

TABLE OF CONTENTS

	Page
List of Figures	iv
List of Tables	vi
Chapter 1: Overview	1
1.1 Measurement of Cross Section	1
Chapter 2: Theory Introduction	3
2.1 Standard Model	3
2.2 Electroweak Physics	3
2.2.1 Z Production	3
Chapter 3: Experimental Setup	4
3.1 CERN	4
3.2 Large Hadron Collider	4
3.3 Compact Muon Solenoid	5
3.3.1 Detection of Particle Interactions	6
3.3.2 Coordinate System	8
3.3.3 Magnet	9
3.3.4 Tracker	10
Pixels	10
Strips	11
3.3.5 Calorimeters	12
ECAL	13
HCAL	13
3.3.6 Muon System	14
3.3.7 Trigger	16
Level-1 Trigger	17
Regional Calorimeter Trigger	18
High-Level Trigger	19
Trigger Menus	19

3.3.8	Luminosity	20
Chapter 4:	Event Simulation	22
4.1	Monte Carlo Event Generation	23
4.1.1	Types of Monte Carlo Simulation Programs	23
4.1.2	Monte Carlo Generator Programs	26
	PYTHIA	27
	POWHEG	29
	Other Generators	30
4.2	Detector Simulation	31
4.2.1	GEANT Detector Model	31
4.2.2	Level-1 Trigger Emulator	32
Chapter 5:	Event Reconstruction	33
5.1	Detector Object Reconstruction	33
5.1.1	Electromagnetic Calorimeter Reconstruction	33
5.1.2	Track Reconstruction	34
	Kalman Filter	34
	Gaussian Sum Filter	35
5.2	Electron Reconstruction	35
5.2.1	ECAL-driven Electron Seeding	35
	Superclusters	36
	Pixel Hit-matching	37
5.2.2	Tracker-driven Electron Seeding	39
5.2.3	Electron Track Reconstruction	40
5.3	Beam Spot Reconstruction	40
5.4	Primary Vertex Reconstruction	40
Chapter 6:	Event Selection	42
6.1	Online	42
6.1.1	Trigger Eras	42
6.1.2	Level-1 Trigger	43
6.1.3	High-Level Trigger	44
6.1.4	Primary Datasets	46
6.2	Offline	47
6.2.1	Electron Selection	47

	Conversion Rejection	48
	Electron Isolation	49
	Electron Identification	50
	Matching Offline-Reconstructed Objects to Trigger Objects	52
6.2.2	$Z \rightarrow e^+e^-$ Event Selection	53
	Acceptance for $Z \rightarrow e^+e^-$ Events	53
	Efficiency of Selection for $Z \rightarrow e^+e^-$ Events	57
	Distribution of Z Kinematic Variables	60
Chapter 7:	Analysis Method	64
7.1	Invariant Mass Spectrum	64
7.2	Cross-Section Extraction	65
7.3	Background Subtraction	67
7.3.1	Monte Carlo Estimation	67
7.3.2	Template Method	67
	Template Definitions	69
	Results	70
7.3.3	Sideband Subtraction	74
7.3.4	Comparison of Background Subtraction Methods	76
7.4	Estimation of Systematic Uncertainties	77
7.4.1	Introduction to Error Analysis	77
7.4.2	Systematic Uncertainty due to Luminosity	79
7.4.3	Systematic Uncertainties from Theory	79
7.4.4	Other Sources of Systematic Uncertainty	80
	Electron Energy Scale	81
	Monte Carlo Sample for Efficiency	81
	Fitting for Efficiency	82
	Background Subtraction/Modeling	82
7.4.5	Summary of Uncertainties	82
Chapter 8:	Results	84
8.1	Cross Section Measurement	84
8.2	Comparison to Theory	84
8.3	Comparison to Previous Experiments	84
Chapter 9:	Summary and Conclusions	85

LIST OF FIGURES

Figure Number		Page
3.1	Diagram of the LHC layout	5
3.2	Expanded view of the CMS detector	6
3.3	Cross-sectional view of the CMS detector	7
3.4	Detection of particles in the CMS detector	8
3.5	Three-dimensional display of a $Z \rightarrow e^+e^-$ event in the CMS detector	9
3.6	Structure of pixel detector layers	11
3.7	Diagram of layers in tracker barrel	12
3.8	Conceptual diagram of drift tube chambers	15
3.9	Diagram of cathode strip chamber	16
3.10	Diagram of resistive plate chamber	17
3.11	Structure of the Level-1 trigger	18
3.12	Luminosity collected by CMS as a function of time, 2010	21
4.1	Diagram of typical steps used to simulate event in parton shower method	27
4.2	Schematics showing the Lund string model of hadronization	29
5.1	Typical energy distribution in a basic cluster	36
5.2	Schematic of pixel-hit search algorithm in electron-seeding	38
6.1	Conversion rejection variables after previous cuts	49
6.2	Electron tracker isolation variables before respective cuts	51
6.3	Electron ECAL isolation variables before respective cuts	52
6.4	Electron HCAL isolation variables before respective cuts	53
6.5	$\Delta\eta_{in}$ electron identification variables after previous cuts	54
6.6	$\Delta\phi_{in}$ electron identification variables after previous cuts	55
6.7	$\sigma_{i\eta i\eta}$ electron identification variables after previous cuts	56
6.8	H/E electron identification variables after previous cuts	57
6.9	Reconstructed electron p_T , η , ϕ spectra after full selection	62
6.10	Kinematic quantities of Z candidates after full selection	63
7.1	Electron-pair invariant mass after full selection	65

7.2	Electron-pair invariant mass by η	66
7.3	Shapes of signal and background templates used in template method of background subtraction	72
7.4	Results of template fit to data	73
7.5	Functional fit of $Z \rightarrow e^+e^-$ invariant mass peak for background subtraction .	76

LIST OF TABLES

Table Number		Page
6.1	Trigger paths used to select data.	45
6.2	Thresholds for WP80 relative isolation.	50
6.3	WP80 electron identification cuts.	52
6.4	Criteria for determining acceptance.	54
6.5	Acceptance values.	56
6.6	Efficiencies of electron selection steps.	59
6.7	Efficiencies of data trigger paths	60
7.1	Number of events passing full selection for each background, estimated from Monte Carlo simulation.	68
7.2	Summary of the standard working point as well as the working point definitions used in the template method. <i>Italics indicates a cut changed relative to WP80.</i>	71
7.3	Signal and background fractions of data sample	74
7.4	Summary of background estimates from different methods	77
7.5	Summary of uncertainties	83

Chapter 1

OVERVIEW

1.1 Measurement of Cross Section

$$\sigma_Z \times \text{BR}_{Zee} = \frac{n_{Zee}}{\mathcal{L} \times \epsilon \times A}$$

where

$$\epsilon = \epsilon_{trig} \times \epsilon_{reco} \times \epsilon_{sel}$$

particles? although that's theory intro (but need SOME particle-specific stuff)

the practical purpose of MC – can simulate all stages of gen/sim/reco/etc and compare everything step-by-step with data, as well as of course getting some quantities that require extra knowledge, like acceptance (and predicting efficiencies, rates, etc) HA, that sounds like a few of those things in the pythia manual intro! and I came up with them myself!

glossary? (explanatory text, with true glossary in appendix?)

- * “event”! “high-energy”

- * “phase space”!

- * cross section!

- * here or in theory chapter define “tree-level”. ALSO IN THEORY: explain how Feynman diagrams are so useful! writing down the matrix element and all that AND PDFs, basically whole process of stuff simulated really happens and should be explained. AND DEF OF PARTONS yeah, basically explain all the stuff in the MC chapter INCLUDING ISR and FSR ALSO define “jets” and talk about how they come from quarks and gluons (define partons) and relate to “hadronization” – not entirely relevant for this analysis, but possible in general AND define “color” AND asymptotic freedom, how it takes more and more energy to get them further apart

- * histogram? errors? (stat/syst) etc?

- * eta, phi, transverse quantities
- * pileup. maybe under, like, “the anatomy of an event” or “the structure of an event?”
- * “kinematic”
- * “hard” vs “soft”
- * I think I (hopefully) got minbias already, and this isn’t exactly the place for it...
- * underlying event!
- * initial- and final-state
- * signal, background
- * acceptance, efficiency (especially for that formula above)
- * invariant mass
- * HERE OR IN THEORY INTRO need to do on-shell/off-shell decays to explain why

Z mass has a spectrum and not a single value

- * statistical and systematic uncertainties or errors
- * “data-driven” stuff? need for efficiency, but also systs??
- * acronyms?
- * ROOOOOOOOOT! other software, like CMSSW?

Chapter 2

THEORY INTRODUCTION

2.1 *Standard Model***2.2** *Electroweak Physics**2.2.1 Z Production*

Chapter 3

EXPERIMENTAL SETUP

3.1 CERN

The European Organization for Nuclear Research, or CERN, is an international laboratory just outside Geneva, Switzerland. It was founded in 1954 as a collaborative effort between twelve European countries. It now has twenty member states, as well as many observer and other non-member states, and is one of the world’s major particle physics facilities.

3.2 Large Hadron Collider

The Large Hadron Collider (LHC) is a circular particle accelerator near Geneva, Switzerland, with a circumference of 27 km. Two proton beams circulate in opposite directions around the ring and cross at several points, which are home to large particle detectors. Figure 3.1 shows a diagram of the LHC layout. The two general-purpose detectors, CMS and ATLAS, sit on opposite sides of the ring, while the two smaller specialty detectors, LHCb and ALICE, sit at the interaction points to either side of ATLAS. Each beam consists of a series of proton bunches, with a maximum of 2835 “buckets” in the beam to be filled by bunches. The bunch structure is such that the nominal bunch crossing rate is 40 MHz.

The proton beams are accelerated from injection energy by an RF chamber, which provides an energy boost with each circulation of the beam until the collision energy is achieved. The beams are steered around the ring by 8-Tesla magnetic fields produced in 15-meter-long superconducting niobium-titanium dipole magnets, and focused by quadrupole magnets, 5-7 m long. The LHC uses a design in which both proton beampipes are contained in the same housing, allowing the same liquid helium cooling system to serve both.

The LHC began colliding proton beams in March 2010, quickly reaching its 2010 center-of-mass operating energy of 7 TeV (3.5 TeV per proton beam). At this energy it delivered over 47 pb^{-1} of collisions, with a maximum instantaneous luminosity of over $2 \cdot 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$.

The LHC maximum design energy is 14 TeV (7 TeV per beam), and its design luminosity is $10^{34} \text{ cm}^{-2}\text{s}^{-1}$.



Figure 3.1: Diagram of the LHC layout.

3.3 Compact Muon Solenoid

The Compact Muon Solenoid (CMS) is one of the two general-purpose experiments for the LHC. Figures 3.2 and 3.3 show the structure of the CMS detector. It is cylindrical, 21.5 m long and with a 15-m diameter, and weighs 12,500 tons. The overall structure of the detector is formed by the iron yoke framework surrounding the inner detector, with the central section, the “barrel”, divided into five ring-shaped slices, while each end of the cylinder, or “endcap”, consists of several circular plates. These sections fit around the solenoid and inner detector, enclosing it completely.

Particle detection is done by several dedicated subdetectors. The electromagnetic and hadronic calorimeters fit inside the solenoid, with the tracking system inside the calorime-

ters. The three components of the muon system are outside the solenoid, integrated with the iron magnetic field return yoke: the drift tube chambers, resistive plate chambers, and cathode strip chambers.

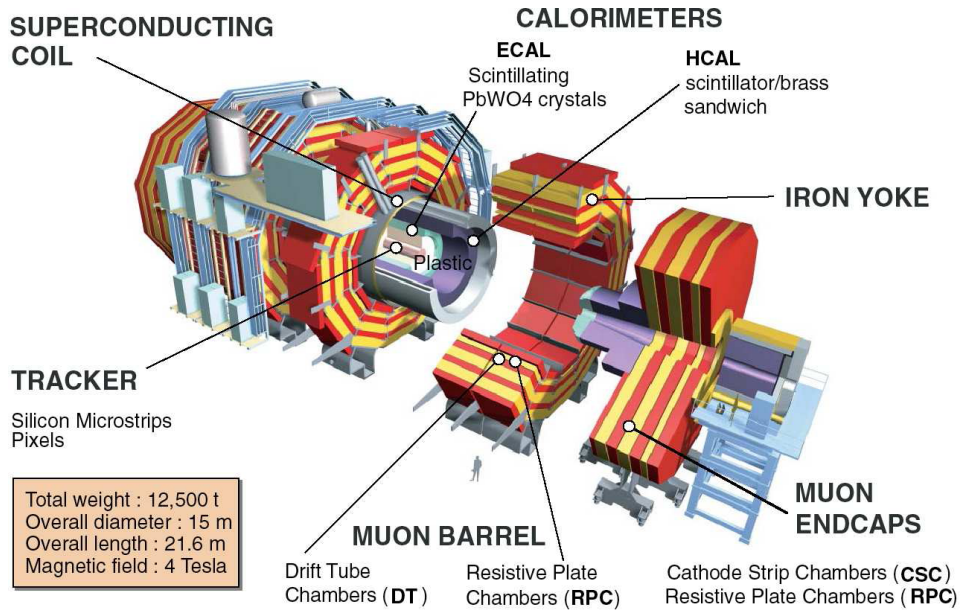


Figure 3.2: Expanded view of the CMS detector, showing the ring and slice structure as well as the placement of each subsystem.

3.3.1 Detection of Particle Interactions

Figure: Particle detection in CMS, Fig. 3.4

The various parts of the CMS detector work together to differentiate the end-products of proton interactions. As the end-product particles traverse their outward trajectory, they pass through the different subdetectors in turn. Figure 3.4 shows how end-product particles interact with the subsystems of the CMS detector. The innermost subdetector is the tracker, which reconstructs a particle's track through it from a set of points where it detected an interaction. The tracker only registers particles with an electric charge; neutral particles pass through undetected. The electromagnetic calorimeter detects particles that interact primarily electromagnetically. In this way the first two "layers" of the detector can

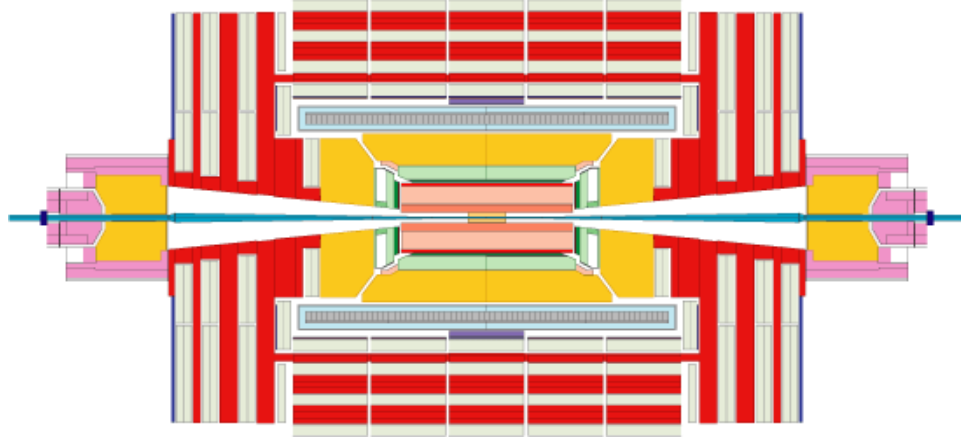


Figure 3.3: Cross-sectional view of the CMS detector.

differentiate electrons from photons: both are seen in the electromagnetic calorimeter, but only electrons leave traces in the tracker. Outside of the electromagnetic calorimeter is the hadronic calorimeter, which primarily detects hadrons. Hadrons are also detected in the electromagnetic calorimeter, but they leave the largest signature in the hadronic calorimeter, which differentiates the hadrons from the electromagnetic particles like electrons and photons. Hadrons are often found in large quantities called “jets”, so summing together the energy deposits from sizable electromagnetic and hadronic calorimeter regions is important in event reconstruction. Surrounding the tracker and the calorimeters is the solenoid, which generates the magnetic field. The charge of a particle can be determined from which way it bends in the magnetic field. Finally, outside the solenoid and interwoven with the panels of the iron return yoke, is the muon system. Muons leave traces in the inner subdetectors, but they are the only particles to live long enough to reach the muon system. This differentiates them from the other particles detected by the inner subdetectors. Since the muon system is outside the solenoid, the magnetic field points in the opposite direction, and hence the muon trajectories bend in the other direction from what they had inside the solenoid. The last category of end-product, neutrinos, generally cannot be detected. However, their direction and energy in the radial plane can be reconstructed by summing the energy of all the particles in the event. Because the initial protons have zero momentum in the radial

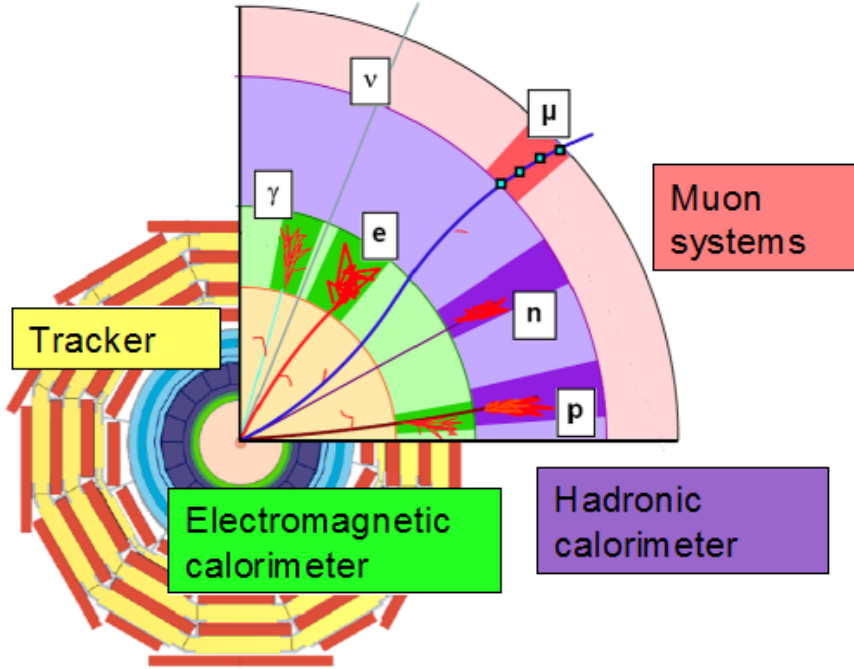


Figure 3.4: Detection of particles in the CMS detector.

plane, and momentum is always conserved, any significant “missing” component to the final energy can be taken to be due to a neutrino.

A CMS event display from 2010 running can be seen in Fig. 3.5.

