

Hexanode detector library test

Content

- [Content](#)
- [Minutes of meetings](#)
- [Data](#)
- [Firmware library](#)
 - [References to download software:](#)
- [Software](#)
 - [Installation of 3-d party library](#)
 - [Installation in conda](#)
 - [Package hexanode](#)
 - [Package expmon](#)
 - [Package graphqt](#)
- [Examples](#)
 - [Performance test](#)
 - [Example with graphics](#)
 - [Number of hits per channel](#)
 - [Spectra of time per channel](#)
 - [Spectra of U, V, W \(ns\)](#)
 - [Spectra of U, V, W \(mm\)](#)
 - [Spectra of Xuv, Xuw, Xvw \(mm\)](#)
 - [Spectra of Yuv, Yuw, Yvw \(mm\)](#)
 - [Time sum \(ns\) for U, V, W](#)
 - [Time sum \(ns\) corrected for U, V, W](#)
 - [Deviation, Consistency Indicator, Reconstruction method](#)
 - [Time sum vs. variable U, V, W](#)
 - [xy image for hit1 and 2](#)
 - [XY image for uv, uw, and vw components](#)
 - [Resolution map](#)
 - [Reflection for all channels](#)
 - [Physics plots t1,x,y vs t0](#)
- [References](#)

Minutes of meetings

2016-12-02 Mtg Razib & Mikhail

1. list of detector types/data sources for support
 - hex-anode
 - quad-anodeconnected to
 - acqiris
 - TDC
2. sample(s) of data (experiment/run) which can be used for test/calibration purpose.
hex-anode : exp=xpptut15:run=280
quad-anode : ??? (Razib)
3. format of software (library of methods, a set of scripts, GUI wrapper, packaging, etc.)
 - detector-manufacturers software represents everything as GUI
 - library of methods/ scripts/ GUI make sense
4. list of monitoring/calibration plots, types and format of constants for calibration purpose
 - Razib will make screen-shots of all interesting plots
 - calibration should provide for each delay line (t0, k)
5. algorithms for implementation
 - peak-finding in waveform
 - plots stc.
6. list of analysis plots and output data formats
 - Razib will make screen-shots of all interesting plots

Data

Timur suggested to use data

- HEX anode detector: amod3814, runs 85, 88, 90, etc. any decently sized file

channel assignments including ACQIRIS <https://pswww.slac.stanford.edu/apps/logbook/attachments/87104/PINOUT>

- QUAD anode: amon2216, runs 299, 301 ... 305, 313, 315

channel assignment was Acqiris-1: 1-X1, 2-X2, 3-Y1, 4-Y2 (it could also be 1-X1, 2-Y1, 3-X2, 4-Y2) and Acqiris-3 - TOF.

Firmware library

Here is the person who should be contacted about getting the statically / dynamically linked library binaries for hit reconstruction and detector calibration:
Achim Czasch czasch@atom.uni-frankfurt.de

[2016-12-07-email-achim-czasch.txt](#)

2016-12-07 email from Achim

Achim Czasch <czasch@atom.uni-frankfurt.de>
Wed 12/7/2016 5:57 AM
To:
Dubrovin, Mikhail;
Cc:
Osipov, Timur;
Action Items
Dear Mikhail and Timur,

I have compiled the lib on CentOS 7 (which is basically RedHat???) on gcc version

Here are 2 example programs that show how to use the lib:

a)
This program can read the LMF-data files that our software "CoboldPC" produces. Timur Osipov may have some sample files from his beam times.

http://www.roentdek.com/download/LCLS_Linux/sort_LMF_1_detector.zip

b)
This is basically the same. But it does not read LMFs. It expects a simple binary file format. So you will start with this one because you have your own ADC data.

http://www.roentdek.com/download/LCLS_Linux/sort_non-LMF_from_1_detector.zip

Roadmap:

- measure the detector signals with your ADCs
- write a program that extracts the timing info from the signals.

Simple method:
look for the 2 samples which are just above and just below A.
A is the half amplitude of this signal.
The interpolate between the 2 points to find the timing value @ half ampl..

Advantage: Very robust. Works always.

better method:
Mimic the CFD-mechanism.
Superimpose the signal with an inverted and delayed copy of itself.
Look for the point where this new signal crosses zero.

Advantage: better resolution (about factor $\sqrt{2}$).

But the simple method is already very precise.

- Now you have the timing values of all 7 detector signals.
Now you can use my example programs.

Some histograms that you will need:

Let's call the signals $u_1, u_2, v_1, v_2, w_1, w_2$ and mcp .
(values in nanoseconds)

```
- 1D:  u1+u2-2*mcp
- 1D:  v1+v2-2*mcp
- 1D:  w1+w2-2*mcp

- 1D:  u1-u2
- 1D:  v1-v2
- 1D:  w1-w2

- 2D   u1+u2-2*mcp  versus u1-u2
- 2D   v1+v2-2*mcp  versus v1-v2
- 2D   w1+w2-2*mcp  versus w1-w2
```

3 position images:

Xuv versus Yuv
Xuw versus Yuw
Xvw versus Yvw

with

```
u= u1-u2
v= v1-v2
w= w1-w2 - w_offset
```

and

```
Xuv = u * fu;
Yuv = (u * fu - 2.*v * fv)*0.5773502691896;
Xuw = u * fu;
Yuw = (2.*w * fw - u * fu)*0.5773502691896;
Xvw = (v * fv + w * fw);
Yvw = (w * fw - v * fv)*0.5773502691896;
```

f_u, f_v, f_w are constants close to 0.7.

w_offset is a constant close to zero.

The exact values for these constants will be extracted from
real data later during the calibration. (Timur can tell you more about this.)

And it is also good to plot this data for each signals:

FWHM versus amplitude

You can calculate the FWHM (full width at half maximum) similar
to the "simple method" that I have described further above.
Just do the same on the trailing edge of the signal.

We will have to exchange some more emails until everything is working.

best,

Achim

2019-08-15 Achim's e-mail about calibration sequence

> Could you remind me please, what is a right order
> to adjust parameters and calibrate new detector?

yes, there is a sequence that must be followed in the correct order:

- a) First you must set the parameters in the config file so that the time sum peaks get shifted to zero.
- b) Then you set the 'runtime'.
- c) Then you set the xy calibration factors so the that image gets the size that you think is right.
(and set the xy offset parameters so that the image is well centered.)
- d) Then you set 'radius'. This value should be 1 or 2 mm larger so that it really includes all hits.

2019-08-16 Achim Czasch about runtime, scalefactors, calibration

Delay-line detector library use

Achim Czasch <czasch@roentdek.com>
2019-08-16, 12:00 AM Dubrovin, Mikhail

Hi Mikhail,

> Scale factors converting time to coordinate are not important now.
> because I am not aware about precise detector geometry.

For the DLD (not HEX) you must set the 2 calibration factors manually later.

Example: If you have MCPs with 40mm active diameter then you must the factors so that the image has a diameter of 40mm.
The factors for x and y will be slightly different to achieve a round image.

> >> b) Then you set the 'runtime'.
> Could you remind me please, how can I get this value(s)?

You plot x1-x2 (units: ns) in a 1D-plot.
Then logy-scale.

Now you can see a distribution that ranges from
-z to +z (z is just a number here).
For 'runtime' you chose z + 2 ns.

> But, distributions of the time_sum for u and v look flat, even without calibration tables.

The errors that are corrected are within about +-4 ns.
So maybe you must zoom in a bit (in the y-scale).
But for a DLD this correction not so important.

> Still, it looks important to have an ability to calibrate potential differential non-linearity
> of the delay lines. I hope this is possible for QUAD-anode, right?

Not as easily as with the HEX. With a HEX we have a 3rd layer.
This layer provides the additional info that we need to make an automatic linearity calibration of the 3 delay lines.
But with a normal DLD we do not have enough information.
So this algorithm can not be applied.
You can use a pin hole mask and measure the non-linearity manually.
Then you can apply another code from us that will spit out correction tables.
But this process is very very tedious and time consuming - and if you change something (cable lengths, even voltages) then you may have to do it again.

best,
Achim

2019-08-19 Achim - about sorter library and QUAD calibration

Achim Czasch <czasch@roentdek.com>

Today, 12:57 AM

Hi Mikhail,

> Does all this means that correction can't be evaluated for QUAD-anode for file with regular events?
> Then for QUAD-anode I do not see any reason to use this library.

The library does 3 things:

a) multi-hit reconstruction

(Important if you have groups/pairs of particles with $dt < 100$ ns)

This algorithm works with a HEX-detectors and DLD-detectors.

