

DESY 76/64
December 1976



On-line Data Acquisition and Reduction at the Magnetic Detector PLUTO

by

G. Franke and R. Schmitz

Abstract:

We present a method for reducing cosmic ray background in a set of cylindrical proportional wire chambers. Data taking and reduction is done by a PDP 11/45 computer.

The method is applied in e^+e^- storage ring experiments with the super-conductive solenoid magnet PLUTO. A reduction of about 50% of the incoming events is achieved.

To be sure that your preprints are promptly included in the
HIGH ENERGY PHYSICS INDEX ,
send them to the following address (if possible by air mail) :

DESY
Bibliothek
2 Hamburg 52
Notkestieg 1
Germany

On-line Data Acquisition and Reduction at the Magnetic Detector PLUTO

G. Franke, R. Schmitz

Deutsches Elektronen-Synchrotron DESY

Abstract:

We present a method for reducing cosmic ray background in a set of cylindrical proportional wire chambers. Data taking and reduction is done by a PDP 11/45 computer. The method is applied in e^+e^- storage ring experiments with the superconductive solenoid magnet PLUTO. A reduction of about 50 % of the incoming events is achieved.

<u>Contents:</u>	<u>page</u>
1. Introduction	2
2. Hardware	2
3. Data-Acquisition Software	4
3.1 Supervisory Program	4
3.2 Organisation of Data Acquisition	4
3.3 Buffer-Organisation	5
3.3.1 Variable Event Length	6
3.3.2 Wrap-around Buffer Technic and Buffer Size	6
3.3.3 Storing, Retrieval, Deletion of Event Segments	7
4. Recognition and Rejection of Cosmic Ray Events	8
4.1 Cosmic Ray Geometry	8
4.2 Recognition and Parametrisation of the Track	10
4.3 Rejection of Cosmic Ray Events	11
4.4 Cut Parameters	11
5. Features of the Recognition Program	12
5.1 Loop Control and Array Indices	12
5.2 Fixed Point Storage	12
5.3 Analyzing Time	13
References	14

1. Introduction

Large solid angles sensitive to the detection of elementary processes - and background - in storage rings force the experimentalist to think about methods of fast data collection and on-line reduction. Minimizing deadtime of both the apparatus and the on-line data collection system should be an important consideration in planning and realizing the programs.

PLUTO is a magnetic detector with the above outlined characteristics ¹⁾. The inner volume of the superconductive coil is packed with cylindrical proportional wire chambers (Fig. 1.1), triggered by a logic which is sensitive to certain well defined topological and geometrical conditions to be met by the tracks seen in the detector for any given event. 85 % of the full solid angle is sensitive to this trigger. Typically 20 events/sec, each consisting of 180 data words on the average is the present data flux from the detector. Roughly 80 % of these events are cosmic rays, most of the rest are due to interactions with the gas, and only a few percent are good events.

Two fast data links provide for collecting data into the PDP 11/45 and sending non-rejected events to an IBM 370/168 computer for storage, checks and further analysis.

2. Hardware

Data-acquisition is done with a PDP 11/45 computer with 24 K words (1 word = 16 bits) of core and 4 K words of bipolar memory. There are two kinds of data sources associated with an event trigger: wire chambers and trigger logic.

Readout of these data is initiated by program which, for its part, is triggered by an interrupt from the trigger logic.

The wire chamber data are stored in flip-flops. Each wire corresponds to one single flip-flop, which are grouped together in about 20 crates. A scan is started for set flip-flops. Each cluster (group of logically

adjacent set flip-flops) found during this scan causes one 16 bit-word to be stored in a 64 word temporary storage. These words contain address and length of the cluster relative to the crate. This is done without program control. As soon as the first crate is scanned the readout electronics produces an interrupt in the PDP. Then the data are transmitted from the crate buffers into the memory of the computer by a non-processor-request transfer, crate by crate in ascending order, and headed by a special word for each crate.

The end of the transfer is indicated by an interrupt.

Data from the trigger logic are gathered in a CAMAC-crate during the analysis process of the event in the trigger logic. The crate is supervised by a special controller²⁾ transferring its data to the computer through a data channel. The program only has to start the readout sequence and to set the channel registers appropriate. End of operation is signalled by interrupt.

The average deadtime produced by the readout process is about 1 ms/event. The contributing sources are

analysis of the event in the trigger logic	40 usec
readout of CAMAC crate for the trigger logic data into memory	400 usec
transfer of contents of the crate buffers of wire chamber data into memory	300 usec
data acquisition software	<u>300 usec</u>
	1040 usec

The time needed to fill the crate buffers is excluded because this happens in parallel to the readout of the CAMAC crate. So this contributes 2.5 % to the total deadtime of 3 % in a normal run.

