97年度 長島研4年生卒業実験

宇宙線中のπ^+中間子の検出

平成10年 10月 21日

新田 和範
原 康二
三宅 秀樹
鄭 鋒

目 次

1 イントロダクション **8**

1.1 動機 8

1.2 背景 8

1.3 概要 8

2 データ収集システム **12**

2.1 検出器 12

2.2 エレクトロニクス 12

2.3 トリガーロジック 12

2.3.1 トリガー系 12

2.3.2 アナログ系 13

2.3.3 TDC-stop 系 13

2.4 オンラインプログラム 13

3 較正 **17**

3.1 PMT 17

3.2 TDC の較正 19

3.3 ADC の較正 21

3.4 ディジタルオシロスコープ 23

4 モンテカルロシミュレーション **26**

4.1 目的 26

4.1.1 π^+の$\frac{dE}{dX}$の計算法 26

4.1.2 e^+の$\frac{dE}{dX}$の計算法 27

4.2 手順 28

4.3 シミュレーション 28

4.3.1 アクセプタンスの計算 28

4.3.2 検出効率の計算 30

4.3.3 立体角の計算 33

5 実験の解析 **35**

5.1 データ解析の方法 35

5.2 イベント選別 35

5.3 バックグラウンドの評価 43

5.4 イベントの解析 44
5.4.1 stop μ イベントに関する μの寿命 44
5.4.2 π-μ イベントでのπ及びμの寿命 44
5.5 結果 46
5.5.1 πのイベント数の見積もり 46
5.5.2 π^+フラックスの計算 51
5.5.3 期待されるπ^+イベント数の計算 51
5.6 π及びμのエネルギーについて 54
5.6.1 πとμのパルスの分割 54
5.6.2 πのエネルギー 54

6 結論 **57**

謝辞 **58**

A ソフトウェア **60**
A.1 オンラインプログラム 60
A.1.1 ヘッダーファイル 60
A.1.2 メインプログラム 62
A.1.3 サブルーチン 69
A.2 サンプルプログラム 76

図 目 次

1.1 トリプルパルス の概念図 9
1.2 検出器の略図 9
1.3 実験の流れ図 10

2.1 検出器の構造図 14
2.2 ブロック図 (トリガー系) 15
2.3 ブロック図 (アナログ&TDC-stop 系) 16

3.1 プラトーの測定 17
3.2 プラトーの例 18
3.3 検出効率の測定 18
3.4 TDC の較正のロジック 20
3.5 TDC カウントと時間の関係 20
3.6 ADC の較正 21
3.7 ADC 値 22
3.8 ディジタルオシロスコープの ADC 値 24
3.9 ADC 値の減衰 25

4.1 アクセプタンス計算の概念図 29
4.2 検出効率計算の概念図 31
4.3 立体角計算の概念図 33

5.1 πイベント 36
5.2 πイベント 37
5.3 πイベント 38
5.4 πイベント 39
5.5 πイベント 40
5.6 πイベント 41
5.7 πイベント 42
5.8 μの寿命 45
5.9 π イベントでの likelihood 関数 47
5.10 $\pi\mu$e イベントでの likelihood 関数 48
5.11 得られたπ イベント及び最尤法で求まった崩壊曲線 49
5.12 得られた$\pi\mu$e イベント及び最尤法で求まった崩壊曲線 50
5.13 パルスの fitting 55

5.14 分解したパルス (a)π,(b)μ 56

表 目 次

1.1 π^+、 μ^+、e^+、ν_μ の性質 [2] 11

3.1 各 PMT の性能 19
3.2 TDC の仕様 19
3.3 ADC の仕様 21
3.4 ディジタルオシロスコープの仕様 23

4.1 アクセプタンスの計算に用いた仮定 29
4.2 検出効率の計算に用いた仮定 30

5.1 トリガーイベント 43
5.2 πのイベント選別 43
5.3 実験及びシミュレーションの結果 51
5.4 U.Camerini と我々の実験との比較 52
5.5 πのエネルギー (単位:MeV) 54

Abstract

今回我々の実験では、プラスチックシンチレーターと光電子増倍管 (PMT) 及びディジタルオシロスコープを用いて、$\pi^+ \to \mu^+ \to e^+$ の Decay Chain を観測した。

実験は 2268966 秒 (26 日 6 時間 16 分 6 秒) 間かけて行なった。データ解析の結果、13 個の π^+候補が得られた。これらのイベントデータを解析した結果、得られたμ^+の寿命は $2.02^{+0.70}_{-0.47}\mu$s で、π^+の寿命は $20.60^{+8.46}_{-5.27}$ns である。これらの値はそれぞれ期待値の 2.2μs 及び 26ns とほぼ一致したため、我々の実験では本物のπ^+を同定することができたと考えられる。

第 1 章

イントロダクション

1.1 動機

素粒子物理学の目的は、物質の究極粒子 (素粒子) とそれを支配する基本法則を明らかにする事である。その幕開けは 1930-40 年代のμ粒子やπ中間子といった素粒子の発見と言える。その後 K 中間子、Λ粒子などの寿命の短い新粒子が続々と発見され、現在では「 素粒子 」の名を冠するものは 100 種類を優に越えるまでに至っている。

これらは極めて短い寿命のため、実験的には直接捕らえることが困難である。そのため、それらの崩壊生成物であるπ中間子やレプトン等を測定してやることにより、対象粒子を研究するのが一般的となっている。したがって、崩壊生成物の諸性質を知る事が非常に重要である。

今回我々が行なった実験は、π中間子の存在を再認識する事を目的とする。また、基本的な性質としてπ中間子の寿命とフラックスを求めた。

1.2 背景

地球には、宇宙空間から宇宙線と呼ばれる、エネルギーの高い放射線が毎秒約 1 個$/cm^2$降り注いでいる。これは大部分が陽子からなり、地表付近で大気によって相互作用を繰り返し$\pi^{\pm}$、$\mu^{\pm}$、ν、γ等の比較的エネルギーの低い粒子に変換されて地上に到達する。

π中間子は核子間に働く核力を担う媒介粒子として、1935 年湯川によって導入された。その後 1947 年 C.F.Powell らのエマルジョンを用いた実験により、宇宙線の中から初めてπ中間子が発見された。また 1948 年に U.Camerini たちは原子核乾板を天井に吊し、下の床から入射したπ中間子を観測する事により、その寿命の測定に成功した。長島研究室でも過去床から上向きのπ^+中間子の観測実験を行った [1]。

今回の実験ではプラスチックシンチレーターと光電子増倍管、及びディジタルオシロスコープを用い、天井からの下向きのπ^+中間子の検出を試みた。

なお、我々の実験でπ^+ を同定する過程で出てくる主な粒子の性質は以下の通り。

1.3 概要

我々の実験で検出するπ中間子は、おもに宇宙線中の陽子等の粒子と天井中の物質の核子との衝突によって生成されたものである。そのうちπ^+中間子は、我々の検出器のプラスチックシンチ

レーター中で止まった際$\pi^+ \to \mu^+ \to e^+$ のディケイチェインが起こる。これにより図 1.1のようなトリプルパルスが発生する。

このディケイチェインを観測する事によりπ^+中間子を検出することができる。ただし、π^- は崩壊以前にシンチレーター中の物質に吸収され吸収反応を起こすので、今回の実験では観測されない。

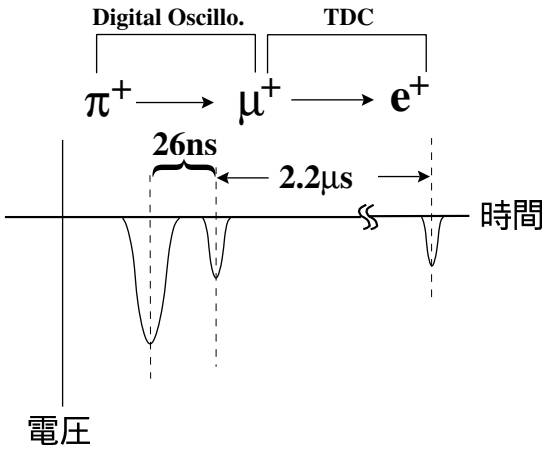


図 1.1: トリプルパルス の概念図

今回の実験では 7 枚のプラスチックシンチレーター (長さ:100cm, 幅:33cm, 厚さ:1cm) とそれに取りつけるライトガイド 及び PMT で作られた図 1.2のような検出器を用いた。

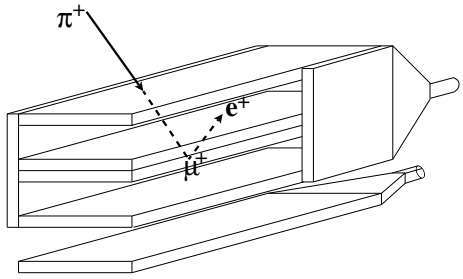


図 1.2: 検出器の略図

$\pi^+ \to \mu^+$と$\mu^+ \to e^+$の崩壊は、シンチレーターの端に付けている光電子増倍管によって電気信号として取りだされ、NIM* CAMAC† モジュールを通してから、NEC の PC-9801 でデータとして保存される。そのデータを Sun の Unix Server にネットワークを通して送り、そこでデータの解析を行った。全体の流れは図 1.3である。

ここで TDC はμe 崩壊のイベントを見るために使っている。

*Nuclear Instrument Modules の略、原子核関連分野での計測システムの国際的規格.
†Computer Automated Measurement And Control の略、データ処理のためのエレクトロニクス規格.

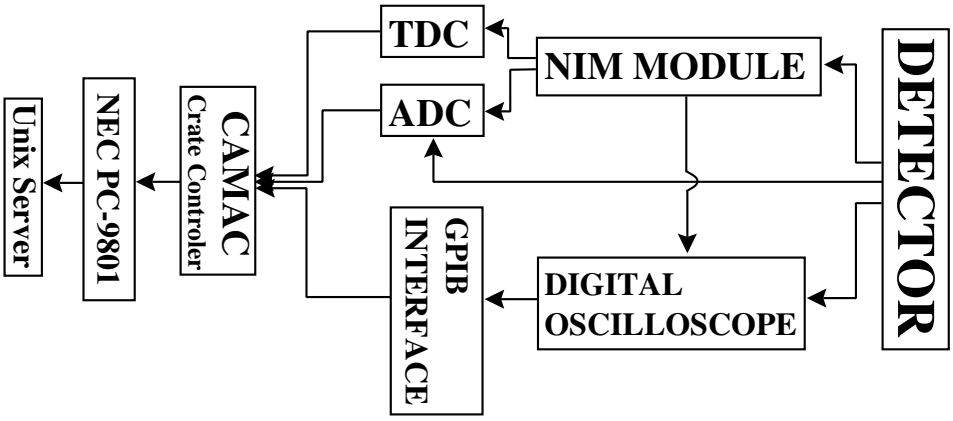


図 1.3: 実験の流れ図

ディジタルオシロスコープでは図 1.1のようなトリプルイベントの内、$\pi^+ \rightarrow \mu^+$のダブルパルスイベントのデータを保存することができる。

このディジタルオシロスコープのデータからπ^+の寿命が計算され、それを期待値の $26ns$ と比較することによって、$\pi\mu$崩壊のイベントを検出することができる。ADC はπ^+がシンチレーターを通過する際に落としたエネルギーを見る。またシンチレーターで止まったπ^+中間子の崩壊により、μ^+のエネルギーは $4.12\mathrm{MeV}$ [‡] であるかどうかもデータ処理する際に一つの判断条件としてチェックした。このため今回の実験はほぼバックグラウンドフリーだと考えられる。

[‡] $\pi^+ \rightarrow \mu^+ + \nu_\mu$は 2 体崩壊のため、$\mu^+$の運動エネルギーが次のように計算できる. エネルギー保存則と運動量保存則により $m_\pi = E_\mu + E_\nu; E_\mu^2 - m_\mu^2 = E_\nu^2 - m_\nu^2$ (自然単位系) が得られる。これに m_π, m_μ, m_ν の値 を代入すると $E_\mu \simeq 109.78 MeV$ が得られ、これからμ の質量を引くと 運動エネルギー $4.12\mathrm{MeV}$ が得られる。

π^+:	
	崩壊モード : $\pi^+ \to \mu^+\nu_\mu$ 99.9877%
	寿命: $2.6 \times 10^{-8}s$
	質量: 139.5685 MeV
μ^+:	
	崩壊モード : $\mu^+ \to e^+\nu_e\bar{\nu_\mu} \approx 100\%$
	寿命: $2.20 \times 10^{-6}s$
	質量: 105.6584 MeV
e^+:	
	寿命: 安定
	質量: 0.5110 MeV
ν_μ :	
	寿命: 安定
	質量: $<$ 0.25 MeV

表 1.1: π^+、 μ^+、e^+、ν_μ の性質 [2]

第 2 章

データ収集システム

この章では、我々が用いたデータ収集システムについて、検出器、エレクトロニクス、トリガーロジック、及びオンラインプログラムに分けて述べる。

2.1 検出器

我々の用いた検出器は、図 2.1のような構造を持っている。すなわち、1 枚のトリガーカウンターを囲むように 4 枚のアンタイカウンターを並べ (トリガーカウンターの下方に 2 枚)、その上方に 2 枚のコインシデンスカウンターを敷いてある。
各カウンターは $100cm \times 33cm \times 1cm$ の大きさを持つプラスティックシンチレーターに 2 インチ PMT を、図にあるように 1 個ないし 2 個取りつけたものである。
このような形状に組む事によって、上方から飛来してトリガーカウンター中で静止するような荷電粒子及び、その崩壊の結果放出される荷電粒子を選択的に検出する事が実現できる。

2.2 エレクトロニクス

我々はエレクトロニクスとして、ADC、TDC を始めとする各種 CAMAC/NIM モジュール、ディジタルオシロスコープ、及び高圧電源などを用いた。ADC、TDC、それにディジタルオシロスコープの使用目的及び性能その他については、後の章に述べてある。

2.3 トリガーロジック

以上に述べたオンラインシステムは、大別するとトリガー系、アナログ系、TDC-stop 系の 3 つの系より成っている。

2.3.1 トリガー系

トリガー系とは、システム全体に測定開始のシグナル (トリガー) を送る系である。測定開始のシグナルと言うのは、前述した様に上方から飛来した荷電粒子が、トリガーカウンター中に静止

したという信号であるから、次のように表せる。なお、用いた記号については、図 2.1参照の事。

$$(T_1 \cap T_2 \cap T_3 \cap T_4 \cap A_1) \cap \overline{(A_2 \cup A_3 \cup C_1 \cup C_2)}$$

このトリガー条件に従ってロジックを組んだものが図 2.2である。なお図中にも示してあるが、スレッショルドは–20mV、パルス幅は特に示していなければ 100ns、wide としてあるものは 200ns として調整してある。

2.3.2 アナログ系

この系では、ADC 及びディジタルオシロスコープに検出器からの生パルスを送っている。ただし、そのままではトリガーパルス (開始信号) と生パルス (実際の波形) との同期が取れていない (トリガー系に入ったパルスが CAMAC/NIM モジュールを通過する時間を無視できない) 為、生パルスに対して 76ns のケーブルディレイを与えた。

2.3.3 TDC-stop 系

この系は、TDC にμ-e 崩壊に由来する電子検出のシグナルを送る系である。このシグナルに関しては、トリガーカウンターを囲む 4 枚の検出器のいずれかがトリガーカウンターと同時に鳴るという条件、すなわち

$$(T_3 \cap T_4) \cap (A_1 \cup A_2 \cup A_3 \cup C_1)$$

が TDC-stop の条件となる。
ただ、このままだと偽のシグナルがかかった時にいつまでも待ち続ける事になってしまう為、ゲートジェネレーターによって 20μs 遅延させたトリガー*を用いて、強制ストップをかけている。

2.4 オンラインプログラム

データ収集システムは、以下の様なプログラムを走らせた PC-9801 により制御される。プログラムはまず、全ての CAMAC モジュール及び、ディジタルオシロスコープを初期化し、ADC の LAM が立つのを待つ。ADC の LAM を確認したのち、TDC のデータを見る。

TDC チャンネルのうち、1 つでも 20μs より小さい値を示していれば、イベントの候補とみなし、ADC、TDC、ディジタルオシロスコープの波形、デッドタイムのデータをハードディスクに記録し、全てのモジュールをクリアーし、再び ADC の LAM が立つのを待つ。そうでなければ、デッドタイムのみを記録し、全てのモジュールをクリアーし、再び ADC の LAM を待つ。

