



浙江大学

本科实验报告

课题题目： μ 子测量实验

姓名学号： 陈书扬 3210106335

陈科 3210104844

指导老师： 陈星

2022 年 12 月 31 日

	目录
1 背景知识	4
1.1 背景相关	4
1.2 课题概述	4
2 原理	4
2.1 基本原理	4
2.2 实验原理	4
2.3 公式推导	5
3 仪器系统	5
3.1 实验器件	5
3.2 实验仪器	5
3.3 实验软件架构	7
3.3.1 示波器的软件架构	7
3.3.2 主要算法的软件架构	8
3.3.3 Sensor CASSY-2 的软件架构	8
3.3.4 图形界面相关软件结构	8
4 实验	9
4.1 实验方法	9
4.1.1 先前方法存在的不足	9
4.1.2 低采样率方法的主要思想	10
4.1.3 采样率的选择	10
4.1.4 峰值修正策略	11
4.1.5 找不到下降沿上的点时的处理	14
4.1.6 寻峰算法	15
4.2 实验步骤	16
4.2.1 采集步骤	16
4.2.2 CASSY SDK 使用步骤	18
4.3 实验数据	20
4.4 实验结果	20
4.4.1 模拟结果	20
4.4.2 实际采样结果	22
4.4.3 平顶修复结果	23
5 分析与讨论	24
5.1 误差来源	24
5.2 误差分析	24
5.2.1 寻峰误差	24
5.2.2 峰值修正误差	25
5.2.3 最终测量结果误差分析	25
5.3 实验分析	25

5.3.1	关于 Sensor-CASSY 2 的可用性	25
6	实验总结	26
7	课题总结	26
7.1	仪器系统照片	26
7.2	课题所用器材清单	27
7.3	花絮和留言	27

1 背景知识

1.1 背景相关

μ 子是一种轻子, 由宇宙中的 π 介子衰变产生, 是宇宙射线到达地面时的一种主要成分。 μ 子会很快衰变成电子、反电子中微子和 μ 子中微子, 寿命很短, 仅有 $2.2\mu\text{s}$ 。由于相对论效应, μ 子的寿命会被延长, 从而有几率到达地球表面, 被捕捉到。在应用上, μ 子质量较大, 不像电子一样只能在直线加速器中加速, 能够在圆环加速器中加速, 在对撞实验中具有重要的意义; μ 子可以进行透射和散射成像, 考古上曾经通过 μ 子成像探测胡夫金字塔的内部。此外, μ 子也可以被用于催化核聚变反应, 对于低温核聚变具有一定的研究价值。[1]

1.2 课题概述

本课题分为实际采集和数值模拟两个部分, 实际采集为直接在实验室中捕获 μ 子并将 μ 子捕获事件转换为数值信号, 通过数据分析测算 μ 子的寿命和能量分布等; 数值模拟为通过随机方法模拟 μ 子进入采集区域的频率, 并对捕获事件进行处理, 从而得到 μ 子寿命和能量分布的模拟结果。本报告将主要描述其中的实际采集部分。

2 原理

2.1 基本原理

使用一块塑料晶体采集 μ 子(具体原理见下), 塑料晶体的一侧置光电倍增管, 光电倍增管接示波器。若有 μ 子进入塑料晶体, 无论是残留在内还是穿透, 都会产生一个脉冲信号; 如果残留在内并发生衰变, 则又会产生一个脉冲信号(具体原理见下)。这样, 如果发现一对发生时间相近的脉冲, 其相距时间在可接受的 μ 子寿命范围之内(实验中取 $1\sim10\mu\text{s}$), 那么便可以判断这一对脉冲属于同一个 μ 子。示波器收集到脉冲信号, 并传输给电脑; 电脑上检索并匹配脉冲信号, 如果判断为 μ 子, 那么计算两者的时间差作为 μ 子寿命, 通过脉冲峰的高度判断出能量值的相对大小。如此, 即可得到 μ 子的寿命分布与能量分布。

2.2 实验原理

实验装置之间的连接关系如图示。塑料晶体的尺寸为 $40 \times 20 \times 10(\text{cm}^3)$, 内部为闪烁体材料。高能粒子(如 μ 子)进入闪烁体时, 动能发生衰减, 减少的动能传递给闪烁体, 使得闪烁体中的电子受到激发。之后电子退激, 能量以光的形式散发出来, 然后被光电倍增管捕捉。

光电倍增管是借用光电效应将微弱光信号转化成可观测电信号的仪器。电子退激所发的光照射在光电倍增管的光阴极上, 产生光电效应, 发射出光电子。在光阴极和光阳极之间有多个打拿极, 电子通过打拿极时不断地被倍增, 最后达到可观测的程度。



图 1: 实验装置连接总图

之后, 这些电信号被传输到示波器上, 示波器再将数据转移到电脑上, 从而获得能够进行处理的时序信号。在这之后存在的问题为:

- (1) 如何从时序信号中找到峰?
- (2) 如何从峰中获得较为准确的 μ 子衰变时间和能量分布?

这部分为我们所做的主要工作。由于篇幅较长, 该部分在“实验方法”中作描述 [4.1], 在此不作赘述。

2.3 公式推导

本次实验没有长篇的公式推导, 故不在此处作描述。少数使用到的数学公式会在使用处详细解释。

3 仪器系统

3.1 实验器件

- (1) 塑料晶体。在 [2.2] 中已经叙述。
- (2) μ 子。 μ 子在实验室中自然采集, 理论上的通量为 $1 \sim 2 \text{cm}^{-2} \cdot \text{min}^{-1}$ [3].

3.2 实验仪器

- (1) 光电倍增管。
- (2) 示波器。示波器的型号为 Tektronix TDS 1002C-EDU. 该示波器为双通道示波器, 存储深度为 2500, 极限采样率为 $1.0 \text{GS}/\text{s}$. 后续会对该示波器的性能进行进一步分析。
- (3) 电脑。系统为 Windows 8.1, 主频为 2.9GHz。由于这一部分实验对电脑本身性能要求不高, 剩余参数不再给出。

(4) Sensor-CASSY 2 及其配套的 MCA Box。Sensor-CASSY 是 Leybold 公司生产的用作教学目的的传感器, Sensor-CASSY 2 是 Sensor-CASSY 的第二代。它能够像示波器一样采集电信号。Sensor-CASSY 2 本身不像示波器一样具有处理数据的能力, 它只能将数据传输回电脑。但另一方面, Sensor-CASSY 2 传输数据的能力比示波器要强得多, 这使得它在该实验上比示波器更适合用于采集数据。其结构如下图所示。

虽然 Sensor-CASSY 2 在该实验中之前的主要用途是搭配 MCA Box 进行多道分析, 但实际上选择合适的连线之后, Sensor-CASSY 2 可以直接从光电信增管中接线到 INPUT B 模块的两极, 从而直接取代示波器的采集功能; 并且 Sensor-CASSY 2 没有显式的单次采样时间限制, 这也是认为其数据采集能力要强于示波器的原因。



图 2: Sensor-CASSY 2 的结构

Sensor-CASSY 2 本身无数据处理功能。但从上图中可以看到, Sensor-CASSY 2 在面板上具有三个模块, 其中 INPUT A 和 INPUT B 上都带有一个串口。Leybold 另外提供 Sensor-CASSY 2 的一些可选配件, 这些配件上也带有串口, 可以接在 Sensor-CASSY 2 的串口上。输入信号先通过附加件, 经过处理后再传输给 Sensor-CASSY 2(如下图所示)。这样, Sensor-CASSY 2 就具有了一定的数据处理能力。

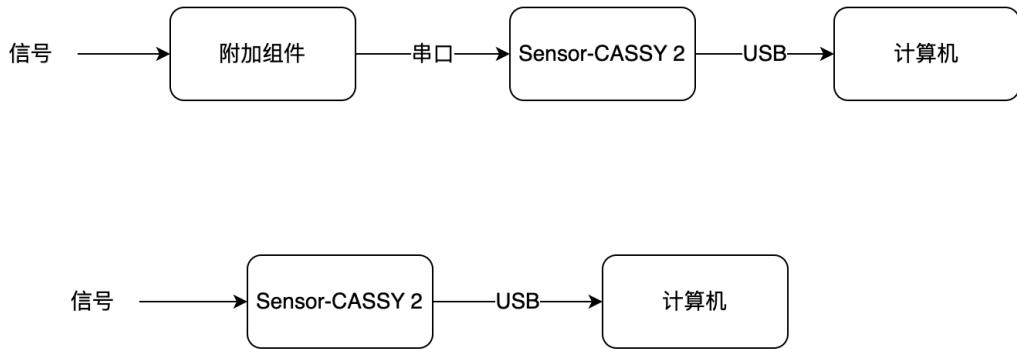


图 3: Sensor-CASSY 2 的两种运行模式

MCA Box(Multiple Channel Analyzer Box, 多道分析仪) 就是其中的一种配件, 结构如下图。信号进入 MCA Box 时, 其中的脉冲信号会被统计, MCA Box 通过脉冲的幅度放入相应的道, 最后形成多道分析的能量谱。



图 4: MCA Box

虽然 Sensor-CASSY 2 采集能力优秀, 但本实验中最后的解决方案并没有用到这两个仪器, 在后续部分中将会陈述原因。[\[5.3.1\]](#) 在实验过程中, 我们完成了对多道分析仪的编程操作。我们认为该仪器还具有相当的探索空间, 因此也将其使用方法记录在该报告中。

Sensor CASSY-2 与 MCA Box 的官方文档见于[此处](#) 和[此处](#)。

3.3 实验软件架构

3.3.1 示波器的软件架构

VISA(Virtual Instrument Software Architecture, 虚拟仪器软件架构), 是通用的仪器编程 I/O 标准与规范的总称, 由 VXI plug&play 制定。本实验中使用的 Tek 示波器支持使用 VISA 标准进行编程。VISA 标准有许多实现, Tektronix 自身也提供一套 VISA 的实现 TekVISA。本实验中使用的是一个更常用的实现 NI-VISA, 该实现由 NI(National Instrument, 美国国家仪器有限公司) 完成。下图为 VISA 标准实现与仪器通信的框架。

