References

[1] V. A. Khoze, A.D. Martin, R. Orava, and M.G. Ryskin. *Eur. Phys. J. C* **19** (2001): 313–322.

[2] S. Eidelman et al. Review of Particle Physics. *Phys. Lett. B* **592** (2004): 31.

[3] M. Gell-Mann and Y. Neémaan. *The Eightfold Way.* New York, Benjamin, 1964.

[4] *Proceedings 3rd International Conference on Electron and Photon Interactions*, (1967).

[5] J. D. Bjorken and E.A. Paschos. *Phys. Rev.* **185** (1975 (1969)).

[6] R. P. Feynman. *Phys. Rev. Lett.* **123** (1969): 1415.

[7] G. Altarelli and G. Parisi. *Nucl. Phys.* **B126** (1977): 298.

[8] H. L. Lai et al. *Phys. Lett.* **B 304** (1993): 159.

[9] G. Arnison et al. *Phys. Lett.* **122B** (1983): 103.

[10] G. Arnison et al. *Phys. Lett.* **126B** (1983): 398.

[11] D. H. Perkins. *Introduction to High Energy Physics.* Cambridge, Cambridge University Press, 2000.

[12] S. Weinberg. Recent Progress in the Gauge Theories of the Weak, Electromanetic and Strong Interactions. *Rev. Mod. Phys.* **46** (1974): 255.

[13] S. van der Meer. Calibration of the effective beam height in the ISR. *Internal Report CERN/ISR-PO* (1968): 68–31.

[14] W. Kienzle et al. (TOTEM Collaboration). Total cross-section, elastic scattering and diffraction dissociation at the LHC. *CERN-LHCC-97* **LHCC-I-11** (1997): 49.

[15] M. Dittmar, F. Pauss, and D. Zürcher. Toward a precise parton luminosity determination at the CERN LHC. *Phys. Rev. D* **56** (1997): 7284.

[16] D. Zürcher. *A Method to Measure Parton Luminosities at the LHC: a Tool to determine Higgs Backgrounds more accurately.* Diploma thesis, 1997.

[17] V. A. Khoze, A.D. Martin, R. Orava, and M.G. Ryskin. Luminosity monitors at the LHC. *hep-ph/0010163* **1** (2000).

[18] A. D. Martin, R.G. Roberts, W.J. Stirling, and R.S. Thorne. Parton distribution and the LHC: W and Z production. *Eur. Phys. J. C* **14** (2000): 133.

[19] A. D. Martin, R.G. Roberts, W.J. Stirling, and R.S. Thorne. *Eur. Phys. J. C* **18** (2000): 117.

[20] D. J. Griffith. *Introduction to Elementary Particles.* New York, John Wiley & Son, 1987.

[21] R. W. Brown, D. Sahdev, and K. O. Mikaelian. *Phys. Rev.* **D 20** (1979): 1164.

[22] CERN 90-10. *Proceeding of LHC Workshop*, volume 153 of *Aachen, ed. by G. Jarlskog and D. Rein.* G. Altarelli, 1990.

[23] The Large Hadron Collider. The Large Hadron Collider, Conceptual Design. *CERN/AC/95-05* (1995).

[24] ATLAS Collaboration and W. W. Armstrong et al. Technical Proposal. *CERN Report No. CERN/LHCC 94-43, LHCC/P2* **15** (1994).

[25] CMS Collaboration and G. L. Bayatian et al. Technical Proposal. *CERN Report No. CERN/LHCC 94-38, LHCC/P1* **15** (1994).

[26] ALICE Collaboration. ALICE: Technical Proposal. *Internal Report CERN* (1995).

[27] LHCb Collaboration. LHCb: Technical Proposal. *Internal Report CERN* (1998).

[28] This is the cross-section for the minimum bias events (msel = 1) in PYTHIA[30].

[29] CMS Collaborations. CMS Posters, 2000.

[30] T. Sjöstrand. *Pythia 6.2 Physics and Manual*, 2002.

[31] S. Agostinelli et al. Geant4 – a simulation toolkit. *Nuclear Instruments and Methods in Physics Research* **NIM A 506** (2003): 250–303.

[32] ORCA 8.4.0 Reference Manual. *http://cmsdoc.cern.ch/swdev/snapshot/ORCA /ReferenceManual/html/ORCA.html.*

[33] René Brun et al. *ROOT: An Object-Oriented Data Analysis Framework, Users Guide.* CERN, 3.1b edition, June 2001.

[34] The ATLAS and CMS Collaboration. High Transverse Momentum Physics at the Large Hadron Collider. *CERN Internal Note* (1983).

[35] CMS Collaboration. Muon Technical Design Report. *CERN/LHCC/97-32* **15** (1997).

[36] D. D. Koelling and G. O. Arbman. Use of energy derivative of the radial solution in an augmented plane wave method: application to copper. *J. Phys. F* **5** (1975): 2041.

[37] M. Bozzo et al. TOTEM Collaboration. *Internal Report CERN/LHCC* (1993).

[38] V. M. Budnev et al. *Phys. Lett. B* **39** (1972): 526.

[39] SM LHC Workshop. *CERN 2000-004* (1999): 156–193.

[40] R. M. Barnett et al. Particle Data Group. *Phys. Rev. D* **54** (1996).

[41] OSCAR Software Group. *OSCAR_3_6_0 User's Guide*, 2002.

[42] ORCA Software Group. *ORCA_8_6_0 User's Guide*, 2004.

[43] D. Froidevaux and E. Richter-Was. *ATLAS Internal Note* **48** (1994).

[44] H. Plothow-Besch. PDFLIB Version 6.06. *CERN Program Library* **W5051** (1995).

[45] H. L. Lai et al. *Phys. Rev. D* **55** (1997): 1280.

[46] The CTEQ Meta-Page. *http://www.phys.psu.edu/ cteq/fortran/pdfs/Ctq5Par.f.*

[47] A. D. Martin, R.G. Roberts, and W.J. Stirling. *Phys. Lett.* **B 306** (1993): 147.

[48] M. Glück, E. Reya, and A. Vogt. Ph.d. thesis, 1994.

[49] M. E. Peskin and D. V. Schroeder. *An Introduction to Quantum Field Theory.* Addison-Wesley, 1995.

[50] P. W. Higgs. Broken Symmetries, Massless Particles and Gauge Fields. *Phys. Lett.* **12** (1964): 132–133.

[51] P. W. Higgs. Spontaneous Symmetry Breakdown without Massless Bosons. *Phys. Rev.* **145** (1966): 1156–1163.

[52] F. Englert and R. Brout. Broken symmetry and the mass of gauge vector meson. *Phys. Rev.* **13** (1964): 321–323.

[53] J. D. Jackson. *Classical Electrodynamics*, 3rd edition. New York, John Wiley & Son, 1998.

[54] LEP Large Electron Positron Collider Division. *Large Electron Positron Collider, LEP.* CERN, Switzerland, 2000.

[55] S. Drell and T. M. Yan. Massive Lepton-Pair Production in Hadron-Hadron Collisions at High Energies. *Phys. Rev. Lett.* **25** (1970): 316.

APPENDICES

Appendix A

Technical Terms

A.1 Rapidity and Pseudorapidity

The rapidity y is defined as:

$$y = \tanh^{-1}(\frac{p_z}{E}) = \frac{1}{2}\ln\left(\frac{E+p_z}{E-p_z}\right) \tag{A.1}$$

where E is the energy of the particle and p_z is the particle's momentum in z-direction. If the four-momentum is much greater than the particle's mass $p \gg m$ we obtain the pseudorapidity η :

$$\eta = -\ln[\tan(\theta/2)] \approx y \tag{A.2}$$

where $\cos\theta = p_z/p$, and θ is the center-of-mass scattering angle. Fig. A.1 shows the direction of the beam on xyz-axis and Fig. A.2 shows the angle θ on the xz-plane.

The rapidity y is used for data analysis in particle physics because its distribution is invariant under Lorentz boost. The term "rapidity" is originated from the "velocity" term in the expression of the modified Lorentz transformation, as will be clarify later. The modified version is preferred for doing data analysis because it is more informative than the standard Lorentz transformation form.

To gain more insight of this notation, let's consider the velocity ratio β. Since β has values between -1 and +1, we choose hyperbolic function *tanh*, which also has values between -1 and +1, to express β in term of rapidity "y."

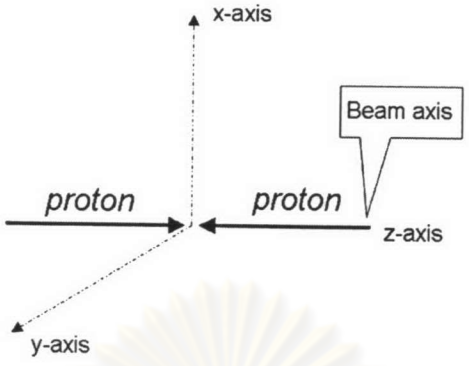

Figure A.1: A diagram showing the z-direction of the beam and the x- and y-axis.

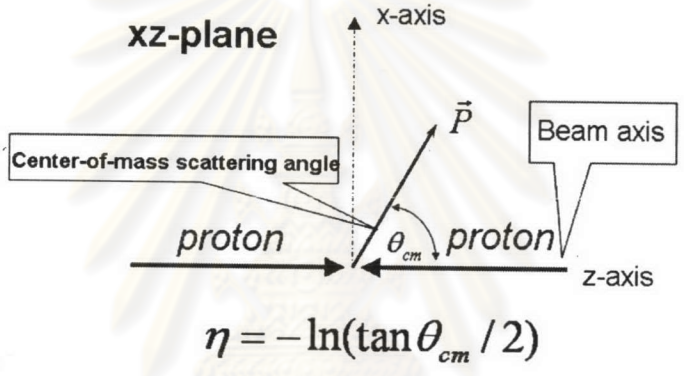

Figure A.2: A diagram showing the center-of-mass scattering angle with respect to the beam direction.

$$\tanh y \equiv \frac{v}{c} = \beta. \tag{A.3}$$

For a given velocity v, we obtain the rapidity:

$$y = \tanh^{-1}(\frac{v}{c}) = \frac{1}{2}\ln(\frac{c+v}{c-v}). \tag{A.4}$$

It is the velocity term in this expression that makes y to be called "rapidity".

