DETERMINANDO LA VIDA MEDIA DEL MUON A PARTIR DE DATOS DE UN DETECTOR DE RADIACIÓN CHERENKOV EN AGUA

LABORATORIO AVANZADO

F. Cabrera

19 de marzo de 2021

RESÚMEN

Se realizo un estudio de los datos recopilados por el experimento LAGO, que detecta muones por medio de la radiación cherenkov. Se seleccionaron los datos donde aparecen pulsos dobles, atribuidos al decaimiento del muon. Se analizaron estos datos ajustándolos a una distribución exponencial negativa para determinar la vida media del muon.

1. Introducción

1.1. Radiación Cherenkov

La radiación de Cherenkov se produce por el paso de partículas cargadas en un medio a una velocidad superior a la velocidad de la luz en ese medio. Los muones provenientes de radiación cósmica que bombardean la tierra constantemente producen esta radiación en niveles muy bajos al cambiar del medio del aire al agua ya que la velocidad de la luz en el agua es inferior a la del aire.

1.2. Proyecto LAGO

El proyecto LAGO es un proyecto internacional de astrofísica de partículas donde participan grupos de múltiples instituciones de América Latina. Entre estas participa la Escuela de Ciencias Físicas y Matemáticas de la USAC. Este proyecto proporciona el detector Cherenkov que consiste en un tanque de agua aislado a la luz y con un tubo foto multiplicador en su interior capaz de detectar muy finos pulsos de radiación Cherenkov.

Se selecciónaron las señales que superaran el umbral de -150 mV y el tubo multiplicador dispuesto a una ganancia de 1.50 V. Los datos fueron recopilados el día 20 de marzo de 2018 entre 17:37:03 y 18:37:04 (GMT-6) en intervalos de 10 minutos. Se realizaron aproximadamente 60 mil mediciones por cada intervalo de 10 minutos. Por medio del software de recolección de datos se generaron 6 archivos que contienen la información de los pulsos que superaron el limite del umbral.

Los pulsos detectados por el experimento representan muones que interactuan con el agua del tanque. Dentro de estos solamente una pequeña cantidad se espera que decaiga mientras estan dentro del tanque. Esto se expresaria por medio de un doble pulso detectado. Se desarrollo un software que discriminara los pulsos simples de los pulsos dobles.

1.3. Discriminación de datos

Del experimento LAGO se disponen 6 archivos codificados en binario. Dentro de estos están las señales de cada pulso detectado según las condiciones establecidas. En promedio se observan 60 mil pulsos por archivo. En la figura 1 podemos ver un pulso normal detectado en el experimento. Según estuvo dispuesto, el equipo recopilo los datos 40 ns antes de que se detectara el pulso y hasta 9 mil ns después de detectado este.

El equipo de recopilación de datos guardaba la señal del tubo foto multiplicador una vez cada 8 ns, de manera que los datos están dispuestos como una lista ordenada donde el valor por cada casilla corresponde al nivel de voltaje en el detector.

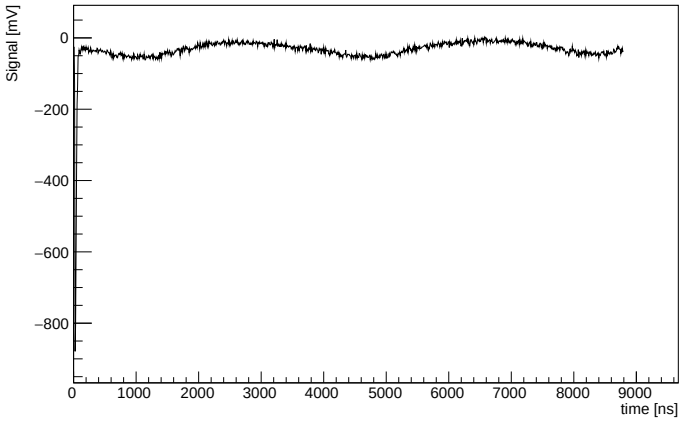

Figura 1: Ejemplo de un pulso detectado por medio del experimento LAGO.