NI-VISA 的网站见[NI-VISA 官网](#)。

Instrument Communications Block Diagram

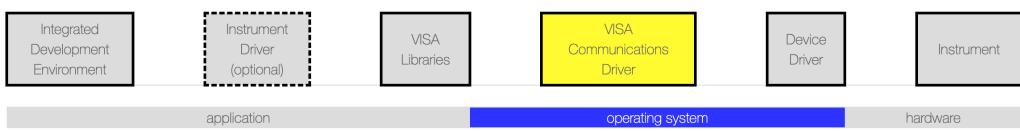


图 5: VISA 框架

NI-VISA 驱动附带 C/C++、VB 的库, 因此可以直接使用 C/C++ 进行编程。在 20 级同学所完成的实验中, 采用的即是 C++ 的 VISA 接口。经过阅读, 我们认为 C++ 的 VISA 接口容易造成较差的可读性, 不利于后续继续开发, 同时观察到该实验对计算性能的要求并不高, 因此采用了 PyVISA。PyVISA 是一个 VISA

的前端，提供较为简单的 Python 接口，能够较大程度上简化代码组成。PyVISA 仅仅是一个前端，需要有一个 VISA 后端（如 NI-VISA）才能运行。

实验室的电脑上，已经安装有 NI-VISA 和 PyVISA。

3.3.2 主要算法的软件架构

由于通信接口上选择了 PyVISA 这样基于 Python 的前端，我们的算法也全部使用 Python 实现。主要涉及 Python 的科学计算相关包环境，如 Numpy、Scipy 等。这类包较为常用，且不是实验重点，在此不作赘述。

我们编写的程序包大致结构如下：

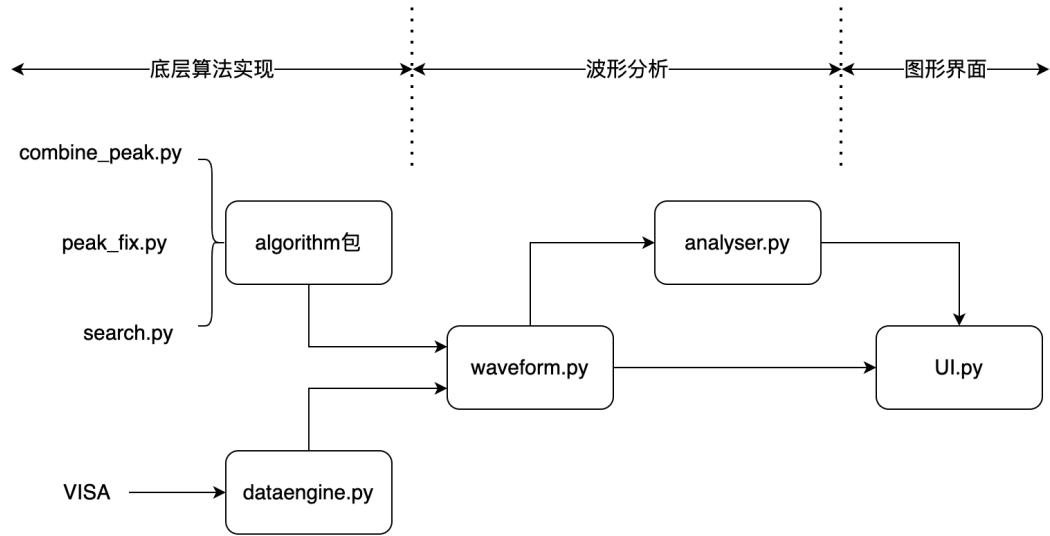


图 6: 软件包结构

其中，algorithm 包中主要包括了峰的合并（即将主峰和次峰匹配起来）、峰值修正（peak_fix）和寻峰算法（search）。dataengine 是对 PyVISA 的再封装。waveform 是建立在 algorithm 基础上的一个波形类，统一调用 algorithm 中的模块，提供更高层次的波形分析接口。analyser 是同一层次上提供多道分析功能的接口。最终全部由 UI 调用，形成图形界面。

3.3.3 Sensor CASSY-2 的软件架构

Sensor CASSY-2 官方提供了两种可供读写的方式。一种是通过官方的 CASSY LAB 2 软件，在 CASSY LAB 上可以比较直观地操作；另一种是通过编程的方式。Leybold 官方提供开发者工具包 (SDK) (见于[此处](#)列表中的 CASSY SDK)。CASSY SDK 基于.NET Framework 框架编写，编程语言为 C#。其中的 Api/LD.Api.dll 类库提供了对 CASSY 所有相关仪器的接口。由于涉及.NET 开发，该部分在进行编程时需要使用 Visual Studio。由于使用了.NET Framework 中有关串口通信的模块，CASSY SDK 对.NET Framework 的版本也有一定的要求；实测.NET Framework 4.8 可以在实验室的电脑上满足要求。

3.3.4 图形界面相关软件结构

我们使用的 GUI 框架是 ttkbootstrap，是一个基于 Python 原生图形库 tkinter 的一个美化库。以往使用的 Qt 图形库比较重量级，开发时往往牵连到 Visual Studio 等重量级工具，并且附加的库繁多，启动方式较为复杂，界面的编写方式也比较复杂。我们使用 ttkbootstrap 的初衷就是使该实验的图形界面运行更简单，同时不失去界面的美观性。ttkbootstrap 由于是基于 Python 原生库开发而成，其安装非常简单，只要需要使用 pip

命令即可。另外，我们编写的 GUI 代码只有一个入口主程序 UI.py，运行只需要 python UI.py 即可，非常简单。不过，我们虽然实现了 GUI 代码和后端实现代码的分离，但 GUI 代码的可读性仍然没有做得很好。

4 实验

4.1 实验方法

4.1.1 先前方法存在的不足

实验室目前采用的示波器型号为 Tektronix TDS 1002C-EDU. 从 Tek 示波器的官方网站上，可以查询到该种示波器的最大存储深度为 2500. 换言之，无论采用何种采样频率，示波器在采样 2500 个点的信息后存储容量就已经达到上限。如果没有连接电脑，继续采集的数据就会使得之前采集的数据被不断抛弃；如果连接了电脑，就必须在采集完 2500 个点之后，先向电脑传输这 2500 个点的数据，传输完毕之后再进行下一次 2500 个点的信息采集。

重点在于，Tek 示波器不具有同时传输数据和采集的能力。所以，整个数据采集的过程体现为如下的循环：

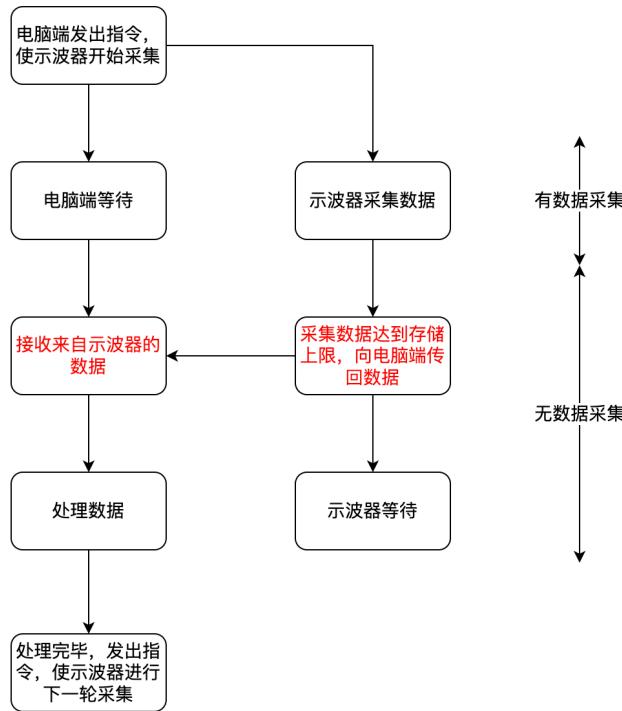


图 7: 编程模型

在先前的同学采用的方法中，优先保证了足够高的采样率，具体为 2500 个点在 $10\mu\text{s}$ 内采集完成，相邻采样点的时间间隔为 4ns 。这样，在一次完整的采样循环中，示波器的有效采样时间为 $10\mu\text{s}$ 。在数据传输和等待计算机处理完成数据的过程中，示波器处于停机状态，没有进行采样。其中，后者由于单次传输数据量不大，可以在 ms 级别完成全部操作；但前者，即示波器向计算机传输数据的时间，有整整 2s 之多。这就意味着，一个完整的采样周期大约为 2s ，其中仅有 $10\mu\text{s}$ 真正用于采集数据。在剩下的时间中，示波器无法进行采集，假如此时有 μ 子通过，是完全无法采集的。而根据有效采样时间和完整采样周期的时间比例，很容易知道：当前的实验装置事实上只收集到了很少的一部分 μ 子。

因此可以判断，在采集实验部分最大的瓶颈并非后续的数据处理操作，而是传输速率。关于这一点，先前

的同学也都有提到 [2]。不可否认，更换硬件实际上是解决该问题的最好办法，因为有大量的仪器可以连续地将数据流传输到电脑端。事实上，实验室已经配备的 SENSORS-CASSY 2 就具有连续的波形读写能力，但是由于采样率稍低而导致不可用 [5.3.1]。之前的同学也尝试过使用 FPGA 代替示波器进行传输，但由于对 FPGA 不熟悉，我们并没有采用这样的做法。我们希望能够基于当前的硬件条件，提高有效采样时间的比例。

4.1.2 低采样率方法的主要思想

基于先前方法的以上缺陷，我们提出了低采样率的方法。

先前的方法中，由于单次采样时间长度只有 $10\mu\text{s}$ ，故一次采样最多只能找到一个 μ 子。同时，我们观察到，在先前的采样率下，在单个峰上就会采样非常多的点，但事实上确定一个峰只需要能采集到峰顶的点就可以了。综合判断，我们认为先前的采样率是过高的。通过降低采样率，在固定的存储深度下，就能相应地延长单次采样时间。在采样时间足够长的时候，虽然不能做到连续采集，但是就会有一次采样捕捉到多个 μ 子的可能，从而提高采集效率。

