Title
Peak Detection to Count Gold Nanoparticles Translocations in Nanopipette

Author
Cheng, Jie

Publication Date
2018

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

PEAK DETECTION TO COUNT GOLD NANOPARTICLES TRANSLOCATIONS IN NANOPIPETTE

A thesis submitted in partial satisfaction
of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Jie Cheng

March 2018

The Thesis of Jie Cheng
is approved:

Professor Hamid Sadjadpour, Chair

Professor Nader Pourmand

Professor Yu Zhang

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Peak Detection to Count Gold Nanoparticles Translocations in Nanopipette

by

Jie Cheng

This research paper presents a detailed documentation of a method designed to count the number of nanoparticle translocations. The report discusses fundamental theory of signal processing techniques and explanation of practical implementation in MATLAB. Experimental data was analyzed and the results are presented and discussed thoroughly.

Acknowledgments

I would like to express my gratitude to my advisor Professor Hamid Sadjadpour for the helpful instructions and suggestions through the completion of my thesis. Furthermore, I would like to thank Professor Nader Pourmand and his group for providing the experimental data and important biological information. Also, thanks to Professor Tom O'Haver that I learned a lot about signal processing from his website. I wrote my own MATLAB program by borrowing some of his ideas in the MATLAB script findpeaks. Finally, I would like to thank my group members Andres and Chenjiang. We worked together and learned from each other.

Chapter 1

Introduction

Detection of nanoparticles moving through a nanopipette has been studied [3] in recent years. Metal nanoparticles (NPs) are particularly important because of their applications in catalysis, sensors, and healthcare. This research focuses on using signal processing and detection theory techniques to accurately count the number of NPs moving through a nanopipette.

For the experiment, there was a simple model. In the beginning, gold nanoparticles were in the buffer solution inside the nanopipette. Outer part of the device was filled with the same buffer solution and with an electrode inside. Figure 1.1 shows the experiment. During the experiment, the current generated in the circuit was recorded.

Every time a gold nanoparticle passes through the nanopipette, the current would increase sharply for a short period of time. The peak generated by this experiment is called resistive pulse and the objective of this research is to accurately count the number of peaks. Figure 1.2 demonstrates one of these peaks.

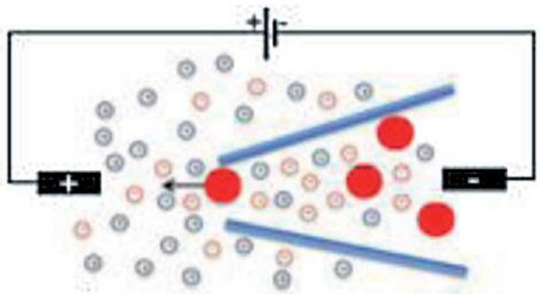

Figure 1.1: A Simple Model of the Experiment [3]

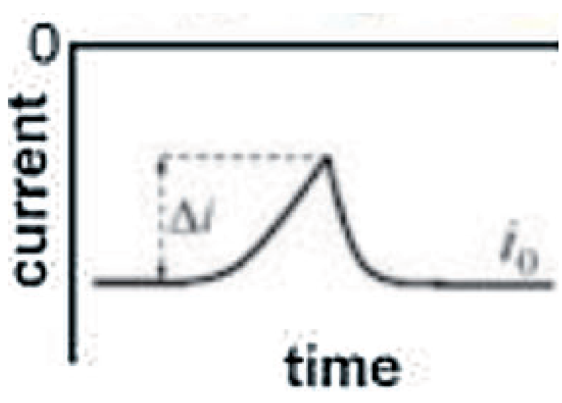

Figure 1.2: Resistive Pulse of Current [3]

Chapter 2

Signal Processing Method

2.1 First Derivative Computation

For the detection of peaks, the general idea is to compute the first derivative of the signal and then look for the downward zero crossing points in the first derivative. "Downward" can ensure (see Figure 2.1) that these points in the first derivative correspond to peaks in the original current signals.

