1-20-2006

Power Signal Analysis of Channel Current Signal Using HMM-EM and Time Domain FSA

Anand Prabhakaran
University of New Orleans

POWER SIGNAL ANALYSIS OF CHANNEL CURRENT SIGNAL USING HMM-EM AND TIME DOMAIN FSA

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
The Department of Computer Science

by

Anand Prabhakaran

B.E., University of Madras, India, 2003

December 2005

ACKNOWLEDGEMENTS

I take this opportunity to thank my advisor Prof. Dr. Stephen Winters-Hilt for his supervision and advising on my Masters thesis. I would also like to thank the Computer Science department, chairman and committee members for their support during my entire Masters program.

Finally, my parents overwhelming support all the way.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The Nanopore Detector using α-hemolysin channel transcribes kinetics of a single molecule along the nanometer-scale pore. The transcribed data is represented by electrical measurements. We present accurate and computationally inexpensive tools to analyze single molecule kinetics. The HMM-EM level projection method de-noises data, retaining the transitions with very high precision. This approach doesn't require input number of levels. Another advantage is the minimal tuning required. The levels are then identified using Finite State Automata (FSAs). Spike Detector algorithm analyzes spikes characterizing behavior of molecule in pore. No commercial tools available are capable of analyzing spikes in presence of noise. The formulation of HMM-EM, FSAs and Spike Detector together provides a robust method for analysis of channel current data. Application of these methods is described for Vercoutere channel blockade dataset which contains signals of radiated and non-radiated molecules. The tools developed were used successfully to differentiate between these two molecules.

1. INTRODUCTION

Nanopore detectors serve as a high throughput single molecule identification device based on the molecules distinct physical, chemical and electrical properties. The nanopore detector measures the ionic current blockages caused by the molecule in the nanometer-scale α-hemolysin channel pore. The DNA molecule translocates along the pore producing a signature electrical conductance. The α-hemolysin protein channel self-assembles with high fidelity and reproducibility and the diameter of the channel is just enough for the translocation to take place.

The DNA molecule translocating along α-hemolysin pore provides valuable information about the kinetics of the molecule. This property is captured as electrical signal under noisy environment and varying temperature forming complex channel current blockade data.

In our nanopore signal analysis, HMM-EM approach is used to de-noise and extract a feature vector from the blockade data which can be used to classify molecules. Hidden Markov Models (HMMs) can characterize current blockades by identifying a sequence of sub-blockades as a sequence of state emissions. HMMs have also been used to estimate state transition and emission probabilities on sequential data in more general contexts, including genomic analysis (Stormo, 2000) and voice recognition (Jelinek, 1997). The parameters of an HMM are usually estimated using a method called Expectation/Maximization (Durbin, 1998). The multiclass computational scalability tends to favor the use of HMMs as feature extractors. FSA's and Channel Current Spike Detectors further extract special feature vectors with FSAs primarily detecting the levels and the rotational kinetics of the molecule. The channel current analysis tools discussed here are capable of extracting high fidelity features for the purpose of training using Support Vector Machines (SVMs) which is then used to classify the molecule.

2. BACKGROUND AND GENERAL METHODS

Channel Current blockade signal acquired from the nanopore detector is the fingerprint of a DNA molecule indicating a variety of attributes like binding (DNAprotein), fraying, and conformational kinetics. This electrical signal data is stored in a binary format using Axon Technologies patch-clamp amplifier. The signal is introduced to several noise sources which make it difficult to transform data directly into features which is then used for training and classification. Thus the challenge lies in efficient noise removal while maintaining the signal transitions with accuracy. We discuss the design of tools to de-noise and extract features from the noisy channel current signal.

2.1 α-Hemolysin Channel Current Detector

A Channel Current signal is data acquired from the nanopore detector which consists of α-hemolysin channel. It is a nanometer dimension pore approximately 2nm in diameter. A DNA molecule is allowed to enter this pore and it modulates the ionic current flow through the channel due to its size and motion. This variation in current flow forms a typical pattern for a molecule. A current flow or the Channel Current signal for a 9-base pair hairpin is shown in Figure 2.1.

Figure 2.1: A 9-base pair hairpin channel current blockade signal.

A detailed trace of the molecule is the signal shown in Figure 2.2. At the start of each analysis, the channel voltage is reset to 0 mV. Then a potential difference of 120 mV is applied which

results in an open channel current of 120 pA which is depicted as Phase A. The DNA molecule is then pulled into the pore due to the applied potential which results in a decrease of the electric current. This trace is shown in Phase B and is know as the "Start of the Blockade" or simply "Blockade Current" or "Blockade". The end of blockade is reached when the molecule is ejected out of the pore by applying a reverse potential of about -40 mV. This is characterized by a sharp spike at the blockade end. Typically this analysis of a molecule is traced for 180 seconds sampled at 20 μS.

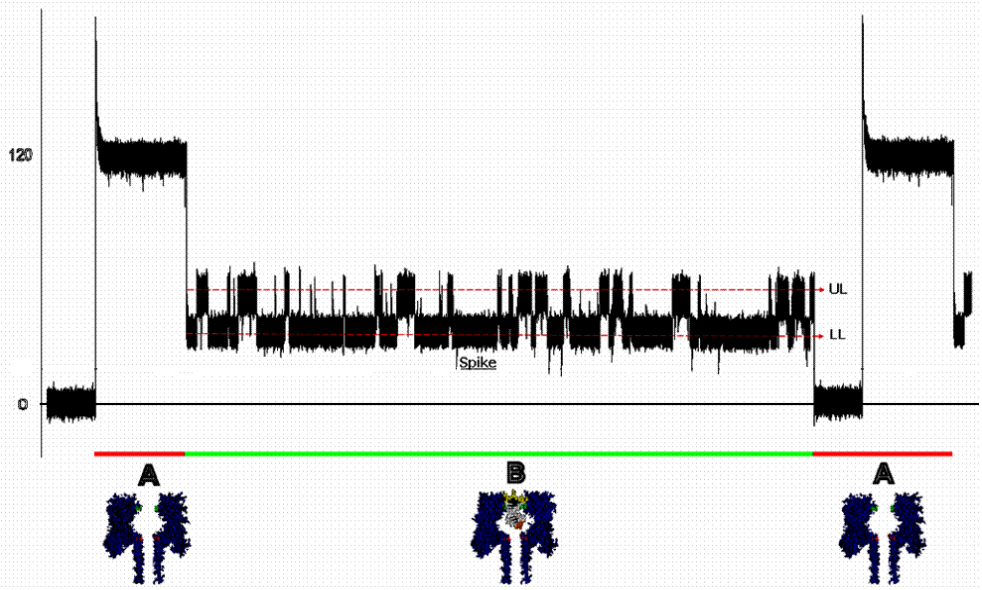


Figure 2.2: A trace diagram of 9-base pair hairpin molecule.

During the blockade i.e. Phase B, there are 3 major dominant levels often called as the Lower Level (LL), Upper Level (UL) and the Intermediate Level (IL). The blockade always starts with the IL followed by the toggling between the UL and the LL.

2.2 Noise in Blockade Signal

Noise can be defined as an unwanted signal that interferes with the measurement of another signal. A noise itself is a signal that conveys information regarding the source of the noise. The sources of noise are many, and vary from AF noise emanating during moving, vibrating or colliding of molecules in the pore. The nanopore detector itself is very sensitive to

AF noise. The main noise is constituted by the Gaussian Noise which is explained by the division of White Noise and Thermal Noise.

2.2.1 White Noise

White noise is defined as an uncorrelated noise process with equal power at all frequencies. A noise that has the same power at all frequencies in the range of $\pm\infty$ would necessarily need to have infinite power, and is therefore only a theoretical concept. However a band-limited noise process, with a flat spectrum covering the frequency range of a band-limited system, is to all intents and purposes from the point of view of the system a white noise process.