付録 Aにプログラムの全文を示す。

*これはμの寿命 (2.2μs) のおよそ 9 倍に相当し、トリガーカウンター中に静止したμのほぼ全て (99.99%) が崩壊する時間である。

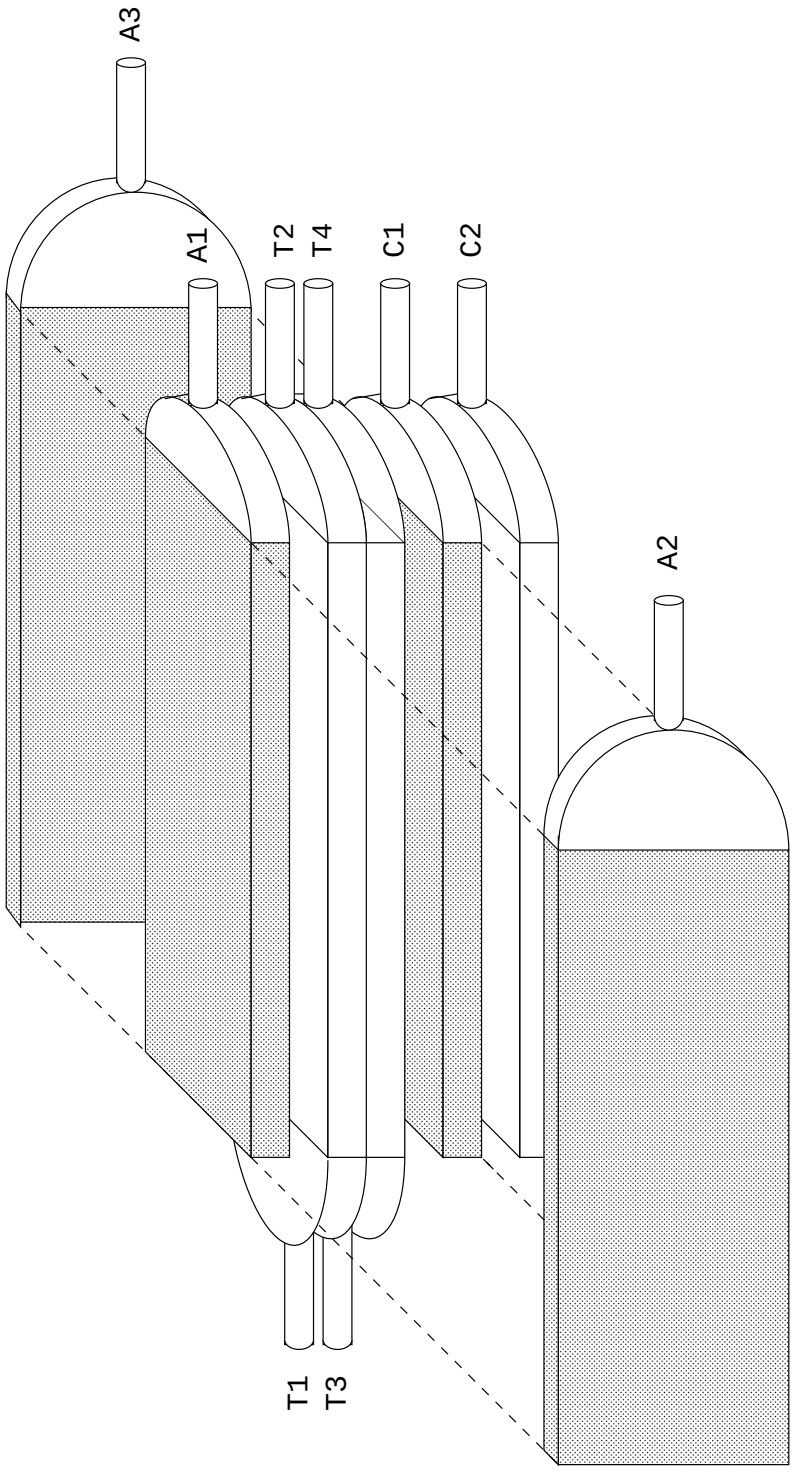
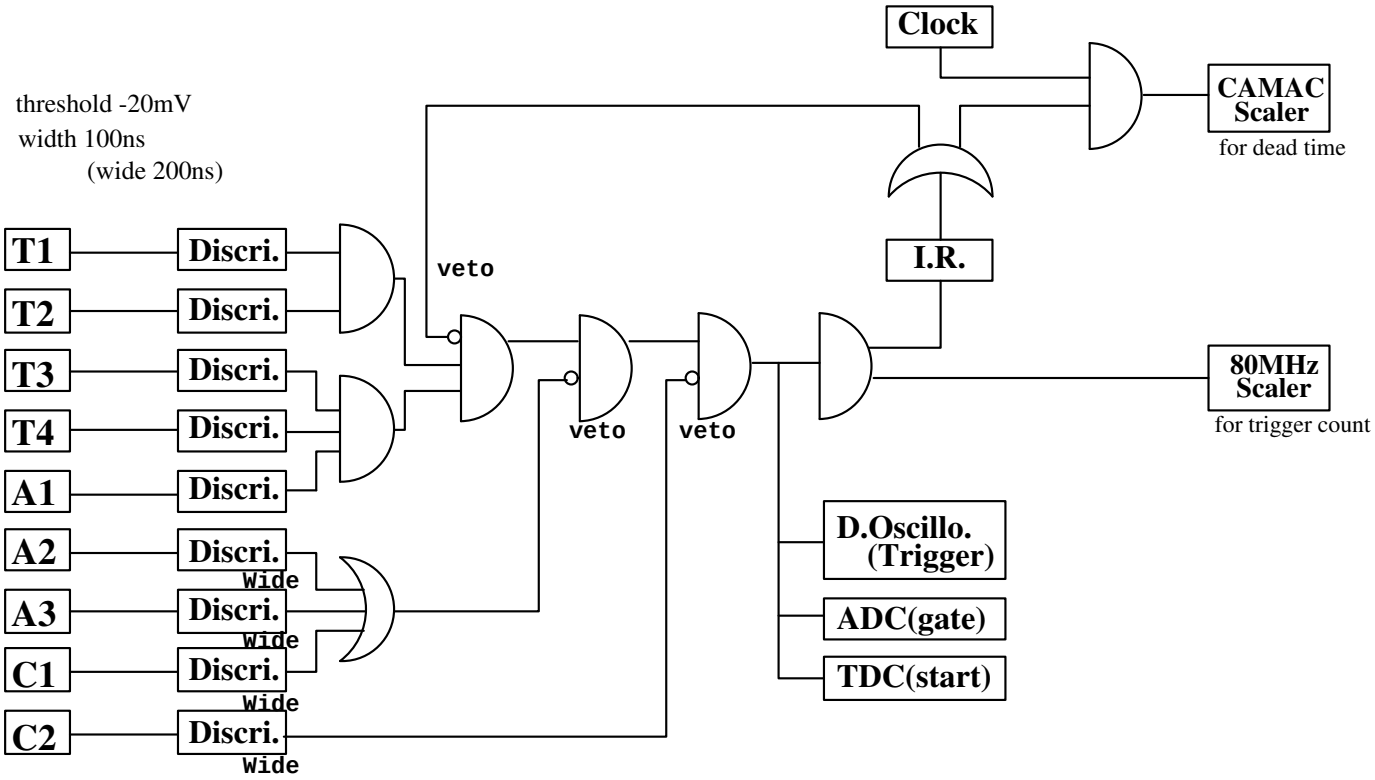


図 2.1: 検出器の構造図

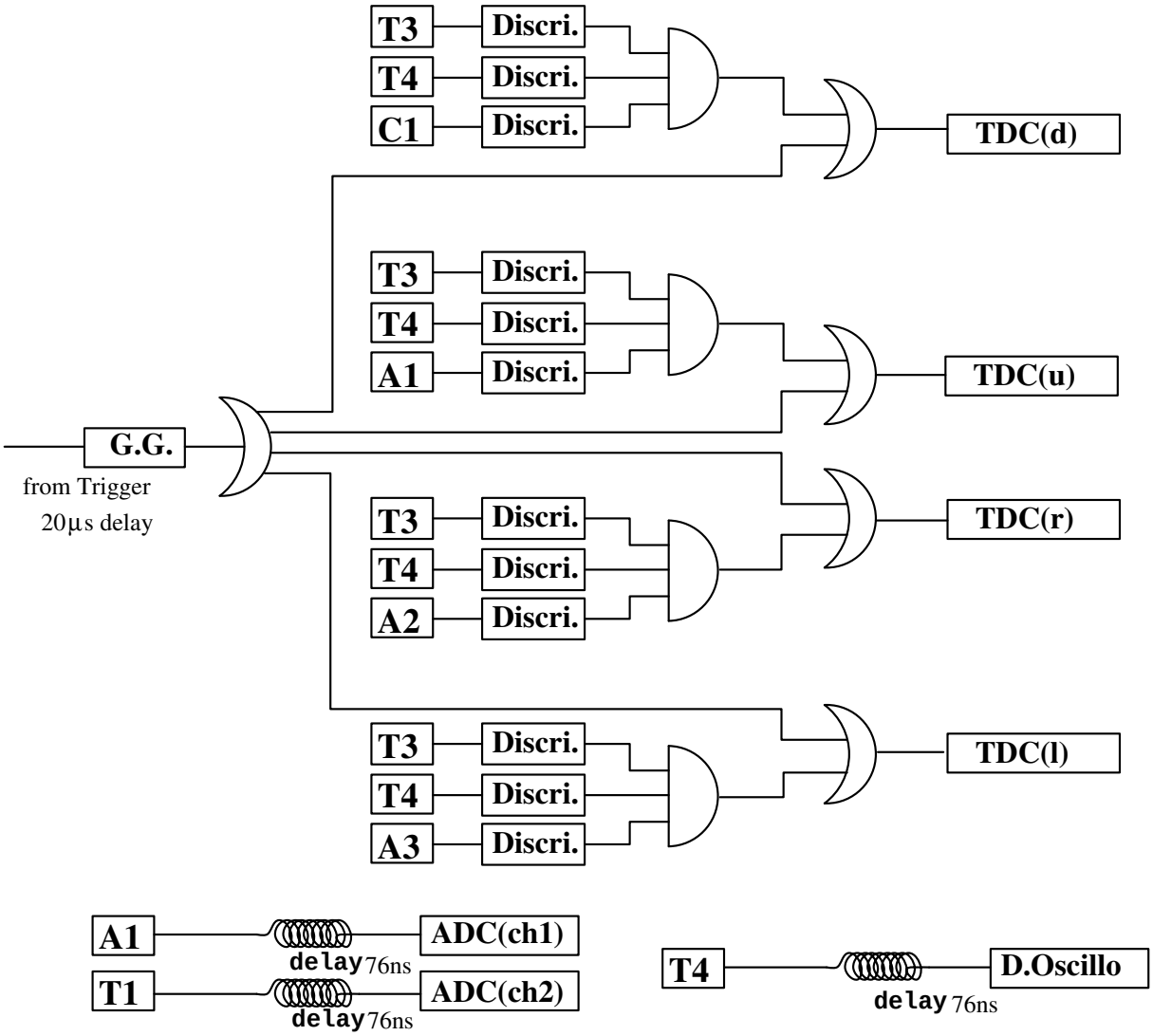
T3、T4 がトリガーカウンター、A1、T1、T2 がコインシデンスカウンター、A2、A3、C1、C2 がアンタイカウンターとなっている。なお網を掛けた部分は、μ-e 崩壊検出の際にトリガーカウンターのコインシデンスカウンターとして働く部分 (A1、A2、A3、C1) をさす。



Block Diagram(Trigger)

図 2.2: ブロック図 (トリガー系)

Discri. はディスクリミネーター、D.Oscillo はデジタルオシロスコープ、I.R. はインタラプトレジスター、Clock はクロックジェネレーターである。また、パルス幅については、wide と特に示したもの以外については $100ns$、wide としたものについては $200ns$ である。



Block Diagram(Analog&TDC)

図 2.3: ブロック図 (アナログ&TDC-stop 系)各記号については図 2.2参照の事。

第 3 章

較正

3.1 PMT

プラトー

PMT にかける電圧は、プラトー領域に設定しなければならない。そのために、図 3.1の様に 2 枚のレイヤーを重ねてコインシデンスをとって、電圧を変えながら計数率を測り、プラトー領域を調べた。なお、我々の実験ではディスクリミネーターの閾値は-20mV に設定した。

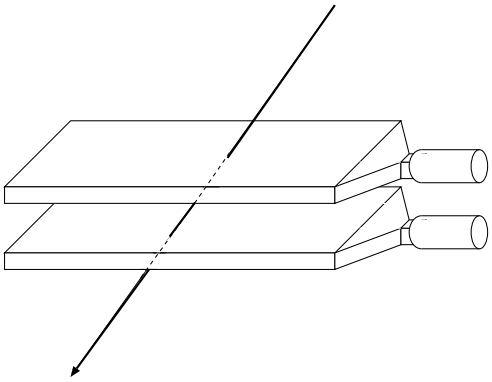


図 3.1: プラトーの測定

結果の一例を図 3.2に示す。これをもとに表 3.1のように各 PMT にかける電圧を決定した。

検出効率

検出効率を測りたい PMT のついたレイヤーを挟んで、3 枚のレイヤーを図 3.3のように重ねる。

$$\text{検出効率} = \frac{\text{全ての } PMT \text{のコインシデンス}}{\text{測りたい } PMT \text{以外の } PMT \text{のコインシデンス}}$$

として測った。

各 PMT の電圧、シングルレート、検出効率を表 3.1に示す。

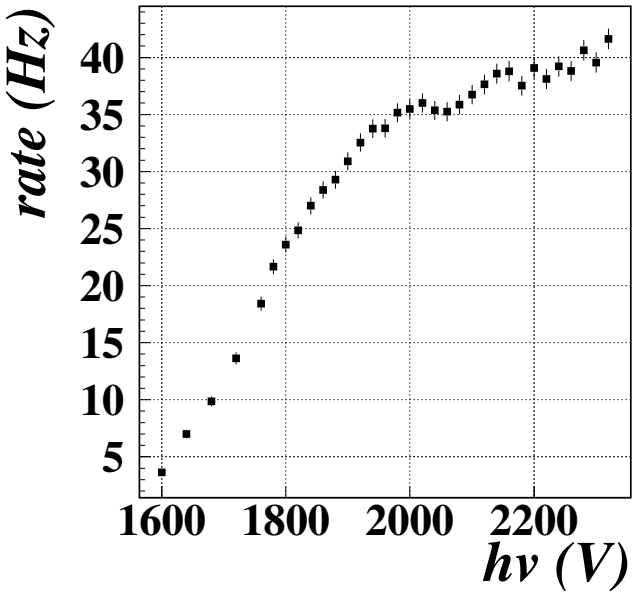


図 3.2: プラトーの例

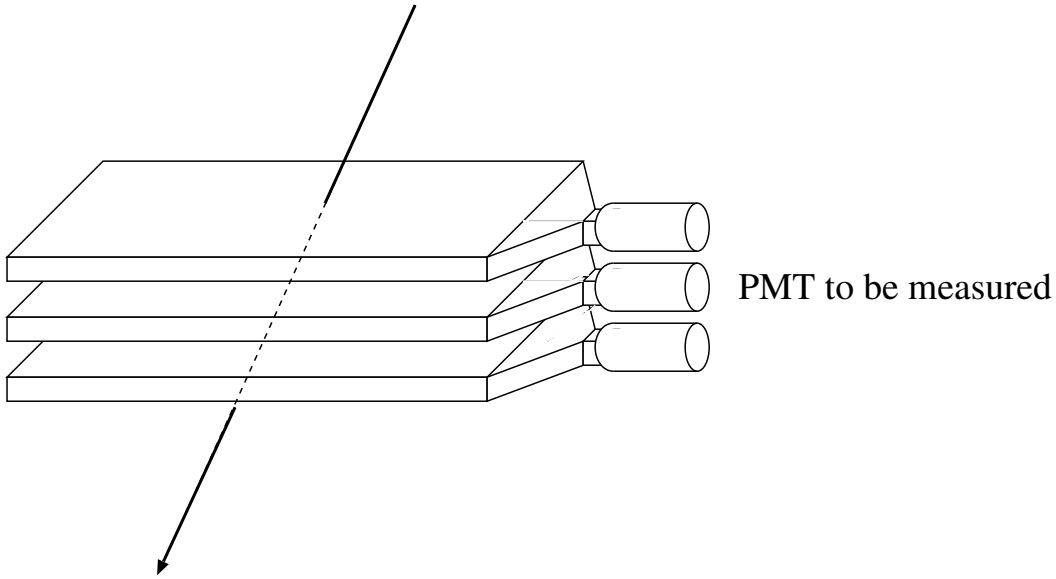


図 3.3: 検出効率の測定

PMT	電圧 (V)	シングルレート (Hz)	検出効率 $(\%)$
T1	2360	1413.9	93
T2	2300	734.2	94
T3	1980	234.1	94
T4	2160	312.2	97
A1	1780	160.8	93
A2	2200	414.3	93
A3	1880	178.1	95
C1	1900	232.2	94
C2	2120	317.1	94

表 3.1: 各 PMT の性能

3.2 TDC の較正

TDC はμ→e 崩壊を検出するために用いる。我々の実験では 20μs まで測るので、較正として 20μs までの TDC カウントと時間の関係を調べた。

我々の使った TDC の仕様は、表 3.2の通りである。

名称	clock	Full Scale
KEK Type-1	50MHz	16bit 1310μs

表 3.2: TDC の仕様

方法

クロックジェネレータから出した信号を二つに分け、片方にゲートジェネレータでディレイをかけて二つの信号の間に時間差を作る。時間差を 0.5μs から 20μs まで変えていき、TDC のカウントとの関係を調べた。

ロジックを、図 3.4に示す。TDC が LAM(データを読み出せる状態になった事を示す信号。) を出さないので、ADC を使って LAM を見るようにしている。

結果

一つの TDC チャンネルの結果を図 3.5に示す。TDC カウントに、時間がほぼ比例していると言える。一次関数でフィットした結果、TDC カウント N と、時間 T(μs) の関係は、

$$T = 1.998 \times 10^{-2} \times N + 1.311 \times 10^{-2}$$

となった。我々の実験では、4 つの TDC チャンネルを使うが、ほかのチャンネルも同様の結果が得られた。

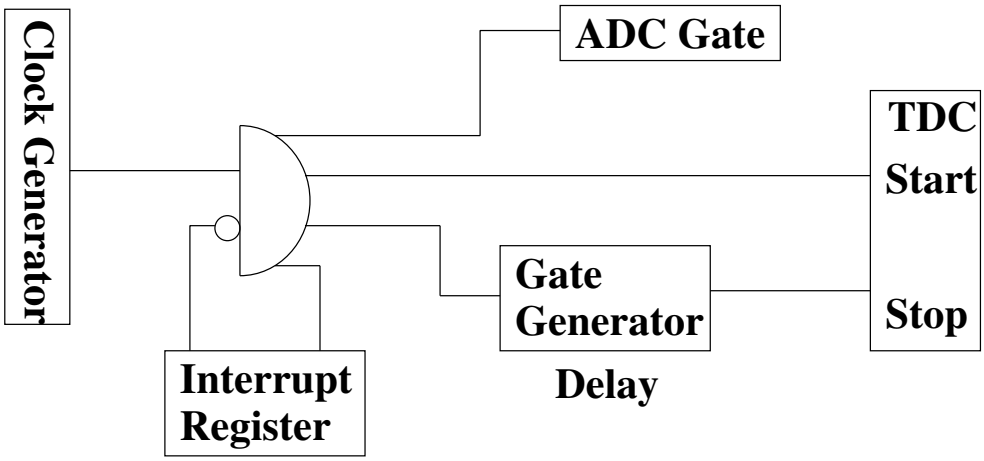


図 3.4: TDC の較正のロジック

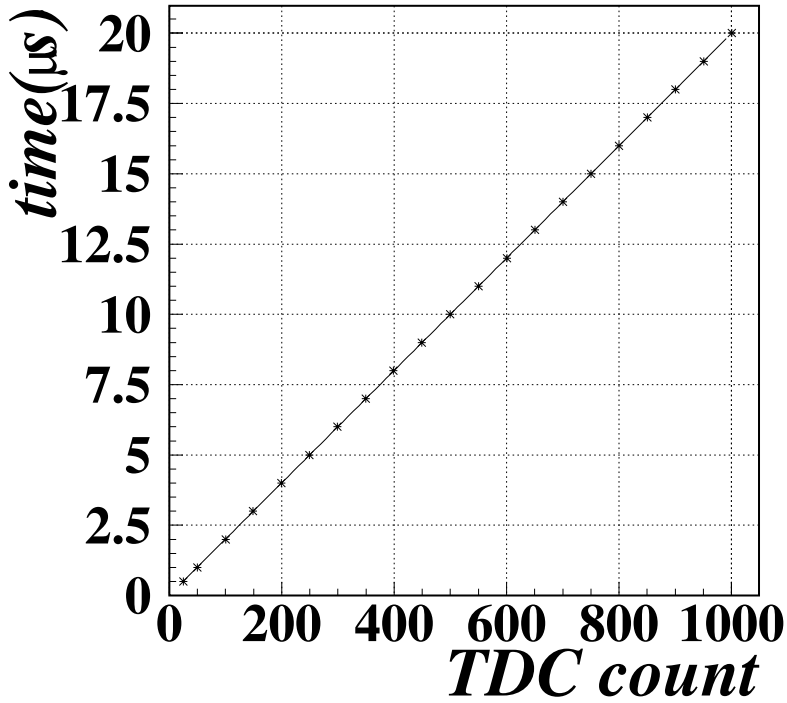


図 3.5: TDC カウントと時間の関係

3.3 ADC の較正

ADC は、πが各シンチレーターに落としたエネルギーを調べるのに用いる。ここでは、ADC 値 1 カウント当たりのエネルギーがいくらかを調べる。我々の使った ADC の仕様は、表 3.3の通りである。

名称	Gain	Full Scale
Le Croy 2249w	-0.25pC/count	~ -500pC($\simeq$ 1980 count)

表 3.3: ADC の仕様

方法

較正には宇宙線μを用いる。宇宙線μがシンチレーターに落とすエネルギーについては、過去の卒業論文 [5] で計算されており、ピーク値は 1.83MeV である。従って、ADC 値のヒストグラムのピーク値が 1.83MeV に対応しているとして、較正すればよい。

シンチレーション光はシンチレーター中で減衰するため、ADC 値は、シンチレーターを通過した場所により異なる。そこで、図 3.6のようにミニシンチレーターで挟み、全ての PMT のコインシデンスを取って通過位置を限定し、PMT に最も近い所、遠い所、中心の 3ヶ所について測った。

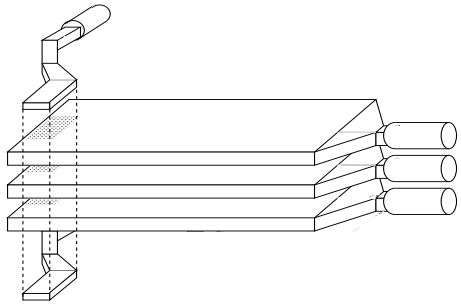


図 3.6: ADC の較正

結果

結果の一例を図 3.7にしめす。我々の実験では、通過位置は分からないので、この位置による違いはエネルギーを求める時の誤差となる。

図 2.1の A1 と T1 の 2 つの PMT について測り、それぞれの ADC 値 1 カウント当たりのエネルギーは、

$$A1: \quad 6.3^{+1.0}_{-3.1} \times 10^{-2} \quad MeV/count$$
$$T1: \quad 4.4^{+4.8}_{-3.5} \times 10^{-2} \quad MeV/count$$

