre-tested functionalities for tasks such as data fitting, histogram creation, and graphing, thereby focusing more on their specific problems. ROOT offers a wide range of functions for data processing, statistical analysis, and visualization, including advanced graphics capabilities and support for various data formats like ROOT Trees [9].

In this project, ROOT was mainly used alongside GEANT4 for calculations and file management tasks involved in determining the simulated efficiency of the detectors. ROOT in this case, is the perfect candidate for managing “.root” file types, which contain crucial parameters such as event number, energy deposition, particle type, etc., extracted from the output of the GEANT4 simulation. When processing a substantial volume of events to ensure robust statistics for each energy region, it becomes essential to execute the analysis efficiently. Therefore, the implementation of file types using a "tree-like" structure was imperative for this purpose.

2.3.3 – Python:

Python [18] is a versatile and powerful programming language widely used for data analysis due to its rich ecosystem of libraries and tools tailored for this purpose. With libraries such as NumPy, Pandas, Matplotlib, and SciPy, Python provides strong capabilities for tasks ranging from data manipulation and cleaning to statistical analysis, visualization, and machine learning.

NumPy is fundamental for numerical computing in Python, offering support for efficient array operations and mathematical functions. Pandas builds upon NumPy, providing high-level data structures and functions for data manipulation and analysis, particularly with tabular data. Matplotlib offers comprehensive plotting functionalities for creating a wide range of static, interactive, and publication-quality visualizations. For statistical analysis, SciPy offers a variety of statistical functions and tests, along with optimization and integration routines. Additionally, the statsmodels library provides more advanced statistical modeling and hypothesis testing capabilities.

Moreover, Python's ecosystem includes powerful machine learning libraries such as Scikit-learn, TensorFlow, and PyTorch, enabling users to build and deploy predictive models for tasks like classification, regression, clustering, and more. Overall, Python's simplicity, readability, and extensive libraries make it an excellent choice for data analysis tasks across various domains.

Based on the above and as the work was reaching the frontier of machine learning, the choice of using Python for the vast majority of the analysis was made. Thus, by using existing and well-tested libraries within the Python environment, accelerated development and improved workflow was achieved [17].

Chapter 3 – Detector Characterization

- *Detection efficiency*
- *Pulse-height/Charge Spectra & Energy Calibration*
- *Rise/Fall time measurements*
- *γ - γ Time Coincidence measurements*

3.1 – Detection Efficiency

General Description:

Radiation detectors are designed to produce an output pulse for each quantum of radiation that interacts within their active volume. For charged particles like alpha or beta particles, ionization or excitation occurs almost immediately upon entering the detector. These particles quickly create enough ion pairs to generate a detectable pulse. Consequently, detectors can often achieve a 100% counting efficiency for charged particles, as they can detect every particle that enters their active volume.

Uncharged radiations such as gamma rays or neutrons, which are the focus point of this chapter, they must undergo a significant interaction within the detector to be detected. Since these radiations can travel considerable distances before interacting, the detectors typically have an efficiency of less than 100%. Therefore, accurately determining the detector's efficiency is crucial to relate the counted pulses to the actual number of neutrons or photons hitting the detector [3].

Concerning convenience, we can divide counting efficiencies into two categories: *absolute* and *intrinsic*. Absolute efficiencies can be denoted as:

$$\varepsilon_{\text{abs}} = \frac{\text{number of pulses recorded}}{\text{number of radiation quanta emitted by the source}} \quad (20)$$

and are dependent not only on detector properties but also on the counting geometry characteristics, mainly the distance from the source to the detector. The intrinsic efficiency on the other hand is denoted as:

$$\varepsilon_{\text{int}} = \frac{\text{number of pulses recorded}}{\text{number of radiation quanta incident on the detector}} \quad (21)$$

and no longer has a dependency on the solid angle subtended by the detector as an implicit factor. We can relate the above efficiencies when using an isotropic source via:

$$\varepsilon_{\text{int}} = \varepsilon_{\text{abs}} \cdot G \quad (22)$$

where $G = 4\pi/\Omega$ is the geometry factor and Ω is the solid angle from the position of the source. It is more practical to list intrinsic efficiency values rather than absolute efficiencies, as the former have a weaker geometric dependence. Intrinsic efficiency primarily relies on the detector material, the radiation energy, and the detector's thickness in the direction of incoming radiation. There is a minor dependence on the distance between the source and the detector, as this affects the average path length of the radiation through the detector [3-5].

The solid angle, Ω , is defined by an integral over the detector surface that faces the source, of the form:

$$\Omega = \int_A \frac{\cos \alpha}{r^2} dA \quad (23)$$

where r refers to the distance between the source and a surface element dA , and α is the angle between the normal to the surface element and the source direction. For the case of a point source located along the axis of a right circular cylindrical detector, Ω is given by:

$$\Omega = 2\pi \left(1 - \frac{d}{\sqrt{d^2 + a^2}} \right) \quad (24)$$

where d the SD (Source-Detector) distance and a the detector radius, as shown below:

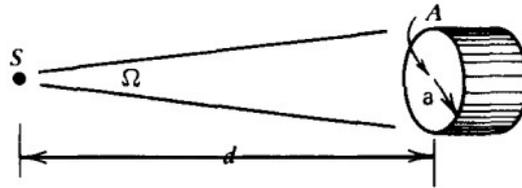


Figure 3.1: Schematic of the source-detector distance, the radius of the detector and the solid angle as seen from the point source [3]

As $d \gg a$, we can approximate the solid angle as the ratio of the detector plane frontal area A visible at the source to the square of the distance:

$$\Omega \simeq \frac{A}{d^2} = \frac{\pi a^2}{d^2} \quad (25)$$

In order to be certain that the above approximation is recreating the correct solid angle for the sample-detector distance of the simulation (at 10cm), a complementary Monte-Carlo simulation was performed using the SACALC5 [53] package. In that way we can use a comparative method to illustrate the performance of the approximation in range of distances. A point source 10cm away from the center of the detector window and the exact same geometrical characteristics as the GEANT4 simulation, were used. Following this discussion, the accumulative results can be seen below.

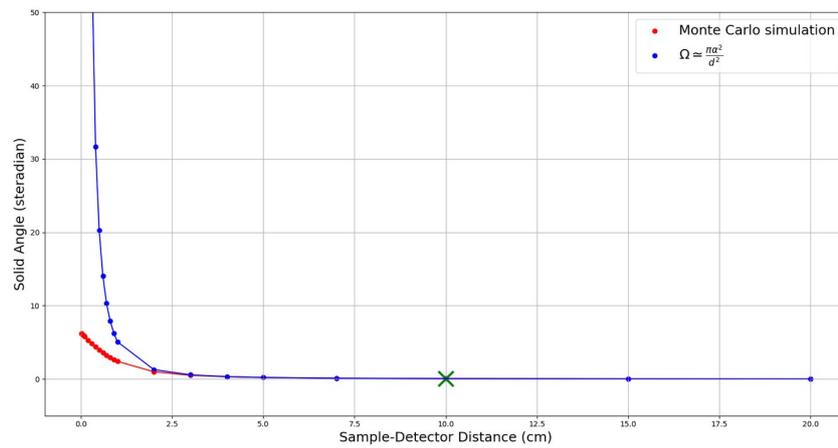


Figure 3.2: Figure showing solid angle calculations. With red the Monte-Carlo method and with blue the Eq. 25. “X” marked with green the SD distance used for the GEANT4 simulated efficiencies calculations.

As we can see in the above Figure, concerning the SD distance of interest (10cm), we can confidently say the the use of an approximation in the solid angle and hence in the geometry factor calculations, is fully acceptable.

Moving forward we can finally present the simulated efficiencies for trans-Stilbene, calculated for an energy range of 0.1 MeV to 10 MeV for both neutrons and gamma-rays. Two separate GEANT4 Physics Lists, the “Shielding” list and the “QGSP_BIC_ALLHP” list, were also comparatively evaluated. The results for the absolute and intrinsic efficiency concerning gamma rays can be seen below.

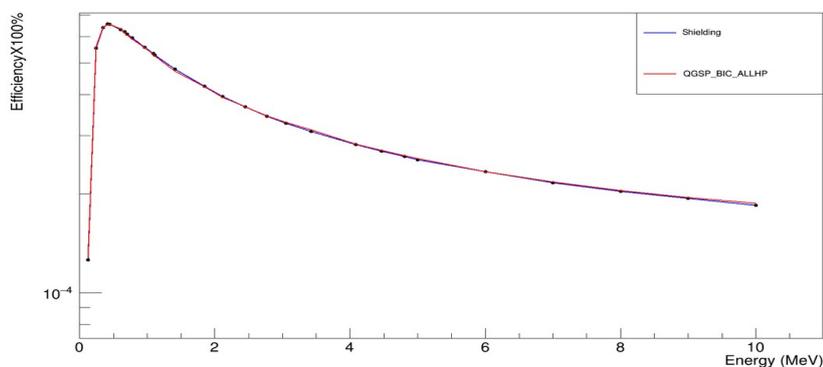


Figure 3.3: Simulated **gamma-ray absolute** efficiency. “Shielding” list (blue). “QGSP_BIC_ALLHP” list (red).

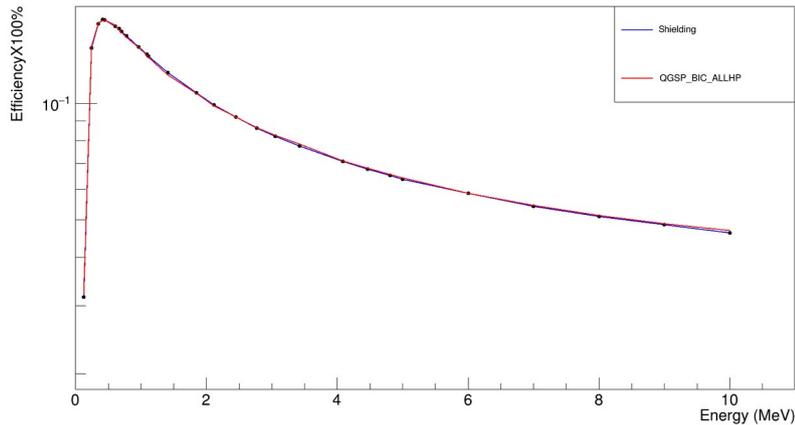


Figure 3.4: Simulated **gamma-ray intrinsic efficiency**. “Shielding” list (blue). “QGSP_BIC_ALLHP” list (red).

As for neutrons, we should first examine the available cross-section data provided by the National Nuclear Data Center (NNDC) [26]. Specifically, we will look at the total cross-sections for neutron interactions with hydrogen (^1H) and carbon (^{12}C), which are the components of stilbene.

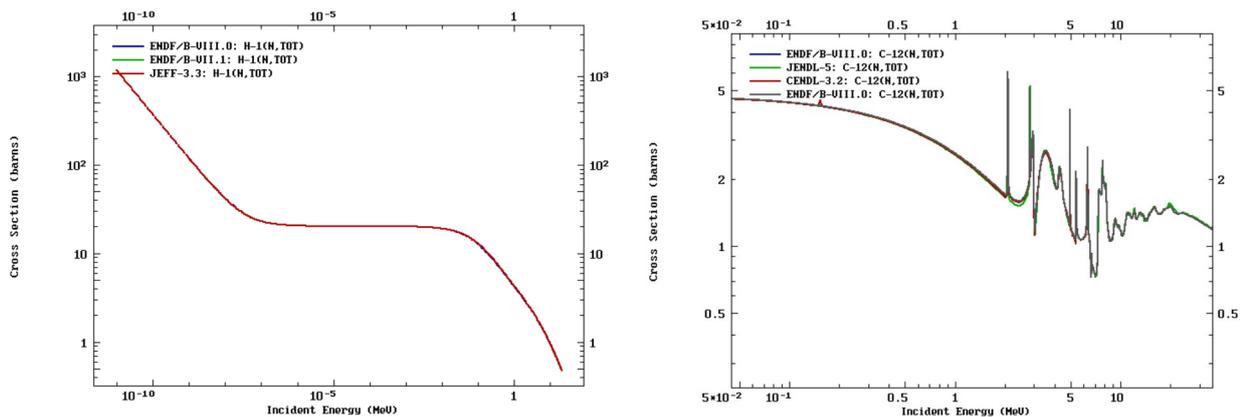


Figure 3.5: Neutron cross-section.
 $^1\text{H} (n, \text{tot})$ – left.
 $^{12}\text{C} (n, \text{tot})$ – right (zoomed in the resonance region) [26].

As we can see, while and reasonably so $^1\text{H}(n, \text{tot})$ is continuous through the whole energy range, $^{12}\text{C}(n, \text{tot})$ has resonances in the energy region of interest. Peaks that occur because the energy of the incoming neutron matches the energy levels of the compound nucleus formed during the interaction, leading to a temporary quasi-bound state. This results in increased probability of interaction, reflected as resonances in the cross-section data.

Therefore, we experience the effect of these peaks in the simulated neutron efficiency curves below.

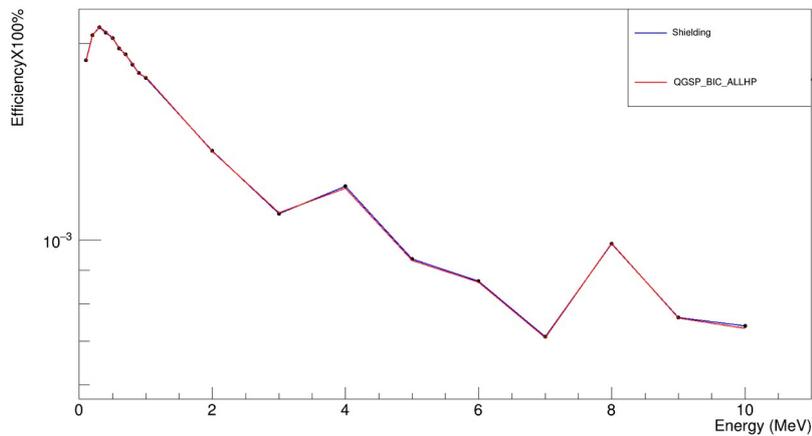


Figure 3.6: Simulated **neutron absolute** efficiency. “Shielding” list (blue). “QGSP_BIC_ALLHP” list (red).

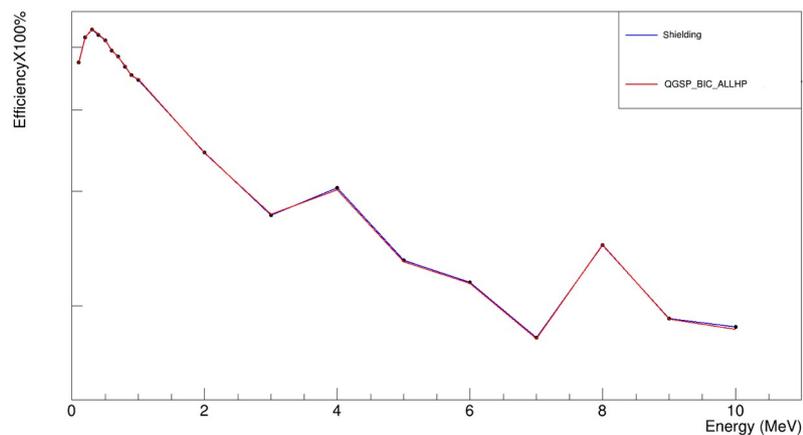


Figure 3.7: Simulated **neutron intrinsic** efficiency. “Shielding” list (blue). “QGSP_BIC_ALLHP” list (red).

To fully characterize a detector, it must be replicated in a GEANT4 simulation. The simulation helps calculate values like detector efficiency under various conditions and correction factors. The simulation needs to be based on the detector's specifications and be validated against experimental results. Future implementations need to build on that aspect.

3.2 – Pulse-height/Charge Spectra & Energy Calibration

When using a radiation detector in pulse mode, the amplitude of each individual pulse provides important information about the charge produced by that specific interaction within the detector. Analyzing a large number of these pulses reveals that their amplitudes vary. These variations can stem from differences in the radiation energy or from fluctuations in the detector's inherent response to monoenergetic radiation. The distribution of pulse amplitudes is a key characteristic of the detector's output and is regularly used to infer details about the incoming radiation or to assess the detector's performance.

Using two radioactive gamma-ray sources, ^{60}Co and ^{137}Cs , placed in front of the detector's window, while keeping the exact same experiment conditions for two crystals (same PM, DC-DC converter, e.t.c), the following pulse-height and charge spectra for each crystal (one INRAD and one PROTEUS) for each source were obtained:

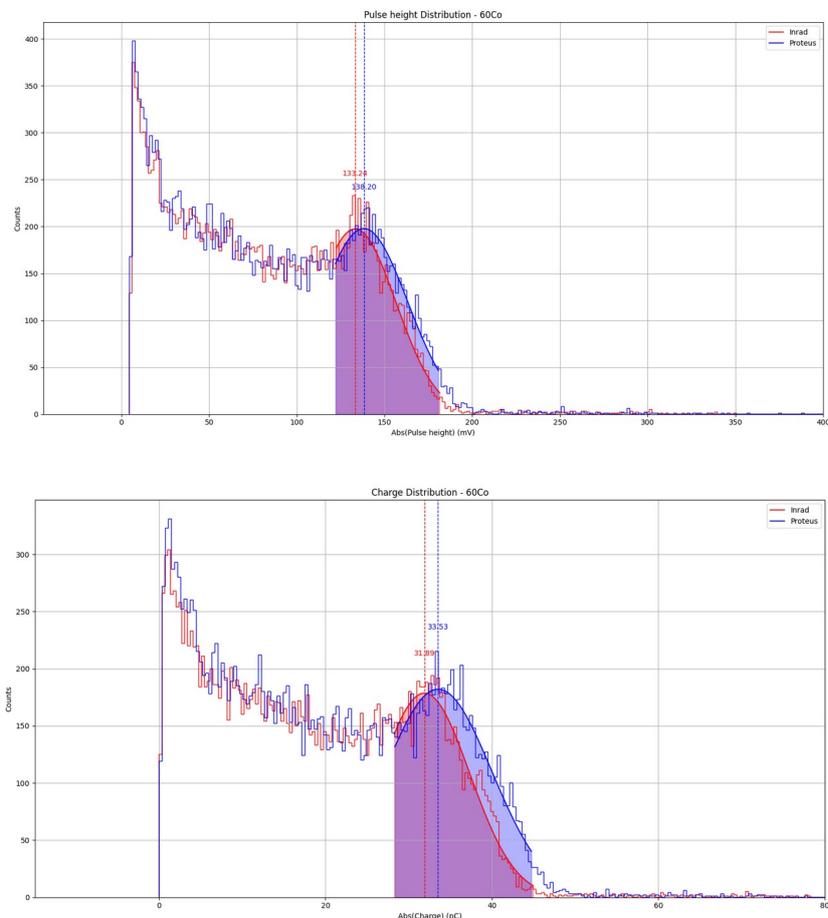


Figure 3.8: Pulse-height (top) and Charge (bottom) spectra obtained using ^{60}Co source. *INRAD* with red. *PROTEUS* with blue.