Para discriminar los datos de pulsos dobles de entre todos los pulsos se utilizo un programa que sigue el siguiente algoritmo:

Algoritmo 1: Detección de pulsos dobles

Inicio

1. Definir variables principales.
   Leemos el archivo de datos.
   th_i, Nivel de referencia para detectar el segundo pulso.
   $Datos$, Colección de pulsos en el archivo.
   $pulso$, Datos de cada pulso individual.
   $doble?$, variable lógica predeterminada como falsa.
   t_0, punto donde el primer pulso termina
2. Determinar:
   pc_i, Cantidad de datos en el archivo,
   ps_i, Cantidad de datos por pulso.
3. **para** $i = 1$ **a** pc_i **hacer**
   - $pulso = Datos(i)$.
   - Determinar: ps_i,
   - Determinar: $t_0 = 0$
4. **para** *j = 5* ps_i **hacer**
   - **si** $pulso(j) > -75$ **entonces**
     - Determinar: $t_0 = j$
     - Pare
5. **para** $k = t_0$ ps_i **hacer**
   - **si** $pulso(k) < th_i$ **entonces**
     - Determinar: $doble? = Verdad$
     - **Salida:** (k-5)*8
     - Pare.

Fin

Se decidió utilizar un ciclo que determinara el fin del pulso inicial, ya que si se desidia a saltar hasta una cantidad de pasos fija después del primer pulso se puede dar la situación de que se pasen por alto pulsos dobles validos que se encuentren en este rango. Podemos ver un ejemplo en las figuras 2 que corresponden al pulso numero 57999 del archivo terminado en 173703. donde podemos observar claramente que este es un pulso doble valido y ocurre dentro de los primeros 100 ns, que se corresponden a los primeros 12 datos recopilados.

En una primera iteración se consiguieron aproximadamente 206 pulsos dobles por archivo, midiéndolos con un umbral de detección de -700 mV. Sin embargo al hacer un primer ajuste a los resultados con este algoritmo de discriminación se

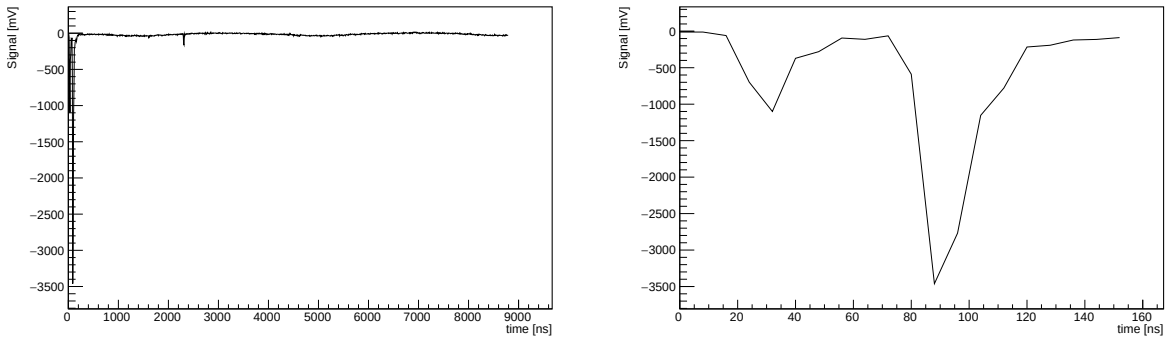

Figura 2: Pulso 57999 del archivo 173703. Podemos observar un pulso doble y un acercamiento que nos permite verificar que este si es un pulso doble aunque sucede en un tiempo muy corto.

alcanzo el resultado deseado con un 13 % de desviación de la medida conocida, con un resultado de $\tau = (1898 \pm 146)$ [ns], y una $\chi^2/ndf = 19{,}67/17$, como se observa en la figura 6.

