LUNFD6/(NFFL-7145)/1997

Beam Parameter Analysis for the VSAT

September 4, 1997

Christina Jarlskog
Lund University
Sweden

Abstract

This is a manual describing the FORTRAN code and PAW kumacs that are used for the beam parameter analysis of VSAT data.

Contents

Part I

User Manual

1.1 Introduction

In this report, we describe how the beam parameter estimation is done for VSAT data and how it is compared with the VD and TPC beamspot determination. The files we use to illustrate the steps of the analysis concern the period 2 of 1995 data. Information per minibunch has been neglected for the sake of simplicity. The procedure we follow can be seen at a glance in fig. 1.1.

1.2 What do we need to start the analysis?

To start the analysis, we first need the VSAT ntuple, mylr95p2.ntp. This is produced by minilumi.for and can be found in wsde76$dka400:[vsat.offline95.luminos.chris]. This ntuple contains one entry per cassette for all the fills of 95 period 2 (2906 to 3042). It concerns only the total volume, the radial cut and fiducial volume being used only for the luminosity analysis. The ntuple variables are filled (a) per module and (b) per module and minibunch number. In the following sections, we will see how the former are processed and compared with the VD/TPC beamspot file.

The VD/TPC beamspot file, 95d2.db, can be found in the DELPHI_DAT area. It contains one entry per cassette. Information per minibunch has not been included.

Once we have fetched mylr95p2.ntp and 95d2.db, we can start our analysis. All the files we will use, can be found in /afs/cern.ch/user/z/zacharit/public/beamp.

1.3 How is the analysis done?

We will now decribe the steps one needs to follow in order to produce the ntuples for the beam parameter analysis. The necessary programs and kumacs can be found in the second part of this report. The formulae can be found in [1].

- **Creating an ntuple for VD/TPC** : we run the kumac makevdntp to read the file 95d2.db into the ntuple vd.ntp. The ntuple variables are: x_{VD}, dx_{VD}, y_{VD}, dy_{VD}, z_{TPC} and dz_{TPC} in mm and the fill number.

- **Extracting cassette numbers for VD/TPC** : we save the 3380 cassette numbers of VD/TPC data in the file casvd.dat by running casvd.f on 95d2.db.

- **Extracting cassette numbers for VSAT** : we run casvsat.kumac on mylr95p2.ntp to produce the file casvsat.dat, which contains the 1214 cassette numbers of VSAT data. The kumac uses the paw function copy.f, which has been generated with the uwfunc command.

- **Finding common cassette numbers** : in order to compare our data with the VD/TPC data, we need to find which cassette numbers they have in common. This is the task of comcas.f, which reads casvd.dat and casvsat.dat and stores 1178 common cassette numbers in casnum.dat.

- **Creating VD/TPC data file for common cassettes** : the program getvddata.f reads the files 95d2.db, casvd.dat and casnum.dat and saves the VD/TPC beamspot entries that correspond to the common cassettes in the file tpc.p2.

- **Extracting general VSAT quantities** : the kumac loopGeneral reads from the ntuple mylr95p2.ntp the following variables: fill number, beam energy, day, month and run (also referred to as "cassette number" in this manual). Each variable is stored in a separate text file in the subdirectory general/.

- **Extracting VSAT quantities per module** : we run the kumac loopModule to read from mylr95p2.ntp the quantities per module, i.e. those including all mini-bunches. The following beam parameters are calculated: x_{VSAT}, y_{VSAT}, z_{VSAT} and their errors and the diagonal asymmetry, θ_x and θ_y. Each variable is saved in a separate text file in the subdirectory module/.

- **Preparing text file with all VSAT data** : the program datamod.f merges all files of general/ and module/ into modp2.dat.

- **Creating VSAT ntuple** : the kumac makevsatntp reads modp2.dat into the ntuple vsat.ntp.

- **Preparing text file with common cassette VSAT data** : the data that correspond to common cassette numbers are saved in module.p2. This file is created by the program getvsatdata.f, which takes as input casnum.dat and the contents of general/ and module/.

- **Reading x-beamspot and z-beamspot mean values** : in order to calculate the normalized differences between our beamspot values for x and z and the values of VD, TPC, we need their mean values that we read with means.kumac from mylr95p2.ntp and vd.ntp.

- **Preparing text file with common cassette VD/TPC/VSAT data** : the files tpc.p2 and module.p2 are merged into data.p2mod by the program getalldata.f. The acollinearities in x and y and the normalized differences for x-beamspot and z-beamspot are also calculated and saved in the new file.

- **Preparing ntuple with common cassette VD/TPC/VSAT data** : data.p2mod is read into the ntuple p2mod.ntp by the kumac p2mod.