となった。

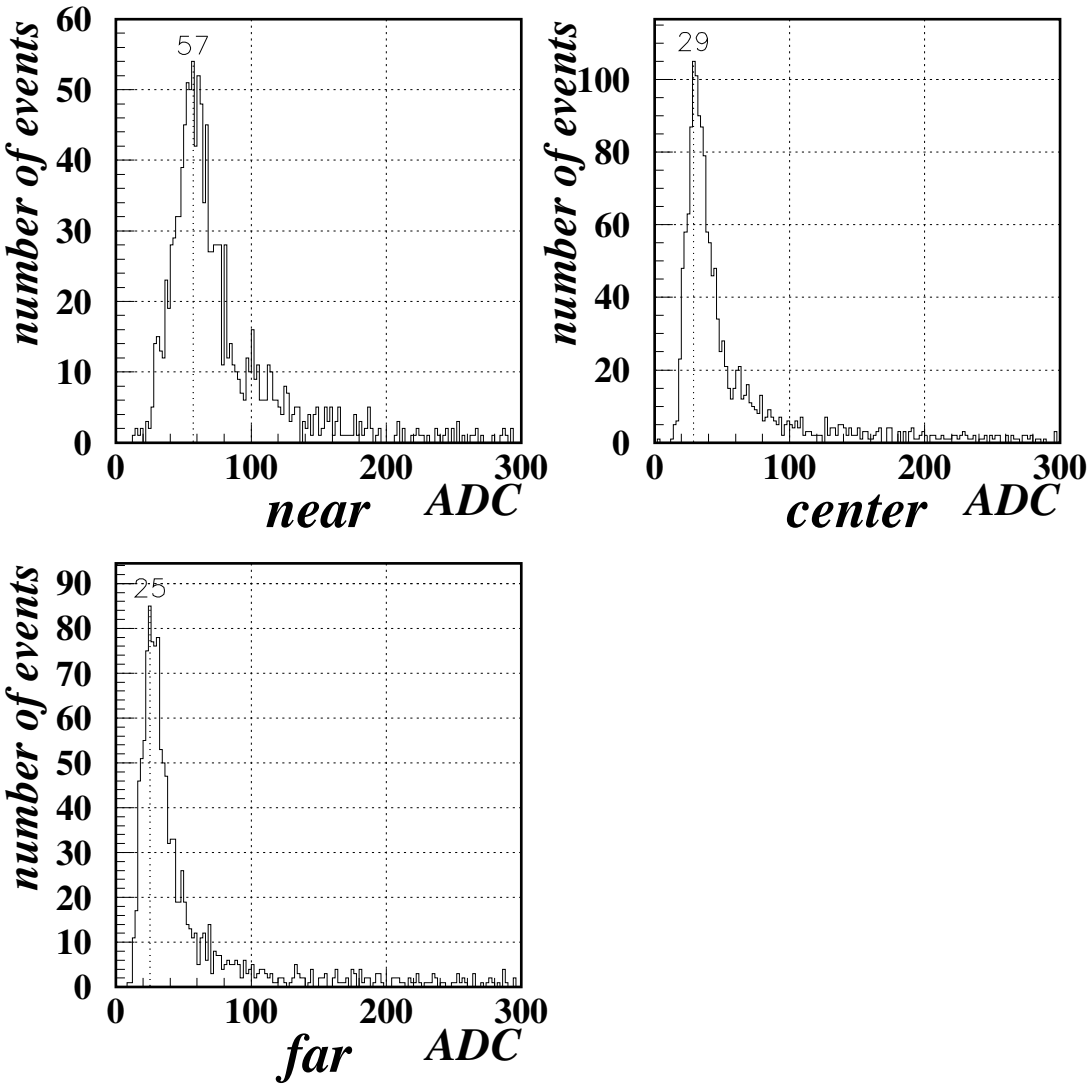

number of events
57
near ADC
number of events
29
center ADC
number of events
25
far ADC

図 3.7: ADC 値

3.4 ディジタルオシロスコープ

ディジタルオシロスコープは、$\pi^+ \longrightarrow \mu^+ \nu_\mu$ 崩壊のダブルパルスをとらえるために用いる。

このとき、静止したπ^+の崩壊により生成されるμ^+の運動エネルギーは、2 体崩壊なので一定の値にきまり、4.12MeV である。我々の実験では、このエネルギーをπ^+の証拠の一つとして用いる。

ディジタルオシロスコープで測った波形の電圧値を、ADC 値に変換する。そして、その ADC 値を宇宙線のつき抜けμの落とすエネルギーで較正し、4.12MeV のμの ADC 値がいくらになるかを求める。

我々の使ったディジタルオシロスコープの仕様は、表 3.4の通りである。

名称	サンプリングレート	データ長	垂直軸レンジ
HP54542A	$2 \times 10^9\ s^{-1}$	512	8div 0~32640

表 3.4: ディジタルオシロスコープの仕様

方法

まず、電圧値を ADC 値に変換する方法について述べる。

ADC の値は、流れた電荷を 0.25pC で割って得られる (表 3.3)。まず、波形の電圧値の和 Sをとる。データ点の間隔は、$1/$サンプリングレート $= 1/(2 \times 10^9) = 0.5ns$ であり、50Ωでカップリングされているので、流れた電荷 $Q(\mathrm{C})$ は、

$$Q(C) = \frac{S}{50} \times 0.5 \times 10^{-9}$$

である。従って、ADC 値 N_{ADC}は、

$$\begin{aligned} N_{ADC} &= \frac{Q}{0.25 \times 10^{-12}} \\ &= 40 \times S \end{aligned}$$

で計算される。

この式を用い、ミニシンチレーターを使って、ADC と同様に測った。

結果

結果を、図 3.8に示す。1.83MeV で

$$\text{ADC 値 } 26\sim122$$

と得られたので、4.12MeV では、

$$\text{ADC 値 } 58\sim274$$

となる。

$\pi^+ \longrightarrow \mu^+$の$\mu^+$のパルスは、ADC 値で、この範囲になければならない。

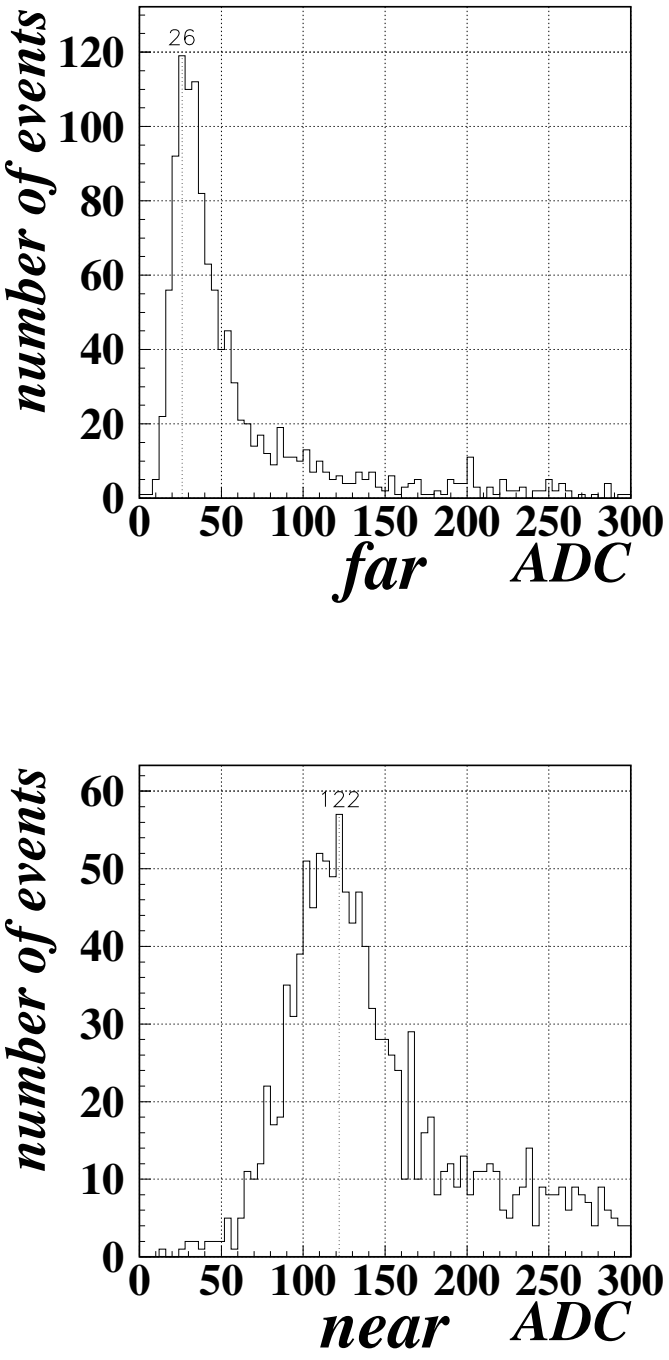


図 3.8: ディジタルオシロスコープの ADC 値

減衰距離

シンチレーション光は、シンチレーターの中で、距離 x 進むごとに $e^{-\frac{x}{\lambda}}$の割合で、減衰する。ここでは、モンテカルロシミュレーションの所で使うので、ディジタルオシロスコープで見るシンチレーターの減衰距離λ を求めておく。3 点の ADC 値と距離の関係をプロットし、$e^{-\frac{x}{\lambda}}$でフィットした (図 3.9)。その結果、$\lambda = 50\mathrm{cm}$ が得られた。

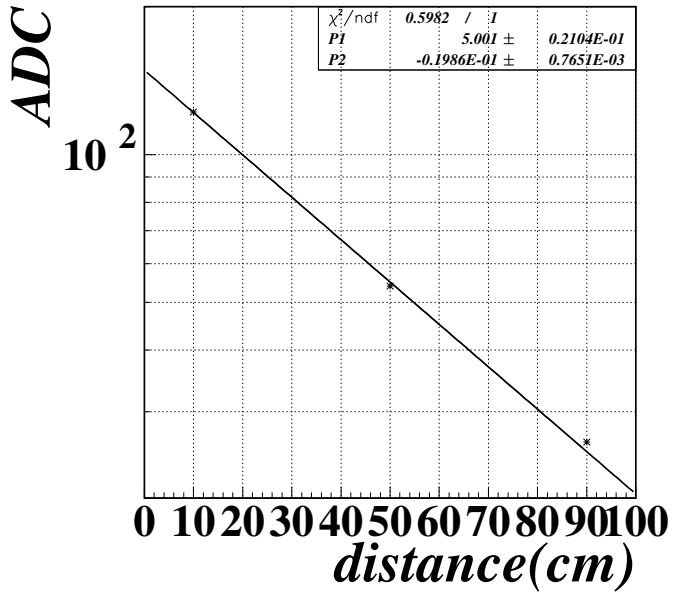


図 3.9: ADC 値の減衰

第 4 章

モンテカルロシミュレーション

4.1 目的

我々の検出器は、一日あたり幾つのπ^+を検出できるのか。また、実験室中のπ^+フラックスはどれほどであるのか。これを知る為には、実験結果に加えて幾つかの情報を、モンテカルロ法を用いて補う必要がある。

4.1.1 π^+の$\frac{dE}{dX}$の計算法

モンテカルロ法をπ^+に対して用いる際には、プラスティックシンチレーターに入射したπ^+が、その内部で単位長さあたりどれ程のエネルギーを落とすのか、すなわち$\frac{dE}{dX}$を知る必要がある。これは、Bethe-Bloch の式 [7] を用いて以下のように表される。

$$-\frac{dE}{dX} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A}\frac{z^2}{\beta^2}\left[\ln\left(\frac{2m_e\gamma^2 v^2 W_{max}}{I^2}\right) - 2\beta^2 - \delta - 2\frac{C}{Z}\right]$$

但し、$2\pi N_a r_e^2 m_e c^2 = 0.1535 MeV cm^2/g$
r_e : 古典電子半径 $= 2.817 \times 10^{-13} m$
m_e : 電子質量
N_a : $Avogadro$ 数 $= 6.022 \times 10^{-23} mol^{-1}$
I :平均励起ポテンシャル
A, Z, ρ :各々吸収物質の原子量、原子番号、密度
z : e を単位とした入射粒子の電荷
v : 入射粒子の速度
$\beta = \frac{v}{c}$、$\gamma = \frac{1}{\sqrt{1-\beta^2}}$

また、δ、Cは今回のエネルギー領域 $(\sim 50 MeV)$ では、それぞれ無視できる程小さい補正項である。
W_{max}は入射粒子の質量が電子質量 m_eに比べて非常に大きな時には、

$$W_{max} \simeq 2m_e c^2 \eta^2、但し \eta = \beta\gamma$$

として扱える。

我々が今回用いたプラスティックシンチレーター (NE110) は、$\frac{H}{C} = 1.10$ の原子比を持つ密度 $1.032 g/cm^3$の混合物である。この場合の$\frac{dE}{dX}$は、構成元素ごとに計算した$\frac{dE}{dX}$の和として近似的に求められる。
この際、各々の密度は原子比に比例するものとみなして計算した。

4.1.2 e^+の$\frac{dE}{dX}$の計算法

後で、陽電子の$\frac{dE}{dX}$も計算する必要が出てくるので、先にここで式を表出しておく事にする。我々の考えるべきエネルギー領域 ($\sim 50MeV$) の陽電子の$\frac{dE}{dX}$は、π^+に比べて制動放射の影響が無視できないほど大きくなる。従って、全体の損失は以下の式のように表す必要がある。

$$\left(\frac{dE}{dX}\right)_{\text{全体}} = \left(\frac{dE}{dX}\right)_{\text{衝突}} + \left(\frac{dE}{dX}\right)_{\text{放射}}$$

e^+の衝突損失

陽電子の場合、Bethe-Bloch の式は、以下のように表せる。

$$-\left(\frac{dE}{dX}\right)_{\text{衝突}} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A}\frac{1}{\beta^2}\left[\ln\frac{\tau^2(\tau+2)}{2(I/m_e c^2)^2} + F(\tau) - \delta - 2\frac{C}{Z}\right]$$

但し、τは $m_e c^2$を単位とした入射粒子の運動エネルギー。なお、ここで説明していない変数の意味は、前節に準ずるとする。また、関数 $F(\tau)$ は電子と陽電子で異なっており、陽電子の場合以下のようになる。

$$F(\tau) = 2\ln 2 - \frac{\beta^2}{12}\left(23 + \frac{14}{\tau+2} + \frac{10}{(\tau+2)^2} + \frac{4}{(\tau+2)^3}\right)$$

e^+の放射損失

制動放射による損失では、以下の遮蔽効果パラメーター ξが重要になる。

$$\xi = \frac{100 m_e c^2 h\nu}{E_0 E Z^{1/3}}$$

但し、E_0 : 入射陽電子の全エネルギー
E : 終状態の陽電子の全エネルギー
$h\nu$: 放出光子のエネルギー, $(E_0 - E)$

また、我々のエネルギー領域では制動放射の微分断面積は以下の式で表せる。

$$d\sigma = 4Z^2 r_e^2 \alpha \frac{d\nu}{\nu}\left\{(1+\varepsilon^2)\left[\frac{\phi_1(\xi)}{4} - \frac{1}{3}\ln Z - f(Z)\right] - \frac{2}{3}\varepsilon\left[\frac{\phi_2(\xi)}{4} - \frac{1}{3}\ln Z - f(Z)\right]\right\}$$

但し、ε：$\frac{E}{E_0}$、α：$\frac{1}{137}$

$$\begin{aligned}
\phi_1(\xi) &= 20.863 - 2\ln[1+(0.55846\xi)^2] - 4[1 - 0.6\exp(-0.9\xi) - 0.4\exp(-1.5\xi)] \\
\phi_2(\xi) &= phi_1(\xi) - \frac{2}{3}(1 + 6.5\xi + 6\xi^2)^{-1} \\
f(Z) &\simeq a^2[(1+a^2)^{-1} + 0.20206 - 0.0369a^2 + 0.0083a^4 - 0.002a^6]
\end{aligned}$$

但し、$a = \frac{Z}{137}$

以上の式から、制動放射による放射損失は以下のように書ける。

$$-\left(\frac{dE}{dX}\right)_{\text{放射}} = N\int_0^{\nu_0} h\nu \frac{d\sigma}{d\nu}(E_0,\nu)d\nu$$

但し、N：原子数$/cm^3, N = \rho\frac{N_a}{A}; \nu_0 = \frac{E_0}{h}$

4.2 手順

シミュレーションは、以下の手順に乗っとって行なわれた。

1. アクセプタンスの計算　我々の見たいπ^+は、天井で生成して検出器に降り注いでいるものと考えられる。従って、そのようなπ^+に対する我々の検出器のアクセプタンスを計算した。
2. 検出効率の計算　我々が見る事のできるのは、プラスティックシンチレーター中で静止したπ^+がμ^+、そしてe^+に崩壊したものである。従って、そのようなe^+に対する我々の検出器の検出効率を計算した。
3. 立体角の計算　π^+の期待値を U.Camerini の実験と比較して計算する為に、これに必要な検出器の幾何学的な立体角を計算した。

4.3 シミュレーション

4.3.1 アクセプタンスの計算

アクセプタンス Acc は以下のように定義した。

$$\text{Acc} = \frac{(\text{検出イベント数})}{(\text{発生イベント数})}$$

仮定した条件は、表 4.1の通り。

エネルギー分布	一様 $(35\pm15\mathrm{MeV})$
方位角分布	一様
極角分布	$\cos^2(2\theta)$
位置分布	一様 $(17.6\mathrm{m}\times5.80\mathrm{m})$

表 4.1: アクセプタンスの計算に用いた仮定

このような仮定の下に発生させたπ^+を、実際のジオメトリー (図 2.1) を模した検出器で検出する。すなわち、検出器に飛び込んできたπ^+は、トリガーカウンターを取り巻くアンタイカウンターやコインシデンスカウンター内でエネルギーを落とし、適当な閾値以下になればそこで静止するものとする。なお、飛行中に崩壊したπ^+は全て除いている。
このようなπ^+の内、トリガー条件を満たしたものだけが、検出イベントとみなされ、カウントされる。

このような仮定のもとにシミュレーションを行なう。具体的な流れは次のようになる (図4.1参照)。

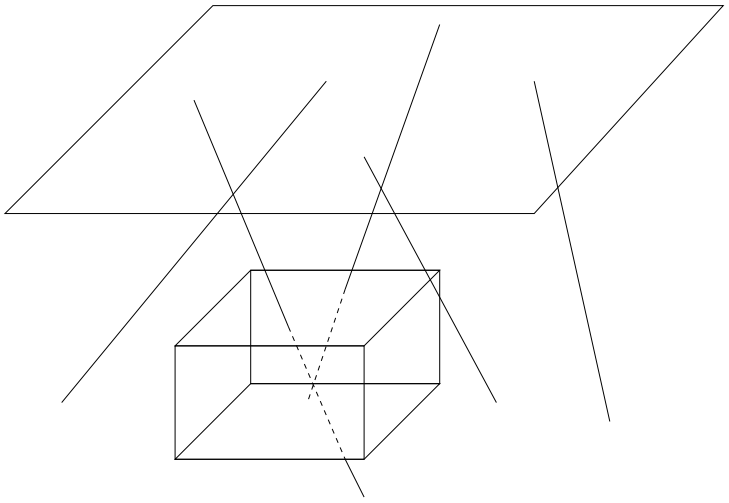


図 4.1: アクセプタンス計算の概念図

- 一様乱数を使って天井 (z = 天井の高さ) におけるπ^+の発生点 (x、y)、極角θ、方位角ϕ及び初期運動エネルギーを決定する。
- 運動エネルギーより直ちに速度が計算される。
- 微小時間 $\triangle t$ 毎に粒子を進める。
- もし e^+がプラスティックシンチレーターの内部を通過していたら、この間に落とすエネルギーを Bethe-Bloch の式より計算する。