3.3.2 Coordinate System

A standardized set of coordinates is used to describe points and directions within the CMS detector. The origin of the coordinate system is at the interaction point, with the x -direction pointing horizontally south towards the LHC center (ignoring the slight tilt of the LHC ring with respect to the vertical), and the y -direction pointing directly upwards. The z -axis points horizontally west along the beam direction, and the magnetic field inside the solenoid points in the positive z -direction. The azimuthal angle ϕ , measured in the x - y plane and ranging from $-\pi$ to $+\pi$ radians, is oriented such that $\phi = 0$ is equivalent to the

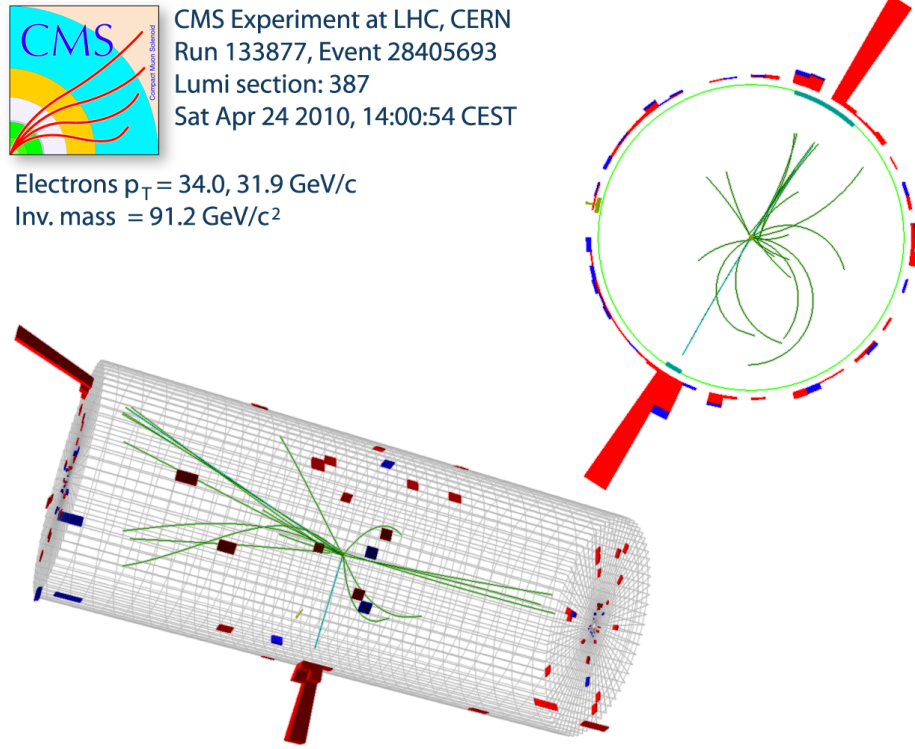


Figure 3.5: Three-dimensional display of a $Z \rightarrow e^+e^-$ event in the CMS detector.

positive x -axis and $\phi = \pi/2$ is equivalent to the positive y -axis. The longitudinal angle θ is measured from the positive z -axis, and the sign of $\eta = -\ln \tan(\theta/2)$ is equal to the sign of z .

3.3.3 Magnet

Since charged particles bend in a magnetic field, particle detectors use some sort of magnetic field to determine the charges of decay products. CMS uses a solenoid, or electromagnet in the shape of a coil, to produce a uniform magnetic field in the detector's inner region. The magnet coil is 12.5 m long with a 6 m diameter, and it weighs 220 metric tons. The CMS solenoid is made of a superconducting material, a niobium-titanium alloy, to handle the large amount of current necessary: it generates a field of 4 Tesla using a current of almost

20 kA in a four-layer winding.

The magnetic field is returned through a 10,000 metric ton iron yoke, which also serves as a support structure for the detector. The yoke consists of three concentric cylinders divided into the same 5-ring barrel and 3-disk endcap structure as mentioned previously.

3.3.4 Tracker

The purpose of the tracker is to detect the tracks from end-product particles; only those which are charged can be detected in the tracker. The CMS tracker contains 75 million channels and is specifically designed to get good performance using only a small number of hits per track. It is the innermost layer of the CMS detector and is made up of two different systems: the pixel detector, which is closest to the interaction point and hence has the finer granularity, and the strip tracker, covering the volume out to the calorimeter. Both systems are made of silicon and register charged particles in the same way. When a charged particle passes through the silicon, it knocks electrons out of the material, creating a net positive charge. The electric current needed to return the silicon to its neutral state is measured and amplified by the readout electronics. This readout technology is fast compared to the 25 ns bunch spacing of the LHC. A single track is reconstructed by stringing together the measurements from each of the layers.

Pixels

The silicon pixel detector has a very fine accuracy due to the high density of tracks near the interaction point. It consists of 65 million readout channels arranged on modules in several layers around the interaction point. In the barrel, the layers are cylindrical and situated at radii of 4, 7, and 11 cm, while each endcap has two disks, at 6 and 15 cm from the interaction point. Each module consists of a rectangular array of $100\text{ }\mu\text{m} \times 150\text{ }\mu\text{m}$ pixels, with a resolution of $15\text{ }\mu\text{m}$. The pixels on each module are read out by a dedicated chip attached to the module. The chip contains readout unit cells, one corresponding to each pixel, which are connected to the pixels with a small bump of solder in the so-called bump-bonding method.



Figure 3.6: Structure of pixel detector layers.

Strips

The silicon strip detector consists of 10 million channels arranged in 10 layers of strips, in both the barrel and the endcap, and extends to a radius of 130 cm from the interaction point and to an $|\eta|$ value of 2.5. The track density at this stage is much less than in the pixel detector, so the granularity does not need to be as fine. Each strip is between about 12 and 16 cm long, with a resolution ranging from 15 μm in the inner barrel to 50 μm in the outer tracker. The strips are arranged in modules of 6 inches in length, with the geometry of each module dependent on its position in the tracker. The strip tracker consists of four different parts: the tracker inner barrel (TIB) with four layers of strips reaching to a radius of 50 cm, the tracker inner endcap disks (TID) with three layers to ± 90 cm in z , the tracker outer barrel (TOB) with six layers to 1.16 m radially, and the tracker outer endcap (TEC) with nine layers, extending to ± 2.8 m in z . Several of the layers are double-sided: they contain two sets of silicon modules. The two sets of modules are arranged in a stereo geometry to accurately measure the longitudinal (or radial, in the endcap) coordinate of the hit as well as the azimuthal.



Figure 3.7: Diagram of layers in tracker barrel.

3.3.5 Calorimeters

The general purpose of a calorimeter is to measure energy passing through it. CMS measures the energy of decay products with a scintillating electromagnetic calorimeter (ECAL), situated just outside the tracker, and a sampling hadronic calorimeter (HCAL), surrounding the ECAL. Both calorimeters are contained within the solenoid. This solves a problem seen in other detector designs, in which energy lost in the magnet material leads to an uncertainty in the energy measurement.

ECAL

The ECAL is made up of 76,000 lead tungstate crystals, arranged in a cylindrical barrel and disk-shaped endcap geometry. The barrel is divided into 18 sections in ϕ and two in η , called supermodules. Each endcap is divided into two half-disks, called dees. Each lead tungstate crystal is 2.2 cm x 2.2 cm wide and 23 cm long for the barrel, and 3 cm x 3 cm wide and 22 cm long in the endcap. The dimensions were chosen such that the width of each crystal is approximately the characteristic size of an electromagnetic shower within it (Moliere radius of 2.19 cm), and the length is 25 times the radiation length of the material (0.89 cm for lead tungstate). The crystals are oriented with the long axis directed towards the interaction point. However, to avoid particles being lost in the cracks between the crystals, the crystals are offset by 3 degrees in both the ϕ and η directions. An electromagnetically-interacting particle will induce a shower inside the crystals, and the resulting light is captured by photodetectors situated at the back of each crystal. The energy resolution was measured to be

$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{2.8\%}{\sqrt{E}}\right)^2 + \left(\frac{0.12}{E}\right)^2 + (0.30\%)^2$$

A preshower detector, consisting of lead radiators and silicon sensors, is situated on the face of each endcap. The preshower serves to identify electron pairs due to low-energy pion decays and differentiate them from more interesting signatures. Pions typically decay on the scale of the distance between the interaction point and the ECAL endcap; therefore there is no preshower detector for the barrel, which is too close to the interaction point.

HCAL

The hadronic calorimeter surrounds the ECAL and covers $|\eta| < 3$ in the barrel and endcap (HB/HE) and up to $|\eta| < 5$ in the forward calorimeter (HF), both within the magnet coil. Additionally, the outer calorimeter (HO) is situated outside the solenoid and serves to detect and measure any leak-through. In general HCAL uses sampling calorimeter technology, with brass and scintillator layers interleaved in the barrel and endcap, and steel plates and quartz fibers for HF; quartz was used in the forward region because of its ability to withstand very

high particle fluxes. In a sampling calorimeter, the layers of metal absorber induce particle showers, which are then “sampled” by the layers of scintillator attached to photodetectors. Not all of the energy is directly detected, some being lost in the metal, so the energy measurement is scaled using experimentally-determined factors. The depth of the HCAL in terms of interaction length varies between $5.82 \lambda_I$ at $\eta = 0$ and $10.6 \lambda_I$ at the edge of HB and throughout the endcap, with the ECAL adding $1.1 \lambda_I$. The HO extends the HCAL depth to at least $11.8 \lambda_I$ in the barrel, with the magnet material acting as the absorber for the HO scintillator layer. (An additional layer of iron and a second scintillator exist at the $\eta = 0$ ring, where the HB depth is the least.) Since there is no ECAL in the forward region, the HF is equipped to distinguish electromagnetic objects from hadronic jets with different lengths of quartz fibers. The long fibers reach the entire depth of HF, while the short fibers only start halfway through the detector. Since electromagnetic showers tend to occur in the first half of the detector, and hadronic showers throughout the full HF, a shower detected only by the long fibers is likely to be electromagnetic in nature.

3.3.6 Muon System

The CMS muon system serves to identify muons, which are the only particles that leave a signature very far from the interaction point, and also to measure their momentum. The system is situated outside the magnet coil, within the iron return yoke, and provides coverage up to $|\eta| < 2.4$. The return yoke also functions as an absorber to filter out backgrounds to the muon signal. The muon system consists of three separate subsystems: drift tube chambers (DTs) in the barrel up to $|\eta| < 1.2$, cathode strip chambers (CSCs) in the endcap with an η range $0.9 < |\eta| < 2.4$, and resistive plate chambers (RPCs) in the region $|\eta| < 1.6$.

The DTs are used in the barrel due to relatively low muon rates in this area, in addition to a uniform magnetic field in the return yoke. Four stations of chambers are interspersed with the iron frame to form concentric cylinders around the beam pipe, with a total of 172,000 detection wires. Orthogonal positioning of the wires within each chamber provides measurement of both azimuthal and longitudinal position. Each wire detects charge that drifts toward it due to an applied voltage when a muon ionizes the surrounding gas, as

illustrated in Figure 3.8. Neighboring chambers overlap to ensure full coverage.

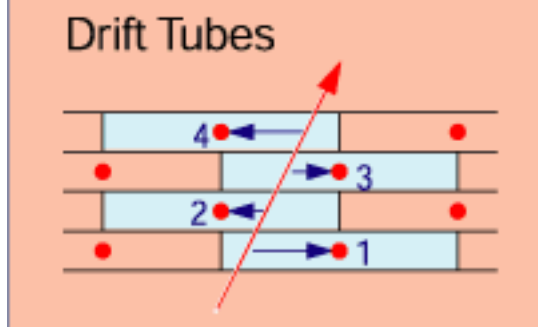


Figure 3.8: Conceptual diagram of drift tube chambers.

The CSCs are used in the endcap, where the muon rate and overall particle flux are higher. Cathode strips in each chamber measure the ϕ -position of a hit, while anode wires oriented perpedicularly measure the η -position. There are on the order of 200,000 readout channels for each type of measurement. Four stations of CSCs are used in each endcap; each muon in the range $1.2 < |\eta| < 2.4$ crosses either 3 or 4 stations. For $|\eta|$ between 0.9 and 1.2, muons are detected both by the CSCs in the endcap and the DTs in the barrel. Coverage is provided by overlapping layers of chambers at each station.

The RPCs provide redundancy for the previous two systems as well as a faster response time in place of a highly accurate position measurement. The RPCs measure the timing of a hit with a 1 ns resolution, compared to the 25 ns LHC bunch spacing and hundreds of ns drift time in the drift chambers. Six layers of RPCs are interspersed with the drift tubes in the barrel, while three layers are used in the endcap with the CSCs. Again, overlapping chambers ensures full coverage. Each chamber consists of two or three double-gap modules, each of which has a single readout module between two thin gas chambers. The readout module consists of metal strips, which capture the electric charge created when a muon ionizes the gas, causing an avalanche of charge. Figure 3.10 shows a diagram of the structure.

Overall, the muon system has a momentum resolution of 9% for relatively moderate- p_T ,



Figure 3.9: Diagram of cathode strip chamber, showing the cathode strips and anode wires for measuring the ϕ and η coordinates, respectively.

low- η muons, and ranging from 15% to 40% for very-high- p_T muons. When combined with the information from the inner tracker, however, this resolution improves to 1% and 5%, respectively.

3.3.7 Trigger

The purpose of the CMS trigger is to identify potentially interesting events. Storing the data from every collision is unfeasible; resources allow only a very small fraction of events to be kept. Therefore the trigger reduces the effective event rate, keeping only the events with an interesting signature, for example a large energy deposit. There are two levels to the CMS trigger. The Level-1 Trigger uses hardware-implemented algorithms to reduce the event rate from the LHC collision frequency of 40 MHz to a maximum of 100 kHz; this must be done quickly since the initial event rate is high. The High-Level Trigger reduces the rate further to 200 Hz, using a computer farm to process each event in more detail; having a lower input rate, it can afford more processing time per event.

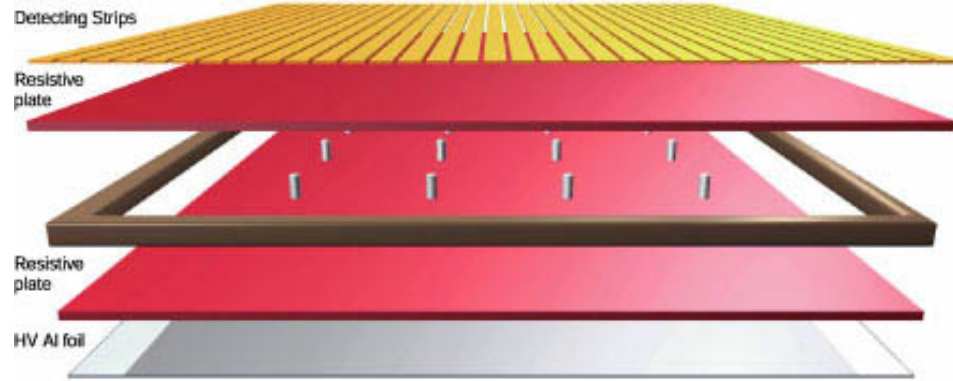


Figure 3.10: Diagram of resistive plate chamber.

Level-1 Trigger

The Level-1 trigger (L1) uses mostly programmable, hardware-based algorithms to identify and roughly reconstruct possibly interesting physics objects, such as electrons and muons. This is done within $3 \mu\text{s}$. The structure of the L1 is shown in Figure 3.11. Parallel chains process the muon system and calorimeter objects separately. Tracking information is not included in the L1, which means the L1 algorithms do not distinguish between electrons and photons. The first step of the Level-1 trigger is the generation of coarse versions of the readout information from each of the muon and calorimeter channels; this information is called trigger primitives. The trigger primitives are passed to regional trigger systems, which combine them into trigger objects. The objects for each region are then passed to a global system, which picks out the highest-energy (and therefore most interesting) objects of each type. The final set of objects gets passed on to the last level, the global trigger, which combines information from both the calorimeter and muon chains and makes a single decision based on the set of criteria implemented.

The trigger system must combine all the pieces of information according to which bunch crossing they originate from. This is complicated by the fact that the time it takes particles to reach the outer edge of the muon system is about that of the bunch crossing interval. In addition, the cable lengths between the subdetectors and the trigger subsystems and within

the trigger are all different, and each of the trigger systems has a different processing time. Despite these challenges, the timing of each piece of the L1 trigger has successfully been tuned to keep everything running synchronously.

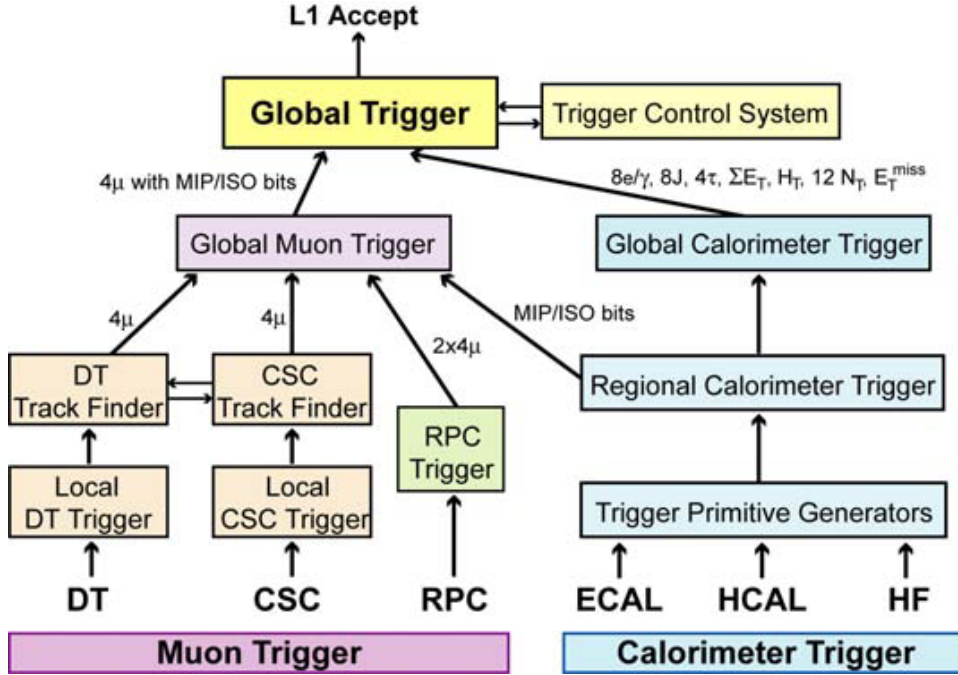


Figure 3.11: Structure of the Level-1 trigger.

Regional Calorimeter Trigger

The Regional Calorimeter Trigger (RCT) was designed and built and is maintained by the University of Wisconsin at Madison. It sits in eighteen crates in nine racks in the CMS underground electronics room. Each crate deals with one slice of η - ϕ space. The general purpose of the RCT is to take energy information from ECAL and HCAL and combine it to generate electron/photon candidates as well as energy sums for jet-finding. The information from the calorimeters arrives at the level of trigger towers, a larger granularity than that sent to the full readout. Each ECAL tower corresponds to an HCAL tower behind it. For each ECAL+HCAL tower, the RCT uses lookuptables to calculate the total energy sum as

well as the ratio of HCAL energy to ECAL energy. If that ratio is higher than a certain value, too much energy has been deposited in the HCAL tower for it to contain a probable electron or photon. The RCT also calculates the total energy sums of square regions of the calorimeter, 4 trigger towers to a side, and sends these energy sums along with a list of possible electron or photons to the global calorimeter trigger level. A more detailed treatment of the electron/photon-finding algorithm exists in a later chapter.

High-Level Trigger

The High-Level Trigger (HLT) is implemented on a computer farm in the CMS above-ground control room building. It serves to reduce the L1 output event rate from 100 kHz to a rate suitable for transfer and storage to disk, around 200 Hz. An event must pass the HLT to be analyzed offline. The input rate at this stage is lower than that at the Level-1, so more time can be afforded to spend on the decision, and more detailed reconstruction can be done. The HLT uses versions of the standard offline physics-object reconstruction algorithms optimized for fast performance. The algorithms need to be more accurate than those of the L1, but absolute accuracy at the offline reconstruction level is not necessary. Still, the HLT algorithms are kept as close to the offline ones as possible. Specific algorithms are dealt with in more detail in a later chapter.

Trigger Menus

Both the L1 and High-Level Triggers use the concept of a trigger menu. There are different sets of criteria defining an “interesting” event at any one time; each set of criteria is called a trigger path. When an event fulfills all the criteria for a given path, that path “fires” and the event is accepted. Some paths may fire at a higher rate than desired, so a prescaling factor is implemented: only a fraction of events firing that path is kept. For example, a path with a prescale of 5 only has 1/5 of its passing events actually saved. In this way, events passing a certain path can still be studied without taking too much of the available resources. The set of paths that are being checked at any given time, along with their prescale factors, is called the trigger menu. The Level-1 and High-Level Triggers each have their own trigger

menus. Since the L1 must do things quickly, its criteria are simple, whereas the HLT may have more complex criteria. Passing a given L1 path causes related HLT paths to be run on that event.

