Data acquisition software development and physics studies for a future e^+e^- linear collider

Tom Coates

Submitted for the degree of Doctor of Philosophy

University of Sussex

March 2019

Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree.

Signature:

Tom Coates

UNIVERSITY OF SUSSEX

Tom Coates, Doctor of Philosophy

Data acquisition software development and physics studies for a future e^+e^- linear collider

Summary

[Summary text] [Max. 300 words for most subjects]

Acknowledgements

Contents

Chapter 1

Introduction

Teaching man his relatively small sphere
in creation, it also encourages him by its
lessons of the unity of Nature.

Annie Jump Cannon

[...]

1.1 The Standard Model

[...]

1.2 The Higgs Boson

[...]

1.3 CP violation in the Higgs sector

[...]

1.4 Data acquisition software and testbeams

[...]

Chapter 2

Future Linear Colliders

Progress is not a straight line.

An Wang

In the post-LHC era, the major unanswered questions in particle physics centre around the Higgs boson and its properties, the identification of additional sources of CP-violation that can account for the universe's abundance of matter and paucity of antimatter, and the discovery of physics beyond the Standard Model. There are many investigations into each of these fields that utilise the Large Hadron Collider, or will leverage the upgrades for the high-luminosity LHC (HL-LHC). However, now that the Higgs boson has been identified successfully, one of the most [x] avenues for further research is the construction and operation of a lepton collider with sufficient centre-of-mass energy to produce Higgs bosons. [...]

These are the primary motivations for the construction of future colliders, especially lepton colliders, to succeed the Large Hadron Collider. The two main candidates are the International Linear Collider (ILC) and the Compact Linear Collider (CLIC). Since both are linear eletron-positron colliders, they share many features, design considerations, and challenges,

2.1 Introduction

[...]

2.2 The physics case for a lepton collider

[...]

2.3 The International Linear Collider

[...]

The proposed site for the ILC is the Kitakami Highlands region of Iwate prefecture, Japan. [...]

2.3.1 The ILD and SiD detectors

[...]

One of the unique features of the ILC is the push-pull detector system. This is a moving platform in the chamber housing the interaction point, upon which two detectors can be mounted. The platform can be moved to change which detector is in the beamline, allowing a linear collider to function with multiple detectors. Switching detectors is expected to take [some] hours. This allows the two detectors to specialise for different physics studies and goals, much like the experiments at the Large Hadron Collider at CERN, which is normally not possible with linear colliders. [?] [...]

The International Large Detector (ILD)

[...]

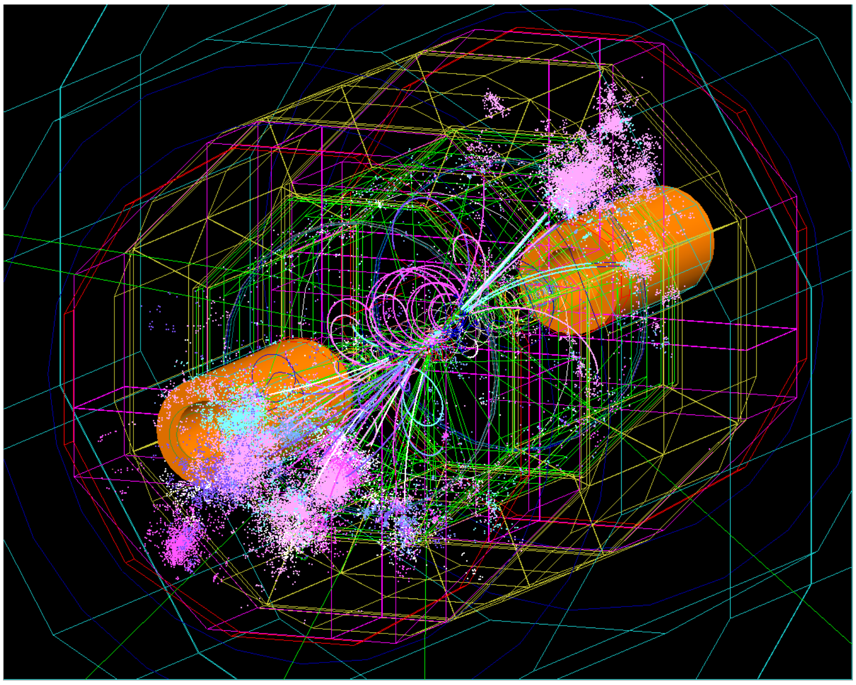

Figure 2.1: Visualisation of a simulated tth event in the ILD. Charged particles can be easily identified by their curved, coiled or spiral paths, and the jets are clearly visible as the light pink and purple areas near the beampipes on either side.

The Silicon Detector (SiD)

[...]

