

# Start Counter

Viviana Arroyave

October 2023

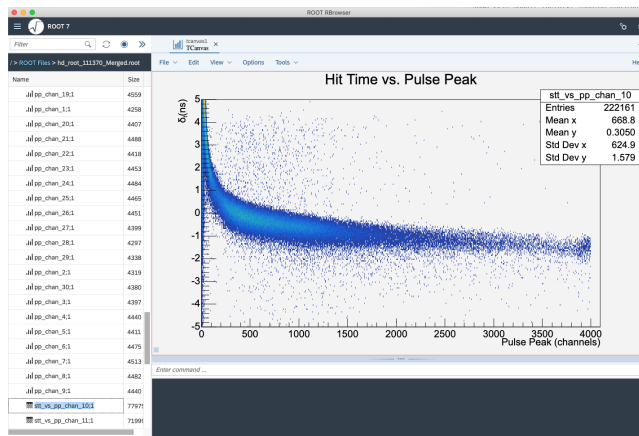
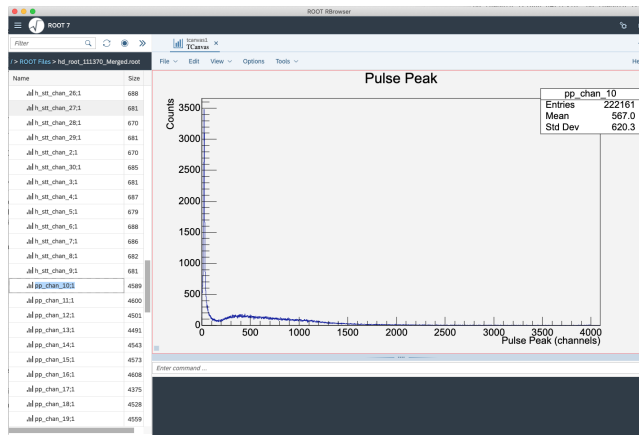
## 1 Timewalk correction

October 2023

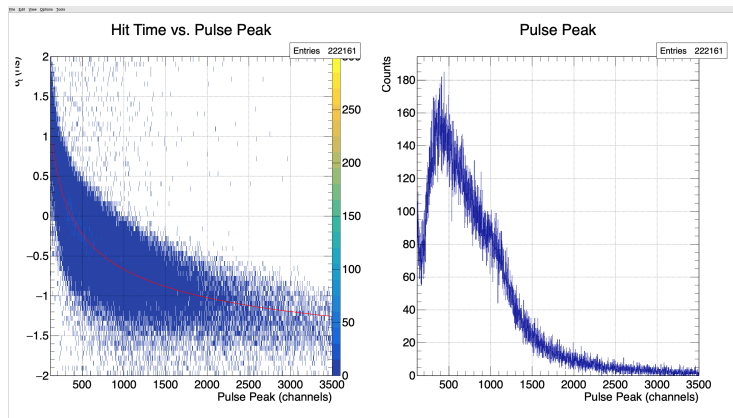
- st\_tw\_corr\_auto
- Calibration using cddb: cddb\_2023.10.16.sqlite
- PrimeX calibration: Run 111370
- Merged 006.evio and 007.evio to get timing distributions.
- path to my files for this calibration: /varroyav/work/StartCounter/PrimeX-2022/
- hd\_root\_111370\_Merged.root

The following histograms are filled when using plugin: "use time-walk correction = 0", the output file is an hd\_root.root file. These are uncorrected timing distributions; In this section you have ST timing which is the difference between TDC and ADC timing, pulse peak (pedestal subtracted). The ST time should look like a bell distribution centered at zero for all the 30 paddles. The Pulse Peak can be seen below and "Hit Time vs Pulse Peak" should have a tail mostly on the zero.

