Implementation of the Domino Sampling Waveform Digitizer in the PIBETA Experiment.

Ying Wang
Qihe, Shandong, China

B.S. University of Science & Technology of China , 1998
M.A. University of Virginia , 2001

A Dissertation presented to the Graduate Faculty of the University of Virginia in Candidacy for the Degree of Doctor of Philosophy

Department of Physics
University of Virginia
July, 2005

Abstract

The Domino Sampling Chip(DSC)-Waveform digitization system is a significant addition to electronics arsenal of PIBETA experiment. It is used to digitize waveforms from every photo tube in the detector. Through carefully programmed offline analysis of its raw data collected during regular runtime, better timing and energy resolution are achieved compared with feast's results. And more importantly, the geometric character of the digitized waveform which contains information of energy deposition of particle decays can be utilized for particle identification, a great advantage that regular unit could not possess. In addition to fastbus , incorporate DSC data through its offline analysis including timing and energy offset, scale calibration will contribute a final more precise result of PIBETA experiment.

Acknowledgments

First I would thank my research adviser Prof. Dinko Pocanic for his patience and inspiration throughout the whole period of my thesis work. I would also thank Dr. Emil Frlez for his directions and insightful instructions. Without guidance from them the essential parts of this thesis work can not be constructed in a timely manner. Also the enlightenment I got in the live discussions with Weidong Li, Brent Vandevender, Maxim Bychkov are strong impetus to my past and future academic endeavors. Such tremendous help from our members of PIBETA collaboration can not be exaggerated. Special thanks are given to our PIBETA collaborator Dr. Stefan Ritt and Dr. Heinz-Peter Wirtz for the DSC software frame work and hardware system construction. Without their excellent cornerstone contribution, the author can not obtain any fruitful results. Any significance, if this thesis work does have, should attribute to other members of PIBETA collaboration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Standard model

1.1.1 The fundamental fermions and their interactions

Practically all experimental data from particle physics and high energy experiments can be accounted for by so-called *Standard Model* of elementary particles and their interactions, formulated in the 1970s. According to this model, all matter is built from three families (generations) of fundamental spin $\frac{1}{2}$ particles, or fermions: six quarks and six leptons and their interactions are described in terms of the exchange of characteristic boson mediators which is summarized in Table 1.1.

Unsatisfied with having four independent and apparently unrelated interaction fields, physicists from Einstein onward have speculated that the different interactions

Table 1.1: the fundamental fermions and boson mediators

particle	flavor		
quarks	u	c	t
	d	s	b
leptons	e	μ	τ
	ν_e	ν_μ	ν_τ

Interaction	Mediator
strong	gluon, G
electromagnetic	photon, γ
weak	$W^\pm$, Z^0
gravity	graviton, g

are different aspects of a single, unified field. In the 1970's, experiments showed that the weak and electromagnetic interactions can indeed be unified into so-called electroweak interaction and would have the same strength at very high energy; only at lower energies is symmetry broken with different apparent strengths.

Within the Standard Model, both the leptons and quarks are assigned to be the left handed doublets and right handed singlets. The electroweak interaction has $SU(2) \times U(1)$ as the gauge group. Not being the same as the weak eigenstates, the quark mass eigenstates are skewed so that the six quark bases are related by the Cabibbo-Kobayashi-Maskawa (CKM) matrix with small but nonzero off-diagonal elements.

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}$$

The quark mixing has a special significance for its systematic description of weak interaction of mesons (made of one quark and one anti-quark) and baryons (three quarks combination) and nuclei. The validity of the Standard Model requires the unitarity of the CKM matrix. Excluding certain hitherto undiscovered processes CKM matrix unitarity implies that the number of quark generations is three. Precise experimental evaluation of the unitarity of the CKM matrix gives a sensitive test of new physics.

1.1.2 Limitations of Standard Model

Although Standard model successfully accounts for an enormous body of experimental data, from laboratory experiments ranging up to the highest available collision energies, it does have limitations which includes: Gravitational interactions are not included, neutrinos are assumed to be massless and there is growing evidence from the solar and atmospheric anomalies that neutrinos do have nonzero masses. With possible manifestation of physics beyond that encompassed by the Standard Model, stringent tests of this model are needed. Our PIBETA experiment is proposed to aid in this effort.

1.2 Motivation of the PIBETA experiment

1.2.1 Why Pion Beta decay?

For nine elements of CKM matrix, many relationships can be tested from experiment. Choosing π^+ which is built from the first quark generation, we can test the unitarity of the first row of CKM matrix, that is,

$$|V_{ud}|^2 + |V_{us}|^2 + |V_{ub}|^2 = 1.0$$

V_{us} was extracted [3] from K_{e3} decays and for many years gave $|V_{us}| = 0.2196 \pm 0.0023$. V_{ub} given in Particle Data Group (PDG) [3] data is $|V_{ub}| = (3.6 \pm 0.7) \times 10^{-3}$, dominated by theoretical uncertainty. The situation changed drastically in 2003 and 2004 when a series of experiments at Brookhaven, Fermilab and CERN reported revised values of K_{l3} decay branching ratios which led to an upward adjustment, by about 2.5σ, of the CKM matrix element V_{us} [43, 43, 45]. In this revolutionary development, a revised check yields full agreement with CKM unitarity [46]

$$\triangle = 1 - |V_{ud}|^2 - |V_{us}|^2 - |V_{ub}|^2 = (3 \pm 14) \times 10^{-4}.$$

The first leading element, V_{ud}, comes from quarks in the first generation and is the element that can be extracted most precisely. It also is the dominant one which needs to be determined as precisely as possible. The value of V_{ud} can be extracted from three distinct sources: the decay of the free neutron, nuclear super-allowed Fermi beta decays, and pion beta decay.

Free neutron decay has the features that it involves no nuclear-structure dependent corrections but is not purely vector-like (with axial-vector contributions). So, besides a lifetime measurement, correlation experiment to separate vector and axial-vector parts is also required. Both types of experiments met with tremendous experimental challenges. The results obtained from free neutron decay experiments show this complexity. Although past results have yielded the unitarity sum above unity, the most recent result (Perkeo II [7]) yields:

$$V_{ud} = 0.9717 \pm 0.0013$$

The uncertainty of $|V_{ud}|$ obtained from neutron decay is dominated by experimental results. With improved experimental technologies, the experimental uncertainty will drop and eventually will be lower than the theoretical correction, which is common to both neutron and nuclear decays.

Compared with neutron decay, nuclear super-allowed Fermi beta decay ($0^+ \rightarrow 0^+$) has the advantage that it depends solely on the vector part of the weak interaction and, with the allowed approximation, the matrix element for these transitions is just the expectation value of the ladder operator if isospion which is independent of any nuclear structure details and is shown to be simply as an $SU(2)$ Clebsch-Gordan coefficient. Thus, the $ft-$Values from experiment are expected to be almost the same for all $0^+ \rightarrow 0^+$ transitions between states of a specific isospin, regardless of the nuclei involved. Extracting V_{ud} from experimental data requires a determination of the $ft-$values for a variety of different nuclei with the same isospin, and a test of their self-consistency. When passing the test, a value for the weak vector coupling constant (G_V) is obtained by taking the average, and from it, V_{ud} is extracted. The result thus obtained in [5] is:

$$|V_{ud}| = 0.9740 \pm 0.0005.$$

The experimental results from nuclear Fermi beta decay rates can be very precise (the uncertainty coming from experiments could be only 0.0001). The largest contributions to the $|V_{ud}|$ uncertainty arise from nucleus-independent part and nuclear structure dependent corrections of radiative correction which have no simple solutions.

Pion beta decay has the advantage over nuclear beta decays in that there are no nuclear structure-dependent corrections to be decided. It also possess the same advantage as the nuclear decays in being a purely vector transition, $0^- \rightarrow 0^-$, so no separation of vector and axial-vector components is needed. The disadvantage, however, is its small decay rate of $\pi^+ \rightarrow \pi^0 e^+ \nu$, at the order of 10^{-8}, which will presents a big experimental challenge.

1.2.2 Pion beta decay experiments—previous and current

The previous experiments measuring the pion beta decay branching ratio showed good agreement with the Standard Model but have large uncertainties. For example, the experiment performed by Depommier et al. [8] used a carbon degrader together with an active CH_2 target to stop 77 MeV pions at a rate of $\sim 3.5 \times 10^4$/s. The calorimeter there consisted of an array of eight lead-glass counters that covered 60% of the $4\,\pi$ solid angle. The radial thickness was equivalent to about 7 radiation lengths. The detector efficiency was calibrated using the single charge exchange reaction $\pi^- + p \rightarrow \pi^0 + n$ (SCX) followed by $\pi^0 \rightarrow \gamma\gamma$ with a precision of 3.6 %. In this way they got a branching ratio of $1.00^{+0.08}_{-0.1} \times 10^{-8}$.

Before our work, the experiment by McFarlane et al [9] had the best uncertainty on pion beta decay branching ratio of 3.9 %. This uncertainty propagates to 1.7 % uncertainty on $|V_{ud}|$ and 3.2 % uncertainty on the unitarity sum. The McFarlane group detected pion decays in flight using an intense pion beam (2×10^8/s). This helped to reduce the background contamination from positrons due to the Michel decay of the muon at the cost of a low detector acceptance for γ pairs from π^0 which is the signature of pion beta decay. For calibration purposes, they inserted either a liquid hydrogen target or a CH_2 target near the pion decay region to obtain the energy scale, efficiency of conversion and absolute timing of their apparatus by detecting monoenergetic π^0 s from either single charge exchange (SCX) or $\pi^+ + \mathrm{C} \to \pi^0 + \mathrm{X}$ ($\pi^- + \mathrm{C} \to \pi^0 + \mathrm{X}'$), respectively. From the averaged counting rate of three beam monitors the total number of pions was determined.

The high uncertainties associated with the previous experiments are due to both a low total number of pion beta decay events detected and to the imprecise determination of the detector acceptance, efficiency and limited precision of counting the number of decaying pions. A measurement of the pion beta decay branching ratio with higher precision requires both an intense pion beam which the accelerator in the Paul Scherrer Institute, Switzerland, offers and high detection efficiency and high acceptance that our custom designed PIBETA detector can achieve.

1.3 Waveform digitizing system in the PIBETA experiment

The detector waveform digitizing system based on the domino sampling chip (DSC) is introduced to the PIBETA experiment in order to achieve better detector timing and energy resolution and thus reduce the error of final result.

The work described in this thesis is a supplement of previous published PIBETA experiment data analysis based on analog-to-digital converter (ADC) and time-to-digital converter (TDC) electronic units. This work is the DSC analog of the previous analysis and uses the same PIBETA branch ratio calculation formula (just replacing each variable in the formula with corresponding DSC version). Comparisons with ADC/TDC data are made when applicable during the whole analysis.

Chapter 2

Pion beta decay theory

2.1 Pion decay in the quark model

2.1.1 Pions: the light pseudoscalar mesons of quark-antiquark combinations

Since 1960's, with great accumulation of data on baryon and meson resonances, patterns or multiplets of hadron states could be simply accounted for in terms of quark constituents. A baryon consists of three quarks and a meson of a quark-antiquark pair. With only u and d quarks and antiquarks we can make $2^2 = 4$ isospin combinations which are shown in Table 2.1. The four combinations consist of isospin $I = 0$ singlet η and $I = 1$ triplet, identified with the π^+, π^0, π^-, provided quark-antiquark combinations are in the $l = 0$ singlet spin state. They are called pseudoscalar mesons

Table 2.1: pseudoscalar mesons

I	I_3	Wavefunction	Q/e
1	1	$u\overline{d} = \pi^+$	+1
1	-1	$-\overline{u}d = \pi^-$	-1
1	0	$\frac{1}{\sqrt{2}}(d\overline{d} - u\overline{u}) = \pi^0$	0
0	0	$\frac{1}{\sqrt{2}}(d\overline{d} + u\overline{u}) = \eta$	0

with quantum numbers $J^P = 0^-$.

2.1.2 Weak interaction and pion decay

The weak interaction theory was first developed by Fermi in 1932 in explaining β decay. When quark model of hardrons was established, the weak interactions of quarks were best understood in terms of two ideas: *lepton-quark symmetry*, and *quark mixing*. The basic processes of weak interactions involve $W^\pm$ which are emitted or absorbed by their constituent quarks.

Lepton-quark symmetry in its simplest form asserts that three generation of quarks

$$\begin{pmatrix} u \\ d \end{pmatrix} \text{and} \begin{pmatrix} c \\ s \end{pmatrix} \text{and} \begin{pmatrix} t \\ b \end{pmatrix}$$

and three generations of leptons

$$\begin{pmatrix} \nu_e \\ \\ e^- \end{pmatrix} \text{and} \begin{pmatrix} \nu_\mu \\ \\ \mu^- \end{pmatrix} \text{and} \begin{pmatrix} \nu_\tau \\ \\ \tau^- \end{pmatrix}$$

have identical weak interactions. That is, one obtains the basic $W^\pm$-quark vertices by making the replacements $\nu_e \to u, e^- \to d, \nu_\mu \to c, \mu^- \to s, \nu_\tau \to t$ and $\tau^- \to b$ in the basic $W^\pm$-lepton vertices, leaving the coupling constant g_w unchanged. That is

$$g_{ud} = g_{cs} = g_{tb} = g_W.$$

This works quite well for many reactions, like the pion decay

$$\pi^+ \to \mu^+ + \nu_\mu$$

which corresponds to

$$u\overline{d} \to \mu^+ + \nu_\mu$$

However, many other decays which are experimentally observed are forbidden in this simple scheme. An example is the kaon decay

$$K^+ \to \mu^+ + \nu_\mu$$

which corresponds to

$$\overline{s}u \to \mu^+ + \nu_\mu$$

This decay includes a forbiden 'usW' vertex which could be incorporated by introducing an hypothesis due to Cabibbo called 'quark mixing'. According to this idea, the d and s quarks participate in the weak interactions via the linear combinations

$$d' = dV_{ud} + sV_{us} + bV_{ub}$$

with unitarity condition

$$|V_{ud}|^2 + |V_{us}|^2 + |V_{ub}|^2 = 1$$

assumed. That is lepton-quark symmetry is assumed to apply to the doublets

$$\begin{pmatrix} u \\ d' \end{pmatrix} and \begin{pmatrix} c \\ s' \end{pmatrix} and \begin{pmatrix} t \\ b' \end{pmatrix}$$

For π^+ used in our experiment, its possible decay modes are:

Decay Mode	Branching Ratio
$\pi^+ \to \mu^+\nu$	$0.9998782 + -0.0000014$
$\pi^+ \to \mu^+\nu\ \gamma$	$\sim 10^{-4}$
$\pi^+ \to e^+\nu$	$\sim 10^{-4}$
$\pi^+ \to e^+\nu\gamma$	$\sim 10^{-7}$
$\pi^+ \to e^+\nu\ \nu\overline{\nu}$	$\sim 10^{-6}$
$\pi^+ \to \pi^0 e^+\nu$	$\sim 10^{-8}$
$\pi^+ \to e^+\nu\ e^+e^-$	$\sim 10^{-9}$

The leading order Feyman diagram for $\pi^+ \to \pi^0 e^+\nu$ analysed in this work is shown in Fig. 2.1.

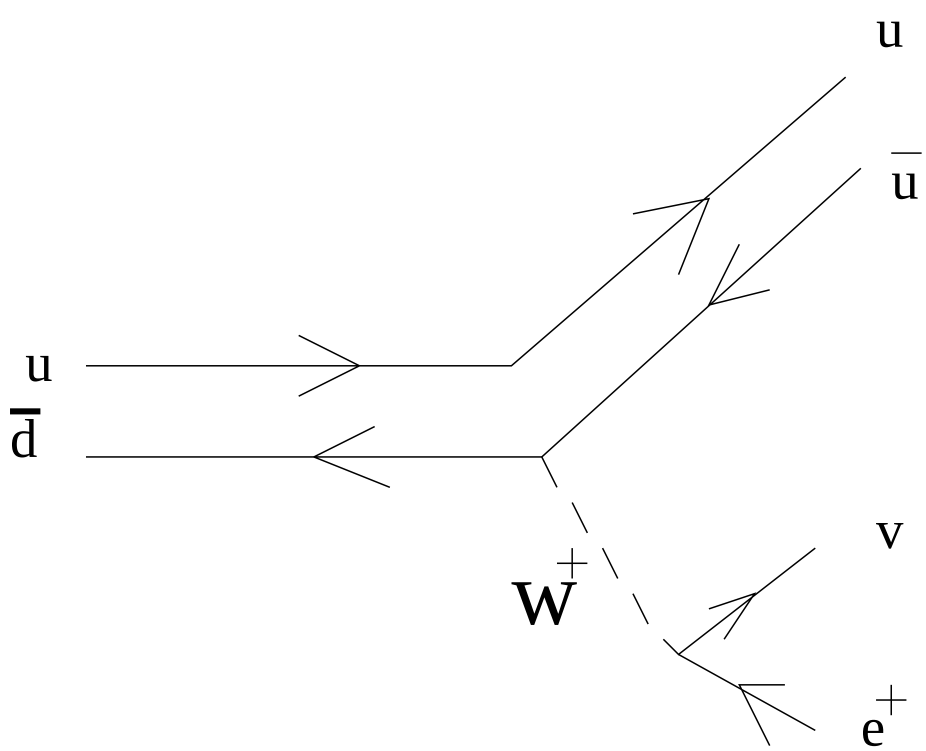


Figure 2.1: One typical Feynman diagram of $\pi^+ \rightarrow \pi^0 e^+ \nu$.

2.2 $\pi^+ \to \pi^0 e^+ \nu$ decay rate and radiative correction

The general expression for the invariant amplitude of weak interaction 2.1 is formulated as:

$$A + B \to C + D \tag{2.1}$$

$$\mathcal{M} = \left(\frac{g}{\sqrt{2}} \overline{\mu}_C \gamma^\mu \frac{1}{2} \left(1 - \gamma^5 \right) \mu_A \right) \frac{1}{M^2 - q^2} \left(\frac{g}{\sqrt{2}} \overline{\mu}_D \gamma_\mu \frac{1}{2} \left(1 - \gamma^5 \right) \mu_B \right). \tag{2.2}$$

in which M is the mass of $W^\pm$ (about 81 GeV) boson and q is the momentum carried by such boson. $g/\sqrt{2}$ is a dimensionless weak coupling constant. In most situations (including our experiment), the momentum q is very small relative to the mass of the W and Eq. 2.2 reduces to Eq. 2.3

$$\begin{aligned} \mathcal{M} &= \frac{4G}{\sqrt{2}} J^\mu J_\mu, \\ &= \frac{G}{\sqrt{2}} \left[\overline{\mu}_C \gamma^\mu \left(1 - \gamma^5 \right) \mu_A \right] \left[\overline{\mu}_D \gamma_\mu \left(1 - \gamma^5 \right) \mu_B \right]. \end{aligned} \tag{2.3}$$

with

$$\frac{G}{\sqrt{2}} = \frac{g^2}{8M^2}, \tag{2.4}$$

and is essentially point interaction for the weak currents.

Specific to decay

$$\pi^+(p_+) \to \pi^0(p_0) e^+(p_e) \nu_e(p_\nu) \tag{2.5}$$

the invariant amplitude reads

$\mathcal{M} = G_F V_{ud}^* l^\mu [f_+^{(0)}(t)(p_+ + p_0)_\mu + f_-^{(0)}(t)(p_+ - p_0)_\mu]$ where $l^\mu = \overline{u}(p_\nu)\gamma^\mu(1-\gamma_5)v(p_e)$. The expression in parentheses corresponds to the matrix element $\langle \pi^0(p_0)|\overline{u}\gamma_\mu d|\pi^+(p_+)\rangle/\sqrt{2}$. The hardronic form factors depend on the single variable $t = (p_+ - p_0)^2$.

The detailed derivation of decay rate is given in [4] and [18, 19], without radiative correction reads

$$\frac{1}{\tau_0} = \frac{G_\mu^2 \left|V_{ud}\right|^2}{30\pi^3} \left(1 - \frac{\Delta}{2M_{\pi^+}}\right)^3 \Delta^5 f\left(\epsilon, \Delta\right), \tag{2.6}$$

and

$$\begin{aligned} f\left(\epsilon, \Delta\right) &= \sqrt{1-\epsilon}\left(1 - \frac{9}{2}\epsilon - 4\epsilon^2\right) \\ &+ \frac{15}{2}\epsilon^2 \ln\frac{1+\sqrt{1-\epsilon}}{\sqrt{\epsilon}} - \frac{3}{7}\frac{\Delta^2}{\left(M_{\pi^+} + M_{\pi^0}\right)^2}. \end{aligned} \tag{2.7}$$

in which G_μ is the Fermi weak coupling constant that can be determined in muon decay. Under the conserved vector current (CVC) hypothesis, $G_V = G_\mu V_{ud}$ obtained from nuclear $0^+ \to 0^+$ β-decay which could be applied to pion β-decay. G_V has a value of $G_V/(\hbar c)^3 = 1.1136 \pm 0.0006\,\mathrm{GeV}^{-2}$ [5]. $\Delta = m_{\pi^+} - m_{\pi^0} = 4.5936 \pm 0.0005\,\mathrm{MeV}$ from PDG04 [3], $M_{\pi^0} = 134.9766 \pm 0.0006\,\mathrm{MeV}$, $M_{\pi^+} = 139.57018 \pm 0.00035\,\mathrm{MeV}$, $\epsilon = m_e^2/\Delta^2$. Thus the numerical value of the branching ratio is calculated to be

$$R = \frac{\tau_{\pi^+}}{\tau_0} = (1.0048 \pm 0.0012) \times 10^{-8}. \tag{2.8}$$

Therefore, the CVC hypothesis can be tested by the measurement of the $\pi^+ \to \pi^0 e^+ \nu$ decay branching ratio. However it will not be the whole story if the agreement is good. In addition to $\pi^+ \rightarrow \pi^0 e^+ \nu$, CVC theory predicts another possibility that the π-meson forms a virtual nucleon-antinucleon pair since π-mesons are strongly coupled to the nucleons. The virtual nucleon can decay through the normal β-interaction and changes its own charge while the antinucleon annihilates with the remaining nucleon with the emission of a π-meson. The charge of the final π-meson is different from that of initial meson due to β emission. Such anomalous decay rate is essentially determined by the β decay rate of the virtual nucleon and will give the lifetime of the anomalous decay that is equivalent to what was just developed. Therefore the possible $\pi^+ \rightarrow \pi^0 e^+ \nu$ decay with its order of magnitude of lifetime are not enough to confirm the validity of the CVC hypothesis. The special feature of this formalism is rather the exact numerical value (Eq. 2.8) of the branching ratio. Therefore, experiments with more precise measurement (of branching ratio) are needed to confirm the conserved current hypothesis.

Radiative Correction

Radiative correction is not included in Eq. 2.8. One can refer to the nuclear independent radiative corrections to $0^+ \rightarrow 0^+$ transitions in nuclear β-decay to calculate the radiative correction. Restricted to $O(\alpha)$, strong interaction effects in nuclear β-decay can be neglected in the correction calculation. The radiative correction function is based on a function $g(E, E_m, m)$ that was derived by Sirlin [17] and has the form

$$g(E, E_m, m) \;=\; 3\ln\left(\frac{m_p}{m}\right) - \frac{3}{4} + 4\left[\frac{\tanh^{-1}\beta}{\beta} - 1\right] \times$$

$$\times \quad \left[\frac{(E_m - E)}{3E} - \frac{3}{2} + \ln\frac{2(E_m - E)}{m}\right] + \frac{4}{\beta}L\left(\frac{2\beta}{1+\beta}\right)$$
$$+ \quad \frac{1}{\beta}\tanh^{-1}\beta\left[2\left(1+\beta^2\right) + \frac{(E_m - E)^2}{6E^2} - 4\tanh^{-1}\beta\right]. \tag{2.9}$$

where E_m is the electron end-point energy, m is the electron mass, $\beta = p/E$, p is the electron momentum, and $L(x)$ is the Spence function:

$$L(x) = \int_0^x \frac{\ln(1-t)}{t}dt \tag{2.10}$$

when applied to the pion beta decay and averaged over the electron energy spectrum, the function $\overline{g}(E_m, m)$ takes the form [20]

$$\overline{g}(E_m, m) = \frac{\int_{m_e}^{E_m} \frac{(E_m-E)^2 p_e E}{\left[1+\frac{2m_+}{m_0}(E_m-E)\right]} g(E, E_m, m)\, dE}{\int_{m_e}^{E_m} \frac{(E_m-E)^2 p_e E}{\left[1+\frac{2m_+}{m_0}(E_m-E)\right]} dE}, \tag{2.11}$$

where m_0 is the π^0 mass and m_+ is the π^+ mass. Numerical calculation gives $\overline{g}(E_m, m) = 8.9619$.

The radiative correction (δ_R) is derived by Marciano and Sirlin [22] and has the form

$$\left\{1 + \frac{\alpha}{2\pi}\left[\ln\left(\frac{m_p}{m_A}\right) + 2C\right] - \frac{\alpha(m_p)}{2\pi}\left[\overline{g}(E_m) + A_g\right]\right\} S(m_p, m_z), \tag{2.12}$$

Where $\alpha(\mu)$ is a QED running coupling which satisfies

$$\mu\frac{d}{d\mu}\alpha(\mu) = b_0\alpha^2(\mu) + \text{higher orders}, \tag{2.13}$$

such that $\alpha(0) = 1/137.089$ and $\alpha(m_p) = 1/133.93$. $S(m_p, m_z)$ is a QED short-distance enhancement factor equal to 1.02256, C is a nuclear structure-dependent correction which should vanishe for $0^+ \to 0^+$ transitions, and A_g is a small perturbative QCD correction estimated to be -0.34. m_A is a low energy cutoff applied to the short-distance part of the γW box diagram with the range from 400 MeV to 1600 MeV. Using this range of m_A, the radiative correction to rates of $0^+ \to 0^+$ transition is found to be in the range from 1.0324 to 1.0340.

The total decay rate $1/\tau_{\pi\beta}$ can be expressed as a product of the uncorrected expression, denoted by $1/\tau_0$, and an overall factor.

$$\frac{\tau_{\pi^+}}{\tau_{\pi\beta}} = \frac{\tau_{\pi^+}}{\tau_0}(1 + \delta_R), \tag{2.14}$$

Other calculations using different models yields essentially consistent results, e.g., Jaus [23] calculated the radiative correction applying a light-front quark model produces a final corrrection $\delta_R = (3.230 \pm 0.002) \times 10^{-2}$. Applying chiral perturbation theory, Cirgliano, et al find a theorectical uncertainty in the loop corrections to pion beta decay rate of less than 5×10^{-4} [47].

2.3 Kinematic variables

Values of kinematic variables that appear in the experiment are determined by simple kinematic 4-momentum calculations [36, 1].

$\pi^+ \to \pi^0 e^+ \nu$ ($\pi\beta$ decay)

$$\begin{aligned} 134.97 &\leq E_{\pi^0}\,(\mathrm{MeV}) \leq 135.05 \\ 0.511 &\leq E_e\,(\mathrm{MeV}) \leq 4.519 \\ 0.000 &\leq E_{\nu_e}\,(\mathrm{MeV}) \leq 4.023 \end{aligned}$$

π^0 possesses maximum kinetic energy of about 75 keV, which results in an opening angle spread between the two γ's from the π^0 decay. The maximum deviation from 180° is around 3.8°. This also results in a spread of γ energy which would be exactly half of the π^0 mass ($m_{\pi^0}/2 = 67.5\,\mathrm{MeV}$) when the π^0 decays at rest.

$\pi^+ \to \mathrm{e}^+\nu$ ($\pi 2e$ decay) This two-body decay gives $E_e = 69.273\,\mathrm{MeV}$ as π^+ decay at rest.

$\pi^+ \to \mu^+\nu_\mu$ This two body decay gives muon kinetic energy of $E^k_\mu = 4.118\,\mathrm{MeV}$. Almost all μ^+'s with this energy remain in the central PIBETA target.

$\mu^+ \to \mathrm{e}^+\nu\overline{\nu}$ (Michel decay)

$$
\begin{array}{ccccc}
0.511 & \leq & E_e\,(\mathrm{MeV}) & \leq & 52.830 \\
0.000 & \leq & E_{\nu_e}\,(\mathrm{MeV}) & \leq & 52.828 \\
0.000 & \leq & E_{\overline{\nu}_\mu}\,(\mathrm{MeV}) & \leq & 52.828
\end{array}
$$

The positron energy distribution E_e takes the form:

$$P(\epsilon)d\epsilon \sim \left\{1 - \epsilon - \frac{2}{9}\rho\,(3 - 4\epsilon)\right\}\epsilon^2 d\epsilon. \tag{2.15}$$

where $\epsilon = 2E_e/m_\mu$, ρ is fixed at 0.75 in the framework of the exact V–A structure of the weak charged current [18].

Chapter 3

Beamline and PIBETA Detector

3.1 Beamline

The PIBETA experiment was mounted and conducted at the Paul Scherer Institute (PSI) in Villigen, Switzerland. The π^+ beam was generated from a primary proton beam of energy of 590MeV produced by the isochronous separated sector cyclotron at PSI. The proton beam with intensity $\sim$ 1.5mA is transported through the primary proton channel to two target stations where pions and muons are generated. The π^+ beam is transported to the PSI $\pi E1$ area via a 16m long beam line carefully designed to supply intense pion beams of low energy and good momentum resolution. The accelerator operates at the frequency of 50.63 MHz producing a microscopic beam structure of about 1 ns wide proton pulses separated by 19.75 ns. A diagram of the beamline layout is shown in Fig. 3.1.

Pions are generated from the second target station E at an angle 8° relative to the incident proton beam line. Fig. 3.1 shows a sketch of the beam line fixed elements from the pion production target to the last pair of focusing magnets (embedded in

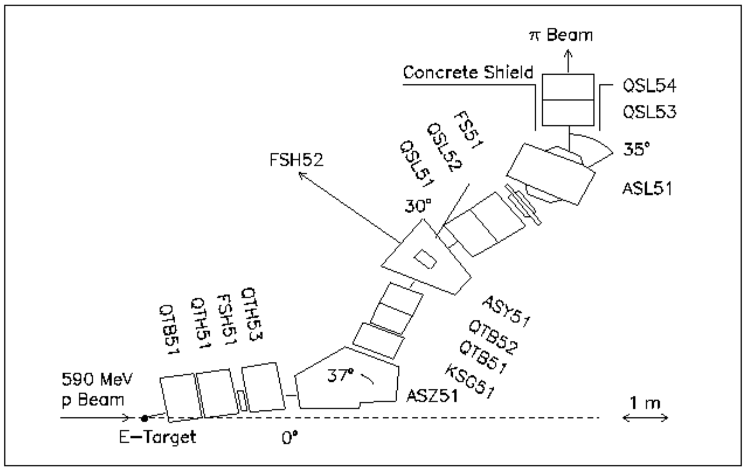


Figure 3.1: Beamline layout in the πE1 area at PSI.

the concrete shielding of the proton channel). Beam intensity is controlled by three slit systems incorporated in the channel through collimating the vertical size of the beam (by FSH51) and the beam halo (by FS51), hence improving the momentum resolution of the transported beam. Refer to [1] for a more detailed description.

Positron contamination from the decay of the π and μ in flight is reduced through a 4 mm thick carbon degrader inserted in the middle a ASY51 dipole magnet. Pions and positrons spatially separated in a horizontal plane due to their different energy losses in the carbon absorber.

Fig. 3.2 depicts a simplified layout of the PIBETA apparatus. A lead brick collimator with 7 mm diameter hole which restricts the spatial spread of the incident π^+ beam is located 3.985 m upstream of the detector center. A 3 mm thick plas-

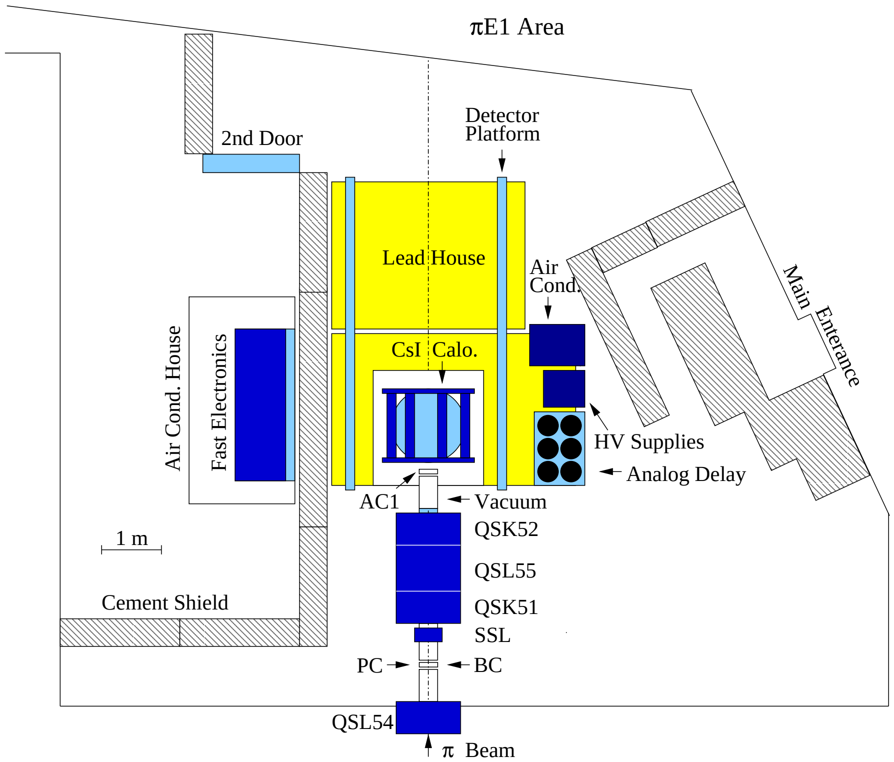


Figure 3.2: Physical layout of πE1 area in the PIBETA experiment.

tic scintillator beam counter (BC) placed immediately downstream of the collimator registers the beam particles first. The pions are subsequently slowed down through a 40 mm long active plastic scintillator degrader (AD) and stopped in an active plastic scintillator target (AT) positioned at the center of the PIBETA detector. Momentum spread at the front face of the degrader is about 1.2MeV/c/113.4MeV/c $\simeq$ 1.1%.

OPTIMA control program [26] was used to adjust the currents in the dipole and beam line quadrupole magnets that steer and focus the π^+ beam into the target. The goal was to obtain the smallest (concentrated), most symmetric beam spot that is consistent with the high π^+ beam intensity. The OPTIMA program allows a user

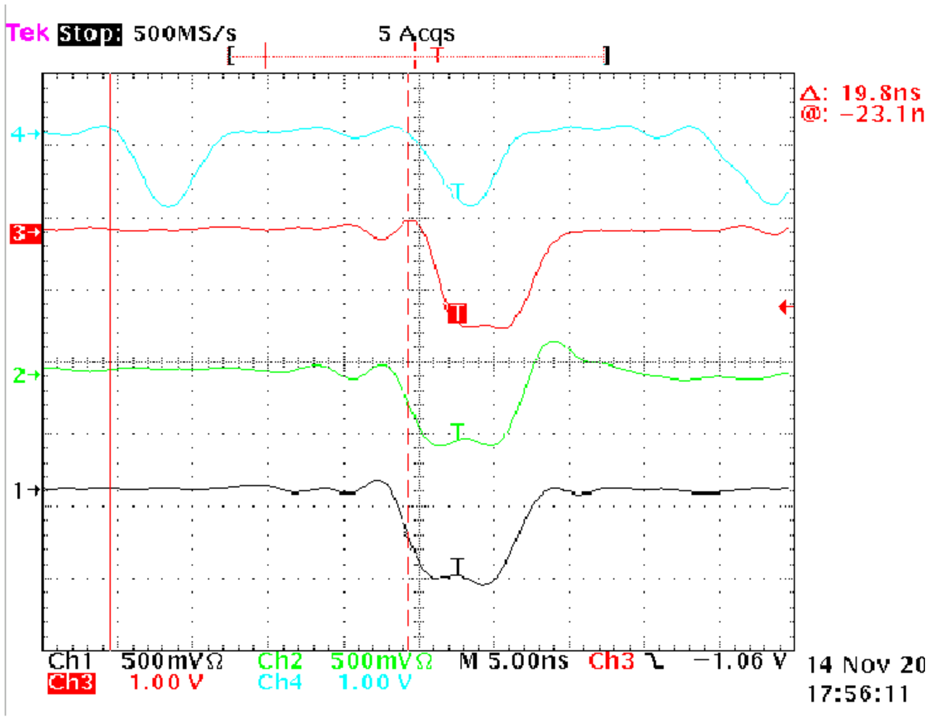


Figure 3.3: Relative timing of signals from the beam counter (BC, top), the target (AT, 2nd from top), the degrader (AD, 3rd from top) and the accelerator (rf, bottom), which defines the π^+-stop signal.

to maximize an experimental rate of choice (normalized to the primary cyclotron current) by iteratively changing the settings of the magnetic elements. We chose to maximize the beam rate of four-fold coincidences between the accelerator rf signal, the forward beam counter (BC), the degrader counter (AD), the active target (AT). A coincidence unit combines these four signals in such a way that their overlap signals a π^+particle stopping in the active target. A snapshot of the relative timing of the signals that form a π^+-stop trigger signal is shown in Fig. 3.3.

