University of California

Santa Cruz

Signal Processing: Peak Detection

A thesis submitted in partial satisfaction

of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

by

Chenjiang Zhao

June 2018

The thesis of Chenjiang Zhao

is approved:

Professor Hamid Sadjadpour

Professor Nader Pourmand

Professor Farid Dowla

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of content

List of Figures

Abstract of signal processing: peak detection, Chenjiang Zhao

This thesis aims to count the translocations of the nanoparticles through the nanopipette. We design the MATLAB scripts to count, using the data from excel files which are provided by the biology group. We implement some signal processing theorems to analyze the data, and use MATLAB to verify the theories.

Acknowledgement

I can’t finish this great work without the help of many people, including students and faculties. Their contributions are sincerely appreciated and acknowledged. I would like to express my great appreciation to the following:

Professor Hamid Sadjadpour, Professor Nader Pourmand, Professor Farid Dowla, Mr. Jie Chen, Mr. Andres Aranda for their support during the whole process of the research.

I really appreciate the help I received from everyone.

Introduction

Monitoring, controlling, and analyzing the nanoparticles are attracting a lot of attention from academic and industry groups. One of the important methods of monitoring is counting the number of nanoparticles that pass through a nanopipette. Using the resistive pulse sensing to realize this idea is a good way to analyze the process, combining with the use of the software.

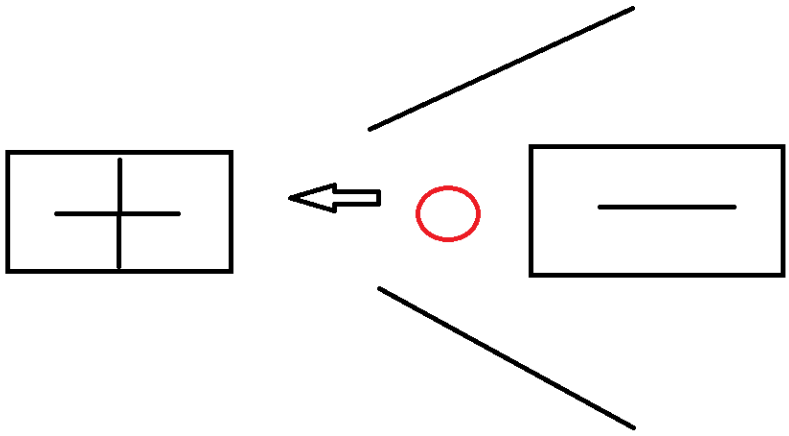


Figure 1 This figure represents the movement of the nanoparticles in the experiments

The idea is when the particle enters the pore, it will cause a current change when the nanoparticles pass through a nanopipette. A resistive pulse is produced by the particle ejection from the pipette to the external solution. So this means the spike shows off every time one particle enters the external solution from the nanopipette.

The UCSC Biology research group directed by Professor Nadar Pourmand simulates this experiment and provide the data. This paper will use the data provided by this group to derive signal processing techniques that can count the number of particles passing through the nanopipette. We use MATLAB to verify our results.

Peak Detection Theory

As we mention above, the spike represents a possible translocation. So the problem becomes counting the number of spikes due to translocation of nanoparticles. Finding the true peaks in the presence of the noise is the main goal.

Signal processing

If we want to count the spikes, we need to know the characters of them. And one important character of spikes is the first derivative of a peak has a downward-going zero-crossing at the peak maximum. But since our signal derived from our data is discrete and there is noise in the signal, we will have some false peaks. One simple and efficient way to get rid of the noise is smoothing. According to our research, the most effective smoothing algorithms are using 'shift and multiply" method, which can be stated as "averaging some of the adjacent points". Since the noise have a higher frequency than the true signal, the idea of smoothing is basically a low pass filter. Taking derivative of the observed data is one way of smoothing the signal. Therefore, there would be less number of false positives.

Our smoothing method is called 7-point "pseudo-Gaussian" which is a three passes of 3-point rectangular smooth, where the rectangular smoothing is

$$S_j = \frac{Y_{j-1}+Y_j+Y_{j+1}}{3}.$$

Repeating the basic 3-point rectangular smoother is very useful technique in most cases.

