

Neutron Detector Signal Processing to Calculate the Effective Neutron Multiplication Factor of Subcritical Assemblies

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information
Service 5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

Neutron Detector Signal Processing to Calculate the Effective Neutron Multiplication Factor of Subcritical Assemblies

By

Alberto Talamo and Yousry Gohar
Nuclear Engineering Division, Argonne National Laboratory

June 2016

NEUTRON DETECTOR SIGNAL PROCESSING TO CALCULATE THE EFFECTIVE NEUTRON MULTIPLICATION FACTOR OF SUBCRITICAL ASSEMBLIES

Table of Contents

	<u>Page</u>
1 Introduction.....	2
2 Neutron Detectors	3
3 Gaseous Ionization Detectors Classification	5
4 Gaseous Ionization Detectors Electronic Components.....	6
5 Dead-Time, Noise, Pile-Up, and Wall Effects	7
5.A. Dead-Time Effect	7
5.B. Electronic Noise Effect	10
5.C. Pile-up Effect	11
5.D. Wall Effect	11
6 Steady State Neutron Source Operation	11
6.A. Average Neutron Counts per Second	11
6.B. Rossi Distribution	12
6.C. Feynman Distribution	14
6.D. The Modified Source Multiplication (MSM) Method	16
7 Pulsed External Neutron Source Operation.....	18
8 MCNP Simulations	21
9 Correction Factors for the Experimental Measurements	25
10 Conclusions.....	26
References	28

NEUTRON DETECTOR SIGNAL PROCESSING TO CALCULATE THE EFFECTIVE NEUTRON MULTIPLICATION FACTOR OF SUBCRITICAL ASSEMBLIES

List of Figures

	<u>Page</u>
Figure 1. Illustration of the active zone of a typical gaseous ionization detector; the cathode (hidden in the figure) consists of a metallic hollow cylinder confining the ionizing gas	31
Figure 2. Illustration of the cross section of the active zone of a typical fission chamber; the ionizing gas, fissile, and metallic materials have yellow, red and blue colors, respectively	31
Figure 3. Variation of the collected charge as a function of the applied voltage in a gaseous ionization detector	32
Figure 4. Amplifier output voltage as a function of time analyzed by a 15 channels MCA	33
Figure 5. Illustration of the dead-time effect; <i>signal</i> in the time scale indicates a neutron event (ionization); in the non-paralyzable model, the detector scores <i>signal(j)</i> and <i>signal(j+2)</i> because <i>signal(j+2)</i> is greater than <i>signal(j)+δ</i> ; in the paralyzable model, the detector only scores <i>signal(j)</i> because <i>signal(j+2)</i> is smaller than <i>signal(j+1)+δ</i> ; in both models the detector does not score <i>signal(j+1)</i> because it is smaller than <i>signal(j)+δ</i>	34
Figure 6. Illustration of the dead-time effect for δ equal to 3.5 and 50 μs when the average intensity of the neutron source is constant over time (Eqs. 1 and 2)	34
Figure 7. Illustration of the pile-up effect in the preamplifier output voltage as a function of time	35
Figure 8. Pulse-height spectrum for a ^3He gaseous ionization chamber operated in pulse mode; the MCA has 1000 channels.....	35
Figure 9. Illustration of the Rossi distribution (bottom plot) for a quadruplet of neutron events (top plot); ³⁵ the left column of the Rossi distribution contains only one pair of neutron events with interval between 0 and 100 μs ; the second (from the left) column of the Rossi distribution contains two pairs of neutron events with interval between 100 and 200 μs ; the third column of the Rossi distribution contains two pairs of neutron events with interval between 200 and 300 μs ; the fourth column of the Rossi distribution contains only one pair of neutron events with interval between 300 and 400 μs	36
Figure 10. Illustration of a typical Rossi distribution	37
Figure 11. Illustration of the Rossi distribution fitting.....	37
Figure 12. Illustration of 10 neutron events contributing to the Feynman Distribution	38

Figure 13.	Illustration of the $c(\Delta tg)$ array for the 10 neutron events displayed in Fig. 12 when the gate is 0.1 μs ; the left column contains neutron events between 0 and 0.1 μs ; the second column (from left) contains neutron events between 0.1 and 0.2 μs ; etc.....	38
Figure 14.	Illustration of the $c(\Delta tg)$ array for the 10 neutron events displayed in Fig. 12 when the gate is 0.2 μs ; the left column contains neutron events between 0 and 0.2 μs ; the second column (from left) contains neutron events between 0.2 and 0.4 μs ; etc.....	39
Figure 15.	Illustration of the $c(\Delta tg)$ array for the 10 neutron events displayed in Fig. 12 when the gate is 0.4 μs ; the left column contains neutron events between 0 and 0.4 μs ; the central column contains neutron events between 0.4 and 0.8 μs ; etc.	39
Figure 16.	Illustration of a typical Feynman distribution and its fitting.....	40
Figure 17.	Illustration of a typical neutron counts distribution for an experiment with 20 ms pulse period. The figure plots the <code>counts</code> array.....	40
Figure 18.	Illustration of the neutron counts distribution fitting	41
Figure 19.	Illustration of a lost <code>signal(i)</code> neutron capture event	41
Figure 20.	Neutron counts distribution for the first six pulses since the start of the accelerator	42
Figure 21.	Neutron counts distribution for six pulses after 20,000 pulses since the start of the accelerator	42
Figure 22.	The top plot shows the simulated detector response, as a function of time, from a single pulse of the particle accelerator; the bottom plot shows the superimposition (shift) of six neutron pulses from -100 to 20 ms	43

NEUTRON DETECTOR SIGNAL PROCESSING TO CALCULATE THE EFFECTIVE NEUTRON MULTIPLICATION FACTOR OF SUBCRITICAL ASSEMBLIES

Abstract

This report describes different methodologies to calculate the effective neutron multiplication factor of subcritical assemblies by processing the neutron detector signals using MATLAB scripts. The subcritical assembly can be driven either by a spontaneous fission neutron source (e.g. californium) or by a neutron source generated from the interactions of accelerated particles with target materials. In the latter case, when the particle accelerator operates in a pulsed mode, the signals are typically stored into two files. One file contains the time when neutron reactions occur and the other contains the times when the neutron pulses start. In both files, the time is given by an integer representing the number of time bins since the start of the counting. These signal files are used to construct the neutron count distribution from a single neutron pulse. The built-in functions of MATLAB are used to calculate the effective neutron multiplication factor through the application of the prompt decay fitting or the area method to the neutron count distribution.

If the subcritical assembly is driven by a spontaneous fission neutron source, then the effective multiplication factor can be evaluated either using the prompt neutron decay constant obtained from Rossi or Feynman distributions or the Modified Source Multiplication (MSM) method.

NEUTRON DETECTOR SIGNAL PROCESSING TO CALCULATE THE EFFECTIVE NEUTRON MULTIPLICATION FACTOR OF SUBCRITICAL ASSEMBLIES

1 Introduction

Signal processing of neutron detector data, using MATLAB¹ scripts, can determine the effective neutron multiplication factor of a subcritical assembly. The assembly can be driven either by a spontaneous fission neutron source (e.g. californium) or a neutron source generated from the interactions of accelerated charged particles with target materials. In the latter case, the accelerator can be operated in a continuous (steady state) or a pulsed mode. Typical target reactions are deuteron-deuteron (D-D) from incident deuterons, deuteron-tritium (D-T) from incident deuterons, spallation from incident protons, and photo-fission from incident electrons via Bremsstrahlung effect. The external neutron source is usually located in the center of the fuel zone to maximize the neutron multiplication.²⁻⁴

When the subcritical assembly is driven by a spontaneous fission neutron source, the Rossi or Feynman distributions⁵⁻⁸ can be used to calculate the effective neutron multiplication factor through the evaluation of the prompt neutron decay constant, as described in Sections 6.B and 6.C. Alternatively, the effective multiplication factor can be also estimated using the Modified Source Multiplication (MSM) method, as discussed in Section 6.D.

If the subcritical assembly is driven by a pulsed external neutron source, the neutron detector signals can be used to measure the effective neutron multiplication factor using the prompt neutron decay fitting⁵ or the area method,⁹ as described in Section 7. The area method is more accurate than noise (Rossi or Feynman) or prompt decay fitting methods because it relies on the integration over a fixed range (the pulse period), rather than the fitting, over an arbitrary range, of the measured data. In addition, the area method does not use the neutron generation time Λ , which depends on the subcriticality level and detector position. The area method cannot be used when the subcritical assembly is driven by a spontaneous fission neutron source or by a particle accelerator operating in a continuous mode.

The neutron population can be divided into prompt and delayed.¹⁰ Prompt neutrons are immediately generated after a fission event up to a few milliseconds. Delayed neutrons are emitted from the decay of the fission fragments, which have half-life in the range of 0.23 to 55.7 seconds for ²³⁵U. Consequently, in the area method experimental setup, the particle accelerator should produce neutron pulses for at least 10 to 15 minutes prior to the starting of the neutron counting. This time interval before starting the neutron counting ensures that the delayed neutron contribution to the detector counting is approximately constant during the pulse period. A particle accelerator operating in pulsed mode ejects particles for a short time interval (pulse

duration) and repeats this ejection at a fixed time interval (pulse period T). Typical values of the pulse duration and period range from 5 to 10 microseconds (μs) and from 20 to 50 milliseconds (ms), respectively. The pulse period must be long enough to allow prompt neutrons to decay and short enough to have a constant contribution from the delayed neutrons during the pulse period. The pulse period can be retrieved either from the accelerator signal file recording the start of each neutron pulse or from the accelerator control system. The pulse duration cannot be retrieved from the accelerator signal file, but it can be obtained from the accelerator control system.

Sections 2 to 5 review some basic principles on neutron detection. Sections 6 and 7 discuss the methodologies to obtain the effective multiplication factor from the experimental detector signal acquired in steady state and pulsed neutron source operation modes, respectively. Section 8 explains how to obtain the effective neutron multiplication factor from Monte Carlo computations simulating the detector signals acquired in steady state and pulsed neutron source operation modes. Section 9 shows how to correct the effective multiplication factor obtained from experiments discussed in Sections 6 and 7 using Monte Carlo computer simulations discussed in Section 8.¹¹⁻¹³

This report focuses on using gaseous ionization detectors since the MATLAB scripts given in Appendixes A to D have been validated using the YALINA Thermal facility experimental data from ^3He neutron detectors.¹³

In this report, all the variables used in the numerical algorithms of the MATLAB scripts are written using the courier mono-space font and capital letters are used for arbitrary constants.

2 Neutron Detectors

Different detectors types are available for neutron and gamma radiation measurements. Gaseous ionization detectors consist of a gas volume between two electrodes, a wire acting as an anode and a metallic hollow cylinder acting as a cathode, as shown in Fig. 1. Typically, 30 to 40 eV radiation energy deposition create an ion pair, which consists of an electron and a positive charged ion.

Scintillation detectors consist of a luminescent material producing UV and/or visible light during ionization events. More precisely, the ionization event elevates an electron from its valence band into a conduction band and the electron de-excitation occurs by photon emission. The electron de-excitation occurs from less than one up to 250 nanoseconds, this usually reduces dead-time effects relative to gaseous ionization detectors. For scintillation detectors, the radiation energy deposition creating a single photoelectron is ~ 100 eV, this value is much bigger than the value required for gaseous ionization detectors of ~ 30 eV. In scintillation detectors, the discrimination between events from neutron, gamma, and alpha particles is performed by pulse-shape analyses. The voltage pulse from neutron events decays slower than the one from gamma events and the voltage pulse from alpha events decays slower than the one from neutron events. This is illustrated in Fig. 4 of Ref. 14, Fig. 2 of Ref. 15, and Fig. 13.16 of Ref.

16.¹⁴⁻¹⁶ Consequently, analyses on the tail (shape) of the voltage pulse allow the discrimination of particle types. Conversely, for gaseous ionization detectors, the discrimination between neutron and gamma contributions is performed by pulse-height (instead of pulse-shape) analyses (Section 4).

Semiconductor detectors consist of doped crystals of silicon, or other materials, which act as diodes. Semiconductor detectors have higher energy resolution relative to gaseous ionization and scintillation detectors because only a few eV (instead of ~30 and ~100 eV as in the case of gaseous ionization and scintillation detectors, respectively) of radiation energy deposition produce a pair of charged carrier (electrons and holes). In addition, semiconductor detectors are solid and therefore have more than thousand times higher density relative to gaseous ionization detectors. In semiconductor detectors, ions and electrons have similar mobility; in gaseous ionization detectors, electrons move much faster than ions. For a semiconductor detector, a typical charge collecting time, which is proportional to the detector dead-time, is a fraction of μs .

In general, neutron detectors have two different operation modes: pulse and current. In pulse mode, the neutron detector generates a signal (voltage pulse) for every single neutron interaction event.¹⁷ When the neutron flux is very high, it is no longer possible to distinguish each single interaction event and the pulse mode does not provide accurate results. In pulse mode, the gamma radiation contribution can be discriminated based on the signal (voltage) strength (pulse-height discrimination) or on the signal (voltage) shape (pulse-shape discrimination). The pulse-height discrimination is performed by the multi-channel analyzer (the MCA described in Section 4) and takes into account that the number of ionizations from gamma ray interactions is much smaller than the one from neutron interactions. The pulse-shape discrimination is performed by the preamplifier (described in Section 4) and takes into account that the voltage pulse generated by neutron events decays slower than the one generated by gamma events.

In current mode, the detector signal is given by the average current induced by all ionization events.¹⁷ This mode can be used with a high neutron flux since the scoring is not affected by the dead-time effect. However, gamma radiation discrimination is not possible because only the average signal (voltage) of many pulses is measured. Consequently, this detector operation mode does not permit discriminating the gamma ray contributions according to the signal (voltage) value or shape of a single ionization event.

In addition to pulse and current modes, fission chambers can be also operated in Campbell mode. In this operational mode, the detector measures the standard deviation of the current produced by many ionization events. The major benefit of this operational mode is that the gamma radiation contribution to the detector signal can be discriminated even with a high gamma flux, which is not possible in pulse and current modes.¹⁸⁻²⁴

Typical fission chambers are made of a thin tungsten wire (~0.1 mm diameter) located at the center of an aluminum or steel hollow cylinder. An inert gases mixture, e.g. 95% argon and 5% nitrogen at 1 to 5 bars pressure, fills the hollow cylinder. Steel

allows higher operation temperatures relative to aluminum. The wire acts as anode and collects electrons; the metallic cylinder acts as cathode and collects ions (Fig. 1). Alternative fission chamber designs use a hollow cylinder as anode (Fig. 2). A thin layer of highly enriched fissile material, e.g. 95% ^{235}U , is deposited on the inner surface of the cylinder. The deposit must be thin enough to allow fission products to escape the fissile material layer. For uranium material, fission products travel up to 7 μm , which corresponds to 13 mg/cm^2 coating mass per unit area. Consequently, the typical uranium coating mass per unit area is in the range of 0.02 to 2 mg/cm^2 .¹⁶ Fission products are emitted in opposite directions, therefore one fission product enters the gas and the other is absorbed in the cylinder. The fission product entering the gas generates millions of electrons and ions since the kinetic energy of fission products is about 160 MeV; dissimilarly, the energy of prompt fission gammas is only 8 MeV. A fission product ionizes the inert gas creating a charge equivalent to that of 1.4×10^6 electrons. Electrons and ions in the gas share this total charge. Geslot et al.²⁰ have found that the total charge divides into 91 and 132 fC for electrons and ions, respectively. This charge occurs over a time interval of 150 ns and 40 μs for electrons and ions, respectively, since ions move much slower than electrons (whereas fission products have high energy and move faster than electrons). Consequently, the average current generated by the electrons (1.4 μA) is greater than that generated by ions (7.2 nA).²⁰

In a fission chamber, fission reactions in the fissile deposit could be also induced by gamma rays, however the cross section of these reactions is very small and peaks in the range of 10 to 20 MeV. Consequently, photo-fission reactions in the fissile deposit can be neglected.

Different particles and radiation contribute to gas ionization reactions that generate the signal detected by a fission chamber:

- Fission products from the fissile deposit;
- Gamma rays from the external neutron source or the assembly fuel;
- Gamma rays from fissions in the fissile deposit;
- Gamma rays from neutron capture reactions in the fission chamber materials;
- Alpha particles from the decay of fission products in the fissile deposit;
- Alpha particles from the uranium decay in the fissile deposit.

3 Gaseous Ionization Detectors Classification

Gaseous ionization detectors can be classified into three different types according to the voltage applied to the electrodes as shown in Fig. 3. If the applied voltage is low, the charge in the gas is not multiplied; ions drift to the cathode and free electrons to the anode; the detector acts as an ionization chamber (white area in Fig. 3). If the applied voltage is higher, electrons acceleration gives rise to avalanche multiplication; the detector acts as a proportional counter (cyan area in Fig. 3). If the applied voltage is very high, multiple avalanches and ultraviolet photons are created; the detector acts as a Geiger-Müller counter (green area in Fig. 3).

If the applied voltage is small (light-purple area in Fig. 3), electrons generated by particle interactions do not reach the anode since they recombine with other ionized gas molecules. In the ionization chamber region, electrons reach the anode without recombination independently of the voltage fluctuations. In the proportional counters region, electrons kinetic energy is enough to cause additional ionizations (avalanche multiplication). In the Geiger-Müller counters region, electrons excite gas molecules, which de-excite by the emission of ultraviolet photons.

Typically, the ionization chamber operation does not suffer from dead-time effects and it can be used with strong neutron sources. It can tolerate long irradiation times without degrading the gas performance. However, a sophisticated electronic circuit is required because the output voltage is low. The output voltage value depends on the gas composition. Proportional counters can measure the radiation energy, discriminate gamma and neutrons events, and suffer from dead-time effects. In addition, their efficiency can be affected by the electrons deposition to the anode wire. Geiger-Müller counters have high output voltage, which simplifies the electronic circuit. They cannot discriminate gamma and neutrons events since they cannot measure the radiation energy, cannot endure long irradiation times because of the gas degradation, and suffer dead-time effects. Ionization chambers detectors usually operate in current mode, whereas proportional counters, Geiger-Müller counters, scintillation, and semiconductor detectors, mostly operate in pulse mode.

