

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 either 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 time stamps are often the only input (e.g., [5]) to the neuronal activity and network analyses algorithms. Thus, the trustworthiness of the results, and the quality of the research is ultimately dictated by the quality of spike detection.

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 (STD_n) and set the TH accordingly to a multiple of STD_n, 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 or 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

* The work of J. M. A. Tanskanen has been supported by Jane and Aatos Erkko Foundation, Finland, under the project Biological Neuronal Communications and Computing with ICT. The work of F. E. Kapucu has been supported by the Academy of Finland under the project Bio-integrated Software Development for Adaptive Sensor Networks, project number 278882, by the Human Spare Parts Project funded by Tekes – the Finnish Funding Agency for Innovation, and by Ella and Georg Ehrnrooth Foundation, Finland. The works of J. M. A. Tanskanen and F. E. Kapucu have also been supported by the 3DNeuroN project in the European Union's Seventh Framework Programme, Future and Emerging Technologies, grant agreement n°296590. The work of I. Vornanen has been funded by the Human Spare Parts Project funded by Tekes – the Finnish Funding Agency for Innovation.

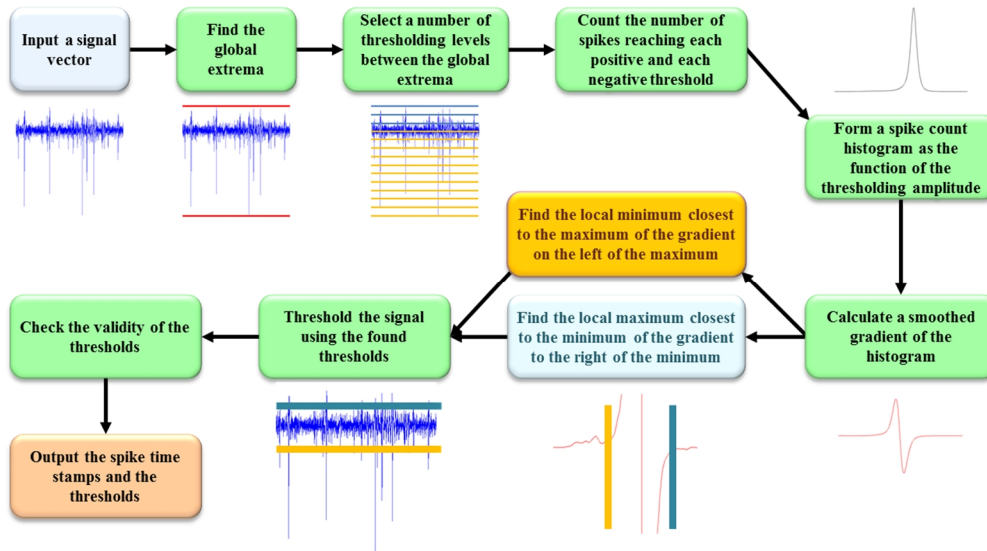


Figure. 1. The proposed thresholding algorithm realized as a Matlab function. The signal shown is a section of the *in vivo* measurement employed [11,12].

simulations and limited real data. Here, we illustrate the workings of the method by analyzing a noise signal, data from computational spiking neuronal network simulations, and real measurements. 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 period 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. However, 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 possibly 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 contributions from noise and spikes might be identifiable in spike count histograms, given that there is a sufficient number of

spikes reaching sufficiently above the background noise in the measurements. This principle is realized by the following algorithm illustrated in Fig. 1.

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 experimentation, 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 running averager of length $L = 10$ samples (the setting can be overridden by the user): given a sample $g(a)$ of the gradient, where a is a discrete amplitude level, the smoothed gradient is given by $g_s(a) = \sum_{l=1}^L \frac{1}{L} g(a+l-1)$. Next, the extrema of the smoothed gradient are found. The local minimum closest to the global maximum, 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 maximum closest to global minimum appearing at an amplitude larger than that of the global maximum. This feature was selected for the THs based in the investigations in [7].

The parameter values noted above for the number of thresholding levels, and the length of the smoothing averager are the defaults in the Matlab function described here, and can be overridden by the user. Other parameters defined in the Matlab function 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

¹ The Matlab function can be downloaded from <http://se.mathworks.com/matlabcentral/fileexchange/?term=id%3A55227>.

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 [8], and 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 [9] 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 [10]. 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 cortex² 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.