Detection and Estimation

Smoothing improves the counting but we attempted to improve our technique further. At this point, most of the peaks we find are legitimate peaks. To eliminate those false peaks, we need to use the theory of detection and estimation.

There are many methods of detection and estimation, such as general minimum variance unbiased estimation, best linear unbiased estimation(B.L.U.E), and maximum likelihood estimation. After some investigation, we decided to use B.L.U.E., since other methods require more information about the data in order to implement them. We used B.L.U.E. to estimate a threshold for the amplitude to eliminate false peaks that are lower than the threshold.

The B.L.U.E. is an approach that restricts the estimator and the data to have a linear relationship. We designed an estimator that is unbiased with minimum variance. In our case, let θ be the threshold to be estimated, and A is the coefficient of the data, and X is our observed data. Then we have

$$\theta = \mathrm{AX}.$$

In this equation, the data and the coefficient are both vectors. Then according to the general linear form of the B.L.U.E., we arrive at

$$\mathrm{X} = \mathrm{H}\theta + \mathrm{W},$$

where W is white Gaussian noise, and for simplicity, we assume that H is equal to 1 in our experiments.

$$\mathrm{X} = \theta + \mathrm{W}$$

Then we have the formula of our estimator as

$$\theta_{Blue} = (H^T C^{-1} H)^{-1} H^T C^{-1} X,$$

where C is the covariance matrix of X.

Since in our experiments, the voltage is constant, we assume that θ is also a constant value in this case. Since the noise is white, we can say

$$\theta_{est} = \bar{X}$$

which is just the average of the data.

For accuracy, we only count the peaks we find earlier from our data since we just need a threshold for peaks.

Realization in MATLAB

We developed two programs in MATLAB to implement our approach.

The first one is *Peak_Translocation_Counter.m,* where users can import the data they want to use. The second one is *PeakMethods.m,* where we perform our signal processing method we discussed before.

Since our data is discrete, we can't use the function in MATLAB directly, we decided to use central difference method to perform the first derivative:

$$Y_j' = \frac{Y_{J+1} - Y_{j-1}}{2\Delta x}$$

for 2<j<n-1.

Then we get the first derivative of our data. After this, we apply our 7-point "pseudo-Gaussian" smooth to get rid of noise. In our script, we choose to pass 3-point rectangular smoothing three times for simplicity, which is equivalent to the "pseudo-Gaussian" smooth.

$$S_j = \frac{Y_{j-1} + Y_j + Y_{j+1}}{3}$$

Then the detection and estimation employed. First, we get the first set of our possible peaks by setting the slope threshold for the downward zero-crossing points. We can get the locations of the possible peaks which are recorded in a vector. Then, we use B.L.U.E. to compute the amplitude threshold, and apply it in our scripts. After comparing the possible

peaks with the estimated threshold, we choose those that are bigger than the threshold to be the final true peaks in our experiment.

Results comparison and analysis

In this section, we will compare, analyze the results by different method.

The biology research group used 3 types of external solutions of 5 L to contain the nanoparticles. A blank solution is PBS which is the reference with no translocations of nanoparticles since there is no Au nanoparticles in the PBS. Besides, there are two other solutions labeled as 30K and 300K referring to the number of nanoparticles in the solution respectively. There are six groups of data in each of experiments for the 3 types of solutions with 6 different nanopipettes, NP1, NP2, NP3, NP4, NP5, NP6. Below describes the set-up of our experiment.