The operating voltage of a proportional counter depends on the pressure and the type of the used gas and on the active volume diameter. For a ^3He proportional counter with 0.5 cm diameter, the operating voltage is the range of 750 to 1650 V for 2 and 10 atm gas pressure, respectively. If the ^3He proportional counter has 1.5 cm diameter, the operating voltage can range between 1000 and 1850 V for 2 and 10 atm gas pressure, respectively.

Detectors can also be classified by the type of information they produce. If the detector counts the number of interactions, it is referred to as a counter. If the detector measures the energy distribution of the interactions, it is referred to as a spectrometer. If the detector measures the total energy of the interactions, it is referred to as a dosimeter.

4 Gaseous Ionization Detectors Electronic Components

The electronic components of a gaseous ionization detector are: the preamplifier, the amplifier, and the multi-channel analyzer (MCA). The preamplifier connects the detector to the rest of the circuit electronics and converts the collected ionization charge into a voltage signal varying with time. The preamplifier output voltage is given as input to the amplifier, which amplifies the voltage pulses and reshapes them into Gaussian-shape pulses (Fig. 4). The analog (continuous with time) output voltage of the amplifier is given as input to the (MCA). This latter electronic component converts the output voltage of the amplifier from analog (continuous signal as a function of time) to digital (discrete signal as a function of time) format. A MCA consists of the union of single

channel analyzers. A single channel analyzer (SCA) has only lower and upper level voltage discriminators. The pulse is counted by the SCA only if the voltage is within the voltage channel set by the lower and upper level voltage discriminators.²⁵

If the MCA has 15 channels and the maximum and minimum voltages are 15 and 0 V, respectively, then the first channel of the MCA counts ionization events producing voltage pulses ranging from 0 up to 1 V; the second channel of the MCA counts ionization events producing voltage pulses ranging from 1 up to 2 V (Fig. 4). Figure 4 gives an illustrative example of the pulse-height discrimination technique; fake neutron counts from noise effects (Section 5) can be discriminated by ignoring the voltage peaks in the first channel; gamma rays can be discriminated by ignoring the voltage peaks in the second channel.

For ³He proportional counters, Langford et al.²⁶ proposed a method to discriminate fake (Section 5), gamma, and neutron contributions using pulse-shape (instead of pulse height) analyses. This method takes into account the rise-time of the voltage pulse generated by the ionization event. If the ionization event is caused by a gamma ray, the rise-time of the voltage pulse is much faster than the one caused by a neutron. Noise (generating fake counting, Section 5) pulses have a rise-time shorter than gamma ray pulses (Fig 3 of Ref 28). This pulse-shape discrimination proposed by Lagford et al.²⁶ is performed on the preamplifier signal, whereas the pulse-height discrimination is performed by the MCA. In addition, Langford et al.²⁶ pulse-shape discrimination is performed on the begin of the voltage pulse, whereas the pulse-shape discrimination applied in scintillation detectors is performed on the tail of the voltage pulse (Section 2).

5 Dead-Time, Noise, Pile-Up, and Wall Effects

5.A. Dead-Time Effect

When the neutron detector operates in pulse mode, the electronic signal is subjected to the dead-time effect.^{27,28} This effect occurs when the time interval between two successive neutron events is shorter than a time interval δ referred to as dead-time. The dead-time is the time elapsed since an event during which the detector cannot count another event. Within the dead-time, the detector cannot score more than one count.

Accurate modeling of the dead-time effect is complicated²⁷ and it is beyond the scope of this report. However, the non-paralyzable and paralyzable models are generally used to correct for the dead-time effect when the average intensity of the external neutron source is constant over time (e.g. a californium neutron source). In both models, if two consecutive neutron events occurred at times $signal(j)$ and $signal(j+1)$, with $signal(j+1)-signal(j) < \delta$, the neutron event occurring at $signal(j+1)$ is not counted. In the non-paralyzable model, the dead-time associated to a neutron count is not extended to next neutron count. For instance, a third neutron event, occurring at time $signal(j+2)$ greater than $signal(j) + \delta$, is scored as

shown in Fig. 5. Equation 1 relates the measured count rate m to the real count rate c for the non-paralyzable model.²⁸

$$c \cong \frac{m}{1 - m\delta} \quad (1)$$

In the paralyzable model, the dead-time associated to a neutron count is extended to next neutron count. If two neutron events are scored at times $\text{signal}(j)$ and $\text{signal}(j+1)$ with $\text{signal}(j+1) - \text{signal}(j) < \delta$, then the dead-time extends from $\text{signal}(j)$ up to $\text{signal}(j+1) + \delta$. A third neutron event occurring at time $\text{signal}(j+2)$, smaller than $\text{signal}(j+1) + \delta$, is not scored, as illustrated in Fig. 5. Section 6 discusses more in detail the signal array. Equation 2 relates the measured count rate m to the real count rate c for the paralyzable model.²⁸

$$m \cong c \cdot \exp(-c\delta) \quad (2)$$

Equations 1 and 2 can be applied only if the average intensity of the external neutron source, driving the subcritical assembly, is constant over time (e.g. a californium neutron source).

For a subcritical assembly driven by a particle accelerator operating in pulsed mode, the dead-time correction depends on the pulse duration, the pulse shape, and on the dead-time value δ . Equation 3 can be used when the accelerator neutron pulse has rectangular shape and lasts longer than the dead-time and when the measured neutron count m is small, since the denominator in the right side must be positive.²⁹

$$c \cong \frac{m}{1 - \frac{m\delta \cdot T}{w} \left(1 - \frac{\delta}{2w}\right)} \quad (3)$$

In Eq. 3, w is the effective pulse duration, which also accounts for the time of the neutron thermalization process, and T is the pulse period. Equation 4 can be used if the dead-time is longer than the pulse duration.³⁰

$$c \cong m - \frac{1}{T} \log(1 - mT) \quad (4)$$

In Eqs. 1 to 4 the unit of c and m is counts per seconds. Figure 6 illustrates the dead-time effect using non-paralyzable and paralyzable models when the dead-time is 3.5 or 50 μs and the average intensity of the external neutron source is constant over time. When the dead-time is 3.5 μs , the real count rate is very close to the measured count rate up to 10,000 counts per second for the paralyzable and the non-paralyzable models. If the dead-time is extremely high, e.g. 50 μs , the real count rate significantly differs from the measured count rate. The curve referring to 50 μs dead-time has been

only plotted for illustration purpose. Helium detectors have a typical dead-time of few μs and fission chambers have a dead-time lower than 0.5 μs . The helium and fission chambers detectors used in the YALINA Thermal experiments have 3.4 and 0.5 μs dead-time, respectively.³¹

Experiments can be used to measure the dead-time. For example, the dead-time can be measured by performing four experiments using two californium neutron sources with strengths A and B .²⁸ In the first experiment, the detector scores the counts from the background neutron source m_{sf} , coming from the spontaneous fission events in the fuel material. In the second and third experiments, the detector scores the counts from the A and B neutron sources (m_A and m_B , respectively). In the fourth experiment, the detector scores the counts from both the A and B neutron sources (m_{AB}). Equations 5 to 8 apply to the four experiments, respectively.

$$c_{sf} \cong \frac{m_{sf}}{1 - m_{sf}\delta} \cong m_{sf} \quad (5)$$

$$c_A + c_{sf} \cong \frac{m_A}{1 - m_A\delta} \cong m_A(1 + m_A\delta) \quad (6)$$

$$c_B + c_{sf} \cong \frac{m_B}{1 - m_B\delta} \cong m_B(1 + m_B\delta) \quad (7)$$

$$c_A + c_B + c_{sf} \cong \frac{m_{AB}}{1 - m_{AB}\delta} \cong m_{AB}(1 + m_{AB}\delta) \quad (8)$$

In Eqs. 5 to 8 the subscript denotes the strength of the neutron source. In Eq. 5, it is assumed that the counting rate from the spontaneous fission neutron source is small enough to neglect dead-time effects. In Eqs. 6 to 8, it is assumed that the dead-time value is small enough to simplify $1/(1-m\delta)$ as $1+m\delta$. The dead-time value can be obtained by Eq. 9, which solves the equations system formed by Eqs. 5 to 8.

$$\delta \cong \frac{m_A + m_B - m_{AB} - m_{sf}}{m_{AB}^2 - m_A^2 - m_B^2} \quad (9)$$

Another technique to evaluate the dead-time is to measure the count rate from a single steady state neutron source (without any subcritical assembly) as a function of the distance R between the detector and neutron source positions. This methodology uses Eq. 10, which comes from the assumptions that: 1) the true counting rate c is inversely proportional to the square of the distance R and directly proportional to the neutron source strength S ; 2) the neutron source strength S is known; 3) the dead-time effects follow the non-paralyzable model.³²

$$\frac{S}{R^2} \cong c \cong \frac{m}{1-m\delta} \quad (10)$$

The dead-time can also be determined by an oscilloscope analyzing the amplifier output voltage as a function of time (Fig. 4). The dead-time is related to the highest full width at half maximum of all voltage peaks. Each voltage peak carries its own dead-time value which is proportional to its full width at half maximum; ideally, the value of δ takes into account all these dead-time values.³² Unfortunately, the oscilloscope technique to measure the dead-time is difficult to apply because of pile-up effects (Section 5.C).³²

For gamma ray detectors, an alternative method to estimate the dead-time is to measure the count rate c_γ from a short lived radioisotope, acting as gamma ray source (without any subcritical assembly), as a function of time. In this case, the real gamma ray count rate $c_\gamma(t)$ is given by Eq. 11.³³

$$c_\gamma(t) = c_\gamma(0)e^{-\lambda t} + c_{\gamma b} \cong c_\gamma(0)e^{-\lambda t} \quad (11)$$

In Eq. 11, λ is the decay of the gamma ray source, t is time, and $c_{\gamma b}$ is the gamma rays background contribution (from cosmic rays) to the gamma ray count rate. The latter term is small and can be ignored. The dead-time of the gamma detector can be obtained by applying Eq. 1 to the left side of Eq. 11, which gives Eq. 12.³³

$$m_\gamma(t)e^{\lambda t} = c_\gamma(0) - c_\gamma(0) \cdot \tau \cdot m_\gamma(t) \quad (12)$$

In Eq. 12, $m_\gamma(t)$ is the measured gamma ray count rate. In Eq. 12 there are two unknowns: the dead-time τ and the gamma ray count at the beginning of time $c_\gamma(0)$. These two unknowns can be retrieved by fitting the value of the left side of Eq. 12 (plotted on the y-axis) as a function of $m_\gamma(t)$ (plotted on the x-axis), as illustrated in Fig. 8 of Ref. 33.

5.B. Electronic Noise Effect

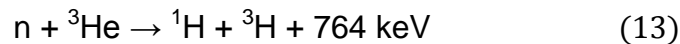
The Johnson–Nyquist electronic noise is generated by the electrons thermal agitation in the circuit electronics and is independent of the applied voltage. As discussed in Section 4, the detector components are the preamplifier, the amplifier, and the MCA. Cables connect these electronic components and longer cables produce more noise relative to shorter cables. Therefore, the electronic noise is attenuated by shorting the cables as much as possible. Consequently, the preamplifier is located as close as possible to the active zone of the detector containing the ionized gas detector. The electronic noise may generate fake neutron counts. These fake neutron counts are associated to very small peaks of voltage and should be discriminated by ignoring the counts from the first channel of the MCA (Fig. 4). Additional fake neutron counts may be generated by temperature variations causing small changes in the resistors and capacitors of the detector components.

5.C. Pile-up Effect

The pile-up effect occurs when two or more pulses overlap one-another, as shown in Fig. 7, which shows the preamplifier analog voltage as a function of time. In Fig. 7 the pulses do not have a Gaussian shape, as those in Fig. 4, because the Gaussian shape is set by the amplifier. The sum of all pulses has a much higher voltage peak relative to the voltage peak of the single pulses. This behavior changes the scoring channel of the MCA. If the gamma flux is very high, the pile-up of multiple gamma rays events can be erroneously counted as a neutron event.³⁴ This effect generates a fake neutron count. A very high gamma ray counting occurs when the external neutron source, driving the subcritical assembly, is generated by accelerated electrons via Bremsstrahlung effect.

5.D. Wall Effect

In ^3He gaseous ionization detectors, neutron interactions with gas molecules follow Eq. 13.



The 764 keV released energy is the sum of the kinetic energies of the proton (574 keV) and of the triton (190 keV). The wall effect happens when either the proton or the triton hit the cathode (the metallic hollow cylinder confining the ionization gas, Fig. 1) without (totally or partially) depositing their energy in the gas. The wall effect reduces the peak voltage generated by the amplifier and the scoring channel of the MCA. Consequently, the wall effect produces a step in-between the gamma and neutron peaks in the pulse-height spectrum, as Fig. 8 illustrates. Figure 8 plots the pulse-height spectrum, which is the number of counts as a function of the MCA channel, for a measurement performed with a low gamma flux. If the gamma flux is high, the gamma peak shown in Fig. 8 can be as high as the neutron peak and have a very long tail (Fig. 13.8 of Ref. 16).

6 Steady State Neutron Source Operation

6.A. Average Neutron Counts per Second

If the subcritical system is driven by a particle accelerator operating in a continuous mode or by a spontaneous fission neutron source, the MCA produces only one binary file. This file contains a list of positive integers and each integer represents the number of time bins (ticks) corresponding to neutron event. The zero value marks the moment when the neutron events counting starts. As discussed in Section 1, the counting starts after 10 to 15 minutes of operation to allow the delayed neutrons to reach an equilibrium condition. The value of the time bin (`TICK`) is set by the hardware of the MCA; a typical `TICK` value is 12.5 nanoseconds for YALINA thermal facility¹³. For instance, if the `signal` array contains an element with 1207 value, then the neutron event happened

after 15 microseconds, 1207 multiplied by 12.5E-9. If the `signal` array has 20 million elements, then about 20 million neutron events occurred in the detector and the file size of the `signal` array is about 100 MB.

Appendix A lists the MATLAB script for processing the detector signal file for a subcritical assembly driven by a steady state external neutron source. In this script, the `signal` array stores the detector signal data (timestamps) from the MCA. The MATLAB script executes the following steps:

- 1) It reads the neutron events data from the detector signal file and stores them into the `signal` array;
- 2) It defines the measured average counts per second m as the total number of events divided the time of the experiment;
- 3) It calculates the real counts per second c by applying the non-paralyzable dead-time model to the measured count rate m .

In step 1, the `fopen` function opens the file created by the MCA, the `fread` function transfers the experimental data from the signal file into the `signal` array in the computer RAM memory, and the `fclose` instruction closes the signal file containing experimental data.

In step 2, the total time of the experiment is obtained as the last value of the `signal` array multiplied by the time bin (`TICK`) value (12.5E-9 seconds for YALINA Thermal Facility). The total number of counts m during the experiment is given by the number of elements of the `signal` array. The MATLAB `length` function calculates the number of elements of an array.

In step 3, the real count rate c is derived by applying Equation 1.

When the experimental facility has an effective multiplication factor around 0.965 and it is driven by a californium neutron source, a typical real count rate is about 2000 counts per seconds. This value is very close to the measured count rate (e.g. 1986 counts per seconds) assuming $\delta = 3.5 \mu\text{s}$.

6.B. Rossi Distribution

The Rossi distribution is given by the number of captured neutron pairs as a function of the time interval between the two capture events (Δt), as illustrated in Fig. 9.³⁵ This distribution contains a random part, which is given by neutron captures caused by neutrons of different fission chains, and a correlated part, which is given by neutron captures caused by neutrons from the same fission chain. The correlated part exponentially vanishes with the Δt time interval following the decay of prompt neutrons, as described in Eq. 14.

$$Rossi(\Delta t) = A + Be^{-\alpha\Delta t} \quad (14)$$

In Eq. 14, Δt is the time interval between two neutron capture events, A is a constant set by neutron captures from different fission chains, B is a constant set by neutron captures from the same fission chain, and α is the prompt neutron decay constant. The Rossi distribution peaks when the time interval between two capture events is small since this condition is satisfied for neutrons generated from the same fission chain. Neutrons from the same fission chain contribute only to the right term on the right side of Eq. 11. Figure 10 illustrates a typical Rossi distribution and distinguishes the events from the same fission chain from those from different fission chains.

The effective neutron multiplication factor of the subcritical assembly can be calculated using Eq. 15.¹³

$$k_{eff} = \frac{1}{1 - \beta_{eff} - \alpha\Lambda} \quad (15)$$

In Eq. 15, α is the prompt neutron decay constant, β_{eff} is the effective delayed neutron fraction, and Λ is the neutron generation time. The α value is obtained by fitting the Rossi distribution obtained from experimental data, as shown in Fig. 11. The R in the legend of Fig. 11 denotes the fitting residual error calculated by the EzyFit MATLAB toolbox. In Eq. 15, the β_{eff} and Λ kinetic parameters are usually calculated by neutron transport programs executed in criticality mode.² Ideally, the α fitting range is chosen in a time interval where the Rossi distribution is linear on a logarithm scale. If the assembly is deeply subcritical, it is not possible to apply Eq. 14 and the decay of prompt neutrons occurs through multiple exponential functions with different decay constants.³¹ In this case, the fitting domain should be shifted to the right as much as possible so that the contribution from higher alphas is negligible. Equation 15 is based on point kinetics formulation.³⁶

Appendix B gives the MATLAB script, which calculates the Rossi distribution from the `signal` array. In this script, `x` is the domain of the Rossi distribution (Δt) and it represents the time interval between two neutron captures, `TBIN` is the time bin of the `x` array, `InvTBIN` is the inverse of the time bin, `maxx` and `maxs` are the lengths of the `x` domain and `signal` arrays, respectively, `rossi` is the Rossi distribution, `interval` is the time between two neutron captures, `h` is an integer holder, and `TICK` is the time bin of the MCA. The `InvTBIN` constant is defined to speedup calculations, since multiplications are computed faster than divisions. The `ceil` function rounds towards plus infinity a real number (e.g. 0.1 is rounded to 1). The `zeros` function creates a matrix with all elements set to zero. The calculation of the Rossi distribution is based on two nested `for` loops both iterating on the neutron captures in the `signal` array. The outer `for` loop iterates on all neutron captures in the `signal` array minus one and uses the `i` iteration index. The inner `for` loop iterates from `i+1`, marking the first neutron capture occurring after the `ith` neutron capture, to the end of the `signal` array minus

one and uses the j iteration index. At each iteration, the time interval between two neutron captures is calculated according to Eq. 16.

$$\text{interval} = \text{signal}(j) - \text{signal}(i) \quad (16)$$

If the interval value is below the maximum value (e.g. 50 ms) of the x array (the domain of the Rossi distribution), the Rossi distribution is updated; otherwise, the inner `for` loop is interrupted by a `break` instruction. The Rossi distribution can be drawn by the `plot` function and the α value can be obtained by fitting the curve with an exponential function, as previously discussed, using the EzyFit MATLAB toolbox.

