A Quick Study of Oscillations in a BCM Neuron

(April 9, 1996)

Objective

The purpose of this study is to examine oscillations in the activity of a BCM neuron, as the neuron adapts to an input. We want to explore how the various aspects of the oscillations (frequency, amplitude, etc...) depend on the parameters of the BCM model.

1 The Theory

This study is to organize my thoughts on the subject, and to present the material in a way which highlights the main ideas and goals of the topic. It is by no means complete, but in writing it, I hope to gain some significant insights.

1.1 Defining the Differential Equations

In this study we will look at the simplified system of one neuron, with a single input and a single output (see Figure 1). We model the activity of the neuron (c) as merely

$$c = m \cdot d \tag{1.1}$$

where d is the input strength d and m is the "efficiency" of the neuron (most commonly refered to as the *weight*). This choice is used because it is the simplest one we can make. Later we may choose to use a slightly more complicated expression, such as $c = g(m \cdot d)$ (with g being a sigmoidal function), in order to preserve some physical reasonability for especially large inputs.

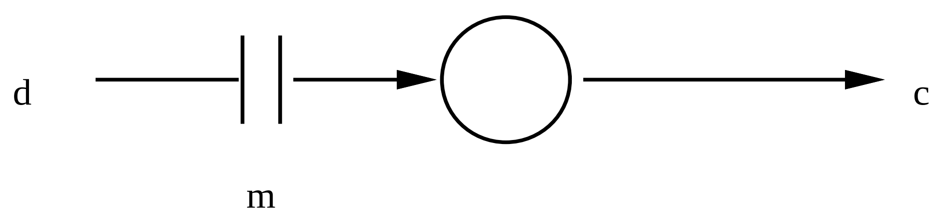

Figure 1: A Single Neuron

Our task now is to obtain a rule for neuron learning, or in terms of the math, an expression for the time derivative of the neuron weight. In this study we will look at two learning rules associated with the BCM model.

$$\dot{m} = \eta c(c - \theta)d \equiv \eta \phi_1(c, \theta)d \tag{1.2}$$

and

$$\dot{m} = \frac{\eta c(c - \theta)d}{\theta} \equiv \eta \phi_2(c, \theta)d \tag{1.3}$$

where θ is given by the expression

$$\theta = \frac{1}{\tau} \int_{-\infty}^{t} c^2(t') e^{-(t-t')/\tau} dt' \tag{1.4}$$

We will refer to η as the "learning rate", because it governs how quickly the weight changes, thus how quickly the neuron learns. The "memory constant", τ (in some cases we use $1/\kappa$), determines how much of the previous behavior of the neuron goes into the calculation of θ. As seen in Figure 2, θ and 0 are the fixed points for the neuron weights. As time moves forward, θ will change in order to *catch up* with the input (see Equation 1.4), which will result in $\dot{m}$ going to zero, and the neuron will have become selective to the input.

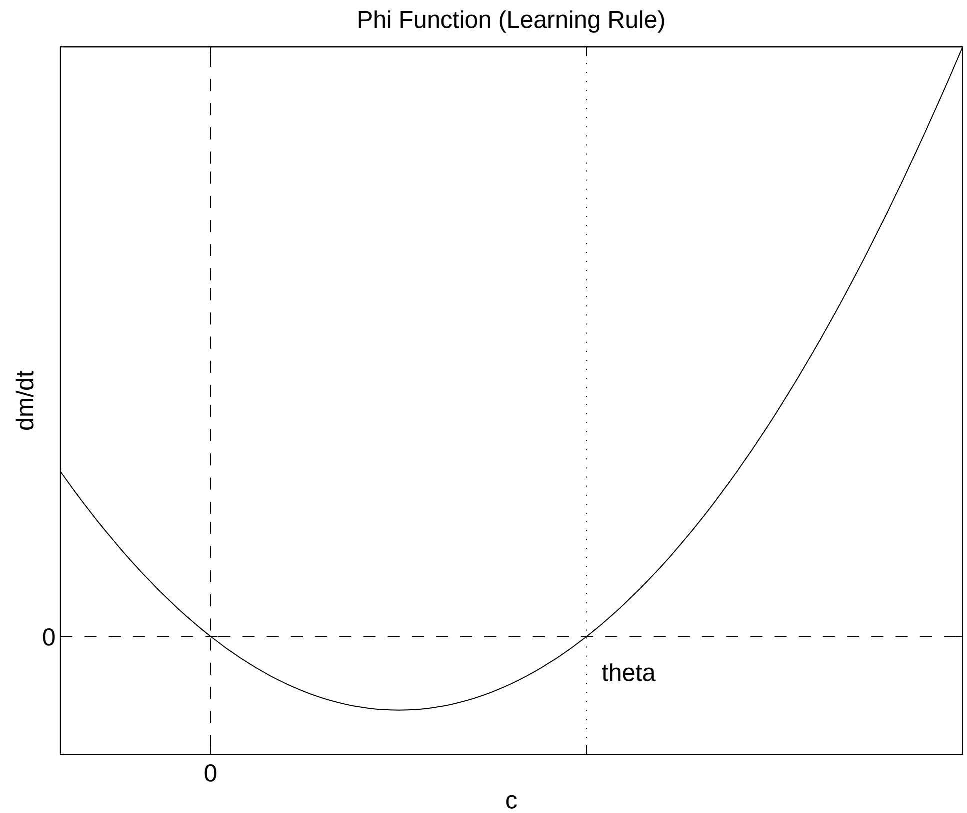

Figure 2: The BCM Learning Rule

1.2 Discretizing the Differential Equations

In order to properly simulate a neural system, we must find a discrete form of the differential equations. We will assume that the simulation will step through time with discrete time steps of magnitude Δt, and that we will be looking at the system after n of these steps. Our goal in this section is to write down, and verify, expressions for $c(n\Delta t)$, $\frac{\Delta m(n\Delta t)}{\Delta t}$, and $\theta(n\Delta t)$ which are the discrete equivalents of Equations 1.1, 1.2, and 1.4. The following equations give these expressions.

$$c(n\Delta t) = m(n\Delta t) \cdot d(n\Delta t) \tag{1.5}$$

$$\frac{\Delta m(n\Delta t)}{\Delta t} \equiv \frac{m((n+1)\Delta t) - m(n\Delta t)}{\Delta t} = \eta c(n\Delta t)(c(n\Delta t) - \theta(n\Delta t))d(n\Delta t) \tag{1.6}$$

$$\theta(n\Delta t) = (1 - \frac{\Delta t}{\tau})\theta((n-1)\Delta t) + \frac{\Delta t}{\tau}c^2(n\Delta t) \tag{1.7}$$

Equations 1.5 and 1.6 are obviously correct, but there are some assumptions in going between Equation 1.7 and Equation 1.4 which bear some closer investigation. First we write down the first few values of θ as time steps forward, in order to obtain an expression for $\theta(n\Delta t)$.

