Exact Ensemble Dynamics for Spike-Timing-Dependent Plasticity

Frank Alan Adrian
B.S., Computer Engineering, University of Illinois at Champaign-Urbana (1978)

Presented to the Division of Biomedical Computer Science within
The Department of Science & Engineering
and the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree of Master of Science
in
Computer Science & Engineering

October 2008

Department of Science & Engineering
School of Medicine
Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is certify that the Master's thesis of

Frank Alan Adrian

has been approved

Dr. Todd K. Leen, Thesis Advisor
Professor

Dr. Patrick D. Roberts
Assistant Professor

Dr. Deniz Erdogmus
Assistant Professor

Dr. Xubo Song
Assistant Professor

Acknowlegements

This thesis has been a long time in the making. An undertaking of this length does not occur without the help of many people.

I wish to acknowledge the assistance of my thesis advisor, Dr. Todd K. Leen. His guidance and insight has been of extraordinary value to me throughout the process of researching and writing this thesis. Thanks, Todd, for your help and patience.

The companies that have employed me while I have been at OGI have provided flexibility for my time commitments and (occasionally) reimbursement. The managers that I worked for during this time have been supportive of my efforts. My thanks to Gary Klimowicz, Steve Scalpone, and Mark Klein at Corillian, Inc.; Gary Klimowicz at uXcomm, Inc.; and David Barashi at McAfee, Inc.

I would also like to acknowledge my good friend Dr. Axel Koenig. His encouragement helped see me through some days when I was about to call it quits and his explanations of certain statistical concepts helped me understand the subject much better than I would have otherwise.

The most sincere and loving acknowlegement must be given to my family – my wife, Sara Anne Karlin, my son Andrew, and my daughter Caroline. I am sure they had their own ideas about what I could have been doing rather than going to classes, researching, and writing this thesis. They have been extraordinarily patient, supportive, and understanding throughout this process. I love and thank them all.

Contents

List of Figures

Abstract

Exact Ensemble Dynamics for Spike-Timing-Dependent Plasticity

Frank Alan Adrian

Master of Science
Division of Biomedical Computer Science within
The Department of Science & Engineering
and the Oregon Health & Science University
School of Medicine

October 2008

Thesis Advisor: Dr. Todd K. Leen

Under the influence of input stimuli, synaptic strength changes based on the relative timing of pre- and postsynaptic events. This mechanism is called Spike-Timing-Dependent Plasticity (STDP) and is recognized as a basis of neural plasticity in biological systems. Changes in synaptic strength (or weight) are described mathematically using a stochastic learning rule that generates a Markov process over the weights. This process determines a master equation whose solution provides the time evolution of the probability density function for the weights. The master equation has an expansion in a perturbation-like series (the Kramers-Moyal expansion), which when truncated after the second term gives a Fokker-Planck equation. Solving this equation provides an approximation to the probability density.

Van Rossum et al. [1] use this approach to analytically predict the equilibrium distribution of synaptic weights governed by anti-symmetric spike-timing-dependent learning rules of the type observed by Bi and Poo [2]. However, the use of the Fokker-Planck equation is ill-advised and does not always lead to an accurate approximation, as was shown by Heskes and Kappen [3, 4] in the context of machine learning.

We show that if for all $k \leq K$, the k^{th} jump moment is a polynomial of order less than or equal to k in the weights, the Kramers-Moyal expansion pro-

duces a recurrence in the first K moments that can be solved in closed form. The model of van Rossum et al. [1] has this property and we find an exact solution for the the equilibrium moments of the probability density. Our simulations validate this result across a broad range of model parameters for the antisymmetric STDP model described in [1].

Chapter 1

Introduction

As a synapse receives stimuli in the form of presynaptic spikes and postsynaptic spikes occur, the strength of the synapse changes. The mechanism behind this strength change depends on the relative timing of the pre- and postsynaptic spikes, and not simply on the rates of input and output spikes as initially thought [5, 6, 7, etc.]. This mechanism, called Spike-Timing-Dependent Plasticity (STDP) [8], is currently accepted as a basis of synaptic plasticity in biological systems.

The change in the synaptic strengths for neurons can be described mathematically via a learning rule that relates the change in synaptic strength to the current synaptic strength and the inputs to the synapse. These changes are stochastic due to natural variations that occur even with specific inputs and due to the randomness of the inputs. Algorithms for machine learning systems use similar learning rules, except that the variations in the changes result only from randomness in the inputs presented to the system. The general forms of the learning rules are the same for both neurons and on-line learning algorithms and generate identical Markov processes over an ensemble of synaptic strengths (or weights). The learning rule determines a single time-step transition probability which, in turn, determines a master equation for the probability density function over the weights. This density carries all information about the learning dynamics of the system.

Unfortunately, for fixed learning rates (as is assumed for biological sys-

tems), the master equation is intractable and one must use approximation to find a solution. One approach starts by expanding the master equation in a power series giving the Kramers-Moyal expansion, an infinite order partial differential equation which describes changes in the probability density with time. When truncated after the second term, the expansion becomes a Fokker-Planck equation (FPE) which can be solved to approximate the probability density at equilibrium.

Van Rossum et al. [1] use this approach to analytically predict the equilibrium distribution of synaptic weights governed by temporally anti-symmetric spike-timing learning rules of the type observed by Bi and Poo [2]. They find a strongly skewed distribution that agrees well with Monte Carlo simulations.

Prior to its use in describing STDP, many authors used the FPE to describe the dynamics of machine learning algorithms. However, the use of the FPE to provide an approximation for the probability density is ill-advised, as shown by Heskes and Kappen [3, 4] who demonstrated that this approximation could be relied on only in limiting cases as the learning rule parameters approach zero, and retains validity only in the lowest order, linear effects.

Heskes' and Kappen's papers effectively halted the use of the FPE within the machine learning community, but their suggestion for an alternative method – that one should base the analysis of these systems on the fluctuation expansion of Van Kampen [9] – is incomplete. Heskes examines only the lowest order terms of the expansion, leading only to Gaussian equlibria. Van Kampen's text does not give a complete development for higher order approximations, applicable to skewed distributions generated by non-linear effects, precisely the kind of distributions found in experimental studies of STDP [10].

For Markov models where for all $k \leq K$, the k^{th} jump moment is a polynomial of order k or less in the weights, the Kramers-Moyal expansion produces a recurrence in the moments of the distribution that can be solved in closed form. Because these results are based on the complete Kramers-Moyal expansion, they are exact. Our main contribution is in showing that van Rossum's model, under reasonable assumptions, has this form and determining the exact values for the moments of the weight distribution for this model. We show that the moments solved for in this manner more closely match simulations over a broader

range of learning rule parameters than do those derived from the Fokker-Planck approximation.

Chapter 2 introduces STDP and the learning rule that governs the behavior of both neural ensembles and on-line learning algorithms. In Chapter 3 we use the learning rule to find the transition probability for a Markov process. Using this transition probability, we derive the master equation that describes the time evolution of the probability distribution arising from this process. We then introduce both the Kramers-Moyal expansion for the master equation and the Fokker-Planck equation used to find an approximate solution for the probability distribution. We finish by reviewing previous work using the Fokker-Planck equation in machine learning and biological systems.

Chapter 4 reviews the findings of van Rossum et al. [1] and summarizes the derivation of his mathematical STDP model based on the Fokker-Planck equation. In Chapter 5 we recap van Rossum's original solution based on the Fokker-Planck equation. In this model, van Rossum made a set of simplifying assumptions that were valid for the biophysical parameters studied, but which limit the validity of the model to a small segment of the parameter space. We give our own solution for the FPE that does not use all of van Rossum's assumptions, allowing its use over a wider range of model parameters. Even though this model is more accurate at both smaller and larger parameter values than those used by van Rossum, it still shows considerable error at larger values of the learning rule parameters. In Chapter 6 we review Heskes and Kappen's [3, 4] findings that explain why the solution of the FPE might not provide an accurate approximation to the actual distribution generated by the master equation or its Kramers-Moyal expansion.

Chapter 7 shows how the full Kramers-Moyal expansion can provide an exact solution for the moments of a master equation and characterize the family of models for which this solution is possible. As an example, we provide an exact solution for the moments of the weight density for the model of van Rossum et al. and examine limits on moment validity based on the forms of these moments. In Chapter 8, we analyze the Kramers-Moyal expansion truncated at the second term, giving exact moment equations for the Fokker-Planck approximation to this model. In Chapter 9, we validate our solution against Monte Carlo Markov Chain

simulations. We confirm that the Fokker-Planck approximation overestimates moment values (sometimes grossly) and examine the relative error between the two approaches. Chapter 10 concludes the paper with a discussion of the results and alternative methods based on the full Kramers-Moyal expansion that might be used to solve models that do not have exact solutions for the moments.

Chapter 2

Spike-Timing-Dependent Plasticity

2.1 Neural Plasticity

In biological neural systems, synapses change their strength dynamically in response to input stimuli. This process is called *synaptic plasticity*. In 1949, Hebb [5] proposed a model for these weight changes which is now known as *Hebbian learning*:

> "Let us assume that the persistence or repetition of a reverberatory activity (or 'trace') tends to induce lasting cellular changes that add to its stability... When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

In the years following Hebb's proposal, many biophysical mechanisms for synaptic strength change were explored. Early studies focused only on the frequency of input and output spike trains [5, 6, 7, for example]. In particular, it was shown that repetitive electrical activity was able to cause persistent increases or decreases in synaptic strength, commonly termed long-term potentiation (LTP) and long-term depression (LTD), respectively. However, starting

in the early 1980s, researchers also started studying how the interval between individual presynaptic and postsynaptic spikes influenced changes in synaptic strength. In 1983, Levy and Steward [11] looked at the effects of relative timing in pre- and postsynaptic action potentials, finding that the order of neuronal activation was critical for synaptic strength change. Markram [12] in 1997 demonstrated that synaptic strength increased when a presynaptic neuron was activated 10 msec before the postsynaptic neuron was activated and that synaptic strength decreased when a postsynaptic neuron was activated 10 mS prior to the presynaptic neuron. In 1987, Bell [13] showed that proper timing of pre- and post-synaptic spikes was necessary for anti-Hebbian learning in the Purkinje-like cells of the mormyrid electric fish, and that for this type of cell, the learning window for LTD was 60 msec. Markram [12] in 1997 demonstrated that synaptic strength increased when a presynaptic neuron was activated 10 msec before the postsynaptic neuron was activated and that synaptic strength decreased when a postsynaptic neuron was activated 10 mS prior to the presynaptic neuron. In 1998, Bi and Poo [2] explored potentiation and depression of neural strength in cultured rat hippocampal cells and found that the window for synaptic strength change was 5-40 msec and that, if the neural activations occurred outside this time window, synaptic strength was essentially unchanged. This model of systematic potentiation and depression of neural strength based on the relative timing of pre- and postsynaptic spikes, is called *Spike-timing Dependent Plasticity* (STDP).

In recent years further studies have shown that plasticity dependent on relative timing of pre- and postsynaptic spikes occurs in a variety of neuron types across many different biological organisms, but that the particulars of the changes vary widely. For example, variations occur in the time window within which the correlated pre- and postsynaptic spikes must happen to cause changes in synaptic strength. Likewise, whether the presynaptic spike preceding the postsynaptic spike will cause LTP or LTD (Hebbian vs. *anti-Hebbian* learning) also varies, as does the dependence of the magnitude of LTP and LTD on initial synaptic strength. All of these variations in the synaptic strength change can be mathematically described using a *learning rule* that describes how the synaptic strength changes based on the current synaptic strength and the relative timing of pre- and postsynaptic spikes. Roberts and Bell [8] provide a fine overview of STDP and survey the various kinds of learning rules that have been found so far.

2.2 Stochastic Modeling of Neural Plasticity

Because synaptic plasticity is biologically based, there are variations in changes in synaptic strength even when all inputs are the same. Additionally, one cannot usually predict the time interval between pre- and postsynaptic activations at any particular synapse, placing another impediment to our ability to predict what will happen at a particular synapse. Due to these factors, the variation in strength changes is best captured via a stochastic learning rule and it is often more interesting to study an *ensemble* of neurons and concern ourselves primarily with the probability distribution of the synaptic strengths over the ensemble.

We mathematically model the effect of STDP on an ensemble of neurons using the equation

$$w(n+1) = w(n) + \eta(n)\ L(w(n), x(n), v(n))\ \ . \tag{2.1}$$

In this equation, $w(n)$ represents the vector of synaptic strengths at the n^{th} pairing of pre- and postsynaptic spikes and $x(n)$ is the inter-arrival time between the pre- and postsynaptic spikes at this pairing. The variation in the strength change even when all other inputs are equal is represented by the stochastic variable $v(n)$, and $\eta(n)$ is a parameter that controls the learning rate. $L(w, x, v)$ is the learning rule that describes the change in synaptic strength at a particular time having the current strength and spike timing. If we combine all sources of uncertainty in the system (i.e., the effect of $v(n)$ and the variation in spike inter-arrival times), we obtain a stochastic model for the changes that occur in the ensemble of synaptic strengths

$$w(n+1) = w(n) + \eta(n)\ L(w(n), \xi(n))\ \ . \tag{2.2}$$

This model generates a Markov process over the space of synaptic strengths.

2.3 Stochastic Models of Machine Learning

Machine learning systems using *on-line* and *stochastic approximation* algorithms have similar models. These systems re-estimate model parameters (or, in the

case of artificial neural networks, change the *synaptic weights*), based on one (or perhaps a small number) example datum $\xi(n)$ at each iteration during the training period [14]. The weight changes at each trial are described mathematically using a learning rule having the same form as (2.2), but now $L(w, \xi)$ specifies the change in system parameters when the example $\xi(n)$ is presented as a training input. When the samples are chosen at random (with replacement) from a set of exemplars or from a parent distribution, ξ is a random variable. This general form for the learning rule (2.2) is the same for both biological and synthetic systems and generates the same Markov process in both.