But with a DLD it has much less information.

So the results will not be as good as if you had used a HEX.

b) position dependent correction of the time sums:

This correction is used by the multi-hit reconstruction algorithm.

Sometimes it must reconstruct a missing MCP signal

by using the info from the 4 anode signals. For this it uses the time sums.

So it is a good idea to use the time sum correction. You will end up with

a more accurate TOF for reconstructed MCP-signals.

This correction is possible with a HEX-detectors and DLD-detectors.

c) position dependent correction of the image non-linearity:

This correction is only possible with a HEX-detector.

So the library does a,b,c if you use a HEX and a,b if you use a DLD.

=====

> However, I had an impression that a couple of correlation plots,

> t_1+t_2 vs t_1-t_2 for u and v (x and y) can be used for calibration purpose to bring time sum to constant...

Yes. This is correction 'b'.

It works for DLD and for HEX.

But the correction tables that you get from this calibration can only

be used for the correction of the time sums.

It can not be used for the linearity correction.

The linearity correction produces completely different tables.

In a way you could say that the 2 corrections are orthogonal to each other.

=====

> Method create_calibration_tables from sort.cpp crashes in case of QUAD-anode, in case of command=3.

> The lines which crash it:

> number_of_columns = sorter->sum_walk_calibrator->sumw_profile->number_of_columns;

sorter->sum_walk_calibrator->sumw_profile

does not exist if you use a DLD

only

sorter->sum_walk_calibrator->sumu

and

sorter->sum_walk_calibrator->sumv

exist.

=====

> If I fix these lines like:

> number_of_columns = (sorter->use_HEX) ? sorter->sum_walk_calibrator->sumw_profile->number_of_columns : 0;

> the file is saved as

> -----

> 49 // number of sum calibration points for layer U

> -101 1.10861

> -96.7917 0.857992

> -92.5833 0.607369

> -88.375 1.11123

> -84.1667 1.11032

> -79.9583 1.10941

> -75.75 1.10849

> ...

> 49 // number of sum calibration points for layer V

> -101 1.57853

> -96.7917 1.57853

> -92.5833 1.57853

> -88.375 1.67824

```

> -84.1667 1.77794
> -79.9583 1.87764
> -75.75 1.97735
> -71.5417 2.07705
> ...
> 0 // number of sum calibration points for layer W (only needed for HEX-detectors)
> 0 // number of pos-calibration points for layer U
> 0 // number of pos-calibration points for layer V
> 0 // number of pos-calibration points for layer W (only needed for HEX-detectors)
> -----
> with non-uniform corrections for layer U and V.
> Does it make any sense to use this file for QUAD-anode?
Yes, this is correct. This is how it should look like.

```

Achim

2019-08-23 Achin-email-gcc-version

Achim Czasch <czasch@roentdek.com>
 Fri 8/23, 12:05 AM Dubrovin, Mikhail
 Hi Mikhail,

> I hope they are backward compatible.

You might need some syntax changes on 1 or 2 lines.
 But no other big changes.
 Just send me the code where you see compiler errors.

> Is it possible to say what has been changed?

Some bugfixes and other improvements.
 I recommend the upgrade.
 I don't know which version you use.
 I can send you the list of changes if you tell me your current version number.

> Could you please tell me, what is the gcc compiler version for these files?

The files that I sent to you were compiled with gcc 4.8.5 20150623.
 Today I will upgrade my gcc. Then I could send you a new compiled version
 if it turns out that you need it.

> If I understood correctly, both waveform-processing algorithm would not work well for strongly overlapped signals.

Yes. Analyzing (de-convoluting) overlapping pulses is extremely difficult.
 The problem here is the fact that the shape of the signals is a function
 of the position on the detector where they were generated.
 So if you want to do a deconvolution using a standard minimization algorithm
 you don't know which shape to feed in (because at this point you don't know
 where the signal was generated).
 We have done this here in our group. But only as part of PhD-projects
 where the student develops a very personal relationship with his/hers data set.
 We have not yet found a way to write an algorithm that can do that with all data
 sets without extensive guidance from the user.

Achim

References to download software:

http://www.roentdek.com/download/LCLS_Linux/sort_LMF_1_detector.zip
http://www.roentdek.com/download/LCLS_Linux/sort_non-LMF_from_1_detector.zip

Reference from 2018-08-14:

http://www.roentdek.com/download/_USA/LCLS_Linux/sort_LMF_1_detector.zip

Software

Installation of 3-d party library

Static library file is installed under

`~/lib/hexanode-lib`

Also in

`/reg/common/package/hexanodelib/0.0.1/x86_64-centos7-gcc485/resort64c.h`

`/reg/common/package/hexanodelib/0.0.1/x86_64-centos7-gcc485/libResort64c_x64.a`

Installation in conda

2019-08-14 David Schneider has moved this library in conda:

`/reg/g/psdm/sw/conda/inst/miniconda2-prod-rhel7/envs/ana-1.4.7/lib/libResort64c_x64.a`

files:

`/reg/g/psdm/sw/conda/manage-ctrl/recipes/external/hexanode_proxy/build.sh`

`/reg/g/psdm/sw/conda/manage-ctrl/recipes/external/hexanode_proxy/meta.yaml` <<<=== change version to 0.0.2

`/reg/g/psdm/sw/conda/downloads/otherpkgs/hexanode-proxy-{{version}}.tar.gz`

preparation of hexanode-proxy-version.tar.gz

`cd /reg/common/package/hexanodelib`

`tar -czvf hexanode-proxy-0.0.2.tar.gz 0.0.2`

`cp hexanode-proxy-0.0.2.tar.gz /reg/g/psdm/sw/conda/downloads/otherpkgs/`

See: [Building External Packages From Git Tags](#)

Package hexanode

Lives in our release system and intended as a cpp/python tester/wrapper of the `resort64c` library

`hexanode/pyext/hexanode_ext.pyx` - cython/python extension module for `resort64c` library

`hexanode/app/` - C test examples

`hexanode/examples/` - for python examples

Package expmon

Lives in our release system and intended to support GUI interfaces for different hutch standard configuration projects.

`expmon/examples` - test of acqiris signals from hex- quad- anode detectors

Package graphqt

Lives in our release system and intended to support qt-based dynamic graphics.

Examples

Performance test

Run scripts on psanaphi110.

Data file hexanode-example-CO_4.lmf supplied by Timur, contains 100K events.

```
ex_sort hexanode-example-CO_4.lmf # C++
hexanode/examples/ex-05-sort.py hexanode-example-CO_4.lmf # Python
```

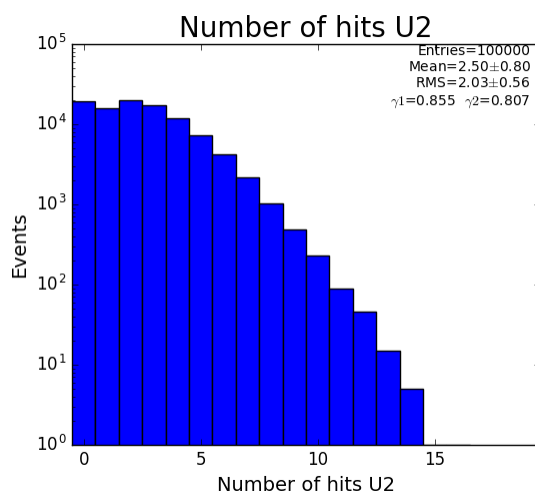
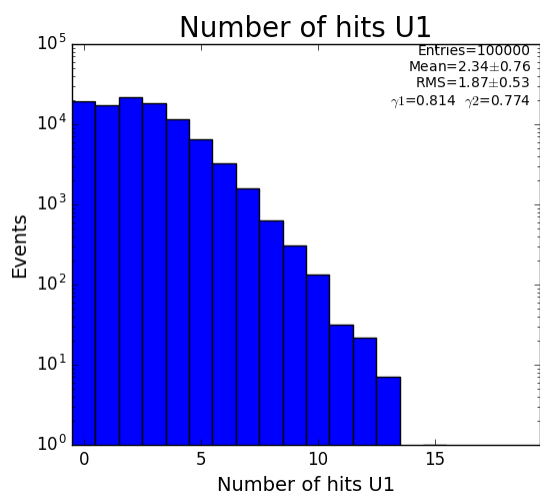
Changing command (1,2,3) in sorter.txt I got

