First validation of the k4MarlinWrapper for ILD standard reconstruction workflows.

DESY Summer Student Programme, 2021

Blae Stacey

University of Leeds

Supervisor

Thomas Madlener

September 8, 2021

Abstract

The Key4hep common software framework aims to obtain a common software stack that can be used by all future collider projects, and uses Gaudi as its experimental framework. The Gaudi experimental framework is used by multiple experiments, whilst the linear collider communities, ILC and CLIC, have collaboratively developed the Marlin experimental framework together. In order to ease the transition for these communities to the Gaudi framework, the k4MarlinWrapper has been developed. The wrapper enables Marlin processors to run unchanged within the Gaudi framework.

This work explores the use of the k4MarlinWrapper in the running of the ILD standard reconstruction chain for the first time. It compares the results obtained by running this chain inside the Marlin framework, with the results obtained from running it using the k4MarlinWrapper inside the Gaudi framework. The ILD standard reconstruction ran successfully inside the Gaudi framework, without any major issues. There was close agreement between the output from the Marlin experiment framework and the k4MarlinWrapper inside the Gaudi framework, although some small discrepancies were discovered and there is still work to be done in this area.

The event data model used in the linear collider communities, LCIO, can be converted into the corresponding event data model for the Key4hep environment, EDM4hep. This conversion was attempted to allow for comparison between the two outputs, which could aid in the identification of any yet unknown problems. The EDM4hep conversion from LCIO was not successful and more work is needed in this area before a comparison can be made between the two.

Contents

1 Introduction

Until now high energy physics, HEP, has not focused on a universal experimental framework for all colliders, although one notable exception is in the collaborative development of the Marlin [1] experimental framework by CLIC and ILC [2]. The Key4hep environment is largely built around the iLCSoft environment, jointly developed by CLIC and ILC, and uses some of the core iLCSoft tools, such as DD4hep, Detector Description for HEP, and PandoraPFA, Pandora Particle Flow Analysis [3]. Another experimental framework that is used by more than one community is Gaudi [4], which was initially designed for LHCb, but is now also used by ATLAS, FCCSW, and smaller experiments. Key4hep has chosen to adopt Gaudi as its experiment framework and will contribute to its development where necessary, as it supports concurrency, is tried and tested from data taking during LHC operations, and is currently under going a modernization. [5]

The integration and migration of the iLCSoft algorithms into Key4hep is in progress to aid in the transition to the Key4hep framework for the linear collider communities, and attempts this through the development of the k4MarlinWrapper [6]. The process of integrating these algorithms is ongoing, as the k4MarlinWrapper has been developed but is still undergoing testing. This work attempts a comparison between running reconstruction using the Marlin processors via the k4MarlinWrapper in the Gaudi framework and using the Marlin processors in the Marlin framework, and it is one of the first to compare the two on a larger scale. The running of the ILD standard reconstruction chain [7] inside the Gaudi framework is tested for the first time in this work.

This project initially investigated the validity of the k4MarlinWrapper by making a direct comparison between the reconstruction of a single set of simulated particle events using the Marlin processor inside the Marlin framework, and the reconstruction of the same set of simulated particle events using the k4MarlinWrapper inside the Gaudi framework.

The comparison in this work was made over a single set of 10,983 simulated events, using only the events where two jets and two muons were reconstructed. The events were simulations of the Higgs recoil process:

$$e^+e^- \to ZH \to \mu^+\mu^- X \tag{1}$$

From the electron-positron collision, a Higgs boson and a Z-boson are produced. The Z-boson then decays into two muons, $\mu^+\mu^-$ and the Higgs boson decays inclusively. The initial collision energy of the electron-positron pair was 250 GeV, so by subtracting the di-muon mass from this, we are left with all of the energy that went into producing the Higgs boson, which gives us the Higgs recoil mass. The results for the Higgs recoil mass, the invariant di-muon mass, and the invariant di-jet mass will allow for comparison between the output from the use of the Marlin framework and that of the k4MarlinWrapper in the Gaudi framework. If the k4MarlinWrapper functions perfectly, the reconstructions should produce results identical to each other.

The reconstructions using both the Marlin and the Gaudi frameworks produce REC files, containing the data. Due to the size of these files, and their resultant lack of suitability for long term storage, the files can be converted to miniDST files through a miniDST workflow, a further set of Marlin processors. These miniDST files can then be used to make the comparison between the two types of reconstruction and investigate any differences.