2.2.2 Thermal Noise

Thermal noise is also known as Johnson noise, is generated by the random movements of thermally energized particles. The concept of thermal noise has its roots in thermodynamics and is associated with the temperature-dependent random movements of free particles such as the molecule in the pore. Although these random particle movements average to zero, the fluctuations about the average constitute the thermal noise.

2.2.3 Membrane Noise

Membrane noise, the inherent electrical fluctuations in biological membranes, is ultimately the source of membrane excitability. It signifies the fluctuations in current or voltage caused by the random opening and closing of ion channels. In the nanopore context the opening and closing of the mouth of the pore due to the translocation causes a membrane excitability of the lipid bilayer. This constitutes the membrane noise in the nanopore detector.

2.2.4 Device Noise

Device noise can be defined as the distortions introduced in the system by the electrical components being used. Often two close wires can cause an inductive effect which can produce a reverse current. These sources of noises cannot be eliminated altogether by minimized by using

appropriate compensating elements in the system. The electrical conductance across the channel is amplified using a path-clamp amplifier and other digitized systems like A/D converter. This constitutes the device noise in the nanopore detector.

2.3 Channel Current Signal

As explained in the above sections, the blockade current associated with one molecule is distinguishable from the other. Features are the individual characteristic properties of the phenomena being observed. Choosing discriminating and independent features is the key factor towards accurate statistical pattern classification. In the Channel Current signal the key features are the lifetimes of levels i.e. UL and LL, ULA and ULB (For temperature data where ULA and ULB are the upper level toggling in the UL) and the spikes. These features are extracted from the first 100 ms of the blockade.

2.3.1 UL and LL

These are the durations of Upper Level and Lower Level life times. Typically this indicates the binding and unbinding of the molecule to the pore of the channel during its downward movement towards the channel. This is shown in Figure 2.3. The red line

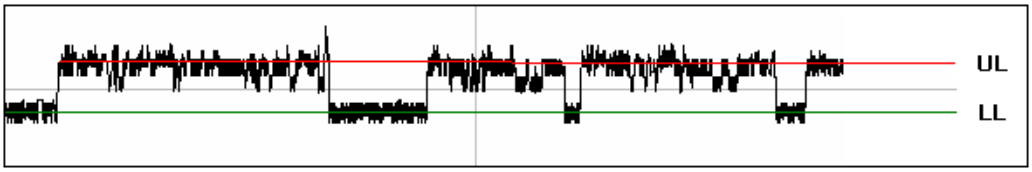


Figure 2.3: Toggling between UL and LL levels.

indicates the Upper Level and green line indicates the Lower Level. As you can see from the image, each Level has a range i.e. a dominant mean, surrounded by the signal noises. So to detect the transitions and retrieve this information the signal must be de-noised.

2.3.2 ULA and ULB

These are the durations of Upper Level A and Upper Level B life times. Typically this indicates the rotation of the molecule along the pore. This is called the Rotational Kinetics of a molecule which is observed in the temperature data. The temperature data is the observation of an individual molecule at varying temperatures. This is shown in Figure 2.4.

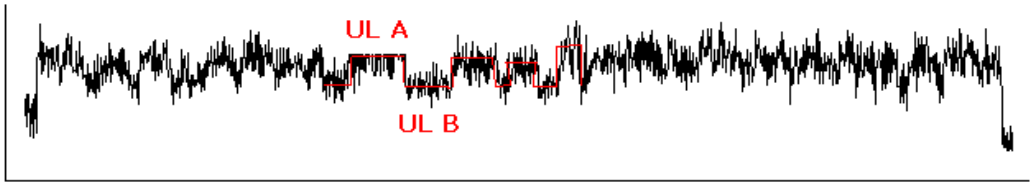


Figure 2.4: Toggling between ULA and ULB levels in the UL.

2.3.3 Spikes

Spikes are defined as an impulse signal for a short duration of time. The point of interest is the spike counts from the Lower Level as a result of the molecule fraying. Chapter 4 explains this in detail in which different methods for spike detection are presented.

2.5 Classification of Features

The features extracted using the channel current analysis tools explained here are trained and classified using Winters-Hilt SVM on Vercoutere channel blockade dataset. Once the SVM is trained it could classify the molecules instantly on the fly.

2.6 Existing Methods

There are tools like Bio-Patch and QuB to analyze single channel blockade data. These methods use algorithms like mean, median filter (smoothening), frequency filters, Gaussian, polynomial fitting and basic Markovian model. Bio-Patch a DOS based program analyses the blockade data. It uses the Gaussian and FIR filters as its primary algorithm to clean the noise by

smoothening. The two major disadvantages with this approach are one, accuracy and data reliability and two, the expert user intervention required to feed tuning parameters. These smoothening algorithms work effective in contexts such as audio/speech processing etc. But in blockade data analysis subtle information is lost. Another major disadvantage with FIR filters is the requirement of 2 power N samples. So as the number of samples increase the processing becomes increasingly tedious.

QUB is software developed by State university of New York for the analysis of single channel patch clamp records. The user selects a segment for each level (class) i.e. manually selects a segment each for LL and UL. These are the input classes. This requires expert usage for selecting LL and UL and prominently IL, ULA and ULB. The program then uses markovian model with calculated mean and standard deviation from the selected segments and then calculates the likelihood. Figure 2.5 shows QUBs inaccurate detection of start of signal. The open channel average was calculated as 129.97 pA. Since the RC transition averages around 160 pA, it detects an inaccurate transition in the open channel to UL.

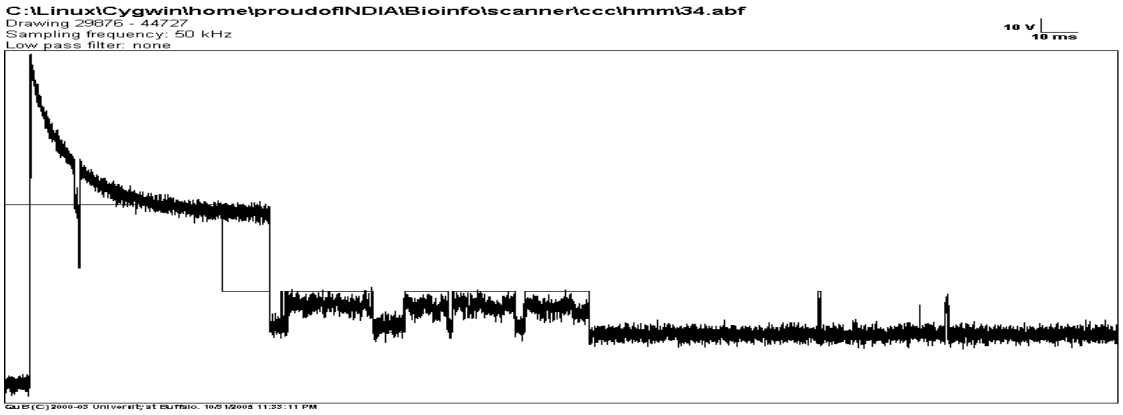


Figure 2.5: Blockade data processed by QUB, In-accurate detection of start of signal.

Figure 2.6 shows another inaccurate transition from LL to UL. A careful analysis of the signal shows a noisy segment which looks like a transition to happen but is actually LL noise. The disadvantage of QUB is that it calculates likelihood at each stage in presence of noise which makes the program to make inaccurate transitions. Our approach to this problem is presented

here. At each processing stage the noise is being reduced which causes the likelihood estimate much more accurate.