The Markov process drives the time evolution of the probability distribution of the weights $P(w, n)$ which, in turn, completely determines the learning dynamics of these systems. If we assume that the learning rate $\eta(n)$ is held constant (as is normal for both biological and on-line computational systems), convergence results are limited in principle to convergence of the distribution [15, 16, 17]. The asymptotic distributions are highly dependent on the form of the function $L(w, \xi)$ and finding a closed-form solution for P is usually impossible.

Chapter 3

The Master Equation and Its Solution

A *master equation* is a differential (or difference) equation that describes the time evolution of a probability distribution over a set of states. The Markov process generated by (2.2) can be used to determine a master equation for the probability distribution $P(w,t)$. This probability distribution contains all the information about the changes in the weights w – the model parameters in machine learning or the strength of synapses in brains.

A random walk equation gives the probability for the system to be in a particular state $p_n(w)$ after n iterations of the learning rule (i.e., n spike pairings on a synapse or n training iterations in machine learning)

$$p_{n+1}(w) \;=\; \int dw'\, W(\, w' \to w \,|\, \xi; n)\, p_n(w') \quad . \tag{3.1}$$

Here, $W(\, w' \to w \,|\, \xi; n)$ represents the single-time-step transition probability for the weight to change from an old weight w' to a new weight w conditioned on ξ at iteration n.

By fixing a particular value of ξ, we can define the transition probability as

$$W(\, w' \to w \,|\, \xi; n) \;=\; \delta\left(w - [w' \;+\; \eta(n)\, L(w', \xi)]\,\right) \quad . \tag{3.2}$$

The variable ξ is stochastic, and so to obtain a single-time-step transition proba-

bility without condition on ξ we average with respect to the density $p(\xi)$[1] giving

$$W(\, w' \to w \,;\, n) \;=\; \int \delta\,(w - [w' \;+\; \eta(n)\, L(w,\xi)]\,)\;\, p(\xi)\, d\xi \;\; . \qquad (3.3)$$

The probability that the ensemble will have value w after n iterations of the learning rule $P(w,n)$ follows from (3.1) as

$$\begin{aligned} P(w,n+1) \;-\; P(w,n) \;\;&=\;\; \int W(w' \to w; n)\; P(w',n)\, dw' \\ &\quad - \int W(w \to w'; n)\; P(w',n)\, dw' \;\; . \end{aligned} \qquad (3.4)$$

In this equation, the first integral term corresponds to changes that enter state w and the second integral term corresponds to changes that leave state w.

In order to transition from a discrete time difference equation to a continuous time differential equation, we arbitrarily choose the time of the iteration steps n and define

$$t_{n+1} \equiv t_n + \Delta t \;\; .$$

If we choose time intervals Δt from the distribution

$$\wp(\Delta t) = \delta(\Delta t - \tau)$$

where $\delta(\cdot)$ denotes the Dirac delta function, the probability $P(w,t)$ that the ensemble is in the state w at time t is given by

$$\begin{aligned} P(w,t+\tau) \;-\; P(w,t) \;\;&=\;\; \int W(w' \to w, t)\; P(w',t)\, dw' \\ &\quad - \int W(w \to w', t)\; P(w',t)\, dw' \end{aligned} \qquad (3.5)$$

which is the equation (3.4) recast in terms of discrete time rather than iterations. Bedeaux et al. [18] describes the relationship between the the *discrete time* random walk (3.5) and the *continuous time* master equation (3.6) and shows that, at long times t, their solutions approach each other. Choosing a Poisson distribution

[1]The density on ξ can depend on w and on n. For notational convenience, we will not explicitly show this dependency.

for Δt, he shows that the the distribution for the time intervals is

$$\wp(\Delta t) = \tau \exp\left[\frac{-\Delta t}{\tau}\right]$$

and the random-walk equation (3.1) transforms into the continuous time master equation

$$\begin{aligned} \tau\, \partial_t P(w,t) &= \int W(w' \to w, t)\; P(w', t)\, dw' \\ &\quad - \int W(w \to w', t)\; P(w, t)\, dw \quad . \end{aligned} \tag{3.6}$$

If we know the initial density $P(w, 0)$, the master equation (3.6) together with the specific transition probability for the learning rule under study (3.3) allows us to determine the probability density $P(w, t)$ for all time. These equations are exact for all time t and learning rates η.

3.1 Solutions and Approximations

For biological and on-line systems, the learning rate $\eta(n)$ is assumed to be a constant, and finding an exact solution for the full master equation is usually impossible. As such, one must use approximative methods. One pathway to an approximate solution begins by expanding the transition probability (3.3) in a power series in the learning rate η. This provides a perturbation-like expansion of the master equation (3.6) called the *Kramers-Moyal* (KM) *expansion* [19]

$$\frac{\partial P(w,t)}{\partial t} = \sum_{j=1}^{\infty} \frac{(-1)^j}{j!}\, \eta^j\, \frac{\partial}{\partial w^j} \left(\alpha_j(w)\; P(w,t) \right) \tag{3.7}$$

where $\alpha_j(w)$ denotes the *jump moments*

$$\alpha_j(w) = \int L^j(w,\xi)\, p(\xi)\, d\xi \equiv E_\xi \left[L^j(w,\xi) \right] \quad . \tag{3.8}$$

The time evolution of the probability density defined by both (3.7) and (3.8) is exact.

Assuming a small learning rate, we should be able to truncate the infinite series to provide approximations to the full KM series. If we retain the first two terms, the result is a *Fokker-Planck equation* (FPE)

$$\frac{\partial P(w,t)}{\partial t} \;=\; -\eta\,\frac{\partial}{\partial\, w}\,(\,\alpha_1(w)\,P(w,t)\,)\;+\;\frac{\eta^2}{2}\,\frac{\partial^2}{\partial\, w^2}\;(\,\alpha_2(w)\;P(w,t)\,)\quad. \tag{3.9}$$

This approximation retains stochastic effects caused by the *drift* and *diffusion* coefficients $\alpha_1(w)$ and $\alpha_2(w)$. When $\alpha_1(w)$ is constant or linear and $\alpha_2(w)$ is constant, this equation leads to an equilibrium distribution that is Gaussian. When $\alpha_1(w)$ is non-linear or $\alpha_2(w)$ is non-constant, a broadening and skewing of the equilibrium distribution results. This approximation is normally used with the understanding that it is supposed to hold only at small learning rates η, letting one ignore terms of order $\mathcal{O}(\eta^3)$ and higher in the Kramers-Moyal expansion (3.7).

In biological systems, it is often not possible to determine a value for η independent of the changes determined by the learning rule $L(\cdot)$. In these cases, the effect of the constant learning rate is assumed to be included in $L(\cdot)$ and, by Taylor-expanding the master equation about $(w\;-\;\Delta w)$, one obtains a Kramers-Moyal expansion equivalent to (3.7), but without the factors η^j

$$\frac{\partial P(w,t)}{\partial t} \;=\; \sum_{j=1}^{\infty}\frac{(-1)^j}{j!}\,\frac{\partial}{\partial w^j}\,(\,\alpha_j(w)\;P(w,t)\,)\quad. \tag{3.10}$$

This equation can also be truncated after the second term to produce a Fokker-Planck equation that does not mention η:

$$\frac{\partial P(w,t)}{\partial t} \;=\; -\frac{\partial}{\partial\, w}\,(\,\alpha_1(w)\,P(w,t)\,)\;+\;\frac{1}{2}\,\frac{\partial^2}{\partial\, w^2}\;(\,\alpha_2(w)\;P(w,t)\,)\quad. \tag{3.11}$$

When using this form of the Fokker-Planck approximation, the understanding that η must be small no longer holds. However, it is replaced by a more stringent requirement – that the magnitudes of the terms ommitted in from the Kramers-Moyal expansion be negligible. Whether or not this constraint holds in any particular case is heavily dependent on the functional form of both the jump moments and the probability distribution and can be difficult to determine. However, for biological systems, there is sometime no choice but to use this formulation.

Retaining more terms from the Kramers-Moyal expansion to increase accuracy of the approximation would seem to be a good idea. However, the Pawula theorem [20] requires either that the KM expansion be truncated after the first or second term or that it must retain all of the terms to obtain physically realistic solutions $P(w,t)$ that have non-negative density. As such, one has no more useful approximation than the FPE based on truncating the KM expansion.

3.2 The Fokker-Planck Equation in Machine Learning

In the early 1990s, researchers used the Fokker-Planck equation to describe the dynamics of machine learning algorithms. Radons [21, 22] used the FPE to describe learning in backpropagation. Orr and Leen [23] and Hansen [24] did the same. These studies successfully predicted the distribution of the time required to hop between basins of attraction. Basin hopping depends greatly on the non-parabolic character of the cost function which arise from non-linearities in the drift coefficient α_1. All of these studies make apparently successful use of the FPE.

Der [25] described the dynamics of Kohonen's self-organizing feature map (also discussed by Orr [23, 26]) using a FPE. Leen and Moody [27] used a FPE to find equilibrium densities for the LMS algorithm. However, their results agreed with Monte Carlo simulations only for extremely small learning rates where the equilibrium distribution became approximately Gaussian.

3.3 The Fokker-Planck Equation in Biological Learning

More recently, many theoretical neuroscientists have also used the Fokker-Planck equation to describe learning dynamics in systems modeled using spike-timing-dependent plasticity. Cateau and Fukai [28] studied learning rules similar to those observed in rat hippocampus CA1 neurons, and in medium ganglion cells in

mormyrid electro-sensory lateral line lobe. They numerically integrated a FPE, determining the equilibrium distribution of synaptic strengths for these neurons.

Kepecs et al. [29] analytically calculated equilibrium distributions produced by learning rules based on STDP using a FPE, and confirmed that Monte Carlo simulations had converged to equilibrium. The theoretical equilibrium distributions obtained by Masuda and Aihara [30], who used the FPE to analyze synaptic competition and the formation of functional clusters. Their results match Monte Carlo simulations well. Burkitt et al. [31] predicted equilibrium synaptic distributions and compared their analytic results with histograms from Monte Carlo simulations. As in the previously-discussed studies, the use of the FPE produced good agreement between the simulations and analytically-predicted distributions.

Another study was undertaken by van Rossum et al. [1]. They used an FPE to analytically predict the equilibrium distribution of synaptic weights governed by learning rules of the type observed by Bi and Poo [2]. Using an anti-symmetric spike-timing model, their FPE-based calculations gave a strongly skewed equilibrium distribution that agreed well with Monte Carlo simulations, and qualitatively agreed with the experimental quantal amplitude distribution observed in a pyramidal neuron [10]. We examine van Rossum's model and relate their findings as a prelude to our own.

Chapter 4

The STDP Model of van Rossum

Van Rossum et al. [1] solve a non-linear Fokker-Planck equation to find the equilibrium density for the antisymmetric STDP observed in hippocampal neurons. The model is based on observations reported by Bi and Poo [2], Markram et al. [12], and Zhang et al. [32]. The numerical values of model parameters for the model are derived from Bi and Poo's results [2].

In this Hebbian learning model, synapses are potentiated if the postsynaptic spike follows the synaptic transmission event, and depressed if the postsynpatic spike precedes the synaptic event. The conductance change due to the potentiation or depression drops off approximately exponentially with increasing time between the synaptic event and the post-synaptic spike. Experimental evidence [2, 33] also shows that the amount of potentiation decreases with increasing initial synaptic strength, while the amount of depression is independent of the initial synaptic strength. When examining Bi and Poo's data, van Rossum, et al. found that a multiplicative noise model best matched model variation in the synaptic changes from single event pairings. Combining all these effects, under potentiation synapses change by

$$w \to w + w_p(w, v, \delta t) = w + (c_p + v\, w)\, e^{-\delta t/\tau} \quad . \tag{4.1}$$

Under depression synapses change by

$$w \to w + w_d(w, v, \delta t) = w + (-c_d\, w + v\, w)\, e^{\delta t/\tau} \quad . \tag{4.2}$$

In these equations, c_p and c_d are positive constants controlling potentiation and depression respectively, v is a Gaussian random variable with mean zero and variance σ^2, δt is the time interval between the synaptic event and the post-synaptic spike (positive for post-synaptic spike following the synaptic event), and τ is a time-constant determining the window for synaptic plasticity.

The reader will notice that, in (4.1) and (4.2), the factor $\eta(n)$ found in (2.2) has disappeared. Van Rossum's model is one where the effect of the constant learning rate is included in the learning rule so, as van Rossum does, we will use an analysis that omits the ηs. As such, we will use the form of the Kramers-Moyal expansion given in (3.10) and the corresponding Fokker-Planck approximation (3.11). For more general discussions about the FPE, the ηs will reappear.

Considering a single neuron, we assume uncorrelated inputs with a Poisson distribution and at each increment of time dt, the neuron undergoes potentiation or depression with probability $p_p(w)$ and $p_d(w)$ respectively. Together with the expression for the synaptic changes under potentiation (4.1) and depression (4.2), these probabilities determine the transition probability for the Markov process that defines the weight dynamics for the model. The single-time-step transition probability (conditioned on v) for the weight to change from w to w' is

$$\begin{aligned} W(w \rightarrow w' \mid v, \delta t) \;=\; & p_p(w)\,\delta(w' - (w + w_p(v,\delta t))) \;+ \\ & p_d(w)\,\delta(w' - (w + w_d(v,\delta t))) \;+ \\ & (1 - p_p(w) - p_d(w))\,\delta(w' - w) \end{aligned} \tag{4.3}$$

where $\delta(\cdot)$ is the Dirac delta function.

