AGH University of Science and Technology
Faculty of Physics and Applied Computer Science

Masters thesis

Jonathan Aguilar

major: **technical physics**

specialisation: **nuclear physics**

Luminosity detector at ILC: Monte Carlo simulations and analysis of test beam data

Supervisor: **Doktor Bogdan Pawlik**

Cracow, February 2012

Aware of criminal liability for making untrue statements, I declare that the following thesis was written personally by myself and that I did not use any sources but the ones mentioned in the dissertation itself.

The subject of the master thesis and the internship by Jonathan Aguilar, student of 5th year majoring in technical physics, specialisation in nuclear physics

The subject of the master thesis: **Luminosity detector at ILC: Monte Carlo simulations and analysis of test beam data**

Supervisor:	Dr Bogdan Pawlik
Reviewer:	Prof. dr hab. Władisław Dąbrowski
A place of the internship:	WFiIS AGH and IFJ-PAN, Kraków

Programme of the masters thesis and the internship

1. First discussion with the supervisor on realization of the thesis.

2. Collecting and studying the references relevant to the thesis topic(s).

3. Simulation:

   - Creating a stand-alone model of LumiCal...,

   - developing software to analyze the simulation data...,

   - discussion with the supervisor focused on the results obtained...,

   - preparation of the internship report.

4. Test beam:

   - Development of SensorMeas software in LabVieW...

   - adaptation of SensorMeas to test bench at AGH...,

   - application and further development of SensorMeas at test beams...,

   - analysis of test beam data and simulation of test beam

5. Final analysis of the results obtained, conclusions – discussion with and final approval by the thesis supervisor.

6. Typesetting the thesis.

Dean's office delivery deadline: 28 February 2012

Na kolejnych dwóch stronach proszę dołączyć kolejno recenzje pracy popełnione przez Opiekuna oraz Recenzenta (wydrukowane z systemu MISIO i podpisane przez odpowiednio Opiekuna i Recenzenta pracy). Papierową wersję pracy (zawierającą podpisane recenzje) proszę złożyć w dziekanacie celem rejestracji co najmniej na tydzień przed planowaną obroną.

dr Bogdan Pawlik
Instytut Fizyki Jądrowej PAN

Merytoryczna ocena pracy przez opiekuna:

Temat pracy dotyczy budowanego w międzynarowej współpracy FCAL elektromagnetycznego kalorymetru LumiCal. Kalorymetr jest bradzo ważnym elementem detektora ILD oraz CLIC. Podstawowym celem kalorymetru jest precyzyjny pomiar świetlności przez detekcje przypadków procesu rozpraszania Bhabha, kluczowym kryterium rozpoznania tego procesu jest pomiar energi. W pierwszej części pracy autor, oprócz zwięzłego opisu procesów fizycznych, które maja byc badane lub odkryte, prezentuje alternatywne metody poprawy dokładności rekonstrukcji energi dla różnych konfiguracji płaszczyzn silikonowych sensorów. Druga część zawiera szczegółowy opis oraz wyniki testów prototypu detektora przeprowadzonych w DESY. Obszerny dodatek zawiera instrukcje oraz kody źrodłowe aplikacji użytych do analizy danych MC i zebranych w trakcie pomiarow testowych.

Sumując, praca jest napisana zwięźle i jasno, zawiera wiele informacji ważnych i użytecznych dla kontynuatorów projektu. Oceniam ją jako istotny wkład do projektu budowy detektora.
użyteczne dla kontynuatorów projektu
Końcowa ocena pracy przez opiekuna: 5.0

Data: 26.2.2012r.   Podpis: ....

Skala ocen: 5.0 – bardzo dobra, 4.5 – plus dobra, 4.0 – dobra, 3.5 – plus dostateczna, 3.0 – dostateczna, 2.0 – niedostateczna

prof. dr hab. inż. Władysław Dąbrowski
Wydział Fizyki i Informatyki Stosowanej AGH
Katedra Oddziaływań i Detekcji Cząstek

Merytoryczna ocena pracy przez recenzenta:

The thesis describes author's contribution to development of the luminosity calorimeter for the experiments at the International Linear Collider. It includes two parts; Monte Carlo simulations to optimise the energy resolution of the calorimeter and results of the test beam measurements performed for the prototype modules of the calorimeter. The author has contributed significantly to each of these two areas performing individually well defined tasks.

The thesis is written very well, including a comprehensive introductory overview of the subject, clear definitions of the tasks and clear presentation of the results. It is worth to underline critical evaluation and judgment of the obtained results. In addition, the thesis includes an Appendix comprising documentation on the developed software, which will be helpful for anyone who will be continuing the development. The list of references includes representative and up-to-date papers and reports in the field of interest.

There are very few editorial flaws, like for example an unfinished sentence below formula (1.14), which could have been easily corrected by another proofreading. Some plots, see for example Figure 19, have much too small fonts in the legends and axis labels.

My final assessment of the presented thesis is as very good.

Końcowa ocena pracy przez recenzenta: 5.0

Data: 25.2.2012r   Podpis: ....

Skala ocen: 5.0 – bardzo dobra, 4.5 – plus dobra, 4.0 – dobra, 3.5 – plus dostateczna, 3.0 – dostateczna, 2.0 – niedostateczna

ABSTRACT

This work is divided into two parts. The first part considers efforts to characterize the performance of the luminosity calorimeter for the future International Linear Collider, LumiCal, through Monte Carlo simulation. A geometrical description of LumiCal was written using Geant4, a software package for the description of detector geometries and simulation of their interaction with radiation. This description was created to be more detailed geometrically than previous efforts, in order to model the effects of the geometry on the energy resolution of the calorimeter. It was discovered that unsensored areas in between silicon sensors contribute to energy leakage and degrade the performance of the calorimeter. Several methods are proposed to correct this effect.

The second part of this work concerns preparation test beams, in which prototypes of LumiCal were brought to an electron beam facility, and their performance was measured. Software to control the behavior of the motorized positioner table, which controlled the position of the sensor in the electron beam, was written in LabVIEW. This software was later adapted to accomodate measurements made at AGH, in which a laser source was used to irradiate the detector. Second, the test beam situation was modeled using Monte Carlo simulation, and the data from simulation was compared to that recorded during the test beam. Data from the test beam were analyzed as well. It was found that all electronic components of the LumiCal readout chain performed according to their specifications.

ACKNOWLEDGMENTS

Many people contributed to the completion of this thesis. I would like first to thank my research supervisors, dr Bogdan Pawlik, dr hab. inz. Marek Idzik, and dr hab. Leszek Zawiejski, who took a young researcher and guided him through to success. I must also thank Christian Joram and the Marie Curie network on Particle Detectors for providing me with generous funding, organizing valuable meetings, and introducing me to the other bright, young MC-PAD fellows.

At AGH I would particularly like to thank Szymon Kulis and Michał Dwużnik for their friendship and help – both scientific and otherwise – as well as Dean Andrzej Lenda, who found a way for me to complete my degree, and prof. dr. Bogdan Muryn, for his teaching and overall friendliness.

The members of the FCAL collaboration also deserve my gratitide, particularly my commiserants at test beam after test beam. Adrian, Itamar, Olga, Sandro, and Szymon (again) made measurements run and night shifts tolerable. prof. dr Wolfgang Lohmann and dr. Hans Henschel also deserve special acknowledgment for guidance and supervision during test beams, as does André Sailer for many fruitful discussions concerning the intricacies of simulation.

I would like to thank my family and my wonderful girlfriend, Katie, for their constant love and support while I've been so far away. And finally, I would like to thank all those who read and commented on my thesis. All remaining mistakes are my own.

CONTENTS

LIST OF TABLES

INTRODUCTION

1.1 STATE OF PARTICLE PHYSICS

Modern fundamental physics seeks to understand the physical universe along three "frontiers" (Figure 1). The cosmic frontier attempts to explain the observations of astronomy and the theories of cosmology with modern quantum mechanics. The intensity frontier explores low-probability events such as neutrino interactions and proton decays. Finally, the energy frontier seeks to produce new massive or rare particles whose properties would test the limits of the Standard Model. Particle accelerators operate on this frontier, using known initial of colliding particles to generate and measure the properties of new particles, or to determine coupling constants to increasing levels of precision.
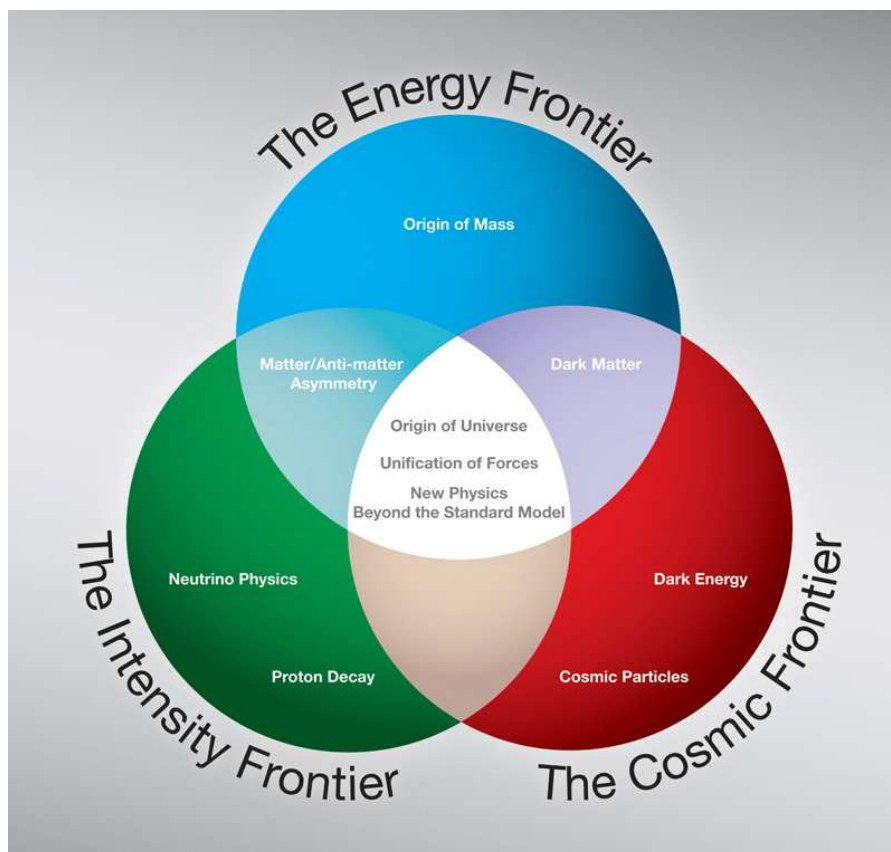
Figure 1: The three modern approaches towards fundamental particle physics.
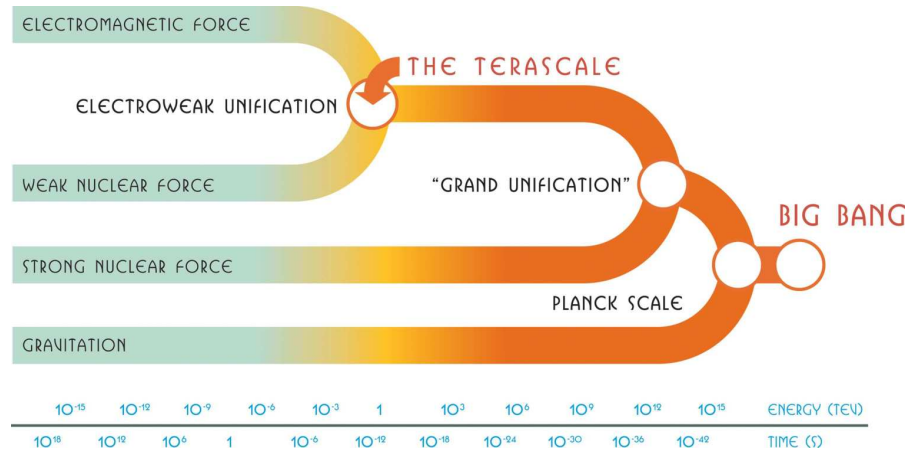
Figure 2: Future physics road map

Currently, high-energy physics has reached terascale energies (Figure 2) at the Large Hadron Collider (LHC)a proton-proton collider at CERN. Since the decomissioning of the Tevatron at Fermilab and the increasing energies at the LHC, the LHC has become the leading machine on the energy frontier. The LHC seeks to test the validity of the Standard Model and point the way to areas where new physics may be found. Protons are a good tool for this because their greater mass allows them to be accelerated in ring accelerators to energies well above what can be achieved using electrons. Furthermore, the strong interactions of their constituent gluons allow decay channels not permitted by lepton colliders. However, this very nature of baryons as composed states introduces important experimental difficulties because their momenta are distributed among their partons in a non-trivial manner. When protons collide, the interaction occurs between one of the valence quarks in each proton, and the initial state of these valence quarks is not well known. The consequence of this is a high background rate and many possible final states with complicated signatures.

1.2 PHYSICS CASE FOR ILC

The proposed International Linear Collider (ILC) solves some of these problems by using leptons (specifically, electrons and positrons), whose point-like nature allows the initial states to be very well described. The ILC will be an e^+e^- collider with a center-of-mass energy $\sqrt{s}$ of 500 GeV, with the option to upgrade to 1 TeV [9]. Although the center-of-mass energy is lower (about 500 GeV for the ILC compared to 14 TeV for the LHC), nearly all of that energy goes into the collision. Particle properties can be measured with much higher precision than in a proton collider, and more rigorous tests of the Standard Model can be performed.

It represents the continuation of a global physics program dedicated to answering fundamental questions about the universe, and in particular, shedding further light on the investigations of the Large Hadron Collider. If the Higgs particle is found, it should be associated with very new physical phenomena that will be accessible to the ILC's energy range: namely, supersymmetry, extra dimensions of space, or even the existence of new forces. The ILC is expected to provide important tests of the results of the LHC in probing physics beyond the Standard Model. An e^+e^- collider is a powerful tool for precision measurements of particle masses and unambiguous particle spin determination [9]. The high energy of the LHC ($\sqrt{s}$ = 14 TeV) gives it a large mass reach, but since it is a hadronic collider it must deal with a large QCD background. The ILC, as an e^+e^- collider, has its strength in high-precision measurement thanks to well-described initial states (including beam polarization [24]).

A great advantage of the ILC is its versatility and the possibility to upgrade after several years of running, if this is required by new physics discovered. It will be capable of operating in GigaZ mode, where it runs on the Z-resonance with high luminosity and with both beams polarized, producing 10^9 hadronic Z decays in less than a year [9]. Secondly, the ILC could also run at the W-pair production threshold for high-precision measurement of the W-mass. Thirdly, both accelerators could accelerate electrons for an e^-e^- collider instead of an e^+e^-. This would help determine the mass of the super-symmetric selectron particle, if it exists within the ILC energy range. Finally, by colliding electrons with a photon beam, the ILC could produce a high energy, high quality photon beam ($e^-\gamma$ or $\gamma\gamma$) collider.

The variety of measurements that will be possible at the ILC are enumerated and described in detail in [15]. They include:

HIGGS PRODUCTION   The existence of the Higgs field is one of the most important theoretical predictions driving the direction of modern experimental particle physics [25]. Its existence would provide a mechanism for electroweak symmetry breaking that would support the validity of the standard model. Several processes available at a lepton collider are posited to result in Higgs production. At low energies, the "Higgsstrahlung" process is expected to dominate where an intermediate Z boson is created, and transfers into a real Z while emitting a Higgs [35].

There are, of course, two other possibilities regarding Higgs searches at LHC: a non-Standard Model Higgs could be found, or no Higgs could be found at all. In that case, the ILC's capability to perform precision measurements of the top quark, Z, and W properties will be instrumental in developing new theories [17].

GAUGE BOSON COUPLING   At the ILC, the process $e^+e^- \to f\bar{f}$, for $f \neq e$, proceeds via the exchange of Z bosons or photons and can be used to measure the couplings of fermions to gauge bosons. This can also be used to measure the weak mixing angle between the W and Z bosons, and to test electroweak symmetry breaking.

TOP QUARK PHYSICS   Determining the mass of the top quark is a critical measurement in testing the limits of the standard model. As the most massive Standard Model particle, it is expected to interact most strongly with the electroweak symmetry breaking mechanism, purported to be the Higgs boson.

SUPERSYMMETRY   Supersymmetry (SUSY) was invented to correct certain failures of the Standard Model, such as the omission of gravity, failure of the energy evolution of the coupling constants to meet at a common point, lack of a candidate particle for dark matter, and very large radiative corrections to the mass of the Higgs boson. SUSY makes a number of predictions that can be tested by the ILC. For each SM particle, it predicts a supersymmetric partner that differs in spin by 1/2. These superpartners have not yet been observed, and this is thought to be because they have greater mass than their SM counterparts. In the case that SUSY is a successful theory, investigation of this mass symmetry breaking will be a priority. Various superpartner decay channels should be able to be measured at the ILC, if supersymmetric theories prove accurate. The ILC should have access to the lowest-energy superparticles.

1.3   ILC DESIGN

The total luminosity required is 500 fb^{-1} during the first four years of operation, and double that during the first phase of operation at 500 GeV. It will have a peak luminosity of $2 \cdot 10^{34}$ cm^{-2}s^{-1}

Figure 3 shows a schematic diagram of the future ILC design. In total it will be ≈32 km long, with two damping rings for low-energy acceleration of electrons and positrons. Acceleration will be performed by approximately 17,000 one meter-long 1.3 GHz superconducting radio-frequency cavities.

A unique feature of the ILC is its ability to produce up to 80% polarized positrons, allowing initial states to be well-known. To generate positrons, the high-energy (≈150 GeV) electron beam is passed through a helical undulator. This generates a beam of photons in the few-MeV range which interacts with a metal plate, producing electromagnetic showers. The positrons are then isolated from the shower and directed to the damping ring. Bunches from the electron and positron sources are shaped and stabilized before injection into the main accelerators. The main linacs accelerate the leptons from
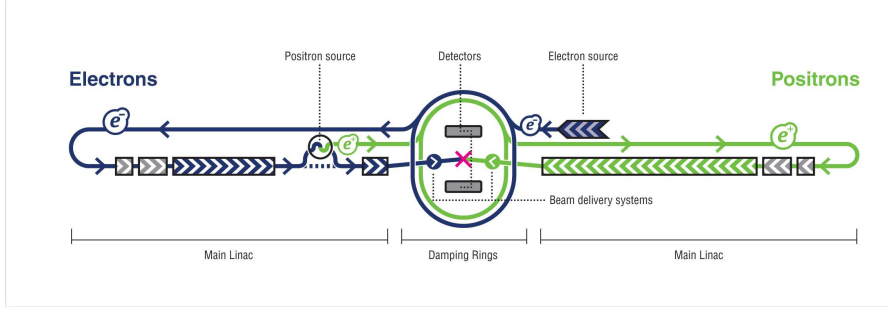
Figure 3: Schematic layout of ILC [28]

15 GeV to their nominal energies of 250 GeV, and guide them to the interaction point (IP) in the center of the detector region. Because a linear accelerator can only have a single (IP), the two detectors chosen will be operated in a push-pull configuration; that is, storage vaults on either side of the IP will store one detector while the other is in use.