In Lorentz transformation, the rapidity can be written as shown below. Instead of expressing $\gamma = \frac{1}{\sqrt{1-v^2/c^2}}$, we write γ in term of rapidity:

$$\gamma = \frac{1}{\sqrt{1-\tanh^2 y}} = \frac{1}{\sqrt{1-\frac{\sinh^2 y}{\cosh^2 y}}} = \frac{\cosh y}{\sqrt{\cosh^2 y - \sinh^2 y}} = \cosh y. \qquad \text{(A.5)}$$

Using Eq. (A.5), the Lorentz transformation can be rewritten as follow:

$$\begin{aligned} x' &= \gamma(x - vt) = \cosh y(x - ct\tanh y) = x\cosh y - (ct)\sinh y, \\ (ct') &= c\gamma(t - vx/c^2) = c\cosh y(t - x/c\tanh y) = (ct)\cosh y - x\sinh y, \\ y' &= y, \\ z' &= z. \end{aligned} \qquad \text{(A.6)}$$

For the CMS detector, a diagram illustrating the scale of rapidity is displayed in Fig. A.3.

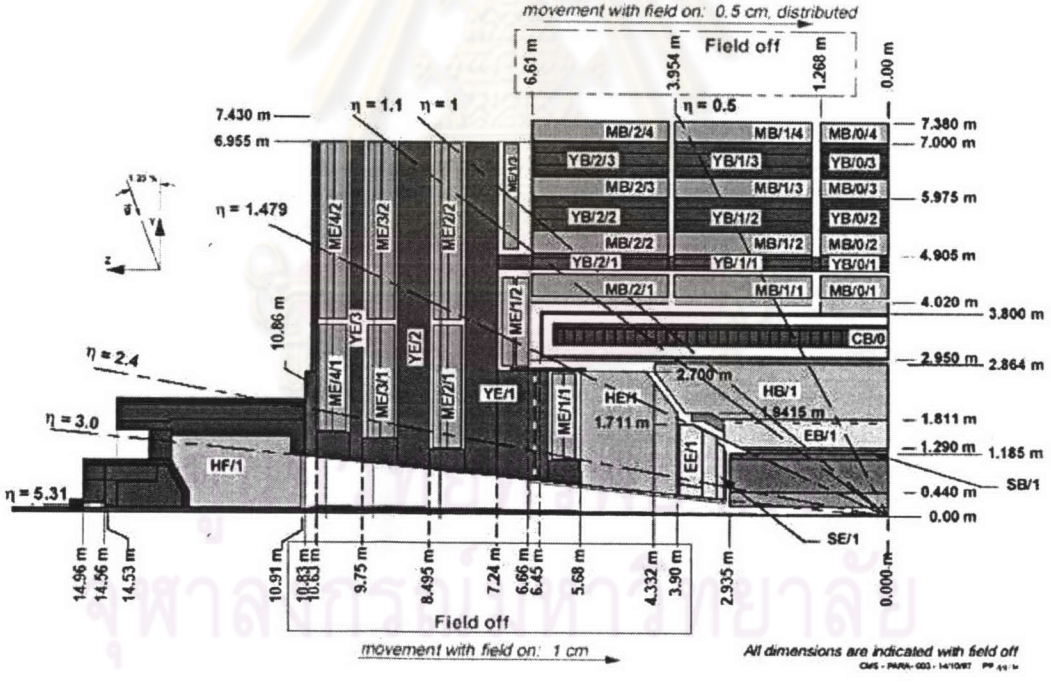

Figure A.3: The scale of rapidity of the CMS detector.

A.2 Azimuthal Angle

The azimuthal angle (ϕ) is the angle on the plane that is perpendicular to the beam line, as shown in Fig. A.4.

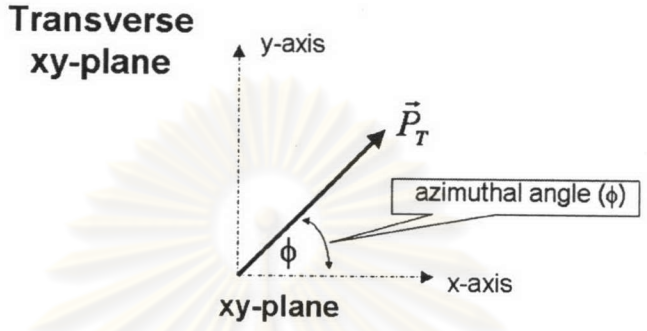

Figure A.4: A diagram showing azimuthal angle (ϕ) with respect to the beam direction (z-axis) and the transverse momentum (P_T).

A.3 Transverse Momentum

The transverse momentum (P_T) is a component of momentum on the plane perpendicular to the beam direction. It is defined by

$$P_T = \sqrt{P_x^2 + P_y^2}. \tag{A.7}$$

Fig. A.4 show the direction of P_T on the transverse xy-plane.

A.4 Drell-Yan Process

The most potential background of our signal is the Drell-Yan process [55], a process named after its two discoverers, S. Drell and T.M.Yan. The Drell-Yan process is a mechanism in proton-proton or proton-antiproton collision when one quark in one proton and one antiquark in the other annihilate into a virtual photon which then produces a lepton pair. The Feynman diagram of the mechanism is displayed in Fig. A.5.

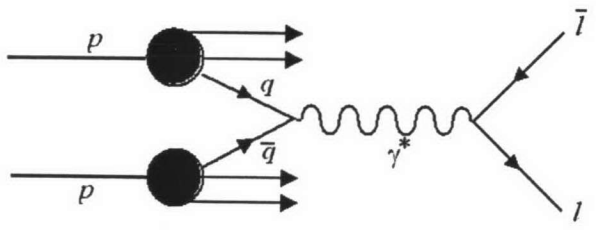

Figure A.5: The Feynman diagram of Drell-Yan process for massive lepton-pair production in *pp* collision.

Because of the close relations to the quark-antiquark pair in the hadron, the Drell-Yan is believed to be one of the most powerful tools to probe the structure of hadrons.

A.5 Minimum bias events

Minimum bias events are events those not being selected or not under any constraint. Consequently, minimum bias will involve lots of background. Minimum bias in hadronic interactions are intrinsically low-pt events; however there is some chance that they possess high-pt particles.

At the LHC luminosity, for each one signal event, there will be approximately 200 minimum bias events.

A.6 Pile up

The "pile up" effect is generated from the fact that pulse are randomly spaced in time which can lead to interfering effects between pulses when counting rates are not low enough. In other words, pile up is an incidence that the tail of a pulse is tapped by the next pulse.

A.7 HEPEVT Ntuple Format

HEPEVT is a HEP (High Energy Physics) standard to store particle kinematics information. In FORTRAN, HEPEVT is a common block as follows

```
PARAMETER (NMXHEP=4000)
      COMMON/HEPEVT/NEVHEP,NHEP,ISTHEP(NMXHEP),IDHEP(NMXHEP),
      & JMOHEP(2,NMXHEP),JDAHEP(2,NMXHEP),
      & PHEP(5,NMXHEP),VHEP(4,NMXHEP)
```

Each entry is described in the table below.

Table A.1: Entries of HEPEVT Ntuple

Entry name	Description
NEVHEP	event number
NHEP	number of entries (particles, partons)
ISTHEP	status code
IDHEP	PDG identifier
JMOHEP	position of 1st and 2nd mother
JDAHEP	position of 1st and last daughter
ISTHEP	status code
PHEP	4-momentum and mass
VHEP	vertex xyz and production time

More details of the HEPEVT entries can be found in PYTHIA manual [30].

Appendix B

CMKIN

B.1 CMKIN Main Source Code

The main source codes of CMKIN to generate physics processes datacards are the two PYTHIA program execute files: **ki_pyt_main.F** and **kine_make_ntpl_pyt.run**.

B.2 Datacard Files

For each physics process, the associated datacard files are listed in Table B.1 below.

Table B.1: The CMKIN generated physics processes and its datacard files

Processes	File name
$q\bar{q} \rightarrow W^+W^- \rightarrow \mu^{\pm}\nu_{\mu}(\bar{\nu_{\mu}})$	ww_2l.txt
$q\bar{q} \rightarrow Z^0Z^0 \rightarrow 2\mu^+2\mu^-$	zz_4l.txt
$q\bar{q} \rightarrow \gamma^*\gamma^* \rightarrow \mu^+\mu^-$	ggvir_ffbar.txt
$q\bar{q} \rightarrow \gamma\gamma \rightarrow \mu^+\mu^-$	gg_ffbar.txt
Drell-Yan Process	dy2mu.txt
minimum-bias	minbias.txt

ki_pyt_main.F

```
*
* $Id: ki_pyt_main.F,v 1.8 2004/10/11 14:47:05 karimaki Exp $
*
* $Log: ki_pyt_main.F,v $
*
#include "sys/CMS_machine.h"
#include "pilot.h"
*CMZ :           18/02/96 17.35.51 by  Lucas Taylor
*-- Author :    Lucas Taylor   14/07/95
*+DOC*************************************************************
       PROGRAM KI_PYT_MAIN
*
*DOC   Author:    Lucas Taylor, Veikko Karimaki
*DOC   Created:   14/07/95
*
*+DOC*************************************************************
#include "cmsi/cmcdes/impnone.inc"
#include "cmsi/cmcdes/chepevt.inc"
#include "cmsi/cmcdes/mc_param.inc"
#include "kine/kicdes/kikeys.inc"
#include "kine/kicdes/kicwn.inc"
*
      INTEGER I
*
      INTEGER ISEL,IFILE
      SAVE    ISEL,IFILE
*
      LOGICAL I_WANT, DUMMY
*-------------------------------------------------------------
*
*   Initialisation phase
*   --------------------
*
      CALL KI_INIT (0)  !CMKIN initialisation (data cards, etc.)
*
      CALL KI_EVT_SEL (-1,DUMMY)!KSEL-dependent selection init
*
      CALL KI_PYT_INIT       ! PYTHIA initialisation
*** tauola usage
      IF (JAKK1.GE.0)
     &CALL PRETAUOLA(-1)
*
      ISEL = 0            !Initialise selected event counter
      IFILE = 0
      IF (N_NFIL.EQ.-99) N_NFIL=N_NSEL
```

```
*
*-------------------------------------------------------------------
*
*   Event generation
*   ---------------
*
      DO I=1,N_TRIG                    ! Event generation loop
        NHEP = 0
        NVRMCP = 0
        CALL PYEVNT                    ! Generate PYTHIA event
*
** tauola usage
        IF (JAKK1.GE.0)
     &  CALL PRETAUOLA(0)

        CALL KI_EVT_SEL (0,I_WANT)     ! KSEL-dependent event
selection
*
        IF (I_WANT) THEN               ! Event was selected
          ISEL = ISEL + 1
        IF (NHEP.EQ.0)
     &    CALL KIU_FILL_HEPEVT         ! Fill HEPEVT common block
          IF (IDCWN.GT.0) THEN
            CALL KI_WR_EVT             ! Save event to ntuple file
          ELSE
            CALL KIR_ANAL
          ENDIF
*
        IF
(MOD(ISEL,10).EQ.1.OR.ISEL.EQ.N_NSEL.OR.MOD(ISEL,N_NFIL).EQ.0)
     &    WRITE(*,*)
     &    '**KIMAIN** Nb. events triggered, selected =',I,ISEL
*
          IF (ISEL.EQ.N_NSEL) GO TO 99 ! End of job
          IF (MOD(ISEL,N_NFIL).EQ.0.AND.IDCWN.GT.0) THEN
            CALL CLOSE_NTUPLE
            IFILE = IFILE + 1
            CALL RENAME_NTUPLE(IFILE)
            CALL OPEN_NEW_NTUPLE
          ENDIF
        ENDIF
      ENDDO
*
*-------------------------------------------------------------------
*   Termination phase
*   ----------------
*
 99 CONTINUE
```

```
*
      CALL KI_EVT_SEL (1,DUMMY)!KSEL-dependent selection terminat
*
      CALL PYSTAT(1)       !Write-out PYTHIA run summary
*
      IF (JAKK1.GE.0) CALL PRETAUOLA(1) ! srs insertion
      IF (IDCWN.GT.0) THEN
        CALL CLOSE_NTUPLE
        IF (ISEL.GT.N_NFIL) THEN
          IFILE = IFILE + 1
          CALL RENAME_NTUPLE(IFILE)
        ENDIF
      ELSE
        CALL KIR_END
      ENDIF
*
      CALL KI_END(0)          !CMKIN termination
*-----------------------------------------------------------------
      END                                  ! End of KI_PYT_MAIN
```

