Analysis of Solid State Nanopore for Signal Processing and Control

Nathan N. Nguyen, Christopher R. O'Donnell, Raj Maitra, William B. Dunbar

University of California - Santa Cruz, Santa Cruz, California, 95064

Abstract **-** Interactions between nanopores and DNA strands show promise for next generation DNA sequencing, which has applications in medicine, forensics, and other biotechnology projects. Single-stranded DNA molecules can move through a nanopore, creating translocation events when nucleotides pass through the pore's channel. Detecting these signals and interpreting them could allow reading of the nucleotides' positions. Novel experimental and computational techniques have to be developed in order to characterize these events. In this study, biology, physics, and computer science concepts were employed to observe if solid state nanopores are viable for research. Solid state nanopores are synthetic, and they can be advantageous over biological ones that are usually used for this type of investigation. Experiments were done by applying voltage across a specialized holder that contains a silicon nitride nanopore and a buffer solution. Data were collected with the software Clampex to see if there is an open channel for current, and eventually single-stranded DNA, to go through. A computer program was written in the software Matlab to simulate estimated data and analyze them for more information. Its job is to utilize exponential random function and 4th order low-pass Bessel filter to generate signals as well as remove extraneous noise so the events can be examine correctly. The script looks for dwell times of translocation events, the period between each event, and the current drop associated with a specific event. The processing results showed that the program is capable of collecting and interpreting data accurately. The next step is to get events from the experiments and analyze them with these codes. Later experiments can be enhanced from this study to further research of solid state nanopores.

Nomenclature/Keyword -

pA = picoampere ms = millisecond Å = angstrom
M = Molar mM = millimolar mL = milliliter
g/mol = gram per mole
mV = millivolt µL = microliter

HEPES = 4-(2-hydroxyethyl)-1-piperazineethanesulfonic acid
EDTA = Ethylenediaminetetraacetic acid

I. Introduction:

Solid state nanopores are drilled with ion beams and the structures can be silicon nitride, silicon oxide, or graphene. Synthetic pores are more stable than biological ones and do not require DNA polymerase, such as phi29, to facilitate DNA's movement. The objective of this research project was to do early testing with the solid state nanopore to observe its' viability for research. Data from these experiments were then analyzed with scripts written in Matlab for useful information. The results from this study can help future investigations regarding solid state nanopores and DNA interactions.

II. Experiment with Solid State Nanopore:

For the basic setup, a Si_3N_4 nanopore is put onto a distinctive holder specifically to house the nanopore. The holder is consisted of five glass parts. The nanopore situates in the middle of the center part, and the other four act as reservoir for liquid to come in and surround the pore. The holder is assembled together with silicone grease, screws, and nuts.

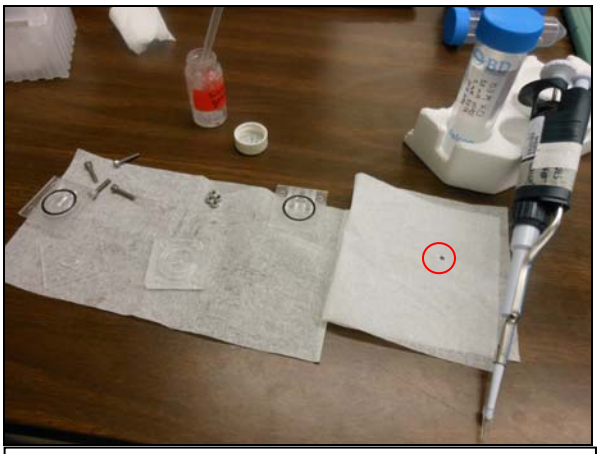

Figure 1. Setup for solid state nanopore. The nanopore is on a chip. Chip is marked in red circle.