2.4 The Compact Linear Collider

[...]

[...] CLIC would be built beneath the existing LHC ring at CERN, stretching across the French-Swiss border and running parallel to the feet of the Jura mountain range. [...]

CLIC's initial centre-of-mass energy will be 380 GeV, with successive upgrades increasing it to 1.5 TeV and 3 TeV.

Chapter 3

Data acquisition software

Before software can be reusable
it first has to be usable.

Ralph Johnson

Data acquisition is a critical component of all particle physics experiments across all stages of technological readiness, from the very beginning of hardware testing in tabletop experiments to full-scale international experiments like the Large Hadron Collider.

In the modern era of particle physics, the interplay of hardware and software at minscule timescales drives everything, and almost all results are highly dependent upon the speed and efficiency of the electronics and computer systems that extract data from the detectors. A massive quantity of work goes into creating, testing and optimising the systems that will acquire, process, sort and transport data before it is ever seen by the physicist operating the experiment.

Of particular interest in this thesis is the data acquisition software during the development phase, where individual detector subcomponents are undergoing prototyping and testing. These development and iteration cycles are tied closely to testbeam facillities such as the Super Proton Synchrotron (SPS) at CERN and the DESY II synchrotron at DESY. At this point in the development cycle, the detectors are beginning to take shape and this is where data acquisition (or DAQ) becomes an important consideration.

In addition to this, the data acquisition solutions used during the testbeam phase of detector development is likely to inform the final data acquisition solution, either directly by evolving into the final software, or indirectly by identifying and evaluating the particular features or challenges of the subdetector components that the software must take into account or accommodate.

During this stage, each individual detector component – such as a vertex tracker or hadronic calorimeter – will be developed by small teams, and the natural tendency is for each of these groups to set their own standards and develop their own tools, prioritising the features that are important to their specific case. However, in the past this approach has generated a variety of *ad hoc* solutions for testbeam software, many of which cannot be applied outside of their original scope. The also results in wasted effort and time, as different teams implement the same solutions anew for each subdetector.

One of the aims of the AIDA-2020 project is to improve this situation by developing generic and reusable software tools for testbeams and particle physics experiments.

The AIDA-2020 project

The AIDA-2020 project is an EU-funded research programme for developing infrastructure and technologies for particle physics detector development and testing, comprising 24 member countries and lead by CERN.

The overarching goal of the project is to develop common infrastructures and tools for physics testbeams, and software is one such important tool. By creating a suite of tools that are designed with a variety of uses in mind, the amount of effort and development time necessary to plan and implement data acquisition and monitoring setups can be significantly reduced or eliminated, speeding up the planning and deployment of physics testbeams. This allows more science to be done faster. The two tools within AIDA-2020 that facillitate this are EUDAQ and DQM4hep, discussed in more detail below.

[My contributions to this chapter]

3.1 EUDAQ

[...]

3.2 DQM4hep

Data Quality Monitoring for High-Energy Physics (abbreviated DQM4hep) is an online monitoring and data quality monitoring framework developed for physics testbeams for high-energy and particle physics. It is designed to be able to fulfil the requirements of monitoring for physics testbeams in a generic way. The structure of the program allows for independent components of the framework to be used, not used, or exchanged, by isolating each function of the program into specific and independent processes. The components that are specific to particular users – the file readers, event streamers, and analysis and standalone modules – are written in standard C++ code, meaning they are capable of performing any data unpacking, processing or analysis that is necessary. The framework then handles packaging this information in a useful way and networking to transmit it to where it is needed, meaning that the user does not have to worry about the mechanics of data storage, serialisation or transmission. It also means that the framework does not need special rules for handling particular datatypes, allowing it to handle *anything* that can be packed into, decoded from, and accessed by normal C++ methods. This results in a framework that is able to deal with any kind of data, including user-defined data types, making it more flexible, portable and easily reusable.

3.2.1 Prerequisites and dependencies

DQM4hep is a C++ application, written in the C++11 standard, that can run on Windows, OSX, or any Linux distribution [?]. The only requirements for installation are a C++11-compliant compiler, cmake 3.4 or higher, and ROOT 6. All other prerequisites or dependencies are downloaded and compiled by the framework's installer.

[...]

3.2.2 Programming paradigms and structure

[...]