The nominal ILC beam parameters can be found in the SB2009 update to the ILC baseline design [30].

1.3.1 *International Large Detector*

The International Large Detector (ILD) concept is a proposed to be a multi-purpose detector which provides excellent precision in spatial and energy resolution over a large solid angle in order to achieve excellent particle track reconstruction. The ILD is one of three detector designs validated by the International Detector Advisory Group in August of 2009 [13]. LumiCal is the luminometer of the ILD, designed to measure the integrated luminosity of the ILC beam.

1.4 LUMINOSITY MEASUREMENT

Relative precision in the luminosity measurement of better that 10^{-3} is required to achieve the goals of the ILC physics program. Small angle Bhabha scattering will be used for this measurement [16]. The integrated luminosity $\mathcal{L}$ is estimated by counting the number of Bhabha events using the well-known Bhabha scattering cross-section, according to the relation $N_{Bh} = \mathcal{L} \cdot \sigma_{Bh}$, so:

$$\mathcal{L} = \frac{N_{Bh}}{\sigma_{Bh}}. \tag{1.1}$$

N_{Bh} must be measured with sufficient precision that $\frac{\Delta\mathcal{L}}{\mathcal{L}} < 10^{-3}$. In this way, cross-sections σ_p for any other process p can be checked by inverting equation 1.1: $N_p = \mathcal{L} \cdot \sigma_p$, where N_p is the number of events of process p measured in the detectors.

1.4.1 *Bhabha scattering*

Bhabha scattering refers to the scattering of electrons by positrons, first described in 1936 [5]. It has been traditionally used to measure luminosity in lepton colliders because of its abundance, well-understood cross section, and large cross section at small polar angles where they will not interfere with large p_T events that are more physically interesting.

To lowest order, the differential cross section as a function of polar angle θ is given (in natural units, with $\hbar = c = 1$) by

$$\frac{d\sigma}{d\theta} = \frac{2\pi\alpha^2}{s}\frac{\sin(\theta/2)}{\sin^4(\theta)} \cong \frac{32\pi\alpha^2}{s}\left(\frac{1}{\theta^3}\right) \tag{1.2}$$

where α is the fine structure constant, and s is the square of the center-of-mass energy in GeV. The approximation is valid for small values of θ. Integrated over the acceptance region for an azimuthally symmetric calorimeter [18], this gives

$$\int_{\theta_{min}}^{\theta_{max}} \frac{d\sigma}{d\theta} = \frac{1.04\text{pb GeV}^2}{s}\left(\frac{1}{\theta_{min}^2} - \frac{1}{\theta_{max}^2}\right) \tag{1.3}$$

The dependence of the cross section on polar angle is demonstrated graphically in Figure 4.
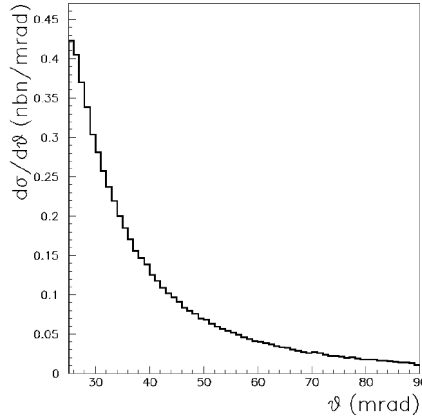
Figure 4: Dependence of $d\sigma/d\theta$, the differential Bhabha cross-section, on the polar angle, θ, at $\sqrt{s}$ = 500 GeV. [33].

Bhabha scattering proceeds by the s- and t-channels, as depicted in Figure 5. At the low angles we are interested in, scattering is dominated by the t-channel.

1.4.2 *Relative error in luminosity*

There are several terms that contribute to the relative error in luminosity:

$$\frac{\Delta\mathcal{L}}{\mathcal{L}} = \frac{\Delta\mathcal{L}}{\mathcal{L}}_{stat} \oplus \frac{\Delta\mathcal{L}}{\mathcal{L}}_{sys} \oplus \frac{\Delta\mathcal{L}}{\mathcal{L}}_{other} \tag{1.4}$$
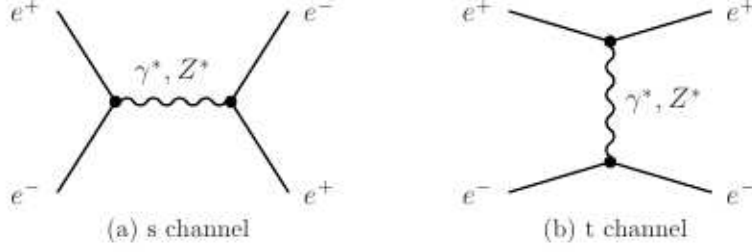
Figure 5: Feynman diagrams of the s- and t-channels of Born-level elastic Bhabha scattering [34].

The statistical contribution is proportional to the variance in the number obseved Bhabha scatterings, N. This is Poisson distributed:

$$\frac{\Delta \mathcal{L}}{\mathcal{L}} = \frac{\Delta N}{N} = \frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}}. \tag{1.5}$$

Therefore, if $\mathcal{O}(10^6)$ Bhabha events will be counted, the required precision in luminosity is $^1/\sqrt{10^6} = 10^{-3}$

The fundamental source of error in luminosity comes from miscounting of Bhabha events:

$$\frac{\Delta \mathcal{L}}{\mathcal{L}} = \frac{\Delta N}{N} = \frac{N_{real} - N_{count}}{N_{real}} \tag{1.6}$$

Counting accuracy is limited by factors such as energy resolution and position reconstruction resolution.

The Bhabha scattering cross-section has a very strong dependence on polar angle [38]:

$$\frac{d\sigma}{d\theta} \propto \frac{1}{\theta^3}, \tag{1.7}$$

making the inner radius of LumiCal perhaps the most important parameter in its construction. It is important to make the inner radius low enough to gather sufficient statistics for luminosity determination, but so low that the background is overwhelming. For some distance d from the interaction point of the ILC, the dependence of $\Delta \mathcal{L}/\mathcal{L}$ on the angle θ_{min} between the beam and the inner radius is given by:

$$\frac{\Delta \mathcal{L}}{\mathcal{L}} = 2\frac{\Delta r_{cluster}}{r_{clus}} \approx 2\frac{\Delta r_{clus}}{d \cdot \theta_{min}}, \tag{1.8}$$

where r_{clus} is the radial position of clusters detected in the calorimeter, Δr_{clus} is its uncertainty, and in the small angle approximation, $\theta_{min} \approx r_{clus}/d$. Because of this strong dependence on polar angle, it is necessary to know the position of the inner radius to within 4 μm [37].

Physics requirements put important limits on the construction of LumiCal. The necessary precision in position reconstruction, especially

because of the sensitivity of the Bhabha cross-section to the inner radius means that the position of LumiCal must be known to a few hundred microns [7].

The actual calculation of integrated luminosity will be performed using not equation 1.1, but instead:

$$\mathcal{L} = \frac{N_{Bh} - \sum_i N_i^{cor}}{\sigma_{Bh}},$$ (1.9)

where $\sum_i N_i^{cor}$ is the number of miscounted events due to various effects. Uncertainties in Bhabha event counting come from beam-beam effects, background from physics processes, calorimeter energy resolution, polar angle resolution, and beam polarization.

1.5 BHABHA SELECTION AND SOURCES OF ERROR

1.5.1 *Bhabha selection*

Particles must have back-to-back topology within the LumiCal fiducial volume and have a total reconstructed energy $\geqslant 0.8E_{beam}$. Matching polar angles for the Bhabha particles is not used because of asymmetric emission of beamstrahlung radiation, as described in [33]. This results in an acollinear final state, illustrated in Figure 6.

However, the reconstructed showers *are* required to be within the fiducial volume of LumiCal on one side, and within $\theta_{min} + 4$ and $\theta_{max} - 7$ on the other (with θ measured in mrad, as before). These criteria are optimized to reduce the impact of beamstrahlung and deflection on Bhabha event counting [19].

1.5.2 *Sources of error in luminosity measurement*

Table 1 lists eight sources of systematic error in luminosity measurement [19]. Theoretical calculations of the Bhabha scattering cross section have been discussed to lowest order in Section 1.4.1, but more precise calculations are the subject of much current research (see e.g. [11]) Energy scaling is discussed briefly in [36]. The other sources of error listed in Table 1 are discussed in [8]. Beam-beam effects and the Bhabha suppression effect are analyzed at length in [33]. This work is primarily concerned with understanding and improving the energy resolution of LumiCal. As demonstrated in Figure 7, under normal ILC operating conditions the energy resolution must be known to better than 3% in order to achieve the target relative luminosity error of better than 10^{-3}. For operation in GigaZ mode, in which the goal is better than 10^{-4}, the requirement becomes more stringent at about 1.5%.
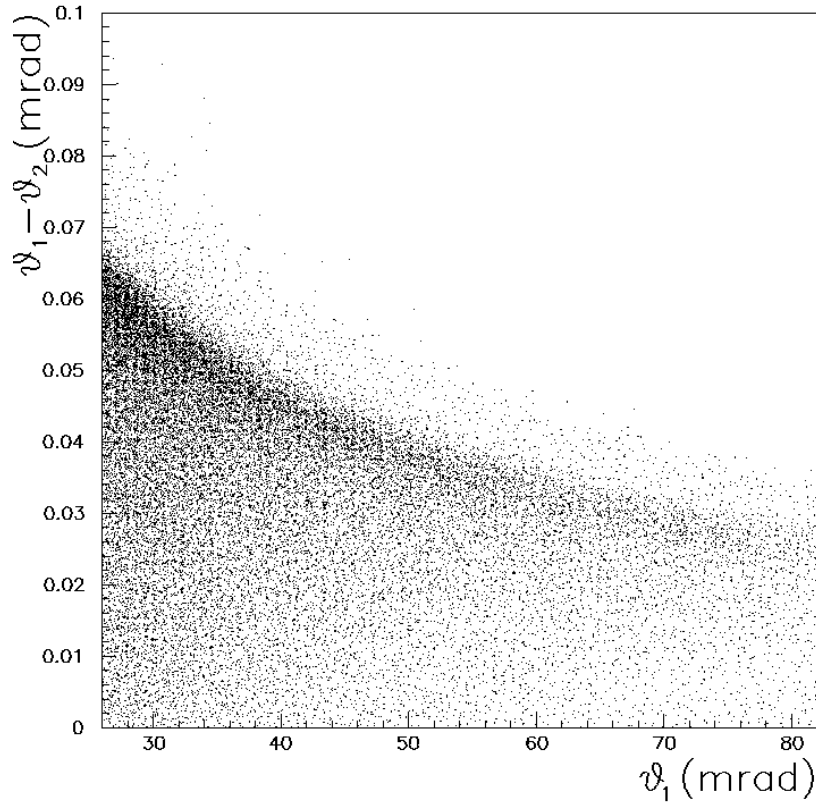
Figure 6: Change in Bhabha scattering final state polar angle due to the deflection induced by the space charge of the opposite bunch as a function of the polar angle at production (from [33]).

Table 1: Summary on systematic erros in luminosity measurement. Errors are assumed to be uncorrelated. Uncertainty of the theoretical cross-section for Bhabha scattering is taken to be as at LEP energies [8].

Source of uncertainty	$\Delta L/L$
Stochastic energy resolution α_{res}	$1.0 \cdot 10^{-4}$
Bhabha cross section σ_B	$5.4 \cdot 10^{-4}$
Polar angle resolution σ_θ	$1.6 \cdot 10^{-4}$
Bias of polar angle $\Delta\theta$	$1.6 \cdot 10^{-4}$
Energy scale	$1.0 \cdot 10^{-4}$
Physics background B/S	$2.3 \cdot 10^{-3}$
Bhaha Suppression Effect (BHSE)	$1.5 \cdot 10^{-3}$
Beam polarization	$1.9 \cdot 10^{-4}$
Total	$3.0 \cdot 10^{-3}$

1.5.3   *Uncertainty in energy resolution*

Uncertainty in the energy resolution of LumiCal plays an important role in the overall error in the luminosity measurement, so the LumiCal detector must be simulated carefully in order to characterize the energy resolution accurately.
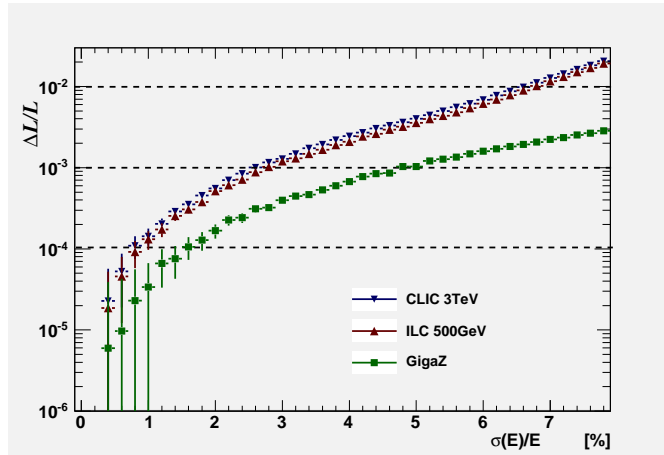
Figure 7: Luminosity error dependence on energy resolution. The goal is better than 10^{-3}.

1.6 CALORIMETRY

Wigmans [40] divides calorimeters into two categories: homogeneous and sampling. In a homogeneous calorimeter, the entire detector is made from a single material which is capable of generating a signal upon interacting with a charged particle. Sampling calorimeters, however, are composed of at least two different materials, one which causes showers and one which generates signals from particle interactions. These are called the passive and active media, respectively. The passive medium is usually some kind of dense metal, such as tungsten or uranium. The active medium is often a scintillator or semiconductor that will generate a signal when interacting with charged particles. This section will focus on sampling calorimeters, and in particular their response to electrons.

1.6.1 *Energy resolution*

The energy resolution of calorimeters is covered in great detail in [23]. The energy released in an electromagnetic shower is proportional to the energy of the incident particle (E_0). The total track length of the shower, T_0, defined as the sum of all ionization tracks due to all charged particles in the cascade, is proportional to

$$T_0(g/cm^2) \propto X_0 \frac{E_0}{\epsilon}, \tag{1.10}$$

where the symbol $\propto$ indicates proportionality and E_0/ϵ is the number of particles in the shower.

An ideal calorimeter of infinite size and perfect collection of deposited energy is nevertheless limited by statistical fluctuations in T_0 due to the random nature of decays. Since shower development is stochastic, the intrinsic energy resolution can be represented by

$$\sigma(E) \propto \sqrt{T_0}, \tag{1.11}$$

so by comparison with the previous equation we see that for an ideal calorimeter, the energy resolution is given by:

$$\frac{\sigma(E)}{E} \propto \frac{1}{\sqrt{T_0}} \propto \frac{1}{\sqrt{E_0}}, \tag{1.12}$$

generally parametrized as

$$\frac{\sigma(E)}{E} = \frac{a}{\sqrt{E}} \tag{1.13}$$

Since this is the fundamental limit for all calorimeters, the parameter a is often quoted as the "energy resolution" for a particular calorimeter, and is dependent on the sampling ratio (the ratio of energy deposited in the active medium to the passive medium).

Real calorimeters are limited in size and construction, and do not have perfect performance. It is therefore useful to add terms to equation 1.13 in order to account for various effects:

$$\frac{\sigma(E)}{E} = \frac{a}{\sqrt{E}} \oplus \frac{b}{E} \oplus c, \tag{1.14}$$

where $\oplus$ indicates addition in quadrature.

The first term is corresponds to stochastic fluctuations in the shower, as previously discussed. From statistical considerations, one can write $\sigma(E)/E \propto 1/\sqrt{N}$, where N is the number of charged particles that cross the active layers of the calorimeter. This in turn is proportional to E_0/t for absorbers of thickness t, so energy resolution can be represented by

$$\frac{\sigma(E)}{E} \propto \sqrt{\frac{t}{E_0[GeV]}} \tag{1.15}$$

The second term in equation 1.14 represents electronic noise from the readout chain, and depends on the characteristics of the readout electronics. Electronics were not included in the simulations presented in this work, so this term was ignored.

The final term in equation 1.14 corresponds to contributions that are independent of the energy of the incident particle. For example, geometric effects within the calorimeter, but also from other sources such as aging, radiation damage, and temperature gradients. None of these other sources were included in simulations, so in the analysis in Chapter 2, this term is used to represent geometric nonuniformities described in Section 1.7.2. Equation 1.14 is therefore simplified to

$$\frac{\sigma(E)}{E} = \frac{a}{\sqrt{E}} \oplus b, \tag{1.16}$$

with only the stochastic term (a) and constant term (b) remaining.

1.6.2   *Calorimetry at LEP*

The Large Electron-Positron Collider (LEP) occupied the current space of the LHC from 1989-2000 and operated at energies of about 200 GeV. Several LEP experiments made use of calorimeters for luminosity measurement, most importantly SiCal at ALEPH [14] and STIC and VSAT at Delphi [10]. SiCal, especially, had a design very similar to LumiCal in fundamental ways and important lessons can be drawn from its performance.

1.6.2.1 *ALEPH SiCal*

The ALEPH (**A**pparatus for **LEP PH**ysics at CERN) experiment used to luminosity calorimeters in a similar configuration to that planned for the ILD detector at ILC. SiCal, or **Si**licon **Cal**orimeter, was first installed in late 1992 and superceded LCAL as the principal luminometer for ALEPH [4]. SiCal was made of two homogeneous cylindrical silicon-tungsten sampling calorimeters surrounding the beampipe at $z = \pm 2.5$ m on either side of the LEP IP. SiCal was twelve layers thick of alternating tungsten absorber and silicon sensors, and covered the angular region from 24-58 mrad. The sensors were made of 300 µm-thick passivated implanted planar silicon. Each of the twelve identical sensor tiles in one layer was divided into a 2×16 array of pads, of dimension 5.225 mm $\times$ 11.25°. The energy resolution of SiCal, calculated from LEP data, was $\sigma_E/E = 34\%/E$, or 4.9% for 45 GeV fully-contained electromagnetic showers. With LumiCal, we hope to achieve $\sigma_E/E \approx 21\%/E$, and better than 1.5% at 250 GeV.