The transfer of non-rejected events to the central computer (IBM 370/168) is done via a fast data-link with channels on both sides³⁾, providing for a transfer rate of about 40 Kwords per second. An important property of this link is the possibility of transmitting blocks of not necessarily adjacent storage locations in the PDP into a contiguous input buffer in the IBM in a single transfer (scatter transfer). With this feature, there is no need to move the events in the PDP into a special output buffer or

to do more than one transfer to fill the input buffer in the IBM to an optimum.

3. Data Acquisition Software

3.1 Supervisory Program

The program for the PLUTO-experiment runs under control of a multi-tasking supervisor⁴, which was designed and built mainly for that purpose and utilizes the features of the PDP 11/45, especially the 2 register sets and the 3 operating modes. The ready tasks get access to the CPU according to a task-priority between 0 and 7. This is independent of the processor priority, which is normally set to 0 in tasks, i.e. tasks are interruptable. They use register set 1 and run in user mode (see ref. 5). Interrupt routines run with processor priority 7, i.e. they are not interruptable, use register set 0 and kernel mode. This is the reason why no register saving and restoring is to be done in interrupt routines.

3.2 Organisation of Data Acquisition

Data acquisition, reduction and transfer is done in three tasks: data-task, analysis-task, IBM-task, and interrupt-routines associated with the trigger interrupt and the interrupts from the channels to the experiment and to the IBM. The size of the programs which organize the data acquisition and reduction (not including the data buffer and the reduction program itself) is 800 words. The IBM transfer organisation, which is used for all transfers to the IBM, not only event data, takes 850 words of core.

All the work necessary to read the data from the wire chambers and the trigger logic into memory is done in interrupt routines. However, in case the buffer is exhausted, the data task is attached, which then waits until the buffer is emptied. This happens very seldom, and is caused by an unusually high trigger rate or non-genuine long events due to showers or by delayed response of the IBM. Normally the interrupt routines handle the trigger interrupt; thus the deadline due to readout is not lengthened by system actions.

If an event is read, the analysis task is triggered (attached) to process the event. This task runs at task priority level 6 and can therefore only be pushed away by the timer task or keyboard tasks. The analysis runs asynchronously to the readout of events. Although it is triggered with each event, it may happen that the task has not yet finished the event-processing from the previous trigger. If all events are analyzed, the task returns. The place of an event which was rejected is released and the event will not be considered thereafter.

For every trigger interrupt, the extent to which the buffer is filled is computed. If a threshold of 3100 words is reached, an IBM transfer is booked in the request list of the IBM-task. Any earlier requests will be completed, and then this data transfer will be done. During this transfer the interrupt routine for word-count overflow of the channel to the IBM does the job of transmitting only non-rejected events by means of the scatter transfer. The smallest unit in this transfer is an event segment (see 3.3.1). The channel to the IBM is set in such a way that an interrupt occurs only in the PDP when an event segment is transferred. Upon this interrupt the space occupied by this segment is freed and the next segment is transmitted until there are no more analyzed events in the buffer or the buffer in the IBM would overflow with the next segment. On completion of the whole transfer the data task receives a message and, if waiting, continues in data taking.

3.3 Buffer Organisation

When designing the programs which manage the data buffer organisation, readout and transfer the following points were kept in mind:

- events as read from the detector have variable length with a maximum of about 10800 words (those events are generated by test pulses which set each second flip-flop)
- the total buffer-size should be as small as possible
- moving of events in storage should be avoided for lack of a fast move instruction for data blocks
- there must be the possibility of processing each event in order to decide whether it shall be transferred to the IBM or not
- the readout-, buffer- and transfer organisation should not increase deadline.

3.3.1 Variable event length

A requirement for the readout organisation was that it should be able to manage any event length up to the maximum of about 10800 words as generated by test pulses. Such events are necessary to check out the readout electronics for proper operation. Obviously it is very uneconomical to provide a buffer of this size while the event length is of 180 words on the average (fig. 3.1).

Another problem arises from the fact that the actual length of an event is not known before the readout sequence starts. Therefore one cannot be sure that the event fits totally in the free portion of the buffer when starting the readout. For these reasons provisions are made to split one event into more than one event segment (fig. 3.3). These event segments are the smallest units the data acquisition-, analyzing- and transfer routines deal with. But 99.9 % of the events are recorded in only one event segment.

3.3.2 Wrap-around Buffer Technique and Buffer Size

The minimal buffer size is determined from an appropriate record length on the data set at the IBM which gathers the non rejected event segments before they are dumped to tape. This is a direct access data set with a fixed record length of 3220 words ('define file' in FORTRAN), i.e. 2 records per track on an IBM 3330 disk. Because of the delay between the transfer request to the IBM and the transmission of the first data word (of about 8 ms on the average), the data flux from the experiment of about 4000 words per second and the fluctuation in both analysis time and time interval between events, a larger buffer is needed to avoid deadline. Finally a buffer of 4000 words is used, and transfer to the IBM is initialized each time the occupied space exceeds 3100 words. A single buffer is used which is treated as a wrap-around one, i.e. there exists no fixed beginning or end of the buffer but the physical end is logically tied to the beginning.

