**Generalized Thresholding for Efficient Spike Acquisition**

**INTRODUCTION**

**METHODS**

**Analog Thresholding**

[describe and cite previous methods, refer to fig 1a]

**FRI**

Vetterli [cite] introduced the concept of signals with finite rate of innovation (FRI). In particular, signals of the following form, the sum of shifted and scaled Dirac delta functions, were considered.



Here,  is the Direc delta function. Kusuma [cite] proposed sampling this type of signal by successive integration as follows.







 samples  are extracted from  as shown above. As shown in [cite], the parameters  and  (for all ) can be reconstructed from the samples, provided that .

In this paper we consider the same sampling scheme, but a different signal with finite rate of innovation: a stream of square waves with uniform height and variable width.

\subsection{Generalized Analog Thresholding (gAT)}

Figure \ref{fig:cartoon} illustrates the difference between conventional analog thresholding (AT) and our proposed approach, generalized analog thresholding (gAT). Both methods start with a raw neural signal, partitioned into blocks of duration . Both methods have the same goal, which is to decide whether a spike occurred in each block, and if so, to report the time at which it occurred. This signal is bandpass filtered to remove local field potentials. A threshold  is chosen, below the amplitude of a typical spike but above the amplitude of typical noise. At this point the two methods begin to differ.

Analog thresholding (AT) applies a latched comparator with threshold . Whenever the signal exceeds , the comparator latches high. At the end of each block, the output of the comparator is sampled and the comparator is reset. This yields a binary value indicating the presence of a spike in this block. For the sake of comparison with gAT, we need AT to report a precise spike time within the block. We chose the midpoint of the block. This can be considered an MMSE (minimum mean squared error) estimate because, provided that the true spike time is equally likely to be at each point in the block, the midpoint minimizes the mean squared error between the true and reported spike time.

Generalized analog threshold (gAT) differs from AT as follows. Instead of applying a latched comparator it simply applies a comparator. This comparator outputs a continuous binary signal that indicates whether the neural waveform exceeds  at each instant in time. Assume for simplicity that we are dealing with the block . To analyze this block, the first and second integrals of the comparator output in the block are calculated. If the comparator output in the block is given by the function  on the domain , the first and second integrals  and  are given by the following.







If  then gAT concludes that there is no spike in the block. Otherwise, a spike is reported at time



See the Appendix for the derivation. Note that this gives gAT super-resolution, meaning the ability to pinpoint the time of a spike on scales smaller than .

**RESULTS**

We ran a number of simulations to compare the performance of AT and gAT. Although the methods themselves were simulated on a computer rather than implemented in analog, we used real neural data recorded from the primary motor cortex of monkeys. [how many monkeys, how many channels, how many neurons] The data included ground truth spike times that were obtained using [exact description]. These are what we will refer to as the true spike times.

In Figure \ref{fig:superres} we tested the ability of gAT to provide super-resolution of spike times on scales smaller than $T$. The metric of interest is simply unsigned error in spike time,  where  and  are the true and reconstructed spike times in a block. Note that this metric can only be computed for blocks that have exactly one true spike and exactly one reconstructed spike. We will refer to these types of blocks as ``valid'' blocks. Given a method (AT or gAT), channel, and sampling period , we computed the average spike time error by averaging the unsigned error in spike time over all valid blocks in 1 minute of data. In Figure \ref{fig:superres} panels (a) and (c) we see that when AT is used, average spike time error increases in proportion to , whereas when gAT is used, average spike time error remains relatively flat. This confirms the ability of gAT to provide super-resolution of spike times.

