Automatic objective thresholding to detect neuronal action potentials

Citation

Year
2016

Version
Peer reviewed version (post-print)

Link to publication
TUTCRIS Portal (http://www.tut.fi/tutcris)

Published in
Proceedings of 2016 24th European Signal Processing Conference (EUSIPCO)

DOI
10.1109/EUSIPCO.2016.7760331

Take down policy
If you believe that this document breaches copyright, please contact tutcris@tut.fi, and we will remove access to the work immediately and investigate your claim.
Automatic Objective Thresholding to Detect Neuronal Action Potentials *

Jarno M. A. Tanskanen  
Computational Biophysics and Imaging Group, BioMediTech  
Department of Electronics and Communications Engineering  
Tampere University of Technology, BioMediTech  
Tampere, Finland  
tanskanen@ieee.org

Fikret E. Kapucu  
Department of Pervasive Computing, and Computational Biophysics and Imaging Group, BioMediTech  
Department of Electronics and Communications Engineering  
Tampere University of Technology  
Tampere, Finland  
fikret.kapucu@tut.fi

Inkeri Vornanen, Jari A. K. Hyttinen  
Computational Biophysics and Imaging Group, BioMediTech  
Department of Electronics and Communications Engineering  
Tampere University of Technology  
Tampere, Finland  
inkeri.vornanen@tut.fi, jari.hyttinen@tut.fi

Abstract — In this paper, we introduce a fully objective method to set thresholds (THs) for neuronal action potential spike detection from extracellular field potential signals. Although several more sophisticated methods exist, thresholding is still the most used spike detection method. In general, it is employed by setting a TH as per convention or operator decision, and without considering whether the undetected or spurious spikes. Here, we demonstrate with both simulations and real microelectrode measurement data that our method can fully automatically and objectively yield THs comparable to those set by an expert operator. A Matlab function implementation of the method is described, and provided freely in Matlab Central File Exchange.

Keywords-neuronal action potential; thresholding; spike detection; microelectrode array; field potential

I. INTRODUCTION

In the analysis of electrophysiological measurements from active neuronal cell ensembles, neuronal networks, action potential spike detection [1,2,3] is necessary for many subsequent analyses. Voltage signals to be analyzed are measured with electrodes or microelectrodes in vivo or in vitro [1,2,3,4]. At simplest, spike detection yields the time stamps of the neuronal action potentials observable in the measurements using the particular spike detection method employed. Thereafter, the analysis ranges from simple statistics to highly complex system for spike detection and classification and subsequent analysis. Still, spike detection; microelectrode array; field potential

Usually, spike detection is performed by mere thresholding [1-4,6], in which any signal reaching above (or below for negative spikes) a threshold (TH) is interpreted as a spike. Thresholding is naturally computationally very efficient, and can be performed online during measurements. It would be desirable to estimate the standard deviation of the measurement noise (STDn) and set the TH accordingly to a multiple of STDn, but during a real measurement, the TH usually is set to a multiple of (e.g., 4 to 6.5 times) the signal standard deviation (STD). Some thresholding methods use either negative or positive THs and thus detect only negative of positive spikes [6], whereas other methods utilize symmetric positive and negative THs to capture spikes of both polarities (see e.g., [3]); such a method must be constructed to detect a biphasic spike as one and not two spikes. In general, the THs are set by convention as noted above, and an expert operator assesses the measurements visually to confirm the THs to maximize the number of detected real neuronal spikes, and to minimize the number of spurious spikes, i.e., false spikes detected due to noise. In real measurements, the actual amount of spurious spikes remains unknown, and is usually also not considered in the analysis; all detected spikes are taken as true spikes. Spike waveform analysis might reveal spurious spikes, but is usually not utilized. Also, THs set this way are always subjective.