kine_make_ntpl_pyt.run

```
#!/bin/csh
#                                    Updated for cms130 30.01.03 / VK
#####################################################################
#
# CMKIN run script (to create a HEPEVT ntuple)
#
# Contact persons:  Lucas Taylor
#                   Veikko Karimäki
#
#####################################################################
#
 set GCCVER=/usr/local/gcc-alt-3.2.3
 if ( ! -d $GCCVER ) set GCCVER=/usr/local/gcc-alt-3.2
 setenv LD_LIBRARY_PATH $GCCVER/lib:/usr/local/lib
#
#  set DATACARDFILE=minbias.txt
#  set DATACARDFILE=dy2mu.txt
#  set DATACARDFILE=w_lnu.txt
#  set DATACARDFILE=zz_4l.txt
#  set DATACARDFILE=ggvir_ffbar.txt
   set DATACARDFILE=gg_ffbar.txt

#
 set npyt=6225
 set DATACARDFILE=datacards/$DATACARDFILE

if ( ! -f ${DATACARDFILE} ) then
  echo ' '
  echo The file ${DATACARDFILE} does not exist
  exit 1
endif
#
 set JOBNAME=kine_make_ntpl_pyt
 set EXEFILE=${SCRATCH}/${JOBNAME}${npyt}
#

if ( ! -f ${EXEFILE}.exe ) then
  echo ' '
  echo The file ${EXEFILE}.exe does not exist
  echo Run the script ${JOBNAME}.com first
  exit 1
endif

if (-f ${JOBNAME}.lis) mv ${JOBNAME}.lis ${JOBNAME}.lis_old
echo
echo 'Starting execution of  ' $EXEFILE.exe
echo '  using datacard file  ' $DATACARDFILE
echo on `date`
```

```
${EXEFILE}.exe >${JOBNAME}.lis <<EOF
LIST
C
C-------- Start of channel independent control cards ------------
C
C CMS energy (GeV)
C
  ECMS 14000.
C -----------
C No. of events to generate
C
  TRIG    9999999
C -------------
C Activate the 'NTPL 0' card, if no ntuple file wanted
C
C NTPL  0
C -------------
C No. of events selected (written out)
C
  NSEL    10000
C -----------
C particle masses                (not always needed)
C --------------
C
C PMAS   6,1=175.                ! top quark
C PMAS  23,1=91.187              ! Z
C PMAS  24,1=80.22               ! W
C
  MSTJ  22 = 2                   ! Decay those unstable particles
  PARJ  71 = 10.                 ! for which c*tau < 10 mm
C
C-------- End of channel independent control cards
C
C-------- Start of channel dependent control cards --------------
`cat ${DATACARDFILE}`
EOF
#
if ( $status ) then
  echo  ----------------------------------------------------
  echo $0 finished on `date` with status code 1
  echo  ----------------------------------------------------
  exit 1
else
  echo ----------------------------------------------------
  echo  $0 finished on `date`
  echo  ----------------------------------------------------
endif

#####################################################################
```

ZZ -> 2mu 2nu datacard

```
C Pythia/JETSET parameters
C ------------------------
C First set random seed
C
  MRPY 1= 563389
  KSEL 213 ! use KIS_HZZ_4L
C
  CFIL 'EVTO' 'zz_4l.ntpl '
C
C --------------
C Set RUN number
C --------------
C KRUN 0
C don't use  PDF library (would be MSTP 52=2)
C
  MSTP 51=7          ! CTEQ 5L in pythia 6.2
C
C General parameters
C ------------------
C
  MSTU 21=1 ! Check on possible errors during program execution
  MSTJ 11=3 ! Choice of the fragmentation function
C
C general QCD parameters
C
  MSTP 81=1   ! multiple parton interactions (1 is Pythia default)
  MSTP 82=4   ! multiple parton interactions (see p209 CERN-TH
7112/93)
  PARP 82=1.9
  PARP 83=0.5
  PARP 84=0.4
  PARP 90=0.16
C
C
  MSEL 0      ! define production mechanism using MSUB
  MSUB 22=1   ! Z pair production
C
C  ICUT 1=1    ! 2e 2mu selections
C  ICUT 1=2    ! 4e selections
   ICUT 1=3    ! 4mu selections
C
  RCUT  3=0.   ! pt cut for mu+, mu-
  RCUT  4=2.4   ! eta cut for mu+, mu-
C
  MSTJ 41=1    ! Switch off Pythia QED bremsshtrahlung
C
```

```
C PYTHIA Kinematics
C
    CKIN 3 = 25.5
    CKIN 41 = 5.
    CKIN 42 = 150.
    CKIN 43 = 5.
    CKIN 44 = 150.

END
EOF
```

WW -> 2mu 2nu (nubar) datacard

```
   MRPY 1= 123456
   CFIL 'EVTO' 'ww_2l.ntpl '
   KRUN 0
C
C cards below for mass production
C
   MSEL 0
   MSUB 25=1
C Switch off or define W decay channels
   MDME 190,1=0
   MDME 191,1=0
   MDME 192,1=0
   MDME 193,1=0
   MDME 194,1=0
   MDME 195,1=0
   MDME 196,1=0
   MDME 197,1=0
   MDME 198,1=0
   MDME 199,1=0
   MDME 200,1=0
   MDME 201,1=0
   MDME 202,1=0
   MDME 203,1=0
   MDME 204,1=0
   MDME 205,1=0
C MDME 206,1=1  !       (W+ --> nu e+     ON)
   MDME 207,1=1  !       (W+ --> nu mu+    ON)
C MDME 208,1=1  !       (W+ --> nu tau+   ON)
   MDME 209,1=0
C
   MSTP 51=7            ! CTEQ 5L in pythia 6.2
```

```
C
  MSTU 21=1 ! Check on possible errors during program execution
  MSTJ 11=3 ! Choice of the fragmentation function
C
C multiple interaction parameters
C
  MSTP 81=1   ! multiple parton interactions (1 is Pythia default)
  MSTP 82=4   ! multiple parton interactions (see p209 CERN-TH
7112/93)
  PARP 82=1.9
  PARP 83=0.5
  PARP 84=0.4
  PARP 90=0.16
C
C This part is added
C
  RCUT  3=0.   ! pt cut for mu+, mu-
  RCUT  4=2.4  ! eta cut for mu+, mu-
C
C
END
EOF
```

Drell-Yan Process pp -> gamma*/ Z^0 -> mu+ mu- datacard

```
C
LIST

C
C Pythia parameters
C -----------------
C
C First set random seed
C
  MRPY 1= 893426
  KSEL 0
  TRIG 1000000
C NSEL 10000
C
  CFIL 'EVTO',  'dy2mu.ntpl '
C
C --------------
C Set RUN number
C --------------
C
  KRUN 1
C
```

```
C CMS energy, (GeV)
C
  ECMS 1400
C
C
C PYTHIA Particle Mass
C
    PMAS 5,1 = 4.2        !mass of b quark
    PMAS 6,1 = 174.3       !mass of top quark
    PMAS 23,1 = 91.1882       !mass of Z
    PMAS 24,1 = 80.419       !mass of W
C
C PYTHIA Process Selection
C
   MSEL = 0       !(D=1) to select between full user control (0,
then use MSUB) and some preprogrammed alternative: QCD hight pT
processes (1, then ISUB = 11, 12, 13, 28, 53, 68), QCD low pT
processes (2, then ISUB = 11, 12, 13, 28, 53, 68, 91, 92, 94, 95)
    MDME 174,1 = 0        !Z decay into d dbar
    MDME 175,1 = 0        !Z decay into u ubar
    MDME 176,1 = 0        !Z decay into s sbar
    MDME 177,1 = 0        !Z decay into c cbar
    MDME 178,1 = 0        !Z decay into b bbar
    MDME 179,1 = 0        !Z decay into t tbar
    MDME 180,1 = -1       !Z decay into b' b'bar
    MDME 181,1 = -1       !Z decay into t' t'bar
    MDME 182,1 = 0        !Z decay into e- e+
    MDME 183,1 = 0        !Z decay into nu_e nu_ebar
    MDME 184,1 = 1        !Z decay into mu- mu+
    MDME 185,1 = 0        !Z decay into nu_mu nu_mubar
    MDME 186,1 = 0        !Z decay into tau- tau+
    MDME 187,1 = 0        !Z decay into nu_tau nu_taubar
    MDME 188,1 = -1       !Z decay into tau'- tau'+
    MDME 189,1 = -1       !Z decay into nu'_tau nu'_tau
    MSTJ 11 = 3       !Choice of the fragmentation function
    MSTJ 22 = 2       !Decay those unstable particles
    MSTP 51 = 7       !Choose CTEQ5L
C   MSTP 51 = 4       !structure function chosen
    MSTU 21 = 1       !Check on possible errors during program
execution
    MSUB 1 = 1       !ffbar -> gam/Z0
    PARJ 71 = 10.       !for which ctau  10 mm
C
C This part is added
C
  RCUT  3=0.    ! pt cut for mu+, mu-
```

```
  RCUT  4=2.4   ! eta cut for mu+, mu-
C
C PYTHIA Kinematics
C
   CKIN 1 = 5.       !(D=2. GeV) (CKIN 1,CKIN 2) = range of allowed
values for m hat = sqrt(s hat), the upper limit is inactive if
'CKIN 2' is negative (D=-1. GeV),to avoid pythia6136 bug with low
higgs mass
C
C PYTHIA Trigger
C
C
C GENERATOR User-defined
C
C
C PYTHIA Rarely used
C


END
EOF
```