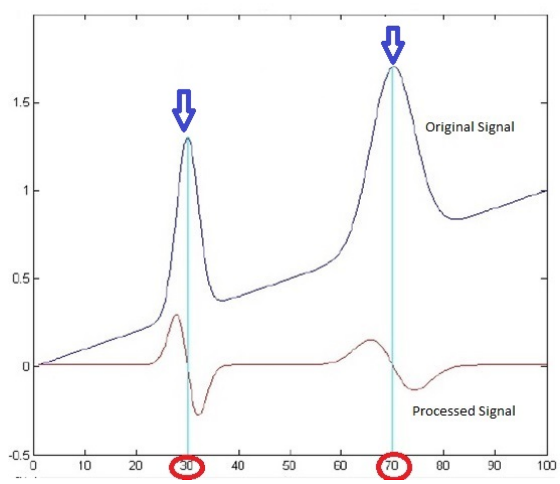

Figure 2.1: Using Downward Zero Crossing to Find Peaks

2.2 Smoothing Algorithm

Since recorded signal corresponding to translocation of nanoparticles has noise, simple first derivative cannot provide a reliable solution. In order to reduce the negative effect of noise, we use a smoothing algorithm. Therefore, we used a three-point unweighted sliding average algorithm to smooth the signal.

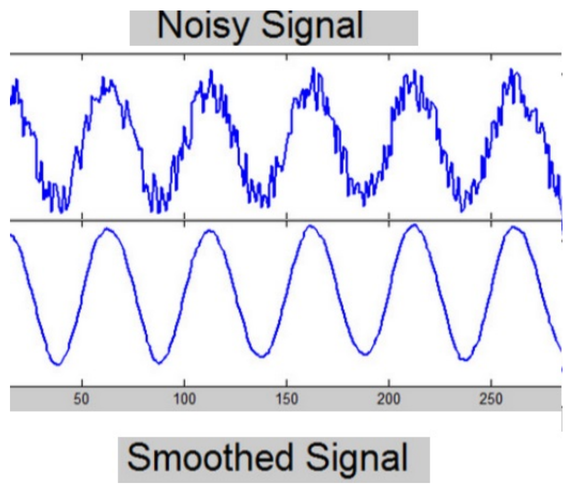

Figure 2.2: Smooth Signals

Smoothing can potentially reduce the adverse effect of noise. However, it can also reduce the amplitude of peaks which makes it harder to detect those peaks. For this reason, instead of smoothing the original signal, we decided to smooth the signal that associates to the derivative of the original signal. Note that this approach does not change the value of the original signal measurements. For peak detection, we only need to find the downward zero crossing of the derivative of the signal and not the peak. The smoothing algorithm used here is a shift and multiply technique and the value of each point is replaced by the new value given by the formula of the smoothing algorithm. Then we used the original idea to find peaks in the smoothed first derivative signal.

2.3 Best Linear Unbiased Estimator

After smoothing the first derivative and finding the downward zero crossing points, we could count the number of peaks in original current signal. To further improve the detection method, we used a minimum threshold such that if the original signal is larger than the threshold, then it is counted as a peak. Otherwise, the zero crossing of the derivative signal is not counted as a peak. In order to accurately select an accurate amplitude threshold for peak detection, Best Linear Unbiased Estimator (B.L.U.E.) technique was selected which is an effective method for this purpose.

The Best Linear Unbiased Estimator (B.L.U.E.) [1] restricts the relationship between the estimator (which estimates the amplitude threshold) and the data (which in this experiment is the recorded current signal) to be linear. Let denote Y as data, θ as the amplitude threshold and $\hat{\theta}$ as the estimated amplitude threshold, then we have

$$\hat{\theta} = AY, \tag{2.1}$$

where A is row vector $[a_0, a_1, ..., a_{N-1}]$ and Y is column vector $[x_0, x_1, ..., x_{N-1}]^T$. Let

$$Y = H\theta + W, \tag{2.2}$$

where H is a known matrix and W is assumed to be white noise with zero mean and some constant variance. There are two important points to consider. First, since B.L.U.E. is an unbiased estimator, the expectation of $\hat{\theta}$ must be $E(\hat{\theta}) = E(AY) = E[A(H\theta + W)] = \theta$. This equality can be achieved if $AH = I$, where I is the identity matrix. The second point to consider is to ensure that B.L.U.E. has minimum variance subject to the unbiased constraint $AH = I$. The variance for $\hat{\theta}$ is ACA^T, where C is the covariance matrix of Y. Using Lagrange Multiplier to deduce the formula for B.L.U.E. under the constraint $AH = I$, and the expression is [1]

$$\hat{\theta} = \hat{\theta}_{BLUE} = (H^T C^{-1} H)^{-1} H^T C^{-1} Y. \tag{2.3}$$

For this experiment, we can simplify the expression as

$$Y = \theta + W. \tag{2.4}$$

This simplification assumes that θ is constant for every time segment (based on the condition of our data) and the noise is zero mean with variance σ. Based on this assumption, B.L.U.E. is given by

$$\hat{\theta} = \bar{Y}, \tag{2.5}$$

which is just the average of the current. Additional improvement is made by averaging the current value of all possible peaks that are found by our peak detection method rather than averaging the value of all points in the original current signal. This approach ensures that only the peaks were utilized in order to estimate the peak threshold.