As spike detection is an important processing step, it has received a lot of attention in the literature, and also complex and elaborate spike detection methods [1-4,6] have been developed. Nevertheless, since thresholding is still more or less the de facto spike detection method for practicing neuroscientists, it would be highly desirable to develop methods to set THs automatically and objectively based on the signals themselves; i.e., without possible operator bias. Automated and objective spike detection is also required for emerging high-throughput electrophysiological measurement systems, e.g., in vitro for toxicology and drug screening. For affordable high-throughput screening, the methods should preferably be of low computational cost.

A new objective thresholding algorithm based on the analysis of spike count histograms is proposed in this paper (Fig. 1). The basics idea of the method was originally described in [7], where its functioning was illustrated with simple 2D and 3D...
simulations and limited real data. Here, we illustrate the work-
ings of the method by analyzing a noise signal, data from com-
putational spiking neuronal network simulations, and real mea-
surements. Also, a few new functionally crucial aspects, such as
optional refractory period, and TH validity checks, have been
implemented in the algorithm. Applying a refractory period is a
standard procedure in spike detection [1,6] (in [6], refractory pe-
riod is called ‘dead time’); in short, no spikes are detected during
the refractory period after a spike has been detected; this causes
also biphasic action potentials to be counted only ones. How-
ever, in a highly active network, spikes from different neurons
may appear arbitrarily close in time, and with a refractory period
in place, such spikes would be lost. The TH validity checks, on
the other hand, can be performed to alarm the operator of possi-
ably artifactual or otherwise suspicious signals.

The full realization algorithm is described in detail, and its
implementation as a Matlab function has now been made freely
available in Matlab Central File Exchange of MathWorks1. The
results shown (Figs. 2-6) have been calculated with the supplied
Matlab function.

As for the spike detection results, as long as the THs used are
equal for the given signal, the actual method of finding the TH
is naturally irrelevant. Thus, since there exist a numerous studies
reviewing and comparing spike detection methods [1-4], also
thresholding against state-of-art methods, such comparisons are
not performed here.

The paper is structured as follow: In Section II, we describe
the proposed thresholding algorithm. In Section III, the data to
be analyzed are described, and the thresholding results are given
in section IV. Further development plans are outlined in Section
V, and conclusions drawn in Section VI.

II. THE THRESHOLDING ALGORITHM

The reasoning behind the proposed algorithm is that contrib-
utions from noise and spikes might be identifiable in spike
count histograms, given that there is a sufficient number of

1 The Matlab function can be downloaded from

spikes reaching sufficiently above the background noise in the
measurements. This principle is realized by the following algo-

First, the global extrema of the input signal are found, and
500 thresholding levels are defined evenly distributed between
them. Next, the input signal is thresholded at every TH. Each
found disjoint signal section exceeding a positive TH or below
a negative TH, is counted as a spike for that TH. Thus, a spike
count histogram (e.g., Fig. 2A) is formed. According to our ex-
perimentation, thresholding at 500 is usually sufficient to form
a histogram for the subsequent analysis. To enhance features of
the histogram, a gradient of the histogram is calculated. Since
the gradient is usually fairly “noisy,” it is smoothed with the run-
ning averager of length $L = 10$ samples (the setting can be over-
ridden by the user): given a sample $g(a)$ of the gradient, where $a$
is a discrete amplitude level, the smoothed gradient is given by
$$g_s(a) = \sum_{b=1}^L g(a+i-1).$$

Next, the extrema of the smoothed gra-
dient are found. The local minimum closest to the global maxi-
mum, appearing at an amplitude smaller than that of the global
maximum, is taken as the negative TH for spike detection (see
Fig. 2A). The positive TH is analogously found at the local max-
imum closest to global minimum appearing at an amplitude
larger than that of the global maximum. This feature was se-
lected for the THs based in the investigations in [7].

