

## GM計数管を比例計数管として使用する、パルス波高分析

### ■放射線はエネルギー

放射線計測実験では、「放射線が数えられる」ことを教えるが、「放射線はエネルギーである」ことを教える機会は少ない。シンチレーション式放射線計測器は $\gamma$ 線のエネルギーを測定できるが、自然にある放射線源では短時間の測定が難しいため、教育現場では実施しにくい。

GM計数管は、そのままでは放射線のエネルギーを測定できないが、同じガス電離を利用した比例計数管では放射線のエネルギーを知ることができる。違いはガス増幅率だけなので、GM管の構造はそのままでも、印加電圧を下げればガス増幅率は低下し、比例計数管として動作することが期待できる。この方式の利点は、自然にある放射線源に対しても、10分以内の短時間でスペクトルが得られる点にある。

ここでは、プログラミング言語 Python を使用して主に低印加電圧域でのパルス波高分析に適用するため、とりあえず 60 秒間の測定からパルスの波高分析を行い、グラフ表示するとともにスペクトル・データをファイルに出力するプログラムを紹介する。測定時間を長くするには、単に「録音時間」を変えればよい。

### ■パルス波高分析のプログラム

以下では、プログラムの流れとポイントを解説する。

冒頭では、必要なライブラリーを呼び込む。

```
import pyaudio #マイク入出力処理
import numpy as np
import matplotlib.pyplot as plt #グラフ処理
```

まず、入力ストリームを開き、バッファーには CHUNK 単位で取り込む。録音時間は 60 秒とした。frames は全データで、リストとして用意し、バッファーからリストに追記を繰り返す。後処理の都合で、リストからアレイに変換してからストリームを閉じる。

```
CHUNK = 1024 #入出力の長さ
FORMAT = pyaudio.paInt16 #2Byte 整数
CHANNELS = 2 #ステレオ
RATE = 44100 #サンプリング周波数(Hz)
RECORD_SECONDS = 60 #録音時間(秒)

p = pyaudio.PyAudio()
stream = p.open(format=FORMAT,
                 channels=CHANNELS,
                 rate=RATE,
                 input=True, #入力のみ
                 frames_per_buffer=CHUNK)

print("** recording")
frames = [] #空の list を用意

for i in range(int(RATE * RECORD_SECONDS // CHUNK)):
    #CHUNK 単位で切り捨て
    data_raw = stream.read(CHUNK) #サンプリング
    frames.append(data_raw) #60sec 分を加算
    frames_int16 = np.array(frames) #numpy アレイに変換

print("** done recording")

stream.stop_stream() #ストリームを閉じる
stream.close()
p.terminate()
```

```

data = np.frombuffer(frames_int16, dtype='int16') #バイトデータを
整数に変換
data_l = data[:,2] #ステレオのLチャンネルを取り出す
data_max = int(max(data_l)) #最大値を求める

nu = 0 #閾値以上を判断
nd = 0 #閾値以下を判断

spectrum = [1 for i in range(1001)] #list を 1000Channel に初期化
threshold = int(data_max / 10) #閾値を最大値の 1/10 とする
peak = int(threshold) #ピーク値の初期化

for k in range(len(data_l)): #全データを逐次処理
    dat = data_l[int(k)] #list の要素 k を取り出す
    if int(dat) > int(threshold): #閾値以上ならば
        nu = nu + 1 #閾値以上の回数を 1 増やす
        if nu == 1: #閾値以上の回数が 1 (初回の意) ならば
            nd = 0 #閾値以下の回数を初期化
        if int(dat) > peak: #これまでのピーク値以上ならば
            peak = int(dat) #ピーク値を更新
    if int(dat) <= int(threshold): #閾値以下ならば
        nd = nd + 1 #閾値以下の回数を 1 増やす
        if nd == 1: #閾値以下の回数が 1 (初回の意) ならば
            peakch = int(peak / data_max * 1000) #規格化
            spectrum[peakch] = spectrum[peakch] + 1 #1 加算
            peak = int(threshold) #ピーク値を再度初期化
            nu = 0 #閾値以上の回数を初期化

with open('spectrum_data.csv', mode='w') as fs: #ファイルを開く
    for m in range(1, 1001): #1000 チャンネル分を逐次処理
        spect = spectrum[int(m)] #list の要素 m を取り出す
        fs.write(str(m) + ',' + str(spect) + '\n') #末尾に追加

plt.subplot(211) #グラフの上半分を指定
plt.plot(data_l) #入力信号をプロット
plt.subplot(212) #グラフの下半分を指定
plt.plot(spectrum) #スペクトルをプロット
plt.show() #描画

```

バイトデータを整数に変換し、ステレオ録音の片チャンネルを抽出する。パルス波高が閾値

放射線教育支援サイト “らでい” > 教材 > 実験集 <https://www.radi-edu.jp/category/experiment>

を超えるとフラッグを立て、最大値の要素番号と値を更新していく。閾値を下回るとフラッグを降ろし、スペクトルの list に書き込む。

スペクトル・データは、全ピークの最大値を元に 1000 チャンネルに規格化し、csv ファイルに出力するとともに、波形とスペクトルをグラフに表示する。

## ■実験方法

印加電圧が可変の GM 計数装置から出力信号を取り出す。イヤホン・ジャックとパソコンのマイク入力端子をケーブルで接続するが、 $0.022\mu\text{F}$  程度のコンデンサを並列に入れて入力過大 (<5V) を防ぐこと。必要があれば、マイクのプロパティを開き、レベルを調整する。

印加電圧を測定し、印加電圧をパラメータとして測定する。印加電圧を下げながら、スペクトルの変化を調べる。GM 領域の下限付近では急に計数が少なくなり、細かい印加電圧の調整が必要で、測定時間も長くなる。

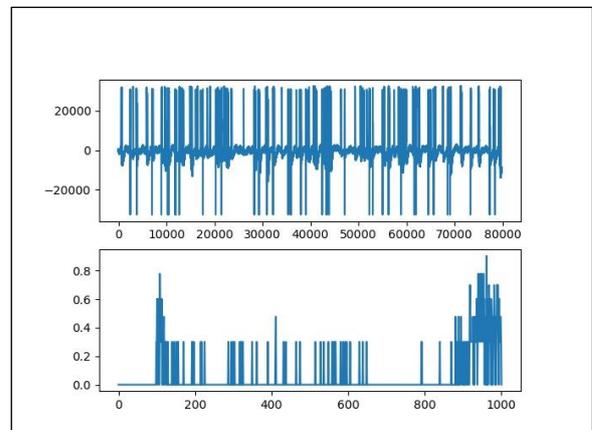


図 グラフの描画例

## ■注意点

GM 計数管が対象とする  $\beta$  線のエネルギーは最大エネルギー以下に広く分布する。最大エネルギーの約  $1/3$  にピークを持つ分布と言われており、本法でもスペクトルはブロードとなり、エネルギー分析は難しい。また、単一放射線源も得にくい。