A KINETIC THEORY APPROACH TO CAPTURING INTERNEURONAL CORRELATION IN FEED-FORWARD NETWORKS

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

CHIN-YUEH LIU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR of PHILOSOPHY

DUANE Q. NYKAMP, Advisor

July, 2009

Acknowledgements

I would like to express the deepest appreciation to my supervisor, Professor Duane Q. Nykamp, for his advice, support and encouragement throughout my graduate studies at University of Minnesota. He was always generous with his time, and it was during our many long and insightful discussions that he helped to cultivate my interest in Mathematical and Computational Neuroscience. Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my instructors of Theoretical Neuroscience and Neural Engineering, Professor A. David Redish and Professor Theoden Netoff, for their guidance to the Neuroscience field which provides me necessary knowledge for my graduate studies in interdisciplinary research. I also thank my committee members, Professor Hans G. Othmer, Professor John Baxter, and Professor Theoden Netoff for their time to have helpful discussions of my thesis work and suggestions.

I acknowledge the financial support of the Ministry of Education of Taiwan and the fellowship of Neuro-physical-computational Sciences Graduate Training from Graduate Program in Neuroscience at University of Minnesota during my graduate work.

Finally, I want to thank my family members and especially my girlfriend Hsin-Yun Tsai for supporting and encouraging me to pursue this degree. Without my girlfriend's encouragement and company, I would not have finished the degree.

Dedication

For my Lord Jesus Christ and my family, who offered me unconditional love.

Abstract

We present an approach for using kinetic theory to capture first and second order statistics of neuronal activity. We coarse grain neuronal networks into populations of neurons and calculate the population average firing rate and output cross-correlation in response to time varying correlated input. We initially derive coupling equations for the populations based only on first and second order statistics of neuronal activity and the network connectivity. This coupling scheme is based on the hypothesis that second order statistics of the network connectivity are sufficient to determine second order statistics of neuronal activity. Using this coupling scheme, we implement a kinetic theory representation of a simple feed-forward network and demonstrate that this kinetic theory model captures key aspects of the emergence and propagation of correlations in the network, as long as the correlations do not become too strong. By analyzing the correlated activity of feed-forward networks with a variety of connectivity patterns, we provide evidence supporting our hypothesis of the sufficiency of second order connectivity statistics. To improve the kinetic theory performance under high correlation in feed-forward networks, we propose an inference method to estimate the rate of synchronous firing by more than two neurons. Then we include the effect of such events in the evolution of the postsynaptic populations by deriving improved coupling equations for populations. With these improved coupling equations, we obtain an improved kinetic theory representation of the simple feed-forward

network. To implement it, we make truncation approximations at different levels in the input and demonstrate that our improved kinetic theory model can capture the behavior of first and second order firing activity under higher correlation.

Table of Contents

List of Figures

Chapter 1

Introduction

Understanding how the brain's neuronal networks perform computations remains a difficult challenge. The interaction of large populations of neurons leads to a complex repertoire of high-dimensional activity patterns that is difficult to analyze. One possibility to reduce the dimensionality and complexity of such networks is to ignore high order interactions among neurons and simply analyze the consequences of low order interactions. Stopping at second order interactions may provide a good description, as recent evidence suggests that pairwise firing statistics among neurons may be sufficient to capture most of the higher order firing patterns (Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu et al, 2008).

We present a kinetic theory approach that is designed to capture the second order interactions among neurons (Liu and Nykamp, 2009). Kinetic theory approaches have been used to model gases and plasmas, where one uses moment closure approximations to derive equations for lower order statistics of a system of particles (Ichimaru, 1973; Nicholson, 1992). We implement a similar approach to track the second order statistics among neurons in a population while explicitly neglecting third and higher order statistics.

1.1 Motivation for approach

To understand the behavior of a neuronal network, one may be interested in uncovering the relationship between the network structure and the behavior. Modeling neuronal networks is challenging as the set of possible connectivity combinations is enormous and we don't have experimental methods to determine the fine details of connectivity. Even if we could exactly determine the connectivity of the network, one would still be faced with the challenge of determining what features of the network underlie the behavior of interest. As the exact connectivity of a given network within the brain may vary widely among individuals while the behavior of interest is maintained, all the details of the connectivity may not be important for the behavior. One would like a method to distill the connectivity down to its key features and to study how these features influence the behavior.

Recent experimental results suggest an approach that may be promising. A number of labs have provided evidence that the second order statistics of interneuronal firing patterns may be sufficient to describe a large fraction of the higher order firing patterns (Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu et al, 2008). In a similar manner, we hypothesize that the second order statistics of connectivity patterns may be sufficient to explain a large fraction of the behavior of the brain's neuronal networks. If the second order connectivity statistics were sufficient to explain the second order statistics of firing patterns, then the above experimental results suggest they should be sufficient to explain a large fraction of the higher order firing patterns of the brain's neuronal networks.

If second order connectivity statistics do capture much of the relevant network behavior, then one can greatly simplify the space of connectivity patterns, parameterizing them by just their second order statistics. One could attempt to understand

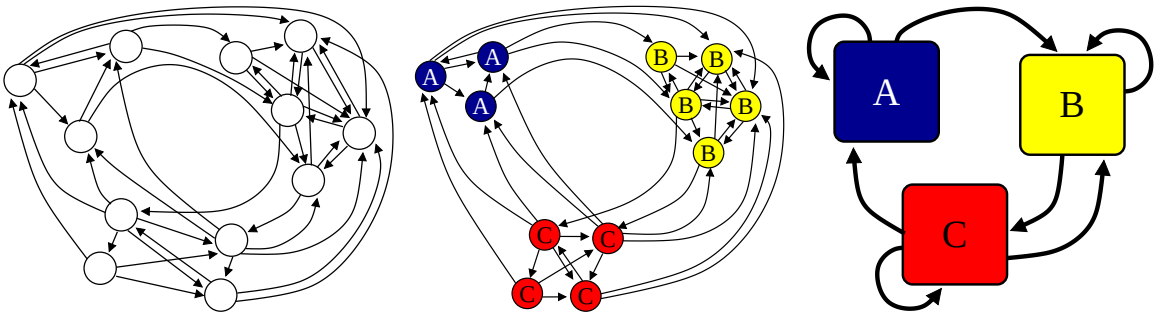


Figure 1.1: Schematic illustration of the procedure for forming kinetic theory approximations of neuronal networks. Starting with a neuronal network (left), we group neurons into populations, which we indicate by letters (middle). For each population, we represent first and second order statistics of neuronal activity with population density functions (right). The coupling among the population density functions is based on first and second order statistics of the original network connectivity.

network behavior by exploring the consequences of these second order connectivity statistics. To do so, one would like a method to analyze network behavior as a function of just the second order connectivity statistics while imposing as little as possible additional structure on the higher order connectivity patterns.

One approach is to use kinetic theory to develop tools to study second order statistics of the activity and connectivity of neuronal networks. With kinetic theory, one can explicitly ignore higher order statistics and simply model second order distributions among the variables of interest. Then, one can use maximum entropy methods (Jaynes, 1957) to infer higher order distributions with minimal structure.

1.2 Principles underlying our kinetic theory implementation

Our kinetic theory implementation begins with a coarse graining step where we group neurons into populations (see left two panels of Figure 1.1). Within each population,

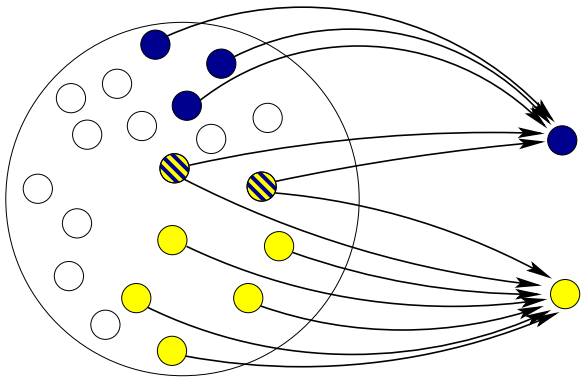


Figure 1.2: Illustration of the second order statistics of connectivity used in our kinetic theory approach. We consider the connections from a presynaptic population (left group) onto a pair of neurons from a postsynaptic population (right). The first order statistic W^1 is the expected number of neurons from the presynaptic population that project to a single postsynaptic neuron (six in this illustration). The second order statistic W^2 is the expected number of presynaptic neurons that project to both postsynaptic neurons (two in this case). The fraction of shared inputs parameter is their ratio $\beta = W^2/W^1$ (which is 1/3 in this case).

we will assume the neurons have identical properties and identical statistics, so this step clearly leads to a loss of detail. For each population, we will track the distribution of neurons over state space with a density function that we call a population density function (Nykamp and Tranchina, 2000). However, unlike previous approaches, we will not assume that the neurons within a population are independent. Instead, we explicitly represent the joint distribution of any pair of neurons with the population density $\rho(\mathbf{x}_1, \mathbf{x}_2, t)$, where each vector $\mathbf{x}_i$ represents a value of the state variables for a single neuron. Roughly speaking, the quantity $\rho(\mathbf{x}_1, \mathbf{x}_2, t)d\mathbf{x}_1 d\mathbf{x}_2$ represents the probability that the state variables $\mathbf{X}_i(t)$ and $\mathbf{X}_j(t)$ at time t of two neurons in the population are near the values $\mathbf{x}_1$ and $\mathbf{x}_2$. Since we assume this holds for any pair of neurons, the population density $\rho(\mathbf{x}_1, \mathbf{x}_2, t)$ clearly must be symmetric in $\mathbf{x}_1$ and $\mathbf{x}_2$.

Once we form a population density $\rho_j(\mathbf{x}_1, \mathbf{x}_2, t)$ for each coarse-grained population j, we need to determine the second order statistics for the connectivity of the popula-

tions. For our kinetic theory implementation, we will use the following two statistics: (see Figure 1.2):

1. W^1_{jk}, the average number of neurons from population j that project onto a single neuron in population k (a first order statistic), and

2. W^2_{jk}, the average number of neurons from population j that simultaneously project onto a pair of neurons in population k (a second order statistic).

In presenting the results, we will typically use the ratio $\beta_{jk} = W^2_{jk}/W^1_{jk}$ for the second order statistic. We refer to β_{jk} as the fraction of shared input parameter.

To derive coupling equations for the network of population densities, one must develop a method to capture the effect of the connectivity statistics on the interactions among the population density functions. Then, one can complete the simplification of the original network (e.g., left of Figure 1.1) into a kinetic theory network of interacting population densities (e.g., right of Figure 1.1).

The goal is to develop a kinetic theory network approach where the population density functions capture the second order statistics in the neuronal activity and the interaction terms capture the second order statistics of the connectivity. Such a tool could be used to explore the consequences of connectivity in a setting where both the connectivity and the neuronal activity are highly simplified and lower dimensional (Nirenberg and Victor, 2007) due to the neglect of higher order statistics. This simplified framework may facilitate exploration of the key features of connectivity that underlie network behavior.

In this thesis, we have more modest goals than alluded to above. As an initial test of the potential of this approach, we focus on developing a kinetic theory approach for modeling the emergence and propagation of correlations through excitatory feed-forward networks of integrate-and-fire neurons. In chapter 2, we develop our initial

feed-forward kinetic theory model, gauge its performance, and test our hypothesis of the sufficiency of second order connectivity statistics. In chapter 3, we develop an improved kinetic theory approach for feed-forward networks and gauge its performance under high correlation. We make some conclusions and future research directions in chapter 4.

Chapter 2

Initial kinetic theory approach

As an initial test of our kinetic theory approach to capturing second order statistics, we seek to develop a second order kinetic theory description of excitatory feed-forward networks of integrate-and-fire neurons. Feed-forward networks are ideal test system for this approach for a few reasons. First, we study feed-forward networks that are naturally divided into layers, and we can group all neurons of a layer into a single population with little adverse effect of the coarse-graining. Second, neurons within each population do not interact, so one can derive a simplified kinetic theory description. Third, it is well-known that correlations emerge and propagate through feed-forward networks (Diesmann et al, 1999), and the feed-forward networks will be a good test case to see if the kinetic theory model correctly captures this build-up of correlation.

2.1 The integrate-and-fire model

To keep the kinetic theory equations as simple as possible, we base our implementation on a simplified integrate-and-fire model of neuron dynamics. We let $V_j(t)$ be the

voltage of neuron j at time t and let the voltage evolve according to the stochastic differential equation

$$\frac{dV_j}{dt} = \frac{E_r - V_j}{\tau} + \sum_i A_j^i \delta(t - T_j^i), \tag{2.1}$$

where τ is the membrane time constant, E_r is the resting equilibrium potential, and $\delta(t)$ is the Dirac delta function. At the times T_j^i of excitatory synaptic input, the voltage jumps up by the amount A_j^i, which is a random variable with probability density function $f_A(x)$. We define spike times t_{sp} as those times when $V(t)$ crosses the firing threshold v_{th}, and we immediately reset the voltage to $V(t_{sp}^+) = v_{\text{reset}}$, where $v_{\text{reset}} < E_r < v_{\text{th}}$. We use this model because the state of the neuron is described by just the single state variable $V(t)$.

We assume the arrival times T_j^i of the input for each neuron j are given by a modulated Poisson process, where the rate is identical for each neuron in a given population. In departure from previous kinetic theory implementations, we allow the inputs to any pair of neurons in the population to be correlated, though for simplicity, we only model instantaneous correlations in the input. We assume that the inputs to any pair of neurons are given by independent Poisson processes to each neuron at rate $\nu_{\text{ind}}(t)$ combined with synchronous input to both neurons from a third independent Poisson process at rate $\nu_{\text{syn}}(t)$. This correlated input will create correlations among the neurons in the population, which we will represent with a population density function. A schematic of the random walk exhibited by a pair of neurons with this input is shown in Figure 2.1. The random walk includes the decay toward E_r in between inputs as well as the voltage reset to v_{reset} after crossing the threshold v_{th}.

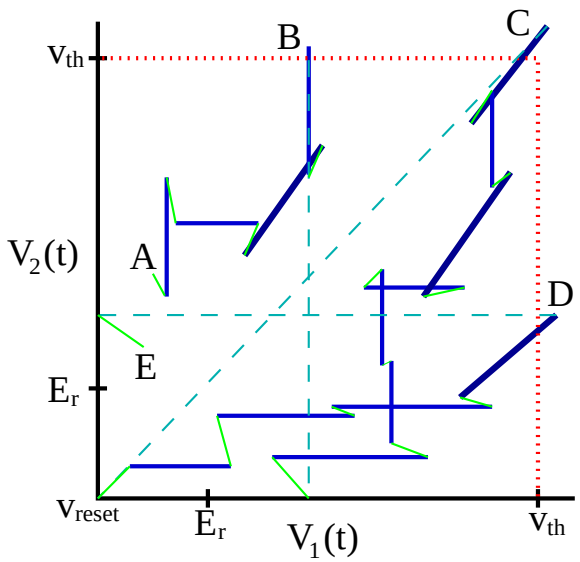


Figure 2.1: Schematic of random walk of the voltages of a pair of neurons. The neuron voltages $(V_1(t), V_2(t))$ start at the point indicated by A. Neuron 2 fires alone at B, the neurons fire synchronously at C, neuron 1 fires alone at D, and the voltages end at E. Synchronous input jumps are indicated by the thick diagonal lines and independent input jumps are indicated by horizontal or vertical solid lines. Voltage decay toward (E_r, E_r) in between inputs is indicated by the thin diagonal lines. After one or more neurons fire by crossing the threshold v_{th} (dotted line), the voltage is reset to v_{reset}, as indicated by the dashed lines. For a large population of neurons, the voltages $(V_j(t), V_k(t))$ of each neuron pair can be viewed as following such a random walk, and the population density $\rho(v_1, v_2, t)$ captures the fraction of neuron pairs in the population with voltages $(V_j(t), V_k(t))$ around (v_1, v_2).

2.2 The kinetic theory equations

To form a population density function, we assume all neurons in the population are identical and that every pair has the same second order statistics. We represent the second order statistics among the state variables $V_j(t)$ by the population density function $\rho(v_1, v_2, t)$, defined by

$$\int_\Omega \rho(v_1, v_2, t) dv_1 dv_2 = \Pr((V_j(t), V_j(t) \in \Omega)$$

where Ω is any region in the v_1-v_2 phase plane and $V_j(t)$ and $V_k(t)$ are the voltages of any two neurons in the population. For simplicity, we will refer to the voltages as $V_1(t)$ and $V_2(t)$. For a large population of neurons, one can view $\int_\Omega \rho(v_1, v_2, t) dv_1 dv_2$ as the fraction of neuron pairs with voltages in the region Ω.

Since the input to a pair of neurons is a correlated Poisson process, the history of the synaptic input does not influence future evolution of the voltage and the evolution of the voltage pair $(V_1(t), V_2(t))$ is a Markov process. The evolution of $(V_1(t), V_2(t))$ contains the deterministic evolution of the voltages toward E_r due to the leakage term from Eq. (2.1), and it contains the jump processes due to independent and synchronous inputs, as well as reset to v_{reset} after spiking. We can write down the differential Chapman-Kolmogorov equation (Gardiner, 2004) to describe these processes

as follows:

$$\begin{aligned}
\frac{\partial \rho}{\partial t}(v_1, v_2, t) = & \frac{1}{\tau}\frac{\partial}{\partial v_1}\left[(v_1 - E_r)\rho(v_1, v_2, t)\right] \\
& + \frac{1}{\tau}\frac{\partial}{\partial v_2}\left[(v_2 - E_r)\rho(v_1, v_2, t)\right] \\
& + \nu_{\text{ind}}(t)\left[\int_{v_{\text{reset}}}^{v_1} f_A(v_1 - \theta_1)\rho(\theta_1, v_2, t)d\theta_1 - \rho(v_1, v_2, t)\right] \\
& + \nu_{\text{ind}}(t)\left[\int_{v_{\text{reset}}}^{v_2} f_A(v_2 - \theta_2)\rho(v_1, \theta_2, t)d\theta_2 - \rho(v_1, v_2, t)\right] \\
& + \nu_{\text{syn}}(t)\Bigg[\int_{v_{\text{reset}}}^{v_1}\int_{v_{\text{reset}}}^{v_2} f_A(v_1 - \theta_1)f_A(v_2 - \theta_2)\rho(\theta_1, \theta_2, t)d\theta_2 d\theta_1 \\
& \qquad\qquad - \rho(v_1, v_2, t)\Bigg] \\
& + \delta(v_1 - v_{\text{reset}})J_{\text{reset},1}(v_2, t) + \delta(v_2 - v_{\text{reset}})J_{\text{reset},2}(v_1, t) \\
& + \delta(v_1 - v_{\text{reset}})\delta(v_2 - v_{\text{reset}})J_{\text{reset},3}(t). \qquad (2.2)
\end{aligned}$$

The first two lines of Eq. (2.2) contain the advection terms due to the leak current of Eq. (2.1) that draws each voltage toward the rest potential E_r. The third line describes the jumps due to independent input to neuron 1. The integral contains the contribution to $\rho(v_1, v_2, t)$ due to a neuron with voltage $V_1(t) = \theta_1$ receiving an input of size $v_1 - \theta_1$ so that it lands at $V_1(t) = v_1$. The second term of that line is the loss of probability at $\rho(v_1, v_2, t)$ due to a neuron with voltage $V_1(t) = v_1$ receiving an independent input and jumping to a higher voltage. The fourth line of Eq. (2.2) is the symmetric term for independent input to neuron 2. The fifth line describes the jumps in both neurons due to synchronous input. The integral contains the contribution to $\rho(v_1, v_2, t)$ when a neuron pair with voltages $(V_1(t), V_2(t)) = (\theta_1, \theta_2)$ receives synchronous input of size $(v_1 - \theta_1, v_2 - \theta_2)$ that jumps the voltages to $(V_1(t), V_2(t)) = (v_1, v_2)$. The second term is due to neuron pairs with voltages at $(V_1(t), V_2(t)) = (v_1, v_2)$ receiving synchronous input that jumps both

voltages higher.

The last two lines of Eq. (2.2) contain terms due to the reset of voltage to $V(t) = v_{\text{reset}}$ immediately after a neuron fires a spike. The factors $J_{\text{reset},k}$ are defined by

$$\begin{aligned}
J_{\text{reset},1}(v_2,t) &= \nu_{\text{ind}}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}} F_A(v_{\text{th}}-\theta_1)\rho(\theta_1,v_2,t)d\theta_1 \\
&\quad + \nu_{\text{syn}}(t)\int_{v_{\text{reset}}}^{v_2}\int_{v_{\text{reset}}}^{v_{\text{th}}} F_A(v_{\text{th}}-\theta_1)f_A(v_2-\theta_2)\rho(\theta_1,\theta_2,t)d\theta_1 d\theta_2, \\
J_{\text{reset},2}(v_1,t) &= \nu_{\text{ind}}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}} F_A(v_{\text{th}}-\theta_2)\rho(v_1,\theta_2,t)d\theta_2 \\
&\quad + \nu_{\text{syn}}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}}\int_{v_{\text{reset}}}^{v_1} F_A(v_{\text{th}}-\theta_2)f_A(v_1-\theta_1)\rho(\theta_1,\theta_2,t)d\theta_1 d\theta_2, \\
J_{\text{reset},3}(t) &= \nu_{\text{syn}}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}}\int_{v_{\text{reset}}}^{v_{\text{th}}} F_A(v_{\text{th}}-\theta_1)F_A(v_{\text{th}}-\theta_2)\rho(\theta_1,\theta_2,t)d\theta_1 d\theta_2, \qquad (2.3)
\end{aligned}$$

where $F_A(x)$ is the complementary cumulative distribution function of the random jump size A, that is, $F_A(x) = \int_x^\infty f_A(t)dt = \Pr(A > x)$. The first integral of $J_{\text{reset},1}(v_2,t)$ reflects the event that a neuron with $V_1(t) = \theta_1$ receives an independent input that jumps the voltage past the threshold v_{th}. The neuron is reset to $V_1(t) = v_{\text{reset}}$ while $V_2(t)$ stays at v_2. The second integral of $J_{\text{reset},1}(v_2,t)$ reflects the event that a pair of neurons with $(V_1(t), V_2(t)) = (\theta_1,\theta_2)$ receives a simultaneous input that jumps $V_1(t)$ past threshold and $V_2(t)$ to the subthreshold voltage v_2. $J_{\text{reset},2}(v_1,t)$ is identical to $J_{\text{reset},1}(v_2,t)$ with the roles of the neurons reversed. $J_{\text{reset},3}(t)$ reflects the event that a pair of neurons with $(V_1(t), V_2(t)) = (\theta_1,\theta_2)$ receives a simultaneous input that jumps both neurons past threshold. In this case, both voltages are reset to v_{reset}. The neuron firings contributing to the $J_{\text{reset},k}$ are illustrated in Figure 2.2.

The addition of the reset terms makes Eq. (2.2) a conservative system so that the integral of ρ remains constant, which we fix to 1. However, the equation is not written in conservative form. To aid in developing a conservative numerical method

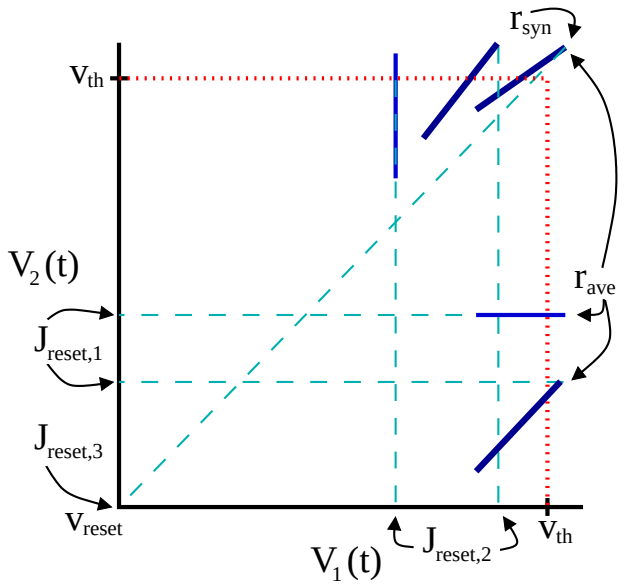


Figure 2.2: Illustration of the three reset terms and their relationship to the average and synchronous firing rates. $J_{\text{reset},1}$ corresponds to the reset to v_{reset} after neuron 1 fires alone due to independent or synchronous input. $J_{\text{reset},2}$ is the equivalent for neuron 2. $J_{\text{reset},3}$ corresponds to the reset of both neurons to v_{reset} after synchronous input causes the neurons to simultaneously cross threshold. The synchronous firing rate r_{syn} is simply $J_{\text{reset},3}$. We can obtain the average firing rate r_{ave} by adding all the ways neuron 1 can fire alone (all the $J_{\text{reset},1}$) to the synchronous firing rate. Plotting convention is the same as in figure 2.1.

to solve Eq. (2.2), we first rewrite the equation in conservative form, and then base our numerical scheme on the conservative form, as discussed in Appendix A.

2.3 Total firing rate and synchronous firing rate

Since the reset terms described above reflect the voltage reset after firing, they can be used to read out the firing rate of population. If we denote $V_j(T^+) = \lim_{t\to T^+} V_j(t)$, we can interpret the reset terms as

$$\begin{aligned} J_{\text{reset},1}(v_2,t)dv_2dt &= \Pr(\text{neuron 1 fires at } T\in(t,t+dt)\text{and } V_2(T^+)\in(v_2,v_2+dv_2)),\\ J_{\text{reset},2}(v_1,t)dv_1dt &= \Pr(\text{neuron 2 fires at } T\in(t,t+dt)\text{and } V_1(T^+)\in(v_1,v_1+dv_1)),\\ J_{\text{reset},3}(t)dt &= \Pr(\text{both neurons simultaneouslyfire at } T\in(t,t+dt)). \end{aligned} \tag{2.4}$$

Note that the factors $J_{\text{reset},1}(v_2,t)dv_2$, $J_{\text{reset},2}(v_1,t)dv_1$, and $J_{\text{reset},3}(t)$ indicate the probability of a spike per unit time. If neuron 1 fires a spike, the spike will be reflected in $J_{\text{reset},3}$ if neuron 2 fires simultaneously; otherwise the spike will be reflected in $J_{\text{reset},1}$. Therefore, the average firing rate for a neuron is simply the sum of $J_{\text{reset},3}(t)$ plus all of the $J_{\text{reset},1}(v_2,t)$,

$$r_{\text{ave}}(t) = \int_{v_{\text{reset}}}^{v_{\text{th}}} J_{\text{reset},1}(v_2,t)dv_2 + J_{\text{reset},3}(t). \tag{2.5}$$

(By symmetry, we could have equally well defined $r_{\text{ave}}(t)$ by replacing $J_{\text{reset},1}$ with $J_{\text{reset},2}$, obtaining the identical firing rate of neuron 2.)

