SPEC inc

3022 Sterling Circle – Suite 200, Boulder, CO 80301.
(303) 449-1105 (303) 449-0132(fax)
www.specinc.com

SPEC FFSSP and FCDP

Preliminary
Data Processing Manual

27 October 2012

Table of Contents

Part 1 Instructions

1.1 Raw data files:

The FFSSP and the FCDP produce 5 output files: *scat.bin that contains the main particle by particle data, *sig.bin and *qual.bin that contain full signal and qualifier waveforms for selected events, *hk.txt that contains one Hz housekeeping data, and a *log.txt file, where '*' stands for 'yymmddhhmmss', the date and time of the file start (after about August 2012 the format was changed to 'yymmdd_hhmmss' for some probes). In addition the true air speed (TAS) is also required for processing. Typically the TAS can be acquired from an independent data file at 1 Hz. To integrate the TAS with the FFSSP and FCDP data the user must create an ASCII text file with the name *TAS.txt and put it with the FFSSP or FCDP data files. The software will automatically look for this file to read. The file format is simply two columns, space separated. The first column is seconds since midnight of the day the file was started. The second column is the TAS in ms^{-1}.

1.2 Standard processing to create one Hz ACSII archive files:

The raw files needed for standard processing are the *scat.bin (FFSSP and FCDP) and *hk.txt (FFSSP).

The first step of any processing is to edit the setup options file (setup.m - table 1 shows the parameters that must be set) and then run 'Main_rewrite_dars_scatbin_into_pbp_format_v#.m'. Its primary purpose is to produce two additional quasi raw data files (*scatPBP.bin and *scatPBPindex.bin) for subsequent processing. This step is slow. Its purpose is to allow subsequent quick access to the data. The program also produces an ASCII text file (*scatPBPcount_TMS.txt) of counts per second which can be used for quick look purposes or for time syncing with data from other probes. A figure of the same time series is created and saved as 'Fig00_particle_counts_TMS_hhmmss_hhmmss_S0_nnnn.jpg'.

setup parameter	e.g. values	description
SH_method	adaptive2, none, ...	sets shattering reduction method
agg_cut_thresh	25000	used in shattering removal algorithms
use_constant_TAS	y, n	if 'y' a fixed default TAS will be used
default_TAS	15000	TAS used if 'use_constant_TAS' = 'y'

tt_method	via size, old way, ...	Method of rejection based on transit times
TAS_method	set	there are currently no other options
remuv_noise	y, n	sets whether noise reduction is employed
show_plots	y, n	sets whether plots are generated or not
fishit	y, n	fishing statistic[1] plots are generated or not
probe_type	FFSSP, F_CDFP, uFSSP	sets type of probe data being processed
project	ICE-T, SPARTICUS	sets the project the data is from
noise_string	Boolean logic strings	sets the logic used to reduce noise
BW_tt	0.020 (cm)	property associated with the effective beam width as seen by the signal detector
BW_SV	0.01252 (cm)	associated with the effective beam width as seen by the qualifier detector
BW_SV_plus	0.020/0.626 (cm)	like BW_SV but used prior to transit time qualification
DOF	0.27	Depth of field in cm
ticklen	2.5 x 10^{-8} (s)	the only value used in the probes to date
binmaxv	...	size calibration parameter
binmax	...	size calibration parameter
C1	1.2047e+011	size dependent transit time qualification calibration coefficient
C3	2.2437e+011	size dependent transit time qualification calibration coefficient

Table 1: Parameters set in the setup options file. Green values are the recommended.

After this, there are several options. The direct path to a one Hz archive file is to run 'Main_make_Fast_Fssp_1Hz_archive_file_v#.m'. This should be run only if it is known that the calibrations are working well. If this is not known then the data and calibrations should be quality controlled first (see following section 1.3).

'Main_make_Fast_Fssp_1Hz_archive_file_v#.m' prompts the user to choose the *scatPBPindex.bin created by the previous step. It also prompts to open a TAS file. Since the TAS is not recorded in the FFSSP or FCDP data files, the user must provide the TAS as recorded in an independent aircraft data file. A fixed TAS may also be set when processing bench test data or bead calibrations. WARNING! THE STANDARD PROCESSING ALGORITHMS WILL NOT PERFORM WELL IF THE TAS IS NOT ACCURATE! To integrate the TAS with the FFSSP and FCDP data the user must create an ASCII text file with the name *TAS.txt and put it with the FFSSP or FCDP data files. The software will automatically look for this file. The file format is simply two columns, space separated. The first column is seconds since midnight of the day the file was started. The second column is the TAS in ms^{-1}. The archive file is created in a subdirectory,

[1] (see J. Atmos. Sci., Oct 2010, 3355 – 3367 &/or J. Atmos. Sci., 1992, 49, 387–404).

called ‘results’, of the folder containing the data. The archive contains the sample volume for each second so that the data may be averaged (sample volume weighted average) over longer periods as need to improve statistics. The number of counts in each size bin, and in total, is provided to guide the user in such averaging. Table 2 shows the file's format.

column	name	unit	description
1	Second	seconds of the day	time mark
2	conc(#/L)	#/L	concentration
3	extn(1/km)	Km^{-1}	extinction
4	lwc(mg/L)	g/m3	liquid water content
5	SV(Liter)	Liters	volume sampled
6	totCNTs	number	number of events accepted for sizing
7 - 27	Bin(#/L/um)##	#/L/micron	bin concentrations
28 - 48	NBin##	number	bin counts

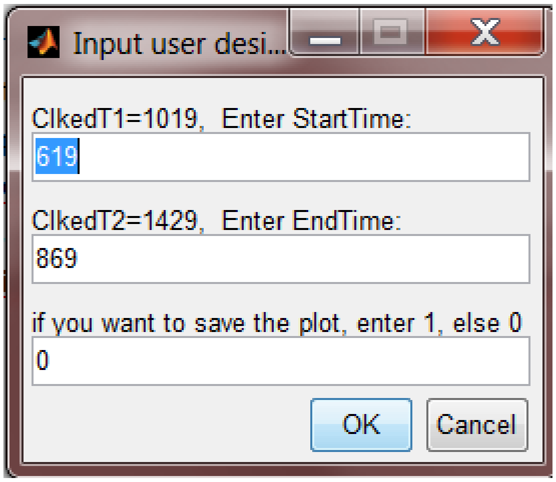
Table 2: FFSSP and FCDP 1 Hz results data file format.

<u>1.3 Other processing options part I</u>:

The one Hz archive file is the main access to the processed data. However, to quality control the data processing[2] and for bench test and calibration data processing, another route is more practical. To learn the time period contained in a data file and approximately the number of detection events each second, run ‘Main_quicklook_Particle_per_second_TMS_v#.m’ or 'Main3_quicklook_Qualified_Particle_per_second_QTMS_v#.m'. When prompted first answer ‘yes’ then ‘no’. A plot is created and you choose a sub period to zoom into for a second figure. You may run it again answering 'no' then 'yes' to zoom in further. Take note of the seconds of

[2] It is imperative that users verify the processing is accurate using this option before batch processing. In particular it should be checked that the transit time - size relationship is working well so that the desired subset of events are being selected by the transit time qualifications steps (section 2.3).

flight (or since file start) values of the start and end of the period you want to process (619 to 869 in the image of the popup shown below, Fig. 1).



ClkedT1=1019, Enter StartTime:
619

ClkedT2=1429, Enter EndTime:
869

if you want to save the plot, enter 1, else 0
0

Figure 1: Popup showing the start and end time (in seconds since file start) of the period to be zoomed.

A selected time period may be processed, using the same core algorithm as used in the standard archive file processing described above, while making various diagnostic plots at each step of the processing. First edit 'Main_GenFSSP_PBP_rawData_frm_T1_to_T2_v#.m', inserting the start and end times of the period into the variables 'Time1' and 'Time2' (currently line 40). Then run 'Main_GenFSSP_PBP_rawData_frm_T1_to_T2_v#.m', which creates '*_frm_#_to_#_PBP.txt'. Then run 'input_file_from_parcel_periods_rv#.m' inputting the '*_frm_#_to_#_PBP.txt' file when prompted and finally run 'process_standard_period_rv#.m', which creates the plots, if 'show_plots' is set to 'y' in setup.m, and the final PSD in a variable called 'PSD_B'. The format of the matrix called PSD_B is 5 columns; column 1 is bin centers, column 2 is bin widths, column 3 is bin counts, column 4 is bin edges, and column 5 is bin concentrations in #/L/um. This is the same PSD as is saved in the 1 Hz batch processed files and has fixed bins determined by the calibration process. Note that if 'show_plots' is set to 'y' in setup.m, figures are produced and the final PSD produced is saved to a file called 'final_DSD_unihist.txt' in a two column format bin edges first and bin concentrations in #/L/um second. Note that this PSD does not have the fixed bins determined by the calibration process but rather variable bin widths so that statistics are more uniform across the domain of the PSD.

.

If 'show_plots' is set to 'y' in setup.m, the following figures are created at the start and after each of the five processing steps discussed in part 2: a distribution of transit times, a distribution of the peak signal voltages, a plot of the DOF acceptance ratio for the F_CDP (ratio of the number of F_CDP in-DOF-events to the total number of events - Qual > Sig), which is also the rejection ratio for the FFSSP and the DOF acceptance ratio for the Hawkeye FCDP (Qual > Sig/2), a distribution of the peak qualifier voltages, a size distribution where each particle is assigned a size linearly interpolated across the standard bins, a distribution of the areas under the signal voltage curves (signal area, i.e. the sum of the discrete voltages), a distribution of the half peak transit times (time from event start to signal peak), scatter plot of half peak transit times versus transit times, scatter plot of signal area versus signal peak, scatter plot of transit times versus signal peaks, scatter plot of half peak transit times versus sizes, scatter plot of the square of the quantity half peak transit times times TAS versus the logarithm of the squared sizes, a plot of the distribution of times between particle detection events. Finally a particle size distribution is plotted and if the variable 'fishit' is set to 'y', the fishing statistic[3] plot is created.

1.4 Calibration processing:

Calibration using glass bead data of various sizes is accomplished via calibrate_glass2water_v#.m. First the bead data must be processed as described above in section 1.3. For bead data the TAS must be manually set in the setup options file[4] such that the desired good events are selected at the transit time qualification step or the required information sometimes be obtained from figures prior to transit time qualification. For each set of bead data of a given size, a peak voltage must be identified (for FFSSP also take note of the laser scale 'lvscale'. This is accomplished via the signal distribution plots at the various stages of processing. A recognizable peak in the distribution must be identified. Matlab's curve fitting options can be helpful for identifying the peak value on these histogram style plots. It is advisable to set 'remuv_noise' = 'n', 'fishit' = 'n', and 'SH_method' = 'none' in setup.m for calibration bead data processing.

The bead sizes and signal peak values (signal peak values times a laser scale value in the case of the FFSSP) are written into the variables x_gmeas and y_gmeas respectively in (calibrate_glass2water_v#.m). Also the variable 'probe' must be set to the correct value (the first lines of code), either 'FFSSP', 'F_CDP', or 'Hawkeye FCDP'. The variable 'mean_lvscale' must

[3] (see J. Atmos. Sci., Oct 2010, 3355 – 3367 &/or J. Atmos. Sci., 1992, 49, 387–404).

[4] This requires rerunning Main_GenFSSP_PBP_rawData_frm_T1_to_T2_v#.m each time the TAS is changed. These steps can be avoided by instead typing TAS = ### (note TAS is input in units cm/s) on the command line after running input_file_from_parcel_periods_rv#.m to change the TAS for that processing run, which is all that is needed for bead data processing.

be set as well. This is done immediately after the variable 'probe' is set. Setting it to the minimum of the laser scale values obtained from the bead data is recommended for the FFSSP (set mean_lvscale = 1 for the FCDP probes). Bins are either optimized for the appropriate Mie curve or ignoring the bumps using a smoothed (quadratic function) Mie curve. In the later case bins may be evenly space, center focused (smaller bins widths near the center of the size range), or small end focused (smaller bin widths at smaller sizes). This is set with the variable 'switch_NB_modes' right after 'mean_lvscale' is set.

The program creates a calibration plot (see Fig. 2) and 2 output files 'calibration_results.txt' and 'calibration_resultsII.txt' that contain the results for bin edge maximums in a nine column, or row, format respectively; 1st, 4th and 7th columns (or rows) in microns, 2nd, 5th and 8th columns (rows) in voltages, and the 3rd, 6th and 9th columns in A-to-D counts. These data are used in the real time configuration file and in the post processing 'sizepbp_rv#.m' file (though set in setup.m). Three options are created, the first three columns are found by using the Mie bumps to choose the bin edges out to a maximum size of about 50 um, the next three columns are found using the quadratic fit to the Mie curve with sizes out to a maximum size of about 50 um, and the third group extends the non Mie bump case out to the maximum size possible given the calibration information. Appendix A discusses the motivation for using the non Mie bump calibrations, though Mie theory does apply.

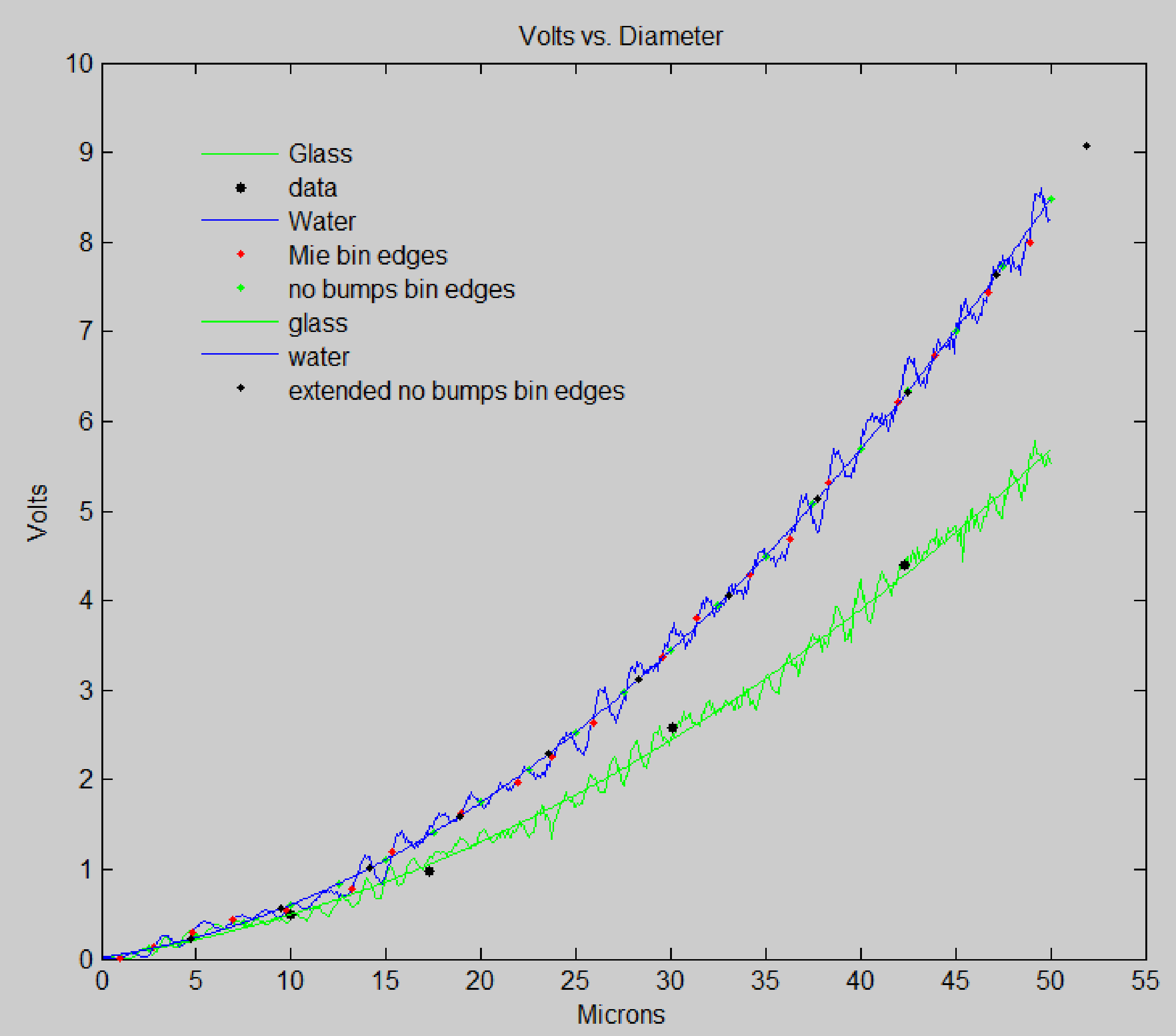


Figure 2: Voltage versus diameter (microns) for water (blue) and glass (green) sphere Mie curves where both have been adjusted by the same multiplicative value, determined to fit the glass curve to the data (black dots) using equal weighting for each data point (the straight average of the adjustment ratio for each point). Also shown are the quadratic fits to the Mie curves. The glass one is used for the fitting, rather than the actual bumpy Mie curve, while the water curve fit is used for the evenly spaced bin edges.

Other calibration factors are needed to calculate the sample volume, namely the Depth of Field (DOF) and beam Width (BW). The BW is also needed for ideal transit time estimates for some older projects that haven't been calibrated using a Gaussian beam profile fit. Otherwise ideal transit times are calculated through two calibtration coefficients C_1 and C_3, which are discussed in the next section and appendix E. All these are set in 'setup.m'. For situations where they differ a BW for calculating sample volume (BW_SV) is set independently of BW_tt, which is used for transit time estimations. Both the BW and DOF may be estimated in the lab. The effective product of the two, the sample area, can also be estimated via modeling the optical system. For the FFSSP, the BW_tt can also be estimated from droplet data via the relationship between transit times, TAS, BW_tt and droplet size (see section 2.3 on transit time rejection).

Part 2 Description of algorithms and examples

2.1 Overview:

Processing takes 4 main steps:

Noise reduction

Shattering reduction

DOF qualification

Transit time qualification

Noise reduction (thresholding the signal and/or transit times) and DOF qualification (signal > qual for the SFFSSP, signal < qual for the F_CDP and signal < 2 x qual for the hawkeye F_CDP) are straight forward.

Shattering reduction is also straight forward but not generally standard, so it will be described below with some examples.

Our transit time qualification is new, non-standard, and makes a significant difference to the results. It will be described with examples.

2.2 Shattering reduction:

When a drop or crystal impacts and shatters it creates a clump of fragments that may move through the sample volume. Figure 3 shows example images of shattered fragments from the 2D-S imaging probe.

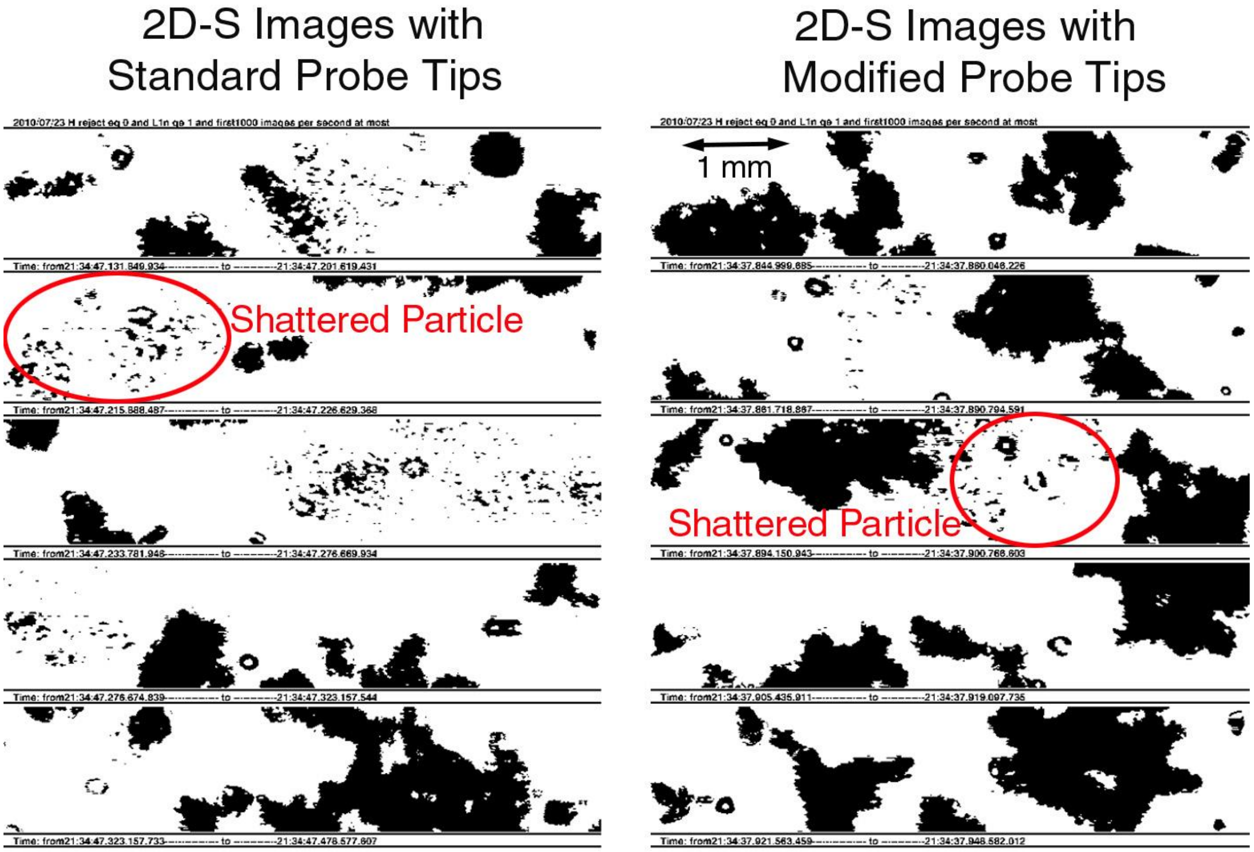

2D-S Images with Standard Probe Tips
2D-S Images with Modified Probe Tips
2010/07/23 H reject eq 0 and L1n ge 1 and first1000 images per second at most
Time: from21:34:47.131.849.934------------ to ---------21:34:47.201.619.431
Shattered Particle
Time: from21:34:47.215.888.487------------ to ---------21:34:47.226.629.368
Time: from21:34:47.233.781.946------------ to ---------21:34:47.276.669.934
Time: from21:34:47.276.674.839------------ to ---------21:34:47.323.157.544
Time: from21:34:47.323.157.733------------ to ---------21:34:47.478.577.607
2010/07/23 H reject eq 0 and L1n ge 1 and first1000 images per second at most
1 mm
Time: from21:34:37.844.999.685------------ to ---------21:34:37.860.046.226
Time: from21:34:37.861.718.887------------ to ---------21:34:37.890.794.591
Shattered Particle
Time: from21:34:37.894.150.943------------ to ---------21:34:37.900.766.603
Time: from21:34:37.905.435.911------------ to ---------21:34:37.919.097.735
Time: from21:34:37.921.563.459------------ to ---------21:34:37.948.582.012

Figure 3: 2D-S Images from 21:34:18–21:37:16UTC on 23 July 2010

In a low density cloud, these fragments are more closely spaced then the natural particles and thus can be identified and removed based on their spacing, as long as two or more fragments are detected from a given shattering event. That is, if a particle's nearest neighbor is closer than a specified cutoff, the particle is rejected. The rare natural particles that are removed by this criterion are accounted for by using standard statistics. In a high density cloud the natural particle spacing can be similar to the shattered fragment spacing and thus the spurious fragments cannot be removed this way (if the fragments are moving slow enough compared to the air speed through the probe, they may be removed using the transit time qualification). However in this case the effect of shattering is small compared to the natural concentration. To optimize shattering removal versus retaining real particles with varying concentrations, the cutoff distance for rejection is varied. The maximum cutoff is set at 25,000 clock ticks (625 us) or about 10 cm at the typical jet true air speed (TAS) of 160 m/s. This is based on 2D-S images of shattering events and on the results of applying the fishing test to the droplet spacing data (Fig. 4).

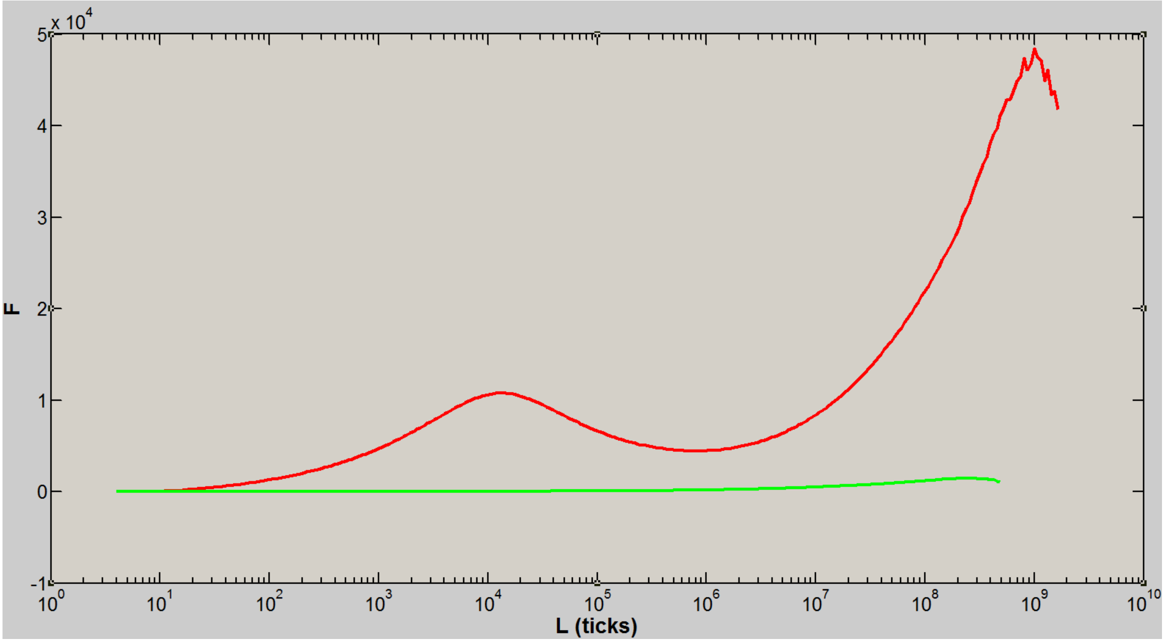


Figure 4: The results of the fishing test applied just before (red) and after (green) the shattering removal processing step. The data are from a period of large ice aggregates from 21:34:18–21:37:16UTC on 23 July 2010 (SPARTICUS).

