

Table of Contents

Sensitivity study of the H_{bb} and H_{cc} couplings for e^+e^- collisions at FCC.....	1
How to obtain the necessary tools.....	1
Pythia8.....	1
Delphes.....	1
Simulation steps.....	2
Run Pythia8.....	2
Run Delphes.....	5
Run Analysis.....	6
List of references.....	6

Sensitivity study of the Hbb and Hcc couplings for e+e- collisions at FCC

-- LeaHeckmann - 2017-09-07

This page describes the methodology for determining the precision of BR(Hbb) and BR(Hcc) by simulating CMS at TLEP at 240 GeV with an integrated luminosity of 3500 fb⁻¹. A detailed description of the project as well as the results can be found in the corresponding summer student report [↗](#).

The first section will describe how to obtain the necessary toolkits. Afterwards the simulation and analysis will be divided in three steps:

- Generate signal and background events for Hbb and Hcc using Pythia8
- Simulate the CMS detector response using Delphes
- Run the data selection, analysis and fit using root and RooFit

How to obtain the necessary tools

Pythia8

Pythia can be downloaded from this website. [↗](#)

Follow the the instructions in the worksheet [↗](#) to install it together with the interface for HepMC.

Delphes

Download Delphes from this website [↗](#) and follow the instructions in the workbook [↗](#) to install it.

After installing Delphes a new CTagging module has to be added. This is done by adding the CTagging.cc and CTagging.h files in the modules folder. Afterwards links to the module have to be added in:

```
/modules/ModulesLinkDef.h  
/modules/TreeWriter.cc  
/modules/TreeWriter.h  
/classes/DelphesClasses.cc  
/classes/DelphesClasses.h
```

Examples for the files can be found in the attachment for the version Delphes-3.4.1.

After editing the files do the installation again by typing:

```
make clean  
./configure  
make
```

Part of this procedure is described in this tutorial [↗](#).

Download and install ExRootAnalysis by following the ExRootAnalysis user manual [↗](#).

Simulation steps

Run Pythia8

In order to generate the signal and background events pythia8 is used. It is based on using pythia cards, which fix the type of events generated.

Four set of events are generated, a signal and a background case for the bb channel and the same for the cc channel. For all of them 100,000 electron-positron collisions at 240 GeV are generated. PDG codes are used to select the possible decay products. A more detailed description as well as examples of the used cards can be found below. The cards are also provided in the attachment.

Signal bb channel Hide Signal bb channel

For the signal a Z and Higgs boson are generated. The Higgs decay is restricted to the bb channel and the Z can only decay to two muons.

```
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter or digit are comments.

! 1) Settings that could be used in a main program, if desired.
Main:numberOfEvents = 100000      ! number of events to generate
Main:numberToList = 0             ! number of events to print
Main:timesToShow = 20            ! show how far along run is
Main:timesAllowErrors = 30       ! max number of allowed errors
Main:showChangedSettings = on    ! print changed flags/modes/parameters
Main:showChangedParticleData = on ! print changed particle and decay data

! 2) LHC beams parameter settings.
Beams:idA = 11                   ! first beam, p = 2212, pbar = -2212
Beams:idB = -11                  ! second beam, p = 2212, pbar = -2212
#Beams:eCM = 91.                 ! CM energy of collision for Z pole run
#Beams:eCM = 161                 ! CM energy of collision for WW run
Beams:eCM = 240.                 ! CM energy of collision for HZ run
#Beams:eCM = 350.                ! CM energy of collision for tt run

! 3) Settings for the hard-process generation
HiggsSM:ffbar2HZ = on
23:onMode = off ! switch off all Z decays
23:onIfAny = 13 ! switch on Z->mu mu only
25:onMode = off ! switch off all H decays
25:onIfAny = 5 ! switch on all decays to b's
#25:offIfAny = 23 ! switch off H-> Z gamma. Only H->gamma gamma is left

! 4) Settings for the event generation process in the Pythia8 library.
PartonLevel:MI = on              ! no multiple interactions
PartonLevel:ISR = on             ! no initial-state radiation
#PartonLevel:ISR = off           ! no initial-state radiation
PartonLevel:FSR = on            ! no final-state radiation
HadronLevel:Hadronize = on      ! no hadronization

! 5) Non-standard settings; exemplifies tuning possibilities.
#SpaceShower:pT0Ref = 2.0       ! dampening of pT -> 0 divergence
#MultipleInteractions:pT0Ref = 3.0 ! dampening of pT -> 0 divergence
25:m0 = 125.0                   ! Higgs mass
```