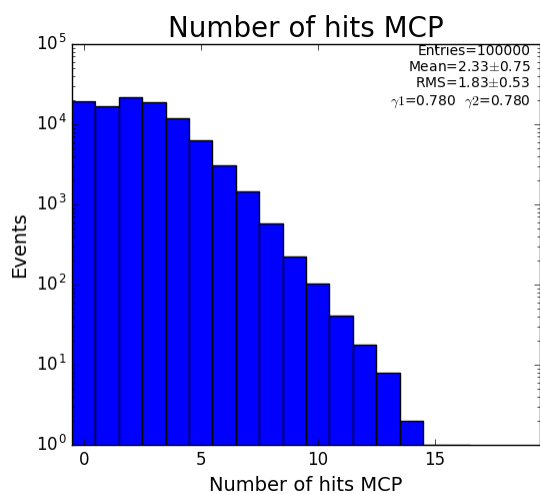
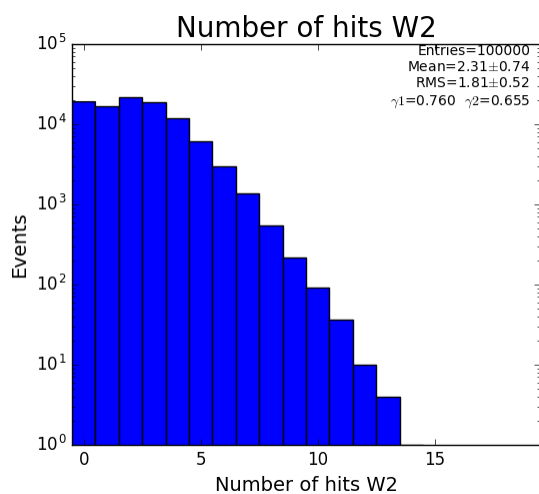
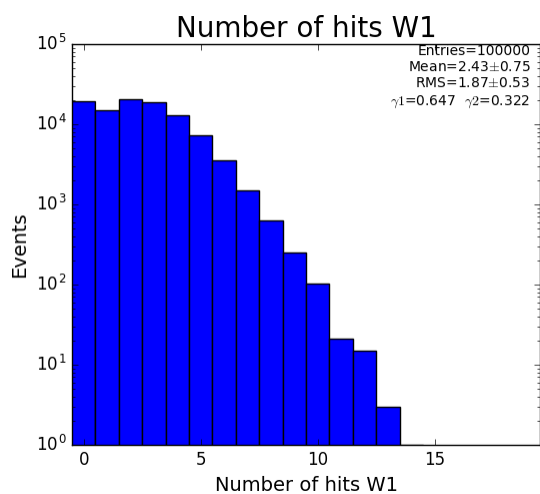
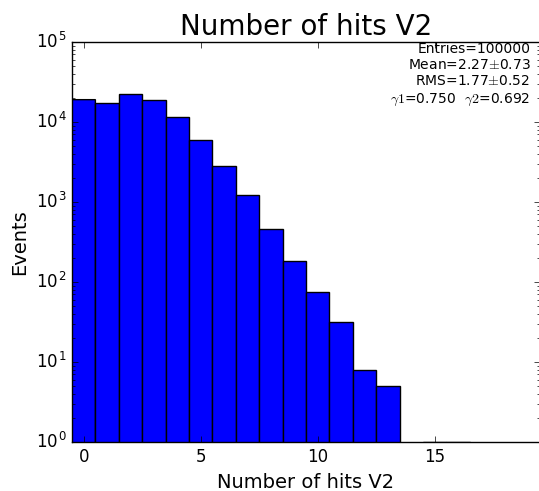
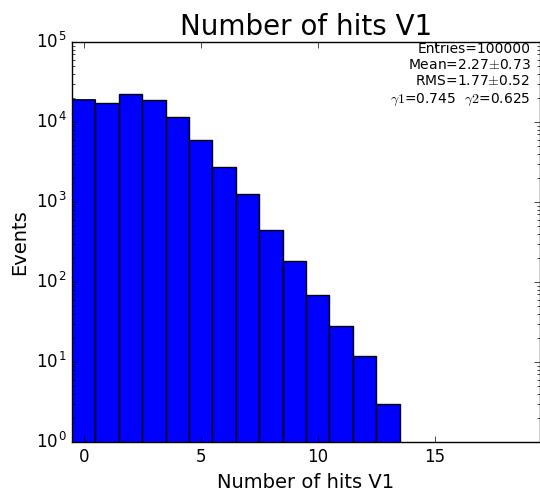
command	C++ (sec)	Python (sec)
1 (data processing)	3.6	4.3
2 (calib)	0.218	0.635
3 (calib)	0.230	0.656

Example with graphics

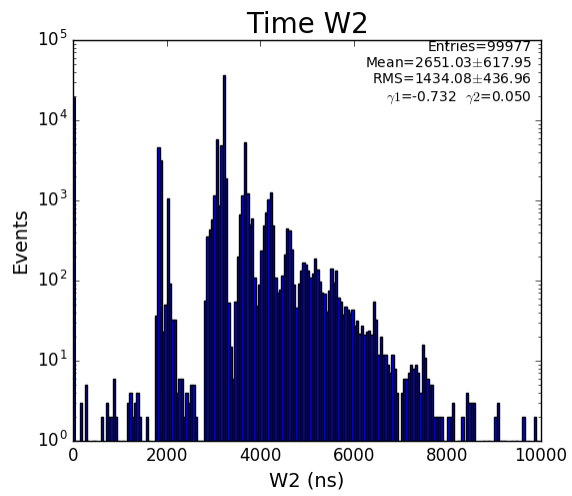
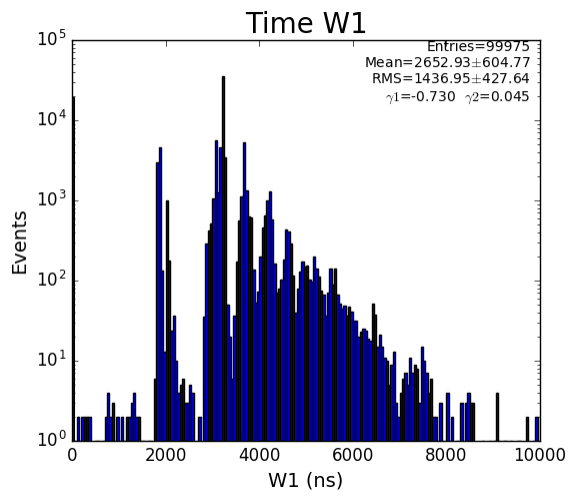
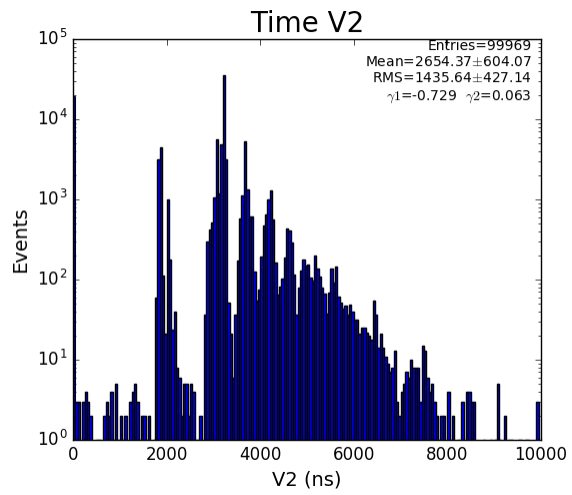
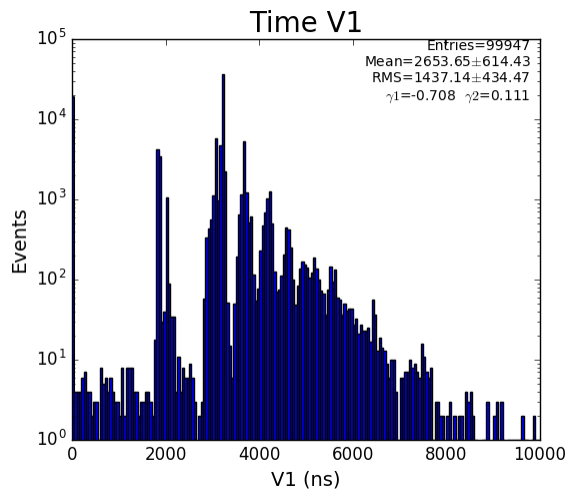
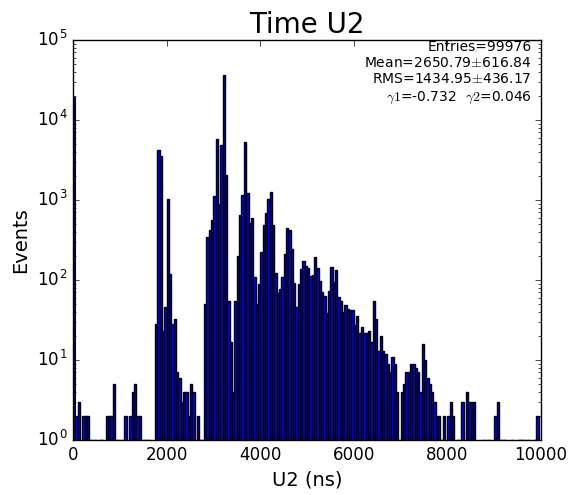
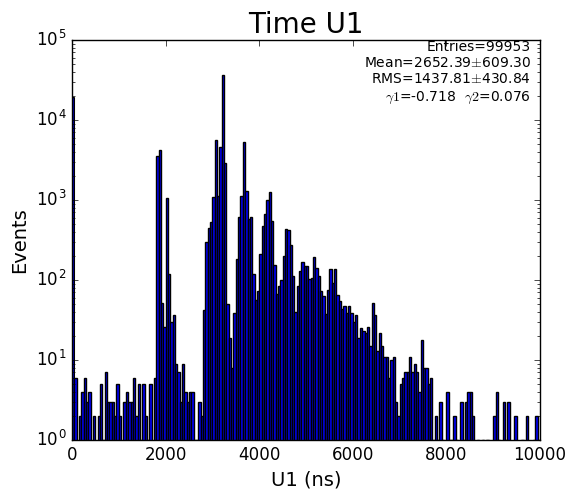
```
hexanode/examples/ex-06-sort-graph.py hexanode-example-CO_4.lmf
```