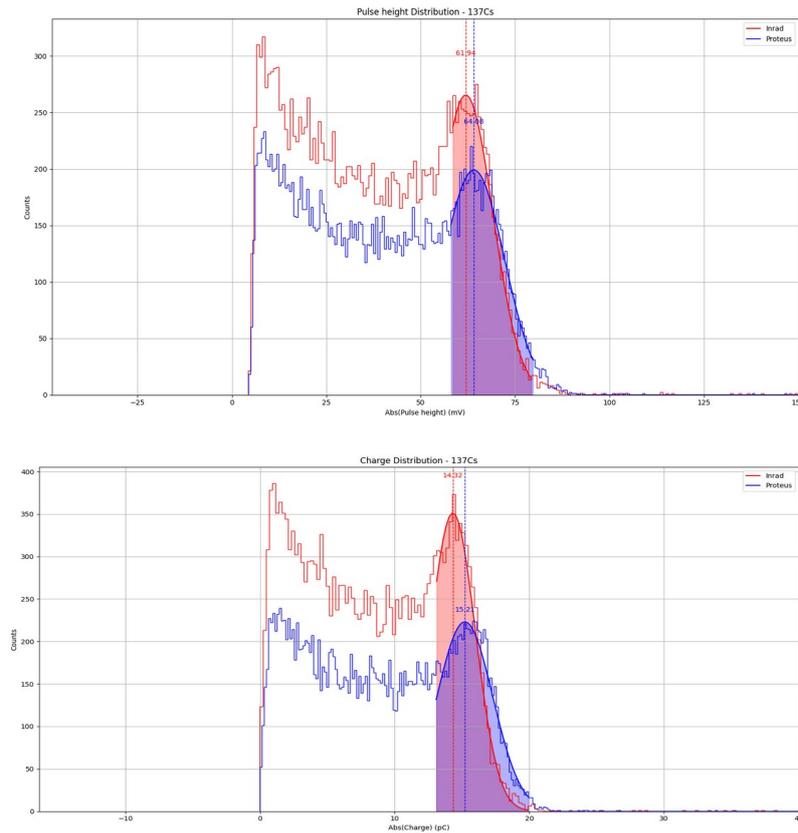


Figure 3.9: Pulse-height (top) and Charge (bottom) spectra obtained using ^{137}Cs source. *INRAD* with red. *PROTEUS* with blue.

As we know, organic scintillators, because of the low Z -value of their constituents (hydrogen, carbon, and oxygen), have a very small photoelectric cross section for gamma rays of typical energies. As a result, typical organic scintillators show no photopeak and will give rise only to a Compton continuum in their gamma-ray pulse height spectra. This can be easily seen in the above Figures, as neither of them shows a typical spectrum of either of the gamma-ray sources and Compton edges are the only distinguishable features. In the case of ^{60}Co , both Compton edges have been convoluted into a single spread-out peak. Because there are no photopeaks, some point on the Compton edge must be selected and associated with the maximum energy of a Compton recoil electron.

By locally fitting a Gaussian function to these peaks and by extracting their mean values, we can obtain a measure of comparison between the performance of the crystals, summarized in the Table below:

Sources	^{60}Co		^{137}Cs	
	Charge (pC)	Pulse-height (mV)	Charge (pC)	Pulse-height (mV)
INRAD	31.9 ± 0.2	133 ± 2	14.3 ± 0.1	61.9 ± 0.3
PROTEUS	33.5 ± 0.2	138 ± 1	15.2 ± 0.1	64.1 ± 0.3
Mean value	5.02 ± 0.04	3.76 ± 0.06	6.29 ± 0.06	3.55 ± 0.02
% Deviation				

Table 3.1: Mean values extracted by locally fitting Gaussian functions to the pulse-height and charge spectra.

Using as reference the INRAD crystal we get the percentage deviation of the mean values. We can see an overall small shift (also visible in the above Figures), possibly due to change of optical grease for the newly (at the time of the experiment) coupled with the PMT PROTEUS crystals. Apart from that, both crystals seem to perform similarly and up to expectations.

Assigning the above values the theoretical Compton edge ones (and the average value of the two Compton edges of ^{60}Co), we can get a “rough” calibration of amplitude and charge to energy seen in the Figure below.

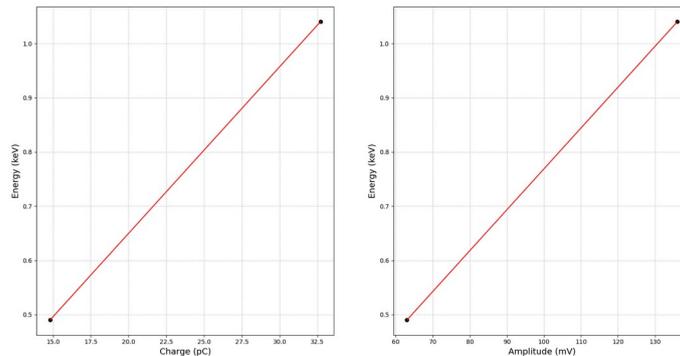


Figure 3.10: Energy calibration using two sources. Charge to energy (left). Amplitude to energy (right).

Thus obtaining the following two relations:

$$\text{Energy} = 0.03 \cdot \text{Charge} + 0.035 \quad (\text{charge in pC})$$

$$\text{Energy} = 0.0075 \cdot \text{Amplitude} + 0.015 \quad (\text{amplitude in mV})$$

Errors are of no importance because we get a fit by using only two points. As mentioned earlier, the above calibration is mostly a qualitative one.

3.3 – Rise/Fall time measurements

In the context of radiation detectors, rise time and fall time are important characteristics of the output signal, reflecting the detector's response to an incident radiation event. These parameters are crucial for understanding the detector's performance and for optimizing the signal processing electronics.

Rise time refers to the time it takes for the output signal to increase from a specified lower percentage to a higher percentage of its maximum amplitude. It is a measure of how quickly the detector responds to an incident radiation event. Typically, rise time is defined as the time interval for the signal to go from 10% to 90% of its peak value (as it is the case in this work). *Fall time* on the other hand refers to the time it takes for the output signal to decrease from a specified higher percentage to a lower percentage of its maximum amplitude. Like rise time, it is usually defined as the time for the signal to drop from 90% to 10% of its peak value. Fall time indicates how quickly the detector recovers after detecting an event, which is important for resolving closely spaced events.

In this context, the typical analysis procedure performed for the calculation of each time, can be visually depicted in the following Figure:

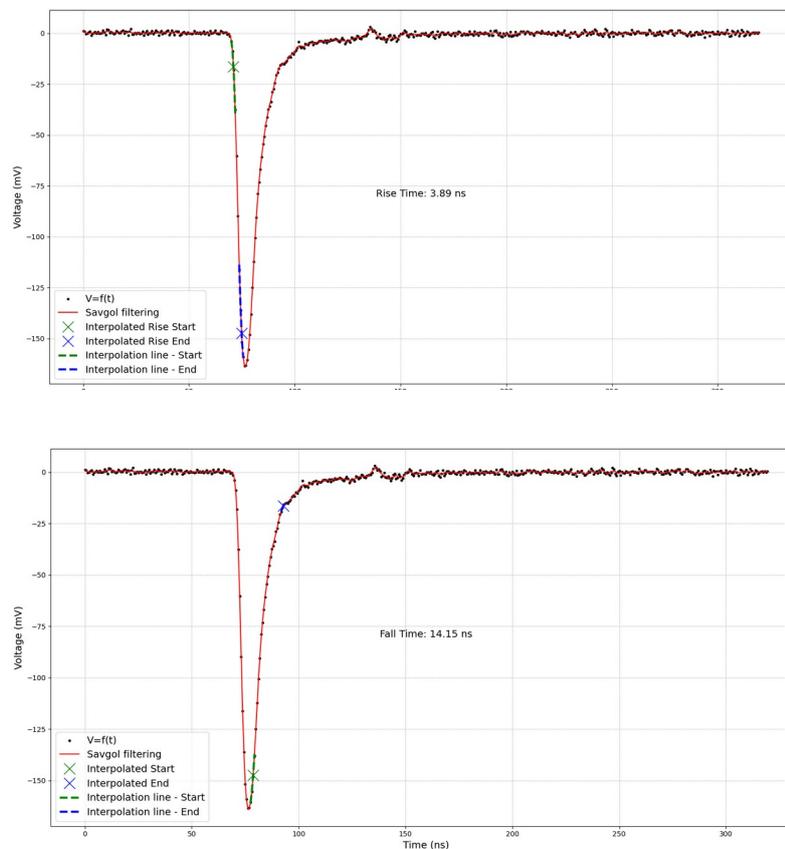


Figure 3.11: Rise/Fall time extraction example. “X” marks on the 10, 90 % on the interpolation lines. Savitzky-Golay filter added for low SNR pulses (denoted as “Savgol”). Rise time 10-90% (top). Fall time 90-10% (bottom).

To minimize the risk of losing events due to the difficulty of interpolating low SNR (Signal to Noise Ratio) pulses for the extraction of 10% and 90% of the maximum amplitude, a Savitzky-Golay filter was implemented.

Savitzky-Golay filtering:

A Savitzky–Golay filter is a digital filter that can be applied to a set of data points for the purpose of smoothing the data, that is, to increase the precision of the data without distorting the signal tendency. This is achieved, through convolution, by fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares. In the analysis code both the number of adjacent data points and the degree of the polynomial were carefully selected in order to leave the signal and its characteristics, as much as possible, untouched while lowering noise fluctuations to a degree suited for correct interpolation of the areas of interest (10%, 90% of the max-amplitude values).

When the data points are equally spaced, an analytical solution to the least-squares equations can be found, in the form of a set of "convolution coefficients" that can be applied to all data sub-sets, to give estimates of the smoothed signal, (or derivatives of the smoothed signal) at the central point of each sub-set [54].

Though it will not be a detailed description of the process involved during the filtering, this and above paragraphs are aiming at showcasing its importance, as it was implemented further, in later sections of the analysis process, where smoothing of noisy signals was crucial in keeping the statistics always high. The following Figure is displaying the use of Savitzky–Golay filtering in a noisy signal.

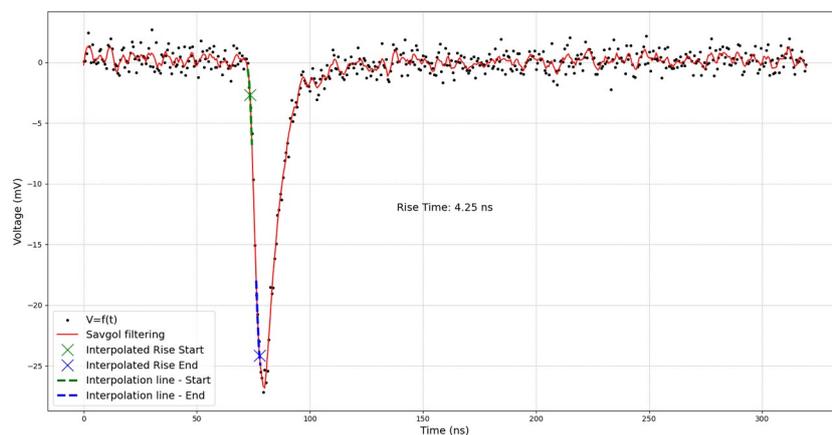


Figure 3.12: Rise time extraction example using Savitzky-Golay filter (denoted as “Savgol”). Initial data points with black. “Savgol” filtered line with red.

By interpolating the area close to 10% and 90% of the max-amplitude of each signal, after applying the filter, and extrapolating the values, we can get a correct measurement of the two times. Averaging over approximately 40000 recorded events we get the following results concerning the two crystals.

	Rise Time (ns)	Fall Time (ns)
INRAD	3.966 ± 0.003	14.13 ± 0.06
PROTEUS	4.051 ± 0.002	14.43 ± 0.05

Table 3.2: Rise/Fall time measurements for INRAD and PROTEUS crystals. Averaged for ~40000 events.

Looking at the above values we can clearly see that the difference in performance is minuscule. Both operate in a similar way and as expected following previous works on this attribute of stilbene crystals.

Once again I ought to stretch the importance of digitization that offered a clear window into a “purely” off-line analysis process, making the above and following work much easier.

3.4 – γ - γ Time Coincidence measurements

3.4.1 – Time Pick-Off Methods:

General Overview:

Concerning a large number of applications, the correct and precise arrival time of a quantum of radiation in the detector is of paramount importance. The accurate timing of the signals depends on both the properties of the detector and the type of electronics used to process them. Limitations in the above factors can have a central role in worsening the timing technique. In earlier years that digitization was on early stage or didn't exist, experimentalists often had to compromise for different timing technics. Off-line time pick-off methods for a set of digitized pulses that can implement any kind of timing technique, once again, have a crucial role in abolishing the above limitations in electronics.

The best timing performance is achieved with the fastest detectors, which collect the signal charge the quickest. Among detectors with similar charge collection times, those that produce the highest number of information carriers (ion pairs or electron-hole pairs) per pulse are less affected by signal "graininess" and thus exhibit better timing properties, i.e. better signal to noise ratio improves time resolution. The timing characteristics of a system are significantly influenced by the dynamic range (ratio of maximum to minimum pulse height) of the signal pulses. When signal pulses are limited to a narrow amplitude range, various timing schemes can yield good results. However, if the pulse amplitude spans a wide range, some loss in timing accuracy is almost inevitable [3].

Time Pick-Off inaccuracies:

The fundamental operation in timing measurements is generating a logic pulse, with its leading edge indicating the time of an input linear pulse, using devices known as time pick-off units or triggers. Timing inaccuracies are inevitable and can be categorized into two types: *time jitter* and *amplitude walk* (or time slewing). Time jitter refers to uncertainties present even when the input pulse amplitude is constant, while amplitude walk arises from varying input pulse amplitudes. Optimal timing performance occurs when input pulses have a narrow amplitude range, minimizing uncertainty to sources of time jitter alone. However, practical applications often involve pulses of varying amplitudes, where amplitude walk further degrades time resolution. An illustration of the above problems can be seen in the figure below:

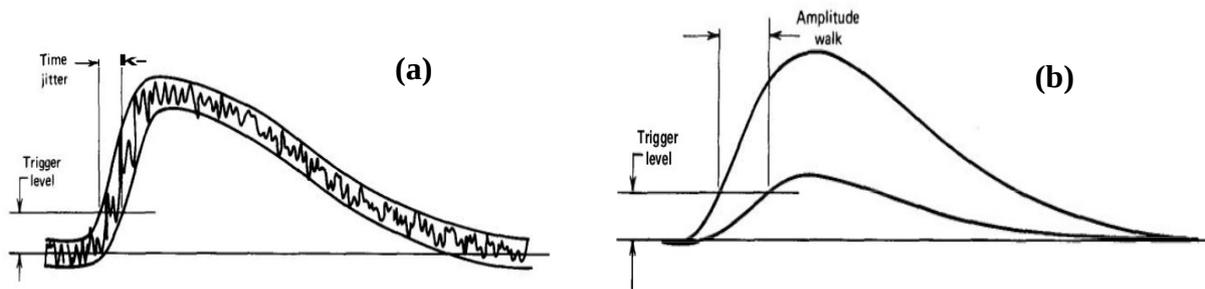


Figure 3.13: **(a)** The time jitter in leading edge triggering arising from random noise. Many signal pulses but with a random contribution of noise [3].

(b) Amplitude walk in leading edge triggering. Two pulses with identical shape and time of occurrence but different amplitude are seen to cross the trigger level at different times [3].

Time jitter is significantly influenced by random fluctuations in signal pulse size and shape. These fluctuations can result from electronic noise added by components processing the linear pulse before time pick-off or from the discrete nature of the electronic signal generated in the detector. When the number of information carriers in the signal is low, statistical fluctuations in their number and timing will cause size and shape variations in the pulse. This effect is most pronounced for small-amplitude pulses and detectors that generate few information carriers, such as scintillation counters [3].

Leading Edge Timing:

Leading edge timing is one of the easiest and most straightforward methods. One has just to measure the time that the pulse crosses a predefined discrimination level. Triggering in this way can be very beneficial when the dynamic range of the input pulses is small, if not the aforementioned timing inaccuracies arise. The effect of time jitter and amplitude walk in leading edge timing can be seen in the Figure 3.13. On one hand, random fluctuations superimposed on signal pulses of identical size and shape may cause the generation of an output logic pulse at somewhat different times with respect to the centroid of the pulse (Fig 3.13 - a). On the other hand, two pulses which have identical true time of origin can give rise to output logic pulses that differ substantially in their timing, that can amount to the full rise time of the input pulse and often creates inaccurate timing, unacceptable in a large dynamic range set-up (Fig 3.13 - b).

To minimize uncertainties due to jitter, the discrimination level should be in the steep slope region of leading edge. Practical compromises in these somewhat conflicting requirements often lead to optimum time resolution for levels that are set at about 10-20% of the average pulse amplitude. In this work trigger thresholds of -5 and -10 mV were tested and leading (trailing) edge measurements were made via the integrated CAEN wave-catcher data acquisition software.

Crossover Timing:

When dealing with a wide range of pulse amplitudes, leading edge triggering can generate the above uncertainties. To cope with that, other methods have been developed that can greatly reduce the dependence on amplitude variations and thus amplitude walk, but only at the expense of increased time jitter. This arises from the fact that this method requires bipolar pulses that ought to have a zero-crossing point as their time stamp. Therefore, methods for shaping the signal into a bipolar one, introduce noise in the shaping stage, thus increasing the risk of jitter due to statistical fluctuations in the zero-crossing point.

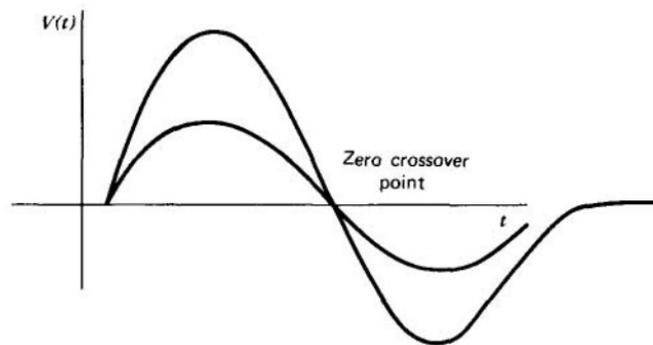


Figure 3.14: Bipolar pulses of different amplitude showing the same zero-crossing time [3].

In this work two types of methods were used in order to create a bipolar pulse. It was achieved through the use of the Derivative of the input signal and the use of a CDF (Constant Fraction Discriminator), solely through off-line analysis.

Derivative method:

The simplest of the two is the derivative method. In the Figure below, we can see an example of the bipolar pulse created when the derivative of the input pulse is taken. A Savitzky-Golay filter is also applied after differentiating the signal to accurately interpolate low SNR pulses. We can then interpolate around the zero-crossing time and extrapolate its value. The Savitzky-Golay filter also reduces any "damage" done during the shaping process of the input signal into a bipolar pulse, thereby limiting the contribution of time jitter, as mentioned earlier. This method, in its simplicity, provides excellent timing results, especially with the advent of digitization. Prior to the digital era, derivative methods were difficult to implement, necessitating the development of alternative methods for determining zero-crossings.

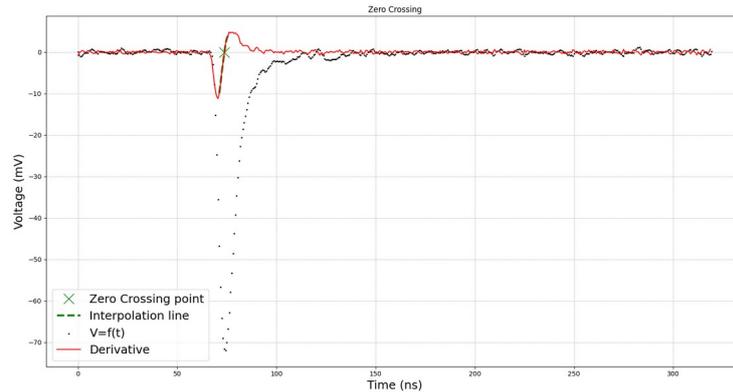


Figure 3.15: Input signal $V=f(t)$ (black dots). The Derivative + Savgol filtering of the input signal (red). Interpolation line and “X” marked zero-crossing time (green).

CFD (Constant Fraction Discriminator) method:

If the dynamic range is small, leading edge timing exhibits superior performance compared to crossover timing due to lower jitter. Empirical evidence suggests that optimal leading edge timing is achieved when the timing discriminator is set to about 10-20% of the pulse amplitude. This insight led to the development of a time pick-off method that generates an output signal at a fixed time after the pulse's leading edge reaches a constant fraction (thus the name) of its peak amplitude. This method ensures that the timing point is independent of pulse amplitude for pulses of consistent shape. Consequently, it can handle a wide dynamic range of pulses with minimal amplitude walk and reduced jitter.