Background bb channel Hide Background bb channel

For the background a second Z boson is generated instead of the Higgs and restricted to decay into bb.

```
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter or digit are comments.

! 1) Settings that could be used in a main program, if desired.
Main:numberOfEvents = 100000      ! number of events to generate
Main:numberToList = 0             ! number of events to print
Main:timesToShow = 20            ! show how far along run is
Main:timesAllowErrors = 30       ! max number of allowed errors
Main:showChangedSettings = on    ! print changed flags/modes/parameters
Main:showChangedParticleData = on ! print changed particle and decay data

! 2) LHC beams parameter settings.
Beams:idA = 11                   ! first beam, p = 2212, pbar = -2212
Beams:idB = -11                  ! second beam, p = 2212, pbar = -2212
Beams:eCM = 240.                 ! CM energy of collision

! 3) Settings for the hard-process generation
WeakDoubleBoson:ffbar2gmZgmZ = on
23:onMode = off ! switch off all decays
23:onIfAny = 5 13 ! switch on Z->mumu and Z->bb only
22:onMode = off ! switch off all decays

! 4) Settings for the event generation process in the Pythia8 library.
PartonLevel:MI = on              ! no multiple interactions
PartonLevel:ISR = on             ! no initial-state radiation
#PartonLevel:ISR = off           ! no initial-state radiation
PartonLevel:FSR = on            ! no final-state radiation
HadronLevel:Hadronize = on      ! no hadronization

! 5) Non-standard settings; exemplifies tuning possibilities.
#SpaceShower:pT0Ref = 2.0       ! dampening of pT -> 0 divergence
#MultipleInteractions:pT0Ref = 3.0 ! dampening of pT -> 0 divergence
```

Signal cc channel Hide Signal cc channel

For the signal a Z and Higgs boson are generated. The Higgs decay is restricted to the cc channel and the Z can only decay to two muons.

```
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter or digit are comments.

! 1) Settings that could be used in a main program, if desired.
Main:numberOfEvents = 100000      ! number of events to generate
Main:numberToList = 0             ! number of events to print
Main:timesToShow = 20            ! show how far along run is
Main:timesAllowErrors = 30       ! max number of allowed errors
Main:showChangedSettings = on    ! print changed flags/modes/parameters
Main:showChangedParticleData = on ! print changed particle and decay data

! 2) LHC beams parameter settings.
Beams:idA = 11                   ! first beam, p = 2212, pbar = -2212
Beams:idB = -11                  ! second beam, p = 2212, pbar = -2212
#Beams:eCM = 91.                 ! CM energy of collision for Z pole run
#Beams:eCM = 161                 ! CM energy of collision for WW run
Beams:eCM = 240.                 ! CM energy of collision for HZ run
#Beams:eCM = 350.                ! CM energy of collision for tt run
```

```

! 3) Settings for the hard-process generation
HiggsSM:ffbar2HZ = on
23:onMode = off ! switch off all Z decays
23:onIfAny = 13 ! switch on Z->mu mu only
25:onMode = off ! switch off all H decays
25:onIfAny = 4 ! switch on all decays to c's
#25:offIfAny = 23 ! switch off H-> Z gamma. Only H->gamma gamma is left

! 4) Settings for the event generation process in the Pythia8 library.
PartonLevel:MI = on ! no multiple interactions
PartonLevel:ISR = on ! no initial-state radiation
#PartonLevel:ISR = off ! no initial-state radiation
PartonLevel:FSR = on ! no final-state radiation
HadronLevel:Hadronize = on ! no hadronization

! 5) Non-standard settings; exemplifies tuning possibilities.
#SpaceShower:pT0Ref = 2.0 ! dampening of pT -> 0 divergence
#MultipleInteractions:pT0Ref = 3.0 ! dampening of pT -> 0 divergence
25:m0 = 125.0 ! Higgs mass

```

Background bb channel Hide Background bb channel

For the background a second Z boson is generated instead of the Higgs and restricted to decay into cc.

```

! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter or digit are comments.

```

```

! 1) Settings that could be used in a main program, if desired.
Main:numberOfEvents = 100000 ! number of events to generate
Main:numberToList = 0 ! number of events to print
Main:timesToShow = 20 ! show how far along run is
Main:timesAllowErrors = 30 ! max number of allowed errors
Main:showChangedSettings = on ! print changed flags/modes/parameters
Main:showChangedParticleData = on ! print changed particle and decay data

```

```

! 2) LHC beams parameter settings.
Beams:idA = 11 ! first beam, p = 2212, pbar = -2212
Beams:idB = -11 ! second beam, p = 2212, pbar = -2212
Beams:eCM = 240. ! CM energy of collision

```

```

! 3) Settings for the hard-process generation
WeakDoubleBoson:ffbar2gmZgmZ = on
23:onMode = off ! switch off all decays
23:onIfAny = 4 13 ! switch on Z->mumu and Z->cc only
22:onMode = off ! switch off all decays

```

```

! 4) Settings for the event generation process in the Pythia8 library.
PartonLevel:MI = on ! no multiple interactions
PartonLevel:ISR = on ! no initial-state radiation
#PartonLevel:ISR = off ! no initial-state radiation
PartonLevel:FSR = on ! no final-state radiation
HadronLevel:Hadronize = on ! no hadronization

```

```

! 5) Non-standard settings; exemplifies tuning possibilities.
#SpaceShower:pT0Ref = 2.0 ! dampening of pT -> 0 divergence
#MultipleInteractions:pT0Ref = 3.0 ! dampening of pT -> 0 divergence

```

By using these cards and the main42 example provided in the pythia installation HepMC files containing the generated events can be produced.

