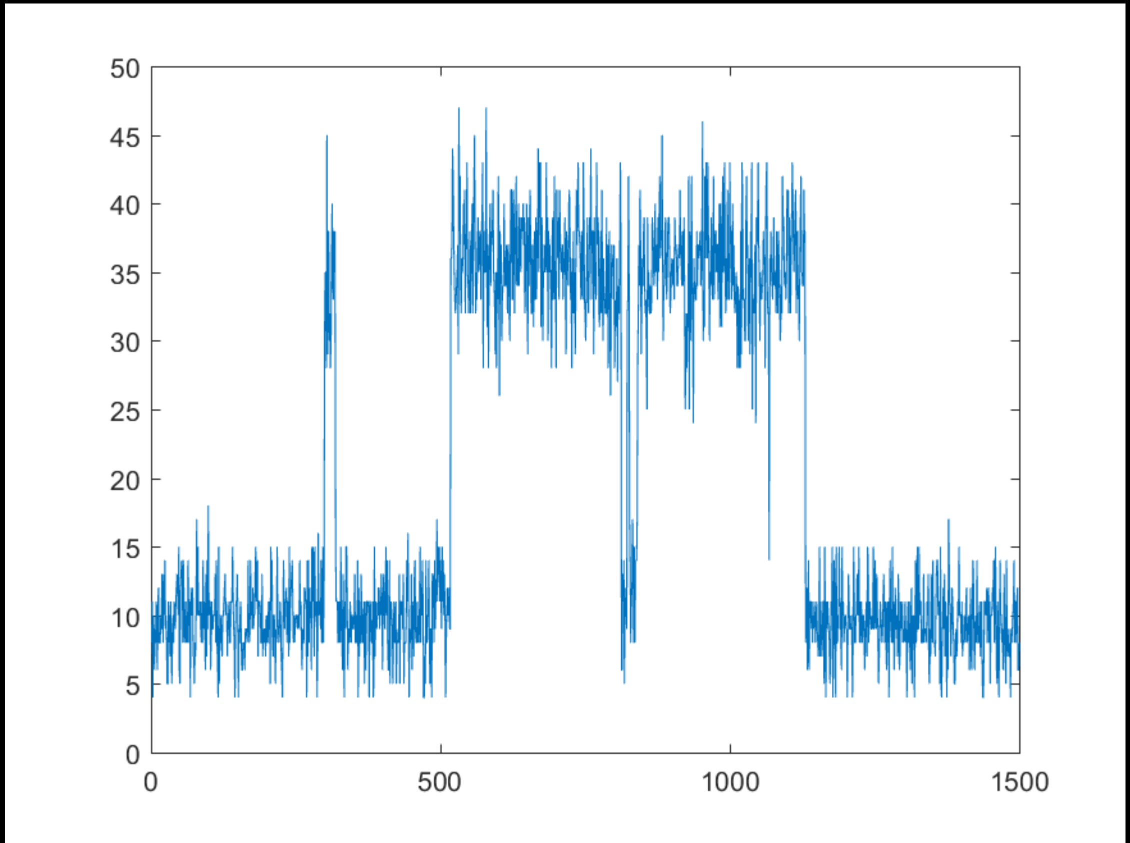
RANDOM TELEGRAPH SIGNAL: NON-LINEAR DYNAMICS AND NON-LINEAR ANALYSIS

BEN HENDRICKSON – PORTLAND STATE UNIVERSITY



WHY RANDOM TELEGRAPH SIGNAL?

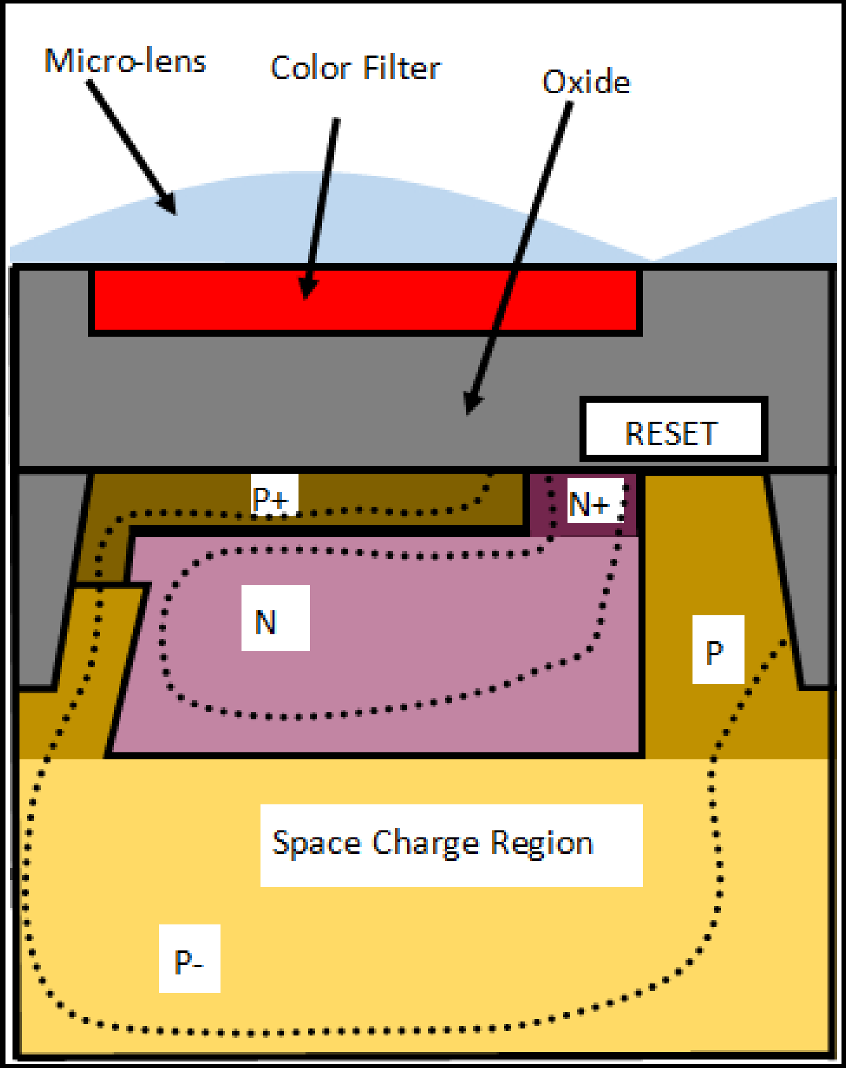
- Becoming more significant as image sensor fabrication becomes more sophisticated at reducing noise sources
- Leads to ‘blinking pixels’ similar to the ‘snow’ seen on old TV sets
- Observed in a variety of physical processes including ion transport in biological membranes and single molecule chemical reactions
- Physically tiny change that leads to macroscopic consequences
- Balls don’t jump over hills!

INTRODUCTION

- Image sensors basic structures
- Introduction to random telegraph signal noise and its key parameters
- Convolutional Filtering reconstruction method
- Discrete wavelet transform and signal reconstruction algorithm
- Machine learning modeling and reconstruction algorithm
- Experimental Results
- Conclusion

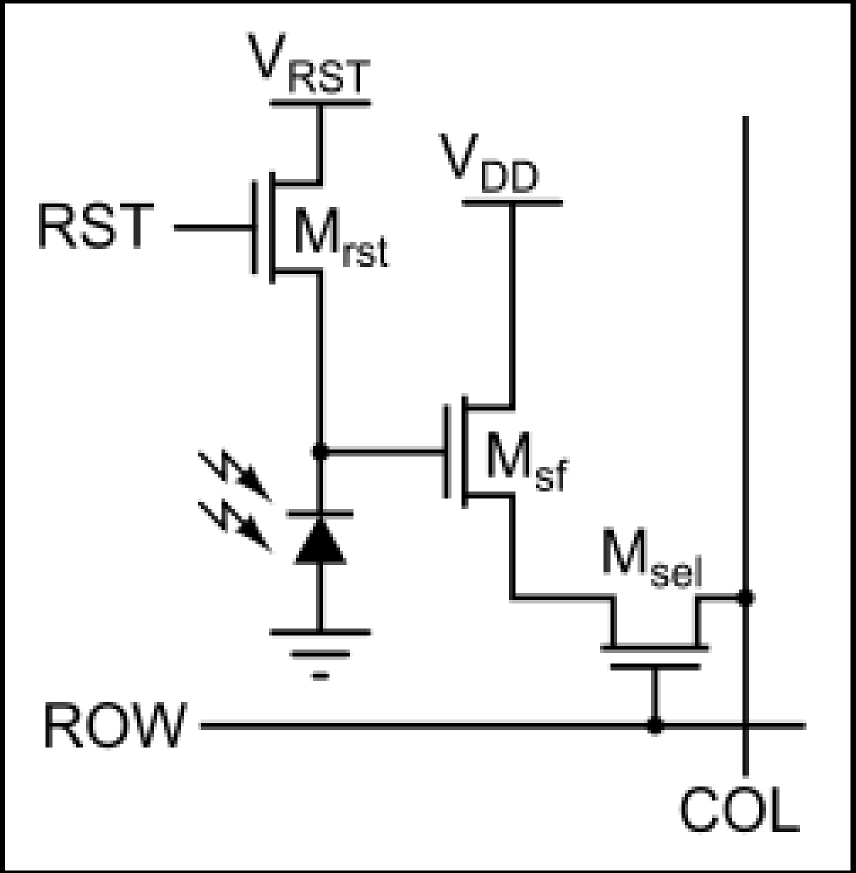
BASIC CMOS IMAGE SENSOR STRUCTURE (3T)

- Microlens and filter
- P+ pinning layer
- PN Junction
- P- layer
- Oxide & Interface



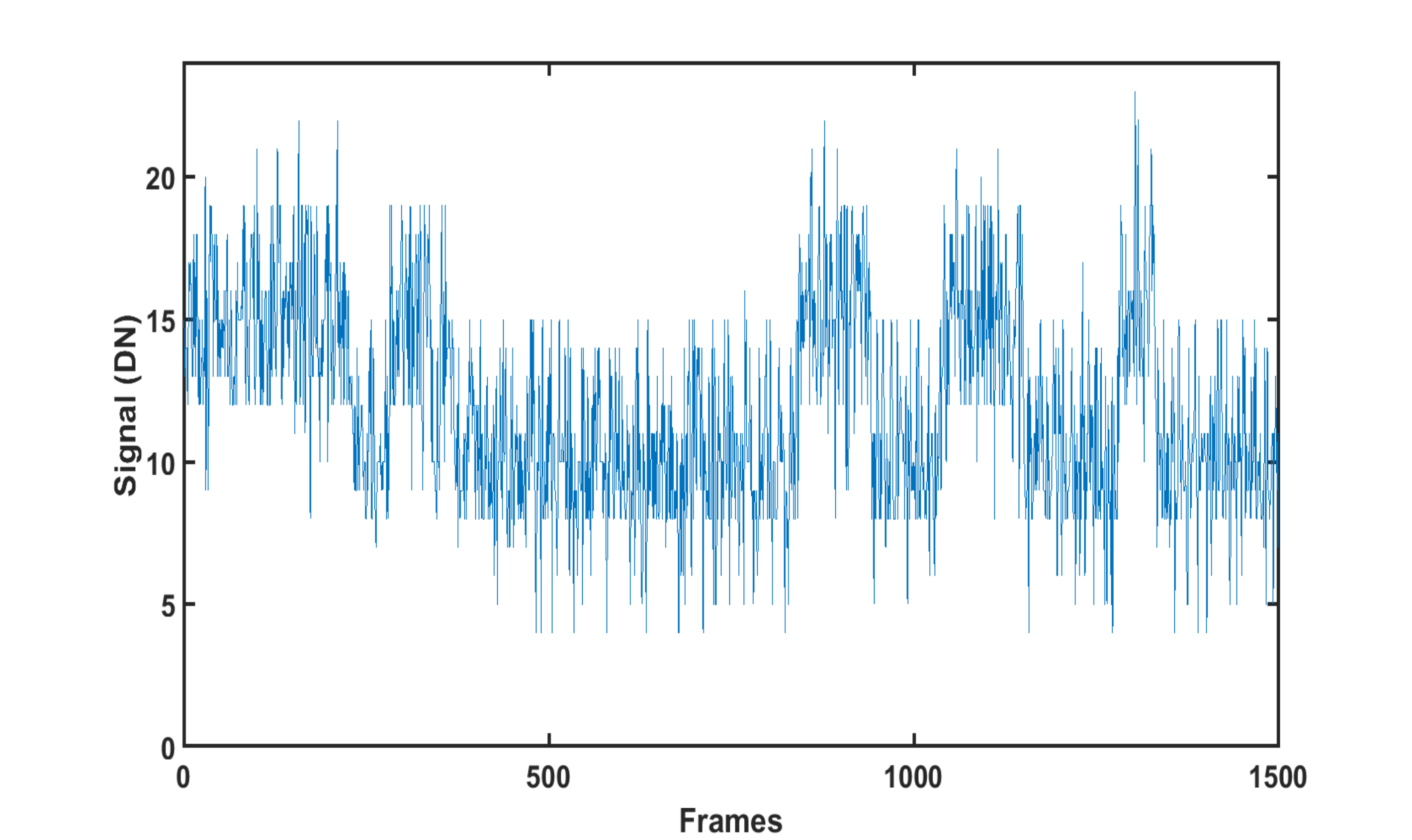
BASIC CMOS IMAGE SENSOR OPERATION (3T)

- Reset pulse
- Photodiode collection
- Source Follower
- Row Select
- Column Bus



RTS NOISE - OVERVIEW

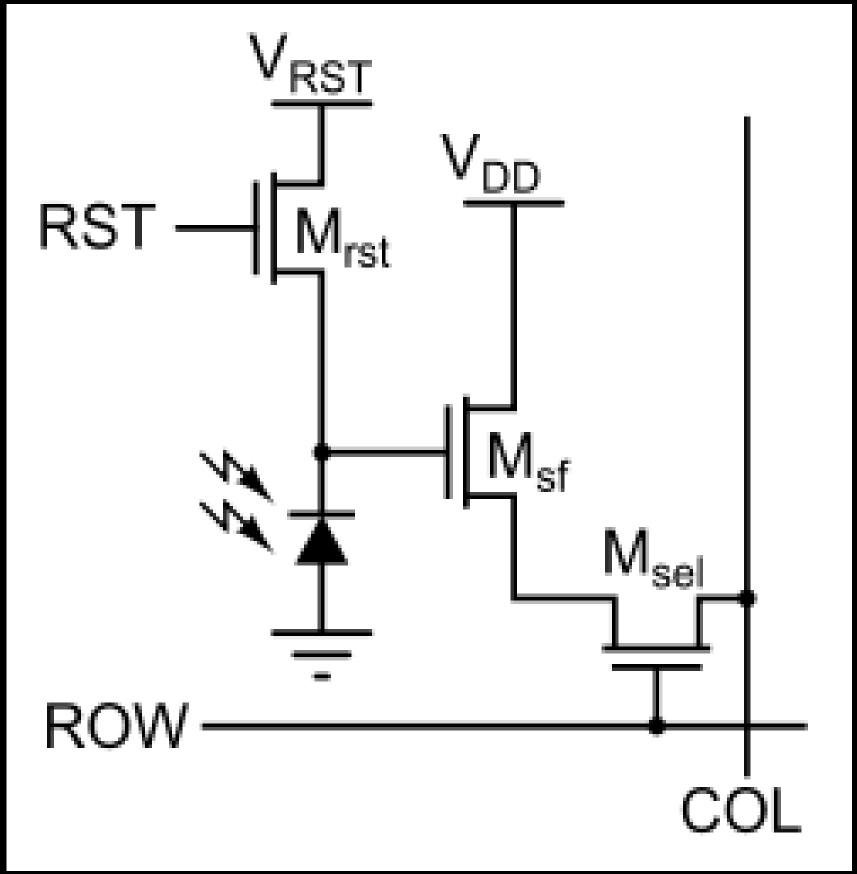
- Defined by discrete changes in signal level
- Stochastic process
- Characterized by similar time constants



