LUNFD6/(NFFL-7154)1998

VSAT Off-line Minibunch Tagging for the 1995 Scan

February 27, 1999

Christina Jarlskog
Lund University
Sweden

Abstract

During the 1995 Z^0 scan data-taking the VSAT detector assigned wrong minibunch number to two classes of events. The most populated one was this of high energy minibunch 3 data which were flagged as minibunch 2 events. We have presented the way to recover those events in a previous paper. This paper deals with the remaining mismatches, which represent a small - yet important for luminosity measurements - fraction of our data.

1 Introduction

This report deals with a small fraction of the VSAT data taken at the scan of 1995. It is essentially the continuation of a previous paper on misidentified events. We will therefore give below only a short description of the running conditions of that period as a detailed account has already been given [1].

1.1 The detector

The Very Small Angle Tagger (VSAT) is one of the luminosity monitors of DELPHI [2]. It is an electromagnetic sampling calorimeter consisting of four rectangular modules called F1, F2 in the forward region and B1, B2 in the backward region. The distance of the modules from the DELPHI origin is approximately 7.7 m. The luminometers are placed symmetrically around a short elliptical section of the beam pipe as shown in fig. 1(a). The process seen by the detector is Bhabha scattering at small angles (5-7 mrad), i.e. electrons and positrons emitted back-to-back carrying approximately the beam energy. We therefore look for coincidences of signals between a module in the forward region and a module in the backward region, defining two diagonals for the trigger: diagonal 1 (modules F1-B2) and diagonal 2 (modules F2-B1).

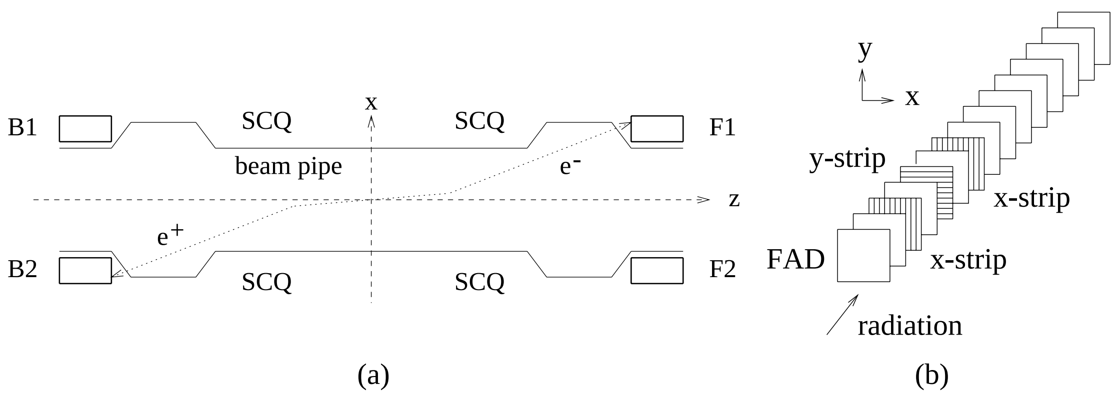

Figure 1: *Layout of (a) the position of the detector in the (x,z) plane, (b) the VSAT modules.*

The dimensions of the calorimeters are 3 cm in x, 5 cm in y and approximately 10 cm in z. Each module contains 12 tungsten absorbers interspaced with 12 silicon planes for energy measurement (FADs) (fig. 1(b)). The center of the electromagnetic shower is given by three silicon strip planes with 1 mm pitch placed close to the shower maximum at 5, 7 and 9 r.l.; the second plane is used for the y coordinate measurement and the other two planes for the x coordinate measurement.

1.2 Minibunch operation

In 1995 the VSAT electronics were modified to record events from 2 to 4 beam minibunch crossings within a fraction of a μs every 25 μs. After each beam crossing a gate generator was enabled producing gates of 50 ns during which the detector checked whether the sum of the signals from the FADs was above threshold [3]. If this was the case, the height of the signal was read out a fixed time later. The minibunch number assigned to the signal was determined by the number of the first gate during which the pulse was above threshold. As each module had an independent system for reading pulse heights, signals from different minibunches could be read in the four modules after the same beam crossing. Because of imperfections in the electronics, this occasionally resulted in the monitoring of coincidences where the electron and the positron seemed to have come from different minibunches. These mismatched events needed to have their minibunch number recovered. The aim of this report is to describe how this was achieved by using their energy distributions.

In section 2, we describe the main program which addressed this task. In section 3, we give the energy distributions of the misidentified events and in section 4 we explain how the correct minibunch number can be deduced from them.

Listings of two routines of the main program mentioned above can be found in appendices A and B. In appendix C, we give the vax disk location of the files used in the analysis. Appendix D contains a paw routine used to combine the data from all scan fills. Some of the resulting distributions can be found in appendix E. Finally, the code for minibunch tagging is presented in appendix F.

2 The data processing

We start our analysis with data which have only been compacted, i.e. there have been no physics cuts applied on them yet. We process them with the program IDMIS. The most important parts of its Bhabha routine are given in appendix A.

There are two important cuts applied here. The first cut removes particles which have hit the detector close to the edges. The reason for doing this cut is that these events cannot be reconstructed properly as the shower is not fully contained in the module. The second cut removes particles whose energy is less than (or equal to) 70 % of the beam energy. An event that satisfies both cuts is considered to be a Bhabha event provided that the two particles have the same minibunch number.

In this report, we will study events which have passed both cuts but which do not have the same minibunch number on the two modules. The way we study them is the following: for each module and minibunch number we fill four energy distributions. The first three correspond to cases where the diagonal module has given minibunch number between 1 and 4. The fourth distribution contains the rest of the events, i.e. those where the opposite module had assigned a minibunch number lower than 1 or greater than 4. We symbolically denote these minibunch numbers as being equal to 5 in the id of the histogram.

The cut in position is applied on both particles of the event, while the energy cut is applied only on the opposite module. Each of the distributions has the false Bhabha events subtracted[1]. The result is distributions with the same qualities as a genuine Bhabha distribution would have except for (a) the minibunch number agreement and (b) the 70 % energy cut.

The energy spectra thus obtained can be suitable for minibunch tagging but they do not tell us whether their events trully represent cases of Bhabha electrons. To check this statement, we need to examine the shape of two-dimensional plots showing the energy of one module plotted versus the energy of the diagonal module. We fill these histograms separately for the Bhabha events and for the false Bhabha events with only the position

[1]A part of the false Bhabha routine is given in appendix B.

cut applied. By then superimposing an acceptably shaped Bhabha distribution and a false Bhabha distribution we will be able to see which events are trully Bhabha events. To make this statement more clear, we show in fig. 2 an example of Bhabha and false Bhabha energy distributions. Both particles have minibunch number 1 and their energy is uncalibrated. A rough interpretation of the plot can be given as follows: all three black rectangles show Bhabha events, the difference between them being that the dashed areas

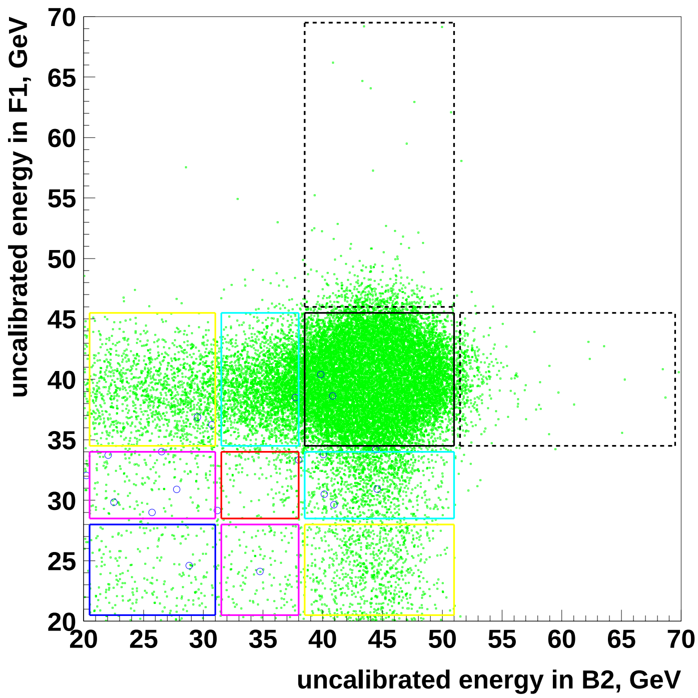

Figure 2: *Bhabha correlation plot for minibunch 1 events of a fill at peak+2. The green dots are the Bhabha events before the false Bhabha events (blue circle) are subtracted.*

represent coincidences between a Bhabha electron in one module and a pile-up[2] in the other module. The cyan rectangles denote the region of coincidences between Bhabha electrons and radiative Bhabha events. The red rectangle contains coincidences between radiative Bhabhas. All these events are kept in the analysis. The rest is rejected by the energy cut and consists of three types of coincidences: (a) background[3] particles (blue rectangle), (b) background electrons with radiative Bhabhas (magenta rectangles) and (c) background with Bhabha electrons (yellow rectangles).

Since most mismatch cases are in general very poorly populated, we processed the data in the following way: first, we ran IDMIS for each fill separately and then, we summed[4] the energy distributions over all fills falling in one of the six categories of beam energy and data taking period (we had to treat separately the data taken before and after the radiation accident of September 15 [4]). We thus grouped all our data in six histogram files, which we discuss in the following section.

[2]A pile-up is an overlap of a Bhabha signal and of a background signal.

[3]The word 'background' here means all background except the false (accidental) Bhabha events, i.e. off-momentum particles, electrons that have been scattered by a gas molecule in the beam pipe or particles that belong to a Bhabha event but have hit a part of the beam pipe, e.g. a flange.

[4]The output histogram files of IDMIS and the kumacs for adding their distributions can be found in the directory wede99$dkb500:[vsat.christina.delb.cal.scan.histo] listed in appendix C.