The input signal is first spited into two. One half is delayed for a time greater than the rise time (similar to the fall time in this work), while the other half is inverted and then attenuated up to a percentage of the maximum amplitude of the signal. Taking the sum of the two gives rise to a bipolar pulse shape. Lastly, we can follow the same procedure for the summed signal as we did with the differentiated one (Savgol filtering, interpolation around the zero-crossing time and extrapolation of its value). An example of the aforementioned procedure can be seen below.

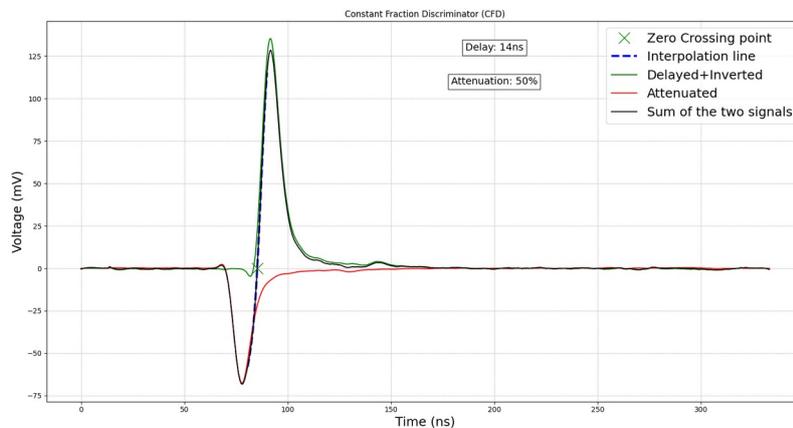


Figure 3.16: Delayed+Inverted (green). Attenuated (red). Sum of the signals (black) Interpolation line (blue) and “X” marked zero-crossing time (green).

3.4.2 – Time Resolution Determination:

Using the multi-detector setup described in Chapter 2.2.2, with a ^{60}Co source placed at the centroid of the tetrahedron, and the three timing techniques mentioned above, time coincidences between pairs of detectors were performed. The fully symmetric setup and the multichannel recording capabilities of CAEN's digitizer allowed for an easy measurement of the timing resolution of the detectors. By accurately determining the time-stamps of a large number of consecutive pulses and plotting their time differences that fall within a specified time window, carefully chosen for our crystals, we can deduce the timing resolution. This provides an easy method for comparing the timing performance of the crystals while at the same time studying the different timing techniques.

After plotting the time-differences distribution, we can fit a Gaussian function to the data and extract its FWHM (Full Width at Half Maximum). Using the equation below, we can then determine the timing resolution of each individual detector as:

$$\text{Time Resolution} = \frac{\text{FWHM}}{\sqrt{2}} \quad (26)$$

The “ $\sqrt{2}$ ” comes from the fact that we use two detectors, assuming equal time resolution, every time we obtain a distribution of time differences and $\text{FWHM}_{\text{Gaussian}} = 2\sqrt{2 \ln 2} \cdot \sigma$. Thus:

$$\text{FWHM}_{\text{combined}} = 2\sqrt{2 \ln 2} \cdot \sigma_{\text{combined}} = 2\sqrt{2 \ln 2} \cdot \sqrt{2} \sigma \Rightarrow \sigma = \text{Time Res.} = \frac{\text{FWHM}_{\text{combined}}}{2\sqrt{2 \ln 2} \cdot \sqrt{2} \sigma} \quad (27)$$

where $\sigma_{\text{combined}} = \sqrt{\sigma_1^2 + \sigma_2^2} = \sqrt{2} \cdot \sigma$. The results can be summarized below in the following Table.

A -10mV trigger threshold was used for the leading edge timing.

Time Resolution (ns)		
	INRAD - PROTEUS	INRAD - INRAD
Derivative	0.74 ± 0.03	1.08 ± 0.02
Leading Edge	0.78 ± 0.03	1.12 ± 0.02
CDF	0.73 ± 0.03	1.13 ± 0.03

Table 3.3: Time Resolution calculations for INRAD - PROTEUS and INRAD - PROTEUS detector pairs, using the three timing technics mentioned above.

As we can see, the detectors seem to perform similarly in this aspect, although PROTEUS shows a slightly better resolution. Further investigation on the above, that is out of this work because of time constraints, is needed in order to be certain.

Chapter 4 – Pulse Shape Analysis (PSA)

- *Pulse Shape Discrimination (PSD)*
- *Principal Component Analysis (PCA)*

4.1 – Pulse Shape Discrimination (PSD)

4.1.1 – PSD Properties in Organic Scintillators:

The prompt fluorescence in most organic scintillators is the primary source of observed light, but a longer-lived component, known as delayed fluorescence, is also present (as described in Chap. 1.5.1). This delayed component can be represented by a sum of two exponential decays, the fast and slow components. The slow component typically has a decay time of several hundred nanoseconds and is influenced by the nature of the exciting particle. This property is used in pulse shape discrimination (PSD) to differentiate between particles of different types that deposit the same energy in the detector.

The slow scintillation component is believed to originate from the excitation of long-lived triplet states along the track of the ionizing particle. Bimolecular interactions between these excited molecules lead to delayed fluorescence. The yield of the slow component depends on the rate of energy loss and is greatest for particles with large energy loss rates. This is confirmed by measurements of the scintillation pulse shape from various organic scintillators [3].

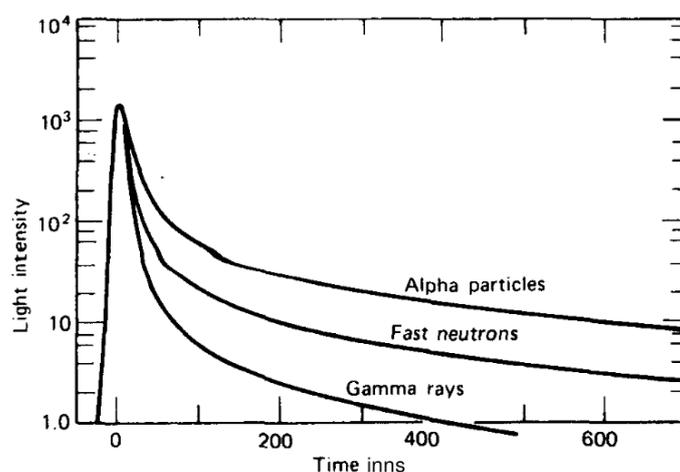


Figure 4.1: The time dependence of scintillation pulses in stilbene (equal intensity at time zero) when excited by radiations of different types [3, 55].

Certain organic scintillators, such as stilbene crystals and commercial liquid scintillators, are favored for PSD due to large differences in the slow component induced by different radiations. These scintillators can not only differentiate between radiations with large energy loss differences but also separate events from various heavy charged particles. Electronic circuits have been designed to perform PSD and are used in various applications. In the presence of a digitizer no such circuitry is needed, thus significantly simplifying the experimental set-up and at the same time improving flexibility.

The pulse shape is influenced by the time profile of the current produced at the detector, which can be affected by factors such as the type of radiation and the detector's collection time constant. Thus PSD methods can be categorized into two main approaches: electronic methods that sense differences in the rise/fall time of the output pulse and methods that integrate the total charge over different time periods. The second approach (which was the one used in this work) involves integrating the total charge over different time periods and calculating the ratio of these signals. This method is less sensitive to pulse amplitude and can be used to distinguish between pulses of different shapes, such as gammas and neutrons (as seen in Figure 4.1).

However, a pioneering approach using multivariate analysis through PCA (Principal Component Analysis), facilitated a method of handling multiple characteristic parameters of a pulse (pulse-height, charge integration on different time windows and pulse-width). In the following chapters we will rigorously interpret the use of this technique complementing the standard PSD.

4.1.2 – FOM (Figure Of Merit):

The figure of merit (FOM) is a key metric used to evaluate the performance of PSD. It is a measure of the separation that can be achieved between different types of events in a given application. The FOM is likely to depend on the dynamic range of the input pulses, which is defined as the ratio between the maximum and minimum amplitude pulses acceptable by the system. In the context of PSD, the FOM is used to quantify the ability of a system to distinguish between different types of radiation, such as neutrons and gamma rays and is typically calculated based on the distribution of PSD values obtained. A schematic of the procedure can be seen below:

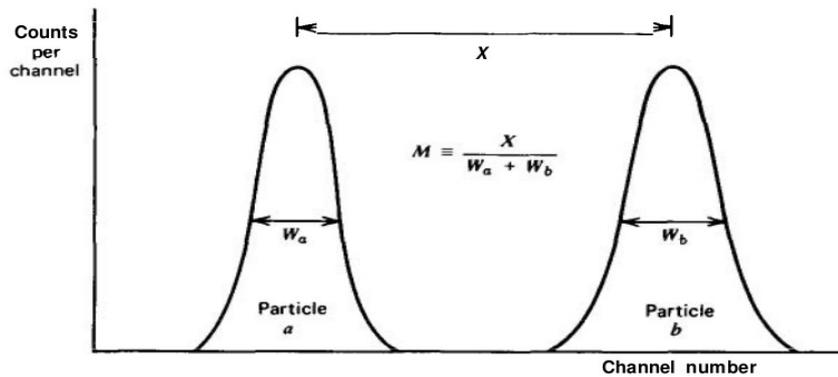


Figure 4.2: Definition of the figure of merit M for PSD [3].

Thus, the FOM is defined as:

$$\text{FOM} = \frac{\mu_2 - \mu_1}{FWHM_1 + FWHM_2} \quad (28)$$

where μ_1 , μ_2 are the mean values of the two Gaussian distributions, and $FWHM_1$, $FWHM_2$ are their Full Width at Half Maximum values. In our case, “1” represents neutrons and “2” gamma-rays.

4.2 – Principal Component Analysis (PCA)

Basic Concept of PCA:

Principal component analysis (PCA) is a linear technique used to reduce the dimensionality of a dataset while preserving the most important patterns and trends. It transforms the data onto a new coordinate system where the directions (Principal Components or PCs) that capture/preserve the largest variation in the data can be easily identified. The principal components are a sequence of unit vectors that best fit the data while being orthogonal to the previous vectors. These directions form a new orthonormal basis where individual dimensions of the data are linearly uncorrelated. PCA is commonly used to plot high-dimensional data in two or three dimensions, making it easier to identify clusters and patterns [58].

The goal of PCA is to find a set of orthogonal axes (principal components) that capture the maximum variance in the data. These new axes are linear combinations of the original variables. In simple terms, the steps to perform PCA are as follows:

- **Data Standardization:** Since PCA is affected by the scales of the variables, standardizing (normalizing) the data to have a mean of 0 and a variance of 1 is a common preprocessing step.
- **Computing the Covariance Matrix:** The sample covariance matrix captures the correlations between pairs of variables.

$$S(x, y) = \text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (29)$$

where x_i , is the i-th observation, and \bar{x} is the mean of the observations.

- **Computing the Eigenvalues and Eigenvectors:** The eigenvalues and eigenvectors of the covariance matrix are computed. The eigenvectors (principal components) determine the directions of the new feature space, and the eigenvalues determine their magnitude (variance) along these new axes.

$$S \cdot \mathbf{v}_k = \lambda_k \cdot \mathbf{v}_k \quad (30)$$

where λ_k and \mathbf{v}_k are the eigenvalue and eigenvector of the covariance matrix S , respectively.

- **Sorting the Eigenvalues and Eigenvectors:** The eigenvalues and their corresponding eigenvectors are sorted in descending order. The top k eigenvalues and their eigenvectors are selected to form the principal components.

- **Transforming the Data:** The original data is projected onto the new principal component axes.

$$Z = X V \quad (31)$$

where Z is the transformed data set, X is the original data matrix, and V is the matrix of eigenvectors (principal components).

Interpreting the above we can deduct that the eigenvalues (λ_k), are indicating the amount of variance captured by each principal component, the eigenvectors (\mathbf{v}_k), are defining the direction of each Principal Component (Z), i.e. the new variables that are linear combinations of the original ones.

PCA helps in identifying the most important features and visualizing high-dimensional data in a lower-dimensional space, which is especially useful in exploratory data analysis, pattern recognition, and data compression.

Scree Plot:

The scree plot, is a subjective method, in its usual form, as it involves looking at a plot of λ_k against k and deciding at which value of k the slopes of lines joining the plotted points are ‘steep’ to the left of k , and ‘not steep’ to the right. This value of k , defining an ‘elbow’ or ‘knee’ or ‘point of inflexion’ in the graph, is then taken to be the number of components m to be retained [58].

Apart from the elbow method, another subjective rule was used to define the number of PC components to be retained. The Kaiser rule, also known as the Kaiser-Guttman criterion, is a method to decide how many principal components to retain by considering their eigenvalues. According to this rule, only the PCs with eigenvalues greater than 1 should be retained. This criterion is based on the idea that a component should account for more variance than a single original variable in standardized data.

Both methods contribute in visualizing the point at which the eigenvalues start to level off, indicating that additional components contribute progressively less to explaining the variance in the data. In the figure below, derived from experimental data that will be discussed in detail in the following chapter, an example of these methods can be seen.

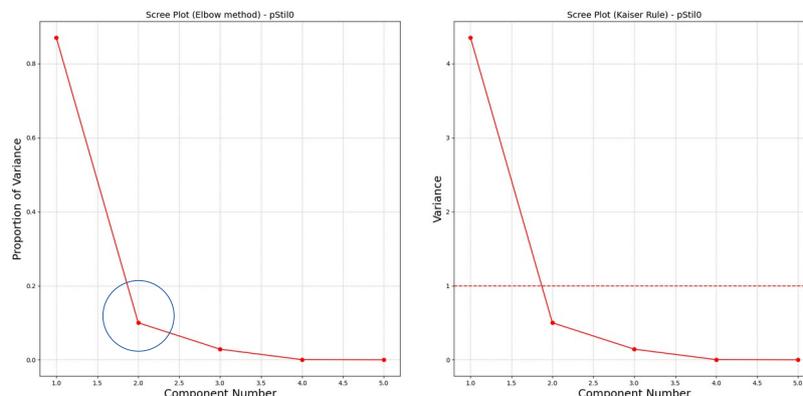


Figure 4.3: Scree plots for five PCs. ‘Elbow’ method (left). ‘Kaiser rule’ (right)

Loadings Plot:

A loadings plot, also known as a factor loadings plot or a biplot when combined with scores (more on that below), is a graphical representation used in PCA to visualize the relationships between the original variables and the PCs. It provides insight into how each variable contributes to the PCs and helps in interpreting the components.

Mathematically speaking, loadings are the coefficients of the original variables in the linear combinations that define the PCs. For a given PC z_k , the loadings are given by the eigenvectors v_k is the eigenvector (loading) corresponding to the k -th PC, as described in Eq. 31.

The description of such plot follows that, the axes represent the PCs, usually the first (PC1) and the second (PC2) and each point or vector represents a variable from the original dataset. The length of the vector indicates the contribution of the variable to the PC, with a longer vector signifying a higher contribution, while the direction shows the correlation between the variables and the PCs.

The loadings plot is a powerful tool in PCA that aids in interpreting the PCs by visualizing the contribution of each original variable. It helps in understanding the underlying structure of the data and in making informed decisions about the variables and components in the analysis [58]. Examples for our data can be seen in the next Chapter.

Biplot:

A biplot is another enhanced graphical representation used in PCA that displays both the PC scores of the observations and the loadings of the variables on the same plot. This dual display helps in understanding the relationships between observations and variables simultaneously. Scores represent the transformed coordinates of the observations in the new PC space. The position of each point indicates its projection of the original data point onto the PCs.

Interpreting the plot, observations that are close to each other in the plot are similar in the PC space. Variables that point in the same direction are positively correlated, and those pointing in opposite directions are negatively correlated. This way it provides a complete view of the data, helping to uncover patterns, relationships, and contributions of variables in the PC space. This dual representation makes biplots valuable for data exploration, interpretation, and communication with the user in PCA.

Chapter 5 – Results

- *PSD - Results*
- *PCA – Results*

5.1 – PSD - Results

5.1.1 – PSD Method & Useful Parameters:

The PSD capability, as mentioned earlier, is one of the most interesting and intriguing parameters for a neutron detector. In this work, the PSD was performed by using an Am-Be (Americium-Beryllium) source provided by the INFN-CT, along with the facilities that the measurements took place, placed at the centroid of the detector configuration described in Chap. 2.2.2. A different approach was used, diverting slightly from the usual charge integration technique, seen in similar works [45, 57], which implement different time windows for charge integration to define ratios such as Q_{short}/Q_{total} , $\frac{Q_{long}-Q_{short}}{Q_{long}} \dots etc$, with *short*, *long* and *total*, denoting the different time intervals for integration (specifically “total” denotes an extended interval, combining both “short” and “long” components). In our case, while testing such discriminators, the ratio between *pulse-height* and Q_{total} was found to provide the best separation between particles of interest. In conclusion, PSD was defined by the following relation:

$$PSD = \frac{Pulse\ Height}{Q_{total}} \quad (32)$$

Pile-Up:

When radiation detector pulses are randomly spaced in time, higher counting rates can cause overlap between pulses, a phenomenon known as pile-up. This problem can be mitigated by reducing the total pulse width as much as possible. However, constraints such as ballistic deficit and signal-to-noise ratio limit how much the pulse width can be reduced, making pile-up effects significant at high rates.

Pile-up can be categorized into two main types, each affecting pulse height measurements differently. The first type, tail pile-up, occurs when pulses overlap with the long-duration tail or undershoot of a preceding pulse. The second type is called peak pile-up and it occurs when two pulses are sufficiently close to each-other so that they are treated as a single pulse by the analysis system.

As illustrated in Fig. 5.1, when pulses with relatively flat tops overlap, they create a combined pulse with an apparent amplitude equal to the sum of the individual amplitudes. Lesser degrees of overlap produce a combined pulse with an amplitude slightly less than the sum. This type of pile-up causes distortions in the recorded spectrum, including occasional sum peaks, and interferes with quantitative measurements. Since peak pile-up results in recording one pulse instead of two, the total area under the recorded spectrum is smaller than the actual number of pulses presented to the system during its live time [3].

The severity of peak pile-up in a given situation can be estimated using the counting rate and the effective width of the signal pulses. Although the effective width is challenging to define precisely, except for rectangular pulses, it can be approximated by the FWHM of the first lobe of the shaped pulse (Fig. 5.1). We can use the interval distribution equation:

$$I_1(t)dt = r \cdot e^{-rt} dt \quad (33)$$

where, $I_1(t)$ in the distribution function for intervals between adjacent random events within a differential time dt after a time interval of length t and r is the average rate of occurrence [3].

By disregarding second-order effects, the above equation can be used to estimate the expected degree of pile-up. The probability of observing an interval greater than τ is given by:

$$P(>\tau) = e^{-n\tau} \quad (34)$$

where n is the true rate of signal pulses [3]. Concerning this statistical approach to pile-up occurrence probability and by calculating $1 - P(>\tau)$ while considering a time window of 320ns and a rate of 1kHz of the detector, we can extrapolate a probability of pile-up of $\sim 0.32 \times 10^{-3} \%$. This means that for $\sim 10^6$ events that were recorded we are going to get roughly ~ 300 -400 pile-up events.

By selecting high charge pulses and superimposing them on a single plot (Fig. 5.2), we can closely inspect the waveform. As a result, we can distinguish saturated pulses and also double pulses superimposing on the DAQ time dynamics (320ns).

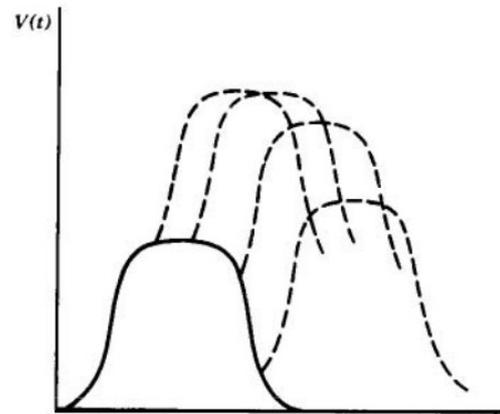


Figure 5.1: Peak pile-up schematic. Two closely spaced signal pulses combine to form one distorted pulse [3].