Dark signal output from a single pixel, 1500 samples

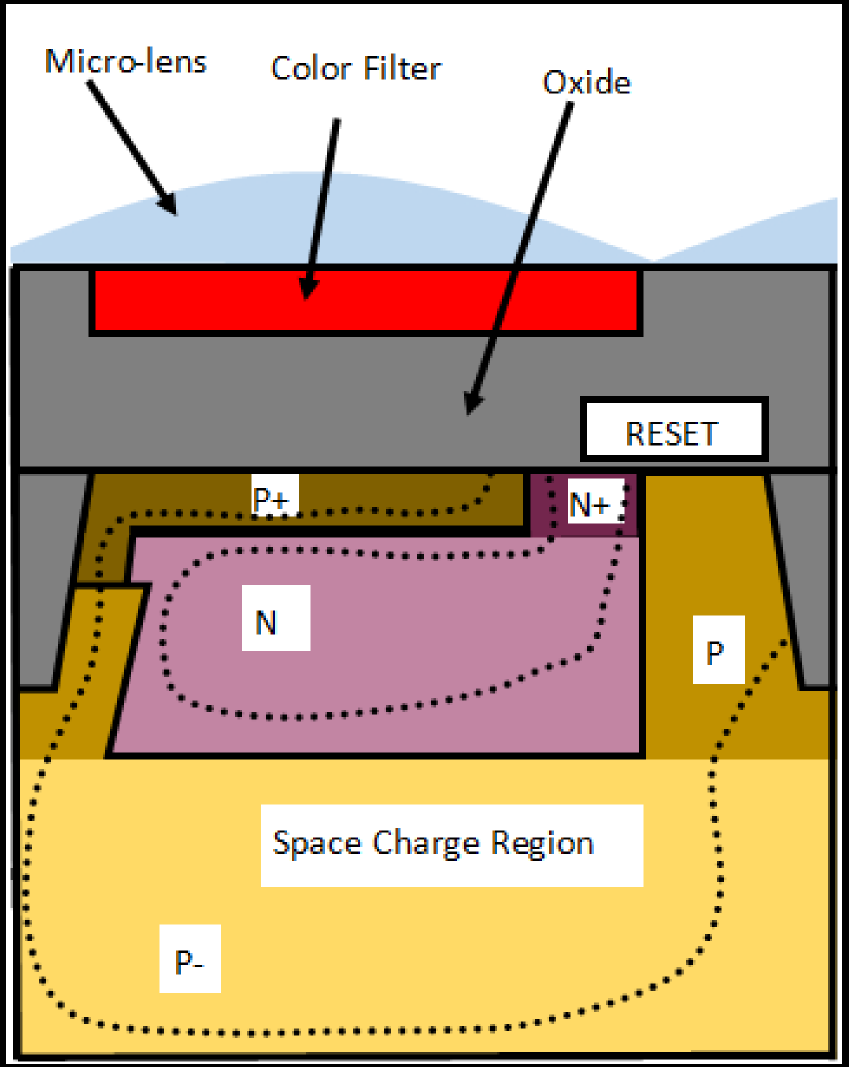
PHYSICS OF RTS NOISE

- Source Follower RTS Noise
  - Trapping/De-trapping of charges changes V_{GS} of the SF-transistor
  - Capture state → lowered mobility → lowered conductivity: $\sigma = \mu n q$
  - Time constants are governed by Shockley Read Hall statistics
    - Capture State → $\frac{1}{\tau_c} = \sigma_t \bar{v} n$
    - Emission State → $\frac{1}{\tau_e} = \sigma_t \bar{v} N_c \exp\left(\frac{E_t}{kT}\right)$



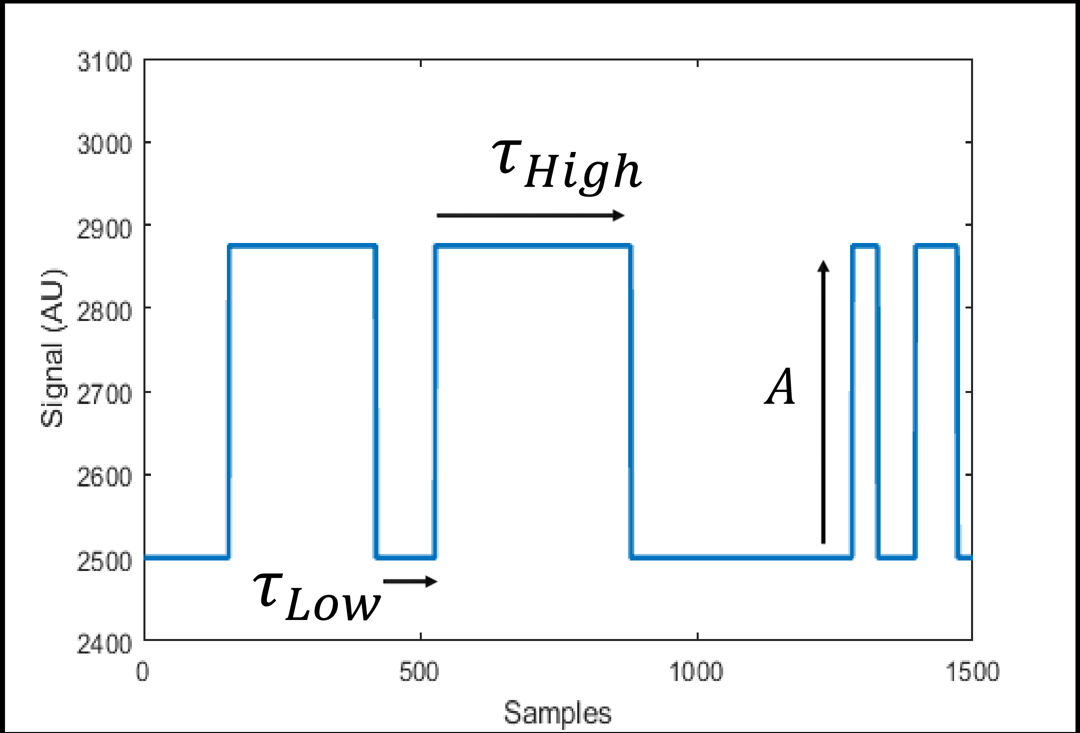
PHYSICS OF RTS NOISE (CONT.)

- Dark Current RTS Noise
  - Origin – metastable Shockley-Read-Hall generation/recombination centers
  - Identified by t_{int} dependence, large amplitudes, and very large time constants
  - Radiation damage effect
    - Protons typically create DC-RTS centers in bulk
    - X-rays/γ-rays typically create DC-RTS centers on Si-SiO2 interface



KEY PARAMETERS

- Primary RTS characteristics
  - State lifetime (time constant)
  - RTS amplitude
- 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



EXPERIMENTAL PARAMETERS

- COTS Omnivision OV5647
- Raw frames taken using a Raspberry Pi 3
- Five sensors irradiated with a continuum of high energy γ and x-rays (peak - $2MeV$)
- Six second integration time
- 1500 frames taken at 0.05 frames/s
  - ~ 8.3 hours total measurement time
- Frames taken in dark at 32°C

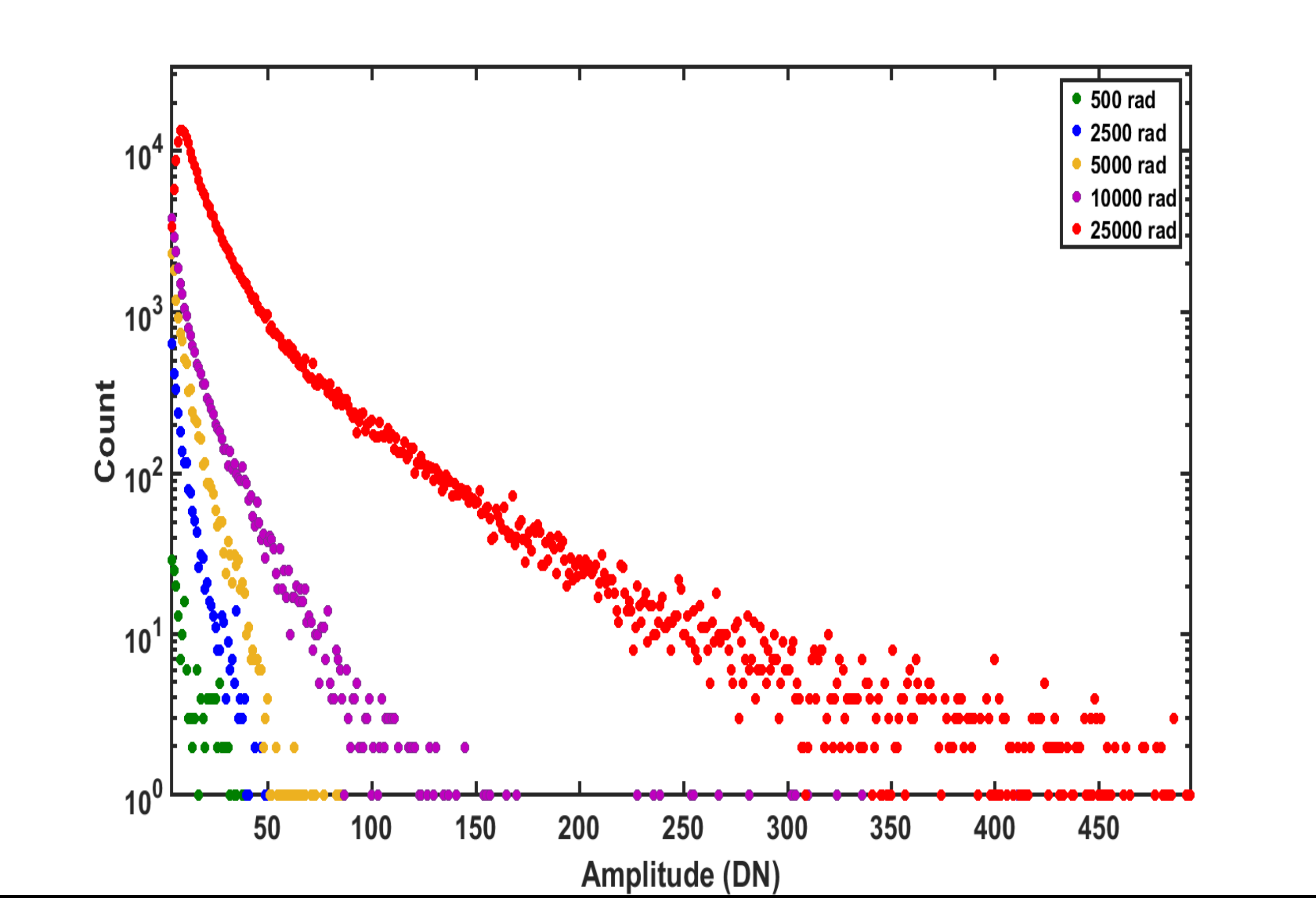
Absorbed Ionizing Dose (rad – Si)
500
2,500
5,000
10,000
25,000

EXPERIMENTAL GOALS

- Each pixel is treated as its own separate device
- 10 megapixel camera → 10 million devices
- Amplitudes and lifetimes are collected and sorted
- Patterns begin to emerge

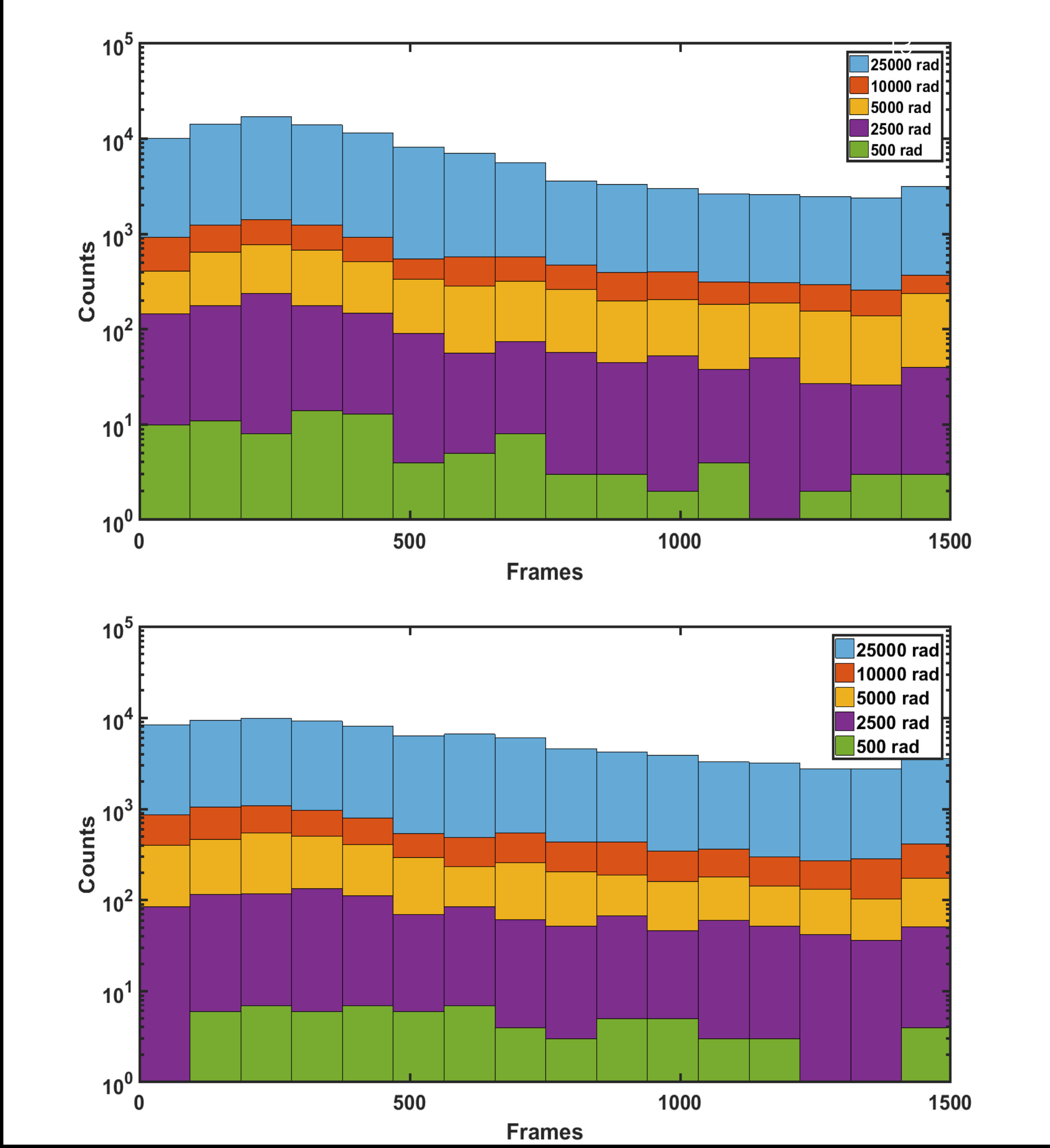
RESULTS - AMPLITUDES

- Amplitudes up to 350 e^{-}/s observed
- Similar shape of curves indicates that higher doses increase the likelihood of creating an RTS center, but the amplitude probability for a center is set
- No correlation seen between RTS amplitude and time constants