Figure 4 shows the fishing before test and after the shattering reduction step. The peak at about 12,500 ticks before shattering reduction indicates that there are clumps of particles on a scale of about 25,000 ticks, or 10 cm, in this case. The reduction in that same peak after shattering reduction indicates a significant shattering reduction effect has been achieved. The distribution of inter-arrival times before and after shattering reduction (Fig. 5) also demonstrates the effect that has been achieved.

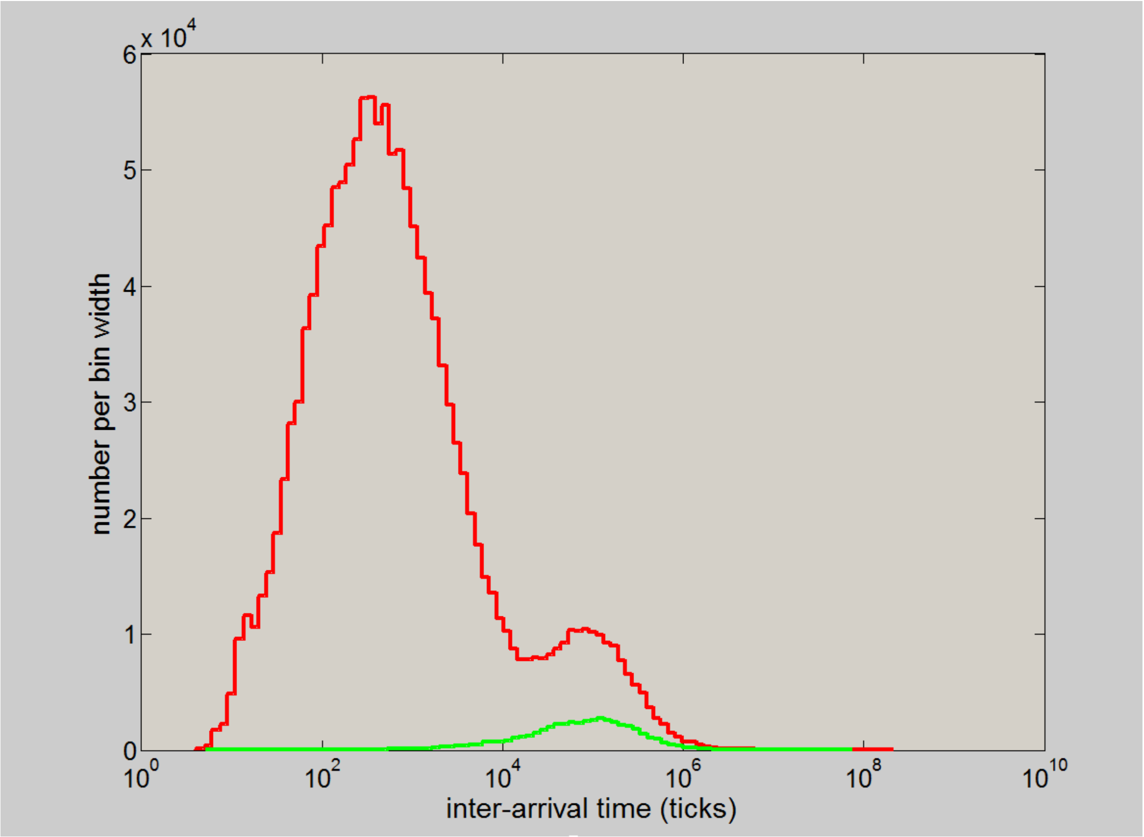


Figure 5: The inter-arrival time distributions before (red) and after (green) the shattering removal processing step. The data are from a period of large ice aggregates from 21:34:18–21:37:16UTC on 23 July 2010 (SPARTICUS).

While acknowledging that significant reduction in the effects of shattering is achieved here (Fig. 6), it is important to emphasize, especially in cases such as this where the majority of detection events are rejected as shatter fragments, that the remaining events may still be partially or entirely shatter fragments that were not removed (if only one fragment passed through the sample volume for example).

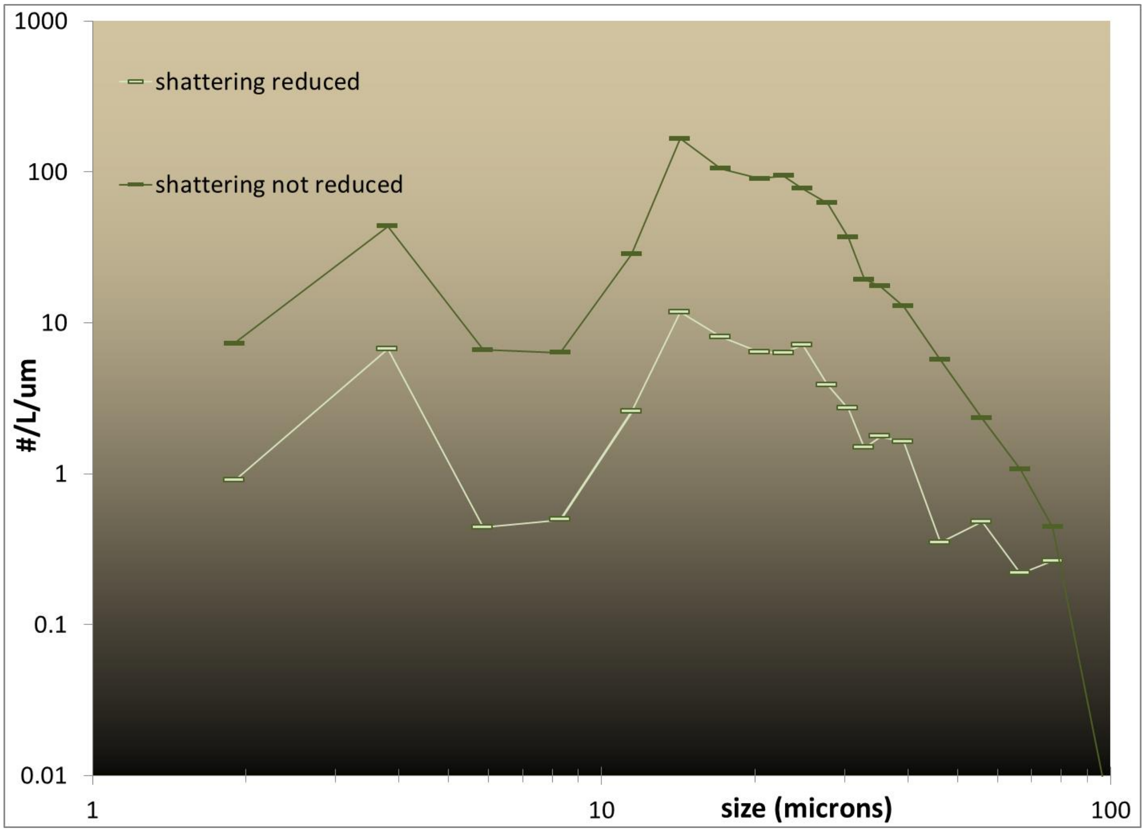


Figure 6: the final PSD produced with and without using the shattering reduction processing step. The data are from a period of large ice aggregates from 21:34:18–21:37:16UTC on 23 July 2010 (SPARTICUS).

As stated above the maximum cutoff used is 25,000 ticks. The minimum cutoff used is the value such that, given the average inter-arrival time of the data, 20% of the events would be rejected, if the data were homogeneous (unless such value would be > 25,000, in which case the cutoff is 25,000). The value used may be anywhere between depending on the situation. The minimum is initially used. If the data are homogeneous, the resulting concentration will not change significantly from the value found without shattering reduction, because of the statistical correction. If however the data are clumped, e.g. due to shattering, more than 20% of the events will be rejected and the resulting concentration will be lower than the value found without shattering reduction. If it is lower by 20% or more, the cutoff is reassigned using the new average inter-arrival time (determined after the shattering reduction step using the previous cutoff) and the process repeated. This iteration continues until the change in concentration is less than 20%. We call this algorithm ‘adaptive’ since it adapts to the situation. An option to use a fixed cutoff threshold of 25,000 ticks we call ‘aggressive’. Figs. 9, 10 and 11 shows additional comparative results using the various shattering removal options.

2.3 Transit Time Qualification:

The first transit time qualification is step compares the half peak transit times (half peak transit time is the time from the event start to the peak signal value) with the full transit times. To qualify the half peak transit time must be between 0.33 and 0.60 times the full transit time. For a smooth symmetric pulse the time to the peak, i.e. the half peak transit time, should be just a bit under half the full transit time, due to the event ending threshold being below the event initiating threshold. Hence the step helps eliminate erratic non-symmetric events and some coincidence events.

The next transit time rejection steps are slightly different for the FFSSP and F_CDP. The transit time qualification serves two purposes for the FSSPs and one for the FCDPs. For both FCDPs and FSSPs it is used to disqualify coincident events (too long transit times) since their sizing and counting will be in error. For the FSSPs it is also used to disqualify events for which the particle did not go through enough of the beam (passed through beam edge only - transit time too short) as sizing of those events will be poor as well.

For FSSPs, transit time qualification was originally implemented to remove the detection events for which a particle passes through the edge of the beam, which results in under sizing and a shorter transit time than a particle passing through the beam center. It is implemented by only accepting events for which the transit time is greater than the mean transit time, theoretically, for a cylindrical beam, this is 0.626 of the events. It works very well when the concentration is low and there are only small particles, especially if the size distribution is narrow. However when coincidence or large particles cause longer than normal transit times, the mean is biased causing the rejection of valid events while accepting the too long transit time events, which tend to be oversized. Similarly, even without coincidence, the technique biases towards large particles since they cause longer transit times. We apply a transit time rejection that is particle size dependent and thus reduces those biases. An ideal, or expected, transit time is calculated for each particle based on its size, the TAS and the probe's unique beam characteristics, which at this time are incorporated through two calibration coefficients C_1 and C_3. A particle is accepted if its transit time is within 0.785 to 1/0.95 of the ideal transit time for the FFSSPs and if its transit time is less than 1/0.95 of the ideal transit time for the FCDPs.

In the following it will be shown:

1) That the transit time measurements, even for ice, are accurate enough to detect that longer transit times are associated with larger particles and that the values are in reasonable

theoretical agreement with the top hat beam profile model for the given TAS, beam width, and particle size.

2) Despite 1) above, the top hat model is inadequate and a Gaussian model performs much better. Assuming a Gaussian beam and scattering proportional the square of the particle size, a linear relationship is expected between the square of the quantity half peak transit times times TAS and the logarithm of the squared sizes. This forms the basis for fitting data to the Gaussian model, which is done via the routine fitcp.m (see tutorial in appendix E). We have found that such a calibration holds reasonably well during a given project if the thresholds are not changed or the probe otherwise altered. However it is expected that the relationship will change and the coefficients must be re-determined if the thresholds or other probe characteristics are changed.

3) That the size dependent transit time qualification algorithm reduces the biases discussed above.

– – – – –

2.3.1 Transit time – size relationship:

The time period analyzed here contains ice particles, both small and large. Fig. 7 shows CPI images. The transit time – size relationship should be stronger for water droplets but demonstrating it for ice is good in that it shows that it is still operable for ice. 0.785 is the mean transit time for a cylindrical beam of diameter 1 and 0.626 is the fraction of transit times larger than the mean. The transit time distribution for a given size drop should peak at the transit time of a drop going through the center of the beam. Table 3 and Fig. 8 show that except for the smallest signals, which are presumably way off because they are dominated by events that went through the edge, the estimated ideal transit times are reasonably realistic.

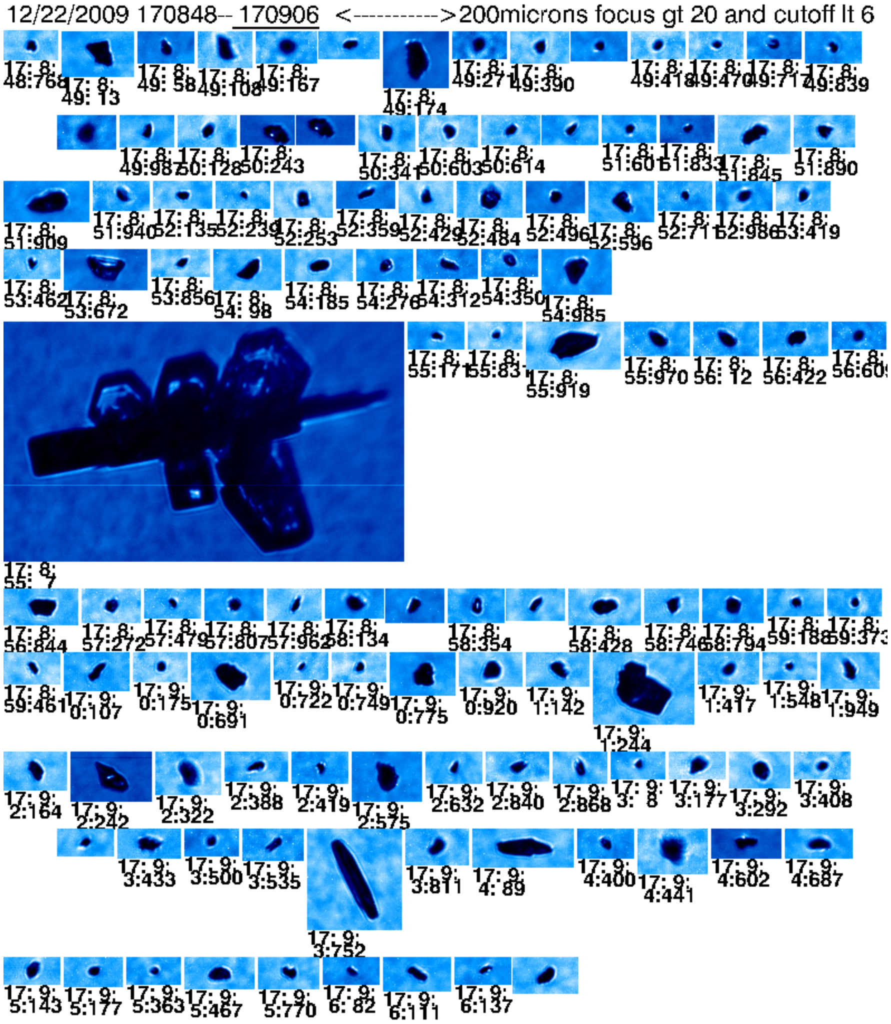


Figure 7: CPI images of ice particles observed from 170848 – 170906 on 22 December 2009.

size range microns	< 5	5 – 13	13 – 24	24 – 34	34 – 42	42 – 61	61 - 82
nominal size microns	5	9	18	29	38	51	71
transit time estimate in ticks	61	63	68	75	80	87	99
mode transit time (mtt) in ticks	30 & 47	60	68	76	78	86	92
mtt after ½ peak rejection in ticks	30 & 48	57	69	75	77	83	88
mtt after shattering rejection in ticks	47	55	65	73	73	85	82

Table 3: Expected (assuming top hat beam profile) and observed mode transit times for a range of particle sizes. Data from 22 December 2009, 17:08:20-17:10:20 (FSSP time), after the DOF rejection step. 5th row includes an additional reject step (half_peak_trans_time <= 1.15*tt/2 & half_peak_trans_time >= 0.7*tt/2). 6th row includes aggressive shattering removal, which rejects the majority of events (see Fig. 7).

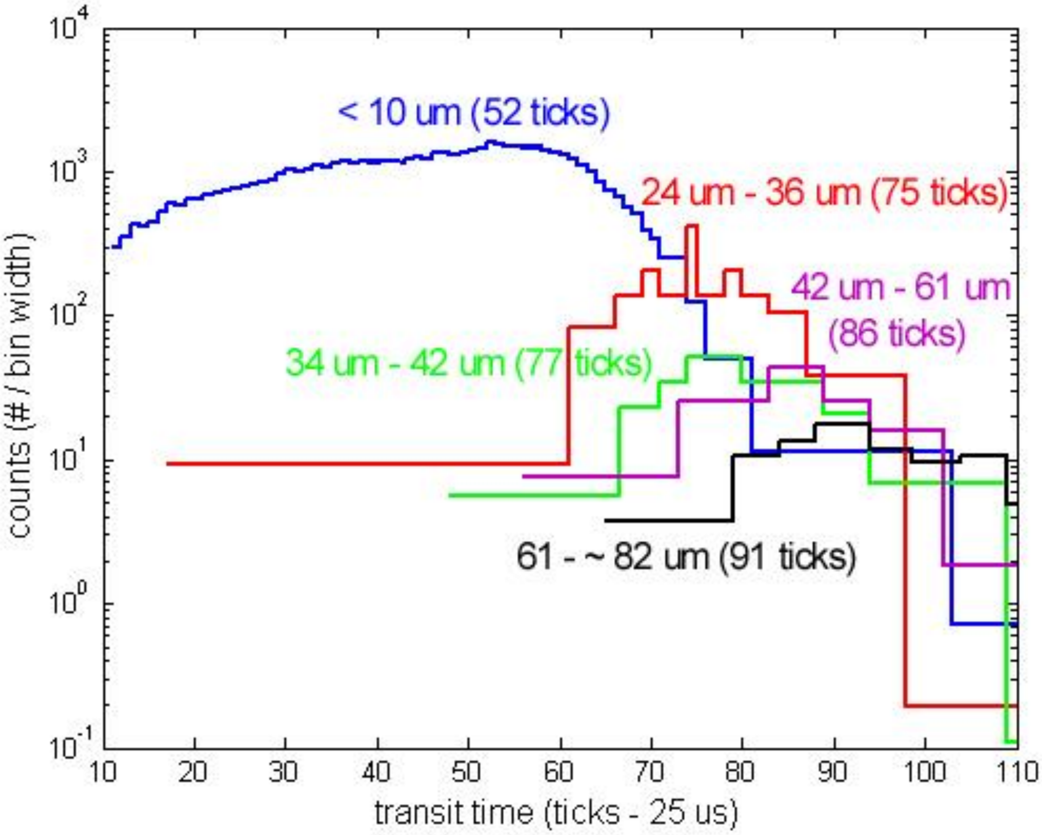


Figure 8: Data from 22 December 2009, 17:08:20-17:10:20 (FSSP time). Transit time distributions for various size ranges created after the DOF rejection step, without shattering removal. The size ranges and location of the peaks are indicated.

2.3.2 top hat model replaced with Gaussian beam profile model:

To demonstrate which beam model yields more effective transit time estimates we plot the measured transit times versus measured sizes along with estimates of the transit times versus size assuming the different beam profile models (top hat and Gaussian). Figure 9 shows an example using the Gaussian model from a FCDP which shows a tight relationship between size and transit time, after qualification by the slit optics comparator, because coincidence (the sparsely scattered blue colored points above line defined by the yellow points) is minimal and the slit configuration eliminates non beam-centered events, which have short transit times, as well as providing DOF qualification. The ideal transit times (yellow points) used for qualifying the real events are targeted (see appendix E) to lie at the top edge of the bulk of good events as shown in this example).

For the FFSSP (Fig. 10), the fit between the top of the dense data and the Gaussian model estimates is similar but not as good as for the FCDP (despite and presumably because of the different beam conditioning). Events where the particle passes through the edge of the beam and are thus too short are not eliminated by the DOF comparator as they are for the FCDP and thus must be removed by the size dependent transit time qualification. The top of the dense data region defines the transit time versus size relationship for beam centered events. Figure 11 shows the much poorer agreement with the top hat profile beam model.

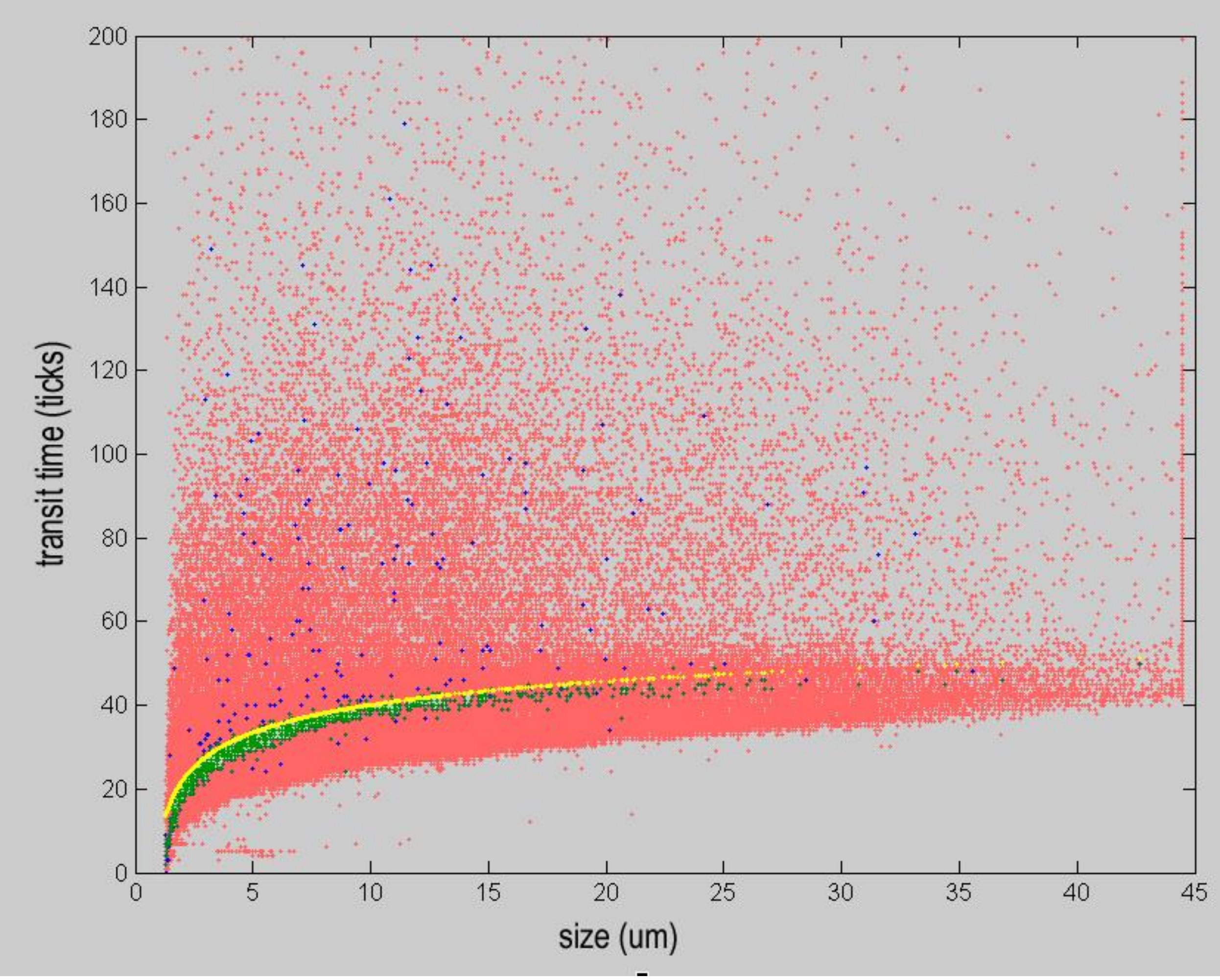


Figure 9: Half peak transit times versus size for the ICE-T FCDP research flight #4 on 7-12-2012 1441 to 1444 seconds of file (file RF4_7-12-2011 110712175445_frm_1441_to_1444_PBP). All

events are plotted as rose colored points, slit optics qualified events are overlaid in blue with both slit optics and transit time qualified events overlaid again in green. The yellow points show the parameterized ideal transit times as a function of size (and air speed).

