Abstract

The DELPHES software provides a framework for fast simulation of particle interactions in a generic high-energy physics collider detector containing a tracking system, electromagnetic and hadronic calorimeters, and a muon system. It is an object-oriented system writen using the C++ programming language. Using input files originating from a Monte-Carlo event generator such as PYTHIA and HERWIG, DELPHES creates "high-level" analysis objects.

1 Introduction

A fast simulation of a typical LHC multipurpose detector response can be used to obtain more realistic observables and fast approximate estimates of signal and background rates for specific channels. DELPHES includes the most crucial detector apects as jet reconstruction, momentum/energy smearing for leptons, photons and hadrons and missing transverse energy. Starting from "particle-level" information, the package provides reconstructed jets, isolated leptons, photons, reconstructed charged tracks, calorimeter towers and the expected transverse missing energy. Although this kind of approach yields much realistic results than a simple "parton-level" analysis, a quick simulation comes at the expense of detector details. Therefore, the interactions not simulated in DELPHES are: secondary interactions, multiple interactions, photon conversion, electron Bremsstrahlung, magnetic field effects, detector dead materials.

The simulation package proceeds in two stages. The first part is executed on the generated events. "Particle-level" informations are read from input files and stored in a GEN ROOT tree. Three varieties of input files can currently be used as input in DELPHES. In order to process events from many different generators, the standard Monte Carlo event structure StdHep can be used as an input. Besides, DELPHES can also provide detector response for events read in "Les Houches Event Format" (LHEF) and ROOT files obtained using the **h2root** converter program. This first stage is performed using three C++ classes: `HEPEVTConverter`, `LHEFConverter` and `STDHEPConverter`. Afterwards, DELPHES performs a simple trigger simulation and reconstruct "high-level objects". These informations are organised in classes and each objects are ordered with respect to the transverse momentum. The output of the various C++ classes is stored in the *Analysis* tree. The program is driven by a datacard (data/DataCardDet.dat) which allow a large spectrum of running conditions by modifying basic detector parameters, including calorimeter and tracking coverage and resolution, thresholds or jet algorithm parameters.

2 Central detector simulation

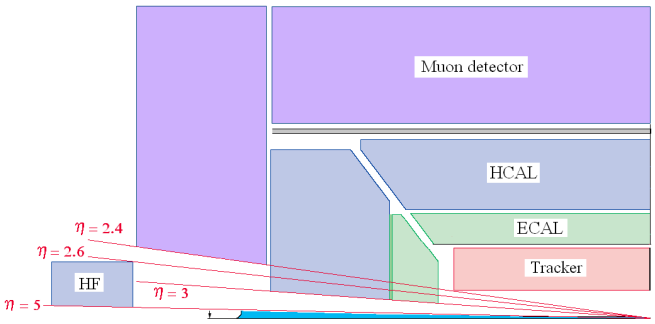

Figure 1: detectorAng.eps

The overall layout of the general purpose detector simulated by DELPHES is shown in figure **??**. A central tracking system surrounded by an electromagnetic (ECAL) and a hadron calorimeter (HCAL). A forward calorimeter ensure a larger geometric coverage for the measurement of the missing transverse energy. The fast simulation of the detector response takes into account geometrical acceptance

of sub-detectors and their finite energy resolution. No smearing is applied on particle direction.

Before starting to loop over events, the `RESOLution` class loads all sub-detector resolutions and coverage from the detector parameter file. If no such file is provided, predifined values are used. The coverage of the various sub-systems used in the default configuration are summarized in table 2.

Sub-system	Card flag	$\|\eta\|^{max}$
Tracking	`MAX_TRACKER`	2.5
Calorimeters	`MAX_CALO_CEN`	3.0
	`MAX_CALO_FWD`	5.0
Muon	`MAX_MU`	2.4

2.1 Simulation of calorimeters response

The energy of all particle considered as stable in the generator particle list are smeared according to a resolution depending which sub-calorimeter is assumed to be used for the energy measurement. For particles with a short lifetime such as the K_s, the fraction of electromagnetic or hadronic energy is determined according to its decay products. The response of the each sub-calorimeter is parametrized as a function of the energy

$$\frac{\sigma}{E} = \frac{S}{\sqrt{E}} \oplus \frac{N}{E} \oplus C, \qquad (1)$$

where S is the stochastic term, N the noise and C the constant term.

The response of the detector is applied to the electromagnetic and the hadronic particles through the `SmearElectron`and `SmearHadron` functions. The 4-momentum p^μ are smeared with a parametrisation directly derived from the detector techinal designs. In the default parametrisation, the calorimeter is assumed to cover the pseudorapidity range $|\eta| < 3$ and consists in an electromagnetic and an hadronic part. Coverage between pseudorapidities of 3.0 and 5.0 is provided by a forward calorimeter. The response of this calorimeter can be different for electrons and hadrons. The default values of the stochastic, noisy and constant terms as well as the "Card flag" names used in the configuration file are given in table 2.1.