To model the time dependence of plasticity in equations (4.1) and (4.2) van Rossum et al. ignore the exponential time dependency, instead using a rectangular windowing function set to unity for $|\delta t| < t_w$ and to zero for $|\delta t| \geq t_w$ where t_w is the time window within which learning is active. If δt is within the

time window, the potentiation and depression rules become

$$w \;\rightarrow\; w + w_p(v) \;=\; w + (c_p + v\,w) \tag{4.4}$$

$$w \;\rightarrow\; w + w_d(v) \;=\; w + (-c_d\,w + v\,w) \tag{4.5}$$

respectively, while if δt is outside the time window, no change in the weight occurs. This approximation leads to a model that allows a closed-form solution of the Fokker-Planck equation.

We combine the expressions for the potentiation (4.4) and depression (4.5) with the conditional transition probability in equation (4.4) and integrate over the distribution of the noise term v to obtain the unconditional transition probability

$$\begin{aligned} W(w \rightarrow w') &= p_p(w) \int \delta(w' - (w + w_p(w,v)\,)\,p(v)\,dv \;+ \\ &\qquad p_d(w) \int \delta(w' - (w + w_d(w,v)\,)\,p(v)\,dv \\ &= p_p(w)\,E_v[\,\delta(w' - (w + w_p(w,v))\,] \\ &\qquad + p_d(w)\,E_v[\,\delta(w' - (w + w_d(w,v))\,]\;. \end{aligned} \tag{4.6}$$

where $E_v[\cdot]$ denotes the expectation with respect to the density on the noise v. Note that unless the post-synaptic spike follows the synaptic event, the the probability of potentiation p_p is zero. It is also zero when the time difference between the events δt is larger than the active learning window width t_w. Similarly, the probability of depression p_d is zero unless the post-synaptic spike precedes the synaptic event, and is also zero when the time difference between the events is larger than t_w.

4.1 Jump Moments

The transition probability (4.6) determines the dynamics of the probability density $P(w,t)$ for the weight ensemble in accordance with the master equation (3.6). As seen in Chapter 2, the Kramers-Moyal expansion (3.7) provides an equivalent description of $P(w,t)$ in terms of the jump moments α_j (3.8). We obtain the jump

moments for this model by substituting the transition probability (4.6) into the definition in Equation (3.8) and expanding the resulting binomial expressions

$$\begin{aligned}
\alpha_n(w) &= \int W(w \to w')\,(w'-w)^n\,dw' \\
&= p_p(w)\,E_v[\,w_p(w,v)^n\,] \;+\; p_d(w)\,E_v[\,w_d(w,v)^n\,] \\
&= p_p(w)\,E_v\,[\,(c_p + v\,w)^n\,] \;+\; p_d(w)\,E_v\,[\,(v-c_d)^n\,w^n\,] \\
&= \sum_{j=0}^{n} \binom{n}{j} \left(\,p_p(w)\,c_p^{n-j}\,w^j \;+\; p_d(w)\,(-c_d)^{n-j}\,w^n\,\right)\,E_v[\,v^j\,] \quad . \qquad (4.7)
\end{aligned}$$

Since v is distributed as $N(0,\sigma^2)$, its odd moments vanish and its even moments are

$$E_v[v^{2\,k}] \;=\; 1\cdot 3\cdot 5\cdots(2\,k-1)\,\sigma^{2k} \;=\; (2\,k-1)!!\,\sigma^{2\,k} \quad .$$

From this, the jump moments are found to be

$$\alpha_n(w) \;=\; \sum_{k=0}^{\lfloor n/2\rfloor} \binom{n}{2\,k} \left(\,p_p(w)\,c_p^{n-2k}\,w^{2k} \;+\; p_d(w)\,(-c_d)^{n-2k}\,w^n\,\right)\,(2k-1)!!\,\sigma^{2k} \qquad (4.8)$$

where $\lfloor n/2\rfloor$ denotes the largest integer equal to, or smaller than $n/2$.

4.2 Potentiation and Depression Probabilities

The jump moments (4.8) depend on p_p and p_d. Van Rossum et al. [1] show that under equilibrium conditions these values are approximately equal. To do this, they consider the activity at a single synapse coincident to a neuron.

In the case of depression, the presynaptic event is neither influenced by nor influences the postsynaptic spike.[1] The neuron receives background input from other synapses, modeled by a constant background current I_0 which causes the neuron to fire with an constant interspike interval t_{isi}. Consequently, the probability of depression depends on how likely it is that the synaptic event

[1] This assumes that the synapse is not part of a recurrent structure, driven by the neuron to which it is coincident.

occurs during the active learning window that follows the postsynaptic spike

$$p_d = t_w / t_{isi} \tag{4.9}$$

and, because t_w and t_{isi} are constant, p_d is also constant.

The probability of potentiation p_p is not constant, but increases with the synaptic weight because the postsynaptic spike follows and is influenced by the synaptic event. Van Rossum et al. model the synaptic current as a small rectangular current pulse of width τ_{syn} having amplitude wV_{syn} where w is the synaptic weight and V_{syn} a synaptic drive which is assumed constant. This pulse causes the membrane voltage to jump by an amount $\delta V = \tau_{syn} w V_{syn}/C$ where C is the membrane capacitance. If the jump does not cause an output spike, the interspike interval is shortened to $t'_{isi} = t_{isi} - \delta V C/I_0$. Letting V_{thr} denote the threshold voltage of the neuron relative to the resting voltage, the time between the synaptic event and the spike δt is distributed as

$$\begin{aligned} \wp(\delta t) = &\quad \tfrac{1}{t_{isi}} + \tfrac{V_{syn}}{V_{thr} C} w \quad \text{if } \delta t < \tau_{syn} \\ = &\quad \tfrac{1}{t_{isi}} \qquad\qquad\quad \text{if } \tau_{syn} < \delta t < t'_{isi} \end{aligned} \tag{4.10}$$

where the additive term in the first case of equation (4.10) shows that small intervals between the synaptic event and a postsynaptic spike are more likely. Integrating (4.10) over δt from $\delta t = 0$ to $\delta t = t_w$ and assuming that $\tau_{syn} \ll t_w$ gives the probability that the synapse is potentiated

$$p_p(w) = p_d \, (1 + w/W_{tot}) \tag{4.11}$$

where $W_{tot} = t_w I_0/(V_{syn}\tau_{syn})$. This is the average current to the neuron from all other synaptic inputs expressed as an instantaneous conductance. If we assume that all input to the neuron is excitatory, W_{tot} can also be expressed as

$$W_{tot} = t_w \rho_{in} N \langle w \rangle \tag{4.12}$$

where ρ_{in} is the (assumed constant) spike arrival rate, N is the number of coincident synapses, and $\langle w \rangle$ is their average weight.

The value W_{tot} increases linearly in both the number of synapses co-

incident to the neuron and their average weight; as these values increase, W_{tot} increases and p_p decreases. Balance of these influences occurs when $p_p \approx p_d$. The value of p_p cannot exceed that of p_d by much or the average synaptic weight starts to climb, increasing W_{tot} and reducing p_p; the larger the number of synapses and the higher the average weight, the more marked is this effect. Because the number of synapses adjacent to most neurons is large, W_{tot} is usually large relative to the weight w of the synapse under consideration[2] and the term of p_p linear in w in equation (4.11) is negligible. As such, it is generally valid to make any of the equivalent assumptions

$$w \ll W_{tot} \ ,$$

$$W_{tot} \to \infty \ , \tag{4.13}$$

or

$$p_d(w) \approx p_p(w) = p \quad (\ p \ \text{ constant and independent of } \ w\ ) \ . \tag{4.14}$$

These assumptions will be crucial in what follows.

[2] W_{tot} is usually thousands of times higher than w.

Chapter 5

Closed Form Solutions of the Fokker-Planck Equation

Van Rossum et al. [1] truncate the Kramers-Moyal expansion (3.10) at the second term to obtain a nonlinear Fokker-Planck equation (3.11). This makes use of only the first two jump moments, the drift and the diffusion coefficients, which follow from Equation (4.8) with

$$\alpha_1(w) \;=\; p_p\, c_p \;-\; p_d\, c_d\, w \tag{5.1}$$

and

$$\alpha_2(w) \;=\; p_p\, c_p^2 \;+\; p_d\, c_d^2\, w^2 \;+\; (p_p + p_d)\, w^2\, \sigma^2 \quad . \tag{5.2}$$

By replacing p_p as in (4.11) and making the assumptions that $c_d^2 \ll \sigma^2$ and that $w \ll W_{tot}$, they approximate the jump moments as

$$\alpha_1'(w) \;=\; p_d\,[\,-\,c_d\, w \;+\; (c_p(1 + w/W_{tot})] \tag{5.3}$$

and

$$\alpha_2'(w) \;=\; p_d\,(2\, w^2\, \sigma^2 \;+\; c_p^2) \quad . \tag{5.4}$$

Substituting (5.3) and (5.4) into (3.11) gives van Rossum's formulation of the FPE

$$\frac{\partial P(w,t)}{\partial t} = -\frac{\partial}{\partial w}[p_d\,(\,-\,c_d\,w\;+\;(c_p(1+w/W_{tot}))\,P(w,t)] + \frac{1}{2}\frac{\partial^2}{\partial w^2}[p_d\,(2\,w^2\,\sigma^2\;+\;c_p^2)\,P(w,t)]\;. \tag{5.5}$$

Assuming equilibrium, $\partial_t P(w,t) = 0$, and the FPE can be solved in closed form as in [20], yielding van Rossum's approximation for the equilibrium distribution

$$P(w) = N\frac{e^{\sqrt{2}\arctan(\sqrt{2}\sigma w/c_p)/\sigma}}{(2\sigma^2 w^2\;+\;c_p^2)^{\frac{2\sigma^2\,-\,c_p/W_{tot}\,+\,c_d}{2\sigma^2}}}\;. \tag{5.6}$$

In general, $c_p \ll W_{tot}$ and neglecting the term c_p/W_{tot} in the denominator introduces little error into the solution.[1] When this is done, Equation (5.6) becomes

$$P(w) = N\frac{e^{\sqrt{2}\arctan(\sqrt{2}\sigma w/c_p)/\sigma}}{(2\sigma^2\;+\;c_p^2)^{\frac{2\sigma^2\,+\,c_d}{2\sigma^2}}}\;. \tag{5.7}$$

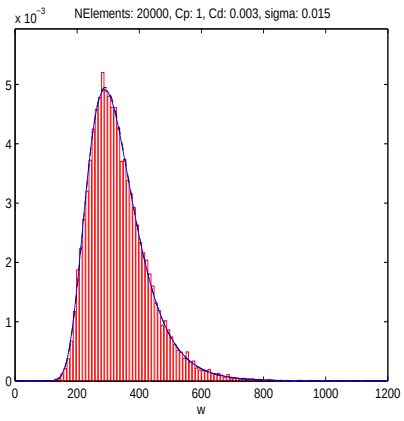
However, the assumption that ${c_d}^2 \ll \sigma^2$, while true in the specific case of the pyramidal neurons studied by van Rossum, artificially limits the range of parameters over which this model is valid, losing accuracy as σ^2 approaches the order of or becomes smaller than c_d^2. We propose an alternate treatment of the model that gives a more accurate approximation that is valid over a wider range of parameters.

We start by making the assumption that $W_{tot} \to \infty$. Note that this implies, via Equation (4.14) that $p_d = p_p = p$. Using the original jump moments (Equations (5.1) and (5.2)) and making this assumption gives our version of the FPE:

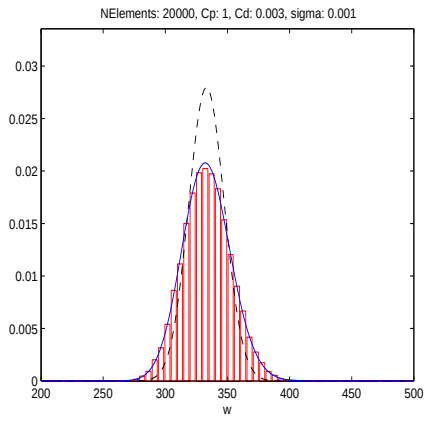
$$\frac{\partial P(w,t)}{\partial t} = -\frac{\partial}{\partial w}[p\,(c_p\;-\;c_d\,w)\,P(w,t)] + \frac{1}{2}\frac{\partial^2}{\partial w^2}[p\,(c_p^2\;+\;c_d^2\,w^2\;+\;2\,w^2\,\sigma^2)\,P(w,t)]\;. \tag{5.8}$$

Letting $2\sigma'^2 \;=\; 2\sigma^2 \;+\; c_d^2$ and, once again solving the equation as in [20], we

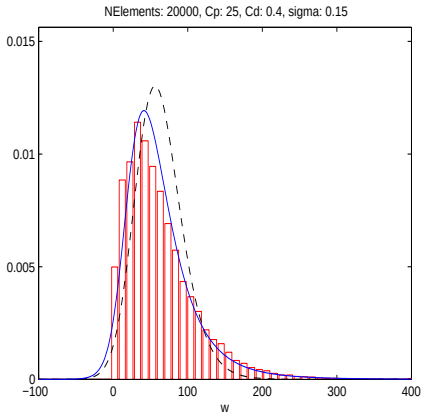
[1] In their paper, van Rossum, et al. draw the same conclusion and verify it using simulation. Note that neglecting this term is equivalent to the assumption (4.13).



(a) Distributions at $c_p = 1$, $c_d = 0.003$, and $\sigma = 0.15$.



(b) Distributions at $c_p = 1$, $c_d = 0.003$, and $\sigma = 0.001$.