The reconstructions using the Marlin processors in either framework produce an LCIO [8] output, the event data model for the iLCSoft environment. Further investigation into the progress of the migration from iLCSoft to Key4hep can be attempted through converting the LCIO output to an EDM4hep output, the event data model for Key4hep [9]. A comparison can then be made between the LCIO and EDM4hep outputs to explore whether the converter has any yet undetected conversion problems. The LCIO output will be generated from the reconstruction using the Marlin processors via the k4MarlinWrapper in the Gaudi framework, and then be passed through the conversion tool, and so the LCIO and EDM4hep outputs should be identical, if there are no conversion issues. The conversion has previously been successfully ran on the CLIC workflows, so the aim of this work is to attempt the conversion on the ILD workflows.

2 Results

The comparison made was between the Marlin processor in the Marlin experimental framework and the k4MarlinWrapper in the Gaudi experimental framework, over a single set of 10,983 simulated events of the Higgs recoil process. It was found during some preliminary work that the random seed used for the reconstruction in the Marlin experimental framework must be fixed to the same value used in the k4MarlinWrapper, 123456, in order for the two reconstructions to be comparable. The random seed in the k4MarlinWrapper is hard-coded and not easily changeable, but the random seed in the Marlin framework is changeable from the command line. This shows an area of under-development in the k4MarlinWrapper, compared to the Marlin framework, but is only due to a lack of available work power in this area.

There was found to be a large discrepancy in the number of events between the two reconstructions, which was not present in the reconstructed DST and REC files, only in the miniDST files which were used to construct the comparison plots. Initially, this pointed to a problem in the miniDST workflow but it was later found to be an issue in some of the Marlin processors. The IsolatedLeptonTagging processor was being passed an empty collection, and because of this did not produce a collection, which meant that the TaJetClustering processor was not being passed a collection, causing the workflow to crash. The Marlin and the Gaudi frameworks handled this case differently, Marlin aborted at this error and produced a broken file, but Gaudi was better equipped to handle it and still produced a complete file, albeit with missing events. There is more information about this issue available here: https://github.com/iLCSoft/MarlinReco/issues/93

2.1 Comparison of Successful Batch Jobs

In order to bypass the errors resulting from the miniDST workflow, a comparison was made using only the batch jobs where all of the events were successful through the miniDST workflow in both reconstructions. This meant there were 5313 events, 48% of the total events, available for the comparison in each reconstruction. Of these 5313 events, 4568 were identified as having two jets and two muons for the reconstruction in the Marlin experimental framework, and 4558 for the reconstruction using the k4MarlinWrapper in the Gaudi experimental framework.