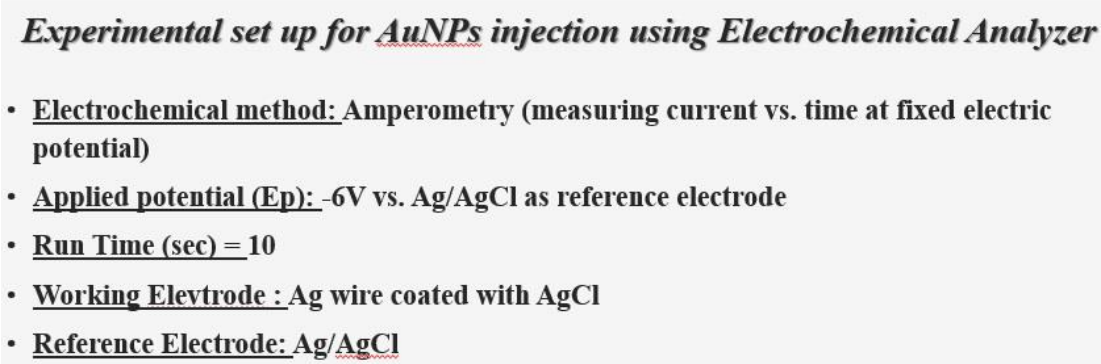
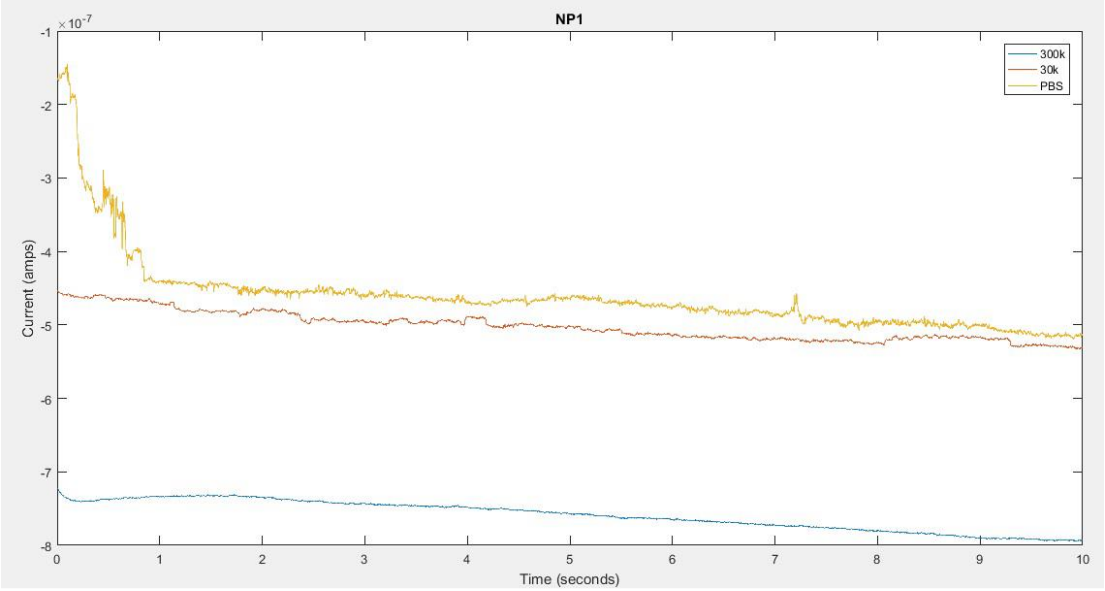
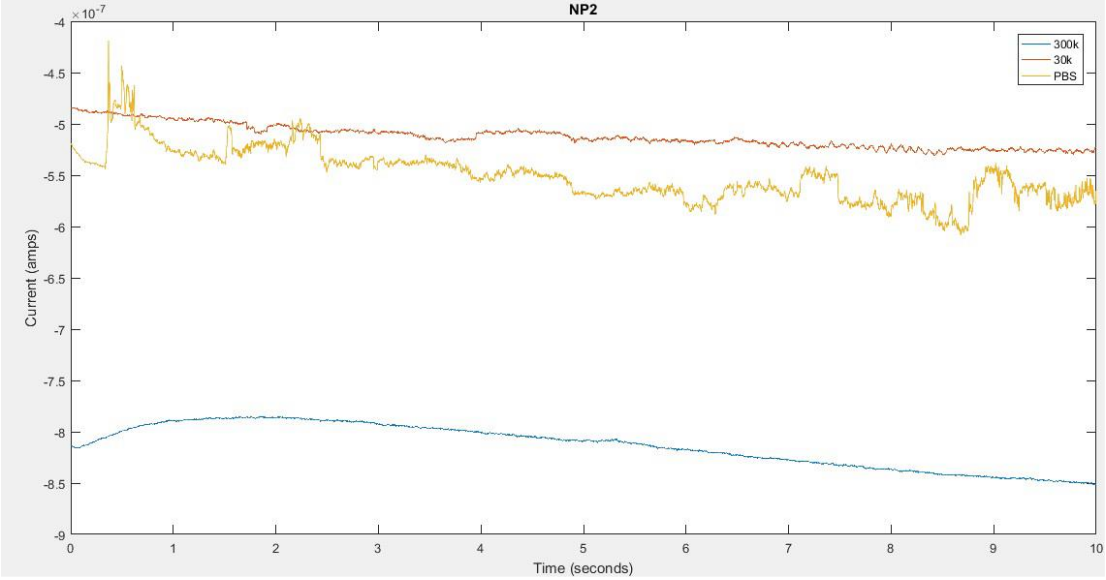
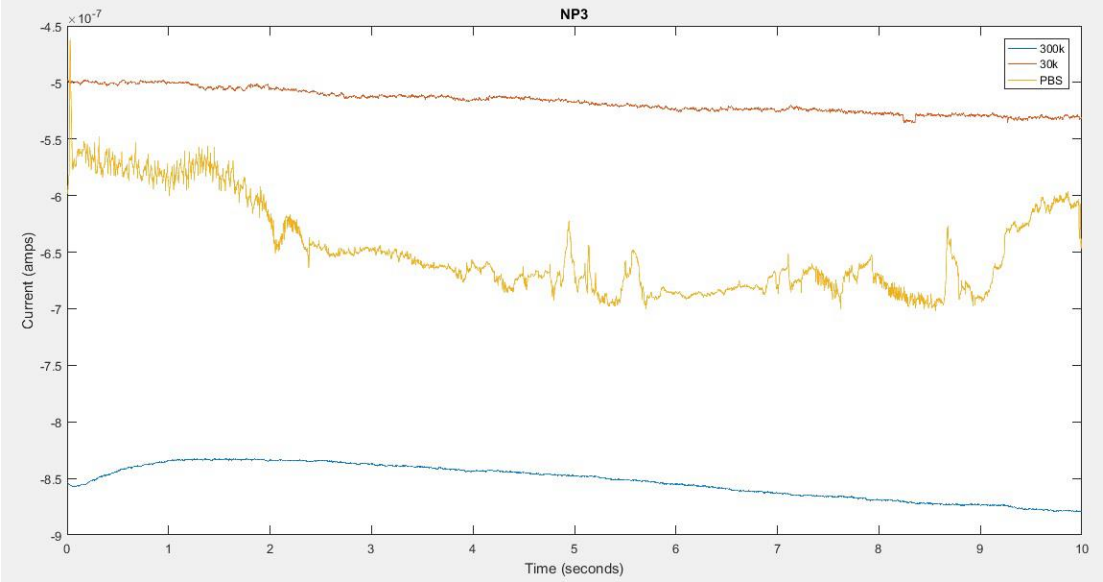