3.3 Storing, Retrieval, Deletion of Event Segments

The event segments in the buffer are logically connected by pointers, thus forming 2 chains, one input chain and one output chain. Each event segment can belong to only one chain. The input chain contains the event segments which are read in but not processed by the analysis task so far: into the output chain however, are put only the non-rejected event segments. The IBM task only knows about the event segments in the output chain.

When storing event segments, only the wrap around free space between the end of the current and the beginning of the earliest one in the buffer is utilized. The space between non-rejected event segments, caused by event segments which are not transferred from the input- to the output chain (rejected event segments is declared to be occupied until these segments will have been transmitted. But due to the speed of the analysis task in processing an event segment compared to the event rate, the latest event segment normally (in 92 % of the cases) will be analyzed before the next event is read and, if it was rejected, it is overwritten by the next one (fig. 3.5). Therefore the records on the IBM are very well utilized (fig. 3.2).

If the space to the end of the buffer is less than 270 words, the next segment is started at the beginning of the buffer. This means that splitting only occurs if an event is longer than 3100 words (maximum length of a segment due to the size of the input buffer in the IBM-program), or an event is longer than 270 words and reaches the physical end of the buffer before all words are read from the detector. In these cases the segment is put into the input chain and a new one is opened to store the missing event data.

Fig. 3.4 shows some instantaneous pictures of the buffer to illustrate its organisation.

4. Recognition and Rejection of Cosmic Ray Events

Cosmic ray events are recognized by reconstructing their tracks by means of the coordinates measured in the PLUTO cylinder chambers and checking whether they originate from the interaction point of the storage ring or not. The procedure does not aim a total rejection of cosmic rays, which in any case is not possible in PLUTO, but tries to get rid of a fairly large fraction of non-genuine events in a short time.

4.1 Cosmic Ray Geometry

Neglecting multiple Coulomb scattering, energy losses and magnetic field inhomogeneities, a cosmic ray track may be described as a helix. While the r - ϕ -projection is represented by a circle (fig. 4.1), in the r - z -projection a hyperbola is quite a good approximation to the track (fig. 4.2). Both ϕ - and z -coordinates are directly measured by most of the inner chambers. In either projection, a symmetry axis is obvious. In r - ϕ this axis is defined by the ϕ -angle of the track point with the minimum distance (r_{min}) to the origin. All pairs of measured ϕ -coordinates along the track with the same radius have (within resolution) the same mean value:

$$\overline{\phi}_i = (\phi_1(r_i) + \phi_2(r_i))/2 = \text{const} \quad (4.1)$$

(i = chamber index)

A similar condition can be defined in the r - z -projection for all pairs of z -coordinates with the same radius:

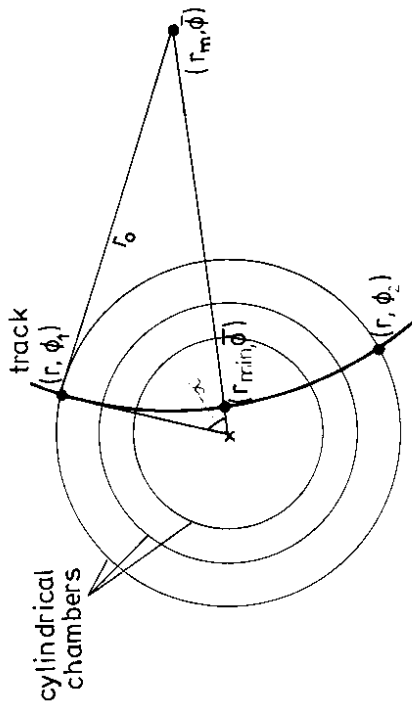
$$\overline{z}_i = (z_1(r_i) + z_2(r_i))/2 = \text{const} \quad (4.2)$$

In this case the symmetry axis is given by $z_{min} = z(r_{min}) = \text{const}$.