gamma* + gamma* --> muon+ muon- datacard

```
LIST
C
C Pythia parameters
C -----------------
C
C First set random seed
C
  MRPY 1= 96154
  KSEL 0
  TRIG 1000000
C  NSEL 10
C
  CFIL 'EVTO', 'ggvir_ffbar.ntpl '
C
C --------------
C Set RUN number
C --------------
C
  KRUN 1
C
C CMS energy (GeV)
C
```

```
   ECMS 14000.
C
C
C PYTHIA Particle Mass
C
C
C PYTHIA Process Selection
C
    MSEL = 1
C-----------------------------------------------------------------
    MSUB 137=1          ! gam*(T) gam*(T) -> f f-bar
    MSUB 138=1          ! gam*(T) gam*(L) -> f f-bar
    MSUB 139=1          ! gam*(L) gam*(T) -> f f-bar
    MSUB 140=1          ! gam*(L) gam*(L) -> f f-bar
C
C
C
C PYTHIA Kinematics
C
    CKIN 1 = 0,        !(D=2. GeV) (CKIN 1,CKIN 2) = range of allowed
values for m hat = sqrt(s hat), the upper limit is inactive if
'CKIN 2' is negative (D=-1. GeV),to avoid pythia6136 bug with low
higgs mass

C
C
C
  RCUT  3=0.    ! pt cut for mu+, mu-
  RCUT  4=2.4   ! eta cut for mu+, mu-

END
EOF
```

gamma + gamma --> muon+ muon- datacard

```
LIST
C
C Pythia parameters
C ------------------
C
C First set random seed
C
  MRPY 1= 140369
  KSEL 0
  TRIG 1000000
C  NSEL 10
C
  CFIL 'EVTO',  'gg_ffbar.ntpl '
C
C --------------
C Set RUN number
C --------------
C
  KRUN 1
C
C CMS energy (GeV)
C
  ECMS 14000.
C
C
C PYTHIA Particle Mass
C
C
C PYTHIA Process Selection
C
   MSEL = 1
C---------------------------------------------------------------

    MSUB 58 = 1        ! Gamma physics
C
C
C PYTHIA Kinematics
C
   CKIN 1 = 0.       !(D=2. GeV) (CKIN 1,CKIN 2) = range of allowed
values for m hat = sqrt(s hat), the upper limit is inactive if
'CKIN 2' is negative (D=-1. GeV),to avoid pythia6136 bug with low
higgs mass
C
C
C This part is added
C
  RCUT  3=0.   ! pt cut for mu+, mu-
```

```
   RCUT  4=2.4   ! eta cut for mu+, mu-

END
EOF
```

Minimum Bias datacard

```
C Pythia/JETSET parameters
C ------------------------
C
C First set random seed
C
   MRPY 1= 123456
   KSEL 0
C
   CFIL 'EVTO',  'minbias.ntpl '
C
C --------------
C Set RUN number
C --------------
C
   KRUN 0
C
C don't use  PDF library (would be MSTP 52=2)
C
   MSTP 51=7          ! CTEQ 5L in pythia 6.1
C
C General parameters
C ------------------
C
   MSTU 21=1 ! Check on possible errors during program execution
   MSTJ 11=3 ! Choice of the fragmentation function
C
C general QCD parameters
C
   MSTP 81=1   ! multiple parton interactions (1 is Pythia default)
   MSTP 82=4   ! multiple parton interactions (see p209 CERN-TH
7112/93)
   MSTP 2=2    ! second order running alpha(s)
   MSTP 33=3   ! K-factor in alfas scale: alfas ->
alfas(parp(33)*Q**2)
   PARP 82=3.20  ! pt cutoff for multi-parton interactions
   PARP 89=14000. ! sqrt(s) for which PARP(82) is set
C

C  TRIG 100000
```

```
C  NSEL 100

  MSEL 1

END
EOF
```

Appendix C

OSCAR and ORCA

I have included the BuildFiles and Configuration File for OSCAR and ORCA in this appendix. The BuildFiles tell us what we can do in our executables and the configuration files display parameters and settings for the OSCAR (ORCA) software.

C.1 OSCAR Files

- OSCAR BuildFile
- OSCAR Run Configuration File

OSCAR BuildFile

```
<environment>

<lib name=MuonSensitiveDetector>
<lib name=TkSensitiveDetector>
<lib name=TkSimTrackSelection>

<lib name=TrackingHitsWriter>
<lib name=PersistentTrackingHits>
<lib name=TkNumbering>
<lib name=TrackerNumberingScheme>
<lib name=MuNumbering>
<lib name=MuonNumberingScheme>

<lib name=CaloSD>
<lib name=PersistentCaloHits>
<lib name=CaloHitsWriter>
<lib name=CaloNumbering>
<lib name=MantisSensitiveDetector>
<lib name=CaloNumberingScheme>

<lib name=Packaging>

<lib name=G4Configuration>
<lib name=MantisApplication>
<lib name=MantisSimEvent>
<lib name=MantisNotification>
<lib name=MantisGeometry>
<lib name=MantisGenerators>
<lib name=MantisPhysics>
<lib name=MantisMagneticField>
<lib name=BaseMagneticField>

<lib name=DDDInterface>
<lib name=G3SmallBank>
<lib name=G3MiniInterface>
<lib name=G3Interface>

<Group name=HBook>
<Group name=SimWriter>
<Group name=BaseUtilities>

<External ref=COBRA use=DDD>
<External ref=COBRA use=CARF>
<External ref=COBRA use=GeneratorInterface>
<External ref=COBRA use=Utilities>

<External ref=cmsim>
<External ref=geant4>
```

```
<External ref=geant>
<External ref=cern-generic>

<bin file=oscar.cpp></bin>

</environment>
```

OSCAR Run Configuration File

```
#
#  general
#
Configuration:List = true
Configuration:History = true
Verbose:test = true
MemoryDebug = true
CobraTrapFPE = false
TextColor = false
#
#  random generator sequence
#
CMSRandom:Seeds = 47 3
#
#  persistency
#
OutputDataSet = /System/OSCAR/zz_41 // Specify the output dataset
GoPersistent = true
DBPopulator:LocalMetaData = 1
DBPopulator:CommitInterval = 1
DBPopulator:MaxDBSize = 0.3
DBPopulator:checkDBInterval = 10
DBPopulator:UnNamedContainers = 1
DataSet:JobsPerDB = 1
DataSet:DBPoolIncrement = 1
#
#  extra packages
#
#  physics: QGSPlist,QGSClist,LHEPlist,FTFPlist
#  generators: MantisNtplReader,MantisGunReader,MantisTxtReader,
#        MantisPythiaReader,MantisStdHepReader
#
ExtraPackages = QGSPlist:MantisNtplReader:OscarExceptionHandler
#
#  run manager
#
RunManager:PhysicsTablesDir = PhysicsTables
RunManager:StorePhysicsTables = false
RunManager:RestorePhysicsTables = false
```

```
RunManager:StoreRNDM = false
RunManager:RestoreRNDM = false
RunManager:Verbose = 1
#
#  exception handler
#
OscarExceptionHandler:OverrideEnabled = true
OscarExceptionHandler:Verbosity = 2
#
#  sim tracks
#
EventAction:SaveSimTracks = true
#
#  geometry
#
DDDParserInstance = true
DDDConfigFile = OSCARconfiguration.xml
#
#  VCAL
#
VCalShowerLibrary:FileName = vcal5x5.rz
VCalShowerLibrary:FilePath = .:/afs/cern.ch/cms/cmsim/cmdb/vcal
#
#  physics
#
Physics:Verbose = 1
# cut in mm
Physics:DefaultCut = 10000
ProductionPhysics:BuildPreciseRangeTables = false
ProductionPhysics:ElectroNuclearProcess = true
ProductionPhysics:SynchrotronRadiation = true
#
#threshold for eBrem-CMS modification for electron rebirth in MeV
#
ProductionPhysics:ThresholdForElectron = 500
#
#  event generation
#
#  select and configure event generator
#
# available generators (readers): MantisNtplReader,
# MantisGunReader,
#  MantisTxtReader, MantisPythiaReader, MantisStdHepReader
#  to be chosen in ExtraPackages card
#
#  if MantisNtplReader
#
#Specify the ntuple file from CMKIN
#
EventNtplReader:NtplFileName =
/root/CMKIN_4_0_0/examples/make_ntpl_jobs/ggvir_ffbar2.ntpl
```

```
EventNtplReader:NtplID = 101
HepEventCmkinNtupleReader:FirstEvent = 1
#
#  if MantisGunReader choose
#  ParticleGun:Gun = FlatRandomPtGun
#  or FlatRandomEGun or FlatSteppingEGun
#  and configure for PID, MinEta, MaxEta, MinPhi, MaxPhi, MinPt,
#   MaxPt etc
#  see available ParticleGun cards in
#  COBRA/GeneratorInterface/HepEvent
#  if MantisTxtReader, specify input file
#  EventTxtReader:TxtFileName = pythia_event.data
#
#Generator:FirstEvent = 0
Generator:MaximumNumberOfDoableEvents = 10001
#
#  acceptance cuts
#
Generator:ApplyPhiCuts = false
Generator:ApplyEtaCuts = true
Generator:ApplyPtCuts = true
#Generator:MinPhiCut(deg) =
#Generator:MaxPhiCut(deg) =
Generator:MinEtaCut = -5.0
Generator:MaxEtaCut = 5.0
Generator:MinPtCut(MeV) = 40
Generator:MaxPtCut(MeV) = 9999999
#
#  vertex generator
#
Generator:VertexGenerator = GaussianEventVertexGenerator
#
#  if GaussianEventVertexGenerator
#
VertexGenerator:MeanX = 0.
VertexGenerator:MeanY = 0.
VertexGenerator:MeanZ = 0.
VertexGenerator:SigmaX = 0.015
VertexGenerator:SigmaY = 0.015
VertexGenerator:SigmaZ = 53.0
#
#  if FlatEventVertexGenerator use MinX,MinY,MinZ,MaxX,MaxY,MaxZ
#  instead of Mean and Sigma
#
#  magnetic field
#
MantisMagneticField:UseMagneticField = true
MagneticField:Name = CMSIMField
gufld:delta = 1.0
G4PropagatorInField:Verbose = 0
#
```

```
#  avoid loops, ping-pong between volumes, crashes
#
SteppingAction:KillBeamPipe = true
SteppingAction:Kick0Steppers = false
SteppingAction:PrintKicks = false
SteppingAction:CriticalEnergyForVacuum(MeV) = 2.0
SteppingAction:CriticalDensity(g/cm3) = 1.e-25
SteppingAction:CriticalStepLength(mm) = 1.e-6
SteppingAction:KickLength(mm) = 0.9e-6
SteppingAction:CriticalNumberOfSteps = 10000
#
#  specific detector actions
#
TrackerSim:ThrowOnBadHits = true
CaloSD:CheckHits = 25
#
#  number of events to be processed  (ignored if NO_G4_MACRO =
false)
#
NumberOfEventsToBeProcessed = 1000
#
#  NO_G4_MACRO = true  => batch processing
#              = false => interactive =>
#              if no G4InputFile specified => command line
#
NO_G4_MACRO = true
#
#G4InputFile =
#
#  disable G4 verbosity (G4cout, G4cerr)
#
G4Verbosity = silent
```

