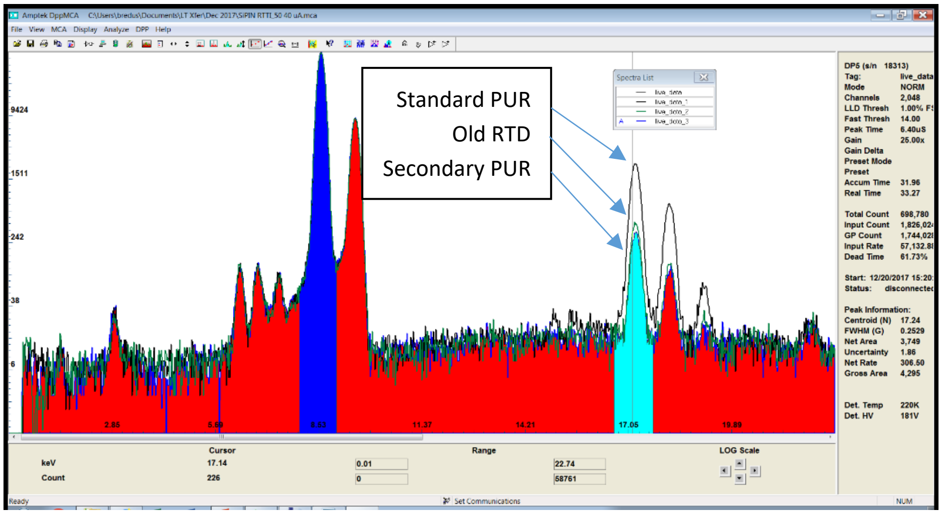
Using Amptek's Secondary Pile-Up Rejection

Summary

- Amptek is releasing a new version of the firmware for its DP5 family of digital pulse processors, found in the DP5, the X123, the DP5-X, and the X-55.
- This firmware includes "secondary pile-up rejection (PUR)" to significantly reduce the pulse pair resolving time and thus sum peak intensity.
- Secondary PUR operates in addition to the regular pile-up rejection, detecting pile-up which cannot be detected in the standard algorithm. The spectra below illustrate its effect.



- Secondary PUR examines the risetime of the input signal. The risetime is longer for pulses which overlap in the fast channel; this is the basis for rejecting closely spaced pulses.
- To use Secondary PUR, one must install the latest FW release (DP5: FW6.10.04/FP7.07; DP5-X: FW6.10.04/FP6.10) and then command a single parameter, called PURS. This is found on the "shaping" tab in DPPMCA. It can also be commanded in the SDK.
- Amptek recommends 120 ns for the 25 mm^2 FAST SDD®, 150 ns for the 70 mm^2 FAST SDD® and the 25 mm^2 SDD, and 300 ns for the SiPIN detectors. The pulse pair resolving time is approximately equal to PURS , in nanoseconds.
- Secondary PUR is not compatible with the fastest fast-channel peaking times (50ns @ 80MHz, 200ns @ 20MHz)
- If the PURS parameter is set too low, it will reject valid pulses. Care must be taken to avoid this.

Why use the new algorithm?

Pile-up rejection (PUR) is very important for high count rate spectroscopy: X-rays interact in the detector at random times, and when the time between interactions is shorter than the peaking time, the result is a distorted pulse with incorrect amplitude. Amptek's firmware has long included pile-up rejection logic based on a fast/slow channel approach, as illustrated below. There are two signal processing chains: a slow channel (optimized to give the best pulse height resolution) and a fast channel (optimized to operate at high speed, to simply detect pulses over threshold). In the standard firmware, if the processor detects two pulses in the fast channel which occur within the peaking time of the slow channel, the slow channel pulse is rejected, i.e. its pulse height is not reflected in the spectrum.

This method works quite well for most pile-up but it has two challenges. First, it cannot distinguish pulses which overlap in the fast channel, i.e. which occur within the fast channels peaking time (as shown in the right below). These pulses lead to a slow channel pulse with amplitude equal to the sum of the input pulses, giving rise to a sum peak in the spectrum (as an aside, the slow channel flat top should always exceed the fast channel peaking time, to produce sum peaks which are of minimal width and Gaussian shape). Second, there is more noise in the fast channel (it has a higher noise bandwidth) so a higher noise threshold; it cannot detect pile-up involving low amplitude pulses, those below the fast channel's threshold. One can address the first challenge by using a shorter fast channel peaking time, but this exacerbates the second challenge by increasing the fast threshold, or vice-versa.

Amptek's new algorithm addresses these shortcomings by adding a parallel signal processing chain, which has a much shorter pulse pair resolving time than the fast channel. It does have a higher noise threshold; it lets one reject large pulses within a very narrow resolving time, while relying on the previous PUR algorithm to reject smaller pulses albeit with a longer pulse pair resolving time.

