ML Reconstruction of Random Telegraph Signal Pixels

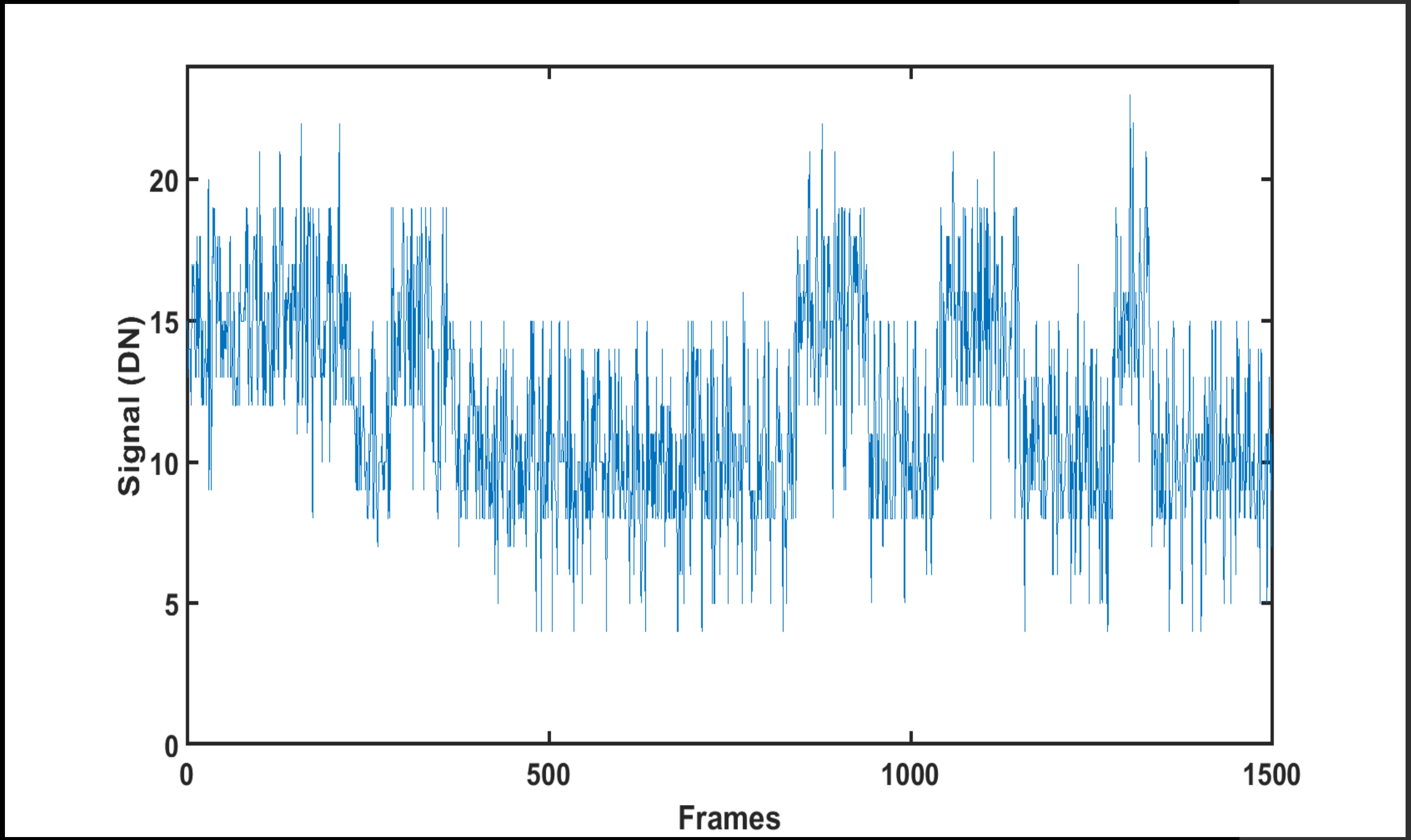
Ben Hendrickson

Dept. of Physics

Portland State University

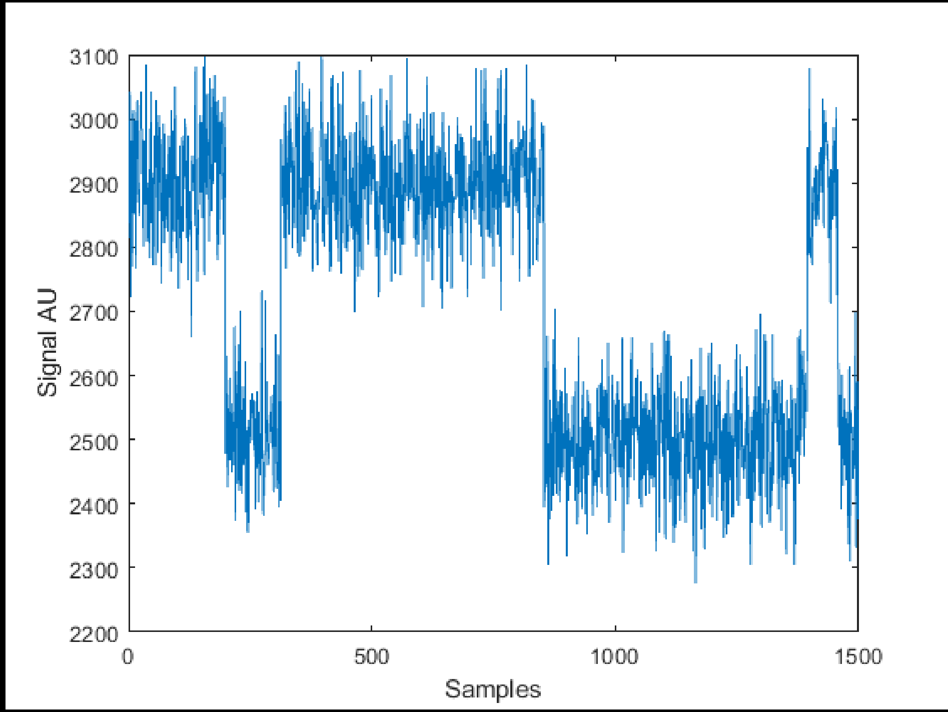
RTS Noise - Overview

