Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Modeling Neuronal Assemblies: Theory and Implementation

Julian Eggert, Jan van Hemmen

2001

Preprint:

This is an accepted article published in Neural Computation. The final authenticated version is available online at: https://doi.org/[DOI not available]

Neural Computation, 13(9):1923–1974, 2001.

Modeling Neuronal Assemblies: Theory and Implementation

J. Eggert*
HONDA R&D Europe (Deutschland) GmbH, Future Technology Research,
Carl-Legien-Straße 30, 63073 Offenbach/Main, Germany

J. L. van Hemmen
Physik Department, Technische Universität München,
85747 Garching bei München, Germany

Abstract

Models that describe qualitatively and quantitatively the activity of entire groups of spiking neurons are getting increasingly important for biologically realistic large-scale network simulations. At the systems and areas modeling level, it is necessary to switch the basic descriptional level from single spiking neurons to neuronal assemblies. In this paper, we present and review work that allows directly a macroscopic description of the assembly activity. We show that such macroscopic models can be used to reproduce in a quantitatively exact manner the joint activity of groups of spike–response or integrate–and–fire neurons. We also show that integral as well as differential equation models of neuronal assemblies can be understood within a single framework, which allows a comparison with the commonly used assembly-averaged graded–response type models. The presented framework thus enables the large-scale neural network modeler to implement networks using computational units beyond the single spiking neuron without loosing much biological accuracy. This paper explains the theoretical background as well as the capabilities and the implementation details of the assembly approach.

1 Introduction

The choice of the right modeling level is a serious challenge for the neural network modeler concerned about large-scale, biologically realistic architectures. While there exists an abundance of detailed single-neuron models, there are only a few models available at the next modeling level of neuronal groups of assemblies [1], which are

*Corresponding author, email: **julian.eggert@hre-ftr.f.rd.honda.co.jp**

[1] Early approaches on assembly dynamics include the work of Wilson & Cowan (1973), an up-to-date analysis is presented by Gerstner (1998), and recent work related to this topic can be found e.g. in Knight

usually described by using a dynamics for their macroscopic variables. The most commonly encountered macroscopic models are of the assembly-averaged graded–response type, but their use is not justified in dynamical regimes away from quasi-stationary activity and their derivation from single neuron dynamics is not exact.

In this paper, we start from quite general assumptions on the dynamics of single spiking neurons and develop a framework that allows for an exact derivation of assembly dynamics. A first central result of the paper is given in Fig. 3, which explains the essence of the assembly dynamics in a graphical way. We then show how the framework can be applied to a well-known and widely studied model of spiking neurons, the spike–response–model (SRM) (for a review, see Gerstner & van Hemmen 1994), which contains in its formulation the integrate–and–fire (I&F) type models as a special case.

The presented work allows to understand some of the models for assembly dynamics available within a common conceptual framework. Many of the known models turn out to be special cases or approximations of the derived assembly dynamics. A few of the results have been derived and presented elsewhere and they are not explained in detail in this work, since our focus is on a common framework that shows the modeler which models and techniques are available and how they are related with each other. Therefore, a second central result of this paper is the diagram of Fig. 11, which summarizes our approach as well as related models and which indicates their interdependencies.

The paper is divided into a series of sections and two appendices. Section 2 deals with the assumptions on which our derivation and the entire framework are based, with special attention on the assembly definition. In section 3, integral equation expressions are derived that serve to model the assembly dynamics. They are presented first in a general form, and then in a specific form which is applicable to assemblies of SRM neurons. In section 5, the integral equation dynamics are converted into a differential equation system, which allows a comparison with other comparable models for assembly dynamics based on differential equations. In section 6, we show how our framework can be used to investigate and understand many well-known properties and existing models of assembly dynamics. In this section, we also show the good quantitative agreement between the derived assembly dynamics and simulations using assemblies of explicitly modeled single neurons. Section 7 summarizes the work and gives an overview of how the different models depend on each other in Fig. 11. Finally, in Apps. A and B, implementation details are summarized that should allow to circumvent the main difficulties of a numerical application of our framework.

2000; Knight, Omurtag, & Sirovich 2000; Nykamp & Tranchina 2000; Omurtag, Knight, & Sirovich 2000; Sirovich, Knight, & Omurtag 2000

2 Assemblies of Spiking Neurons

2.1 Assembly Definition

It is out of question that large, biologically realistic network models of cortical systems cannot be described yet on a single-unit basis because of the sheer amount of required computational resources. But do we have to model single neurons explicitly or is it sufficient to group large numbers of neurons together and describe simply their joint activity? In the primate cortex, there are numerous dendrites running orthogonally to the cortical layer structure, collecting input from many of the layers they cross on their way. This means that the corresponding neurons potentially have access to the information available in all layers of their column. Accordingly, it might make sense to group all neurons of a column into a functional unit that can be described by a single model, neglecting the fine neuronal structure.

Another motivation to group neurons is the experimental observation that cortical neurons of the same type that are located near to each other tend to receive similar inputs. In experiments one often finds that neurons of the same type that are located close to each other are activated simultaneously, or in a correlated fashion. In cortical networks, this may be due to reciprocal connections and common convergent input.

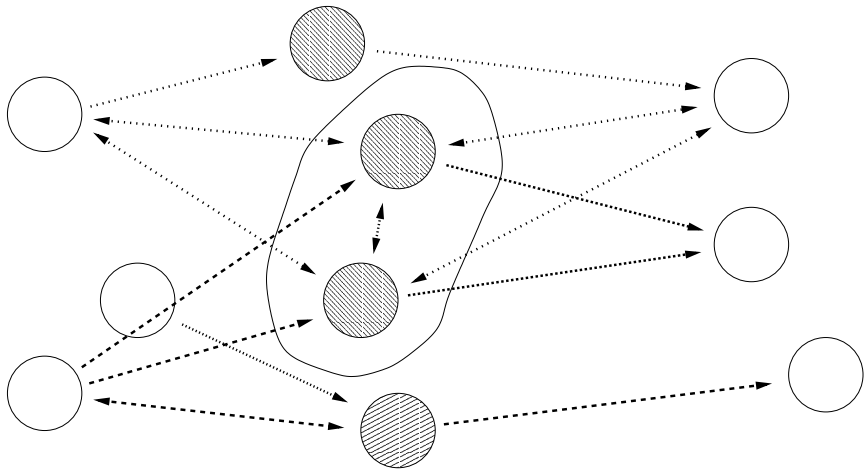

Figure 1: A "pool" or "assembly" of neurons. We assume that pools are the basic computational elements of some cortical networks. Neurons belonging to the same pool or assembly are characterized by having the same input/output connectivity pattern. Furthermore, all neurons of the same pool have the same parameters. This is an idealized definition that is fulfilled in some parts of the cortex to a good approximation. In the figure, different types of neurons and connections are characterized by different textures (white neurons are of any type). According to the assembly definitions, only the two neurons in the oval belong to the same pool. This rather restrictive pool definition is consistent in the sense that it allows, for a network with fixed connectivity, a complete description of the architecture and dynamics in terms of interacting assemblies instead of single neurons.

In modeling studies it therefore seems sensible to consider all neurons of the same

type in a small cortical volume as a computational unit of a neuronal network. We will call this computational unit a neuronal **"pool"** or **"assembly"** (sometimes also called a "neuronal ensemble" or "neuronal clique"). All pool neurons have to be equivalent in the sense that they have the same input/output connection characteristics and, additionally, the same dynamics parameters. This is explained in Fig. 1. All neurons that constitute a pool feel a common synaptic input (because of the identical input connectivity structure), but still, each neuron evolves according to its own internal dynamics (because it is subject to an independent realization of common stochastic noise).

In the visual cortex, we can for example assume that all neurons of the same type located in the same layer and in the same cortical column, and which additionally have a similar stimuli selectiveness, constitute an assembly. In the following, we will assume that, for some parts of the brain, these assemblies constitute the basic building blocks which can be connected to simulate large, biologically realistic neural networks.

2.2 Assumptions on Single-Neuron Dynamics

In this section, we present 3 assumptions which are central to our dynamics derivation in later sections. They are: The concept of spike-trains composed of δ-pulses for describing neuronal activity, the renewal property of spiking neurons and the stochastic process for spike release. In the following, we describe shortly what we mean by these assumptions.

2.2.1 Spike-Trains

Common to many threshold models of spiking neurons is the assumption that the *exact* form of the action potential *does not vary* considerably from spike to spike, so that it can be neglected in network simulations. All that matters for neuronal behavior (e.g., for the neuronal refractory properties and neuron-neuron interactions) are the exact **spike times** t^f, $f \in \mathbb{N}$. This assumption also implicates that the maximal temporal resolution of single neuronal dynamics will be in the order of the width of the action potential (e.g. about 1 ms in typical neurons of the visual cortex of primates).

If we neglect the form of the action potentials, the **spike train** $S(t)$ elicited by a neuron can be characterized by a sequence $\{t^f\}$ of delta-pulses at times t^f,

$$S(t) = \sum_{f=1}^{\infty} \delta(t - t^f) \ , \tag{1}$$

with $t^f \leq t$ and ordered from present to past, so that $t^{f+1} < t^f$. For the calculation of the *synaptic input* to other neurons, it is then sufficient to consider the *incoming* spike trains only, characterized by the set of firing times $\{t^f\}$ from presynaptic neurons. Similarly, for the *refractory behavior* of a neuron, it is sufficient to know *its own* past spike times.

2.2.2 Renewal Hypothesis

The most recent spike time,

$$t^* := t^1 \ , \tag{2}$$

can be special in the sense that in some models it has a leading influence on the neuronal dynamics. This is the case if the refractory properties of a neuron only depend on the time

$$s = t - t^* \tag{3}$$

elapsed since the last action potential. This assumption is also referred to as *short-term refractory memory approximation*, or **renewal hypothesis**, (Tuckwell 1988) for neuronal models. It is often justified to use models of spiking neurons with renewal hypothesis because the last spike can have shunting properties that "reset" the state of a neuron so that the influence of the older spikes gets neglectable. The renewal property of spiking neurons is a requirement for the assembly dynamics derivation in later sections.

2.2.3 Stochastic Spike Release

In biological neuronal systems, the role and origin of noise is still a debated issue. Although a single, isolated neuron seems to be a deterministic device with a fixed membrane potential threshold for spike generation (see e.g. Mainen & Sejnowski 1995), neurons in networks elicit spike trains with a great variability of apparently random temporal patterns. One intrinsic source of noise identified in biological neural networks is the interneuronal synaptic transmission process using chemical transmitters. Another factor that enhances noisy effects may be a delicate balance between excitation and inhibition in neuronal networks, so that neurons are mainly driven very close to their firing threshold, enabling small input fluctuations to trigger action potentials. In any case, the input to neurons often "looks" noisy, making necessary the incorporation of stochasticity into networks.

In the following, we will be dealing with models of spiking neurons that release spikes according to a stochastic process that can be described by a release probability density function λ. We assume that, for neurons with renewal properties, the probability of releasing a spike during the interval $(t, t + \Delta t]$ depends on the internal state of the neuron, which can be quantified in terms of t and t^* because it depends on the time s since its last spike release, and the driving input at time t. The spike release probability of a neuron i is then

$$\text{Prob}\{i \text{ fires in } (t, t + \Delta t]\} = \Delta t \, \lambda(t, t^*) \ . \tag{4}$$

The function $\lambda(t, t^*)$ will play a central role when deriving the assembly dynamics.

2.3 Macroscopic Quantities and Neuronal Density

2.3.1 The Macroscopic Pool Activity $A(t)$

We will denominate pools by bold letters, e.g. $\mathbf{x}$, $\mathbf{y}$, or subindices m, n. Looking at the pool as a whole, we can define a pool activity using the spike trains of the pool

neurons

$$\hat{A}(\mathbf{x},t) = \sum_{j\in\mathbf{x}} S_j(t) \ . \tag{5}$$

For extensively many pool neurons, we introduce a continuous **pool activity** (or **spike density**) $A(\mathbf{x},t)$ by integrating $\hat{A}(\mathbf{x},t)$ over a small time interval Δt,

$$A(\mathbf{x},t)\Delta t = \int_t^{t+\Delta t} \mathrm{d}t'\, \hat{A}(\mathbf{x},t') = \int_t^{t+\Delta t} \mathrm{d}t' \sum_{j\in\mathbf{x}} S_j(t') \ . \tag{6}$$

Then, $A(\mathbf{x},t)\Delta t$ is the total number of spikes released by the pool neurons in the interval $(t, t+\Delta t]$ [2]. Figure 2 shows the interpretation of the pool activity by summation of spikes in a time interval. The pool activity $A(\mathbf{x},t)$ has the dimension spikes/time and is *extensive*. If desired, it can be normalized by the **number of pool neurons**,

$$N(\mathbf{x}) = \sum_{j\in\mathbf{x}} 1 \ . \tag{7}$$

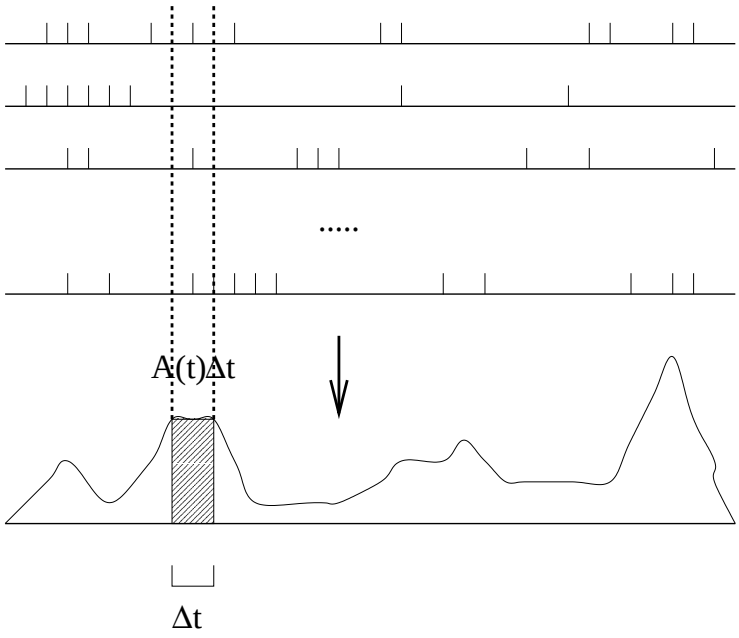

Figure 2: The joint activity of an assembly of spiking neurons. $A(t)\Delta t$ is the total number of spikes elicited by neurons of the assembly during the interval $(t, t+\Delta t]$.

Since neurons communicate with each other by means of their spike trains, the joint pool activity $A(\mathbf{x},t)$ is essentially everything a neuron "feels" from a pool $\mathbf{x}$.

[2] Under rigourous considerations, to get results that are independent of Δt, Eq. 6 with $\lim \Delta t \to 0$ has to be used for calculations with the pool activity $A(\mathbf{x},t)$. But since this does not modify the results of our calculations, we will use Eq. 5 and 6 interchangeably.

2.3.2 Neuronal Density

So far we have always calculated a sum over all neurons of an assembly to calculate pool averaged quantities. In the following we define the discrete assembly average as

$$\langle \ldots \rangle (\mathbf{x}, t) := \sum_{j \in \mathbf{x}} \ldots \quad . \tag{8}$$

Then, we can write for the total number of pool neurons

$$N(\mathbf{x}) = \langle 1 \rangle \ . \tag{9}$$

For extensively many neurons in an assembly, the sum can be approximated by an integral over the relevant state variables, if the **neuronal density** function $\rho(\ldots)$ is known.

If the internal state of a neuron (without considering input from external sources) is independent of input prior to the last firing at t^*, i.e., if the renewal hypothesis from section 2.2.2 holds, we may formulate the neuronal density as a function of t^*. The current pool density $\rho(\mathbf{x}, t, t^*)$ then quantifies the number of neurons of pool $\mathbf{x}$ at time t which are in a refractory state defined by their last spike at t^* [3]. For such a system we define an assembly average as

$$\langle \ldots \rangle_{t^*} (\mathbf{x}, t) := \int_{-\infty}^{t} \mathrm{d}t^* \, \rho(\mathbf{x}, t, t^*) \ \ldots \quad . \tag{10}$$

In the following sections, we will use a density function that is normalized to the total number of neurons that are available in the assembly, so that

$$N(\mathbf{x}) = \langle 1 \rangle_{t^*} \ . \tag{11}$$

For pools composed of large numbers of neurons (say, $\geq 10^2$), it is often convenient to describe assembly dynamics directly in terms of neuronal density dynamics, instead of regarding single neurons individually. This will be our approach in the next sections.

3 Integral Equation Assembly Dynamics

In section 2 we argued that a description at the level of single neurons is too detailed in many cases, and proposed the level of neuronal assemblies as the appropriate to understand both neuronal dynamics and functional organization. The neuronal assembly may thus constitute the computational unit for large-scale biologically realistic networks. In this section, we derive expressions in form of integral equations that can be used to model the dynamics of the joint activity of an assembly of spiking neurons with renewal property in a quantitatively exact manner.

[3] We interprete $\rho(\mathbf{x}, t, t^*)$ as a density function of t^*, so that $\mathbf{x}$ and t can be understood as parameters. Therefore, we could also write $\rho_{\mathbf{x},t}(t^*)$.

3.1 Derivation of the Assembly Activity Dynamics

The assembly dynamics of a pool of spiking neurons with renewal (i.e., neurons with refractory properties that depend only on the time $t - t^*$ elapsed since their last spiking time t^*) can be described using the neuronal density $\rho(\mathbf{x}, t, t^*)$ of section 2.3.2. By definition, the density is ≥ 0 only for $t^* < t$ (and 0 elsewhere) and goes to 0 for $t - t^* \to \infty$. This is shown in Fig. 3 a).

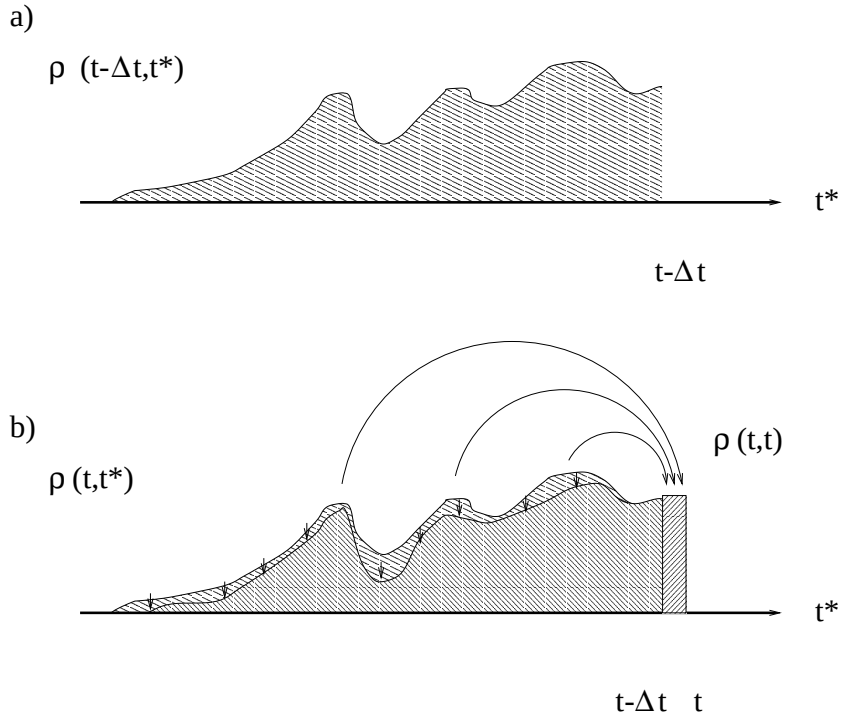

Figure 3: Densities $\rho(t - \Delta t, t^*)$ and $\rho(t, t^*)$ of spiking neurons with renewal, with fixed last spiking times t^*. a) **Old density** $\rho(t - \Delta t, t^*)$. b) **New density** $\rho(t, t^*)$. All neurons that spike cause a decrease of the density (small arrows). The same amount contributes to the density $\rho(t, t)$ (right end of the function in b), since their last spike time was modified to $t^* = t$, and, in addition, the total number of neurons remains constant (i.e., the density remains normalized).