The theoretical curve for AT, shown in black, is the line $y = T/4$ (where $y$ is the y-axis value and $T$ is the x-axis value). To calculate this, first note that for the block $[0,T]$, the AT reconstruction spike time is always $t\_{reconstructed} = T/2$. Then assume that the true spike time $t\_{true}$ follows a uniform distribution on $[0,T]$. The expected value of \mbox{$|t\_{true} - t\_{reconstruted}|$} is $T/4$. For large values of $T$, the AT curve drops slightly below the theoretical AT curve. To explain this, recall that average spike time error can only be computed using ``valid'' blocks, those with exactly one true spike and one reconstructed spike. When $T$ becomes large, the probability of having multiple true spikes in a block becomes significant. As a result, given that a block is valid, the true spike is likely to be near the center of the block. To understand why this is, consider for example the extreme case where the true spikes are spaced apart by exactly $T/2 + \varepsilon$, for some small $\varepsilon > 0$. The only way there can be a valid block is if it is centered on a real spike.

The fact that panels (a) and (c) of Figure \ref{fig:superres} use only valid blocks is potentially worrying. If only a few blocks were valid, the results of these analyses would be misleading. Panels (b) and (d) show that this is not the case. Recall that a valid block is a block with exactly one true spike and exactly one reconstructed spike. Define an ``active'' block to be a block with one or more true spikes. Define the fraction of valid blocks as follows.

\begin{equation}

\label{eq:valid}

\mathrm{Fraction\,\,of\,\,Valid\,\,Blocks} = \frac{\mathrm{\#\,\,Valid\,\,Blocks}}{\mathrm{\#\,\,Active\,\,Blocks}}

\end{equation}

The active blocks are those with true spiking activity, so they are the blocks that we would ideally include in the analysis. The valid blocks are those that were actually included in the analysis, so the fraction of valid blocks is a measure of how much of the data was used for the analysis. We see that the fraction of valid blocks is generally above $0.5$. As a result, the results in panels (a) and (c) represent most of the data. However, there is still a significant number of blocks that needed to be thrown out. Next we consider a different error metric that takes all blocks into account in order to asset overall performace.

In the analysis detailed in Figure \ref{fig:ROC}, instead of comparing true and reconstructed spikes on a block-by-block basis, we compared a 1-minute-long spike train of true spikes with the corresponding spike train of reconstructed spikes. One issue that arises in this setting is the possibility that a true spike intersects the boundary between two blocks, thus triggering threshold crossings in both adjacent blocks creating two reconstructed spikes. To solve this problem, we chose to enforce a mandatory refractory period $\Delta = 1.1\,ms$ in the reconstructed spike train. Any reconstructed spike whose spike time exceeds that of the previous reconstructed spike by less than $\Delta$ is thrown out.

After imposing a refractory period, we compared the two spike trains by compuing the number of false positives and false negatives. We chose an error tolerance value $\tau = 5\,ms$ and paired up reconstructed spikes with true spikes that were no more than $\tau$ away. The true spikes that could not be paired up in this manner are considered false negatives and the reconstructed spikes that could not be paired up are considered false positives.

Given a true and reconstructed spike train, there is a tradeoff between false positives and false negatives that is controlled by the spike detection threshold $\delta$. Increasing $\delta$ lowers the number of false positives but increases the number of false negatives. In panels (a) and (b) of Figure \ref{fig:ROC} we see that gAT gives a more favorate tradeoff than AT. We also wanted a way to assign a single error value to a pair of spike trains (one true and one reconstructed) rather than an entire tradeoff curve. We did this by computing the number of error (false positives plus false negatives) at the point on the tradeoff curve at which the number of errors is minimized. In panels (c) and (d) we see that gAT offers lower error values than AT for large values of $T$. The two methods have similar performance when $T$ is small, as should be expected.

Next we explore potential applications of gAT. There are various application areas in which detecting spikes in an analog signal is essential. These benefit from the super-resolution provided by gAT to various extents. For some applications, pinpointing exact spike times are critical, whereas for others, approximate spike time is just as good. The benefits of gAT are most advantageous in applications in which exact spike times are critical.