Equation (4.1) is a necessary condition for a cosmic ray track. In addition it has to be proved that the coordinate pairs fulfilling condition (4.1) lie on a circle

$$r^2 + r_m^2 - 2r r_m \cos \alpha = r_0^2 \quad (4.3)$$

with $\alpha = (\phi_1(r) - \phi_2(r))/2$



Equation (4.3) is 'linearized' assuming $x = r^2$ and $y = r \cos \alpha$ being the variables:

$$y = ax + b \quad (4.4)$$

with the parameters

$$a = \frac{1}{2 \cdot r_m}, \quad b = \frac{r_m^2 - r_0^2}{2 r_m} \approx r_m - r_0 = r_{min}$$

The parameters are calculated guessing that the first and the last point $(x_f, y_f), (x_l, y_l)$ measured on the track satisfy (4.4):

$$a = \frac{r_f \cos \alpha_f - r_l \cos \alpha_l}{r_f^2 - r_l^2} = \kappa \quad (4.5)$$

$$b = \frac{r_f^2 r_l \cos \alpha_l - r_l^2 r_f \cos \alpha_f}{r_f^2 - r_l^2} = r_{\min}$$

4.2 Recognition and Parametrisation of the Track

Recognition of the track both in $r-\phi$ - and in $r-z$ -projections starts with searching for a chamber with only two clusters. If none is found, recognition stops and the event is kept. The first cluster pair recognized (primary pair) is utilized to define the above mentioned symmetry axis $\bar{\phi}_f$ or \bar{z}_f for the corresponding projections. In the next step pairs of clusters are looked for in the other chambers meeting conditions (4.1) or (4.2) in the form

$$|r_i \bar{\phi}_i - r_f \bar{\phi}_f| < \epsilon_1 \quad (4.6)$$

or

$$|\bar{z}_i - \bar{z}_f| < \epsilon_2 \quad (4.7)$$

for the respective projections. These conditions are called road conditions. The number of clusters in the detectors not defining the primary pair may exceed two but is limited to a small number (about 5). A track candidate is assumed to be recognized if a certain number (about 5) of cluster pairs lie on the road.

In the $r-z$ -projection no further checks will be done, because the two branches of the track are almost straight (fig. 4.2). The track is characterized by one parameter

$$z_{\min} = \frac{z_f r_l - z_l r_f}{r_l - r_f} \quad (4.8)$$

(f, l = indices of first and last point).

In the $r-\phi$ -projection the cluster pairs undergo one more road check, testing whether the points lie on a circle:

$$|r_i \cos \alpha_i - r_l \cos \alpha_l| < \epsilon_1 \quad (4.9)$$

with the track parameters κ and r_{\min} as defined in (4.5).

4.3 Rejection of Cosmic Ray Events

Once a track is defined in at least one of the two projections, and the parameters (4.5 or 4.8) are calculated, the decision whether an event has to be rejected or kept consists of only a cut in z_{\min} or r_{\min} . The actual values for the experiments done in 1976 were:

$$\begin{aligned} |r_{\min}| &< 30 \text{ mm} \\ |z_{\min}| &< 150 \text{ mm} \end{aligned}$$

Figures 4.3 and 4.4 represent distributions of both r_{\min} and z_{\min} showing that the applied cuts are quite safe.

4.4 Cut Parameters

The efficiency of the recognition program depends upon preselection of some parameters, two of which were already mentioned, namely the cuts for r_{\min} and z_{\min} . There are some more important parameters, one of them checking for a general event feature, the others being used as cuts in track recognition.

Already in the phase of decoding event data and calculating coordinates the number of clusters found in any chamber is limited to a fixed number independently in the $r-\phi$ - and the $r-z$ -projection. If this number is exceeded at least once (for only one chamber), the event is kept. Complicated events with no chance in having been triggered by a cosmic ray are excluded from an extensive analysis by this means.

The most important track recognition parameters are the road widths ϵ_1 , ϵ_2 . They were chosen as small as possible, because they serve not only as recognition parameters, but also replace the function of a χ^2 -cut. Thus noncoplanar two-prong events are saved from being rejected. For ϵ_1 a value of 5 mm was found to be reasonable, ϵ_2 was set to 30 mm.

Finally, the number of coordinate pairs on the track has to obtain or exceed a certain limit. This was set to 5 for $r - \phi$ and to 4 for $r - z$ -tracks. All these parameters can be entered via a keyboard before starting a data run. They were optimized by systematic changes, and comparing results with those of a more sophisticated pattern recognition program, running on the IBM 370/168 also used as on-line computer during data runs. By a software switch the event rejection may be suppressed, also the recognition can be switched off.

5. Features of the Recognition Program

The program is written in PDP11-MACRO-11-assembly language. The alternative of writing the program in FORTRAN was rejected due to strong disadvantages being expected in analysis time and also storage requirements. A FORTRAN program was guessed to be slower by a factor of 4 and would require roughly 4 times more core. Storage required by the assembler program is 2.7 Kwords.

5.1 Loop Control and Array Indices

Loops and arrays are extensively used throughout the program. A connection of loop control and array indexing was desirable. This was realized by taking registers for loop control being 'autoincremented' at the end of a loop and without any change of contents serving for array addressing by 'index mode', 5). Thus nesting of three loops in maximum was possible without intermediately saving contents of registers.