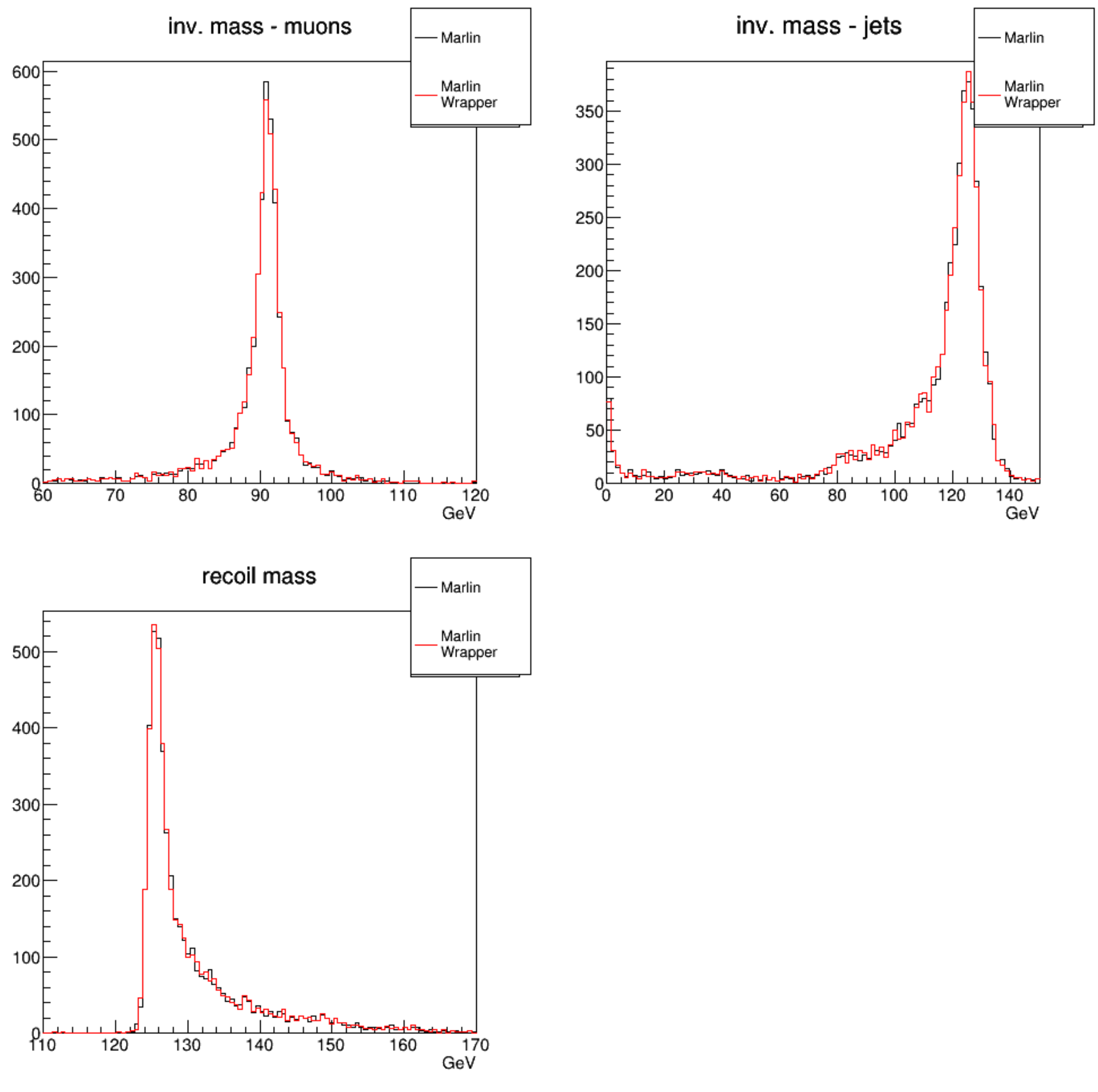

Figure 1: Comparison of invariant di-muon mass (a), invariant di-jet mass (b), and Higgs recoil mass (c) using only the events that were successful during the miniDST workflow.

The discrepancy in the number of events for the two reconstructions was small (10 events, 0.2%) and had likely arisen from slightly different handling of the random seeds within the two frameworks. Figure 1 shows differences between the two reconstructions, despite the fixed seed and removal of errors introduced during the miniDST workflow. This did not clearly point to a single source of error, so further investigation was needed.

2.2 Full Comparison

In order to negate the impact of any errors introduced to the data by the miniDST workflow crashes, a comparison was produced between the reconstruction in the Marlin framework and the reconstruction using the k4MarlinWrapper in the Gaudi framework, using an adapted miniDST workflow to avoid the previous crashes. The nature of the adapted miniDST workflow removed the requirement of two jets, as the workflow ended after the IsolatedMuonTagging processor, and before identifying the jets.

There were 9093 events available for the comparison, reduced from the original 10,983 because of errors with the batch system, and of these 9093 events, 8079 had two muons in the Marlin framework, and 8092 in the Gaudi experiment framework. The difference in the number of events (13 events, 0.16%) was very small and, again, was likely to have come from slightly different handling of the random seed.

Figure 2 shows some small discrepancies between the two reconstructions, similar to the initial comparison in Figure 1, but with a smaller impact due to the larger number of events included in the comparison. Further investigation into the handling of the random seed and the nature of the Marlin processors is needed in order to align perfectly the output of reconstructions using the k4MarlinWrapper inside the Gaudi framework with the output of reconstructions using the Marlin processors inside the Marlin framework.