但是，采样率是不能无限制降低的，采样率需要保证能够准确地找到峰位置。在示波器的数据中， μ 子表示为一个尖锐的脉冲，如下图。由于峰上升和下降的斜率非常大，因此采样率选择稍有不恰就会导致采集到的脉冲高度低于真实值，甚至把整个脉冲漏掉。

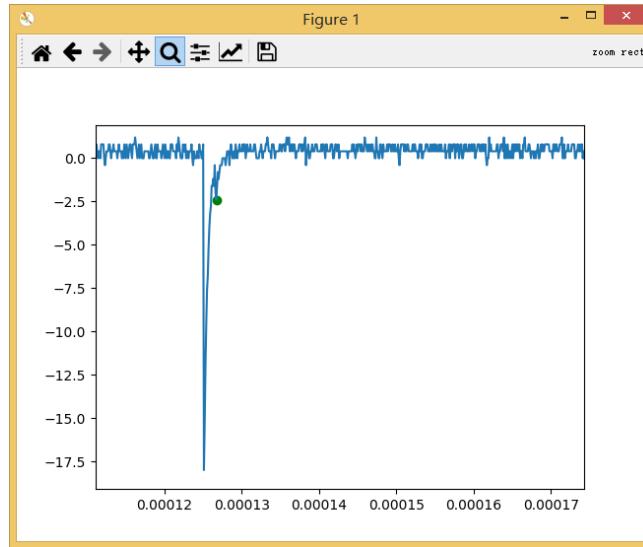
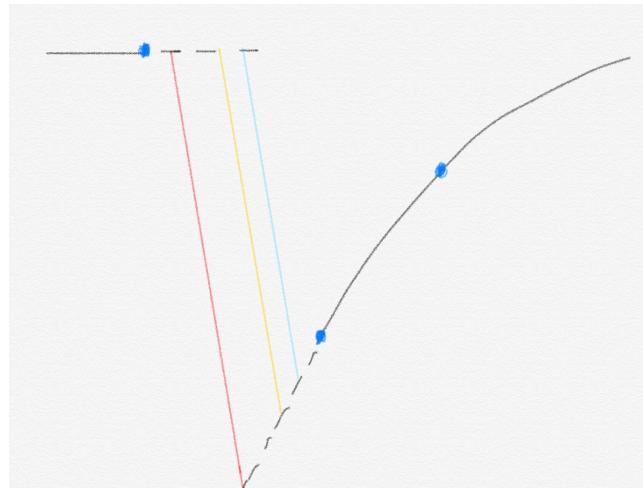


图 8: 脉冲图示

因此，我们只能在一定范围内选择采样率。但即使在相对合理的采样率区间内，较低的采样率也会对峰值的测量造成一定的误差。为此，我们又基于脉冲的规律设计了峰值修正的方法，事实证明是很有效的。

4.1.3 采样率的选择

示波器的采样率只能够逐档选择，不能够连续变化。在采样率降低到原来的 $\frac{1}{50}$ 即采样点稀疏 50 倍时，我们得到的散点大致呈现如下形状。肉眼可以判断这里存在一个峰，但无法判断峰位置。这里的最低点只是右侧上升沿的一个点，由于采样率已经很低，不能直接作为峰顶使用，否则会产生约 30% 的误差，这是不可接受的。并且，从理论上考虑就可知道无法根据这张图的信息还原出峰的位置，因为如图的蓝线、黄线、红线都可能是下降沿的位置，单纯依靠图中信息，无法确定下降沿在何时出现。因此该采样率不可用。

图 9: 先前采样率 $\frac{1}{50}$ 下的情形

上述采样率不可用，主要在于无法知道下降沿的位置。下降沿的时间非常短，在该采样率下几乎不可能在下降沿上采到点。但是，如果将该采样率再扩大一倍，即提升至先前采样率的 $\frac{1}{25}$ ，就有较多的样本能够在下降沿上取到一个点。利用这一特征，我们可以对峰位置进行较为有效的修正。因此，最终我们的采样率定为每 $0.1\mu\text{s}$ 取一个点，采样时间是先前的 25 倍。

4.1.4 峰值修正策略

理论前提

Law 1 下降沿的总用时是一个定值。

事实上，只要在高采样率下找到开始下降的位置以及峰顶的位置，两者相减，并多次实验，即可验证这条规律。在探索这条规律时，我们使用的方法相对比较繁琐：通过拟合计算下降沿的斜率绝对值 k ，并与峰高 h 作 $k-h$ 散点图。如果下降沿用时 d 是一个定值，那么有 $h = dk$ ，即 $k-h$ 散点图应当表现出近似正比的关系。事实上，也确实是这样。在近 1000 份数据上的表现如下图所示。

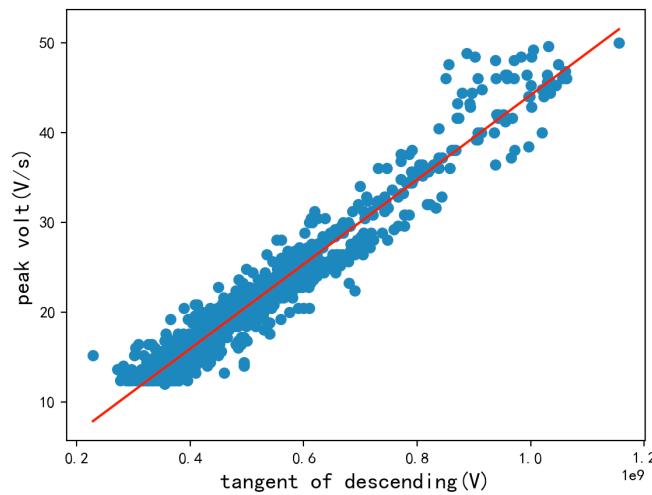


图 10: k-h 图

对 $k-h$ 关系作线性拟合, 得到的斜率即为所求的 d 。我们得到的结果是: 下降沿所用的时间为 $4.71 \times 10^{-8}s$, 拟合直线的截距为 $-2.93V$, 是一个相对较小的值, 可以认为几乎通过原点。即 h 和 k 存在正比关系。

在这张图中, 容易发现散点的分布具有一定的宽度。我们认为这是因为没有作任何滤波操作导致的随机波动。

Law 2 上升沿总是符合负指数上升关系 $V = e^{-at+b}$, 且 a 是一个定值。

类似地, 我们也可以通过在高采样率下取上升沿的点拟合, 并且比较大量样本的 a 值得到结论。下图中, 横坐标为某个样本的峰高度, 纵轴为对应样本的 a 值。可以看到, a 值是基本不随着峰的高度变化而变化的, 呈现出一条水平线。

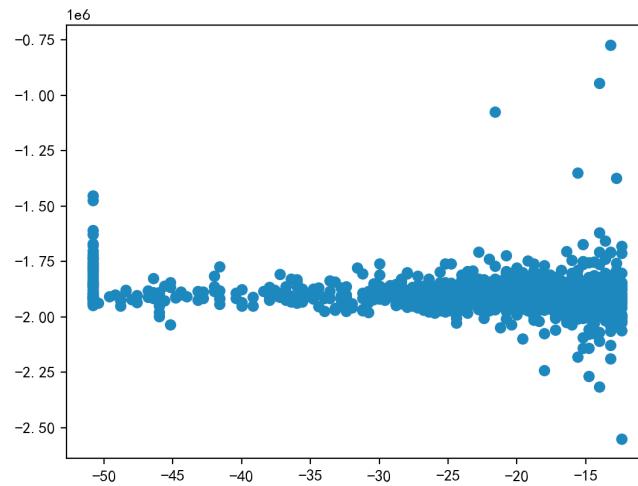


图 11: h-a 图

在图像的最左侧可以看到一条积聚的直线, 似乎违背了上述规律。实际上, 这是由于高电平极限造成的平顶带来的直线; 峰高度在一定高度以上的数据会在示波器处被截断, 从而形成一部分看似峰高度相同的无效数据。将这部分数据去除并对 a 值平均, 得到平均的 a 值, 为 $1.91 \times 10^6 s^{-1}$.

判断是否存在下降沿上的信号点 以下操作都是建立在已经完成寻峰的基础上进行的, 即已经知道了当前峰的最低采样点位置。关于如何寻找最低采样点, 请阅读寻峰算法相关部分。[\[4.1.6\]](#)

先进行一步简单的筛选。如果从最低点出发, 向左的两个采样点都发生了较大的爬升(如图), 那么最低点左侧的点必然是在下降沿上的点。图中灰色线代表理想信号线, 蓝点代表采样点, 蓝线代表直接连接采样点形成的信号线。判断“较大爬升”取为上升幅度超过噪声阈值。

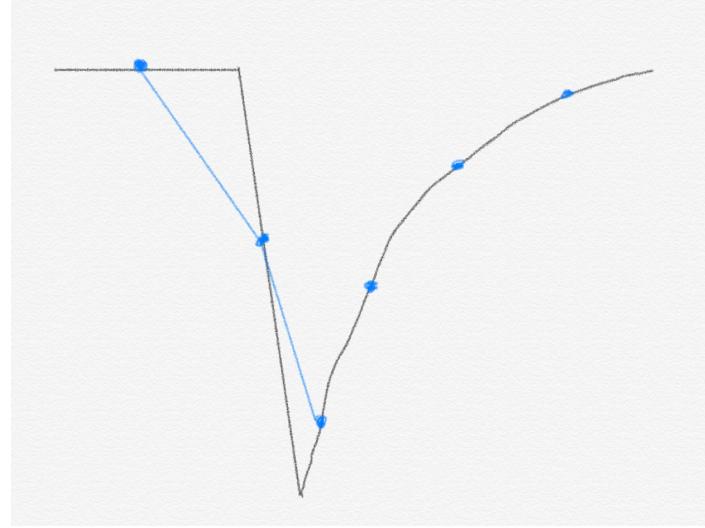


图 12: 较简单的判断情况

但是, 也有可能出现以下两种情况, 其中后者指的是前两个蓝色点之间的差距太小, 以至于低于噪声阈值。后者的情况其实该点已经几乎不在下降沿上了, 我们认为已经在水平上, 放到后续处理。前一种情况下, 实际上下降沿上的点已经成为采样最低点。此时作出上升沿的拟合曲线, 将最低点的 t 值代入, 得到一个预测点。假如该点不在下降沿, 那么必然在上升沿, 上升沿上的点几乎都在拟合曲线上, 那么真实点和预测点应该几乎重合; 假如该点在下降沿, 那么由于下降沿的斜率非常大, t 值稍有偏离就会导致快速偏离上升沿的拟合直线, 从而真实点和预测点的距离应该很远。根据这两种判据, 我们可以判断是否有点在下降沿上。