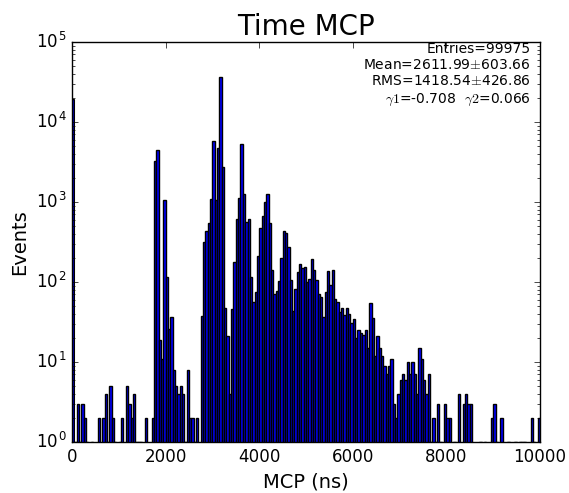
Number of hits per channel



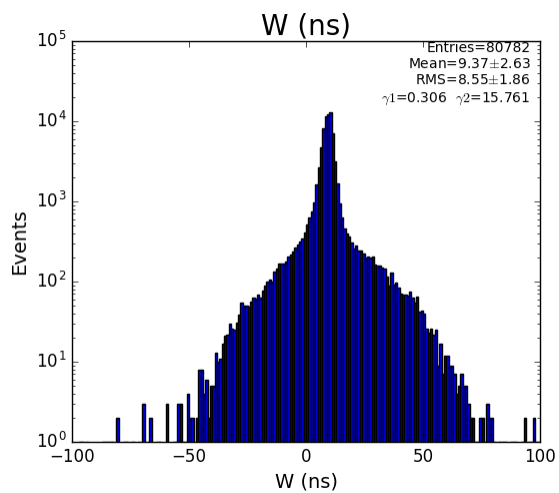
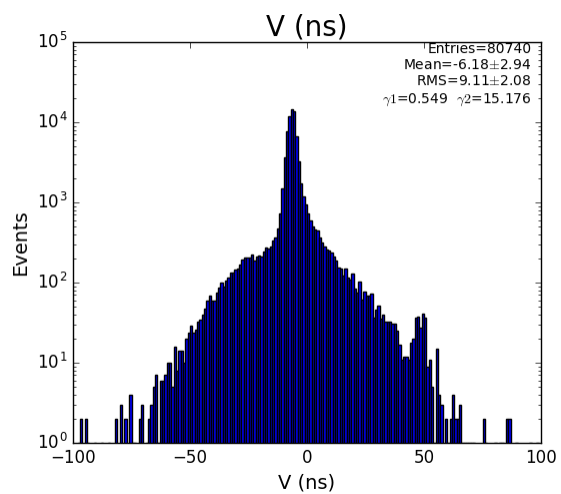
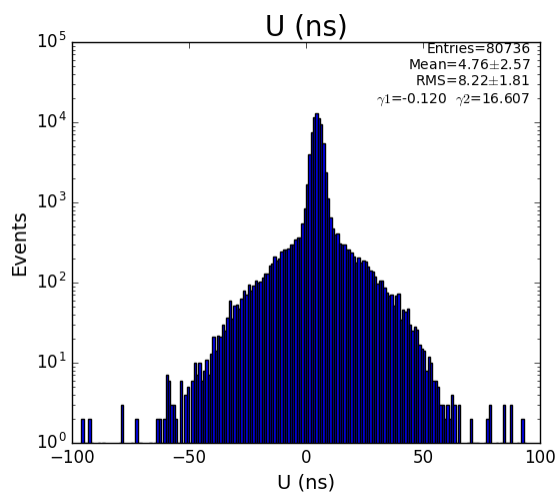


Spectra of time per channel

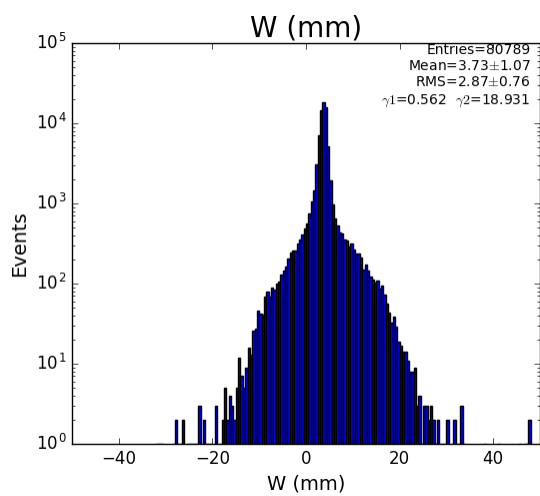
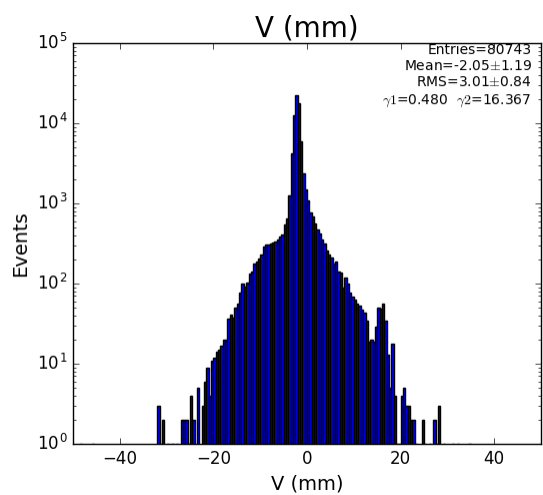
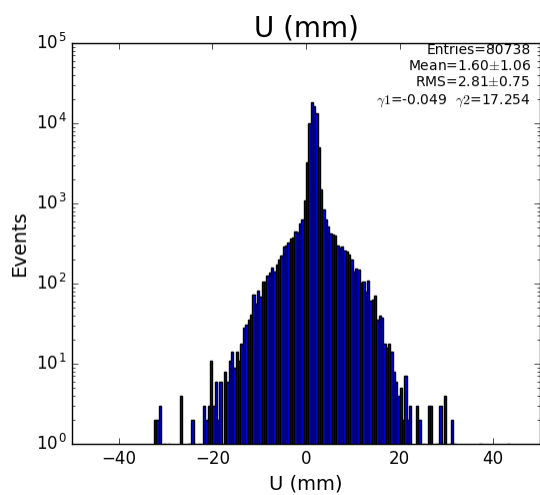




