

HRS Optics Instructions ***Under Development***

Scott Barcus

Last Updated: 12/12/16

E-mail: skbarcus@email.wm.edu

Introduction

The following instructions will describe how to optimize the optics transport matrix for the Hall A high resolution spectrometers (HRS). For the purpose of these instructions I will assume that you will be working on the JLab ifarm although all relevant scripts can be run on your local machine as well. These instructions include:

1. How set up your ifarm account to run the analyzer as well as what links and libraries are needed to run the optics matrix optimization scripts.
2. Replaying a data run.
3. How to apply cuts to a production run ROOT file.
 - (a) Vertex Cuts
 - (b) DP Cuts
 - (c) Sieve Cuts
4. How to combine cut files and prepare data for optimization.
5. How to optimize the transport matrix.
 - (a) For θ .
 - (b) For ϕ .
 - (c) For Y.

Initial Requirements

To run the scripts needed for optimizing the optics transport matrix you will require the following things:

1. ROOT version 5.34.13.
2. The Hall A analyzer.
3. Various scripts, libraries, and databases.

To work with ROOT files on the ifarm you will need access to a version of ROOT which can be done by following the instructions located here <https://root.cern/downloading-root>. It is probably easier to simply add a path to ROOT to your .login file (cshrc shell) in the home directory of your JLab account. This will allow you to use the correct version of ROOT to run the scripts on the ifarm. Figure 1 shows the .login file used to access the correct version of ROOT. The section in the right half second from the bottom shows how to set up ROOT. Note: You will have to change the paths to match your system.

You will also need access to a version of the Hall A analyzer which is built on top of ROOT. The analyzer can be downloaded here <http://hallaweb.jlab.org/podd/download/> and instructions for installation can be found here <http://hallaweb.jlab.org/podd/doc/install.html>. Version 1.5.31 is stable and should work correctly. After installing the analyzer on your account you will need to add it

```

emacs@ifarm1101.jlab.org
File Edit Options Buffers Tools Sh-Script Help
#ident "(#).login ver 1.0 Aug 20, 1996"
# Default user .login file.
#
# This file is executed after ~/.cshrc at login.
# This file should contain commands that set the options
# for a terminal and/or window and commands that are
# only needed during an interactive session.
#####
# Source the site-wide syslogin and sysapps files.
# The syslogin file displays motd and notifies users of
# news and mail during login. The sysapps file sets up
# the user's environment for commonly used applications.
# These lines should not be deleted.
#####
source /site/env/syslogin
source /site/env/sysapps
#####
# Set up the terminal.
# The stty command sets certain terminal I/O options
# for the device that is the current standard input.
# The lines below define the actions of the keys on
# different keyboards (ie. delete and backspace).
#####
# Uncomment this if you are using an NCD Xterminal keyboard.
stty erase "^?" kill "^U" intr "^C" eof "^D" susp "^Z" hupcl ixon ixoff tostop tabs
#####
# Define the string that will prompt you for interactive input.
# You may pick one or define your own. Only one should be uncommented.
#####
set prompt = "`hostname`> " # Name of the machine you are on.
#set prompt = "`hostname`[!]" # Name of the machine and history number
#set prompt = "`whoami`> " # Your user name.
#set prompt = "`whoami`[!]" # Your user name and history number
#####
# Defines the LINES and COLUMNS environment variables for users
#####
set prompt = "`hostname`> " # Name of the machine you are on.
#set prompt = "`hostname`[!]" # Name of the machine and history number
#set prompt = "`whoami`> " # Your user name.
#set prompt = "`whoami`[!]" # Your user name and history number
#####
# Defines the LINES and COLUMNS environment variables for users
#####
if ($?tcsh) then
    setenv LINES ""
    setenv COLUMNS ""
endif
#####
# User specific additions should be added below.
#####
setenv JLAB_ROOT /site/12gev_phys
setenv ROOTSYS /u/apps/root/5.34.13/root # for ifarm
setenv JLAB_VERSION production
setenv JLAB_VERSION 1.2
unalias ls
#source $JLAB_ROOT/ce/releases/$JLAB_VERSION/jlab.csh
source $JLAB_ROOT/production.csh ${JLAB_VERSION}
#####
setenv DB_DIR /work/halla/triton/skbarcus/DB
setenv LD_LIBRARY_PATH ~/analyzer-1.5.31:${ROOTSYS}/lib
#May not need below.
#setenv DB_DIR ~/analyzer-1.5.31/DB
set path = ( ~/analyzer-1.5.31 $path )
setenv ANALYZER ~/analyzer-1.5.31
alias ls 'ls -lrt --color'
alias la 'ls -lra --color'
alias lx 'ls -X'
set prompt = "\n%\{033[1;32m%\}n@m %{\033[1;34m%\}~ %{\033[0m%\}>"

```

Figure 1. .login File. This file sets up your JLab ifarm environment such that ROOT 5.34.13 is used and the Hall A analyzer version 1.5.31 is used.

to your .login file as with ROOT (see Figure 1). The bottom section in the right half shows how to set up the Hall A analyzer.

The scripts, libraries, and databases needed to run the optics optimization code can be found on the ifarm in at `/lustre/expphy/work/halla/triton/Gmp_optics`. The files needed to replay a data run from the spectrometers can be found in `/lustre/expphy/work/halla/triton/replay_gmp`. For the purposes of this guide I will assume that the reader copied the directories and scripts exactly as they are found. Note: Many of the scripts automatically create files and place them in these already created directories so if any of the names are changed the scripts will need to be modified as well. In the directories you wish to run these scripts such as `Gmp_optics`, `Sieve1`, and `replay_gmp` you will need a `rootlogon.C` file which links to the analyzer, various libraries needed for the scripts, and (optionally) pre-loads some scripts when the analyzer is opened. Figure 2 shows this `rootlogon.C` file.