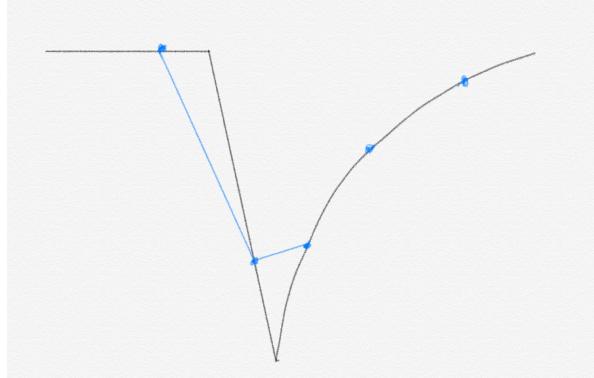


图 13: 采样最低点在下降沿上

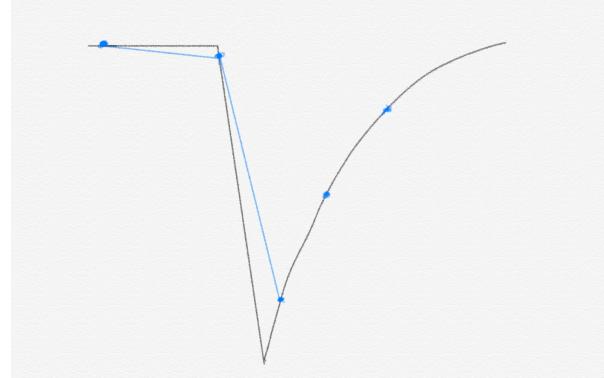


图 14: 左侧上升幅度过小

根据下降沿上的点修正峰位置 假如在上一步中找了下降沿上的点, 那么可以根据下述方法预测出峰位置。

假设真实峰位置为 (t_0, v_0) , 前一步得到的下降沿上的点为 (t_1, y_1) 。首先, 峰应当在上升沿的拟合曲线上, 得到方程

$$v_0 = e^{-at_0+b}. \quad (1)$$

其次, 根据两点构造直线, 交 t 轴于一点 $(t_2, 0)$, 该点即为开始下降的位置。三点共线, 容易得出

$$\frac{v_0}{t_0 - t_2} = \frac{v_1}{t_1 - t_2}. \quad (2)$$

又有固定的约束条件

$$t_0 - t_2 = d. \quad (3)$$

联立以上三式, 解得

$$\frac{e^{-at_0+b}}{d} = \frac{v_1}{t_1 - t_0 + d}. \quad (4)$$

此时可以解出 t_0 。值得一提的是, 进行数值解的时候最好变换为乘积形式, 否则有数值发散的风险。

如上即可解出峰位置。理论上, 得到的解应该是精确解, 实践上该方法的表现精度也非常高。但是, 实际上有些波形不符合在上述讨论中采用的脉冲模型, 特别有时脉冲的尖端可能并不“尖”, 而是形成一个较为平滑的回拐。这时该方法会发生相对大的误差(超过 5%)。这种情况较少见, 目前作为测量错误处理。

4.1.5 找不到下降沿上的点时的处理

如果下降沿上没有找到采样点, 那么就只能进行估计, 但由于此时已经确定下降沿上没有采样点, 估计的精度已经相对较高。如图, 因为没有找到下降沿上的点, 那么整个下降沿必然被包括在两个采样点之间。这两个采样点中的左边点可以视为下降沿的起始点(橙色点)的合理估计。

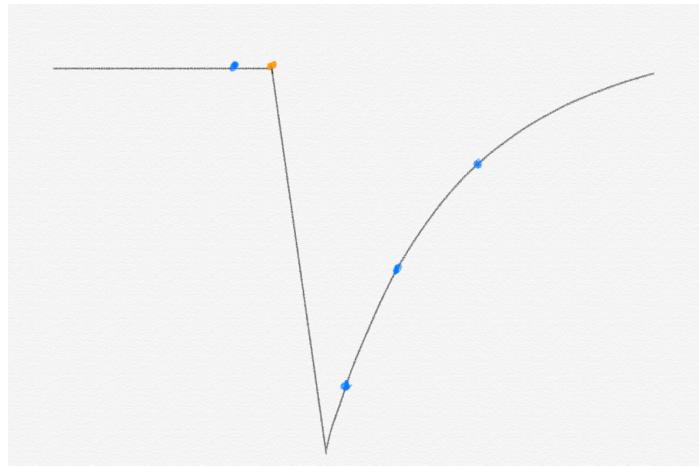


图 15: 近似估计

记相邻两个采样点之间的时间间隔为 Δt , 那么该点距离真实的下降沿起始点的间距最多为 $\Delta t - d$, 最少为 0, 平均为 $e = \frac{1}{2}(\Delta t - d)$. 此时, $\Delta t = 1 \times 10^{-7}s$, $d = 0.471 \times 10^{-7}s$, $e = 0.265 \times 10^{-7}s$ 已经是一个可以接受的误差。此时, 从该点出发向前移动 d 的时间, 就是估计峰位置的 t 坐标。由于峰必然在拟合曲线上, 将 t 值代入上升沿拟合曲线的方程, 即可得到峰高度的预测值。

此处还有一个小修正。由于取左侧的点总是比实际的开始下降点要来得提前, 故直接使用该点作为下降点的估计会使得每次预测的峰值都偏大。由于该点到真实下降点的位置可以认为是随机的, 因此人为地将该点附加上值等于平均误差值一半的修正, 即

$$t' = t + \epsilon = t + \frac{1}{2}e. \quad (5)$$

其中 t' 指的是找到的左侧点的 t 坐标。用 $(t', 0)$ 作为下降点的估计, 使得结果偏大和偏小的概率基本相等; 同时, 也使得最大误差值减为原来的一半, 平均误差为 0.(以上结论建立在 t 服从均匀分布的前提下)

综上, 整个峰值修正的流程如下:

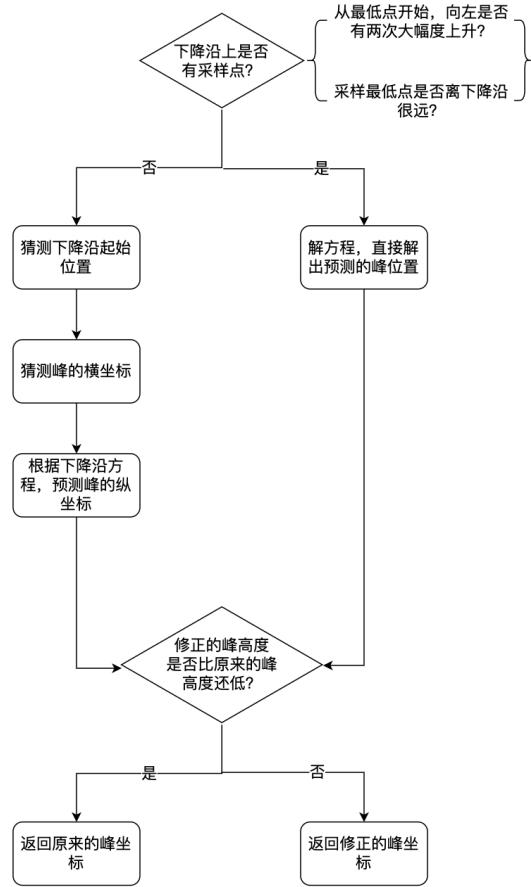


图 16: 峰值修正流程

4.1.6 寻峰算法

在以往的程序中，寻峰的任务往往是在数据中定位唯一的一个峰的位置，因为高的采样率下只允许采到一个峰。于是，以往通常将全局最低点直接当作主峰，将向后寻找次峰作为寻峰算法的重点。

但是，我们降低采样率的目的就是让一次采样中出现多个峰。这个场景下，直接用最低点作为主峰的算法过于朴素，不再适用。对此，我们用凸包算法外加维护区间最小值的方法代替寻峰，使得算法能够尽量找出一次信号中所有包含的峰。

凸包算法大致上是使用一个最小凸多边形来包住波形，凸多边形的顶点处就是所需要寻找的峰位置。由于原始的凸包算法并不能很好地完成寻峰任务，我们对寻峰算法作了很多附加的、基于经验的修正。另外，为了防止漏掉峰，我们在凸包算法之外又外加了一层区间最小值维护。其大致的思想是，要检查两个峰之间是否有未检出的峰，我们构造从前一个峰的峰顶到后一个峰的峰顶的区间，并控制区间不断向内收缩。在考虑了噪音的情况下，如果两峰之间没有其他的峰，那么区间的最小值应该始终在区间边界上；反之，则总有一个时刻区间的最小值会变化到区间内部。这个方法主要用于搜索凸包容易漏掉的次峰。