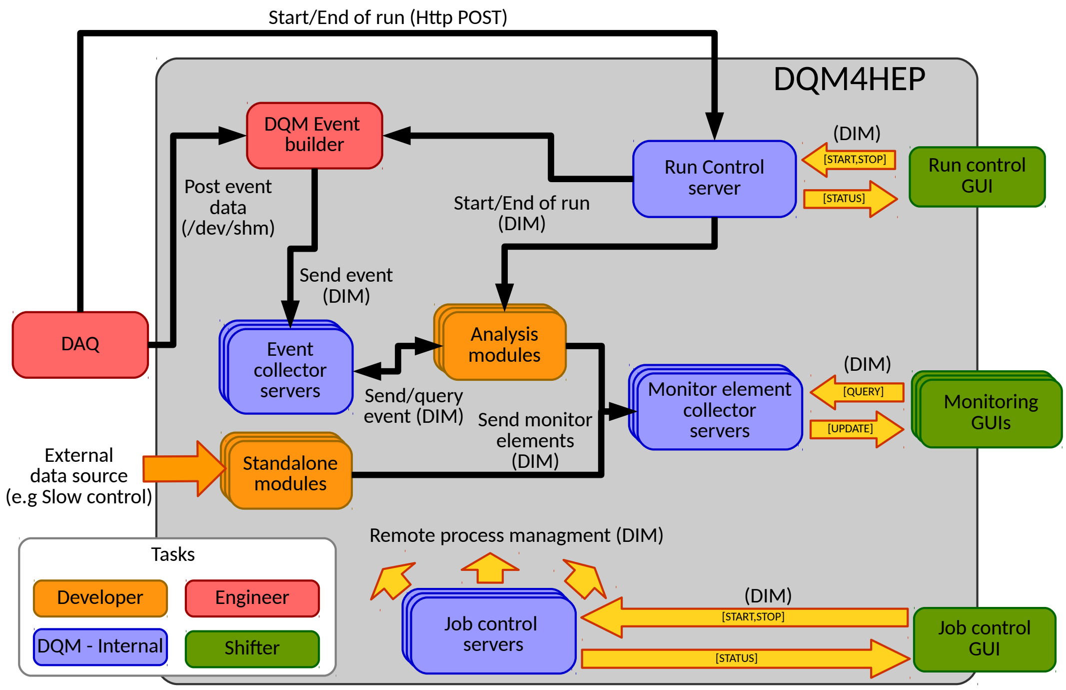

Figure 3.1: The global online architecture of DQM4hep.

3.2.3 Visualisation and graphical user interface

[...]

3.2.4 Data quality testing

One of the important areas of DQM4hep that was not yet completed was data quality monitoring, which is an array of tests or programs that assess the data being taken in real time to allow testbeam operators and shifters without detailed knowledge of the hardware, software, or physics to determine whether the device under test is performing as intended, and to quickly identify and address any errors or inconsistencies. Data quality monitoring (DQM) uses a variety of methods for measuring the 'quality' or 'goodness' of data, mainly relying upon statistical or comparative methods.

DQM4hep did not have any infrastructure to support data quality tests, but this was added during the core refactoring for the release version [?]. Once this was in place, a variety of data quality tests were developed and implemented, ranging from basic tests, such as comparing the mean of a data sample against predefined values of mean and standard deviation, to more complex tests, such as the Kolmogorov-Smirnov test, which is a comparison between a sample and a reference histogram.

[Illustrative pictures would go here; one of a pair of histograms demonstrating the quality of data with a mean/stddev test, and the other demonstrating a reference histogram, then by the side of it a picture of a sample histogram compared to the reference, maybe colour-highlighted?]

[...]

3.3 Adaptation to other detectors

[...]

To utilise DQM4hep with any new detector, either two or three new plugins must be created depending on whether the detector is to be monitored online, offline, or both.

If the data is to be monitored offline, then a file reader plugin must be written. If the data is to be monitored online, then a streamer plugin must be written. Both of these plugins are similar in structure and differ only on where they get the data from – a file reader loads a file from disk, whereas a streamer loads it from the data acquisition system. Once the information is accessible from the reader plugin, it packages the data into events, and emits them to the framework's network handling to be received by any other plugins that are listening for them. The incoming data can be of any type, since the methods for reading it are provided in the reader plugin itself. Previously, reader plugins have been created to read binary files, raw text files, SLCIO data files, and ROOT trees.

An analysis module is a type of plugin which takes data that is packaged into events by a reader plugin and performs some analysis on it. Each analysis module can only read events from one reader, but [...]

Writing new readers and analysis modules is relatively simple, especially if the data has already been packaged into well-structured formats such as ROOT or SLCIO. [...]

3.4 Integration with EUDAQ

[...]

3.5 Documentation and user guide

[...]

[Doxygen]

Of importance was producing a set of user information distinct from the developer documentation. These would be user's guides and walkthroughs, intended for *users* of the framework, who have no interest in the mechanics of the program, but simply using it for their testbeam.

[...]

3.5.1 File reader plugins

[...]

3.5.2 File streamer plugins

[...]

3.5.3 Analysis and standalone modules

[...]