$$
\begin{aligned}
\theta(0) &= \theta_o \\
\theta(\Delta t) &= (1-\frac{\Delta t}{\tau})\theta_o + \frac{\Delta t}{\tau}c^2(\Delta t) \\
\theta(2\Delta t) &= (1-\frac{\Delta t}{\tau})^2\theta_o + (1-\frac{\Delta t}{\tau})\frac{\Delta t}{\tau}c^2(\Delta t) + \frac{\Delta t}{\tau}c^2(2\Delta t) \\
\theta(3\Delta t) &= (1-\frac{\Delta t}{\tau})^3\theta_o + (1-\frac{\Delta t}{\tau})^2\frac{\Delta t}{\tau}c^2(\Delta t) + (1-\frac{\Delta t}{\tau})\frac{\Delta t}{\tau}c^2(2\Delta t) + \frac{\Delta t}{\tau}c^2(3\Delta t) \\
&\vdots \\
\theta(n\Delta t) &= (1-\frac{\Delta t}{\tau})^n\theta_o + \sum_{j=1}^{n}(1-\frac{\Delta t}{\tau})^{n-j}\frac{\Delta t}{\tau}c^2(j\Delta t)
\end{aligned}
$$

which then can be approximated, for small $\frac{\Delta t}{\tau}$ as

$$\theta(n\Delta t) \overset{\frac{\Delta t}{\tau}\ll 1}{\approx} e^{-n\Delta t/\tau}\theta_o + \sum_{j=1}^{n} e^{-(n-j)\Delta t/\tau}\frac{\Delta t}{\tau}c^2(j\Delta t)$$

and finally, with $\Delta t \ll 1$, $n\Delta t \equiv t$, and $j\Delta t \equiv t'$ (the *summed* variable), the equation becomes

$$\theta(t) = \frac{1}{\tau}\int_0^t c^2(t')e^{-(t-t')/\tau}dt' + e^{-t/\tau}\theta_o \tag{1.8}$$

One immediately notices that Equation 1.8 is different than Equation 1.4 in two ways: the extra $e^{-t/\tau}\theta_o$ term in Equation 1.8, and the limits of the integration. Actually, these differences compensate for each other if we take $c(t<0)=0$ which then automatically sets the initial value of θ.

1.3 Immediate Consequences of the Model

In Section 1.1 it was mentioned that the value of θ would eventually catch up with the activity c, and result in a stead state. This evolution is shown clearly in Figure 3. It is also possible to obtain *oscillations*, where θ overshoots the activity, causing the activity to decrease ($\dot{m}$ would be less than zero), which would then force the θ function to decrease again and overshoot again. This oscillation is shown in Figure 4. Our goal is to try to extract everything we can from the oscillations, and find out how we can obtain them, and how the various neural parameters affect them. We can discuss oscillations in the activity (c), neural weight (m) or the threshold function (θ).

The first characteristic we can extract is the equilibrium value of the activity, given a constant input of d.

$$
\begin{aligned}
d &\equiv d_o \text{ (constant)} \\
c \equiv c_o \text{ (constant)} &\Rightarrow \dot{m} = 0 \\
&\Rightarrow c = \theta = c_o = \frac{1}{\tau}\int_{-\infty}^{t} c_o^2 e^{-(t-t')/\tau}dt' = c_o^2 \\
&\Rightarrow c_o = 1
\end{aligned}
$$

We can do the same type of analysis with a periodic d, say, with a value of 1 every $1/\beta$ iterations (thus with a frequency of β). c will no longer be a constant, but $\overline{c}$ will be.

$$\overline{d} \equiv \beta d_o\text{(constant)}$$

$$
\begin{aligned}
\overline{c} \equiv c_o(\text{constant}) \quad &\Rightarrow \quad \overline{\dot{m}} = 0 \\
&\Rightarrow \quad \overline{c} = \overline{\theta} = c_o = \frac{\beta}{\tau}\int_{-\infty}^{t} c_o^2 e^{-(t-t')/\tau} dt' = \beta c_o^2 \\
&\Rightarrow \quad c_o = \frac{1}{\beta}
\end{aligned}
$$

1.4 Sinusodal Behavior of the Neuron

When one looks at the differential equations we are using, it is quite apparant that we wouldn't expect a pure sinusoidal oscillation of any of the parameters in any but the simplest situations. From the simulation results, which will be explained later, one does see in parts of the phase space fairly simple looking oscillatory behavior. This motivates us to assume a sinusoidal perturbation and see what restraints this places on the parameters, ie. under what circumstances should we expect to see sinusoidal behavior.

We start with

$$
\begin{aligned}
c &= md \\
\dot{m} &= \eta c(c-\theta)d \\
\theta &= \frac{1}{\tau}\int_{-\infty}^{t} c^2(t') e^{-(t-t')/\tau} dt'
\end{aligned}
$$

and assume

$$c = c_o + c_1 \sin \omega t$$

where $c_1 \ll 1$.

It follows that

$$
\begin{aligned}
\dot{m} &= \eta(c_o + c_1 \sin\omega t)(c_o + c_1 \sin \omega t - \theta)d && (1.9)\\
\dot{c} &= c_1 \omega \cos\omega t = \dot{m}d && (1.10)\\
\theta &= \frac{1}{\tau}\int_{-\infty}^{t} (c_o + c_1 \sin\omega t')^2 e^{-(t-t')/\tau} dt' \\
&\overset{c_1 \ll 1}{\approx} \frac{1}{\tau}\int_{-\infty}^{t} (c_o^2 + 2c_o c_1 \sin \omega t') e^{-(t-t')/\tau} dt' \\
&= c_o^2 + 2c_o c_1 \frac{1}{\tau} e^{-t/\tau}\left[\frac{e^{t'/\tau}(\frac{1}{\tau}\sin\omega t' - \omega\cos\omega t')}{\frac{1}{\tau^2}+\omega^2}\right]\Bigg|_{-\infty}^{t} \\
&= c_o^2 + \frac{2c_o c_1/\tau}{\frac{1}{\tau^2}+\omega^2}(\frac{1}{\tau}\sin\omega t - \omega \cos \omega t)
\end{aligned}
$$

From Equation 1.9,

$$\dot{m} = \eta(c_o + c_1 \sin\omega t)(c_o + c_1 \sin\omega t - c_o^2 - \frac{2c_o c_1/\tau}{\frac{1}{\tau^2}+\omega^2}(\frac{1}{\tau}\sin\omega t - \omega\cos\omega t))d$$

and combining this with Equation 1.10 one obtains

$$c_1 \omega \cos\omega t \overset{c_1 \ll 1}{\approx} d^2 \eta(c_o^2 + c_o c_1 sin\omega t - c_o^3 - \frac{2c_o^2 c_1/\tau}{\frac{1}{\tau^2}+\omega^2}(\frac{1}{\tau}\sin\omega t - \omega \cos\omega t) + c_1 c_o \sin\omega t - c_o^2 c_1 \sin\omega t)$$

We now set $c_o = 1$ ($\beta = 1$), match sine and cosine terms, and obtain a final solution.

$$\left(\frac{2d^2\eta c_1/\tau^2}{\frac{1}{\tau^2} + \omega^2} - d^2\eta c_1 \right) \sin \omega t = \left(\frac{2d^2\eta c_1 \omega/\tau}{\frac{1}{\tau^2} + \omega^2} - c_1\omega \right) \cos \omega t$$

$$(\text{sine term}) \quad \frac{2}{\tau^2} = \frac{1}{\tau^2} + \omega^2 \quad \Rightarrow \quad \omega^2 = \frac{1}{\tau^2} \tag{1.11}$$

$$(\text{cosine term}) \quad \frac{2d^2\eta}{\tau} = \frac{1}{\tau^2} + \omega^2 \quad \Rightarrow \quad \frac{1}{\tau} = d^2\eta \tag{1.12}$$

Equation 1.11 implies that whenever we get *simple* oscillations, the frequency of those oscillations depends only on the memory constant. This simple relationship is a deceiving, because Equation 1.12 puts a strong restriction on the memory constant and learning rate in order to achive sinusoidal oscillations. This does, however, give us a region of phase space around which to look for tractible results. As we deviate from this, we expect more complicated, non-linear behavior.