关于这两种算法的详细运用与正确性检验等，见陈科同学的论文。

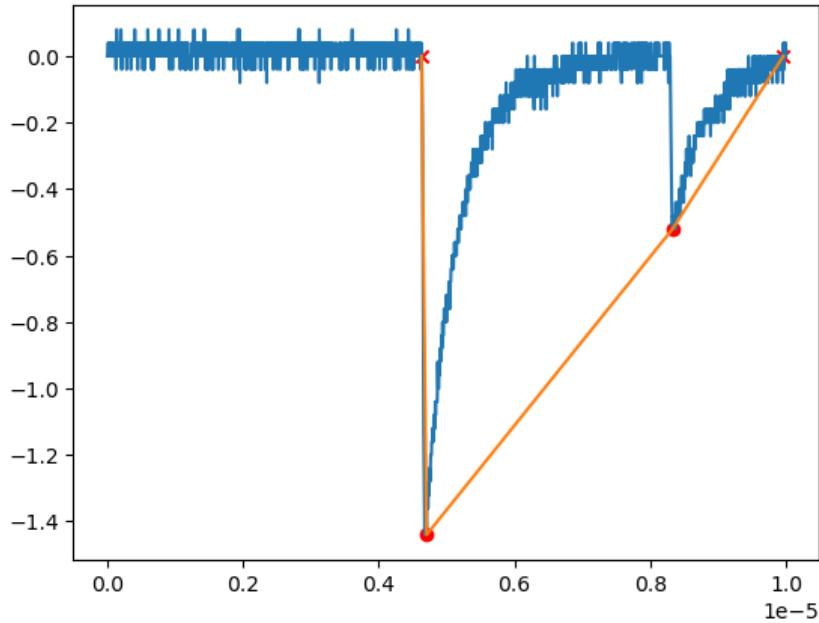


图 17: 凸包

4.2 实验步骤

此部分主要说明我们所编写的采集程序的使用方法。另外, CASSY SDK[3.3.3] 的使用方法虽然没有运用到最终的程序中去, 但使用方法记载在这里, 以供参考。

4.2.1 采集步骤

打开我们编写的采集软件界面, 如下。

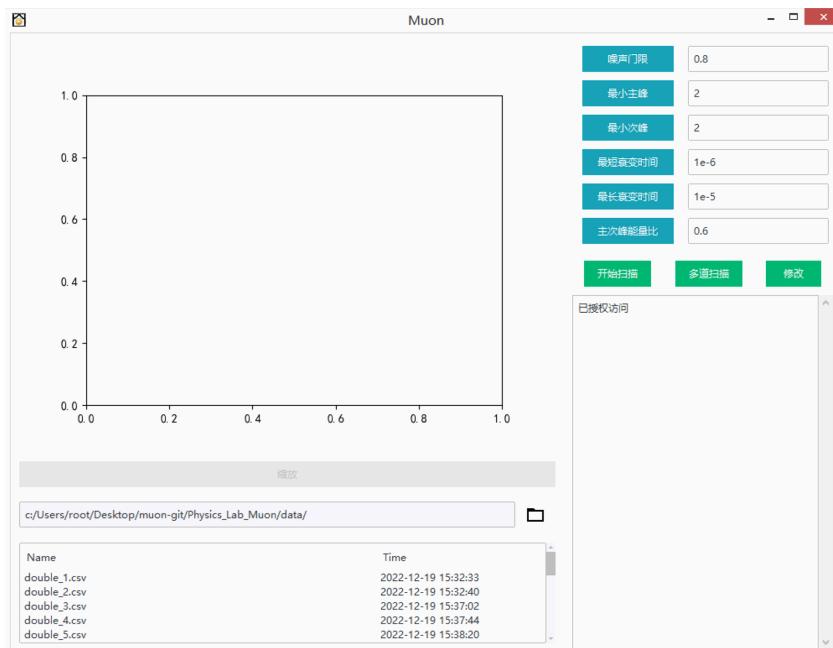


图 18: 采集软件界面

可以通过下方的窗口滑动查看数据, 双击文件名, 可以在上方图形窗口中显示图像。图像中, 散点标识的是识别出峰的位置, 峰的位置已经经过修正。红色表示的成对点是经过匹配得到的双峰, 橙色表示没有匹配的单峰。

由于在低采样率下峰很窄, 未必能够看清, 我们实现了局部放大功能。点击“缩放”, 图片进入放大模式, 此时每次点击“缩放”, 电脑会依次放大显示图中每一个峰周围的详细图像。在放大显示模式下, 我们会画出相关的辅助线。辅助线太多会不利于观察, 我们只保留上升沿的拟合曲线, 用红色虚线表示。

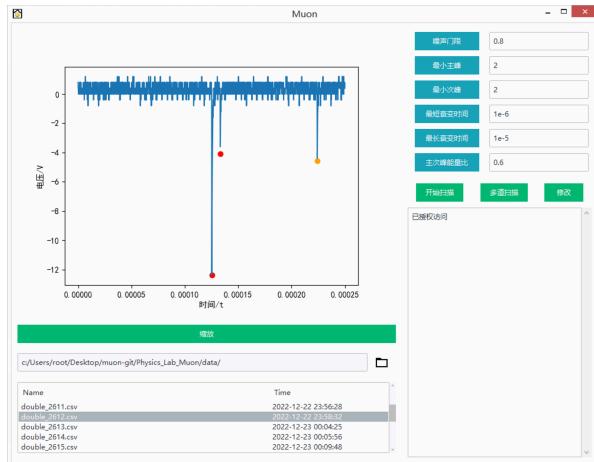


图 19: 浏览图像



图 20: 放大显示

在选定的数据文件夹内, 点击右侧的“多道扫描”, 可以进行多道分析。多道分析的道数目前只支持在程序内修改, 且运行速度较慢, 约 6000 份的数据需要运行 2~3 分钟。

多道分析运行结束后, 绘图区会显示电子能量分布和 μ 子能量分布的结果。初始状态默认两个分布绘制在一张图上, 点击下方的“切换”可以依次显示单个分布。同时, 右侧的日志框中会显示所探测到 μ 子的数量和分析得到的平均衰变时间。

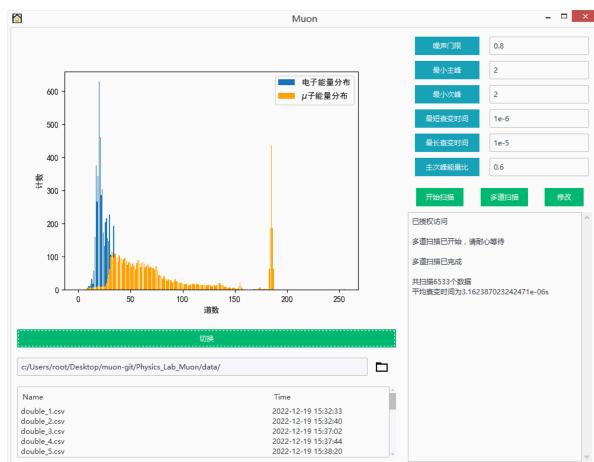


图 21: 多道分析结果

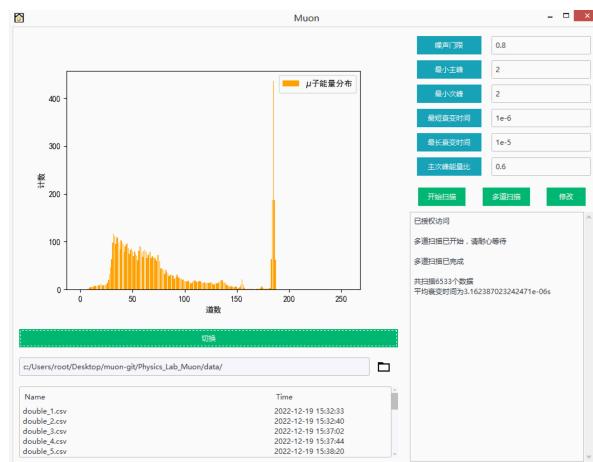


图 22: 单幅展示分析结果

在使用采集功能时, 右侧提供了六个允许修改的数据处理参数。

噪声门限 在指定值以下的电压会被视为噪音。

最小主峰 若峰的高度小于指定值，则该峰不会被识别为主峰。

最小次峰 若峰的高度小于指定值，则该峰不会被识别为次峰。

最短衰变时间 若两个峰之间的距离小于指定值，则不认为这两个峰可能是衰变的关系，因为太近了。

最长衰变时间 若两个峰之间的距离大于指定值，则不认为这两个峰可能是衰变的关系，因为太远了。

主次峰能量比 如果后峰高度与前峰高度之比大于指定值，则不认为这两个峰可能是衰变的关系，因为次峰不可能和主峰差不多高。

参数需要在采集开始之前预先设定，建议使用默认值。

在选定好存储数据的目录后，点击“开始采集”。该程序只存储有衰变信号的数据，默认展示最新采集到的数据。右侧的日志框会显示当前的记录情况。



图 23: 采集中的情形

再点击上方的“停止扫描”，即可停止采集。此时点击“多道分析”，可以进行数据分析。

4.2.2 CASSY SDK 使用步骤

(1) 下载 CASSY SDK([下载地址](#)).CASSY SDK 并不是一个需要安装的软件包，只是一个内含接口库和示例项目的一个压缩包。

(2) 打开 Visual Studio 2019，新建项目。选择模版时，搜索找到：控制台应用 (.NET Framework)。



图 24: 创建应用

.NET Framework 在实验室电脑上已经安装好。如果其他情况，可以在 Microsoft 官网上下载.NET Framework 4.8.