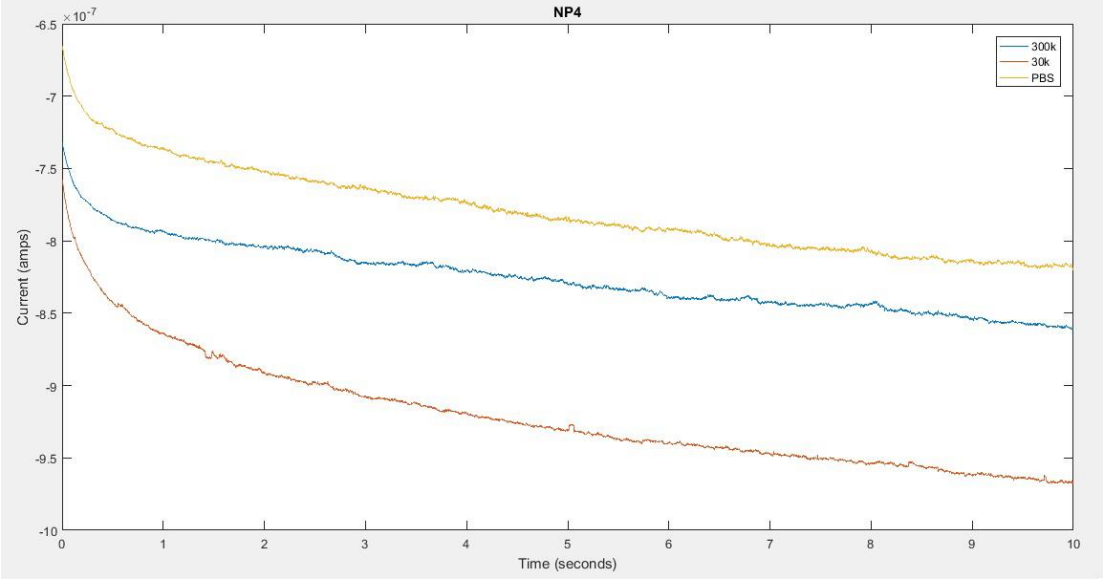
Figure 2 The setup of the experiments.