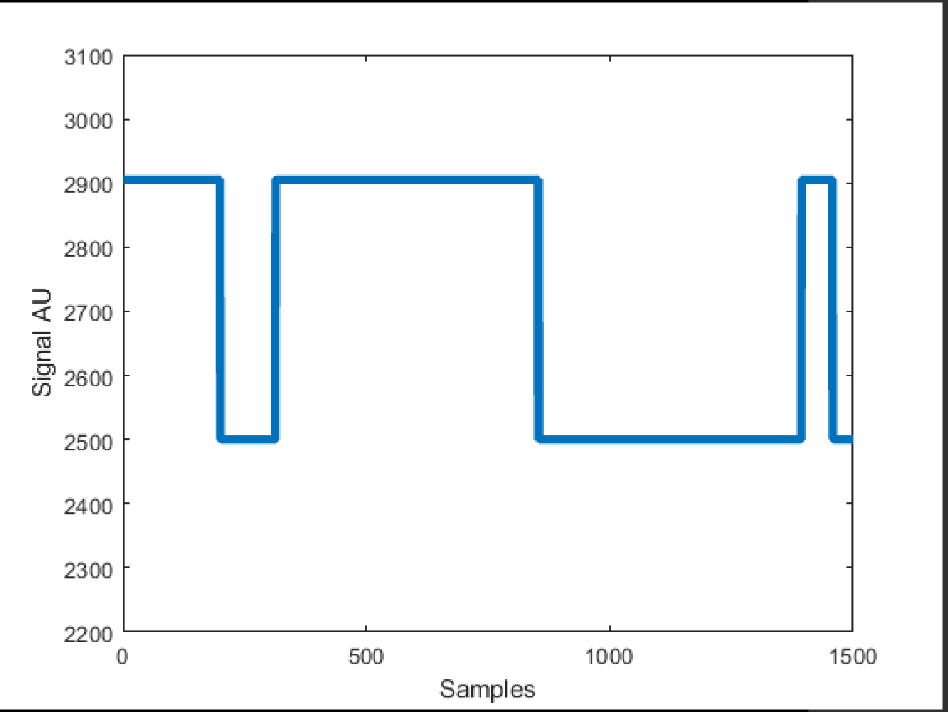
- Seen in a the output of a single pixel in an image sensor
- Defined by discrete changes in signal level (blinking pixels)
- Similar to ‘snow’ on old TV sets
- I want to see understand these jumps