C.2 ORCA Files

- ORCA BuildFile
- ORCA Digitization Configuration File
- ORCA DST Recording Configuration File
- ORCA DST Reading BuildFile
- ORCA DST Reading Configuration File

ORCA BuildFile

```
#
# This is to write SimHits into a DB
#
<environment>
  <Group name=CaloHitWriter>
  <Use name=Calorimetry>

  <ignore> To produce Muon barrel, endcap and RPC hits </ignore>
  <Group name=MuonSimHitWriter>
  <Use name=Muon>

  <ignore> To produce all Tracker hits </ignore>
  <Group name=TkHitWriter>
  <Use name=Tracker>

  <Group name=G3Reader>
  <external ref=COBRA Use=CARF>

  <bin file=writeHits.cpp></bin>
</environment>


#
# This is to write RecHits/Digis into a DB
#
<environment>
  <ignore> To produce Calorimetry digis</ignore>
  <Group name=CaloHitReader>
  <Group name=CaloRHitWriter>
  <Group name=CaloRHitReader>
  <Group name=TriggerPrimitiveWriter>
  <Use name=Calorimetry>

  <ignore> To produce all Muon digis </ignore>
  <Group name=MuonDigiWriter>
  <Use name=CommonReco>
  <Use name=Muon>

  <ignore> To produce Tracker digis  </ignore>
  <lib name=DDDInterface></lib>
  <lib name=TkHitLoader></lib>
  <Group name=TkOscarReader>
  <Group name=TkDigiWriter>
  <Use name=Tracker>

  <ignore> To store L1 global trigger </ignore>
  <Group name=L1TRIGGER>
  <Use name=Trigger>
```

```
  <Group name=G3>
  <Group name=SimReader>
  <external ref=COBRA Use=CARF>

  <bin file=writeAllDigis.cpp></bin>
</environment>

#
# This is to produce a DST
#
<environment>
  <Group name=L1TRIGGER>

  <Group name=CaloRecHitReader>
  <lib name=EcalPlusHcalTower>
  <lib name=CaloCluster>

  <Group name=EgammaAnalysis>
  <Group name=EgammaSelections>
  <Group name=EgammaPhoton>
  <Group name=EgammaElectron>
  <lib name=EgammaOfflineReco>
  <Group name=EgammaTracks>
  <Group name=EgammaSetup>
  <Group name=EgammaL1>
  <Group name=EgammaMC>
  <lib name=EgammaCalibObject>
  <use name=ElectronPhoton>

  <group name=METreconstruction>

  <group name=TkTracks>
  <group name=PrincipalVertexFinder>
  <lib name=PixelTrackFinder>
  <lib name=TkPartialReco>
  <lib name=TkDigiSetUp>

  <lib name=MuonInternalReco>
  <group name=MuonReconstruction>
  <group name=MuonIsolation>

  <Use name=MuonReco>
  <Use name=Tracker>
  <Use name=TrackerReco>
  <Use name=Vertex>
  <Use name=Calorimetry>
  <Use name=ElectronPhoton>

  <lib name=PersistentJet>
  <Use name=Jets>
  <Use name=MET>
```

```
   <Use name=Trigger>

   <Group name=RecReader>
   <Group name=GeneratorCARFReader>
   <External ref=COBRA Use=GeneratorInterface>
   <External ref=COBRA Use=CARF>

   <bin file=writeDST.cpp>To write standard DSTs</bin>
   <bin file=writeStreams.cpp>To write standard streams</bin>
</environment>
```

Digitization Configuration File

```
MaxEvents = 1000  // Specify number of events to be digitized

InputCollections = /System/OSCAR/zz_41  // Specify input dataset
OutputDataSet    = /System/DIGIS/zz_41  // Specify output dataset

# - minbias as 2x10^33 s^-1 cm^-2 -------------------------------
#PUGenerator:Collection = /System/SimHits/minbias03/minbias03
#PUGenerator:AverageEvents = 3.5
#PUGenerator:MinBunch = -5
#PUGenerator:MaxBunch  = 3
#PUGenerator:FirstEvent = -2

GoPersistent = true

#-- database technicalities ---------------------------------------
  DBPopulator:CommitInterval = 15
  DBPopulator:MaxDBSize = 1.5
  DBPopulator:checkDBInterval = 15
  DataSet:JobsPerDB = 1
  DataSet:DBPoolIncrement = 1
  EVD:Default = OnDemand
  Events:EVDCreate = PreAllocate
  Collections:EVDCreate = PreAllocate
#-- location of objects -------------------------------------------
  TrackerDigi*:Location = Digis
  TrackerAssoc*:Location = Associations
  CD:Location = Digis
  HD:Location = Digis
  CR:Location = RecHits
  HR:Location = RecHits
  CS:Location = Digis
  CT:Location = Digis
  HT:Location = Digis
  MuonDigisBarrel:Location = Digis
  MuonDigisEndcap:Location = Digis
  MuonDigisRPC:Location = Digis
```

```
    MuonAssociationsBarrel:Location = Associations
    MuonAssociationsEndcap:Location = Associations
    MuonAssociationsRpc:Location = Associations
    L1Trigger*:Location = L1Trigger
  #-- what to store
    TrackerDigi*:Request = Auto
    TrackerAssoc*:Request = Auto
    CD:Request = Auto
    CR:Request = Nop
    HR:Request = Nop
    CS:Request = Auto
    MuonDigis*:Request = Auto
    MuonAssociations*:Request = Auto
    L1*:Request = Auto
  #-- pileup selection -------------------
    PUGenerator:AverageEvents = 3.5
    PUGenerator:Burst = 3
    PUGenerator:FirstEvent = -2
    PUGenerator:MaxBunch = 3
    PUGenerator:MinBunch = -5
    Calo:EBRY:MaxBunch = 3
    Calo:EBRY:MinBunch = -5
    Calo:EFRY:MaxBunch = 3
    Calo:EFRY:MinBunch = -5
    Calo:ESFX:MaxBunch = 1
    Calo:ESFX:MinBunch = -2
    Calo:HCAL:MaxBunch = 3
    Calo:HCAL:MinBunch = -5
    Muon:Barrel:MaxBunch = 15
    Muon:Barrel:MinBunch = -15
    Muon:Endcap:MaxBunch = 6
    Muon:Endcap:MinBunch = -6
  #-- the parameters ------------------------
  #- Tracker
  #only for pileup  PixelDigitizer:AddPixelInefficiency = 2
  #- ECAL
    CaloDataFrame:SuppressionStyle = 0
    CaloRecHit:SuppressionStyle = 2
    CaloRecHit:EcalBarrel = 1
    CaloRecHit:EcalEndcap = 1
    Ecal:Barrel:noise = 0.04
    Ecal:Barrel:photostatistics = 2.25
    Ecal:Barrel:threshold = 0.09
    Ecal:Endcap:noise = 0.150
    Ecal:Endcap:photostatistics = 1.8
    Ecal:Endcap:threshold = 0.45
    EcalTrigPrim:Threshold = 0.3
    CaloRecHit:Preshower = 1
    Presh:noise = 0.000015
    Presh:threshold = 0.000045
    CaloRecHit:SuppressUsingEt = 0
```

```
   Ecal:SelectiveReadoutExceptionHandling = 0
   Ecal:SelectiveReadoutGridSize = 1
   Ecal:SelectiveReadoutSingles = 1
   Ecal:SelectiveReadoutSingles:Barrel:threshold = 0.0
   Ecal:SelectiveReadoutSingles:Endcap:threshold = 0.0
   Ecal:SelectiveReadoutSingles:usethreshold = 0
   Ecal:SelectiveReadoutThreshold:high = 5.0
   Ecal:SelectiveReadoutThreshold:low = 2.5
   Ecal:SelectiveReadoutTowers:Barrel:threshold = 0.0
   Ecal:SelectiveReadoutTowers:Endcap:threshold = 0.0
   Ecal:SelectiveReadoutTowers:usethreshold = 0
#- HCAL
   CaloRecHit:HcalBarrel = 1
   HCALBarrel:Depth1GainFactor = 58.
   HCALBarrel:Depth2GainFactor = 117.
   HCALBarrel:Depth3GainFactor = 117.
   HCALBarrel:Depth4GainFactor = 150.
   HCALEndcap:Depth1GainFactor = 88.
   HCALEndcap:Depth2GainFactor = 178.
   HCALForward:Depth1GainFactor = 3.08
   HCALForward:Depth2GainFactor = 2.27
   HCALForward:Depth3GainFactor = 0.
   Hcal:noise = 3.0
   Hcal:noiseHF = 0.125
   Hcal:FADCcalibation = 4.0
   Hcal:FADCcalibationHF = 0.43
   Hcal:threshold = 0.5
   Hcal:presamples = 0
   Hcal:samples = 2
   Hcal:CollectionFactor = 1.139
   CaloTrigPrim:Hcal = 1
   HcalTrig:presamples = 0
   HcalTrig:samples = 2
   HcalTrig:CollectionFactor = 1.139
   HcalTrigPrim:Threshold = 0.3
#- Muon
```

