UNIVERSITEIT VAN AMSTERDAM

Higgs spin analysis in Collins-Soper frame using opening angles of different-flavour final state leptons

Bram Hoonhout
10201289

July 10, 2014

Supervisors: Koen Oussoren, Stan Bentvelsen (NIKHEF)

Abstract

The CS-frame is used to successfully discriminate between different spin-parity samples of Higgs bosons in the $gg \to H \to W^+W^- \to e\mu\nu\bar{\nu}$ channel. Boosting to the Higgs rest frame is made possible by an algorithm that reconstructs the P_z of the dineutrino system. Spin extraction is done by comparing angular distributions and invariant mass of the final state leptons. Using official analysis cuts and detector simulated MC sets, the spin could still be extracted from these distributions, but comparing these plots to background processes and ATLAS data does not allow for any conclusions to be drawn about the actual spin of the Higgs. This is because the ATLAS data is blinded and therefore has a completely different shape than the signal and background simulation.

Contents

1 Introduction

In the Standard Model (SM), the Higgs boson is an elementary particle that has a spin of 0 and an even (+) parity. In this research the $gg \rightarrow H \rightarrow W^+W^- \rightarrow e\mu\nu\bar{\nu}$ channel will be analysed where the opening angle between the two final state leptons contains information about the spin of their parental Higgs boson. This information can be examined by boosting the system to the Collins-Soper (CS) frame, a special kind of Higgs rest frame that provides fixed axes for spin measurement. With the Higgs being at rest, the CS-frame also ensures that the opening angles of final state leptons are independent of the original momentum of the Higgs boson. The quantities of interest that are related to spin are $\Delta\phi_{ll}$, the angle difference of the leptons in the transverse plane, and M_{ll} , the invariant mass of the two leptons. To boost to the Higgs boson's rest frame, its four-vector is constructed by summing the four-vectors of its decay products. However, since the P_z of the neutrinos cannot be measured, it will be reconstructed using an algorithm that makes use of an estimate for the invariant mass of the neutrinos.

This research consists of two analyses: a CS-frame test and a background study. The first one is to test whether the CS-frame can be used for spin analysis by examining the leptonic opening angles in the CS-frame for different spin-parity combinations and see if spin and parity information can be extracted, albeit comparatively. For this test, Monte Carlo datasets are to be used where events have been simulated with Higgs being a 0^+, 0^-, 1^+, 1^- or 2^+ particle. No cuts will be placed, as it is mainly an orientation procedure to test the possibilities of the CS-frame.

In the background study, spin-parity models will be tested by using signal and background MC datasets that have undergone detector sensitivity simulation and comparing them to actual ATLAS data. All these datasets will be cut using the cuts that have been developed by the ATLAS research group, so that the shape of the data can be compared to signal+background to see which spin-parity model fits the data best.

This thesis provides theoretical background needed for proper understanding in Sec. 2. In Sec. 3, the used data is discussed for each of the previously described analyses, followed by the analysis methods for the data in Sec. 4. The results will be shown and subsequently discussed in Sec. 5 and 6 respectively.

2 Theory

2.1 Higgs generation

At proton-proton (pp) collisions in the LHC, Higgs bosons can be produced through several channels. The two most prominent channels are gluon-gluon fusion (ggF) and vector-boson fusion (VBF). At ggF, two gluons of the colliding protons fuse into a Higgs boson through a top quark loop, while at VBF two quarks of the protons emit W/Z bosons that fuse into a Higgs boson. A good way to differentiate between these production modes is to look at the number of jets, as VBF results in at least 2 jets. Because of the significantly higher cross section of ggF compared to VBF, this research focuses solely on the former.

2.2 Higgs decay

In table 1, one can find the branching ratios (*branchios*) for the decay for several processes. In this research, the channel $H \rightarrow W^+W^- \rightarrow e\mu\nu\bar{\nu}$ has been chosen, because of the decent branchio, accurate measurability of two final state different-flavour leptons (e, μ) and the absence of jets. The fact that the leptons differ in flavour provides a clear distinction from a significant part of background events that arise from γ/Z-bosons decaying into l^+l^- pairs. A downside of this channel is that it has two final state neutrinos, the both of which will not be detected. This means one cannot directly boost to the Higgs rest frame due to lack of information, so a special algorithm will be used to reconstruct P_z of the neutrinos based on an estimate of the dineutrino mass ($M_{\nu\nu}$), as discussed in 2.5.3. Combining the picked generation mode and decay mode, the complete channel will be (see Fig. 1):

$$gg \rightarrow H \rightarrow W^+W^- \rightarrow e\mu\nu\bar{\nu}$$

Decay Mode	Branchio
$H \rightarrow b\bar{b}$	58.8 %
$H \rightarrow W^+W^-$	21.6 %
$H \rightarrow gg$	8.6 %
$H \rightarrow \tau^+\tau^-$	6.4 %
$H \rightarrow c\,\bar{c}$	2.9 %
$H \rightarrow ZZ$	2.7 %
$H \rightarrow \gamma\gamma$	0.2 %
$W^+ \rightarrow e^+\nu$	10.75 %
$W^+ \rightarrow \mu^+\nu$	10.57 %
$W^+ \rightarrow \tau^+\nu$	11.25 %
$W^+W^- \rightarrow e\mu\nu\nu$	2.27 %

Table 1: Branchios of interest for Higgs bosons and W bosons. [1][2]

2.3 Initial Higgs Pt

While the total transverse momentum (P_T) of each *proton* is initially 0, the *partons* do not necessarily have a Pt of 0 as well – as long as the P_T of all partons sum up to 0. Therefore the P_T of each colliding parton can be described with some distribution function that has to be symmetric around 0, assuming the P_T distribution of the partons within the proton has no preferred direction. When the partons fuse into a Higgs boson their P_T will carry over, resulting in initial state Higgs bosons that do not necessarily have a P_T of 0. However, this contribution to Higgs P_T is rather small, with an order of magnitude of about 1-10 GeV. Another source for initial Higgs P_T is a “gluon kick” during ggF: before the gluons collide, one of them emits a third gluon which possibly has high P_T . Each time this happens a jet is created and the P_T of the Higgs will change, potentially significantly.

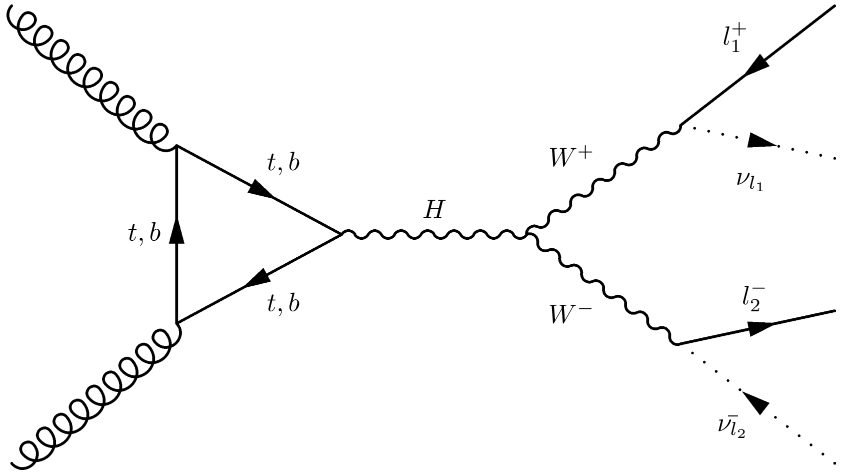


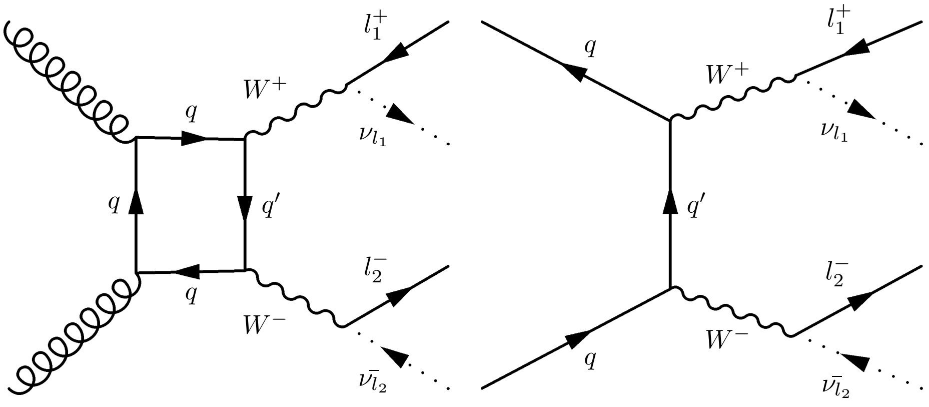
Figure 1: The channel analysed in this research, $gg \to H \to W^+W^- \to e\mu\nu\bar{\nu}$. Note that $l_1, l_2 \in \{e, \mu\}$; $l_1 \neq l_2$

2.4 Background processes

The background process that is toughest to differentiate from the signal is known as “WW”, with corresponding formula $gg/q\bar{q} \to Z/\gamma \to W^+W^- \to e\mu\nu\nu$, and is shown in Fig. 2a and 2b. After analysis cuts have been applied, this channel yields about 60% of all expected background events (See App. B). Another process that will be treated in this research is “$t\bar{t}$”, which has a significantly smaller contribution with circa 4% of the expected background: $gg/q\bar{q} \to g \to t\bar{t} \to W^+W^-b\bar{b} \to e\mu\nu\bar{\nu}b\bar{b}$ (see Fig. 3). This produces two b quarks, each of which will decay into a jet, whereas the signal process does not necessarily result in any jet. Hence, selecting only events with zero jets would be a good requirement in order to reject nearly all $t\bar{t}$ background events. Other background processes will not be analysed in this research, because of the limited scope of this research and availability of datasets.

2.5 Kinematics

In the $gg \to H \to W^+W^- \to e\mu\nu\bar{\nu}$ channel (Fig. 1), the Higgs spin can be analysed by examining the distributions of observables related to the opening angles between the final state leptons. However, due to initial state radiation



(a) $gg \rightarrow W^+W^- \rightarrow e\mu\nu\nu$

(b) $q\bar{q} \rightarrow W^+W^- \rightarrow e\mu\nu\nu$

Figure 2: WW background processes, again: $l_1, l_2 \in \{e, \mu\}$; $l_1 \neq l_2$

and inherent parton P_T , the Higgs boson potentially has initial P_T (discussed in Sec. 2.3), for which opening angle variables are very sensitive. For this reason the system needs to be boosted to the Higgs rest frame, allowing correct measurements of these quantities. Therefore, the Collins-Soper frame is used (Sec. 2.5.2), which not only boosts to the Higgs rest frame but also conserves spin and parity information.

2.5.1 Lab frame

The lab frame has its origin at the central collision point of the ATLAS detector, with the z-axis defined along the beam momentum, the y-axis pointing upwards and the x-axis pointing horizontally inwards to the centre of the LHC. Because of the radial symmetry along the z-axis, particle kinematics are often represented using angular coordinates, with θ being the angle between the particle's direction and the positive z-axis, and ϕ the angle in the XY-plane. Sometimes the pseudorapidity η is used instead or θ, defined as:

$$\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$$

Generally, the advantage of using η over θ is that differences in pseudorapidity are independent of Lorentz boosts along the z-axis.