3.2 PIBETA detector

The cross section of the PIBETA detector is shown in Fig. 3.4. It consists of several components. The major part is the 240-CsI calorimeter covering $\sim 3\pi$ sr of solid angle. Inside the spherical calorimeter, there is a charged particle hodoscope made up of 20 plastic veto staves (PV) and two cylindrical Multi-Wire Proportional Chambers (MWPC). The active target (AT), made of plastic scintillator, locates at the center of the detector. An active degrader (AD), also made of plastic scintillator, locates right in front of the central target. Two sets of active collimator rings (AC), each having four segments, are in front of the degrader. The detector system also includes a beam counter (BC) that is located $\sim 3.8\,$m in front of the target. The entire detector setup (except BC) is enclosed in a 300 mm thick lead house, which is covered by an active cosmic muon veto consisting of five extensive scintillator planes on the top and its four sides.

3.2.1 Forward passive beam collimator

The lead collimator PC is located immediately upstream of the fixed beam-line's exit window, approximately at the focal plane between the last quadrupole doublet (QSL53/54) and the final beam focusing triplet (QSK51,QSL55,QSK52). This passive collimator consists of two stacked lead brick blocks, each with individual dimensions of $250 \times 250 \times 50$ mm^3, have a central hole with a diameter of 50 mm and a step bore that extends the hole to 70 mm. Only beam pions can pass through the collimator,

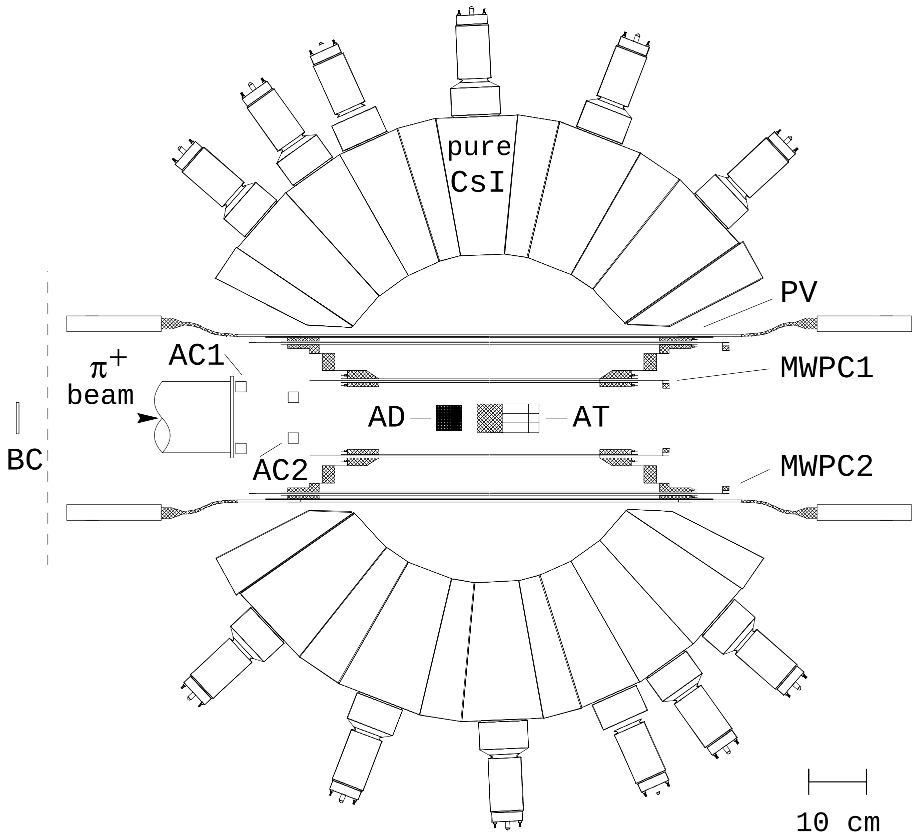


Figure 3.4: The cross section of the PIBETA detector system

while positrons (or electrons) stop in the collimator.

3.2.2 Forward beam counter

Placed immediately downstream of the lead collimator and just upstream of the vertical bending magnet SSL51, the forward beam counter BC is the first active detector counter that tags beam particles that pass through the collimator.

A rectangular piece of BICRON BC-400 plastic scintillator with dimensions 25 $\times$ 25 $\times$ 2 mm^3 make up the central part of the beam counter. This scintillator is optically coupled on all four edges to tapered acrylic light-guides. Each of the four-light guides is air-gap coupled to a Hamamatsu R7400U mini Photomultiplier tube.

3.2.3 Active beam collimator pair

Two active beam collimators AC_1 and AC_2 suppress backgrounds caused by detector hits that are not associated with a π-stop event but from beam-related particles that reach the PIBETA calorimeter accidentally in coincidence with the π-stop signal. Both collimators are cylindrical rings made of 25.4 mm thick BICRON BC-400 plastic scintillator. AC_1 is placed around the beam vacuum tube, about 550 mm upstream from the center of the detector. AC_2 is attached to aluminum ring mounting of the active degrader (AD) and covers its struts. It located 310 mm upstream of the center of the detector. Detailed description is given in Ref [2].

3.2.4 Active degrader

Pions reaching the central detector area have a relative high momentum, 113.4 MeV/c in order to maximize pion beam transmission along the whole length of the beam line. They are subsequently moderated (slowed down) by the active degrader AD. The degrader counter reduces the average pion kinetic energy from 40.3 MeV to 27.6 MeV. The value of this energy loss can be used to calibrate Digitized Waveform energy reading (DADC). Ref. [2] gives a detailed description.

3.2.5 Active PIBETA stopping target

Segmented into 9 elements, the active target is a cylindrical plastic scintillator counter with 50 mm length and 40 mm diameter. These 9 pieces are made optically isolated by wrapping them individually with an aluminized Mylar foil. The segments are pressed together and the whole assembly is wrapped using black plastic tape. Each target element is coupled to a miniature (8mm photo-cathode) Hamamatsu R7400U PMT via a tapered acrylic light guide of 60mm long.

A custom-made passive splitter is applied to divide the analog signal from each of the target segments into two branches. One side is discriminated in a 16-channel Phillips Scientific PS 7106 CAMAC time-over-threshold discriminator. This output is digitized by a FASTBUS time-to-digital converter (TDC) LeCroy Model 1877, as well as counted using a scaler unit (Phillips PS 7132H) that is read out every 10 ns. The second branch is connected to an 12-bit FASTBUS analog-to-digital converter (ADC) LeCroy Model 1882F, gated with a 100 ns wide event trigger gate.

3.2.6 Multi-wire proportional chambers

Multi-wire proportional chambers were introduced to the PIBETA experiment for charged particle tracking. Two such concentric cylindrical chambers were installed, each having one anode wire plane along the z direction, and two cathode strip planes in stereoscopic geometry. Specifically designed for the PIBETA experiment, they possess these features :

- high intrinsic efficiency, better than 99.9%;
- low mass, in order to minimize the γ's converting into e^+e^- pairs;
- stable operation and good radiation hardness.
- high rate capability, up to 10^7 minimum-ionizing particles (MIP) per second;

Refs. [32, 33] give a detailed description of the PIBETA MWPCs.

3.2.7 Plastic veto detector

The plastic veto hodoscope detector is placed around the two concentric wire chambers, in the interior of the calorimeter. The hodoscope consists of 20 independent plastic scintillator staves that are arranged to form a complete 98 mm long cylinder with a 132 mm inner radius. Each plastic stave is 3.2 mm thick. The PV covers the entire geometrical solid angle subtended by the CsI calorimeter as seen from the active target center. The rise and decay times of the fast scintillator pulses are 0.9 ns and 2.4 ns, respectively (ideal for triggering purpose). Ref. [34] contains detailed description of the plastic scintillator used.

Two readouts from either end of each PV stave were recorded. The energy deposited in each stave was calculated as the geometric mean of these two signals in order to cancel the light attenuation effects. Suppose the charged particle passing through a plastic veto stave generates initial light of intensity E_0 at a distance x from one end, then the two energy readouts from either end are:

$$E_1 = E_0 \exp\left(\frac{-x}{l}\right),$$
$$E_2 = E_0 \exp\left(\frac{-(L-x)}{l}\right), \tag{3.1}$$

where l is the attenuation length of our plastic scintillators with averaged value $396 \pm 13\,\mathrm{mm}$ [35], and L is the length of the plastic veto stave. Then, the geometric mean of these two readouts is

$$E_{mean} = \sqrt{E_1 E_2} = E_0 \exp\left(\frac{-L}{l}\right), \tag{3.2}$$

which is independent of the impact position where the initial light signal was generated. The energy calculated in this way for protons and positrons and is illustrated in Fig. 3.5. The energy resolution measured for minimum ionizing particles is $\sigma_E/E = 33.2\%$.

3.2.8 Modular pure CsI calorimeter

The CsI shower calorimeter is the heart of the PIBETA detector with its active volume made of pure Cesium Iodide [28, 29, 30]. Appendix C summarized the physical and optical properties of pure CsI. The CsI calorimeter (shown in Fig. 3.6 and Fig. 3.7) consists of 240 CsI crystals in nine different geometrical module shapes: one regular pentagon (PENT), two trapezohedrons (V1,V2), and four irregular hexagonal truncated pyramids (HEX-A,B,C,D). 220 HEX's two irregular half-hexagonal truncated pyramids (HEX-D1,D2) and two trapezohedrons (V1,V2). 220 PENT's and HEX's

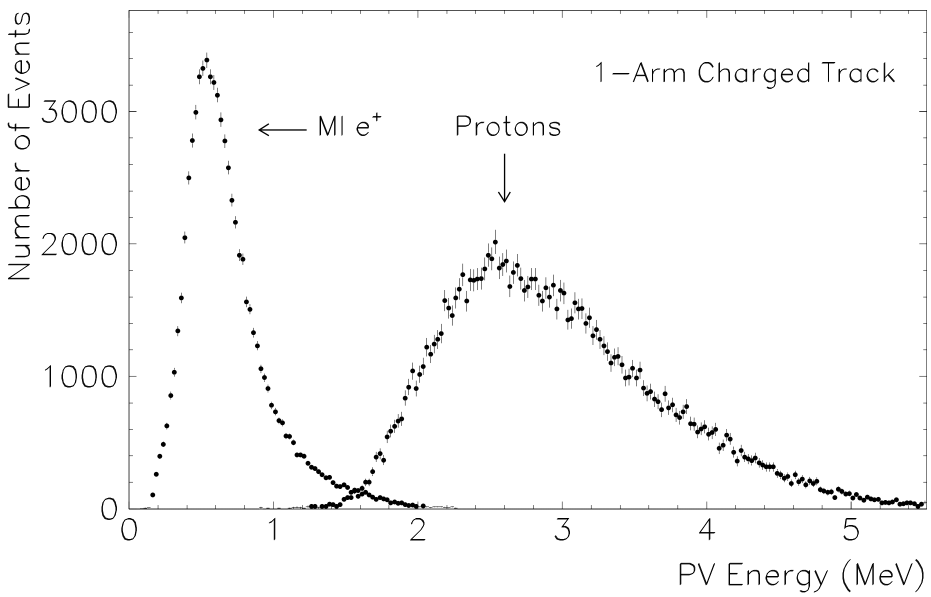


Figure 3.5: Energy response of PV detectors. Positrons and protons were selected separately and the energies deposited in the PV's were filled into histograms for each kind of particles.

cover a total solid angle of $0.77\times4\pi$. 20 CsI vetos cover two openings of detector sphere enabling the beam entry and exit and also act as electromagnetic shower leakage vetoes. The module axial length is 22 cm, corresponding to 12 CsI radiation lengths ($X_0 = 1.85\,\mathrm{cm}$) (see Appendix C), and the inscribed radius of the calorimeter sphere is 26 cm.

3.3 Geometrical features of photomultiplier waveforms for various detectors

We could have some idea of what the signal waveform look like on the screen of oscilloscope from beamcounter (3.8),degrader (3.9),and target (3.10). All these waveforms are reconstructed from Digitizer(DSC) data which for each signal is just 127 voltage

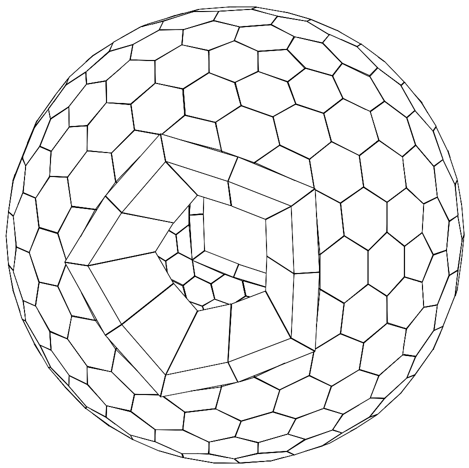


Figure 3.6: Spatial layout of the CsI crystals.

(amplitude) values sampled 127 equal spacing(time) points on the waveform.

3.4 PIBETA trigger generating scheme

An essential requirement on the detector system is that it produce selective, bias-free triggers capable of handling high event rates. Relevant trigger schemes are explained in this section. The detailed description of all implemented triggers in PIBETA experiment can be found in Ref. [35].

3.4.1 Signals used to generate 1-arm and 2-arm calorimeter triggers

A basic element of the trigger logic is the one-arm calorimeter energy signal. Calorimeter response to 70 MeV positrons from $\pi 2e$ decays and photons from $\pi\beta$ decay at rest can be used to make preliminary simulation study [36]. This simulation indicated that:

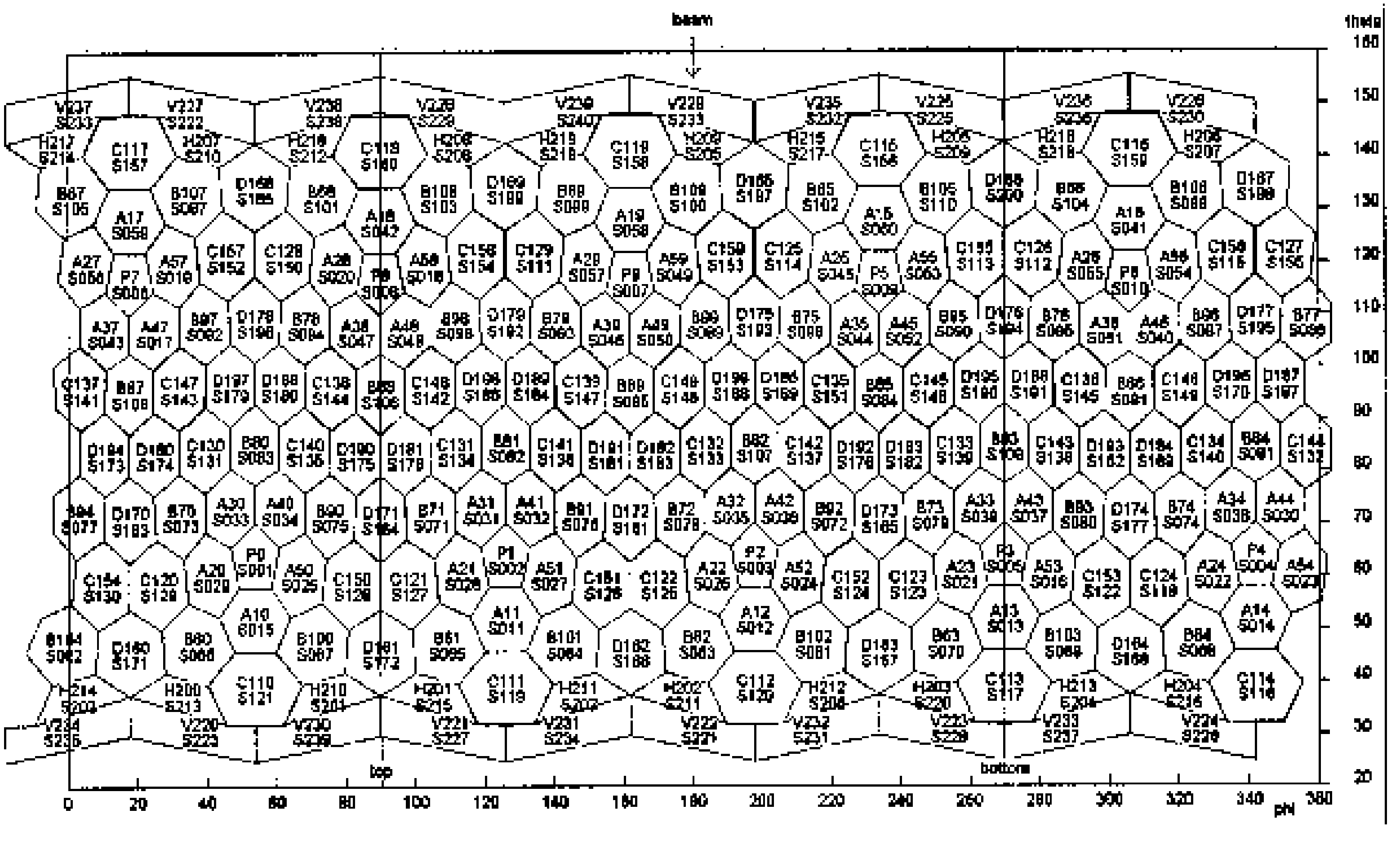


Figure 3.7: Mercator projection of CsI Calorimeter. In trigger generating scheme, these crystals were grouped into clusters and superclusters (see Section 3.4). For example, cluster 0 contains crystal 60, 200, 160, 120, 20, 10, 110, 0, 100. Supercluster 0 includes clusters 0, 10, 20, 30, 40, 50.

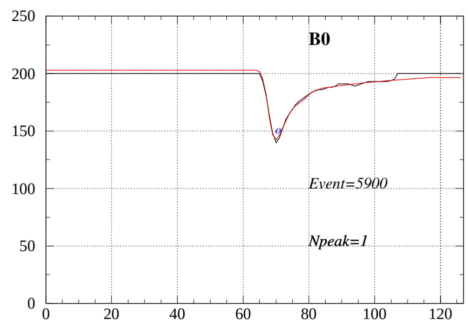


Figure 3.8: Waveform of the B0 pulse.

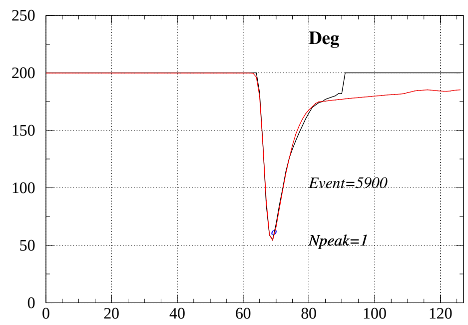


Figure 3.9: Waveform of the Degrader pulse.

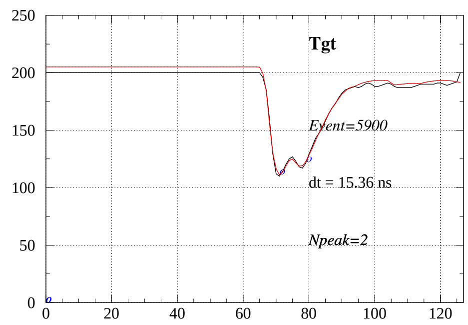


Figure 3.10: Waveform of the Target pulse.

- a centrally hit CsI module receives on average 90% of the deposited energy; at most three modules contain a significant part of the shower energy; and a group of 9 detectors (a CsI "cluster") constitutes an excellent summed energy trigger as it registers on average $\geq 98\%$ of the incident particle energy.

- electromagnetic shower profiles of the mean deposited energies are similar for photons and positrons, in particular for $\theta_c \leq 12°$, with θ_c being the half angle of a conical bin concentric with the direction of an incident particle;

- the average deposited energy and the corresponding energy resolution of the calorimeter both reach saturation within a cone of 12° half-angle;

Thus CSI clusters are the building blocks of physics triggers (see Fig 3.7). We define 60 such clusters [35] with CsI veto detectors excluded from the scheme. For each CsI cluster, there is a symmetric partner in the antipodal calorimeter hemisphere. However there is a restriction that each CsI module belongs to no more than three clusters. This limitation helps to minimize the degradation of analog pulses from excessive signal splittings. By trigger design studies that looked at the energy recorded by a single cluster as a figure of merit, it was found that such clustering scheme, in conjunction with a discrimination threshold of 50 MeV, gives 98.6% and 99.3% triggering efficiency for 70 MeV positrons and photon, respectively.

A CsI "supercluster" is a group of six adjacent CsI clusters. There are ten such superclusters in the calorimeter. A supercluster fires if at least one of its constituent clusters fires. A cluster fires if the summed energy of its member modules is greater than the preset discriminator threshold. If at least one supercluster fires due to energy above high (low) threshold (around $\sim 50\,\mathrm{MeV}$) (about $\sim 5\,\mathrm{MeV}$), then we get a $\mathrm{CsI_{HI}}$ ($\mathrm{CsI_{LO}}$) trigger. If at least two opposite superclusters in the sphere fire, then we have $\mathrm{CsI^2_{HI}}$ (if over high threshold) or $\mathrm{CsI^2_{LO}}$ (if over low threshold) signal.

3.4.2 Triggers in the PIBETA experiment

The following table list the triggers generated in PIBETA experiment.

Trigger type	Definition
BEAM	$\mathrm{BC} \cdot \mathrm{AD} \cdot \mathrm{rf}$
PISTOP	$\mathrm{BEAM} \cdot \mathrm{AT}$
PIBETA HIGH	$\pi\mathrm{G} \cdot \mathrm{CsI^2_{LO}} \cdot \overline{\mathrm{BEAM}}$
PIENU HIGH	$\pi\mathrm{G_{ps}} \cdot \mathrm{CsI_{HI}} \cdot \overline{\mathrm{BEAM}}$
PIENU LOW	$\pi\mathrm{G_{ps}} \cdot \mathrm{CsI_{LO}} \cdot \overline{\mathrm{BEAM}}$
PROMPT	$\pi\mathrm{G_{ps}} \cdot \mathrm{CsI_{HI}}$
COSMIC	$\mathrm{CV_{ps}} \cdot \mathrm{CsI_{HI}} \cdot \overline{\mathrm{BEAM}}$

Here $\pi\mathrm{G}$ is the pion gate signal initiated by PISTOP, a 180 ns window, whose delay is adjusted to begin about 50 ns ahead of the pion stop time t_0. $\mathrm{CV_{ps}}$ is the prescaled cosmic veto scintillator signal. Not listed in the table is the RANDOM trigger. This trigger is generated by a small plastic scintillator counter placed outside of (away from) the detector lead house, parallel to the πE1 beamline area floor, and shielded from the experimental radiation. At about 1-2 Hz, it counts only cosmic muons and random background events with a stable counting rate and is independent of the beam. The Digitizer time scale calibration uses the random trigger and the fact that difference between two signal peak voltage time is fixed in one such trigger period.

Chapter 4

PIBETA detector waveform digitizing system-introduction

The Domino Sampling chip (DSC) is introduced to PIBETA experiment in order to achieve better energy resolution, background suppression, and especially better timing resolution. It is a significant addition to the electronics arsenal. Beginning mid-year 2000, with the latest version of analyzer, DSC is applied to every photo-tube in the detector.

The DSC was originally considered for a more advanced stage of the experiment. However, successful initial tests made the chip available sooner. The device is now fully integrated into the PIBETA apparatus and routinely operational. In addition to its other benefits, it is also fast (up to 1.2GHz) and inexpensive.

4.1 First look at DSC

Each DSC chip operates at about 500MHZ producing waveforms 256ns wide steps of about 2ns. The analog waveforms are pre-analyzed in DAQ front-end computer.

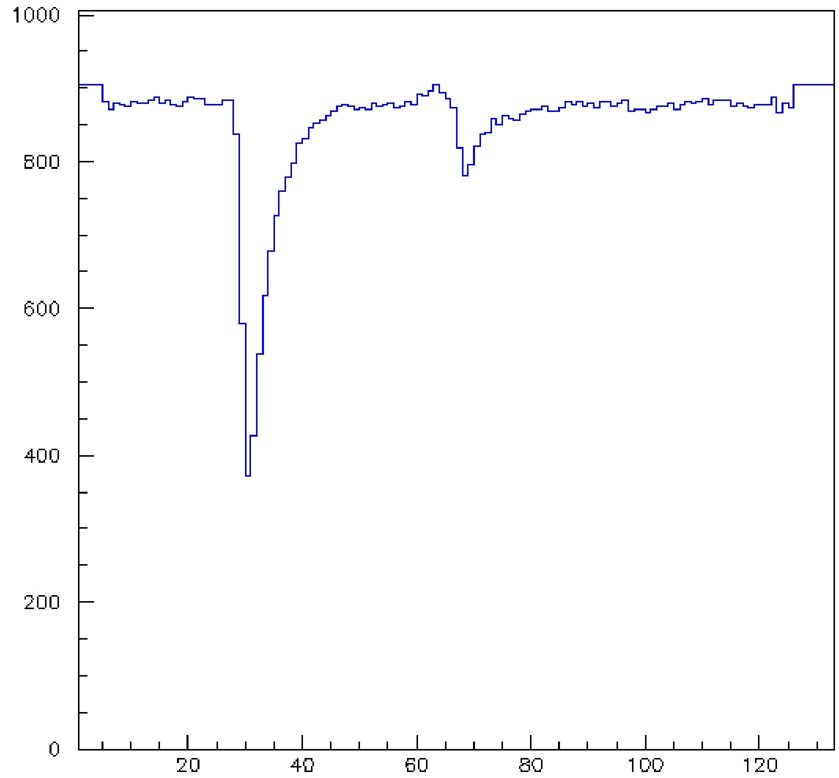


Figure 4.1: Raw DSC sampled waveform.

Zero suppression is applied in software to reduce the data rate by a factor of ten while retaining the interesting sections of the waveforms for offline analysis.

A typical raw DSC waveform is shown in Fig. 4.1.

It is reconstructed from 128 equal-spacing points sampled on original signal waveform. It includes background noise. This kind of Raw DSC is generated once every 20 or 30 compressed DSC (CDSC) and it is used to update system function (used for CDSC fitting) by averaging over lots of normalized RDSC waveforms.

A typical compressed DSC waveform is shown in Fig. 4.2 Compared with raw DSC, it is reconstructed from data points around signal peaks. The background

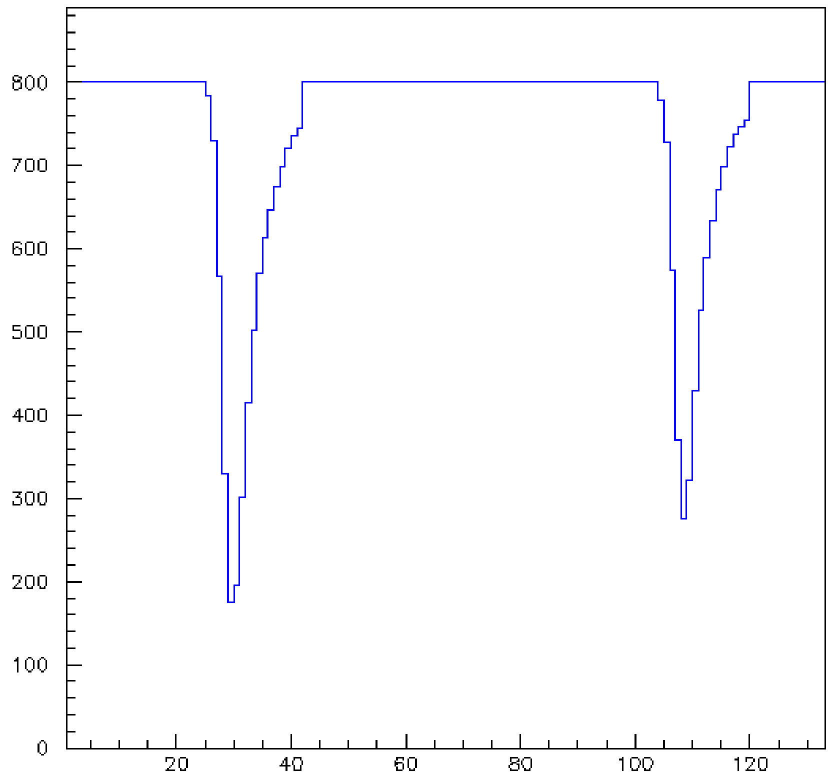


Figure 4.2: Raw DSC sampled waveform.

noise is suppressed as baseline (default value).

4.2 Summary of reasons of introducing waveform digitzer

Fig. 4.3 shows regular fastbus TDC reading that has signal energy amplitude dependence, even when signals coming at the same time. Digitizer timing obtained by fitting peak position does not have this kind of problem.

Also, multi-peak information in the signal is lost in the single value read from TDC or ADC. DSC analysis can easily visualize multi-peak information of waveform.

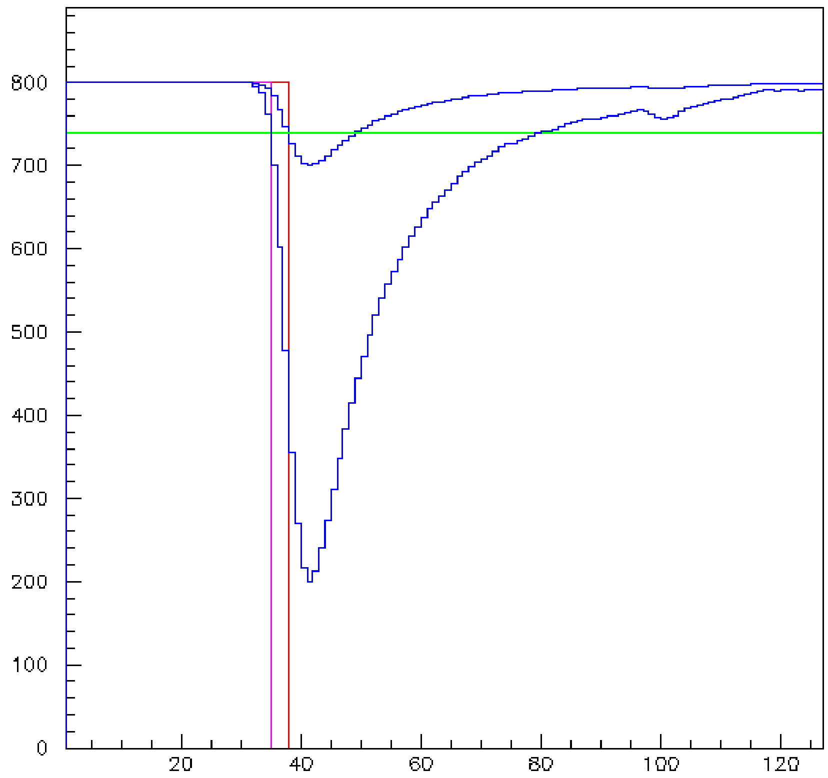


Figure 4.3: Amplitude dependence of timing reading.

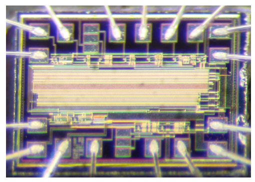


Figure 4.4: Real image of Domino Sampling Chip.

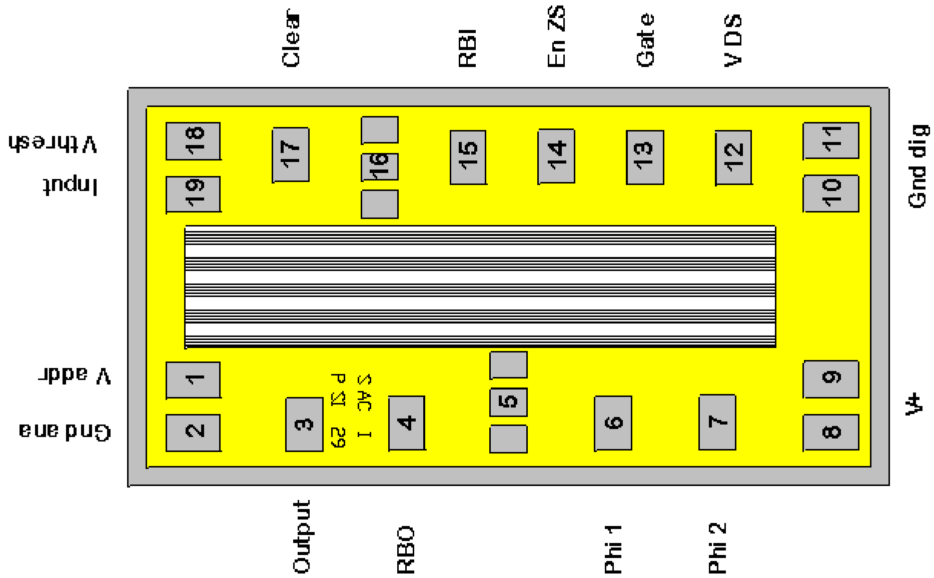


Figure 4.5: Domino Sampling Chip input and output connections.

4.3 DSC data generation-Hardware look

4.3.1 Domino Sampling Chip

Fig. 4.4 and 4.5 show a photograph of a real Domino Sampling Chip.

4.3.2 The Domino Sampling Chip carrier

Fig. 4.6 shows how the Domino Sampling Chip carrier looks like.

It is designed at PSI for PIBETA experiment and is ideal for many channel detectors with sampling speed $500-1200$MHZ, readout speed $500-1200$MHZ, readout

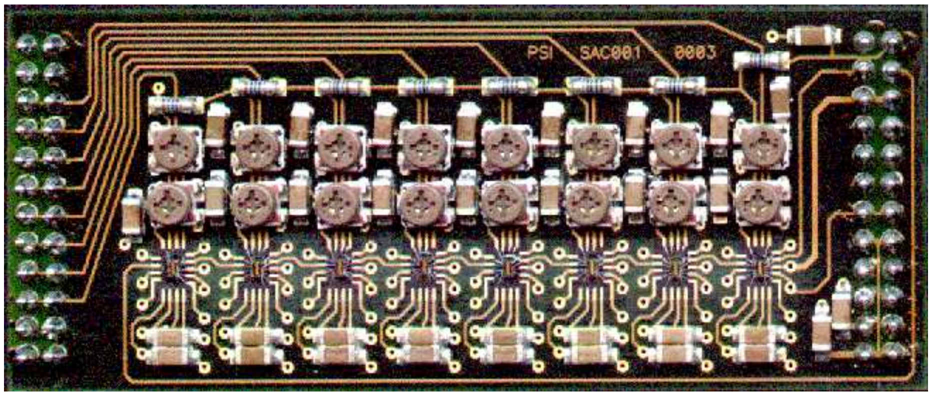


Figure 4.6: DSC carrier.

speed 5MHZ via flash ADC. They are daisy-chained in groups of 48 with zero suppression applied.

Fig. 4.7 and Fig. 4.8 show the DSC motherboard and its setup in experiment area.

4.4 DSC Operation principle

Fig. 4.9 gives the circuit diagram showing how the DSC works.

It includes following key points.

- A logic pulse is presented at GATE. When the pulse is at Cell i it opens S_i.
- The waveform to be sampled is at IN. Voltage at IN is stored as charge on C_i.
- Readout cycle is similar with clocked shift register.