DST Recording Configuration File

```
InputCollections =/System/DIGIS/zz_41   // Specify input data set
FirstEvent = 0
MaxEvents = 1000  // Specify number of events to be processed

GeaneUsed = true

HighLevelTriggerXML:XMLfile = 2x1033HLT.xml

# - generic selection - buggy
UseAllforDST = false
```

```
UseMCforDST = true
UseHLTforDST = true
# - specific selection -
UseL1triggerforDST = true
UseVtxforDST = true
UseTkforDST = true
UseTkPxlforDST = true
UseCaloforDST = true
UseBasicClusterforDST = true
UseClusterforDST = true
UseSuperClusterforDST = true
UseEndcapClusterforDST = true
UseL2BarrelCandidateforDST = true
UseL2EndcapCandidateforDST = true
UseOfflineBarrelforDST = true
UseOfflineEndcapforDST = true
UseHLTElectronTracksforDST = true
UseOfflineElectronTracksforDST = true
UseHLTPhotonCandidateforDST = true
UseOfflinePhotonCandidateforDST = true
UseHLTElectronCandidateforDST = true
UseOfflineElectronCandidateforDST = true
UseEGCalibrationforDST = true
UseMuforDST = true
UseMuL2forDST = true
UseMuL3forDST = true
UseMuGlobalforDST = true
UseMuCaloEtIsoforDST = true
UseMuCaloEffIsoforDST = true
UseMuTkPtIsoforDST = true
UseMuTkEffIsoforDST = true
UseKtJet1forDST = true
UseKtJet2forDST = true
UseIterConeJet1forDST = true
UseIterConeJet2forDST = true
UseBJetforDST = true
UseMETephtforDST = true
UseMETcrhforDST = true
UseMETicJetforDST = true
UseMETktJetforDST = true
UseMETparticleforDST = false
UseMETL1triggerforDST = true
UseHLTforDST = true

OutputDataSet = /System/DST/zz_4l  // Specify the output dataset
OutputRunNumber = 1

CMSRandom:Seeds = 47 2

#----
GoPersistent = true
```

```
SetUpClone:Reference = true
Digi:Update = true

DBPopulator:CommitInterval = 10
DBPopulator:MaxDBSize = 1.9


#---- the stuff for the Digis to add ----
CR:Request = Auto
HR:Request = Auto
L1trigger:Request = Auto
L1Trigger*:Location = L1Trigger
CR:Location = RecHits
HR:Location = RecHits
Events:Location = Events
Collections:Location = Events
*SkipAssocBuilding = true


#--- the name of the RecAlgorithm to store the output from
CombinatorialTrackFinder:Persistent = true
PixelTrackFinderFromTriplets:Persistent = true
PrincipalVertexFinder:Persistent=true
TowerBuilder:Persistent = true
EGBCluster:Persistent = true
EGCluster:Persistent = true
EGSCluster:Persistent = true
EGECluster:Persistent = true
EGCand*:Persistent = true
EPTrack*:Persistent = true
EGElectron:Persistent = true
EGPhoton:Persistent = true
EgammaMC*:Persistent = true
EgammaCalibObject:Persistent = true
L2Muon*:Persistent = true
L3Muon*:Persistent = true
StandAloneMuonReconstructor:Persistent = true
GlobalMuonReconstructor:Persistent = true
MuonCaloEtIsolator:Persistent = true
MuonCaloEffIsolator:Persistent = true
MuonTrackerPtIsolator:Persistent = true
MuonTrackerEffIsolator:Persistent = true
PersistentJetFinder:Persistent = true
METReconstructor*:Persistent = true
CombinedBTagging:Persistent = true
BTagJetWithTracksAlgo:Persistent = true
EgammaCalibObject:Persistent = true
HighLevelTriggerXML:Persistent = true
CombinatorialAndTrigger:Persistent = true
L2MuTrigger:Persistent = true
L3MuTrigger:Persistent = true
```

```
L25PixelTauTrigger:Persistent = true
L25TrackerBTrigger:Persistent = true
L25TrackerTauTrigger:Persistent = true
L2JPsiTrigger:Persistent = true
L2TauTrigger:Persistent = true
L3BJetTrigger:Persistent = true
L2JetTrigger:Persistent = true
L2MetTrigger:Persistent = true
L25EleTrigger:Persistent = true
L2EleTrigger:Persistent = true
L3EleTrigger:Persistent = true
L3PhotonTrigger:Persistent = true
##--- where to store it -----------------------
CombinatorialTrackFinder*:Location = DST
PixelTrackFinderFromTriplets*:Location = DST
PrincipalVertexFinder*:Location = DST
TowerBuilder*:Location = DST
EGBCluster*:Location = DST
EGCluster*:Location = DST
EGSCluster*:Location = DST
EGECluster*:Location = DST
EGCand*:Location = DST
EPTrack*:Location = DST
EGElectron:Location = DST
EGPhoton:Location = DST
EgammaMC*:Location = DSTMC
L2Muon*:Location = DST
L3Muon*:Location = DST
StandAloneMuonReconstructor*:Location = DST
GlobalMuon*:Location = DST
MuonCaloEtIsolator*:Location = DST
MuonCaloEffIsolator*:Location = DST
MuonTrackerPtIsolator*:Location = DST
MuonTrackerEffIsolator*:Location = DST
PersistentJetFinder*:Location = DST
METReconstructor*:Location = DST
CombinedBTagging*:Location = DST
BTagJetWithTracksAlgo*:Location = DST
EgammaCalibObject*:Location = Calib
HLTresult:Location = HLT
HLTcandidate:Location = HLT
```

DST Reading BuildFile

```
<environment>
  <group name=RecReader>
  <external ref=COBRA Use=CARF>

  <lib name=Workspace></lib>
```

```
  <group name=HLTXML>
  <group name=HLTElements>
  <use name=HLT>

  <Use name=Vertex>
  <Use name=Jets>
  <Use name=MET>
  <group name=BTagCount>
  <Use name=BReco>


  <group name=MuonRecoReader>
  <use name=MuonReco>

  <group name=MuonDSTReading>
  <use name=Muon>

  <group name=EgammaReco>
  <group name=EgammaBase>
  <group name=EgammaSelections>
  <group name=EgammaL1>
  <group name=EgammaElectron>
  <group name=EgammaPhoton>
  <lib name=EgammaOfflineReco>
  <use name=ElectronPhoton>

  <group name=L1TRIGGER>
  <use name=Trigger>

  <group name=CaloRecHitReader>
  <group name=CaloRHitReader>
  <group name=TriggerPrimitiveReader>
  <use name=Calorimetry>

  <group name=GeneratorCARFReader>
  <External ref=COBRA Use=GeneratorInterface>
  <External ref=root>

  <bin file=ORCAExercise.cpp name=Tutorial></bin>

</environment>

<Architecture name=Linux__2.4>
  BFARCH := Linux24-gcc3
</Architecture>

FILTEREDINCLUDE_tmp1=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 1,1000,$(INCLUDE)))
FILTEREDINCLUDE_tmp2=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 1000,2000,$(INCLUDE)))
```

```
FILTEREDINCLUDE_tmp3=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 2000,3000,$(INCLUDE)))
FILTEREDINCLUDE_tmp4=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 3000,4000,$(INCLUDE)))
FILTEREDINCLUDE_tmp5=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 4000,5000,$(INCLUDE)))
FILTEREDINCLUDE_tmp6=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 5000,6000,$(INCLUDE)))
FILTEREDINCLUDE_tmp7=$(shell $(SCRAMPERL)
$(TOOL_HOME)/FilterIncludes $(wordlist 6000,$(words
$(INCLUDE)),$(INCLUDE)))
INCLUDE:=$(shell $(SCRAMPERL) $(TOOL_HOME)/FilterIncludes
$(FILTEREDINCLUDE_tmp1) $(FILTEREDINCLUDE_tmp2)
$(FILTEREDINCLUDE_tmp3) $(FILTEREDINCLUDE_tmp4)
$(FILTEREDINCLUDE_tmp5) $(FILTEREDINCLUDE_tmp6)
$(FILTEREDINCLUDE_tmp7))
```

DST Reading Configuration File

```
CARFVerbosity = silent
UtilVerbosity = silent

CobraSignalHandler = false
FinalAbort = true
ForceExit = true
TextColor = false
TimingReport = true
MemoryDebug = false
MonRecAlisaBuilder = true

FirstEvent = 0
MaxEvents = 1000

InputFileCatalogURL =
@{xmlcatalog_file:/root/ORCA_8_4_0/src/Workspace/PoolFileCatalog.
xml}@
InputCollections =/System/DST/zz_41
```

C.3 DST Analysis Source Code

For DST analysis, the source codes for extracting muon information are **DST-analysis.cc** and **DSTanalysis.h**.