3 Summed energy distributions

In Table 1, we give the number of Bhabha events (without the cut on 70 % of the beam energy) contained in each of the above mentioned six histogram files. The mismatches where B2, F2 have given minibunch number 2 and F1, B1 have given minibunch number 3 have been treated in a previous report, which is why we have subtracted them in the ninth and tenth row of Table 1. The numbers in parentheses are the corresponding ratios over the total number of events. The detector has behaved essentially in the same way during the six periods[5], so in this and in the following two sections we will only present the data taken at peak-2 before the accident.

	peak-2	**peak**	**peak+2**
total number of events			
before	3362983	1380417	3471321
after	3324945	905646	3689461
number of mismatched events			
before	314649 (9.4 %)	155101 (11.2 %)	389678 (11.2 %)
after	259793 (7.8 %)	75399 (8.3 %)	345547 (9.4 %)
number of mismatched events, except outer(2)-inner(3)			
before	36417 (1.0 %)	16073 (1.2 %)	36236 (1.0 %)
after	22924 (0.7 %)	8445 (0.9 %)	25067 (0.7 %)
number of discarded events			
before	240 (0.071 permille)	107 (0.078 permille)	189 (0.055 permille)
after	57 (0.017 permille)	64 (0.071 permille)	160 (0.043 permille)

Table 1: *Statistics of the periods before and after the accident at the three energies .*

In Table 2, we have registered the number of events as seen by modules F1 and B2 for all combinations of minibunch numbers between 1 and 4. The first row of the table

B2 \ F1	**1**	**2**	**3**	**4**
1	287219 284637	3774 1359 box 1	1188 357 box 2	4 0
2	1709 2519 box 3	266283 263361	35451 31729	19 12
3	972 458 box 4	1598 1780 box 5	222801 221514	187 81
4	4 12	4 11	73 458	1 0

Table 2: *Number of Bhabha events (false Bhabha events subtracted) for diagonal 1.*

contains the minibunch number seen by module F1 whereas the first column gives the minibunch number seen by the diagonal module, B2. The number is the left (right) corner of each box shows counts in B2 (F1). The energy cut has been applied only on the diagonal

[5] This is illustrated by the plots of appendix E.

module. The result for the other two modules is similar. We see the good assignments in the blue boxes and the high-energy 2-3 mismatch in the red box. The mismatches in the yellow boxes are minibunch 3 events (all fills of the scan had up to three minibunches). Cases with less than 50 events (e.g. those with one minibunch number equal to 5) are not significant and have been discarded (see Table 1). What is thus left for us to examine is the mismatches contained in the green boxes. The corresponding (uncalibrated) energy distributions are given in fig. 3. Figs. 3(a) and 3(b) refer to box 1, figs. 3(c) and 3(d) refer

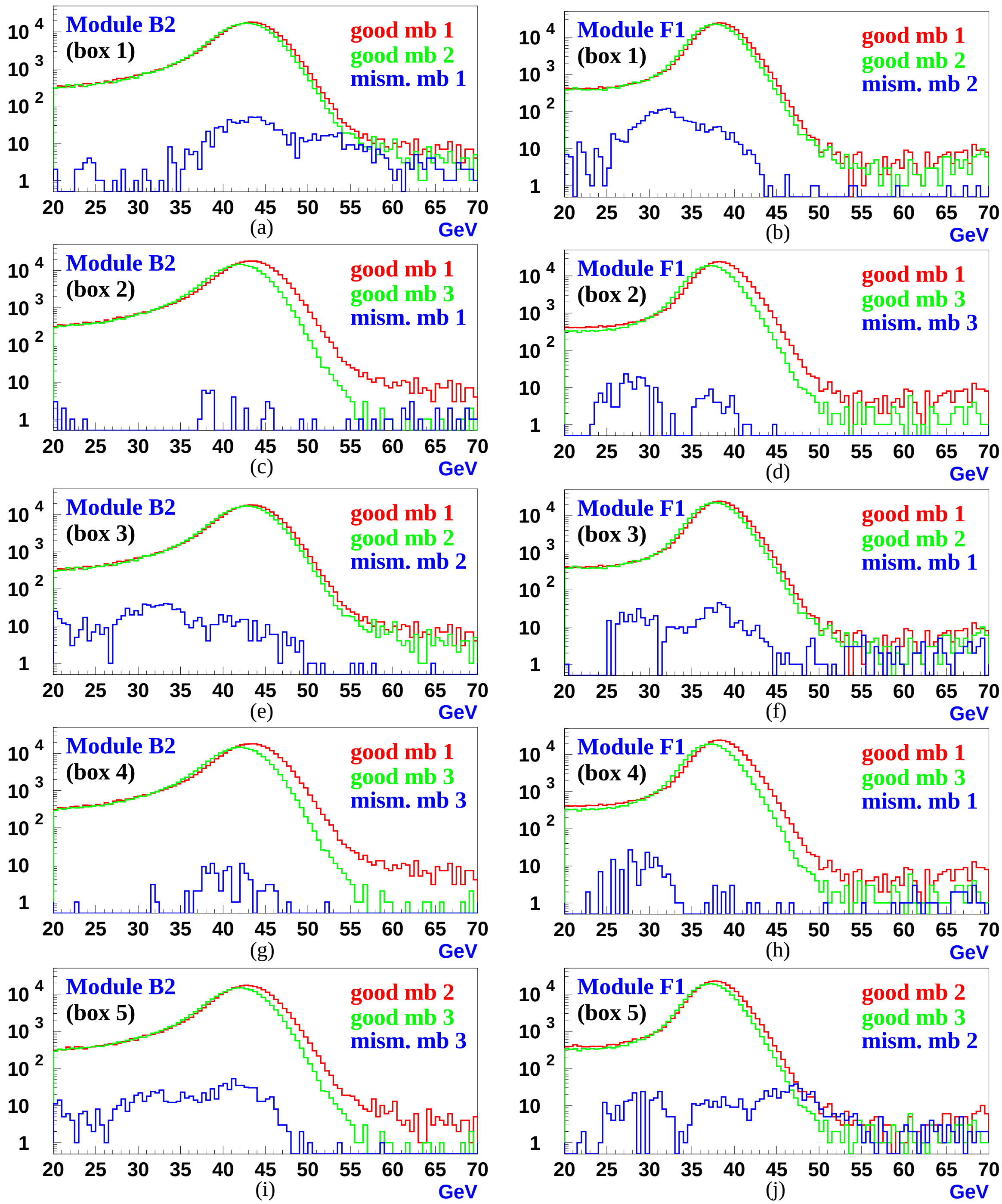

Figure 3: *Energy distributions for the mismatches of the green boxes of Table 2.*

to box 2 and so on. The green and red histograms show events where the two modules agreed on the minibunch number. The blue distributions refer to events for which F1 and

B2 gave different minibunch numbers. The corresponding correlations plots are shown in fig. 4. There is no energy cut applied and the false Bhabha events have not been

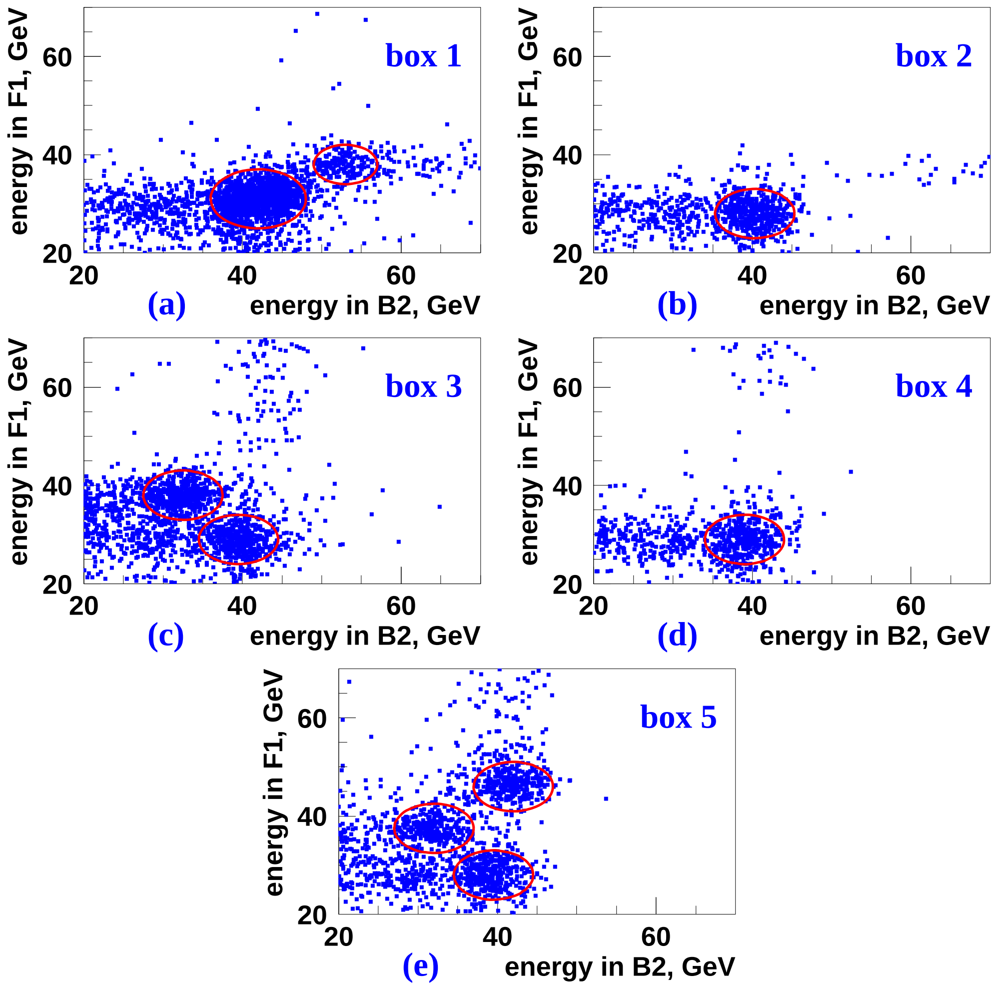

Figure 4: *Correlation plots for the blue histograms of fig. 3.*

subtracted. Their distributions have been omitted for clarity. The red ellipses are located at the peak positions of the blue spectra of fig. 3. It is clear that they contain Bhabha events. The correct minibunch number for those events can be deduced by the relative position of the mismatch and good assignment curves of fig. 3. This is discussed in the next section.

4 Minibunch tagging

In order to understand what the distributions of fig. 3 signify, we need to recall how the electronics assign the minibunch number. As was described briefly in section 1.2, there are three gates G1, G2 and G3 (fig. 5) looking for pulses above a certain threshold. When such pulses are found, the holds H1, H2 and H3 are sent to read the height of the signals. In fig. 5 we see pulses of correctly flagged events belonging to minibunch one (blue signal), two (red signal) and three (green signal). The mismatches can occur when a particle has energy which is sufficiently lower or higher than the average height signal. In fig. 5(a), we see a low energy pulse which has minibunch number equal to one (cyan signal). This pulse is below threshold (cyan line) while gate one is active. However, it is above threshold while the next gate is looking for high enough pulses and therefore it is assigned a wrong minibunch number equal to two. The energy which is subsequently read out by H2 is lower than the correct value that H1 should have seen. In fig. 5(b), we see a high energy minibunch two pulse (magenta signal), which is of course above threshold during G2, but which is also above threshold during G1. As G1 is generated earlier than G2, this pulse is wrongly assigned a minibunch number equal to one. In this case, H1 sees a higher energy value than the one H2 would have measured. In conclusion, low pulses are flagged as of higher minibunch number than the real one, whilst high pulses are flagged as of lower minibunch number than the correct one. The mismatch effect acts in the opposite direction as fas as energy values are concerned, i.e. high pulses are registered with even higher energy, whereas low pulses are registered with lower energy than the proper one.