At the beginning of collision running, the event rate was low enough to allow very loose trigger menus to be used. At the HLT, a “keep-everything” pass-through menu recorded essentially all activity. As the luminosity increased, however, more stringent criteria had to be applied to maintain an acceptable event rate, for example, raising the amount of energy needed to accept an “interesting” object. The successive algorithms have been studied in detail and commissioned successfully throughout the 2010 running conditions.

3.3.8 Luminosity

CMS measures luminosity with the forward calorimeter, which sits between $|\eta|$ of 3 and 5, close to the beam pipe. Two different methods of measurement are used. The first method counts the number of occupied towers in the HF (above some energy threshold to avoid noise), using the principle that that number is proportional to the luminosity. However, at high luminosity the number of occupied towers may begin to saturate, making this method less accurate. The second method does not saturate in this way: it measures the total energy deposited in the HF, which is also proportional to the luminosity. The information from both of these methods is combined into one luminosity value.

Fig. 3.12 shows the total integrated luminosity at CMS as a function of time for the 2010 running period. The amount of data validated for physics analysis was 36.1 pb^{-1} .



Figure 3.12: Luminosity collected by CMS as a function of time, 2010.

Chapter 4

EVENT SIMULATION

Within a high-energy physics experiment's role of making discoveries and measurements, it is necessary to know ahead of time what exactly should be expected. What will the detector signature of a new process look like? How will we know whether we've seen something unexpected? This is where event simulation contributes. Essentially, a series of programs is used to carefully generate and calculate all the relevant quantities for a set of fake events. This information is then used to aid analysis in ways for which real data alone cannot suffice.

Having the ability to simulate physics processes serves multiple purposes. It aids in detector design: knowing the expected typical characteristics of particle interactions is essential to design a detector suited for those interactions. Once the detector design is settled, simulation of the detector is useful to design the algorithms used to reconstruct particles from their signature interactions with the detector material. In addition, simulating the physics processes can give estimates of how many events of a particular type are expected, further aiding the design process. The event simulation also contributes directly to many analyses, in the way of calculating the acceptance, the fraction of events that can theoretically be detected (see Section ??): it is impossible to know from observation how many events are missed by the detector, because of the very fact that they are unseen. It is instead necessary to get that fraction from a framework in which the characteristics of all events are inherently known, not reconstructed. Finally, simulated data is directly compared with "real" data from the detector to interpret the real-data results. If the data shows something significantly different from the simulation, then something is missing: perhaps a calibration needs to be applied, or the response of a detector unit needs to be further understood. Or, perhaps, there is a new physics process appearing for the first time, which was not previously known and which was therefore not present in the simulation. In this case simulation of proposed new physics processes may narrow down the identity of the observed new process.

Whatever the case, discrepancy between observed data and the event simulation indicates that further investigation is necessary.

The entire detection process is simulated, including the protons' direct interaction and any subsequent particle decays, as well as how the end-product particles interact with the detector as they pass through and how the detector itself responds. This latter part includes not only the material of the individual subdetectors but also the algorithms of the Level-1 Trigger, which are implemented in hardware (see Section 3.3.7). The High-Level Trigger does not need to be simulated in this way; since its algorithms are all software-based, the same code can be run without modification on both real data and simulated data.

4.1 Monte Carlo Event Generation

“Event generation” refers to creating a set of fake physics events, usually realistic, for purposes of study and comparison with real data. The term “Monte Carlo” is connected with this process because at the most fundamental level it is based on random numbers. All physics interactions happen in terms of probabilities: process X is more likely to happen than process Y, so process X is observed more often. For any given fake event, which physics process is actually simulated is randomly determined according to those probabilities. The kinematics of the event (energies and directions of involved particles) are also assigned according to probabilities. The final result is that having enough events will reproduce the original probabilities of each interaction. (It is worth noting that in terms of software, nothing is really random; in general, programs to create “random numbers” are really only pseudorandom, because computers by nature do things according to programmed rules. However, for the purpose of event generation, pseudorandom numbers are suitable if they are sufficiently random-like. Moreover, using a deterministic pseudorandom generation algorithm aids in debugging – the results of the algorithm are reproducible, which cannot happen with truly random numbers.)

4.1.1 Types of Monte Carlo Simulation Programs

There are two sorts of programs that can be called “Monte Carlo event simulators”. They both use the matrix elements derived from the Feynman diagrams of the physics interactions

of interest (explained in Section ??), but they use the matrix element formalism in different ways to achieve different ends. As a reminder, the equation to obtain the differential cross section element from the matrix element for a given diagram is

$$d\sigma = \frac{1}{\hat{s}} |\mathcal{M}|^2 \frac{d\cos\theta d\phi}{8(2\pi)^2}$$

The first type of Monte Carlo program outputs a single quantity or set of quantities characteristic of the given process, such as the cross section. These programs are called “cross section integrators”. They calculate $d\sigma$ for each of a uniform distribution of phase space values (i.e. uniform increments of $\cos\theta$ and ϕ). The value of $d\sigma$ represents each event’s “weight”; the total cross section σ is then obtained by integrating over the full collection of $d\sigma$ values. The final result is a value for the cross section; hence the name “cross section integrator”. However, since the events themselves are distributed uniformly in $\cos\theta$ and ϕ , their characteristics do not represent actual event distributions (real event distributions in data have non-uniform distributions, in $\cos\theta$ in particular). Therefore they can only be used to calculate a few select quantities; they can’t be used to predict distributions.

The second type of program are called “event generators”. They also use the matrix element to calculate the differential cross section element with the equation given above. However, instead of using $d\sigma$ as an event weight, it “unweights” the events to obtain the physical distributions. It does this through the so-called acceptance-rejection method: first the maximum possible value of $d\sigma$ is calculated, $d\sigma_{max}$. Each event’s individual $d\sigma$ value, the ratio $\frac{d\sigma}{d\sigma_{max}}$ is taken; this number is necessarily a fraction between 0 and 1. A random number g between 0 and 1 is then generated. If the ratio $\frac{d\sigma}{d\sigma_{max}}$ is greater than g , then the event is kept; otherwise, it is rejected. This method causes the generated events to have the properties and distributions that real events would have – sections of phase space with a higher “event weight” are more likely to be represented.

However, using a single Feynman diagram is not typically sufficient to make very accurate calculations. In general, there are significant contributions from the higher-order diagrams (diagrams with more legs and loops, but still representing the same final arrangement of particles). These corrections must be taken into account. In addition, the matrix elements

and Feynman diagrams for these calculations only consider bare quarks, that is, quarks extracted from the framework of hadrons such as the proton. These bare quarks cannot exist in reality, so a method of accounting for the quarks' presence inside hadrons must also be used. There have been two different paths taken to deal with the above issues. In the first, the “matrix element” method, higher-order diagrams are explicitly included in the matrix element calculation. Since the higher the order, the more complex the calculation, only diagrams of up to a given order are generally used. In one implementation, more “legs” can be dealt with, but diagrams containing virtual loops are ignored. In the other, all diagrams, including those with virtual loops, of a single order are used, but at the moment there exist satisfactory solutions only for next-to-leading-order diagrams (or NLO, explained in Section ??). The matrix element method does not explicitly take care of the bare-quark problem; this can be handled in multiple ways. Some cross section integrators deal with the interactions of quarks inside hadrons using Feynman's factorization theorem. Programs which do not deal with them in this way can be interfaced with programs following the second of the two paths, the “parton shower” method, which is the method used by many event generators (as opposed to cross section integrators). In the parton shower method, instead of including diagrams representing higher orders and quarks-within-hadrons into the matrix element calculation, this part of the interaction is taken care of by a separate routine. This routine takes the products of the basic tree-level interaction and decays them according to probabilities and rules coded into the program. The process is iterative, repeating for any quarks or gluons that branch off at any point. In this way the parton shower method is more flexible than the matrix element method – it can deal with many branchings and legs. However, it is also less precise formally, since it avoids the use of the matrix elements from the Feynman diagrams.

The philosophy of parton-shower generators in particular is that the entire process of simulating a physics event can be done in a modular fashion. Fig. 4.1 shows the typical steps necessary to simulate an event. The incoming protons are modeled in terms of the parton distribution functions (Section ??). The “hard subprocess” refers to the main physics interaction. Not shown in the diagram are other interactions involving different parts of the protons or other protons in the same bunch; these interactions, known as the “underlying

event,” are often not as interesting physics-wise because they are typically not as energetic. The participant particles in the main interaction are then “evolved” through the showering process, which includes emissions of photons or gluons by the initial- or final-state particles, as well as “branchings” of final particles. Any resulting quarks and gluons are formed into “colorless” hadrons, which may then decay into other particles. These final particles are the ones seen in the detector. The fact that the full process can be separated into these steps makes it easier for different programs to handle individual steps. For example, one program may simulate the hard process, while another is better-equipped to take care of the showering and hadronization, and a third is specialized in the decay of a particular particle. Taking the theme of modularity even further, the set of fully-decayed particles is then plugged into a program simulating the passage of the particles through the detector and the detector’s response (Section 4.2).

The process of hadronization is not directly relevant to this analysis, since it does not require hadrons to exist in the final state. However, since initial- or final-state radiation may cause jets incidental to the main interaction that nevertheless affect its kinematics, hadronization is dealt with here. In addition, it simplifies things to treat electrons and photons in the final state as “partons”, in the same category as the quarks and gluons. Even though they are not constituents of the proton, they can be modeled as having “distribution functions” and subject to “showers” of photons and electrons. Therefore different sorts of processes can be plugged into the same machinery and treated the same way.

4.1.2 Monte Carlo Generator Programs

Several Monte Carlo programs were used to fulfill different roles in this analysis. Two programs were used to generate the simulation events used: one general-purpose generator for background events, and a more specialized generator for the signal sample. A few of the background samples used a dedicated decay program to decay specific particles in the end state. Finally, a cross section integrator was used to obtain a highly precise estimate of the theoretical value of the interaction cross section. These programs will be detailed in the following sections.

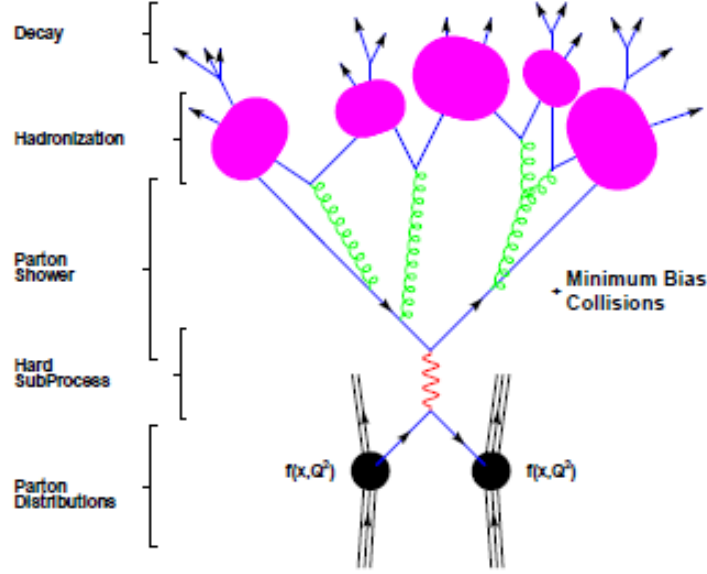


Figure 4.1: Diagram of typical steps used to simulate event in parton shower method. The incoming protons are modeled in terms of the parton distribution functions. The “hard subprocess” refers to the main physics interaction. The participant particles in the main interaction are then “evolved” through the showering process, which includes emissions of photons or gluons by the initial or final particles, as well as “branchings” of final particles. Any resulting quarks and gluons are formed into “colorless” hadrons, which may then decay into other particles. These final particles are the ones seen in the detector.

PYTHIA

CMS makes use of one of the standard Monte Carlo event generation workhorses, a program known as PYTHIA. PYTHIA is a general-purpose event generator using the parton-shower approach to corrections. It encompasses most of the steps necessary to generate an event: the parton distribution functions that describe the colliding protons, the main physics interaction process, the non-interacting remnants (underlying event) and how the end products shower, hadronize, and decay. Over its long history PYTHIA has built up a large collection of tools, which include many main physics processes and various methods of dealing with each of the steps mentioned above. PYTHIA also includes many non-standard-model physics processes in order to study new physics theories. Most of what it does can be

adjusted by the user via changing parameters; however, its default values are chosen to produce valid output without the need to necessarily adjust settings. Because PYTHIA is such a wide-ranging program, it does not treat everything it handles in depth. For this reason, specialized programs are sometime used to work with specific areas of the simulation process, including the ones in the following sections. PYTHIA was used to generate the background samples used in this analysis.

This analysis does not directly deal with quarks or jets; however, since jets can be present in the final state and can play a role in the kinematics of the $Z \rightarrow e^+e^-$ interaction, the method of dealing with hadronization is relevant here. PYTHIA by default uses the Lund string model (illustrated in Fig. 4.2) to evolve quarks and gluons through the process of hadronization and formation into jets. Conceptually, an interaction’s final state may have quarks “flying off in different directions”. However, quarks cannot exist alone: they must always exist with other quarks, held together by the strong force. If the individual quarks do not have very much energy as they move apart, the increasing energy of their interaction will hold them together. However, if the quarks have much energy, the interaction energy will not be enough to hold them together, and they will need to form new “quark partners” in order to remain in a bound state with other quarks. According to the Lund string model, the energy between opposite color charges increases approximately linearly with distance. The energy is concentrated into a linear area, like a tube or string. These strings are always stretched between two quarks. Enough energy on the part of the quarks will “break” the string and create a new quark-antiquark pair, which then fills the need for “quark partners”, as illustrated in Fig 4.2b. The process continues until the quarks do not have enough energy to overcome the force holding them together.

The underlying event can be represented in PYTHIA by different configurations of parameters, known as “tunes.” The tunes have been chosen to correctly model already-existing distributions from previous experiments. This analysis uses samples generated according to the default tune used in CMS, Tune Z2. However, differences between the tunes are very slight and affect final results very little; these affects are accounted for in the systematic errors (Section 7.4.4).

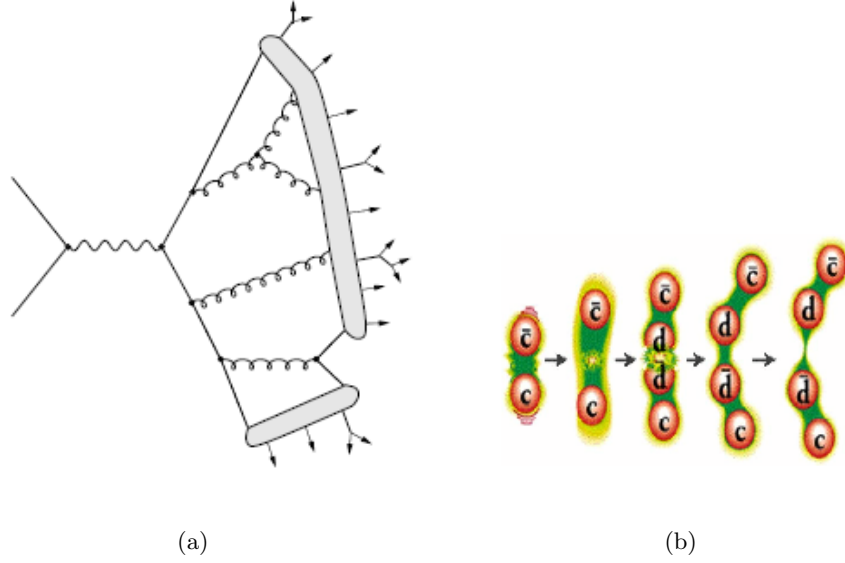


Figure 4.2: Schematics showing the Lund string model of hadronization. (a) In the Lund string model, the quarks are joined by strings which “break” to form new quark-antiquark pairs. The process continues until the quarks’ energies are not enough to overcome the force holding them together. The gray areas represent chains of quarks joined by such strings. (b) An illustration of a possible creation of a quark-antiquark pair.

POWHEG

POWHEG (for “POsitive Weight Hardest Emission Generator”) is a method for making next-to-leading order (NLO) corrections within a parton shower framework. At low energies, the showering algorithms implemented in parton shower generators (such as PYTHIA) model the corrections well. However, at higher energies, a next-to-leading order matrix calculation is more accurate. An ideal solution combines the two approaches to get the best of each. Since POWHEG does this, it therefore produces a more accurate picture of the radiation corrections and their effects on final-state kinematic quantities when compared to a standard parton-shower generator, which only includes some (the real part) of these corrections. POWHEG consists of a number of routines representing various interactions, including $Z \rightarrow e^+e^-$. Processes that are not already included in the set may be implemented by the user.

POWHEG was used to generate the hard process for each event of the $Z \rightarrow e^+e^-$ signal sample; this part of the interaction was then fed through PYTHIA (by the modular procedure previously described) to undergo hadronization. It was more important for the signal sample to be accurate to higher order than for the background samples; the background samples are only of interest in terms of how much they may contaminate the signal sample. Slight variations in their distributions due to missing higher-order corrections are therefore not relevant.

Other Generators

Two other Monte Carlo event generators played roles in the analysis. In the first case, particles in some of the background samples called for a specialized decay package. In the second, a precise theoretical prediction was needed to compare with data, and the standard programs used to generate the events cannot reach the necessary precision.

Many particle decays are handled by the primary event generator, PYTHIA. However, there are several aspects of decays that PYTHIA does not deal with, such as spin polarization. Because these aspects can have small effects on the kinematic distributions of the decayed particles, CMS decided to use a separate package, TAUOLA, to decay tau particles in the samples that included taus. TAUOLA specifically takes spin polarization into account, and includes first order QED corrections. The TAUOLA package was only necessary for the $Z/\gamma^* \rightarrow \tau^+\tau^-$ and $W \rightarrow \tau\nu$ background samples.

The theoretical prediction for the cross section value was obtained with FEWZ, a cross section integration program specifically dealing with production of the W and Z. Previously, all calculations of the cross section had been done at next-to-leading order (NLO) in QCD, which entails several percent uncertainty. However, since the statistical error on W and Z results at the LHC was quickly expected to reach the order of 1%, it was deemed necessary to undertake a calculation at next-to-next-to-leading order (NNLO), which gives a theoretical uncertainty of approximately the same value. (In essence, all higher-order corrections beyond NNLO contribute about a percent or less of uncertainty.) FEWZ was chosen to calculate the cross sections for the CMS analysis because of this treatment of NNLO QCD

corrections.

4.2 *Detector Simulation*

After the generation of the main physics process, and the parton showering and hadronization and decays, the role of the detector itself must be taken into account. The end-state, decayed particles interact with the detector, which causes signals representing the particle's position and energy to be recorded as data. These must be properly modeled to know, for example, what sort of signal would result from the passage of a particle with a given momentum. In addition, the detector itself is not completely passive, in that it can change the properties of the particles passing through it during the course of measurement. Both of these roles necessitate a detailed simulation of the detector and its response.

4.2.1 *GEANT Detector Model*

The CMS collaboration uses the software package GEANT4 to model the detector within the framework of CMSSW, the CMS-specific analysis software (see Section ??). GEANT4 provides a toolkit designed to accurately represent all facets of the detector simulation. These include the ability to construct a precise geometrical model of the detector itself, complete with any actual misalignments and a model of the detector's magnetic field, as well as the physical properties of the materials used. In addition, the package also handles the properties of the fundamental particles involved, how they behave in the modeled magnetic field, and how they interact with the different materials present in the detector. For example, the detection of particles in the electromagnetic calorimeter depends on those particles causing a shower of other particles upon interacting with the material of the calorimeter; this must be modeled. Finally, GEANT4 also provides the ability to model the response of the electronics used to detect the particles, for example the response of a photomultiplier tube to a light signal caused by the aforementioned shower.