In order to be able to use the main42 example do:

```
make main42
```

in the example folder to produce a main42.exe. Then copy the cards to the same location and use the following commands to produce the HepMC files:

```
./main42.exe HZ_H2bb_Z2mm.pythia output_ZHbb.hepmc > log_hzbb.log
./main42.exe zz_bb.pythia output_ZZbb.hepmc > log_zzbb.log

./main42.exe HZ_H2cc_Z2mm.pythia output_ZHcc.hepmc > log_hzcc.log
./main42.exe zz_cc.pythia output_ZZcc.hepmc > log_zzcc.log
```

Run Delphes

For the simulation of the detector response Delphes is used. It uses cards to specify the detector geometries. The CMS cards with added CTagging method is used for this purpose and the included CTagging method can be seen below.

CTagging method Hide CTagging method

```
module CTagging CTagging {
  set JetInputArray JetEnergyScale/jets

  set BitNumber 0

  # add EfficiencyFormula {abs(PDG code)} {efficiency formula as a function of eta and pt}
  # PDG code = the highest PDG code of a quark or gluon inside DeltaR cone around jet axis
  # gluon's PDG code has the lowest priority

  # based on arXiv:1211.4462

  # default efficiency formula (misidentification rate)
  add EfficiencyFormula {0} {0.01+0.000038*pt}

  # efficiency formula for b-jets (misidentification rate)
  add EfficiencyFormula {5} {0.25*tanh(0.018*pt)*(1/(1+ 0.0013*pt))}

  # efficiency formula for c-jets
  add EfficiencyFormula {4} {0.85*tanh(0.0025*pt)*(25.0/(1+0.063*pt))}
}
```

Go to the Delphes directory and use the following commands to simulate the detector response:

```
./DelphesHepMC cards/delphes_card_CMS.tcl outputZHbb_btag100_ctag100.root output_ZHbb.hepmc
./DelphesHepMC cards/delphes_card_CMS.tcl outputZZbb_btag100_ctag100.root output_ZZbb.hepmc

./DelphesHepMC cards/delphes_card_CMS.tcl outputZHcc_btag100_ctag100.root output_ZHcc.hepmc
./DelphesHepMC cards/delphes_card_CMS.tcl outputZZcc_btag100_ctag100.root output_ZZcc.hepmc
```

Afterwards edit the CTagging algorithm in the Delphes card by using the configurations described in the summer student report. Edit the output file names to either btag100, btag110, btag 120, btag130 and ctag100, ctag50, ctag20, ctag10 according to the CTagging efficiencies in order to allow the analysis scripts to run. An example for btag130 and ctag20 can be seen below:

Modulated CTagging method Hide Modulated CTagging method

```

module CTagging CTagging {
  set JetInputArray JetEnergyScale/jets

  set BitNumber 0

  # add EfficiencyFormula {abs(PDG code)} {efficiency formula as a function of eta and pt}
  # PDG code = the highest PDG code of a quark or gluon inside DeltaR cone around jet axis
  # gluon's PDG code has the lowest priority

  # based on arXiv:1211.4462

  # default efficiency formula (misidentification rate)
  add EfficiencyFormula {0} {0.01+0.000038*pt}

  # efficiency formula for b-jets (misidentification rate)
  add EfficiencyFormula {5} {0.2*(0.25*tanh(0.018*pt)*(1/(1+ 0.0013*pt)))}

  # efficiency formula for c-jets
  add EfficiencyFormula {4} {1.3*(0.85*tanh(0.0025*pt)*(25.0/(1+0.063*pt)))}
}

```

Run Analysis

After obtaining the simulated events the following root script can be used to select and analyse the data.

The first script to be used is `zh_analysis_FCC.cpp`. It loops over all the different CTagging efficiencies and therefore only has to be started once. Change the input path in the script to the directory, where your Delphes output files are located and then run the script.

Afterwards the `NormalizePlot.cpp` is used to weigh the simulated events and merge the ZHbb, ZZbb, ZHcc and ZZcc into one root file, which will then be used for the fit. It also loops over all different CTagging efficiencies and has to be used only once.

Finally `RootFitAnalysis_Crystal_poly.cpp` is used to fit the obtained data and determine the branching ratios Hbb and Hcc and its uncertainties simultaneously. It has to be started for each CTagging efficiency possibility separately, while choosing the possibility inside the script.

List of references

1. Basic FCC-ee Analysis using Pythia, Delphes, and ROOT
2. First Look at the Physics Case of TLEP [↗](#)

This topic: CMSPublic > FCCeeHbbHccAnalysis

Topic revision: r4 - 2017-09-07 - LeaAlinaHeckmann



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback