



Anomalous Couplings Project

Progress Report and documentation of Tools

Annik Olbrechts

Started 4 April 2014
Version: December 11, 2014



Contents

1	Pending issues	1
2	Used Tools and Techniques	3
2.1	MadGraph	3
2.1.1	MadGraph_v155 version	3
2.1.2	New MadGraph5_aMC@NLO version	3
2.2	MadWeight	4
2.2.1	MC_PERM MadWeight use	5
2.2.2	Checking MadWeight on localgrid	6
2.2.3	Influence of the used Transfer Function	6
2.2.4	Complementarity with Git	6
2.2.5	Adapting MadWeight to run continuously on localgrid	6
2.3	MadAnalysis	7
2.3.1	How to run MadAnalysis in expert mode	7
2.3.2	Content of analysis file in MadAnalysis	8
2.3.3	Analyzing the MadGraph files	8
2.3.4	Width of the decay	9
2.4	FeynRules	9
3	Generator level bottlenecks	10
3.1	Transfer Functions	10
3.1.1	Creating the Transfer Function	11
3.1.1.1	Using p_T dependency	12
3.1.1.2	Creation of the Transfer Functions	12
3.1.1.3	Importance of start values	13
3.1.1.4	Splitting in separate $ \eta $ bins	14
3.1.1.5	Separating narrow and wide gaussian	15
3.1.1.6	Coping with low statistics in last $ \eta $ bin	15
3.1.1.7	Extrapolation method	15
3.1.1.8	Actual implemenation in MadWeight	16
3.1.2	Obtained distributions	16
3.1.3	Control checks for Transfer Functions	19
3.1.3.1	Compare results with normal (single) Gaussian	19
3.1.3.2	Compare result with previous analyses	19
3.1.3.3	Compare results with predefined ROOT class	19
3.2	Cross Section distribution for new grid	20
3.3	First results: Wrong log(likelihood) minimum	20
3.4	Correct normalisation of Matrix Element probability	22

3.4.1	Measurement of top quark mass using Matrix Element Method . .	22
3.4.2	Influence of the acceptance term	24
3.4.2.1	Understanding 1D-variation of V_L	27
3.4.2.2	Fixing normalisation for 1D-variation of V_R only	29
4	Preliminary Results	31
5	Understanding Results Obtained With MadWeight	32
6	Analyzing created FeynRules model	33
7	Event Selection	34
8	Event corrections and reconstruction	35

Chapter 1

Pending issues

- Does this created FeynRules model still contain Effective Field Theory?
- If the kinematics doesn't change for coupling parameters larger than 1, how is it than possible to differentiate the different configurations which will be studied?
- Access to Mathematica for Width studies ... ?
- XS values which are used are the ones calculated using the MadGraph_v155 version. Should this be calculated again using the new MadGraphv5_aMC@NLO since normally this shouldn't change ...?
- Create branch for localgrid scripts! (Now already a branch exists for the documentation !)
- Understand *CorrectPhi* comment in MadWeight name
→ Maybe related with phi issues of neutrino's ...
- EventWeight calculated in analyzer should be integrated into MadWeight output (Hence multiply MadWeight weight with the EventWeight for each event!)
Otherwise all the effort to include the JES and PU stuff has no influence at all and is not taken into account!
- Update EventNumberInformation file to only cout the information when the event has passed the eventSelection requirements. Otherwise just discard this event from the output file in order to avoid a long .txt file and long running time ...
- When creating ntuples for Data/MC comparison the EventNumberInformation output should be stored here as well. This will allow to calculate the likelihood and weight distributions for specific p_T cuts without having to run the entire base code again. So also the different python scripts which have been created should be moved to this .cc code in the future, so better not to spent too much time on improving these python scripts to avoid double work ...
For storing the generator decay channel information the c++ enum should be used which can store multiple variables (**Check this out!**)
- Need to understand how MadWeight deals with the permutations ... Should this permutation of the light jets be done within MadWeight or should two .lhco files be created and sent separately to MadWeight. If the latter is the case, how should they be combined afterwards ?

- Be careful with used Beam Energy when using MadGraph. In the newest MadGraph directory this still seems to be set to 7TeV in stead of the required 4 GeV.
Could this have an influence on the obtained MadWeight results when using the new MadWeight version (connected to the newest MadGraph version) ...
- Specific MadWeight question for the latest version:
*The 'refine' option allows you to realunch the computation of the weights which have a precision lower than X. **But how can you find this precision?***
- Should a ScaleFactor correction be applied for the Branching ratio ?
This is not the case in code of James, but should find recommendations somewhere ...
- **To Check:** Does MadWeight give a different weight when the Neutrino Mass changes?
→ Currently mass is manually set to zero ...
- Running the analyzer code with the PileUp and JES/JER influences correct and active, the number of preselected events in the Event Selection Table has changed. Need to understand if it is expected that these corrections have an influence (lowering the number) on the number of preselected events ...
- Check whether the created root file can be trusted when the number of considered b-tags is greater than 1 ... !!

Chapter 2

Used Tools and Techniques

2.1 MadGraph

2.1.1 MadGraph_v155 version

Currently the MadGraph version which should be used is the *MadGraph_v155* since this is the only one which was compatible with the used FeynRules version at the time of the creation of the model. According to a mail of Olivier Mattelaere on 6 February 2014 this issue should now be fixed.

In this project MadGraph is only used to create .lhe files in order to understand the created model. Especially for Cross Section studies it is extremely useful. Hence two python scripts have been created in the following directory:

```
/user/aolbrech/AnomalousCouplings/MadGraph_v155/MassiveLeptons/  
MadGraph5_v1_5_5/Wtb_ttbarSemiElMinus/RVL_RVR_XSGrid003.py  
/user/aolbrech/AnomalousCouplings/MadGraph_v155/MassiveLeptons/  
MadGraph5_v1_5_5/XSScript.py
```

The first one automatically creates all the desired configurations, for different RVL, RVR, Lepton Pt Cut and Jet Pt Cut values. The desired output directory can be specified in this file as well.

In order to save some disk space the events.lhe file is deleted using this script and only the unweighted_events.lhe are kept for analysis. These are the relevant events (*according to Alexis so understand what is the difference*).

The second file calculates the cross section of the desired configurations. It again loops over all the different variables and creates a .txt table and a .pdf table.

Maybe this can be created automatically when executing the first python script! This to avoid copying the considered variables from one file to another!

2.1.2 New MadGraph5_aMC@NLO version

This new MadWeight version was released in the spring of 2014 and is accompanied with a new MadWeight version as well.

Normally the cross section calculations shouldn't differ between the different MadGraph versions, but for at least one of the decay channels this should be checked!

In order to use MadWeight and calculate the cross section values for the desired processes and decay channels the following commands should be used:

`./bin/mg5_aMC` (2.1)

`importmodelMassiveLeptons` (2.2)

`generatepp > tt , (t > bw+, w+ > mu + vm), (t > b w-, w- > jj)@1QED = 2` (2.3)

`outputdirName` (2.4)

`exit` (2.5)

Once this directory is created the *index.html* file contains all the Feynmann Diagrams which correspond to the considered process. This file can be opened using *firefox* on mtop.

Since the FeynRules diagram is created in such a way that it contains a **NP**, **QCD**, **QED** and **TEST** variable representing the different interaction vertices. Hence asking the variable **QED** to be equal to 2 results in the 16 independent diagrams which are required to represent the anomalies in the Wtb vertex. Currently the decay of the top quark is altered to contain all the different coupling constants while the other interaction vertices represent the Standard Model expectations.

When the cross section should be calculated for a specific process the following configuration files in the Cards directory should be updated.

- **run_card.dat** where the number of events which should be generated is defined together with the beam energy of the considered collision process. Also should be ensured that no kinematic cuts are active on the particles since these cuts will have a different effect then the cuts applied in the event selection of the analysis. These cuts are applied on the generator level instead of the jet-level which results in much more stringent and tighter effects when using the same p_T value.
- **param_card.dat** where all the parameters of the considered model should be defined.
- **proc_card.dat** contains the event which will be generated and the models which have been imported. This file shouldn't be changed but is particularly useful to ensure that the correct event is considered for generation.

All these files can still be changed once the `./bin/generate_events` command has been executed. The MadGraph software always asks whether one or more of the configuration files should be adapted before the command is submitted and allows a waiting time of 60s. Hence in order to run MadGraph continously using a script the **me5_configuration.txt** file should be adapted to enable this waiting time since otherwise when using `nohup` to run the script results in a crash and the script is terminated.

2.2 MadWeight

The running of the MadWeight event generator has to be done on localgrid since the *lhco* file containing all the events has to be splitted in multiple jobs. The number of jobs

which can be submitted is limited to 2000, hence the events per job should be taken as low as possible while not exceeding this limit. Running MadWeight with a large number of events per job implies that it will take a very long time since the submitted jobs will have a long walltime and therefore their priority will reduce.

Another important factor for the MadWeight event generator is defining the number of interaction points which have to be considered. This number has to be rather large (around 10 000 - 30 000 according to Lieselotte **TO CHECK MYSELF**) to ensure the uncertainty on the weight to be smaller than the weight itself.

Currently two different MadWeight event generator versions exist, and up to now (May 6 2014) only the oldest one has been tested and used extensively. The newer one was only installed beginning of May but should be less CPU intensive and allows the user to split the interaction points in two distinct steps. Since September 2014 solely the newest version is being used. The two versions are still installed on localgrid and are given here:

- madweight_mc_perm
- MadGraph5_aMC@NLO

The installation commands of these two versions are the following (the bzr command package has been installed on the m-machines, but not on mtop):

- bzr branch lp: maddevelopers/madgraph5/madweight
- bzr branch lp: maddevelopers/mg5amcnlo/madweight

Whenever MadWeight is first used a personal directory should be initialized. This can be done in the following way:

- Import the created model (called MassiveLeptons) in the model directory of MadGraph
- Activating the MadWeight event program and initializing a personal directory.
 1. `./bin/mg5_aMC` (The mg5 executable should not be used!)
 2. `import model MassiveLeptons`
 3. `generate p p > t t , (t > b w+ , w+ > mu+ vm) , (t > b w- , w- > j j)`
`@1 QED = 2`
 4. `output madweight`
 5. `exit`

2.2.1 MC_PERM MadWeight use

The most important steps which have to be executed to use this MadWeight version are the following ones ¹:

1. Initialize MadWeight to run on localgrid!
2. Update the MadWeight_card.dat

¹Full detailed explanation about the use of this madweight version can be found in the documentation of Bettina.

3. Set the correct transfer functions
4. Run MadWeight

For this first step the following two files have to be changed in the /blablaMadWeight/bin/internal directory:

- madweight_interface.py
- cluster.py

2.2.2 Checking MadWeight on localgrid

Different command which should be used to check whether jobs are running are running on localgrid and how they should be killed when something went wrong:

qstat @cream02 | grep aolbrech (2.6)

qdel 394402.cream02 (2.7)

2.2.3 Influence of the used Transfer Function

2.2.4 Complementarity with Git

In order to keep track of all the changes done to the configuration files of MadWeight, and keep to separate branches for both the original E -dependent Transfer Functions and the newly created p_T -dependent Transfer Functions the MadWeight directory on localgrid has been added to the AnomalousCouplings GitHub repository.

However this has an important restrictions on the use of the m-machines for compiling and running MadWeight and for performing GitHub related activities. This restriction is caused by the initialization command needed to run MadWeight, namely activating python 2.7.3, which results in a conflict with the GitHub requirements. Therefore the GitHub related activities have to be done either in a separate terminal window where no python initialization is done or otherwise on *mtop* where this command can't be executed. The compiling and running of MadWeight on the other hand can only be done in a different terminal window on the m-machines.

This implies that in the terminal window where the analysis is being executed no Git check can be done for checking the active branch or changing to another branch. These commands have to be done in the separate window.

2.2.5 Adapting MadWeight to run continously on localgrid

A single test with the following configuration (5000 events - 20 events/job - 10 000 init-Points) took almost 16h to finish so the full generator event sample ($> 2\,470\,000$ events) would take more than 329 days!²

Even the full reconstructed event sample ($> 210\,000$ events) requires more than 28 days of running this way.

TODO: Time of running should be compared against the new MadWeight version!!

²From this can be concluded that the events on generator level should never be all run. Only a limited selection of these events should be considered.

A possible solution to reduce the CPU time to process these events is instead of sending 2000 jobs, waiting until they are all finished and only then submitting the next bunch of 2000 jobs trying to adapt the MadWeight script to send continuously 2000 jobs. This would imply that the script should check whether one of the jobs have been finished and immediately submitting a new one.

For this the above mentioned scripts should be adapted!

According to Olivier (D) this can't be changed by the grid admins, but should be adapted by the MadGraph developers.

2.3 MadAnalysis

At first sight it seems that since the update of the m-machines to Scientific Linux 6 (*slc6*) MadAnalysis can only be used on *mtop*. Whenever MadAnalysis is trying to be compiled on any of the other m-machines an error message about a missing library appears, while compiling on *mtop* works from the first go.

2.3.1 How to run MadAnalysis in expert mode

Version v112 should be used since the expert mode in this version works exactly as explained in the manual (arXiv: 1206.1599). In the more recent version v115 the expert mode doesn't work out-of-the-box and the python files need to be adapted ...

Starting with a new analysis directory in MadAnalysis in expert mode can be done by typing the following command:

$$./bin/ma5 \quad - \quad -expert \quad (2.8)$$

In the questions asked by MadAnalysis after executing this command the name of the directory which needs to be created and the name of the analysis has to be given. The name of this analysis shouldn't be made too complex since it has to be typed every-time when creating plots with MadAnalysis.

After the correct directory is created, the *Name/SampleAnalyzer* directory should be initialized by executing the following two commands:

$$source \quad setup.sh \quad (2.9)$$

$$make \quad (2.10)$$

The actual analysis should be created in the Analysis directory, and a similar approach to user.cpp and user.h should be adopted.

Everytime a change has been made to these two files *make* should be executed in the SampleAnalyzer directory in order to process these changes.

The .lhe files (MadAnalysis cannot process .lhe.gz files so they should be unpacked using gunzip .lhe.gz) which should be considered should be written down in a .txt files which is saved in the SampleAnalyzer directory.

The actual running of MadAnalysis is done with the following command:

$$./SampleAnalyzer \quad --analysis = "Name\ of\ analysis" \quad List.txt \quad (2.11)$$

2.3.2 Content of analysis file in MadAnalysis

The latest analysis file which has been used with all the necessary information can be found in the following directory on the m-machines:

AnomalousCouplings/MadAnalysis_v112/Wtb_PtCutInfluence/SampleAnalyzer/Analysis

The two analysis files with full detailed information are the LeptonPtCutInfluence.cpp and the JetPtCutInfluence.cpp files. They both consist of two different functions, namely the *Execute* and the *Finalize* function. The first one allows to access the information of each event while the second one is entered for each file. Therefore the particle content is reconstructed in the *Execute* function and the histograms for all the considered files are constructed in the *Finalize* function. Currently these analyzer files look at 28 different kinematic variables. All the kinematic information of each of the particles present in the expected semi-leptonic $t\bar{t}$ event is created.

In order to separate the two b-quarks in the event, the Particle Id of the leptonic top quark needs to be known. Therefore an integer *LeptonicTopPdgId* is used and the kinematic information of the b-quarks can only be stored when this integer is different from zero. This will normally not result in by-passing the b-quark information since the events in the .lhe files are read in in the same order as they are created by the MadGraph command. So first the top quarks are considered and only then the final state particles.

? Isn't it safer to use the daughter information of the top quark? Because in the case of $t\bar{t} \rightarrow b\bar{b}jj$ $b\bar{b}l\bar{v}$ the lepton comes after the b-quark ... Should investigate what happens in this case ... **TO CHECK:** I think some extra safety is incorporated and the actual loop of filling the event content is only done when the b's are reconstructed!

In order to automatically create the 28 histograms for all the different files and distinguish the different RVR and RVL values, an automatic name-giving loop is used. The name of the content of the histogram is each of the time combined with the correct name of the RVL/RVR content of the .lhe file and the decayChannel.

Currently the name of the considered .lhe files has to be adapted every time the files are changed. Maybe this should be made automatically read in from the .txt file to ease the processing of different configurations. If the .lhe files have a clear name where the RVL, RVR and PtCut values can be obtained from using a python script it shouldn't be too much work to save this in the .cpp file with this script. **TO DO!**

2.3.3 Analyzing the MadGraph files

Currently the created model should be completely understood and the behavior of the model when the coupling coefficients are larger than 1 should be investigated. Therefore

new MadGraph files have been processed for the following configuration:

$$\begin{aligned} \text{Re}(V_L) &\in [0.7, 1.3] \\ \text{Re}(V_R) &\in [-0.3, 0.3] \end{aligned}$$

These files can be found in the following directory on the m-machines and contain 100 000 events.

*/user/aolbrech/AnomalousCouplings/MadGraph_v155/MassiveLeptons/
MadGraph5_v1_5_5/Wtb_ttbarSemiElMinus/ResultsXSGrid003*

2.3.4 Width of the decay

2.4 FeynRules

Should be understood why the mass of the top quark is equal to 180 within the created FeynRules model!

Chapter 3

Generator level bottlenecks

The goal is to obtain as fast as possible results on generator level in order to ensure that no bottlenecks are found when running MadWeight. The advantage of only using generator level results is that one can be completely sure that the studied events are actual semi-leptonic $t\bar{t}b\bar{a}$ events. Hence MadWeight should not have any problems calculating the weight for these kind of events and no CPU time will be spent on uncorrect events. This implies that any deviation from the expected results implies a bias, or even a problem, concerning the MadWeight output.

Once the results correspond to the expectations these preliminary results should be easily extended to reconstructed events. Finalizing the event selection then allows to fully trust the results obtained on reconstructed level and make sure that any deviation should be explained by the influence of the applied event selection.

These results can then be used to optimize the event selection with respect to the MadWeight output and CPU time needed.

3.1 Transfer Functions

In order to obtain results with MadWeight, the Transfer Functions which link the reconstructed energy distribution with the actual energy distributions should be calculated. In the case of generator level events this is not that important since no smearing of the energy is expected, but in order to avoid any bias from the used Transfer Functions it is advised to use the real Transfer Functions with a smaller width.

The method to obtain the parameters describing the energy smearing is (partly) explained in the PhD Thesis of Arnaud Pin¹ which can be found in:

*https://cp3.irmp.ucl.ac.be/upload/theses/phd/Pin_Arnaud.pdf
/home/annik/Documents/Vub/PhD/ThesisSubjects/AnomalousCouplings/
PrepareGenLevelRunning_Sep2014/TransferFunctions*

The method to obtain the parameters of the Transfer Functions is based on the code

¹It should be noted that in the PhD Thesis of Arnaud, and in the current double Gaussian transfer function syntax in MadWeight, only 5 parameters are used. The narrow gaussian distribution doesn't have a normalisation parameter in front, but is normalized afterwards in the MadWeight code.

received from Petra and Lieselotte, which was used in the Master thesis of Lieselotte². However it should be noted that the original code only had 4 bins due to limited statistics, and now 10 bins are used.

This code is almost identical to the ROOT class `FitSlicesY()` but has some minor differences. Some of these have already been changed in order to match better with the ROOT class.

One of the most important differences between the two approaches was the treatment of the underflow and overflow bin. In the received code these two bins were respectively added to the first and last bin and hence included in the fitting range. This is not the desired behavior since the size of underflow/overflow bin can be relatively large compared to the first/last bin and significantly change the value of these bins. This would then imply that the position of the first and last bin is not located at the correct position and will, especially in the case of a limited number of bins, have a significant influence on the fit result. Now these underflow and overflow bins are just discarded from the fit range and will have no influence on the final result.

Another important, but useful, difference between the two methods is the number of histograms which are saved. The received code saves for each distribution which is considered the `ProjectionY` distribution together with the double Gaussian fit for this bin. This can not be changed in the ROOT class which only stores the distribution of the 6 parameters of the double Gaussian fit formula.

However, even after carefully ensuring that both methods are identical the obtained results are not. Up to now it is not clear what is the reason for the discrepancy between the two results and the only way to found out is comparing the distributions and results for a significant amount of statistics.

One possibility is the used fit ranges and number of bins. If the last bins are low on statistics their distribution might not agree with a double Gaussian distribution and hence result in a failed fit. Therefore the distribution for each `ProjectionY` bin is now closely studied for all of the considered histograms. This can be found in the following section 3.1.2.

3.1.1 Creating the Transfer Function

In this subsection the different steps which were performed in order to built the actual Transfer Functions and implemented in `MadWeight` will be discussed in detail. The Transfer Functions used in this analysis are assumed to be uncorrelated, as shortly discussed in the PhD thesis of Arnaud. This assumption is necessary since otherwise it would be impossible to build them ourselves. The implication of this assumption is given in Equation 3.1 and should be checked by looking at E vs θ , E vs ϕ and θ vs ϕ histograms.

$$W(E, \theta, \phi) = W(E)W(\theta)W(\phi) \quad (3.1)$$

A first important remark which should be made is the choice for using p_T dependent Transfer Functions, in stead of the general E -dependent Transfer Functions already implemented in `MadWeight`. The changes which have to be done in order to correctly

²Additional feedback was also received from Arnaud, but he was using a completely different unbinned likelihood method. Since this only arrived after a couple of weeks waiting, it was decided to continue with the binned likelihood method used by Petra and Lieselotte.

implement the Transfer Function in MadWeight are discussed in 3.1.1.1.

The following equation, Equation 3.2, gives explicitly the functional form of both the double Gaussian fit and the p_T -dependent calorimeter formula. As can be seen from this equation, the calorimeter part should not be taken to literally since it is only the actual calorimeter p_T dependency in the case of the σ -parameters of the two Gaussian fits.

$$\begin{aligned}
& a_3 * e^{-\frac{(x-a_1)^2}{2*a_2^2}} + a_6 * e^{-\frac{(x-a_4)^2}{2*a_5^2}} \\
& a_{x,0} + a_{x,1} * \sqrt{p_T} + a_{x,2} * p_T \quad (\text{for } x = 2 \ \& \ 5) \\
& a_{x,0} + a_{x,1} * p_T + a_{x,2} * p_T^2 + a_{x,3} * p_T^3 + a_{x,4} * p_T^4 \quad (\text{for } x = 1, 3, 4 \ \& \ 6)
\end{aligned} \tag{3.2}$$

3.1.1.1 Using p_T dependency

The Transfer Function configuration files in MadWeight are flexible enough to change the kinematic variables used for the Transfer Function calculations. Hence it seems to be more relevant to utilize the transverse momentum in stead of the energy of the considered partons and jets. Especially since the LHCO file used in the MadWeight calculations has the transverse momentum as input variable. Therefore it seems much more useful and realistic to use this parameter and not the energy. This implies that the Transfer Function configuration file should be adapted to use the $pt(p)$ and $pt(perp)$ variables and not the currently used $p(0)$ and $perp(0)$.

After actually implementing the created Transfer Function files³ and trying to use this Transfer Function one issue showed up which is currently not completely solved yet. MadWeight contains hard-coded files which only allow the use of the E, THETA and PHI variable for the Transfer Function dependencies. Therefore the so-called “block name” PT is not accepted by the *change_tf.py* file. However it seems that there is no clear physical motivation for this limited choice of kinematic variables, but in order to be completely sure additional information is needed from Olivier and Pierre.

Once this issue is resolved the following file, containing the restricted list of kinematic variables, should be adapted⁴:

bin/internal/madweight/change_tf.py

3.1.1.2 Creation of the Transfer Functions

The Transfer Functions are created from the simulated $t\bar{t}$ sample which will be used throughout the entire analysis. A very small nTuple is created from this simulated sample containing only the TLorentzVector information of both generated and reconstructed particles. From this the necessary diagrams, such as the 2D distributions of p_T , θ and η difference between the generated and reconstructed particle with respect to the generated value, are created and saved in a ROOT file:

AnomalousCouplings/TFInformation/PlotsForTransferFunctions_FromTree.root

³This was tested the first time on 30 November 2014.

⁴Another solution is to keep the block name equal to E but just use the transverse momentum information. However this might result in confusion when in some files the hard-coded E parameter gets written down (for example on plots).

This analyzer, called *TFFit.cc*, also performs the double Gaussian fit of these 2D histograms and afterwards the E -dependent calorimeter fit. The technicalities and specific details of these fit procedures are documented in the following class:

AnomalousCouplings/PersonalClasses/src/TFCreation.cc
AnomalousCouplings/PersonalClasses/interface/TFCreation.h

The results of the two consecutive fits performed on the Y-projections of these 2D distributions are stored in another ROOT file, together with the original 2D histograms. Also the function form of the double Gaussian fit formula using the obtained fit parameters, added in order to test the robustness of the fit results outside the fitted range, can be found in this ROOT file.

AnomalousCouplings/TFInformation/CreatedTFFromDistributions_FromTree.root

This analyzer also has the flexibility to perform the fit on the entire range or on pre-defined ranges set by the user. For this a separate function, called *SetFitRange*, is created where for each histogram the fit range for each separate bin can be defined. This is extremely useful to optimize the doubleGaussian fit which has to cover both the peak and the tail of the distributions in order to correctly calculate the 6 fit parameters. Another useful aspect of this analyzer is the automatic creation of the necessary .dat Transfer Function files needed for implementation in MadWeight. This is done in the *WriteTF* class and the created files are:

AnomalousCouplings/TFInformation/TF_user.dat
AnomalousCouplings/TFInformation/transfer_card_user.dat

This first file contains the functional form of both the E -dependent calorimeter fit and the functional form of how these 6 parameters should be included in the double Gaussian formula. Also the width for the different kinematic variables is defined within this file. For the moment the method used in the Transfer Functions already implemented in MadWeight is followed, implying that the width of the Transfer Function is defined as the maximum of the σ -parameter of the two Gaussians considered in the double Gaussian fit. The only difference is that instead of the generated kinematic information, which is the variable on the abscissa of the considered 2D histograms, the reconstructed one is used.

The second file contains the actual values of the different fit parameters for all the considered kinematic variables and particle types. Therefore this file is an extensive list of values to which is referred in the previous *TF_user.dat* file. For each particle type and kinematic variable the numbering used should be unique such that the correct values are implemented in the functional forms of the used fit formulas.

3.1.1.3 Importance of start values

Since the double Gaussian fit really needs accurate information of both the peak and the tails, detailed review of the start values for each of the 6 fit parameters significantly improves the success rate of the fitting method. However it is important to note that setting these start values clearly improves the results obtained from the fit so special care should be taken to ensure the correctness of these start values. One way is by intensively comparing the fit distributions for the different particles since in the case of light jets and

b-jets some similar behavior is expected.

After quite a while it is possible to quickly see whether the fit distribution has the expected shape and hence whether the used start values can be considered as stable. The start values used for the different 2D histograms is given in Table 3.1

2D histogram	Used start value for fit parameter					
	First (narrow) gaussian			Second (wide) gaussian		
	Mean a_1	Sigma a_2	Amplitude a_3	Mean a_4	Sigma a_5	Amplitude a_6
b-jet $\Delta\phi$	0.0002	0.022	8000	0.0002	0.06	3000
b-jet Δp_T	10	-12	20000	13	10	-5000
b-jet $\Delta\theta$	0	0.013	6000	0	0.04	2000
light jet $\Delta\phi$	0	0.022	8000	0.0004	0.002	3000
light jet Δp_T	0	8	4000	0	12	4000
light jet $\Delta\theta$	0	-0.014	6000	0	-0.05	2000
electron $\Delta\phi$	0	0.0012	1500	0	0.006	600
electron Δp_T	0	0.9	1500	0	-2	600
electron $\Delta\theta$	0	0.0013	2500	0	0.007	600
muon $\Delta\phi$	0	0.0004	800	0	0.0026	600
muon $\frac{1}{\Delta p_T}$	0	0.0003	2000	0	0.0006	500
muon $\Delta\theta$	0	0.002	500	0	0.0004	500

Table 3.1: caption ... Need to make sure that the narrow and wide gaussian is always the same ... Otherwise are the start values not correct for the different eta-bins ...!!

Need to look at the sigma value to decide on the narrow/wide gaussian!

3.1.1.4 Splitting in separate $|\eta|$ bins

Since the kinematic variables tend to depend on the pseudorapidity η the considered 2D histograms are created for four distinctive $|\eta|$ regions. It has been chosen to split the barrel region into three separate bins while the entire endcap region is contained within one single bin. The chosen binning is given in Table 3.2 together with the percentage of events present in each of the considered $|\eta|$ bins. This clearly shows the lower statistics available in the endcap region which results in larger difficulties of properly reconstructing the fit parameters in this region. This is shortly discussed below.

	$ \eta \leq 0.375$	$0.375 < \eta \leq 0.75$	$0.75 < \eta \leq 1.45$	$1.45 < \eta \leq 2.5$
Relative # events	26.21 %	23.91 %	32.54 %	17.34 %

Table 3.2: Different $|\eta|$ bins used for the Transfer Function creation. It is important to note that the η -values used for this binning are the reconstructed ones since generated values could be higher than the 2.5 cut-value.

The analyzer mentioned above is developed in such a way that both the fit results for all events as the results for the four separate $|\eta|$ bins are stored together. Therefore all the results can always be compared in the created ROOT files and in the distinct *dat* files.

One important difference between the 2D-distributions containing all events and the

2D-distributions specific for one of the $|\eta|$ bins is the number of bins used. Since the statistics is significantly lower for the $|\eta|$ specific histograms, the predefined bin number is lowered with 25%. This ensures a more stable tail for the distribution and still a correct reconstruction of the peak.

However for the last $|\eta|$ bin considered, the endcap part, a slightly different method is used. Since the statistics is much lower in this part of the detector the used range had to be slightly stretched in order to ensure a nice overview of the tails. This is necessary for the correctness of the double Gaussian fit. Therefore the range of the abscissa is enlarged on both sides with 20% and the used number of bins for this axis is identical to the one used for the overall 2D-distribution.

3.1.1.5 Separating narrow and wide gaussian

Remark: Should check whether this is actually necessary for the MadWeight implementation. MadWeight only seems to need the general fit formula and doesn't need to know which of the two distributions is the narrow and which is the wide one.

In order to select the narrow and wide gaussian both the amplitude and the σ -parameter should be compared, but keeping in mind that this second one is the most important variable. However it is expected that the distribution with the narrowest distribution also has the highest peak. In order to easily compare the two distributions a stacked canvas is added to the ROOT file which shows both distributions for different $p_{T,gen}$ values in one stacked canvas. The distinction between narrow and wide gaussian distribution is currently being made by the size of the σ variable. Hence the histogram with the narrowest distribution is plotted in red while the widest one is plotted in green.

Such a canvas is made for each of the considered 2D-histograms, both for the one containing all events and for the ones splitted in separate $|\eta|$ bins. They should be analyzed in detail in order to understand the correctness of the double Gaussian fit applied for the creation of the Transfer Functions. For the moment there is still a couple of 2D-histograms which don't show the expected behavior. Additional investigation of these stacked canvases should be performed as soon as possible.

3.1.1.6 Coping with low statistics in last $|\eta|$ bin

Solution : Excluding bins! + combining bins

3.1.1.7 Extrapolation method

The implementation in MadWeight should be done in such a way that the value of the outermost bins is used for all the p_T values outside the fitted region. This is necessary since the extrapolation using the obtained fit parameters doesn't result in the desired double Gaussian behavior for these p_T values. For values outside the fitted region, an inverted double Gaussian distribution or a distribution with two distinct peaks occurs rather often. This can be seen in Figure 3.1

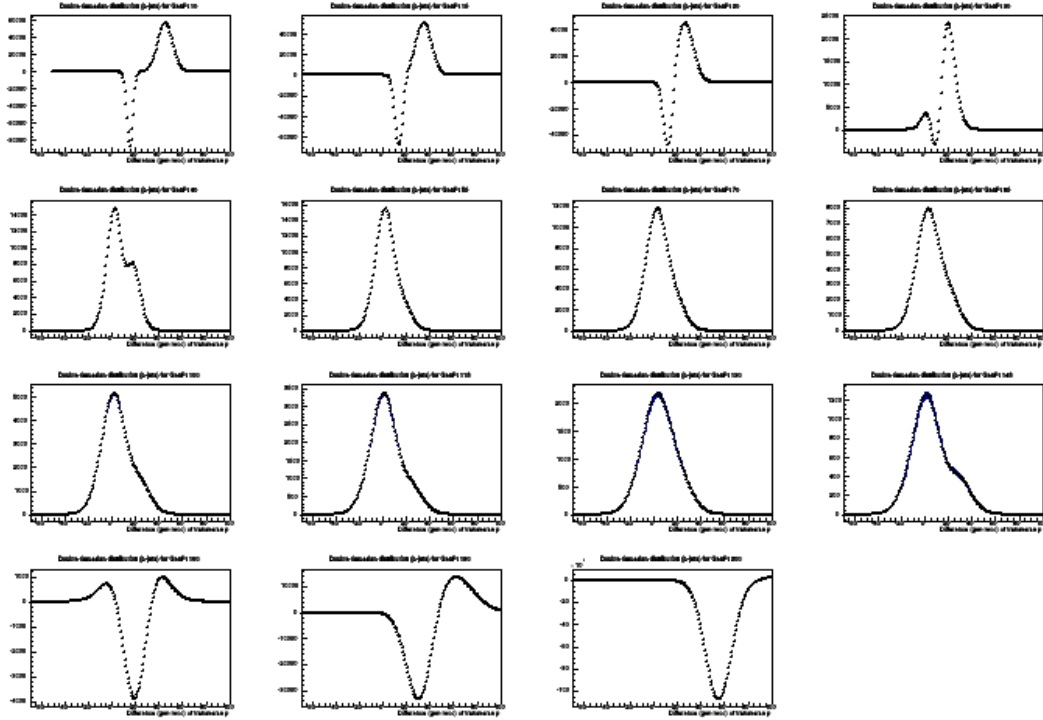


Figure 3.1: Extrapolation obtained using the fit parameters of the double Gaussian functional form. Values outside the fitted range show a distinctly different shape with respect to the ones actually fitted. *Maybe consider to lower the number of bins and make the titles more visible (Only p_T cut value is really important!)*

3.1.1.8 Actual implemenation in MadWeight

All the possible Transfer Functions which are implemented in MadWeight can be found in the *Source/MadWeight/transfer_function/data* directory. Any file can be added to this list and used within MadWeight.

The relevant files used for the creation of the Transfer Functions are given below. The first one is the translation of the used *TF_user.dat* into a MadWeight readable file which can be implemented. The second file is only relevant around line 292 where the function used for the Transfer Function creation is explained. This is the general MadWeight constructor file where all the different MadWeight functions are defined.

Source/MadWeight/transfer_function/transfer_function.f
bin/internal/madweight_interface.py

The commands which have to be executed in order to build this MadWeight readable file for the Transfer Function are given here:

./bin/mw_options
define_transfer_fct

3.1.2 Obtained distributions

Each of the considered histograms is a 2D histogram where the abscissa represents the transverse momentum of the generator level parton and the ordinate the difference between the generator level parton and the reconstructed matched particle. This is done

for the difference in transverse momentum and in θ and ϕ angles.

All the interesting histograms can be created automatically, for each of the desired $|\eta|$ bins separately, using the following ROOT analyzer:

AnomalousCouplings/TFInformation/FitDistributions/SaveFitHistograms.C

This analyzer is able to create each of the histograms separately, both as *pdf* and *png*, and automatically saves all the histograms of one type in a large stacked canvas. This allows to quickly see the used 2D distributions for each of the particle types (b-jets, light jets, electrons and muons) and the three kinematic variables (p_T , η and ϕ). This stacked canvas is shown in Figure 3.2. Also the general behavior of each of the fitted diagrams together with the overall χ^2 distribution is created for each 2D histogram as can be seen from Figure 3.3, showing this for the p_T distribution of the b-jets. Finally the distribution of the 6 double Gaussian fit parameters together with the fit result of the E -dependent calorimeter formula is also collected in a stacked canvas, as can be seen in Figure 3.4.

The 2D distributions for the difference in transverse momentum tend to show a slightly

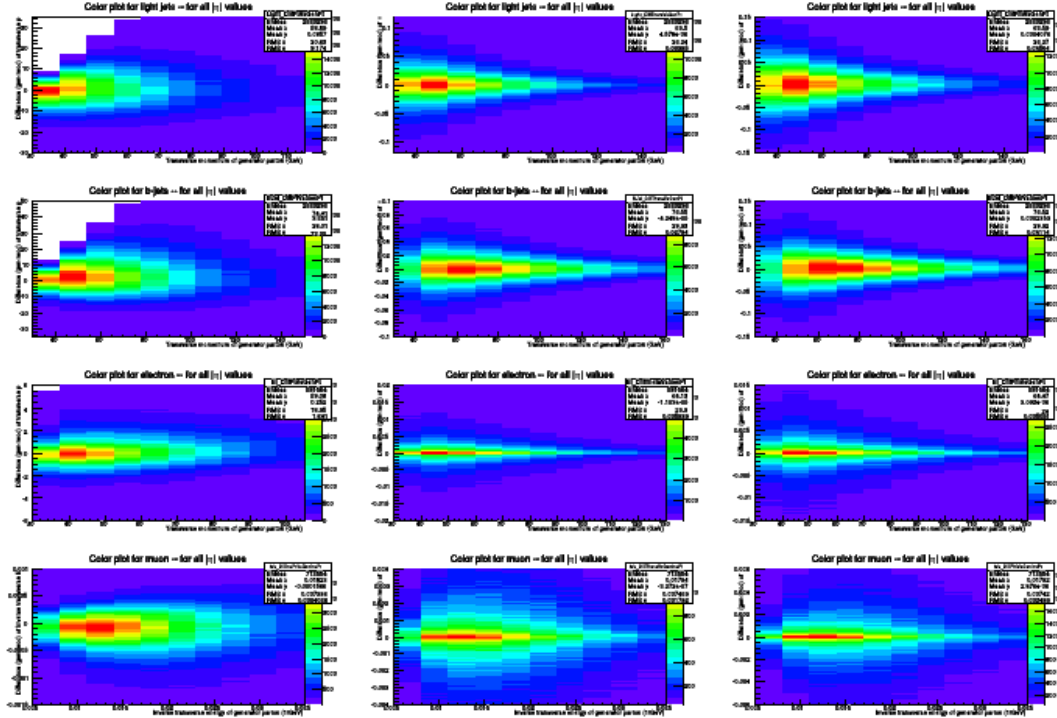


Figure 3.2: Used 2D distributions for double Gaussian fit. **IMPROVE CAPTION!**

asymmetric behavior, as can be seen from Figure 3.2. This can be explained by the influence of the event selection, which has a difference effect on the generated particle than the reconstructed particle. This because a particle surviving the p_T cut actually has a different p_T value on generated level due to **bad resolution, detector effects (???)**. This effect is almost negligible for the ϕ and θ angles (**Definitely sure that this is the case ??**).

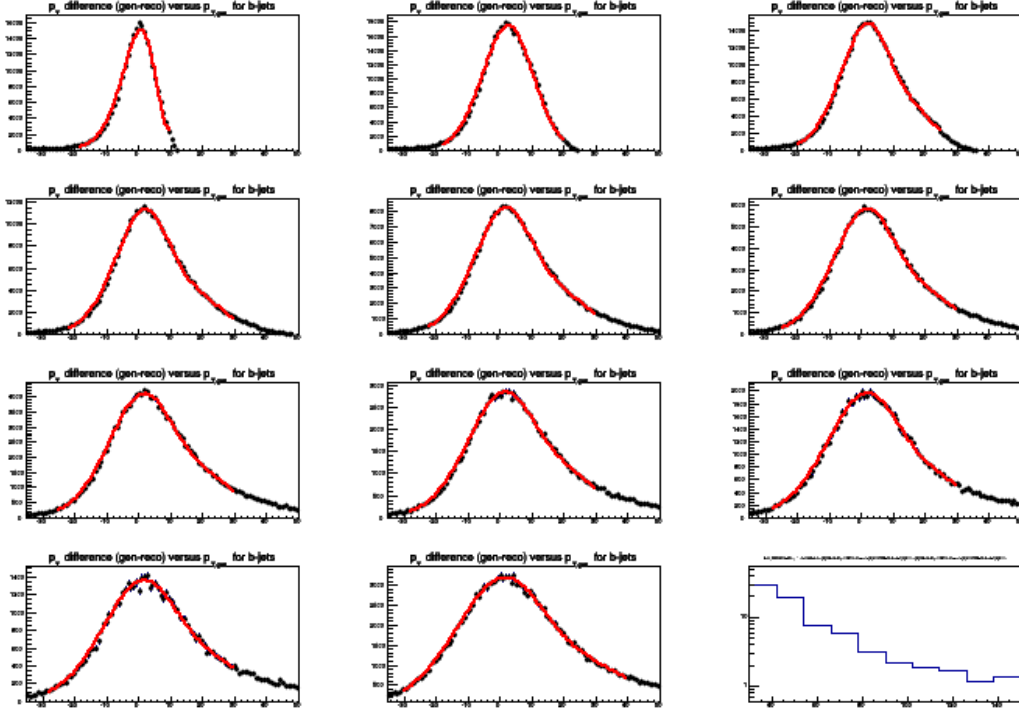


Figure 3.3: Distribution of the energy difference between the generator level parton and the corresponding light quark jet for each of the 10 considered bins and the overflow bin. All distributions were fitted with a double Gaussian function.

STILL TO REVIEW

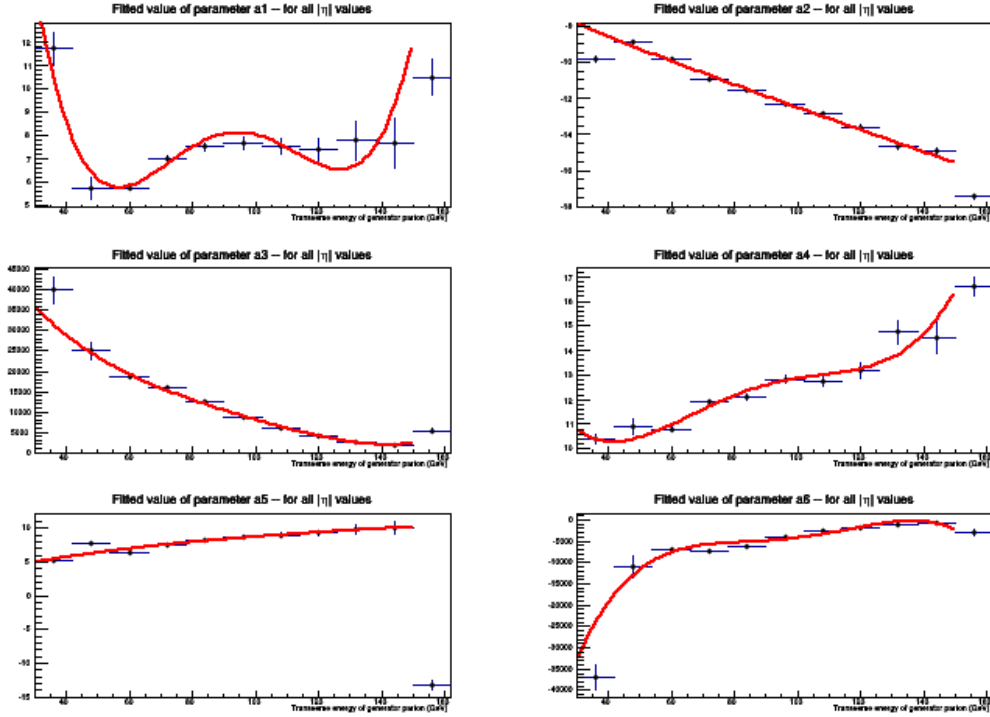


Figure 3.4: Energy dependency of the 6 parameters of the double Gaussian fit function. The result of the double Gaussian fit for each p_T bin is combined in a p_T dependent histogram and then fitted with the Calorimeter energy function as explained in the PhD Thesis of Arnaud Pin.

3.1.3 Control checks for Transfer Functions

3.1.3.1 Compare results with normal (single) Gaussian

3.1.3.2 Compare result with previous analyses

This will imply to put $p_{T, reco}$ information on the abscissa in stead of the current $p_{T, gen}$ one. Also it will result in a huge change of the start values which are however necessary to perform a succesful double Gaussian fit.

3.1.3.3 Compare results with predefined ROOT class

3.2 Cross Section distribution for new grid

Remark: Cross section comparison

Need to check how it is possible that the XS values in the XS grid are identical for both versions (first check whether this is indeed the case), but are different for the top quark mass simulation ... Table 3.4.

3.3 First results: Wrong log(likelihood) minimum

The first obtained MadWeight results for the enlarged grid ($V_L \in [0.8, 1.2]$ and $V_R \in [-1, 1]$) using only parton-level ttbar events did not result in the expected minimum of $(V_L, V_R) = (1, 0)$. This can be seen from Figure 3.5, which shows the distribution of the log(likelihood) for each point in the considered grid.

One of the possible influences on the displaced minimum of the log(likelihood) distribu-

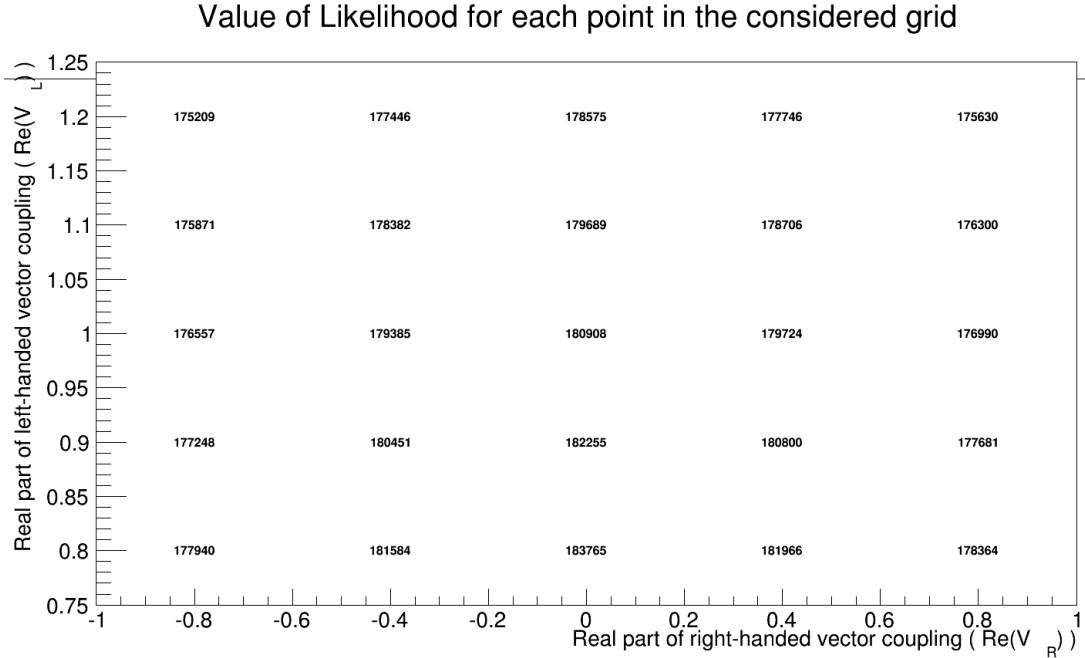


Figure 3.5: Distribution of the log(likelihood) for each point in the grid using 3200 parton-level positive semi-muonic ttbar events. The transfer function used to smear the parton-level kinematics is the single-gaussian function standard included in MadWeight.

tion could be the normalisation of the Cross Section influence. This XS normalisation ($\frac{XS}{XS^{SM}}$) should be multiplied with the likelihood value, not the log(likelihood). Hence in order to correctly take this into account the obtained log(likelihood) value for each point in the grid should be corrected using the logarithm of this XS normalisation. The distribution of the normalisation on the Cross Section can be found in Figure 3.6 together with the log(likelihood) distribution after correctly taking into account this XS normalisation.

The formula which has been used is the following (Equation 3.3):

$$P(y|a) = \frac{1}{\sigma(a) * Acc(a)} \int W(y|x, a) * Eff(x, a) |M(x, a)|^2 T(x, a) dx \quad (3.3)$$

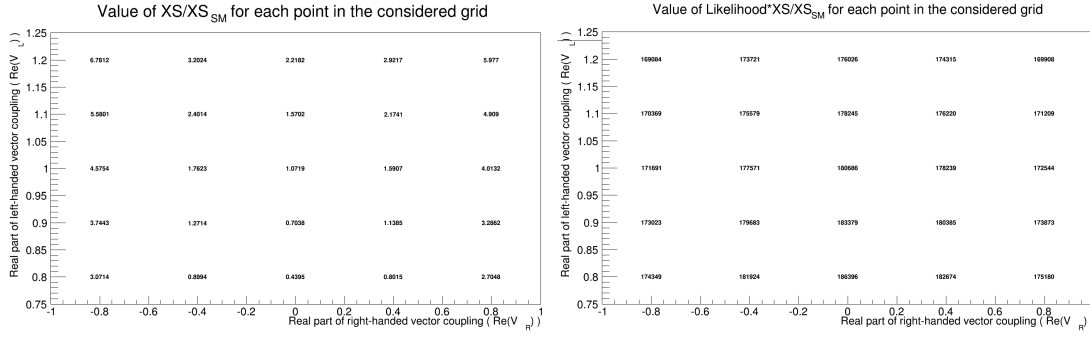


Figure 3.6: Distribution of the XS normalisation for positive semi-muonic $t\bar{t}b\bar{b}$ events (left) and distribution of the $\log(\text{likelihood})$ after taking into account this normalisation. As in the previous figure 3200 positive semi-muonic have been used to obtain this distribution and a single gaussian transfer function has been applied to smear the kinematics of these parton-level events.

$$\mathcal{L} = \prod P(y|a) \quad (3.4)$$

The normalisation which has been applied in Figure 3.6 is given in Equation 3.5:

$$\mathcal{L}_{Norm} = -\ln(\sum P(y|a) * \frac{XS}{XS_{SM}}) = -\ln(\mathcal{L}) - \ln(\frac{XS}{XS_{SM}} * N) \quad (3.5)$$

It should be checked whether this is the correct method to take into account the normalisation of the XS. Currently it has been assumed that this XS normalisation should be applied for each weight and hence is multiplied with the number of considered events. In the case that this normalisation should just be multiplied with the overall likelihood value (\mathcal{L}) the sum over the number of considered events drops out of the equation implying a very small influence of the XS on the $\log(\text{likelihood})$ distribution.

As highlighted in the general Matrix Element Method formula (Equation 3.3), the probability to measure the observed quantities y already has a normalisation factor for the cross section. This factor is defined as the channel cross section and is calculated using Equation 3.6:

$$\sigma(a) = \int_{X_i} |M(x, a)|^2 T(x, a) dx \quad (3.6)$$

Hence it should be investigated in detail whether this XS normalisation should still be applied. From the above equations could be concluded that the change in cross section is actually already incorporated in the weight obtained from MadWeight. This could make sense since MadWeight has all the necessary information to calculate the cross section for each point in the considered grid. The cross section values for each point have been calculated using MadGraph and the same model as used for the MadWeight calculations. Unfortunately it is not completely clear from the MadWeight documentation whether this is actually included in the weight or not.

3.4 Correct normalisation of Matrix Element probability

As could be seen in Equation 3.3 a term $\sigma(a)$ is included in the general Matrix Element Techniques formula. However it is not clear whether this cross section normalisation is actually performed within the MadWeight calculations or whether this normalisation should be done afterwards.

This question is closely related to the order of the current obtained weight values with MadWeight. Up to now no weight larger than 10^{-22} have been obtained, resulting in a very large $\log(\text{likelihood})$ value.

This small value can be caused by many different reasons for which the most plausible ones are listed here:

- The normalisation of the MadWeight probability should still be done and is not performed within the Matrix Element Techniques formula.

This can only be ruled out by contacting the Madweight experts and asking explicitly what is done in the Madweight calculations. Also a possible hint could be found inside the MadWeight python files (but this should only be done if the received answer is not perfectly clear).

- The smallness of the weight could be caused by an error inside the created FeynRules model. *Should also look for the mail where one of the MadWeight experts (Olivier/Pierre or even Celine) answered about the possible explanation for the smallness of the weight and whether this implies some wrong assumptions).*

A possible way to exclude that the origin of this problem is the AnomalousCouplings FeynRules model is by comparing the results for the top mass fit when both the SM FeynRules model and the AnomalousCouplings model is used. If the weights are also this small when the SM model is used this smallness should be solved by an additional normalisation factor.

Update 31/10/2014: Probability function NOT normalized (according to mail Olivier)

As was expected from the smallness of the obtained MadWeight probabilities should the cross section normalisation be applied afterwards. Only in the older versions of MadWeight (based on MG) was this normalisation included automatically.

3.4.1 Measurement of top quark mass using Matrix Element Method

Comparing SM model with AnomalousCouplings model

As a first step the Feynmann diagrams belonging to the two different models should be compared. This information can be found in the *index.html* file in the following directories (and the files should be opened using firefox on mtop since this is the only m-machine with a working browser):

/AnomalousCouplings/MadGraph5_aMC@NLO/madgraph5/SM_ttbarSemiMuPlus
/AnomalousCouplings/MadGraph5_aMC@NLO/madgraph5/ttbarSemiMuPlus_QED2

Comparing SM cross section with MassiveLeptons cross section

In order to be sure that both models have the same Standard Model base, the cross sections for both models have been compared. This resulted in an unexpected outcome, namely that the obtained cross sections differ significantly depending on which MadGraph version is used to generate the considered events. A summary can be found in Table 3.4.

Top quark mass	MadGraph aMC@NLO		MadGraph v155	
	SM model	MassiveLeptons model	SM model	MassiveLeptons model
153	9.23 pb	9.645 pb	6.692 pb	6.984 pb
163	11.12 pb	11.63 pb	7.844 pb	8.199 pb
173	12.98 pb	13.54 pb	8.897 pb	9.281 pb
183	14.77 pb	15.4 pb	9.884 pb	10.3 pb
193	16.5 pb	17.22 pb	10.78 pb	11.25 pb

Table 3.3: Cross section values for semi-muonic (+) ttbar decay obtained using two different MadGraph versions.

From this table can be seen that there is, for both considered MadGraph versions, a small difference between the SM FeynRules model and the MassiveLeptons one. This could be caused by the different treatment of the leptons. In the SM model they are considered to be massless while in the MassiveLeptons one they are defined to have their actual mass.

A larger difference occurs when both MadGraph versions are compared. From the answer received by Olivier it is not clear whether this difference is worrisome or could be explained by the LO theoretical uncertainties. Should also be investigated whether this difference is related to the NLO behavior of the newest MadGraph version. In case the MadGraph v155 version is not up to NLO a difference in cross section is definitely expected.

Understanding why Top Mass simulation does result in correct Likelihood minimum

When using the *MadGraph5_aMC@NLO* MadWeight version is used to scan over the different top mass values the correct minimum is obtained directly, so without any normalisation or acceptance influences. This simulation only uses the Standard Model information and doesn't consider the AnomalousCouplings FeynRules model. The results can be found in Table ??

Top quark mass	MadGraph aMC@NLO MadWeight log(Likelihood)
153	196 499
163	188 002
173	181 027
183	186 284
193	192 926

Table 3.4: caption ..

3.4.2 Influence of the acceptance term

In order to finalize the normalisation of the weight obtained from MadWeight also the influence of the acceptance term $Acc(a)$ has to be investigated. Since no reconstructed events exist are created for the different vector coupling coefficients, the influence of the applied event selection can only be applied on generator-level. This will normally introduce a slight bias since applying the reco-level cuts on generator-level will result in more stringent cuts than actually applied on the reconstructed events due to the smaller width of the kinematic distributions. However it is the only way to get an idea of the influence of the event selection and, as long as a flat dependency is found throughout the vector couplings, an acceptable one.

The event selection influence is investigated by looking both at the change in cross section and at the difference of different kinematic distributions. This first is achieved by simulating events with different vector coupling coefficients and analyze the change in cross section. This is done for 1-dimensional scans of both the left-handed as right-handed vector coupling coefficients while the other is set to its Standard Model expectation value. The results together with the procentual reduction in cross section by the application of the event selection is given in Table 3.5 and Table 3.6 for the 1D change of V_L and V_R , respectively. All the created MadGraph files are located in the following directory:

MadGraph5_aMC@NLO/madgraph5/TopMassCheckQED2_ttbarSemiMuPlus_ML

Top quark mass	1D change of $Re(V_L)$		
	All events	Reco p_T cuts applied	Reduction (%)
(0.8, 0.0)	3.62605 pb	0.9423 pb	25.99
(0.9, 0.0)	5.81248 pb	1.51 pb	25.98
(1.0, 0.0)	8.85979 pb	2.30454 pb	26.01
(1.1, 0.0)	12.96357 pb	3.37064 pb	26.00
(1.2, 0.0)	18.3674 pb	4.768 pb	25.96

Table 3.5

Top quark mass	1D change of $Re(V_R)$		
	All events	Reco p_T cuts applied	Reduction (%)
(1.0, -1.0)	37.7415 pb	11.98 pb	31.74
(1.0, -0.5)	14.5606 pb	4.466 pb	30.67
(1.0, 0.0)	8.85979 pb	2.661 pb	30.03
(1.0, 0.5)	13.1236 pb	4.04 pb	30.78
(1.0, 1.0)	33.1415 pb	10.61 pb	32.01

Table 3.6

From these Tables can be concluded that the influence of the event selection on the cross section is flat and equal throughout the entire vector coupling grid. This would be a positive result since it would imply that no additional analytical function is necessary to normalise the MadWeight output. Hence the MadWeight output should only be normalized for the Cross Section influence, but not the event selection one.

The second method of analyzing the influence of the event selection on the vector couplings is by comparing the kinematical distributions before and after the event selection is applied. Special attention is awarded to ensuring no difference in shape is observed for the different vector coupling coefficients after the event selection is applied. If this would be the case the influence of the event selection wouldn't be flat as suggested by the change in cross section. However, comparing all kinematic distributions indeed suggests that no significant change is observed when comparing the different kinematic distributions. The full list of distributions can be found in the directory listed below, but the ones for the $\cos\theta^*$ variable, the p_T distribution for the lepton, for the b-quark originating from the hadronically decaying top quark and for the down-quark originating from the W-boson decay are given in Figure 3.7, Figure 3.8, Figure 3.10 and Figure ??, respectively.

.../ThesisSubjects/AnomalousCouplings/KinematicDistributions_AcceptanceTerm_Dec2014

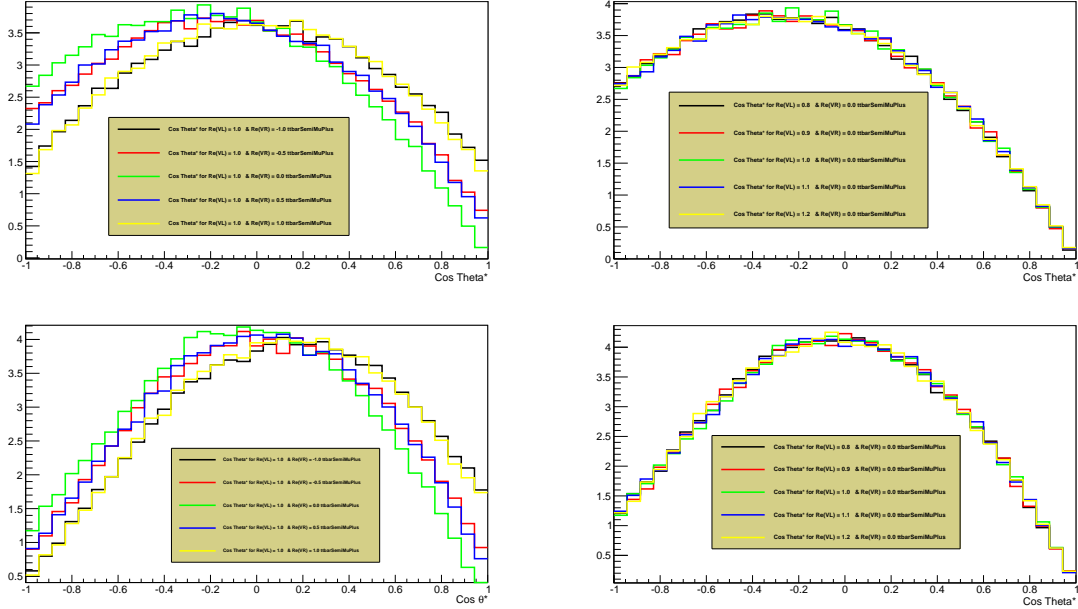


Figure 3.7: Distribution of $\cos\theta^*$ variable for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

From the distributions given above can easily be concluded that some shape difference are visible for the 1D-variation of the right-handed vector coupling. However for the left-handed vector coupling V_R no influence is visible, besides some minor statistical fluctuations, when varying the value of V_R . It is also clear that the influence of the V_R 1D-variation is not the same for each of the kinematical distributions, and even negligible for the p_T distribution of the b-jet originating from the hadronically decaying top quark. Since the shift of the distribution is different for the p_T distribution of the lepton and the down-type quark of the W-boson, the result is still in agreement with the flat change in cross section. This because the additional events for one distribution are balanced out by

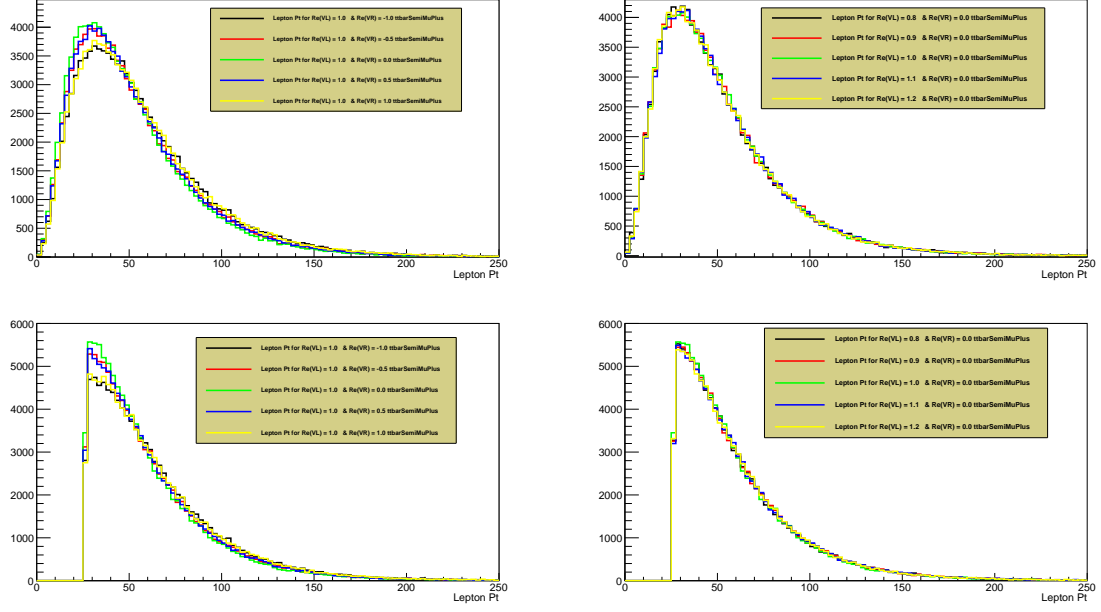


Figure 3.8: Distribution of transverse momentum of the lepton for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

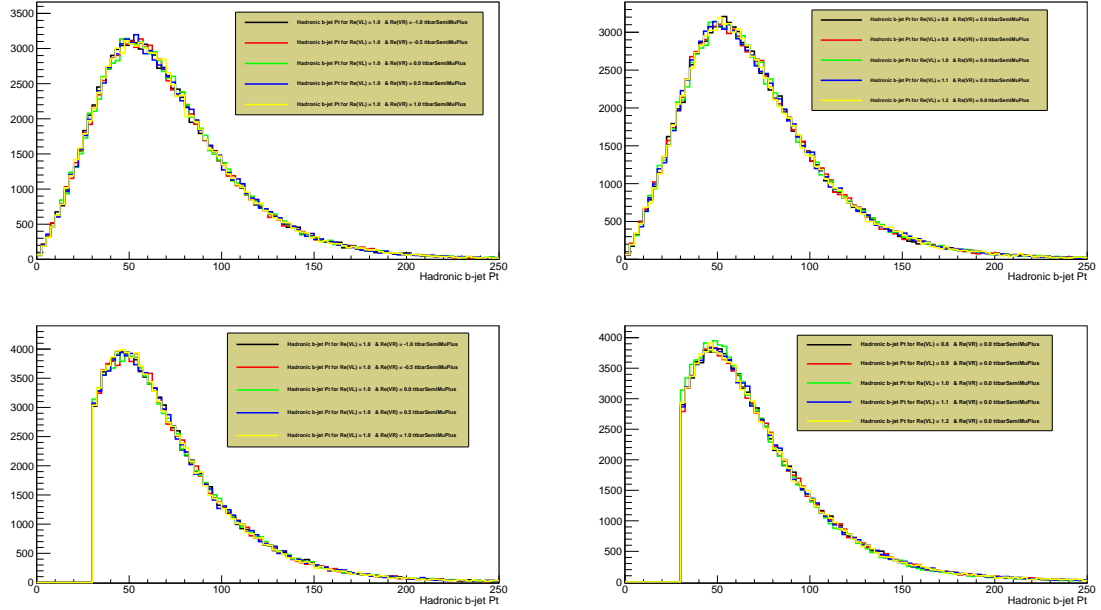


Figure 3.9: Distribution of the transverse momentum of the b-quark originating from the hadronically decaying top quark for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

the reduced number of events for another distribution resulting a net effect of zero and hence a flat behavior throughout the entire vector coupling coefficients grid.

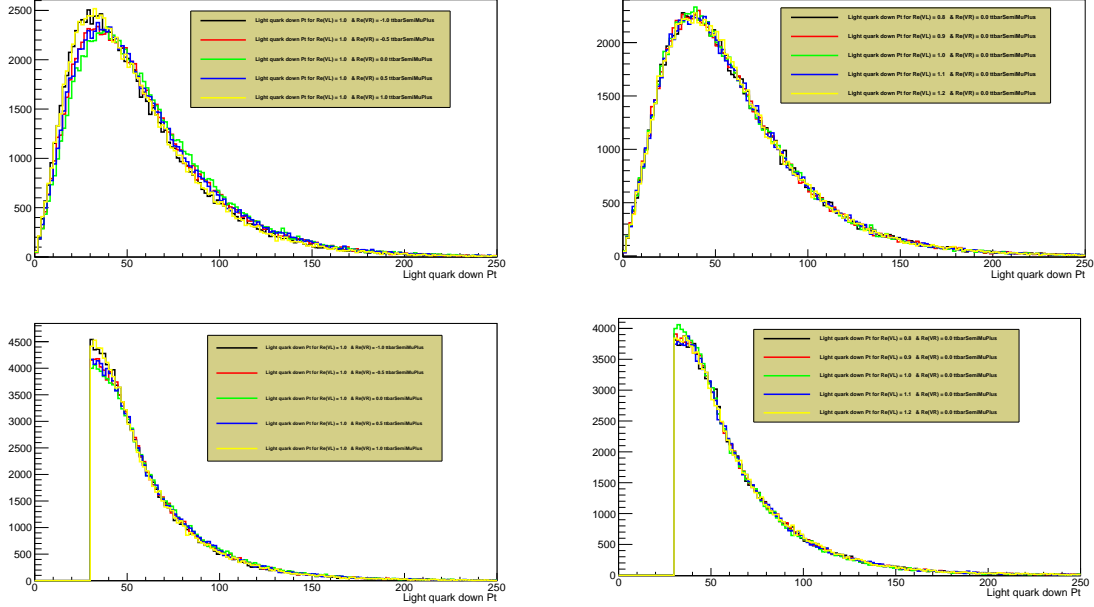


Figure 3.10: Distribution of the transverse momentum of the down-type quark originating from the W-boson for both 1D-variation of the vector coupling coefficients. The top figures depict the distributions before the application of any event selection while the lower ones show the same distribution but after the reco-level p_T cuts have been applied.

3.4.2.1 Understanding 1D-variation of V_L

The influence of the variation of the right-handed vector coupling V_R on the kinematic distributions was rather satisfactory and in some way agreeing with expectations. However the same is definitely not true for the variation of the left-handed vector coupling V_L . On the contrary, it can even be concluded that the obtained result was a complete surprise and needs to be understood as soon as possible in order to keep confidence in the created FeynRules model. This because a possible explanation can always be a wrong configuration of the anomalousCouplings FeynRules model which would result in a major setback for the analysis and the tight time schedule.

Therefore new MadGraph files, and kinematic distributions, have been created with the same 1D-variation of the left-handed vector coupling V_L but instead of setting the V_R value equal to its Standard Model expectation of 0 it was set to 0.2. The reason behind this different value for V_R is the idea that when the right-handed vector coupling is excluded from the full Wtb Lagrangian any change of V_L only affects the $|V_{tb}|$ value and hence the cross section. This would imply that no real change of physical concepts is done since the Lagrangian is in a sense unchanged because no mixing of the different vector couplings appears.