First we consider an analysis of LFP (local field potential) phase locking. For some neurons, spikes are most likely to occur during particular phase of LFP oscillation. Given spike times and LFP data we can perform the following analysis to detect this phase locking behavior. For each spike we calculate the phase of the 20-30 Hz oscillation in the LFP at the instant in time at which the spike occurred. We can then plot the distribution of these phase values, and calculate the preferred phase. Figure \ref{fig:LFP}(a) shows an example of this plot for a particular neuron using 3 methods: true spikes, AT, and gAT. Since phase locking is strongest at high amplitude LFP, we threw out spikes that occurred when the LFP amplitude was below some threshold. This threshold was chosen to be the median of the LFP amplitude values at the true spike times. As a result, half of the true spikes were thown out. Approximately half of the spikes were thrown out in each of the other methods, AT and gAT.

Figure \ref{fig:LFP}(b) investigates how the error in preferred direction changes as a function of the sampling period $T$ using each of the methods AT and gAT. Error in preferred direction is defined as the absolute difference between the preferred direction of the true spikes and the preferred direction of the reconstructed spikes. We see that with gAT, the error in preferred direction remains flat as $T$ increases, whereas with AT, the error increases with $T$. Figure \ref{fig:LFP}(c) investigates how the shape of the phase distribution shown in (a) changes as a function of $T$. The error metric used is the average (over all phase values) of the absoulute difference between the phase distribution calculated using the true spikes and the phase distribution calculated using the reconstructed spikes. Once again we see smaller error with gAT than with AT.

As a result of the analysis shown in Figure \ref{fig:LFP} we can conclude that gAT outperforms AT in fidelity of phase locking analysis. We now investigate an application in which gAT does not provide a significant advantage over AT. We used the same data from monkey primary motor cortex but incorporated knowledge of the monkey's arm velocity at the time of spiking activity. The monkey's arm movements were tracked simulateously with neural data. Many neurons are tuned to this arm movement, meaning they have a preferred direction and fire most frequently when the monkey's arm is moving in a direction similar to this preferred direction. Figure \ref{fig:tuning} details this analysis. We see that both methods, AT and gAT, give nearly identical performance. Although this analysis is similar to the phase locking analysis shown earlier, there is an important distinction. The phase of 20-30 Hz LFP oscillations changes much more quickly than monkey arm velocity. As a result, precise spike time is critical for the LFP analysis but unnecessary for the tuning curve analysis.

\begin{figure}[!t]

\centering

\includegraphics[width=\columnwidth]{eps/fig1\_final}

\caption{{\bf Comparison of analog thresholding (AT) and generalized analog thresholding (gAT).} Both methods process the data in blocks of length $T$, reconstructing either 0 or 1 spikes in each block. Furthermore, both methods report the existence of a spike in a block if and only if the signal crosses an appropriately-chosen threshold $\delta$ somewhere in that block. Analog thresholding (AT) provides no information about which part of the block contains the spike, so the spike is simply reported to be in the center of the interval. Generalized analog thresholding (gAT) is able to pinpoint the location of a spike within a block. Note that both methods break down when two or more true spikes (threshold crossings) occur in one block.}

\label{fig:cartoon}

\end{figure}

\begin{figure}[!t]

\centering

\includegraphics[width=\columnwidth]{eps/fig2\_new}

\caption{{\bf Super-resolution spike-time recovery with gAT.} {\bf(a)} Average unsigned spike-time reconstruction error of the two methods as a function of the sampling period $T$, with a curve for each of the 10 channels. Average spike time error was computed using all blocks that contain exactly 1 true spike and 1 reconstructed spike. This is because spike-time error is only easily-defined for such blocks. The theoretical curve $y = T/4$ for AT is included. {\bf (b)} Fraction of valid blocks as a function of sampling period $T$, with a curve for each of the 10 channels. The fraction of valid blocks is defined by (\ref{eq:valid}). These values represent how much of the data was useable for computing the values in (a). {\bf (c)} Same data as in (a) but aggregated over all channels. The mean of all 10 channels is shown, along with 95\% bootstrap confidence intervals on the mean. {\bf (d)} Same data as (b) but aggregated over all channels, showing mean and 95\% bootstrap confidence interval on the mean.}