Signal Reconstruction

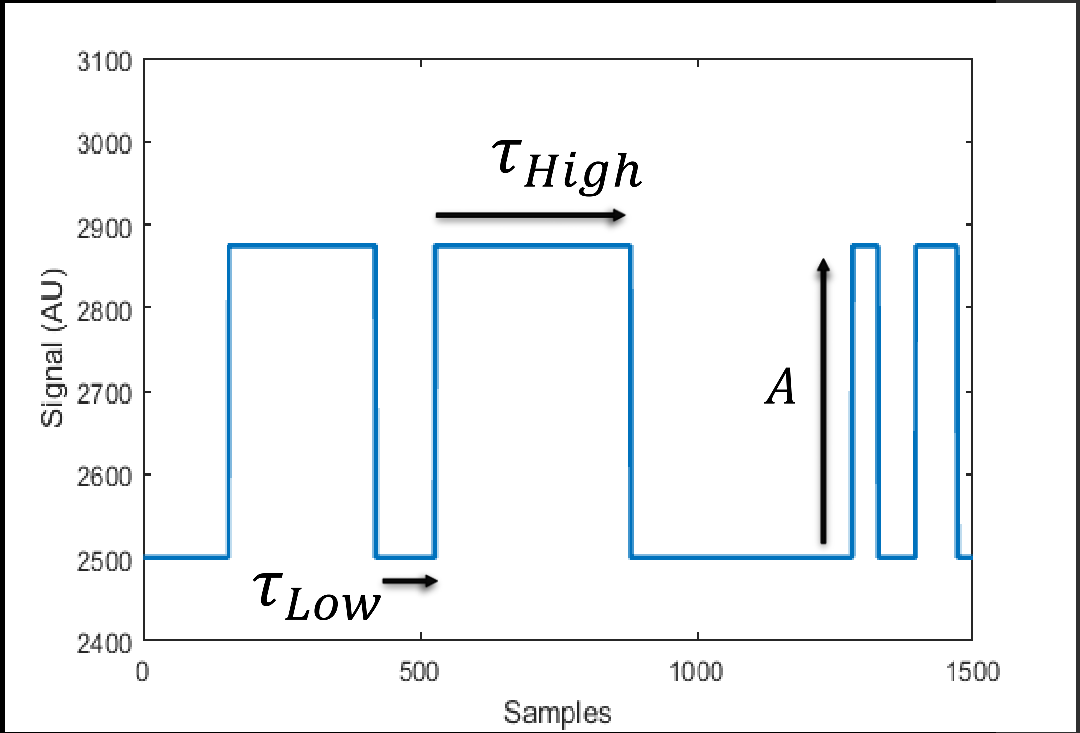
- What does perfect approximation look like?
- Zero white noise contribution
- Perfect RTS representation in shape and scale





