# 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

# 第 1 章 General Introduction

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

**注意：**

1. *waveforms raw data @linght intensity about 0.1pe*
2. *waveforms raw data @linght intensity about 1pe*

## 1.1 读取波形文件

我们存储下来的 lvm 文件如图1.1每行只有一个数字，代表一个电压值（单位 mV)；如果你在 labview 中设置采样长度是 1000，那么每 1000 行就组成一个波形；每一个点的时间间隔是 1ns。

```
592.000000
592.000000
592.000000
592.000000
591.000000
592.000000
592.000000
592.000000
592.000000
593.000000
592.000000
```

**图 1.1:** lvm 文件

## 1.2 计算 PMT 的增益的方法

我们从 lvm 文件读取到很多波形的信息，根据这些波形的统计特征来计算对应的 PMT 增益。

数据采集 ADC 的阻抗是 50Ω，我们用波形的面积 ($\int UT$) 除以电阻可以得到每一个信号的电荷量大小。实验中我们看到的 PMT 信号来自于 $\mu$ 子在闪烁体中沉积能量产生的发光，我们简单假设每个 $\mu$ 子事件平均产生光子数为 5000 个，系统总的的探测效率是 10%，那么我们预期看到的 PMT 信号就是 $5000 \times 10\% = 500$ 个电子被放大之后的结果。PMT 对电子的放大倍数，也就是增益可以用公式1.1来计算：

**公式 1.1: 增益计算**

$$\delta = \frac{Q_{out}}{q} = \frac{\overline{\int UT}}{Rq}. \tag{1.1}$$

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。

**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 $\mu$ 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 $\mu$ 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 $\mu$ 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。

[1] e 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
[4] 信号时间差小于一定范围

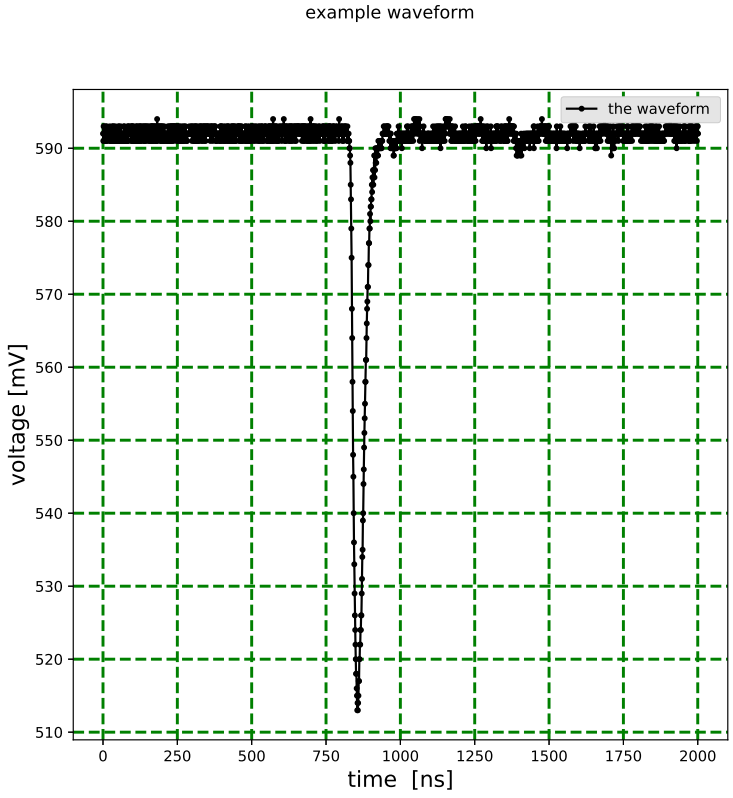


**图 1.2:** 一个 PMT 信号波形

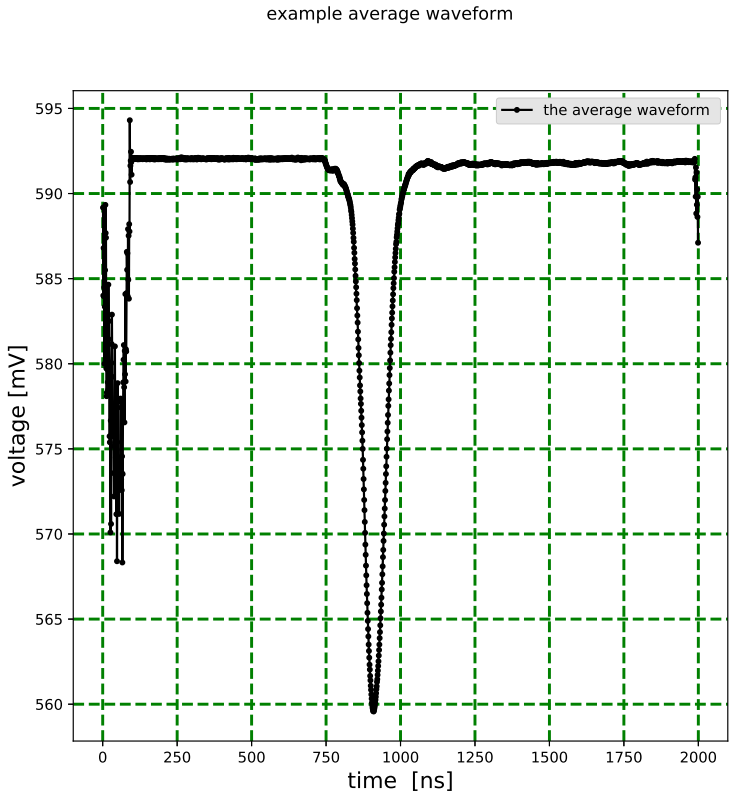


**图 1.3:** PMT 信号平均波形

# 附录 读取 example 文件的 python 程序

## A.1 python 读取一个波形

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

# A.2 python 计算平均波形

```
#!/usr/bin/env python
# -*-coding:utf-8 -*-
import numpy as np
from scipy import interpolate
import pylab as pl
f = open('/media/tao/_dde_data/arlierwork/simulation/tao/exp220/example.lvm')#换成你自己的文件名
s = f.readline()
a1=[]
nevent=580
rec_length=1999    #1999ns 是取数时的采样长度。
count=0
eventnum=0
time=0
avewave=np.zeros( rec_length )
while (count<rec_length*nevent):
      arr=s.split('␣')
      a1.append(float(arr[0]))
      s=f.readline()
      avewave[time]+=float(arr[0])/nevent
      time+=1
      if(count%rec_length==0):
          eventum=eventnum+1
          time=0
      count+=1
x=np.array(a1)
y=np.linspace(1,rec_length,rec_length)
pl.figure(figsize=(8,8))
fig=pl.plot(y,avewave,"r.",label='the␣average␣waveform␣',c=(0,0.0,0.0),alpha=0.5,linestyle='-')
pl.suptitle('example␣average␣waveform')
pl.style.use('ggplot')
pl.grid(True)
pl.grid(color='g',linewidth=2,alpha=0.2,ls='--',lw=1)
pl.xlabel('time␣␣[ns]',color='k',fontsize=15,rotation=0)
pl.ylabel('voltage␣[mV]',color='k',fontsize=15,rotation=90)
pl.legend(loc="upper␣right")
pl.savefig("avewave.eps")
pl.show()
```

参考上面的程序，处理多个波形的电荷积分。