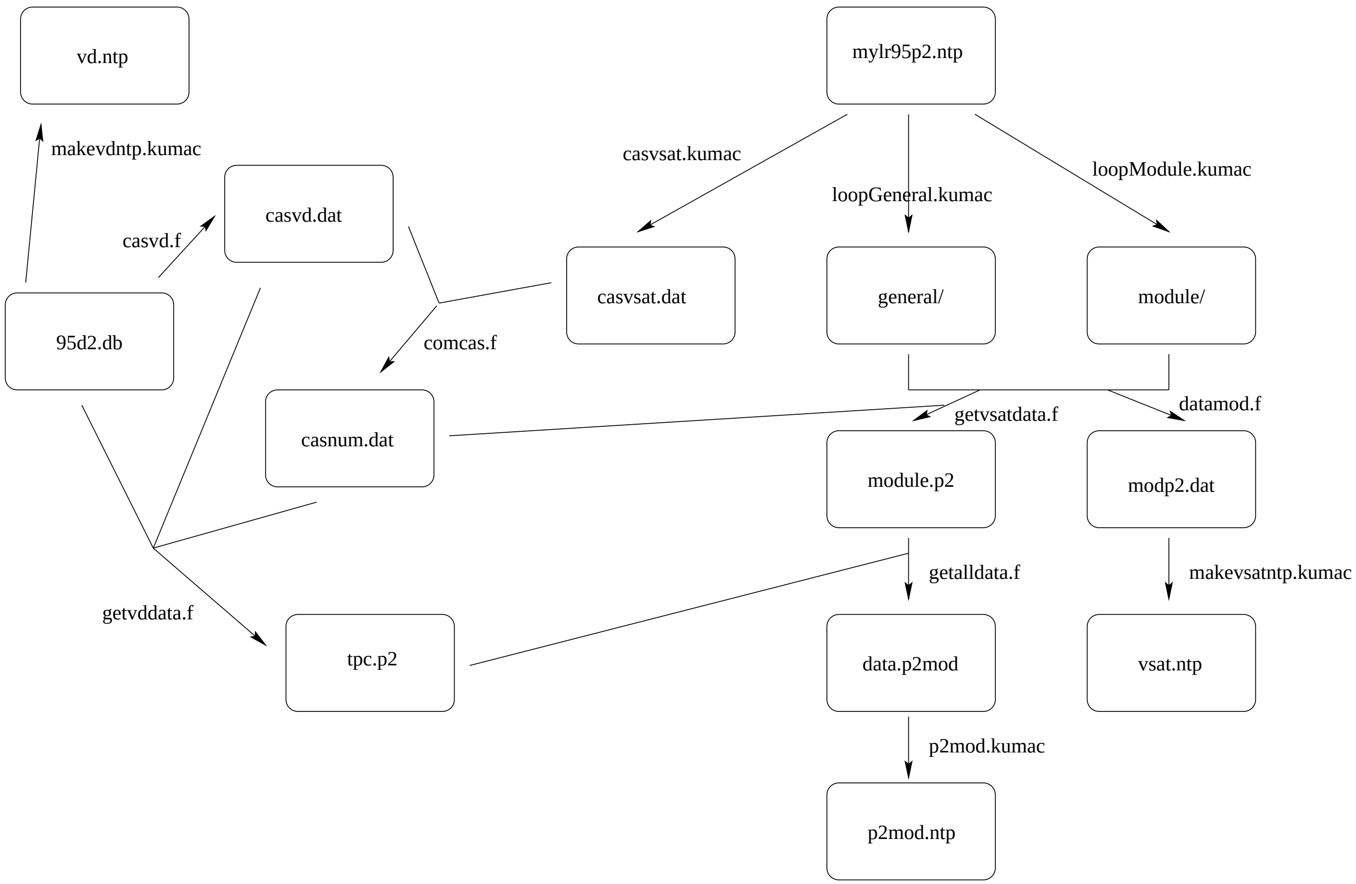

Figure 1.1: *Beam Parameter Data Processing.*

Part II

Reference Manual

2.1 Kumac makevdntp

```
* ***********************************************
*
* File : ~/public/beamp/makevdntp.kumac
* Created : 20-8-97  Christina Jarlskog
* Purpose: creates ntuple with vd/tpc data
* Input : 95d2.db
* Output : vd.ntp
*
* ***********************************************

 macro makevdntp

   ntu/create 95 vd.ntp 7 ! ! x dx y dy z dz fill
   ntu/read 95 95d2.db (9x,f8.4,e10.3,f8.4,e10.3,f8.2,f6.2,35x,i4)
   hi/file 2 vd.ntp ! n
   hrout *
   close 2

 return
```

2.2 Program casvd.f

```
************************************************
*
* File : ~/public/beamp/casvd.f
* Created : 20-8-97  Christina Jarlskog
* Purpose : extracts vd/tpc cassette numbers
* Input : 95d2.db
* Output : casvd.dat
*
************************************************

      DIMENSION K(3390)
      DIMENSION KK(3390)
      OPEN(1, STATUS='OLD', FORM='FORMATTED',
     1     FILE='95d2.db')
      OPEN(2, STATUS='NEW', FILE='casvd.dat')
      DO 10 I=1, 3380
          READ(1, 12) K(I), KK(I)
          WRITE(2, 21) K(I)*100+KK(I)
   10 CONTINUE
   12 FORMAT(1X, I5, 1X, I2)

   21 FORMAT(1X,I7)
```

```
      STOP
      END
```

2.3 Kumac casvsat

```
* ****************************************
*
* File : ~/public/beamp/casvsat.kumac
* Created : 20-8-97   Christina Jarlskog
* Purpose: extracts vsat cassette numbers
* Input : mylr95p2.ntp
* Output: casvsat.dat
*
* ****************************************

   macro casvsat

close 2

hi/file 2 mylr95p2.ntp
nt/loop 70 copy.f(run)
vec/print X
vec/write X casvsat.dat i8
vec/del X

  return
```

2.4 Program comcas.f

```
*****************************************************
*
* File : ~/public/beamp/comcas.f
* Created : 20-8-97  Christina Jarlskog
* Purpose : finds common cassette numbers for vsat/vd
* Input : casvd.dat, casvsat.dat
* Output : casnum.dat
*
*****************************************************

      DIMENSION KV(1220)
      DIMENSION KT(3390)
      OPEN(1, STATUS='OLD', FILE='casvd.dat')
      OPEN(2, STATUS='OLD', FILE='casvsat.dat')
      OPEN(3, STATUS='NEW', FILE='casnum.dat')
      DO 20 I=1, 3380
      READ(1, 51) KT(I)
```

```
        REWIND (2)
          DO 10 J=1, 1214
          READ(2, 51) KV(J)
            IF (KT(I).EQ.KV(J)) GOTO 40
            IF (KT(I).LT.KV(J)) GOTO 20
   10     CONTINUE
   20 CONTINUE
C
      IF (KT(I).NE.KV(J)) GOTO 60
   40 WRITE(3, 51) KT(I)
      GOTO 20
   51 FORMAT(1X, I7)
   60 STOP
      END
```