- また、この間に崩壊が起こったかどうかも乱数によって決定する。
- もしエネルギーが適当な閾値以下に落ちていれば、そこで静止したとみなす。*
- これを繰り返し行ない、トリガー条件を満たしてトリガーカウンター内で静止したものの数を数える。

結果

最終的に 10^7個のπ^+を発生させ、3880 個を検出した。
従って、立体角をも含んだ我々の検出器のアクセプタンス Acc は、

$$\mathrm{Acc} = 3.88 \times 10^{-4}$$

となる。

4.3.2 検出効率の計算

トリガー条件を満たしたπ^+がトリガーカウンター内で 2 体崩壊する事により、μ^+が発生する。このμ^+の運動エネルギーは単色性で 4.12MeV (10ページ参照) と計算されており、プラスティックシンチレーター中における飛程はたかだか 2mm である。従って、μ^+の検出効率は 100%とみなしても差し支えない。結局、検出効率としては、 μ^+-e^+崩壊によって放出される陽電子の検出効率のみを考えれば良い事になる。

検出効率 Eff は以下のように定義した。

$$\mathrm{Eff} = \frac{(\text{検出イベント数})}{(\text{発生イベント数})}$$

仮定した条件は、表 4.2の通り。
これは、入射してきたπ^+がランダムな位置に静止する事で、μ^+の運動量の方向分布、位置分布、

エネルギー分布	Michel スペクトル
方位角分布	一様
極角分布	一様
位置分布	一様 (1.00m×0.33m×0.01m)

表 4.2: 検出効率の計算に用いた仮定

偏極方向がランダムに決定され、結果的に e^+の運動量の方向分布がランダムになるという理由による。
また、Michel スペクトルというのは、μ^+-e^+崩壊が本質的に 3 体崩壊であるという所に起因する

*Bethe-Bloch の式は充分低いエネルギー領域では不正確になる為、運動エネルギーが 0 になるまで追っていっても正しい値を返さない。

もので、μ^+-e^+崩壊におけるe^+のエネルギー分布を表している。その表式は、以下の様に表される。

$$\frac{d\Gamma}{dE'} = \frac{G^2}{12\pi^3} m^2 E'^2 \left(3 - \frac{4E'}{m}\right)$$

但し、Γ : μ^+の崩壊率
m : μ^+の質量
E' : e^+のエネルギー
G : Fermi 定数 (弱い相互作用の結合定数)

このような仮定の下に発生させたe^+を、実際のジオメトリーを模した検出器で検出する。すなわち、トリガーカウンター内で発生したe^+は、これを取り巻く 4 枚のカウンターのいずれかの中でエネルギーを落とし、パルスを発生させる。このようなe^+の内、ディスクリミネーターに於てスレッショルド $(-20mV)$ を越える大きさのパルスを発生させられる様なe^+だけが、検出イベントとみなされ、カウントされる。なお、プラスティックシンチレーター中におけるパルスハイトの扱いについては、後述する。

このような仮定のもとにシミュレーションを行なう。具体的な流れは次のようになる (図 4.2参照)。

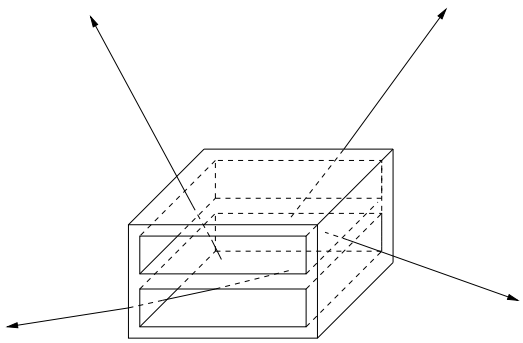


図 4.2: 検出効率計算の概念図

- 一様乱数を使ってトリガーカウンター中におけるe^+の発生点 $(x$、y、$z)$、極角θ、方位角ϕを決定する。
- 運動エネルギーより直ちに速度が計算される。
- 微小時間 $\triangle t$ 毎に粒子を進める。
- もしe^+がプラスティックシンチレーターの内部を通過していたら、この間に落とすエネルギーを Bethe-Bloch の式より計算する。

- これを繰り返し行ない、e^+の落としたエネルギーの総和を計算する。
- 後述するパルスハイトの換算式より、パルスハイトを求めスレッショルドと比較する。
- 条件を満たすイベントを e^+として数える。

結果

10^5個のμ^+-e^+イベントを発生させ、65459 個を検出した。
従って、μ^+-e^+イベントに対する我々の検出器の検出効率は、

$$\mathrm{Eff} = 65.5\%$$

となる。

以上の結果より、我々の検出器のアクセプタンスを含んだπ^+-μ^+-e^+イベントに対する検出効率 $\mathrm{Eff_{tot}}$は、

$$\mathrm{Eff_{tot}} = 3.88 \times 10^{-4} \times 0.655 = 2.54 \times 10^{-4}$$

となった。

パルスハイトの較正

荷電粒子がプラスティックシンチレーターを通過すると、その落としたエネルギーに比例した大きさのパルス (シンチレーション光) が発生する。この波形は下記に示す関数 4.1で表される事が分かっている [7]。

$$V(t) = a\left[\exp\left(-\frac{t-b}{c}\right) - \exp\left(-\frac{t-b}{d}\right)\right] \tag{4.1}$$

(但し t は時間、a, b, c, d は定数)

シンチレーション光はプラスティックシンチレーター中で指数的に減衰する為、落としたエネルギーおよび PMT からの距離によってはパルスハイトがディスクリミネーターのスレッショルド以下に下がってしまい、検出されない事も起こり得る。従って、シミュレーションにおいてはこの事を考慮する必要がある。
本来ならば、パルスハイトのエネルギー依存性及び位置依存性について調べたデータがあれば良いのだが、あいにく本実験では測定されていなかった。従って、この欠けたデータをモンテカルロ法によって補完する事にした。
幸い本実験では、ディジタルオシロスコープを用いて多数のμ-e 崩壊イベントを検出している。これを e のエネルギーを横軸としたヒストグラムにしたものに対して、式 4.1を用いて生成した疑似パルスによって作られたヒストグラムをフィッティングする事で、式 4.1の未知定数を決定する。さらに、今必要となるのはパルスハイトのみであるから、式 4.1からそのような関数を得る事にする。また、パルスの減衰については、実験結果より得られた減衰長を用いた。(3.4 節 25ページ参照の事。)
結果として、落としたエネルギー $E(MeV)$、PMT からの距離 $L(cm)$ で発生した e に対するパル

スハイト $H(V)$ は以下の様になった。

$$H = -0.35 \times \sqrt{\frac{E}{1.83}} \times \exp\left(-\frac{L}{50}\right)$$

4.3.3 立体角の計算

立体角はπ^+フラックスの計算には必要ないが、π^+の期待値の計算に必要となるのでここで求めておく。
検出器の形は複雑であるので、立体角を直接解析的に求める事は一般に非常に困難である。従って、以下に示す近似的な計算によってこれを求める事にした。
まず検出器を内包する、充分大きな半径 r_0の球体を考える。この空間における『完全な検出器』というものを考えてやるとすれば、それは半径 r_0 の円盤であり、その立体角$\Omega_{\text{完全}} = 4\pi$、有効面積 $S_{\text{完全}} = \pi r_0^2$であると考えられる。一方で、我々の検出器は有効面積 $S_{\text{我々}} = 1(m) \times 0.33(m)$ を持っている (横方向の面積については、トリガー条件を満たさないから考えなくとも良い)。
今 N本の粒子がこの空間上に降ると考えると、『完全な検出器』はその全てを捕らえられるが、我々の検出器はその 1 部、$N_{\text{検出}}$本しか捕らえられないであろう。この場合、これは立体角及び有効面積の違いに起因すると考えられるから、以下の式を立てる事ができる。

$$\frac{N_{\text{検出}}}{N} = \frac{S_{\text{我々}}}{S_{\text{完全}}} \frac{\Omega_{\text{我々}}}{\Omega_{\text{完全}}}$$

ところで左辺は粒子を検出する確率そのものであるから、モンテカルロ法によって計算する事が可能である。今これを Pで置き換え、既知の値を代入してやると、上式は次のように変形できる。

$$\Omega_{\text{我々}} = \frac{4\pi^2 r_0^2}{1 \times 0.33} P$$

従って、立体角を求める為には、検出の確率 Pを求めればよい事になる。実際のシミュレーションは、以下の様に行なわれた (図 4.3参照)。

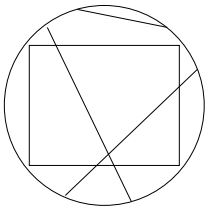


図 4.3: 立体角計算の概念図

- まず、先ほどの半径 r_0の球面上に任意の 2 点をとり、これを線分で結ぶ。
- この線分が、トリガー条件を満たして検出器を貫いていれば検出イベントとみなし、そうでなければ棄却する。
- これを繰り返して精度良く検出の確率 P を決定する。

なお、球面上に点を打つ際には、面密度が一定になる様な乱数分布を取った。

結果

全部で 10^7イベントを発生させ、そのうち 182801 イベントを検出した。従って、検出の確率 $P = 1.83 \times 10^{-2}$となる。しかし、我々が今回見ているのは下向きのπのみである事から、これに $\frac{1}{2}$をかける必要がある。

これらを考慮した結果、我々の検出器の立体角として、

$$\Omega_{\text{我々}} = 1.57Sr$$

を得た。

第 5 章

実験の解析

5.1　データ解析の方法

本実験は、1998 年 1 月 15 日 (木) から 2 月 16 日 (月) までの間断続的に行ない、測定時間は合計 2268966 秒 (26 日 6 時間 16 分 6 秒) であった。ここから dead time 324969 秒を差し引いた時間 1943997 秒 (22 日 11 時間 59 分 57 秒) が data taking time となる。

この間に 20505 イベントが得られた。

5.2　イベント選別

得られた 20505 イベントはＡ１及びＴ１、Ｔ２、Ｔ３、Ｔ４のコインシデンスとＡ２、Ａ３、Ｃ１のアンタイコインシデンスによって得られたイベントである。πはμに崩壊し、さらにμ-e 崩壊をおこして電子を放出する。従ってＴ３、Ｔ４の周りの板Ｃ１、Ａ１、Ａ２、Ａ３のうち１つだけ TDC が電子を検出したイベントをμ イベントとして取り出した。このチェックによって 15981 イベントを得ることができた。

次にＴ３からつながっているデジタルオシロスコープの波形を見て、π-μ崩壊のイベントを eye-scan を行なって取り出した。選別は以下のような条件で行い、それらの中でダブルパルスが見られるものをπイベントとした。

- 2 パルス間の幅が 130ns 以内。
  πの寿命の期待値は 26ns なので、これは 5 life time 分に相当する。この範囲で 99.3 %のπが崩壊するので、ほぼすべてのπ崩壊を捉えることができる。

- 2 つ目のパルスの高さが -75mV 以下。
  4.12MeV のμは最低 58 の ADC の値をもつので、パルスが 20ns 程度の矩形波とした場合 72mV の高さをもつ。従って 75mV 程度のしきい値を入れた。

- 崩壊電子による TDC の値が 1000ch(20μs) 以内。
  μの寿命の期待値は約 2.0μs なので、これは 10life time 分に相当する。この範囲ではほぼ 100 %のμが崩壊する。したがってμ崩壊はすべて捉えることができる。

こうして最終的に 13 イベントを取り出す事ができた (図 5.1 -5.7)。これらを表 5.1、表 5.2にまとめた。

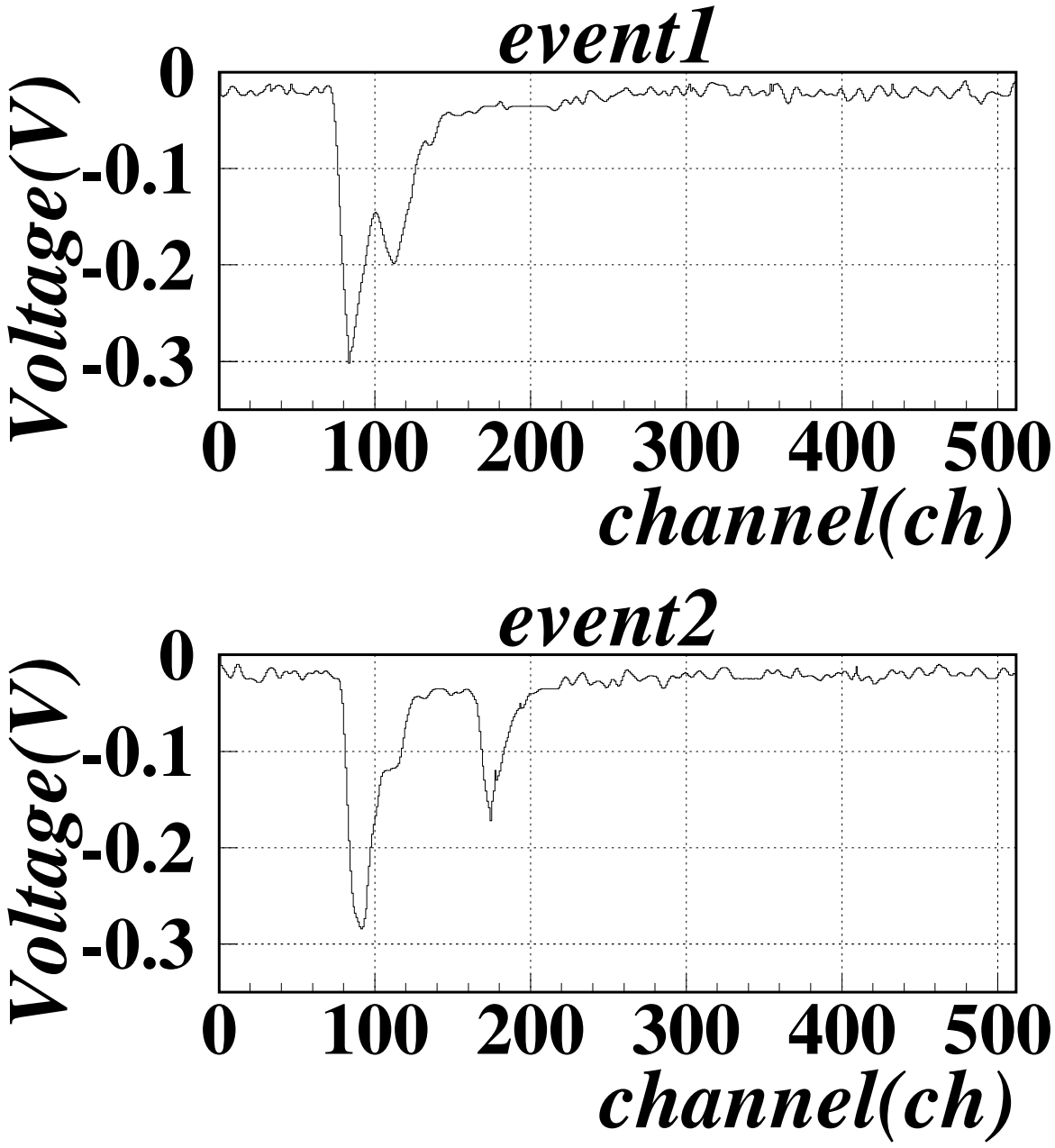

event1
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event2
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.1: πイベント

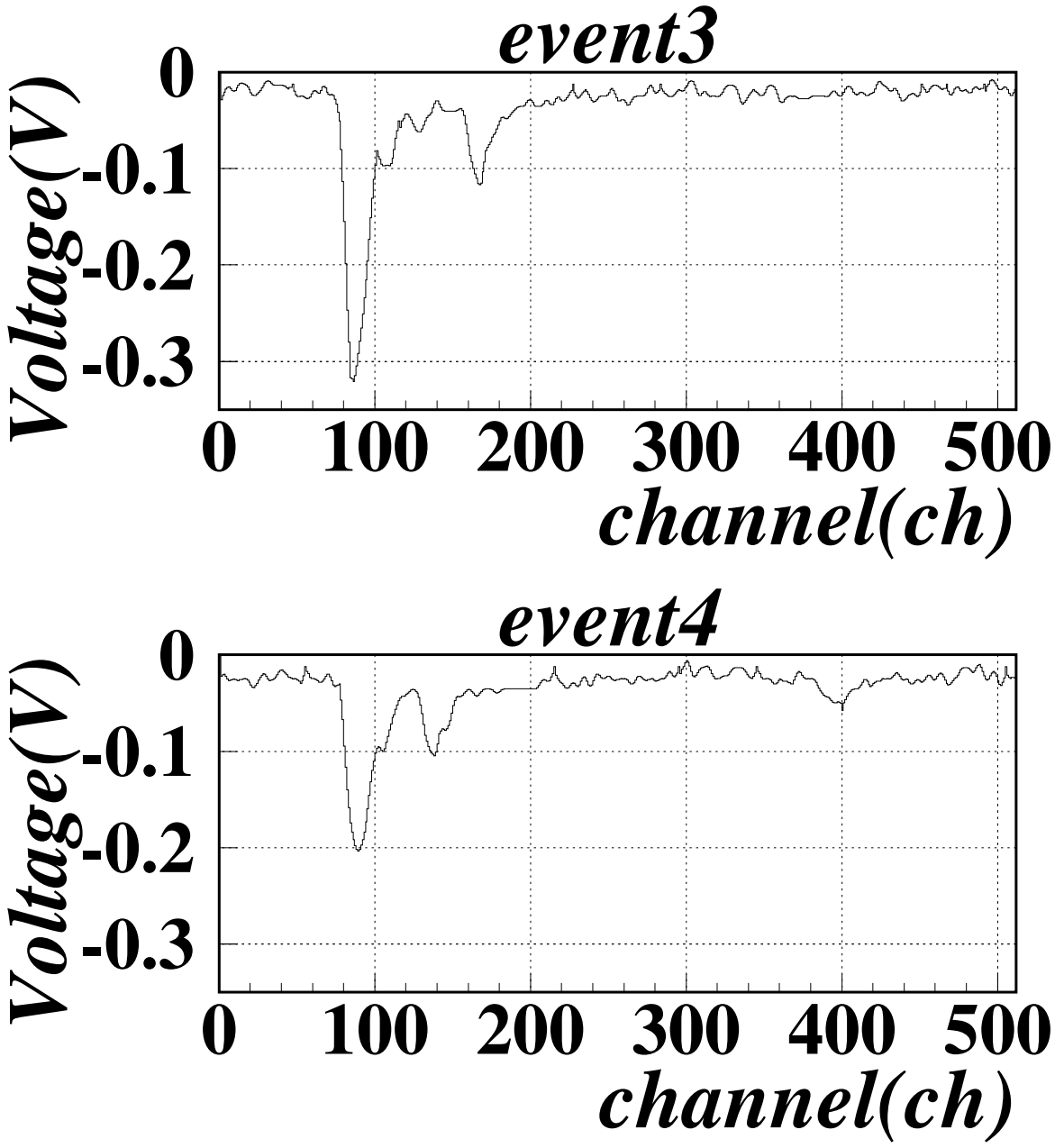

event3
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event4
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.2: πイベント