(c) Distributions at $c_p = 24$, $c_d = 0.4$, and $\sigma = 0.15$.

Figure 5.1: Comparison of histograms derived from simulations with distributions calculated from Fokker-Planck equations (5.6) (dashed lines) and (5.9) (solid lines).

obtain our approximation to the equilibrium weight distribution

$$P(w) = N \frac{e^{\sqrt{2}\arctan(\sqrt{2}\sigma' w/c_p)/\sigma'}}{(2\sigma'^2 w^2 \; + \; c_p^2)^{\frac{2\sigma'^2 + c_d}{2\sigma'^2}}} \quad . \tag{5.9}$$

Note that the solution given in (5.9) retains the same form (5.7). However, the approximation that $c_d^2 \ll \sigma^2$ is *not* made.

Figures 5.1(a) through 5.1(c) show plots of weight distributions at particular parameter values. Dashed lines in the figures show the FPE solution of van Rossum, et al. (5.6), using the assumptions that $w \to \infty$ and $c_d^2 \ll \sigma^2$. Solid

lines in the figures show our treatment of the FPE (5.9) that uses the assumption $w \to \infty$, but that does not use the assumption $c_d^2 \ll \sigma^2$. The normalized histograms in the figures show results from Markov Chain Monte Carlo simulations using methods described in chapter 9.

Figure 5.1(a) uses the same parameter values as in van Rossum's study – $c_p = 1$, $c_d = 0.003$, $\sigma = 0.15$. At this choice of parameter values, the distributions calculated from both versions of the FPE closely match those derived from the simulation. But when we make σ small with respect to c_d as shown in figure 5.1(b) where $c_p = 1$, $c_d = 0.003$, and $\sigma = 0.001$, the distribution computed from our treatment of the FPE (5.9) matches the simulated distribution much more closely than the one computed using van Rossum's approximations (5.6). Similarly, when when we increase σ and c_d, as shown in Figure 5.1(c) where $c_p = 24$, $c_d = 0.4$, and $\sigma = 0.15$, the distribution computed from our version of the FPE (5.9) again more closely matches that of the simulation than does the distribution computed from (5.6).

The distributions derived from van Rossum's treatment of the FPE are valid at parameters derived from the neurons he was studying, but the assumptions he made for the model's parameters based on these biological limits need not apply to the model in general. These biologically-derived assumptions artificially limit the range of parameters over which the model is valid. Our treatment of the FPE has no such limitations and is applicable over a wider range of parameter values.

Note that the equilibrium distributions produced by these solutions of the Fokker-Planck equations are highly skewed at all but the smallest values of the learning rate parameters c_p, c_d, and σ. When the drift coefficient $\alpha_1(w)$ is linear or constant and the diffusion coefficient $\alpha_2(w)$ is constant, the FPE can only produce a Gaussian distribution. This *linear noise regime* is often held to be the only area where the FPE approximation is valid. As we decrease the values of the model's parameters, the terms of $\alpha_2(w)$ quadratic in w become negligible and the distribution approaches this regime, resulting in an equilibrium distribution which is close to Gaussian.

As shown in Figure 5.1(a), the FPE sometimes makes apparently suc-

cessful predictions far from this regime.[2] However, the FPE does not always lead to such a felicitous result – figure 5.1(c) shows the danger in relying on the Fokker-Planck approximation. Although the distribution computed from our treatment of the FPE (5.9) matches the simulated distribution more closely than that of the distribution computed from van Rossum's treatment(5.6), there still appears to be considerable error between the simulation and the model's prediction.

[2]The skew in the equilibrium distribution generated by the full master equation (3.6) also depends on the third and higher jump moments α_3, α_4, and so forth.

Chapter 6

The FPE is Not a Valid Approximation

The FPE has seen successful use in the fields of machine learning and biological systems. As discussed in Sections 3.2 and 3.3, the FPE has been used successfully, even where nonlinearities are important; where the equilibrium distributions are non-Gaussian and even strongly skewed. That these results hold is surprising, especially as there is no sound mathematical reason for this to be true, as was demonstrated to the machine learning community by Heskes and Kappen [3, 4]. Their work showed that the equilibrium density computed from the FPE cannot be considered an approximate solution to the full Kramers-Moyal equation (3.7) and that truncating the expansion at the second term can lead to significant error. Their findings are rooted in the work of Van Kampen as summarized in his text [9] on statistical physics and chemistry.

Given a one-dimensional FPE, the equilibrium distribution is

$$P_{FP}(w) \;=\; \frac{K}{\alpha_2(w)}\;\exp\left(\frac{2}{\eta}\int\frac{\alpha_1(w)}{\alpha_2(w)}\,dw\right)\quad. \tag{6.1}$$

This equilibrium distribution has maxima near zeros of the drift $\alpha_1(w_0) = 0$ where $d\alpha_1(w_0)/dw < 0$. These are the points where $dE[w]/dt = \eta\ \alpha_1(w)$ (the mean weight dynamics) are asymptotically stable. As the learning rate becomes small, this solution becomes sharply peaked around w_0. Heskes Taylor expands both α_1 and α_2 about w_0, and retains the smallest non-trivial pieces. He assumes

that $\alpha_2(w_0) > 0$, giving the solution for the equilibrium distribution

$$P_{FP,\eta<<1}(w) \;=\; K \;\exp - \left(\frac{|\alpha_1^{'}(w_0)|}{\eta\, \alpha_2(w_0)} \, (w - w_0)^2 \right) \quad .$$

This Gaussian distribution has mean w_0 and variance $(\eta\, \alpha_2(w_0))/(2\ |\alpha_1^{'}(w_0)|\ )$. This result is easy to understand – we expect the variance to increase when the learning rate increases (since with an increased learning rate the weight changes are larger at each step) and we also expect this increase when the magnitude of the diffusion coefficient grows larger, because this causes the distribution to spread. We also expect the variance to decrease when $\alpha_1^{'}$ increases, because this quantity measures the restoring force tending to draw the distribution towards w_0. For extremely small learning rates, approximately Gaussian distributions for all learning rules are a corollary.

Now consider whether or not the equilibrium solution (6.1) obtained from the FPE is a real approximation to the equilibrium computed from the full Kramers-Moyal equation (3.7). If it were, we would expect that if we substitute (6.1) into (3.7), we would obtain remainder terms that approach zero for small learning rate η. However, Heskes and Kappen [3, 4] show that this substitution results in higher order terms that are each of $\mathcal{O}(\eta^0)$ – the Fokker-Planck equilibrium solution fails to satisfy the Kramers-Moyal expansion by an infinite sum of terms each of which is order unity! The discarded terms do not approach zero as $\eta \to 0$[1] and discarding them can lead to substantial error.

If the FPE equilibrium were a valid approximation to the Kramers-Moyal equilibrium, we should be able to construct a perturbation series of the form

$$P(w) \;=\; P_{FP} \;+\; \eta^{\gamma}\, P^{(1)}(w) \;+\; \eta^{2\,\gamma}\, P^{(2)} \;+\; \ldots \tag{6.2}$$

for some positive constant γ. Were this series substituted into the Kramers-Moyal expansion, we would expect equations from which we could compute the corrections $P^{(1)}, P^{(2)}, \ldots$ order-by-order in the learning rate η^{γ}. Unfortunately, this substitution does not lead to such equations and the Fokker-Planck equilibrium cannot be considered the lowest order term in an actual perturbation

[1] This is, in part, a remnant of the fact that P_{FP} is *not* independent of the learning rate η as one expects for the zero-order term in a perturbation expansion.

expansion of the master equation. This analysis shows that any success the FPE has in predicting equilibrium distributions is accidental. This theoretical tool, currently used in in the theoretical neuroscience literature and previously used by the machine learning community, is irrevocably flawed.

In light of this analysis, the machine learning community stopped using the Fokker-Planck equation to determine equilibrium distributions. Heskes and Kapen suggested an alternative analysis based on the fluctuation expansion developed by Van Kampen [9]. This suggestion was problematic, not only because the development of the fluctuation expansion in Van Kampen's text is very incomplete, but also because their suggestion left the community with only a very weak analytical tool.

Heskes and Kappen discard the FPE, but do not suggest anything that deals with nonlinear effects. Their papers, and most of Van Kampen's work, use only the lowest order portion of the fluctuation expansion. This approximation yields a diffusion equation that cannot model non-Gaussian equilibrium distributions. They claim [4] that "only the lowest order term has *any* validity", and quote van Kampen's statement that "all features beyond that approximation are spurious and cannot be taken seriously." The implication is that the particulars of skewed equilibrium distributions or transient phenomena that depend on a nonlinear FPE are completely invalid. Heskes and Kappen and van Kampen's text strongly advise against using the FPE for anything but the lowest order, linear effects, providing only an extremely weak analytical tool useful only in the linear noise regime.

More troubling is the fact that the FPE doesn't always work even in the linear noise regime. As mentioned in section 3.2, the equilibrium distributions found by Leen and Moody [27] for the LMS algorithm agree with Monte Carlo simulations only for very small learning rates, where the distribution is essentially Gaussian. Furthermore, the FPE's success in predicting synaptic weight distributions in biological systems may depend on fortuitously-occurring values of the natural learning rate parameters. For example, the FPE model presented by van Rossum et al. [1] agrees well with Monte Carlo simulations for the physiologically-observed depression and potentiation parameters. However a fifty-fold increase in those parameters takes us into a regime where the FPE grossly fails to predict

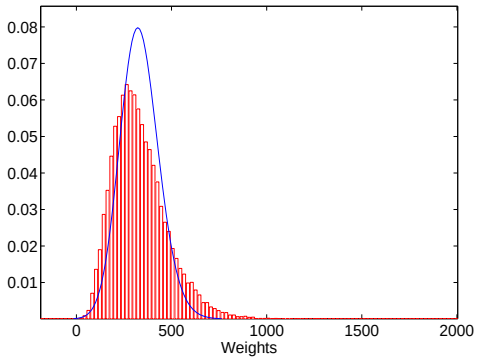


Figure 6.1: Equilibrium distribution from Monte Carlo (histogram) and FPE (curve) following the model in van Rossum et al. [1] but with substantially larger learning rate parameters.

the equilibrium distribution, as in Figure 6.1. One might object that the FPE is an approximation should not be expected to hold for such large increases in the learning rule parameters. The overall point, however, remains valid – the applicability of the FPE for modeling biological systems inherently depends on the fact that the observed physiological learning parameters are small enough. In cases where this does not hold, we have no computational tools with which to work.

So, we are left with a thorny problem. The FPE has been successful in calculating transient and equilibrium effects in both machine and biological learning systems. There are many successes in regimes where nonlinearities are important, such as predictions of strongly skewed distributions and for estimates of basin-hopping. These are regimes in which the lowest order terms, the only ones deemed reliable by Heskes and Kappen, are completely inadequate. On the other hand, the application of the FPE can also give grossly incorrect results that seem to disagree strongly with Monte Carlo simulations and demonstrate clearly that the theoretical warnings about the use of the FPE to approximate the KM series are frequently valid.

As difficult as it may seem, there is no solution to the inadequacies of the FPE approximation. The theoretical analysis of its validity as an approximation to a perturbation expansion is inescapable, as is its inability to provide *reliable* results. The FPE is simply an invalid approximation.

Acknowledging this fact requires us to explore alternate approaches. If we start with the master equation, we cannot simply truncate the Kramers-Moyal expansion and retain validity. Luckily, for some learning rules, such as the one studied by van Rossum, we can glean information about the distribution without making approximations to the Kramers-Moyal expansion. Because no approximation to the KM expansion is made, this information is exact.

Chapter 7

Moments From the Kramers-Moyal Expansion

The master equation (3.6) usually cannot be solved directly to obtain an equilibrium distribution. However, for some Markov models, the exact values of the moments for the equilibrium distribution can be determined. In the case of van Rossum's model, under the assumption that $w << W_{tot}$, the k^{th} jump moment is a polynomial of order k in the weights. For all models where this property holds, the Kramers-Moyal expansion leads to a set of moment equations that form a recurrence which can be solved exactly.

To start, we multiply the KM expansion (3.10) by the k^{th} power of the weight and integrate to find the equation that gives the time evolution of the k^{th} moment of the distribution

$$\partial_t E[w^k] \;=\; \int w^k \sum_{j=1}^{\infty} \frac{(-1)^j}{j!} \eta^j \frac{\partial^j}{\partial w^j} \left[\, \alpha_j(w)\, P(w,t\,)\right] dw \;\; . \tag{7.1}$$

We exchange the integral and summation, integrate by parts, and simplify, yielding

$$\partial_t E[w^k] \;=\; \sum_{j=1}^{k} \binom{k}{j} \eta^j \, E[w^{k-j}\, \alpha_j(w)], \quad k = 1, 2, \ldots \tag{7.2}$$

which defines an equivalent form for the time derivative of the moments.

The key observation is that when all jump moments $\alpha_j(w)$ are polyno-

mials of order j or less in w, the differential equation for each k^{th} moment can be solved using only the solutions of order less than k. To see this, let J_j be the highest power of the weight w in $\alpha_j(\text{w})$. Then the j^{th} jump moment can be defined as

$$\alpha_j(w) = \sum_{i=0}^{J_j} \alpha_{ij} w^i \quad , \tag{7.3}$$

where α_{ij} is the coefficient of the i^{th} power of w in the polynomial defining the j^{th} jump moment.