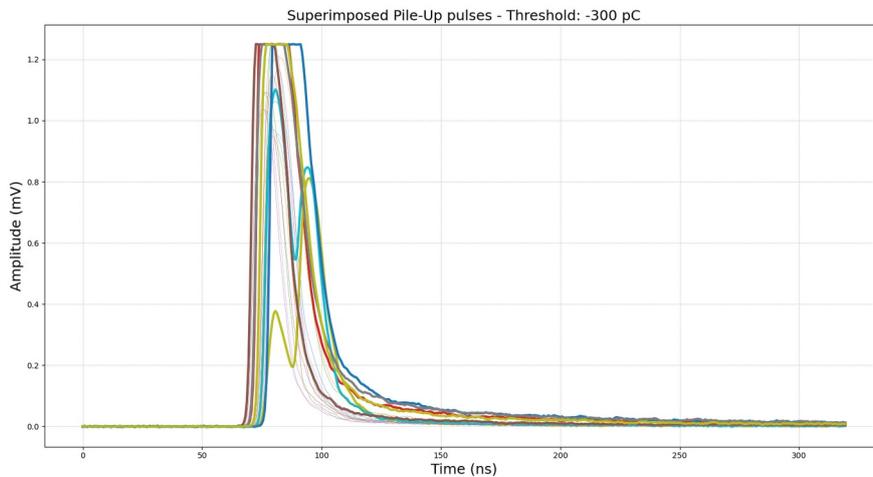


Figure 5.2: A sample of 30 pulses with charge below a threshold of -300pC. Thicker lines were used to distinguish saturated pulses and double-pulses.

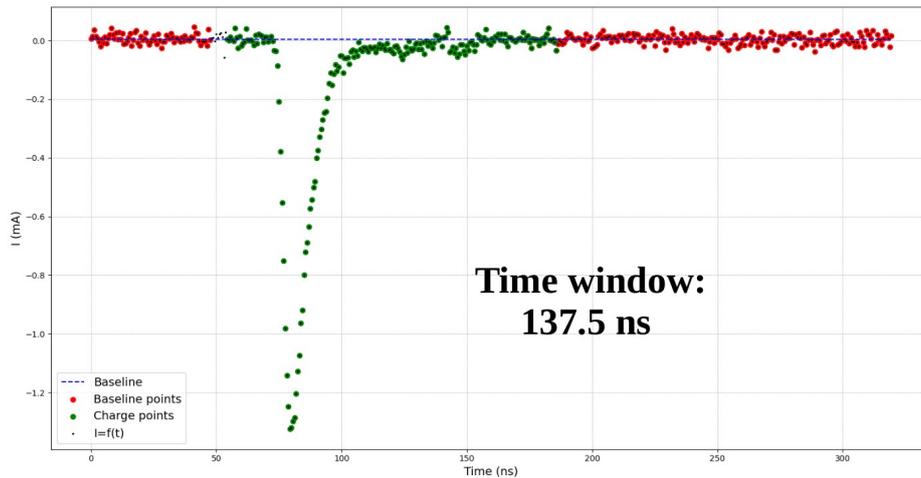
As count rate increases, it reaches a point that it starts to saturate because of pile-up events. The pile-up also affects the PSD analysis when the detectors are exposed to a mixed n/γ radiation field. In this small sample of pulses, apart from saturated or double pulses that may be easy to distinguish and reject, there are also very closely spaced events that are convoluted in a single peak, making it especially hard to separate. Thus, when calculating in the whole range of events, we get a pile-up count of ~ 95 pulses. The deviation from the expected number (~ 300 pulses) is due to the above phenomenon. Hence, a choice was made to introduce a threshold of -300pC to reject all possible pile-up events.

In this way, the non-linearity problem in the PSD distributions, that was prominent in the high-end of detector dynamics, was clearly mitigated by a large factor. This was related to the fact that the charge integration method was over/under estimating the area under the pulse because of discrepancies resulting from pile-up.

5.1.2 – PSD Distributions:

The procedure can be spitted into two steps. First is the extraction and sorting of all the useful parameters, concerning the analysis, for each channel, obtained via CAEN's digitizer wavecatcher software. Then a separate analysis process is applied to the raw extracted data, using Python scripts, by the comparison of the two until a good agreement is reached.

Pulse-height extraction is based just on the minimum value of the signal vector. Charge (Q_{total}) is based on finding the same minimum vector value and using it as a reference point to select an integration window. For both procedures, the baseline is fitted (by a 0^{th} degree polynomial) using the first and last values and then is subtracted. In the following figure we can see an example of the procedure leading to charge integration.



*Figure 5.3: An example of the charge integration via off-line analysis scripts.
With red the values used for baseline fitting.
With green the values used for charge integration.
Dotted blue line depicts the fitted baseline.*

We can finally present the PSD distribution for each individual channel/detector, denoted as pStil#, in the following figure. pStil0 refers to the PROTEUS crystal, while the rest refer to the three INRAD ones.

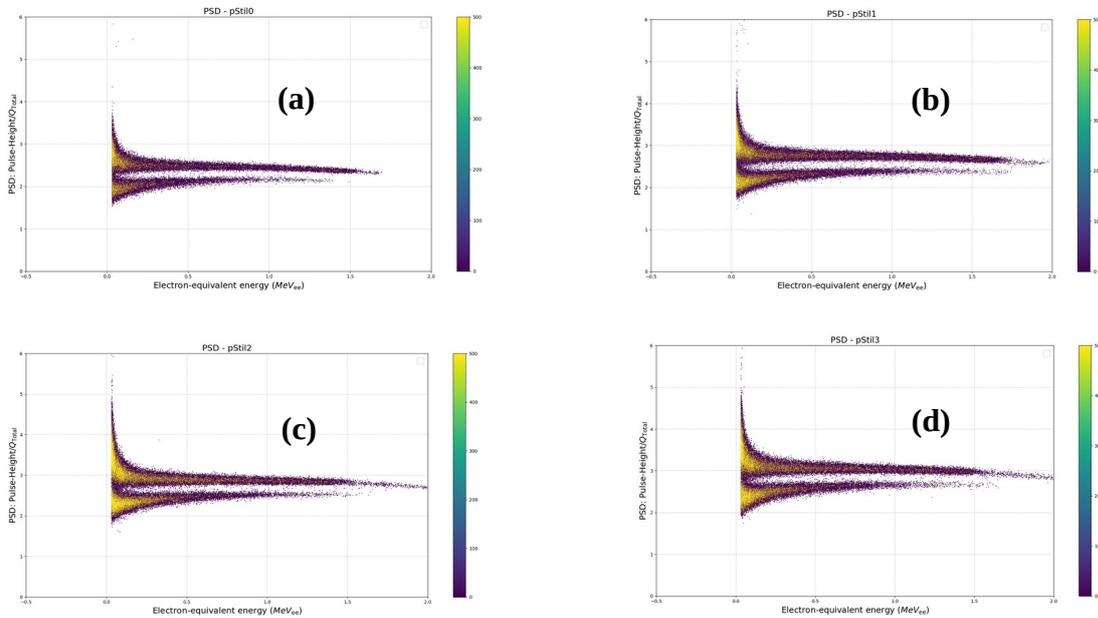


Figure 5.4: 2D PSD histogram for the four stilbene crystals irradiated with a Am-Be source PSD as a function of electron-equivalent energy (MeVee), with a pile-up threshold of -300pC. pStil0 (a), pStil1 (b), pStil2 (c), pStil3 (d).

Again all crystals perform similarly. They show excellent separation between gamma-rays and neutrons, validating their PSD capabilities. Further improvements may include adjustments in the charge integration windows, with possible implementations of a moving time window.

Moving forward we can calculate the PSD FOM as described by Eq. 28, in two separate energy regions. In the following figure we see a fitted double-Gaussian distribution to the PSD values, for an energy range of 60-360 keVee and 800-1100 keVee respectively.

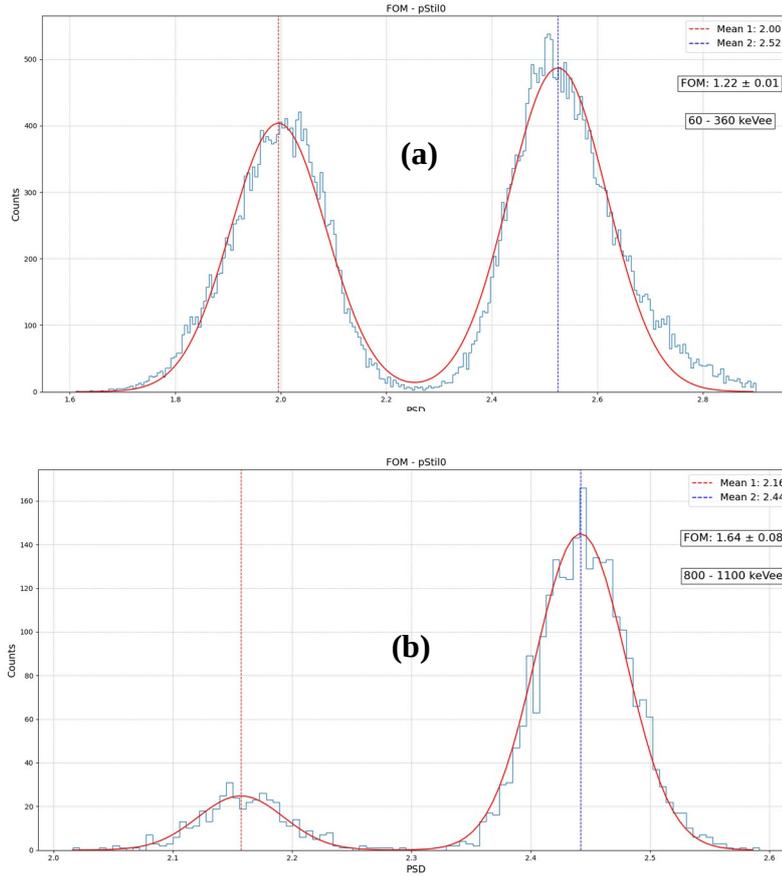


Figure 5.5: Example of PSD FOM distributions (blue histograms) for $pStil0$, with a double-Gaussian fitted (red).
 Low energy region 60-360keVee (a)
 High energy region 800-1100keVee (b)

The above figures serve as an insight to the correlation of PSD FOM to low and high energies. Summarizing for all the detectors, we get the comparative results in the Table below.

	PSD FOM	
	60-360 keVee	800-1100 keVee
pStil0	1.22 ± 0.01	1.64 ± 0.08
pStil1	1.29 ± 0.01	1.69 ± 0.03
pStil2	1.22 ± 0.01	1.86 ± 0.06
pStil3	1.22 ± 0.01	1.60 ± 0.04

Table 5.1: PSD FOM values calculated for the four detectors in two energy regions.

We can then slice FOM as a function of energy as depicted in Figure 5.6 for the four pStil detectors. The following figure is depicting the above idea.

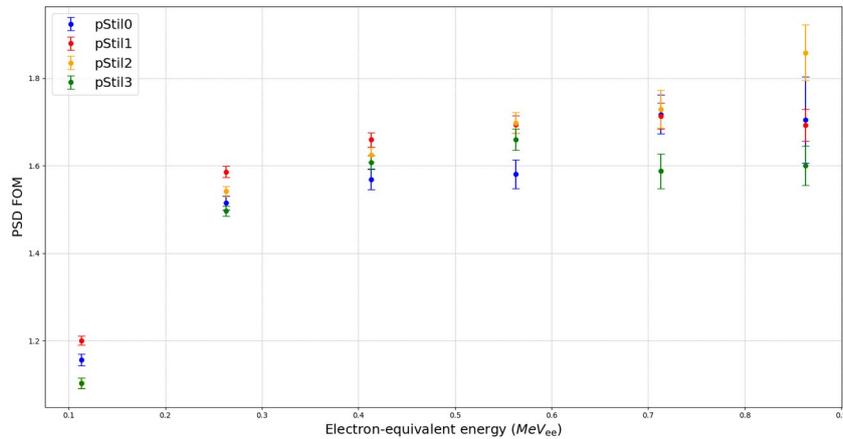


Figure 5.6: PSD FOM as a function of energy.

As a final note, I ought to address that it is extremely difficult to comparatively assess the results of the quality of PSD FOM to other works. The FOM metric is a quality factor that depends on various parameters such as the type of the scintillator, the quantum efficiency of the PMT, the energy of the incident neutrons and the performance of electronics. As a matter of fact, just accounting for the quantum efficiency of a PMT, differences in FOM up to 30% can be introduced.

5.2 – PCA - Results

5.2.1 – PCA Method & Useful Parameters:

The PCA method used in this work follows the exact description of Chap. 4.2. We adapt Eq. 29 to 31 to the PSD extracted data, converting it into PCs, with the subsequent analysis and characterization through Scree plots, Loadings plots and Biplots. The evaluated results will be discussed at the end of this Chapter.

We can finally present the initial parameters. Five were used: **Pulse-height**, three for the charge parameter, **Charge (short, long, tail)** with short, long tail denoting the time window for charge integration; and the introduction of the last discriminatory parameter, **Pulse-width**. On a short note, pulse-width can be schematically represented by the following figure as the difference between trailing and leading edge timing of the signal.

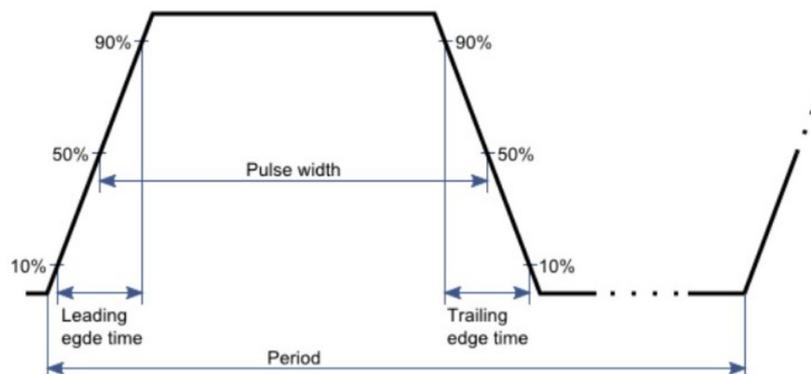


Figure 5.7: A schematic representation of pulse-width.

5.2.2 – Implementation:

After the extraction of the five PCs we need, as aforementioned, to select which components contribute more in preserving the variance and find the point at which the eigenvalues start to level off. For this we plot the Scree plots of the five components, that can be seen below.

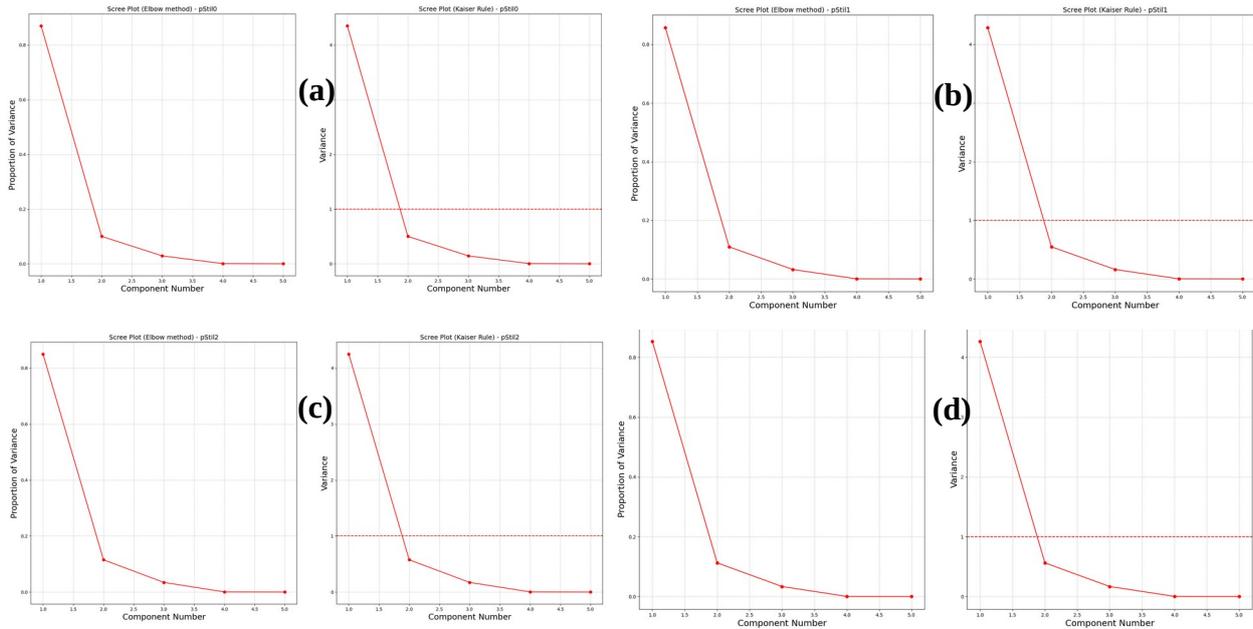


Figure 5.8: Scree plots for each detector (a to d). “Elbow” method and Kaiser rule applied for each one.

As we clearly see, for each detector both methods denote a large drop-off of variance after the first PC. Being a physics related thesis though, we ought to include and subsequently check at least the second and/or third component. Even a small additional percentage of variation being preserved, may hint on something worth evaluating. As the above methods are of subjective nature, the choice of taking into account the second component was made.

Moving on we can now present the loadings plots for the four channels.

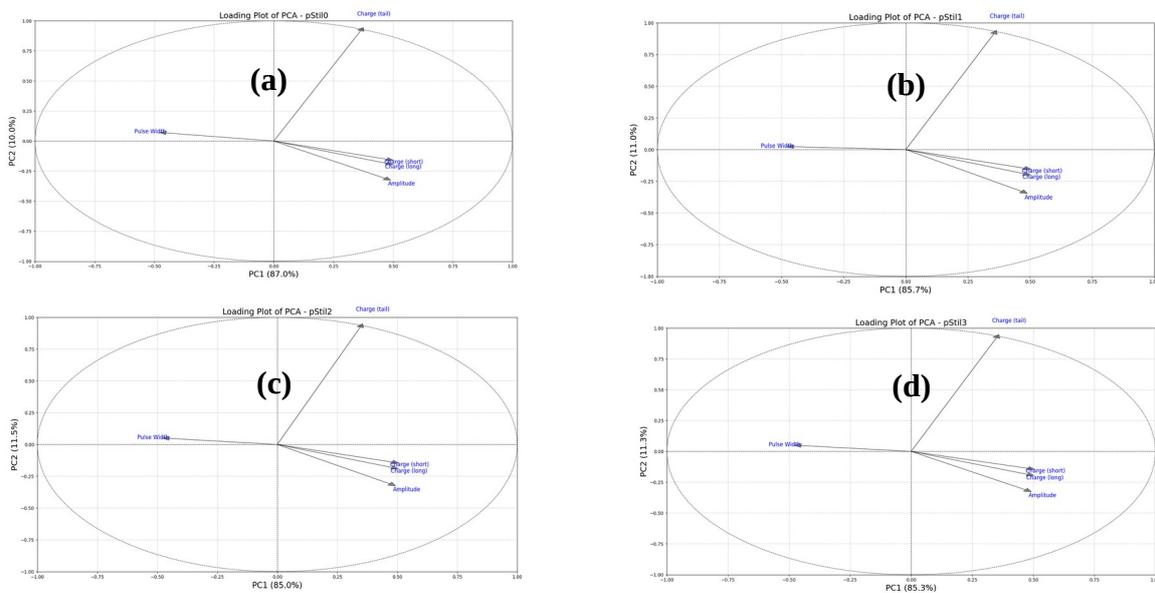


Figure 5.9: Loadings plots for each detector (a to d).

As mentioned earlier, each vector represents a variable from the original dataset, and the length indicates the contribution of the variable to the PC. Charge (short and long) and amplitude (pulse-height) variables are pointing in the same direction as PC1, which preserves at least 85% of the variance, and are positively correlated with it. Charge (tail) is primarily positively correlated with PC2 and has a stronger positive loading on it. Conversely, the pulse-width variable is negatively correlated with PC1 and has a small loading on it.