We also define the synchronous firing rate, which is the rate at which both neurons

fire simultaneously. Given Eq. (2.4), the synchronous firing rate is clearly

$$r_{\text{syn}}(t) = J_{\text{reset},3}(t). \tag{2.6}$$

Since the reset terms $J_{\text{reset},k}$ are calculated in the course of solving for $\rho(v_1, v_2, t)$, we can easily obtain the firing rates $r_{\text{ave}}(t)$ and $r_{\text{syn}}(t)$. The threshold crossings corresponding to the average and synchronous firing rates are illustrated in Figure 2.2.

2.4 Capturing output correlation

The average firing rate $r_{\text{ave}}(t)$ captures the first order statistics of the population output, and the synchronous firing rate $r_{\text{syn}}(t)$ captures one type of second order statistic of the output, namely correlation with zero delay. However, $r_{\text{syn}}(t)$ does not capture all of the correlation between two neurons. If a pair of neurons receives independent Poisson input at rate $\nu_{\text{ind}}(t)$ and synchronous Poisson input at rate $\nu_{\text{syn}}(t)$, their spikes may become correlated at some non-zero delay, which is not captured by $r_{\text{syn}}(t)$.

Consider the snapshot of $\rho(v_1, v_2, t)$ in the left panel of Figure 2.3. The synchronous input $\nu_{\text{syn}}(t)$ has increased the probability of the voltage combinations $(V_1(t), V_2(t))$ close to the diagonal, so that the positive correlation between $V_1(t)$ and $V_2(t)$ is clearly seen. For this reason, the probability for $(V_1(t), V_2(t))$ to be in the upper right corner (point A in middle panel of Figure 2.3) is higher than it would be if the neurons were independent. If the neurons received a synchronous input while their voltages were near the upper right corner, they would be highly likely to simultaneously spike and reset (points B→C), contributing to the synchronous firing rate $r_{\text{syn}}(t)$. However, if the neurons each received independent input not exactly at the

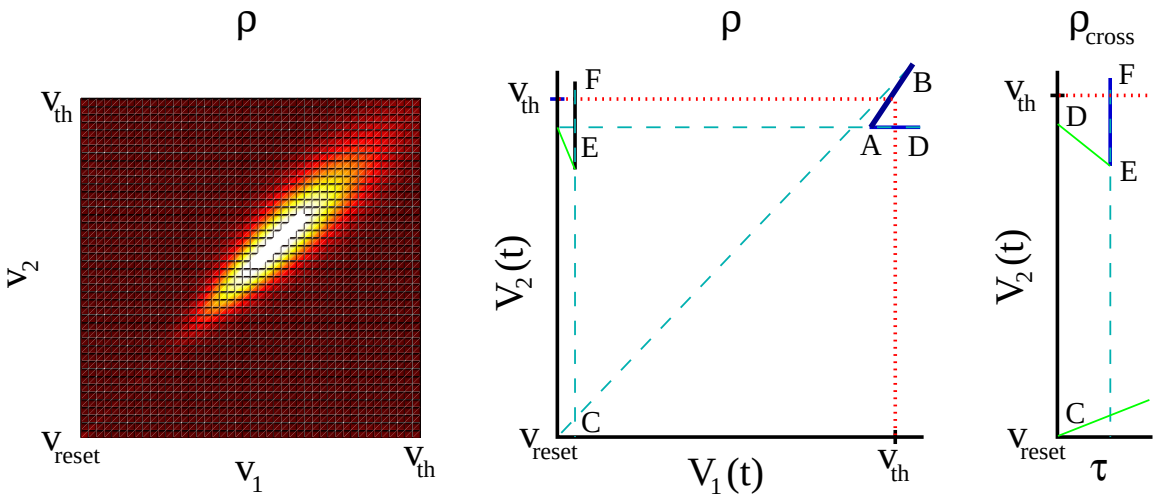


Figure 2.3: Left: Pseudocolor plot of highly correlated population density function $\rho(v_1, v_2, t)$. Light colors correspond to high probability. Middle: Illustration of delayed correlation that is likely to result from a correlated population density. Given the correlated density, the likelihood that the voltages $(V_1(t), V_2(t))$ of a pair of neurons is in the upper right corner (A) is higher than if the voltages were independent. Receiving synchronous input might lead to synchronous firing and reset of the neuron pair (A→B→C). However, if the neurons each received independent input, the neurons would be likely to fire within a short delay (A→D→E→F). After the first neuron fires, the voltage is reset to the upper left corner (E), and the fact that the first neuron fired recently is lost. Right: Illustration of method to track delayed correlation. After the first neuron fires and the voltage pair is reset to (v_{reset}, v_2) in $\rho(v_1, v_2, t)$, the firing is simultaneously recorded by injecting into $\rho_{\text{cross}}(v_2, \tau; t)$ at $(v_2, 0)$ (point D on right diagram). As the voltage of the second neuron evolves (D→E), the time since the first neuron fired is tracked by τ. When the second neuron fires (E→F), the pair of spikes with delay τ (i.e., at times t and $t + \tau$) is recorded by the crossing of threshold in $\rho_{\text{cross}}(v_2, \tau; t)$ at the point (v_{th}, τ). If the neurons had fired simultaneously, we would record this by injecting into $\rho_{\text{cross}}(v_2, \tau; t)$ at $(v_{\text{reset}}, 0)$ (point C on right diagram). Plotting convention of the right two panels is the same as in figure 2.1.

same time, then the neurons would be highly likely to fire with a short delay between their firing times (A→D→E→F in middle panel of Figure 2.3). With a voltage distribution such as in left panel of Figure 2.3, this independent input would still lead to correlations in the firing between the neurons because they would be more likely to fire within a short delay of each other than predicted by an independent distribution.

We cannot directly read out this delayed correlation from $\rho_2(v_1, v_2, t)$. If the voltage pair $(V_1(t), V_2(t))$ is near the upper right corner and the first neuron received an independent input, it would be highly likely to fire by itself, after which its voltage would be reset to v_{reset}. The voltage pair $(V_1(t), V_2(t))$ jumps to the upper left corner of $\rho_2(v_1, v_2, t)$ (point E in middle panel of Figure 2.3), and the fact that the first neuron had just spiked is lost. Even if the second neuron receives an input shortly thereafter and fires a spike (point F), there is no way of determining the spike time of the first neuron from the fact that $(V_1(t), V_2(t))$ crosses the upper boundary of $\rho_2(v_1, v_2, t)$ somewhere near the upper left corner.

To read out the delayed correlation, we construct another density, which we call $\rho_{\text{cross}}(v_2, \tau; t_0)$, defined so that $\rho_{\text{cross}}(v_2, \tau; t_0)/r_{\text{ave}}(t_0)$ is the probability density of $V_2(t_0 + \tau)$ conditioned on neuron 1 firing at time t_0 (right panel of Figure 2.3). To compute the evolution of $\rho_{\text{cross}}(v_2, \tau; t_0)$ with respect to τ for a fixed t_0, we initialize it with the distribution of neuron 2 conditioned on neuron 1 spiking, i.e., $\rho_{\text{cross}}(v_2, 0; t_0) = J_{\text{reset},1}(v_2, t_0) + \delta(v_2 - v_{\text{reset}}) J_{\text{reset},3}(t_0)$. (This corresponds to injecting points C and D in the right panel of Figure 2.3.) Then, we evolve $\rho_{\text{cross}}(v_2, \tau; t_0)$ according a modified version of equation Eq. (2.2), where we integrated out v_1 and replaced t with $t_0 + \tau$. We then define $r_{\text{cross}}(\tau; t_0)$ as the reset term from this modified equation (the D→E threshold crossing and subsequent reset in the right panel of Figure 2.3). In this way, $r_{\text{cross}}(\tau; t_0)/r_{\text{ave}}(t_0)$ is the probability per unit time that neuron 2 fires at time $t_0 + \tau$ conditioned on neuron 1 firing at time t_0. We subtract

$r_{\text{ave}}(t_0+\tau)$, the marginal probability per unit time that neuron 2 fires at time $t_0+\tau$, and multiply by $r_{\text{ave}}(t_0)$. Finally, we add the zero delay correlation $r_{\text{syn}}(t_0)$ to obtain the cross correlation between the neurons at time t_0 with delay τ:

$$C(\tau;t_0) = \delta(\tau)r_{\text{syn}}(t_0) + r_{\text{cross}}(\tau;t_0) - r_{\text{ave}}(t_0)r_{\text{ave}}(t_0+\tau). \tag{2.7}$$

The cross-correlation has units of spikes per unit time squared and is the correlation between spikes of one neuron at time t_0 and the spikes of a second neuron at time $t_0+\tau$. We extend the definition for negative delays $-\tau$ to be the correlation between the spikes of one neuron at time $t_0-\tau$ and the spikes of another neuron at time t_0: $C(-\tau;t_0) = C(\tau;t_0-\tau)$. To summarize the correlation at a time t, we compute the area of the peak of positive correlation around delay 0,

$$C_{\text{peak}}(t) = \int_{-\tau_1}^{\tau_2} C(\tau;t)d\tau, \tag{2.8}$$

where $-\tau_1$ and τ_2 are the delays where the correlation first becomes negative: $\tau_1 = \min\{\tau > 0 \,|\, C(-\tau,t) \le 0\}$ and $\tau_2 = \min\{\tau > 0 \,|\, C(\tau,t) \le 0\}$.

2.5 Derivation of network equations

The kinetic theory equation Eq. (2.2) can be used to compute the evolution of a population of neurons to prescribed independent and synchronous input rates, $\nu_{\text{ind}}(t)$ and $\nu_{\text{syn}}(t)$ respectively. Along with the evolution of the population density $\rho(v_1,v_2,t)$, we also compute the output statistics: the average firing rate $r_{\text{ave}}(t)$, the synchronous firing rate $r_{\text{syn}}(t)$, and the output correlation $C(\tau;t)$. To apply our results to a feed-forward network, we need a method to transform the output of a presynaptic population to the input of a postsynaptic population, taking into account the statistics

of the connectivity.

As mentioned in Section 1.2, we will characterize the connectivity between two populations by both first and second order statistics. The first order statistic is W_{jk}^1, the expected number of presynaptic neurons from population j that project to any postsynaptic neuron in population k. The second order statistics is W_{jk}^2, the expected number of presynaptic neurons from population j that simultaneously project to any pair of postsynaptic neurons in population k. Our goal is to compute the input rates $\nu_{\text{ind}}^k(t)$ and $\nu_{\text{syn}}^k(t)$ to population k based on these connectivity statistics, the output of each population j, and any external independent input to population k at rate $\nu_{\text{ext}}^k(t)$. We assume that neurons from different populations are uncorrelated and neurons within a population are uncoupled.

We consider a pair of neurons in population k. Let N_1^j be the number of neurons from population j that project onto the first neuron and N_2^j be the number of neurons that project onto the second neuron. Furthermore, let N_3^j be the number of neurons from population j that project onto both postsynaptic neurons. (In the illustration of Figure 1.2, $N_1^j = 5$, $N_2^j = 7$ and $N_3^j = 2$.) We view N_1^j, N_2^j, and N_3^j as three random numbers with expected values specified by the connectivity statistics: $E(N_1^j) = E(N_2^j) = W_{jk}^1$ and $E(N_3^j) = W_{jk}^2$. To calculate the input rates $\nu_{\text{ind}}^k(t)$ and $\nu_{\text{syn}}^k(t)$, we will first calculate them conditioned on particular values of $\mathbf{N}_{\text{in}} = \{N_i^j\}$ and then take the expected values over the $\mathbf{N}_{\text{in}}$.

Define $\nu_{\text{syn}}^k(t; \mathbf{N}_{\text{in}})$ to be the rate of synchronous input onto neurons 1 and 2,

conditioned on particular values of the $\mathbf{N}_{\text{in}}$. We calculate that this input rate is

$$\begin{aligned}\nu^k_{\text{syn}}(t;\mathbf{N}_{\text{in}}) &= \sum_j N_3^j (r^j_{\text{ave}}(t) - r^j_{\text{syn}}(t)) \\ &\quad + \sum_j \Bigg[(N_1^j - N_3^j)(N_2^j - N_3^j) + (N_1^j - N_3^j)N_3^j \\ &\qquad + (N_2^j - N_3^j)N_3^j + 2\binom{N_3^j}{2} \Bigg] r^j_{\text{syn}}(t) \\ &= \sum_j \left[N_3^j r^j_{\text{ave}}(t) + (N_1^j N_2^j - 2N_3^j) r^j_{\text{syn}}(t) \right] \end{aligned} \tag{2.9}$$

The neurons will receive synchronous input any time one of the N_3^j neuron projecting to both neurons fires a spike all by itself (i.e., no other of the presynaptic neurons fires synchronously with it). Since we are calculating only up to second order statistics, we can approximate the probability that just a single neuron fires as $r^j_{\text{ave}}(t) - r^j_{\text{syn}}(t)$. Multiplying by the number N_3^j of common input neurons, we obtain the first sum of Eq. (2.9).

The second sum of Eq. (2.9) accounts for the synchronous input to neurons 1 and 2 that results from one of the N_1^j neurons projecting to neuron 1 firing synchronously with one of the N_2^j neurons projecting to neuron 2. This could happen in four ways, corresponding to the four terms in the square brackets. First, one of the $(N_1^j - N_3^j)$ neurons projecting to neuron 1 alone could fire synchronously with one of the $(N_2^j - N_3^j)$ neurons projecting to neuron 2 alone. Second, one of the $(N_1^j - N_3^j)$ neurons projecting to neuron 1 alone could fire synchronous with one of the N_3^j common input neurons. In this case, neuron 1 receives a double input that is synchronous to a single input to neuron 2. Since the kinetic theory implementation described in this chapter does not represent double inputs to any neuron, we represent this event just as synchronous input to the pair. Third, we handle in the same way the event where

one of the N_3^j common input neuron fires synchronously with one of the $(N_2^j - N_3^j)$ neurons projecting to neuron 2 alone. Fourth, if two of the N_3^j common input neurons fire synchronously, both neurons 1 and 2 receive synchronous double inputs. Since we do not represent double inputs at this point, we model the double synchronous inputs as two synchronous inputs. Hence, we multiply the number of pairs $\binom{N_3^j}{2}$ by two. All these combinations of inputs are multiplied by the probability $r_{\text{syn}}^j(t)$ that two input neurons fire synchronously.

The approximations used for the last three terms in the second sum of Eq. (2.9) can be justified under the condition that $N_3^j \ll N_1^j$ and $r_{\text{syn}} \ll r_{\text{ave}}$. If common input neurons are relatively rare and the synchrony isn't too high, then these last three terms will be small (quadratic in a small parameter) compared to the other terms of Eq. (2.9). For this reason, we expect the kinetic theory implementation of this chapter to perform best under conditions where the synchronous firing and fraction of shared input are not too high.

Next, we take the expected value of $\nu_{\text{syn}}^k(t; \mathbf{N}_{\text{in}})$ over $\mathbf{N}_{\text{in}}$. The only expected value not explicitly specified by our second order connectivity statistics is the product $E(N_1^j N_2^j)$. We simply assume that N_1^j and N_2^j are uncorrelated so that $E(N_1^j N_2^j) = E(N_1^j)E(N_2^j)$. We obtain that the expected rate of synchronous input is

$$\begin{aligned}
\nu_{\text{syn}}^k(t) &= E(\nu_{\text{syn}}^k(t; \mathbf{N}_{\text{in}})) \\
&= \sum_j W_{jk}^2 r_{\text{ave}}^j(t) + \sum_j (W_{jk}^1 W_{jk}^1 - 2W_{jk}^2) r_{\text{syn}}^j(t) \\
&= \sum_j \beta_{jk} W_{jk}^1 r_{\text{ave}}^j(t) + \sum_j W_{jk}^1 (W_{jk}^1 - 2\beta_{jk}) r_{\text{syn}}^j(t), \qquad (2.10)
\end{aligned}$$

where $\beta_{jk} = W_{jk}^2 / W_{jk}^1$ is the fraction of shared input parameter.

The average input to population k is simply the external input rate $\nu_{\text{ext}}^k(t)$ plus

the sum of the $W^1_{jk} r^j_{\text{ave}}(t)$, the average coupling times the average firing rate. As discussed in Section 2.4, the output of population j will contain correlations beyond the synchronous output of $r^j_{\text{syn}}(t)$. However, our kinetic theory equations do not include delayed correlation in the input. (We neglected delayed correlation to create a Markov process without adding additional state variables.) For our first approximation which we term KT0, we will simply assume that inputs that are not synchronous are independent. Under this approximation, the independent input rate to population k is simply the average input rate minus the synchronous input rate:

$$\nu^k_{\text{ind}}(t) = \nu^k_{\text{ext}}(t) + \sum_j W^1_{jk} r^j_{\text{ave}}(t) - \nu^k_{\text{syn}}(t). \tag{2.11}$$

This formula not only neglects delayed correlation, but it also approximates as independent any double input to a single neuron (such double inputs would occur when a pair of neurons projecting to the given neuron fire synchronously.)

2.6 Accounting for delayed correlation

The delayed correlation described in Section 2.4 may be more substantial than the instantaneous correlation reflected in $r_{\text{syn}}(t)$. In this case, we may severely underestimate the correlation by assuming all spikes not captured by $r_{\text{syn}}(t)$ are independent, as we did for deriving the coupling conditions (2.10) and (2.11). Hence, we developed another method to capture additional correlation in our kinetic theory implementation. We approximated delayed correlation in the output of a population as though it were correlation with zero delay.

Approximating delayed correlation as instantaneous is not as simple as adding $C_{\text{peak}}(t)$ of Eq. (2.8) to $r_{\text{syn}}(t)$. The cross-correlation peak area $C_{\text{peak}}(t)$ evaluated

at time t is based on output from times $t + \tau$ that are later than t. If we included $C_{\text{peak}}(t)$ in the input rates at time t, we would be attempting to use future information to calculate the evolution of the network at time t.

To avoid using future information, we calculate a modified version of the cross-correlation (2.7) based on a quasi-steady state approximation. Rather than using the density $\rho_{\text{cross}}(v_2, \tau; t_0)$ introduced in Section 2.4, we calculate the evolution of another density-like quantity that we call $\rho_{\text{delay}}(v_2, \tau; t_0)$. We calculate the evolution of ρ_{delay} in the same way as ρ_{cross} except that we freeze the input rates at $\nu_{\text{ind}}(t_0)$ and $\nu_{\text{syn}}(t_0)$. Moreover, in ρ_{delay}, we wish to look only at the first spike of neuron 2 after the spike of neuron 1. For this reason, we do not include the reset terms when evolving ρ_{delay}. To subtract off the expected distribution of v_2 under the assumption that the neurons were independent, we initialize ρ_{delay} by subtracting off the marginal distribution of ρ, multiplied by the average firing rate: $\rho_{\text{delay}}(v_2, 0; t_0) = J_{\text{reset},1}(v_2, t_0) - r_{\text{ave}}(t_0) \int \rho(v_1, v_2, t_0) dv_1$. Although we do not re-inject the reset term into the evolution equation for ρ_{delay}, we still define $c_{\text{delay}}(\tau, t_0)$ as the magnitude of this reset term (or, equivalently, the flux across threshold). Since we subtracted off the marginal density in defining $\rho_{\text{delay}}(v_2, 0; t_0)$, the reset quantity $c_{\text{delay}}(\tau, t_0)$ reflects how much more likely neuron 2 was to fire than it would have fired if the neurons were independent. In other words, $c_{\text{delay}}(\tau, t)$ represents the delayed correlation between the spikes of one neuron at time t and the subsequent spike of another neuron, subject to the quasi-steady state approximation that uses no future information beyond the time t. Given the quasi-steady state approximation, the correlation with negative delay is symmetric to the correlation with positive delay, and we define $C_{delay}(-\tau, t) = C_{delay}(\tau, t)$ for $\tau > 0$ The resulting correlation magnitude

analogous to $C_{\text{peak}}(t)$ is

$$\tilde{r}_{\text{syn}}(t) = r_{\text{syn}}(t) + \int_{-\tau_0}^{\tau_0} c_{\text{delay}}(\tau, t) d\tau \tag{2.12}$$

where τ_0 is the delay where $c_{\text{delay}}(\tau, t)$ first becomes negative: $\tau_0 = \min\{\tau > 0 \,|\, c_{\text{delay}}(\tau, t) \leq 0\}$. We denote this correlation by $\tilde{r}_{\text{syn}}(t)$ because we will use $\tilde{r}_{\text{syn}}(t)$ in place of the synchronous firing rate $r_{\text{syn}}(t)$ in the coupling equations Eqs. (2.10) and (2.11), effectively assuming that this correlation was due to synchronous spiking of the presynaptic neurons. Since we have not included reset in the calculation of c_{delay}, we know that the condition $\tilde{r}_{\text{syn}}(t) \leq r_{\text{ave}}(t)$ is always satisfied.

Our new connectivity equations (reintroducing superscripts to denote population index) are

$$\nu^k_{\text{syn}}(t) = \sum_j \beta_{jk} W^1_{jk} r^j_{\text{ave}}(t) + \sum_j W^1_{jk}(W^1_{jk} - 2\beta_{jk}) \tilde{r}^j_{\text{syn}}(t) \tag{2.13}$$

$$\nu^k_{\text{ind}}(t) = \nu^k_{\text{ext}}(t) + \sum_j W^1_{jk} r^j_{\text{ave}}(t) - \nu^k_{\text{syn}}(t). \tag{2.14}$$

We refer to this improved kinetic theory implementation that accounts for delayed correlations as KT1.

Note that there is no guarantee that $\nu^k_{\text{ind}}(t)$ defined by the above equation will be positive, despite the fact that we know $\tilde{r}^j_{\text{syn}}(t) \leq r^j_{\text{ave}}(t)$. It is possible for the synchronous input $\nu^k_{\text{syn}}(t)$ calculated from Eq. (2.13) to exceed the total input from presynaptic populations $\sum_j W^1_{jk} r^j_{\text{ave}}(t)$. Such a nonphysical result is due to neglecting the possibility of three or more simultaneous inputs. As discussed in detail in the Section 2.8, such combinations become highly likely as the correlation increases. To keep physically meaningful values of $\nu^k_{\text{syn}}(t)$, we simply truncate it to $\sum_j W^1_{jk} r^j_{\text{ave}}(t)$,

the total input from presynaptic populations, if the quantity calculated from Eq. (2.13) or Eq. (2.10) exceeds that sum.

The equations Eqs. (2.13) or (2.10) for $\nu_{\text{syn}}(t)$ contain two sources for the correlated input to a population. The first sum describes the emergence of correlations due to the shared input between a pair of neurons. For example, when either of the common input neurons of Figure 1.2 spike, it will send synchronous input into the pair of postsynaptic neurons. The second sum describes the propagation of correlations through the network. For example, when a light and dark presynaptic neuron of Figure 1.2 spike synchronously, they send synchronous input to the postsynaptic neuron pair even though no connections are shared. The subtraction of 2β is an approximate correction to control for over-counting when a common input neuron fires synchronously with another input neuron. In this way, the kinetic theory network is designed to capture the emergence and propagation of correlations in the network and describe the second order statistics of the network activity.

2.7 Results of initial approach

We demonstrate our kinetic theory approach by benchmarking its output against Monte Carlo simulations of networks of integrate-and-fire neurons. We first compare simulations of a single population of non-interacting neurons in order to demonstrate that the kinetic theory does accurately represent the response of neurons to correlated input. Second, we compare simulations of feed-forward networks in order to investigate how well our kinetic theory implementation captures the emergence and propagation of correlations through a network. We simulated the Monte-Carlo networks as described in (Nykamp and Tranchina, 2000).

2.7.1 Single Population

As an initial test of our kinetic theory approach, we simulated the response of a single population of neurons to correlated input. We numerically solved the kinetic theory equations (2.2) as described in Appendix A in response to specified independent $\nu_{\text{ind}}(t)$ and synchronous $\nu_{\text{syn}}(t)$ input rates. Since these single population runs did not include any network interactions, they simply reflected the ability of the kinetic theory to capture the correlated output of neurons in response to correlated input.

To test the accuracy of the kinetic theory solution, we simulated 500,000 realizations of a pair of integrate-and-fire neurons in response to the same input rates. We stimulated each neuron with an independent Poisson process with rate $\nu_{\text{ind}}(t)$. We also stimulated both neurons with a third Poisson process with rate $\nu_{\text{syn}}(t)$. Since the input to the neurons was Poisson and correlated only with zero delay, these Monte Carlo simulations exactly matched the assumptions of our kinetic theory model. In this way, the Monte Carlo simulations served simply to test if we correctly implemented our kinetic theory equations.

