UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ELECTRONIC CONTROL OF DNA POLYMERASE BINDING AND UNBINDING TO SINGLE DNA MOLECULES TETHERED IN A NANOPORE

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

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Noah A. Wilson

March 2009

The Dissertation of Noah A. Wilson
is approved:

Professor William B. Dunbar, Chair

Professor Mark Akeson

Professor Jacob Rosen

Lisa C. Sloan
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Electronic control of DNA polymerase binding and unbinding to single DNA molecules tethered in a nanopore

by

Noah A. Wilson

This work combines computer control with a nanopore sensor to detect and manipulate single DNA molecules and DNA/enzyme complexes captured in the pore. In the setup, a membrane protein self-assembles into a lipid bilayer and an electric potential applied across the membrane creates a measurable ionic current that flows through the pore. A sufficiently large potential produces an electric field capable of capturing DNA molecules and enzyme-bound DNA complexes in the pore. Captured unbound DNA molecules pass through the pore. Because enzymes are too large to enter the pore, enzyme bound to captured DNA must dissociate from the DNA before the DNA translocates through the pore. The basis for detection of translocating polynucleotides, such as DNA, is the measurable reduction in ionic current through the pore. Variations in the current amplitude can differentiate unbound DNA from enzyme-bound DNA in the pore and enable real-time detection of enzyme dissociation from captured DNA. A finite state machine automates detection and reaction to the dissociation of enzymes from captured DNA in real time. Moreover, additional characteristics of the current signal permit identification of the base present at the enzyme's catalytic site, providing a means for single-base sequencing. Combining these capabilities, this research explores methods

for repeated control of a single tethered DNA molecule, suspended in the nanopore by voltage control and biochemical alteration. In the tethered configuration, the DNA is available for rapid and repeated binding of enzymes above the pore. The repeated binding and subsequent dissociation of enzymes represents a dramatic advance in the level of control of DNA in a nanopore. With such control of DNA, titration experiments allow investigation into the detectable limits of complex assembly, at the single-molecule level, with statistical significance and provide pre-steady-state biochemical information.

Acknowledgments

I have been very fortunate to have found such an excellent research group. I would like to thank my advisor, Prof. William B. Dunbar, for his friendship and constant enthusiasm for research that helped fuel my own enthusiasm and interest. Prof. Mark Akeson welcomed me with open arms into his lab and reminded me constantly that no accomplishment was ever too small. Prof. David Deamer provided continual input on the research and I thank him for his optimistic and insightful perspectives on our results. Thank you Prof. Jacob Rosen for taking an interest in the research and for being on my committee.

I would like to thank Robin Abu-Shumays for her tireless efforts, patience and friendship during the many long experiments that made up the bulk of this research. Seico Benner openly extends her calmness to everyone she interacts with and I thank her for her help in the initial fishing experiments and for keeping things in perspective. I want to thank Kate Lieberman for teaching me never to take an assumption for granted; a perspective that helped me develop as a scientist. Brett Gyarfas, Daniel Garalde and my other lab mates provided me with a audience to test my ideas on and a camaraderie that made my experience more enjoyable. I want to thank the School of Engineering staff for doing what they do so well and making the administrative aspects of my graduate experience relatively painless.

Finally, I would like to thank my family and my lovely, understanding wife, Diana, for their endless support throughout my entire graduate experience.

Chapter 1

Introduction

Single-molecule measurement techniques have emerged as an indispensable venue for the study of biological phenomena. Details not visible using traditional bulk phase techniques can be captured at faster time scales with nano-scale precision [20]. The nanopore has proven itself as a sensor that produces variations in ionic current representative of molecules captured in the pore. An applied voltage controlled through a voltage-clamp amplifier pulls charged molecules such as biological polymers through the pore.

The work described here aims for and achieves the goal of controlling single-molecules through active voltage control. I have developed an infrastructure for designing and executing finite state machines (FSMs) on a field-programmable gate array (FPGA) for detection and reaction of molecules captured in the nanopore. Extra care was taken during development to ensure a user friendly interface during design and operation of the FSM, particularly as user backgrounds range from engineering graduate students to staff research scientists (non-engineers, non-technicians). The application of the interface, and the research question addressed in this work, is to capture a single DNA molecule, hold it in the pore, and repeatedly expose the molecule for protein binding and dissociation above the pore. These results and the interface itself represent novel contributions to nanopore research. Experimental procedures not previously possible are now being performed using the state machine infrastructure for active voltage control of several different complexes of scientific interest, including polymerases (KF and T7) and exonucleases (λ-exo).

1.1 Background

We use the protein channel α-Hemolysin (α-HL) of *Staphylococcus aureus* for the nanopore. When created in an ionic buffer of 0.3 M KCl, an applied voltage across the pore produces a steady open channel current signal (60 pA at 180 mV *trans*-positive). Biological polymers, such as single-stranded DNA (ssDNA) or RNA, can then be electrophoretically driven through the nano-scale pore. The voltage across the pore creates a field that forces polymers to translocate through the pore due to the polymer's net (negative) charge [2, 13, 29]. These polymers block ion flow through the pore during translocations, thus attenuating the open channel current. Modulations in the current signal can then be used to identify the translocating polymer.

Controlling polymer capture using a electric field allows nanopores to rapidly examine many different molecules in series. Other single-molecule methods such as atomic force microscopy (AFM) or optical tweezers require the sample of interest to be attached to the measurement instrument (the cantilever arm or a dielectric bead in the case of AFM or optical tweezers, respectively) [20]. Nanopore assays do not require this extensive sample preparation, and generate a high number of measurements in a short amount of time (100 translocations per minute, [3]). In contrast, AFM or optical tweezers typically capture more spatially precise measurments compared to existing nanopore techniques, which has made them popular in recent single-molecule assays [20, 34].

1.1.1 Solid-state nanopores

Many research groups are exploring and developing solid-state nanopores to achieve low-cost, high throughput nanopore-based sequencing [37]. A nanopore-based method would reduce sequencing costs to the minimal DNA prep involved. DNA amplification would be unnecessary since the nanopore device would monitor sequence-dependent variations in the ionic current at the single-molecule level, in principle, for each translocating DNA molecule. Several challenges, primarily related to achieving single-base resolution, have to date kept nanopore-based sequencing from becoming a viable technology [6].

In recent years, the increased production reliability and structural robustness of synthetic nanopores have caused them to gain favor over their biological counterparts. The primary method of manufacturing solid-state nanopores uses a focused electron beam to bore a nanometer-scale hole in a semiconductor such as SiN [14]. Atomic deposition then can be used to reduce the hole size to within a tolerance of several nanometers [14, 45]. This manufacturing variability makes solid-state pores difficult to characterize when compared to biological pores that have identical dimensions from one channel to the next. In spite of these challenges, research involving solid-state pores continues to move forward for a variety of single molecule sensing applications [14].

1.1.2 Biological nanopore using α-Hemolysin

My work utilizes a biological nanopore setup consisting of a single α-hemolysin protein channel in a planar lipid bilayer. Seven α-hemolysin proteins self-assemble to form a channel through the lipid bilayer with the trans-membrane portion primarily comprised of a β-barrel [5, 40]. The channel (pore) measures 15 nm in length and varies in diameter. The *cis*-opening of the pore measures 2.6 nm wide and opens to a 3.6 nm vestibule before narrowing to a limiting 1.5 nm at the beginning of the stem. The remainder of the stem up to the *trans*-opening is 2 nm wide, shown in Figure 1.1 [40]. The vestibule can accommodate double-stranded DNA (dsDNA), but the limiting stem is just wide enough for ssDNA to pass through and requires the individual base pairs of dsDNA to dissociate before ssDNA translocation occurs. The structure of the α-HL channel makes it an excellent candidate for DNA research since the channel's limiting aperture forces serial analysis of DNA molecules (only one molecule per translocation), and ensures the bases will pass through the pore in the correct sequence order.

An Ag-AgCl electrode in each of the two volumes on either side of the pore

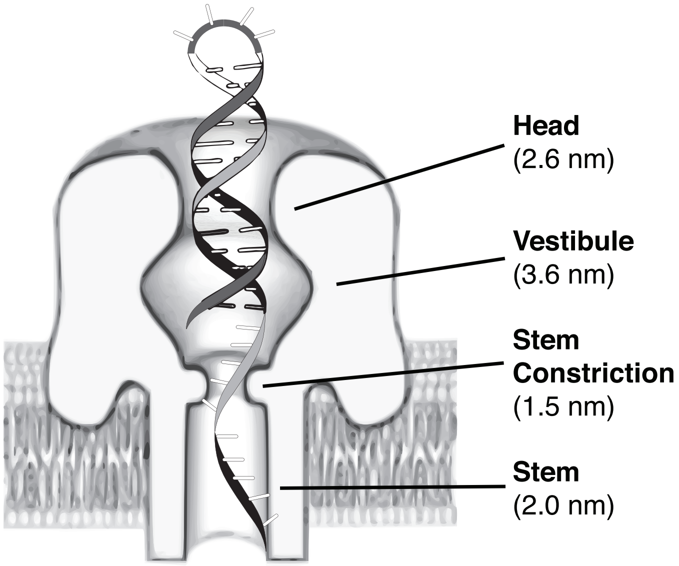


Figure 1.1: The α-hemolysin pore can accommodate both ssDNA and dsDNA in the vestibule, but only ssDNA through the stem constriction and stem.

create the field that produces the ionic current through the pore. The pore serves as the only passageway between the two electrodes since the lipid bilayer membrane separates the two volumes otherwise. The electric field electrophoretically pulls the negatively charged phosphate backbone of the polymer through the pore, passing from the *cis* side to the *trans* side of the pore with a *trans*-side positive applied voltage [13, 29]. As molecules translocate, the pore becomes partially blocked with the molecule impeding the transport of ions through the channel. This translocation event shows up in the ionic current signal as a deflection from the open channel current [13, 29] and can be characterized by the amplitude of the blockade current and the time the molecule occupies the pore, defined to be the event's *amplitude* and *dwell time*, respectively. For example, upon capture of a 20 base pair hairpin (20bphp) with a single-stranded overhang in the pore, the ssDNA region enters first and the hairpin remains in the

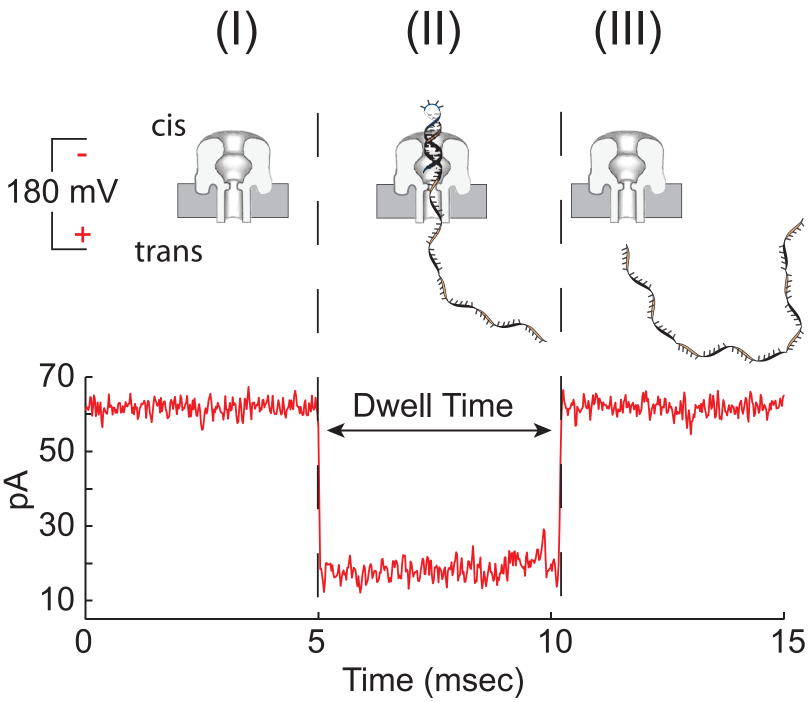


Figure 1.2: Schematic of nanopore and DNA, and plot of representative ionic current signal during a 20 bphp DNA translocation event under 180 mV applied potential. (I) At 180 mV and 0.3 M KCl, ions pass through the open channel resulting in ∼64 pA current. (II) Upon capture of the single-stranded end of the DNA molecule into the *cis* opening of the pore, the flow of ions is reduced to ∼20 pA. (III) After ∼5 ms, the voltage unzips the hairpin, causing ssDNA to pass through the pore into the *trans* chamber, completing the measured blockade event. The duration of the event is referred to as dwell time and the mean current amplitude during the event is referred to as the event amplitude. Reproduced from [46]

vestibule until the duplex region dissociates and the polymer translocates into the *trans* volume, as illustrated in Figure 1.2.

1.1.3 Patch-clamp amplifier principles of operation

Patch-clamp amplifiers use the voltage drop across a large feedback resistor ($R_f = 500\text{M}\Omega$) to measure small changes in current at a known applied voltage [38].

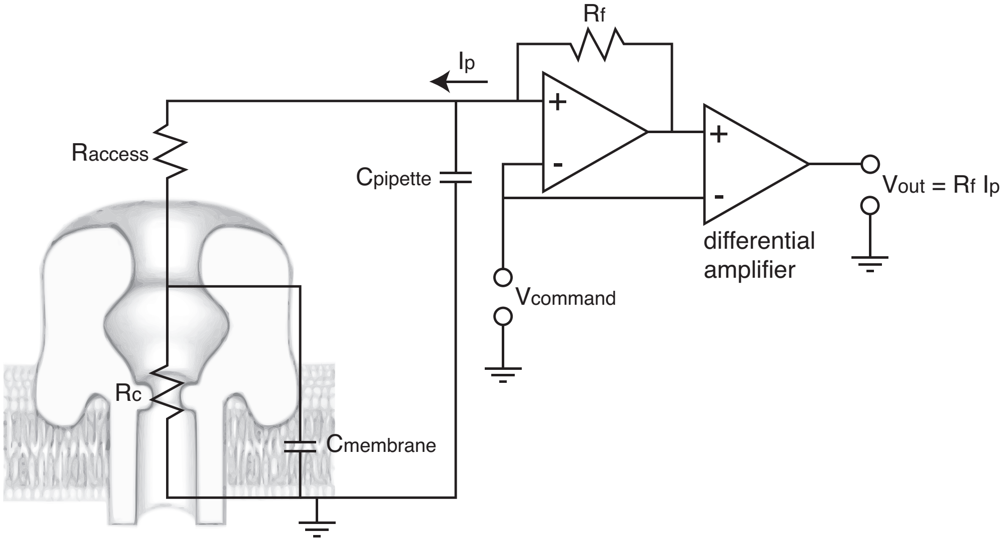


Figure 1.3: The patch-clamp circuit uses the voltage drop across the R_f resistor to measure the current through the pore. The capacitive effects of the amplifier, pore and lipid membrane induce a transient on the current signal immediately following a change in the command voltage, V_{command}, that limits the response time of the FSM.

Figure 1.3 shows a basic schematic of a patch-clamp circuit. A translocating molecule blocks the pore and causes less current to flow, thus increasing the effective resistance of the pore. Since the amplifier is clamping the transmembrane voltage V_m to match the command voltage, the ionic current I_p must drop when the pore resistance R_c increases, according to Ohm's law $V_m = R_c I_p$. Subsequently, the voltage across the feedback resistor drops proportionally with the change in pipette current according to $V_{out} = I_p R_f$. The voltage across the feedback resistor passes through an differential amplifier to condition the measured signal for recording I_p in real-time.

The capacitive elements of the patch-clamp amplifier and nanopore system cause a transient in the current signal immediately following a change in V_{command} that can last up to 12 ms, though it is typically less than 5 ms. Lower KCl concentration also cause longer lasting transients. The transient requires the FSM to wait until the transient settles before using the current signal to diagnose translocation events. Feed-forward injection of current from a secondary source can reduce the effect of the transient following a voltage change [38]. Circuitry on the Axopatch 200B provides this compensation but the amplifier does not allow the compensation to be precisely (digitally) controlled. At present, we run the patch-clamp amplifier without the capacitive compensation. A systematic approach for transient compensation is under development.

1.1.4 Detection of translocating molecules using a nanopore

Correlations between the ionic current amplitude and the nucleotide identity of individual DNA or RNA molecules translocating through the pore has been shown through various assays using α-hemolysin nanopores [2]. A near direct correlation between the number of molecules passing through the pore and the number of deflections in the ionic current signal has been demonstrated and confirms that the observed drops in current correspond to individual translocating molecules [29]. Homopolymers of ssDNA and block copolymers of RNA can also be differentiated based on the measurable differences in the blockade current amplitude or kinetics [2, 36]. However, rapid translocation rates (as fast as to 3 μs per nucleotide, [2]) prevent sequencing individual nucleotides in heterogeneous single-stranded polymers using existing biological nanopores [14]. Even at slower translocation rates, the 10–15 bp that occupy the pore during capture could occlude details specific to any individual base [6, 40].

Since single-base resolution cannot yet be achieved using the pore [6], creative DNA template design must be employed to exploit other biological or physical phenomenon of the captured molecules for discrimination of different molecular states [4]. Here and in other nanopore studies [4, 32, 44], using DNA with single and double-stranded segments increases the dwell time of nucleotides in the pore, with the ssDNA/dsDNA junction located as shown in Figure 1.2. Another approach utilizes DNA-binding proteins (enzymes) to increase the DNA dwell time in the pore [4, 23].

1.1.5 Coupling DNA-binding proteins with DNA captured in the nanopore

Adding DNA-binding enzymes to nanopore experiments creates DNA/enzyme complexes that can also be captured in the nanopore. Under an applied voltage, the ssDNA end of enzyme-bound DNA is captured in the nanopore while the enzyme resides on top of the nanopore since it is too large to pass through the channel, Figure 1.4. Binding of enzymes to DNA in this configuration shows an increase in the dwell time of DNA in the nanopore by up to two orders of magnitude [4]. Kinetics of *Escherichia coli* exonuclease I binding to ssDNA has been quantified with voltage ramps using nanopore-based force spectroscopy [23]. The electric field force exerted on the ssDNA causes it to dissociate from the enzyme after several milliseconds before translocating. The time-to-dissociation in turn can be correlated to enzyme binding rate constants.

The Klenow fragment of *Escherichia coli* DNA polymerase I (KF) is a model for studying DNA polymerase structure and function [28, 26]. Crystal structures of

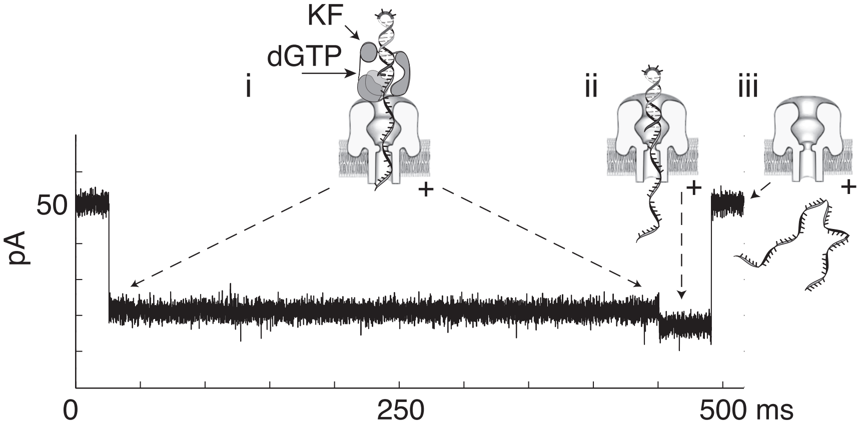


Figure 1.4: Representative current trace and illustration of corresponding molecular events during nanopore capture of a KF/DNA/dGTP ternary complex. The initial ∼21.5 pA blockade (i) arises from capture of the enzyme bound DNA, with the duplex held atop the pore vestibule bound to KF. The shorter ∼18.5 pA terminal step (ii) occurs upon KF dissociation, when duplex DNA is pulled into the nanopore vestibule, followed by (iii) translocation of the DNA through the nanopore and return to the open channel current. Reproduced from [47]

these enzymes, which catalyze template-dependent DNA replication, reveal a highly conserved catalytic domain that resembles a partially open right hand [7]. The palm subdomain contains residues essential for catalysis, the thumb subdomain positions the primer/template DNA duplex in the active site, and the fingers subdomain is essential for binding incoming deoxynucleoside triphosphate (dNTP) substrates. The KF catalytic cycle has been characterized in kinetic studies [27, 26]. In an ordered assembly mechanism, KF recognizes the double-strand/single-strand junction of its primer/template substrate to form a binary complex, to which incoming dNTP binds to form a ternary complex. Binary and ternary complex crystal structures of A-family polymerases closely related to KF reveal a large conformational transition between these two functional states [15, 25, 31]. In polymerase-DNA-dNTP ternary complexes, the fingers subdomain rotates toward the active site to achieve a tight steric fit with the nascent base pair. In this closed ternary complex, the affinity of KF for its DNA substrate is increased by ∼5 to 8-fold [9].

In [4], the interaction of DNA with KF was explored. In the absence of KF, capture and subsequent unzipping of 20 bphp at constant 180 mV reveals blockades with 20 pA mean amplitude and 4 ms median dwell time. Addition of KF and the dNTP complementary to the DNA template base in the KF catalytic site yielded a substantial increase in blockade dwell times (110 ms median lifetime for dGTP), attributable to ternary (DNA/KF/dGTP) complexes. Closer investigation of such blockades revealed a two-step pattern in greater than 97% of the blockades, the first step at 24 pA mean amplitude, and the second (terminal) step at 20 pA mean amplitude lasting 4 ms consistent with the hairpin kinetics alone. It was demonstrated that the transition from step one to two resulted in dissociation of KF from DNA first, followed by hairpin dropping into the pore vestibule until unzipping occurred, illustrated in Figure 1.4. As a initial effort at voltage control of enzyme-bound DNA, we demonstrated efficient automated detection of individual ternary complexes ($<$ 3 ms), based on the characteristic 24 pA amplitude, and truncation of the blockade time by voltage reversal after 20 ms [4]).

1.2 Related work

1.2.1 Active voltage control and nanopore methods

Direct control of ssDNA in a nanopore has been demonstrated where detection of DNA is based on monitoring raw current amplitude relative to a threshold level [3]. Voltage level changes in [3], comparable to those employed in this dissertation, were commanded to explore the effects of zero and low voltage on ssDNA-pore interactions. Basic threshold-based detection/reaction affords very fast detection times but does not yield the level of specificity required for our research objectives.