Replaying a Data Run

Before the optics transport matrix optimization code can be used ROOT files of the data from the spectrometer runs must be created from the raw data. The raw data is stored on the ifarm in `/mss/halla/gmp12/raw/`

```

emacs@ifarm1101.jlab.org
File Edit Options Buffers Tools C++ Help
////////////////////////////////////
//
// rootlogon.C
//
// Load Lib, paths and possible scripts to the analyzer upon start
//
////////////////////////////////////
//
// Author : Jin Huang <mailto:jinhuang@jlab.org> Mar 2008
//          Chao Gu      Update for g2p use      Mar 2012
//          Kalyan Allada Update for GMP expt.    Dec 2013
//
////////////////////////////////////
{
  TString Arch(gSystem->GetBuildArch());
  TString Arch32("linux");
  TString Arch64("linuxx8664gcc");
  if(Arch==Arch32){
    printf("\nrootlogon.C: Loading Replay Core Library...\n");
    gSystem->Load("/adaqfs/home/a-onl/gmp/longwu/replay/ReplayCore_C.so");
    gSystem->Load("libFpp.so");
  }

  else if(Arch==Arch64){
    printf("\nrootlogon.C: Loading Replay Core Library...\n");
    gSystem->Load("ReplayCore64_C.so");
    //gSystem->Load("libFpp64.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libFpp64.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmp5C.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmpFpp.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmpNormAna.so");
    //gSystem->Load("libBeam.so");
    gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmp_Rastered_Beam.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmpShower.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmp_Beam_Eloss.so");
    //gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmp_Track_Eloss.so");
  }

  //Load more libs here, if necessary.
  //Make sure it's in path of $LD_LIBRARY_PATH

  printf("\nrootlogon.C: Adding include directories...");

  gSystem->AddIncludePath("-ISANALYZER");
  gInterpreter->AddIncludePath("SANALYZER/");

  gSystem->AddIncludePath("-ISANALYZER/src");
  gInterpreter->AddIncludePath("SANALYZER/src/");

  gSystem->AddIncludePath("-ISANALYZER/hana_decode");
  gInterpreter->AddIncludePath("SANALYZER/hana_decode/");

  gSystem->AddIncludePath("-ISANALYZER/hana_scaler");
  gInterpreter->AddIncludePath("SANALYZER/hana_scaler/");

  gROOT->LoadMacro("cut_R.C");
  gROOT->LoadMacro("ROpticsOpt.C++");
  gROOT->LoadMacro("ROpticsOptScript.C");

  printf("\nrootlogon.C: Done!\n\n");
}
////////////////////////////////////
rootlogon.C Top L2 (C++/L Abbrev)
Structure Messages Log Pdf Viewer Source Viewer Ready

```

```

emacs@ifarm1101.jlab.org
File Edit Options Buffers Tools C++ Help
//gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmp5C.so");
//gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmpFpp.so");
//gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libGmpNormAna.so");
//gSystem->Load("/work/halla/gmp12/longwu/gmp_analysis/replay/libBeam.so");
}

//Load more libs here, if necessary.
//Make sure it's in path of $LD_LIBRARY_PATH

printf("\nrootlogon.C: Adding include directories...");

gSystem->AddIncludePath("-ISANALYZER");
gInterpreter->AddIncludePath("SANALYZER/");

gSystem->AddIncludePath("-ISANALYZER/src");
gInterpreter->AddIncludePath("SANALYZER/src/");

gSystem->AddIncludePath("-ISANALYZER/hana_decode");
gInterpreter->AddIncludePath("SANALYZER/hana_decode/");

gSystem->AddIncludePath("-ISANALYZER/hana_scaler");
gInterpreter->AddIncludePath("SANALYZER/hana_scaler/");

gROOT->LoadMacro("cut_R.C");
gROOT->LoadMacro("ROpticsOpt.C++");
gROOT->LoadMacro("ROpticsOptScript.C");

printf("\nrootlogon.C: Done!\n\n");
}
rootlogon.C Bot L36 (C++/L Abbrev)

```

Figure 2. rootlogon.C File. This file links to various libraries used by the scripts as well as pre-loads some scripts when the analyzer is opened.

and `/cache/mss/halla/gmp12/raw/` directories. To replay these files go to your `replay_gmp` directory and open the analyzer. In the analyzer type “`.x replay_R.C(run #)`” where `run #` is the number of the raw data run found in the directory holding the raw data (such as 22828). The script will then ask how many events you wish to analyze and then it will produce a `.root` file in the `rootfiles` directory which can then be used with the optimization scripts.

Applying Cuts to the Raw Data

Once you have everything installed and all the scripts, libraries, and data files required to run the optimization code cuts must be made to the raw data so that undesired events are removed from the dataset. There are three different cuts that need to be made to the data from the nine carbon foils. The first are the vertex cuts which delineate where each of the nine foils are located. The second set of cuts are the DP cuts on the momentum. The third set of cuts are the sieve cuts which indicate where the holes in the optics sieve plate are located. Once these cuts are made they will be combined into a single ROOT file and then used to optimize the θ , ϕ , and Y components of the optics transport matrix.

In the `Gmp_optics` directory open the `cut_R.C` file and enter the ROOT files you want to use for

optimization and the path to them. Next open the analyzer. If you copied the logonroot.C file exactly then the cut_R.C script should load automatically. If not type “.L cut_R.C” to load the script. The first set of cuts to be made are the vertex cuts which show where the nine carbon foils are located. To make these cuts type “CutVertex(1)”. A new window will pop up that looks like Figure 3. The parameter ‘1’ means that you wish to make new cuts to the data (it will overwrite old cuts if there are any) and the parameter ‘0’ will display the current cuts being used.

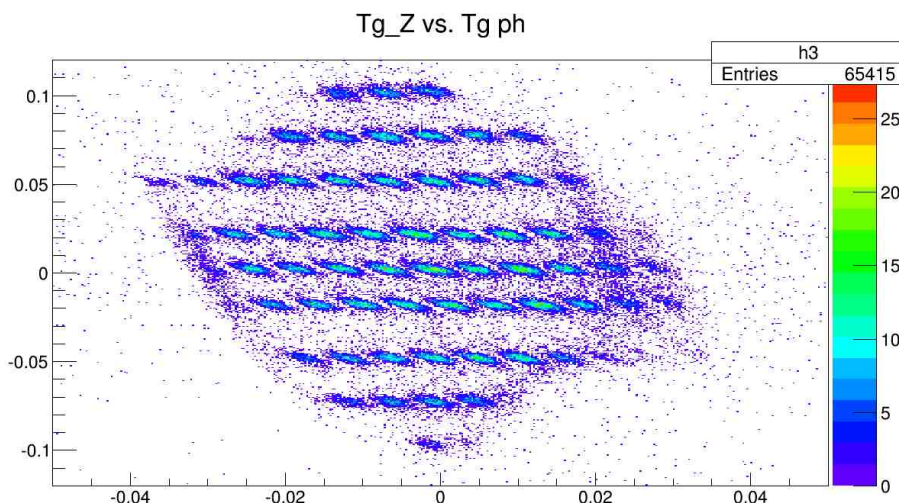


Figure 3. Vertex Cut Plot. This plot shows what the data looks like before any cuts are applied. Note that nine distinct foils can be seen.