Chapter 4

Testbeams

I love fools' experiments.
I am always making them.

Charles Darwin

One of the most important aspects of testing and developing DQM4hep was to ensure that it was as generic as it was intended to be, and this meant deploying and using the framework on physics testbeams.

[...]

4.1 Introduction

Regular testbeams were held at the DESY II synchrotron at DESY in Hamburg and at the Super Proton Synchrotron (SPS) at CERN.

[...]

DQM4hep was largely tested and developed during testbeams of the SiWECAL [...]

4.2 CALICE testbeams

[...]

[...] DQM4hep was used in testbeams of the AHCAL over the course of the years 2016-2018, occurring predominantly at the DESY II facility, with one in May 2017 taking place at the CERN SPS.

4.2.1 May 2016 at DESY II

[...] [This was the three-week one, where we got some real shit done.]

4.2.2 December 2016 at DESY II

[...]

4.2.3 May 2017 at CERN SPS

[...] During the CERN testbeam, development of the analysis modules for DQM4hep had largely been completed. This meant that after the initial set-up stage, very little management of the monitoring software was necessary, and instead it was simply being used during shifts to monitor the beam and hardware, as intended. The monitor was used [...]

[Pictures]

4.3 DREAM combined testbeam

[...]

The combined testbeam took place between 5th-12th September 2018 at the CERN SPS beamline facility. [...]

Importantly, none of these detectors or the teams responsible for their construction and operation were part of AIDA-2020, which was useful as a testbed for the generic nature of the DQM4hep framework – previous testbeams had only used AIDA-2020 detectors, many of which used filetypes or structures defined within the collaboration, which were already supported by the framework. By attempting to use DQM4hep to monitor non-AIDA-2020, it was possible to test that the design of the framework was truly generic and adaptable to any kind of detector with any filetype.

4.3.1 Detectors present at the combined testbeam

The combined testbeam comprised four separate dectors: a calorimeter, a muon detector and preshower, a drift chamber, and a silicon photomultiplier. One of the biggest challenges involved in the testbeam was operating these four different detectors [...]

RD52 calorimeter

The calorimeter was formed of two layers of 36 tiles each, totaling 72 tiles, stacked behind each other. One layer used Cherenkov detectors, the other used scintillator tiles. In addition, there was a group of leakage detectors that detected whether individual events were contained within the calorimeter or not. [DWC - Delayed Wire Chamber?]

Muon chamber and preshower

[...]

Silicon photomultiplier GEM

[...]

Drift chamber

[...]

4.3.2 Results

Existing monitoring within the DREAM collaboration could produce accurate histograms with raw data, creating simple plots of the total energy of each channel per event. This facility was reproduced in DQM4hep quickly using for-loops in both the C++ code and XML steering files, allowing this to be done with comparatively little code.

Further to this, the flexibility of using C++ code rather than ROOT macros allowed some analysis to be done in a nearly-online fashion. One of the first important quantities is R, also called the energy ratio:

$$R = \frac{E_1}{\sum_{i=1}^{10} E_i}$$

Where E_i is the energy of the i^{th} most energetic channel in the event, e.g. E_1 is the most energetic channel. Once the ratio R is calculated, a plot can be made of E_{total} vs. R for an entire run that shows separation of electrons from muons and pions:

[Figure]

Using an appropriate cut on this plot, it is possible to distinguish electron events. Adding in the information from the muon detector [the one in RD52], muons and pions can also be separated. Using this, we can then produce spectra for each type of particle in the run:

[Figure]

[...]

Chapter 5

Physics studies for the Compact Linear Collider

Somewhere, something incredible is
waiting to be known.

Carl Sagan

One of the primary goals of future lepton colliders like the ILC and CLIC is to become "Higgs factories" – machines that can produce large numbers of Higgs bosons in a variety of final states, allowing the Higgs sector of the Standard Model to be probed with unprecedented accuracy and coverage.

One of the uniquely accessible measurements for these colliders is a precision measurement of the top-Higgs Yukawa coupling. This serves as a further test for the Standard Model and [...]

Another important avenue to pursue is CP-violation in the Higgs sector. Since Higgs physics is still an emerging field, it is not yet known whether CP-violation is present in the Higgs sector to the degree that the Standard Model predicts. It is also a fertile area for investigation of BSM physics, as many BSM models predict additional Higgs bosons, or Higgs bosons with characteristics that differ from the SM Higgs.