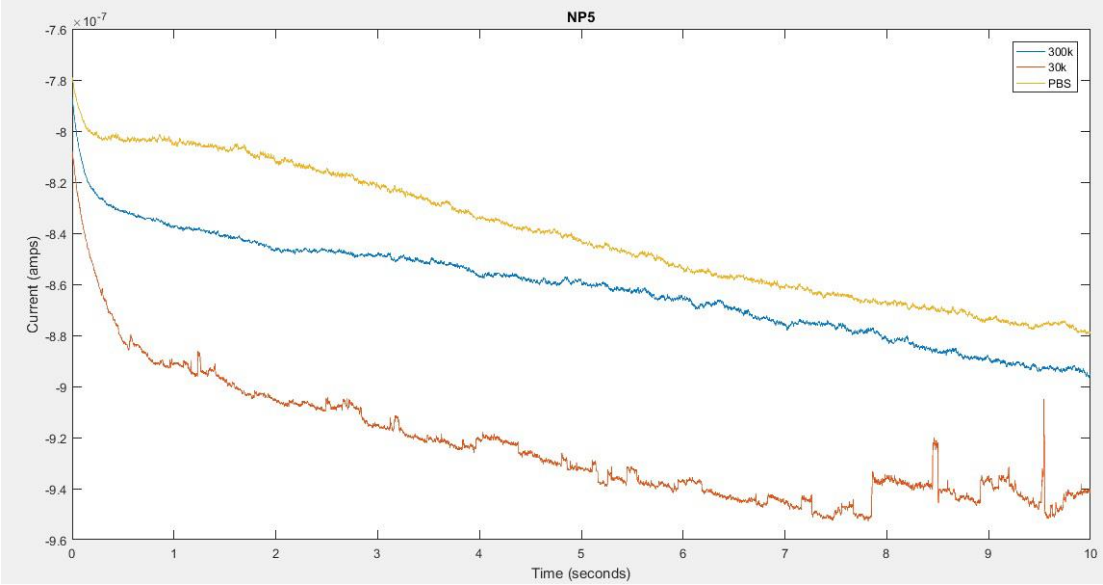
The signals of each nanopipette for each solution are plotted below. Blue refers to 300k, Red refers to 30k, and PBS refers to yellow-orange.


NP1
×10^-7
-1
-2
-3
-4
-5
-6
-7
-8
Current (amps)
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
300k
30k
PBS


NP2
×10^-7
-4
-4.5
-5
-5.5
-6
-6.5
-7
-7.5
-8
-8.5
-9
Current (amps)
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
300k
30k
PBS


NP3
×10^-7
-4.5
-5
-5.5
-6
-6.5
-7
-7.5
-8
-8.5
-9
Current (amps)
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
300k
30k
PBS


NP4
×10^-7
-6.5
-7
-7.5
-8
-8.5
-9
-9.5
-10
Current (amps)
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)
300k
30k
PBS