4.2.2 Level-1 Trigger Emulator

In addition to modeling the detector components that do the actual detecting, it is necessary to include the response of the trigger system (Section 3.3.7), serving multiple purposes. Modeling the trigger system enables the estimation of event rates for a given luminosity before the experiment actually runs with that luminosity. In particular, this permitted the estimation of rates long before the detector was fully constructed, greatly aiding the experiment’s physics planning. In addition, comparison of the real-time “online” performance of the trigger with its software counterpart can indicate and aid in diagnosing problems with the online running.

The High-Level part of the trigger system (HLT), consisting of algorithms implemented in software, is simply run “offline” as an element of the detector simulation. However, the Level-1 Trigger (L1), being implemented in electronics instead of software, must be explicitly modeled. The software package that does this is called the Level-1 Trigger Emulator, since it aims to be a bit-by-bit emulation of the trigger hardware instead of just a simulation of the algorithms implemented. The L1 Emulator is modular in the same sense as the hardware system; information is passed between separate packages representing the different subsystems of the L1. Each element of the hardware, namely each type of processing card, has its own software class to handle its own specific functions. The software therefore aims to model each of the trigger’s processing steps in accurate detail, so that the overall result is identical.

Chapter 5

EVENT RECONSTRUCTION

The information from the detector for any given event gets read out over millions of channels from the many different subdetectors. These signals must be combined to provide meaningful physics information about the interaction that took place within the detector, namely the measurable quantities (like direction and energy) of the outgoing decay products.

5.1 *Detector Object Reconstruction*

The first step of the particle reconstruction involves translating the information from each detector channel, such as position and electronics signal, into useful quantities such as position in η - ϕ space and energy or momentum. This is done by algorithms mapping each individual detector cell to its η - ϕ placement and calculating its energy, which necessarily takes into account the position of each cell. The second step involves connecting information from different channels to build up a more complex picture of the particles that passed through the detector.

5.1.1 Electromagnetic Calorimeter Reconstruction

The information coming directly from the ECAL is read out in terms of each cell's electronics response. The ECAL reconstruction algorithms translate this into an energy value, which is then calibrated according to the the cell energy as well as its η -position. The algorithms also calculate quantities relevant to the quality of the deposit, in particular whether or not the deposit happened at a time consistent with the LHC bunch collisions, i.e. if it was “out of time”. This information is especially useful in identifying apparent energy deposits that have been found to come from the electronics themselves; these fake deposits were discovered during commissioning and are known as “ECAL spikes”. Since the spikes can imitate energy deposits from actual decay products, they must be removed from the

information used to reconstruct electrons and other particles. They can be distinguished from real particle deposits due to their timing as well as the fact that the surrounding crystals have no trace of a deposit. Therefore, if a high-energy deposit has no or very little energy in its neighboring crystals, or if its timing is not consistent with a bunch crossing, it is considered a spike and taken out of the reconstruction.

5.1.2 Track Reconstruction

A track is a sequence of hits in the tracker, reconstructed to trace the trajectory of a passing particle. The Kalman Filter algorithm used by default for CMS tracks, but not good at catching abrupt changes in electron trajectory due to bremsstrahlung. Therefore, GSF algorithm used for electrons – able to find those hits in unexpected directions.

Kalman Filter

The Kalman Filter (KF) tracking method is seeded by pairs of hits in the pixel detector. The seeding algorithm first searches for a hit in one of the outer layers of the pixel detector. If a hit is found, another hit is sought in a layer closer to the interaction point, looking in an η - ϕ window around the original hit. Only two out of the three pixel layers are required to have hits in order to maintain a high efficiency. Once a track seed is found, further hits are sought in each successive tracking layer. The Kalman filter method is used to predict possible locations for hits in the next layer based on the currently-known track parameters. Hits in that area are then sought and included in the track, combining each hit with the predicted location for that layer and weighting each value depending on the value's uncertainty. If there are multiple candidate hits in the next layer, a possible trajectory is calculated for each hit, along with a trajectory accounting for the case in which the hit in that layer was lost. This procedure is repeated for each layer: each of the calculated trajectories for a given layer is matched to hits in the next layer, for which new trajectories are calculated. To prevent the number of necessary calculations from growing too large, the number of possible trajectories at each step is capped at five. In the case of multiple possible trajectories for a given fully-reconstructed track, defined by two trajectories sharing more

than half of each of their hits, the track candidate with fewer hits (or, in the case of the same number of hits, the track with the worse fit) is discarded.

Gaussian Sum Filter

The Kalman Filter track-fitting does not accurately account for the behavior of particles in a material, specifically energy lost in the material and changes in direction due to radiation. The KF description is in particular inadequate for electrons, which readily interact with matter in these ways. The Gaussian Sum Filter (GSF) algorithm was therefore developed: essentially a Kalman Filter with a more sophisticated energy loss modeling. Instead of approximating the energy loss (which is non-Gaussian) with a Gaussian distribution, which KF implicitly does, the GSF method models it as a weighted sum of Gaussian distributions. This greatly improves the performance of the track-fitting for electrons.

5.2 Electron Reconstruction

As an electron passes through the CMS detector, it leaves a signature pattern in the sub-detectors. Characteristically, electrons produce a track in the tracker system, as well as an energy deposit in the electromagnetic calorimeter. Therefore an electron is reconstructed by matching tracks and energy clusters that appear to have been made by the same particle. There are currently two approaches used to initiate electron reconstruction: the “ECAL-driven” method and the “tracker-driven” method, named according to which subdetector’s information initiates the reconstruction. Each method produces a collection of basic electron candidate objects, each consisting of a calorimeter cluster and hits in the first few layers of the tracker. These collections are then combined into one and used as seeds to reconstruct the electron tracks, producing fully-reconstructed electrons.

5.2.1 ECAL-driven Electron Seeding

The ECAL-driven electron reconstruction method starts with an energy deposit (or “super-cluster”) in the ECAL and searches for hits in the pixel system that matches the position of the calorimeter deposit.

Superclusters

The calorimeter energy deposit used in the electron reconstruction is called a supercluster: a cluster of clusters. A single electron or photon deposits most of its energy in a 5×5 square of ECAL crystals, as shown in Fig. 5.1. However, an electron passing through the tracker material in a strong magnetic field produces bremsstrahlung and electrons from converted photons, causing the original electron's energy to have spread out in ϕ once it reaches the calorimeter. The process of creating a supercluster gathers these energy deposits to form the initial energy of the electron.

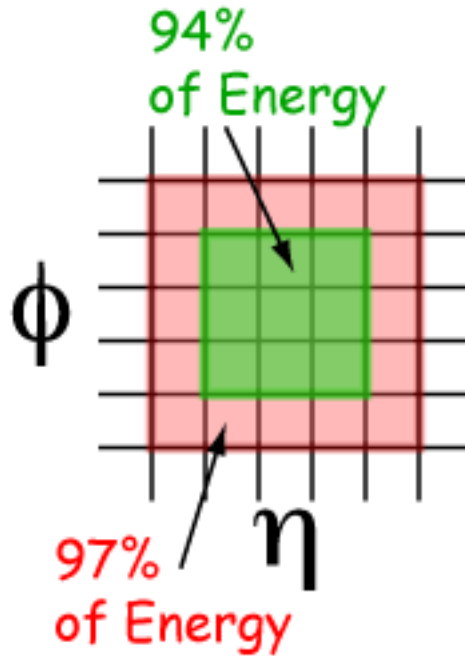


Figure 5.1: Typical energy distribution in a basic cluster.

The electron reconstruction currently uses two different superclustering methods, the “hybrid” supercluster method in the barrel, and the “multi5x5” clustering method in the endcap. The hybrid algorithm forms superclusters and basic clusters from the same energy deposit information, while the multi5x5 algorithm first forms a single collection of basic clusters, which are then separately combined into superclusters.

The hybrid algorithm starts with a list of all ECAL crystals with an energy greater than a threshold; these crystals are considered seeds. Beginning with the highest seed crystal, the area around each seed is searched for energy deposits in steps of one crystal in ϕ . For each ϕ step, a “domino” of crystals is formed in η : one crystal to each side of the central crystal, or, if the energy sum of those three crystals is greater than a threshold, two crystals to each side of the central crystal, for a total of five crystals per domino. The process is done for a given number of ϕ steps in both the positive and negative directions (currently 17 steps). As other seeds are encountered in this process, they are removed from the list of available seeds. Each domino with a higher energy than its two neighbors is considered as the seed for a basic cluster, and the collection of dominos formed by the search process is the basis for the supercluster. The basic clusters are formed by summing the energy of each local-maximum domino with those of its neighboring lower-energy dominos, excluding dominos that have already been used. The supercluster is then formed by summing the energies of the constituent basic clusters, with its seed considered to be the highest-energy basic cluster and its position given by the energy-weighted positions of the constituent clusters.

The multi5x5 algorithm first creates a collection of basic clusters from seeds, which are ECAL crystals with an energy higher than a given threshold. If a seed represents a local maximum, then its energy deposit is combined with those from the crystals in the surrounding 5x5 square to form a basic cluster, ignoring any crystals that have already been used to form the central 3x3 part of another cluster. The basic clusters are then used as seeds for the superclustering algorithm. For each seed cluster with energy higher than a threshold, the energies of any other clusters within an η - ϕ window of the seed (longer in ϕ than in η) are combined with its own energy to create the supercluster, leaving out clusters that have already been used to form another supercluster. The position of the supercluster is given by the energy-weighted position of the individual clusters.

Pixel Hit-matching

To reconstruct the full electron, the supercluster must be matched with a track. This is done by extrapolating the probable position of tracker hits from the supercluster’s position,

as illustrated in Fig. 5.2. Given the supercluster energy and the fact that a non-radiating electron would have hit the calorimeter at the energy-weighted center of the supercluster's energy spread, the electron's initial (pre-radiation) trajectory through the tracker can be calculated for both the positive and negative electron charge cases. The reconstruction algorithm then looks for a matching hit in the first pixel detector layer within a ϕ - z window of this position for each charge case. If a hit is found, another hit is sought in the second pixel layer within a different ϕ - z window of the hit in the first layer.

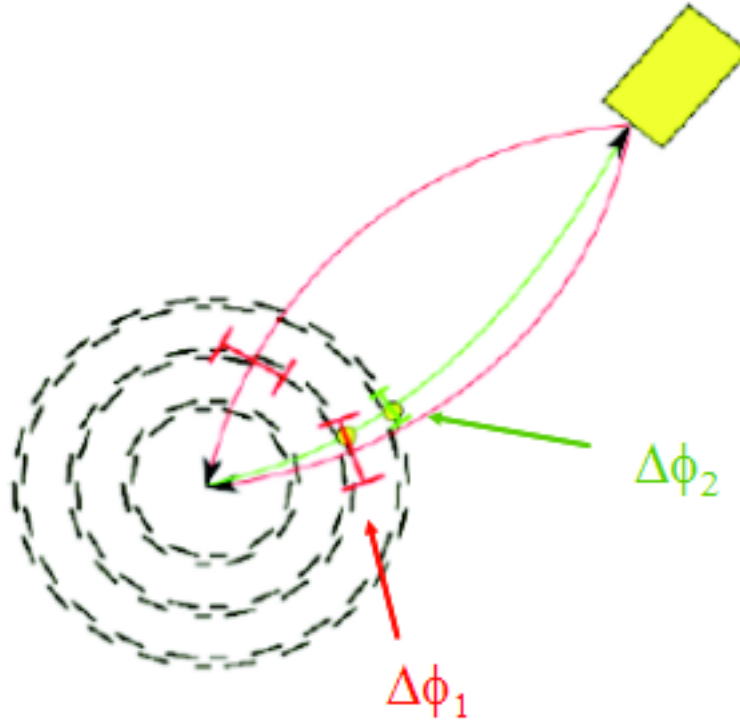


Figure 5.2: Schematic of pixel-hit search algorithm in electron-seeding. For both the positive and negative electron charge cases, a possible trajectory is extrapolated back from the calorimeter deposit given the deposit's energy. A hit is sought along either trajectory in either the inner or middle pixel layer within a window ($\Delta\phi_1$). If a hit is found, a second hit is sought along that trajectory in the next pixel layer(s) ($\Delta\phi_2$).

5.2.2 Tracker-driven Electron Seeding

In the tracker-driven electron seeding method, electron-finding is seeded by tracks, which are followed out to the ECAL to search for calorimeter clusters associated with each track. This method was developed to better deal with low- p_T electrons, for which the energy deposits from bremsstrahlung may be too widely separated to be matched, and non-isolated electrons within jets, whose energy deposits are not necessarily distinguishable from those of surrounding hadrons. For these electrons the calorimeter cluster-based seeding does not perform as well as for isolated, high- p_T electrons.

The tracker-driven electron seeding method takes as input the default collection of tracks, obtained by an iterative usage of the KF algorithm. The method first tries to follow each track all the way to the calorimeter and search for nearby energy clusters. All clusters with an acceptable E/p ratio with respect to the track are tested, and the cluster closest in η - ϕ space to the track is the one chosen.

Out of the tracks that were not well associated to any energy cluster, a preselection is applied to select those tracks that may have resulted from radiating electrons, which would not have been found by the cluster-matching method. If the track was not matched because it had too few tracker hits and could not be traced to the calorimeter, or if the KF algorithm did not fit the track well enough because of possible bremsstrahlung, the tracks are passed on to the next step of the selection. A reduced GSF track-fitting algorithm is then run on these tracks, in an attempt to recover the bremsstrahlung-induced changes in track direction. This reduced version of the GSF algorithm runs much faster than that with the default parameters, but the performance is slightly reduced. Finally, a multi-variate analysis using boosted decision trees is performed on these candidates. The MVA is trained on a sample of electrons from $b\bar{b}$ events, which are primarily non-isolated, and a sample of $Z \rightarrow e^+e^-$ events, which provides high- p_T electrons. The output of the MVA is a number indicating the likelihood of the object being an electron. Those whose MVA output is greater than a threshold are considered electron candidates. These candidates are then combined with those previously identified from the cluster-matching to produce the final collection of tracker-driven electron seeds.

5.2.3 *Electron Track Reconstruction*

The full collection of electron seeds, both ECAL- and tracker-driven, are used as the basis for reconstructing the electron track and hence completing the electron reconstruction. As mentioned previously, the electron reconstruction uses the GSF algorithm to reconstruct tracks. The electron kinematics are assigned to be a weighted combination of those of the constituent supercluster and track, and the final electron collection is saved for further analysis.

5.3 *Beam Spot Reconstruction*

The region where the beams overlap and protons may interact, the “luminous region”, is called the beam spot. The beam spot location can be used as an estimate for the initial position of tracks measured in the detector, and is therefore useful in track-seeding. Its position can be determined from measurements of tracks in the general track collection. This is done by making a distribution of each track’s distance of closest approach to the nominal beam spot $(0,0,0)$ as a function of the track’s ϕ angle at its closest approach. If the beam spot is far from $(0,0,0)$, the distribution looks sinusoidal. This distribution is then fit with a sine curve parametrized in the beam spot’s actual x - and y -positions, and these values are extracted.

5.4 *Primary Vertex Reconstruction*

Since protons may collide anywhere within the region of beam overlap, the main interaction in a given event may not necessarily happen at the exact center of the detector. In particular, there may be a significant displacement in the z -direction. Determining the location of the main interaction, also called the primary vertex, is an integral part of reconstructing the event. For example, an electron may leave a deposit in a given calorimeter cell that has a fixed η position relative to the nominal interaction point, the very center of the detector $(0,0,0)$. However, if the interaction producing the electron actually takes place at a different point, the electron’s actual η position will be different relative to the interaction. Therefore the position of the primary vertex should be taken into account when calculating η .

The process of reconstructing the primary vertex involves creating primary vertex candidates from the tracks in the event, matching each track to the interaction that likely produced it. This is done by grouping the tracks according to their z -position at the point at which they pass most closely to the beam spot. The tracks are required to pass relatively close to the beamline, and they are also required to have a p_T greater than some threshold (with a default of 1.5 GeV). Tracks within a small distance (default 1 mm) of each other at the beam spot are considered to have come from the same vertex, and the primary vertex candidates are formed from these groups of tracks. At this point a fit is applied to each primary vertex candidate, and tracks found to be incompatible with the candidate are discarded until all tracks meet a given compatibility level. Finally, vertices that are incompatible with the position of the beam spot are discarded. (A second algorithm uses the beam spot as constraint on the position of the vertex fit itself, but this algorithm was not fully commissioned at the time of this study.) If no reconstructed primary vertex candidates are found in an event, a “fake” primary vertex is added at the position of the beam spot. Since more than one interaction per event is possible but generally only one is of interest (the “primary” interaction), multiple candidates are ordered according to transverse energy of their constituent tracks, and the highest of these is taken to be the primary vertex.

Chapter 6

EVENT SELECTION

Most analyses, including this one, are only interested in particle interactions that happen relatively rarely during proton collisions. Therefore, methods have been developed to sort through all the events and select the interesting ones for further study. This selection takes place both “online”, in real time as the detector is running, and “offline”, after the previously-selected data has been stored. The online selection serves primarily for reducing the volume of data enough to feasibly store it, and so it is done centrally. In addition, the online selection is fairly general, while the offline selection is much more detailed and specific to individual analyses, and hence it is mostly done by the user of the data.

6.1 Online

The online selection consists mainly of the trigger system, described in technical detail in Section 3.3.7. The implementation of the trigger system’s algorithms that are relevant to this analysis is discussed in the following sections, including both the Level-1 and the High-Level Triggers. Both levels reconstruct possible physics objects and apply criteria to select the interesting events. In general, the processing at the Level-1 is quicker and coarser in order to sort through a larger volume of data, while the High-Level Trigger processing can afford to be longer and more thorough because it only deals with data not already discarded by the Level-1. In addition, as a last step, the information from the High-Level Trigger is used to group the data into primary datasets which are used as the basis for the offline selections.

6.1.1 Trigger Eras

Since the LHC began taking data the event rate has changed by several orders of magnitude, and the trigger menus have evolved accordingly to deal with the changes. Initially, the

collision rate was so low that minimum bias triggers ran unprescaled. However, it soon increased to the point where the minimum bias trigger rate became too high and had to be prescaled, and photon triggers were used instead. Eventually the photon trigger rate at the desired threshold also became too high, necessitating the extra constraints of the electron trigger at the same threshold. The rate for that trigger also increased, so to avoid prescaling the rate or increasing the threshold, tighter selection criteria were applied to the objects that fired the trigger. These criteria are looser than those applied in the course of analysis and so are not expected to seriously affect the trigger efficiency.

6.1.2 *Level-1 Trigger*

Reconstructing and triggering on an electron with the Level-1 Trigger involves information from the calorimeters, Regional Calorimeter Trigger, Global Calorimeter Trigger, and Global Trigger. Electrons and photons are indistinguishable in the L1 and are therefore treated the same, as “ e/γ ” objects.

The ECAL trigger primitive generation system (TPG) combines the energy deposits from a 5x5 square of ECAL crystals into a single energy value corresponding to one trigger tower. In addition, for each tower the ECAL TPG calculates a “fine grain veto” bit: if most of the tower’s energy is deposited in a 2x5 strip of crystals, then the energy pattern is considered inconsistent with that of an electron or photon, and the veto bit is set. Similarly, the HCAL also combines energy deposits into trigger towers; however, the HCAL fine grain structure bit is not used in electron-finding. The information from the ECAL and HCAL is compressed and sent to the Regional Calorimeter Trigger. The RCT does the main job of electron finding using several algorithms. The decompressed ECAL energy of each trigger tower is examined and compared to that of its four adjacent neighbors. If the energy of the tower is greater than that of its neighbors, it is considered an e/γ candidate. The candidate energy is set to the energy of the tower in question plus that of its highest-energy neighbor. However, if the tower’s fine grain bit is set, or if the ratio of the tower’s HCAL energy to ECAL energy is too high, it is vetoed as a candidate. A candidate is additionally considered “isolated” if, out of the eight towers directly surrounding it, there are five contiguous towers

forming an L-shape whose energies are low and who pass the fine grain and HCAL/ECAL ratio veto. Otherwise, the candidate is “non-isolated”. The RCT sends the collections of isolated and non-isolated candidates to the Global Calorimeter Trigger, which sends the highest-energy ones to the final step, the Global Trigger. The Global Trigger then applies the criteria determining whether or not this event passes the current set of trigger paths, for example, “two e/γ objects with energy above 10 GeV” or “one e/γ object with energy above 17 GeV”.

6.1.3 High-Level Trigger

The algorithms used to reconstruct trigger objects in the High-Level Trigger are the same as those used in the offline analysis and are described in Chapter 5. However, because of timing considerations, the algorithms are combined and configured differently at the HLT. The HLT has a limited amount of time in which to make a keep/discard decision for a given event in order to keep up with the flow of events. Therefore the algorithms are optimized for the best performance in the shortest time, which in practice means fewer steps and less complexity than the implementation at the offline level, where no such constraint exists. In particular, while offline-reconstructed electrons include tracker-driven electrons, those reconstructed at the HLT are ECAL-drive only, and the supercluster reconstruction algorithm uses a lower number of steps in the search process. Also, the size of the windows in which to search for hits in the pixel-matching part of the reconstruction is larger than that used offline.

The luminosity increase and consequent evolution of trigger menus described in Section 6.1.1 required incremental changes in the definitions of the current HLT paths. However, a few requirements remained common to all the paths used. In particular, the execution of each path begins by taking the Level-1 objects that passed the given L1 seed path and retrieving the information from the ECAL regions around those objects. From this information, clusters of energy deposits are identified, and clusters having too narrow a spatial distribution are rejected as being spikes. The trigger paths used in this analysis, shown in Table 6.1 all required (at least) one e/γ or electron candidate above a certain energy

Table 6.1: Trigger paths used to select data.

Trigger
HLT_Photon15_Cleaned_L1R
HLT_Ele15_LW_L1R
HLT_Ele15_SW_L1R
HLT_Ele15_SW_CaloEleId_L1R
HLT_Ele17_SW_CaloEleId_L1R
HLT_Ele17_SW_TightEleId_L1R
HLT_Ele17_SW_TighterEleIdIsol_L1R

threshold, with possible other criteria applied to the candidate as well.

Initially, the LHC collision rate was low enough that loose requirements and low thresholds could be used without overwhelming the system. The first trigger used did not require any matches to hits in the pixel part of the tracker, in contrast to all subsequent trigger paths used – without tracking requirements, it was by definition a photon trigger. In addition, after a few menu iterations the energy threshold applied to the candidate increased from 15 GeV to 17 GeV. The size of the window used to search for potential pixel matches also decreased (from “large-window” to “startup-window”). Later, various further criteria were applied to the shape of the total energy deposit ($\sigma_{i\eta i\eta}$, H/E) and to the quality of the match between the energy cluster and the track that make up the electron ($\Delta\eta_{in}$, $\Delta\phi_{in}$), as well as to the amount of energy surrounding the electron candidate (isolations). These further criteria are described in more detail in Section 6.2.1; in the current section it is sufficient to say that they were added in steps to the trigger paths to gradually tighten the selection.

6.1.4 *Primary Datasets*

The final step done centrally is the grouping of events passing similar HLT bits into primary datasets for storage. For example, in early running, all events passing a minimum-bias trigger were put into the MinimumBias dataset. Once the event rate increased and physics triggers fired more often, events were put into more specific datasets such as EG for all events passing e/γ (electron or photon) triggers, or later, Electron for all events specifically passing electron triggers. These primary datasets are the groupings of stored data that the offline analyses begin with.

6.2 Offline

The offline selection consists of any criteria applied to the data after it has already been stored, i.e. criteria applied to an already-existing primary dataset. This analysis uses data from two consecutive primary datasets: the EG dataset gathered during earlier data-taking (Run2010A, from run 135821 to run 144114), and the Electron dataset from later data-taking (the Run2010B era, runs 146240 to 149711). In addition, it was required that the primary vertex of each event be “good”: that is, within 24 cm of the nominal interaction point in the longitudinal direction and 2 cm in the transverse direction, and having a minimum of 4 degrees of freedom, which is a measure of the quality of the vertex fit. More detailed selections were made on the event contents, which will be discussed in the following sections. The electrons in these events were required to pass a series of criteria to determine “good” electrons. Finally, specific selections were made to the characteristics of the event as a whole in order to get the final collection of $Z \rightarrow e^+e^-$ candidate events.

6.2.1 Electron Selection

The electron selection starts with the full set of objects reconstructed as electrons, i.e. ECAL energy deposits with nearby tracks. The objects in this collection are not necessarily real electrons but are consistent with the electron signature at a low level. In order to select the reconstructed objects most likely to be real electrons from a Z, sets of further conditions are applied. First the electrons are required to be within the η region of ECAL in which they can be fully measured, in the barrel, $0. < |\eta| < 1.4442$, and in the endcap, $1.566 < |\eta| < 2.5$. A 25-GeV cut on the supercluster transverse energy is applied, followed by a set of cuts designed to reject electrons from photon conversions. Another set of cuts aims to distinguish real electrons from other energy deposits which may mimic electrons, and finally a set of cuts to make sure the electron is isolated, i.e. not part of a jet unrelated to the Z.

The collaboration (or egamma group) created several different electron selections of varying tightness of cuts, based on a given efficiency of the cuts (ref to iterative algo stuff). A tighter selection in general results in a purer sample of true electrons, but throws away many

more real electrons than a looser selection. These “working points” have target efficiencies ranging from 95%-efficient at the loosest (WP95) to 60% at the tightest (WP60). For early data-taking the WP95 and WP80 selections were chosen as the baseline selections for the Zee and Wenu analyses, respectively. As more data accumulated, it became clear that WP80 provided better performance than WP95 for the Zee analysis, as well. A deciding factor in the switch was the fact that the evolving trigger paths eventually used selections that were tighter than those used in the offline (WP95) analysis. This analysis uses the WP80 set of cuts for the complete dataset.

The selection cuts are displayed in the following cuts applied only to simulated data. The QCD Monte Carlo sample has effective electron isolation cuts already applied, so until the corresponding selection cuts are applied to data, the data does not agree with the Monte Carlo distributions. However, after all cuts have been applied the agreement is good; this will be demonstrated in a later section.

Conversion Rejection

The conversion rejection step of the electron selection aims to eliminate photons that have converted into electron pairs. Several variables have been chosen to distinguish between conversions and prompt electrons, focusing on the track properties. Since prompt electrons are expected to hit the first layer of the tracker, any electron-associated track that is missing initial hits can be rejected as a conversion that occurred within the tracker. The looser requirements of the 95 percent efficient working point allow an electron to have one missing hit; tighter working points such as WP80 require zero missing hits. Fig. 6.1a shows the number of missing hits for each electron. In addition, the conversion rejection algorithm searches for candidate electron-pair partner tracks, i.e. track pairs that appear to come from the same point. Every electron track is compared to all opposite-charge tracks within a dR cone of 0.3, and two values are calculated: a measure of the difference in the angles in the r-z plane between the two tracks ($\Delta \cot \theta$, Fig. 6.1b), and the distance of closest approach between the two tracks in the x-y plane ($dist$, Fig. 6.1c). Electron pairs from conversions are expected to have low values for both quantities, since both electron tracks begin at the

point of conversion and initially have the same direction as the converting photon. The WP80 requirements apply a threshold at 0.02 for both quantities.

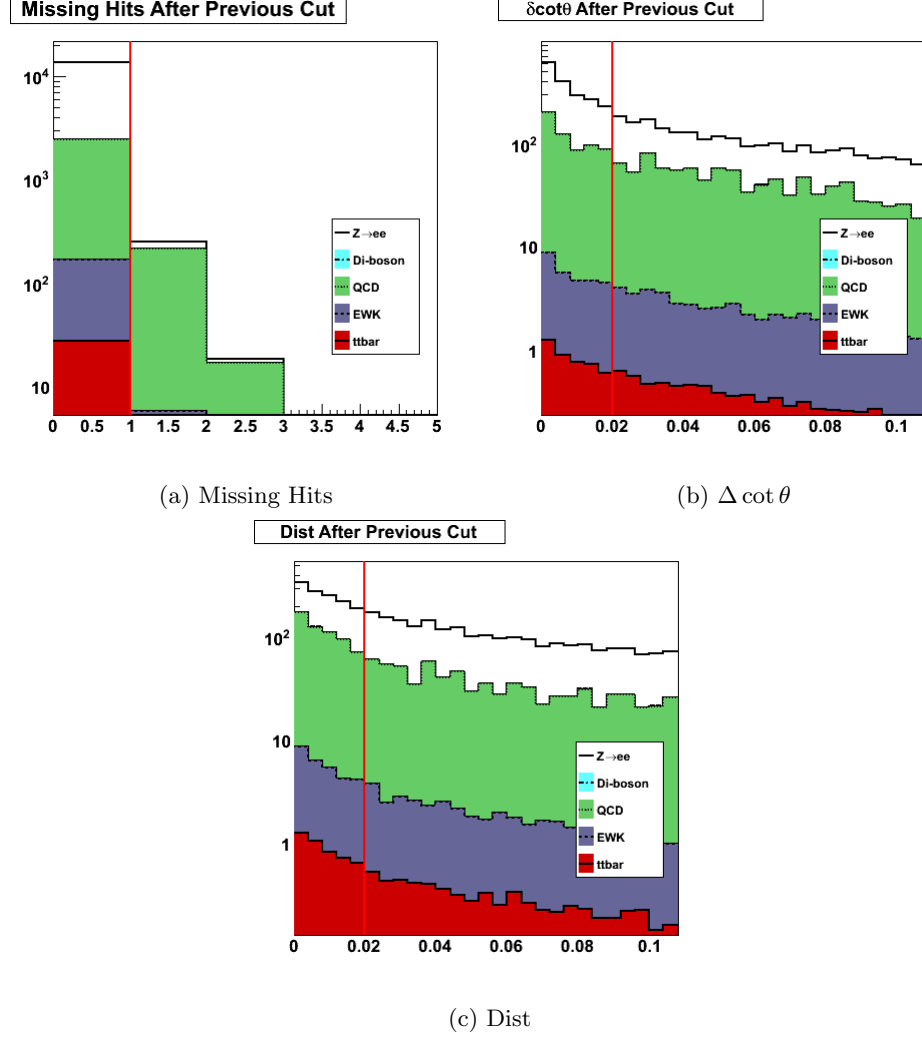


Figure 6.1: Conversion rejection variables after previous cuts.

Electron Isolation

It is expected that the products of a Z decay will not exit the interaction region in the vicinity of many other particles, since the Z decay products are not part of a jet. We want to select

those real electrons that are most likely to be prompt decay products, not those that come from some secondary interaction with many other end products. Therefore we apply a set of cuts requiring little detector activity in the area immediately surrounding the reconstructed electron. The so-called isolation cuts look at the energy deposits in the electromagnetic and hadronic calorimeters separately, as well as the total p_T of neighboring tracks, within specified ΔR cones around the electron-associated deposit/track. These energies are divided by the p_T of the electron itself to arrive at a relative value, which allows for higher-energy particles that may deposit more energy in the surrounding area. The thresholds for each of the three types of isolation is set separately for barrel and endcap, as given in Table 6.2. Plots of the separate isolation distributions for both barrel and endcap are shown after previously-applied selection cuts in Fig. 6.2 for track isolation, Fig. 6.3 for ECAL isolation, and Fig. 6.4 for HCAL isolation.

Table 6.2: Thresholds for WP80 relative isolation.

Cut variable	Barrel	Endcap
Track	0.09	0.04
ECAL	0.07	0.05
HCAL	0.10	0.025

Electron Identification

Several “electron identification” variables are used to differentiate genuine electrons from other particles or conditions that may mimic electron behavior. Since an electron is reconstructed from a supercluster matched to a track, it is desired that the match is a good one. The quality of matching can be estimated by looking at measures of the distance between the two components. The position in η and ϕ of the supercluster is compared with the η and ϕ position of the track extrapolated back to the vertex, with the differences being labeled $\Delta\eta_{in}$ and $\Delta\phi_{in}$, respectively. It is expected that a supercluster and track from the same

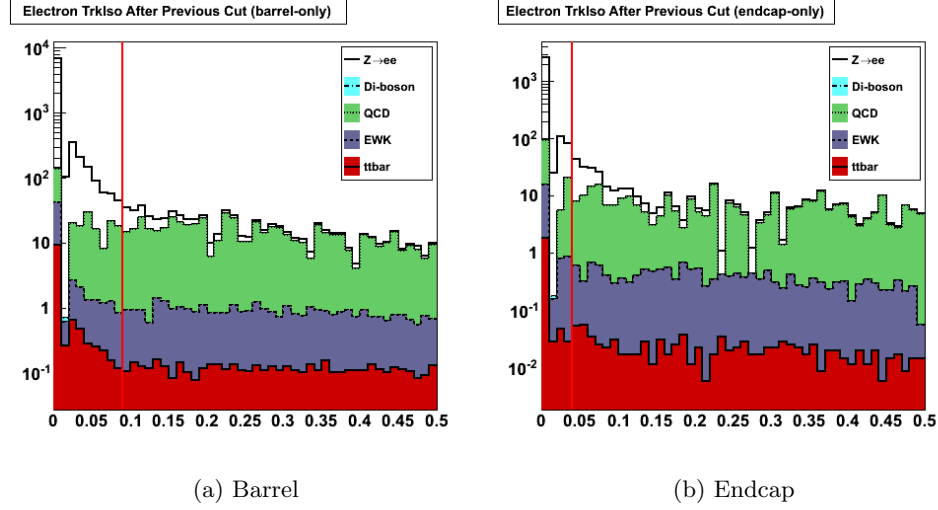


Figure 6.2: Electron tracker isolation variables before respective cuts.

electron will be close together, so upper limits on these quantities are defined as in Table 6.3. Distributions of electron $\Delta\eta_{in}$ and $\Delta\phi_{in}$, each plotted after the previous selection step, are shown in Fig. 6.5 and Fig. 6.6.

The electron's energy deposit is also expected to be narrow relative to that of a jet. A measure of the width in the η direction is therefore calculated using the energies and positions of the individual ECAL crystals, $\sigma_{i\eta i\eta}$, and this quantity is required to be below the thresholds listed in Table 6.3. (The ϕ direction is not examined because bremsstrahlung from the electron can cause a significant spread of the energy in ϕ .) The $\sigma_{i\eta i\eta}$ distribution for electrons after all previous cuts is shown in Fig. 6.7. In addition, an electron is expected to deposit most of its energy in the electromagnetic portion of the calorimeter. A large hadronic energy deposit is more likely to come from a jet, so a limit on the ratio of the energy deposit in the electromagnetic calorimeter to that in the hadronic calorimeter, H/E , is applied (values given in the table). The distribution of H/E for electron candidates is plotted after all previous selection steps in Fig. 6.8.

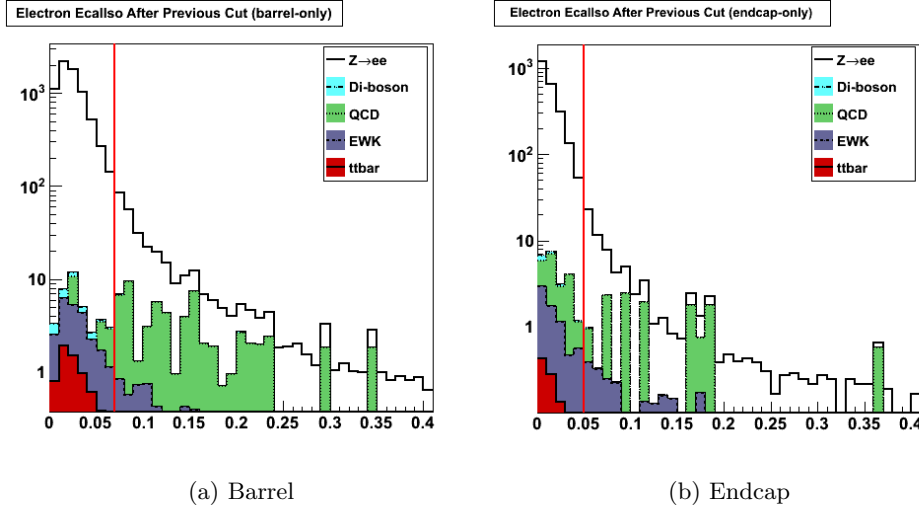


Figure 6.3: Electron ECAL isolation variables before respective cuts.

Matching Offline-Reconstructed Objects to Trigger Objects

In order to ensure that the offline-reconstructed objects being examined were also reconstructed by the trigger, each offline object is compared with the list of trigger objects firing a given path. If the offline object lies close enough to a trigger object (measured by ΔR), it is considered to have passed the trigger. Each event in this analysis was required to have at least one electron trigger-matched in this way, within a ΔR distance of 0.2. This demonstrates that the constituent objects in the event did in fact fire an appropriate trigger.

Table 6.3: WP80 electron identification cuts.

Cut variable	Barrel	Endcap
$\sigma_{i\eta i\eta}$	0.01	0.03
$\Delta\phi_{in}$	0.06	0.03
$\Delta\eta_{in}$	0.004	0.007
H/E	0.04	0.025

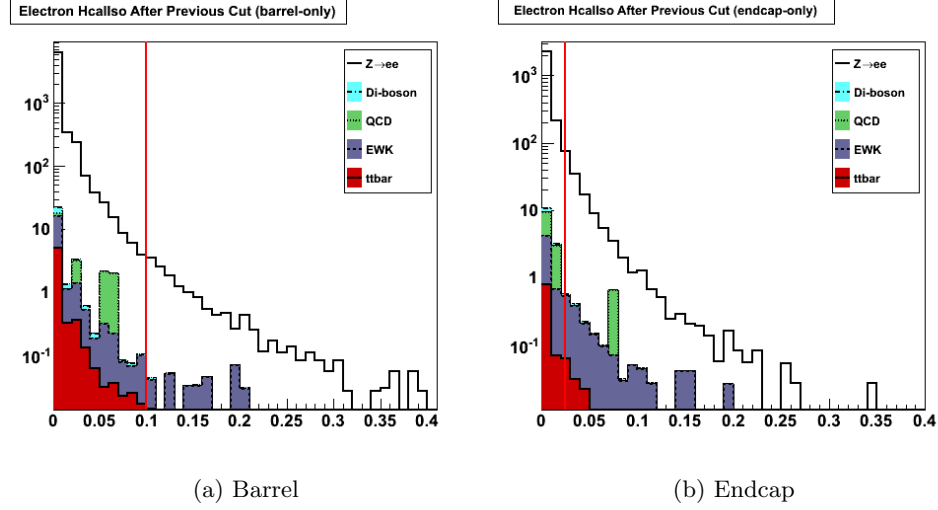


Figure 6.4: Electron HCAL isolation variables before respective cuts.

6.2.2 $Z \rightarrow e^+e^-$ Event Selection

In order to be included in the final selected sample of $Z \rightarrow e^+e^-$ events, an event must have two electrons passing the selection cuts described in Section 6.2.1. Only one of the electrons is required to match to a trigger object, however; even if only one electron fires the trigger, the event is kept and the trigger’s purpose is served, and there is no extra necessity for both electrons to fire the trigger. The two electrons are also required to have an invariant mass between 60 and 120 GeV; this defines a window around the Z mass peak at about 91 GeV.

Acceptance for $Z \rightarrow e^+e^-$ Events

The acceptance (abbreviated A) is the factor representing the fraction of events theoretically able to be detected with the given experiment. It includes considerations related to a particle’s kinematic quantities (its position and momentum) as well as to the precise definition of the signal interaction. In particular, the particle-detecting elements of any given detector, including CMS, do not fully cover the area around the interaction point – by necessity, some of that area is taken up by the beam pipe. This means that any end-product particle that goes into a “dead” space instead of a detecting element will not be detected.