Results of one such test are shown in figure 2.4. For this example, we set the independent input rate $\nu_{\text{ind}}(t)$ to 150 spikes per second and the synchronous input rate $\nu_{\text{syn}}(t)$ to 100 spikes per second, and allowed the population to achieve steady state. Then, we kept the same input rates until $t = 0.05$ seconds, at which point we doubled the input rates to $\nu_{\text{ind}}(t) = 300$ and $\nu_{\text{syn}}(t) = 200$ spikes per second. In response, the average firing rate $r_{\text{ave}}(t)$, the synchronous firing rate $r_{\text{syn}}(t)$, and the cross-correlation peak area $C_{\text{peak}}(t)$ immediately jumped to higher values and then settled down to new steady state values. As expected, the kinetic theory and Monte Carlo estimates matched exactly in all three cases (figure 2.4A). Note that $C_{\text{peak}}(t)$ began to increase even before the time of the input change due to the fact that it

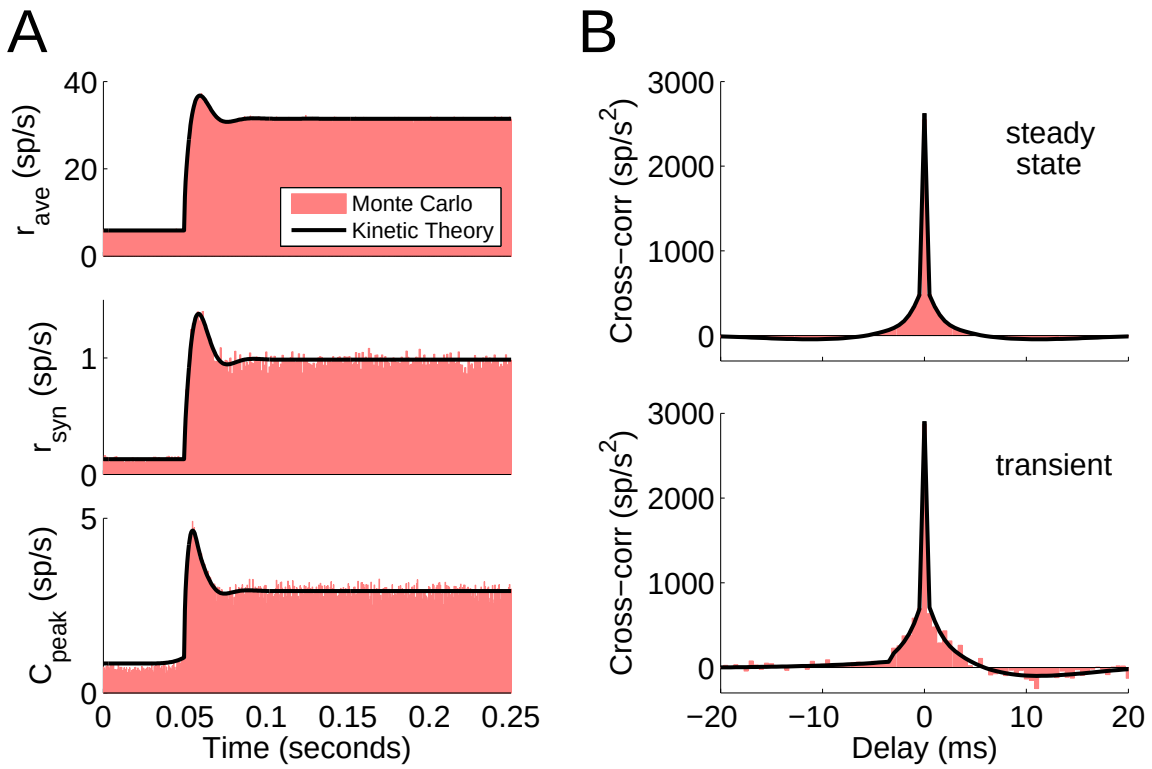


Figure 2.4: Results from a single population of uncoupled neurons. The kinetic theory results exactly match those from Monte Carlo simulations. **A**. Average firing $r_{\text{ave}}(t)$ (top), synchronous firing rate $r_{\text{syn}}(t)$ (middle), and cross-correlation peak area $C_{\text{peak}}(t)$ (bottom) are plotted in response to a jump in independent $\nu_{\text{ind}}(t)$ and synchronous $\nu_{\text{syn}}(t)$ input rate at time $t = 0.05$ seconds. The histograms from the Monte Carlo simulations coincide with the values obtained from solving the kinetic theory equations. **B**. Cross-correlation averaged over the steady state ($t > 0.15$ seconds) and cross-correlation from the transient peak in $C_{\text{peak}}(t)$ ($t = 0.054$ seconds) are plotted versus delay. The Monte-Carlo data was binned using a bin width of $\Delta t = 0.5$ ms. The kinetic theory cross-correlation contains a delta function at zero delay. The delta-function magnitude was divided by $\Delta t = 0.0005$ seconds and added to the value of the continuous correlation at zero delay. Since the kinetic theory correlation is sampled every half millisecond, this procedure effectively smooths the delta-function over a half millisecond window so that the results match the bin width used in Monte Carlo.

includes correlation of spikes at time t with spikes at future times.

The kinetic theory also accurately captures the form of the delayed correlation resulting from the correlated input. As shown in figure 2.4B, the structure of the cross-correlation averaged over all neuron pairs is precisely matched by the kinetic theory. At the steady state (figure 2.4B, top), the cross-correlation is symmetric with respect to delay because it is an average over all pairs of neurons. The asymmetry in the transient cross-correlation (figure 2.4B, bottom) is due to the asymmetric population activity around the sampled time point. We obtained equally good matches between kinetic theory and Monte Carlo simulations for every other input rate ($\nu_{\text{ind}}(t)$ and $\nu_{\text{syn}}(t)$) combination we tested. These results are not surprising because the input in the Monte Carlo simulations was chosen to exactly match the assumptions of our kinetic theory implementation. The results simply serve as verification that we are solving our kinetic theory equations correctly.

2.7.2 Feed-forward network

To test the ability of our kinetic theory implementation to capture the emergence and propagation of correlations through a network, we examined its performance with a ten layer feed-forward network. It is well-known that synchronous activity develops in deeper layers of a feed-forward network (Diesmann et al, 1999; Câteau and Fukai, 2001; Reyes, 2003; Hasegawa, 2003; Wang et al, 2006; Masuda and Aihara, 2002; Litvak et al, 2003; van Rossum et al, 2002; Doiron et al, 2006). By comparing the kinetic theory results to Monte Carlo simulations of the feed-forward network, we could assess how well the kinetic theory could model the build-up of correlations underlying such synchrony.

To illustrate the build-up of correlations, we simulated the response of feed-

forward networks to a step input, where at time t_0 we instantaneously increased the independent Poisson input to layer 1 from $\nu^1_{\text{ind,ext}}(t) = 200$ spikes per second to $\nu^1_{\text{ind,ext}}(t) = 300$. Each layer $k > 1$ received independent Poisson input at constant rate $\nu^k_{\text{ind,ext}}(t) = 200$ in addition to the input from the previous layer. Unlike the single population simulations, we did not add any external synchronous input as we wished to explore the synchrony that emerged from the network.

We created networks with N neurons per layer and randomly connected neurons from each layer onto the subsequent layer so that the expected number of connections onto each neuron was $W = 10$. (To accomplish this, we randomly selected each of the possible N^2 connections between a pair of layers with probability W/N.)

An example of the emergent synchronous activity in feed-forward networks is illustrated in figure 2.5A. For the network with $N = 100$ neurons per layer (figure 2.5A, left column), peaks in the histogram begin to emerge in layer 6 and become prominent in layer 10. These peaks correspond to many neurons firing synchronously within a $\Delta t = 10$ ms time window, as the histogram is based on a bin width of Δt.

In the network with $N = 1000$ neurons per layer (figure 2.5A, right column), this synchrony does not appear to emerge, at least in 10 layers, as the histograms remain relatively flat even in layer 10. Such observations make it seem that such synchronization is a finite-size effect and that the number of neurons N plays a critical role in determining the emergence of synchrony (Doiron et al, 2006). We will reexamine this hypothesis below.

The build-up of correlations is illustrated in figure 2.5B. For the network with $N = 100$ neurons per layer (figure 2.5B, left column), the cross-correlation for layer 6 contains a large peak centered around zero delay, and this peak is much larger in layer 10. On the other hand, when $N = 1000$, the cross-correlation has only a small peak even in layer 10 (figure 2.5B, right column). Our goal is to capture this build-up

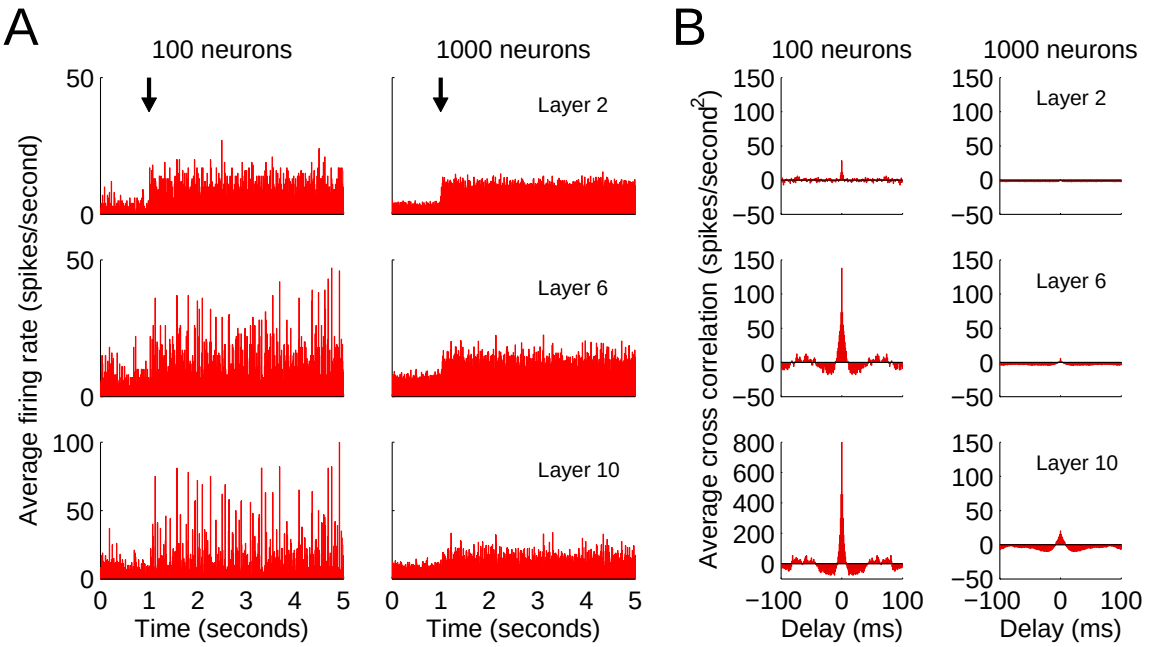


Figure 2.5: Illustration of the build-up of correlations within a feed-forward network with 10 layers. **A** A histogram of the firing times of all neurons in a layer binned at $\Delta t = 10$ ms, normalized to firing rate in spikes per second. The input rate to layer 1 was increased at time $t_0 = 1$ second (indicated by the arrow). Layers 2, 6, and 10 are shown for a network with $N = 100$ neurons per layer (left column) and $N = 1000$ neurons per layer (right column). For the $N = 100$ network, sharp peaks in the histograms of layers 6 and 10 indicate simultaneous firing of many neurons within a single time bin. For the $N = 1000$ network, although the neurons are firing at a similar rate, no strong peaks indicating synchrony are visible. **B**. The cross-correlation calculated from the steady state ($t > 2$ seconds) and averaged among all pairs of neurons. Each panel corresponds to the simulation of the respective panel in A. For the network with $N = 100$ neurons per layer, a strong peak in the cross-correlation is evident in layers 6 and 10 (note different scale for layer 10). For the $N = 1000$ network, the correlation is small even in layer 10.

of correlation with our kinetic theory model.

The difficulty of capturing the synchrony with kinetic theory models is illustrated in figure 2.6. The kinetic theory can be thought of as representing the distribution of neuron responses over many realizations of the network response. Figure 2.6 shows histograms of the spikes of layer 10 of the network with $N = 100$ neurons per layer. For each of the four top panels, the same input used for figure 2.5A was presented, except that the step in input rate to layer 1 (i.e. the stimulus) occurred at $t_0 = 0.05$ seconds. In this case, we bin the spikes at a smaller resolution of $\Delta t = 2$ ms and show only 0.4 seconds of data. In response to each of the four stimulus presentations, the layer 10 neurons show a high degree of synchrony as evidenced by the sharp peaks in the histograms. However, as the stimulus is constant, there is no temporal structure that could align the times at which those peaks occur. Hence, in each of the four presentations, the synchronous peaks occur at different times.

The average over 4000 stimulus presentations is shown in the bottom panel of figure 2.6. In this peristimulus time histogram (PSTH) we see a transient of increased firing rate soon after the stimulus, but then the firing rate settles down to a constant value. There is no indication that the neurons actually continued to fire synchronously. Due to the distribution of different times of the synchronous peaks, the average removes all of the temporal structure in the firing rate.

Since kinetic theory represents this average response, the average firing rate $r_{\text{ave}}(t)$ will not include any evidence of the the synchronous firing, but instead should approach a constant value during the steady state period. We cannot capture the synchronous peaks of figure 2.5A. Instead, our goal is a kinetic theory representation of the feed-forward network that captures the build-up of correlations of figure 2.5B.

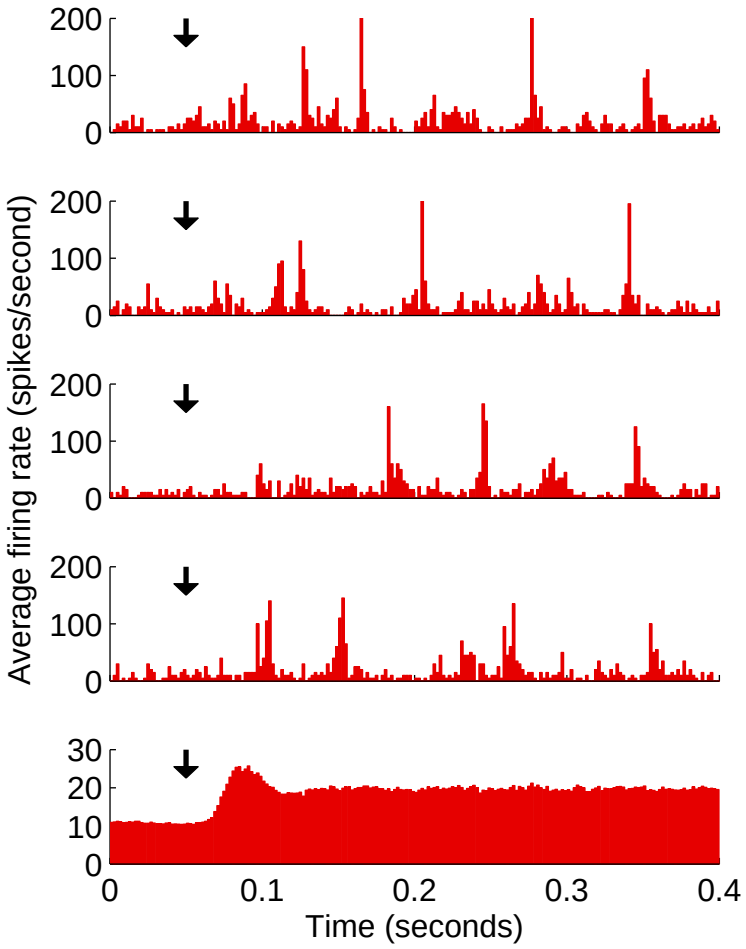


Figure 2.6: The timing of synchronous bursts of activity will vary with different presentations of the same input. The feed-forward network with $N = 100$ neurons per layer was repeatedly stimulated with the input of figure 2.5, except that that step of the input occurred at $t_0 = 0.05$ seconds (arrows). The histogram of activity of the layer 10 neuron to four different presentations is shown in the top four panels. Plot is the same as the bottom left of figure 2.5 except for a smaller bin width of $\Delta t = 2$ ms and smaller range of times. The bottom panel is a peristimulus time histogram (PSTH) showing the average of 4000 such presentations of the stimulus. Since the synchronous peaks in response to different presentations occurred at different times, the PSTH averages out the peaks and is flat after an initial transient.

2.7.3 Demonstration of initial kinetic theory results

To compare the kinetic theory with the Monte Carlo simulations of the feed-forward networks, the first step is to calculate the connectivity statistics for the kinetic theory discussed in section 2.5. As described above, we generate connections between two layers so that each connection has probability W/N, where N is the number of neurons per layer and $W = 10$. Since for a given postsynaptic neuron, N possible connections are chosen with probability W/N, the number W is the expected number of connections.

The expected number of shared connections onto any pair of postsynaptic neurons can be calculated from the assumption that all connections are generated independently. Any of the N possible presynaptic neurons has probability W/N of being connected to the first neuron of the pair and probability W/N of being connected to the second. Given that the connections are generated independently, the probability of it being connection to both neurons in the pair is simply W^2/N^2. Multiplying by the N possible presynaptic neurons, we calculate that the expected number of shared connections onto any pair of neurons is W^2/N. Hence, the fraction of shared connections parameter β is $W^2/N/W = W/N$.

We simulated feed-forward networks with $N = 50$, 100, 200, and 1000 neurons per population. We randomly generated a single network and presented the above step input $400,000/N$ times, as in figure 2.6. Before beginning each presentation, we allowed the network to equilibrate to steady state in response to the lower input rate ($\nu^k_{\text{ind}}(t) = 200$ spikes per second for all layers). Then, we began to record spikes and jumped the input to layer to up to $\nu^1_{\text{ind}}(t) = 300$ spikes per second at time $t_0 = 0.05$ seconds.

For each N, we calculated $\beta = W/N$ and created a kinetic theory representation

of the network. The kinetic theory network consisted of 10 population, where each population corresponded to a layer of the network. We initially numerically solved the kinetic theory equations in response to the lower input rate ($\nu^k_{\text{ind}}(t) = 200$ spikes per second for all layers) to calculate the steady state distribution for that input. Using that distribution as the initial conditions, we numerically solved the kinetic theory equations in response to the input described above, increasing the input to layer one at time $t_0 = 0.05$ seconds.

For each network, we computed the kinetic theory response using two methods of approximating delayed correlation in the output of each population. In one method, we ignored any correlation with non-zero delay in the output of each presynaptic population and computed the input to the corresponding postsynaptic population as though non-synchronous spikes were independent (using equations (2.10) and (2.11) to connect populations). In the second method, we approximated delayed correlation as though it were correlation with zero delay (using equations (2.13) and (2.14) to connect populations). In either case, the input to each population $k > 1$ was a combination of independent and synchronous Poisson input; we did not include delayed input correlation in our kinetic theory implementation.

The results for $\beta = 0.05$ (i.e., $N = 200$ in the Monte Carlo simulations) are shown in figure 2.7. The average firing rate $r_{\text{ave}}(t)$ (figure 2.7A) of the kinetic theory closely matched the Monte Carlo, independent of which method we used for approximating delayed correlation. At least for this network, assuming Poisson input to each layer did not greatly alter the first order output statistic $r_{\text{ave}}(t)$.

For the cross-correlation peak area $C_{\text{peak}}(t)$ (figure 2.7B), the results had a dramatic dependence on the delayed correlation method. If we completely ignored delayed correlation, the kinetic theory failed to show the build-up in the correlation in the deeper layers of the network. On the other hand, when we approximated de-

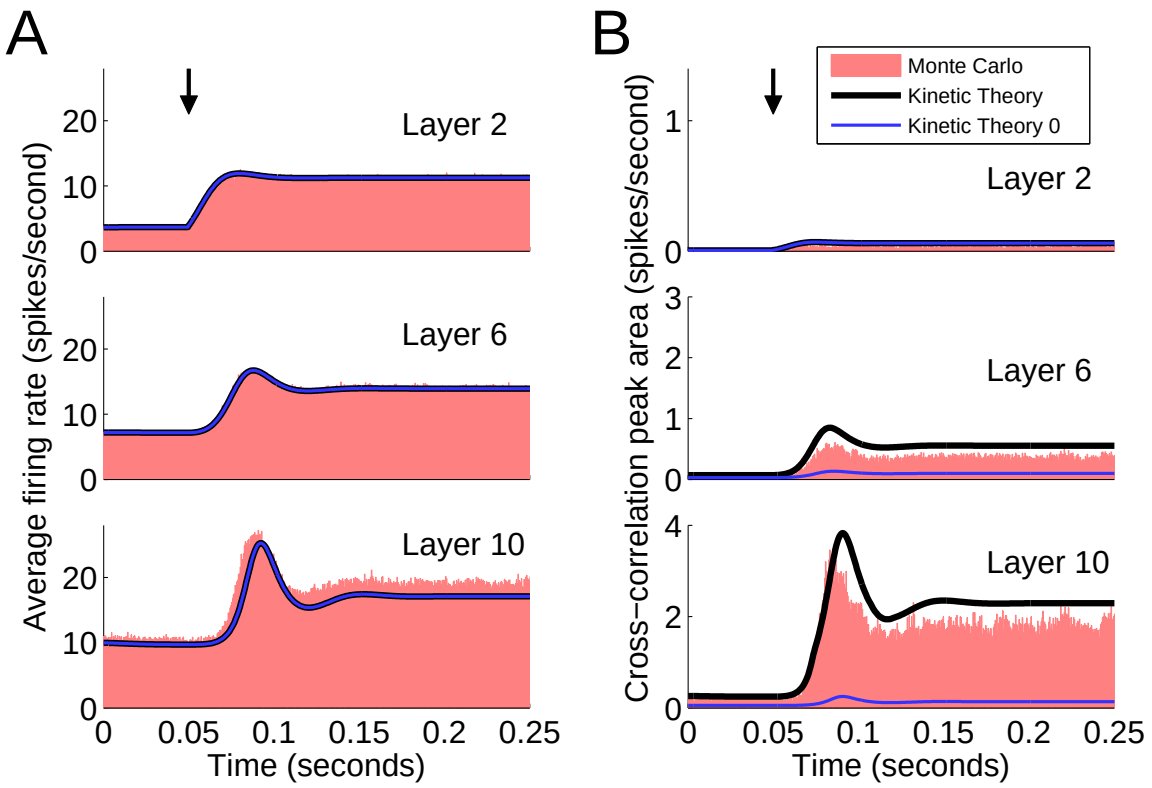


Figure 2.7: Comparison of Monte Carlo and kinetic theory results for the network with $\beta = 0.05$ ($N = 200$ in Monte Carlo). The input rate to layer 1 is stepped up at time $t_0 = 0.05$ seconds (arrows). **A** The average firing rate $r_{\text{ave}}(t)$ is plotted for layers 2, 6, and 10. For Monte Carlo simulations, we plot a normalized peristimulus time histogram. The kinetic theory results based on ignoring delayed correlation in the neural output (Kinetic Theory 0) are plotted by the thin line. The thick line indicates the kinetic theory results where delayed correlation is approximated as though it were instantaneously correlation (Kinetic Theory). Both kinetic theory approximations result in nearly identical estimates of $r_{\text{ave}}(t)$, which underestimate the Monte Carlo results only slightly in the deeper layers. **B**. The cross-correlation peak area $C_{\text{peak}}(t)$ is plotted for layers 2, 6 and 10. The same plotting convention as panel A is followed. Ignoring delayed correlation results in kinetic theory estimates that fail to show the build-up of correlation in deeper layers. Approximating delayed correlation as instantaneous correlation leads to good estimate of the correlation build-up, though correlation magnitude is overestimated.

layed correlation as instantaneous correlation, the kinetic theory was able to show the correlation increase. As instantaneous correlation has a stronger effect on neural response than delayed correlation, the approximation resulted in an overestimation of the actual magnitude of the correlation peak. But, the method captured the essence of how the correlations built-up in the network.

Clearly, completely ignoring delayed correlation fails miserably to capture the build-up of correlations. Approximating delayed correlation as instantaneous correlation appears to be a much better method to handle the delayed correlation. From now on, we will focus our attention on this second method of approximating delayed correlation, and will refer to this method when using the unmodified term "kinetic theory." The method where we ignore delayed correlation will be termed "kinetic theory 0."

If we doubled the fraction of shared input to $\beta = 0.1$ (halved the number of neurons to the $N = 100$ of figures 2.5 and 2.6), our kinetic theory implementation matched the Monte Carlo results less accurately (figure 2.8). The kinetic theory captures the build-up of correlations up through layer 6, still overestimating the correlation magnitude. However, by layer 10, correlation peak area calculate by the kinetic theory is only about 70% that of the Monte Carlo at steady state. Even the first order statistic $r_{\text{ave}}(t)$ is slightly underestimated by the kinetic theory. The approximations underlying our kinetic theory implementation do indeed begin to break down at this high level of correlation, as predicted by the assumptions of our derivation in section 2.5. We discuss some practical remedies in the Discussion.

We examine the structure of the steady-state cross-correlation for these two networks in figure 2.9. Since we approximated delayed correlation as instantaneous, we do not expect the kinetic theory to accurately reproduce all the structure of the cross-correlation as a function of delay. Indeed, for the $\beta = 0.05$ network, the kinetic

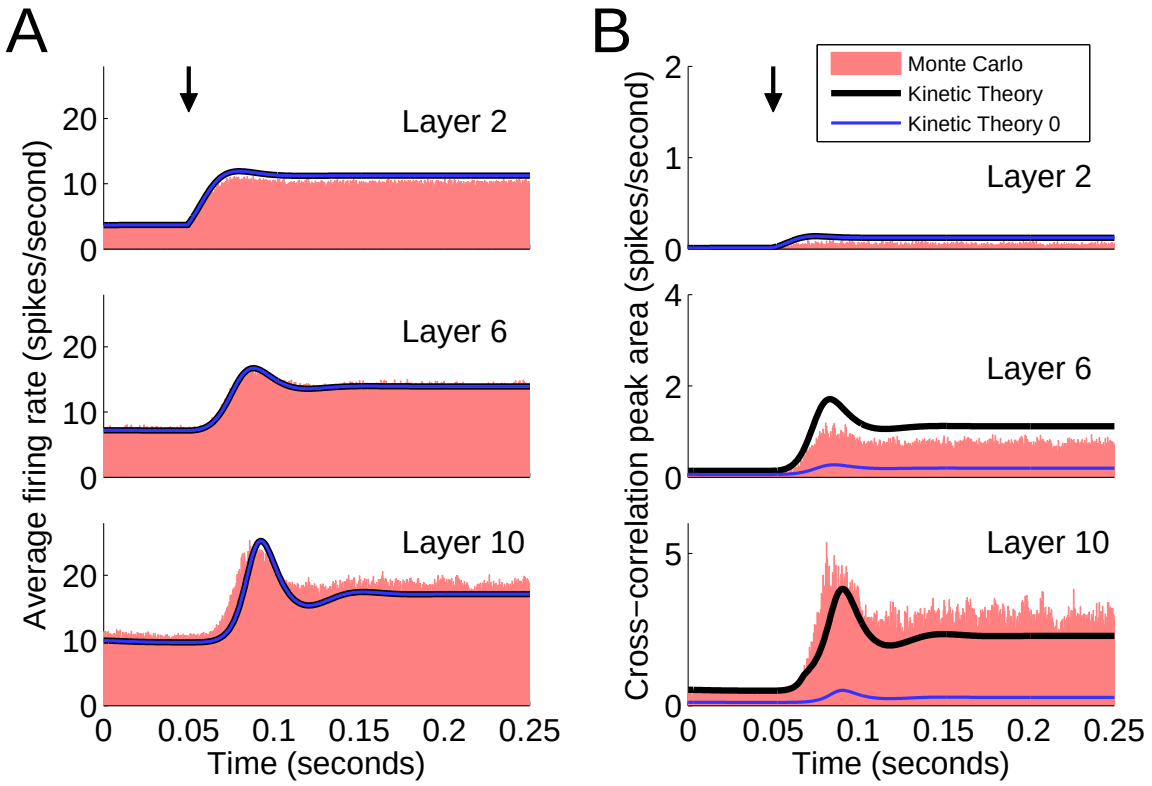


Figure 2.8: Comparison of Monte Carlo and kinetic theory results for the network with $\beta = 0.1$ ($N = 100$ in Monte Carlo). Panels as in figure 2.7. For this large fraction of shared connections β, the kinetic theory captures the initial build up of correlations as shown in layer 6 (overestimating the actual magnitude as expected), but underestimates the increased correlation in deeper layers. The average firing rate is also slightly underestimated in deep layers. The kinetic theory that ignores delayed correlations (kinetic theory 0) grossly underestimates the correlations.

