MODELING OF RECURRENT THRESHOLD CROSSINGS

DUE TO NOISE WITH LONG MEMORY

A Thesis

by

ABHISHEK NARAYAN SINGH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2005

Major Subject: Electrical Engineering

MODELING OF RECURRENT THRESHOLD CROSSINGS

DUE TO NOISE WITH LONG MEMORY

A Thesis

by

ABHISHEK NARAYAN SINGH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee:	Laszlo B. Kish
Committee Members:	Don R. Halverson
	C.B. Su
	William H. Marlow
Head of Department:	C. Georghiades

December 2005

Major Subject: Electrical Engineering

ABSTRACT

Modeling of Recurrent Threshold Crossings due to Noise

with Long Memory. (December 2005)

Abhishek Narayan Singh, B.S., The University of Texas at Arlington

Chair of Advisory Committee: Dr. Laszlo B. Kish

This thesis addresses the recurrent threshold crossing behavior of long-time correlated noise. The behavior of long-time correlated noise like $1/f$, $1/f^{1.5}$, and $1/f^{2}$ can be associated with the behavior of many phenomena in nature, so it is of interest to study the behavior of this noise. Our method of modeling their recurring behavior relies on setting a particular threshold level for a particular level of noise and observing how frequently the noise crosses the threshold level. We also add a periodic drive to the noise which enables it to cross the threshold level easily when it is at peak, and vice versa. This technique provides a model for the changing seasons that occur during every year. We also compare the recurrence behavior of threshold crossings from our computer simulations with theoretical results from the Rice formula. We have related the recurrence of these threshold crossings with the recurrence of natural disasters. Therefore we are providing a model to predict the recurrence of a natural disaster once that disaster has previously occurred. From our results, we conclude that once a natural disaster has occurred, there is a high probability of its recurrence in a short time, and this probability gradually decreases with time.

DEDICATION

To my parents and brother, for making this possible.

ACKNOWLEDGEMENTS

There are many people I would like to thank who have helped me achieve great success in my career thus far. I would like to express my deep gratitude and respect to Dr. Laszlo B. Kish for his guidance and all the advice and continuous encouragement that he gave me throughout this project. I would also like to thank Dr. C.B. Su, Dr. W.H. Marlow and Dr. D.R. Halverson, the members of my thesis committee, for their time and inputs. A vote of thanks is also due to Ms. Tammy Carda and Ms. Jeanie Marshall who constantly helped me during this time at Texas A&M in innumerable ways. I will always be indebted to my family for their encouragement and support.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

A. Recurrent Events in Nature

Natural hazards are assuming ever greater economic importance, not only on a regional but also on a global scale. The growth of major cities in hazard prone areas, and the public anxiety associated with risks to critical facilities such as nuclear reactors [1], has focused attention on the problems of insurance against natural hazards, disaster mitigation, and disaster prevention.

In the pre-scientific age of the world, very natural things like earthquakes, volcanic eruptions, tidal waves, typhoons and devastating floods (not forgetting Noah's ark and the flood), did take place. Man on earth here explained them in his own simple way, elevating these phenomena at times to the level of divine activities or heavenly forces. Elsewhere, such elemental violence or misbehavior like hailstorms, directly descending from the skies, were explained as expressions of divine wrath or heaven's vengeance on man for his sinful behavior on earth. They were looked upon as acts of punishment sent down from above, to which man had to helplessly succumb. Whatever be the explanation man on earth gives to these phenomena which the ancients reckoned as heaven sent, they are known to everyone today as frequent events in the world we live, taking place with fair regularity.

This thesis follows the style and format of *Fluctuation and Noise Letters.*

It has been observed that many of the natural disasters are recurrent in nature, that is, they occur more than once over a span of time [2]. We all have seen the recurrence of tsunamis in Asia killing thousands of people. The recurrence of hurricanes, tornados, and floods are just a few examples of natural disasters that occur more than once over a period of time. Humans have not yet been able to develop any technology than can prevent these disasters; all we can do is to predict their happening just to better prepare ourselves to face them and thus minimize the damage caused and loss of lives. Many theories have been developed to provide a model for natural disaster prediction, however no theory can be hundred percent accurate in practice. In this thesis I have proposed a theory about the recurrence behavior of natural disasters based on the recurrence behavior of rare threshold crossings of long-time correlated noise. Natural Disasters like the Tsunami, Earthquakes, Drought, Famine, Flood, etc. are generated by a stochastic process crossing a certain threshold. We have related the occurrence of a natural disaster with the occurrence of rare threshold crossing of a long-time correlated noise. It has been observed that if the noise has a long correlation time, after a threshold crossing, there are several more threshold crossings in a short time. In other words, these threshold crossings tend to come in clusters which tell us that once the threshold is crossed, there is a higher probability of another crossing. The frequency of these threshold crossings decreases significantly as more time goes on. This implies that after one threshold crossing, there is an increased probability of a subsequent threshold crossing; this directly relates to the proposition that once a natural disaster has occurred, there is an increased probability of its recurrence in a short time. Consider an example of a dam; if the level of water in a dam starts to rise for some reason, the dam will be able to hold the pressure only till a certain level, so called the threshold level. Once the water level crosses that threshold, there is a great chance that the dam wall will break.

B. Recurrent False Alarms

Almost every alarm system commercially available suffers from a problem of False alarms. False alarm means the abrupt activation of an alarm without any reason. One of the most important considerations when viewing an alarm system is its False Alarm Rate [3]. False Alarm Rate can be defined as the probability of the alarm system becoming activated when actually it should not. Besides external factors, there is no actual reasoning for the occurrence of False alarms. We believe that like Natural Disasters, false alarms are also caused by stochastic processes crossing a certain threshold. As we shall see in Chapter III, stochastic processes lead to threshold crossings that come in clusters. Since we are relating every threshold crossing with the occurrence of a false alarm, we shall see that if a false alarm has occurred once, there is an increased probability of its recurrence in a short time. Also the probability of false alarm recurrence will continuously decrease as more time passes; we shall view this in detail in chapter III.

C. Why Study the Behavior of Noise with Long Memory

Noise in physical systems is found to occur with various spectral forms. The power spectrum of fluctuations scales with frequency as $S(f) = 1/f^{\alpha}$ in a large variety of physical, chemical, and biological systems. This power law behavior $1/f^{\alpha}$ often persists over several orders of magnitude with cutoffs present at both high and low frequencies, and with typical values of α in the range $0.8 \le \alpha \le 2$ [4]. In a somewhat loose terminology, all these systems are said to display $1/f$ noise although good quality data with α very close to 1 exist only for the voltage fluctuations when a current is flowing through a resistor. Phenomena with $1 < \alpha \le 2$ however, are abundant, a few examples include the occurrence of natural disasters like earthquakes, floods, volcanic eruptions, tsunamis, hurricanes, etc., white-dwarf light emissions, the flow of sand through hourglass, ionic current fluctuations in membrane channels, number of daily trades in the stock market, water flows of rivers, the spike train of nerve cells, the traffic flow on the highway, the electric noise in carbon nanotubes and in nanoparticle films, the interference fluctuations in wireless communication systems [5], and many more.

The $1/f$ spectrum of noise has often been claimed to be some form of fundamental process which is very basic to the laws of nature because it is found in so many systems. The basic property of $1/f$ noise is that there is the same intensity in any decade of frequency compared with white noise where there is the same intensity per Hz bandwidth.

The purpose of this study is to get some insight into the rare threshold crossing events of long-time correlated noise. Since we are claiming that the recurrent threshold crossing behavior of noise with long memory can be related to the recurrence behavior of many phenomena in nature - more specifically natural disasters, this study will help us provide a model for predicting the recurrence behavior of natural disasters based on the recurrent threshold crossing behavior of long-time correlated noise. Since climatic changes are periodic in nature and these climatic changes in some ways contribute to the occurrence of natural disasters, we will add a periodic drive to our long-time correlated noise. This periodic drive is a sinusoidal signal added to the noise and will affect the overall behavior of the threshold crossings of noise. We will study how the behavior of threshold crossings will change once we introduce the sine wave. The idea behind introducing the sine wave will be to provide a model for the changing seasons like summer, winter, rainy and dry seasons and analyze any observed changes in behavior. This process is illustrated in the figures 1 through 6 below for $1/f$, $1/f^{1.5}$ and $1/f^{2}$ noise.

