Chapter 3

Event Simulation & Data Collection

The final states in high energy particle collision need to be very well understood. Experimentally, the final states need to be extracted from the detector signatures using well established algorithms at the online trigger, and with more sophisticated ones at the off-line level. The motive is to form correct physics objects from the signals readout by the various sub-detectors following the collision and avoid overlap among the various physics objects thus formed. At high luminosities, high beam intensities and small bunch crossing time, very huge amount of data is readout from the detector per second. The need of the hour is to have fast, efficient and reliable trigger and data acquisition system (as explained in Section 2.3). The reduced data collected per second is stored in reliable storage machines at CERN and at various storage and computing facilities across the globe. On the other side of the high energy physics world, lies another extremely challenging problem of replicating these particle collisions given to us by nature, by using computer simulations. Monte Carlo (MC) methods are used to model such scenarios as accurately as possible. This chapter deals with both the experimental (data collection and handling) as well as the theoretical aspects (MC simulations) encountered during the proton-proton collisions at the LHC.
3.1 Event Simulation

A Monte Carlo event generator is a computer program designed to simulate the final states of high-energy collisions in full detail down to the level of individual stable particles. A large number of simulated collision events are generated with a probability approximately proportional to the probability of producing similar kind of events in the real world. MC simulations make use of weighted random numbers to simulate the event-to-event fluctuations of the intrinsic process involved, based on the theoretical models being reproduced. Each simulated event consists of a list of final-state particles and their respective flavour and kinematic properties information.

Figure 3.1 shows the whole complex process being simulated in hadron-hadron collision, right from the hard sub-process to the final stages of event generation which are to be observed inside the detector as explained further in this section.

![Simulation of an event in hadron-hadron collision.](image)

The whole complex process can be sub-divided into simpler distinct steps inside the MC simulation program. During the first step, the simulation starts with partons from the colliding particles (proton) (green blob) interacting through a hard suprocess (red blob),

62
producing few outgoing fundamental objects such as quarks, leptons, photons, bosons or any hypothetical particles from some new theory. The cross section of this subprocess is calculated using the matrix element (ME) according to the Feynman rules. Next, the partons (colored particles - quarks and gluons) involved, radiate gluons which further radiate gluons or produce quark-antiquark pairs, and eventually lead to the formation of the parton showers (blue and red lines). The parton showers coming from incoming partons are known as initial state radiation (ISR) and those coming from outgoing partons are known as final state radiation (FSR). The colored particles produced during this shower can not exist as free particles due to quark confinement property (explained in Section 1.3.1). Eventually, as the strong interaction coupling rises the partons are bound into colorless hadrons through hadronization (light green ovals). Apart from these, the other constituents of the proton undergo multiple interactions producing underlying event interactions (magenta oval). In the end, any unstable (short lifetime) hadrons produced are subjected to further decays into stable particles.

### 3.1.1 Monte Carlo Generators

A variety of Monte Carlo event generators exist in the present day world, which can be used to simulate proton-proton collisions at LHC energies. Most of the MC generators are general-purpose, while few others exist which deal with specific processes. Some of the general-purpose generators include, PYTHIA [152], ALPGEN [153], MadGraph [154], Herwig++ [155] and SHERPA [156]. These are all leading order (LO) programs. There exist some next-to-leading order programs like JETPHOX [78][79], which produce only parton level events. The text in the following sub-sections describes the basic details of some of these generators. It includes a description of how each one of these generators incorporates combinations of the approaches and models needed to legitimize the whole collision process. It shows while these programs have much in common but they are not identical and nor they are interchangeable, because of the rational physics choices and modelling involved with each and also since the models chosen to explain one part of the
simulation influences the value of parameters of other parts that are needed to give the best agreement with experimental data.