The cuts are made by clicking on a position on the plot in the new window and then drawing a shape around the foil as in Figure 4. Double clicking completes the cut and you can then make the next cut. There are nine foils that need to be cut numbered zero through eight and the bottom foil on the plot is foil zero. The first cut you make will be foil zero. After this cut is made (you will see the cut turn magenta after double clicking) the cut image will disappear as the canvas updates and you will be prompted to make the next cut on foil one above the cut on foil zero. Repeat this process until all nine cuts are made. The ROOT files and images of these cuts will be saved in the Sieve1 directory.

After the vertex cuts have been made the DP momentum cuts must be made. These are made in a similar manner to the vertex cuts. To make the DP cuts type “CutDP(1)”. As before this will pop up a new plot to be cut as seen in Figure 5. As previously done draw a cut around the area containing the data in it and then double click to complete the cut. The result should look similar to Figure 6. After making a cut a new window with a new DP distribution will pop up to be cut. Repeat the process until all nine DP cuts are completed.

The next set of cuts to be made are the sieve cuts. These cuts indicate where the holes on the optics sieve plate are located. Once again cuts need to be made for each of the nine foils. The sieve plate for GMp has 25 columns but the ROpticsOptScript code uses 27 columns numbered zero to 26. This means that columns zero and 26 should never have data in them and the 25 columns of data are from columns one to 25. The odd numbered columns have seven rows of holes but the bottom row can’t be seen in the data and is thus ignored. Therefore starting from the second hole from the bottom they are numbered zero to five. The even numbered columns have six rows of holes labelled from the bottom up zero to five. There are two large holes to help find your position when looking at the plots they are (13,3) and (11,1).

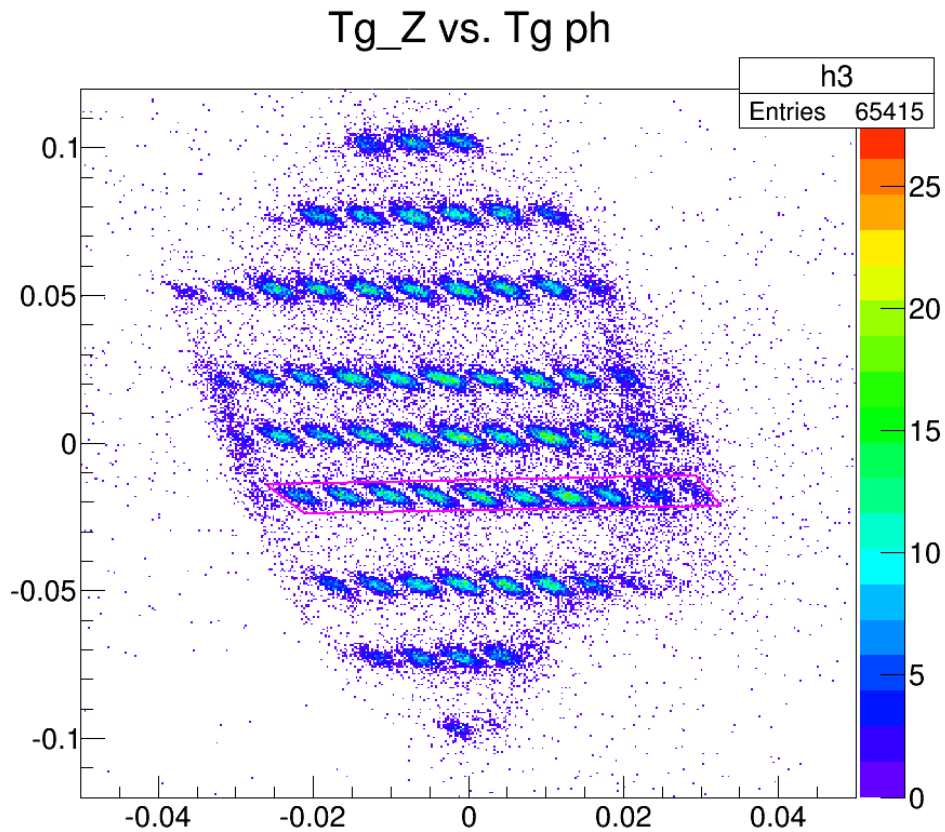


Figure 4. Vertex Cut Foil Three. This plot shows what the vertex cut on foil three looks like. Recall there are nine foils numbered zero to eight and the foil at the bottom of the plot is foil zero.

The Gmp optics RHRS sieve plate is shown in Figure 7.

The command to make these cuts is “CutSieve(foil # ,column #,1)” where foil # is the foil you wish to cut (numbered zero to eight) and column # is the column of holes you want to cut (numbered one to 25). After selecting which foil and row is to be cut the script will ask how many holes are in the column being cut. Count the number of clearly visible holes seen in the plot of the selected column and enter that number. The script will then ask for the ‘minimum hole ID’ which is the lowest value hole from zero to five (zero is the lowest hole) in the chosen column. Then starting from the bottom hole in the selected column the cuts for each hole can be made as before. The resulting cuts should look like Figure 8. When cutting the different foils you will sometimes be unable to see the larger holes to tell which column is which. This means we need another way to determine the column number. This can be done by applying the geometry found in Figure 9.