In contrast with simple thresholding of the raw ionic current amplitude, the signal processing employed in this research filters the current noise. Detection requires the filtered signal to remain within a preset amplitude range ($<$ 4 pA in spread) for multiple consecutive comparisons lasting for a set time.

Force spectroscopy experiments involving α-HL channels have been performed with DNA hairpins and DNA/protein complex. Applying constant and ramping forces to a captured DNA hairpins allows kinetic information to be calculated based on the time to unfold [16, 32]. By applying Kramers approximation to the time to unfold under several different forcing functions, kinetic information of the bond under test can be estimated [16]. Optical tweezers (OT) and atomic force microscopy (AFM) are traditionally better suited for such measurements, but this work overcomes several of the obstacles that currently limit the applicability of nanopore force spectroscopy and will facilitate application of theory to nanopore-based assays.

Unlike OT or AFM, the nanopore does not require manual connection of the molecule under test and can generally produce more measurements in a given amount of time than these methods. The primary hurdle in nanopore force spectroscopy is the measurement quality. The measurements performed by a nanopore instrument are generally only capable of measuring the time to unfold where OT and AFM can measure the time to unfold and the distance the molecule unfolded. The distance data allow more straightforward calculations to estimate the rupture forces. More recent work has shown that distance information is available with nanopores by careful placement of abasic inserts in the DNA, but this is a very recent development.

Similarly, preliminary binding kinetics of ssDNA with Exonuclease I can be ascertained using ramping voltage forces with ssDNA/Exonuclease I complex captured in the pore [23]. The FSM/FPGA infrastructure featured in this dissertation offers a variety of different voltage forcing functions and schedules possible due to the design flexibility of a FSM.

Studies most closely resembling the tethering work of this dissertation involve capture of a DNA-PEG copolymer in a α-HL channel under a *trans*-negative applied voltage [10]. Biotin-streptavidin attached to the PEG end of the copolymer can not pass through the pore due to its large size, thus tethering the molecule in the pore from the *trans* end. A DNA primer in the *cis* chamber anneals with the captured molecule and allows DNA polymerase to bind to the ssDNA/dsDNA junction. The polymerase then adds one base to the ssDNA which shifts the PEG further into the stem of the pore. The amount of PEG present in the pore at 40 mV can be determined from the observed ionic current. This reveals information about the state of the capture complex, specifically the number of bases added to the captured DNA by the polymerase. Monitoring the current allows observation of single-base additions by a polymerase, though not in real-time. Real-time detection would be difficult since the size of current steps produced by single-base additions are on the order of the noise.

1.2.2 Other single-molecule methods

Optical tweezers (OT) and atomic force microscopy (AFM) allow for direct measurement of DNA structure and its interactions with proteins [8]. Recently, OT assays have been performed that have monitored individual biomolecular motion [20]; this has not been accomplished with nanopores.

For example, OT have been used to sequence DNA by attaching a processive enzyme, RNA polymerase (RNAP), to a polystyrene bead [1, 19]. Four experiments were run where a well containing three of the four nucleotides necessary for polymerization was monitored for enzyme motion via the connection of an polystyrene bead to RNAP, RNAP to an RNA template, and that template to another polystyrene bead. Each of the beads are captured in an optical trap in a dual-trap dumbbell configuration [20], capable of measuring RNAP's motion along the RNA template. Limiting amounts of ATP, TTP, GTP or CTP were introduced into the well such that saturating concentrations of three nucleotides and a limiting concentration of the fourth nucleotide were present. Under these conditions, RNAP worked along the RNA template and paused whenever it required the rate-limiting nucleotide. The data from the experiments with all four nucleotides at rate limiting concentrations were be lined up and the pauses in the enzyme's motion were correlated to the sequence of the template strand. This sequencing method is not efficient or economical enough to be used for bulk sequencing, but illustrates the kind of spacial precision possible using OT.

Sequencing aside, many of the details surrounding the operation of RNA polymerase has been elucidated using OT: elongation kinetics [12], transcription termination [30], transcription [21], sequence specific pausing [22], behavior dependence on temperature [33], and backtracking during transcription [39]. Lambda Exonuclease [35] and Kinesin [17] motion has also been studied using similar techniques.

Presently, greater spatial and temporal resolution of individual DNA polymerization has been achieved with OT than with nanopores [34]. However, these methods generally require more preparative steps and far fewer molecules can be analyzed over a common time period when compared to nanopore-based assays.

1.3 Dissertation outline

This research presents a hardware/software infrastructure for the automated control of DNA in a nanopore using FSM-based control. The dissertation begins with an explanation of the FSM infrastructure followed by several experiments that test its operation. The success of these experiments prove the utility of FSM-based control in nanopore research.

Chapter 2 discusses the design choices made during the design of the FSM infrastructure and elaborates on the different types of inputs available for triggering an action by the FSM. The FSM runs on FPGA hardware controlled by LabVIEW software (Version 8.0, National Instruments, Austin, TX). A control panel running on a host PC downloads the FSM to the FPGA and manages the starting and stopping of the FSM. The `FSMCreator` program facilitates FSM design input for use on the FPGA.

The dissertation continues with Chapter 3 where several tests of the FSM's control authority over DNA molecules captured in the nanopore are described. DNA hairpins with single-stranded overhangs are captured in the pore and are detected and reacted to within 2 ms. Two reaction techniques are explored: ejection of DNA immediately upon detection and detection of DNA with subsequent voltage reduction until translocation.

Chapter 4 represents the culmination of the research and describes extensions of the control demonstrated in Chapter 3 to capture and tether of a single DNA molecule in the pore. Once captured and secured in the pore using duplex DNA at either end, controlled binding and subsequent dissociation from KF is achieved through FSM-based control. Varying the DNA's exposure time to enzyme yielded information about KF's binding rate under the conditions present during the experiment.

The conditions and specific reagents used in the research are covered in Chapter 4.2. Chapter 5 restates the motivation of the work and suggests some future directions given the present state of the research presented in the dissertation.

Appendix A.1 outlines the process of inputing and running an FSM on the FPGA via LabVIEW and `FSMCreator`. Appendix B.1 describes the data analysis process using `eventDetector`. The `eventDetector` software parses the recorded data from a nanopore experiment and extracts DNA translocation events, computing the amplitude and duration for each. Only initial user interaction is required and `eventDetector` automates many of the tasks that needed to be performed manually with previous data analysis workflows.

Chapter 2

Design and Implementation of Finite State Machine-based Voltage Control

2.1 Introduction

Investigation of single-molecule phenomenon using the nanopore setup requires rapid execution of voltage control logic. Computer hardware dedicated to the task of monitoring the nanopore ionic current and applying command voltage changes must operate faster than the time scales of enzymatic function. This hardware platform also needs to be flexible so that as new experimental procedures are devised, the hardware remains usable. The field-programmable gate array (FPGA) platform is a relatively low-cost solution ($5,000) that offers the performance of hardware yet allows future redesign capabilities, thus providing an excellent tool for executing finite state machine (FSM) control logic. An FSM can be designed to perform simple or complicated tasks and controls the execution of commands based on a set of inputs and the current state [18].

2.2 System overview

National Instruments' LabVIEW software (Version 8, National Instruments, Austin, TX) approaches laboratory instrumentation in a non-traditional manner. A graphical user interface (GUI) runs on a personal computer (PC) and is programmed to control a laboratory device. The GUI is referred to as a *virtual instrument* (VI) and consists of two parts: the front panel and the block diagram. The front panel serves as the user interface for the VI and emulates the appearance of an actual bench top appliance. The block diagram defines the functionality of the VI and allows programming by means of a graphical schematic editing interface. Programming simple operations such as adding two variables can be cumbersome using the block diagram interface but begins to show its strengths when implementing more complicated operations. For example, setting up a first in, first out (FIFO) queue for communication with the FPGA becomes a trivial operation in LabVIEW. The same functionality would take significantly more development time using a hardware description language such as VHDL or Verilog.

We chose a National Instruments FPGA development board, PCI-7830R, that interfaces with a LabVIEW VI running on the host PC via the PCI bus. Data heavy communication such as logging channel or FSM data can be done with little CPU overhead using a direct memory access (DMA) FIFO. The DMA FIFO requires specialized

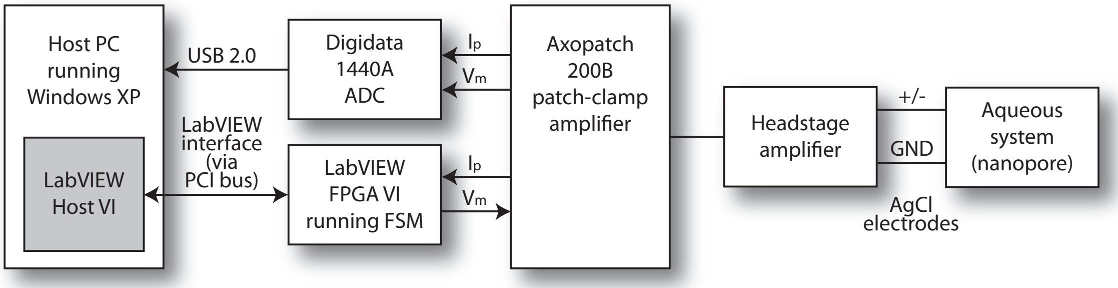
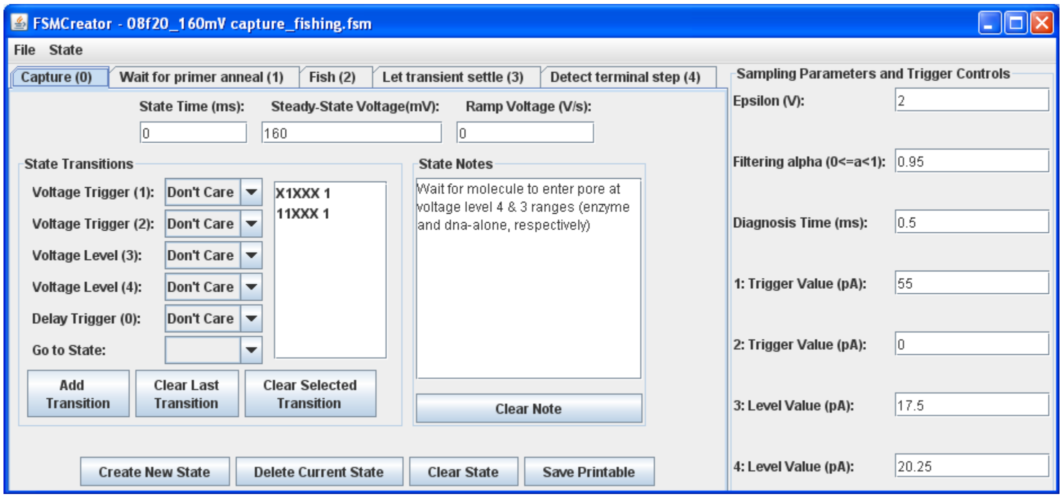


Figure 2.1: Top level schematic of the nanopore setup. A patch-clamp amplifier measures the ionic current through the pore while applies a command voltage across the pore. The digitizer logs voltage and current signal. The PC hosts the FPGA running the FSM and manages the FSM configuration.

LabVIEW system calls that may not be well suited for performing one-time read and write operations; the case when modifying FSM parameters at run time. For passing these parameters back and forth between the host and the FPGA, the front panel inputs on the host PC can be connected to inputs of the VI running on the FPGA. Data transmission over this channel is subject to operating system interrupts and not time critical, so only data that can afford to be delayed is sent using this method. Figure 2.1 provides a high level schematic of the nanopore setup.

2.3 Finite state machine design and execution

The FSM definition file (`.fsm` file extension) defines the sequence of possible state executions for a given FSM. These files were originally written using a plain text format and parsed using a script written using the Ruby scripting language for use with our platform. A SURF-IT 2007 undergraduate researcher (Ayla Solomon), adapted the



State Time (ms): 0
Steady-State Voltage(mV): 160
Ramp Voltage (V/s): 0

Voltage Trigger (1): Don't Care
Voltage Trigger (2): Don't Care
Voltage Level (3): Don't Care
Voltage Level (4): Don't Care
Delay Trigger (0): Don't Care
Go to State:

X1XXX 1
11XXX 1

Wait for molecule to enter pore at voltage level 4 & 3 ranges (enzyme and dna-alone, respectively)

Epsilon (V): 2
Filtering alpha (0<=a<1): 0.95
Diagnosis Time (ms): 0.5
1: Trigger Value (pA): 55
2: Trigger Value (pA): 0
3: Level Value (pA): 17.5
4: Level Value (pA): 20.25

Figure 2.2: The `FSMcreator` program window allows creation of states and transitions between states through a tabbed interface. Each tab represents a different state and provides controls for changing the properties of each state of the FSM.

Ruby script into a GUI program, written in Java, to facilitate the creation of FSM designs and reduce the chances of user error. Figure 2.2 shows the GUI with an FSM loaded. Appendix A.3 provides detailed instruction for creating FSM designs using `FSMcreator`.

Each FSM can have a maximum of 256 states where transitions between each state trigger off a combination of five inputs: two basic current level triggers, two current level detection inputs, and a state timer input. Outputs can be constant or ramping voltages. The FSM is made up of an input and an output stage running in parallel with each other. Analysis of the current and next state lookup occurs in the input stage while digital-to-analog conversion of the state output occurs in the output stage.

2.3.1 Input stage

The input stage of the FSM contains all the logic for establishing a true/false value for the inputs and the next state lookup. Each input has a triggering threshold or amplitude with which the analog signal is compared. Depending on the value of these thresholds and the current signal, the resulting input will return a true or false value in the form of a `1` or a `0` bit value. The combination of these binary values and the FSM's current state determine the next state the FSM will execute.

The input stage runs at 188.68 kHz, primarily limited by the 5 μs conversion time of the analog-to-digital converter (ADC). During each cycle of the input stage, the ADC samples the channel current. The analog to digital conversion finishes by the beginning of the next cycle and is used for next state lookup during the next cycle.

2.3.1.1 Low-pass filtering

Signal filtering was done on the FPGA inside the input stage of the FSM using a digital approximation of a single-pole low-pass RC filter. From Kirchoff's laws, the equation,

$$V_{\text{in}}(t) - V_{\text{out}}(t) = RC\frac{dV_{\text{out}}(t)}{dt}, \qquad (2.1)$$

shows the voltage difference between the input and output of the filter. Converting to the discrete time domain and reducing Eq. 2.1 yields an algorithm usable onboard the

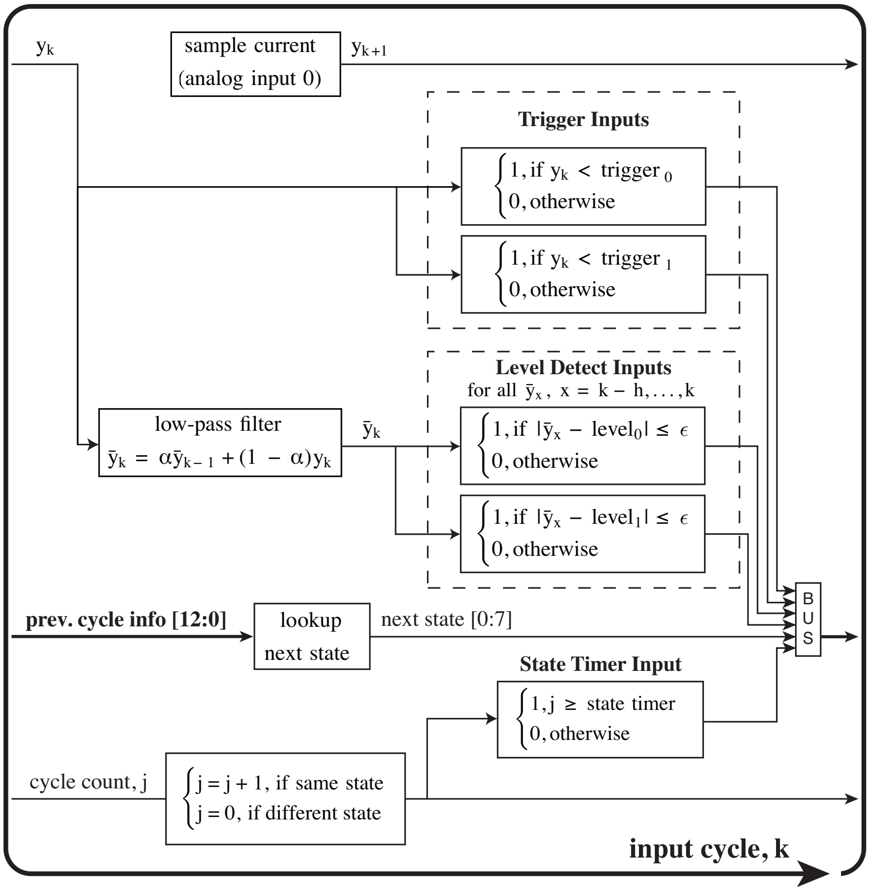

y_k
sample current
(analog input 0)
y_{k+1}
Trigger Inputs
$\begin{cases}1, \text{if } y_k < \text{trigger}_0 \\ 0, \text{otherwise}\end{cases}$
$\begin{cases}1, \text{if } y_k < \text{trigger}_1 \\ 0, \text{otherwise}\end{cases}$
Level Detect Inputs
for all $\bar{y}_x$, $x = k - h, \ldots, k$
low-pass filter
$\bar{y}_k = \alpha\bar{y}_{k-1} + (1 - \alpha)y_k$
$\bar{y}_k$
$\begin{cases}1, \text{if } |\bar{y}_x - \text{level}_0| \leq \epsilon \\ 0, \text{otherwise}\end{cases}$
$\begin{cases}1, \text{if } |\bar{y}_x - \text{level}_1| \leq \epsilon \\ 0, \text{otherwise}\end{cases}$
prev. cycle info [12:0]
lookup
next state
next state [0:7]
BUS
State Timer Input
$\begin{cases}1, j \geq \text{state timer} \\ 0, \text{otherwise}\end{cases}$
cycle count, j
$\begin{cases}j = j + 1, \text{ if same state} \\ j = 0, \text{ if different state}\end{cases}$
input cycle, k

Figure 2.3: The input FSM controls the flow of execution from the current state to the next based on the inputs. a) The input FSM samples the ionic current every 5.3 μs for subsequent state calculation during the next cycle.

FPGA:

$$y_k - \bar{y}_k = RC\frac{y_k - y_{k-1}}{\Delta t}, \tag{2.2}$$

$$\bar{y}_k = \frac{\Delta t}{\Delta t + RC}y_k + \frac{RC}{\Delta t + RC}\bar{y}_{k-1}, \tag{2.3}$$

where y_k represents the raw data sample, $\bar{y}_k$ the filtered data sample for cycle k of the input stage. The sample period Δt for our system is 5.3 μs. Substituting $\frac{RC}{\Delta t+RC} = \alpha$

reduces the equation further and provides a single tuning parameter α:

$$\bar{y}_k = (1-\alpha)y_k + \alpha\bar{y}_{k-1}. \tag{2.4}$$

The filtering parameter $\alpha = [0,1) \in \Re$ adjusts the amount of filtering; $\alpha = 0$ applies no filtering with the amount of filtering increasing as $\alpha \to 1$. The $\tau = RC$ time constant can be calculated as $RC = \frac{\Delta t}{1-\alpha} - \Delta t$. Figure 2.5 illustrates the difference between the raw current and filtered current signals.

2.3.1.2 Trigger level inputs

The trigger level inputs monitor the ionic current signal and compare it against a preset amplitude threshold. At the beginning of each input cycle, the FSM compares the latest current sample to the trigger threshold and if less than the threshold, the operation outputs a `1` to indicate the current amplitude has crossed the threshold. To prevent rapid trigger toggling due to a signal passing near the threshold, a simple hysteresis was added: the signal is required to pass the threshold plus an offset before triggering. Figure 2.4 illustrates where the trigger would occur given a noisy input signal. The trigger level inputs can react within two cycles (10.6 μs) of a current change; useful for rapidly detecting gross changes in current.

2.3.1.3 Current level inputs

Using the current trigger inputs to detect subtle changes in the current signal can lead to unreliable diagnosis due to false triggering off of signal noise. This effect is

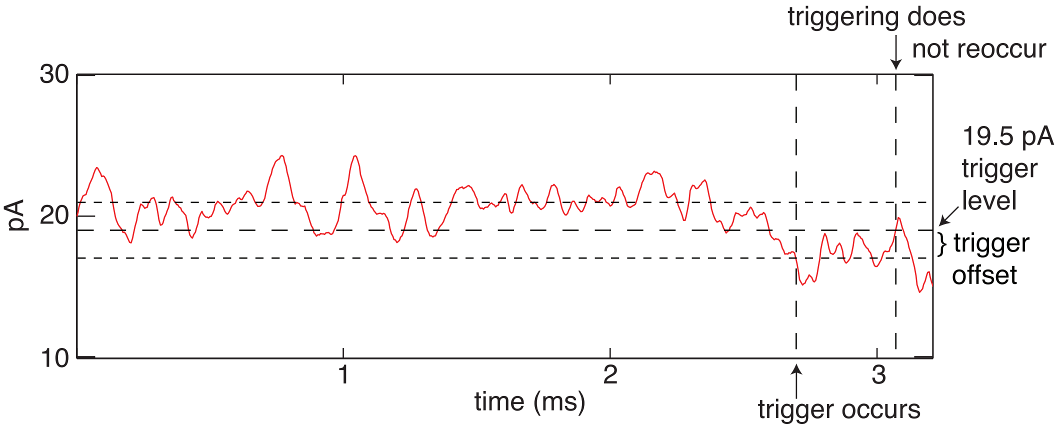


Figure 2.4: Trigger level inputs monitor the raw current signal and wait for the signal to drop below a threshold. A trigger offset minimizes the possibility of undesired trigger toggling by requiring the signal to pass the trigger offset before a trigger is considered to have occurred. A trigger will not re-occur when the signal passes back above the threshold; the signal must rise above the threshold plus offset to re-trigger.

mitigated by using the current level inputs. The current level inputs monitor a low-pass filtered version of the current signal and waits for the current amplitude to enter the detection zone, a preset amplitude plus/minus a tolerance parameter ϵ. Upon detection of this amplitude, the FSM waits for the current to stay within the detection zone for an period of time referred to as the *diagnosis time* before outputting a `1`. The diagnosis time is a global parameter of the FSM and can be set to a maximum of 22763.33 s. Figure 2.5 shows how the current level inputs use the filtered signal for detecting current amplitude changes.