3.1.1.1 PYTHIA

PYTHIA [152], one of the oldest and well established Monte Carlo event generators, is most widely used in the form of PYTHIA 6.4 (a fortran based version) since the last two decades. It is a general-purpose event generator and is extensively used for $e^+e^-$, $ep$, $p\bar{p}$ and $pp$ physics at LEP, HERA, Tevatron, and most recently at the LHC. PYTHIA provides a large number of of $2 \rightarrow 2$ hard coded subprocesses at leading order for the hard process based on analytical calculations and various QCD-based models. It can simulate all the generation steps explained earlier. PYTHIA models higher order corrections through parton showers simulated using parton shower algorithms based on an evolution variable. It uses the Lund string model [58] for hadronization and a highly developed multiple-interaction model for the underlying event [157]. The subprocess cross sections take into account the parton distribution functions to obtain the event rates. Several proton PDFs are hard coded inside PYTHIA and can be used easily by the user. Further sets are available through an interface to the LHAPDF library [158]. The PYTHIA samples used for the measurements in this thesis have been generated using PYTHIA 6.4. However, a more advanced C++ based version of PYTHIA 8 [159] exists nowadays. Inherited from PYTHIA 6.4, this version is developing fast. Along with the standard model processes PYTHIA also covers many beyond standard model scenarios.

3.1.1.2 MadGraph

MadGraph [154] is a matrix element calculator. It is capable for creating amplitudes for the relevant subprocess automatically and produces the mappings for the integration over the phase space. The next step of calculating event cross sections and generation of unweighted events is handled by another program called MadEvent. The parton level event information (particle IDs, momenta, spin etc.) is stored in the Les Houches format.
files. The rest of the simulation is carried on by PYTHIA. MadGraph calculates tree level amplitudes for the hard process plus 0, 1, 2 .... jets (2 → N final states). Thus, the hard gluon radiation is also calculated at the matrix element level. The various subprocesses from different number of final states are then combined with proper weights to give an inclusive final state simulated on matrix element level, giving a precise description of the event topology. The parton level events from MadGraph are then interfaced with PYTHIA which handles the rest of the generation steps such as parton showering and hadronization. Such kind of treatment of radiation both at the matrix element level and at parton shower level could lead to double counting of events. Such scenarios are avoided by using Matrix element-Parton shower (ME-PS) merging algorithms. They define regions of phase space so that the jets generated from matrix element level calculations can be separated from jets from the parton showering. The Madgraph-PYTHIA interface employs the MLM [161] kind of matching algorithm. MadGraph samples are generally produced with cuts on the scalar sum of the transverse momenta of the outgoing partons, \( H_T = \sum P_T \), unlike other generators where a cut on transverse momenta of the outgoing partons is simply used.

### 3.1.1.3 SHERPA

“Simulation of High Energy Reactions of Particles” or better known as SHERPA [156] is one of the relatively new event generator coded in C++ from the beginning. The program is in itself a complete package with inbuilt matrix-element generators, AMEGIC++ and Comix, and phase-space generator Phasic which automatically calculate and integrate tree-level amplitudes for the implemented models. Thus, SHERPA can be used both as a cross section integrator and a parton-level event generator. It covers subprocesses for the Standard Model (both 2 → 2 and higher multiplicity) and for new models given the Feynman rules. Comix is the default matrix-element generator for multijet processes. A dipole formulation is used for parton showering [162], and a cluster model for hadronization [163]. For the underlying event SHERPA uses a multiple-interaction model
based on that of PYTHIA but differing in some respects [164]. An extended version of
the Catani-Krauss-Kuhn-Webber (CKKW) algorithm [165] is implemented as the ME-
PS merging algorithm in SHERPA ensuring proper phase space coverage. It has been
extensively validated with previous collider data [166][167]. The treatment for prompt
photon production inside SHERPA differs from other generators like PYTHIA, with the
fragmentation contribution being modelled by the incorporation of QED effects into the
parton shower. Additionally, the ME-PS merging technique has been extended to pro-
cesses involving photons [167]. The tree level matrix elements of variable photon and
QCD parton multiplicity are combined with a QCD+QED parton shower. So on top of
the leading order matrix elements the parton shower produces interleaved QCD+QED
emissions, thus treating the QCD and QED corrections fully democratically or on equal
footing. This overall yields an excellent agreement with existing experimental data on
prompt photon production at both the $e^+e^-$ and hadron colliders [167]. This makes pre-
dictions with next-to-leading order accuracy while still providing parton-shower merging
that can be used to study the detector response. The SHERPA package can be interfaced
with LHAPDF package to choose the required set of PDF sets needed for the generation.