All that remains to do is to calculate an explicit expression for the *dynamics of the density* $\rho(\mathbf{x}, t, t^*)$. Having accomplished this, every macroscopic (i.e., assembly-averaged) quantity can be calculated, including the activity $A(\mathbf{x}, t)$. The argumentation that allows to calculate the density dynamics is the following. The density is modified by a single process. When moving in time, $t - \Delta t \to t$, neurons with a *fixed* t^* can fire with a probability $\Delta t\, \lambda(\mathbf{x}, t, t^*)$ [Eq. (4)], which causes them to jump out from the subpopulation of neurons with last firing time t^* and accordingly to diminish the density $\rho(\mathbf{x}, t, t^*)$. The process is shown in Fig. 3 b), together with the resulting new density $\rho(\mathbf{x}, t, t^*)$. At the right end of the density function in b) the new density $\rho(\mathbf{x}, t, t)$ appears, which has a special significance that will be explained

below.

Since $\rho(\mathbf{x}, t, t^*)$ is modified solely by the firing of neurons with their respective probabilities, taking $\lim \Delta t \to 0$ we get the dynamics

$$\frac{\partial}{\partial t}\rho(\mathbf{x}, t, t^*) = -\lambda(\mathbf{x}, t, t^*)\, \rho(\mathbf{x}, t, t^*) \ . \tag{12}$$

The minus sign indicates that the neuronal density at a fixed t^* can only *decrease* in time when neurons from that subpopulation fire.

Equation (12) is only valid for $t^* < t$. For a step forward in time, we simply assume (yet unknown) boundary conditions $\rho(\mathbf{x}, t^*, t^*) = \rho_0(\mathbf{x}, t^*)$. Integration over t including the respective boundary conditions yields the analytic expression

$$\rho(\mathbf{x}, t, t^*) = \rho_0(\mathbf{x}, t^*) \exp\left\{ - \int_{t^*}^{t} \mathrm{d}t' \, \lambda(\mathbf{x}, t', t^*) \right\} \ . \tag{13}$$

This is basically all that we need to know to calculate the pool dynamics. We see from (13) that $\rho(\mathbf{x}, t, t^*)$ decreases steadily in time starting at its maximum value $\rho_0(\mathbf{x}, t^*)$. Therefore, the quantity $\rho_0(\mathbf{x}, t^*)$ is the maximum number of neurons ever with last spiking times at t^*. This number is equivalent to the *number of neurons with last spiking times at t^* available at time t^**, just before the subpopulation of neurons with last spiking times t^* begins to get decimated because of firing of single neurons. But this is just the number of neurons that *spiked* at t^*, because these are the neurons that contributed to $\rho(\mathbf{x}, t^*, t^*)$! Therefore, we have that

$$A(\mathbf{x}, t^*) = \rho(\mathbf{x}, t^*, t^*) = \rho_0(\mathbf{x}, t^*) \ . \tag{14}$$

Putting all this together, the explicit dynamics for $A(\mathbf{x}, t)$ is gained in a straightforward way. We assume that $\rho_0(\mathbf{x}, t^*)$ is known for $t^* \in [-\infty, t - \Delta t]$, because we stored the past activities $A(t^*)$ in memory. The expectation value for the activity (i.e., the pool spike density) at time t is then gained by averaging the spiking probability density $\lambda(\mathbf{x}, t, t^*)$ over all pool neurons, using the density at the preceding time step $\rho(\mathbf{x}, t - \Delta t, t^*)$ from Eq. (13), so that we get

$$A(\mathbf{x}, t) = \langle \lambda(\mathbf{x}, t, t^*) \rangle_{t^*} \ . \tag{15}$$

Using $\lim \Delta t \to 0$, we get the time-continuous form of the dynamics.

The calculation of the activity $A(\mathbf{x}, t)$ can also be understood by argueing that all neurons that spike at t [and thus modify $\rho(\mathbf{x}, t, t^*)$] contribute to increase $\rho(\mathbf{x}, t, t) = A(\mathbf{x}, t)$. This is indicated by the arc-shaped arrows in Fig. 3 b). It follows that $A(\mathbf{x}, t)$ can be gained by adding $-\partial/\partial t\, \rho(\mathbf{x}, t, t^*)$ for all $t^* < t$. Using Eqs. (13) this results again in Eqs. (15).

The two derivations [averaging the spiking probability over all pool neurons vs. adding $-\partial/\partial t\, \rho(\mathbf{x}, t, t^*)$] are equivalent because the density $\rho(\mathbf{x}, t, t^*)$ is normalized to the number of pool neurons [see also Eq. (11)],

$$N(\mathbf{x}) = \int_{-\infty}^{t} \mathrm{d}t^* \, \rho(\mathbf{x}, t, t^*) \ . \tag{16}$$

Differentiating in time,

$$\frac{\mathrm{d}}{\mathrm{d}t}N(\mathbf{x}) = 0 = \underbrace{\rho(\mathbf{x},t,t)}_{=A(\mathbf{x},t)} + \int_{-\infty}^{t} \mathrm{d}t^* \, \frac{\partial}{\partial t}\rho(\mathbf{x},t,t^*) \ , \tag{17}$$

we get for the activity

$$A(\mathbf{x},t) = -\int_{-\infty}^{t} \mathrm{d}t^* \, \frac{\partial}{\partial t}\rho(\mathbf{x},t,t^*) \ . \tag{18}$$

Together with equation (12), this equation is again equivalent to (15). From it, it can also be seen that the time-continuous calculation of the activity [i.e., taking $\lim \Delta t \to 0$ in (15)] is valid, because (18) leads to the same expressions as (15) but has been calculated using the actual density $\rho(\mathbf{x},t,t^*)$ instead of the density at the preceding time step $\rho(\mathbf{x},t-\Delta t,t^*)$.

Now we have all necessary ingredients for the calculation of a pool's activity. From (12), $\partial\rho(\mathbf{x},t,t^*)/\partial t$ is known, while $\rho(\mathbf{x},t,t^*)$ is known from (13) together with (14). The full, closed form of the dynamics as a function of the past activity is gained by taking (15) and calculating the assembly average using (13) with (14), to get

$$A(\mathbf{x},t) = \int_{-\infty}^{t} \mathrm{d}t^* \, \lambda(\mathbf{x},t,t^*) \, A(\mathbf{x},t^*) \exp\left\{-\int_{t^*}^{t} \mathrm{d}t' \, \lambda(\mathbf{x},t',t^*)\right\} \tag{19}$$

Summarizing, we have gained an integral equation that can be used to calculate $A(\mathbf{x},t)$ as a function of the past activities $A(\mathbf{x},t')$ and the past spike release probability density $\lambda(\mathbf{x},t',t^*)$, $t' < t$ [4]. Eqs. (15), (18) or (19) are equivalent representations of the general equation for the calculation of macroscopic assembly dynamics for pools of extensively many spiking neurons with renewal. Eq. (19) has appeared previously (most notably in Gerstner & van Hemmen 1994) in similar forms which will be analyzed in section 4.1; in this paper, we have presented a very short and intuitive derivation of the dynamics and we will concentrate on explaining how a large series of seemingly different models for assembly dynamics can be understood starting from (15), (18) or (19).

3.2 Survival Function Form

The density $\rho(\mathbf{x},t,t^*)$ can be normalized by the number of neurons that are originally found in the subgroup with fixed t^*, [5]

$$D_\lambda(\mathbf{x},t,t^*) := \frac{\rho(\mathbf{x},t,t^*)}{\rho(\mathbf{x},t^*,t^*)} \ . \tag{20}$$

We gain by this way an expression that tells us something about the probability [6] that a neuron that was in a state with last firing time t^* at time t^* is still in the

[4] The function is causal since $\partial\rho(\mathbf{x},t,t^*)/\partial t \to 0$ for $t^* \to t$, so that $A(\mathbf{x},t)$ does not appear on the right side of Eq. (18).

[5] The suffix λ indicates that the corresponding functions depend on the past spike release probability density function $\lambda(\mathbf{x},t',t^*)$, $t' < t$.

[6] To be precise, we should speak of the probability density, but we will use both terms interchangeably.

same state at time t. This can only be true if the neuron *did not* spike again during $(t^*, t]$. It is therefore called the survival probability function or, in short, **survival function**. It is the probability that a neuron that spiked last at t^* remained silent up to t, i.e., that it "survived".

Normalization of the negative partial time derivative of $\rho(\mathbf{x}, t, t^*)$ gives

$$S_\lambda(\mathbf{x}, t, t^*) := -\frac{\partial}{\partial t} \frac{\rho(\mathbf{x}, t, t^*)}{\rho(\mathbf{x}, t^*, t^*)} = -\frac{\partial}{\partial t} D_\lambda(\mathbf{x}, t, t^*) \ , \tag{21}$$

This is the probability that a neuron that was in a state with last firing time t^* at time t^* jumped out of this state at time t. This can only be true if the neuron remained silent during (t^*, t) and spiked at t. It is therefore called the **first spiking probability function** of a neuron that spiked last at t^*.

Using (14) and (21), the activity $A(\mathbf{x}, t^*)$ can be extracted from the dynamics Equation (18)

$$A(\mathbf{x}, t) = -\int_{-\infty}^{t} \mathrm{d}t^* \underbrace{\frac{\partial}{\partial t} \frac{\rho(\mathbf{x}, t, t^*)}{\rho(\mathbf{x}, t^*, t^*)}}_{=-S_\lambda(\mathbf{x}, t, t^*)} \underbrace{\rho(\mathbf{x}, t^*, t^*)}_{A(\mathbf{x}, t^*)} \ . \tag{22}$$

The result is the often frequented expression (Gerstner 1995) for assembly dynamics

$$A(\mathbf{x}, t) = \int_{-\infty}^{t} \mathrm{d}t^* \, S_\lambda(\mathbf{x}, t, t^*) \, A(\mathbf{x}, t^*) \ . \tag{23}$$

This equation is easy to understand, since $A(\mathbf{x}, t^*)$ is the number of neurons that spiked last at t^*, and $S_\lambda(\mathbf{x}, t, t^*)$ is the probability that a single neuron that spiked last at t^* contributes to spiking at t, so that $S_\lambda(\mathbf{x}, t, t^*) \, A(\mathbf{x}, t^*)$ is the probability that any neuron of the assembly $\mathbf{x}$ that spiked last at t^* contributes to spiking at t.

In its full form, the survival function $D_\lambda(\mathbf{x}, t, t^*)$ and its corresponding first spiking probability function $S_\lambda(\mathbf{x}, t, t^*)$ can be calculated according to Eqs. (20), (21), (14), (13), and (12). We then get

$$\begin{aligned} D_\lambda(\mathbf{x}, t, t^*) &= \exp\left\{ -\int_{t^*}^{t} \mathrm{d}t' \, \lambda(\mathbf{x}, t', t^*) \right\} \\ S_\lambda(\mathbf{x}, t, t^*) &= \lambda(\mathbf{x}, t, t^*) \, D_\lambda(\mathbf{x}, t, t^*) \ , \end{aligned} \tag{24}$$

The survival function $D_\lambda(\mathbf{x}, t, t^*)$ has a monotonously decaying form from 1 to 0, with a faster/slower decay for larger/smaller spike release probabilities λ. The first spiking probability function $S_\lambda(\mathbf{x}, t, t^*)$ is its negative derivative with respect to time, see Eq. (21).

4 Consequences

4.1 Special Forms of the Assembly Dynamics

4.1.1 Microscopic Neuronal Dynamics: Spiking Neurons as Threshold Firing Devices

Simplified models of single neurons that do not take into consideration the spatial organization of the dendritic tree and the axon usually rely upon the assumption that

the neuron works as a threshold device that releases a spike (an action potential) at those moments t^f in time t when a single scalar variable [the **membrane potential** $v(t)$] crosses a fixed **threshold** ϑ from below. Examples of models that fall into this category are the **spike–response–model (SRM)** (Gerstner 1990; Gerstner 1995; Gerstner & van Hemmen 1992; Gerstner & van Hemmen 1994) and the **integrate–and–fire (I&F)** type models (see e.g. Tuckwell 1988). These latter constitute a subcategory of the SRM, since they can be described entirely within the SRM framework. In addition, it can be shown that biologically more detailed models such as the **Hodgkin and Huxley (HH)** model can be mapped quantitatively onto the SRM model, so that HH neurons can also be regarded as threshold devices (Kistler, Gerstner, & van Hemmen 1997).

In threshold models, the membrane potential usually has two components, one that describes the *refractory influences* (the **"refractory field"** $v^{\rm ref}$) and another one that takes into account the synaptic input arriving at a cell through synaptic transmission, the **synaptic field** $h(t)$. In the simplest case, the membrane potential $v(t)$ is a *linear* summation of both terms,

$$v(t) = v^{\rm ref}(t) + h(t) \ . \tag{25}$$

In short, $v^{\rm ref}(t)$ is the contribution to the membrane potential generated by the neuron's response properties (such as its refractarity) whereas $h(t)$ is the contribution to the membrane potential from synaptic inputs (which can include synaptic connections from a neuron back to itself).

In the case of renewal, $v^{\rm ref}(t)$ depends on the time $s = t - t^*$ elapsed since the last firing time t^*, so that it is $v^{\rm ref}(t) = \eta(t - t^*)$ with the **refractory function** $\eta(s)$. The linearly composed membrane potential (25) then reads:

$$v(t) = \eta(t - t^*) + h(t) \ . \tag{26}$$

In the following, we write $v(t, t^*)$ instead of $v(t)$ in (26) to indicate explicitly the dependency of the membrane potential from the last spiking time t^*.

4.1.2 Threshold Noise

Noise can be added to threshold models of spiking neurons in a variety of ways. Here, we will restrict ourselves to a single type of noise, namely **threshold noise**. This type of noise is quite general in that it can be used to approximate other types of noise and the results still match quantitatively well (Gerstner 1998).

In section 2.2.3, we introduced a spike release probability λ. Here, we move a step forward and express λ in terms of the membrane potential for neurons with renewal as presented in Eq. (26). This means that we have a λ that depends on the distance of a cell's actual membrane potential v to its **firing threshold** ϑ. *The closer v is to the threshold, the more likely it is that a fluctuation will allow the neuron to reach threshold and trigger a spike.* We define this *escape rate* as a general function λ of the following form,

$$\lambda(t, t^*) := \{\tau^{\rm SRM}[v(t, t^*)]\}^{-1} \tag{27}$$

(written using a reciprocal so that the letter τ is indicating units of time). For an infinitesimally small time interval of length Δt, the spiking probability is then equal to $\Delta t[\tau^{\mathrm{SRM}}(v)]^{-1}$.

As a special realization of $[\tau^{\mathrm{SRM}}(v)]^{-1}$, the exponential Ansatz (Gerstner 1995) for the escape rate, i.e.,

$$\lambda(t,t^*) := \{\tau^{\mathrm{esc}}[v(t,t^*)]\}^{-1} := \tau_0^{-1}\exp\{2\beta[v(t,t^*)-\vartheta]\} \tag{28}$$

will be called in the following **escape noise**. The constant β is the noise level, which is something like an inverse temperature: The larger beta, the noisier (hotter) the system behaves. The constant τ_0^{-1} is the escape rate at threshold.

4.1.3 Escape Noise and the Activation Function

The escape noise as it is defined by Eq. (28) has a nice property: Using a membrane potential for neurons with renewal $v(t,t^*) = \eta(t-t^*) + h(t)$ as in (26), the contributions of the refractory and the synaptic components to a cell's noisy behavior factorize, allowing an easy interpretation of the neuronal dynamics.

For this purpose, we introduce a three-state neuron which can be in one of three different states: it can be *inactivated* (**i**), it can be *activated* (**a**), or it can be *firing* (**f**). A neuron can only fire, i.e., release a spike, if it is activated. If this is the case, the neuron fires with some probability $\Delta t/\tau(h)$ during the interval $(t, t+\Delta t]$, depending on its synaptic input field h. After the release of a spike, the neuron remains inactivated for a certain time period of length γ^{abs}. During this period, it is not allowed to spike, so that it is in an **absolute refractory state**. Following the absolute refractory period, the neuron enters a **relative refractory period** during which the neuron has a certain probability $p_{\mathrm{A}} > 0$ of getting activated, and thus a non-vanishing total probability for a spike release. We assume that only the time elapsed since the very last spike of a neuron at t^* determines its refractoriness, so that we end up with an **activation probability function** $1 > p_{\mathrm{A}}(t-t^*) \geq 0$ for $t \geq t^*$ (this is the "renewal hypothesis" for spiking neurons introduced in section 2.2.1).

Figure 4 shows the three possible internal states of a single neuron and the allowed transitions. We assume the transitions between the inactivated and the activated state to occur at a fast timescale as compared to the transition from the activated to the firing state and the modification of the activation probability p_{A} with time. It is therefore sufficient to regard the *mean occupation* p_{A} of the activated state of a neuron. From the activated state, and depending on the synaptic input field h, a neuron can be pushed into the firing state with a rate $[\tau(h)]^{-1}$. A neuron in the firing state releases a single spike and drops immediately back into the inactivated state.