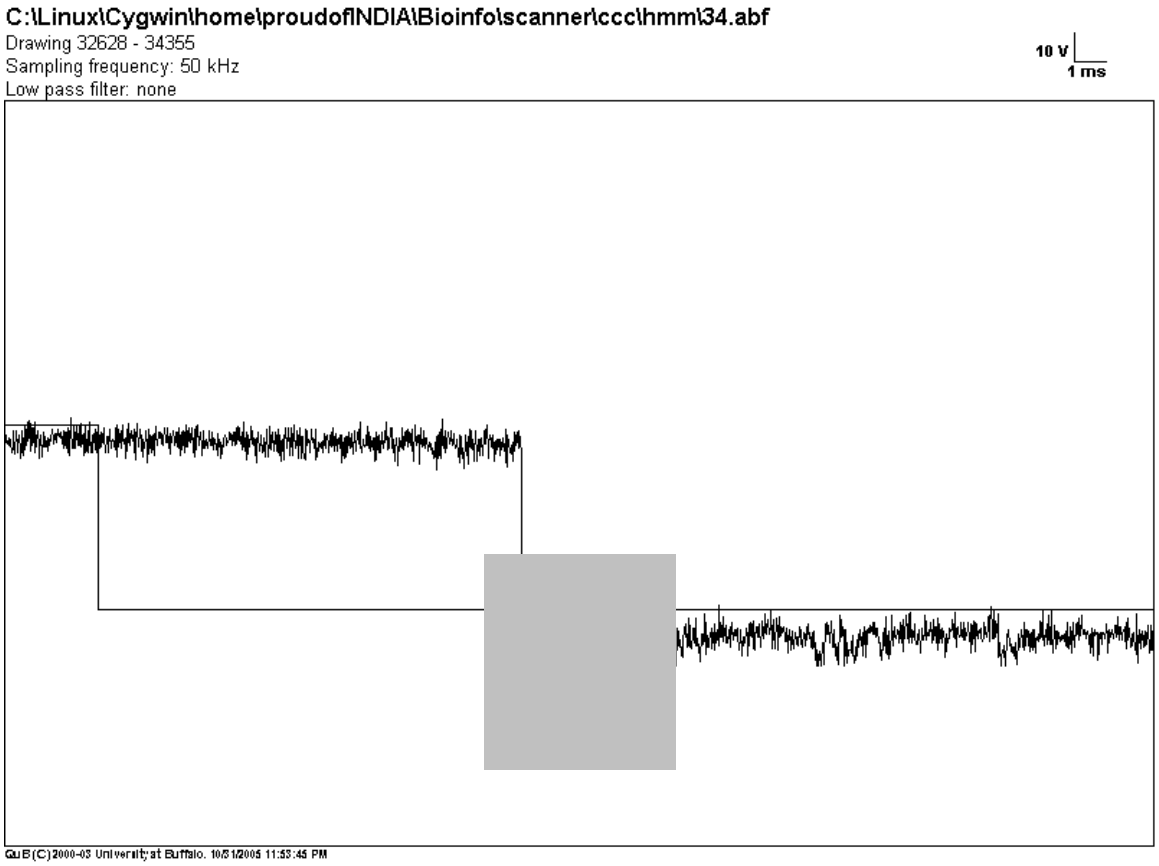


Figure 2.6: Blockade data processed by QUB, In-accurate detection of level transitions.

3. METHODS

3.1 HMM-EM Algorithm

A Hidden Markov Model (HMM) is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters, from the observable parameters, based on this assumption. The extracted model parameters can then be used to perform further analysis, in the case of Channel Current signal we would like to predict the signal level.

A basic sketch of the HMM-EM and Projection algorithm is described below.

1. Pre-Process the CC signal.
2. Initialize HMM parameters.
3. Calculate Forward-Backward
4. Calculate Expected Values i.e. new emission and transition probabilities.
5. Adjust parameters to maximize the likelihood of these expected values.
6. Iterate till optimal convergence.

3.1.1 Signal Pre-Processing

The raw signal is pre-processed before the HMM processing. The first 100 ms of the blockade is taken as the raw signal. This data is quantized and compressed. The quantization involves shifting the entire signal level with respect to ideal channel current signal with prototype baseline of 120 pA. This step facilitates the use of 50 possible states from 20 pA to 69 pA corresponding to a current blockade. The compression step involves reduction of the signal size by a factor of 8. This is done by calculation of average of the window frame sized 8. Hence, the signal size is reduced from 5000 samples to 625 samples. This is necessary to speedup the HMM for real-time operation, as the HMM algorithm needs construction of a dynamic programming table of dimensions states X sample.

3.1.2 Parameters Initialization

The HMM emission states consists of 50 possible states each corresponding to a current blockade of 20 pA to 69 pA. So state 10 refers to 30 pA blockade level. The prior probabilities are initialized to the probability distribution of different states of this pre-processed signal. From the distribution shown in Figure 3.1 we can see 2-4 dominant levels which references to LL, IL and UL and maybe a bifurcation of the UL. The initial emission probabilities were initialized using discrete Gaussian function with mean μ and variance σ and the transition probabilities were initialized using the initial emissions.

Figure 3.1: Distribution of initial states.

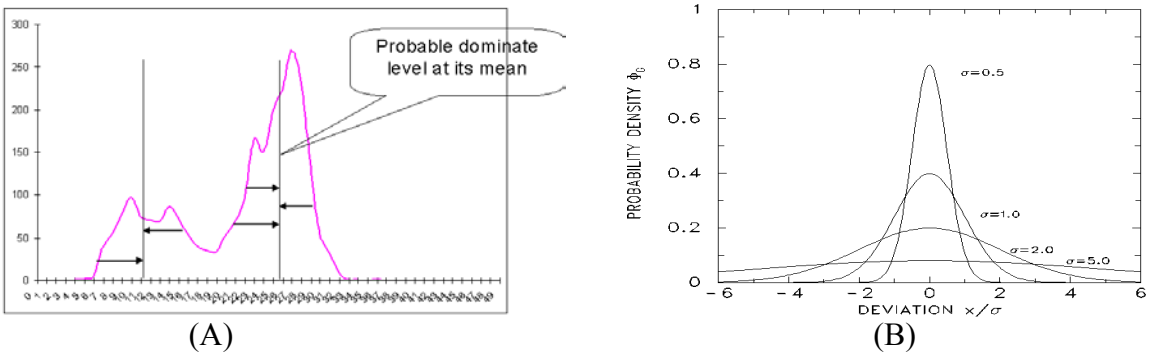
3.1.3 Expectation-Maximization (EM)

An Expectation Maximization (EM) algorithm is a fast maximum likelihood parameter estimation algorithm for partially observed data. The EM algorithm alternates between performing an expectation (E) step, which computes the expected value, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters given the data and setting the variables to their expectation. In practice we iterate the EM as it converges to a local maximum of the observed likelihood function. Once it converges to an optimum, we need to trace back the best possible state path. This is done using the Viterbi algorithm. It must be noted that EM is not an algorithm but a class where the HMM-EM is the Baum-Welch algorithm.

3.1.4 Gaussian Projection

As described in section 3.1.3 the EM algorithm converges to local maximum. To de-noise the signal we need to push the estimates by adjusting its emissions. We have seen the model follows a Gaussian; hence we increase the variance by a factor to rapidly push states towards their dominant level. Thus this is the tunable parameter in the entire approach which shifts the intensity of de-noising. This is a major advantage over existing methods. Figure 3.2.a shows the distribution and desired direction of the states shift. By increasing the variance the emissions distribution becomes much flatter that more transitions are possible from one state nearer to its dominant level.

Figure 3.2.b shows the Gaussian distributions for various values of variances. It can be noted as the variance increases the curve becomes much flatter. For example, the calculated variance from the emissions is 0.5 and we project by increasing its variance by 300% which is 2.0 then the curve extends till -4 to +4 rather than -2 to +2 for variance 0.5



(A) (B)

Figure 3.2: A) States moving towards the dominant level's mean. B) Gaussian distribution for various values of variance σ.

This technique allows those states in between -4 to -2 and +2 to +4 to move towards zero, thereby efficiently shifting the states towards its dominant.

3.2 TIME DOMAIN FSA

FSA or the Finite State Automaton is a model of a very basic machine, which can be in one of a finite number of states. In certain conditions, it can switch to another state. This is called

a *transition* . When the automaton starts working, it can be in one of its initial states. There is also another important subset of states of the automaton: the final states. If the automaton is in a final state when it stops working, it is said to *accept* its input. The input is a sequence of symbols. The interpretation of the symbols usually represent events, or can be interpreted as ``the event that a particular data became available". The symbols must come from a finite set of symbols, called the *alphabet* . If a particular symbol in a particular state triggers a transition from that state to another one, that transition is *labeled* with that symbol. The labels of transitions can contain one particular symbol that is not in the alphabet.