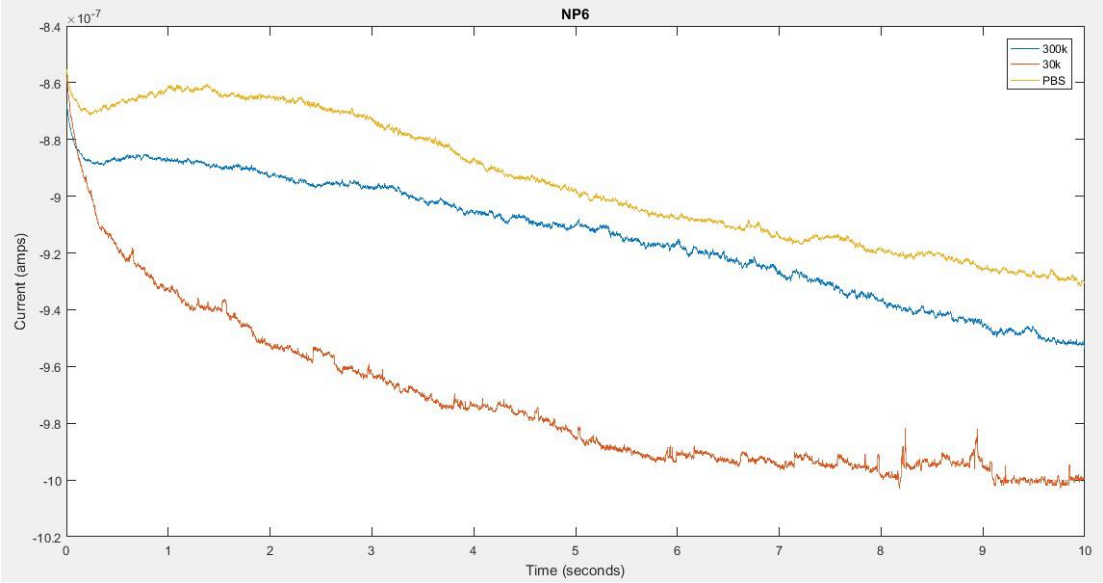


Figure 3 The plot of each nanopipette for the solution.

Here we have 4 different groups of results, getting from MATLAB peakfind function, Biology group's method, our method and the results counted by our hands.

MATLAB FUNTION			
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

Figure 4 The results counting by MATLAB function

Nader's 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

Figure 5 The results counting by Nader's group

SOFTWARE	300k		30K		Blank(PBS)	
Whole File	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

Figure 6 The results counting by our method.

Hand Results:	Andres	Jie	Chenjiang	Averaged Total
NP1	521	450	339	436.6666667
NP2	463	383	309	385

Figure 7 The results counting by hand

From the results above, we can arrive at the following conclusions.

1. When we use our algorithm in PBS, there are also peaks. This means the PBS will also cause the fluctuation of the current. So there must be noise in other 2 kinds of solutions. And removing noise is necessary.

2. From the results of MATLAB and Biology group, we found a contradiction. The peaks should be more in 300K solution, but the results shows there are less peaks in 300K. We think it should be the noise that influence the counting of true peaks and there is no efficient way to remove noise in those two methods. So our method seems to be more accurate.

3. After counting the peaks by hand, where we count peaks as true peaks only if they fluctuate highly enough. Then we get the Figure 4. Although the results might be subjective, it could be a reference in some way.

In conclusion, our developed algorithm written in MATLAB seems to be the most accurate approach.

Conclusion

Our method is reasonable but can be improved. We can improve our method by combining more details and knowledge about the data. For example, we can record how the current changes when a single particle passes the pore or in what condition the accumulation at the pore will happen. If we had more detail information, we could set the parameters better, then the results will be more accurate.

Our detection algorithm has some connection with machine learning. Data science also has some applications in signal processing. If we could combine those together, I believe we could find a better result.

Bibliography

[1] Chaves, G. (2017). Counting Spikes in Nanopipette Injections Using Gold Nanoparticles.

[2] S.-A., Gold Nanoparticles: Properties and Applications, Sigma-Aldrich. [Online]. Available: http://www.sigmaaldrich.com/technicaldocuments/articles/materialsscience/nanomaterials/gold-nanoparticles.html.[Accessed: 04-Sep-2017].

[3] T. C. O. H., A Pragmatic* Introduction to Signal Processing,Introduction to Signal Processing: Table of Contents. [Online]. Available: https://terpconnect.umd.edu/

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