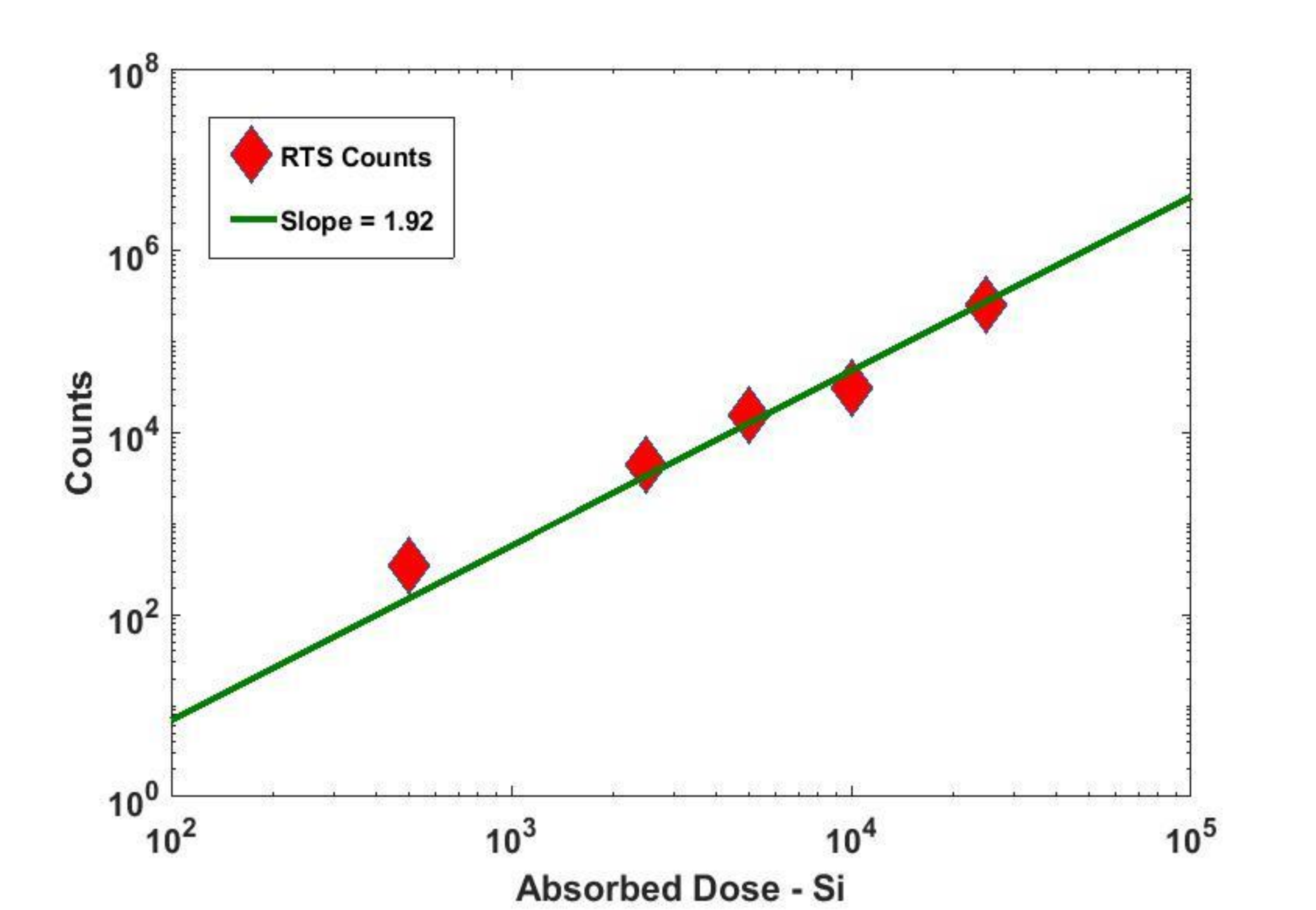
RESULTS – TIME CONSTANTS

- Time constants are calculated by averaging the time spent in the high or low states
- Both high and low states display an exponential distribution
- The low state time constant distribution is slightly flatter than the high state, indicating that the low state is the more stable of the two



RESULTS – SECOND-ORDER DEFECT GENERATION

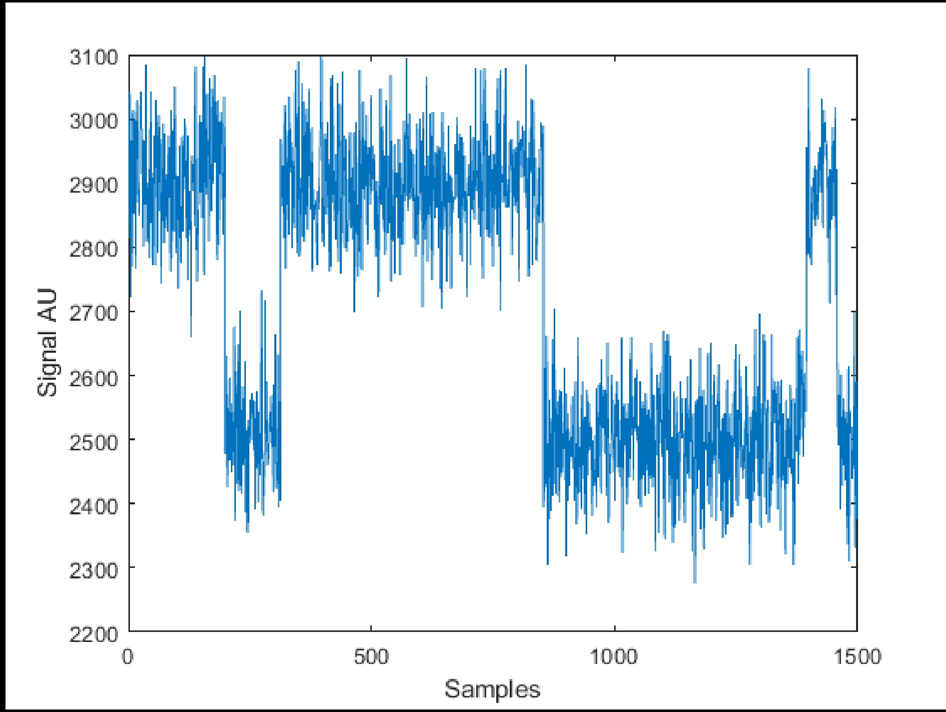
- The number of RTS centers increases ~quadratically with absorbed dose
- This indicates that the particular defect responsible for this RTS noise is of second-order
- There is a precedent for this kind of defect, specifically the double vacancy oxygen (V_2O) complex[1]

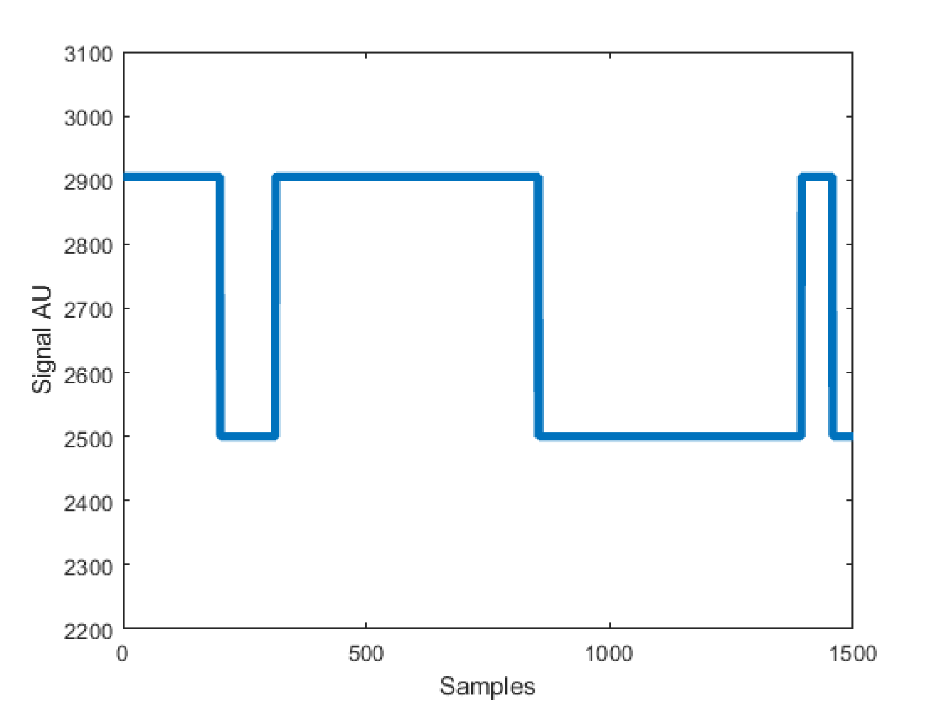


[1] I. Pintilie, E. Fretwurst, G. Lindström, J. Stahl, "Second-order generation of point defects in gamma-irradiated float-zone silicon, an explanation for 'type inversion'," *Applied Physics Letters*, vol. 82, pp. 2169-2171, Mar. 2003.

SIGNAL RECONSTRUCTION

- Makes collection of key parameters trivial
- Zero white noise contribution
- Perfect RTS representation in shape and scale





CONVOLUTION METHOD

- Code provided by V. Goiffon et. al. (2009)

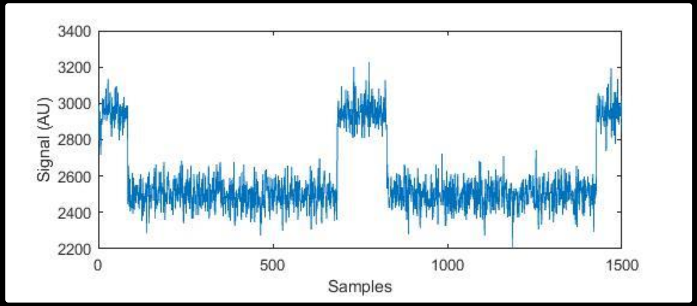
2132 IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 56, NO. 4, AUGUST 2009

Multilevel RTS in Proton Irradiated CMOS Image Sensors Manufactured in a Deep Submicron Technology

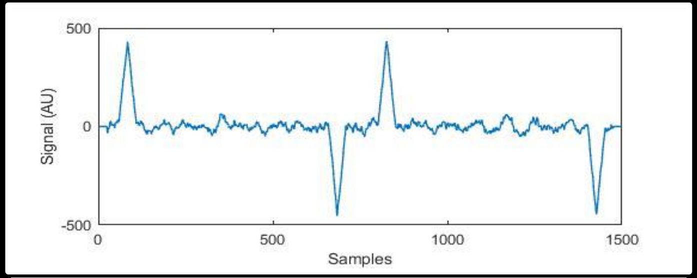
V. Goiffon, *Member, IEEE*, G. R. Hopkinson, *Member, IEEE*, P. Magnan, *Member, IEEE*, F. Bernard, G. Rolland, and O. Saint-Pé

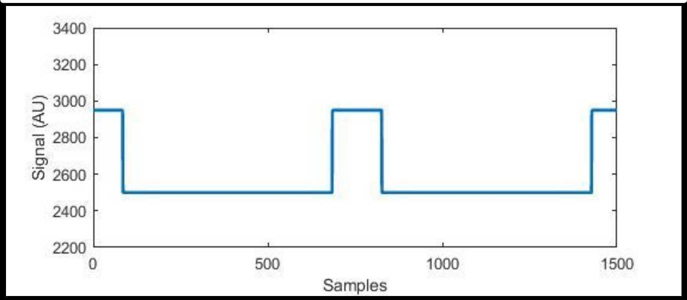
CONVOLUTION METHOD – CONT.

- Applies a step shaped filter to a signal
- Detects RTS if $A_{max} > \sigma_{sig}$
- Measures the mean value between spikes to estimate RTS signal levels
- Sorts RTS levels and approximates Gaussian noise-free RTS signal



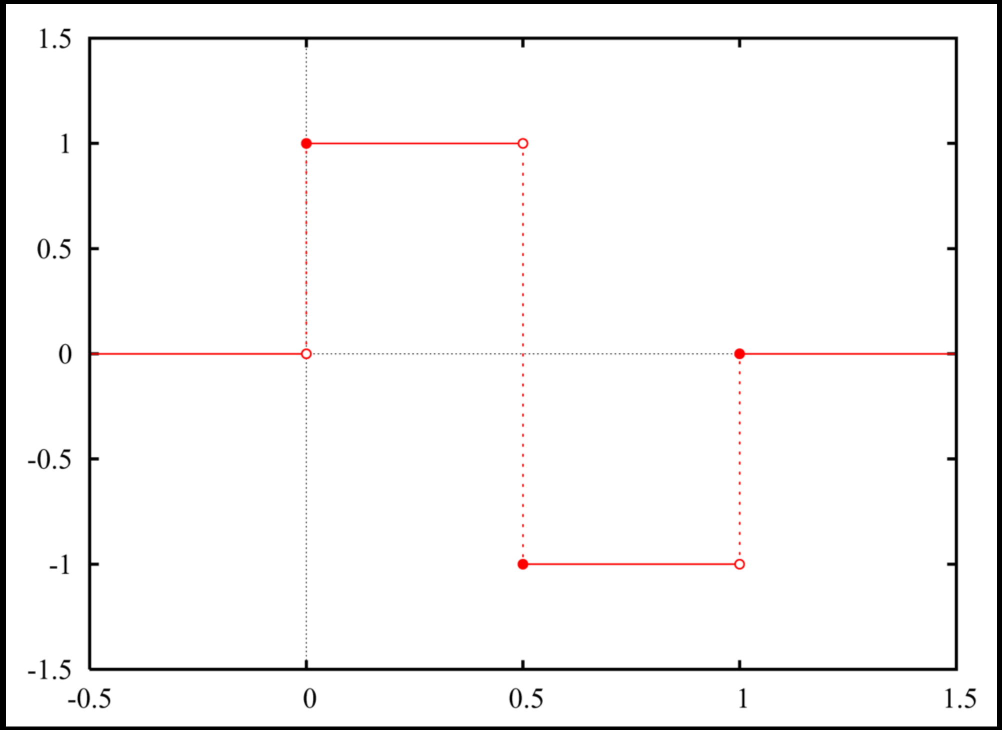
$$H(z) = \frac{2}{L}\left(-\sum_{i=0}^{\frac{L}{2}-1} z^{-i} + \sum_{i=\frac{L}{2}}^{L-1} z^{-i} \right)$$





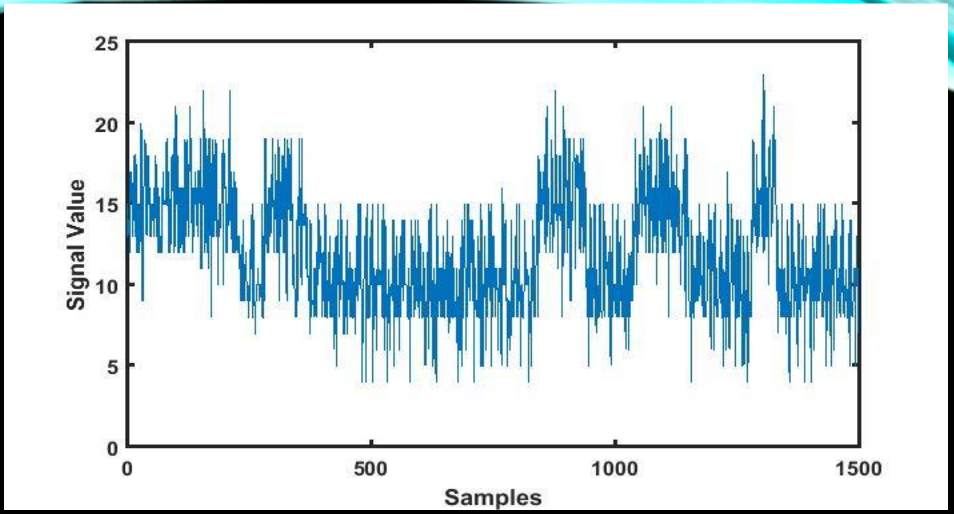
THE HAAR WAVELET & DISCRETE WAVELET TRANSFORM

- An orthonormal basis set developed by Alfred Haar in 1909
- Left largely in obscurity until DeBauchies pioneering work constructing and using wavelets for digital signal processing and analysis
- DWT Useful for edge detection applications

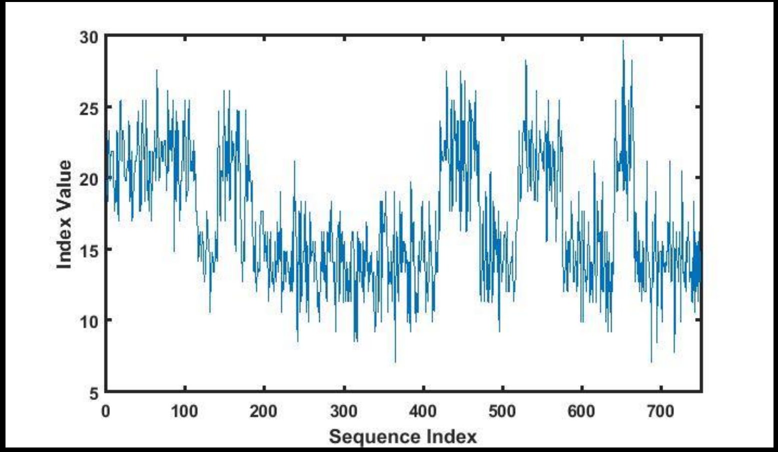


WAVELET ANALYSIS – DWT

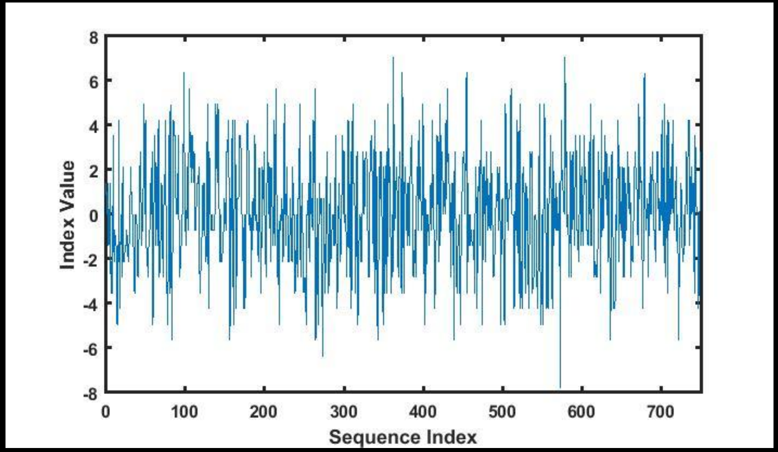
- The discrete wavelet transform (DWT) breaks **f** of length N into two 'daughter' sequences of length $N/2$



  - Trend Sequence Members
    - $a_m = \frac{f_{2m-1}+f_{2m}}{\sqrt{2}} \qquad 1 < m \leq N/2$



  - Details Sequence Members
    - $d_m = \frac{f_{2m-1}-f_{2m}}{\sqrt{2}} \qquad 1 < \ m \leq N/2$



J. S. Walker, A Primer on Wavelets and their Scientific Applications. Boca Raton [Fla.]: Chapman & Hall/CRC, 2nd ed., 2008.