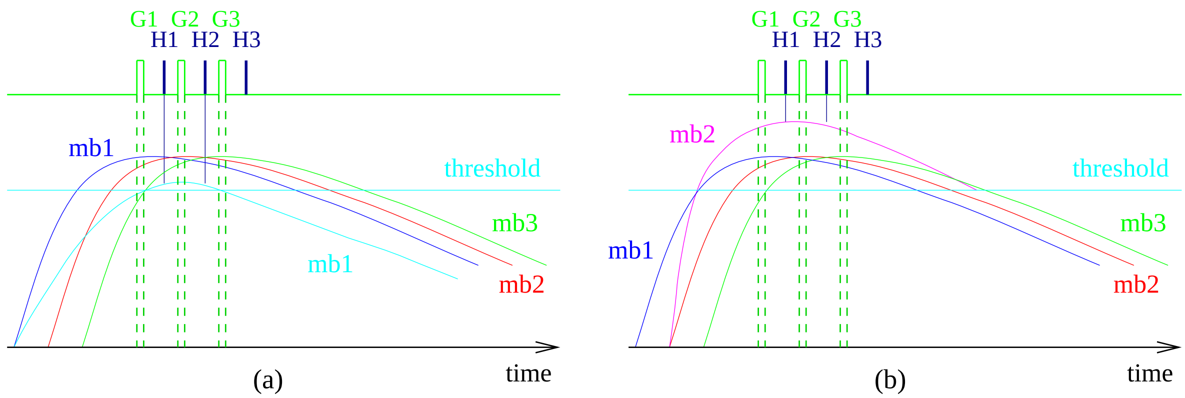

Figure 5: *Signals of correct minibunch assignments (blue, read and green curves) and mismatched pulses of (a) low energy (cyan curve) and (b) high energy (magenta curve).*

Bearing the above in mind, we can easily disentangle the various minibunch groups contained in the blue histograms of fig. 3. The red and green curves of the good assignments will help us because they show the position of the correctly flagged spectrum for the specific minibunch number. For example, in fig. 3(a), this suggests that module B2 was right about the first blue peak (at 44 GeV) because the red curve also peaks at about the same energy value. The second blue peak (at 54 GeV) is clearly a wrong assignment for B2. Indeed, the corresponding blue peak (at 39 GeV) of fig. 3(b) is close enough to the green peak to convince us that it originates from minibunch two events. The second blue peak of fig. 3(a) has a marked shift to the right with respect to the green peak as it contains signals higher than the average, as required for a wrong assignment to a lower minibunch number by module B2. Conversely, the minibunch one mismatches are expected to be of low energy in F1, since they have been assigned a higher minibunch number there. The subsequent energy read-out must have shifted them to even lower energies. This is precisely where the first blue peak of fig. 3(b) is found.

The situation we encounter in figs. 3(c) and (d) is somewhat different. The two peaks are visible in fig. 3(d) only. In order to match appropriate energy intervals for the Bhabha coincidences in the two modules, we need therefore to resort to fig. 4(b). We see one Bhabha peak for B2 energies between 35 and 45 GeV and for F1 energies from 25 to 35 GeV approximately. The corresponding events in fig. 3(c) are located close to the correct assignment peaks, therefore we accept the B2 minibunch number for them (one).

They also appear as very low energy events in F1, as they should since they were above threshold only when gate G3 was active. The F1 events which have energies between 35 and 40 GeV are in the proper place for being correctly flagged by this module. However, there is no sign of them in fig. 3(c). The explanation for this comes again from fig. 4(b). What we see there is that, within the 35-40 GeV F1 region, the B2 events are spread all over the interval between 35 and 70 GeV. What these hits really are is pile-ups between ordinary minibunch three events and background electrons. Since the resulting sum of their energies is really high, these overlaps are wrongly flagged as minibunch one events by B2. As the spectrum of the background is continuous, the shape of the original minibunch three spectrum in B2 is destroyed, hence no visible peak.

In figs. 3(e) and (f), we see two blue peaks. Let us first discuss the easier case, i.e. the second mismatch peak of fig. 3(f). This one has the right position to have been flagged correctly by F1. This means that it must appear at low energies in fig. 3(e) (being tagged with a higher minibunch number there) as indeed it does (at 30-35 GeV). Let us now look at the second blue peak of fig. 3(e). This seems to be correctly placed for corresponding to minibunch two events but this would entail that its partner peak in fig. 3(f) seats at high energies (being flagged as of a smaller minibunch number by F1). This is however not the case, as the latter energy range is between 25 and 30 GeV only. We thus conclude that these are minibunch one events as well, of low energy as confirmed in fig. 3(f) just below the threshold during G1 in B2[6].

Figs. 3(g) and (h) have only one mismatch peak which groups events between 35 and 45 GeV in B2 and between 25 and 35 GeV in F1. Those events being of low energy in fig. 3(h) cannot originate from minibunch three. The correct minibunch number is therefore the one given by module F1.

The situation is slightly more complicated in figs. 3(i) and (j), where three mismatch peaks appear very close to each other. From fig. 4(e), we see that the first blue peak of B2 (25-35 GeV) corresponds to F1 energies between 35 and 40 GeV. These are minibunch two events, of low energy in B2, where they were therefore flagged as of minibunch three. The first blue peak of F1 (23-32 GeV) corresponds to B2 energies between 35 and 45 GeV, as we see in fig. 4(e). These are also minibunch two events. Lastly, the third mismatch peak of F1, ranging from 42 to 50 GeV, matches B2 energies from 37 to 47 GeV (fig. 4(e)). Those events belong to minibunch three and are of high energy in F1, thus being assigned wrong minibunch number two by the forward module.

As we have now recovered the correct minibunch numbers for the green boxes of Table 2, we can proceed to split the corresponding energy spectra in order to group the Bhabha events according to their minibunch number (as required by the beam parameter and luminosity analyses). This is the subject of the following section.

5 Separation of energy distributions

The mismatches of boxes 2 and 4 do not require any further processing. On the hand, we need to separate the individual minibunch spectra of the remaining cases. To this end, we do gaussian fits to the energy distributions of the module that has the largest distance between the peaks. The energy value at which the resulting gaussian curves meet

[6] It may appear as if there were an overlap of two peaks in fig. 3(e) at the range 35-50 GeV but this is only because the plot is in log scale.

is used as our energy cut for the separation of events[7]. The fits for the blue histograms of fig. 3 are given in the first column of fig. 6. In the second column, we see the resulting individual gaussian distributions.

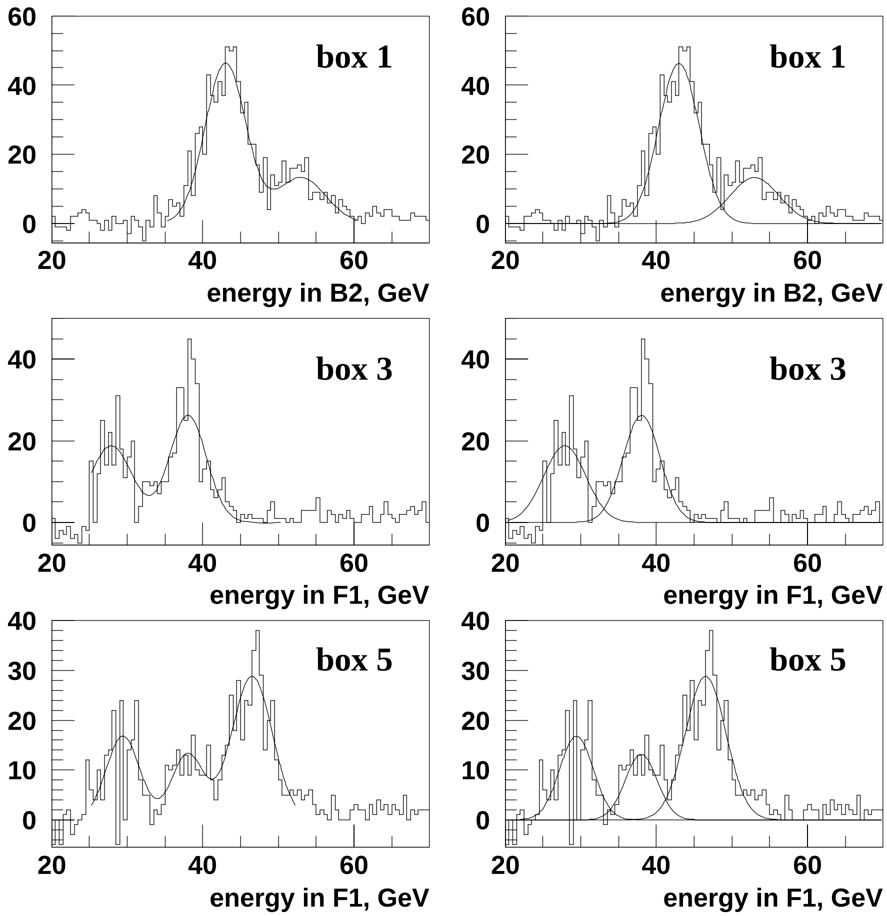

Figure 6: *Gaussian fits for the mismatches of the green boxes 1, 3 and 5 of Table 2.*

The calibration constants that we used for those low statistics mismatch cases are the corresponding ones of the good assignment distributions. Further corrections are not necessary since the relative population of events is of the order one percent.

Conclusions

The data of the 1995 scan included misidentified events representing approximately 10% of the total sample. With this analysis we have concluded the elaborate task of recovering them. The result has been satisfactory, the unretrieved data accounting now for less than 0.1 permille of the complete set.

[7]The code including all energy cuts for all beam energies, data taking periods and minibunch combinations is given in appendix F.

Acknowledgements

I would like to thank Andreas Nygren for recovering files from backups and for letting me use a substantial part of his disk space all the time that this analysis has lasted. Special thanks are due to Sverker Almehed for discussing the energy distributions with me. Lastly, I would like to thank Ulf Mjörnmark, Björn Lundberg and Janus Sørensen for taking a lot of trouble setting up my pc so that I could work at home. Without their help this report would not have been finished.

References

[1] Ch. Jarlskog, *VSAT energy calibration for 1995 minibunch data*, LUNFD6/(NFFL-7129)/1996.

[2] Almehed et al., *A silicon tungsten electromagnetic calorimeter for LEP*, Nucl. Instr. Meth. A305 1991.

[3] U. Mjörnmark, private communication.

[4] I. Kronkvist, *Data Base and Slow Controls System of the DELPHI VSAT and Two-Photon Physics using DELPHI at LEP* , PhD thesis, LUNFD6/(NFFL-7128) 1996.