In summary, the total firing probability during an interval $(t-\Delta t, t]$ of a neuron i is given by the joint probability that the neuron is in an activated state and that it is pushed into the firing state by its synaptic field h during that time interval,

$$\begin{aligned}&\mathrm{Prob}\,\{i \in \mathbf{a} \text{ and } i \text{ fires in } (t,t+\Delta t] \text{ due to field } h\}\\ &= \mathrm{Prob}\{i \text{ fires in } (t,t+\Delta t] \text{ due to field } h \mid i \in \mathbf{a}\}\,\mathrm{Prob}\{i \in \mathbf{a}\}\end{aligned}$$

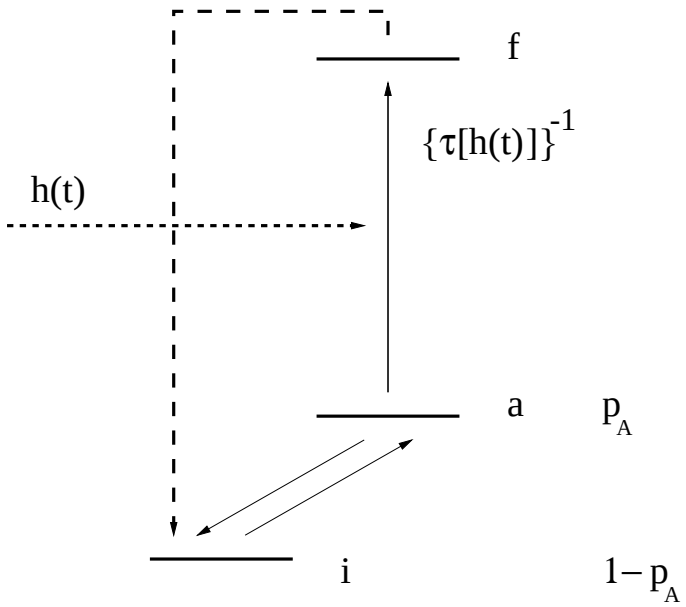

Figure 4: Definition of the microscopic model of a stochastically spiking neuron. In the escape noise case, a neuron can be interpreted as beeing in either of three states: inactivated (**i**), activated (**a**), or firing (**f**). Transitions between the three states are allowed between the **i** and the **a** levels (fast, with transition rates that depend on the last spike time t^*), from the **a** to the **f** level [slow, with a transition rate that depends on the synaptic input field $h(t)$] and from the **f** back to the **i** level (fast). The mean occupation of the **a** level is given by the *activation probability* $p_{\rm A}(t-t^*)$. Refractoriness means $p_{\rm A}(t-t^*) < 1$. A neuron can only release a spike if it is activated (**a**). The *firing probability for activated* neurons in a time interval of length Δt is field-dependent and equal to $\Delta t\{\tau[h(t)]\}^{-1}$. After firing, t^* is reset and the neuron is inactivated [$p_{\rm A}(t-t^*)=0$].

$$= \Delta t[\tau(h)]^{-1}\, p_{\rm A}(t-t^*)\ . \tag{29}$$

The refractory properties are governed by the time course of the activation probability function $1 > p_{\rm A}(t-t^*) \geq 0$, which in this work will also be called in short the **"activation function"**. It is divided into two parts. For a period of length $\gamma^{\rm abs}$, we have the *absolute refractory period*, with $p_{\rm A}(s) = 0$ and $s = t - t^*$ the time elapsed since the last spike. After that period, the neuron enters the *relative refractory period*, during which $p_{\rm A}(s)$ rises from some value $p_{\rm A}(\gamma^{\rm abs})$ towards 1 for $s \to \infty$, according to a differentiable function $P_{\rm A}(s)$. Between the two refractory periods, we allow a discontinuity of the function $p_{\rm A}(s)$ at $\gamma^{\rm abs}$,

$$p_{\rm A}(s) = \begin{cases} 0 & \text{for } 0 \leq s < \gamma^{\rm abs} \\ P_{\rm A}(s) & \text{for } s \geq \gamma^{\rm abs} \end{cases}\ . \tag{30}$$

Comparing (28) and (29) and using $v(t,t^*) = \eta(t-t^*) + h(t)$ (26), we can express the spike probability density for activated neurons and the activation probability for refractory neurons as a function of the synaptic field h and the refractory field $\eta(s)$ as follows:

$$[\tau(h)]^{-1} := \tau_0^{-1} \exp[2\beta(h-\vartheta)] \tag{31}$$

and

$$p_{\rm A}(t-t^*) := \exp[2\beta\eta(t-t^*)]\ , \tag{32}$$

so that for escape noise it is

$$\lambda(t,t^*) := \{\tau^{\rm esc}[v(t,t^*)]\}^{-1} = \{\tau[h(t)]\}^{-1}\, p_{\rm A}(t-t^*)\ . \tag{33}$$

In the following, we will use both implementations of noise [(27) and (33)] in the formulation of our assembly dynamics, since they lead to different forms of equations. Nevertheless, we remark here that, for the case of non-vanishing noise β and exponential escape rate (28), both implementations of noise are equivalent.

4.1.4 Different Forms of Integral Equation Assembly Dynamics

The different noise models, viz., threshold noise using $\tau^{\rm SRM}(v)$ (27) or escape noise which uses $\tau^{\rm esc}(v)$ [in the form of Eq. (33)] have slightly different representational and dynamical consequences, which will be extended in the following sections. Whereas the more general threshold noise (27) allows for a description of assembly dynamics even for the no-noise, deterministic limit, it exhibits a normalization problem when implemented numerically in simulations (see App. A.2). On the other hand, the escape noise (33) form of assembly dynamics does not allow a description of deterministic but only of noisy dynamics, with the advantage that it does not have normalization problems, that it delivers a more intuitive interpretation of the dynamics and that it can be converted into a differential equation system. Therefore, in this section, we will write down the assembly dynamics for both noise models explicitly. We start with the general formulation for assembly activity dynamics using (15) and the respective spike release probability functions (27) or (33), which results in

$$A(\mathbf{x},t) = \left\langle \frac{1}{\tau^{\rm SRM}[h(\mathbf{x},t) + \eta(t-t^*)]} \right\rangle_{t^*} \tag{34}$$

for the general threshold noise case, or, alternatively, for the escape-noise case, in

$$A(\mathbf{x},t) = \left\langle \frac{1}{\tau[h(\mathbf{x},t)]}\, p_{\mathrm{A}}(t-t^*) \right\rangle_{t^*} \quad . \tag{35}$$

The survival function form (23) (Gerstner 1995, repeated here for completeness) of the integral equation dynamics [7]

$$A(\mathbf{x},t) = \int_{-\infty}^{t} \mathrm{d}t^*\, S_h(\mathbf{x},t,t^*)\, A(\mathbf{x},t^*) \tag{36}$$

is also often encountered with either of the noise models (27) or (33). The explicit analytical expressions for the different microscopic noise models then yields

$$\begin{aligned} D_h(\mathbf{x},t,t^*) &= \exp\left\{ -\int_{t^*}^{t} \mathrm{d}t' \, \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x},t') + \eta(t'-t^*)]} \right\} \\ S_h(\mathbf{x},t,t^*) &= \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x},t) + \eta(t-t^*)]}\, D_h(\mathbf{x},t,t^*) \ , \end{aligned} \tag{37}$$

or, for the escape-noise case,

$$\begin{aligned} D_h(\mathbf{x},t,t^*) &= \exp\left\{ -\int_{t^*}^{t} \mathrm{d}t' \, \frac{1}{\tau[h(\mathbf{x},t')]}\, p_{\mathrm{A}}(t'-t^*) \right\} \\ S_h(\mathbf{x},t,t^*) &= \frac{1}{\tau[h(\mathbf{x},t)]}\, p_{\mathrm{A}}(t-t^*)\, D_h(\mathbf{x},t,t^*) \ . \end{aligned} \tag{38}$$

These two forms of the integral assembly dynamics can be used equivalently, depending on the type of noise model and on whether we choose to represent the refractory properties of a neuron by its refractory function $\eta(s)$ [leading to the assembly dynamics (36) and (37)] or its activation probability function $p_{\mathrm{A}}(s)$ [leading to the assembly dynamics (36) and (38)].

4.1.5 Escape Noise Assembly Dynamics

Eqs. (18) and (23) are fairly general in a sense that they can be used to model assembly dynamics of pools of any type of SRM neurons with renewal. In the case of escape noise, we can derive a different, equivalent expression for pool dynamics which has an intuitive interpretation and which is sometimes preferably used instead of (23). In addition, the new expression allows to derive pool dynamics in the form of a differential equation system, which will be explained in the next section.

We start with the noise model from section 4.1.3. Important here is the notion of the **mean inactivation level**, or *total number of inactivated neurons* $N_{\mathrm{I}}(\mathbf{x},t)$ of a pool. This quantity tells us something about the responsiveness of a pool to incoming synaptic input. If $N_{\mathrm{I}}(\mathbf{x},t)$ is large, many neurons will be *inactivated*, and little response to a stimulation will occur. If it is small, only a few neurons are inactivated, and many neurons are ready to respond to stimulation, yielding a fast

[7] Now the suffix h indicates that the corresponding functions depend on the past synaptic field $h(\mathbf{x},t')$, $t' < t$.

reaction to input. The mean inactivation level thus quantifies the "inertia" of a pool to incoming stimulation. For the calculation of $N_{\rm I}(\mathbf{x},t)$, we use $p_{\rm A}(s)$ from section 4.1.3, which is the probability of a neuron of being activated. Then we get

$$N_{\rm I}(\mathbf{x},t) = \langle 1 - p_{\rm A}(t-t^*)\rangle_{t^*} \ . \tag{39}$$

Since $N_{\rm I}(t)$ determines how fast the response of a pool's activity will be, it is of central importance for the description of pool dynamics. With

$$\rho(\mathbf{x},t,t^*) = D_h(\mathbf{x},t,t^*)A(\mathbf{x},t^*) \tag{40}$$

[see (20)], Equation (39) can be written using the past activity as

$$N_{\rm I}(\mathbf{x},t) = \int_{-\infty}^{t} {\rm d}t^* \left[1 - p_{\rm A}(t-t^*)\right] D_h(\mathbf{x},t,t^*)A(\mathbf{x},t^*) \ . \tag{41}$$

The number of pool neurons that *can* contribute to the activity $A(t+\Delta t)\Delta t$ during the next time step $(t, t+\Delta t]$, is given by the total number of *activated* neurons, $N(\mathbf{x}) - N_{\rm I}(\mathbf{x},t)$. Because the activated neurons contribute to spiking with a probability $\{\tau[h(\mathbf{x},t)]\}^{-1}\Delta t$ (see section 4.1.3), we get for the activity

$$A(\mathbf{x},t+\Delta t)\Delta t = \frac{\Delta t}{\tau[h(\mathbf{x},t)]}\left[N(\mathbf{x}) - N_{\rm I}(\mathbf{x},t)\right] \ . \tag{42}$$

This equation is valid as long as $A(\mathbf{x},t+\Delta t)\Delta t \ll N_{\rm I}(\mathbf{x},t)$. That is to say, as long as $N_{\rm I}(\mathbf{x},t)$ can be considered as approximately constant during a time interval of length Δt. We note that in this case the activation $A(\mathbf{x},t+\Delta t)$ does not depend on the length of the time interval Δt. For any small enough Δt, the result will be the same. Hence we can take the limit $\Delta t \to 0$, resulting in

$$A(\mathbf{x},t) = \frac{1}{\tau[h(\mathbf{x},t)]}\left[N(\mathbf{x}) - N_{\rm I}(\mathbf{x},t)\right] \ . \tag{43}$$

Upon substitution of (41) this becomes an integral equation for the time evolution of the activity $A(\mathbf{x},t)$ for spike–response neurons with escape noise. As in Eq. (36), the influence of other assemblies is hidden in the synaptic field $h(\mathbf{x},t)$, which can depend on the activities of all other pools that provide synaptic input, including itself [see the second line of Eq. (46) in the next section 4.2.1].

For escape noise, Eqs. (36) and (43) are mathematically equivalent. This can be seen directly by transforming (35) into

$$\begin{aligned} A(\mathbf{x},t) &= \left\langle \frac{1}{\tau[h(\mathbf{x},t)]} - \frac{1}{\tau[h(\mathbf{x},t)]}\left[1 - p_{\rm A}(t-t^*)\right] \right\rangle_{t^*} \\ &= \frac{1}{\tau[h(\mathbf{x},t)]}\{ \underbrace{\langle 1\rangle_{t^*}}_{=N(\mathbf{x})} - \underbrace{\langle 1 - p_{\rm A}(t-t^*)\rangle_{t^*}}_{=N_{\rm I}(\mathbf{x},t)}\} \ . \end{aligned} \tag{44}$$

Although equivalent, (43) has some analytical and numerical advantages over (36). We have derived and introduced (43) because the interpretation of $\langle 1 - p_{\rm A}(t-t^*)\rangle_{t^*}$ as the number of inactivated neurons allows an easier understanding of assembly dynamics and because it allows us to establish a link between integral and differential equation models of assembly dynamics.

4.2 Networks of Interacting Assemblies

4.2.1 Synaptic Field

In threshold models of spiking neurons (SRM and the SRM equivalent of integrate–and–fire models), the synaptic field of a single neuron is calculated as follows. The presynaptic neurons j release series of action potentials at times t_j^f each of which, after a fixed delay period, reaches a postsynaptic neuron i. This causes a temporal variation of the form $J_{ij}\,\alpha_{ij}(t-t_j^f)$ of the *membrane potential* at the postsynaptic neuron (the constant J_{ij} is the coupling strength for a synaptic connection from a presynaptic neuron j to a postsynaptic neuron i; the subindices i,j indicate that the constants of the α-function [see Eq. (45)] may vary for different connections).

The total variation of the postsynaptic membrane potential due to the incoming action potentials is the synaptic field $h_i(t)$ from Eqs. (25) or (26). Since we assume passive conducting characteristics of the dendritic tree, the synaptic field is calculated as a sum of the contributions of single action potentials. This means that

$$h_i(\mathbf{x},t) = \sum_j J_{ij} \int_{-\infty}^{t} \mathrm{d}t'\, \alpha_{ij}(t-t')\, S_j(t') \;. \tag{45}$$

The coupling strength from a connection between pools $\mathbf{x}$, $\mathbf{y}$ conveying signals from a neuron $j \in \mathbf{y}$ to a neuron $i \in \mathbf{x}$ is designated with $J_{ij}(\mathbf{x},\mathbf{y})$. Since the neuron indices do not matter if all neurons of a pool have the same connectivity characteristics, we can omit them and simply write $J(\mathbf{x},\mathbf{y})$. The same occurs with any other connectivity function or parameter, e.g. with $\alpha_{ij}(t-t_j^f)$. Therefore, we will omit as well the neuronal i,j indices for the α-function.

Using the pool definition, we can then write

$$\begin{aligned} h(\mathbf{x},t) &= \sum_{\mathbf{y}} \sum_{j\in\mathbf{y}} J(\mathbf{x},\mathbf{y}) \int_{-\infty}^{t} \mathrm{d}t'\, \alpha(\mathbf{x},\mathbf{y},t-t')\, S_j(t') \\ &= \sum_{\mathbf{y}} J(\mathbf{x},\mathbf{y}) \int_{-\infty}^{t} \mathrm{d}t'\, \alpha(\mathbf{x},\mathbf{y},t-t')\, A(\mathbf{y},t') \;. \end{aligned} \tag{46}$$

Therefore, interactions between pools occur exclusively mediated by the pool activities $A(\mathbf{y},t)$. In addition, all neurons i of a pool $\mathbf{x}$ feel the *same synaptic field* $h(\mathbf{x},t)$.

4.2.2 Reduction to a Network of Interacting Neuronal Assemblies

The consequences that can be drawn from Eq. (46) are significant. From the assembly dynamics derived throughout section 3, we know that the activity of an assembly $\mathbf{x}$ can be described entirely by means of its past activity $A(\mathbf{x},t')$ and its past synaptic input field $h(\mathbf{x},t')$, $t'<t$. This means that for a given synaptic input, it suffices to remember the past activities and everything relevant about the pool dynamics is known or can be calculated. On the other hand, we have just seen that the synaptic input fields can be calculated by knowing the past activities of $A(\mathbf{y},t')$ of

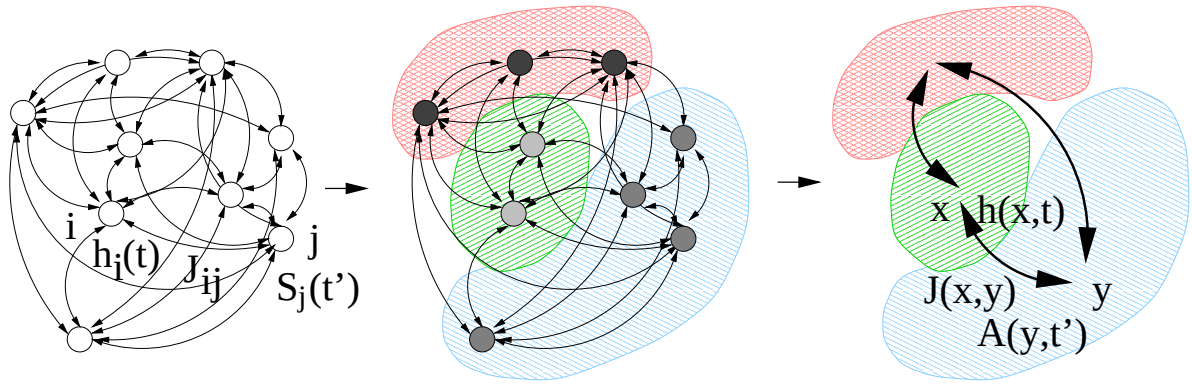

Figure 5: Reduction of a network of interacting spiking neurons to a network of interacting assemblies. Left: Network of interconnected spiking neurons. All a target neuron i needs for computing its synaptic contribution that arrives from a given source neuron j is the incoming spike train $S_j(t')$, $t' < t$. Middle: Grouping of the spiking neurons into non-overlapping assemblies according to the assembly definition and the connectivity architecture of the network. Here, the description moves from target and source neurons i, j and neuronal connections J_{ij} to target and source assemblies $\mathbf{x}$, $\mathbf{y}$ and inter-assembly connections $J(\mathbf{x},\mathbf{y})$. Right: The corresponding network of interconnected assemblies of spiking neurons. The synaptic contribution to a target pool $\mathbf{x}$ from a source pool $\mathbf{y}$ is calculated using the past activity $A(\mathbf{y},t')$, $t' < t$.

all assemblies $\mathbf{y}$ (including itself) that, by virtue of the network connections $J(\mathbf{x},\mathbf{y})$ from $\mathbf{y}$ to $\mathbf{x}$, provide a source of input signals to the pool $\mathbf{x}$.

This means that, under the architectural constraint of the assembly definition of section 2.1, a *network of interacting spiking neurons* which are threshold firing devices with renewal as explained in section 4.1.1, can be fully reduced to a *network of interacting neuronal assemblies*, in which each assembly is described by the macroscopic dynamics derived in this paper and with internal dynamics and inter-assembly-interactions that can be calculated from the past activities $A(\mathbf{x},t')$, $t' < t$. Fig. 5 explains this concept graphically.