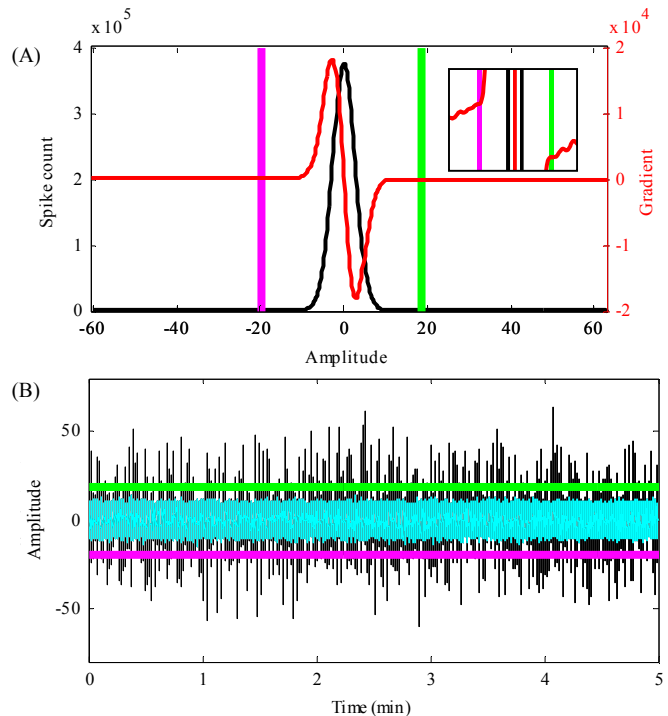


Figure 2. The results of thresholding the noise signal. (A) Spike count histogram (black), smoothed gradient of the histogram (red), the found negative TH (magenta), and the found positive TH (green), with detailed view to the gradient features in the inset. (B) The algorithm input signal (black), the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

² Data available in Carmen (<http://www.carmen.org.uk/>, login required) as a Matlab file: 280808 029 Export Export.mat, <https://portal.carmen.org.uk/#link=URN:LSID:portal.carmen.org.uk:metadata:49>.

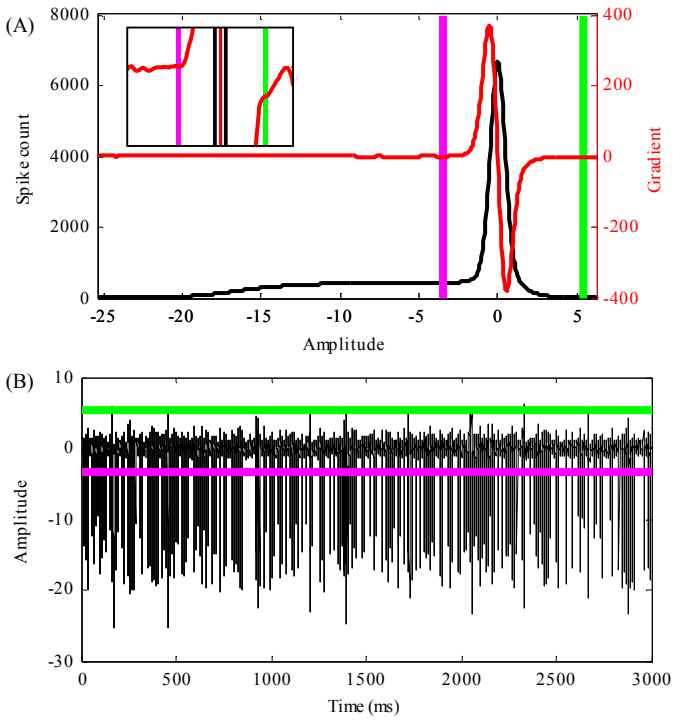


Figure 3. Thresholding results for the signal generated with the computational neuronal network model. (A) Spike count histogram, its gradient, and the found THs with details in the inset. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.