DSTAnalysis.cc

```
#include "Utilities/Configuration/interface/Architecture.h"

//---------------------
// This Class's Header --
//---------------------

#include "Workspace/MyAnalysis.h"

//--------------
// C++ Headers --
//--------------

#include <iostream>
#include <iomanip>
#include <vector>
#include <string>

//-----------------------------
// Collaborating Class Headers --
//-----------------------------

#include "TFile.h"
#include "TH1.h"
#include "TTree.h"

#include "CARF/G3Event/interface/G3EventProxy.h"
#include "CARF/SimEvent/interface/SimEvent.h"
#include "CARF/BaseSimEvent/interface/SimVertex.h"
#include "CARF/Reco/interface/RecQuery.h"
#include "CARF/Reco/interface/RecCollection.h"
#include "CARF/Reco/interface/FilteredRecCollection.h"

#include "GeneratorInterface/RecParticle/interface
/RawRecParticleTypeFilter.h"

#include "GeneratorInterface/RecParticle/interface
/RawRecParticleStableFilter.h"

#include "GeneratorInterface/RecParticle/interface
/RawRecParticle.h"

#include "Trigger/L1Trigger/interface/L1Trigger.h"
#include "HLT/HLTSteering/interface/HighLevelTrigger.h"
#include "Muon/PersistentMuon/interface/RecMuon.h"
#include "ElectronPhoton/EgammaElectron/interface
/ElectronCandidate.h"
#include "ElectronPhoton/EgammaOfflineReco/interface
```

```
/OfflineElectronReco.h"

const int MyAnalysis::MAXGEN;
const int MyAnalysis::MAXREC;

//---------------
// Constructors --
//---------------
MyAnalysis::MyAnalysis(const string& filename) {

 // first define a ROOT file
  theFile = new TFile(filename.c_str(),"RECREATE");

 // book histograms

  theHisto1 = new TH1F("H1","lepton pt spectrum
(sim)",100,0.0,200.0);
  theHisto2 = new TH1F("H2","lepton pt spectrum
(rec)",100,0.0,200.0);
  theHisto3 = new TH1F("H3","muon invariant mass
(sim)",100,0.0,190.0);
  theHisto4 = new TH1F("H4","muon invariant mass
(rec)",100,0.0,190.0);
  theHisto5 = new TH1F("H5","muon transverse momentum
(sim)",100,0.0,400.0);
  theHisto6 = new TH1F("H6","muon transverse momentum
(rec)",100,0.0,400.0);
  theHisto7 = new TH1F("H7","muon eta (sim)",100,-2.8, 2.8);
  theHisto8 = new TH1F("H8","muon eta (rec)",100,-2.8, 2.8);
  theHisto9 = new TH1F("H9","muon phi (sim)",100,0.,380.0);
  theHisto10 = new TH1F("H10","muon phi (rec)",100,0.,380.0);


 // book a simple ROOT Tree
  theTree = new TTree("T1","Analysis tree");

 // GENERAL block
  theTree->Branch("Run", &m_Run, "Run/I");
  theTree->Branch("Event", &m_Event, "Event/I");
  theTree->Branch("Accept", &m_Accept, "Accept/B");
  theTree->Branch("SingleMu", &m_SingleMu, "SingleMu/B");
  theTree->Branch("DiMu", &m_DiMu, "DiMu/B");
  theTree->Branch("HLT", &m_HLT, "HLT/B");

 // GEANT block
  theTree->Branch("Ngen", &m_Ngen, "Ngen/I");
  theTree->Branch("Pxgen", m_Pxgen, "Pxgen[Ngen]/F");
```

```
    theTree->Branch("Pygen", m_Pygen, "Pygen[Ngen]/F");
    theTree->Branch("Pzgen", m_Pzgen, "Pzgen[Ngen]/F");
    theTree->Branch("Pgen", m_Pgen, "Pgen[Ngen]/F");
    theTree->Branch("Ptgen", m_Ptgen, "Ptgen[Ngen]/F");
    theTree->Branch("Etagen", m_Etagen, "Etagen[Ngen]/F");
    theTree->Branch("Phigen", m_Phigen, "Phigen[Ngen]/F");
    theTree->Branch("Chagen", m_Chagen, "Chagen[Ngen]/I");
    theTree->Branch("Codegen", m_Code, "Codegen[Ngen]/I");
    theTree->Branch("Vxgen", m_Vxgen, "Vxgen[Ngen]/F");
    theTree->Branch("Vygen", m_Vygen, "Vygen[Ngen]/F");
    theTree->Branch("Vzgen", m_Vzgen, "Vzgen[Ngen]/F");
    theTree->Branch("Invgen", m_Invgen, "Invgen[Ngen]/F");


  // reconstructed muon block

    theTree->Branch("Nmu", &m_Nmu, "Nmu/I");
    theTree->Branch("Mhits", m_Mhits, "Mhits[Nmu]/I");
    theTree->Branch("MDof", m_MDof, "MDof[Nmu]/I");
    theTree->Branch("MChi2", m_MChi2, "MChi2[Nmu]/F");
    theTree->Branch("MX", m_MX, "MX[Nmu]/F");
    theTree->Branch("MY", m_MY, "MY[Nmu]/F");
    theTree->Branch("MZ", m_MZ, "MZ[Nmu]/F");
    theTree->Branch("MCha", m_MCha, "MCha[Nmu]/I");
    theTree->Branch("MPx", m_MPx, "MPx[Nmu]/F");
    theTree->Branch("MPy", m_MPy, "MPy[Nmu]/F");
    theTree->Branch("MPz", m_MPz, "MPz[Nmu]/F");
    theTree->Branch("MP", m_MP, "MP[Nmu]/F");
    theTree->Branch("MPt", m_MPt, "MPt[Nmu]/F");
    theTree->Branch("MEta", m_MEta, "MEta[Nmu]/F");
    theTree->Branch("MPhi", m_MPhi, "MPhi[Nmu]/F");
    theTree->Branch("MInv", m_MInv, "MInv[Nmu]/F");

}


//-------------
// Destructor --
//-------------
MyAnalysis::~MyAnalysis() {

  theFile->cd();
  theTree->Write();

  theHisto1->Write();
  theHisto2->Write();
  theHisto3->Write();
  theHisto4->Write();
  theHisto5->Write();
  theHisto6->Write();
```

```
  theHisto7->Write();
  theHisto8->Write();
  theHisto9->Write();
  theHisto10->Write();


  theFile->Close();


  delete theFile;

}


//-------------
// Operations --
//-------------


//
// analyze event
//
void MyAnalysis::analyzeEvent(G3EventProxy* ev) {

 //
 // run and event number
 //
  m_Run = ev->simSignal()->id().runNumber();
  m_Event = ev->simSignal()->id().eventInRun();

 //
 // first check the Level-1 trigger decision
 // for the definition of the Level-1 Trigger menu see:
 // http://cmsdoc.cern.ch/swdev/snapshot/ORCA/ReferenceManual
// /html/L1GlobalTrigger.html#L1GlobalTrigger

  L1Trigger l1trig;
  m_Accept = l1trig.decision();
  m_SingleMu = l1trig.decisionWord().element(0);
  m_DiMu = l1trig.decisionWord().element(1);
  cout.setf(ios::boolalpha);
  cout << "Level-1 Trigger decision: " << m_Accept << endl;
  cout << "Single/Di-muon trigger: "
       << m_SingleMu << "/"
       << m_DiMu << endl;

  cout << endl;
```

```
 //
 // now check HLT
 // for the definition of the HLT menu see:
 // Data/HLTconfig/2x1033HLT.xml
 //

 /*
  RecQuery q("HighLevelTriggerXML");
  RecCollection<HighLevelTriggerResult> hlt(q);
  m_HLT = (*(hlt.begin()))->getGlobalDecision();
  cout.setf(ios::boolalpha);
  cout << "High-Level Trigger decision: " << m_HLT << endl;

  vector<bool> response = (*(hlt.begin()))->getGlobalResponse();
  HighLevelTrigger* trigger =

dynamic_cast<HighLevelTrigger*>(RecoRegistry::current().find(RecC
onfig(q)));

 //vector< pair<string,string> > vetoLegend = trigger-
>anaVetoPattern();
  vector< pair<string,string> > responseLegend = trigger-
>anaResponsePattern();
  vector<bool>::const_iterator i = response.begin();
  vector< pair<string,string> >::iterator j =
responseLegend.begin();
  int cnt = 1;
    while ( i != response.end() ) {
    cout << setw(3) << cnt++ << ":  " << setw(5) << (*i) << "  " <<
j->second << " " << j->first << endl;
    i++; j++;
  }
  */

  cout << " " << endl;

 //
 // analyze Monte Carlo information
 //
  analyzeMCEvent();

 //
 // analyze simulated event (GEANT information)
 //
  analyzeSimEvent(ev);

 //
 // analyze reconstructed muons
 //
```

```
   analyzeMuons();

  //
  // calculate invariant mass
  //
   calculateInvarianMass();

  //
  // fill ROOT tree
  //
   theTree->Fill();

}


//
// analyze Monte Carlo event information
//
void MyAnalysis::analyzeMCEvent() {

  //
  // get MC muons and electrons
  //
   RecQuery q("RawParticleRecon");
   q.setParameter("FakeDet","Particles");
   RawRecParticleStableFilter stableFilter;
   RawRecParticleTypeFilter muonFilter("mu-","mu+");
   RawRecParticleTypeFilter electronFilter("e-","e+");
   Capri::MultipleFilterOr<RawRecParticle> f1;
   f1.add(&muonFilter);
   f1.add(&electronFilter);
   Capri::MultipleFilterAnd<RawRecParticle> f2;
   f2.add(&stableFilter);
   f2.add(&f1);
   FilteredRecCollection<RawRecParticle> mcParticles(q,f2);
   FilteredRecCollection<RawRecParticle>::const_iterator it =
mcParticles.begin();
   for ( it = mcParticles.begin(); it != mcParticles.end(); it++ )
{
   (**it).print();
   }

}


//
// analyze GEANT information
//
```

```
void MyAnalysis::analyzeSimEvent(G3EventProxy* ev) {

 //
 // get GEANT muons and electrons from signal event (no pile-up)
 //
  const float ptmin  = 0.0;
  const float etamax = 2.4;

  cout << "GEANT muons: " << endl;
  int igen = 0;
  int ntracks = ev->simSignal()->tracks()->size();
  for ( int it = 0; it < ntracks; it++ ) {
    SimTrack tk(ev->simSignal()->track(it));
    int ipart = tk.type();
    if ( abs(ipart) != 13 )continue;    // select muons/electrons
    string code = ( abs(ipart) == 13 )? "mu" : "e ";
    const HepLorentzVector& mom = tk.momentum();
    float pt  = mom.perp();
    float eta = mom.pseudoRapidity();
 //  float invmass = mom.invariantMass();
     if ( pt > ptmin && fabs(eta) < etamax ) {
      int charge = static_cast<int>(tk.charge());
      float phi = mom.phi();
      if ( phi < 0 )phi = 2*M_PI + phi;
      phi = (phi*180)/M_PI;
      const HepLorentzVector& vtx = tk.vertex().position();
      if ( vtx.perp() > 1.0 )continue;  // select only prompt
muons/electrons
      cout.setf(ios::showpoint);
      cout << setiosflags(ios::showpoint | ios::fixed)
           << setw(2)<< igen+1 << " : " << code << " "
           << "pt = "<< setw(5)<< setprecision(1)<< pt<< " GeV  "
           << "charge = " << setw(2)<< charge << " "
           << "eta = "<< setw(6)<< setprecision(3)<< eta << "  "
           << "phi = "<<setw(5)<<setprecision(3)<<phi<< " rad \t"
           << "vertex = (" << vtx.x() << "," << vtx.y() << "," <<
vtx.z() << ") " << endl;

     theHisto1->Fill(pt);
   if ( abs(ipart) == 13 ){

     theHisto5->Fill(pt);
     theHisto7->Fill(eta);
     theHisto9->Fill(phi);
   }

     if ( igen < MAXGEN ) {
       m_Pxgen[igen] = mom.px();
```

```
            m_Pygen[igen] = mom.py();
            m_Pzgen[igen] = mom.pz();
            m_Pgen[igen] = mom.rho();
            m_Ptgen[igen] = pt;
            m_Etagen[igen] = eta;
            m_Phigen[igen] = phi;
            m_Chagen[igen] = charge;
            m_Code[igen] = abs(ipart);
            m_Vxgen[igen] = vtx.x();
            m_Vygen[igen] = vtx.y();
            m_Vzgen[igen] = vtx.z();

            igen++;
          }
        }
      }

      m_Ngen = igen;

}


//
// analyze reconstructed muons
//

void MyAnalysis::analyzeMuons() {

  RecQuery q("GlobalMuonReconstructor");
  RecCollection<RecMuon> recmuons(q);
  RecCollection<RecMuon>::iterator muon = recmuons.begin();

  int nrec = recmuons.size();
  cout << "Number of reconstructed muons: " << nrec << endl;

  if ( nrec > 0 ) cout << "Reconstructed muons: " << endl;
  int idx = 0;

  while ( muon != recmuons.end() ) {
    if ( idx >= MAXREC ) break;
    int nmeas      = (**muon).foundHits();
    int dof        = (**muon).degreesOfFreedom();
    double chi2    = (**muon).chiSquared();
    double normChi2 = (**muon).normalisedChiSquared();

    m_Mhits[idx] = nmeas;
    m_MDof[idx] = dof;
    m_MChi2[idx] = chi2;

    cout << setiosflags(ios::showpoint | ios::fixed)
```

```
        << setw(2) << idx+1 << '\t' << "number of measurements = "
        << setw(2) << nmeas << "  "
        << "Chi2/DoF = " << setw(6) << setprecision(2) << chi2
        << "/" << setw(2) << dof
        << " = " << setw(6) << setprecision(3) << normChi2 << endl;

//
// state at closeat approach to vertex in the transverse plane
//
  FreeTrajectoryState* traj = (**muon).ipState();

  if ( traj ) {

    GlobalVector mom = traj->momentum();
    GlobalPoint pos  = traj->position();
    int charge       = traj->charge();

    float pt    = mom.perp();
    float eta   = mom.eta();
    float phi   = mom.phi();
    if ( phi < 0 ) phi = 2*M_PI + phi;
    phi = (phi*180)/M_PI;
    m_MX[idx] = pos.x();
    m_MY[idx] = pos.y();
    m_MZ[idx] = pos.z();
    m_MCha[idx] = charge;
    m_MPx[idx] = mom.x();
    m_MPy[idx] = mom.y();
    m_MPz[idx] = mom.z();
    m_MP[idx] = mom.mag();
    m_MPt[idx] = pt;
    m_MEta[idx] = eta;
    m_MPhi[idx] = phi;


    cout << setiosflags(ios::showpoint | ios::fixed)
         << setw(2) << idx+1 << '\t'
         << "pt = "<< setw(5) << setprecision(1) << pt<< " GeV  "
         << "charge = " << setw(2) << charge << " "
         << "eta = " << setw(6) << setprecision(3) << eta << "  "
         << "phi = " << setw(5) << setprecision(3) <<phi<< " rad"
         << "position = " << pos << endl;

    theHisto2->Fill(pt);
    theHisto6->Fill(pt);
    theHisto8->Fill(eta);
    theHisto10->Fill(phi);
  }
```

```
      else {
        m_MX[idx] = 0.;
        m_MY[idx] = 0.;
        m_MZ[idx] = 0.;
        m_MCha[idx] = 0;
        m_MPx[idx] = 0.;
        m_MPy[idx] = 0.;
        m_MPz[idx] = 0.;
        m_MP[idx] = 0.;
        m_MPt[idx] = 0.;
        m_MEta[idx] = 0.;
        m_MPhi[idx] = 0.;
      }

      idx++;
      muon++;
    }

    m_Nmu = idx;

}

//
// calculate invariant mass
//

void MyAnalysis::calculateInvarianMass() {

    const double mm = 0.105658357;
    const double me = 0.0005109989;

  //
  // analyze sim event
  //

     if ( m_Ngen <  2 ) return;

  // Pairing 2 muons from events with  2 and 3 muons

     if ( m_Ngen ==  2 && m_Ngen == 3) {
     cout<<"=====================================";
  // if(m_Ngen == 2){
   for ( int i = 0; i < m_Ngen; i++ ) {
      if ( m_Ptgen[i] < 5.0 ) continue;
      double m = ( m_Code[i] == 13 ) ? mm : me;
```

```
    HepLorentzVector
v1(m_Pxgen[i],m_Pygen[i],m_Pzgen[i],sqrt(m_Pgen[i]*m_Pgen[i] +
m*m));
    for ( int j = i+1; j < m_Ngen; j++ ) {
      if ( m_Code[i] != m_Code[j] ) continue;
      if ( m_Ptgen[j] < 5.0 ) continue;
      if ( m_Chagen[i] == m_Chagen[j] ) continue;
      HepLorentzVector
v2(m_Pxgen[j],m_Pygen[j],m_Pzgen[j],sqrt(m_Pgen[j]*m_Pgen[j] +
m*m));
      try {
        double invm = v1.invariantMass(v2);

      if (abs(91. - invm)<30){
      if ( m_Code[i] == 13 ) {
         theHisto3->Fill(invm);
         cout << i <<"   " << invm;
           m_Invgen[i] = invm;
      }
      }
      } catch (...) {
      cout << " exception caught" << endl;
      }
    }
  }
    }

 // Pairing 2 muons for events with 4 muons

   if ( m_Ngen == 4) {
   cout<<"========================================";
  for ( int i = 0; i < m_Ngen; i++ ) {
    if ( m_Ptgen[i] < 5.0 ) continue;
    double m = ( m_Code[i] == 13 ) ? mm : me;
    HepLorentzVector
v1(m_Pxgen[i],m_Pygen[i],m_Pzgen[i],sqrt(m_Pgen[i]*m_Pgen[i] +
m*m));
    for ( int j = i+1; j < m_Ngen; j++ ) {
      if ( m_Code[i] != m_Code[j] ) continue;
      if ( m_Ptgen[j] < 5.0 ) continue;
      if ( m_Chagen[i] == m_Chagen[j] ) continue;
      HepLorentzVector
v2(m_Pxgen[j],m_Pygen[j],m_Pzgen[j],sqrt(m_Pgen[j]*m_Pgen[j] +
m*m));
      try {
        double invm = v1.invariantMass(v2);
 // Check that muons are within Z mass window 75-105 GeV
      if (abs(91. - invm)<15){
```

```
        if ( m_Code[i] == 13 ) {
         theHisto3->Fill(invm);
         cout << i <<"   " << invm;
           m_Invgen[i] = invm;
      }
      }
      } catch (...) {
      cout << " exception caught" << endl;
      }
    }
  }
    }


//
// analyze rec event
//

      if ( m_Nmu <  2) return;
   if (m_Nmu == 2 ){
   for ( int i = 0; i < m_Nmu; i++ ) {
    if ( m_MPt[i] < 5.0 ) continue;
    HepLorentzVector
v1(m_MPx[i],m_MPy[i],m_MPz[i],sqrt(m_MP[i]*m_MP[i] + mm*mm));
    for ( int j = i+1; j < m_Nmu; j++ ) {
      if ( m_MPt[j] < 5.0 ) continue;
      if (m_MCha[i] == m_MCha[j] ) continue;
      HepLorentzVector
v2(m_MPx[j],m_MPy[j],m_MPz[j],sqrt(m_MP[j]*m_MP[j] + mm*mm));
      try {
        double invm = v1.invariantMass(v2);
            // Check that muons are within Z mass window 75-105 GeV
        if (abs(91. - invm)<30){
        theHisto4->Fill(invm);
        m_MInv[i] = invm;
      }
      } catch (...) {
      cout << " exception caught" << endl;
      }
    }
  }
   }


   else {
   for ( int i = 0; i < m_Nmu; i++ ) {
    if ( m_MPt[i] < 5.0 ) continue;
```

```
    HepLorentzVector
v1(m_MPx[i],m_MPy[i],m_MPz[i],sqrt(m_MP[i]*m_MP[i] + mm*mm));
    for ( int j = i+1; j < m_Nmu; j++ ) {
      if ( m_MPt[j] < 5.0 ) continue;
      if (m_MCha[i] == m_MCha[j] ) continue;
      HepLorentzVector
v2(m_MPx[j],m_MPy[j],m_MPz[j],sqrt(m_MP[j]*m_MP[j] + mm*mm));
      try {
        double invm = v1.invariantMass(v2);
// Check that muons are within Z mass window 75-105 GeV
            if (abs(91. - invm)<30){
        theHisto4->Fill(invm);
        m_MInv[i] = invm;
      }
      } catch (...) {
      cout << " exception caught" << endl;
      }
    }
  }
   }

}
```