4.5 DSC data processing-Software look

4.5.1 PMT waveform fitting

First step: Find DSC relative scale calibration. That is, find the ns per bin for each DSC channel, making sure that the scale ratio between any two channels is correct.

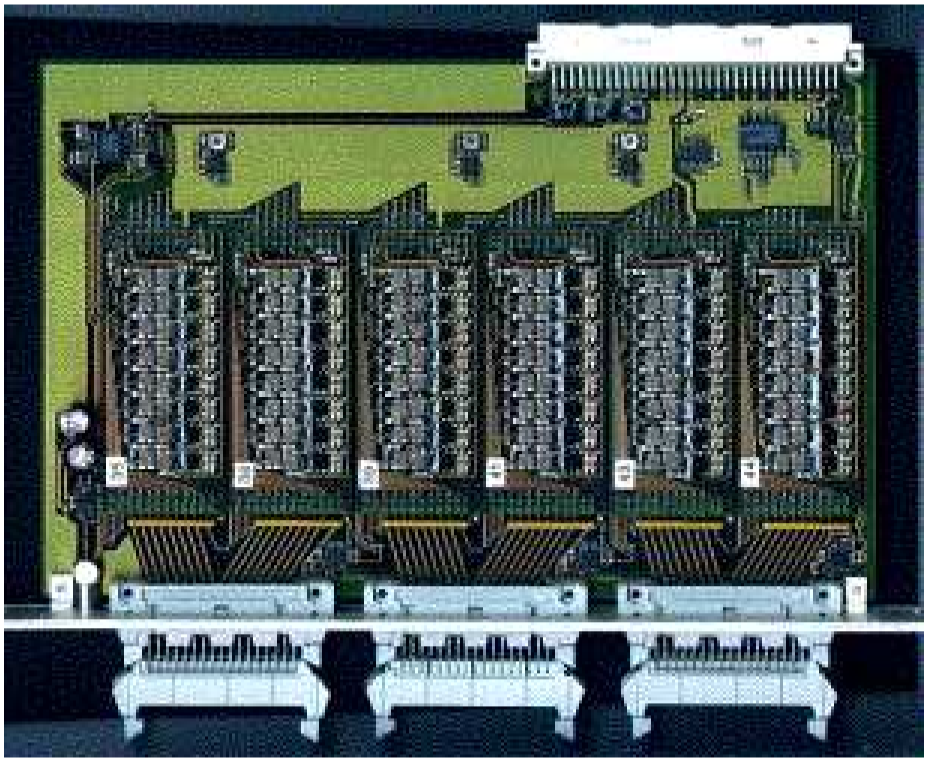


Figure 4.7: DSC motherboard.

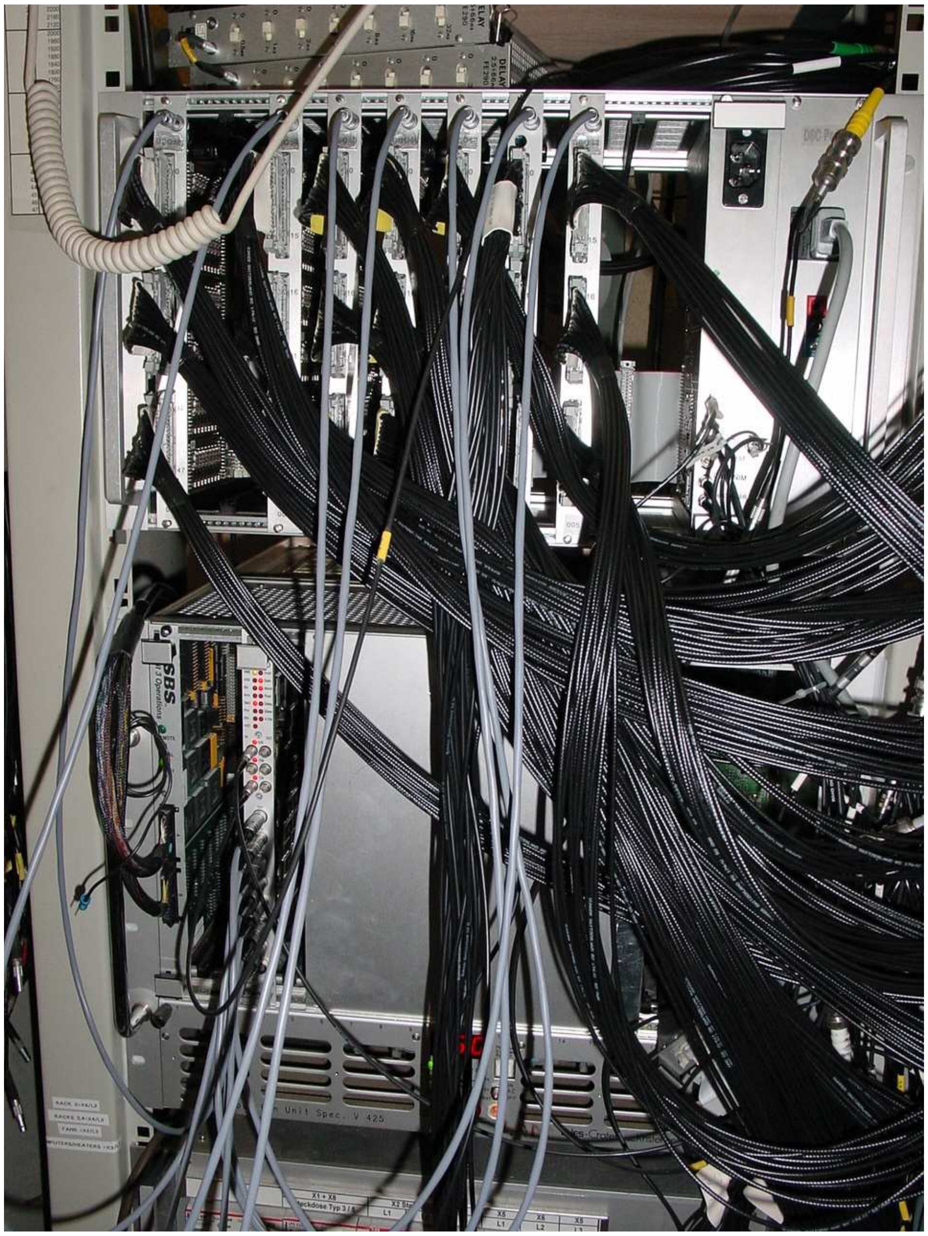


Figure 4.8: DSC in experiment.

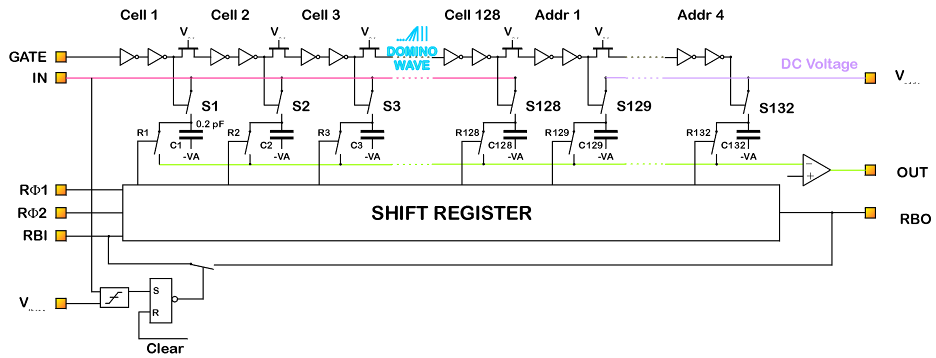


Figure 4.9: DSC working diagram.

This is accomplished using random trigger events, as actual timing difference between two random peaks is supposed to be fixed at 100 ns for any channel. For each channel, dividing 100ns by the number of bins between two random peaks ends with the scale calibration constant for that channel.

Second step: Finding overall calibration scale that is to be multiplied to each channel's calibration constant to make each channel's absolute calibration scale correct. This is accomplished by looking at the timing spectrum of prompt events (dominantly single charge reaction followed by $\pi^0 \to \gamma\gamma$). Fig. 4.10 shows the histogram filled using timing difference between CsI detector and degrader. The averaged spacing between adjacent peaks should correspond to 19.75 ns, the RF value from cyclotron frequency 50.63MHz. The period for our data is found to be 15.56 ns. So the overall calibration constant is 19.75/15.56.

Third step: Find timing offset for each channel, that is the time zero position to be subtracted from DSC time reading to get the correct time reading. This is accomplished by looking at each channel's timing spectrum of prompt events. Do

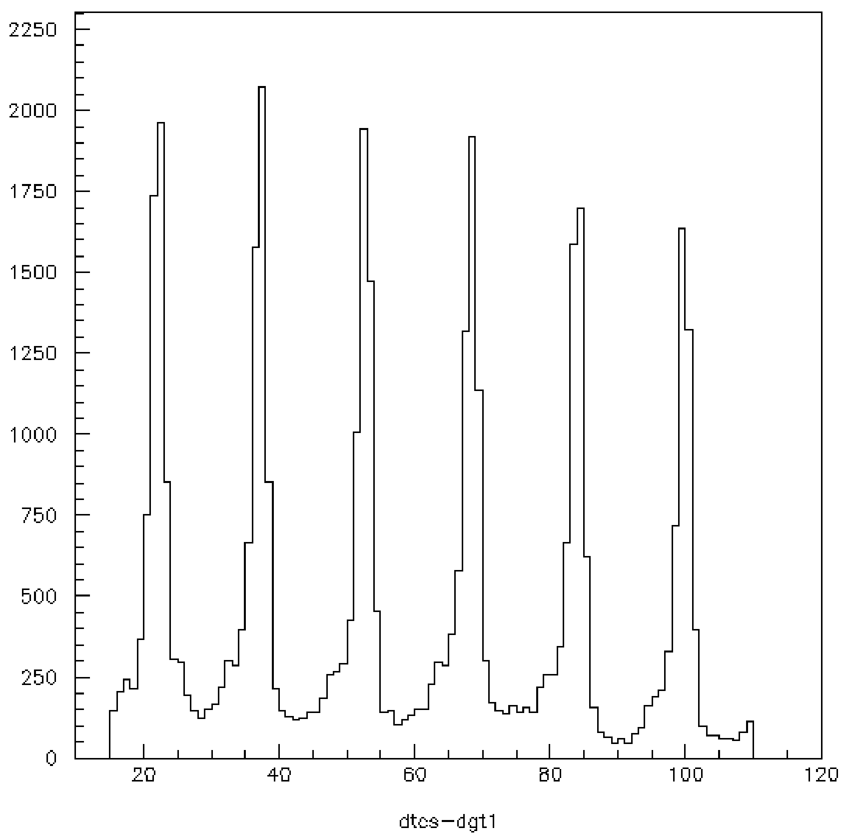


Figure 4.10: Prompt timing.

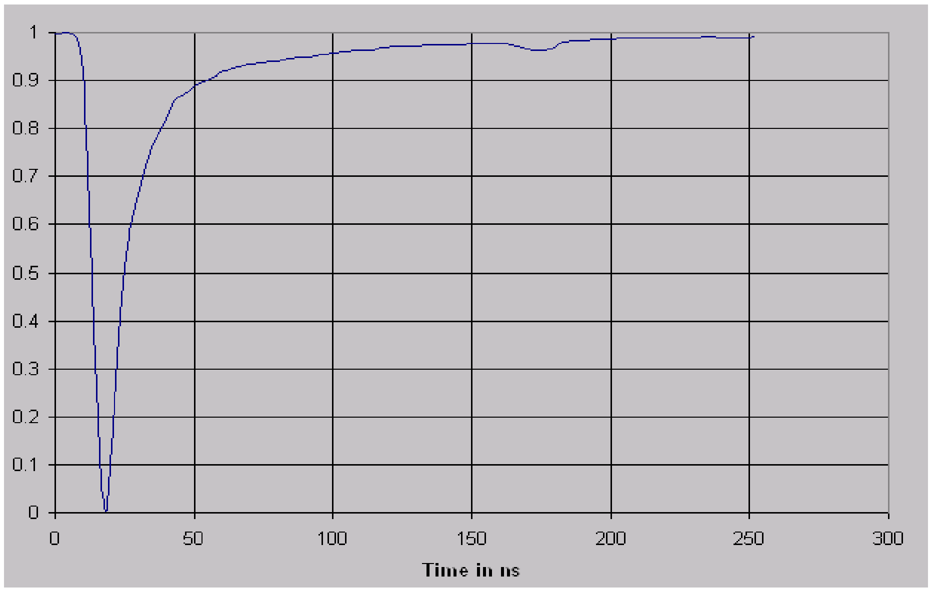


Figure 4.11: System function.

Gaussian fit around main timing peak. The peak position multiplied by that channel's calibration scale is the timing offset.

Fourth step: Generate system function for compressed DSC waveform fitting. Fig. 4.11 shows a typical system function derived by averaging over 10000 RDSC events which contained only one hit in that particular channel.

4.5.2 Waveform fitting

It includes the following key points.

- Assumption: π^+ and e^+ produce identical waveforms.
- Fit all signals in a channel with sums of its system function

$$f(t) = \sum_{i=1}^{N} A_i \cdot g(t - \tau_i)$$

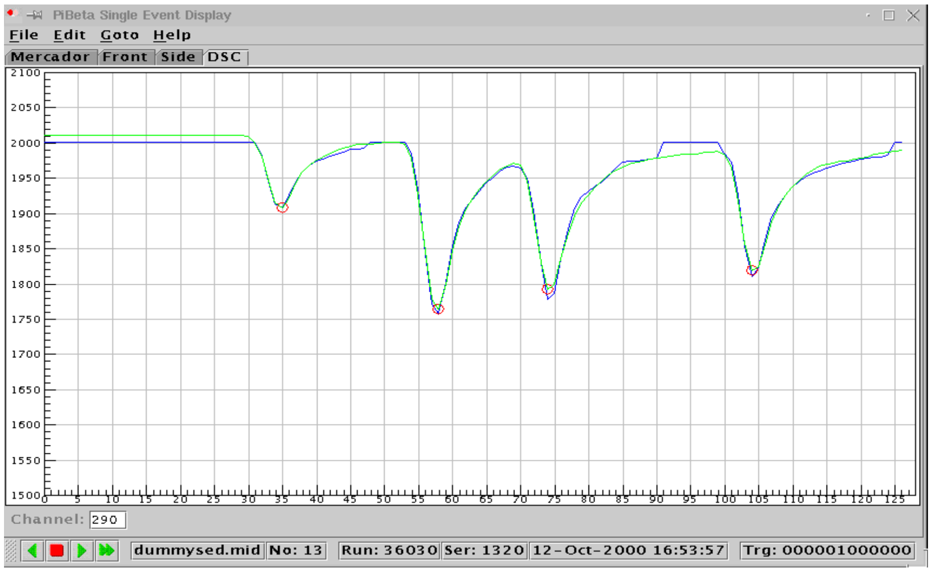


Figure 4.12: Waveform of degrader.

That is, find $\tau_i, A_i (i = 1, .., N)$ that minimize

$\chi^2 = \sum_{j=0}^{126} W_j \cdot [y_j - \sum_{i=1}^{N} A_i \cdot g(t_j - \tau_i)]^2$

4.5.3 Waveform fitting algorithm overview

First step: Determine the number of peaks (events) N in original waveform, selecting from the maximum amplitude peak to the smallest one with limit $N < 6$. For each peak, select about 5 data points around the peak to fit using single system funtion to get trial value of τ_i,A_i.

Second step: Put N trial values of τ_i,A_i into $f(t)$ and call routine that minimize chisqr to get final output value τ_i,A_i.

4.5.4 Some examples of good waveform fitting.

Fig. 4.12, 4.13 show some examples of and good fitts.

Figure 4.13: Waveform of target.

Chapter 5

Digitizing system performance evaluation-General

5.1 Improve DSC analysis efficiency-skimmed MIDAS file analysis method

When the execution of DSC computer code is included in analyzer, the analysis is 20 to 30 times slower compared with the case when it is switched off. Therefore the data on one computer node may take a whole month to analyze. So instead of analyzing original data file (MIDAS format file in our experiment), we work on skimmed midas file with file size (number of event unit) at least two order of magnitude less than original MIDAS file. We proceed as following: we first generate skimmed MIDAS file which contain only the candidate events we are interested in, such as PIBETA candidate events, $\pi^+ \to e^+\nu$ candidate events, etc. The list of candidate event identification numbers is obtained by filtering the original MIDAS files through cuts in analyzer with DSC function switched off. To skim the MIDAS files, we pick up the data block in original MIDAS file corresponding to the event identification number

Figure 5.1: MIDAS data block head format

bit0 bit31

event ID	trigger mask
serial number	
time stamp	
data size	

in the ID list from previous filtering process, we need to know the detailed structure of MIDAS file. Refer to Fig. 5.1, which give the structure of the 16 Byte event header of MIDAS file. The header contains key information of one event data block, most important of which are 'serial number' (event identification number) and 'data size' corresponding to the data contents follows the event header. The C program to generate skimmed MIDAS file is in the Appendix.

5.2 Optimizing the functioning of DSC

5.2.1 Optimizing by modifying system function

Refer to Fig. 5.2 which shows an unsatisfactory fitting caused by bad system function which maybe too old for most recent data. This kind of problem can be fixed by updating the system function constantly. Fig. 5.3 shows the improved fitting by applying an updated system function.

5.2.2 Recovering misfitted event peaks

Missed peaks can be recovered by adjusting amplitude threshold for finding peaks. And at the same time avoiding picking up background noise with two small threshold.

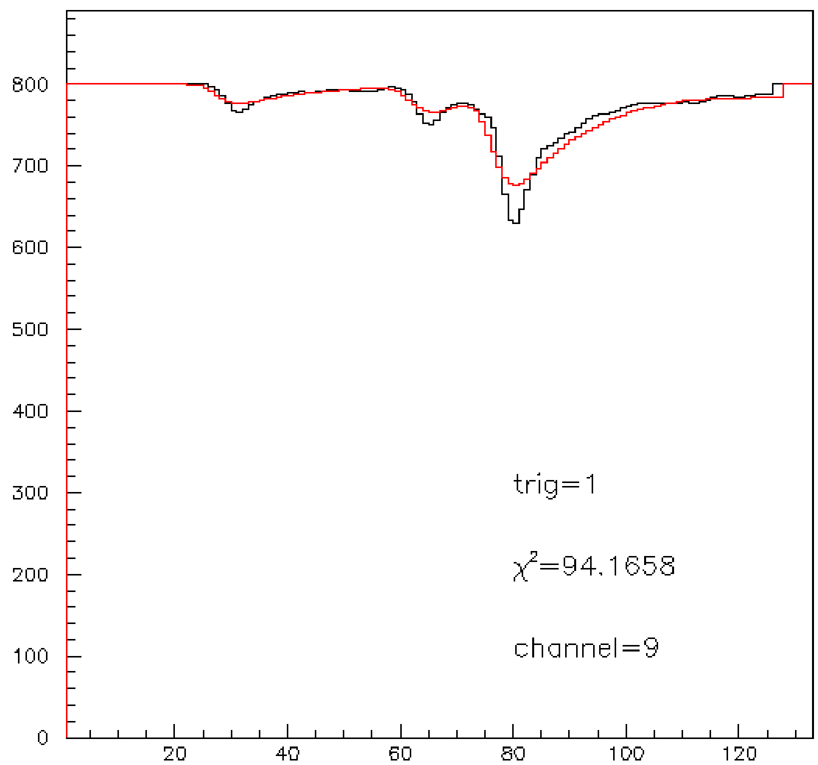


Figure 5.2: Fitting with bad system function.

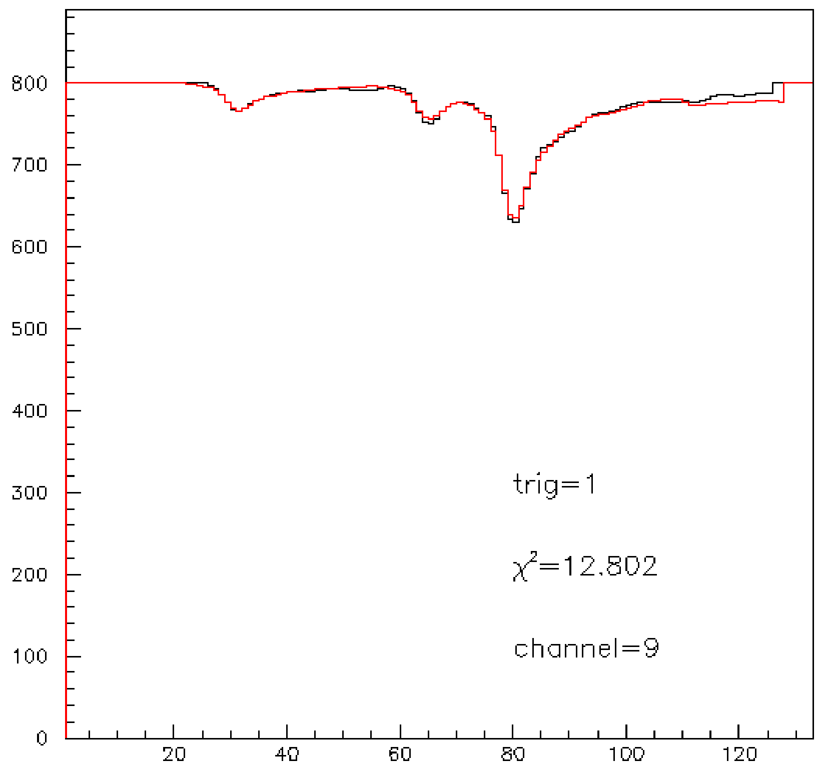


Figure 5.3: Fitting with improved system function.

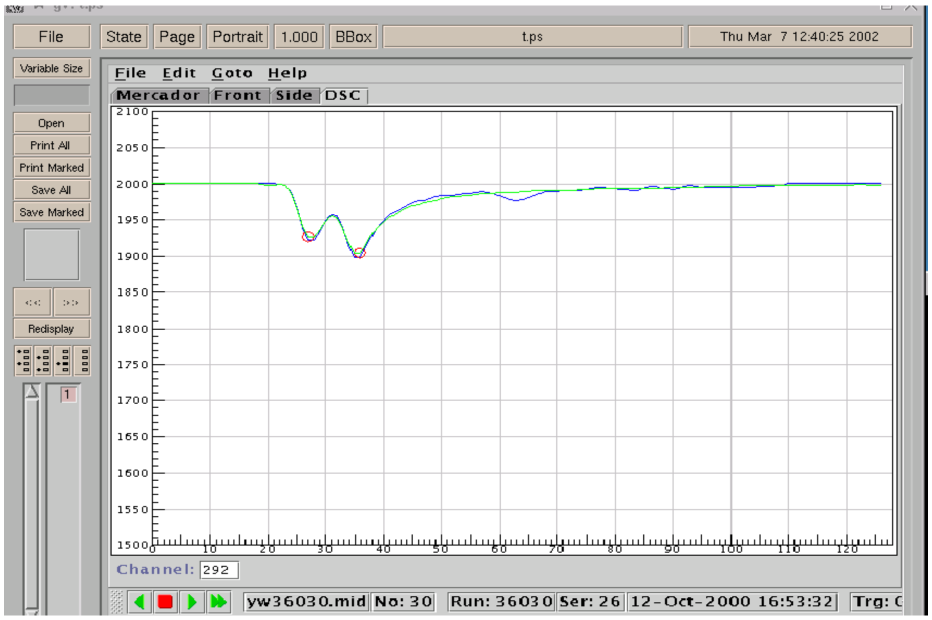


Figure 5.4: Fitting that misses a peak.

So we must decide on the choice of the best threshold cuts. Fig. 5.4 shows a DSC fitting that misses a significant peak which is recovered by lowering the threshold which is shown in Fig. 5.5.

5.3 DSC timing resolution compared with TDC timing.

Since DSC timing is obtained by fitting to get the signal peak position which is of double precision and has no signal amplitude dependence. It has significant advantages over TDC timing reading. Here are two examples.

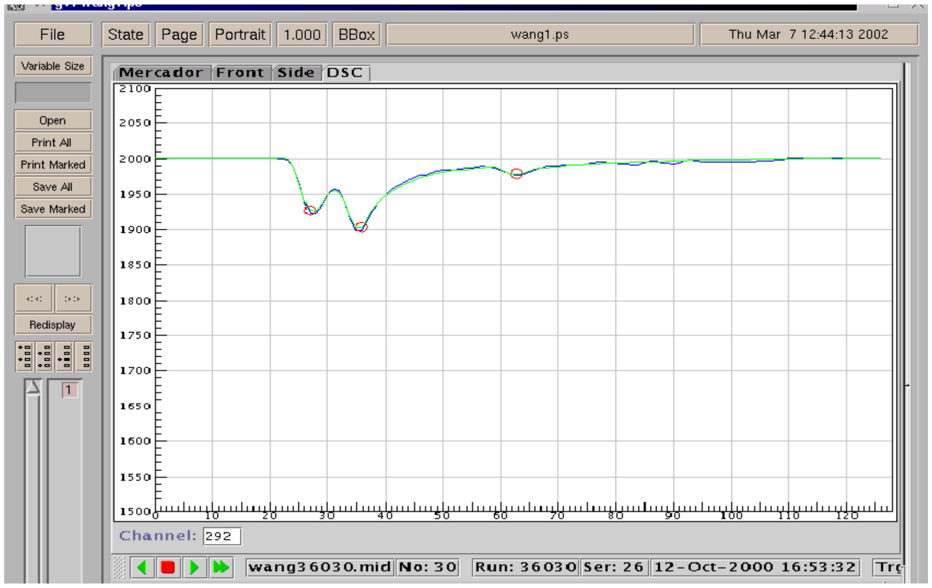


Figure 5.5: Fitting that recovers the missing peak.

5.3.1 DSC vs Lecroy LRS1877 Multihit TDC

Refer to Fig. 5.6. We can clearly see that FASTBUS spectrum (top one) is grainy due to LSB (least significant Bit) resolution and DSC timing distribution (bottom one) is significant sharper.

5.3.2 DSC timing vs TDC timing spectrum of timing difference of two gammas from $\pi^0 \rightarrow \gamma\gamma$ decay

Refer to Fig. 5.7. We can clearly see the DSC timing resolution (bottom one with sigma 0.80 ns) is significantly better than TDC timing resolution (top one with sigma 0.94 ns).

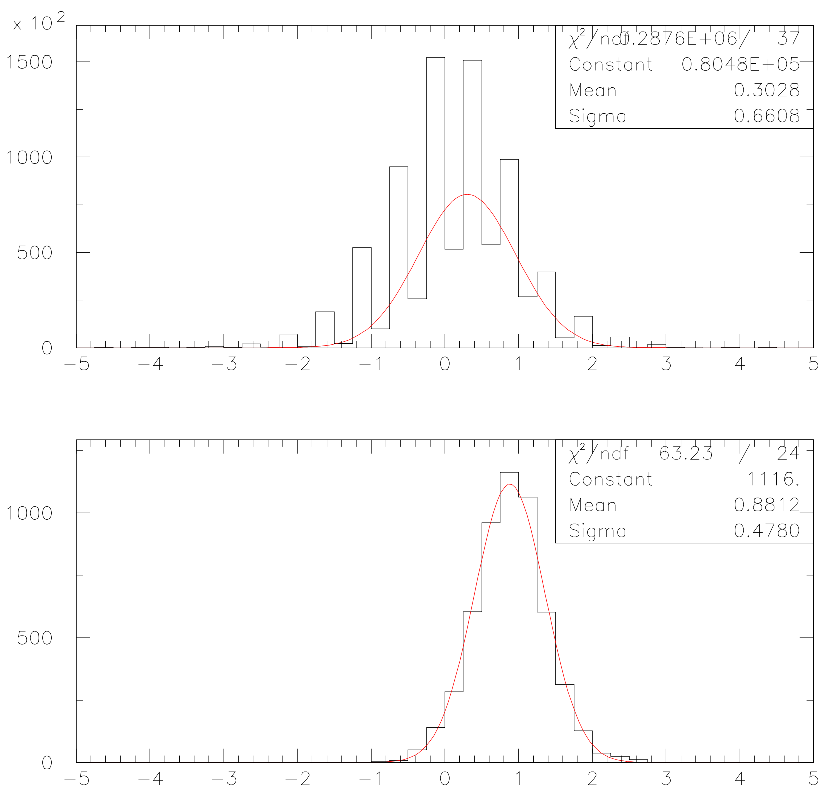


Figure 5.6: DSC vs Lecroy TDC.

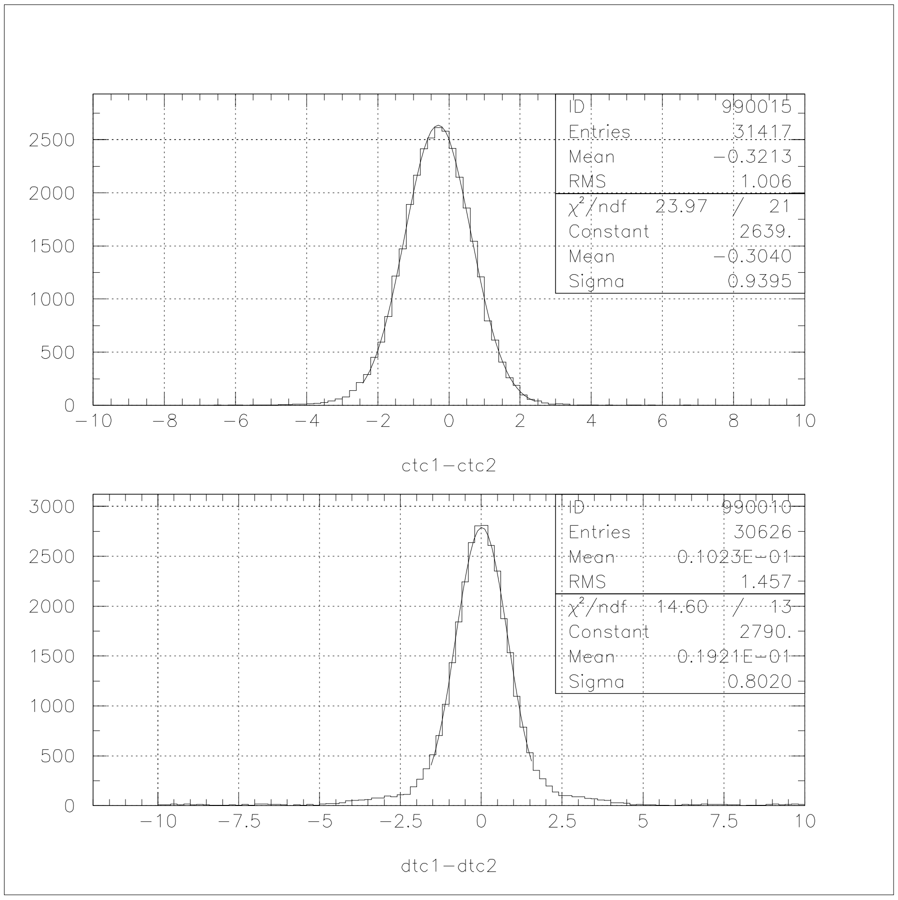


Figure 5.7: DSC vs TDC $\pi^0 \to \gamma\gamma$ decay.

5.4 DSC energy resolution (DADC) vs TDC energy resolution(CADC).

DSC energy is obtained from signal peak area (corresponds to amplitude fitting parameter) times a calibration constant. So good fitting could insure good DSC energy reading resolution.

ADC energy (CADC) is read from hardware called analog to digital converter (ADC) with its numerical value proportional to the area between signal curve and straight baseline. It can not resolve multi-event cases. So ADC reading may contain additional background contamination.

Here are two examples for comparison.

5.4.1 DSC vs Lecroy LRS1882 ADC for degrader energy spectrum.

Refer to Fig. 5.8. The resolution of charge (TDC) ADC is $\sigma/\overline{E} \sim 11\%$. For Domino Sampling Chip, $\sigma/\overline{E} \sim 8\%$ which gives better resolution for single channel (degrader in this case) energy reading.

5.4.2 DSC vs charge ADC of summed target energy spectrum.

For the 9-piece target, in addition to 9 channels which are assigned to nine individual target pieces, there are additional three channel assignment corresponding to central piece(target 0), middle ring (target1 + target2 + target3 + target4), outer

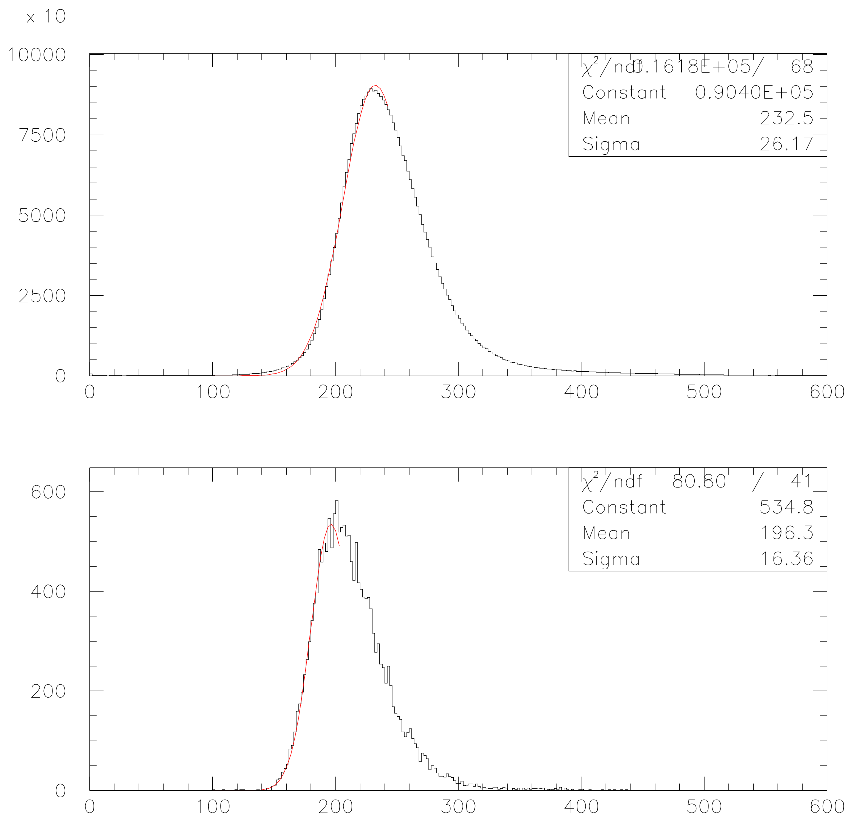


Figure 5.8: DSC vs LeCroy LRS 1882 charge ADC.

ring $(\text{target5} + \text{target6} + \text{target7} + \text{target8})$. Here we only investigate such three channel sum. Refer to Fig. 5.9. The top spectrum is the CAMAC target energy sum$(\text{badc}[2] + \text{badc}[3] + \text{badc}[4])$ has resolution $\sigma/\overline{E} \sim 12.34\%$. The bottom one is the calibrated DSC three channel target energy sum$(\text{dadc}[291] + \text{dadc}[292] + \text{dadc}[293])$ with resolution $\sigma/\overline{E} \sim 10.62\%$. We can see that DSC has improved energy resolution as well as improved timing resolution.

5.5 DSC's role in particle ID identification.

The PIBETA experiment has used the stopped π^+ method. As a π^+ stop in the target, it generates a signal . Its decay into secondary particles generates another signal. Hence, if the target DSC waveform has more than one significant peak, this may be due to π^+ decay . If the secondary decay product does not have enough energy to escape the target and also decays (μ^+ in our case), it will generate a third peak in the DSC waveform which will be candidate event for $\mu^+ \rightarrow e^+\nu\overline{\nu}$. Fig. 5.10 shows one example of three-peak event. If DSC waveform has two peaks and the amplitude of the second peak is significantly smaller than the first peak, it will be candidate event for $\pi^+ \rightarrow e^+\nu$ decay. Fig. 5.11 shows one example of two-peak event. Two peaks with similar amplitudes and spacing of multiples of 19.75 ns will be the signature of π^+ pileup.