Resolution Term		Card flag	Value
Central ECAL	S	`ELG_Scen`	0.05
	N	`ELG_Ncen`	0.25
	C	`ELG_Ccen`	0.0055
Forward ECAL	S	`ELG_Sfwd`	2.084
	N	`ELG_Nfwd`	0.0
	C	`ELG_Cfwd`	0.107
Central HCAL	S	`HAD_Shcal`	1.5
	N	`HAD_Nhcal`	0.
	C	`HAD_Chcal`	0.05
Forward HCAL	S	`HAD_Shf`	2.7
	N	`HAD_Nhf`	0.
	C	`HAD_Chf`	0.13

The energy of electron and photon particles found in the particle list are smeared using the ECAL resolution terms. Charged and neutral final state hadrons interact with the ECAL, HCAL and the forward calorimeter. Some long-living particles, such as the K_s, possessing lifetime $c\tau$ smaller than 10 mma are considering as stable particles although they decay in the calorimeters. The energy smearing of such particles is performed using the expected fraction of the energy, determined according to their decay products, that whould be deposited into the ECAL (E_{ecal}) and into the HCAL (E_{hcal}). Defining F as the fraction of the energy leading to a HCAL deposit, the two energy values are given by

$$E_{hcal} = E \times F \text{ and } E_{ecal} = E \times (1 - F), \qquad (2)$$

where $0 \leq F \leq 1$. The electromagnetic part is handled as the electrons, while the resolution terms used for the hadronic part are `HAD_Shcal`, `HAD_Nhcal` and `HAD_Chcal`. The resulting final energy given after the application of the smearing is then $E = E_{hcal} + E_{ecal}$.

2.2 Muon smearing

Muons candidates are searched The smearing ot the muon 4-momentum p^μ is given by a Gaussian smearing of the p_T function `SmearMuon`. Only the p_T is smeared, but neither η nor ϕ.

2.3 Tracks reconstruction

All stable charged particles lying inside the fiducial volume of the tracking coverage provide a track. The reconstructio efficiency is manageable in the input datacard through the `TRACKING_EFF` term. By default, a track is assumed to be reconstructed with 90% probability.

2.4 Calorimetric towers

All final particles, which are neither muons nor neutrinos are produce a calorimetric tower. The same particles enter in the calculation of the missing transverse energy. *what is used is the particle smeared momentum, not the calorimetric towers!*

2.5 Isolated lepton reconstruction

Photon and electron candidates are reconstructed if they fall into the acceptance of the tracking system and have a transverse momentum above the `ELEC_pt` value (10 GeV by default). Muons candidates are searched

Lepton isolation demands that there is no other charged particles with $p_T > 2$ GeV within a cone of $\Delta R < 0.5$ around the lepton.

3 "High-level" objects reconstruction

3.1 Jet reconstruction

Jets are reconstructed using a cone algorithm with $R = 0.7$ and make only use of the smeared particle momenta. The reconstructed jets are required to have a transverse momentum above 20 GeV and $|\eta| < 3.0$. A jet is tagged as b-jets if its direction lies in the acceptance of the tracker, $|\eta| < 0.5$, and if it is associated to a parent b-quark. A b-tagging efficiency of 40% is assumed if the jet has a parent b quark. For c-jets and light/gluon jets, a fake b-tagging efficiency of 10% and 1% respectively is assumed.

3.2 $b$tagging

The simulation of the b-tagging is based on the detector efficiencies assumed (1) for the tagging of a b-jet and (2) for the mis-identification of other jets as b-jets. This relies on the TAGGING_B, MISTAGGING_C and MISTAGGING_L constants, for (respectively) the efficiency of tagging of a b-jet, the efficiency of mistagging a c-jet as a b-jet, and the efficiency of mistatting a light jet (u,d,s,g) as a b-jet. The (mis)tagging relies on the particle ID of the most energetic particle within a cone around the observed (eta,phi) region, with a radius CONERADIUS.

3.3 Tau identification

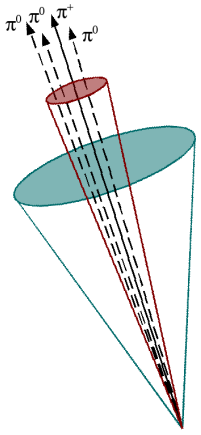

Figure 2: detectorAng.eps

Jets originating from τ-decay are identified using an identification procedure consistent with the one applied in a full detector simulation. The tagging rely on two tau properties. First, in roughly 75% of the time, the hadronic τ-decay products contain only one charged hadron and a number of π^0. Second, the particles arisen from the τ-lepton produce narrow jets in the calorimeter.