Fig. 6 shows simulations of an assembly of integrate–and–fire neurons, modeled either using $n = 5{*}10^4$ single I&F neurons or the corresponding macroscopic assembly dynamics [in this simulation, we used the SRM assembly Eqs. (36) and (38)]. The I&F neuron model had an absolute refractory period of 4 ms, and an RC membrane decay time constant of 6 ms. This means that every time a spike was elicited, there was a 4 ms dead period, after which a value of 5 was substracted from the membrane potential and normal RC relaxation continued. The used noise model was escape-noise, given by (28), with $\tau_0 = 1$ ms, the inverse temperature $\beta = 1/T = 1/0.35$ and threshold at $\vartheta = 0.75$. The external input (in form of an additional contribution to the membrane potential) was varied in a series of random steps to induce different

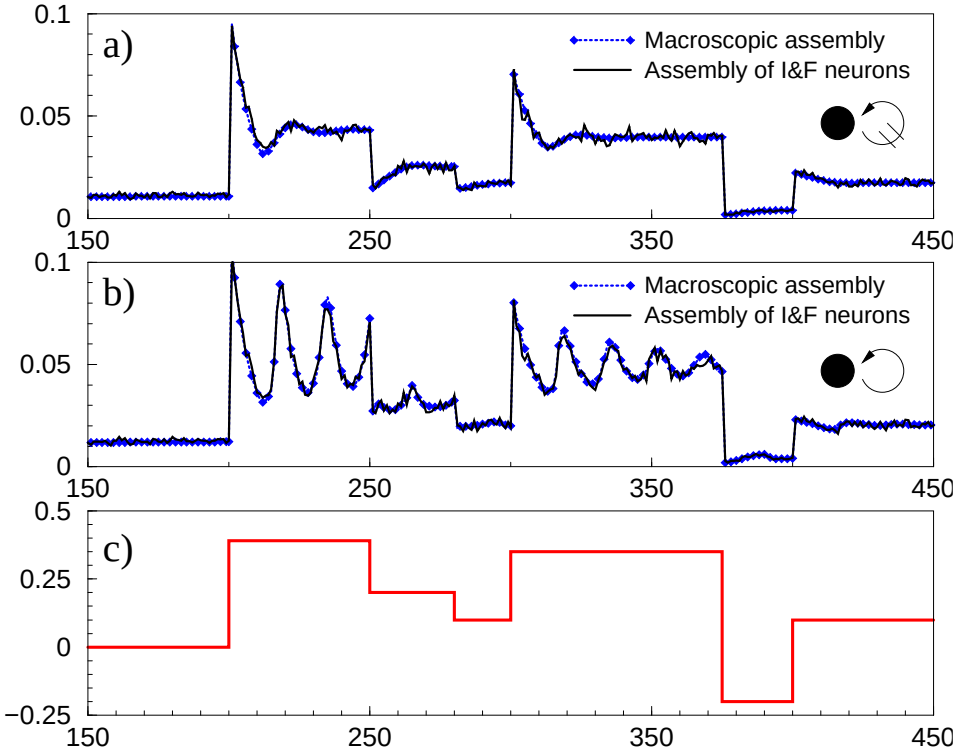

Figure 6: Simulations of the activity dynamics of an assembly of $5 * 10^4$ I&F neurons and the corresponding macroscopic assembly dynamics. In c), the variation of the external input is shown, included into the model as an additional contribution to the membrane potential. In the assembly without interactions of a), the simulations show repetitive relaxation processes to new equilibria when the external input changes. In b) the same assembly is shown, but with reciprocal synaptic connections. The parameters are chosen so that there is a resonance between the internal relaxation processes and the synaptic feedback. The simulations show that the quantitative correspondence is excellent.

responses of the assembly (Fig. 6 c). Fig. 6 a) shows the activity dynamics of an assembly without neuronal connections. Here, the approach of the activity to the new equilibria after a change of the piecewise constant external input to the membrane potential can be seen. In this case, the relaxation of the assembly dynamics to the microscopically correct stationary activities occurs approximately in a time period of about 20 ms, approximately determined by the length of the absolute refractory period and the RC relaxation time constant. Fig. 6 b) shows the activity dynamics of the very same assembly with synaptic interaction, with all neurons of the extensive assembly being coupled reciprocally with a weight $J_{ij} = 1/n$, and, in the assembly case, a coupling of the assembly to itself of $J(\mathbf{x}, \mathbf{x}) = 1$. The synaptic α-function (see section 4.2.1) was in both cases the same (for the I&F neurons, the corresponding variation function of the input *current* was used), and was chosen with delay and time constants that a resonance between the RC relaxation and the synaptic feedback occurs. This resonance can be seen in the more pronounced oscillations of the activity in Fig. 6 b) as compared to Fig. 6 a). The simulation shows that even under the adverse conditions of substantial feedback (which potentiates accumulated errors) and dynamics with sharp transients, there is a good quantitative agreement even for time periods surpassing 100 ms.

5 Differential Equation Assembly Dynamics

In section 3.1, we derived expressions for calculating the macroscopic assembly activity dynamics of spiking neurons with renewal. This was accomplished by introducing a neuronal density that is a function of the state of the neurons defined by their only remaining state variable t^*, their last spiking time. In sections 4.1.4 and 4.1.5, specific expressions for the assembly dynamics of SRM neurons in form of integral equations were presented. In this section, we take advantage of the escape noise assembly dynamics of section 4.1.5 to derive an equivalent differential equation system for describing assembly dynamics, which can then be used for further calculations and for comparison with existing differential equation models.

5.1 Special Activation Probability Functions

It can be asked how the integral equations models for describing neuronal assembly dynamics relate to the widely used differential equation models of the **graded–response** type. An exact reduction of the nonlinear integral equation Eq. (43) [or, equivalently, (36)] is possible for specific activation probability functions $p_{\mathrm{A}}(s)$ (Eggert & van Hemmen 2000) and, accordingly, for specific functions $P_{\mathrm{A}}(s)$ (30) for the relative refractory period. We will restrict ourselves to the exponential case (**exp**), the case of a sigmoid-like time evolution of the activation function after the absolute refractory period (**sigm**), and the case of an inverse decay (**inv**):

$$P_{\mathrm{A}}(s) = \begin{cases} 1 - p_0 \exp[-(s - \gamma^{\mathrm{abs}})/\tau_{\mathrm{ref}}] & \mathrm{exp} \\ 1 - p_0/\{1 + \exp[(s - s_0)/\tau_{\mathrm{ref}}]\} & \mathrm{sigm} \\ 1 - \tau_{\mathrm{ref}}/(s - s_0) & \mathrm{inv} \end{cases} . \tag{47}$$

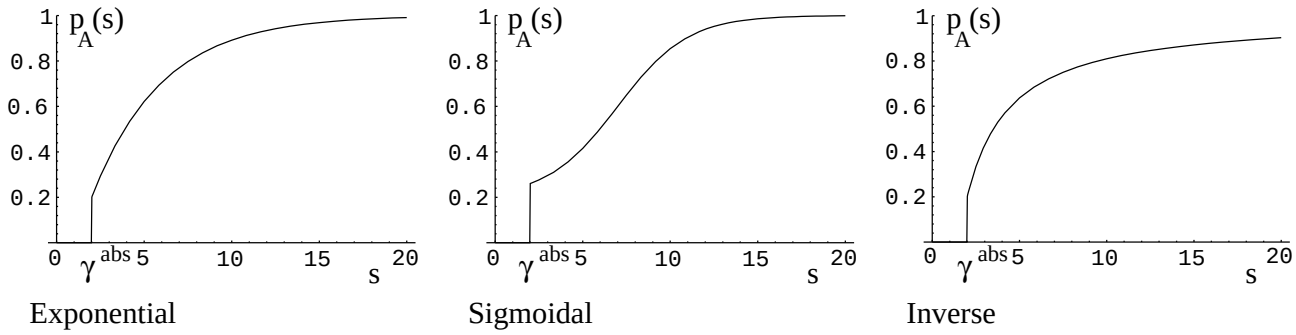

Figure 7: The different activation functions $p_{\mathrm{A}}(t-t^*)$ used for the derivation of the pool dynamics. At $s=\gamma^{\mathrm{abs}}$, the function may have a discontinuity. For $t \searrow t^*$, the neuron is in an absolute refractory state because it just spiked and $p_{\mathrm{A}}(t-t^*) \searrow 0$. For $t \to \infty$, the refractory effects vanish and $p_{\mathrm{A}}(t-t^*) \to 1$. In case of activation functions with a discontinuity, γ^{abs} is the length of the absolute refractory period.

The constants p_0, τ_{ref} and s_0 are free parameters of the activation function. It has to be verified that $0 \leq P_{\mathrm{A}}(s) \leq 1$ for $\gamma^{\mathrm{abs}} \leq s \leq \infty$; for example, for the inverse activation function in (47), this means that we effectively set $\gamma^{\mathrm{abs}} \geq \tau_{\mathrm{ref}} + s_0$, so that $P_{\mathrm{A}}(s)$ is only evaluated for $s \geq \tau_{\mathrm{ref}} + s_0$.

Figure 7 shows the different activation functions. It is $p_{\mathrm{A}}(s) = 0$ during the absolute refractory period and afterwards $p_{\mathrm{A}}(s)$ tends towards 1.

5.2 Recovery Variables

The number $N_{\mathrm{I}}(\mathbf{x},t)$ [Eq. (39)] of inactivated neurons of a pool is the assembly-averaged mean inactivation probability and can thus be written as $N_{\mathrm{I}}(\mathbf{x},t) = \langle 1 - p_{\mathrm{A}}(t-t^*)\rangle_{t^*}$. The kernel $1-p_{\mathrm{A}}(t-t^*)$ determines the influence of the past activity on the quantity $N_{\mathrm{I}}(\mathbf{x},t)$. Instead of using integral equations for the pool dynamics as in Eqs. (36) and (43), which incorporate the past activity by means of equidistant time slices (imagine a Riemann sum approximation of the integral equations), we could try to incorporate the past using a set of temporal kernels similar to $1-p_{\mathrm{A}}(t-t^*)$. The underlying problem is that of the reducibility of an integro-differential equation to a system of differential equations. It has been treated by a number of authors, e.g., see Fargue 1973; Fargue 1974. In principle, a reduction of Eqs. (36) or (43) into a chain of differential equations is possible for a suitable choice of intermediary variables. The problem is that there is no systematic derivation of these additional variables, so that we have to guess. The derivation of Eggert & van Hemmen (2000) makes use of the function $1-p_{\mathrm{A}}(t-t^*)$ for this purpose. [In Eq. (43), the difficult part of the assembly dynamics is hidden in the term $N_{\mathrm{I}}(\mathbf{x},t)$.]

To accomplish the reduction of (43) to a system of differential equations, we first have to introduce a further interesting assembly-averaged quantity: the number of pool neurons which are in their absolute refractory period because of recent spiking.

This quantity can be calculated as [8].

$$M(\mathbf{x},t) = \langle 1 - \Theta[(t-t^*) - \gamma^{\mathrm{abs}}] \rangle_{t^*} \quad . \tag{48}$$

Then, we redefine the number of inactivated neurons $N_{\mathrm{I}}(\mathbf{x},t)$ to $N^{(1)}(\mathbf{x},t)$, the total number of pool neurons, $N(\mathbf{x})$, to $N^{(0)}(\mathbf{x})$, and the number $M(\mathbf{x},t)$ of inactivated neurons for a pool with absolute refractory period only, to $N^{(\infty)}(\mathbf{x},t)$. Furthermore, we remark that the definitions (11) of $N(\mathbf{x}) = N^{(0)}(\mathbf{x})$, (39) of $N_{\mathrm{I}}(\mathbf{x},t) = N^{(1)}(\mathbf{x},t)$ and (48) of $M(\mathbf{x},t) = N^{(\infty)}(\mathbf{x},t)$ are equivalent to

$$\begin{aligned} N^{(0)}(\mathbf{x}) &= \left\langle [1 - p_{\mathrm{A}}(t-t^*)]^0 \right\rangle_{t^*} \quad , \\ N^{(1)}(\mathbf{x},t) &= \left\langle [1 - p_{\mathrm{A}}(t-t^*)]^1 \right\rangle_{t^*} \quad , \\ N^{(\infty)}(\mathbf{x},t) &= \langle [1 - p_{\mathrm{A}}(t-t^*)]^{\infty} \rangle_{t^*} \quad . \end{aligned} \tag{49}$$

Extending these definitions, in addition to $N_{\mathrm{I}}(\mathbf{x},t) = N^{(1)}(\mathbf{x},t)$ we get a series of time-dependent inactivation quantities, or **"recovery variables"**,

$$N^{(m)}(\mathbf{x},t) := \langle [1 - p_{\mathrm{A}}(t-t^*)]^m \rangle_{t^*} \tag{50}$$

so that, for $m \in \mathbb{N}$,

$$N^{(m)}(\mathbf{x},t) = \int_{-\infty}^{t} \mathrm{d}t^* \, [1 - p_{\mathrm{A}}(t-t^*)]^m \, \rho(\mathbf{x},t,t^*) \quad . \tag{51}$$

The $N^{(m)}(\mathbf{x},t)$ obey the relationship

$$N^{(0)}(\mathbf{x}) \geq N^{(1)}(\mathbf{x},t) \geq N^{(2)}(\mathbf{x},t) \geq \ldots \geq N^{(\infty)}(\mathbf{x},t) \quad \forall t \tag{52}$$

Figure 8 shows an example of a sigmoidal activation function $p_{\mathrm{A}}(s)$ with an absolute refractory period of length γ^{abs} and the recovery kernels $[1 - p_{\mathrm{A}}(s)]^m$.

5.3 Differential Equation System

Using the properties of the recovery variables, a recursive set of differential equations can be obtained (for the exact derivation, see Eggert & van Hemmen 2000)

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} N^{(m)}(\mathbf{x},t) &= A(\mathbf{x},t) - \{1 - [1 - p_{\mathrm{A}}(\gamma^{\mathrm{abs}})]^m\} A(\mathbf{x}, t - \gamma^{\mathrm{abs}}) \\ &\quad - \frac{1}{\tau[h(\mathbf{x},t)]} [N^{(m)}(\mathbf{x},t) - N^{(m+1)}(\mathbf{x},t)] \\ &\quad - \begin{cases} \frac{m}{\tau_{\mathrm{ref}}} [N^{(m)}(\mathbf{x},t) - M(\mathbf{x},t)] & \text{exp. } p_{\mathrm{A}}(s) \\ \frac{m}{\tau_{\mathrm{ref}}} \left\{ N^{(m)}(\mathbf{x},t) - M(\mathbf{x},t) - \frac{[N^{(m+1)}(\mathbf{x},t) - M(\mathbf{x},t)]}{p_0} \right\} & \text{sigm. } p_{\mathrm{A}}(s) \\ \frac{m}{\tau_{\mathrm{ref}}} [N^{(m+1)}(\mathbf{x},t) - M(\mathbf{x},t)] & \text{inv. } p_{\mathrm{A}}(s) \end{cases} . \end{aligned} \tag{53}$$

[8] Θ is the Heaviside step function, with $\Theta(x) = 1$ for $x \geq 0$ and 0 otherwise.

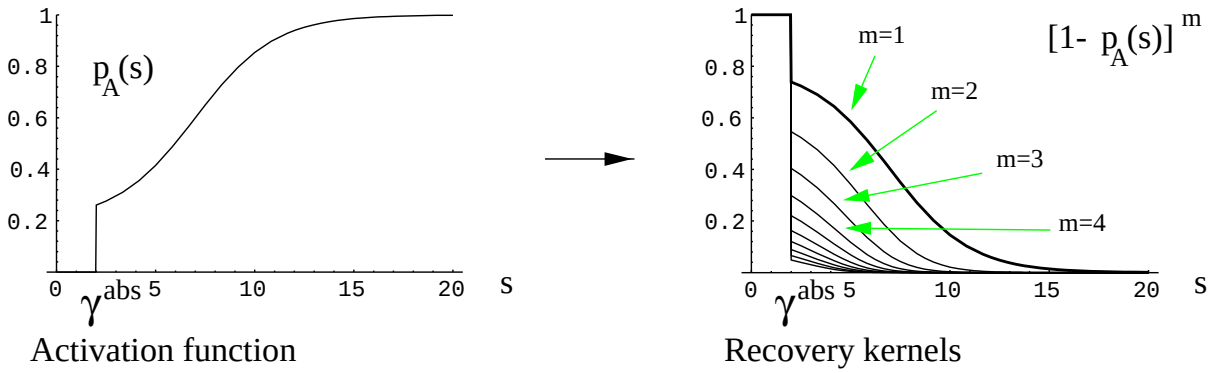

Figure 8: The activation function $p_{\rm A}(s)$ is shown (left) with its recovery kernels $[1-p_{\rm A}(s)]^m$ (right). With growing m, the kernels include less and less of the past time s. The pool dynamics is expressed with the help of a series of recovery variables $N^m(\mathbf{x},t) := \langle[1-p_{\rm A}(t-t^*)]^m\rangle_{t^*}$ calculated by computing the pool average of the corresponding recovery kernel function.

The last recovery variable $N^{(\infty)}(\mathbf{x},t) = M(\mathbf{x},t)$ increases with the number of spiking neurons and decreases with the number of neurons that are released from their absolute refractory phase,

$$\frac{\mathrm{d}}{\mathrm{d}t}M(\mathbf{x},t) = A(\mathbf{x},t) - A(\mathbf{x},t-\gamma^{\rm abs}) \ . \tag{54}$$

This completes the pool dynamics. Remind that for reciprocally connected pools, the system (53) contains the recovery variable $N^{(1)}(\mathbf{x},t')$ also through the activity $A(\mathbf{x},t')$ in the synaptic field $h(\mathbf{x},t)$. The complete dynamics is therefore defined by the field dynamics together with the dynamics of the recovery variables given by Eqs. (53) and (54). The spike density acts only as an auxiliary variable that is calculated from the first recovery variable using the main equation (43).

$$A(\mathbf{x},t) = \frac{1}{\tau[h(\mathbf{x},t)]}[N(\mathbf{x}) - N^{(1)}(\mathbf{x},t)] \ . \tag{55}$$

Other pools $\mathbf{y}$ influence the dynamics of pool $\mathbf{x}$ through the activities $A(\mathbf{y},t')$ in (46).

Using a differentiable activation function $p_{\rm A}(s)$ *without* absolute refractory period, the system (53) reduces to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}N^{(m)}(\mathbf{x},t) = A(\mathbf{x},t) - \frac{1}{\tau[h_i(\mathbf{x},t)]}[N^{(m)}(\mathbf{x},t) - N^{(m+1)}(\mathbf{x},t)] \\ - \begin{cases} \frac{m}{\tau_{\rm ref}}N^{(m)}(\mathbf{x},t) & \text{exp. } p_{\rm A}(s) \\ \frac{m}{\tau_{\rm ref}}\left[N^{(m)}(\mathbf{x},t) - N^{(m+1)}(\mathbf{x},t)/p_0\right] & \text{sigm. } p_{\rm A}(s) \\ \frac{m}{\tau_{\rm ref}}N^{(m+1)}(\mathbf{x},t) & \text{inv. } p_{\rm A}(s) \end{cases} .\end{aligned} \tag{56}$$

6 Connection with Other Models

In this section, we compare our model [integral forms (36) or (43), differential equation forms (53) resp. (56)] with other neuronal models. Specifically, we show that standard gain functions and graded–response models can be understood in terms of our pool dynamics, and that this allows us to interprete the parameters of gain functions and graded–response models in terms of the microscopic parameters of the underlying neuronal model. In adition, we show how other integral equation models can be understood as special cases of our dynamics, and how they can be used to understand single neuron spike statistics.

6.1 Gain Function

It can be shown that the stationary solution of the pool dynamics (36), (43), (53) or (56) is a sigmoidal gain function [for the detailed calculation, see App. B.2]. In case of neurons with escape noise and absolute refractory period only we get an equation of the same form as the standard **logistic gain function** known from computational neuroscience literature,

$$\begin{aligned} G[h(\mathbf{x})] &= \frac{1}{\gamma^{\mathrm{abs}}} \frac{1}{1+\exp\{-2\beta[h(\mathbf{x})-\vartheta']\}} \\ &= \frac{1}{\gamma^{\mathrm{abs}}} \frac{1}{2} \left(1+\tanh\{\beta[h(\mathbf{x})-\vartheta']\}\right) . \end{aligned} \tag{57}$$

with the modified threshold

$$\vartheta' = \vartheta + 1/(2\beta) \ln(\tau_0/\gamma^{\mathrm{abs}}) \tag{58}$$

Since $A_{\max} := 1/\gamma^{\mathrm{abs}}$ is the maximal spiking activity of the neurons, and normalizing the activity $A(\mathbf{x},t) \to A(\mathbf{x},t)/N(\mathbf{x})$, we get

$$A(\mathbf{x}) = A_{\max} \frac{1}{2} \left(1+\tanh\{\beta[h(\mathbf{x})-\vartheta']\}\right) . \tag{59}$$

This means that for spiking neurons we can use the standard logistic gain function to get realistic stationary results, and we know how each parameter of the gain function can be interpreted in terms of the microscopic parameters of the underlying neuronal model from section 4.1.1. For neurons with absolute and relative refractory period, a similar **assembly-averaged gain function** can be calculated, see App. B.2, Eq. (121).