A The Bhabha routine

We give below an extract of the Bhabha routine of the program IDMIS. The x-coordinates (in cm) of the edges of the module are denoted by `CUTINF` and `CUTSUP`. An event in the diagonal F1-B2 (F2-B1) of the detector is assigned a number `IBH=1` (`IBH=2`). The outer modules F2 and B2 are denoted by `IMOD1=3` and `IMOD1=1`. The inner modules F1, B1 have `IMOD2=4, 2`, respectively. The x-coordinate of the shower itself as measured by the first (second) x-plane is called `XREC(1,IMOD)` (`XREC(2,IMOD)`), where `IMOD` is the module number. The energy of the particle in module `IMOD` is called `ENECOR(IMOD)`. The first Bhabha cut removes particles that have the maximum energy deposit on the outer strips of the module. An event which survives this cut has `INEW_FLAG` set to 1. The second Bhabha cut removes mainly off-momentum particles. The remaining events have `IFLAG_BABA`=1. The minibunch number of the electron that hit module `IMOD` is denoted by `imini(imod)`.

```
+DECK,AN02_BH. ===========================================================
      SUBROUTINE AN02_BH(IENTRY)
*************************************************************************
* AN02_BH
*
*  Created     : 29-JUN-1992   Author : Maria Paola Clara
*
*  Function    : bhabha selection
*
*  Last modified: 9-10-97  S.Rinaudo Ch.Jarlskog
*
*************************************************************************
      ...
        CUTINF = 6.05
        CUTSUP = 8.95
      ...
      IF (IENTRY.EQ.1.AND.IENDCASSETTE.EQ.0) THEN
      ...

* determination of the arm

        IF(IBTRIB.EQ.4.OR.IBTRIB.EQ.2) THEN
          IF (IBTRIB.EQ.4.AND.BHHIT(1).AND.BHHIT(4)) THEN
            IBH = 1       ! diagonal 1
            IMOD1 = 1     ! module B2
            IMOD2 = 4     ! module F1
          ENDIF
          IF(IBTRIB.EQ.2.AND.BHHIT(2).AND.BHHIT(3)) THEN
            IBH = 2       ! diagonal 2
            IMOD1 = 3     ! module F2
            IMOD2 = 2     ! module B1
          END IF
```

```
      ...

* get xrec and calculate dx, dy, sx and diffy

* If the information from the first plane of x strips is missing, we take the
* information from the second plane

          IF (XREC(1,IMOD1).EQ.0..OR.XREC(1,   ! xrec(1,imod) means x-plane 1
     &        IMOD1).EQ.-200.) THEN
            XREC(1,IMOD1) = XREC(2,IMOD1)      ! xrec(2,imod) means x-plane 2
          ENDIF
          IF (XREC(1,IMOD2).EQ.0.OR.XREC(1,
     &        IMOD2).EQ.-200.) THEN
            XREC(1,IMOD2) = XREC(2,IMOD2)
          ENDIF

          DELTAXN = (ABS(XRECN(1,IMOD2)) - ABS(XRECN(1,IMOD1)))
          SUMMXN = (ABS(XRECN(1,IMOD2)) + ABS(XRECN(1,IMOD1)))

          DELTAX = (ABS(XREC(1,IMOD2)) - ABS(XREC(1,IMOD1)))
          SUMMX = (ABS(XREC(1,IMOD2)) + ABS(XREC(1,IMOD1)))
          DELTAY = YREC(1,IMOD1) + YREC(1,IMOD2)
          DIFFY = YREC(1,IMOD2) - YREC(1,IMOD1)

* flag event which passes the bhabha cuts

          IFLAG_BABA=0
          INEW_FLAG = 0

          IF(ABS(XREC(1,IMOD1)).GT.CUTINF.AND.ABS(XREC(1,IMOD1)).LT.
     &          CUTSUP.AND.ABS(XREC(1,IMOD2)).GT.CUTINF.AND.
     &          ABS(XREC(1,IMOD2)).LT.CUTSUP)  THEN

            INEW_FLAG = 1                     ! first Bhabha cut

* cut in energy of the Bhabha events

            IF(ENECOR(IMOD1).GT.0.7*EBEAM
     &          .AND.ENECOR(IMOD2).GT.0.7*EBEAM) THEN

              IFLAG_BABA=1                    ! second Bhabha cut

* counting BHABHA arm by arm

              NB(IBH) = NB(IBH) + 1

            ENDIF
```

```
      ENDIF

* ==================== Minibunch histograms ====================

C ---------- minibunch number different

      IF ((IMINI(2).NE.IMINI(3).AND.IMINI(2)*IMINI(3).NE.0).OR.
     1        (IMINI(1).NE.IMINI(4).AND.IMINI(1)*IMINI(4).NE.0)) THEN

        DO IM=1,2
          IF (IM.EQ.1) IMOD=IMOD1
          IF (IM.EQ.2) IMOD=IMOD2

          ID=IMOD*100+imini(imod)*10

          IF (INEW_FLAG.EQ.1.AND.ENECOR(5-IMOD)/
     &             EBEAM.GT.0.7) THEN

            if (imini(imod).eq.1) then
              if (imini(5-imod).eq.2) then
                CALL HFILL(71002+ID,ENECOR(IMOD),0.,1.)
              else if (imini(5-imod).eq.3) then
                CALL HFILL(71003+ID,ENECOR(IMOD),0.,1.)
              else if (imini(5-imod).eq.4) then
                CALL HFILL(71004+ID,ENECOR(IMOD),0.,1.)
              else
                CALL HFILL(71005+ID,ENECOR(IMOD),0.,1.)
              endif
            endif

            if (imini(imod).eq.2) then
              if (imini(5-imod).eq.1) then
                CALL HFILL(71001+ID,ENECOR(IMOD),0.,1.)
              else if (imini(5-imod).eq.3) then
                CALL HFILL(71003+ID,ENECOR(IMOD),0.,1.)
              else if (imini(5-imod).eq.4) then
                CALL HFILL(71004+ID,ENECOR(IMOD),0.,1.)
              else
                CALL HFILL(71005+ID,ENECOR(IMOD),0.,1.)
              endif
            endif

            if (imini(imod).eq.3) then
              if (imini(5-imod).eq.2) then
                CALL HFILL(71002+ID,ENECOR(IMOD),0.,1.)
              else if (imini(5-imod).eq.1) then
                CALL HFILL(71001+ID,ENECOR(IMOD),0.,1.)
```

```
          else if (imini(5-imod).eq.4) then
            CALL HFILL(71004+ID,ENECOR(IMOD),0.,1.)
          else
            CALL HFILL(71005+ID,ENECOR(IMOD),0.,1.)
          endif
        endif

        if (imini(imod).eq.4) then
          if (imini(5-imod).eq.2) then
            CALL HFILL(71002+ID,ENECOR(IMOD),0.,1.)
          else if (imini(5-imod).eq.3) then
            CALL HFILL(71003+ID,ENECOR(IMOD),0.,1.)
          else if (imini(5-imod).eq.1) then
            CALL HFILL(71001+ID,ENECOR(IMOD),0.,1.)
          else
            CALL HFILL(71005+ID,ENECOR(IMOD),0.,1.)
          endif
         endif

      ENDIF

c-- 2d histos for bhabhas without energy cut

      ID=IMOD*100+imini(imod)*10

      IF (INEW_FLAG.EQ.1) then

        if (imini(imod).eq.1) then
          if (imini(5-imod).eq.2) then
            CALL HFILL(1002+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else if (imini(5-imod).eq.3) then
            CALL HFILL(1003+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else if (imini(5-imod).eq.4) then
            CALL HFILL(1004+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else
            CALL HFILL(1005+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          endif
        endif

        if (imini(imod).eq.2) then
          if (imini(5-imod).eq.1) then
            CALL HFILL(1001+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else if (imini(5-imod).eq.3) then
            CALL HFILL(1003+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else if (imini(5-imod).eq.4) then
            CALL HFILL(1004+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
          else
```

```
              CALL HFILL(1005+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            endif
          endif

          if (imini(imod).eq.3) then
            if (imini(5-imod).eq.2) then
              CALL HFILL(1002+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else if (imini(5-imod).eq.1) then
              CALL HFILL(1001+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else if (imini(5-imod).eq.4) then
              CALL HFILL(1004+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else
              CALL HFILL(1005+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            endif
          endif

          if (imini(imod).eq.4) then
            if (imini(5-imod).eq.2) then
              CALL HFILL(1002+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else if (imini(5-imod).eq.3) then
              CALL HFILL(1003+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else if (imini(5-imod).eq.1) then
              CALL HFILL(1001+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            else
              CALL HFILL(1005+ID,ENECOR(IMOD),ENECOR(5-IMOD),1.)
            endif
          endif

        endif

      enddo
    ENDIF

C ----------- minibunch number ok in both modules

    IF ((IMINI(2).EQ.IMINI(3).AND.IMINI(2)*IMINI(3).NE.0).OR.
 1      (IMINI(1).EQ.IMINI(4).AND.IMINI(1)*IMINI(4).NE.0))  THEN

      DO IM=1,2
        IF (IM.EQ.1) IMOD=IMOD1
        IF (IM.EQ.2) IMOD=IMOD2
        IWAG=IMINI(IMOD)
        ID=IMOD*100+IWAG*10+iwag

        IF (INEW_FLAG.EQ.1.AND.ENECOR(5-IMOD)/EBEAM.GT.0.7)
 &          CALL HFILL(71000+ID,ENECOR(IMOD),0.,1.)
```

```
            ENDDO

          ENDIF  !end of analysis of minibunch number ok in both modules
*
        ENDIF  !end of bhabha analysis
      ENDIF  !end of ientry=1
      ...
```