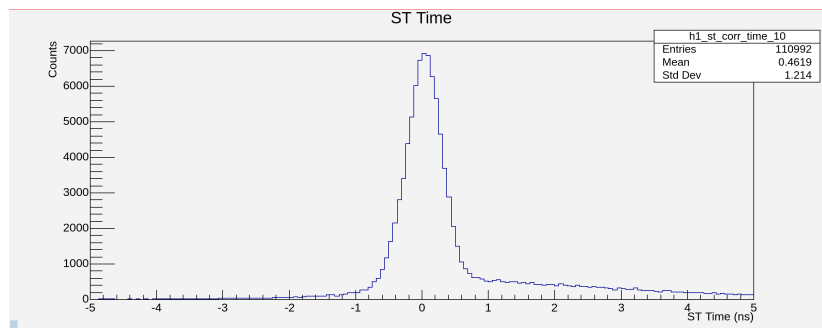


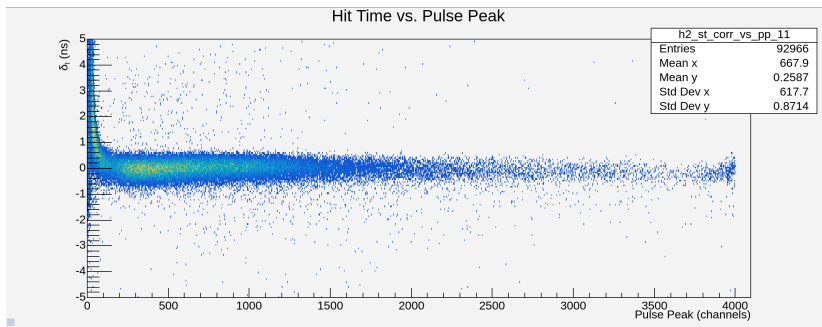
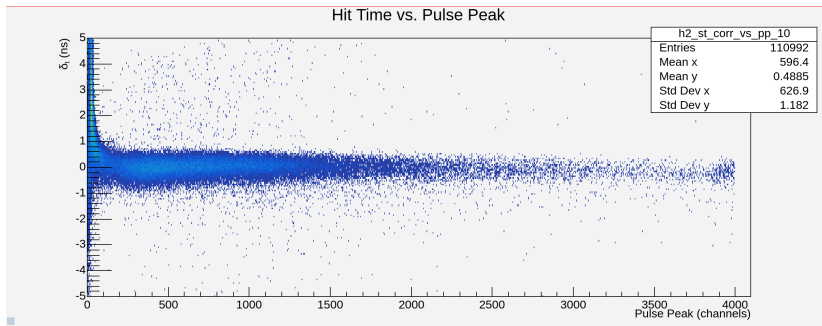


Next step is to use the macro `st_tw_fits.C`. On the left side you are able to see a red line which is fitting the pattern of the distribution. This macro should have created a `st_timewalks.txt` that needs to be uploaded to the database (time-walk correction constants). The table to be filled with these histograms is `START_COUNTER/timewalk-parms-v2`.