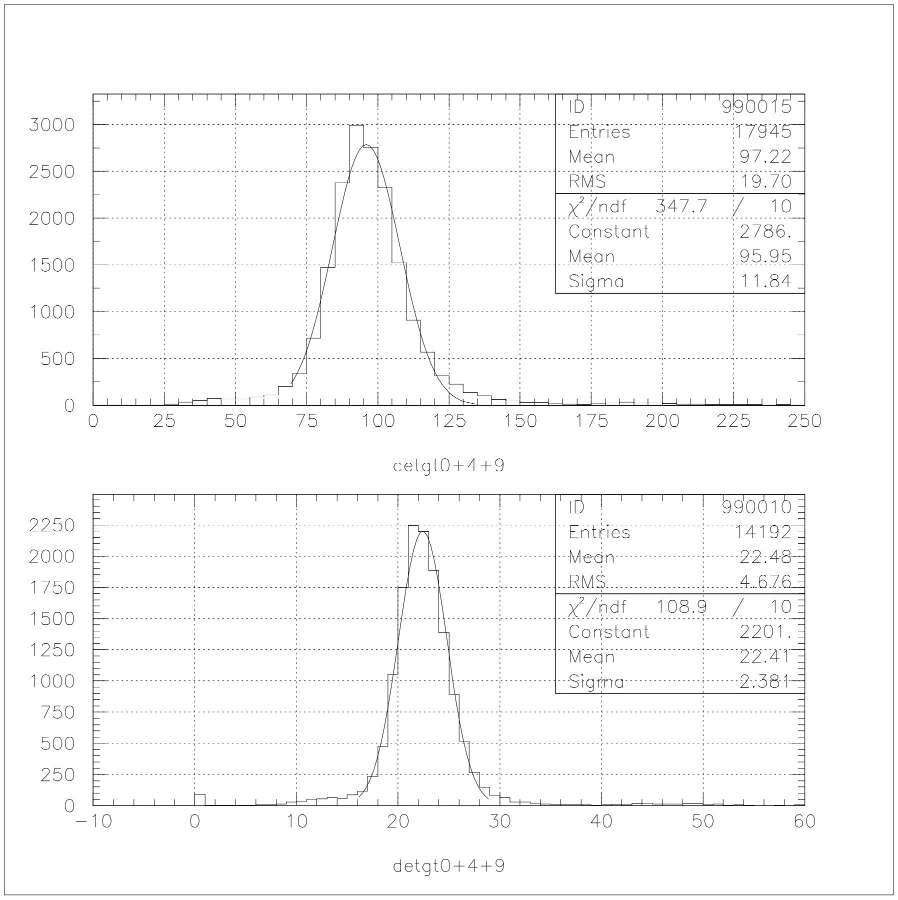


Figure 5.9: DSC (DADC) vs BADC of summed target energy.

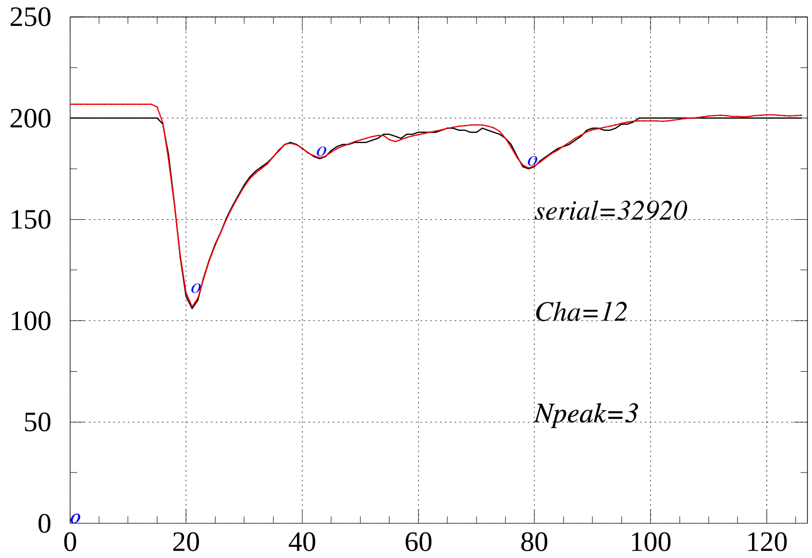


Figure 5.10: Three-peak $\mu^+ \to e^+\nu\overline{\nu}$ candidate event.

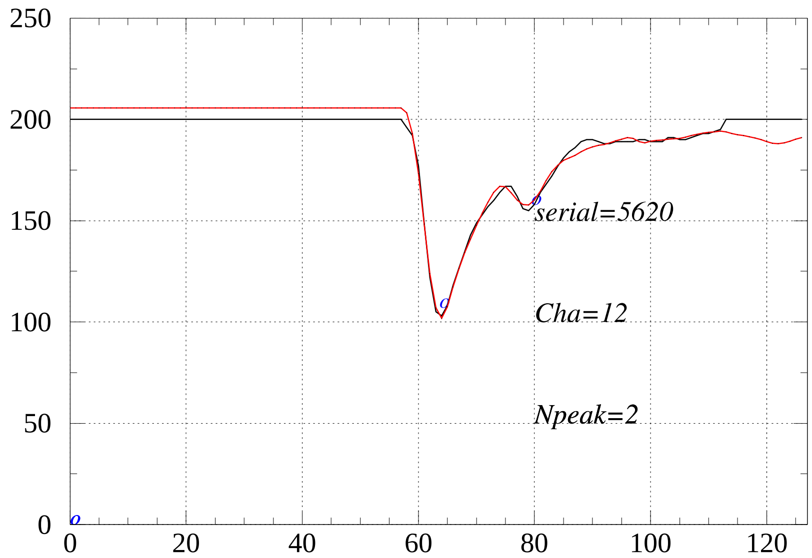


Figure 5.11: Two-peak $\pi^+ \to e^+\nu$ candidate event.

5.6 Test of DSC energy(dadc) reading's particle identity independence

The DSC energy(dadc) is obtained in our experiment from the amplitude of the signal peak (times a software gain). But the physical energy is proportional to the area(total charge) between the signal curve and the baseline. So the hidden assumption is that the geometrical feature of the signal curve should be the same for all kind of particles, γ,e^+,π^+,etc, or the amplitude of the signal is proportional to the area(between signal and baseline). DSC data analysis confirm this assumption. Fig. 5.12 compare the ratio of π^+ and e^+ hits in degrader. Note the same distribution peak position for them despite the different distribution sigma which reflect the different particle identity. Fig. 5.13 show the comparison between γ and e^+ for hits in CSI calorimetor. Fig. 5.14 gives the two dimensional plot of ratio over dadc which shows that the ratio of area over dadc is independent of the signal amplitude (dadc). This kind of plot can be used as an additional check for a missed signal pulse.

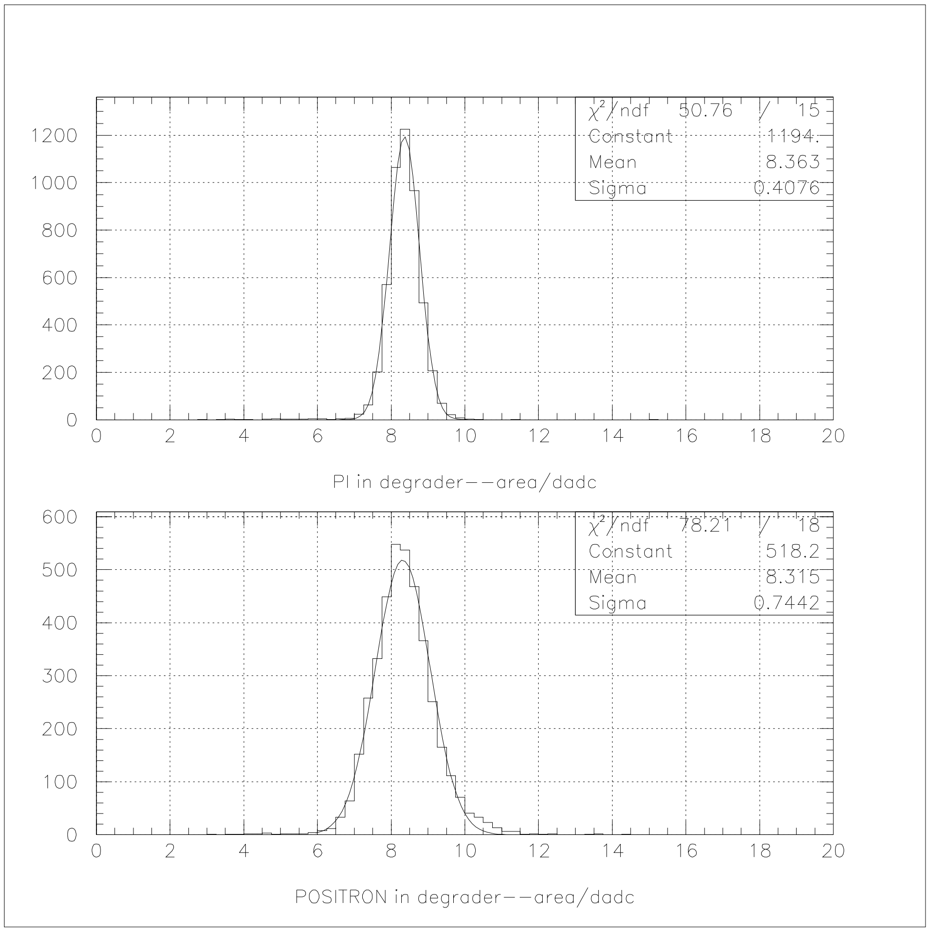


Figure 5.12: degrader's ratio of area over dadc for hits of π^+ and e^+.

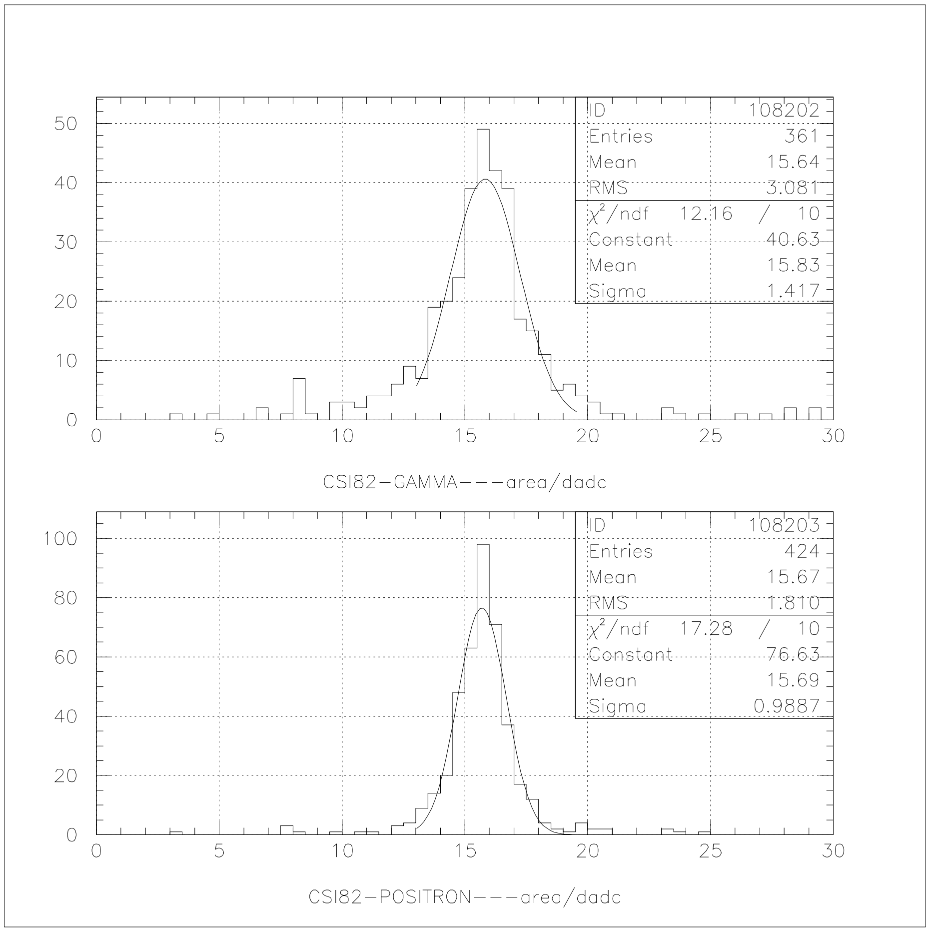


Figure 5.13: CSI's ratio of area over dadc for hits of γ and e^+.

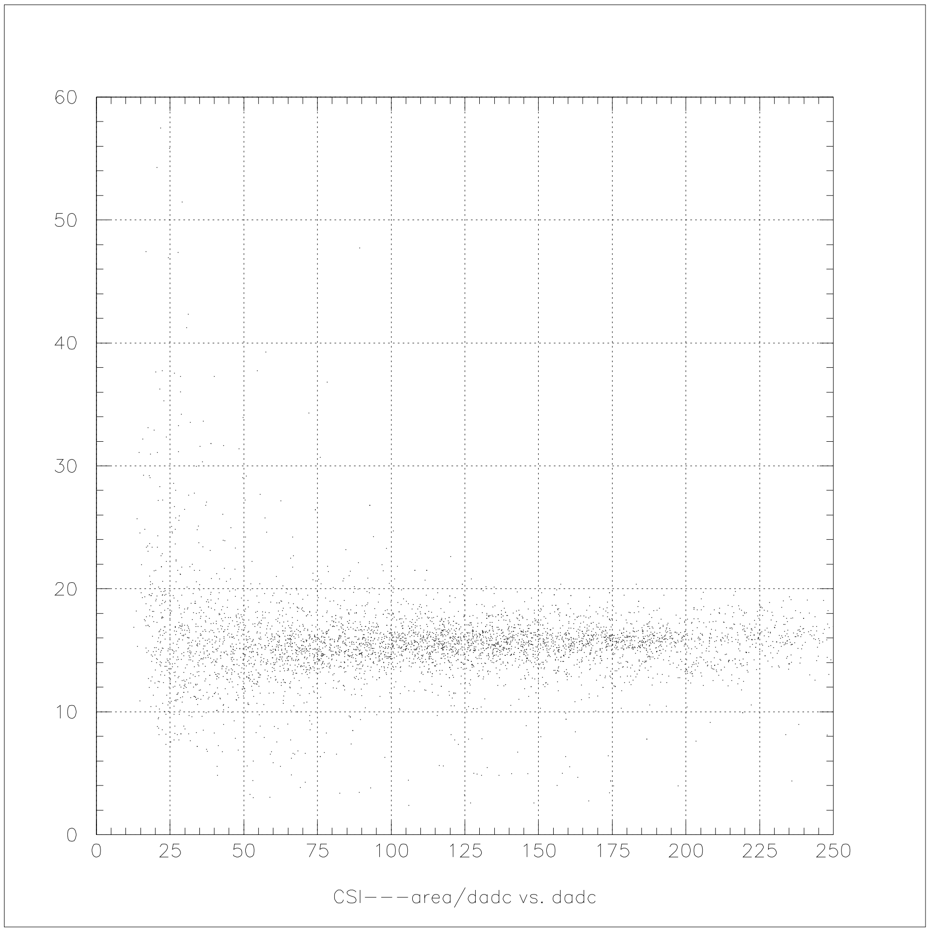


Figure 5.14: CSI's ratio of area over DADC vs DADC.

Chapter 6

Digitizer waveform fitting—keys to optimizing DSC data analysis

6.1 DSC waveform fitting guideline

Waveform fitting plays essential role in DSC data analysis and is usually done offline, as it takes most of CPU time and very slow. Best DSC timing and energy resolution rely on perfect waveform fitting which also lays a solid foundation for all related physical analysis. Guiding principles for work on waveform fitting are given here.

- Write a good least χ^2 fit program which should be fast and efficient giving few failed fits. This kind of program could be constructed from available standard software packages. In our PIBETA DSC analysis it is included in the file 'util.c' of the latest version of analyzer.

  Make sure that system function is good. Usually, a system function is updated constantly with data analysis when Raw DSC data are present. If for a specific channel system function can not be updated because of low statistic of Raw

DSC, a system function could also be optimized by averaging CDSC waveforms containing only one hit around several data points at the main peak position, although background (noise) line shape is not updated. But in the most cases, background is zero-suppressed and will not affect overall fitting quality much.

- Make sure fitting around significant peaks is adequate($chisqr = 1$) since χ^2 is very sensitive to fitting parameters which provide timing (peak position) and energy (amplitude) of the event (peak). This is partially taken care of by assigning highest fitting weight for data points around peaks and lowest weight for data points at baseline. Generally speaking, good fitting around significant peaks can be guaranteed when a system function is appropriate. But sub-standard fitting can occur when the number of peaks is large and when adjacent peaks are very close to one another. This configuration needs careful peak selection algorithm which is discussed in the next section.

Fig. 6.1 shows a bad fit that miss a hidden peak. For the same original waveform, Fig. 6.2 shows an improved fit that recover the missed peak. Fig. 6.3 show an appropriate fit for an event with many peaks.

6.2 Waveform fitting algorithm

At present, two algorithms are applied to PIBETA DSC data analysis. The C program code is attached at Appendix A. The advantages and drawbacks of the two algorithms are discussed as follows.

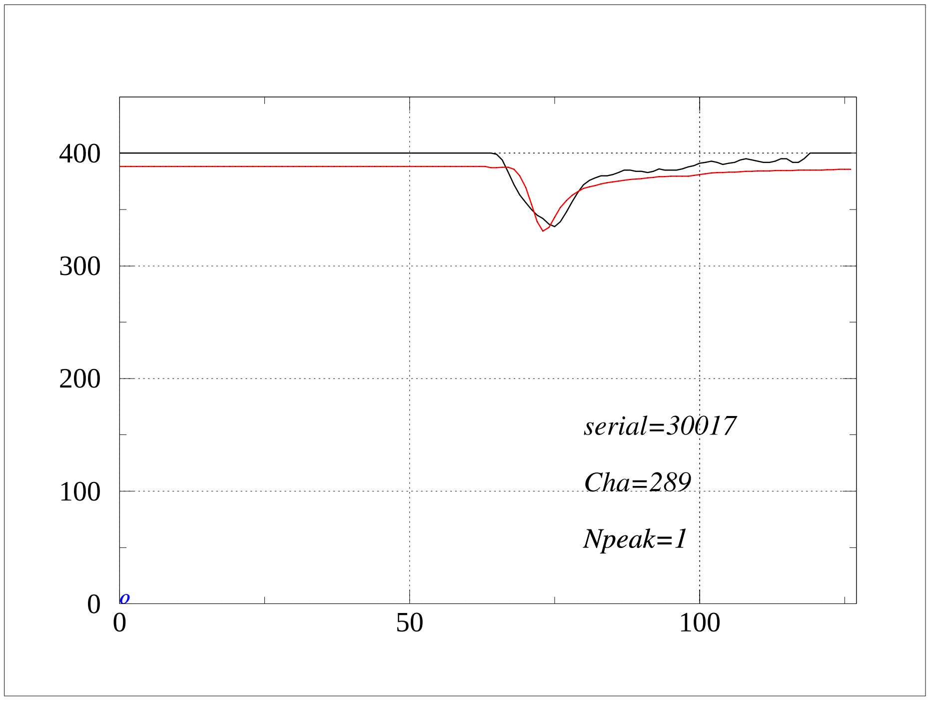


Figure 6.1: Bad fitting missing hidden peak.

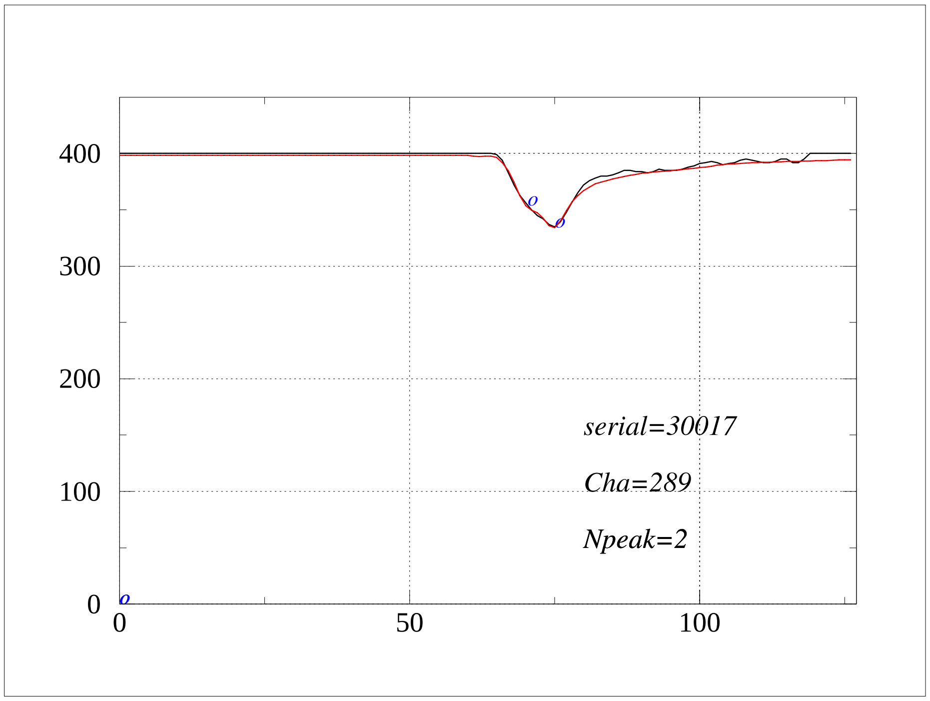


Figure 6.2: Improved fitting recovering hidden peak.

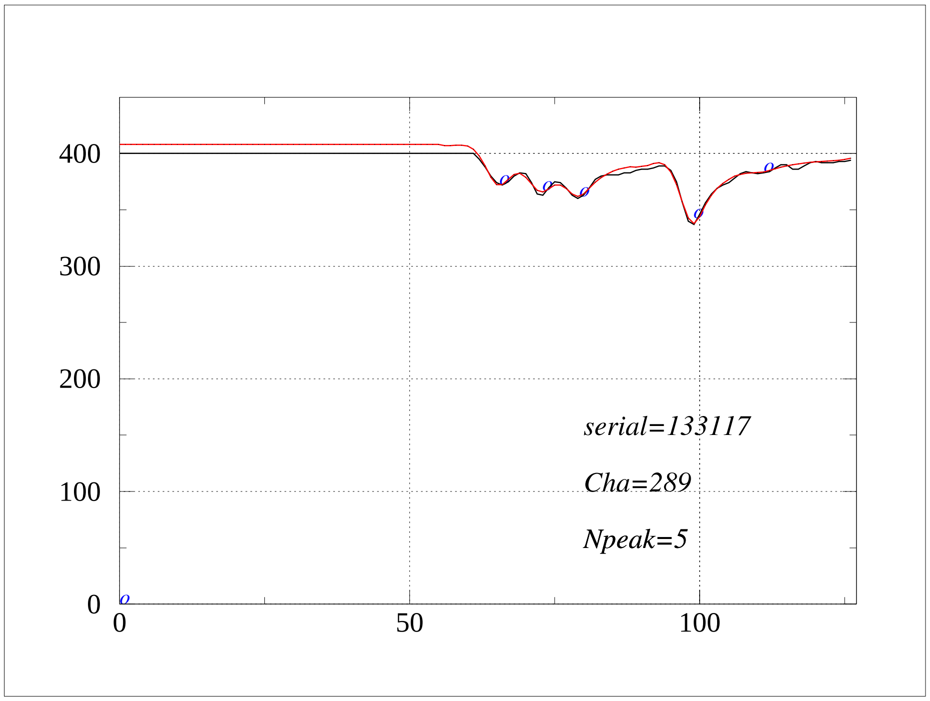


Figure 6.3: Appropriate fit for many-peaks event.

- Algorithm One. This is the one discussed in the DSC introduction. Significant peaks are picked out from the biggest one to the smallest one. This is the only algorithm used in early phase of DSC analysis. Since new peak is obtained from the adjusted waveform with the superposition of all previous peaks, it is pretty efficient at finding hidden peaks in the waveform. It works satisfactorily for channels with small (less than 3) number of peaks. But for channels with many hits (peak number greater than 3), such as B0 detector and Degrader detector, this algorithm's performance is not satisfactory. Failure symptoms include missing significant peaks or picking out non-existing peak at the baseline. And the efficiency (proportion of successful fits) is far from 100% for channels like

B0 and Degrader.

Algorithm Two

This algorithm is developed to fix the improper functioning of the first algorithm for channel B0 and degrader. The significant peaks are picked out by scanning the sampled data points on the waveform from left to right one timing block by one timing block, with the size of the block carefully adjusted to avoid missing a peak which lies at range edge, while also applying amplitude threshold cut as in previous algorithm to avoid picking up background noise. The apparent advantage of this algorithm is that it will not miss main signal peaks and most importantly, it will not pick up non-existent peaks at the baseline as the first algorithm. The drawback of this algorithm is that it will not find the hidden peaks (the ones that are small and very close to the main peak). This drawback could be fixed by adding a final diagnostic process which checks the value of fit χ^2 around every significant peak. The peak with abnormally high χ^2 may hide a small peak. Use a peak-finding method of first algorithm to try to pick up a hidden peak. If the χ^2 of the fit with an added hidden peak is significantly improved, we think we recovered a hidden peak. Finally, this algorithm is better overall compared to the first algorithm and is applied to almost all channels for later stage DSC analysis. Refer to Appendex B for the C code written using the algorithm discussed above.

Fig. 6.4 shows a fitting using algorithm one which miss one real peak. Fig. 6.5

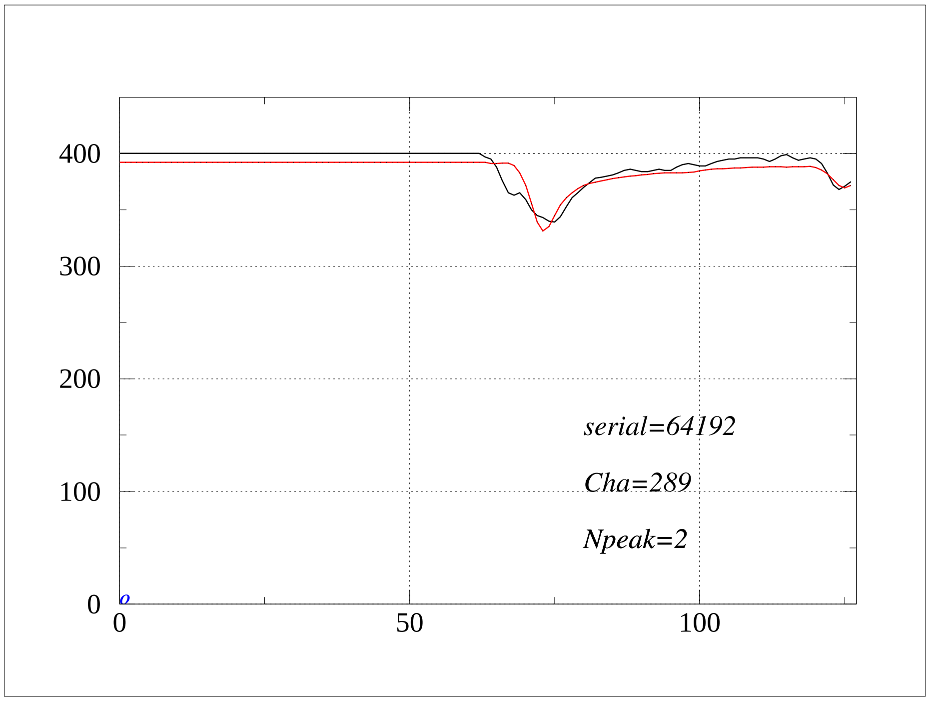
- 

Figure 6.4: Fitting using algorithm one(missing one peak).

shows the fitting on the same original waveform using algorithm two that recover the first real peak.

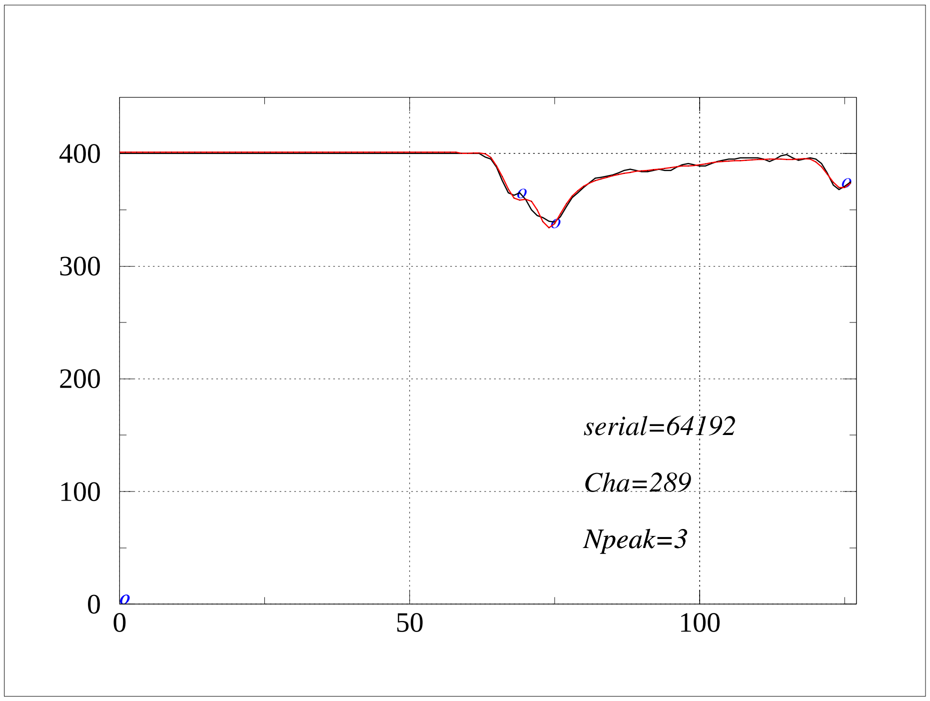


Figure 6.5: Fitting using algorithm two(found the first peak).

Chapter 7

Extracting $\pi^+ \to \pi^0 e^+ \nu$ and $\pi^+ \to e^+ \nu$ events using digitizer

7.1 DSC analysis toward clean $\pi^+ \to \pi^0 e^+ \nu$ events

PIBETA candidate events are pre-selected by generating skimmed '.mid' files that retained only PIBETA high trigger events of the original data files. The data size to be analyzed by Digitizer is reduced by about factor about 50. DSC waveform fitting is done only on events that meet the following cuts:

- two-arm high energy threshold trigger(PB_{HI}),

  no cosmic events, which requires that total energy in CsI calorimeter is less than 200 MeV and timing registered in cosmic veto detector is outside a 140 ns wide pion stop window (no in-time hits in the cosmic veto detector),

- no in-time hits in two active collimators,

- no charged particles detected in the direction of candidate clumps.

The cuts, list above are broader than conventional standard cuts specified in Weidong Li's thesis [1] just to avoid real PIBETA events leakage (inefficiency). Additional cuts will be applied to get clean PIBETA events after DSC waveform fitting is done and use DSC variables.

When good, high efficient waveform fitting is done, two kind of cuts with DSC information are applied to get clean PIBETA events. When applicable corresponding TDC analysis is supplied following DSC analysis just to show the special features of the Digitizer.

7.1.1 Two-dimensional timing cut to eliminate Prompt events contamination

Two-dimensional timing cut works on two dimensional plot of degrader (or B0) energy vs timing difference of CsI calorimeter and degrader. It is workable since prompt events will register with a unique energy (higher)-timing correlation in B0 and degrader that makes them separated from PIBETA events in the two dimensional plot.

Fig. 7.1 shows how the cut using degrader CADC energy is defined. Fig. 7.2 shows the spectrum of timing difference between calorimeter $((t_{\gamma 1}+t_{\gamma 2})/2.0-t_{degrader}$ under this cut.

Fig. 7.3 shows how the cut using B0 CAMAC energy is defined. Fig. 7.4 shows the spectrum of timing difference between CSI $((t_{\gamma 1}+t_{\gamma 2})/2.0-t_{degrader}$ under this cut.

Fig. 7.5 shows how the cut using degrader DSC energy is defined. Fig. 7.6 shows the spectrum of timing difference between Calorimeter $((t_{\gamma 1}+t_{\gamma 2})/2.0-t_{degrader}$ under

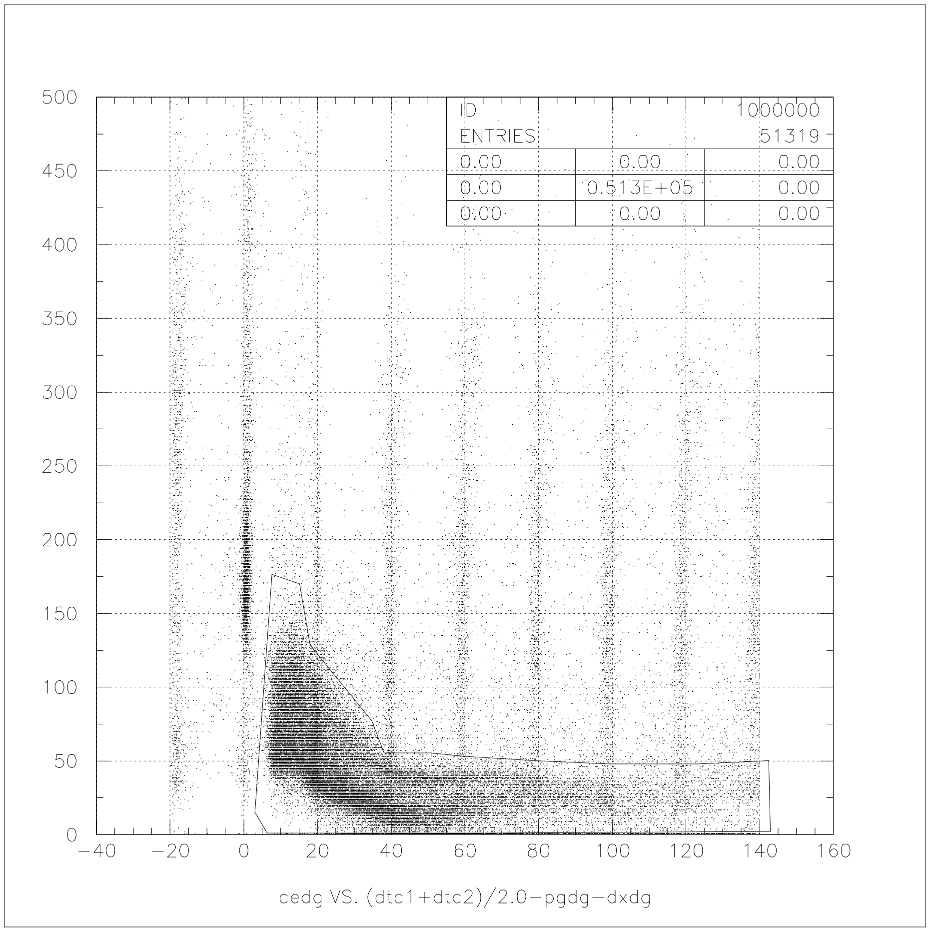


Figure 7.1: Sausage cut using degrader energy.