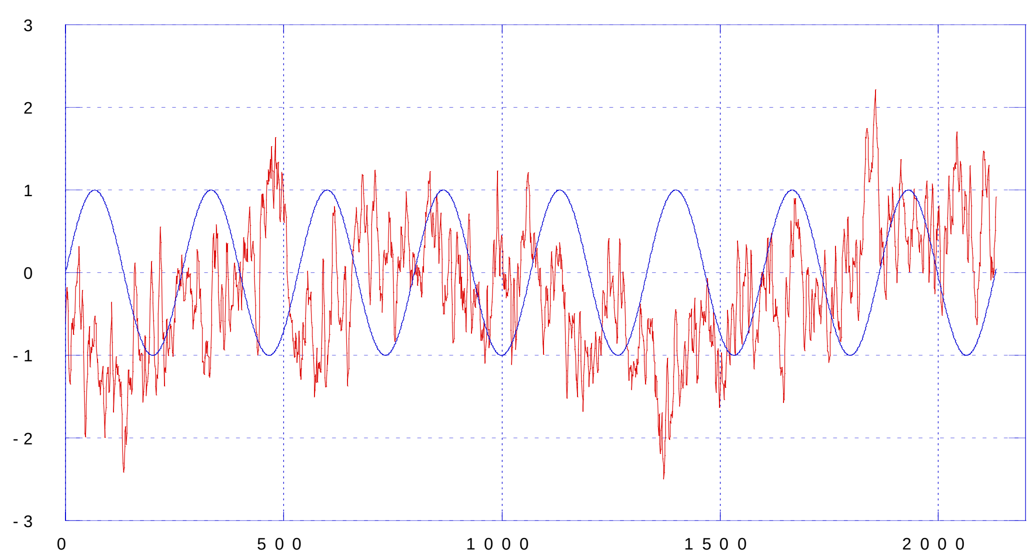

Fig. 1. $1/f$ Noise with Periodic Drive

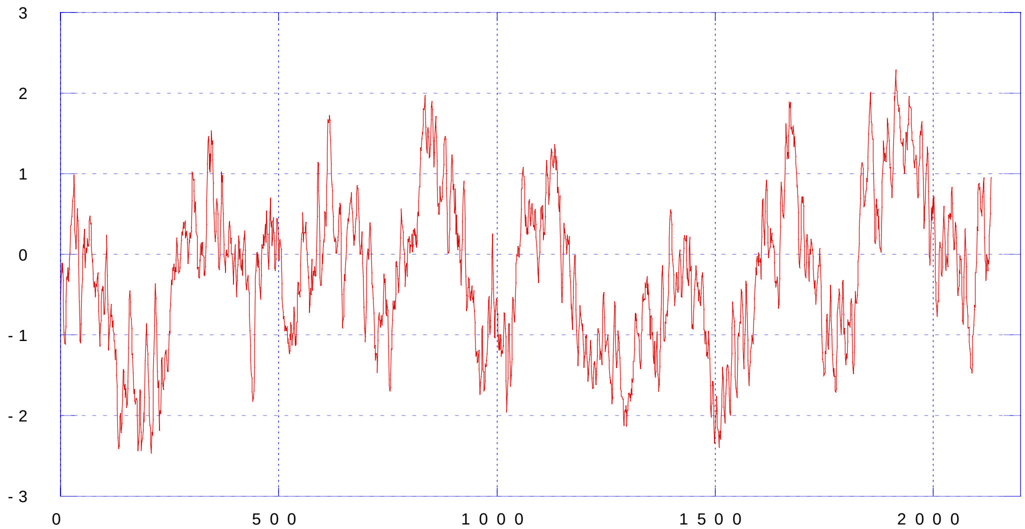

Fig. 2. Additive of $1/f$ Noise and Periodic Drive

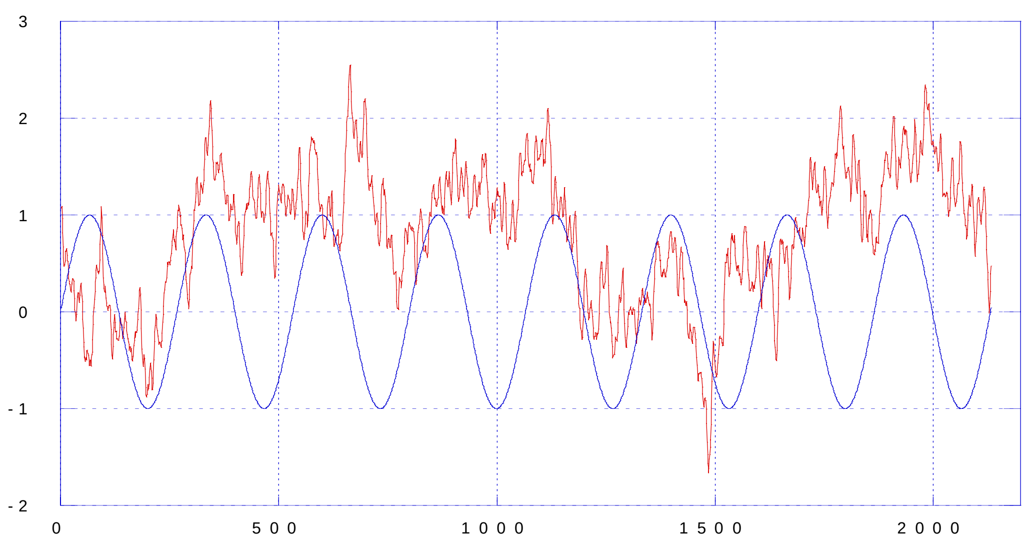

Fig. 3. $1/f^{1.5}$ Noise with Periodic Drive

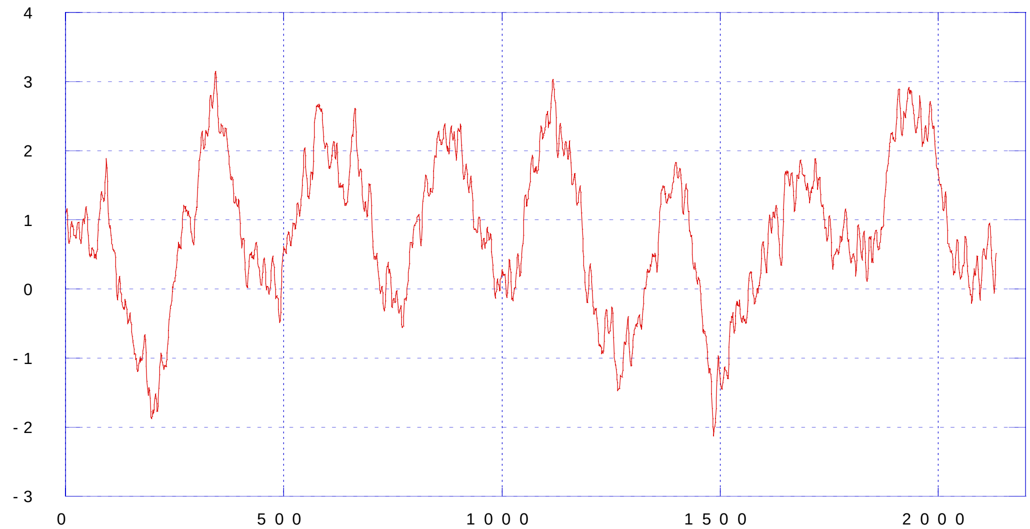

Fig. 4. Additive of $1/f^{1.5}$ Noise and Periodic Drive

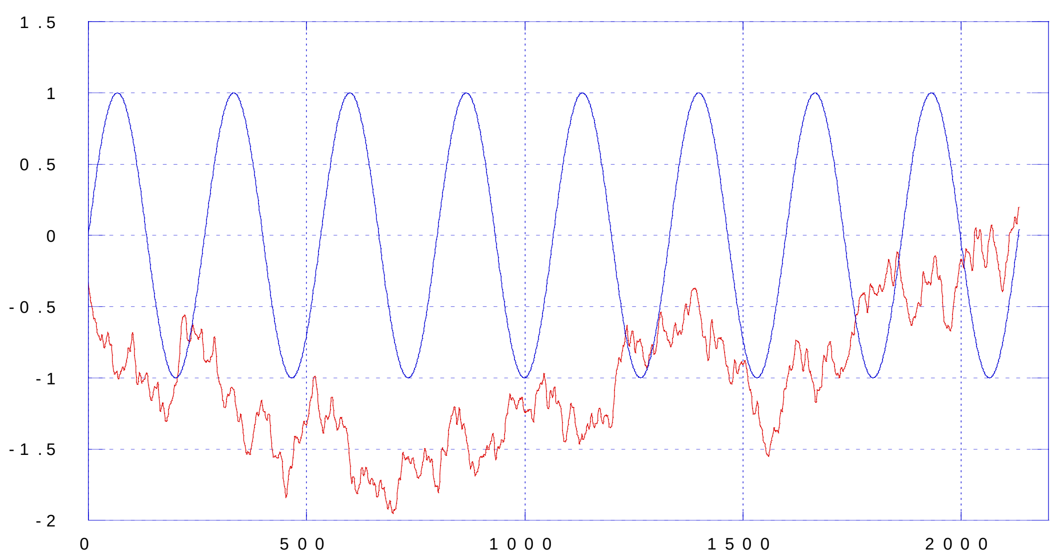

Fig. 5. $1/f^{2}$ Noise with Periodic Drive

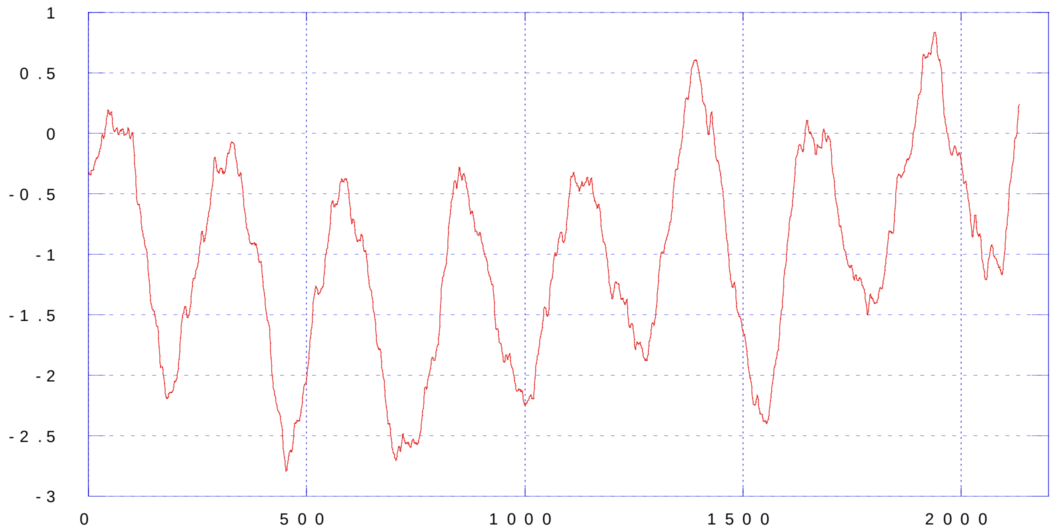

Fig. 6. Additive of $1/f^2$ Noise and Periodic Drive

D. Background: Stochastic Resonance Phenomena

Many scientists have been concerned lately with Stochastic Resonance (SR), a paradoxal phenomenon in which an optimal noise intensity maximizes the information transfer through a threshold device [6]. There is a particular amount of "spoiling" of the signal by an additive Gaussian noise which optimizes the SNR at the output of a stochastic resonator. Using a simple Level Crossing Detector (LCD) which is a stochastic resonator without internal dynamics, we are able to achieve a maximum at the output of the LCD for a particular level of input noise. The LCD works in such a way that a short uniform spike is initiated at its output whenever the voltage at its input goes through a given voltage level, the threshold voltage, in the increasing direction. Feeding an LCD by the sum of a periodic signal and a Gaussian noise of variable strength yields a sharp maximum of the output SNR at a particular strength of the input noise. A very important observation is that the location of the maximum of SNR and generally the whole SNR curve is independent of the signal frequency in wide range of frequency which proves that the level crossing dynamics of that noisy signal inherently contains the SR effect and the stochastic resonator, the LCD, is not causing but only detecting the SR phenomenon. The illustration of stochastic resonance phenomenon is illustrated in figure 7 below:

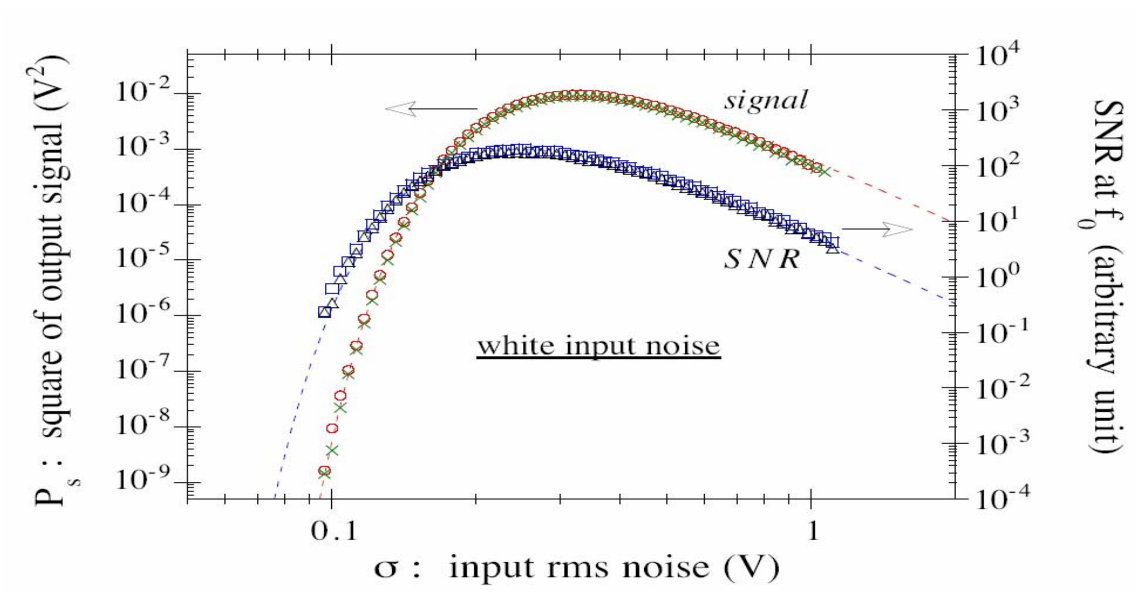

Fig. 7. Illustration of Stochastic Resonance Phenomenon

E. Description of the Thesis

In order to model the recurrent threshold crossings of long-time correlated noise, we will analyze the behavior of $1/f$, $1/f^{1.5}$, and $1/f^2$ noise; these noise are long-time correlated and thus said to have “long memory”. To explore their behavior we will define three quantities: Noise Root-Mean-Square (rms) value, Threshold level, and Window size. We will observe how often the noise crosses a threshold value that we ourselves set. We will then study these threshold crossings of noise for definite time lengths, called window-sizes. Each window size would represent one time unit. For example we may decide to examine the threshold crossings of $1/f$ noise with an rms value of 0.1 using a threshold value of 0.8 and window size 100. This is shown in figure 8 below:

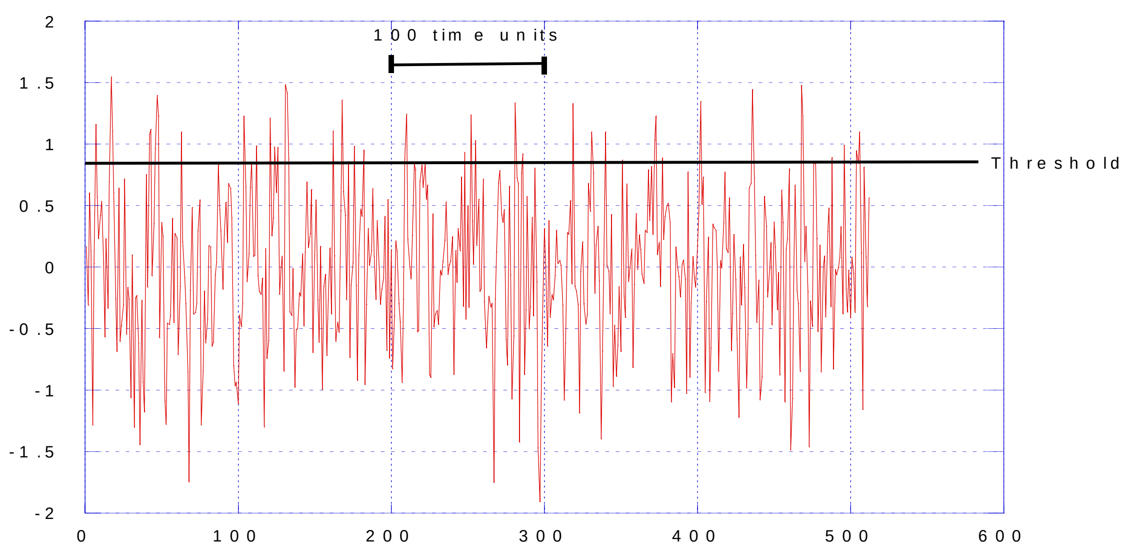

Fig. 8. Illustration of Threshold Crossing Phenomena for $1/f$ noise

Looking more closely, we will observe the fluctuation of noise and check when it crosses the threshold level. Starting at that point, we will examine the number of times the noise will cross the threshold level during a definite time period specified by the window size. The number of threshold crossings in the fixed time window will give us some idea about the behavior and random fluctuation of the noise. We will then study the change in the number of threshold crossings as we vary our window size. In other words, we will observe how frequently the noise is crossing the threshold level as time goes on. We will define the probability of threshold crossing as the number of actual threshold crossings in a given window size divided by the window size. We will also add a periodic drive to the noise which will provide a model for the changing seasons. This periodic drive is just a sinusoidal signal of fixed amplitude added to the noise. The idea behind adding the sinusoid to the noise is that when the sine wave will peak, it will constructively add to the noise and as a result increase the probability of the noise in crossing the threshold level. Likewise when the sinusoid will be at a minimum, it will subtract from the noise and decrease the noise level resulting in lower probability of the noise in crossing the threshold. To understand the practical significance of this experiment, we will first have to re-iterate the reason for studying the behavior of this long-time correlated noise. As already stated in section C above, the recurrence of rare threshold crossings of $1/f$, $1/f^{1.5}$, and $1/f^{2}$ noise is similar to the recurrence of many phenomena in nature, more specifically natural disasters [7]. This is because both of these behaviors are caused by random stochastic processes crossing a certain threshold. The introduction of the periodic drive increases or decreases the probability of the threshold crossing. Practically this can be understood as different seasons during a year affecting the frequency and intensity of natural disasters. For example, compared to any other time of the year, there is a greater chance of flood during the rainy season. Similarly, there is a greater chance of drought during the dry/summer season, and so on. More generally a periodic drive can be thought of as a model for the different external factors contributing to the recurrence of any natural disaster.

For this thesis, I have computed the number of threshold crossings of $1/f$, $1/f^{1.5}$, and $1/f^{2}$ noise for different time lengths - ranging from as little as 10^{1}, to as large as 10^{4}. I took 100 recordings for the number of threshold crossings in each of the window sizes and computed their average and shown them on a graph (in chapter III).

CHAPTER II

MATHEMATICAL CONSIDERATIONS

A. Rice Theory of Threshold Crossings for Recurrent Events

The distribution of threshold crossings has no exact theoretical solution. However there is a theory for the mean frequency of crossings. Rice determined the mean frequency ν_0 of zero-crossings of the amplitude of Gaussian noise. In order to obtain the mean frequency $\nu(U_t)$ of crossings of arbitrary levels U_t we derive the Rice-formula in a different and simpler way which yields the required generalization [8]. We are concerned with physical noise which means that the amplitude $y(t)$ of the Gaussian noise and its velocity dy/dt have finite root-mean-square (rms) values σ and Δ, respectively. Thus, their power spectrum have cut-offs (at least from high frequencies).

First, it is shown that the amplitude $y(t)$ of the Gaussian noise is statistically independent of its velocity dy/dt. According to the random-phase-oscillators representation of Gaussian process, the amplitude of a Gaussian noise can be written as:

$$y(t) = \sum_{n=1}^{\infty} a_n \cdot \sin(2\pi \cdot f_n t + \phi_n) \tag{1}$$

where a_n, f_n and ϕ_n are the amplitude, the frequency and the random initial phase of the n^{th} oscillator, respectively. The oscillator frequency is given as $f_n = n \cdot \delta f$ where

δf is infinitesimally small. The random set of ϕ_n (uniformly distributed in the range of $(0,2\pi]$) is different for each representation of the given noise process. From equation 2.1, the velocity can be written as:

$$\frac{dy(t)}{dt} = \sum_{n=1}^{\infty} 2\pi \cdot f_n \cdot a_n \cdot \cos(2\pi \cdot f_n t + \phi_n) \qquad (2)$$

It is easy to see that the cross-correlation function of equations 2.1 and 2.2 is zero due to the orthogonality of sine and cosine functions with the same frequency, and due to the orthogonality of sine waveoidal functions with different frequencies:

$$\left\langle y(t)\, dy(t)/dt \right\rangle_t = 0 \qquad (3)$$

where the index t represents time-averaging (as $y(t)$ and dy/dt are stationary and ergodic processes). Using the well known fact that a zero-correlation of two different Gaussian processes implies their statistical independence, we can conclude that the instantaneous amplitude and the instantaneous velocity of a Gaussian noise are statistically independent from each other.

In the next step, we determine the functional form of the mean frequency $\nu_0(U_t)$ of the crossing of level U_t by the noise amplitude $y(t)$. Let us consider the behavior of the noise in an infinitesimally narrow amplitude interval $U_t - \partial U \le y(t) \le U_t + \partial U$

around U_t. The infinitesimal smallness of the interval $2\partial U$ has two important implications: i) the amplitude distribution function of the noise will be uniform in this range; ii) for a long but finite duration of the noise, the number of those amplitude trajectories which enter into the interval and, after changing their direction, leave the interval without crossing the level U_t, is zero. Note the last implication pre-requires the frequency band limited property of the physical Gaussian noise (see above). On the other hand, the number of amplitude trajectories which enter into the interval and, without changing their direction, leave the interval via crossing the level U_t, is not zero and it is related to the mean frequency $\nu(U_t)$ of crossing this level. On the basis of these considerations the following relations can be written for the probability of finding the instantaneous amplitude $y(t)$ within the interval:

$$2 \cdot g(U_t)\partial U = \nu(U_t)\partial t \qquad (4)$$

where $g(U)$ is the amplitude distribution of the noise, that is, $g(U_t) = (2\pi)^{-1/2} \cdot \sigma^{-1} \cdot \exp[-(U_t / \sigma)^2 / 2]$ and ∂t is the mean passing time of trajectories via the interval. The right hand side of equation 2.4 represents the fraction of time which the noise amplitude spends in the interval. From equation 2.4, the level crossing frequency can be given as follows:

$$\nu(U_t) = 2 \cdot \frac{\partial U}{\partial t} (2\pi)^{-1/2} \cdot \sigma^{-1} \cdot \exp[-(U_t / \sigma)^2 / 2] \tag{5}$$

The quantity $2\partial U/\partial t$ is equal to the mean velocity of the noise in the interval. Due to the statistical independence of the velocity and the amplitude, $\partial U/\partial t$ will be independent from the location of the interval, so the value of U_t. On the other hand, $\partial U/\partial t$ is proportional to the rms velocity of the noise, consequently:

$$\nu(U_t) = c \cdot \Delta \cdot \sigma^{-1} \cdot \exp[-(U_t / \sigma)^2 / 2] \tag{6}$$

where c is a constant. With the help of the power-density spectrum $S(f)$ of the noise $y(t)$, the delta term can be replaced by the integral of $(2\pi)^2 \cdot f^2 \cdot S(f)$:

$$\nu(U_t) = C \cdot \sigma^{-1} \exp\left[\frac{-U_t^{\,2}}{2\sigma^2}\right] \cdot \left[\int_0^{\infty} f^2 \cdot S(f) \cdot df\right]^{1/2} \tag{7}$$

where $C = 4\pi^2 c$. Equations 2.6 and 2.7 describe the functional form of the dependence of ν on $S(f)$ and U_t. The determination of the constants c and C is based on the following trick: we take $U_t = 0$ as threshold (so the value of the exponent term becomes 1) and a narrow-band noise with the following spectrum: $S(f) = S_0$ for $f_0 - \partial f < f < f_0 + \partial f$ and otherwise it is zero.

Then, in the limit ∂f→0, equation 2.7 will approach the zero-crossing frequency of a sine waveoidal signal with frequency f_0, that is, $\nu = 2 \cdot f_0$. In this way we have obtained our final formula:

$$\nu(U_t) = 2 \cdot \sigma^{-1} \exp\left[\frac{-U_t^{\,2}}{2\sigma^2}\right] \cdot \left[\int_0^\infty f^2 \cdot S(f) \cdot df\right]^{1/2} \tag{8}$$

which is the generalization of the Rice-formula (derived for the frequency of zero-crossings). Indeed, in the limit of $U_t = 0$, equation 2.8 becomes identical with the formula derived by Rice.