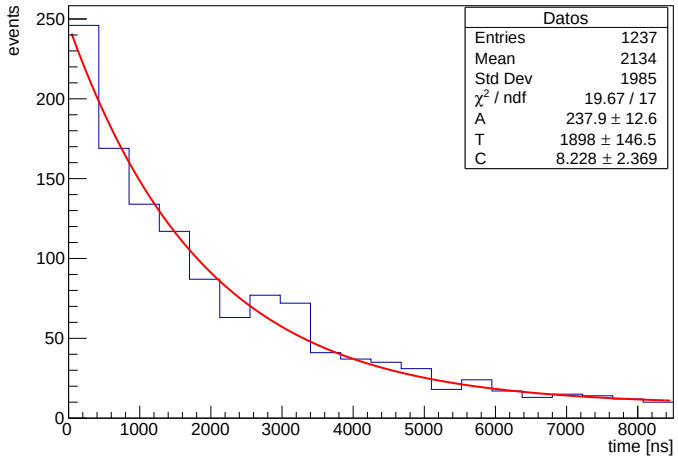

Figura 3: Ajuste preliminar de los datos utilizando el algoritmo 1

Para refinar la medida observamos algunos de los pulsos extremos y surgieron eventos como el que podemos ver en la figura 4. Sorprendidos de que un pulso singular quedara registrado realizamos una ampliación a la forma del pulso. En esta figura podemos observar en el rango 40 - 60 que el flujo experimenta un pico, sin embargo no podemos atribuir esto a un segundo pulso.

Se propuso entonces que el primer ciclo del filtro, el que determina donde acaba el primer pulso no utilice el mismo valor de umbral que el que se utilizara para determinar donde empieza el segundo pulso. Se utilizo un valor fijo para este umbral del primer ciclo. Para determinar el valor óptimo del umbral que determina donde termina el primer pulso observamos varias figuras de pulsos individuales, de manera fortuita el pulso 1 del archivo 182704 presento una energía bastante baja lo que permitió observar la forma del fondo de ruido electrónico que podemos ver en la figura 5. Podemos observar que el ruido de fondo oscila entre los -50 y los -100 mV, por lo tanto se escogió que el umbral del primer filtro fuera fijo a -75 mV. De esta manera podemos asegurarnos de que el detector esperara a que la señal baje hasta el nivel del ruido de fondo antes de empezar a buscar un segundo pulso.

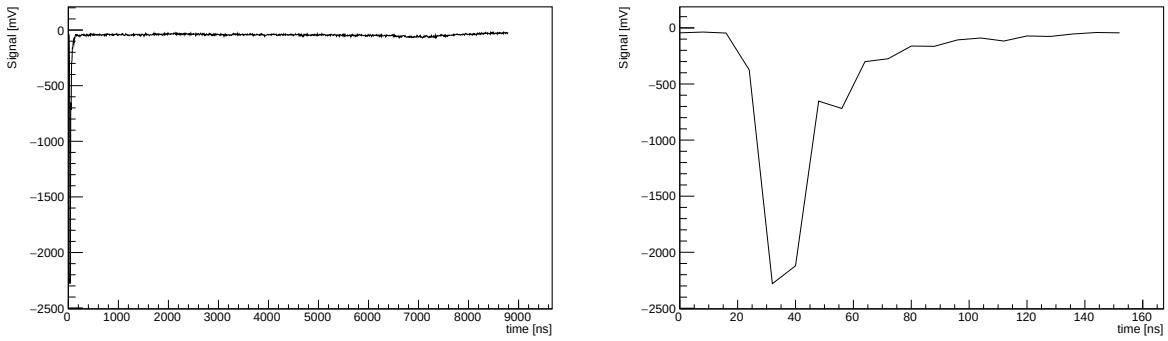

Figura 4: Pulso 88441-182704, evento singular que paso por el filtro propuesto en el algoritmo 1 con el umbral inicial igual al umbral de detección de -700 mV, a la derecha una ampliación del mismo.