DSTAnalysis.h

```
#ifndef MY_ANALYSIS_H
#define MY_ANALYSIS_H
//--------------
// C++ Headers --
//--------------

#include <string>

//--------------------
// Base Class Headers --
//--------------------------------
// Collaborating Class Declarations --
//--------------------------------

class TFile;
class TH1F;
class TTree;
class G3EventProxy;

//        --------------------
//        -- Class Interface --
//        --------------------

class MyAnalysis {

  public:

  /// constructor
    MyAnalysis(const string& filename = "ww_2l.root");

  /// destructor
    virtual ~MyAnalysis();

  /// analyze event
    void analyzeEvent(G3EventProxy*);

  private:

  /// analyze Monte Carlo  information
    void analyzeMCEvent();

  /// analyze GEANT information
    void analyzeSimEvent(G3EventProxy*);

  /// analyze reconstructed muons
    void analyzeMuons();
```

```
/// calculate invariant mass;
  void calculateInvarianMass();

private:

  TFile* theFile;

  TH1F*  theHisto1;
  TH1F*  theHisto2;
  TH1F*  theHisto3;
  TH1F*  theHisto4;
  TH1F*  theHisto5;
  TH1F*  theHisto6;
  TH1F*  theHisto7;
  TH1F*  theHisto8;
  TH1F*  theHisto9;
  TH1F*  theHisto10;


     TTree* theTree;

  static const int MAXGEN = 1500;
  static const int MAXREC = 1500;

  int m_Run;
  int m_Event;

  bool m_Accept;
  bool m_SingleMu;
  bool m_DiMu;
  bool m_HLT;

  int m_Ngen;
  float m_Pxgen[MAXGEN];
  float m_Pygen[MAXGEN];
  float m_Pzgen[MAXGEN];
  float m_Pgen[MAXGEN];
  float m_Ptgen[MAXGEN];
  float m_Etagen[MAXGEN];
  float m_Phigen[MAXGEN];
  int m_Chagen[MAXGEN];
  int m_Code[MAXGEN];
  float m_Vxgen[MAXGEN];
  float m_Vygen[MAXGEN];
  float m_Vzgen[MAXGEN];
  float m_Invgen[MAXGEN];
```

```
    int m_Nmu;
    int m_Mhits[MAXREC];
    int m_MDof[MAXREC];
    float m_MChi2[MAXREC];
    float m_MX[MAXREC];
    float m_MY[MAXREC];
    float m_MZ[MAXREC];
    int m_MCha[MAXREC];
    float m_MPx[MAXREC];
    float m_MPy[MAXREC];
    float m_MPz[MAXREC];
    float m_MP[MAXREC];
    float m_MPt[MAXREC];
    float m_MEta[MAXREC];
    float m_MPhi[MAXREC];
    float m_MInv[MAXREC];

};

#endif
```

Vitae

Mr. Raksapol Thananuwong was born on 17 October 1976 in Surin province, Thailand. He received his Bachelor of Arts in Physics from University of Chicago, IL, USA, in 2000.

Conference Presentations:

2004 R. Thananuwong and B. Asavapibhop. Luminosity measurement for the LHC using Z^0 production. *4th National Symposium on Graduate Research*, Lotus Hotel Pang Suan Kaew, Chiangmai (10-11 August 2004): O-ST-176

2004 R. Thananuwong and B. Asavapibhop. Luminosity measurement for the LHC using Z^0 production. *8th Annual National Symposium on Computational Science and Engineering*, Suranaree University of Technology (21-23 July 2004): 175

2004 R. Thananuwong and B. Asavapibhop. Luminosity estimation for the LHC using Z production *12th Annual Academic Conference*, Faculty of Science, Chulalongkorn University (18-19 March 2004): PH 8