B. Evaluation of Rice Formula for $1/f$, $1/f^{1.5}$, and $1/f^2$ noise:

From our equation (2.8), we have the generalization of Rice formula given as:

$$\nu(U_t) = 2 \cdot \sigma^{-1} \exp\left[\frac{-U_t^{\,2}}{2\sigma^2}\right] \cdot \left[\int_{f_1}^{f_2} f^2 \cdot S(f) \cdot df\right]^{1/2} \tag{9}$$

where $\nu(U_t)$ is the mean frequency of crossings of arbitrary level U_t, σ is the rms value of noise, and $S(f)$ is the power-density spectrum of the noise. We will use this formula to theoretically evaluate the mean threshold crossing frequency according to the lower and upper cutoff frequencies that are used in our computer models. We will then plot the graphs of the theoretically calculated mean threshold crossing frequencies for different window sizes. This will be done by scaling the upper cutoff frequency of the noise that which corresponds to that window size. We will then compare our computer simulation results with our theoretical results calculated using the Rice formula. The expressions for lower and upper cutoff frequencies based on window sizes are given below:

$$f_{lower} = \frac{1}{1} = 1 \qquad\qquad f_{upper} = \frac{1}{windowlength}$$

1. Rice formula for $1/f$ noise

Here we have,

$$\sigma = \sqrt{\int_{f_1}^{f_2} S(f) \cdot df} \qquad \text{where} \qquad S(f) = \frac{A}{f}, \text{ where A is a constant.}$$

Therefore after integration we get:

$$\sigma = \sqrt{\int_{f_1}^{f_2} \frac{A}{f} \cdot df} = \left[A \cdot \ln \frac{f_2}{f_1} \right]^{1/2} \tag{10}$$

In our case the lower frequency f_1 is just the inverse of the total time period (16384 steps) and the upper frequency f_2 is the inverse of one time step given as:

$$f_1 = \frac{1}{16384}, \quad \text{and} \qquad f_2 = \frac{1}{1} = 1$$

From our experiments, Noise rms, σ was calculated as 1.65 and the threshold level U_t was set to 1.82

Therefore from equation (2.10) we can calculate the value of constant A.

$$1.65 = \left[A \cdot \ln \frac{1}{1/16384} \right]^{1/2} \Rightarrow A = 0.28055$$

Now we can calculate the mean frequency of crossings using equation (2.9)

$$\nu(1.82) = 2 \cdot (1.65)^{-1} \exp\left[\frac{-(1.82)^2}{2(1.65)^2} \right] \cdot \left[\int_{1/16384}^{1} f^2 \cdot \frac{0.28055}{f} \cdot df \right]^{1/2}$$

thus, $\nu(1.82) = 0.248$

2. Rice formula for $1/f^{1.5}$ noise

Here we have,

$$\sigma = \sqrt{\int_{f_1}^{f_2} S(f) \cdot df} \qquad \text{where} \qquad S(f) = \frac{A}{f^{1.5}}, \text{ where A is a constant.}$$

Therefore after integration we get:

$$\sigma = \sqrt{\int_{f_1}^{f_2} \frac{A}{f^{1.5}} \cdot df} = \left[2 \cdot A \cdot \left(f_1^{-0.5} - f_2^{-0.5} \right) \right]^{1/2} \tag{11}$$

In our case the lower frequency f_1 is just the inverse of the total time period (16384 steps) and the upper frequency f_2 is the inverse of one time step given as:

$$f_1 = \frac{1}{16384}, \quad \text{and} \qquad f_2 = \frac{1}{1} = 1$$

From our experiments, Noise rms, σ was calculated as 1.66 and the threshold level U_t was set to 1.83

Therefore from equation (2.11) we can calculate the value of constant A.

$$1.66 = \left[2 \cdot A \cdot \left((1/16384)^{-0.5} - 1^{-0.5}\right)\right]^{1/2} \Rightarrow A = 0.011$$

Now we can calculate the mean frequency of crossings using equation (2.9)

$$\nu(1.83) = 2 \cdot (1.66)^{-1} \exp\left[\frac{-(1.83)^2}{2(1.66)^2}\right] \cdot \left[\int_{1/16384}^{1} f^2 \cdot \frac{0.011}{f^{1.5}} \cdot df\right]^{1/2}$$

thus, $\nu(1.83) = 0.056$

3. Rice formula for $1/f^2$ noise

Here we have,

$$\sigma = \sqrt{\int_{f_1}^{f_2} S(f)\cdot df} \qquad \text{where} \qquad S(f) = \frac{A}{f^2}, \text{ where A is a constant.}$$

Therefore after integration we get:

$$\sigma = \sqrt{\int_{f_1}^{f_2} \frac{A}{f^2}\cdot df} = \left[A\cdot\left(f_1^{-1} - f_2^{-1}\right)\right]^{1/2} \tag{12}$$

In our case the lower frequency f_1 is just the inverse of the total time period (16384 steps) and the upper frequency f_2 is the inverse of one time step given as:

$$f_1 = \frac{1}{16384}, \quad \text{and} \qquad f_2 = \frac{1}{1} = 1$$

From our experiments, Noise rms, σ was calculated as 1.65 and the threshold level U_t was set to 1.82.

Therefore from equation (2.12) we can calculate the value of constant A.

$$1.65 = \left[A \cdot \left((1/16384)^{-1} - 1^{-1}\right)\right]^{1/2} \Rightarrow A = 0.000166$$

Now we can calculate the mean frequency of crossings using equation (2.9)

$$\nu(1.82) = 2 \cdot (1.65)^{-1} \exp\left[\frac{-(1.82)^2}{2(1.65)^2}\right] \cdot \left[\int_{1/16384}^{1} f^2 \cdot \frac{0.000166}{f^2} \cdot df\right]^{1/2}$$

thus, $\nu(1.83) = 0.0085$

CHAPTER III

SIMULATION RESULTS

A. Evaluation of Threshold Crossings without Periodic Drive

As mentioned in Chapter II, we have computed the average number of threshold crossings for $1/f$, $1/f^{1.5}$, and $1/f^2$ noise versus increasing window sizes, without the introduction of the sinusoidal signal, and plotted the results in figures 9 through 11. From the graphs we see that the average number of threshold crossings decreases as our window size gets larger. This means that once the threshold level is crossed by the noise, there is an increased probability that it will be crossed again in a shorter time; and as time goes on, that probability gets smaller. Since we are relating every threshold crossing of noise with the occurrence of a natural disaster, it is logical to say that after the occurrence of any natural disaster, there is an increased probability of its recurrence. The theoretical results for the mean number of threshold crossings derived from the Rice formula are also plotted in the figures 9 through 11. As expected, the slopes of both the theoretical and experimental graphs are very close. There is a slight difference in the slope for the case of $1/f$ noise due to some aliasing effects. There is also a shift in the theoretical curves; this is primarily due to the un-adjustable calibration of computer and theoretical models. However, more important aspect in the graphs is the slope of the curves which is very similar. This theoretically proves our proposition from the computer simulations that the rate of threshold crossings decreases with increasing window sizes.

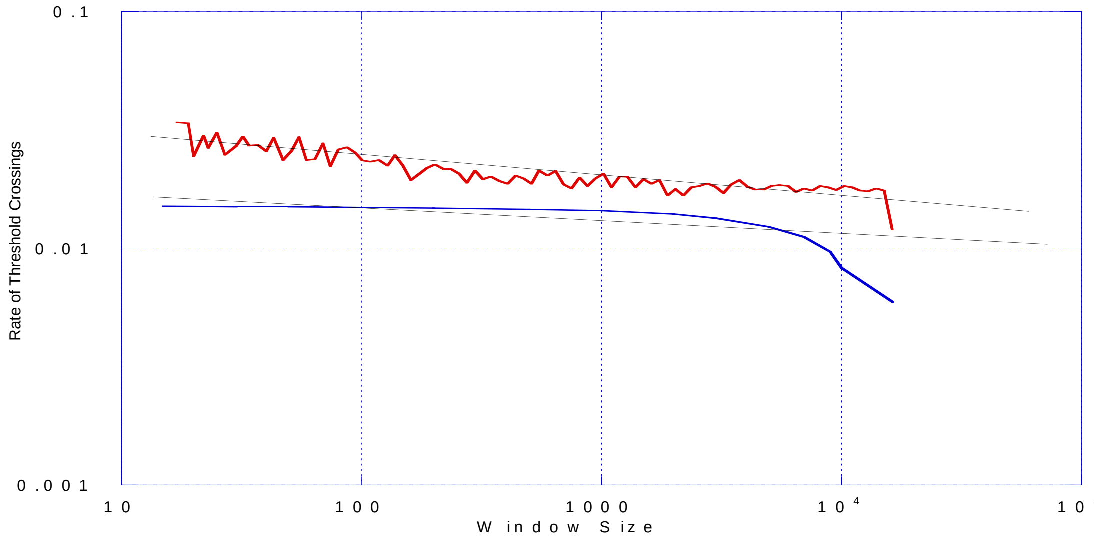

Fig. 9. Representation of Threshold Crossings for $1/f$ noise (Nrms=1.65, Th=1.82)

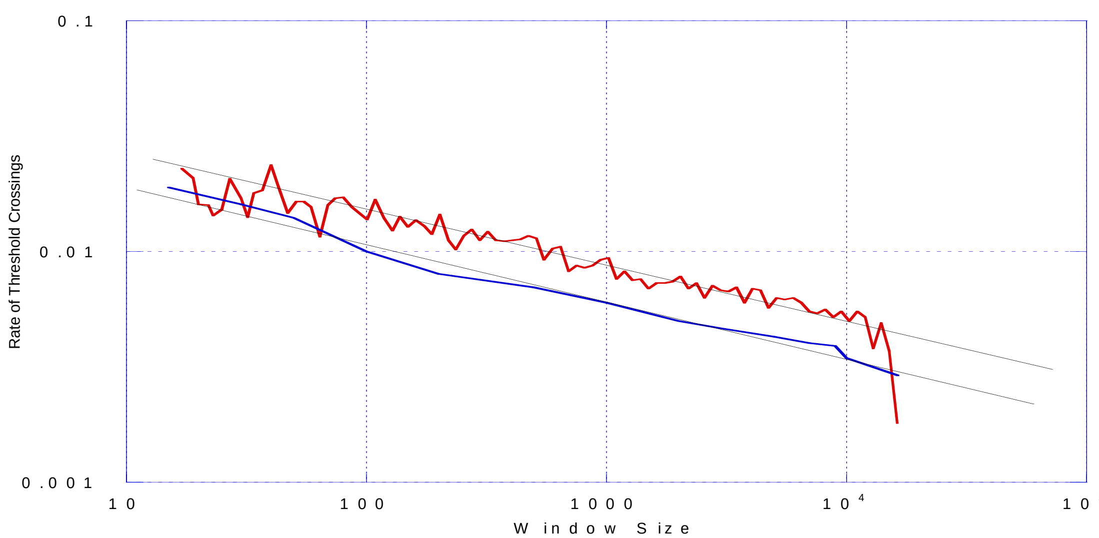

Fig. 10. Representation of Threshold Crossings for $1/f^{1.5}$ noise (Nrms=1.66, Th=1.83)

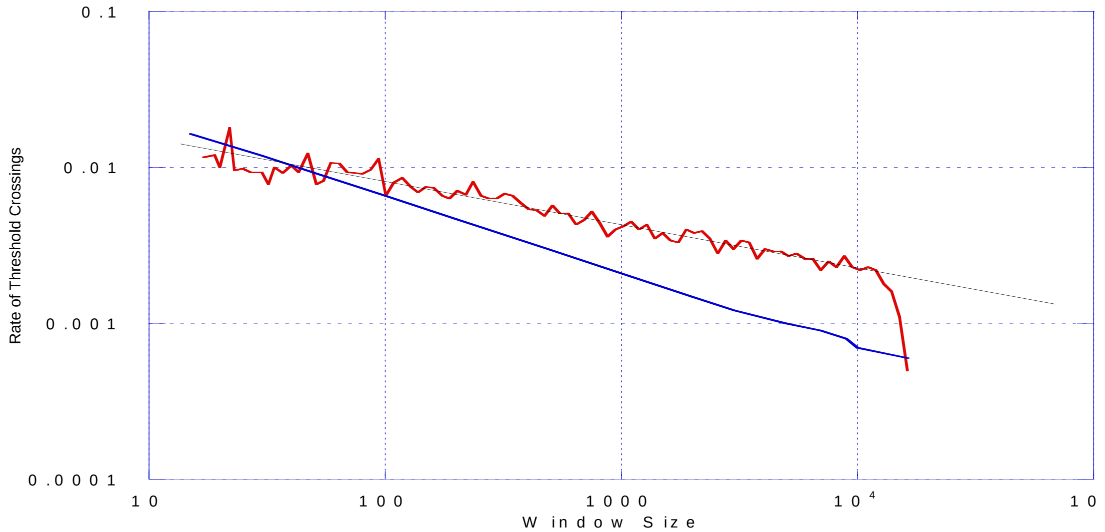

Fig. 11. Representation of Threshold Crossings for $1/f^2$ noise (Nrms=1.65, Th=1.82)