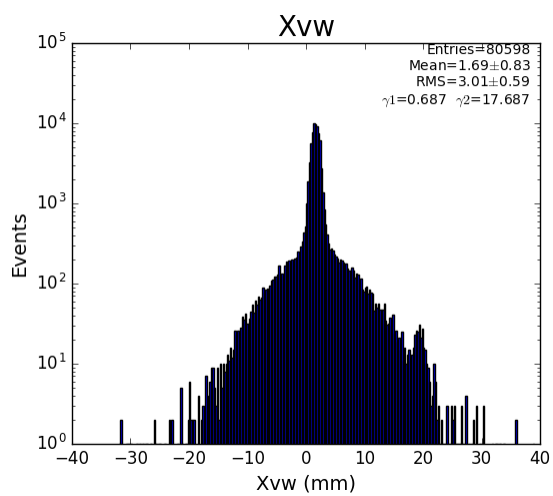
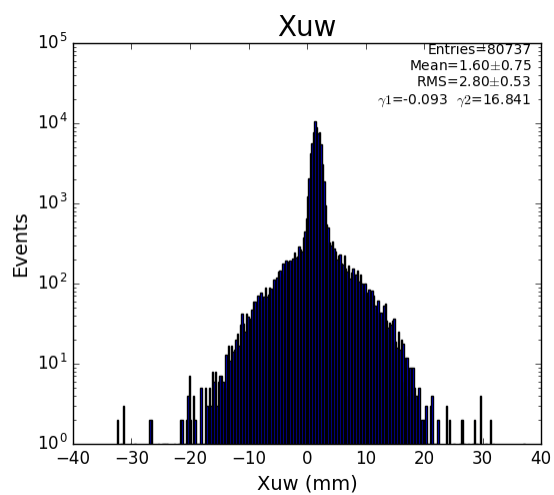
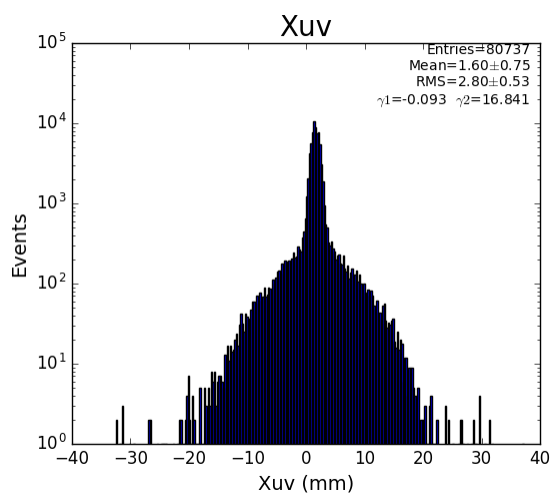
Spectra of U, V, W (ns)



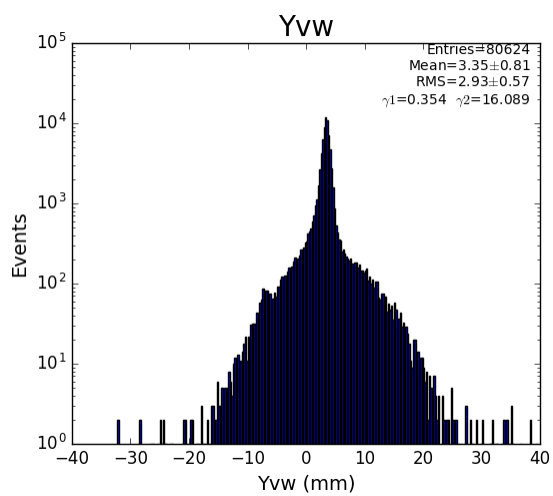
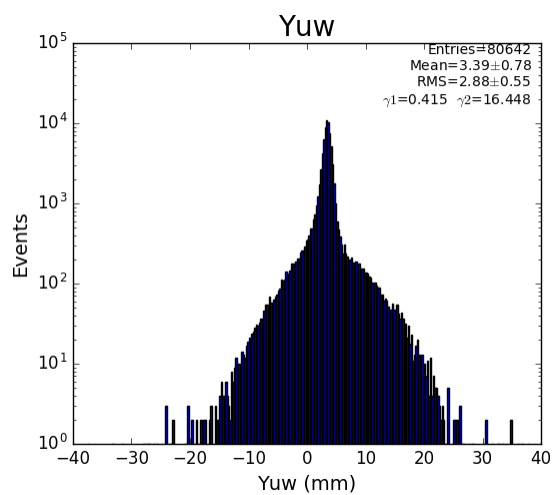
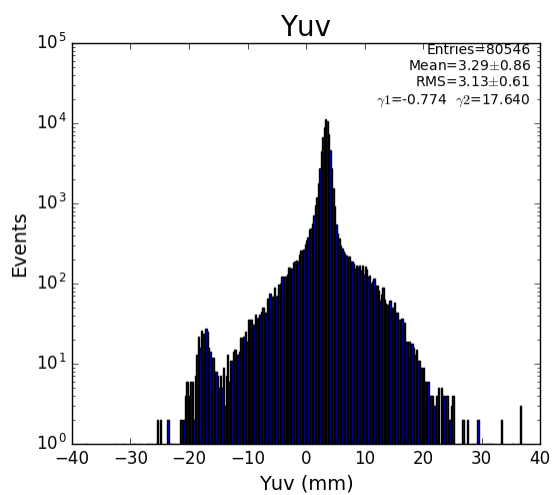
Spectra of U, V, W (mm)



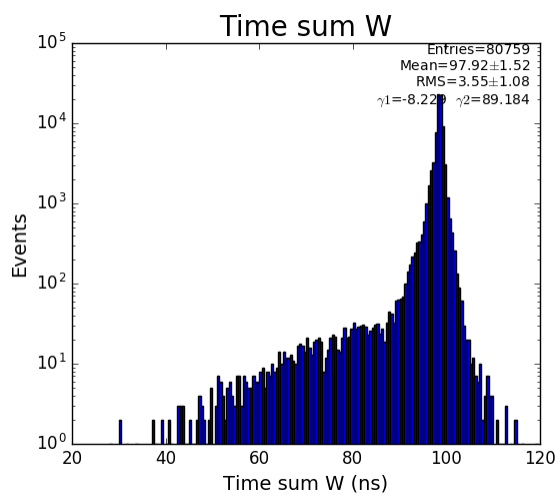
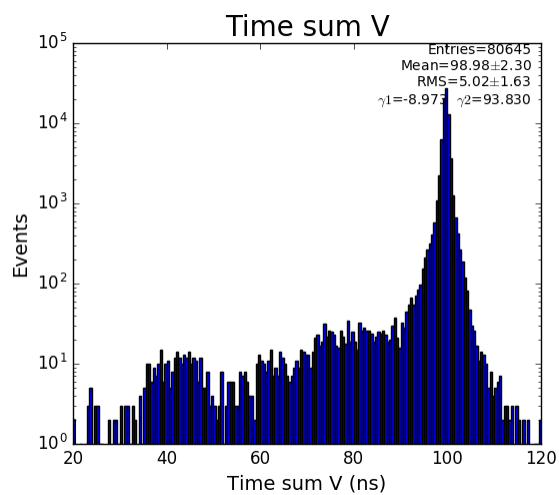
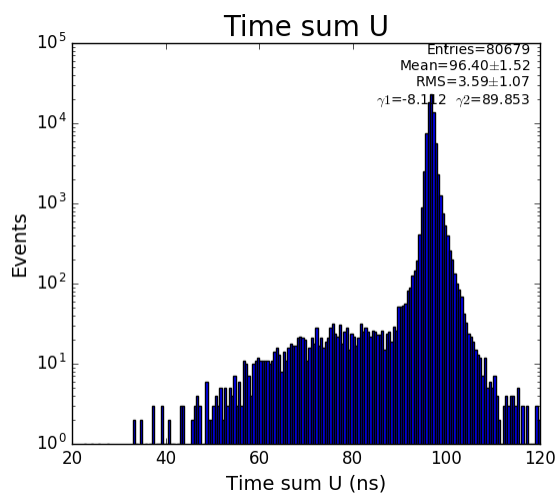
Spectra of Xuv, Xuw, Xvw (mm)