2.5.2 Collins-Soper frame (CS-frame)

The CS-frame is constructed by boosting to the Higgs boson's rest frame, and then rotating in such a way that measured variables remain sensitive to spin and CP properties of the Higgs boson[3]. This is done by defining redefining the z-axis as the inner bisector of P_{g_1} and $-P_{g_2}$, with g_i being the gluons that

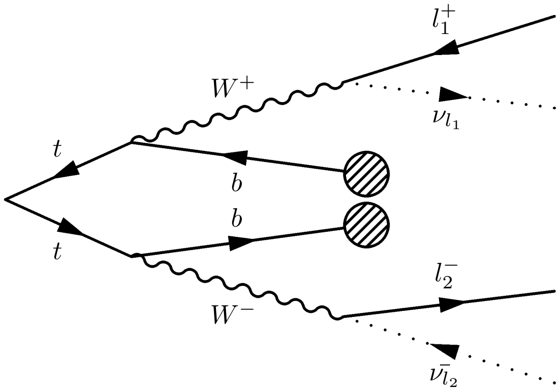


Figure 3: Second part of the $t\bar{t}$background process: $(gg/q\bar{q} \to g \to)\ t\bar{t} \to W^+W^-b\bar{b} \to e\mu\nu\bar{\nu}b\bar{b}$, with the omitted part enclosed in parentheses, b quarks resulting in jets are denoted by blobs.

fuse into the Higgs boson. The x-axis becomes the outer bisector, ie the original Higgs momentum vector, and the y-axis will be set perpendicular to both other axes. As no information on the gluons is available, the actual boosting will be performed by multiplying the to-be-boosted four-vector by a 4×4 matrix that is created using information of both final state leptons and the dineutrino system. Eq. 1 shows the matrix that is used to boost, taken from [4]. In this matrix Q is the four-vector of the Higgs boson, and $X_T = \sqrt{Q^2 + p_T^2}$.

Opening angles between decay products are very sensitive to the any momentum of the parent particle, and with the Higgs boson now being at rest, $\Delta\phi_{ll}$ will now be independent of (initial) Higgs P_T .

$$\begin{pmatrix} P_x \\ P_y \\ P_z \\ E \end{pmatrix}_{CS} = \begin{pmatrix} -\frac{P_T}{\sqrt{Q^2}} & 0 & -\frac{Q_3}{\sqrt{Q^2}} & \frac{Q_0}{\sqrt{Q^2}} \\ \frac{X_T}{\sqrt{Q^2}} & 0 & \frac{P_T Q_0}{\sqrt{Q^2} X_T} & -\frac{P_T Q_0}{\sqrt{Q^2} X_T} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{Q_0}{X_T} & -\frac{Q_3}{X_T} \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \\ P_z \\ E \end{pmatrix}_{Lab} \tag{1}$$

2.5.3 Dineutrino P_z reconstruction

In order to properly boost to the CS-frame, the Higgs needs to be fully reconstructed by summing the four-vectors of its (final) decay products:

$$P_H = P_{l_1} + P_{l_2} + P_{\nu\nu}$$

with $\nu\nu$ representing the dineutrino system. Evaluating this equation cannot be done as of yet because only the transverse momentum is known for the dineutrino system by making use of the E_T^{miss} quantity. E_T^{miss} is calculated by summing the negative momenta of every measured particle. leaving a “missing” energy in a certain direction due to not all particles being measured, such as the neutrinos. P_z will be calculated by using an algorithm that solves the quadratic equation that arises when trying to calculate the square of the above equation. The resulting equation has only two unknown quantities, $M_{\nu\nu}$ and $P_z^{\nu\nu}$, and will return $P_z^{\nu\nu}$ after using estimate for $M_{\nu\nu}$. The value used for $M_{\nu\nu}$ will be 30 GeV throughout this thesis; for an explanation on the details of the algorithm and a study of reconstruction accuracy as a function of this $M_{\nu\nu}$ value, see [5].

2.6 Helicity and chirality

The helicity of a particle is the projection of its spin onto the particle's momentum vector. The resulting vector points either along with the momentum vector, called "right-handed", or opposite ("left-handed"). Chirality is whether a particle is left-handed or right-handed, which, for massive particles, is dependent on the frame of reference as one can pick a frame where the particle will be overtaken, causing it to reverse chirality. Massless particles move with the speed of light and therefore cannot be overtaken in any frame, resulting in that a particle's chirality will always be equal to its helicity.

Currently, only left-handed neutrinos have been observed during weak interactions, which means that the spin of weakly interacting neutrinos tends to point opposite to the direction in which they are going. Antineutrinos therefore must be right-handed, so that their spins point along their momentum vectors.

2.7 Spin models

The spin of the Higgs boson will be deduced by examining the distributions of $\Delta\phi_{ll}$ (the difference in ϕ between the two leptons, in the range of $[0, \pi]$), and M_{ll} (mass of the dilepton system) in the CS-frame. The reason these distributions are sensitive to Higgs spin is that when a Higgs boson decays, its child particles (W^+W^-) together must have the same spin as the Higgs. When considering the Higgs rest frame, the W bosons will be emitted back-to-back ($P^{W^+} = -P^{W^-}$), so for a spin 0 Higgs this means that the W bosons (spin 1 particles) have opposite spins. The bosons then decay into $l\nu_l$ pairs: $(W^+)(W^-) \to (l_1^+\nu_{l_1})(l_2^-\bar{\nu_{l_2}})$, with $l_1, l_2 \in \{e, \mu\}$; $l_1 \neq l_2$. Because of the chirality rules described in Sec. 2.6, the antineutrino that originates from the W^- tends to move in the direction of the spin of the W^-, while the neutrino from the W^+ moves opposite to the projected spin of the W^+. Combining these findings with the fact that the W bosons have opposite spins, this results in a tendency for neutrinos to move in the *same* direction, ie have a small opening angle. The leptons follow this behaviour, albeit in the opposite direction (due to conservation of momentum), and therefore prefer small opening angles as well.

2.7.1 $\Delta\phi_{ll}$ shape

Deductions similar to the one done for spin 0 in 2.7 can be made for models with higher spin by requiring that the spins of the W bosons add up to 1 or 2.

Due to the extensiveness of the procedure, only the results will be featured in this thesis, which can be summarised as follows:

Spin 0:
$\Delta\phi_{ll}$ tends to be small

Spin 1:
$\Delta\phi_{ll}$ shows no preference, meaning a flat distribution is expected

Spin 2:
$\Delta\phi_{ll}$ prefers to be large (towards the maximum, π)

2.7.2 M_{ll} shape

Making use of the expected $\Delta\phi_{ll}$ shapes (Sec.2.7.1), one can make predictions about the shape of the dilepton mass (M_{ll}) distribution as well. Calculating the invariant mass of two arbitrary particles 1 and 2, the following equation can be derived:

$$M_{1,2}^2 = P_\nu^1 P_2^\nu = E_1 E_2 - P_i^1 P_2^i = E_1 E_2 - |\vec{P_1}||\vec{P_2}|cos(\theta)$$

with θ the angle between the momenta of the particles. This equation shows that the invariant mass is at its maximum and minimum when opening angle θ is equal to 0 and π respectively. The average opening angle thus determines the average invariant mass, so that the position of the peak of M_{ll} distributions can be estimated for each spin model relative to other spin models: M_{ll} of the spin 1 model should peak at a higher value than at the model with spin 0, and the spin 2 M_{ll} is supposed to peak at an even higher value.

2.7.3 Parity: P_{ll}^{dif}

The parity operation involves flipping all axes of a system. The parity of a particle is said to be positive (+) or negative (-) if it is or is not invariant under parity transformation, respectively. Besides spin, parity of the Higgs boson can be analysed in the CS-frame as well. The quantity especially sensitive to parity is $P_{ll}^{\text{dif}} \equiv \Big| |\vec{P_{l_1}}| - |\vec{P_{l_2}}| \Big|$, the difference of the magnitude the lepton momenta. The exact reason why this observable is sensitive to parity is beyond the scope of this thesis, therefore only visible differences in the shapes of the distributions will be analysed qualitatively.

3 Data acquisition

For the first CS-frame test, Monte Carlo (MC) datasets of about 200,000 events have been generated with Higgs spins of 0^+, 0^-, 1^+, 1^- and 2^+, using generators MadGraph_aMC@NLO. These will be used primarily to try and see if the CS-frame can be properly reconstructed and whether spin and CP information can be extracted as discussed in Sec. 2.7. Information on these MC sets can be found in Table 2a. These datasets contain truth-level data for the neutrinos as well as regular E_T^{miss} , making them very useful to check whether the neutrino reconstruction algorithm (Sec. 2.5.3) is working correctly.

The background study requires "official" signal and background datasets, which have undergone detector sensitivity simulation. These sets and their generators can be found in Table 2b. A small dataset of ATLAS data was provided for use in this research. Uncut, this set contains about 51,000 different-flavour events.

Dataset	Events
Sig 0^+	200,000
Sig 0^-	200,000
Sig 1^+	200,000
Sig 1^-	200,000
Sig 2^+	200,000

(a) MC datasets used in the first analysis, generated with MadGraph_aMC@NLO

Dataset	Generator	Events
Sig 0^+	JHU+Pythia8	132,529
Sig 0^-	JHU+Pythia8	139,446
Sig 2^+	JHU+Pythia8	128,947
WW (gg)	gg2WW3.1.2	13,516
WW ($q\bar{q}$)	Powheg+Pythia6	389,120
WW (tot)	—	402,636
$t\bar{t}$	Alpgen+Herwig	253,736

(b) Official MC datasets

Table 2: MC datasets that have been used, with the (uncut) number of different-flavour events indicated.

3.1 Event selection

No cuts have been placed on the test run datasets, as they were merely used for exploration and only contained signal. For the official datasets, different sets of cuts have been composed in accordance to the official analysis (see App. B). A subset of these cuts can be found in Table 3, both of which will be applied on the official MC and ATLAS datasets.

Preselection cuts	0-jet cuts
$E_T^{miss} > 20$ GeV	$n_{jets} = 0$
$P_T^{lep_{lead}} > 22$ Gev	$P_T^{ll} > 20$ GeV
$P_T^{lep_{sub}} > 15$ Gev	$M_{ll} < 80$ GeV
$\lvert\eta^{lep_{lead}}\rvert < 2.5$ Gev	$\Delta\phi_{ll} < 2.8$
$\lvert\eta^{lep_{sub}}\rvert < 2.5$ Gev	$0.5 \cdot m_H \leq m_T \leq 1.2 \cdot m_H$

Table 3: Official analysis cuts, as used on the official MC datasets and set of ATLAS data.

In this table, "lep_{lead}" means leading lepton, ie the lepton with highest energy. Analogously, "lep_{sub}" is the subleading lepton, with second highest energy.

4 Data analysis methods

The data is analysed using ROOT in combination with C++, where an extensive custom class (see App. C) is used to iterate over multiple files as a batch process. Two main analyses have been done in this research, the first one focusing solely on signal to check whether spin/parity information can be retrieved, and one that takes background processes and experimental data into consideration to try and see which spin model fits the data best.

4.1 CS-frame test

The first analysis uses the test run datasets (signal only) and should answer the question whether the CS-frame can be constructed properly and whether spin or CP information can be deducted from the resulting plots. The plots will also be used to get a feeling for the shapes of distributions of a wide variety of kinematic quantities (P_T , P_z , ϕ, θ, etc.), as well as special quantities ($\Delta\phi_{ll}$, M_{ll} , P_{ll}^{dif}).

For each event of a dataset, the leptons will be reconstructed using the momentum and energy that is be found within the data file. After that, the dilepton system can be created instantly, simply by summing the two four-vectors of the leptons. Then, two different four-vectors for the dineutrino system can be created: one "true" and one "recon" (reconstructed), using neutrino truth information and the dineutrino reconstruction algorithm respectively. Correspondingly, two Higgs boson four-vectors can be made by summing the leptons with either the true or the reconstructed dineutrino system. Then, using the custom class, every quantity of interest will be calculated for each reconstructed particle, in every desired frame, and all results will be added to their corresponding histograms. After looping over every entry and every data file, the histograms are ready to be interpreted. The true Higgs boson can be compared to the recon Higgs boson to test how well the reconstuction algorithm works.

4.2 Comparison to ATLAS data

At the second analysis, the detector simulated MC datasets will be used, along with the set of actual ATLAS data. These sets do not contain truth information about the neutrinos, and therefore only the E_T^{miss} can be used to construct the initial dineutrino system. From this point, the steps will be the same as in Sec. 4.1, with the exception that both sets of cuts as found in Table 3 will be applied: events that do not comply with these requirements will be rejected and therefore not be added to the histograms. As the datasets contain a different number of events, the ratio between the integrals of each set is quite random, which will be fixed by scaling background and signal histograms so that the full integral of any histogram for a specific channel will match the number given in an official cutflow diagram (App. B). Essentially, this means that for every channel its histograms will be divided by the number of passed events, resulting in histograms that have cuts applied as well as an integral of 1. Then, the histograms are multiplied by the correct number in the cutflow diagram. Of course, the data will not be scaled in any way.

Finally, signal is stacked onto the background for each spin model and the data is plotted over the resulting histogram. Since only a part ($\sim 64\%$) of the background processes have been analysed, the data is expected to lie above the background stack at every point, and the background stack should approach the data more every time a different background process is added to the analysis. When enough of the expected background is analysed, one of the signal models could be stacked on top of the background to see which model fits the remaining gap between background and data best. In this research, just two background processes are analysed which would allow only a qualitative conclusion about the spin of the Higgs.

5 Results

The results of the CS-frame test are shown in Fig. 4 to 6. All plots are displayed in the recon CS-frame, with angular units in radians and all other units in GeV. The y-axis of the plots show particle counts, which do not carry a specific meaning as only the shape is examined.

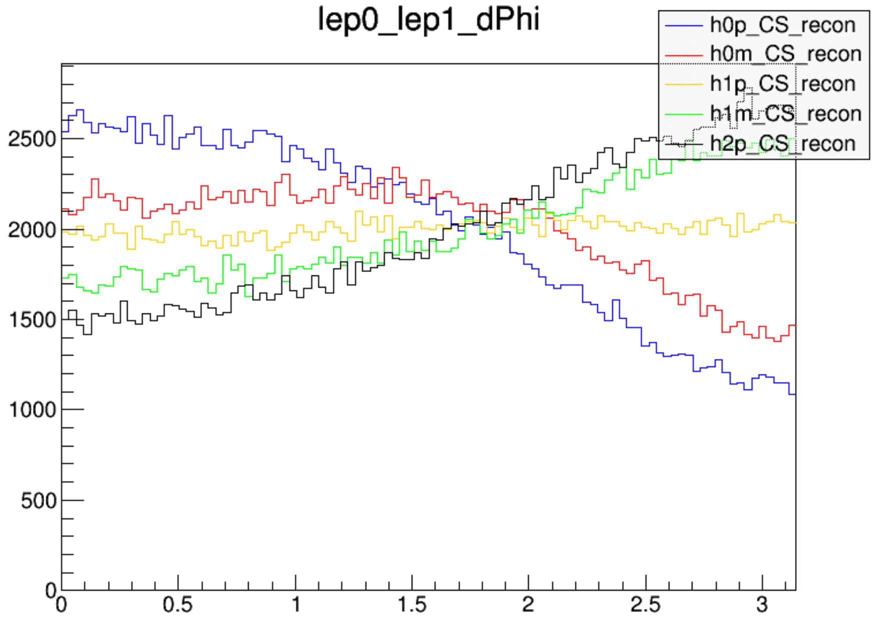


Figure 4: $\Delta\phi_{ll}$ distribution of the CS-frame test.

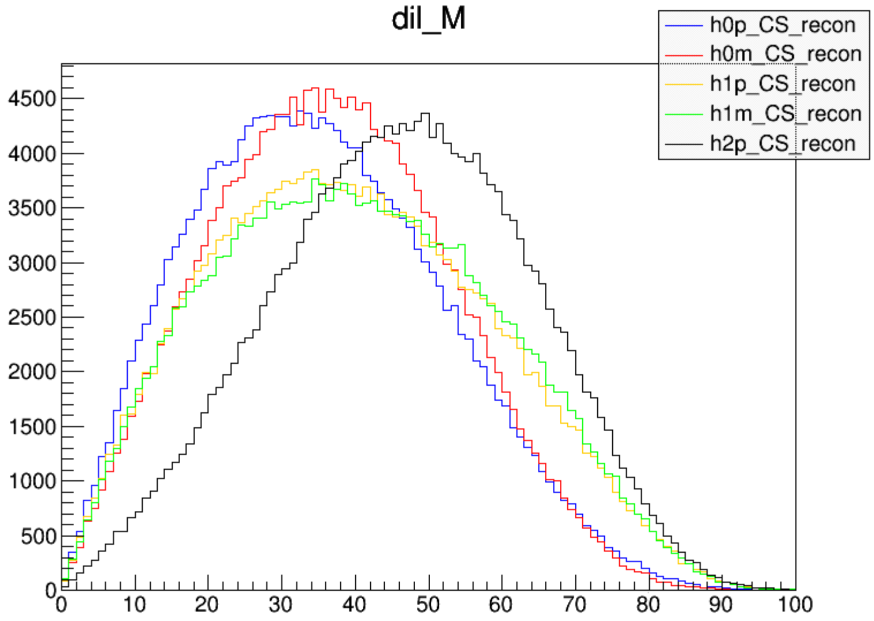


Figure 5: M_{ll} distribution of the CS-frame test.

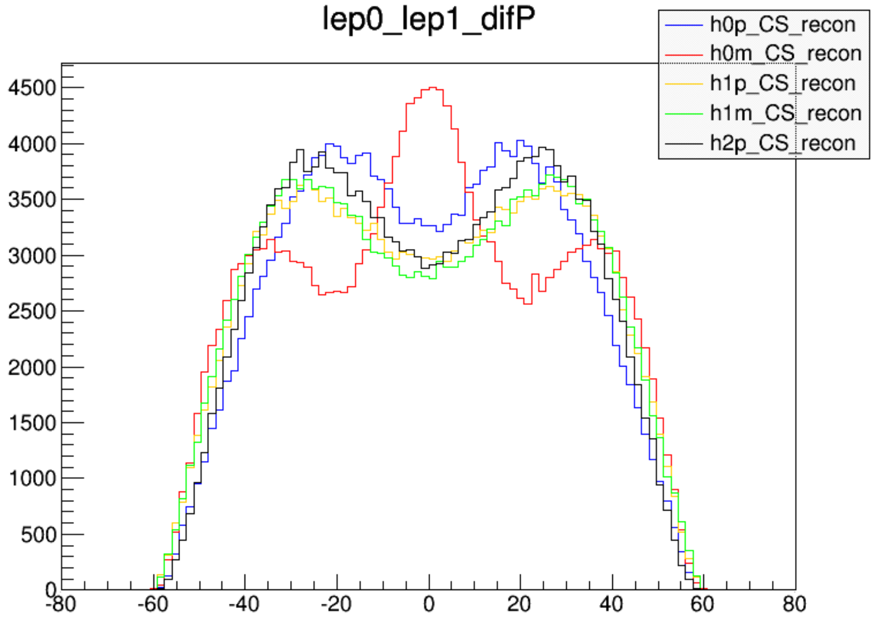


Figure 6: P_{ll}^{dif} distribution of the CS-frame test.

The signal distributions that were created during the background study can be found in Fig. 7 to 9. These histograms were scaled vertically in order to fit the expected signal according to the cutflow diagram (App. B).

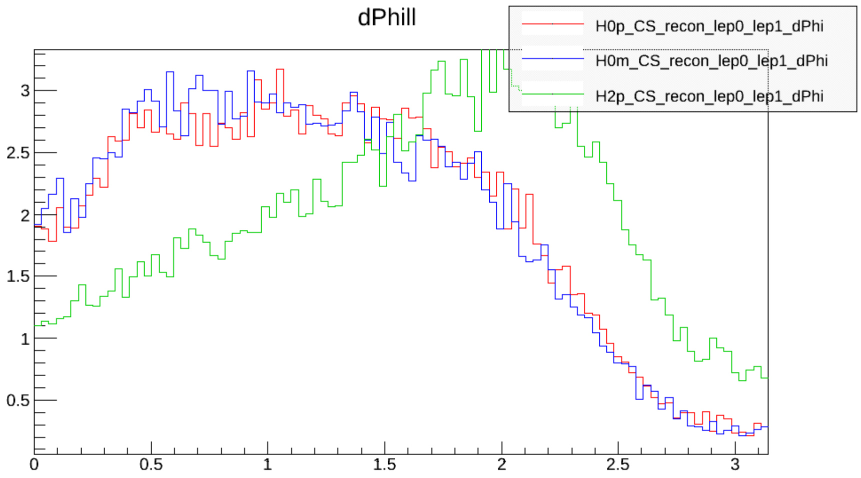


Figure 7: $\Delta\phi_{ll}$ distribution of the signal as part of the background study.

The stacked histograms, with signal, background and data plotted at once, are shown in Fig. 10 to 13. The signal and background processes are scaled so that the integral of each process matches their corresponding value of the cutflow diagram (at the row of the 0jet section saying $0.5 \cdot m_H \leq m_T \leq 1.2 \cdot m_H$, App. B). Data is not scaled in any way.

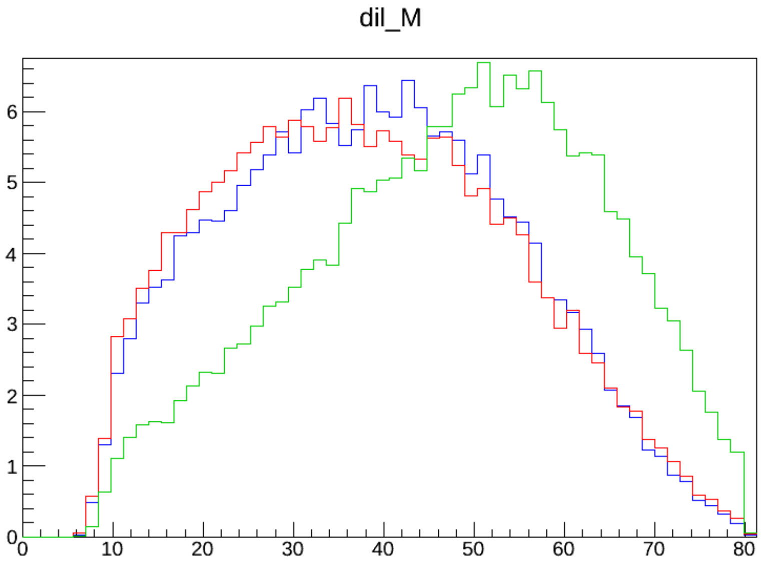


Figure 8: M_{ll} distribution of the signal as part of the background study. It uses the same legend as Fig. 9.

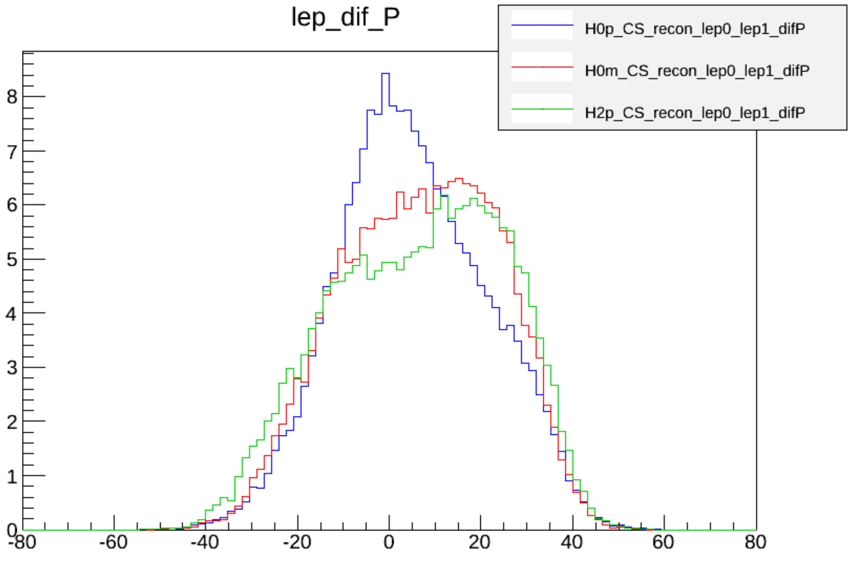


Figure 9: $P_{ll}^{\rm dif}$ distribution of the signal as part of the background study.

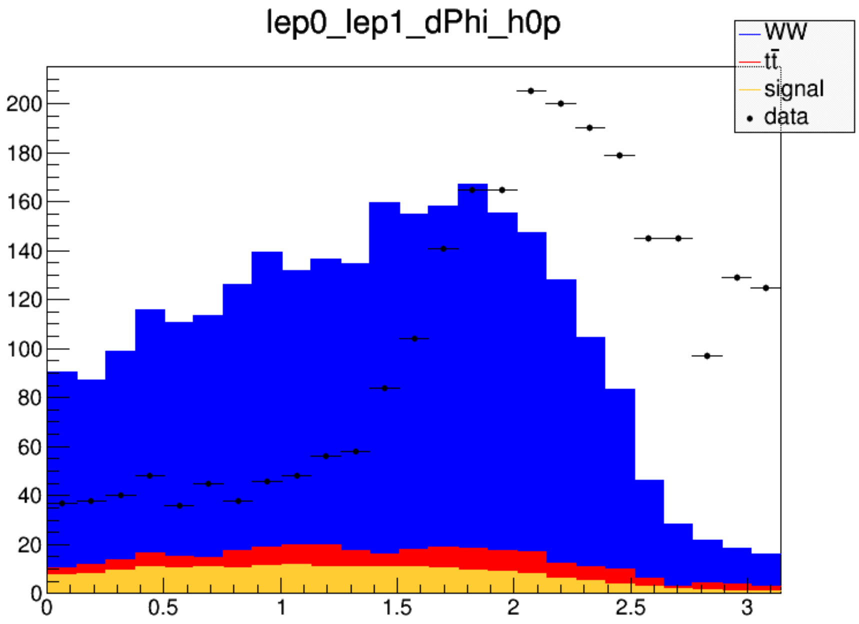


Figure 10: Stacked $\Delta\phi_{ll}$ distribution with 0^+ as signal.

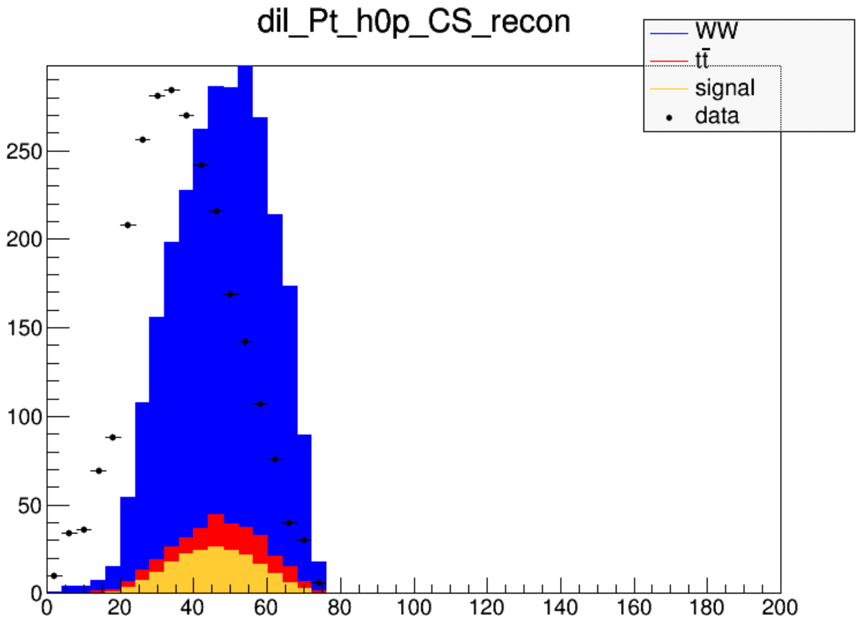


Figure 11: Stacked P_T distribution with 0^+ as signal.

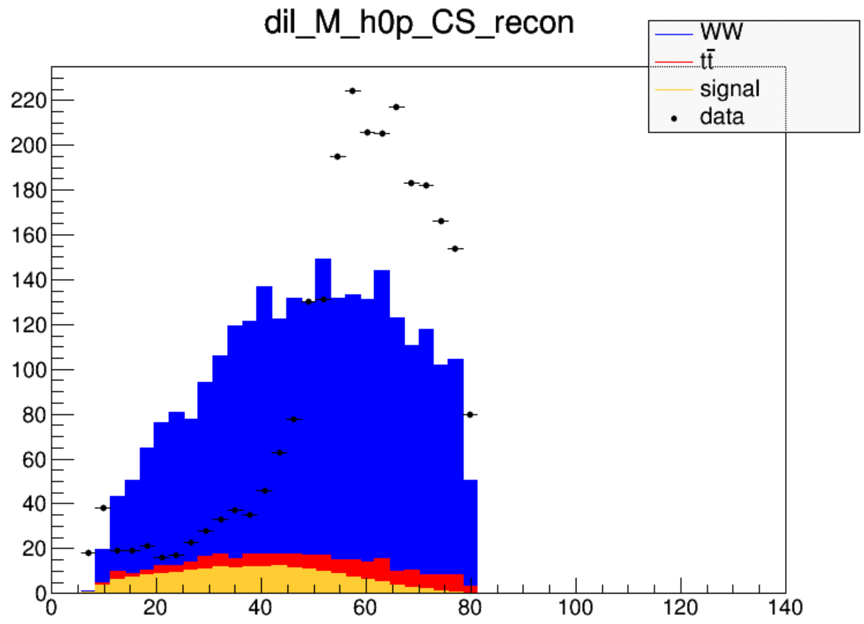


Figure 12: Stacked M_{ll} distribution with 0^+ as signal.

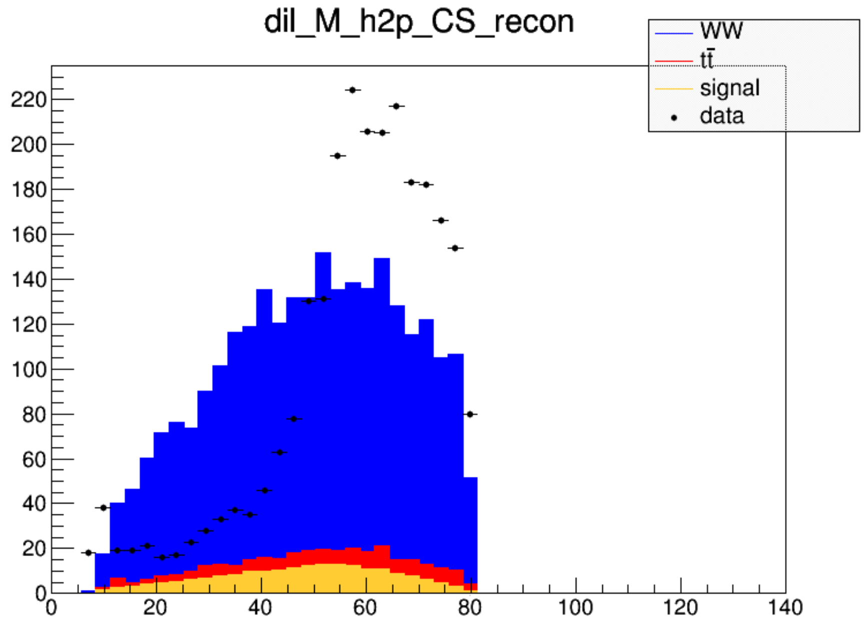


Figure 13: Stacked M_{ll} distribution with 2^+ as signal.

6 Conclusion and discussion

The $\Delta\phi_{ll}$ distribution in Fig. 4 clearly confirms the that the CS-frame is working as the shapes are exactly as predicted in Sec. 2.7.1, with spin 0 models favouring low opening angles, spin 1 models returning a flat distribution and spin 2 models preferring large opening angles. M_{ll} plots in Fig. 5 also show peaks corresponding to expectation, allowing spin to be deducted comparatively. However, it begs the question whether a single value for $M_{\nu\nu}$ allows for accurate boosting for every spin model, as $M_{\nu\nu}$ is probably just as sensitive to spin as M_{ll}. Moreover, the value that is used for $M_{\nu\nu}$ is based on an *average* taken during 0 spin simulation, which might result in other spin models being reconstructed with less accuracy. Though, $\Delta\phi_{ll}$ is less sensitive to reconstruction faults as it measures the angle in the transverse plane while the reconstruction only has an effect on the boost in the z-direction, and M_{ll} is invariant under any boost anyway. Further research must show if separate $M_{\nu\nu}$ values must be used for each spin model. More information and a precursory research on the effect of varying $M_{\nu\nu}$ can be found in [5]. Furthermore, P_{ll}^{dif} plots show a clear distinction between the 0^+ and 0^- spin models (with positive or negative parity, respectively), whereas other spin/P models do not demonstrate major differences.

The CS-frame $\phi_{l_1} : \phi_{l_2}$ scatter plots give more insight in how the distribution of $\Delta\phi_{ll}$ arises as the ϕ's of the leptons are plotted against each other. See App. A for the scatter plots and more information about them.

Judging from the $\Delta\phi_{ll}$ histograms that are part of the background study, Fig. 7 to 9, different spin models can still be differentiated when using the processed MC signal sets, as the 0 spin models have a downward slope, and the spin 2 model shows a rising slope. Be aware that the cut $\Delta\phi_{ll} < 2.8$ has been applied, which is why the $\Delta\phi_{ll}$ of the 2^+ starts a significant drop around that threshold. The cut is applied on lab variables, which is why there is no hard cut-off visible at $\Delta\phi_{ll} = 2.8$. The M_{ll} histograms show that the spin 2 model peaks at higher energies than the spin 0 models, conform expectation. The requirement $M_{ll} < 80$ GeV can be seen in Fig. 12 and 13 in the CS-frame as a hard cut, because M_{ll} is Lorentz invariant.

Information about parity, which should emanate from differences in shape of the plots of Fig. 9, cannot be extracted from these MC samples in the same way as the CS-frame test analysis, since the 0^- histogram is not the only one that distinguishes itself enough from the others. If spin 1 samples become available, they could be added to the analysis for further research.

In Fig. 10 to 13, the plots show that the signal+background simulation stack does not seem to have a shape similar to the ATLAS data. This is because the data has been blinded, which means a different set of cuts has been applied in order to cut a lot of signal away. Blinding is done with the intention of forcing physicists to fully understand the background before trying to examine signal, so that no bias towards signal can arise. As of now, no conclusion can be drawn about the spin of the Higgs particle. The expectation is that when the full dataset will be released, the data would fit the simulation significantly better.

References

[1] Beringer, J. and others. Review of Particle Physics (RPP). *Phys.Rev.*, D86:010001, 2012.

[2] Mark Thompson. *Modern Particle Physics.* 2013.

[3] John C. Collins and Davison E. Soper. Angular distribution of dileptons in high-energy hadron collisions. *Phys. Rev. D*, 16:2219–2225, Oct 1977.

[4] Mirkes, E. and Ohnemus, J. W and Z polarization effects in hadronic collisions. *Phys. Rev. D*, 50:5692–5703, Nov 1994.

[5] David Hendriks. Analyzing Higgs in the Collins-Soper frame using neutrino reconstructions.

A Scatter plots

The ϕ scatter plots for the leptons of the 0^+ dataset (during CS-frame test analysis) have been shown in Fig. 14 to 16, all in the reconstructed CS-frame. The preference for $\Delta\phi_{ll}$ manifests itself in $\phi_{l_1} : \phi_{l_2}$ plots as the difference between the x and y value on the diagonal "bands" where particles are most frequent. The x-axis belongs to the electron, and the y-axis to the muon.

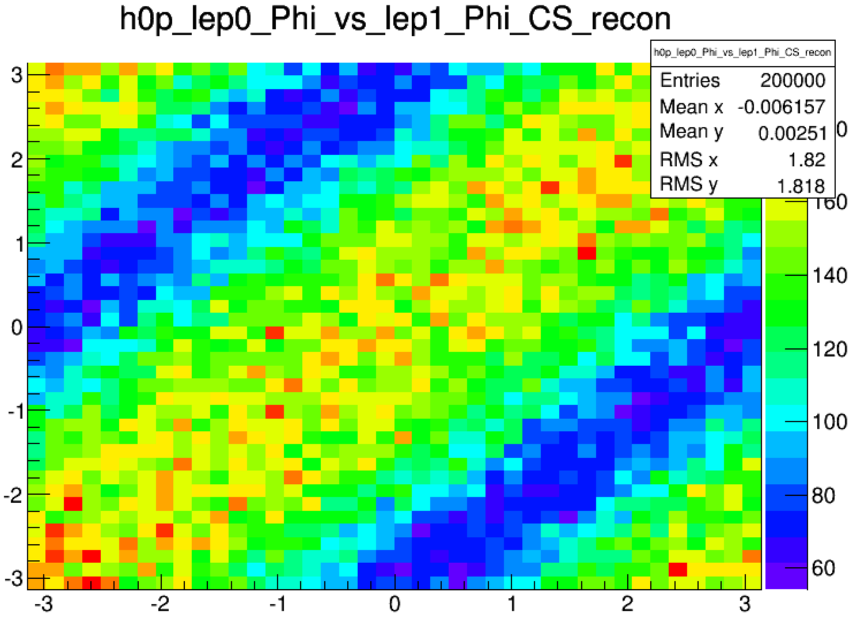


Figure 14: $\phi_{l_1} : \phi_{l_2}$ for 0^+ signal

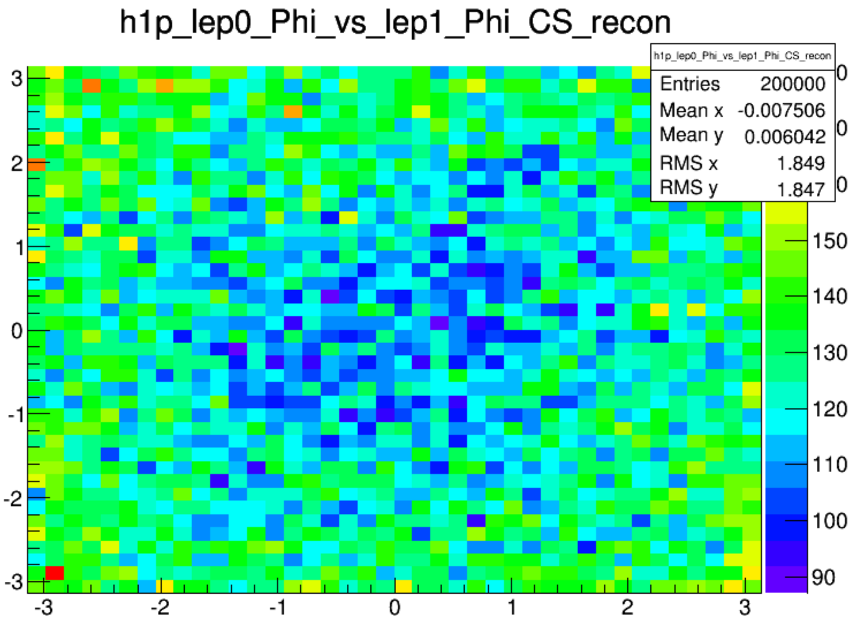


Figure 15: $\phi_{l_1} : \phi_{l_2}$ for 1^+ signal

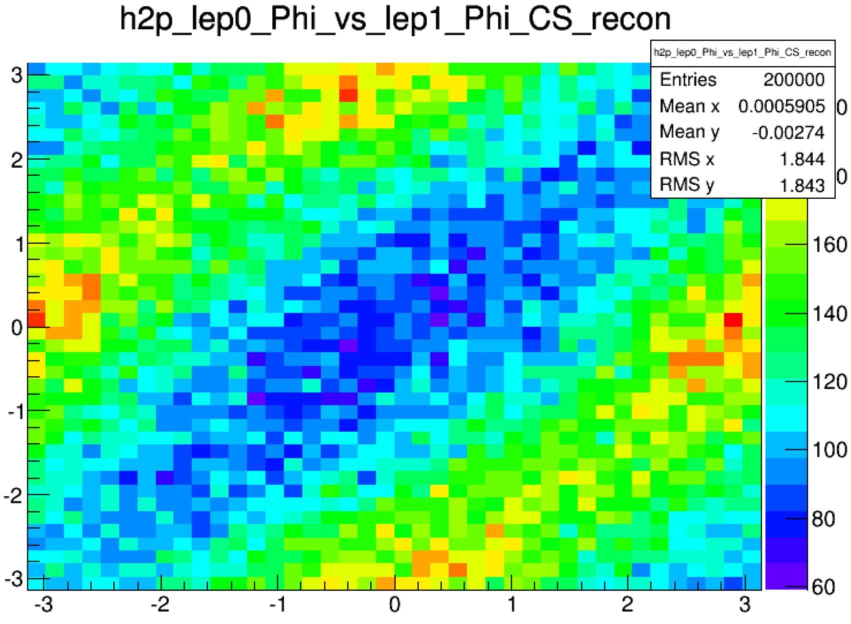


Figure 16: $\phi_{l_1} : \phi_{l_2}$ for 2^+ signal

Scatter plots for θ are shown in Fig. 17 through 19, the first in the lab frame, and the other two were boosted to the CS-frame using truth level neutrinos or reconstructed neutrinos. The first thing to notice is that in the lab frame, the leptons move very close to the z-axis, due to the immense P_z of the Higgs boson. This P_z arises from the imbalance of momentum carried by the colliding gluons when the Higgs is created. After boosting to the Higgs rest frame, the particles tend to move more in the transverse direction. However, the reconstruction has a strange effect on the $\theta_{l_1} : \theta_{l_2}$ distribution, seemingly pushing particles away from the $1.5\pi : 1.5\pi$ region. Further research is required to investigate this behaviour.

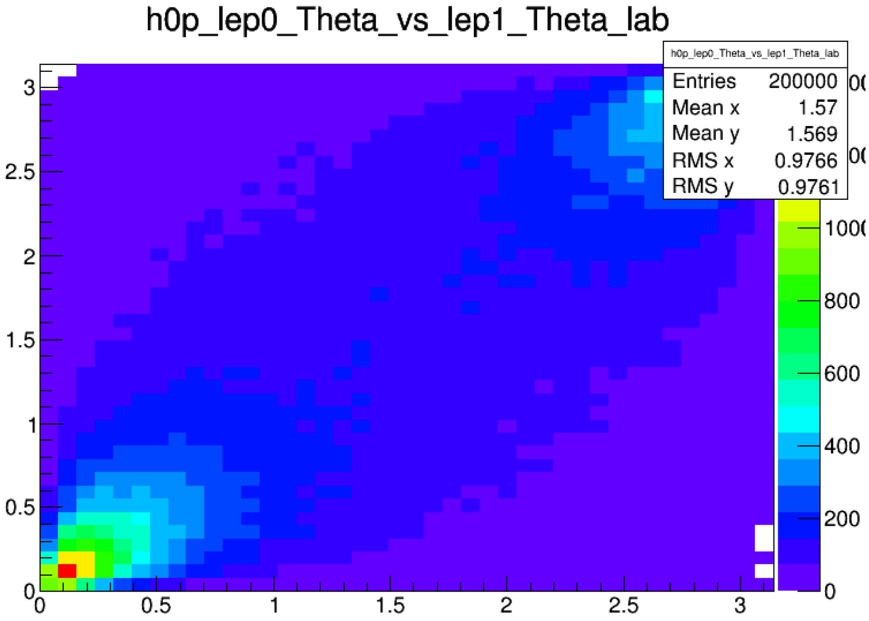


Figure 17: $\theta_{l_1} : \theta_{l_2}$ for 0^+ signal, in the lab frame

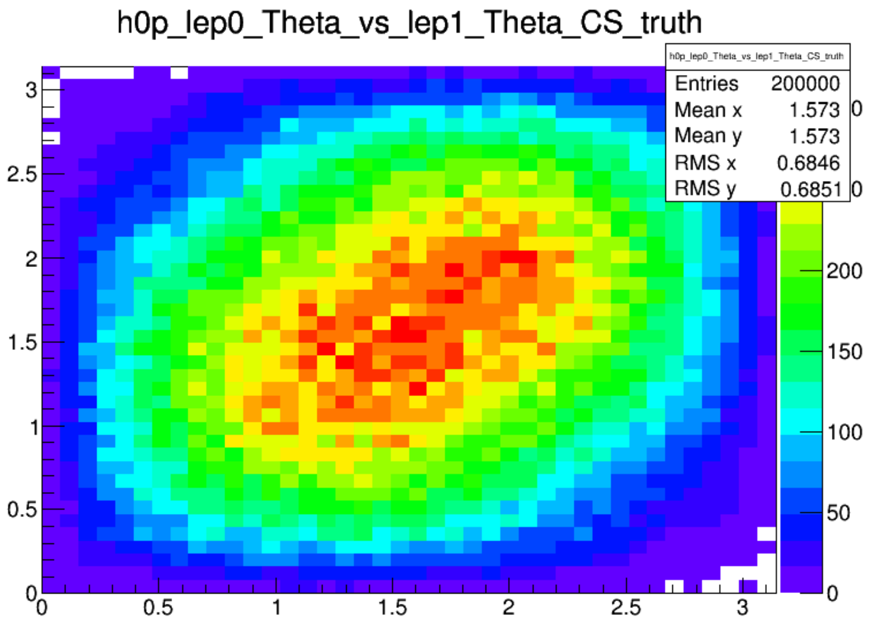


Figure 18: $\theta_{l_1} : \theta_{l_2}$ for 0^+ signal, in the truth CS-frame

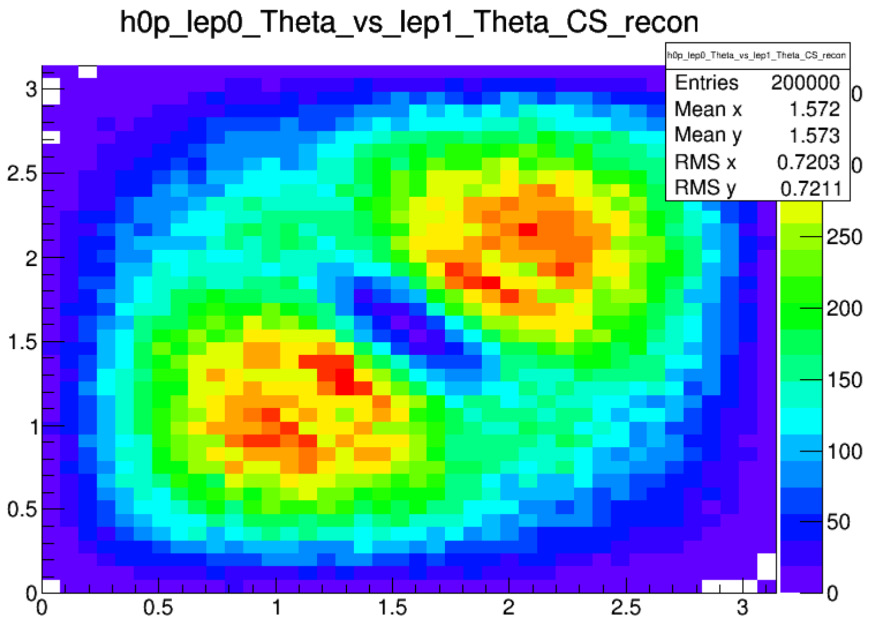


Figure 19: $\theta_{l_1} : \theta_{l_2}$ for 0^+ signal, in the recon CS-frame

In Fig. 20 the effect of the detector sensitivity simulation is visible. The scatter plot shows the $t\bar{t}$ background in the lab frame with only the preselection cuts applied, where the horizontal and vertical lines correspond to crack regions of the detector. The vertical crack regions belong to the electromagnetic calorimeter, responsible for measuring the electron, which has a transition from barrel to endcap at $|\eta| = 1.4$ ($\theta \approx 0.5, 2.7$). The horizontal line at $\theta = \pi$ is due to the cabling of the muon spectrometer prohibiting measurements at that angle.

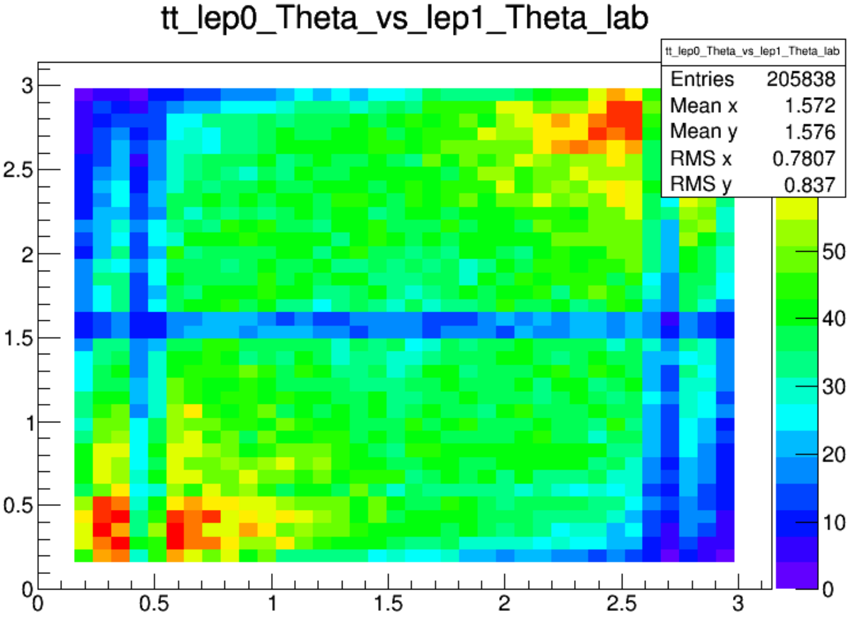


Figure 20: $\theta_{l_1} : \theta_{l_2}$ for 0^+ signal, in the lab frame. It showcases the effect of the detector sensitivity simulation, as the crack regions become visible.

B Cutflow diagram

	ggF [125 GeV]	WW	$WZ/ZZ/W\gamma$	$t\bar{t}$	Single Top	$Z \to \ell\ell + \gamma$/jets	$Z \to \tau\tau + \gamma$/jets	W+jets	Total Bkg.	Observed	Data/MC
blinding	536.96 ± 2.16	10032.91 ± 13.64	3780.90 ± 19.31	59507.95 ± 28.75	5683.26 ± 7.20	1140.54 ± 59.81	18023.96 ± 55.46	18716.38 ± 46.83	116885.89 ± 101.40	108758	0.93 ± 0.00
lepton p_{T}	432.33 ± 1.94	9367.45 ± 13.19	2768.15 ± 16.14	55842.75 ± 27.88	5320.19 ± 6.79	343.12 ± 24.17	11698.95 ± 44.73	7217.62 ± 27.67	92558.22 ± 67.89	92564	1.00 ± 0.00
Scale factors				NF = 1.05	NF = 1.05						
OS leptons	431.74 ± 1.94	9334.88 ± 13.17	1384.33 ± 11.46	58352.60 ± 29.18	5529.02 ± 6.82	223.08 ± 23.32	11653.70 ± 44.64	4288.32 ± 21.61	90765.92 ± 64.85	88772	0.98 ± 0.00
$m_{\ell\ell} > 12, 10$ GeV	431.74 ± 1.94	9334.88 ± 13.17	1384.33 ± 11.46	58352.60 ± 29.18	5529.02 ± 6.82	223.08 ± 23.32	11653.70 ± 44.64	4266.66 ± 21.54	90744.26 ± 64.83	88772	0.98 ± 0.00
Z veto (for $ee, \mu\mu$)	431.74 ± 1.94	9334.88 ± 13.17	1384.33 ± 11.46	58352.60 ± 29.18	5529.02 ± 6.82	223.08 ± 23.32	11653.70 ± 44.64	4266.66 ± 21.54	90744.26 ± 64.83	88772	0.98 ± 0.00
Scale factors				NF = 1.05	NF = 1.05		NFs applied				
$E_{\mathrm{T,clj}}^{\mathrm{miss,track}} > 20$	373.93 ± 1.81	8553.38 ± 12.61	1074.03 ± 10.12	35394.88 ± 22.72	4081.78 ± 5.86	142.35 ± 19.87	7532.50 ± 35.34	1937.82 ± 13.15	58716.73 ± 51.27	57953	0.99 ± 0.00
Z validation region (incl)	431.74 ± 1.94	9334.88 ± 13.17	1384.33 ± 11.46	58352.60 ± 29.18	5529.02 ± 6.82	223.08 ± 23.32	11653.70 ± 44.64	4266.66 ± 21.54	90744.26 ± 64.83	88772	0.98 ± 0.00
Top validation region (incl)	39.20 ± 0.61	564.23 ± 3.34	139.65 ± 3.59	42600.73 ± 24.97	3422.16 ± 5.22	19.13 ± 6.92	889.30 ± 14.20	497.80 ± 8.16	48133.00 ± 31.48	48133	1.00 ± 0.00
Scale factors		NF = 1.19		NF = 1.05	NF = 1.05		NFs applied				
0j: jet veto	210.22 ± 1.35	6478.19 ± 11.98	517.97 ± 7.28	750.21 ± 3.31	370.55 ± 1.89	83.35 ± 14.61	2844.79 ± 22.29	938.47 ± 8.01	11983.53 ± 31.39	10670	0.89 ± 0.01
0j: $p_{\mathrm{T},\ell\ell}$ >20 GeV	205.17 ± 1.33	6195.35 ± 11.72	492.66 ± 7.12	719.06 ± 3.24	357.76 ± 1.86	64.34 ± 10.99	1566.59 ± 16.49	802.95 ± 6.98	10198.70 ± 25.36	8875	0.87 ± 0.01
0j: $m_{\ell\ell} < 80$ GeV	204.18 ± 1.33	2878.06 ± 7.97	323.76 ± 5.91	234.53 ± 1.85	133.94 ± 1.22	44.18 ± 8.03	1403.17 ± 15.57	437.48 ± 5.63	5455.12 ± 21.02	4321	0.79 ± 0.01
0j: $\Delta\phi_{\ell\ell} < 2.8$	200.54 ± 1.32	2812.89 ± 7.87	311.34 ± 5.81	229.61 ± 1.83	131.59 ± 1.19	31.18 ± 5.10	514.85 ± 9.14	366.06 ± 4.85	4397.51 ± 15.28	3270	0.74 ± 0.01
0j: $0.5 \cdot m_{\mathrm{H}} \leq m_{\mathrm{T}} \leq 1.2 \cdot m_{\mathrm{H}}$	197.96 ± 1.31	2338.36 ± 7.17	294.73 ± 5.70	138.63 ± 1.42	79.78 ± 1.00	30.18 ± 5.07	513.76 ± 9.13	356.00 ± 4.78	3751.43 ± 14.80	2593	0.69 ± 0.01
0j: Z validation region	210.22 ± 1.35	5459.22 ± 10.10	517.97 ± 7.28	747.95 ± 3.30	369.44 ± 1.88	83.35 ± 14.61	2948.68 ± 23.10	2300.34 ± 15.88	12426.95 ± 34.19	10670	0.86 ± 0.01
Scale factors							NFs applied				
0j: $Z \to \tau\tau$ control region	5.63 ± 0.22	86.15 ± 1.26	18.23 ± 1.32	8.91 ± 0.35	3.97 ± 0.28	14.74 ± 6.24	1922.82 ± 18.53	128.26 ± 4.08	2183.08 ± 20.06	2198	1.01 ± 0.02
0j: $SSCR$ control region	0.29 ± 0.05	5.90 ± 0.33	320.42 ± 5.88	1.01 ± 0.12	0.90 ± 0.15	26.13 ± 2.86	4.65 ± 0.98	196.86 ± 3.78	555.87 ± 7.63	547	0.98 ± 0.04
Scale factors		NF = 1.19		NF = 1.05	NF = 1.05		NFs applied				
0j: WW control region	0.99 ± 0.09	2196.69 ± 7.00	116.47 ± 3.31	278.96 ± 2.02	139.56 ± 1.15	14.23 ± 6.05	140.21 ± 4.95	277.94 ± 3.68	3164.06 ± 11.83	3169	1.00 ± 0.02
Scale factors		NF = 1.04		NF = 1.03	NF = 1.03		NFs applied				
1j: one jet	127.38 ± 1.06	2581.24 ± 7.06	425.43 ± 6.10	7719.82 ± 10.53	2119.39 ± 4.19	55.49 ± 13.42	4105.00 ± 24.45	720.19 ± 8.41	17726.55 ± 32.62	17296	0.98 ± 0.01
1j: b-jet veto	110.65 ± 0.98	2254.18 ± 6.56	360.37 ± 5.62	1518.72 ± 4.70	517.55 ± 2.25	46.98 ± 12.42	3575.16 ± 22.61	579.97 ± 7.34	8852.92 ± 28.66	8361	0.94 ± 0.01
1j: $Z \to \tau\tau$ veto	92.57 ± 0.90	1558.40 ± 5.45	233.52 ± 4.59	1034.91 ± 3.87	362.83 ± 1.88	16.71 ± 5.34	536.20 ± 9.30	236.23 ± 4.12	3978.80 ± 14.19	3508	0.88 ± 0.02
1j: $m_{\ell\ell} < 80$ GeV	92.32 ± 0.90	785.43 ± 3.86	152.14 ± 3.83	479.14 ± 2.63	179.45 ± 1.40	10.86 ± 3.71	514.13 ± 9.09	145.23 ± 3.53	2266.38 ± 12.14	1811	0.80 ± 0.02
1j: $\Delta\phi_{\ell\ell} < 2.8$	90.00 ± 0.88	749.20 ± 3.77	144.93 ± 3.73	464.40 ± 2.59	173.65 ± 1.34	5.07 ± 1.26	223.87 ± 5.93	127.67 ± 3.16	1888.79 ± 9.13	1455	0.77 ± 0.02
1j: $0.5 \cdot m_{\mathrm{H}} \leq m_{\mathrm{T}} \leq 1.2 \cdot m_{\mathrm{H}}$	87.73 ± 0.87	540.87 ± 3.20	130.72 ± 3.58	274.80 ± 1.99	106.02 ± 1.14	4.73 ± 1.22	223.58 ± 5.92	122.67 ± 3.08	1403.38 ± 8.63	1009	0.72 ± 0.02
1j: Z validation region	144.90 ± 1.13	2793.36 ± 7.21	535.60 ± 6.83	8417.00 ± 11.08	2337.74 ± 4.43	100.46 ± 15.81	6065.32 ± 29.08	1086.22 ± 10.34	21335.71 ± 38.00	21211	0.99 ± 0.01
Scale factors							NFs applied				
1j: $Z \to \tau\tau$ control region	6.06 ± 0.23	85.38 ± 1.25	20.67 ± 1.30	49.04 ± 0.84	15.93 ± 0.48	5.33 ± 3.36	916.88 ± 11.40	54.07 ± 2.69	1147.31 ± 12.35	1163	1.01 ± 0.03
1j: $SSCR$ control region	0.10 ± 0.03	1.76 ± 0.18	140.75 ± 3.66	2.26 ± 0.18	1.86 ± 0.42	5.78 ± 1.28	1.62 ± 0.54	77.17 ± 2.54	231.20 ± 4.69	245	1.06 ± 0.07
Scale factors		NF = 1.04		NF = 1.03	NF = 1.03		NFs applied				
1j: WW control region	0.43 ± 0.06	1141.95 ± 4.68	130.28 ± 3.22	833.44 ± 3.48	268.96 ± 1.54	15.42 ± 8.67	79.02 ± 3.45	171.68 ± 3.02	2640.75 ± 11.95	2646	1.00 ± 0.02
Scale factors		NF = 1.04		NF = 1.03	NF = 1.03		NFs applied				
1j: Top control region	14.37 ± 0.37	285.26 ± 2.45	54.07 ± 2.15	5450.32 ± 8.85	1398.57 ± 3.30	2.71 ± 1.80	147.72 ± 5.21	88.40 ± 3.25	7427.06 ± 11.86	7416	1.00 ± 0.01