(3) 创建成功之后，在窗口右侧的“解决方案资源管理器”中，右击“引用”，点击“添加引用”。选择“浏览”，在本地文件中找到下载的 CASSY SDK 中的 Api/LD.Api.dll，并添加到当前项目。

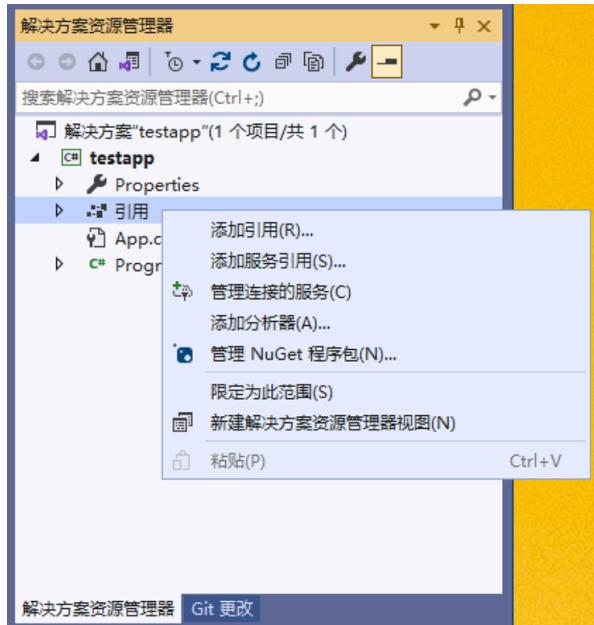


图 25: 解决方案资源管理器

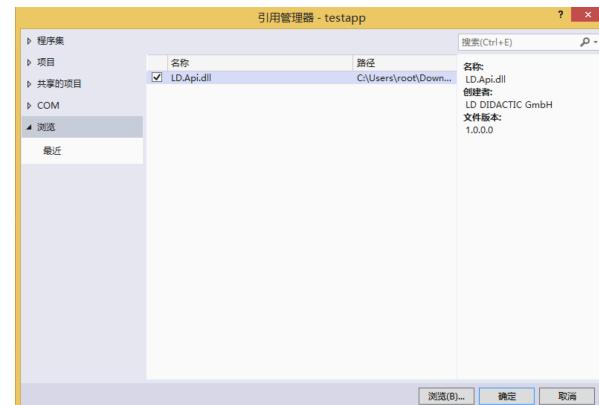


图 26: 添加引用

(4) 此时已经具备运行条件。若要尝试运行，CASSY SDK 的示例程序中包括了 MCA Box 的控制程序，可以将该示例项目的 Program.cs 代码拷贝到当前项目的 Program.cs 中。点击运行，可以在控制台中观察数据运行情况。

如果出现一些错误, 多半是因为类库引用不全。如果是这种原因, 可以像上述第 3 步类似地解决。

4.3 实验数据

该实验为了能顺利进行多道分析, 最终需要采集到的数据量比较大。在经过我们尝试之后, 发现要进行 128 道及以上的数据采集往往需要进行 3 天及以上的采集时间。因此每次采集都会得到较为庞大的数据。示波器每次捕捉的数据都保存为一个 csv 格式的文件, 文件的内容大致为:

```
data > double_15.csv
1  8.0000000119209289551e-01
2  4.000000059604644775e-01
3  4.000000059604644775e-01
4  -4.000000059604644775e-01
5  0.0000000000000000e+00
6  4.000000059604644775e-01
7  0.0000000000000000e+00
8  4.000000059604644775e-01
9  -4.000000059604644775e-01
10 -8.000000119209289551e-01
11 0.0000000000000000e+00
12 4.000000059604644775e-01
13 0.0000000000000000e+00
14 0.0000000000000000e+00
15 4.000000059604644775e-01
16 4.000000059604644775e-01
17 4.000000059604644775e-01
18 0.0000000000000000e+00
19 0.0000000000000000e+00
20 0.0000000000000000e+00
21 4.000000059604644775e-01
22 8.000000119209289551e-01
23 4.000000059604644775e-01
24 0.0000000000000000e+00
25 0.0000000000000000e+00
26 4.000000059604644775e-01
27 4.000000059604644775e-01
```

图 27: 数据记录格式概览

共 2500 行, 1 列, 记录的一列数据为 2500 个点分别对应的纵坐标。横坐标是时间数据, 无论在哪一组数据中总是相同, 为了节省存储空间略去不记。

在采集的过程中, 可以将示波器产生的每一个波形都记录下来, 也可以在检测到衰变时再记录波形。这在下一部分会进行讨论。[\[4.4.2\]](#)

4.4 实验结果

4.4.1 模拟结果

该实验中进行的一个重要工作是在低采样率下修正峰的高度, 所以必然要在实际数据上验证修正方法的准确性。但是, 直接在低采样率下获得的数据本身就是不准确的, 即使对其进行修复也无法验证到底是否准确修复。对此, 我们采取的方式是: 先在获得高采样率下的数据, 然后在得到的数据上间隔取点, 构成等效的低采样率的数据。用该数据运行修正方法, 与高采样率下的峰值作为参考值进行比对, 从而获知修正的精确度。

我们在 2000 份左右的数据上用上述方法运行过测试, 结果显示该方法在超过 80% 的数据上能够将峰的高度修正至参考值误差的 5% 以内。其中, 能够在下降沿上寻找采样点的情形修正更为准确, 例如下图。(注: 由于模拟测试是较早期进行的, 未留存图片, 这里用实际采样中获得的相同情况的图像进行说明)

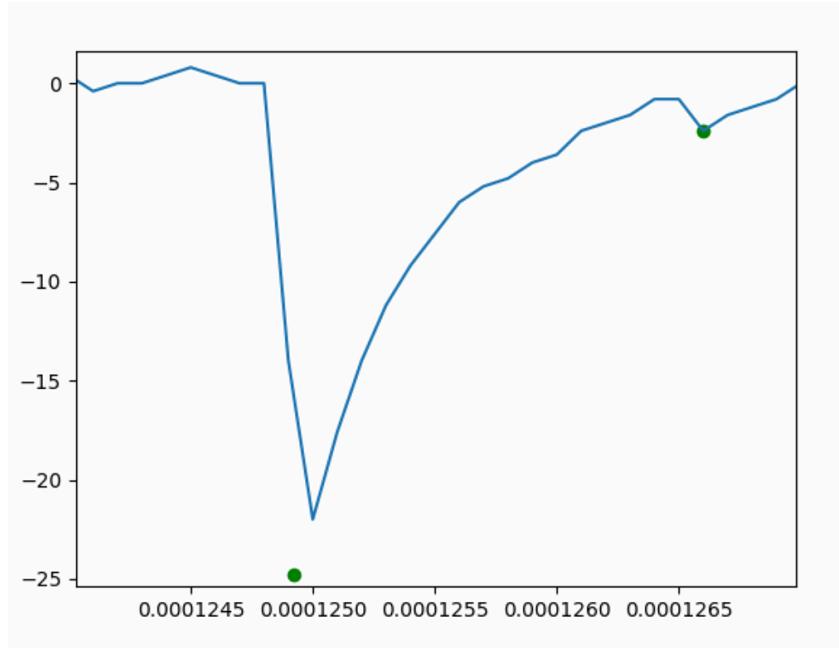


图 28: 较好的修正情况

如上图, 图上看似是一个正确采样的峰, 但仔细观察可以发现左侧下降沿上有一个轻微的拐弯点, 上下两段并不形成一条直线, 显然不是真实的下降沿。由此可以判断, 示波器在该拐点进行采样之后略过了顶峰, 而是直接采到了上升沿的点。可以想像, 如果把下降沿的前半段进行直线延伸, 把上升沿也按照曲线趋势往回延伸, 两条虚构的延展线大约会交于绿色点, 即预测峰位置。这从直观感知上也可以验证修复的正确性。

