If you have trouble viewing this, try the pdf of this post. You can download the code used to produce the figures in this post.

SNR with pileup

In this post I conclude my discussion of my “SNR with pileup ...” paper[1]. I will present and explain the code to reproduce the “bottom line” figures 3 to 5 of the paper that show the decrease of the SNR as the pileup parameter η increases. The decrease is rapid and when the value of η reaches 1, all the counting and PHA detectors have SNR smaller than the energy integrating detector. The NQ detector SNR decreases rapidly and approaches but marginally stays above the Q SNR since it uses that signal.

The formulas and code

The complete code to compute the SNR for all detector types as a function of η is in cell 4 of the overall code for this post, $P53SNRpileup6.m$

The parameters required to compute the SNR with pileup are the spectrum incident on the detector and η. The SNR of the ideal, Q, N, and NQ detector are computed by the $SNR\_with\_pileup.m$ function. The ideal detector SNR was computed using Eq. 36 of my “Near optimal ...” paper[2].

$$SNR^2_{ideal} = \frac{t_f^2}{2} \int n_T(E) \left[\delta\mu(E)\right]^2 dE. \quad (1)$$

The parameters are explained in the paper: $n_T(E)$ is the spectrum transmitted through the object and incident on the detector, t_f is the thickness of the feature in the same units as the linear attenuation coefficient μ, $\delta\mu(E)$ is the difference of the basis function attenuation coefficients. The leading term, $t_f^2/2$ was set equal to one in the code since it is the same for all of the detectors. The ideal SNR was implemented with this code

```
dMus = diff(specdat.mus,1,2); % difference along rows
SNR_ideal = trapz(specdat.egys,specdat.specnum.*(dMus.^2));
```

The Q detector SNR, from Eq. 34 of the “Near optimal ...” paper[2], is

$$SNR^2_Q = \frac{\lambda t_f^2}{2F} \left\langle \delta\mu \right\rangle^2_Q . \quad (2)$$

where the notation

$$\left\langle \mu(E) \right\rangle_Q = \int \mu(E)\hat{q}(E)dE$$

means the effective value in the normalized energy spectrum $\hat{q}(E)$. In the equation, λ is the integral of the photon number spectrum, which is the total number of photons, F is the excess variance

$$F = \frac{\left\langle \mathbf{E}^2 \right\rangle_N}{\left\langle \mathbf{E} \right\rangle^2_N}$$

where the notation $\langle\ \rangle_N$ means the effective value of a variable in the normalized photon number spectrum. This was implemented with the following code

```
specdat.specegy = specdat.egys.*specdat.specnum;
specdat.specegy_norm = specdat.specegy/trapz(specdat.egys,specdat.specegy);
    % SNR of Q (energy integrating) not affected by pileup
dmusQ = trapz(specdat.egys,specdat.specegy_norm.*dMus);
SNR_Q = nbar_incident*dmusQ^2/F;
```

The photon counting N detector SNR used the variance and the M matrix with pileup computed by the NQ detector CRLB function, CRLB_NQ_with_pileup.m. with Eq. 20 of the “Near optimal ...” paper

$$SNR_N^2 = (\Delta A)^T M^T R_L^{-1} M (\Delta A)$$

where $\mathbf{\Delta A}$ is the difference of the A-vectors in the background and feature regions. As explained in the “Near optimal ...” paper, by using the attenuation coefficients of the feature and background regions and, since $t_f^2/2$ was set equal to one in all of the SNR formulas, this was set equal to

$$\mathbf{\Delta A} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

For the detectors with pileup, the SNR was computed as

$$SNR^2 = \mathbf{\Delta A^T C_A^{-1} \Delta A} \qquad (3)$$

where $\mathbf{C_A}$ is the A-vector covariance. The CRLB was used for the covariance since it gives an optimal value that does not depend on the type of estimator. The CRLB functions used the full formula for the Fisher matrix discussed in the last post. Since the Fisher matrix $\mathbf{F}$ is the inverse of the CRLB, this was used directly in Eq. 3 instead of taking the inverse twice.
The NQ and N detector SNR were implemented with this code

```
    % SNR NQ detector
[~, nqstats] = CRLB_NQ_with_pileup(specdat,eta);
SNR_NQ = dA'*nqstats.F*dA;
    % SNR N detector with pileup
M_N = nqstats.M(1,:); % eff mus for photon count spectrum with pileup
SNR_N = dA'*M_N'*(1/nqstats.RL(1,1))*M_N*dA; % nqstats.RL is cov of log of NrecQ
```