1.7   LUMICAL DESIGN

LumiCal is the luminosity calorimeter of the ILD. Its primary purpose is to count Bhabha scattering events in order to measure the integrated luminosity of the beam interaction. LumiCal's location in the ILD is shown in Figure 8.
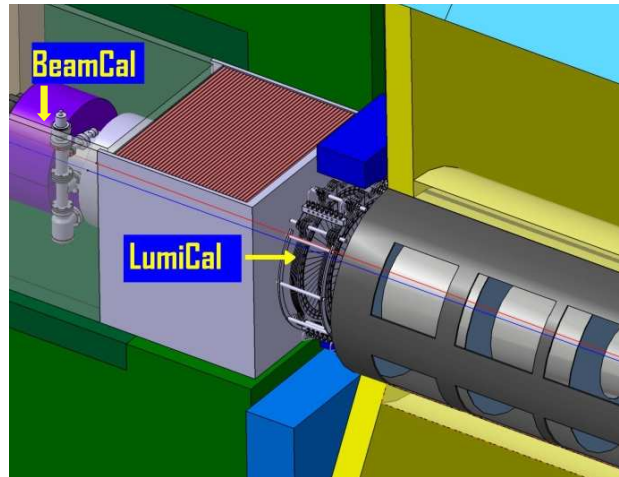
Figure 8: LumiCal and BeamCal in the ILC forward region.

1.7.1   *Requirements*

The performance requirements for LumiCal are detailed in [19, 34]. Most relevant to this work is the requirement that energy resolution be better than 1.5% in order to achieve the target precision in luminosity.

1.7.2   *LumiCal geometrical parameters*

Below is presented a summary of the salient points of the LumICal design. A detailed description of LumiCal can be found in [12]. A CAD drawing is shown in Figure 9. Each LumiCal module will consist of 30 layers of tungsten absorber with silicon sensors and electronics attached. There will also be space for mechanical support and cooling. The tungsten will extend radially from 76 mm for the inner edge to 280 mm (tentative) for the outer. The sensitive radius will extend from 80 mm to 195.2 mm, with the rest of the space on the outside reserved for electronics and support.

The sensor region is divided azimuthally into twelve equal tiles, each covering 30°. Between each tile, there will be an uninstrumented gap of 2.4 mm. Each tile is further divided azimuthally into four sectors, each covering 7.5°, and radially into 64 pads with a 1.8 mm pitch. In order to fit around the beam pipe, the LumiCal modules will be built in two halves and then connected. A schematic of a half-
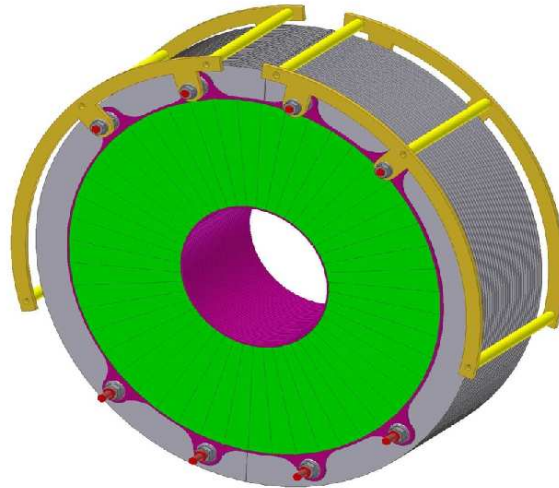
Figure 9: Mechanical model of LumiCal

plane of sensors is shown in Figure 10. The sensors are made from 300 μm-thick Si with Al metallization, produced by the Hamamatsu corporation. They are described in detail in [6]. A prototype sensor tile is shown in Figure 11.
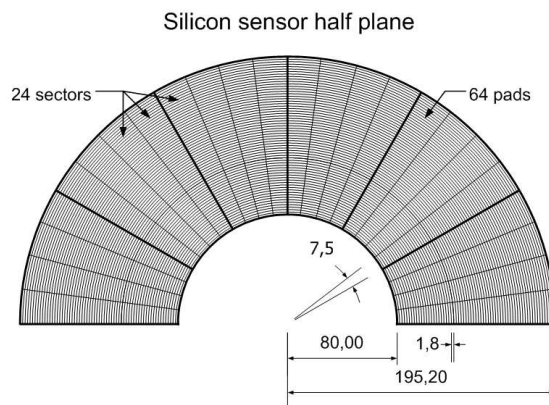
Figure 10: LumiCal sensor half-plane schematic.

1.7.3 *Electronics and readout chain*

The LumiCal readout chain begins with the Si sensor pads. The signal from the pads is collected by the pad metallization and passed through the fanout to the front end (FE) electronics. There, the data is converted into digital format and transmitted to the external DAQ hardware. A detailed description of the custom FE readout developed for LumiCal can be found in [27]
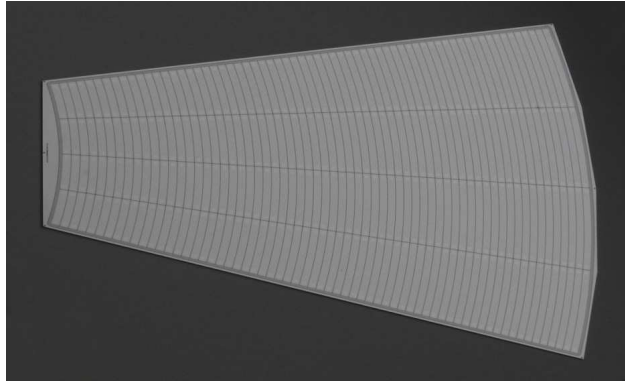
Figure 11: Photograph of LumiCal sensors.

1.7.4 *Geometric effects from LumiCal*

The effect of different geometric parameters on luminosity measurement were examined in [38]. These parameters include, among others, the inner LumiCal radius, radial offset, distance between calorimeters, tilt of the calorimeters with respect to the beam axxis, and longitudinal offset of the calorimeters. The effect of the mechanical gaps between sensor tiles on energy resolution was not discussed, and is the major focus of this thesis.

SIMULATIONS

2.1 LUCAS SOFTWARE

LuCaS, the **Lu**mi**Ca**l **s**imulation program, was constructed using Geant4 [22, 20] (version 4.9.1), a C++ object-oriented toolkit for simulating the interaction of particles and matter. Geant4 makes it possible to describe complex geometries and materials, as model various modes of interaction with different energetic particles. LuCaS also takes advantage of code written for the ILD detector software package Mokka [2], by incorporating the standard physics lists used by other ILD simulations and ensuring that the LumiCal geometrical design stayed within the allowed physical constraints. Nevertheless, LuCaS was not built to directly use the Mokka software package. Since LuCaS depends only on Geant4 and ROOT, it is very portable. This was especially helpful when migrating the simulation to the Cyfronet computing facilities [1].

LuCaS can be run in either interactive mode or batch mode. The LuCaS documentation can be found in Appendix A. In interactive mode, LuCaS uses the standard Geant4 interactive interface to generate particles and explore the geometry. In either modes, the geometry is specified separately in a setup file and loaded at run-time. This allows the geometry to be changed without LuCaS having to be recompiled. The geometry file, as well as other options (listed in the aforementioned appendix), can be selected at run time using command-line switches.

In a typical usage case, LuCaS is run in batch mode to generate a few thousand particles, one at a time, at various energies. An electron, for example, is generated with a well-defined energy at a distance from the LumiCal geometry equivalent to the and an initial angle randomly chosen from a uniform distribution that will direct the particle to an impact point on the surface of the detector. The resulting shower is then propagated according to the physics list loaded by the user, which can be specified in the setup file. The current physics list is "QGSP_BERT", described in [32].

After each particle is generated and propagated, LuCaS checks the folder from which it was launched for a file named "aStopRun". If this file is found, LuCaS will stop the run. If several instances of LuCaS are running simultaneously, this can be used to control the running if the instances are launched from different directories.

The output of LuCaS is a ROOT file containing a ROOT tree. The tree contains the following branches:

- Vertex x, y, and x

- Number of primary particles generated for each run (1)

- Number of hits from each particle shower

- Total energy deposited from each primary particle

- Maximum energy deposited by a single hit in each shower

- Track information: 3-momentum, PDG particle ID

- Hit information:

  - Cell ID and cell center x, y, z position

  - Actual hit x, y, z position

  - Hit energy

  - Hit time of flight

A "hit" refers to an instance of energy being registered in the sensor cells. This tree stores the data for analysis, to be discussed later.

2.2   GEOMETRY

2.2.1   *Motivation*

A prominent aspect of the design, relevant to the majority of Monte Carlo studies performed under this project, are the 2.5 mm uninstrumented mechanical gaps between each of the 12 silicon tiles described in Section 1.7. Energy deposited by primary particles from the interaction point that strike this gap will not be recorded, resulting in poor energy resolution. Alternating layers were to be rotated by 3.75° in order to offset the tile gaps.

No previous studies of LumiCal had implemented these uninstrumented gaps. These simulations modeled the energy resolution as

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}}, \tag{2.1}$$

including only a parameter corresponding to statistical shower fluctutations (a) and not to energy leakage (b in Equation 1.16). The statistical parameter is often the one quoted for the energy resolution of the detector. LumiCal is sufficiently deep that little energy is lost out of the back. It is important to see how strongly leakage through tile gaps affects energy resolution, using the equation

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \oplus b, \tag{2.2}$$

as explained in Section 1.6.1.

2.2.2 *Simulated geometry*

The simulated model for LumiCal was constructed to reproduce as accurately as possible the actual geometry, as described in Section 1.7. Table 2 shows the geometrical parameters of the simulated calorimeter:

The electronics were modeled as a solid layer consisting of a blend of kapton, copper, and epoxy in the proportions specified in [39]. An image of the tile gap as implemented in the simulation is shown in figure 12.
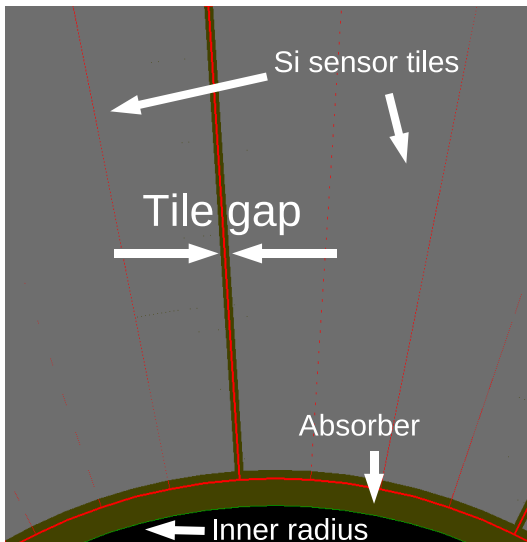
Figure 12: Image of the tile gap taken from the LumiCal simulation

2.3 SIMULATION PROCEDURE

To estimate energy resolution, one calorimeter module was bombarded with monoenergetic single electrons at 50 GeV, 100 GeV, 150 GeV, 200 GeV, 250 GeV, and 500 GeV. 4000 primary electrons were generated per energy and were randomly distributed across the polar angle and azinumthal angle range of LumiCal. The LHEP physics list was used to determine interactions.

Electrons inside the detector interact with the detector material, showering and depositing energy according the physics list. If the energy deposition occured inside a silicon sensor volume, the energy was recorded.

When an electron's track intersected with the first plane of the detector, Geant4 would begin to step through interactions with the calorimeter material and transfer energy from the primary particle until the stepping reached the range cut-off of 5 μm. This meant that the particle had lost sufficient energy that over the next step, it would travel less than 5 μm in the medium. At this point, the particle was

Table 2: List of geometric parameters used in the simulation, and the resulting polar angles.

LumiCal simulation parameters			
Number of radial cells	:	64	
Number of sectors	:	48	
Number of planes	:	30	
Length	:	132.4	[mm]
Distance from IP	:	2502.6	[mm]
Inner radius	:	76.0	[mm]
Outer radius	:	224.5	[mm]
Sensor inner radius	:	80.0	[mm]
Sensor outer radius	:	195.2	[mm]
Cell radial pitch	:	1.762	[mm]
Sector width	:	7.5	[deg]
Gap between absorber plates	:	0.915	[mm]
Air gap	:	0.200	[mm]
Tile gap (2x per tile)	:	1.200	[mm]
Layer ϕ offset	:	3.750	[deg]
Front-end chip thickness	:	1.000	[mm]
Front fanout thickness	:	0.160	[mm]
Back fanout thickness	:	0.235	[mm]
Si sensor thickness	:	0.320	[mm]
Tungsten thickness	:	3.500	[mm]
Total plane thickness	:	4.415	[mm]
Mass of LumiCal (1 module)	:	280.972	[kg]

LumiCal polar angle (θ) acceptance [rad]			
		geometrical	fiducial
Detector θ_{min}	:	0.031	0.033
Detector θ_{max}	:	0.077	0.073
Front-end θ_{min}	:	0.075	
Front-end θ_{max}	:	0.089	

stopped by Geant4 and all remaining energy was deposited into the current volume.

The beam crossing angle was aligned with the central axis of Lumi-Cal. The magnetic field was 3.5 T everywhere. Typical run parameters are given in Table 3.

Table 3: Simulation parameters

Parameter	Value
Particles	Single e-
Azimuthal range [rad]	$\phi \in [0, 2\pi]$
Polar angle range [mrad]	$\theta \in [0.033, 0.073]$
Energies [GeV]	5, 25, 50, 100, 150, 200, 250, 500
Events/energy	4000
Physics lists	QGSP_BERT
Range cut	5 μm

2.4 ANALYSIS OF DATA

Energy resolution was calculated from the simulation data as follows. The program to perform this analysis is available from the lumifun repository at lumifun.ftj.agh.edu.pl. The ROOT tree holds this data for each particle, or "event", generated byGeant4. First each event is checked that it is within the fiducial volume. If not, it is rejected. This results in approximately 28% of the particles being rejected. For a tree with 4000 events per energy, there are between 3850 and 3900 useable events per energy.

The position of the particle is estimated using the shower according to the algorithm outlined in [34], in which the polar angle is reconstructed. In this work, it was important to also reconstruct the azimuthal angle in order to accurately guess which particles were incident on the tile gaps.

Using the cylindrical coordinate system (z, r, φ), for each hit in the shower the hit coordinates were taken to be the center of the cell which recorded the energy. The location of incidence of the primary particle was taken to be the log-weighted energy barycenter of the shower. See Appendix C for the code used to calculate the shower center.

2.4.1   *Calculating energy resolution*

2.4.1.1   *Visible energy and the tile gap effect*

The tile gaps cause LumiCal to behave differently compared to an ideal model with no gaps. Figure 13 shows a comparison of energy deposition in the reference design of LumiCal (3.75° rotation between layers, with gaps), an idealized model of LumiCal ("No gaps" in legend), and a model in which the tile gaps are all aligned ("No rotation"). The "No gaps" design shows a good Gaussian distribution of visible energy due to stochastic fluctiations, with very little leakage. The other two models show Gaussian distributions with long left tails. Figure 14 demonstrates that this loss of energy is due to primary particles incident near the tile gaps for whom part of the energy is not collected.

Energy deposited by 250 GeV e-
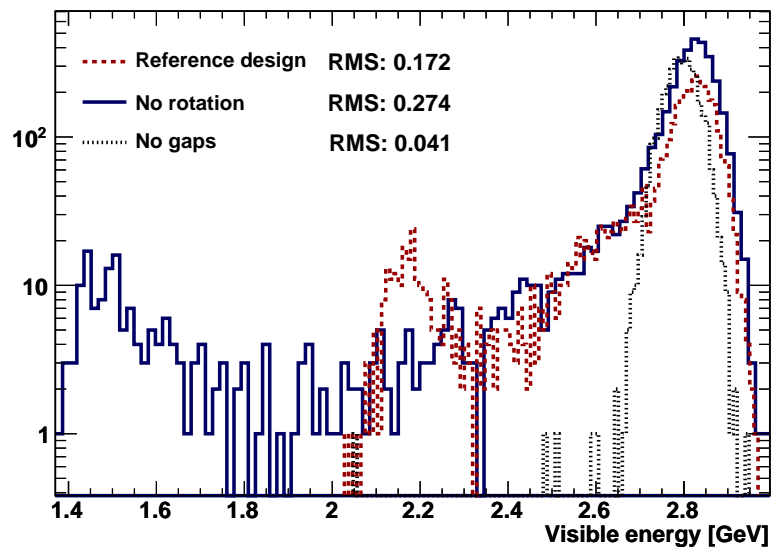
Figure 13: Visible energy at 250 GeV, showing the largest tail for the unrotated geometry and no tail for the gapless, ideal geometry. The y-axis is shown in log scale to emphasize the tails.

This provides motivation for investigating the effect of energy loss on energy resolution - specifically, how the value of the constant parameter in equation 1.16 changes. It would also be interesting to see if the stochastic parameter is influenced.

Figure 15 shows the energy spectrum for different energy electrons for the reference design.

2.4.1.2   *Gap correction methods*

GAP CUT METHOD   The simplest method of correcting for leakage in the tile gaps is use the position information to simply reject particles that are incident on the tile gaps, or incident within a certain distance
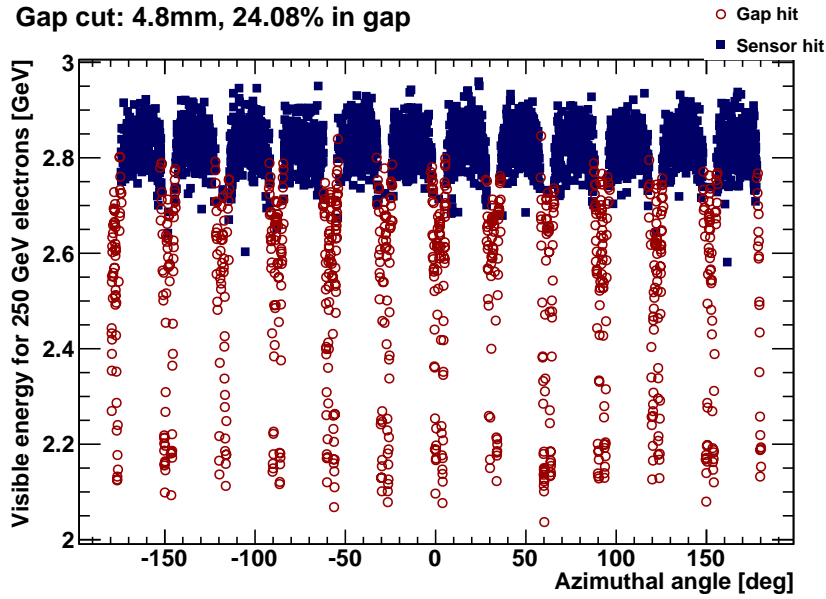
Figure 14: Total visible energy for 250 GeV electrons plotted against azimuthal angle shows the 3.75° separation between the tile gaps of subsequent layers.