HAAR WAVELET ANALYSIS – DWT (CONT.)

- The DWT is similar to a microscope because it is repeatable
  - The trend sequence is treated as the new ‘mother’ signal
  - Multiresolution analysis
- Each time a subsequent transform is performed the ‘daughter’ sequences are of half size
  - The new ‘daughter’ sequence represents twice as many values from the original signal

THE INVERSE TRANSFORM

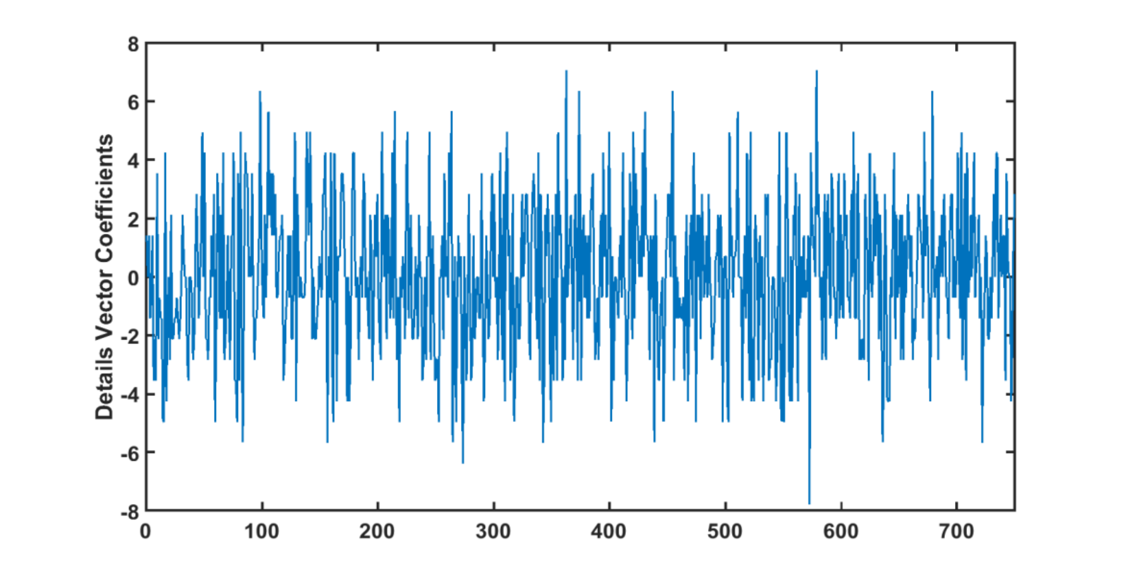
$$\mathbf{f} = \left(\frac{\mathrm{a}_1 + d_1}{\sqrt{2}}, \frac{\mathrm{a}_1 - d_1}{\sqrt{2}}, \dots \frac{\mathrm{a}_{\frac{N}{2}} + d_{\frac{N}{2}}}{\sqrt{2}}, \frac{\mathrm{a}_{\frac{N}{2}} - d_{\frac{N}{2}}}{\sqrt{2}}\right)$$

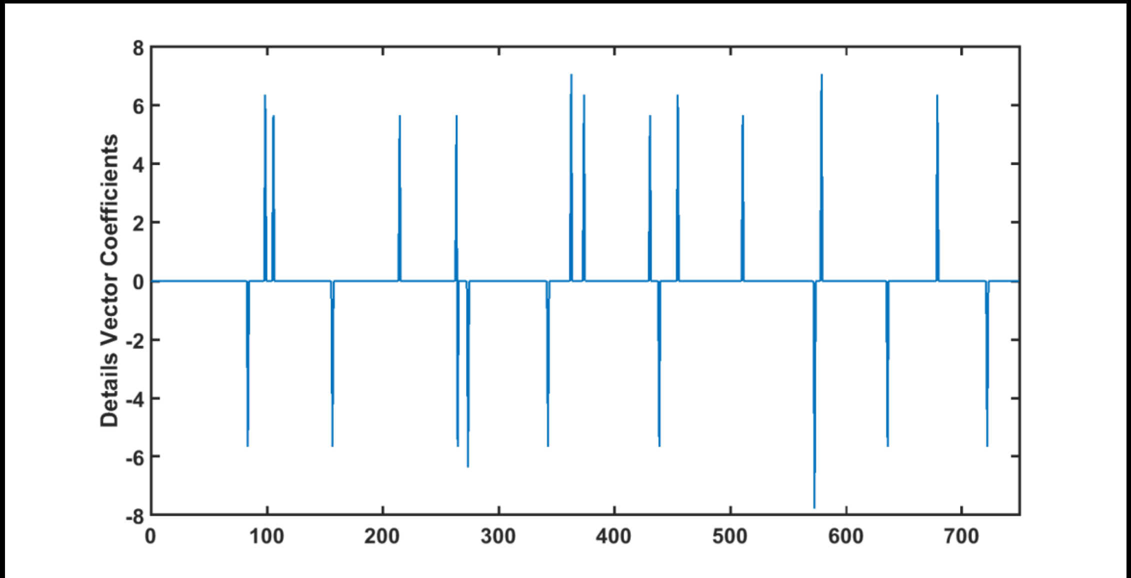
DWT DENOISING METHOD

- White noise is suppressed by thresholding the details sequence
  - Similar to a high-pass/low-pass filter
  - Based on magnitude rather than frequency
- The threshold is statistically derived

$$T = \hat{\sigma}\sqrt{2\log(n)}$$

- **T** is the universal threshold derived by Donoho and Johnstone†
- Values below the threshold are set to zero

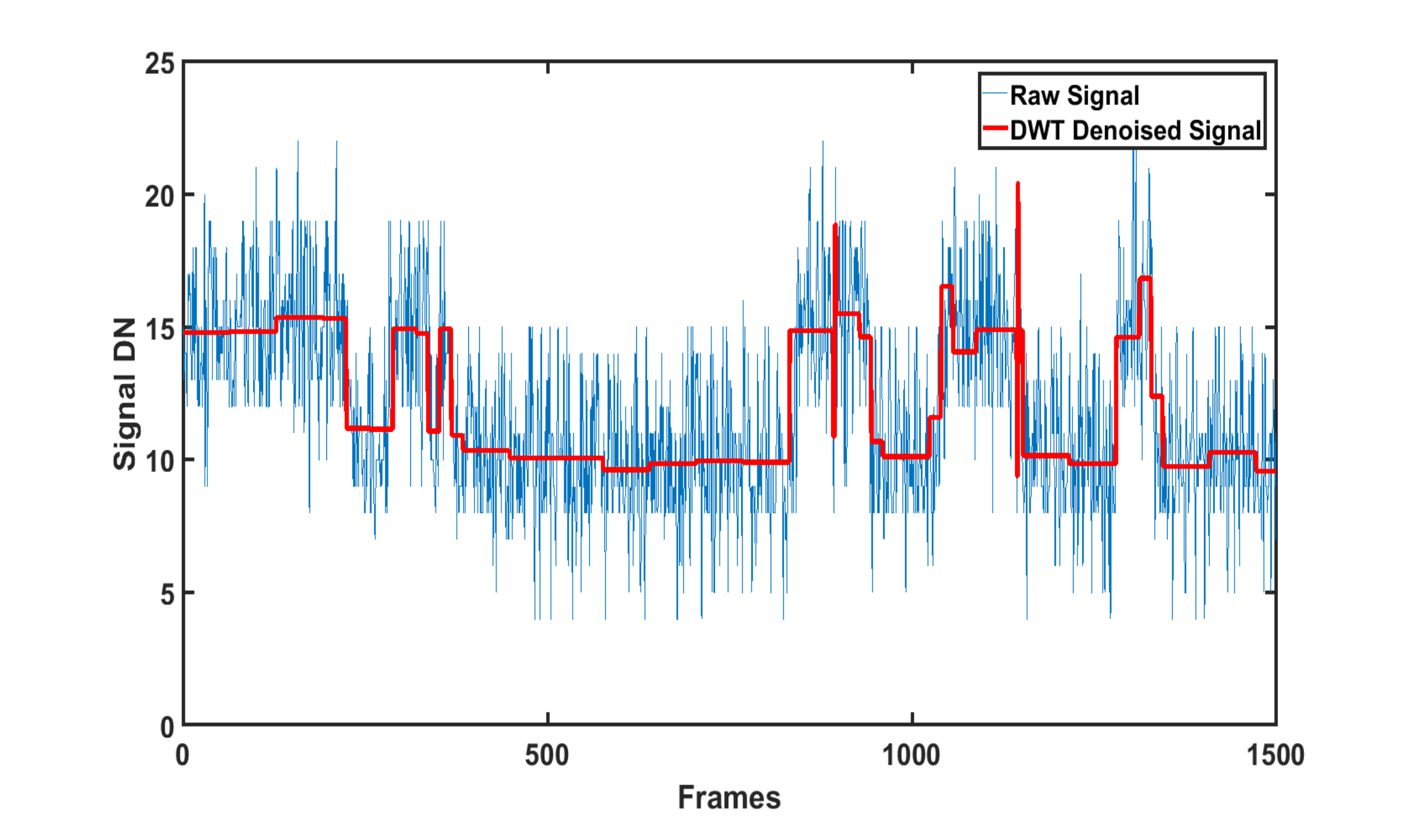




† G. P. Nason, "Choice of the threshold parameter in wavelet function estimation," Wavelets and statistics, vol. 103, pp. 261–280, 1995.

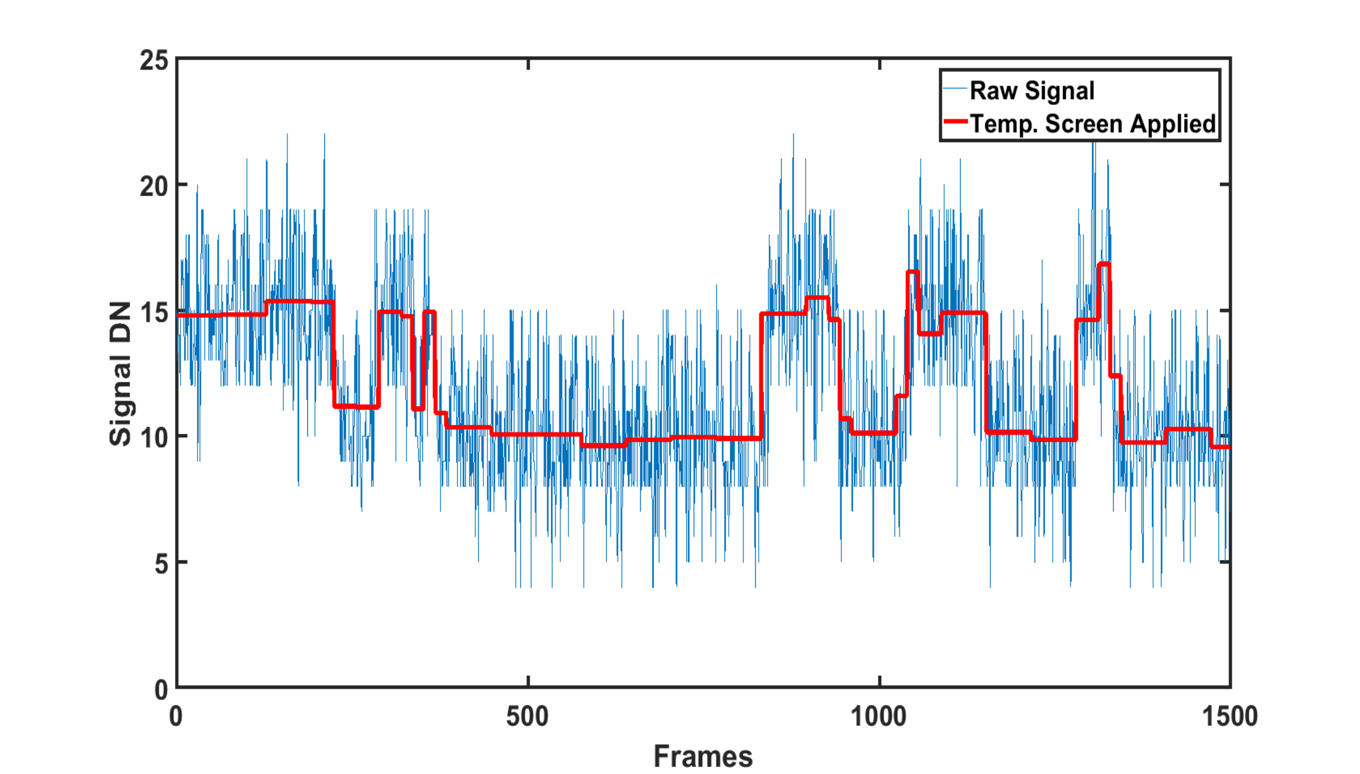
DWT DENOISING METHOD

- The signal is run through the DWT denoising method as described
- The white noise is greatly reduced, but a few transients remain