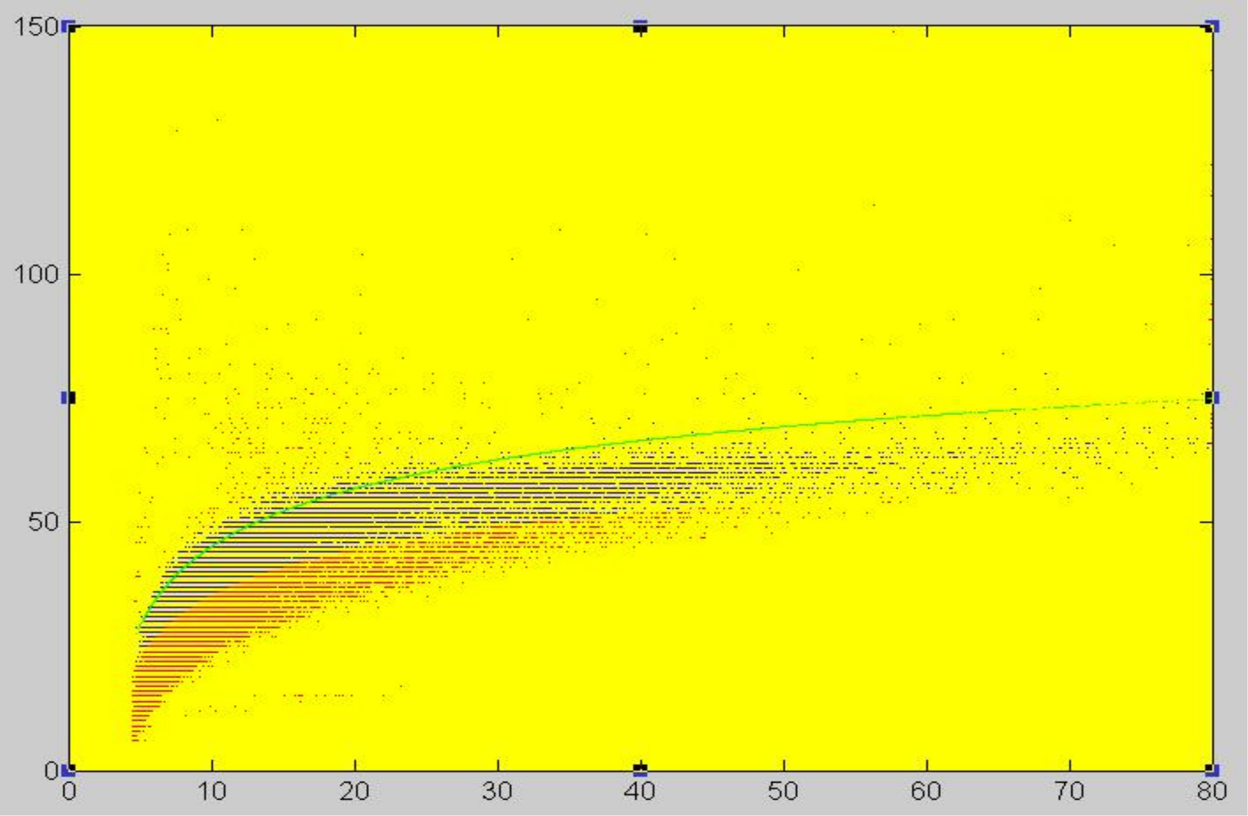


Figure 10: Half peak transit times versus size for FFSSP ICE-T data from 19 July 2011, 1431 to 1438 seconds of file (file name : 'good_eg_4_tt_rej_110719190402_frm_1431_to_1438_PBP.txt'). Gaussian beam profile model ideal transit time estimates shown in green. Red are all events passing DOF qualification and half peak vs full transit time qualification while those passing the size dependent transit time qualification as well are shown overlaid in blue. 34.4% of the events are removed as too small of transit times compared with an expectation of 38.4%. Note that with only a minority of events being coincident (red points above the green line), they are easy to identify by eye and most are removed as too long of transit times by this procedure.

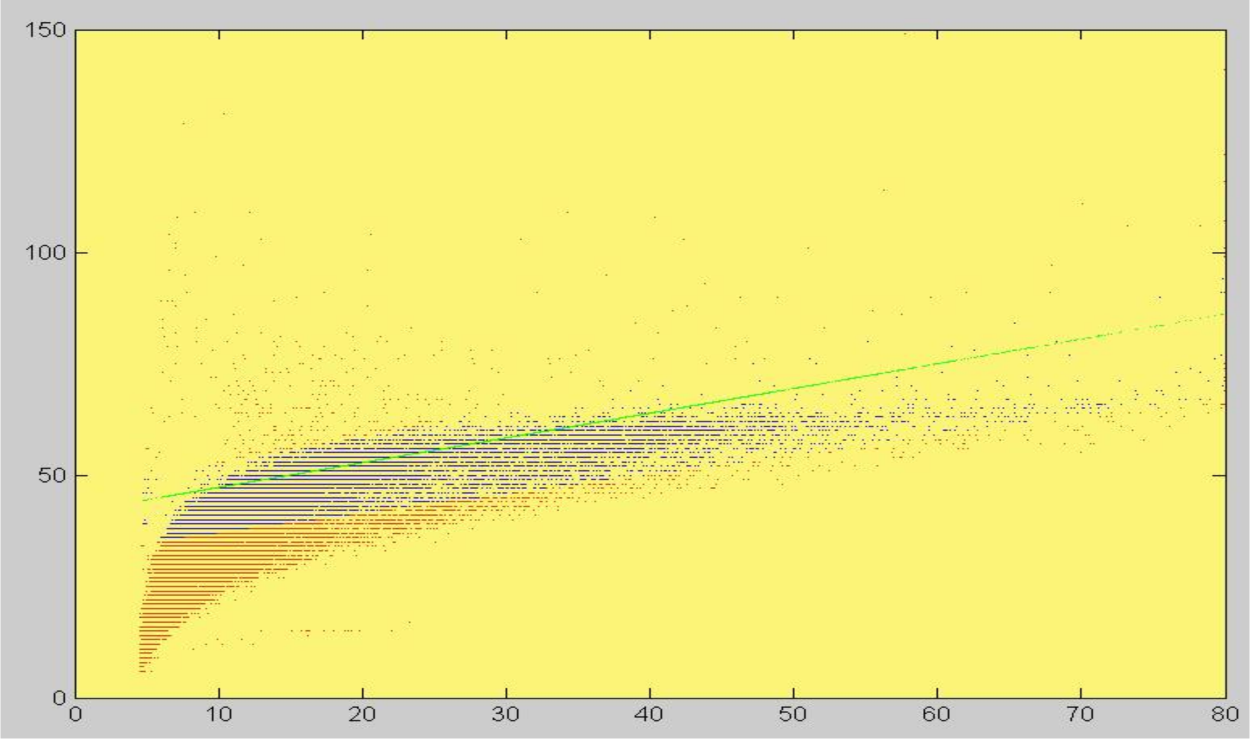


Figure 11: : Half peak transit times versus size for FFSSP ICE-T data from 19 July 2011, 1431 to 1438 seconds of file, like figure 10 above but for a top hat beam profile model instead of the Gaussian beam model. It can be seen that there is room for improvement as the data is clearly nonlinear and the parameterization linear, though it works fairly well anyway in this case.

<u>2.3.3 size dependent transit time qualification reduces biases</u>:

Figures 12 and 13 show the expected small effect that the various transit-time rejection options have on the PSD when the mean is not significantly biased.

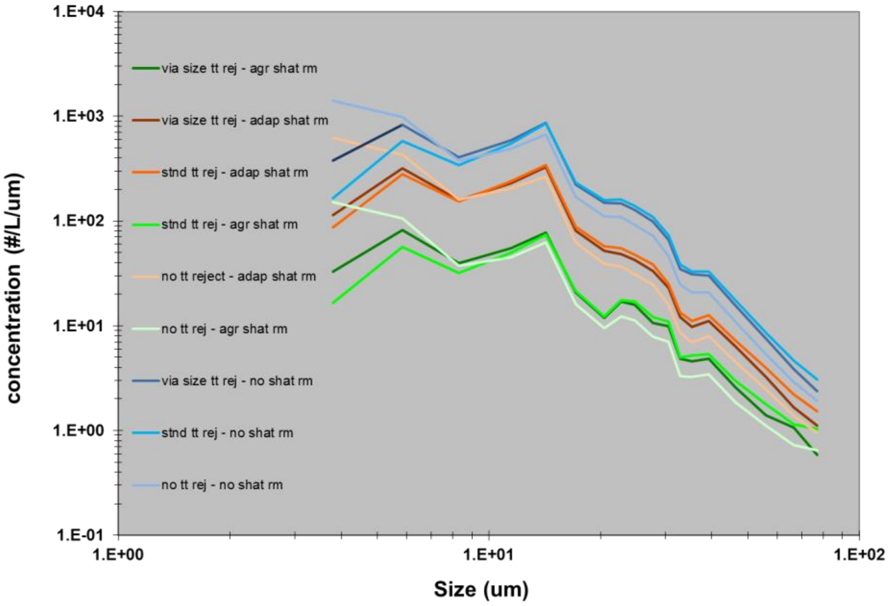


Figure 12: PSDs using various transit-time and shattering rejection options for the same data shown in Figs. 7 and 8.

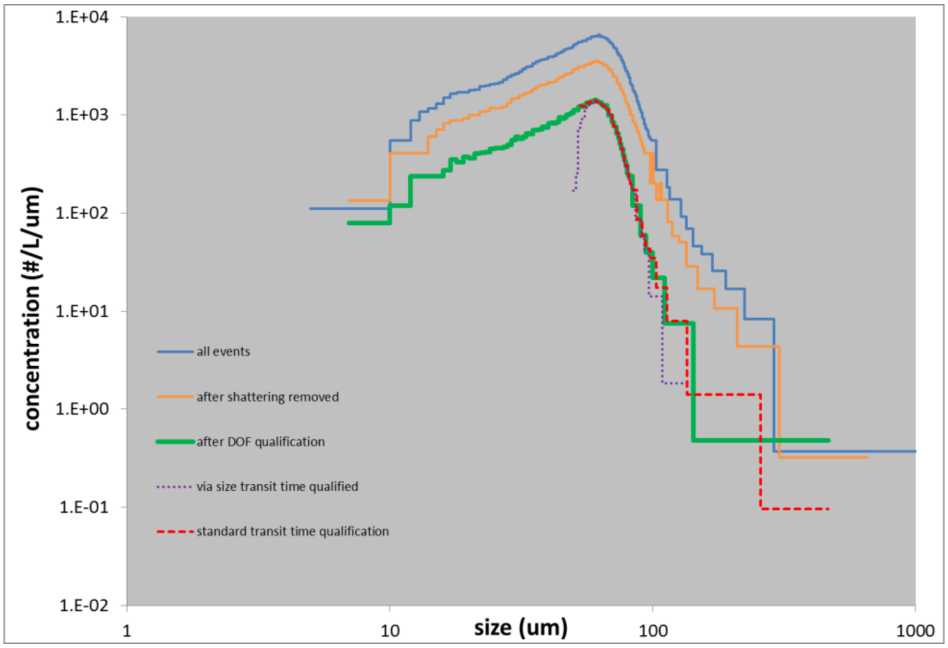


Figure 13: transit time distributions at various stages of the processing and for each of the final transit time qualification steps; via size and standard, for the same data as in Figs. 7, 8, and 12.

The next example shows data from a period of large ice aggregates from 21:34:18–21:37:16UTC on 23 July 2010. 2D-S images are shown in Fig. 3. Adaptive shattering removal runs 4 iterations where it reaches the maximum cutoff threshold of 25,000. This seems to remove most of the shattering as indicated by the fishing test and inter-arrival time distributions before and after the shattering removal step (Fig. 14). This however does not unequivocally imply the remaining events are not also shattering remnants. Figure 15 shows the results of processing four ways (with and without shattering removal (indicated as adapt or none respectfully) plus transit time qualification via size and the standard way) along with the 2D-S PSDs. Figures 16 and 17 show that for this case with a heavily biased mean transit time, the new size dependent transit time qualification chooses better events (those at the peak of the distribution) for sizing than the standard transit-time-greater-than-the-mean qualification does.

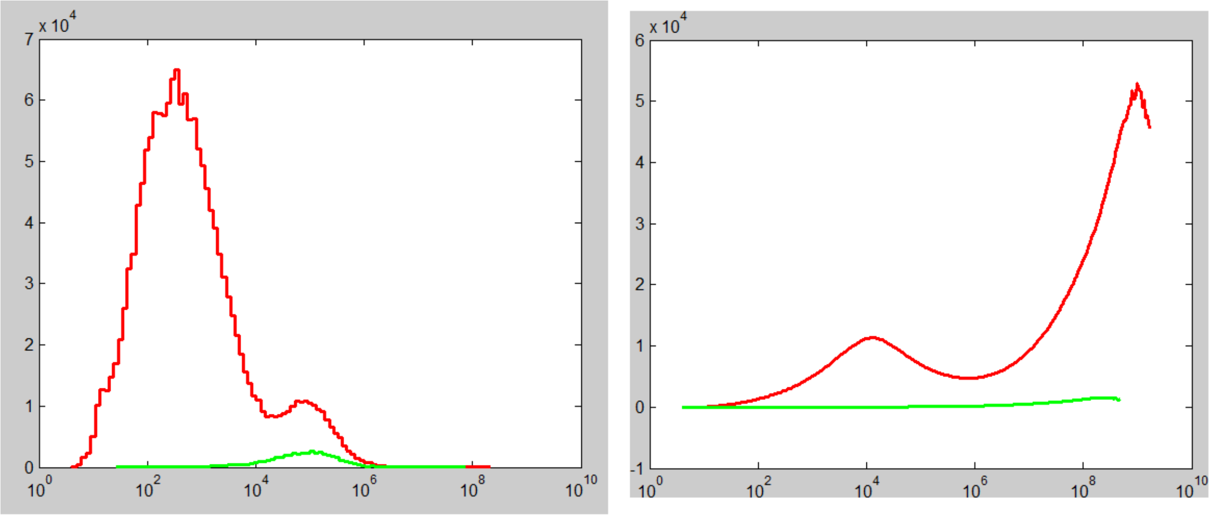


Figure 14: inter-arrival times (left) and fishing results (right) from just before (red) and after (green) the shattering removal (adaptive = aggressive in this case) step

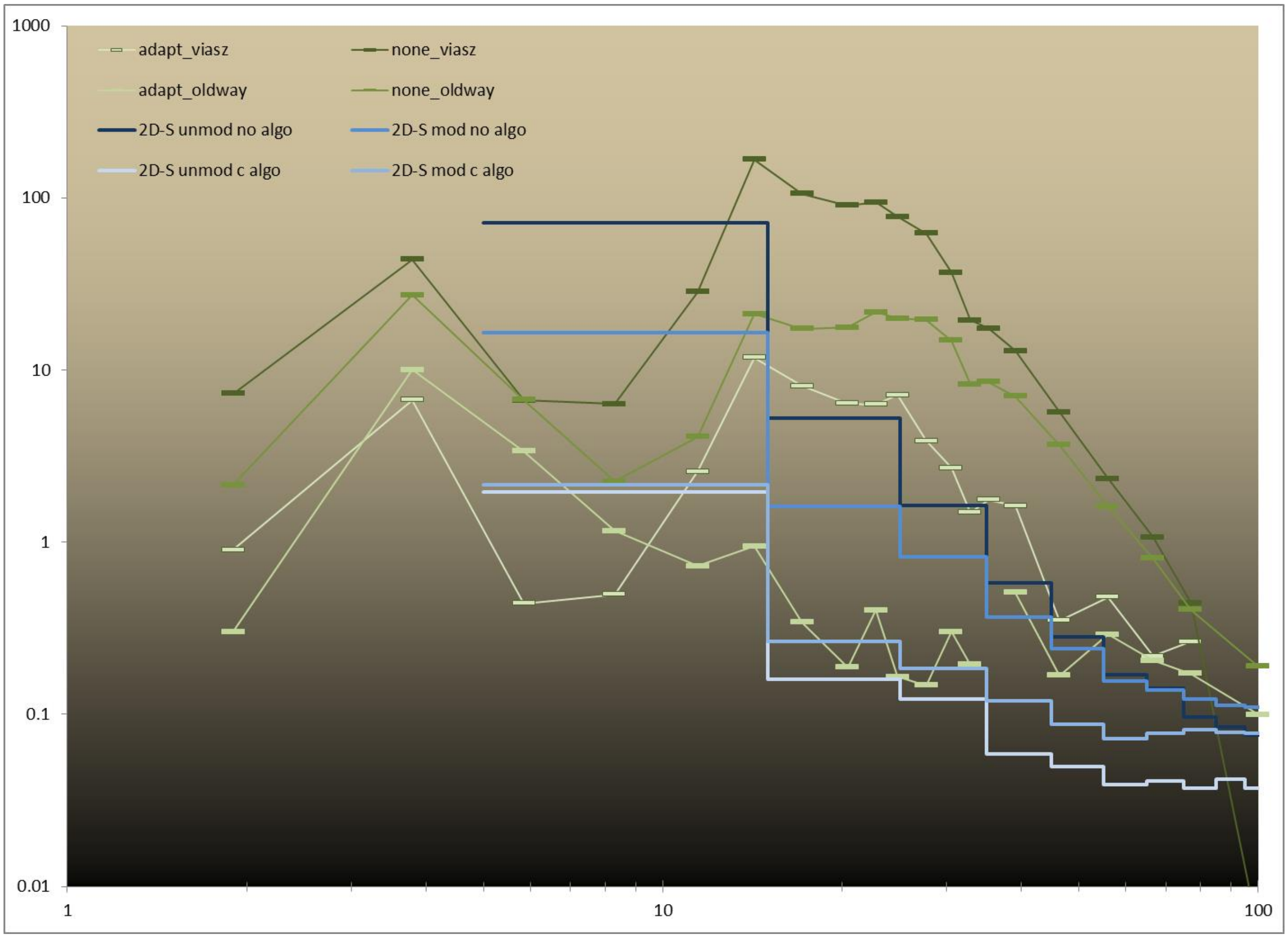


Figure 15: 2D-S and FFSSP PSDs via various processing options. ‘Adapt’ and ‘none’ refer to adaptive and no shattering removal respectively. ‘viasz’ and ‘oldway’ refer to the transit time qualification size dependent or standard, respectively. For the two 2D-S probes, ‘mod’ and ‘unmod’ refer to whether they had tips modified to reduce shattering effects or not, respectively, while ‘c algo’ and ‘no algo’ refer to whether the particle proximity algorithm for shattering removal was used or not, respectfully.

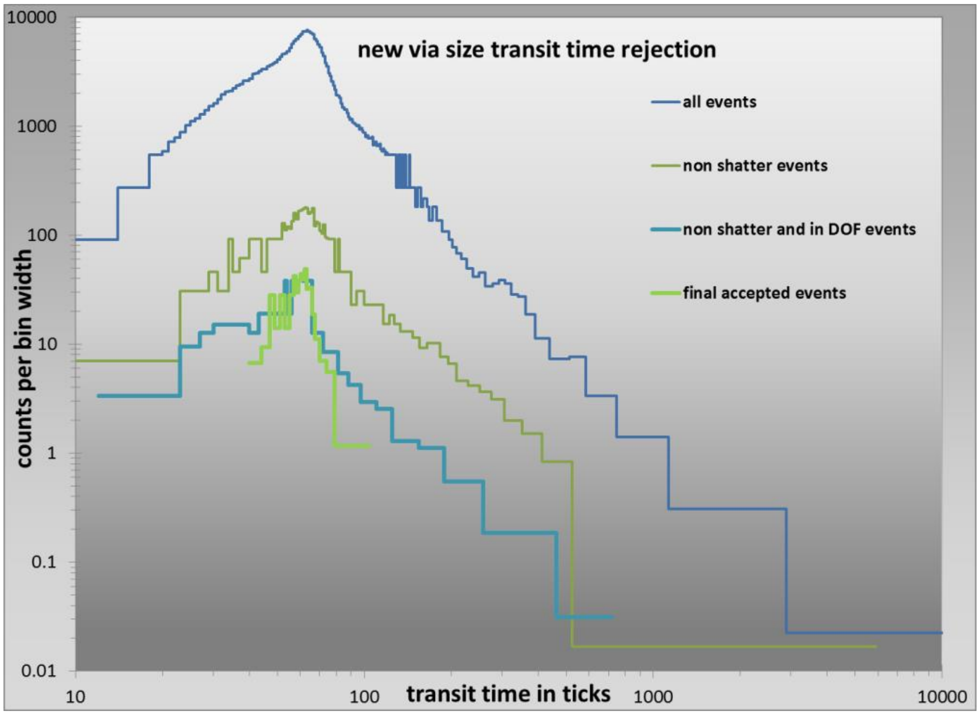


Figure 16: transit time distributions at various stages of the processing, indicating that the new via-size transit time rejection worked well here

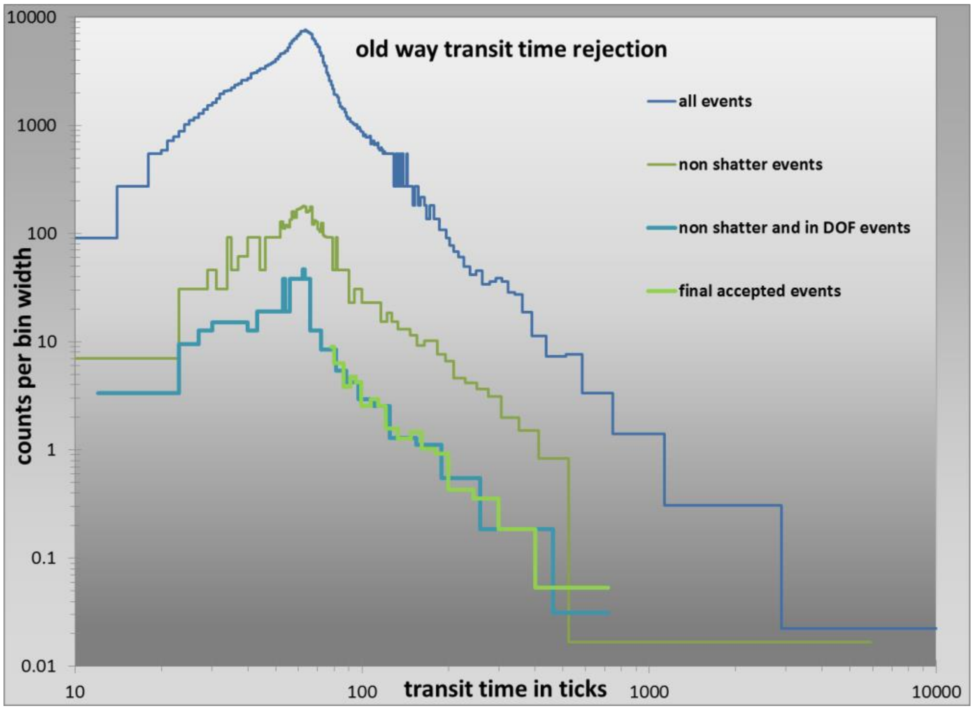


Figure 17: transit time distributions at various steps of the processing, indicating that the standard (mean transit time sets the cutoff) transit time rejection worked poorly here

2.3.4 Other possible effects of the size dependent transit time qualification:

We have one case (17 April 2010 SPARTICUS) of clear air for which apparent noise events exceeded the noise threshold and were thus counted as real events using the standard transit time qualification, yielding an incorrect concentration of about 1000 L^{-1}. However, because the noise events had transit times inconsistent with their apparent sizes, they were all removed by the size dependent transit time qualification, yielding 0 L^{-1} concentration.

We have another single case where it appears that shatter events were more efficiently removed by the size dependent transit time qualification, again due to transit times inconsistent with their sizes, than by the proximity algorithm specifically designed for shattering removal. This may or may not be an anomalous case. More study is required.

Appendix A - Setting the size bins - Mie bump considerations

Note that all the PSDs shown so far (Figs. 6, 9, and 12) are at least bi-modal. In the following we will show that this is likely a spurious effect, due to the way the bin edges have been chosen to accommodate the Mie bumps.

During processing, besides the PSD using the standard fixed bins, an additional PSD with fixed statistics but variable bins (unihist) is produced. To make the unihist PSD, every particle must be assigned a size. This is accomplished in the subroutine 'sizepbp_rv#.m' by interpolating the signal voltages across the fixed bins that they fall into. I.E. if a given signal voltage lies half way between its two voltage bin edges than it is assigned the size that lies half way between the corresponding two size bin edges. This PSD helps reveals that there is likely a problem with our technique, of determining bins, that takes into account the Mie bumps. The calibration routine creates three calibration tables, one using the Mie bumps and two with evenly spaced bins, one out to the same size as the Mie bump calibration extends and another out to the largest size possible (see Fig. 2).

Assuming that the size distribution is smoothly varying, then because of the Mie bumps, the signal distribution will be bumpy. Consider e.g. the Mie curve and the bin edge points for the FFSSP shown in Fig. A1. The bin from ~7 to ~10 μm has a wide size width and small signal width relative to surrounding bins, whereas the bin from ~13 to ~15 has just the opposite, a narrow size bin width and wide signal bin width. Thus for a smoothly varying, or uniform, size distribution, we can expect that the signal distribution will have relatively more, and less respectively, in these two bins than neighboring bins.