The parameter values noted above for the number of thresh-
olding levels, and the length of the smoothing averager are the
defaults in the Matlab function described here, and can be over-
ridden by the user. Other parameters defined in the Matlab func-
tion are:

- Start and stop sample indices for the signal segment based
  on which the THs are determined. By default, the first minute
  of the input signal is analyzed. If the input signal is shorter
  than one minute, the entire signal is analyzed.
- The default refractory period is 1.5 ms.
- Validity check levels for the THs: The default is that the
  found positive TH is to lay at an amplitude between 3 to 10
  times the STD (inclusive the limits), and the found negative
  TH between -3 to -10 times the STD (inclusive). If the found

Figure 1. The proposed thresholding algorithm realized as a Matlab function. The signal shown is a section of the in vivo measurement employed [11,12].
THs do not satisfy the amplitude validity check levels, a warning is displayed.

By default, the function plots in one figure the spike count histogram, its smoothed gradient, and the found THs, and in another figure, the input signal and the found THs, but plotting can be suppressed. Since the units of the input data are unknown, amplitude units are not displayed on the plotted figures. The function outputs the time stamps of the detected spikes in samples, and the found THs in the same units as the input data, and in STDs.

III. DATA

A. Noise Simulations

First, to illustrate the algorithm, we investigated its functioning with noise input signal formed by generating a vector of zero mean Gaussian white noise of variance three to simulate measurement background noise, and adding to every 1000th sample a sample of zero mean Gaussian noise of variance 20. Five minutes of a measurement at 5 kHz sampling frequency were simulated. This resulted in signal seen in Fig. 2B, with the noise approximately between –8 mV to 8 mV. Action potential spikes appearing five times a second, with the largest of them reaching above 60 mV. Similar amplitudes and spiking frequencies could be observed also in real measurements.

B. Computational Neuronal Network Simulations

Here, we investigate and illustrate the functioning of the algorithm and the Matlab function with a simulated signal (Fig. 3B). An action potential signal was formed simulating 40 excitatory and 10 inhibitory integrate-and-fire neurons driven by Gaussian white noise current with short-term plastic synapses in a fully connect network. Produced data simulated a measurement via one microelectrode at 1 kHz sampling frequency. The neuronal cell and network simulations were performed in NEST. Spike waveforms were simulated by sine cardinal functions with random amplitudes. Zero mean white Gaussian noise of variance 0.5 was added to simulate background noise at a level which might be seen in real MEA measurements.

C. In Vivo Data

The algorithm and the Matlab function were also tested with two publicly available recordings, one from mouse cortex in vivo [11,12] (for the experiment description see also [13]), and one from a slice of the freshly-resected human temporal cortex surgically removed to treat complex partial seizures (see, e.g., [14]).

The mouse data is shown in full in Fig. 5B. The same data was employed in [7], but here the effect of large spikes (occurring between 3.2 and 4.6 s in Fig. 5) on the thresholding is illustrated in contrast with thresholding a normally spiking signal (Fig. 5 vs. Fig. 4, respectively). The in vitro human data is shown in Fig. 6B.

IV. RESULTS

A. Noise Signal Thresholding Results

Running the algorithm on the noise input signal revealed a characteristic histogram and its gradient (Fig. 2A). The features for setting the THs were quite small, but still detectable. Observing the signal and the background noise (Fig. 2B), the THs are seen to be positioned alike an expert operator might have placed them. The THs in Fig. 2 were found approximately at -5.6 and 5.7 times the STD.

![Figure 2](image-url)

Figure 2. The results of thresholding the noise signal. (A) Spike count histogram (black), smoothed gradient of the histogram (red), the found negative TH (magenta), and the found positive TH (green), with detailed view to the gradient features in the inset. (B) The algorithm input signal (black), the added Gaussian noise by itself (cyan), and the found THs.

B. Computational Neuronal Network Simulation Thresholding Results

The results of thresholding the signal produced with the computational neuronal network are shown in Fig. 3. In Fig. 3B, it is seen that the proposed algorithms worked well also with a signal that was not zero mean, here exhibiting spikes of far mostly negative polarity, which resulted in a skewed spike count histogram. Still, our principle of TH detection worked well, although possible small amplitude positive spikes may have gone undetected. The THs were found approximately at -1.6 and 1.7 times the STD of the signal, and warning was issued. Observing Fig. 3B, it is seen that in this case the warning caused by the default TH validity check levels was overly cautious.