Using this definition, we rewrite equation (7.2) as

$$\partial_t E[w^k] \;=\; \sum_{j=1}^{k} \binom{k}{j} \eta^j \, E\left[w^{k-j} \sum_{i=0}^{J_j} \alpha_{ij} \, w^i \right] \;=\; \sum_{j=1}^{k} \sum_{i=0}^{J_j} \binom{k}{j} \eta^j \, \alpha_{ij} \, E\left[\, w^{k-j+i} \, \right] \quad . \tag{7.4}$$

and isolating $E[w^k]$ in this equation gives

$$\partial_t E[w^k] \;=\; \sum_{j=1}^{k} \binom{k}{j} \eta^j \, \alpha_{jj} \, E[w^k] \;+\; \sum_{j=1}^{k} \sum_{i=0,\, i\neq j}^{J_j} \binom{k}{j} \eta^j \, \alpha_{ij} \, E[w^{k-j+i}] \quad . \tag{7.5}$$

If $J_j \leq j$ holds for all $j \leq k$, the order of the moments in the second term of the right hand side of equation (7.5) are all less than or equal to k and the equation for $E[w^k]$ depends only on the lower order moments, $E[w], E[w^2], \ldots, E[w^{k-1}]$ (i.e., the set of moment equations form a recurrence which can be solved exactly). However, if $J_j > j$ for one or more $j \leq k$, the k^{th} moment will depend not only on moments of order less than or equal to k, but also on moments of order greater than k. In this case, the moment hierarchy cannot be solved exactly – specifically, any $J_j > j$ will require that the $k + J_j - j^{\text{th}}$ moment equation be solved simultaneously with the k^{th}.

In the case of van Rossum's model, the importance of neglecting the w-dependence of $p_p(w)$ in (4.11) is now evident – retaining this dependency causes each j^{th} jump moment to contain a term having power $j+1$ in the weight. Thus, the equation for each moment would depend on the next higher order moment, preventing us from finding the exact solution for the moments.

To find the equilibrium moments for the distribution, we set $\partial_t \, E[w^k] = 0$

in equation (7.5) and isolate terms containing $E[w^k]$ to give

$$E[w^k] = \frac{N_k}{D_k} = \frac{-\sum_{j=1}^{k} \sum_{i=0,\, i\neq j}^{J_j} \binom{k}{j} \eta^j \alpha_{ij} E[w^{k-j+i}]}{\sum_{j=1}^{k} \binom{k}{j} \eta^j \alpha_{jj}} \tag{7.6}$$

where N_k is the numerator and D_k of this general expression for $E[w^k]$.

7.1 Equilibrium Moments for van Rossum's Model Using the Kramers-Moyal Expansion

In the case of van Rossum's model, the equilibrium value for the k^{th} moment is

$$E[w^k] = \frac{-\sum_{j=1}^{k} \binom{k}{j} \sum_{m=0,\, 2m\neq j}^{\lfloor j/2 \rfloor} \binom{j}{2\,m} (2m-1)!!\, \sigma^{2m} c_p^{j-2m} E[w^{k-(j-2m)}]}{\sum_{m=0}^{\lfloor k/2 \rfloor} \binom{k}{2\,m} (2m-1)!!\, \sigma^{2m} (-c_d)^{k-2m}} \ . \tag{7.7}$$

Because the equations form a recurrence, we can set equation (7.2) equal to zero and solve the moment equations iteratively from $k = 1$ to find all equilibrium moments. For van Rossum's model, we solve the first moment equation to find the value of the mean

$$\begin{aligned} 0 &= E[\,\alpha_1(w)] \\ &= -p\, c_d\, E[w] \ + \ p\, c_p \end{aligned}$$

which gives us

$$E[w] = \frac{c_p}{c_d} \ . \tag{7.8}$$

Similarly, the equation for the second moment is

$$\begin{aligned} 0 &= E[\,2\, w\, \alpha_1(w) \ + \ \alpha_2(w)] \\ &= [2\, p\, \sigma^2 \ + \ p\, c_d\, (c_d \ - \ 2)]\, E[w^2] \ + \ 2\, p\, c_p\, E[w] \ + \ p\, c_p^2 \end{aligned}$$

and substituting (7.8) for $E[w]$, we find its value

$$E[w^2] = \frac{c_p^2\, (c_d \ + \ 2)}{c_d^2\, [c_d\, (2 \ - \ c_d) \ - \ 2\sigma^2]} \ . \tag{7.9}$$

Continuing in the same manner, we find expressions for the third and fourth moments

$$E[w^3] \;=\; \frac{-c_p^3\,(c_d\,\sigma^2 \;-\; c_d^3 \;-\; c_d^2 \;+\; c_d \;+\; 6)}{c_d\,[2\,\sigma^2 \;+\; c_d\,(c_d \;-\; 2)][3\,(c_d \;-\; 2)\,\sigma^2 + c_d\,(c_d^2 \;-\; 3\,c_d \;+\; 3)]} \tag{7.10}$$

$$E[w^4] \;=\; \frac{c_p^4\,n}{d} \tag{7.11}$$

where

$$\begin{aligned} n \;&=\; \sigma^4\,[12\,(c_d^2 \;-\; 2)] \;+ \\ &\quad\; \sigma^2\,[c_d\,(c_d^3 \;-\; 20\,c_d \;+\; 42\,c_d \;+\; 76)] \;- \\ &\quad\; (c_d^6 \;+\; c_d^5 \;+\; 17\,c_d^4 \;-\; 40\,c_d^3 + \;c_d^2 \;+\; 72\,c_d \;+\; 24) \end{aligned} \tag{7.12}$$

and

$$\begin{aligned} d \;&=\; c_d[2\,\sigma^2 \;+\; c_d\,(c_d \;-\; 2)] \;\times \\ &\quad\; [3\,(c_d \;-\; 2)\,\sigma^2 \;+\; c_d\,(c_d^2 \;-\; 3c_d \;+\; 3)] \;\times \\ &\quad\; [6\,\sigma^4 \;+\; 6\,(c_d^2 \;-\; 2\,c_d \;+\; 2)\,\sigma^2 \;+\; c_d\,(c_d^3 \;-\; 4\,c_d^2 \;+\; 6\,c_d \;-\; 4)] \end{aligned} \tag{7.13}$$

7.2 Singularities in the Moments

For each moment, there exists parameter values where the denominator $D_k = 0$ and the value of the moment is undefined.[1] For example, the denominator of the second moment is $D_2 \;=\; c_d^2\,[c_d\,(2 - c_d) - 2\sigma^2]$. Assuming that $c_d \ll 2$, the denominator becomes zero at $\sigma^2 \approx c_d$ and the second moment is singular at this combination of parameter values. In addition, since $N_2 > 0$ for all values c_d and σ, the variance of the distribution is positive only when

$$\sigma^2 \lessapprox c_d \tag{7.14}$$

and the moment is valid only when this constraint holds.

A similar result holds for higher order moments. We start from the value

[1] The fact that there are points where the moment is undefined is not necessarily surprising – there are many continuous probability distributions that have infinite moments.

of D_k in equation (7.6) and make the assumption that $c_d \ll 1$ and $\sigma < 1$. We can now neglect all but the last two terms of the sum and make the approximation

$$D_k \approx (k-1)!!\sigma^{k-1}(-c_d) - \binom{k}{k-2}(k-3)!!\sigma^{k-3}c_d \tag{7.15}$$

Using this approximation, $D_k \approx 0$ when $\sigma^2 \approx c_d/(k-1)$ and at the point where D_k becomes zero, the moment is undefined. As c_d grows larger and the constraint

$$\sigma^2 \lessapprox \frac{c_d}{(k-1)} \tag{7.16}$$

is violated, the k^{th} moment becomes negative and is not valid.

7.3 The Moment Form

We use an induction argument to show that $E[w^k]$ is of the form

$$E[w^k] \;=\; c_p^k \frac{N'_k(c_d,\sigma)}{D'_k(c_d,\sigma)} \tag{7.17}$$

For $k = 1$, equation (7.8) gives us $E[w] = c_p/c_d$, and the statement is true for the base case. Now assume that (7.17) holds for $k = 1 \ldots n-1$. For $k = n$, the terms of the numerator in (7.6) are multiples of $c_p^{j-2m}\, E[w^{n-(j-2m)}] = c_p^{j-2m}[c_p^{n-(j-2m)}\, N'_{n-(j-2m)}(c_d,\sigma)/D'_{n-(j-2m)}(c_d,\sigma)]$. Thus, each term is a multiple of c_p^n and a rational function depending only on c_d and σ. When the terms are summed and the fractions simplified, the moment has the form given in (7.17). We will use this result in the next chapter to show that the relative error between the moments determined from the KM expansion and those determined using the FPE are independent of c_p.

Chapter 8

Moments From the Fokker-Planck Equation

If we set all jump moments beyond α_2 in the Kramers-Moyal expansion (7.2) equal to zero, we obtain the FPE (5.8) solved analytically in chapter 5. Because for all k, the k^{th} jump moment for this approximation is also a polynomial of order less than or equal to k in the weight, the moments for the FPE can also be found exactly.

The first two moments for the Fokker-Planck approximation are the same as those derived from the full KM expansion. For moments of order greater than two, we have

$$\partial_t E_{FP}[w^k] \;=\; k\, E_{FP}[w^{k-1}\alpha_1(w)] \;+\; \frac{k\,(k-1)}{2} E_{FP}[w^{k-2}\alpha_2(w)] \tag{8.1}$$

Using the definition of α_{ij} from equation (7.3), we expand and obtain

$$\begin{aligned} \partial_t E_{FP}[w^k] \quad = \quad & k\{[\alpha_{11} + (k-1)\,\alpha_{22}]\, E[w^k] + \\ & + [\alpha_{01} + (k-1)\,\alpha_{12}] E[w^{k-1}] \\ & + (k-1)\alpha_{02} E[w^{k-2}]\} \quad . \end{aligned} \tag{8.2}$$

To find the equilibrium moments, we set the partial derivative to 0 and

solve giving

$$E[w^k] \;=\; \frac{-\{[\alpha_{01} + (k-1)\,\alpha_{12}]E[w^{k-1}] + (k-1)\alpha_{02}E[w^{k-2}]\}}{[\alpha_{11} + (k-1)\,\alpha_{22}]} \quad . \tag{8.3}$$

8.1 Equilibrium Moments for van Rossum's Model Using the FPE

For van Rossum's model, we have

$$E_{FP}[w^k] \;=\; \frac{-(c_p\,E_{FP}[w^{k-1}] + (k-1)c_p^2 E_{FP}[w^{k-2}])}{-c_d + (k-1)(c_d^2 + 2\sigma^2)} \quad . \tag{8.4}$$

Specifically, the third and fourth moments for the FPE solution are

$$E_{FP}[w^3] \;=\; \frac{-c_p^3\,(2\sigma^2 \,+\, c_d^2 \,-\, 3c_d \,-\, 2)}{c_d\,[2\sigma^2 \,+\, c_d(c_d \,-\, 2)]\,[2\sigma^2 \,+ c_d(c_d \,-\, 1)]} \tag{8.5}$$

and

$$E_{FP}[w^4] \;=\; \frac{c_p^4\,[2(3c_d + 8)\sigma^2 \,+\, 3c_d^3 \,+\, 5c_d^2 \,-\, 12c_d \,-\, 4]}{c_d\,[2\sigma^2 \,+\, c_d(c_d \,-\, 2)]\,[2\sigma^2 \,+ c_d(c_d \,-\, 1)][6\sigma^2 \,+\, c_d(3c_d \,-\, 2)]} \quad . \tag{8.6}$$

The first and second moments of the FPE solution for van Rossum's model are identical to those derived from the full Kramers-Moyal expansion, given in (7.8) and (7.9) respectively. Thus, the constraint (7.14) must hold for the second moment of the FPE to be valid. When we look at the third (8.5) and fourth (8.6) moments we see that the constraint defined by (7.16) must be satisfied for these moments to be valid, as well.

In the expression for the third moment(8.5), assuming that $c_d << 1$, the third factor in the denominator becomes 0 when $c_d \approx \sigma^2/2$. A examination of the equation shows that the expression switches signs at this point and that beyond this point, the third moment becomes negative. A similar finding holds for the fourth moment (8.6), where the fourth factor in the denominator becomes 0 when $c_d \approx \sigma^2/3$ and becomes negative past this point.

Using an analysis similar to that performed in section 7.2, higher order

moments must also obey the constraint given in (7.14). Where both $E_{FP}[w^3]$ and $E[w^3]$ are valid, $E_{FP}[w^3] - E[w^3] > 0$, i.e., the FPE overestimates the value of the third moment. This finding also holds for fourth and higher moments.

8.2 Moment Form and Relative Error

An induction argument similar to that in section 7.3 shows that $E_{FP}[w^k]$ has a form similar to that of $E[w^k]$

$$E_{FP}[w^k] \;=\; c_p^k \, \frac{N'_{k,\,FP}(c_d, \sigma)}{D'_{k,\,FP}(c_d, \sigma)} \tag{8.7}$$

We can now define the relative error of the k^{th} moment as

$$\rho_k(\sigma, c_d) = \frac{E_{FP}[w^k] \;-\; E[w^k]}{E[w^k]} \quad . \tag{8.8}$$

Since the quantities in the numerator and in the denominator are of the form $c_p^k \, F(c_d,\, \sigma)$, the relative error for all moments is independent of c_p, and depends only on c_d and σ.

Chapter 9

Experimental Results

We validate our results by comparing them to Monte-Carlo simulations of an ensemble of neural weights undergoing the changes (4.4) and (4.5) specified by van Rossum's model at a variety of model parameter values. We collect statistical measures for the distributions obtained from these simulations and compare them with the values of the corresponding statistical measures obtained from the analytical solutions of the model moments (7.6) and (8.4) derived in sections 7 and 8 respectively. The MATLAB©[1] code for the simulation is found in Appendix A.