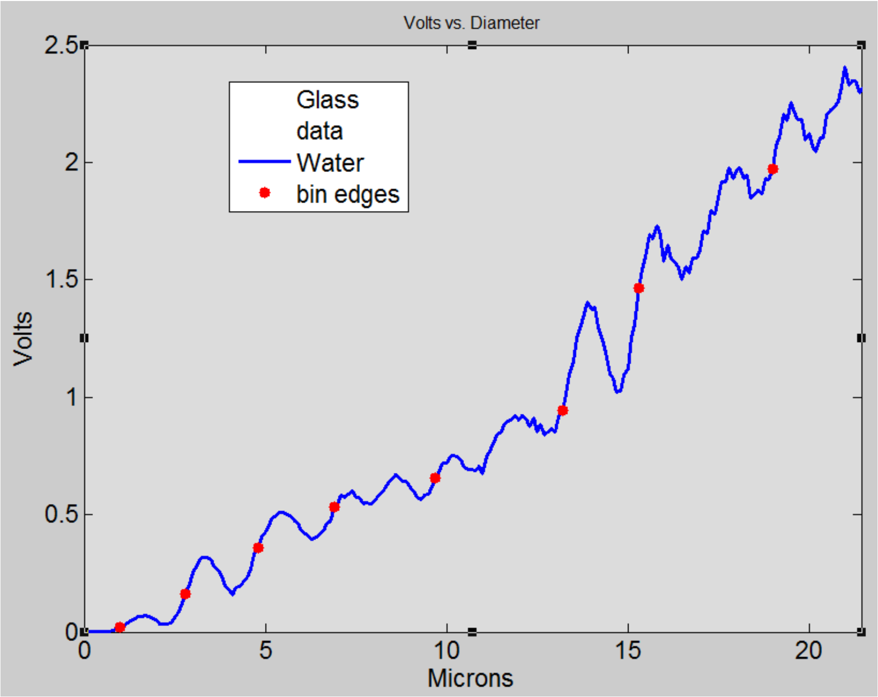


Figure A1: Mie curve, signal voltage versus size, and bin edges for the FFSSP.

What we observe in the data however, can be just the opposite. The signal distribution can be smooth (Fig. A2) and the size distribution, using the Mie bumps calibration, has bumps or jumps (Fig. A3) specifically at the above mentioned bin's edges. The same effect occurs to the usual PSD (Fig. A4) but is not as obvious. I.e. we might believe the bimodality seen in Figs. 6, 9, 12, and A4 as a cloud physics effect rather than seeing it for the spurious effect that it seems to be. Figs. 2 - 4 were made using all the data events instead of just the qualified events because it is easier to observe the phenomenon this way. Most ICE-T data periods have too few droplets, after qualification, in these small bins to show the effect clearly. Figs. 5 - 7 show another data period (ICE-T RF7, seconds 2983 to 2990) using qualified events only, chosen because it does exhibit the same effects, though more subtly. Note that the spikes at ~1.8 V in the signal distributions and ~24 μm in the PSDs are a separate spurious effect that still needs resolving.

These results suggest it may be better to ignore the Mie bumps. This is likely due to a number of effects that smear out the Mie bumps. In particular, the Mie curve varies with position relative to the focal plane. That is, the Mie curves shown in Figs. 2 and A1 were calculated for a droplet at the focal plane of the optical system. Away from the focal plane but still within the depth of field, the curves vary enough to effectively blur the bumps away.

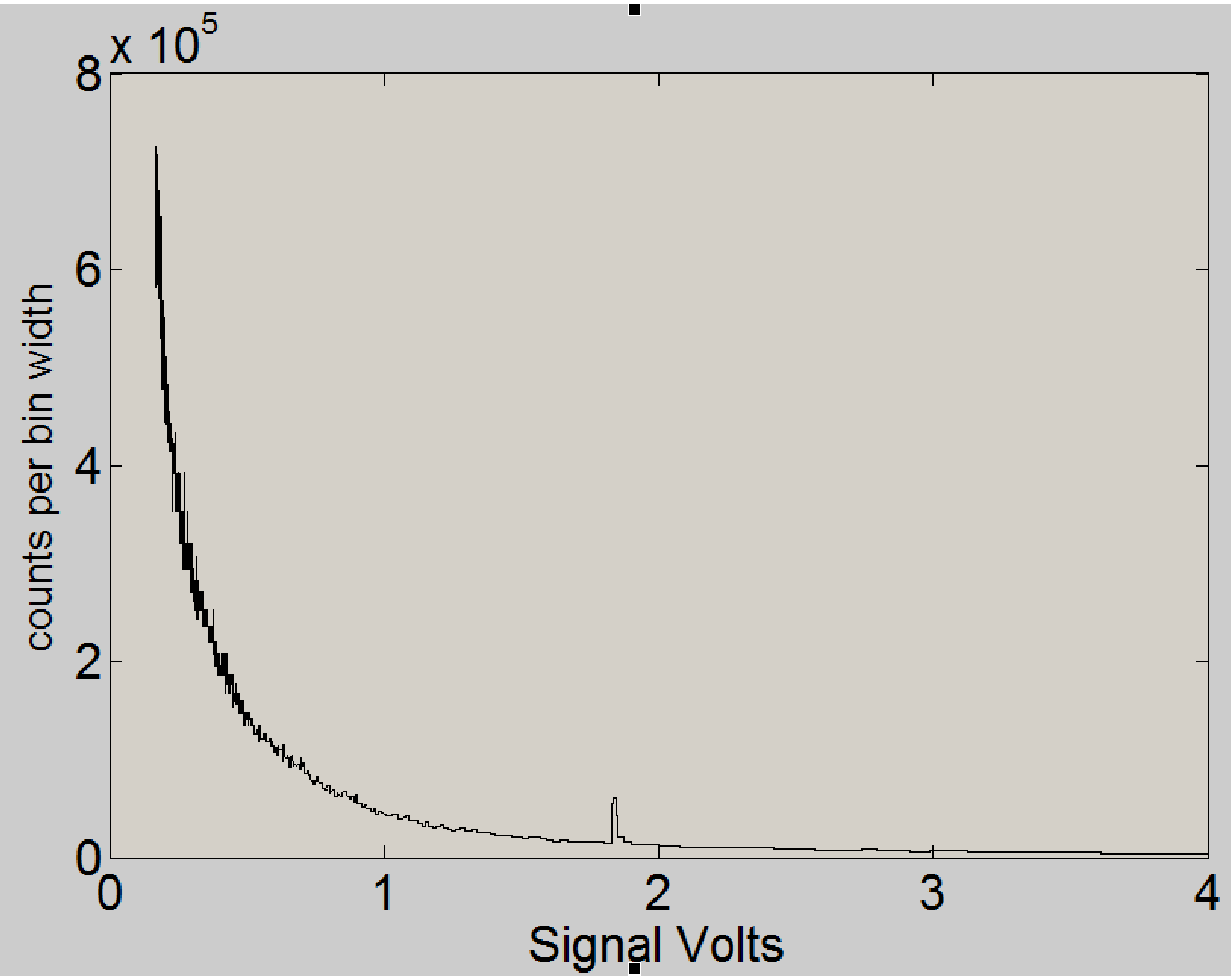


Figure A2: Example of a typical signal peak distribution. The spike near 1.8 V is a quirk that has already been documented but not solved.

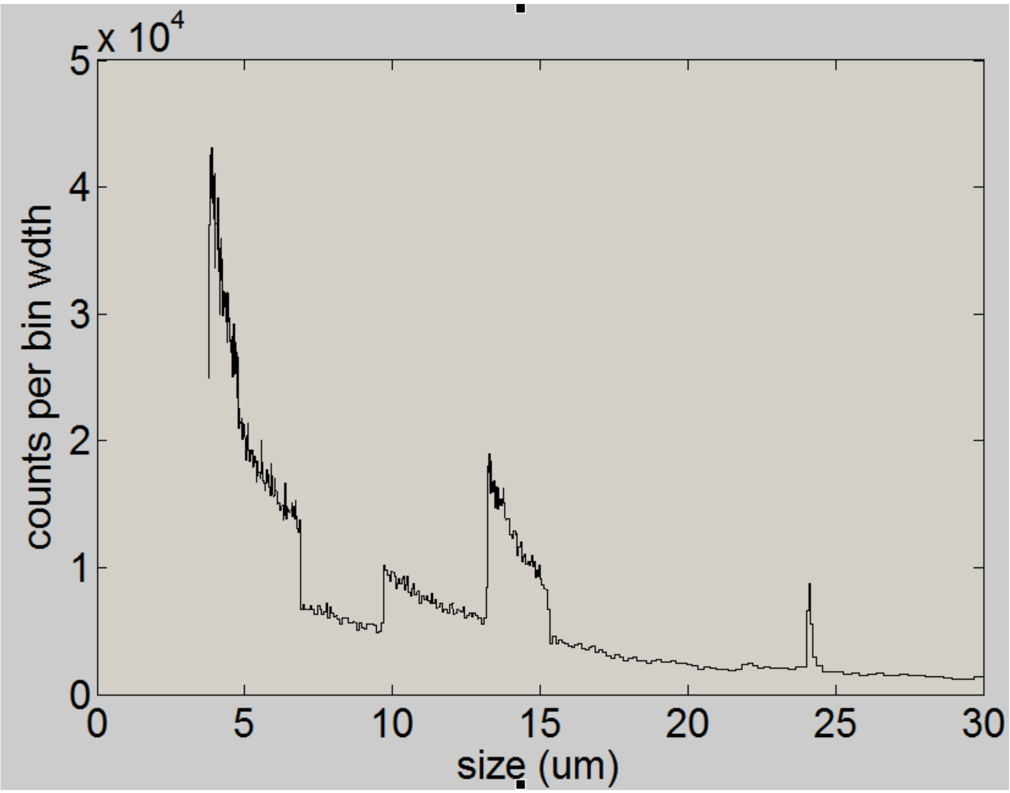


Figure A3: Unihist sizes distribution, using the binning determined to account for the Mie bumps, for the signals shown in Fig. A2.

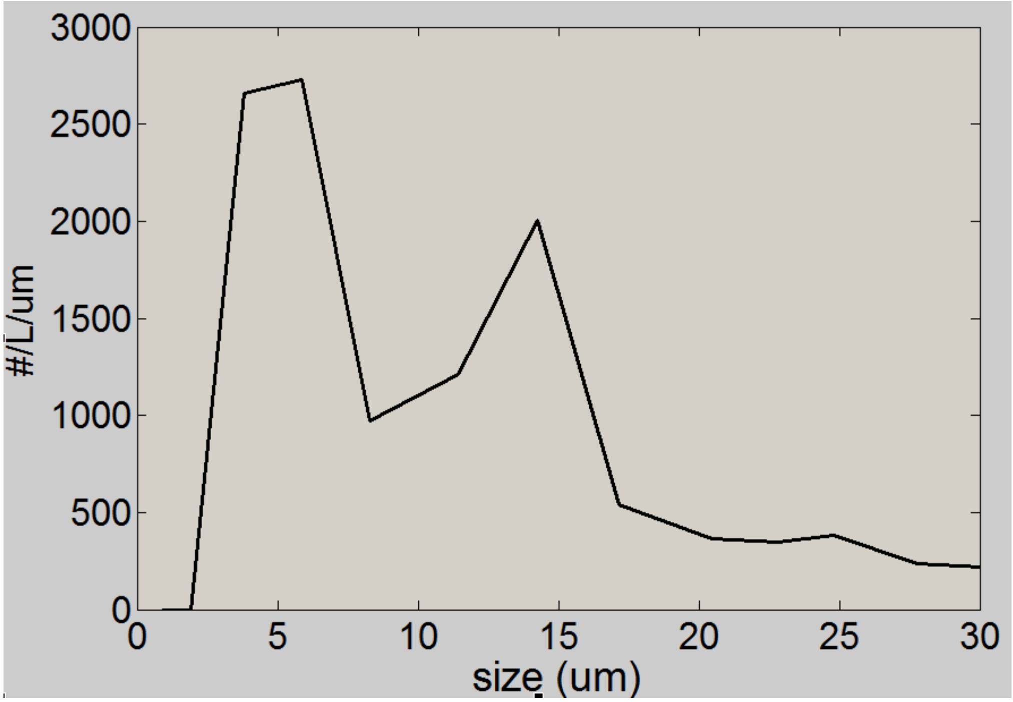


Figure A4: standard PSD, using the binning determined to account for the Mie bumps, for the data shown also in Figs. A2 and A3.

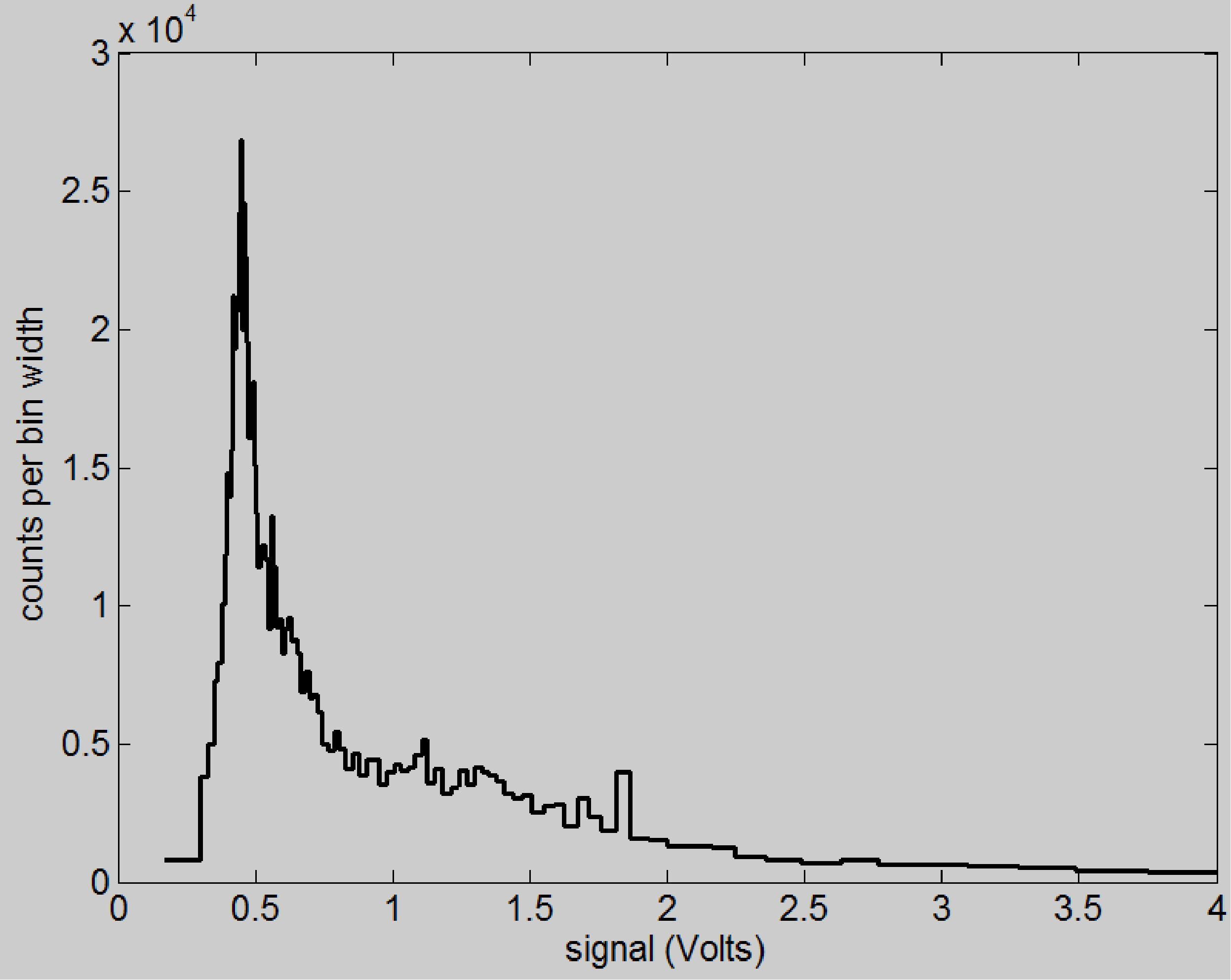


Figure A5: Distribution of peak signal voltages for fully qualified events only.

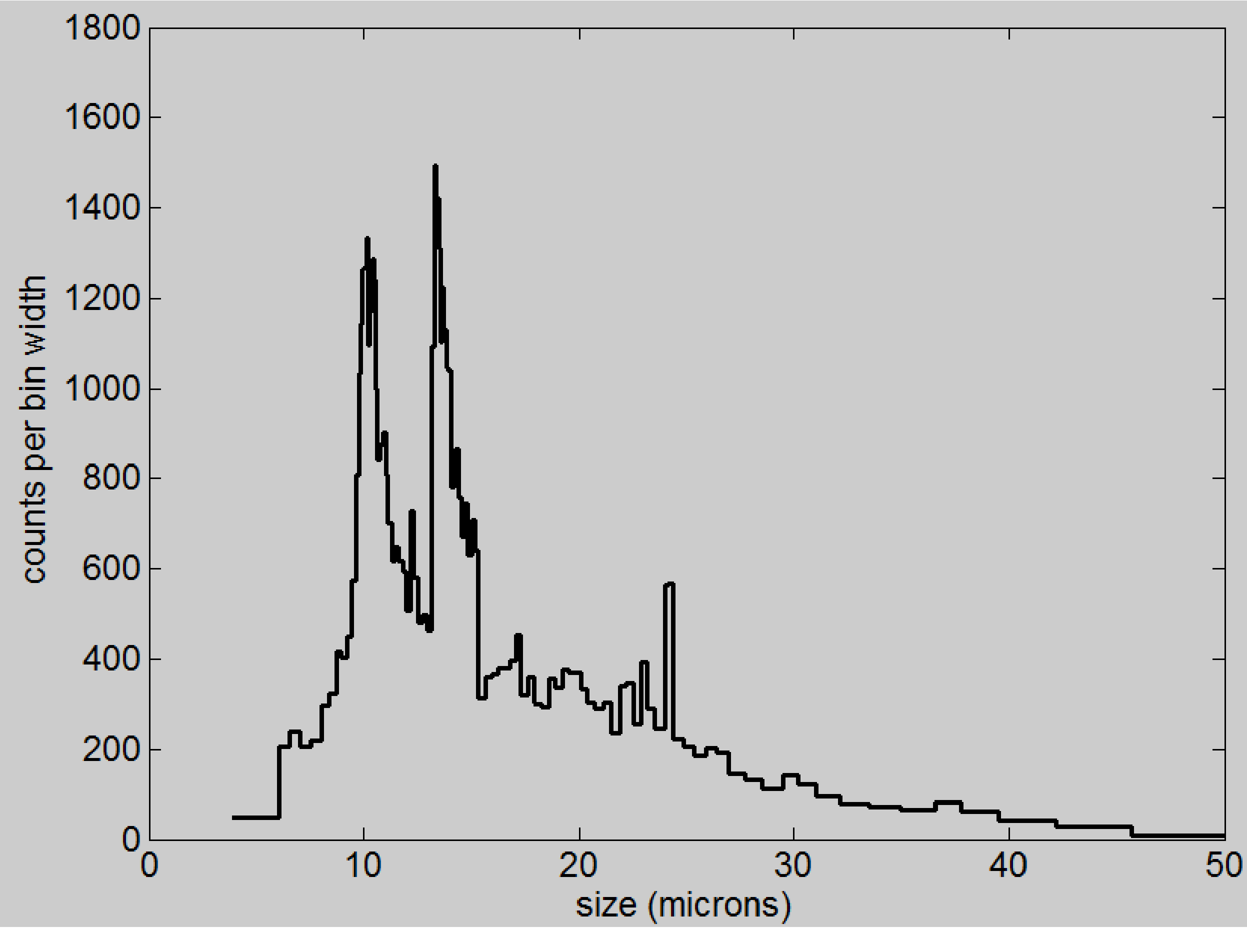


Figure A6: Size distribution using the binning determined to account for the Mie bumps.

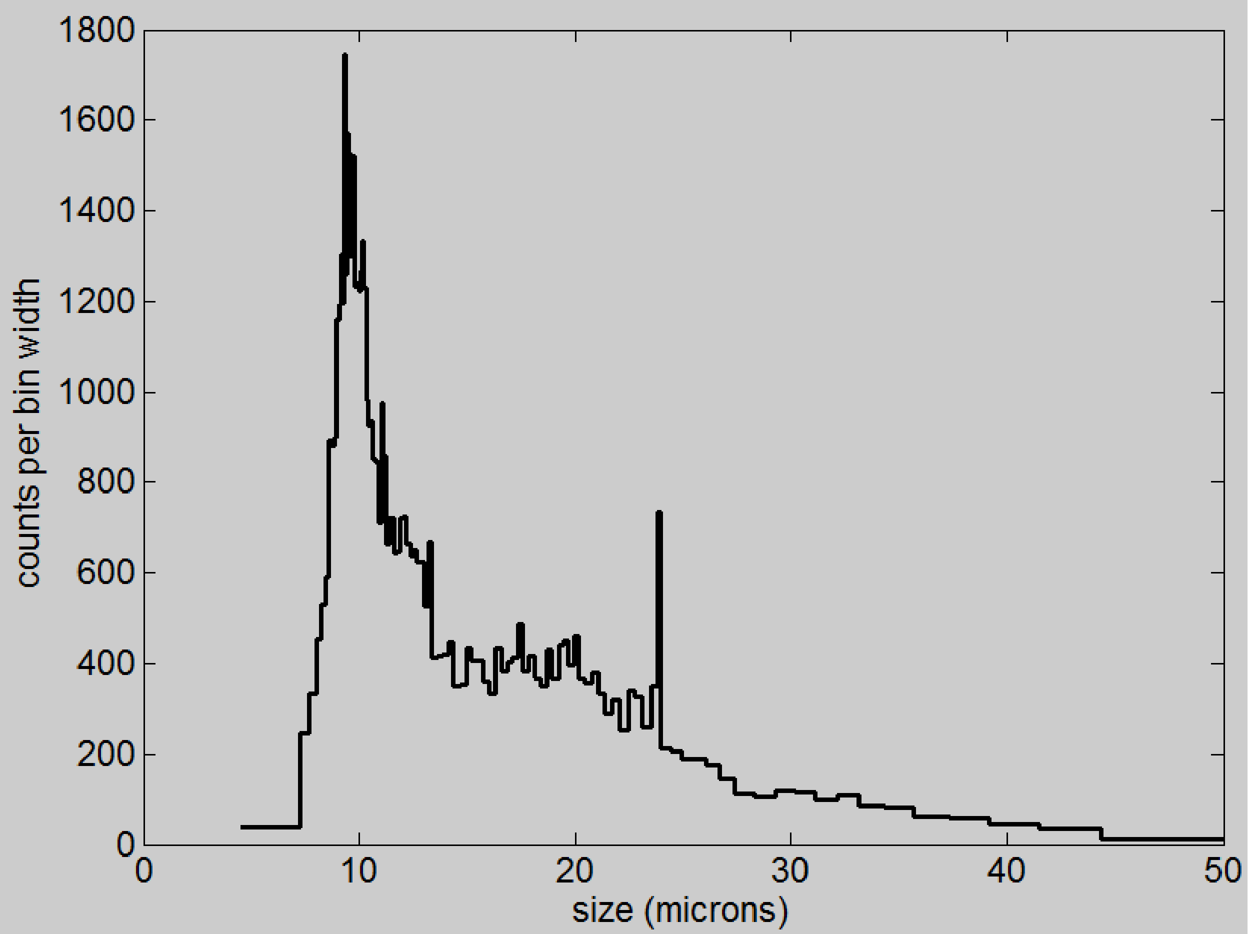


Figure A7: Size distribution using the binning determined ignoring the Mie bumps.

Addendum: The above figures were made when the particle by particle sizes were determined as linear interpolation across each bin. This causes jumps similar to those shown above and has been remedied by using the mean Mie parabola instead. This removes jumping for both non Mie bump calibrations and for Mie bump calibrations. Be aware that even for Mie bump calibrations this PSD does not use the Mie bumps. The fixed bin PSD does use the Mie bumps when that is specified. The jumps shown above are in excess to those just discussed. So the points made remain valid.

Note that when transit times max then the w8 time and tt for that event are in error as are live and dead time calculations based on those unless those events are thrown out in the way that puts the w8 time as dead time. Need to check on that. Meanwhile Chris has more bits that can be added to the tt counter.

Appendix B - half peak versus full transit time qualification

The following is excerpted from a report on the FFSSP performance on the 9-17-2012 LEAR2012 (second) test flight.

Finally for those interested (not everyone will want to bother with the following), I noticed something I wasn't looking for but that is worth documenting for future reference. One step of processing/qualification that I implemented from the beginning but was never able to test/investigate/calibrate is the half peak transit time versus full transit time comparison. The excerpt copied below from the manual explains the algorithm and its motivation.

The first transit time qualification is the same for both the FFSSP and F_CDP. This step compares the half peak transit times (half peak transit time is the time from the event start to the peak signal value) with the full transit times. To qualify the half peak transit time must be between 0.33 and 0.60 times the full transit time. For a smooth symmetric pulse the time to the peak, i.e. the half peak transit time, should be just a bit under half the full transit time, due to the event ending threshold being below the event initiating threshold. Hence the step helps eliminate erratic non-symmetric events.

Figure 4 shows the half peak transit times versus full transit times for the same period shown in plots above. I have seen this bifurcation many times but this time it sparked a synapse to fire, or something, and an idea came to mind. Figure 5 shows the same figure with regions highlighted. The lightly highlighted region is about where we would expect good data to lie. The darkened highlights might be explained as due to coincident events, the lower region where the larger particle came first and the upper darkened region where the larger particle came last. This brought to mind the opportunity to see if the essentially arbitrary cutoff values I chose when I first implemented the algorithm are at all close to where we might want them. Figure 6 suggests yes but that they might be tightened very slightly. Figure 7 shows the effect on the half peak transit time versus size plot suggesting that it works quite well.