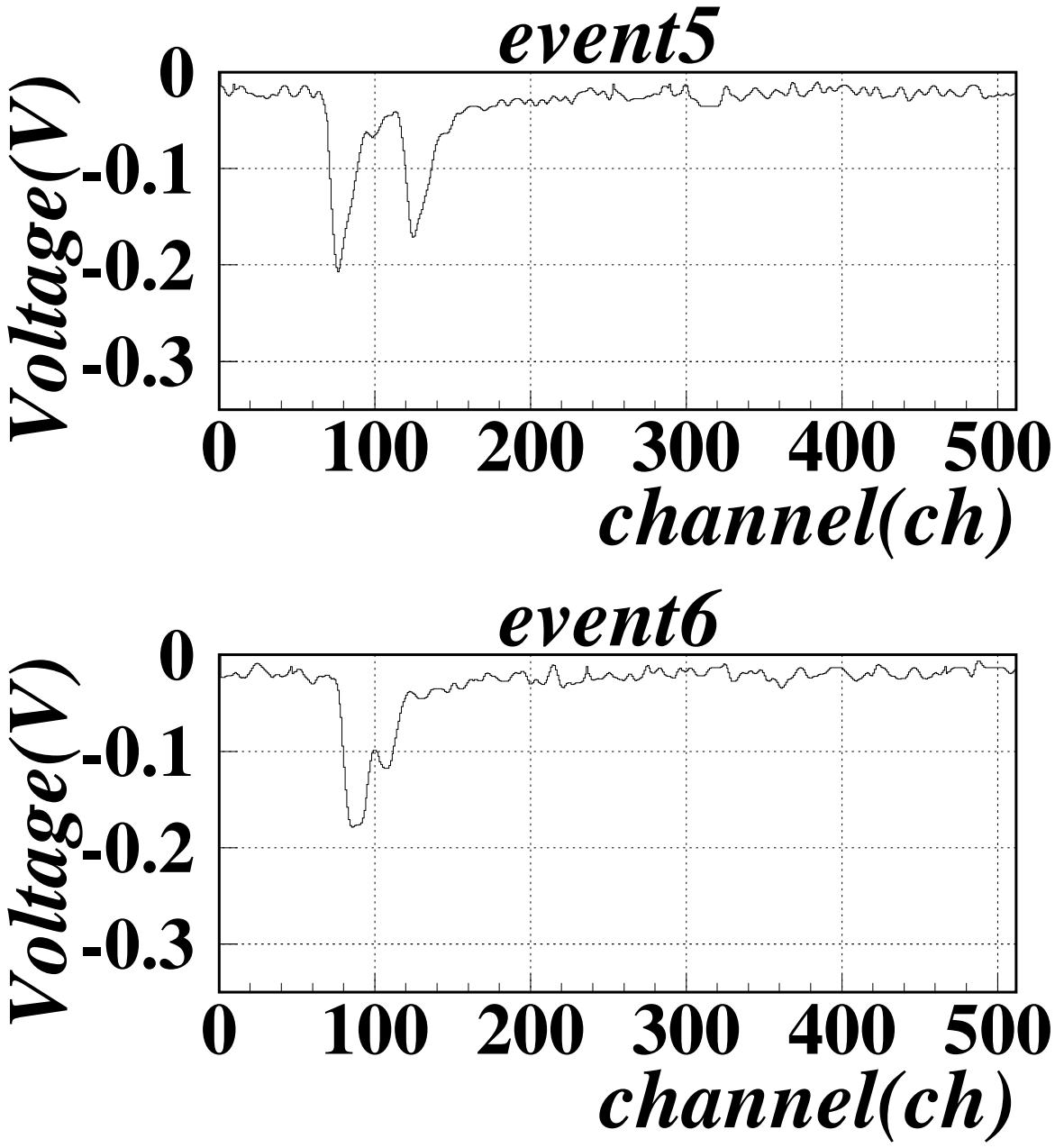

event5
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event6
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.3: πイベント

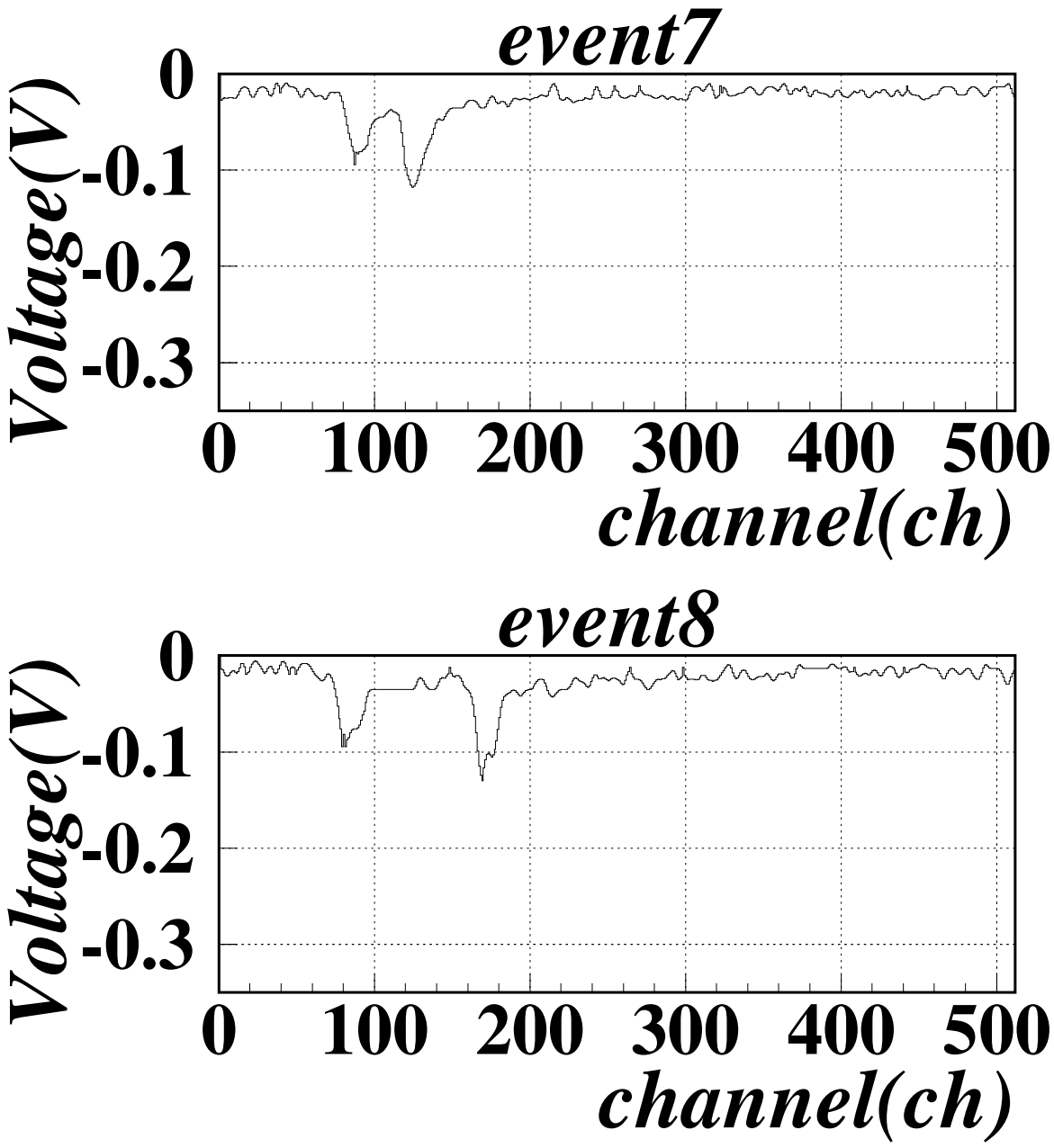

event7
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event8
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.4: πイベント

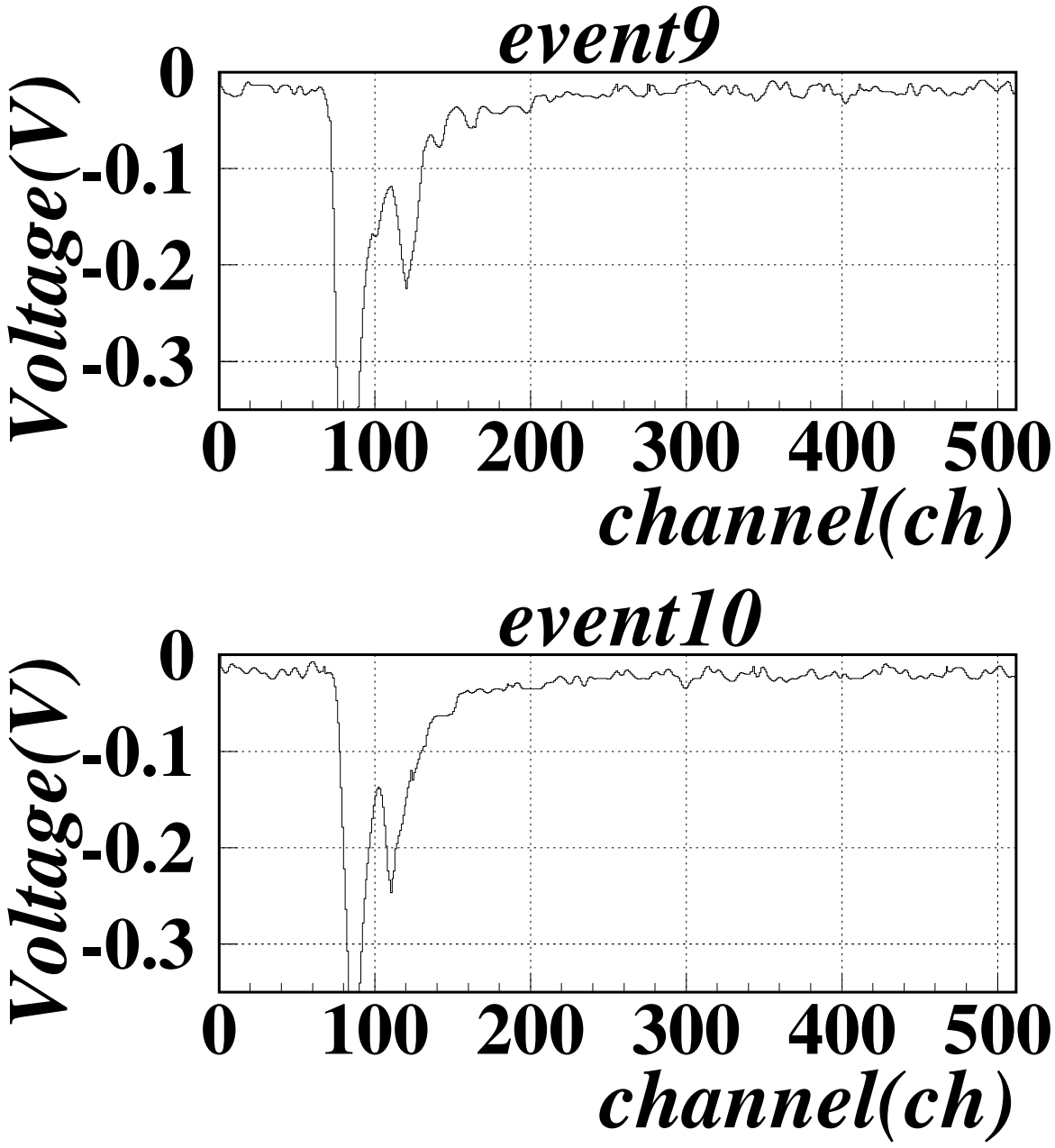

event9
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event10
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.5: πイベント

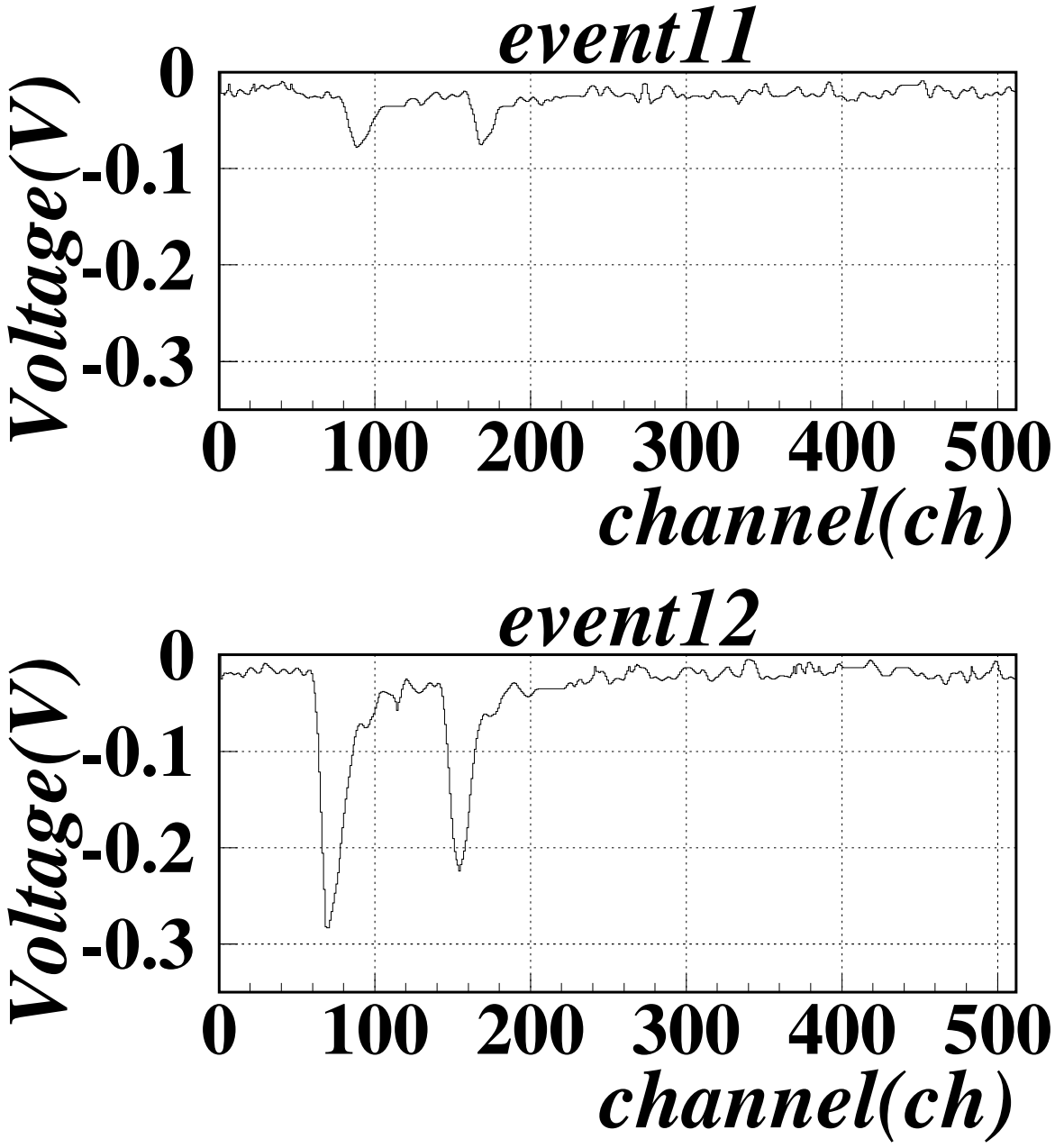

event11
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)
event12
Voltage(V)
0
-0.1
-0.2
-0.3
0
100
200
300
400
500
channel(ch)

図 5.6: πイベント

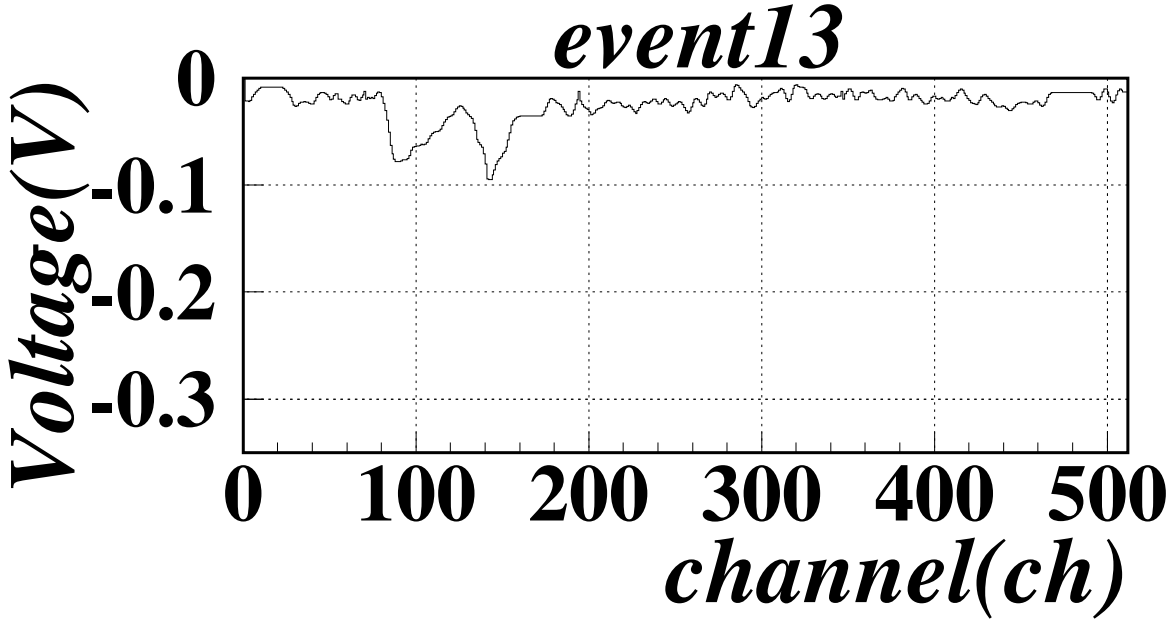


図 5.7: πイベント

live time	1943992 sec.
trigger event	1332828 event
trigger rate	0.686 Hz

表 5.1: トリガーイベント

全 event	20505 event
μ event	15981 event
π event	13 event

表 5.2: πのイベント選別

5.3 バックグラウンドの評価

トリプルパルスのバックグラウンドを評価する。π イベントに対するバックグラウンドとしては、
条件Ａ：Ａ１及びＴ１、Ｔ２を通り、Ｔ３、Ｔ４で止まる (first pulse)。
条件Ｂ：trigger 後、130ns 以内に second pulse が発生する。
条件Ｃ：20μs 以内に third pulse がある。
の条件を満たさなければならない。これを満たすバックグラウンドは 4 つあり、各々以下のように計算される。

1. stop μ + 崩壊 e(130ns 以内) + accidental(20μs 以内)
   条件Ａをみたす stop μは 15981 イベントあり、そのμが 130ns 以内に崩壊する割合は、1-exp(-$\frac{0.13}{2.09}$) となる。さらに 20μs の時間幅にＴ１、Ｔ２の周り、つまりＡ１、Ａ２、Ａ３、Ｃ１のどれかが鳴るイベント数は、(1.40+0.91+0.57+1.85)×20μs であるから、1. を満たすイベント数は、

$$15981 \times (1 - exp(-\frac{0.13}{2.09})) \times (4.73 \times 20 \times 10^{-6}) \simeq 9.1 \times 10^{-2} event$$

2. stop μ + accidental(130ns 以内) + 崩壊 e(20μs 以内)
   stop μは 15981 イベント、それが 130ns から 20μs の間に崩壊する割合は、exp(-$\frac{0.13}{2.09}$)-exp(-$\frac{20}{2.09}$) である。そしてＴ１∩Ｔ２のレートが 312.2Hz なので、130ns 幅に鳴るイベント数は 312.2×130ns となる。以上より 2. を満たすイベント数は、

$$15981 \times (exp(\frac{0.13}{2.09}) - exp(\frac{20}{2.09})) \times (312.2 \times 130 \times 10^{-9}) \simeq 0.61 event$$

3. accidental + μ-e 崩壊
   accidental なトリガーイベント数が 1332828 イベント。130ns 幅に stop μイベントがくる数は 15981×$\frac{130ns}{1943997sec}$×130ns となって、20μs 以内にほぼ 100 %崩壊するので、3. を満たすイベント数は、

$$1332828 \times (15981 \times \frac{130 \times 10^{-9}}{1943997}) \times (130 \times 10^{-9}) \simeq 1.4 \times 10^{-3} event$$

4. accidental × 3
   accidental トリガーが 1332828 イベント。130ns 幅の accidental トリガーイベント数が 0.686×130ns。20μs 幅の accidental トリガーイベント数が 4.73×20μs となる。従って 4. を満たすイベント数は、

$$1332828 \times (0.686 \times 130 \times 10^{-9}) \times (4.73 \times 20 \times 10^{-6}) \simeq 1.1 \times 10^{-5} event$$