Each of the nine foils has an offset from the origin of the z-axis starting with foil 0 at a ΔZ of -10 cm up to foil nine with a ΔZ of +10 cm. We can use this ΔZ to calculate the ΔY offset on the sieve plate using Equation 1 where θ_S is the scattering angle which for this data was 48.66 degrees. We also know that the distance between the center of every two columns is 1.25 cm such that the distance between column 13’s center and column 11’s center is 1.25 cm. So for every ΔY of 1.25 cm we have to shift the

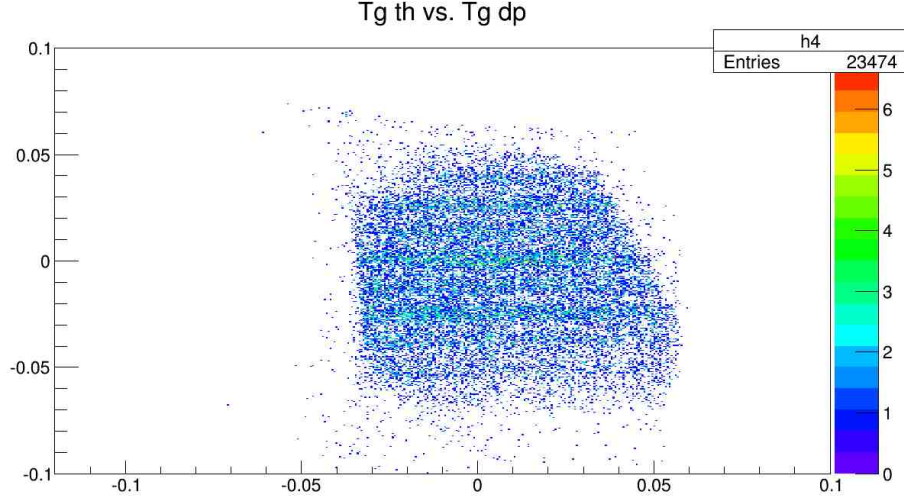


Figure 5. DP Cut Plot. This plot shows what the DP momentum distribution looks like before being cut. This is the first of nine plots to be cut.

central column of the foil by two columns. So if we divide each foil's ΔY by 1.25 cm and multiply the result by two we can find the offset of the central column of each foil. Table 1 gives these values and ultimately which column is the central column for each of the nine foils.

$$\Delta Y = \Delta Z \sin(\theta_S) \quad (1)$$

Table 1. High Voltage Values for PMTs

Foil #	0	1	2	3	4	5	6	7	8
ΔZ cm	-10	-7.5	-5	-2.5	0	2.5	5	7.5	10
ΔY cm	-7.51	-5.63	-3.75	-1.50	0	1.50	3.75	5.63	7.51
ΔY cm/1.25 cm	-6.0	-4.5	-3.0	-1.2	0	1.2	3.0	4.5	6.0
Shift in Central Column #	-12	-9	-6	-2	0	2	6	9	12
Central Column #	25	22	19	15	13	11	7	4	1

For the final optics optimization it is important to cut every hole on each foil, but it is often a good idea to first cut say all of the middle foil and 20% of the others and run the optimization on these to see if the results look reasonable. If the results don't look correct and the cuts need to be changed you can save a large amount of time this way.

Combining Cut Files and Preparing for Optimization

Once all three types of cuts have been made in the Sieve1 directory where all the cuts are stored type `./gen.sh` to run the script which combines all of the cuts into a single ROOT file. This will produce the files `gmp_22828.root.SieveCut.cut`, `gmp_22828.root.SieveCut.ncut`, and `gmp_22828.root.FullCut.root`. Next go to the Gmp-optics directory and type `./SieveTree2Ascii` to produce the last two files (`Sieve.full.f22828`

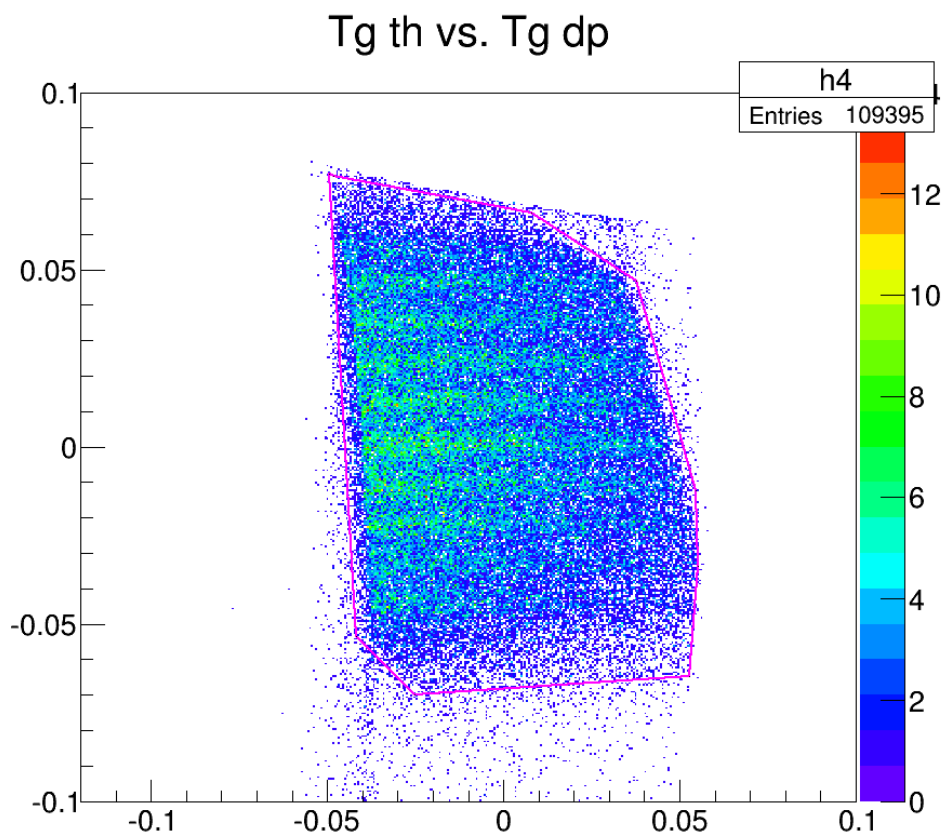


Figure 6. Individual DP Cut. This plot shows the fourth DP cut out of nine.

and Sieve.0.f22828) needed to run the optimization script. Running SieveTree2Ascii can take over 20 minutes to run depending on the amount of data being analyzed and your machine.