如果高度较小的衰变峰也展现出上述特征, 那么该方法也可以进行比较有效的修复。如下图。

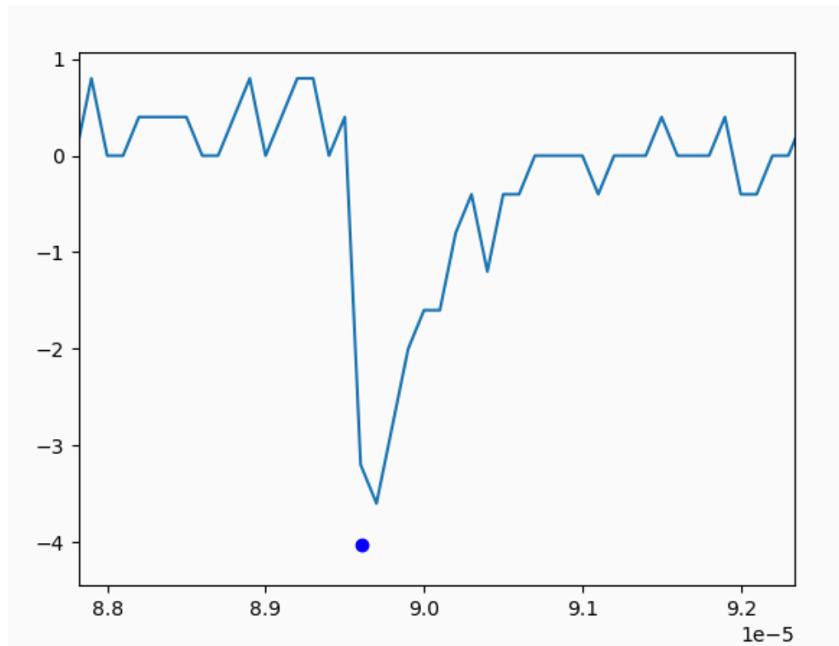


图 29: 衰变峰上较好的修正情况

当然也会有一些并不理想的修正情况。其中预测可能不准确的采样结果一例如下。产生误差的原因等见 [5.1]。

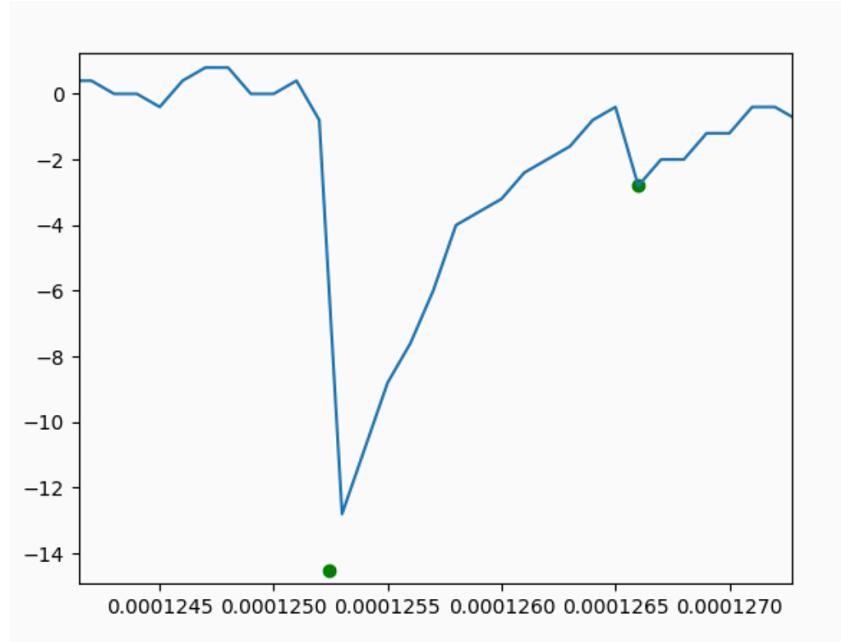


图 30: 可能不准确的修正情况

4.4.2 实际采样结果

在最后一次长时间的数据采集中，我们在低采样率下进行了总时长 205.4h 的采集。在采集时，我们只记录了检测到衰变信号的数据，后续的数据分析也只使用这些带有衰变的数据进行。在往届同学留下的程序中，所有数据都记录下来（包括单峰的数据）并参与了分析的过程。当然，这两种方式在最后处理数据时都不会构成错误。大部分的单峰数据虽然不能找到衰变的信号，但是肉眼基本可以判断如此明显的峰基本就是来自于 μ 子，因此单峰参与能量分布计算是合理的。如果单峰的数据也记录下来，最终形成的数据量就会相当大。保险起见，我们最后只收集了检测到衰变信号的数据，以确保获取到的都是 μ 子。

在上述采集时间内，我们共获取到 6536 个 μ 子，平均个小时 31.82 个。我们也使用上一届编写的程序进行过长时间的采集；在一次总时长为 239.5h 的采集中，获取到的 μ 子数量为 2259 个，平均个小时 9.43 个。在其他次的采集中，往届所编写的程序每小时平均采集 μ 子数最多不超过 15 个；以此估计，我们的采样方法大致可以将 μ 子的采集效率提高至原来的 2 倍左右。

这样的比较并不完全严谨。首先，我们的程序和往届的程序并不是同时运行，因为不能同时有两个程序读取示波器的信息。这就不能保证实验时的真实 μ 子数是一致的。其次，我们所采用的寻峰算法不同，可能两个程序都会有一定的误识别。不过，粗略地可以认为，我们的程序效率大约是以往的两倍。

如果以收集到的全部峰（包括单峰）数据作为比较，我们的程序效率大约比以往提高 30%。这个指标与上面所说并不是一个指标，不过这个指标计算起来要更准确。因为往届的算法一次采样必定只能获取一个峰，最后获取的总峰数就是采样次数。利用这一点，只运行我们的程序，同时统计采样次数就可以计算。

只保留衰变数据尽管存储空间占用小，但多道分析时就不能划分太大的道数。我们所获取的数据最多支持 512 道分析，1024 道时图像质量会比较差。下面给出的都是 256 道下的结果。

（注：因为没有能量标定，这里每一道的能量值无实际意义。事实上，之前的所有图中峰的高度也只有相对大小意义。）

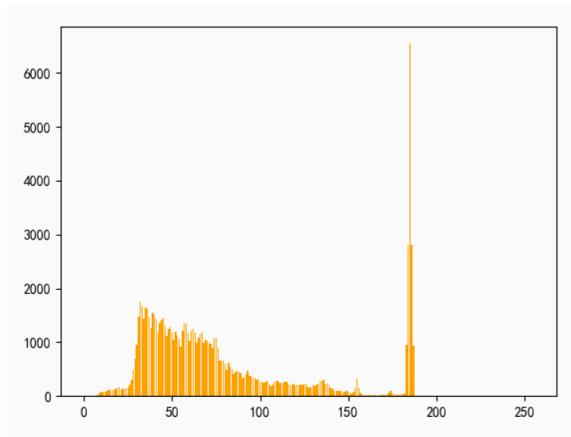
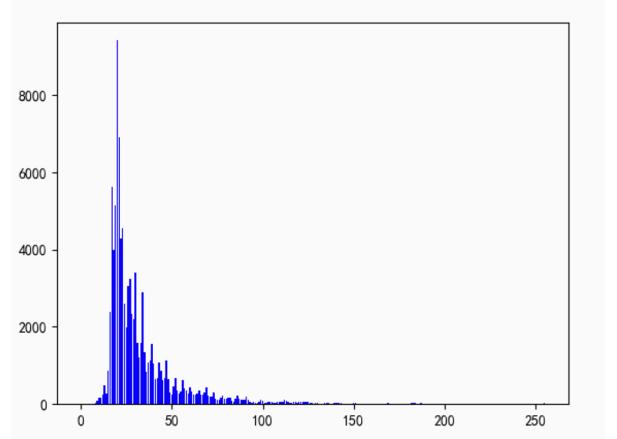
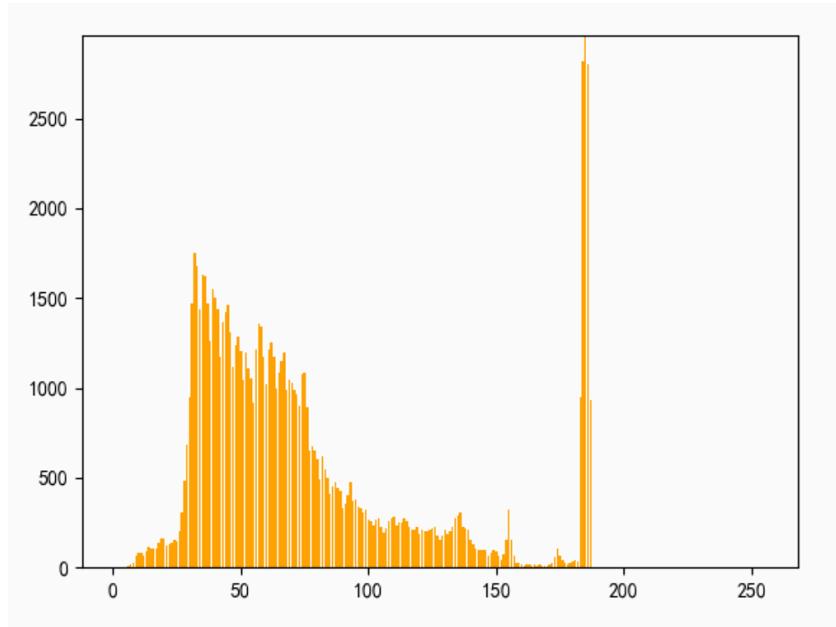
图 31: μ 子能量分布

图 32: 电子能量分布

μ 子能量分布中, 右侧异常突出的线在往届所提供的能量分布图上也有出现, 并且在我们的图像上这条线还要更高。我们猜测这是平顶所造成的, 有可能虽然平顶经过修复, 但是修复后的值也是集中在这一高度。我们可以对 μ 子的能量分布作一个局部放大, 截掉上面过高的部分, 这样就能看出来轮廓符合 μ 子能量分布的外形。

图 33: 放大的 μ 子能量分布

我们利用这些 μ 子计算得的平均寿命为 $3.16\mu s$. 与理论值相比稍微偏长。在处理时, 认为主次峰间隔在 $1\mu s \sim 10\mu s$ 范围之外的都不是 μ 子。

4.4.3 平顶修复结果

平顶修复是我们产生的一个副产品。因为我们对每一个峰都从左右两侧估计了新的峰高度, 平顶自然也就被补掉了。但是, 由于实际数据本身就是有平顶的, 所以虽然我们做了修复, 也无法对修复的精度作精确计算。不过, 由肉眼估计, 修复效果还是不错的。下面是修复的情形。

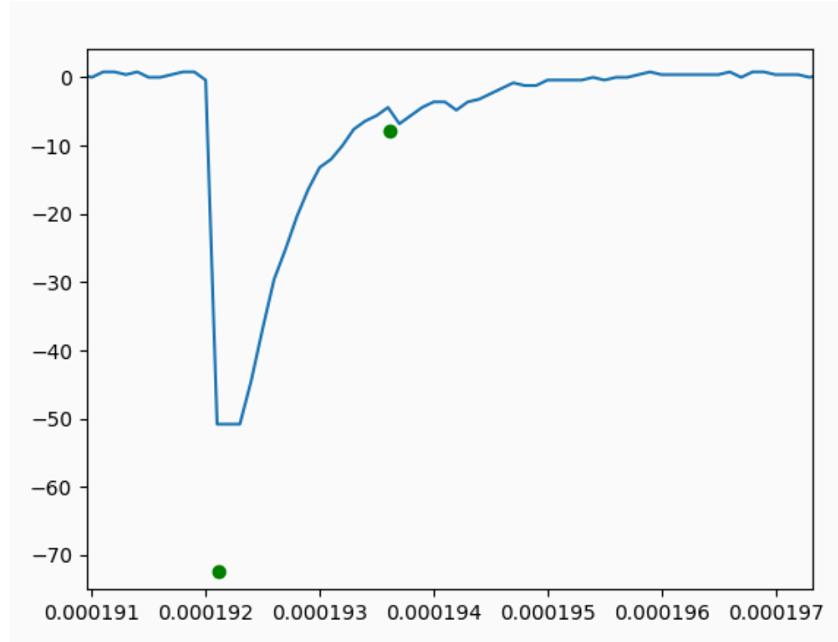


图 34: 平顶修正

不过, 平顶修正并没有很大的用处, 因为动能如此大的 μ 子几乎可以肯定会被穿透, 检测到衰变信号也可能如上图所示, 更可能是误判。

5 分析与讨论

5.1 误差来源

我们的方案可能产生的误差比往届的要多。一方面, 在寻峰中会产生误差, 即会漏掉峰或者计入一些并不是峰的东西; 另一方面, 峰值的修正也会有一定的偏差。不理想的修正出现的情形主要有:

- 下降沿上不能够找到点, 必然产生随机误差。极端的情况下可能达到 10%-15%.
- 真实波形在波峰处并不完全尖锐, 不符合预测模型。这种情形只有在模拟测试时才能发现, 并且往往导致比较严重的误差, 可能达到 20%.