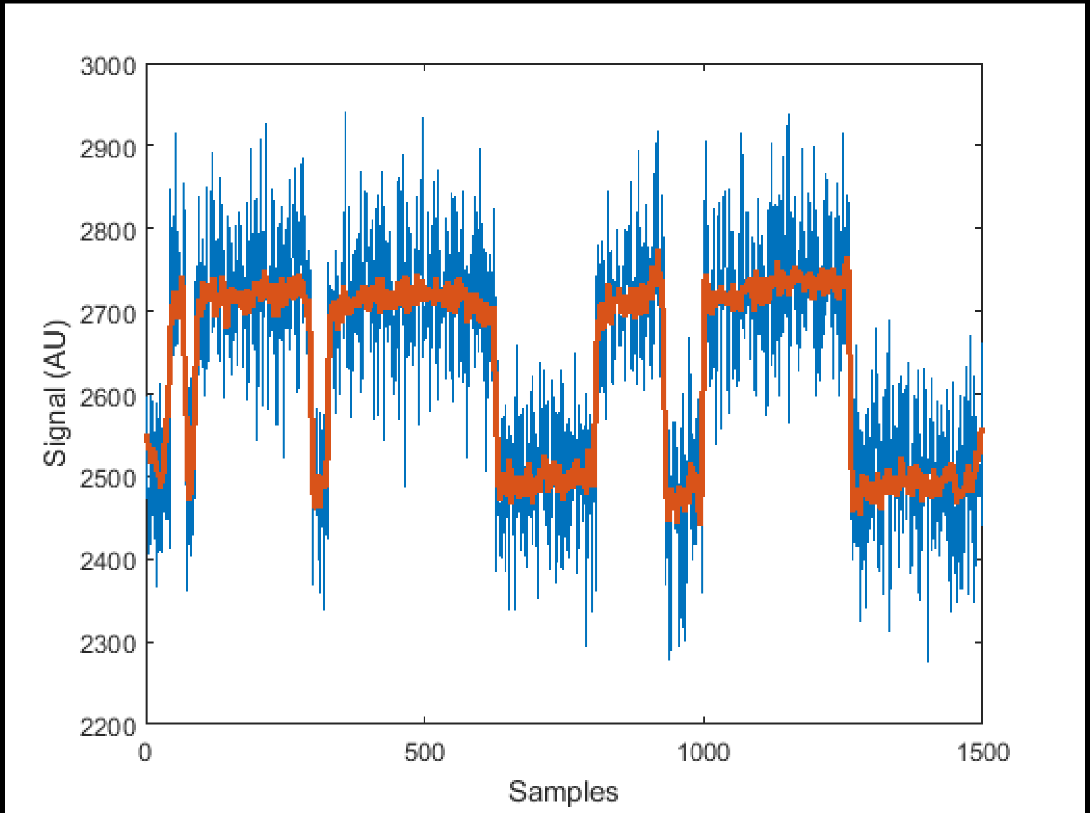
Signal Reconstruction

- Primary RTS characteristics
  - State lifetime (time constant)
  - RTS amplitude
- Challenges:
  - No well defined limits in τ and A for RTS signals
  - Small τ's and A's make RTS transitions difficult to distinguish from normal Gaussian noise
  - Maybe 10,000 interesting pixels in a 10M pixel camera