2 The Simulations

In the simulations, the computer steps through time, iterating Equations 1.5-1.7. Time will be given in iterations, on the order of 10^7, and the paramters $\frac{1}{\tau}$ and η will be on the order of 10^{-5}. This ensures that $\frac{\Delta t}{\tau} \ll 1$, and validates the assumptions of the theory.

2.1 Exploring the Sinusoidal Region

An example *simple* oscillation is shown in Figure 5. We'd like to look at the frequency and the amplitude of the oscillations.

The first thing we'd like to check is how close to sinusoidal the oscillations are, given the parameters restricted by Equations 1.11 and 1.12. Notice that we will have to look at the long time behavior, because it takes a while for the system to settle into regular oscillitory behavior. Figures 6-10 display the oscillations for several values of the input strength. They compare well to the fitted sine wave of the frequency predicted by Equation 1.11. Here the frequency is chosen to be equal to the memory constant, and the amplitude and phase are chosen to fit a single data point along the oscillation curve (the theory doesn't yet predict either of these).

One notices immediately that the *amplitude* gets smaller as time goes on. A question which then comes up is, does the oscillation amplitude approach a certain value? Figure 11 displays that the oscillations die off, regardless of the conditions, but that for smaller input it takes a longer time.

2.2 Changing the Input Frequency

In Section 1.3 we found that, if we made the input frequency some fraction of iterations β, then the fixed point about which the activity oscillates should be $c = 1/\beta$. This is clearly verified in Figure 12.

2.3 Breaking the Sinusoidal Barrier

One doesn't have to wander far to get extremely non-sinusoidal behavior from the neuron. One does expect this, because of the non-linearities in the differential equations. Unfortunately analytic approaches to the problem either end up with contradictions, if the assumptions are too simple, or just intractable algebra if the assumptions are more complicated. The best we can hope for at the moment is an observation of the regions of phase space beyond the sinusoidal region, and an attempt at some hand-waving explanations of the behavior seen.

2.3.1 Amplitude

A quick glance at Figure 13 makes it clear that the neuron is quite sensitive to changes in the memory constant. It will be useful to think of the memory constant τ as the number of iterations *remembered*. By the term "remembered" here, I mean it is the number of iterations which go into the calculation of θ (see Equation 1.4) before the exponent $e^{-t/\tau}$ cuts off significantly.

If the neuron *remembers* too much, then the old behavior is weighted more heavily and the behavior will be sluggish and more prone to large oscillations. If the neuron remembers very little, then the oscillations will tend to die out. This is seen in Figure 13. As τ (memory) decreases, the oscillitory behavior dies out. Figure 14 show how the amplitude of the oscillations increases as τ increases. At the point where $\frac{1}{\tau} = d^2\eta$ we've seen that the oscillations die out very slowly, and smaller values of τ damp the oscillations out fairly quickly.

2.3.2 Frequency

Another thing we might notice from Figure 13 is that the frequency of the oscillations is dependent on τ. It won't do us any good to look at $\tau < d^2\eta$, so we will focus our attention on the non-damped oscillations. One quickly finds that the shape of the oscillations is not at all sinusoidal as one heads away from the simple region. The oscillations tend to be sharply peaked on the rising side, and then quite flat on the falling side, as seen in the extreme case of Figure 15.

We find it immediately difficult to obtain the frequency of the oscillations, given that the pure sinusoidal behavior is lost, and the theory is intractable. One can, however, do a numerical Fourier transform on the data, and see which frequencies are dominant. For a comparison, Figure 16 shows the power spectrum for the simple oscillations talked about earlier. The peaks in the spectrum fall at exactly the position predicted by the theory.

As we move away from the sinusoidal region, changing τ to be larger, we expect the frequency to decrease. We expect this because the neuron becomes more sluggish for larger τ. Figure 17 shows this decline clearly. Eventually the frequency becomes so small that we cannot reasonably do a Fourier transform on the data set. There seems to be a nice linear relationship here between the memory constant and the frequency. I'm unsure about what this implies.

When we look at the same memory constant dependance of the frequency, but this time with a different *input* frequency β we see *absolutely no change* in the behavior of the neuron. The dominant frequencies are identical to the constant input case.