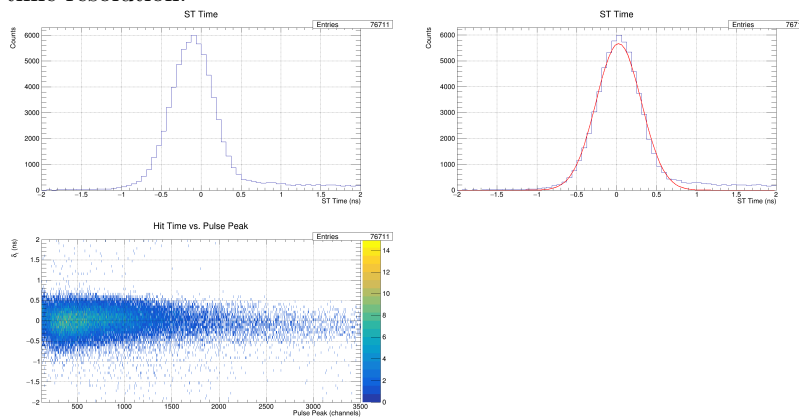


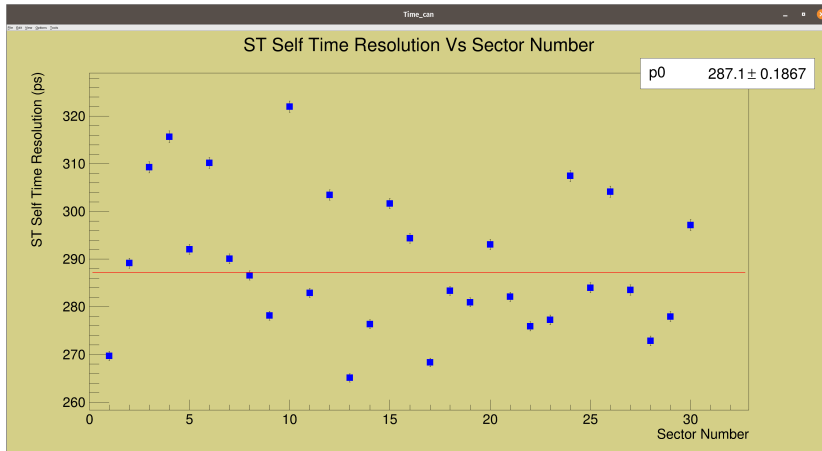
Once constants are uploaded, you can continue to run the plugin again with "use timewalk correction = 1". These should use the constants uploaded before to correct the time walk. This plugin will produce a root file with corrected time distributions; these corrections can be seen in "h1 st corr time" and "h2 st corr vs pp". If the correction was correctly done, then both histograms should have most of the population at zero.





We can use the "st tw resols.C("hd root.root")" to see the corrections and the time resolution:



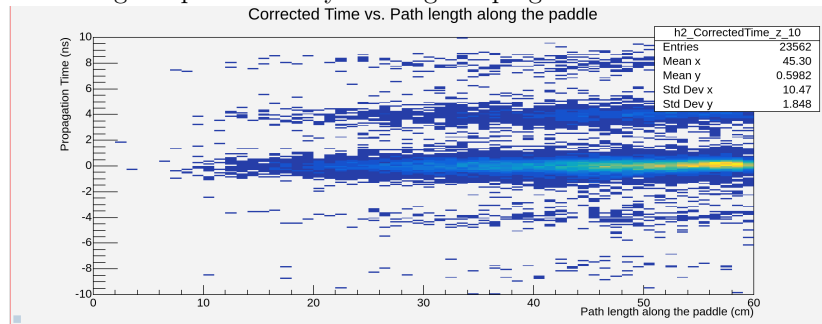


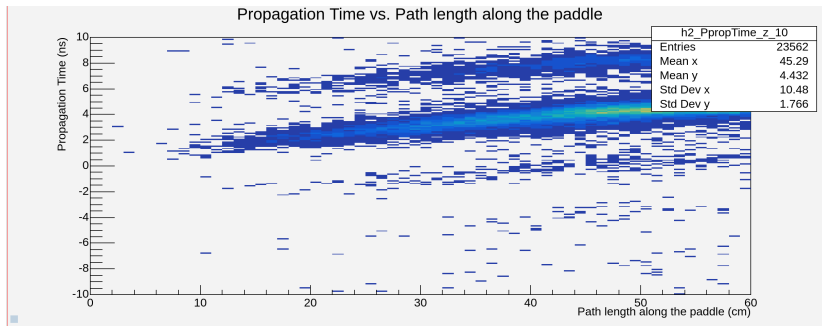
This will create a file: "st time resol.txt ", in this file you have the time resolution per paddle. You are done with timewalk correction here.