したがって実験期間中に予想しうるバックグラウンドは、計 0.70 となる。これはそれほど実験結果には影響ないと思われる。

5.4 イベントの解析

5.4.1 stop μ イベントに関する μの寿命

得られた 15981 の stop μイベントはμ-e 崩壊を行うので、周りの板 (A 1、A 2、A 3及びC 1) の TDC からμの崩壊時間 (すなわちμの寿命) を知ることができる。(図 5.8)

こうして得られたμの寿命τ_μは、

$$\tau_\mu = 2.09 \pm 0.02(\mu s)$$

となった。実際の観測においては、μ^-の吸収によって、見掛けの寿命が短くなる。したがって実験結果は期待値と矛盾しないので、検出器は正しく作動していることが確認できた。

5.4.2 π-μ イベントでのπ及びμの寿命

π イベントと思われるイベント数が 13 と少ないので、少数統計に対して有効な最尤法 (maximam-likelihood method) を使った [8]。

likelihood 関数 $\mathcal{L}(\tau)$ として、

$$\mathcal{L}(\tau) = \prod_{i=1}^{13} \mathcal{P}_i(\tau)$$

$$\mathcal{P}_i(\tau) = \frac{\tau^{-1} exp(\frac{t_i}{\tau})}{exp(\frac{t_{min}}{\tau}) - exp(\frac{t_{max}}{\tau})}$$

と定義する。但し t_iは各イベントでの観測された時間である。また、t_{min}及び t_{max}は実験の測定範囲で、今回の実験では最大値として $t_{max} = 130$ns、最小値には得られたπイベントの最小値 $t_{min} = 12$ns を用いた。

最尤法では関数 $\mathcal{L}(\tau)$ が最大となるときのτの値が最も適当な値であると推定される。したがって微分した値が 0 に等しいとする方程式、

$$\frac{d}{d\tau} log\mathcal{L}(\tau) = \sum_{i=1}^{13} log\frac{d\mathcal{P}_i(\tau)}{d\tau} = 0$$

を解けばよい。対数をとってあるのは、各イベントの積を和に直すことによって、微小化を押さえる事ができるためである。こうして得られたπの寿命τ_πは、

$$\tau_\pi = 20.60^{+8.46}_{-5.27}(ns)$$

となった。図 5.9に likelihood 関数 $\mathcal{L}(\tau)$ を示す。ここでは最大値を 1 に規格化してある。

また、$\pi \to \mu \to$e と崩壊するので、このイベントでのμの寿命も求めることができ、13 イベントでのμの寿命$\tau_{\pi\mu e}$を同様に最尤法によって求めると、

$$\tau_{\pi\mu e} = 2.02^{+0.70}_{-0.47}(\mu s)$$

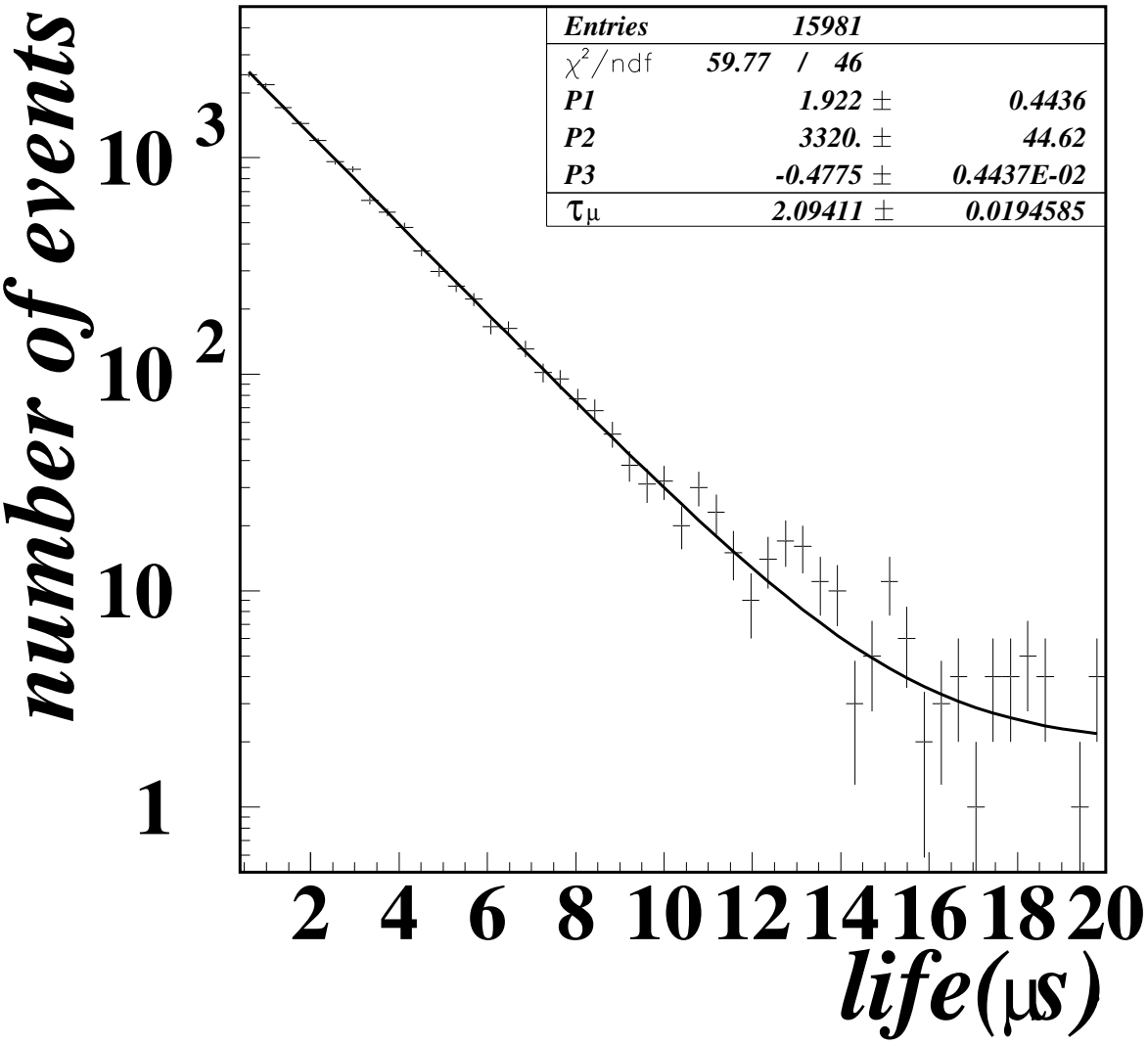

Entries 15981
χ²/ndf 59.77 / 46
P1 1.922 ± 0.4436
P2 3320. ± 44.62
P3 -0.4775 ± 0.4437E-02
τμ 2.09411 ± 0.0194585
number of events
10 3
10 2
10
1
2 4 6 8 10 12 14 16 18 20
life(μs)

図 5.8: μの寿命

となった。図 5.10に$\pi\mu$e イベントの likelihood 関数を示す。

これらは共に期待値に矛盾しないので、我々が選んだイベントはπイベントあると言える (図 5.11、5.12)。

5.5 結果

5.5.1 πのイベント数の見積もり

likelihood の最小値として、得られたπ イベントの最小値 12ns を適用したため、図 5.11で見られるように短寿命側に明らかな数え落としが見られる。従ってこの数え落とし、及び 130ns 以上の長寿命側を考慮に入れると、$23.44^{+5.20}_{-3.66}$ イベントが見積もられる。つまり数え落としは 10 イベントほどであった。