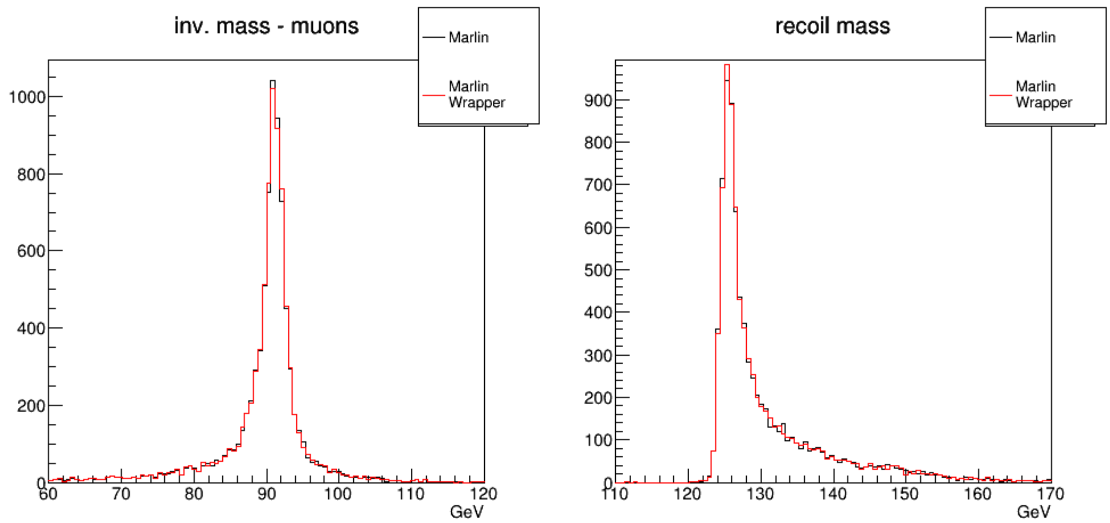

Figure 2: Comparison of invariant di-muon mass (a) and Higgs recoil mass (b) over all available events.

2.3 Marlin Comparison

By running a comparison between two reconstructions, both done in the Marlin experimental framework and using the same random seeds, it was possible to verify whether the original Marlin reconstruction is consistent, as had initially been supposed. The original miniDST workflow was used in this comparison, as it was found from Figure 2 that this did not have a significant impact on any discrepancies. There were 6123 events read from each set of the miniDST files, and of these events, 5265 had two jets and two muons. This was the same for both of the reconstructions, which further indicates that the previous event discrepancy between the reconstructions in the Marlin and Gaudi frameworks came from different handling of the random seed.