Overall, more than 96% of the variation is preserved in the first two PCs. The summarized results can be seen on the Table below. For completeness reasons the results of other component combinations can be seen in Appendix A.

	% Variance			
	pStil0	pStil1	pStil2	pStil3
PC1	87.0	85.7	85.0	85.3
PC2	10.0	11.0	11.5	11.3
SUM	97.0	96.7	96.5	96.6

Table 5.2: Percentage variation preserved by each PC for each detector extracted variables.

Before providing a physical interpretation of the results, it would be really helpful to showcase the results of Biplots (scores+loadings). This is where also PSD and PCA are combined. In order to introduce clustering, as we have two different “target groups” (types of particles) in our dataset, we used the results from PSD with the implementation of a custom class for the dynamic selection of an area of indices, solely defined by the user. This step was crucial because, with fixed “borders” between the “gamma-ray area” and the “neutron area” in the PSD distributions, even minor fluctuations in the initial parameters could cause significant variations in the distribution positions.

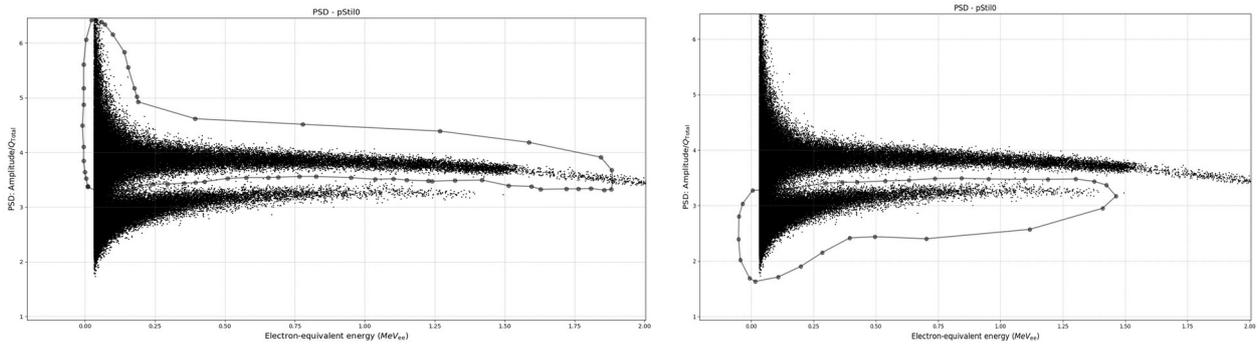
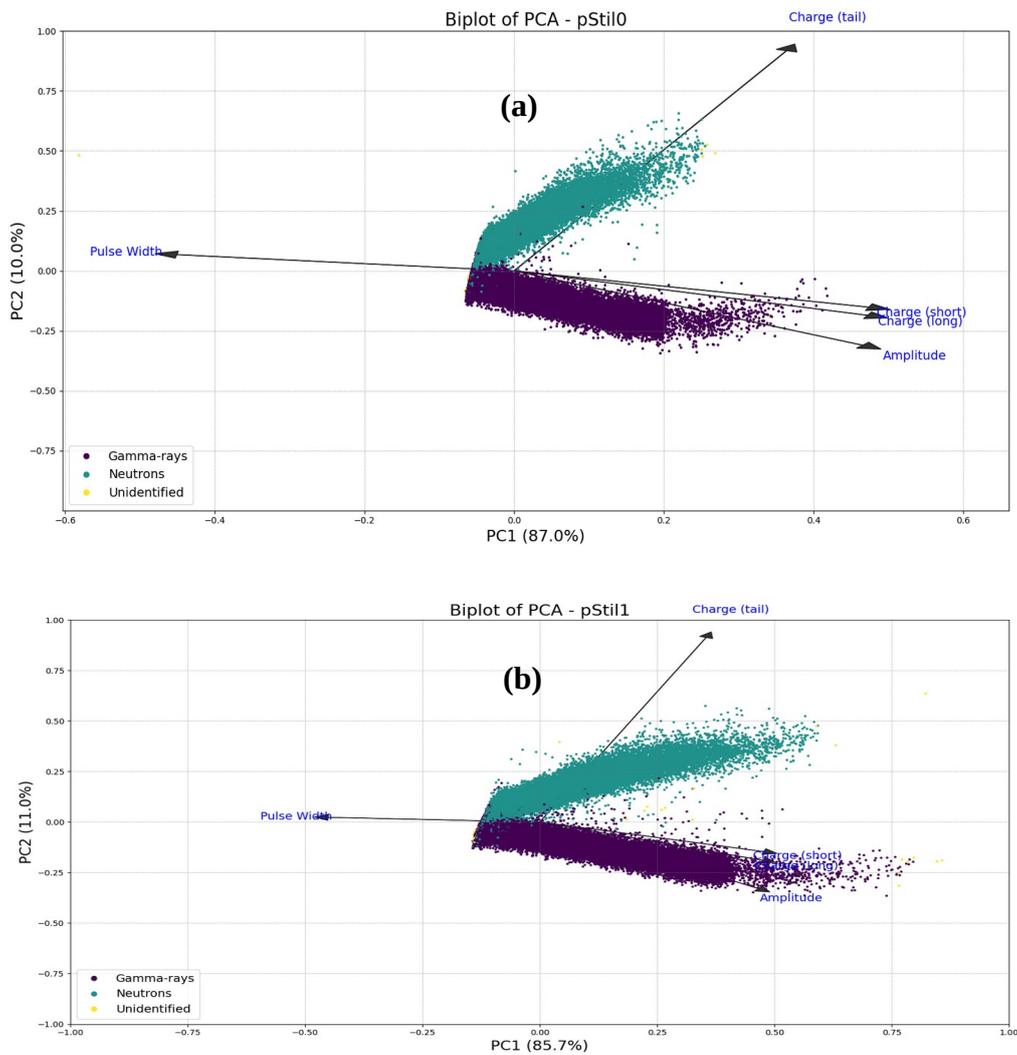


Figure 5.10: Indices selection custom class. Selection using only the left-click mouse button

After the selection of gamma-ray and neutron indices, we can finally present the Biplots obtained.



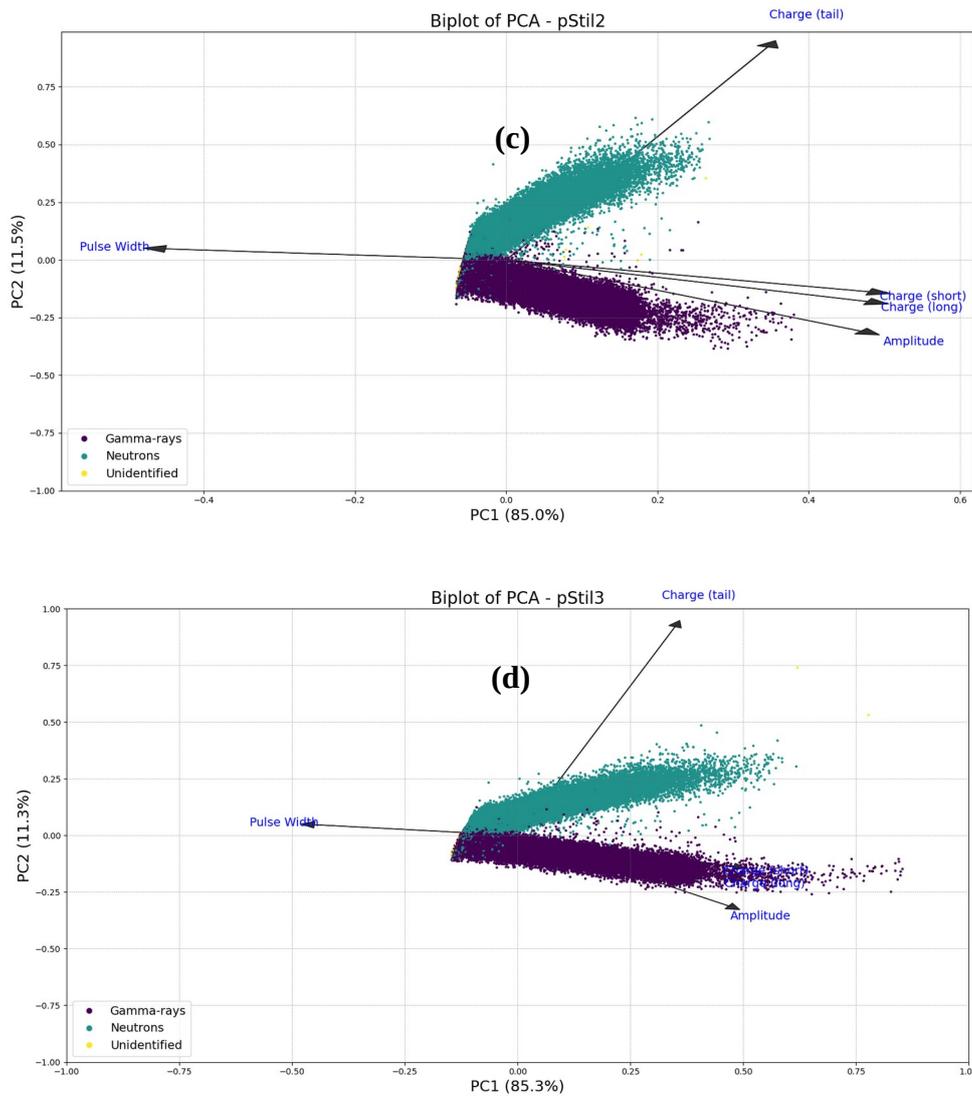


Figure 5.11: Biplots for each detector (a to d).
Gamma-rays with purple. Neutrons with green.
“Unidentified” refers to unselected or very high energy particles.

First of all all detectors perform similarly. Their scores and loadings match, and the separation is excellent and as expected. On a physical note, we can say the following:

Principal Component 1 – PC1:

- **Pulse-height, Charge (short), Charge (long):** These variables have high positive loadings on PC1, indicating they contribute significantly to this component. In the context of particle detection, this suggests that PC1 represents the overall signal strength or energy deposited by the particle, as these parameters are directly related to the signal amplitude and integrated charge.

- **Pulse-width:** This variable has a negative loading on PC1, indicating it is inversely correlated with the other variables on this component. Pulse-width is strictly related to deposited energy (Time over Threshold) and consequently its loading must stay parallel to PC1 axis (as it does).

Principal Component 2 – PC2:

- **Charge (tail):** This variable has a high positive loading on PC2, indicating a strong contribution. This reflects the PSD contribution of slower signal components, confirming the basis of PSD by charge integration.
- **Pulse-height and Charge (long):** These variables have relatively small loadings on PC2, suggesting they are equivalent for the purpose of discrimination, as expected.

Physical Implications:

- **Energy Deposition:** The high positive loadings of pulse-height and charge parameters on PC1 indicate that this component captures the energy deposition characteristics of the particles. This is crucial for identifying the type and energy of the particles detected.
- **Signal Shape and Duration:** The negative correlation of pulse-width with PC1 and its small positive loading on PC2 suggests that the pulse width is related to the pulse amplitude (same vector direction). The anti-correlation could be associated to the fact that the pulse width is calculated at 50% of the signal. This differentiation can help in distinguishing between different particle interactions or types.
- **Tail Charge:** The strong positive loading of charge (tail) on PC2 highlights the importance of the signal's tail in identifying slower processes or particles with extended interaction times, such as certain heavy ions or particles that cause prolonged ionization (this is what we expect).

Overall we can say that PCA is a technique worth implementing in nuclear physics. We showcased that it helps in interpreting the characteristics of detected signals, such as energy deposition, signal shape, and duration. By analyzing the loadings and biplots, we can gain insights into the physical processes underlying the data, aiding in the identification and classification of particles and their interactions.

Furthermore, it creates an opportunity to go beyond usual PSD technics, by implementing Clustering methods such as: Partitioning Methods (K-means, K-medoids), Hierarchical Methods (Agglomerative), etc, together with PCA. They are both unsupervised machine learning techniques, serving different purposes, but together they can be used to enhance data analysis in nuclear physics.

This work provides a starting point to a different approach in nuclear physics analysis, that can greatly complement preexisting technics.

Chapter 6 - Conclusions

➤ *Conclusions*

6 – Conclusions

This research aimed to improve neutron capture measurements by leveraging advanced detection techniques. It explored the use of trans-stilbene organic scintillators provided by two different companies as a safer and more efficient alternative to traditional C_6D_6 detectors, highlighting their excellent neutron-gamma discrimination, good timing performance, and non-toxic properties for both stilbene alternatives. These innovations contribute remarkably to the precision of neutron capture measurements, essential for understanding stellar nucleosynthesis processes. The successful implementation of PCA opens new possibilities for their application in other areas of nuclear physics and related fields.

Beyond the characterization of each individual detector, a fully symmetrical multi-detector arrangement was developed and characterized, placing the trans-Stilbene units in a regular tetrahedral pyramid, such to achieve optimal geometric configuration.

Future research should focus on refining these techniques, exploring their applications in different experimental setups, and investigating additional discriminatory parameters to gain deeper insights into particle interactions. This work lays a strong foundation for enhancing data analysis techniques in nuclear physics, ultimately contributing to a better understanding of fundamental physical processes.

Chapter 7 - Bibliography

➤ *Bibliography*

7 – Bibliography

- [1]: K. S. Krane, Introductory Nuclear Physics, John Wiley and Sons, 1988
- [2] G. R. Satchler, Εισαγωγή στις Πυρηνικές Αντιδράσεις, μετάφραση: Ν. Νικολής, Εκδόσεις Πελεκάνος
- [3]: G. F. Knoll, Radiation Detection And Measurement, John Wiley and Sons, 2000
- [4]: W. R. Leo, Techniques for Nuclear and Particle Physics Experiments – A how to approach, Springer-Verlag, 1987
- [5]: G. R. Gilmore, Practical Gamma-ray Spectrometry, 2 nd Edition, John Wiley & Sons, 2008
- [6]: TALYS User Manual
- [5]: Franz Kaepfeler, R Gallino, Sara Bisterzo, Wako Aoki, The s process: Nuclear physics, stellar models, and observations, Article in Reviews of Modern Physics, January 2011, <https://doi.org/10.1103/RevModPhys.83.157>
- [6]: Electronic Devices And Circuit Theory (10th Edition) Author: Robert L. Boylestad, Louis Nashelsky, Publisher: Prentice Hall Pages: 912 Published: 2008-07-31
- [7]: P.F. Blosler, J.S. Legere, C.M. Bancroft, J.M. Ryan, M.L. McConnell, Balloon flight test of a Compton telescope based on scintillators with silicon photomultiplier readouts, Nucl. Instrum. Methods Phys. Res. A 812 (2016) 92-103, <https://doi.org/10.1016/j.nima.2015.12.047>
- [8]: <https://root.cern/>
- [9]: ROOT User's Guide
- [10]: C. Domingo-Pardo , V. Babiano-Suarez, J. Balibrea-Correa, L. Caballero, I. Ladarescu1, J. Leredegui-Marco, J. L. Tain, A. Tarifeño-Saldivia, O. Aberle, V. Alcayne, S. Altieri, S. Amaducci, J. Andrzejewski, M. Bacak, C. Beltrami, S. Bennett, A. P. Bernardes, E. Berthoumieux, M. Boromiza, ... , The n_TOF Collaboration, Advances and new ideas for neutron-capture astrophysics experiments at CERN n_TOF, Eur. Phys. J. A (2023) 59:8 <https://doi.org/10.1140/epja/s10050-022-00876-7>
- [11]: Agostinelli S. et al, Geant4 – a simulation toolkit, Nucl. Instr. Meth. Phys. Res. A, 506, 3, (2003), 250-303.
- [12]: GEANT4 User's Guide: For Application Developers
- [13]: https://home.cern/science/experiments/n_tof [last visited on 30/05/2024]
- [14]: Cristian Massimi, and The n_TOF Collaboration, Nuclear astrophysics activities at the n_TOF facility at CERN, EPJ Web of Conferences 275, 01009 (2023), <https://doi.org/10.1051/epjconf/202327501009>

- [15]: Sabaté-Gilarte, M., Barbagallo, M., Colonna, N. et al. High-accuracy determination of the neutron flux in the new experimental area n_TOF-EAR2 at CERN. *Eur. Phys. J. A* 53, 210 (2017)
- [16]: G. Tagliente, O. Aberle, V. Alcayne, J. Andrzejewski, L. Audouin, V. Babiano-Suarez, M. Bacak, M. Barbagallo, V. Bécaries, Th. Benedikt, S. Bennett, E. Berthoumieux, J. Billowes, D. Bosnar, A. Brown, M. Busso, M. Caamaño, L. Caballero-Ontanaya, F. Calviño, ... ,P. Žugec19 and the n_TOF Collaboration, *Nuclear Astrophysics at n_TOF facility, CERN, CERN Proc. 1* (2019) pp.251-258
- [17]: The Python Standard Library, <https://docs.python.org/3.10/library/index.html>, [last visited on 16/05/2024]
- [18]: <https://www.python.org/>
- [19]: Claus E. Rolfs, William S. Rodney, *Cauldrons in the Cosmos*, 1988
- [20]: Phillip M. Rinard, “Neutron Interactions with Matter” <http://www.fas.org/sgp/othergov/doe/lanl/lib-www/la-pubs/00326407.pdf>
- [21]: D. L. Garber and R. R. Kinsey, “Neutron Cross Sections, Vol. II, Curves,” Brookhaven National Laboratory report BNL 325 (1976).
- [22]: Evaluated Nuclear Data File ENDF/B-V (available online and maintained by the National Nuclear Data Center of Brookhaven National Laboratory).
- [23]: J. R. Lamarsh, *Introduction to Nuclear Reactor Theory* (Addison-Wesley, Reading, Massachusetts, 1966)
- [24]: S. Glasstone and A. Sesonske, *Nuclear Reactor Engineering* (D. Van Nostrand Co., Inc., Princeton, New Jersey, 1967).
- [25]: Chanh Kim, Jung-Yeol Yeom, Geehyun Kim, Digital n-γ Pulse Shape Discrimination in Organic Scintillators with a High-Speed Digitizer, *Journal of Radiation Protection and Research* 2019;44(2):53-63, <https://doi.org/10.14407/jrpr.2019.44.2.53>
- [26]: National Nuclear Data Center, <https://www.nndc.bnl.gov/>
- [27]: Esam M.A. Hussein, *Radiation Mechanics: Principles and Practice*, 1st Edition, Elsevier - September 19, 2007
- [28]: Vlado Valković, *Radioactivity in the Environment*, 2000
- [29]: J. B. Birks, *The Theory and Practice of Scintillation Counting*. Copyright 1964 by Pergamon Press, Ltd.
- [30]: J. Zhou, N. Gaughan, F.D. Becchetti, R.O. Torres-Isea, M. Febraro, N. Zaitseva, A. Di Fulvio, Light output quenching in response to deuterium-ions and alpha particles and pulse shape discrimination in deuterated trans-stilbene, *Nucl. Instrum. Methods Phys. Res. A* 1027 (2022) 166287

- [31]: R. Batchelor, W. B. Gilboy, J. B. Parker, J. H. Towle, The Response Of Organic Scintillators To Fast Neutrons, *Nucl. Instrum. Methods Phys. Res.* 13 (1961) 70-82
- [32]: N. Patronis, M. Kokkoris, D. Giantsoudi, G. Perdikakis, C.T. Papadopoulos, R. Vlastou, Aspects of GEANT4 Monte-Carlo calculations of the BC501A neutron detector, *Nucl. Instrum. Methods Phys. Res. A* 578 (2007) 351–355
- [33]: R.A. Weldon Jr., J.M. Mueller, C. Awe, P. Barbeau , S. Hedges, L. Li, M. Mishra, J. Mattingly, Characterization of stilbene’s scintillation anisotropy for recoil protons between 0.56 and 10 MeV, *Nucl. Instrum. Methods Phys. Res. A* 977 (2020) 164178
- [34]: C.F. Macrae, I.J. Bruno, J.A. Chisholm, P.R. Edgington, P. McCabe, E. Pidcock, L. Rodriguez-Monge, R. Taylor, J. van de Streek, W.P. A. Mercury, CSD 2.0-new features for the visualization and investigation of crystal structures, *J. Appl. Crystallogr.* 41 (2008) 466–470.
- [35]: Patronis et al. EPJ Techniques and Instrumentation, Status report of the n_TOF facility after the 2nd CERN long shutdown period, 2023, <https://doi.org/10.1140/epjti/s40485-023-00100-w>
- [36]: O. Aberle, V. Alcayne, M. Bacak, J. Balibrea-Correa, N. Colonna, D. Cano-Ott, A. Casanovas, C. Domingo-Pardo, O. Fjeld, F. Gunsing, J. Leredegui-Marco, C. Lederer-Woods, C. Massimi, E. Mendoza, A. Mengoni, A. Manna, A. Musumarra, N. Patronis, M.G. Pellegriti, Letter of Intent to the ISOLDE and Neutron Time-of-Flight Committee, Development of new solid-state total-energy detectors for neutron-capture measurements at CERN n_TOF
- [37]: Ian J. Thompson, Filomena M. Nunes, *Nuclear Reactions for Astrophysics Principles, Calculation and Applications of Low-Energy Reactions*, Cambridge University Press, July 2009 (published), ISBN 978-0-521-85635-5 hardback
- [38]: Ferrari M, Senajova D, Aberle O, Aguiar YQ, Baillard D, Barbagallo M, Bernardes A-P, Buonocore L, Cecchetto M, Clerc V, Di Castro M, Garcia Alia R, Girod S, Grenard J-L, Kershaw K, Lerner G, Maeder MM, Makovec A, Mengoni A, Perez Ornedo M, Pozzi F, Almagro CV, Calviani M. Design development and implementation of an irradiation station at the neutron time-of-flight facility at cern. *Phys Rev Accel Beams.* 2022;25:103001. <https://doi.org/10.1103/PhysRevAccelBeams.25.103001>.
- [39]: Patronis N, Colonna N et al. The CERN n_TOF NEAR station for astrophysics- and application-related neutron activation measurements. 2022. <https://doi.org/10.48550/arXiv.2209.04443>.
- [40]: Babiano V, Balibrea J, Caballero L, Calvo D, Ladarescu I, Leredegui J, Mira Prats S, Domingo-Pardo C. First i-ted demonstrator: a compton imager with dynamic electronic collimation. *Nucl Instrum Methods Phys Res, Sect A, Accel Spectrom Detect Assoc Equip.* 2020;953:163228. <https://doi.org/10.1016/j.nima.2019.163228>.