6.C. Feynman Distribution

In the Poisson probability distribution, the variance is equal to the mean and the next event probability is independent of the elapsed time since the previous event. Radioactive decay and particle scattering processes follow the Poisson probability distribution. Neutron captures events do not follow the Poisson probability distribution because they are correlated events via fission chains. Consequently, the Feynman distribution Y is defined as the variance to mean ratio minus one, of the neutron counts array $c(\Delta t_g)$ as a function of a fixed time interval Δt_g (gate), as shown in Eq. 17.

$$Feynman(\Delta t_g) = Y(\Delta t_g) \equiv \frac{\text{var}(c(\Delta t_g))}{\text{mean}(c(\Delta t_g))} - 1 \quad (17)$$

An illustration of the neutron counts array $c(\Delta t_g)$ for different gates is given in Figs. 12 to 15. In this example, it is assumed that 10 neutron ionization events, occurring between 0.02 and 0.92 μs as shown in Fig. 12, contribute to the neutron counts array. In this illustrative example, the Feynman distribution is obtained by a deterministic procedure.

Alternatively, the neutron counts array $c(\Delta t_g)$ can be obtained by randomly sampling the `signal` array, which is plot in Fig. 12 and including in each $c(\Delta t_g)$ unit all the following neutrons captures within the Δt_g gate. In this case, the Feynman distribution is obtained by a Monte Carlo (random) procedure.

The prompt neutron decay constant α can be obtained by fitting the Feynman distribution $Y(\Delta t)$ with the function given in the right side of Eq. 18, as discussed in Ref. 5.

$$Y(\Delta t_g) \approx \frac{B}{\alpha^2} \left(1 - \frac{1 - e^{-\alpha \Delta t_g}}{\alpha \Delta t_g} \right) \quad (18)$$

In Eq. 18, B is a constant. Equation 18 is based on point kinetics and assumes that the subcritical assembly operates close to criticality, i.e. no multiple alphas.

Figure 16 illustrates a typical Feynman distribution. Fitting the Feynman distribution by Eq. 18 allows obtaining the prompt neutron decay constant α , as shown in Fig. 16. The effective multiplication factor of the subcritical assembly can be obtained by Eq. 15 as discussed in Section 6.B.

Appendix C reports the MATLAB script to calculate the Feynman distribution from the `signal` array data. In this script, the array `gate` is the domain of the Feynman distribution and each element of this array represents the time interval Δt_g used to build the neutron counts array $c(\Delta t_g)$, as shown in Figs. 13 to 15. In order to construct the neutron counts array $c(\Delta t_g)$ using the deterministic procedure, the length of the experiment is divided into time intervals of length Δt_g . The length of the experiment is given by the value `signal(end)*TICK`; the first term is the last unit of the `signal` array, whereas the latter term is the time bin of the MCA. `MAXT` is the maximum of the `gate` array (e.g. 50 ms). `TBIN` is the time bin of the `gate` array. `maxg` is the length of the `gate` array. `maxs` is the length of the `signal` array shorted by the `MAXT` value. `feynman1` and `feynman2` are the deterministic and the random Feynman distributions, respectively. The array `c` is the neutron counts array $c(\Delta t_g)$. The `rand` function generates a random number between 0 and 1; this function is used only in the random procedure calculating the Feynman distribution `feynman2`. The `max` function calculates the maximum value of an array. The `find` function returns all indexes of an array satisfying an equality or inequality condition. The `var` function calculates the variance of an array. The `mean` function calculates the mean of an array. The `warning` function displays a warning message. The `fprintf` function displays a string message. The user-defined `function_feynman` function returns the deterministic and random Feynman distributions and takes as input data: the `signal` array, the `gate` array, `maxs`, `maxg`, and a user defined string `mystring`, e.g. the date of the experiment. The `maxi` value is the length of the `c` array for the random procedure calculating the Feynman distribution `feynman2`; this number should be large and has been arbitrary set to hundred millions, since the length of the `signal` array is a few millions. The `hist(y,x)` function takes as input the `x` and `y` arrays and returns the distribution of `y` among bins with centers specified by `x`; this function is used only for the deterministic procedure calculating the Feynman distribution.

The random procedure is based on a `while` loop embedded into two nested `for` loops; this algorithm structure is therefore more expensive to compute relative to the Rossi distribution, which requires only two nested loops. The outer `for` loop iterates on all `gate` values using the `m` integer index. At each outer `for` loop iteration, the `c` array with `maxi` elements is set to zero. The inner `for` loop iterates on all integers between 1 and `maxi` using the `i` integer index. At each inner `for` loop iteration, a random neutron capture is sampled from the `signal` array; the integer `s` (abbreviation of *start*) is the index of this neutron capture. The `while` loop stores into the `c` array all neutron captures occurring from the time `signal(s)` up to the time `signal(s) + gate(m)`. The

random Feynman distribution value associated to the `gate(m)` value is calculated at the end of the outer `for` loop.

The deterministic procedure is based on a single `for` loop iterating on all elements of the `gate` array through the `m` index. At each iteration the `c` array is built using the `hist` function with inputs the `signal` array and an array with a length given by the duration of the experiment (shortened by `MAXT`) divided the `gate(m)` value. The first value of the latter array is `gate(m)/2` because the `hist(y,x)` returns the distribution of `y` among bins with centers specified by `x`; the `x` values do not represent the bins boundaries.

6.D. The Modified Source Multiplication (MSM) Method

For a subcritical assembly driven by a steady state external neutron source, the neutron detector counting c is equal to the product of the detector efficiency ε , the Spriggs factor g^* , the multiplication factor M , and the source intensity S , as defined in Eqs. 19 to 24.³⁷

$$c = \varepsilon g^* M S \quad (19)$$

$$\begin{aligned} \varepsilon &\equiv \frac{\int_{V_D} d\vec{r} \int_0^{+\infty} dE \int_{4\pi} d\vec{\Omega} \Sigma_D(\vec{r}, E) \varphi_s(\vec{r}, \vec{\Omega}, E)}{\int_V d\vec{r} \int_0^{+\infty} dE \int_{4\pi} d\vec{\Omega} \left[s(\vec{r}, \vec{\Omega}, E) + \frac{\chi(\vec{r}, E)}{4\pi} \int_0^{+\infty} dE' \int_{4\pi} d\vec{\Omega}' \nu \Sigma_f(\vec{r}, E') \varphi_s(\vec{r}, \vec{\Omega}', E') \right]} \\ &= \frac{\int_V d\vec{r} \int_0^{+\infty} dE \int_{4\pi} d\vec{\Omega} \Sigma_D(\vec{r}, E) \varphi_s(\vec{r}, \vec{\Omega}, E)}{S + F} \quad (20) \end{aligned}$$

$$g^* \equiv \frac{1 - k_{eff}}{1 - k_{src}} \quad (21)$$

$$k_{src} \equiv \frac{F}{F + S} \quad (22)$$

$$M = \frac{1}{1 - k_{eff}} \quad (23)$$

$$S = \int_V d\vec{r} \int_0^{+\infty} dE \int_{4\pi} d\vec{\Omega} s(\vec{r}, E, \vec{\Omega}) \quad (24)$$

In Eqs. 19 to 24, \vec{r} , $\vec{\Omega}$, E , Σ_D , Σ_f , χ , ν , V , V_D , φ_s , s , k_{src} , and k_{eff} are: the neutron position, the neutron flight direction, the neutron energy, the macroscopic cross section of the gas in the detector, the fission macroscopic cross section of the fuel, the fission neutron spectrum, the number of fission neutrons, the assembly volume, the detector volume, the angular neutron flux solving the inhomogeneous neutron transport equation (with the external neutron source term), the space-energy-angle distribution of the external neutron source, the source multiplication factor, and the effective multiplication factor, respectively. In Eq. 2, the (n,xn) reactions have been neglected in order to simplify the equations, although those reactions can be accounted for similarly to the fission reactions.³⁸

If there are two configurations of the subcritical assembly with different reactivities, labeled by subscripts 1 and 2, then the ratio between the detector counting rates is given by Eq. 25, which introduces the f_{MSM} factor.

$$\frac{c_2}{c_1} = \frac{\varepsilon_2 g_2^* M_2}{\varepsilon_1 g_1^* M_1} \equiv f_{MSM} \frac{M_2}{M_1} \quad (25)$$

The f_{MSM} factor can be calculated (e.g. by MCNP simulations)³⁹ according to Eq. 26.^{6,40}

$$f_{MSM} = \frac{c_2^{MCNP} M_1^{MCNP}}{c_1^{MCNP} M_2^{MCNP}} \quad (26)$$

Four MCNP simulations are required to obtain the f_{MSM} value, because the two detector counts are obtained by two source mode simulations (with the external neutron source), whereas the two multiplication values are obtained by two criticality mode simulations (without the external neutron source).

Assuming that the experimental value of the multiplication factor for configuration 1 $k_{eff1,c}^{exp,reference}$ is known, the modified source method (MSM) experimental value of the multiplication factor for configuration 2 ($k_{eff2}^{exp,msm}$) can be obtained using Eq. 27.

$$k_{eff2}^{exp,msm} = 1 - \frac{f_{MSM} c_1^{exp} (1 - k_{eff1,c}^{exp,reference})}{c_2^{exp}} \quad (27)$$

In Eq. 27, c_1^{exp} and c_2^{exp} are the detector neutron counts obtained from experimental measurements for configuration 1 and 2, respectively.⁴¹ It is assumed that the two detector counts are measured in the same experimental channel with the same detector at slightly different reactivity levels of the subcritical assembly.

The known reference value $k_{eff1,c}^{exp,reference}$ is equal to 1 if the experimental facility can be operated in criticality condition. Otherwise, this value must be obtained by other methods, e.g. Rossi, Feynman, or Area methods.

7 Pulsed External Neutron Source Operation

When a particle accelerator operates in pulsed mode and drives the subcritical assembly, the MCA produces two files. The first file is similar to the one discussed in Section 6, since it stores the timestamps when a neutron is captured. The second file stores the timestamps when the accelerator starts delivering a neutron pulse; therefore, the stored timestamps increase by the pulse period T , e.g. 20 ms. In both files, timestamps are stored as integers representing the number of time bins, in MCA ticks unit. The two arrays are referred to as `signal`, for the detector data, and `signalt`, for the accelerator beginning of pulse data. Consequently, the index i of the `signal` array distinguishes different neutron captures in the detector, whereas the index j of the `signalt` array distinguishes different neutron pulses from the particle accelerator. In an ideal experiment, all elements of the `signalt` array satisfy Equation 28.

$$\text{signal}_t(j+1) - \text{signal}_t(j) = \text{PERIOD} \quad (28)$$

In a real experiment, the single pulse period can have a small deviation from the average value of the pulses periods (`PERIOD`), which invalidates Eq. 28. For instance, for some pulse, the period may be 19.95 or 20.05 ms instead of 20 ms. If the `signalt` array has 70,000 elements, then the particle accelerator produced $\sim 70,000$ pulses during the experiment. In the previous example, if the pulse period is 20 ms and the time bin `TICK` is 12.5 ns, then the acquisition time of the detector is ~ 23 minutes and the last element of the `signalt` array would contain the value $112 \cdot 10^9$. In addition, the total operation time of the particle accelerator would be 33 to 38 minutes including 10 to 15 minutes before the detector starts counting to establish the equilibrium condition for delayed neutrons.

A typical neutron counts distribution is shown in Fig. 17, which highlights in different colors the contributions from prompt and delayed neutrons. This distribution is obtained by processing the `signal` and `signalt` arrays by the MATLAB script listed in Appendix D. The prompt neutron decay constant can be obtained by fitting the neutron counts distribution as shown in Fig. 18. The effective neutron multiplication factor of the subcritical assembly can be obtained by using Eq. 15 as discussed in Section 6.B. This procedure is usually referred to as the prompt decay fitting method.⁵

The area method is an alternative accurate procedure to calculate the effective multiplication factor k_{eff} of the experimental facility through processing the `signal` and `signalt` arrays.^{42,43} This method is based on point kinetics and estimates the effective multiplication factor according to Eq. 29.⁹

$$k_{eff} \cong \frac{1}{1 + \frac{A_p \cdot \beta_{eff}}{A_d}} \quad (29)$$

In Equation 29, the prompt (A_p) and the delayed (A_d) areas are the areas underneath the curve of the neutron counts distribution during the pulse period as shown in Fig. 17. β_{eff} is the effective delayed neutron fraction, which is usually calculated from a criticality computer simulation. Unlike Eq. 15, the area method does not use the prompt neutron generation time. This latter parameter has a strong dependence on the subcriticality level of the assembly and the detector position. By contrast, the effective delayed neutron fraction is not highly sensitive to the subcriticality level of the assembly.^{2,3}

In the MATLAB script listed in Appendix D, the elements of the neutron count distribution plotted in Fig. 17 are stored into the `counts` array. Consequently, the effective multiplication factor can be calculated from the `counts` array data using Eq. 29. The `counts` array is obtained by processing the `signal` and `signal_t` arrays from the MCA experimental data and contains a number of elements equal to `PERIOD/TBIN`. Each element of the `counts` array represents the average number of neutron captures within a `TBIN` time interval. Typical values of the `TBIN` and `PERIOD` parameters and of the `counts` array length are 4 μ s, 20 ms, and 5,000 elements, respectively. In this case, each element of the `counts` array would be the average number of neutron captures within 4 μ s. The `TBIN` value (4 μ s) that defines the time bin in the `counts` array is arbitrarily set. On the contrary, the `TICK` value (12.5 ns) that defines the time bin in the `signal` array is set by the MCA electronic hardware.

The MATLAB script given in Appendix D executes the following steps:

- 1) It defines the script constants;
- 2) It opens the `signal` and `signal_t` files, reads all data, and stores the data (timestamps) into the `signal` and `signal_t` arrays;
- 3) It converts the unit of `signal` and `signal_t` arrays from time bins to seconds;
- 4) It calculates the `PERIOD` value as the average of all pulses periods; a single pulse period is obtained as the difference between consecutive data of the `signal_t` array (Eq. 28);
- 5) It defines the `counts` array; each element of this array stores the average number of neutron captures within a `TBIN` time interval; the time domain of the `counts` array ranges from the beginning of a neutron pulse to the pulse period;
- 6) It updates the `counts` array using the `signal` and `signal_t` arrays data;
- 7) It divides the `counts` array by the total number of pulses in the experiment; the latter value is approximately given by the size of the `signal_t` array;
- 8) It calculates the effective neutron multiplication factor `keff` using Eq. 29;
- 9) It plots the neutron counts distribution using the counts per seconds unit; this distribution is given by the `c/TBIN` array data.

In step 1, the delayed neutron fraction, the MCA time bin, the `counts` array time bin, and the inverse of the latter parameter are defined in the constants: `BEFF`, `TICK`, `TBIN`, and `ITBIN`, respectively. The latter parameter is only defined to reduce the computation time since a multiplication is computed faster than a division.

Step 2 is the same as step 1 of Appendix A. As discussed at the beginning of this Section, for a pulsed external neutron source operation, the MCA generates two binary files. These two files are processed in the same manner.

In step 3, the time unit of the `signal` and `signalt` arrays is changed from time bins to seconds. This is accomplished by multiplying all elements of the arrays by the `TICK` value.

In step 4, the average pulses period is calculated and stored into the `PERIOD` variable. The period of each pulse is calculated according to left side of Eq. 28. The `diff` MATLAB function calculates the difference between adjacent elements of an array; the `mean` MATLAB function calculates the average value of an array. In the averaging process, the first value of the `signalt` array is discharged because the period of the first pulse might be quite different from the average value of the pulses periods. This situation may originate when the start of the detector scoring and files writing are not synchronized with the beginning of a pulse from the particle accelerator.

In step 5, the `counts` array is defined and initialized to zero. Fig. 17 plots the elements of the `counts` array for an experimental setup with 20 ms period. The array `x` is the time domain of the neutron counts distribution `counts`.

In step 6, the index `i` of the `signal` array is advanced until the current value is greater than the second element of the `signalt` array. This procedure skips all neutron captures occurring before the beginning of the second pulse. For all data in the `signal` array, the difference between the time when a neutron is captured `signal(i)` and the time when a pulse begins `signalt(j)` is stored into the `TimeFromLastTriggerPulse` variable. If the `TimeFromLastTriggerPulse` value is positive and smaller than the average of the pulses periods (`PERIOD`), then the value of the `counts` array with index approximately equal to `TimeFromLastTriggerPulse` divided by `TBIN` is increased by one. The index approximation is carried out by the `ceil` MATLAB function, which rounds a real number towards plus infinity. The use of the `ceil` function avoids having zero as index of an array; unlike the C language, MATLAB scripts do not allow arrays indexes to start from zero. If the `signal(i)` value is greater than the `signalt(j+1)` value (which marks the beginning of the next pulse), then the index of the `signalt` array is increased by one and the previous procedure is reapplied. The increase of the `j` index means that all neutron captures in the `j` pulse have been processed and neutron captures occurring in the `j+1` pulse begin being processed. If the index of the `signalt` array reaches its maximum value (which occurs if the condition `j >= length(signalt)` is satisfied), all pulses have been analyzed

and the `counts` array is no longer updated; this condition is handled by the `break` MATLAB instruction which interrupts the `while` loop.