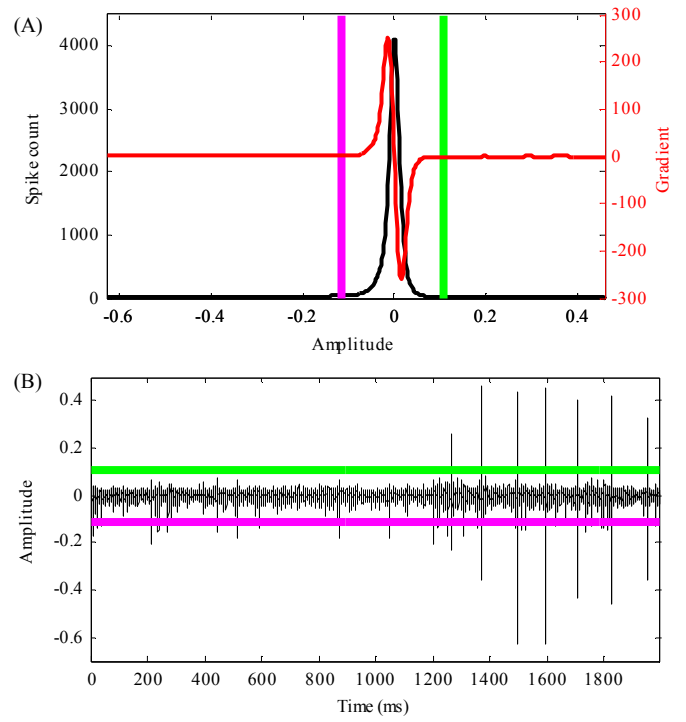


Figure 5. The thresholding results for the section of the *in vivo* mouse measurement signal with sections of both normal spike contributions and artifacts. (A) Spike count histogram, its gradient, and the found THs. (B) The algorithm input signal, and the found THs. For the colors see the caption of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

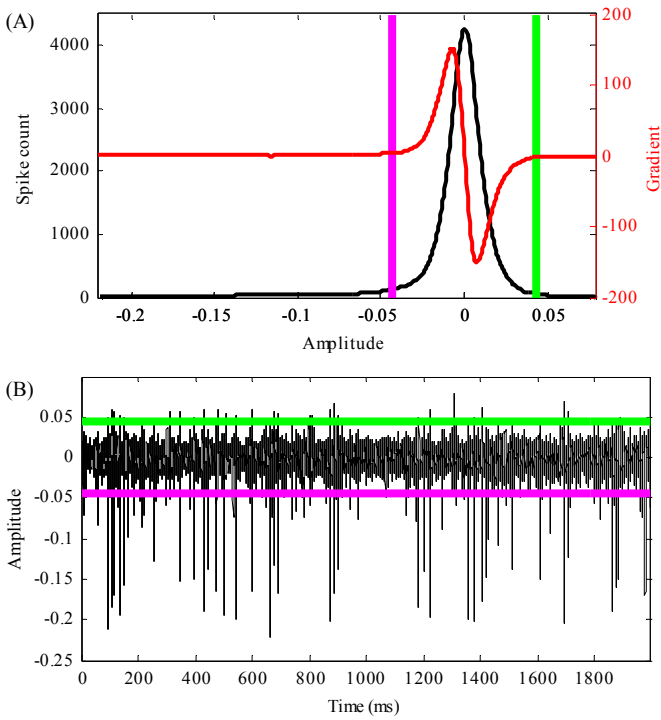


Figure 4. The thresholding results for the section of the *in vivo* measurement signal with normal spike contributions. (A) Spike count histogram, its gradient, and the found THs. (B) The algorithm input signal, and the found THs. For the colors see the caption of Fig. 2.

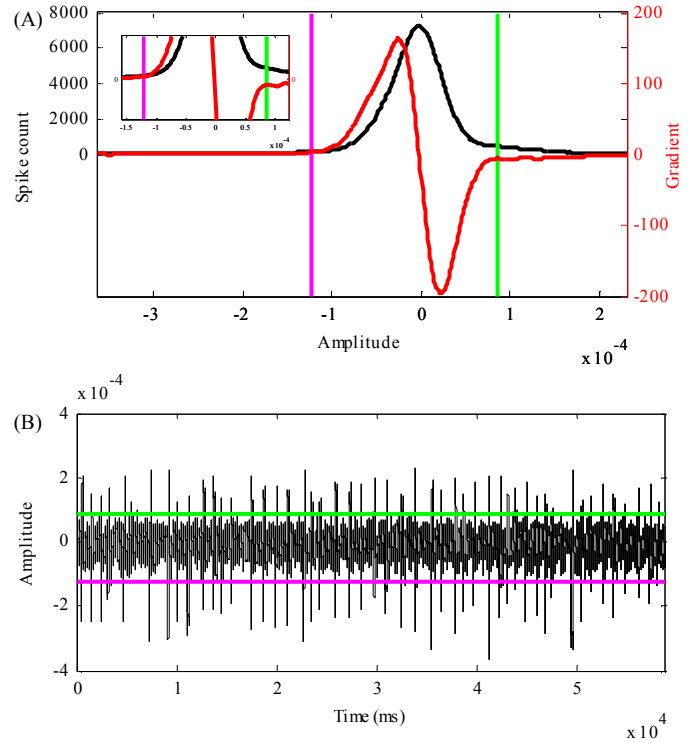


Figure 6. The thresholding results for the section of the *in vitro* human measurement signal. (A) Spike count histogram, its gradient, and the found THs. The spike count axis has been raised to better observe the features in the inset. (B) The algorithm input signal, and the found THs. For the colors 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 well 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