It is convenient to present automata as directed graphs. The vertices denote states. They are portrayed as small circles. The transitions form the edges - arcs with arrows pointing from the source state (the state where the transition originates) to the target state. They are labeled with symbols. Unless it is clear from the context, the initial states have short arrows that point to them from ``nowhere". The final states are represented as two concentric circles.

3.2.1 τ-FSA (Time Domain FSA)

The Channel Current signal is subjected to a τ-FSA to detect the start of the blockade, the transitions from and between UL, LL and IL and the end of blockade.

3.2.2 Detecting Transitions and Level Durations using τ-FSA

The τ-FSA was used to detect transitions and calculate the level durations. First the signal is pre-processed to retrieve some statistics about the signal which helps in dynamic adjusting of cutoff-variables in the FSA model. The entire duration of the signal was scanned sequentially and slotted to 80 states by converting the floating value of current in pA to its integer. The 80 states were between 20 pA to 100 pA which is the range of the blockade intensity. The probability of each state was calculated and the distribution showed a fine mixed Gaussian curve with 2 dominating Gaussians. These 2 dominating Gaussians were referenced to the UL and the LL.

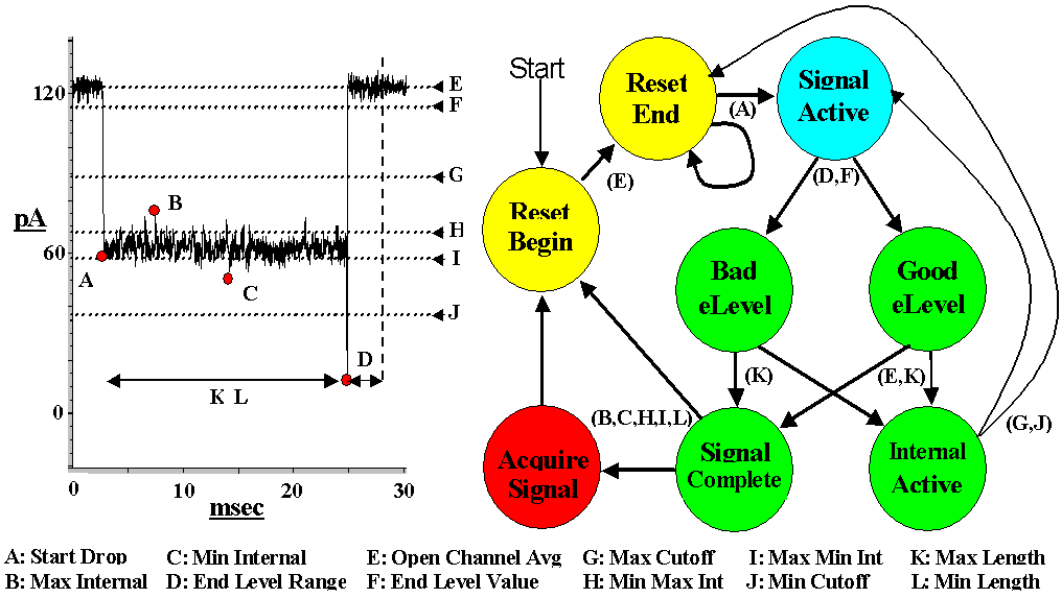


Figure 3.3: τ-Finite State Automaton (FSA).

Then a *Peak Detector* is used to detect the peaks of the dominant Gaussians. These values of the peaks were referenced to the approximate mean (pA) of the UL and LL. The peak detector was programmed using a low-pass filter which sequentially cutoff the lower frequencies. A smoothening cutoff constant was used to eliminate local peaks. Using these approx. UL and LL mean values, the approx. margin (pA) separating the UL and LL was determined which referenced to the lower density between these 2 dominant peaks. This is called the ULLL Margin. To detect the Intermediate Level, an approximate calculation of a range where IL toggles was done. This range was named as the Lower Range IL and Upper Range IL which was calculated as averages between UL and LL to the ULLL Margin.

3.2.2A Detecting Blockade

The signal is scanned sequentially using a moving window of size 100. The signal hovers around 120 pA which is called baseline and its average is recorded for the last 1000 samples i.e. 20 ms. This baseline average is used to quantize the signal to the prototype. The signal drops to about half its intensity which indicates the entry of the molecule into pore. Now the start of blockade is recorded and lasts until a sharp downward spike signals the end of the blockade by

jumping back to either zero or a sharp rise indicating the molecule ejection from the pore. Now the blockade is processed to check if the state is good or bad. If the blockade does not last for at least 100 ms or conforms to the range of the approx. LL average then the blockade is discarded as bad end level. Then the signal reset is activated to again start detecting further blockades in the signal.

An alternate method can be used to detect the blockade of the signal known as the Differential RC process. In this method a 2-bit window is processed and moved along time. The product of absolute difference and the RC constant produced the differential data. This is also known as the RC differentiator and is the logic used in the IC differentiator Chip. The advantage of this method is the blockade intensity is trimmed down to near zero intensity which will now help in easily detecting the blockade. The baseline still hovers around the 120 pA following the RC path of the original signal baseline. The drop from baseline to blockade is retained very precisely. Figure 3.4 shows pre and post processed signals with multiple blockades. Figure 3.5 shows an amplified image area of the first blockade's pre and post processed signals.

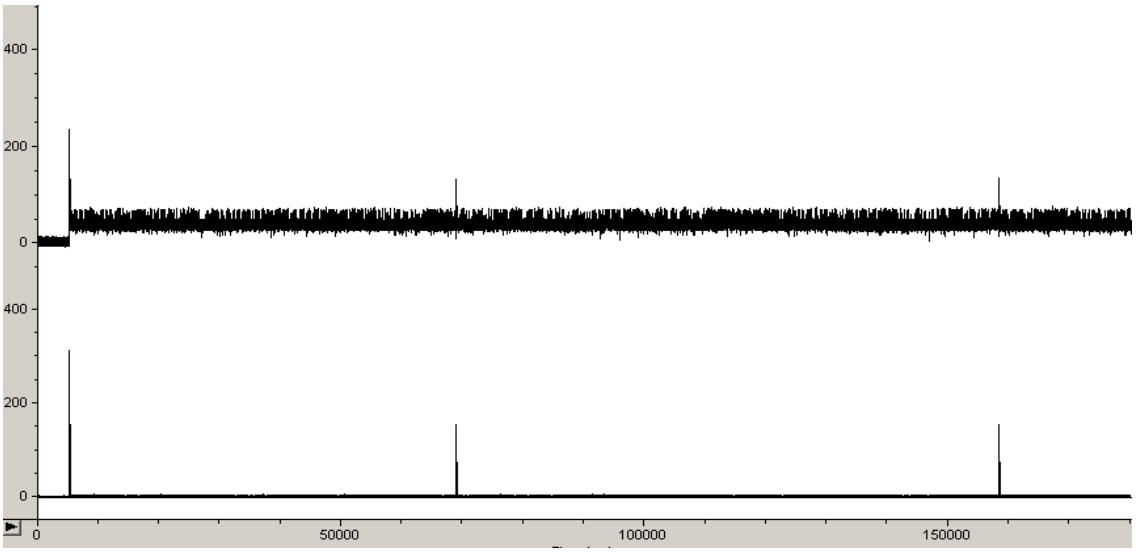


Figure 3.4: Entire original signal with multiple blockades and the Differential RC processed signal.