Chapter 3

MATLAB Implementation

3.1 Workflow

Next figure demonstrates the workflow of the main peak detection method.

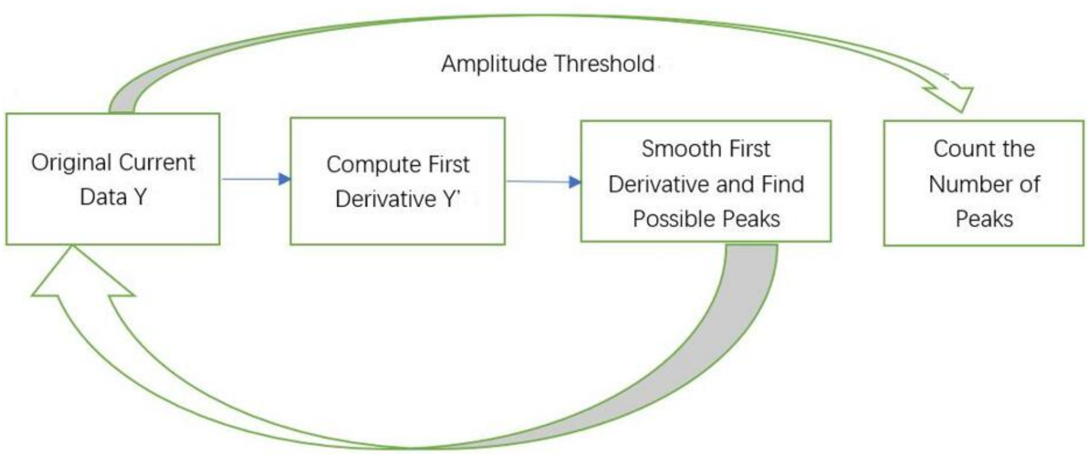

Figure 3.1: Workflow

3.2 Computation of First Derivative

Since the sampled data is discrete, we adopted a simple formula to compute the first derivative. Suppose we want to compute the first derivative for sample N. Then its derivative can be computed by dividing the difference between its two adjacent sample points by the time interval. The mathematical expression is

$$Y_N{}' = \frac{Y_{N+1} - Y_{N-1}}{X_{N+1} - X_{N-1}}. \tag{3.1}$$

Since the sample rate is a constant, the time interval between adjacent points is also a constant. Equation (3.1) can be rewritten as

$$Y_N{}' = \frac{Y_{N+1} - Y_{N-1}}{2\Delta X}. \tag{3.2}$$

According to previous formula, we did not expect the value of the first derivative for a peak to be exactly equal to zero. Our computation on the data confirms this observation. Therefore, we could not find the peaks directly by searching for downward zero crossing points. Fortunately, MATLAB allows us to compare the sign of the values. After comparing the sign of the first derivative, we can decide which point correspond to a peak (This idea was taken from the MATLAB script "findpeaks"). Specifically, if the sign of one point is larger than that of the point after it, this point may correspond to a peak because the sign could be either positive or negative. The change of the sign indicates the appearance of a peak. In addition to consider the change of the sign, we also considered the tendency of increasing to make our detection result more precise. For example, if we find that the sign of the first derivative for a point j is larger than that of the point $j+1$, we continue our detection by checking the sign of the first derivative for the point $j-1$. If the sign is also positive, we conclude that there is a tendency of increasing in the original current signals. Then we count the point j as a peak.