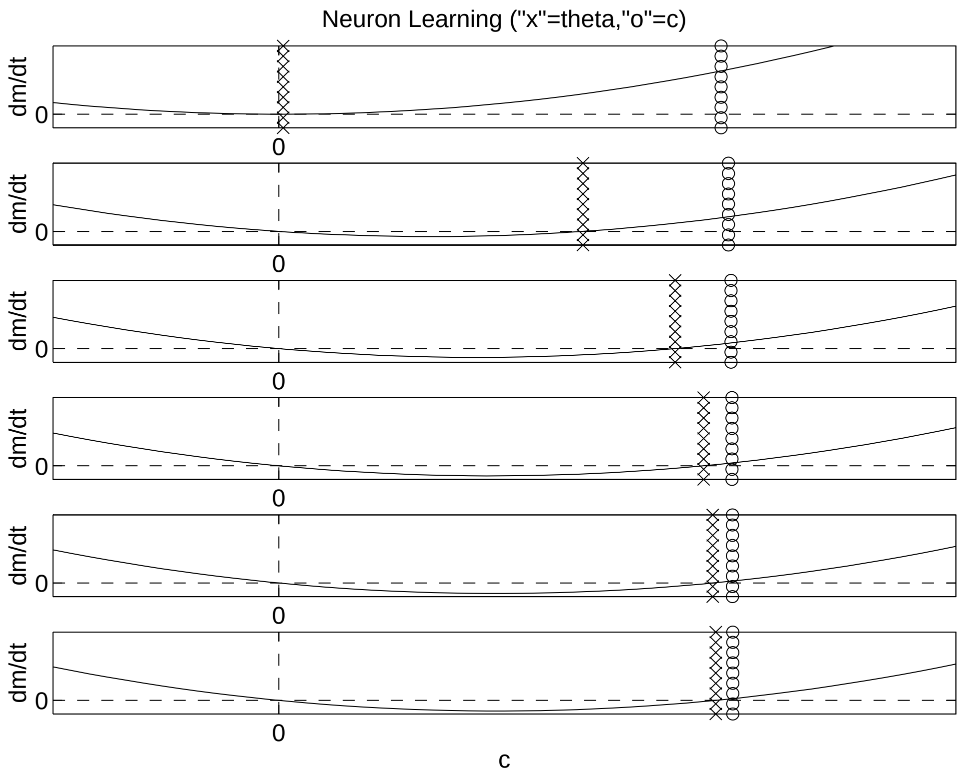

Figure 3: A Neuron Becoming Selective

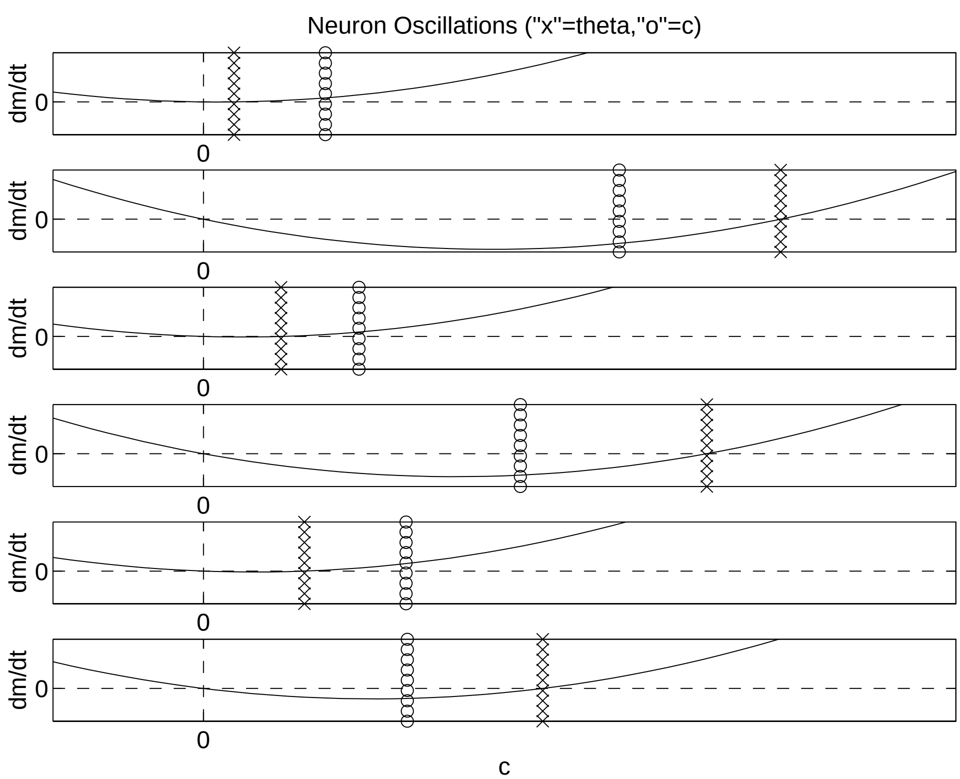

Figure 4: A Neuron Experiencing Oscillations

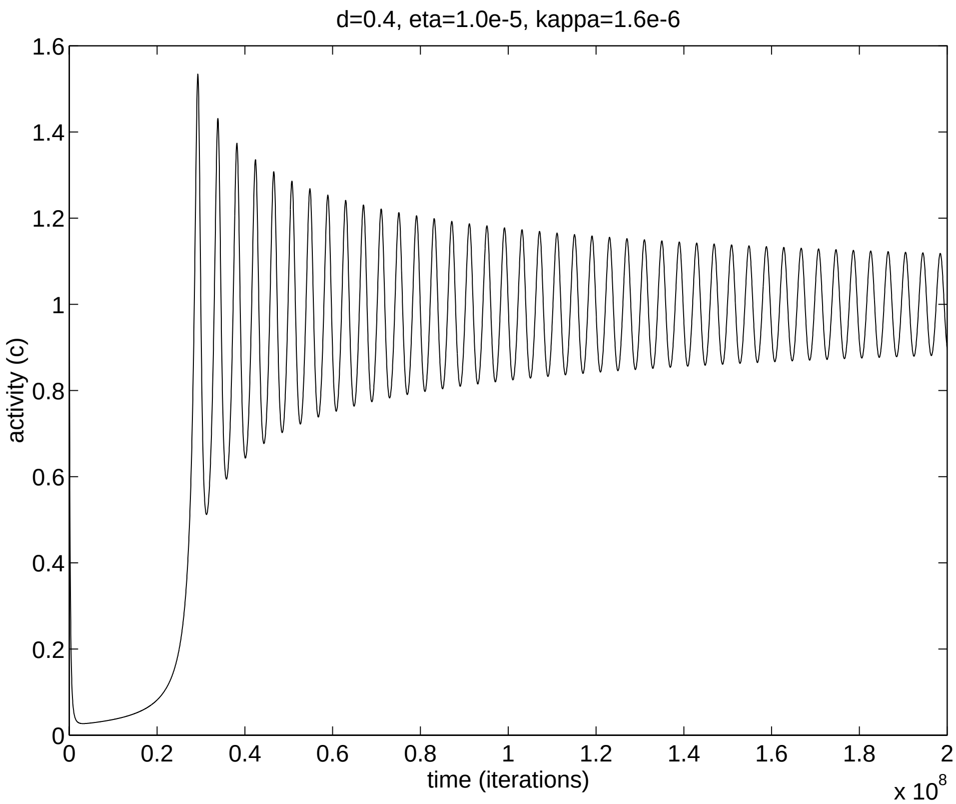

Figure 5: Example Neuron Oscillation