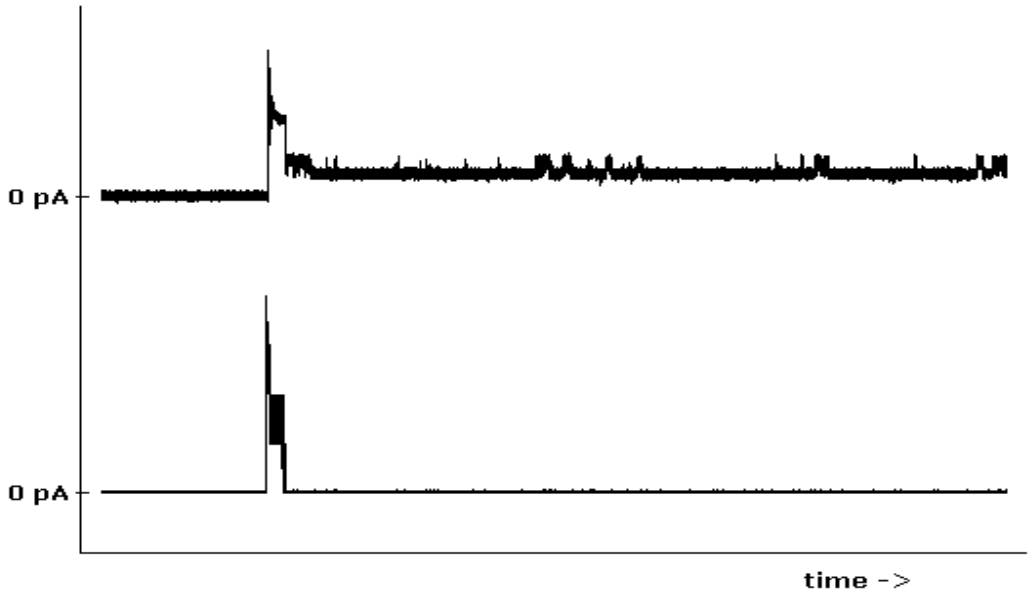


Figure 3.5: Original signal and the Differential RC processed signal.

3.2.2B Detecting Transitions

The transitions were detected using moving comparisons. A window of size 5 is moved along time. The integral of absolute difference between the window points against the approx. UL and LL means is calculated. A comparison is made to check if there is a transition from one level to another. For example, when the actual level is LL, this sum will be lesser than the differential sum against the UL. Thus accurate transitions can be recorded.

```
while (index <= length) {

conv1 = abs(databuff[0] - LL_AVG) + abs(databuff[1] - LL_AVG) +
        abs(databuff[2] - LL_AVG) + abs(databuff[3] - LL_AVG) +
        abs(databuff[4] - LL_AVG);

conv2 = abs(databuff[0] - UL_AVG) + abs(databuff[1] - UL_AVG) +
        abs(databuff[2] - UL_AVG) + abs(databuff[3] - UL_AVG) +
        abs(databuff[4] - UL_AVG);

if(state == 0){     //Currently in LL
    if(conv1 > conv2){
    state = 1;
    whichLevel = 0;
    if(averageLevel > lower_range_IL && averageLevel < upper_range_IL){
        whichLevel = 2;     //Detects Level as IL
    }

    }
}else{     //Currently in UL
    if(conv2 > conv1){
    state = 0;
    }
}

fread(&d,2,1,datafile);
rescale = (double) d/scale;
index++;

databuff[i] = rescale;
i = (i + 1) % 5;

}
```

Figure 3.6: Source code for detecting level transitions.

The advantage of this algorithm is its pretty accurate, fast and has a very little overhead when processing huge number of samples. When a transition is detected, further processing is done on the level, for example, recording mean, standard deviation, start and end of the level, duration, checking noise level and call to the spike detector to record the number of spikes etc.

3.3 SPIKE DETECTOR

Spike is defined as sudden transient variation in voltage or current. Theoretically a spike is equivalent to a Gaussian with zero width and infinite height. But in practice a signal spike is impinge of short duration and finite intensity. In Channel Current signal the spikes of interest are from the Lower Level which spike downwards.

3.3.1 Noise Removal

The introduction to Channel Current noise has been given under section 2.2. The statistical methods dealt here to detect spikes are little sensitive to noise as it works with the standard deviation. Hence noise needs to be eliminated i.e. detect if the level is noisy and reject it.

PSEUDOCODE:

1. CALCULATE LEVEL STD. DEVIATION σ
2. IF σ > 6 THEN REJECT LEVEL
3. FOR ALL SAMPLES
   CALCULATE (SAMPLE INTENSITY – LEVEL MEAN) = δ
   IF δ < 6 THEN INCREAMENT DISTRIBUTION_COUNT
4. DISTRIBUTION_PROBABILITY = DISTRIBUTION_COUNT / N
5. IF DISTRIBUTION_PROBABILITY < 0.6826 THEN REJECT LEVEL
6. DETECT AND CALCULATED SPIKES

We start by calculating the standard deviation of the level and the probability of number of samples that falls within the prototype deviation σ′ of 6 i.e. calculate the absolute difference of a sample point and the level average. This probability is an approx. measure that the signal level hovers with a standard deviation of 6. According to the Confidence Interval of the Standard Deviation, 68.26% of the values lie within the range of σ. We then check if the distribution for σ is greater than 68.26%. If not the level is rejected as noise. The actual standard deviation is set to fall within the range of 6 else the level is again rejected as noise.

3.3.2 Level Deviation Method

This method makes use of the standard deviation of the level to evaluate spikes. As you see from Figure 3.7, the signal has lots of spikes but what is the definition of a spike in the context of Channel Current? We need to evaluate the true count! The Figure 3.7 shows a steady LL with a standard deviation of 5.7. But the transition from one sample to another is so unsteady we may actually detect a false spike. Thus we need to determine the true spike which excludes the false classifications.

3.3.2A Algorithm

To calculate the true spike count, we calculate the spike count for different intensities of spike i.e. from standard deviation + 0 to standard deviation + 30. As the intensity lowers it picks up Gaussian noise. Thus using extrapolation we determine the true spike count of the Lower Level. The algorithm is as follows:

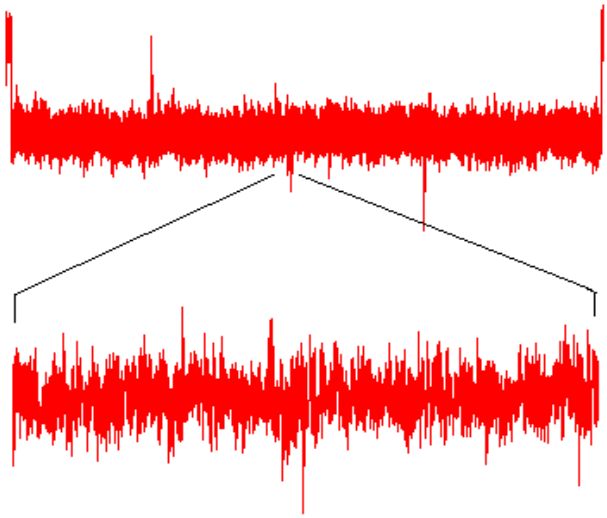


Figure 3.7: A Lower Level at top and its expanded segment at bottom.

For each increase in spike intensity, first we calculate the spike intensity cutoff as Level Mean – Standard Deviation – Spike Intensity ($\mu - \delta - k$). A moving window of size 3 is scanned sequentially and at every time t, we evaluate if the window is a spike.

PSEUDOCODE:

1. CALCULATE SPIKE INTENSITY, CUTOFF = (μ - δ – k)
2. LOAD THE SAMPLE DATAS INTO WINDOW
3. IF DATA [1] < CUTOFF AND DATA [1] > -20 NEXT STEP ELSE REJECT AND GOTO STEP 2
4. IF NOT DATA [1] < DATA [0] AND DATA [1] < DATA [2] GOTO STEP 2
5. COUNT AS SPIKE AND INCREAMENT SPIKE COUNT
6. GOTO STEP 2 AND ITERATE TILL END OF LEVEL.

This algorithm will result in the spike counts at different intensities. Then we need to extrapolate this data to produce the true spike count.