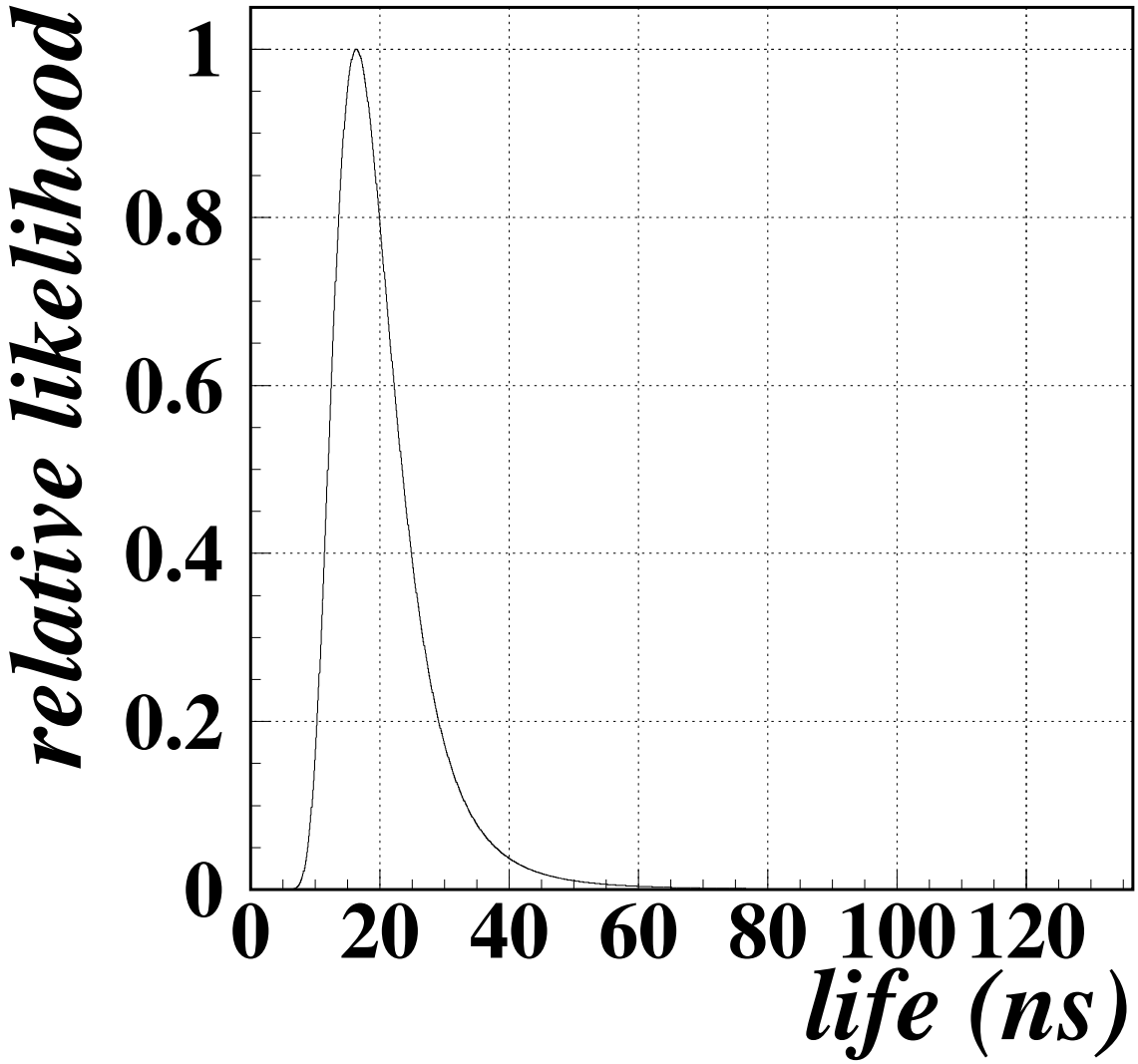


図 5.9: π イベントでの likelihood 関数

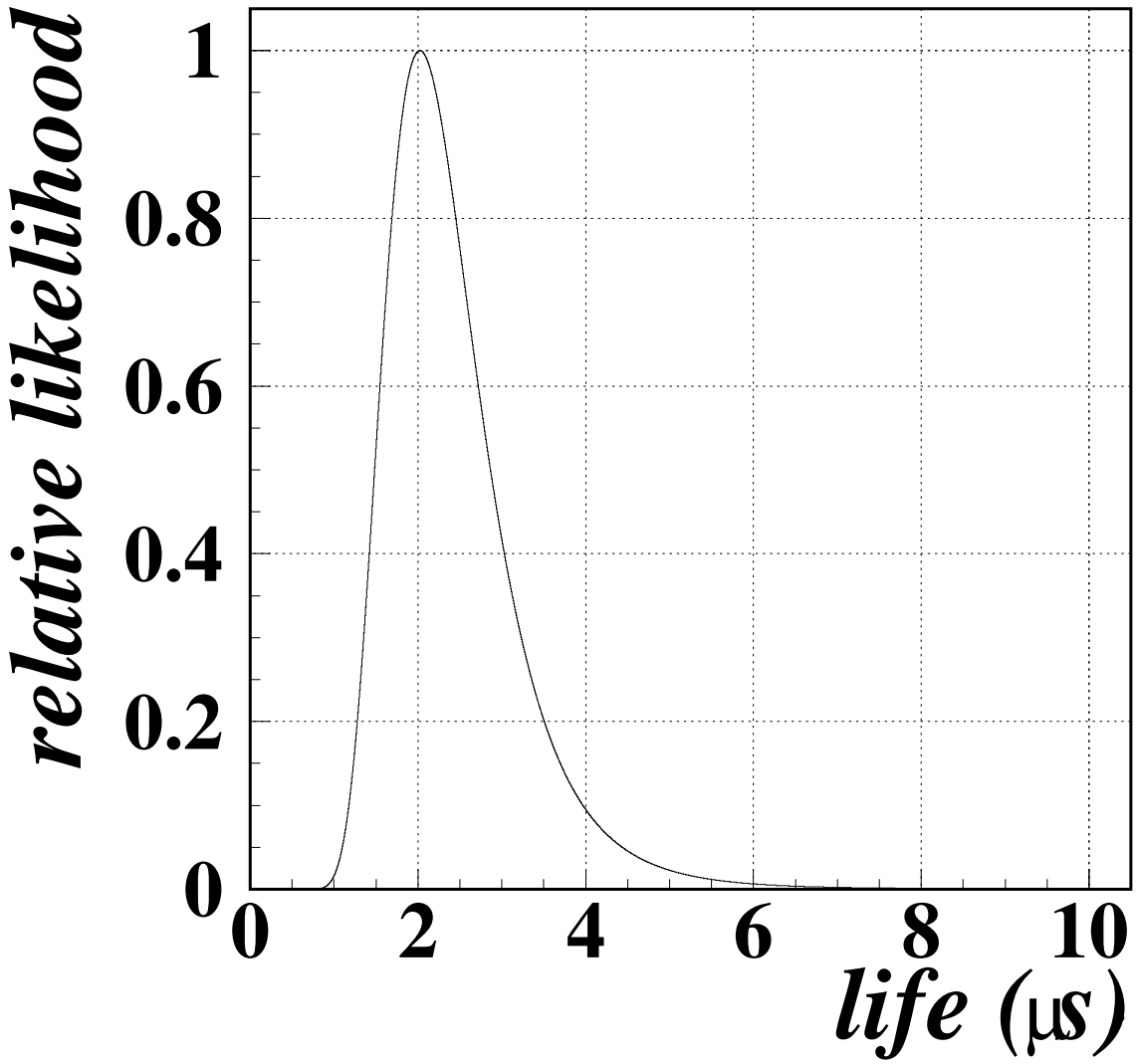


図 5.10: $\pi\mu$e イベントでの likelihood 関数

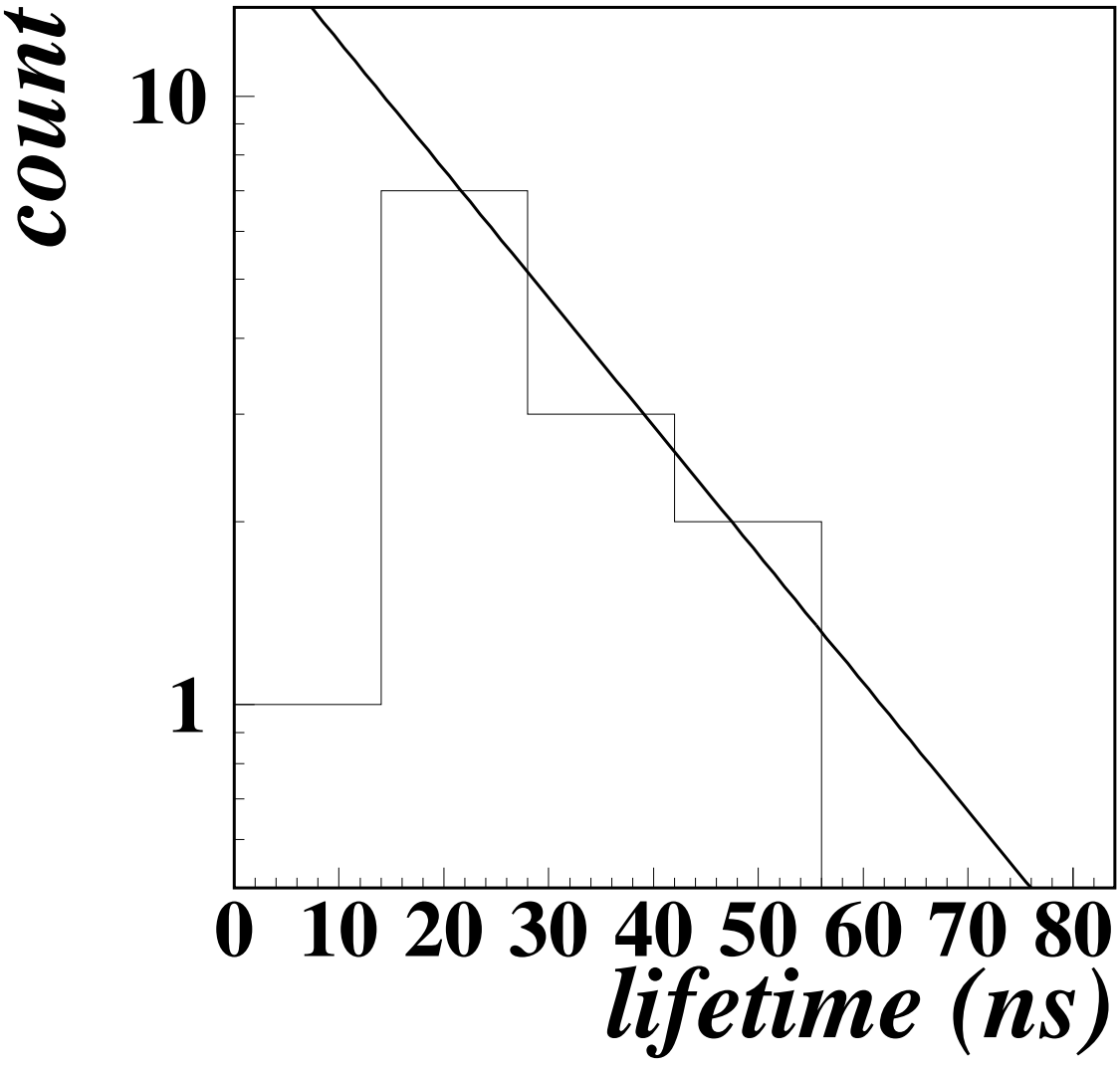


図 5.11: 得られたπ イベント及び最尤法で求まった崩壊曲線

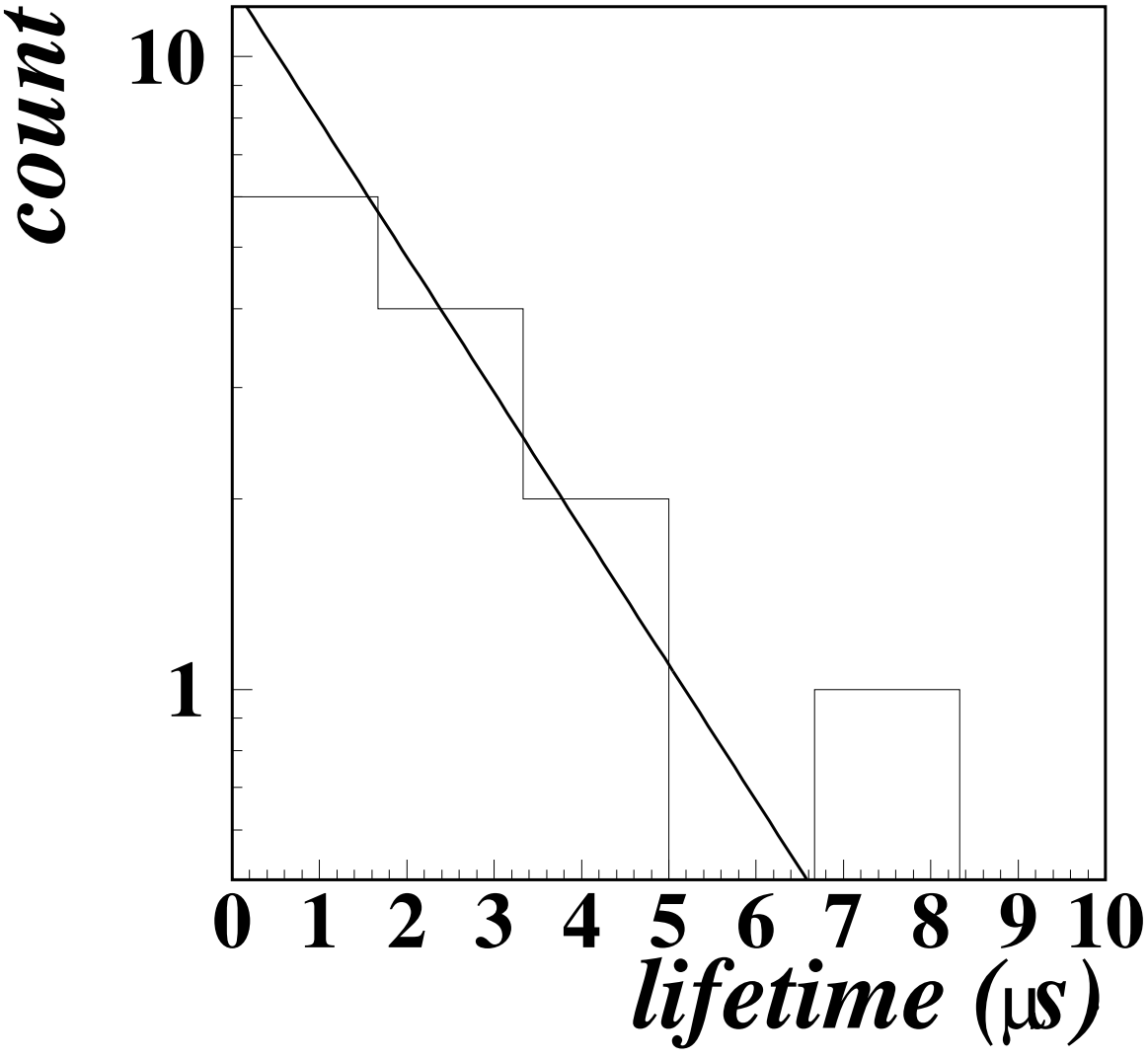


図 5.12: 得られた$\pi\mu$e イベント及び最尤法で求まった崩壊曲線

5.5.2 π^+フラックスの計算

フラックス fという量を、単位面積・単位立体角の領域を単位時間内に貫く粒子の個数として定義すると、我々の検出した粒子の数 n は以下の式で表し得る。

$$n = f \times (\text{天井面積}) \times (\text{Live time}) \times (\text{立体角}) \times (\text{検出効率})$$

なお、我々のシミュレーションでは立体角は既に検出効率の計算に含まれている事を付け加えておく。
これより逆にπ^+のフラックスを見積もってやる事にする。表 5.3に示す実験およびシミュレーションの結果を代入してやればよくて、結果は、

$$f = 4.7^{+1.0}_{-0.7} \times 10^{-8} cm^{-2}\,\mathrm{sec}^{-1}\,Sr^{-1} \tag{5.1}$$

となった。

Ceiling space	$1.02 \times 10^6 cm^2$
Live time	1943997sec
Efficiency including solid angle	2.54×10^{-4}
Number of detected π^+	$23.44^{+5.20}_{-3.66}$イベント

表 5.3: 実験及びシミュレーションの結果

5.5.3 期待されるπ^+イベント 数の計算

我々が期待すべきπ^+イベント数の計算は、U.Camerini の実験 [4] と比較する事で行なわれた。(表 5.4)。

まず始めに、Camerini の実験は Jungfraujoch という高高度の山中で行なわれていた。従って、低地で実験を行なった我々とは明らかに陽子フラックスの強度が異なっている事を考慮せねばならない。小田 [3] と Gaisser [6] によれば、陽子フラックスは飛程とエネルギーについて指数的に減衰し、$100GeV$の N 成分の場合以下の式で表される。

$$\text{フラックス } f(x,E) = f(0)\exp(-\frac{x}{120})E^{-0.7}$$

ここに、$f(0)$ は大気頂上部のフラックスであるが、以下に示すように最終的にはフラックスの比を問題とするので無視して良い。また実際には減衰長$\Lambda \simeq 120g \cdot cm^{-2}$というのはエネルギーおよび高度によって変化しているが、これも荒い近似で無視するとする。
また、飛程 $x(g{\cdot}cm^{-2})$ についてであるが、これは地表面における大気の典型的な深さが $1000g{\cdot}cm^{-2}$ である事及び、我々の実験室における天井及び床面のトータルな厚みが $225g \cdot cm^{-2}$(文献 [5]) である事を考慮する事で、

$$x = 1225g \cdot cm^{-2}$$

であるとした。

	U.Camerini et al.	Our experiment
Detector component	Emulsion(Ilford C2)	Plastic scintillator(NE110)
Volume(cm^3)	9.6	3300
Density($g \cdot cm^{-3}$)	3.815	1.032
Solid angle(Sr)	4π	1.57
Detection efficiency	1.0	0.65
Number of $\pi^+(events/day)$	6.53	$1.0^{+0.23}_{-0.16}$

表 5.4: U.Camerini と我々の実験との比較

ここで、1 日あたりに期待されるπ^+イベント数 $N(\mathrm{events/day})$ というのは、π^+フラックス $f_\pi(E, \cos(\theta))$ を用いて

$$N = \varepsilon n \int f_\pi(E, \cos(\theta)) \sigma_{\pi N}(E) dE d\Omega$$

と表せる。
但し、ε:検出効率
n:ターゲット数
$\sigma_{\pi N}$:π-N 散乱の断面積

いま話を簡単にする為に、以下の仮定を行なう。

1. 陽子フラックス f_pからのπフラックス f_π生成微分断面積は、両実験で等しく、

   $$f_\pi = k f_p$$

   の関係がある。但し、k は生成微分断面積を含む定数係数である。

2. 陽子フラックス及びπフラックスのエネルギー及び天頂角 θ 依存性は、我々の見ている範囲で無視できる。

3. π-N 散乱の断面積についても同様とする。

4. 入射粒子は物質中のどの核子に対しても、等しい確率で反応する。

そうすると、上の式は

$$N = \varepsilon n k f_p \sigma_{\pi N} E \Omega$$

と書き直される。またこの時ターゲット数 n は、

$$n = \rho \times V \times N_a$$

(但し、ρ:密度、V:体積、N_a:Avogadro 数)
表わされるので、結局求める式は、

$$N = \varepsilon \rho V N_a f_p \sigma E \Omega$$

となる。
これを U.Camerini と我々の実験について立てた式を比較すれば、我々の 1 日に期待すべきπ^+イベント数 N が分かる。実際には、等しい値を消去して

$$N_{我々} = \frac{\varepsilon_{我々}\rho_{我々}V_{我々}f_{我々}\Omega_{我々}}{\varepsilon_{Came}\rho_{Came}V_{Came}f_{Came}\Omega_{Came}} \cdot N_{Came}$$

と表わせる。後はこれに既知の値を代入してやればよくて、

$$N_{我々} = \frac{0.65 \times 1.032 \times 3300 \times \exp(-\frac{1225}{120}) \times 1.57}{1.0 \times 3.815 \times 9.6 \times \exp(-\frac{690}{120}) \times 4\pi} \cdot 6.53 \simeq 0.57(\text{イベント}/1\text{ 日}) \qquad (5.2)$$

となる。

5.6 π及びμのエネルギーについて

5.6.1 πとμのパルスの分割

デジタルオシロスコープで見られるシングルパルスは、式 (4.1) で表される。したがって実験で得られたダブルパルスを上式で fitting する。実際はパルスは 2 つあるため、この関数 2 つとコンスタントなノイズ部分を含めたパラメータ 9 つでフィッティングを行った。(図 5.13)

これによってπとμのパルスを分割する事ができた。(図 5.14)

5.6.2 πのエネルギー

π-μ崩壊によるμのエネルギーは、4.12 MeV であることが分かっているので、πのエネルギーをこれを基準にしてデジタルオシロスコープのπとμのパルスを比例計算により求めた。

さらに、止まるまでに通過した 2 枚の板に落としていったエネルギーを ADC で測ってやることによって、πのエネルギーが算出できる。これは、先に求めた ADC 値の範囲内にあった。(表 5.5)

Camerini の論文によればπ^+のエネルギーは約 35MeV である。この値と比較すると、今回の実験で得られた値とは矛盾するイベントもある。その理由としては、

- 板と板との間は実際には空いており、空気によるエネルギー損失を考慮していない。
- A 1 の 1ch あたりのエネルギーのエラー幅が 10 倍以上違う。
- パルスを分割する際にノイズ部分の constant を多くとりすぎた。

といったことがあげられる。

表 5.5: πのエネルギー (単位:MeV)

イベント	πのエネルギー
1	$29.3^{+18.2}_{-14.5}$
2	$24.4^{+10.2}_{-9.4}$
3	$25.2^{+9.9}_{-8.1}$
4	$26.0^{+10.5}_{-9.5}$
5	$33.3^{+27.1}_{-21.1}$
6	$19.5^{+10.8}_{-9.8}$
7	$18.2^{+11.8}_{-10.0}$
8	$38.4^{+23.4}_{-18.4}$
9	$21.5^{+11.3}_{-9.4}$
10	$15.2^{+5.8}_{-6.0}$
11	$19.7^{+14.8}_{-11.0}$
12	$21.1^{+8.1}_{-8.9}$
13	$24.0^{+8.2}_{-8.9}$
平均	$24.3^{+13.1}_{-11.2}$

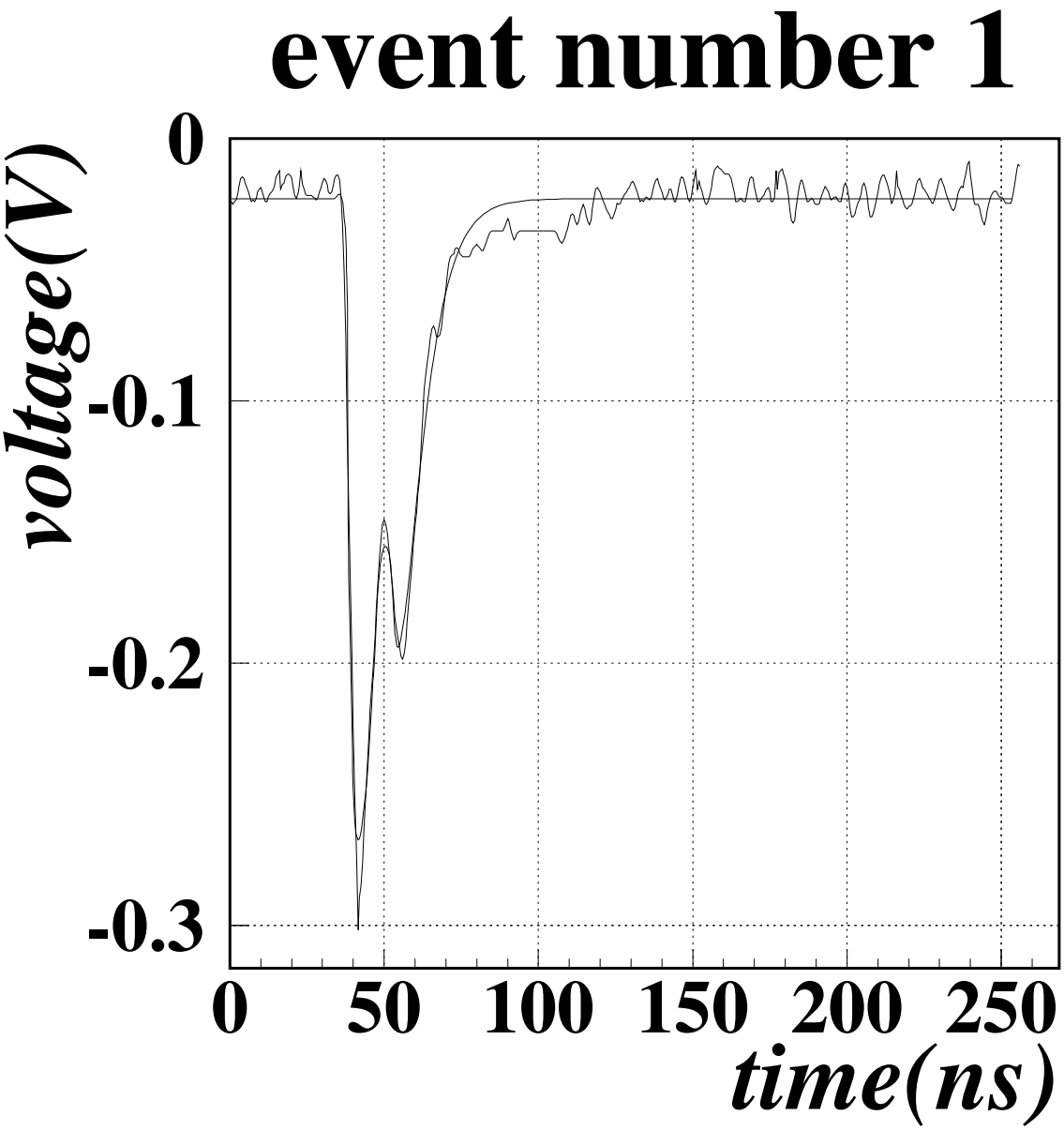


図 5.13: パルスの fitting

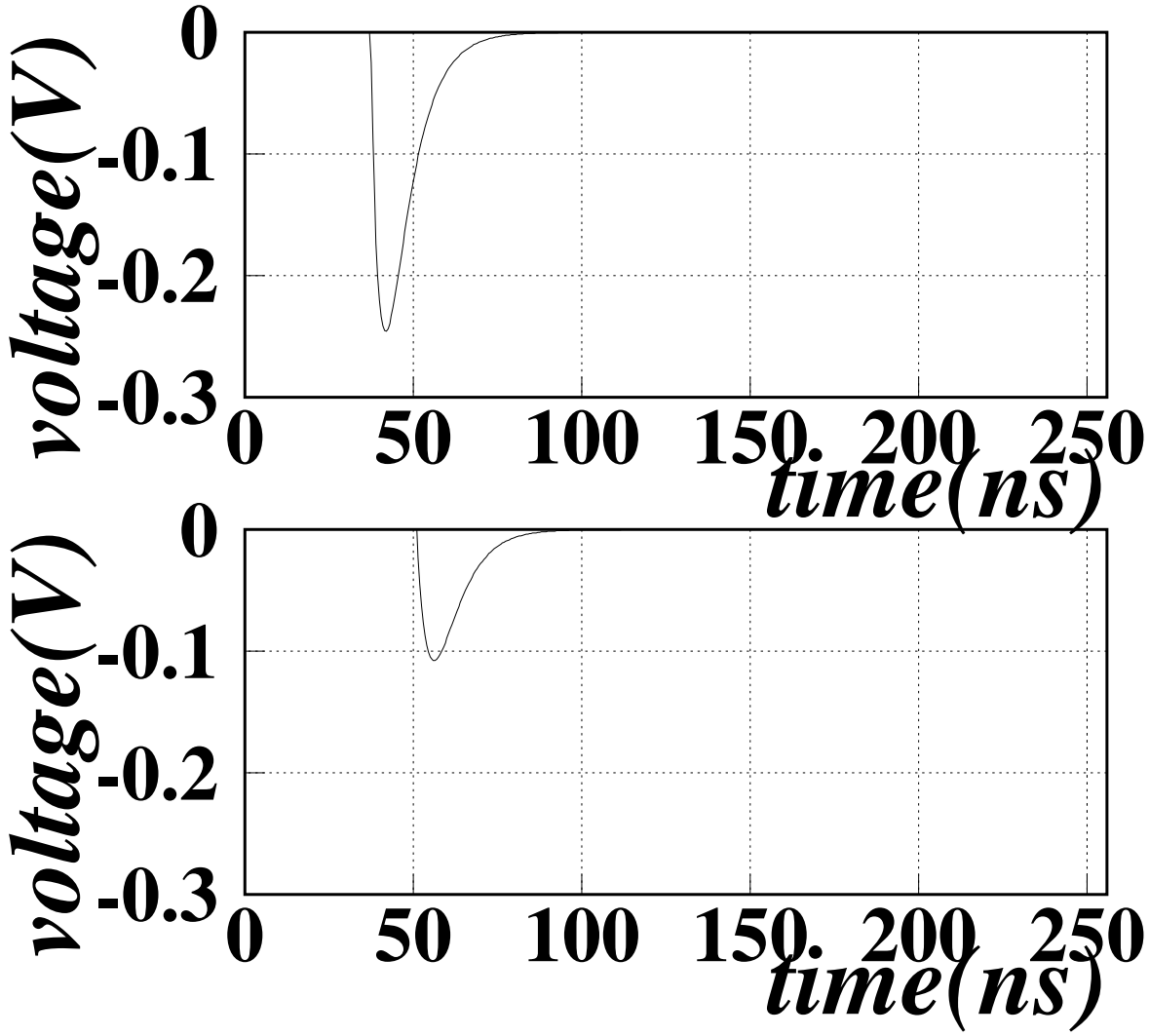


図 5.14: 分解したパルス (a)π,(b)μ

第 6 章

結論

我々は、イベント選別の結果、πイベントとして、13 イベントを得た。これは、0.74 のバックグラウンドよりも十分多く、πとμの寿命についても期待値と一致している。また、数え落しを含めた一日当りのイベント数は、$1.04^{+0.23}_{-0.16}$であり、式 5.2で求めた期待値 0.58 と矛盾しない。従って、この 13 イベントは、$\pi^+ \longrightarrow \mu^+ \longrightarrow e^+$ 崩壊を捕らえた物と言える。

式 5.1から、π^+の我々の実験室での下向きのフラックスは、

$$f = 4.7^{+1.0}_{-0.7} \times 10^{-8} cm^{-2}\,\mathrm{sec}^{-1}\,Sr^{-1}$$

となる。

謝辞

まず最初に、今回の実験を行なう機会を与えて下さり、また学問の上でもにも人間的にもまだまだ至らぬ所の多い我々に、幅広い知識と教養を啓蒙して頂いた長島順清先生に深くお礼を申し上げます。

原隆宣先生には、今回の実験において、右も左も分からぬ我々に基本中の基本から手取り足取り御指導していただき、本当にありがとうございました。

瀧田正人先生には、我々の気づかないような深い観点からのアドバイスや様々な教育的指導をいただき、大変勉強になりました。ありがとうございました。

山中卓先生と羽澄昌史先生には、多忙にも関わらず、折りに触れ我々の実験に対する様々な御指摘をいただきました。ありがとうございました。

秘書の続木佐知子さんには、事務関係で数々の御面倒をおかけしながらも大変お世話になり、またお忙しい中、時には話し相手にもなっていただき、研究生活を楽しくしていただきました。ありがとうございました。

長島研究室の博士課程に在学中の、山口さん、川崎さん、花垣さん、貞本さん、千代さん、住澤さん、日高さん、修士課程に在学中の、香山さん、持田さん、吉田さん、柳幸さん、紅林さん、火口さん、北條さん、森さん、山田さんには、あらゆる面にわたって我々のよき相談相手、話相手となっていただきました。

しばしば、先輩方の仕事の手を止めてしまったり、夜遅くまで我々につきあわせてしまいましたが、どんな時でも快く教えて頂きました。本当にありがとうございました。

参考文献

[1] 1990 年度長島研究室 4 年生卒業実験.