The CRLB with PHA was computed by the CRLB_PHA_with_pileup.m function. This function used the idx_threshold field of the spectrum structure, which is the indexes into the $egys$ vector of the PHA bins. This was implemented with the following code from the overall function for this post $P53SNRpileup6.m$:

```
for kbin = 1:nbins2test
    specdat.idx_threshold = idx_thresholds{kbin};
    [~,pha_stats] = CRLB_PHA_with_pileup(specdat,eta);
    SNR_PHA = dA'*(pha_stats.F)*dA; % F is the inverse of the A-vector CRLB
    ...
```

The optimum PHA bins for each value of η are computed by the $OptimNbinsThresholdsSNR.m$ function. This uses a brute force algorithm of computing the SNR for each possible set of bins that span the energy spectrum. Since this rapidly leads to a huge number of combinations the function uses the min_bin_width variable to set the minimum bin. The search is then in two steps: first the optimal bins are computed to within a minimum width, then the combinations around this optimal set are searched. Even with this approach, the computation takes a long time. The code to implement this is in the box

```
if get_optimal_bins && (kbin==2)
        %compute PHA4 SNR with optimal bins with pileup
    sp_w_pileup = SpectrumWithPileup(specdat,eta);
         % recompute mus for energies in pileup spectrum
    sp_w_pileup.mus = MusFromMaterials(materials,sp_w_pileup.egys);
       % use 2 steps to get optimal bins.
       % First coarse search
    [idx_threshold1,Ethreshold1] = OptimNbinsThresholdsSNR(sp_w_pileup,nbins(kbin), ...
        'min_bin_width',2,'dynamic_range', 100,'verbose');
       % then fine search
   [idx_thresholdSNR,EthresholdSNR] = OptimNbinsThresholdsSNR(sp_w_pileup,nbins(kbin), ...
        'min_bin_width',1,'dynamic_range', 100, ...
        'close_egys',Ethreshold1, 'close_dist',5,'verbose');
    specdat.idx_threshold =  idx_thresholdSNR;
    [~,pha_stats] = CRLB_PHA_with_pileup(specdat,eta);
    SNR_PHA_opt = dA'*(pha_stats.F)*dA;
    SNR_with_deadtime_opt(keta, 4+kbin) = SNR_PHA_opt;
    Ethreshold_opt = [Ethreshold_opt EthresholdSNR];
end
```

Results

The graphs in Figs. 1 and 3 should be compared with Figs. 3 and 5 of the paper. Fig. 2 was not in the paper but is interesting.

Conclusions

The implications of these results are described in the paper. As discussed in the first post of this series,

> The basic assumption that I will make is that these idealized models provide an upper limit to the SNR of real detectors. Obviously, I cannot prove this but it is reasonable to assume that any additional distortions of the data will decrease the SNR. The results will apply to a single measurement so all bets are off if you are, for example, spatially filtering to combine the results from several detectors or making more than one measurement at different times with the same detector and combining the results. Iterative algorithms with “regularity” or smoothness conditions also combine results from different measurements nonlinearly so they will have different SNRs. But in all these cases, having higher quality input data will most likely lead to better final results.

Last edited February 9, 2015

References

[1] R. E. Alvarez, “Signal to noise ratio of energy selective x-ray photon counting systems with pileup,” *Med. Phys.* **41**, no. 11, –, (2014).

[2] R. E. Alvarez, “Near optimal energy selective x-ray imaging system performance with simple detectors,” *Med. Phys.* **37**, 822–841, (2010).