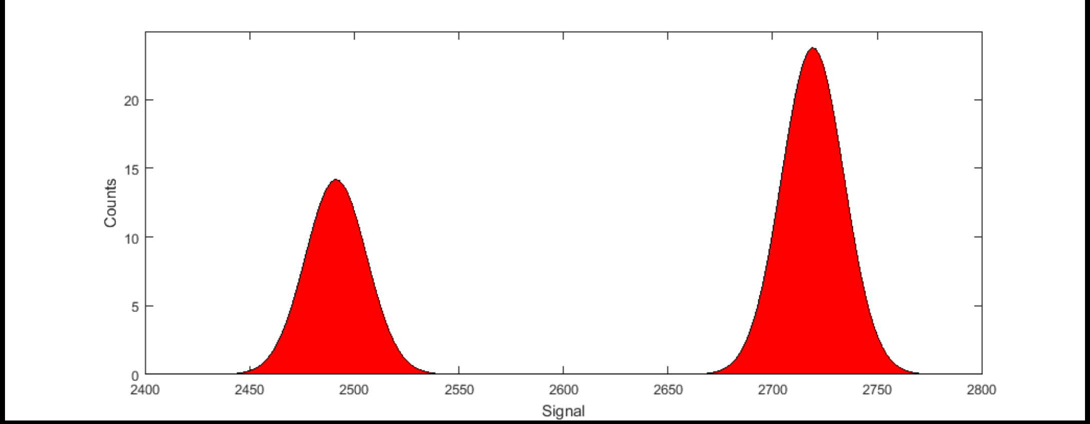
Machine Learning Method

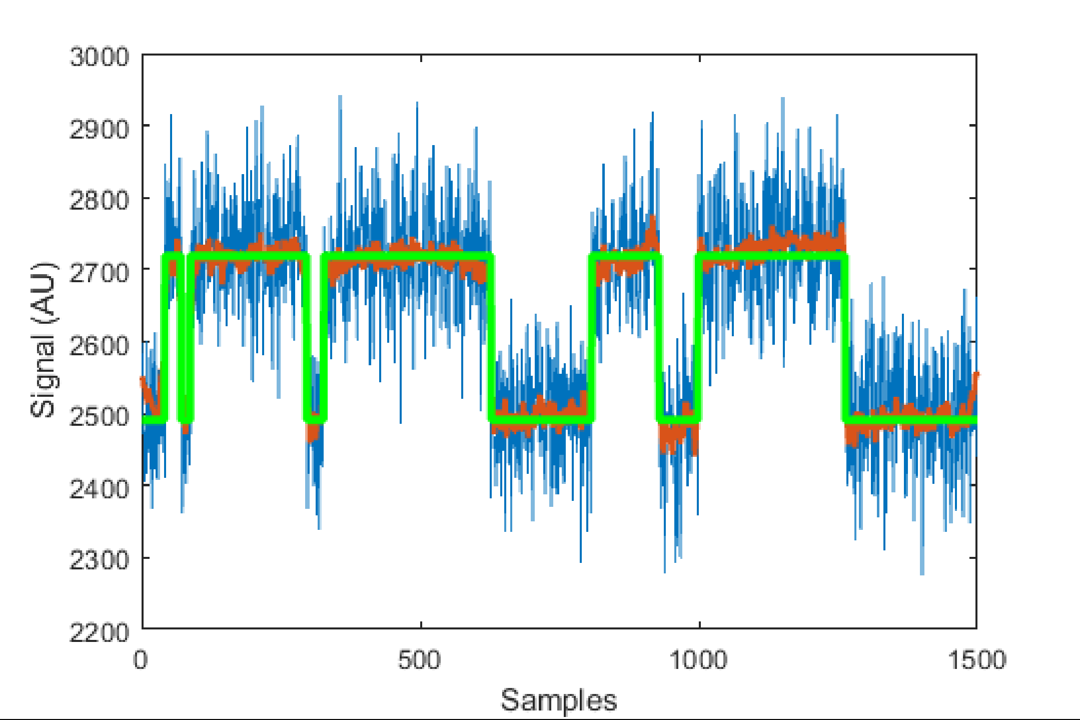
- RTS detection is performed by a classification model
  - Similar to image classification
  - Takes a signal and returns a zero for RTS or one for non-RTS
- WN reduction is performed by an autoencoder
  - Trained by creating gaps in signals, and 'learning' the best way to fill in those gaps
  - Takes a noisy signal and returns a clean signal



Machine Learning Method

- To finish approximation with the ML method, a histogram is created from the autoencoder output
- The result is fitted as the sum of two Gaussian distributions
- The peaks are taken as the RTS signal levels, and the signal is reconstructed where each sample from the autoencoder snaps to its closest value from the histogram fit