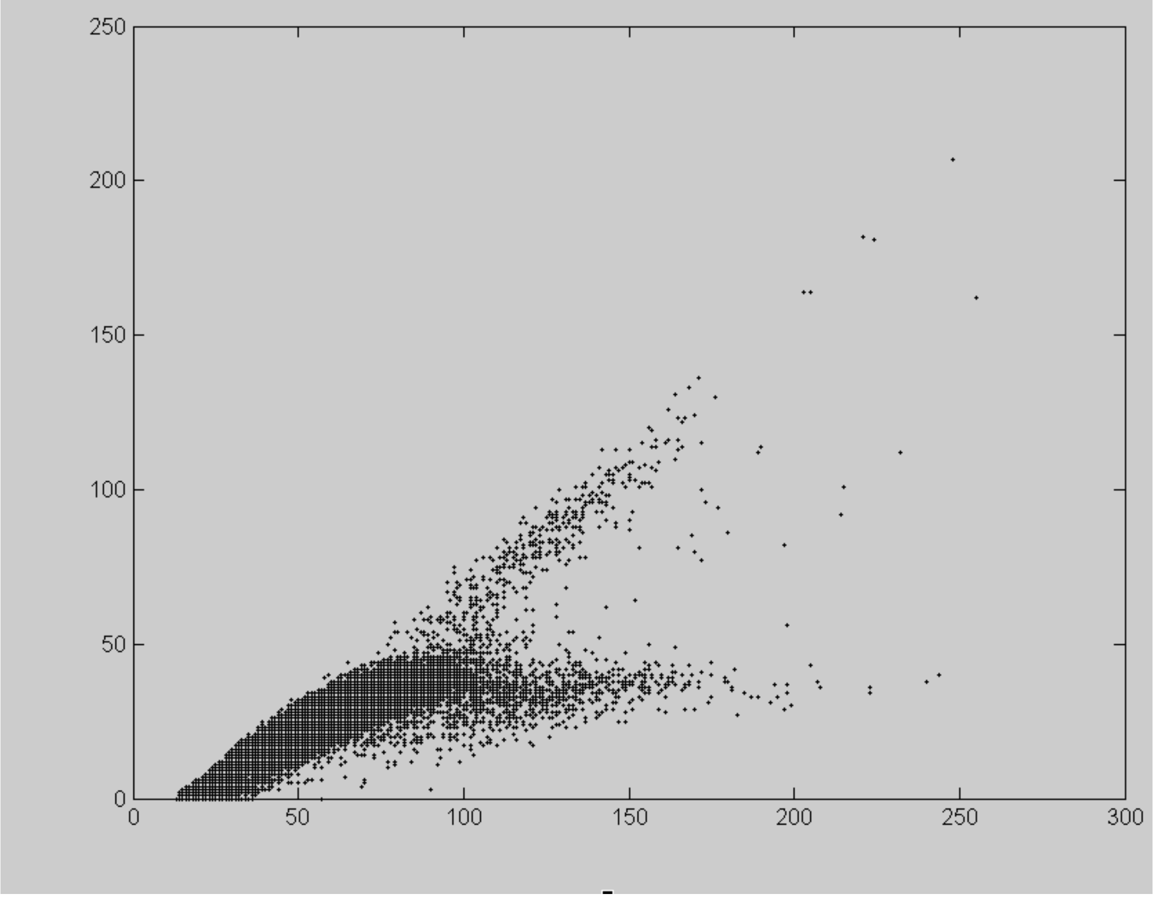


Figure 4: Half peak transit times versus full transit times for the period 14:45:23 - 14:45:30 (1603 - 1610 in seconds of file) FFSSP time.

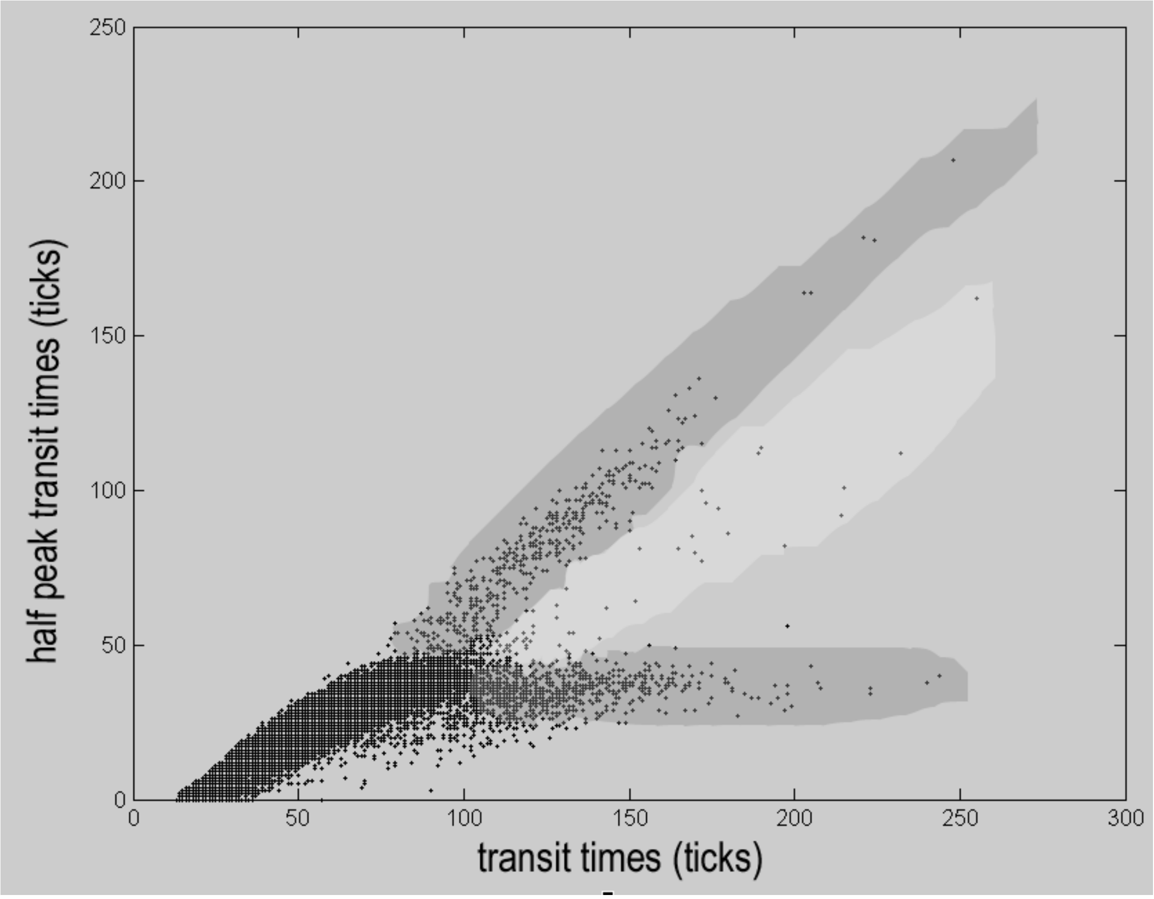


Figure 5: Half peak transit times versus full transit times for the period 14:45:23 - 14:45:30 (1603 - 1610 in seconds of file) FFSSP time. With regions highlighted for discussion in the text.

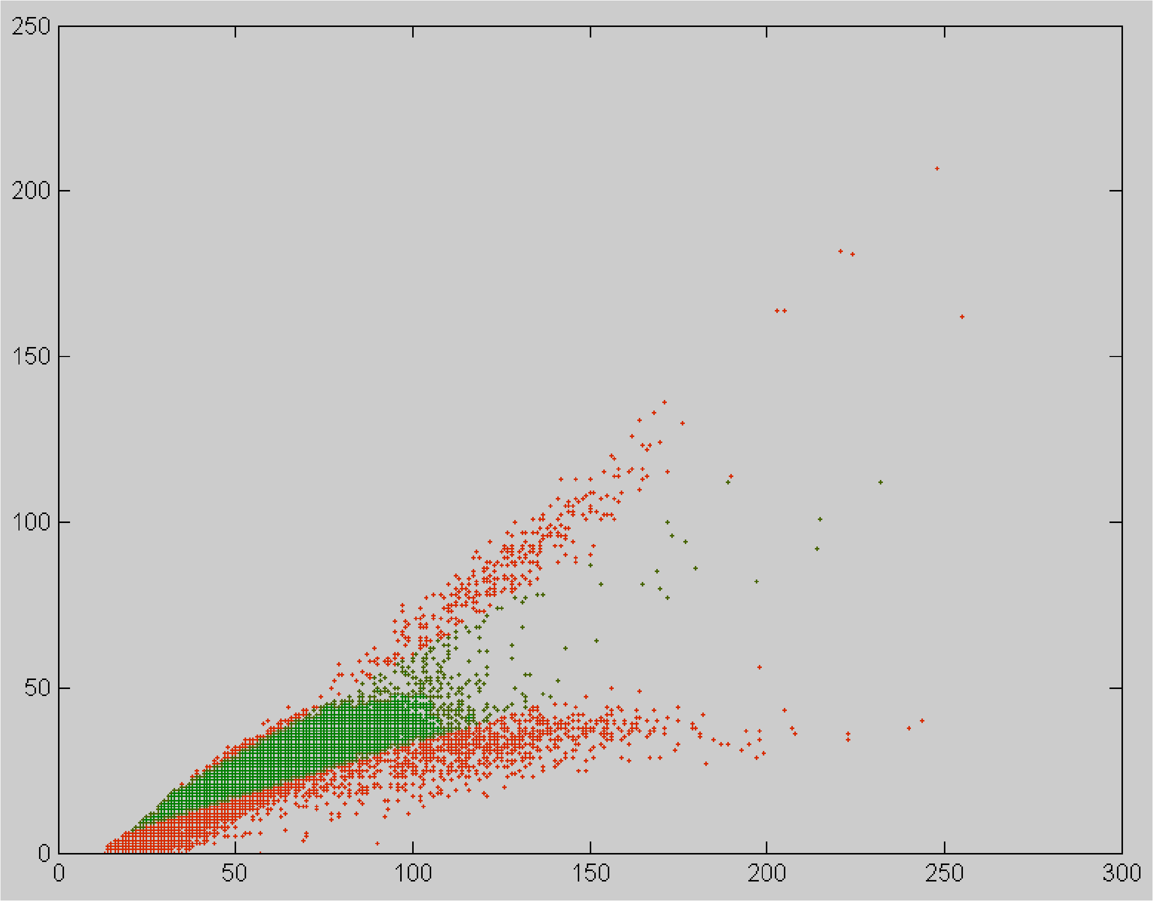


Figure 6: Half peak transit times versus full transit times for the period 14:45:23 - 14:45:30 (1603 - 1610 in seconds of file) FFSSP time. Red indicates before the half peak versus full transit time qualification while green above red indicates those points that remain after the half peak versus full transit time qualification.

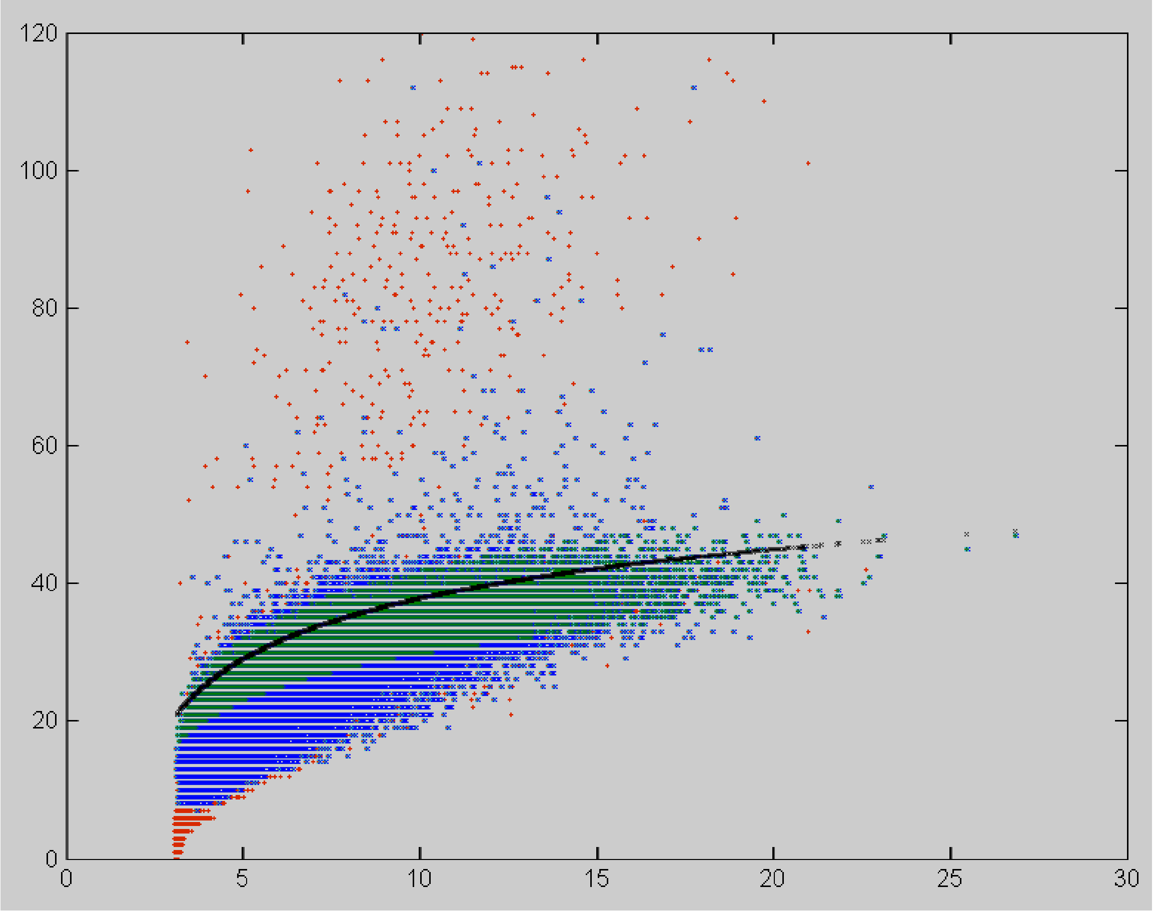


Figure 7: Transit time (y-axis in ticks) versus size (x-axis in um). Red indicates points after DOF qualification only, blue indicates after DOF and half peak versus full transit time qualification, green is after size dependent transit time qualification as well as the others.

Incidentally, the points highlighted (both light and dark) in Fig. 5 are removed by the size dependent transit time qualification anyway so the half peak to full transit time step may be largely irrelevant. However close inspection does reveal a few points that would have been accepted by the size dependent transit time qualification but they were already rejected by the half peak to full transit time step, so perhaps it is a worthwhile step.

Appendix C size calibration processing tutorial

After obtaining the bead calibration data (preferably a separate set of files for each run), you must do the usual first processing step of running:

'main_rewrite_dars_scatbin_into_pbp_format' on each set.

Then use quick look to find the time period of the actual run which should be a pulse of high count rate (see Fig. 0). Then use:

'Main_GenFSSP_PBP_rawData_frm_T1_to_T2'

to create a particle by particle file for each of those periods. Next you must find a peak in the signal voltage distribution associated with the beads for each case. I'm writing this and saving figures as I process FFSSP bead data taken by Seth on 9/4/2012. The data exemplifies a number of the problems that can occur and allows demonstration of some solutions for dealing with them. I generally start with noise reduction, shattering reduction, and fishit 'off' in the setup so there are less plots to deal with. If warranted I'll go back and rerun with noise reduction 'on' and an appropriate 'noise string' set.

Seth took five sets of data for five bead sizes. I segregated the files according to file name time and date marks:

120904161915:	10.1 ±0.5
120904162226:	15.9±0.6
120904162508:	30.1±1.1
120904162645:	40.6±2.8
120904162807:	49.0±1.4

<u>120904161915:</u> this was an easy case. All the signal distributions (i.e. at each step of the processing) have a single peak at about the same location. Since I haven't adjusted the TAS to optimize the transit time qualification steps (the final steps), I use the last signal distribution (Fig. 1) before those steps. That is just after a step that is also a transit time based qualification but one that only compares the half peak transit times with the full transit times and does not require a TAS. Figures 1 and 1b show the polyfits (in this case 10th degree) that I used to find the center of the peak (0.48 V) using the data tip tool.

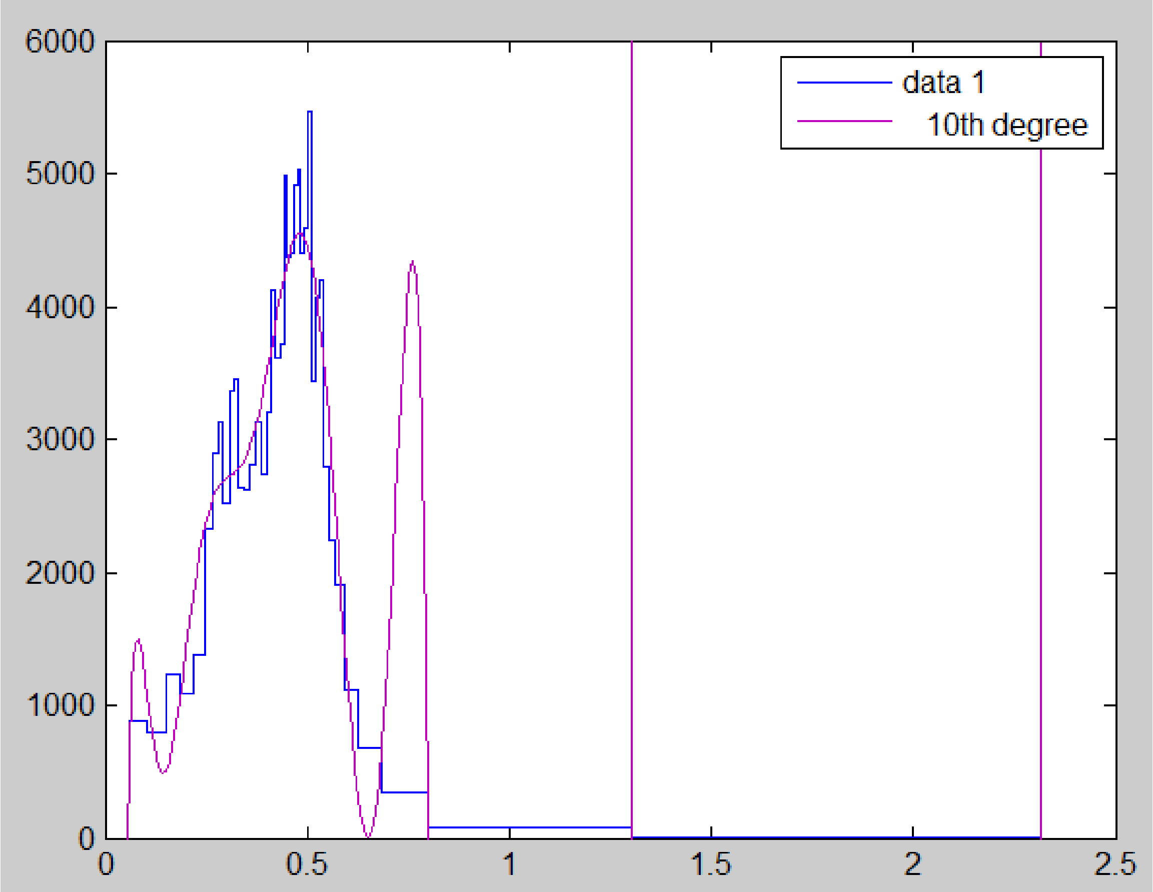


Figure 1: 120904161915 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps). A tenth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

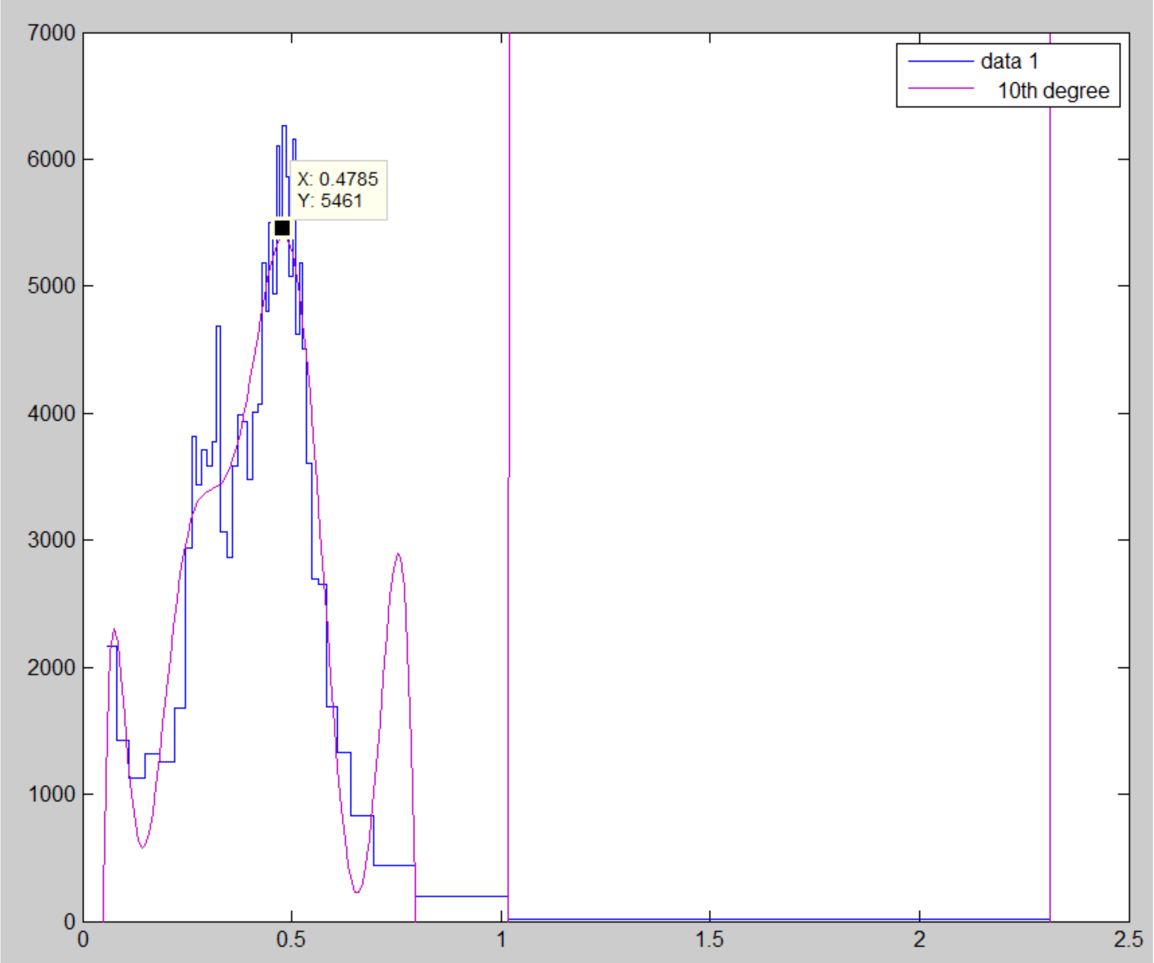


Figure 1b: 120904161915 signal distribution after the DOF qualification step. A tenth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

120904162226: is a case where I needed to adjust the TAS after inputting the data (typed TAS = 3000 on command line, in this case) in order to use the final signal distribution. This is because prior to the last processing steps (which are the transit time qualifications that require the TAS) there are only bimodal signal distributions (e.g. Fig. 3) and no way to tell which peak is the correct one. The TAS was originally set at 1000 cm/s in the setup. The points in the yellow swath of figure 4 indicate approximately the points that past the final transit time qualification steps with that TAS value (10 m/s). In that case the signal distribution became mono-modal at the smaller of the two peaks. But these points are not what I would normally consider good ones. They look like coincidence induced long transit time events. With bead data perhaps there are other causes of longer transit times. I re-input the data and changed the TAS to 30 m/s and reprocessed. This accepted the points shown as green in Fig. 4 that are where I would normally consider the good points and resulted in a mono-modal signal distribution (Fig. 5) at the larger mode. I'm guessing these are the points we want, though I don't know why the long transit time events would tend to have smaller signals. If there were caused by coincidence the opposite would be expected. If I have chosen poorly, that will come out when I run the calibration. Figure 5 also shows the polyfit (in this case 5th degree) that I used to find the center of the peak (0.68 V) using the data tip tool.