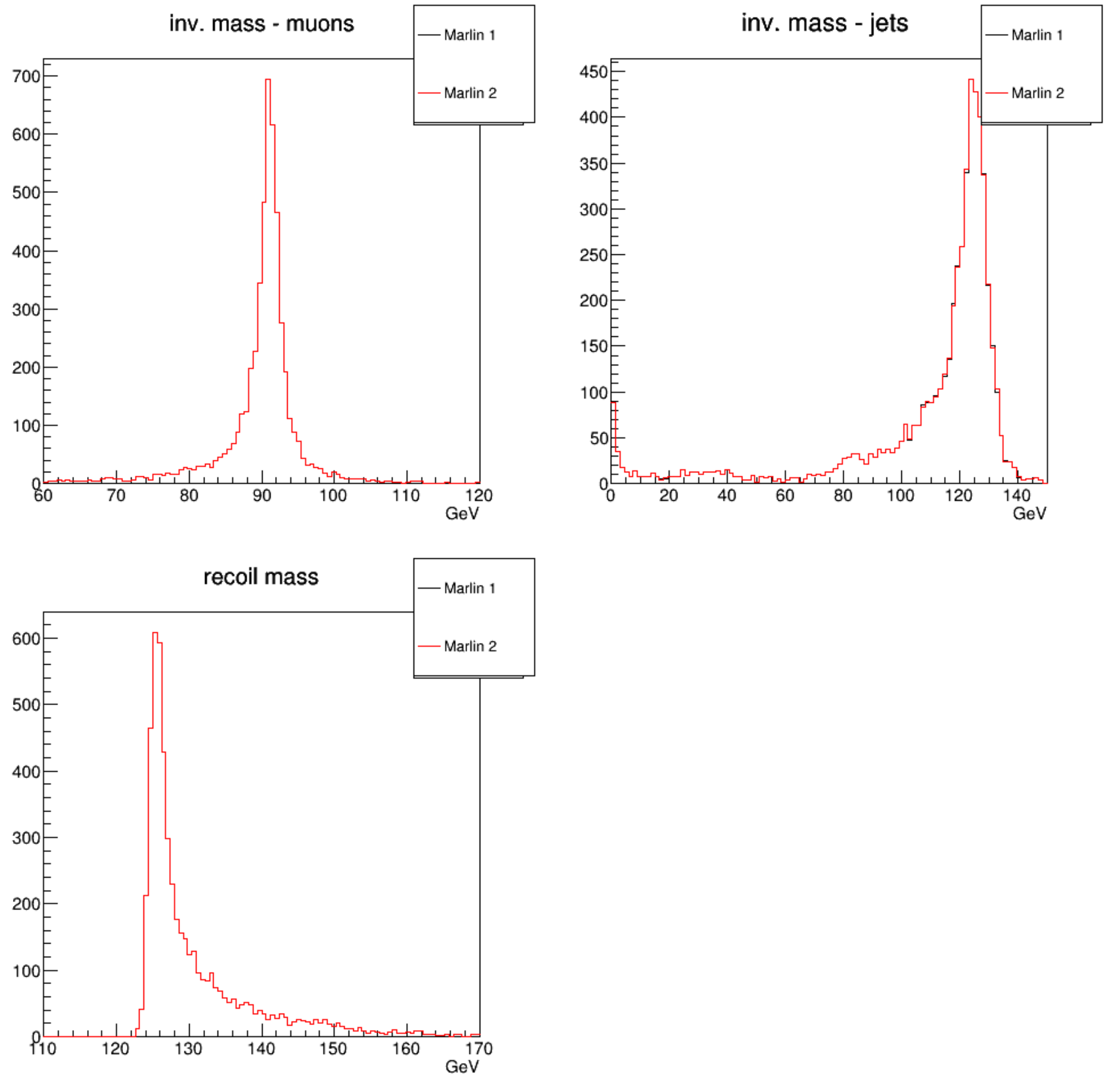

Figure 3: Comparison of invariant di-muon mass (a), invariant di-jet mass (b), and Higgs recoil mass (c) from two runs of a reconstruction in the Marlin experimental framework.

It can be seen from Figures 3 and 4, that the results for the invariant di-muon mass (a) and the Higgs recoil mass (c) are reproducible with multiple runs of the reconstruction, but the results for the invariant di-jet mass are not consistent. This result for the invariant di-jet mass was unexpected, as the basis of this work relied upon the output from the Marlin processor in the Marlin experimental framework being consistent, in order to be able to replicate this through the k4MarlinWrapper in the Gaudi experimental framework. The differences present in Figure 4(b) have arisen from a difference in the kinematics of the jets, however, it is currently unknown what is causing these, and further work by the experts is required.