6.2 Wilson and Cowan Integral Equation Model

With absolute refractory period only, it is $p_{\mathrm{A}}(s) = 0$ for $s < \gamma^{\mathrm{abs}}$ and $p_{\mathrm{A}}(s) = 1$ for $s \geq \gamma^{\mathrm{abs}}$. In this case, and for normalized activity $[A(\mathbf{x},t) \to A(\mathbf{x},t)/N(\mathbf{x})]$, Eq. (43) can be written as

$$\begin{aligned} A(\mathbf{x},t) &= \frac{1}{\tau[h(\mathbf{x},t)]} \left\{1 - \int_{-\infty}^{t} \mathrm{d}t^* \, \Theta[\gamma^{\mathrm{abs}} - (t-t^*)] A(\mathbf{x},t^*)\right\} \\ &= \frac{1}{\tau[h(\mathbf{x},t)]} \left\{1 - \int_{t-\gamma^{\mathrm{abs}}}^{t} \mathrm{d}t^* \, A(\mathbf{x},t^*)\right\} . \end{aligned} \tag{60}$$

This equation is formally identical to the **W+C integral equation model** postulated by Wilson & Cowan (1972) for the description of the activity of a population of spiking neurons. But in the Wilson and Cowan model, instead of $[\tau(h)]^{-1}$ a sigmoid function $S(h)$ is used, which is motivated by a variation of neuronal parameters among the neuronal population. This would introduce additional correlations (which, notably, are explicitly neglected in the Wilson and Cowan model). Furthermore, we do not need to postulate a sigmoid form for $[\tau(h)]^{-1}$ a priori, a sigmoidal shape of the assembly activity gain function [i.e., $A(\mathbf{x})$ as a function of a constant synaptic field h] is rather a consequence of the refractory characteristics of the single neuron. The Wilson and Cowan equation is thus a special case of (43), with the difference that (43) incorporates relative refractory behavior and that in (43) all parameters are given by microscopic (i.e., neuronal) parameters.

6.3 Synfire–Chain Pulse Propagation

The factor $N_{\mathrm{I}}(\mathbf{x}, t)$ from Eq. (43) is computed according to (41), using the past activity $A(\mathbf{x}, t^*)$ convolved with two functions, namely one minus the activation function $\zeta(t - t^*) := 1 - p_{\mathrm{A}}(t - t^*)$ and the survival function $D_h(\mathbf{x}, t, t^*)$. In some cases, the survival function is close to 1 for quite a long time period $t - t^*$ into the past, so that the function $\zeta(t - t^*)$ dominates the dynamics. In particular, this can be the case for low noise $\beta \to \infty$. In this case we can neglect the effect of the survival function and write for the normalized activity $[A(\mathbf{x}, t) \to A(\mathbf{x}, t)/N(\mathbf{x})]$.

$$A(\mathbf{x}, t) = \frac{1}{\tau[h(\mathbf{x}, t)]} \left\{ 1 - \int_{-\infty}^{t} \mathrm{d}t^* \, \zeta(t - t^*) A(\mathbf{x}, t^*) \right\} \tag{61}$$

instead of Eq. (43) [remark the similarity with the first line of (60)]. This equation is formally identical to the integral equation postulated by Aertsen and Gewaltig for the description of the propagation of pulse packets through the stages of a synfire–chain (**Pulse-propagation model**, M. O. Gewaltig, personal communication, and Gewaltig 1999). Therefore, Eq. (61) is an approximation of the exact equation Eq. (43).

6.4 Activity-Folding Models

One of the disadvantages of the exact equations (36) and (43) is the double integral evaluation [the integral over past spiking times t^*, but also the integral hidden in the survival function $D_h(\mathbf{x}, t, t^*)$, see Eq. (37) or (38)]. There have been a few attempts to introduce approximations that avoid this double evaluation and reduce the assembly dynamics basically to a single integral equation that folds the past activity. Eq. (61), which is gained from (43) with $D_h(\mathbf{x}, t, t^*) = 1$ is such an approximation, but it seriously underweights the activity of the more recent past, thus producing qualitatively wrong results on the long run.

Instead, a viable alternative turns out to be to start from the low-noise limit of (36). It can be shown that in this case, the evaluation of an integral equation can be completely avoided. In the noise-free case, a neuron with renewal and last spiking

time t^* spikes at t^*+s^* when its membrane potential (26) reaches the firing threshold ϑ from below, i.e., when

$$v(t) = \eta(s^*) + h(t) = \vartheta \ . \tag{62}$$

Eq. (62) implicitly [9] defines the backwards interval $s^*(\mathbf{x},t)$, meaning that, at time t, all those neurons contribute to spiking that had last spiking times $t - s^*(\mathbf{x},t)$, and these neurons are just given by the neuronal density $\rho[\mathbf{x},t,t-s^*(\mathbf{x},t)]$, which on its turn is proportional to the past activity $A[\mathbf{x},t-s^*(\mathbf{x},t)]$ because the system is deterministic. But we have to take care of distortions of the neuronal density due to the remapping of t^* to s^*, which results in $\rho[\mathbf{x},t,t-s^*(\mathbf{x},t)] = |1 - \mathrm{d}/\mathrm{d}t\, s^*(\mathbf{x},t)|\ A[\mathbf{x},t-s^*(\mathbf{x},t)]$, so that we finally get for the activity dynamics (Eggert & van Hemmen 2000)

$$A(\mathbf{x},t) = \left|1 - \frac{\mathrm{d}}{\mathrm{d}t} s^*(\mathbf{x},t)\right|\ A[\mathbf{x},t-s^*(\mathbf{x},t)] \ . \tag{63}$$

For monotonous $\eta(s)$, we can calculate the explicit results $s^*(\mathbf{x},t) = -\eta^{-1}[h(\mathbf{x},t)-\vartheta]$ and $\mathrm{d}/\mathrm{d}t\, s^*(\mathbf{x},t) = h'(\mathbf{x},t)/\eta'[s^*(\mathbf{x},t)]$, [10] and get the simplified expression (Eggert & van Hemmen 2000)

$$A(\mathbf{x},t) = \left|1 + \frac{h'(\mathbf{x},t)}{\eta'[s^*(\mathbf{x},t)]}\right|\ A[\mathbf{x},t-s^*(\mathbf{x},t)] \ . \tag{64}$$

This equation can already be used for simulations and is exact for deterministic spike–response neurons with renewal. All it requires is to solve for $s^*(\mathbf{x},t)$, and to calculate $\eta'[s^*(\mathbf{x},t)]$ (analytically) and $h'(\mathbf{x},t)$ (using the ongoing simulation data) at every time step. Eq. (64) is causal because it is always $s^*(\mathbf{x},t) > 0$. Eq. (63) is called the **noise-free activity-folding model** (although no folding takes place in this particular case).

Now, we can look at low, but finite noise approximations of (36). We see from comparison of (36) and (63) that in the noise-free case the spiking probability function is $S_h(\mathbf{x},t,t^*) \propto \delta[(t-t^*) - s^*(\mathbf{x},t)]$, that is, its function is to *filter out the valid contributions* from the past activity $A(\mathbf{x},t^*)$, according to the threshold condition (62). For small but increasing noise level, $S_h(\mathbf{x},t,t^*)$ broadens to a gaussian-shaped function of t^* centered around $t - s^*(\mathbf{x},t)$. In this case, we approximate the real spiking probability function by $S(\mathbf{x},t,t^*) \propto \left(1/\sqrt{2\pi\sigma^2}\right) \exp\left\{-[(t-t^*) - s^*(\mathbf{x},t)]^2/(2\sigma^2)\right\}$.

Taking into account the normalization properties of $S_h(\mathbf{x},t,t^*)$ and assuming that σ remains small so that $S_h(\mathbf{x},t,t^*)$ still resembles a δ–distribution, we can then calculate an expression for the noisy activity so as to get

$$A(\mathbf{x},t) = \left|1 - \frac{\mathrm{d}}{\mathrm{d}t} s^*(\mathbf{x},t)\right|\ \tilde{A}[\mathbf{x},t-s^*(\mathbf{x},t)] \ , \tag{65}$$

or, in case of monotonous $\eta(s)$,

$$A(\mathbf{x},t) = \left|1 + \frac{h'(\mathbf{x},t)}{\eta'[s^*(\mathbf{x},t)]}\right|\ \tilde{A}[\mathbf{x},t-s^*(\mathbf{x},t)] \ , \tag{66}$$

[9] For monotonous $\eta(s)$, $s^*(\mathbf{x},t)$ can even be calculated explicitly to $s^*(\mathbf{x},t) = -\eta^{-1}[h(\mathbf{x},t) - \vartheta]$.
[10] Primes denote derivatives.

with $\tilde{A}(\mathbf{x}, t')$ being the past activity folded with a gaussian function around time t',

$$\tilde{A}(\mathbf{x}, t') := \int_{-\infty}^{t} \mathrm{d}t^* \, \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(t'-t^*)^2}{2\sigma^2}\right] A(\mathbf{x}, t^*) \ . \tag{67}$$

Eq. (65) is called the **noisy activity-folding model**.

From (65) resp. (66) we gain a new understanding of the activity dynamics. In the presence of noise, basically two processes compete with each other. On one hand, noise broadens the spiking probability function, causing a smearing out of the activity during the convolution when calculating $\tilde{A}(\mathbf{x}, t')$ in (67). On the other hand, the normalization factor contracts or expands the activity as in the noise-free case (63) resp. (64).

The unknown width σ in (67) has to be calculated at every simulation time step according to some approximation of the spiking probability function, since we do not know the exact $S_h(\mathbf{x}, t, t^*)$. A proposal of how this can be accomplished is presented in App. A.4.

Eq. (65) resp. (66) is formally equivalent to an integral equation derived using a gaussian noise at the reset after spiking (Gerstner 1998), instead of threshold noise as introduced in section 4.1.2. Because of the particular type of noise used, in Gerstner (1998), Eq. (65) is exact with a fixed σ. Although not widely spread in the neuroscience community, (65) resp. (66) allow for quantitative and qualitative modeling with much less numerical cost as compared to the main Eqs. (36) or (43).

An interesting property of (63) resp. (64) is that a linearization of $s^*(\mathbf{x}, t)$ around a time t^0 directly yields a dynamics which develops according to the **locking theorem** (Gerstner, van Hemmen, & Cowan 1996) for SRM neurons, see Eggert & van Hemmen (2000). This means that the dynamics will also be able to develop coherent oscillations as predicted by the locking theorem. The same linearization can also be done for (63) resp. (64), resulting in a locking theorem for noisy SRM neurons.

6.5 Standard Graded–Response Models, Wilson and Cowan Differential Equation Model

The commonly used **assembly-averaged graded–response** models, including the Wilson and Cowan **W+C graded–response** model (Wilson & Cowan 1972), can be motivated as follows from our pool dynamics. We look again at the normalized form $[A(\mathbf{x}, t) \to A(\mathbf{x}, t)/N(\mathbf{x})]$ of equation (43). In a quasistationary regime we define a dynamics by an exponential relaxation towards the stationary solution of (43), which e.g. for neurons with absolute refractory period only can be calculated solving for $A(\mathbf{x}, t)$ in $0 = -A(\mathbf{x}) + \{\tau[h(\mathbf{x})]\}^{-1}[1 - \gamma^{\mathrm{abs}} A(\mathbf{x})]$:

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{x}, t) = -A(\mathbf{x}, t) + \frac{1}{\tau[h(\mathbf{x}, t)]}[1 - \gamma^{\mathrm{abs}} A(\mathbf{x}, t)] \ . \tag{68}$$

This equation, and its simpler variant [for low activity $\gamma^{\mathrm{abs}} A(\mathbf{x}, t) \ll 1$, **standard graded–response** model]

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{x}, t) = -A(\mathbf{x}, t) + \frac{1}{\tau[h(\mathbf{x}, t)]} \ , \tag{69}$$

are of the same form as the widespread assembly-averaged graded–response models. Eq. 68 is formally identical to the Wilson and Cowan graded–response model.

In addition, using the stationary solutions of (43) for neurons including absolute and relative refractory period (see App.B.2 for a detailed calculation), by the same procedure we get

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{x},t) = -A(\mathbf{x},t) + \frac{1}{\tau[h(\mathbf{x},t)]}\{1 - [\gamma^{\mathrm{abs}} + \kappa_h^{(1)}(\mathbf{x},t)]A(\mathbf{x},t)]\} \tag{70}$$

with terms γ^{abs} and $\kappa_h^{(1)}(\mathbf{x},t)$ that account for *absolute and relative refractory effects*, respectively. Eq. (70) will relax towards the *correct microscopic solutions* (i.e., solutions that are in accordance with those obtained from simulations with single spiking neurons), incorporating absolute and relative refractory effects. There is no necessity of "time-coarse-graining" or other temporal averaging procedures to arrive at Eqs. (68), (69) or (70) for quasistationary activity. Graded–response models as in (68), (69) and (70) may thus present a valid approach, if the assembly dynamics are always close to the stationary state calculated from the microscopic parameters.

Again, as the preceding sections, it is now possible to understand how the parameters of the derived models (here: graded-response models) can be interpreted in terms of the microscopic parameters of the underlying neuronal model from section 4.1.1. The only exception is the *arbitrary relaxation time constant* τ. For a derivation of a first-order differential equation model without arbitrary time constants, we have to make use of the differential equation (53) or (56) in combination with appropriate approximations. This is done in App. B.3. Such a model will be better suited for the description of dynamics away from the case of stationary activity.

6.6 Graded–Response Models with Refractory Effects

We can enhance the standard graded response model (68), (69) or (70) by incorporating an additional term for the dynamics of the first recovery variable. Together with an exponential relaxation dynamics of $A(\mathbf{x},t)$, using a single differential equation from (56) for the dynamics of the number of inactivated neurons $N_{\mathrm{I}}(\mathbf{x},t)$ [here we break the chain of differential equations according to (115), see App. B.1 for systematic approximations], and assuming neurons with relative refractory period only, we find a **graded–response model with refractory effects**,

$$\begin{aligned} \tau \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{x},t) &= -A(\mathbf{x},t) + \frac{1}{\tau[h(\mathbf{x},t)]}[1 - N_{\mathrm{I}}(\mathbf{x},t)] \ , \\ \frac{\mathrm{d}}{\mathrm{d}t} N_{\mathrm{I}}(\mathbf{x},t) &= A(\mathbf{x},t) - \left[\frac{1}{\tau[h(\mathbf{x},t)]} + \frac{1}{\tau_{\mathrm{ref}}}\right] N_{\mathrm{I}}(\mathbf{x},t) \ . \end{aligned} \tag{71}$$

Integrating the spike density over a small fixed interval T during which $A(\mathbf{x},t)$ can be regarded as constant, we get the absolute number of neurons that released a spike recently, $f(\mathbf{x},t) \approx TA(\mathbf{x},t)$. We define further $r(\mathbf{x}) := N_{\mathrm{I}}(\mathbf{x},t)$, $\beta := 1/\tau_{\mathrm{ref}}$, $\sigma_1[h(\mathbf{x},t)] := 1/\tau[h(\mathbf{x},t)]T/\tau$, $\sigma_2[h(\mathbf{x},t)] := 1/\tau[h(\mathbf{x},t)]$, $\alpha_a = 1/\tau$ and $\alpha_r = 1/T$ and rewrite the equations (71) as

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\mathbf{x},t) = -\alpha_a f(\mathbf{x},t) + \{1 - r(\mathbf{x},t)\}\sigma_1[h(\mathbf{x},t)] \ ,$$

$$\frac{\mathrm{d}}{\mathrm{d}t} r(\mathbf{x},t) \quad = \quad \alpha_r f(\mathbf{x},t) - r(t)\{\beta + \sigma_2[h(\mathbf{x},t)]\} \ . \tag{72}$$

The result is a system that is very similar to the **Neural Network Master Equation (NNME)** proposed in Cowan (1991). Again, we can interprete the parameters of the model in terms of their microscopic parameters. The system (72) now depends as before in section 6.6 on non-intrinsic time constants, namely an arbitrary integration time constant T and an arbitrary relaxation time constant τ. For quantitative modeling away from the stationary state, it is therefore better to use the full assembly models presented in this paper, which are based exclusively on microscopic parameters.

6.7 Connection with Continuous Neural Field Models

The motivation for choosing pools as the basic units of biologically realistic neuronal networks was based on the experimental observation that, under identical stimulus conditions, groups of neurons often tend to behave similarly. However, a pool is not necessarily defined by *spatial proximity*, but by the input-output connection characteristics of its constituting neurons. One could imagine cases in which neurons have learned the same connectivity pattern and thus are members of the same neuronal pool, in spite of a considerable interneuronal distance.

In case of the input-output connection characteristics building a topology in real space, pools could be defined by neuronal types *and* spatial proximity. Then $\mathbf{x}$ does not designate a pool index but a point in space. If in a network volume (or surface, etc.) the coupling strengths vary only slowly as compared to the distance between neurons (that is, if the number of neurons is high in a volume in which the coupling strengths do not vary noticeably), we can take a continuum limit. Continuous "neural field" models are based on this type of description. Of course, the continuum limit can also be applied to the assembly models described in this paper. Let us consider a volume composed of a single type of neurons. We introduce a spatial neuronal density

$$\rho^{\mathrm{field}}(\mathbf{x}) = \sum_j \delta(\mathbf{x} - \mathbf{x}_j) \tag{73}$$

with the neuronal position vectors $\mathbf{x}_j$. To each *pool index* $\mathbf{x}$ we assign a neighborhood $\Omega(\mathbf{x})$ which comprises all neurons of the pool. That is to say, a neuron i belongs to a pool $\mathbf{x}$, if its position is $\mathbf{x}_i \in \Omega(\mathbf{x})$. The number of neurons in the neighborhood is then

$$N_\Omega(\mathbf{x}) = \int_{\Omega(\mathbf{x})} \mathrm{d}\mathbf{y}\, \rho^{\mathrm{field}}(\mathbf{y}) \ . \tag{74}$$

A spatial *activity density* $A'(\mathbf{x},t)$ is introduced by counting all spikes that occur in a small neighborhood $\Omega(\mathbf{x})$ at time t divided by the number of neurons in $\Omega(\mathbf{x})$.

$$A'(\mathbf{x},t) = \frac{\sum_{j\in\Omega(\mathbf{x})} S_j(t)}{\sum_{j\in\Omega(\mathbf{x})} 1} = \frac{A(\mathbf{x},t)}{N_\Omega(\mathbf{x})} \tag{75}$$

or, equivalently,

$$A(\mathbf{x},t) = \int_{\Omega(\mathbf{x})} \mathrm{d}\mathbf{y}\, \rho^{\mathrm{field}}(\mathbf{y})\, A'(\mathbf{y},t) \ . \tag{76}$$

All other extensive variables of the dynamics and pool-averaged quantities have to be calculated accordingly, since now we are dealing with spatial densities. Instead of the densities ρ, we now use $\rho' = \rho^{\text{field}}\rho$ to get

$$\langle \ldots \rangle'_{t^*}(\mathbf{x}, t) := \frac{\int_{\Omega(\mathbf{x})} \mathrm{d}\mathbf{y} \int_{-\infty}^{t} \mathrm{d}t^* \, \rho'(\mathbf{y}, t, t^*) \ \ldots}{N_\Omega(\mathbf{x})} \ . \tag{77}$$

Of course, the considerations of this section are not only valid for assembly dynamics calculated using integral equations in combination with densities ρ, but also for all other, derived models. Spatially continuous field equations often appear in relation with graded–response-like assembly averaged models, such as the Wilson and Cowan differential equation model of section 6.5. They have been used in a number of cases for the analysis of spatially homogeneous networks; see e.g. Ermentrout & Cowan 1979a; Ermentrout & Cowan 1979b; Ermentrout & Cowan 1980; Feldman & Cowan 1975; Wilson & Cowan 1973. Special interest has been evoked by the capabilities of such networks to produce spatiotemporal activity patterns. Spatially homogeneous networks of spiking neurons have also been studied recently in Kistler, Seitz, & van Hemmen 1998. So far, spatially homogeneous networks of assemblies of spiking neurons modeled at the macroscopical level such as presented in this work have not been studied yet, although they would allow to model a much larger biological network.