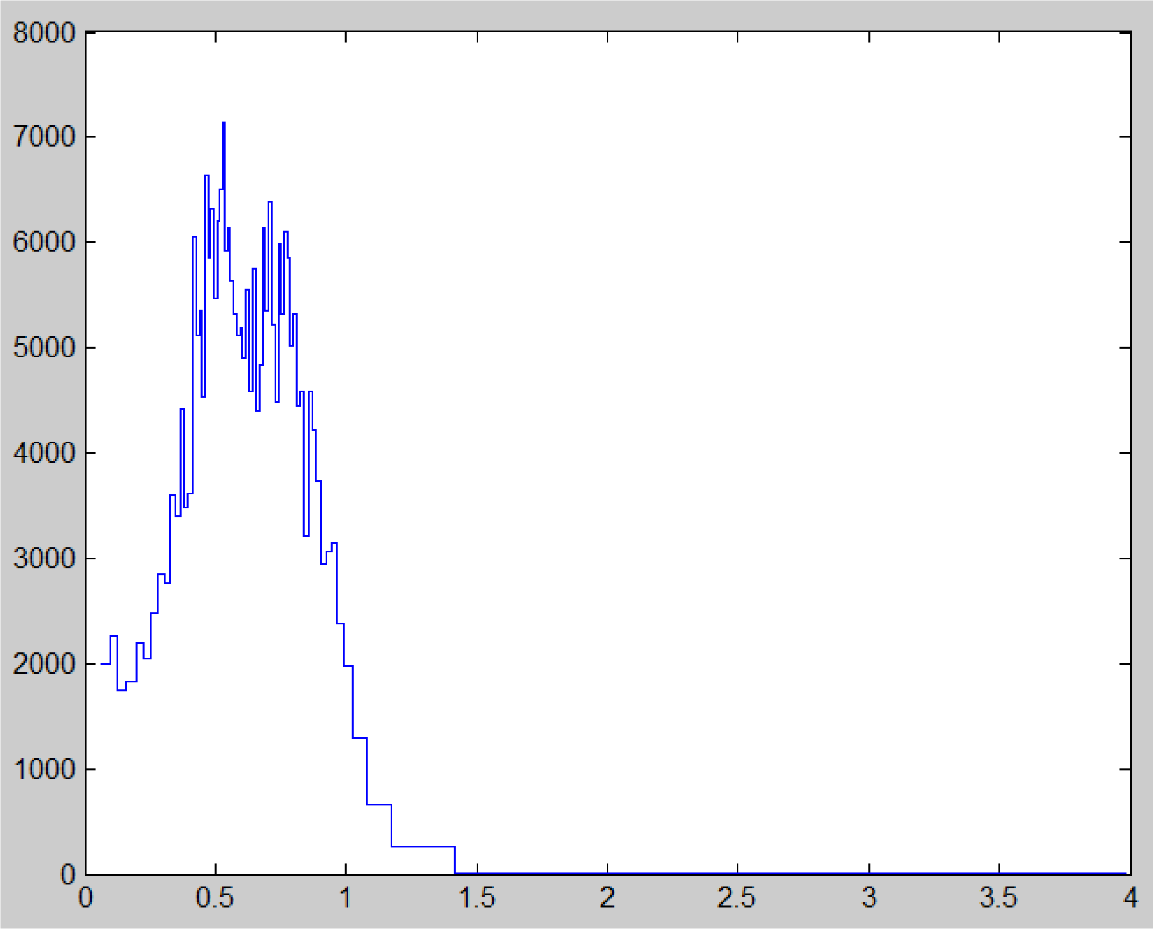


Figure 3: 120904162226 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps).

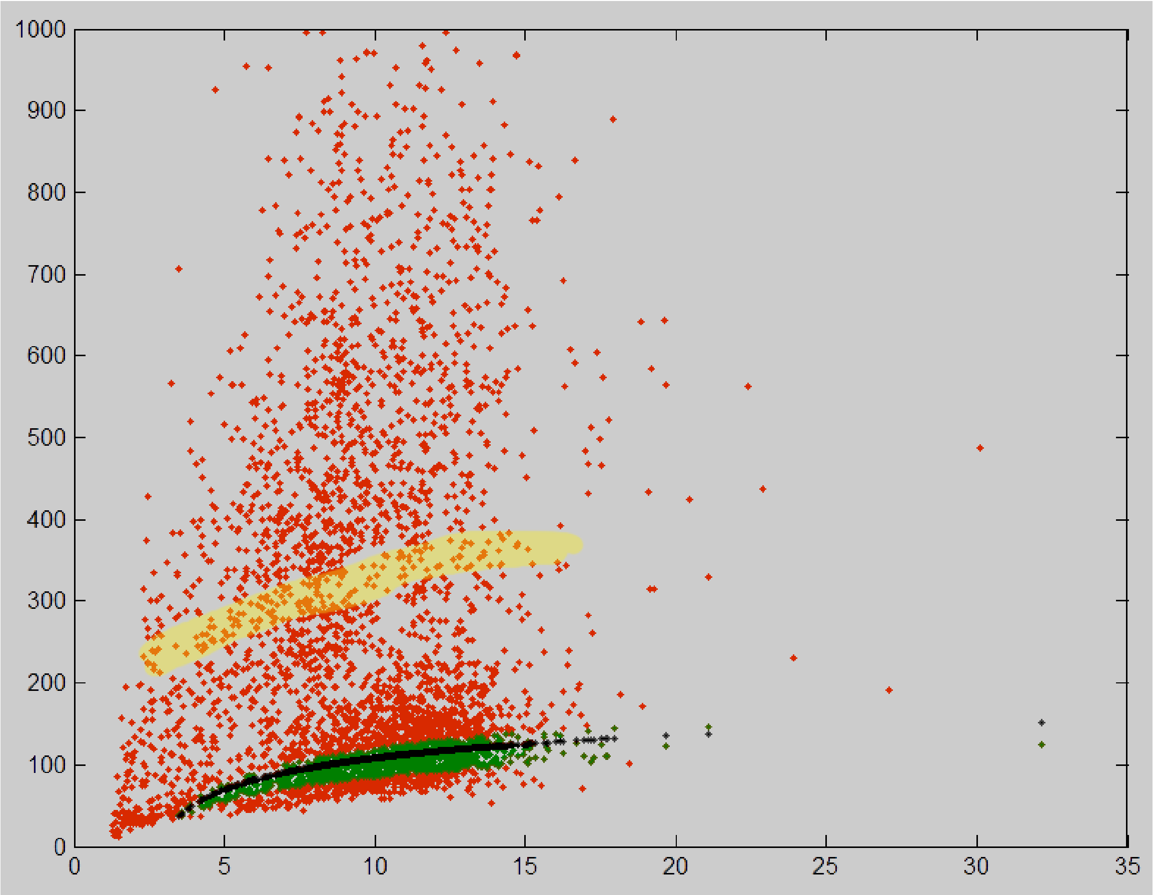


Figure 4: Half peak transit times versus size for 120904162226. Red are all data points just before the final transit time qualification processing steps. Green are the points remaining after the transit time qualification (TAS = 30 m/s). Black are the calculated transit time estimates. The yellow swath shows approximately the points that were transit time qualified (would have been green) when the TAS was set at 10 m/s.

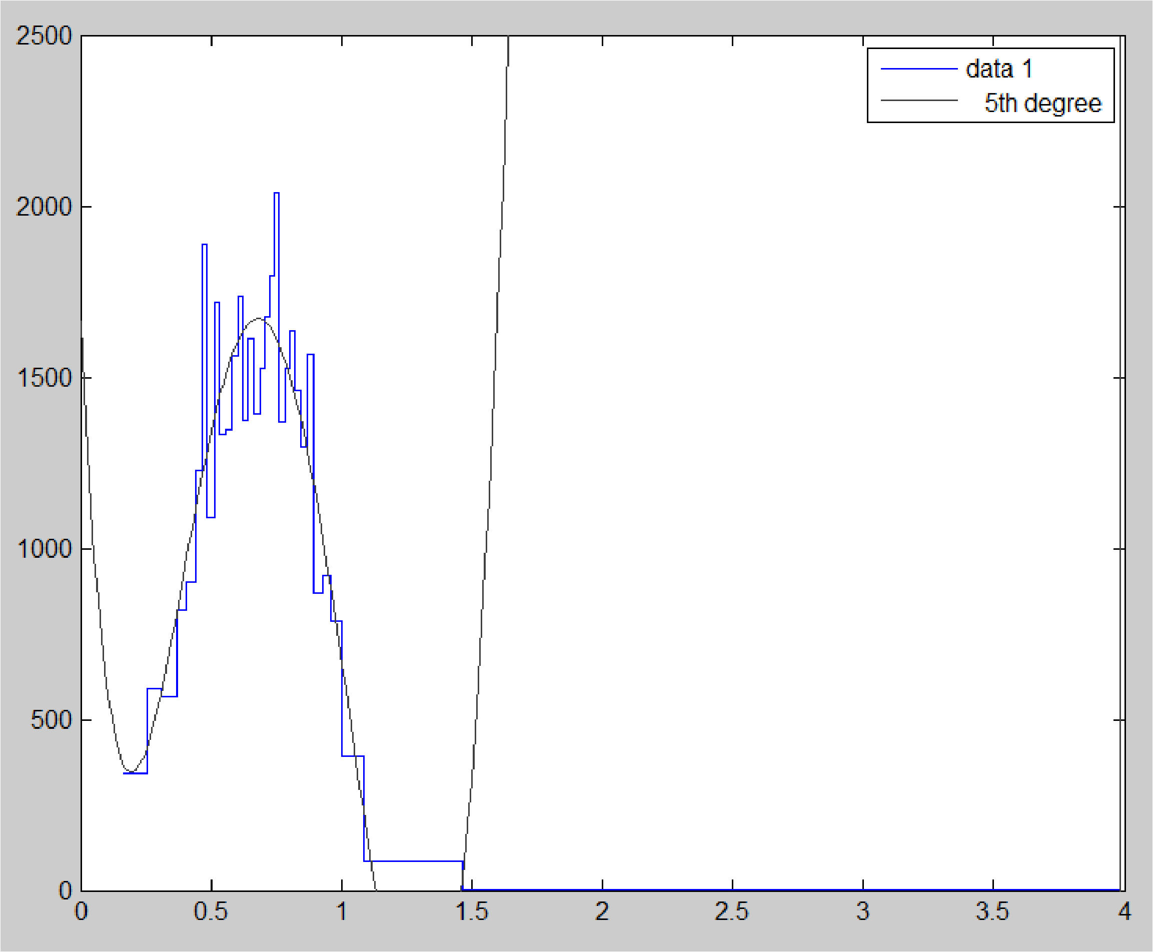


Figure 5: 120904162226 final stage signal distribution (with TAS = 30 m/s) with 5th degree poly fit.

<u>**120904162508:**</u> This case shows another difficulty that can occur. It is a good case as each step of the way there is a bimodal signal distribution but it is clear that the larger mode is due to the larger beads while the smaller mode may be due to remaining smaller beads and room aerosol. What makes it difficult is that the large mode peak is not very narrow and estimates of the value at the peak varies between the plots. Figures 6 - 8 show these distributions and possible peak value estimates after DOF qualification (Fig. 6), after half peak versus full transit time qualification (Fig. 7), and the final case after transit time qualifications (Fig. 8). I made the TAS 30 m/s again, which still works well for this data but does not leave much of the data. I suspect that there is some tendency for beads to go through at 30 m/s but that the majority travel slower. The estimates vary from 1.7 V to above 1.9 V. I'm inclined to go with the final estimate (1.7 V) but am concerned by the poor statistics for this case. So I try another trick. By setting a noise threshold, of 1.1 V in this case, and re-running, I can obtain better resolution distributions of the larger mode at the earlier stages. These are shown in Figs. 9 and 10. Both indicate a peak value of 1.9 V. So I change my inclination and go with the 1.9 V for this run, but I will make note of

my uncertainty for this point. When I run the calibration a single or even two outliers, due to such uncertainties as this, can be identified and either de-weighted or ignored entirely.

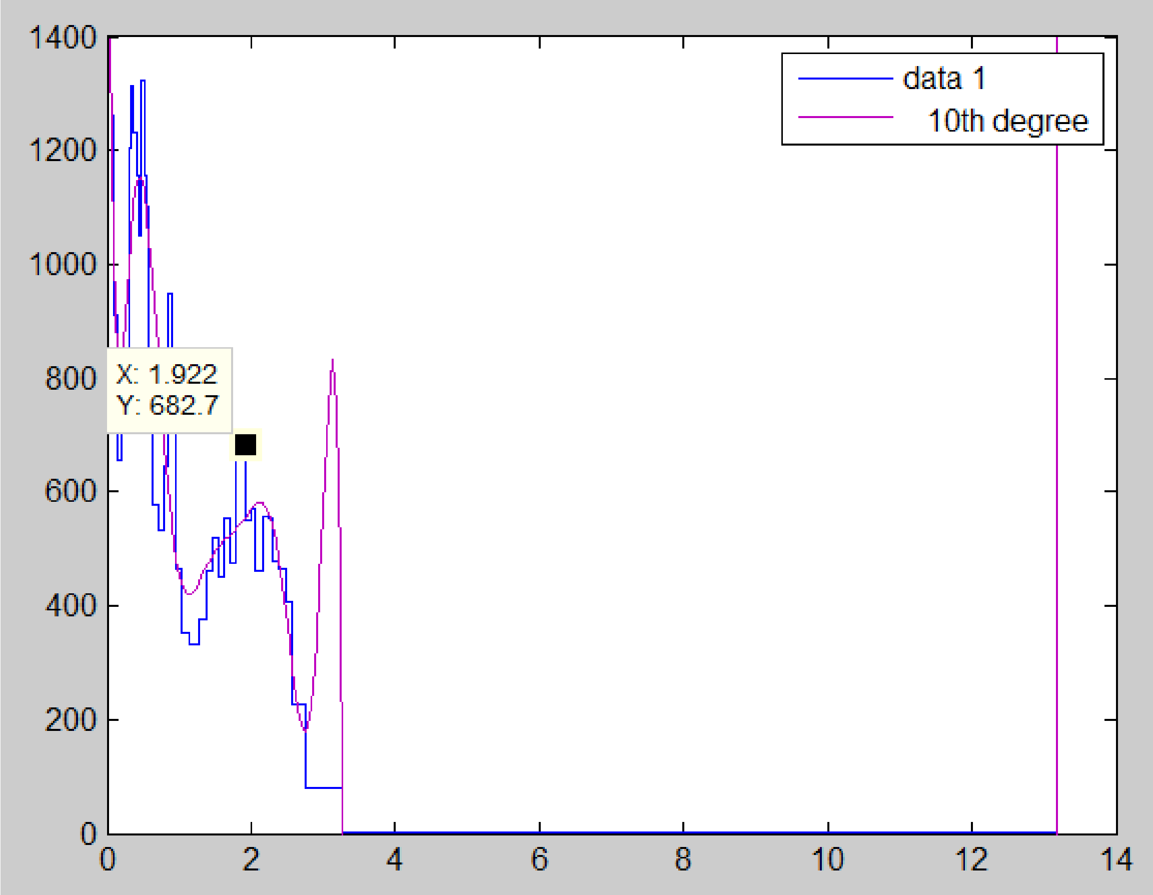


Figure 6: 120904162508 signal distribution after the DOF qualification step. A tenth degree poly fit is also shown that was used together with the data tip tool to identify a possible value for the peak (note that it is not the peak of the fit curve in this case).

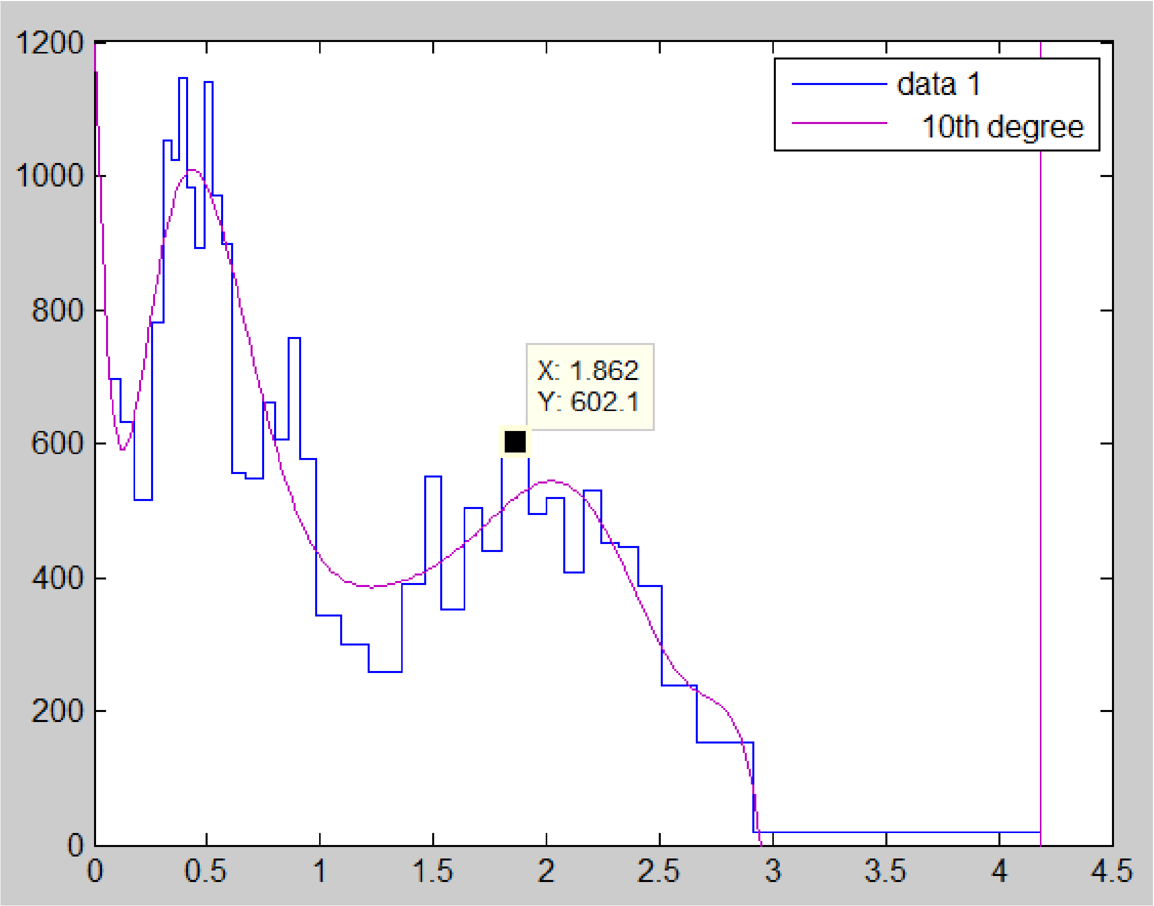


Figure 7: 120904162508 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps). A tenth degree poly fit is also shown that was used together with the data tip tool to identify a possible value for the peak (note that it is not the peak of the fit curve in this case).

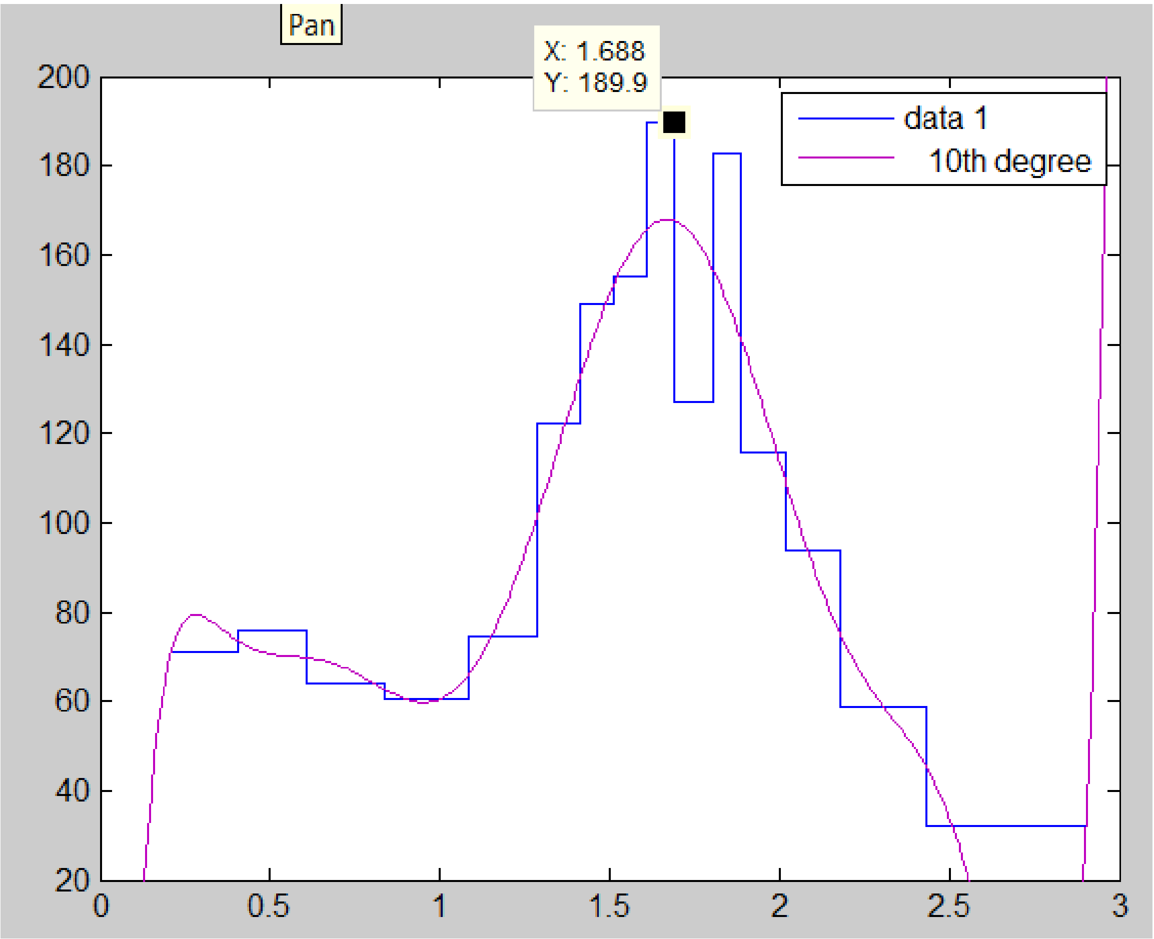


Figure 8: 120904162508 signal distribution after the final transit time qualification steps. A tenth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

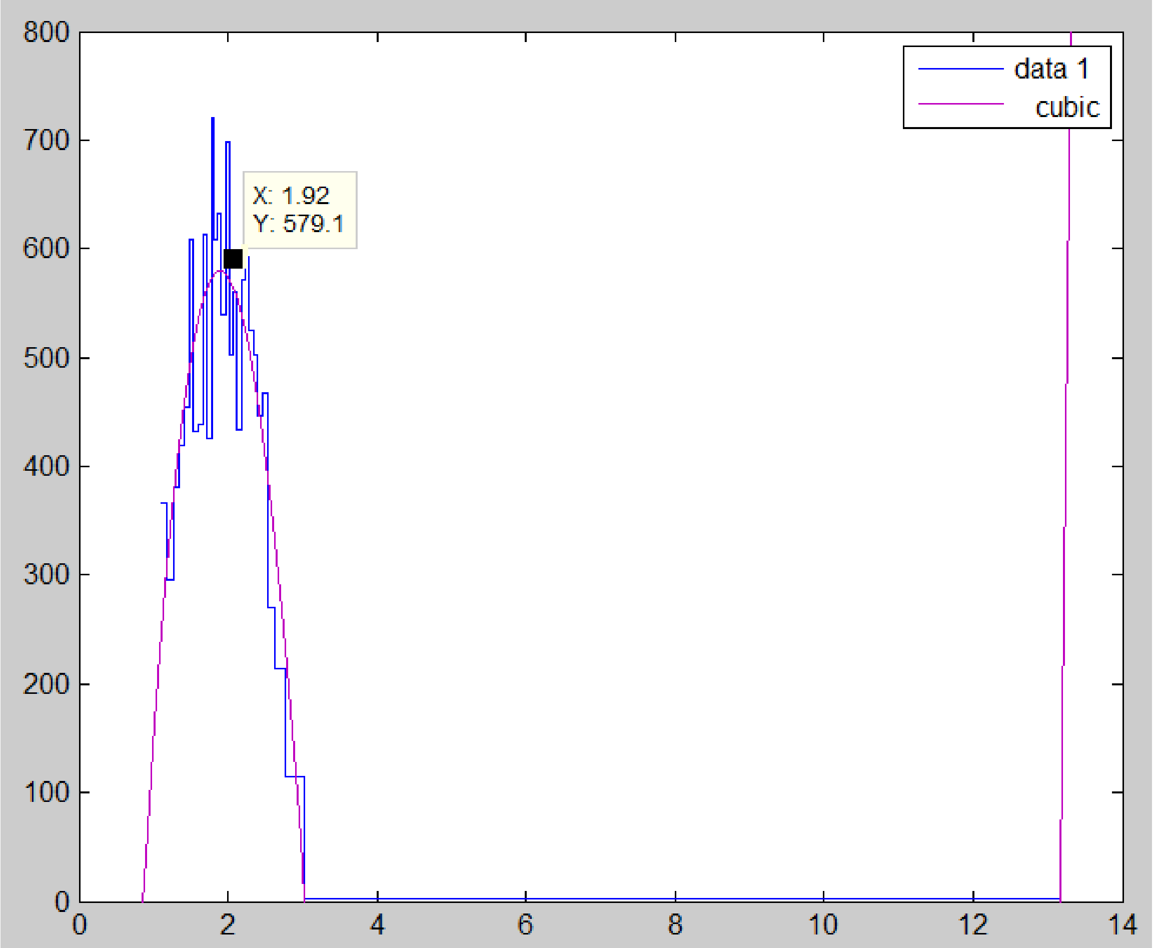


Figure 9: 120904162508 signal distribution after the DOF qualification step and after the small end was removed by using the noise filter. A cubic poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

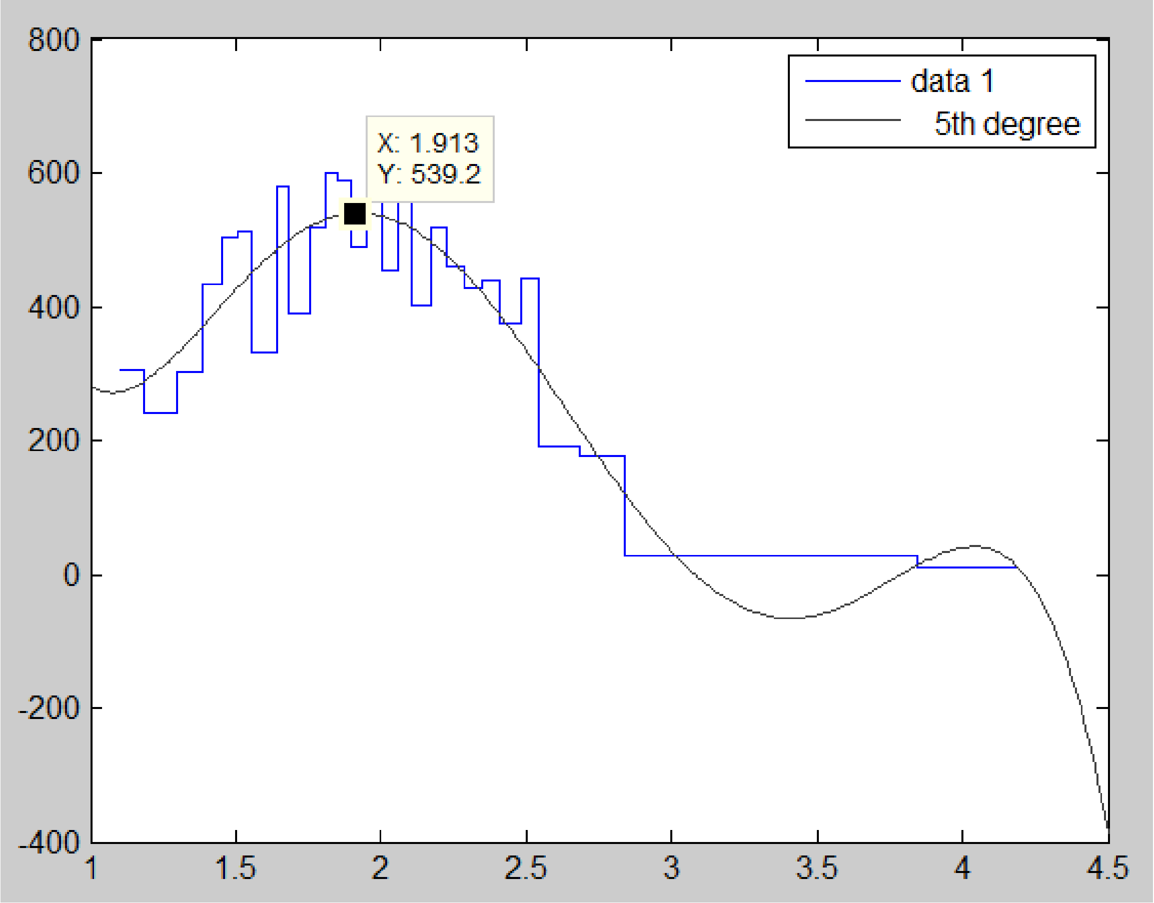


Figure 10: 120904162508 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps) and after the small end was removed by using the noise filter. A fifth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

<u>120904162645:</u> This one goes very much like the previous one, so I do not repeat all the plots and description. The final transit time qualified set has very little data (I'm guessing that the velocities vary more the larger the beads) so I go with the earlier plots (figs. 11 and 12) after using a noise filter of 1.2 V and obtain a reasonable but not ideal 3.1 V for these beads.