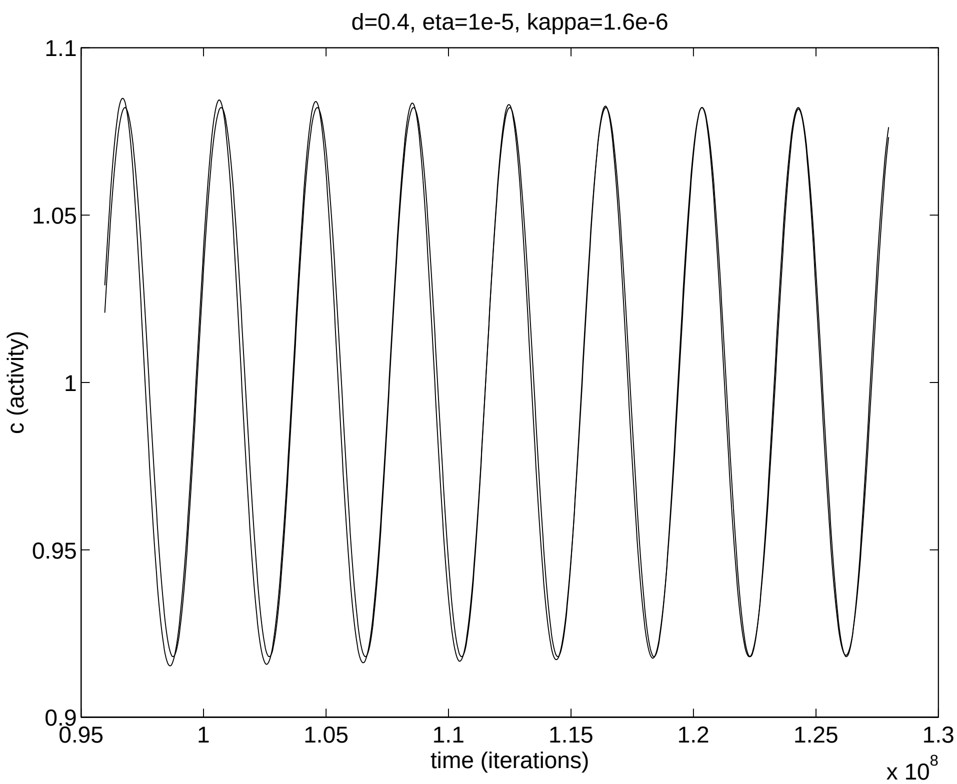

Figure 6: Oscillations compared to Sine curve

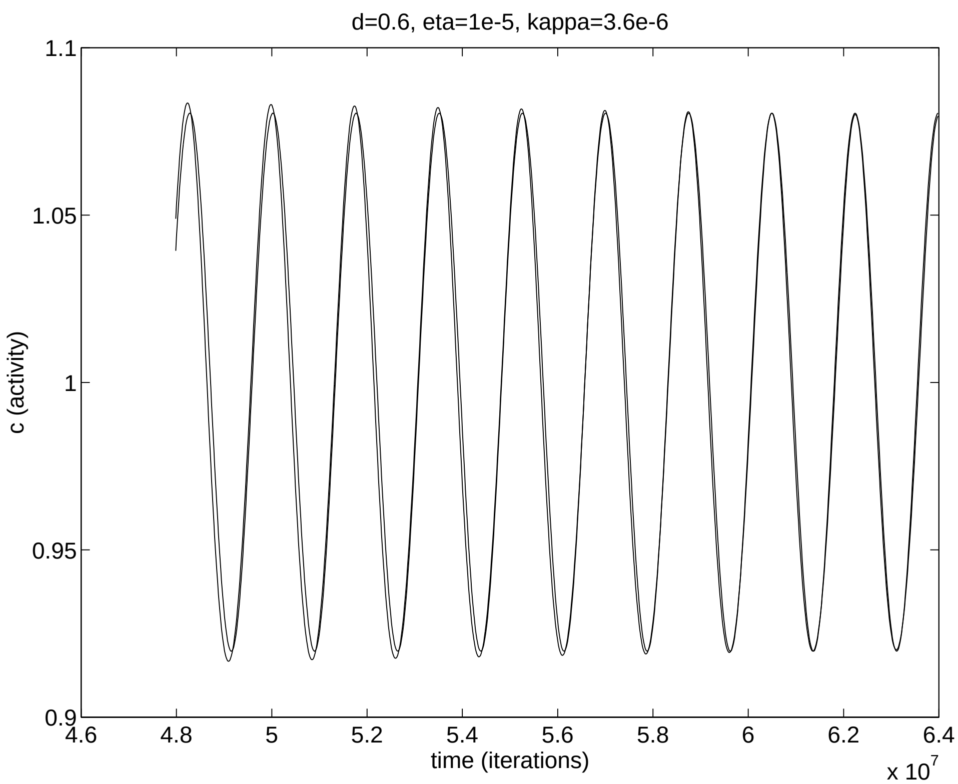

Figure 7: Oscillations compared to Sine curve

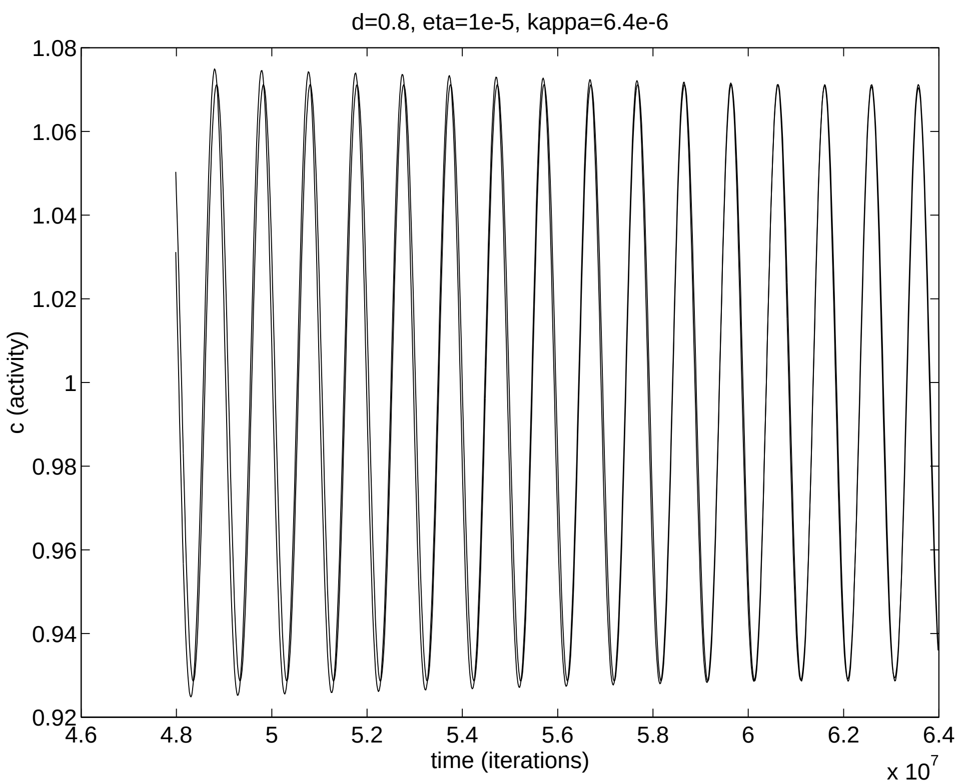

Figure 8: Oscillations compared to Sine curve