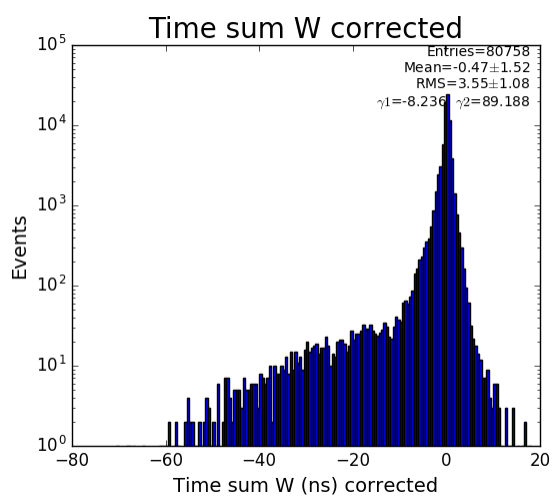
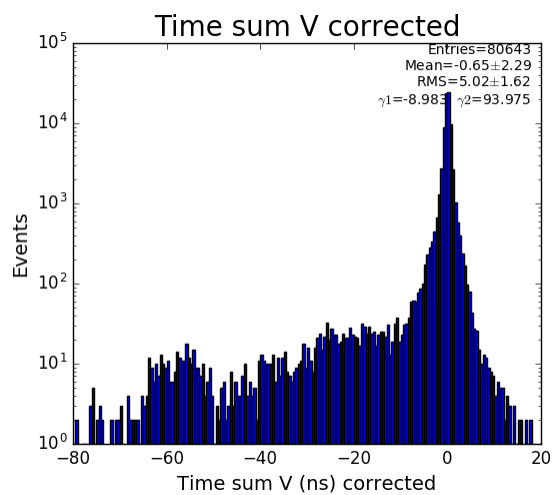
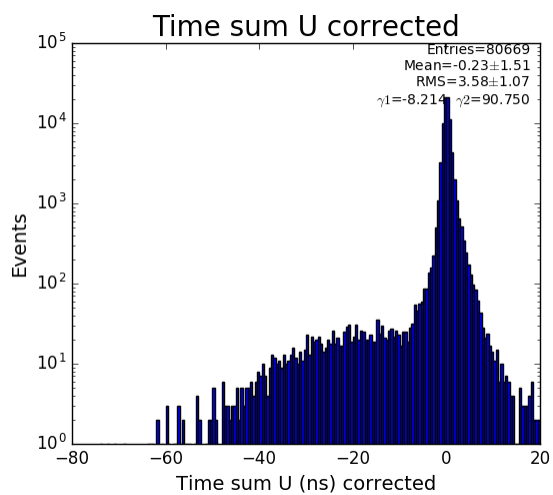
Spectra of Yuv, Yuw, Yvw (mm)



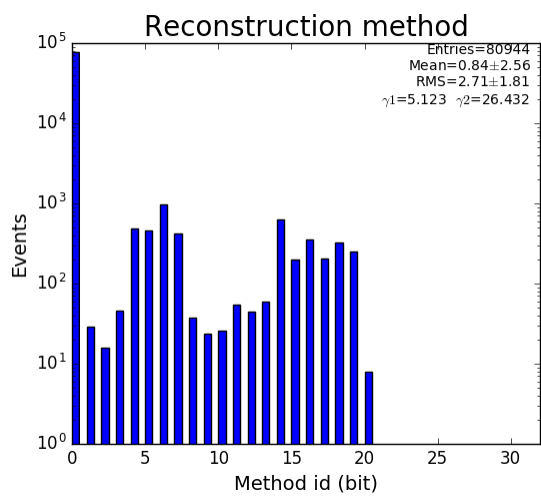
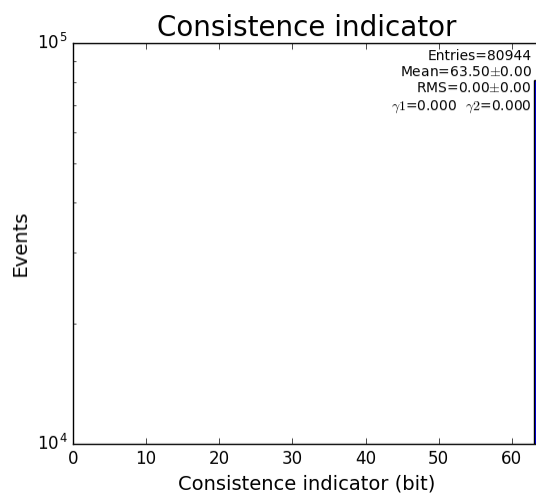
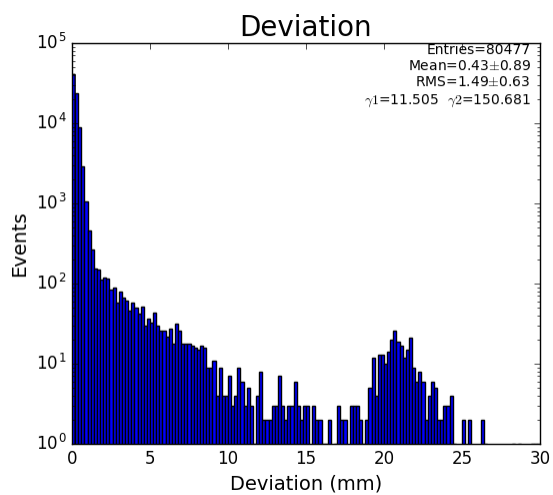
Time sum (ns) for U, V, W



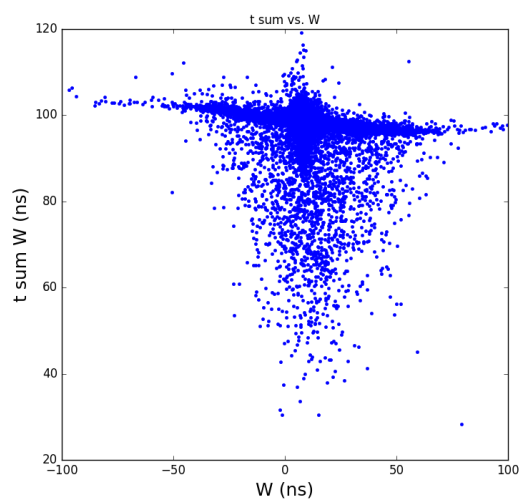
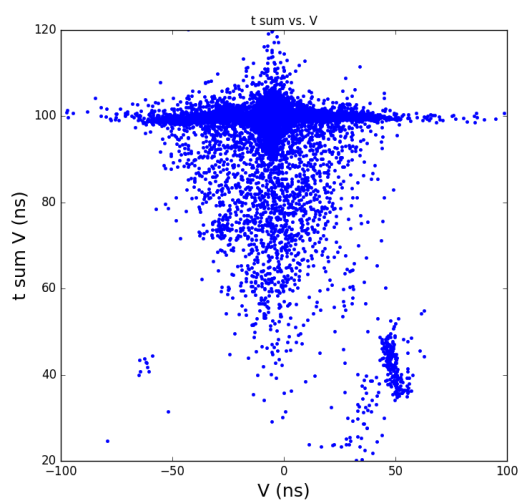
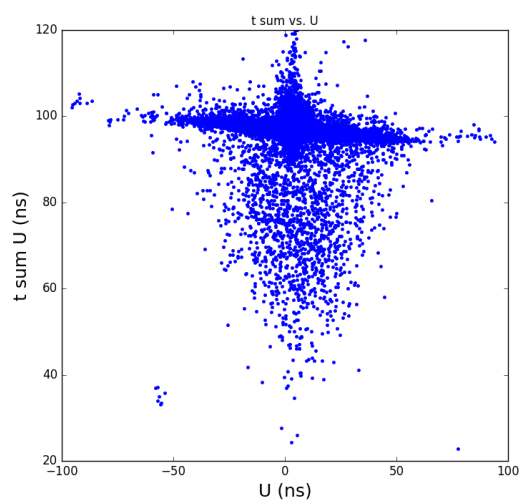
Time sum (ns) corrected for U, V, W