All simulations start with 20,000 weights in the simulated ensemble. Letting $W_t = [w_j(t)]$ denote the vector of simulated weights at time step t, where $w_j(t)$ denotes the j^{th} weight at time t, the initial ensemble W_0 is randomly drawn from a Gaussian distribution with mean and variance derived from equations (7.8) and (7.9). At every time step each weight in the ensemble is either decreased or increased with probability $p = p_d = p_p = 0.25$ in accord with equation (4.14). If the weight is decreased, the decrease is $w_j(t+1) = w_j(t) - w_j(t)c_d + vw_j(t)$ as specified in (4.5), where v is a random variable drawn from a normal distribution with mean 0 and variance σ^2. If the weight is increased, we use the increase specified in (4.4): $w_j(t+1) = w_j(t) + c_p + vw_j(t)$. At each time step during the data collection period, we collect the k^{th} powers of the weights $[w_j^k(t)]$ for $k = 1$ to 4 and average them over the ensemble to give the ensemble means for the raw moments of the weights $\bar{W_t^k}$. The k^{th} ensemble mean $\bar{W_t^k}$ approximates $E[w^k]$. Simultaneously, we also collect the ensemble means of other statistical measures

[1]MATLAB is a registered trademark of The MathWorks, Inc.

derived from the raw moments such as the variance, skew, kurtosis, and central moments.

The simulations are *burned in* for a period of 10,000 time steps and then run for a *data collection period* of 90,000 time-steps where statistical measures are collected. The number of time steps for the burn-in and run periods were determined by visually inspecting several runs. For all simulations, Gewecke's Z-diagnostic [34] was then applied to the collected moments from the first 10,000 and final 10,000 time steps of the data collection period to check that these values had converged for the data collection period.

When viewed as a time series over the data collection period, the ensemble mean for any statistical measure is itself a stochastic variable. To compute a single representative value for the statistical measures to compare with our analytical results, we average the ensemble means $\bar{W}_t^k$ over the time steps in the data collection period to give $\hat{W^k}$, the time-averages of the ensemble means. The time-averages for the variance, skew, kurtosis, and central moments are also computed and collected.

The simulations were run using parameter values in the range $0.1 \leq c_p \leq 500$, $0 < c_d < 1$ and $0 < \sigma < 1$. We compared these time-averaged values with the analytical values for these measures derived from equations (7.6) and (8.4).

Figure 9.1 plots the third central moment $\mu_3 = E[w^3] - E[w]^3$ derived from the simulation (solid line), the Kramers-Moyal formula (circles), and the Fokker-Planck formula (dots) at the parameter values $[c_p,\, c_d,\, \sigma] = k\,[1,\, 0.003,\, 0.015]$. The parameter values used in van Rossum's study correspond to the point $k = 1$ and this value is marked with an 'X'. Note that for the small parameter values used (including the value used in van Rossum's study) all three values match closely. Even so, at the larger end of this parameter set, the values derived using the Fokker-Planck approximation are beginning to diverge.

Figure 9.2 shows the value of third central moments at the parameter values $[c_p,\, c_d,\, \sigma] = k\,[1,\, 0.015,\, 0.015]$. As the parameter values grow larger, the values for the Fokker-Planck approximation diverge more pronouncedly from those of the simulation. The Kramers-Moyal solution continues to track the simulation with no error.

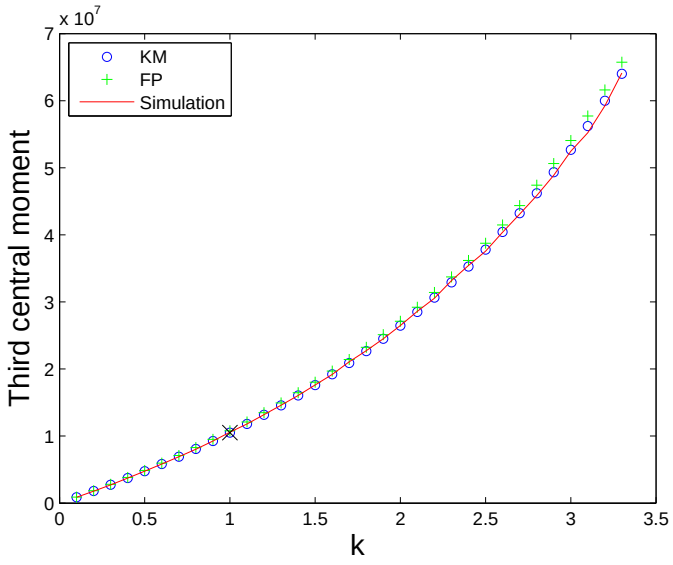


Figure 9.1: Third central moment for the simulation, the Kramers-Moyal expansion, and the Fokker-Planck approximation for values $[c_p,\, c_d,\, \sigma] = k\,[1,\, 0.003,\, 0.015]$. The 'X' indicates the parameters values for the simulation used in van Rossum's study.

Figure 9.3 shows the third central moment at the points $[c_p,\, c_d,\, \sigma] = k\,[25,\, 0.075,\, 0.015]$, respectively. Even when $k > 2$ and the parameters become very large when compared with those in van Rossum's study, the values of the moments derived from the full Kramers-Moyal expansion track the simulations exactly. The values derived from the Fokker-Planck approximation diverge to an ever greater degree.

From these comparisons covering a broad range of parameter values, it is clear that the full Kramers-Moyal expansion gives an exact solution for the third central moment. Similar findings hold for the raw third moment, $E[w^3]$, and skew $\mu_3/\mu_2^{3/2}$ where μ_i denotes the i^{th} central moment.

As discussed in section 8, the Fokker-Planck approximation overestimates the values for the third central moment (and for the raw third moment, as well). This finding holds across all values of the parameter space until the constraint noted in (7.16) is violated.

Figure 9.4 plots discrepancies in the fourth central moments between the Fokker-Planck approximation (dots) and the Kramers-Moyal expansion (circles), and the simulation (solid line) at the parameter values $[c_p, c_d, \sigma] = k\,[1, 0.015, 0.015]$

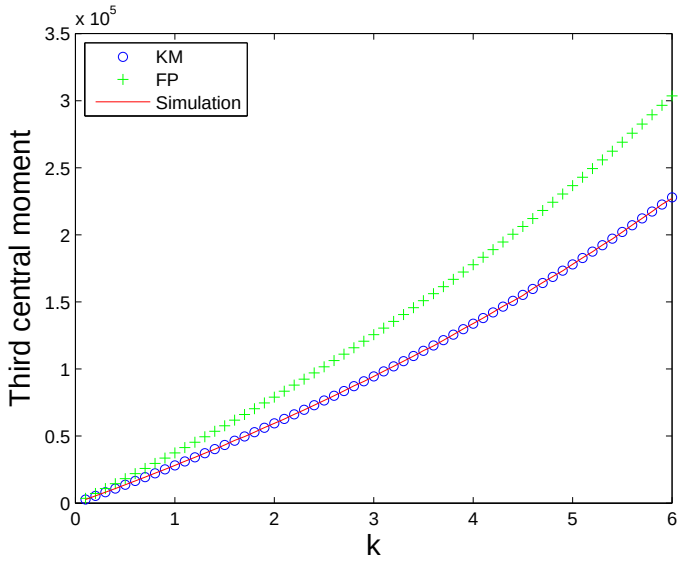


Figure 9.2: Third central moment for the simulation, the Kramers-Moyal expansion, and the Fokker-Planck approximation for values $[c_p,\, c_d,\, \sigma] = k\,[1,\, 0.015,\, 0.015]$.

(LEFT) and $[c_p, c_d, \sigma] = k\,[25, 0.075, 0.015]$ (RIGHT). Again, each plot shows considerable error in using the Fokker-Planck approximation, while the Kramers-Moyal solution gives results that track the simulation exactly. In these cases, the raw fourth moment is overestimated, though this fact is not necessarily evident in the plot of the fourth central moment.[2] As is the case with the third moment, this overestimation continues until the constraint noted in (7.16) is violated.

Figure 9.5 shows a contour plot of ρ_3 the relative error in the raw third moments between the FPE and Kramers-Moyal expansion given by equation (8.8). Since the relative error is independent of c_p, it can be displayed as a contour on the c_d-σ plane. The relative error of the Fokker-Planck approximation is quite low where σ and c_d are small, but grows quickly as these parameters become larger. One can see the contours of the relative error bunch together as the parameters approach singular points in the parameter space, indicating rapid increase in the relative error as model parameters approach these points. The

[2]The fourth central moment μ_4 is defined in terms of the raw moments $E[w]$ through $E[w^4]$ as

$$\mu_4 = E[w^4] - 8E[w^3]E[w] + 6E[w^2]^2 + E[w]^4 \quad .$$

Whether or not the fourth central moment is underestimated or overestimated (or neither) depends on the relative magnitudes of the raw moments.

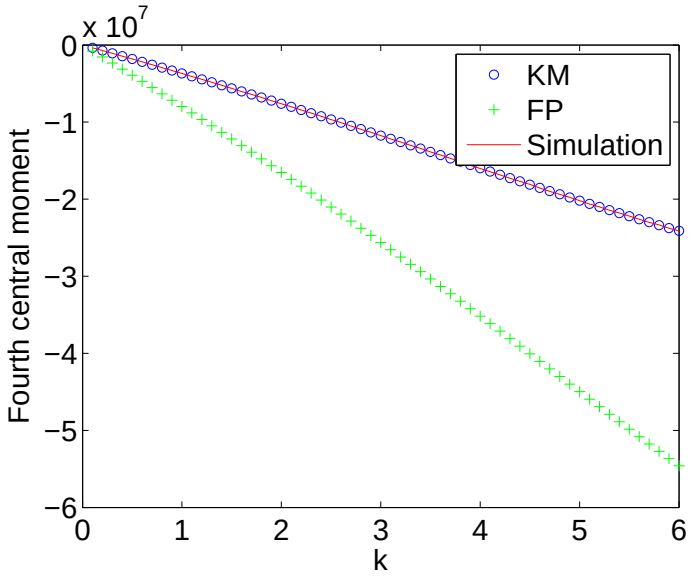


Figure 9.3: Third central moment for the simulation, the Kramers-Moyal expansion, and the Fokker-Planck approximation for values $[c_p,\, c_d,\, \sigma] = k\,[25,\, 0.075,\, 0.015]$.

dotted lines on the graph indicate the traces in the c_d-σ plane from which the plots in figures 9.1, 9.2, and 9.3 were taken. Again, 'X' indicates the parameter values used in van Rossum's simulation. Figure 9.6 plots a similar set of contours for ρ_4 the relative error in the raw fourth moment where the dotted lines denote the traces in the c_d-σ plane from which the plots in figure 9.4 were taken.

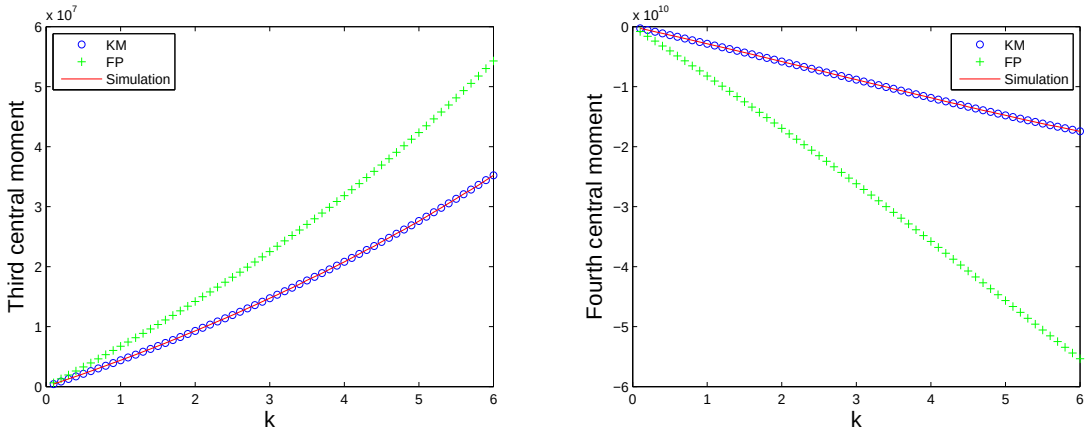


Figure 9.4: Fourth central moments for the simulation, the Kramers-Moyal expansion, and the Fokker-Planck approximation. LEFT: $[c_p,\, c_d,\, \sigma] = k\,[1,\, 0.015,\, 0.015]$ RIGHT: $[c_p,\, c_d,\, \sigma] = k\,[25,\, 0.075,\, 0.015]$

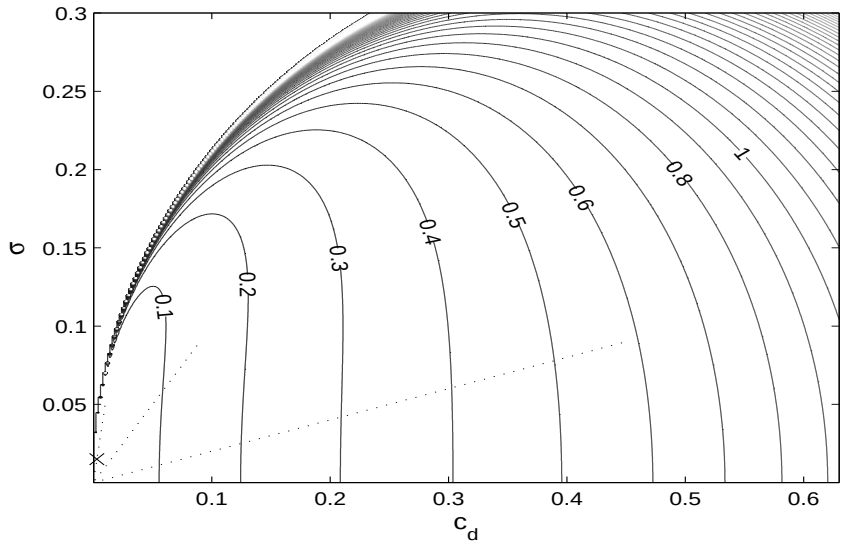


Figure 9.5: Contour plot of relative error of the raw third moment of the Fokker-Planck solution compared with the Kramers-Moyal solution.