3.1.1.4 JETPHOX

JETPHOX [78][79] is a program dealing with calculation for the reactions $hh \rightarrow
\gamma/hadron + jet + X$. It produces events at parton level and with a next-to-leading order
accuracy. By simply integrating over the jet, one can calculate cross sections for single
inclusive photon/hadron cross section at NLO. JETPHOX can be switched in between
photon/hadron production mode automatically according to the choice of fragmentation
functions for photons/hadrons in the input. JETPHOX is a cross section integrator which
uses Monte Carlo integration over the requested phase space of photon $P_T$ and rapidity
$\eta$ in order to compute the cross section binned in both variables. The end result yields
a double differential cross section, $d^2\sigma/dP_Td\eta$. The calculation is dependent on the def-
initions of the fragmentation functions, the parton distribution functions and the choice
of theoretical scales too. These are provided as inputs to the JETPHOX program. JETPHOX can calculate cross section of inclusive prompt photon production, regardless of the isolation of the photon from additional hadronic activity. However, the isolation energy of prompt photons at the parton level can be estimated and restricted, and thus yielding the isolated prompt photon cross section [79] at the parton level. This feature enables the comparison of data with these NLO predictions in a realistic manner. JETPHOX is also interfaced with LHAPDF package for getting the standard sets of PDFs and also enables use of the required fragmentation functions for providing as input for the cross section calculation. However, JETPHOX does not include contributions from the underlying event.

It is understood that PYTHIA, MadGraph and SHERPA are all leading order generators. The prompt photon production using the various MC programs includes at leading order, contributions from the Compton \((qg \rightarrow \gamma q)\) and annihilation \((q\bar{q} \rightarrow \gamma g)\) Feynman diagrams. While PYTHIA takes care of the additional radiation using parton shower, SHERPA and MadGraph also have the higher multiplicity matrix element contributions. JETPHOX on the other hand is a next-to-leading generator, and so the predictions from JETPHOX give results at NLO accuracy. MC simulations from the above mentioned MC programs have been used at various points in the thesis to compare data to the current available theoretical models, estimate acceptances and efficiencies, study background processes, derive signal yields, derive calibration corrections or correct data for any detector effects so that direct comparison to theory can be made, and also to understand systematic uncertainties in measurements derived from data.

3.1.2 Detector Simulation and Digitization

For doing data-driven studies, obtaining calibration co-efficients or obtaining efficiencies etc., the end products of the pp collisions obtained from the modelling of the collision event using a MC generator are required to be put through a simulation of the detector. This is accomplished through two steps. Firstly, GEANT4 provides a toolkit [168][169] for
the simulation of the behaviour of various particles traversing different kinds of material in each of the sub-detectors inside the CMS. Thus, it provides a way to characterize the CMS detector by modelling response of each of the sub-detectors. For modelling the material interactions, the detector simulation also takes into account effects coming from the magnetic fields and the specific geometry of the detector. Effects of energy loss, and the showering in the detector material are fully simulated with each particle leaving distinguishable signatures inside the detector. In the end well simulated hits in the detectors are obtained. In the second step the detector response is further built by emulating the response of the readout and trigger electronics of the detector to these hits, through the process of digitization. Noise and other factors are also taken care of during this step. In the end the raw output obtained is similar to the format as will be given by the data acquired from the real operation of the detector. The event can now be reconstructed using the CMS reconstruction software, described in the next chapter “Reconstruction of Physics Objects”. In this way one can obtain precise prognosis and validation of the entire system of experimental setup.

3.2 Data Collection and Handling

CMS data processing is divided into three stages - online data collection, a high level trigger selection of data and an offline reconstruction workflow. Data selection for analysis takes place both online, when the detector is experiencing live collisions, and offline, after the online selected data has been stored for further processing. Online selection is performed using the high level trigger algorithms and performs the online reduction of data based on some general physics object algorithms. A full event reconstruction is then performed offline on the data passing the trigger selections. The raw data is in fact reconstructed using more sophisticated reconstruction algorithms at the offline level. Later an offline selection is also performed on these reconstructed objects based on object and analysis specific criterion. This step is usually performed by the end user of the data. This
chapter will be give a general overview of the online data collection procedure along with briefing about the tiered structure of CMS responsible for handling the data worldwide. Also one section is completely dedicated to the description of triggers being used in the measurements summarized in this thesis in subsequent chapters.