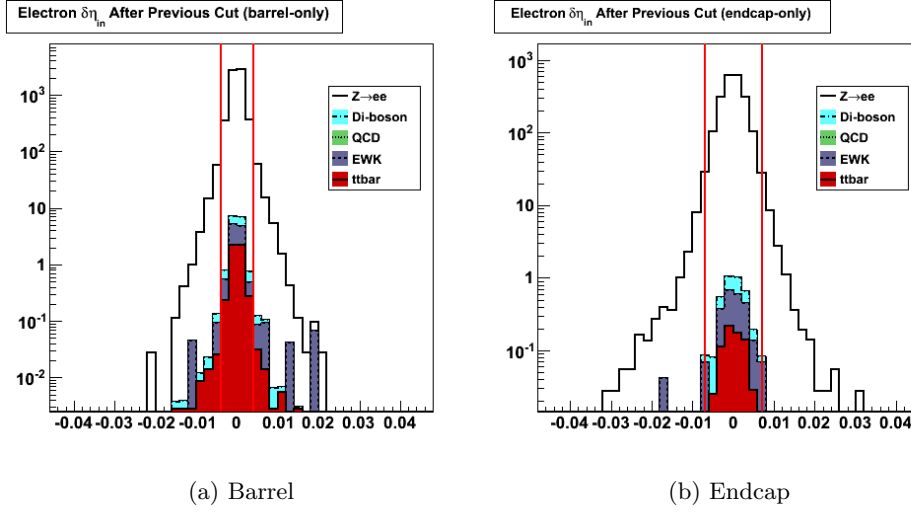


Figure 6.5: $\Delta\eta_{in}$ electron identification variables after previous cuts. A low $\Delta\eta_{in}$ indicates a good match between an electron candidate's supercluster and track. Values shown for (a) barrel and (b) endcap.

Therefore the acceptance factor takes into account the actual solid-angle area covered by the relevant parts of the detector. In addition, it is generally more difficult to distinguish very low-energy objects from background noise, so the acceptance also includes a lower limit on the energy of the particle.

In this analysis, the signal was defined to be those $Z \rightarrow e^+e^-$ events whose invariant mass was between 60 and 120 GeV; this captures the Z peak around 90 GeV. The acceptance was defined as the fraction of those signal events whose electron end products fall into the η region used for the analysis, $|\eta| < 1.4442$ or $1.566 < |\eta| < 2.5$, and whose transverse energies were both above 25 GeV. These values can also be found in Table 6.4.

Table 6.4: Criteria for determining acceptance.

Kinematic quantity	Requirement
η	$ \eta < 1.4442$ or $1.566 < \eta < 2.5$
E_T	$< 25\text{GeV}$

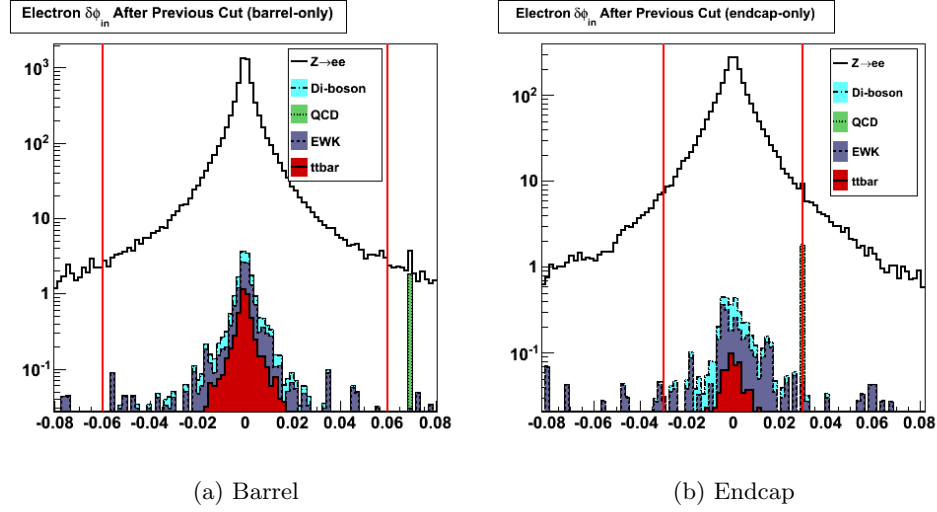


Figure 6.6: $\Delta\phi_{in}$ electron identification variables after previous cuts. A low $\Delta\phi_{in}$ indicates a good match between the electron candidate's supercluster and track. Values shown for (a) barrel and (b) endcap.

Acceptance cannot be calculated from data, because it is impossible to know how many events were missed – the only events we have are those which were not missed. Instead acceptance is determined by applying the defining criteria to a sample of simulated data, a sample large enough that statistical uncertainties become small enough to ignore. Since the simulated sample is expected to model real data well enough, this number is taken as the true acceptance for real data. However, the information used to generate the simulated sample is only a theoretical model; there are many ways in which it may actually differ slightly from what is used in the generation. These possible differences are taken into account in the uncertainty calculated for the acceptance, which will be detailed in a later section.

It is also useful to combine the supercluster reconstruction efficiency with the acceptance. The supercluster reconstruction efficiency is impossible to determine from data for the same reason as for the acceptance – it would require knowing about events that were missed, in this case electrons that had impacted the detector (specifically ECAL) but which had not resulted in the formation of a supercluster. Therefore this is also determined from

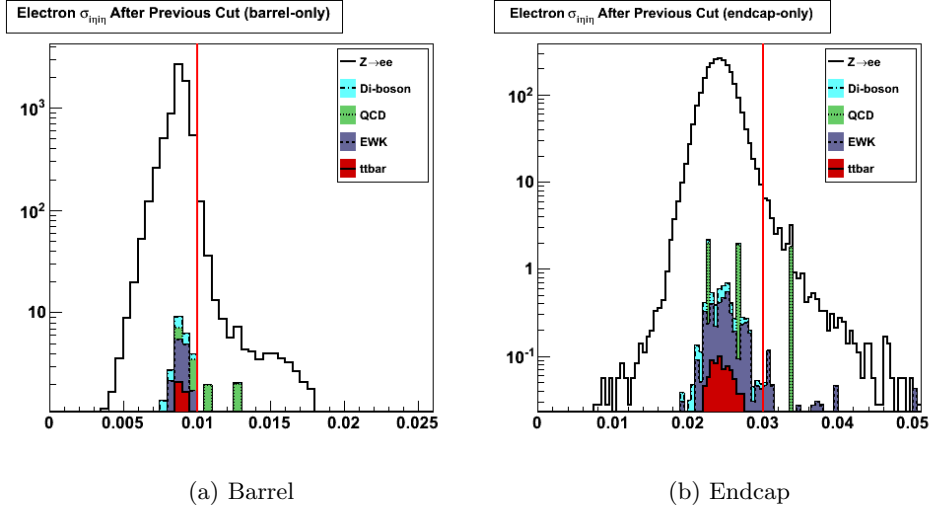


Figure 6.7: $\sigma_{i\eta i\eta}$ electron identification variables after previous cuts. $\sigma_{i\eta i\eta}$ is a measure of the spread of the electron candidate’s energy deposite in η . A true electron is expected to be relatively narrow in η . Values shown for (a) barrel and (b) endcap.

simulated data. Effectively it is combined with the acceptance by applying an additional requirement on the acceptance: the simulated event should have two electrons passing the η and transverse energy requirements, and those electrons should also both be matched to reconstructed superclusters. The fraction of total events passing these combined criteria is referred to as “ECAL acceptance” or A_{ECAL} .

The pure kinematic acceptance for this analysis was determined with the POWHEG-generated Monte Carlo sample to be 0.423 and the ECAL acceptance was likewise calculated as 0.387. These values can also be found in Table 6.5.

Table 6.5: Acceptance values.

Quantity	Value
A	0.423
A_{ECAL}	0.387

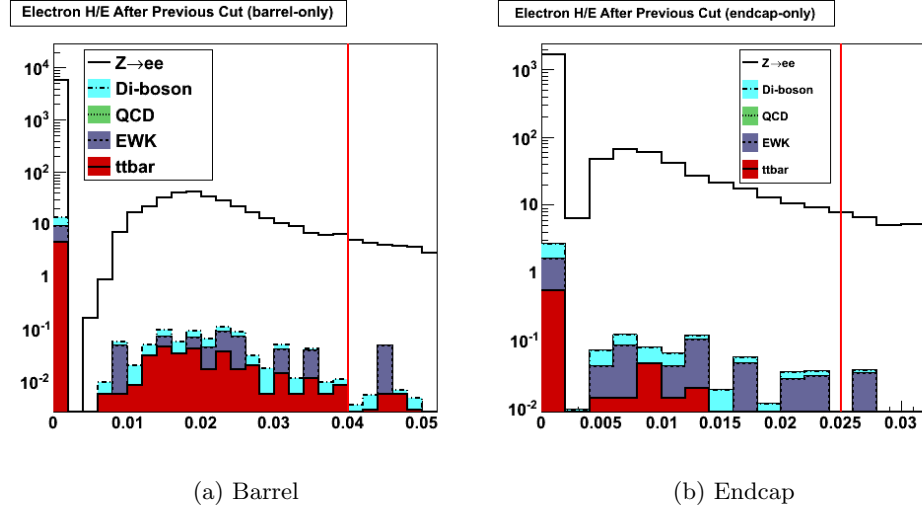


Figure 6.8: H/E electron identification variables after previous cuts. H/E is the ratio of the electron candidate’s energy deposited in HCAL to that deposited in ECAL. A true electron is expected to deposit relatively little energy in the HCAL. Values shown for (a) barrel and (b) endcap.

Efficiency of Selection for $Z \rightarrow e^+e^-$ Events

It is important to know how well the selection steps perform their task of keeping signal events; any signal events that are missed must be accounted for in the final calculation to get an accurate result. This factor is quantified in the efficiency, which is defined as the fraction of total objects or events passing a given selection. Simulated data can be used to calculate efficiency, since it contains the “truth” information – the true total number of events, whether or not they survive a given selection step. However, the simulation may not correctly model some aspect of the real data, which would cause a discrepancy with the real-data efficiency and a corresponding inaccuracy in the final result. Therefore it is desirable to use so-called “data-driven” methods, determining the efficiency as closely as possible using only the data itself. Because the truth information does not exist in real data, the number of events passing or failing a given selection must be determined with another method.

The tag-and-probe method is used to determine the efficiency of a given set of electron

selection steps. In this method, a set of data events is identified which appear to be good $Z \rightarrow e^+e^-$ candidate events. One of the electrons in each of these events must pass all selection cuts and essentially be identified as a golden electron; this is the tag. The criteria applied to the other electron are looser: it is not required to pass the selection cuts being tested. This is the probe electron. The event's status as a Z candidate is determined by enforcing a mass window cut on the invariant mass of the tag-probe system (this is sufficient to select a high-purity sample of events in which both legs are real electrons). The naive selection efficiency is then given by the fraction of probe electrons that pass the requirement being tested.

However, a more sophisticated analysis of the efficiency can be performed by fitting the invariant mass spectra of the tag-probe pairs for both passing and failing probes to a lineshape representing both the signal and the background. Then the distribution of passing probes can be represented as the signal lineshape multiplied by the efficiency and the total number of signal events added to the background lineshape multiplied by the number of passing background events. The failing probe distribution can be represented in the analogous way, using $(1 - \text{efficiency})$ instead. From this pair of fits the total number of events as well as the efficiency can be extracted simultaneously.

However, the tag-and-probe may possibly introduce a bias into the calculation, in particular if observing one leg makes it more likely that the other leg is observed. For example, if the electrons are essentially back-to-back, one is more likely to fall within the detector coverage if the other one does as well. In addition, there is a slight discrepancy between the efficiencies calculated using data and those using the standard Monte Carlo sample. Therefore, the strategy in this analysis is to calculate the “true” efficiency from the simulation sample, as well as the tag-and-probe efficiencies from the real data and simulation samples. The “true” efficiency is then corrected by the ratio of the data and simulation tag-and-probe efficiencies:

$$\epsilon = \epsilon_{MC}^{true} \times \left(\frac{\epsilon_{data}^{T\&P}}{\epsilon_{MC}^{T\&P}} \right)$$

This implies a certain amount of trust in the generator-level simulation of the interaction. The corresponding uncertainty in the efficiency due to the generator-level simulation is dealt

with in a later chapter.

The event selection efficiency in this analysis is determined by first obtaining the efficiencies of the individual selection steps given above. This also includes the initial step of reconstructing the electron objects from superclusters. Since each selection step is applied one after the other, the overall selection efficiency is given by the product of the individual efficiencies. In addition, since most steps are required to apply to both electrons, the single-electron efficiencies of these steps (the result of the tag-and-probe method) are squared to get the efficiency of both electrons passing that step. The trigger step, however, is a special case since only one electron is required to pass. In this case, the quantity used is the probability that at least one electron passes, which is equivalent to the probability both electrons fail subtracted from 1.

The formula used to calculate the overall event selection efficiency from the individual step efficiencies is therefore

$$\epsilon_{total} = \epsilon_{reco}^2 \times \epsilon_{isol}^2 \times \epsilon_{ID}^2 \times \left(1 - (1 - \epsilon_{trig})^2\right)$$

The efficiencies are given for each selection step in Table 6.6 and for all the triggers used in data individually in Table 6.7. The full event selection efficiency was calculated to be $61.0 \pm 0.5\%$.

Table 6.6: Efficiencies of electron selection steps.

Step	True MC	MC T&P	Data T&P	Ratio	Efficiency
Reconstruction	0.965	0.972	0.971 +/- 0.002	0.999 +/- 0.002	0.964 +/- 0.002
Isolation	0.926	0.927	0.910 +/- 0.003	0.976 +/- 0.003	0.905 +/- 0.003
Identification	0.906	0.907	0.897 +/- 0.003	0.989 +/- 0.003	0.896 +/- 0.003
Trigger	0.959	0.941	0.972 +/- 0.001	1.032 +/- 0.001	0.991 +/- 0.001
Full Event selection	0.654	0.665	0.621 +/- 0.005	0.933 +/- 0.007	0.610 +/- 0.005

Table 6.7: Efficiencies of all the triggers used in data as determined with the tag-and-probe method.

Trigger	Efficiency	Error (fit)
HLT_Photon15_Cleaned_L1R	0.976	0.002
HLT_Ele15_LW_L1R	0.961	0.005
HLT_Ele15_SW_L1R	0.981	0.003
HLT_Ele15_SW_CaloEleId_L1R	0.986	0.003
HLT_Ele17_SW_CaloEleId_L1R	0.992	0.002
HLT_Ele17_SW_TightEleId_L1R	0.973	0.002
HLT_Ele17_SW_TighterEleIdIsol_L1R_v1	0.973	0.003
HLT_Ele17_SW_TighterEleIdIsol_L1R_v2	0.977	0.002
HLT_Ele17_SW_TighterEleIdIsol_L1R_v3	0.974	0.003
Overall	0.9763	0.0009

Distribution of Z Kinematic Variables

Since a significant number of $Z \rightarrow e^+e^-$ candidate events have been identified, it is useful to examine the kinematic properties of these events. Even though this analysis does not depend on these properties, comparing the distributions from the data with those from the simulated samples can provide an additional check of validity of the simulation.

Fig. 6.10 shows the distributions of standard kinematic variables for the reconstructed Z candidates: transverse momentum, rapidity, and azimuthal angle. The data (black dots) is compared with the simulation including background samples (lines/filled areas). As the ratio plots show, the agreement is good in general; large error bars indicate bins where the event count is not high enough to give a very precise estimate (such as the higher range of the Z p_T plot). However, there are areas in which there is an evident discrepancy between the data and the simulation, most notably along the outer edges of the Z rapidity plot. Possible causes might include a slightly inaccurate description of the particle interaction in

the generator, or a slight miscalibration of the detector. Since the difference is slight and does not affect this analysis, though, it can safely be ignored and the general agreement accepted.

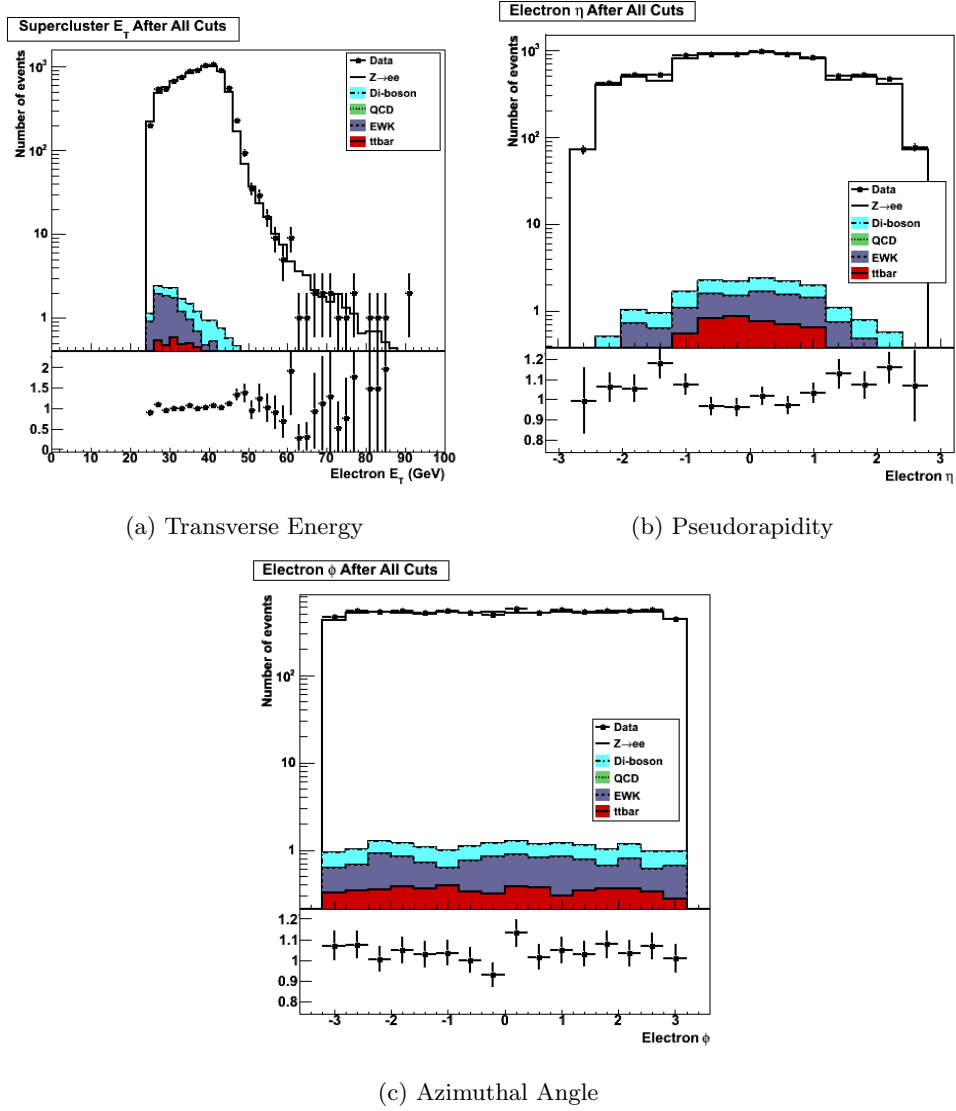


Figure 6.9: Spectra of reconstructed electron kinematic quantities after all electron selection cuts. The plots show distributions for electron p_T , η , and ϕ respectively. The agreement between real data and the simulated sample is good, demonstrating that the effects of the cuts on the data are understood.