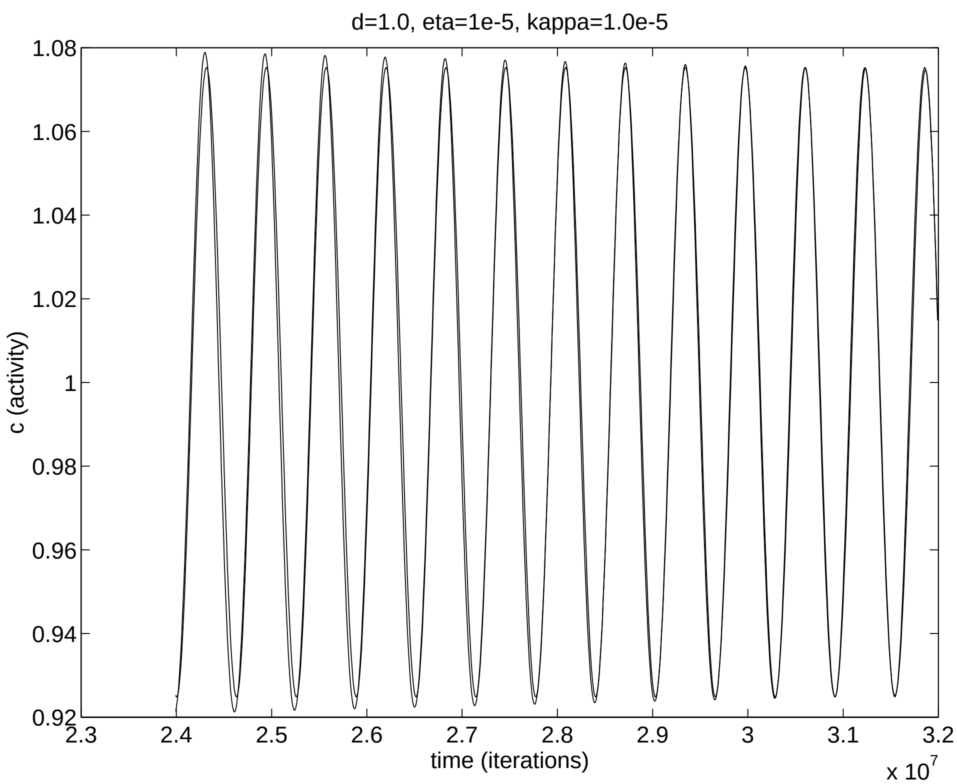

Figure 9: Oscillations compared to Sine curve

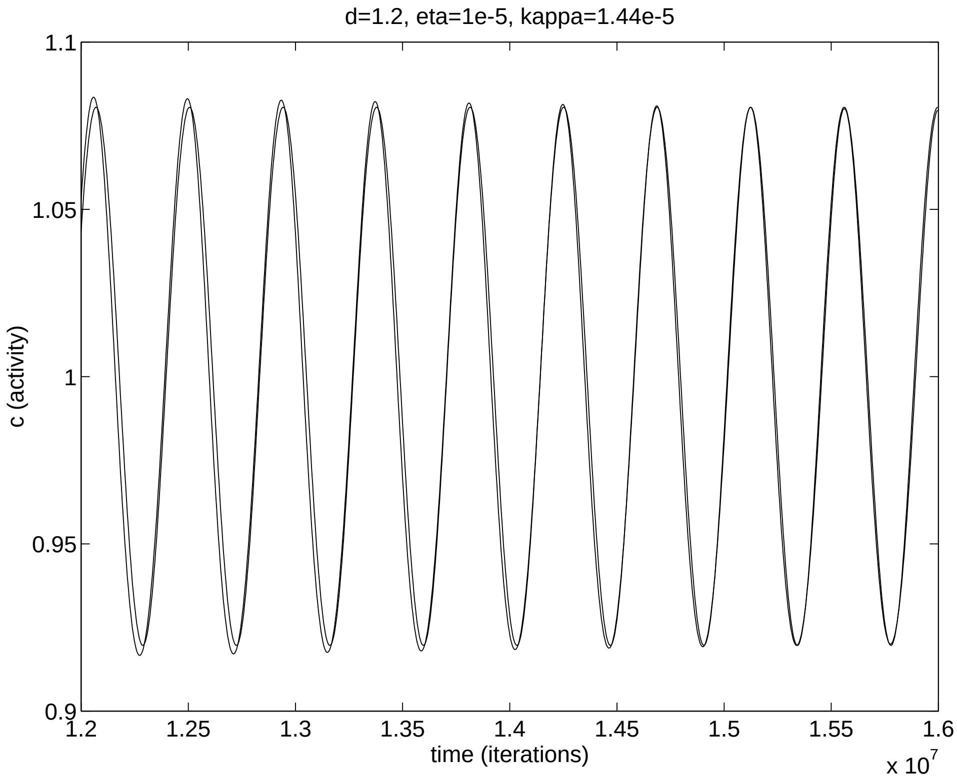

Figure 10: Oscillations compared to Sine curve

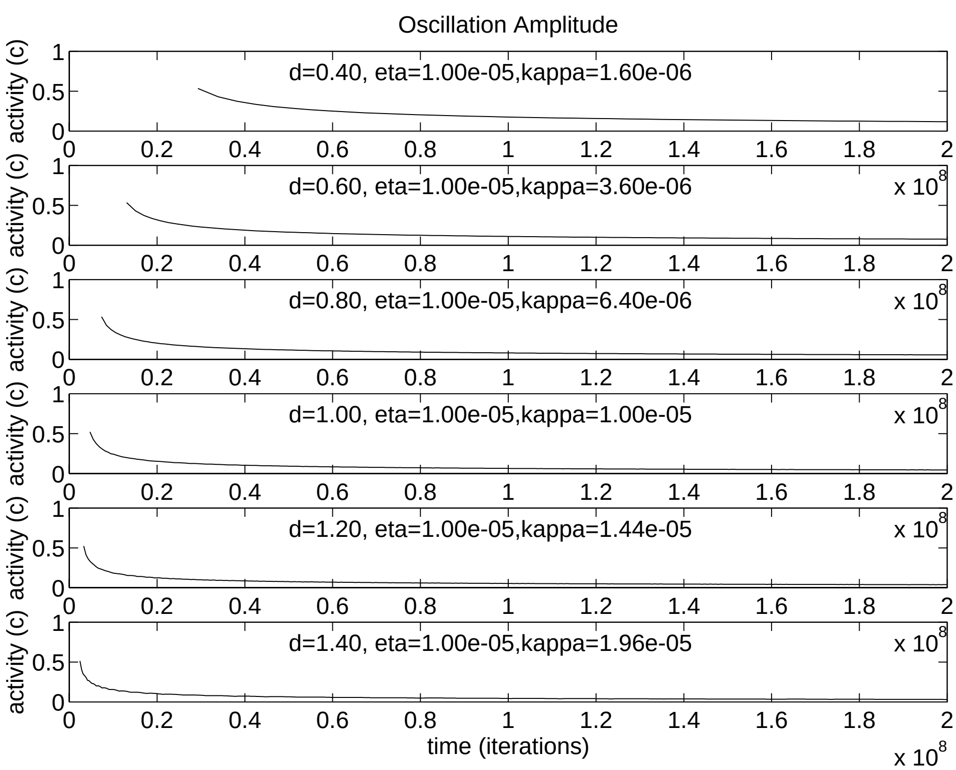

Figure 11: Oscillation Amplitude

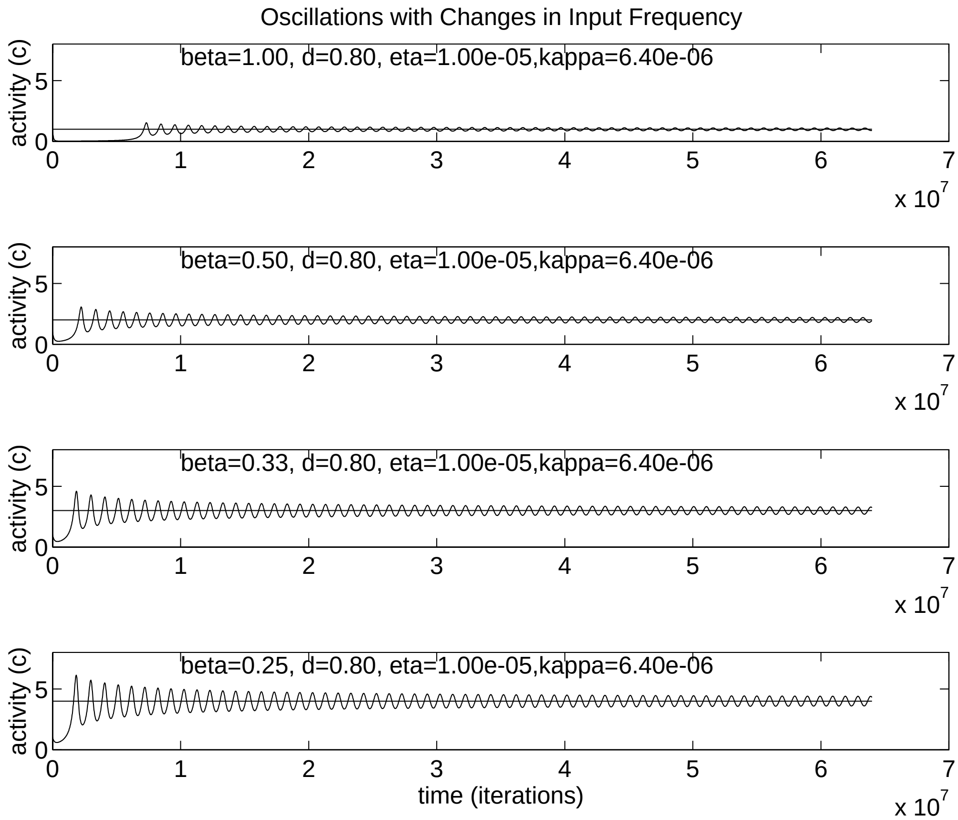