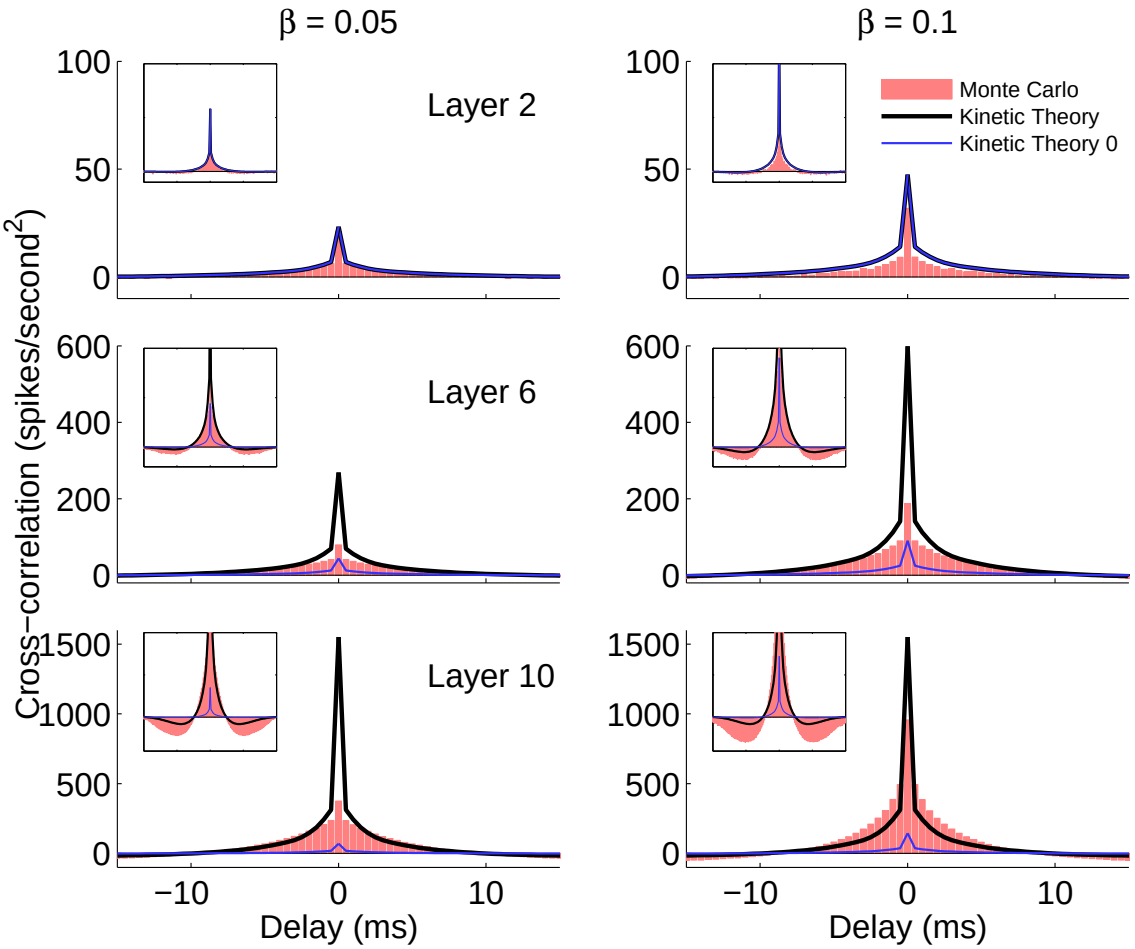


Figure 2.9: Comparison of the structure of steady state cross-correlation for layers 2, 6, and 10 of the $\beta = 0.05$ network of figure 2.7 (left column) and the $\beta = 0.1$ network of figure 2.8 (right column). The Monte Carlo and kinetic theory cross-correlations were computed as in figure 2.4B; in particular, the delta-function component of the kinetic theory correlation was smoothed over a $\Delta t = 0.5$ ms bin centered at the origin. In each panel, the central peak of the cross-correlation, which is used to calculate the peak area $C_{\text{peak}}(t)$, is shown in main plot. For the deeper layers of the $\beta = 0.05$ network and the intermediate layers of the $\beta = 0.1$ network, the kinetic theory overestimates the correlation at zero delay. By layer 10 of the $\beta = 0.10$ network, the kinetic theory no longer overestimates the zero delay correlation and significantly underestimates it at non-zero delay. The insets show the correlation over a larger range of delays to reveal the negative correlation at longer delays, which are not captured by the kinetic theory. The kinetic theory that ignores non-zero delays (kinetic theory 0) uniformly underestimates the correlation by a large margin.

theory grossly overestimates the correlation at zero delay in the deeper layers (left column of figure 2.9). This greater correlation at zero delay explains the overestimate of the peak area $C_{\text{peak}}(t)$ as compared to Monte Carlo. As shown by the insets, the kinetic theory also misses most of the negative correlation at larger delays, although this correlation is not included in the peak area $C_{\text{peak}}(t)$ plotted in figure 2.7B.

For the $\beta = 0.10$ network, the kinetic theory overestimates the correlation with zero delay only in the intermediate layers. By layer 10, the zero-delay correlation from the Monte Carlo simulations is as large as the kinetic theory correlation despite the fact that the kinetic theory assumes the all the input correlation is at zero delay. The kinetic theory significantly underestimates the correlation at non-zero delay so that the total area in the peak is well under that of the Monte Carlo, as shown in figure 2.8B.

The cross-correlation of layer 10 for the $\beta = 0.01$ (i.e., $N = 1000$ for Monte Carlo) and $\beta = 0.2$ (i.e., $N = 50$ for Monte Carlo) networks are shown in figure 2.10. For $\beta = 0.01$, only a small amount of correlation has built up over 10 layers (c.f., 1000 neuron plots of figure 2.5). As with the $\beta = 0.05$ case, the kinetic theory overestimates the cross-correlation peak area $C_{\text{peak}}(t)$ for all times after stimulus onset (top of figure 2.10A) due to an overestimate of small-delay correlation (top of figure 2.10B). For the $\beta = 0.2$ network (bottom of figure 2.10), the underestimate of the correlation already seen with $\beta = 0.1$ has become much more dramatic. The kinetic theory estimate differs little from that of $\beta = 0.1$, although the correlation in the Monte Carlo network doubled. Clearly, the implementation of the initial kinetic theory approach cannot capture the high degree of correlation seen with the higher β networks. Even the correlation at zero delay (bottom of figure 2.10B) is substantially underestimated by the kinetic theory.

The steady-state results from all four networks are shown in figure 2.11. The

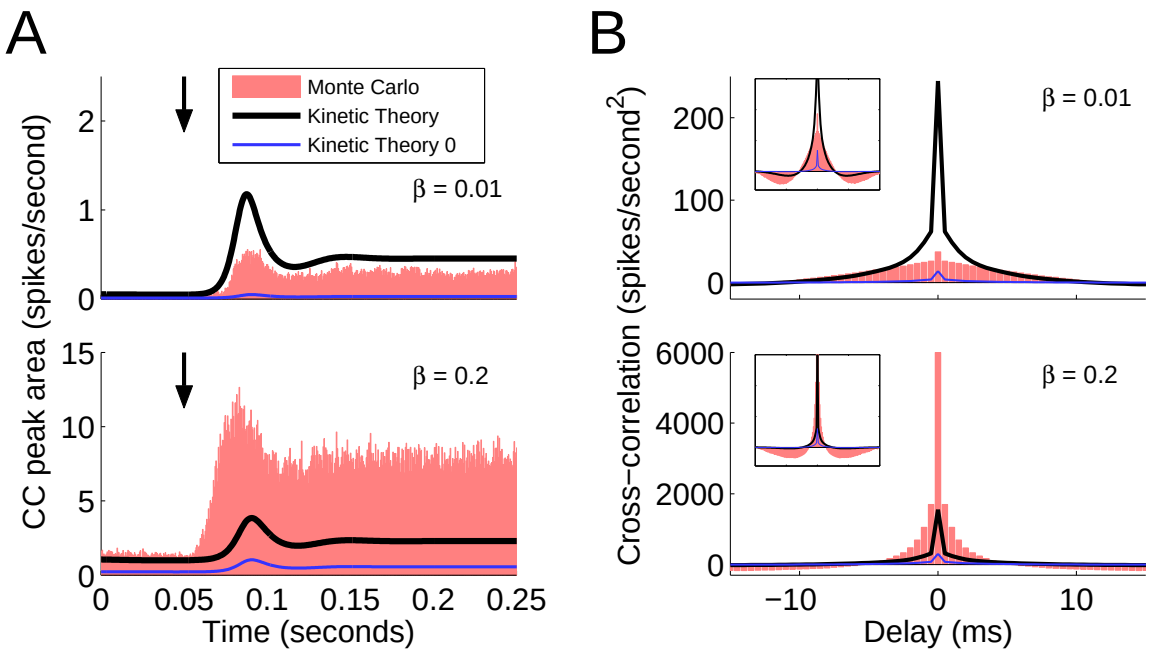


Figure 2.10: Comparison of the cross-correlation in layer 10 for the networks with $\beta = 0.01$ ($N = 1000$ for Monte Carlo) and $\beta = 0.2$ ($N = 50$ for Monte Carlo). **A**. The cross-correlation peak area $C_{\text{peak}}(t)$ is plotted versus time. Panels as in figure 2.7B. The small correlation in the $\beta = 0.01$ network is slightly overestimated by the kinetic theory. The large correlation in the $\beta = 0.2$ network is substantially underestimated by the kinetic theory. **B**. The cross-correlation is plotted versus delay. Panels as in figure 2.9. The kinetic theory overestimates the small-delay correlation for the $\beta = 0.01$ network and uniformly underestimates the correlation for the $\beta = 0.2$ network. The kinetic theory that ignores non-zero delays (kinetic theory 0) misses the correlation in all cases.

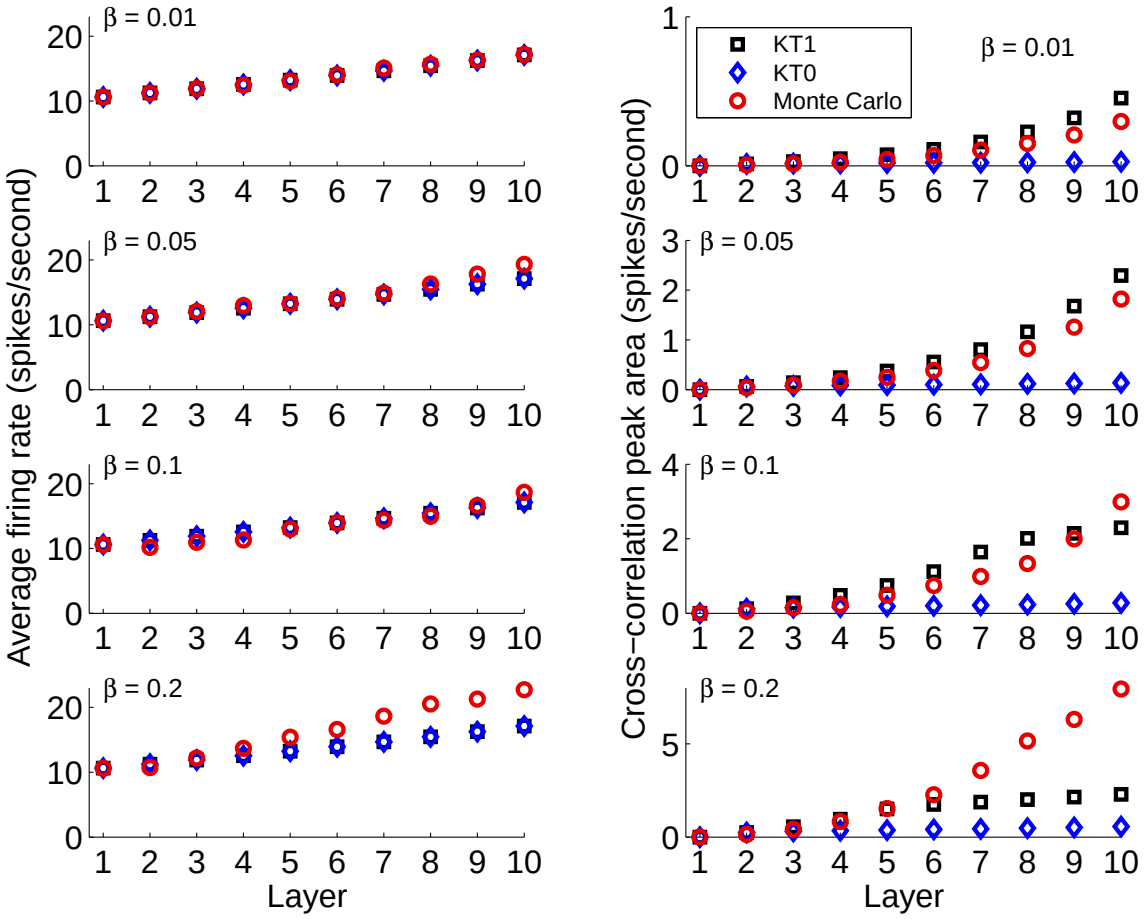


Figure 2.11: Comparison of the steady state values of the average firing rate r_{ave} and cross-correlation peak area C_{peak} for all layers of the four tested networks with $\beta = 0.01$, 0.05, 0.1 and 0.2. The kinetic theory approximates the steady-state firing rate fairly well for all networks. For $\beta \leq 0.05$, the kinetic theory captures the build-up of correlations through all 10 layers of the network, though it overestimates the magnitude. For the $\beta = 0.1$ and 0.2, the kinetic theory only captures the initial build-up of correlation in the first layers of the network but fails to estimate the large degree of correlation that occurs in the deeper layers.

average firing rate (left column) is accurately estimated by the kinetic theory, except for small underestimates in the deeper layers for the larger βs. The kinetic theory that ignores delayed correlation (kinetic theory 0) fails to capture even the small correlations of the small β networks. On the other hand, the kinetic theory that approximates delayed correlation as instantaneous correlation (kinetic theory) does a reasonable job estimating (albeit overestimating) the correlation as long as the correlation isn't too large. Since the kinetic theory estimate of cross-correlation peak area C_{peak} seems to saturate at around 2.4, the correlation estimated by the kinetic theory falls well behind the actual correlation observed in the Monte Carlo networks in the $\beta = 0.1$ and $\beta = 0.2$ networks. By layer 10 for $\beta = 0.1$ and layer 7 for $\beta = 0.2$, the networks appear to be too correlated for the approximations underlying our kinetic theory implementation. We conclude that the kinetic theory captures the build-up of steady-state cross-correlation in feed-forward networks up through moderate levels of correlation but fails in cases with strong correlation.

As a further test of the kinetic theory, we ran additional groups of simulations. We used three different values of expected number of connections: $W = 8$, 10, and 12. For each of these values, we ran two groups of simulations: one where we set the external input rates $\nu^k_{\text{ind,ext}}$ so that each population fired around 10–15 spikes per second (the "low rate" group) and another where we set the $\nu^k_{\text{ind,ext}}$ so that each population fired around 50–60 spikes per second (the "high rate" group). The simulations shown above were therefore from the $W = 10$, low rate group. For each group, we simulated networks with various β, comparing the kinetic theory and Monte Carlo results at steady state.

As shown in figure 2.12, the kinetic theory captures the steady state correlation in all networks as long as the correlation isn't too high. For each layer 2 and above (for layer 1, the correlation is zero), we plotted a point corresponding to the steady

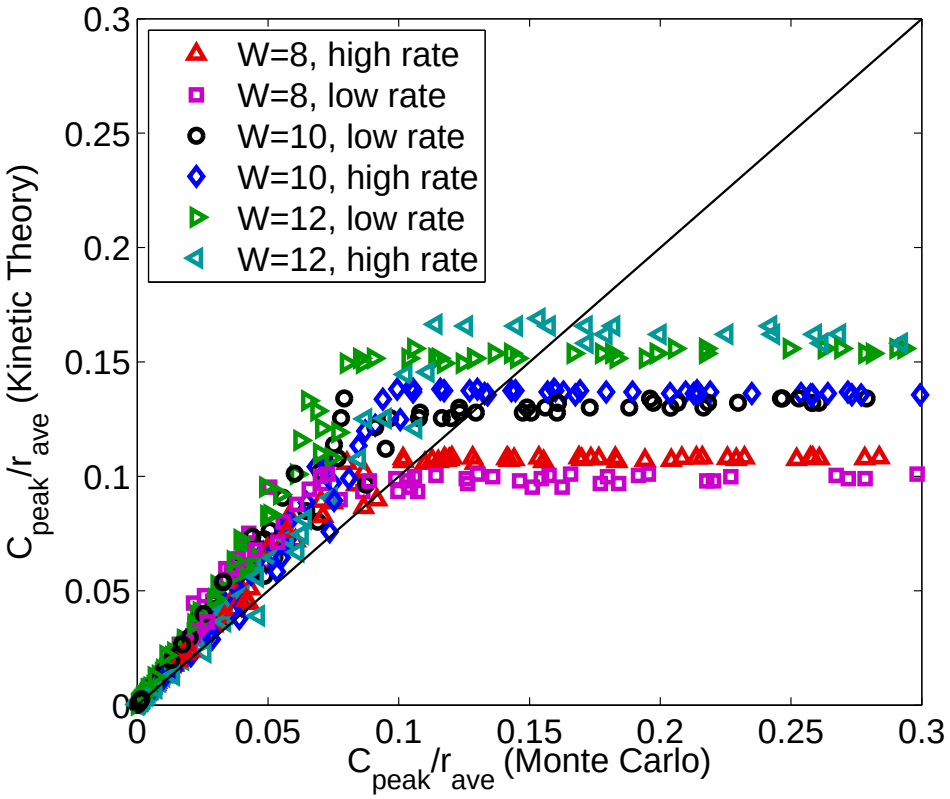


Figure 2.12: Comparison of the steady state correlation calculated from kinetic theory and Monte Carlo for many networks and layers 2 through 10. The cross-correlation peak area C_{peak} is plotted as a fraction of the average firing rate r_{ave}. The kinetic theory network tends to overestimate the cross-correlation until the kinetic theory saturates at its maximum possible correlation. (Points where the Monte Carlo $C_{\text{peak}}/r_{\text{ave}} > 0.3$ were omitted; in these cases, the kinetic theory values simply stayed at the saturation point.) For all "low rate" simulations, we set the input to layer to to be $\nu^1_{\text{ind,ext}} = 300$ spikes per second, and for all "high rate" simulations, we set $\nu^1_{\text{ind,ext}} = 600$ spikes per second. For the remaining layers $k > 1$, the external input $\nu^k_{\text{ind,ext}} = \nu_0$ varied by networks as follows (all values in spikes per second). $W = 8$: low rate $\nu_0 = 220$, high rate $\nu_0 = 300$. $W = 10$: low rate $\nu_0 = 200$, high rate $\nu_0 = 200$. $W = 12$: low rate $\nu_0 = 175$, high rate $\nu_0 = 110$.

state correlation C_{peak} as a fraction of the average firing rate r_{ave} calculated by kinetic theory and Monte Carlo simulations. For $C_{\text{peak}}/r_{\text{ave}}$ less than about 0.1, the points lie above the diagonal, confirming that the kinetic theory captures the build-up of correlations in all these networks, slightly overestimating the magnitude. Depending on the average connectivity W and firing rate, the kinetic theory reaches a point where the correlation it can represent saturates at some maximal value. For network layers where the correlation estimated by Monte Carlo simulations exceeded this maximum value, the kinetic theory limitation lead to underestimation of the correlation.

2.7.4 The sufficiency of second order connectivity statistics

We are developing our kinetic theory approach based on the hypothesis that second order statistics of neuronal activity are sufficient to describe much of the behavior of neuronal networks (Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu et al, 2008). We do not test this hypothesis directly here, as we have been examining output statistics of only first and second order (average firing rate and cross-correlation). Even so, we can indirectly test one implicit assumption of the approach: that second order connectivity statistics (expected number of inputs W and fraction of shared inputs β) are sufficient to determine the second order statistics of the network output. We can test this hypothesis by simulating networks with different types of connectivity patterns that have the same second order connectivity statistics but differ in third and higher order statistics.

In the Appendix B, we calculate the statistics W and β for networks with prescribed outgoing degree distributions and incoming degree distributions equivalent to the random networks described above. Then, we generated multiple 10 layer feed-forward networks with $W = 10$ and various values of β between 0 and 0.2. We

simulated the networks to the same input conditions as in the $W = 10$, low rate, networks described above and calculated the steady-state values of the average firing rate r_{ave} and cross-correlation peak area C_{peak} for layer 10.

For each set of network parameters, we performed two types of Monte Carlo simulations. First, we randomly generated a single network and simulated the response of this fixed network to $400 \max(1, 1000/N)$ presentations of the input, where N is the number of neurons per layer. This first type of simulation corresponds to the Monte Carlo simulations employed above. For the second type of Monte Carlo simulation, we simulated the network response to the same number of presentations of the input, but this time, for each input presentation, we regenerated a new realization of the network. By rewiring the network for each realization, we averaged over the fluctuations due to particular network configurations so that we could better estimate how the output statistics depend on the type of network. Note that the kinetic theory networks are designed to capture the behavior of the average network with particular network statistics as represented by the latter type of Monte Carlo simulations.

We simulated Monte Carlo networks with seven different degree distributions. The first network class was composed of random networks, as described above (which have a binomial degree distribution). The second consisted of networks with outgoing degree distributions that followed a power law. Then, we simulated networks with a power-law outgoing degree distributions, except that we truncated the maximum outgoing degree at different values: 100, 500, 2000, and 5000. (Clearly, these differed from the networks with the unmodified power-law distribution only if the number of neurons per layer N was larger than the cutoff.) Lastly, we simulated networks with outgoing degree-distributions that were Gaussian.

As described in Appendix B, each class of networks has a single free parameter once the layer size N was specified. This parameter was set to match the condition

that the expected number of incoming connections was $W = 10$. The value of that parameter determined the fraction of shared input β. In this way, for each class of network, β was a function of N. Since this function depended on the class of network, we could obtain different values of β for each N. By examining the output statistics r_{ave} and C_{peak} evaluated for the steady state of layer 10, we could investigate if these output statistics were primarily determined by β (as we hypothesize in the formulation of the kinetic theory equations) or by the layer size N (as figure 2.5 seemed to indicate).

The results of these simulations are shown in figure 2.13. If we look just at networks where we regenerated the network circuitry for each realization (filled symbols), we see that both the average firing rate r_{ave} (figure 2.13A) and the cross-correlation peak area C_{peak} (figure 2.13B) are very well approximated as functions of β. (The open symbols corresponding to simulations of fixed networks have a large spread around the filled symbols.) Moreover, for $\beta \leq 0.05$, the functions of β given by the filled symbols are close to that predicted by the kinetic theory (though the kinetic theory curve overestimates the magnitude the cross-correlation). Importantly, even for large β the filled symbols nearly lie along a curve, indicating that the second order connectivity statistics are sufficient to determine the second order output statistics. Although the kinetic theory curve strongly deviates from the Monte Carlo curve for large β, the fact that the Monte Carlo points lie along a single curve indicate that it may be possible for a kinetic theory based on W and β to capture the full range of correlations (see Discussion).

In contrast, the plot of C_{peak} versus N (figure 2.13C) indicate that layer size N is not a good predictor of the correlation if the class of network is not specified. Even the filled symbols have a large spread for all layer sizes. If one restricts attention to a particular network class, then the filled symbols do fall along a single curve, which

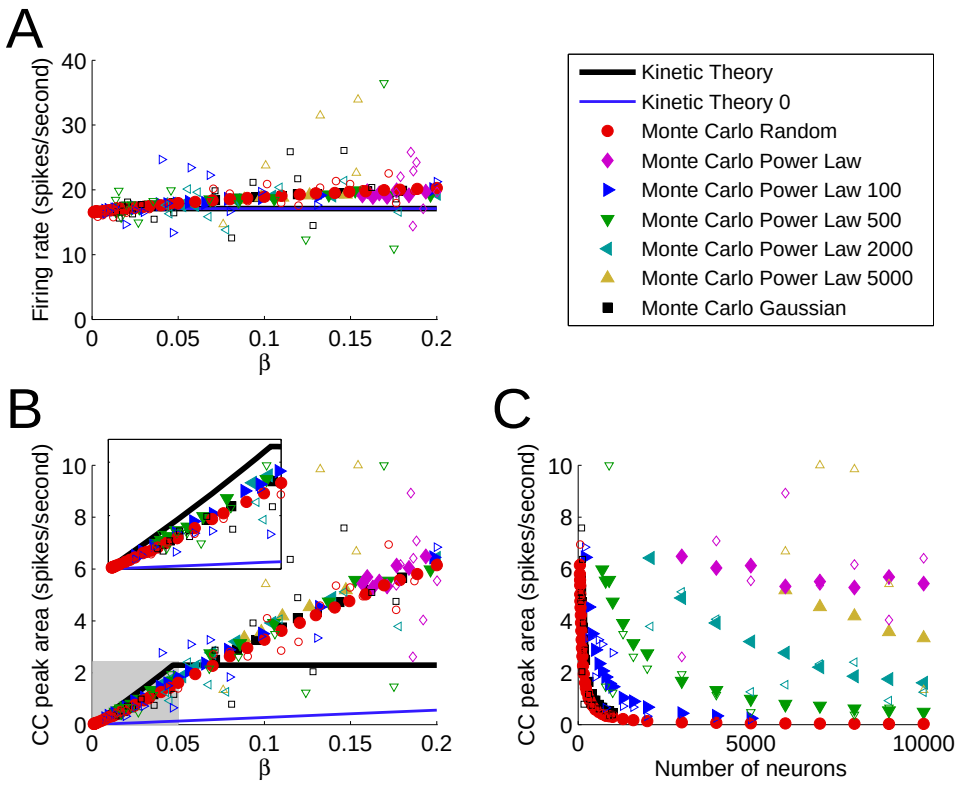


Figure 2.13: Steady state values of average firing rate r_{ave} and cross-correlation peak area C_{peak} in layer 10 for different classes of networks. Filled symbols correspond to Monte Carlo simulations of averaged networks, where the network connectivity was regenerated at each realization. Open symbols correspond to a simulations of a single fixed network of the same network class as the corresponding filled symbol. The number for the power law networks indicates the maximum outgoing degree allowed. **A**. The firing rate for averaged networks increases modestly with β, and the fixed network values are more scattered. The kinetic theory results are similar to the averaged networks, though they show little increase with β. **B**. The cross-correlation peak area C_{peak} for averaged networks appears to be nearly a function of β, as the filled symbols lie along a single curve. Open symbols are scattered around the filled symbols, indicating the variability observed among the fixed networks. Inset is detail of results for $\beta \leq 0.05$ (shaded region). In this range, the kinetic theory is a good approximation of the averaged network results, though the magnitude is overestimated. For larger β, the kinetic theory does not show the increase in C_{peak} with β. The kinetic theory that ignores non-zero delay correlation (kinetic theory 0) fails to capture C_{peak} even for small β. For display purposes, C_{peak} was truncated to 10 for two points. **C**. The cross-correlation peak area C_{peak} is not well approximated as a function of layer size N. For any value of N, the correlation depends strongly on the type of network. Only points with $\beta \leq 0.2$ are shown to correspond to panel B.

is due to the fact that β is a function of N when the network class is fixed. These results clearly demonstrate that the fraction of shared input β not layer size N is critical for determining the build-up of correlations in feed-forward networks.