Running the Optimization Script

The next step is to optimize the optics transport matrix for θ , ϕ , and Y . Note: In the following figures I have only cut some of the holes in each sieve plane except for the center one for efficiency. The script for this optimization is `ROpticsOptScript.C` and is in the `Gmp_optics` directory. Note: Before running this script the flags at the beginning of the file for θ , ϕ , and Y optimization need to be set to true. In the `Gmp_optics` directory the unmodified data can be viewed by starting the analyzer and typing `PlotDataBase("Path to Database")` as seen in Figure 10. The database is in the `Opt1` directory in `Gmp_optics` and called `db_R.vdc.dat.0`. Figure 10 shows the sieve plane for each of the nine foils. The cross-hairs indicate where the center of the sieve holes should theoretically be located. As you can see the data is not yet in good agreement with where the holes are expected to be.

The first part of the optimization is the θ variable. This optimization is done by opening the analyzer in the `Gmp_optics` directory and typing `ROpticsOptScript("theta", "path to database", "path to output file")`. Recall that the database is in the `Opt1` directory in `Gmp_optics` and called `db_R.vdc.dat.0`. It is

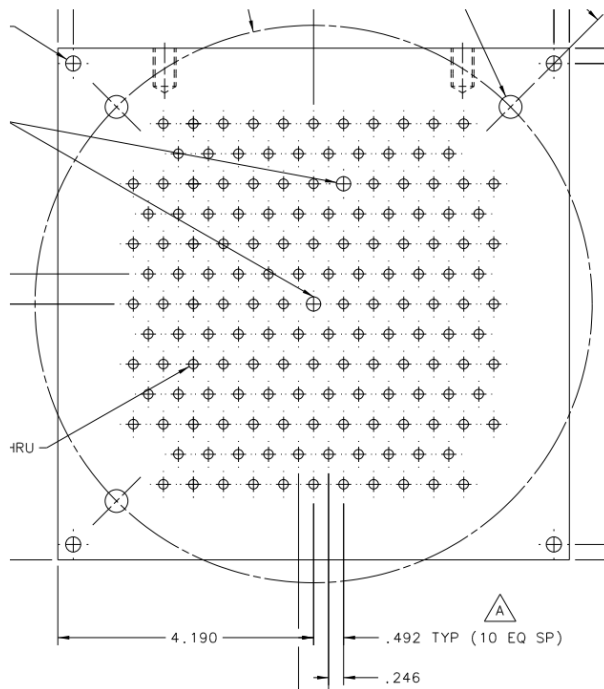


Figure 7. Sieve Plate for GMp RHRS. This plot shows the sieve plate used with the RHRS for the Fall 2016 run of GMp.

easiest to put the output file in this same directory with some name like db.R.vdc.dat.theta. The results of this first level of optimization can be seen in Figure 11. There is a clear improvement in the agreement between the data and the theoretical hole locations but it still isn't very accurate.

After optimizing θ the ϕ variable needs to be optimized. This is done similarly to the θ optimization by typing "ROpticsOptScript("phi", "path to optimized θ file", "path to output file")". The only changes from the θ optimization is using 'phi' instead of 'theta' and using the previously optimized file as the input. The results of optimizing ϕ as well as θ can be seen in Figure 12. Now the data agrees fairly well with the theoretical location of the sieve holes.

The last variable to be optimized here is the Y variable. This is done as before by typing "ROpticsOptScript("y", "path to optimized θ and ϕ file", "path to output file")". The output of this command is different than the first two optimizations. Two windows will pop out one showing the target Y and one showing the target Z as seen in Figures 13 and 14 respectively. In these plots the blue lines represent the ideal position of the peak of the red curve which is fit to the data. The Y optimization looks good for all of the foils except foil zero and foil five are still a bit off. The Z optimization looks decent for all foils except zero and eight and looks slightly off for foil five. There are several reasons that the optimizations may be off. The initial cuts to the data might not be good enough and some may need to be redone. The InputR.h file may not have the correct survey data or have some other incorrect inputs. The database being used might also be outdated or otherwise flawed.

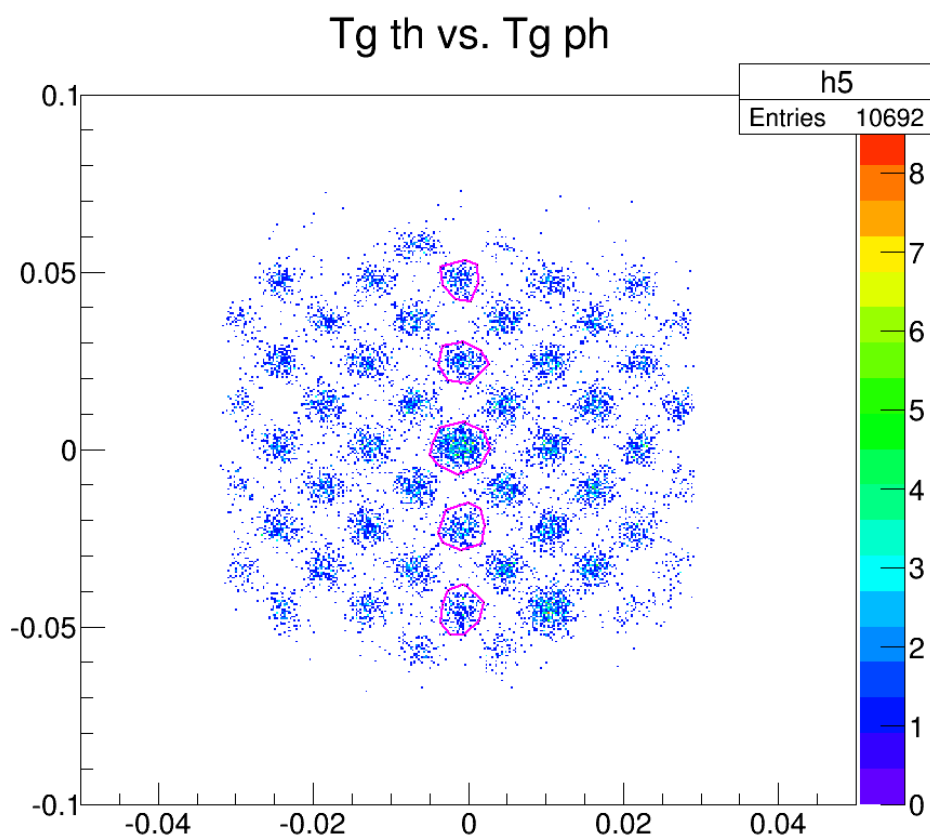


Figure 8. Sieve Cuts. This plot shows one column of sieve cuts. This cut is on foil four and column 13 and contains holes numbered one to five.