## 2 Propagation

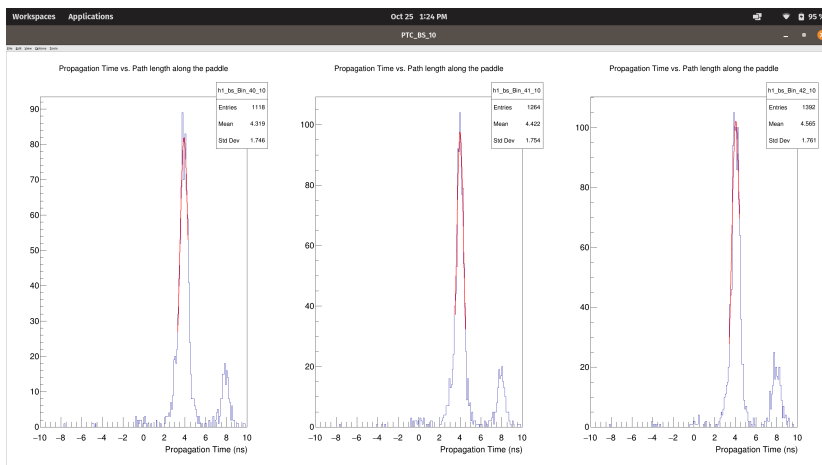
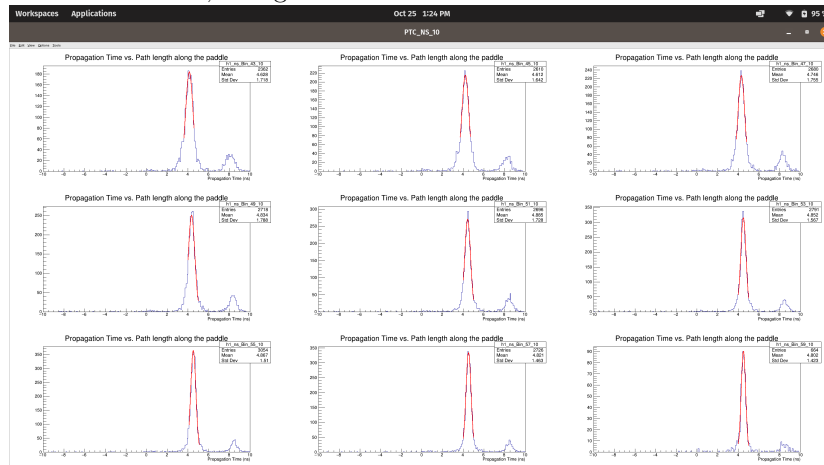
- plugin: ST\_Propagation\_Time

According to the instructions on Github: First, we will run the plugin over the data (here I have done a complete run), in this case : Run number 111370. This plugin will create an hd\_root.root file with 2D histograms of the propagation time vs Z for the three regions : nose section, straight section, bent section). The idea is to be able to add constants on the cddb table: propagation\_time\_corr. The following are plots filled by running the plugin:



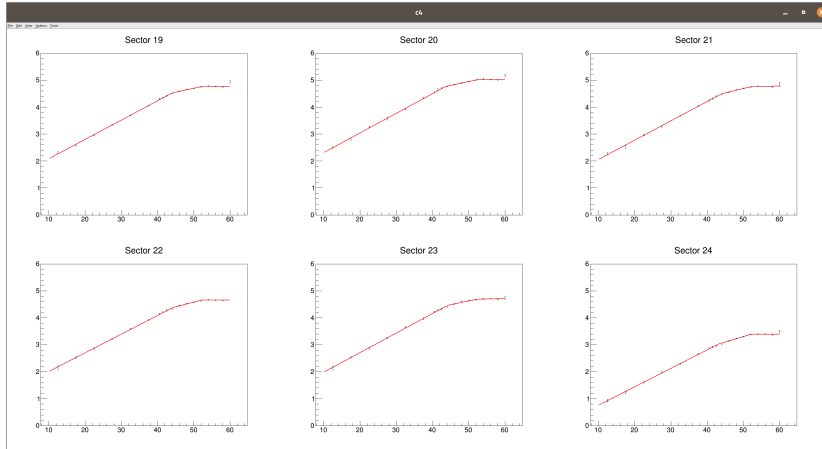


Once the file is obtained, run the macro SC\_PTC.C This macro calculates the mean value of time vs z, the histograms created (90) are separated according to the section : nose, straight and bent.