2.5 Program getvddata.f

```
************************************************************
*
* File : ~/public/beamp/getvddata.f
* Created : 22-8-97  Christina Jarlskog
* Purpose : prepares vd/tpc data file with common cassettes.
*           All variables are in mm.
* Input : 95d2.db, casvd.dat, casnum.dat
* Output : tpc.p2
*
************************************************************

C--- vd-tpc variables
C
      DIMENSION CASTPC(3400)
      DIMENSION XVD(3400)
      DIMENSION YVD(3400)
      DIMENSION ZTPC(3400)
      DIMENSION DXVD(3400)
      DIMENSION DYVD(3400)
      DIMENSION DZTPC(3400)
C
C--- list of common cassettes
C
      DIMENSION CASNUM(1200)
C
C--- vd-tpc data files
C
      OPEN(17, STATUS='OLD', FILE='casvd.dat')
      OPEN(18, STATUS='OLD', FILE='95d2.db')
```

```
C
C--- common cassettes file
C
      OPEN(19, STATUS='OLD', FILE='casnum.dat')
C
C--- new file with common data
C
      OPEN(21, STATUS='NEW', FILE='tpc.p2')
C
C--- extract common vd-tpc data
C
          DO 50 L=1, 3380
          READ(17, 91) CASTPC(L)
          READ(18, 93) XVD(L), DXVD(L), YVD(L), DYVD(L),
     & ZTPC(L), DZTPC(L)
                  REWIND (19)
                DO 40 K=1, 1178
                  READ(19,91) CASNUM(K)
            IF (CASNUM(K).EQ.CASTPC(L)) GOTO 55
            IF (CASNUM(K).GT.CASTPC(L)) GOTO 50
   40           CONTINUE
   50 CONTINUE
      IF (CASNUM(K).NE.CASTPC(L)) GOTO 56
   55 WRITE(21, 96) K, CASTPC(L), XVD(L), DXVD(L), YVD(L), DYVD(L),
     & ZTPC(L), DZTPC(L)
      GOTO 50
C
 56   CONTINUE
C
   91 FORMAT(1X, I7)
   93 FORMAT(9X, F8.4, E10.3, F8.4, E10.3, F8.2, F6.2)
   96 FORMAT(1X, I4, 1X, I7, F8.4, E10.3, F8.4, E10.3, F8.2, F6.2)
  100 STOP
      END
```

2.6 Kumac loopGeneral

```
* *********************************************
*
* File : ~/public/beamp/loopGeneral.kumac
* Created : 21-8-97 Christina Jarlskog
* Purpose : creates one text file per variable
* of the summary ntuple
* Input : mylr95p2.ntp, copy.f
* Output : general/
*
```

```
* ********************************************

 macro loopGeneral

 close 1
 hi/file 1 mylr95p2.ntp

* -----------------------------
*       GENERAL VARIABLES
* -----------------------------

* fill
nt/loop 70 copy.f(fill)
vec/write X general/fill.p2 f20.10
vec/del X
wait

* LEP energy
nt/loop 70 copy.f(energy)
vec/write X general/energy.p2 f20.10
vec/del X
wait

* day
nt/loop 70 copy.f(day)
vec/write X general/day.p2 f20.10
vec/del X
wait

* month
nt/loop 70 copy.f(month)
vec/write X general/month.p2 f20.10
vec/del X
wait

* run
nt/loop 70 copy.f(run)
vec/write X general/run.p2 f20.10
vec/del X
wait

 return
```