3.3.2B Extrapolation to True Spike Count

We observe the spike count distribution produces an exponential distribution. This is shown in Figure 3.8. The same kind of distribution pattern followed for tests on many different files. From the above distribution at lower intensities an interesting linear pattern is observed. This then transforms to an increasing exponential as intensity decreases which is expected as it picks up the Gaussian noise. We determine the true spike on linear extrapolation to the linear region. This is done using the Least Squares Algorithm for Linear Fit.

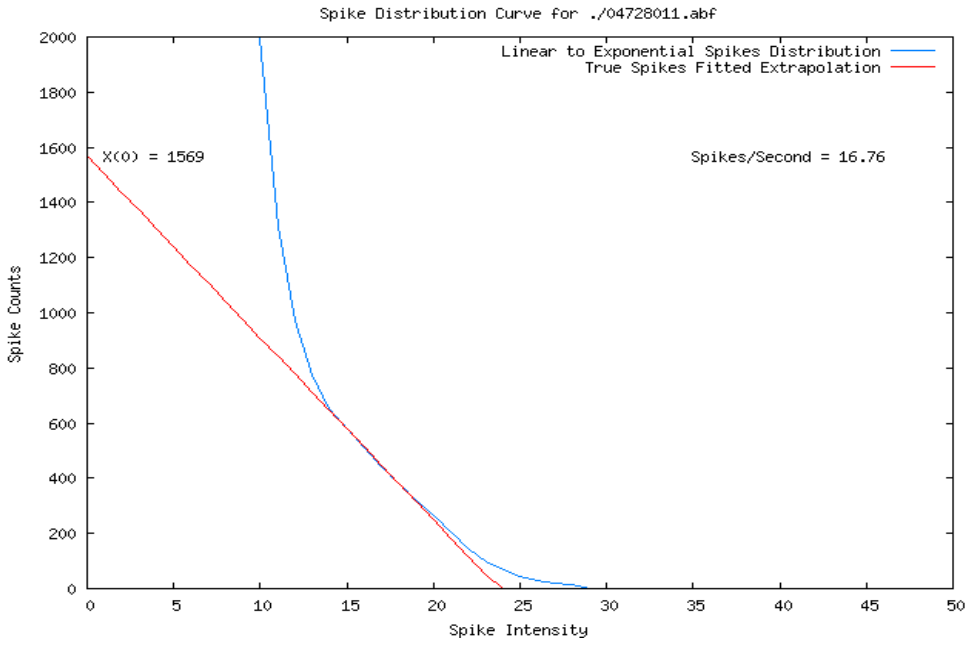


Figure 3.8: Spike counts at different spike intensities.

3.3.2C Extrapolation Fitting

The Least Squares Method algorithm was programmed to fit the linear region of the form y = mx + b. But to determine the linear region a comparison on the correlation coefficient *r* was performed. This can be done evaluating a fit for points decreasing from the top. The correlation coefficient *r* would rapidly improve and at once stage becomes steady. This is the linear region and the fitting produces the slope and intercept.

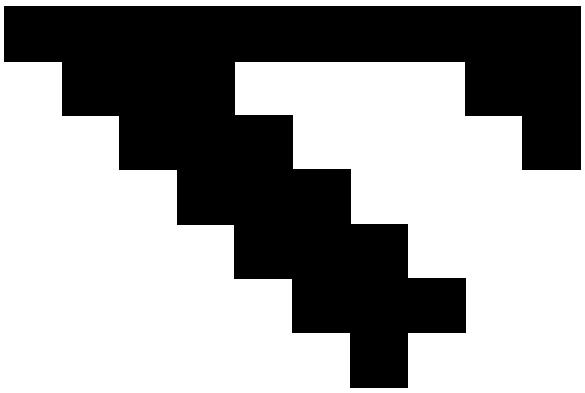
Once we determine the slope m and intercept b for points say n to m (0…n….m…), we calculate the value of y at zero i.e. m*n + b gives the best coordinate intercept. This value of y is referred as the true spike count. This method has a very little overhead and determines the output very quickly in about 10 seconds for a signal of 180 seconds duration.

Algorithm for Linear Fit:

The best fit line associated with the n points (x_1, y_1), (x_2, y_2) …. (x_n, y_n) has the form

$y = mx + b$

Where,



3.3.3 Gaussian Method

Theoretically a spike is equivalent to a Gaussian with zero width and infinite height. Practically a spike can be equated to a Gaussian of finite height and width. This rule is applied to detect the spikes. This method results in the same pattern of linear-exponential distribution as in Level Deviation Method.

3.3.3A Algorithm

In this algorithm, a Gaussian curve is generated for a defined window width. Its depth is accordingly calculated using the Gaussian distribution. A moving window of size 3 or 5 is chosen and the Gaussian is overlaid or convoluted to verify if it satisfies the fit. A cutoff of 75-80% overlay fitting is considered to be a spike fit to Gaussian. These are shown in Figure 3.9, window A is a perfect spike fit, and window B is also a fit. A traditional method would either reject the window or classify it a 2 different spikes. Window C is another case of good fit. Sometimes a traditional method might reject it depending on the window size but, the Gaussian does a fitting rather than comparing threshold values. Hence, better classification of spikes.

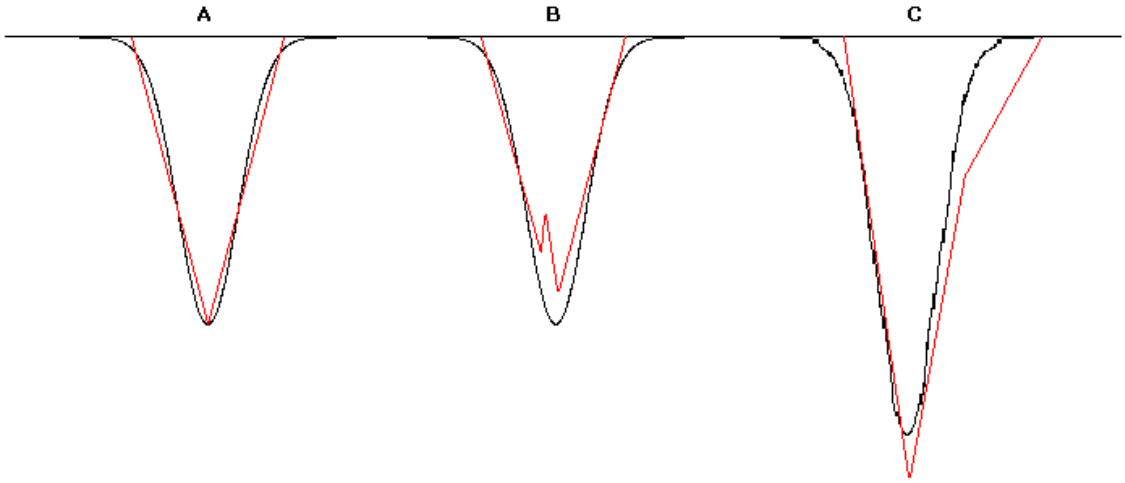


Figure 3.9: Gaussian fits to the spikes of different intensities and variations.

For cases of rejection, the fit might be unsteady with less than 75% fit. Figure 3.10 explains it below.

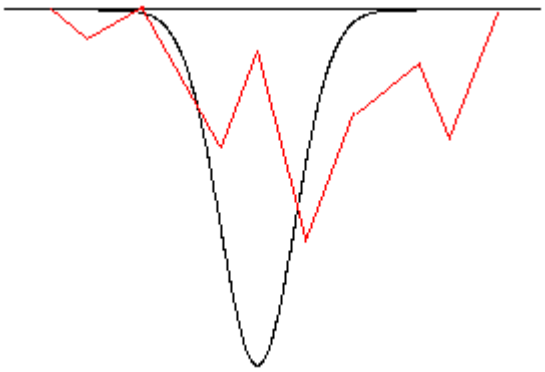


Figure 3.10: A Gaussian fit which rejects the sample window

This Gaussian fitting is done for increasing spike intensities as in the Level Deviation Method and the pseudo code changes only for step 2-4. Then the distribution is extrapolated as in section 3.3.2B using the Least Square Fitting mentioned in section 3.3.2C.