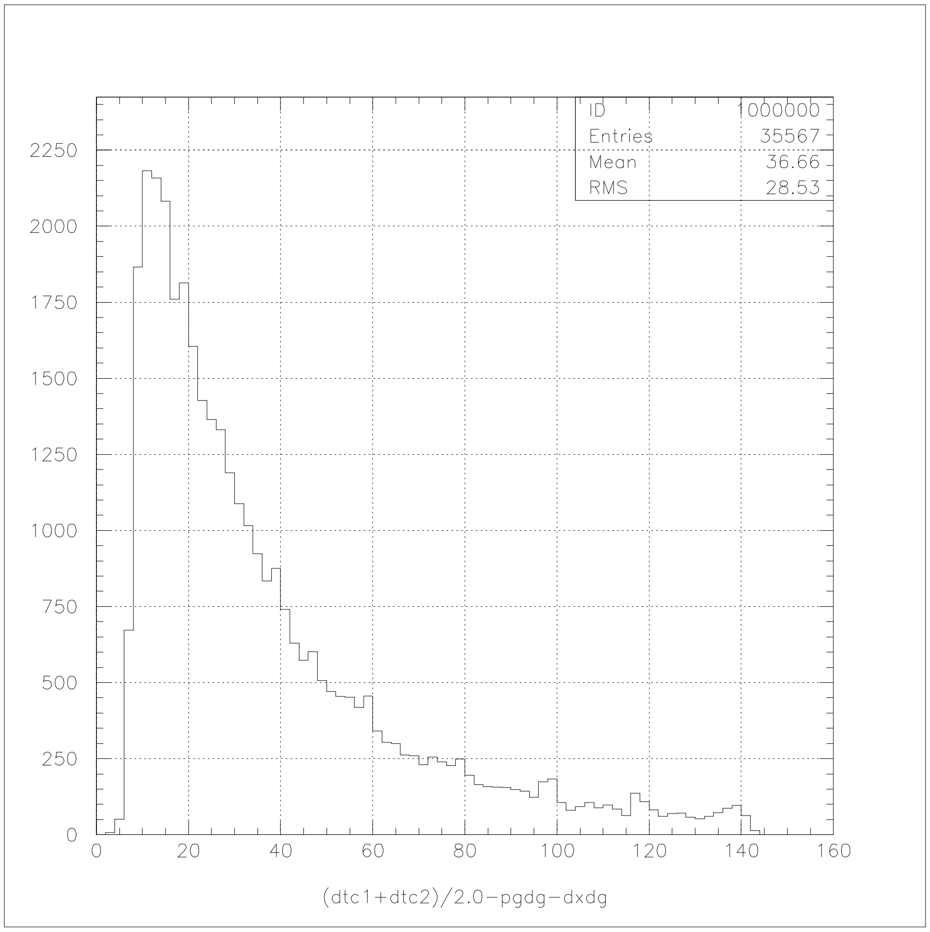


Figure 7.2: Timing spectrum of CSI-Degrader under degrader sausage cut.

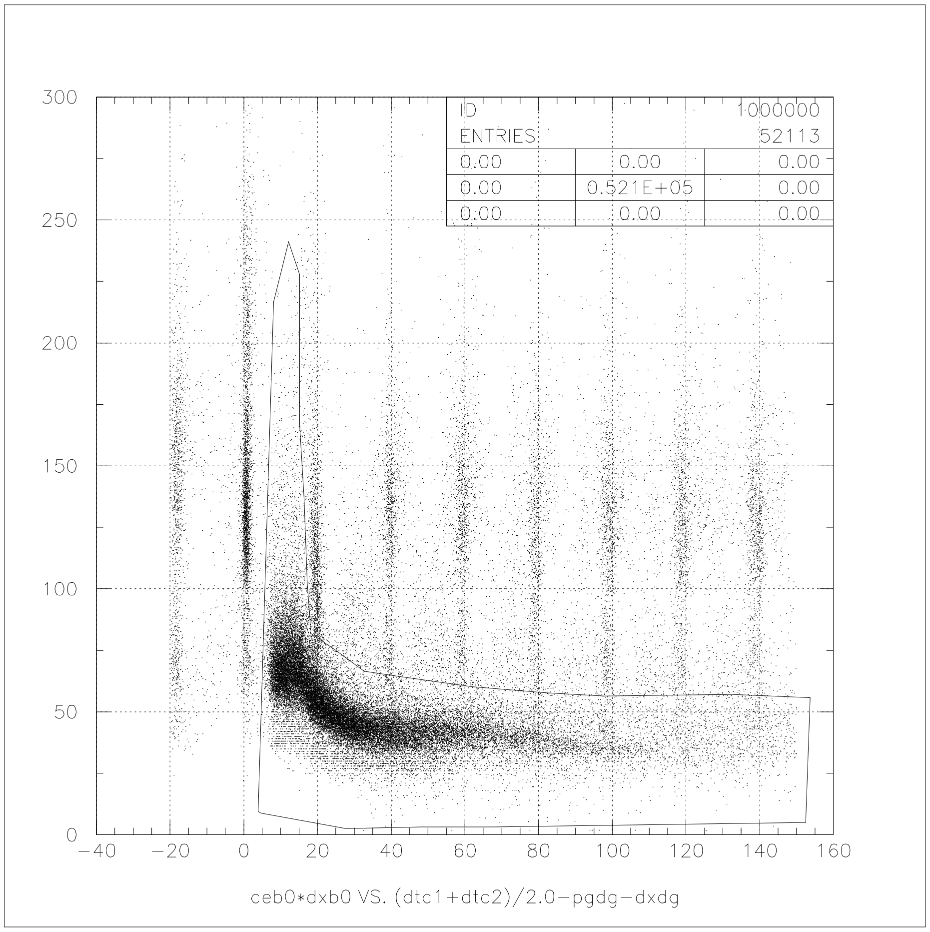


Figure 7.3: Prompt events cut using degrader energy.

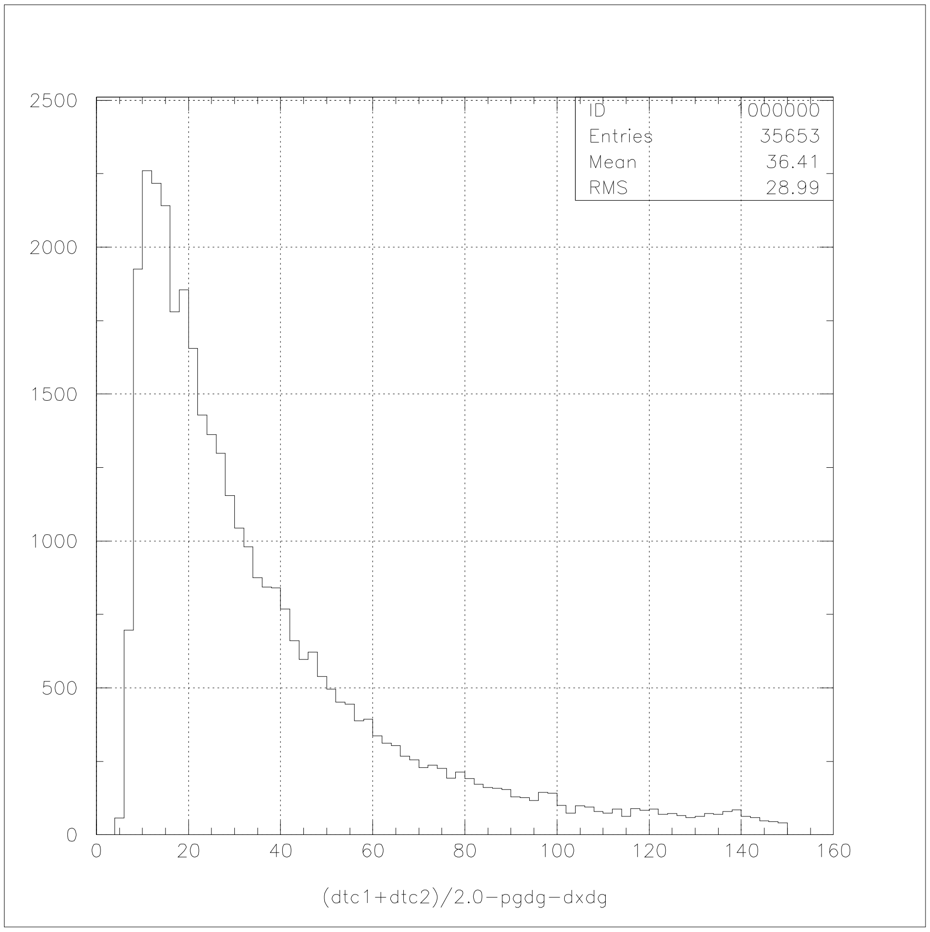


Figure 7.4: Timing spectrum of CSI-Degrader under B0 prompt cut.

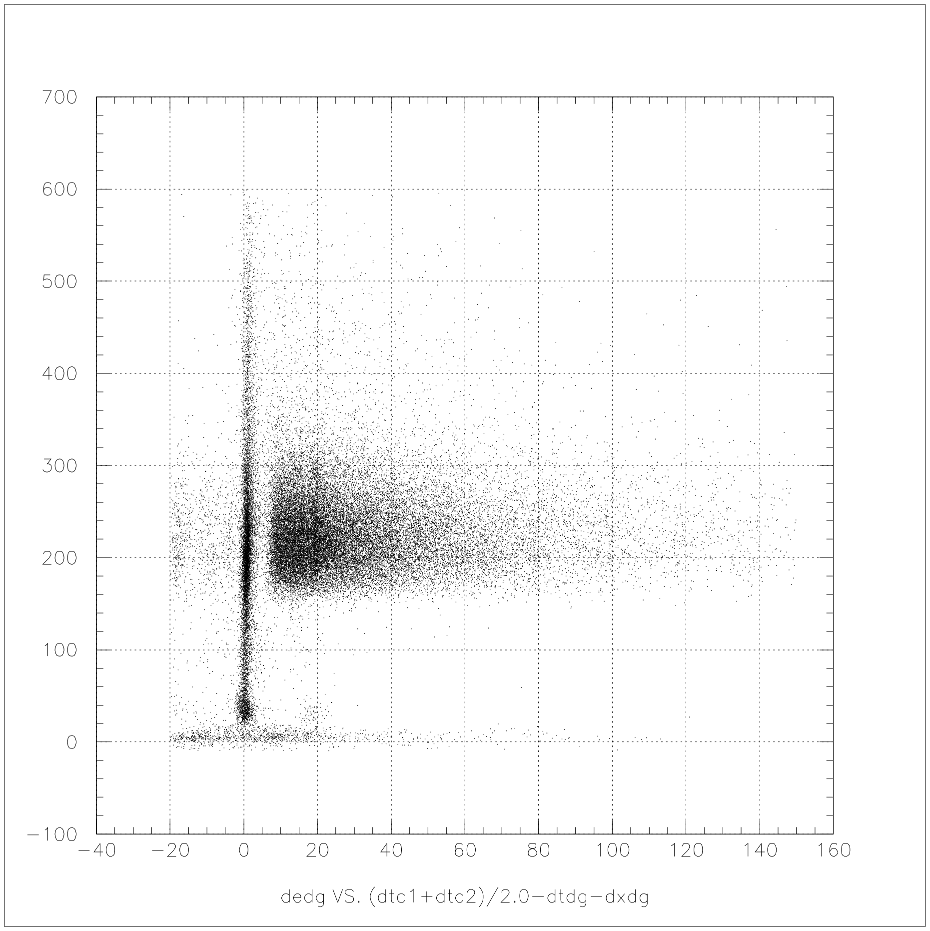


Figure 7.5: Prompts events cut using DSC degrader energy.

this prompt cut. Please note that we use the DSC timing of the degrader that is closest to trigger timing instead of that one that closest to pi-gate. We can see that prompt events are centered around time zero.

For comparison purpose prompt cut on TDC timing is shown in Fig. 7.7 using CAMAC degrader energy vs TDC timing of calorimeter. Fig. 7.8 show the TDC timing spectrum under this prompt cut.

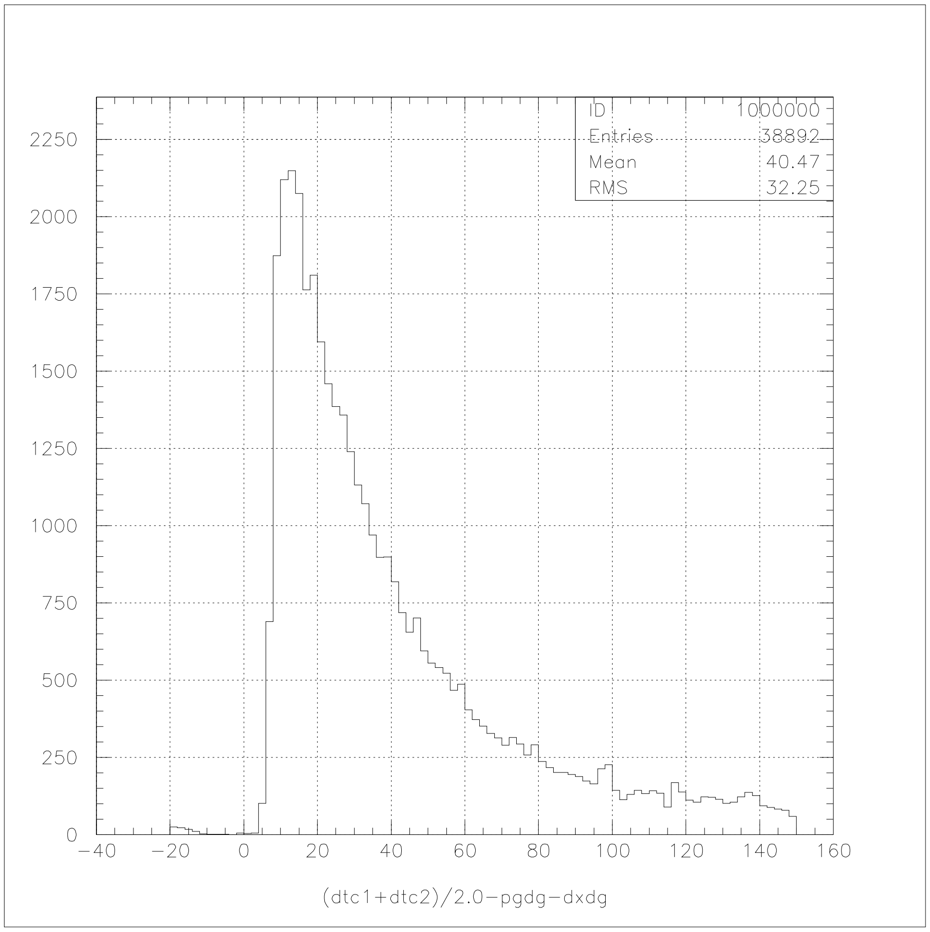


Figure 7.6: Timing spectrum of CsI Calorimeter-Degrader under DSC timing cut.

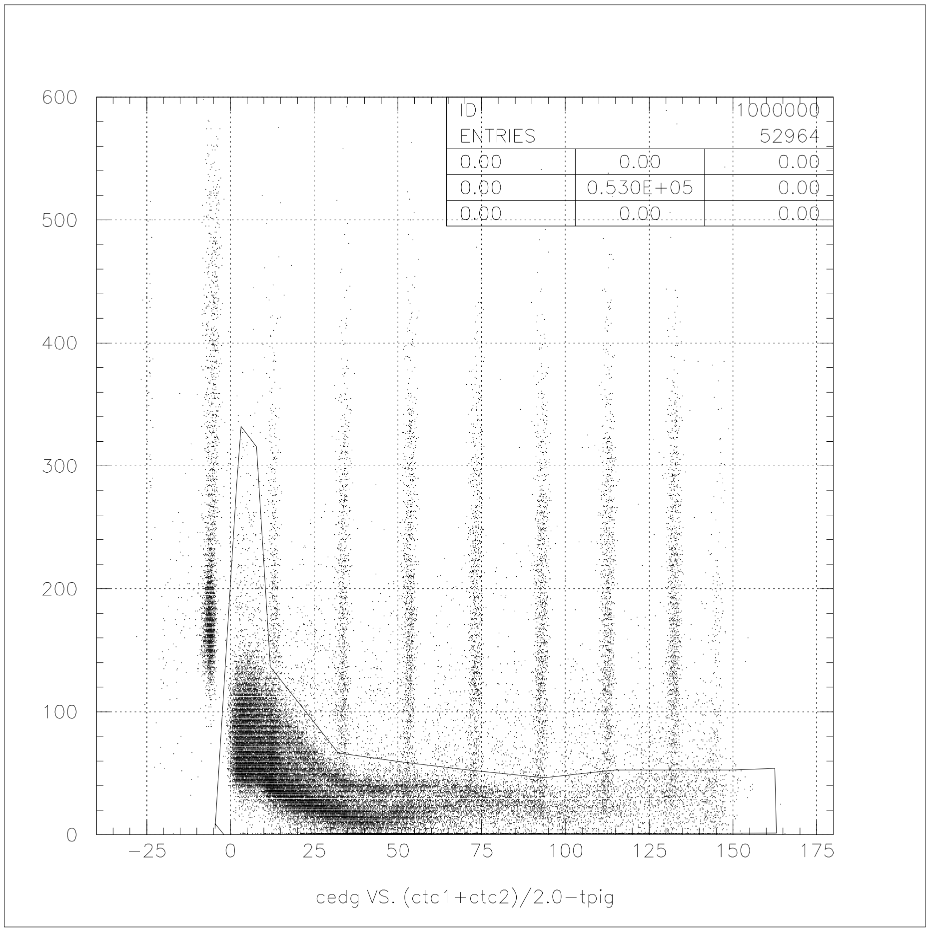


Figure 7.7: Prompt cut using DSC degrader energy for TDC timing.

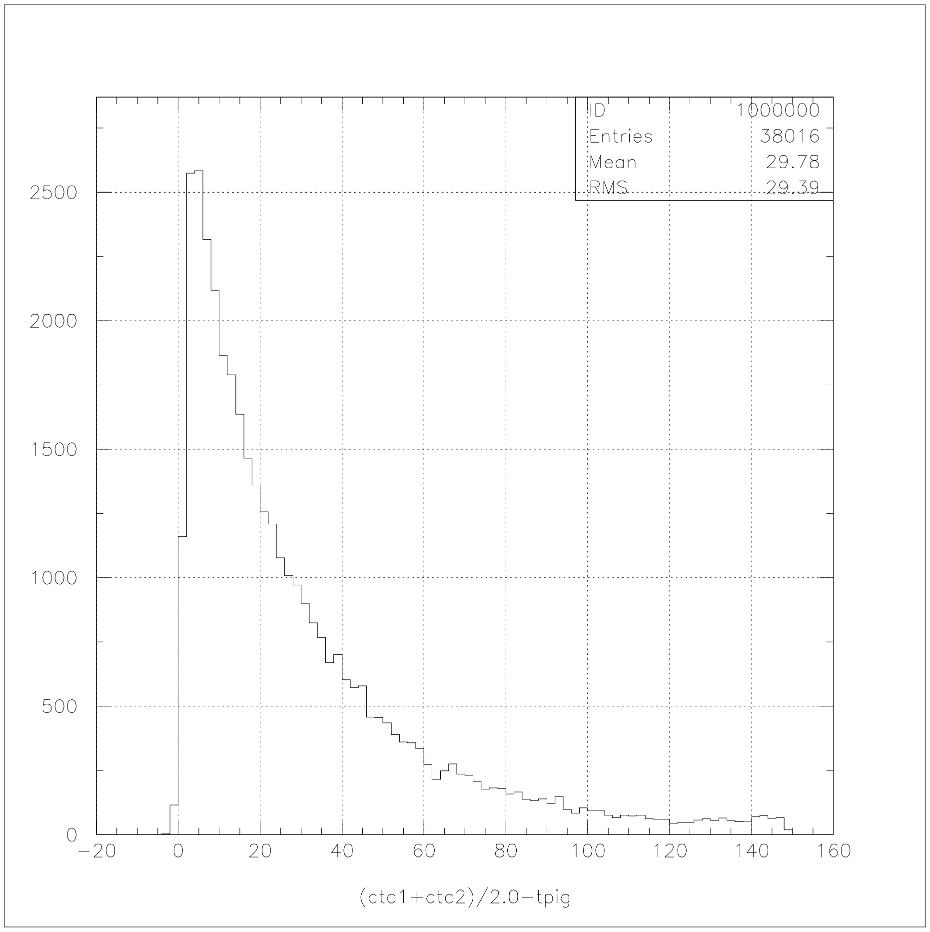


Figure 7.8: TDC timing spectrum of CSI-Degrader under DSC prompt cut.

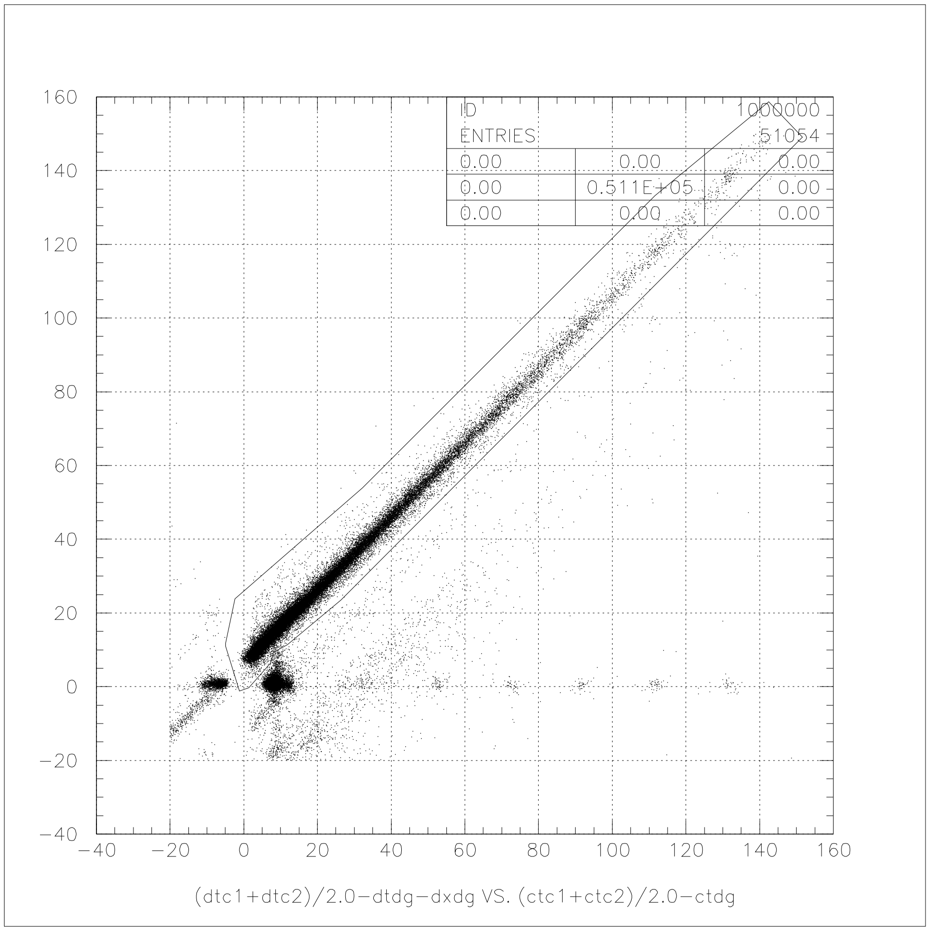


Figure 7.9: Prompt cut using DSC degrader energy for fast bus timing.

7.1.2 Two-dimensional plot of DSC timing vs Fastbus timing

For the same physical event (PIBETA), with both DSC timing and FASTBUS TDC timing are available could construct two dimensional of these two variables and pick up the real PIBETA events that corresponds the data points that lie at the diagonal of the plot.

Fig. 7.9 shows how such 2-dimensional cut is defined. Fig. 7.10 gives the projected timing spectrum (with the same variables as before) with the 2-dimensional cut.

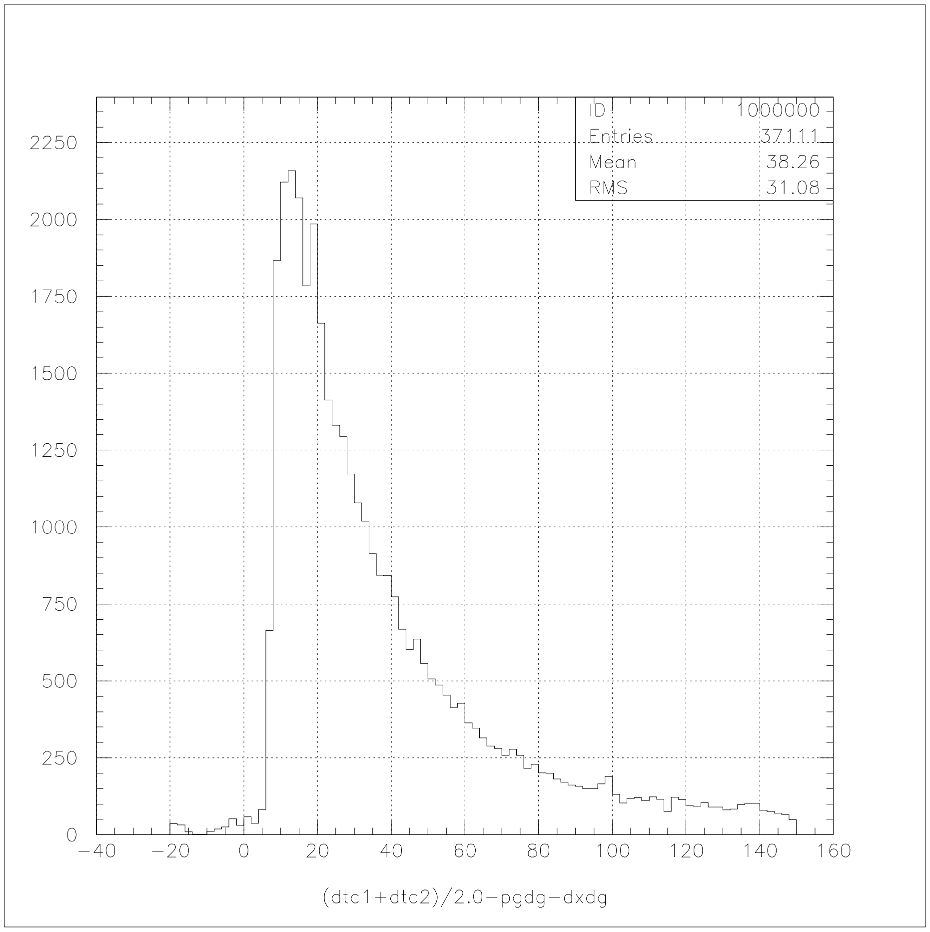


Figure 7.10: FASTBUS TDC timing spectrum of CSI-Degrader under DSC prompt cut.

7.1.3 Positron spectrum of energy deposition in the target for $\pi^+ \to \pi^0 e^+ \nu$ decay

A unique feature that DSC possesses is that it can track the e^+ energy deposited in the target from $\pi^+ \to \pi^0 e^+ \nu$ decay by finding the amplitude of the small signal peak that follows the π^+ stop signal peak of a PIBETA event. FASTBUS (CADC) can not provide us with this information since the positron signal is totally submerged into the whole π-stop signal for early decays, and cannot be separated by the single value output of CADC.

Based on $\pi^+ \to \pi^0 e^+ \nu$ event generator, Geant simulation gives the e^+ energy spectrum as shown in Fig. 7.11.

We can see that maximum energy of e^+ from $\pi^+ \to \pi^0 e^+ \nu$ is about 4.3 MeV which means that e^+ can not escape the target. So energy reading of from digitizer in target gives total energy of the e^+.

Fig. 7.12 gives a typical DSC target channel's waveform of a PIBETA event with second peak indicating the e^+ from $\pi^+ \to \pi^0 e^+ \nu$ decay.

Fig. 7.13 gives the DSC summed target energy ($dadc[291] + dadc[292] + dadc[293]$ in our experiment) spectrum of e^+ from $\pi^+ \to \pi^0 e^+ \nu$ decay. Since target DSC energy is well calibrated , we can see that energy of e^+ indeed ends at about 4.3 MeV with background noise extending into higher energies.

Cleaner DSC positron target energy spectrum can be obtained from two- dimensional plot of summed target energy vs timing difference of CsI calorimeter and de-

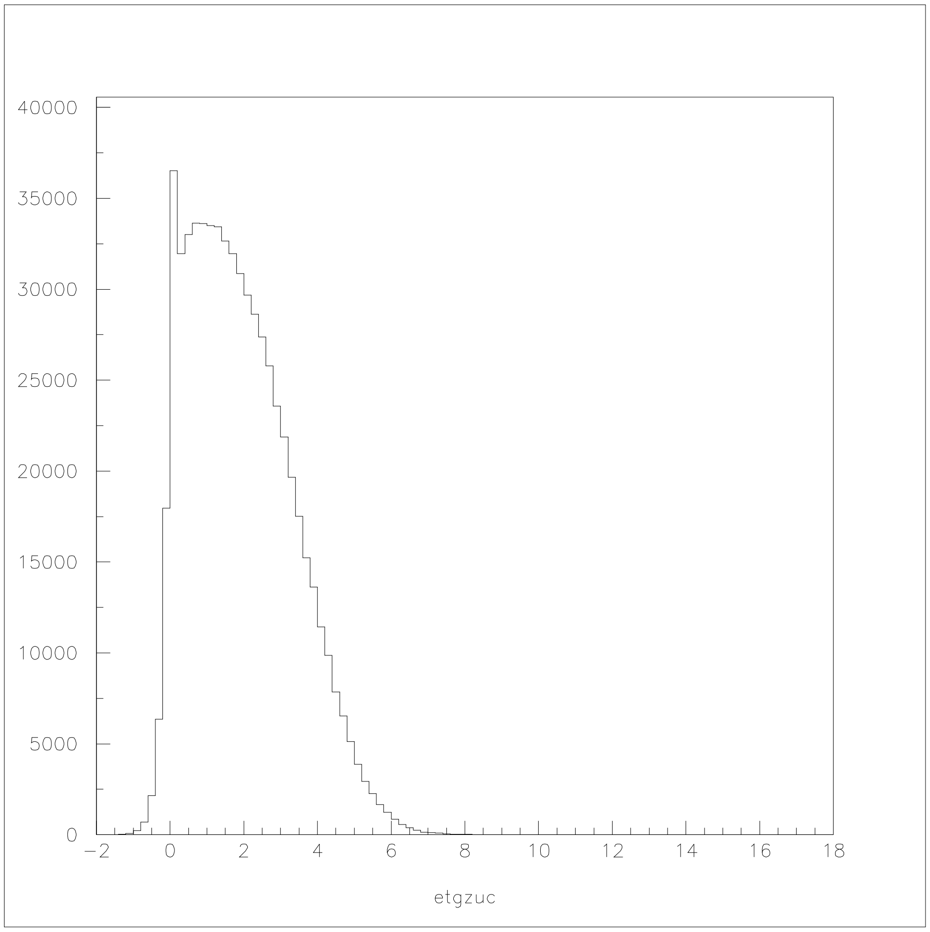


Figure 7.11: e^+ energy from $\pi^+ \to \pi^0 e^+ \nu$ decay Caculated in Geant simulation.

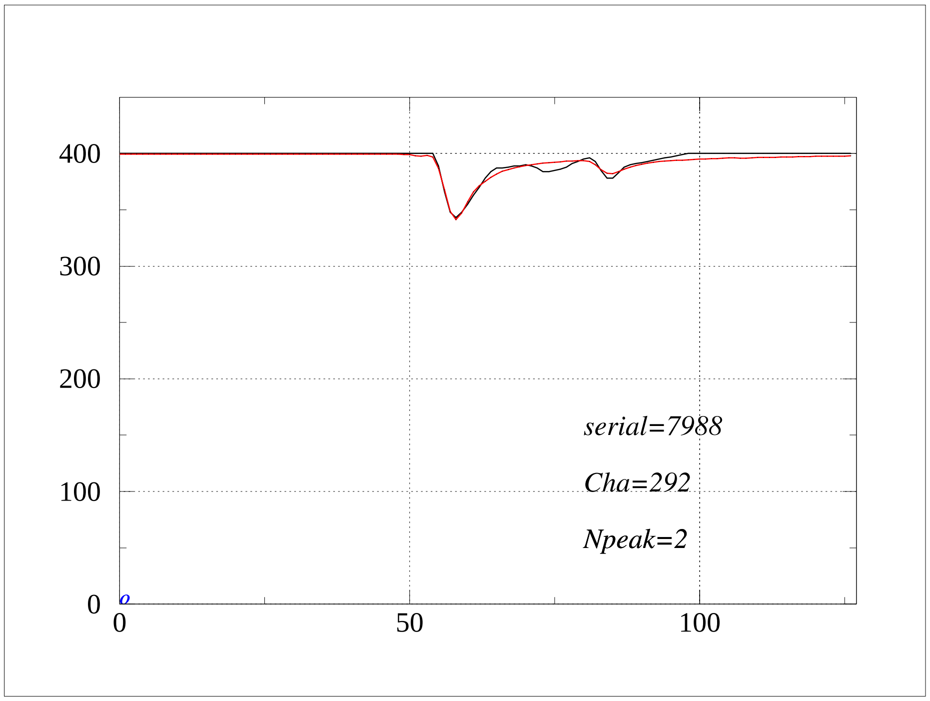


Figure 7.12: $\pi^+ \to \pi^0 e^+ \nu$ decay waveform in the target.

grader. Fig. 7.14 show the cut that picks up the straight band in energy range 0–5 MeV which clearly stands out for positron events. Fig. 7.15 give positron energy spectrum in target using this cut.

7.2 DSC analysis extracting $\pi^+ \to e^+ \nu$ events

This section could be $\pi^+ \to e^+ \nu$ analog of previous PIBETA analysis with very similar cuts to eliminate prompt events contamination. But in addition to prompt events, there is large $\mu^+ \to e^+ \nu \overline{\nu}$ background contamination. So there is additional near-flat $\mu^+ \to e^+ \nu \overline{\nu}$ background subtraction process to get clean $\pi^+ \to e^+ \nu$ spectrum.

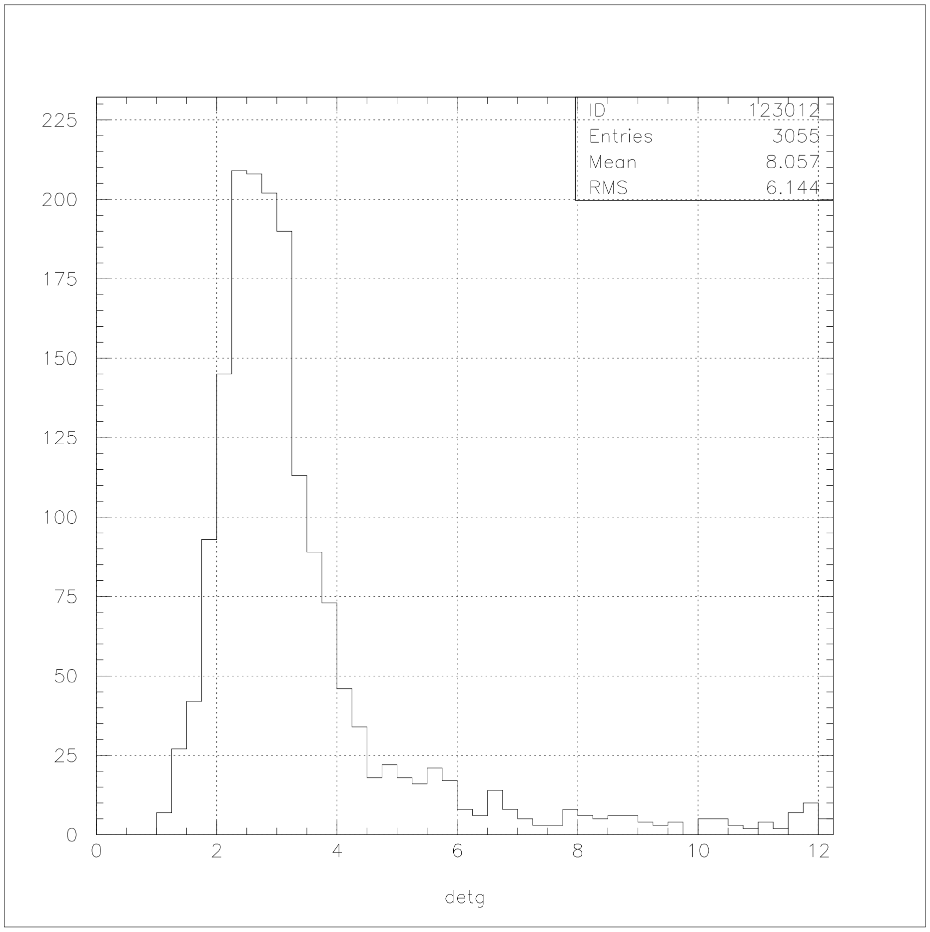


Figure 7.13: Summed DSC target energy of e^+ from $\pi^+ \to \pi^0 e^+ \nu$ decay.