(a) 5 GeV

(b) 25 GeV

(c) 50 GeV

(d) 100 GeV

(e) 150 GeV

(f) 200 GeV

(g) 250 GeV

(h) 500 GeV
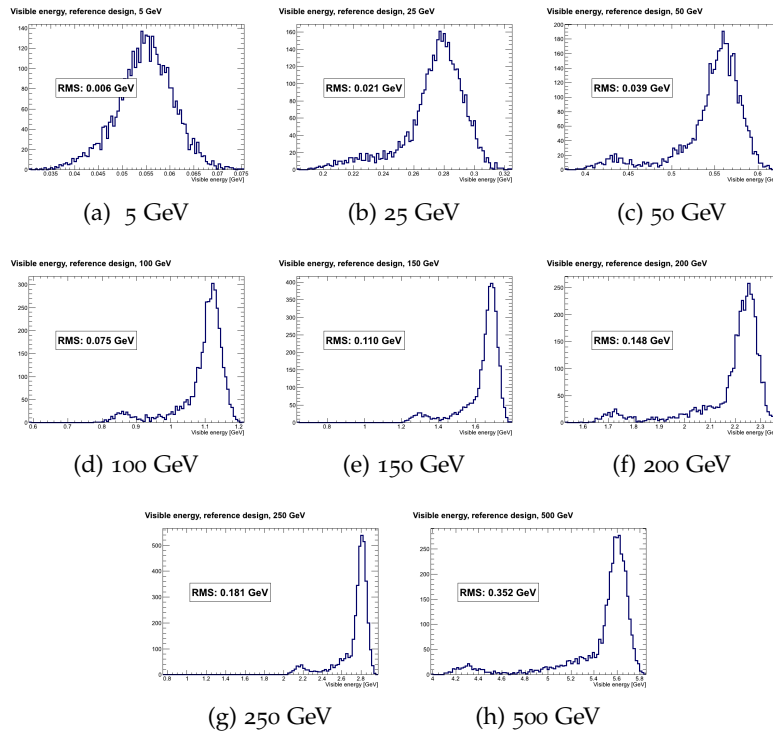
Figure 15: Energy deposited in the detector at each energy, for the LumiCal reference design.

from the closest gap. For example, in Figure 14, all the particles labeled as "gap" particles (red circles) would be thrown out. In the following sections, this will be referred to as the "gap cut" method.

FIT CORRECTION METHOD   A more sophisticated method is to plot the energy deposition against distance from the nearest tile gap (d), and fit the distribution using a d-dependent function. This would introduce a compensation factor for particles near the tile gaps, and would normalize their deposition with respect to particles incident near the center of the sensitive region of the tiles (the fitting procedure is described in more detail below). In the following sections, this will be referred to as the "gap-fitting" method. An example of such a fitted plot is shown in Figure 16.

The shortest distance from the place where a primary particle first intersected the plane of LumiCal was plotted against against the total energy deposited by the shower from that particle in the calorimeter. The energy deposition was also normalized to the average deposition in the center of the sensor tiles. This made it simple to fit the data from different energies with a correction function, based on a Lorentz distribution function:

$$f(d) = \frac{A}{B + (d - d_0)^2},\tag{2.3}$$

where the variable d represents distance from the closest tile gap.

The Lorentz function provided good fitting to the sides, and was combined with an Gaussian function to provide better fitting to the peak:

$$f(d) = A - \frac{B}{(1 + (\frac{d-C}{D})^2} - E \cdot e^{-F \cdot (d-C)^2}.\tag{2.4}$$

The combined function was fit to the data and used to rescale the data points inside the dip. Figure 16 shows an example of such a plot for all electrons with an initial energy of 250 GeV (for contrast, see Figure 14 where the energy deposition is plotted against the actual azimuthal coordinate of the primary particle). In Equation 2.4, parameter *A* corresponds to the normalized energy deposition from one particle at the center of the sensor tiles. Therefore, the fitted functions shown in Figure 17, when divided by *A*, can be understood as the percent of "tile-center" energy that a particle deposits for a given distance from the tile gaps.

Figure 18 shows the ratio of the minimum of the fitting function to the tile center energy, for the reference geometry (left) and the no-rotation geometry (right). The ratio of the reference design is about halfway between the baseline and the no-rotation design, which is expected because a longitudinal cross-section of LumiCal in the gap centers has twice as much material in the reference design as in the no-rotation design, as compared to the tile centers. More importantly, the ratio is more or less independent of the primary particle energy, justifying the attribution of leakage parameter value (which is assumed
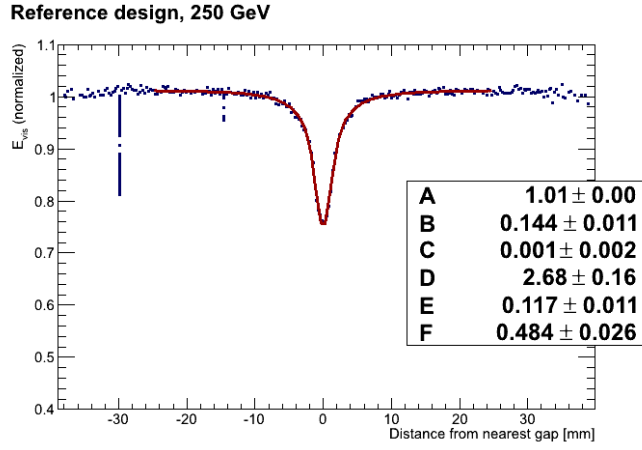
Figure 16: Normalized energy deposition for the reference LumiCal geometry at 250 GeV. The double-gap structure seen in Figure 14 is no longer visible because the independent axis here is distance from the nearest tile gap, rather than ϕ-coordinate. The parameter values correspond to Eq. 2.4 The fitted function is drawn as a solid red line.

(a) 5 GeV

(b) 25 GeV

(c) 50 GeV

(d) 100 GeV

(e) 150 GeV

(f) 200 GeV

(g) 250 GeV

(h) 500 GeV
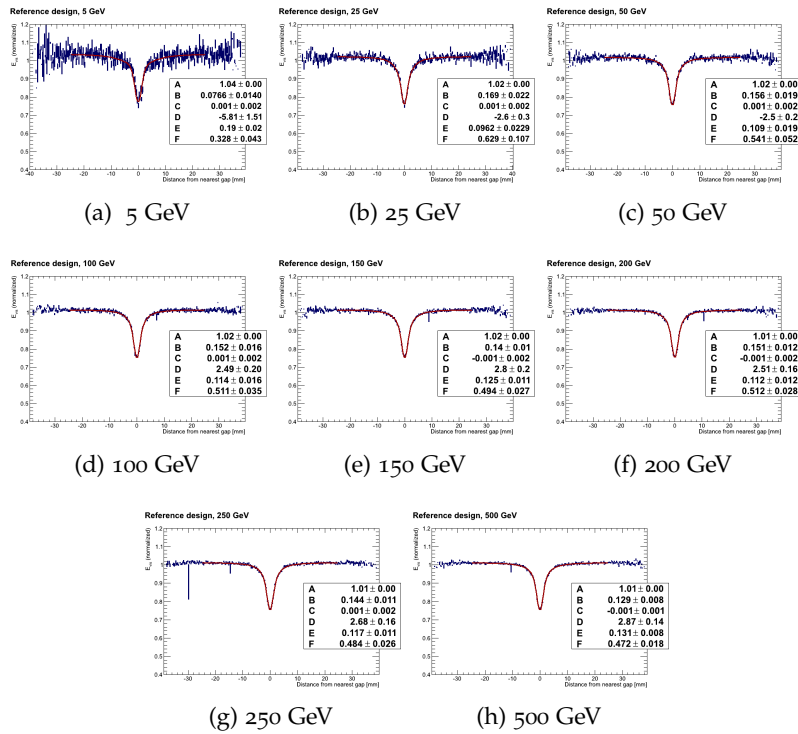
Figure 17: E_{norm}-vs-d plots for all energies. The dips are all approximately of the same magnitude.

in Equation 1.14 to be energy-independent) to energy loss in the tile gaps.

The "flattening" of the visible energy greatly improves the RMS of the visble energy distributions, which in turn results in improved

(a) Reference geometry   (b) No-rotation geometry
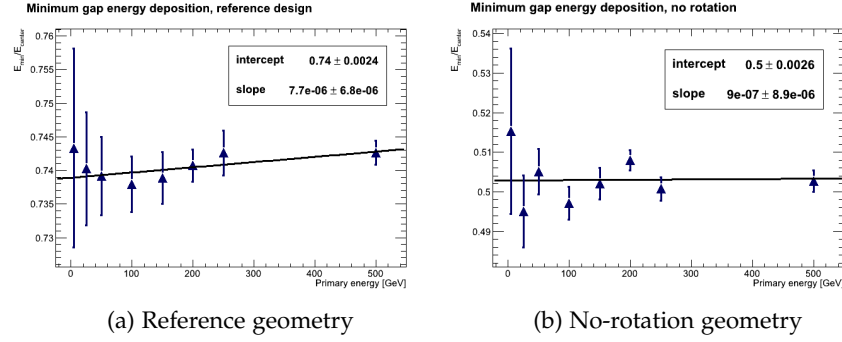
Figure 18: Ratio of the "tile-centered" energy deposition to the gap-centered energy deposition (in practice, the minimum of Eq. 2.4 divided by the parameter "A").

energy resolution. A comparison of the pre- and post-flattening visible energy spectra for particles of each energy is given in Figure 19

The next step in determining energy resolution is to scale the deposited energy by some correction factor in order to approximate the initial energy of the primary particle. If the relationship between primary energy and visible energy is linear, as in Equation 2.5, this can be determined by a simple linear function:

$$E_{vis} = m \cdot E_{gen} + b \qquad (2.5)$$

Figure 20 plots the primary particle energy against the maximum likelihood value (MLV) of the visible energy (flattened or not) of the spectra of the 4000 particles generated at that energy. There is no substantial difference between the flattened energy and raw energy graphs because the presence of a tail in the spectrum does not affect the MLV.

The only affect of the gap-fitting procedure on the energy scaling is to slightly improve the χ^2 value of the fit, as shown in Table 4. This is due to the smaller error bars for the E_{flat} plot, which come from the RMS of the deposited energy spectra.

Table 4: Energy scaling fit parameters and quality.

	E_{vis}	E_{flat}
$1/m$	88.9458 ± 2.42970	88.6014 ± 0.80283
b	$-2.25e\text{-}3 \pm 6.39e\text{-}3$	$-3.01e\text{-}4 \pm 4.96e\text{-}3$
χ^2/NDF	$1.569e\text{-}3$	$4.096e\text{-}2$

In this table, values m and b correspond to the variable from Equation 2.5. The value $1/m$ is given because that is the value by which the observed energy is multiplied to obtain a guess of the primary particle's original energy.
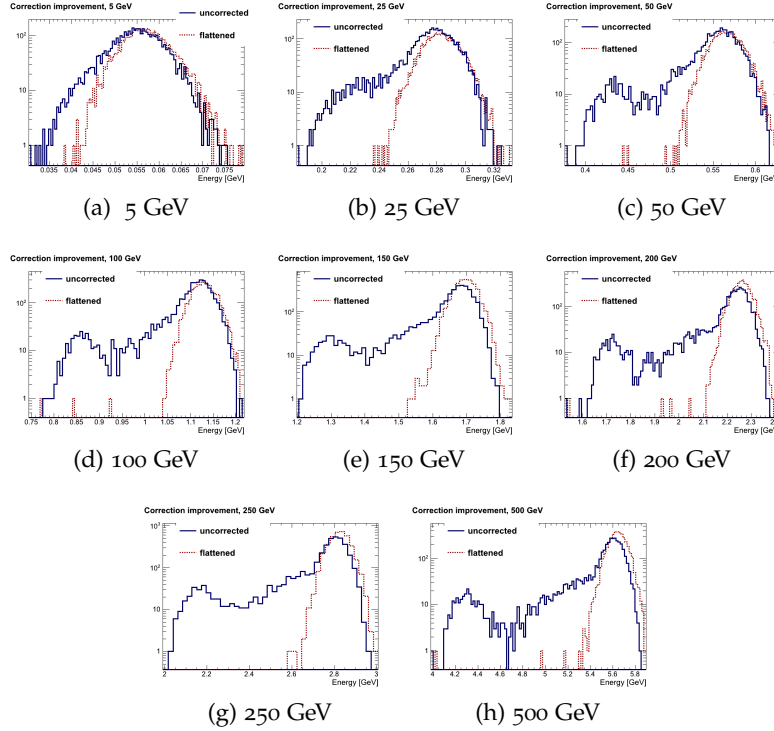
Figure 19: Comparisons of the visible energy and the flattened energy spectra for the reference geometry. The gap-fitting procedure almost completely eliminates the tails present in the visible energy spectra, with a corresponding improvement in RMS values.

In determining energy resolution, however, the reconstructed energy was generated using the correction factor based on the MLVs of the E_{flat} distributions, rather than using the linear functions in Table 4. The data points in Figure 20 are the correction factors for the individual energies. The differences between the correction factors and the scaling factor predicted by Equation 2.5 are given in Figure 21.

2.4.2 Energy resolution

The energy resolution of a calorimeter depends on the precision with which it can collect energy from an incident particle. Over many particles of the same initial energy, the amount of energy collected by the calorimeter will vary. The more narrow the width of the distribution, the better the energy resolution of the calorimeter.

2.4.2.1 Algorithm for reconstructing energy

The algorithm for calcuating energy resolution is as follows. First, the Monte Carlo data were generated as described in Section 2.3, and the resulting ROOT file was loaded into memory. 4000 particles were

Scaling comparison, reference design
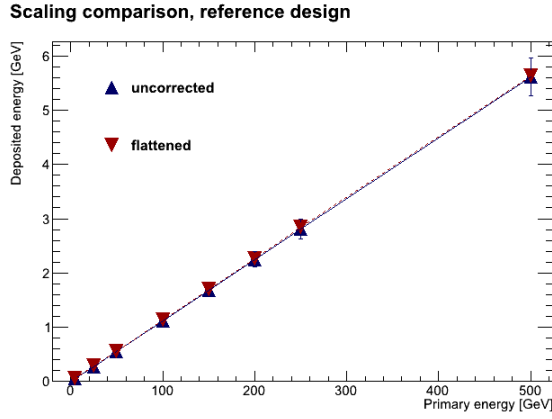
Figure 20: Scaling between visible and generated energy, reference design. Correcting for the tile gaps has no effect because the value used for the y-value is the maximum likelihood value of the deposited energy.

Scaling linearity
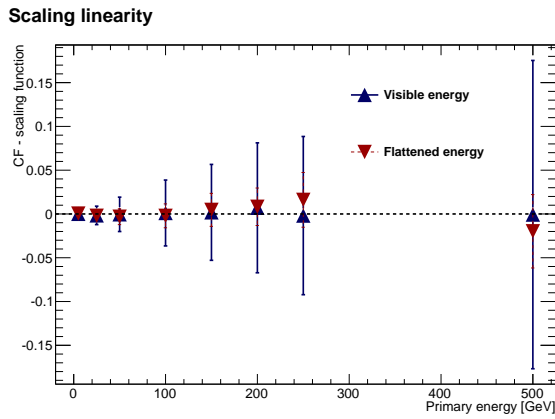
Figure 21: Difference between correction factors (data points on Figure 20) and scaling function (plotted lines on Figure 20). The error bars correspond to the RMS of the E_{vis} or E_{flat} distributions. In this plot only the E_{vis} error bars can be seen because they are much larger.

generated at each of eight discrete energies: 5, 25, 50, 100, 150, 200, 250, and 500 GeV. The following procedure is performed on all particles of a single energy, and is repeated for each energy.

1. Get visible energy (E_{vis}) and position in z, r, and φ.

2. Use position to calculate distance to nearest gap, d.

3. If using a "gap cut", throw out all data less than a set distance x from the nearest gap center, and skip to step 7.

4. If using the "gap-fitting method", sort the E_{vis} values of hits according to their d values, and normalize them to the average of

the largest half of the sorted E_{vis} values. This gives "normalized" E_{vis}, or E_{norm}.[1]

5. Plot E_{norm} against d and fit using Equation 2.4.

6. Divide E_{vis} by $\frac{f(d)}{A}$ to "flatten" the E_{vis} distribution (E_{flat}).

7. Divide the initial generated energy, E_{gen}, by E_{flat}, histogram the results, and take the MLV as a correction factor (CF) by which all E_{flat} values will be scaled.

8. Multiply all E_{flat} by CF to get reconstructed energy, E_{rec}.

9. Histogram E_{rec} values and calculate the RMS. The resolution of LumiCal for this particular energy is $RMS(E_{rec})/E_{gen}$.

2.4.2.2 *Comparison of methods*

Figure 22 shows a comparison of the energy resolution for the following cases:

- ideal, gapless LumiCal geometry,

- no corrections (raw energy deposition),

- 2.5 mm data cut, exactly the width of the gap,

- 20 mm data cut, and

- gap-fitting correction (i.e. the flattening procedure discussed above).

Figure 22 shows that, as expected, the worst case is with no correction, and the best case is with the gapless, ideal LumiCal, which also represents the best possible value of energy resolution. The gap-fitting correction outperforms all other corrections and closely approaches the performance of an ideal calorimeter with respect to leakage, since the larger values of the energy resolution (as given by Equation 1.14) are dominated by leakage at high energies.

Table 5 compares the values and errors of the parameters for the different corrections. The amount of leakage using gap fitting is almost twice that of the ideal geometry's, but nearly *six* times less than without any correction at all. Although the 20mm cut also exhibits improved leakage, using gap-fitting does not result in any loss of data (and therefore poorer statistics for the luminosity calculation).

It can be seen from the table that another effect of the gap-fitting correction, besides reducing the leakage parameter versus no correction, is that it also greatly improves the error on the stochastic

---

1 This normalizes the visible energy to the average visible energy of particles that were incident in the center of the sensor tiles. This was performed in order to automate the fitting and correction algorithm for all energies, as well as to be able to compare parameters across energies.
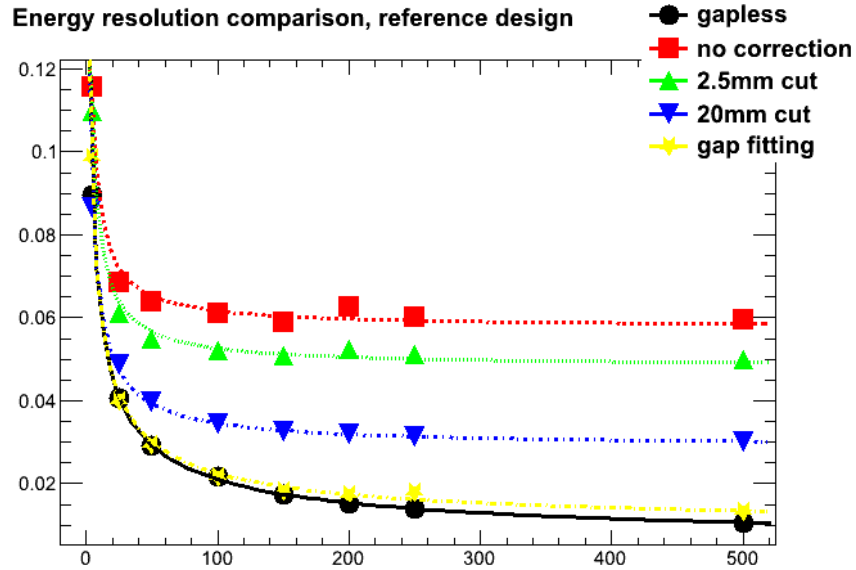
Figure 22: Energy resolution for the reference geometry, for five different correction cases. The 2.5mm cut removes exactly those particles which hit inside the gap.