B. Evaluation of Threshold Crossings with Periodic Drive

We have also computed the average number of threshold crossings for $1/f$, $1/f^{1.5}$, and $1/f^2$ noise with the introduction of a small periodic drive. For this case we shall investigate the number of successful sinusoidal periods that contain at least one threshold crossing. It should be noted that we are not interested in evaluating the number of times the threshold is crossed during a certain window period. Instead we are keen to know how many successive sinusoids that are added to the noise, contain at least one threshold crossing. In practical terms, this means that if a particular natural disaster occurs in a particular year, we are computing the successive number of years that it is likely to recur. After looking at the figures 12 through 23, it has been observed that once the threshold is crossed, the probability that it will be crossed again in the very next sinusoid is the highest. The probability gets smaller as we start examining more successive sinusoids. Practically this means that once a natural disaster occurs in a particular year, it highly likely that it will recur in the next year, less likely that it will recur in the year after that, and even less likely that will recur in the following one, and so on. Practically this implies that the probability of recurrence of a natural disaster decreases at a very high rate as time passes and it is highest in the year just after the occurrence of the first disaster. A very important observation from the graphs is that the exponential decay is greatest for $1/f^2$ noise and smallest for $1/f$ noise. Basically it is observed that larger the negative exponent of the noise is, faster is the rate of decay of the number of threshold crossings. From the graphs 12 through 23 below, we have obtained the curve-fit equations; these equations are summarized in table 1 below:

Table 1. Summary of Curve-fits for Graphs with Periodic Drive

	$1/f$ Noise	$1/f^{1.5}$ Noise	$1/f^{2}$ Noise
Graph 1:	$543.19\times e^{-0.25363\cdot x}$	$736.91\times e^{-0.46571\cdot x}$	$1092\times e^{-0.69386\cdot x}$
Graph 2:	$1222\times e^{-0.45979\cdot x}$	$846.49\times e^{-0.49251\cdot x}$	$331.79\times e^{-0.44813\cdot x}$
Graph 3:	$636.16\times e^{-0.27634\cdot x}$	$953.56\times e^{-0.51868\cdot x}$	$676.87\times e^{-0.69206\cdot x}$
Graph 4:	$258.71\times e^{-0.25451\cdot x}$	$883.02\times e^{-0.3485\cdot x}$	$1309.1\times e^{-0.67278\cdot x}$

It is visible from the table above that the rate of decay of the threshold crossings increases as the negative exponent of the noise increases. Also for the case when the noise level and the sine wave amplitude add up to the threshold value (graphs 1 above), the number of overall threshold crossings is highest for $1/f^{2}$ noise and least for $1/f$ noise. This implies that for $1/f^{2}$ noise case, there will be much more threshold crossings in the very next sine wave as opposed to $1/f$ noise case. As mentioned earlier, figures 12 through 23 show the average number of threshold crossings for the different noise with the introduction of the periodic drive.

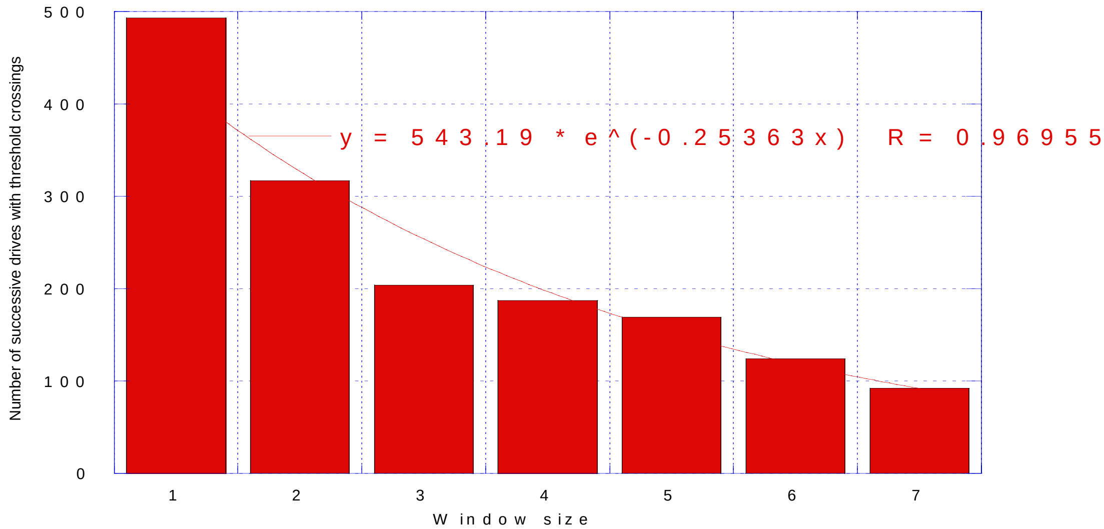

Fig. 12: Number of Periods of Successive Threshold Crossings (1/f, Nrms=1.14, Srms=0.1, Th = 1.24)

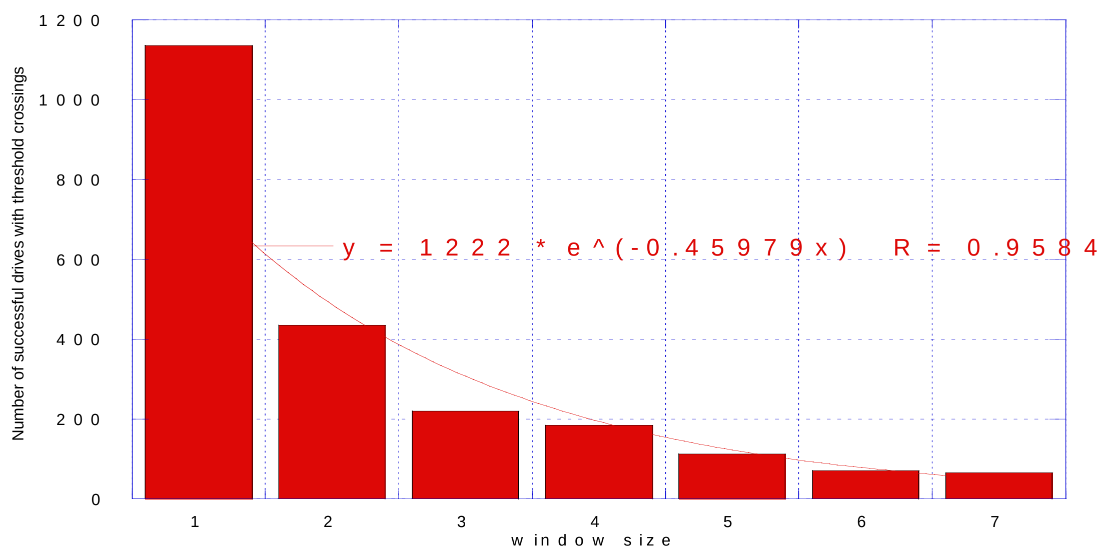

Fig 13: Number of Periods of Successive Threshold Crossings (1/f, Nrms=1.34, Srms=1.34, Th = 2.68)

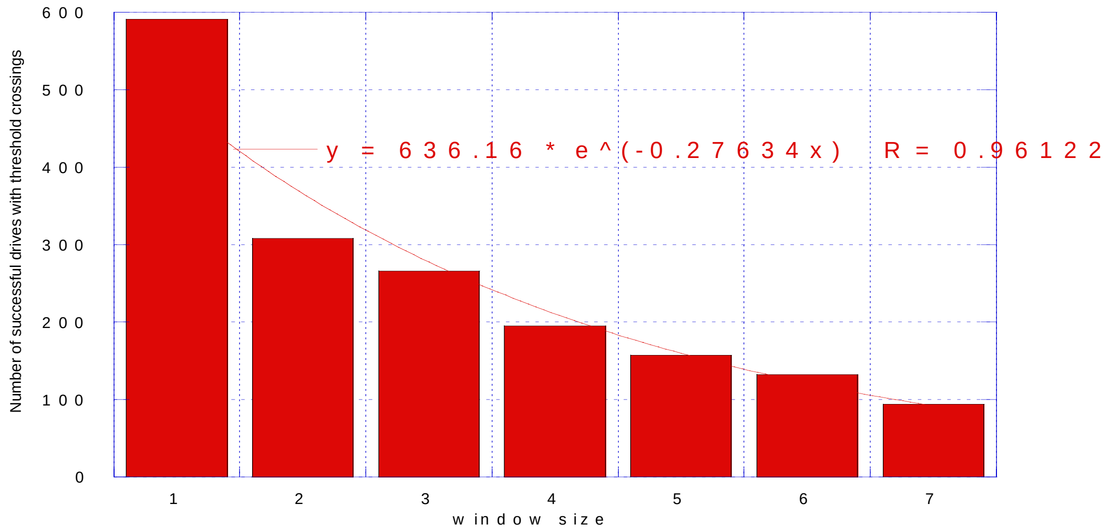

Fig 14: Number of Periods of Successive Threshold Crossings (1/f, Nrms=1.4, Srms=0, Th = 1.4)

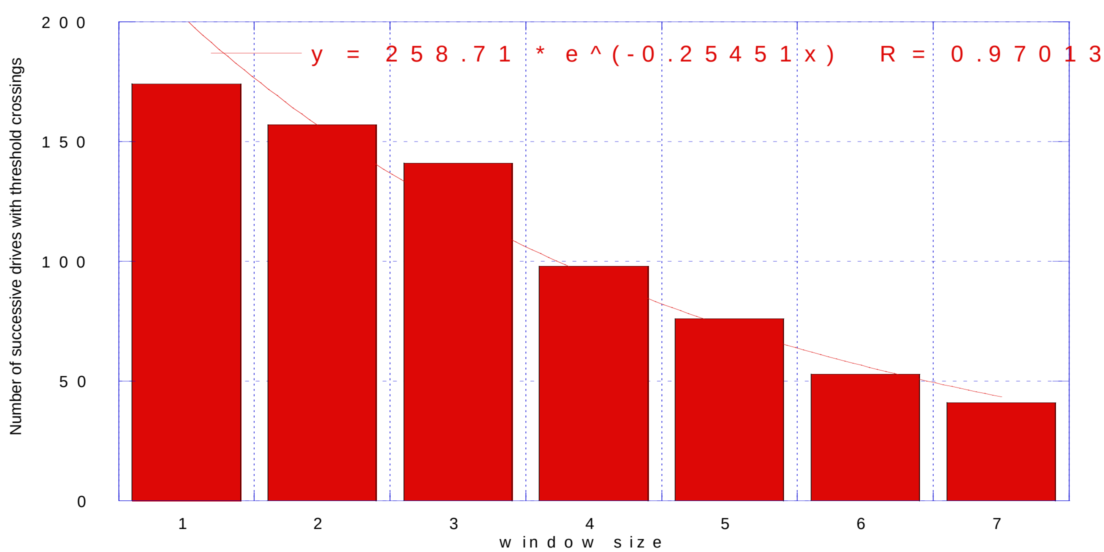

Fig 15: Number of Periods of Successive Threshold Crossings(1/f, Nrms=1.4, Srms=0, Th =0)