As a final illustration of the fact that the first and second order connectivity statistics, not network size, determine the network behavior, we created four different networks with vastly different sizes and the same connectivity statistics. In each case, we created networks to match the connectivity statistics $W = 10$ and $\beta = 0.05$ of the network from figure 2.7. If we used a power law outgoing degree distribution capped at 5000, we needed $N = 17,500$ neurons per layer to reduce β down to 0.05. Capping the outgoing degree at 2000 and 500 reduced the layer size to $N = 8350$ and $N = 2750$, respectively. In comparison, the random network of figure 2.7 had only $N = 200$ neurons per layer. For each network, we stepped up the input rate at $t = 0.05$, using the input rates for figure 2.7. We presented this input 2000 times and regenerated the circuitry for each realization.

The neuronal output of layer 10 of each network is shown in figure 2.14. The activity of each network is similar, indicating the first and second order connectivity statistics did determine most of the behavior. Some minor differences are that the random network had slightly low steady state correlation and some of the power law networks appeared to have a larger transient correlation in response to the input step. Hence, although the second order connectivity statistics did not determine every detail of the neuronal activity, the connectivity statistics W and β did capture the essential features of the connectivity needed to describe most of the second order statistics of the network activity.

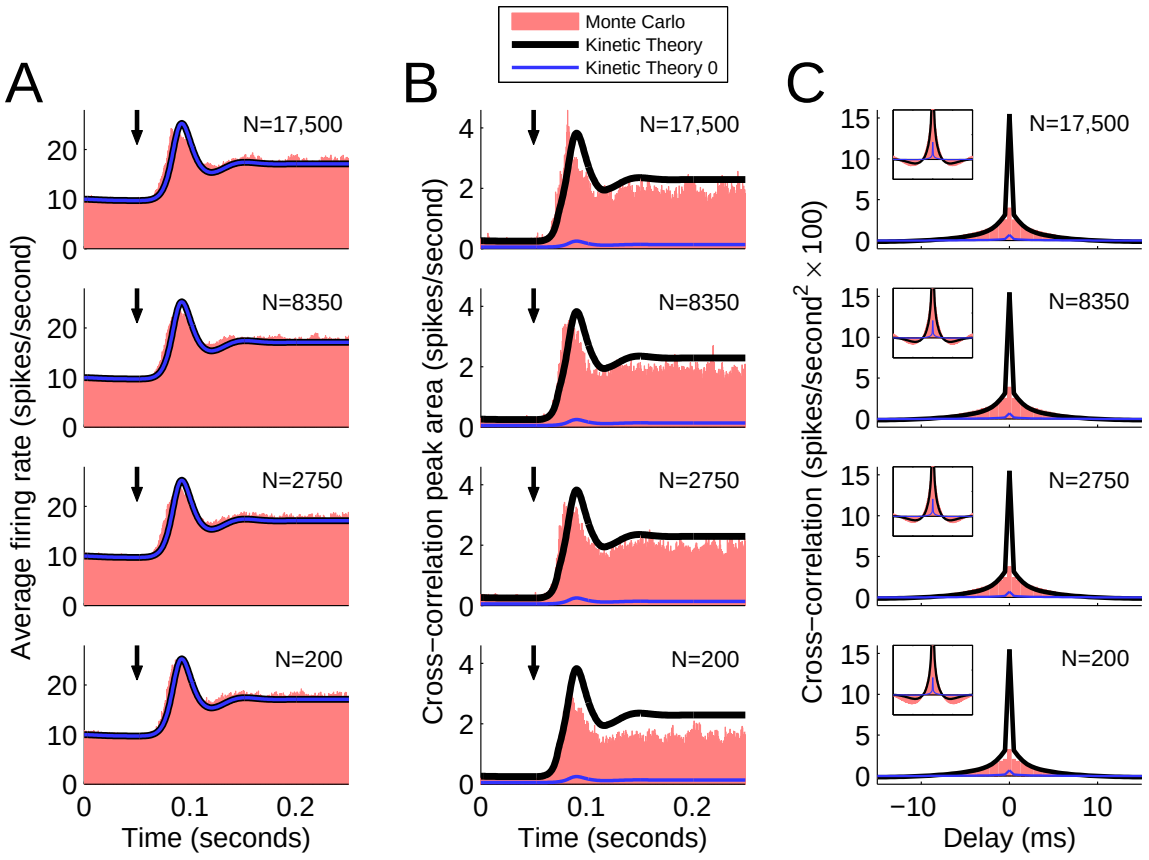


Figure 2.14: Comparison of layer 10 results from four different networks with different numbers of neurons but the same connectivity statistics ($W = 10$ and $\beta = 0.05$). The top row is a network with $N = 17,500$ neurons per layer and a power law outgoing degree distribution capped at a maximum outgoing degree of 5000. In the second and third row, the power law degree distribution was capped at 2000 and 500, respectively, and the number of neurons per layer was 8350 and 2750, respectively. The fourth row is a random network with $N = 200$ neurons per layer. In each case, network activity was obtained in response to 2000 realizations of same input as for figures 2.7–2.11, and the network circuitry was regenerated for each realization. Except for this circuitry regeneration, the fourth row is the same as the bottom row of figure 2.7 and the bottom left of of figure 2.9. The kinetic theory results are simply replotted for each row. **A**. Average firing rate $r_{\text{ave}}(t)$ plotted as in figure 2.7A. **B**. Cross-correlation peak area $C_{\text{peak}}(t)$ plotted as in figure 2.7B. **C**. Steady state cross-correlation structure plotted as in figure 2.9. Despite the vastly differing network size, the network output is similar because for each network class, we chose the network size to match the connectivity statistics $W = 10$ and $\beta = 0.05$.

2.8 Discussion of initial approach

We have demonstrated that the implementation of our initial kinetic theory approach works well up to moderate levels of correlation but that it fails to capture the network behavior once the correlation becomes too large. The result is understandable given the assumptions we invoked to derive the network equations in Section 2.5. In particular, the kinetic theory equations (2.2) include input rates representing only independent input to a single neuron and synchronous arrival of a single input to each of a pair of neurons. The equations neglect the possibility that two (or more) synchronous inputs could arrive at the same neuron. Such an event would lead to a voltage jump of double (or larger) size. A double jump is more effective in driving a neuron than two independent jumps of normal size. Since such double jumps become more common with increased network synchrony, they are presumably behind the increase in firing rate $r_{\text{ave}}(t)$ with β that is not captured by KT1 (Figure 2.13A).

The double-sized jumps have an even more profound effect on cross-correlation because the double jumps in one neuron's voltage could occur simultaneously with jumps (or double jumps) in another neuron's voltage. These synchronous jumps of double (or larger) size lead to correlations that cannot be captured by our initial kinetic theory equations, as they are limited to synchronous jumps of normal size. The saturation to maximal possible correlation in Figures 2.12 and 2.13B occurs when the expression (2.13) for the synchronous input rate ν_{syn} becomes larger than the total input from presynaptic populations. To ensure that the independent input rate ν_{ind} given by Eq. (2.14) is not smaller than the external input rate ν_{ext} (which we know is independent), we truncate ν_{syn} to the total input from presynaptic populations. For the given average firing rate and the given external input rates, it is simply impossible for the kinetic theory equations to yield higher correlation than observed

in Figures 2.12 and 2.13B. By looking at single population with prescribed input rates ν_{ind} and ν_{syn}, we confirmed that the saturation points of our initial kinetic theory implementation correspond to the point where all input to the population is synchronous except the independent input corresponding to the external input rate ν_{ext}.

To extend the validity of our kinetic theory equations to higher levels of correlation, we must augment the equations with terms corresponding to double jumps. We would need to add terms corresponding to independent double jumps in one neuron, synchronous double jumps in both neurons, and double jumps in one neuron that were synchronous with single jump in the other. The corresponding input rates which we denote by $\nu_{2,0}$, $\nu_{0,2}$, $\nu_{2,2}$, $\nu_{2,1}$ and $\nu_{1,2}$ can be determined by the same analysis as done in the expression (2.13). In this way, we could fully represent the pattern of inputs to a pair of neurons that could result from a pair of presynaptic neurons firing synchronously.

Such an augmentation of the equations would not account for the input patterns from a highly synchronous presynaptic population. In the highly synchronous case, three or more neurons may be highly likely to fire simultaneously. Such firing patterns will lead to combinations of double or larger sized voltage jumps that cannot be captured by an analysis of just pairs of presynaptic neurons firing synchronously. Although it may naively appear that we need to represent third or higher order statistics to capture such firing patterns, one can still infer the presence of higher order firing patterns from second order statistics.

Imagine, for example, that a presynaptic population contained three neurons with identical firing rates and pairwise correlation, as assumed by our kinetic theory. If the correlation between each pair was high, then it would be highly likely that each pair was either silent or firing together (recall, we are ignoring delayed correlation).

Since in this extreme case, it would be unlikely to find a pair where one neuron is firing and another is silent, we immediately observe that it would be rare to have only one out of the three neurons fire or have two of the three neurons fire without the remaining neuron joining in the firing. The likely states would be that either all three neurons are silent or all three neurons are firing together.

This extreme example illustrates some of the implications of second order statistics on the likelihood of observing higher order firing patterns. Thus, in order to improve the kinetic theory performance under high correlation, we can not neglect the effect of these higher order firing patterns of a presynaptic population on the first and second order firing statistics of a postsynaptic population in our feed-forward kinetic theory model, which is the main topic of the next chapter.

Chapter 3

Improved kinetic theory approach

The implications of second order firing statistics on higher order statistics become an important consideration for designing our second order kinetic theory approach. As pairwise correlations increase, so does the likelihood that more than two presynaptic neurons may fire simultaneously. In this case, not only do the double jumps described above become prevalent, but also triple and larger jumps start to have an effect. Hence, we propose to improve the kinetic theory performance under high correlation by estimating the rate of synchronous firing by more than two neurons and including the effect of such events in the evolution of the postsynaptic populations.

There is a lot flexibility in what higher order statistics could correspond to prescribed first and second order statistics if one isn't given any additional structure of the higher order statistics. One can infer predictions about higher order statistics by assuming minimal structure using maximum entropy estimates of the higher order statistics (Jaynes, 1957). Such methods were behind efforts to determine if one needed to assume higher order structure in neuronal activity in order to predict firing patterns among simultaneously recorded neurons (Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu et al, 2008). For point processes, such calculations could

be based on the entropy rates of point processes (Mcfadden, 1965). There is, however, no well-defined entropy rate of a point process given in terms of temporal units such as the firing of neurons. In such a situation, one may propose to infer predictions about higher order statistics of point processes based on the well-defined concept of relative entropy rates. For example, one could approximate the rates at which three or more neurons fire synchronously as those rates that maximize the relative entropy rate of the joint firing of those neurons with respect to the average firing process.

Unfortunately, as one will see later, though this method is theoretically feasible, even under the simple framework of multivariate Poisson processes, it is hard to solve the resulting optimization problem in an efficient way for a large number of neurons. Therefore, we propose another inference method which is simple and efficient. This method is developed under the assumptions of a multivariate Poisson process and a certain kind of conditional independence structure that we will describe later. Thus we will call it the conditional independence method. These assumptions will be discussed in detail in context. Briefly, we pretend that any higher order correlation occurs instantaneously in the same way as we approximated delayed as instantaneous in the previous chapter. We then prescribe a specific conditional independence structure on the probability distribution induced by the multivariate Poisson process. From the viewpoint of entropy, this structure also maximizes the entropy of some specific conditional probability distribution.

To include the effect of such inferred firing patterns in the evolution of the postsynaptic populations, we have to derive a new set of network coupling equations by transforming the inferred firing patterns of each presynaptic population's output into the input to each postsynaptic population. This new coupling equations will be based on (i) the assumption of modulated Poisson processes made in the input to each population and (ii) the first and second order statistics of connectivity. We will also

need to make truncation approximations to determine input that we can tractably implement with kinetic theory network equations.

In this chapter, we introduce our assumption of a multivariate Poisson process in section 3.1. Since it is natural to think of the maximum relative entropy rate method under this assumption, we present it and indicate its difficulty in section 3.2. Our conditional independence method is presented in section 3.3. We derive our improved network equations in section 3.4. We gauge the performance of this improved kinetic theory approach in section 3.5. We discuss this new implementation in section 3.6.

3.1 Multivariate Poisson processes assumption

Recall that in the derivation of the network coupling equations in the previous chapter (section 2.5) we pretended that delayed correlation of each presynaptic population was instantaneous and implicitly assumed independence among synchronous firing processes of different pairs of neurons. We made this simplification so that the input to each population would be a modulated Poisson process, as our model assumed. To keep this input structure under high correlation, we pretend in the same manner that any higher order correlation occurs instantaneously and explicitly assume the independence among different firing processes. For example, we assume the synchronous firing of one group of neurons is independent from any other group of neurons. Under such assumptions, one possible structure is the multivariate Poisson process specified below. Also, as one will see, the multivariate Poisson process can provide a simple framework to develop an efficient approach to inferring various firing processes.

We use the multivariate Poisson process to describe the firing patterns of any group of neurons in the population. Recall that we assume each neuron in the population is identical, meaning the firing statistics of each neuron are identical. Because we

ignore delayed correlations in the output of the population, the firing activity at each time point is independent of all other time points and the firing process of any group of neurons in the population can be described by a multivariate Poisson process.

Consider a group of M identical neurons indexed by i. We assume neuron i fires alone according to an independent Poisson process at rate $r_1(t)$. This is independent of i since we assume identical neurons. Moreover, we assume that any subset of m neurons fires synchronously according to an independent Poisson process. Let $r_m(t)$ be the rate at which exactly neurons $i_1, \cdots, i_m$ (and no other neurons in the group) fire synchronously. This rate depends only on size m since we assume identical neurons. Under this assumption, it follows that the firing process of any one neuron (ignoring all other neurons in the group) is a Poisson process. The rate of this process must be the average firing rate $r_{\text{ave}}(t)$. Similarly, the firing process by which any pair of neurons fires synchronously (ignoring the activity of all other neurons in the group) is also a Poisson process. This synchronous firing rate must be $\tilde{r}_{\text{syn}}(t)$.

To derive an expression relating the $r_m(t)$ to $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$, we start by considering the case of $M = 3$ neurons. The multivariate Poisson process description of the firing patterns of three identical neurons is illustrated by the Figure 3.1. First of all, each neuron can independently generate spikes according to a Poisson process at rate $r_1(t)$, which is indicated by the thin vertical lines. Second, each pair of neurons can independently generate synchronous spikes according to another Poisson process at rate $r_2(t)$, which is indicated by the thick vertical lines. Note the Poisson processes at rate $r_1(t)$ are independent of those at rate $r_2(t)$. Furthermore, all three neurons can fire spikes simultaneously according to another independent Poisson process at rate $r_3(t)$, which is indicated by the broken thick vertical lines. If we look at the firing process of any one neuron, for example Neuron 1, we can find its spikes can be alone, synchronous only with Neuron 2 or Neuron 3, or synchronous with both Neuron 2

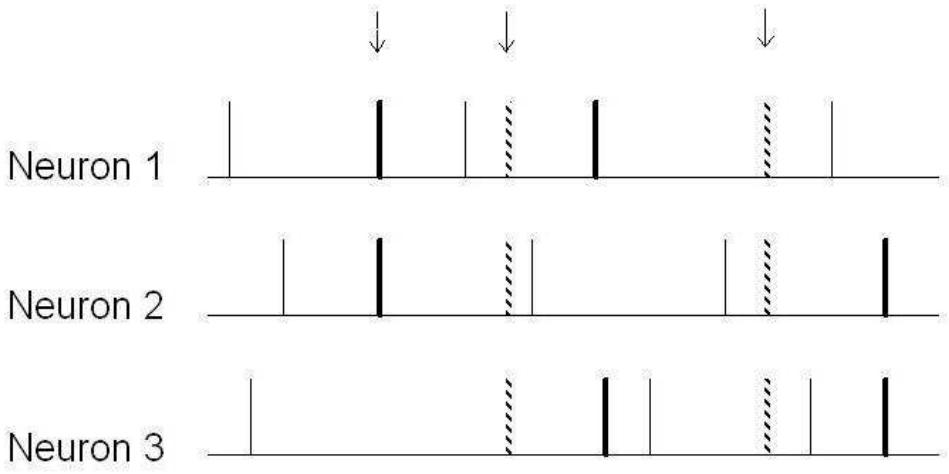


Figure 3.1: Illustration of the multivariate Poisson process description of three neurons' firing patterns. The thin vertical lines represent the spikes generated by each neuron according to independent Poisson processes with identical rates. The thick vertical lines represent the synchronous spikes generated by each pair of neurons according to another set of independent Poisson processes with identical rates. The broken thick vertical lines represent the synchronous spikes generated by all three neurons at the same time according to another independent Poisson process. Therefore, the firing process of any one neuron such as Neuron 1 is a Poisson process. Similarly, the synchronous firing process of any pair of neurons such as Neuron 1 and Neuron 2 as indicated by the arrows is also a Poisson process.

and Neuron 3. It is similar to Neuron 2 and Neuron 3. Therefore, the firing process of any one neuron is a Poisson process at rate $r_1(t) + 2r_2(t) + r_3(t)$. Since the rate of any one neuron's firing process must be the average firing rate $r_{\text{ave}}(t)$, we thus can relate $r_1(t)$, $r_2(t)$, and $r_3(t)$ to $r_{\text{ave}}(t)$ by the following

$$r_{\text{ave}}(t) = r_1(t) + 2r_2(t) + r_3(t). \tag{3.1}$$

On the other hand, if we look at the synchronous firing process of any pair of neurons, for example Neuron 1 and Neuron 2, we can find their simultaneous spikes can be alone or synchronous with Neuron 3. It is similar to any other pair of neurons. Consequently, the synchronous firing process of any pair of neurons is a Poisson process at rate $r_2(t) + r_3(t)$. By the fact that the rate of the synchronous firing process of any pair of neurons is equal to $\tilde{r}_{\text{syn}}(t)$, we thus have another relationship between the $r_i(t)$ and $\tilde{r}_{\text{syn}}(t)$ as follows

$$\tilde{r}_{\text{syn}}(t) = r_2(t) + r_3(t). \tag{3.2}$$

By the same reasoning, we can derive the following expression relating $r_m(t)$, $m = 1, \cdots, M$ to $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$ for any group of M identical neurons if the underlying firing process can be described by a multivariate Poisson process

$$\begin{aligned} r_{\text{ave}}(t) &= \sum_{i=1}^{M} \binom{M-1}{i-1} r_i(t), \\ \tilde{r}_{\text{syn}}(t) &= \sum_{i=2}^{M} \binom{M-2}{i-2} r_i(t). \end{aligned} \tag{3.3}$$

As a result, under the assumption of a multivariate Poisson process, given the first and second order firing statistics $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$, the rates of various firing patterns

of any group of M identical neurons will be constrained by Eq. (3.3).

3.2 Maximum relative entropy rate method

Under the assumption of a multivariate Poisson process, one possible approach to inferring various firing rates $r_1(t), r_2(t), \cdots, r_M(t)$ out of any group of M identical neurons in a population under the constraint (3.3) is based on the well-defined concept of relative entropy rate for point processes. The relative entropy rate at time t of a Poisson process of rate $r(t)$ with respect to another Poisson process of rate $r_{\text{ref}}(t)$ is given by

$$R_H(r; r_{ref})(t) = r(t)\left(1 - \log \frac{r(t)}{r_{ref}(t)}\right). \tag{3.4}$$

This expression can be generalized to the firing processes of any M sampled neurons with the average firing process $r_{\text{ave}}(t)$ as the reference process. It is given by

$$R_H(r_1, r_2, \cdots, r_M; r_{\text{ave}})(t) = \sum_{i=1}^{M} \left(\begin{array}{c} M \\ i \end{array} \right) r_i(t)\left(1 - \log \frac{r_i(t)}{r_{\text{ave}}(t)}\right). \tag{3.5}$$

The reason that it can be expressed in terms of the sum and the coefficient $\left(\begin{array}{c} M \\ i \end{array} \right)$ is because we have assumed independence among different firing processes. By following the same idea of maximum entropy estimates of higher order statistics (Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu et al, 2008), we propose to estimate $r_1(t), r_2(t), \cdots, r_M(t)$ by maximizing the relative entropy rate given by (3.5) under the constraint (3.3). Thus, this inference problem can be formally expressed as the

following optimization problem

$$\text{Maximize } R_H(r_1, r_2, \cdots, r_M; r_{\text{ave}})(t) = \sum_{i=1}^{M} \binom{M}{i} r_i(t) \left(1 - \log \frac{r_i(t)}{r_{\text{ave}}(t)}\right)$$

$$\text{Subject to } r_{\text{ave}}(t) = \sum_{i=1}^{M} \binom{M-1}{i-1} r_i(t), \text{and}$$

$$\tilde{r}_{\text{syn}}(t) = \sum_{i=2}^{M} \binom{M-2}{i-2} r_i(t). \tag{3.6}$$

We will not discuss the existence and uniqueness of solution for this optimization problem here. Instead, we focus on how this optimization problem can be transformed into solving a system of nonlinear algebraic equations, and consequently how the computational challenge can arise as the sampled size M becomes large.

By taking the first order derivative of the function R_H with respect to each variable r_i for $i = 3, \cdots, M$ and setting them to zero, we obtain an expression of r_i in terms of r_1 and r_2 for $i = 3, \cdots, M$. Then, by inserting these expressions into the constraints (3.3), the problem (3.6) can be transformed into solving the following system of nonlinear algebraic equations with two unknown r_1 and r_2.

$$\begin{aligned} r_1 &= r_{\text{ave}} - a_1 \tilde{r}_{\text{syn}} + \sum_{i=3}^{M} a_i r_1^{b_i} r_2^{c_i} r_{\text{ave}}^{d_i} \\ r_2 &= \tilde{r}_{\text{syn}} - \sum_{i=3}^{M} e_i r_1^{b_i} r_2^{c_i} r_{\text{ave}}^{d_i} \end{aligned} \tag{3.7}$$

where $a_1 = \binom{M-1}{1}$, $a_i = \binom{M-1}{1}\binom{M-2}{i-2} - \binom{M-1}{i-1}$,

$e_i = \binom{M-2}{i-2}$, $b_i = a_i \binom{M}{i}^{-1} \binom{M}{1}$, $c_i = e_i \binom{M}{i}^{-1} \binom{M}{2}$,

$$d_i = 1 + a_i \begin{pmatrix} M \\ i \end{pmatrix}^{-1} \begin{pmatrix} M \\ 1 \end{pmatrix} + e_i \begin{pmatrix} M \\ i \end{pmatrix}^{-1} \begin{pmatrix} M \\ 1 \end{pmatrix}$$

Though there are just two equations with two unknowns, the degrees appearing in the power of r_1 and r_2 become very large as the sampled size M becomes large, which makes it challenging to solve this system of nonlinear algebraic equations numerically. Fortunately, we may not need to devise an efficient method to solve the system (3.7) since it may not be necessary to use the maximum relative entropy rate method. From the Monte Carlo simulation results of Figure 2.13, we have evidence that the behavior of first and second order statistics of firing activity may not be sensitive to details of higher order statistics. Hence, as long as we constrain the lower order statistics, we may have flexibility in choosing how to infer the higher order statistics. Therefore, in the next section, we propose a simple and efficient method to infer higher order firing patterns constrained by lower order statistics.

3.3 Conditional independence method

Given first and second order firing statistics $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$ of any group of M identical neurons with the underlying firing process described by a multivariate Poisson process, another possible approach to make predictions about various firing rates $r_1(t), r_2(t), \cdots, r_M(t)$ is based on the conditional independence method which we propose below.

Similar to the presentation of multivariate Poisson processes, we present our conditional independence method by the illustration of $M = 3$ first. Recall that since we assume homogeneity across neurons, we need only to determine the rates $r_1(t)$, $r_2(t)$ and $r_3(t)$ of one, two and three neurons firing synchronously.

Given a multivariate Poisson process with $M = 3$, in terms of probability over the time interval dt, we can express $r_1(t)$, $r_2(t)$ and $r_3(t)$ by

$$\begin{aligned}
&\Pr(\text{just Neuron 1 fires a spike over the interval } [t, t+dt)) \\
&\qquad = r_1(t)dt + O(dt^2), \\
&\Pr(\text{just Neuron 1 and Neuron 2 fire a spike over the interval } [t, t+dt)) \\
&\qquad = r_2(t)dt + O(dt^2), \\
&\Pr(\text{all three neurons fire a spike over the interval } [t, t+dt)) \\
&\qquad = r_3(t)dt + O(dt^2). \qquad (3.8)
\end{aligned}$$

If we define $P(x_1, x_2, x_3)$ as the probability distribution of the spike number x_1, x_2 and x_3 of Neuron 1, 2 and 3 respectively over interval dt, by utilizing conditional probability, the above expressions can be rewritten as

$$\begin{aligned}
P(1,0,0) &= P(0,0 \mid 1)P(1) = r_1(t)dt + O(dt^2), \\
P(1,1,0) &= P(1,0 \mid 1)P(1) = r_2(t)dt + O(dt^2), \\
P(1,1,1) &= P(1,1 \mid 1)P(1) = r_3(t)dt + O(dt^2). \qquad (3.9)
\end{aligned}$$

where $P(1)$ is the probability of Neuron 1 firing a spike over the interval $[t, t+dt)$ and $P(\cdot, \cdot \mid 1)$ denotes the probability distribution of Neuron 2 and Neuron 3 firing a spike or not over $[t, t+dt)$ conditioned on Neuron 1 firing a spike over $[t, t+dt)$.