If the `signal(i)` value is smaller than the `signalt(j+1)` value and the latter value is higher than the average value of the pulses periods (`PERIOD`), then the neutron count from `signal(i)` is lost and the `lost` variable (which is initialized to zero) is increased by one. This condition is illustrated in Fig. 19, where the red and blue dashed vertical lines are associated to the begin of the pulse data and the magenta solid vertical line is associated to the lost neutron capture event.

In step 7, the `counts` array is divided by the number of pulses minus one. The number of pulses is diminished by one to take into account the discharge of the first pulse (steps 4 and 6). After this normalization the `counts` array stores the average number of neutron capture events, within a `TBIN` time interval, during a single pulse.

In step 8, the total area is calculated as the sum of the `counts` array elements. The `sum` function returns the sum of an array. The total area is given by the sum of the prompt area A_p (yellow color in Fig. 17) and the delayed area A_d (lavender color in Fig. 17). Consequently, the total area is the area beneath the green curve of Fig. 9, which plots the normalized `counts` array elements. In Fig. 17, the maximum of the `count` array is normalized to one. The average value of the last 10% of the `counts` array elements multiplied by the total number of elements gives the delayed area. If the pulse period is 20 ms, the delayed area comes from the average of the `counts` array elements in the last 2 ms. In the averaging process, the last three elements of the `counts` array are discharged since their values may have been affected by the difference in the pulses periods. In the application of Eq. 29, the prompt area is calculated as the difference between the total and the delayed areas.

The number of neutron captures per second can be obtained by dividing the `counts` array by the `TBIN` value (step 9).

8 MCNP Simulations

The exact time when a neutron is captured in the detector can also be obtained by MCNP simulations. Consequently, processing Monte Carlo simulations output data can be used to construct the Rossi (Section 6.B), the Feynman (Section 6.C), and the neutron counts (Section 7) distributions. In addition, MCNP simulations can be used to calculate the f_{MSM} factor (Eq. 26) used to apply the MSM method (Section 6.D).

MCNP stores the time when a neutron capture occurred in the PTRAC output file. However, in the original version of MCNP, the neutron capture timestamp does not take into account the time when the neutron was emitted by the external neutron source. This issue can be solved by applying the following patch to the `ptrak.F90` file (line 168) for MCNP version 6.1.1 beta version.

```

168c168
<      write(iupw,'(i10,1pe15.5,7i8)')nps,pbl%r%tme-pbl_src%r%tme,ic,kn(1), &
---
>      write(iupw,'(i10,1pe35.25,7i8)')nps,pbl%r%tme,ic,kn(1), &

```

The above patch also increases the precision of the neutron capture timestamp. The original version of MCNP uses low (up to the millisecond) precision, which is inadequate for the application of noise (Rossi and Feynman) methods.

Appendix E gives an example of the MCNP input deck to obtain the timestamps of the ^3He neutron captures in cell 1700 defining the gas region of the neutron detector for a subcritical assembly driven by a californium neutron source. The californium neutron source emits neutrons for 1 hour. In order to obtain the neutron capture timestamps, the `ptrac` card must be coupled to the `F8` tally. The `ptrac` card requires analog captures, without any variance reduction techniques, and serial computations, no MPI or OPENMP parallel computing platforms. However, the results of multiple serial computations can be easily combined together provided that each computation starts with a different random number seed using the `rand` card. MCNP output data need to be sorted in increasing order; this is accomplished by the `sort` MATLAB function.

The PTRAC output data from MCNP can be processed using MATLAB scripts similar to those listed in Appendices B and C (with no major modification) to construct the Rossi and Feynman distributions, respectively.

For the MSM method (Section 6.D), the calculation of the f_{MSM} factor (Eq. 26) is straightforward accomplished by performing two source mode simulations with the californium neutron source and two criticality mode simulations without the californium neutron source. In the source mode simulations, the count rates are obtained by the `F4` tally. For the MSM method, the MCNP simulations are simple because they are time independent. Conversely, all other methods discussed in this report require time dependent MCNP simulations to obtain the effective multiplication factor.

For a subcritical assembly driven by a particle accelerator operating in pulsed mode, constructing the neutron counts distribution from MCNP data requires additional algorithms relative to the procedure that applies to experimental results. MCNP simulates only one neutron pulse, which is not enough to set the equilibrium condition of delayed neutrons. In order to set the equilibrium condition of delayed neutrons, MCNP results from a single pulse must be shifted and superimposed.^{42,43} When the particle accelerator is switched on, the delayed neutrons contribution increases with time, as illustrated in Fig. 20. After 10 to 15 minutes from the start of the particle accelerator, delayed neutrons reach the equilibrium condition, as shown in Fig. 21. The pulses superimposition methodology uses the MCNP results of a single pulse, as shown at the top plot of Fig. 22, and superimposes the single pulse results $\sim 20,000$ times.^{42,43} For instance, the bottom plot of Fig. 22 illustrates 6 consecutive pulses obtained from shifting a single pulse data. In the bottom plot of Fig. 22, the detector signal between 0 and 20 ms is given by: 1) prompt and delayed neutrons from the blue pulse; 2) delayed

neutrons (from the tail of) black, magenta, cyan, green, and red pulses. The equilibrium condition of delayed neutrons is obtained when delayed neutrons from the previous 20,000 pulses contribute to the detector signal between 0 and 20 ms.

The pulses superimposition is computed extremely fast by the application of the procedure discussed below. It is assumed that the subcritical system is driven by a pulsed neutron source and that the detector response from delayed neutrons is constant (at equilibrium) during the pulses period. In addition, the pulse period is very small relative to the half-lives of the delayed neutrons precursors and very large relative to the half-life of prompt neutrons as discussed in Section 1. Under all these assumptions, the detector response (the area underneath the reaction rate curve) is obtained from the integration of the neutron reaction rate (r) during the pulse period (T), as shown in Eq. 30.

$$A = A_p + A_d = \int_0^T r(t)dt = \int_0^\infty r_1(t)dt \quad (30)$$

In Eq. 30, t is time, r_1 is the (e.g. ${}^3\text{He}(n,p)$) reaction rate from a single pulse of the particle accelerator when delayed neutrons are not at equilibrium as shown in Fig. 20 and top plot of Fig. 22, and r is the (e.g. ${}^3\text{He}(n,p)$) reaction rate from a single pulse of the particle accelerator when delayed neutrons are at equilibrium as shown in Fig. 21. The integral in the right side of Eq. 30 can be split into the sum of two integrals, as shown in Eq. 31.

$$\int_0^\infty r_1(t)dt = \int_0^T r_1(t)dt + \int_T^\infty r_1(t)dt \quad (31)$$

The first term on the right side of Eq. 31 represents the integration of the neutron detector response from a single pulse during the period T . Prompt and delayed neutrons dominate the first and second terms, respectively. However, the first term is not exactly equal to the prompt area A_p because some delayed neutrons are emitted during the period T . Analogously, the second term on the right side of Eq. 31 is not exactly equal to the delayed area A_d .

The pulses superimposition decomposes the second term in the right side of Eq. 31 into a series of integrals, as shown in the right side of Eq. 32. This method has been used in Refs. 42 and 43 and it requires handling a large array of $\sim 5 \times 10^7$ elements.

$$\int_T^\infty r_1(t)dt = \sum_{n=1}^{\infty} \int_{nT}^{(n+1)T} r_1(t + nT)dt \quad (32)$$

A much more efficient way to compute the pulse superimposition methodology is to apply Eq. 33. This method is much faster, since it does not require any large array, and it has been introduced in Reference 44.

$$r(t) = r_1(t)dt + \frac{1}{T} \int_T^{\infty} r_1(t)dt \quad \forall t \in [0, T] \quad (33)$$

Consequently, the MATLAB processing of the PTRAC output file data (timestamps) from MCNP simulations can be accomplished by the MATLAB script listed in Appendix F. In this MATLAB script, `N` is the number of elements of the neutron counts distribution array `counts` (Fig. 17), `PERIOD` is pulses period of the accelerator, `TBIN` is the time bin of the `counts` array, `signal` is the array with the sorted PTRAC neutron capture timestamps from MCNP simulations, `k` is the number of neutron captures within the period, and `i` and `j` are iteration indexes. The `counts` array is updated by two nested `for` loops. The outer `for` loop iterates on all neutron captures from the MCNP timestamps stored in the `signal` array, which range from 0 up to ~500 seconds. The inner `for` loop iterates on all the time bins of the `counts` array to properly update the element corresponding to the neutron capture occurring at time `signal(i)`; the `counts` array is updated only with neutron captures occurring within the pulse period.

The efficient computational methodology does no longer rely on the right side of Eq. 32 and takes instead advantage of Eq. 33. This approach considerably reduces the computing time and the statistical fluctuations of the neutron captures from delayed neutrons.⁴⁴

The MCNP input to calculate the PTRAC output data for the MATLAB script of Appendix F is shown in Appendices G, H, and I for D-D and D-T, proton, and electron neutron sources, respectively. In these appendices, the D-D and D-T neutron sources are generated by 250 keV accelerated deuterons (the target consists of deuterium or tritium nuclei); the proton neutron source is generated by 100 MeV accelerated protons; the electron neutron source is generated by 200 MeV accelerated electrons via Bremsstrahlung effect and photo-fission reactions. The following procedure applies to all the previous neutron sources:

- 1) The simulation models only one pulse of the particle accelerator; e.g. the `time` entry of the `src` card sets the particle emission duration to 5 μ s, which is the accelerator pulse width;
- 2) The simulation tallies the neutron capture reaction rate, in the detector, including the delayed neutrons contribution, e.g. the `cut` card first entry is equal to 5×10^{10} so that the delayed neutrons are transported up to 500 seconds;
- 3) The simulation is performed in analog mode without any variance reduction techniques and without multiprocessing option;
- 4) the `PTRAC` card coupled to the `F8` tally writes the timestamps when a neutron is scored (e.g. captured) in the (^3He) detector; the input has the following input card: `PTRAC event=cap file=asc type=n write=all max=2e9 coinc=lin tally=8;`
- 5) The special capture treatment applies to the `F8` tally; the input for scoring neutron captures by the ^3He atoms contains: `FT8 cap -1 -1 2003.`

9 Correction Factors for the Experimental Measurements

In a subcritical assembly, the measured reactivity depends on the detector position.¹² In order to properly take into account this behavior, Bell and Glasstone introduced the correction factor described in Eq. 34,⁴⁵ where $\tilde{\rho}^{\text{exp}}$ and ρ^{exp} represent the corrected and the uncorrected reactivities from experimental measurements, respectively.

$$\tilde{\rho}^{\text{exp}} = c \cdot \rho^{\text{exp}} \quad (34)$$

The correction factor c in Eq. 34 is calculated by Eq. 35, where ρ_{cri} is the reactivity calculated by computer codes in criticality mode (without the external neutron source) and ρ_{src} is the reactivity calculated by computer codes in source mode (with the external neutron source).

$$c \equiv \frac{\rho_{\text{cri}}}{\rho_{\text{src}}} = - \frac{1}{\rho_{\text{src}}} \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} \quad (35)$$

If the area method is used, then the correction factor can be calculated by Eq. 36.

$$c \equiv \frac{\rho_{\text{cri}}}{\rho_{\text{src}}} = - \frac{A_d}{A_p \cdot \beta_{\text{eff}}} \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} \quad (36)$$

In Equation 36, the effective delayed neutron fraction β_{eff} and the effective multiplication factor k_{eff} are calculated by MCNP or any other neutron transport computer program in criticality mode. To the contrary, the prompt (A_p) and the delayed (A_d) areas are calculated by MCNP in source mode.¹¹

Alternatively, the experimental effective multiplication factor $k_{\text{eff}}^{\text{exp}}$ can be also calculated by Eq. 37 using the prompt neutron decay constant α^{exp} from experimental data (as discussed in Sections 6.B, 6.C and 7).

$$k_{\text{eff}}^{\text{exp}} = \frac{1}{1 - \beta_{\text{eff}} - \alpha^{\text{exp}} \Lambda} \quad (37)$$

In Eq. 37, the effective delayed neutron fraction β_{eff} and the neutron generation time Λ are usually calculated with MCNP simulations, or other neutron transport program, in criticality mode. The corrected experimental effective multiplication $\tilde{k}_{\text{eff}}^{\text{exp}}$ factor can be obtained by Eq. 38.

$$\tilde{k}_{eff}^{exp} = \frac{1}{1 - \beta_{eff} - c_{\alpha} \alpha^{exp} \Lambda} \quad (38)$$

In Eq. 38, the correction factor c_{α} is calculated by Eq. 39.¹³

$$c_{\alpha} \equiv \frac{\alpha_{cri}}{\alpha_{src}} \quad (39)$$

In Eq. 39, α_{src} is the prompt neutron decay constant calculated by a neutron transport computer program in source mode with a pulsed neutron source and α_{cri} is the prompt decay constant calculated by a neutron transport computer program in criticality mode, as defined in Eq. 40.

$$\alpha_{cri} \equiv \frac{\rho_{cri} - \beta_{eff}}{\Lambda} \quad (40)$$

10 Conclusions

MATLAB scripts were developed to analyze the neutron detector data (timestamps) obtained by either an electronic digital acquisition system or by MCNP computer simulations.

For a subcritical assembly operating in a steady state condition, the electronic digital acquisition system generates one file, which can be processed with the scripts given in Appendices A, B, and C to obtain the real neutron counts (e.g. $^3\text{He}(n,p)$ neutron captures) per second, the Rossi distribution, and the Feynman distribution, respectively. The Rossi and Feynman distributions can be used to obtain the experimental prompt neutron decay constant. Within the point kinetics framework, this latter parameter can be used to obtain the effective neutron multiplication factor by a simple formula. Alternatively, computer simulations can be used to calculate the f_{MSM} factor to obtain the experimental multiplication factor by the MSM method.

For a subcritical assembly driven by a pulsed neutron source, the electronic digital acquisition system generates two files. One file contains the timestamps when a neutron capture occurs and the other contains the timestamps when a neutron pulse starts. In both files the time unit is not seconds but time bins. These two files can be processed to calculate the average neutron counts distribution during a pulse period. The script in Appendix D accounts for unequal pulse periods of the particle accelerator. The prompt decay fitting or the area methods can be used to obtain the effective neutron multiplication factor of the subcritical assembly.

Both the prompt neutron decay constant and the effective neutron multiplication factor obtained by the above procedures require a correction factor that takes into account the detector position and type. Computer simulations must calculate this correction factor.

The MATLAB scripts used to process experimental data can be also used to process data from the Monte Carlo neutron transport code MCNP. The original version of MCNP must be updated to increase accuracy of the timestamps and to take into account the time when the neutron is emitted from the external neutron source. For pulsed neutron sources, the MATLAB script used to process experimental data cannot be directly used to process MCNP data. Additional algorithms must be used to take into account that MCNP simulations model a single neutron pulse and delayed neutrons are not at equilibrium.

ACKNOWLEDGEMENTS

This work is supported by the U.S. Department of Energy, Office of Material Management and Minimization (M3), National Nuclear Security Administration.

The authors acknowledge and thank Drs. Sergey Sadovich (CERN, Switzerland) and Klaus Ott (Helmholtz-Zentrum Berlin, Germany) for the fruitful discussions.

References

- [1] [MATLAB](#), *The language of Technical Computing*, 2016.
- [2] A. TALAMO et al., *MCNPX, MONK, and ERANOS Analyses of the YALINA Booster Subcritical Assembly*, [Nuclear Engineering and Design](#) **241/5**, pp. 1606-1615, 2011.
- [3] A. TALAMO et al., *Monte Carlo and Deterministic Neutronics Analyses of YALINA Thermal Facility and Comparison with Experimental Results*, [Nuclear Technology](#) **184/2**, pp. 131-147, 2013.
- [4] A. TALAMO and Y. GOHAR, *Monte Carlo Modeling and Analyses of YALINA-Booster Subcritical Assembly Part I: Analytical Models and Main Neutronics Parameters*, Argonne National Laboratory, [ANL-NE-08/13](#). Chicago, IL, USA, 2008.
- [5] BERGLÖF, *On Measurement and Monitoring of Reactivity in Subcritical Reactor Systems*, Royal Institute of Technology KTH, [Ph.D. Dissertation](#), Stockholm, Sweden, 2010.
- [6] F. MELLIER, *The MUSE Experiments for Sub Critical Neutronics Validation*, [MUSE Final Report](#), CEA, 2005.
- [7] I. PÁZSIT and L. PÁL, *Neutron Fluctuations: a Treatise on the Physics of Branching Processes*, Elsevier, [ISBN 9780080450643](#), 2008.
- [8] D.S. O'KELLY, *Operation and Reactivity Measurements of an Accelerator Driven Subcritical TRIGA Reactor*, [Ph.D. Dissertation](#), University of Texas at Austin, 2008.
- [9] N. SJÖSTRAND, *Measurement on a Subcritical Reactor Using a Pulsed Neutron Source*, *Arkiv för Fysik* **11**, 1956.
- [10] G.R. KEEPIN et al., *Delayed Neutrons from Fissionable Isotopes of Uranium, Plutonium, and Thorium*, [Physical Review](#) **107/4**, 1957.
- [11] A. TALAMO et al., *Monte Carlo and Deterministic Calculation of the Bell and Glasstone Spatial Correction Factor*, [Computer Physics Communications](#) **183/9**, pp. 1904-1910, 2012.
- [12] A. TALAMO et al., *Impact of the Neutron Detector Choice on Bell and Glasstone Spatial Correction Factor for Subcriticality Measurement*, [Nuclear Instruments and Methods in Physics Research A](#) **668**, pp. 71-82, 2012.
- [13] A. TALAMO et al., *Correction Factor for the Experimental Prompt Neutron Decay Constant*, [Annals of Nuclear Energy](#) **62**, pp. 421-428, 2013.
- [14] M. HÖÖK, *Study of the Pulse Shape as a Means to Identify Neutrons and Gammas in a NE213 Detector*, [M.Sc. Thesis](#), UPPSALA Universitet (Sweden), 2006.
- [15] S. POZZI et al., *Pulse-height Distributions of Neutron and Gamma Rays from Plutonium-Oxide Samples*, [Nuclear Instruments and Methods in Physics Research A](#) **608**, pp. 310-315, 2009.
- [16] T.W. CRANE and M.P. BAKER, *Neutron Detectors*, in *Passive Nondestructive Assay of Nuclear Materials*, ed. by D. REILLY et al., Nuclear Regulatory Commission [NUREG/CR-5550](#), pp. 393-394 and 399, 1991.
- [17] G.F. KNOLL, *Radiation Detection and Measurement*, John Wiley & Sons, [ISBN 978-0470131480](#), 2010.
- [18] P. FILLIATRE et al., *Estimating the γ -Ray Contribution to the Signal of Fission Chambers with Monte Carlo Simulations*, [Nuclear Instruments and Methods A](#) **648**, pp. 228-237, 2011.