[2] 物理学編集委員会. 物理学辞典 *1986* 年. 培風館

[3] 小田 稔.「宇宙線」. 裳華房

[4] U.Camerini. *Nature,***162***(1948)433.*

[5] 1989 年度長島研究室 4 年生卒業実験.

[6] Thomas K.Gaisser. *Cosmic Rays and Particle Physics*. Cambridge University Press

[7] William R.Leo. *Techniques for Nuclear and Particle Physics Experiments*. Springer-Verlag

[8] Philip R.Bevington. D.Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill,Inc.

付録 A

ソフトウェア

ここでは、我々の実験のデータ収集システムを制御するのに用いた、オンラインプログラムのメインプログラム、サブルーチンを掲載する。サブルーチンには、主に、GPIB インターフェイス CAMAC モジュールを介してディジタルオシロスコープを操作する関数が入っている。

また、このサブルーチンを使って、ディジタルオシロスコープを操作する簡単なサンプルプログラムも載せた。GPIB インターフェイス CAMAC モジュールの使い方を知りたい場合は、このサブルーチンとサンプルプログラムを参考にできるだろう。

A.1 オンラインプログラム

A.1.1 ヘッダーファイル

```
#include<stdio.h>
#include<camac.h>
#include<dos.h>
#include<stdlib.h>
#include<time.h>

#define N_TDC 4
#define N_ADC 2
#define N_IR  7
#define N_GPIF 12
#define N_SCAL 16
#define N_IR2 20

#define NOCOMMA(P)   (P) == ',' ? '\n' : (P)
#define NOCOMMA2(P)  (P) == ',' ? '' : (P)
#define MAXLINE 1024
#define  paste(f,b) "f" ## "b"

char *whmd(Uint a,Uint b, Uint c);
void pq(char *s, Uint j);
```

```
int c_break(void);
void die(const char *s,const char *t);
void stripcomment(char *s);
int blank(char *s);

int mystrcat(char *sre, const  char *sto, const char *sfr);
int get_adc(const char *filen, Uint ch, Uint count);
```

A.1.2 メインプログラム

```
#include "test.h"

Uint sumfrom, sumend, adcch, adcct;
double preamble[10];
FILE *log_file;

void main()
{
  int ac, nostop=0;
  char *commfile="comIn";
  Uint  i, j, MAX, Q;
  time_t mt;

  FILE *adcfp,*tdcfp,*dfp;
  char filename[12];
  char adfn[12], gpfn[12], tdfn[12], amble[12],dfn[12];
  char wafn[12];
  int fnl;
  char evnum[25];
  Uint qx,dead;

  FILE *initfp;
  char str[1024];
  char  s[256],tmp[256];
  Uint d1, d2;

  Uint event,adata[5],tdata[8];
  char chk,trig;
  int ter;

  /* *** open log file for LOG all message */
  if((log_file = fopen("log_file.log", "a")) == NULL) {
    printf("Cann't open log file");
    exit (1);
  }
  mt=time(NULL);
  fprintf(log_file,"# Start prog at: %s\n", ctime(&mt));


  /* get initization info. */
  if((initfp=fopen("setup", "r")) == NULL)
```

```
    {
      printf("Cannot open initization file %s\n", "setup");
      exit (-1);
    } else {
      printf("Ok open init. file %s\n", "setup");
    }
  while(fgets(str, MAXLINE, initfp))
    {
      stripcomment(str);
      if(blank(str)) continue;

      strcpy(tmp,str);
      sscanf(tmp, "%s %d %d", s, &d1, &d2);
      if(!strcmp(s, "GPIF:"))
        {
          sumfrom=d1; sumend=d2;
        } else if (!strcmp(s, "ADC:"))
          {
            adcch=d1; adcct=d2;
          } else {
            printf("bad line format\n");
          }
    }
  fclose(initfp);
  fprintf(log_file,"adc: channel is %d , count is %d\n", adcch, adcct);

#ifdef DEBUG
  printf("GPIF sumfrom is %d sumend is %d", sumfrom,  sumend);
  printf("ADC num of ch is %d, ct is %d\n", adcch, adcct);
#endif

  /*************** open file*********/
  printf("Input file name : ");
  gets(filename);
  fnl = strlen(filename);
  if( fnl == 0 || fnl >4 || strchr(filename,'.')){
    printf("The filename must be shorter than 8\n");
    printf("and don't attach extension \".dat\" after filename\n");
    exit (1);
  }

  mystrcat(adfn,filename,"A.dat");
  mystrcat(tdfn,filename,"T.dat");
```

```
mystrcat(amble,filename,".amb");
mystrcat(gpfn,filename,"G");
mystrcat(dfn,filename,"D.dat");

printf("adc File name is %s\n",adfn);
printf("gpif File name is %s#.ch#\n",gpfn);
printf("td File name is %s\n",tdfn);
printf("preamble File name is %s\n",amble);
printf("dead time File name is %s\n",dfn);

if((adcfp = fopen(adfn, "w")) == NULL)
  die("Cann't open adc data file",adfn);
else
  fprintf(log_file,"Ok open adc data file %s\n", adfn);
if((tdcfp = fopen(tdfn, "w")) == NULL)
  die("Can't open tdc data file",tdfn);
else
  fprintf(log_file,"Ok open tdc data file %s\n", tdfn);
if((dfp = fopen(dfn, "w")) == NULL)
  die("Can't open dead time file",dfn);
else
  fprintf(log_file,"Ok open dead time data file %s\n", dfn);

/************init*****************/

setcn(0);
delay(10);
execc();
delay(10);
execz();
delay(10);

camac_0(N_ADC, 0, 26);
camac_0(N_TDC, 0, 26);
camac_0(N_IR, 1, 26);
camac_0(N_SCAL, 1, 9);
camac_0(N_SCAL, 1, 10);

init_gpif2();

if(nostop)
  printf("nostop\n");
else
```

```
  ctrlbrk(c_break);

talk();
send(commfile);
get_preamble(amble);
sleep(1);

for(event=1;event<=adcct;)
  {
    ctrlbrk(c_break);
    talk();
    send("*CLS_");
    delay(10);
    /*check operation complete*/
    send("*OPC?_");
    listen();
    printf("OPC=%d\r",get_num());
    talk();
    send(":RUN_");
    camac_0(N_ADC, 0, 9);
    camac_0(N_TDC, 0, 9);
    delay(10);
    /*check operation complete*/
    send("*OPC?_");
    listen();
    printf("OPC=%d\r",get_num());

    /*dead time*/
    camac_16(N_SCAL, 1, 2, &dead);
    fprintf(dfp,"%u\n",dead);

    camac_0(N_IR, 0, 9);

    /*check LAM of ADC*/
    while((qx=camac_0(N_ADC, 0, 8))>0){
      if(qx>=2){
        printf("error!\n");
        fprintf(log_file,"error:adc q > 1 \n");
        fcloseall();
        exit(1);
      }
    }
```

```
/*trigger check*/
delay(100);
i=0;
do{
  talk();
  send(":TER?_");
  delay(10);
  listen();
  ter=get_num();
  printf("\t\t\t\ttrigger=%d\r",ter);
  if(!ter)delay(100);
  i++;
}while(!ter && i<50);

if(!ter){
  mt=time(NULL);
  printf("Digital Oscillo was not trig'd. Modules are cleared.\n");
  fprintf(log_file,"Digttal Oscillo not trig'd: %s\n", ctime(&mt));
}

else{
  /*get tdc data*/
  chk=0;
  trig=0;
  for(i=0;i<8;i++){
    camac_16(N_TDC, i, 0, &tdata[i]);
    if(tdata[i]>=10 && tdata[i]<=1000)chk++;
  }
  /*event check*/
  if(chk>=1){

    fprintf(tdcfp,"%3d",event);
    for(i=0;i<8;i++){
      fprintf(tdcfp,",%4u",tdata[i]);
    }

    printf("########################################");
    printf("########################################\n");
    printf("event# %d\ntdc ch0 - ch7 saved\n",event);
    printf("tdc data:%4d",tdata[0]);
    for(i=1;i<8;i++)printf(",%4d",tdata[i]);
    printf("\n");
    /*get adc data*/
```

```
    fprintf(adcfp,"%d",event);
    for(i=0;i<adcch;i++){
      camac_16(N_ADC, i, 0, &adata[i]);
      fprintf(adcfp,",%4u",adata[i]);
    }
    fprintf(adcfp,"\n");

    printf("adc ch0 - ch%d saved\n",i-1);
    printf("adc data:%4d",adata[0]);
    for(i=1;i<adcch;i++)printf(",%4d",adata[i]);
    printf("\n");

    fprintf(tdcfp,",%4u\n",adata[4]);

    /*get wave data*/
    talk();
    send(":WAVEFORM:SOURCE CHANNEL1_");
    itoa(event,evnum,10);
    strcpy(wafn,gpfn);
    mystrcat(wafn,gpfn,evnum);
    strcat(wafn,".ch1");
    printf("digital oscillo ch1,filename:%s\n",wafn);
    send(":WAVEFORM:DATA?_");
    listen();
    save_wave(wafn);

    talk();
    send(":WAVEFORM:SOURCE CHANNEL2_");
    itoa(event,evnum,10);
    strcpy(wafn,gpfn);
    mystrcat(wafn,gpfn,evnum);
    strcat(wafn,".ch2");
    printf("digital oscillo ch2,filename:%s\n",wafn);
    send(":WAVEFORM:DATA?_");
    listen();
    save_wave(wafn);

    printf("########################################");
    printf("########################################");

    event++;
  } else {
    printf("skip a event\t");
```

```
        }

      }

    }

  printf("end\n");
  mt=time(NULL);
  fprintf(log_file,"# End prog at %s\n\n",ctime(&mt));
  fcloseall();

}
```

A.1.3 サブルーチン

```
#include "test.h"

extern Uint sumfrom, sumend, adcch, adcct;
extern double preamble[10];
extern FILE *log_file;


/* initialize GPIB interface */
void init_gpif2(void)
{
  Uint Q;
  if((Q=camac_0(N_GPIF, 0, 25)) != 0){          /* Interface clear */
    printf("NOTES: Interface is not BE STARTED");
    exit (-1);
  } else {
    printf("%d\n",Q);
  }
  camac_0(N_GPIF, 0, 26);               /* All devices acceptance */
  camac_0(N_GPIF, 1, 26);               /* Remote enable */
  checkport(N_GPIF,1,27);               /*Check the state of output ready*/
  camac_0(N_GPIF, 4, 26);               /*Enables the Universal and Addressed command inhibi
}

/* send command to digital oscilloscope */
void send(char *str)
{
  int i;
  char line[1024];
  FILE *infp;
  if(str[0] == '*' || str[0] == ':') {
    for(i=0; i<strlen(str); i++){
      if(str[i] == '_' ) str[i] = '\n';
#ifdef DEBUG
      printf("length [%d], char [%c], digit [%d]\n",strlen(str),str[i],str[i]);
#endif
      checkport(N_GPIF,1,27);
      camac_16w(N_GPIF, 0, 16, str[i]);
    }/*printf("Completed send %s\r",str);*/
  } else  {
    if((infp=fopen(str, "r")) == NULL)
      { die("Send: cann't openfile : ", str);}
```

```
    else { printf("Send:Ok open file [%s]\n", str); }

    while((fgets(line, MAXLINE, infp)) != NULL)
      {
        stripcomment(line);
        if (blank(line)) continue;

        for(i=0; i < strlen(line); i++)
          {
#ifdef DEBUG
            printf("length [%d], char [%c], digit [%d]\n",strlen(line),line[i],line[i]);
#endif
            checkport(N_GPIF,1,27);
            camac_16w(N_GPIF, 0, 16, line[i]);
          }
      }
    fclose(infp);
    printf("Send: Completed sennding comm. file [%s] over.\n", str);
  }
}

/* make GPIB interface talker, digital oscilloscope listener */
void talk(void){
  camac_0(N_GPIF, 0, 26);              /* All devices acceptance */
  camac_16w(N_GPIF, 0, 16, 36); /* HP listen address = 4+ 32*/
  camac_16w(N_GPIF, 0, 16, 71); /* GP-IF talk address  = 7+ 64 */
  while(camac_0(N_GPIF, 1, 27));               /*Check the state of output ready*/
  camac_0(N_GPIF, 0, 24);                      /* End command mode */
  printf("Completed TALK\r");
}

/* make GPIB interface listener, digital oscilloscope talker */
void listen(void)
{

  camac_0(N_GPIF, 0, 26);              /* All devices acceptance */
  camac_16w(N_GPIF, 0, 16, 68); /* HP TALK address = 4+ 64*/
  camac_16w(N_GPIF, 0, 16, 39); /* GP-IF LISTEN address  = 7+ 32 */
  while(camac_0(N_GPIF, 1, 27));               /*Check the state of output ready*/
  camac_0(N_GPIF, 0, 24);                      /* End command mode */
  printf("Completed LISTEN\r");
}
```

```
/* save wave data on hard disk */
void save_wave(char *str)
{
  FILE *fp;
  Uint Q;
  Uint i=0,z,max;
  int l;
  int nodata=0;

  double volt;
  char tmpstr[16]="";
  char k=0,j;

  if ((strlen(str)) != 0 && (fp = fopen(str, "w")) != NULL )
    {
      fprintf(stderr,"datafile is %s\n", str);
      max=preamble[2];Q=1;
      printf("%u data points\n",max);
      for(i=0;i<max && Q != 10;){
        z=camac_16(N_GPIF, 0, 0, &Q);
        if(!z)
          {
            if(Q != ',' && Q != '\n')
              {
                tmpstr[k++]=Q;
              }
            else{
              i++;
              /* convert data into voltage */
              l=atoi(tmpstr)&32767;/* fix bug of oscilloscope */
              volt=(l-preamble[9])*preamble[7]+preamble[8];
              fprintf(fp, "%lE\n", volt);
              k=0; for(j=0;j<16;j++) tmpstr[j]=0;
            }
          }
        else if(z > 1)
          {
            printf("Error!\a\a\a\a\a\a\n");
            break;
          }
        else
          {
            printf("wait...\r");
```

```
            nodata++;
            delay(10);
          }
      }

    } else {
      fprintf(stderr,"Cannot open file %s",str);
      exit(1);
    }
  printf("%u points saved, total %d nodata-\n",i,nodata);
  fclose(fp);
}

/* check whether a module is available */
int checkport(Uint md, Uint ckm, Uint ckn)
{

  long ck=0;
  if(camac_0(md, ckm, ckn) == 0){
    return 0;
  } else {
    fprintf(stderr,"Waiting for %s ...\r", whmd(md,ckm,ckn));
    while(camac_0(md, ckm, ckn))
      {
        ck++;
        if(ck > 100000){
          fprintf(stderr, "Check %s time out\n",whmd(md,ckm,ckn));
          exit (-1);
        }
      }
    return 1;
  }
}

char *whmd (Uint mp,Uint p1, Uint p2)
{
  static char str[256];
  static char ostr[256];
  switch(mp) {
  case N_TDC:   strcpy(str,"TDC"); break;
  case N_ADC:   strcpy(str,"ADC"); break;
  case N_GPIF:  strcpy(str,"GPIF"); break;
  case N_IR:    strcpy(str,"IR"); break;
```

```
            }
  sprintf(ostr, "Module %s: %d_%d", str, p1, p2);
  return  ostr;
}

void die(const char *s,const char *t)
{
  fprintf(stderr,"%s : [%s]\n", s, t);
  fprintf(log_file,"%s : [%s]\n", s,t);
  fclose(log_file);
  exit (-1);
}

int mystrcat(char *sre, const char *sto, const char *sfr)
{
  strcpy(sre, sto);
  strcat(sre,sfr);
  return (strlen(sre));
}

void stripcomment(char *s)
{
  char *t = (char *)strchr(s,'#');
  if(t && (t==s ||t[-1] != '\\')) *t ='\0';
}

int blank(char *s)
{
  while(*s == ' ' || *s == '\t' || *s == '\n') ++s;
  return !*s;
}

/* break program when CTRL-C is pressed */
int c_break(void)
{
  int closed_files;
  time_t et;
  et=time(NULL);
  fprintf(log_file,"# End prog at %s\n\n",ctime(&et));
  closed_files=fcloseall();

  if (closed_files == EOF)
    perror("test");
```

```
  else
    printf("%d streams were closed.\n", closed_files);
  printf("CTRL-C pressed. Program aborting ...\n");
  return 0;
}

/* get preamble of digital oscilloscope */
void get_preamble(char *str)
{
  char temp[30]="";
  char i=0,j=0,k=0,q=1;
  Uint Q;
  FILE *fp;

  talk();
  send(":WAVEFORM:PREAMBLE?_");
  listen();
  checkport(N_GPIF,0,27);
  if ((strlen(str)) != 0 && (fp = fopen(str, "w")) != NULL ){
    for(k=0;k<10 && Q != 10;){
      q=camac_16(N_GPIF, 0, 0, &Q);
      if(!q){
        if(Q != ',' && Q != '\n'){
          temp[i]=Q;
          i++;
        }
        else{
          preamble[k++]=atof(temp);
          for(j=0;j<30;j++)temp[j]=0;
          i=0;
        }
      }
      else{
        printf("nodata-");
        delay(10);
      }
    }
  }
  else {
    fprintf(stderr,"Cannot open file %s",str);
    exit(1);
  }
  for(i=0;i<10;i++)
```

```
    {
      fprintf(fp,"%lE\n",preamble[i]);
      printf("%lE\n",preamble[i]);
    }
  fclose(fp);
}

/* get one return value from digital oscilloscope */
int get_num(void)
{
  Uint Q=0;
  Uint z,i=0;
  int l;
  int nodata=0;
  char temp[32];
  char j;
  while(Q != 10 && nodata<200){
    z=camac_16(N_GPIF, 0, 0, &Q);
    if(!z)
      {
        temp[i++]=Q;
      }
    else if(z > 1)
      {
        printf("Error!\a\a\a\a\a\a\n");
        break;
      }
    else
      {
        nodata++;
        delay(10);
      }
  }
  return(atoi(temp));
}
```

A.2 サンプルプログラム

このサンプルプログラムを使うと、PC-9801 のキーボードで、ディジタルオシロスコープを操作することが出来る。

使い方を簡単に説明する。まず、プログラムを走らせると、“command ?” と表示される。ここで、ディジタルオシロスコープのコマンドを入力すると、送信される。ディジタルオシロスコープからデータをもらうようなコマンドを送った場合、そのあとに、’ L ’ を入力するとデータが画面に表示される。’ Q ’ を入力すると終了する。

```
#include "test.h"

void main()
{
  char *commfile="comIn";

  char command[32];

  setcn(0);
  execc();
  execz();

  /*initialize*/
  init_gpif2();

  talk();
  send(commfile);

  for(;;){
    ctrlbrk(c_break);
    printf("command ?");
    gets(command);

    switch(command[0]){
    case 'T':
      talk();
      break;
    case 'L':
      listen();
      get_data();
      talk();
      break;
    case ':':
      strcat(command,"_");
```

```
      send(command);
      break;
    case '*':
      strcat(command,"_");
      send(command);
      break;
    case 'Q':
      exit(0);
      break;
    default:
      printf("invalid command, or use CAPITAL LETTER\n");
    }
  }
}

void get_data(void)
{
  Uint Q=0;
  Uint z;
  int nodata=0;

  while(Q != 10 && nodata<200){
    z=camac_16(N_GPIF, 0, 0, &Q);
    if(!z)
      {
        printf("%c",Q);
      }
    else if(z > 1)
      {
        printf("Error!\a\a\a\a\a\a\n");
        break;
      }
    else
      {
        nodata++;
        delay(10);
      }
  }
}
```