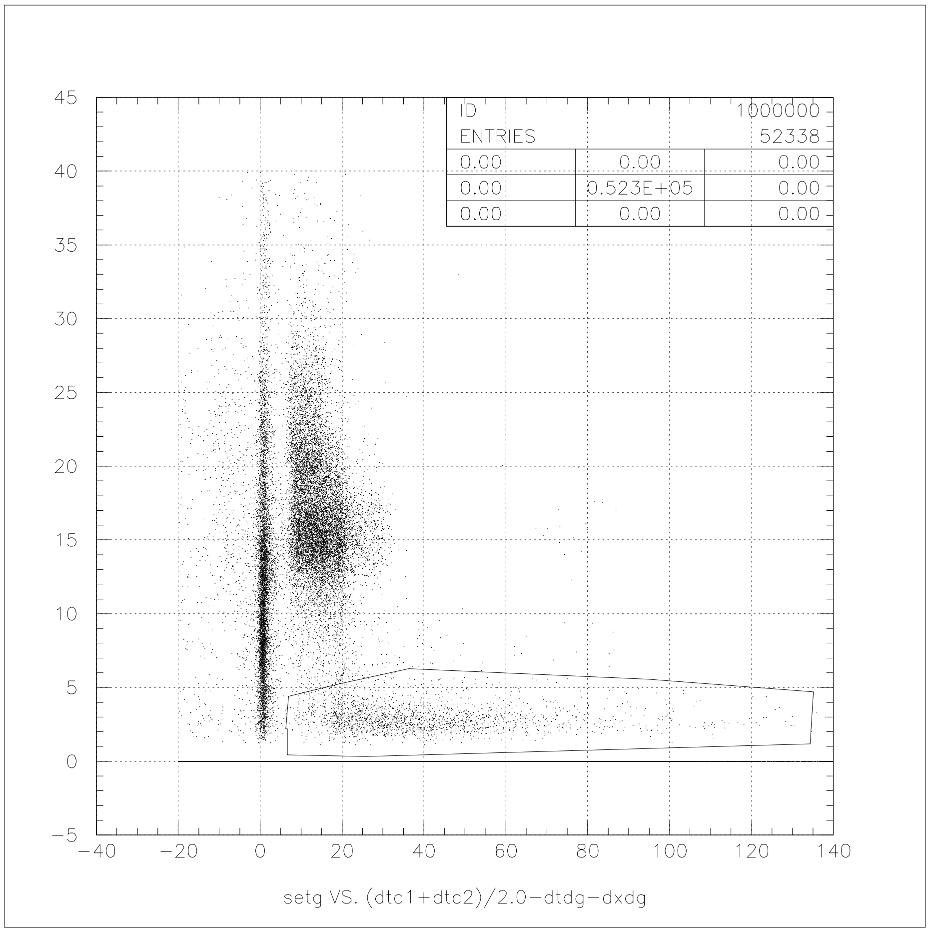


Figure 7.14: DSC summed e^+target energy vs timing difference of Calorimeter and degrader.

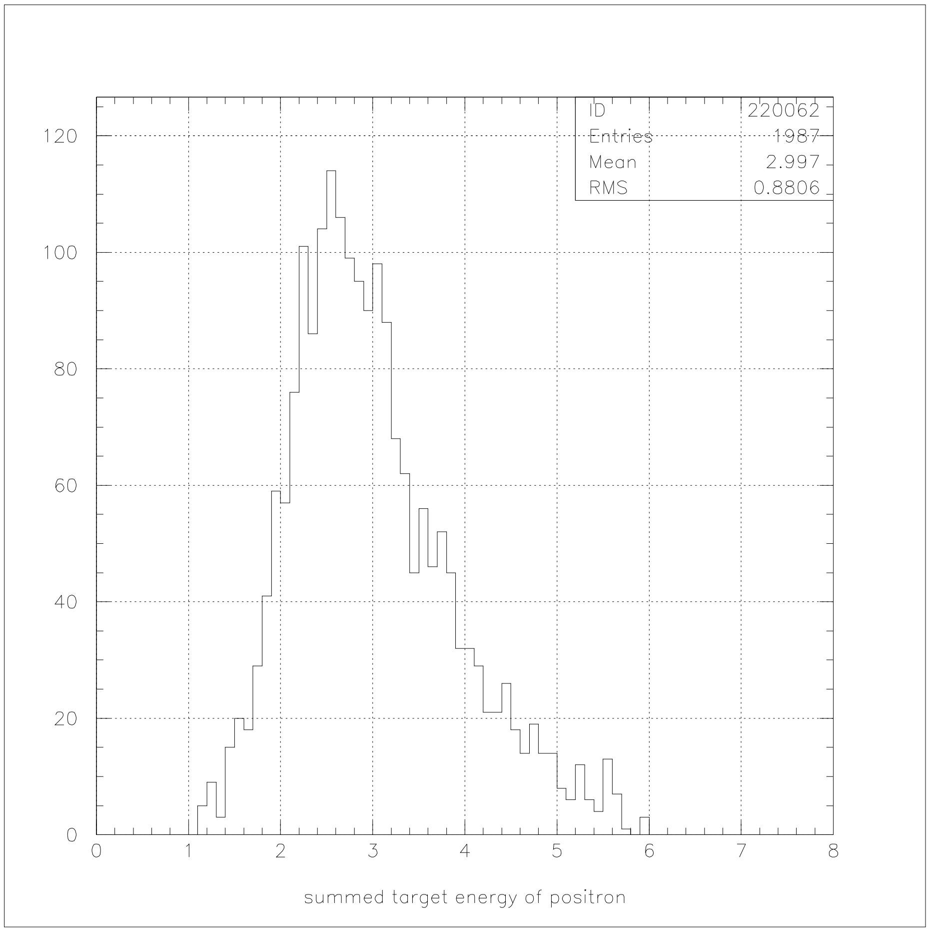


Figure 7.15: DSC summed e^+target energy under 2-D cut of DSC energy vs timing.

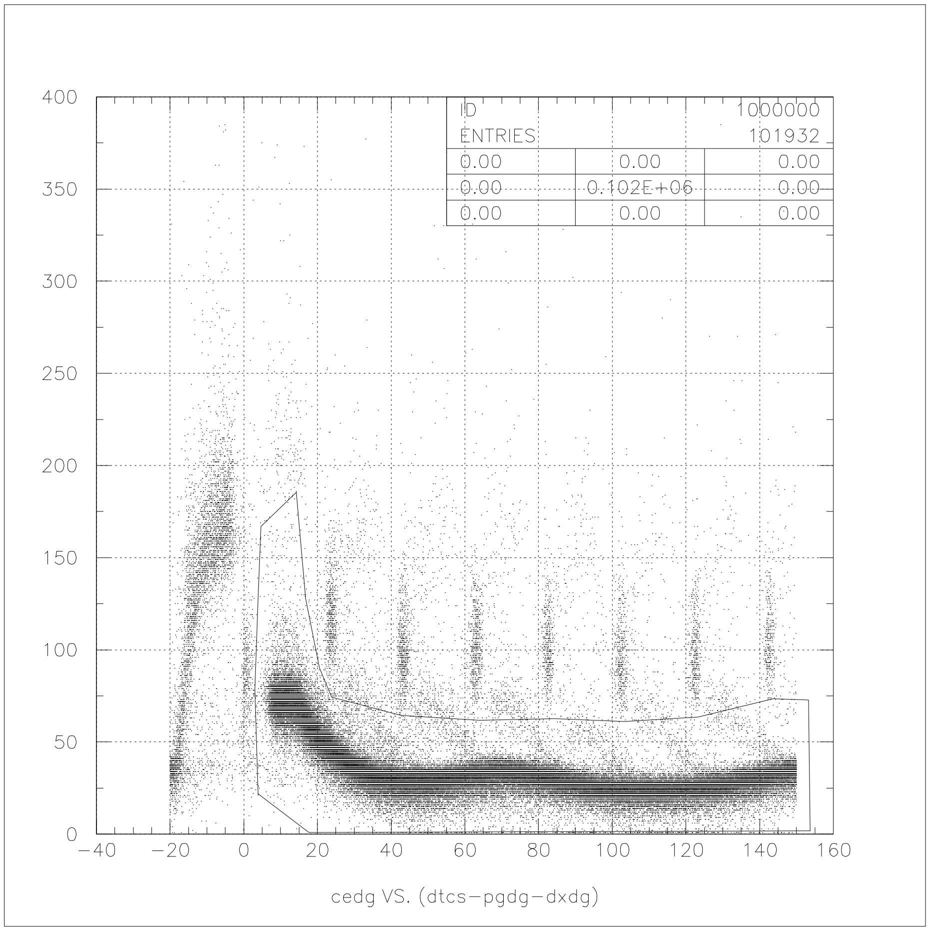


Figure 7.16: Prompt cut using CAMAC degrader ADC.

7.2.1 Prompt events cuts to eliminate background

Here we analyze the same two dimensional plot as that for PIBETA candidate events. The only difference is that here physical variable $tcsi_{e^+}$ replaces PIBETA's $(t_{\gamma 1} + t_{\gamma 2})/2.0$.

Fig. 7.16 shows how the cut using CAMAC degrader energy in y axis is defined. Fig. 7.17 shows the $\pi^+ \to e^+\nu$ timing spectrum with such cut. Background is subtracted(approximately) using a fit with Gaussian function plus constant.

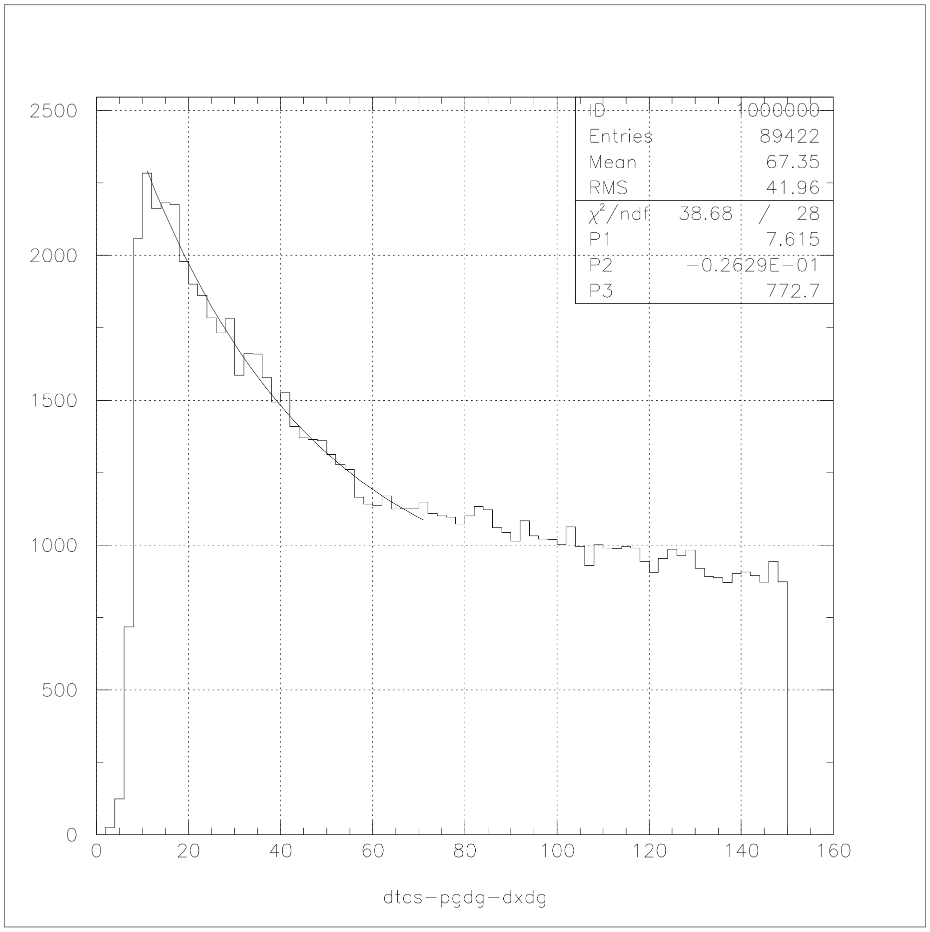


Figure 7.17: $\pi^+ \to e^+\nu$ timing spectrum using CAMAC degrader prompt cut.

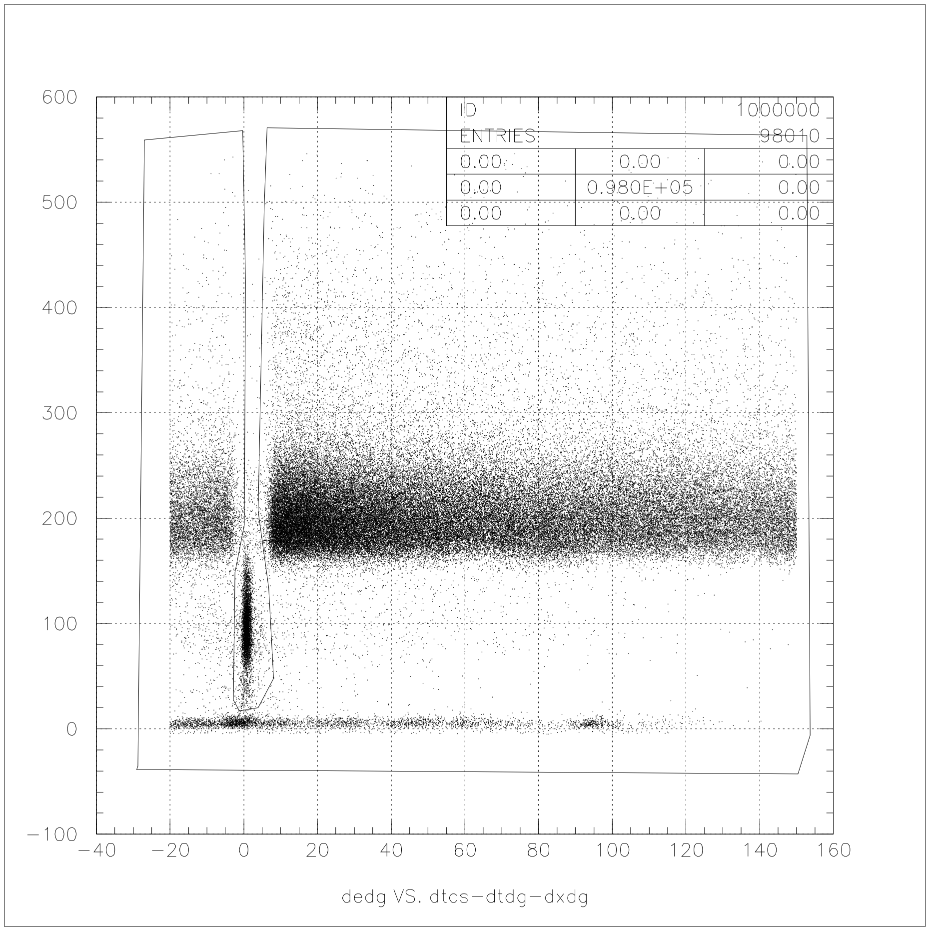


Figure 7.18: Prompt cut using DSC degrader ADC.

Fig. 7.18 shows how prompt cut using DSC degrader energy is defined. Fig. 7.19 is the corresponding $\pi^+ \to e^+\nu$ timing spectrum using such cut.

Fig. 7.20 shows how prompt cut using DSC B0 energy is defined. Fig. 7.21 is the corresponding $\pi^+ \to e^+\nu$ timing spectrum with such cut.

Also, for comparison purpose, corresponding FASTBUS ADC/TDC analysis is given as follows. Fig. 7.22 shows how prompt cut using CAMAC degrader energy and FASTBUS time is defined. Fig. 7.23 is the corresponding $\pi^+ \to e^+\nu$ timing

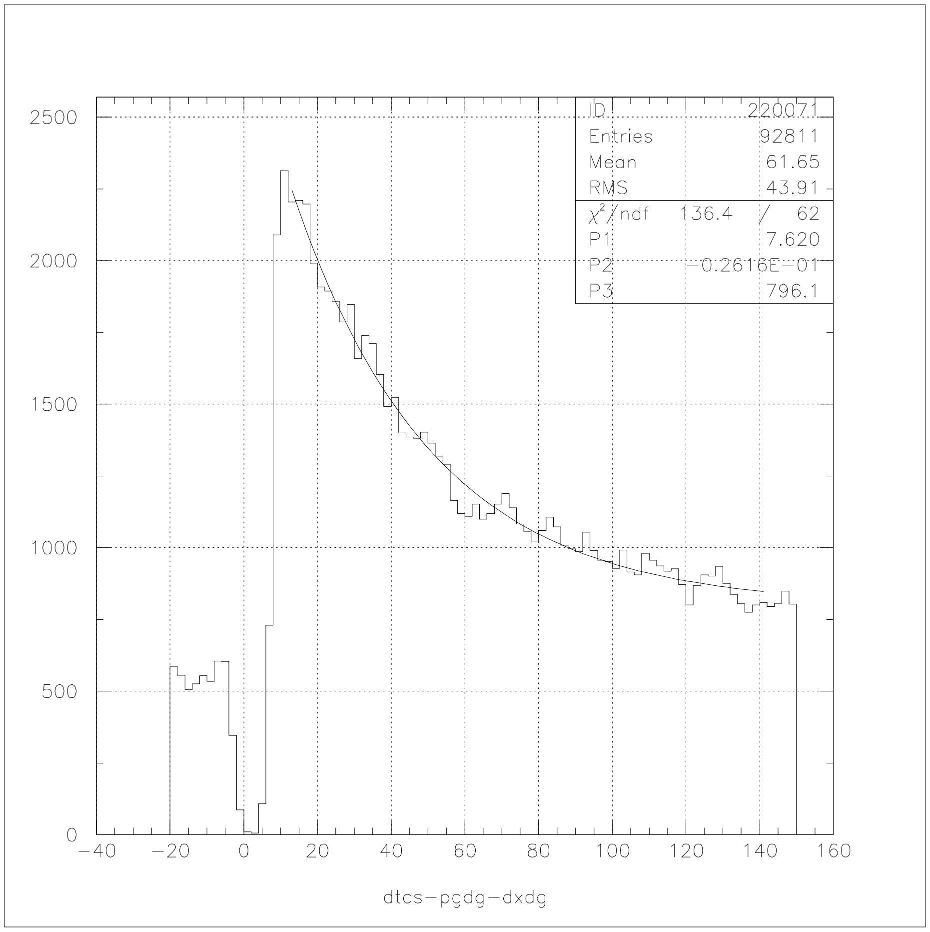


Figure 7.19: $\pi^+ \rightarrow e^+\nu$ timing spectrum using DSC degrader prompt cut.

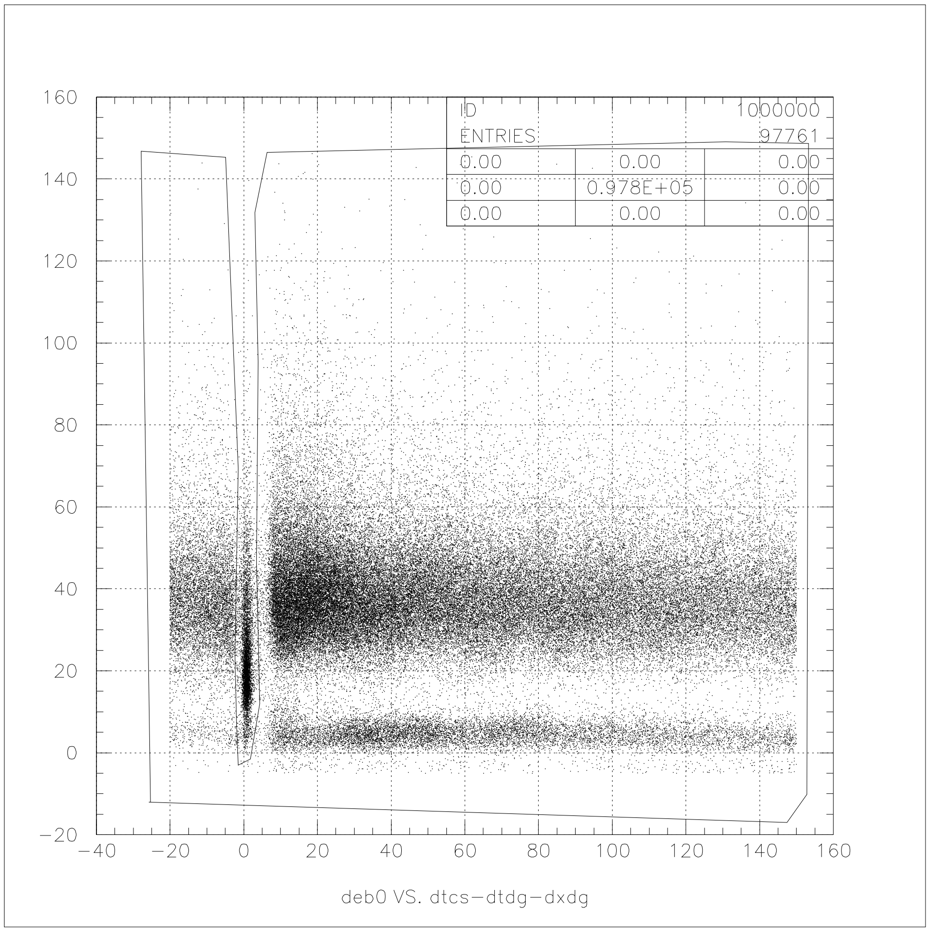


Figure 7.20: Prompt cut using DSC B0 ADC.

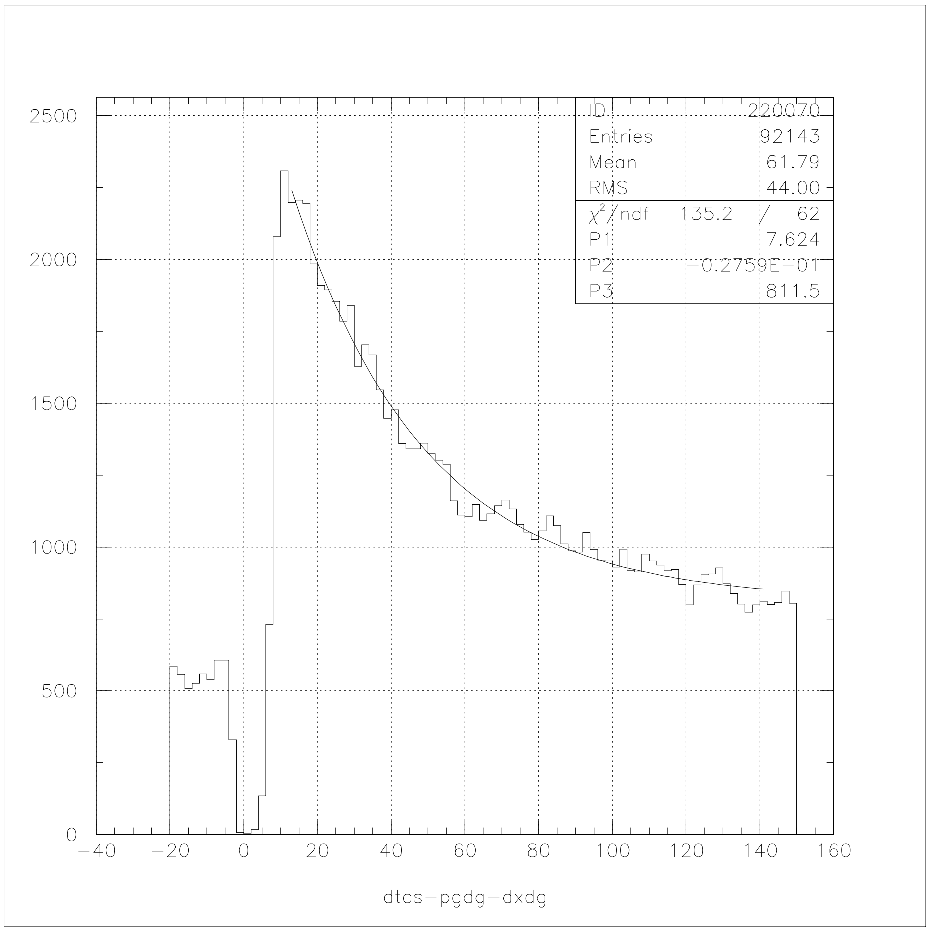


Figure 7.21: $\pi^+ \to e^+\nu$ timing spectrum using DSC B0 prompt cut.

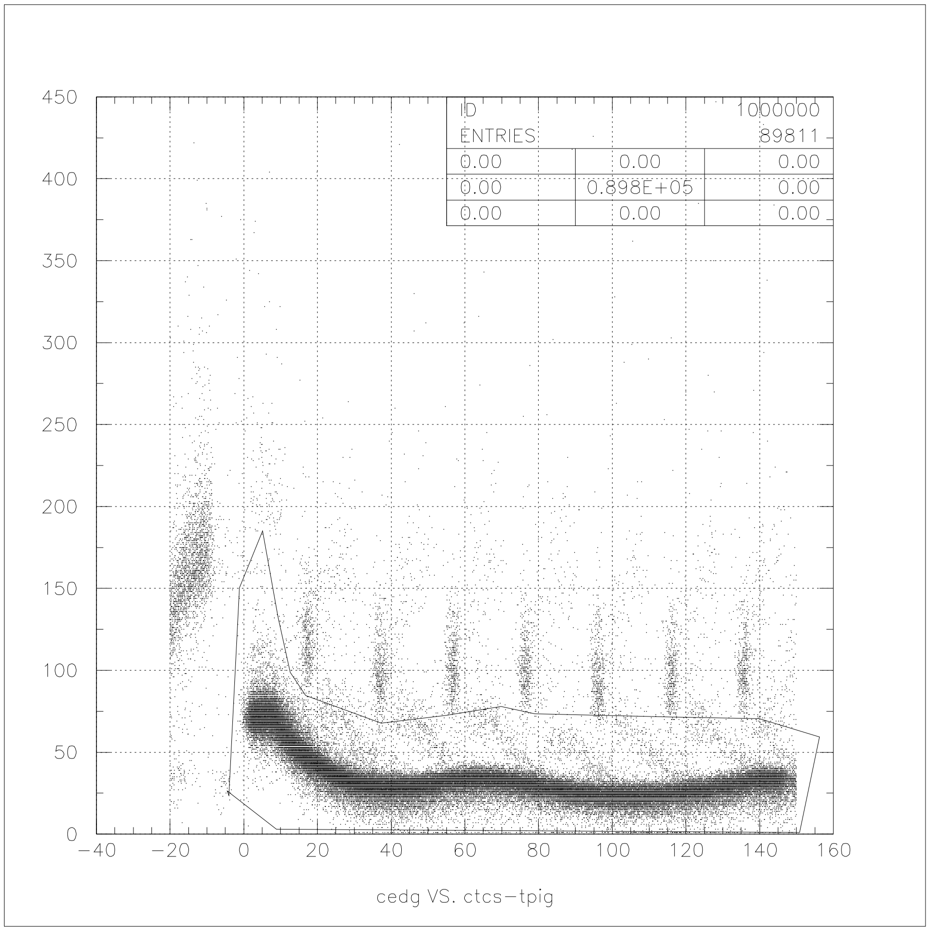


Figure 7.22: Prompt cut using DSC B0 ADC.

spectrum with such cut.

7.2.2 Two-dimensional plot of DSC vs FASTBUS ADC/TDC timing for $\pi^+ \to e^+\nu$ analysis

This analysis is just the $\pi^+ \to e^+\nu$ analog of previous section's PIBETA corresponding analysis.

Fig. 7.24 shows how the two-dimensional cut of DSC timing vs corresponding FASTBUS timing is defined. Fig. 7.25 is the corresponding $\pi^+ \to e^+\nu$ timing

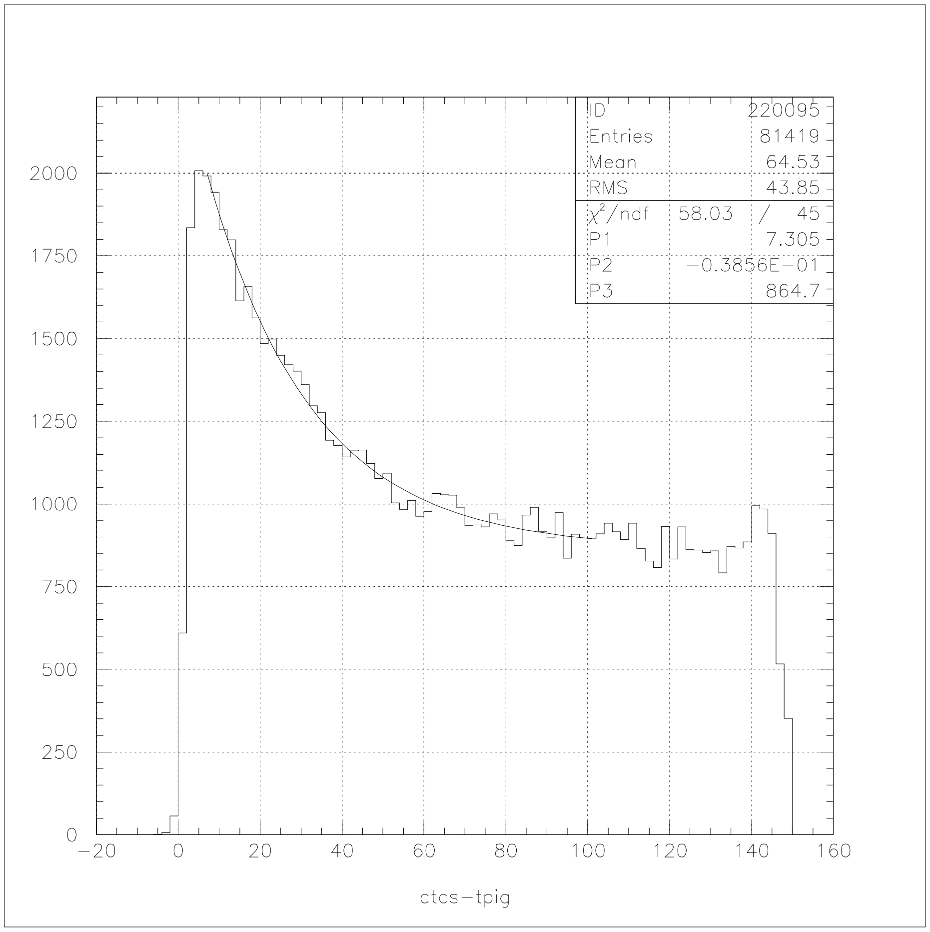


Figure 7.23: $\pi^+ \rightarrow e^+\nu$ fast bus timing spectrum using CAMAC degrader prompt cut.

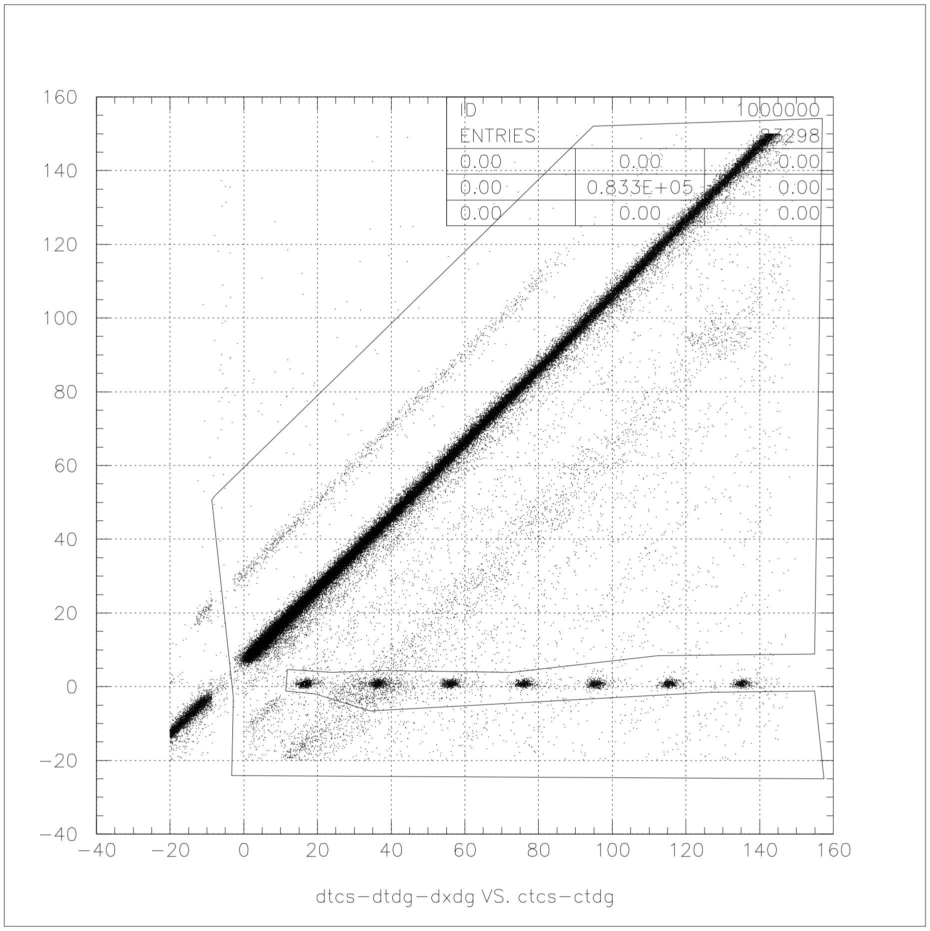


Figure 7.24: DSC timing difference of Calorimeter and degrader vs that of FASTBUS timing difference.

spectrum with such cut.

7.2.3 Positron spectrum of energy deposition in target from $\pi^+ \to e^+\nu$ decay

This part is the $\pi^+ \to e^+\nu$ analog of previous corresponding PIBETA analysis. Although the detected particle e^+ is same in both cases, the present $\pi^+ \to e^+\nu$ positron differs from e^+ of $\pi^+ \to \pi^0 e^+\nu$ since it is energetic (> 50MeV) and escapes the target with minimum ionization. From the geometry of the PIBETA target, it

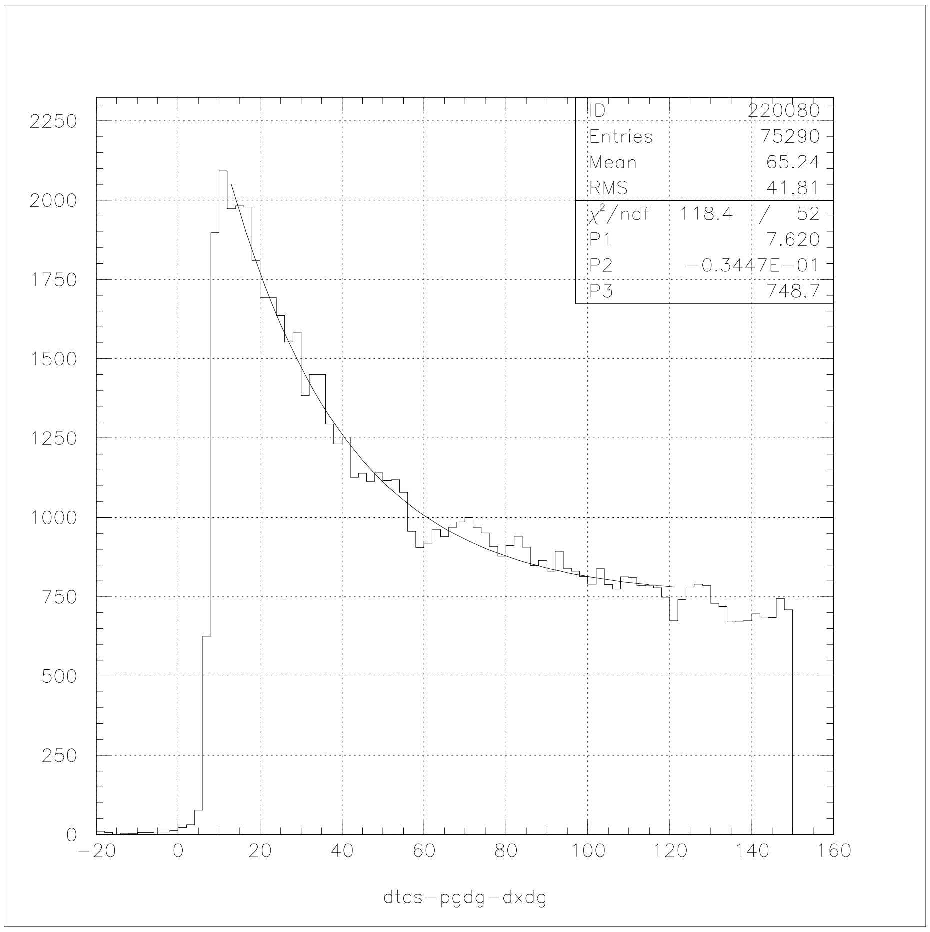


Figure 7.25: $\pi^+ \to e^+\nu$ timing spectrum for 2-D DSC vs FASTBUS timing cut.