3.2.1 Online Data Collection

As described in Section 2.3, the CMS trigger and data acquisition system is designed to collect and analyze the information collected from the detector during proton-proton collisions at the LHC bunch-crossing rate. The triggers are designed in a way to simplify the tasks for the end-users or the analyzers of the data. The two level trigger system processes huge amount of information coming from the data collected by the calorimeters and the muon chambers and selects events with some interesting signatures. Triggering at the LHC represents a very challenging task as it is given the crucial role of picking out events suitable for physics given that the production cross sections for the benchmark phenomena ranges over tens of order of magnitude. It has to deliver a huge reduction factor, suppressing millions of background events, while maintaining a very high efficiency at the same time, choosing events for physics. Also, the trigger decisions have to be taken really fast and should be able to easily adapt to the different running conditions and physics targets.

A given HLT path is a collection of a sequence of reconstruction and selection modules. Each module is responsible for performing a well-defined task which range from getting the reconstructed physics objects (electrons, muons, jets, $E_T^{miss}$, etc.) to evaluating the final decision for a trigger path. The CMS software framework ensures that in case an intermediate filter module decision on a trigger path is negative, the trigger path rejects that event and skips the execution of the remaining modules in the path. However, an event is expected to pass through rest of the HLT trigger paths even after being accepted or rejected by a previous path. Thus, a single event can fire any number trigger paths depending on whether it satisfies their selection criteria. The first essential filter module
of any HLT trigger path is based on a suitable L1 seed (consisting of L1 bit[s] and L1 object[s]), which serves as a starting point for that specific HLT trigger. In other words, each HLT trigger path must be seeded by one or more L1 trigger bit seeds.

The HLT uses “Trigger Menus” to specify the kind of pre-selection of events required by the two-step trigger system. As the instantaneous luminosity varies the menus need to be re-adjusted to obtain the best possible output. These trigger menus are thus configurable according to the amount of data reduction required. A given HLT menu comprises of a set of trigger paths optimized for a particular value of instantaneous luminosity. Therefore, the later trigger menus (with higher instantaneous luminosity) either require tighter selections at the trigger level or the triggers are “prescaled”. In other words, the triggered events are not accepted always but reduced to a fraction given by a prescale factor. A prescale factor of N means only 1 in N events are accepted by the trigger path.

3.2.1.1 Photon Data

The measurements described in this thesis require a photon (electromagnetic object) in the final state. To perform such measurements, the analyzers use data sets formed out of the electromagnetic trigger paths. This section gives a description of the online trigger selection used in the L1 and HLT algorithms used for the production of such data.

**Level-1 Electromagnetic Trigger:** The combined information processing by the Regional Calorimeter Trigger (RCT), Global Calorimeter Trigger (GCT), and Global Trigger (GT) contribute towards the formation of the L1 electromagnetic trigger [150]. Since the electrons and photons (\(e/\gamma\)) are indistinguishable at L1 level, hence they are treated as the same object. An arrangement of 5 \(\times\) 5 array of crystals corresponds to a trigger tower (TT) both in barrel and endcaps of the ECAL. The trigger primitives (TP’s) are formed by summing up the transverse energy detected by the crystals in a single TT. These trigger primitive’s reach the RCT, where L1 trigger candidates are formed by combining 4 \(\times\) 4 TT’s. The GCT then sends four most energetic candidates to the GT. GT is responsible for generating the final L1 decision by applying thresholds on \(E_T\) of the
candidate. The sliding window of $3 \times 3$ trigger tower’s is responsible for implementing the $e/\gamma$ L1 algorithm. For candidate with $E_T > 4 \text{ GeV}$, a Fine Grain (FG) veto bit is used to show that the energy pattern of the candidate is consistent with an electron or a photon. The bit is set to 1 if candidate’s central tower contains 2 adjacent strips ($2 \times 5$) with a significant portion ($\sim 90\%$) of the tower’s energy. The energy contribution from the associated HCAL towers is also calculated. This kind of compressed information from the ECAL and HCAL is forwarded through the RCT. Here the ECAL energy of each trigger tower is examined and compared to four of its adjacent neighbours. If it is greater than its neighbours, it is considered a suitable $e/\gamma$ candidate, with its $E_T$ given as the sum of the central TP of the sliding window and the largest deposit in one of its 4 neighbour towers adjacent by side. Along with the FG veto bit, the associated HCAL energy contribution of the central tower is bound below a threshold value of 5%. Non-isolated candidates are required to pass these mentioned criterias. However, a candidate is tagged “isolated”, if out of the eight nearest TT’s surrounding the central tower, there are atleast five contiguous towers (forming an L shape) with a transverse energy below a threshold of 3.5 GeV. Figure 3.2 shows the pictorial representation of the L1 $e/\gamma$ algorithm described in this text.