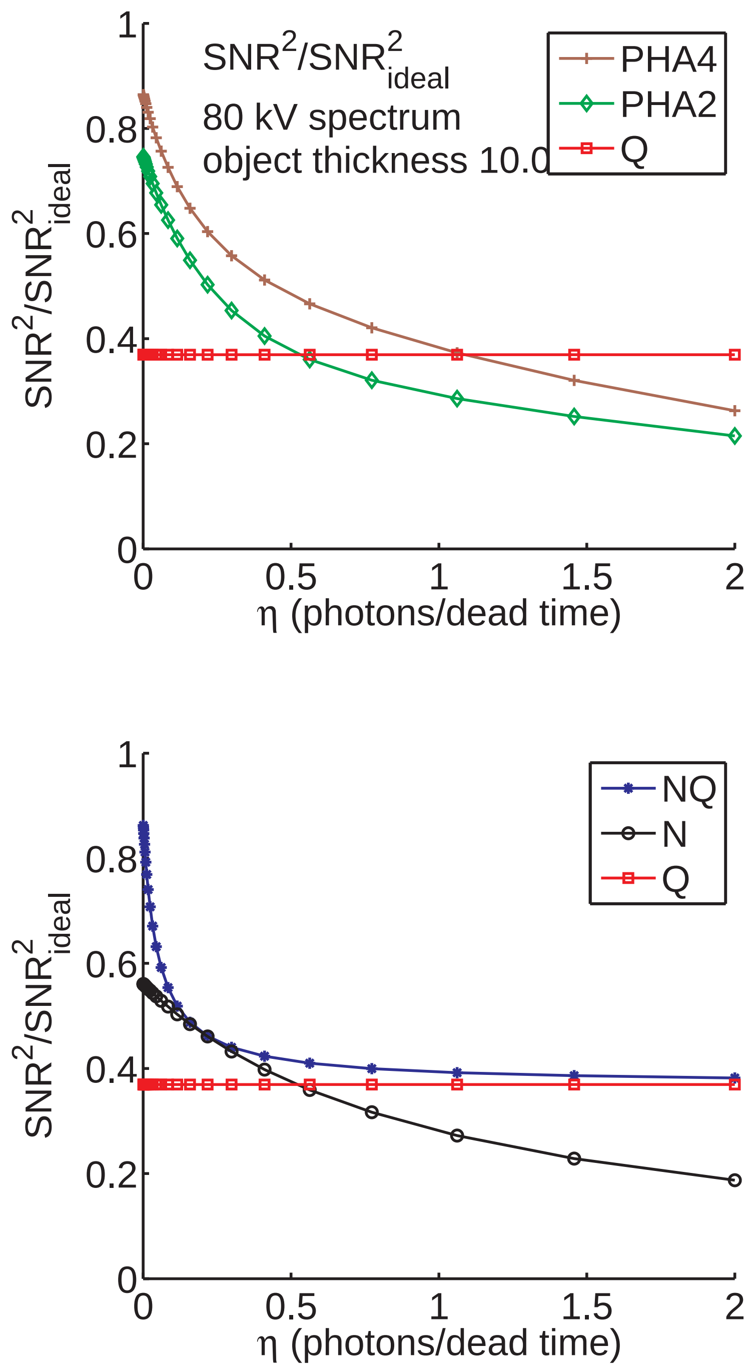

Figure 1: Normalized SNR as a function of counts per dead time. The SNR values are normalized by dividing by the ideal SNR with complete energy spectrum information.