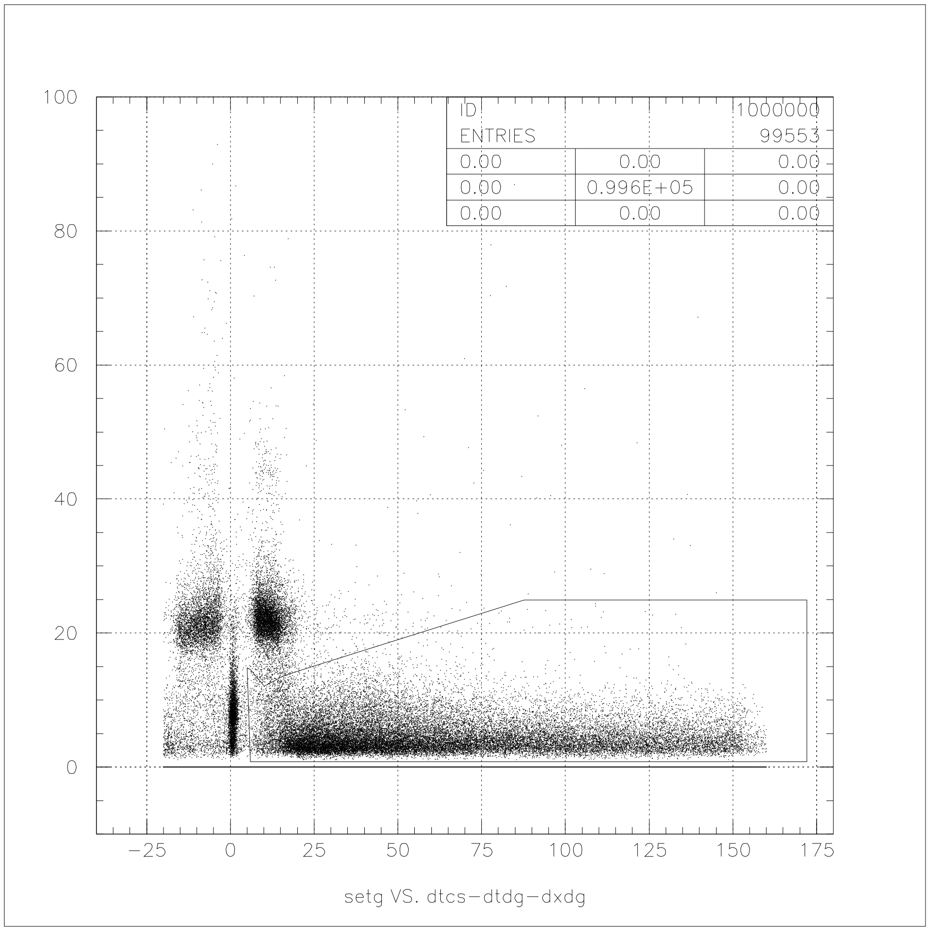


Figure 7.26: Summed target energy vs timing difference of Calorimeter and degrader.

will deposit about 4 MeV energy in the target.

Fig. 7.26 shows cut on two- dimensional plot of DSC summed target energy vs DSC timing difference between Calorimeter and degrader. This cut separates out very clean e^+ events.

Applying the cuts just defined, Fig. 7.27 shows the e^+ summed target energy from CAMAC (top one) and from DSC (bottom one). We can see that DSC has better resolution, that is, smaller $\sigma/\overline{E}$ value. (For camac, $\sigma/\overline{E}$ = 37.7%, while for DSC

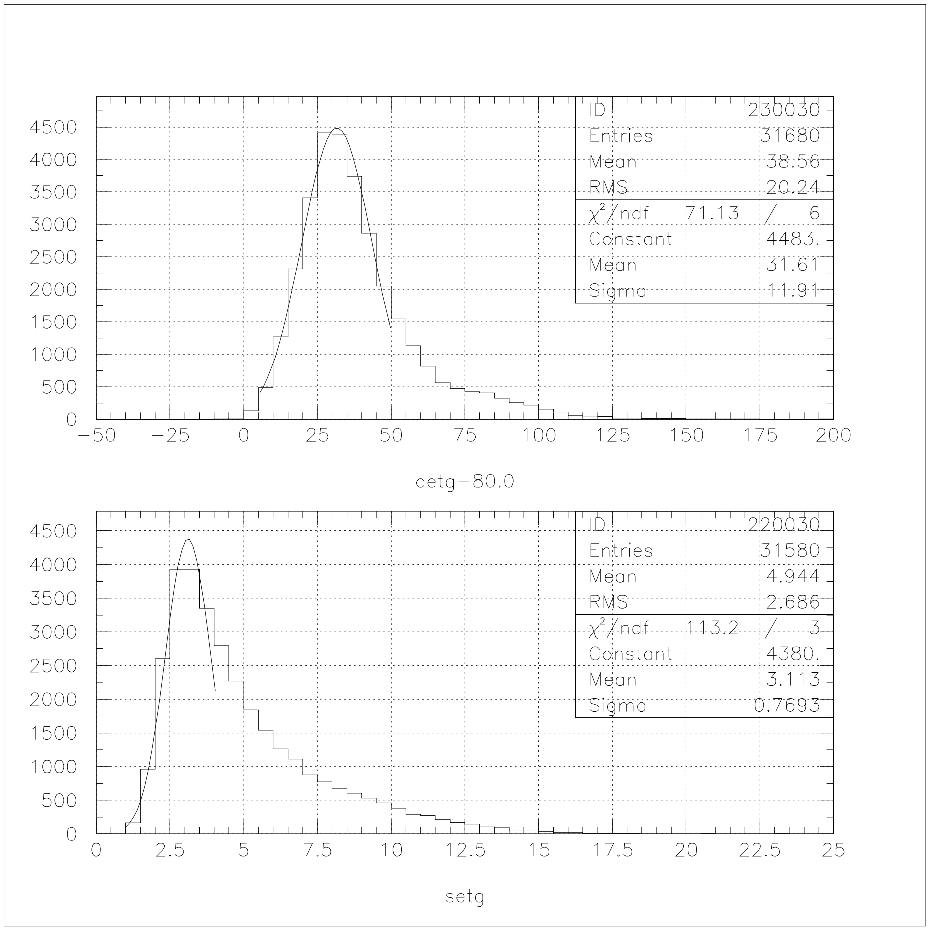


Figure 7.27: Summed e^+target energy of CAMAC (top) and DSC (bottom).

$\sigma/\overline{E} = 24.7\%$.)

Chapter 8

Results and Discussions

We choose DSC timing spectrum to extract $\pi^+ \to \pi^0 e^+ \nu$ and $\pi^+ \to e^+ \nu$ events. The cuts applied are exactly the same as ones used in [1], just replacing every FASTBUS variable with corresponding DSC one. This analysis is done on the data from mid year 2000 to the end of PIBETA experiment when digitizing system are applied to all detectors, specifically CsI calorimeters modules. That data sample with full DSC takes about two thirds of the whole PIBETA data.

The fitting algorithms used to extract $\pi^+ \to e^+ \nu$ and $\pi^+ \to \pi^0 e^+ \nu$ events are the same as that used in [1]. The final and complete DSC $\pi^+ \to \pi^0 e^+ \nu$ timing spectrum is shown in Fig 8.1. The final and complete DSC $\pi^+ \to e^+ \nu$ timing spectrum is shown in Fig 8.2.

In order to determine $\pi^+ \to \pi^0 e^+ \nu$ and $\pi^+ \to e^+ \nu$ gate fractions and their uncertainties, we need to determine time zero position and error for $\pi^+ \to \pi^0 e^+ \nu$ and $\pi^+ \to e^+ \nu$ respectively. Proceeding in a same manner as in [1], we work on prompt timing spectrum with corresponding DSC time. Fig. 8.3 show the prompt

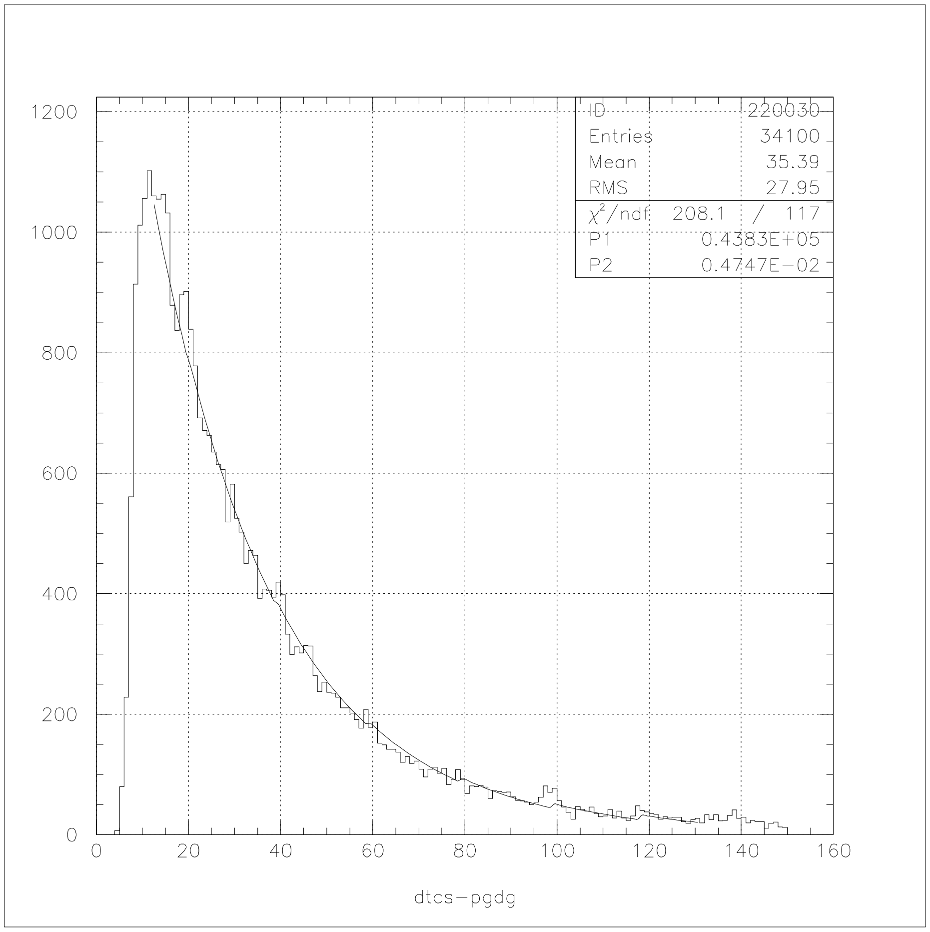


Figure 8.1: Final DSC $\pi^+ \rightarrow \pi^0 e^+ \nu$ timing spectrum.

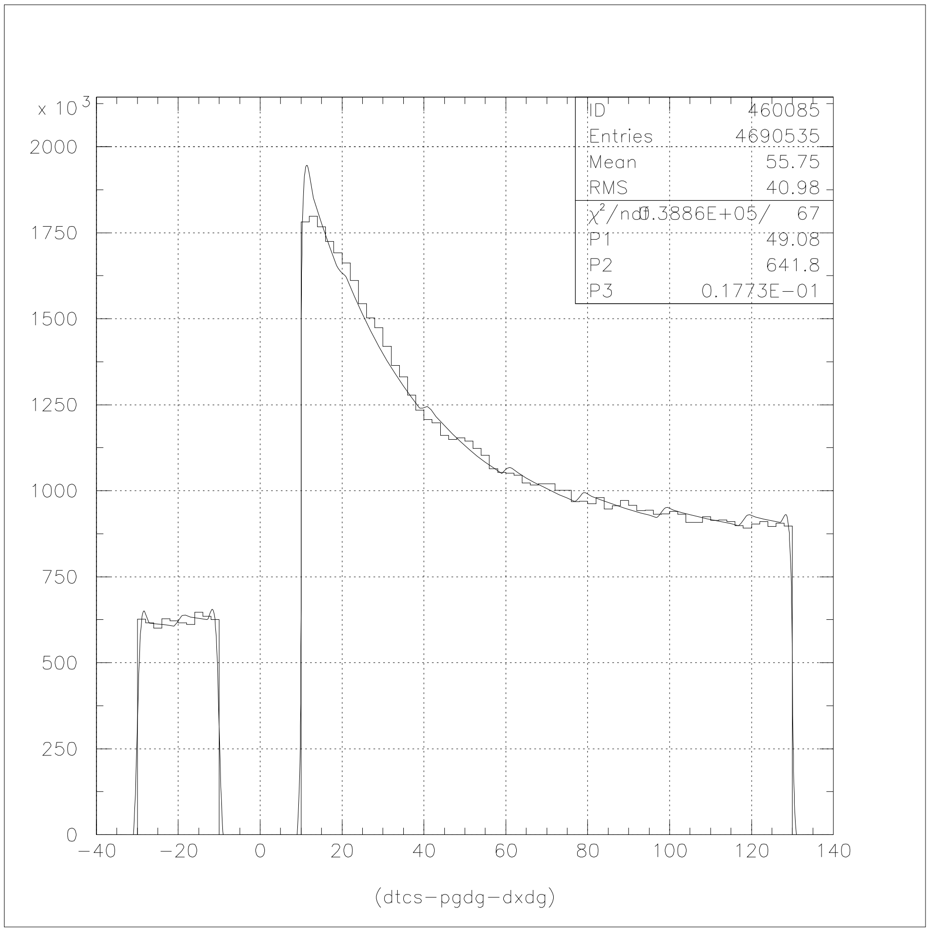


Figure 8.2: Final DSC $\pi^+ \to e^+\nu$ timing spectrum.

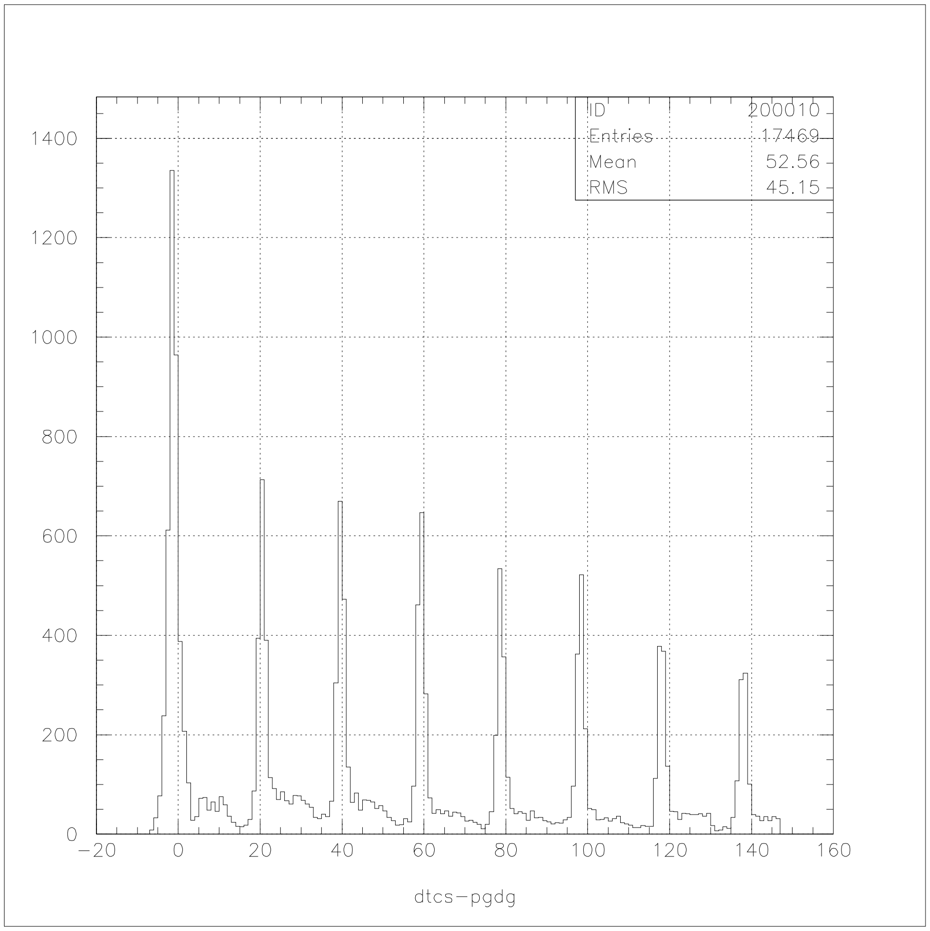


Figure 8.3: DSC $\pi^+ \to \pi^0 e^+ \nu$ Prompt timing spectrum.

spectrum from $\pi^+ \to \pi^0 e^+ \nu$ candidate events. Fig. 8.4 show the prompt spectrum from $\pi^+ \to e^+ \nu$ candidate events.

Fitting around prompt peaks, we find the time zero positions as follows:

for $\pi^+ \to \pi^0 e^+ \nu$, $t_0 = 0.25 \pm 0.019$ns.

for $\pi^+ \to e^+ \nu$, $t_0 = 0.64 \pm 0.011$ns.

We use the same formula as [1] to calculate $\pi^+ \to \pi^0 e^+ \nu$ branching ratio:

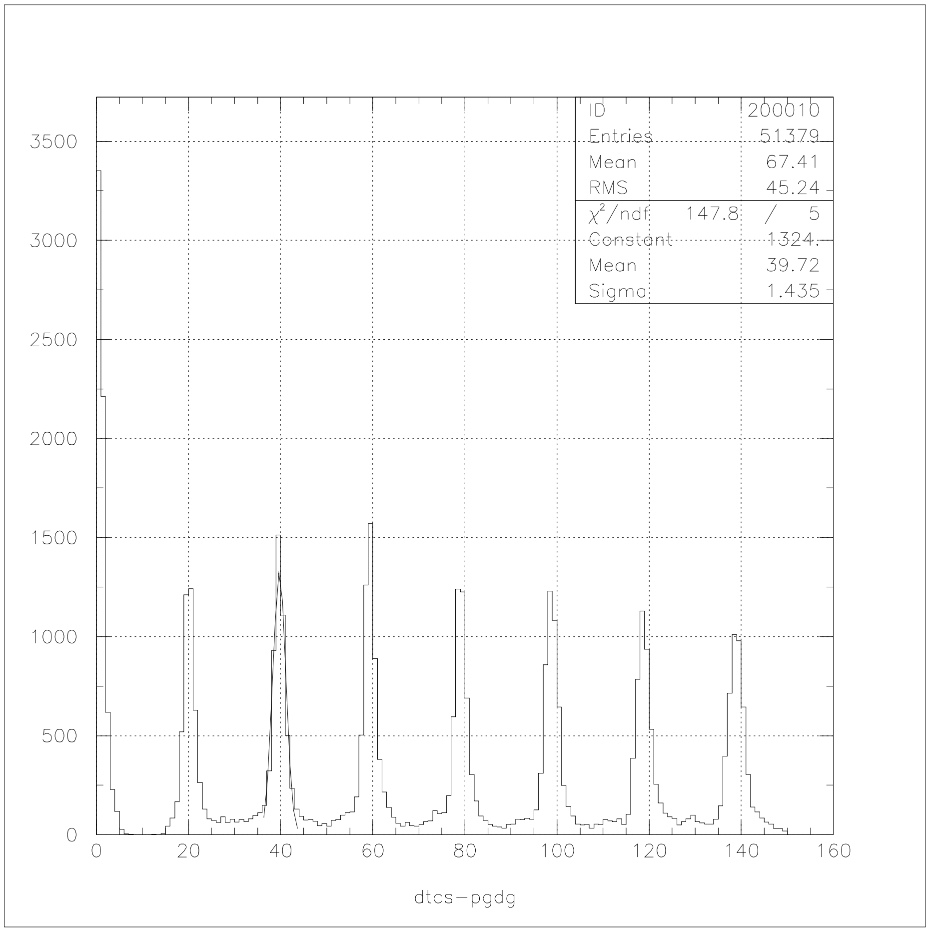


Figure 8.4: DSC $\pi^+ \to e^+\nu$ Prompt timing spectrum.

$$\begin{aligned}\Gamma_{\pi\beta} &= \frac{\mathrm{N}_{\pi\beta}\Gamma_{\pi 2\mathrm{e}}}{\mathrm{N}_{\pi 2\mathrm{e}}} \\ &= \frac{1}{\tau}\cdot\frac{\mathrm{N}_{\pi\beta}}{\mathrm{g}_{\pi\beta}\mathrm{A}_{\pi\beta}\Gamma_{\pi^0}}\cdot\frac{\mathrm{g}_{\pi 2\mathrm{e}}\mathrm{A}_{\pi 2\mathrm{e}}\mathrm{e}_{\mathrm{pv}}\mathrm{e}_{\mathrm{ch1}}\mathrm{e}_{\mathrm{ch2}}\Gamma_{\pi 2\mathrm{e}}}{20\mathrm{N}_{\pi 2\mathrm{e}}}\end{aligned} \tag{8.1}$$

The calculation of DSC branching ratio as well as its combination with non-DSC one is summarized in Table 8.1.

Discussion of results

The value of the branching ratio for the $\pi^+ \to \pi^0 e^+ \nu$, found in this analysis by normalizing to the experimental $\Gamma_{\pi^+\to\mathrm{e}^+\nu}$ is

$$\Gamma_{\pi^+\to\pi^0\mathrm{e}^+\nu} = [1.043 \pm 0.004(\mathrm{stat}) \pm 0.004(\mathrm{syst}) \pm 0.003(\pi^+ \to \mathrm{e}^+\nu)] \times 10^{-8}$$

When normalizing to theoretical $\Gamma_{\pi^+\to e^+\nu}$ from Marciano and Sirlin ($\Gamma_{\pi^+\to e^+\nu} = (1.2352 \pm 0.0005) \times 10^{-4}$, 1993), the branching ratio of $\pi^+ \to \pi^0 e^+ \nu$ is:

$$\Gamma_{\pi^+\to\pi^0\mathrm{e}^+\nu} = [1.047 \pm 0.004(\mathrm{stat}) \pm 0.004(\mathrm{syst}) \pm 0.0004(\pi^+ \to \mathrm{e}^+\nu)] \times 10^{-8}$$

The DSC analysis affects the internal systematic error in the determination of the branching ratio of pion beta decay, leaving the statistical and external sources of uncertainty unchanged. This thesis concetrates mostly on introducing DSC analysis methode rather on detailed description of how to obtain the physics result- $\pi^+ \to \pi^0 e^+ \nu$ branching ratio which is already obtained with excellent accuracy in [1]. Hope is that DSC work done in PIBETA experiment(described in chapter 4-7) will pave a way for an subsequent experiment to get the most accurate $\pi^+ \to e^+\nu$ branching ratio in PSI. In this effort DSC will play a vital and indispensable role (in event identification, for example).

Table 8.1: Variables for $\pi^+ \to \pi^0 e^+ \nu$ branching ratio calculation

	non-DSC	DSC	weighted mean	Remarks
$N_{\pi\beta}$	23140 ± 152	35210 ± 188	58350 ± 241	Number of $\pi^+ \to \pi^0 e^+ \nu$ events
$g_{\pi\beta}$	0.704 (2)	0.714 (2)	0.710 (2)	$\pi^+ \to \pi^0 e^+ \nu$ gate fraction
$A_{\pi\beta}$	0.6612 (7)	0.6622 (7)	0.6618 (7)	$\pi^+ \to \pi^0 e^+ \nu$ acceptance
Γ_{π^0}		0.9880 (3)		$\pi^0 \to \gamma\gamma$ branching ratio
$N_{\pi e\nu}(\times 10^7)$	1.170 (2)	2.250 (2)	3.420 (4)	$\pi^+ \to e^+ \nu$ events
$g_{\pi e\nu}$	0.6042 (9)	0.6122 (3)	0.6091 (6)	$\pi^+ \to e^+ \nu$ gate fraction
$A_{\pi e\nu}$	0.7035 (6)	0.7041 (2)	0.7039 (4)	$\pi^+ \to e^+ \nu$ acceptance
e_{pv}	0.9890 (1)	0.9839 (1)	0.9876 (1)	Plastic veto efficiency
e_{ch1}	0.9454 (1)	0.9379 (1)	0.9372 (1)	Chamber 1 efficiency
e_{ch2}	0.9792 (1)	0.9745 (1)	0.9781 (1)	Chamber 2 efficiency
$\Gamma_{\pi e\nu}$		$1.230\,(4) \times 10^{-4}$		$\pi^+ \to e^+ \nu$ branching ratio
$\Gamma_{\pi\beta}(\times 10^{-8})$	1.031 ± 0.008	1.051 ± 0.008	1.043 ± 0.006	$\pi^+ \to \pi^0 e^+ \nu$ branching ratio

Appendix A

Program to generate skimmed MIDAS file

Following C program was used to compress original MIDAS files to those that contain the interested candidate events. This program's running require input data file that contains list of event id numbers following each run number.

```
#include <stdio.h>
#define NUME 10000
#define N4 4
#define M4 4
#define SE 4
#define SD 12
#define SI 0
#define EVENT_ID_EOR 0x8001
#define Npath  54  //52
//int event[NUME];


char pathname_mid[Npath][35]={
  "/scratch14/data97/", "/scratch14l/data97/","/scratch14l/data98/",
  "/scratch14m/data98/","/scratch1/data99/",   "/scratch2/data99/",
  "/scratch3/data99/",  "/scratch4/data99/",   "/scratch5/data99/",
  "/scratch6/data99/",  "/scratch7/data99/",   "/scratch8/data99/",
  "/scratch9/data99/",

  "/scratch9/data00/",  "/scratch10/data00/",  "/scratch11/data00/",
  "/scratch12/data00/", "/scratch1l/data00/",  "/scratch2l/data00/",
  "/scratch3l/data00/", "/scratch4l/data00/",  "/scratch5l/data00/",
```

```
  "/scratch6l/data00/", "/scratch7l/data00/",  "/scratch8l/data00/",
  "/scratch9l/data00/", "/scratch10l/data00/", "/scratch11l/data00/",
  "/scratch12l/data00/","/scratch13l/data00/", "/scratch10m/data00/",
  "/scratch11n/data00/",

  "/scratch1l/data01/", "/scratch5n/data01/",  "/scratch6n/data01/",
  "/scratch7n/data01/", "/scratch8n/data01/",
  "/scratch9n/data01/",
  "/scratch13m/data01/chunck2/",
  "/scratch9n/data01/",
  "/scratch10n/data01/", "/scratch11m/data01/",
  "/scratch12n/data01/","/scratch5m/data01/",  "/scratch6m/data01/",

  "/scratch13m/data01/chunck1/",
  "/scratch6m/data01/",
  "/scratch7m/data01/", "/scratch8m/data01/",  "/scratch9m/data01/",

  /* add 2/17/05 */
  "/scratch13m/data01/chunck3/",
  "/scratch9m/data01/",
  /* add end */

  "/scratch10l/data00/longruns/",
  "/scratch11l/data00/lr/",
};

int runnum_mid[Npath]={54,345,10018,10153,20081,20150,20380,20590,
                    20775,21130,21480,21878,22217,30000,30150,30410,
                    30680,30950,31220,31420,31690,35130,35310,35490,
                    35670,35830,35980,36225,36390,36560,36720,36950,
                    40180,40620,40890,41150,42060,42330,
       42420,42470,
                    42700,42950,
                    43200,43500,43710,
                    43900,43940,
                    43960,44160,44370,
       /* add 2/17/05 */
       44430,44520,
                       /* add end */
                    361133,361171,
                   };
```

```
char f99out_name[14][20]={"wy_99node1.dat", "wy_99node2.dat", "wy_99node3.dat",
                   "wy_99node4.dat", "wy_99node5.dat", "wy_99node6.dat",
                   "wy_99node7.dat", "wy_99node8.dat", "wy_99node9.dat",
                   "wy_99node10.dat","wy_99node11.dat","wy_99node12.dat",
                   "wy_99node13.dat","wy_99node14.dat",
                  };

char f00out_name[14][20]={"wy_00node1.dat", "wy_00node2.dat", "wy_00node3.dat",
                   "wy_00node4.dat", "wy_00node5.dat", "wy_00node6.dat",
                   "wy_00node7.dat", "wy_00node8.dat", "wy_00node9.dat",
                   "wy_00node10.dat","wy_00node11.dat","wy_00node12.dat",
                   "wy_00node13.dat","wy_00node14.dat",
                  };

char f01out_name[14][20]={"wy_01node1.dat", "wy_01node2.dat", "wy_01node3.dat",
                   "wy_01node4.dat", "wy_01node5.dat", "wy_01node6.dat",
                   "wy_01node7.dat", "wy_01node8.dat", "wy_01node9.dat",
                   "wy_01node10.dat","wy_01node11.dat","wy_01node12.dat",
                   "wy_01node13.dat","wy_01node14.dat",
                  };


FILE *f_p;
char path2[55];
void find_file(int runn)
{int i,i1,i2,ip,itemp,factor,runnum;
 char temp[15];//path2[40];
 i1=0;
 //i2=Npath-1;
 i2=Npath;
 while((i2-i1)>1)
   {itemp=(i1+i2)/2;
   if(runn<runnum_mid[itemp]) i2=itemp;
   else i1=itemp;
   }

 strcpy(path2,pathname_mid[i1]);

 runnum=runn;
  temp[0]='r';temp[1]='u';temp[2]='n';
```

```
  /*
 factor=10000;
 for(i=0;i<5;i++)
   {//if(factor<1) return 0;
    ip=runnum/factor; temp[3+i]=(char)(48+ip);
   runnum-=ip*factor; factor/=10;
   }
 temp[8]='.';temp[9]='m';temp[10]='i';temp[11]='d'; temp[12]='\0';

  */

if(runnum<100000)
  {
   factor=10000;
   for(i=0;i<5;i++)
    {//if(factor<1) return 0;
    ip=runnum/factor; temp[3+i]=(char)(48+ip);
    runnum-=ip*factor; factor/=10;
    }
   temp[8]='.';temp[9]='m';temp[10]='i';temp[11]='d'; temp[12]='\0';
  }
 else
   {
   factor=100000;
   for(i=0;i<6;i++)
    {//if(factor<1) return 0;
    ip=runnum/factor; temp[3+i]=(char)(48+ip);
    runnum-=ip*factor; factor/=10;
    }
   temp[9]='.';temp[10]='m';temp[11]='i';temp[12]='d'; temp[13]='\0';
   }

 strcat(path2,temp);

 f_p=fopen(path2,"r");
}

FILE *f_pout;
int i_year;
char outmid_name[50];
void create_file(int runn,char fpath[])
```

```
{int i,ip,runnum,factor;
 char temp[20],temp2[40];
 strcpy(temp2,fpath);

 runnum=runn;
 // temp[0]='p';temp[1]='b';
 temp[0]='e'; temp[1]='u';
  if(i_year==1)      {temp[2]='9';temp[3]='9';}
  else if(i_year==2) {temp[2]='0';temp[3]='0';}
  else               {temp[2]='0';temp[3]='1';}
 temp[4]='_';
 if(runnum<100000)
 {
  factor=10000;
  for(i=0;i<5;i++)
   {//if(factor<1) return 0;
    ip=runnum/factor; temp[5+i]=(char)(48+ip);
   runnum-=ip*factor; factor/=10;
   }
  temp[10]='.';temp[11]='m';temp[12]='i';temp[13]='d'; temp[14]='\0';
 }
 else
 {
  factor=100000;
  for(i=0;i<6;i++)
   {//if(factor<1) return 0;
    ip=runnum/factor; temp[5+i]=(char)(48+ip);
   runnum-=ip*factor; factor/=10;
   }
  temp[11]='.';temp[12]='m';temp[13]='i';temp[14]='d'; temp[15]='\0';
 }

 strcat(temp2,temp);
 strcpy(outmid_name,temp2);
 f_pout=fopen(temp2,"w");

}

int gene_fname(char path[],char path2[],int runnum)
{int i,factor,ip;
 char temp[15];
```

```
 strcpy(path2,path);
 if (runnum>=50000) return 0;
 temp[0]='r';temp[1]='u';temp[2]='n';
 factor=10000;
 for(i=0;i<5;i++)
   {if(factor<1) return 0;
    ip=runnum/factor; temp[3+i]=(char)(48+ip);
   runnum-=ip*factor; factor/=10;
   }
 temp[8]='.';temp[9]='m';temp[10]='i';temp[11]='d'; temp[12]='\0';
 strcat(path2,temp);
 // printf("File name with path:%s\n",path2);
 return 1;
}
int order(int *p,int n)
{int i,j,ev[NUME],tp;
 for(i=0;i<n;i++) ev[i]=p[i];
 for(i=0;i<n-1;i++)
   {for(j=0;j<n-1-i;j++)
     {if(ev[j]>ev[j+1]){tp=ev[j];ev[j]=ev[j+1];ev[j+1]=tp;}
     }
   }
 //for(i=0;i<n;i++) printf("%d ",ev[i]); printf("\n");
 i=0;
 while(i<n-1)
   {if(ev[i]==ev[i+1]){for(j=i+1;j<n-1;j++) ev[j]=ev[j+1]; n--;}
   else {i++;}
   }
 for(i=0;i<n;i++) p[i]=ev[i];
 return n;
}
main()
{FILE *in1,*in2,*out;
 char fname1[40],//="wyeve1.dat",
   fname2[40],//="/data/weidong/run36042.mid",
   fname3[40];//="wydata.mid";
 /* add variable for wymid3.c */
 char pathfname[50],pathfname2[50],runname[15],yn;
 int runnumber,run_low,run_up,n_run,i_run,ntmp,wflag=1;
 /* add end */
 int i,j,ne=0,iflag=0;
```

```
 int event[NUME];
 int ds,si,sn,sp,nn,factor=1;
 char ctp,cc[N4*M4+4];

 int i_node;
 do
   {printf("Which year's data to be analyzed?(1='99',2='00',3='01'):");
    scanf("%d",&i_year);
    printf("Which node's data to be analyzed?(1-14):");
    scanf("%d",&i_node);
   }while(i_node<1||i_node>14||i_year<1||i_year>3);

 strcpy(fname1,"/home/yw4f/mid_work/eu_skim/");
 if(i_year==1)
   {strcat(fname1,f99out_name[i_node-1]);
    strcpy(fname3,"/home/yw4f/mid_work/mid_pb/pb99_mid/");
   //strcpy(fname3,"/home/yw4f/cwork/pb99_mid/");
   //strcpy(fname3,"");
    }
 else if (i_year==2)
    {strcat(fname1,f00out_name[i_node-1]);
     strcpy(fname3,"/home/yw4f/mid_work/mid_pb/pb00_mid/");
    //strcpy(fname3,"/home/yw4f/cwork/pb00_mid/");
    //strcpy(fname3,"");
    }
 else { strcat(fname1,f01out_name[i_node-1]);
       strcpy(fname3,"/home/yw4f/mid_work/mid_pb/pb01s_mid/");
       printf("Enter output directory:");
       scanf("%s",fname3);
      //strcpy(fname3,"/home/yw4f/cwork/pb01_mid/");
      //strcpy(fname3,"");
      }

 in1=fopen(fname1,"r");
 if(in1==NULL) {printf("Can not open file '%s'\n",fname1);return;}


 fscanf(in1,"%d",&n_run);

/* read eventid form file named fname1 */
 wflag=1;
```

```
for(i_run=0;i_run<n_run;i_run++)
  {
   fscanf(in1,"%d",&runnumber);

   fscanf(in1,"%d",&ne);
   if(ne<1) {printf("Invalid data(number of event) in file '%s'\n",fname1);
           goto endf;}
   //else if(ne>NUME) ne=NUME;
   printf("ne=%d\n",ne);
   for(i=0;i<ne;i++)
     {if(i<NUME)
      fscanf(in1,"%d",&event[i]);
     else fscanf(in1,"%d",&ntmp);
     }
   ne=order(event,ne); //put event id number in increasing order
 //printf("ne=%d\n",ne);
 //for(i=0;i<ne;i++) printf("%d ",event[i]); printf("\n");
 /* read file end */

   //if(gene_fname(pathfname,pathfname2,runnumber)!=1)
   //{printf("Illegal runnumber found!\n");return;}

   //in2=fopen(pathfname2,"r");
   find_file(runnumber);
 if(f_p==NULL) {printf("Cannot open file '%s'\n",path2);goto jump2;}
 else {printf("File '%s' loaded successfully!\n",path2); in2=f_p;}

 create_file(runnumber,fname3);
 if(f_pout==NULL){printf("Cannot create file '%s'!\n",outmid_name);
                 goto jump1;
                 }
 else{out=f_pout;}

 for(i=0;i<N4*M4;i++)
   { cc[i]=fgetc(in2); if(wflag) fprintf(out,"%c",cc[i]);}
 factor=1;ds=0;
 for(i=0;i<N4;i++)
   {if((int)cc[SD+i]>=0) ds+=factor*(int)cc[SD+i];
   else ds+=factor*(256+(int)cc[SD+i]);
   factor*=256;
   }
```

```
for(i=0;i<ds;i++){ctp=fgetc(in2);if(wflag) fprintf(out,"%c",ctp);}
wflag=1;

nn=0;sp=0;
while(nn<=ne&&!feof(in2))
  {for(i=0;i<N4*M4;i++) cc[i]=fgetc(in2);
  sn=0;factor=1;
  for(i=0;i<N4;i++)
    {if((int)cc[SE+i]>=0) sn+=factor*(int)cc[SE+i];
    else sn+=factor*(256+(int)cc[SE+i]);
    factor*=256;
    }
  ds=0;factor=1;
  for(i=0;i<N4;i++)
    {if((int)cc[SD+i]>=0) ds+=factor*(int)cc[SD+i];
    else ds+=factor*(256+(int)cc[SD+i]);
    factor*=256;
    }
  if(sn==event[nn])
    {for(i=0;i<N4*M4;i++) fprintf(out,"%c",cc[i]);
    for(i=0;i<ds;i++){ctp=fgetc(in2);fprintf(out,"%c",ctp);}
    iflag=1;
    //printf("sn=%d\n",sn);
    }
  else
    {if((sn-sp)<0&&(sn-sp)>2)
      {printf("Unexpected inconsistancy!\n");goto endf;}
    if(sn>event[nn]||iflag) {iflag=0;nn++;}
    if(sn==event[nn])
      {for(i=0;i<N4*M4;i++) fprintf(out,"%c",cc[i]);
    for(i=0;i<ds;i++){ctp=fgetc(in2);fprintf(out,"%c",ctp);}
    iflag=1;
    //printf("sn=%d\n",sn);
      }
    else {for(i=0;i<ds;i++) ctp=fgetc(in2);}
    }
  sp=sn;
  }

/* yw4f add to get tail odb information */
iflag=1;
```

```
while(iflag&&!feof(in2))
  {


  for(i=0;i<N4*M4;i++) cc[i]=fgetc(in2);

  si=0;factor=1;
  for(i=0;i<N4/2;i++)
    {if((int)cc[SI+i]>=0) si+=factor*(int)cc[SI+i];
    else si+=factor*(256+(int)cc[SI+i]);
    factor*=256;
    }

  sn=0;factor=1;
  for(i=0;i<N4;i++)
    {if((int)cc[SE+i]>=0) sn+=factor*(int)cc[SE+i];
    else sn+=factor*(256+(int)cc[SE+i]);
    factor*=256;
    }
  ds=0;factor=1;
  for(i=0;i<N4;i++)
    {if((int)cc[SD+i]>=0) ds+=factor*(int)cc[SD+i];
    else ds+=factor*(256+(int)cc[SD+i]);
    factor*=256;
    }
  if(si==EVENT_ID_EOR&&sn==runnumber)
  //if(sn==runn)
    {printf("Found ODB dump!\n");
      // printf("Found runn=%d, EventID=%d\n",runn,si);
     for(i=0;i<ds;i++)
     {ctp=fgetc(in2); fprintf(out,"%c",ctp);
     }
     iflag=0;
    }
   else
    {for(i=0;i<ds;i++) ctp=fgetc(in2);}
 }
if(iflag) printf("ODB dump not found!\n");
/* yw4f add end */

printf("File %s successfully created!\n",outmid_name);
```

```
 fclose(out);
 jump1:  fclose(in2);
 jump2: iflag=0;
}
 printf("End of Program!\n");
 endf:
 fclose(in1);
}
```