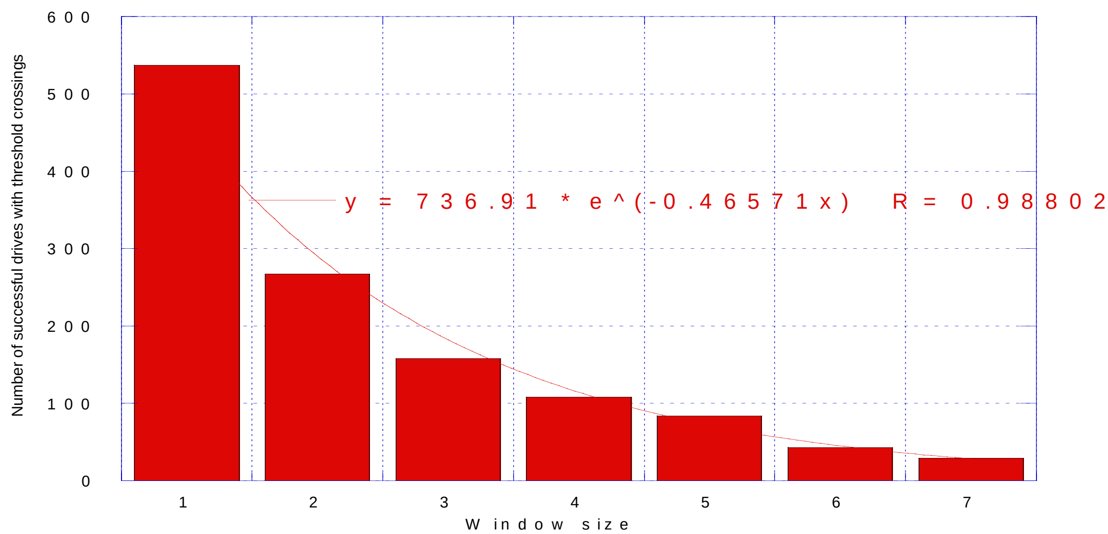

Fig 16: Number of Periods of Successive Threshold Crossings ($1/f^{1.5}$, Nrms=1.65, Srms=0.1, Th=1.75)

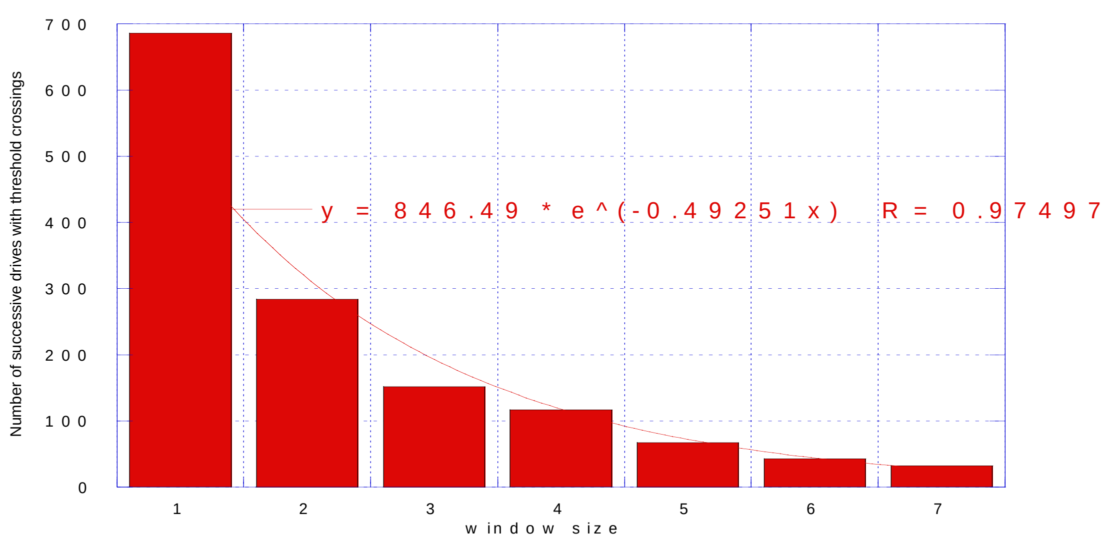

Fig 17: Number of Periods of Successive Threshold Crossings ($1/f^{1.5}$, Nrms=1.13, Srms=1.13, Th=2.26)

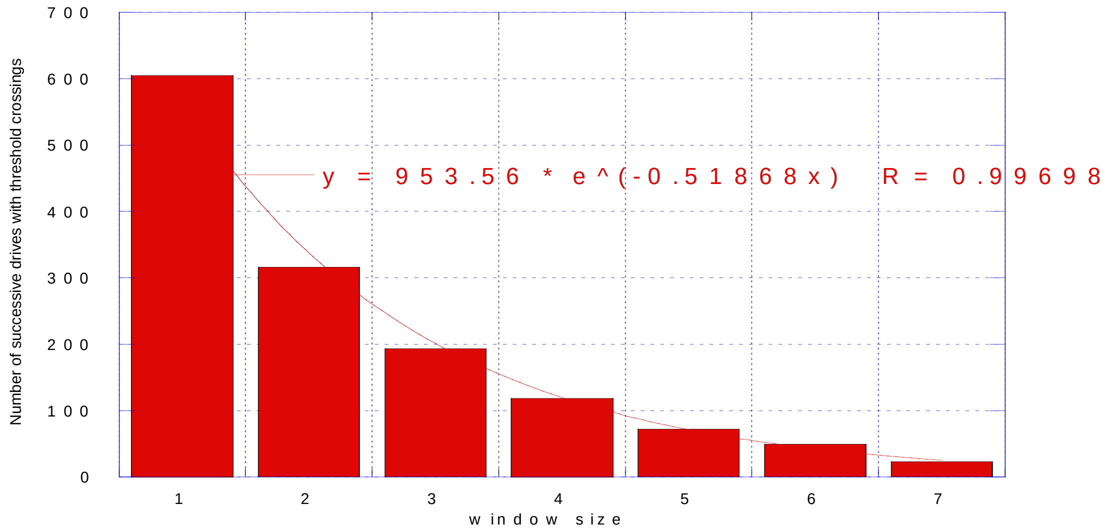

Fig 18: Number of Periods of Successive Threshold Crossings ($1/f^{1.5}$, Nrms=1.64, Srms=0, Th=1.64)

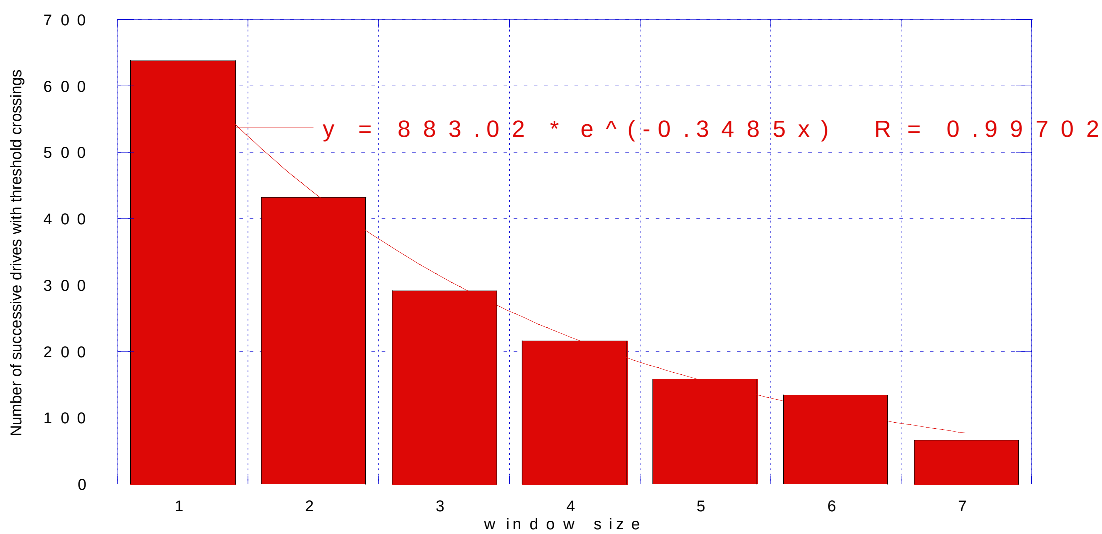

Fig 19: Number of Periods of Successive Threshold Crossings ($1/f^{1.5}$, Nrms=1.43, Srms=0, Th=0)

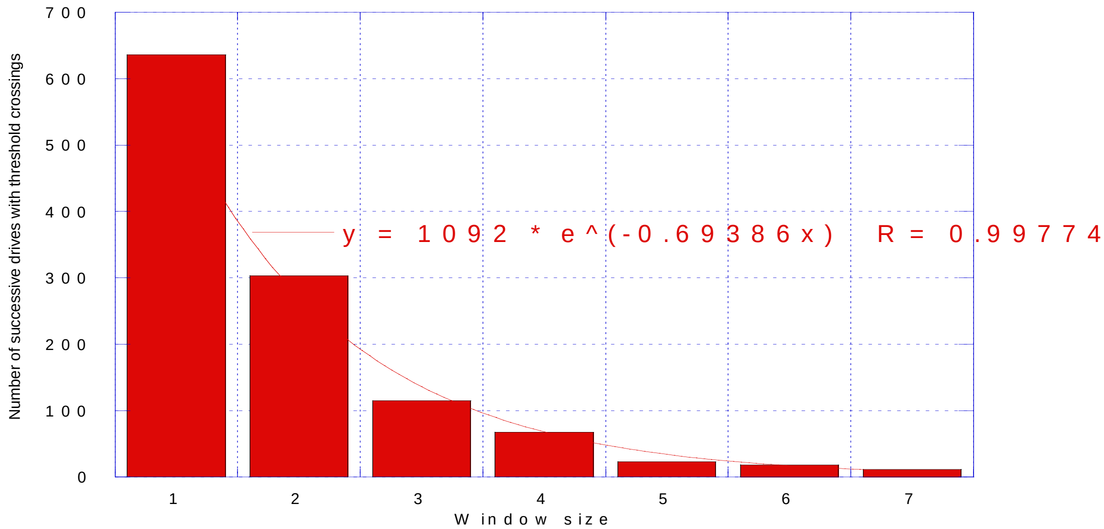

Fig 20: Number of Periods of Successive Threshold Crossings ($1/f^2$, Nrms=0.9, Srms=0.1, Th = 0.99)

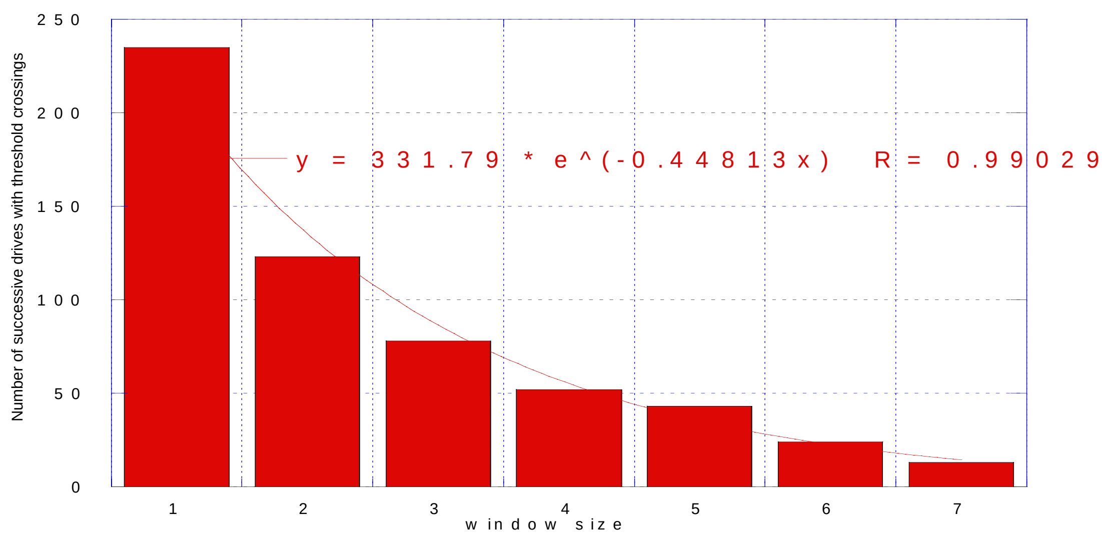

Fig 21: Number of Periods of Successive Threshold Crossings ($1/f^2$, Nrms=1.4, Srms=1.4, Th = 2.8)