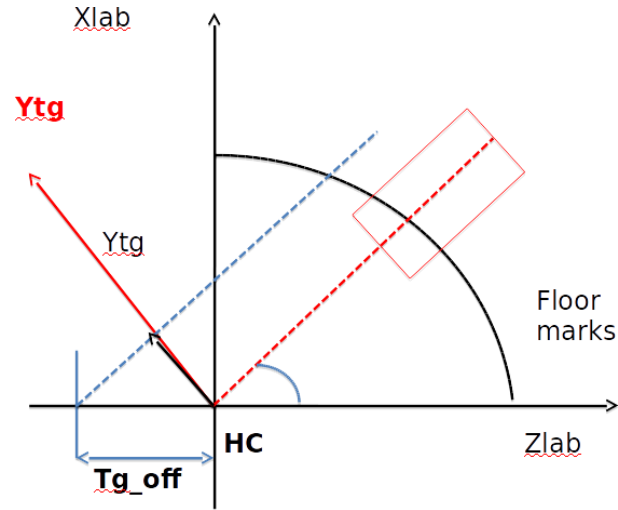


Figure 9. Target Offset. This diagram shows the geometry of the target when it has an offset from its ideal position. The light red box represents the spectrometer. Diagram made by Dien Nguyen.

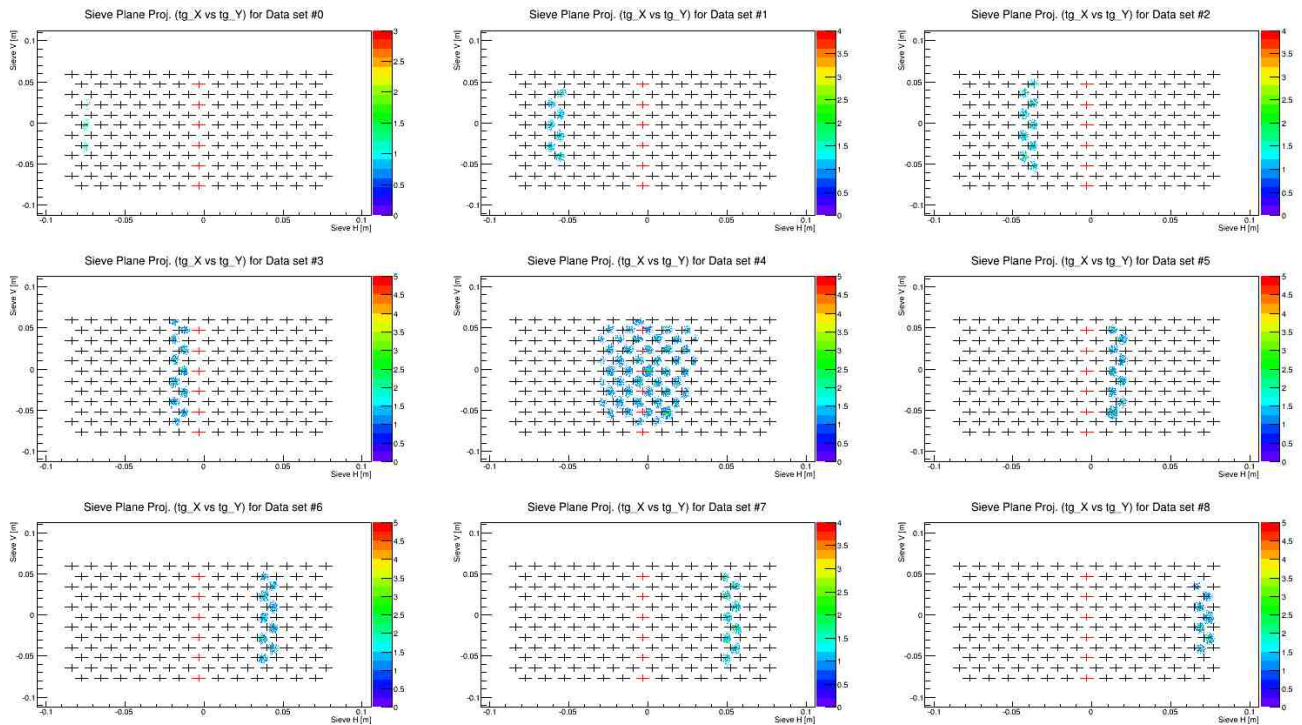


Figure 10. Plot of Database without Optimization. This plot shows the sieve plane for every foil without any optimization applied. The cross-hairs show where holes should be theoretically centered and it is clear that without optimization the agreement is poor.