Figure 12: Fixed Point of Oscillation compared with β

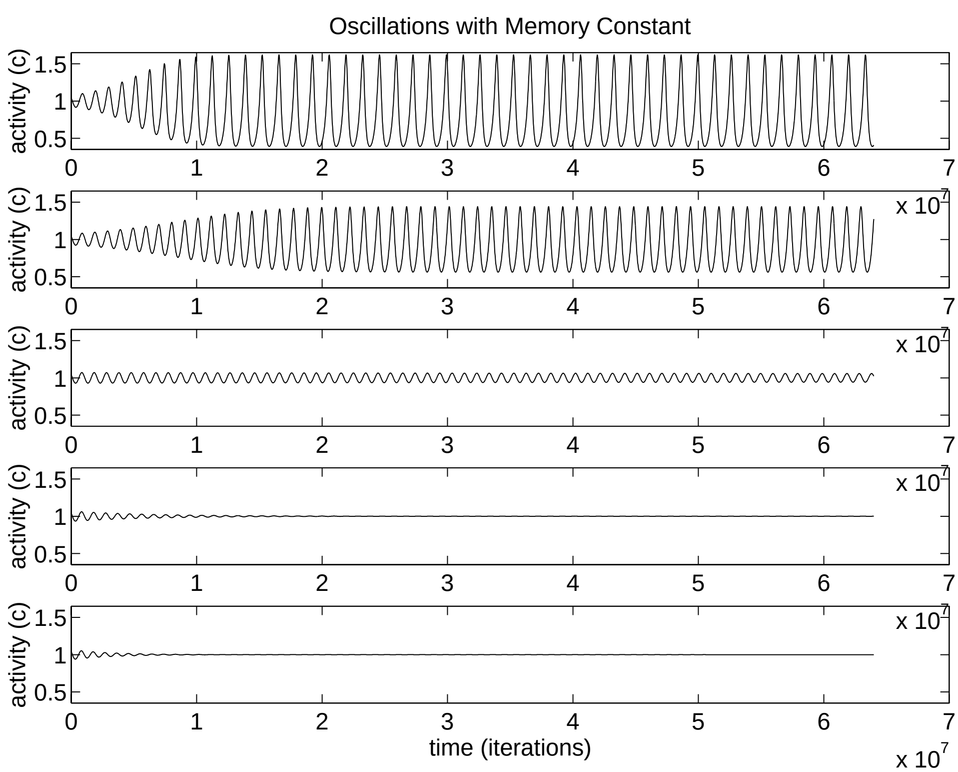

Figure 13: Changing the Memory Constant

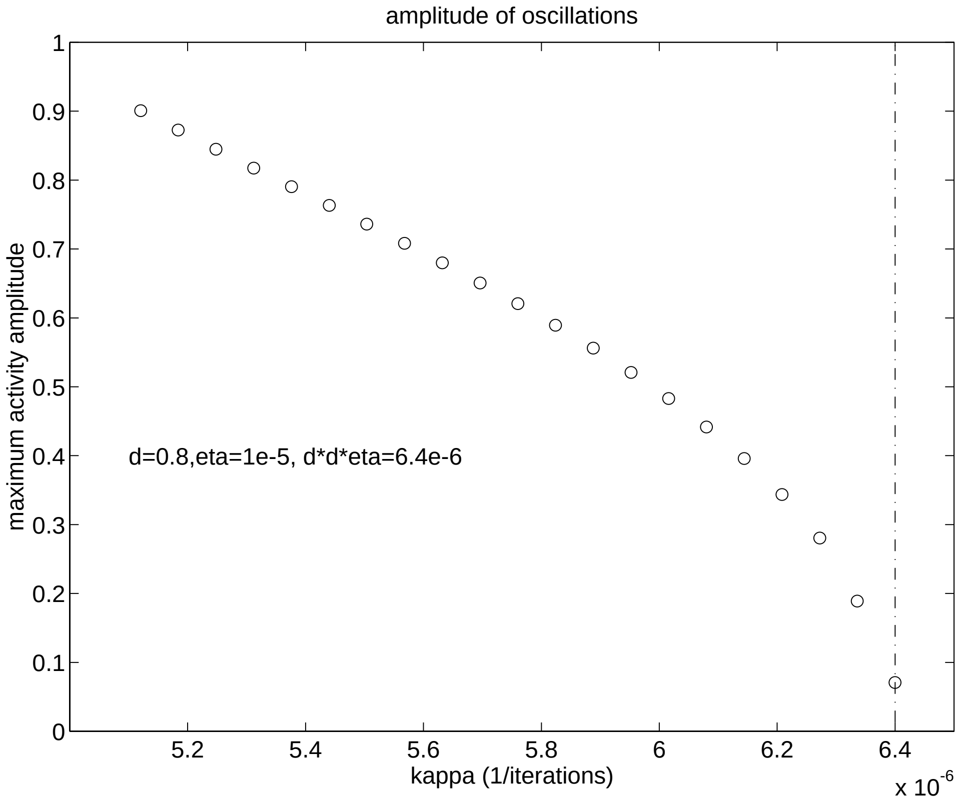

Figure 14: Amplitude of Oscillation versus Memory