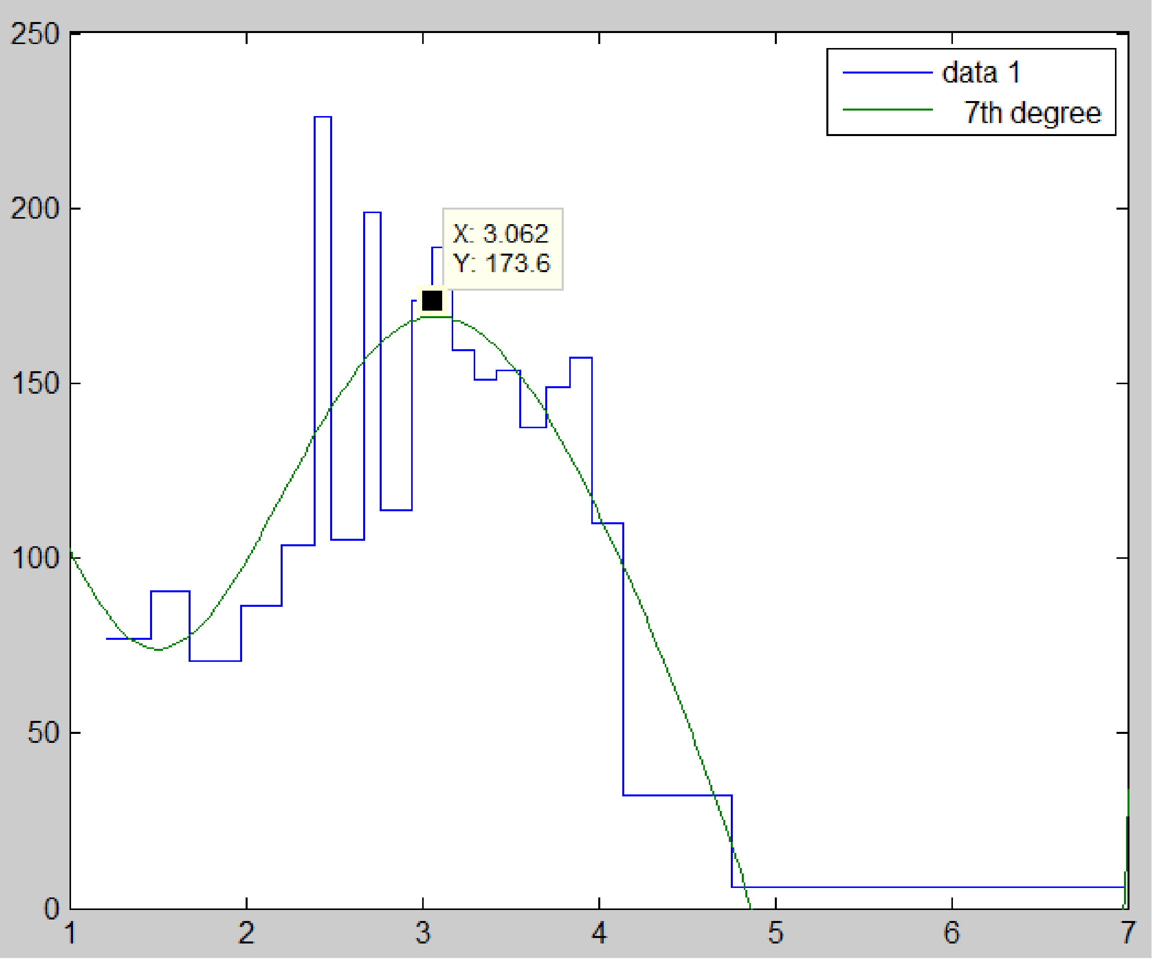


Figure 11: 120904162645 signal distribution after the DOF qualification step and after the small end was removed by using the noise filter. A seventh degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

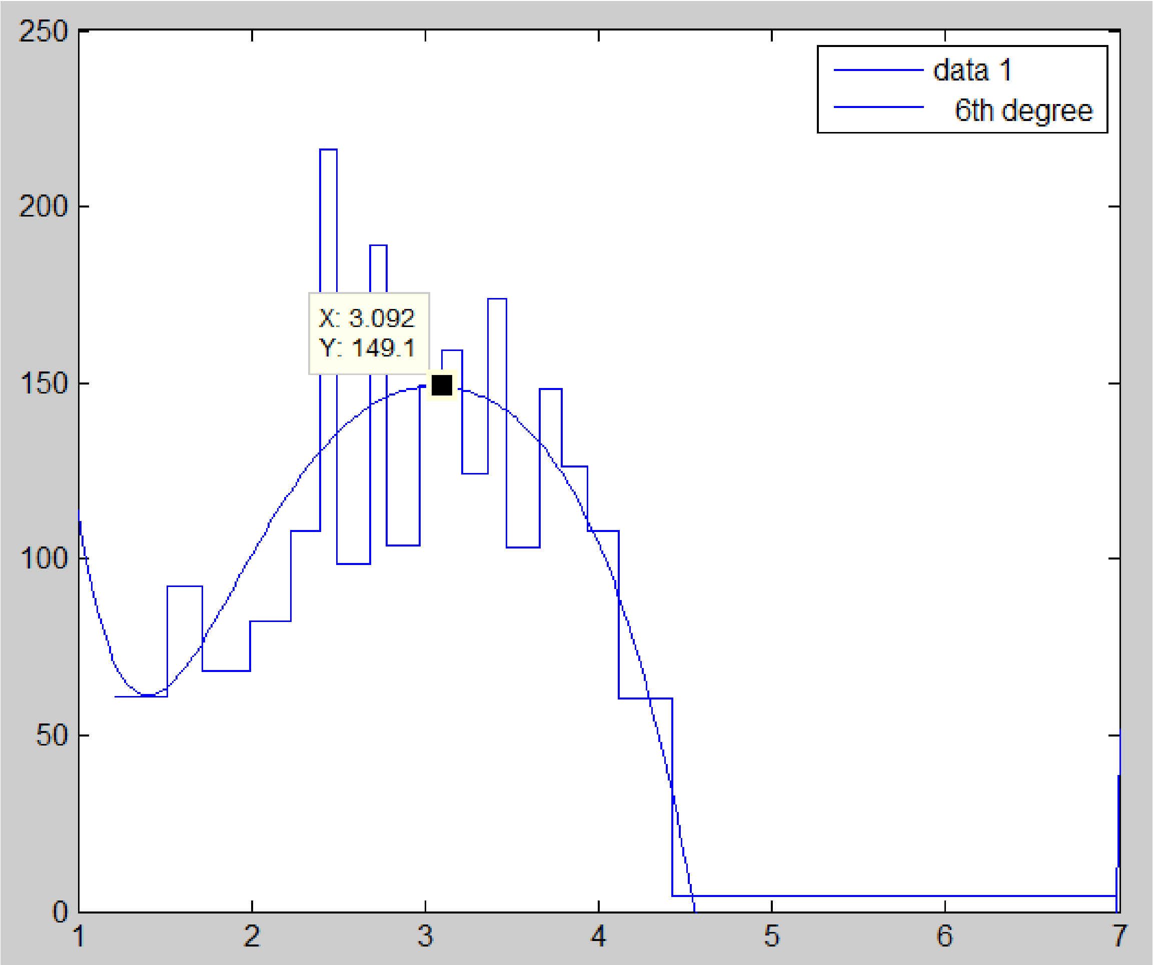


Figure 12: 120904162645 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps) and after the small end was removed by using the noise filter. A sixth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

120904162807: Again this one goes very much like the previous two. Figures 13 and 14 show the results after using a noise filter of 2.0 V and obtaining 4.5 V for these beads. Note the dip around 4 V. I'm guessing this is the gain switch point for this probe but am not sure about that. I was at around 2 V before.

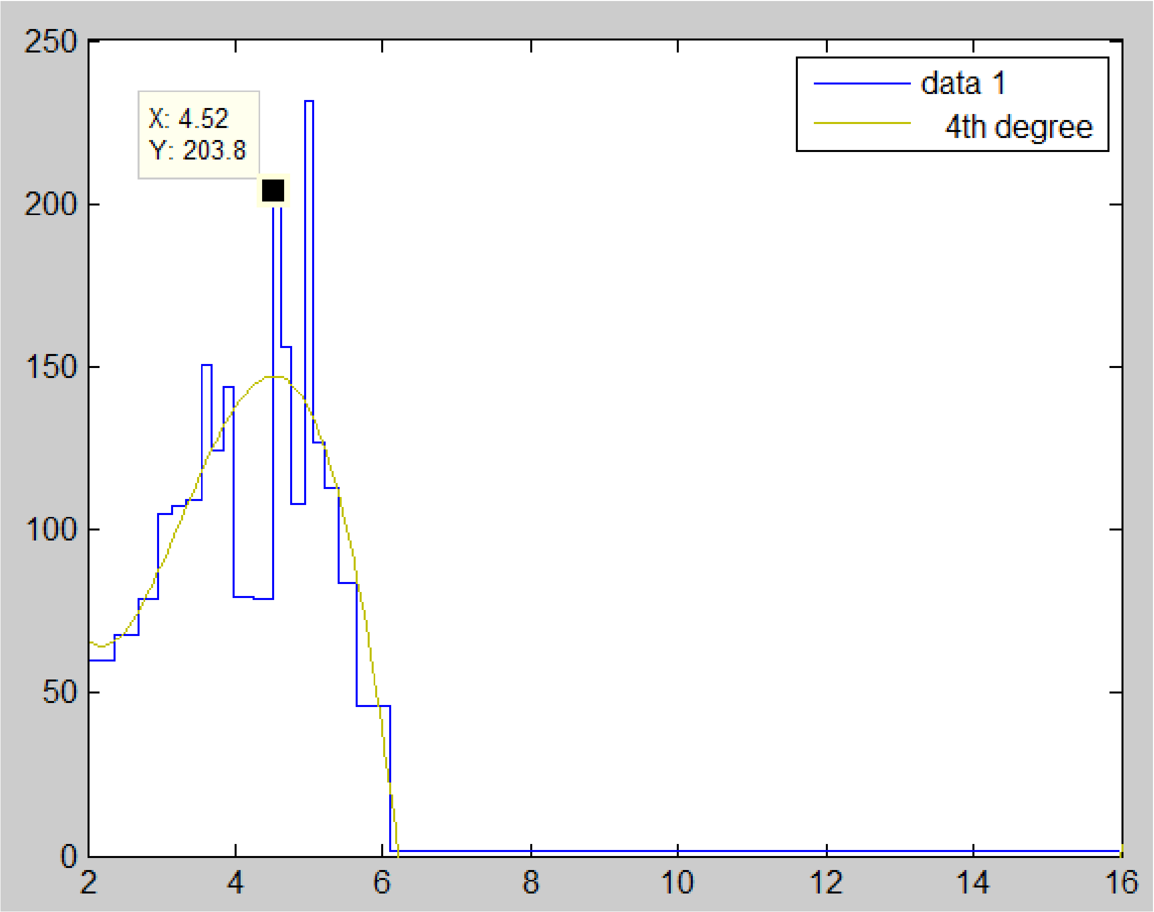


Figure 13: 120904162807 signal distribution after the DOF qualification step and after the small end was removed by using the noise filter. A fourth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

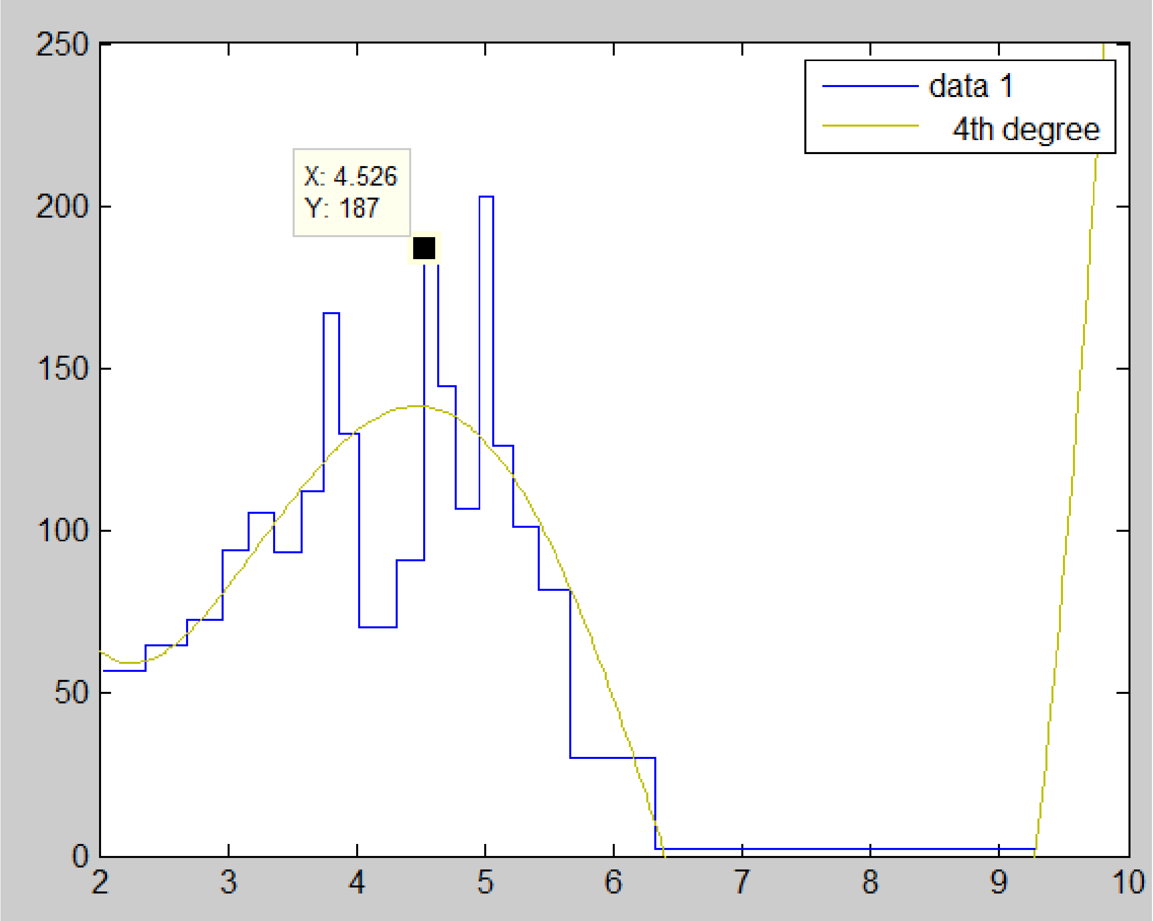
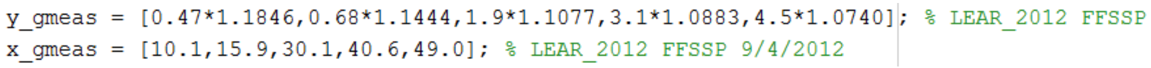


Figure 14: 120904162807 signal distribution after the half peak transit time versus full transit time qualification step (i.e. just before the final transit time qualification steps) and after the small end was removed by using the noise filter. A fourth degree poly fit is also shown that was used together with the data tip tool to identify the value at the peak.

Running the calibration: Besides these Voltages and bead sizes, for the FFSSP one other piece of information is needed, laser scale values. After inputting the data for each run, check the mean laser value. I.E. mean(laser) ... ans = 1.0740. The peak voltages obtained above must each be multiplied by its associated mean laser scale value and put into 'calibrate_glass2water_v#.m'. The values I obtained with these data are:

[1.1846,1.1444,1.1077,1.0883,1.0740]

They are used in 'calibrate_glass2water_v#.m' as multiplies to the peak voltage values obtained above, as follows:



```
y_gmeas = [0.47*1.1846,0.68*1.1444,1.9*1.1077,3.1*1.0883,4.5*1.0740]; % LEAR_2012 FFSSP
x_gmeas = [10.1,15.9,30.1,40.6,49.0]; % LEAR_2012 FFSSP 9/4/2012
```

Further, the variable mean_lvscale must be set. We recommend the minimum of the set of mean laser scale values obtained so far. E.G.:

mean_lvscale = min([1.1846,1.1444,1.1077,1.0883,1.0740]); % LEAR_2012 FFSSP calib data Seth 9/4/12

Then run ''calibrate_glass2water_v#.m ' and obtain the calibration plot (Fig. 15) and files. In this case the data points all look good except the first data point (10.1 um) is significantly high compared to the others, which is odd since it had the easiest to determine peak.

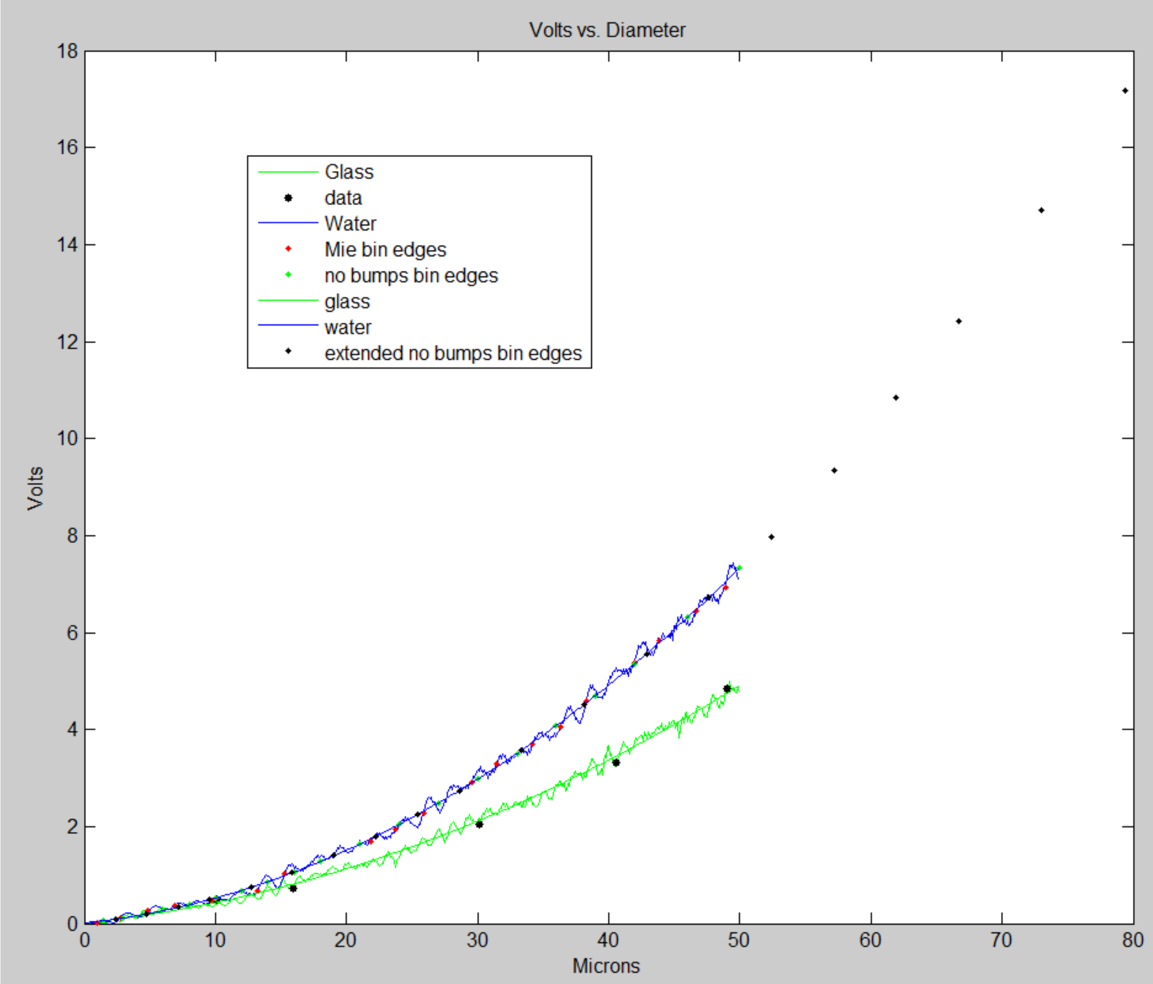


Figure 15:

Appendix D calling process particles and other code internal details

Copied from an earlier document back when the code was started:

SFFSSP code and results as of 3/2/2010:

The goal of 'parcel_periods.m' (Mo, I don't know what you call your program so I came up with this name) and 'process_standard_period.m' is to march through a data file processing on short periods (e.g. 1 Hz.) while yielding exactly the same results as processing on longer periods does. This however is not possible, without extreme measures and creating unacceptably odd situations. It is possible for most situations however and I have attempted to comment in the code situations where it will fail (e.g. at a transition from clear air (empty seconds) to cloud or where some events make it to the last rejection step (large transit time reject) but none pass the last step).

'parcel_periods.m' reads and parcels out the data into the sub periods for subsequent processing by 'process_standard_period.m'. Alternatively, 'input_txt_form.m' inputs set parameters and the particle data via a formatted text file produced from Darren's older code.