B The false Bhabha routine

This is the false Bhabha routine of the program idmis.car. The two diagonals of the detector are denoted by `IFBH=1` and `IFBH=2`. The x-coordinate of the shower in the first x-plane is called `XFB(1,IMOD,IFBH)`, where `IMOD` is the module number. The energy of the particle in module `IMOD` is called `EFB(IMOD,IFBH)`. The same cuts on position and energy are applied here as in the Bhabha routine of appendix A. The minibunch number of a false Bhabha particle that hit module `IMOD` is stored in the variable `iminib(imod)`.

```
+DECK,AN02_FB. ==============================================================
      SUBROUTINE AN02_FB(IENTRY)
************************************************************************
* AN02_FB
*
*  Created     : 29-JUN-1993   Author : Luca Zanini
*
*  Function    : false bhabha selection
*
*  Last modified: 9-10-97 S.R. Ch.J.
************************************************************************
      ...
        CUTINF = 6.05
        CUTSUP = 8.95
      ...
        DO IMOD = 1 ,NMOD
          XFB(1,IMOD,IFBH) = XREC(1,IMOD)
          XFB(2,IMOD,IFBH) = XREC(2,IMOD)
          YFB(1,IMOD,IFBH) = YREC(1,IMOD)
          EFB(IMOD,IFBH) = ENECOR(IMOD)
          ...
          IMINI_FB(IMOD,IFBH) = IMINI(IMOD)
          ...
        ENDDO
        IF(IFBH.EQ.1)    GOTO 999
*       if IFBH = 2 go on with analysis

* modules hit

        DO IMOD = 1 , NMOD
          FBHHIT(IMOD,IFBH-1) = (JBIT(KTRIFB(IFBH),5-IMOD).EQ.1)
          FBHHIT(IMOD,IFBH) = (JBIT(IBTRFB(IFBH),10-2*IMOD).EQ.1)
        END DO

* arm selection

        IF(FBHHIT(4,IFBH-1).AND.FBHHIT(1,IFBH)) THEN
          IFB = 1
```

```
          IMOD1 = 1
          IMOD2 = 4
        ELSE IF(FBHHIT(2,IFBH-1).AND.FBHHIT(3,IFBH)) THEN
          IFB = 2
          IMOD1 = 3
          IMOD2 = 2
        END IF
      ...

* If the information from the first plane of x strips is missing,
* we take the information from the second plane

          IF (XFB(1,IMOD1,IFBH).EQ.0..OR.
     &      XFB(1,IMOD1,IFBH).EQ.-200.)
     &      XFB(1,IMOD1,IFBH) = XFB(2,IMOD1,IFBH)
          IF (XFB(1,IMOD2,IFBH-1).EQ.0..
     &      OR.XFB(1,IMOD2,IFBH-1).EQ.-200.)
     &      XFB(1,IMOD2,IFBH-1) = XFB(2,IMOD2,IFBH-1)

* Deltax for False Bhabha

          DELTAXFB = ABS(XFB(1,IMOD2,IFBH-1)) - ABS(XFB(1,IMOD1,IFBH))
          DELTAYFB = YFB(1,IMOD1,IFBH) + YFB(1,IMOD2,IFBH-1)
          SUMMXFB = ABS(XFB(1,IMOD2,IFBH-1)) + ABS(XFB(1,IMOD1,IFBH))
          DIFFYFB = YFB(1,IMOD2,IFBH-1) - YFB(1,IMOD1,IFBH)

* energies for False Bhabha

          ENERFB(IMOD1)=EFB(IMOD1,IFBH)
          ENERFB(IMOD2)=EFB(IMOD2,IFBH-1)

* flag of event passing the bhabha cut

          IFLAG_BABA=0
          INEW_FLAG = 0

* apply same cut in position as for the Bhabha events

          IF(ABS(XFB(1,IMOD1,IFBH)).GT.CUTINF.AND.ABS
     1          (XFB(1,IMOD1,IFBH)).LT.
     &          CUTSUP.AND.ABS(XFB(1,IMOD2,IFBH-1)).GT.CUTINF.AND.
     &          ABS(XFB(1,IMOD2,IFBH-1)).LT.CUTSUP)  THEN

            INEW_FLAG = 1

* apply the same cut in energy as for the Bhabha events
```

```
          IF(EFB(IMOD1,IFBH).GT.0.7*EBEAM
     &          .AND.EFB(IMOD2,IFBH-1).GT.0.7*EBEAM) THEN

            IFLAG_BABA=1

          ENDIF
        ENDIF

* ===================== Minibunch histograms =====================

C ----------- minibunch number different

        IMINIB(1)=IMINI_FB(1,IFBH)
        IMINIB(2)=IMINI_FB(2,IFBH-1)
        IMINIB(3)=IMINI_FB(3,IFBH)
        IMINIB(4)=IMINI_FB(4,IFBH-1)
        IF ((IMINIB(2).NE.IMINIB(3).AND.IMINIB(2)*IMINIB(3).NE.0).OR.
     1      (IMINIB(1).NE.IMINIB(4).AND.IMINIB(1)*IMINIB(4).NE.0)) THEN

          DO IM=1,2
            IF (IM.EQ.1)   THEN
              IMOD=IMOD1
              IND=IFBH
            ENDIF
            IF (IM.EQ.2)   THEN
              IMOD=IMOD2
              IND=IFBH-1
            ENDIF

            ID=IMOD*100+IMINIB(IMOD)*10

            IF (INEW_FLAG.EQ.1.AND.ENERFB(5-IMOD)/
     &           EBEAM.GT.0.7) THEN

              if (iminib(imod).eq.1) then
                if (iminib(5-imod).eq.2) then
                  CALL HFILL(71002+id,EFB(IMOD,IND),0.,-1.)
                else if (iminib(5-imod).eq.3) then
                  CALL HFILL(71003+ID,EFB(IMOD,IND),0.,-1.)
                else if (iminib(5-imod).eq.4) then
                  CALL HFILL(71004+ID,EFB(IMOD,IND),0.,-1.)
                else
                  CALL HFILL(71005+ID,EFB(IMOD,IND),0.,-1.)
                endif
              endif
              if (iminib(imod).eq.2) then
                if (iminib(5-imod).eq.1) then
```

```
              CALL HFILL(71001+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.3) then
              CALL HFILL(71003+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.4) then
              CALL HFILL(71004+ID,EFB(IMOD,IND),0.,-1.)
            else
              CALL HFILL(71005+ID,EFB(IMOD,IND),0.,-1.)
            endif
          endif

          if (iminib(imod).eq.3) then
            if (iminib(5-imod).eq.2) then
              CALL HFILL(71002+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.1) then
              CALL HFILL(71001+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.4) then
              CALL HFILL(71004+ID,EFB(IMOD,IND),0.,-1.)
            else
              CALL HFILL(71005+ID,EFB(IMOD,IND),0.,-1.)
            endif
          endif

          if (iminib(imod).eq.4) then
            if (iminib(5-imod).eq.2) then
              CALL HFILL(71002+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.3) then
              CALL HFILL(71003+ID,EFB(IMOD,IND),0.,-1.)
            else if (iminib(5-imod).eq.1) then
              CALL HFILL(71001+ID,EFB(IMOD,IND),0.,-1.)
            else
              CALL HFILL(71005+ID,EFB(IMOD,IND),0.,-1.)
            endif
          endif

        ENDIF

c-- 2d histos for false bhabhas without energy cut

        ID=IMOD*100+iminib(imod)*10

        IF (INEW_FLAG.EQ.1) then

          if (iminib(imod).eq.1) then
            if (iminib(5-imod).eq.2) then
              CALL HFILL(2002+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
            else if (iminib(5-imod).eq.3) then
              CALL HFILL(2003+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
```

```
          else if (iminib(5-imod).eq.4) then
            CALL HFILL(2004+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else
            CALL HFILL(2005+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          endif
        endif

        if (iminib(imod).eq.2) then
          if (iminib(5-imod).eq.1) then
            CALL HFILL(2001+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.3) then
            CALL HFILL(2003+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.4) then
            CALL HFILL(2004+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else
            CALL HFILL(2005+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          endif
        endif

        if (iminib(imod).eq.3) then
          if (iminib(5-imod).eq.2) then
            CALL HFILL(2002+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.1) then
            CALL HFILL(2001+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.4) then
            CALL HFILL(2004+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else
            CALL HFILL(2005+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          endif
        endif

        if (iminib(imod).eq.4) then
          if (iminib(5-imod).eq.2) then
            CALL HFILL(2002+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.3) then
            CALL HFILL(2003+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else if (iminib(5-imod).eq.1) then
            CALL HFILL(2001+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          else
            CALL HFILL(2005+ID,ENErfb(IMOD),ENErfb(5-IMOD),1.)
          endif
        endif

      endif
    ENDDO
  ENDIF
```

```
C ----------- minibunch number ok in both modules

          IF ((IMINIB(2).EQ.IMINIB(3).AND.IMINIB(2)*IMINIB(3).NE.0).OR.
     1      (IMINIB(1).EQ.IMINIB(4).AND.IMINIB(1)*IMINIB(4).NE.0))  THEN

            DO IM=1,2
              IF (IM.EQ.1)   THEN
                IMOD=IMOD1
                IND=IFBH
              ENDIF
              IF (IM.EQ.2)   THEN
                IMOD=IMOD2
                IND=IFBH-1
              ENDIF
              IWAG=IMINIB(IMOD)
              ID=IMOD*100+IWAG*10+iwag

              IF (INEW_FLAG.EQ.1.AND.ENERFB(5-IMOD)/EBEAM.GT.0.7)
     &          CALL HFILL(71000+ID,EFB(IMOD,IND),0.,-1.)

            ENDDO

          ENDIF  !end of analysis of minibunch number ok in both modules
*
        ENDIF  !end of false bhabha selection
      ENDIF   !end of ientry=1
      IFBH = 0
      ...
```

C Data files, FORTRAN code and kumacs

We give here the list of the main files we used for this analysis.

- **IDMIS** The code for the program and the files containing pedestals and beam energies are in the directory:

```
Directory WSDE99$DKB500:[VSAT.CHRISTINA.DELB.CAL.SCAN]

CALIBCONST.TXT   ENERGYCALB.TXT   HISTO.DIR        IDMIS.CAR
IDMIS.COM        IDMIS.COM_ALL    IDMIS.CRA        IDMIS.TIT
PEDESTAL.DAT     RES.DIR          ZACHARIT.DIS
```

- **Calibration constants** IDMIS calculates calibration constants for all cases of mismatched events.

```
Directory WSDE99$DKB500:[VSAT.CHRISTINA.DELB.CAL.SCAN.RES]

MISM_2906_B.RES  MISM_2907_B.RES  MISM_2908_B.RES  MISM_2931_B.RES
MISM_2932_B.RES  MISM_2933_B.RES  MISM_2934_B.RES  MISM_2936_B.RES
MISM_2938_B.RES  MISM_2939_B.RES  MISM_2943_B.RES  MISM_2944_B.RES
MISM_2945_B.RES  MISM_2948_B.RES  MISM_2949_B.RES  MISM_2951_B.RES
MISM_2952_B.RES  MISM_2953_B.RES  MISM_2955_B.RES  MISM_2956_B.RES
MISM_2958_B.RES  MISM_2959_B.RES  MISM_2962_B.RES  MISM_2965_B.RES
MISM_2969_B.RES  MISM_2970_B.RES  MISM_2973_B.RES  MISM_2975_B.RES
MISM_3007_B.RES  MISM_3008_B.RES  MISM_3010_B.RES  MISM_3012_B.RES
MISM_3013_B.RES  MISM_3016_B.RES  MISM_3017_B.RES  MISM_3019_B.RES
MISM_3020_B.RES  MISM_3022_B.RES  MISM_3023_B.RES  MISM_3024_B.RES
MISM_3025_B.RES  MISM_3027_B.RES  MISM_3029_B.RES  MISM_3030_B.RES
MISM_3032_B.RES  MISM_3036_B.RES  MISM_3039_B.RES  MISM_3042_B.RES
```

- **Histograms** The output histogram files of IDMIS are called `MISM_xxxx.OUT`, where `xxxx` is the fill number. The summed energy distributions are stored in the hbook files. In the name of the file we read the nominal energy (`P` for peak, `PM2` for peak-2 and `PP2` for peak+2) and whether the data were taken before or after the radiation accident.

```
Directory WSDE99$DKB500:[VSAT.CHRISTINA.DELB.CAL.SCAN.HISTO]

ADD1.KUMAC       ADD2.KUMAC       ADD3.KUMAC       ADD4.KUMAC
ADD5.KUMAC       ADD6.KUMAC       LAST.KUMAC       MISM_2906.OUT
MISM_2907.OUT    MISM_2908.OUT    MISM_2931.OUT    MISM_2932.OUT
MISM_2933.OUT    MISM_2934.OUT    MISM_2936.OUT    MISM_2938.OUT
MISM_2939.OUT    MISM_2943.OUT    MISM_2944.OUT    MISM_2945.OUT
MISM_2948.OUT    MISM_2949.OUT    MISM_2951.OUT    MISM_2952.OUT
MISM_2953.OUT    MISM_2955.OUT    MISM_2956.OUT    MISM_2958.OUT
MISM_2959.OUT    MISM_2962.OUT    MISM_2965.OUT    MISM_2969.OUT
```