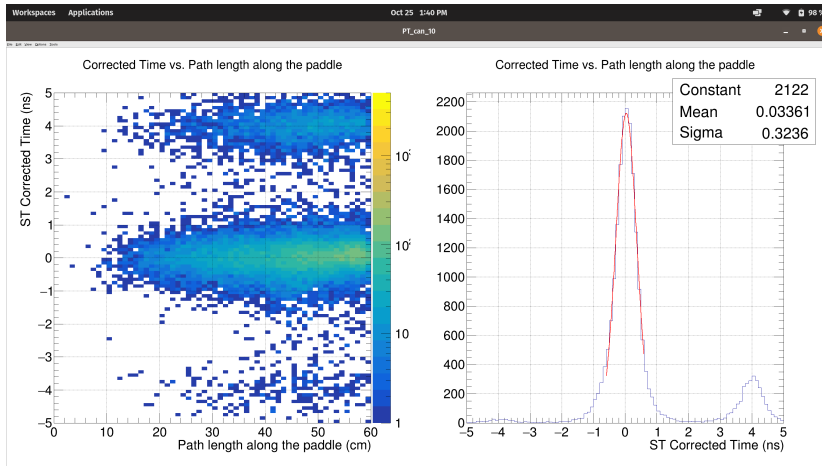


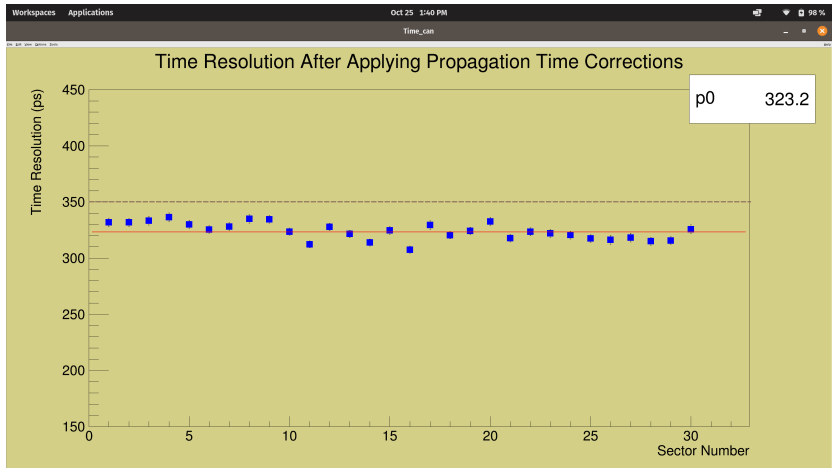
Next step is to run the macro SC\_PTC\_Plot.C, which plots the mean value

from the previous step vs Z. This macro will filled a text file PTC\_1DFits.txt which needs to be uploaded to table propagation\_time\_corr.



To check the time resolution after uploading the constants can be done by running the plugin again obtain the root file again, and then use the macro Resolution.C. To check the time resolution versus Z, you can use the macro SC\_PTC\_RvsZ.C and plotted with SC\_PTC\_ResolvsPlot.C.