5.2 Fixed Point Storage

All operations of the program during a run are done by the arithmetic and logical unit. The floating point processor is much slower in corres-

ponding operations (except multiplication and division) and therefore was only used to produce a table of cosine-function before the run. This is stored in steps of one degree in the form

$$\text{SINB}14(\alpha') = \sin(\alpha) \times 2^{14}$$

which guarantees an absolute accuracy of 10^{-4} , where the right side is evaluated in floating mode and then converted to integer. Angles α' are represented in minutes of a degree, lengths in 1/10 mm. Both are still much better than resolution of the chambers.

5.3 Analyzing Time

Processing time of course is one of the most important program parameters. A program contributing a lot in deadtime to the whole data taking system would be undesirable. Fig. 5.2 shows a spectrum of the analyzing time for single events. The two peaks are due to events being totally analyzed and to those where processing stops in the phase of decoding. The averaged time for analysis of one event is evidently about 4 ms, but does not contribute significantly to deadtime. The actual event rate is known to be 20-25 events/sec which means an utilization of 8-10 % of processor time.

References

1. L. Criegee et al., Proc. 1973 Int. Conf. on Instrumentation for High Energy Physics, 707 (1973)
2. F. Huebler, A. Krolzig, R. Pforte; Ein CAMAC-Crate-System-Controller für Hochenergieexperimente; Internal Report DESY F51-73/1 (1973)
3. D. Moenkemeyer; EXP-Manual, Anleitung zur Benutzung der DESY Online-Verbindung; Internal Report DESY RI-76/o4 (1976)
4. W. Krechlok, R. Schmitz; Ein PDP 11/45 Multitasking Supervisor; to be published
E. Raubold; Multitasking auf kleinen Rechnern; DESY DV-71/1 (1971)
5. Digital Equipment Corporation; PDP 11/45 Processor-Handbook (1971)