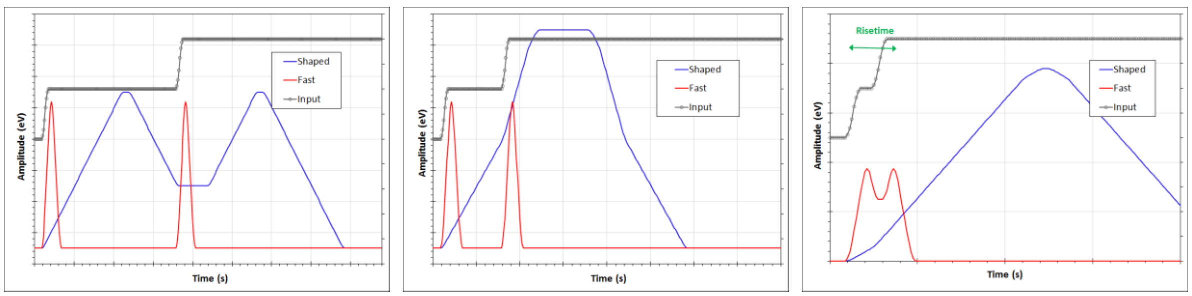


Figure illustrating secondary PUR operation. Left: Two pulses which are well separated in time, not piled up. The system accepts both and measures the correct amplitude for both. Middle: Two pulses which are rejected by the standards PUR. Though overlapping in the slow channel, they are distinct in the fast channel. Right: Two pulses which are piled up closely. Standard PUR does not identify this as pile-up, since there is a single pulse in the fast channel. Secondary PUR will identify these are pileup due to the long risetime in the input signal.

How does it work?

The new algorithm is based on measuring the risetime of the pulse input to the ADC. As can be seen in the plots above, when pulses overlap in the fast channel, the risetime is equal to the sum of the two risetimes plus the delay. Even when the delay is less than the risetime, the overall risetime is longer than that of a single pulse. So the algorithm directly measures this risetime. High frequency noise does limit the amplitude of pulses which are detected and adds jitter to the risetime measurement itself. The signal pulses exhibit a range of risetimes, so the minimum pulse pair resolving time is the maximum signal risetime plus some margin for jitter.

Amptek's previous firmware had a risetime discrimination algorithm (RTD) which could also be used to reduce the sum peak intensity, but the new algorithm provides both better operation (a shorter pulse pair resolving time and lower jitter) and an improved user interface (a single parameter with a clear physical meaning and which is independent of other settings). The pulse pair resolving time is very close to the commanded PURS parameter, which is set in nsec. The pulse pair resolving time (which determines the sum peak intensity) equals PURS.

Note that normal pulses have some risetime; if PURS is commanded to a value below this, then valid pulses are rejected. This is not recommended.

How do I use it?

Install the firmware

Use the Firmware Manager application to update the DP5 or DP5-X firmware to the versions listed above, or later. Note: FW versions which support Secondary PURS do not support List Mode.

Configuring the parameters

There is a single parameter which must be set, the *PURS parameter*: This is the risetime threshold; pulses with longer risetime are rejected. Setting PURS=0 disables advanced PUR and the system relies purely on standard PUR. The maximum setting is 310 ns for an 80 MHz clock and 1240 ns for a 20 MHz clock. Once the system has been configured, the fast and slow channel peaking times, gains, etc can be altered without needing to change the PURS parameter.

We recommend setting the flat top duration to be longer than PURS. If the flat top duration is too short, then the sum peak exhibits a clearly non-Gaussian shape with a low side tail.

Testing the setup

If the settings are incorrect, i.e. if PURS is set too low, the logic will reject valid events. If PURS is too high, there will be more sum peak counts than desirable. Therefore, after configuring the system we recommend the following validation check:

1) Configure everything in the system. Then, set PURS=0 in order to turn off the risetime algorithm. Acquire a spectrum using a sample with several photopeaks, broadly spread in energy, at a low count rate, where sum peaks and pile-up are negligible and save this spectrum.
2) Set PURS. With excitation conditions identical, i.e. same low rate, acquire another spectrum for the same amount of time. Save the spectrum, measure the photopeak intensities, and compare to the first spectrum. There should be negligible change.
3) If the second set of photopeak intensities is lower than the first, increase PURS. .
4) Set PURS=0 to turn off advanced PUR. Now acquire a spectrum with only one or two very strong peaks at a high count rate (50% dead time). Save the spectrum and measure both photopeak and sum peak intensities.
5) Set PURS to your selected value. Acquire a spectrum under the same conditions. Save the spectrum and measure both photopeak and sum peak intensities. The sum peak intensity should be considerably reduced.

The pulse pair resolving time T_{pair} can be estimated from the photopeak count rate (R_1) and the sum peak count rate (R_{sum}) using the expression below. This assumes the main sum peak has energy $2xE_1$, the main peak in the spectrum. The parameter T_{pair} is important for software algorithms used to correct for sum peaks.

$$R_{sum} = \frac{1}{2} R_1^2 T_{pair} \quad \Rightarrow \quad T_{pair} = \frac{2R_{sum}}{R_1^2}$$

The sum peak should have a photopeak shape which is Gaussian and with a FWHM determined from the sum of energies of the two events. The peak at E_1+E_2 will have width

$$\delta E = \sqrt{ENC^2 + F_{Fano}\left(E_1 + E_2\right)}$$

where ENC is the electronic noise and F_{Fano} is the Fano factor.