Table 5: Reference design parameter comparison.

	Stochastic		Leakage	
	Value	Error	Value	Error
No correction	0.2147	0.00322	0.05775	0.02887
2.5mm cut	0.2094	0.00294	0.04829	0.02415
20mm cut	0.1865	0.00258	0.02905	0.01453
gap-fitting	0.2006	0.00141	0.01008	0.00504
gapless	0.2027	0.00131	0.00559	0.00279

parameter. The precision to which this value can be determined is important in determining the precision of the relative luminosity error estimate. However, as the knowledge of this parameter is also a function of statistics, this is not considered a very important effect of the gap-fitting correction method.

2.4.2.3 *Comparison of geometries*

The current design of LumiCal attempts to compensate for the tile gap energy losses by rotating alternate layers. However, if (as this work attempts to show) the losses can be compensated for *analytically*, then perhaps the alternate layers of LumiCal would *not* need to be rotated with respect to each other. This would help simplify the design of the calorimeter electronics and support systems (mechanical, cooling, etc).

As shown in Figure 18, the depth of the central dip in the E_{vis} vs d plots is lower for the no-rotation geometry than for the reference geometry. Unlike the reference design, gap-fitting with the no-rotation design results in similar energy resolution to the 20mm gap cut, but this is expected because wider cuts are required on the reference design in general to acheive the same energy resolution as the no-rotation design. Both gap-fitting and the 20mm cut result in similar energy resolution to the gapless ideal geometry. This is shown in Figure 23.
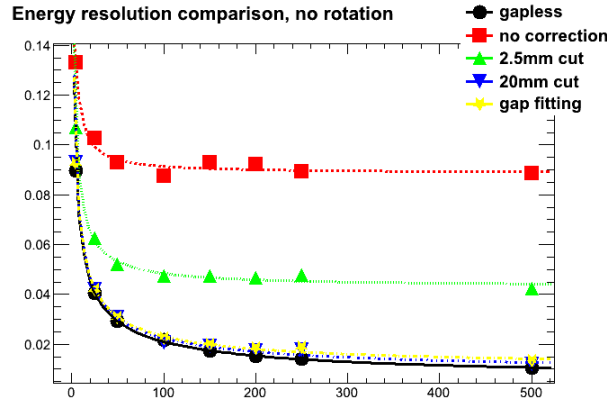
Figure 23: No rotation method comparison

The important result is that the gap-fitting method for both the referece geometry and the no-rotation geometry approach the energy resolution of an ideal calorimeter. Figure 24 shows a direct comparison of the energy resolution for the three LumiCal geometries - gapless, reference, and no-rotation.
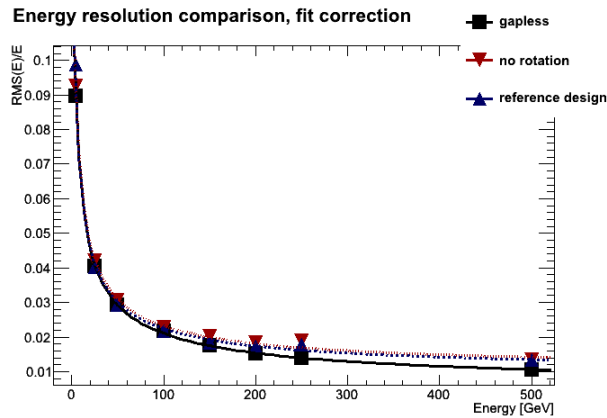
Figure 24: Geometry performance comparison

The fit parameters are given in Table 6. The error on the stochastic parameters for no-rotation and reference geometry are nearly identical.

Table 6: Geometry parameter comparison.

	Stochastic		Leakage	
	Value	Error	Value	Error
No-rotation	0.2053	0.00143	0.01084	0.005418
Reference	0.2006	0.00141	0.01008	0.005040
gapless	0.2027	0.00131	0.00559	0.00279

2.4.3   *Reconstructed position*

The simulations discussed above were performed using the true posi-
titions of the primary particles, known to the simulation program.
A comparison using the true position is valuable because it shows
the effect of only the correction method, and does not include effects
from errors in position reconstruction. The position reconstruction
algorithms are given in Appendix C.

As explained in Section 2.4.2, the radial and azimuthal coordinates
were used to calculate the point of impact of primary particles on
LumiCal. Having good position reconstruction therefore depends on
the reconstruction accuracy of these two values. The algorithm to
do this was adapted from the polar angle reconstruction algorithm
presented in [34]. Figure 25 shows the distributions of $\phi_{rec} - \phi_{gen}$,
and the relationship of the RMS of the distributions with energy. At 5
GeV, the RMS of $\Delta\phi$ is approximately 0.011. For the inner radius of
80mm and the outer radius of 195.2mm, this corresponds a range in
the error on the distance to the gap from $\pm$0.88mm to $\pm$2.15mm, or
nearly the width of the gap.

The gap-fitting function depends strongly on the distance to the
gap; thus, it is expected that a small error in position can lead to a
large error in flattening and therefore in energy resolution. Figure 26
shows LumiCal's energy resolution when the position of the primary
particle is reconstructed from the shower. It can be seen that the energy
resolution is poorer than in the case using the true position. Gap-fitting
still clearly outperforms all correction methods except for the ideal
gapless LumiCal, whose performance does not depend on position
reconstruction at all.

As could be expected, the reconstructed position energy resolution
does not perform as well as the true position energy resolution. In
Figure 27, reconstructed position is compared with true position. In
this case, the best performance is given by the "no rotation" design.
This is because there are more gap particles in the reference design,
and therefore more particles are likely to be given the wrong flattening
factor.

(a) 5 GeV
(b) 25 GeV
(c) 50 GeV
(d) 100 GeV
(e) 150 GeV
(f) 200 GeV
(g) 250 GeV
(h) 500 GeV
(i) Δφ RMS v E
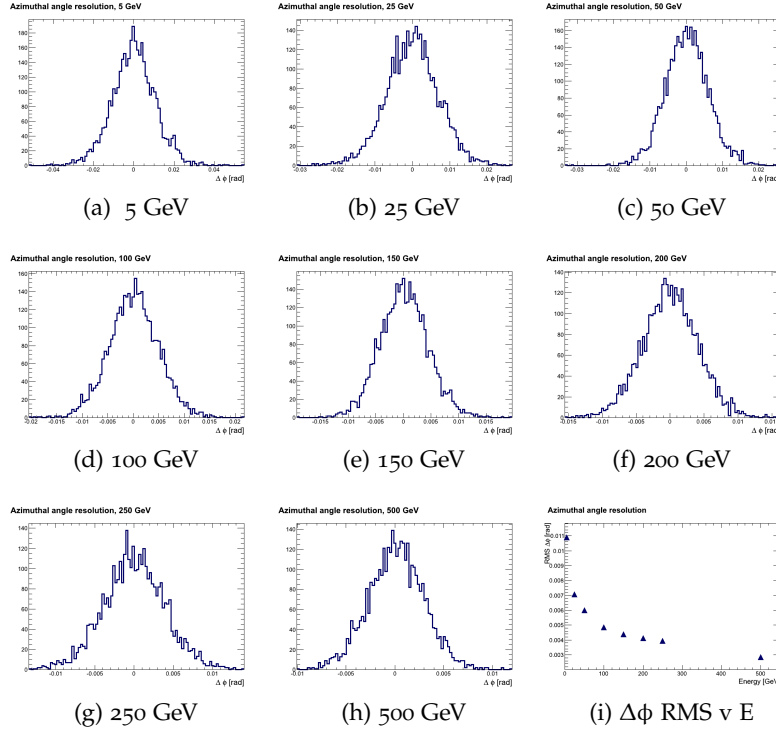
Figure 25: Energy deposited in the detector at each energy, for the LumiCal reference design. Figure 25i shows the change in RMS across energies. The error bars are too small to see.
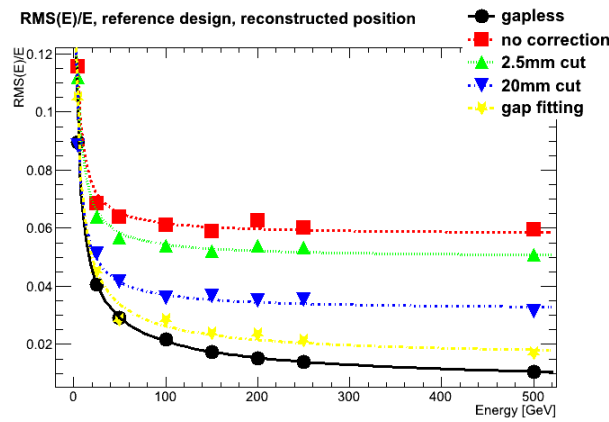
Figure 26: Energy resolution for the reference design, using initial positionr reconstructed from the shower.

Table 7 compares the fitting parameters of different cases. Once again, we see a significant improvement in the leakage parameter over the "no correction" case. The leakage using reconstructed positions is within error of the leakage using true positions.
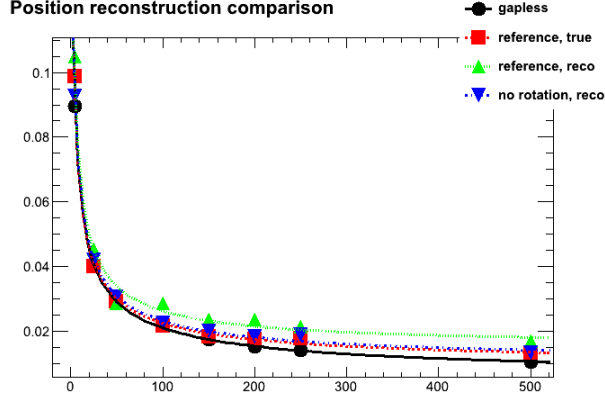
Figure 27: Energy resolution for reconstructed position versus true position as well as the gapless LumiCal.

Table 7: Parameter comparison for different position reconstruction methods.

	Stochastic		Leakage	
	Value	Error	Value	Error
gapless	0.2027	0.00131	0.00559	0.00279
No rotation, true	0.2053	0.00144	0.01084	0.00542
Reference, true	0.2006	0.00141	0.01008	0.00504
Reference, true, no corr	0.2147	0.00322	0.05775	0.02888
Reference, reco	0.2132	0.00166	0.01538	0.00769

2.4.4 *Simulation conclusions*

The gap-fitting correction is a promising method for several reasons. Most importantly, it removes phi-dependence from the energy deposition without losing statistics. This is in contrast to the method developed for preventing leakage with respect to the polar angle, described in detail in [34].

Secondly, it improves the error on the stochastic parameter by more than a factor of 2 over not having any corrections at all; in fact, Table 5 shows that the stochastic parameters for the "no correction" and "gap-fitting" cases are not within error of each other. LumiCal performance assumptions were based on an ideal, gapless design, and the relative error on the stochastic parameter using the gap-fit method is less than 8% larger than that for an ideal model, as opposed to over 200% larger for the uncorrected case.

Thirdly, the improved performance using the gap-fitting method opens the possibility for an unrotated LumiCal design. However, more studies must be made to see the effect of no-rotation on issues like hermeticity before such a design would be adopted. Additionally, the position reconstruction algorithms used in this analysis have not been

optimized. It is possible that by refining these algorithms, the energy resolution will be improved.

In order to have a better idea of the performance of this algorithm during ILC operation, it is important to first optimize the position reconstruction algorithm with respect to radial and azimuthal coordinates and then to redo the analysis.

HARDWARE TESTING

3.1 TEST BENCH

A sensor characterization suite was developed for use at AGH. The sensors and electronics are mounted on a 3-dimensional motorized positioning table (see Section 3.1.2) and connected to the readout system, which can be an oscilloscope or a computer running specialized DAQ software. The radiation source is an infrared pulsed diode laser with a 0.7 mm spot size. The laser is driven by a PDL 800-D picosecond pulsed diode laser driver from PicoQuant. The sensors can be moved into the focal point of the laser by the third axis of the positioning system. The setup is encased in an aluminum faraday cage, which protects from electromagnetic interference and prevents the silicon sensors from light-induced charge generation.

A photograph of the 3-D positioning system is shown in Figure 28.

Figure 28: Protoype testing setup at AGH.

SensorMeas allows the data from the sensors and electronics to be read out by an Agilent MSO7034B oscilloscope and written to the table-controlling computer. Otherwise, the data must be acquired by a separate DAQ system provided by the user.

3.1.1   *Motorized positioners and controller*

LumiCal prototypes can be tested either in the lab, using lasers or a radiation source such as Strontium-90, or at an electron beam facility, as described later. In either case, it is necessary to be able to move the prototype in the two-dimensional plane perpendicular to the beam of incoming radiation. For this purpose, a two-dimensional motorized positioner, model number 8MTF-102LS05, was purchased from Standa Ltd. of Vilnius, Lithuania [29]. An image of the stage is shown in Figure 29. It has a range of 102 mm in both axes, and a step size of 2.5 μm. It is capable of moving in $1/8^{th}$-step increments, but if the power is shut off in an intermediate position the motor will lose track of its location. The table is rated to hold up to 6 kg when placed in the vertical position.

Figure 29: 8MTF-102LS05 motorized scanning stage.

The stage is controlled using an 8SMC1-USBhF Microstep Driver controller unit (also supplied by Standa) which communicates with a control computer using a usb interface. It is capable of connecting to three independent axis of movement. It is powered by a 36 V, 3.33 A power supply.

3.1.2   *Table control software*

Standa provides proprietary software, called SMCView, that can control a variety of positioners through the 8SMC1-USBhF controller. They also provide a set of virtual instruments for developing custom programs using LabView. All software is intended to be run on Windows 2000 or a newer version of Windows. These virtual instruments were used to create a custom LabView control program called SensorMeas.

Two versions of the software were developed, one for sensor testing at AGH and the other for use at test beams.

3.1.2.1 *Software design*

The philosophy behind the design of SensorMeas was to use event-driven programming that will respond to events generated by the user. An event-driven design avoids the situation in which the computer is constantly running a loop to check all state variables. This programming does not scale well with processor power and program complexity, and can lead to overheating or a delayed response. Instead, the user interface connects to a central loop that waits to execute until an event, generally a user interaction, is triggered. The design was based on software examples provided by Standa for operation of the 8MTF. The source is available online in the lumifun server.

3.1.2.2 *SensorMeas*

SensorMeas talks to the 8SMC1-USBhF controller using the Virtual Instrument Software Architecture (VISA) communications standard. It was designed based on the examples provided by Standa. It has two main loops, one for control and one for monitoring the table status. A screenshot of the program is shown in figure FIG. It is capable of controlling the 8MTF in two dimensions. A users' guide with diagrams of the user interface is included in Appendix B.

3.1.2.3 *SensorMeas3D*

For sensor testing with the laser source, it is necessary to move the sensor along the axis of the beam into the focal point of the laser. SensorMeas was accordingly adapted to control a second, one-dimensional positioner (the 8MT) in addition to the two-dimensional 8MTF.

Power cycling was important to reduce noise for LumiCal, but reduced the resolution of the motion to 10 um. This was nevertheless larger than the motion required for the LumiCal sensors. When used on the Pixel Detector by Imran, noise was not a problem since the small pixels (25µm x 25µm) do not create as big an antenna as the LumiCal sensor pads, so power cycling was not needed and resolution was the nominal positioner resolution of 2.5 µm.

Unlike at a test beam, where the large beam size makes it unnecessary to move the sensor frequently, the fine spot size of the laser at AGH requires that many data points at different positions on the detector be taken. It is therefore necessary to automate the movement of the detector and synchronize it with data acquisition device. This was done in two ways, depending on the device:

OSCILLOSCOPE   The Agilent MSO7034B oscilloscope is capable of using the VISA communication protocol. Using the VISA libraries for LabView, SensorMeas is written so that once motion to a new position has stopped, the oscilloscope will wait for a laser pulse, read data from the electronics, and notify SensorMeas when it is safe to proceed to the next position.

COMPUTER   VISA libraries for 64-bit Linux computers are not available, so SensorMeas was developed and run on a 32-bit Windows laptop. This presented a challenge when trying to synchronize measurements with a separate Linux computer that performed the data acquisition. In order to overcome this difficulty, a virtual machine running Windows was set up on the DAQ computer, with a shared folder in which both SensorMeas and the DAQ software look. When the motion is complete, SensorMeas writes a file to this folder. the DAQ software checks for the presence of this file, takes a measurement, and when the measurement completes, deletes the file. When SensorMeas sees the file has been deleted, it moves to the next position and starts over again.

3.2   TEST BEAM

3.2.1   *Test beam 1*

The first test beam was performed in July 2010 at DESY-Hamburg by members of the FCAL Collaboration from AGH UST, IFJ, and DESY-Zeuthen. The purpose of the tests was to examine sensor characteristics for Si (LumiCal) and GaAs (BeamCal), as well as testing some of the front-end electronics for LumiCal. The results for BeamCal will not be covered in this thesis. The results reported in this section can also be found in a published EUDET memo [21]. A description of the setup has been included, but parts of the analysis not performed by the author have been removed. The reader is referred to [21] for full results.

The test beam was performed at DESY-Hamburg using 4.5 Gev electrons from the DESY II accelerator. Photons are extracted from the accelerator and converted to e+/e- pairs. The electrons are separated using a magnetic field and passed through a collimator to define the beam. The electron energy can be tuned by varying the strength of the magnetic field.

In these measurements the readout chain under test included the silicon sensors, kapton fanout and front-end electronics. Figure 30 shows the structure of the LumiCal detector. The yellow color shows the sensor tile which was produced by the Hamamatsu Photonics and used in the test beam measurements.
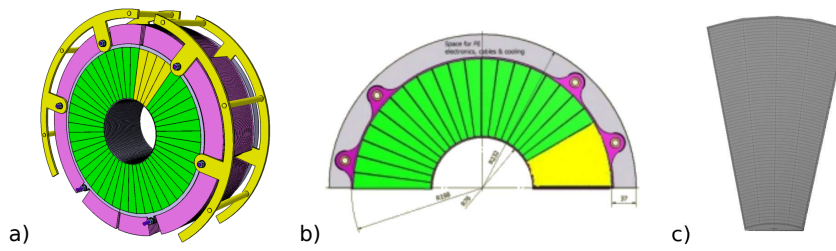
Figure 30: a) LumiCal calorimeter design b) Half plane, silicon sensor tail
marked on yellow c) Silicon sensor tail prototype used during test
beam measurements together with dedicated front-end electronics.