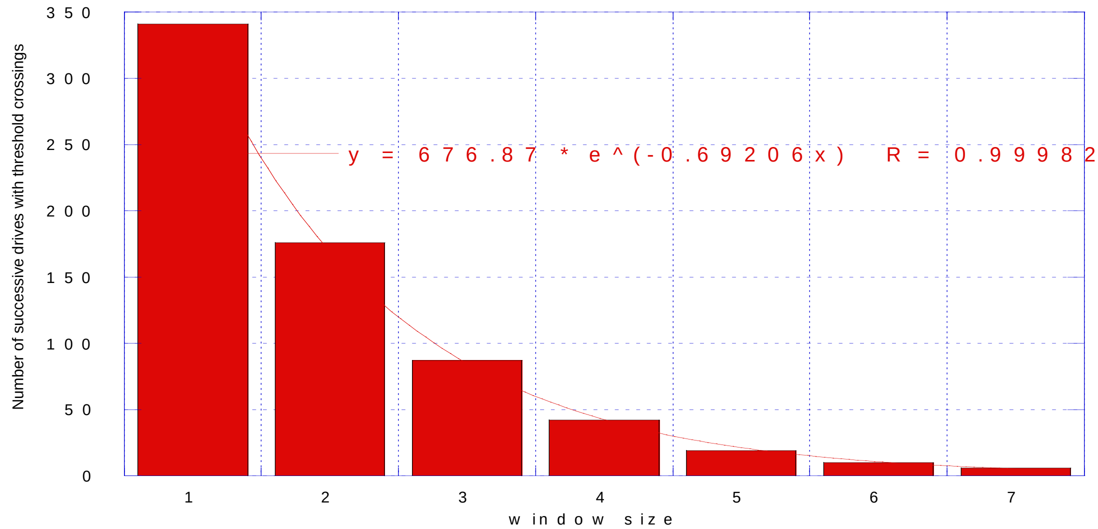

Fig 22: Number of Periods of Successive Threshold Crossings ($1/f^2$, Nrms=1.64, Srms=0, Th = 1.64)

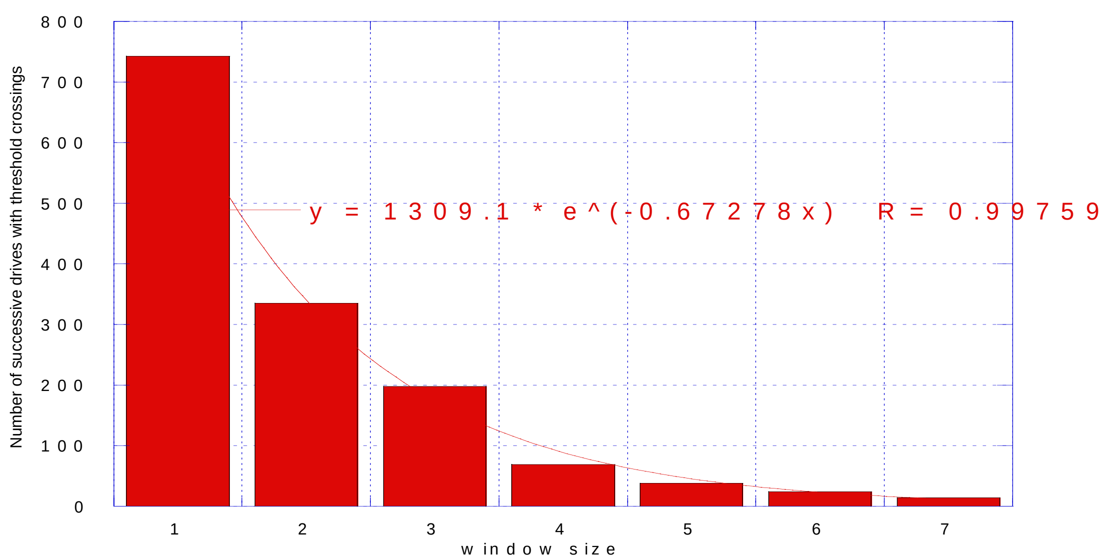

Fig 23: Number of Periods of Successive Threshold Crossings ($1/f^2$, Nrms=0.88, Srms=0, Th = 0)

CHAPTER IV

CONCLUSION

In this thesis we have focused on the recurrence behavior of rare threshold crossings due to noise with long memory. The behavior of long-time correlated noise can be related to the behavior of many natural phenomena – natural disasters and false alarms for our case. We have observed that the probability of recurrence of threshold crossings varies as we observe the noise process for different time periods. The probability of a second threshold crossing is highest within a short time after the first threshold crossing. We have compared the behavior of threshold crossing events of $1/f$, $1/f^{1.5}$, and $1/f^{2}$ noise from our computer simulations with analytical results from the Rice formula and have observed many similarities. We have seen that the rate of decrease of number of threshold crossings as we observe the crossings for longer times is the same. Both analytical and computer simulation emphasize the fact that the probability of a second threshold crossing is highest in a shorter time after the first crossing. Since we are relating the occurrence of every threshold crossing with the occurrence of a natural disaster or a false alarm, we can safely conclude that once a natural disaster or false alarm occurs, there is a high probability that it will recur in a short time.

REFERENCES

[1] D. Vere-Jones, *Forecasting earthquakes and earthquake risk*, Institute of Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand, (1995) SSDI 0169-2070 (95) 00621-4.

[2] S. Kurkovsky and R. Loganantharaj, *Modeling of, and reasoning with recurrent events with imprecise durations*, Dept. of Computer Science, Columbus State University, Columbus, (2000).

[3] J.K. Lindsey, LUC Diepenbeek, ULG Liege, *The Statistical Analysis of Stochastic Processes in Time*, John Wiley and Sons, Inc., New York (2001).

[4] B. Kaulakys, V. Gontis, and M. Alaburda, *Point process model of* $1/f$ *noise vs. a sum of Lorentzians*, Institute of Theoretical Physics and Astronomy, Vilnius University, Vilnius, Lithuania, (2005).

[5] T. Antal, M. Droz, G. Gyorgyi, and Z. Racz, *Roughness distributions of* $1/f^{\alpha}$ *signals*, University of Paris-Sud, 91405 Orsay Cedex, France, (2001).

[6] K. Loerincz, Z. Gingl, and L.B. Kiss, *A stochastic resonator is able to greatly improve signal-to-noise ratio, Phys. Lett.* A **224** (1996) 1.

[7] S. Watts, *Duration of radar false alarms in band-limited Gaussian noise*, in IEE Proc.-Radar, Sonar Navigation, **146,** 6, (1999) 213-276.

[8] L.B. Kiss, *Possible breakthrough: Significant improvement of signal to noise ratio by stochastic resonance*, *Chaotic, Fractal, and Non-linear Signal Processing,* R. Katz, editor, American Institute of Physics Press, Melville, New York (1996) 1-56396-443-0.

APPENDIX A

MATLAB CODE FOR THE SIMULATION MODEL

A. Threshold Crossing Case without Periodic Drive

This part of Matlab simulation code was divided into five M-files, each of which had a unique function to perform. The whole task of evaluating the threshold crossings was broken down into simpler, smaller tasks and performed accordingly. These simpler tasks were Noise Generation, Level Crossing Detection, Crossing Calculation, RMS evaluation, and a function for controlling all of these simpler tasks to work hand-in-hand. Looking more closely, the first step would be to generate an array of noise specified by the user; this array would then be sent to a level crossing detector module that would indicate instances of threshold crossings in regards to a particular threshold value provided to it. More specifically it would generate an array of ones and zeros; ones being the points where the noise successfully crossed the threshold value and vice versa for zeros. Once this was accomplished, this array of ones and zeros would be passed on to a function called crossing detector. The function would calculate how many ones and zeros were present in the array and thus evaluate the probability of threshold crossing, which is just the number of threshold crossings in a particular window size divided by the size of the window. We also had to define an RMS calculation function that would help us calculate the rms value of the noise. This was helpful in cases where we had to define the threshold value as some percentage of the rms value of the noise. There was also a controlling function called “main” which was responsible for calling all the functions mentioned above to work hand-in-hand. This was very helpful because we had to take multiple records of the threshold crossings and average the statistics to come up with our graphs. The code for each module is given below:

Level Crossing Detector:

```
function output_lcd = Level_Crossing_Detector(incoming_signal,
TH_value, size_value)
prev_amp = 0;
poscrossing = 0;
negcrossing = 0;
for i = 1:1:(size_value)
    curr_amp = incoming_signal(i);
    if ((prev_amp < TH_value) & (curr_amp > TH_value))
        output_lcd(i) = 1;
        poscrossing = poscrossing + 1;
    else
        output_lcd(i) = 0;
    end
    prev_amp = curr_amp;
end
return;
```

Crossing Detector:

```
function Average_crossing = crossing_detector(lcd_output, SIZE,
WINDOW)

for i = 1:1:SIZE
    if ((lcd_output(i) == 1)&((i + WINDOW) <= SIZE))
        crossing_sum = 0;
        for j = (i+1):1:(i + WINDOW)
            crossing_sum = crossing_sum + lcd_output(j);
        end
        break;
    else
        crossing_sum = 0;
    end
end
Average_crossing = (crossing_sum/WINDOW);
return;
```

RMS Calculator:

```
function rms_value = rms_calc(input_array, SIZE)
input_array_square = input_array.*input_array;
input_array_mean_square = mean(input_array_square);
rms_value = input_array_mean_square^0.5;
return;
```

Noise Generator:

```
function averaged_noise = Noise_Generate_Average(length, k,
noisefactor)

pack
format compact

while length>15;
      'length ² 15  !!!'
      length = input('length must be smaller than 15, enter again:');
end

noiselength=2^length;


      noise = (rand(1,noiselength)-0.5);

      spectrum=fft(noise);

    for t = 1:1:noiselength
        theta = 2*pi*(rand(1)) - pi;
        spectrum_scramble_phase(t) =
complex((abs(spectrum(t)))*cos(theta) ,
(abs(spectrum(t)))*sin(theta));
    end

      p = k/2 ;

      for f=1:noiselength
            spectrum_scramble_phase(f) = f^p *
spectrum_scramble_phase(f);
      end

      energy=sum((abs(spectrum_scramble_phase)).^2);

      power=energy/noiselength;
      norm=power^0.5;

      noise = (1 +
10*noisefactor)*real(ifft(spectrum_scramble_phase)/norm);

      averagenoise = sum(noise)/noiselength;

      noise = noise - averagenoise;

for x = 1:1:(noiselength-4)
    averaged_noise(x) = (noise(x) + noise(x+1) + noise(x+2) +
noise(x+3) + noise(x+4))/5;
end
averaged_noise(noiselength-3) = noise(noiselength-3);
averaged_noise(noiselength-2) = noise(noiselength-2);
averaged_noise(noiselength-1) = noise(noiselength-1);
averaged_noise(noiselength) = noise(noiselength);
return;
```