2.3.1.4 State timer input

The state timer input tracks the time spent in the current state and returns a `1` after execution has remained in the same state for longer than the current state's

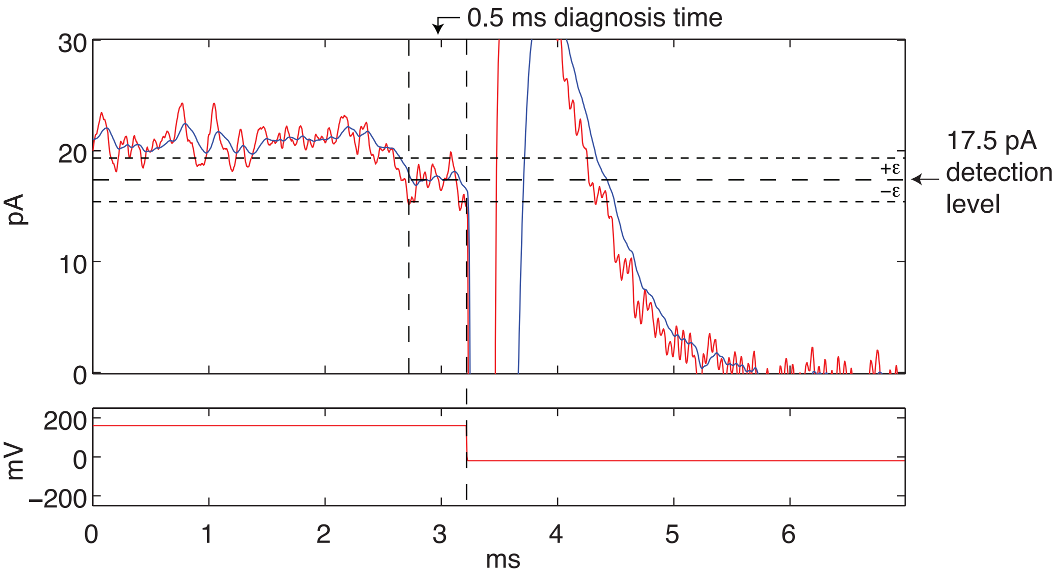


Figure 2.5: Current level detection inputs use a low-pass filtered version of the current signal (blue) to reduce the effect of noise spikes on detection. The figure above illustrates the current level detect input set to an amplitude of 17.5 ± 2 pA with a diagnosis time of 0.5 ms. In this data, enzyme dissociation from DNA is detected. Note that several times the raw current signal (red) passed into the detection zone and could have caused a false detect if not filtered.

`delay value`, defined for each state in the `.fsm` file. Upon transitioning into a new state, the FSM sets `state timer` to zero. Over the course of execution of that state, the FSM keeps track of the number of cycles spent in that state, `cycle count`, and increments the state timer based on the following schedule:

1. Increment `cycle count` once per cycle until `cycle count` > `delay multiplier`.
2. Increment `state timer` by one.
3. If `state timer` ≤ current state's `delay value`, return to Step 1. Otherwise, execution of the state has exceeded `delay value` and the state timer input now returns `1`.

The state timer input forces states to run for a pre-specified amount of time before moving to another state. The state timer can run for up to 22763.33 s.

2.3.2 Output stage

The output stage of the FSM manages the digital-to-analog converter (DAC) that commands the voltage applied across the pore by the patch-clamp amplifier. The output voltage signal can take the form of either a constant output or a ramp. For a faster voltage update interval, the output stage runs four times faster than the input stage (754.72kHz) which minimizes stepping artifacts that occur due to how the ramp signal is generated.

For constant voltage signals, -10 V to 10 V can be applied by the FPGA. This translates to an applied voltage range of either ± 200 mV (front switched) or ± 1 V (rear switched) depending on the gain setting of the external command of the Axopatch 200B patch-clamp amplifier.

Application of constant slope voltage (ramp) enables nanopore experiments to collect more detailed biological information than possible with constant voltage [16, 23, 32]. The (digitized) ramp signal is made up of many small voltage steps that when output in rapid succession produce a smooth signal, as illustrated in Fig. 2.6. When applying a ramp output, the output stage uses the `step increment` and `step period` parameters to define the shape of the ramp signal. The output stage increments the signal by `step increment` bits, which can be positive or negative for increasing or decreasing ramps, every `step period` cycles of the output stage (1.325 μs). The

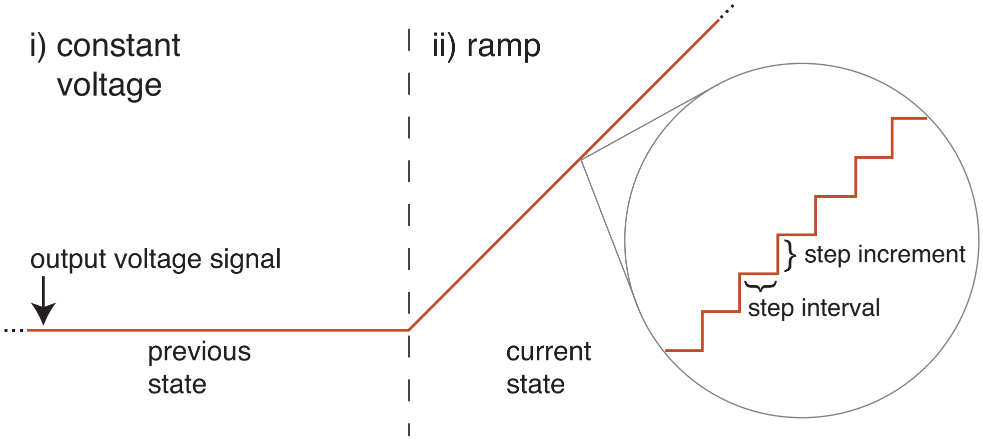


Figure 2.6: The FSM can output either a i) constant voltage or a ii) ramping voltage. For ramping voltages, the signal is made up of very small steps starting at the output amplitude for the current state and changing by `step increment` every `step interval` cycles.

voltage ramp's amplitude begins at the output voltage defined for the state. A zero `step increment` value implies a constant output.

The `step increment` and `step period` parameters are calculated automatically by `FSMcreater` using a brute force optimization that minimizes `step increment`. Starting with low values, the algorithm cycles through values of `step increment`. For each `step increment` value, the algorithm cycles through every possible value of `step period` before incrementing `step increment`. Once the squared error between the calculated ramp rate and the desired ramp rate is less than 0.05, the optimization stops. The 0.05 epsilon was determined experimentally.

2.4 Saving and downloading finite state machine designs using FSMcreator and LabVIEW

After creating a design in the FSMcreator GUI, two files are created: one with a .fsm extension and one with a .fpga extension. The .fsm file saves the FSM in a format that FSMcreator can understand and read back in when making changes to the design. The .fpga file stores the FSM in the format the FSM uses so downloading to the FPGA requires no further conversion.

The memory on the FPGA is organized into six sections to hold all the information necessary for a FSM to run: next state, delay value, delay multiplier, output voltage, step period, and step increment. For the next state lookup, the FSM concatenates the results of the five inputs (5 bits) and the current state (8 bits) in the following order (highest order bit first):

$$\begin{aligned}\textbf{cycle info}[12:0] = [\ & \textbf{trigger input}_0, \\ & \textbf{trigger input}_1, \\ & \textbf{level detect input}_0, \\ & \textbf{level detect input}_1, \\ & \textbf{current state}[7{:}0], \\ & \textbf{state timer input}\ ].\end{aligned}$$

The lookup of all the other parameters uses the current state plus an offset that depends on the desired parameter. Fig. 2.7 shows the layout of the various parameters in memory

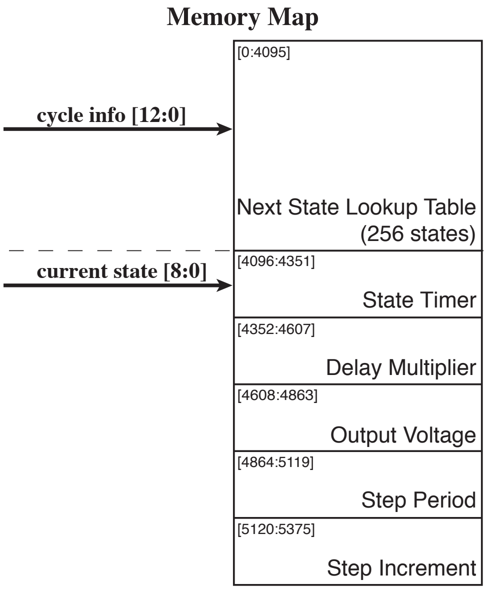
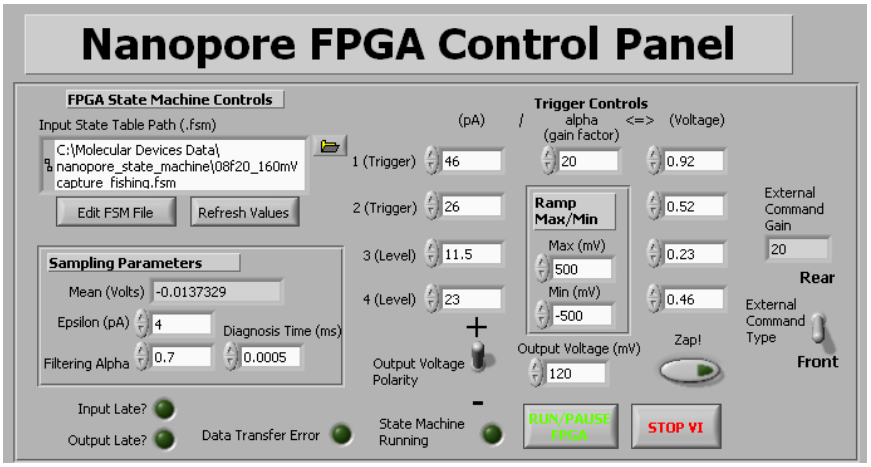


Figure 2.7: Memory organization on the FPGA. The 13 bits of information carried through each cycle of the FSM serve as the lookup address for the next state. The additional state parameters use the current state plus an offset as a lookup address.

and the address for each section of memory.

FSM design download takes place in two parts: download and verification. During the download step, the host VI iterates through each 16-bit word in the `.fpga` file and transmits it to the appropriate address in FPGA memory. Next, the host VI waits for acknowledgment from the FPGA. Once a write has been acknowledged, the host then queues up the next word and continues the process until the entire FSM design has been downloaded to the FPGA. To minimize download time, the host VI only sends non-zero memory contents. Download time scales with the number of states in a FSM, though for most typical designs encountered thus far, download is instantaneous from the point of view of the user. The download process then proceeds in reverse with the



Nanopore FPGA Control Panel

FPGA State Machine Controls

Input State Table Path (.fsm): C:\Molecular Devices Data\nanopore_state_machine\08f20_160mV capture fishing.fsm

Sampling Parameters

Mean (Volts): -0.0137329
Epsilon (pA): 4
Filtering Alpha: 0.7
Diagnosis Time (ms): 0.0005

Trigger Controls

	(pA)	/ alpha (gain factor)	<=> (Voltage)
1 (Trigger)	46	20	0.92
2 (Trigger)	26		0.52
3 (Level)	11.5		0.23
4 (Level)	23		0.46

Ramp Max/Min
Max (mV): 500
Min (mV): -500

Output Voltage (mV): 120

External Command Gain: 20

Figure 2.8: The LabVIEW front panel manages all the top-level actions of the FSM running on the FPGA such as loading the `.fsm`/`.fpga` files, downloading the FSM to the FPGA and setting amplifier and external command gain, to name a few. When a FSM is not running, the *Output Voltage (mV)* text box controls the voltage applied to the pore.

FPGA serving the memory contents to the host for verification of the download. This process of downloading and verifying is performed every time the FSM is started to allow for updates to the FSM in between runs.

2.5 Controlling finite state machine through LabVIEW front panel

Basic control of the patch-clamp amplifier, such as commanding constant voltages, can be performed from within the FPGA/FSM front panel in LabVIEW, Fig. 2.8, without designing and loading a FSM. A detailed description of the controls on the LabVIEW front panel and how to use them is available in Appendix A.1.

Chapter 3

Detection and Ejection of Captured DNA Molecules

3.1 Introduction

This chapter presents an approach for automated detection and manipulation of DNA hairpins, an extension of the control results presented in [4]. Rapid detection (< 2 msec) is based on computing a filtered mean amplitude of the ionic current in real time, and monitoring the mean relative to an amplitude range consistent with DNA hairpin blockades (20 ± 2.8 pA). Upon detection, two methods of voltage control are demonstrated. In method 1, dwell time extension is achieved by prompt voltage reduction, with the reduced voltage applied until the hairpin unzips. A higher relative voltage for capture increases the number of molecules examined, and the reduced voltage post-capture increases the dwell time which could be used to extract kinetic information about the hairpin dissociation process, for example.

Extending the life of DNA hairpins in the pore also increases the time within which a terminal base identification could be acheived using machine learning methods [44]. In method 2, voltage reduction is applied for a preset time (10 msec) followed by voltage reversal to expel the molecule prior to hairpin unzipping. This demonstrates our control authority to aggregate the dwell times of hundreds of blockade events. Additionally, it complements prior work [4], confirming the ability to detect DNA-enzyme blockades and DNA hairpin blockades. Confirmation of the ability to discern between each blockade type in real time is addressed in the next chapter. Generally, nanopore-based characterization of enzyme dynamics will require direct detection and control of multiple DNA conformations relative to the enzyme, and the direct control of enzyme-free DNA presented in this chapter is a prerequisite toward developing this capability.

3.2 Control logic setup

Blockade events, quantified by the blockage current and dwell time, can be detected and monitored in real time using the FSM/FPGA. The nanopore system was setup as described in Materials and Methods (Chapter 4.2) with exception that data was sampled at 50 kHz and the FPGA employed a mean filtering scheme rather than a low-pass filter. The mean filter applied to the incoming current signal on the FPGA removed a large portion of the peak-to-peak noise. Specifically, every 5.3 μsec, the FPGA sampled the ionic current and computed a windowed mean amplitude based on the previous 0.75 ms of signal. Every 0.2 ms, the FPGA tested if the mean was within 20±2.8 pA (17.2 to 22.8 pA range). The basis for choosing this range was that ∼20 pA is the median amplitude for DNA 20 base pair hairpin events at 180 mV. If the mean entered and remained within the detection range for four consecutive tests, the FSM logic diagnosed the blockade as a DNA hairpin event. The *nominal detection time*, between DNA translocation event and diagnosis of the event, was 2.0 ms; 0.75 ms for the windowed mean to first enter the 17.2 to 22.8 pA range, 0.6 ms for three more confirmed tests, and 0.65 ms of delay[1].

3.3 Results and discussion

In the first experiment, the objective was to efficiently detect individual DNA hairpin events, and increase the blockade dwell time by lowering the applied voltage from 180 mV to 150 mV upon detection. This is referred to as **dwell time extension control**. The next goal was to aggregate the extended blockade dwell times, by expelling the DNA using voltage reversal of -50 mV after 10 ms at 150 mV. This is referred to as **dwell time aggregation control**. The motivation was to increase the nominal hairpin dwell time, and expel the molecule before unzipping the hairpin. A typical 20 bphp event at constant 180 mV voltage is shown in Figures 1.2 and 3.1aI.

The probability histogram of the base 10 logarithm of dwell time (Figure

[1] Certain inefficiencies in FPGA signal routing into the sampling loop caused the additional 0.65 ms of delay in the reaction time. By bringing global signals inside the sampling loop, the delay has since been eliminated, reducing detection time to 1.4 ms.

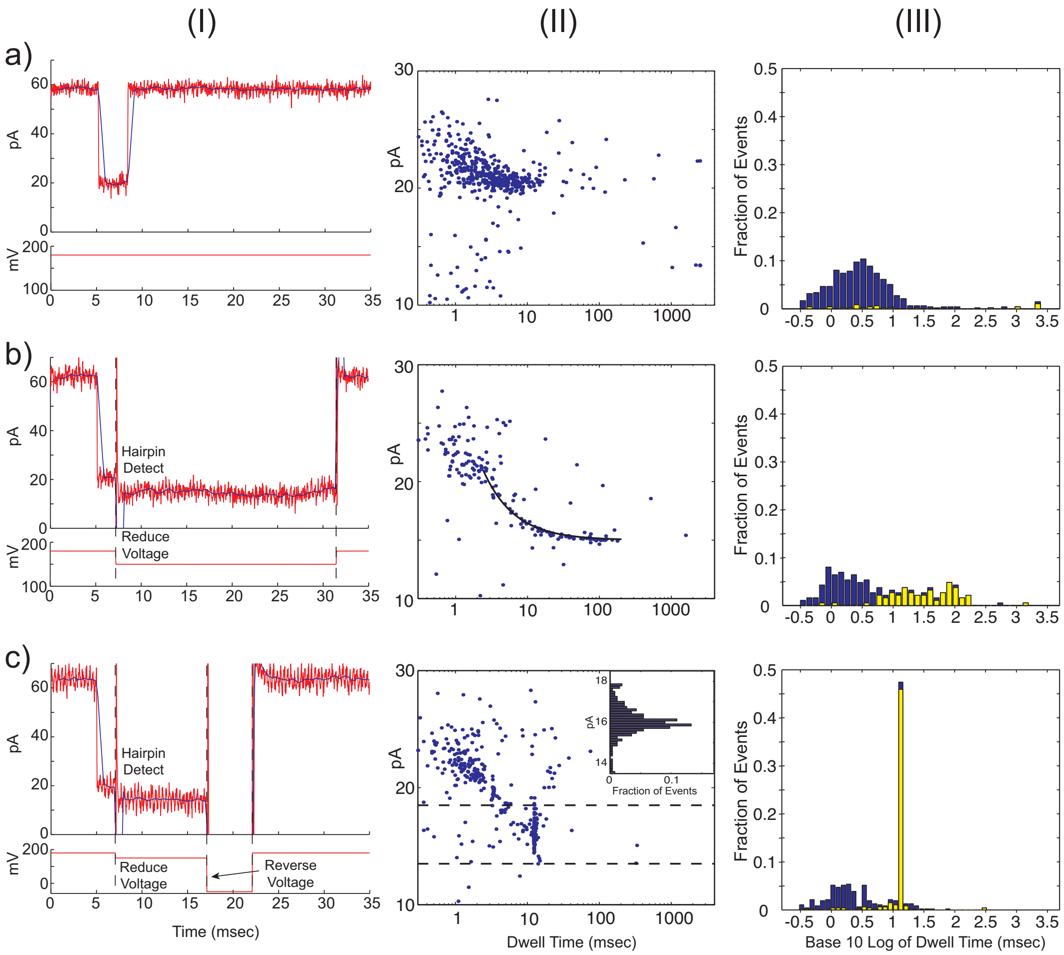


Figure 3.1: Regulation of 20 bphp dwell time using FSM control. (I) The red current signals are low-pass filtered at 5kHz, the blue signal is a mean filtered current, and the red voltage signal is the commanded voltage. Typical events and corresponding voltage signals under a) constant 180 mV voltage, b) dwell time extension control, and c) dwell time aggregation control. (II) Event plot of DNA events, showing average amplitude vs. dwell time for each event (point). Equation (3.1) (line) fit to events in bII), and amplitude histogram for events within 13-18 pA (dashed line) range in cII). (III) Probability histograms of the base 10 logarithm of dwell time for all events (blue), and for subset of events in 13-18 pA range (yellow).

3.1aIII, blue) is unimodal, with median dwell time of 2.8 ms. The median amplitude of the event plot in Figure 3.1aII is 20.9 pA with an interquartile range (IQR) of 1.7 pA. Only 6% of events are in the subset range of 13-18 pA (3.1aIII, yellow). For the same experiment at constant 150 mV voltage (data not shown), the events cluster around a median amplitude of 15 pA and 87% of 150 events are in the 13-18 pA range. Thus, under extension and aggregation control for which the voltage is reduced to 150 mV for all detected events, a larger percentage of blockades should have a mean amplitude within the 13-18 pA range.

3.3.1 Dwell time extension

Upon diagnosis of a DNA hairpin event using the mean filtered current, the command voltage is reduced to 150 mV until the hairpin unzips and the DNA translocates through the pore. Using 180 mV for capture results in more events than 150 mV, while reducing to 150 mV extends the life of the hairpin. Dwell time extension is useful for terminal base-pair sequencing by machine learning methods [44]. After each translocation, the FPGA resets the voltage to 180 mV. A representative event is shown in Figure 3.1bI. The event plot (Figure 3.1bII) pattern shows that events faster than the nominal diagnosis time of 2.0 ms are unaffected by extension control, and events with longer dwell times converge to the ∼15 pA mean amplitude as expected. The concave trend is also consistent with an equation for event's mean amplitude vs. dwell time. In particular, for an event at 21 pA (median amplitude at 180 mV) for 2.4 ms [2], and at 15 pA (median amplitude at 150 mV) for x ms, an approximate mean amplitude $\bar{I}$ is

[2] Step changes in voltage induce a capacitive transient, and the transient at the end of each event is ∼0.4 ms for changing from 150 mV to 180 mV. Thus, 2.4 ms at 21 pA is 2.0 ms of detection time and 0.4 ms of transient time. While the 0.4 ms transient varies in amplitude, assuming 21 pA is sufficient for line fitting.

$$\bar{I} = \frac{2.4 * 21 + 15 * x}{2.4 + x}. \tag{3.1}$$

When $x \approx 24$ ms, as in Figure 3.1bI, $\bar{I} = 16$ pA. Equation (3.1) closely matches the mean amplitude vs. dwell time data (Fig. 3.1bII). Also, the fraction of events within the subset range 13-18 pA increased to 41%, as shown in the yellow histogram overlaid on the blue probability histogram (Fig. 3.1bIII).