6.8 Connection with Models of Spiking Neurons

Eqs. (36) and (43) have been derived directly using neuronal properties and assembly averaging, so that an extensive $[N(\mathbf{x}) \gg 1]$ pool of singly modeled SRM neurons behaves in a way equivalent to the integral equations. (The case of finite-size pools is handled in section 6.10). The differential equation system (53) or (56) is also exact for pools of extensively many SRM neurons with renewal. However, it has been derived for special functions $p_{\mathrm{A}}(s)$ resp. $P_{\mathrm{A}}(s)$ (47). In other cases (i.e., for finite pool sizes or arbitrary refractory functions), the differential equation form of the assembly dynamics can be used as an approximation. In this section, we show how the parameters of the differential equation pool dynamics can be mapped to single-neuron parameters of the SRM and viceversa.

From section 4.1.3, Eqs. (31) and (32), we know the connection between the spike probability density for activated neurons $[\tau(h)]^{-1}$ and the activation probability $p_{\mathrm{A}}(s)$ with the SRM escape rate $\{\tau^{\mathrm{SRM}}[v(t)]\}^{-1}$ (28). For completeness, we repeat here Eq. (32) for $p_{\mathrm{A}}(s)$,

$$p_{\mathrm{A}}(s) := \exp[2\beta\eta(s)] \ , \tag{78}$$

which can be solved for $\eta(s)$ to get

$$\eta(s) = \frac{1}{2\beta} \ln[p_{\mathrm{A}}(s)] \ . \tag{79}$$

Figure 9 shows an example of an exponential $p_{\mathrm{A}}(s)$ and the corresponding SRM refractory function calculated through (79). We see that, when $p_{\mathrm{A}}(s) = 0$ (which is the case for very recent spiking so that $s < \gamma^{\mathrm{abs}}$, i.e., the neuron is in its absolute

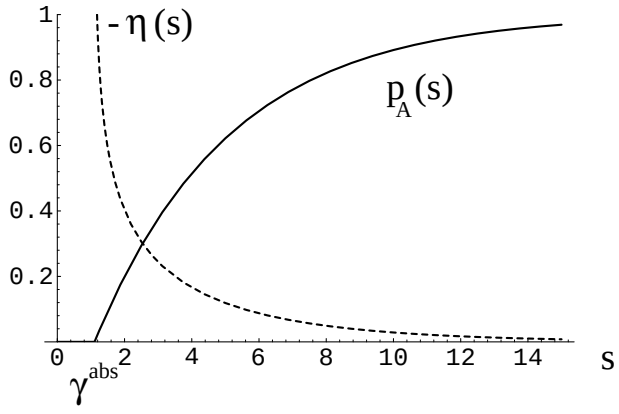

Figure 9: Correspondence between the activation function $p_{\rm A}(s)$ (solid line) and the negative refractory function $-\eta(s)$ of the SRM (dotted line). In this case, we used an exponential $p_{\rm A}(s)$ with an absolute refractory period of $\gamma^{\rm abs} = 1$. At $s = \gamma^{\rm abs}$, the refractory function $\eta(s)$ diverges to $-\infty$.

refractory phase), the negative refractory contribution to the membrane potential is infinitely high, so that the neuron is unable to release any spikes until $\eta(s)$ decreases in magnitude.

Alternatively, we can start from frequently used refractory functions $\eta(s)$ and search for systematic approximations of these function through the corresponding $p_{\rm A}(s)$ from (47). This is for example the case for integrate–and–fire neurons, which have an exponential $\eta(s) = \eta_{\rm exp}(s)$. This refractory function and another one that is frequently used for the SRM are indicated below,

$$\eta_{\rm exp}(s) = \begin{cases} 0 & \text{for } s < 0 \\ -\infty & \text{for } 0 \le s < \gamma^{\rm abs} \\ -\eta_0 \exp[-\frac{s-\gamma^{\rm abs}}{\tau_\eta}] & \text{for } s \ge \gamma^{\rm abs} \end{cases} \tag{80}$$

and

$$\eta_{\rm inv}(s) = \begin{cases} 0 & \text{for } s < 0 \\ -\infty & \text{for } 0 \le s < \gamma^{\rm abs} \\ -\frac{\tau_\eta}{s-\gamma^{\rm abs}} & \text{for } s \ge \gamma^{\rm abs} \end{cases} . \tag{81}$$

For small $\eta(s)$, i.e., in case that the synaptic field is small enough so that neurons do not spike again until their refractory field has already decreased considerably, we can approximate the activation function $p_{\rm A}(s)$ corresponding to the refractory function (80) by using (32)

$$\begin{aligned} p_{\rm A}(t-t^*) &= \exp[2\beta\eta(t-t^*)] \\ &\approx 1 + 2\beta\eta(t-t^*) = 1 - 2\beta\eta_0 \exp\left[-\frac{s-\gamma^{\rm abs}}{\tau_\eta}\right] . \end{aligned} \tag{82}$$

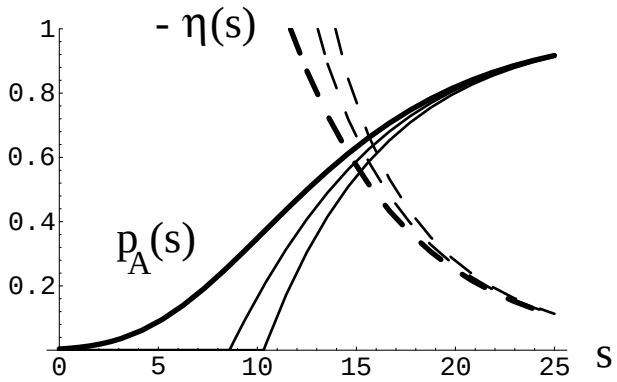

Figure 10: The exponential refractory field function $-\eta(s)$ (dashed thick line) is plotted together with its corresponding activation probability function $p_{\rm A}(s)$ (solid thick line). The other four functions are approximations of the desired refractory function (dashed thin line) and the desired activation function (solid thin line) using a sigmoidal (better fit of the thin curves) and an exponential $p_{\rm A}(s)$. This approximation is particularly suited for undercritical stimulation conditions, since the curves coincide for large s, i.e., when neurons spike again when their refractory field has already decreased noticeably.

Comparing this with the exponential activation function or the sigmoidal activation function in (47), we get

$$p_0 = 2\beta\eta_0,\ \tau_{\rm ref} = \tau_\eta,\ \text{and}\ s_0 = \gamma^{\rm abs}\ . \tag{83}$$

Similarly, in the case of the inverse refractory function (81) we can approximate

$$\begin{aligned} p_{\rm A}(t-t^*) &= \exp[2\beta\eta(t-t^*)] \\ &\approx 1 + 2\beta\eta(t-t^*) = 1 - 2\beta\frac{\tau_\eta}{s-\gamma^{\rm abs}} \end{aligned} \tag{84}$$

and compare this with the inverse activation function [last Eq. of (47)] so as to get

$$\tau_{\rm ref} = 2\beta\tau_\eta,\ \text{and}\ s_0 = \gamma^{\rm abs}\ . \tag{85}$$

Figure 10 shows an exponential refractory function as used for integrate–and–fire neurons and its approximation in terms of $p_{\rm A}(s)$. We see that, for large s, the curves coincide. This means that specially in undercritical synaptic driving conditions, during which the synaptic input is much smaller than the highest amplitude of the refractory field, the presented approximation scheme allows for a precise quantitative description of the activity of pools composed of stochastic SRM or integrate–and–fire neurons.

Of course any other approximation scheme can be used as well. This enables to simulate pools of neurons with different refractory fields by means of the differential equation model (53).

Finally, here is the cooking recipe to put the whole theory together for simulations of networks of pools of spiking neurons. First, choose one of the two noise models for

a stochastically spiking neuron as described in section 4.1.2, i.e., either the general threshold noise (27) or escape noise (33). Then, select the desired refractory function $\eta(s)$ or the corresponding activation function $p_{\mathrm{A}}(s)$ to describe the refractory properties of a neuron. After that, choose initial past activities $A(\mathbf{x}, t^*)$ and, using them, calculate the normalized state density $\rho(\mathbf{x}, t-\Delta t, t^*)$ of the assembly at time $t-\Delta t$. The activity $A(\mathbf{x}, t)$ at time t is then calculated using either (34) or (35) (according to the chosen noise model). The newly calculated activity $A(\mathbf{x}, t)$ modifies the density so that now we are able to calculate the new normalized state density $\rho(\mathbf{x}, t, t^*)$, and, accordingly, $A(\mathbf{x}, t+\Delta t)$, and so on. (For hints concerning the normalization of the density, see App. A). At the same time, keep track of the pool-to-pool interactions by calculating the synaptic fields according to Eq. (46).

Accordingly, we can use the differential equation assembly dynamics from section 5 for the calculation of the very same assembly dynamics, or any other of the approximation schemes presented in section 6. E.g., from the microscopic parameters of the noise model and the refractory function $\eta(s)$ or the corresponding activation function $p_{\mathrm{A}}(s)$, we can calculate the gain function of the pool, i.e., its steady firing vs. steady input relation (its stationary activity for constant input). How this is achieved is explained in section 6.1 for neurons with absolute refractory period only, and in section B.2 for neurons with absolute and relative refractory period (see also Fig. 12).

6.9 Connection with Single Neuron Spike Statistics

Poisson spike trains have an interspike interval (ISI) distribution that is exponential with a probability density (here, t is the interspike interval, and $\bar{t}$ is the mean interspike interval)

$$p(t) = \frac{1}{\bar{t}} \exp\left(-\frac{t}{\bar{t}}\right) \ . \tag{86}$$

Such spike trains are generated by a stochastic Poisson process which assumes a complete independence between the time of occurrence of neighboring spikes.

In a neuronal assembly, we do not average over different stochastic realizations of a single neuron, as it would be the case for gaining the probability density (86), but over all neurons that belong to the same pool. But since all pool neurons are equivalent, for the special case that all neurons fire together at some arbitrary moment in time, the assembly average and the realization average are also equivalent, which lets us calculate the ISI's of spiking neurons using the pool equations (36) or (43).

Wo.l.o.g., we assume that all neurons spike together at $t = 0$, so that $A(t^*) = \delta(t^*)$, and use this in (36) or (43). We get $A(\mathbf{x}, t) = \{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t)+\eta(t)]\}^{-1} D_h(\mathbf{x}, t, 0)$ or $A(\mathbf{x}, t) = \{\tau[h(\mathbf{x}, t)]\}^{-1} p_{\mathrm{A}}(t) D_h(\mathbf{x}, t, 0)$, which in the expanded form [using (37) or (38)] means

$$A(\mathbf{x}, t) = \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t) + \eta(t)]} \exp\left\{-\int_0^t \mathrm{d}t' \, \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t') + \eta(t')]}\right\} \tag{87}$$

or

$$A(\mathbf{x}, t) = \frac{1}{\tau[h(\mathbf{x}, t)]} p_{\mathrm{A}}(t) \exp\left\{-\int_0^t \mathrm{d}t' \, \frac{1}{\tau[h(\mathbf{x}, t')]} \, p_{\mathrm{A}}(t')\right\} \ . \tag{88}$$

Eqs. (87) and (88) correspond to an **inhomogeneous Poisson process** with time-varying mean interspike intervals $\bar{t} = \tau^{\mathrm{SRM}}[h(\mathbf{x},t)+\eta(t)]$ and $\bar{t} = \tau[h(\mathbf{x},t)]\{p_{\mathrm{A}}(t)\}^{-1}$. In case of constant synaptic field $h(\mathbf{x},t) = h(\mathbf{x})$, Eq. (88) is modified to

$$A(\mathbf{x},t) = \frac{1}{\tau[h(\mathbf{x})]} p_{\mathrm{A}}(t) \exp\left\{-\frac{t}{\tau[h(\mathbf{x})]}\,\Gamma_{\mathrm{A}}(t)\right\} \ , \tag{89}$$

which describes a Poisson process with mean interspike interval $\bar{t} = \tau[h(\mathbf{x})]$ and a modification of the ISI by refractory properties (**Poisson process with refractory properties**). These are taken into account by the functions $p_{\mathrm{A}}(t)$ and $\Gamma_{\mathrm{A}}(t)$, with $\Gamma_{\mathrm{A}}(t)$ calculated according to

$$\Gamma_{\mathrm{A}}(t) := \frac{\int_0^t \mathrm{d}t'\, p_{\mathrm{A}}(t')}{\int_0^t \mathrm{d}t'\, 1} \ \in [0,1] \ . \tag{90}$$

For a neuron with constant synaptic field and absolute refractory properties only, we get that $\Gamma_{\mathrm{A}}(t) = p_{\mathrm{A}}(t)$ and arrive at the Poisson ISI with offset (**Poisson process with offset**),

$$A(\mathbf{x},t) = \Theta(t-\gamma^{\mathrm{abs}})\,\frac{1}{\tau[h(\mathbf{x})]} \exp\left\{-\frac{t}{\tau[h(\mathbf{x})]}\right\} \ . \tag{91}$$

which, in case of zero γ^{abs}, finally reduces to the pure Poisson ISI of (86).

We conclude that the pool equations can be used to derive properties of the ISI statistics from single neuron models. They reduce to the known Poisson forms (86) or (91) for neurons without relative refractory properties. Furthermore, by fitting experimentally gained ISI's with (89) using the constant $\tau[h(\mathbf{x})]$ and the functions $p_{\mathrm{A}}(t)$ and $\Gamma_{\mathrm{A}}(t)$, it may be possible to analyze refractory properties of single neurons, described by $p_{\mathrm{A}}(t-t^*)$.

6.10 Finite-Size Assembly Dynamics

Eqs. (36), (43) or (53) are valid for pools with extensively many neurons. For finite-size pools, they will describe an assembly activity only to a certain approximation. However, they remain valid for the mean over different stochastic realizations. Using the central limit theorem (Lamperti 1966), it can be calculated that the expected number of spiking neurons, $\langle X(\mathbf{x},t)\rangle$, [11] in a time interval of length Δt around t is (Eggert & van Hemmen 2000)

$$\langle X(\mathbf{x},t)\rangle = A(\mathbf{x},t)\,\Delta t \tag{92}$$

with $A(\mathbf{x},t)$ calculated as before in (36), (43) or (53), so that our equations can still be used for calculating assembly dynamics. It should be noted, however, that it is also important to know the deviation σ of the results of the finite-size case from the

[11] Remark that now the braquets mean averaging over different stochastic realizations, and not assembly population averaging!

extensively calculated activity $A(\mathbf{x},t)$. For the deviation it can be shown that, using (43), it is (Eggert & van Hemmen 2000)

$$\begin{aligned}\sigma^2(\mathbf{x},t) &:= \left\langle (X(\mathbf{x},t) - \langle X(\mathbf{x},t)\rangle)^2 \right\rangle \\ &= A(\mathbf{x},t)\Delta t \left[1 - \frac{\Delta t}{\tau[h(\mathbf{x},t)]}\right] - \frac{\Delta t^2}{\tau[h(\mathbf{x},t)]^2}\left[N^{(1)}(\mathbf{x},t) - N^{(2)}(\mathbf{x},t)\right] ,\end{aligned} \tag{93}$$

with the recovery variables $N^{(1)}(\mathbf{x},t)$ and $N^{(2)}(\mathbf{x},t)$ as defined in (50).

We see that, for small discretization time intervals of length Δt [and neglecting the terms of the order $(\Delta t)^2$], we have a relative width of the probability distribution function of X that is equal to

$$\frac{\sigma(\mathbf{x},t)}{\langle X\rangle(\mathbf{x},t)} = \frac{1}{\sqrt{A(\mathbf{x},t)\Delta t}} . \tag{94}$$

For higher activity, the relative width decreases, so that the effect of noise induced by finite-size effects is reduced. If all neurons spike together at t, we get the minimal relative width of $\sqrt{1/N(\mathbf{x})}$.

Concluding, we can use the assembly dynamics (36), (43) or (53) for a calculation of the expected activity $\langle X(\mathbf{x},t)\rangle$ in simulations. For more realistic dynamics of finite-size assemblies, stochastic effects can then be included by assuming that the real number of spiking neurons $X(\mathbf{x},t)$ has a gaussian distribution around its mean $A(\mathbf{x},t)\,\Delta t$ with a width $\sigma(\mathbf{x},t)$ given by (93). This is a rather crude approximation but turns out to yield results that are in very good correspondence with the results gained from simulations using pools of singly modeled spiking neurons.

7 Discussion and Summary

In this work, we have presented a theoretical framework that describes in a quantitatively exact manner the dynamics of an assemblies of spiking neurons of the spike–response or integrate–and–fire type with renewal. The equations and their numerical implementation enable the efficient simulation of large, biologically realistic networks. The derivations are based on the notion of a static neuronal assembly, which takes the role of the basic computational unit of the network. Together with the assembly definition, we can move the descriptional level one level upwards from the single neuron, since the entire network can be described in terms of assemblies interacting by means of their *macroscopic* activities $A(\mathbf{x},t)$.

Since the derivations of the assembly dynamics do not rely upon temporal averaging, they serve to describe the dynamics of networks of spiking neurons with high temporal resolution. Assemblies react arbitrarily fast to rapid onsets of the input and are able to generate self-sustained coherent oscillations of activity. This has been confirmed in extensive simulations and theoretical work and can be read elsewhere (see e.g. Eggert & van Hemmen 2000; Gerstner 1998). The main message is: If a network of spiking neurons can be split into assemblies according to the definition of section 2.1, then simulations using assembly dynamics will be in quantitative accordance with the same network simulated with explicitly modeled neurons.

Figure 11 shows a diagram containing the most common models for spiking neurons and assembly dynamics used in the neuroscience community, embedded in a common framework with the models derived in this paper. The lower left corner indicates models of single spiking neurons. The upper left corner indicates models gained by assembly averaging, while the lower right corner shows models of single neurons gained by temporal averaging. The upper right corner finally shows models gained by both temporal and assembly averaging. These comprise the so-called assembly-averaged graded-response models and their variations.

The exact derivations and their conditions are indicated in Fig. 11 by solid arrows. Whereas standard assembly-averaged graded-response models cannot be "derived" exactly under any conditions, with the presented theory it is possible to establish a link between the upper left and the upper right sides of the diagram. This is accomplished by the differential equation form of the integral equation assembly dynamics. The numerical implementation of the differential equation form is indicated by an elongated dark rectangle that fully extends from left to right, indicating that a wide range of temporal averaging can be covered by this model. It is important to note here, however, that the temporal averaging is not performed during the derivation of the model, but instead a posteriori, when the numerical implementation takes place. This marks an important difference over previous models and derivations.