Main:

```
clear;
%###################################################################
global SIGNAL_FREQ SIGNAL_AMP THRESHOLD LENGTH SIZE NOISE_TYPE WINDOW

%SIGNAL_FREQ = input('Enter Signal Frequency (in Hz):  ');
%SIGNAL_AMP = input('Enter Signal Amplitude:  ');

NOISE_TYPE = input('Type of noise to generate (f^k), enter k:  ');
PERCENT_THRESHOLD = input('Enter Threshold percentage (#%):  ');
LENGTH = 14;     %input('Enter Array length (2^length), enter length:
');
SIZE = 2^LENGTH;
TOTAL_RECORD = input('Enter total number of records:  ');
NOISE_LEVEL = input('Enter Noise level:  ');

for x = 1:1:8
    window_length(x) = x + 1;
end
for x = 30:1:126
    window_length(x-21) = round(1.08^(x));
end
NF = NOISE_LEVEL;

Tester_noise_array = Noise_Generate_Average(LENGTH, NOISE_TYPE, NF);
Noise_rms_value = rms_calc(Tester_noise_array, SIZE);
THRESHOLD = (PERCENT_THRESHOLD/100)*Noise_rms_value;

%Input_Signal = Signal_Generate(SIGNAL_AMP, SIGNAL_FREQ, SIZE);

Avg_per_Window = [];

for window_number = 1:1:size(window_length,2)
    WINDOW = window_length(window_number);

    Avg_crossings_holder = [];

    for RECORD = 1:1:TOTAL_RECORD
        Input_Noise = Noise_Generate_Average(LENGTH, NOISE_TYPE, NF);

        %Signal_Noise = Input_Signal + Input_Noise;

        LCD output  =  Level Crossing Detector(Input Noise,  THRESHOLD,
SIZE);
        Avg_crossings = crossing_detector(LCD_output, SIZE, WINDOW);
        Avg_crossings_holder(RECORD)         =         Avg crossings;
%This line puts all crossings wrt. RECORD number
    end

    Avg per Window(window number)     =    mean(Avg crossings holder);
%This line puts all avg. crossing wrt. window length
end

Avg_per_Window
window_length
plot (window_length, Avg_per_Window)
clear;
```

B. Threshold Crossing Case with Periodic Drive

This part of Matlab simulation code was divided into six M-files, each of which had a unique function to perform. The whole task of evaluating the threshold crossings with the sinusoidal signal was broken down into simpler, smaller tasks and performed accordingly. These simpler tasks were Noise Generation, Signal Generation, Level Crossing Detection, Crossing Calculation, RMS evaluation, and a function for controlling all of these simpler tasks to work hand-in-hand. Looking more closely, the first step would be to generate an array of noise specified by the user, this was then added to a sinusoidal signal array of specified frequency and amplitude; this array would then be sent to a level crossing detector module that would indicate instances of threshold crossings in regards to a particular threshold value provided to it. More specifically it would generate an array of ones and zeros; ones being the points where the noise successfully crossed the threshold value and vice versa for zeros. Once this was accomplished, this array of ones and zeros would be passed on to a function called crossing detector. The function would calculate how many ones and zeros were present in the array within each period of the sinusoidal signal. This was done to predict how many times the noise plus signal cross the threshold level within each period of the sine wave. This function would then evaluate how many successive sinusoids had at least one threshold crossing. A total of seven successive sinusoids were analyzed for each noise process. The result would be the maximum number of successive sinusoids with at least one threshold crossing. We also had to define an RMS calculation function that would help us calculate the rms value of the noise. This was helpful in cases where we had to define the threshold value as some percentage of the rms value of the noise. There was also a controlling function called “main” which was responsible for calling all the functions mentioned above to work hand-in-hand. This was very helpful because we had to take multiple records of the threshold crossings and average the statistics to come up with our graphs. The code for each module is given below:

Level Crossing Detector:

```
function output_lcd = Level_Crossing_Detector(incoming_signal,
TH_value, size_value)
prev_amp = 0;
poscrossing = 0;
for i = 1:1:(size_value)
    curr_amp = incoming_signal(i);
    if ((prev_amp < TH_value) & (curr_amp > TH_value))
        output_lcd(i) = 1;
        poscrossing = poscrossing + 1;
    else
        output_lcd(i) = 0;
    end
    prev_amp = curr_amp;
end
return;
```

Crossing Detector:

```
function sequence_sum_array = crossing_detector(lcd_output, SIZE,
SIGNAL_FREQ)
crossing_sum = [];
crossing_per_cycle = [];
for i = 1:1:SIGNAL_FREQ
    for j = (1 + ((i-1)*(SIZE/SIGNAL_FREQ))):1:(i*(SIZE/SIGNAL_FREQ))
        k = j - ((i-1)*(SIZE/SIGNAL_FREQ));
        crossing_per_cycle(k,i) = lcd_output(j);
    end
end

crossing_sum = sum(crossing_per_cycle);

i = 1;
j = 1;
while i ~= 128

 if ((i+8 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~= 0
&& crossing_sum(i+2) ~= 0 && crossing_sum(i+3) ~= 0 &&
crossing_sum(i+4) ~= 0 && crossing_sum(i+5) ~= 0 && crossing_sum(i+6)
~= 0 && crossing_sum(i+7) ~= 0))
     sequence_sum = 8;
```

```
     i = i + 8;
 elseif ((i+7 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0 && crossing_sum(i+2) ~= 0 && crossing_sum(i+3) ~= 0 &&
crossing_sum(i+4) ~= 0 && crossing_sum(i+5) ~= 0 && crossing_sum(i+6)
~= 0))
     sequence_sum = 7;
     i = i + 7;
 elseif ((i+6 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0 && crossing_sum(i+2) ~= 0 && crossing_sum(i+3) ~= 0 &&
crossing_sum(i+4) ~= 0 && crossing_sum(i+5) ~= 0))
     sequence_sum = 6;
     i = i + 6;
 elseif ((i+5 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0 && crossing_sum(i+2) ~= 0 && crossing_sum(i+3) ~= 0 &&
crossing_sum(i+4) ~= 0))
     sequence_sum = 5;
     i = i + 5;
 elseif ((i+4 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0 && crossing_sum(i+2) ~= 0 && crossing_sum(i+3) ~= 0))
     sequence_sum = 4;
     i = i + 4;
 elseif ((i+3 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0 && crossing_sum(i+2) ~= 0))
     sequence_sum = 3;
     i = i + 3;
 elseif ((i+2 < 128) && (crossing_sum(i) ~= 0 && crossing_sum(i+1) ~=
0))
     sequence_sum = 2;
     i = i + 2;
 elseif ((i+1 < 128) && (crossing_sum(i) ~= 0))
     sequence_sum = 1;
     i = i + 1;
 else
     sequence_sum = 0;
     i = i + 1;
 end
 sequence_sum_array(j) = sequence_sum;
 j = j + 1;
end
 sequence_sum_array;
return;
```

Noise Generator:

```
function averaged_noise = Noise_Generate_Average(length, k,
noisefactor)

pack
format compact

while length>15;
      'length ² 15  !!!'
      length = input('length must be smaller than 15, enter again:
');
end

noiselength=2^length;

      noise = (rand(1,noiselength)-0.5);
```

```
      spectrum=fft(noise);

    for t = 1:1:noiselength
        theta = 2*pi*(rand(1)) - pi;
        spectrum_scramble_phase(t) =
complex((abs(spectrum(t)))*cos(theta) ,
(abs(spectrum(t)))*sin(theta));
    end

      p = k/2 ;

      for f=1:noiselength
            spectrum_scramble_phase(f) = f^p *
spectrum_scramble_phase(f);
      end

      energy=sum((abs(spectrum_scramble_phase)).^2);

      power=energy/noiselength;
      norm=power^0.5;

      noise = (1 +
10*noisefactor)*real(ifft(spectrum_scramble_phase)/norm);

      averagenoise = sum(noise)/noiselength;

      noise = noise - averagenoise;

for x = 1:1:(noiselength-4)
    averaged_noise(x) = (noise(x) + noise(x+1) + noise(x+2) +
noise(x+3) + noise(x+4))/5;
end
averaged_noise(noiselength-3) = noise(noiselength-3);
averaged_noise(noiselength-2) = noise(noiselength-2);
averaged_noise(noiselength-1) = noise(noiselength-1);
averaged_noise(noiselength) = noise(noiselength);
return;
```

Signal Generator:

```
function A = Signal_Generate(SIGNAL_AMP, SIGNAL_FREQ, SIZE)
t = 0:0.001:((SIZE-1)*0.001);
A = SIGNAL_AMP*sin(2*pi*t*SIGNAL_FREQ/(0.001*SIZE));
return;
```

RMS Calculator:

```
function rms_value = rms_calc(input_array, SIZE)
input_array_square = input_array.*input_array;
input_array_mean_square = mean(input_array_square);
rms_value = input_array_mean_square^0.5;
return;
```

Main:

```
global SIGNAL_FREQ SIGNAL_AMP THRESHOLD LENGTH SIZE NOISE_TYPE

LENGTH = 14;
SIZE = 2^LENGTH;
SIGNAL_FREQ = 128;
NOISE_TYPE = input('Type of noise to generate (f^k), enter k:  ');
NF = input('Enter Noise factor:  ');
PERCENT_THRESHOLD = input('Enter Threshold percentage (#%):  ');

Tester_noise_array = Noise_Generate_Average(LENGTH, NOISE_TYPE, NF);
Noise_rms_value = rms_calc(Tester_noise_array, SIZE)
THRESHOLD = (PERCENT_THRESHOLD/100)*Noise_rms_value

SIGNAL_AMP = input('Enter Signal Amplitude:  ');
TOTAL_RECORD = input('Enter total number of records:  ');

Input_Signal = Signal_Generate(SIGNAL_AMP, SIGNAL_FREQ, SIZE);
%plot(Input_Signal)

sequence_holder = [];

for RECORD = 1:1:TOTAL_RECORD

Input_Noise = Noise_Generate_Average(LENGTH, NOISE_TYPE, NF);
Signal_Noise = Input_Signal + Input_Noise;
LCD_output = Level_Crossing_Detector(Signal_Noise, THRESHOLD, SIZE);

sequence_sum_array = crossing_detector(LCD_output, SIZE,
SIGNAL_FREQ);

for i = 1:1:size(sequence_sum_array,2)
    array_sequence(RECORD, i) = sequence_sum_array(i);
end
end
array_sequence;


k = 1;
for RECORD = 1:1:TOTAL_RECORD
    for i = 1:1:size(array_sequence,2)
        if array_sequence(RECORD, i) ~= 0
            final_array(k) = array_sequence(RECORD, i);
            k = k+1;
        end
    end
end

final_array

A = 1:1:7;
B = final_array;
hist(B, A)

clear;
```

VITA

Abhishek Narayan Singh was born in Uttar Pradesh, India. He obtained the Bachelor of Science in electrical engineering degree specializing in electronic circuit design from the University of Texas at Arlington in August 2003. He joined the Solid State Electronics group in the Electrical Engineering Department at Texas A&M University in Fall 2003 to pursue the Master of Science degree. His prime area of interest is design and test of Low Noise Electronic Devices for Wireless Communications. His permanent address is c/o Dr. Laszlo B. Kish, Department of Electrical Engineering, Texas A&M University, College Station, Texas - 77840; and email address is abhishek125@hotmail.com.

The typist for this thesis was Abhishek Narayan Singh.