Appendix B

Revised fitting routine for CDSC fitting using different algorithm

Following C code gives alternative fitting routines for CDSC waveform fitting. The first one is optimized for multi-hit channels like B0, Degrader, Target. The second one is optimized for single-hit channel like CsI.

```
#define Nlimi 100
#define Narr 127
#define Nwave 130

#define Nge 4 //6
#define Nma 2
#define Nstep1 2  //3
#define Nstep2 4  //5

#define PP4 4 //12

/* add 1/19/05 */
#define b0_lim1 61.8 //63.0
#define b0_lim2 78.0 //77.0
#define thresh2 9.0

#define chisqr_lim 0.14 //0.17 0.18
#define PP8 8

/* add 1/26/05 */
#define YW_pth1 15.0
#define YW_pth2 12.0
```

```
/* add end */

int maxp(int j1,int j2)
{if(j1>j2) return j1; else return j2;}
int minp(int j1,int j2)
{if(j1<j2) return j1; else return j2;}
/* add end */


int dsc_fit(double yr[],double yf[],int nl,double param[])
{
  int i,j;
  double        error[N_PARAM], chisqr,
             yd[127], yfit[127], weight[127];
  static double x[127];

  int ip[Nlimi],dleft[Nlimi],dright[Nlimi],npeak,i1,i2;
  double ratio[Nlimi],tp1,tp2;
  double xp[Nlimi],yamp[Nlimi];
  int protect[Narr];
  int k1,k2;

  int il,p1,p2,flag=0;

  double mv,nv;
  int itp,Nrange;
  int Nstep;

  /* add 1/19/05 */
  double param2[Nlimi];
  int npeak2;
  float chisqr02;
  double x2[127],yd2[127],yfit2[127];
  double param_tp[2];

  float chisqr2;

  // add 1/24/05
  double chisqr00;
  /* add end */
```

```
  /* add 3/24/05 */
  int ct_refit;
  /* add end */

 for(i=0;i<nl;i++)
      {
yd[i]=yr[i];//(double)(Y_BASE-yr[i]);
       x[i]=(double)i;
      }
 for(i=0;i<Narr;i++) protect[i]=0;
 npeak=0;

 il=4; flag=0; Nrange=Nge;

 while(il<126) // il<122
   { flag=0;
     p1=il-Nrange/2; p2=il+Nrange/2;
     if(p1<0) p1=0;
     if(p2>126) p2=126;
     mv=-1000.0;
     for(j=p1;j<=p2;j++)
       {if(yd[j]>=mv) {mv=yd[j]; i=j;}}

     if(i<il+Nrange/2&&i>il-Nrange/2&&mv>1.0&&!protect[i]) {flag=1;}
       else {il=il+Nrange/2;}
// if(yd[i]>=yd[i-1]&&yd[i]>=yd[i+1]&&2.0*yd[i]>(yd[i-1]+yd[i+1]))
    if(flag)
     {ip[npeak]=i;
     //i1=i-1;
     // while(i1>0&&yd[i1-1]<yd[i1]&&yd[i1]>0.0) i1--;
     // if(yd[i1]==0.0) i1++;
     // i2=i+1;
     // while(i2<Narr-1&&yd[i2+1]<yd[i2]&&yd[i2]>0.0) i2++;
     // if(yd[i2]==0.0) i2--;
     i1=i; itp=i1-1;

     /* add1 */
     Nstep=Nstep2;
     /* add1 end */
     while(itp<i1)
       {i1=itp; nv=yd[itp];
```

```
/* add */
        if(i-i1>1) Nstep=Nstep1;
/* add end */
       for(j=i1;j>i1-Nstep; j--)
         {if(j<0||yd[j]==0.0) break;
 if(yd[j]<nv) {nv=yd[j];itp=j;}
         }
       }
     i2=i; itp=i2+1;

     /* add1 */
     Nstep=Nstep2;
     /* add1 end */
     while(itp>i2)
       {i2=itp; nv=yd[itp];
       /* add1 */
       if(i2-i>1) Nstep=Nstep1;
       /* add1 end */
       for(j=i2;j<i2+Nstep;j++)
         {if(j>126||yd[j]==0.0) break;
 if(yd[j]<nv) {nv=yd[j]; itp=j;}
         }
       }
     for(j=i1;j<i2;j++) protect[j]=1;
      dleft[npeak]=i-i1; dright[npeak]=i2-i;
      if(i-i1>9) tp1=(yd[i]-yd[i-9])/(1.0-ynorm[f_channel][0]);
      else tp1=(yd[i]-yd[i1])/(1.0-ynorm[f_channel][X_PEAK-i+i1]);
      if(i2-i>16) tp2=(yd[i]-yd[i+16])/(1.0-ynorm[f_channel][X_PEAK+16]);
      else tp2=(yd[i]-yd[i2])/(1.0-ynorm[f_channel][X_PEAK+i2-i]);
      ratio[npeak]=(tp1+tp2)/2.0;
      if(ratio[npeak]>9.0)
        {
  if(i-i1>PP4) k1=i-PP4; else k1=i1;
  if(i2-i>PP4) k2=i+PP4; else k2=i2;
        for(j=0;j<k2-k1+1;j++) weight[j]=1.0;
        param[1]=ratio[npeak]; param[2]=(double)ip[npeak];
        f_n_peak=1;
        chisqr=fit(x+k1,yd+k1,weight,f_dsc_jump,k2-k1+1,2,param+1,error,yfit);
        if(chisqr!=-1.0)
  {xp[npeak]=param[2]; yamp[npeak]=param[1];}
        else {xp[npeak]=(double)ip[npeak]; yamp[npeak]=ratio[npeak];}
```

```
        npeak++;
 if(chisqr!=-1.0&&param[1]<10) npeak--;
        }
      flag=0;
      if(i2<Narr-2*Nma)
        {if(i2<Narr-Nge) {il=i2+Nge/2-1; Nrange=Nge;}
else {il=(int)(i2+Narr-1)/2; Nrange=(int)(Narr-i2-1)/2;}
        }
      else {il=Narr;}
     }//if()
   }//(while)


 while(npeak>N_PEAKS)

   {tp1=10000000.0; i1=0;
    for(i=0;i<npeak;i++)
      {//if(ratio[i]<tp1){tp1=ratio[i];i1=i;}
if(yamp[i]<tp1) {tp1=yamp[i];i1=i;}
      }
    for(i=i1;i<npeak-1;i++)
      {ip[i]=ip[i+1];dleft[i]=dleft[i+1];dright[i]=dright[i+1];
      ratio[i]=ratio[i+1];
      xp[i]=xp[i+1]; yamp[i]=yamp[i+1];
      }
    npeak--;
   }
 for(i=0;i<Narr;i++) weight[i]=0.0;
 for(i=0;i<npeak;i++)
   {for(i1=ip[i]-dleft[i];i1<=ip[i]+dright[i];i1++)
     {if(i1>=0&&i1<127&&yd[i1]>0.0) weight[i1]=1.0;}
   }

 param[0]=0.0;
 for(i=0;i<npeak;i++)
   {//param[2*i+1]=ratio[i]; param[2*i+2]=(double)ip[i];
     param[2*i+1]=yamp[i]; param[2*i+2]=xp[i];
   }

 /* add 3/24/05 */
 ct_refit=0;
```

```
refit:
/* add end */
f_n_peak=npeak; f_n_hpeak=0;

/* add 1/19/05 */
chisqr=-1;
/* add end */

if(f_n_peak>0)
  {
  chisqr = fit(x, yd, weight, f_dsc_offset, Narr, npeak*2+1, param, error, yfit);  //original
  if(chisqr!=-1)
    {for(i=0;i<Narr;i++) yf[i]=yfit[i];
    /* add 3/24/05 */
    goto refine;
    /* add end */
    }
  /* modify 3/24/05 */
  // else {return -1;}//printf("Bad fitting!\n");}
  else
    {
      ct_refit++;
      if(ct_refit>15) return -1;
    }
  /* modify end */
  }
else return 0; //printf("No peaks!\n");


/* add 3/24/05 */
flag=0;
for(i=0;i<npeak;i++)
  {
    if(param[2*i+1]<YW_pth2&&npeak>0)
      {for(j=i;j<npeak-1;j++)
{param[2*j+1]=param[2*(j+1)+1]; param[2*j+2]=param[2*(j+1)+2];
/* add 1/30/05 */
yamp[j]=yamp[j+1]; xp[j]=xp[j+1];
        ip[j]=ip[j+1];dleft[j]=dleft[j+1];dright[j]=dright[j+1];
        ratio[j]=ratio[j+1];
        /* add end */
```

```
        }
      npeak--;
      //goto refit;
      /* change 1/30/05 */
      //goto refit;
      //goto refit_o;
      flag=1;
      /* change end */
      }
  }//for(i)

 if(flag&&npeak>0)
  {
   for(i=0;i<Narr;i++) weight[i]=0.0;
   for(i=0;i<npeak;i++)
   {for(i1=ip[i]-dleft[i];i1<=ip[i]+dright[i];i1++)
     {if(i1>=0&&i1<127&&yd[i1]>0.0) weight[i1]=1.0;}
   }
   goto refit;
  }
 /* add end */

 refine:
 /* add 3/25/05 */
 if(npeak>=6) return 6;
 /* add end */
 /* add 1/24/05 */
 if(chisqr!=-1) chisqr00=chisqr;
 /* add end */

 /* add 1/19/05 */
 chisqr2=-1000.0;
 if(chisqr!=-1)
  {nv=-10000.0; k2=0;
   for(i=0;i<f_n_peak;i++)
     { if(param[2*i+2]>b0_lim1&&param[2*i+2]<b0_lim2)
       {if(param[2*i+1]>nv) {nv=param[2*i+1];itp=i;}
if(param[2*i+1]>thresh2) k2++;
       }
     }
   /* add 1/20/05 */
```

```
    if(k2==0)
        {return npeak;}
    /* add end */
    chisqr2=0.0;
    i=ip[itp];
    /* change 1/24/05 */
    //i1=maxp(i-8,i-dleft[itp]);
    i1=minp(i-PP8,i-dleft[itp]);
    if(i1<b0_lim1-1.0) i1=b0_lim1-1; // add 1/24/05
    /* change end */
    i1=maxp(i1,0);
    i2=minp(i+8,i+dright[itp]); i2=minp(i2,125);
    /* add 1/24/05 */
    if(i2<b0_lim2-0.0) i2=b0_lim2-0.0;
    /* add end */
    //for(j=i-dleft[itp];j<=i+dright[itp];j++)
    k1=0;
    for(j=i1;j<=i2;j++)
      {if(yd[j]>0.0)
{chisqr2+=(yfit[j]-yd[j])*(yfit[j]-yd[j])/yd[j]/yd[j];k1++;}
      }
    //chisqr2/=(dleft[itp]+dright[itp]+1.0);
    chisqr2/=k1;
    chisqr2=sqrt(chisqr2);
   }
 /* change 1/24/05 */
 //if(chisqr2>0.19&&k2<2)
 if(chisqr2>chisqr_lim&&k2<2)
 /* change end */
   {npeak2=f_n_peak;
    for(j=0;j<=2*npeak2;j++) param2[j]=param[j];
    for(j=0;j<127;j++) {yfit2[j]=yfit[j]; yd2[j]=yd[j];}
    chisqr02=chisqr2;
    k1=ip[itp]; k2=k1+PP4;
    if(k2>126) k2=126;

    f_n_peak=1;
    param[1]=yamp[itp]; param[2]=(double)k1;
    offsett=0.0;
    for(j=0;j<=k2-k1;j++) weight[j]=1.0;
    chisqr=fit(x+k1,yd+k1,weight,f_dsc_jump,k2-k1+1,2,param+1,error,yfit);
```

```
    if(chisqr!=-1)
      {param_tp[0]=param[1]; param_tp[1]=param[2];
      k1=b0_lim1;
      /* change 1/24/05 */
      //k2=(int)param[2];
      k2=b0_lim2;
      /* change end */
      f_n_peak=1; f_n_hpeak=0;
      nv=-1000.0; il=k1;
      for(j=k1;j<=k2;j++)
         {yd2[j]=yd[j]-f_dsc_offset((double)j,param);
 if(yd2[j]>nv) {nv=yd2[j];il=j;}
         }
      if(il>k1+1&&il<k2-1&&nv>8.0)
        {k1=maxp(il-PP4,k1); k2=minp(il+PP4,k2);
f_n_peak=1;f_n_hpeak=0;
offsett=0.0; param[1]=nv; param[2]=(double)il;
        for(j=k1;j<=k2;j++) weight[j]=1.0;
    chisqr=fit(x+k1,yd2+k1,weight,f_dsc_jump,k2-k1+1,2,param+1,error,yfit);
         if(chisqr!=-1)
           {
     npeak=npeak2+1;
             for(j=npeak-1;j>itp;j--)
               {yamp[j]=yamp[j-1];xp[j]=xp[j-1];
       ip[j]=ip[j-1];
       dleft[j]=dleft[j-1]; dright[j]=dright[j-1];
               }
             yamp[itp+1]=param_tp[0]; xp[itp+1]=param_tp[1];
             ip[itp+1]=(int)xp[itp+1];
             dleft[itp+1]=0; dright[itp+1]=minp(dright[itp+1],PP4);

             yamp[itp]=param[1]; xp[itp]=param[2];
             ip[itp]=il;
             dleft[itp]=il-k1; dright[itp]=k2-il;
     /* fit 1/19/05 */
             for(i=0;i<Narr;i++) weight[i]=0.0;
             for(i=0;i<npeak;i++)
                {for(i1=ip[i]-dleft[i];i1<=ip[i]+dright[i];i1++)
                 {if(i1>=0&&i1<127&&yd[i1]>0.0) weight[i1]=1.0;}
                }
```

```
               param[0]=0.0;
               for(i=0;i<npeak;i++)
               {//param[2*i+1]=ratio[i]; param[2*i+2]=(double)ip[i];
                param[2*i+1]=yamp[i]; param[2*i+2]=xp[i];
               }

               f_n_peak=npeak; f_n_hpeak=0; offsett=0.0;
               chisqr = fit(x, yd, weight, f_dsc_offset, Narr, npeak*2+1, param, error, yfit);  //original
             /* fit end */
       // cal chisqr2
                chisqr2=-1000.0;
if(chisqr!=-1)
  {nv=-10000.0; k2=0;
  for(i=0;i<f_n_peak;i++)
    { if(param[2*i+2]>b0_lim1&&param[2*i+2]<b0_lim2)
      {if(param[2*i+1]>nv) {nv=param[2*i+1];itp=i;}
      if(param[2*i+1]>thresh2) k2++;
      }
    }
  chisqr2=0.0;
  i=ip[itp];
  /* change 1/24/05 afternoon */
  //i1=maxp(i-8,i-dleft[itp]);
                 i1=minp(i-PP8,i-dleft[itp]);
                 /* change end */
                 i1=maxp(i1,0);
  i2=minp(i+8,i+dright[itp]); i2=minp(i2,125);
  //for(j=i-dleft[itp];j<=i+dright[itp];j++)
  k1=0;
  for(j=i1;j<=i2;j++)
    {if(yd[j]>0.0)
      {chisqr2+=(yfit[j]-yd[j])*(yfit[j]-yd[j])/yd[j]/yd[j];k1++;}
    }
  //chisqr2/=(dleft[itp]+dright[itp]+1.0);
  chisqr2/=k1;
  chisqr2=sqrt(chisqr2);
}
       // cal end
if(chisqr2<0) chisqr2=-chisqr2;
/* change 1/24/05 */
```

```
                //if(chisqr2<chisqr02)
/* modify 3/24/05 */
// if(chisqr2<chisqr02&&chisqr<3.0*chisqr00)
  if(chisqr!=-1&&chisqr2<chisqr02&&chisqr<3.0*chisqr00)
    /* modify end */
  /* change end */
  {
                    for(j=0;j<Narr;j++)
                      //yf[j]=Y_BASE-yfit[j];
                      yf[j]=yfit[j];
                    return npeak;
                  }
                else {chisqr2=chisqr02;
       npeak=npeak2;
      /* add 1/24/05  afternoon */
                      param[0]=param2[0];
                      for(j=0;j<npeak;j++)
      {param[2*j+1]=param2[2*j+1]; param[2*j+2]=param2[2*j+2];}
      /* add end */
                      return npeak;
                      }
           }
        }
      }
   }
 /* add end */

 return npeak;
}

int dsc_fit_csi(double yr[],double yf[],int nl,double param[])
{
  int i,j;
  double        error[N_PARAM], chisqr,
              yd[127], yfit[127], weight[127];
  static double x[127];

  int ip[Nlim],dleft[Nlim],dright[Nlim],npeak,i1,i2;
  double ratio[Nlim],tp1,tp2;
  double xp[Nlim],yamp[Nlim];
  int protect[Narr];
```

```
 int k1,k2;

 int il,p1,p2,flag=0;

 double mv,nv;
 int itp,Nrange;
 int Nstep;

 float dpeak=3.5;

 /* add 1/19/05 */
 double param2[Nlim];
 int npeak2;
 float chisqr02;
 double x2[127],yd2[127],yfit2[127];
 double param_tp[2];

 // add 1/24/05
 double chisqr00;
 /* add end */

 /* add 1/31/05 */
 int ct_refit;
 /* add end */

for(i=0;i<nl;i++)
  {yd[i]=yr[i];//(double)(Y_BASE-yr[i]);
    x[i]=(double)i;
   }
for(i=0;i<Narr;i++) protect[i]=0;
npeak=0;



il=4; flag=0; Nrange=Nge;
// for(i=1;i<126;i++)
while(il<126) // il<122
  { flag=0;
    p1=il-Nrange/2; p2=il+Nrange/2;
    if(p1<0) p1=0;
    if(p2>126) p2=126;
```

```
     mv=-1000.0;
     for(j=p1;j<=p2;j++)
       {if(yd[j]>=mv) {mv=yd[j]; i=j;}}

     if(i<il+Nrange/2&&i>il-Nrange/2&&mv>1.0&&!protect[i]) {flag=1;}
       else {il=il+Nrange/2;}
// if(yd[i]>=yd[i-1]&&yd[i]>=yd[i+1]&&2.0*yd[i]>(yd[i-1]+yd[i+1]))
    if(flag)
     {ip[npeak]=i;
     //i1=i-1;
     // while(i1>0&&yd[i1-1]<yd[i1]&&yd[i1]>0.0) i1--;
     // if(yd[i1]==0.0) i1++;
     // i2=i+1;
     // while(i2<Narr-1&&yd[i2+1]<yd[i2]&&yd[i2]>0.0) i2++;
     // if(yd[i2]==0.0) i2--;
     i1=i; itp=i1-1;

     /* add1 */
     Nstep=Nstep2;
     /* add1 end */
     while(itp<i1)
       {i1=itp; nv=yd[itp];
/* add */
        if(i-i1>1) Nstep=Nstep1;
/* add end */
       for(j=i1;j>=i1-Nstep; j--)
         {if(j<0||yd[j]==0.0) break;
 if(yd[j]<nv) {nv=yd[j];itp=j;}
         }
       }
     i2=i; itp=i2+1;

     /* add1 */
     Nstep=Nstep2;
     /* add1 end */
     while(itp>i2)
       {i2=itp; nv=yd[itp];
       /* add1 */
       if(i2-i>1) Nstep=Nstep1;
       /* add1 end */
       for(j=i2;j<=i2+Nstep;j++)
```

```
        {if(j>126||yd[j]==0.0) break;
 if(yd[j]<nv) {nv=yd[j]; itp=j;}
        }
      }
     //if((i-i1<=2&&yd[i]-yd[i1]<=dpeak)||(i2-i<=2&&yd[i]-yd[i2]<=dpeak))
     //{flag=0; il=il+Nrange/2; continue;}
     for(j=i1;j<i2;j++) protect[j]=1;
      dleft[npeak]=i-i1; dright[npeak]=i2-i;
      if(i-i1>9) tp1=(yd[i]-yd[i-9])/(1.0-ynorm[f_channel][0]);
      else tp1=(yd[i]-yd[i1])/(1.0-ynorm[f_channel][X_PEAK-i+i1]);
      if(i2-i>16) tp2=(yd[i]-yd[i+16])/(1.0-ynorm[f_channel][X_PEAK+16]);
      else tp2=(yd[i]-yd[i2])/(1.0-ynorm[f_channel][X_PEAK+i2-i]);
      ratio[npeak]=(tp1+tp2)/2.0;
      //printf("ratio=%f \n",ratio[npeak]);
      //if(ratio[npeak]>9.0)
      if(ratio[npeak]>YW_pth1)
      //if(ratio[npeak]>6.0&&(i2-i)>1&&(i-i1)>1)
        {
        if(i-i1>PP4) k1=i-PP4; else k1=i1;
        if(i2-i>PP4) k2=i+PP4; else k2=i2;
        for(j=0;j<k2-k1+1;j++) weight[j]=1.0;
        param[1]=ratio[npeak]; param[2]=(double)ip[npeak];
        f_n_peak=1;
/* change 1/30/05 night */
      //chisqr=fit(x+k1,yd+k1,weight,f_dsc_jump,k2-k1+1,2,param+1,error,yfit);
param[0]=0.0; f_n_hpeak=0;
chisqr=fit(x+k1,yd+k1,weight,f_dsc_offset,k2-k1+1,3,param,error,yfit);
        /* change end */
        if(chisqr!=-1.0)
  {xp[npeak]=param[2]; yamp[npeak]=param[1];}
        else {xp[npeak]=(double)ip[npeak]; yamp[npeak]=ratio[npeak];}
        //printf("np=%d,chi=%8.3f \n",npeak,chisqr);
         npeak++;
 if(chisqr!=-1.0&&param[1]<YW_pth1)
            npeak--;
        }
      flag=0;
      if(i2<Narr-2*Nma)
        {if(i2<Narr-Nge) {il=i2+Nge/2-1; Nrange=Nge;}
else {il=(int)(i2+Narr-1)/2; Nrange=(int)(Narr-i2-1)/2;}
        }
```

```
      else {il=Narr;}
     }//if()
   }//(while)

 /* change 1/21/05 */
 // while(npeak>N_PEAKS)
 while((f_channel!=289&&npeak>N_PEAKS)||(f_channel==289&&npeak>=N_PEAKS))
 /* change end */
   {tp1=10000000.0; i1=0;
    for(i=0;i<npeak;i++)
      {//if(ratio[i]<tp1){tp1=ratio[i];i1=i;}
if(yamp[i]<tp1) {tp1=yamp[i];i1=i;}
      }
    for(i=i1;i<npeak-1;i++)
      {ip[i]=ip[i+1];dleft[i]=dleft[i+1];dright[i]=dright[i+1];
      ratio[i]=ratio[i+1];
      xp[i]=xp[i+1]; yamp[i]=yamp[i+1];
      }
    npeak--;
   }
 for(i=0;i<Narr;i++) weight[i]=0.0;
 for(i=0;i<npeak;i++)
   {for(i1=ip[i]-dleft[i];i1<=ip[i]+dright[i];i1++)
     {if(i1>=0&&i1<127&&yd[i1]>0.0) weight[i1]=1.0;}
   }

 param[0]=0.0;
 for(i=0;i<npeak;i++)
   {//param[2*i+1]=ratio[i]; param[2*i+2]=(double)ip[i];
     param[2*i+1]=yamp[i]; param[2*i+2]=xp[i];
   }

 /* add 1/31/05 */
 ct_refit=0;
 /* add end */
 refit:
 f_n_peak=npeak; f_n_hpeak=0;

 /* add 1/19/05 */
 chisqr=-1;
 /* add end */
```

```
/* add 1/30/05 night */
param[0]=offsett;
/* add end */

if(f_n_peak>0)
  {
  chisqr = fit(x, yd, weight, f_dsc_offset, Narr, npeak*2+1, param, error, yfit);  //original
  if(chisqr!=-1)
    {for(i=0;i<Narr;i++) yf[i]=yfit[i]; //Y_BASE-yfit[i];
    }
  else
    {
      /* change 1/31/05 */
      //return -1; //{printf("Bad fitting!\n");}
      ct_refit++;
      if(ct_refit>15) return -1;
     /* change end */
    }
  }
else return 0; //printf("No peaks!\n");

/* add 1/24/05 */
if(chisqr!=-1) chisqr00=chisqr;
/* add end */

/* add 1/19/05 */
//chisqr2=-1000.0;

/* add 1/30/05 */
flag=0;
/* add end */

/* add 1/26/05 */
for(i=0;i<npeak;i++)
  {
    if(param[2*i+1]<YW_pth2&&npeak>0)
      {for(j=i;j<npeak-1;j++)
{param[2*j+1]=param[2*(j+1)+1]; param[2*j+2]=param[2*(j+1)+2];
/* add 1/30/05 */
yamp[j]=yamp[j+1]; xp[j]=xp[j+1];
```

```
        ip[j]=ip[j+1];dleft[j]=dleft[j+1];dright[j]=dright[j+1];
        ratio[j]=ratio[j+1];
        /* add end */
        }
      npeak--;
      //goto refit;
      /* change 1/30/05 */
      //goto refit;
      //goto refit_o;
      flag=1;
      /* change end */
      }
  }//for(i)
 /* add end */

 /* add 1/30/05 */
 if(flag&&npeak>0)
   {
    for(i=0;i<Narr;i++) weight[i]=0.0;
    for(i=0;i<npeak;i++)
    {for(i1=ip[i]-dleft[i];i1<=ip[i]+dright[i];i1++)
      {if(i1>=0&&i1<127&&yd[i1]>0.0) weight[i1]=1.0;}
    }
    goto refit;
   }
 /* add end */

 return npeak;
}
```

Appendix C

Properties of CsI scintillators

Table C.1: Optical properties of the pure CsI scintillators used for the PIBETA calorimeter (Manufacturers: Bicron Corporation and Kharkov Institute for Single Crystals).

Quantity	Average Value
Density	$4.53\,\mathrm{g/cm^3}$
Radiation Length	$1.85\,\mathrm{cm}$
Refractive Index at 500 nm	1.80
Refractive Index at 315 nm	1.95
Nuclear Interaction Length	$167\,\mathrm{g/cm^2}$
$dE/dx\|_{\mathrm{min}}$	$1.243\,\mathrm{MeV/g/cm^2}$
Photonuclear Absorption (π^0 photons)	$0.675\,\%$
Light Attenuation Length	103 cm
Fast Component Wavelength	305 nm
Fast Component Decay Time	7 ns
Slow Component Wavelength	450 nm
Slow Component Decay Time	35 ns
Lower Wavelength Cutoff	260 nm
Fast-to-Total Light Output Ratio	0.76
Light Output	66.2 Photoel./MeV
Light Output Nonuniformity	$+0.28\,\%/\mathrm{cm}$
Light Output Temperature Coefficient	$-1.56\,\%/^\circ\mathrm{C}$
Time Resolution (wrt PV tag)	0.68 ns
Stability	Slightly Hygroscopic

Bibliography

[1] Weidong Li,A Precise Measurement of $\pi^+ \to \pi^0 e^+ \nu$ Branching Ratio ,Doctoral Dissertation.

[2] Design, Commissioning and Performance of the PIBETA Detector at PSI,2004

[3] (Particle Data Group), Phys. Rev. D 66, 2002.

[4] V.Cirigliano, The Pionic beta decay in chiral perturbation theory(2002).

[5] I.S. Towner et al, arXiv:nucl-th/9809087 29 Sep 1998.

[6] B.G. Erozolimsky et al, Sov. J. Nucl. Phys. 53:260-262,1991.

[7] M.B. Kreuz et al, arXiv:hep-ph/0312124, Sep 2002.

[8] P. Depommier et al. Nucl. Phys. B4 (1968) 189.

[9] W.K. McFarlane et al, Phys. Rev. D 43, 547(1985).

[10] G. Czapek et al., Pys. Rev. Lett. 70(1993) 17.

[11] B.B. Binon et al., Nucl. Instr. and Meth.332(1993) 419.

[12] F. Halzen et al, Quarks and Leptons, J. Wiley & Sons, 1984.

[13] T.D. Lee and C.N. Yang, Phys. Rev. 104, 254(1956).

[14] C. S. Wu et al., Phys. Rev. 105, 1413 (1957).

[15] R.P. Byron, Particl Phys. at the New Millennium, Springer-Verlag NY, 1996

[16] S.S. Gershtein, I.B. Zeldovich, Soviet Phys. Jetp 2,576(1956); R.P. Feynman, M. Gell-Mann, Phys. Rev. 109, 193(1958).

[17] A. Sirlin. Phys. Rev. 164, 5(1967).

[18] Elementary Particle Physics, Addison-Wesley Pub. Company, Inc (1964).

[19] A. Sirlin. Rev. Mod. Phys. 50, 573 (1978).

[20] A. Sirlin, Letter to D. Pocanic, 19 Mar. 1989.

[21] Maple 6, Waterloo Maple Inc.

[22] W.J. Marciano & A. Sirlin, Phys. Rev. Lett. 56(1986) 1(22:25).

[23] W. Jaus, Phys. Rev. D 63, 053009(2001).

[24] K.L. Brown, D.C. Carey, Ch. Iselin, and F. Rothacker, Transport: A Computer Program for Designing Charged Particle Beam Transpost Systems, CERN Yellow Reposts 73-16/80-04 (CERN, Geneva, 1973/1980).

[25] K. L. Brown, Ch. Iselin, and D. C. Carey, Decay Turtle, CERN Yellow Repost 74-2 (CERN, Geneva, 1974).

[26] U. Rohrer, Computer Control for Secondary Beam Lines at PSI, PSI Annual Report, Annex I (1988) 7.

[27] E. Frlez et al. Design, Commissioning and Performance of the PIBETA Detector at PSI, hep-ex/0312017, Submitted to Nucl. Instrum. & Meth.

[28] H. Kobayashi, A. Konaka, K. Miyake, T. T. Nakamura, T. Nomura, N. Sasao, T. Yamashita, S. Sakuragi, and S. Hashimoto, Kyoto University Preprint KUNS-900 (1987).

[29] S. Kubota, H. Murakami, J. Z. Ruan, N. Iwasa, S. Sakuragi, and S. Hashimoto, Nucl. Inst. and. Meth. A 273 (1988) 645.

[30] S. Kubota, S. Sakuragi, S. Hashimoto and J. Z. Ruan, Nucl. Inst. and Meth. A 268 (1988) 275.

[31] E. Frlez et. al. Nucl. Instr. and Meth. A 459(2001).

[32] V. V. Karpukhin et. al. , Nucl. Instrum. Methods A 418 (1998) 306.

[33] V. V. Karpukhin et. al. , Instrum. Exp. Tech. 42 (1999) 335.

[34] Bicron Corp. Catalog (Bicron Corp, Newbury, 1989).

[35] E. Frlez, et. al. Nucl. Inst. and Meth. A 440 (2000) 57.

[36] K. A. Assamagan, PhD Thesis (University of Virginia, Charlottesville, 1995).

[37] R. Brun et al., GEANT 3.21 DD/EE/94-1 (CERN, Geneva, 1994).

[38] F. James et al., MINUIT 94.1 D506 (CERN, Geneva, 1994).

[39] The Pion Beta Decay Experiment and A Remeasurement of the Panofsky Ratio. T. Flügel. Doctoral Dissertation.

[40] R. Gordon. IEEE Transactions on Nuc. Sci. Vol.NS-21, June 1974.

[41] W. J. Marciano, Phys. Rev. Lett. 71, 3629 (1993).

[42] E. Frlez et. al. hep-ex/0312017

[43] A. Sher et al.(BNL 865 Collaboration), Physics Rev. Lett. 91, 261802(2003).

[44] T. Alexopoulos et al. (KTeV Collaboration), Physics Rev. Lett. 93, 181802(2004).

[45] A. Lai et al.(CERN NA48 Collaboration), Phys. Lett. B 602, 41(2004).

[46] A. Czarnecki, W.J. Marciano, and A. Sirlin, Phys. Rev. D 70,093006(2004).

[47] Eur. Phys. J. C 27, 255 (2003).