The $e^+e^- \to t\bar{t}h$ event (see Fig. 5.1) is one process that is both accessible to CLIC's design energy and extremely useful for interrogating the Higgs sector for CP-violation and BSM physics. The production of Higgs bosons allows for several observables that would be sensitive to any Higgs bosons with an odd CP quantum number (or "CP-odd" Higgs bosons). Determining the detectors' sensitivity to the ratio of CP-odd and CP-even Higgs bosons (also called the CP mixing angle) will allow further understanding of the limits of the Standard Model, as well as the limits on the various BSM physics models, and regions of interest for possible new physics.

[...]

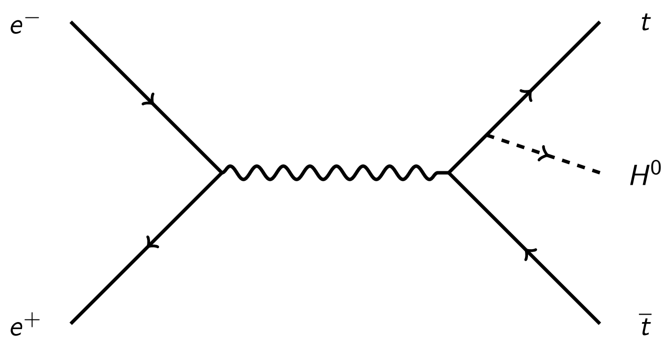

Figure 5.1: A Feynman diagram of the tth event.

There are three final states of the tth event, which depend on the decays of the $W^{\pm}$ boson. The $W^{\pm}$ can decay into either a quark-antiquark pair, or a lepton-neutrino pair, so there are three possible final states: the *fully hadronic* case, where both $W^{\pm}$ particles decay into quark pairs; the *leptonic* case, where both decay into lepton-neutrino pairs; and the *semi-leptonic* case, where one decays into a quark pair and the other into a lepton-neutrino pair. In general, the leptonic final state is not utilised for this analysis, so is not discussed further. Extended Feynman diagrams of the fully hadronic and semi-leptonic final states are shown in Figs. 5.2 and 5.3

[...]

[...] The invariant mass of the Higgs boson can be determined by summing the invariant masses of pairs of bottom quarks and computing the χ^2 for each possible combination; the combination with the lowest χ^2 shows the pair that has decayed from the Higgs boson.

[...]

5.1 Physics generation and samples

[...]

The Monte Carlo samples were generated predominantly using Whizard 1.95, though for the Higgs events, Physsim was used due to technical contraints of Whizard [ref?]. All samples were simulated at $\sqrt{s} = 1.4 TeV$ and unpolarised beams, assuming an integrated luminosity of 1.5 ab^{-1} and a light Standard Model Higgs boson with mass 125 GeV/c^2. See Table 5.1 for a summary of all of the used samples. The first two rows are the ttH signal channels, all other rows are background. The number of jets refers to the number of jets in the final state that have come from the decay of the top-quark pair. The number of