DWT DENOISING METHOD

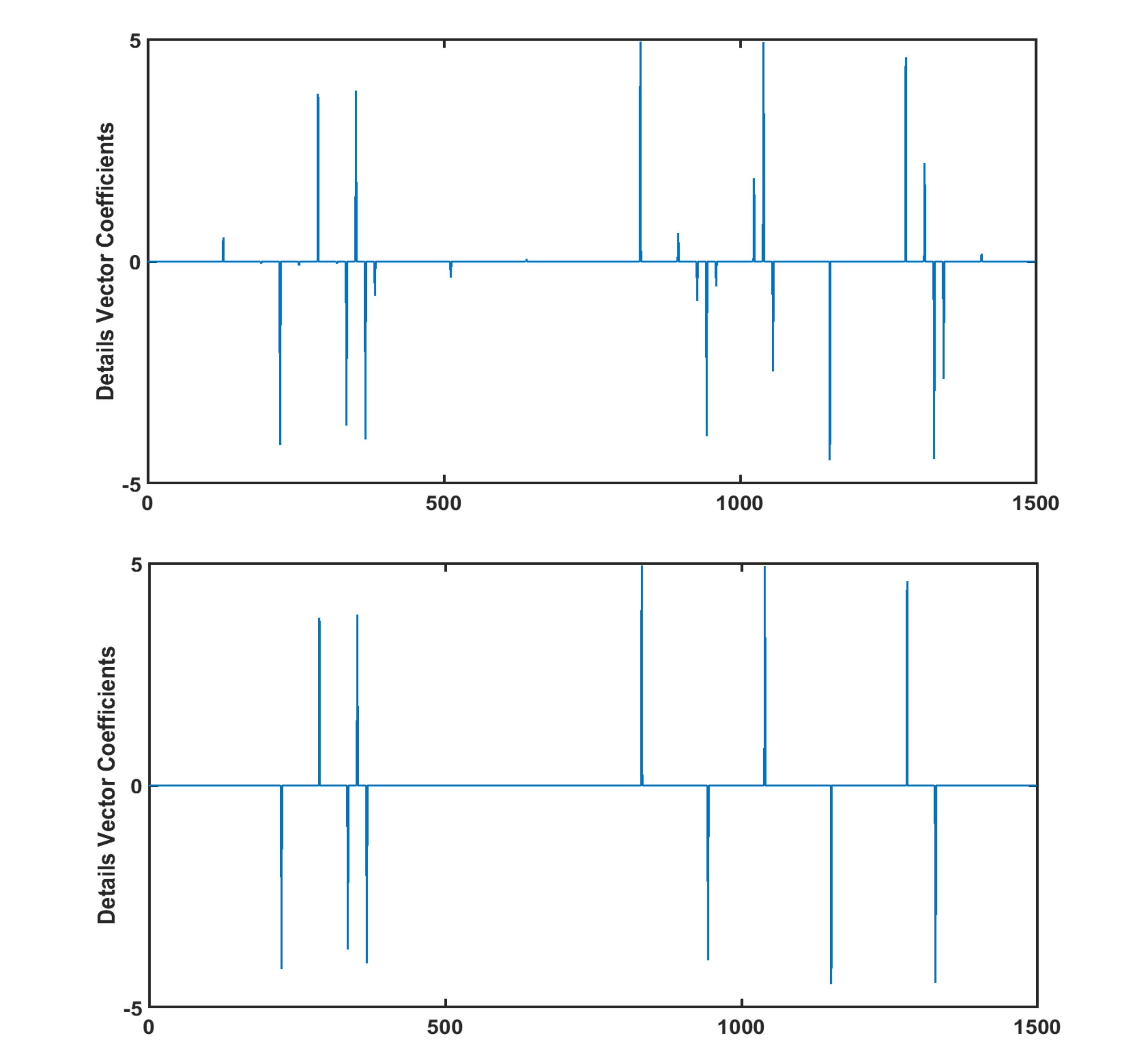
- To remove transients, a simple running comparison is implemented to verify the stability of a transition
- When a change in signal occurs at frame k, its value is compared to the next l frames where $l = 10$
- If the value is unchanged the transition is considered stable and left alone
- If the value changes is considered a transient, and is changed to the value at frame $k - 1$



DWT DENOISING METHOD

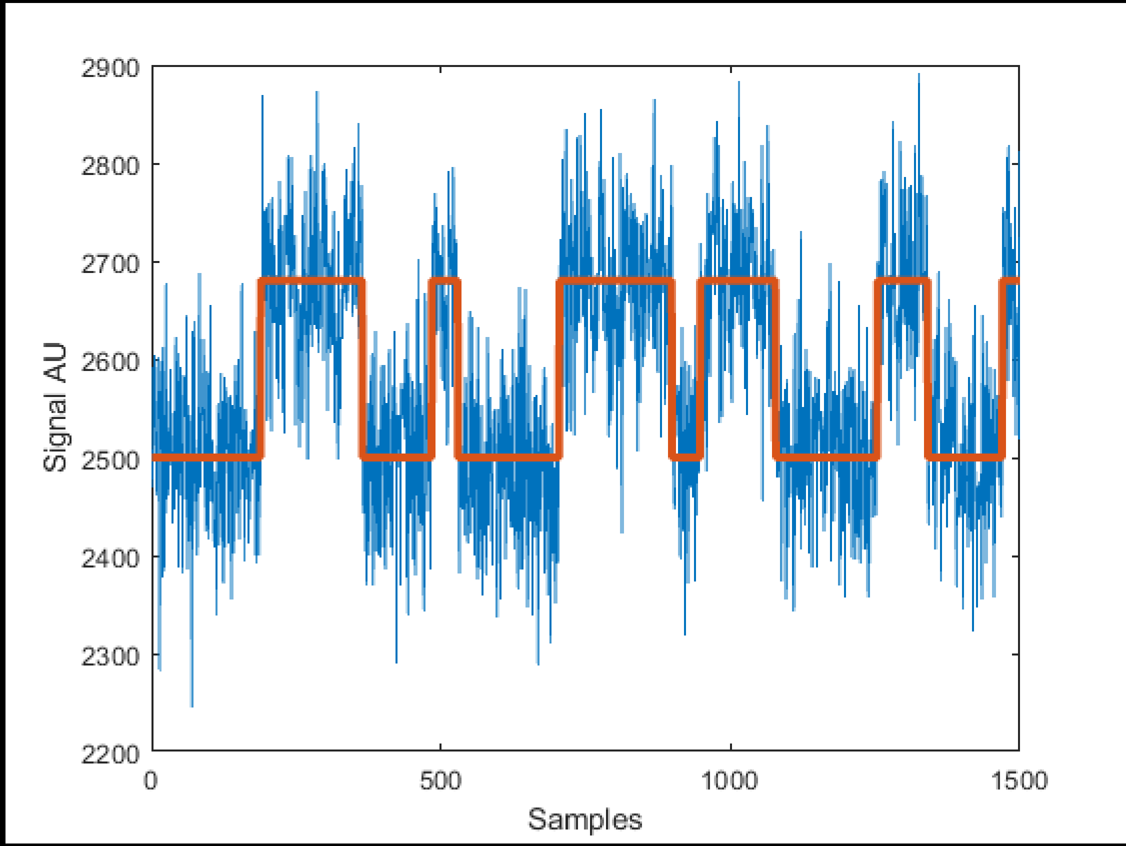
- Nearly all of the white noise is removed, but a few small changes remain
- A new details sequence is creating by subtracting each frame value by the previous frame value
- The new details sequence s is of $N - 1$ where N is the size of the original signal
- Because the noise is already suppressed, the threshold need not be so discriminatory, as such

$$T_s = s_{MAX} * u_{0.75}$$



DWT DENOISING METHOD

- Inverse DWT is performed on heavily the thresholded signal
- Mean levels are sorted and a Gaussian noise free signal is constructed



DEEP LEARNING ANALYSIS

- Convolutional filters are used again
- The filters aren't necessarily step shaped
- Filter shapes start random, and are altered during the 'learning' process
- Models are trained by minimizing some cost, or error function
- Results from convolutions are always run through some activation function → non-linearity

MACHINE LEARNING METHOD

- RTS signals are detected and approximated using convolutional neural networks
- 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 cleaner signal
- Both models are trained on simulated data
  - 90,000 RTS signals and 90,000 Gaussian noise only signals
    - 89,000 of each in the training set, 1,000 for the testing set
  - RTS signals have a distribution of amplitudes and state lifetimes
  - Gaussian noise is added over the transitions

MACHINE LEARNING METHOD

- The ML method requires special data preparation to work properly
- The training and input signals need to be rescaled between zero and one so that only the shape of the signals (RTS), not the magnitude, offers the defining signal characteristics
- Each value in the signal x is subtracted by a number just below the minimum of the signal creating a new vector x_s
- Then, x_s is divided by a number just above its maximum to create the scaled vector x_{sd}
- Because the scaling must be reversable, a key is maintained of scaling constants for each signal

$$x_s = x - (0.99 * \min(x))$$

$$x_{sd} = x_s / (1.01 * \max(x_s))$$

MACHINE LEARNING METHOD

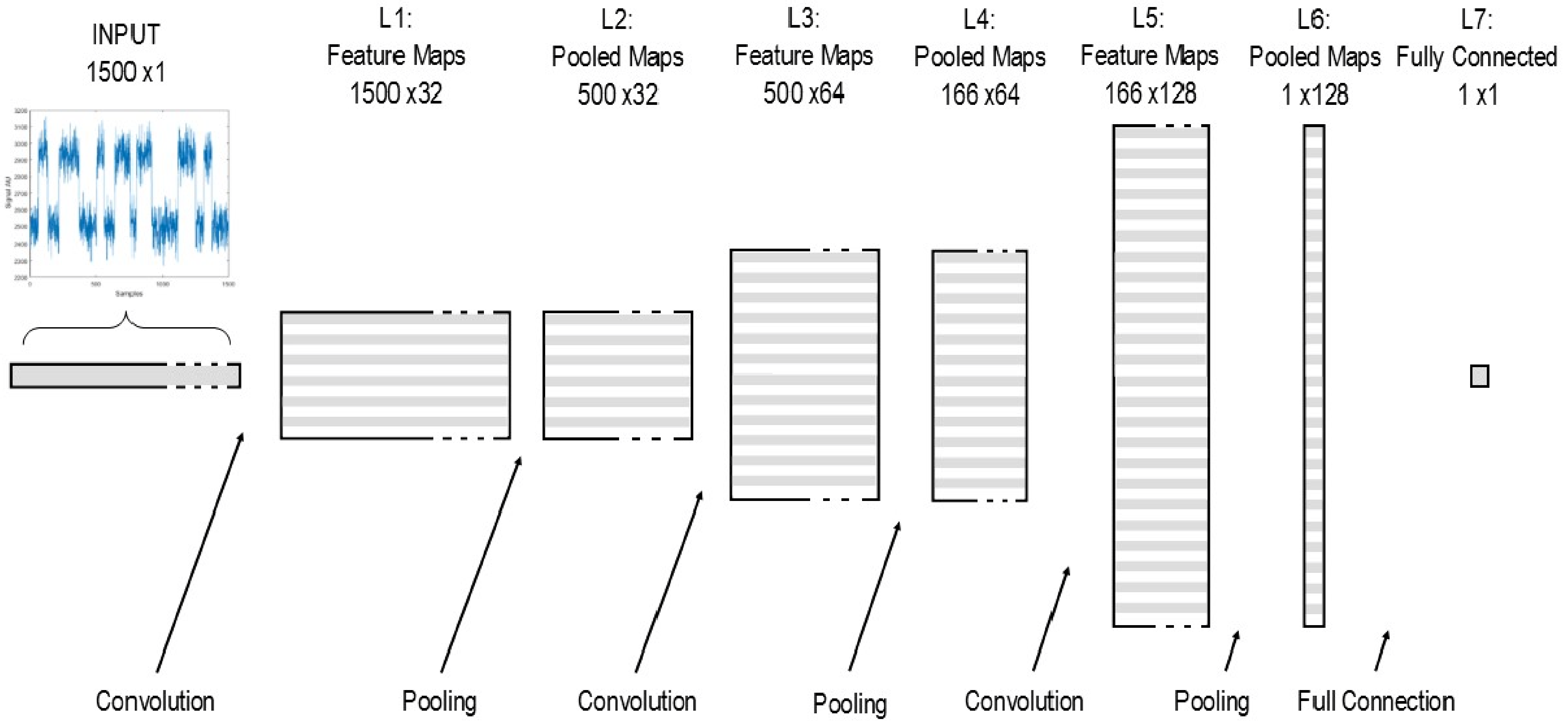
- Signal classifier works similarly to image classifiers
- Developed in Python using a Keras wrapper over Tensorflow
- Convolutional layers extract prominent features and use them to differentiate RTS from non-RTS
- The convolutional layers use the ‘relu’ activation function while the final layer uses a sigmoid to force a choice between zero and one

$$Conv(32,12) \rightarrow Pool(3) \rightarrow Drop(0.5) \rightarrow$$

$$Conv(64,12) \rightarrow Pool(3) \rightarrow Drop(0.5) \rightarrow$$

$$Conv(128,12) \rightarrow Pool() \rightarrow Drop(0.5) \rightarrow$$

$$Fully\ Connected(1)$$



INPUT
1500 x1
L1:
Feature Maps
1500 x32
L2:
Pooled Maps
500 x32
L3:
Feature Maps
500 x64
L4:
Pooled Maps
166 x64
L5:
Feature Maps
166 x128
L6:
Pooled Maps
1 x128
L7:
Fully Connected
1 x1
Convolution
Pooling
Convolution
Pooling
Convolution
Pooling
Full Connection

MACHINE LEARNING METHOD

- The autoencoder squeezes the noisy signal, extracting and prioritizing prominent features with convolutional and pooling layers
- It then expands the signal to original size and creates gaps with upsampling
- The model is trained to fill the gaps with a series of 'wrong' noisy signals and a corresponding set of 'correct' clean signals
- Each convolutional layer used the 'relu' activation function while the last layer uses the linear function to avoid zero values in the reconstructed signal

Layertype(NumFilters,KernelSize)

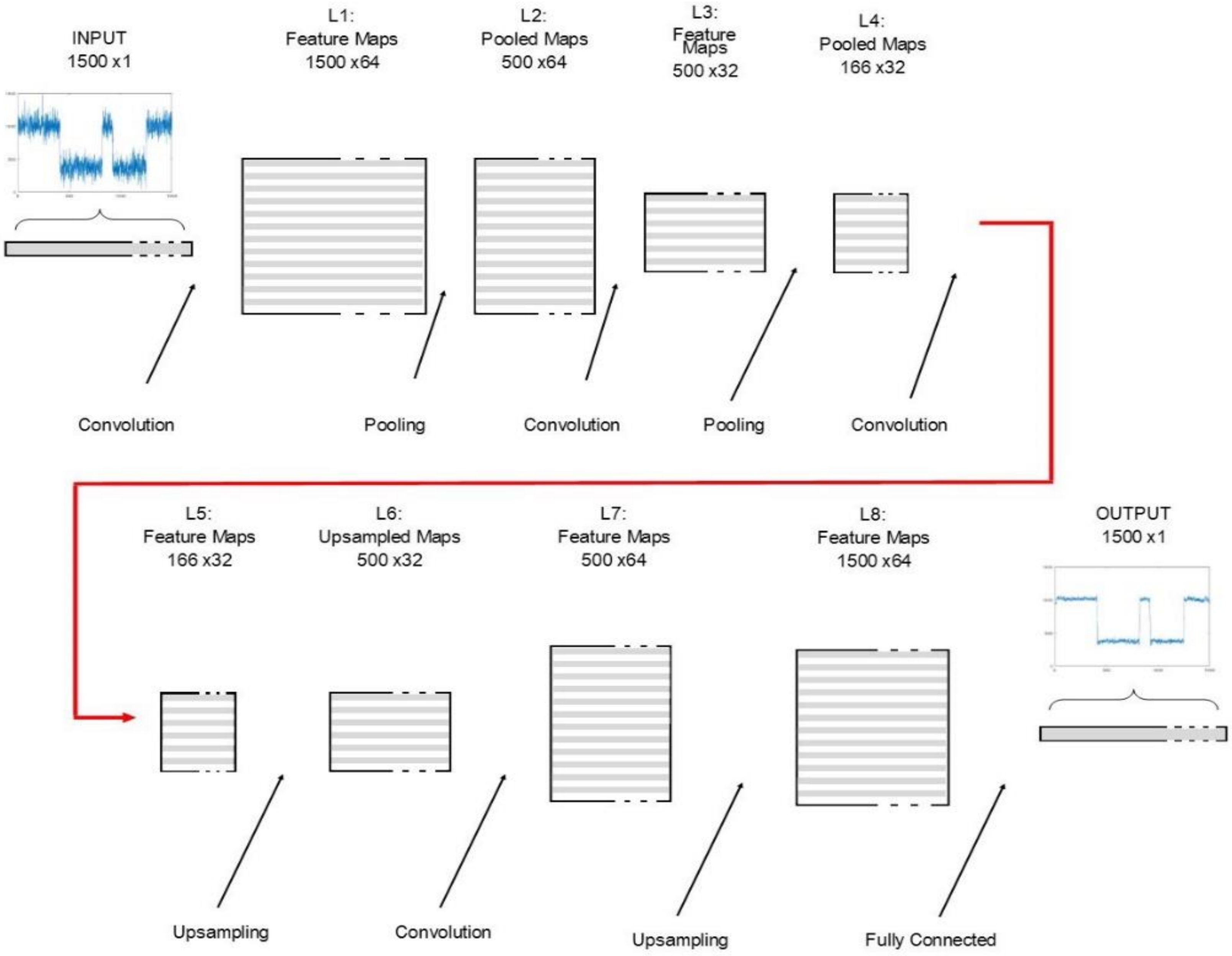
$Conv(64,12) \rightarrow Pool(3) \rightarrow$