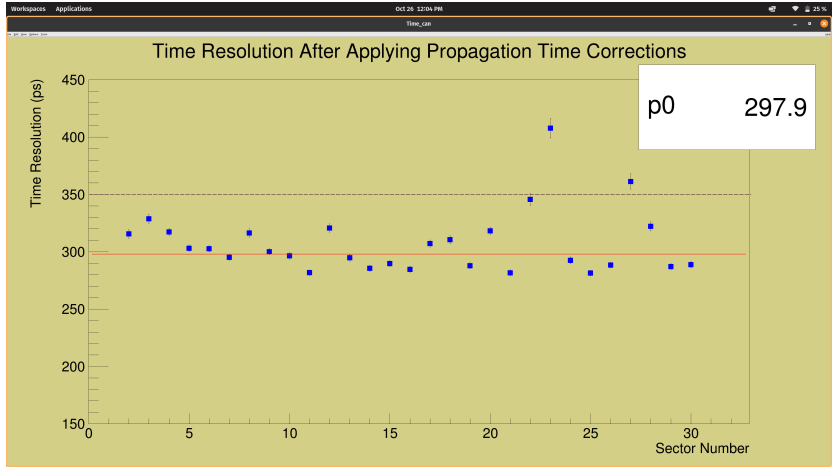


in GitHub, there are two extra files : `st_prop_time_corr_v1.C` and `st_prop_time_v1.C` which are older versions.

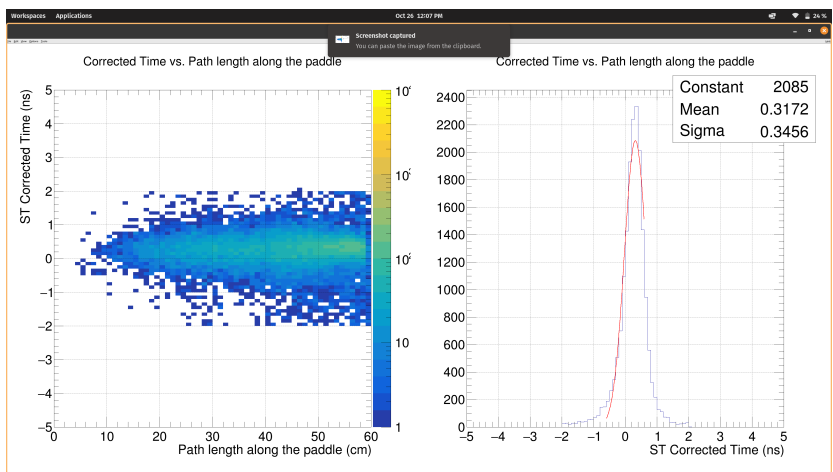
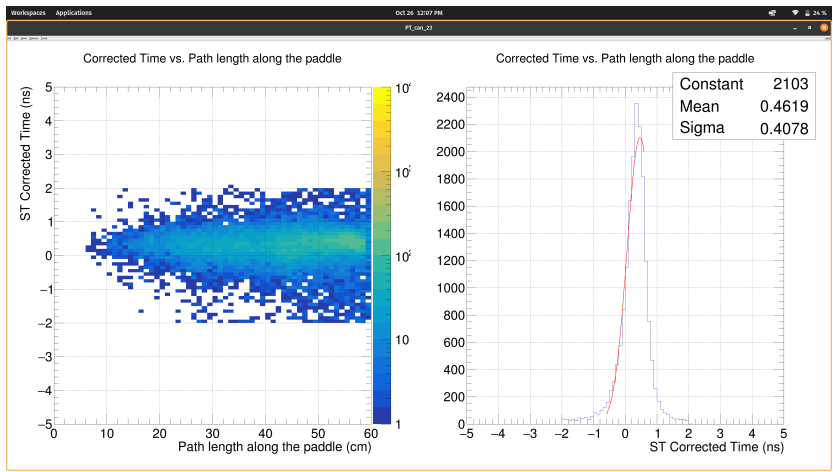
To get the constants you only need to run the plugin and use `SC_PTC.C` and `SC_PTC_Plot.C` macros to get the text file as mentioned above. The rest of the macros allows you to get the resolution.

### 3 T\_Tresolution

Once both tables had been uploaded, we can run the plugin `ST_Tresolution`, this plugin will create a ROOT file which has both time walk and propagations corrected, then you can use the macro `Resolution.C` in order to get the overall resolution:







\*\*\*\*\* Notes: I need to take more files for the propagation constants... 5-10 runs distributed across the run period Once this is done and time res looks better... move on to GluexII