MISM_2970.OUT	MISM_2973.OUT	MISM_2975.OUT	MISM_3007.OUT
MISM_3008.OUT	MISM_3010.OUT	MISM_3012.OUT	MISM_3013.OUT
MISM_3016.OUT	MISM_3017.OUT	MISM_3019.OUT	MISM_3020.OUT
MISM_3022.OUT	MISM_3023.OUT	MISM_3024.OUT	MISM_3025.OUT
MISM_3027.OUT	MISM_3029.OUT	MISM_3030.OUT	MISM_3032.OUT
MISM_3036.OUT	MISM_3039.OUT	MISM_3042.OUT	PAFT.HBOOK
PAFT.KUMAC	PAW.METAFILE	PBEF.HBOOK	PBEF.KUMAC
PM2AFT.HBOOK	PM2AFT.KUMAC	PM2BEF.HBOOK	PM2BEF.KUMAC
PP.KUMAC	PP.OUT	PP2AFT.HBOOK	PP2AFT.KUMAC
PP2BEF.HBOOK	PP2BEF.KUMAC		

D The *analyze_sum* routine

This routine is executed from paw. It takes as input the histogram files of the summed distributions of Appendix C. It creates an ntuple file and a histogram file, where the plots of Appendix E are stored. The ntuple was used to produce Table 2.

```
c     file /us2/christin/lic/mism/analyze_sum.f

      subroutine analyze_sum

c     analyzes the summed energy spectra stored in the hbook files
c     of wsde99$dkb500:[vsat.christina.delb.cal.scan.histo] (output
c     of idmis.car)   last update: 4-feb-98   author: ch.j.

c     event=histogram
c     entries per event: iebeam (nominal energy),ifill (=1,bef, =2, aft),
c     imod,imini,iop (mb number of opposite module),
c     entries,emean,erms (statistics),
c     aver,sigma,chi2 (of gaussian fit)

c     each hbook file contains 4(mod)*4(mb)*5(opp mb) histos

      parameter (nwpawc = 300000)
      common /pawc/ hmemor(nwpawc)
      dimension xdat(12)
      character*8 chtags(12)
      data chtags/'ieb','ifi','imod','imini','iop','entries',
     & 'emean','erms','aver','sd','chi2','enom'/

      call hropen(2,'ntuple','anal.ntp','n',1024,istat)
      if (istat.ne.0) write(*,*) 'cannot open ntuple'
      call hbookn(35,'Scan 95',12,'//ntuple',5000,chtags)

      call hropen(3,'plots','plots.hbook','n',1024,istat)
      if (istat.ne.0) write(*,*) 'cannot open plot-file'
      do jeb = 1,3
        do jfi = 1,2
          do j = 1,2
            do i = 1,4
              call hbook2(100*jeb+10*jfi+i+1000*(j-1),'imini vs iop',
     1                    13,0,6,13,0,6,0.)
            enddo
          enddo
        enddo
      enddo

c     iebeam = 1 => peak, iebeam = 2 => p-2, iebeam = 3 => p+2
```

```
      call process(1,1,'pbef.out','data','//data') !  pbef.hbook
      call process(1,2,'paft.out','datb','//datb') !  paft.hbook
      call process(2,1,'pm2b.out','datc','//datc') !  pm2bef.hbook
      call process(2,2,'pm2a.out','datd','//datd') !  pm2aft.hbook
      call process(3,1,'pp2b.out','date','//date')  !  pp2bef.hbook
      call process(3,2,'pp2a.out','datf','//datf') !  pp2aft.hbook

      call hcdir('//ntuple',' ')
      call hrout(35,icycle,' ')
      call hrend('ntuple')
      call hcdir('//plots',' ')
      call hrout(0,icycle,'n')
      call hrend('plots')

 99   continue
      end

************************************************************

      subroutine process(iebeam,ifill,filename,dname,dname1)

************************************************************

      character*8 filename
      character*4 dname
      character*6 dname1
      dimension par(3),sigpar(3),xdat(11)
      total = 0  ! total number of events
      totmis = 0 ! total number of mismatched events
      lowmis = 0 ! total number of mismatched events except h.e. 2-3
      rejected = 0   ! total number of 'ignored' events
      xdat(1) = iebeam
      xdat(2) = ifill

      lrecl=0
      call hropen(1,dname,filename,' ',lrecl,istat)
      call hrin(0,999999,0)

      do imod = 1,4
        xdat(3) = imod
        idall = iebeam*100+ifill*10+imod
        idmis = iebeam*100+ifill*10+imod+1000
        do imini = 1,4
          xdat(4) = imini
          do iop = 1,5
            xdat(5) = iop
```

```
        id=5000+imod*100+imini*10+iop
        call hnoent(id,noent)
        xdat(6) = noent
        total = total+noent
        if (imini.ne.iop) totmis=totmis+noent
        if (noent.lt.50) rejected = rejected+noent
        if (noent.ne.0) then
          xdat(7) = hstati(id,1,' ',0)
          xdat(8) = hstati(id,2,' ',0)
          call hfithn(id,'g',' ',4,par,step,pmin,pmax,sigpar,chi2)
          xdat(9) = par(2)
          xdat(10) = par(3)
          xdat(11) = chi2
        else
          xdat(7) = 0
          xdat(8) = 0
          xdat(9) = 0
          xdat(10) = 0
          xdat(11) = 0
        endif

        iflag = 0
        if (imod.eq.1.and.imini.eq.2.and.iop.eq.3) iflag=1
        if (imod.eq.2.and.imini.eq.3.and.iop.eq.2) iflag=1
        if (imod.eq.3.and.imini.eq.2.and.iop.eq.3) iflag=1
        if (imod.eq.4.and.imini.eq.3.and.iop.eq.2) iflag=1

        if (iebeam.eq.1) xdat(12)=45.66
        if (iebeam.eq.3) xdat(12)=46.51
        if (iebeam.eq.2) xdat(12)=44.74

        call hcdir('//ntuple',' ')
        call hfn(35,xdat)
        call hcdir('//plots',' ')
        call hfill(idall,xdat(4),xdat(5),xdat(6))
        if (xdat(4).ne.xdat(5).and.iflag.eq.0) then
          call hfill(idmis,xdat(4),xdat(5),xdat(6))
          lowmis=lowmis+noent
        endif
        call hcdir(dname1,' ')

      enddo
    enddo
  enddo

  call hrend(dname)
```

```
 do imod = 1,4
   do imini = 1,4
     do iop = 1,5
       id=5000+imod*100+imini*10+iop
       call hdelet(id)
     enddo
   enddo
 enddo

 write(*,*) 'Total number of events = ',total
 write(*,*) 'Total number of mism. events = ',totmis,totmis/total
 write(*,*) 'Total number of mism. events, ex. h.e. 2-3= ',lowmis,
1    lowmis/total
 write(*,*) 'Total number of rejected events = ',rejected,
1    rejected/total

 return
 end
```

E Bhabha populations for different minibunch combinations

In figs. 7-12, we give the number of Bhabha events counted by modules F1 and B1. The plots cover all minibunch number combinations at all beam energies and for both periods before and after the radiation accident. Plots (a) and (c) contain the good assignments and all mismatch cases to give an impression of their relative importance. Plots (b) and (d) contain the mismatches treated in this report. The figures demonstrate that the detector has been consistent as far as the Bhabha populations are concerned in all minibunch cases and under all data taking conditions.