C Histloop class and its subclasses

Histloop

The Histloop class has been written to make it easier to set up an analysis loop when managing ROOT files. Normally, for each combination of particle of interest and quantity, a fields need to be edited manually, resulting in time-consuming work that in any programmer's opinion should be automated. The Histloop class does exactly that: it enables users to add components such as particles (Lorentz four-vectors), quantities (e.g. E, M, P_T ...), frame calculation functions and ROOT files to a Histloop object, which in turn will fill a histogram for *every combination* of these added components. The histograms are then saved to a new ROOT file, while also exporting the histograms to .png files. These histograms will be merged where necessary, creating a good comparison of the measured quantities for different frames and files.

Histloopscatter

A derivative of the Histloop class is Histloopscatter. Unlike the Histloop class, sets of particle quantities will be calculated and plotted against each other, thus creating scatter plots (2-D histograms). This can be used to analyze the relation between quantities of different particles.

Histloopspec

Another derivative of the Histloop class is Histloopspec, where 'spec' stands for 'special'. This class allows for more complex quantities to be calculated by performing special functions that take two particles' four-vectors as arguments, instead of standard quantities that are calculated from a sole four-vector. This way, quantities such as the sum or difference of the absolute momenta of two particles can be made visible.

Basic usage

The power of the Histloop classes lie within their ability to let the user easily change what will be analyzed. The classes keep track of which components have been added and makes changes the looping structure accordingly. Starting off with a fresh new iterator class, made by performing the MakeClass function of a Tree within ROOT, the following needs to be included at the top in order to make use of the Histloop classes:

```
// change paths to correct location
#include "Histloops.h"
#include "Histloop.C"
#include "Histloopscatter.C" // optional, if you want scatter plots
#include "Histloopspec.C" // optional, if you want to use special quantities
```

```
#define PI TMath::Pi() // 3.14159265358979323846
using namespace std;
```

Next, write (global) functions that will be used to calculate a any quantity for any particle in a specific frame of reference. These "frame functions" accept a pointer to the particle's four-vector, as well as a pointer to a member function of ROOT's TLorentzVector class that calculates a single quantity. The user should use both of these arguments to create a function that calculates the quantity after boosting the vector to its desired frame. An example is shown below, the firsts of which simply calculates the quantity in the lab frame (no boosting), and the second boosts the input vector to the CS frame by using some external script and then calculates the quantity from the new vector.

```
double CM_frame(TLorentzVector *v, fptr f) {
    // applies function f on vector v
    return (*v.*f)();
}
double CS_recon(TLorentzVector *v, fptr f) {
    // boosts v to the CS frame using global TLorentsVectors l1, l2 and dint
    // then applies function f on the resulting four-vector
    return (CS_lorentz(l1, l2, dinr, *v).*f)();
}
```

When using the Hisloopspec class, two types of additional functions need to be written: functions that calculate the quantity from two vectors, and functions that boost two vectors to the right frame. The latter is very similar to the functions written before, with the main difference that it takes a different arguments (additional TLorentzVector* and a different kind of function pointer). Below is an example of each of those functions:

```
// calculates the sum of absolute momenta of two four-vectors (particles)
double lepsumP(TLorentzVector v1, TLorentzVector v2) {
  return (v1.P() + v2.P());
}
// boosts two four-vectors to the CS frame
// then passes them to functions likes the one above (lepsumP)
double CS_recon_spec(TLorentzVector* v1, TLorentzVector* v2, specfptr f) {
  return (*f)(CS_lorentz(l1, l2, dinr, *v1), CS_lorentz(l1, l2, dinr, *v2));
}
```

Next up is to tell ROOT that this is a batch process, meaning it will not open canvases user interfaces upon drawing. This has to be set in the ::Loop() function of your iterator class, preferably at the very beginning:

```
gROOT->SetBatch();
```

I will now show how to further implement the Histloop class, with an example that uses both subclasses as well. The first step is to instantiate the classes, accompanied by some verbosity to make the output more user friendly:

```
cout << "Creating Histloops... "; cout.flush();
    // Create an instance of Histloop (and optional: its derivatives)
    Histloop*           hl   = new Histloop                ("bkg_WW2em_multi");
    Histloopscatter*         hls = new Histloopscatter   ("bkg_WW2em_multi", hl);
    Histloopspec*            hlc = new Histloopspec            ("bkg_WW2em_multi", hl);
    // Adding hl as second argument to hls and hlc changes some functionality,
    // making them slaves of Histloop. Do not add if using any of these classes alone
cout << "done" << endl;
```

The following step is to add components that will be analyzed, such as particles,

```
// Add particles, max 10
hl->addPart("lep0", &l1); // l1 is a (global) TLorentzVector
hl->addPart("lep1", &l2);
hl->addPart("W0", &w1);
hl->addPart("W1", &w2);
hl->addPart("H", &H);
...
```

and quantities, with their corresponding TLorentzVector function and histogram boundaries,

```
// Add quantities, max 10
hl->addQuant("E",       0,      250,    &TLorentzVector::E);
hl->addQuant("Pt",      0,      200,    &TLorentzVector::Pt);
hl->addQuant("Phi",     -PI,    PI,     &TLorentzVector::Phi);
hl->addQuant("Theta",   0,      PI,     &TLorentzVector::Theta);
hl->addQuant("M",       0,      140,    &TLorentzVector::M);
...
```

and frames, with references to the frame functions written earlier,

```
// Add frames, max 3
hl->addFrame("lab",         &CM_frame);
hl->addFrame("CS_recon",    &CS_recon);
...
```

and finally files will be added. This step is optional: if no files are added, the default tree will be loaded, which is the tree that was used to create the iterator class through MakeClass. Note that the files will be given a short name, along with the name of the correct tree within the that should be used and the path to the actual ROOT file. Also note that all trees need to have the very same internal structure as the default tree, as leaf names are specific to the simulation program used to create the tree.

```
// Add files (optional), max 10
hl->addFile("h0p",  "Tree",     "../h0pww_aMCatNLO_corrected_200k_tree.root");
```

```
hl->addFile("h0m",  "Tree;6",   "../h0mww_aMCatNLO_corrected_200k_tree.root");
hl->addFile("h1p",  "Tree;5",   "../h1pww_aMCatNLO_corrected_200k_tree.root");
hl->addFile("h1m",  "Tree;5",   "../h1mww_aMCatNLO_corrected_200k_tree.root");
hl->addFile("h2mix","Tree;6",   "../hmix2mpww_aMCatNLO_corrected_200k_tree.root");
...
```

The Histloopscatter instance undergoes a similar process, with the exception of adding files. This is because in this case the Histloopscatter class is a slave of the Histloop, and will therefore let the file management be done by its master. The same goes for the Histloopspec instance, below is an example of how to prepare these two subclasses after preparing the Histloop:

```
// --> Histloopscatter
hls->addPart("lep0", &l1, "lep1", &l2);
hls->addQuant("Phi",    -PI,    PI, 40, &TLorentzVector::Phi);
hls->addQuant("Theta",  0,      PI, 40, &TLorentzVector::Theta);
hls->addFrame("CS_recon", &CS_recon);
// --> Histloopspec
hlc->addPart("lep0", &l1, "lep1", &l2);
hlc->addQuant("lepsumP",    0,      150,    &lepsumP);
hlc->addQuant("lepdifP",    -80,    80,     &lepdifP);
hlc->addFrame("CS_recon", &CS_recon_spec);
// note that 'lepsumP' and 'CS_recon_spec' have been described above
```

After adding every component of interest, it is time to prepare the Histloop-like instances for looping. This is simply done by calling their `init()` function. It will generate the necessary histogram objects and several other output functionalities.

```
cout << "Initializing histograms... " << endl;
    hl ->init();        // Ready the Histloop for looping
    hls->init();        // Ready the Histloopscatter for looping
    hlc->init();        // Ready the Histloopsspec for looping
cout << "done" << endl;
```