Now we introduce the independence structure to the conditional probability distribution $P(\cdot, \cdot \mid 1)$, that is, we assume the equality $P(\cdot, \cdot \mid 1) = P(\cdot \mid 1)P(\cdot \mid 1)$ holds.

Then, we have

$$
\begin{aligned}
P(0,0 \mid 1) &= P(0 \mid 1)P(0 \mid 1) = \left(1 - \frac{\tilde{r}_{\text{syn}}(t) + O(dt)}{r_{\text{ave}}(t) + O(dt)} + O(dt)\right)^2, \\
P(1,0 \mid 1) &= P(1 \mid 1)P(0 \mid 1) = \left(\frac{\tilde{r}_{\text{syn}}(t) + O(dt)}{r_{\text{ave}}(t) + O(dt)}\right)\left(1 - \frac{\tilde{r}_{\text{syn}}(t) + O(dt)}{r_{\text{ave}}(t) + O(dt)} + O(dt)\right), \\
P(1,1 \mid 1) &= P(1 \mid 1)P(1 \mid 1) = \left(\frac{\tilde{r}_{\text{syn}}(t) + O(dt)}{r_{\text{ave}}(t) + O(dt)}\right)^2.
\end{aligned} \tag{3.10}
$$

where we have utilized the facts that $P(0 \mid 1) = 1 - P(1 \mid 1)$ and $P(1 \mid 1) = \frac{P(1,1)}{P(1)} = \frac{\tilde{r}_{\text{syn}}(t)+O(dt)}{r_{\text{ave}}(t)+O(dt)}$.

By inserting the expressions (3.10) into (3.9), utilizing the fact that $P(1) = r_{\text{ave}}(t)dt$, dividing dt on both sides, and taking the limit of $dt \to 0$, we can obtain the following expressions of $r_1(t)$, $r_2(t)$ and $r_3(t)$ in terms of the first and second order statistics $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$,

$$
\begin{aligned}
r_1(t) &= (1 - a(t))^2 r_{\text{ave}}(t), \\
r_2(t) &= a(t)(1 - a(t)) r_{\text{ave}}(t), \\
r_3(t) &= a(t)^2 r_{\text{ave}}(t).
\end{aligned} \tag{3.11}
$$

where we denote by the variable $a(t)$ the ratio $\frac{\tilde{r}_{\text{syn}}(t)}{r_{\text{ave}}(t)}$. Similarly, we can have the following general expression of $r_i(t)$ of $i = 1, \cdots, M$ for any group of M identical neurons,

$$
r_i(t) = a(t)^{i-1}(1 - a(t))^{M-i} r_{\text{ave}}(t), \qquad i = 1, \cdots, M \tag{3.12}
$$

Furthermore, this simple expression of $r_i(t)$ satisfies the constraint conditions (3.3) which can easily be verified by using the binomial expansion. For the first condition

of the constraints (3.3), we have

$$\begin{aligned}\sum_{i=1}^{M}\binom{M-1}{i-1}r_i(t) &= \sum_{i=1}^{M}\binom{M-1}{i-1}a(t)^{i-1}(1-a(t))^{M-i}r_{\text{ave}}(t)\\ &= \sum_{j=0}^{M-1}\binom{M-1}{j}a(t)^{j}(1-a(t))^{M-1-j}r_{\text{ave}}(t)\\ &= (a(t)+(1-a(t)))^{M-1}r_{\text{ave}}(t) = r_{\text{ave}}(t). \qquad (3.13)\end{aligned}$$

Similarly, for the second condition of the constraints (3.3), we have

$$\begin{aligned}\sum_{i=2}^{M}\binom{M-2}{i-2}r_i(t) &= \sum_{i=2}^{M}\binom{M-2}{i-2}a(t)^{i-1}(1-a(t))^{M-i}r_{\text{ave}}(t)\\ &= \sum_{i=2}^{M}\binom{M-2}{i-2}a(t)^{i-2}(1-a(t))^{M-i}a(t)r_{\text{ave}}(t)\\ &= \sum_{i=2}^{M}\binom{M-2}{i-2}a(t)^{i-2}(1-a(t))^{M-i}\tilde{r}_{\text{syn}}(t)\\ &= \sum_{j=0}^{M-2}\binom{M-2}{j}a(t)^{j}(1-a(t))^{M-2-j}\tilde{r}_{\text{syn}}(t)\\ &= (a(t)+(1-a(t)))^{M-2}\tilde{r}_{\text{syn}}(t) = \tilde{r}_{\text{syn}}(t) \qquad (3.14)\end{aligned}$$

Therefore, by introducing a specific conditional independence structure, we can relate the rates $r_i(t), i = 1, \cdots, M$ to the first and second order statistics $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$ by a simple expression (3.12) which satisfies the constraints (3.3) and thus provides us an efficient method to predict the rates $r_i(t), i = 1, \cdots, M$ under the lower order constraints (3.3).

3.4 Derivation of improved network equations

In the previous chapter, we derived our initial network equations by a method which transforms the output of a presynaptic population to the input of a postsynaptic population. This method was developed on the basis of an analysis of just pairs of presynaptic neurons firing synchronously, completely ignoring the synchronous firing of more than two neurons. In order to improve the kinetic theory performance under high correlation, we have so far developed methods to estimate various firing patterns of the multivariate Poisson process which we suppose to represent the firing process of any sampled group of neurons. The next step we have to do is to include the effect of such patterns in the evolution of the postsynaptic populations.

In this section, we will derive our improved network coupling equations under the same framework as in section 2.5, except that the three random variables N_1^j, N_2^j and N_3^j will be redefined as the number of neurons from population j that project onto only the first neuron, the number of neurons from population j that project onto only the second neuron, and the number of neurons from population j that project onto both postsynaptic neurons respectively. Moreover, they will be specified by some probability distribution $P(N_1^j, N_2^j, N_3^j)$ which satisfies the connectivity statistics $E(N_1^j + N_3^j) = E(N_2^j + N_3^j) = W_{jk}^1$ and $E(N_3^j) = W_{jk}^2$.

We consider a pair of neurons in population k and denote by $\nu_{m,n}^k(t)$ the rate of input onto both neurons with which the underlying Poisson process would cause m-tuple jumps for neuron 1 and simultaneously n-tuple jumps for neuron 2. To calculate such input rates $\nu_{m,n}^k(t)$ for $m, n = 0, 1, 2, \cdots$, we will first calculate them conditioned on particular values of $\mathbf{N}_{\text{in}} = \{N_i^j\}$ and then take the expected values over $\mathbf{N}_{\text{in}}$.

Define $\nu_{m,n}^k(t; \mathbf{N}_{\text{in}})$ to be such an input rate conditioned on particular values of

the $\mathbf{N}_{\text{in}}$. We calculate that this input rate is

$$\nu^k_{m,n}(t;\mathbf{N}_{\text{in}}) = \sum_j \Bigg(\sum_{\substack{\alpha,\beta,\gamma \\ \alpha+\gamma=m \\ \beta+\gamma=n}} \binom{N_1^j}{\alpha} \binom{N_2^j}{\beta} \binom{N_3^j}{\gamma} r^j_{\alpha+\beta+\gamma}(t;\mathbf{N}_{\text{in}}) \Bigg) + \delta(m+n-1)\nu^k_{\text{ext}}(t). \tag{3.15}$$

where $r^j_{\alpha+\beta+\gamma}(t;\mathbf{N}_{\text{in}})$, as defined before, is the synchronous firing rate at which exactly $\alpha+\beta+\gamma$ fixed neurons out of the $N_1^j+N_2^j+N_3^j$ sampled neurons in population j fire simultaneously, $\delta(\cdot)$ is the Dirac delta function, and $\nu^k_{\text{ext}}(t)$ is the external independent input rate to population k.

The event that there are exactly α neurons out of the N_1^j neurons projecting only to neuron 1, exactly β neurons out of the N_2^j neurons projecting only to neuron 2, and exactly γ neurons out of the N_3^j neurons projecting to both neurons firing simultaneously, can result in the synchronous input onto both neurons with $(\alpha+\gamma)$-tuple jumps for neuron 1 and $(\beta+\gamma)$-tuple jumps for neuron 2. Moreover, the rate at which this event can occur is equal to $\binom{N_1^j}{\alpha}\binom{N_2^j}{\beta}\binom{N_3^j}{\gamma} r^j_{\alpha+\beta+\gamma}(t;\mathbf{N}_{\text{in}})$, and consequently contributes to part of the input rate $\nu^k_{m,n}(t;\mathbf{N}_{\text{in}})$ with $m=\alpha+\gamma$ and $n=\beta+\gamma$. Therefore, the contribution of population j to $\nu^k_{m,n}(t;\mathbf{N}_{\text{in}})$ for $m,n=0,1,2,\cdots$ can be easily obtained by taking the sum of $\binom{N_1^j}{\alpha}\binom{N_2^j}{\beta}\binom{N_3^j}{\gamma} r^j_{\alpha+\beta+\gamma}(t;\mathbf{N}_{\text{in}})$ over all possible combinations of the integers α, β and γ with $0\le\alpha\le N_1^j$, $0\le\beta\le N_2^j$, $0\le\gamma\le N_3^j$, $\alpha+\gamma=m$, and $\beta+\gamma=n$. In addition, any external independent input to population k at rate $\nu^k_{\text{ext}}(t)$ can contribute to independent input rates with single jumps $\nu_{1,0}(t;\mathbf{N}_{\text{in}})$ and $\nu_{0,1}(t;\mathbf{N}_{\text{in}})$.

Next, we take the expected value of $\nu^k_{m,n}(t;\mathbf{N}_{\text{in}})$ over $\mathbf{N}_{\text{in}}$. Without specifying the

probability distribution $P(N_1^j, N_2^j, N_3^j)$ for $\mathbf{N}_{\text{in}}$, it simply is

$$\begin{aligned}\nu_{m,n}^k(t) &= E(\nu_{m,n}^k(t;\mathbf{N}_{\text{in}})) \\ &= \sum_j \Bigg(\sum_{N_1^j,N_2^j,N_3^j} \sum_{\substack{\alpha,\beta,\gamma \\ \alpha+\gamma=m \\ \beta+\gamma=n}} \binom{N_1^j}{\alpha} \binom{N_2^j}{\beta} \binom{N_3^j}{\gamma} r_{\alpha+\beta+\gamma}^j(t;\mathbf{N}_{\text{in}}) P(N_1^j, N_2^j, N_3^j) \Bigg) \\ &\quad + \delta(m+n-1)\nu_{\text{ext}}^k(t). \end{aligned} \tag{3.16}$$

This expression indicates that the expected values of $\nu_{m,n}^k(t;\mathbf{N}_{\text{in}})$ over $\mathbf{N}_{\text{in}}$ would depend on the choice of the probability distribution $P(N_1^j, N_2^j, N_3^j)$, and thus would not only depend on the first and second order connectivity statistics, but also higher order statistics. Fortunately, by the evidence of Monte Carlo simulation results, we expect that the results shouldn't be too sensitive to the specific probability distribution $P(N_1^j, N_2^j, N_3^j)$ as long as it satisfies $E(N_1^j + N_3^j) = E(N_2^j + N_3^j) = W_{jk}^1$ and $E(N_3^j) = W_{jk}^2$. The corresponding expected values of $\nu_{m,n}^k(t;\mathbf{N}_{\text{in}})$ should be an appropriate candidate for the input to population k for the purpose of capturing the first and second order firing statistics. Thus, to make the calculation as simple as possible, we specify the probability distribution $P(N_1^j, N_2^j, N_3^j)$ to be the multinomial distribution which corresponds to the random network structure. That is, under the condition of $E(N_1^j + N_3^j) = E(N_2^j + N_3^j) = W_{jk}^1$ and $E(N_3^j) = W_{jk}^2$, we use the probability distribution

$$P(N_1^j, N_2^j, N_3^j) = \frac{N!}{N_1^j! N_2^j! N_3^j! (N - N_1^j - N_2^j - N_3^j)!} p_1^{N_1^j} p_2^{N_2^j} p_3^{N_3^j} p_4^{N - N_1^j - N_2^j - N_3^j}. \tag{3.17}$$

where N is the smallest integer greater than or equal to W_{jk}^1/W_{jk}^2 which can be viewed as the approximated population size, $p_1 = (W_{jk}^1/N)(1 - W_{jk}^1/N)$ is the probability that a given presynaptic neuron is connected only to neuron 1, $p_2 = (W_{jk}^1/N)(1 - W_{jk}^1/N)$

is the probability that a given presynaptic neuron is connected only to neuron 2, $p_3 = (W^1_{jk}/N)^2$ is the probability it is connected to both, and $p_4 = (1 - W^1_{jk}/N)^2$ is the probability it is connected to neither. If we further infer predictions about $r^j_{\alpha+\beta+\gamma}(t; \mathbf{N}_{\text{in}})$ by equation (3.12), we can then evaluate $\nu^k_{m,n}(t)$ for all possible m, n by

$$\begin{aligned}
\nu^k_{m,n}(t) &= E(\nu^k_{m,n}(t; \mathbf{N}_{\text{in}})) \\
&= \sum_j \Bigg(\sum_{N^j_1, N^j_2, N^j_3} \sum_{\substack{\alpha,\beta,\gamma \\ \alpha+\gamma=m \\ \beta+\gamma=n}} \binom{N^j_1}{\alpha} \binom{N^j_2}{\beta} \binom{N^j_3}{\gamma} \\
&\qquad (1 - a(t))^{(N^j_1 + N^j_2 + N^j_3 - (\alpha+\beta+\gamma))} a(t)^{(\alpha+\beta+\gamma-1)} r_{\text{ave}}(t) P(N^j_1, N^j_2, N^j_3) \Bigg) \\
&\quad + \delta(m + n - 1)\nu^k_{\text{ext}}(t). \qquad (3.18)
\end{aligned}$$

where $a(t) = \frac{\tilde{r}_{\text{syn}}(t)}{r_{\text{ave}}(t)}$ and $P(N^j_1, N^j_2, N^j_3)$ can be evaluated by formula (3.17).

If we take all these possible inputs to population k into account, the evolution

equation of $\rho^k(v_1, v_2, t)$ can be rewritten as

$$\begin{aligned}
\frac{\partial \rho^k}{\partial t}(v_1, v_2, t) = & \frac{1}{\tau}\frac{\partial}{\partial v_1}\left[(v_1 - E_r)\rho^k(v_1, v_2, t)\right] + \frac{1}{\tau}\frac{\partial}{\partial v_2}\left[(v_2 - E_r)\rho^k(v_1, v_2, t)\right] \\
& + \sum_m \nu_{m,0}(t)\left[\int_{v_{\text{reset}}}^{v_1} f_{mA}(v_1 - \theta_1)\rho^k(\theta_1, v_2, t)d\theta_1 - \rho^k(v_1, v_2, t)\right] \\
& + \sum_m \nu_{0,m}(t)\left[\int_{v_{\text{reset}}}^{v_2} f_{mA}(v_2 - \theta_2)\rho^k(v_1, \theta_2, t)d\theta_2 - \rho^k(v_1, v_2, t)\right] \\
& + \sum_{\substack{m,n \\ m\geq 1 \\ n\geq 1}} \nu_{m,n}(t)\Bigg[\int_{v_{\text{reset}}}^{v_1}\int_{v_{\text{reset}}}^{v_2} f_{mA}(v_1 - \theta_1)f_{nA}(v_2 - \theta_2)\rho^k(\theta_1, \theta_2, t)d\theta_2 d\theta_1 \\
& \qquad\qquad - \rho^k(v_1, v_2, t)\Bigg] \\
& + \delta(v_1 - v_{\text{reset}})J^k_{\text{reset},1}(v_2, t) + \delta(v_2 - v_{\text{reset}})J^k_{\text{reset},2}(v_1, t) \\
& + \delta(v_1 - v_{\text{reset}})\delta(v_2 - v_{\text{reset}})J^k_{\text{reset},3}(t). \qquad (3.19)
\end{aligned}$$

The interpretation of Eq. (3.19) is similar to that of Eq. (2.2) which we omit here. The factors $J^k_{\text{reset},i}$ are reformulated by

$$\begin{aligned}
J^k_{\text{reset},1}(v_2, t) = & \sum_m \nu_{m,0}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}} F_{mA}(v_{\text{th}} - \theta_1)\rho^k(\theta_1, v_2, t)d\theta_1 \\
& + \sum_{\substack{m,n \\ m\geq 1 \\ n\geq 1}} \nu_{m,n}(t)\int_{v_{\text{reset}}}^{v_2}\int_{v_{\text{reset}}}^{v_{\text{th}}} F_{mA}(v_{\text{th}} - \theta_1)f_{nA}(v_2 - \theta_2)\rho^k(\theta_1, \theta_2, t)d\theta_1 d\theta_2, \\
J^k_{\text{reset},2}(v_1, t) = & \sum_m \nu_{0,m}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}} F_{mA}(v_{\text{th}} - \theta_2)\rho^k(v_1, \theta_2, t)d\theta_2 \\
& + \sum_{\substack{n,m \\ n\geq 1 \\ m\geq 1}} \nu_{n,m}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}}\int_{v_{\text{reset}}}^{v_1} F_{nA}(v_{\text{th}} - \theta_2)f_{mA}(v_1 - \theta_1)\rho^k(\theta_1, \theta_2, t)d\theta_1 d\theta_2, \\
J^k_{\text{reset},3}(t) = & \sum_{\substack{m,n \\ m\geq 1 \\ n\geq 1}} \nu_{m,n}(t)\int_{v_{\text{reset}}}^{v_{\text{th}}}\int_{v_{\text{reset}}}^{v_{\text{th}}} F_{mA}(v_{\text{th}} - \theta_1)F_{nA}(v_{\text{th}} - \theta_2)\rho^k(\theta_1, \theta_2, t)d\theta_1 d\theta_2
\end{aligned} \qquad (3.20)$$

where $f_{mA}(x)$ is the probability density function of the random variable mA for the voltage jump size as neurons receive m simultaneous synaptic inputs and $F_{mA}(x)$ is the corresponding complementary cumulative distribution function, that is, $F_{mA}(x) = \int_x^\infty f_{mA}(t)dt = \Pr(mA > x)$. Since we have assumed the random variables for the voltage jump size induced from different synaptic inputs are independent and identically-distributed (i.i.d.) with probability density function $f_A(x)$, the random variable mA is just the sum of m such i.i.d. random variables and thus the probability density function $f_{mA}(x)$ is the the convolution product of m probability density functions $f_A(x)$.

3.4.1 Truncation approximations

Explicitly calculating large numbers of input patterns (3.18) would make intractable the numerical solution of the network equations for the evolution of the second order density of each population (3.19). To reduce the complexity, we propose a truncation method which consists of the following two procedures. This truncation method is based on the principle of preserving the average input rate and the ansatz of a diffusion approximation.

In the first procedure, we truncate all $\nu^k_{m,n}(t)$ for either m or n larger than H in Eq. (3.18) first. Thus, at this point, the truncated input $\tilde{\nu}_{m,n}$(t) is given by

$$\tilde{\nu}^k_{m,n}(t) = 0 \qquad \text{for either } m \text{ or } n \text{ larger than } H$$
$$\tilde{\nu}^k_{m,n}(t) = \nu^k_{m,n}(t) \qquad \text{for } 0 \leq m, n \leq H,$$

To preserve the average input rate, we update the independent input for single jumps

by

$$\tilde{\nu}^k_{1,0}(t) = \nu^k_{1,0}(t) + \sum_j W^1_{jk} r^j_{\text{ave}}(t) - \sum_{m=1}^{H}\sum_{n=0}^{H} m\nu^k_{m,n}(t)$$
$$\tilde{\nu}^k_{0,1}(t) = \nu^k_{0,1}(t) + \sum_j W^1_{jk} r^j_{\text{ave}}(t) - \sum_{n=1}^{H}\sum_{m=0}^{H} n\nu^k_{m,n}(t)$$

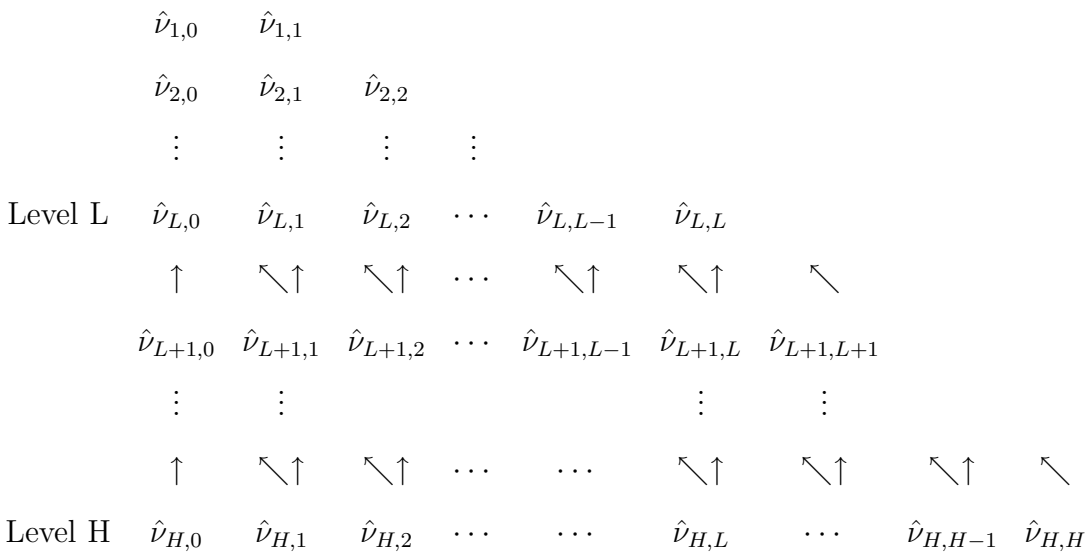
In other word, we simply approximate the inputs with rate $\nu^k_{m,n}(t)$ for either m or n larger than H as the combination of two independent inputs with rates $\nu^k_{1,0}(t) = m \cdot \nu^k_{m,n}(t)$ and $\nu^k_{0,1}(t) = n \cdot \nu^k_{m,n}(t)$. This first procedure of our truncation method is fairly crude since all higher order correlations above H are mapped into independent firing rates. Under the condition that $\tilde{r}_{\text{syn}}(t) < r_{\text{ave}}(t)$, it can be verified from Eq. (3.18) that the rates $\nu^k_{m,n}(t)$ are small for either m or n sufficiently large. Consequently, the resulting higher order correlations become small. In order to not lose too much higher order correlation, we choose a large value of H. In the next section of our implementation of the improved kinetic theory feed-forward network, we choose a fixed value of $H = 20$.

For computational efficiency, we want to make further truncation below $H = 20$, but want to do it in a way that preserves more information about the higher order statistics. To guide our choice of truncation method, we examine the structure of a diffusion approximation of equation (3.19), which would be valid for small jump sizes. The resulting equation is an advection-diffusion equation where the advection term and diffusion term can be characterized by a vector $\nu^k_{m,n}(t) \cdot \begin{pmatrix} m, n \end{pmatrix}$ and a matrix $\nu^k_{m,n}(t) \cdot \begin{pmatrix} m^2 & mn \\ mn & n^2 \end{pmatrix}$ respectively for each input $\nu^k_{m,n}(t)$. One can check that the average input of equation (3.19) is exactly the vector $\sum_{\substack{m=0 \\ n=0}} \nu^k_{m,n}(t) \cdot \begin{pmatrix} m, n \end{pmatrix}$. Since the first truncation procedure preserved the average input, this vector is equal to

$\sum_{\substack{m=0\\n=0}}^{H} \tilde{\nu}^k_{m,n}(t) \cdot \begin{pmatrix} m, n \end{pmatrix}$. To preserve the average input in the second procedure, the further truncated inputs $\hat{\nu}^k_{m,n}(t)$, $0 \leq m, n \leq L$ have to comply with the following equation

$$\sum_{\substack{m=0\\n=0}}^{L} \hat{\nu}^k_{m,n}(t) = \sum_{\substack{m=0\\n=0}}^{H} \tilde{\nu}^k_{m,n}(t) \tag{3.21}$$

There are a number of choices of the truncated inputs $\hat{\nu}^k_{m,n}(t)$, $0 \leq m, n \leq L$ which can comply with Eq. (3.21). For example, one can completely throw out correlations above L just the same as the first procedure. However, in our second procedure, we want to preserve more information about the higher order statistics of inputs $\tilde{\nu}^k_{m,n}(t)$ for either m or n between L and H, which we characterize by using the diffusion matrix with certain structure $\tilde{\nu}^k_{m,n}(t) \cdot \begin{pmatrix} m^2 & mn \\ mn & n^2 \end{pmatrix}$. In order to not lose too much higher order statistics of inputs $\tilde{\nu}^k_{m,n}(t)$ for either m or n between L and H, we choose the step-by-step update and truncation procedure which is illustrated by the following diagram.