- [19] K. ASAI, [*Nuclear Design Study on Advanced Neutron Measurement System for D-T Burn Control in Fusion Experimental Reactor*](#), Nagoya University, 2007.
- [20] B. GESLOT et al., *Method to Calibrate Fission Chambers in Campbell Mode*, Idaho National Laboratory, [INL/CON-11-22128](#), 2011.
- [21] S. TAVERNIER, *Experimental Techniques in Nuclear and Particle Physics*, Springer, [ISBN 978-3-642-00828-3](#), 2010.
- [22] L. VERMEEREN et al., *Experimental Verification of the Fission Chamber Gamma Signal Suppression by the Campbell Mode*, [Proceedings of the ANIMMA Conference](#), Marseille, France, 2009.
- [23] N.R. CAMPBELL and V.J. FRANCIS, *A Theory of Valve and Circuit Noise*, [Journal of IEEE](#) **93/3**, 1946.
- [24] R.A. DUBIDGE, *Campbell Theorem – System Concepts and Results*, [IEEE Trans. Nucl. Sci., NS14, 1](#), 1967.
- [25] A. RAO, [Summary of High Energy Particle Detector Elements](#), 2016.
- [26] T.J. LANGFORD, *Event Identification in ³He Proportional Counters Using Risetime Discrimination*, [arXiv:1212.4724](#), 2012.
- [27] L.G. EVANS, *Pulse Train Analysis Applied to the Re-Evaluation of Deadtime Correction Factors for Correlated Neutron Counting*, University of Birmingham, [Ph.D. Dissertation](#), 2009.
- [28] A.E. PROFIO, *Experimental Reactor Physics*, John Wiley and Sons, [ISBN-13: 978-0471700951](#), pp. 160-167, 1976.
- [29] K. OTT et al., *Dead-Time Effects of Neutron Detectors due to Pulsed Radiation*, [Radiation Protection Dosimetry](#) **155/2**, pp. 125-140, 2013.
- [30] J.C. LIU et al., *Neutron Dosimetry at SLAC: Neutron Sources and Instrumentation*, Stanford Linear Accelerator Center, [SLAC-TN-91-3](#), 1991.
- [31] C. BERGLÖF et al., *Auto-Correlation and Variance-to-Mean Measurements in a Subcritical Core Obeying Multiple Alpha-Modes*, [Annals of Nuclear Energy](#) **38**, p. 198, 2011.
- [32] M.C. MOXON, [The Effect of Dead Times](#), 2003.
- [33] A. PATIL, *Dead Time and Count Loss Determination for Radiation Detection Systems in High Count Rate Applications*, [Ph.D. Dissertation](#), Missouri University of Science and Technology, 2010.
- [34] D.P. ROUX and J.C. ROBINSON, *Formulation of the Gamma Pile-Up Effect in Fission Counters*, Oak Ridge National Laboratory, [CONF-720607—12](#), 1972.
- [35] R. KURAMOTO et al., *Rossi- α Experiment in the IPEN/MB-01 Research Reactor*, [Brazilian Journal of Physics](#) **35**, 2005.
- [36] J.J. DUDERSTADT and L.J. HAMILTON, *Nuclear Reactor Analysis*, John Wiley & Sons, [ISBN 978-0471223634](#), pp. 216-218, 1976.
- [37] G.D. SPRIGGS, R.D. BUSCH, T. SAKURAI, and S. OKAJIMA, *The Equivalent Fundamental-Mode Source*, [Annals of Nuclear Energy](#) **26**, pp. 237-264, 1999.
- [38] A. TALAMO et al., *Alternative Definitions of Kinetic Parameters for Accelerator Driven Systems*, [Annals of Nuclear Energy](#) **44**, pp. 12-21, 2012.
- [39] D.B. PELOWITZ et al., *MCNP6 User's Manual, Code Version 6.1.1beta*, Los Alamos National Laboratory, LA-CP-14-00745, 2014.
- [40] W. UYTENHOVE et al., *Experimental Results from the VENUS-F Critical Reference State for the GUINEVERE Accelerator Driven System Project*, [IEEE Transactions on Nuclear Science](#) **59/6**, 2012.

- [41] A. TALAMO et al., *Reactivity Measurements in Subcritical Assemblies Combining MSM and Area Methods*, [ANS Winter Meeting](#), Anaheim, CA, USA, 2014.
- [42] A. TALAMO et al., *Pulse Superimposition Computational Methodology for Estimating the Subcriticality Level of Nuclear Fuel Assemblies*, [Nuclear Instruments and Methods in Physics Research A 606/3](#), pp. 661-668, 2009.
- [43] A. TALAMO, Y. Gohar and C. Rabiti, *Monte Carlo Modeling and Analyses of YALINA-Booster Subcritical Assembly Part II: Pulsed Neutron Source*, Argonne National Laboratory, [ANL-08/33](#), 2008.
- [44] A. TALAMO and Y. GOHAR, *Numerical Application of the Sjöstrand Method without the Pulses Superimposition Methodology*, [ANS Winter Meeting](#), Washington, DC, USA, 2015.
- [45] G.I. BELL and S. GLASSTONE, *Nuclear Reactor Theory*, Van Nostrand Reinhold Company, [ISBN 978-0442206840](#), 1970.

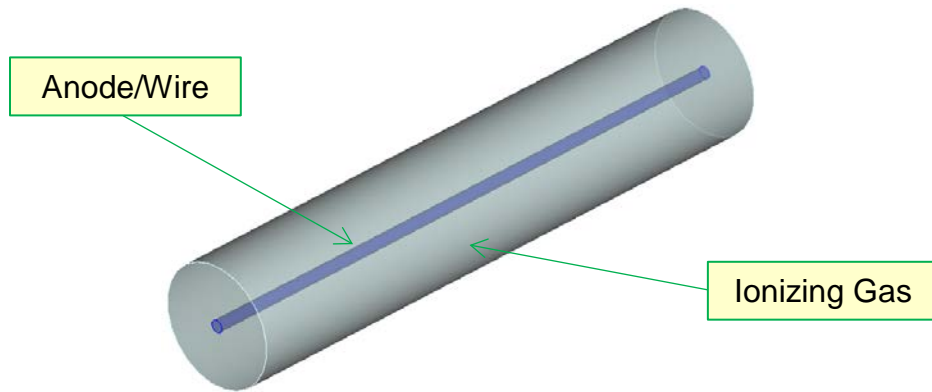


Figure 1. Illustration of the active zone of a typical gaseous ionization detector; the cathode (hidden in the figure) consists of a metallic hollow cylinder confining the ionizing gas

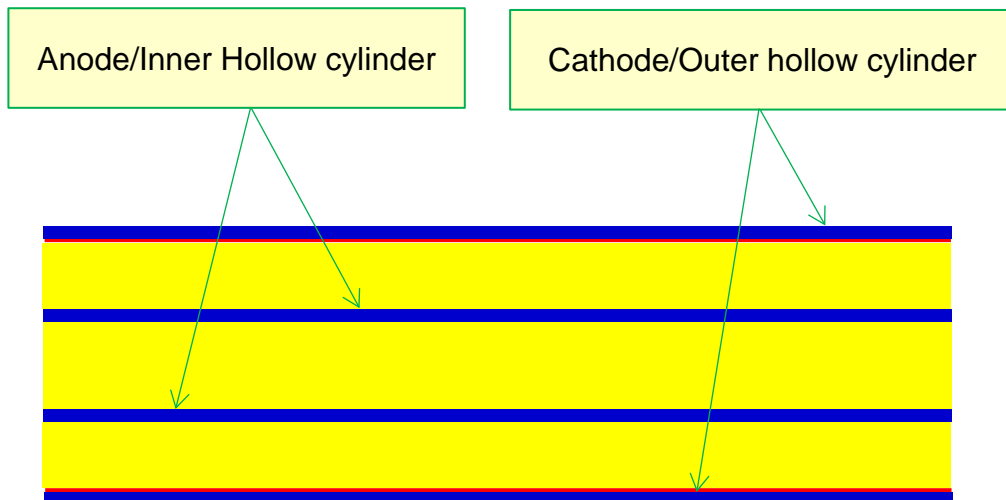


Figure 2. Illustration of the cross section of the active zone of a typical fission chamber; the ionizing gas, fissile, and metallic materials have yellow, red and blue colors, respectively

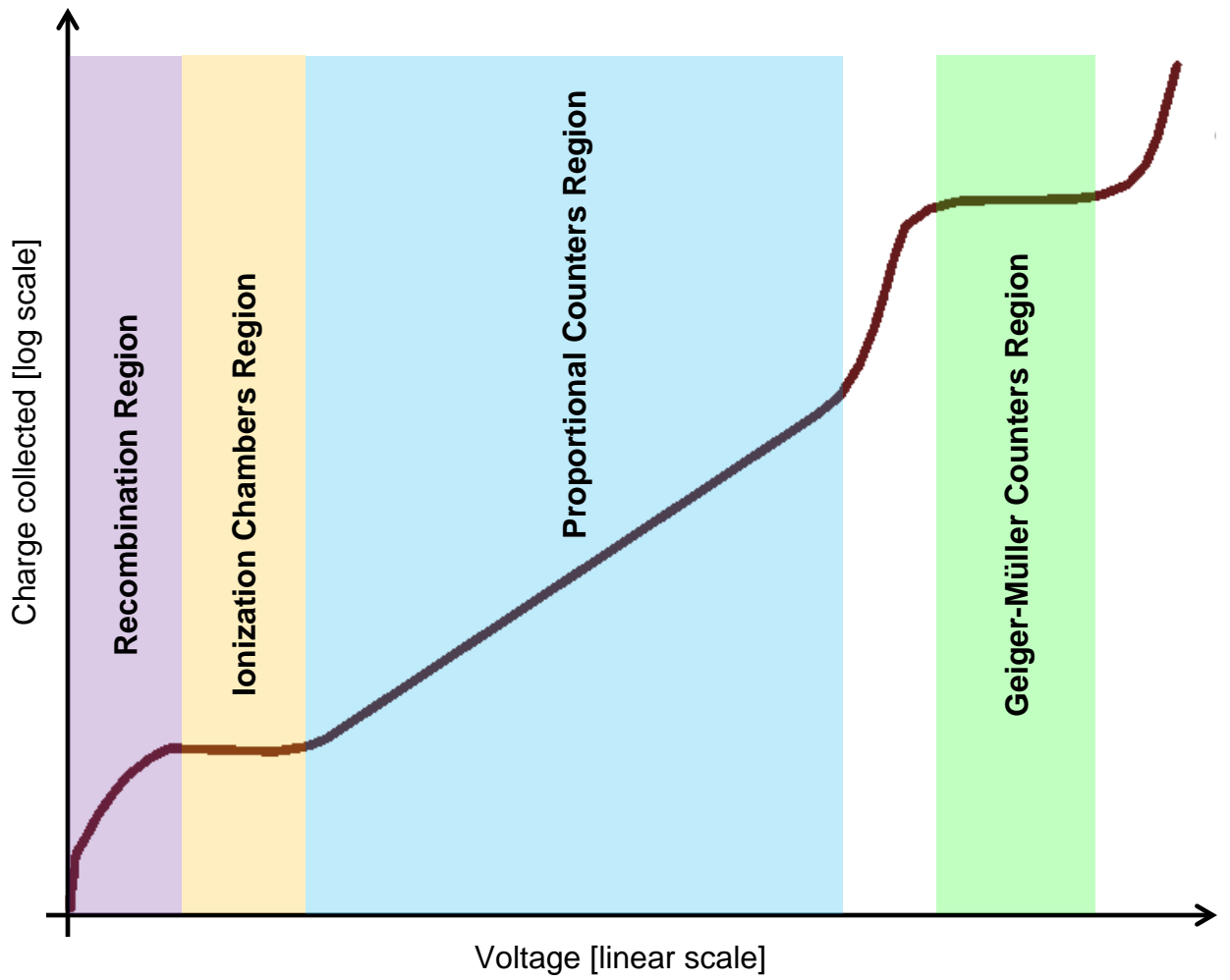


Figure 3. Variation of the collected charge as a function of the applied voltage in a gaseous ionization detector

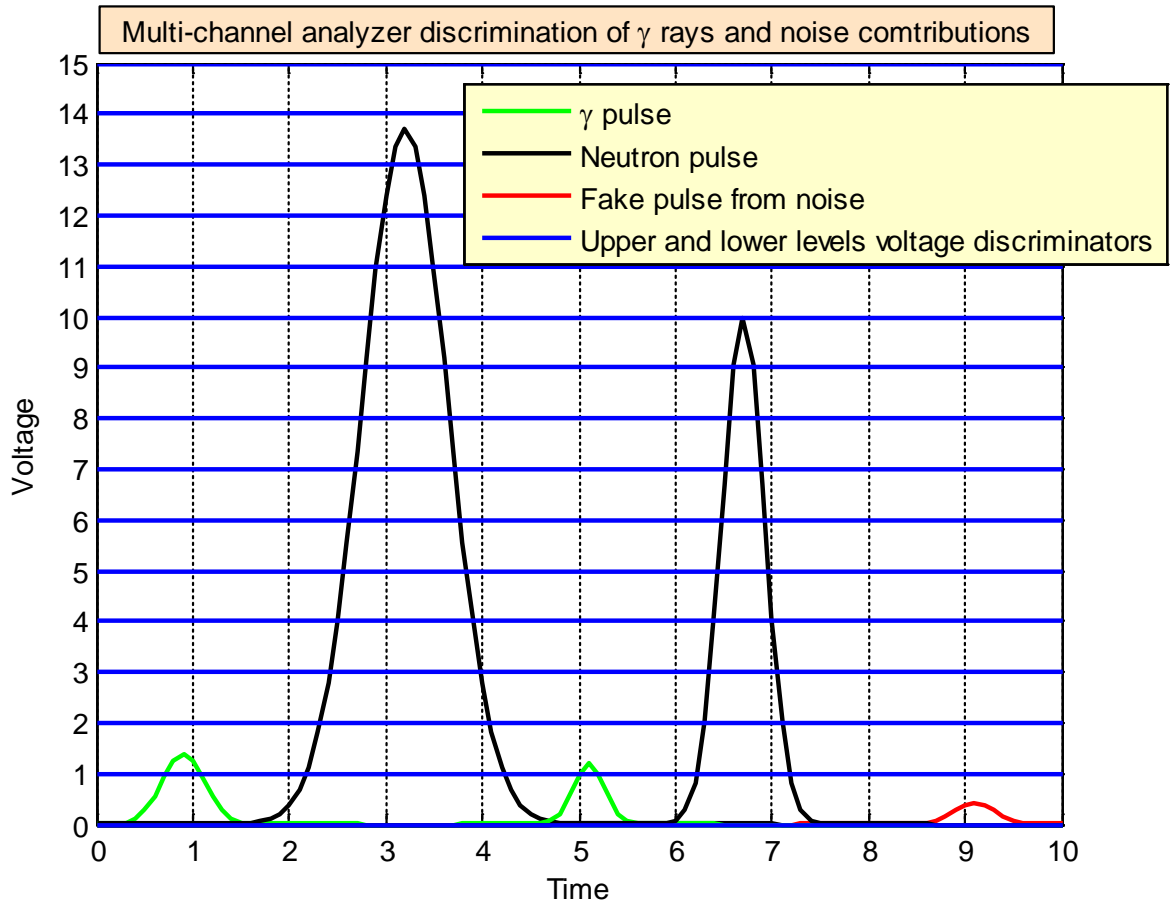


Figure 4. Amplifier output voltage as a function of time analyzed by a 15 channels MCA

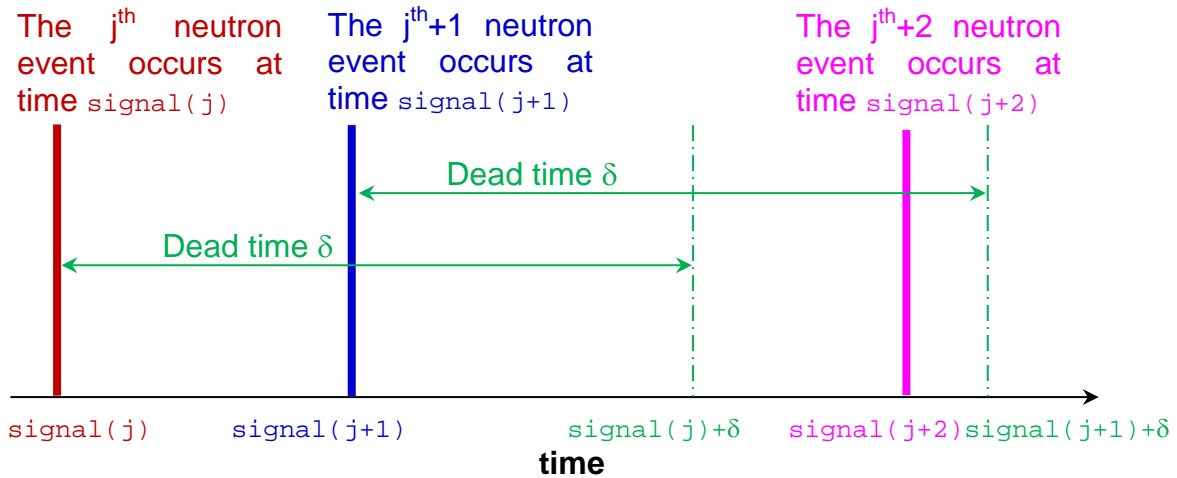


Figure 5. Illustration of the dead-time effect; signal in the time scale indicates a neutron event (ionization); in the non-paralyzable model, the detector scores $\text{signal}(j)$ and $\text{signal}(j+2)$ because $\text{signal}(j+2)$ is greater than $\text{signal}(j)+\delta$; in the paralyzable model, the detector only scores $\text{signal}(j)$ because $\text{signal}(j+2)$ is smaller than $\text{signal}(j+1)+\delta$; in both models the detector does not score $\text{signal}(j+1)$ because it is smaller than $\text{signal}(j)+\delta$