2.7 Kumac loopModule

```
* ********************************************
```

```
*
* File : ~/public/beamp/loopModule.kumac
* Created : 21-8-97 Christina Jarlskog
* Purpose : creates one text file per variable
*           of the summary ntuple and calculates
*           beam parameters per run (per module).
*           Calculations are done in mm.
*           thetax is in mrad, thetay is in rad.
* Input : mylr95p2.ntp, copy.f
* Output : module/
*
* *********************************************

 macro loopModule

 close 1
 hi/file 1 mylr95p2.ntp

* -------------------------------------------
*  A. EFFECTIVE LENGTHS AND FACTORS
* -------------------------------------------

 vec/crea q(1214)
 appl sigma
 lx=q+12600
 dlx=q+20
 ly=q+3500
 dly=q+20
 df=q+0.2
 dfx=q+0.1
 fy=q+0.1
 dfy=q+0.02
 exit
 vec/del q

* -----------------------------------
*  A1. copy A-vectors to files
* -----------------------------------

 vec/write  lx   module/lx.p2   f20.10
 vec/write dlx   module/dlx.p2  f20.10
 vec/write  ly   module/ly.p2   f20.10
 vec/write dly   module/dly.p2  f20.10
 vec/write  df   module/df.p2   f20.10
 vec/write  dfx  module/dfx.p2  f20.10
 vec/write  fy   module/fy.p2   f20.10
 vec/write dfy   module/dfy.p2  f20.10
```

```
 wait

* ---------------------------------
*  B. NTUPLE VARIABLES PER MODULE
* ---------------------------------

* sx1
 nt/loop 70 copy.f(tvsx1)
 sigma sx1=10*X

* sx2
 nt/loop 70 copy.f(tvsx2)
 sigma sx2=10*X

* dsx1
 nt/loop 70 copy.f(tvdsx1)
 sigma dsx1=10*X

* dsx2
 nt/loop 70 copy.f(tvdsx2)
 sigma dsx2=10*X

* sy1
 nt/loop 70 copy.f(tvsy1)
 sigma sy1=10*X

* sy2
 nt/loop 70 copy.f(tvsy2)
 sigma sy2=10*X

* dsy1
 nt/loop 70 copy.f(tvdsy1)
 sigma dsy1=10*X
 wait

* dsy2
 nt/loop 70 copy.f(tvdsy2)
 sigma dsy2=10*X

* nb1
 nt/loop 70 copy.f(tvnb1)
 sigma nb1=X

* nb2
 nt/loop 70 copy.f(tvnb2)
 sigma nb2=X
 wait
```

```
* dx1
 nt/loop 70 copy.f(tvdx1)
 sigma dx1=10*X

* dx2
 nt/loop 70 copy.f(tvdx2)
 sigma dx2=10*X

* ddx1
 nt/loop 70 copy.f(tvddx1)
 sigma ddx1=10*X

* ddx2
 nt/loop 70 copy.f(tvddx2)
 sigma ddx2=10*X

* dy1
 nt/loop 70 copy.f(tvdy1)
 sigma dy1=10*X

* dy2
 nt/loop 70 copy.f(tvdy2)
 sigma dy2=10*X

* ddy1
 nt/loop 70 copy.f(tvddy1)
 sigma ddy1=10*X
 wait

* ddy2
 nt/loop 70 copy.f(tvddy2)
 sigma ddy2=10*X

* f factor from fastsim, f=2*fx
 sigma f=4.36-0.35*((ddx1+ddx2)/2)
 sigma fx=f/2

* corrected dx1, dx2
 sigma dx1c=dx1*(1+0.1*(ddx1-2.6))-0.1*(dy1+7)
 sigma dx2c=dx2*(1+0.1*(ddx2-2.8))-0.05*(dy2+8)
 sigma dx1o=dx1
 sigma dx2o=dx2
 vec/del dx1 ; vec/del dx2
 sigma dx1=dx1c
 sigma dx2=dx2c
 vec/del dx1c ; vec/del dx2c
```

```
* errors for corrected dx1, dx2

 sigma a=(1+0.1*(ddx1-2.6))*ddx1/sqrt(nb1)
 sigma b=0.04*dx1*(ddx1-2.6)
 sigma c=0.1*ddy1/sqrt(nb1)
 sigma d=0.04*(dy1+7)
 sigma edx1=sqrt(a*a+b*b+c*c+d*d)
 vec/del a ; vec/del b
 vec/del c ; vec/del d

 sigma a=(1+0.1*(ddx2-2.8))*ddx2/sqrt(nb2)
 sigma b=0.04*dx2*(ddx2-2.8)
 sigma c=0.05*ddy2/sqrt(nb2)
 sigma d=0.02*(dy2+8)
 sigma edx2=sqrt(a*a+b*b+c*c+d*d)
 vec/del a ; vec/del b
 vec/del c ; vec/del d
 wait

* -----------------------------------
*  B1. copy B-vectors to files
* -----------------------------------

 vec/write sx1 module/sx1.p2 f20.10
 vec/write sx2 module/sx2.p2 f20.10
 vec/write dsx1 module/dsx1.p2 f20.10
 vec/write dsx2 module/dsx2.p2 f20.10
 vec/write sy1 module/sy1.p2 f20.10
 vec/write sy2 module/sy2.p2 f20.10
 vec/write dsy1 module/dsy1.p2 f20.10
 vec/write dsy2 module/dsy2.p2 f20.10
 vec/write nb1 module/nb1.p2 f20.10
 vec/write nb2 module/nb2.p2 f20.10
 vec/write dx1 module/dx1.p2 f20.10
 vec/write dx2 module/dx2.p2 f20.10
 vec/write dx1o module/dx1o.p2 f20.10
 vec/write dx2o module/dx2o.p2 f20.10
 vec/write edx1 module/edx1.p2 f20.10
 vec/write edx2 module/edx2.p2 f20.10
 vec/write ddx1 module/ddx1.p2 f20.10
 vec/write ddx2 module/ddx2.p2 f20.10
 vec/write dy1 module/dy1.p2 f20.10
 vec/write dy2 module/dy2.p2 f20.10
 vec/write ddy1 module/ddy1.p2 f20.10
 vec/write ddy2 module/ddy2.p2 f20.10
 vec/write f module/f.p2 f20.10
```

```
 vec/write fx module/fx.p2 f20.10
 wait

* ---------------------------------
*  C. BEAM PARAMETERS PER MODULE
* ---------------------------------

 appl sigma

 ddx1=edx1
 ddx2=edx2
 dx=(dx1+dx2)/2
 xvsat=dx/f
 a=ddx1*ddx1/nb1+ddx2*ddx2/nb2
 aa=(dx1+dx2)*df/(2*f)
 dxvsat=(sqrt(a+aa*aa))/(2*f)

 yvsat=(dy1+dy2)/(4*fy)
 qq=ddy1*ddy1/nb1+ddy2*ddy2/nb2+(dy1+dy2)*(dy1+dy2)*dfy*dfy/(16*fy*fy)
 dyvsat=sqrt(qq)/(4*fy)

 denom=(sx1+sx2)*fx
 zvsat=((dx2-dx1)*lx)/denom
 b=lx*ddx2/(denom*sqrt(nb2))
 bb=lx*ddx1/(denom*sqrt(nb1))
 bbb=zvsat*dlx/lx
 c=zvsat*dfx/fx
 dsx=sqrt(dsx1*dsx1/nb1+dsx2*dsx2/nb2)
 cc=zvsat*dsx/(sx1+sx2)
 dzvsat=sqrt(b*b+bb*bb+bbb*bbb+c*c+cc*cc)

 ad=(nb1-nb2)/(nb1+nb2)

 thetax=1.75*ad

 thetay=(sy1-sy2)/(4*ly)

 exit

 vec/del a ; vec/del aa
 vec/del denom ; vec/del qq
 vec/del b ; vec/del bb ; vec/del bbb
 vec/del c ; vec/del cc ; vec/del dsx
 wait

* ---------------------------------
*  C1. copy C-vectors to files
```

```
* ---------------------------------

 vec/write xvsat  module/xvsat.p2   f20.10
 vec/write dxvsat module/dxvsat.p2  f20.10
 vec/write yvsat  module/yvsat.p2   f20.10
 vec/write dyvsat module/dyvsat.p2  f20.10
 vec/write zvsat  module/zvsat.p2   f20.10
 vec/write dzvsat module/dzvsat.p2  f20.10
 vec/write ad     module/ad.p2      f20.10
 vec/write thetax module/thetax.p2  f20.10
 vec/write thetay module/thetay.p2  f20.10

 return
```