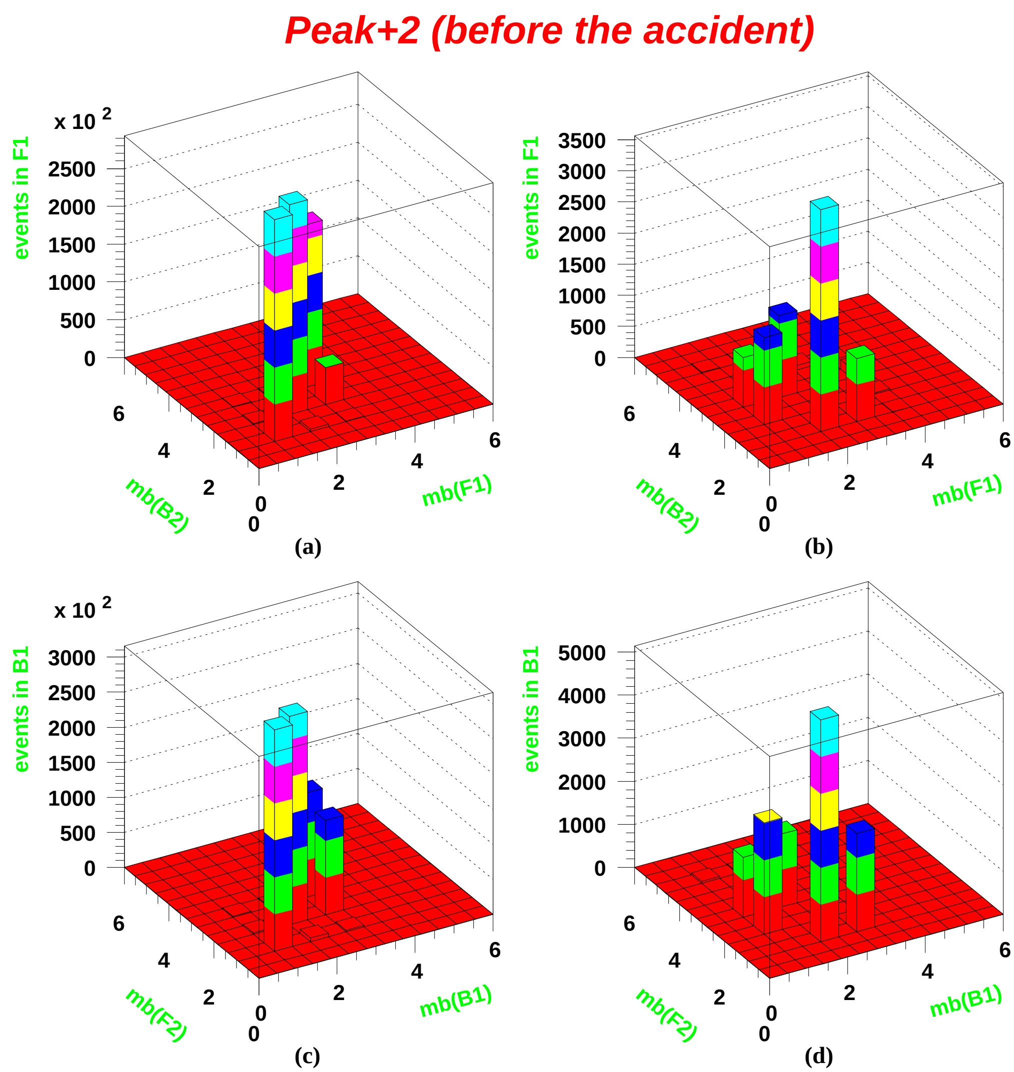

Figure 7: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

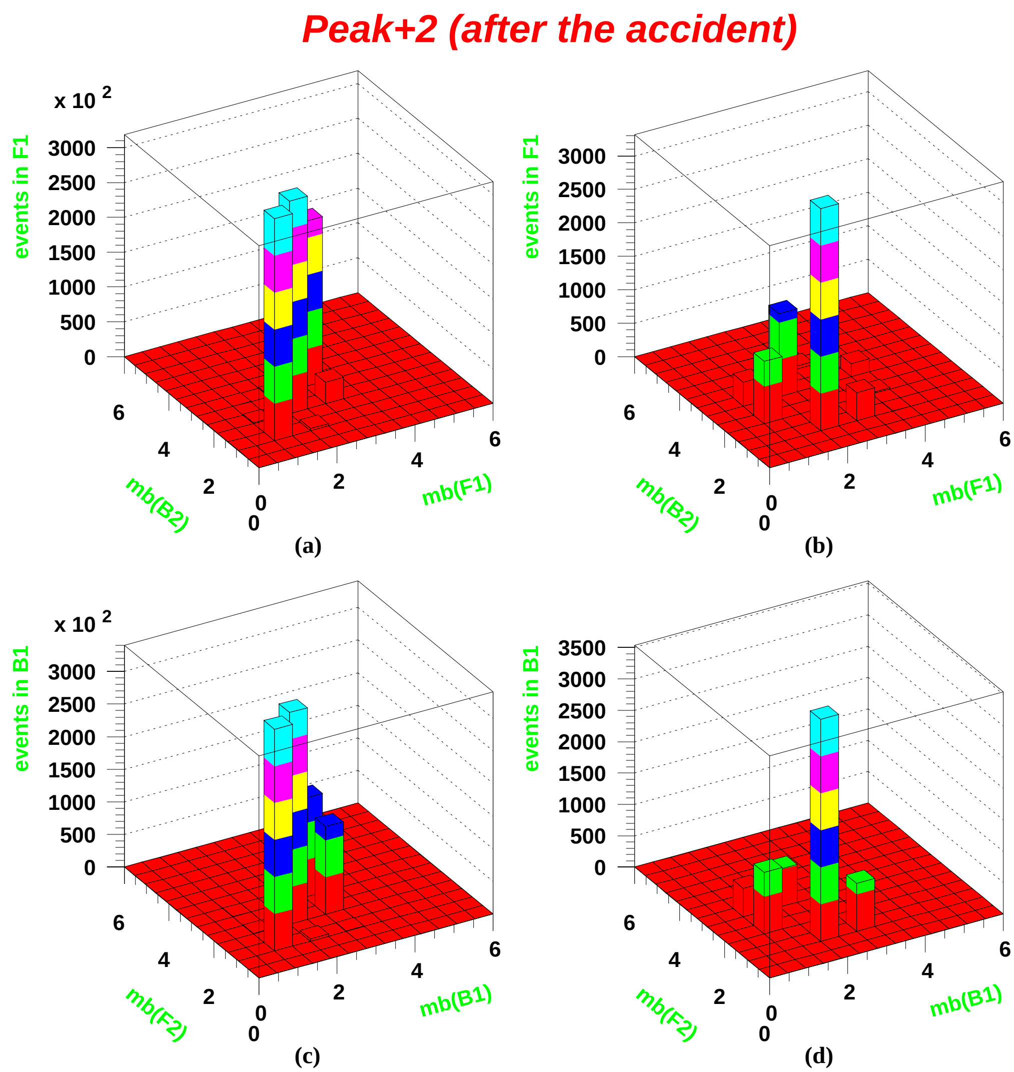

Figure 8: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

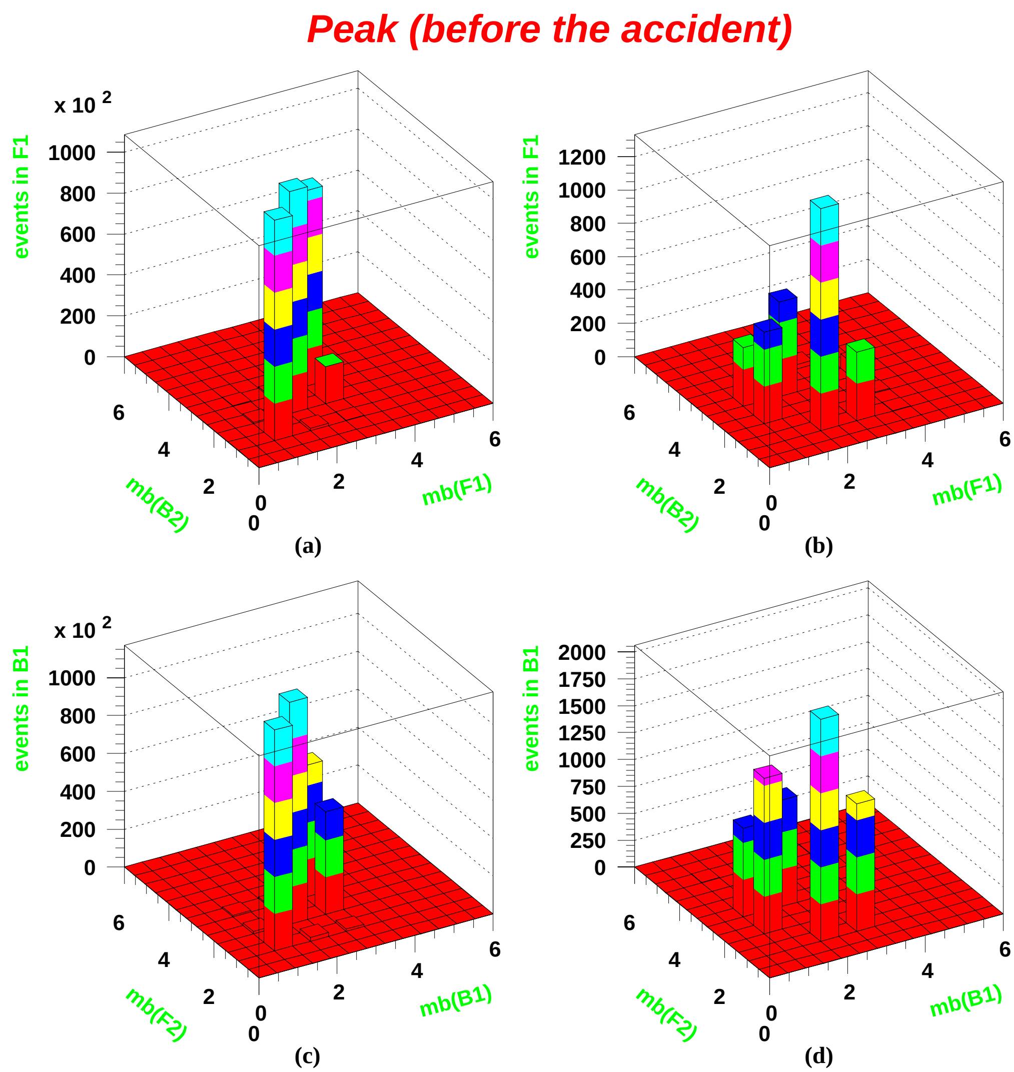

Figure 9: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

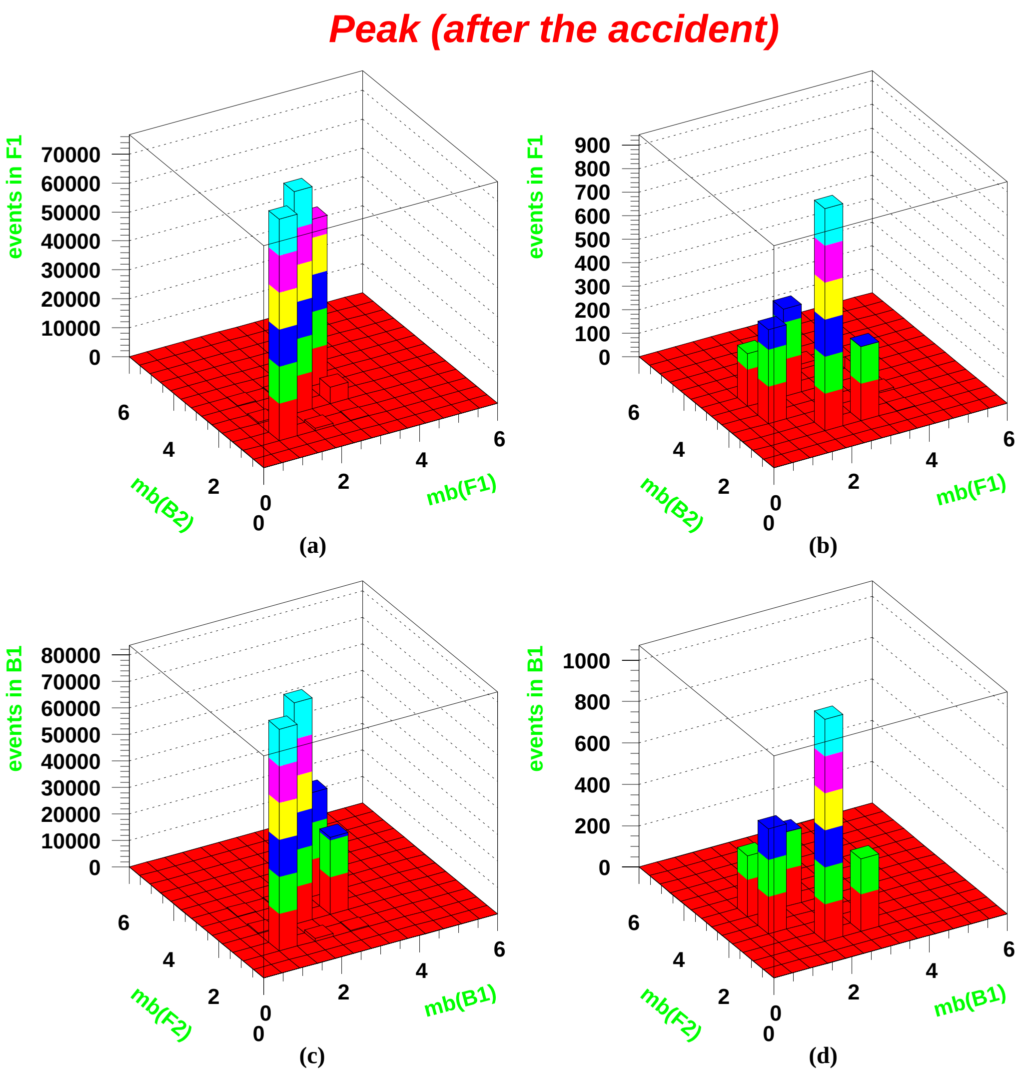

Figure 10: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

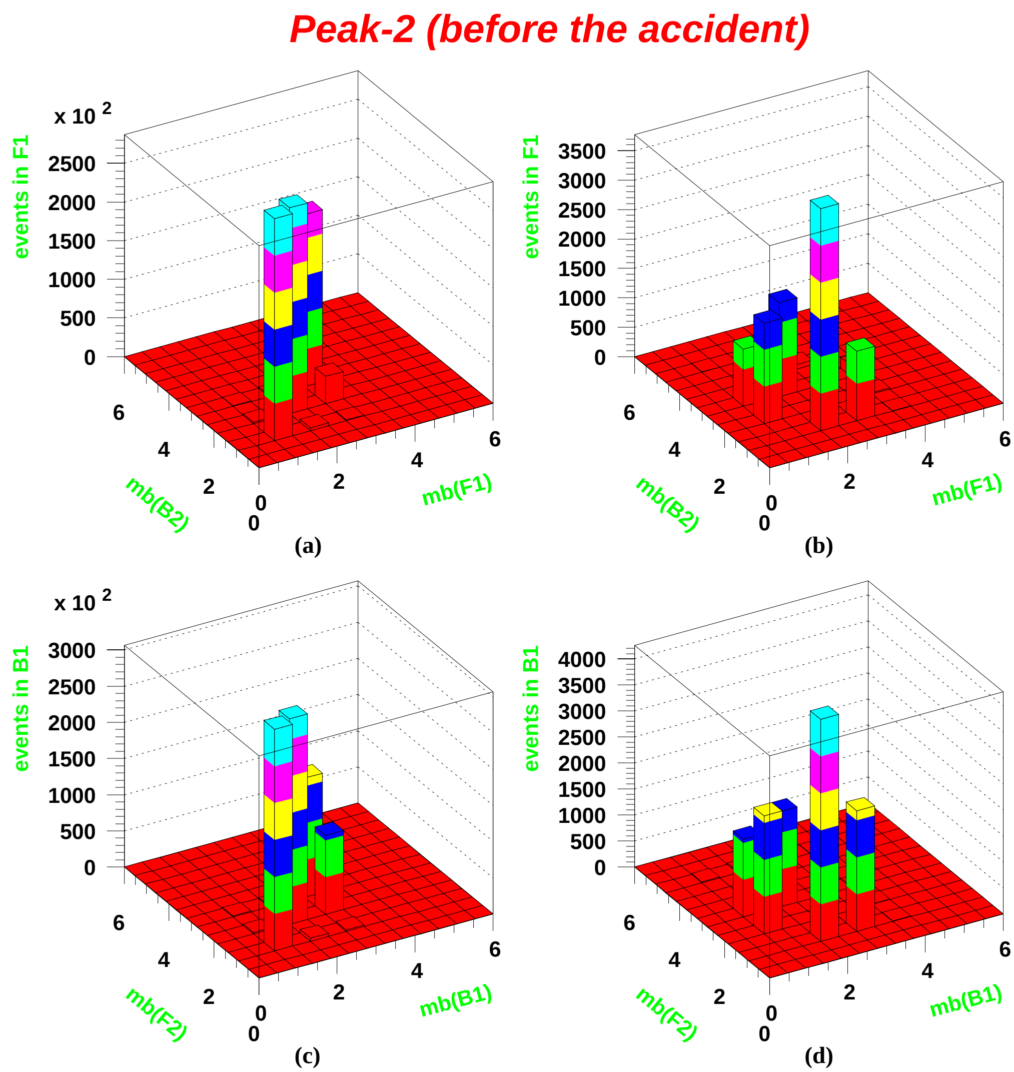

Figure 11: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

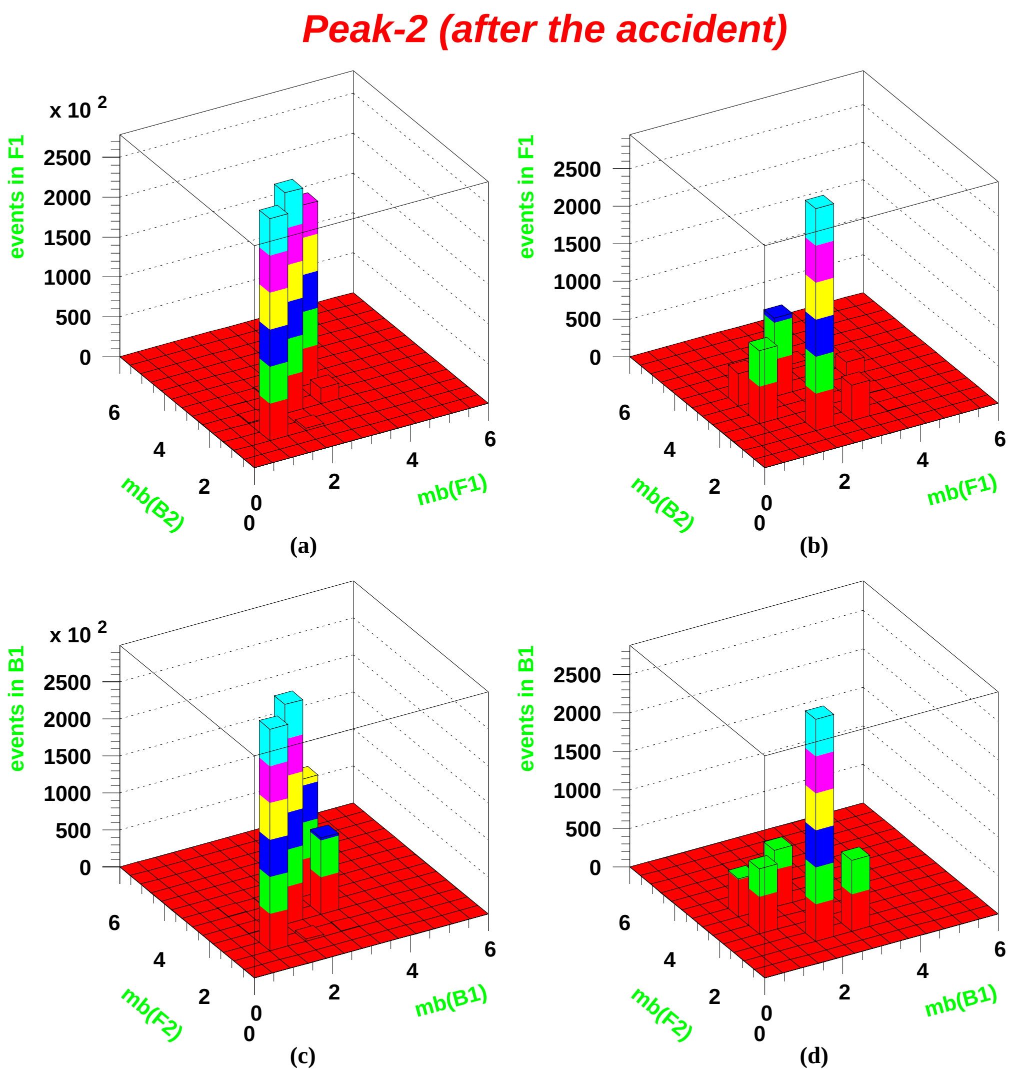

Figure 12: *Number of Bhabha events counted by the inner modules. The 70 % cut on the beam energy has not been applied.*

F Minibunch number tagging in vsdst

In this section, we give a listing of the part of the Bhabha routine that performs the minibunch number assignment for all events. It comes from the program vsdst.car, which is the main program for the final analysis of the data.

```
          ibun1 = imini(imod1) ! imod1=B2,F2
          ibun2 = imini(imod2) ! imod2=B1,F1
          iwag = 5
* the assignments are done as described in my report, 21-8-98, ch. j.

* GREEN BOXES
* box 1
       if(ibun1.eq.1.and.ibun2.eq.2) then
         if (ifill.lt.3000) then
           if (ebeam.lt.45..and.ibh.eq.1) cut=48.5
           if (ebeam.gt.46..and.ibh.eq.1) cut=51.
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.1) cut=49.7
           if (ebeam.lt.45..and.ibh.eq.2) cut=40.
           if (ebeam.gt.46..and.ibh.eq.2) cut=42.
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.2) cut=41.
         endif
         if (ifill.gt.3000) then
           if (ebeam.lt.45..and.ibh.eq.1) cut=47.8
           if (ebeam.gt.46..and.ibh.eq.1) cut=47.8
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.1) cut=47.8
           if (ebeam.lt.45..and.ibh.eq.2) cut=40.
           if (ebeam.gt.46..and.ibh.eq.2) cut=42.
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.2) cut=41.
         endif