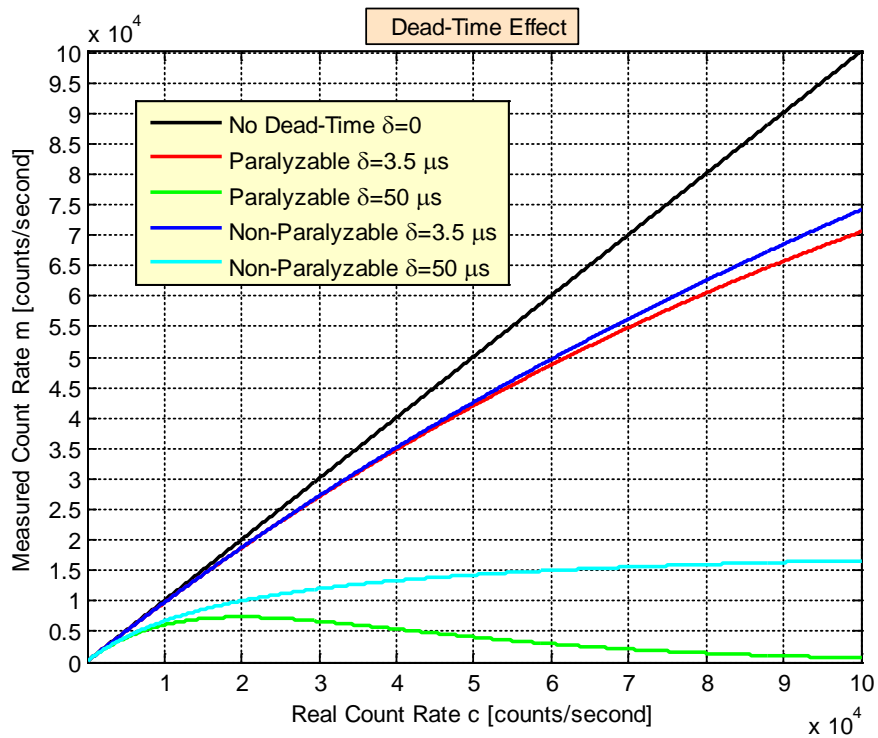


Figure 6. Illustration of the dead-time effect for δ equal to 3.5 and 50 μs when the average intensity of the neutron source is constant over time (Eqs. 1 and 2)

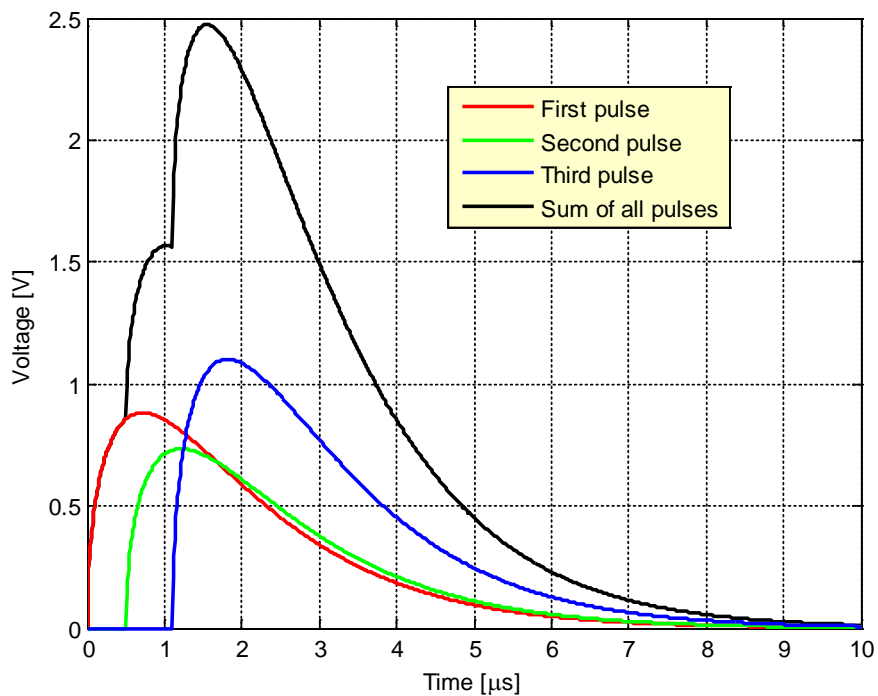


Figure 7. Illustration of the pile-up effect in the preamplifier output voltage as a function of time

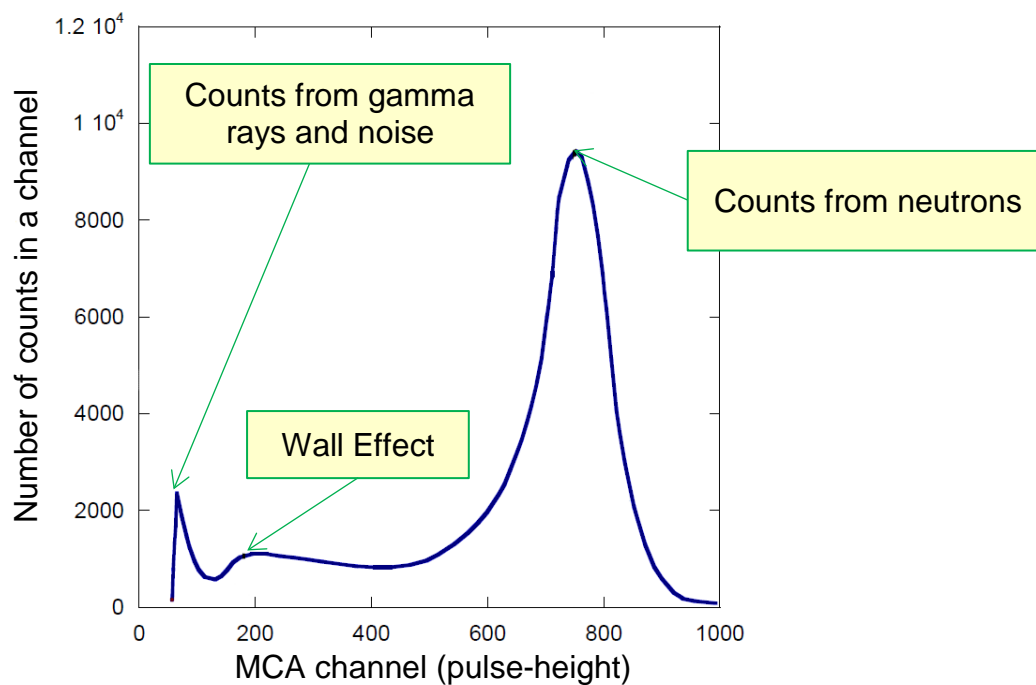


Figure 8. Pulse-height spectrum for a ^3He gaseous ionization chamber operated in pulse mode; the MCA has 1000 channels

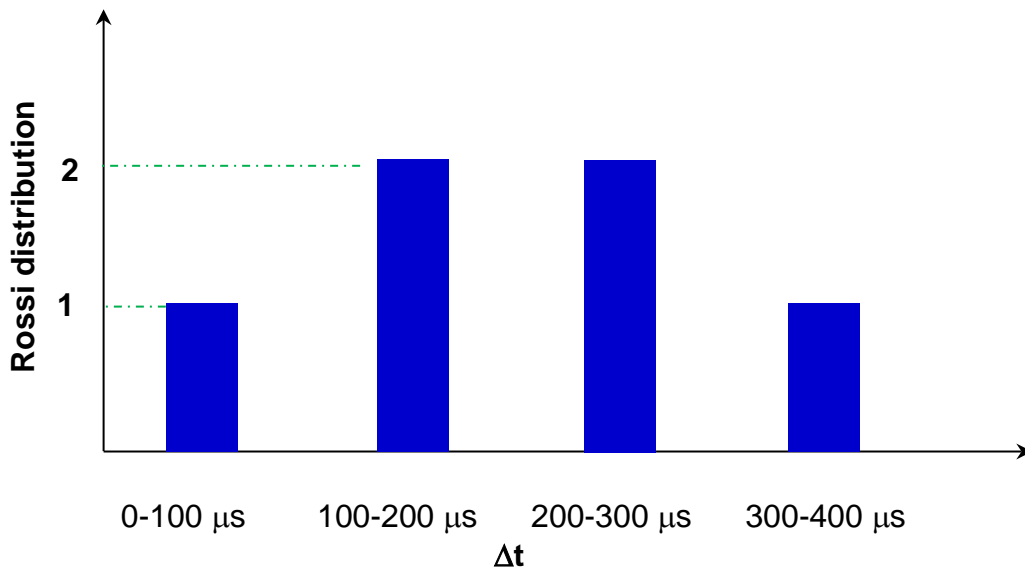
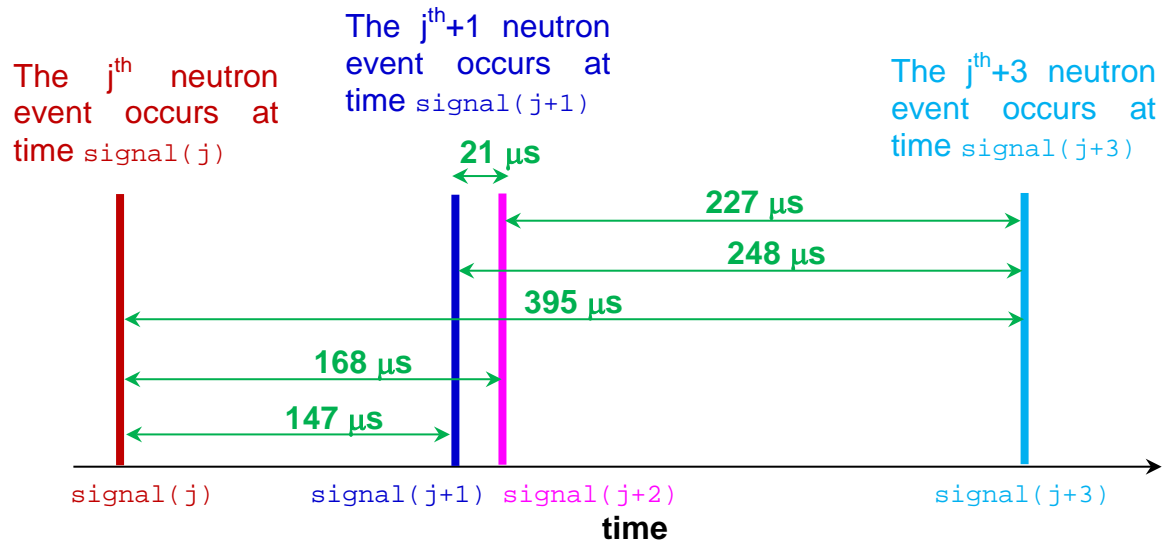


Figure 9. Illustration of the Rossi distribution (bottom plot) for a quadruplet of neutron events (top plot);³⁵ the left column of the Rossi distribution contains only one pair of neutron events with interval between 0 and 100 μs ; the second (from the left) column of the Rossi distribution contains two pairs of neutron events with interval between 100 and 200 μs ; the third column of the Rossi distribution contains two pairs of neutron events with interval between 200 and 300 μs ; the fourth column of the Rossi distribution contains only one pair of neutron events with interval between 300 and 400 μs

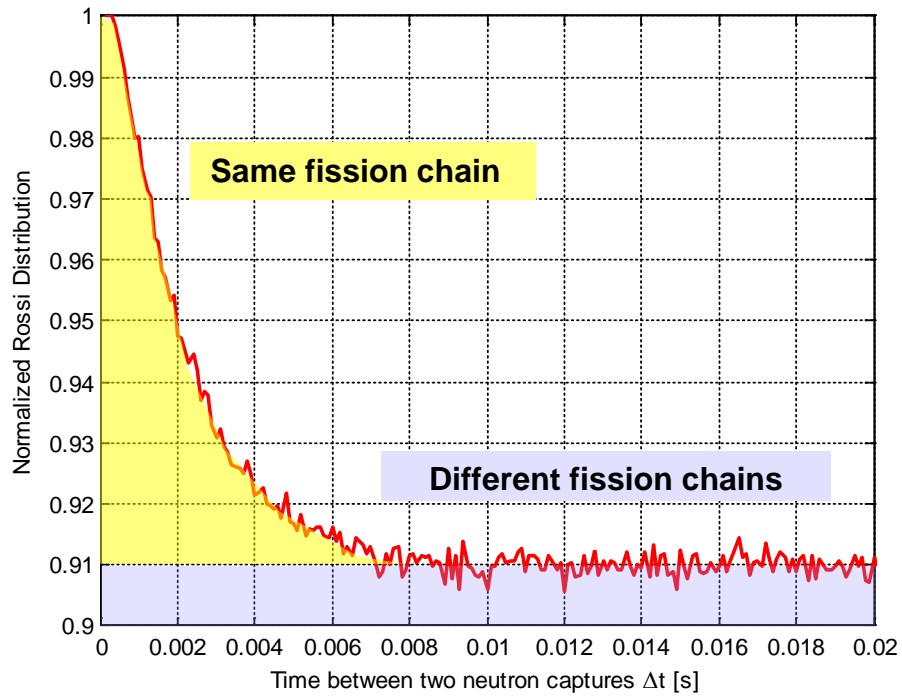


Figure 10. Illustration of a typical Rossi distribution

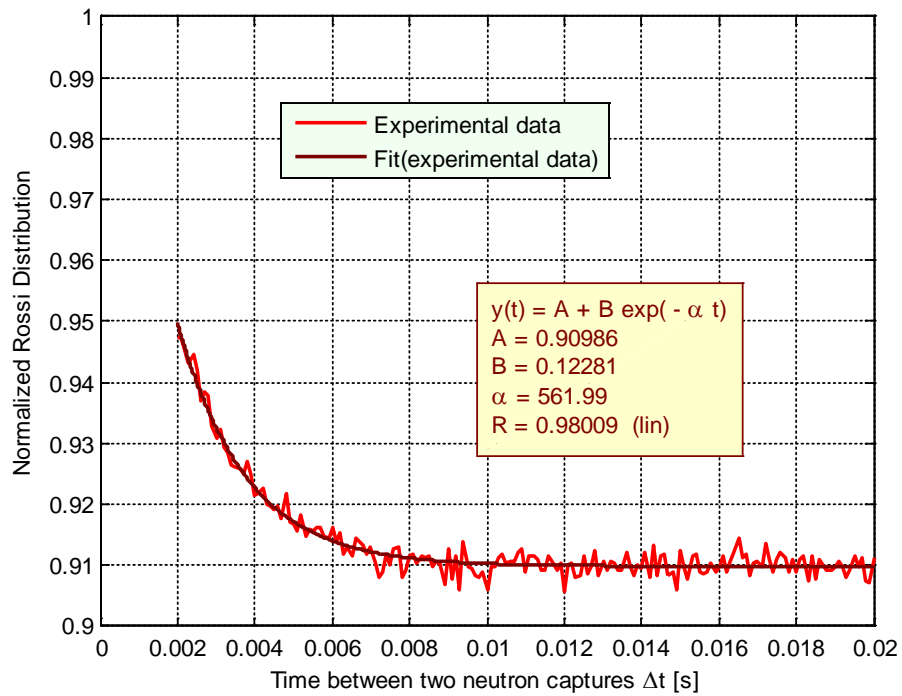


Figure 11. Illustration of the Rossi distribution fitting

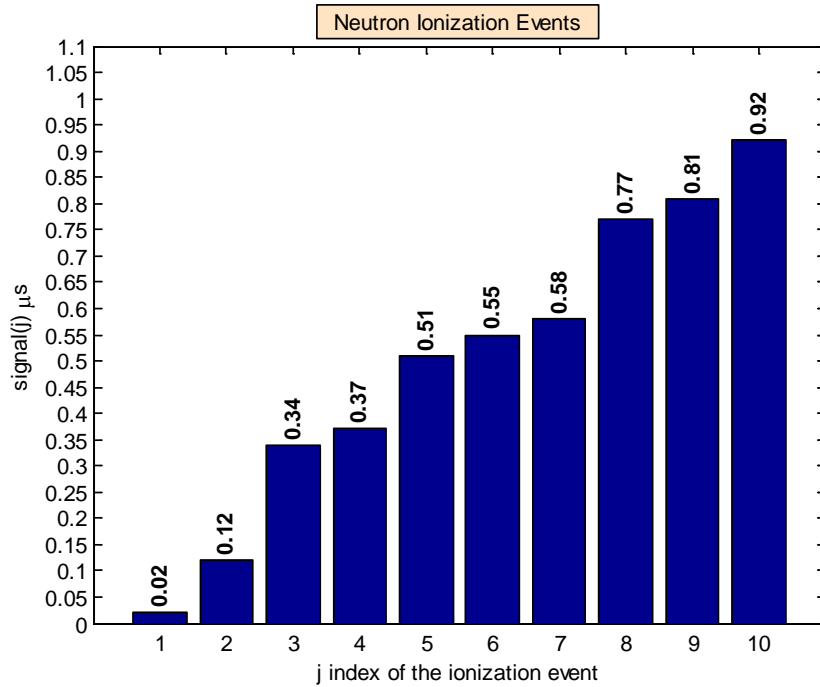


Figure 12. Illustration of 10 neutron events contributing to the Feynman Distribution

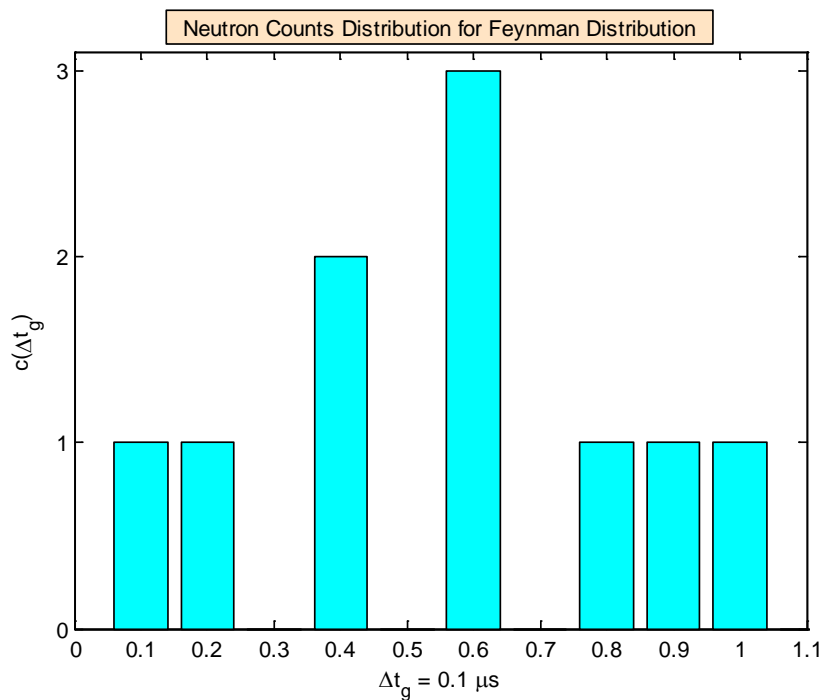


Figure 13. Illustration of the $c(\Delta t_g)$ array for the 10 neutron events displayed in Fig. 12 when the gate is $0.1 \mu\text{s}$; the left column contains neutron events between 0 and $0.1 \mu\text{s}$; the second column (from left) contains neutron events between 0.1 and $0.2 \mu\text{s}$; etc.