2.8 Program datamod.f

```
*********************************************************
*
* File : ~/public/beamp/datamod.f
* Created : 21-8-97  Christina Jarlskog
* Purpose : creates a file with all p2 vsat data per run.
*           Variables are in mm.
* Input : general/, module/
* Output : modp2.dat
*
*********************************************************
C
C--- DECLARE VARIABLES
C
      DIMENSION RUN(1220)
      DIMENSION ENERGY(1220)
      DIMENSION FILL(1220)
      DIMENSION DAY(1220)
      DIMENSION MONTH(1220)
C
      DIMENSION DX1(1220)
      DIMENSION DX10(1220)
      DIMENSION DDX1(1220)
      DIMENSION EDX1(1220)
      DIMENSION DX2(1220)
      DIMENSION DX20(1220)
      DIMENSION DDX2(1220)
      DIMENSION EDX2(1220)
      DIMENSION B1(1220)
      DIMENSION B2(1220)
      DIMENSION DY1(1220)
```

```
      DIMENSION DY2(1220)
      DIMENSION DDY1(1220)
      DIMENSION DDY2(1220)
C
      DIMENSION LX(1220)
      DIMENSION DLX(1220)
      DIMENSION LY(1220)
      DIMENSION DLY(1220)
      DIMENSION F(1220)
      DIMENSION DF(1220)
      DIMENSION FY(1220)
      DIMENSION DFY(1220)
C
      DIMENSION AD(1220)
      DIMENSION THETAX(1220)
      DIMENSION THETAY(1220)
C
      DIMENSION X(1220)
      DIMENSION DX(1220)
      DIMENSION Y(1220)
      DIMENSION DY(1220)
      DIMENSION Z(1220)
      DIMENSION DZ(1220)
C
C--- OPEN DATA FILES
C
      OPEN(1, STATUS='OLD', FILE='general/run.p2')
      OPEN(2, STATUS='OLD', FILE='general/energy.p2')
      OPEN(3, STATUS='OLD', FILE='general/fill.p2')
      OPEN(4, STATUS='OLD', FILE='general/day.p2')
      OPEN(5, STATUS='OLD', FILE='general/month.p2')
      OPEN(6, STATUS='OLD', FILE='module/dx1.p2')
      OPEN(60, STATUS='OLD', FILE='module/dx1o.p2')
      OPEN(7, STATUS='OLD', FILE='module/ddx1.p2')
      OPEN(70, STATUS='OLD', FILE='module/edx1.p2')
      OPEN(8, STATUS='OLD', FILE='module/dx2.p2')
      OPEN(80, STATUS='OLD', FILE='module/dx2o.p2')
      OPEN(9, STATUS='OLD', FILE='module/ddx2.p2')
      OPEN(90, STATUS='OLD', FILE='module/edx2.p2')
      OPEN(10, STATUS='OLD', FILE='module/nb1.p2')
      OPEN(11, STATUS='OLD', FILE='module/nb2.p2')
      OPEN(12, STATUS='OLD', FILE='module/dy1.p2')
      OPEN(13, STATUS='OLD', FILE='module/dy2.p2')
      OPEN(14, STATUS='OLD', FILE='module/ddy1.p2')
      OPEN(15, STATUS='OLD', FILE='module/ddy2.p2')
      OPEN(16, STATUS='OLD', FILE='module/lx.p2')
      OPEN(17, STATUS='OLD', FILE='module/dlx.p2')
```

```
      OPEN(18, STATUS='OLD', FILE='module/ly.p2')
      OPEN(19, STATUS='OLD', FILE='module/dly.p2')
      OPEN(20, STATUS='OLD', FILE='module/f.p2')
      OPEN(21, STATUS='OLD', FILE='module/df.p2')
      OPEN(22, STATUS='OLD', FILE='module/fy.p2')
      OPEN(23, STATUS='OLD', FILE='module/dfy.p2')
      OPEN(24, STATUS='OLD', FILE='module/ad.p2')
      OPEN(25, STATUS='OLD', FILE='module/thetax.p2')
      OPEN(26, STATUS='OLD', FILE='module/thetay.p2')
      OPEN(27, STATUS='OLD', FILE='module/xvsat.p2')
      OPEN(28, STATUS='OLD', FILE='module/dxvsat.p2')
      OPEN(29, STATUS='OLD', FILE='module/yvsat.p2')
      OPEN(30, STATUS='OLD', FILE='module/dyvsat.p2')
      OPEN(31, STATUS='OLD', FILE='module/zvsat.p2')
      OPEN(32, STATUS='OLD', FILE='module/dzvsat.p2')

      OPEN(33, STATUS='NEW', FILE='modp2.dat')
C
C---  READ DATA
C
      DO 4 I=1, 1214
          READ(1, 92)  RUN(I)
          READ(2, 92)  ENERGY(I)
          READ(3, 92)  FILL(I)
          READ(4, 92)  DAY(I)
          READ(5, 92)  MONTH(I)
          READ(6, 92)  DX1(I)
          READ(60, 92) DX10(I)
          READ(7, 92)  DDX1(I)
          READ(70, 92) EDX1(I)
          READ(8, 92)  DX2(I)
          READ(80, 92) DX20(I)
          READ(9, 92)  DDX2(I)
          READ(90, 92) EDX2(I)
          READ(10, 92) B1(I)
          READ(11, 92) B2(I)
          READ(12, 92) DY1(I)
          READ(13, 92) DY2(I)
          READ(14, 92) DDY1(I)
          READ(15, 92) DDY2(I)
          READ(16, 92) LX(I)
          READ(17, 92) DLX(I)
          READ(18, 92) LY(I)
          READ(19, 92) DLY(I)
          READ(20, 92) F(I)
          READ(21, 92) DF(I)
          READ(22, 92) FY(I)
```

```
          READ(23, 92) DFY(I)
          READ(24, 92) AD(I)
          READ(25, 92) THETAX(I)
          READ(26, 92) THETAY(I)
          READ(27, 92) X(I)
          READ(28, 92) DX(I)
          READ(29, 92) Y(I)
          READ(30, 92) DY(I)
          READ(31, 92) Z(I)
          READ(32, 92) DZ(I)
 4      CONTINUE
C
C---    CREATE DATA FILE WITH ALL VARIABLES
C
        DO 6 K=1, 1214
           WRITE(33, 99) K, RUN(K), ENERGY(K), FILL(K), DAY(K),
     &     MONTH(K), DX1(K), DX10(K), DDX1(K),EDX1(K), DX2(K),
     &     DX20(K),DDX2(K),EDX2(K),B1(K),B2(K),
     &     DY1(K), DY2(K), DDY1(K), DDY2(K),
     &     LX(K), DLX(K), LY(K),
     &     DLY(K), F(K), DF(K), FY(K), DFY(K),
     &     AD(K), THETAX(K), THETAY(K),
     &     X(K), DX(K), Y(K), DY(K),Z(K), DZ(K)
   6       CONTINUE
C
C
C
   92 FORMAT(1X, F19.10)
 99   FORMAT(1X, I4, 36(1X, F19.10))
C
      STOP
      END
```