Electromagnetic collimation

To use the narrowness of the τ-jet, the *electromagnetic collimation* (C_τ^{em}) is defined as the sum of the energy in a cone with $\Delta R =$`TAU_CONE_ENERGIE` around the jet axis divided by the energy of the reconstructed jet. The energy in the small cone is calculated using the towers objects. To be taken into account a calorimeter tower should have a transverse energy above a given threshold `M_SEEDTHRESHOLD}`.A large fraction of the jet energy, denominated here with `TAU_EM_COLLIMATION` is expected in this small cone. The quantity is represented in figure 3 for the default values (see table 3.3)

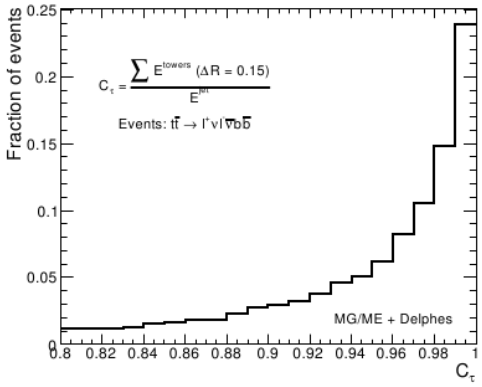

Figure 3:

τ selection using tracks

The tracking isolation for the τ identification requires that the number of tracks associated to a particle with $p_T >$ `PT_TRACK_TAU` is one and only one in a cone with $\Delta R =$ `TAU_CONE_TRACKS`. This cone should be entirely included in the tracker to be taken into account. This procedure selects taus decaying hadronically with a typical efficiency of 60%. Moreover, the minimal p_T of the τ-jet is required to be `TAUJET_pt`(default value: 10 GeV).

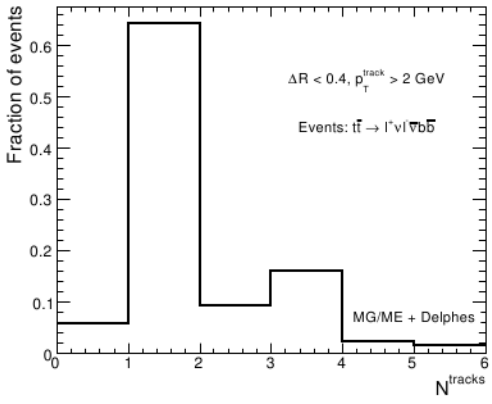

Figure 4:

Tau definition	Card flag	Value
$\Delta R^{for\ em}$	`TAU_CONE_ENERGIE`	0.15
min E_T^{tower}	`M_SEEDTHRESHOLD`	1.0 GeV
C_τ^{em}	`TAU_EM_COLLIMATION`	0.95.
$\Delta R^{for\ tracks}$	`TAU_CONE_TRACKS`	0.4
min p_T^{tracks}	`PT_TRACK_TAU`	2 GeV

3.4 Transverse missing energy

4 Very forward detectors simulation

Some subdetectors have the ability to measure the time of flight of the particle. This correspond to the delay after which the particle is observed in the detector, after the bunch crossing. The time of flight measurement of ZDC and FP420 detector is implemented here. For the ZDC, the formula is simply

$$t_2 = t_1 + \frac{1}{v} \times (\frac{s-z}{\cos\theta}), \tag{3}$$

where t_2 is the time of flight, t_1 is the true time coordinate of the vertex from which the particle originates, v the particle velocity, s is the ZDC distance to the interaction point, z is the longitudinal coordinate of the vertex from which the particle comes from, *theta* is the particle emission angle. This as-

sumes that the neutral particle observed in the ZDC is highly relativistic, i.e. travelling at the speed of light c. We also assume that $\cos\theta = 1$, i.e. $\theta \approx 0$ or equivalently η is large. As an example, $\eta = 5$ leads to $\theta = 0.013$ and $1 - \cos\theta < 10^{-4}$. The formula then reduces to

$$t_2 = \frac{1}{c} \times (s - z) \tag{4}$$

NB : for the moment, only neutrons and photons are assumed to be able to reach the ZDC. All other particles are neglected

To fix the ideas, if the ZDC is located at $s = 140$ m, neglecting z and θ, and assuming that $v = c$, one gets $t = 0.47$ μs.

5 Simulation physics validation

6 conclusion

Attention : in SmearUtil::NumTracks, the function arguments 'Eta' and 'Phi' have been switched. Previously, 'Phi' was before 'Eta', now 'Eta' comes in front. This is for consistency with the other functions in SmearUtil. Check your routines, when using NumTracks !

In the list of input files, all files should have the same type

Attention : in SmearUtil::RESOLution::BJets, the maximal energy was looked in CONERADIUS/2 instead of CONERADIUS. This bug has been removed.

Attention : for the tau-jet identification : CONERADIUS /2 was used instead of CONERADIUS !