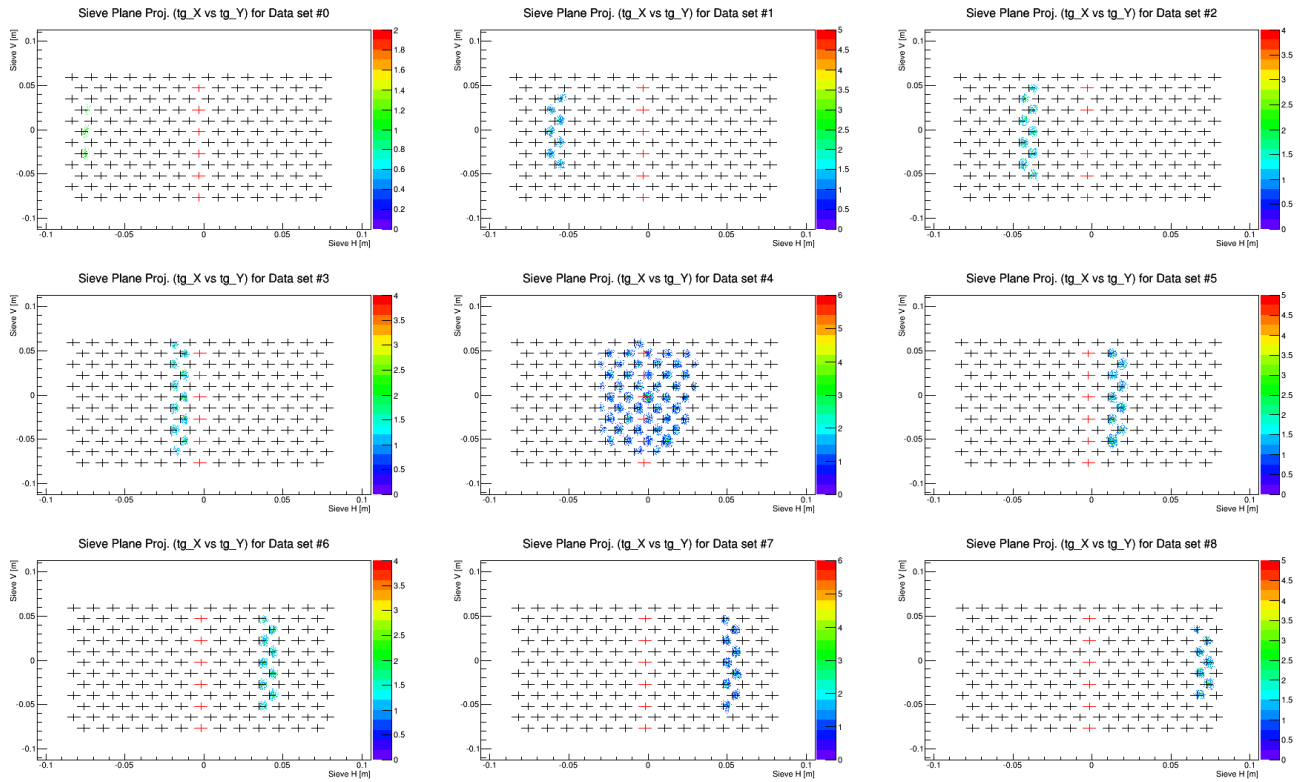


Figure 11. Sieve Plane Theta Optimized. This plot shows the sieve plane for every foil with the theta variable having undergone optimization. The agreement between where the holes are expected to be and the data is now improved but still off.

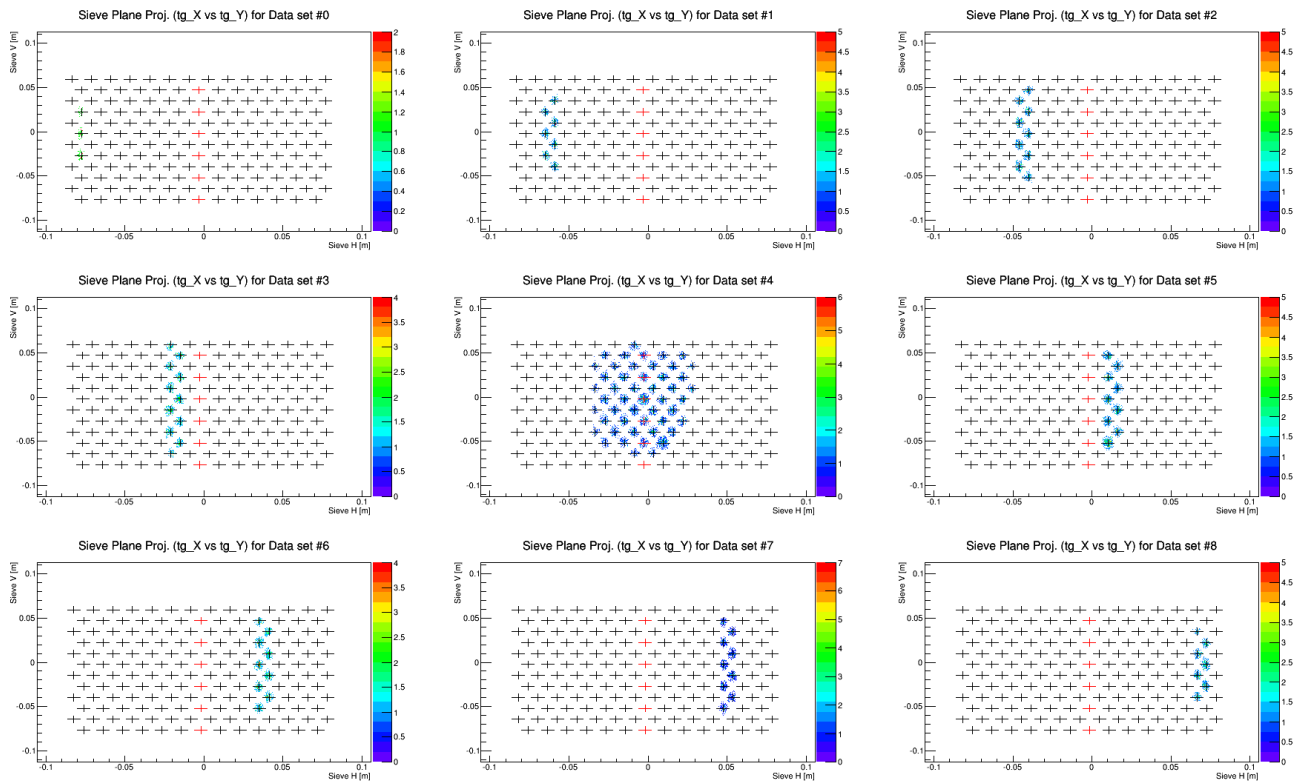


Figure 12. Sieve Plane Theta and Phi Optimized. This plot shows the sieve plane for every foil with the theta and phi variables having undergone optimization. The agreement between where the holes are expected to be and the data has once again improved and now looks reasonable.