3.2.2 *Test beam setup*

Figure 31 describes the experimental setup used during test beam
measurements. The three ZEUS telescope planes [3] enabled the tra-
jectory of the electron beam to be precisely measured, allowing the
impact point on the sensor tile to be calculated. This was only true for
the case without tungesten absorbers in front of the sensors (n=0, in
figure 31
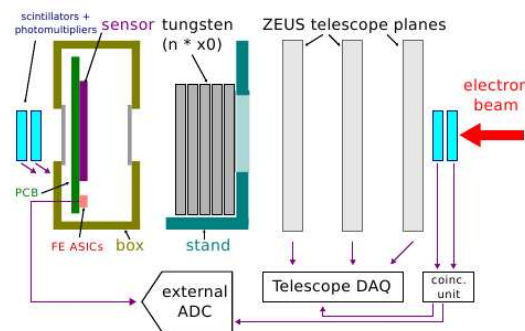
Figure 31: Experimental setup used during test beam measurements.

The PCB indicated in Figure 31 contains:

- one tile of silicon sensors comprising 256 pads (of which 16 were
  connected to readout chips),

- kapton fanout connecting the sensor and front-end electronics,

- front-end ASICs [26],

- power supply and biasing circuits,

- line drivers

Figure 32: Photograph of experimental setup.

This PCB was developed and produced at AGH, and mounted within the box. Analog signals were driven out of the box and sent to an external commercial sampling ADC (v1724, 14 bit, 100 Msps) provided by CAEN. A movable X-Y table (Section 3.1) was used to precisely position the DUT box with respect to the beam line. Three scintillators connected to photomultiplers working in coincidence provided triggering for both the ZEUS telescope and LumiCal data acquisition systems (DAQ). A veto scheme (through BUSY signal) was used to ensure that both DAQs acquire the same number of events. Event building was performed off-line. An additional stand was foreseen to allow adding up to several tungsten layers for studying the response of the readout chain to electromagnetic shower.

A variant of SensorMeas was used to control the X-Y table motion. Since the beam width of a test beam is not very precise, for test beams the detector is moved infrequently, and in large increments. Therefore it is not necessary to automate the motion of the positioner, or to synchronize it with data acquisition.

3.2.3   *Test beam results*

3.2.3.1   *Position reconstruction*

The ZEUS telescope and related reconstruction software were used to obtain information about hit positions in the sensor under test. In Figure 33 one can see the combined information from the LumiCal DAQ and the telescope DAQ. The coordinates of each point are given by the telescope data, but its color is assigned by the LumiCal DAQ - a different color for each sensor pad. As one can see the structure of LumiCal sensor, plotted in black, is reflected in the reconstructed data. Unforunately, due to large uncertainties in the position of the sensor during the test beam, we were unable to accurately examine

how generated charge changes as a function of position within a single pad.
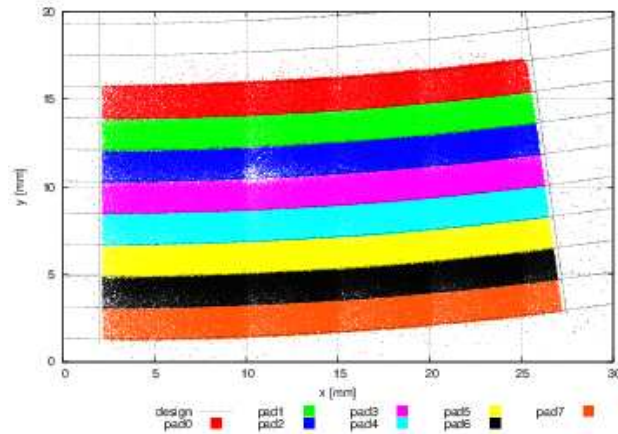
Figure 33: Reconstructed position of beam particle impact point combined with signals registered in LumiCal sensor pads.

3.2.3.2 *Response of sensors to electromagnetic shower*

During the test beam the response of readout chain to electromagnetic shower was studied. Figure 34 shows the histogram of energy deposition in the whole instrumented area (8 sensor pads) under the $2\text{-}X_0$ (left) and $4\text{-}X_0$ (right) tungsten absorber. The measurement results (red) were compared with prediction of GEANT4 Monte Carlo simulations (blue) where the experimental setup was implemented. Good agreement was found between them.
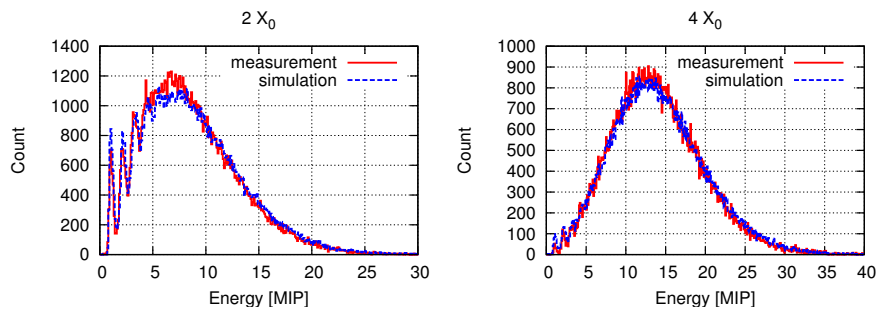
Figure 34: Energy deposition in instrumented area under the $2 X_0$ (left) $4 X_0$ (right)

The average charge deposited in instrumented area as a function of tungsten thickness is shown in Figure 35. The reasonable agreement with Monte Carlo was found.
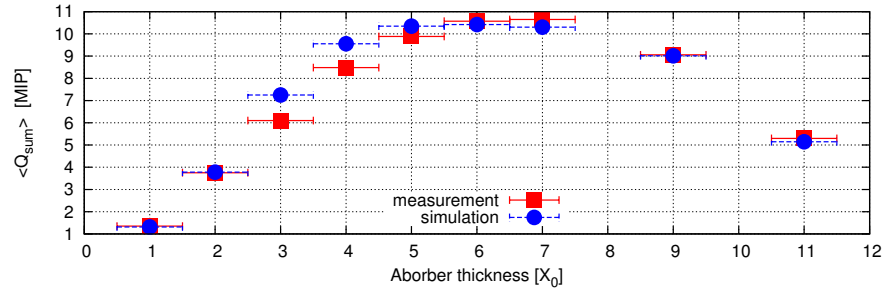
Figure 35: Average charge deposited in instrumented area as a function of tungsten thickness.

3.2.4 *Conclusions from the test beam*

The first tests performed at 4.5 GeV electron beam at DESY allowed verification of the response of partial readout chain to charged particles. The results indicate good performance of all components (silicon sensors, kapton fanout and front-end electronics), matching expected performance values. Test beam measurements also allowed studies of shower development using tungsten plates as absorbers, and will be useful in test beams going forward in which more layers of tungsten are used. The SensorMeas software was useful, but the test beam also indicated areas in which it could be improved, such as configurable safety features to prevent crashing depending on cabling, and an improved system of remembering pad coordinates.

APPENDIX

LUCAS SIMULATION USERS' GUIDE

A.1 RUNNING LCAL-PRO FROM THE COMMAND LINE

A.1.1 *Environment*

The working version of LumiCal is typically **Lcal-pro**. Normally it does not appear as an executable until you set the environment. To do this, log in as **fcal**, go to the home folder and run the **env.sh** script:

```
[fcal@nz12-25 fcal] . env.sh
```

A.1.2 *Command line arguments*

The syntax for running Lcal-pro is **Lcal-pro [options]**. **Lcal-pro -h** prints a list of command line arguments. The complete list is:

```
 -h                print this help message.
-b               batch mode
-i               interactive mode

-m <filename>    specifies a macro file to be executed before running.
                  default none

-o <filename>    specifies file name for ROOT output.No default
-M <mode>        specifies ROOT file opening mode ( default is CREATE
                  to avoid accidental file overwriting ).
                  Possible values are RECREATE/UPDATE
-A <Yes/No>      specifies whether events from entire Run are to be
                  written in one Tree ( suitable only for beam background data )
                  default is 'No'

-c <double>      specifies the Geant 4 production range cut in mm.
                  (default is 0.005 mm)
-x <double>      specifies the Beam Crossing Angle in mrad
                  (default is 0 [mrad])
-s <filename>    specifies name of the file with geometry setup
-P <int>         specifies printout level ( default is 0= minimum print
                                                3= debug printout
```

Note: -M currently has no default setting (see fcal/LumiCal-pro/src/Setup.cc)

Typically the most common options are **-b**, **-m**, and **-o**. **-o** should be followed by a filename ending in **.root**. You can write the Geant4 output to file by appending » **logfilename.log** to the end.

A.2   BATCH MODE, MACROS

The option **-b** tells Lcal-pro to execute a macro, as opposed to running in interactive mode. Specifiy the macro by following **-b** with **-m macroname.mac**:

```
[fcal@nz12-25 LumiCal-pro] Lcal-pro -b -m
  geant4macro.mac -o rootfile.root > logfile.log
```

Of course, the macros and output files can be stored in other directories, so if this is the case just prefix them with the path. Use "&>" if you want stderr sent to the log file in addition to stdout.

A.3   SETUP FILE OPTIONS

Lcal-pro receives geometric parameters (and a few others) for the LumiCal detector from an external file. A sample file is shown in Appendix ? This allows for the geometry to be changed without having to recompile the program. To change the geometry, change the file. It typically has the suffix **.ini**. It's probably a good idea to have the file name reflect the changes in geometry.

A.4   PAUSING AND RESTARTING A RUN

A.4.1   *Pausing*

You can pause a run by placing a file in the current working directory called "aStopRun". The file's contents don't matter. Lcal-pro periodically checks the current directory for this file, and quits cleanly if it finds it. My favorite way to do this is with **touch**:

> **touch** aStopRun

Then, check the current processes until this instance of Lcal-pro has stopped running. Warning: all Lcal instances that were started in this folder will be able to see **aStopRun**, so the one that reads and deletes it first is the one that will stop. For this reason, I typically have a dedicated folder for each instance.

A.4.2   *Restarting*

To restart a run, use the same command and options you used to start it in the first place, with the following changes:

- Make sure the Setup file has RootFileMode set to UPDATE!

- Use » to append the output to the same log file you used before.

- *IMPORTANT:* The run will start over *from the beginning* so if you don't want to repeat events you must change the macro accordingly! For example, if you have a 20-event run and you stop after 6, you should change the macro to specify 14 runs when you restart, otherwise you will end up with 26 events in your root file.

Use the same filenames and macro as before.

A.4.3 *Managing several instances*

My current computer has two processors; therefore, I like to leave two instances of LumiCal running over the weekend to use the computer more efficiently. It is helpful to be able to stop and start them independently of each other when you want to check on their progress, but if you launch them from the same folder, whichever one sees **aStopRun** first will pause and the other will keep running. Instead, launch each instance from a separate folder, e.g. **LumiCal-pro/job1** and **LumiCal-pro/job2**. Then, you can choose which instance to pause by creating an **aStopRun** file inside its corresponding folder.

A.5 ROOT FILE OUTPUT

Lcal-pro is currently designed to write a ROOT tree once the Geant4 run is finished. It creates the file when the run starts, so you can check quickly to see if the file has been properly created. The tree has the following branches:

- numPrim - the number of primary particles

- pX[1], pY[1], and pZ[1] - momentum components of the primary(ies)

- primID[1] - primary particle ID number

- primPDG[1] - the type of primary (PDG code)

- numHits - the number of recorded hits (1 per unique cell)

- cellID[numHits] - the IDs of the cells which recorded hits

- rCell[numHits], phiCell[], zCell[] - cells' cylindrical coordinates

- xHit[numHits], yHit[], zHit[] - hits' cartesian coordinates

- eHit[numHits] - energy deposited into each cell

- Etot1 - total energy deposited in the run, in module 1

- Etot2 - total energy deposited in the run, in module 2

- numTracks - the number of primary tracks

- trackPID[1] - the ID of the track

A.6 TROUBLESHOOTING

A.6.1 *Make errors*

1. I updated the version and now make quits with a "touch: cannot touch" error.
   - Make sure the executable in the makefile, the directory, and the main (foo.cc) all have the same name.

A.6.2 *User problems*

1. Lcal-pro isn't writing ROOT files.
   - Try running Lcal-pro in the foreground. Is it pausing after the run finishes? You might be running in interactive mode. Check that your setup file has **batchMode** = 1.

2. A run quits with a message similar to

   ```
   symbol lookup
   error:  Lcal-pro: undefined symbol: _ZN6TClass8GetClassERKSt9type_infob
   ```

   - Lcal can't find the appropriate ROOT file. This happened when I updated my version of ROOT without recompiling Lcal-pro. If you update ROOT, make sure you compile Lcal-pro again so it knows where all the files are.

A.6.3 *Command errors*

1. Lcal-pro isn't behaving according to the commands I gave from the terminal.
   - Setup files override command line options. If you are using the -s option with a setup file, check that the options in the file match with the ones you want to give from the terminal prompt.

A.7 SAMPLE LUCAS STEERING FILE

```
#
#---------------------------------------------------------------
# This file contains list of all geometry setup
```

```
# parameters with its default values
# Only first two "black" fields in a line are
# used by Setup program.
# lines starting with "#" are treated as a comment
# any black field after first two is also treated as a comments
# parameter units must be: length -> [mm], angle ->[rad], field->[tesla]
# parameter name/value  must not exceed 80 characters.
#-----------------------------------------------------------------
#  globals
#---------------------------------
#batchMode              1    bhabha-00.mac
batchMode               1        single-ele-scan.mac
PrintLevel              0
EventStartNumber        0
MaxStepsNumber          20
StepSizeMin             0.
LogFreq                 100                    // 0 = no EndOfEvent logging
PhysicsListName         QGSP_BERT
RootFileName            single-ele-scan-smeared-rot3.root
RootFileMode            UPDATE
AccumulateEvents No
rangeCut         0.005
Beam_Crossing_Angle     0.
Nominal_field_value     3.5
Build_Beampipe          Yes
Build_LHcal             No
Build_BCal              No
Build_Mask              No
#--------------------------------
#  Cuts per regionSetup-RD.ini
LCal_Region_Cut          0.005
BCal_Region_Cut          1.000
LHcal_Region_Cut         1.000
Mask_Region_Cut          1.000


#---------------------------------
#  for World
#  world box dimensions [mm]
#---------------------------------
world_hdx              600.
world_hdy              600.
world_hdz             5000.
#------------------------------------
#  for beam pipe
#------------------------------------
Beam_pipe_thickness          1.0
```

```
Lcal_to_BeamPipe_clearance    1.0
#----------------------------------
# visual attributes
#
Beam_pipe_VisSolid            1                // <0> implies wireframe
LHcal_VisSolid                0
BCal_VisSolid                 0
Mask_VisSolid                 0                // <0> implies wireframe
Lcal_VisSensSolid             0                // <0> implies wireframe
Lcal_VisAbsSolid              0                // <0> implies wireframe
#----------------------------------
# for LCAL
#----------------------------------
Lcal_layer_fan            0
Lcal_virtual_cells        1
Lcal_n_layers             30
Lcal_n_tiles              12
Lcal_n_sectors            48
Lcal_n_rings              64
#----
Lcal_z_end            2635.0
Lcal_inner_radius       76.
Lcal_outer_radius      197.2                // [224.5] absorber outer radius
Lcal_SensRadMin         80.
Lcal_SensRadMax        195.2
#---
Lcal_space_for_ears      25.5
Lcal_sector_dead_gap      1.2
Lcal_layers_phi_offset    3.75              // 15 deg [3.75 deg]
Lcal_Phi_Offset           0.                // 15 deg [3.75 deg]
Lcal_layer_gap            0.1
Lcal_silicon_thickness    0.32
Lcal_pad_metal_thickness  0.02
Lcal_tungsten_thickness   3.5
#---
Lcal_ChipCaveDepth        2.6
Lcal_FEChip_rmax        250.0                // [224.0]
Lcal_FEChip_space         .5                 // chip thickness
Lcal_PCB_thickness        2.0
#---
Lcal_use_absorber         1
Lcal_use_fanout           1
Lcal_use_FE               1
#end
```

SENSORMEAS USERS' GUIDE

You have at your fingertips the single most powerful software designed by the Nuclear Elecronics and Detectors group at AGH to control the Standa 8MTF-102LS05 motorized x-y scanning stage with the 8SMC1-USBhF Microstep Driver that has ever been made. The main control panel is shown in Figure 36. Remember: with great power comes great responsibility. Using this handy guide and a working brain, you should hopefully be able to avoid ruining thousands of euros of equipment and hundreds of hours of effort. Good luck, and Godspeed.
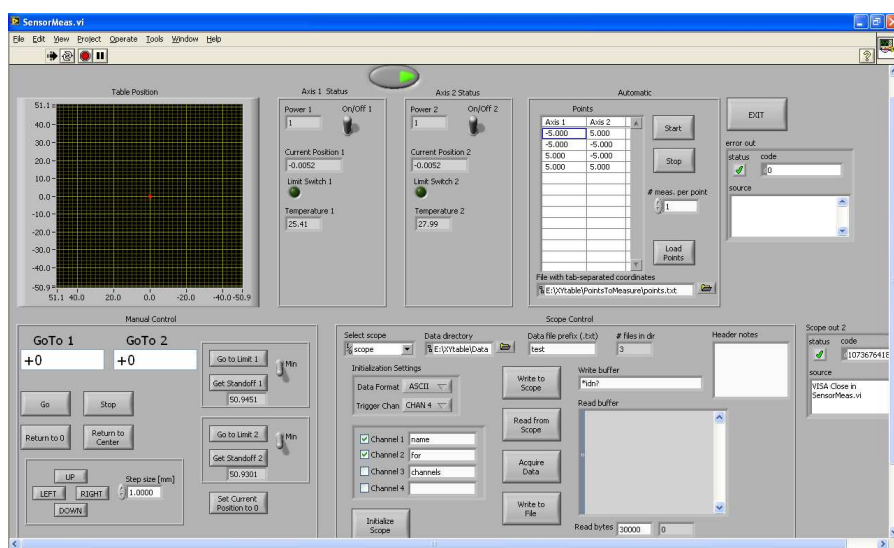
Figure 36: Most of the SensorMeas front panel interface. Note: no guarantee that any of the pictures of the interface are up to date. Consider them to be a general guide.

B.1 DEFAULT FILE LOCATIONS

B.2 X-Y TABLE CONTROL

The most prominent feature of the user interface is the large graph that tracks the position of the center of the table. The center and scales can be adjusted as described below. The plot is oriented so that the directions are the same as if the user were facing the table. The sign of the axes is also correct. The graph (and the code) makes more sense if the "x" axis (parallel to the long side of the faraday cage) is connected to the Axis 1 port on the controller, and the "y" axis (parallel to the

Table 8: X-Y table default folder paths

Use	Path
Parent folder	E:\XYTable\
Data storage	E:\XYTable\Data\
Measurement points	E:\XYTable\PointsToMeasure\
Standa VIs	E:\XYTable\Development Kit\LVdevKit_821\uSMC\
Sensor Meas VIs	E:\XYTable\lumisenlv\trunk\SensorMeas\
Table interface VIs	E:\XYTable\lumisenlv\trunk\TableFiles\
Scope interface VIs	E:\XYTable\lumisenlv\trunk\ScopeFiles\

short side of the faraday cage) is connected to Axis 2, but in principle these axes can be switched withouth problems.