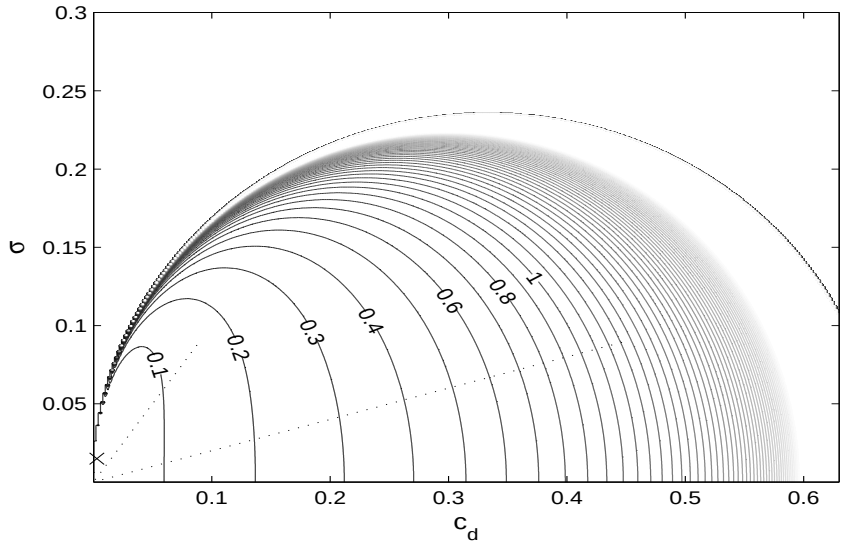


Figure 9.6: Contour plot of relative error of the raw fourth moment of the Fokker-Planck solution compared with the Kramers-Moyal solution.

Chapter 10

Discussion and Conclusion

The weight distribution $P(w,t)$ holds all information about the learning dynamics of a neural system. For the asymmetric spike-timing-dependent model determined in van Rossum et al. [1], the FPE (5.6) accurately modeled the weight distribution of cultured rat hippocampal neurons. Our study of this model shows that the FPE closely matched the physical result only for the biophysically determined model parameters used in his study. Van Rossum's assumptions about the relative sizes of the parameters c_d and σ^2 lead to inaccuracies when σ approaches or becomes less than c_d. When c_p, c_d, or σ grow larger than the biophysically determined constant, the error of the FPE approximation grows and the approximate solution may not match the actual results produced by the model.

The inability of the FPE solution to accurately and reliably approximate the equilibrium distribution seriously limits its use as an analytical tool. Heskes and Kappen [3, 4] and Van Kampen [9] showed that the use of the FPE is justified only in the linear noise regime where effects of a non-linear drift coefficient $\alpha_1(w)$ and non-constant diffusion coefficient $\alpha_2(w)$ can be ignored and the distributions are Gaussian. If the FPE consistently gave good results whenever this limitation was observed and consistently failed when it was not, it might be salvaged for general use. However, as discussed in Chapter 6, in both the machine learning and biological arenas, there are examples where the FPE gives good predictions even when the distribution is highly skewed and far from the linear-noise limit; other examples exist where the FPE's predictions fail, even when the distribution is close to Gaussian and the parameters are small. Because the FPE gives

inconsistent results, it cannot be a reliable tool to analyze learning dynamics.

Conversely, solutions based on the full Kramers-Moyal expansion are exact. In the case of van Rossum's model, the Kramers-Moyal expansion can be used to determine the exact moments of the equilibrium weight distribution. Our comparison of these moments with simulations confirm that the analytical results derived from the Kramers-Moyal expansion are indeed exact across a broad range of parameter values, including those where the FPE fails. Comparison against moments derived using the FPE shows that the moments derived from the Kramers-Moyal expansion give more accurate predictions of moment values than the FPE.

Specifically, if a learning rule produces a Markov process where for $k = 1, 2, \ldots, K$, the k^{th} jump moment is a polynomial of order k or less in the weight, the Kramers-Moyal expansion gives a moment recurrence that can be solved exactly for the K^{th} moment of the equilibrium distribution. The values for the equilibrium moments obtained in this way, being exact, will *always* be more accurate than the value of the moments derived from the FPE approximation. As we have shown with van Rossum's model, this provides a new tool that can be used to analyze learning dynamics for systems governed by learning rules that meet these criteria.

Finally, this study shows that methods based on the full Kramers-Moyal expansion can be useful tools. One approach, unexamined here, starts with the equation (4.12) giving the relationship $p_p(w) = p_d(w)(1 + W_{tot})$. Using this equation, one can rewrite the jump moments (3.8) in terms of $p_d(w)$. Substituting this form of the jump moments into the Kramers-Moyal expansion (3.7) leads to an equation that can be cast in terms of w/W_{tot}. This quantity is small enough that one should be able to *accurately* approximate the equilibrium distribution as a proper perturbation series in w/W_{tot}. Solving this perturbation series should give increasingly accurate approximations of $P(w, t)$ as the number of retained terms in the series increases. This approach should be explored for van Rossum's model and other models with learning rules having equal time windows for potentiation and depression, where equation (4.12) holds.

Another method that bears additional investigation is suggested by Leen in [35]. This work suggests starting with the fluctuation expansion of Van Kam-

pen [9]. When this expansion is substituted into the Kramers-Moyal expansion, a proper perturbation expansion results. Iterative solution of truncated versions of this expansion should provide approximate distributions which increase in accuracy as more terms are added. This approach would provide computational tools for analyzing learning dynamics where current tools are misleading, inaccurate, or both. Further research is needed to validate this approach.

Appendix A

Code for Monte Carlo Markov Chain Simulation

We show here the MATLAB© code for the simulations used to validate our results. The general strategy is that simulation variables are globally held, with parameters for the simulation set before the run and values extracted from the globals after the run and saved. Simulation state and values are cached in files and reloaded when necessary to obviate the need for re-running simulations to plot different values or to extend the simulation time.

A.1 set_simulation_parameters.m

This function declares all global parameters used in the simulation and sets the variables that control the simulation.

```
function set_simulation_parameters(ne, pr, cp, cd, s, nb, na)

% Cd, Cp, p, sigma as in van Rossum et al.  NElements = # of
% simulated weights, NBurnin = burn in period, NAvg = data
% collection period.
global Cd Cp p NElements sigma NBurnin NAvg;

% w = weight vector, SMeans = Simulated means (for each time step),
```

```
% SMom2s = Simulated raw second moment (for each time step), etc.
global w SMeans SMom2s SMom3s SMom4s;

% ASMean = Average Simulated Mean (averaged over NAvg time steps),
% ASVar = Average Variance (averaged over NAvg time steps), etc.
global ASMean ASMom2 ASMom3 ASMom4;
global ASVar ASMom3C ASSkew ASMom4C ASKurtosis;

% ThMean = Theoretical Mean (calculated from KM expansion),
% ThVar = Theoretical Variance (calculated from KM expansion), etc.
global ThMean ThMom2 ThMom3 ThMom4;
global ThVar ThMom3C ThSkew ThMom4C ThKurtosis;

% FPMom3 = Raw third moment (calculated from FPE),
% FPMom3C = Third central moment  (calculated from FPE), etc.
global FPMom3 FPMom4;
global FPMom3C FPSkew FPMom4C FPKurtosis;

% Miscellaneous constants needed to normalize probability
% distribution
global LogFPEAScale LogNorm;

NElements = ne;
p = pr;
Cd = cd;
Cp = cp;
sigma = s;
NBurnin = nb;
NAvg = na;

% Set theoretical values - Mean, 2'nd moment and variance
ThMean = Cp/Cd;
ThMom2 = -(2*Cp*ThMean+Cp^2)/(2*sigma^2+Cd^2-2*Cd);
ThVar=ThMom2-ThMean*ThMean;
```

```
% Set theoretical values for full KM model- 3'rd and 4'th
% moments, skew and kurtosis
ThMom3num = 3*Cp*(sigma^2+1)*ThMom2+3*Cp^2*ThMean+Cp^3;
ThMom3denom = 3*(Cd-2)*sigma^2+Cd^3-3*Cd^2+3*Cd;
ThMom3 = ThMom3num/ThMom3denom;
ThMom3C=ThMom3-ThMean*ThMean*ThMean;
ThSkew = ThMom3C/sqrt(ThVar*ThVar*ThVar);

ThMom4num = (12*cp*ThMom3+6*Cp*Cp*ThMom2)*sigma^2+...
4*Cp*ThMom3+6*Cp^2*ThMom2+4*Cp^3*ThMean+Cp^4;
ThMom4denom=6*sigma^4+6*(Cd^2-2*Cd+2)*sigma^2+...
Cd*(Cd^3-4*Cd^2+6*Cd-4);
ThMom4=-ThMom4num/ThMom4denom;
ThMom4C = ThMom4+ThMean^4-8*ThMean*ThMom3+6*ThMom2*ThMom2;
ThKurtosis = (ThMom4C/(ThVar*ThVar))-3;


% Set theoretical values for truncated model - 3'rd and 4'th
% moments, skew and kurtosis
FPMom3num = -(Cp*ThMom2+3*Cp^2*ThMean);
FPMom3denom = 2*sigma^2+Cd^2-Cd;
FPMom3 = FPMom3num/FPMom3denom;

FPMom3C=FPMom3-ThMean*ThMean*ThMean;
FPSkew=FPMom3C/sqrt(ThVar*ThVar*ThVar);

FPMom4num = Cp*(2*FPMom3+3*Cp*ThMom2);
FPMom4denom = 6*sigma^2+Cd*(3*Cd-2);
FPMom4=-FPMom4num/FPMom4denom;
FPMom4C = FPMom4+ThMean^4-8*ThMean*FPMom3+6*ThMom2*ThMom2;
FPKurtosis = (FPMom4C/(ThVar*ThVar))-3;
```

```
end
```

A.2 make_initial_population.m

This function initializes the global vector `w` to random values with mean and variance derived from equations (7.8) and (7.9).

```
function make_initial_population()
global NElements ThMean ThVar w;

% Initialize ensemble of synapses about asymptotic
% mean with asymptotic variance
w = ThMean * ones(NElements,1);
noise = sqrt(ThVar)*randn(NElements,1);
w = w+noise;
end
```

A.3 data_exists.m

Because the MCMC simulations took so long to run, after each simulation we saved the final vector of simulated weights and their averaged raw moments from each time step. If the data at that set of parameter values was needed again (e.g., for generating plots), we reloaded the data rather than running a new simulation.

This function checks to see if a simulation for this parameter set already exists.

```
function res = data_exists()

fDir = archive_directory_name();
res = isdir(fDir);
end
```

A.4 archive_directory_name.m

This function maps the simulation parameters into a name for a directory where the simulation data will be stored.

```
function fdir = archive_directory_name()
global Cd Cp p NElements sigma NBurnin NAvg;

fdir = sprintf('data\\S%d-%08.4f-%08.4f-%08.4f', ...
NElements, Cp, Cd, sigma);
end
```