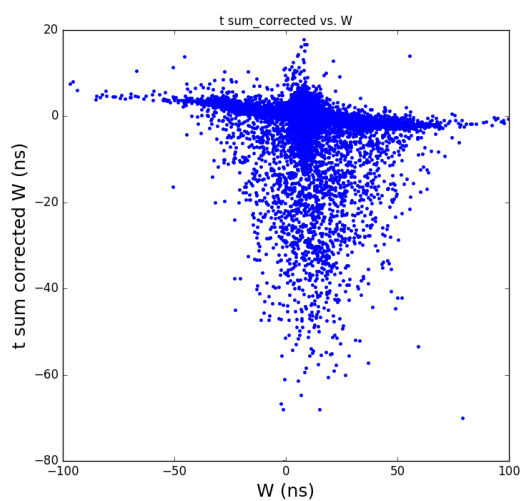
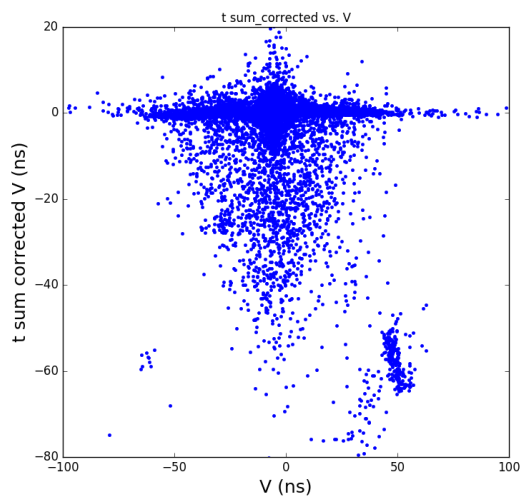
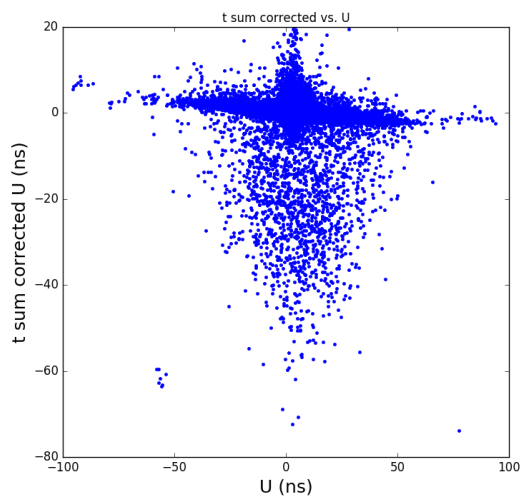


Deviation, Consistency Indicator, Reconstruction method

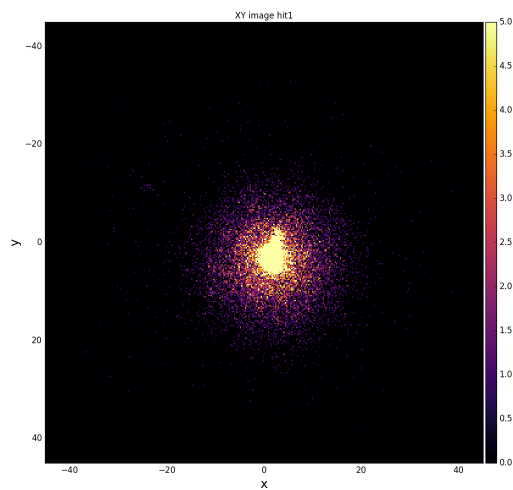
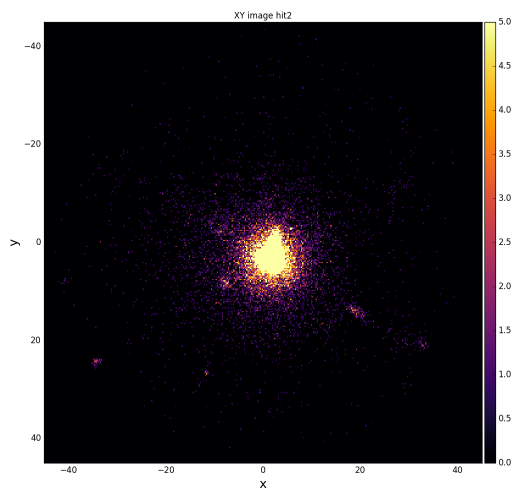