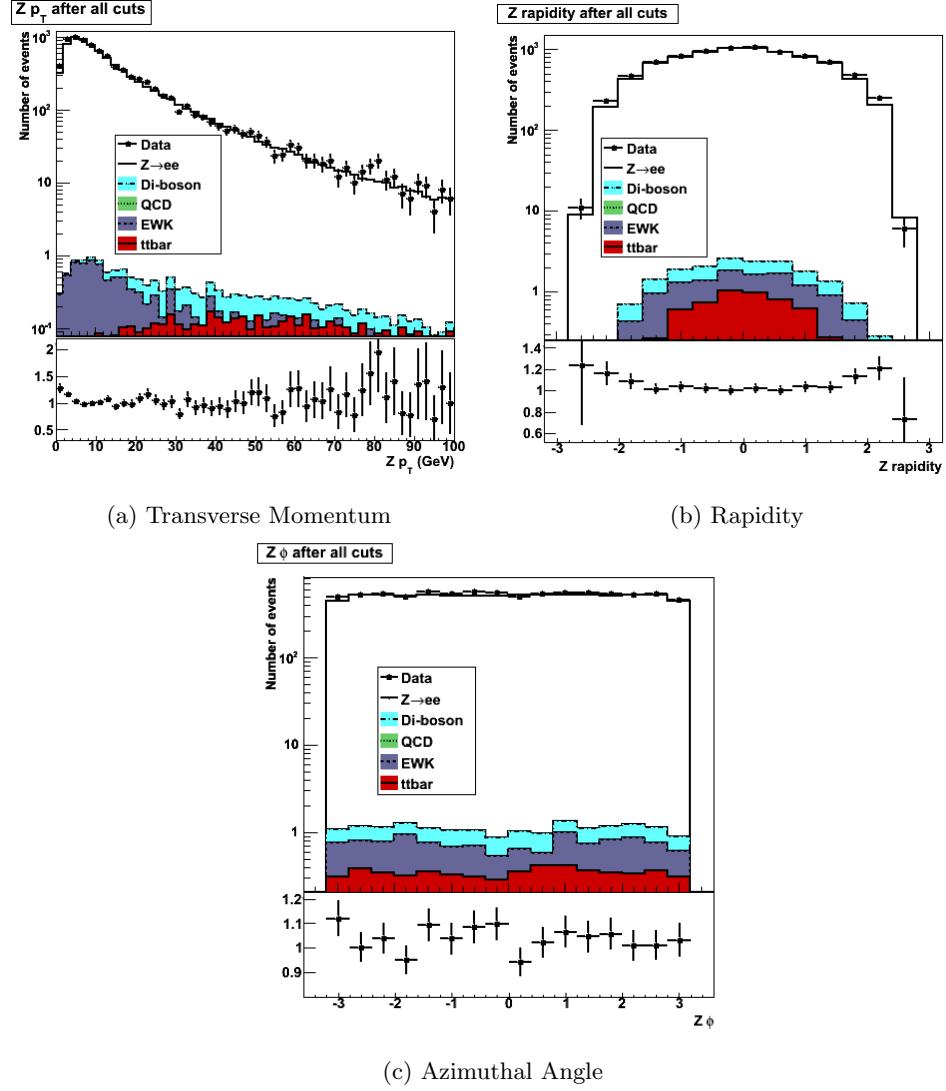


Figure 6.10: Kinematic quantities of Z candidates after full event selection.

Chapter 7

ANALYSIS METHOD

The previous chapter explained how the data events used in the analysis were selected. The endpoint of that chapter was a set of $Z \rightarrow e^+e^-$ candidate events: events that had two well-reconstructed electrons with an invariant mass between 60 and 120 GeV. This chapter explains the rest of the process needed to obtain the result, the final cross section. This includes assembling all the necessary components of the calculation, as well as determining the uncertainty inherent in each component and combining the uncertainties into one overall value.

7.1 Invariant Mass Spectrum

As explained in Section TBD, the invariant mass of a set of decay-product particles is essentially the mass of the original particle that decayed. It is defined mathematically by

$$M_{inv}^2 = \left(\sum E \right)^2 - \left\| \sum \mathbf{p} \right\|^2$$

where \mathbf{p} is the momentum and E the energy for a given particle; \sum denotes the sum over all particles – a vector sum in the case of momentum, while $\|\mathbf{p}\|$ denotes the momentum's magnitude. In our case, for a two-particle system, this can be written as

$$M_{inv} = \sqrt{(E_1 + E_2)^2 - \|\mathbf{p}_1 + \mathbf{p}_2\|^2}$$

The invariant mass distribution within the mass window for the electron pairs in the events surviving all selection cuts is shown in Fig. 7.1. An explanation of how the different simulation samples were combined can be found in Section FIXME. The peak around the Z mass (91 GeV) is clearly visible in both the Monte Carlo (solid) and data (points) distributions. The background contributions estimated from Monte Carlo simulation are shown as the colored (grayscale) areas. A slight shift in the peak position along the x-axis is evident, due to slight miscalibrations of the calorimeter response to given energies. This difference

is accounted for in the systematic uncertainty due to the electron energy scale (see Section 7.4.4). The selected events are also divided into categories according to which part of the detector, barrel or endcap, each of their electron legs were located, and the resulting invariant mass spectra are shown in Fig. 7.2. Agreement in each case is good to within the uncertainty.

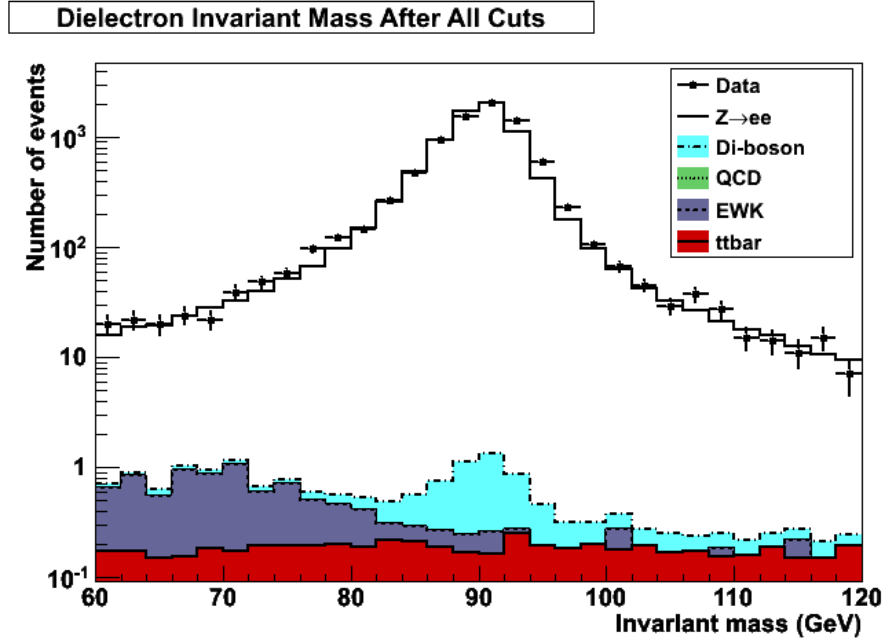


Figure 7.1: Electron-pair invariant mass after full selection. The peak around the Z mass (91 GeV) is clearly visible in both the Monte Carlo (solid) and data (points) distributions. The background contributions estimated from Monte Carlo simulation are shown as the colored (grayscale) areas. A slight shift in the peak position along the x-axis is evident, due to slight miscalibrations of the calorimeter response to given energies. This difference is accounted for in the systematic uncertainty due to the electron energy scale.

7.2 Cross-Section Extraction

The formula to calculate the cross section from Section 1.1 is

$$\sigma_Z \times \text{BR}_{Zee} = \frac{n_{\text{signal}}}{\mathcal{L} \times \epsilon \times A}$$

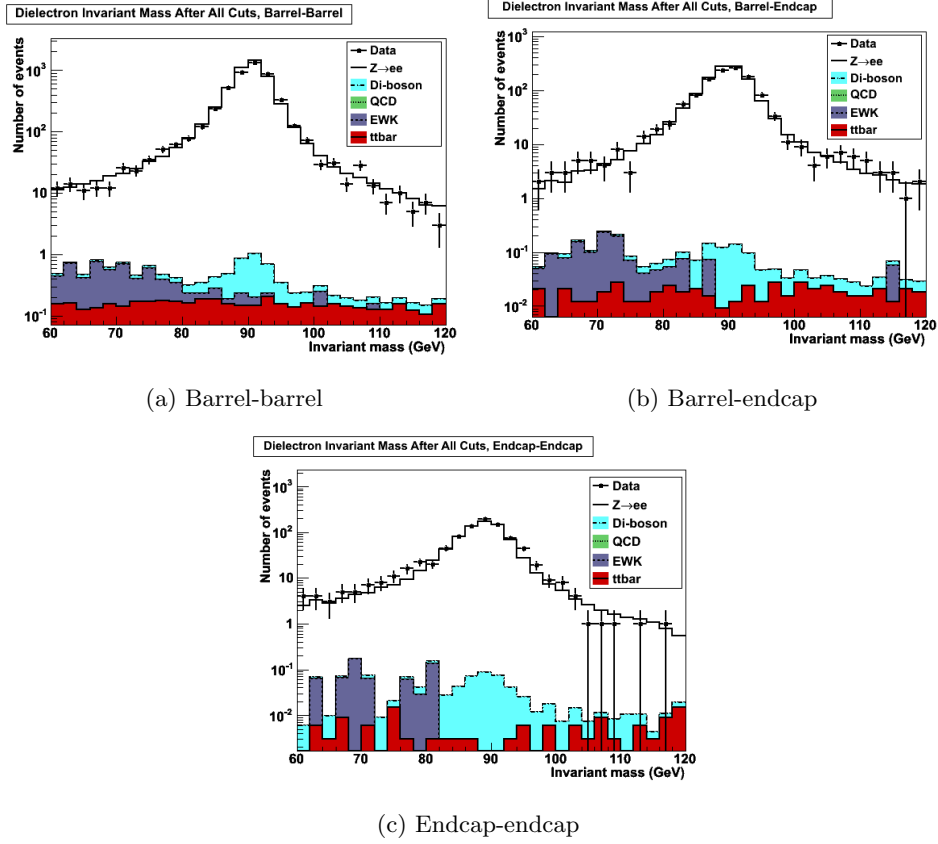


Figure 7.2: Electron-pair invariant mass by η after full selection. Each combination of electron η -positions is distributed separately: both electrons in the barrel (a), one electron each in the barrel and the endcap (b), and both electrons in the endcap (c). In each case the agreement between data and Monte Carlo simulation is good.

where \mathcal{L} is the luminosity, $n_{\text{signal}} = n_{\text{total}} - n_{\text{background}}$ is the number of signal events, derived from the total number of selected events and the estimated number of background events, and

$$\epsilon = \epsilon_{\text{reco}} \times \epsilon_{\text{sel}} \times \epsilon_{\text{trig}}$$

is the event selection efficiency, composed of the efficiencies of the individual steps: reconstruction, selection, and trigger, respectively. Many of the elements of these formulas have been determined in previous sections: the event selection efficiency ϵ was calculated as 0.610 in Section 6.2.2, and the acceptance as 0.387 in Section 6.2.2. The luminosity is known to

be 36.1 pb^{-1} (Section 3.3.8). The remaining component to be determined is the number of signal events, which is obtained from the total number of selected events and the estimated number of background events. The total number of selected events is the number of events in the invariant mass histogram, Fig. 7.1: this is 8453 events. The final element, estimated number of background events, will be calculated in Section 7.3.

7.3 Background Subtraction

It is expected that some number of non-signal events will look enough like signal events to pass the full selection. These background events need to be accounted for in the final calculation to arrive at the most accurate result. Therefore, the number of background events should be estimated and subtracted from the total number of events; this estimation can be done in a variety of ways. The methods used in this analysis are detailed in the following sections.

7.3.1 Monte Carlo Estimation

One estimate of the number of background events can be made directly from the Monte Carlo simulation, by counting the number of simulated events that pass the full selection. The numbers of passing events for the simulated background samples used are shown in Table 7.1, with a total of 18.5 events. The accuracy of this method may be questionable due to unaccounted differences between the simulation and real data, however. In particular, the detector may not be correctly modeled in the simulation, causing slight differences in event numbers that are non-negligible at this level. Therefore, data-driven methods of background subtraction are also used, and the results from all the methods are checked against each other to confirm the accuracy of each method.

7.3.2 Template Method

The template method is a data-driven method to estimate the background contribution in a selected sample. In general, it uses modified cuts to select data events that are especially likely to be signal or background; these subsamples are called templates. Extra-tight

Table 7.1: Number of events passing full selection for each background, estimated from Monte Carlo simulation.

Background Sample	Number of Events
Wenu	0.36
Wtaunu	0.0
Ztautau	6.4
Electroweak Total	6.7
ttbar	5.8
QCD	0.0
WW	1.5
WZ	1.4
ZZ	3.0
Di-boson Total	6.0
Total Background	18.5

cuts select a “signal-rich” sample, while a combination of loose and inverted cuts select a “background-rich” sample. A particular selection variable is chosen which has significantly different distributions for signal and background; here, the relative tracker isolation variable is used. The template method compares the distributions of that variable from the selected signal and background templates to the distribution from the full data sample. Essentially, the method determines what combination of the signal and background templates best fits the data distribution, in an effort to determine what fraction of the data sample is signal and what fraction is background. This method is only useful to determine the background from QCD; its technique relies on “background” electrons being qualitatively different from “good electrons” (measured by the variable being used), but electrons from interactions similar to the signal interaction, such as $W \rightarrow e\nu$, are also “good electrons.” Objects arising from QCD interactions that are identified as “electrons” are more likely clusters of hadronic particles that come out looking like electrons, so they can be distinguished from “good electrons” and therefore the template method can identify them as background.

Template Definitions

Several new working points, based on WP80 (see Section 6.2.1), were defined to select the signal-like and background-like template samples, as well as the data distribution used in the comparison.

The first working point, used to select the data, is called “Semi-Tight” and is identical to WP80 except for the removal of its relative track isolation requirement. This is solely to allow the full spectrum of track isolation values to be examined. All events with two electrons passing the Semi-Tight working point are included in the data sample.

The working point used to select the “signal-rich” data sample is called “Tight” and is based on the Semi-Tight working point but with a few cuts made tighter. (This is to ensure a very pure sample; however, a very tight working point would not work for standard data selection because its efficiency is lower – it cuts out too many good events. For the template method, though, a lower number of events is sufficient.) The tighter cut values were taken from WP70, another of the standard working points. Specifically, the values for $\Delta\phi_{in}$ in

both the barrel and the endcap, H/E in the barrel, and $\Delta\eta_{in}$ in the endcap were modified. To select the signal template sample, each event was required to have two electrons passing the Tight working point, and the two electrons were required to have opposite signs.

The final working point used for the template method is the “Loose” working point, which is used in selecting the background template. The Loose working point is modeled on the Semi-Tight working point, as is the Tight, but for the Loose working point all the thresholds for the isolation and electron isolation variables are increased by a factor of 5. In addition, in order to get enough data for a reasonable background template, the ECAL isolation variable had to be further loosened, from 0.07 to 2.5 in the barrel and from 0.05 to 1.0 in the endcap, and the supercluster E_T cut had to be lowered from 25 GeV to 20 GeV. Events selected for the background template must have two electrons, one passing the Semi-Tight working point and one passing the Loose, and these two electrons are required to have the same sign in order to reject signal.

Table 7.2 shows the cut values for WP80 and how each of the working points used in the template method is modified with respect to WP80. Cut values changed with respect to WP80 are displayed in italics.

The mechanics of the fit are done by the ROOT class `TFractionFitter`, whose purpose is to perform the exact sort of fit needed for the template method. It determines the composition of template distributions that best fits a given “data” distribution using a likelihood fit. Statistical uncertainties in both the templates and the data are taken into account in the fitting process.

Results

The relative tracker isolation distributions of the signal and background templates are shown in Fig. 7.3. Each curve independently integrates to 1 and shows the fraction of events of the given type with relative tracker isolation in each bin. The difference in shape between signal (solid, black) and background (dotted, red) is clear: the background has a much higher fraction of events with higher values of isolation, while the data has a much higher fraction of events with lower values. The lowest bin contains 90% of signal compared with

Table 7.2: Summary of the standard working point as well as the working point definitions used in the template method. Italics indicates a cut changed relative to WP80.

Cut Variable	WP80		Semi-Tight		Tight		Loose	
	Barrel	Endcap	Barrel	Endcap	Barrel	Endcap	Barrel	Endcap
Conversion Rejection								
missing hits	0	0	0	0	0	0	0	0
dist	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
$\delta \cot \theta$	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Relative Isolation								
ECAL	0.07	0.05	0.07	0.05	0.07	0.05	<i>2.5</i>	<i>1.0</i>
HCAL	0.1	0.025	0.1	0.025	0.1	0.025	<i>0.5</i>	<i>0.125</i>
track	0.09	0.04	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>	<i>N/A</i>
Electron Identification								
$\sigma_{i\eta i\eta}$	0.01	0.03	0.01	0.03	0.01	0.03	<i>0.05</i>	<i>0.15</i>
$\Delta\phi_{in}$	0.06	0.03	0.06	0.03	<i>0.03</i>	<i>0.02</i>	<i>0.3</i>	<i>0.15</i>
$\Delta\eta_{in}$	0.004	0.007	0.004	0.007	0.004	<i>0.005</i>	<i>0.02</i>	<i>0.035</i>
H/E	0.04	0.025	0.04	0.025	<i>0.025</i>	0.025	<i>0.2</i>	<i>0.125</i>
Supercluster E_T	25		25		25		<i>20</i>	

around 65% of background. These features enable the template method in general and the TFractionFitter in particular to fit the shape of the templates to the distribution of the data. The data distribution is shown in Fig. 7.4 as the hollow circles (black), while the fit as determined by the TFractionFitter is displayed as asterisks (red). The agreement is very good, showing that TFractionFitter was able to make a good match between the data and its combination of templates.

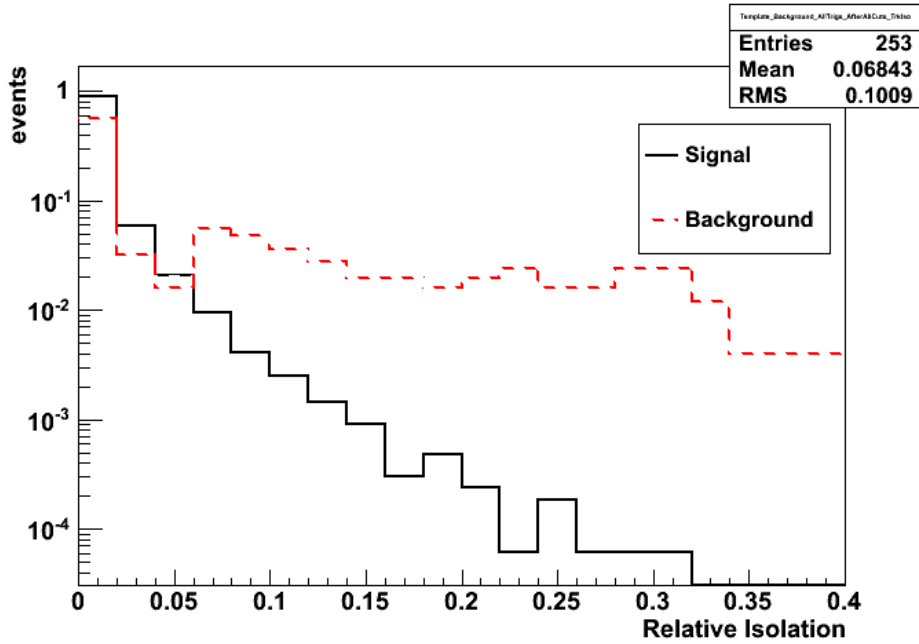


Figure 7.3: Shapes of signal and background templates used in template method of background subtraction. Each curve independently integrates to 1 and shows the fraction of events of the given type with relative tracker isolation in each bin. The solid (black) curve represents signal and the dotted (red) shows background. The difference in shape between the two categories is clear: the background has a much higher fraction of events with higher values of isolation, while the data has a much higher fraction of events with lower values. The lowest bin contains 90% of signal compared with around 65% of background. This difference is exploited in the template method to determine the signal vs. background composition of a given data sample.