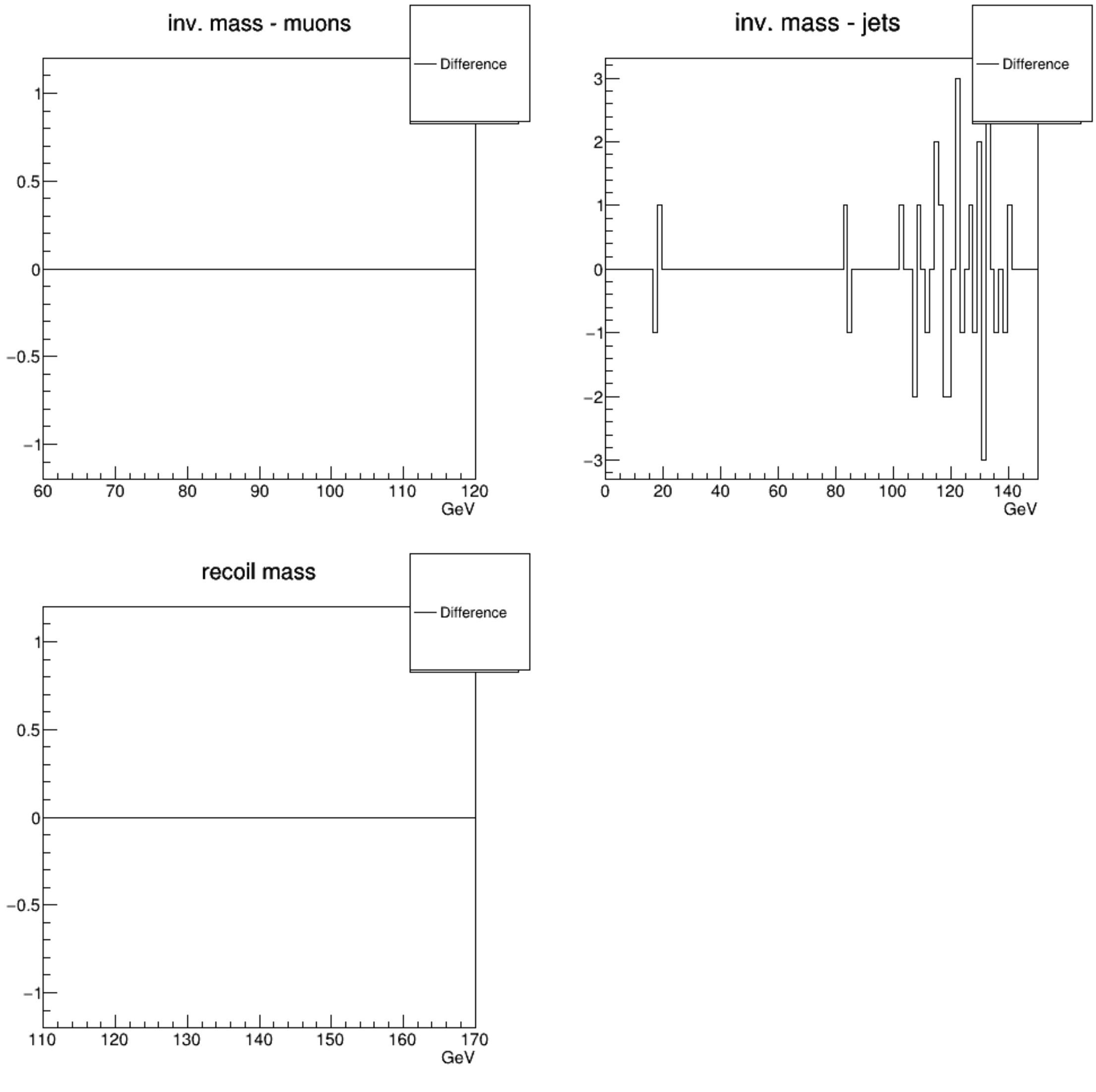

Figure 4: Discrepancies in the comparison plots for the invariant di-muon mass (a), invariant di-jet mass (b), and Higgs recoil mass (c) from two runs of a reconstruction in the Marlin experimental framework.

2.4 k4MarlinWrapper Comparison

Due to the results of the comparison between two reconstructions in the Marlin framework, it would now not be expected for two reconstructions using the k4MarlinWrapper in the Gaudi framework to be exactly reproducible. A comparison was also made between two k4MarlinWrapper reconstructions, in order to verify whether the inconsistencies between these two reconstructions are of the same size as they were using the Marlin experiment framework.

There were 7061 events read from the miniDST files for each reconstruction, and of these events, 6059 had two jets and two muons in the first reconstruction, and 6057 had these in the second. Here, a very small discrepancy in the number of events for the two reconstructions has arisen (2 events, 0.03%), but this is not present in the comparison for the two reconstructions in the Marlin framework. The original miniDST workflow was also used in this comparison, and the small difference in the number of events is likely related to the handling of the random seed in the wrapper.