Time sum vs. variable U, V, W

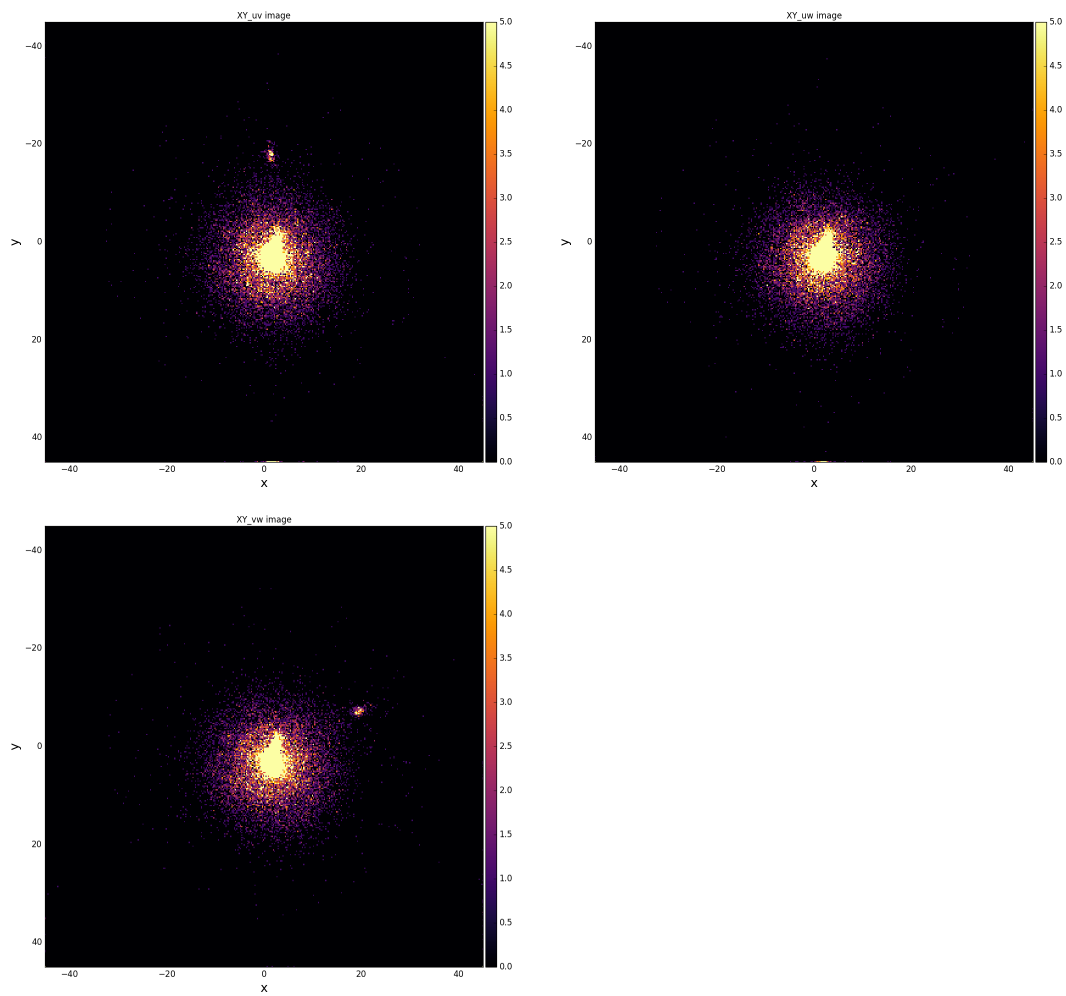




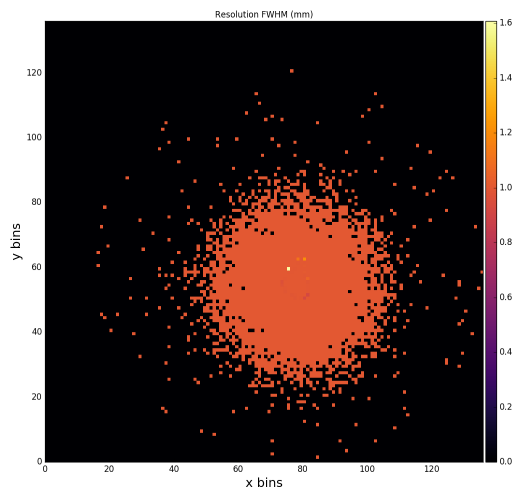
xy image for hit1 and 2



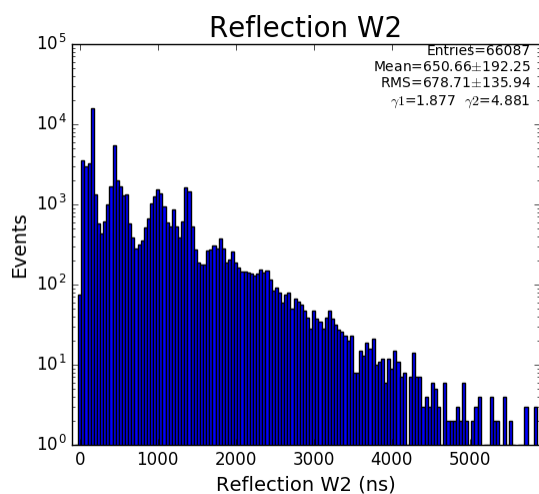
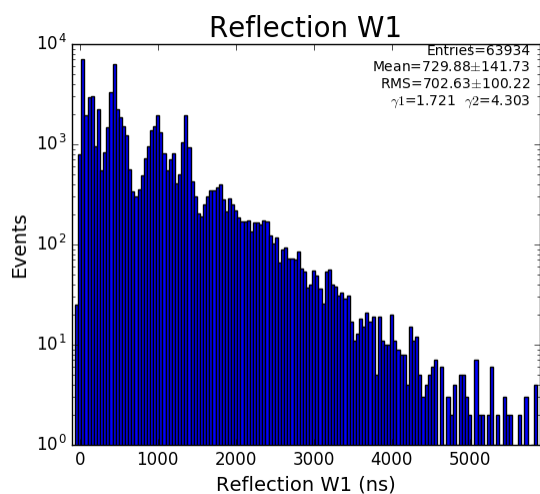
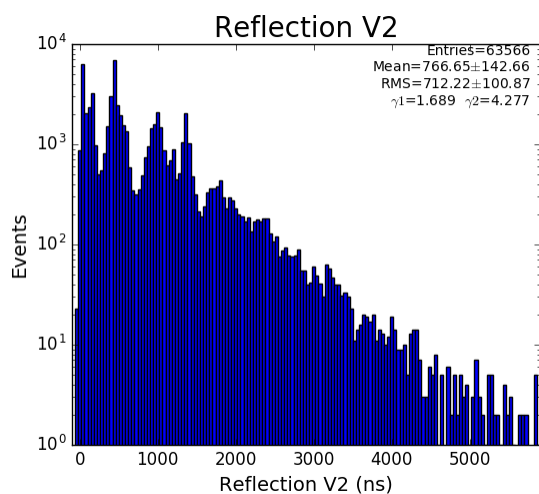
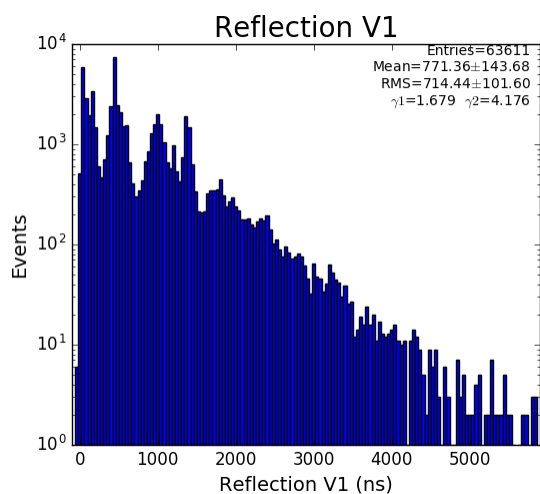
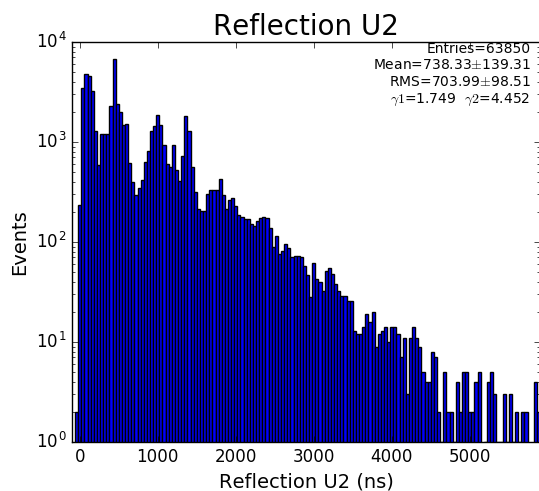
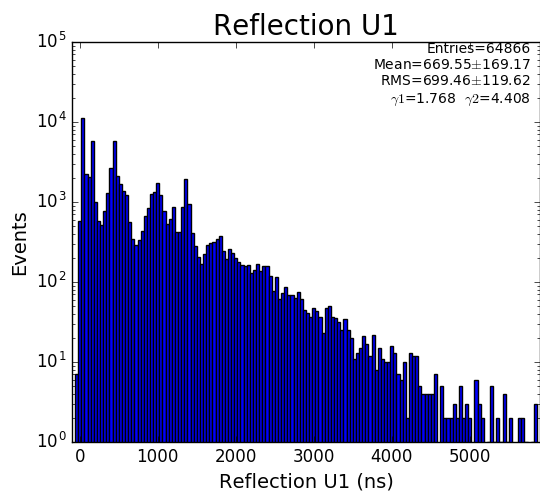
XY image for uv, uw, and vw components



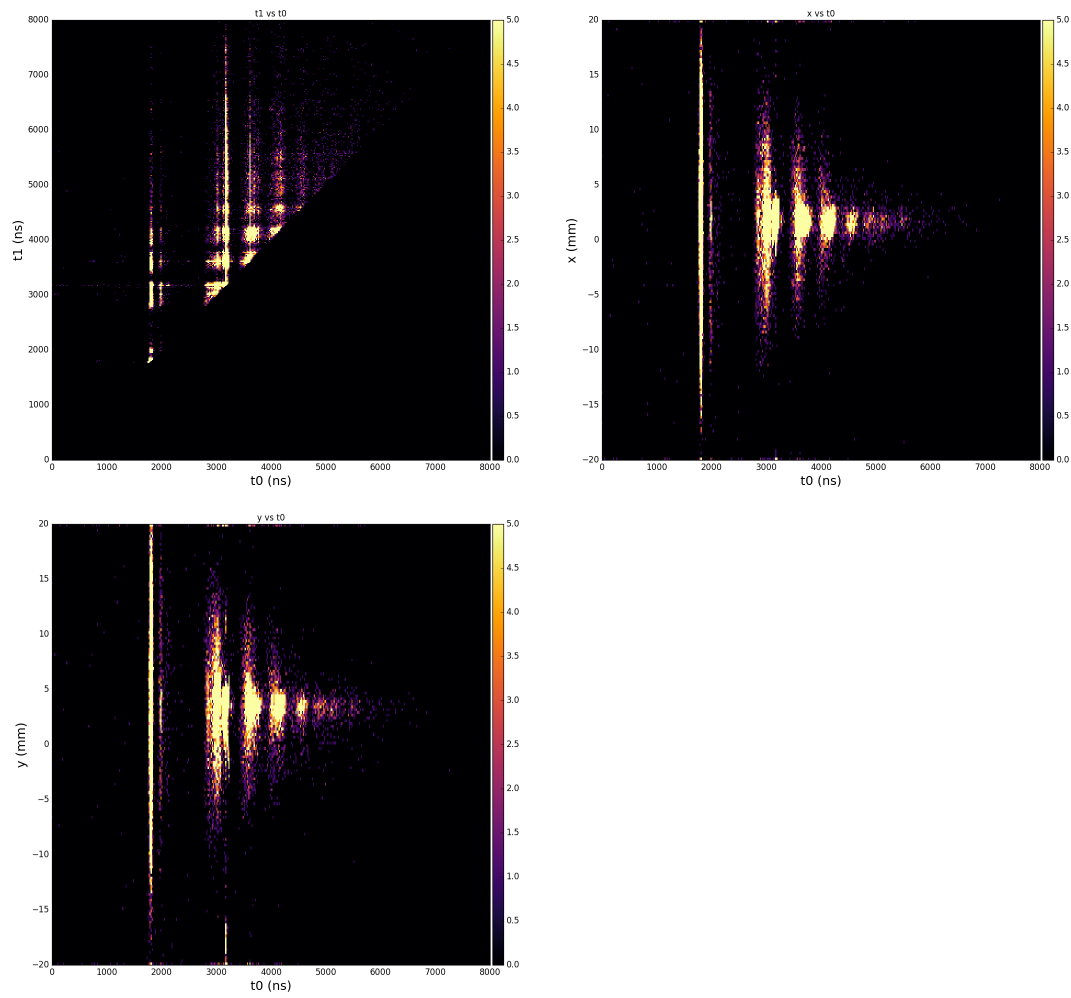
Resolution map



Reflection for all channels



Physics plots t1,x,y vs t0



References

- [2016-11-30-HexAnodeSoftware.pdf](#)
- [2016-12-07-email-achim-czasch.txt](#)
- [2018-01-11-RoentDec-RD01_man01_ja.pdf](#)
- [Building External Packages From Git Tags](#)