Two solutions were made for this experiment. A buffer was made and act as liquid medium around the pore. The buffer includes 0.3 M of potassium chloride (KCl), 10 mM HEPES, 1 mM EDTA, and 50 mL of deionized water (dH_2O). This solution was titrated with potassium hydroxide (KOH) until pH reaches 8 and put into a 50 mL test tube. This buffer, specifically KCl, is important for DNA's movement across the nanopore. Piranha solution also was prepared for this experiment. It is made from sulfuric acid (H_2SO_4) and hydrogen peroxide (H_2O_2). The purpose of this solution is to hydroxylate the pore's surface and inner channel, making these areas hydrophilic so the buffer and DNA can pass through the pore.

After having all the necessary components, all the parts are greased. Silicone grease help stick the glass joints together, and also prevent buffer from leaking out of the holder because of its hydrophobic nature. The nanopore was then placed in a hole in the middle of the center part with some buffer added. Next, the other parts are united with the center part to make the holder, and the parts were secured with four pairs of screws and nuts. Using the p1000 micropipette, 1400 µL of buffer was placed into the holder on each side of the pores through the openings. The holder was placed in a Farraday cage. Two electrodes in the cage are submerged on both sides of the holder. An Axon Instruments amplifier was set to 100 mV and introduced voltage into

the surrounding medium the nanopore is in. Voltage gives energy to this system, so DNA strand can move toward the pore and go through. The pore's channel acts as a resistor, which means voltage through the pore creates current. When DNA goes through the pore, a drop in current would be introduced. The reason for this phenomenon is that current is proportional to the pore's radius. DNA's nucleotides passing through the channel would reduce the radius, therefore reducing the current.

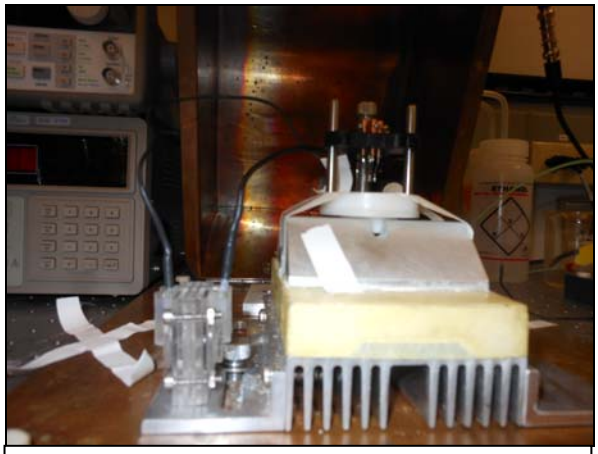

Figure 2. Completed holder with nanopore is connected with electrodes from the Faraday cage.

III. Programming for Data Examination:

A script were written in the software Matlab, and it tests for three criteria of the translocation events, like one shown in Fig. 3: the dwell time of translocation events, the time between each event, and the amount of current dropped from base value for each event. The script was done to simulate the data that would be obtained from the experiments. The algorithm analyzes the events simulated and extract the necessary information.

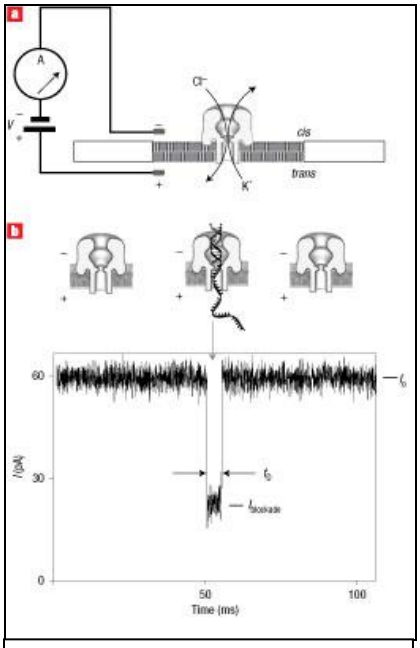

Figure 3. DNA moves through pore creates translocation event.