2.9 Kumac makevsatntp

```
* ***********************************************
*
* File : ~/public/beamp/makevsatntp.kumac
* Created : 25-8-97  Christina Jarlskog
* Purpose: creates ntuple with all p2 vsat data
* Input : modp2.dat
* Output : vsat.ntp
*
* ***********************************************

 macro makevsatntp
```

```
   ntu/create 90 vsat.ntp 37 ! ! nevnt run en fill day month dx1
dx1o ddx1 edx1 dx2 dx2o ddx2 edx2 b1 b2 dy1 dy2 ddy1 ddy2 lx dlx
ly dly f df fy dfy ad thetax thetay x dx y dy z dz

   ntu/read 90 modp2.dat
   hi/file 2 vsat.ntp ! n
   hrout *
   close 2

 return
```

2.10 Program getvsatdata.f

```
************************************************************
*
* File : ~/public/beamp/getvsatdata.f
* Created : 21-8-97  Christina Jarlskog
* Purpose : prepares vsat data file with common runs.
*            All variables are in mm.
* Input : general/, module/, casnum.dat
* Output : module.p2
*
************************************************************
C
C--- vsat variables
C
      DIMENSION CASVSAT(1220)
      DIMENSION ENERGY(1220)
      DIMENSION FILL(1220)
      DIMENSION DAY(1220)
      DIMENSION MONTH(1220)
C
      DIMENSION DX1(1220)
      DIMENSION DDX1(1220)
      DIMENSION EDX1(1220)
      DIMENSION DX2(1220)
      DIMENSION DDX2(1220)
      DIMENSION EDX2(1220)
      DIMENSION B1(1220)
      DIMENSION B2(1220)
      DIMENSION DY1(1220)
      DIMENSION DY2(1220)
      DIMENSION DDY1(1220)
      DIMENSION DDY2(1220)
```

```
C
      DIMENSION LX(1220)
      DIMENSION DLX(1220)
      DIMENSION LY(1220)
      DIMENSION DLY(1220)
      DIMENSION F(1220)
      DIMENSION DF(1220)
      DIMENSION FY(1220)
      DIMENSION DFY(1220)
C
      DIMENSION AD(1220)
      DIMENSION THETAX(1220)
      DIMENSION THETAY(1220)
C
      DIMENSION X(1220)
      DIMENSION DX(1220)
      DIMENSION Y(1220)
      DIMENSION DY(1220)
      DIMENSION Z(1220)
      DIMENSION DZ(1220)
C
C--- list of common cassettes
C
      DIMENSION CASNUM(1200)
C
C--- vsat data files
C
      OPEN(1, STATUS='OLD', FILE='general/run.p2')
      OPEN(2, STATUS='OLD', FILE='general/energy.p2')
      OPEN(3, STATUS='OLD', FILE='general/fill.p2')
      OPEN(4, STATUS='OLD', FILE='general/day.p2')
      OPEN(5, STATUS='OLD', FILE='general/month.p2')
      OPEN(6, STATUS='OLD', FILE='module/dx1.p2')
      OPEN(7, STATUS='OLD', FILE='module/ddx1.p2')
      OPEN(70, STATUS='OLD', FILE='module/edx1.p2')
      OPEN(8, STATUS='OLD', FILE='module/dx2.p2')
      OPEN(9, STATUS='OLD', FILE='module/ddx2.p2')
      OPEN(90, STATUS='OLD', FILE='module/edx2.p2')
      OPEN(10, STATUS='OLD', FILE='module/nb1.p2')
      OPEN(11, STATUS='OLD', FILE='module/nb2.p2')
      OPEN(12, STATUS='OLD', FILE='module/dy1.p2')
      OPEN(13, STATUS='OLD', FILE='module/dy2.p2')
      OPEN(14, STATUS='OLD', FILE='module/ddy1.p2')
      OPEN(15, STATUS='OLD', FILE='module/ddy2.p2')
      OPEN(16, STATUS='OLD', FILE='module/lx.p2')
      OPEN(17, STATUS='OLD', FILE='module/dlx.p2')
      OPEN(18, STATUS='OLD', FILE='module/ly.p2')
```

```
      OPEN(19, STATUS='OLD', FILE='module/dly.p2')
      OPEN(20, STATUS='OLD', FILE='module/f.p2')
      OPEN(21, STATUS='OLD', FILE='module/df.p2')
      OPEN(22, STATUS='OLD', FILE='module/fy.p2')
      OPEN(23, STATUS='OLD', FILE='module/dfy.p2')
      OPEN(24, STATUS='OLD', FILE='module/ad.p2')
      OPEN(25, STATUS='OLD', FILE='module/thetax.p2')