\label{fig:superres}

\end{figure}

\begin{figure}[!t]

\centering

\includegraphics[width=\columnwidth]{eps/fig3}

\caption{{\bf gAT improves spike detection accuracy over AT.} {\bf (a)} Each curve shows the tradeoff between false positive and false negatives (reported as a fraction of true spikes) for a different channel, calculated using all the blocks in 1 minute of data. Different points on a single curve were obtained by changing the value of the spike detection threshold $\delta$. Increasing $\delta$ lowers the number of false positives but increases the number of false negatives. The sampling period $T = 15\,ms$, error tolerance $\tau = 5\,ms$, and refractory period $\Delta = 1.1\,ms$ were held constant throughout. The large dark data point on each curve shows the optimal operating point, the point with the fewest number of total errors (false positives plus false negatives). {\bf (b)} The analysis in (a) was performed for various values of $T$. The number of errors per true spike was calculated at the optimal operating point for the particular channel and method in question. Each curve corresponds to a particular channel. Data for all 10 channels is shown. {\bf (c)} Same data as (a) but showing only 3 representative channels: the best, worst, and median. {\bf (d)} Same data as (b) but aggregated over all channels, showing mean and 95\% bootstrap confidence interval on the mean.}

\label{fig:ROC}

\end{figure}

\begin{figure}[!t]

\centering

\includegraphics[width=\columnwidth]{eps/fig4\_final}

\caption{{\bf gAT outperforms AT in LFP phase locking analysis.} {\bf (a)} A single-channel example showing phase locking using 3 methods: true spikes, AT, and gAT. Only spikes occurring during high amplitude LFP (above the median of LFP amplitude values at true spike times) were considered. The parameters $T = 30\,ms$ and $\Delta = 1.1\,ms$ were used. The distribution of spikes over various phases of the oscillation of 20-30 Hz LFP is shown for each method. The relative risk of spiking, shown on the y-axis, is the firing rate divided by the average spiking rate. {\bf (b)} The data from (a) is now fit to a generalized linear model $\lambda(\theta) = e^{b\_0 + v\_1 b\_1 + v\_2 b\_2}$. Here $\theta$ is the phase, $\lambda(\theta)$ is the firing rate, and $(v\_1, v\_2)$ is a unit vector with angle $\theta$. The parameters $b\_0$, $b\_1$, and $b\_2$ are calculated from the data. The underlying firing rate is $e^{b\_0}$ and the preferred phase is the angle of the vector $(b\_1, b\_2)$. The relative risk of spiking is the firing rate $\lambda(\theta)$ divided by the baseline firing rate $e^{b\_0}$. Circles are used to mark preferred phase. {\bf (c)} Investigation of how the shape of the phase distribution shown in (a) changes as a function of $T$ for both methods AT and gAT. Average distribution error is defined to be the average (over all phase values) of the difference between the true distribution curve and the reconstructed distribution curve, both of which are shown in (a). Data is aggregated over all channels, showing mean and 95\% bootstrap confidence intervals. {\bf (d)} Error in preferred phase is shown as a function of $T$ for both methods AT and gAT. Error in preferred phase is defined to be the absolute difference between preferred phase of true spikes and preferred phase of reconstructed spikes. Data is aggregated over all channels, showing mean and 95\% bootstrap confidence intervals.}

\label{fig:LFP}

\end{figure}

\begin{figure}[!t]

