
Container PMT Testing Data in ROOT Format



地点：物理楼 &220

更新时间：March 24, 2019

邮箱: zhaor25@mail2.sysu.edu.cn

版本: 1.0

目 录



1	General Introduction	3
1.1	how to access one ROOT file	3
1.2	structure of the root file	3
1.3	波形和积分	3
1.4	电压和增益的关系	4
1.5	μ 子的通量测量	4
A	读取 example 文件的 python 程序	6
A.1	python 读取一个波形	6
A.2	python 计算平均波形	7

第 1 章 General Introduction



For each PMT, the following information is available from a '.root' file:

1. waveforms raw data @light intensity about 0.1pe
2. waveforms raw data @light intensity about 1pe
3. auxiliary information about the testing conditions

If you are interested in the testing data of one specific PMT, you just need to process a root file which contains all testing raw data.

1.1 how to access one ROOT file

sever domain: data directory:

1.2 structure of the root file

公式 1.1: 增益计算

$$\delta = \frac{Q_{out}}{q} = \frac{\int UT}{Rq}. \quad (1.1)$$

其中 Q_{out} 是放大之后波形的电荷量， q 是未经放大的电荷量；我们假设每个事件的平均光电子数为 500，也就是 $q = 500 \times e$ 。¹

1.3 波形和积分

在 example 文件中，采样长度是 1999ns，也就是每 1999 点组成一个波形。我们需要算出没有信号时的基线的大小，再用基线值减去对应的信号大小积分出信号的面积。

对于每一个波形，我们都可以计算出一个积分面积，然后算出所有波形²积分面积的平均值³作为期望的电荷积分；再利用公式1.1就可以计算出当前工作电压下的增益大小。

¹e 是电子电荷量

²取决于事件数

³也可以填充直方图进行拟合



注意： 选择合理的时间范围进行积分，不要让信号超出积分的区间。

1.4 电压和增益的关系

调节不同的 PMT 工作电压，计算相应的增益大小，可以做出一条增益和电压相关的曲线，并在一定范围内可以用直线对其拟合。

1.5 μ 子的通量测量

当我们同时⁴在闪烁体两端的 PMT 看到信号时，我们认为测到了一个 μ 子的信号，通过统计一定时间的符合计数并结合探测器的有效探测面积就可以估算 μ 子的通量大小。