A.5 save_data.m

This function saves the simulation data in case it is needed again.

```
function save_data()

global Cd Cp p NElements sigma NBurnin NAvg;
global w SMeans SMom2s SMom3s SMom4s;
global ASMean ASMom2 ASMom3 ASMom4;
global ASVar ASMom3C ASSkew ASMom4C ASKurtosis;
global ThMean ThMom2 ThMom3 ThMom4;
global ThVar ThMom3C ThSkew ThMom4C ThKurtosis;
global FPMom3 FPMom4;
global FPMom3C FPSkew FPMom4C FPKurtosis;
global AFPMean AFPMom2 AFPVar AFPMom3 AFPMom3C AFPSkew;
global AFPMom4 AFPMom4C AFPKurtosis;
global vrAFPMean vrAFPMom2 vrAFPVar vrAFPMom3 vrAFPMom3C;
global vrAFPSkew vrAFPMom4 vrAFPMom4C vrAFPKurtosis;

fDir = archive_directory_name();

if(~data_exists())
```

```
    mkdir(fDir);
end

NTot = NBurnin+NAvg;
params = [NElements, Cp, Cd, sigma, NTot];
paramFile = strcat(fDir, '\p');
save(paramFile, 'params', '-ascii');

weightFile = strcat(fDir, '\w');
save(weightFile, 'w', '-ascii');

m1File = strcat(fDir, '\m');
save(m1File, 'SMeans', '-ascii');

m2File = strcat(fDir, '\m2');
save(m2File, 'SMom2s', '-ascii');

m3File = strcat(fDir, '\m3');
save(m3File, 'SMom3s', '-ascii');

m4File = strcat(fDir, '\m4');
save(m4File, 'SMom4s', '-ascii');

allStats = [params, NBurnin, NAvg, ASMean, ThMean, ASMom2, ...
ThMom2, ASVar, ThVar, ASMom3, ThMom3, FPMom3, ASMom3C, ...
ThMom3C, FPMom3C, ASSkew, ThSkew, FPSkew, ASMom4, ...
ThMom4, FPMom4, ASMom4C, ThMom4C, FPMom4C, ASKurtosis, ...
ThKurtosis, FPKurtosis, AFPMean, AFPMom2, AFPVar, ...
AFPMom3, AFPMom3C, AFPSkew, AFPMom4, AFPMom4C, ...
AFPKurtosis, vrAFPMean, vrAFPMom2, vrAFPVar, vrAFPMom3, ...
vrAFPMom3C, vrAFPSkew, vrAFPMom4, vrAFPMom4C, ...
vrAFPKurtosis ...
];
ssFile = strcat(fDir, '\ss');
```

```
save(ssFile, 'allStats', '-ascii');

cumDataFileName='data\\allRuns';
cID = fopen(cumDataFileName, 'at+');
for i=1:length(allStats)
    fprintf(cID, '%g  ', allStats(i));
end
fprintf(cID, '\n');
fclose(cID);

statsFile = strcat(fDir, '\stats-print');
fID = fopen(statsFile, 'wt+');
fprintf(fID, 'NElements=%d\n', NElements);
fprintf(fID, 'NTimesteps=%d\n', NBurnin+NAvg);
fprintf(fID, 'NAvg=%d\n', NAvg);
fprintf(fID, 'p=%g\n', p);
fprintf(fID, 'Cp=%g\n', Cp);
fprintf(fID, 'Cd=%g\n', Cd);
fprintf(fID, 'sigma=%g\n\n', sigma);

fprintf(fID, 'model\t\t\t\t\t\t\tMean\t\t2''nd Moment');
fprintf(fID, '\t\tVariance\t\t3''d Moment');
fprintf(fID, '\t\tCentral 3''rd Moment\t\tSkew');
fprintf(fID, '\t\t4''th Moment\t\tCentral 4''th Moment');
fprintf(fID, '\t\tKurtosis\n');
fprintf(fID, 'Simulation\t\t\t\t\t\t%g\t\t%g\t\t\t%g', ...
ASMean, ASMom2, ASVar);
fprintf(fID, '\t\t\t%g\t%g\t\t\t%g\t\t%g\t%g\t\t\t%g\n', ...
ASMom3, ASMom3C, ASSkew, ASMom4, ASMom4C, ASKurtosis);
fprintf(fID, 'Full K-M\t\t\t\t\t\t%g\t\t%g\t\t\t%g', ...
ThMean, ThMom2, ThVar);
fprintf(fID, '\t\t\t%g\t%g\t\t\t\t%g\t\t%g\t%g\t\t\t\t%g\n', ...
ThMom3, ThMom3C, ThSkew, ThMom4, ThMom4C, ThKurtosis);
fprintf(fID, 'FP (Truncated K-M)\t\t\t\t---\t\t\t---\t\t\t\t---');
```

```
fprintf(fID, '\t\t\t\t%g\t%g\t\t\t\t\t%g\t\t%g\t%g\t\t\t\t%g\n', ...
FPMom3, FPMom3C, FPSkew, FPMom4, FPMom4C, FPKurtosis);
fprintf(fID, 'FP (Integrated estimate)\t\t%g\t\t%g\t\t\t\t%g', ...
AFPMean, AFPMom2, AFPVar);
fprintf(fID, '\t\t\t%g\t%g\t\t\t\t\t%g\t\t%g\t\t%g\t\t\t\t%g\n', ...
AFPMom3, AFPMom3C, AFPSkew, AFPMom4, AFPMom4C, AFPKurtosis);
fprintf(fID, 'FP (Integrated estimate - VR)\t%g\t\t%g\t\t\t\t%g', ...
vrAFPMean, vrAFPMom2, vrAFPVar);
fprintf(fID, '\t\t\t%g\t%g\t\t\t\t\t%g\t\t%g\t%g\t\t\t\t%g\n',
vrAFPMom3, vrAFPMom3C, vrAFPSkew, vrAFPMom4, vrAFPMom4C, ...
vrAFPKurtosis);
fclose(fID);

end
```

A.6 load_data.m

This function reloads simulation data for a particular set of simulation parameters.

```
function params = load_data()
global w SMeans SMom2s SMom3s SMom4s;
global NBurnin NAvg;

fDir = archive_directory_name();

NTot = NBurnin+NAvg;
paramFile = strcat(fDir, '\p');
p = load(paramFile);

weightFile = strcat(fDir, '\w');
w = load(weightFile);

m1File = strcat(fDir, '\m');
```

```
SMeans = load(m1File);

m2File = strcat(fDir, '\m2');
SMom2s = load(m2File);

m3File = strcat(fDir, '\m3');
SMom3s = load(m3File);

m4File = strcat(fDir, '\m4');
SMom4s = load(m4File);

params = p;

end
```

A.7 run_simulation.m

This function checks to see if the simulated data exists. If it does, it loads the data, otherwise, it initializes the vector of simulated weights using the function `make_initial_population`. It then runs runs the simulation for any additional time-steps and saves the data.

```
function run_simulation(tell_step)
global Cd Cp p NElements sigma NBurnin NAvg;
global w SMeans SMom2s SMom3s SMom4s;

Pd = p;
Pp = p;

% Check to see if we already have some simulation data to
% start from...

if (data_exists())
    load_data();
```

```
else
    make_initial_population();
    SMeans = [];
    SMom2s = [];
    SMom3s = [];
    SMom4s = [];
end

NSteps=2000;
TSteps=length(SMeans);
while (check_vrsim_convergence(NBurnin+NAvg, ...
SMeans, SMom2s, SMom3s, SMom4s))
    if tell_step
        TSteps
    end
  % Do Nsteps steps of random walk
  for step = 1:NSteps
    toss = rand(NElements,1);
    noise = sigma*randn(NElements,1);
    noisew = noise.* w;
    depress = (toss <= Pd);
    potentiate = (toss>Pd)&(toss<(Pd+Pp));
    wp = potentiate .* (Cp+noisew);
    wd = depress .* (-Cd*w + noisew);
    w =  w + wp + wd;
    TSteps = TSteps+1;
    m=mean(w);
    w2=w.*w;
    m2=mean(w2);
    w3=w2.*w;
    m3=mean(w3);
    w4=w3.*w;
    m4=mean(w4);
    SMeans = [SMeans; m];
```

```
    SMom2s = [SMom2s; m2];
    SMom3s = [SMom3s; m3];
    SMom4s = [SMom4s; m4];
  end % for step
end % while

get_average_simulation_data();

% Now that we've simulated, save the data...
save_data();
end
```

A.8 get_average_simulation_data.m

This function calculates the means of all of the statistical variables across all time steps.

```
function get_average_simulation_data()
global NBurnin NAvg;
global w SMeans SMom2s SMom3s SMom4s;
global ASMean ASMom2 ASMom3 ASMom4;
global ASVar ASMom3C ASSkew ASMom4C ASKurtosis;

% Analysis: Average values of simulated quantities
ASMean = mean(SMeans((NBurnin+1):(NBurnin+NAvg)))
ASMom2 = mean(SMom2s((NBurnin+1):(NBurnin+NAvg)))
ASMom3 = mean(SMom3s((NBurnin+1):(NBurnin+NAvg)))
ASMom4 = mean(SMom4s((NBurnin+1):(NBurnin+NAvg)))
sm2 = SMeans.*SMeans;
var = SMom2s-sm2;
ASVar = mean(var((NBurnin+1):(NBurnin+NAvg)))
sm3 = sm2.*SMeans;
m3c = SMom3s-sm3;
ASMom3C = mean(m3c((NBurnin+1):(NBurnin+NAvg)))
```

```
skew =m3c./sqrt(var.*var.*var);
ASSkew = mean(skew((NBurnin+1):(NBurnin+NAvg)))
sm4 = sm3.*SMeans;
m4c =SMom4s+SMeans.^4-8*SMeans.*SMom3s+6*SMom2s.^2;
ASMom4C = mean(m4c((NBurnin+1):(NBurnin+NAvg)))
kurtosis = (m4c./(var.*var))-3;
ASKurtosis = mean(kurtosis((NBurnin+1):(NBurnin+NAvg)))

end
```

A.9 check_vrsim_convergence.m

This function was to have checked to see if the simulated data converged. Theoretical convergence criteria were all too inexact to determine when to halt the simulation, so a simple test against the number of time steps was substituted. We determined the number of time-steps to simulate by visually inspecting several runs and determining that the values of the moments and derived statistical measures stabilized within these time periods. Elegant? No... Effective? Yes!

```
function cont = check_vrsim_convergence(StepMax, ...
means, mom2s, mom3s, mom4s)
  s = length(means);
  cont = (s < StepMax);
end
```

Bibliography

[1] M.C.W. van Rossum, G.Q. Bi, and G.G. Turrigiano. Stable Hebbian learning from spike timing-dependent plasticity. *J Neuroscience*, 20:8812–8821, 2000.

[2] Q. Bi and M. Poo. Precise spike timing determines the direction and extent of synaptic modifications in cultured hippocampal neurons. *J Neuroscience*, 18:10464–10472, 1998.

[3] Tom Heskes. On fokker-planck approximations of on-line learning processes. *J Phys A*, 27:5145–5160, 1994.

[4] Tom M. Heskes and Bert Kappen. On-line learning processes in artificial neural networks. In J. Taylor, editor, *Mathematical Foundations of Neural Networks*, pages 199–233. Elsevier, 1993.

[5] D.O. Hebb. *The Organization of Behavior*. Wiley, NY, 1949.

[6] T.J. Sejnowski. Statistical constraints on synaptic plasticity. *J Theor Biol*, 21:385–9, 1977.

[7] Elie L. Bienenstock, Leon N. Cooper, and Paul W. Munro. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J Neuroscience*, 2:32–48, 1982.

[8] Patrick D. Roberts and Curtis C. Bell. Spike timing dependent synaptic plasticity in biological systems. *Biol Cybern*, 87:392–403, 2002.

[9] N. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, Amsterdam, 1981.

[10] Desai N.S. Rutherford L.C. Turrigiano G.G., Leslie K.R. and Nelson S.B. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391:892–896, 1998.

[11] W. B. Levy and D. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8:791–797, 1983.

[12] H. Markram, J. Lubke, M. Fortscher, and B. Sakmann. Regulation of synaptic efficacy by conicidence of postsynaptic AP's and EPSP's. *Science*, 275:213–215, 1997.

[13] C.C. Bell, V.Z. Han, Y. Sugawara, and K. Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387:278–281, May 1997.

[14] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford Unitversity Press, Oxford, 1995.

[15] B. Widrow and M. Hoff, Jr. Adaptive switching circuits. In *IRE WESCON Conv Rec part 5*, pages 96–104, 1960.

[16] Halbert White. Learning in artificial neural networks: A statistical perspective. *Neural Comp*, 1:425–464, 1989.

[17] Chung-Ming Kuan and Kurt Hornik. Convergence of learning algorithms with constant learning rates. *IEEE Trans Neural Networks*, 2:484–489, 1991.

[18] Dick Bedeaux, Katja Lakatos-Lindenberg, and Kurt E. Shuler. On the relation between master equations and random walks and their solutions. *Journal of Mathematical Physics*, 12:2116–2123, 1971.

[19] C.W. Gardiner. *Handbook of Stochastic Methods, 2nd Ed.* Springer-Verlag, Berlin, 1990.

[20] H. Risken. *The Fokker-Planck Equation*. Springer-Verlag, Berlin, 1989.

[21] G. Radons, H.G. Schuster, and D. Werner. Fokker-Planck description of learning in backpropagation networks. In *International Neural Network Con-*

ference - INNC 90, Paris, pages II 993–996. Kluwer Academic Publishers, July 1990.

[22] G. Radons. On stochastic dcynamics of supervised learning. *J Phys A*, 26:3445–3461, 1993.

[23] Genevieve B. Orr and Todd K. Leen. Weight space probability densities in stochastic learning: II. Transients and basin hopping times. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[24] Lars Kai Hansen, Raj Pathria, and Peter Salamon. Stochastic dynamics of supervised learning. *J Phys A*, pages 63–71, 1993.

[25] R. Der and Th. Villmannn. Dynamics of self organized feature mapping. Technical report, Universitat Leipzig, Inst. f. Informatik, 1993.

[26] Genevieve B. Orr. *Dynamics and Algorithms for Stochastic Search*. PhD thesis, Oregon Graduate Institute, October 1996.

[27] Todd K. Leen and John E. Moody. Weight space probability densities in stochastic learning: I. Dynamics and equilibria. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[28] H. Cateau and T. Fukai. A stochastic method to predict the consequence of arbitrary forms of spike-timing-dependent plasticity. *Neural Comp*, 15(3):597–620, 2003.

[29] Adam Kepecs, Mark C.W. van Rossum, Sen Song, and Jesper Tegney. Spike-timing-dependent plasticity: common themes and divergent vistas. *Biol Cybern*, 2002.

[30] Naoki Masuda and Kazuyuki Aihara. Self-organizing dual coding based on spike-time-dependent plasticity. *Neural Comp*, 16:627–663, 2004.

[31] Anothony N. Burkitt, Hamish Meffin, and David B. Grayden. Spike-timing-dependent plasticity: the relationship to rate-based learning for models with weight dynamics determined by a stable fixed point. *Neural Comp*, 16:855–940, 2004.

[32] Li I. Zhang, Huizhong W. Tao, Christine E Holt, William A Harris, and Mu ming Poo. A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395:37–42, 1998.

[33] D. Debanne, BH Gahwiler, and S.M. Thompson. Heterogeneity of synaptic plasticity at unitary CA1-CA3 and CA3-CA3 connections in rat hippocampal slice cultures. *J Neuroscience*, 19:10664–10671, 1999.

[34] J. Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In A.P. Dawid J.M. Bernardo, J. Berger and A.F.M. Smith, editors, *Bayesian Statistics 4*, pages 169–193. Oxford University Press, Oxford, 1992.

[35] Todd K. Leen. Exact and perturbation solutionhs for the ensemble dyamics. In David Saad, editor, *On-line Learning in Neural Networks*, pages 45–61. Cambridge University Press, Cambridge, 1999.