$Conv(32,12) \rightarrow Pool(3) \rightarrow$

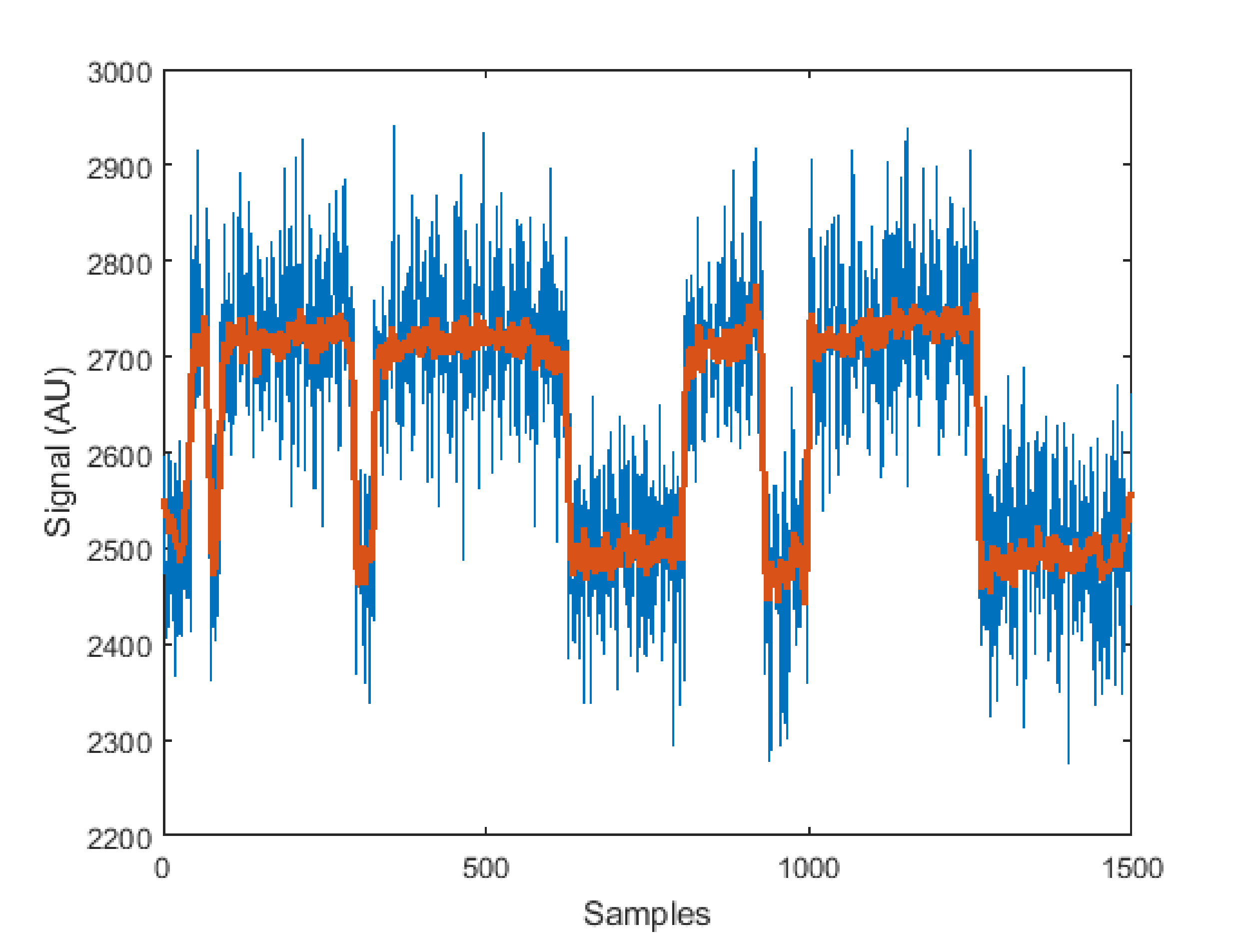
$Conv(32,12) \rightarrow Upsample(3) \rightarrow$

$Conv(64,12) \rightarrow Upsample() \rightarrow$

$Fully\ Connected\ (1500)$

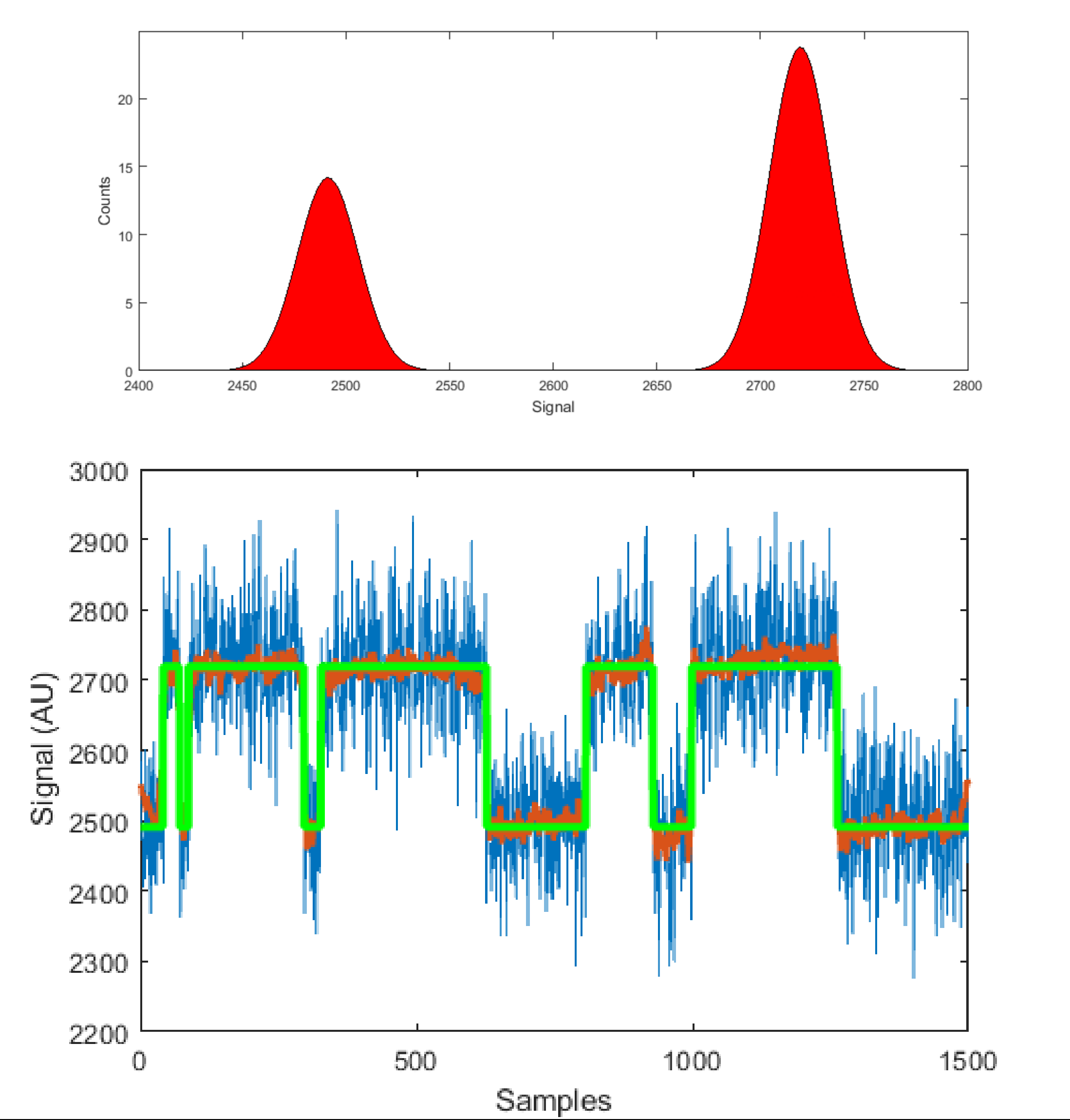

INPUT
1500 x1
L1:
Feature Maps
1500 x64
L2:
Pooled Maps
500 x64
L3:
Feature Maps
500 x32
L4:
Pooled Maps
166 x32
Convolution
Pooling
Convolution
Pooling
Convolution
L5:
Feature Maps
166 x32
L6:
Upsampled Maps
500 x32
L7:
Feature Maps
500 x64
L8:
Feature Maps
1500 x64
OUTPUT
1500 x1
Upsampling
Convolution
Upsampling
Fully Connected

MACHINE LEARNING METHOD



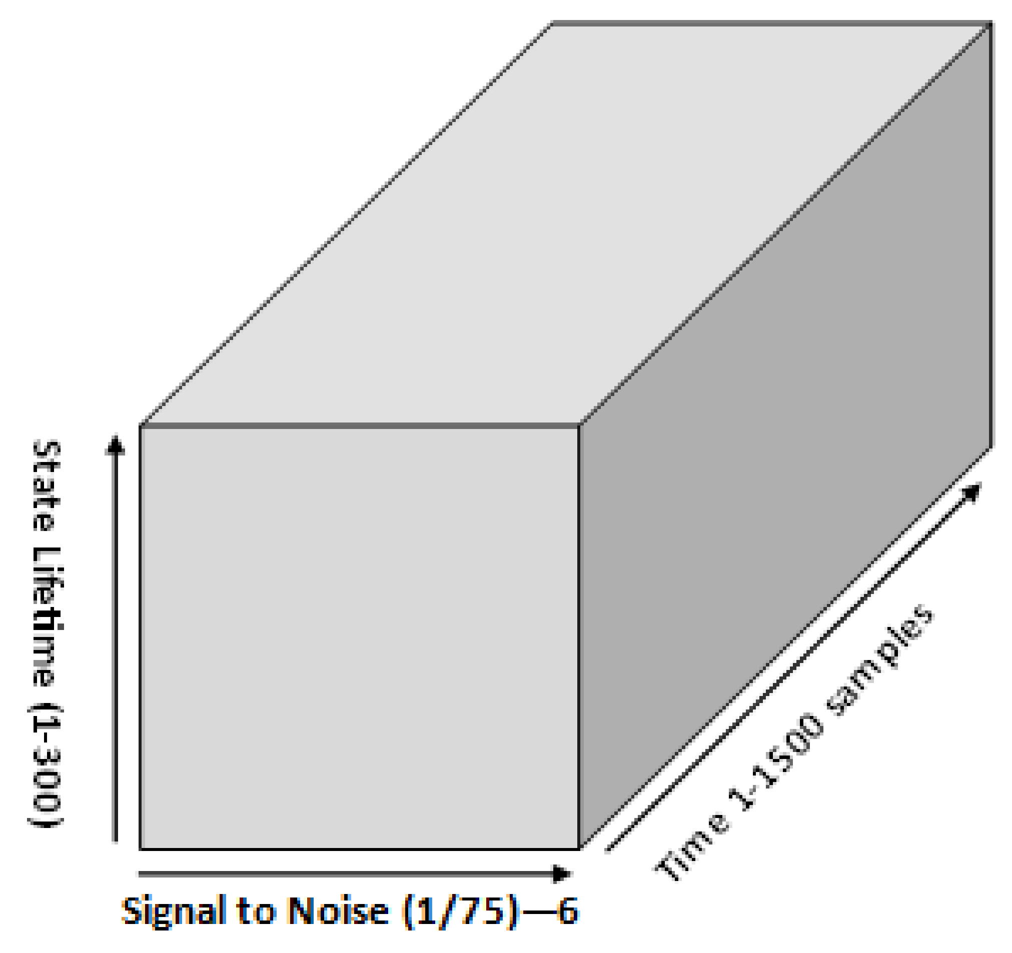
MACHINE LEARNING METHOD

- 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

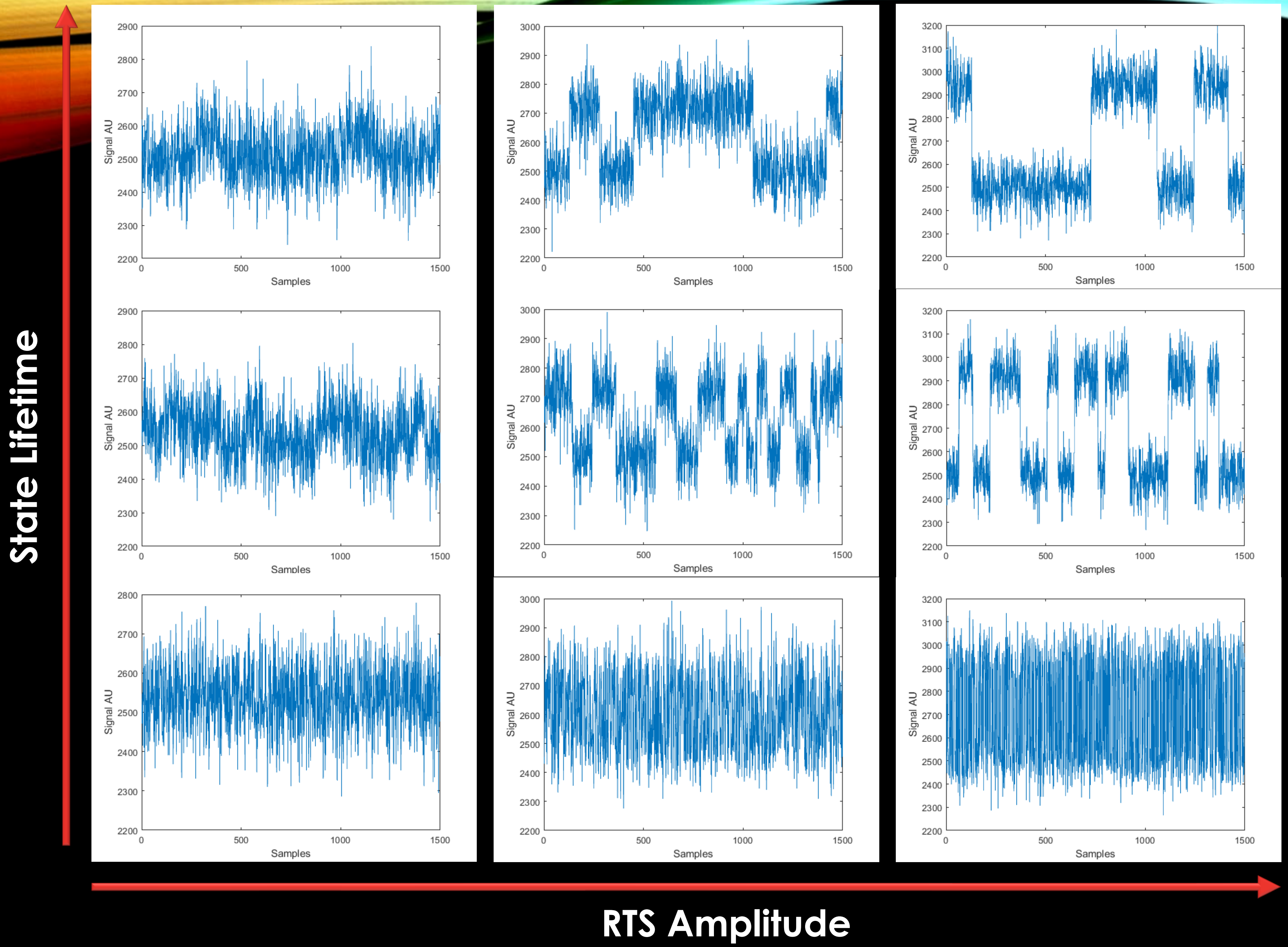


TESTING PROCEDURE

- Each method is tested for detection and approximation on a data block of 90,000 simulated RTS signals
- The signals begin clean, then have Gaussian noise added over the top
- One dimension of the block spans the signal to noise, defined as RTS amplitude/white noise floor, from ~zero to 6
- The other dimension spans the state lifetime of a signal from 1 to 300 samples
- The approximation of each signal is scored by its correlation coefficient against the noiseless version of input signal
- Each method is tested for false positive detection on a block of 90,000 non-RTS signals