Finally, we see that a large number of known models and network effects can be understood within the presented theoretical framework. This understanding starts with assembly dynamics (e.g., the propagation of fast transients of activity, the propagation of pulse packets along a synfire–chain of assemblies, the generation of oscillatory activity in a group of neurons according to the locking theorem, etc.) and extends to the point that it enables us to investigate properties of single neurons (e.g., by calculation of exact gain functions including refractory effects, or by looking at deviations from Poisson ISI spiking statistics), since the assembly dynamics incorporate the microscopical parameters of the underlying neuronal model. The models, functions and analytical results that can be understood within our framework are the **logistic gain function**, **assembly-averaged gain functions** including absolute and refractory effects, the Wilson and Cowan **W+C integral equation model**, the synfire-chain **pulse-propagation model**, the **noise-free activity-folding model** and the **noisy activity-folding model**, the **assembly-averaged graded–response** models including the Wilson and Cowan **W+C graded–response** model and the **standard graded–response** model, **graded–response models with refractory effects** including the "Neural Network Master Equation" **(NNME)**, the **inhomogeneous Poisson process** including the **Poisson process with refractory properties** and the standard **Poisson process with offset**, and the influence of **finite-size effects on assembly dynamics**.

A series of appendices show details of the numerical implementation of the integral equation assembly dynamics (36) or (43), and of the differential equation assembly dynamics (53) or (56). They are intended for the interested assembly modeler for a straightforward implementation of the dynamics for simulations.

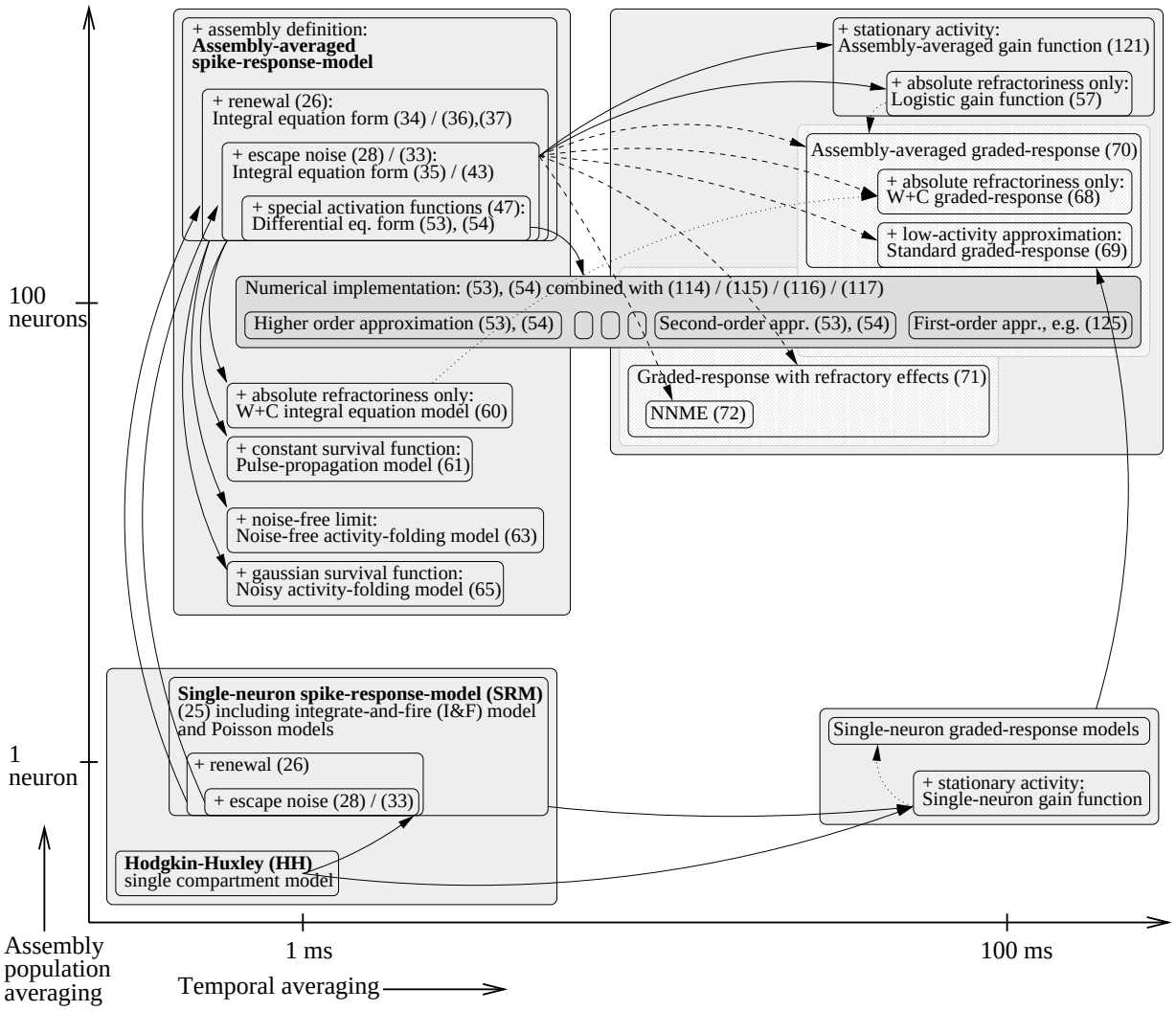

Figure 11: Overview of the different types of models and their derivations. The x and y achses indicate temporal averaging and assembly averaging. Braquets denote equation numbers. Derivations are indicated by arrows. Solid arrows indicate exact derivations under the condition added by a "+"; dashed arrows indicate "derivations" of dynamics by an exponential relaxation with arbitrary time constants, and dotted arrows indicate that the target model is motivated rather than derived by the source model. Four blocks of models can be divised (indicated by grey rectangles). In the lower left corner, we see models of single spiking neurons with high temporal resolution. From the HH model, the SRM formalism can be derived. From there, we can perform temporal or assembly population averaging. Population averaging results in the integral and differential equations for assembly dynamics (large block in the upper left). Standard assembly-averaged graded-response models are located in the upper right block (first and second order differential-equation models are indicated separatedly by two dashed subblocks). The differential equation model can be used to approximate assembly dynamics with low or high temporal precision, indicated by the elongated dark rectangle.

A Numerical Implementation of the Integral Equation Pool Dynamics

The integral Eqs. (36) or (43) suffice to describe the dynamics of an extensive pool of spiking neurons with renewal. They can be implemented to simulate the pool averaged activity in a quantitatively exact way. Here we discuss some numerical issues of the implementation of the equations for simulations. Two main points arise. First, we discuss the errors that are introduced into simulations because of limited storage resources, meaning a limited memory of the past history of the system. Second, we discuss how finite-size effects can be incorporated into our dynamics. This is important because biological neuronal assemblies are composed of a limited number of neurons (typically $\propto 10^2$).

A.1 Limited-Memory Error Effects

In principle, both for Eqs. (36) or (43), the complete density $\rho(\mathbf{x}, t, t^*)$ for all possible past spike times $t^* \leq t$ is needed. In a simulation, the entire density function for all t^* has to be kept in memory, so that we have to restrict ourselves to some past interval of length $s_{\max}$, with $t - s_{\max} \leq t^* \leq t$. This means that systematic errors will be introduced into the calculation of pool averaged variables at every simulation step. We denominate with hats all variables that are calculated with a truncated past interval $[t - s_{\max}, t]$. For example, instead of a fixed number of pool neurons $N(\mathbf{x})$ we get a time-varying quantity $\hat{N}(\mathbf{x}, t)$ with a **normalization error** $\epsilon(\mathbf{x}, t)$, defined by

$$\hat{N}(\mathbf{x}, t) := N(\mathbf{x})[1 - \epsilon(\mathbf{x}, t)] = \int_{t-s_{\max}}^{t} \mathrm{d}t^* \, \rho(\mathbf{x}, t, t^*) \; . \tag{95}$$

Analogously, in simulations we are dealing with the numerical equivalents $\hat{N}_{\mathrm{I}}(\mathbf{x}, t)$ and $\hat{A}(\mathbf{x}, t)$ of the number of inactivated neurons $N_{\mathrm{I}}(\mathbf{x}, t)$ and the activity $A(\mathbf{x}, t)$.

With these definitions, and linearizing for small interval lengths Δt, we can calculate from (95)

$$\begin{aligned} \hat{N}(\mathbf{x}, t + \Delta t) = \Delta t \hat{A}(\mathbf{x}, t + \Delta t) - \Delta t \rho(\mathbf{x}, t + \Delta t, t + \Delta t - s_{\max}) \\ + \int_{t-s_{\max}}^{t} \mathrm{d}t^* \, \rho(\mathbf{x}, t, t^*)[1 - \Delta t \{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t) + \eta(t - t^*)]\}^{-1}] \end{aligned} \tag{96}$$

for the SRM case [(34) or (36)] and

$$\begin{aligned} \hat{N}(\mathbf{x}, t + \Delta t) = \Delta t \hat{A}(\mathbf{x}, t + \Delta t) - \Delta t \rho(\mathbf{x}, t + \Delta t, t + \Delta t - s_{\max}) \\ + \int_{t-s_{\max}}^{t} \mathrm{d}t^* \, \rho(\mathbf{x}, t, t^*)[1 - \Delta t \{\tau[h(\mathbf{x}, t)]\}^{-1} p_{\mathrm{A}}(\mathbf{x}, t - t^*)] \end{aligned} \tag{97}$$

for the escape-noise case [(35) or (43)]. We get three terms. The first two account for neurons that enter and fall out of the limited time window, respectively.

In addition to the error due to limited memory, an error is introduced by the numerical integration of the dynamics. This error will not be considered further here because it depends on the implementation details.

A.2 Normalization Error

For the escape noise assembly dynamics from section 4.1.5, the main equation (43) reads in its memory-limited, discretized form:

$$\hat{A}(\mathbf{x}, t + \Delta t) = \frac{1}{\tau[h(\mathbf{x}, t)]} \left[N(\mathbf{x}) - \hat{N}_{\mathrm{I}}(\mathbf{x}, t) \right] . \tag{98}$$

Notice that here, we have written $N(\mathbf{x})$ instead of its truncated numerical equivalent, $\hat{N}(\mathbf{x}, t)$, since the pool size is known and does not need to be calculated over and over at every time step. This has important numerical implications, as will be seen when we analyze the implementation of the SRM assembly integral equation [(34) or (36)].

Using (98) in (97), we can calculate that

$$\begin{aligned} \hat{N}(\mathbf{x}, t + \Delta t) &= N(\mathbf{x})[1 - \epsilon(t + \Delta t)] = \\ &\quad -\Delta t \rho(\mathbf{x}, t + \Delta t, t + \Delta t - s_{\max}) \\ &\quad +N(\mathbf{x})\{1 - \epsilon(t)[1 - \Delta t\{\tau[h(\mathbf{x}, t)]\}^{-1}]\} + O(\Delta t^2) . \end{aligned} \tag{99}$$

Linearizing for small time intervals of length Δt and taking $\lim \Delta t \to 0$, this yields for the new normalization error:

$$\frac{\mathrm{d}}{\mathrm{d}t}\epsilon(t) = -\frac{1}{\tau[h(\mathbf{x}, t)]}\epsilon(t) + \frac{1}{N(\mathbf{x})}\rho(\mathbf{x}, t, t - s_{\max}) . \tag{100}$$

The numerical implementation of the dynamics thus produces two error terms. The second one comes from the limited past time window and quantifies the neurons $\rho(\mathbf{x}, t, t - s_{\max})$ that slide out of our time window and have been "forgotten" in the current time step. This contribution causes a permanent increase of $\epsilon(t)$. But there is another term which decreases the normalization error. This happens exponentially fast with a time constant $\tau[h(\mathbf{x}, t)]$, meaning that for increasing synaptic fields $h(\mathbf{x}, t)$ we will get $\hat{N}(\mathbf{x}, t) \to N(\mathbf{x})$. Therefore, the error becomes small for time periods of high synaptic input, which normally are the important time periods in simulations. The numerical implementation of the pool dynamics (98) thus performs an error correction through automatic normalization.

Repeating the same procedure for the memory limited implementation of (34) [or the corresponding (36)],

$$\hat{A}(\mathbf{x}, t + \Delta t) = \int_{t - s_{\max}}^{t} \mathrm{d}t^* \, \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t) + \eta(t - t^*)]} \rho(\mathbf{x}, t, t^*) , \tag{101}$$

we get for the number of pool neurons at time $t + \Delta t$

$$\hat{N}(\mathbf{x}, t + \Delta t) = \hat{N}(\mathbf{x}, t) - \rho(\mathbf{x}, t, t - s_{\max})\Delta t + O(\Delta t^2) . \tag{102}$$

Here, we have a problem, because there is no error correction through automatic normalization. Instead, the error accumulates and grows infinitely, i.e., the activity "drifts" away from any reasonable bounds. This is plain to understand, since (101)

is invariant to uniform shifts of the activity. But there is a solution to this problem. We calculate the new activity according to

$$\hat{A}(\mathbf{x}, t+\Delta t) = \frac{N(\mathbf{x})}{\hat{N}(\mathbf{x}, t)}\, \hat{A}_{\mathrm{old}}(\mathbf{x}, t+\Delta t) \approx \hat{A}_{\mathrm{old}}(\mathbf{x}, t+\Delta t)[1+\epsilon(t)] \tag{103}$$

for small normalization errors $\epsilon(t) \ll 1$ [$\hat{A}_{\mathrm{old}}$ is the activity as calculated with Eq. (101)].

Using (103) in (97), we can now calculate that

$$\begin{aligned}\hat{N}(\mathbf{x}, t+\Delta t) = N(\mathbf{x})[1-\epsilon(t+\Delta t)] = \\ -\Delta t \rho(\mathbf{x}, t+\Delta t, t+\Delta t - s_{\max}) \\ +N(\mathbf{x})\{1-\epsilon(t)[1-\Delta t \hat{A}_{\mathrm{old}}(\mathbf{x}, t)]\} + O(\Delta t^2)\end{aligned} \tag{104}$$

As before, linearizing for small time intervals of length Δt and taking $\lim \Delta t \to 0$ yields for the new normalization error:

$$\frac{\mathrm{d}}{\mathrm{d}t}\epsilon(t) = -\hat{A}_{\mathrm{old}}(\mathbf{x}, t)\epsilon(t) + \frac{1}{N(\mathbf{x})}\rho(\mathbf{x}, t, t-s_{\max})\ . \tag{105}$$

This means that now, a similar error correction and normalization as with (98) occurs. For clarity, we write down the numerical implementation of Eq. (103) in its full extension,

$$\hat{A}(\mathbf{x}, t+\Delta t) = N\, \frac{\int_{t-s_{\max}}^{t} \mathrm{d}t^*\, \{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t) + \eta(t-t^*)]\}^{-1} \rho(\mathbf{x}, t, t^*)}{\int_{t-s_{\max}}^{t} \mathrm{d}t^*\, \rho(\mathbf{x}, t, t^*)}\ , \tag{106}$$

with the density

$$\rho(\mathbf{x}, t, t^*) = \hat{A}(\mathbf{x}, t^*) \exp\left\{ -\int_{t^*}^{t} \mathrm{d}t'\, \frac{1}{\tau^{\mathrm{SRM}}[h(\mathbf{x}, t') + \eta(t'-t^*)]} \right\}\ . \tag{107}$$

We see that the synaptic field always appears as $h(\mathbf{x}, t)$ so that the numerical implementation is causally consistent. Using the past presynaptic activities of pools $\mathbf{y}$, $\hat{A}(\mathbf{y}, t^*)$, for $t^* \leq t$, we get the field $h(\mathbf{x}, t)$, and this in turn is used together with the stored own past activities $\hat{A}(\mathbf{x}, t^*)$ ($t^* \leq t$) and fields $h(\mathbf{x}, t')$ ($t' \in [t^*, t]$) to calculate the new activity $\hat{A}(\mathbf{x}, t+\Delta t)$.

A.3 Activity Error

Now that we know how to implement the integral equation pool dynamics (36) and (43) numerically, we can ask for the systematic error introduced into the activity by the truncation of the past memory. The calculation is straightforward. For escape noise, we use (98) and get

$$|A(\mathbf{x}, t+\Delta t) - \hat{A}(\mathbf{x}, t+\Delta t)| \leq \max_{t^* \in [-\infty, t-s_{\max}]} [1-p_{\mathrm{A}}(t-t^*)] \frac{1}{\tau[h(\mathbf{x}, t)]} N(\mathbf{x}) |\epsilon(t)|\ , \tag{108}$$

so that the activity error is bound by the normalization error $\epsilon(t)$.

Using (103), the same calculation for the SRM yields for escape noise

$$|A(\mathbf{x},t+\Delta t)-\hat{A}(\mathbf{x},t+\Delta t)| \leq \min_{t^*\in[t-s_{\max},t]}[1-p_{\mathrm{A}}(t-t^*)]\frac{1}{\tau[h(\mathbf{x},t)]}N(\mathbf{x})|\epsilon(t)| \ , \quad (109)$$

which is again bound by $\epsilon(t)$. For monotonous $p_{\mathrm{A}}(s)$, both expressions for the activity error are equivalent, namely

$$|A(\mathbf{x},t+\Delta t)-\hat{A}(\mathbf{x},t+\Delta t)| \leq [1-p_{\mathrm{A}}(s_{\max})]\frac{1}{\tau[h(\mathbf{x},t)]}N(\mathbf{x})|\epsilon(t)| \ . \quad (110)$$

A.4 Numerical Implementation of Activity-Folding Models

How can (65) resp. (66) be used for simulations? A further approximation is necessary to accomplish this. We still do not know σ. For simulations, we approximate the past synaptic field $h(\mathbf{x},t')$ needed by the survival function and the spiking probability function by a constant h [12]. In this case and for escape noise, we can calculate the survival function to

$$D_h(\mathbf{x},t,t^*) = \exp\left[-\frac{(t-t^*)}{\tau(h)}\Gamma_{\mathrm{A}}(t-t^*)\right] \ , \quad (111)$$

with

$$\Gamma_{\mathrm{A}}(s) := \frac{\int_0^s \mathrm{d}t' \, p_{\mathrm{A}}(t')}{\int_0^s \mathrm{d}t' \, 1} \ \in [0,1] \ . \quad (112)$$

A good choice is to select the constant synaptic field to be that of the maximum of the real spiking probability function, which is approximately located at $t-s^*(\mathbf{x},t)$. This finally results in

$$S_h(\mathbf{x},t,t^*) = \frac{1}{\tau[h(\mathbf{x},t-s^*(\mathbf{x},t))]}\, p_{\mathrm{A}}(t-t^*)\, \exp[-\frac{(t-t^*)}{\tau\{h[\mathbf{x},t-s^*(\mathbf{x},t)]\}}\Gamma_{\mathrm{A}}(t-t^*)] \ . \quad (113)$$

Since now we have an analytic expression for the spiking probability, from (113), we can calculate the momentary σ. The procedure is then as follows. At any moment t, calculate $s^*(\mathbf{x},t)$ using the actual synaptic field $h(\mathbf{x},t)$. Then, calculate σ as a function of the past synaptic field $h[\mathbf{x},t-s^*(\mathbf{x},t)]$, using (113). Finally, calculate the folded past activity $\tilde{A}[\mathbf{x},t-s^*(\mathbf{x},t)]$ using the calculated σ and $S_h(\mathbf{x},t,t^*)$. After that, proceed with the calculation as in the noise-free case using (66).

For a numerical implementation, the procedure can be simplified even further, by setting up a table of the σ's as a function of the past fields $h[\mathbf{x},t-s^*(\mathbf{x},t)]$. This reduces the calculation of the assembly dynamics to a single integral evaluation when calculating the folded past activity $\tilde{A}[\mathbf{x},t-s^*(\mathbf{x},t)]$.

[12] Of course, the same procedure described here can be applied using linear or higher-order approximations of h.