3.3.3B Comparison

A comparison between the two algorithms shows that the Gaussian method has far lesser number of false alarms i.e. it is resistant to the unsteady, varying, noisy signal than the Level Deviation method. Figure 3.11 displays this comparison.

Figure 3.11: A False Spike accepted by Level Deviation and rejected by Gaussian method.

In this figure, the left hand side displays the acceptance by level deviation method. It just depends on the Spike Intensity Cutoff calculated from the overall level mean whereas the Gaussian method fits a curve from the average of the window base. Figure 3.12 shows an example of 3 back-to-back spikes. The table below shows comparison between spikes predicted between the above two methods.

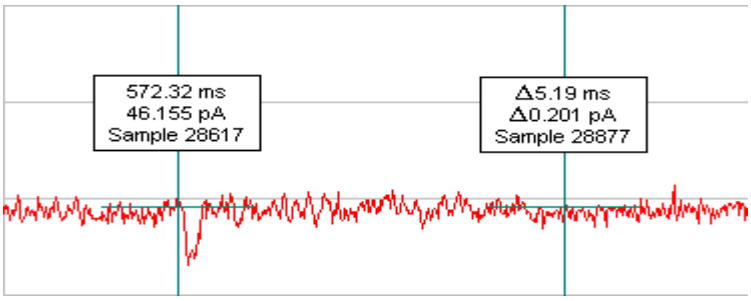


Figure 3.12: Three close back-to-back spikes near sample 28617.

The Level Deviation Method identifies all 3 back-to-back spikes at sample numbers 28623, 28628, 28631. The Gaussian Method identifies only one spike at sample number 28628.

Level Deviation Method		Gaussian Method	
Sweep/Sample#	Amplitude (pA)	Sweep/Sample#	Amplitude (pA)
3/28601	36.169434	3/28601	36.169434
3/28623	**15.954589**	**3/28628**	**19.665527**
3/28628	**19.665527**	3/28657	35.473633
3/28631	**29.425047**	3/28692	38.507080
3/28657	35.473633	3/28709	35.638428

4. RESULTS

Using the methods described above, the channel current blockade signal was cleaned and the features were extracted using the HMM-EM algorithm. These were tested for various values of sigma. The cleaned signal is shown in Figure 4.1, it can be noted that for every increase in sigma value the values shift to local maximum and hence cleaner the signal. The best feature of the method is the transition points are retained precisely. With this accuracy the FSA model was used to detect the level transitions in this signal namely Lower Level, Upper Level and Intermediate Level. The HMM-EM and FSA methods combined produced extraordinary results. The identified Lower Level segments were passed through the spike detector to detect the spikes, shown in Figure 4.2.

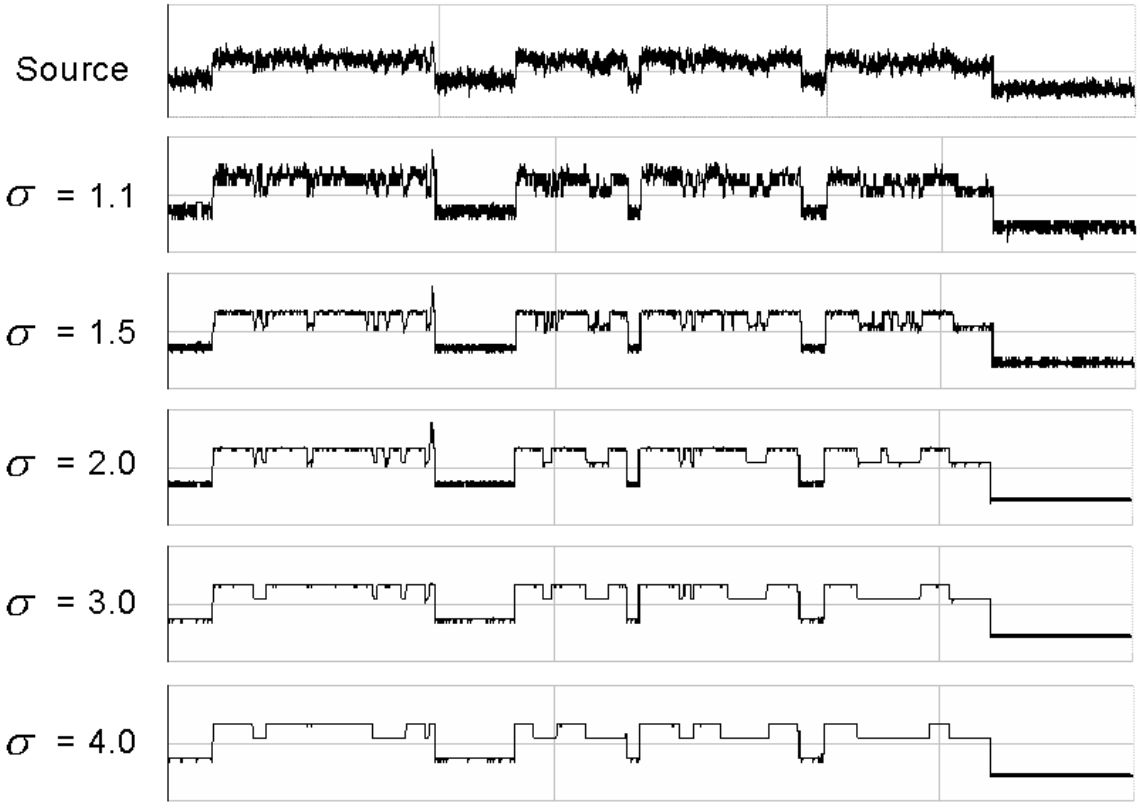


Figure 4.1: Results of HMM-EM Projection for various increase in σ. Each result was produced with 10 rounds of EM and 2 projections.

For each signal file the spikes were detected from all the Lower Levels combined and as described in method 3.3 the plot were extrapolated to get the total number of true spikes.

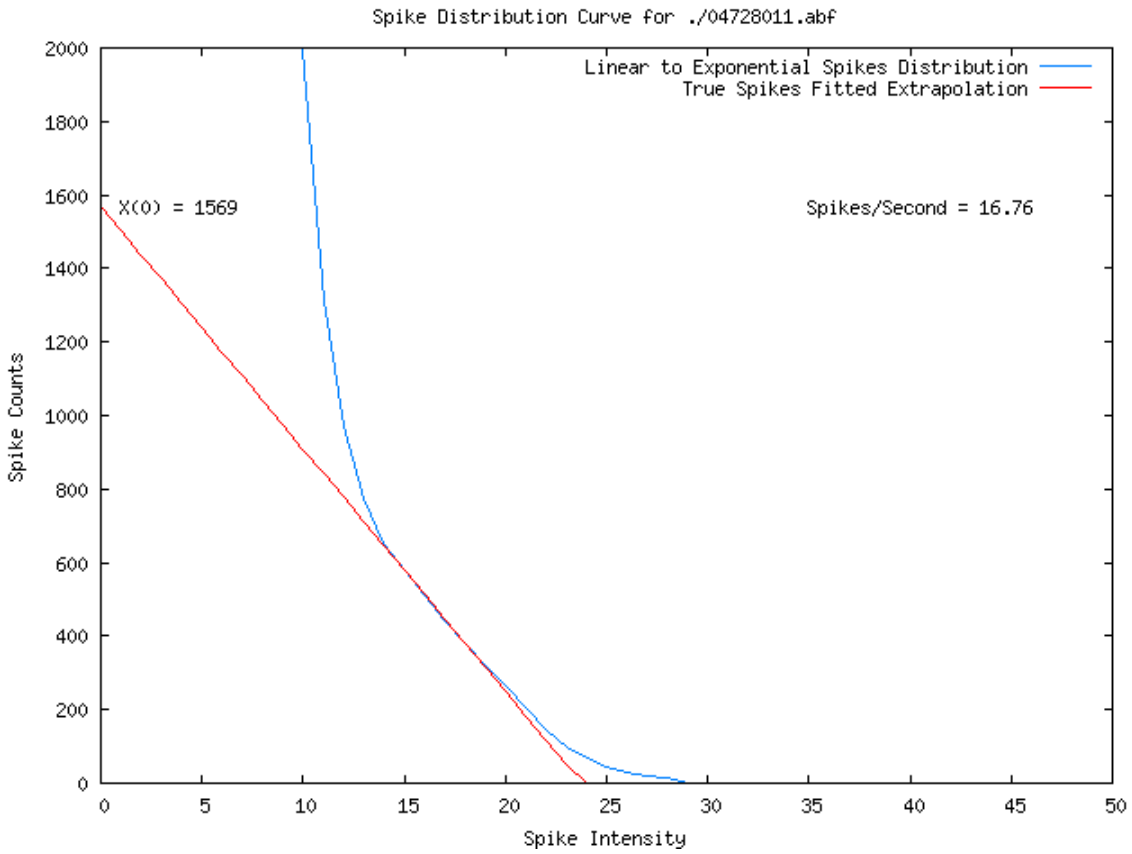


Figure 4.2: True Spike counts by linear extrapolation.