3.3 3-point Unweighted Sliding-Average Smoothing Algorithm

The smoothing algorithm can be expressed by the following equation.

$$Y1_N{}' = \frac{Y_{N-1}{}' + Y_N{}' + Y_{N+1}{}'}{3} \tag{3.3}$$

The derivative of point N is replaced by the average derivative of point $N-1$, N and $N+1$. It is often useful to apply a smoothing operation more than once, that is, to smooth an already smoothed signal, in order to improve the smoothing process [2]. If we smooth the first derivative the second time, the result is

$$Y2_N{}' = \frac{Y1_{N-1}{}' + Y1_N{}' + Y1_{N+1}{}'}{3}, \tag{3.4}$$

where $Y1_{N-1}{}'$, $Y1_{N-1}{}'$, and $Y1_{N-1}{}'$ are the results after implementing the smoothing algorithm once. For our MATLAB program, we implemented this algorithm three times to get a better smoothing effect.

3.4 Final Result for the Number of Peaks

By implementing the smoothing algorithm and peak detection method, we counted the number of peaks and then compared these peaks with our amplitude threshold estimator (B.L.U.E.). If the peaks are higher than the threshold, then we count them as valid peaks. Otherwise, we discard the rest.

Chapter 4

Results and Analysis

4.1 Experimental Set Up and Preparation

To detect the translocations of AuNPs, we used two different concentration solutions of AuNPs—300K and 30K. PBS was a solution without AuNPs which was used to compare with the two experimental groups. Six different experiments (NP1,NP2,NP3,NP4,NP5,NP6) were conducted. After conducting the experiments, all the data were recorded by the device and saved as Excel files. Here are some of the specific parameters.

Experimental set up for AuNPs injection using Electrochemical Analyzer

- **Electrochemical method:** **Amperometry (measuring current vs. time at fixed electric potential)**
- **Applied potential (Ep):** **-6V vs. Ag/AgCl as reference electrode**
- **Run Time (sec) =** **10**
- **Working Elevtrode :** **Ag wire coated with AgCl**
- **Reference Electrode:** **Ag/AgCl**

Figure 4.1: Experimental Set Up for AuNPs Injection Using Electrochemical Analyzer

Preparation of Gold Nanoparticle (AuNP) Solution

- **We use 5μL of AuNP backfilling the Nanopipette (NP);**
- **5μL * 1.0×10^{-4}M = 5×10^{-10} moles in 5μL AuNP solution;**
- **(5×10^{-10}) * (6×10^{23}) = 30 * 10^{13} = 3×10^{14} particles**
- **3×10^{14} particles = 3×10^{5} $\times 10^{9}$**
- **Dilute the stock solution 1 billion times to end up with 300,000 AuNP, in 5 μL solution**

Figure 4.2: Preparation of Gold Nanoparticle (AuNP) Solution

Table 4.1: Findpeak's Result

total	300K	30K	Blank(PBS)
NP1	1474	1749	1877
NP2	1484	1867	1824
NP3	1423	1754	1689
NP4	1576	1676	1794
NP5	1515	1776	1717
NP6	1571	1682	1641

4.2 Comparing Different Method's Result

The experimental files were analyzed by four methods. These four methods were: MATLAB script findpeaks, Bio Group's method, Main method in this thesis and counting the peaks by hand. The results are analyzed separately and then compare with each other.

4.2.1 MATLAB Function Findpeaks's Result

The result of MATLAB script—findpeaks is displayed in Table 4.1. According to the result, the number of peaks continues to increase from 300K to PBS. As can be observed, as the number of AuNP particles increased, the number of peaks detected by MATLAB is reduced. These results clearly demonstrate that MATLAB command for peak detection is unable to accurately count the number of peaks.