In the original code, the actual looping takes place in the `for` loop that looks like this:

```
for (Long64_t jentry=0; jentry<nentries;jentry++) {
    ... // here is where the iterations take place
}
```

Because we (might) have multiple files, or a file different from the standard file, we nest the current loop within a new loop, that ranges from -1 to 0 (one iteration) in case no custom files are added (i.e. the user did not use `addFile()` at all), or from 0 to n (thus $n-1$ iterations), with n the number of files added. A boolean keeps track of the fact whether custom files have been added. If this is the case, the iterator instance has to prepare for this new file, by calling the Histloop's `prepareFile()` function. This results in the following new structure:

```
bool custfiles = (hl->getNumFiles() > 0);
for (int l = -!custfiles; l < hl->getNumFiles(); l++) {
    // check if custom files are added. If not, the standard file will be loaded
    if (custfiles) {
        hl->prepareFile(this,l);
    }
    for (Long64_t jentry=0; jentry<nentries;jentry++) {
        ... // here is where the iterations take place
    }
}
```

Moving on to the interior of the inner `for` loop (”where the iterations take place”), this is where the actual physics are described. The entries of the tree will be loaded iteratively, and for each entry several particles will be constructed by using the leaves present in the entry. This is first done by setting setting up four-vectors with quantities found in the leaves, and then creating other particles with the set four-vectors:

```
// note that leaf names such as 'l1Pt' etc are specific
// to the tree used in this example!
l1.SetPtEtaPhiE(l1Pt,l1Eta,l1Phi,l1E); // set the four-vector of lepton 1
n1.SetPtEtaPhiE(v1Pt,v1Eta,v1Phi,v1E); // same for neutrino 1
...
w1 = l1+n1;
H = w1+w2; // Higgs boson
...
```

The actual processing of these newly set four-vectors will be done by the Histloop, and can be accomplished by simply calling the Histloop's `iterate()`. The user must supply the weight of the entry (extracted from one of the leaves), and also the correct file number (0 without custom files and $l \in [0, n-1]$ otherwise):

```
// !custfiles = 1 when no files are added, 0 otherwise
// this results in a second argument of:
//      -1 + 1 = 0 when no custom files are specified,
//      l + 0 = l when custom files are specified
hl ->iterate(mcEventWeight, l+!custfiles);
hls->iterate(mcEventWeight, l+!custfiles);
hlc->iterate(mcEventWeight, l+!custfiles);
```

The following code can be added to make the program more verbose, showing which entry it's currently processing:

```
if ((ientry+1) % 1000 == 0) { // update status every 1000th entry
    cout << "\r\tLooping over entries...    " << (ientry+1) << "/" << nentries;
    cout.flush();
}
```

After both the inner (entry) and outer (file) `for` loops have finished, the histograms have been filled and are ready to be exported, which is done by the Histloop's `plotexport()` function:

```
// Note that directories need to exist before trying to export to them
hl ->plotexport("multiplots");  // write files to the "multiplots" directory
hls->plotexport("scatterplots");// ... etc
hlc->plotexport("specialplots");
```

Correct usage of the class yields a final ROOT output like the following:

```
Creating Histloop... done
Preparing components... done
Initializing histograms... Created 315 histograms for
        9 particles
        7 quantities
        1 frame
        5 files
done
File: h0p        Tree name: Tree
        Looping over entries...    200000/200000... done
File: h0m        Tree name: Tree;6
        Looping over entries...    200000/200000... done
File: h1p        Tree name: Tree;5

        Looping over entries...    200000/200000... done
File: h1m        Tree name: Tree;5
        Looping over entries...    200000/200000... done
File: h2mix        Tree name: Tree;6
        Looping over entries...    200000/200000... done
Time spent looping: Real time 0:01:43.507, CP time 102.510

Histograms output directory: multiplots
Plots saved:
lep0_E  lep0_P  lep0_Pt  lep0_Phi  lep0_Eta  lep0_Theta  lep0_M
lep1_E  lep1_P  lep1_Pt  lep1_Phi  lep1_Eta  lep1_Theta  lep1_M
neu0_E  neu0_P  neu0_Pt  neu0_Phi  neu0_Eta  neu0_Theta  neu0_M
neu1_E  neu1_P  neu1_Pt  neu1_Phi  neu1_Eta  neu1_Theta  neu1_M
W0_E  W0_P  W0_Pt  W0_Phi  W0_Eta  W0_Theta  W0_M
W1_E  W1_P  W1_Pt  W1_Phi  W1_Eta  W1_Theta  W1_M
din_E  din_P  din_Pt  din_Phi  din_Eta  din_Theta  din_M
dil_E  dil_P  dil_Pt  dil_Phi  dil_Eta  dil_Theta  dil_M
H_E  H_P  H_Pt  H_Phi  H_Eta  H_Theta  H_M
Histograms saved to h012_lepdif_multi_hists.root
```

Note that the Histloop is serving its purpose at it has taken care of 315 histograms at the same time. Another example output, which uses all three classes at once, is shown below:

```
Creating Histloops... done
Preparing components... done
Initializing histograms...
Created 70 histograms for
        5 particles
        7 quantities
        1 frame
        2 files
Created 4 scatterplots for
        1 particle
        2 quantities
        1 frame
        2 files
Created 4 special histograms for
        1 particle set
        2 quantities
        1 frame
        2 files
done
File: WW2em        Tree name: HWWTree_em
        Looping over entries...    7170/7170... done
File: WW2me        Tree name: HWWTree_me
        Looping over entries...    7170/7170... done
Time spent looping: Real time 0:00:02.268, CP time 2.070

Histograms output directory: multiplots
Plots saved:
lep0_E  lep0_P  lep0_Pt  lep0_Phi  lep0_Eta  lep0_Theta  lep0_M
lep1_E  lep1_P  lep1_Pt  lep1_Phi  lep1_Eta  lep1_Theta  lep1_M
dinr_E  dinr_P  dinr_Pt  dinr_Phi  dinr_Eta  dinr_Theta  dinr_M
dil_E  dil_P  dil_Pt  dil_Phi  dil_Eta  dil_Theta  dil_M
H_E  H_P  H_Pt  H_Phi  H_Eta  H_Theta  H_M
Histograms saved to bkg_WW2em_multi_hists.root

Scatterplots output directory: multiplots
Scatterplots saved:
WW2em_lep0_Phi_vs_lep1_Phi_CS_recon  WW2me_lep0_Phi_vs_lep1_Phi_CS_recon
WW2em_lep0_Theta_vs_lep1_Theta_CS_recon  WW2me_lep0_Theta_vs_lep1_Theta_CS_recon
Histograms saved to bkg_WW2em_multi_scatterplots.root

Special plots output directory: multiplots
Special plots saved:
CS_recon_lep0_lep1_lepsumP  CS_recon_lep0_lep1_lepdifP
Histograms saved to bkg_WW2em_multi_specialplots.root
```

This showcases the use of the Histloop class and its derivatives, and its relative simplicity to use. After successfully analyzing a set of files, the user can easily move on to the next analysis by simply changing a small number of lines at the beginning of the code.

Advanced usage example

A user can easily apply cuts to the loop by entering a `continue` statement right before the `iterate(...)` commands in case cuts requirements are not met. In other cases, a more specialized analysis method is needed where the events need to be sectioned based on certain criteria. For example, a user would want to see how plots would look for different sections of η, and has written a sectioning function that returns an integer value based on η, ranging from 0 to the $n_{sections} - 1$. Instead of creating a single instance of each type of Histloop object, the user would then make *arrays* of Histloop objects, and repeat the initialization procedure by looping over every element of the array (unfortunately, Histloop cloning functions are not available as of yet). Right before the `iterate(...)` functions are called, the sectioning function is executed which will decide which Histloop instance in the array will perform the `iterate(...)` command.

For more information on how to use the classes, or to see the final usage scripts as described in this section, please contact me.

Samenvatting voor 6VWO

Deeltjes uit het standaardmodel hebben vele eigenschappen. Één van die eigenschappen is spin, dat doet denken aan dat het deeltje om zijn as draait alsof het een tol is. De 'spin' van het deeltje, S, geeft aan hoe hard die tol dan maximaal zou kunnen draaien, maar het deeltje zal uiteindelijk draaien met een waarde die ligt tussen S en $-S$. Deze waarde noemt men de *(spin)projectie* van de spin, en kan alleen veranderen in stappen van 1. Bijvoorbeeld, een spin-1 deeltje zou dus een spinprojectie kunnen hebben van 1, 0 of -1.

Electronen, muonen en neutrino's zijn deeltjes met spin $\frac{1}{2}$, dus kunnen een projectie hebben van $\frac{1}{2}$ of $-\frac{1}{2}$. Neutrino's hebben ook nog de eigenschap dat (in de gevallen die hier besproken worden) de spinprojectie altijd wijst in de richting waar de neutrino vandaan komt, en dat die van antineutrino's juist met de antineutrino mee wijst.

Bosonen zijn deeltjes die een spin hebben van een geheel getal: 0, 1, 2, etc. W bosonen hebben bijvoorbeeld spin 1. Onlangs is het higgsboson ontdekt en als het standaardmodel klopt, betekent dat dat het Higgs boson een spin van 0 heeft. In dit onderzoek is gekeken naar het volgende gebeurtenissen die er ongeveer zo uitzien:

$$H \to W^+ + W^- \quad \text{en daarna} \quad W^+ \to \mu^+\nu \text{ en } W^- \to e^-\bar{\nu}$$

Oftewel, een higgsboson dat vervalt naar een W^+ en een W^- die een tegenovergestelde richting hebben, waarvan W^+ daarna vervalt naar een antimuon (μ^+, een muon is een soort 'zwaar' electron) en een neutrino, en de W^- vervalt naar een electron (e^-) en een antineutrino. Als de spin van het higgsboson 0 is, is dus de projectie ook 0. Spinprojectie is altijd behouden, dus als je begint met een projectie van 0, eindig je daar ook mee. De W bosonen moeten dus een projectie hebben die samen 0 is, oftewel: een tegengestelde projectie. Elk van de W bosonen valt uiteen in twee deeltjes met spin $\frac{1}{2}$ die samen weer dezelfde projectie moeten hebben als het W boson. Dit betekent dus dat de spin van de neutrino en de antineutrino ook tegengesteld is! Als we de spin van de neutrino's weten, weten we ook welke kant ze (ongeveer) op gaan. Het electron en het muon moeten samen dus de andere kant op gaan vergeleken met de neutrino's, zodat alle snelheidsvectoren bij elkaar geteld je weer op 0 uitkomt. Op deze manier is dus beredeneerd dat het electron en het muon min of meer dezelfde kant op gaan, dus een kleine hoek tussen elkaar hebben. Voor higgsbosonen met spin 1 of 2 kan hetzelfde gedaan worden, en daar komt uit dat spin 1 geen voorkeur heeft voor hoeken, en bij spin 2 het electron en het muon graag een grote hoek tussen elkaar hebben. Je kan dus aan de hoek zien wat de spin van het higgsdeeltje is!

In dit onderzoek is gekeken of deze theorie klopt door honderdduizenden gebeurtenissen te bekijken met *gesimuleerde* higgsbosonen van spin 0, 1 of 2, en elke keer de hoek tussen elektron en muon te meten. Daarna wordt er gekeken of het mogelijk is om in echte data te kijken of te zien is welke spin higgsboson heeft. Het is daadwerkelijk gelukt om de voorkeur voor de hoek terug te zien, maar de echte data was nog niet compleet genoeg om iets over de spin te zeggen.