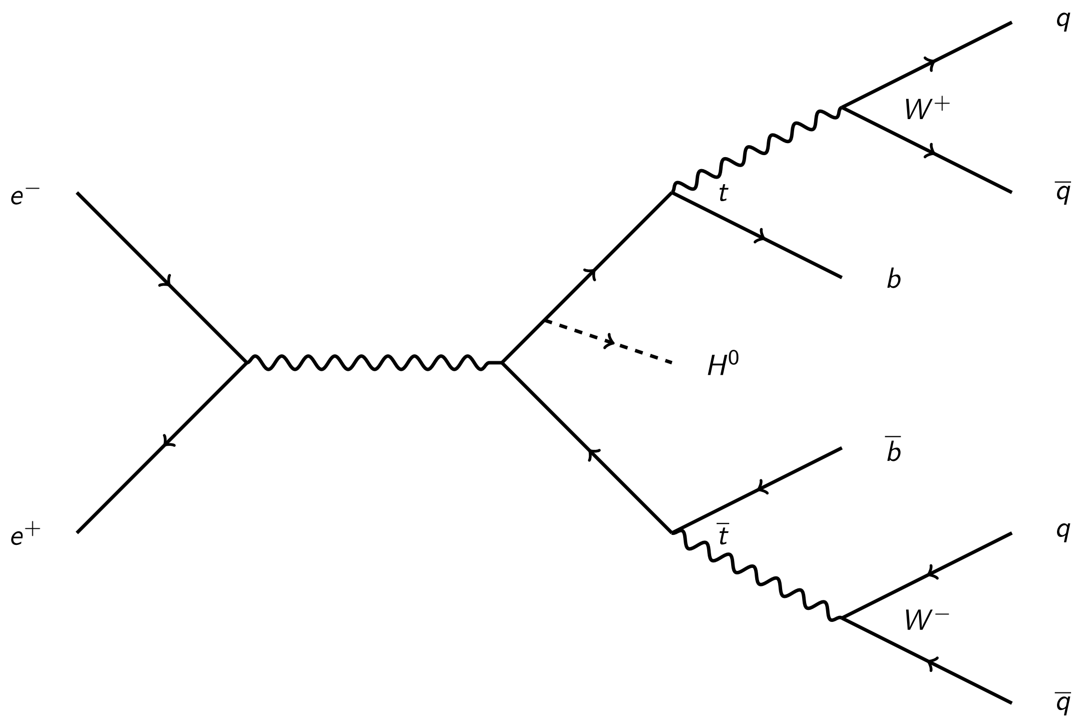

Figure 5.2: An extended Feynman diagram of the tth event, showing the fully-hadronic decay channel with the final state $q\overline{q}q\overline{q}b\overline{b}$, where q and $\overline{q}$ indicate a quark-antiquark pair.

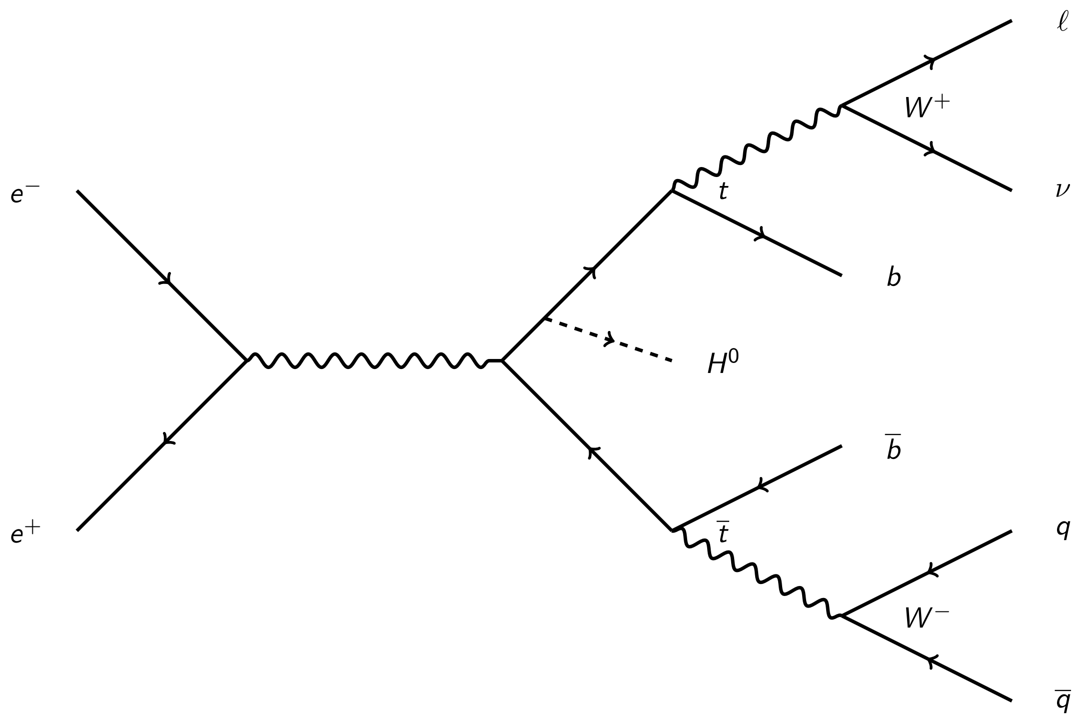

Figure 5.3: An extended Feynman diagram of the tth event, showing the semi-hadronic decay channel with the final state $\ell\nu q\overline{q}b\overline{b}$, where ℓ and ν indicate a lepton-neutrino pair of the same flavour but opposite [sign?].

events in 1.5 ab^{-1} has been calculated from the integrated luminosity and sample weight.

ProdID	Process	Cross-section (fb)	Sample weight	Events in 1.5 ab^{-1}
2435	$t\bar{t}H$, 6 jets, $H \to b\bar{b}$	0.431	0.03	647
2441	$t\bar{t}H$, 4 jets, $H \to b\bar{b}$	0.415	0.03	623
2429	$t\bar{t}H$, 2 jets, $H \to b\bar{b}$	0.100	0.006	150
2438	$t\bar{t}H$, 6 jets, $H \not\to b\bar{b}$	0.315	0.02	473
2444	$t\bar{t}H$, 4 jets, $H \not\to b\bar{b}$	0.303	0.02	455
2432	$t\bar{t}H$, 2 jets, $H \not\to b\bar{b}$	0.073	0.004	110
2450	$t\bar{t}Z$, 6 jets	1.895	0.1	2843
2453	$t\bar{t}Z$, 4 jets	1.825	0.1	2738
2447	$t\bar{t}Z$, 2 jets	0.439	0.03	659
2423	$t\bar{t}b\bar{b}$, 6 jets	0.549	0.03	824
2426	$t\bar{t}b\bar{b}$, 4 jets	0.529	0.03	794
2420	$t\bar{t}b\bar{b}$, 2 jets	0.127	0.008	191
2417	$t\bar{t}$	125.8	1.5	203700

Table 5.1: Table of all signal and background samples used for this analysis.