First of all, we make $\hat{\nu}^k_{m,n}(t)$ equal to $\tilde{\nu}^k_{m,n}(t)$ for each $0 \leq n \leq m \leq H$. Second, for each step i from H to $L+1$, we update the inputs $\hat{\nu}^k_{i-1,j}(t)$ for $0 \leq j \leq i-1$ by mapping the statistics of inputs at level i into the statistics of inputs at level $i-1$. Third, by symmetry, we define $\hat{\nu}^k_{n,m}(t)$ by equal to $\hat{\nu}^k_{m,n}(t)$ for $0 \leq n \leq m \leq H$. Finally, we truncate the inputs $\hat{\nu}^k_{m,n}(t)$ for either m or n larger than L. The method of mapping the statistics of inputs at level i into the statistics of inputs at level $i-1$ is based on the nearest representation of each $\hat{\nu}^k_{i,j}(t)$ by the inputs at level $i-1$. This procedure preserves the advection vector and captures more of the diffusion matrix from $\hat{\nu}^k_{i,j}(t)$ than the first truncation procedure. The update can be done according the following algorithm

for $i = M-1$ to L **do**

$\mathbf{v}(i) \leftarrow \mathbf{v}(i) + A(i)\mathbf{v}^t(i+1)$

end for

where $\mathbf{v}(i)$ is a $(i+1)$-dimensional vector for each i and $\mathbf{v}(i) = (\hat{\nu}_{i,0}, \hat{\nu}_{i,1}, \cdots, \hat{\nu}_{i,i}, \hat{\nu}_{i,i+1})$. $A(i)$ is a $(i+1)$ by $(i+2)$ matrix for each i and is given by

$$\begin{pmatrix}
\frac{i+1}{i} & \frac{1}{i} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\
0 & 1 & \frac{2}{i} & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\
\vdots & 0 & \frac{i-1}{i} & \frac{3}{i} & 0 & \dots & \dots & \dots & \dots & 0 \\
\vdots & \vdots & 0 & \frac{i-2}{i} & \ddots & \ddots & \dots & \dots & \dots & 0 \\
\vdots & \vdots & \vdots & 0 & \ddots & \ddots & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \frac{i-2}{i} & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & 0 & \frac{3}{i} & \frac{i-1}{i} & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & \frac{2}{i} & 1 & 0 \\
0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 & \frac{1}{i} & \frac{i+1}{i}
\end{pmatrix}$$

3.5 Results of improved approach

To demonstrate the better performance of improved kinetic theory approach than the initial approach under high correlation (if we make appropriate truncation approximations), we implemented the feed-forward networks of section 2.7.3 with two different sets of truncated kinetic theory network equations which we term KT2 and KT4 and compared the results to the Monte Carlo simulations. Our KT2 and KT4 network equations were derived based on the second procedure of truncation method at level $L = 2$ and $L = 4$ respectively.

The results for $\beta = 0.1$ (i.e., $N = 100$ in the Monte Carlo simulations) are shown in Figure 3.2. For the average firing rate $r_{\text{ave}}(t)$ (left column), as compared with KT1 shown in Figure 2.8, both KT2 and KT4 show better results than KT1 by layer 10. This is because KT2 and KT4 include the dependence of the average firing rate of a postsynaptic population on the higher order firing statistics of a presynaptic population in the input while KT1 does not. On the other hand, for the cross-correlation peak area $C_{\text{peak}}(t)$ (middle column), KT2 displays similar behavior to KT1 except the slight difference at the response time of transient behavior, whereas KT4 shows better results than KT1 under high level of correlation which occurs in deeper layers, though it overestimates the correlation magnitude and the response time of transient behavior. The overestimation of steady-state correlation magnitude probably results from the instantaneous approximation of delayed correlation, which can be explained by the structure of the steady-state cross-correlation (right column).

If we doubled the fraction of shared input to $\beta = 0.2$ (i.e., $N = 50$ in the Monte Carlo simulations), for the cross-correlation peak area $C_{\text{peak}}(t)$ (middle column of Figure 3.3), it shows that KT4 captures a higher percentage of the Monte Carlo under very high level of correlation which occurs by layer 10 as compared with the

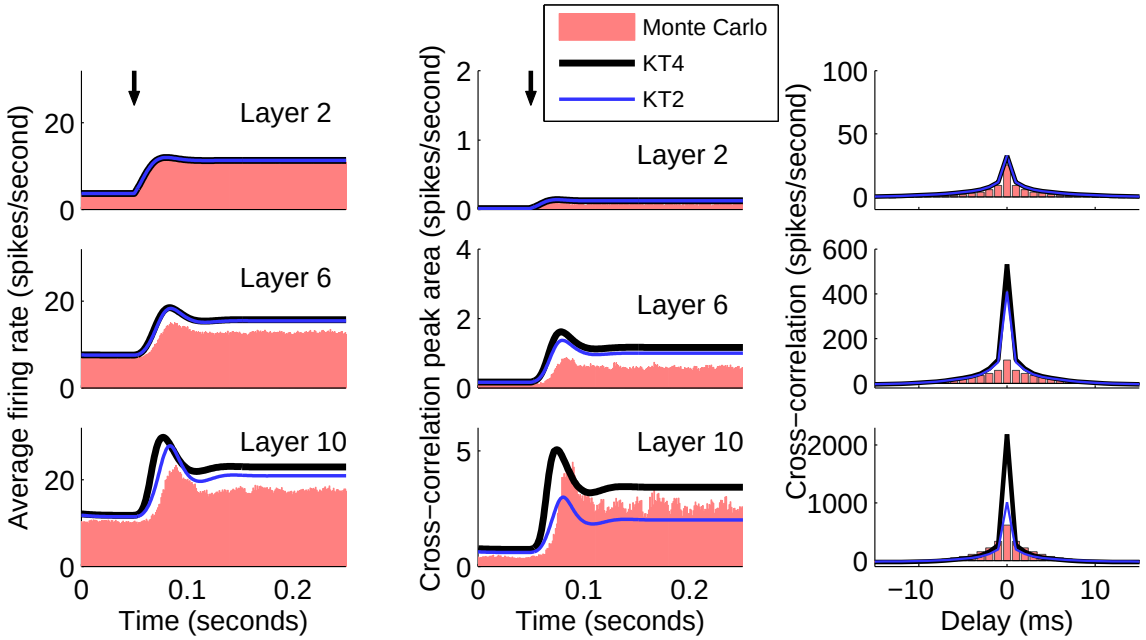


Figure 3.2: Demonstration of the performance of KT2 and KT4 for the network with $\beta = 0.1$. Both KT2 and KT4 slightly overestimate the average firing $r_{\text{ave}}(t)$, and they underestimate the delay in response to the input transient (left column). For the cross-correlation peak area $C_{\text{peak}}(t)$, both KT2 and KT4 show the build-up of correlation. Only KT4 makes a good approximation of the correlation magnitude in the deeper layers, though it overestimates the magnitude and underestimates the delay in response to the input transient (middle column). The overestimation of correlation magnitude at the steady state can be explained by the steady-state cross-correlation structure shown in the right column. KT4 overestimates the correlation at zero delay.

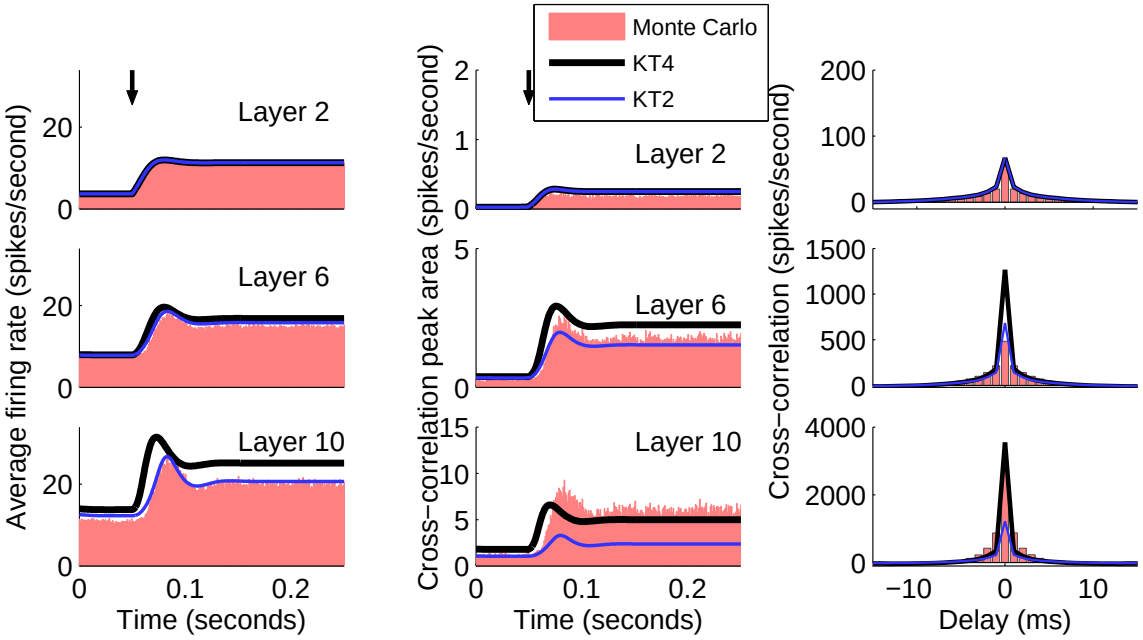


Figure 3.3: Demonstration of the performance of KT2 and KT4 for the network with $\beta = 0.2$. Though KT2 matches well the Monte Carlo for the average firing rate $r_{\text{ave}}(t)$, it poorly captures the cross-correlation peak area $C_{\text{peak}}(t)$ at the high level of correlation by layer 10. On the other hand, KT4 performs much better in capturing the cross-correlation peak area $C_{\text{peak}}(t)$, though it overestimates the average firing rate and slightly underestimates the correlation magnitude by layer 10. Its underestimation of the correlation magnitude at the steady state can be explained by the steady-state cross-correlation structure which shows that KT4 underestimates the correlation at non-zero delay.

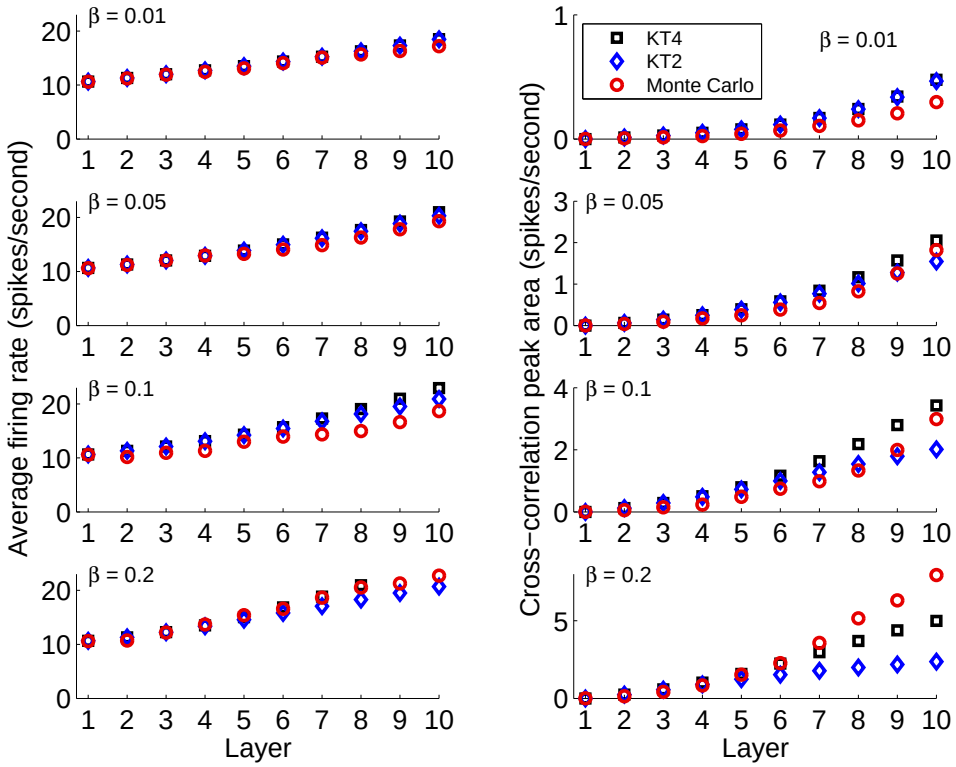


Figure 3.4: Demonstration of the performance of KT2 and KT4 at the steady-state values of the average firing rate r_{ave} and cross-correlation peak area C_{peak} for all layers of the four tested networks with $\beta = 0.01$, 0.05, 0.1 and 0.2. Both KT2 and KT4 capture well the steady-state average firing rate r_{ave} for all layers of these four networks, though KT2 slightly underestimates it in deeper layers of the network with $\beta = 0.2$ where the correlation is at a quite high level. For the steady-state cross-correlation peak area C_{peak}, it shows that KT4 captures well the build-up of the correlation with layer for each network, though it underestimates the steady-state correlation magnitude in deeper layers of the network with $\beta = 0.2$ where the correlation is at a quite high level. Though KT2 also shows the build-up of correlation, it performs more poorly than KT4 when correlation becomes high which occurs in the layer 10 of the networks with $\beta = 0.05$ and 0.1 and in the deeper layers of the network with $\beta = 0.2$.

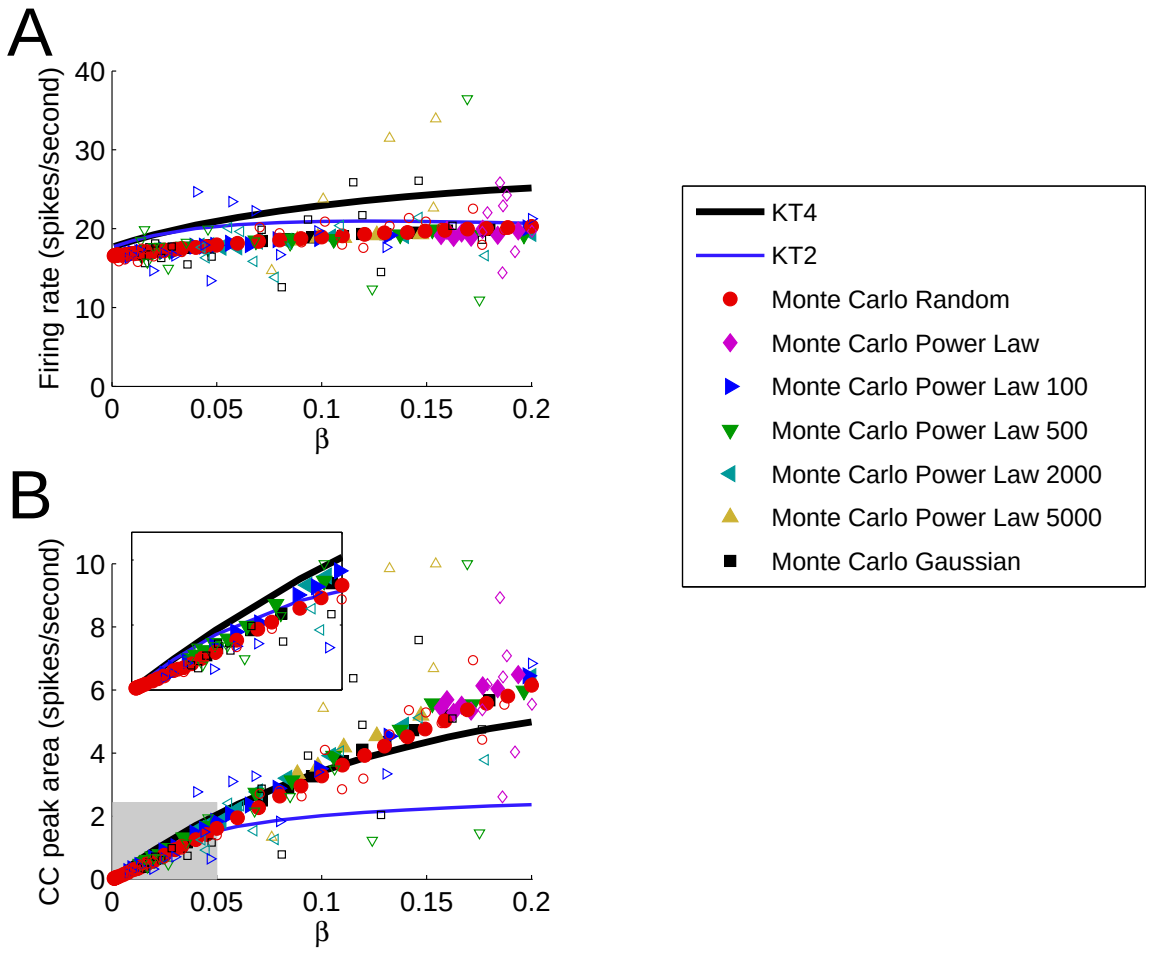


Figure 3.5: Demonstration of the performance of KT2 and KT4 for steady state values of average firing rate r_{ave} and cross-correlation peak area C_{peak} in layer 10 for $0 < \beta \leq 0.2$. Both KT2 and KT4 show the effect of the fraction of shared connections β on the steady-state average firing rate r_{ave} in layer 10, though KT2 shows the behavior of saturation. The increasing pattern of the steady-state cross-correlation peak area C_{peak} with β can be captured by both KT2 and KT4. However, only KT4 makes good approximation of the steady-state correlation magnitude C_{peak} in layer 10 for networks with large β.

performance of KT1 (bottom panel of Figure 2.10(a)). Instead, KT2 captures less percentage of the Monte Carlo which is similar to the performance of KT1, though KT2 matches perfectly the Monte Carlo for the average firing rate $r_{\text{ave}}(t)$.

The better performance of KT4 under high levels of correlation can be further demonstrated by the steady-state results. The steady-state values of the average firing rate $r_{\text{ave}}(t)$ and the cross-correlation peak area $C_{\text{peak}}(t)$ for all layers of the four tested networks with $\beta = 0.01, 0.05, 0.1$ and 0.2 are shown in Figure 3.4. As compared with the performance of KT1 shown in Figure 2.11, it shows that KT4 makes a better approximation of the average firing rate $r_{\text{ave}}(t)$ and the cross-correlation peak area $C_{\text{peak}}(t)$ than KT1 for the network with $\beta = 0.2$. Furthermore, KT4 shows the build-up of correlation magnitude with layer while KT1 shows saturation by layer 6. Similarly, the build-up of steady-state correlation magnitude in layer 10 with β can be captured by KT4 which is shown in Figure 3.5**B**, whereas KT1 also shows saturation at $\beta = 0.05$ (Figure 2.13**(b)**). Besides, KT4 can capture the behavior of the dependence of the steady-state average firing rate in layer 10 on the fraction of shared input β (Figure 3.5**A**), whereas KT1 does not show this dependence in Figure 2.13**(a)**.

3.6 Discussion of improved approach

We have demonstrated that our improved kinetic theory implementation can work well under high levels of correlation if we make appropriate truncation approximations, though it may result in the overestimation of the average firing rate $r_{\text{ave}}(t)$ and the cross-correlation peak area $C_{\text{peak}}(t)$. The overestimation can be expected since we approximate delayed correlation as instantaneous. Furthermore, our improved kinetic theory implementation can capture the effect of the fraction of shared input parameter

β on the steady-state average firing rate (Figure 3.5**A**) which was not captured by the initial approach (Figure 2.13**A**).

Theoretically, the results of the improved kinetic theory implementation should depend on the truncation method used to truncate the input to each population. We chose a particular method to truncate the input based on an ansatz from a diffusion approximation. We haven't, however, shown that this is the best method. It means that there may be other methods that work better. Moreover, even given a particular truncation method, one must determine at what level to truncate. Our results show that low levels work fine for low correlations, but that one may need higher levels when correlation becomes large. We plan to explore methods for determining appropriate truncation levels based on the amount of correlation. (i.e., size of $\tilde{r}_{\text{syn}}$ compared to r_{ave}).

The basic assumption of our improved kinetic theory approach is that the firing activity of any group of identical neurons can be described by a multivariate Poisson process. That is, we completely ignore the effect of the structure of delayed correlation. This may result in the overestimation of the average firing rate $r_{\text{ave}}(t)$ since the effect of multiple correlated inputs with zero delays on the firing rate of one neuron is stronger than multiple correlated inputs with non-zero delays. Furthermore, ignoring the structure of delayed correlation makes the improved kinetic theory approach unable to capture the cross-correlation structure and thus may affect the accuracy of estimation for the cross-correlation peak area $C_{\text{peak}}(t)$. Though one can introduce two more variables (the times since neuron 1 and neuron 2 received their last inputs) in the population density to include cross-correlation at non-zero delays, it is challenging to make predictions about high dimensional correlation structures at non-zero delays when one wants to include higher order correlation with non-zero delays in the input of each population. We plan to explore methods to diminish the effect of approximat-

ing delayed correlation as instantaneous on the overestimation of the average firing rate and cross-correlation peak area by using a diffusion approximation.

Even if the firing activity of any group of identical neurons is a multivariate Poisson process, the predictions about various firing rates should depend on the inference method. Up to now, we only implement a particular method (conditional independence method) to infer various firing rates constrained by lower order statistics $r_{\text{ave}}(t)$ and $\tilde{r}_{\text{syn}}(t)$. We have not compared the results to other possible inference methods. Furthermore, our inference method is based on the assumption that higher order structure is not important. If higher order structure is important then we will obtain bad results by our lower order inference method. Therefore, our improved kinetic theory approach might not always give good results.

Chapter 4

Conclusions and future research direction

We have developed a kinetic theory approach that captures the build-up of correlations in feed-forward neural networks. The method works even for high correlation if we make an appropriate truncation approximation. We demonstrated that although approximating delayed correlation of any order as instantaneous resulted in an overestimate of the firing rate and the magnitude of output cross-correlation, it accurately represented how the firing rate and correlations increased with respect to layer and with respect to fraction of shared inputs in the connectivity structure.

4.1 Sufficiency of second order connectivity

Our goal is to use a second order kinetic theory description to describe the behavior of neuronal networks. We base this goal on the assumption that describing neuronal activity via second order statistics will be sufficient to explain much of the behavior of the networks (c.f., Schneidman et al, 2006; Shlens et al, 2006; Tang et al, 2008; Yu

et al, 2008). In order to derive equations by which to couple the populations densities of the kinetic theory, we distilled the network connectivity patterns between two populations into first and second order statistics. Our hypothesis was that these first and second order connectivity statistics would be sufficient to specify the resulting first and second order statistics of the neuron activity. It is not obvious that such a hypothesis should be true, and we expect that higher order statistics will have some effect. For example, the second order connectivity statistic β does have a small effect on the first order output statistic r_{ave}, as shown by the slight increase of the filled dots with β in figure 2.13A. The fact that we demonstrated that the cross-correlation peak area is primarily determined by β (for fixed first order statistic W and fixed input rates) is encouraging support of the notion that our kinetic theory approach may be able to capture the fundamental mechanisms underlying the emergence and propagation of correlations.

One caveat should be stressed. In all the Monte Carlo networks we used to generate figure 2.13, we varied only the *outgoing* degree distribution of the nodes. We did not, for example, create networks with power law incoming degree distributions. Especially since we do not include inhibition, we could not include neurons that had 100 times more inputs than average. If we did, those neurons would have fired at unreasonable rates. Hence, in all networks, we always assigned postsynaptic targets randomly.

4.2 Related kinetic theory and correlation analyses

Our kinetic theory approach is based on the population density formulation of interacting neuronal populations introduced by Knight, Sirovich, and colleagues (Knight et al, 1996; Omurtag et al, 2000b; Sirovich et al, 2000; Omurtag et al, 2000a; Knight, 2000; Casti et al, 2002; Sirovich et al, 2006) that was further developed by Tranchina and colleagues (Nykamp and Tranchina, 2000, 2001; Haskell et al, 2001; Apfaltrer et al, 2006). Although there are many examples of using kinetic theory to model neuronal networks (Abbott and van Vreeswijk, 1993; Barna et al, 1998; Brunel and Hakim, 1999; Gerstner, 2000; Câteau and Fukai, 2001; Hohn and Burkitt, 2001; Mattia and Del Giudice, 2002; Moreno et al, 2002; Meffin et al, 2004; Câteau and Reyes, 2006; Doiron et al, 2006; Cai et al, 2006; Huertas and Smith, 2006), most approaches do not explicitly represent correlations among neurons but instead assume each neuron in a population is an independent sample from the distribution.

A notable exception is upcoming work by de la Rocha et al (20008) where they also develop a kinetic theory based on a pair of neurons with correlated input. In their case, they assume the inputs are Gaussian white noise and are even able to derive a analytical formula for the output cross-correlation. Their goal is also to study how correlated activity propagates over multiple layers of a network.

There have been many studies characterizing how correlated inputs to uncoupled neurons lead to correlated outputs (Dorn and Ringach, 2003; Galán et al, 2006; Svirskis and Hounsgaard, 2003; Moreno-Bote and Parga, 2006; Binder and Powers, 2001; Shadlen and Newsome, 2001). Recent results have shown how under certain circumstances, the amount of input correlation that is transferred to output correlation depends primarily on the firing rate (de la Rocha et al, 2007), and one can further

characterize this relationship between input and output correlation (Shea-Brown et al, 2008). One additional challenge we have faced in our kinetic theory implementation is how to transform the output correlation of a presynaptic population into the input correlation of its postsynaptic population.

Our application of kinetic theory to correlations in feed-forward networks was motivated by experimental observations of the emergence of sustained synchrony in such networks (Reyes, 2003) and applications of kinetic theory to understand their properties (Câteau and Reyes, 2006; Doiron et al, 2006). The authors discovered typical kinetic theory implementations miss the build-up of sustained synchrony. Although an ad hoc finite-size correction (Brunel and Hakim, 1999; Mattia and Del Giudice, 2002; Hohn and Burkitt, 2001) can restore elements of this synchrony (Doiron et al, 2006), it does not explicitly model the correlations that underlie the synchrony.

4.3 Including autocorrelations

In our kinetic theory formulation, we assumed the input to each neuron was a modulated Poisson process, completely ignoring any autocorrelations in the input. Such autocorrelation in the firing times of neurons are clearly present even in our simplified integrate-and-fire model, as a neuron needs time to integrate up threshold after firing. Moreover, since it is well-known that the output of neurons is not Poisson (Stevens and Zador, 1998; Shinomoto et al, 2003; Salinas and Sejnowski, 2002), it is clear that our Poisson assumption is unrealistic. We suspect that such a Poisson assumption has decreased the accuracy of our kinetic theory approach.

Recent kinetic theory approaches have included autocorrelations of the input using colored white noise (Câteau and Reyes, 2006; Doiron et al, 2006) or a renewal process description (Ly and Tranchina, 2009). We could incorporate such autocorrelations

along with the cross-correlations. Including such autocorrelations would require the addition of two more variables to our population density (for example, the times since neuron 1 and neuron 2 received their last inputs). Those same variables could also be used to include cross-correlations at non-zero delays, leading to a more complete description of the second order statistics in the neuronal activity in terms of correlated renewal processes. Hence, this extension may reduce the overestimate of correlation that we currently observe from our method of collapsing delayed correlation to make it instantaneous. Unfortunately, simulating the evolution of a four-dimensional population density function would be computationally expensive. To make their numerical solution more tractable, we would need to search for techniques to reduce the dimension, searching along the lines, for example, of a rescaling technique to used approximate a colored noise process by a white noise process (Moreno et al, 2002).

4.4 Fast numerical methods

The focus of this work has been to explore how well one could capture the emergence and propagation of correlation with a kinetic theory description that represents some of the second order statistics of neuronal activity and connectivity patterns. At this point, we have not developed a fast numerical method to quickly solve the equations (2.2). Fortunately, a wide range of techniques have been developed for efficient numerical solution of such equations.