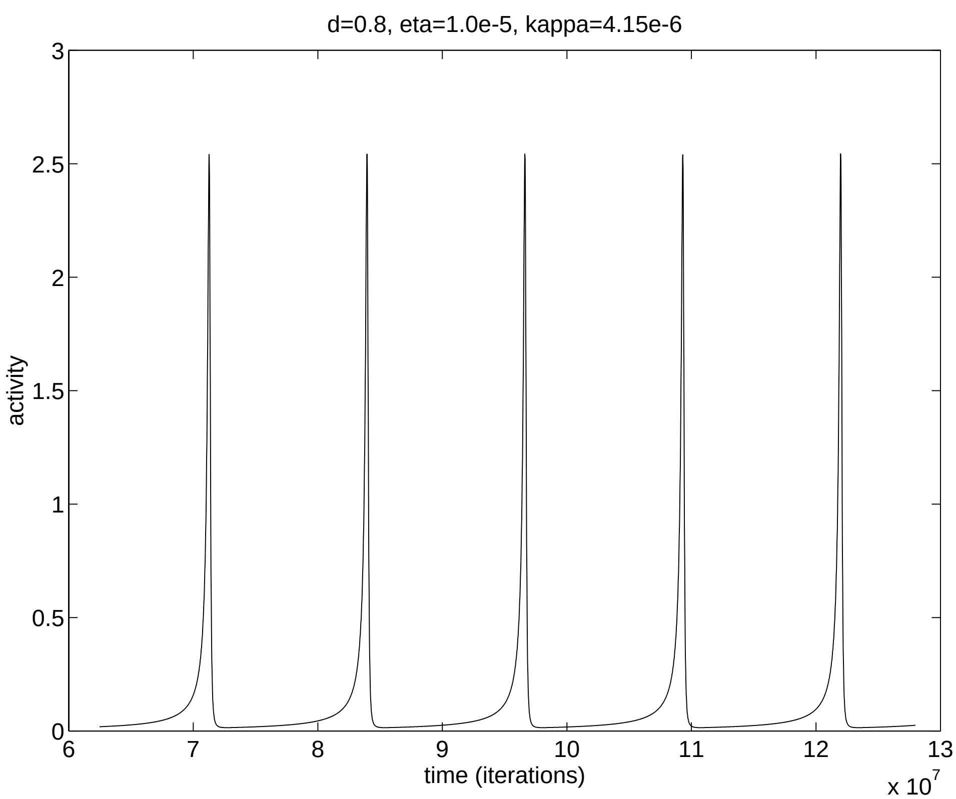

Figure 15: Extremely Non-linear Oscillations

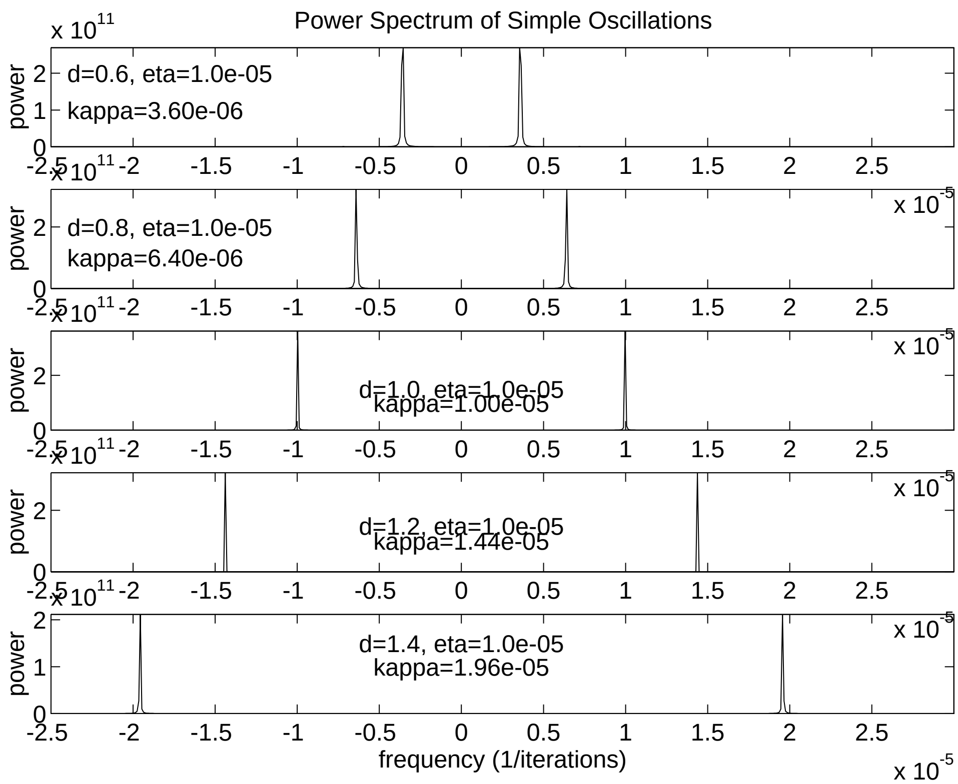

Figure 16: Power Spectrum of Simple Oscillations

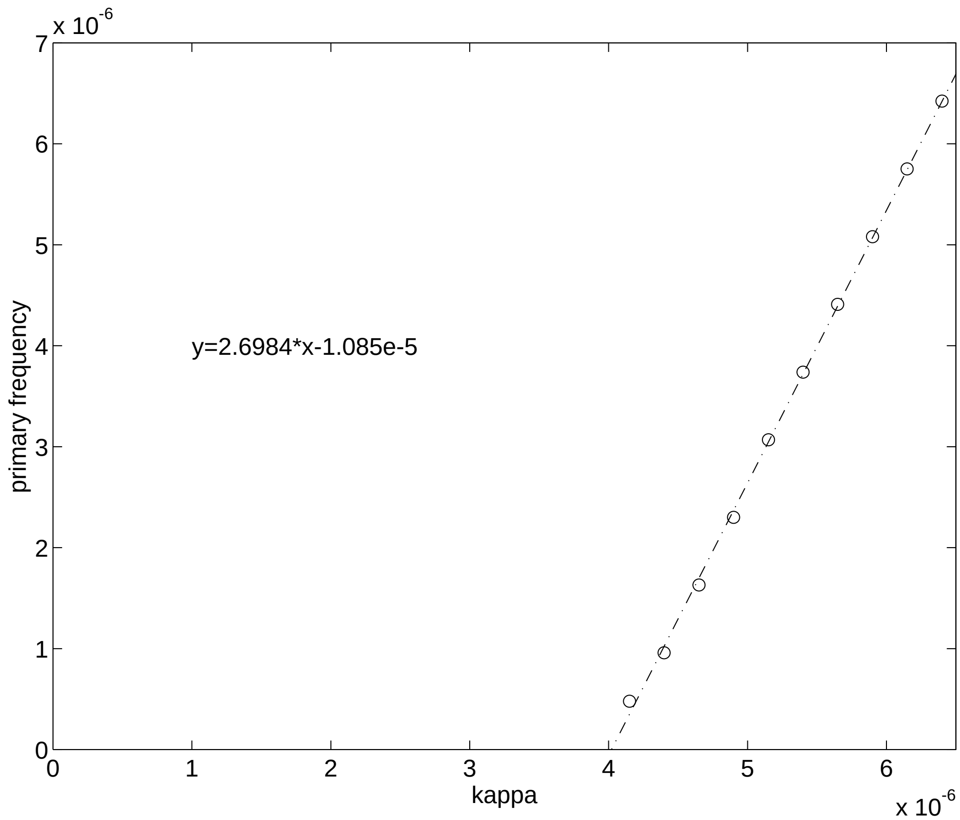

Figure 17: Memory Dependance of the Oscillation Frequency