如图 [30], 就是第一种情况下产生的可能偏差。

另外, 我们设计的修正方法主要对主峰 (μ 子峰) 适用, 次峰往往不像主峰一样从 0 电压处开始下降, 下降沿也未必能找到点。我们也没有专门测算过次峰上升沿的弯曲程度, 不知道其是否与主峰相同。因此, 对次峰进行的高度修正准确度会相对低。

5.2 误差分析

5.2.1 寻峰误差

寻峰相关的误差由陈科同学进行论证。下面是引自他的论文的论述:

除非主峰极其罕见的几乎叠加出现, 否则对于主峰的搜寻理论上而言是没有误判的, 主要的误判来源是来自次峰的搜寻。

对于次峰, 若是我们在凸包过程中搜寻到的, 那大概率也的确会是次峰。但若是在维护区间最小值过程中找到的, 就很可能会是噪声峰, 这种噪声峰难以剔除, 因为他与次峰十分的相似, 很难找出他们之前不同的某

个特殊特征来进行筛选判断。实验结果也说明我们的次峰标准寻找过于宽松, 导致 μ 子的寿命长于平均值。对于这个现在也没有很好的解决思路, 但可以把他看作拟合匹配问题, 这是有可能被转化为一个优化问题, 运行一些迭代的算法进行计算与判断的。

5.2.2 峰值修正误差

对于下降沿采不到点的情况, 理论上就会产生较大的误差, 只能尽量减小。只要维持低的采样率, 就必然会有这种情况出现。因此, 我们认为想要解决该种情况的问题, 不如研究其他的采样方法来得有效。[\[5.3.1\]](#)

对于真实波形不符合预测模型的情况, 我们也观察到多次。但在理论上, 峰的顶部应该是完全尖锐的。观察到这种情况下波峰位置的抖动往往比较大, 我们初步认为这是噪声或产生峰时的其他干扰所导致。

对于次峰的修正, 如果要修正次峰上的误差, 就需要有确定次峰下降沿起点的方法。但这是相当麻烦的, 因为次峰很有可能在主峰的上升沿上出现, 其左侧不一定是水平线, 确定起来比较复杂。另外, 次峰的下降沿很短, 采到点的概率也相对小。最后, 从修正的价值上来看, 次峰其实没有太大修正的必要, 因为次峰是电子峰, 所产生的是电子的能量分布。我们主要想要得到的是 μ 子的能量分布, 电子的能量分布可以看作一个副产品。

5.2.3 最终测量结果误差分析

平均衰变时间 $t = 3.16\mu s$, 相对于理论值其实偏高不少。时间上的度量与峰高度修正的算法显然无关, 我们选择的采样率也不至于在采集时把峰漏掉, 因此我们认为主要的误差来自寻峰算法。进一步推测, 寻峰算法最主要的误差应该不是漏峰, 而是把一些并不是次峰的波形当做了次峰。这些被当作次峰的杂波往往距离主峰有一定距离, 从而拉长了平均衰变时间。我们曾经将 μ 子的寿命上限调整到 $20\mu s$ 进行实验, 结果 μ 子的平均寿命计算结果达到了 $6\mu s$ 以上, 显然在距离主峰 $10 \sim 20\mu s$ 的区间内又有误被当作次峰的点。这也是我们作出此分析的主要原因。

对于 μ 子和电子的能量分布图, 我们得到的图像外观上与往届得到的结果大致相同。由于能量没有标定, 因此不进行定量的分析。

5.3 实验分析

5.3.1 关于 Sensor-CASSY 2 的可用性

在上述报告中, 我们多次提到 Sensor-CASSY 2 具有优秀的连续采集能力, 但是我们没有采用。原因在于, Sensor-CASSY 2 的极限采样频率只有 $1\mu s^{-1}$, 即每 $1\mu s$ 采样一个点。这个采样率是该实验无法接受的。在采样率分析 [\[4.1.3\]](#) 中, 我们所确定并最终使用的采样率已经是可能进行有效修复的最小采样率, 而这个采样率恰好是 $0.1\mu s^{-1}$, 为 Sensor-CASSY 2 极限采样率的 10 倍。我们也专门在 $1\mu s^{-1}$ 的采样率下进行观察, 发现该采样率下甚至有可能会把整个峰给漏掉, 这显然是不可接受的。

当然, 这并不是说 Sensor-CASSY 2 完全没有用于采样的可能性。我们有两种思路: 其一, Sensor-CASSY 2 在极限采样率下并不是所有采集到的数据都不可用。如果能使用 [\[4.1.4\]](#) 进行修正, 应该还是能还原相当精度的结果的。这样, 采集到的信号将会只有一部分可用。但是, 由于 Sensor-CASSY 2 的采样时间很长, 两者权衡之下, 最终平均探测到的 μ 子数量可能会比原来更多。在我们的实际测试中, Sensor-CASSY 2 至少可以连续采样 1s(可能伴有少许数据延迟), 即有效采样时间比示波器要高 2~3 个数量级, 因此在理论上还是可行的。但是, 这样的做法是否会影响到最终平均寿命和能量分布的计算, 目前还没有讨论。

其二, 可以考虑 MCA Box。该器件在官网上给出的实验示例中就包括 μ 子的探测(见[此处](#)), 因此该附件应该是具有足够的分辨能力的; 并且该器件应该也不是简单的计数器件, 一定能够记录峰的高度数据, 否则无法绘制出能量分布。目前不清楚其是否能够记录时序数据。文档在 CASSY SDK 的说明文档中给出了链接, 但是 Leybold 的编程手册全部为德文, 阅读起来也有一定的困难。

6 实验总结

在本次实验中, 我们更新了往届遗留的采样方式, 创建了峰值修正算法, 使得 80% 以上的峰值复原至参考值的 5% 误差以内, 并解决了平顶修正问题; 同时, 更新了寻峰算法, 使得算法能够在一组信号中具有多个峰的场景下工作。

经过我们的改造, 现在的采样方式同等时间下能够找到的 μ 子数量大约是以往的两倍。不过, 测得 μ 子的平均寿命 $3.16\mu s$ 有些偏大, 寻峰算法还有待改正; Sensor-CASSY 2 的潜力也还没有成功开发出来。

7 课题总结

7.1 仪器系统照片

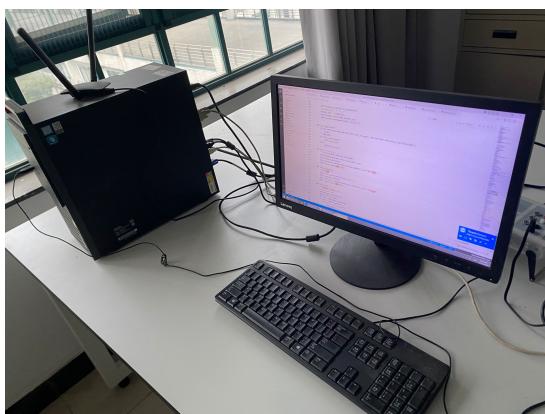


图 35: 电脑



图 36: 塑料晶体

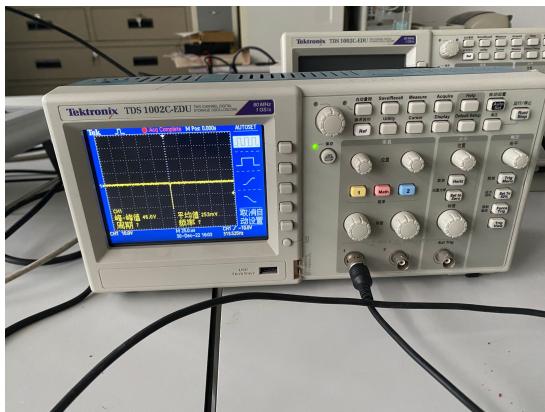


图 37: 示波器



图 38: CASSY-Sensor(老款)



图 39: 连接系统全貌

7.2 课题所用器材清单

Tektronix TDS 1002C-EDU 示波器两台, Windows8 电脑一台, Sensor-CASSY 2 一个, 塑料晶体一个, 光电倍增管一个, 三通头一个, 各种型号线缆若干。

7.3 花絮和留言

ck: 我就只写了寻峰和 GUI, 还被 csy 吐槽代码写得丑, 最后寻峰算法也还要很多问题, 转移到采样率降低的情况也没有经过细致的测试, GUI 也是赶工赶出来的, 不过功能倒是都有了。这个 GUI 写的是真的久, 从头开始搭还是过于耗费时间了, 不过这也和我比较摸鱼有关, 一般除实验课也不会主动去写代码, 都是最后补的天, 下次还是要在平时努力。其他所有的工作都是抱 csy 的大腿完成的, 被带飞真的爽。

下面请队友 c · 实验主导大腿 · sy 发表完结感言:

csy: 感谢我的队友 ck 的发言。(鼓掌) 啪啪啪啪啪↑

这个实验总体感觉工作量还是挺大的。当时提出降低采样率, 觉得只是一个很小的话题, 没想到真就做了一个学期……可能和中间有些时候没思路摸鱼也有关系吧 (x). 另外感觉分工还是挺重要的, 我自己做可能得累死。寻峰算法比较接近 OI, ck 全包了。GUI 也是。不过我自己由于这一次没有查询多少相关文献, 完全就事论事自己想办法, 也还是挺耗脑筋的。最后没能把 Sensor-CASSY 2 用上, 模拟那边也没来得及看, 算是一点小遗憾。

参考文献

- [1] 吕治严. 宇宙线 μ 子寿命测量实验及电子学设计 [D]. 中国科学技术大学, 2009.
- [2] 宇宙射线 μ 子的探测与测量, 刘畅.(前几届同学的工作)
- [3] 宇宙射线 μ 子的探测与测量, 杨凯, 陈星, 鲍德松.