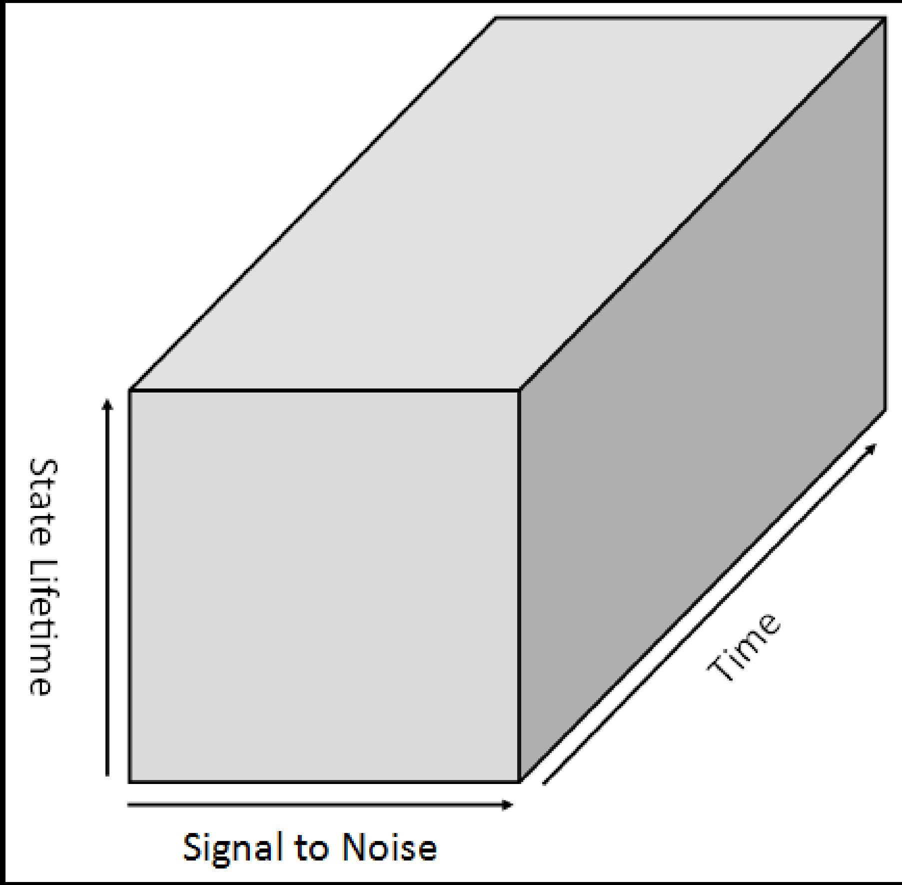


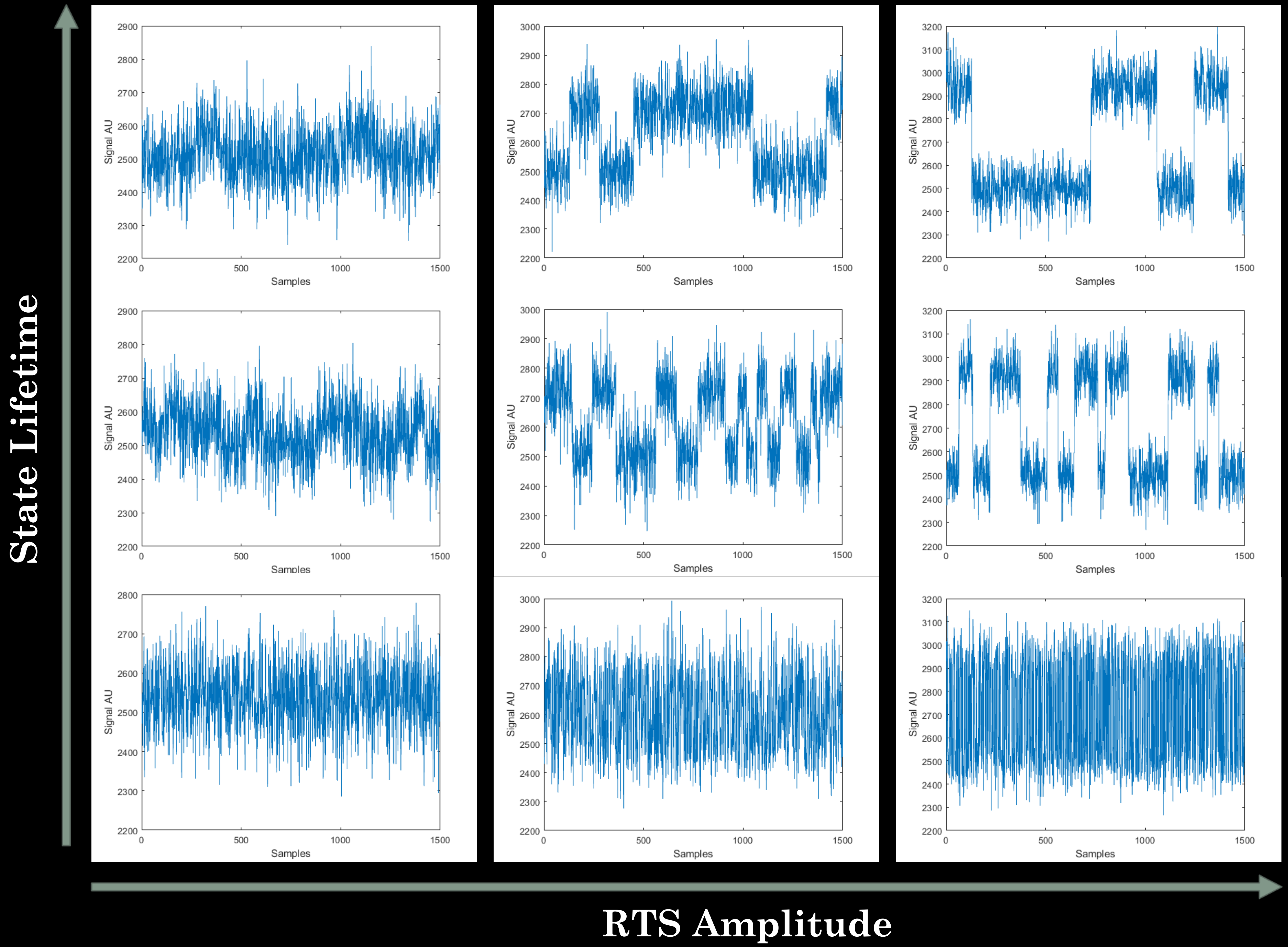


Testing Procedure

- Big simulated data block (300x300x1500)
- One dimension of the block spans a number of RTS amplitudes
- The other dimension spans a number of state lifetimes
- Each reconstruction is scored by a correlation coefficient (1 = good, 0 = bad)
- Tested for false positives on a block of purely Gaussian signals