The results for this 1D-variation can be found in Table 3.7 and in Figure 3.12, and shown unfortunately identical results to the 1D-variation of V_L with V_R fixed to 0.

The obtained results for the 1D-variation of V_L with V_R fixed to 0.2 seems to suggest that either the left-handed vector component of the full Wtb Lagrangian only influences the cross section and alters in no way the kinematic distributions of the decay particles or otherwise that this component is wrongly implemented in the created FeynRules model. The following test which has been performed is looking at a larger 1D-variation of V_L

Top quark mass	1D change of $\text{Re}(V_L)$		
	All events	Reco p_T cuts applied	Reduction (%)
(0.8, 0.2)	4.223 pb	1.201 pb	28.43
(0.9, 0.2)	6.615 pb	1.882 pb	28.45
(1.0, 0.2)	9.995 pb	2.787 pb	27.88
(1.1, 0.2)	14.46 pb	4.043 pb	27.96
(1.2, 0.2)	20.26 pb	5.695 pb	28.11

Table 3.7

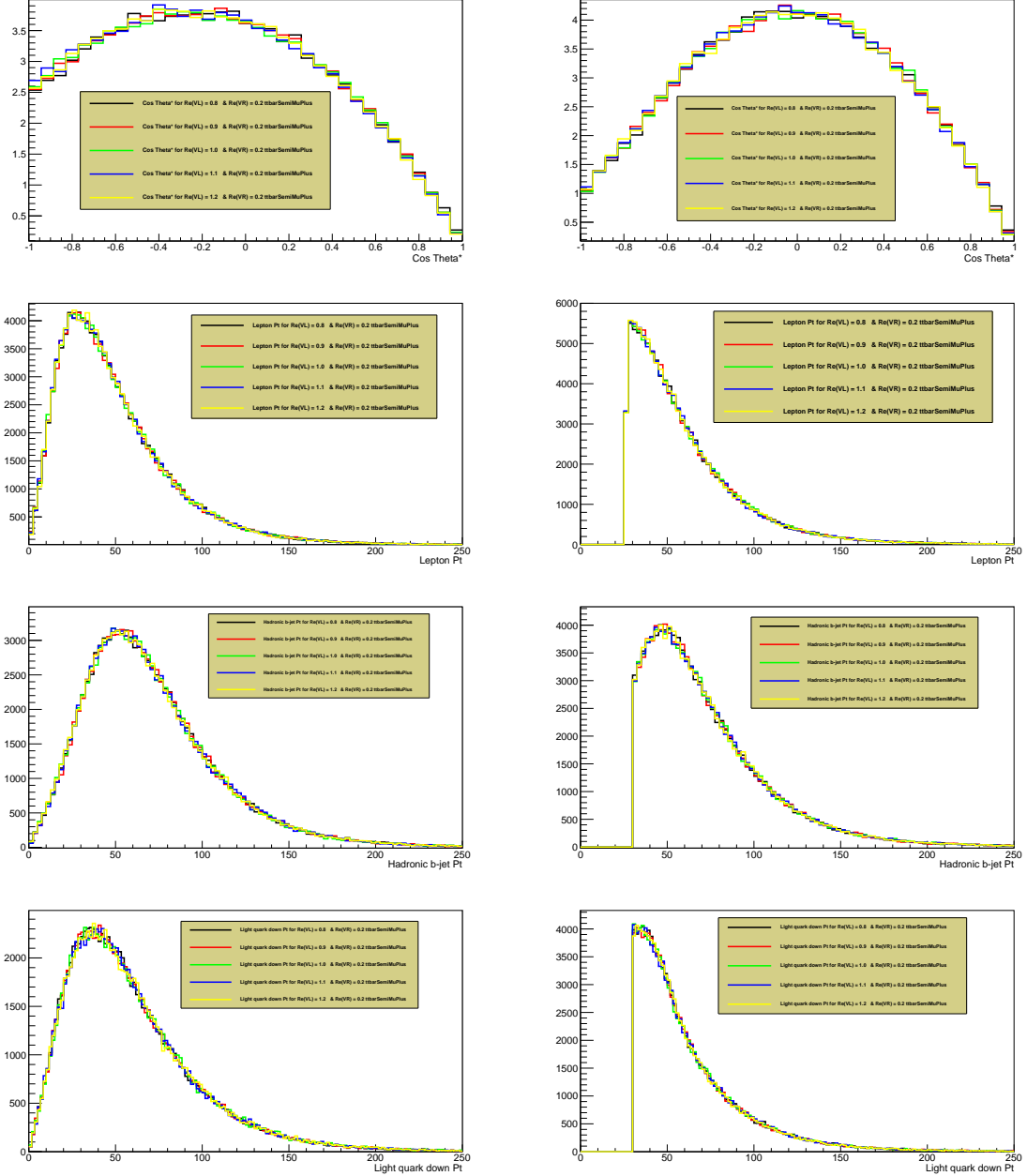


Figure 3.11

while still keeping the V_R component equal to 0.2. This resulted in a slightly unexpected outcome, as can be seen in Figure ??, since some deviation of the kinematic distribution

is found for the configuration where V_L is equal to 0. However there is no difference between the four remaining configurations considered for the large V_L 1D-variation. This is surprising since there was actually some hope that the lack of influence on the kinematic distributions when varying the V_L component could be caused by the small variation applied. This was motivated by the small influence when changing the V_R component to the boundaries of the coupling coefficients grid while originally the V_L coupling was contained within a small region due to its precise measurement. However the result seems to suggest that there is really no influence on the kinematic distributions and that only the V_R component is responsible for these distortions. Probably a similar result would have been found when a different value of V_R would have been used.

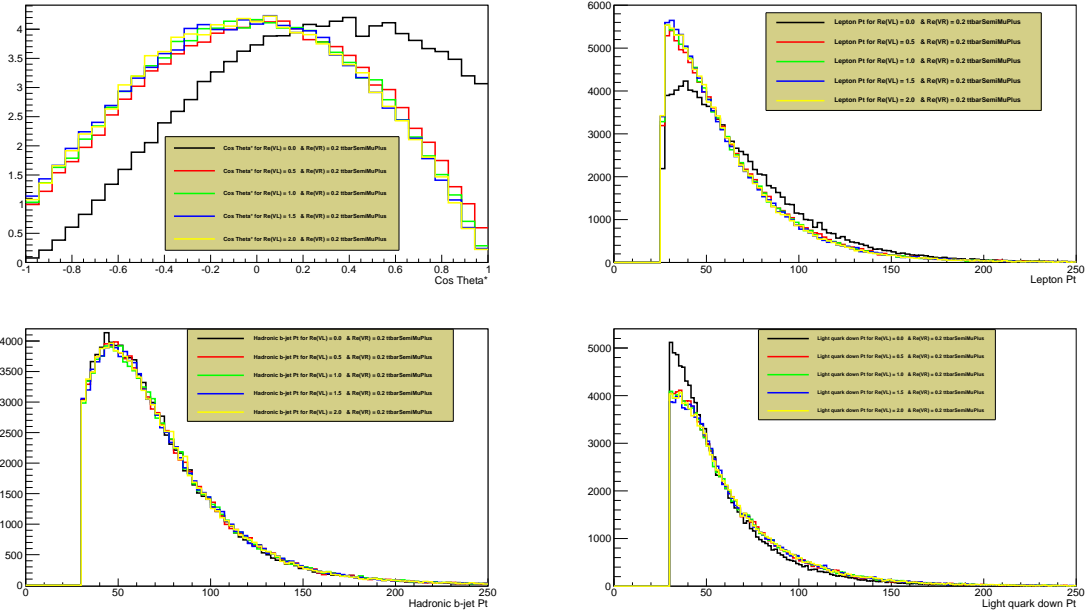


Figure 3.12

3.4.2.2 Fixing normalisation for 1D-variation of V_R only

Since the above mentioned results seem to predict that the left-handed vector coupling actually only influences the cross section and not the kinematics, it has been decided to ensure a correct cross section normalization for the 1D-variation of V_R only. This because the influence of the left-handed vector coupling

Remark: Correct way of thinking?

Isn't it just the opposite normalisation which should be corrected? The left-handed vector coupling only influences the cross section so the log(likelihood) should actually be normalized in such a way that it results in a flat distribution for the 1D-variation of V_L . This is which is currently still not correct because of the large difference in relative cross section for values with V_L smaller than 1.0. While I think that the 1D-variation will, just as was the case for the top quark mass, result in the correct minimization of the log(likelihood) distribution. This because both the influence of this 1D-variation on

the kinematic distribution and the variation of the cross section is symmetric around the Standard Model expectation value of 0.

Possible suggestion could be to check the influence of another coupling constant, for instance the imaginary part of the left-handed vector coupling. The distinct difference between the left-handed vector coupling and all the other coupling parameters implemented in the FeynRules model is its Standard Model value. The left-handed vector coupling is the only parameter which is supposed to contribute to the Wtb Lagrangian, and hence the only one different from 0. Therefore it could be a possible test of the created FeynRules model to check both the variation of the cross section as the influence on the kinematic distributions when a different coupling constant is considered. If it would only be that the real part of the V_R doesn't contribute to the kinematics of the decay particles while all the other ones do have a distinct effect, it could be a clear indication that the created FeynRules model is not capable of dealing with coupling constants higher than 1.

Thinking ...

However this last conclusion is not completely correct because then a possible effect would have been visible when the V_L variable was varied in a larger range. In this case also the configurations $V_L = 0.5$ and $V_L = 0.0$ was compared to the Standard Model expectation of 1.0 but no effect was found. So there is more behind the non-influence of the left-handed vector coupling V_L than just a problem with dealing with coupling coefficients larger than 1.0 ...

Chapter 4

Preliminary Results

Chapter 5

Understanding Results Obtained With MadWeight

Chapter 6

Analyzing created FeynRules model

Chapter 7

Event Selection

Chapter 8

Event corrections and reconstruction