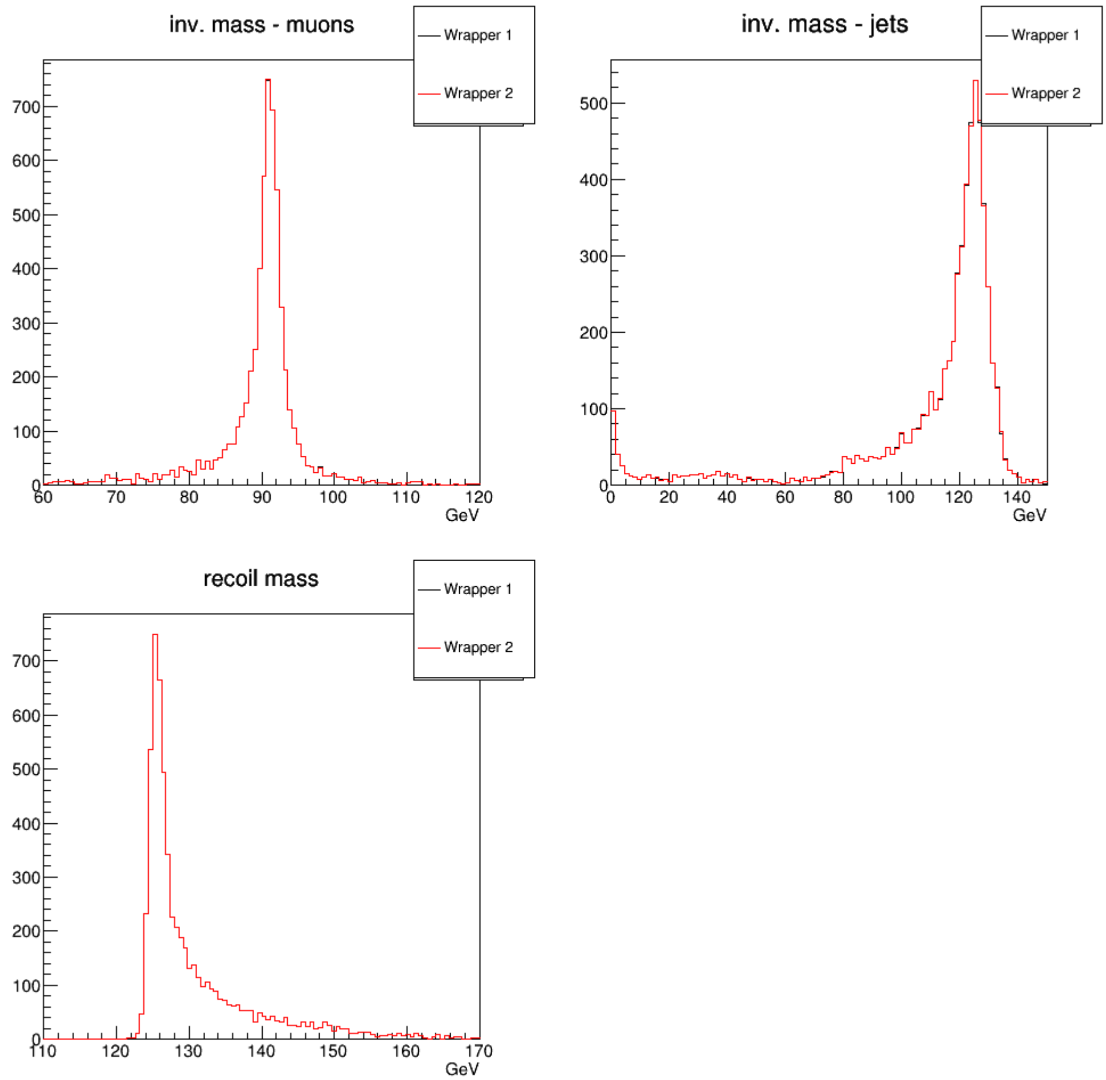

Figure 5: Comparison of invariant di-muon mass (a), invariant di-jet mass (b), and Higgs recoil mass (c) from two runs of a reconstruction using the k4MarlinWrapper in the Gaudi experimental framework.

Unlike for the reconstructions in the Marlin framework, Figures 5 and 6 show discrepancies in all three of the results. The differences in the results for the invariant di-muon mass (a) and the Higgs recoil mass (c) appear to be related to some small bin migration, or the slight difference in the number of events, but these are not present in the plots for the reconstructions in the Marlin framework. The discrepancies in the results for the di-jet mass, shown in Figure 6(b), are much large than those for the Higgs recoil mass and the invariant di-muon mass, but are only slight larger to that of the reconstructions in the Marlin framework.

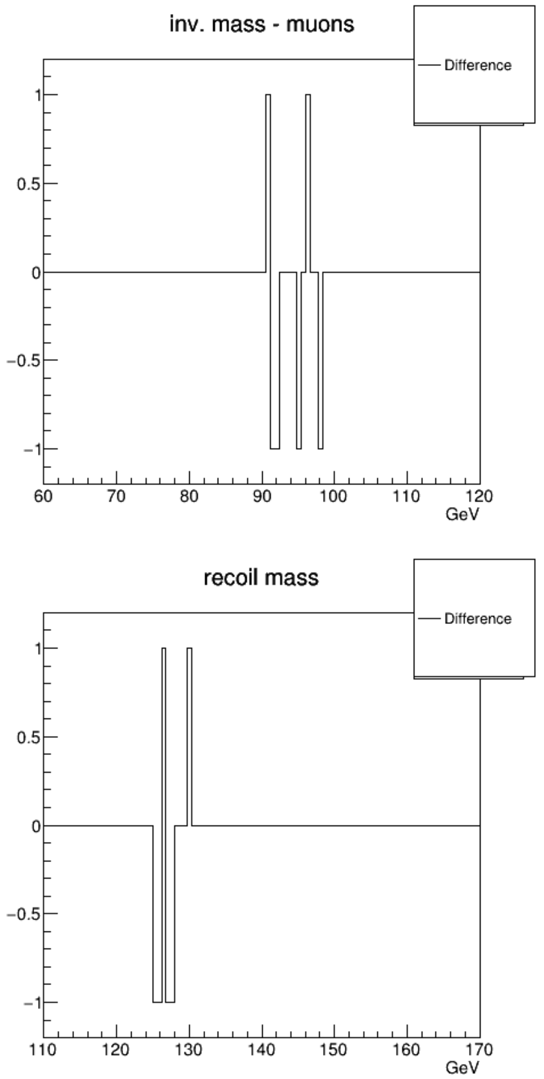

Figure 6: Discrepancies in the comparison plots for the invariant di-muon mass (a), invariant di-jet mass (b), and Higgs recoil mass (c) from two runs of a reconstruction using the k4MarlinWrapper in the Gaudi experimental framework.