- [41]: Babiano-Suárez V, Leredegui-Marco J, Balibrea-Correa J, Caballero L, Calvo D, Ladarescu I, Real D, Domingo-Pardo C, Calviño F, Casanovas A, Tarifeño-Saldivia A, Alcayne V, Guerrero C, Millán-Callado MA, Rodríguez-González T, Barbagallo M et al. Imaging neutron capture cross sections: i-ted proof-of-concept and future prospects based on machine-learning techniques. *Eur Phys J A*. 2021;57(6):197. <https://doi.org/10.1140/epja/s10050-021-00507-7>.
- [42]: Audouze J., & Vauclair, S. (1980), An Introduction To Nuclear Astrophysics The Formation And The Evolution Of Matter In The Universe, <https://doi.org/10.1007/978-94-009-9477-5>.
- [43]: INFN Catania, <https://www.ct.infn.it/>
- [44]: INFN Laboratori Nazionali del Sud, <https://www.lns.infn.it/>
- [45]: Scintillation properties of solution-grown trans-stilbene single crystals, N. Zaitseva, A. Glenn, L. Carman, H. P. Martinez, R. Hatarik, H. Klapper, S Payne, *Nucl. Instrum. Methods Phys. Res. A* 789 (2015) 8–15 <http://dx.doi.org/10.1016/j.nima.2015.03.090>
- [46]: A. Di Chicco, A. Sardet, M. Petit, R. Jacqmin, V. Gressier, B. Stout, Gamma-response characterization of a solution-grown stilbene based detector assembly in the 59 keV–4.44 MeV energy range; an alternative low-resolution gamma spectrometer, *Nucl. Instrum. Methods Phys. Res. A* 1034 (2022) 166740 <https://doi.org/10.1016/j.nima.2022.166740>
- [47]: InradOptics, INRAD Scintinel™, https://inradoptics.com/pdfs/datasheets/InradOptics_Datasheet_Stilbene_Final.pdf
- [48]: PROTEUS, <https://proteus-pp.com/>
- [49]: PROTEUS, Organic Scintillation Crystals, <http://www.ggg-tech.co.jp/maker/proteus/organics.html>
- [50]: Hamamatsu Photonics, https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/etd/R7378A_TPMH1288E.pdf
- [51]: PS1807 Data Sheet Photomultiplier Power Base (Negative), <https://www.sens-tech.com/wp-content/uploads/2023/03/senstech-PS1807-v1.pdf>
- [52]: CAEN-DT5743, 8 Channel 12bit 3.2 GS/s Switched Capacitor Digitizer <https://www.caen.it/products/dt5743/>
- [53]: SACALC5, Average solid angle (and geometry factor) by a Monte Carlo Method <https://sites.google.com/site/averagesolidangle2/average-solid-angle-and-geometry-factor-by-a-monte-carlo-method>
- [54]: Savitzky, A.; Golay, M.J.E. (1964). "Smoothing and Differentiation of Data by Simplified Least Squares Procedures". *Analytical Chemistry*. 36 (8): 1627–39. Bibcode:1964AnaCh..36.1627S. doi:10.1021/ac60214a047.

- [55]: L. M. Bollinger, G. E. Thomas, Measurement of the Time Dependence of Scintillation Intensity by a Delayed-Coincidence Method, *Rev. Sci. Instrum.* 32, 1044–1050 (1961), <https://doi.org/10.1063/1.1717610>
- [56]: Jolliffe, Ian T.; Cadima, Jorge, "Principal component analysis: a review and recent developments". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences.* 374 (2065): 20150202, <https://doi.org/10.1098/rsta.2015.0202>
- [57]: F. Ferrulli, N. Dinar, L. Gallego Manzano, M. Labalme, M. Silari, Characterization of stilbene and EJ-276 scintillators coupled with a large area SiPM array for a fast neutron dose rate detector, *Nucl. Instrum. Methods Phys. Res. A* 1010 (2021) 165566, <https://doi.org/10.1016/j.nima.2021.165566>
- [58]: I. T. Jolliffe, *Principal Component Analysis*, Second Edition, Springer Series in Statistics, ISBN 0-387-95442-2
- [59]: H. F. Kaiser (1960), The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20 (1), 141-151. <https://doi.org/10.1177/001316446002000116>

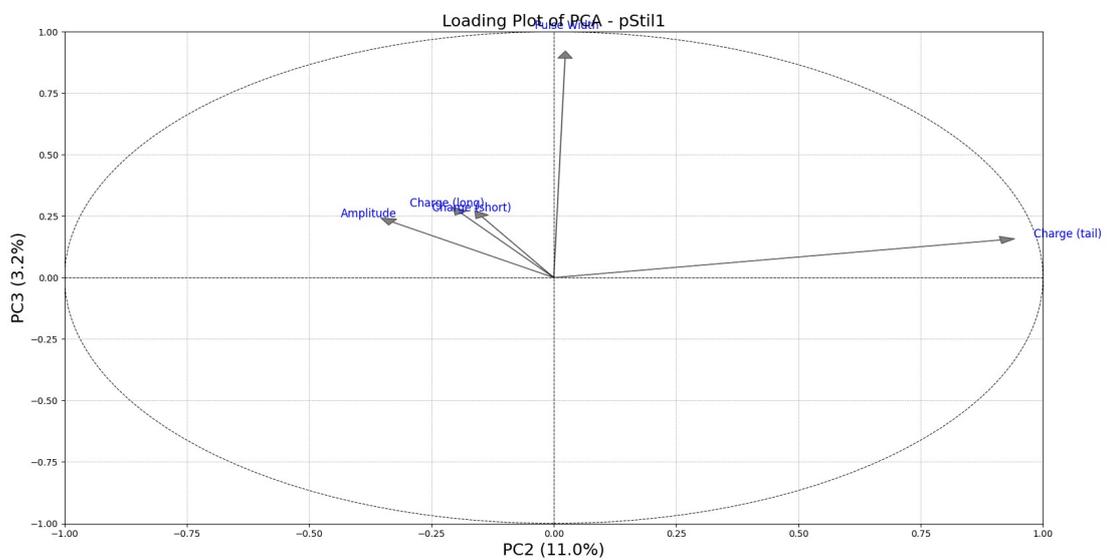
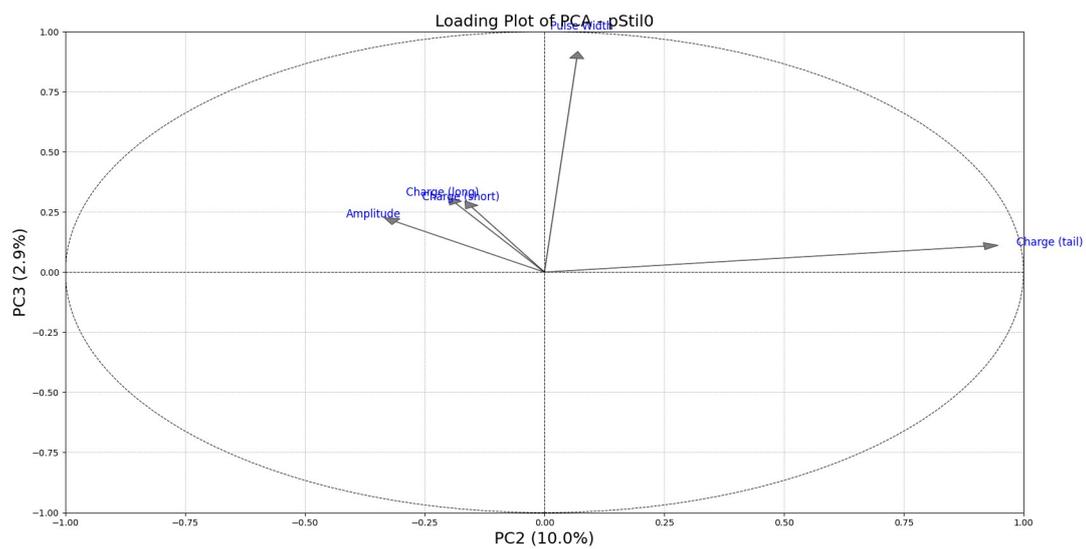
Chapter 8 - Appendices

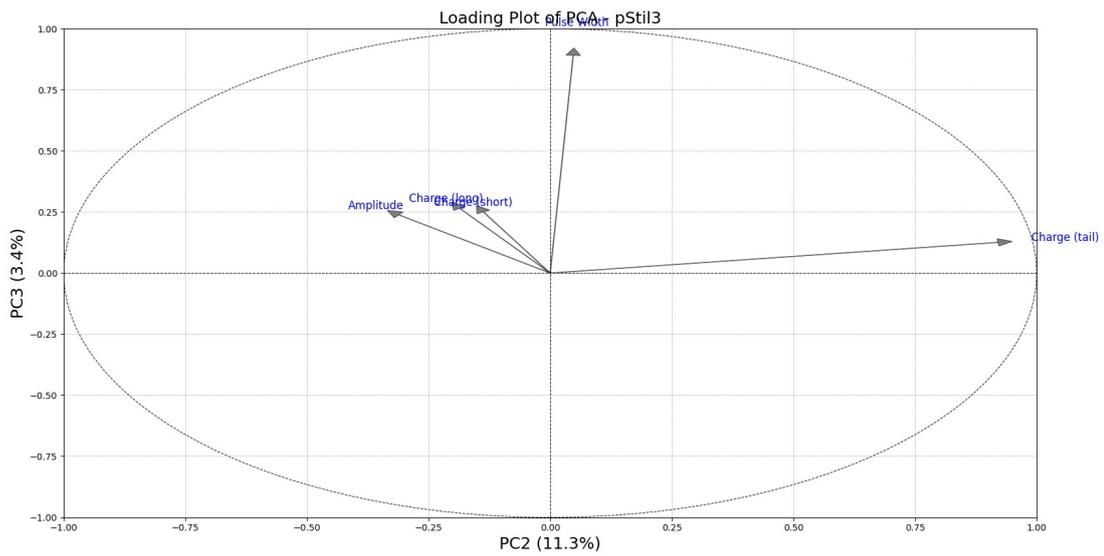
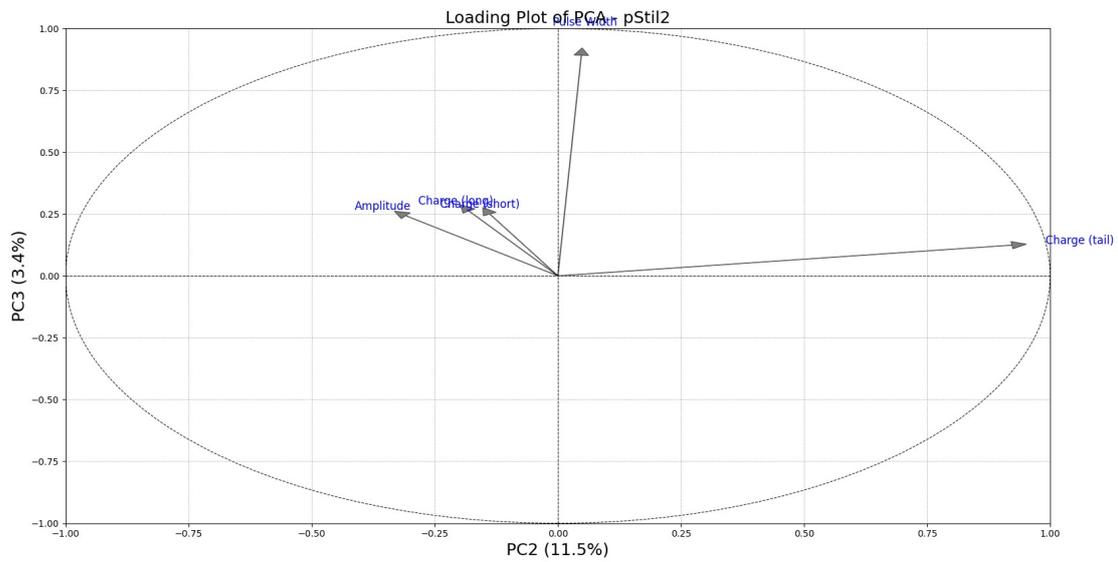
- *Appendix A*
- *Appendix B*
- *Appendix C*

8.1 – Appendix A

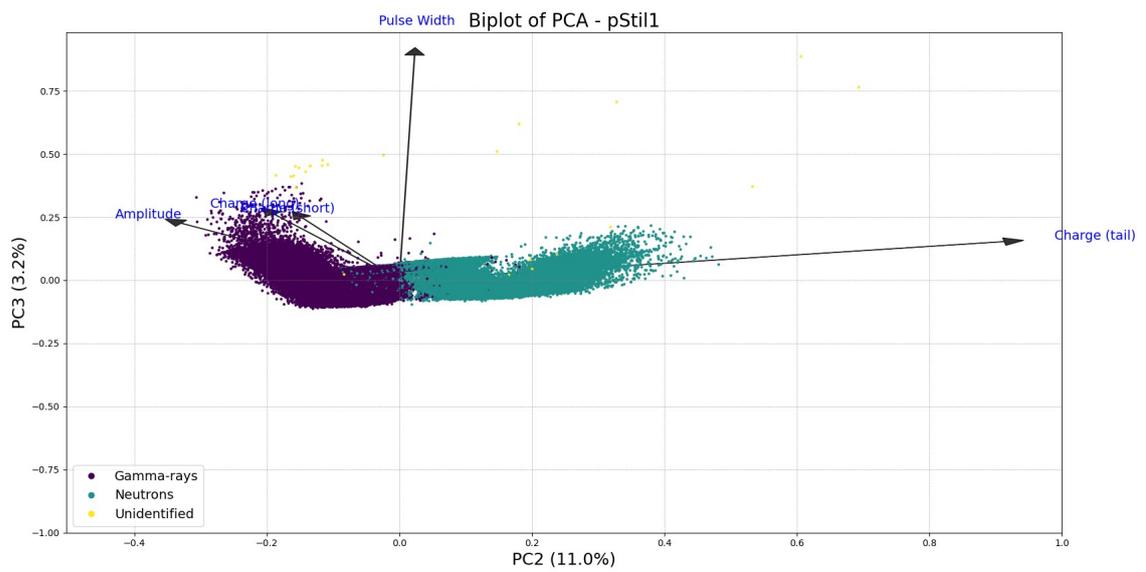
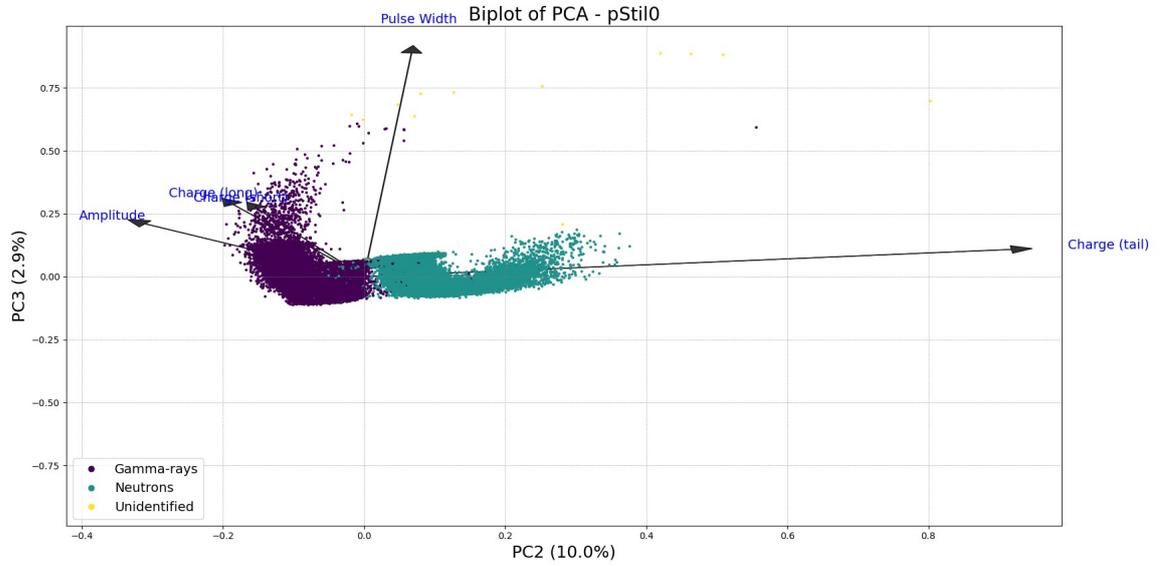
Additional PCA plots:

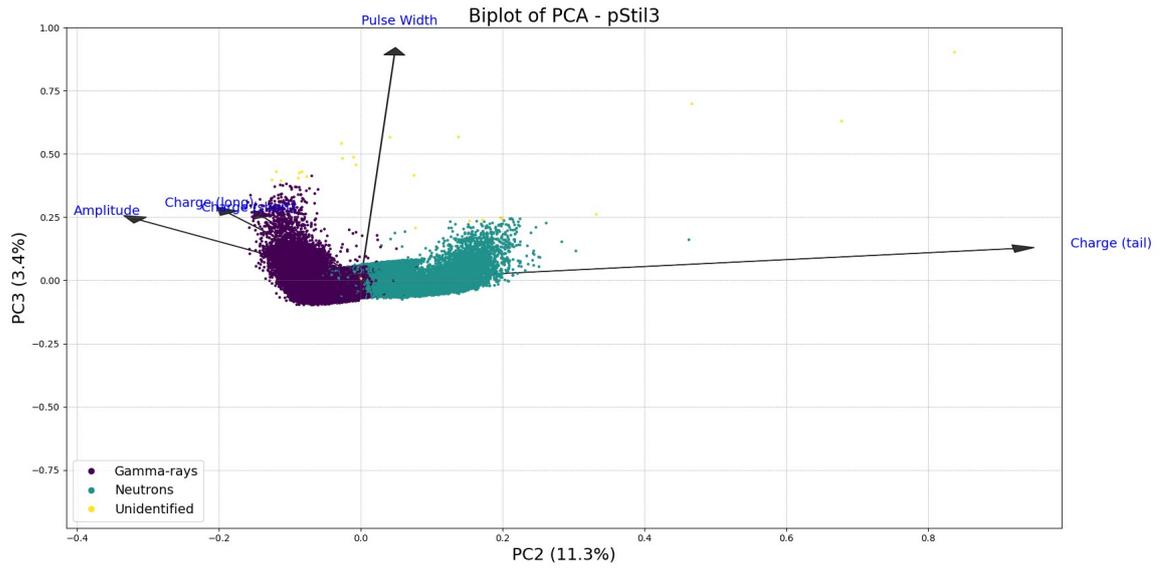
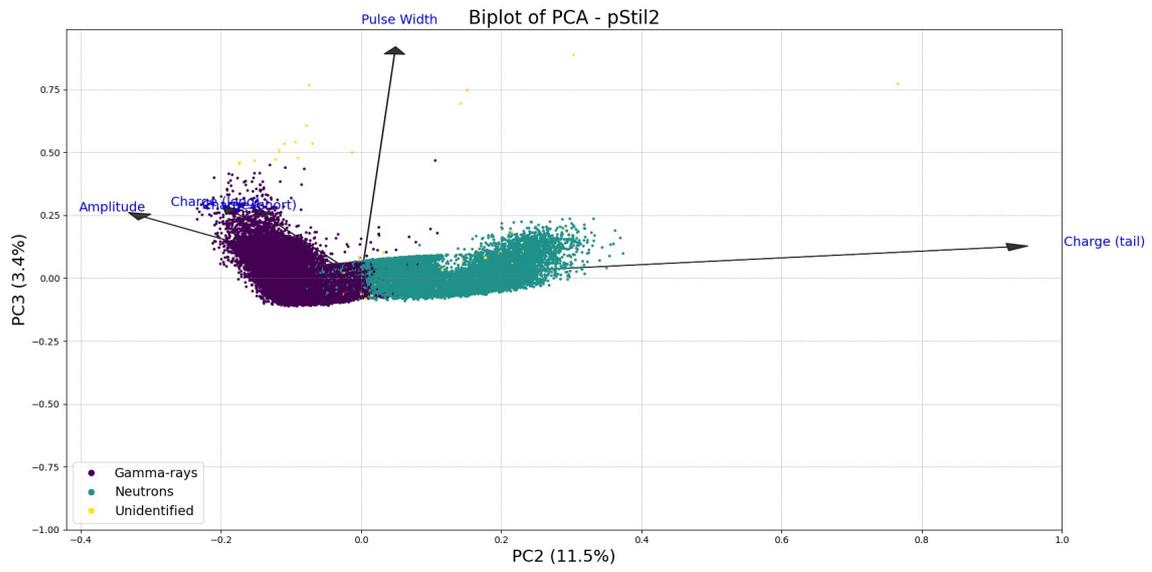
➤ **Loadings plots: PC2 – PC3:**





- **Biplots: PC2 – PC3:**





8.2 – Appendix B

Analysis Related Code Snippets:

- **List of imported libraries:**

```
#System
import os
import os.path
import re
import csv
from __future__ import print_function
#Math - plotting
import math
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from matplotlib.lines import Line2D
from matplotlib.patches import Circle
from sklearn import datasets

from scipy.optimize import fsolve
from scipy.integrate import simps
from scipy.integrate import trapz
from scipy.interpolate import interp1d
from scipy.signal import find_peaks

from lnfit.models import SplineModel, BreitWignerModel, VoigtModel, \
    LinearModel, LorentzianModel, GaussianModel, ExponentialModel
from lnfit import Model, Parameters

##### PCA - libraries #####
# Training set and Testing set
from sklearn.model_selection import train_test_split
# performing preprocessing part
from sklearn.preprocessing import StandardScaler
# Applying PCA function on training
from sklearn.decomposition import PCA
# Fitting Logistic Regression To the training set
from sklearn.linear_model import LogisticRegression
# making confusion matrix
from sklearn.metrics import confusion_matrix
from matplotlib.colors import ListedColormap
##### PCA - Libraries #####

##### Custom Classes #####
from SelectIndices import BrushSelector
from PCA_ import PCA_
##### Custom Classes #####
```

- **Index selection custom class:**

```
from matplotlib.image import AxesImage
from matplotlib.widgets import PolygonSelector
from matplotlib.path import Path
import numpy as np

class BrushSelector:
    def __init__(self, ax, scatter):
        self.ax = ax
        self.scatter = scatter
        self.selected_indices_lists = []
        self.canvas = self.scatter.figure.canvas

        self.press = None
        self.selector = None

        self.cid_press = self.canvas.mpl_connect('button_press_event', self.on_press)
        self.cid_release = self.canvas.mpl_connect('button_release_event', self.on_select)

    def on_press(self, event):
        if event.inaxes != self.scatter.axes:
            return
        if self.selector is None:
            self.selector = PolygonSelector(self.ax, self.on_select)

    def on_select(self, verts):
        try:
            path = Path(verts)
            selected_indices = np.where(path.contains_points(self.scatter.get_offsets()))[0]
            self.selected_indices_lists.append(selected_indices.tolist())
            print("Selected indices:", selected_indices)
            self.selector.set_visible(False)
            self.selector = None
        except TypeError:
            pass
```

- **CAEN-DT5743 Digitizer – Wavecatcher DAQ software: Sorting Routine** (using Regular Expressions)

```

sample_pattern = r'Sampling Period: (\d+\.\d+) ps'
dig_amp_pattern = r'Amplitude:\s+(-?\d+\.\d+)\s+V'
dig_charge_pattern = r'Charge:\s+(-?\d+\.\d+)\s+pC'
dig_baseline_pattern = r'Baseline:\s+(-?\d+\.\d+)\s+V'
dig_leading_time_pattern = r'LeadingEdgeTime:\s+(-?\d+\.\d+)\s+ns'
dig_trailing_time_pattern = r'TrailingEdgeTime:\s+(-?\d+\.\d+)\s+ns'

exp_amp_pattern = r"\n(-?\d+\.\d+(?:\s-?\d+\.\d+)*)"

parent_dir = "/home/los/Master/Am_Be_source_new"

output_files_exp = [f'out_ch{i}.dat' for i in range(4)] # 4 channels
def write_experimental_values() -> None:
    try:
        with open(output_files_exp[0], 'w', encoding='utf-8') as out_ch0, \
            open(output_files_exp[1], 'w', encoding='utf-8') as out_ch1, \
            open(output_files_exp[2], 'w', encoding='utf-8') as out_ch2, \
            open(output_files_exp[3], 'w', encoding='utf-8') as out_ch3:

            for subdir in os.listdir(parent_dir):
                if subdir.startswith('wavecatcher') and os.path.isdir(os.path.join(parent_dir, subdir)):
                    subdir_path = os.path.join(parent_dir, subdir)

                    files = os.listdir(subdir_path)
                    files.sort()

                    for file_path in files:
                        if file_path.startswith("wavecatcher"):
                            full_file_path = os.path.join(subdir_path, file_path)
                            with open(full_file_path, 'r', encoding='utf-8') as file:
                                print(f"File {full_file_path} is being processed")
                                content = file.read()

                                amp_values = re.findall(exp_amp_pattern, content, re.DOTALL)
                                dig_charge_values = re.findall(dig_charge_pattern, content, re.DOTALL)

                                cleaned_amp = [re.findall(r'-?\d+\.\d+', values) for values in amp_values]
                                cleaned_charge = [re.findall(r'-?\d+\.\d+', values) for values in dig_charge_values]

                                for index, (amp_vals, charge_vals) in enumerate(zip(cleaned_amp, cleaned_charge)):

                                    for amp_val, charge_val in zip(amp_vals, charge_vals):

                                        float_item = [float(val) for val in amp_vals]
                                        if min(float_item) < -20*10**(-3):

                                            if index % 4 == 0:
                                                out_ch0.write(' '.join(amp_vals)+'\n')
                                            elif index % 4 == 1:
                                                out_ch1.write(' '.join(amp_vals)+'\n')
                                            elif index % 4 == 2:
                                                out_ch2.write(' '.join(amp_vals)+'\n')
                                            elif index % 4 == 3:
                                                out_ch3.write(' '.join(amp_vals)+'\n')

    except IOError as e:
        print(e)
    except TypeError as e:
        print(e)

```

```

output_files_dig = [f'out_dig_ch{i}.dat' for i in range(4)] # 4 channels
def write_digitizer_values() -> None:
    try:
        with open(output_files_dig[0], 'w', encoding='utf-8') as out_dig_ch0, \
            open(output_files_dig[1], 'w', encoding='utf-8') as out_dig_ch1, \
            open(output_files_dig[2], 'w', encoding='utf-8') as out_dig_ch2, \
            open(output_files_dig[3], 'w', encoding='utf-8') as out_dig_ch3:

            for subdir in os.listdir(parent_dir):
                if subdir.startswith('wavecatcher') and os.path.isdir(os.path.join(parent_dir, subdir)):
                    subdir_path = os.path.join(parent_dir, subdir)

                    files = os.listdir(subdir_path)
                    files.sort()

                    for file_path in files:
                        if file_path.startswith("wavecatcher"):
                            full_file_path = os.path.join(subdir_path, file_path)
                            with open(full_file_path, 'r', encoding = 'utf-8') as file:
                                print(f"File {full_file_path} is being processed")
                                content = file.read()

                                channel_values = re.findall(exp_amp_pattern, content, re.DOTALL)
                                cleaned_list = [re.findall(r'?\d+\.\d+', values) for values in channel_values]

                                dig_amp_values = re.findall(dig_amp_pattern, content, re.DOTALL)
                                dig_charge_values = re.findall(dig_charge_pattern, content, re.DOTALL)
                                dig_leading_time_values = re.findall(dig_leading_time_pattern, content, re.DOTALL)
                                dig_trailing_time_values = re.findall(dig_trailing_time_pattern, content, re.DOTALL)

                                cleaned_amp = [re.findall(r'?\d+\.\d+', values) for values in dig_amp_values]
                                cleaned_charge = [re.findall(r'?\d+\.\d+', values) for values in dig_charge_values]
                                cleaned_leading_time = [re.findall(r'?\d+\.\d+', values) for values in dig_leading_time_values]
                                cleaned_trailing_time = [re.findall(r'?\d+\.\d+', values) for values in dig_trailing_time_values]

                                for index, (amp_vals, charge_vals, leading_time_vals, trailing_time_vals, voltage_vals) \
                                    in enumerate(zip(cleaned_amp, cleaned_charge, cleaned_leading_time, cleaned_trailing_time, cleaned_list)):
                                    for amp_val, charge_val, leading_time_val, trailing_time_val, voltage_val \
                                        in zip(amp_vals, charge_vals, leading_time_vals, trailing_time_vals, voltage_vals):

                                        float_item = [float(val) for val in voltage_vals]
                                        if min(float_item) < -20*10**(-3):

                                            if index % 4 == 0:
                                                out_dig_ch0.write(f"{amp_val} {charge_val} {leading_time_val} {trailing_time_val}\n")
                                            elif index % 4 == 1:
                                                out_dig_ch1.write(f"{amp_val} {charge_val} {leading_time_val} {trailing_time_val}\n")
                                            elif index % 4 == 2:
                                                out_dig_ch2.write(f"{amp_val} {charge_val} {leading_time_val} {trailing_time_val}\n")
                                            elif index % 4 == 3:
                                                out_dig_ch3.write(f"{amp_val} {charge_val} {leading_time_val} {trailing_time_val}\n")

    except IOError as e:
        print(e)
    except TypeError as e:
        print(e)

```

- **Rise/Fall time extraction**

```

def rise_time_all(chunk_size: int=1) -> None:
    try:
        with open('out_exp.dat', 'r', encoding='utf-8') as out:
            av_rise_time = []
            while True:
                try:
                    lines = [next(out) for _ in range(chunk_size)]
                except StopIteration:
                    break
                voltage = []
                for line in lines:
                    values = [float(value) for value in line.split()]
                    voltage.append(values)
                for i in range(len(voltage)):
                    volts = np.array([x * 10 ** 3 for x in voltage[i]]) # mV
                    time = np.array(np.arange(0, sample_period*len(voltage[i])*10**-3, sample_period*10**-3))

                    peak_idx = np.argmin(volts)
                    try:
                        # Implement savgol filtering for low SNRs
                        savgol_filter_volts = savgol_filter(volts, window_length=9, polyorder=3)

                        volts_before_peak = np.array(savgol_filter_volts[peak_idx-20:peak_idx])
                        time_before_peak = np.array(time[peak_idx-20:peak_idx])

                        rise_start = 0.1 * np.min(savgol_filter_volts)
                        rise_end = 0.9 * np.min(savgol_filter_volts)

                        rise_start_idx = np.argmin(np.abs(volts_before_peak - rise_start))
                        rise_end_idx = np.argmin(np.abs(volts_before_peak - rise_end))

                        volts_for_search_start = volts_before_peak[ rise_start_idx - 2 : rise_start_idx + 2 ]
                        time_for_search_start = time_before_peak[ rise_start_idx - 2 : rise_start_idx + 2 ]

                        volts_for_search_end = volts_before_peak[ rise_end_idx - 2 : rise_end_idx + 2 ]
                        time_for_search_end = time_before_peak[ rise_end_idx - 2 : rise_end_idx + 2 ]

                        interp_func_start = interp1d(volts_for_search_start, time_for_search_start, kind='linear', fill_value='extrapolate')
                        interp_func_end = interp1d(volts_for_search_end, time_for_search_end, kind='linear', fill_value='extrapolate')

                        interp_rise_start, interp_rise_end = interp_func_start(rise_start), interp_func_end(rise_end)

                        rise_time = interp_rise_end - interp_rise_start
                        av_rise_time.append(rise_time)
                        print(f'Event: {len(av_rise_time)}')

                    except ValueError:
                        pass

            except IOError as e:
                print(e)

        pm_unicode = "±"
        average = np.mean(av_rise_time)
        std_dev = np.std(av_rise_time, ddof=1)
        sem = std_dev / np.sqrt(len(av_rise_time))
        print(f'Rise Time average: {average:.4f} {pm_unicode} {sem:.4f} ns')

```

```

def fall_time_all(chunk_size: int=1) -> None:
    try:
        with open('out_exp.dat', 'r', encoding='utf-8') as out:
            av_fall_time = []
            while True:
                try:
                    lines = [next(out) for _ in range(chunk_size)]
                except StopIteration:
                    break
                voltage = []
                for line in lines:
                    values = [float(value) for value in line.split()]
                    voltage.append(values)
                for i in range(len(voltage)):
                    volts = np.array([x * 10 ** 3 for x in voltage[i]]) # mV
                    time = np.array(np.arange(0, sample_period*len(voltage[i])*10**-3, sample_period*10**-3))

                    peak_idx = np.argmin(volts)
                    try:
                        # Implement savgol filtering for low SNRs
                        savgol_filter_volts = savgol_filter(volts, window_length=9, polyorder=3)

                        volts_after_peak = np.array(savgol_filter_volts[peak_idx:peak_idx+30])
                        time_after_peak = np.array(time[peak_idx:peak_idx+30])

                        fall_start = 0.9 * np.min(savgol_filter_volts)
                        fall_end = 0.1 * np.min(savgol_filter_volts)

                        fall_start_idx = np.argmin(np.abs(volts_after_peak - fall_start))
                        fall_end_idx = np.argmin(np.abs(volts_after_peak - fall_end))

                        volts_for_search_start = volts_after_peak[ fall_start_idx - 2 : fall_start_idx + 2 ]
                        time_for_search_start = time_after_peak[ fall_start_idx - 2 : fall_start_idx + 2 ]

                        volts_for_search_end = volts_after_peak[ fall_end_idx - 2 : fall_end_idx + 2 ]
                        time_for_search_end = time_after_peak[ fall_end_idx - 2 : fall_end_idx + 2 ]

                        interp_func_start = interp1d(volts_for_search_start, time_for_search_start, kind='linear', fill_value='extrapolate')
                        interp_func_end = interp1d(volts_for_search_end, time_for_search_end, kind='linear', fill_value='extrapolate')

                        interp_fall_start, interp_fall_end = interp_func_start(fall_start), interp_func_end(fall_end)

                        fall_time = interp_fall_end - interp_fall_start
                        av_fall_time.append(fall_time)
                        print(f'Event: {len(av_fall_time)}')

                    except ValueError:
                        pass

            except IOError as e:
                print(e)

    pm_unicode = "±"
    average = np.mean(av_fall_time)
    std_dev = np.std(av_fall_time, ddof=1)
    sem = std_dev / np.sqrt(len(av_fall_time))
    print(f'Fall Time average: {average:.3f} {pm_unicode} {sem:.3f} ns')

```

- **PSD + FOM calculation:**

```

def psd_segmented_FOM_double_gauss(ChannelNo: int, plot: bool=True) -> list:
    with open(f'charge_slow_Am_Be_ch{ChannelNo}.dat', 'r', encoding = 'utf-8') as out_slow, \
         open(f'charge_fast_Am_Be_ch{ChannelNo}.dat', 'r', encoding = 'utf-8') as out_fast, \
         open(f'charge_tail_Am_Be_ch{ChannelNo}.dat', 'r', encoding = 'utf-8') as out_tail, \
         open(f'out_dig_ch{ChannelNo}.dat', 'r', encoding = 'utf-8') as out_time, \
         open(f'pulse_height_Am_Be_ch{ChannelNo}.dat', 'r', encoding = 'utf-8') as out_pulse:
        slow = [-float(line.strip()) for line in out_slow.readlines()]
        fast = [-float(line.strip()) for line in out_fast.readlines()]
        tail = [-float(line.strip()) for line in out_tail.readlines()]
        pulse_height = [-float(line.strip()) for line in out_pulse.readlines()]
        amp_dig = []
        charge_dig = []
        leading_time = []
        trailing_time = []
        for line in out_time.readlines():
            point = line.strip().split()
            amp_dig.append( -float(point[0])*10**3 )
            charge_dig.append( -float(point[1]) )
            leading_time.append( float(point[2]) )
            trailing_time.append( float(point[3]) )

    slow = np.array(slow)
    fast = np.array(fast)
    tail = np.array(tail)
    pulse_height = np.array(pulse_height)
    leading_time = np.array(leading_time)
    trailing_time = np.array(trailing_time)
    amp_dig = np.array(amp_dig)
    charge_dig = np.array(charge_dig)
    # Implement a threshold
    energy_from_charge = np.array( [ (slope_charge * Q + intercept_charge) for Q in slow if Q<300 ] )
    energy_from_amp = np.array( [ (slope_amp * V + intercept_amp) for V, Q in zip(pulse_height, slow) if Q<300 ] )
    # from MeV -> MeVee
    electron_eq_energy = np.array( [ (-0.000583965*E**3 + 0.0252841*E**2 + 0.206746*E - 0.00303) \
                                     for E in energy_from_amp ] )

    PSD = np.array( [ x/y for x, y in zip(pulse_height, slow) if y<300 ] )
    pulse_width = np.array( [ x-y for x,y in zip(trailing_time, leading_time) ] )
    ''' Adjust the window and starting energy '''
    window = 0.15
    num_of_segments = 6
    start = 0.038

    stop = start + window * num_of_segments
    vert_start = np.arange(start, stop, window)
    FOM_PSD_values = []
    energy_eq_values = []
    err_FOM_PSD_values = []

    for step in vert_start:
        vert_start = lambda y: step
        vert_stop = lambda y: step + window
        top_line = [2.9, 3.2, 3.45, 3.8]
        horiz_top = lambda x: top_line[ChannelNo]
        horiz_bottom = lambda x: 1.6
        y_vals_vert_start = np.linspace(-50, 50, len(PSD))
        y_vals_vert_stop = np.linspace(-50, 50, len(PSD))
        x_vals_horiz_top = np.linspace(-50, 1000, len(PSD))
        x_vals_horiz_bottom = np.linspace(-50, 1000, len(PSD))
        x_vals_vert_start = [vert_start(y) for y in y_vals_vert_start]
        x_vals_vert_stop = [vert_stop(y) for y in y_vals_vert_stop]
        y_vals_horiz_top = [horiz_top(x) for x in x_vals_horiz_top]
        y_vals_horiz_bottom = [horiz_bottom(x) for x in x_vals_horiz_bottom]

        condition_ind = np.where( (electron_eq_energy >= x_vals_vert_start) & (electron_eq_energy <= x_vals_vert_stop) \
                                 & (PSD <= y_vals_horiz_top) & (PSD >= y_vals_horiz_bottom) )

        if plot:
            plt.plot(x_vals_vert_start, y_vals_vert_start, color='blue', alpha=0.8, linewidth=1, linestyle='--')
            plt.plot(x_vals_vert_stop, y_vals_vert_stop, color='red', alpha=0.8, linewidth=1, linestyle='--')
            plt.plot(x_vals_horiz_top, y_vals_horiz_top, color='blue', alpha=0.8, linewidth=1, linestyle='--')
            plt.plot(x_vals_horiz_bottom, y_vals_horiz_bottom, color='red', alpha=0.8, linewidth=1, linestyle='--')
            # ===== Plotting ===== #
            plt.hexbin(electron_eq_energy, PSD, gridsize=5000, bins=len(PSD)/10, \
                       vmin=0, vmax=500, cmap='viridis', mincnt=1, edgecolors='face')
            plt.colorbar()
            plt.xlabel('Electron-equivalent energy (MeV_{\mathrm{ee}})', fontsize=18)
            plt.ylabel('PSD: Pulse-Height/$Q_{\mathrm{Total}}$', fontsize=18)
            plt.title(f'PSD - pStil{ChannelNo}', fontsize=18)
            plt.legend(fontsize=18)
            plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
            plt.ylin(top=6)
            plt.ylin(bottom=0)
            plt.xlin(right=2)
            plt.xlin(left=-0.5)
            if plot:
                plt.show()