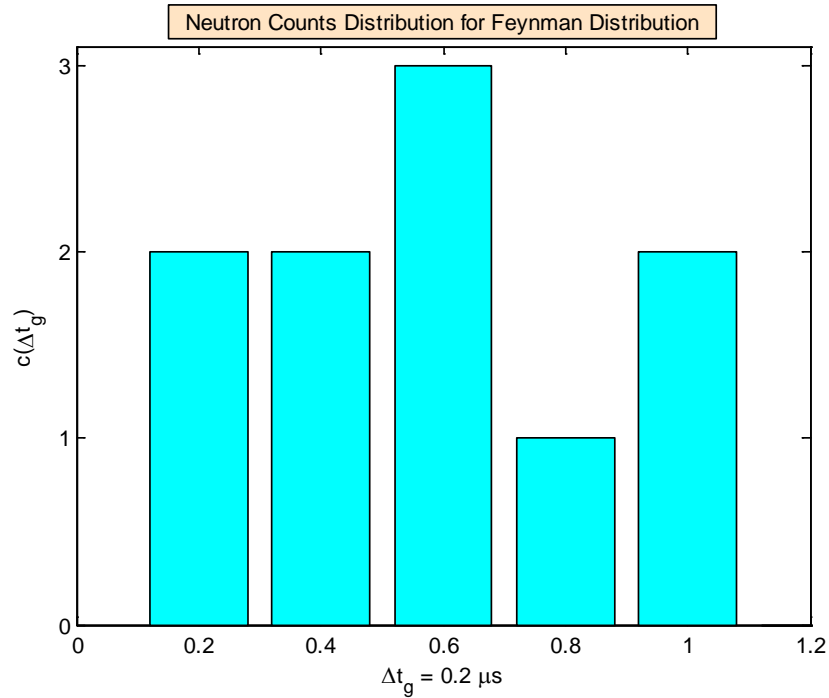


Figure 14. Illustration of the $c(\Delta t_g)$ array for the 10 neutron events displayed in Fig. 12 when the gate is $0.2 \mu\text{s}$; the left column contains neutron events between 0 and $0.2 \mu\text{s}$; the second column (from left) contains neutron events between 0.2 and $0.4 \mu\text{s}$; etc.

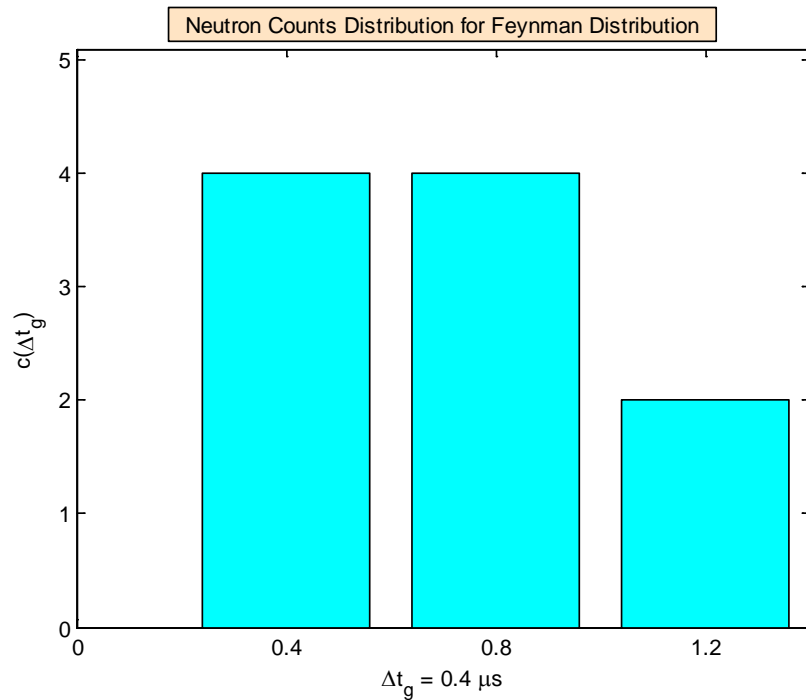


Figure 15. Illustration of the $c(\Delta t_g)$ array for the 10 neutron events displayed in Fig. 12 when the gate is $0.4 \mu\text{s}$; the left column contains neutron events between 0 and $0.4 \mu\text{s}$; the central column contains neutron events between 0.4 and $0.8 \mu\text{s}$; etc.

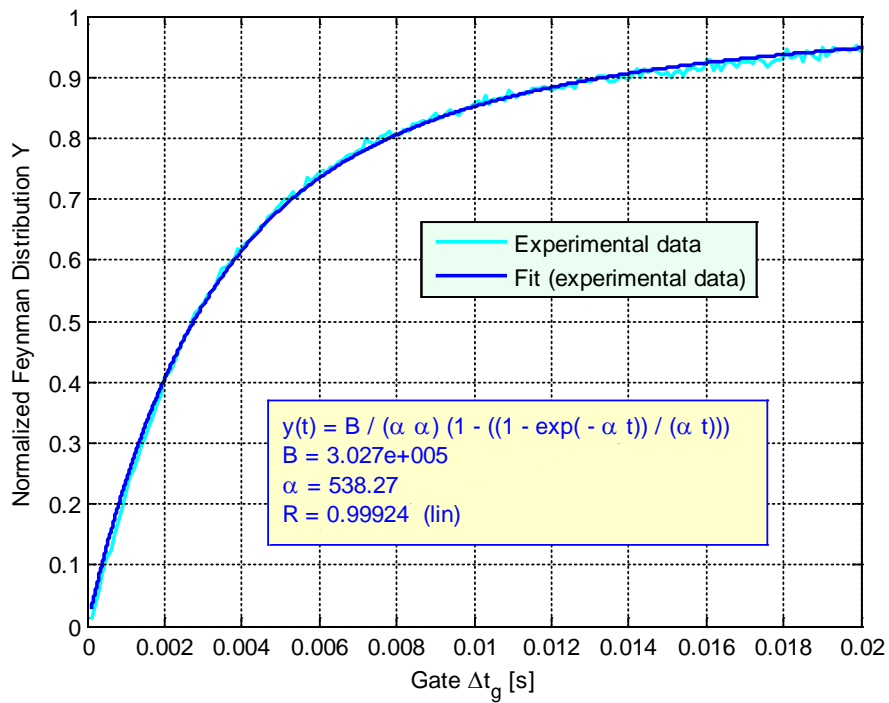


Figure 16. Illustration of a typical Feynman distribution and its fitting

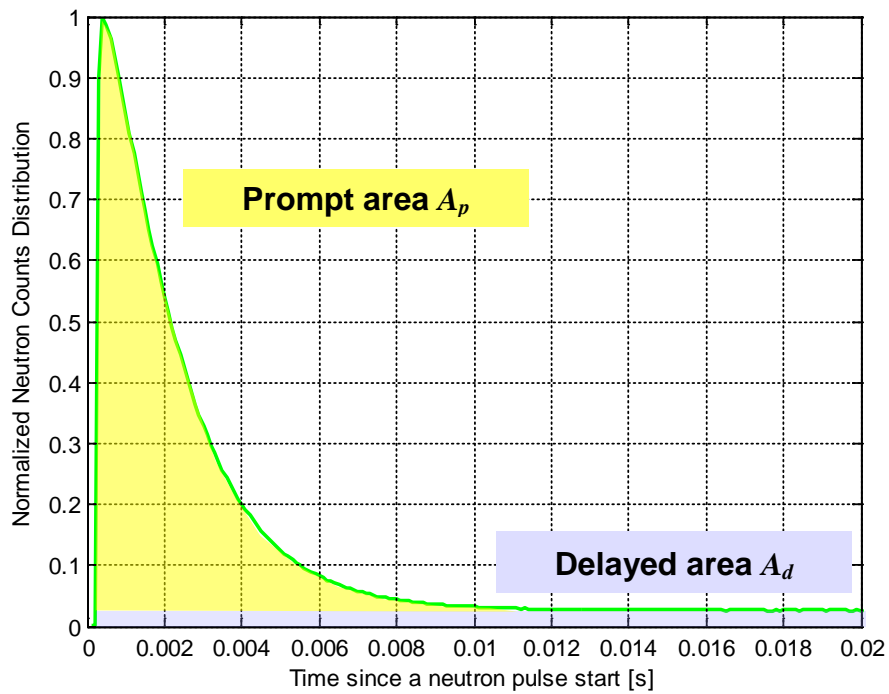


Figure 17. Illustration of a typical neutron counts distribution for an experiment with 20 ms pulse period. The figure plots the counts array

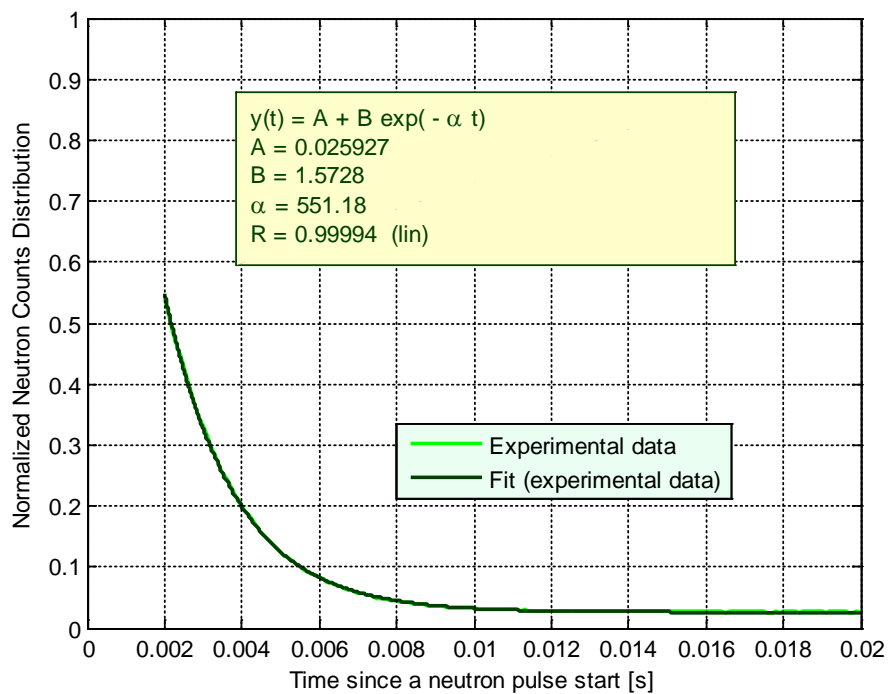


Figure 18. Illustration of the neutron counts distribution fitting

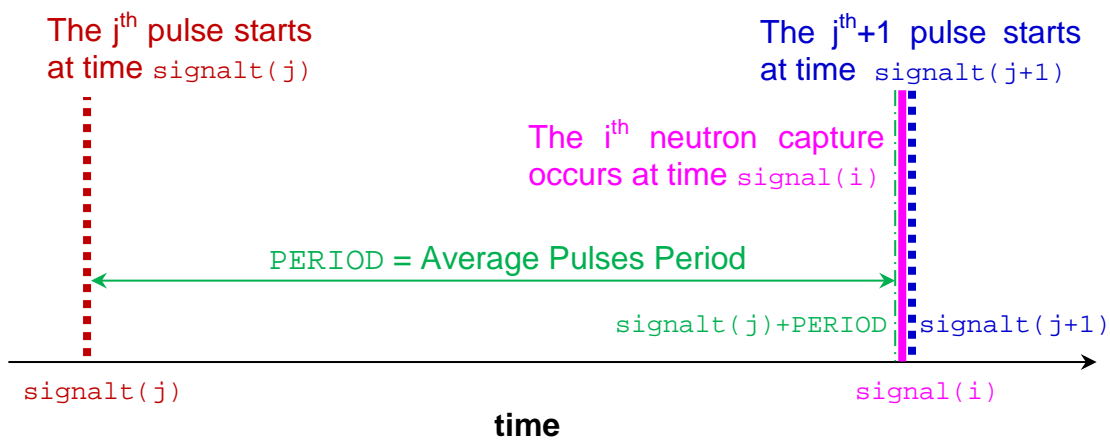


Figure 19. Illustration of a lost $\text{signal}(i)$ neutron capture event

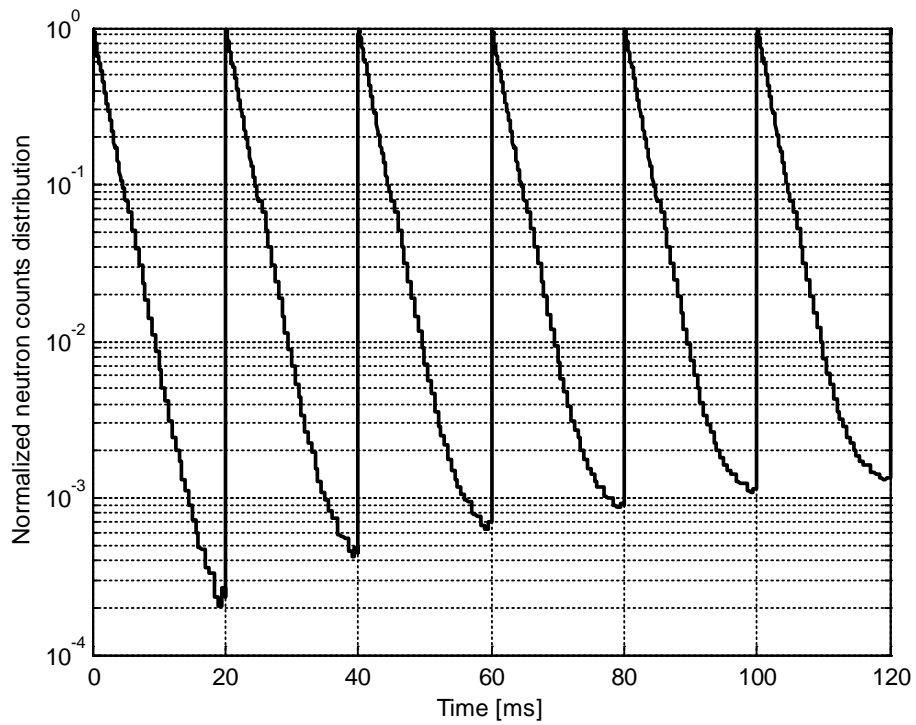


Figure 20. Neutron counts distribution for the first six pulses since the start of the accelerator

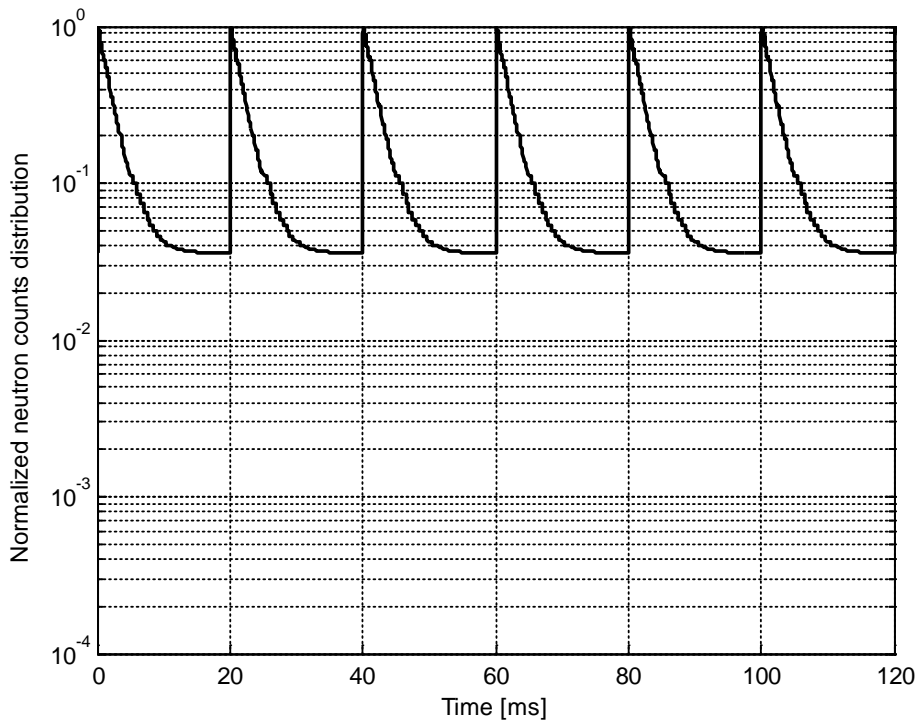


Figure 21. Neutron counts distribution for six pulses after 20,000 pulses since the start of the accelerator