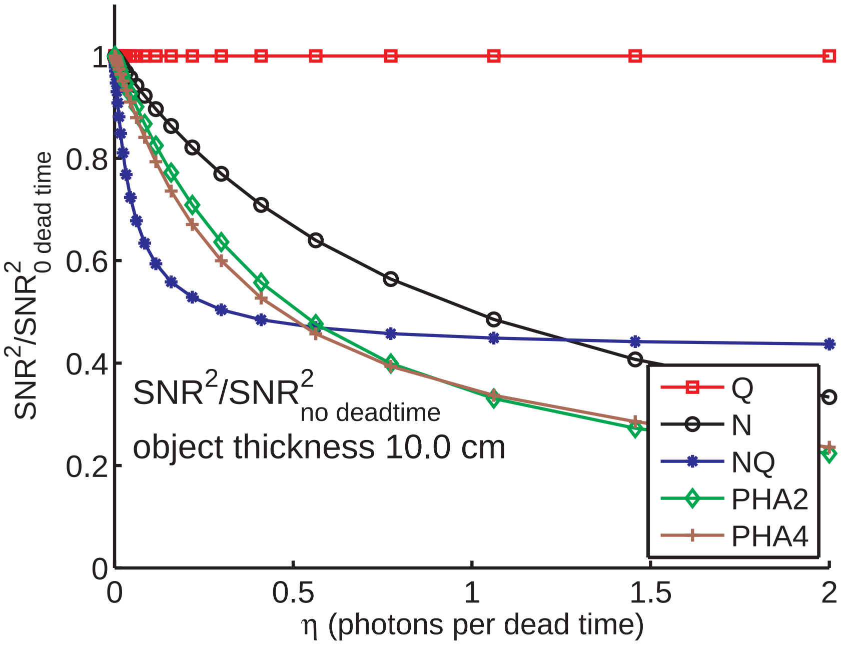

Figure 2: SNR divided by SNR at zero dead time as a function of counts per deadtime.

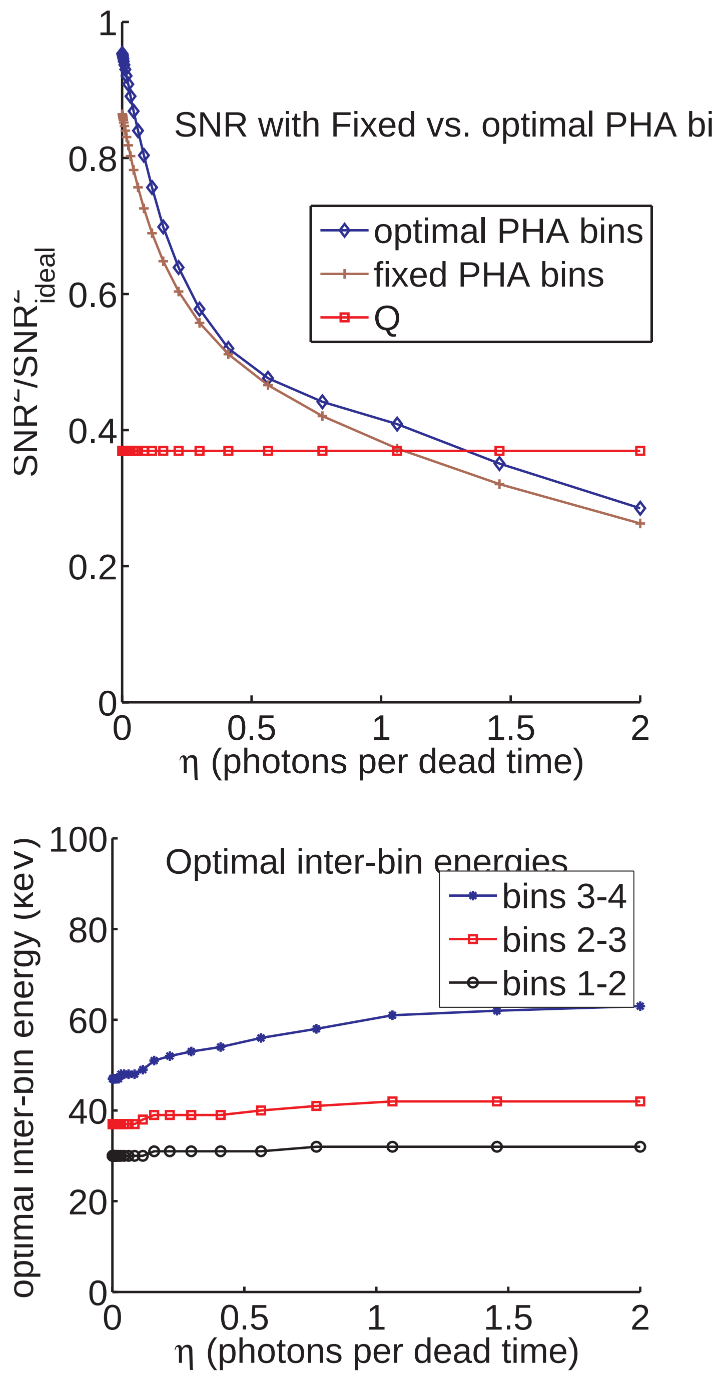

Figure 3: SNR vs counts per dead time with four bin PHA with inter-bin energies optimized for each value of η. The top panel shows the SNR while the bottom panel shows the optimal inter-bin energies.