为了确定信号的到达时间，可以在 labview 中设置触发阈值读取触发时间；也可以在数据处理时寻找波形过阈时间。

example waveform

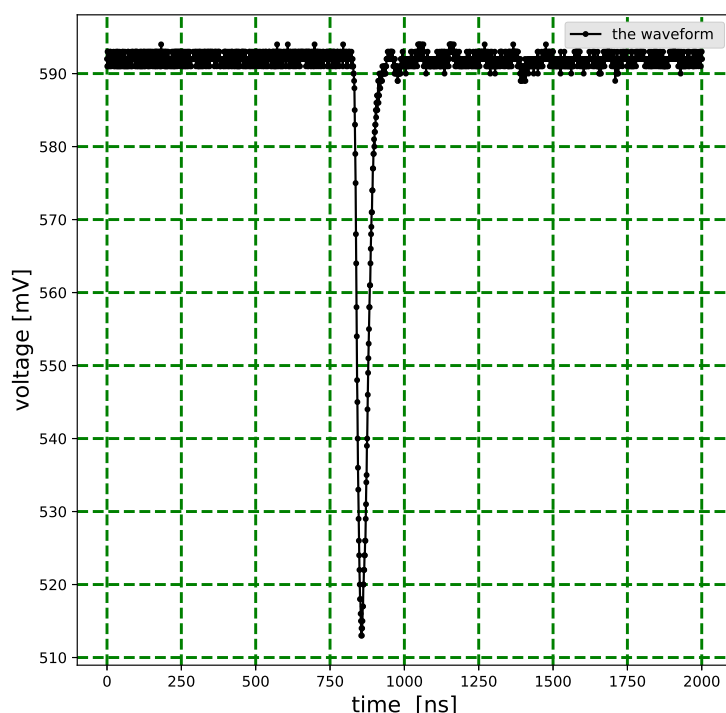


图 1.1: 一个 PMT 信号波形

⁴信号时间差小于一定范围



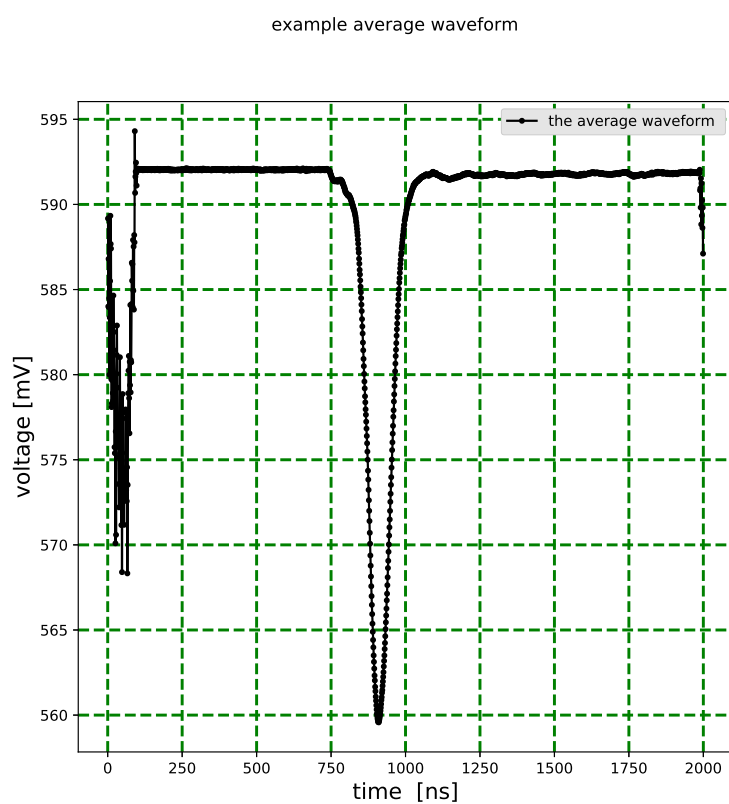


图 1.2: PMT 信号平均波形



附录 读取 example 文件的 python 程序



A.1 python 读取一个波形

```
1  #!/usr/bin/env python
2  # -*-coding:utf-8 -*-
3  import numpy as np
4  from scipy import interpolate
5  import pylab as pl
6  f = open('/media/tao/_dde_data/arlierwork/simulation/tao/exp220/example.
    lvm') #换成你自己的文件名
7  s = f.readline()
8  a1=[]
9  a2=[]
10 nevent=1
11 rec_length=1999    #1999ns 是取数时的采样长度。
12 count=0
13 eventnum=0
14 while (count<rec_length*nevent):
15     arr=s.split(' ')
16     a1.append(float(arr[0]))
17     a2.append(count)
18     s=f.readline()
19     count+=1
20 x=np.array(a1)
21 y=np.array(a2)
22 pl.figure(figsize=(8,8))
23 fig=pl.plot(y,x,"r.",label='the waveform',c=(0,0.0,0.0),alpha=0.5,
    linestyle='--')
24 pl.suptitle('example waveform')
25 pl.style.use('ggplot')
26 pl.grid(True)
27 pl.grid(color='g',linewidth=2,alpha=0.2,ls='--',lw=1)
28 pl.xlabel('time[ns]',color='k',fontsize=15,rotation=0)
29 pl.ylabel('voltage[mV]',color='k',fontsize=15,rotation=90)
30 pl.legend(loc="upper right")
31 pl.savefig("wave.eps")
32 pl.show()
```

A.2 python 计算平均波形

```

1  #!/usr/bin/env python
2  # -*-coding:utf-8 -*-
3  import numpy as np
4  from scipy import interpolate
5  import pylab as pl
6  f = open('/media/tao/_dde_data/arlierwork/simulation/tao/exp220/example.
    lvm')#换成你自己的文件名
7  s = f.readline()
8  a1=[]
9  nevent=580
10 rec_length=1999    #1999ns 是取数时的采样长度。
11 count=0
12 eventnum=0
13 time=0
14 avewave=np.zeros( rec_length )
15 while (count<rec_length*nevent):
16     arr=s.split(' ')
17     a1.append(float(arr[0]))
18     s=f.readline()
19     avewave[time]+=float(arr[0])/nevent
20     time+=1
21     if(count%rec_length==0):
22         eventum=eventnum+1
23         time=0
24         count+=1
25 x=np.array(a1)
26 y=np.linspace(1,rec_length,rec_length)
27 pl.figure(figsize=(8,8))
28 fig=pl.plot(y,avewave,"r.",label='the average waveform',c=(0,0.0,0.0),
    alpha=0.5,linestyle='-')
29 pl.suptitle('example average waveform')
30 pl.style.use('ggplot')
31 pl.grid(True)
32 pl.grid(color='g',linewidth=2,alpha=0.2,ls='--',lw=1)
33 pl.xlabel('time [ns]',color='k',fontsize=15,rotation=0)
34 pl.ylabel('voltage [mV]',color='k',fontsize=15,rotation=90)
35 pl.legend(loc="upper right")
36 pl.savefig("avewave.eps")
37 pl.show()

```

参考上面的程序，处理多个波形的电荷积分。