Either way the **input data and parameters are:** (all times are in ticks)

start_time (start time of the period being processed)

end_time (end time of the period being processed)

ser (an array of start times of each particle-detection-event relative to the period start time) (Mo, raw data uses event end times so be aware and get this right)

tt (an array of transit times of each particle-detection-event)

sig (an array of the signal peak value of each particle-detection-event)

qual (an array of the qualifier peak value for each particle-detection-event)

volt_sum (an array of values of the integrated signal for each particle-detection-event)

half_peak_trans_time (an array of values of time from event-start to event-signal-peak, for each particle-detection-event (Chris verify?))

laser (an array of values, one for each particle-detection-event that is a scaling factor used when converting signal peak value to particle size, it accounts for fluctuations in laser intensity (Chris verify?))

seq (an array of values, one for each particle-detection-event that is the time from the previous event's start time to the current event's start time (seq=ser(2:n)-ser(1:n-1)) the first seq value is the same as the first ser value (i.e. the time from period start time to the first event's start time))

w8s (an array of values, one for each particle-detection-event that is the time from the previous event's end time to the current event's start time (w8s=ser(2:n)-ser(1:n-1)-tt(1:n-1)) the first w8s value is the same as the first ser value (i.e. the time from period start time to the first event's start time))

overflow_dead_time (the amount of time the probe spent in overflow between the period start and end times)

last_live_period (the time from the end of the period's last event to the period's end_time)

first_w8 (the time from the end of the last non-noise event before the period's start_time to the start of the first non-noise event in the time period. note 'remuv_noise' must be set to 'y' or it is just the first (last) event etc. (set to 25001 if the is no qualifying event before the period's start time))

next_w8_a (the time from the end of the last non-noise event in the time period to the start of the next non-noise event after the end time of the period (default if no such next event exists = 25001) (note that next_w8_a of one period is the first_w8 of the next period)

next_w8_b (the time from the end of that first non-noise event after the period's end_time to the start of the next non-noise event (2nd non-noise event following the period being processed))

next_wait_a (this is the time from the period's end_time to the start of the next non-noise event. Note that if, and only if, the period's last event is not a noise event, then last_live_period + next_wait_a = next_w8_a (default if no such next event exists = 25001)) note all 25001 should really be aggressive_cut_thresh + 1. Mo will implement that in parcel_periods.m

TAS_method (may be either 'set' or 'est', 'set' implies the processing will use the value input here, 'est' implies the program will estimate the TAS from transit times)

TAS (a true air speed must be input, if none available use 16850 (cm/s) and set TAS_method = 'est')

show_plots (may be 'y' or 'n', 'y' implies the program will output figures and text values at each stage of the processing)

fishit (is just like 'show_plots' but specifically for the fishing test results, this is because fishing can take much longer than the other plots and so one may want to suppress fishing while still viewing the other many plots and printouts)

remuv_noise (may be 'y' or 'n' and tells the program whether to remove noise based on the next parameter or not) noise_string (a string that specifies the Boolean logic statement that determines which events are accepted as non-noise events (e.g. 'accept=find(tt > 10 & sig > 0.05);'))

SH_method (may be 'none   ', 'adaptive2 ', or 'aggressive' and sets which of these methods the program will use for shattering rejection (adaptive1 was disabled to save time but could be resurrected if need be))

tt_method (may be ‘via size’, ‘old way’, or ‘none’ and sets which of these methods is used for transit time rejection)

probe_type (may be ‘FFSSP’, ‘uFSSP’, or ‘F_CDP’ and sets parameters according to the type of probe data being processed)

agg_cut_thresh (the cutoff threshold for doing aggressive shattering removal, this had to be put in parcel_periods.m so that the default values of next_ect_etc could be set to agg_cut_thresh + 1, instead of 25001 as written above)

October 2012 add new parameter to pass 'next_tt the transit time of the first real event noise or not to follow the period's end time.

Note some of the special inputs above refer to 'any event' versus 'non-noise events'. If remove_noise = 'y' then parcel periods uses noise_string to identify noise events.

<u>The program outputs are:</u>

PSD_B (the size distribution in counts per bin and concentration per bin (#/L/um))

SV (the sample volume which can be used to weight the time periods when averaging)

tot_cnts (the number of accepted events in the time period).

There are many other parameters calculated by process_standard_period.m but they are not to be output. In particular, concentration, extinction, and LWC are to be calculated from PSD_B in parcel_periods.m before outputting to the archive file. tot_cnts however should not be calculated from PSD_B, it must be obtained from process_standard_period.m. This is because I have used tot_cnts in a special way for the cases where one or more events make it to the last rejection step (large transit times) but none make it past that step. In this case averaging the one Hz data will not equal the longer period processing unless those events are accounted for. So in that situation where tot_cnts is zero, I make tot_cnts equal to the negative of the number of events that were rejected as too long of transit times.

Add how to use those events to obtain the longer period result from the 1 Hz results...

The output archive file and a separate processing_log.txt file will contain the following:

Raw data input file name, all processing parameter settings (listed above as inputs to 'process_standard_period.m'), the date and time the processing is completed (or started if that is easier), output filename(s). In addition the processing_log.txt file will contain any warnings due to idiosyncratic processing issues (none are defined at this time but Mo is checking as he reads and inputs the data for any timing irregularities).

Other details:

Overflow periods: These are essentially ignored for all relative time periods (w8 times e.g.) and are summed up and input as 'overflow_dead_time' as defined above. There are strange issues however, that must be dealt with. First, one would expect an overflow period to begin at the end of a detection event. i.e. if the probe is validly live 'now' and nothing is detected, why should it suddenly go into overload? However, Mo finds (for the limited data searched through so far) that overflow periods start some 13 or 14 ticks after a real event ends. Chris says this is because it is writing some 12 words to the FIFO and those take 1 tick per word to write. So Mo is making the overflow periods start at the end of the previous particle detection event (he is also testing that this backup of the overflow start time does not add more than 15 ticks and will keyboard stop if it does, as a check). Second is that there are particle detection events that have start times well (more than 14 ticks e.g.) before the end of an overflow period. These occur because the writing of an event does not occur until the end of the event is reached; the FIFO is cleared for writing by this time. For these, Mo is extending the overflow dead period until the end of the event. This seems a reasonable choice because keeping the event would imply a different sample volume depending on particle size. This is accomplished during the first pre-processing run through the data file and instead of extending the overflow period he is making that next period into its own pseudo overflow period.

Events that cross period boundaries: If an overflow starts in one period and ends in another, it is simply split between the periods. If a noise event starts in one period and ends in another, the event is simply completely ignored (Mo's idea, OK cause bulk processing will do the same). The first part of the noise event, at the end of the first period, is included in last_live_period. The last part of the noise event, at the beginning of the second period, must be included in next_wait_a while all of the noise event transit time must be included in next_w8s_a and next_w8s_b. If a non-noise particle detection event starts in one period and ends in another, we move the end_time of the first period to the end of that event, so that it exists entirely in the period. OK Later Mo prefers to treat noise events that crossover the same as non-noise particle detection events that cross over.

Appendix E Tutorial on transit time calibration (C_1 and C_3)

I'm writing this as I attempt to use my just written routine (fit_cp.m) to facilitate this calibration for the first time. I'll save plots as I go and hopefully turn this into a more finished tutorial in time. I pioneered the proof of concept manually using FFSSP data. Now I want to try it on F_CDP data. SPEC's most recent F_CDP data has a problem with spuriously extended transit times and thus is not suitable for this work. So I am going back to ICE-T data. I randomly scan the data and see I picked out four periods already from RF4_7-12-2011 so I'll just try that. The first period (110712175445_frm_1053_to_1057_PBP) has little actual data (probably was looking at clear air noise/aerosol situ) so skipping that. Next period (110712175445_frm_1441_to_1444_PBP) is excellent for this as the data looks clean and there is only an easily recognizable and small amount of coincidence.

So I input it and run process particles (shattering and noise reduction are off). Figure 1 shows the post half peak versus full transit time qualification step plot of half peak transit times versus size.

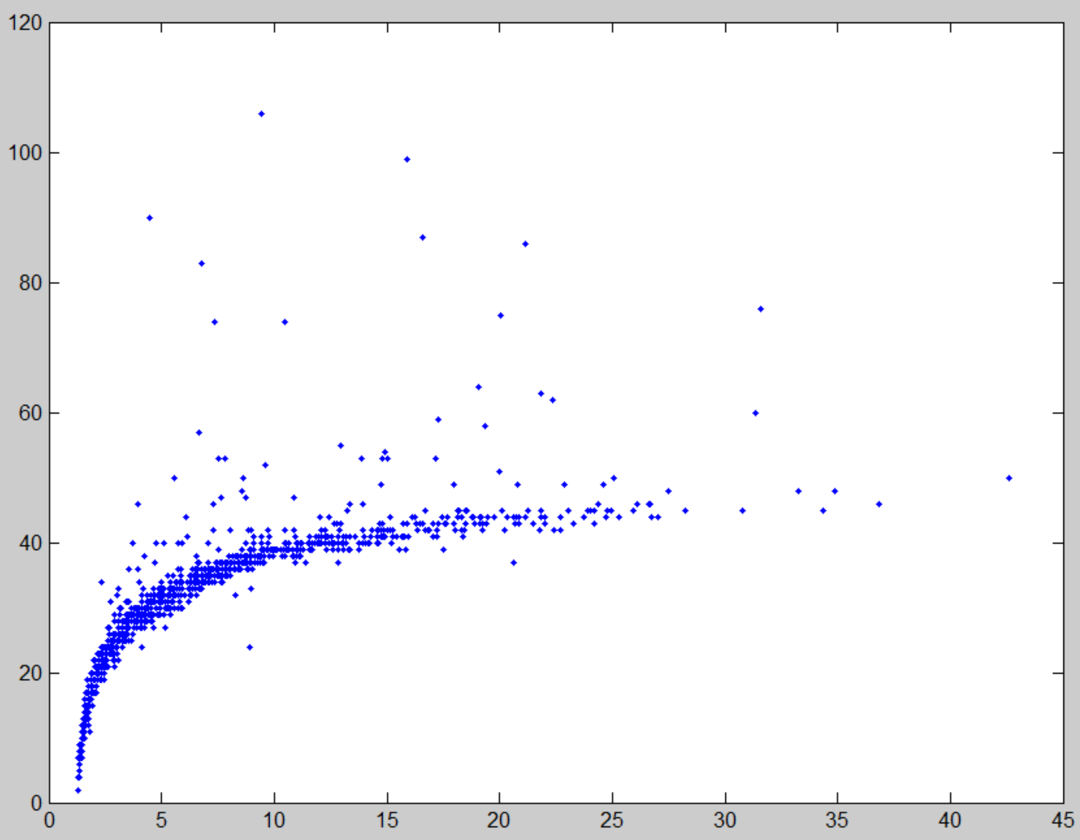


Figure 1: Half peak transit times versus size for the ICE-T FCDP RF4_7-12-2011 110712175445_frm_1441_to_1444_PBP.

I click on the 'data_cursor' tool button and then click on one of the data points that defines the upper boundary between good data and coincident events, then right click on the datatip and click Export Cursor Data to Workspace (Fig. 2) and name the data cp1. I repeat this 8 times changing the data names to cp2, cp3 etc... each time.

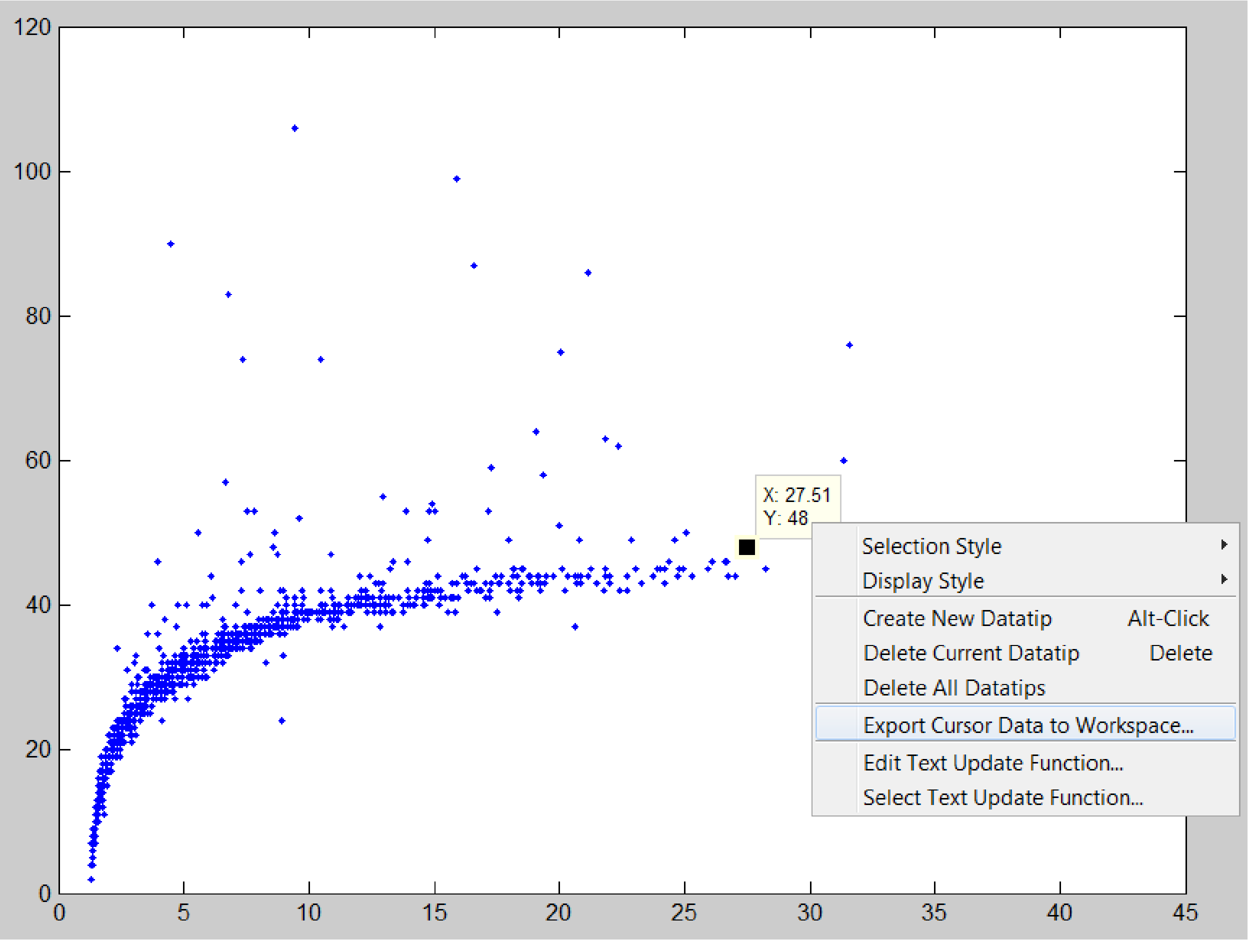

120
100
80
60
40
20
0
0
5
10
15
20
25
30
35
40
45
X: 27.51
Y: 48
Selection Style
Display Style
Create New Datatip Alt-Click
Delete Current Datatip Delete
Delete All Datatips
Export Cursor Data to Workspace...
Edit Text Update Function...
Select Text Update Function...

Figure 1: Half peak transit times versus size for the ICE-T FCDP RF4_7-12-2011 110712175445_frm_1441_to_1444_PBP, demonstrating the use of the data cursor to capture calibration data points.

Next I run fit_cp.m, which looks for and uses data structures named cp1, cp2, etc... and yields the calibration coefficients C_1 and C_3 and the plot with linear regression used to determine them. I insert those values in setup.m and rerun the same data file, with the excellent result shown in Fig. 3.

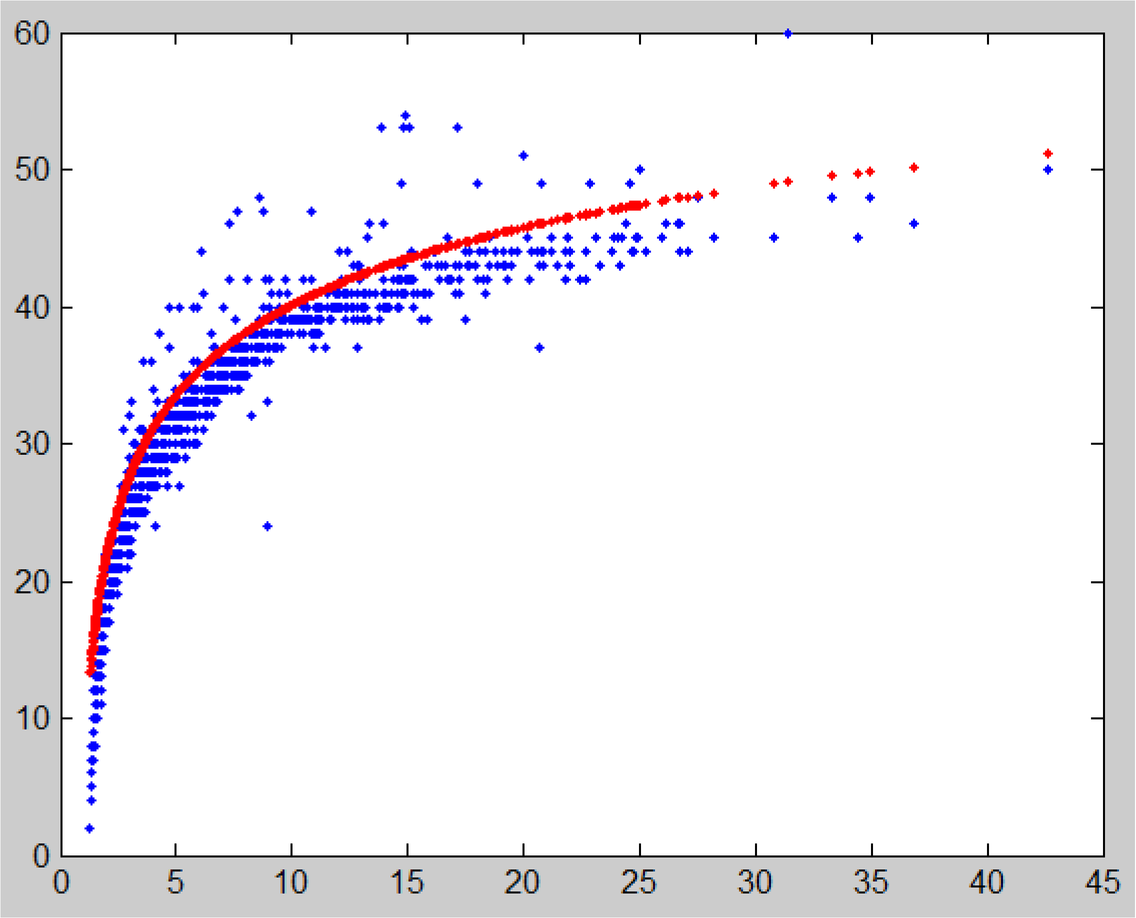


Figure 3: the same as figs. 1 and 2 but with the ideal estimated transit times versus size in red.

Next I input and run on the period 110712175445_frm_5228_to_5236_PBP yielding results shown in Figs. 4 and 5 and indicating that this method works better than my previous method for FCDP and also that I need to adjust the criteria for acceptance in the code. Figure 5 also shows the result after adjusting the criteria.

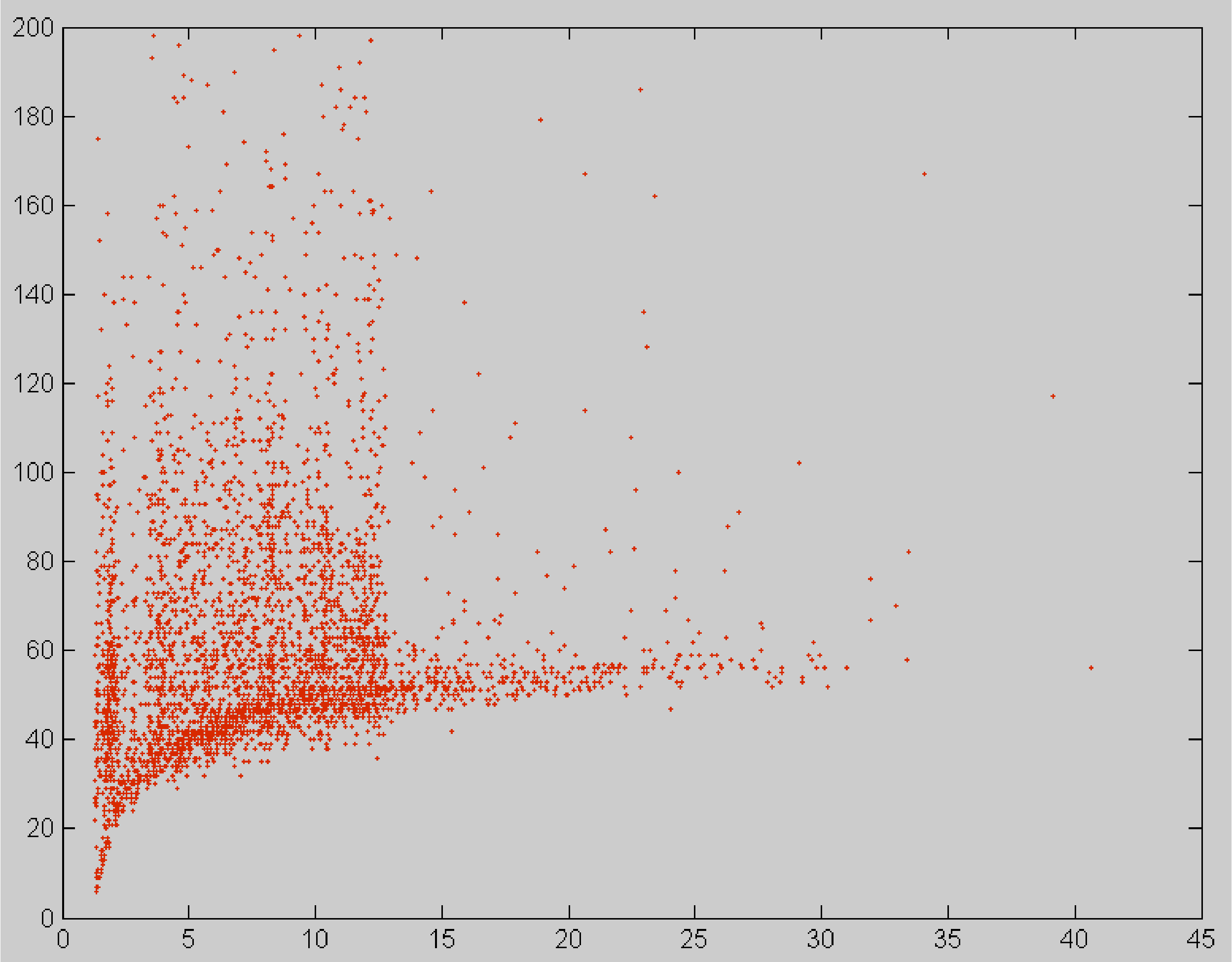


Figure 4: Half peak transit times versus size for the ICE-T FCDP 110712175445_frm_5228_to_5236_PBP data after slit comparator qualification.

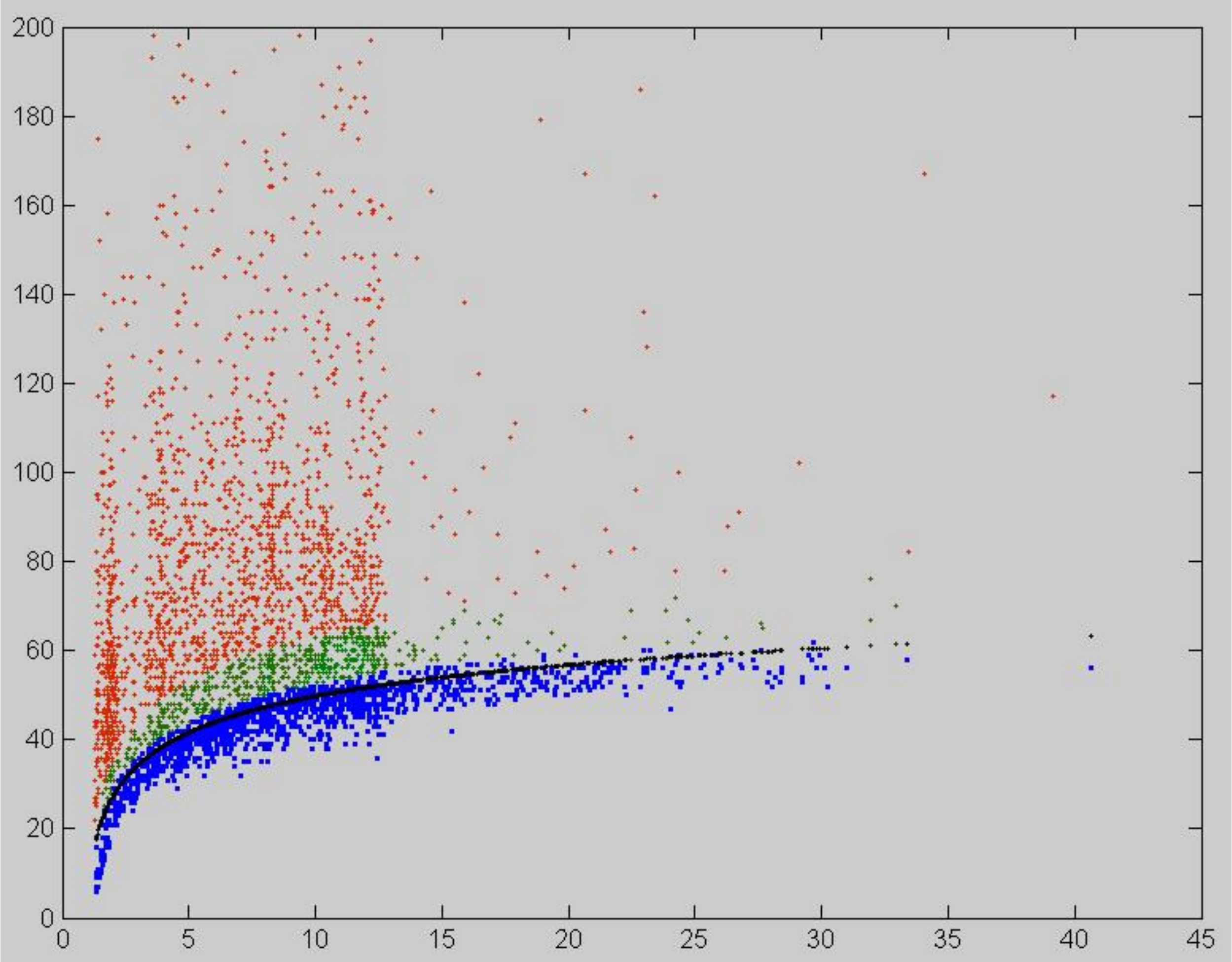


Figure 5: Half peak transit times versus size for the ICE-T FCDP 110712175445_frm_5228_to_5236_PBP data after slit comparator qualification in red. Black data shows the ideal transit times estimated from sizes while green and blue overlays show those events also size dependent transit time qualified by the older and new criteria respectively.