Note: If you have the table set up the way you like it and the sensors and laser are all in place, this is the time to check the **Laser in box?** button. This button disables the controls that tell the table to move all the way to the end of its range of motion, so you're less likely to bump the sensor box against the laser. It also disables the **Set Current Position to 0** control, so you don't accidentally mess up the coordinate system.

b.2.1   *Table status*

The table status section (Figure 37) is a bit primitive but allows control of the power of both axes together (**Power**), or each axis independently (**Power 1(2)**). This section also displays the current position and temperature, which are updated every 50 ms. The **Limit Switch 1(2)** lights are used to indicate if the limit switch for axis 1(2) has been flipped, meaning that the axis is at the limit of its range of motion. This is useful for moving the table to a reference position.

If the temperature gets too high (> 70 C, I think) the motors will automatically shut off. I am not sure what will happen in this situation if the table is positioned vertically (like in a beam test) and has heavy tungsten plates attached.

The motor speed and step size are also reported, but these are of dubious value (so far).

b.2.2   *Automated control*

Most of the time, the data will be taken in automatic mode. The principal thing to remember about automatic mode is that it will automatically take and store measurements. The interface is shown in Figure 38.
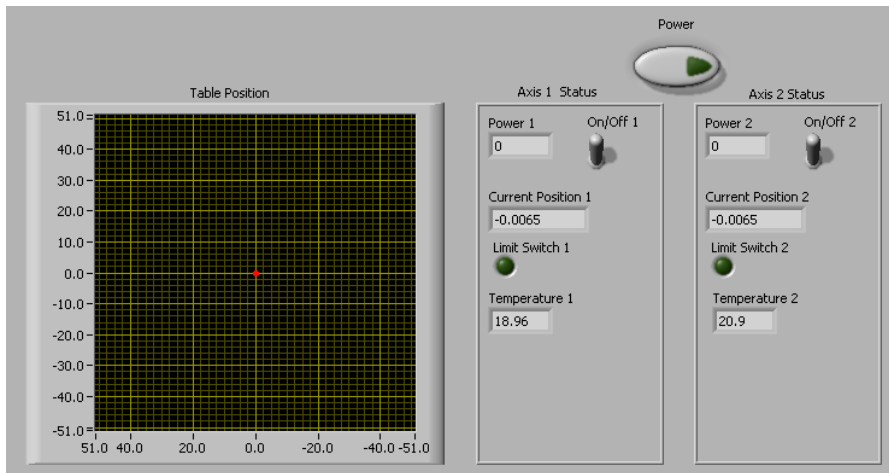
Figure 37: Motor status monitoring and control
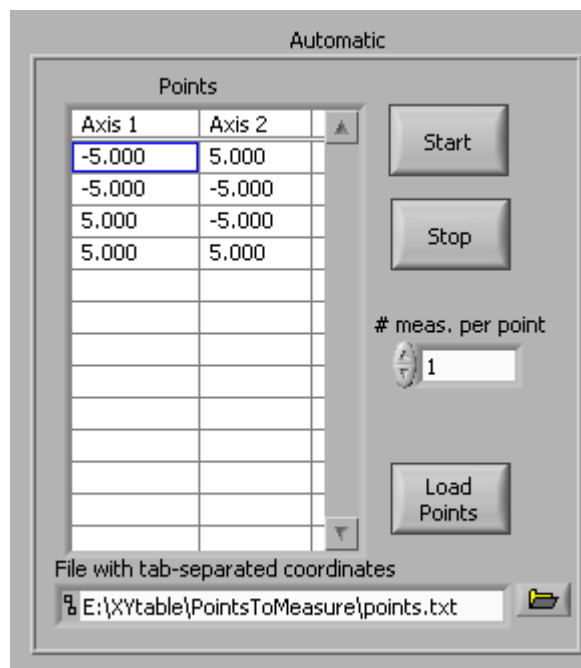
Figure 38: Automated run control

B.2.2.1 *Reading points from file*

- Create a file with TAB-SEPARATED coordinates (Axis1 <tab> Axis2), one point per line.

- Place the file in the directory E:\XYTable \PointsToMeasure, for simplicity.

- Tell SensorMeas where this file is in the **Automatic** panel, in the box labeled "File with tab-separated coordinates".

- Press **Load points**.

B.2.2.2 *Generating points with Start, Stop, and Step*

- Simplest case - move along one axis, hold the other constant (see "Beware" section below)

- Set starting point, stopping point, and step size for the axis you want to move along (**Start 1(2), Stop 1(2), Step 1(2)**).

- For the constant axis, set the Start and Stop values and **make the step size 0**.

- Yes, you really have to set all six values. Go back and set the step size for the constant axis to 0.

- Press **Generate points**.

Beware:

- If the **Stop** value is not an integer number of increments away from the **Start** value, the last increment that takes you beyond the **Stop** value will be entered as a point (i.e. the table will go too far).

- When neither axis is constant, both axes will increment until the axes with the most steps terminates. In this case it's better to read points from a file; you're more likely to get the behavior you want.

B.2.2.3 *Start the run*

- Set the number of measurements per coordinate with the **# meas. per point** control.

- Set the oscilloscope up (see Section B.3).

- Press **Start**. Unfortunately at this time it not possible to stop a run before it is complete (listed in "Bugs", Section B.8).

During an automated run, the power to the motors is turned off before a measurement is taken, and turned on again once the measurement is complete. This is because the noise in the data significantly increases when the power is on. This could potentially cause problems during beam tests when the table is in a vertical position, especially if there are heavy tungsten plates attached to it. We should make sure that it doesn't slip.

B.2.3 *Manual control*

The manual control interface is show in figure 39.

There are a few things you can do to manually control the position of the table:
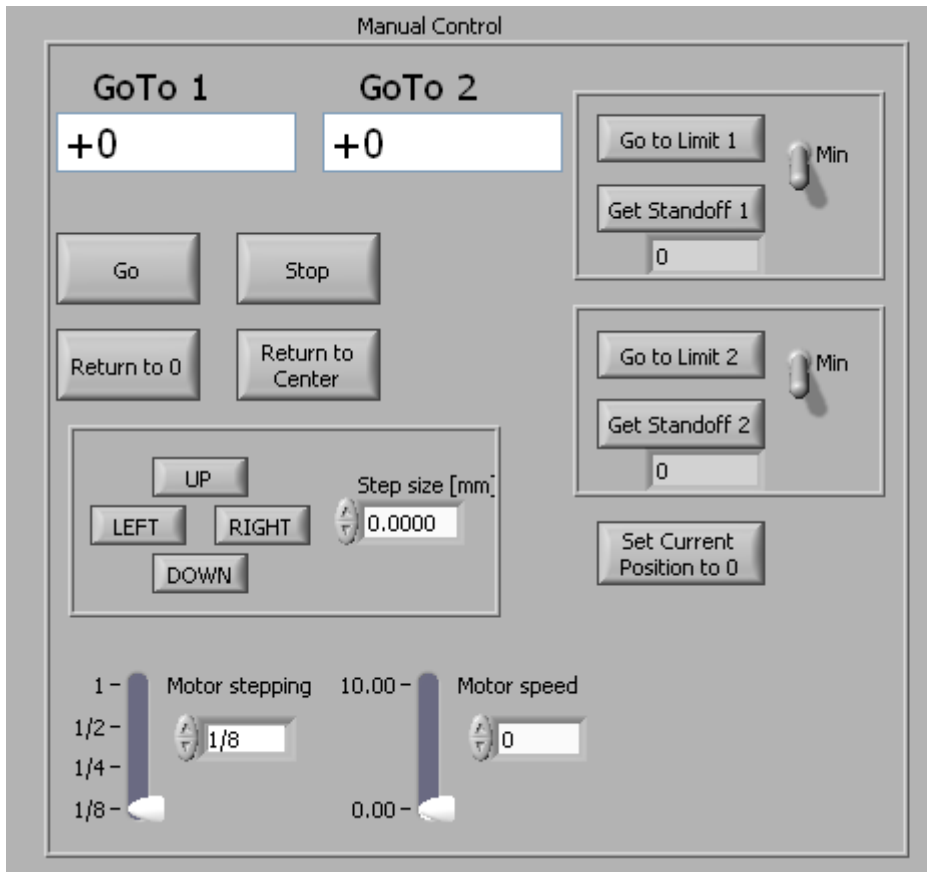
Figure 39: Manual control interface

- **Power/Power 1(2)**: Turn power on and off, either for both axes or for each individually.

- **Goto 1(2)**: Enter desired position in mm. 1→x-axis, 2→y-axis (corresponds to 8SMC1 controller inputs). Click "Go".

- **Return to 0**: Move the table back to the 0 position.

- **Return to Center**: Move the table to the center of its range of motion.

- Step table: **Up**, **Down**, **Left**, or **Right** according to the step size specified in the box on the right.

- **Go to Limit 1(2)**: Move along axis 1 or 2 until you hit the end (limit switch turns on). Direction is determined by the switch on the right (min→negative, max→positive).

- **Get Standoff 1(2)**: Measure the distance from the current position to the end (direction determined by min/max switch on right). Note: yes, the table actually has to move to do this so make sure it has enough room.

- **Set Current Position to 0**: zero the coordinate system to your current position. This will also rescale the table position graph axes. MAKE SURE TO GET THE STANDOFFS FIRST so that the position graph axes will rescale correctly.

- Motor step size and speed: Set how fast the motor moves and with what speed. Helps you control how accurately you move. Probably best to leave at default.

B.3   OSCILLOSCOPE CONTROL

This assumes that you're using an Agilent MSO7034B Mixed Signal Oscilloscope or similar (meaning that the interface is exactly the same). The first thing is to find the scope using the Measurement and Automation explorer. Go to Network Devices, delete the old scope entry since it probably has the wrong IP, and click **Add Network Device→VISA TCP/IP Resource**. Autodetect or Manual, check IP address against what the scope says its IP is (**Utilities→I/O** on the scope). Give it the alias "scope" (this just makes it easier for SensorMeas). In the Scope interface section of SensorMeas, select "scope" in the box titled **Select scope**. Yes, very confusing. If you didn't give the scope the alias "scope", you can either panic, OR you can select the alias you *did* give it. Your choice.
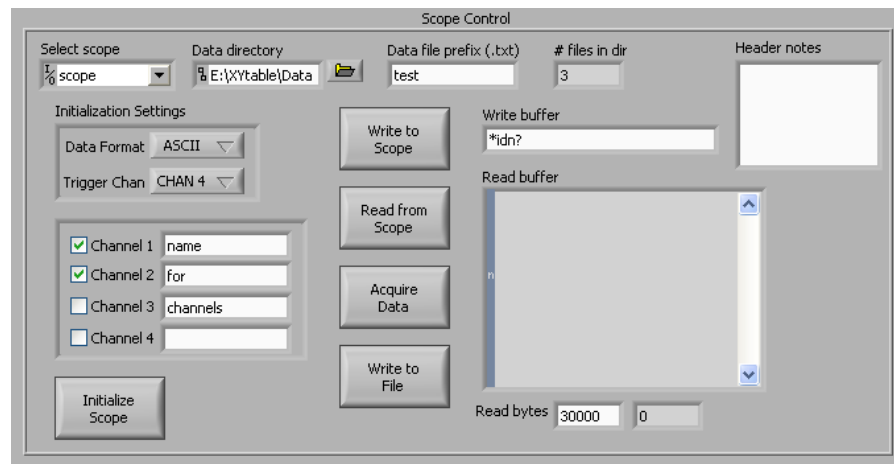
Figure 40: Scope control interface

B.3.1   *Scope settings/Initialization*

This may be expanded in the future.

- **Data format**: "BYTE" is very compact, but "ASCII" is easy to p. Depending on the number of points the scope is taking per measurement, make sure the **Read bytes** box is large enough to get all the data. This tells the scope how much data to read for

each measurement. 20000 should do; for ASCII format you get around 13k bytes of data per channel.

- **Trigger**: Wherever the trigger is plugged in. Channels 1-4 or EXT.

- **Channels**: Select which channels you want to read data from, and give them a title if you like.

B.3.2 *Data file information*

B.3.2.1 *File names*

Specify the location where the data files will be stored in the box labeled **Data directory**. Specify the prefix you want in the box labeled **Data file prefix**, for example "sensor_pads_8-10". The file name will be < prefix >< #### >.txt where < #### > is a 4-digit number that is based on how many files are in the current data directory (it increments the number). This number will be 1 + (value of **# files in dir**).

B.3.2.2 *Header and data format*

Each data file starts with a header. The header has the following information:

```
date time (to 0.01s)
Position: x, y [mm](to 1 um)
Measurement #: <#> (measurement number for this position)
Any notes

x-interval

Channel #i: channel title
y-scale [V]
<data>

Channel #i+1: channel title
y-interval
<data>

...
```

Brief explanation:

- date: retrieved from system

- time: retrieved from system

- actual position: current value of **Current Position 1(2)**

- user-set position: current value of **Goto1(2)**

- Measurement #: ranges from 1 to value of **# meas. per point** (see B.2.2 OR value -1 indicates that the measurement was taken manually.

- Notes: Whatever is currently in the **Header notes** box.

- Channel #: channel title: Channel 1, 2, 3, or 4 (however many are checked) plus the value of the text box next to the check box.

- x-interval: the x scale of the scope window (output of scope command :wav:xinc?).

- y-interval: the y-scale for the channel being measured (output of scope command :chan:scale?). Units of volts.

- $< data >$: a 10-character word followed by the channel data.

All lines in the data end in an ASCII "Carriage return" character.

Example data file:

```
21/04/2011 13:15:12.76
Pos_real: -0.9965, 0.0035
Pos_user: -1.0000, 0.0000
Measurement #: 12
This is a sample data file.

dx: +2.00000000E-009

Channel 1: some data
dy: +5E-03
#800012999 1.25000e-03, 1.25000e-03, 1.25000e-
03,...

Channel 2: some more data
dy: +5E-03
#800012999 3.64999e-03, 9.90000e-03, 9.90000e-
03,...
```

B.3.3   *Interactive scope control*

In the scope interface section of SensorMeas there are two buttons that allow interactive control of the scope. The **Write to Scope** button sends whatever is in the **Write buffer** to the scope (e.g. commands like *LRN? and :trig:ext). If the command sent to the scope is a request for a response (i.e. it ends in a "?"), press the **Read from Scope** to get

the response. It will appear in the **Read buffer**. Make sure that **Read bytes** is not set to 0.

B.3.4 *Interactive data taking*

If you want to interactively take data, move the table to the position you want and press **Acquire data**. This oes *not* turn the table motors off - you must turn them off *by hand* if you want a low-noise measurement. The data will appear in **Read buffer**. This will *not* write the data to file. Set the file prefix you want (as before) and press **Write Data to File**. Everything in the **Read buffer** text box will be written to the file with the prefix specified in **prefix** and a number assigned as described previously.

B.3.5 *Laser interface*

The SensorMeas program has no direct access to the laser. Therefore, you must set the laser trigger output frequency by hand on the laser control box. Wire the **Sync out** port from the laser control box to your favorite triggering input on the scope. You can select this input for triggering either on the scope itself or on the scope interface part of SensorMeas (but make sure to click **Initialize Scope** before taking data).

B.4 RECOMMENDATIONS

B.4.1 *Reference coordinate system*

Set (0,0) to be the lower right-hand corner of the X-Y table range (minimum values for axes 1 and 2). Click **Go to Limit 1** and **Go to Limit 2**, and make sure the **Limit Switch** lights turn on. Then click **Get Standoff 1(2)** (toggle the switch to "Min" first) and finally click **Set Current Position to 0**.

B.4.2 *Other scope settings*

There are still many scope settings that cannot be set by SensorMeas, for example the x- and y-scales. Also, SensorMeas does not have an easy way to reset the scope to some known initial state with the exception of interactively sending (see Section B.3.3)the *RST command. So, anything that is not on the SensorMeas front panel should be set on the scope itself or by using the **Write to Scope** functionality of SensorMeas.

B.5   SHUTTING DOWN

Pressing **Exit** will close the VISA session to the motors after shutting off their power. Perhaps this sentence was not worth its own section of the users' guide.

B.6   TROUBLESHOOTING

This section will be expanded as more problems are discovered.

B.6.1   *No response to "Read from Scope"/Read buffer never fills*

You probably have **Read bytes** (below the **Read buffer**) set to 0. Set it not equal to 0. If you don't get the whole response you expect, keep increasing it until you do.

B.6.2   *"Insufficient location information..."*

> Error - 1073807343 occured at VISA Find Resource in Find Devices (uSMC)
> Possible reason(s): VISA: (Hex 0xBFFF0011) Insufficient location information or device is not present in the system.

LabView can't find the 8SMC1 controller. Make sure the USB cable is plugged in (both ends!) and the power cable is in. Does the Measurement and Automation Explorer report the 8SMC1 axes under "Devices and Interfaces"? As a last resort, plugging in all the connections and rebooting is a drastic, but typically successful, solution.

B.6.3   *Can't find X-Y table motors*

In the Measurement and Automation Explorer, make sure that the X-Y table motors show up. Try removing the USB cable to the 8SMC1-USB controller and plugging it back in. Watch the "Devices and Interfaces" menu - they should pop up there.

B.7   EDITING IT YOURSELF

B.7.1   *Design philosophy*

SensorMeas is designed using event-driven programming - that is, the actions taken by the program are generally triggered by user events generated by the user interacting with the SensorMeas front panel. Therefore, on the left hand side of the block diagram you will see a

whole bunch of controls which aren't wired to anything. Instead, either local variables or property nodes are used to get or set their values as needed in the program. For example, local variables or property nodes for "Goto 1" and "Goto 2" can be found in the "Start Run" case, the "Start Manual" case, and the "Step" cases.

B.7.2 *Parallel loops*

Three loops run in parallel: the actions loop and the table monitoring loop, and the scope control loop. They have 50 ms delays built in to avoid eating up too much processor time. This allows the state of the X-Y table to be monitored, and the scope to be accessed, while other things are happening.

B.8 IMPROVEMENTS/TODO

This section lists bugs that need to be fixed or ideas that should be implemented.

Ideas

- Do an automatic run without taking and storing data.

- **Acquire Data** subvi interface should be more elegant and recyclable

- During an automatic run, select a data set at random and plot it to check that things are working correctly.

- Safer data file numbering system - currently, you might generate files with the same name if you change data directories.

- Initialize oscilloscope to some known state.

- Instead of dy for each channel, use a better index of the oscilloscope y-scale

- Clear oscilloscope buffers before reading

Bugs/Fixes

- Stop an automatic run during the run

- Elegantly handle the case where Init Devices cannot find the X-Y table axes

SIMULATION ANALYSIS CODE

This appendix contains selected functions from the code used to analyze the simulation data. It is written in Python, using ROOT bindings.