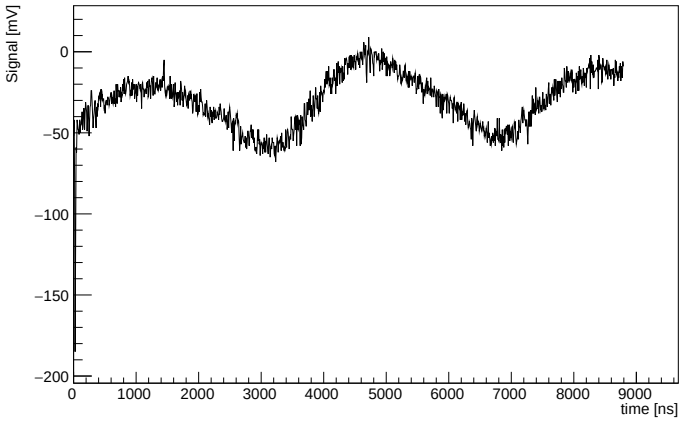

Figura 5: pulso 1-182704

Entonces se analizaron nuevamente los datos de los 6 archivos y se redujo la cuenta a unos 198 eventos por archivo aproximadamente, con un umbral de detección de -700 mV. Con estos nuevos datos se realizo un ajuste, como podemos ver en la figura 7.

2. Resultados

A partir de los datos de pulsos dobles obtenidos se realizaron un histograma por cada archivo de datos, a -700 mV como umbral de detección del segundo pulso. Posteriormente estos histogramas se sumaron para tener la totalidad de datos de pulsos dobles presentes en los 6 archivos.

A este histograma final se le realizo un ajuste utilizando la herramienta de software ROOT, siguiendo la ecuación de decaimiento exponencial:

$$y = A \exp \frac{-t}{T} + C\,, \tag{1}$$

donde las variables A y C dependen de la cantidad de datos de la muestra y del ruido de fondo presente en las mediciones, respectivamente. La variable T corresponde entonces al tiempo de vida medio del muon. Para realizar el ajuste se subministraron los valores de $A = 300$, $T = 2000$ y $C = 5$.

Con estas mismas condiciones se realizaron los ajustes con el algoritmo que discrimina el primer pulso con un umbral igual al que se usa para detectar el segundo pulso, fig 6 y con el umbral fijo en -75, fig 7. A partir del segundo ajuste se determino la vida media del muon como $\tau = (2127 \pm 173)$ ns, con una $\chi^2/ndf = 14{,}88/17$ Esta medida difiere del valor conocido de la vida media del muon por un 3 %.

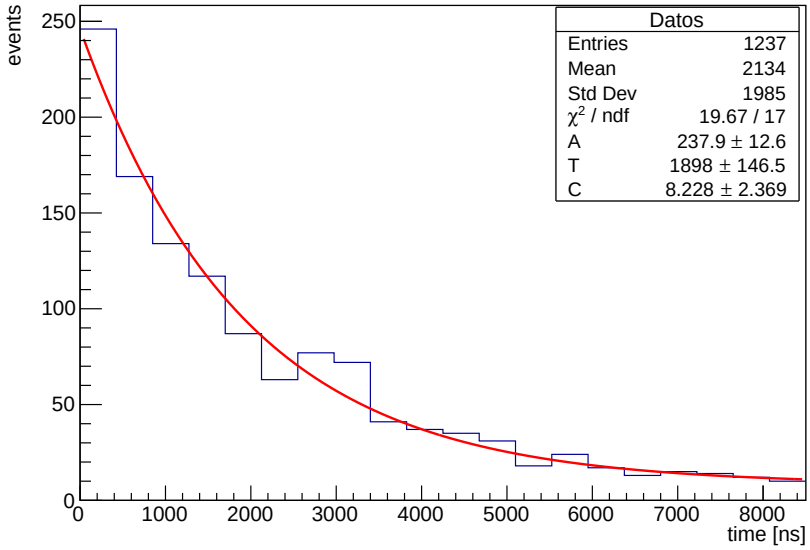

Figura 6: Ajuste preliminar de los datos utilizando el algoritmo 1 con umbral de discriminación del primer pulso en -700.