4.2.2 Bio Group's Result

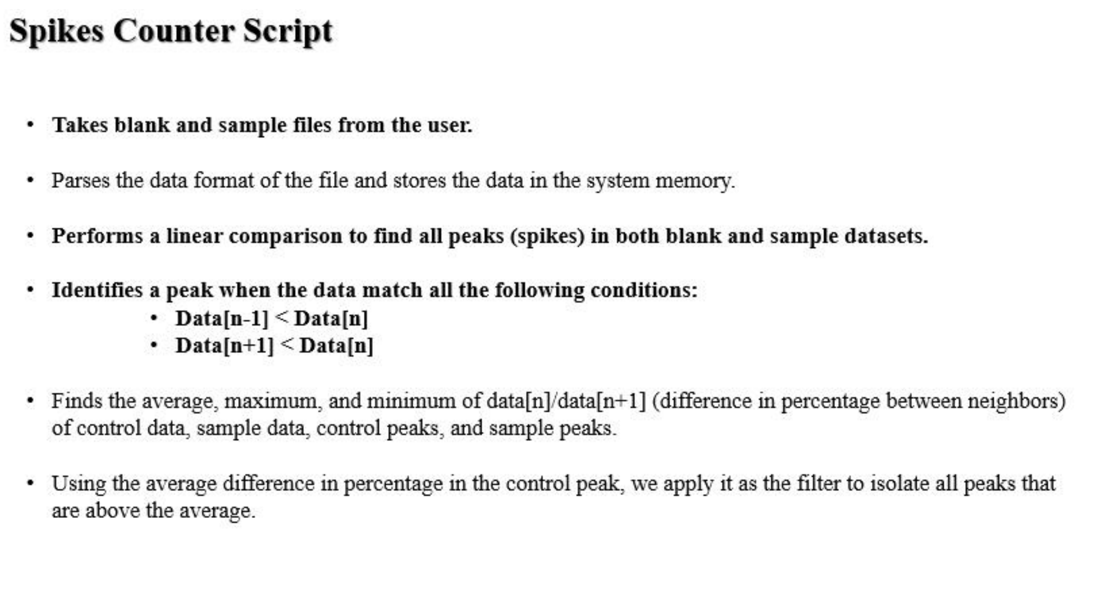

Figure 4.3: Bio Group's Method

The method used by bio group to count the number of peaks also shows inconsistencies in the result. Normally, one would expect to see higher number of peaks in the 300K experiment followed by 30K and PBS. However, the results in Table 4.2 shows higher number of peaks for 30K experiment than 300K one. Therefore, we conclude that these results are not reliable.

Table 4.2: Bio Group's Result

Bio Group	300k		30K		Blank(PBS)	
	Spikes	Total	Spikes	Total	Spikes	Total
NP1	274	797	340	1171	691	1496
NP2	251	738	518	1255	621	1357
NP3	126	745	428	1114	597	1529
NP4	394	951	427	990	487	1179
NP5	339	839	505	1136	428	1024
NP6	356	856	472	1051	398	974

Table 4.3: Main Method's Result

Main Method	300k		30K		Blank(PBS)	
	Peaks	Possible	Peaks	Possible	Peaks	Possible
NP1	403	782	331	710	297	784
NP2	415	759	337	751	346	729
NP3	422	768	368	752	237	665
NP4	342	759	309	755	353	802
NP5	413	757	323	769	377	782
NP6	398	747	313	762	364	773

4.2.3 Result of Main Method

The counting results from our proposed peak detection demonstrate a consistent outcome for all six experiments. The highest number of peaks were detected in 300K solution while the lowest number of peaks in PBS solution as expected. We also noticed that the number of peaks in PBS solution is considerable which implies that our peak counting technique is not perfect. In theory, the number of peaks in PBS solution should be much smaller than the other two solutions.

4.2.4 Hand count Result

Further, we did hand counting of NP1 and NP2 experiments and averages the number of peaks that were calculated independently by three people. It turned

Table 4.4: Hand Count's Result

Hand Count	Andres	Jie	Chenjiang	average
NP1	521	450	339	437
NP2	463	383	309	385

out that the average number of peaks is very close to the technique introduced in this thesis.

In summary, all these four methods have shortcomings. Future work should focus on better incorporating the experimental circuit into peak detection. Unfortunately, many of the details of experimental circuits in the devices used were not known to us due to lack of information from manufacturer of the experimental devices. Future work should focus on using more accurate information regarding experimental devices in order to be able to design better peak detection techniques for this type of applications.

Chapter 5

Conclusion

5.1 Conclusion

The method presented in this thesis utilized some signal processing techniques to detect the peaks. In comparison and considering all known facts, the proposed method worked better in peak detection compared to some other existing approaches. However more background knowledge is needed, such as the issues brought up in previous sections. Noise will still be an issue and better formulation of noise can result in better smoothing algorithms.

Bibliography

[1] Steven M.Kay. *Fundamentals of Statistical Signal Processing:Estimation Theory*. Prentice Hall PTR, 1993.

[2] Tom O'Haver. Smoothing. `https://terpconnect.umd.edu/~toh/spectrum/Smoothing.html`, December 2017. Accessed on 2017-12-29.

[3] Yixian Wang, Huijing Cai, and Michael V.Mirkin. Delivery of Single Nanoparticles from Nanopipettes under Resistive-Pulse Control. *ChemElectroChem*, 8 Dec.2014.