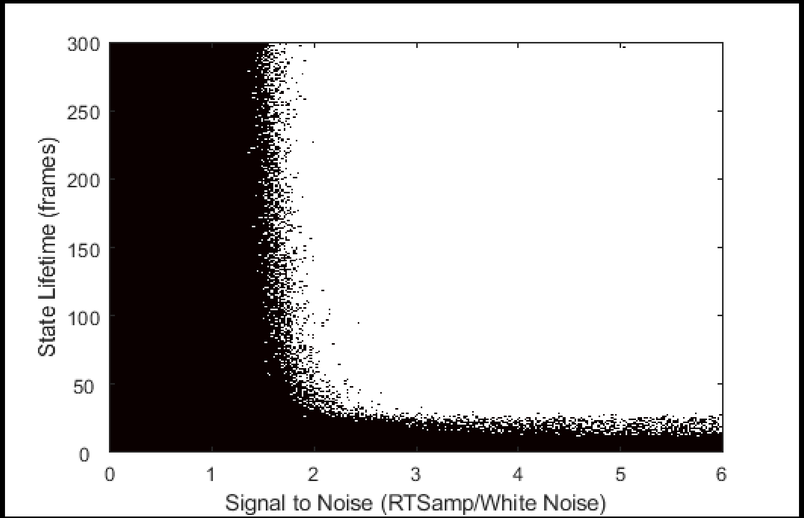
$$C_{xy} = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2}\sqrt{\Sigma(y-\bar{y})^2}}$$


State Lifetime
RTS Amplitude
Signal AU
Samples
0
500
1000
1500
2200
2300
2400
2500
2600
2700
2800
2900
3000
3100
3200

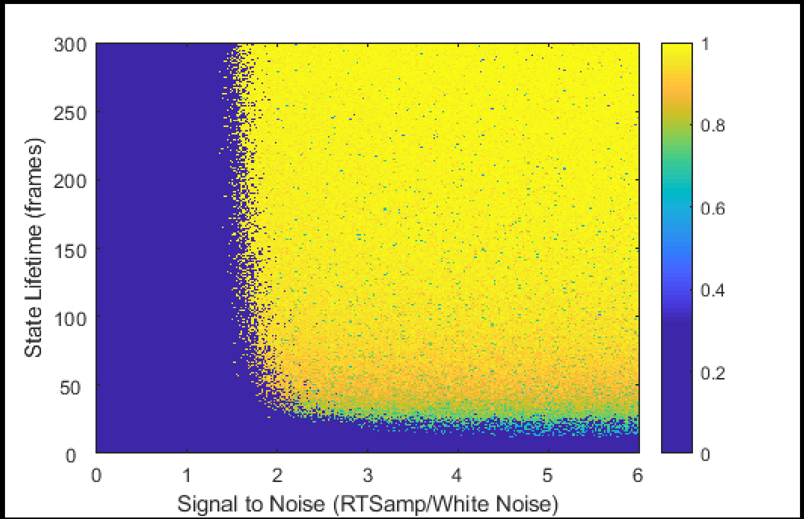
RESULTS – CONVOLUTION METHOD

- Reliably works with $SNR > \sim 2$ and $\tau > \sim 50$ frames
- 66% RTS detection rate
- Mean C_{xy} of for detected signals: 0.9474
- Zero(!) non-RTS false positives

Detection



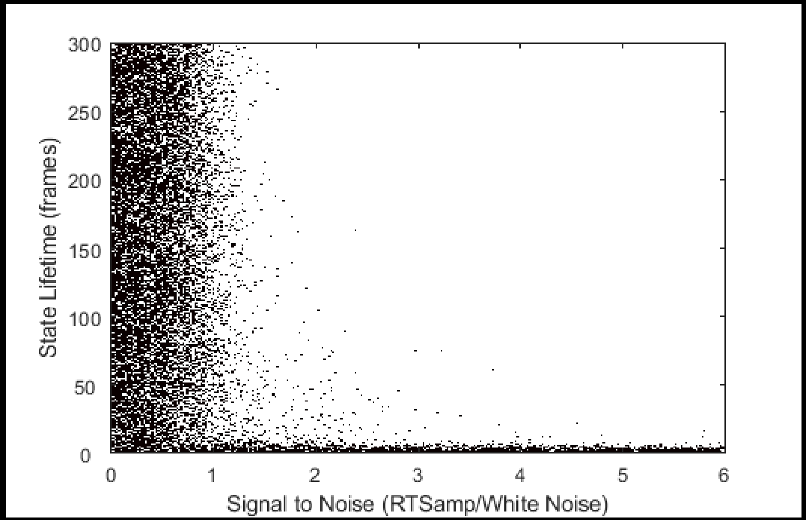
Correlation



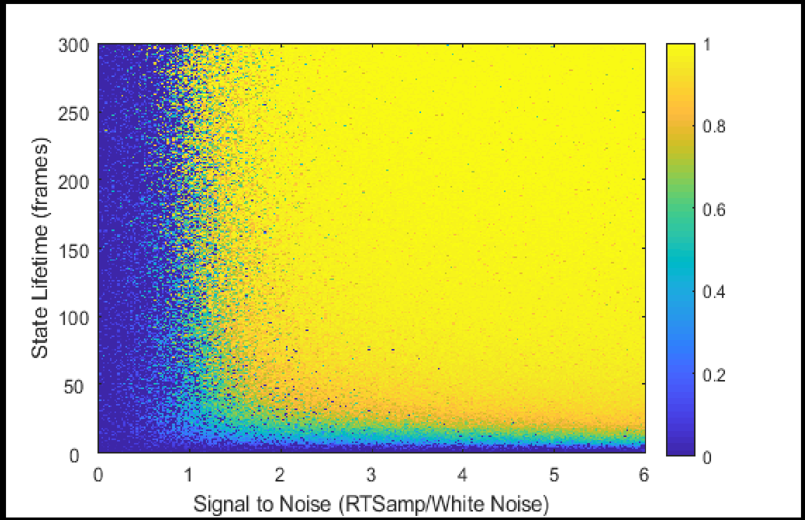
RESULTS – WAVELET METHOD

- Reliably works with $SNR > \sim 2$ and $\tau > \sim 50$ frames
- 86.6% RTS detection rate
- Mean C_{xy} of for detected signals: 0.8644
- 21.7% non-RTS false positive detection

Detection



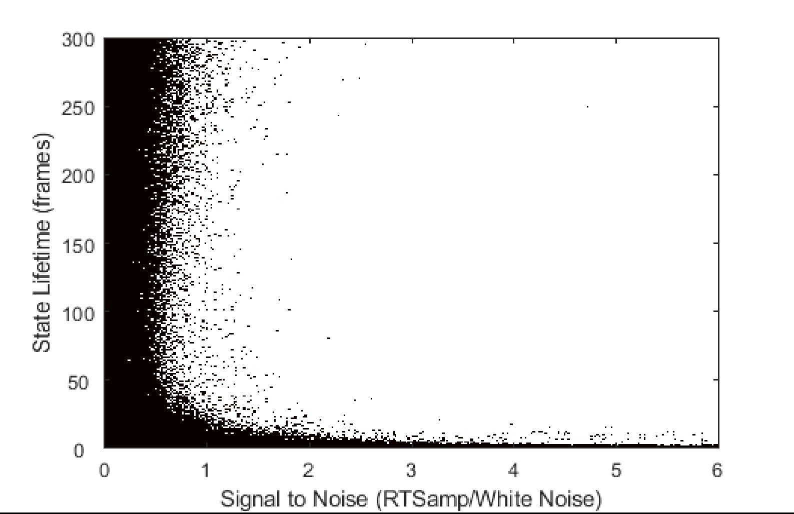
Correlation



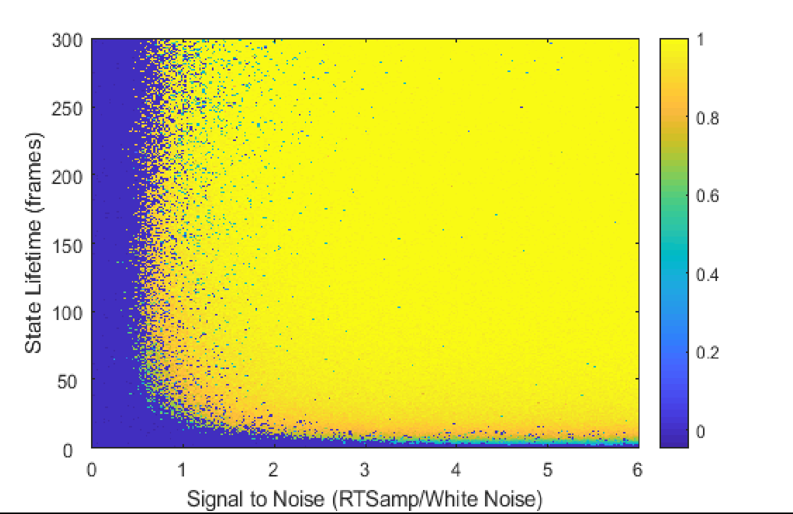
RESULTS – MACHINE LEARNING METHOD

- Reliably works with $SNR > \sim 1$ and $\tau > \sim 25$ frames
- 83.5% RTS detection rate
- Mean C_{xy} of for detected signals: 0.9780
- Zero(!) non-RTS false positives

Detection

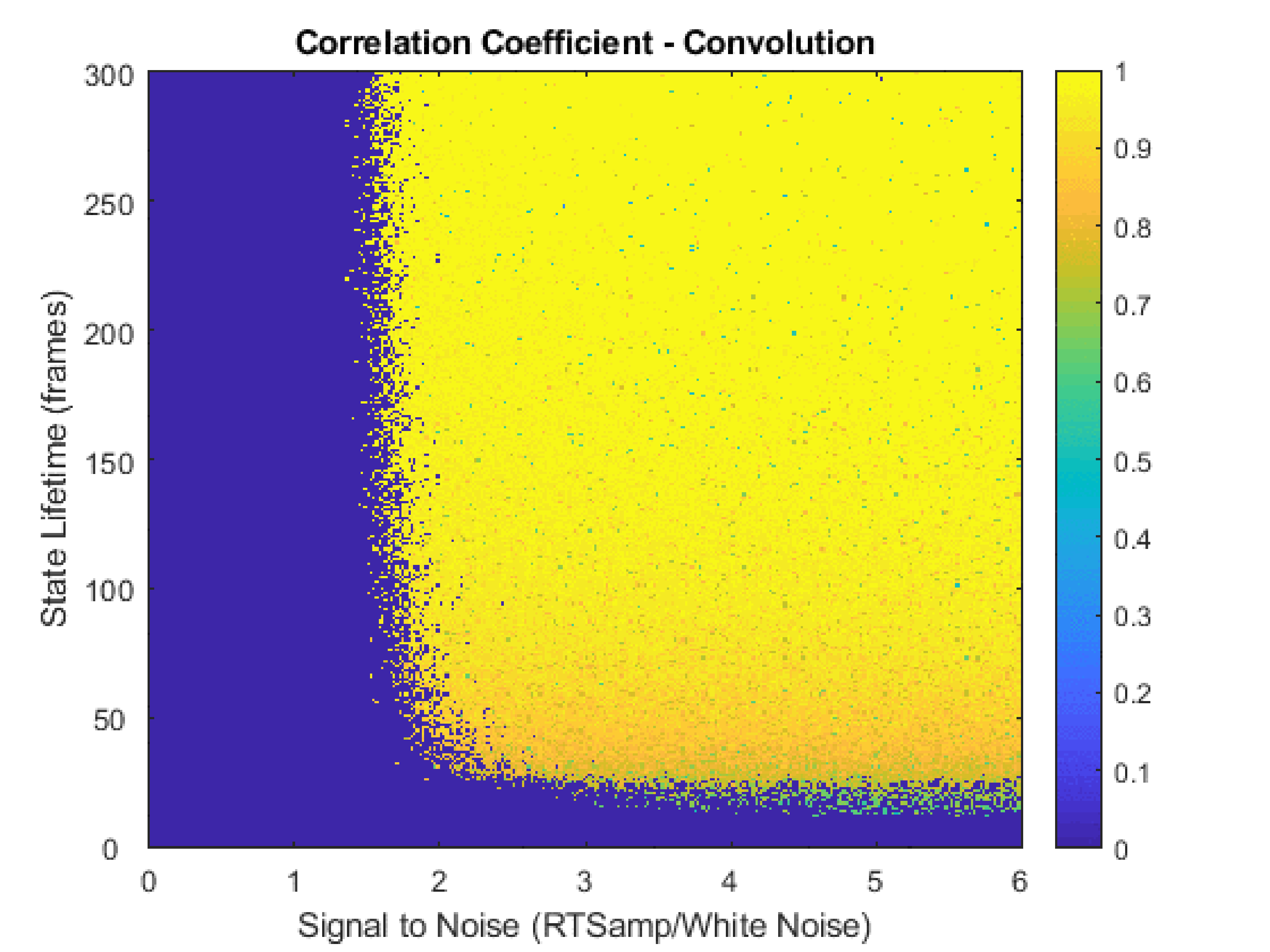


Correlation



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



RESULTS - DISCUSSION

- Convolution
  - Threshold: High
  - Correlation: High
- Wavelets
  - Threshold: Low
  - Correlation: Low
- Machine Learning
  - Threshold: Low
  - Correlation: High