3.3.2 Dwell time aggregation

The objective was to aggregate the dwell times of the extended events by applying 150 mV for 10 ms upon diagnosis of a hairpin event, followed by voltage reversal of -50 mV for 5 ms. The reversal time of 5 ms is known to be sufficient to clear the DNA from the channel, prepping the pore for the next event. Aggregation control would imply a measure of control over the distribution of the events, in addition to temporal control of individual molecular events. A representative event is shown in Figure 3.1cI. As before, the event plot (Fig. 3.1cII) pattern shows that events faster than the nominal diagnosis time of 2.0 ms are unaffected by aggregation control. Within the subset range of 13-18 pA, the median amplitude is 16 pA with 0.7 pA IQR (amplitude histogram shown in Fig. 3.1cII). The 16 pA median is consistent with (3.1), since for $x = 10.0$ ms, $\bar{I} = 16$ pA. Also in the subset range, and the median dwell time is 12.4 ms with 0.1 ms IQR. The low IQR indicates a high degree of control over the distribution of events that extend to at least 10 ms at 150 mV. The median dwell time of 12.4 ms

Table 3.1: Summary statistics for DNA molecule capture experiments (control, dwell time extension, dwell time aggregation).

Figure No.	**No. of Events**	**Median Dwell Time, ms**	**IQR, ms**
3.1aIII[a]	472[b]	2.8	4.2
3.1bIII[c]	76[d]	31.6	62.0
3.1cIII[c]	256[e]	12.4	0.1

[a] Blue histogram, for events within 10 to 30 pA range.

[b] 6% (27 events) within subset 13-18 pA range.

[c] Yellow subset histogram, for events within 13-18 pA range.

[d] 41% of the 187 events within 10 to 30 pA range.

[e] 55% of the 466 events within 10 to 30 pA range.

is commensurate with 2.0 ms of detection time, 10 ms at 150 mV, and 0.4 ms due to a transient that is included at the end of each event resulting from voltage reversal[3].

Summary statistics for the histograms in Figure 3.1III are reported in Table 3.1.

In [32], the authors characterize hairpin unzipping at a set of constant voltages by fitting a curve to an unzipping probability data profile. Specifically, for an unzipping time t at voltage V, the unzipping probability is the fraction of events with dwell time less than t, divided by the total number of events[4]. For a set of t values, the unzipping probability data profile is shown in Fig. 3.2 for our experiments at 180 mV constant, under extension control, and at 150 mV constant.

As in [32], fitting a line to the data reveals a characteristic unzipping time τ_{u}^{V} for constant voltage V. For amplitude range 10-30 pA and dwell time range 0.3-500

[3] The transient due to the 180 mV to 150 mV change is included within the 10 ms waiting time under aggregation control.

[4] The authors formulate an alternative but equivalent definition for unzipping probability.

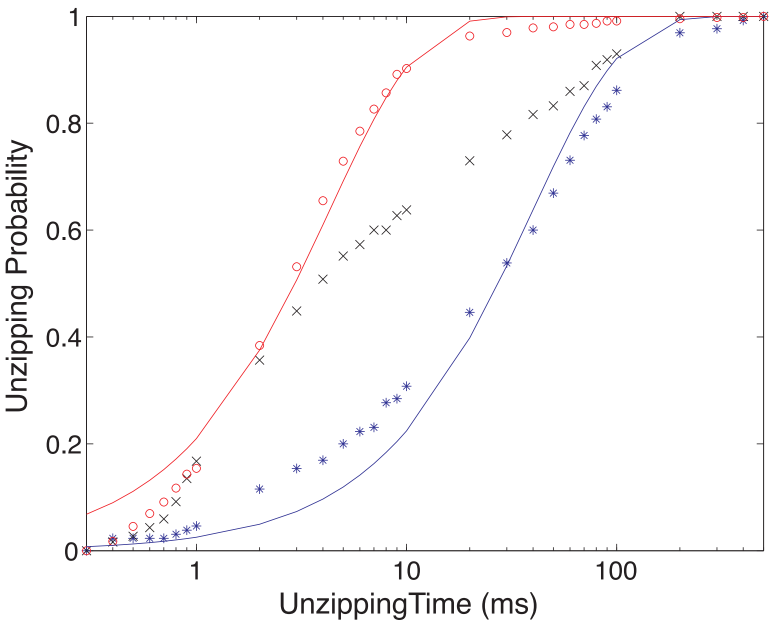


Figure 3.2: Unzipping probability data profile, defined as fraction of events with dwell time less than each unzipping time t, and line $1 - \exp[t/\tau_{\mathrm{u}}^{V}]$ fit to profile for constant voltages $V = 180$ mV (red) and $V = 150$ mV (blue). Characteristic unzipping time constant τ_{u}^{V} at constant voltage V is generated by fit. Symbols: $\circ$ for $V = 180$ mV, $\times$ for extension control (transitions from 180 mV to 150 mV), and $*$ for $V = 150$ mV.

ms, it is revealing to compare the median dwell times with the fitted τ_{u}^{V} constants. For $V = 180$ mV, the median is 2.6 ms and $\tau_{\mathrm{u}}^{180} = 4.2$ ms. For $V = 150$ mV, the median is 25.2 ms and $\tau_{\mathrm{u}}^{150} = 39.4$ ms. The data trimming has a significant affect over the quality of the fit to the data, and consequently over the value for τ_{u}^{V}. In contrast, the median does not vary as much, suggesting a sensitivity of τ_{u}^{V} to outliers, in addition to the fitting method used. For example, for a dwell time range of 0.3-4000 ms at $V = 150$ mV, the median is 34.9 ms (+ 9.7) and $\tau_{\mathrm{u}}^{150} = 55.7$ ms (+ 16.3). Careful selection and analysis of statistical models appropriate for the data (with outliers always present) is a crucial part of nanopore data analysis.

3.4 Summary

This work has shown that single DNA hairpin molecules captured in a biological nanopore can be detected and reacted to using a finite state machine implemented on a field-programmable gate array. The dwell time of such translocation events can be extended to gain more signal, which can in turn be analyzed offline using machine learning methods to yield terminal base-pair specific signatures. The signatures can then be used for real-time identification of terminal base pairs. Additionally, the finite state machine is capable of ejecting a molecule from the pore after it has been detected but prior to unzipping the hairpin. Rapid DNA hairpin detection ($<$ 2 msec) relied on a mean filtered amplitude, which was required to remain within a preset amplitude range ($<$ 6 pA in spread) for multiple consecutive threshold comparisons. The method was refined to differentiate DNA-enzyme blockades from DNA alone blockades in real time, as detailed in the next chapter.

Chapter 4

Electronic Control of DNA Polymerase Binding and Unbinding to Single DNA Molecules

This chapter describes a nanopore-based technique for detecting and manipulating individual KF molecules as they bind to DNA. We recently characterized the current signatures of KF binary and ternary complexes captured in the nanopore [4]. We used DNA template or hairpin substrates bearing a 2-3 dideoxycytidine (ddC) terminus, which permit ternary complex formation without catalytic turnover. While DNA primer/template or hairpin substrates absent KF yield fast events (Fig. 4.1i), capture events for KF-DNA binary complexes (Fig. 4.1ii) or KF-DNA-dNTP ternary complexes (Fig. 4.1iii), are longer and display two current levels. The initial current level corresponds to the enzyme bound state (EBS) of the captured DNA, with the complex

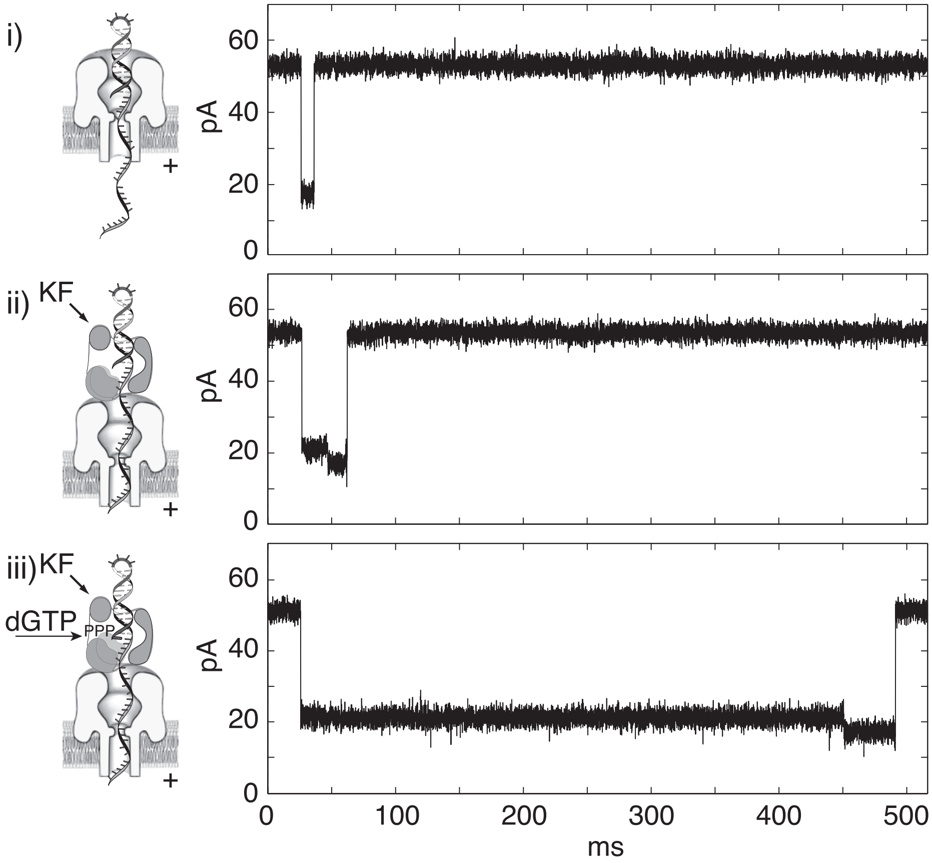


Figure 4.1: KF binding to individual DNA molecules captured in a nanopore. Illustrations and representative current traces for nanopore capture events of (i) unbound DNA hairpin, (ii) DNA-KF binary complex, and (iii) DNA-KF-dNTP ternary complex.

perched atop the pore vestibule and the single-stranded template threaded through the nanopore lumen (Fig. 4.2i). This first EBS segment is longer for ternary complexes than for binary complexes (Fig. 4.1ii,iii). The second current segment results upon voltage-promoted enzyme dissociation, when the duplex DNA drops into the nanopore vestibule (Figs. 1.4ii) followed by duplex unzipping and translocation through the pore (Figs. 1.4iii). The duration of this terminal current step is the same as the duration of capture events for the DNA substrate alone, as a function of both applied voltage and duplex length [4]. In other words, lower voltage and longer duplexes extend complex dwell time.

In this study, we employed this two step current signature for DNA-KF complexes (Figs. 1.4, 4.1iii) to develop repetitive, real-time detection and control of KF association and dissociation from single DNA molecules. Individual primer/template DNA molecules were topologically tethered with their ssDNA segment threaded through the nanopore and duplex segments on either side of the pore (Figure 4.2). We tethered individual primer/template DNA molecules with their ssDNA segment threaded through the nanopore. We applied finite state machine (FSM) logic implemented on a field programmable gate array (FPGA) to monitor current levels and control voltage with sub-millisecond precision in response to current changes. We exploited the complementarity of the α-HL nanopore structure to the ds-ssDNA junction to achieve control of voltage promoted DNA-KF dissociation. By varying the duration of exposure to KF and dNTP, we probed the assembly of DNA-KF complexes, and the effect of dGTP on the rate of assembly.

4.1 Results and Discussion

Our objective was to measure and control iterative KF dissociation and reassociation to an individual DNA molecule tethered in the α-HL nanopore. To achieve this, we exploited the current change that accompanies KF dissociation from its DNA substrate (Figs. 1.4, 4.1iii). Using an FSM implemented on an FPGA to exert active voltage control [4, 23, 42], we reasoned that we could diagnose the EBS of captured complexes. We could then monitor the current level to detect and react to the terminal current step with a voltage reversal executed after KF dissociation, but before DNA translocation through the nanopore.

Specifically, our strategy was to capture a 23 base-pair DNA hairpin substrate bearing a $2'$-$3'$ dideoxy terminus and a $5'$ overhang from the *cis* chamber, and promptly lower the voltage to a level sufficient to hold the duplex in the nanopore vestibule with its single-stranded end protruding into the *trans* chamber (Fig. 4.2i). This allows a 20-mer oligonucleotide to anneal to the $5'$ end of the DNA, tethering it in the nanopore by duplex regions at both ends (Fig. 4.2ii). In a step termed ‘fishing’, a small negative voltage is then applied for a programmed time interval to expose the double-strand/single-strand junction of the DNA to the bulk phase in the *cis* chamber (Fig. 4.2iii). The *cis* chamber contained either DNA alone, DNA and KF, or DNA, KF, and dGTP (the complement to the template base at $n = 0$). In a step termed ‘probing’, a positive voltage is applied next, drawing the double-strand/single-strand junction to the nanopore (Fig. 4.2iv). By monitoring the ionic current, the FSM determines if KF is bound to the DNA. If KF binding is not detected, the FSM reverses the voltage to fish again. If KF binding is diagnosed, the probing voltage is maintained until detection of the terminal step that reports enzyme dissociation. A voltage reversal can then be executed before the DNA molecule translocates through the nanopore in order to fish again (Fig. 4.2iii).

The duration of the first current segment of DNA-KF binary events is extended

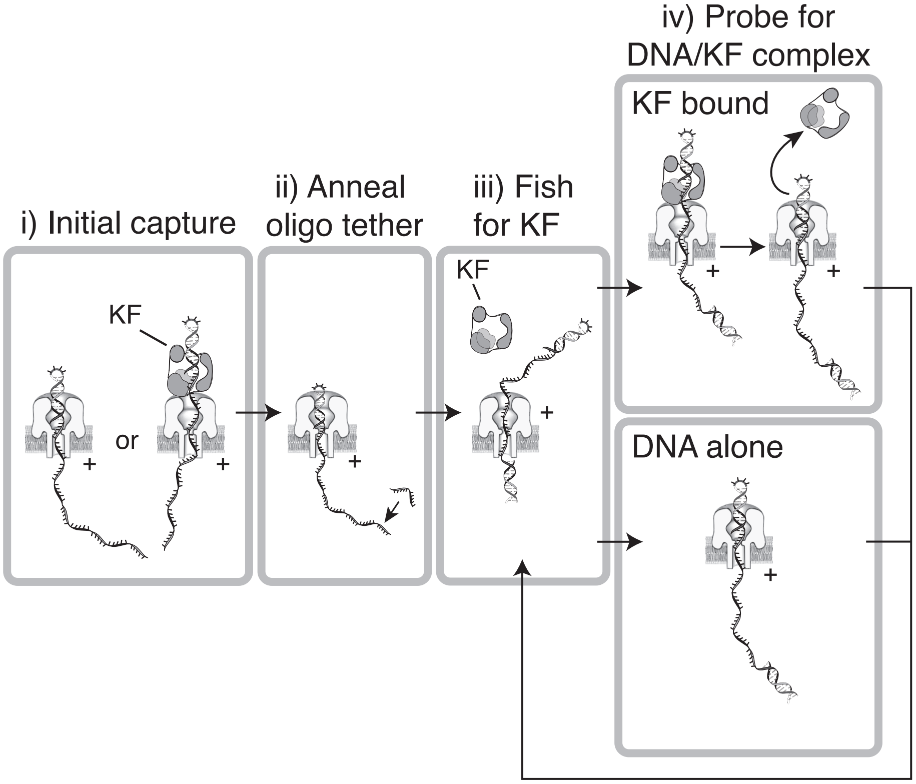


Figure 4.2: Use of the two step current signature of DNA-KF and DNA-KF-dNTP complexes to control KF association and dissociation from a DNA substrate tethered in the nanopore. Strategy for control of iterative KF association and dissociation from a nanopore-tethered DNA molecule. (i) A DNA hairpin, either KF bound or unbound, is captured under applied voltage. When the amplitude level that distinguishes capture of DNA alone or the terminal step of KF bound events is detected, (ii) the FSM lowers the voltage to hold the hairpin duplex in the pore vestibule long enough to anneal an oligonucleotide to the single-stranded end protruding into the *trans* chamber. With the DNA tethered in the pore by duplexes at both ends, a negative voltage is applied (iii) to expose the DNA to KF in the *cis* chamber ('fishing'). After a programmed time period, the voltage is reversed (iv) to draw the duplex back to the pore and diagnose whether it is KF bound based upon amplitude ('probing'). Detection of unbound DNA prompts a return to the fishing voltage. Diagnosis of the KF bound state results in continued application of the probing voltage until the terminal step is detected, which then prompts a return to fishing.

by the presence of dNTP complementary to the template base in the enzyme active site (Fig. 4.1ii,iii), consistent with assignment of this segment to the EBS [4]. Because the tethered probing experiments rely on the accuracy of this assignment, we performed an additional test. We designed a template bearing a block of 6 abasic ($1'$, $2'$-dideoxy) residues. Using the known dimensions of the nanopore [40] and the length of single-stranded DNA, the abasic residues were placed so that when the template is hybridized to a 23 mer primer, they span residues +12 to +17 relative to position $n = 0$ in the KF catalytic site. If DNA is perched atop the nanopore vestibule when it is bound to KF, captured complexes formed with this substrate should position the abasic segment in the nanopore lumen, where it is predicted to impede ionic current through the pore less than standard DNA residues. Thus the current level of the initial segment of the events should be higher than for an all DNA substrate. In contrast, when duplex DNA, absent KF, is captured in the pore vestibule, the abasic residues would be displaced into the *trans* compartment, where they would be predicted to have minimal effect on current levels. When KF and dGTP were present with the primer/template substrate bearing 6 abasic residues, the median amplitude of the EBS of capture events at 180 mV was 33.1 pA, compared to 24.8 pA for the control substrate consisting of all DNA residues (Fig. 4.3b, ii). The median amplitude of primer/template capture events, absent KF and dGTP, was 21.6 pA for the substrate containing abasic residues, and 22.3 pA for the all DNA substrate (Fig. 4.3b, i). Thus the effect of the abasic residues is exerted in the first current segment, verifying that it corresponds to the EBS of capture events.

Diagnosis of the state of complexes captured in the nanopore requires characterization of their current signatures at an appropriate probing voltage. This voltage

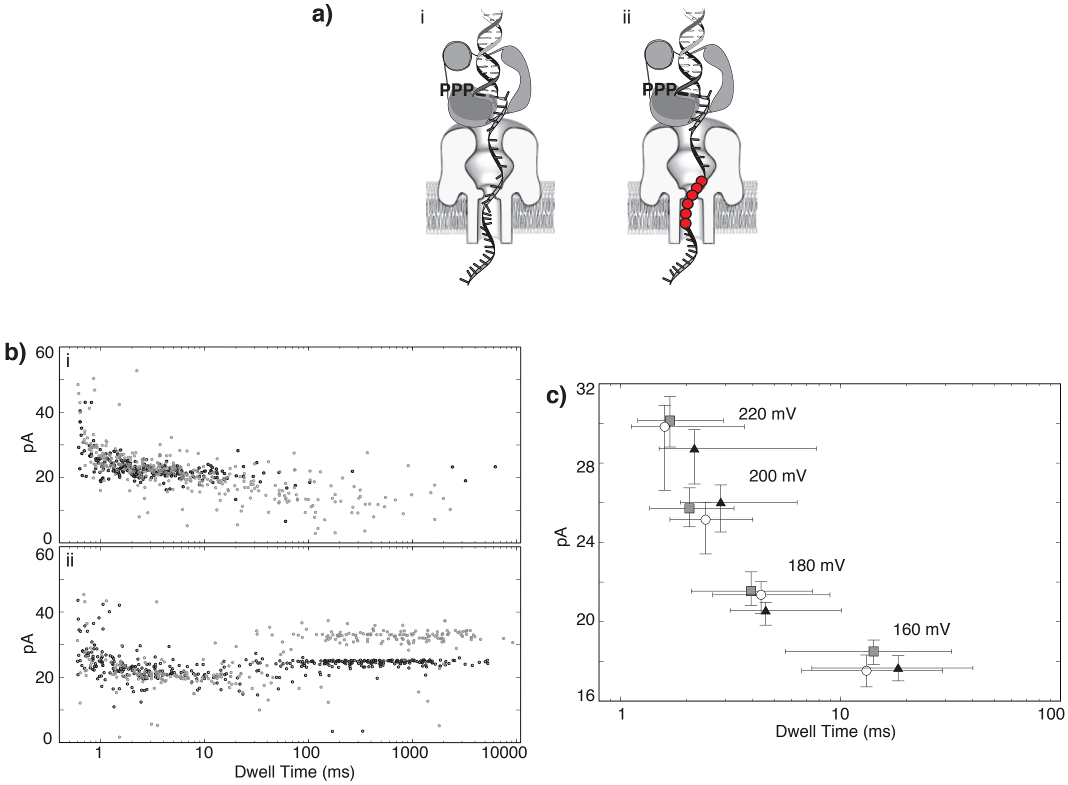


Figure 4.3: Characterization of molecular events that cause the two step current signature of DNA-KF complexes in the nanopore. a) Diagram of nanopore capture of a DNA-KF-dGTP ternary complex with (i) the all DNA template, and (ii) the 6 abasic residue-containing template. Abasic residues are in template positions +12 to +17, and are shown as red circles. b) Superimposed plots showing amplitude vs. dwell time for capture events for DNA substrates with a 6 abasic residue insert between template positions +12 to +17 (black dots) or an all DNA template (grey dots). In (i) 1 μM DNA is present but KF and dGTP are absent; in (ii) 1 μM DNA ,1 μM KF and 200 μM dGTP are present. Offline data analysis described in Methods was used to extract the EBS and terminal steps from each event in a(ii). This experiment was conducted at 180 mV applied voltage. c) Median amplitude vs. median dwell time plot for untethered DNA alone (grey squares), or the terminal current steps of binary (white circles) or ternary complexes (black triangles) captured from the bulk phase in the *cis* chamber at the indicated constant voltage levels (error bars are defined by first and third quartiles). Each data point was obtained from analysis of 181 to 552 events.

must yield sufficient ionic current for reliable discrimination between KF bound and unbound states. It must also support dwell times long enough to allow subsequent diagnosis by the FSM. We compared current patterns for DNA alone or KF bound events with a 23 base-pair all DNA hairpin substrate at 220, 200, 180, and 160 mV to identify an optimal voltage for real-time detection and control. Consistent with previous results [4], at each voltage examined, values for the amplitude and duration of the terminal step scaled with values for DNA alone (Fig. 4.3c). At 160 mV, the median amplitude and dwell time for DNA alone events were 18.5 pA and 14.7 ms, respectively, while corresponding values for the terminal step of DNA-KF complexes were 17.7 pA and 13.1 ms. The duration of unbound DNA events or terminal steps at 160 mV exceeds the ∼5 ms settling time for the capacitive transient superimposed on the current signal immediately following a voltage change, allowing reliable FSM reaction to this state prior to translocation. Using online filtering of the current signal to mitigate noise, the ∼4 pA difference between the amplitude of the enzyme bound state (21.5 pA) and the terminal current step is sufficient for accurate event diagnosis.

We conducted fishing experiments using this 23 base-pair all DNA hairpin, which bears a 50 nucleotide single-stranded overhang. Fig. 4.4 shows an experiment in which a single DNA substrate molecule was tethered in the nanopore and ∼500 fishing and probing cycles were achieved in the presence of KF and dGTP. Current traces show the repeated fishing/probing events (Fig. 4.4a, iii), and an expanded view of a single ternary complex capture event (Fig. 4.4b) with the corresponding levels of applied voltage shown below the expanded current trace. Starting from the open channel current (Fig. 4.4a, i), a DNA hairpin molecule is captured at 160 mV and detected, and the FSM logic lowers the voltage to 50 mV to hold the DNA for 10 seconds (Fig. 4.4a, ii). This period is sufficient to anneal an oligonucleotide complementary to the last 20 nucleotides at the 5′ end of the substrate without promoting translocation. Once the DNA strand is tethered in the nanopore by duplex regions at both ends, the FSM applies -20 mV for a programmed fishing time interval (250 ms in the experiment shown), to expose the tethered DNA to the bulk phase in the *cis* chamber containing DNA, KF, and dGTP. After a voltage reversal to 160 mV, the state of the captured DNA molecule is identified as either enzyme bound or unbound by the current level. If the DNA is unbound, the -20 mV fishing voltage is reapplied. If the DNA is KF bound (Fig. 4.4b), the probing voltage is maintained until detection of the terminal step that reports KF dissociation, after which the -20 mV fishing voltage is reapplied. Eventual translocation of the DNA, detected as a return to the open channel current (Fig. 4.4a, iv), prompts the FSM to await detection of another capture event at 160 mV and then return to the oligonucleotide annealing step (Fig. 4.4a, ii) to begin the process again.

Our next goal was to use this electronic control capability to probe and control the assembly of KF complexes by varying the duration of the fishing interval (Fig. 4.5; Table 4.1). We first examined events for a long fishing interval (500 ms). When only the DNA hairpin (1 μM) was present in the nanopore chamber, a population of probing events with a median amplitude of 16.8 pA and median dwell time of 2.1 ms was detected (Fig. 4.5a). The majority of the events in the presence of DNA alone (98%) had dwell times below 5.61 ms (Table 4.1, line 1). Upon addition of 2 μM KF, a new population of events emerged at $\sim$21 pA. Under these conditions, the dwell times for $\sim$38% of the

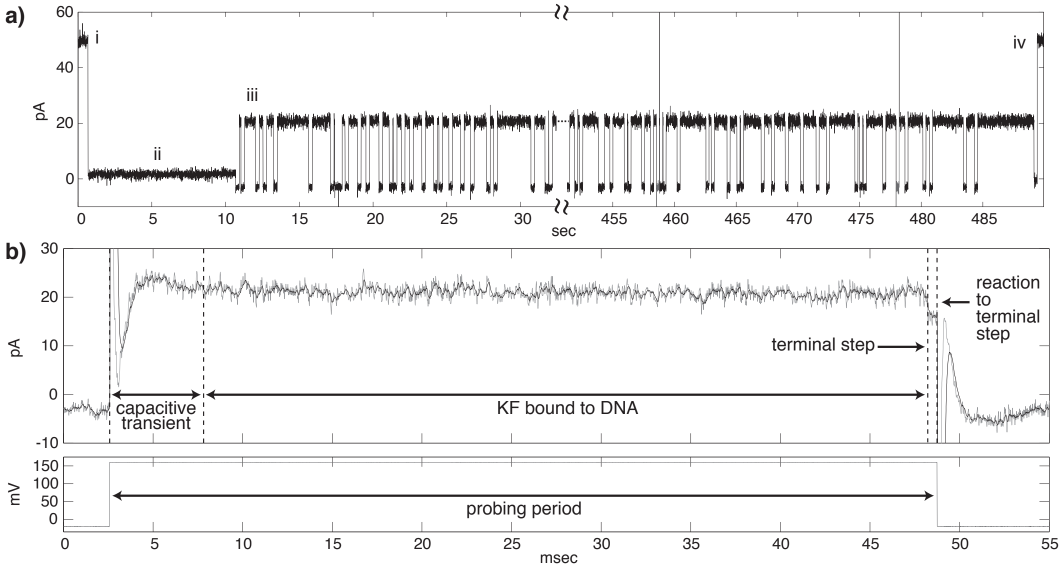


Figure 4.4: Fishing for KF with DNA tethered in the nanopore. a) Representative current trace for (i) DNA capture, (ii) DNA tethering via annealing of *trans* side oligonucleotide, (iii) ∼500 cycles of fishing and probing and (iv) DNA translocation and return to open channel current. In this experiment the fishing interval was 250 ms, with 1 μM DNA, 2 μM KF, and 400 μM dGTP in the *cis* chamber. b) Expanded current trace and corresponding applied voltage levels (below) during a single fishing and probing cycle. The trace shows the ∼5 ms capacitive transient that follows the change from the -20 mV fishing voltage to the 160 mV probing voltage. The filtered signal (black trace) mitigates noise present in the raw signal (grey trace) to avoid false detection of terminal steps. This event was diagnosed as enzyme bound since its amplitude was above [14.75, 18.75] pA, the range employed to diagnose DNA alone events and terminal steps. The 160 mV probing voltage was maintained until detection of the drop in current to within [14.75, 18.75] pA, followed by an additional 0.5 ms to ensure accurate diagnosis. A voltage reversal to -20mV to fish again is then applied.

total population were longer than 5.61 ms (Fig. 4.5b; Table 4.1, line 2), indicating that at least 38% of events are attributable to binary complexes. Addition of 400 μM dGTP yielded an additional population of events, with ∼75% of the total population lasting longer than 75.22 ms (Fig. 4.5c; Table 4.1, line 3). Therefore, at least 75% of the total events were characteristic of ternary complexes.

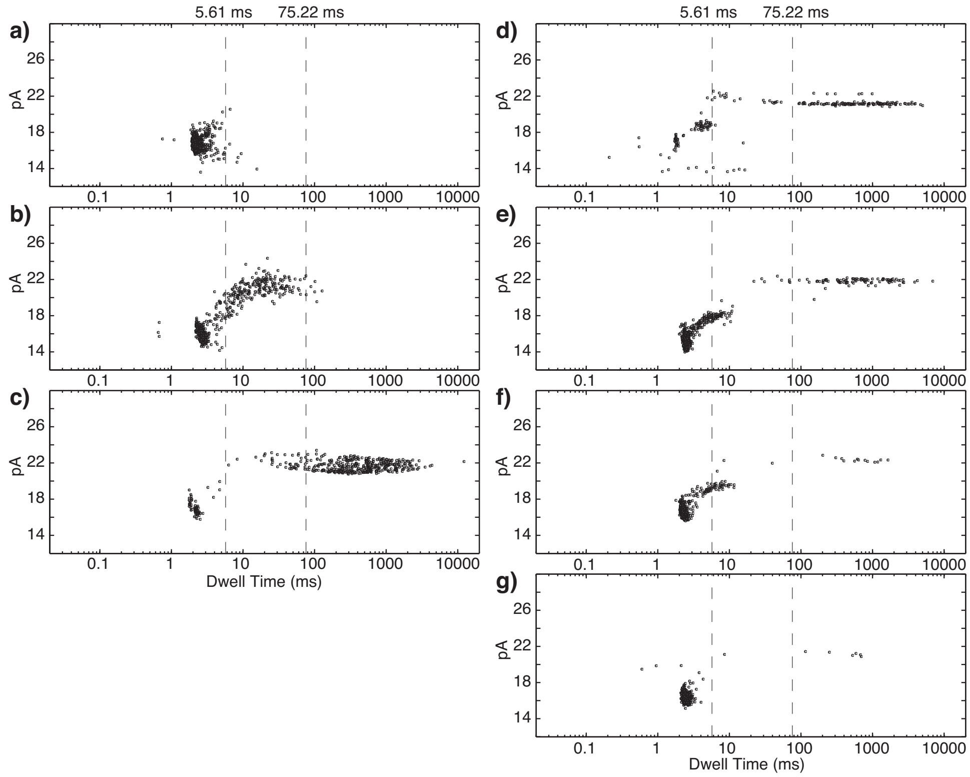


Figure 4.5: Control of complex assembly by varying fishing duration. Plots of amplitude vs. dwell time for events detected in the probing step after a 500 ms fishing interval with a) DNA alone (1 μM), b) DNA and KF (2 μM), or c) DNA, KF and dGTP (400 μM) present in the bulk phase in the nanopore *cis* chamber. With DNA, KF, and dGTP present, the fishing time interval was reduced to d) 50 ms, e) 10 ms, f) 7.5 ms, or g) 5 ms. The two dashed vertical lines through the plots indicate the upper dwell time limits for DNA alone (5.61 ms) and KF-DNA binary complexes (75.22 ms), determined as described in Table 4.1. The upper dwell time limit for DNA alone is shorter than the median dwell time for DNA alone (14.7 ms) in Fig. 4.3c because in fishing experiments these events are truncated by a voltage reversal (Fig. 4.2).

We used these three data sets determined at a 500 ms fishing interval to designate two dwell time boundaries (Table 4.1). The first boundary (5.61 ms) assigns an upper dwell time limit for events attributable to capture of unbound DNA (Fig. 4.5a). The second boundary (75.22 ms) assigns an upper dwell time limit for binary complex events (Fig. 4.5b). Thus, events lasting longer than 5.61 ms were attributed to KF

Table 4.1: Percentage of total fishing events in specified dwell time ranges

Components in chamber[a]	**Fishing time, ms**	**Total events**	**Events < 5.61 ms**[b]	**Events ≥ 5.61, < 75.22 ms**[c]	**Events ≥ 75.22 ms**
DNA	500	609	98.01%	1.99%	0%
DNA, KF	500	853	61.14%	38.04%	0.82%
DNA, KF, dGTP	500	536	14.99%	9.51%	75.5%
DNA, KF, dGTP	100	537	14.63%	9.01%	76.36%
DNA, KF, dGTP	50	269	39.42%	11.3%	49.28%
DNA, KF, dGTP	10	1052	82.36%	7.15%	10.49%
DNA, KF, dGTP	7.5	670	91.55%	5.54%	2.92%
DNA, KF, dGTP	5	668	97.96%	0.73%	1.31%

[a] The concentration of each component in this experiment, when present as indicated, was: 1 μM DNA, 2 μM KF, and 400 μM dGTP.

[b] This upper dwell time limit for unbound DNA events was calculated as the time below which 98% of events occurred when the DNA hairpin was in the *cis* chamber absent KF and dGTP (Fig. 4.5a).

[c] An upper dwell time limit (75.22 ms) for binary complex events was defined as the time below which 98% of the events that exceed the first boundary of 5.61 ms occur, when both DNA and KF were present (Fig. 4.5b).

bound DNA, and events lasting longer than 75.22 ms, when DNA, KF, and dGTP were present in the chamber (Fig. 4.5c), were considered to be due to ternary complexes.

We then examined whether shifts in the event populations occurred as the fishing interval was shortened, in the presence of all components required for ternary complex formation (1 μM DNA, 2 μM KF, and 400 μM dGTP). At this high dGTP concentration we expect that for all fishing intervals, almost every time that a DNA-KF complex is drawn back to the pore, the dwell time measured during the probing phase will be characteristic of a ternary complex. This is because regardless of whether the complex is binary or ternary at the moment of capture from the bulk phase, dGTP can bind, unbind, and rebind to the complex during the probing period, extending its lifetime atop the pore [24]. Thus, in the presence of 400 μM dGTP, as fishing time is changed, variation in the percentage of EBS events is dictated primarily by the rate of KF binding to the tethered DNA.

When the fishing time was reduced from 500 ms (Fig. 4.5c) to 100 ms, the percentage of total events within the designated dwell time boundaries was virtually indistinguishable (Table 4.1, lines 3 and 4), suggesting that equilibration with the the bulk phase occurs between these time intervals. When the fishing time was reduced to 50 ms, the number of events longer than 75.22 ms decreased to $\sim$50% of the total population, with a concomitant increase in putative DNA events (<5.61 ms) to $\sim$40% (Fig. 4.5d; Table 4.1, line 5). No significant increase in events with dwell times between 5 ms and 75 ms was observed. Further reductions of the fishing time to 10 ms and 7.5 ms shifted the populations further (Fig. 4.5e and f; Table 4.1, lines 6 and 7). At 5 ms fishing time, $\sim$98% of events were of an amplitude ($\sim$16 pA) and duration characteristic of unbound DNA (Fig. 4.5g; Table 4.1, line 8).

Fig. 4.6 shows the effect of dGTP on the percentage of EBS events as a function of fishing time. This representation of the data does not distinguish among different EBS events based upon dwell time. It is therefore sensitive to the effect of dGTP in the bulk phase on the probability of drawing a KF-bound complex back to the pore, but is

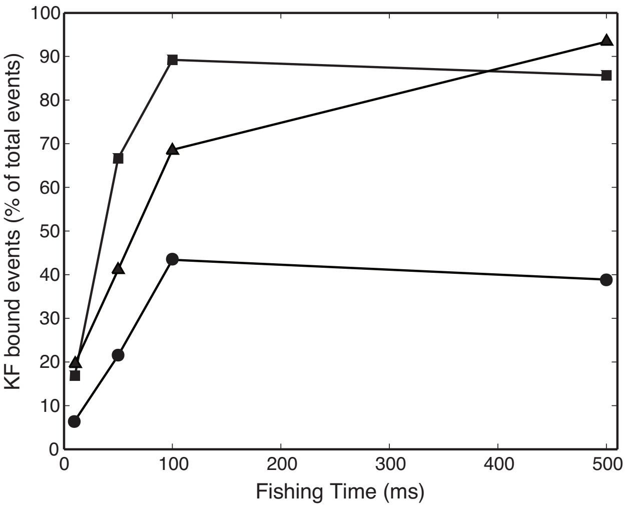


Figure 4.6: Effect of dGTP on DNA-KF complex assembly. Plot of the percentage of total events diagnosed as KF-bound when the fishing interval was varied in the presence of 1 μM DNA, 2 μM KF, either absent dGTP (circles), or in the presence of 50 μM dGTP (triangles), or 400 μM dGTP (squares). The percentage of EBS events was determined as described in Methods. Percentage values for each plotted point were determined from at least 255 and up to 7828 events (Table 4.2).

independent of the effect of dGTP binding and rebinding to KF-bound complexes while they reside atop the pore during the probing period, which causes a dGTP-concentration dependent increase in dwell time [24].

With a 500 ms fishing interval, 39% of events detected absent dGTP were enzyme bound (Fig. 4.6, Table 4.2), compared to 93% and 86% in the presence of either 50 or 400 μM dGTP, respectively. This reflects the increased affinity of KF for DNA conferred by the presence of dNTP [27]. At this long fishing interval, EBS events converge to ~90% for the dGTP concentrations examined (Fig. 4.6, Table 4.2). This behavior is predicted by the value of Kd ~5 μM for bulk phase dNTP binding to DNA-KF complexes [24, 11]. Thus at dGTP concentrations above ~10 μM, the ability of

Table 4.2: Percentage of total fishing events diagnosed as enzyme bound

dGTP concentration, μM	**Fishing time, ms**	**Total events**	**EBS events**[a]	**Percentage of events with EBS**
0	10	7828	511	6.5%
0	50	6781	1454	21.4%
0	100	997	433	43.4%
0	500	854	332	38.9%
50	10	1635	320	19.6%
50	50	749	308	41.4%
50	100	653	448	68.6%
50	500	1031	963	93.4 %
400	10	1050	177	16.9%
400	50	255	170	66.7%
400	100	520	464	89.2%
400	500	531	455	85.7 %

[a] Number of events with dwell times longer than the upper dwell time limit for unbound DNA. The upper limit is the time below which 98% of events occurred when the DNA hairpin was in the *cis* chamber absent KF and dGTP.

dNTP to increase the affinity of KF for DNA at equilibrium is saturated.

At fishing intervals shorter than 500 ms, decreases in the percentage of EBS events are observed. The extent of these decreases varied with dGTP concentration. Thus the slopes of the curves for the percentage of KF-bound events as a function of fishing time increased as dGTP concentration was increased from 0 to 50 to 400 μM (Fig. 4.6). This suggests that the rate of approach to equilibrium is dependent upon dGTP concentration. That is, for fishing intervals within the approach to equilibrium, increasing dGTP concentration increased the probability of the DNA molecule being KF-bound at the moment it was drawn back to the pore.

Experiments in which the fishing interval is varied thus probe complex assembly on a time scale that superimposes with the rate of KF binding to DNA (Fig. 4.5), and with the time scale of dGTP binding to binary complexes (Fig. 4.6). Each fishing and probing event uses an individual DNA molecule, known to be unbound by enzyme at the moment that it is exposed to KF and dNTP in the bulk phase. This yields the potential to measure pre-steady state complex assembly at the single molecule level. This approach is statistically robust, as hundreds to thousands of events can be analyzed in each experiment. Finer temporal resolution using additional fishing time intervals, in combination with mathematical modeling [24], should permit the determination of binding rate constants.

4.2 Methods

4.2.1 Materials

DNA oligonucleotides were synthesized by Midland Certified Reagent Company or the Stanford University Protein and Nucleic Acid Facility and purified by denaturing PAGE. Sequences of the oligonucleotides used in this study are shown below. The sequences of the the 6 abasic residue-containing 79-mer template and the all DNA control are identical except for the region replaced by abasic residues, each indicated by an X. The $3'$ terminal residue of the 23 base pair hairpin (bphp) and 23-mer primer was 2′-3′ dideoxycytidine (ddC), enabling ternary complex formation without catalytic turnover. The $n = 0$ template base for all substrates was C; thus the complementary incoming nucleotide was dGTP.

23 bphp:

5′-CTCACCTATCCTTCCACTCATTCCTTACCATTCATTCACATCTCACTATCG

CATTCTCATGCAGGTCGTAGCCTTTTGGCTACGACCTGCATGAGAATGddC 3′

20-mer tethering oligonucleotide:

5′-TGAGTGGAAGGATAGGTGAG-3′

23-mer primer:

5′-GGCTACGACCTGCATGAGAATGddC- 3′

All DNA control 79-mer template:

5′-CTCACCTATCCTTCCACTCATTCCAATTAATTACCATTCATTCAGATCTCA

CTATCGCATTCTCATGCAGGTCGTAGCC-3′

6 abasic residue-containing 79-mer template

5′-CTCACCTATCCTTCCACTCATTCCAATTAATTACCATTXXXXXXGATCTCA

CTATCGCATTCTCATGCAGGTCGTAGCC-3′

20 bphp used in detection and ejection experiments (Chapter 3):

5′-CTCACCTATCCTTCCACTCATTTTCCTTAACCATCGCATTCTCATGCAGGT

AGCCTTTTGGCTACCTGCATGAGAATGddC 3′

Prior to use, the 23 bp hairpin was denatured at 95°C for 2.5 min and quickly annealed in ice water to prevent intermolecular hybridization. The 23-mer primer was annealed to each 79-mer template as previously described [4]. Klenow fragment (exo-, 100,000 U/ml, 20,000 U/mg) was obtained from New England Biolabs.

4.2.2 General nanopore methods

Experiments were conducted at 23C in 10 mM HEPES/KOH, pH 8.00 0.05, 0.3 M KCl, 5 mM $MgCl_2$, which are conditions shown to support KF catalytic function [4]. In all experiments 1 μM DNA substrate was present in the *cis* chamber. KF and dGTP were used, when indicated, at 2 μM and 400 μM, respectively, unless otherwise indicated in figure legends. Single α-HL channels were formed as described [43]. A patch-clamp amplifier (AxoPatch 200B, Molecular Devices, Sunnyvale, CA) was used to apply trans-membrane voltage and measure ionic current, with the 4-pole Bessel filter set at 5 kHz bandwidth. A digitizer (Digidata 1440A, Molecular Devices) stored data sampled at 200 kHz.

The lipid bilayer formed across a ~20 μm teflon aperture in a KCl solution allows α-HL channel formation. A patch-clamp amplifier applies a trans-membrane voltage that produces an ionic current through the pore. In absence of sample in the system and a single α-HL channel inserted into the bilayer, a steady and predicable ionic current can be seen for a given applied voltage. On either side of the bilayer, Ag-AgCl electrodes convert the electric current from the amplifier to an ionic current through

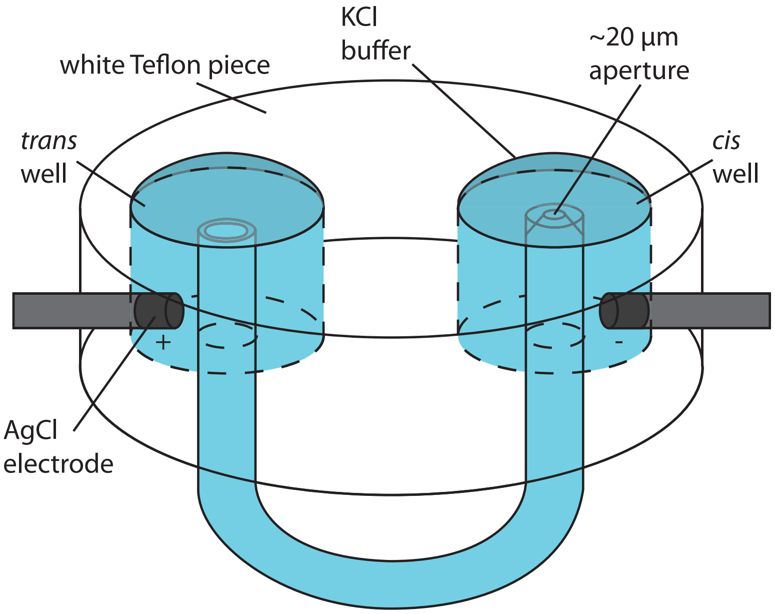


Figure 4.7: Nanopore set-up with white Teflon piece.

the pore, Figure 4.7.

4.2.3 Active voltage control experiments

Active voltage control logic was implemented as a finite state machine (FSM), a logic construct where program execution is broken into a series of individual states. Measurements determine which state the logic is in, and therefore the sequence of executed commands over time [18]. Changes in the FSM control logic are made as necessary, then re-downloaded to run on the FPGA. This achieves a balance between speed and flexibility, by enabling the system to react to events on the order of a microsecond, while also allowing for the control logic to be reconfigured as necessary between experiments. The FSM was programmed with LabVIEW software (Version 8, National Instruments, Austin, TX) and implemented on an FPGA system (PCI-7831R, National Instruments), a reconfigurable hardware platform [41]. The FPGA was connected to the Axopatch 200B and permitted trans-membrane voltage control and ionic current measurements at 5.3 μs updates. To improve the signal-to-noise ratio, the ionic current signal was filtered on the FPGA using a single-pole low pass filter with 1.58 kHz cutoff frequency and a 10%-90% rise time of $\sim$0.2 ms.

For the tethered fishing experiments, the FSM initialized tethering by detecting capture of a DNA molecule at 160 mV, and reducing the voltage to 50 mV for 10 s. With the 20-mer tethering oligonucleotide present in the *trans* chamber at 4 μM, this period yielded 70% oligonucleotide annealing efficiency. The FSM then triggered the -20 mV fishing voltage, which was sustained for the time periods specified in the figures and table. During each 160 mV probing period, the FSM logic diagnosed the DNA as unbound if the low-pass filtered current remained within amplitude thresholds for 0.5 ms. The thresholds were calibrated at the beginning of each experiment, by monitoring bound and unbound tethered DNA probing event amplitudes, and was not varied during a given experiment at a set voltage and KF and dGTP concentrations. Experiment-to-experiment variations in the thresholds were due largely to evaporation-induced drift in the open channel current by 1-3 pA, over the course of several experiments. Example ranges used are [14.25, 18.74] pA (for 0 μM and 50 μM dGTP at all reported fishing times) and [15.5, 19.5] 2 pA (for 400 μM dGTP at 500 ms fishing time). These threshold levels for unbound DNA diagnosis were consistent with the observed amplitude separation between enzyme bound (21.5 pA) and unbound (18.5 pA) DNA in constant 160 mV experiments. If the low-pass filtered current remained outside the amplitude thresholds, the DNA was diagnosed as enzyme bound. Upon diagnosis of unbound DNA, the FSM re-triggered and sustained the fishing voltage period.

In nanopore systems using patch-clamp technology, a step change in voltage induces a capacitive transient superimposed on the ionic current measurement. The transient masks information in the measured current for a time proportional to the magnitude of the net voltage change. The change from the -20 mV fishing voltage to the +160 mV probing voltage excited a capacitive transient that lasted $\sim$5 ms. Since this transient time exceeds the filter settling time (0.2 ms), initial diagnosis of the tethered DNA as enzyme bound or unbound was dictated by the transient settling time. In experiments with untethered DNA at a constant voltage of 160 mV, 50% of binary complex bound states lasted longer than 5.8 ms. Therefore up to 50% of binary complexes captured may be diagnosed as unbound DNA events by the FSM control logic because the EBS was masked by the transient. At 50 μM and 400 μM dGTP the EBS duration is extended, significantly reducing the likelihood that the transient would mask an EBS and subsequent terminal step. When any EBS lasted longer than the transient settling time, KF dissociation was diagnosed by the FSM within $\sim$0.7 ms of when the terminal step occurred, dictated by the sum of the filter rise time (0.2 ms) and the control logic diagnosis time (0.5 ms).

4.2.4 Offline Data Processing

Recorded current blockades reported in the figures were tabulated offline using MATLAB (2007a, The MathWorks, Natick, MA) with software developed in our lab. In experiments with untethered DNA, an event was identified when the open channel current (∼53 pA at 160 mV) dropped by at least 8 pA for at least 0.2 ms. In the experiments with tethered DNA, the dwell time of each event was defined as the period during which the FSM applied the probing voltage.

To identify and quantify terminal steps in constant voltage experiments with untethered DNA (Fig. 3), a baseline amplitude was calculated as the mean of the first 15% of the event amplitude for all events longer than 1 ms. Events that ended with a segment of at least 3.5 pA below the baseline amplitude were identified as having a terminal step, and the dwell time and mean amplitude of each end segment were used to quantify the terminal step. Offline analysis was used to estimate the performance of the FSM control logic in tethered DNA experiments. On average, 96% of terminal steps were detected and reacted to within 2 ms.

In Fig. 4.5, the mean amplitude of each probing event was computed from the time that the transient crossed 12 pA (1-2 ms after the start of probing) to the end of the probing period. To compute the percentage of EBS events in an experiment (Fig. 4.6, Table 4.2), a probing event with a dwell time longer than the 98%-cutoff time for unbound DNA was diagnosed as enzyme bound. The 0 μM and 400 μM dGTP experiments were performed on one nanopore station (with a corresponding unbound

DNA cutoff time of 5.61 ms (Fig. 4.5a)), and the 50 μM dGTP experiments were performed on another nanopore station (with a corresponding unbound DNA cutoff time of 9.22 ms). The transient pattern dictated the unbound DNA diagnosis time; thus, differences in the transient duration on the two stations determined the different 98%-cutoff times characteristic of unbound DNA.

Chapter 5

Future work and conclusions

5.1 Future research

5.1.1 System hardware design

The FSM system presented here can be expanded in several ways. Presently, all the computational components and data capture hardware reside in separate devices. Communications between all these devices makes the system susceptible to noise. The integration of all the components into one system would reduce the noise caused by all the long interconnects of the system.

Consolidation of the electronics would also remove the need to purchase the proprietary devices currently in use, primarily the patch-clamp amplifier and the digitizer. This would reduce the overall cost of a nanopore setup. Additionally, using a third-party amplifier and digitizer typically necessitates the use of expensive software to read the closed data format employed during data collection. Several vendors offer an open file format that would facilitate the development of software by our lab for more specialized data analysis.

Full control of the patch-clamp amplifier, including capacitive compensation controls, would allow for complete automation of the electrical portion of the nanopore system. With the FPGA controlling the amplifier, none of the human interface electronics, knobs and indicators, are necessary. A small, single purpose amplifier can be designed for nanopore voltage control and ionic current measurement via FPGA interfacing in mind from the beginning.

The 16-bit analog-to-digital converters on the FPGA are more than capable of logging the data from the nanopore in addition to their current detecting task. After the data is compared against the FSM's detection levels, it can then in turn be transmitted to the host over DMA with little overhead. Design of a single device to perform all the control, voltage actuation and current recording would be the most beneficial situation. Such a device would reduce system noise as well as allow for the design to be expanded to more than single-channel control. Analysis of the current amplifier design as well as design of an infrastructure for communication with a PC for data viewing and system-level control are necessary prerequisites for developing such a device.

The capacitive transient remains the major bottleneck in the reaction time of the system. As more complex control schemes are devised and applied to the nanopore system, they will require increased voltage control that creates a capacitive transient in the current signal with each voltage change. Increasing the number of transients decreases the amount of usable signal during an experiment. Waiting even 5 ms for an accurate current signal is often too long when working at biological time scales. The aforementioned electronic control of the compensation circuitry in the patch-clamp amplifier would allow automation of the capacitive compensation for measurable and deterministic operation calibrated specially for each experiment. This systematic removal of the transient through current injection would allow reaction to events almost immediately following a voltage change.

5.1.2 Data analysis and classification

Real-time event classification during an experiment would save time spent during offline analysis and ensure that valuable pore time is efficiently used. Before adding additional reagents, an irreversible and often costly step, the number and rough classification of molecules that have translocated thus far can be tabulated to determine the appropriate time for proceeding to the next stage of the experiment. Parameters input before the experiment would determine how to classify particular current signatures. A running tally of each type of event can indicate whether enough data have been collected to be statistically significant, allowing the experiment to proceed only after enough data of one type has been collected.

Further modeling of the pore and its dynamics with and without captured molecules needs to be a necessary part of future research. Capture rates of DNA under various voltages and concentrations would enable control of the DNA arrival rate. Furthermore, similar characterization of the pore during capture of various compositions of DNA might facilitate identification of certain sequences. Nanopore-based control is far from reaching a stopping point. In fact, this field is just beginning to develop as more laboratories begin to conduct nanopore research with improved measurement and control circuitry and logic.

5.2 Summary of main results

The research described in this dissertation demonstrates the capture of single DNA hairpins and DNA/enzyme complexes in a biological nanopore and the subsequent diagnosis and reaction using a FSM implemented on a FPGA. Integration of such hardware technology with the nanopore system provides means for automatic voltage actuation based on the ionic current flowing through the pore.

DNA or DNA/enzyme complex captured in the pore was detected and reacted to in real time using active voltage control. The FSM is capable of extending the dwell time of such translocation events by reducing the voltage. This produces more signal and can aid offline analysis. Additionally, ejection of DNA hairpins from the pore after detection but prior to unzipping the hairpin was achieved. Rapid DNA hairpin detection (< 2 ms) used a low-pass filtered amplitude that required the current to remain within a preset amplitude range for a pre-programmed period of time. This method of detection requires more time than a single raw amplitude threshold test, as used in previous nanopore control experiments, however it is more robust to false detections and offers specific identification of captured molecules.

Experimental results have shown repeated capture and dissociation of KF from a single DNA hairpin is not only possible but also creates the possibility for new enzyme-based experimental protocols. The method was tuned to differentiate DNA/enzyme complex from DNA-alone in real time. Ultimately, the demonstrated control authority was refined for exposure of a single captured DNA for repeated binding/unbinding or KF. This demonstrated the nanopore's ability to detect and monitor multiple macromolecular states and takes novel steps toward enabling capabilities for nanopore-based sequencing and characterization of enzyme-DNA dynamics through enzyme fishing.

Appendices

Appendix A

Designing and Running Experiments Using the Nanopore FPGA Control Panel

This appendix aims to provide the reader with the information necessary to implement finite state machine (FSM) designs using `FSMCreator` and run them on the FPGA using the Nanopore FPGA Control Panel under LabVIEW. The Nanopore FPGA Control Panel controls the loading and execution of FSM designs on the FPGA.

A.1 Introduction

A.1.1 Finite state machines

An FSM models system behavior using a discrete number of states. Each state represents a specific configuration of the system being modeled and execution of these states is based on the previous state and some inputs. The type of FSM, Moore or Mealy , determines how the state outputs are computed. With a Moore machine, the output is dependent only on the current state, where with a Mealy machine, the output depends on both the current input and state. Here, the FSM runs on the FPGA as a Moore machine: the output only depends on the current state.

A.1.2 Necessary files

Several files are necessary to run of the Nanopore FPGA Control Panel and are listed in Table A.1. These files are usually located in the `nanopore_state_machine` directory on the `C:` drive.

After all the necessary files are in place, FSM design can begin. FSM creation and implementation can be broken down into five main steps:

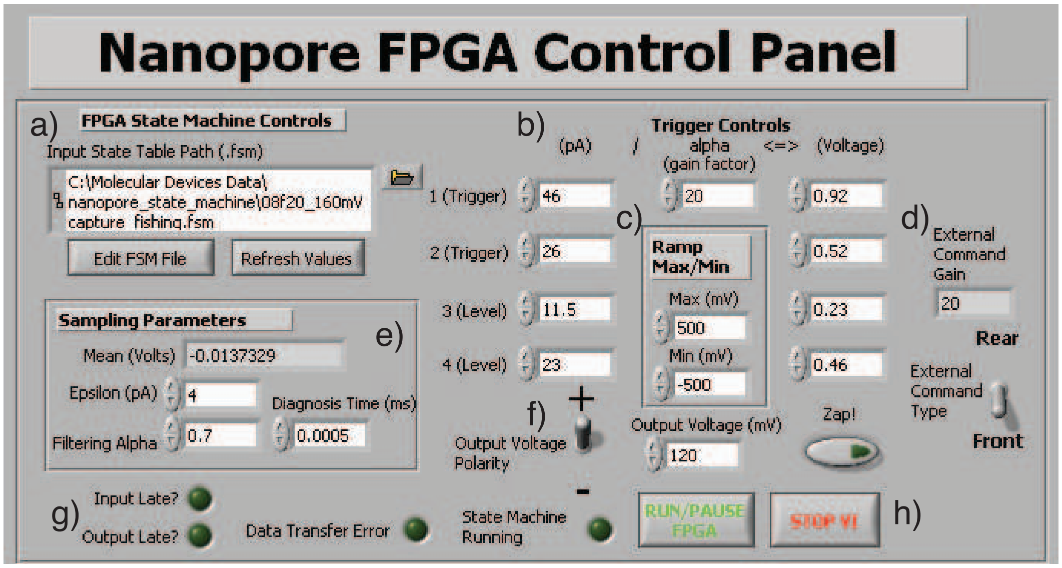
1. Design an FSM for the desired experiment
2. Enter the design into `FMSCreator` to produce `.fsm` and `.fpga` files
3. Select the FSM file (`.fsm`) in the Nanopore FPGA Control Panel
4. Confirm the FPGA parameters in the Nanopore FPGA Control Panel
5. Start the FSM

File Name	**Function**
`state_table.fsm`	FSM file that codes state machine behavior
`state_table.fpga`	File with converted FSM for download to FPGA
`state_machine.lvproj`	LabVIEW project file
`noahstatemach.vi`	LabVIEW VI runs FPGA control panel
`input_fpga.vi`	LabVIEW VI for FPGA FSM
`state_tbl_init.vi`	LabVIEW subVI downloads FSM file to FPGA
`StateCurrentFIFOwrite.vi`	LabVIEW subVI writes current signal to host via the DMA FIFO
`search_array_count.vi`	LabVIEW subVI for counting array elements
`roundToZeroIfLessThanZero.vi`	LabVIEW subVI rounds less than zero values to zero
`rampOutputController.vi`	LabVIEW subVI controls the FPGA ramp output
`nextStateLookup.vi`	LabVIEW subVI returns the next state based on the current state and inputs
`lookupFSMoutputs.vi`	LabVIEW subVI looks up the currents state's output value
`hysteresisTrigger.vi`	LabVIEW subVI determines the state of the FSM's trigger inputs
`expLowPass.vi`	LabVIEW subVI low-pass filters the input current signal
`detectionLevelCompare.vi`	LabVIEW subVI determines the state of the current level detect inputs
`computeLookupAddr.vi`	LabVIEW subVI computes the lookup address for the next state given the current state's inputs
`computeDelayBit.vi`	LabVIEW subVI computes the time spent in the current state for delay timer triggering
`FSMCreator.jar`	Java application that creates `.fsm` and `.fpga` files.

Table A.1: Necessary file to run Nanopore FPGA Control Panel under LabVIEW.

A.2 The Nanopore FPGA Control Panel GUI window

Control of the FSM/FPGA is performed from the FPGA control panel window. Open LabVIEW 8.0 (note that compatibility for versions higher than LabVIEW 8.0 has not been achieved at the time of this writing) and load the `state_machine.lvproj` LabVIEW project. Double-click on the `noahstatemach.vi` in the project window to open the Nanopore FPGA Control Panel window, Figure A.1. Click on the *Run* arrow button in the toolbar to start the control panel. Figure A.2 shows the sequence of steps necessary to open the FPGA control panel from within LabVIEW.



Nanopore FPGA Control Panel

FPGA State Machine Controls

Input State Table Path (.fsm): C:\Molecular Devices Data\nanopore_state_machine\08f20_160mV capture_fishing.fsm

Sampling Parameters

Mean (Volts): -0.0137329

Epsilon (pA): 4

Filtering Alpha: 0.7

Diagnosis Time (ms): 0.0005

Trigger Controls

	(pA)	/ alpha (gain factor)	<=> (Voltage)
1 (Trigger)	46	20	0.92
2 (Trigger)	26		0.52
3 (Level)	11.5		0.23
4 (Level)	23		0.46

Ramp Max/Min

Max (mV): 500

Min (mV): -500

Output Voltage (mV): 120

External Command Gain: 20

External Command Type: Rear / Front

Output Voltage Polarity: + / -

Figure A.1: The main FPGA control window handles the setting of various parameters for the state machine running on the FPGA: a) `.fsm` loading/editting dialog, b) FSM parameter viewer, c) ramp max/min setting, d) external command gain switch (front/rear external command), e) filtering/sampling parameters for current signal level detect input, f) output voltage controls for use when FSM is not running, g) diagnostic indicators, h) FSM start/stop controls.

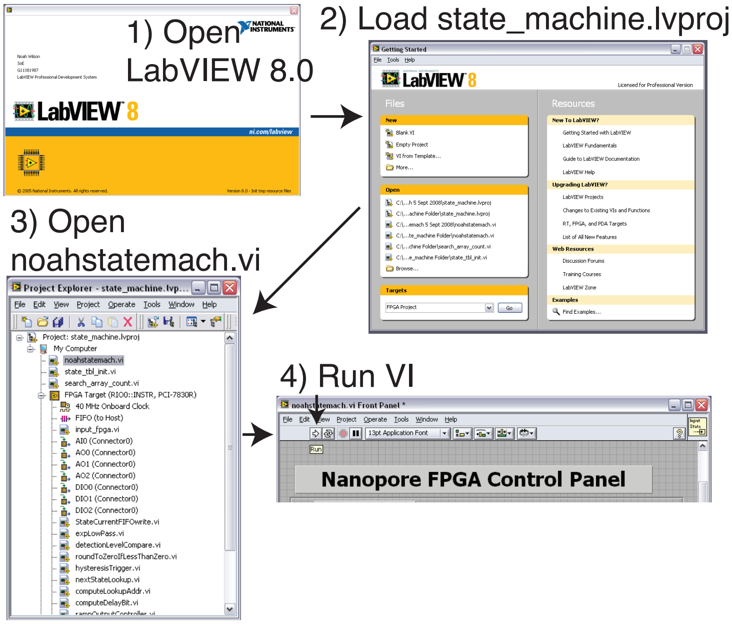


Figure A.2: Run the Nanopore FPGA control panel by opening the `state_machine.lvproj` project and running `noahstatemach.vi`.

When an FSM is not running, the FPGA control panel functions in constant voltage output mode. The polarity and magnitude of the voltage applied to the pore can be set via the *Output Voltage Polarity* and *Output Voltage (mV)* controls, respectively (Figure A.1f). Pressing the *Zap!* button will apply maximum voltage (200 mV for front external command and 1000 mV for rear external command) to the pore while the button is pressed. This is usually used for the purpose of rupturing the bilayer though 200 mV is typically not enough to rupture a bilayer.

A.2.1 Loading FSM designs

Saved FSM designs (`.fsm` file extension) are loaded into the control panel by clicking the folder icon (Figure A.1a). Clicking this button brings up a file selection dialog where the desired file can be selected. After selecting a file, the *Refresh Values* button below the FSM file selection box will update the various FSM parameters displayed in the FPGA control panel window to the values stored in the `.fms` file. Creation of these FSM files is covered in Section A.3.

A.2.2 Viewing and adjusting FSM parameters

After a FSM has been loaded into the control pannel, the FPGA control panel loads all the non-state specific parameters display fields in the window. These fields can be adjusted as necessary but should be updated in `FSMCreator` to ensure the changes made to the parameters are saved. The FPGA control panel displays the ϵ for current level detection, filtering α, diagnosis time, and the trigger and level threshold values. All the fields correspond to the FSM parameters described in more detail in Chapter 2.

In addition to displaying FSM parameters, the FPGA control panel also controls certain gain settings to correctly interface with the patch-clamp amplifier. The α or gain factor setting (Figure A.1b) must be set to the same α set on the amplifier. A difference between the value in the FPGA control panel and the amplifier causes the current trigger and level inputs to have the wrong threshold values. Similarly, the *External Command Gain* toggle switch needs to be set to 'front' or 'rear' depending on the external command input being used on the amplifier. Ramping voltages applied by the FSM can be clipped at a minimum and maximum amplitude by setting bounds in the *Ramp Max/Min* controls (Figure A.1c).

A.2.3 Executing FSM designs

To run a FSM, ensure that the desired FSM is loaded and then press the green *RUN/PAUSE FPGA* button (Figure A.1h). The FPGA control panel will then download the design to the FPGA, verify the download and begin execution of the FSM. Clicking *RUN/PAUSE FPGA* again while the FSM is running will stop the FSM and return to constant voltage output mode. If a change has been made to a FSM, the design needs to be saved in `FSMCreator` and if the FSM is running, it needs to be stopped and then started again for the new version of the FSM to download to the FPGA.

Several diagnostic lights display status information for the FSM running on the FPGA (Figure A.1g). The *Input Late?* and *Output Late?* indicators refer to the input and output state machine loops running on the FPGA. These lights should never come on unless LabVIEW is having difficulty communicating with the FPGA or the FPGA control panel is being run on a version of LabVIEW other than 8.0. The *Data Transfer Error* light indicates if there was an error sending the FSM design to the FPGA. Again, this light should never come on unless there is a problem with the communication between the host PC and the FPGA, which is typically a result of incorrectly configured hardware drivers under the Windows operating system. Finally, the *State Machine Running* indicator lights up when a FSM is running on the FPGA and is off when the FPGA control panel is operating in constant voltage mode.

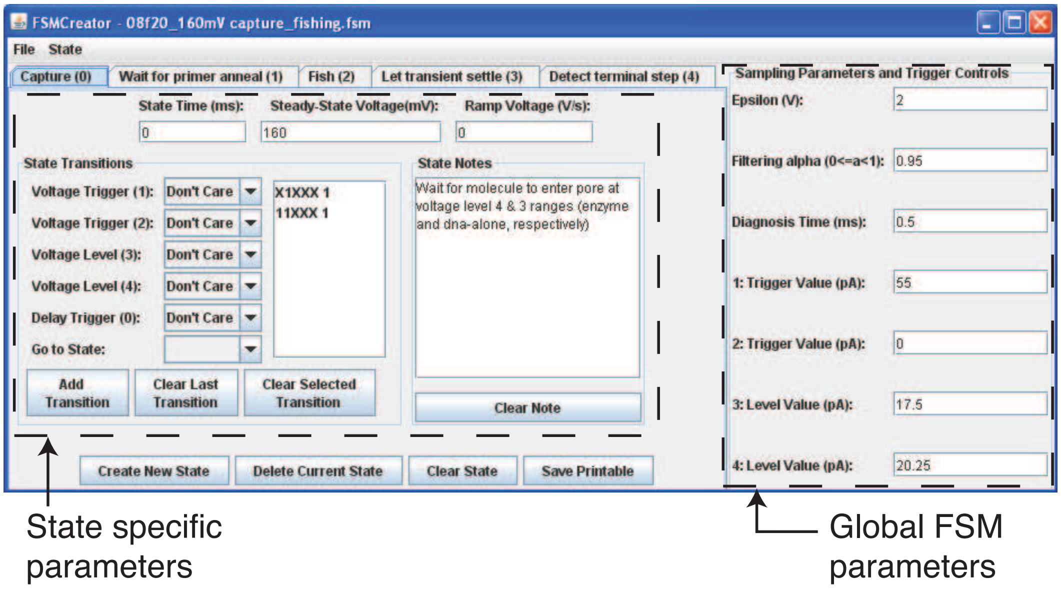
To stop the FPGA control panel (*i.e.* when quitting LabVIEW), click on the red *STOP VI* button twice. This will stop the VI and set all output voltages of the FPGA to 0 volts, allowing the user to exit with the FPGA in a known stable state.

A.3 FSM entry using FSMCreator

Global FSM parameters and parameters particular to each state are defined in a `.fsm` file using `FSMCreator`. `FSMCreator` can be opened from the Nanopore FPGA Control panel by clicking the *Edit FSM file* button. Make sure that the *External Command Gain* (Figure A.1d) is set to the appropriate setting for the nanopore setup being used. If using the front external command, set the switch to 'front', otherwise set it to 'rear.' This switch ensures that the right gain value is sent to `FSMCreator` when it is called using the *Edit FSM file* button.

The `FSMCreator` window is shown in Figure A.3. Take note that the *Edit FSM file* button will open a new instance of `FSMCreator` rather than return to the previously opened window. To return to a previous window, use the Windows taskbar.

In `FSMCreator`, each state of the FSM has its own tab where all the state transitions and parameters are entered. Creating a new state will bring up a new tab that will be labeled with the name of the state. Renaming a state is performed either from the *Rename state* menu item or from the dialog that opens when a new state is created.



State Time (ms): 0
Steady-State Voltage(mV): 160
Ramp Voltage (V/s): 0

State Transitions
Voltage Trigger (1): Don't Care
Voltage Trigger (2): Don't Care
Voltage Level (3): Don't Care
Voltage Level (4): Don't Care
Delay Trigger (0): Don't Care
Go to State:

```
X1XXX 1
11XXX 1
```

State Notes
Wait for molecule to enter pore at voltage level 4 & 3 ranges (enzyme and dna-alone, respectively)

Sampling Parameters and Trigger Controls
Epsilon (V): 2
Filtering alpha (0<=a<1): 0.95
Diagnosis Time (ms): 0.5
1: Trigger Value (pA): 55
2: Trigger Value (pA): 0
3: Level Value (pA): 17.5
4: Level Value (pA): 20.25

State specific parameters

Global FSM parameters

Figure A.3: The `FSMCreator` application handles editing and design of FSMs.

State transitions are defined inside a state tab by selecting one or more of the five inputs that trigger a particular action (*i.e.* transition to a different state). More detail on how each input operates is given in Chapter 2.

The defined values for the triggers or delay value of a state determine how the state will respond to these inputs. Previously defined state transitions will be stored in the listbox next to the input selection boxes the with the following format:

```
XXXXX N
```

where each `X` represents one of five inputs to the state machine and `N` is the state to jump to when that input combination is satisfied. Possible values for each input are `1`, `0`, or `X`, where `1` or `0` correspond to inputs of true or false, respectively, and the `X` signifies a “don't care” on the input (*i.e.* the input can be a 1 or a 0 and still be satisfied). The identity of each input is shown in Figure A.4. Double-clicking on a transition in the state transition listbox will load the input values into the pull-down boxes for easier translation.

Caution should be taken when defining state transitions to ensure that the input space of each transition does not overlap (*i.e.* two defined state transitions do not have conditions where the current value of the inputs would cause both would be true). This can lead to unexpected state changes due to contention between the two state change options. To mitigate this, multiple state transitions with fewer, more specific input constraints and should be defined that point to the same next state. In the case where several input transitions point to the sane state, the input constraints will be OR'd (only one needs to be true) by the FSM logic to determine if FSM execution will move to that state. For a given state transition, all the input constraints are AND'd together (they must all be true).

As `FSMCreator` creates the FSM files during saving, processing starts at the top and goes down the state transition list, inserting each state transition into the state transition table. If the input conditions of two or more state transitions overlap, the state transition closest to the bottom of the list will take priority since it was added last, overwriting any part of another state transition's definition that occupied the latest state transition's space in the transition table.

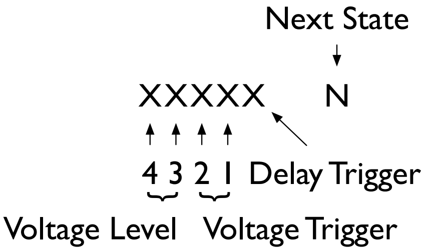


Figure A.4: Each input is represented by a `0`, `1` or `X` in the state transition list.

A.3.1 Running FSM designs in FPGA control panel

Click the *Run/Pause FPGA* button (green text) to start the FSM. FSM execution begins when the *State Machine Running* indicator lights up. A change to the state machine can be made quickly by pressing the *Run/Pause FPGA* button and editing the appropriate `.fsm` file in `FSMCreator`, saving the text file, and pressing *Run/Pause FPGA* again. Clicking the *Refresh Values* button will show any changes to the FSM parameters.

To close the FPGA control panel window, click the *Stop VI* button (red text) at the bottom right of the control panel. This will download the new state tables to the FPGA and restart execution.

Appendix B

eventDetector Matlab program usage.

B.1 Introduction

Nanopore data files store the digital representation of the ionic current flowing through a single nanopore (in our case a α-Hemolysin pore). The current of an unblocked channel has a characteristic amplitude referred to as the *open channel current*. As biological polymers are captured in the nanopore, they produce downward deflections in the open channel current. These deflections are the main feature of interest while processing a data file because each of these events provide information about the molecule/complex captured in the nanopore.

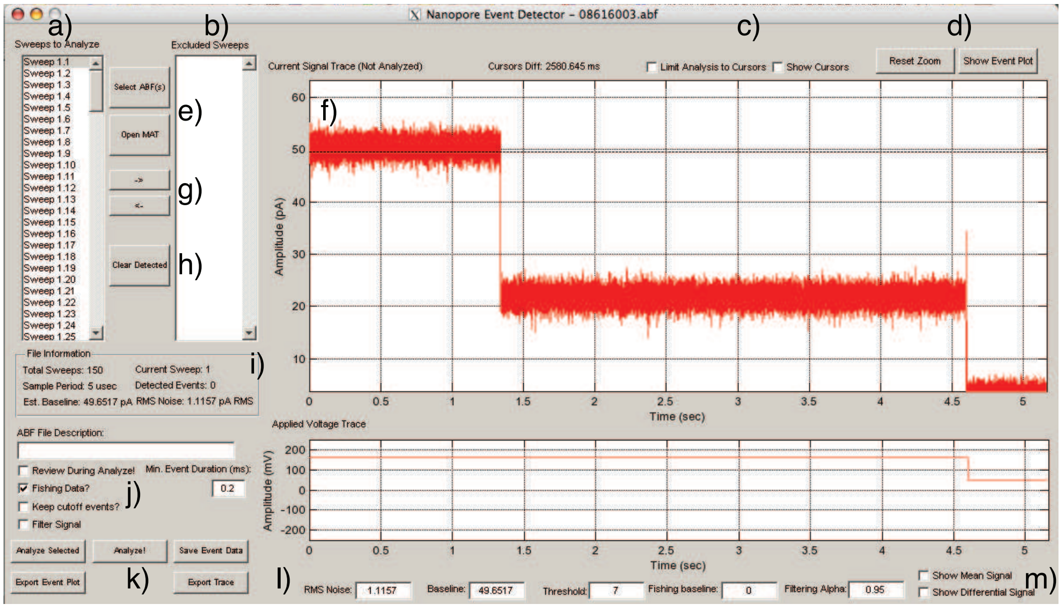
The `eventDetector` program runs under MATLAB and loads the nanopore current and voltage data stored in Molecular Devices' (Sunnyvale, CA) proprietary Axon ABF2 file format (.abf file extension), scans the signals, and extracts the location and amplitude of events. The summary statistics can then be studied to extract information about the molecules under test. With `eventDetector`, many of the post-processing steps are automated and the time necessary to analyze a data file is reduced.

B.2 The `eventDetector` GUI window

Upon running `eventDetector`, the program window will open and present the controls displayed in Figure B.1. Nanopore data files are divided into equal time segments, referred to as sweeps, for easier viewing. The main program window allows the user to review the sweeps of the data file and detect translocation events either a single sweep at a time or by processing all the sweeps at once. Processing an entire file at once has the advantage of detecting events that cross over into multiple sweeps, so single sweep analysis should only be used for testing detection settings prior to running a full file analysis. Before any event detection can occur, however, one or more ABF files needs to be loaded.

B.3 Loading ABF files

The `eventDetector` GUI can load one or more ABF files for event detection. Data files can only store a preset number of sweeps and the number of sweeps is programmed prior to data capture. Loading and analyzing multiple files eliminates the need to process each file from the same experiment individually, which occurs when a particular experiment lasts longer than a single data file can hold. The data from multiple files are loaded with the sweeps from each file labeled in the sweep list boxes



Sweep 1.1
Sweep 1.2
Sweep 1.3
Sweep 1.4
Sweep 1.5
Sweep 1.6
Sweep 1.7
Sweep 1.8
Sweep 1.9
Sweep 1.10
Sweep 1.11
Sweep 1.12
Sweep 1.13
Sweep 1.14
Sweep 1.15
Sweep 1.16
Sweep 1.17
Sweep 1.18
Sweep 1.19
Sweep 1.20
Sweep 1.21
Sweep 1.22
Sweep 1.23
Sweep 1.24
Sweep 1.25

Cursors Diff: 2580.645 ms

□ Limit Analysis to Cursors □ Show Cursors

File Information

Total Sweeps: 150 Current Sweep: 1
Sample Period: 5 usec Detected Events: 0
Est. Baseline: 49.6517 pA RMS Noise: 1.1157 pA RMS

ABF File Description:

□ Review During Analyze! Min. Event Duration (ms): 0.2
☑ Fishing Data?
□ Keep cutoff events?
□ Filter Signal

RMS Noise: 1.1157 Baseline: 49.6517 Threshold: 7 Fishing baseline: 0 Filtering Alpha: 0.95

□ Show Mean Signal
□ Show Differential Signal

Figure B.1: Data review and event detected is performed through the controls on main `eventDetector` window: a) sweeps to analyze list b) sweeps excluded from analysis list c) cursor options d) graph viewer controls e) open ABF and open MAT buttons g) include/exclude sweep controls h) clear detected events button i) the *File Information* frame displays the ABF file attributes j) event analysis options k) event analysis and data export controls l) nanopore signal attribute settings m) alternative nanopore signal views

(Figure B.1a, b). The sweep list boxes denote the file and sweep number using a two part notation. 'Sweep 2.34', for example, would refer to sweep 34 in file 2. The open files are listed in file number order in the title bar of the program window and in the ABF file selection dialog box. Any detected event that crosses between files, if any, is not included in the analysis due to the brief time delay the data logging software produces when creating a new file.

To load one or more ABF files, click the *Select ABF(s)* button to bring up the ABF selection dialog box shown in Figure B.2. From here, the user can click the *Load ABF* button to open dialog box where one or more ABF files can then be selected. Once

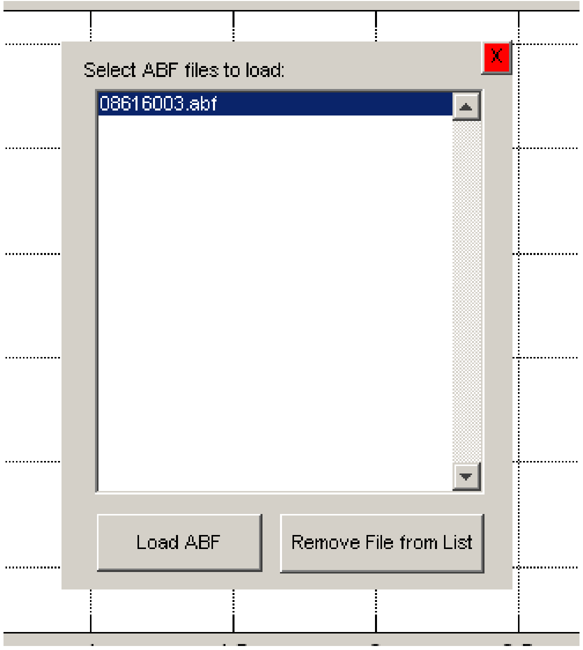


Figure B.2: The ABF selection dialog box manages the list of nanopore data files under analysis. Files can be added or removed from analysis using this window.

all the desired files are selected, click *Open* to return the ABF selection dialog. To add more files, repeat the *Load ABF* process outlined above as necessary. Remove files by selecting a file in the listbox and clicking the *Remove File from List* button. After all the desired files have been loaded in the listbox, click the red *X* in the top right corner of the dialog to finalize the file choices and return to the main `eventDetector` window. Once loaded, the sweep currently highlighted in the *Sweeps to Analyze* listbox (Figure B.1a) will be dispalyed in the signal window.

B.4 Analyzing data files

Before proceeding with detection of translocation events, several parameters need to adjusted to ensure accurate event identification. At the very least, the *ABF File Description*, *Baseline* and *RMS noise* need to be set. The *ABF file description* (Figure B.1j) should be entered immediately upon opening a data file to ensure the data is not labeled incorrectly. This description makes up part of the filename for any exported plots/data, so it should be as short possible while remaining informative.

B.4.1 Setting detection parameters

With constant voltage data files, `eventDetector` should accurately detect the baseline amplitude from the first sweep loaded in the data file unless the current signal is significantly distorted. In the case of a poor baseline detection, it can be set manually by either typing a picoamp value in the *Baseline* text box (Figure B.1l) or by dragging the baseline marker in the current signal window (Figure B.1f). When positioning the mouse above the baseline (black dashed line), the cursor will change from an arrow to a vertical resize cursor and the baseline can be dragged up or down as appropriate, as shown in Figure B.3. If the baseline estimation puts the baseline out of the viewable area of the current signal window, the user will need to type a picoamp value within the viewable window before fine-tuning using the mouse.

The first 512 samples of the file are used to estimate the RMS noise of the open channel signal. The magnitude of the noise determines how far below the baseline

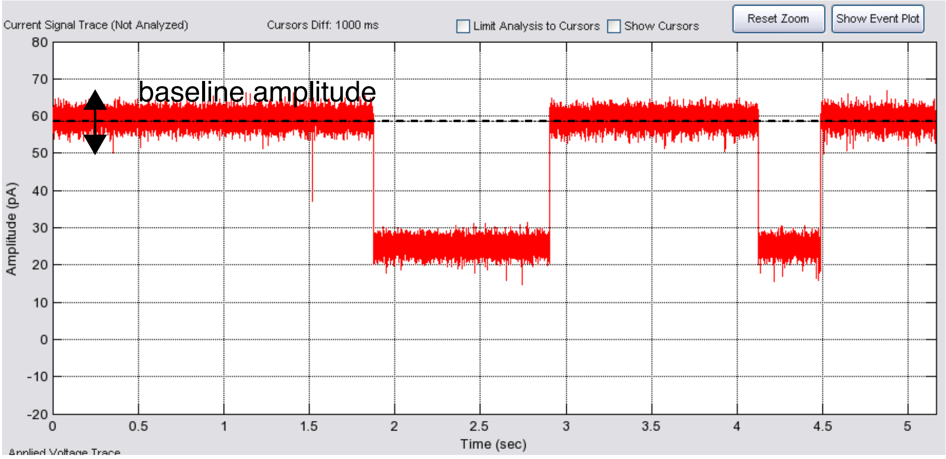


Figure B.3: The open channel current value, or baseline, can be set by dragging the black dashed line such that it overlaps with the open channel amplitude in the current signal window (Figure B.1f).

amplitude `eventDetector` needs to look to avoid mistaking noise spikes for events. If uncharacteristic noise happens to occur in the first 512 samples of the first sweep, the RMS noise will be incorrectly estimated and will need to be manually adjusted in the *RMS Noise* text box (Figure B.1l). By default, `eventDetector` looks for deflections in the open channel current that are greater than 7 times the RMS noise and classifies them as events. This RMS noise multiplier can be adjusted using the *Threshold* text box (Figure B.1l) and it is usually necessary to adjust it when working with less than 0.3 M KCl data. In addition to the RMS noise, `eventDetector` will load other file statistics from the ABF file and show them in the File Information frame (Figure B.1i).

Any sweeps that contain stuck molecules or otherwise unusable data can be excluded from analysis either before or after event detection. The *Sweeps to Analyze* listbox contains the sweeps to scan for events and the *Excluded Sweeps* listbox contains sweeps to ignore during analysis. The include and exclude buttons (Figure B.1g) move a sweep from one list to another. When the focus is in the *Sweeps to Analyze* lisbox, the up and down arrow keys will move to the previous and next sweeps in the file, respectively. As a shortcut while reviewing sweeps, the 'e' key on the keyboard will exclude that sweep from analysis in the same manner as the exclude button.

B.4.2 Performing sweep analysis

After setting the RMS noise and the baseline, `eventDetector` can detect translocation events. Click the *Analyze Selected* button to analyze the currently displayed sweep or click *Analyze!* to analyze all the sweeps one after another (Figure B.1k).

When performing an analysis of every sweep, a progress bar will appear above the `eventDetector` window showing the progress for the current file. There will be more than one progress bar in the case of a multi-file analysis. To cancel, close the progress bar window and agree "Are you sure?" dialog box. Previously analyzed sweeps will be skipped during subsequent analysis, so any single sweep detections performed can potentially interfere with detection of multi-sweep events. In most cases, the currently detected sweeps will need to be cleared by clicking the *Clear Detected* button (Figure B.1h) before performing a full file analysis.

During event detection, `eventDetector` looks at every contiguous portion of the current signal that is below the baseline current by $Threshold \times RMS\ Noise$. The transitions from the baseline current at the beginning and end of every segment are then trimmed to the first and last point the current signal changes by less than the RMS noise level defined for the file. The event's start and end points are then updated to these points. The length of the event, referred to as the molecule's dwell time, is calculated as the number of samples from the beginning to the end of the event multipled by the sample period of the signal. Events lasting less than the minimum event duration, set in *Min. Event Duration (ms)* text box, will not be saved during event detection. The default minimum event duration value is 0.2 ms. The event's amplitude is computed as the mean of the event's amplitude values.

Checking *Review During Analyze* (Figure B.1j) will cause each sweep's detected events to show up in the current signal window as `eventDetector` analyzes each sweep. This is useful for reviewing the data and check for anomalies while analyzing it, however, detection takes more time with this option enabled.

B.4.3 Analyzing fishing data

With fishing data, the *Fishing data* checkbox needs to be checked before performing event detection. Once checked, a fishing baseline marker (magenta dashed line) will appear in the current signal window in a similar manner as the original baseline marker. For most data files, the fishing baseline can be safely set to 0. If 0 pA events start appearing in the event plot, this will need to be increased slightly, to around 5 pA, to avoid detecting near zero amplitudes as events. The *Fishing Baseline* text box allows input of exact current values (Figure B.1l).

With fishing data, care needs to be taken when setting the baseline. The current signal only returns to open channel between tethered DNA molecule events. Consequently, many consecutive sweeps may not exhibit the open channel amplitude. Usually several sweeps need to be reviewed before the current returns to open channel, at which point, the baseline can be set as described in the previous section.

`eventDetector` uses the same process to detect events in fishing data as in constant voltage data with a few changes that exploit characteristics of the fishing current signal for classifying events as either probing events or regular events. Portions of the current signal that are below the baseline current by $Threshold \times RMS\ Noise$ and above 5 pA get marked as potential events. Next, the voltage signal surrounding the beginning and end of each event is analyzed to determine if there is a voltage change present. The absence or existence of a voltage change at the beginning and/or end of an event provides information for determining the type of event. Events can be classified as one of four types: regular (no voltage changes), initial capture (voltage change at end of event), probing (voltage change at beginning and end of event), and translocation (voltage change at beginning of event). This information can then be accessed later using MATLAB code.

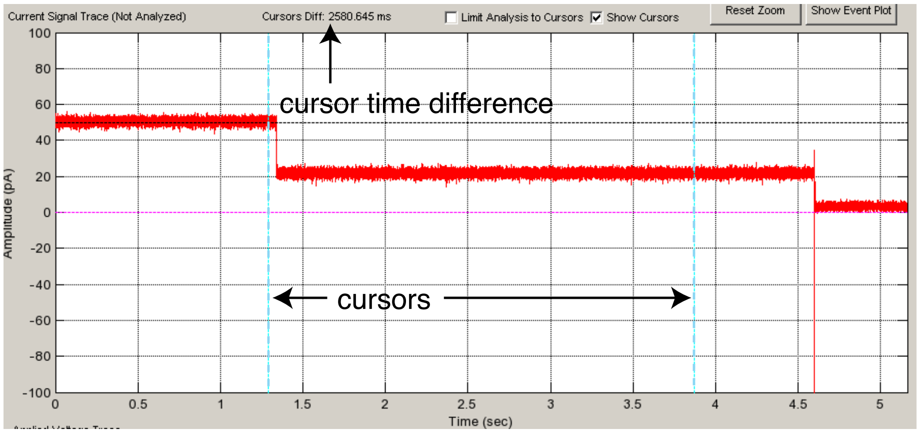
Different types of events get plotted differently in the 2-D event plot. Non-probing events appear as squares while probing events are plotted as dots. Each tethered DNA molecule is also tracked from the sequence of initial capture to probing to translocation events in the data. Each DNA molecule has a unique color on the event plot which is useful when determining how much any one DNA contributes to a distribution of events.

B.4.4 Using cursors to limit sweep analysis or measure signal length

In some cases, collected data cannot be expected to be continuous across multiple sweeps. This is primarily the case with constant voltage data that has the voltage reversed at the end of each sweep to clear any molecules stuck in the pore or to prevent the channel from gating. In this situation, the cursors can be enabled and positioned near the beginning and end of the event to identify the portion of the sweep that contains useful data, Figure B.4.

To enable the cursors, click the *Show Cursors* checkbox (Figure B.1c). Two vertical light blue dashed lines will appear 25% and 75% of the way through the current signal window. These lines can be dragged to the appropriate place in the sweep and any portion of the signal outside the two cursors will be ignored during event detection if the *Limit Analysis to Cursors* is checked. Events cutoff by the cursors during limited analysis, *i.e.* the event does not return to the baseline amplitude before the cursors mark the end of the sweep, can be included or excluded from analysis by checking or un-checking the *Keep cutoff events?* checkbox (Figure B.1j). When cutoff events are included, they appear as red dots on the 2-D event plot.

The cursors can also be used to measure the length of events or portions of signal. The time between the cursors is displayed above the current signal plot any time the cursors are visible (Figure B.4).



Current Signal Trace (Not Analyzed)

Cursors Diff: 2580.645 ms

☐ Limit Analysis to Cursors ☑ Show Cursors

Figure B.4: The current window cursors can exclude portions of a sweep from analysis or measure the duration of events or signal features.

B.4.5 Filtering current signal before analysis

The current signal can be filtered using a low-pass Butterworth filter with a cutoff frequency of 5kHz by enabling the *Filter Signal* checkbox (Figure B.1j). When checked, the signal will be filtered in the current signal window and during event detection.

To summarize, perform the following steps to analyze constant voltage data:

1. Load desired ABF file(s).
2. Set baseline amplitude to the middle of the open channel signal using either the text box or by dragging the black dotted baseline indicator in the graph window.
3. Check and ensure the pA RMS noise is accurate (below 2.0 pA RMS is typical).
4. Check all the sweeps and exclude any that contain stuck molecules or otherwise invalid data.

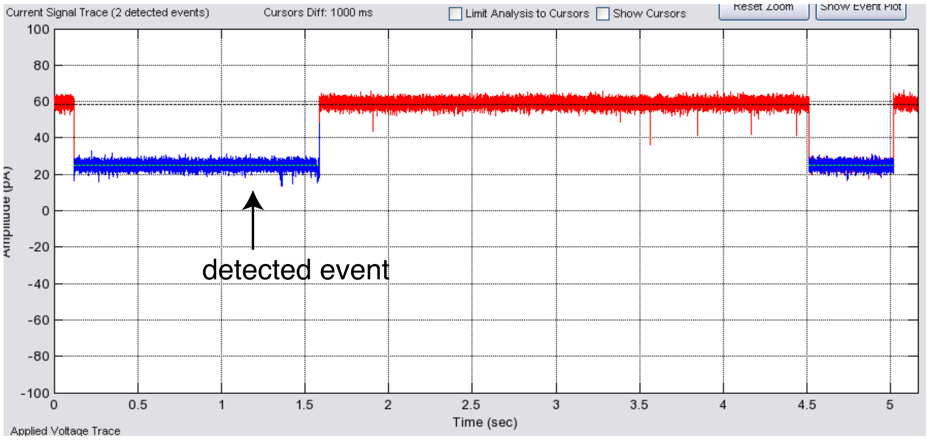


Figure B.5: Detected events are highlighted in blue for constant voltage events or magenta (not shown) for probing events from fishing data. The event's mean amplitude is plotted as a green dashed line.

5. Click the "Analyze!" button and check all the selected sweeps for events.

B.5 Reviewing detected events

After `eventDetector` has finishing analyzing the file or sweep, focus will return to the program window and the detected events can be reviewed by cycling through the *Sweeps to analysis* listbox. Detected events will be highlighted in blue for non-probing events and in magenta for probing events. A green line shows average amplitude in the current signal window, Figure B.5. The number of events detected in each sweep is displayed above the current signal window.

A 2-D event plot showing each detected event's amplitude and dwell time can be displayed by clicking the *Show Event Plot* button (Figure B.1d). Figure B.6 shows

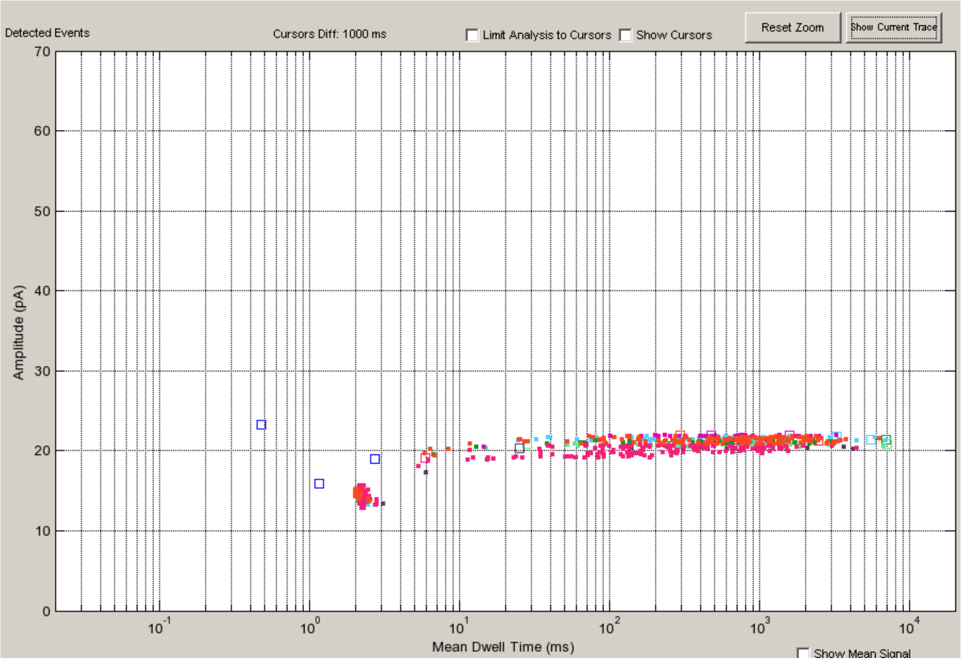


Figure B.6: Each dot on the 2-D event plot represents a single translocation event where the event's dwell time is plotted along the x-axis and the event's amplitude is plotted along the y-axis.

an example plot.

B.5.1 Adjusting plot view

The current signal, voltage signal and event plot windows can be zoomed in to examine fine details in the signal by clicking and dragging in the area along the x or y-axis, shown in Figure B.7. When the mouse cursor is in the appropriate area for zooming, the cursor will change from an arrow to a vertical or horizontal resize cursor, dependent on the zooming direction. While in the zoom area, dragging the mouse will display two dark blue dashed lines with one following the current position of the mouse pointer. The area between the two zoom lines will be enlarged once the mouse button

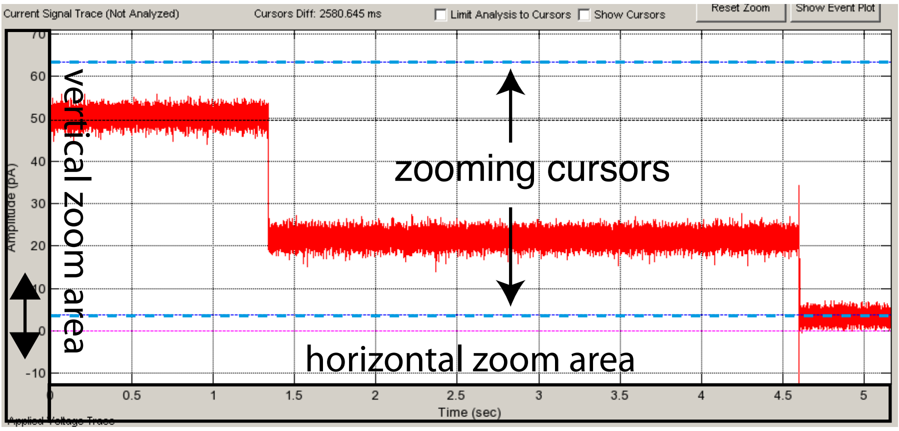


Figure B.7: Dragging the mouse and releasing the mouse button within the zoom area causes the signal window to zoom to the area between the cursors. Zooming can be performed on the current signal, voltage signal and event plot windows.

is released within the zoom area.

Additional signals can be superimposed on the current signal to show more information. Clicking the *Show Mean Signal* checkbox (Figure B.1m) displays the same first order low-pass filtered signal the FPGA uses when detecting current level changes. The amount of filtering is a function of the value of the *Filtering Alpha* text box. A differential representation of the current signal can be shown by clicking the *Show Differential Signal* checkbox.

B.6 Saving results and exporting plots

After the ABF file has been analyzed and event detection is complete, the results and the state of the `eventDetector` program can be saved for use with other

MATLAB programs/code or for loading back into `eventDetector` by clicking the *Save Event Data* button (Figure B.1k). This will create a `.mat` file in the same directory as `eventDetector.m` and have the following naming scheme:

filename1, ... **filenameN** - analyzed DD-Mon-Year_**ABF file description**.mat

The `.mat` file contains all the parameters currently entered in the `eventDetector` GUI window for later restoring the user's state (loaded files, detected events, detection parameters, etc.) in a data structure called **poreEventData**. Most of the entries in this file are only useful to `eventDetector` with a few exceptions.

All the detected event data is stored in the **poreEventData.detectedEvents** and **poreEventData.detectedEvents_ms** matrices. The **poreEventData.detectedEvents** matrix stores the data with a sample-based time scale, and the **poreEventData.detectedEvents_ms** matrix stores the data with a millisecond-based time scale. Each row of either matrix refers to an individual detected event, thus the number of detected events is equal to the number of rows.

Here is the breakdown of each column of **poreEventData.detectedEvents**:

poreEventData.detectedEvents = [

1, sweep number where event ends,

2, event start sample,

3, event end sample,

4, event start sweep number,

5, event end sweep number,

6, cutoff event flag,

7, DNA molecule number in case of fishing data,

8, event classification,

9, numer of samples before event a voltage change occurs,

10, file index for sweep event is located in,

11, minimum amplitude value for the event]

And for **poreEventData.detectedEvents_ms**:

poreEventData.detectedEvents_ms = [

1, sweep number where event ends,

2, event start time (ms),

3, event end time (ms),

4, event dwell time (ms),

5, average event amplitude (pA),

6, event start sweep number,

7, event end sweep number,

8, cutoff event flag,

9, DNA molecule number in case of fishing data,

10, event classification,

11, numer of samples before event a voltage change occurs,

12, file index for sweep event is located in]

Other useful structure entries include:

poreEventData.filename - cell array of filename,

poreEventData.pathname - cell array of corresponding pathnames,

poreEventData.numSamples - total number of samples in a sweep,

poreEventData.samplePeriod - number of μs between samples,

where the file index in **poreEventData.detectedEvents**$(x, 10)$ or **poreEventData.detectedEvents_ms**$(x, 12)$ indexes the appropriate file in **poreEventData.filename** and **poreEventData.pathname** for the event referenced by row x.

Clicking the *Export Event Plot* button (Figure B.1j) will save an `.eps` of the event plot (using the current zoom) with the same filename prefix as the `.mat` file, described above.

Clicking the *Export Trace* button (Figure B.1j) will take the current and voltage signal plots and export to an `.eps` file in the same directory as `eventDetector.m` using the current zoom. Accompanying this file will be a `.mat` file with the current, voltage and time vectors for the plot.

Bibliography

[1] Elio A Abbondanzieri, William J Greenleaf, Joshua W Shaevitz, Robert Landick, and Steven M Block. Direct observation of base-pair stepping by rna polymerase. *Nature*, 438(7067):460–5, Nov 2005.

[2] Mark Akeson, Daniel Branton, John J Kasianowicz, E Brandin, and David W Deamer. Microsecond time-scale discrimination among polycytidylic acid, polyadenylic acid, and polyuridylic acid as homopolymers or as segments within single rna molecules. *Biophysical journal*, 77:3227–33, 1999.

[3] Mark Bates, Michael Burns, and Amit Meller. Dynamics of dna molecules in a membrane channel probed by active control techniques. *Biophys J*, 84(4):2366–72, Apr 2003.

[4] Seico Benner, Roger J A Chen, Noah A Wilson, Robin Abu-Shumays, Nicholas Hurt, Kate R Lieberman, David W Deamer, William B Dunbar, and Mark Akeson. Sequence-specific detection of individual dna polymerase complexes in real time using a nanopore. *Nature Nanotechnology*, 2(11):718–24, Nov 2007.

[5] Orit Braha, Barbara Walker, Stephen Cheley, John J Kasianowicz, Langzhou Song,

J Eric Gouaux, and Hagan Bayley. Designed protein pores as components for biosensors. *Chem Biol*, 4(7):497–505, Jul 1997.

[6] Daniel Branton, David W Deamer, Andre Marziali, Hagan Bayley, Steven A Benner, Thomas Butler, Massimiliano Di Ventra, Slaven Garaj, Andrew Hibbs, Xiaohua Huang, Stevan B Jovanovich, Predrag S Krstic, Stuart Lindsay, Xinsheng Sean Ling, Carlos H Mastrangelo, Amit Meller, John S Oliver, Yuriy V Pershin, J Michael Ramsey, Robert Riehn, Gautam V Soni, Vincent Tabard-Cossa, Meni Wanunu, Matthew Wiggin, and Jeffery A Schloss. The potential and challenges of nanopore sequencing. *Nat Biotechnol*, 26(10):1146–53, Oct 2008.

[7] Chad A Brautigam and Thomas A Steitz. Structural and functional insights provided by crystal structures of dna polymerases and their substrate complexes. *Curr Opin Struct Biol*, 8(1):54–63, Feb 1998.

[8] Carlos Bustamante, Zev Bryant, and Steven B Smith. Ten years of tension: single-molecule dna mechanics. *Nature*, 421(6921):423–7, Jan 2003.

[9] Scott L Cockroft, John Chu, Manuel Amorin, and M. Reza Ghadiri. A single-molecule nanopore device detects dna polymerase activity with single-nucleotide resolution. *J Am Chem Soc*, 130(3):818–20, Jan 2008.

[10] Scott L Cockroft, John Chu, Manuel Amorin, and M. Reza Ghadiri. A single-molecule nanopore device detects dna polymerase activity with single-nucleotide resolution. *J Am Chem Soc*, 130:818–+, Jan 2008.

[11] Maria E Dahlberg and Stephen J Benkovic. Kinetic mechanism of dna polymerase i (klenow fragment): identification of a second conformational change and evaluation of the internal equilibrium constant. *Biochemistry*, 30(20):4835–43, May 1991.

[12] Ravindra V Dalal, Matthew H Larson, Keir C Neuman, Jeff Gelles, Robert Landick, and Steven M Block. Pulling on the nascent rna during transcription does not alter kinetics of elongation or ubiquitous pausing. *Mol Cell*, 23(2):231–9, Jul 2006.

[13] David W Deamer and Daniel Branton. Characterization of nucleic acids by nanopore analysis. *Accounts of chemical research*, 35:817–25, 2002.

[14] Cees Dekker. Solid-state nanopores. *Nature Nanotechnology*, 2(4):209–15, Apr 2007.

[15] S Doublié, S Tabor, A M Long, C C Richardson, and T Ellenberger. Crystal structure of a bacteriophage t7 dna replication complex at 2.2 a resolution. *Nature*, 391(6664):251–8, Jan 1998.

[16] Olga K Dudko, Jérôme Mathe, Attila Szabo, Amit Meller, and Gerhard Hummer. Extracting kinetics from single-molecule force spectroscopy: nanopore unzipping of dna hairpins. *Biophys J*, 92(12):4188–95, Jun 2007.

[17] Adrian N Fehr, Charles L Asbury, and Steven M Block. Kinesin steps do not alternate in size. *Biophys J*, 94(3):L20–2, Feb 2008.

[18] Arthur Gill. Introduction to the theory of finite-state machines. *McGraw-Hill*, 1962.

[19] William J Greenleaf and Steven M Block. Single-molecule, motion-based dna sequencing using rna polymerase. *Science*, 313(5788):801, Aug 2006.

[20] William J Greenleaf, Michael T Woodside, and Steven M Block. High-resolution, single-molecule measurements of biomolecular motion. *Annual review of biophysics and biomolecular structure*, 36:171–90, Jan 2007.

[21] Kristina M Herbert, William J Greenleaf, and Steven M Block. Single-molecule studies of rna polymerase: motoring along. *Annu Rev Biochem*, 77:149–76, Jan 2008.

[22] Kristina M Herbert, Arthur La Porta, Becky J Wong, Rachel A Mooney, Keir C Neuman, Robert Landick, and Steven M Block. Sequence-resolved detection of pausing by single rna polymerase molecules. *Cell*, 125(6):1083–94, Jun 2006.

[23] Breton Hornblower, Amy Coombs, Richard D Whitaker, Anatoly Kolomeisky, Stephen J Picone, Amit Meller, and Mark Akeson. Single-molecule analysis of dna-protein complexes using nanopores. *Nat Methods*, 4(4):315–317, 2007.

[24] Nick Hurt, Hongyun Wang, Mark Akeson, and Kate R Lieberman. Specific nucleotide binding and rebinding to individual dna polymerase complexes captured on a nanopore. *In press, Journal of the American Chemical Society*, 2009.

[25] Sean J Johnson, Jeffrey S Taylor, and Lorena S Beese. Processive dna synthesis observed in a polymerase crystal suggests a mechanism for the prevention of frameshift

mutations. *Proceedings of the National Academy of Sciences of the United States of America*, 100(7):3895–900, Apr 2003.

[26] Catherine M Joyce and Stephen J Benkovic. Dna polymerase fidelity: kinetics, structure, and checkpoints. *Biochemistry*, 43(45):14317–24, Nov 2004.

[27] Catherine M Joyce, Olga Potapova, Angela M Delucia, Xuanwei Huang, V Basu, and Nigel D F Grindley. Fingers-closing and other rapid conformational changes in dna polymerase i (klenow fragment) and their role in nucleotide selectivity. *Biochemistry*, 47(23):6103–6116, May 2008.

[28] Catherine M Joyce and Thomas A Steitz. Function and structure relationships in dna polymerases. *Annu Rev Biochem*, 63:777–822, Jan 1994.

[29] John J Kasianowicz, E Brandin, Daniel Branton, and David W Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proc Natl Acad Sci U S A*, 93(24):13770–3, Nov 1996.

[30] Matthew H Larson, William J Greenleaf, Robert Landick, and Steven M Block. Applied force reveals mechanistic and energetic details of transcription termination. *Cell*, 132(6):971–82, Mar 2008.

[31] Ying Li, Sergey Korolev, and Gabriel Waksman. Crystal structures of open and closed forms of binary and ternary complexes of the large fragment of thermus aquaticus dna polymerase i: structural basis for nucleotide incorporation. *EMBO J*, 17(24):7514–25, Dec 1998.

[32] Jérôme Mathe, H Visram, V Viasnoff, Yitzhak Rabin, and Amit Meller. Nanopore unzipping of individual dna hairpin molecules. *Biophysical journal*, 87:3205–12, 2004.

[33] Yara X Mejia, Hanbin Mao, Nancy R Forde, and Carlos Bustamante. Thermal probing of e. coli rna polymerase off-pathway mechanisms. *J Mol Biol*, 382(3):628–37, Oct 2008.

[34] Jeffrey R Moffitt, Yann R Chemla, Steven B Smith, and Carlos Bustamante. Recent advances in optical tweezers. *Annu Rev Biochem*, Feb 2008.

[35] Thomas T Perkins, Ravindra V Dalal, Paul G Mitsis, and Steven M Block. Sequence-dependent pausing of single lambda exonuclease molecules. *Science*, 301(5641):1914–8, Sep 2003.

[36] Robert F Purnell, Kunal K Mehta, and Jacob J Schmidt. Nucleotide identification and orientation discrimination of dna homopolymers immobilized in a protein nanopore. *Nano Lett*, 8(9):3029–34, Sep 2008.

[37] Minsoung Rhee and Michael Burns. Nanopore sequencing technology: research trends and applications. *Trends in biotechnology*, 24:580–6, 2006.

[38] Bert Sakmann and Erwin Neher. Single-channel recording. *New York : Plenum Press*, 1995.

[39] Joshua W Shaevitz, Elio A Abbondanzieri, Robert Landick, and Steven M Block.

Backtracking by single rna polymerase molecules observed at near-base-pair resolution. *Nature*, 426(6967):684–7, Dec 2003.

[40] Langzhou Song, Michael R Hobaugh, Christopher Shustak, Stephen Cheley, Hagan Bayley, and J Eric Gouaux. Structure of staphylococcal alpha-hemolysin, a heptameric transmembrane pore. *Science*, 274(5294):1859–66, Dec 1996.

[41] Stephen M Trimberger. Field-programmable gate array technology. *Springer*, 1994.

[42] Carolina Tropini and Andre Marziali. Multi-nanopore force spectroscopy for dna analysis. *Biophysical journal*, 92(5):1632–7, Mar 2007.

[43] Wenonah Vercoutere and Mark Akeson. Biosensors for dna sequence detection. *Current opinion in chemical biology*, 6(6):816–22, Dec 2002.

[44] Wenonah Vercoutere, Stephen Winters-Hilt, Veronica S DeGuzman, David W Deamer, SE Ridino, JT Rodgers, HE Olsen, A Marziali, and Mark Akeson. Discrimination among individual watson-crick base pairs at the termini of single dna hairpin molecules. *Nucleic acids research*, 31:1311–8, 2003.

[45] Meni Wanunu and Amit Meller. Chemically modified solid-state nanopores. *Nano Lett*, 7(6):1580–5, Jun 2007.

[46] Noah A Wilson, Robin Abu-Shumays, Seico Benner, Kate R Lieberman, David W Deamer, Mark Akeson, and William B Dunbar. Finite state machine control of single dna and enzyme-bound dna molecules in a nanopore. *Biodevices Conference*, 2008.

[47] Noah A Wilson, Robin Abu-Shumays, Brett Gyarfas, Hongyun Wang, Kate R Lieberman, Mark Akeson, and William B Dunbar. Electronic control of dna polymerase binding and unbinding to single dna molecules. *Submitted to ACS Nano*, 2009.