      OPEN(26, STATUS='OLD', FILE='module/thetay.p2')
      OPEN(27, STATUS='OLD', FILE='module/xvsat.p2')
      OPEN(28, STATUS='OLD', FILE='module/dxvsat.p2')
      OPEN(29, STATUS='OLD', FILE='module/yvsat.p2')
      OPEN(30, STATUS='OLD', FILE='module/dyvsat.p2')
      OPEN(31, STATUS='OLD', FILE='module/zvsat.p2')
      OPEN(32, STATUS='OLD', FILE='module/dzvsat.p2')
C
C--- common cassettes file
C
      OPEN(33, STATUS='OLD', FILE='casnum.dat')
C
C--- new file with common run number data
C
      OPEN(34, STATUS='NEW', FILE='module.p2')
C
C--- read vsat data
C
      DO 4 I=1, 1214
          READ(1, 92)  CASVSAT(I)
          READ(2, 92)  ENERGY(I)
          READ(3, 92)  FILL(I)
          READ(4, 92)  DAY(I)
          READ(5, 92)  MONTH(I)
          READ(6, 92)  DX1(I)
          READ(7, 92)  DDX1(I)
          READ(70, 92) EDX1(I)
          READ(8, 92)  DX2(I)
          READ(9, 92)  DDX2(I)
          READ(90, 92) EDX2(I)
          READ(10, 92) B1(I)
          READ(11, 92) B2(I)
          READ(12, 92) DY1(I)
          READ(13, 92) DY2(I)
          READ(14, 92) DDY1(I)
          READ(15, 92) DDY2(I)
          READ(16, 92) LX(I)
          READ(17, 92) DLX(I)
          READ(18, 92) LY(I)
          READ(19, 92) DLY(I)
```

```
          READ(20, 92) F(I)
          READ(21, 92) DF(I)
          READ(22, 92) FY(I)
          READ(23, 92) DFY(I)
          READ(24, 92) AD(I)
          READ(25, 92) THETAX(I)
          READ(26, 92) THETAY(I)
          READ(27, 92) X(I)
          READ(28, 92) DX(I)
          READ(29, 92) Y(I)
          READ(30, 92) DY(I)
          READ(31, 92) Z(I)
          READ(32, 92) DZ(I)
  4   CONTINUE
C
C--- extract useful vsat data
C
      DO 20 I=1, 1178
      READ(33, 91) CASNUM(I)
        REWIND (1)
          DO 10 J=1, 1214
          READ(1, 92) CASVSAT(J)
            IF (CASNUM(I).EQ.CASVSAT(J)) GOTO 40
            IF (CASNUM(I).LT.CASVSAT(J)) GOTO 20
   10     CONTINUE
   20 CONTINUE
      IF (CASNUM(I).NE.CASVSAT(J)) GOTO 41
 40   WRITE(34, 95) I, CASVSAT(J), ENERGY(J), FILL(J), DAY(J), MONTH(J),
     &  DX1(J), DDX1(J), EDX1(J),DX2(J),
     &  DDX2(J), EDX2(J),B1(J), B2(J), DY1(J), DY2(J), DDY1(J), DDY2(J),
     &     LX(J), DLX(J), LY(J), DLY(J), F(J), DF(J), FY(J), DFY(J),
     &     AD(J), THETAX(J), THETAY(J), X(J),
     &   DX(J), Y(J), DY(J), Z(J), DZ(J)
      GOTO 20
C
 41   CONTINUE
C
   91 FORMAT(1X, I7)
   92 FORMAT(1X, F19.10)
   95 FORMAT(1X, I4, 1X, I7, 33(1X, F20.10))
  100 STOP
      END
```

2.11 Kumac means

```
*************************************************************
```

```
*
* File : ~/beam95/p2/means.kumac
* Created : 25-8-97  Christina Jarlskog
* Purpose : plots x and z for VSAT, VD/TPC
* Input : mylr95p2.ntp, vd.ntp
* Output : -
*
**************************************************************

 macro means

   close 1
   hi/file 1 mylr95p2.ntp
   opt stat

* x_int
   ntu/pl 70.((0.5*((10*tvdx1*(1+0.1*(10*tvddx1-2.6))-0.1*(10*tvdy1+7))+
(10*tvdx2*(1+0.1*(10*tvddx2-2.8))-0.05*(10*tvdy2+8))))/(4.36-0.35*((10*
tvddx1+10*tvddx2)/2)))
   wait

* z_int
   ntu/pl 70.((((10*tvdx2*(1+0.1*(10*tvddx2-2.8))-0.05*(10*tvdy2+8))-
(10*tvdx1*(1+0.1*(10*tvddx1-2.6))-0.1*(10*tvdy1+7))))/((4.36-0.35*
((10*tvddx1+10*tvddx2)/2))*((10*tvsx1+10*tvsx2)/(2*12600))))
   wait

   close 1
   hi/file 1 vd.ntp

* x_VD
   ntu/pl 95.x
   wait

* z_VD
   ntu/pl 95.z

 return
```