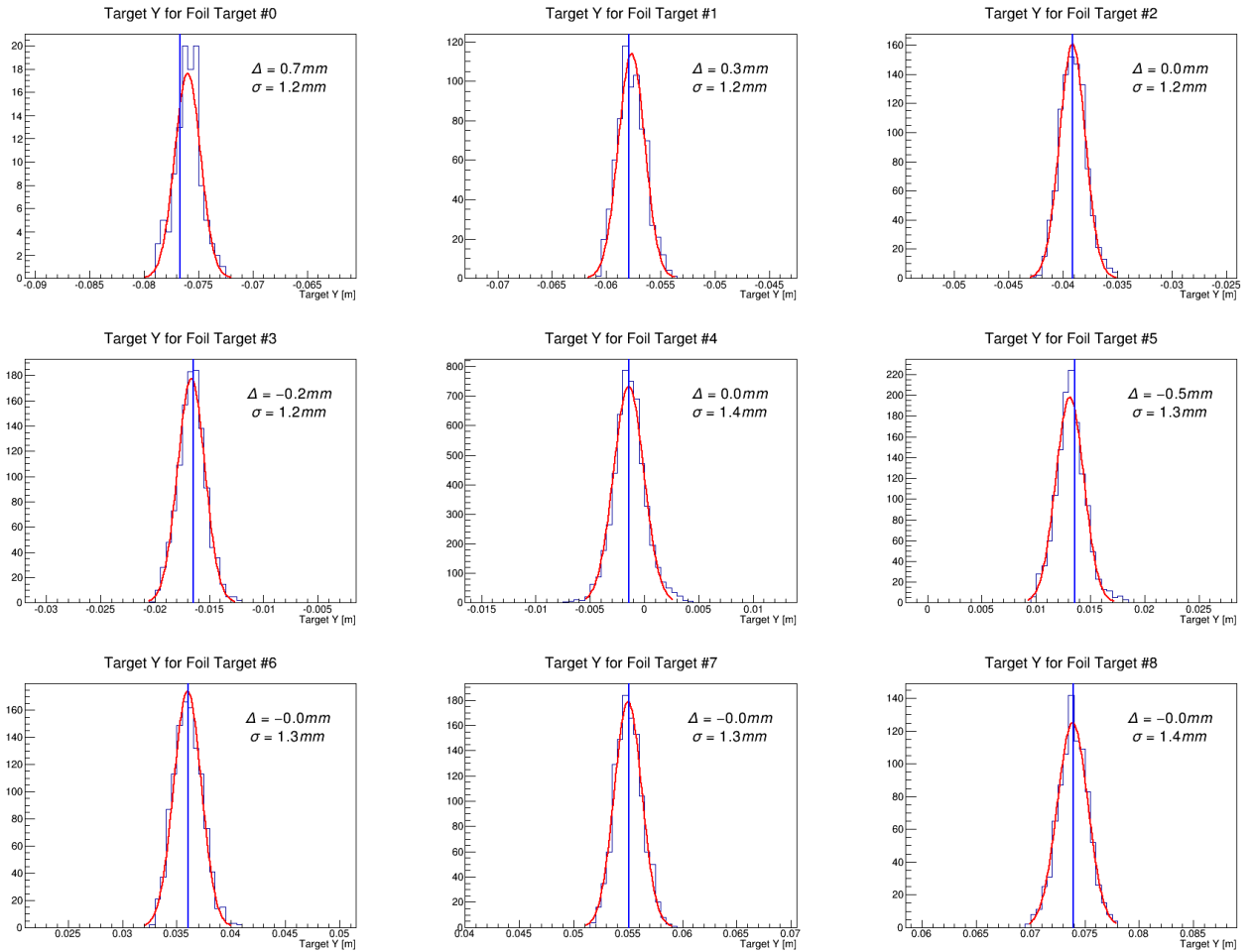


Figure 13. Y Optimization. This plot shows the Y optimization for each of the nine sieve planes. The results look good for all foils except foil zero and foil five look slightly off still.

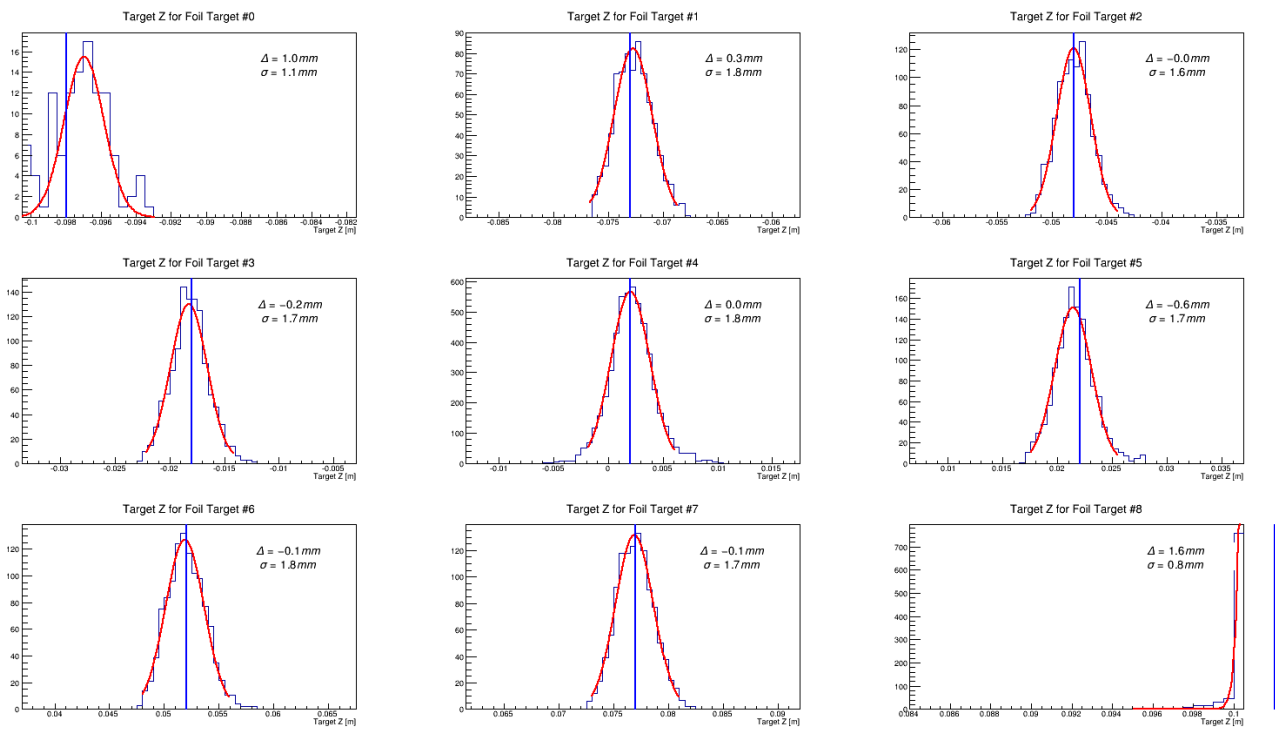


Figure 14. Z Optimization. This plot shows the Z optimization for each of the nine sieve planes. The results look good for all foils except for foils zero and eight and foil five looks slightly off as well.