         ene=enecor(imod1)*cal_minib(imod1,ibun1)
         if(ene.le.cut) then
           iwag=1
         else
           iwag=2
         endif
       endif
* box 2
       if(ibun1.eq.1.and.ibun2.eq.3) then
         if (ifill.lt.3000) then
           if (ibh.eq.1) cut=34.
           if (ebeam.lt.45..and.ibh.eq.2) cut=32.
           if (ebeam.gt.46..and.ibh.eq.2) cut=34.
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.2) cut=33.
         endif
```

```
         if (ifill.gt.3000) cut=34.

         ene=enecor(imod2)*cal_minib(imod2,ibun2)
         if(ene.le.cut) then
           iwag=1
         else
           iwag=3
         endif
       endif
* box 3
        if(ibun1.eq.2.and.ibun2.eq.1) iwag=1
* box 4
        if(ibun1.eq.3.and.ibun2.eq.1) iwag=1
* box 5
       if(ibun1.eq.3.and.ibun2.eq.2) then
         if (ifill.lt.3000) then
           if (ebeam.lt.45..and.ibh.eq.1) cut=41.2
           if (ebeam.gt.46..and.ibh.eq.1) cut=42.5
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.1) cut=41.8
           if (ebeam.lt.45..and.ibh.eq.2) cut=42.
           if (ebeam.gt.46..and.ibh.eq.2) cut=44.
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.2) cut=43.
         endif
         if (ifill.gt.3000) then
           if (ebeam.lt.45..and.ibh.eq.1) cut=41.2
           if (ebeam.gt.46..and.ibh.eq.1) cut=41.7
           if (ebeam.lt.46..and.ebeam.gt.45..and.ibh.eq.1) cut=41.4
           if (ibh.eq.2) cut=44.
         endif

         ene=enecor(imod2)*cal_minib(imod2,ibun2)
         if(ene.le.cut) then
           iwag=2
         else
           iwag=3
         endif
       endif

* YELLOW BOXES
        if(ibun1.eq.3.and.ibun2.eq.4.and.ibh.eq.1) iwag=3
        if(ibun1.eq.4.and.ibun2.eq.3.and.ibh.eq.1) iwag=3
* BLUE BOX (good assignment)
          if (ibun1.eq.ibun2.and.ibun1.gt.0.and.ibun1.le.3)
     1                                                iwag = ibun1
* RED BOX (high energy 2-3)
          if (ibun1.eq.2.and.ibun2.eq.3)              iwag = 4
```