$$C_{xy} = \frac{\Sigma(x-\overline{x})(y-\overline{y})}{\sqrt{\Sigma(x-\overline{x})^2}\sqrt{\Sigma(y-\overline{y})^2}}$$

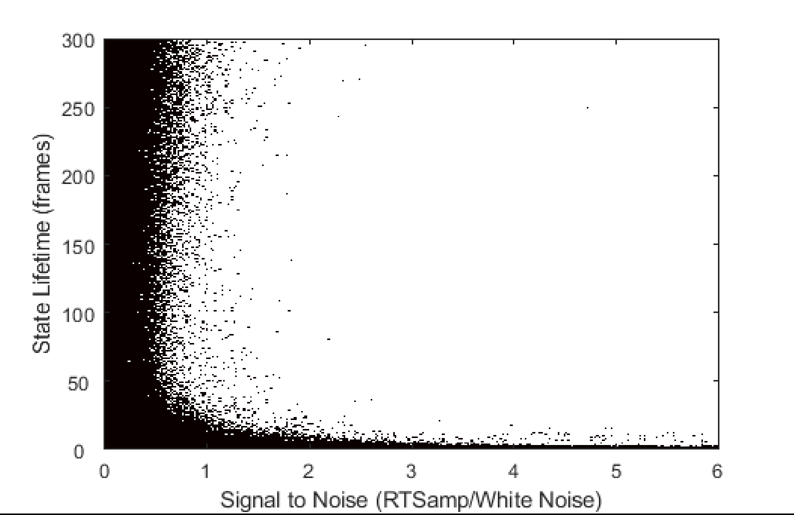



State Lifetime
RTS Amplitude
Signal AU
Samples

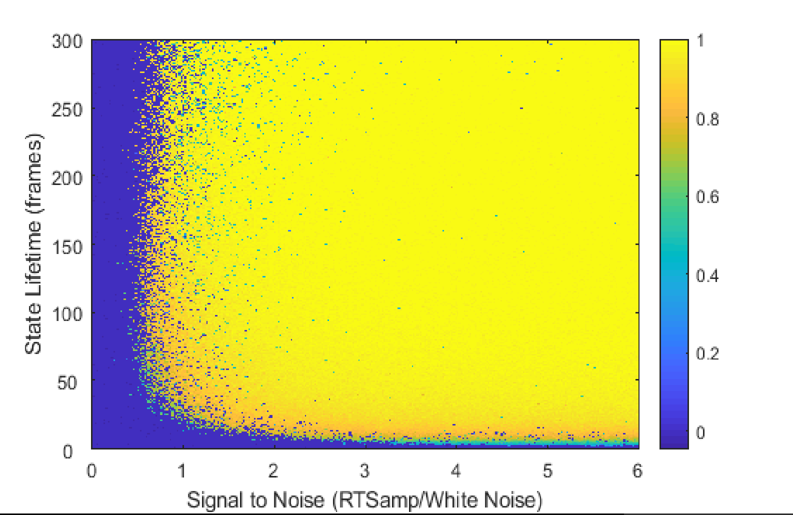
Results – Machine Learning Method

- Reliably works with $SNR > \sim 1$ and $\tau > \sim 25$ frames
- 85.4% RTS detection rate
- Mean C_{xy} of for detected signals: 0.9564
- Zero(!) non-RTS false positives