Figure 3.2: Pictorial representation of the L1 $e/\gamma$ algorithm.
Anomalous Signals: The CMS detector observed certain anomalous signals in the ECAL barrel region [170] during collision data taking. These signals led to fake isolated high energy deposits in individual crystals. These were identified as being caused by the direct ionization within the sensitive volume of the avalanche photodiode (APD’s) linked with the crystals, by highly ionizing particles. They are found at a rate of 1 in $10^3$ minimum bias events. These so called “Spikes” induce very high trigger rates at both L1 and HLT level. If left untreated these contribute majorly to the 100 kHz CMS L1 trigger rate bandwidth [170] at high luminosities. An incident $e/\gamma$ candidate, deposits most of its energy ($\sim 80\%$) in a central crystal and most of the remaining energy in four adjacent crystals. Thus, the spikes can be discriminated from em signals by taking advantage of this lateral energy distribution. A “Swiss Cross” variable, $s = 1 - E_4/E_1$, where $E_4$ is the sum of $E_T$ of the four adjacent crystals and $E_1$ is the $E_T$ of the central crystal, has been implemented offline for this purpose. A similar kind of topological variable also exists at the online level for on-detector electronics, termed the strip Fine Grain Veto Bit (sFGVB) and used at the L1 level. Each TP has an assigned sFGVB bit, which is set to 1 if out of the five constituent strips of the TP any one of it has at least two crystals with a $E_T$ above a programmable sFGVB threshold ($\sim$ few hundred $MeV$). However, if the sFGVB has been set to zero and further the trigger tower $E_T$ lies above a killing threshold, the deposit can be archived as “spike-like”. That tower no longer supplies to triggering the corresponding event and its energy is re-defined to zero. After the detailed optimization, the sFGVB and killing thresholds have been set to 258 $MeV$ and 8 $GeV$ respectively.

The final step towards the L1 decision is taken by the GT. The latter applies the necessary criteria on the collections of isolated and non-isolated candidates and determines whether the event passes the set of L1 trigger paths or not.

The High Level Trigger: The CMS HLT uses similar kind of specialized reconstruction algorithms as used in the offline analysis. The offline reconstruction algorithms are described in detail in the chapter on “Reconstruction of Physics Objects”. However, since the HLT has limited amount of time to process the information as compared to the
time available offline, the online HLT algorithms are optimized for best performance in less time. The events processed by the HLT are seeded by a L1 trigger object. The photon selection at the HLT level relies on information from the ECAL and HCAL sub-detectors. Certain loose requirements are placed on the photon candidates to reduce contribution coming from fakes.

**Photon Triggered dataset:** For the data being analyzed for the subject of this thesis, the single photon triggered datasets are used. The events, to be selected using required HLT paths, are seeded by a L1 single $e/\gamma$ trigger object with an $E_T$ threshold of 8 or 20 GeV (L1 SingleEG8 or L1 SingleEG20). $E_T$ thresholds at L1 level are usually set to low values in order to avoid performance issues while using higher threshold values at the HLT level. At the HLT level, the level-2 filter requires a HLT reconstructed photon candidate to be above a minimum $E_T$ threshold. Further optionally, the HLT level-3 filters are used to enforce some other loose requirements on the candidate in terms of being isolated in the tracker and calorimeters or satisfying certain shower shape requirements. This thesis will cover two similar analysis but based on data taken at different value of instantaneous luminosities of the LHC and during two separate running periods. The instantaneous luminosity increased during the change from one run period to another and hence the trigger menus were re-adjusted. The trigger paths thus evolved over the time and so the details of the specific trigger paths used for a particular analysis will be covered in the respective chapters.