B Numerical Implementation of the Differential Equation Pool Dynamics

In this section, we are going to look in detail at the consequences that follow from using the system (53) for the calculation of assembly dynamics. Equations (53) are exact for assemblies composed of extensively many spiking neurons. Nevertheless, for a numerical implementation of the dynamics, the infinite chain of differential equations has to be approximated by a *finite* differential equation system. Breaking the chain earlier or later leads to dynamics that follows the exact results in a smooth fashion or in every detail. One can therefore approximate the pool dynamics with the desired accuracy. Here we discuss systematic approximations to the differential equation system.

B.1 Systematic Approximations

Contrary to previous work on pool dynamics, the differential equation system (53) is exact for pools of extensively many neurons, since it does not rely upon time-averaging for its derivation. This allows to quantitatively model pool activities well beyond the quasistationary regime. But for numerical simulations, the infinite chain of differential equations has to be approximated by a finite system.

Because of property (52) of the recovery variables, we can approximate the infinite chain of differential equations (53) by breaking it at a desired recovery variable $N^{(n+1)}(\mathbf{x},t)$ and by introducing an appropriate dynamics for this quantity. In this section, we will proceed to analyze different approximations of the differential equation system (53).

There are two sensible ways of approximating $N^{(n+1)}(\mathbf{x},t)$, which differ according to the desired dynamical simulation range. Assuming that n is large enough, the influence of the relative refractory field on the $(n+1)$-th recovery variable can be neglected, and we can approximate $N^{(n+1)}(\mathbf{x},t) \approx M(\mathbf{x},t)$; or $N^{(n+1)}(\mathbf{x},t) \approx 0$ if we are dealing with neurons without absolute refractory period.

- For *fast, transient dynamics* with sharp activity steps, $N^{(n+1)}(\mathbf{x},t)$ is then calculated according to the dynamics of $M(\mathbf{x},t)$,

$$\frac{\mathrm{d}N^{(n+1)}}{\mathrm{d}t}(\mathbf{x},t) \approx \frac{\mathrm{d}M}{\mathrm{d}t}(\mathbf{x},t) = A(\mathbf{x},t) - A(\mathbf{x},t-\gamma^{\mathrm{abs}}) \;, \tag{114}$$

  or, for neurons without absolute refractory period, we use equation (56) for $\mathrm{d}N^{(n)}(\mathbf{x},t)/\mathrm{d}t$ and $N^{(n+1)}(\mathbf{x},t) \approx 0$. In the case of exponential or sigmoidal $p_{\mathrm{A}}(s)$, this gives for the n'th recovery variable

$$\frac{\mathrm{d}N^{(n)}}{\mathrm{d}t}(\mathbf{x},t) \approx A(\mathbf{x},t) - \left[\frac{1}{\tau[h(\mathbf{x},t)]} + \frac{n}{\tau_{\mathrm{ref}}}\right] N^{(n)}(\mathbf{x},t) \;. \tag{115}$$

- For *slow* dynamics, we can approximate $N^{(n+1)}(\mathbf{x},t)$ by its stationary value. For slow dynamics, the activity $A(\mathbf{x},t)$ and the field $h(\mathbf{x},t)$ are approximately

constant during the time period of the kernel $[1 - p_{\rm A}(s)]^{n+1}$ of $N^{(n+1)}(\mathbf{x}, t)$. This means that

$$N^{(n+1)}(\mathbf{x}, t) \approx [\gamma^{\rm abs} + \kappa_h^{(n+1)}(\mathbf{x})] A(\mathbf{x}, t) \tag{116}$$

with $\kappa_h^{(n+1)}(\mathbf{x})$ being the time constant of the $(n+1)$-th kernel due to relative refractory effects,

$$\kappa_h^{(n+1)}(\mathbf{x}) = \int_{\gamma^{\rm abs}}^{\infty} {\rm d}s \, [1 - p_{\rm A}(s)]^{n+1} \, D_h(\mathbf{x}, t, t-s) \; . \tag{117}$$

Here $\kappa_h^{(n+1)}(\mathbf{x})$ has been evaluated for a quasistationary field $h(\mathbf{x}, t)$ (i.e., the field is assumed to be constant for a period during which the expression in the integral is large), and, thus, it is written without an explicit dependency on t. It depends, however, on the field h. For neurons with *absolute refractory period only*, $\kappa_h^{(n+1)}(\mathbf{x}) = 0$ and we can use $N^{(n+1)}(\mathbf{x}, t) \approx \gamma^{\rm abs} A(\mathbf{x}, t)$ as the stationary approximation. Similarly, in case of pools with *relative refractory period only*, we use $N^{(n+1)}(\mathbf{x}, t) \approx \kappa_h^{(n+1)}(\mathbf{x}) A(\mathbf{x}, t)$.
The slow approximation is exact when the activity approaches a stationary value.

B.2 Zeroth-Order Approximation: Stationary Solution and Gain Function

For constant input field $h(\mathbf{x}, t) \equiv h(\mathbf{x})$ and stationary activity $A(\mathbf{x}, t) \equiv A(\mathbf{x})$, we can start directly with Eq. (43),

$$A(\mathbf{x}) = \frac{1}{\tau[h(\mathbf{x})]} [N(\mathbf{x}) - N_{\rm I}(\mathbf{x})] \; . \tag{118}$$

Using this equation, and the expression (116) for $N_{\rm I}(\mathbf{x}) = N^{(1)}(\mathbf{x})$ in the stationary case,

$$N_{\rm I}(\mathbf{x}) = [\gamma^{\rm abs} + \kappa_h^{(1)}(\mathbf{x})] A(\mathbf{x}) \; , \tag{119}$$

we can calculate the normalized stationary spike density (the exact **assembly-averaged gain function**) to

$$G[\mathbf{x}, h(\mathbf{x})] := \frac{A(\mathbf{x})}{N(\mathbf{x})} = \frac{1}{\gamma^{\rm abs} + \tau[h(\mathbf{x})] + \kappa_h^{(1)}(\mathbf{x})} \; . \tag{120}$$

If $\tau(h)$ is a monotonously decreasing function of h, we get a *gain function* $G(h)$ that saturates at large h and has a sigmoidal-like appearance.

Using our escape noise Ansatz from (31), $\tau(h) = \tau_0 \exp[-2\beta(h - \vartheta)]$, with spike-rate at threshold τ_0^{-1}, noise parameter β, and threshold ϑ, we get

$$G[\mathbf{x}, h(\mathbf{x})] = \frac{1}{\gamma^{\rm abs}} \frac{1}{1 + \exp\{-2\beta[h(\mathbf{x}) - \vartheta']\} + \kappa_h^{(1)}(\mathbf{x})/\gamma^{\rm abs}} \tag{121}$$

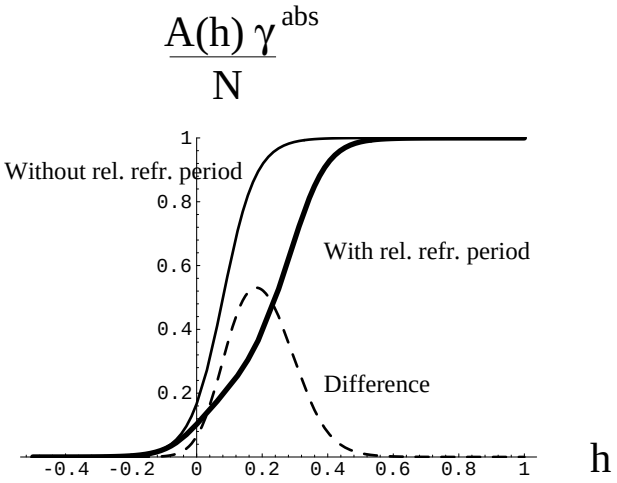

Figure 12: The stationary spike density for a pool of spiking neurons which receives a constant synaptic input field h can be expressed by a gain function that is similar to the logistic gain function. Here we show the gain function for a pool of neurons with absolute refractory period only (thin solid line) and for a pool with absolute and relative refractory period (thick solid line). The relative refractory period reduces the activity for fields h close to the threshold (dashed line, difference between the gain function without and the gain function including relative refractory effects).

with the modified threshold

$$\vartheta' = \vartheta + 1/(2\beta)\ln(\tau_0/\gamma^{\text{abs}}) \tag{122}$$

and the relative refractory period time constant $\kappa_h^{(1)}(\mathbf{x})$ as defined in Eq. (117).

Figure 12 shows the stationary spike density $A(\mathbf{x})$ as a function of the synaptic field $h(\mathbf{x})$. The pool spike rate $A(\mathbf{x})$ saturates at $N(\mathbf{x})/\gamma^{\text{abs}}$ as it is bounded by the inverse length of the *absolute* refractory period. The time constant $\kappa_h^{(1)}(\mathbf{x})$ quantifies the influence of the *relative* refractory period on the stationary pool spike rate. It reduces the activity for intermediate fields $h(\mathbf{x}) \approx \vartheta'$. The noise factor β and the effective threshold ϑ' determine the slope and the inflection point of the gain function. Increasing the length of the absolute refractory period γ^{abs} or decreasing the firing rate at threshold τ_0^{-1} shifts the effective threshold towards higher values. We see that our assembly dynamics let us understand the gain function quantitatively in terms of the *microscopic neuronal parameters* γ^{abs}, $\kappa_h^{(1)}$, τ_0, β, and ϑ. This marks a difference to standard gain functions as those commonly used in graded–response pool models.

B.3 First-Order Approximation: Quasistationary Dynamics and Graded–Response

The graded–response models presented in section 6.5 have a serious disadvantage: they have free dynamical parameters which can be chosen at will, such as the time

constant τ in (68), (69) or (70). Of course the free parameters could be fitted, but still it would be difficult to interprete the data, because the free parameters stem from the exponential relaxation dynamics or from temporal averaging over arbitrary time intervals, and not from the microscopic properties of the neurons.

Here we move in the opposite way. We start from our main Eqs. (53) and derive a closed expression for the simplest possible assembly dynamics. This results in a graded–response-like relaxation dynamics [i.e., a first-order differential equation for $A(\mathbf{x},t)$] that follows smoothly and coarsely the real dynamics of the assembly. Additionally, all its parameters can be interpreted in terms of the microscopic neuronal parameters.

Since we want to gain a relaxation dynamics without delays, we assume that there is no discontinuity in the activation function $p_{\mathrm{A}}(s)$, i.e., the neurons have a relative refractory behavior but no absolute refractory period. We start with a first-order approximation of our main Eqs. (56). We use the slow-dynamics approximation (116) [13], and chop the differential equation system at $n=1$, so that

$$N^{(2)}(\mathbf{x},t) \approx \kappa_h^{(2)}(\mathbf{x}) A(\mathbf{x},t) \tag{123}$$

with $\kappa_h^{(2)}(\mathbf{x})$ calculated as specified in (117) [depending on $h(\mathbf{x},t)$]. This means that we only have two state variables, namely the activity $A(\mathbf{x},t)$ and the first recovery variable $N^{(1)}(\mathbf{x},t) = N_{\mathrm{I}}(\mathbf{x},t)$, which is the number of neurons in the inactivated state.

The number of inactivated neurons is obtained from $A(\mathbf{x},t) = \{\tau[h(\mathbf{x},t)]\}^{-1}[N - N_{\mathrm{I}}(\mathbf{x},t)]$ so as to give

$$N_{\mathrm{I}}(\mathbf{x},t) = N(\mathbf{x}) - \tau[h(\mathbf{x},t)] A(\mathbf{x},t) \ . \tag{124}$$

We now turn to the activity $A(\mathbf{x},t)$. We assume that, for quasistationary activity, the fields evolve more slowly than the activity, and neglect the changes of $h(\mathbf{x},t)$ [14]. This leaves us with $\mathrm{d}A(\mathbf{x},t)/\mathrm{d}t \approx -\{\tau[h(\mathbf{x},t)]\}^{-1}\,\mathrm{d}N_{\mathrm{I}}(\mathbf{x},t)/\mathrm{d}t$, and inserting Eqs. (123) and (124) into the dynamics equation (56) for $m=1$ and exponential $p_{\mathrm{A}}(s)$ [the same steps can be applied to the other functions $p_{\mathrm{A}}(s)$], we can solve for $A(\mathbf{x},t)$ and arrive at

$$\begin{aligned} \tau_{\mathrm{g}}[h(\mathbf{x},t)]\,\frac{\mathrm{d}}{\mathrm{d}t}A(\mathbf{x},t) &= -A(\mathbf{x},t) \\ &+\frac{1}{\tau[h(\mathbf{x},t)]}\left\{N(\mathbf{x}) - \tau_{\mathrm{g}}[h(\mathbf{x},t)]\left[1 - \frac{\kappa_h^{(2)}(\mathbf{x})}{\tau[h(\mathbf{x},t)]}\right] A(\mathbf{x},t)\right\} \end{aligned} \tag{125}$$

[13] The slow dynamics approximation now involves temporal averaging. The difference to standard graded–response models is that the averaging occurs over intrinsic neuronal time intervals, and not a priori over some arbitrary interval of length T. Therefore, it does not introduce additional dynamical parameters. Moreover, since here we are looking at slow, or even quasistationary dynamics, the temporal averaging is justified. We also remind that we can avoid temporal averaging completely if we use the fast dynamics approximation, resulting in a first order approximation that has the form of a differential equation with delay.

[14] Wo.l.o.g., we can include a term that considers the variation of $A(\mathbf{x},t)$ due to changes of $h(\mathbf{x},t)$, so that this assumption is not really necessary for the calculation of the dynamics. It is omitted only to gain an equation that can be compared to other graded–response models.

with

$$\frac{1}{\tau_{\mathrm{g}}[h(\mathbf{x},t)]} = \frac{1}{\tau[h(\mathbf{x},t)]} + \frac{1}{\tau_{\mathrm{ref}}} \,. \tag{126}$$

Not only has this graded–response type equation the microscopically correct stationary solutions but it also provides us with the relaxation time constant $\tau_{\mathrm{g}}[h(\mathbf{x},t)]$. This means that if we are interested in a realistic quasistationary pool behavior, graded–response equations with fixed relaxation time constants as those of sections 6.5 and 6.6are not sufficient.

Equation (125) is the correct way to introduce a systematically derived graded–response-type dynamics for pools of spiking neurons using the chain of differential equations. The effect of (125) is a dynamics that follows the real activity dynamics by smoothing out sharp activity peaks. Nevertheless, it will do so following the envelope curve of the activity, and it will still approach the correct stationary solutions for a constant field $h(\mathbf{x})$.

B.4 Higher-Order Approximations: Realistic Assembly Dynamics

Higher-order approximations serve to model in a quantitatively accurate way the dynamics of assemblies of spiking neurons. Using the fast dynamics approximation (114), the differential equation model is capable of reproducing the time-course of the activity of a pool composed of extensively many neurons, including fast transients and sharp activity peaks as those occurring when the activity approaches oscillatory solutions.

The different recovery variables serve as memory buffers for the past activity. Higher (with larger n) recovery variables are responsible for the more recent past and influence the response of the pool to fast transients. Taking only one or two recovery variables results in activities that follow the real activity in a smooth, approximated way. If we include more recovery variables, the assembly dynamics also follows the finer details of the real activity. It is therefore possible to control the temporal accuracy of simulations, as well as the numerical cost.

Acknowledgements

We gratefully thank Bruce W. Knight for valuable comments and suggestions on this article.

References

Cowan, J. D. (1991). -. Volume 3 of Advances in Neural Information Processing Systems, San Mateo, pp. 62. Morgan Kaufmann.

Eggert, J. & van Hemmen, J. L. (2000). Unifying framework for neuronal assembly dynamics. Phys. Rev. E 61(2), 1855–1874.

Ermentrout, G. B. & Cowan, J. D. (1979a). A mathematical theory of visual hallucination patterns. Biol. Cybern., 34, 137–150.

Ermentrout, G. B. & Cowan, J. D. (1979b). Temporal oscillations in neuronal nets. J. Math. Biol., 7, 265–280.

Ermentrout, G. B. & Cowan, J. D. (1980). Large scale spatially organized activity in neural nets. SIAM J. Appl. Math., 38, 1–21.

Fargue, D. (1973). Réductibilité des systèmes héréditaires à des systèmes dynamiques. Compt. Rend. Acad. Sci. B(277), 471–473.

Fargue, D. (1974). Réductibilité des systèmes héréditaires. Int. J. of Non–Linear Mechanics, 9, 331–338.

Feldman, J. L. & Cowan, J. D. (1975). Large-scale activity in neural nets I: Theory with application to motoneuron pool responses. Biol. Cybern., 17, 29–38.

Gerstner, W. (1990). Associative memory in a network of 'biological' neurons. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), Advances in Neural Information Processing Systems 3, San Mateo CA, pp. 84–90. Morgan Kaufmann Publishers.

Gerstner, W. (1995). Time structure of the activity in neural network models. Phys. Rev. E, 51, 738–758.

Gerstner, W. (1998). Populations of spiking neurons. In W. Maass & C. M. Bishop (Eds.), Pulsed Neural Nets, Chapter 10, pp. 261–296. Cambridge, MA: MIT Press.

Gerstner, W. & van Hemmen, J. L. (1992). Associative memory in a network of 'spiking' neurons. Network, 3, 139–164.

Gerstner, W. & van Hemmen, J. L. (1994). Coding and information processing in neural networks. In E. Domany, J. L. van Hemmen, & K. Schulten (Eds.), Models of Neural Networks II, Berlin, pp. 1–93. Springer.

Gerstner, W., van Hemmen, J. L., & Cowan, J. D. (1996). What matters in neuronal locking? Neural Comput. 8(8), 1653–1676.

Gewaltig, M.-O. (1999, December). Evolution of synchronous spike volleys in cortical networks – Network simulations and continuous probabilistic models. Ph. D. thesis, Fakultät für Physik und Astronomie, Ruhr-Universität Bochum, 44780 Bochum, Germany.

Kistler, W. M., Gerstner, W., & van Hemmen, J. L. (1997). Reduction of the Hodgkin-Huxley equations to a single-variable threshold model. Neural Comput., 9, 1015–1045.

Kistler, W. M., Seitz, R., & van Hemmen, J. L. (1998). Modelling collective excitations in cortical tissue. Physica D, 114, 273–295.

Knight, B. W. (2000). Dynamics of Encoding in Neuron Populations: Some General Mathematical Features. Neural Comput., 12, 473–518.

Knight, B. W., Omurtag, A., & Sirovich, L. (2000). The approach of a neuron population firing rate to a new equilibrium: an exact theoretical result. Neural Computation, 12, 1045–1055.

Lamperti, J. (1966). Probability. New York: Benjamin. Chapter 7.

Mainen, Z. F. & Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. Science, 268, 1503–1505.

Nykamp, D. Q. & Tranchina, D. (2000). A pouplation density approach that facilitates large-scale modeling of neural networks: Analysis and an application to orientation tuning. Journal of Computational Neuroscience.

Omurtag, A., Knight, B. W., & Sirovich, L. (2000). On the simulation of large populations of neurons. J. Comput. Neurosci., 8, 51–63.

Sirovich, L., Knight, B. W., & Omurtag, A. (2000). Dynamics of neuronal populations: The equilibrium solution. SIAM J. of Applied Math. to appear, Dec.2000.

Tuckwell, H. C. (1988). Introduction to Theoretical Neurobiology. Cambridge: Cambridge University Press.

Wilson, H. R. & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. Biophys. J., 12, 1–24.

Wilson, H. R. & Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. Kybernetik, 13, 55–80.