Over the full range of the relative tracker isolation values used for the template fit, the fraction of signal was determined to be 0.998 ± 0.014 , while the fraction of background

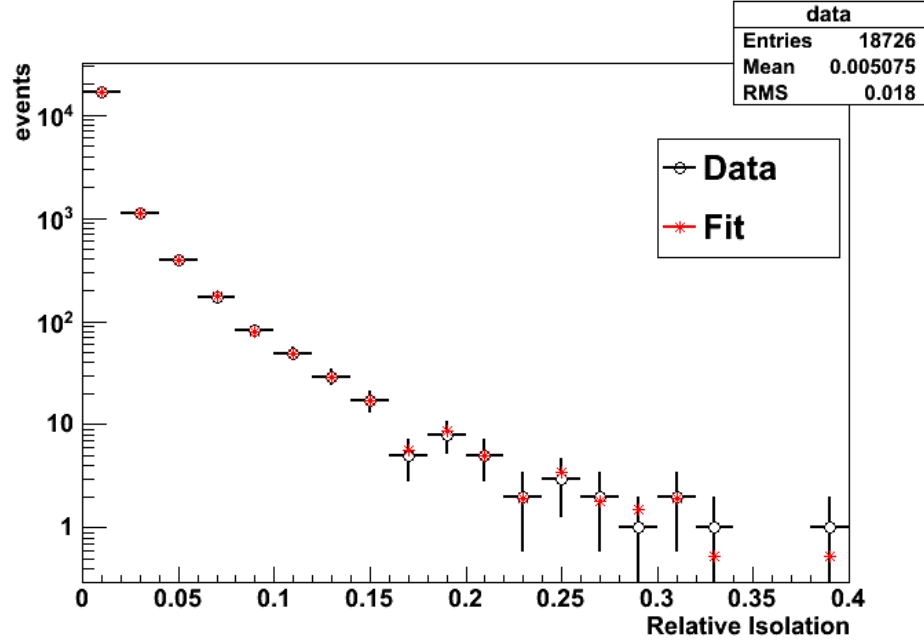


Figure 7.4: Results of template fit to data. The relative track isolation distribution for the data sample is shown as the hollow circles (black), while the TFractionFitter-determined template fit, using a combination of the signal and background templates, is shown as the asterisks (red). The fit agrees very well to the data; small deviations are visible but are contained within the data error bars.

was calculated to be 0.0016 ± 0.0020 . However, the signal fraction over the full range is not useful, because this includes background events that would fail the tracker isolation selection cut. Therefore, the tracker isolation cut is effectively reapplied by only looking at events which pass the cut. In effect this is done by integrating the number of signal and background events below the threshold and redetermining the signal and background fractions. This is represented in the formula

$$f_{sig}^{thresh} = \frac{f_{sig} \times n_{sig}^{thresh}}{f_{sig} \times n_{sig}^{thresh} + f_{bg} \times n_{bg}^{thresh}}$$

which in essence scales the signal fraction according to the relative number of events below the threshold. In this case f_{sig} and f_{bg} are the signal and background fractions determined from the full isolation spectrum, while f_{sig}^{thresh} is the signal fraction below the isolation threshold. n_{sig}^{thresh} and n_{bg}^{thresh} are the integrated signal and background event numbers

below the threshold. Since the barrel and endcap thresholds for relative tracker isolation are different, the integration and fraction redetermination was performed with each threshold separately. However, the differences in the final fractions were negligible (non-existent within significant figures), so using the barrel threshold vs. the endcap threshold was determined not to be a significant factor. The signal fraction below the threshold was determined to be 1.000 ± 0.014 , and the background fraction 0.000 ± 0.002 . It is expected that the background fraction below the cut threshold would be lower than over the entire spectrum, since more of the background distribution was above the cut threshold than the data distribution. The calculated values therefore agree with the expectation in that sense.

The signal and background fractions of the data sample determined both from the full isolation spectrum and from the spectrum below the cut threshold are shown in Table 7.3.

Table 7.3: Signal and background fractions of data sample, for both the full relative tracker isolation spectrum and for the part of the spectrum below the cut threshold.

	Full spectrum	Below threshold
Signal fraction	0.998 ± 0.014	1.000 ± 0.014
Background fraction	0.0016 ± 0.0020	0.000 ± 0.002

The background fraction for the data sample measured below the isolation cut threshold translates to a event count of 0 ± 16.8 events. (It should be recalled that this method is only considered valid for estimating QCD background.) The relative uncertainty on the number of signal events corresponding to the error (16.8 events) is 0.2%.

7.3.3 Sideband Subtraction

An additional method of estimating background was attempted, to cross-check the other two. This method involved fitting the invariant mass peak with a lineshape to model the signal plus one to model the background, in the standard sideband subtraction method. This is similar to the template method used in the previous section, in that a distribution

is fitted to a composition of signal and background contributions. However, the sideband subtraction method uses a different variable distribution, the invariant mass peak, as well as a different set of components with which to fit the distribution. Getting the same results with such different methods adds strength to the claim that the results are accurate. The name “sideband subtraction” refers to taking the distribution in the areas away from the peak, the “sidebands”, to represent the background part of the contribution, extrapolating it into the peak area, and subtracting it from the total distribution to leave only the signal contribution.

The data invariant mass spectrum is fitted with the same lineshapes used to determine the efficiency with the tag-and-probe method (see Section 6.2.2). That is, the data is fit with the $Z \rightarrow e^+e^-$ spectrum shape taken from the generator-level, convolved with a Crystal Ball function times a Gaussian function, which represents the signal, while using an exponential function to represent the background. The resulting fit is shown in Fig. 7.5. Fig. 7.5a shows the fit in linear scale, while Fig. 7.5b shows the same plot in logarithmic scale. The total composition of lineshapes is shown as the solid line. The background contribution is shown as a dotted line, and is only visible in Fig. 7.5a, the linear-scale plot. In logarithmic scale the exponential contribution is drawn around 10^{-3} and hence is not displayed on the logarithmic-scale plot.

In general it appears that the agreement of the fit with the actual data points is quite good. However, there are certain features of the fit that bear explaining. In particular, at the lower and upper tails of the peak, the fit appears to be higher than the data points, and at the upper end an “uptick” in the fit curve appears. This is due to statistical fluctuations in the generator-level $Z \rightarrow e^+e^-$ spectrum histogram. That histogram is very finely binned, and small fluctuations affect the shape of the final fit. The discrepancy at the lower end is likewise due to the shape of the generator-level histogram. The contribution of the background function to the overall fit is negligible. In addition, there appears to be a “shoulder” in the data points between about 75 and 80 GeV, as well as a fluctuation around 108 GeV. At this point in time these features in the shape are not understood but have been seen in other analyses as well. However, since they involve relatively few events, and because the analysis does not largely depend on the shape of the spectrum, for this analysis

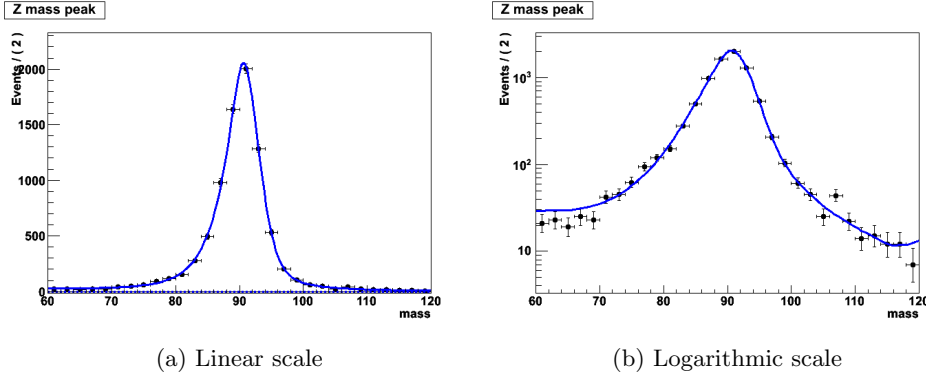


Figure 7.5: Fit of $Z \rightarrow e^+e^-$ invariant mass peak for the purpose of background subtraction. (a) shows the fit in linear scale, while Fig. (b) shows the same plot in logarithmic scale. The peak is fit with a combination of lineshapes representing the signal, the generator-level lineshape convolved with a Crystal Ball function convolved with a Gaussian, and a Gaussian lineshape representing the background. The total composition of lineshapes is shown as the solid line. The background contribution is shown as a dotted line, and is only visible in (a), the linear-scale plot. In logarithmic scale the exponential contribution is drawn around 10^{-3} on the vertical scale and hence is not displayed on the logarithmic-scale plot.

they can be ignored.

The composition of signal and background functions to the overall lineshape gives the number of background events to be 0 ± 14 events.

7.3.4 Comparison of Background Subtraction Methods

The three methods of background estimation detailed in the previous sections have yielded roughly consistent results, summarized in Table 7.4. It is interesting that both of the methods that looked at data arrived at a total of 0 background events. The template method only gives an estimate for QCD events, but the estimate from invariant mass fit applies to all sources of background. These values of zero inherently look suspicious; a value of 0 anywhere in a fit means that something has hit its limit in the optimization algorithm. However, investigations into the two methods have yielded nothing inherently faulty. With the template method in particular it took many iterations of the criteria to obtain a “background-like” sample large enough to work with, supporting the idea that

the selection cuts are very effective at eliminating background. In addition, more than one independent method of analysis is giving the same result. Furthermore, the errors on both of those numbers are approximately the value of the background predicted by Monte Carlo simulation. Therefore, the actual background number could be 0 or 19 or somewhere in between and still be basically consistent with all three methods. For the purposes of this analysis, the most conservative values obtained with any of the three methods were taken for both the estimated number of background events and the error estimate. This yielded a background estimate of 19 events from the Monte Carlo estimate, and an error of 17 events from the template method.

Table 7.4: Summary of background estimates from the three different methods used in this analysis. All three methods give roughly consistent results. In general the most conservative values have been used.

Method	Background events	Error (events)
Monte Carlo	18.5	N/A
Template Method	0	16.8
Sideband Subtraction	0	14

7.4 Estimation of Systematic Uncertainties

7.4.1 Introduction to Error Analysis

When making any sort of measurement it is essential to know how accurate the result is. This accuracy can be quantified in the “error” or “uncertainty” on a measurement, meaning the bounds around the measured value within which the true value can be assumed to lie. A measurement with a smaller error is therefore a more accurate or “better” result, and hence calculating the error on a result is an extremely important part of any analysis.

An uncertainty on a quantity can be expressed in two different ways. The first way is as an absolute uncertainty, with the same units as the quantity, for example “20.0 +/- 0.2 grams”. The second way is as an uncertainty relative to the value of the quantity itself,

such as “20.0 grams \pm 1%”. The “1%” comes from the value of the absolute uncertainty divided by the value of the quantity: 0.2 grams/20.0 grams = 0.01 = 1%. The relative uncertainty is therefore unitless.

In order to combine the effects of different (independent) uncertainties, it is necessary to add them in quadrature, as shown:

$$\left(\frac{\delta X_{tot}}{X_{tot}}\right)^2 = \sum_{i=1}^N \left(\frac{\delta X_i}{X_i}\right)^2 = \left(\frac{\delta X_1}{X_1}\right)^2 + \left(\frac{\delta X_2}{X_2}\right)^2 + \dots + \left(\frac{\delta X_N}{X_N}\right)^2$$

where X is the value of the given quantity and δX is the absolute error, making $\frac{\delta X}{X}$ the relative error. Here, *tot* denotes the total uncertainty and $i = 1, 2, \dots, N$ the uncertainties from individual sources. Only the relative uncertainties can be combined in this way, due to the lack of different units. This formula may be understood conceptually by analogy to the x-y coordinate plane, in particular the formula for a given point’s distance to the origin. In the same way the square of the total distance from zero is given as the sum of the squares of the (independent) component distances, the square of the “total uncertainty” is given by the sum of the squares of the individual (independent) component uncertainties. Instead of calculating a distance in coordinate space, this distance is calculated in “uncertainty space”.

In high energy physics the sources of uncertainty are generally divided into two categories: statistical and systematic. Statistical error arises from the fluctuations possible when counting elements in a population. The higher the number of elements used in the study, the more accurate the measurement is likely to be. In a high energy physics analysis this corresponds to the number of events used. The relative statistical error on a measurement is equal to the square root of the total number of events divided by the total number of events:

$$\frac{\delta x}{x} = \frac{\sqrt{N}}{N}$$

Where x is the quantity being measured and N is the total number of events used. In our case, the statistical error on the final result is given by

$$\frac{\sqrt{8453}}{8453} = 1.1\%$$

The statistical error is fixed for a given number of events; the only way to reduce the statistical error is to analyze more events.

The second type of error, systematic error, encompasses everything else that may inadvertently affect the result of the analysis. The particular sources are specific to the analysis in question, ranging from slight differences in theoretical predictions, reflecting our incomplete knowledge of the subject, to possible miscalibrations of the detector, to possible biases inherent in reconstruction or identification algorithms. The error is evaluated by changing something in the analysis and seeing how much the result itself would change. Some changes have very little effect, while some have larger effects. In general, it is only necessary to account for the larger effects, “larger” being defined by the scale of the statistical uncertainty. If the systematic error due to a particular source is much smaller than the overall statistical error, its contribution to the total uncertainty is relatively very small, and it can safely be ignored. It is therefore standard to try to adjust an analysis if necessary such that its systematic error is at most about the same as the statistical error. Such corrections might include using a reconstruction method whose errors are shown to be lower, or doing a more thorough analysis in which case some errors can be reduced.

7.4.2 Systematic Uncertainty due to Luminosity

As one of the necessary components in the cross-section calculation, any uncertainty on the measurement of the luminosity is relevant to the final result. The luminosity is measured by the forward hadronic calorimeter (HF) as explained in Section 3.3.8. For the 2010 dataset, the luminosity measurement has an accepted relative uncertainty value of 4%.

7.4.3 Systematic Uncertainties from Theory

Uncertainties in theoretical predictions do not necessarily play a role in an experimental measurement. However, the theoretical models do affect the calculation of the acceptance (see Section 6.2.2). The acceptance must inherently be obtained from Monte Carlo simulation, making it dependent on the individual model used. Since there are many different possible corrections to the given Monte Carlo modeling, all of these variations should be taken into account. The official CMS $Z \rightarrow e^+e^-$ analysis calculated uncertainties in the acceptance due to various elements of theory, and these are given here.

The Monte Carlo simulated data sample used in the analysis does not include corrections from resummation or NNLO QCD corrections. There are tools that account for these corrections, however, namely the ResBos event generator, so this generator was used to calculate the uncertainty due to the missing corrections.

Since calculations are generally done up to a fixed order and depend on the renormalization and factorization scales, varying these scales can estimate the uncertainty due to the missing higher-order calculations. The scales were therefore varied each way by a factor of two and the maximum difference in results was taken, using FEWZ to calculate the cross section.

There are several well-known parton distribution function (PDF) sets in existence, and the output of simulations using these PDFs vary slightly from one another. Three different PDF sets were used: CT10, MSTW2008, and NNPDF2.0. The value of the acceptance with each PDF was calculated in order to determine the spread between the three sets. In addition, the error function corresponding to each PDF was evaluated, and these were combined with the acceptance spread over the PDFs.

The Monte Carlo simulation sample used for the analysis also does not fully account for next-to-leading-order (NLO) electroweak corrections. In addition, the existing corrections, namely final-state radiation, may not be completely accurate. To deal with these sources of uncertainty, a separate generator, HORACE, was used to make the full set of corrections. The output was then compared with the standard simulation sample to estimate the uncertainty due to the missing corrections.

The uncertainty due to the PDF set used was found to be 0.9%, while the rest of the sources contributed a total of 1.4%, for a combined value of 1.7% relative uncertainty.

7.4.4 *Other Sources of Systematic Uncertainty*

In addition to the analysis-independent uncertainty due to luminosity, and the uncertainty in the acceptance from the theoretical modeling, there are several sources of uncertainty specific to this analysis that also need to be accounted for.

Electron Energy Scale

A typical source of systematic uncertainty is the energy scale of the physics objects used in the analysis, depending on the type of physics object. The electron energy scale is determined by the accuracy of the electromagnetic calorimeter (see Section 3.3.5) at measuring the energy deposited by the particles passing into it. The response of the ECAL is measured using a test beam setup: particle beams of known energies are directed at the calorimeter material, and the response is recorded. This response is then correlated with the original particle energies. However, the correlation may not be completely accurate. Scaling the electron energy by an amount slightly less than or greater than 1 and then observing the resulting change in the number of $Z \rightarrow e^+e^-$ events gives an estimate of the systematic error due to the electron energy scale. Uncertainties of 2% and 3% for the ECAL barrel and endcap respectively were given as conservative values. The electrons' energy were then multiplied by 1.02 or 1.03 for the upper-end calculation, and 0.98 or 0.97 for the lower-end calculation, according to their position (barrel or endcap). Scaling the energy up resulted in finding 0.82% more $Z \rightarrow e^+e^-$ candidate events. On the other hand, scaling the energy down caused the loss of 1.07% of the events. These two changes were averaged to give an overall relative error of 0.95% for the final result.

Monte Carlo Sample for Efficiency

As discussed in Chapter 4, any given Monte Carlo simulation does not necessarily give a completely accurate model of particle interactions. Therefore, since part of the efficiency calculation was based on Monte Carlo simulation samples, it is quite possible that the numbers derived from simulation are not fully accurate. This must be accounted for as a systematic uncertainty. This accounting is done by redoing the calculation using samples from different Monte Carlo generators. In addition to the standard POWHEG generator sample, two samples generated by PYTHIA were used, Tune Z2 and Tune D6T. The “Monte Carlo true” and “Monte Carlo tag and probe” efficiencies were evaluated with each sample. The difference was quantified by taking the maximum spread between the values for each of the three samples and dividing that number by 2. The maximum difference in efficiency

occurred between the POWHEG and PYTHIA D6T samples, with a difference of 2.37%. The total relative systematic uncertainty that can be attributed to variation in the Monte Carlo sample is therefore 1.19%.

Fitting for Efficiency

The efficiency calculations in Section 6.2.2 had associated systematic errors due to the goodness of the fit, which are the errors shown in Table 6.6. The error on the final event selection efficiency is 0.0047 in terms of absolute error, or 0.76% relative error.

Background Subtraction/Modeling

As described in Section 7.3.4, the uncertainty on the number of background events was taken to be 17 events, from the template method. This produces a relative uncertainty on the number of signal events (and hence final cross section) of 0.2%.

7.4.5 Summary of Uncertainties

A summary of the uncertainties on the final cross section resulting from various sources is shown in Table 7.5. The luminosity contributes 4% of uncertainty, statistics 1.1%, theoretical systematics 1.7%, and experimental systematics a total of 1.7%.

Table 7.5: Summary of relative uncertainties from sources including luminosity, theoretical uncertainties, and experimental uncertainties.

Source	Value
Luminosity Uncertainty	4%
Statistical Uncertainty	1.1%
Theoretical Systematics	1.7%
Experimental Systematics	1.7%
Electron Energy Scale	0.95%
MC Sample for Efficiency	1.2%
Efficiency Fitting	0.76%
Background Subtraction/Modeling	0.2%

Chapter 8

RESULTS**8.1 *Cross Section Measurement***

Cross section calculation

8.2 *Comparison to Theory***8.3 *Comparison to Previous Experiments***

Chapter 9

SUMMARY AND CONCLUSIONS