### 3.2.2 Data Handling Structure

**Dataset division:** Each HLT path is usually constructed in a way in order to select specific physics signatures (e.g. photons, electrons, jets, muons etc.). The events accepted by the HLT are delivered to the storage manager system via the event building network. The storage manager stores the event data into files using the definition of the output streams. A given stream groups events selected through specific HLT paths, each group dealing with specific kind of physics or experimental studies (e.g. primary physics stream,
express stream, calibration streams, etc.). Within a particular stream a set of similar paths selecting analogous physics signatures are further grouped into primary datasets (PD’s). Each PD is thus a subset of a particular stream and consists of the events satisfying a certain group of paths selected by that stream. The PD’s are further divided into secondary datasets (SD) and central skims (CS). There are separate dataset working groups which are responsible to define, manage, monitor and evolve the PD’s, SD’s, and the CS’s based on the guidelines from the computing, trigger, and the offline working groups. The offline analysis usually begins with the groups of stored data which are defined as primary datasets. The specific datasets used for the analysis described in this thesis are given in the respective chapters.

**The Computing Model:** Managing data at the CMS faces challenges due to its huge data volume and the necessary computing resource requirements. Due to technical and funding reasons and also due to the global outreach of the CMS experiment in terms of collaborators outside CERN, the CMS computing structure has been constructed as a global distributed system of computing services and resources. These interact with each other using the Grid resources and services, irrespective of physical locations and on a fair share basis. The infrastructure exploits the computing, storage and connectivity resources required for a variety of activities such as processing, archiving, and accessing of data, reconstruction and analysis of data, even Monte Carlo event generation, and all other of computing-related activities. It is based on elements of the Worldwide LHC Computing Grid project [171]. Centered on the CERN’s central computing system, the hierarchial struture of the Grid is organised in a tiered architecture. Each of the tier levels provides different resources and services. The Tier-0 (T0) center at CERN hosts the initial processing of data coming from the detector. It accepts the RAW data from the CMS Online Data Acquisition and Trigger System (TriDAS), packs the data into PD’s based on trigger info, assists in first pass reconstruction of RAW data (forming RECO and Analysis Object Data (AOD)) and distributes the RAW and the RECO/AOD data to the various Tier-1 (T1) sites. The T0 does not provide any analysis resources. The
Tier-1 is in general used on large scale and for more centrally organised activities and can distribute data (RECOs, skims and AOD) with various other T1 and with associated Tier-2 (T2) centers. It usually hosts a subset of the data from the T0, provides CPU for re-reconstruction, skimming, calibration activities and also provides secure storage and redistribution for MC events generated by the T2’s. While Tier-2 are smaller centers but they have substantial CPU resources, provide capacity for user analysis, calibration studies, and Monte Carlo production. They mostly rely on T1 for access and secure storage of data (generally Monte Carlo) produced at the T2, which are distributed to other T2’s by the T1. The T2’s provide resources for local communities for doing Grid based analysis (T2 resources are available to the whole experiment through the Grid) and Monte Carlo simulation for the whole experiment. There can be further Tier-3 level sites located at individual universities and institutes for physics analysis purposes, which are not part of the Grid though. Figure 3.3 shows the organization of the Tier sites associated with central Tier-0 CERN.

![Figure 3.3: CMS Tier Level Organization Chart.](image)

**CMS software:** The users have access to a dedicated object-oriented structure based on C++ and python called the CMS software framework (CMSSW) [172]. The single software framework of CMS supports a variety of user applications covering the entire range
of experimental work, including the online trigger system, reconstruction and simulation
executables, data quality monitoring, analysis-based event skimming and user data anal-
ysis. CMSSW software regularly follows a fast development cycle, resulting in a regular
upgradation of releases. A new stable CMSSW version is released only after a complete
release validation process has been conducted. Further, the final analysis of the end out-
put is performed using analysis tools such as ROOT [173], the data formats of the end
output being compatible with the ROOT formats.