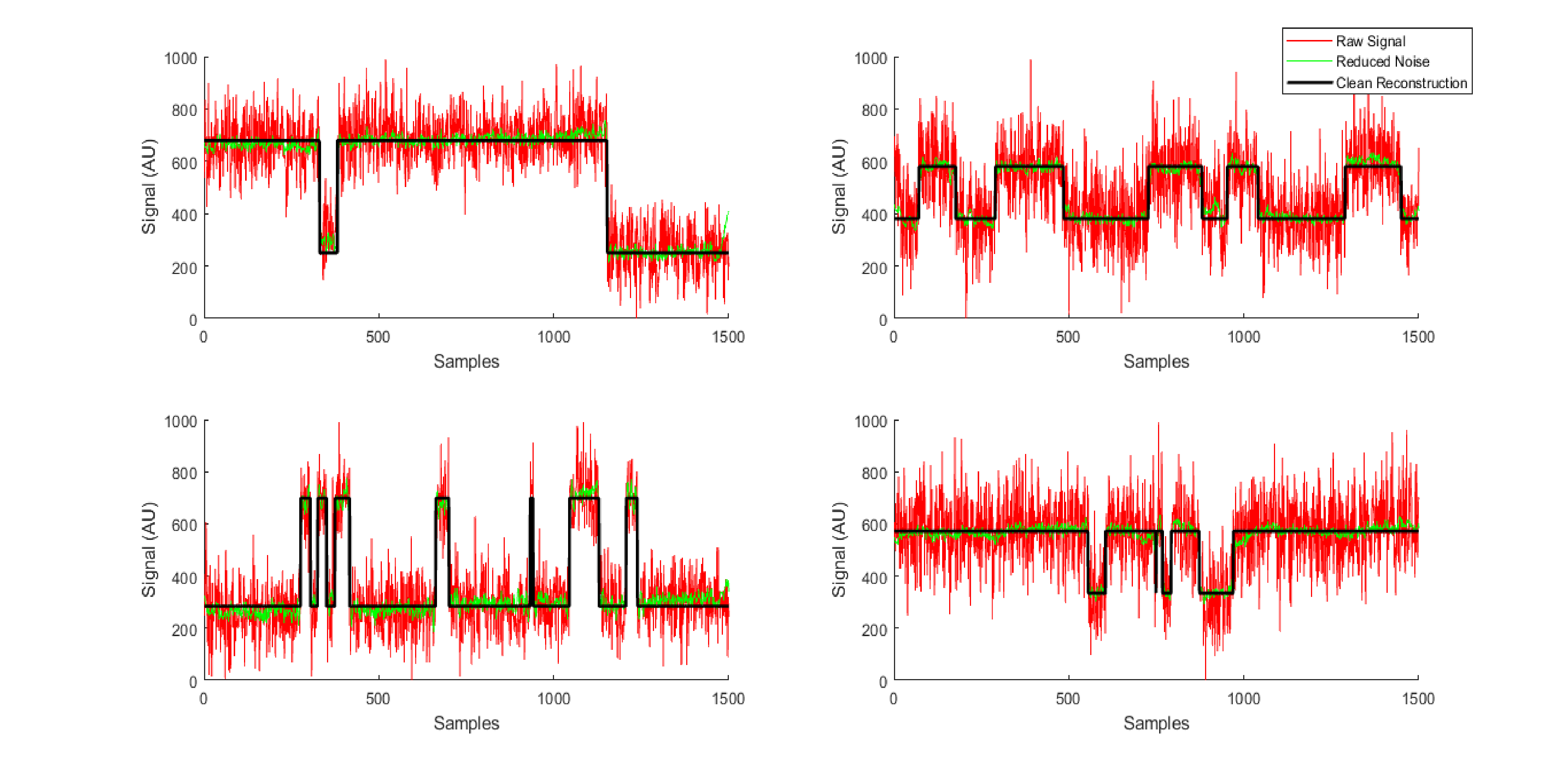
Detection



Correlation



Results – Machine Learning Method



Thanks for listening

- Questions?

Results – Correlation Comparison

- Direct comparison of the three methods
- All three perform well for signals $SNR > 2$ and $\tau > 50$ frames
- ML method performs reliably at half of those limits