- [1] M. S Lewicki, “A review of methods for spike sorting: the detection and classification of neural action potentials,” *Netw. Comput. Neural Syst.*, vol. 9, no. 4, 1998, pp. R53-R78. http://dx.doi.org/10.1088/0954-898X_9_4_001
- [2] M. Aghagolzadeh, A. Mohebi, and K. G. Oweiss, “Sorting and tracking neuronal spikes via simple thresholding,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, July 2014, pp. 858-869. <http://dx.doi.org/10.1109/TNSRE.2013.2289918>
- [3] E. Biffi, D. Ghezzi, A. Pedrocchi, and G. Ferrigno, “Development and validation of a spike detection and classification algorithm aimed at implementation on hardware devices,” *Comput. Intell. Neurosci.*, vol. 2010, article 659050. <http://dx.doi.org/10.1155/2010/659050>
- [4] S. B. Wilson and R. Emerson, “Spike detection: a review and comparison of algorithms,” *Clin. Neurophysiol.*, vol. 113, no. 12, Dec. 2002, pp. 1873-1881. <http://www.sciencedirect.com/science/article/pii/S1388245702002973>
- [5] F. E. Kapucu, J. M. A. Tanskanen, J. E. Mikkonen, L. Ylä-Outinen, S. Narkilahti, and J. A. K. Hyttinen, “Burst analysis tool for developing neuronal networks exhibiting highly varying action potential dynamics,” *Front. Comput. Neurosci.*, vol. 6, article 38, June 2012. <http://dx.doi.org/10.3389/fncom.2012.00038>
- [6] Multi Channel Systems MCS GmbH, *MC_Rack Manual*, 2013, pp. 142-144. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/MC_Rack_Manual.pdf
- [7] J. M. A. Tanskanen, F. E. Kapucu, and J. A. K. Hyttinen, “On the threshold based neuronal spike detection, and an objective criterion for setting the threshold,” in *Proc. 7th Annual International IEEE EMBS Conference on Neural Engineering*, Montpellier, France, Apr. 2015, pp. 1016-1019. <http://dx.doi.org/10.1109/NER.2015.7146799>
- [8] Multi Channel Systems MCS GmbH, *Microelectrode Array (MEA) Manual*, 2013, pp. 51. http://www.multichannelsystems.com/sites/multichannelsystems.com/files/documents/manuals/MEA_Manual.pdf
- [9] M. Tsodyks, A. Uziel, and H. Markram, “Synchrony generation in recurrent networks with frequency-dependent synapses,” *J. Neurosci.*, vol. 20, no. 1, Jan. 2000, RC50. <http://www.jneurosci.org/content/20/1/RC50>
- [10] M.-O. Gewaltig and M. Diesmann, “NEST (Neural Simulation Tool),” *Scholarpedia*, vol. 2, no. 4, 2007, pp. 1430. <http://dx.doi.org/10.4249/scholarpedia.1430>
- [11] N. Li, C. R. Gerfen, and K. Svoboda, “Extracellular recordings from anterior lateral motor cortex (ALM) neurons of adult mice performing a tactile decision behavior,” data deposited in *CRCNS.org*, 2014. <http://dx.doi.org/10.6080/K0MS3QNT>
- [12] N. Li, T.-W. Chen, Z. V. Guo, C. R. Gerfen, and K. Svoboda, “A motor cortex circuit for motor planning and movement,” *Nature*, vol. 519, Mar. 2015, pp. 51-56. <http://dx.doi.org/10.1038/nature14178>
- [13] Z. V. Guo, N. Li, D. Huber, E. Ophir, D. Gutnisky, J. T. Ting, G. Feng, and K. Svoboda, “Flow of cortical activity underlying a tactile decision in mice,” *Neuron*, vol. 81, no. 1, Jan. 2014, pp. 179-194. <http://dx.doi.org/10.1016/j.neuron.2013.10.020>
- [14] A. K. Roopun, J. D. Simonotto, M. L. Pierce, A. Jenkins, C. Nicholson, I. S. Schofield, R. G. Whittaker, M. Kaiser, M. A. Whittington, R. D. Traub, and M. O. Cunningham, “A nonsynaptic mechanism underlying interictal discharges in human epileptic neocortex,” *PNAS*, vol. 107, no. 1, Jan. 2010, pp. 338-343. <http://dx.doi.org/10.1073/pnas.0912652107>