2.12 Program getalldata.f

```
**************************************************************
*
* File : ~/public/beamp/getalldata.f
* Created : 22-8-97  Christina Jarlskog
* Purpose : a. calculates the acollinearity (in mrad) and
```

```
*           the normalized differences in x and z beamspot
*           b. prepares vd/vsat data file with common runs.
*           All variables are in mm.
* Input : tpc.p2, module.p2
* Output : data.p2mod
*
*************************************************************
C
C--- vsat variables
C
      DIMENSION CASVSAT(1200)
      DIMENSION ENERGY(1200)
      DIMENSION FILL(1200)
      DIMENSION DAY(1200)
      DIMENSION MONTH(1200)
C
      DIMENSION DX1(1200)
      DIMENSION DDX1(1200)
      DIMENSION EDX1(1200)
      DIMENSION DX2(1200)
      DIMENSION DDX2(1200)
      DIMENSION EDX2(1200)
      DIMENSION B1(1200)
      DIMENSION B2(1200)
      DIMENSION DY1(1200)
      DIMENSION DY2(1200)
      DIMENSION DDY1(1200)
      DIMENSION DDY2(1200)
C
      DIMENSION LX(1200)
      DIMENSION DLX(1200)
      DIMENSION LY(1200)
      DIMENSION DLY(1200)
      DIMENSION F(1200)
      DIMENSION DF(1200)
      DIMENSION FY(1200)
      DIMENSION DFY(1200)
C
      DIMENSION AD(1200)
      DIMENSION THETAX(1200)
      DIMENSION THETAY(1200)
C
      DIMENSION X(1200)
      DIMENSION DX(1200)
      DIMENSION Y(1200)
      DIMENSION DY(1200)
      DIMENSION Z(1200)
```

```
      DIMENSION DZ(1200)
C
C--- vd/tpc variables
C
      DIMENSION XVD(1200)
      DIMENSION YVD(1200)
      DIMENSION ZTPC(1200)
      DIMENSION DXVD(1200)
      DIMENSION DYVD(1200)
      DIMENSION DZTPC(1200)
C
C--- acollinearity and normalized differences
C
      DIMENSION EX(1200)
      DIMENSION EY(1200)
      DIMENSION XNORM(1200)
      DIMENSION ZNORM(1200)
C
C--- vsat-vd/tpc data files
C
      OPEN(1, STATUS='OLD', FILE='module.p2')
      OPEN(2, STATUS='OLD', FILE='tpc.p2')
C
C--- new file with common data
C
      OPEN(3, STATUS='NEW', FILE='data.p2mod')
C
C--- merge vsat and vd/tpc data
C
      DO 60 KK=1, 1178
           READ(1,90) CASVSAT(KK), ENERGY(KK), FILL(KK), DAY(KK),
     &  MONTH(KK), DX1(KK), DDX1(KK), EDX1(KK), DX2(KK),
     &  DDX2(KK),EDX2(KK), B1(KK), B2(KK), DY1(KK), DY2(KK), DDY1(KK),
     &  DDY2(KK),LX(KK), DLX(KK), LY(KK), DLY(KK), F(KK), DF(KK),
     &  FY(KK),DFY(KK),AD(KK), THETAX(KK), THETAY(KK), X(KK),
     &  DX(KK), Y(KK),DY(KK), Z(KK), DZ(KK)
           READ(2,91) XVD(KK), DXVD(KK), YVD(KK),
     & DYVD(KK), ZTPC(KK), DZTPC(KK)
C
C--- calculate acollinearity and normalized differences
C
       EX(KK)=((DX1(KK)+DX2(KK))/2-F(KK)*XVD(KK))/12.6
       EY(KK)=((DY1(KK)+DY2(KK))/2-0.2*YVD(KK))/3.5
       XNORM(KK)=(X(KK)-0.1612-XVD(KK)-3.203)/SQRT(DX(KK)**2+
     &            DXVD(KK)**2)
       ZNORM(KK)=(Z(KK)+47.65-ZTPC(KK)-8.845)/SQRT(DZ(KK)**2+
     & DZTPC(KK)**2)
```

```
C
         WRITE(3,92) KK, CASVSAT(KK),ENERGY(KK), FILL(KK), DAY(KK),
     &  MONTH(KK), DX1(KK), DDX1(KK), EDX1(KK), DX2(KK),
     &  DDX2(KK), EDX2(KK), B1(KK), B2(KK), DY1(KK), DY2(KK), DDY1(KK),
     &  DDY2(KK),LX(KK), DLX(KK), LY(KK), DLY(KK), F(KK), DF(KK),
     &  FY(KK),DFY(KK),AD(KK), THETAX(KK), THETAY(KK), X(KK),
     &  DX(KK), Y(KK),DY(KK), Z(KK), DZ(KK),
     &  EX(KK), EY(KK), XNORM(KK), ZNORM(KK), XVD(KK),
     &  DXVD(KK), YVD(KK), DYVD(KK), ZTPC(KK), DZTPC(KK)
   60 CONTINUE
C
 90   FORMAT(6X, I7, 33(1X, F20.10))
 91   FORMAT(13X, F8.4, E10.3, F8.4, E10.3, F8.2, F6.2)
 92   FORMAT(1X, I4, 1X, I7, 37(1X, F20.10), 1X, F8.4, E10.3,
     &        F8.4, E10.3, F8.2, F6.2)
  100 STOP
      END
```

2.13 Kumac p2mod

```
* ***********************************************
*
* File : ~/beam95/p2/p2mod.kumac
* Created : 22-8-97  Christina Jarlskog
* Purpose: creates ntuple with vsat/vd/tpc data
* Input : data.p2mod
* Output : p2mod.ntp
*
* ***********************************************

 macro p2mod

   ntu/create 85 p2mod.ntp 45 ! ! nevnt run en fill day month
dx1 ddx1 edx1 dx2 ddx2 edx2 b1 b2 dy1 dy2 ddy1 ddy2 lx dlx ly
dly f df fy dfy ad thetax thetay x dx y dy z dz ex ey xnorm
znorm xvd dxvd yvd dyvd ztpc dztpc

   ntu/read 85 data.p2mod
   hi/file 2 p2mod.ntp ! n
   hrout *
   close 2

 return
```

Acknowledgements

I would like to thank my supervisors, B. Lörstad and G. Rinaudo, as well as P. Jonsson, for checking the draft. Thanks are also due to U. Egede for help with PAW.

Bibliography

[1] Ch. Jarlskog, 'Interaction point estimation and beam parameter variations in DELPHI with the VSAT', LUNFD6/(NFFL-7110)/1995.