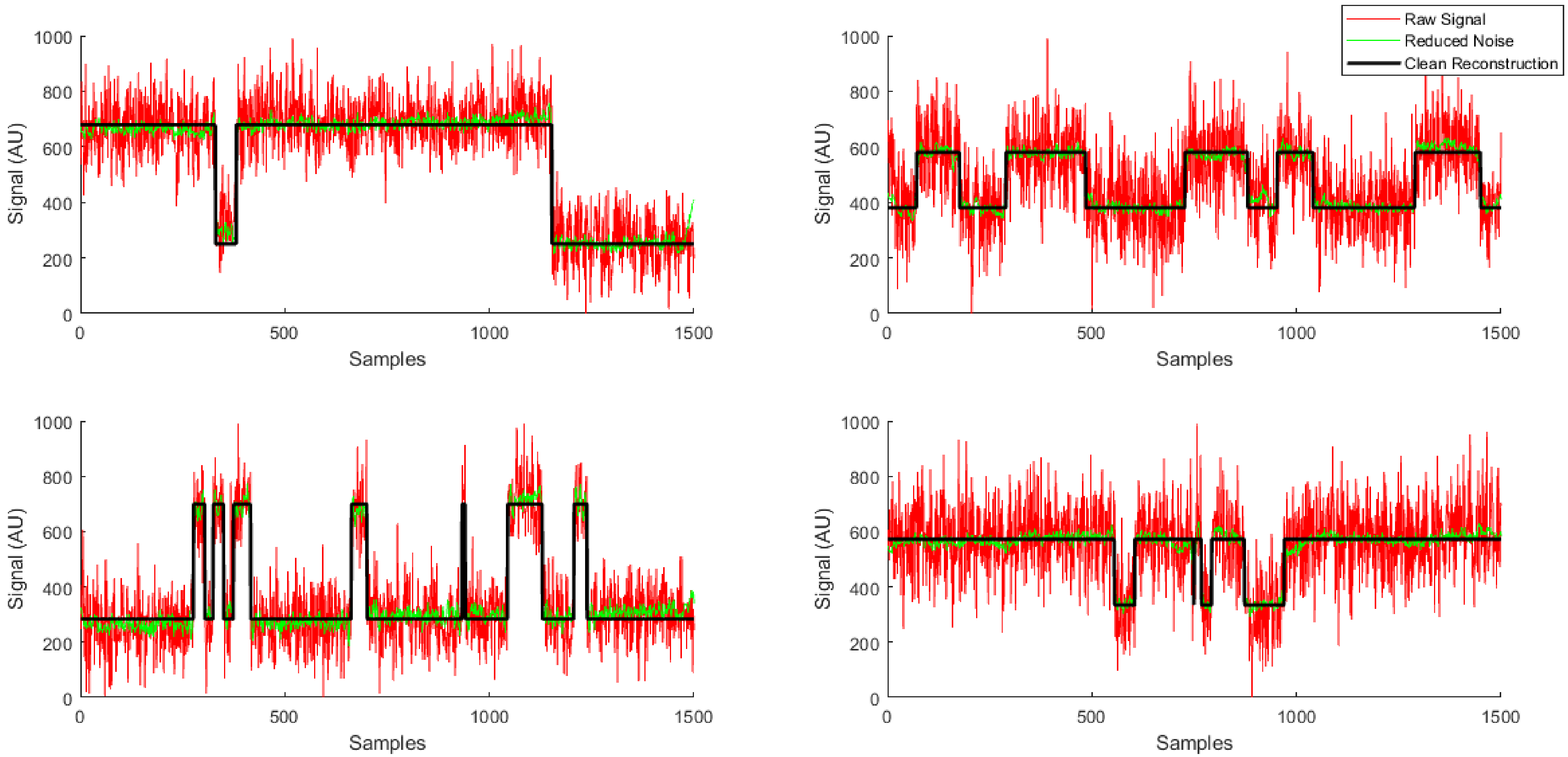
Ideal

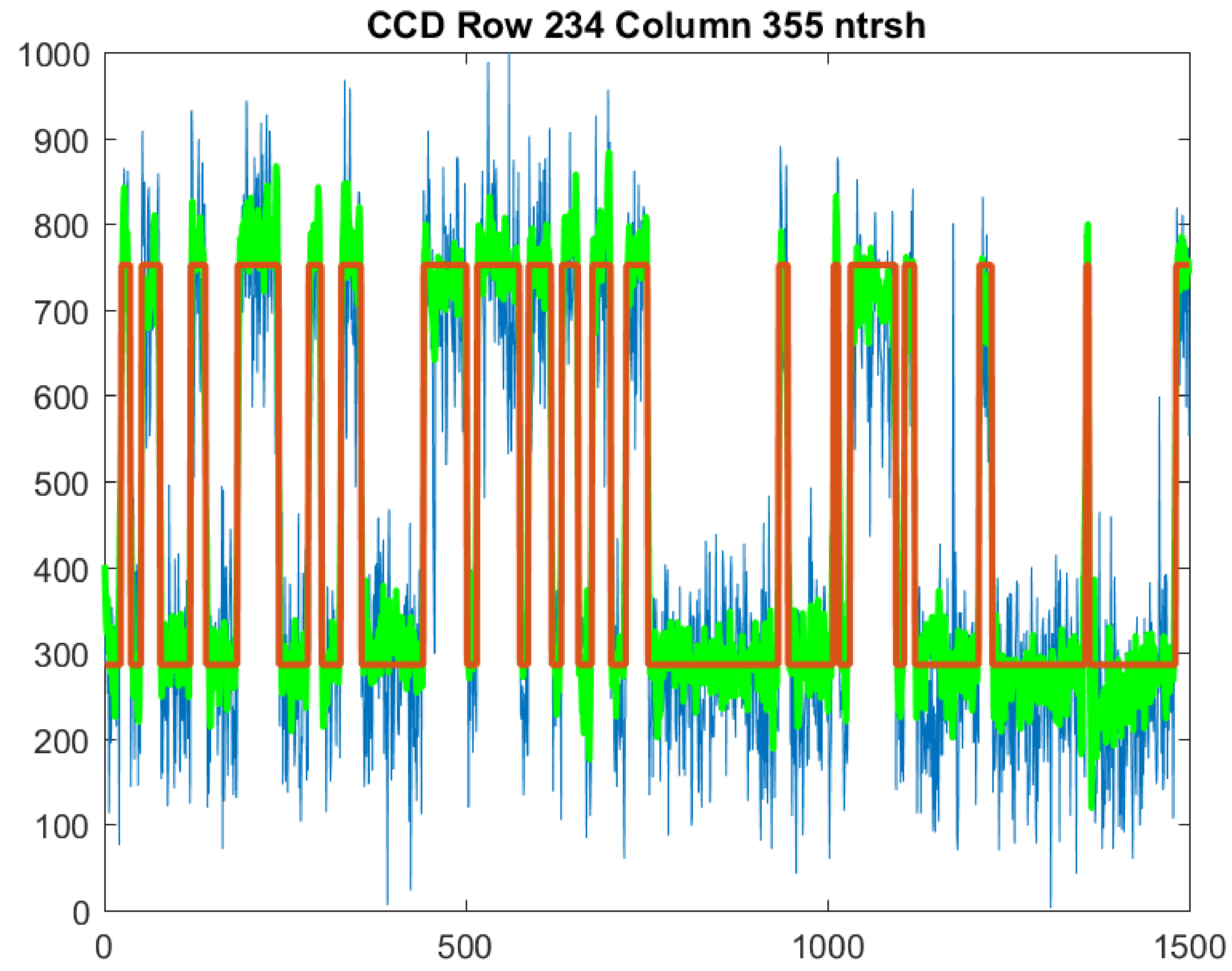
Correlation Range	Convolution Counts	Wavelet Counts	M.L. Counts
$0 < C_{xy} < 0.4$	6	5,342	68
$0.4 \leq C_{xy} < 0.6$	162	3,746	257
$0.6 \leq C_{xy} < 0.7$	674	2,772	251
$0.7 \leq C_{xy} < 0.8$	2,404	3,755	730
$0.8 \leq C_{xy} < 0.9$	5,271	6,460	2,523
$0.9 \leq C_{xy} < 0.99$	42,660	47,600	21,613
$C_{xy} \geq 0.99$	8,143	8,274	49,659

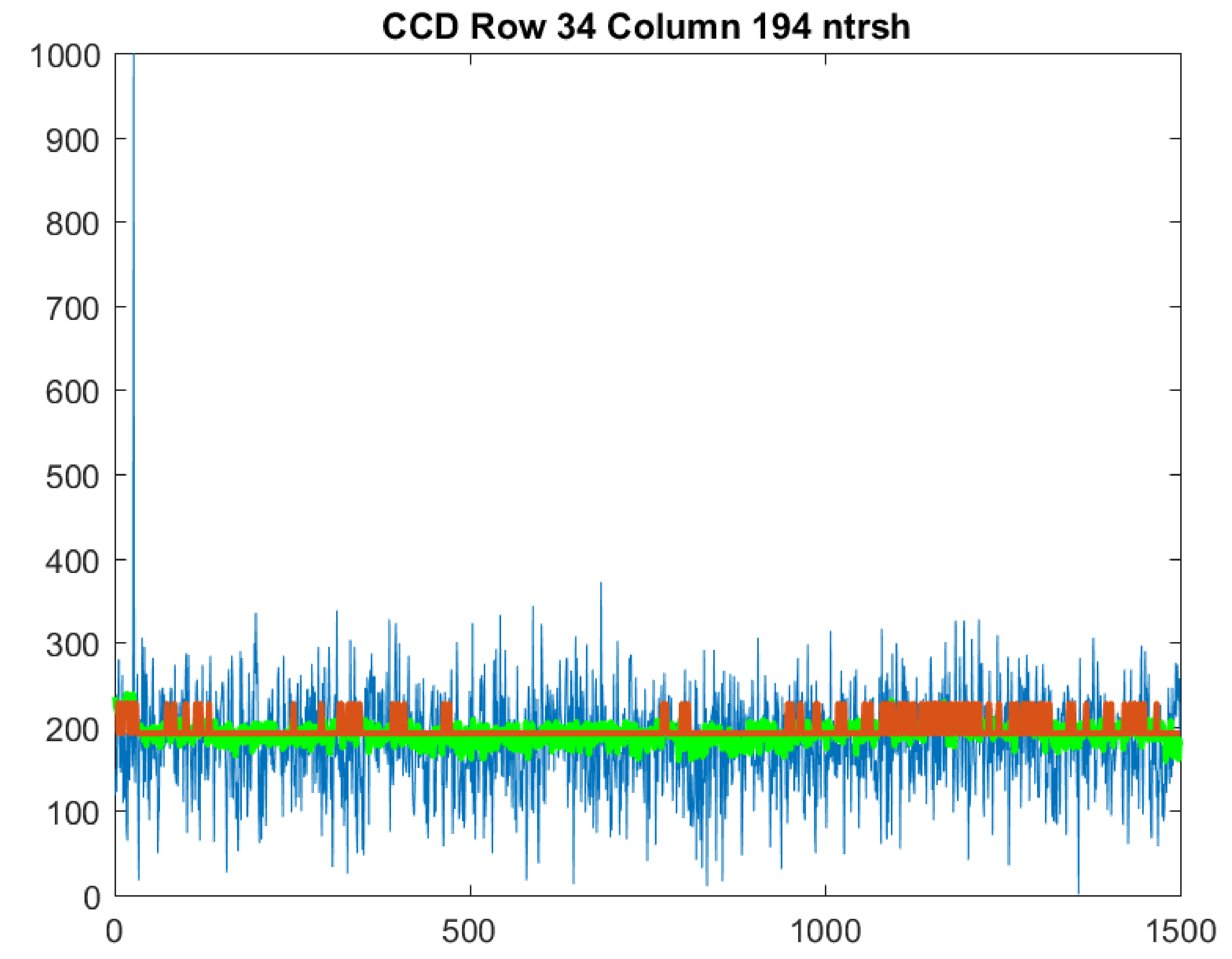
THANKS FOR LISTENING

- Questions?

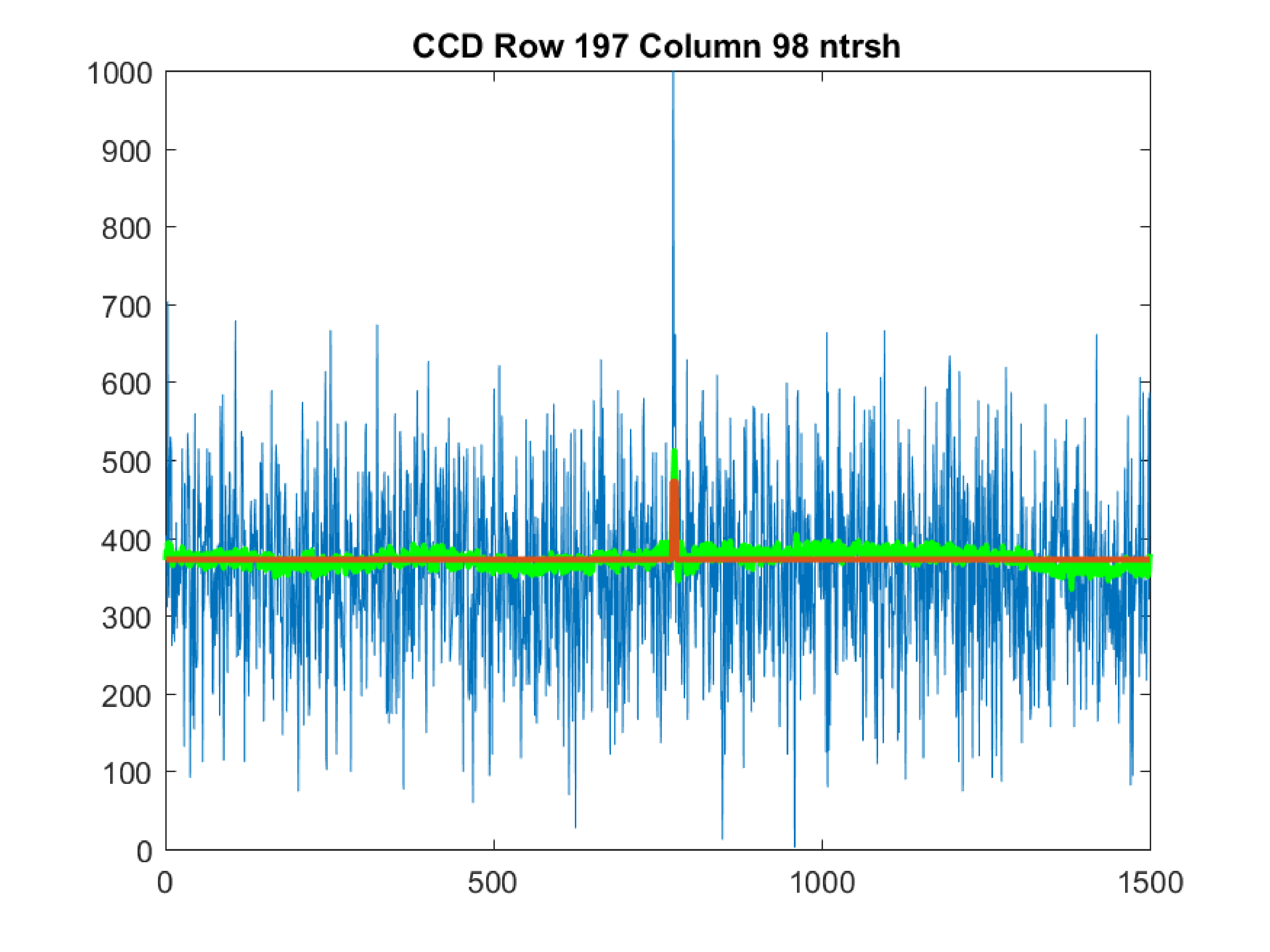
BACKUP SLIDES


Raw Signal
Reduced Noise
Clean Reconstruction
Signal (AU)
Samples
0
200
400
600
800
1000
500
1500


CCD Row 234 Column 355 ntrsh
1000
900
800
700
600
500
400
300
200
100
0
0
500
1000
1500



CCD Row 34 Column 194 ntrsh
1000
900
800
700
600
500
400
300
200
100
0
0
500
1000
1500


CCD Row 197 Column 98 ntrsh
1000
900
800
700
600
500
400
300
200
100
0
0
500
1000
1500

WAVELET OPERATORS – TRENDS SEQUENCE

- $\mathbf{V}_1^1 = \frac{1}{\sqrt{2}}(1,1,0,0,0,\ldots); \ \mathbf{V}_2^1 = \frac{1}{\sqrt{2}}(0,0,1,1,0,\ldots)$

$$a_1 = \frac{f_1+f_2}{\sqrt{2}} = \mathbf{f}\cdot\mathbf{V}_1^1; \ a_2 = \frac{f_3+f_4}{\sqrt{2}} = \mathbf{f}\cdot\mathbf{V}_2^1$$

$$a_m = \frac{f_{2m-1}+f_{2m}}{\sqrt{2}} = \mathbf{f}\cdot\mathbf{V}_m^1$$

WAVELET OPERATORS – DETAILS SEQUENCE

- $\mathbf{W}_1^1 = \frac{1}{\sqrt{2}}(1, -1, 0, 0, 0, \dots)$, $\mathbf{W}_2^1 = \frac{1}{\sqrt{2}}(0, 0, 1, -1, 0, \dots)$

$$d_1 = \frac{f_1 - f_2}{\sqrt{2}} = \mathbf{f} \cdot \mathbf{W}_1^1; \quad d_2 = \frac{f_3 - f_4}{\sqrt{2}} = \mathbf{f} \cdot \mathbf{W}_2^1$$

$$d_m = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} = \mathbf{f} \cdot \mathbf{W}_m^1$$