One promising option to speed up the computations is recently developed dynamic basis set techniques for efficient simulation of this class of problems (Knight, 2000; Mattia and Del Giudice, 2002). This method relies on the observation that the state of a system such as $\rho(v_1, v_2, t)$ is typically concentrated in a relatively small subspace. One constructs a dynamic basis set of eigenfunctions and can reduce computational

complexity by tracking only a small number of eigenmodes. Such an approach has been successfully used with two-dimensional integrate-and-fire models (Apfaltrer et al, 2006), and may be a good method to speed up the higher-dimensional densities involving correlations. Operator splitting is another technique to speed up simulations (Apfaltrer et al, 2006) that could be employed with our model.

Another way to speed up the solution of (2.2) is to turn the integro-differential equation into a diffusion equation under the assumption that the jump size A is small. The partial differential equation can be solved more quickly becomes one obtains a sparser matrix when discretizing the derivatives with respect voltage. However, the diffusion approximation loses accuracy as the jump size becomes large. For this reason, the diffusion approximation may be highly inaccurate when the correlation is high and double-sized (or larger) voltage jumps become common.

4.5 A tool to analyze network behavior and connectivity

Our goal for developing a second order kinetic theory implementation is a tool with which to investigate the consequences of network structure on network behavior. We are aiming for a method that one can use to distill the connectivity down to its key features and to study how these features influence the network behavior. Although the development of the tool is still in its early stages (we haven't yet addressed recurrent connections within a population), it has shown promise that it can capture basic aspects of the behavior of second order statistics in feed-forward networks. If one could parameterize network connectivity by second order statistics and use the kinetic theory tool to explore their consequences on second order statistics of neu-

ronal activity, this simplified description may help uncover key network parameters underlying network behaviors.

Already at these initial stages, we have gained some insight into the network parameters that underlie the build up of correlations. Previous studies (Doiron et al, 2006) have concluded that network size may be the critical factor in determining this behavior. Our kinetic theory analysis suggested that it was the fraction of shared input parameter rather than network size that played the key role. Simulations of networks with different higher order connectivity statistics provided convincing evidence supporting the kinetic theory prediction.

To implement the kinetic theory analysis, one needs methods to determine the second order connectivity patterns at the level of a population coarse-graining. One would like to know what classes of neurons project onto a neuron of a given class (our first order statistic W^1) and what classes of neurons project common input connections onto pairs of neurons of a given class (our second order statistic $W^2 = \beta W^1$). One of us has recently developed a connectivity analysis (Nykamp, 2005, 2007b,a) that contains ambiguity that could be at the same level as the population grouping. Although this connectivity analysis is designed to distinguish common input (even originating from unmeasured neurons) from causal connections among a set of measured neurons, more work needs to be done to identify the classes of the common input neurons. With that extension, the connectivity analysis may be one way to determine the second order statistics of the connectivity needed for the kinetic theory analysis. A tool to estimate second order connectivity and a tool to explore the consequences of second order connectivity could prove a powerful combination for probing basic properties of neural circuits and exploring the link between connectivity patterns and network behavior.

Bibliography

Abbott LF, van Vreeswijk C (1993) Asynchronous states in networks of pulse-coupled oscillators. Phys Rev E 48:1483–1490

Apfaltrer F, Ly C, Tranchina D (2006) Population density methods for stochastic neurons with realistic synaptic kinetics: firing rate dynamics and fast computational methods. Network: Comput Neural Syst 17:373–418

Barna G, Gröbler T, Érdi P (1998) Statistical model of the hippocampal CA3 region - II. the population framework: model of rhythmic activity in the CA3 slice. Biol Cybern 79:309–321

Binder MD, Powers RK (2001) Relationship between simulated common synaptic input and discharge synchrony in cat spinal motoneurons. J Neurophysiol 86:2266–2275

Brunel N, Hakim V (1999)) Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. Neural Comp 11(7):1621–1671

Cai D, Tao L, Rangan A, McLaughlin D (2006) Kinetic theory for neuronal network dynamics. Comm Math Sci 4:97–127

Casti AR, Omurtag A, Sornborger A, Kaplan E, Knight B, Victor J, Sirovich L (2002) A population study of integrate-and-fire-or-burst neurons. Neural Comp 14:957–86

Câteau H, Fukai F (2001) Fokker-Planck approach to the pulse packet propagation in synfire chain. Neural Networks 14:675–685

Câteau H, Reyes AD (2006) Relation between single neuron and population spiking statistics and effects on network activity. Phys Rev Let 96:058,101

Diesmann M, Gewaltig MO, Aertsen A (1999) Stable propagation of synchronous spiking in cortical neural networks. Nature 402:529–533

Doiron B, Rinzel J, Reyes A (2006) Stochastic synchronization in finite size spiking networks. Phys Rev E 74:030,903

Dorn JD, Ringach DL (2003) Estimating membrane voltage correlations from extracellular spike trains. J Neurophysiol 89:2271–2278

Galán RF, Fourcaud-Trocmé N, Ermentrout GB, Urban NN (2006) Correlation-induced synchronization of oscillations in olfactory bulb neurons. J Neurosci 26:3646–3655

Gardiner CW (2004) Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, 3rd edn. Springer-Verlag, New York

Gerstner W (2000) Population dynamics of spiking neurons: Fast transients, asynchronous states, and locking. Neural Comp 12:43–89

Hasegawa H (2003) Dynamical mean-field theory of spiking neuron ensembles: Response to a single spike with independent noises. Phys Rev E 67:041,903

Haskell E, Nykamp DQ, Tranchina D (2001) Population density methods for large-scale modeling of neuronal networks with realistic synaptic kinetics: Cutting the dimension down to size. Network: Comput Neural Syst 12(2):141–174

Hohn N, Burkitt AN (2001) Shot noise in the leaky integrate-and-fire neuron. Phys Rev E 63:031,902

Huertas MA, Smith G (2006) A multivariate population density model of the dlgn/pgn relay. J Comp Neurosci 21:171–89

Ichimaru S (1973) Basic principles of plasma physics: A statistical approach. Benjamin, New York

Jaynes ET (1957) Information theory and statistical mechanics. Phys Rev 106:62–79

Knight BW (2000) Dynamics of encoding in neuron populations: Some general mathematical features. Neural Comp 12:473–518

Knight BW, Manin D, Sirovich L (1996) Dynamical models of interacting neuron populations. In: Gerf EC (ed) Symposium on Robotics and Cybernetics: Computational Engineering in Systems Applications, Cite Scientifique, Lille, France

Litvak V, Sompolinsky H, Segev I, Abeles M (2003) On the transmission of rate code in long feedforward networks with excitatory-inhibitory balance. J Neurosci 23:3006–3015

Liu CY, Nykamp DQ (2009) A kinetic theory approach to capturing interneuronal correlation: the feed-forward case. J Comp Neurosci 26:339–368

Ly C, Tranchina D (2009) Spike train statistics and dynamics with synaptic input from any renewal process: A population density approach. Neural Comput 21:360–396

Masuda N, Aihara K (2002) Bridging rate coding and temporal spike coding by effect of noise. Phys Rev Let 88:248,101

Mattia M, Del Giudice P (2002) Population dynamics of interacting spiking neurons. Phys Rev E 66:051,917

Mcfadden JA (1965) The entropy of a point process. Journal of the Society for Industrial and Applied Mathematics 13:988–994

Meffin H, Burkitt AN, Grayden DB (2004) An analytical model for the "large, fluctuating synaptic conductance state" typical of neocortical neurons in vivo. J Comp Neurosci 16:159–175

Moreno R, de la Rocha J, Renart A, Parga N (2002) Response of spiking neurons to correlated inputs. Phys Rev Let 89:288,101

Moreno-Bote R, Parga N (2006) Auto- and crosscorrelograms for the spike response of leaky integrate-and-fire neurons with slow synapses. Phys Rev Let 96:028,101

Nicholson D (1992) Introduction to plasma theory. Krieger, Malabar, Florida

Nirenberg S, Victor J (2007) Analyzing the activity of large populations of neurons: how tractable is the problem? Curr Opin Neurobiol 17:397–400

Nykamp DQ (2005) Revealing pairwise coupling in linear-nonlinear networks. SIAM J Appl Math 65:2005–2032

Nykamp DQ (2007a) Exploiting history-dependent effects to infer network connectivity. SIAM J Appl Math 68:354–391

Nykamp DQ (2007b) A mathematical framework for inferring connectivity in probabilistic neuronal networks. Math Biosci 205:204–251

Nykamp DQ, Tranchina D (2000) A population density approach that facilitates large-scale modeling of neural networks: Analysis and an application to orientation tuning. J Comp Neurosci 8:19–50

Nykamp DQ, Tranchina D (2001) A population density approach that facilitates large-scale modeling of neural networks: Extension to slow inhibitory synapses. Neural Comp 13:511–546

Omurtag A, Kaplan E, Knight B, Sirovich L (2000a) A population approach to cortical dynamics with an application to orientation tuning. Network: Comput Neural Syst 11:247–260

Omurtag A, Knight BW, Sirovich L (2000b) On the simulation of large populations of neurons. J Comp Neurosci 8:51–63

Reyes AD (2003) Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro. Nat Neurosci 6:593–599

de la Rocha J, Moreno-Bote R, Câteau H (20008) Propagation of temporally correlated spike trains: a Fokker-Planck analysis, in preparation

de la Rocha J, Doiron B, Shea-Brown E, Josić K, Reyes A (2007) Correlation between neural spike trains increases with firing rate. Nature 448:802–806

van Rossum MCW, Turrigiano GG, Nelson SB (2002) Fast propagation of firing rates through layered networks of noisy neurons. J Neurosci 22:1956–1966

Saad Y, Schultz MH (1986) Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 7:856–869

Salinas E, Sejnowski T (2002) Integrate-and-fire neurons driven by correlated stochastic input. Neural Comput 14:2111–2155

Schneidman E, II MJB, Segev R, Bialek W (2006) Weak pairwise correlations imply strongly correlated network states in a neural population. Nature 440:1007–1012

Shadlen MN, Newsome WT (2001) Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. J Neurophysiol 86:1916–1936

Shea-Brown E, Josić K, de la Rocha J, Doiron B (2008) Correlation and synchrony transfer in integrate-and-fire neurons: Basic properties and consequences for coding. Phys Rev Let 100:108,102

Shinomoto S, Shima K, Tanji J (2003) Differences in spiking patterns among cortical neurons. Neural Comput 15:2823–2842

Shlens J, Field GD, Gauthier JL, Grivich MI, Petrusca D, a Sher, Litke AM, Chichilnisky EJ (2006) The structure of multi-neuron firing patterns in primate retina. J Neurosci 26:8254–8266

Sirovich L, Knight B, Omurtag A (2000) Dynamics of neuronal populations: The equilibrium solution. SIAM J Appl Math 60:2009–2028

Sirovich L, Omurtag A, Lubliner K (2006) Dynamics of neural populations: stability and synchrony. Network: Comput Neural Syst 17:3–29

Stevens C, Zador A (1998) Input synchrony and the irregular firing of cortical neurons. Nat Neurosci 1:210–207

Svirskis G, Hounsgaard J (2003) Influence of membrane properties on spike synchronization in neurons: theory and experiments. Network: Comput Neural Syst 14:747–763

Tang A, Jackson D, Hobbs J, Chen W, Smith JL, Patel H, Prieto A, Petrusca D, Grivich MI, Sher A, Hottowy P, Dabrowski W, Litke AM, Beggs JM (2008) A

maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro. J Neurosci 28:505–518

Wang S, Wang W, Liu F (2006) Propagation of firing rate in a feed-forward neuronal network. Phys Rev Let 96:018,103

Yu S, Huang D, Singer W, Nikolic D (2008) A small world of neuronal synchrony. Cereb Cortex 18:2891–2901

Appendix A

Method for solving the kinetic theory equations

The first step in solving the kinetic theory equations (2.2) is to rewrite them in conservative form, i.e., divergence form $\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J}$, where $\mathbf{J}$ is a flux density. The terms describing the advection due to the leak current are already the divergence of a flux density. We need to rewrite the integrals corresponding to the voltage jumps in response to excitatory input. Recall that $F_A(x)$ is the complementary cumulative distribution function and $f_A(x)$ the probability density of the random jump size A, so that $\frac{\partial F_A}{\partial x} = -f_A(x)$. Applying this identity and the fundamental theorem of calculus,

we obtain the following equalities, we rewrite the terms due to independent input as

$$\begin{aligned}
&\int_{v_{\text{reset}}}^{v_1} f_A(v_1-\theta_1)\rho(\theta_1,v_2,t)d\theta_1 - \rho(v_1,v_2,t)\\
&\qquad = -\frac{\partial}{\partial v_1}\int_{v_{\text{reset}}}^{v_1} F_A(v_1-\theta_1)\rho(\theta_1,v_2,t)d\theta_1,\\
&\int_{v_{\text{reset}}}^{v_2} f_A(v_2-\theta_1)\rho(v_1,\theta_2,t)d\theta_2 - \rho(v_1,v_2,t)\\
&\qquad = -\frac{\partial}{\partial v_2}\int_{v_{\text{reset}}}^{v_2} F_A(v_2-\theta_2)\rho(v_1,\theta_2,t)d\theta_2,
\end{aligned} \tag{A.1}$$

The integrals involving F_A are probability flux densities analogous to though defined in (Nykamp and Tranchina, 2000).

The jumps due to synchronous input are a little more complicated. There are many ways to write them in conservative form. Since the evolution of ρ is symmetric in v_1 and v_2, we kept the equation symmetric by dividing the diagonal jumps into a symmetric pair of horizontal and vertical jumps.

$$\begin{aligned}
&\int_{v_{\text{reset}}}^{v_1}\int_{v_{\text{reset}}}^{v_2} f_A(v_1-\theta_1)f_A(v_2-\theta_2)\rho(\theta_1,\theta_2,t)d\theta_1 d\theta_2\\
&\qquad\qquad - \rho(v_1,v_2,t)\\
&= -\frac{1}{2}\frac{\partial}{\partial v_1}\left(\int_{v_{\text{reset}}}^{v_1} F_A(v_1-\theta_1)\rho(\theta_1,v_2,t)d\theta_1\right.\\
&+ \left.\int_{v_{\text{reset}}}^{v_1}\int_{v_{\text{reset}}}^{v_2} F_A(v_1-\theta_1)f_A(v_2-\theta_2)\rho(\theta_1,\theta_2,t)d\theta_2 d\theta_1\right)\\
&- \frac{1}{2}\frac{\partial}{\partial v_2}\left(\int_{v_{\text{reset}}}^{v_2} F_A(v_2-\theta_2)\rho(v_1,\theta_2,t)d\theta_2\right.\\
&+ \left.\int_{v_{\text{reset}}}^{v_2}\int_{v_{\text{reset}}}^{v_1} F_A(v_2-\theta_2)f_A(v_1-\theta_1)\rho(\theta_1,\theta_2,t)d\theta_1 d\theta_2\right).
\end{aligned} \tag{A.2}$$

Although these integrals don't correspond to physical horizontal and vertical jumping

of the voltage, we view their formulation simply as an intermediate step to developing a numerical scheme to solve the original equation (2.2).

As a result, we obtain the following integro-differential equation:

$$\begin{aligned}\frac{\partial \rho}{\partial t}(v_1, v_2, t) &= -\nabla \cdot \mathbf{J}(v_1, v_2, t) \\ &+ \delta(v_1 - v_{\text{reset}}) J_{\text{reset},1}(v_2, t) + \delta(v_2 - v_{\text{reset}}) J_{\text{reset},2}(v_1, t) \\ &+ \delta(v_1 - v_{\text{reset}}) \delta(v_2 - v_{\text{reset}}) J_{\text{reset},3}(t),\end{aligned} \tag{A.3}$$

where

$$\begin{aligned}&\mathbf{J}(v_1, v_2, t) = \mathbf{J}_{\text{leak}}(v_1, v_2, t) + \mathbf{J}_{\text{ind}}(v_1, v_2, t) + \mathbf{J}_{\text{syn}}(v_1, v_2, t), \\ &\mathbf{J}_{\text{leak}} = (J^1_{\text{leak}}, J^2_{\text{leak}}), \mathbf{J}_{\text{ind}} = (J^1_{\text{ind}}, J^2_{\text{ind}}), \mathbf{J}_{\text{syn}} = (J^1_{\text{syn}}, J^2_{\text{syn}}), \\ &J^1_{\text{leak}}(v_1, v_2, t) = -\frac{v_1 - Er}{\tau} \rho(v_1, v_2, t), \\ &J^1_{\text{ind}}(v_1, v_2, t) = \nu_{\text{ind}}(t) \int_{v_{\text{reset}}}^{v_1} F_A(v_1 - \theta_1) \rho(\theta_1, v_2, t) d\theta_1, \\ &J^1_{\text{syn}}(v_1, v_2, t) = \frac{1}{2} \nu_{\text{syn}}(t) \Bigg(\int_{v_{\text{reset}}}^{v_1} F_A(v_1 - \theta_1) \rho(\theta_1, v_2, t) d\theta_1 \\ &\quad + \int_{v_{\text{reset}}}^{v_1} \int_{v_{\text{reset}}}^{v_2} F_A(v_1 - \theta_1) f_A(v_2 - \theta_2) \rho(\theta_1, \theta_2, t) d\theta_2 d\theta_1 \Bigg),\end{aligned} \tag{A.4}$$

All second components are analogous by symmetry. The reset terms are defined in (2.3).

We do not have local conservative equality between the jump flux density across the threshold ($J^1_{\text{ind}}(v_{\text{th}}, v_2, t) + J^1_{\text{syn}}(v_{\text{th}}, v_2, t)$) and the reset $J_{\text{reset},1}(v_2, t)$. Due to our definition of the flux density $J^1_{\text{syn}}(v_{\text{th}}, v_2, t)$, it includes the effect of synchronous inputs where the voltage of neuron 2 jumps from v_2 to higher voltages (and hence appears in the reset term $J_{\text{reset},1}(v_2, t)$ at those higher voltage or even in $J_{\text{reset},3}(t)$ if neuron 2 also crossed threshold). Nonetheless, one can verify that the system is globally

conservative because the total flux across threshold equals the total reset:

$$\begin{aligned}\int_{v_{\text{reset}}}^{v_{\text{th}}} (J^1_{\text{ind}}(v_{\text{th}}, v_2, t) + J^1_{\text{syn}}(v_{\text{th}}, v_2, t))dv_2 \\ + \int_{v_{\text{reset}}}^{v_{\text{th}}} (J^2_{\text{ind}}(v_1, v_{\text{th}}, t) + J^2_{\text{syn}}(v_1, v_{\text{th}}, t))dv_1 \\ = \int_{v_{\text{reset}}}^{v_{\text{th}}} J_{\text{reset},1}(v_2, t)dv_2 + \int_{v_{\text{reset}}}^{v_{\text{th}}} J_{\text{reset},2}(v_1, t)dv_1 \\ + J_{\text{reset},3}(t) \end{aligned} \tag{A.5}$$

To numerically solve the equations, we used the finite volume approach. We discretized the domain $[0, v_{\text{th}}] \times [0, v_{\text{th}}]$ in (v_1, v_2) space into squares of size $\Delta v \times \Delta v$ with $\Delta v = 0.0125$. We let $v_{1,j} = v_{2,j} = (j - 1/2)\Delta v$ so that the point $(v_{1,j}, v_{2,k})$ is the center of square (j, k). Each point $\rho_{j,k}(t)$ represented the integral of $\rho(v_1, v_2, t)$ over the square (j, k). The divergence $\nabla \cdot \mathbf{J}$ at the center of each square was approximated by differences in the integral of the flux along the boundary of the square, and we approximated the resulting integrals along each line segment with the midpoint rule. We linearly interpolated $\rho_{j,k}$ to estimate its value along the boundary for J_{leak}. To estimate the value of the integrals for J_{ind} and J_{syn}, we used Simpson's rule.

We reduced the dimension of the system of equations by exploiting the symmetry $\rho_{j,k} = \rho_{k,j}$. We discretized in time using the trapezoid method with $\Delta t = 0.5$ ms. At each time step, we solved the system of linear equations using GMRES (Saad and Schultz, 1986). In this way, we used a numerical method that is second order accurate in both time and space.

Since we set $v_{\text{reset}} < E_r$, the advection flux $\mathbf{J}_{\text{leak}}$ is non-zero along locations of voltage reset, preventing the $J_{\text{reset},1}$ and $J_{\text{reset},2}$ delta-function source terms from forming a delta-function in $\rho(v_1, v_2, t)$. However, the double delta-function source due to the reset $J_{\text{reset},3}$ will form a delta-function component of ρ. The flux $J_{\text{reset},3}$ represents a

finite probability per unit time that both neurons reset to $(v_{\text{reset}}, v_{\text{reset}})$. At that point, the voltage pair will advect along the line $\{(v_1, v_2)|v_1 = v_2, 0 < v_1 < E_r\}$ until either neuron receives an input. Hence, there will be finite probability that the voltages of a neuron pair reside along this line. The delta-function in ρ along this line will prevent second order convergence of the above numerical method if we apply it directly to equation (A.3). To ensure convergence, we divide ρ into two components, a smooth component ρ_s and a delta function component along the diagonal with weight function ρ_δ:

$$\rho(v_1, v_2, t) = \rho_s(v_1, v_2, t) + \delta(v_1 - v_2)\rho_\delta(v_1) \tag{A.6}$$

Plugging this expression into the original evolution equation (A.3) and then grouping the delta-function terms will result in coupled evolution equations for ρ_s and ρ_δ. Since ρ_s and ρ_δ are smooth, we can solve the resulting equations using our numerical scheme and achieve second order accuracy.

Appendix B

Calculating connectivity statistics for different network classes

We calculate the expected number of inputs W and the fraction of shared inputs β for networks with arbitrary outgoing degree distributions and incoming connections determined randomly. We examine the connectivity from a presynaptic population 1 with N_1 neurons onto a postsynaptic population 2 with N_2 neurons.

Let $\hat{W}_{ij} = 1$ if there is a connection from neuron j in population 1 onto neuron i in population 2; otherwise, let $\hat{W}_{ij} = 0$. Let $d_j = \sum_{i=1}^{N_2} \hat{W}_{ij}$ be outgoing degree of neuron j in population 1, i.e., the number of connections from neuron j onto all neurons in population 2. Let the function $f(k)$ be the outgoing degree distribution so that

$$\Pr(d_j = k) = f(k)$$

for $k = 1, 2, \ldots, N_2$. The expected total number of connections (out of $N_1 N_2$ possible) is $N_1 \sum_{k=1}^{N_2} k f(k)$ so that the expected number of connections onto any neuron in

population 2 is

$$W = \frac{N_1}{N_2} \sum_{k=1}^{N_2} k f(k). \tag{B.1}$$

If $f(k)$ was given by a one-parameter family of distributions, prescribing W would determine the particular distribution as a function of population sizes N_1 and N_2.

Neuron j in population 1 has d_j connections onto neurons in population 2, which we assume are assigned randomly to neurons in population 2. Then, conditioned on this value of d_j, the probability of a connection from neuron j onto any given pair (indexed by i_1 and i_2) of neurons in population 2 is

$$\Pr(\hat{W}_{i_1 j} = 1 \,\&\, \hat{W}_{i_2 j} = 1 \,|\, d_j) = \frac{\binom{N_2-2}{d_j-2}}{\binom{N_2}{d_j}} = \frac{d_j(d_j-1)}{N_2(N_2-1)}$$

Multiplying by the probability distribution of d_j (i.e., the outgoing degree distribution) and summing over all possible values of d_j, we determine that the probability a given neuron j in population 1 is connected to a given pair of neurons in population 2 is

$$\sum_{k=1}^{N_2} \Pr(\hat{W}_{i_1 j} = 1 \,\&\, \hat{W}_{i_2 j} = 1 \,|\, d_j = k) \Pr(d_j = k)$$

$$= \sum_{k=1}^{N_2} \frac{k(k-1)}{N_2(N_2-1)} f(k).$$

To calculate the total expected number of shared inputs from all N_1 neurons, we simply need to multiply by N_1:

$$\sum_{k=1}^{N_2} \frac{N_1 k(k-1)}{N_2(N_2-1)} f(k).$$

Dividing by W (B.1) gives the fraction of shared input parameter β in terms of the degree distribution

$$\beta = \frac{\sum_{k=1}^{N_2} k(k-1)f(k)}{(N_2-1)\sum_{k=1}^{N_2} kf(k)}. \tag{B.2}$$

For a random network, the degree distribution is a binomial distribution

$$f(k) = \binom{N_2}{k} p^k (1-p)^{N-k}$$

so that

$$W = N_1 p \qquad \text{and} \qquad \beta = p, \tag{B.3}$$

agreeing with the results from section 2.7.3. If the degree distribution is given by a power law with maximum degree $d_{\max} \leq N_2$

$$f(k) = \begin{cases} k^{-\gamma} / \sum_{n=1}^{d_{\max}} n^{-\gamma} & \text{if } k \leq d_{\max}, \\ 0 & \text{otherwise,} \end{cases}$$

the expressions for W and β become

$$W = \frac{N_1 \sum_{k=1}^{d_{\max}} k^{1-\gamma}}{N_2 \sum_{k=1}^{d_{\max}} k^{-\gamma}} \qquad \text{and} \qquad \beta = \frac{\sum_{k=1}^{d_{\max}} (k-1)k^{1-\gamma}}{(N_2-1)\sum_{k=1}^{d_{\max}} k^{1-\gamma}}. \tag{B.4}$$

If the degree distribution is given by a Gaussian

$$f(k) = \frac{e^{-k^2/2\sigma^2}}{\sum_{n=1}^{N_2} e^{-n^2/2\sigma^2}}$$

the expressions for W and β become

$$W = \frac{N_1 \sum_{k=1}^{N_2} k e^{-k^2/2\sigma^2}}{N_2 \sum_{k=1}^{N_2} e^{-k^2/2\sigma^2}} \quad \text{and} \quad \beta = \frac{\sum_{k=1}^{N_2} k(k-1) e^{-k^2/2\sigma^2}}{(N_2 - 1) \sum_{k=1}^{N_2} k e^{-k^2/2\sigma^2}}. \tag{B.5}$$

In our simulations, we set $N_1 = N_2 = N$ and solved the first equation of (B.3), (B.4) or (B.5) for the p, γ or σ, respectively that gave the chosen W. Then,we used the second equation of (B.3), (B.4) or (B.5) to calculate β.