C.1 MODULES AND CLASSES IMPORTED

```python
import sys
import numpy as np

import ROOT as rt
from ROOT import TChain
from ROOT import gDirectory

rt.gROOT.ProcessLine(".L include/loadDict.C+")
```

C.2 GLOBAL CONSTANTS

```python
E2Q = 44.4 # [pC/GeV] - convert between e_vis and charge
    generated

# list of energies, useful for looking up energy indices
# string and int format
ENERGIES = ['5', '25', '50', '100', '150', '200', '250', '500']#,
    '1500']
ENE = np.array([5, 25, 50, 100, 150, 200, 250, 500],float)

# bin limits, not used
lowerlims = np.array([ 0., 0.1, 0.45, 0.9, 1.5, 2. , 2., 4.,
    15.])
upperlims = np.array([0.1, 0.4, 0.65, 1.3, 1.9, 2.4, 3., 6.,
    20.])


# Shifts in phi due to b-field as a function of energy [rad]
bfieldshifts = {'5':-0.2673, '25':-0.05364, '50':-0.02669, '100'
    :-0.01338, \
                '150':-0.00882, '200':-0.006697, '250':-0.005422,
                     \
                '500':-0.002684, '1500':-0.0008005} # [rad]

# RMS of phi reconstruction resolution as a function of energy [
    rad]
```

```python
phirms = {'5':0.01463, '25':0.00786, '50':0.006145, '100'
    :0.005103, \
                '150':0.004586, '200':0.004326, '250':0.004136, \
                '500':0.00363, '1500':0.003061} # [rad]

# Tile gap centers (rad)
gapcenters = [-3.14159265359, -2.61799387799, -2.09439510239,
    -1.57079632679, \
            -1.0471975512, -0.523598775598,
            0.0, \
            0.523598775598, 1.0471975512, 1.57079632679,
                2.09439510239, \
            2.61799387799, 3.14159265359]

Lcal_z = 2501.95 # [mm] front of LumiCal, measured from IP
Lcal_rin  = 80.0 # [mm] inner radius
Lcal_rout = 195.2 # [mm] outer radius
```

C.3 POSITION CALCULATION

```python
def BShiftFunc(energy):
  ''' calculate B-field shift for any
  generated energy parameters set by fitting a graph of
      bfieldshifts
  vs gen energy ''' shift = -(1.33662/energy) + -4.69815e-06
      return
  shift

def GetMomentum(chain):
    """Checks for file format and returns momentum"""
  px, py, pz = 0, 0, 0
  try:
      px, py, pz = chain.Tracks[0].pX, chain.Tracks[0].pY, chain
          .Tracks[0].pZ
  except AttributeError:
      try:
          px, py, pz = chain.pX[0], chain.pY[0], chain.pZ[0]
      except AttributeError:
          print "Error: momentum not found"
          sys.exit()
  return px, py, pz

def GenPos(chain, jentry, ROT=False):
    '''
    Generated position
    Return order: radius [mm], theta [rad], phi [rad]
    '''
    #chain.LoadTree(jentry)
    #chain.GetEntry(jentry)
    px, py, pz = GetMomentum(chain)
```

```
    # phi [rad]
    phi = np.arctan2(py, px)
    # Shift due to magnetic field
    energy=str(int(np.sqrt(px**2+py**2+pz**2)))/1000)
    phi = phi - bfieldshifts[energy] # this can push phi to
        illegal values, so correct
    if phi >  np.pi: phi = phi - 2*np.pi
    if phi < -np.pi: phi = phi + 2*np.pi

    # theta [mrad]
    theta = np.arctan(np.sqrt(px**2 + py**2)/(pz))

    # radius [mm]
    rad = Lcal_z*np.tan(theta)

    return rad, theta, phi

def HitGenPos(chain, ROT=False):
    """
    Generated position, determined by the hit with the lowest z-
        value
    Order: radius [mm], theta [rad], phi[rad]
    """
    try:
        assert(chain.numHits > 0)
        zhits = np.array([h.zHit for h in chain.Hits])
        xhits = np.array([h.xHit for h in chain.Hits])
        yhits = np.array([h.yHit for h in chain.Hits])
    except AssertionError:
        print "Error in HitGenPos: no hits"
    # reorder hits
    #zorder = zhits.argsort()
    #xhits = xhits[zorder]
    #yhits = yhits[zorder]
    #zhits = zhits[zorder]

    rad = np.sqrt(xhits[0]**2+yhits[0]**2)
    theta = np.arctan(np.sqrt(xhits[0]**2 + yhits[0]**2)/(zhits
        [0]))
    phi = np.arctan2(yhits[0], xhits[0])

    return rad, theta, phi

def RecPos(chain, jentry, ROT=False):
    '''
    Reconstruct position
    Return order: radius, theta, phi
    '''

    # dictionary of phi weighting constants
    if ROT == True:
        phiweights = {'5':7, '25':8, '50':8, \
```

```
                          '100':8, '150':8, '200':8,\
                          '250':8, '500':9, '1500':9} # lookup in [
                              GeV]
      elif ROT == False:
          phiweights = {'5':14, '25':14, '50':13, \
                          '100':11, '150':11, '200':11,\
                          '250':11, '500':10, '1500':12} # lookup in
                              [GeV]
      thetaweights = {'5':5, '25':7, '50':8, \
                          '100':8, '150':10, '200':9,\
                          '250':9, '500':9, '1500':10}# nogap LumiCal
                              lookup in [GeV]


      chain.LoadTree(jentry)
      chain.GetEntry(jentry)
      # Get energy
      px, py, pz = GetMomentum(chain)
      energy = str(int(np.sqrt(px**2+py**2+pz**2)/1000))
      # Get weight const
      phiwconst = phiweights[energy]
      thwconst = thetaweights[energy] # weighting constant

      phisumwt, thsumwt = 0., 0.
      phiwt, thwt = 0., 0
      xsum  = 0.
      ysum  = 0.
      thetasum = 0.
      Etot = max(chain.Etot[0], chain.Etot[1])

      for i in range(0, chain.numHits):
          # phi
          phiwt = max(0., phiwconst + np.log(chain.Hits[i].eHit/
              Etot))
          phisumwt = phisumwt + phiwt
          xsum = xsum + chain.Hits[i].xCell*phiwt
          ysum = ysum + chain.Hits[i].yCell*phiwt
          # theta
          thwt = max(0., thwconst + np.log10(chain.Hits[0].eHit/
              Etot))
          thsumwt = thsumwt + thwt
          thetasum = thetasum + thwt*np.arctan2(np.sqrt(chain.Hits[
              i].xCell**2  \
                                                 +chain.Hits[
                                                     i].yCell
                                                     **2),\
                                            chain.Hits[i].zCell
                                                )
      xsum = xsum/phisumwt
      ysum = ysum/phisumwt
      phi = np.arctan2(ysum, xsum)
```

```
    if thsumwt <= 0:
        print jentry, "sumwt = 0"
    theta = thetasum/thsumwt

    rad = Lcal_z*np.tan(theta)

    return rad, theta, phi
```

C.4 GAP CALCULATIONS

```python
def CheckGap(phi, rad, hgap = 1.25, ROT = False):
    '''
    Set for "Strict" gap selection
    All calculations done in _radians_, input must be in radians
    rotation is the rotation between layers, default 3.75 degrees
        .
    Uses the radial impact point to determine the angular gap
    width.
    Return FALSE if particle in gap, TRUE if in sensor!
    '''
    # Geometric constants
    numsectors = 12
    dsec = 30*np.pi/180.  # tile width, radians
    if ROT:
        rotation = 3.75 # degrees
    elif not ROT:
        rotation = 0
    offset = rotation*np.pi/180 # convert to rad
    # hgap -> Half-gap width - play with it [mm]

    # Get delta phi based on radius
    dphi = np.arctan2(hgap, rad)

    for i in range(-1*numsectors/2, numsectors/2+1):
        if phi > (i*dsec - dphi) and phi < (i*dsec + dphi):
            return False;
        if phi > (i*dsec - dphi - offset) and phi < (i*dsec +
            dphi - offset):
            return False;
    # if you make it through the loop, phi must be ok!
    return True;

def GetClosestGap(hit_phi, ROT=False):
    '''
    Takes in the phi coordinate of a hit, and finds the
    phi-coordinate of the closest gap.
    Input and output are in radians, but calculation is in
        degrees.
    '''
    hit_phi = hit_phi*180/np.pi
    dphi = []
```

```python
    numsectors = 12
    dsec = 30 # tile width
    offset = 3.75 # rotation between layers

    # gap0/1 - gap phi values without/with rotation
    gap0 = [i*dsec for i in range(-1*numsectors/2, numsectors
        /2+1)]
    gap1 = [i*dsec+offset for i in range(-1*numsectors/2,
        numsectors/2)]

    # find closest element in gap0 or gap1
    for g in range(len(gap0)):
        dphi.append(abs(hit_phi - gap0[g]))
        #closest_gap = gap0[dphi.index(np.array(dphi).min())]
        #return closest_gap

    if ROT: # in case of rotation, also check gap1
        for g in range(len(gap1)):
            dphi.append(abs(hit_phi - gap1[g]))

    # no rotation: indices 0-12
    # rotation: indices 13-25
    gap = dphi.index(min(dphi)) # index of minimum value of dphi
    if gap < 13:
        closest = gap0[gap]
    elif gap >= 13:
        closest = gap1[gap-13]

    closest = closest*np.pi/180
    return closest

def GetDist2Gap(hit_rad, hit_phi, ROT=False):
    '''
    Calculate the straight-line distance to the nearest gap.
    '''
    tilephi = (30+3.75)*np.pi/180
    try:
        assert (-np.pi <= hit_phi <= np.pi)
    except AssertionError:
        print "Bad phi angle, %f"%(hit_phi)
    gap_phi = GetClosestGap(hit_phi, ROT)
    #dphi = abs(hit_phi) - abs(gap_phi)
    dphi = hit_phi - gap_phi
    try:
        assert (abs(dphi) < tilephi)
    except AssertionError:
        print "Bad dphi! %4f, %4f, %4f"%(dphi, hit_phi, gap_phi)

    # phi is discontinuous at -180/180
    #if abs(dphi) >= np.pi:
    #    if dphi < 0:
    #        dphi = dphi + 2*np.pi
```

```python
#    elif dphi > 0:
#        dphi = dphi - 2*np.pi


    dist = hit_rad*np.sin(dphi)
    return dist
    #return dphi*180/np.pi
```

C.5 SCALING FUNCTIONS AND CORRECTION FACTORS

```python
def CorrFunc(evis, ROT=False):
    '''
    Calculate original energy (Egen) using parameters from a fit
        to a
    plot of Evis vs Egen
    Parameters calculated by LucAna.py (taken from the
        evisegenhist)
    '''
    if ROT == False:
        m =  0.011268
        b = -0.000996
    elif ROT == True:
        m =  0.011286
        b = -0.00225

    erec = (evis - b)/m

    return erec

def GetCorrFac(egen, evis):
    """Returns MLV of the distribution of egen or evis"""
    # first, check input type
    try:
        assert type(egen) == np.ndarray
        assert type(evis) == np.ndarray
    except AssertionError:
        print "Bad value passed to GetCorrFac(array1, array2)"
        sys.exit()
    corrfacs = egen/evis
    diff = (corrfacs.max()-corrfacs.min())/1000
    hist = rt.TH1D('cf','CF',\
            100, corrfacs.min()-diff,corrfacs.max()+diff)
    for i in corrfacs:
        hist.Fill(i)
    cf = hist.GetBinCenter(hist.GetMaximumBin())
    del hist
    return cf
```

BIBLIOGRAPHY

[1] Cyfronet, waiting for response about citation format.

[2] Mokka web resource. http://ilcsoft.desy.de/portal/software_packages/mokka/.

[3] MVD telescope web resource. http://www.desy.de/~gregor/MVD_Telescope/short_intro.html.

[4] D. Bédérède et al. SICAL - a high precision silicon-tungsten luminosity calorimeter for ALEPH. *Nuclear Instruments and Methods A*, 365:117–134, 1995.

[5] H.J. Bhabha. The scattering of positrons by electrons with exchange on Dirac's theory of the positron. *Proc.Roy.Soc.Lond.*, A154: 195–206, 1936.

[6] J. Błocki, W. Daniluk, E. Kielar, J. Kotuła, A. Moszczyński, K. Oliwa, B. Pawlik, W. Wierba, L. Zawiejski, and J. Aguilar. Silicon sensors prototype for lumical calorimeter. *Eudet-Memo-2009-07*, 2009. Available at http://www.eudet.org/e26/e28/.

[7] J. Błocki, W. Daniluk, E. Kielar, et al. LumiCal new mechanical structure. *Eudet-Memo-2009-10*, 2009.

[8] I. Bozovic-Jelisavcic, H Abramowicz, P. Bambade, T. Jovin, M. Pandurovic, B. Pawlik, C. Rimbault, I. Sadeh, and I. Smiljanic. Luminosity measurement at ILC. 2010.

[9] James Brau, Yasuhiro Okada, and Nicholas Walker. International linear collider reference design report vol. 1: Executive summary. Technical report, ILC Global Design Effort and Worldwide Study, 2007.

[10] DELPHI Collaboration. URL http://delphiwww.cern.ch/.

[11] M. Czakon, J. Gluza, and T. Riemann. Master integrals for massive two-loop bhabha scattering in QED. *Phys.Rev.*, D71:073009, 2005. doi: 10.1103/PhysRevD.71.073009.

[12] W. Daniluk, E. Kielar, J. Kotuła, A Moscczyński, K. Oliwa, B. Pawlik, W. Wierba, L. Zawiejski, and J.A. Aguilar. Redesign of LumiCal mechanical structure. *Eudet-Memo-2010-06*, 2010. Available at http://www.eudet.org/e26/e28/.

[13] M. Davier et al. Idag report on the validation of letters of intent for ilc detectors. Technical report, International Detector Advisory Group, August 2009.

[14] D. Decamp et al. ALEPH: A detector for electron-positron anni-
hilation at lep. *Nuclear Instruments and Methods A*, 294:121–178,
1990.

[15] Abdelhak Djouadi, Joseph Lykken, Yasuhiro Okada Klaus Mönig,
Mark Oreglia, and Satoru Yamashita. International linear collider
reference design report vol. 2: Physics at the ilc. Technical report,
ILC Global Design Effort and Worldwide Study, 2007.

[16] Abdelhak Djouadi, Joseph Lykken, Klaus Monig, Yasuhiro Okada,
Mark Oreglia, and Satoru Yamashita. International linear collider
reference design report vol. 4: Detectors. Technical report, ILC
Global Design Effort and Worldwide Study, 2007.

[17] ILC Global Design Effort and World Wide Study. International
Linear Collider reference design report. Technical report, 2007.

[18] D. Bederede et. al. SICAL - a high precision silicon-tungsten
luminosity calorimeter for aleph. *Nuc. Inst. Meth. A*, 1995.

[19] H. Abramowicz et al. Forward instrumentation for ilc detectors.
Journal of Instrumentation, 5(P12002), 2010.

[20] J.Allison et al. Geant4 developments and applications. *IEEE
Transactions on Nuclear Science*, 53(1):270–278, 2006.

[21] S. Kulis et. al. Test beam studies of the lumical prototype. Tech-
nical report, EUDET, September 2010.

[22] S.Agostinelli et al. Geant 4 - a simulation toolkit. *NIM-A*, 503(1):
250–303, 2003.

[23] Christian W. Fabjan and Fabiola Gianotti. Calorimetry for particle
physics. *Reviews of Modern Physics*, 75, October 2003.

[24] Rohini M. Godbole. LHC-ILC synergy. *Acta Physica Polonica B*,
37:1225–1236, 2006.

[25] Peter W. Higgs. Broken symmetries and the masses of gauge
bosons. *Phys. Rev. Lett.*, 13:508–509, Oct 1964. doi: 10.1103/
PhysRevLett.13.508. URL http://link.aps.org/doi/10.1103/
PhysRevLett.13.508.

[26] M. Idzik, S. Kulis, and D. Przyborowski. Development of front-
end electronics for the luminosity detector at ILC. *Nucl. Instrum.
Meth. A*, 608:169–174, 2009.

[27] M. Idzik, Sz. Kulis, and D. Przyborowski. Development of
front-end electronics for the luminosity detector at ilc. *Nu-
clear Instruments and Methods in Physics Research Section A: Ac-
celerators, Spectrometers, Detectors and Associated Equipment*, 608
(1):169 – 174, 2009. ISSN 0168-9002. doi: 10.1016/j.nima.2009.

06.059. URL http://www.sciencedirect.com/science/article/pii/S0168900209013151.

[28] ILC. Ilc web page.

[29] Standa Ltd. http://www.standa.lt/.

[30] A. Yamamoto M. Ross, N. Walker et al. Sb2009 proposal document. Technical report, ILC Global Design Effort, December 2009.

[31] Andre Miéde. Classic Thesis, 2011. http://www.miede.de/index.php?page=classicthesis. Accessed Jan 24, 2011.

[32] LHEP physics list. URL http://www.slac.stanford.edu/comp/physics/geant4/slac_physics_lists/ilc/lhep_physics_list.html.

[33] C Rimbault, P Bambade, K MÃ¶nig, and D Schulte. Impact of beam-beam effects on precision luminosity measurements at the ilc. *Journal of Instrumentation*, 2(09):P09001, 2007. URL http://stacks.iop.org/1748-0221/2/i=09/a=P09001.

[34] Iftach Sadeh. LumiCal thesis. Master's thesis, Tel Aviv University, 2009.

[35] Ringo Schmidt. *Optimisation of a Testbeam Setup and Background Estimates for Detectors at the ILC using Monte Carlo Simulations*. PhD thesis, Brandenburg University of Technology Cottbus, 2007.

[36] I. Smiljanic et al. Towards a final selection for luminosity measurement. In *Proceedings of the Workshop of the Collaboration on Forward Calorimetry at ILC*, 2008. Belgrade Serbia.

[37] A. Stahl. Luminosity measurement via bhabha scattering: Precision requirements for the luminosity calorimeter. *LC-DET-2005-004*, 2005.

[38] A. Stahl. Luminosity measurement via bhabha scattering: Precision requirements for the luminosity calorimeter. *LC Notes*, 2005. http://www-flc.desy.de/lcnotes/.

[39] W. Wierba, J. Błocki, W. Daniluk, A. Galas, E. Kielar, J. Kotuła, A. Moszczński, K. Oliwa, B. Pawlik, L. Suszycki, and L. Zawiejski. Lumical mechanical design proposal and integration with ild. *EUDET memo*, 2008.

[40] Richard Wigmans. *Calorimetry: Energy measurement in particle physics*. Number 107 in International Series of Monographs on Physics. Oxford University Press, 2000.