5.2 Detector models

[...]

[...] henceforth referred to as CLIC_SiD.

5.3 Sensitivity to cross-sections and Yukawa coupling

[...]

5.3.1 Analysis method

[...]

Sample processing

The first step is the initial jet clustering. This is done using the k_t algorithm with parameters [?], using an exclusive clustering mode to form 8 jets – 6 jets from the produced quarks and 2 beam jets. The k_t algorithm is used over choices like anti-k_t and Valencia, because the important features are the *relative* shapes of the jets, rather than absolute properties, so there is no need to use more computationally-intensive [is this true?] algorithms.

Once initial jet clustering is finished, a Marlin processor that finds isolated leptons is used. It searches for either 0, 1, or 2 isolated leptons, and this information can be used to make decisions about whether to process the event already:

Leptons	Channel	Action
0	Fully hadronic	Use for fully hadronic analysis
1	Semi-leptonic	Use for semi-leptonic analysis
2	Fully leptonic	Discard

Following this, the two beam jets are removed from the processing, and a further step of jet re-clustering is performed, using the Durham algorithm, to [...]

[...] [Flavour tagging]

[...] [Tau finding]

The final step is to use use PandoraPFAs to generate Particle Flow Objects (PFOs) of the undetectable particles, especially the top quarks, $W^{\pm}$, and Higgs. [...]

Analysis processing [?]

Once the sample has been processed, it must be analysed. The first step of this is a program used to extract various kinematic variables of both particles in the events (m_0, p_t) and the event itself (Ψ, T). This is the Treemaker program, and [...] [Chi-squared extraction of invariant masses] [Feeding into TMVA to generate BDTs]

[...]

A flow diagram of the analysis process, and the rejection points, is shown in Fig. 5.4.

[...]

5.3.2 Results

[...]

The combined uncertainty for the cross-section of both decay channels is:

Cross-section:

$$\Delta\sigma = 7.30\%$$

Then using this and a linear approximation from QCD [ref], the value of the uncertainty on the top-Higgs Yukawa coupling can be computed:

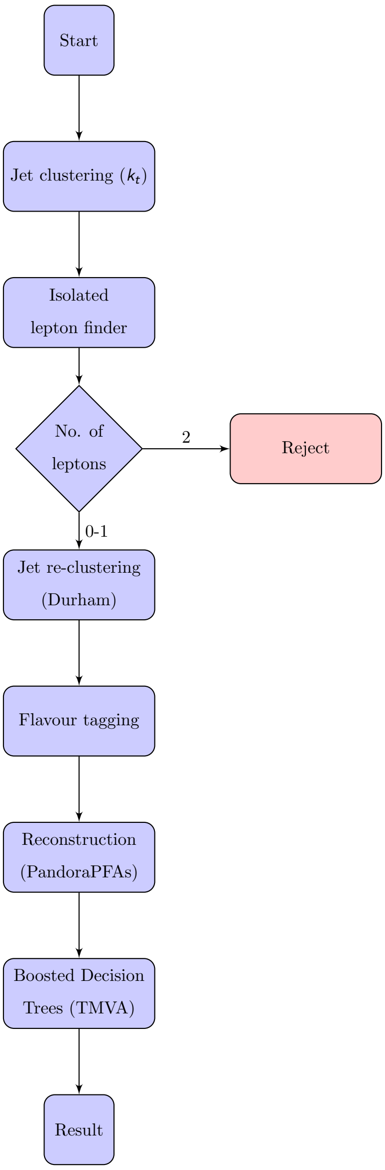

Figure 5.4: A flow diagram of the algorithm for analysis of ttH events, and rejection.

$$\frac{\Delta g_{tth}}{g_{tth}} = 0.503 \frac{\Delta\sigma(t\bar{t}H)}{\sigma(t\bar{t}H)} = 3.86\%$$

These results were contributed to a paper that summarised the top physics potential for CLIC at $\sqrt{s}$ = 1.4 TeV, published in [journal][ref] and will be submitted to CERN's European Strategy Update in [month] 2019 [ref].

5.4 Determination of sensitivity to CP-violation

[...]