First, the script uses functions from Matlab to construct matrices that house values tau (τ) and rho (ρ). These values are generated from the random exponential distribution function. τ stands for the values of the starting positions, and ρ stands for the values of the dwell times. These values were simulated in millisecond, with τ mean = 100 ms and ρ mean = 1 ms. Two matrices are made, one with 50 τ's, and the other with 50 ρ's. The values were taken from the exprnd function with the appropriate means. The random distribution is to account for the randomness in time when DNA's travel through the pore.

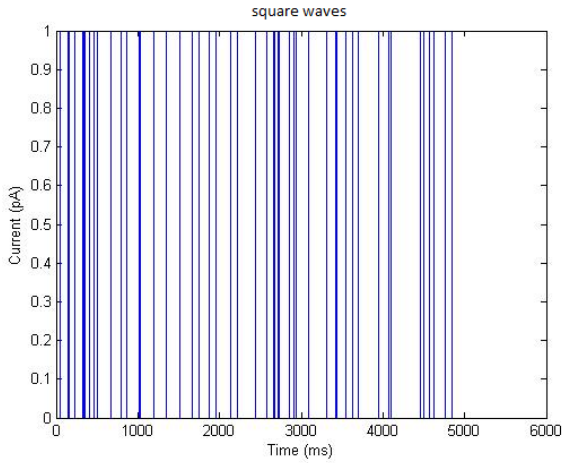

Figure 4. 50 random events generated from exprnd() function in Matlab.

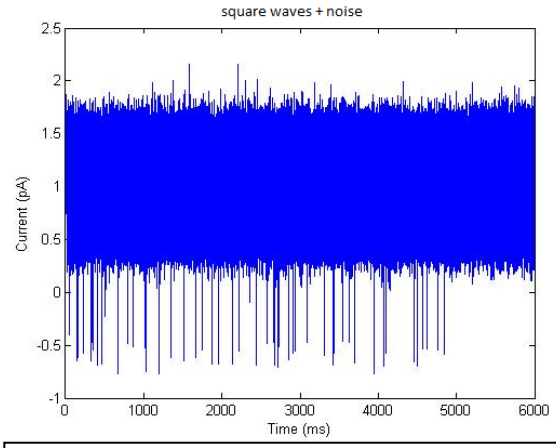

Figure 5. The same events from Fig.4, but with extra noise.

Square waves were made following the τ as start position on x-axis and ρ as the width of each wave. An event on y-axis was set arbitrarily to happen when current drop from 1 pA to 0 pA. Random noises were introduced to the square waves, as shown in Fig. 5, effectively mimicking the events from experimental data. The noise has a standard deviation of 0.2 mV. The script then called a filter function to take out the noise, with the purpose to see if it is possible to get back the events. This low-pass filter is the Bessel Filter, with sample frequency set at 1000 kHz and cutoff frequency at 5kHz. A low-pass filter allows lower frequencies to go through, but higher frequencies will not be able to. Since the noise has higher frequency than the cutoff frequency of 5 kHz, the filter correctly removed the noise and the events were shown clearly in the next graph.

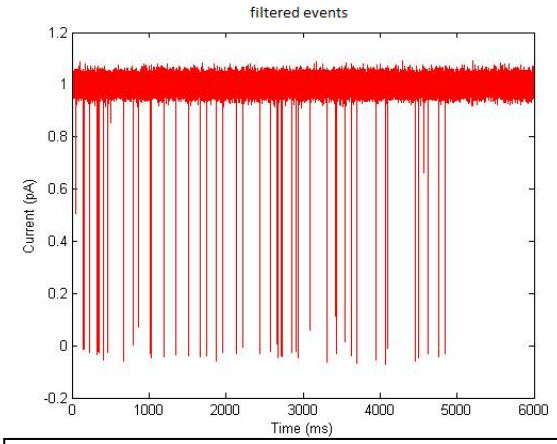

Figure 6. Events after filtering process by Bessel filter.

To analyze these events, two thresholds were established to capture the events and get all the information wanted. The thresholds chosen were at current level 0.9 pA and at 0.1 pA. The upper threshold was set at 0.9 because the Bessel filter cannot get rid of all noise, so the region between 1.1 and 0.9 pA is difficult to distinguish an event from a noise signal. The lower threshold at 0.1 pA act as limit to ensure that signals counted as events are clearly detected with an evident drop in current. Events that the script cannot detect

would have amplitudes not reaching down pass 0.1, and this is a situation that can happen when real data is being examined with the script.

The script established start points and end points of each event. The start point is where the curve of event hit 0.1 on the y axis and the end point is at 0.9 when the curve goes back up. The reason for the locations of these start and end points is because the values are measured in step function. The distance between the two green dots of the same event would be the event's dwell time in ms.

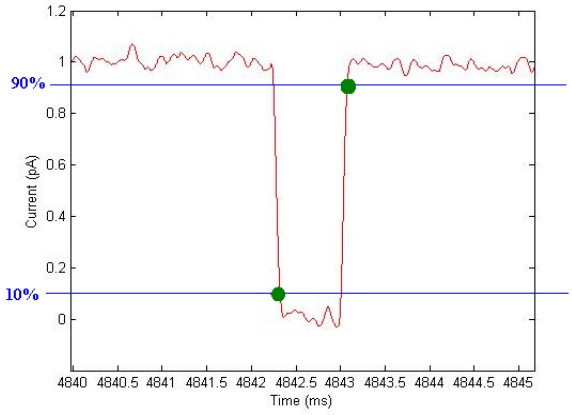

Figure 7. Closeup of an event in Figure 6 with thresholds marked as piecewise constant.

To get the distance between events, A subtraction can be done by the script between the start of one event and the start of the previous event, as is shown in Fig. 8.

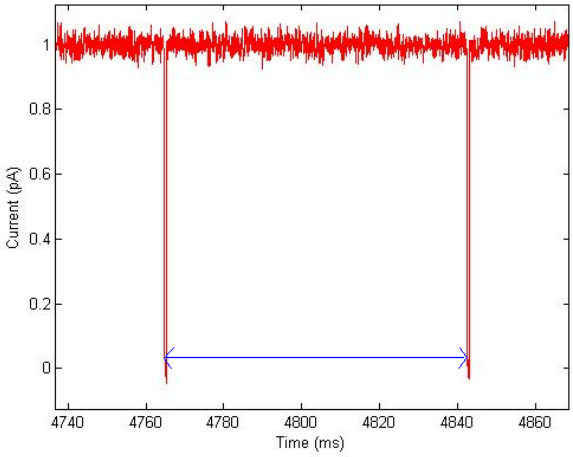

Figure 8. Dwell time is the distance of the double-headed arrow.

IV. Results:

Shown in graphs below are the results from Fig.4 (blue) comparing to Fig.6 (red). The script computed the amplitudes, dwell times, and time period between events, in both the original graph with square waves and the final filtered graph. After all the numbers are calculated and put into matrices in Matlab, the averages for these three criteria were computed and shown below. For this particular example, out of 50 events created, 47 events were recovered after the addition of noise and the filter process.

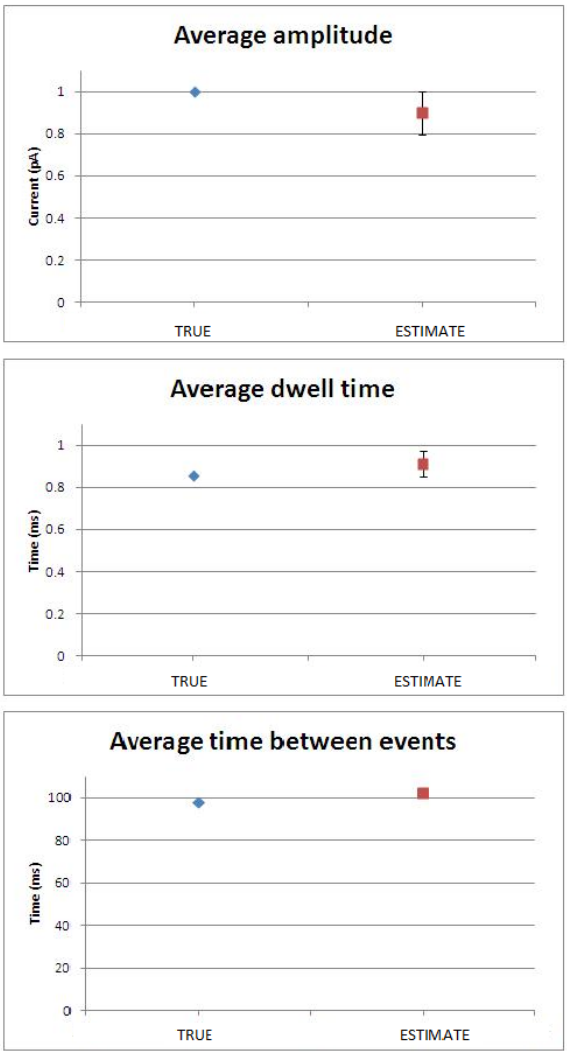

Figure 9. Averages for the three tests with percent errors.

The results from multiple executions of the script showed that the codes work in detecting the events and examine them correctly. The percent errors for average dwell times and for average time between events measurements are around 5% to 15%. The percent errors for average amplitude is 10%. This makes sense because for the amplitude, the cutoff was set to be at 0.9 for analysis of the filtered graph, so it is not possible to get an average greater than 0.9. When compared with the average amplitudes from the original graph with amplitude equals 1, it is clear that the error is within the range of +-0.1.

V. Future Work:

With the program in Matlab working, the next step would be to get the solid state nanopores to work and do more testing with actual DNA strands. After performing the experiments and collect the data, the results will be analyzed with Matlab.

Acknowledgment

The author thanks professor William Dunbar, Christopher O'Donnell, Raj Maitra, and Shea Ellerson for guidance with the research project. The author also thanks Matthew Guthaus and Colt Hangen for their support at UC Santa Cruz. The solid-state pore device and chips were purchased from Nanopore Solutions Ltd. This project was sponsored by the UCSC SURF-IT program and the National Science Foundation grant No.1156606.

References

[1] Wilson, N., Abu-Shumays, R., Gyarfas, B., Wang, H., Lieberman, K., Akeson, M., & Dunbar, W. (2009). Electronic Control of DNA Polymerase Binding and Unbinding to Single DNA Molecules. *ACS Nano*, *3*(4), 995-1003. doi:10.1021/nn9000897

[2] Hornblower, B., Coombs, A., Whitaker, R., Kolomeisky, A., Picone, S., Meller, A., & Akeson, M. (2007). Single-molecule analysis of DNA-protein complexes using nanopores. *Nature Methods*, *4*, 315-317. doi:10.1038/nmeth1021

[3] Gyarfas, B., Abu-Shumays, R., Wang, H., & Dunbar, W. (2011). Measuring Single-Molecule DNA Hybridization by Active Control of DNA in a Nanopore. *Biophysical Journal*, *100*(6), 1509-1516. doi:10.1016/j.bpj.2011.01.029

[4] Dekker, C. (2007). Solid-state nanopores. *Nature Nanotechnology*, *2*, 209-215. doi:10.1038/nnano.2007.27

[5] Benner, S., Chen, R., Wilson, N., Abu-Shumays, R., Hurt, N., Lieberman, K., Akeson, M. (2007). Sequence-specific detection of individual DNA polymerase complexes in real time using a nanopore. *Nature Nanotechnology*, *2*(11), 718-724. doi:10.1038/nnano.2007.344