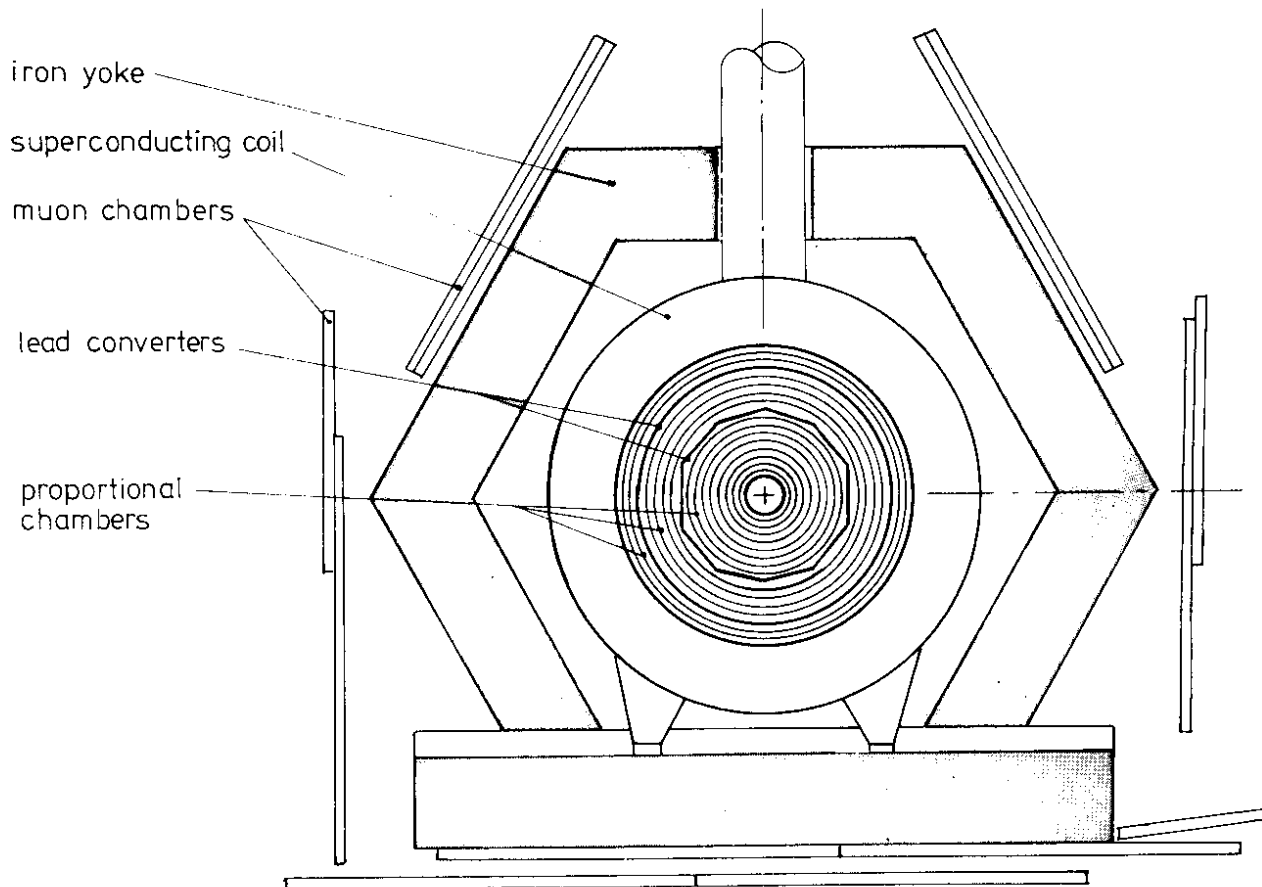
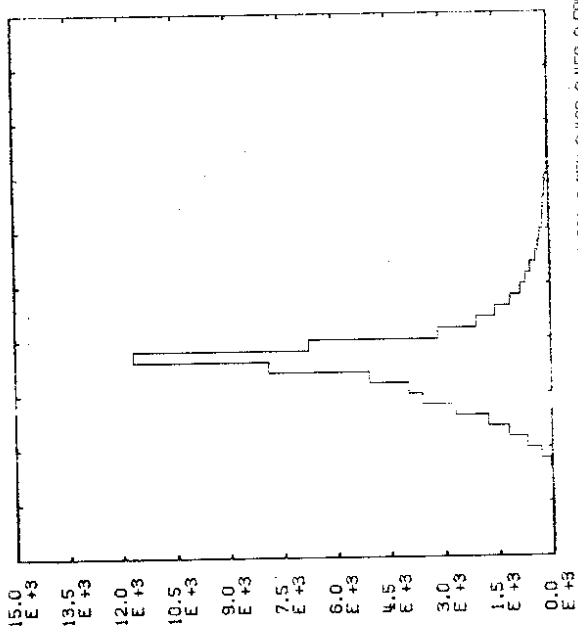
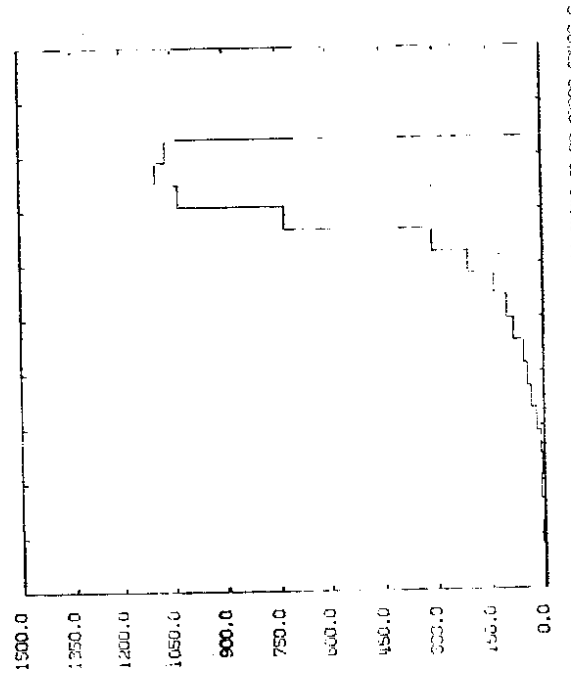


Fig. 1.1 PLUTO, Jan. 76, cross section



0.0 50.0 100.0 150.0 200.0 250.0 300.0 350.0 400.0 450.0 500.0



0.0 250.0 500.0 750.0 1000.0 1250.0 1500.0 1750.0 2000.0 2250.0 2500.0

POINTER TO NEXT SEGMENT
-1
-2 EVENT SEGMENT MARK
-3
LENGTH OF SEGMENT
TYPE OF EVENT AND SEGMENT CODE
REL. POINTER TO TRIGGER LOGIC DATA
REL. POINTER TO WIRE CHAMBER DATA
EVENT NUMBER
RESULTS FROM ANALYSIS PROGRAM

EVENT TYPE INDICATES WHETHER THE EVENT IS ANALYZED OR GENERATED BY TEST PULSES OR NOT.

SEGMENT CODE ARE 2-BITS WHICH DESCRIBE THE POSITION OF THE SEGMENT RELATIVE TO THE OTHER SEGMENTS (IF ANY) OF AN EVENT:

- 11 SEGMENT CONTAINS INDIVIDUAL EVENT
- 01 1-SEGMENT OF MULTISEGMENT EVENT
- 10 LAST SEGMENT OF MULTISEGMENT EVENT
- 00 SEGMENT IS NEITHER FIRST NOR LAST SEGMENT