```

```
#####FOM#####
FOM_values = PSD[condition_ind]
if plot:
    y_data_orig, bins, _ = plt.hist(FOM_values, bins=int(math.sqrt(len(FOM_values)) + 50), histtype='step', alpha=1, visible=True)

y_data_orig, bins = np.histogram(FOM_values, bins=int(math.sqrt(len(FOM_values)) + 50) )
gauss1 = GaussianModel(prefix='g1_')
gauss2 = GaussianModel(prefix='g2_')

peaks, _ = find_peaks(y_data_orig, height=(min(y_data_orig) + max(y_data_orig)) / 2)
peak_positions = bins[:-1][peaks]
peak_positions_sorted = sorted(peak_positions, reverse=True)[:2]

amplitude_guess = max(y_data_orig)
sigma_guess = 0.1
mid_point = (min(bins) + max(bins)) / 2
pars = gauss1.make_params(center=dict(value=peak_positions_sorted[0], min=min(bins), max=mid_point),
                           sigma=dict(value=sigma_guess, min=0),
                           amplitude=dict(value=amplitude_guess, min=0))
pars.update(gauss2.make_params(center=dict(value=peak_positions_sorted[0], min=mid_point, max=max(bins)),
                                sigma=dict(value=sigma_guess, min=0),
                                amplitude=dict(value=amplitude_guess, min=0)))

mod = gauss1 + gauss2

x_data = bins[:-1]
y_data = y_data_orig
init = mod.eval(pars, x=x_data)
out = mod.fit(y_data, pars, x=x_data)
best_fit = out.best_fit
if plot:
    print(out.fit_report(correl_mode='table'))

mean_val1 = out.params['g1_center'].value
mean_val2 = out.params['g2_center'].value
FWHM1 = out.params['g1_fwhm'].value
FWHM2 = out.params['g2_fwhm'].value
mean_val1_err = out.params['g1_center'].stderr
mean_val2_err = out.params['g2_center'].stderr
FWHM1_err = out.params['g1_fwhm'].stderr
FWHM2_err = out.params['g2_fwhm'].stderr
u = FWHM1 + FWHM2
FOM = abs(mean_val2-mean_val1) / u
FOM_err = (1/u) * math.sqrt( mean_val1_err**2 + mean_val2_err**2 + \
                            ((mean_val2-mean_val1)/u)**2) * (FWHM1_err**2 + FWHM2_err**2) )

pm_unicode = "±"
FOM_PSD_values.append(FOM)
err_FOM_PSD_values.append(FOM_err)
energy_eq_values.append((vert_stop(0)+vert_start(0))/2)
if plot:
    # ===== Plotting ===== #
    plt.plot(x_data, best_fit, color='red')
    plt.axvline(mean_val1, color='red', linestyle='--', linewidth=1)
    plt.axvline(mean_val2, color='blue', linestyle='--', linewidth=1)
    legend_line1 = Line2D([0], [0], color='red', linestyle='--', label=f'Mean 1: {mean_val1:.2f}')
    legend_line2 = Line2D([0], [0], color='blue', linestyle='--', label=f'Mean 2: {mean_val2:.2f}')
    plt.legend(handles=[legend_line1, legend_line2], loc='upper right', fontsize=18)

    plt.text(0.92, 0.82, f'FOM: {FOM:.2f} {pm_unicode} {FOM_err:.2f}', ha='center', va='center', \
             fontsize=18, transform=plt.gca().transAxes, bbox=dict(facecolor='white', alpha=0.8))
    plt.text(0.92, 0.72, f'{int(start*1e3)} - {int(stop*1e3)} keVee', ha='center', va='center', \
             fontsize=18, transform=plt.gca().transAxes, bbox=dict(facecolor='white', alpha=0.8))
    plt.title(f'FOM - pStil{ChannelNo}', fontsize=18)
    plt.xlabel('PSD', fontsize=18)
    plt.ylabel('Counts', fontsize=18)
    #plt.xlim(right=0.8)
    #plt.ylim(top=60)
    plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
    if plot:
        plt.show()

return FOM_PSD_values, energy_eq_values, err_FOM_PSD_values
```

- **FOM as a function of energy:**

```
def FOM_func(ChannelNo: int, n_degree: int, co_plot: bool = False, fit: bool = True) -> None:
    colors = ['blue', 'red', 'orange', 'green']
    fit_funcs = []
    if co_plot:
        for i in range(0,4): # number of channels
            FOM_PSD_values, energy_eq_values, err_FOM_PSD_values = psd_segmented_FOM_double_gauss(i, plot=False)
            plt.errorbar(energy_eq_values, FOM_PSD_values, yerr=err_FOM_PSD_values, \
                        fmt='o', capsize=5, label=f'pStil{i}', color=f'{colors[i]}') #, linestyle='--'

            if fit:
                # Fit a polynomial function
                coeffs = np.polyfit(energy_eq_values, FOM_PSD_values, n_degree)
                poly_func = np.poly1d(coeffs)
                result = poly_func(energy_eq_values)

                plt.plot(energy_eq_values, result, '--', color=f'{colors[i]}')
                # Print the polynomial expression
                expression = f'f_ch{i}(E) = '
                for j in range(n_degree+1):
                    expression += f'{coeffs[j]:.3f} * t^{n_degree-j}'
                    if j != n_degree:
                        expression += ' + '
                fit_funcs.append(expression)

            # Print expressions for the current fit
            for expr in fit_funcs:
                print(expr)
            plt.xlabel('Electron-equivalent energy ($MeV_{\mathrm{ee}}$)', fontsize=18)
            plt.ylabel('PSD FOM', fontsize=18)
            plt.legend(fontsize=18)
            plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
            plt.show()
    else:
        FOM_PSD_values, energy_eq_values, err_FOM_PSD_values = psd_segmented_FOM_double_gauss(ChannelNo, plot=False)
        plt.errorbar(energy_eq_values, FOM_PSD_values, yerr=err_FOM_PSD_values, fmt='o', capsize=5, label=f'pStil{ChannelNo}')
        if fit:
            # Fit a polynomial function
            coeffs = np.polyfit(energy_eq_values, FOM_PSD_values, n_degree)
            poly_func = np.poly1d(coeffs)
            result = poly_func(energy_eq_values)

            plt.plot(energy_eq_values, '--', result) #
            plt.xlabel('Electron-equivalent energy ($MeV_{\mathrm{ee}}$)', fontsize=18)
            plt.ylabel('PSD FOM', fontsize=18)
            plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
            plt.legend(fontsize=18)
            plt.show()
```

- **PCA – Scree plots, Loadings plots, Biplots:**

```

def principal_component_analysis(ChannelNo: int, PC_No1: int, PC_No2: int):
    gamma_neutron_ind = write_to_csv_format_for_pca(ChannelNo, pile_up=True)
    csv_file = f"pca_data_ch{ChannelNo}.csv"
    par_names = ['Amplitude', 'Charge (long)', 'Charge (short)', 'Charge (tail)', 'Pulse Width']

    df = pd.read_csv(csv_file)
    df = pd.DataFrame(data=df, columns=par_names)
    ''' Initializes a StandardScaler object, which is used to standardize the data '''
    scaler = StandardScaler(with_mean=True, with_std=True)
    ''' Scales the data using the fitted scaler, standardizing each feature by removing the mean and scaling to unit variance '''
    data_scaled = scaler.fit_transform(df)

    df_scaled = pd.DataFrame(data=data_scaled, columns=par_names)
    ''' Initializes a PCA object with n principal components '''
    pca = PCA(n_components=len(par_names))
    ''' Fits the PCA model to the scaled data and transforms it into principal components '''
    PC = pca.fit_transform(df_scaled)

    prop_var = pca.explained_variance_ratio_
    eigenvalues = pca.explained_variance_

    PC_numbers = np.arange(pca.n_components_) + 1
    ''' Scree plots '''
    plt.figure(figsize=(10, 6))
    # Scree plot (Elbow method)
    plt.subplot(1, 2, 1)
    plt.plot(PC_numbers, prop_var, 'ro-')
    plt.title(f'Scree Plot (Elbow method) - pStil{ChannelNo}', fontsize=14)
    plt.ylabel('Proportion of Variance', fontsize=18)
    plt.xlabel('Component Number', fontsize=18)
    plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)

    # Scree Plot (Kaiser Rule)
    plt.subplot(1, 2, 2)
    plt.plot(PC_numbers, eigenvalues, 'ro-')
    plt.title(f'Scree Plot (Kaiser Rule) - pStil{ChannelNo}', fontsize=14)
    plt.ylabel('Variance', fontsize=18)
    plt.xlabel('Component Number', fontsize=18)
    plt.axhline(y=1, color='red', linestyle='--')
    plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)

    plt.tight_layout()
    plt.show()

    pca = PCA(n_components=5)
    PC = pca.fit_transform(df_scaled)

    pca_data = pd.DataFrame(data = PC,
                            columns = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5'])

    ''' Check for reversed components & correct them '''
    for i in range(len(pca.components_)):
        if np.sum(pca.components_[i]) < 0:
            pca.components_[i] *= -1

```

```

''' Loadings Plots '''
plt.figure(figsize=(10, 10))
for i, (component, par_name) in enumerate(zip(pca.components_.T, par_names)):
    plt.arrow(0, 0,
              component[PC_No1],
              component[PC_No2],
              head_width=0.03,
              head_length=0.03,
              color='black',
              alpha=0.5)
    plt.text(component[PC_No1] * 1.15,
             component[PC_No2] * 1.15,
             par_name,
             fontsize=12,
             color='blue',
             ha='center',
             va='center')

plt.title(f'Loading Plot of PCA - pStil{ChannelNo}', fontsize=18)
plt.xlabel(f"PC{PC_No1+1} ({prop_var[PC_No1]*100:.1f}%)", fontsize=18)
plt.ylabel(f"PC{PC_No2+1} ({prop_var[PC_No2]*100:.1f}%)", fontsize=18)
plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
plt.xlim(-1, 1)
plt.ylim(-1, 1)

plt.axhline(y=0, linestyle='--', linewidth=0.8, color='black', alpha=0.9)
plt.axvline(x=0, linestyle='--', linewidth=0.8, color='black', alpha=0.9)

# Circle
ax = plt.gca()
circle = Circle((0, 0), 1, linestyle='--', linewidth=0.8, alpha=0.9, edgecolor='black', fill=False)
ax.add_patch(circle)

plt.show()

''' Define target groups '''
''' Initialize an array for target groups '''
target_groups = np.arange(len(pca_data))
gamma_ind = gamma_neutron_ind[0]
neutron_ind = gamma_neutron_ind[1]

''' Assign label for group 1 - Gammas '''
target_groups[gamma_ind] = 0
''' Assign label for group 2 - Neutrons '''
target_groups[neutron_ind] = 1
''' Assign labels for the rest of the indices - Other particles '''
gamma_neutron_ind = np.concatenate( (gamma_ind, neutron_ind) )

other_indices = np.setdiff1d(target_groups, gamma_neutron_ind)
target_groups[other_indices] = 2

```

```
##### user-defined biplot() #####

def biplot(score, coef, labels=None, target_groups=None):

    xs = score[:,PC_No1]
    ys = score[:,PC_No2]

    n = coef.shape[0]
    scalex = 1.0/(xs.max() - xs.min())
    scaley = 1.0/(ys.max() - ys.min())

    for i in range(n):
        plt.arrow(0, 0, coef[i,PC_No1],
                 coef[i,PC_No2],
                 head_width=0.03,
                 head_length=0.03,
                 color = 'black',
                 alpha = 0.5)

        plt.text(coef[i,PC_No1] * 1.15,
                 coef[i,PC_No2] * 1.15,
                 labels[i],
                 fontsize=16,
                 color = 'blue',
                 ha = 'center',
                 va = 'center')

    scatter = plt.scatter(xs * scalex,
                          ys * scaley,
                          c=target_groups,
                          cmap='viridis',
                          s=4)

    particle_names = { 0: 'Gamma-rays', 1: 'Neutrons', 2: 'Unidentified' }

    ''' Get handles and default labels using legend_elements '''
    handles, default_labels = scatter.legend_elements()
    ''' Define custom labels based on particle names '''
    custom_labels = [particle_names[label] for label in np.unique(target_groups)]

    plt.xlabel(f"PC{PC_No1+1} ({prop_var[PC_No1]*100:.1f}%)", fontsize=18)
    plt.ylabel(f"PC{PC_No2+1} ({prop_var[PC_No2]*100:.1f}%)", fontsize=18)
    plt.title(f'Biplot of PCA - pStil{ChannelNo}', fontsize=18)
    plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)
    plt.legend(handles,
              custom_labels,
              loc="lower left",
              fontsize=18)

    plt.xlim(-1, 1)
    plt.ylim(-1, 1)
    plt.show()

##### user-defined biplot() #####

biplot(PC, pca.components_.T, par_names, target_groups)
```

8.3 – Appendix C

GEANT4 Related Code Snippets:

- **Detector Construction & Placement:**

```
G4double D_Stilbene_OutR = (2.54/2.)*cm; // 1x1 : diameter - height
G4double D_Stilbene_InR = 0.*cm;
G4double D_Stilbene_Halfh = (2.54/2.)*cm;
G4double SD_dist = 10.*cm;

G4Tubs *D_Stilbene_tube
= new G4Tubs("D_Stilbene_tube",
            D_Stilbene_InR,
            D_Stilbene_OutR,
            D_Stilbene_Halfh,
            startFi, endFi);

G4LogicalVolume *D_Stilbene_log = new G4LogicalVolume(D_Stilbene_tube,
                                                    D_Stilbene,
                                                    "D_Stilbene_log",
                                                    0, 0, 0);

G4VPhysicalVolume *D_Stilbene_phys = new G4PVPlacement(0,
                                                    G4ThreeVector(0.*cm, 0.*cm, SD_dist+D_Stilbene_Halfh),
                                                    D_Stilbene_log,
                                                    "D_Stilbene_tube",
                                                    World_log,
                                                    false,0);
```

- **Generation of Primaries
Default Generator:**

```
PrimaryGeneratorAction::PrimaryGeneratorAction()
: G4VUserPrimaryGeneratorAction(),
  fParticleGun(0)

{
  G4int n_particle = 1;
  fParticleGun = new G4ParticleGun(n_particle);

  G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();
  G4ParticleDefinition* particle = particleTable->FindParticle("neutron");

  fParticleGun->SetParticleDefinition(particle);
  fParticleGun->SetParticleEnergy(5*MeV);
  fParticleGun->SetParticlePosition(G4ThreeVector(0.*cm,0.*cm,0.*cm));

  fParticleGun->SetParticleMomentumDirection(G4ThreeVector(1.,0.,0.));
}

PrimaryGeneratorAction::~~PrimaryGeneratorAction()
{
  delete fParticleGun;
}
```

Isotropic point-source creation (with randomization):

```

void PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    // Randomized momentum
    G4double cosTheta = 2*G4UniformRand() - 1., phi = 2*CLHEP::pi*G4UniformRand();
    G4double sinTheta = std::sqrt(1. - cosTheta*cosTheta);
    G4double u_x = sinTheta*std::cos(phi),
             u_y = sinTheta*std::sin(phi),
             u_z = cosTheta;

    fParticleGun->SetParticleMomentumDirection(G4ThreeVector(u_x, u_y, u_z));

    // Randomized position
    G4double x0=4*cm, y0=4*cm, z0=4*cm;

    G4double rho=5./2.*mm;
    G4double dz0=0.65/2.*mm;

    back:
    x0 = rho*(1.-2.*G4UniformRand());
    y0 = rho*(1.-2.*G4UniformRand());
    if ((x0*x0+y0*y0)>(rho*rho)) goto back;
    z0 = dz0*(G4UniformRand()-0.5);
    fParticleGun->SetParticlePosition(G4ThreeVector(x0,y0,z0));

    fParticleGun->GeneratePrimaryVertex(anEvent); //create vertex
}

```

- **Event Action – ROOT Analysis Manager:**

```

EventAction::EventAction()
{}

EventAction::~EventAction()
{}

void EventAction::BeginOfEventAction(const G4Event* evt)
{
    // resetting energy accumulator...
    Depos = 0.0*MeV;
}

void EventAction::EndOfEventAction(const G4Event* evt)
{
    if ((evt->GetEventID()+1) % 5000 == 0)
        G4cout << ">>> Event " << evt->GetEventID()+1 << G4endl;

    evtNo = evt->GetEventID();

    //FILE *opf1 = fopen("out2.dat", "a");

    if (Depos>0.)
    {
        G4RootAnalysisManager* analysisManager = G4RootAnalysisManager::Instance();
        analysisManager->FillNtupleIColumn(0, 0, evtNo+1);
        analysisManager->FillNtupleIColumn(0, 1, ParticleType);
        analysisManager->FillNtupleDColumn(0, 2, Depos/MeV);
        analysisManager->AddNtupleRow(0);
    }

    //fclose(opf1);
}

```

- **Stepping Action – “.root” output file:**

```
void SteppingAction::UserSteppingAction(const G4Step* aStep)
{
    // collect the energy deposited in the absorbers

    const G4VPhysicalVolume *currentVolume1 = aStep->GetPreStepPoint()->GetPhysicalVolume();
    const G4ParticleDefinition *particle = aStep->GetTrack()->GetParticleDefinition();

    G4Track *aTrack = aStep->GetTrack();

    //for Trans-Stilbene
    if (currentVolume1!=NULL) {
        if (aStep->GetTotalEnergyDeposit(>0) {
            if (currentVolume1->GetName() == "H_Stilbene_tube" ) {
                Depos += aStep->GetTotalEnergyDeposit();
            }
        }
    }

    if ( particle==G4Gamma::Definition() && ( currentVolume1->GetName() == "H_Stilbene_tube" ) )
        ParticleType = 1;

    if ( particle==G4Neutron::Definition() && ( currentVolume1->GetName() == "H_Stilbene_tube" ) )
        ParticleType = 2;

    else
        ParticleType = 3;
```


Papanikolaou Dimitrios

Postgraduate Student

Master of Sciences Thesis

Department of Physics, University of Ioannina