The spike counts as a feature were outstanding. The Vercoutere Dataset (Dr. Wenonah Vercoutere, NASA) has DNA that were radiated and non-radiated were tested. Spike Detector produced results that were distinguishable between radiated and non-radiated signals. The radiated set contained 17 spikes/second whereas the non-radiated set contained 9 spikes/second. These are shown in Figure 4.3a and 4.3b.

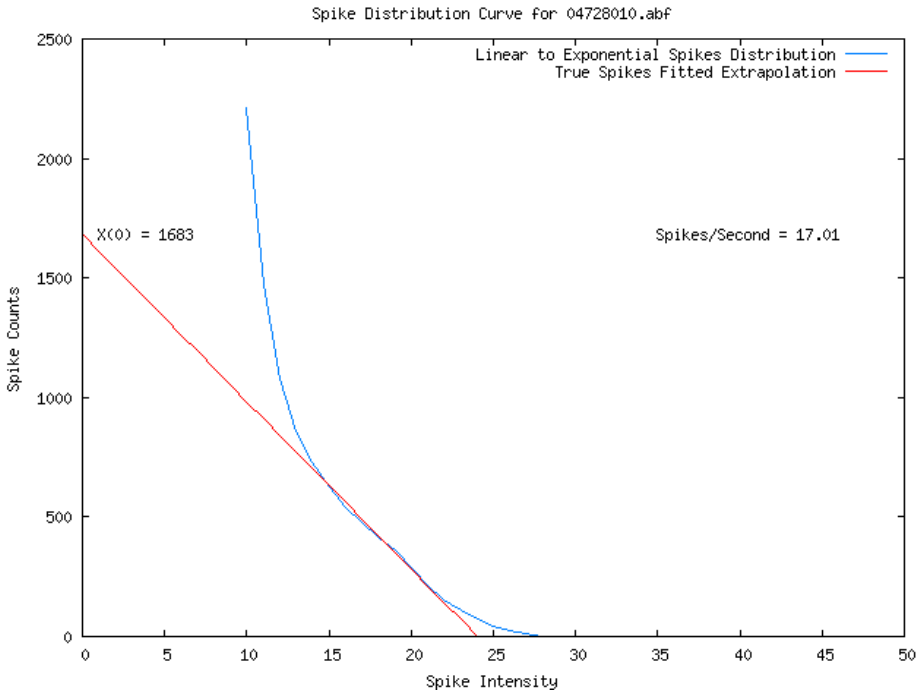


Figure 4.3a: True Spike counts for Vercoutere Radiated BP.

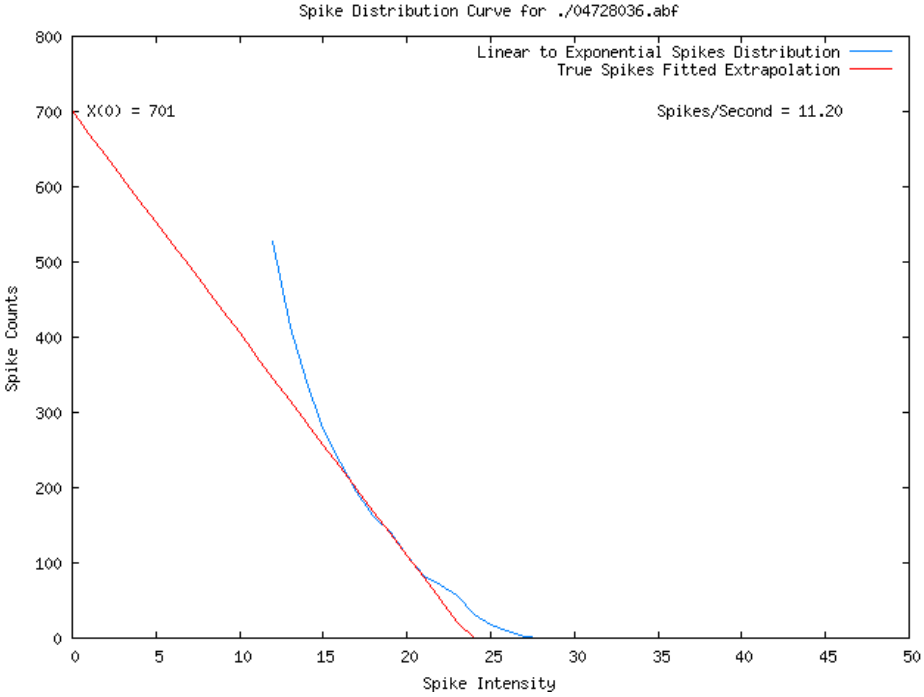


Figure 4.3b: True Spike counts for Vercoutere Non-Radiated BP.

5. CONCLUSION

Various types of Channel Current datasets namely radiated, non-radiated, temperature and voltage varied were tested, cleaned and features extracted. Examination of these results showed accurate cleaning and retaining of transitions by the HMM-EM algorithm. The FSA was equally accurate in level detection. The spike detector provided valuable information on rapid toggling of molecule which produces spikes. These were analyzed from the Vercoutere Dataset. Thus, the HMM-EM and FSA methods combined have produced extraordinary results. These features are now being used for training and classification of molecules by the SVM's.

REFERENCES

1. Vercoutere W., S. Winters-Hilt, H. Olsen, D. Deamer, D. Haussler, and M. Akeson. “Rapid discrimination among individual DNA hairpin molecules at single-nucleotide resolution using an ion channel”

2. Tuzlukov, V. P. “Signal Processing Noise”

3. Vaseghi, Saeed V. "Advanced digital signal processing and noise reduction"

4. Stephen Winters-Hilt, Anand Prabhakaran. “Channel Current Kinetic Analysis using FSAs and HMMs”, in preparation for BMC Bioinformatics.

5. Lawrence R. Rabiner. “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”.

6. Arthur Dempster, Nan Laird, and Donald Rubin. "Maximum likelihood from incomplete data via the EM algorithm". Journal of the Royal Statistical Society, Series B, 39(1):1–38, 1977.

7. Stephen Winters-Hilt, Wenonah Vercoutere, Veronica S. DeGuzman, David Deamer, Mark Akeson and David Haussler**.** “Highly Accurate Classification of Watson-Crick Basepairs on Termini of Single DNA Molecules”

VITA

Anand Prabhakaran was born on November 1982 in Madras, India. It was formerly called Madras now known as Chennai. He completed his high school from the acclaimed D.A.V Higher Secondary School, Madras. His final common examinations score stood in the top 100 in the state. Later in April 2003 he received his bachelor's degree in Electronics and Communications Engineering from University of Madras, India. His area of research during bachelors was Embedded Systems and Security. His bachelor's dissertation was Mobile communication using PN Sequence – 'A Spread Spectrum Technology'. He believed an amalgam of Electronics and Computer Science was much more efficient in a technical career. Later he pursued his master's studies. He graduated from University of New Orleans in December 2005 with a Masters degree in Computer Science.