\centering

\includegraphics[width=\columnwidth]{eps/fig5\_new}

\caption{{\bf Tuning curves are insensitive to spike time errors.} {\bf (a)} An example of tuning curves for a particular neuron. {\bf (b)} Tuning curves fit to the generalized linear model $\lambda(v\_1, v\_2) = e^{b\_0 + v\_1 b\_1 + v\_2 b\_2}$. Here $(v\_1 v\_2)$ is the 2-dimentional vector describing the monkey's arm velocity and $\lambda$ is the firing rate. The parameters $b\_0$, $b\_1$, and $b\_2$ are calculated from the data. Circles show preferred directions. Given an angle $\theta$, the average firing rate is $\lambda(v\_{\theta 1}, v\_{\theta 2})$ where $(v\_{\theta 1}, v\_{\theta 2})$ is a vector whose angle is $\theta$ and whose magnitude is the monkey's average arm speed. {\bf (c)} Preferred direction error as a function of $T$ for both AT and gAT. Preferred direction error is defined to be the absolute difference between the preferred direction of true spikes and the preferred direction of reconstructed spikes. Data is aggregated over all channels, showing mean and 95\% bootstrap confidence intervals.}

\label{fig:tuning}

\end{figure}

% An example of a floating figure using the graphicx package.

% Note that \label must occur AFTER (or within) \caption.

% For figures, \caption should occur after the \includegraphics.

% Note that IEEEtran v1.7 and later has special internal code that

% is designed to preserve the operation of \label within \caption

% even when the captionsoff option is in effect. However, because

% of issues like this, it may be the safest practice to put all your

% \label just after \caption rather than within \caption{}.

%

% Reminder: the "draftcls" or "draftclsnofoot", not "draft", class

% option should be used if it is desired that the figures are to be

% displayed while in draft mode.

%

%\begin{figure}[!t]

%\centering

%\includegraphics[width=2.5in]{myfigure}

% where an .eps filename suffix will be assumed under latex,

% and a .pdf suffix will be assumed for pdflatex; or what has been declared

% via \DeclareGraphicsExtensions.

%\caption{Simulation Results}

%\label{fig\_sim}

%\end{figure}

% Note that IEEE typically puts floats only at the top, even when this

% results in a large percentage of a column being occupied by floats.

% An example of a double column floating figure using two subfigures.

% (The subfig.sty package must be loaded for this to work.)

% The subfigure \label commands are set within each subfloat command, the

% \label for the overall figure must come after \caption.

% \hfil must be used as a separator to get equal spacing.

% The subfigure.sty package works much the same way, except \subfigure is

% used instead of \subfloat.

%

%\begin{figure\*}[!t]

%\centerline{\subfloat[Case I]\includegraphics[width=2.5in]{subfigcase1}%

%\label{fig\_first\_case}}

%\hfil

%\subfloat[Case II]{\includegraphics[width=2.5in]{subfigcase2}%

%\label{fig\_second\_case}}}

%\caption{Simulation results}

%\label{fig\_sim}

%\end{figure\*}

%

% Note that often IEEE papers with subfigures do not employ subfigure

% captions (using the optional argument to \subfloat), but instead will

% reference/describe all of them (a), (b), etc., within the main caption.

% An example of a floating table. Note that, for IEEE style tables, the

% \caption command should come BEFORE the table. Table text will default to

% \footnotesize as IEEE normally uses this smaller font for tables.

% The \label must come after \caption as always.

%

%\begin{table}[!t]

%% increase table row spacing, adjust to taste

%\renewcommand{\arraystretch}{1.3}

% if using array.sty, it might be a good idea to tweak the value of

% \extrarowheight as needed to properly center the text within the cells

%\caption{An Example of a Table}

%\label{table\_example}

%\centering

%% Some packages, such as MDW tools, offer better commands for making tables

%% than the plain LaTeX2e tabular which is used here.

%\begin{tabular}{|c||c|}

%\hline

%One & Two\\

%\hline

%Three & Four\\

%\hline

%\end{tabular}

%\end{table}

% Note that IEEE does not put floats in the very first column - or typically

% anywhere on the first page for that matter. Also, in-text middle ("here")

% positioning is not used. Most IEEE journals use top floats exclusively.