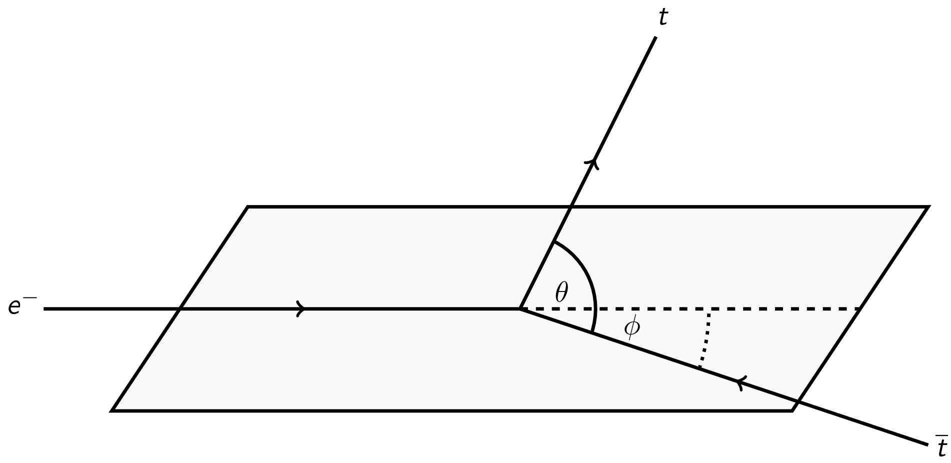

Figure 5.5: Geometric diagram of the up-down asymmetry in tth events. The paths of the electron and antitop quark, and the angle ϕ between them define a plane. “Up-going” top quarks go above the plane, “down-going” top quarks go below.

5.4.1 CP-sensitive observables

[...]

Up-down asymmetry

The up-down asymmetry is a conceptually simple observable that has already been identified for investigating CP-violation in the tth process. It is found by defining a plane from the vectors of the incoming electron and produced antitop, then finding the ratio of top quarks that are emitted above and below this plane (see Fig. 5.5). If there is no CP-violation in this process, the ratio will be even.

Using the up-down asymmetry as an observable requires that the W^+ and W^- can be distinguished from each other, and has thus far only been used for the semi-leptonic decay channel. In this case, the lepton produced by the decay of one of the W bosons identifies its charge, and thus the charge of the top quark that it has decayed from. While this method was not previously possible in the fully hadronic case, a method for applying it by using jet charge determination is discussed in Section 5.4.2.

[other ones]

[...]

5.4.2 Jet charge determination

Previous analyses that have utilised the up-down asymmetry as an observable have focused exclusively on the semi-leptonic decay channel of tth events, as the presence of a lepton emitted by the top or antitop quark offers a simple and statistically robust method to distinguish between the two top quarks. In the hadronic decay channel each top emits a jet, and even in the ideal case where each particle resulting from the jet can be accurately reconstructed, the net charge will *still* be an integer, since no particles with a non-integer charge can result from these decays.

However, techniques developed in recent years, intended primarily for observations in ATLAS at the LHC, have refined methods for this, and using the work of [reference], it is now possible to obtain the total net charge of the jet – that is, the charge of the initial quark that creates the jet.

This technique is strongly-dependent upon the accuracy and efficiency of both particle reconstruction and jet clustering, but these techniques are constantly improving, and Pandora Particle Flow Algorithms (PFAs) and new jet clustering methods are becoming increasingly sophisticated. Combined with the cleaner final states in a lepton collider, these techniques allow the charge of a jet to be determined with useful confidence.

The charge of a jet can be determined by summing the charges of all particles in the jet, weighted by p_T and normalised by the p_T of the entire jet:

$$\mathcal{Q}^i_\kappa = \frac{1}{(p_T^{jet})^\kappa} \sum_{j \in jet} Q_j (p_T^j)^\kappa$$

Where κ is some parameter between 0 and 1, typically set to 1. With this technique, it is possible to determine between quarks with charges of $+1/3e$, $-1/3e$, $+2/3e$, and $-2/3e$ with [some level of confidence].

Jet clustering algorithms

Since this method relies upon jets it is strongly dependent upon jet reconstruction, and thus on the choice of jet clustering algorithm and parameters. Previous analyses of tth events have used a two-step reclustering approach, using the k_t algorithm for the initial clustering and the Durham algorithm for reclustering. These algorithms were chosen as the relative difference between the jet shapes is more important than their absolute shapes, so other algorithms do not provide any benefits.

The Valencia algorithm, however, gives improved performance in the cleaner final states of a lepton collider, for which it was especially designed, and many analyses are now transitioning to using the Valencia algorithm.

5.4.3 Results

[...]

Chapter 6

Discussion and Conclusions

What we know is really very, very little
compared to what we still have to know.

Fabiola Gianotti

[...]

Appendix A

Code

```
10 PRINT "LOOK AROUND YOU"
20 GOTO 10
```