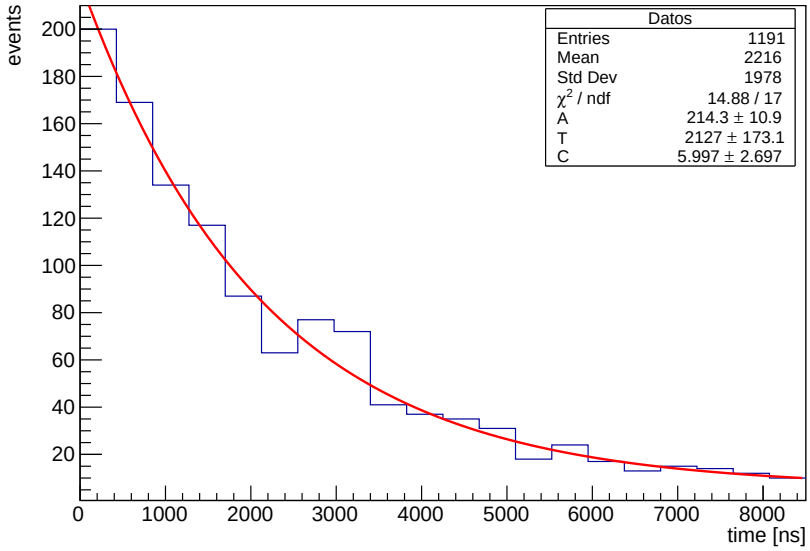

Figura 7: Ajuste de los datos utilizando el algoritmo 1 con el umbral de discriminación del primer pulso a -75 y el umbral para detectar el segundo pulso a -700

Anexos

Modificación incluida en el código original de `paa_reader` para detectar pulsos dobles.

```
/*************************************************************************/
/*        MODIFICACIÓN PARA LEER TODOS LOS PULSOS                        */

/* Declaramos variables importantes */
5int pc_i, ps_i, tl_i, low, event, t;
pc_i = mi_archivo.GetPulseEntries();      // Pulsos por archivo
ps_i = mi_archivo.GetPulseSize();         // Puntos por pulso
tl_i = cmd_args.GetLArgThreshold();    // Triger

bool Doble = false;                       // Verifica si hay doble pulso

std::vector<int> * Double_pls_list = new std::vector<int>;    // Lista de pulsos dobles


std::cout << "\n Procesing Data... " << std::endl;
std::cout << "Trigger used to get double pulses: " << tl_i << std::endl;

/* Hacemos un ciclo para cada dato */
for( int i = 0; i < pc_i; i++){
    pulseData_v = mi_archivo.GetPulse(i); // Va tomando los datos de cad pulso
    event = 1;  // Ponemos 0 eventos
    low = 0;    // Ponemos el contador a 0
    /*  Determinamos el tiempo en el que baja la señal   */
    /* Como los puntos antes del triger son 5 */
    /* Este ciclo quita el primer pulso para que solo detectemos el segundo */
    for(int j = 5; j < pulseData_v->size(); j++){
        int temp1;
        temp1 = pulseData_v->at(j); // at(j) == [0][j]
        if (temp1 > -75){ // Este es con el primer th fijo a -75
        //if (temp1 > tl_i){ // este es con el primer th = tl_i
            low = j;
            break;
        }
    }

    /* Hacemos un ciclo dentro del pulso, buscando el segundo pulso */
    for( int k = low; k < pulseData_v->size(); k++){
        int temp;
        temp = pulseData_v->at(k);
        if (temp < tl_i){
            event = event + 1;
            t = (k-5)*8;
            pulseData_t->push_back(t);
            break;
          }
    }
    /* Si hay 2 eventos guardamos el numero del pulso */

    if(event == 2){
    Double_pls_list->push_back(i);
    std::cout << i << " "<< low << " "<< t <<std::endl;
    Doble = true;
    }
}
/*************************************************************************/
```