% Note that, LaTeX2e, unlike IEEE journals, places footnotes above bottom

% floats. This can be corrected via the \fnbelowfloat command of the

% stfloats package.

% if have a single appendix:

%\appendix[Proof of the Zonklar Equations]

% or

%\appendix % for no appendix heading

% do not use \section anymore after \appendix, only \section\*

% is possibly needed

% use appendices with more than one appendix

% then use \section to start each appendix

% you must declare a \section before using any

% \subsection or using \label (\appendices by itself

% starts a section numbered zero.)

%

\appendix[Derivation of $gAT$ Reconstruction]

\indent\indent Assume that in the interval $[0,T]$ we have either 0 or 1 spikes. A spike manifests itself in the comparator output $x(t)$ as a square wave centered at $t\_1$ with width $w\_1$ and height 1. We want to detect whether there is a spike and, if there is one, find its parameters $t\_1$ and $w\_1$. If there is a spike, the comparator output is given by the signal\\

\begin{equation}

x(t) = \left\{

\begin{array}{l l}

1 & \mathrm{if}\,\,\, t\_1-\frac{w\_1}{2} \le t \le t\_1+\frac{w\_1}{2}\\

0 & \mathrm{otherwise}\\

\end{array}

\right.

\end{equation}

Assume the spike is contained completely within the interval, i.e. $0 \le t\_1 - \frac{w\_1}{2}$ and $t\_1+\frac{w\_1}{2} \le T$. Take the first and second integrals $x\_1(t)$ and $x\_2(t)$ of this signal.\\

$\displaystyle x\_1(t) \equiv \int\_0^t x(\tau) d\tau $\\

\begin{equation}

= \left\{

\begin{array}{l l}

0 & \mathrm{if}\,\,\, t < t\_1 - \frac{w\_1}{2}\\

t - (t\_1-\frac{w\_1}{2}) & \mathrm{if}\,\,\, t\_1-\frac{w\_1}{2} \le t \le t\_1+\frac{w\_1}{2}\\

w\_1 & \mathrm{if}\,\,\, t > t\_1 + \frac{w\_1}{2}\\

\end{array}

\right.

\end{equation}

$\displaystyle x\_2(t) \equiv \int\_0^t x\_1(\tau) d\tau$\\

\begin{equation}

= \left\{

\begin{array}{l l}

0 & \mathrm{if}\,\,\, t < t\_1 - \frac{w\_1}{2}\\

\frac{1}{2}[t - (t\_1-\frac{w\_1}{2})]^2 & \mathrm{if}\,\,\, t\_1-\frac{w\_1}{2} \le t \le t\_1+\frac{w\_1}{2}\\

\frac{1}{2}w\_1^2 + w[t - (t\_1+\frac{w\_1}{2})] & \mathrm{if}\,\,\, t > t\_1 + \frac{w\_1}{2}\\

\end{array}

\right.

\end{equation}

\noindent Sample $x\_1(t)$ and $x\_2(t)$ at $t = T$.\\

$y\_1 \equiv x\_1(T) = w\_1$\\

$y\_2 \equiv x\_2(T) = \frac{1}{2}w\_1^2 + w\_1[T - (t\_1+\frac{w\_1}{2})] = w\_1(T-t\_1)$\\

\noindent We can therefore recover the parameters $t\_1$ and $w\_1$ of the spike as follows.\\

$w\_1 = y\_1$\\

$t\_1 = T - \frac{y\_2}{y\_1}$\\

Having 0 spikes in the interval is equivalent to having 1 spike with width $w\_1 = 0$. Therefore, report 0 spikes in the interval if $w\_1 = 0$ (i.e. $y\_1 = 0$). Otherwise, report 1 spike in the interval with parameters $t\_1$ and $w\_1$ as calculated above.

% use section\* for acknowledgement

\section\*{Acknowledgment}

The authors would like to thank...