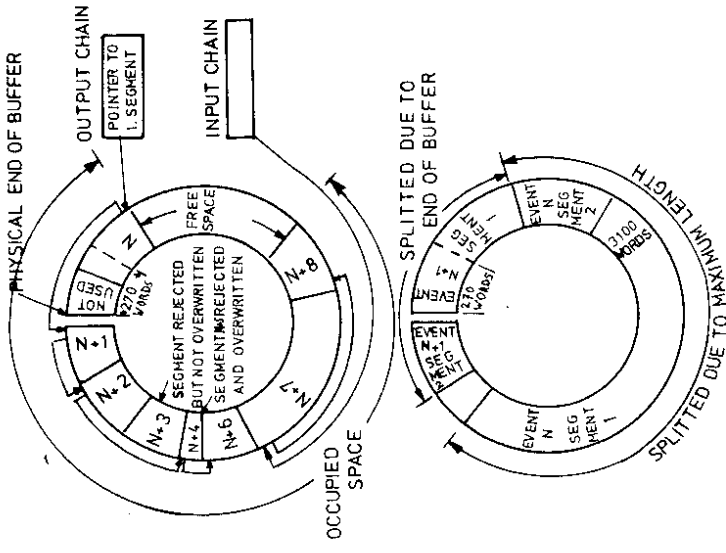
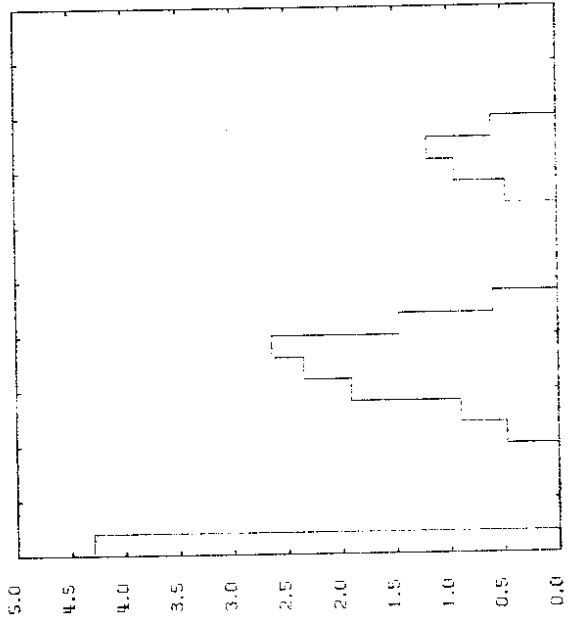


FIG. 3.4 WRAP AROUND BUFFER ORGANISATION

FIG. 3.3 FORMAT OF AN EVENT SEGMENT



0.0 50.0 100.0 150.0 200.0 250.0 300.0 350.0 400.0 450.0 500.0

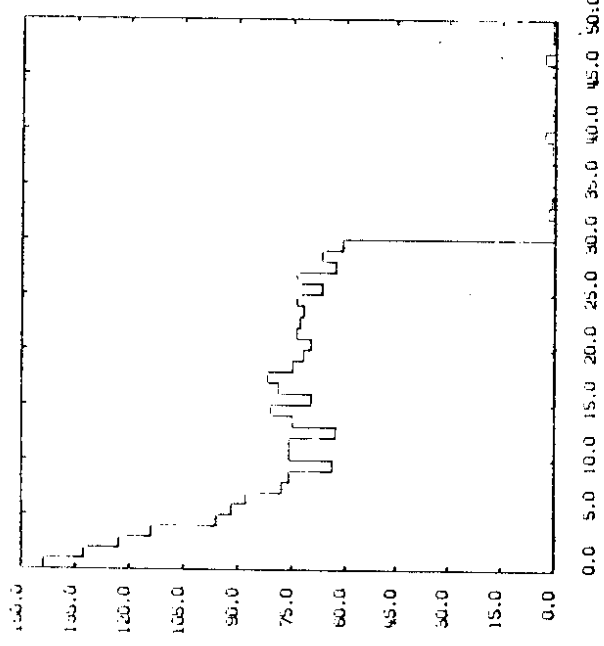


FIG. 4.3 DISTRIBUTION OF AMIN (MM)

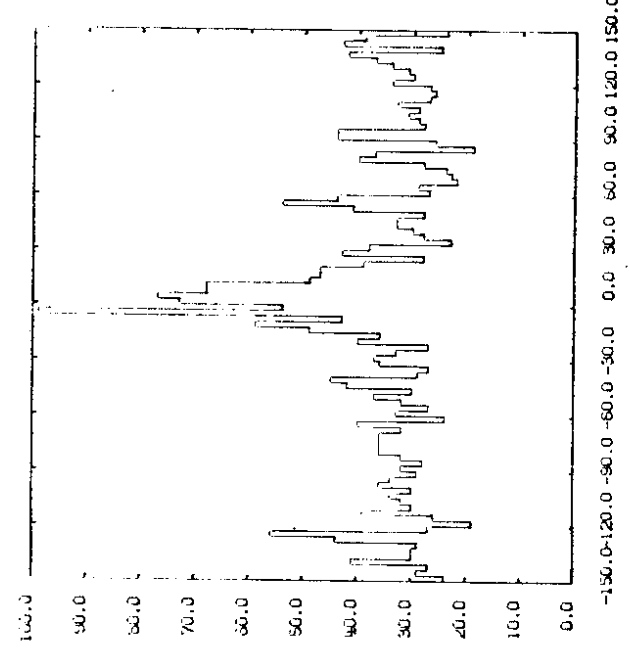


FIG. 4.4 DISTRIBUTION OF ZMIN (MM)