C. In Vivo Data Thresholding Results

The results of thresholding a normal spiking section of the in vivo measured mouse signal are shown in Fig. 4. The THs were found approximately at -2.8 and 2.8 times the STD, and a warning was issued. An expert operator might have placed the THs alike set by our algorithm: nicely around the visually observable background noise amplitude band.

Figure 3. Thresholding results for the signal generated with the computational neuronal network model. (A) Spike count histogram, its gradient, and the found THs with details in the inset. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.

Figure 4. The thresholding results for the section of the *in vivo* measurement signal with normal spike contributions. (A) Spike count histogram, its gradient, and the found THs. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.

Figure 5. The thresholding results for the section of the *in vivo* mouse measurement signal with sections of both normal spike contributions and artifacts. (A) Spike count histogram, its gradient, and the found THs. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.

Figure 6. The thresholding results for the section of the *in vitro* human measurement signal. (A) Spike count histogram, its gradient, and the found THs. The spike count axis has been raised to better observe the features in the inset. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.
The results of thresholding the section of the in vivo measurement exhibiting both normal brain activity and artifacts are shown in Fig. 5, where it is seen that the artifact, although appearing only during less than half of the duration of the section concerned, has caused the THs to be found further away in amplitude from the real spiking neuronal signal. Here, the THs were found approximately at -6.0 and 5.5 times the STD, which is comparable to an expert operator setting the THs according to the convention. Nevertheless, the artifact caused our method to find too high THs.

The thresholding results of the in vitro human measurement are shown in Fig. 6. The THs were found approximately at -2.7 and 1.9 times the STD, and a warning was shown. In this case, the algorithm placed THs very tightly around the visually observable background noise.

V. DISCUSSION

We have demonstrated that our algorithm can behave much like an expert operator. Since no ground truth about the true spikes is available with real measurements, achieving human operator-like performance has here been taken as the thresholding-quality criterion. However, experimentation has led us to the following observations, which call for further development of the algorithm and Matlab function. In its current form, the algorithm and function are fully usable, but expert supervision is advisable to assess the appropriateness of the default parameter values for the signal at hand.

A. The Effects of Parameter Values

In further experimentation, the results of the algorithm were seen to be greatly affected by the length of the smoothing averager and by the length of the signal segment analyzed. Based on the experimentation, we arrived at the default parameter values for adequate performance in several cases, but it would be beneficial to automatically adapt them to the signal at hand.

A shorter averager may help in catching a feature in a gradient without strong features. On the other hand, in a case with clear “shoulders” in the spike count histogram, indicating the amplitude region right above and below the background noise level, a longer averager may be useful to find clear smooth gradient features for setting the THs.

B. Planned Developments

The published Matlab function is the version 1.0, i.e., the first fully functioning implementation of the proposed thresholder. To develop the Matlab function further, the following functionality are planned to be implemented in a future version: 1. A possibility to automatically select of the section of the input signal to be analyzed for setting the THs for the entire input signal would be desirable. The selected section should exhibit an appropriate amount of spiking activity with sufficiently high amplitudes for finding THs reliably, regardless of possibly periodical inactivity or excessive noise. 2. Power line noise and baseline drift detection and suppression would also be desirable, since power line noise or baseline drift will either drive the THs away from the otherwise desirable levels, or cause an excessive number of detected spurious spikes. 3. During some measurements, background noise level may change dramatically, thus, adaptive thresholding in a running window would be beneficial.

VI. CONCLUSIONS

We have proposed, demonstrated, and implemented an objective and automatic method to determine action potential detection amplitude THs for neuronal electrophysiological measurements. A Matlab function is provided, and further developments outlined. We hope that this work will find its way to the automated analyzes of electrophysiological measurements, thus in part paving the way to efficient utilization of high-throughput electrophysiological methods.

REFERENCES


}