2.5 EDM4hep Conversion

The EDM4hep conversion from LCIO was unsuccessful due to the conversion type not currently being supported. This meant that a comparison between these outputs was not possible, and so it remains a source of interest for when this conversion is made feasible.

3 Conclusions & Outlook

This work successfully ran the standard ILD reconstruction chain inside the Gaudi experiment framework for the first time, and without any major problems. There are still some smaller issues that need to be investigated, as producing statistically consistent running the processors in the Marlin framework and running them via the k4MarlinWrapper in the Gaudi framework was unachievable. The problem has potentially been traced to a problem with the handling of the random seed in the Marlin processors, as multiple reconstructions using the Marlin processors in the Marlin framework were not exactly reproducible, leading to the assumption that some of the Marlin processors may not be governed by the global random seed.

The miniDST workflow, that is used in both the reconstruction in the Marlin framework and the reconstruction in the Gaudi framework, is a major source in the loss of events; for the case where there is an empty collection, latter processors cannot handle this and so the event crashes. The crashes caused by these problems were handled slightly differently in the Marlin and Gaudi frameworks, the processor in the Marlin framework aborted and produced a broken file, whilst the Gaudi framework managed to write a proper file, although it is also without the later events. Here, the Key4hep environment has identified a previously undiscovered problem in one of the standard workflows for the ILD.

A further point of interest for future investigations into the k4MarlinWrapper could be around the random seed used in the wrapper, which is fixed and not easily changeable from the command line and this disrupts the versatility of the wrapper.

The EDM4hep conversion from the LCIO output of the k4MarlinWrapper was unsuccessful for this sample. The conversion has been successfully achieved by colleagues at CLIC on their workflows, so further investigation is needed to find whether the problem is in the converter's compatibility with the ILD workflows, or if the conversion could at least be attempted on a sample where there is a PrimaryVertex, as this was not present in this sample.

Acknowledgements

Firstly, I wish to thank my supervisor, Thomas Madlener, for his patience and support throughout this project. My time working on this project has been incredibly challenging, and rewarding thanks to him. The feedback I received from Frank Gaede was very helpful and aided greatly in my understanding. I would also like to thank the FTX(SFT) group for allowing me the opportunity to present my work frequently and receive feedback to help me complete the aims of this project; the insight and encouragement that I received were extremely helpful. Lastly, I would like to show my appreciation to the DESY organisation team for hosting this great opportunity and ensuring that everything ran smoothly.

References

[1] Gaede, F. 2006. Marlin and LCCD: Software tools for the ILC. Nucl. Instrum. Meth. A. 559, 177-180.

[2] Deutsches Elektronen-Synchrotron DESY. 2021. iLCSoft. [Online]. [06/09/2021]. Available from: https://ftx.desy.de/research/sft/ilcsoft/

[3] Deutsches Elektronen-Synchrotron DESY. 2021. Key4hep - Turnkey Software Stack. [Online]. [06/09/2021]. Available from: https://ftx.desy.de/research/sft/Key4hep/

[4] Barrand, G., et al. 2001. GAUDI - A software architecture and framework for building HEP data processing applications. Comput. Phys. Commun. 140, 45-55.

[5] Madlener, T. 2021. Key4hep and EDM4hep - The way forward. [Online]. [06/09/2021]. Available from: https://agenda.linearcollider.org/event/9199/contributions/48127/attachments/36784/57541/Key4hep_edm4hep_ild_meeting_040621.pdf

[6] Fernandez Declara, P., Sailer, A., Volkl, V., Petric, M. 2021. k4MarlinWrapper. [GitHub Repository]. Available from: https://github.com/Key4hep/k4MarlinWrapper

[7] Deutsches Elektronen-Synchrotron DESY. 2021. ILCILD. [Online]. [06/09/2021]. Available from: https://www.ilcild.org

[8] Gaude, F., et al. 2003. LCIO: A Persistency framework for linear collider simulation studies. eConf. C0303241, TUKT001.

[9] Fernandez Declara, P. 2021. k4MarlinWrapper. [GitHub Repository]. Available from: https://github.com/key4hep/k4MarlinWrapper/blob/master/doc/edmConverters.md

[10] Fernandez Declara, P., et al. 2021. Key4hep: Status and Plans. EPJ Web Conf. 251, 03025.