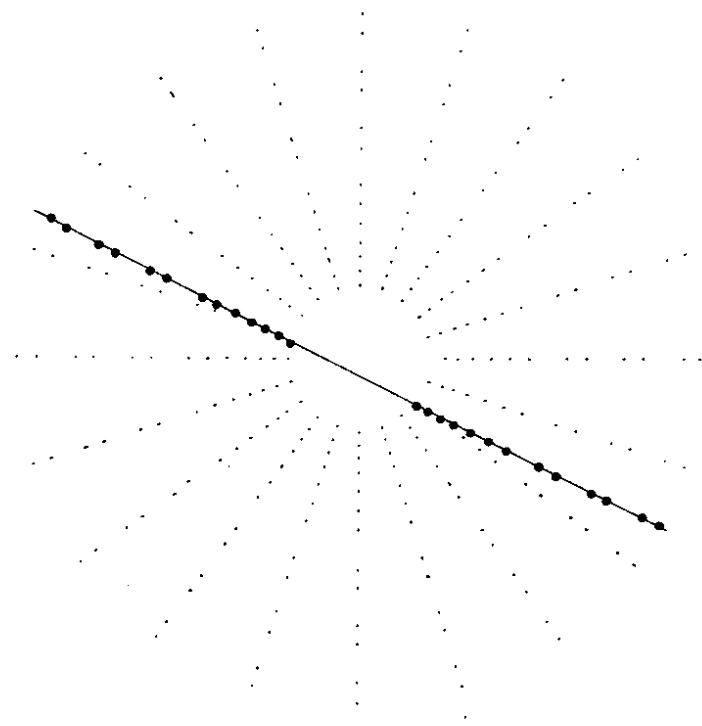


Fig. 4.1 Cosmic ray track (r - ϕ - proj.)

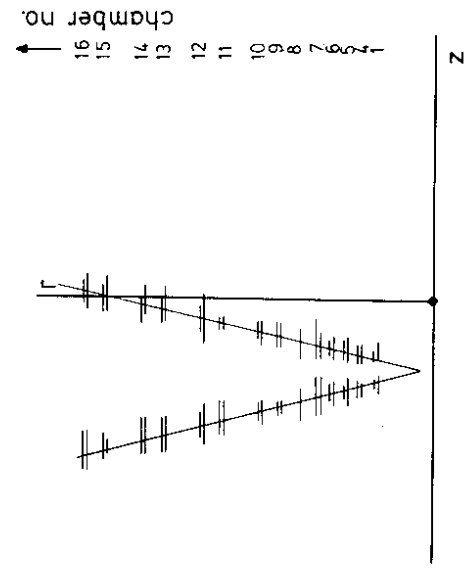


Fig. 4.2 Cosmic ray track (r - z - proj.)

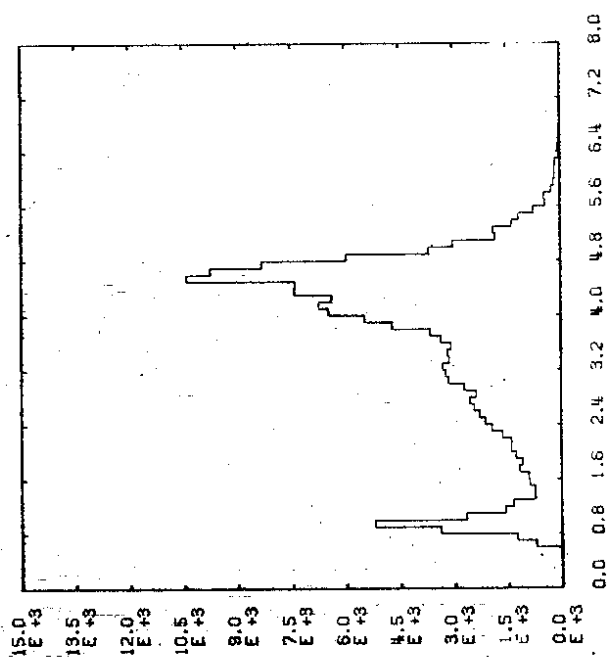


FIG. 5.1 SPECTRUM OF ANALYZING TIME (MSEC)