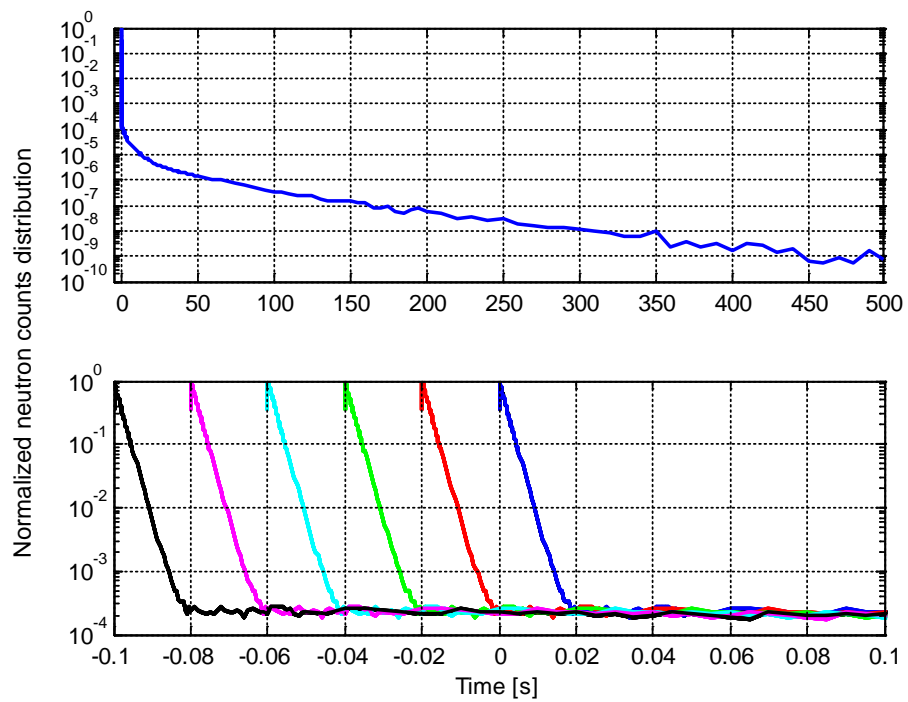


Figure 22. The top plot shows the simulated detector response, as a function of time, from a single pulse of the particle accelerator; the bottom plot shows the superimposition (shift) of six neutron pulses from -100 to 20 ms

Appendix A

MATLAB Script for Calculating the Neutron Counts from the Experimental Data Obtained with a Steady State External Neutron Source

```
% STEP 1 - file opening and reading
f=fopen('signal_file', 'r'); % it opens the signal file from the detector
signal=fread(f,2e7,'uint64','b'); % it reads the signal data from the binary (64bits) signal
file; data is stored into the signal array
fclose(f); % it closes the signal file

% STEP 2 - calculation of the measured neutron counts per second
TICK=12.5e-9; % time bin of the MCA
m=length(signal)/(signal(end)*TICK); % m = total counts / experiment duration [s]

% STEP 3 - calculation of the real neutron counts per second
DEADTIME=3.5e-6;
c=m/(1-m*DEADTIME); % Eq. 1; non-paralyzable model
```

Appendix B

MATLAB Script for Calculating the Rossi Distribution from the Experimental Data Obtained with a Steady State External Neutron Source

```
% STEP 1 - file opening and reading
f=fopen('signal_file', 'r'); % it opens the signal file from the detector
signal=fread(f,2e7,'uint64','b'); % it reads the signal data from the binary (64bits) signal
file; data is stored into the signal array
fclose(f); % it closes the signal file

% STEP 2 - calculation of the Rossi distribution
TICK=12.5e-9; % time bin of the MCA
signal = signal*TICK; % convert signal data to seconds
TBIN = 0.1e-3; % time bin for Rossi distribution [s] (0.1 milliseconds)
InvTBIN = 1/TBIN; % variable to optimize speed (divisions penalize loops efficiency)
x = [TBIN:TBIN:50e-3]'; % analyze 50 milliseconds with 0.1 milliseconds time bin
maxx = length(x); % length of the Rossi distribution array
maxs = length(signal); % length of the signal array
maxs1 = maxs-1; % adjust the length of the signal array for the 'for' loop
rossi = zeros(maxx,1); % initialize the Rossi distribution to zero
for i=1:maxs1
    for j=i+1:maxs1
        interval=signal(j)-signal(i); % time difference between two neutron captures
        h=ceil(interval*InvTBIN); % index in the Rossi distribution
        if (h<maxx)
            rossi(h,1)=rossi(h,1)+1; % update the distribution with the two neutron captures
        else
            break; % break the cycle if the second neutron capture occurs after 50 ms
        end
    end
end
end
plot(x,rossi); % plot the Rossi distribution
```

Appendix C

MATLAB Script for Calculating the Feynman Distribution from the Experimental Data Obtained with a Steady State External Neutron Source

```
% STEPS 1 - constants definition
TICK      = 12.5e-9;          % time bin of the MCA
MAXT      = 50e-3;           % maximum gate time [s]
TBIN      = 0.1e-3;         % time bin of the gate [s]
gate      = [TBIN:TBIN:MAXT]'; % array of gate times [s]; analyze MAXT with TBIN time bin
maxg      = length(gate);    % length of the Feynman distribution and gate arrays

% STEP 2 - file opening and reading
f=fopen('signal_file', 'r'); % it opens the signal file from the detector
signal=fread(f,2e7,'uint64','b'); % it reads the signal data from the binary (64bits) signal
file; data is stored into the signal array
fclose(f); % it closes the signal file

% STEP 3 - calculation of the Feynman distribution
signal = signal*TICK; % convert signal data (timestamps) to seconds
maxs = max(find(signal<(signal(end)-MAXT))); % length of the signal array shorted by MAXT
values
[feynman1 feynman2]=function_feynman(signal,gate,maxs,maxg,'Experiment date');
plot(gate,feynman1); % plot the deterministic Feynman distribution
plot(gate,feynman2); % plot the random Feynman distribution

% FUNCTION FEYNMAN
function [feynman1,feynman2] = function_feynman(signal,gate,maxs,maxg,mystring)
maxi=1e8;
feynman1=zeros(maxg,1); % initialize the deterministic Feynman distribution to zero
feynman2=zeros(maxg,1); % initialize the random Feynman distribution to zero
for m=1:maxg % loop on the Feynman distribution index
    c=hist(signal(1:maxs),[gate(m)/2:gate(m):signal(maxs)]); % initialize the neutron counts array
    feynman1(m,1)=var(c)/mean(c)-1; % calculate the Feynman distribution
    % feynman1(m,1)=(sum((c-mean(c)).^2)/length(c))/mean(c)-1;
    if (feynman1(m,1)<0) % debug if Feynman distribution value < 0
        warning('Negative deterministic distribution value; m=%d\n',m);
    end
    clear c;
    c=zeros(maxi,1); % initialize the neutron counts array to zero
    for i=1:maxi % loop on the number of random samples
        s=ceil(rand*(maxs)); % open the gate at index s
        j=s+1; % first neutron capture since the gate opening
        while ((signal(j)-signal(s))<gate(m)) % loop on all neutron captures within the gate
            c(i)=c(i)+1; % increment the neutron counts array
            j=j+1; % increment the neutron capture index
        end
    end
    feynman2(m,1)=var(c)/mean(c)-1; % calculate the Feynman distribution
    if (feynman2(m,1)<0) % debug if Feynman distribution value < 0
        warning('Negative random distribution value; m=%d\n',m);
    end
    clear c;
    fprintf('%s m=%d\n',mystring,m);
end
```

Appendix D

MATLAB Script for Calculating the Neutron Counts Distribution from the Experimental Data Obtained with a Pulsed External Neutron Source

```
% STEP 1 - constants definition
BEFF = 750e-5; % delayed neutron fraction value
TICK = 12.5e-9; % time bin in the signal array [s]; it is set by the MCA electronic circuit
TBIN = 4e-6; % time bin in the counts array [s]; it is an arbitrary value
InvTBIN = 1/TBIN; % InvTBIN defined for optimization speed ; multiplication is performed faster
than division

% STEP 2 - files opening and reading
f=fopen('signal_file', 'r');
signal=fread(f,2e7,'uint64','b'); % detector signal; it stores the time (in TICK unit) when one
neutron is captured in the detector
fclose(f);
f=fopen('signalt_file', 'r');
signalt=fread(f,2e7,'uint64','b'); % pulse signal; it stores the time (in TICK unit) when a
neutron pulse begins
fclose(f);

% STEP 3 - unit conversion
% it converts the signal and signalt data from TICK (time bins) unit to seconds
signal = signal * TICK;
signalt = signalt * TICK;

% STEP 4 - calculation of the average period
% it calculates the average of the pulses periods from the signalt array data (timestamps)
PERIOD=mean(diff(signalt(2:end,1))); % it discharges the first value since it may be far off

% STEP 5 - definition of the counts array (neutron captures) histogram
% the counts array has a number of elements (length) equal to the average period (PERIOD) divided
by the arbitrary TBIN value
x = [TBIN:TBIN:(PERIOD+TBIN)]; % time domain
counts = zeros(length(x),1); % neutron counts distribution; all values are set to zero
```

```

% STEP 6 - updated of the counts array (neutron captures) histogram
i=1;
while signal(i) <= signalt(2) % skip all signal data (neutron captures) before the beginning of
the second pulse
    i=i+1;
end;
minmax=min(max(signal),max(signalt)); % it sets a common maximum time (maxtime) between pulses
beginning and detector data
maxsignalt=max(find(signalt<minmax)); % it finds the pulse before the common maximum time
maxtime=signalt(maxsignalt); % it sets the time end
j=1;
lost=0; % this variable stores the counts lost due to not equal periods, Fig. 19
while signal(i) < maxtime % it processes all signal data until maxtime
    TimeFromLastTriggerPulse = signal(i)-signalt(j); % it calculates the time since the pulse
    started
    if (TimeFromLastTriggerPulse <= PERIOD) && (TimeFromLastTriggerPulse>0) % true if the
    signal value (capture time) is within the current pulse period
        h=ceil(TimeFromLastTriggerPulse*InvTBIN); % the ceil function rounds number toward
        positive infinity
        counts(h)=counts(h)+1; % it scores one neutron count
        i=i+1; % it increases the signal index
    else
        if signal(i) >= signalt(j+1) % true if the signal value (neutron capture time) is
        smaller than the starting time of the next pulse
            if j < length(signalt) % true if the last pulse (end of the experiment) is not
            reached
                j=j+1;
            else
                break; % this instruction is executed only if the last pulse is reached (it
                breaks the while cycle)
            end
        else
            % this lost originates because the pulses periods are not exactly equal and there might be a
            small deviation from the average value; if the period of the current pulse (given by
            signalt(j+1)-signalt(j)) is greater than the average period it may happen that
            TimeFromLastTriggerPulse > PERIOD but signal(i) < signalt(j+1) (this scenario is illustrated in
            Fig. 19); if the current pulse period is smaller than the average period, then the j index is
            incremented without any lost count
            lost=lost+1;
            i=i+1;
        end
    end
end
end

% STEP 7 - calculation of the measured neutron counts per second
c=counts/(j-1); % it normalizes the neutron counts to the total number of pulses minus one

% STEP 8 - calculation of the effective multiplication factor
N=length(c)-3; % it discharges the last 3 values, which might be affected by the not equal
periods effect (see step 6)
% it calculates the effective multiplication factor by the area method Eq. 29
% the delayed area takes into account the last 10% of the c array which covers the last 10% of
the pulse period
keff=1/(1+BEFF*(sum(c(1:N,1))-N*mean(c(round(0.9*N):N,1)))/(N*mean(c(round(0.9*N):N,1))));

% STEP 9 - plotting the neutron count distribution
plot(x,c/TBIN); % plot the neutron counts distribution [counts per second]

```

Appendix E

MCNP Input for Calculating Rossi and Feynman Distributions for a Californium Neutron Source

```
c      Cf neutron source for 1 hour
mode  n
print 40 50 60 72 98 102 115 120
lost  1
nps   50e6
c      tally      runtpe  mctal  runtpe  rendezvous
prdmp 5e6      5e6      0      1      5e6
phys:n 30 30 $ upper energy cutoff 30 MeV - analog capture
cut:n  3600e8 0 0 0 $ tally for 1 hour
ptrac  buffer=1000 file=asc max=2e9 write=all coinc=lin event=cap type=n
tally=8
rand   seed=17373738176025
sdef   pos=0 0 -6.5
       axs=0 0 1
       vec=0 0 1
       rad=d1
       ext=d2
       erg=d3
       tme=d4
si1    h  0 0.235
sp1    -21 1
si2    h  0 0.1
sp2    -21 0
sp3    -3  1.025  2.926
si4    h  0 3600e8 $ neutron emission for 1 hour
sp4    d  0 1
c      TALLY CARDS
f8:n   1700
fc8    PTRAC TALLY
ft8    cap -1 -1 2003
```

Appendix F

MATLAB Script for Calculating the Neutron Counts Distribution from a Single Pulse MCNP PTRAC Data (Timestamps)

```
N=2000; % number of time bins in the counts array
counts=zeros(N,1); % neutron counts array
PERIOD=20e-3; % pulses period, 20 milliseconds
TBIN=T/N; % time bin
for i=1:size(signal,1)
    for j=1:N
        if ((signal(i,1)>(j-1)*TBIN)&&(signal(i,1)<j*TBIN))
            counts(j,1)=counts(j,1)+1;
            break;
        end
    end
    if (signal(i,1)>PERIOD)
        k=i-1; % k stores the number of neutron captures within the period
        break; % size(signal,1) is the total number of neutron captures
    end
end
% sum of the terms in the right side of Eq. 33
counts=counts+(size(signal,1)-k)/N;
```


Appendix G

MCNP Input for Calculating the Neutron Counts Distribution from a Single Pulse from a D-D or D-T Neutron Source

```

mode      n
print     40 50 60 72 98 102 115 120
lost      1
nps       10e6
c         tally      runtpe  mctal  runtpe  rendezvous
prdmp     2e6        2e6      0      1      2e6
phys:n    30 30 $ upper energy cutoff 30 MeV - analog capture
cut:n     500e8 0 0 0
ptrac     buffer=1000 file=asc max=2e9 write=all coinc=lin event=cap type=n
          tally=8
rand      seed=17373738176025
sdef      pos=0 0 0
          axs=0 0 1
          vec=0 0 1
          rad=d1
          ext=d2
          dir=d3
          erg=fdir d4
          tme=d5
si1       h 0 0.235
sp1       -21 1
si2       h 0 0.1
sp2       -21 0
c         Deuterium-Deuterium Neutron Source
c         Handbook on Nuclear Activation Data, Technical Reports Series No.273,
c         p. 116, IAEA, Vienna (1987).
c         energy-angle distribution for d+d reaction 250 keV
si3 a     -1.      -.966   -.866   -.707   -.500   -.259   0
          .259     .5      .707   .866   .966   1
sp3       4.75    4.5     4.1     3.6     3.0     2.5     2.3
          2.5     3.2     4.25    5.7     6.9     7.3
ds4       2.01    2.03    2.07    2.14    2.24    2.37    2.51
          2.66    2.80    2.94    3.04    3.11    3.14
c         Deuterium-Tritium Neutron Source
c         Handbook on Nuclear Activation Data, Technical Reports Series No.273,
c         p. 132, IAEA, Vienna (1987).
c         energy-angle distribution for d+t reaction 250 keV
si3 a     -1.      -.966   -.866   -.707   -.500   -.259   0
          .259     .5      .707   .866   .966   1
sp3       122.    122.5   124.    125.    127.    129.5   132.
          135.    137.    139.    141.    142.    142.5
ds4       13.1    13.125  13.23   13.4    13.6    13.83   14.13
          14.4    14.675  14.9    15.075  15.2    15.24
si5       h 0 500 500e8
sp5       d 0 1 0
c         TALLY CARDS
f8:n      1700
fc8       PTRAC TALLY
ft8       cap -1 -1 2003

```

Appendix H

MCNP Input for Calculating the Neutron Counts Distribution from a Single Pulse from a Proton Neutron Source

```
mode      n h /
phys:n    101 101      $ upper energy cutoff 101 MeV - analog capture
phys:h    101 101      $ upper energy cutoff 101 MeV - analog capture
c lca     j j j
cut:n     500e8 0 0 0  $ max time - energy cutoff - analog capture
cut:h     500e8 0 0 0
cut:/     500e8 0 0 0
nps       1e12
c         tally      runtpe  mctal  runtpe  rendezvous
prdmp    2e6        2e6      0      1      2e6
print    40 50 60 72 98 102 115 120
lost     1
ptrac    buffer=1000 file=asc max=2e9 write=all coinc=lin event=cap type=n
         tally=8
rand     seed=17373738176025
sdef     par=h
         pos=0 0 0
         axs=0 0 1
         vec=0 0 1
         rad=d1
         dir=1
         erg=100
         tme=d2
si1      h 0 2
sp1      -21 1
si2      h 0 500 500e8
sp2      d 0 1 0
c        TALLY CARDS
f8:n     1700
fc8      PTRAC TALLY
ft8      cap -1 -1 2003
```

Appendix I

MCNP Input for Calculating the Neutron Counts Distribution from a Single Pulse from an Electron Neutron Source

```
mode      n e p
phys:n    201 201          $ upper energy cutoff 201 MeV - analog capture
phys:e    201              $ upper energy cutoff 201 MeV
phys:p    201 j j -1      $ upper energy cutoff 201 MeV - analog production
c lca     j j j
cut:n     500e8 0 0 0    $ max time - energy cutoff - analog capture
cut:e     500e8 1 0 0
cut:p     500e8 0.01 0 0
mphys     on
nps       1e12
c         tally      runtpe  mctal  runtpe  rendezvous
prdmp     2e6        2e6      0      1      2e6
print     40 50 60 72 98 102 115 120
lost      1
ptrac     buffer=1000 file=asc max=2e9 write=all coinc=lin event=cap type=n
          tally=8
rand      seed=17373738176025
sdef      par=e
          pos=0 0 0
          axs=0 0 1
          vec=0 0 1
          rad=d1
          dir=1
          erg=200
          tme=d2
si1       h 0 2
sp1       -21 1
si2       h 0 500 500e8
sp2       d 0 1 0
c         TALLY CARDS
f8:n      1700
fc8       PTRAC TALLY
ft8       cap -1 -1 2003
```



Nuclear Engineering Division

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov



Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC