

# From spiking neurons to rate models: A cascade model as an approximation to spiking neuron models with refractoriness

Yuval Aviel and Wulfram Gerstner\*

*Ecole Polytechnique Fédérale de Lausanne (EPFL), School of Computer and Communication Sciences and Brain Mind Institute,  
CH 1015 Lausanne, Switzerland*

(Received 26 October 2005; published 16 May 2006)

A neuron that is stimulated repeatedly by the same time-dependent stimulus exhibits slightly different spike timing at each trial. We compared the exact solution of the time-dependent firing rate for a stochastically spiking neuron model with refractoriness (spike response model) with that of an inhomogeneous Poisson process subject to the same stimulus. To arrive at a mapping between the two models we used alternatively (i) a systematic parameter-free Volterra expansion of the exact solution or (ii) a linear filter combined with nonlinear Poisson rate model (linear-nonlinear Poisson cascade model) with a single free parameter. Both the cascade model and the second-order Volterra model showed excellent agreement with the exact rate dynamics of the spiking neuron model with refractoriness even for strong and rapidly changing input. Cascade rate models are widely used in systems neuroscience. Our method could help to connect experimental rate measurements to the theory of spiking neurons.

DOI: [10.1103/PhysRevE.73.051908](https://doi.org/10.1103/PhysRevE.73.051908)

PACS number(s): 87.19.La, 05.10.Gg

## I. INTRODUCTION

Descriptions of neuronal activity range from detailed biophysical neuron models [1] to formal neurons used in artificial neural networks [2]. Somewhere in between these extremes lies the class of formal spiking neurons [3] such as the integrate-and-fire neuron [4]. In the presence of noise, formal spiking neuron models show a high variability of action potential firing [5] with realistically looking interval distributions and coefficients of variation. The transition from stochastically spiking integrate-and-fire models (i.e., a time-dependent renewal point process) to stochastically spiking rate models (i.e., an inhomogeneous Poisson point process) looks, at first sight, rather innocent: Such a transition amounts to neglecting all spike-after effects, in particular refractoriness. However, it is well known that important differences exist between Poisson processes and spiking neurons of the integrate-and-fire type. For example, a neuron model based on linear signal integration followed by nonlinear stochastic spike generation [a linear-nonlinear-Poisson (LNP) cascade model] shows a reverse correlation function which directly reflects the linear filtering stage [6,7] whereas linear signal integration followed by a threshold-and-reset mechanism as in integrate-and-fire neurons shows a reverse correlation function which includes components of refractoriness [8–10].

The question of the appropriate level of model is intimately linked to the problem of neural coding—i.e., spike coding versus rate coding [3,11,12]. A pragmatic experimental way of defining a time-dependent firing rate of a neuron is by building up a peri-stimulus time histogram (PSTH) across several repetitions of the same time-dependent stimulus. Such a time-dependent rate (or PSTH density) is then the natural starting point for an interpretation in terms of firing

rate coding. If a rate coding picture is true, the PSTH density could be seen as the firing rate of an inhomogeneous Poisson process. However, such an interpretation misses the correlations between spikes in a single spike train generated by neuronal refractoriness [13,14], among other factors such as adaptation.

In this paper we discuss the transition between spiking neuron models and Poisson rate models in more detail and focus on the role of refractoriness in the PSTH. As a starting point, we exploit the fact that, for formal spiking neuron models such as the leaky integrate-and-fire model, an exact transition between the PSTH and underlying single-neuron dynamics is known. Depending on the details of the models and the type of noise, this relation can be derived in the framework either of partial differential equations [15–17] or of an integral equation [18,19]; see also [20]. For practical reasons we will in the following consider the integral equation of the PSTH density derived from a spike response model with escape noise [19].

In order to transform the integral equation, which is rather cumbersome to interpret, into a rate model of a more standard form we proceed in several steps. First, we use a systematic expansion of the integral equation into a Volterra series. The first-order term of the series is a linear filter which gives the reverse correlation function and PSTH density in the small-signal limit as discussed in earlier studies [8,21]. The second-order term is also discussed in this paper.

Second, and as an alternative to the systematic series expansion to higher orders, we also explore the possibility of truncating the Volterra series after the first order and approximating the remaining terms by an *ad hoc* nonlinear “squashing” function. The interest of such an approach lies in the fact that it leads directly to the standard LNP cascade model as widely used in systems of neuroscience, in particular vision [6,7]. By construction, the LNP cascade model derived by our method is correct to first order. The advantages and shortcomings of such an approach compared to the full so-

\*Electronic address: [wulfram.gerstner@epfl.ch](mailto:wulfram.gerstner@epfl.ch)

lution by an integral equation and the alternative Volterra expansion approach are studied in simulations for various stimulation paradigms.

The paper is organized as follows. The spiking neuron model we use throughout the paper is discussed in Sec. II. In Sec. III we review the exact solution of the time-dependent PSTH density. The evolution of the PSTH density will be presented in the form of an integral equation derived in earlier studies [3,18,19]. In Sec. IV we present the systematic Volterra expansion of the equation and in Sec. V the alternative cascade model (CM) approach. Both approaches are compared on a couple of stimulation paradigms. Finally, in Sec. VI, we briefly discuss the connection of our approach to earlier studies on the relation between spike and effective rate models.

## II. NEURON MODEL

We use a spike response model with escape noise [3]. The membrane potential is given by

$$u(t|\hat{t}) = \eta(t - \hat{t}) + \int_0^\infty \kappa(s)I(t-s)ds, \quad (1)$$

where  $I(t)$  is a time-dependent input current,  $\kappa$  characterizes the passive membrane properties, and  $\eta$  implements a partial reset of the membrane potential after each spike. The firing time of the last spike is denoted by  $\hat{t}$ . For the sake of simplicity we take simple exponentials with membrane time constant  $\tau_m$  for both  $\eta$  and  $\kappa$ , thus  $\kappa(s) \stackrel{\text{def}}{=} \tau_m^{-1} \exp(-s/\tau_m)$  and  $\eta(s) \stackrel{\text{def}}{=} -\eta_0 \exp(-s/\tau_m)$  for  $s > 0$  (for  $s \leq 0$  the two functions vanish due to causality). The parameter  $\eta_0$  controls the amount of reset after a spike. With this choice of parameters the spike response model is closely related, but not identical, to the integrate-and-fire model; for details see [3]. Note that  $\eta(s)$  implements a form of refractoriness that depends only on the most recent spike in the past. In particular, the above model does not account for adaptation.

We assume that the neuron is under the influence of a certain amount of noise. Spike firing is hence stochastic and governed by an instantaneous firing rate  $f$  that depends on the momentary distance from the threshold  $\vartheta$ . For the sake of simplicity we take a simple Arrhenius escape model [22]

$$f(u) = \frac{C}{\tau_m \sigma} \exp\left[-\frac{(u - \vartheta)^2}{\sigma^2}\right], \quad (2)$$

where  $(u - \vartheta)$  is the distance between the momentary membrane voltage and the formal threshold. Intuitively, a spike can be triggered if the noise is strong enough to overcome this distance. The strength of the noise is characterized by a parameter  $\sigma$ . We note that in the low-noise limit ( $\sigma \rightarrow 0$ ), firing occurs only if the membrane potential reaches threshold. The relation of such an Arrhenius escape noise to diffusive noise in the input as could be generated by stochastic spike arrival has been discussed in [22]. Other choices of the escape function  $f$  can be found in [3,21,22]. Essentially, the Arrhenius escape function is a reasonable approximation

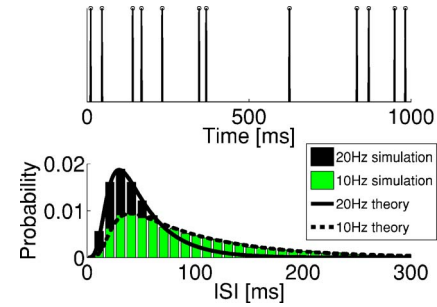


FIG. 1. (Color online) Model spike trains. Upper panel: an example of a spike train generated by a spike response model. The neuron fires at a mean rate of 10 Hz. Lower panel: the inter-spike-interval histogram (bars) and the theoretical distribution  $P_0$  (lines). Both the histograms and theory are plotted for 10 Hz (gray bars, dashed line) and 20 Hz (black bars, solid line). The theory agrees very well with the simulations. The distributions vanish for small intervals due to refractoriness. Also, a higher output rate yields a narrower distribution, leading to more regular spike trains.

only in the subthreshold regime—i.e., if the membrane potential is far away from threshold. The scaling with  $\sigma^{-1}$  has been introduced as an *ad hoc* factor in order to allow us to compare the behavior at high- and low-noise levels.

Our standard set of parameters is  $\tau_m = 10$  ms,  $C = 1$ ,  $\sigma = 1$ ,  $\vartheta = 3$ , and  $\eta_0 = 1$ . A typical simulation run with a constant stimulus of  $I(t) = 1.5$  or  $I(t) = 2$  is shown in Fig. 1. During each time step  $[t, t + \Delta t]$  of the simulation, a spike is generated with probability  $p = 1 - \exp[-f(u(t|\hat{t}))\Delta t]$ . For  $\Delta t \rightarrow 0$  the probability reduces to  $f(u(t|\hat{t}))\Delta t$  as it should since  $f$  denotes the instantaneous rate. The formula for  $p$  allows us to work with finite time steps  $\Delta t$ . In the simulation of Fig. 1 we have used  $\Delta t = 1$  ms and the bin size of the PSTH was set to 10 ms. The neurons in Fig. 1 fire at a mean firing rate of approximately 10 Hz or 20 Hz and show a broad distribution  $P(s)$  of interspike intervals. Effects of refractoriness are clearly visible in both distributions.

The advantage of the escape noise model is that the inter-spike interval distribution can be calculated analytically using standard expressions from renewal theory [23]. We introduce the instantaneous firing rate  $\rho(s)$  of a neuron which has fired its last spike at  $s = 0$ —i.e.,  $\rho(s) = f(u(\hat{t} + s|\hat{t}))$  where  $u(\hat{t} + s|\hat{t}) = \eta(s) + \int \kappa(s')I(t - s')ds'$ . By definition, the interval distribution  $P(s)$  gives the probability density of finding an interval of length  $s$ . This is equivalent to the probability of firing at time  $s$  given a spike at time  $s = 0$  and not firing in between. Hence,

$$P(s) = \rho(s) \exp\left[-\int_0^s \rho(s')ds'\right]; \quad (3)$$

the term  $\rho(s)$  accounts for the instantaneous rate at time  $s$  and the exponential factor gives the probability of not firing between 0 and  $s$ ; see [23] or Sec. 5.2.3 of [3].

An extension of the above reasoning, we can also consider the survivor function  $S_h(t|\hat{t})$ —i.e., the probability of neuron to “survive” without firing from  $\hat{t}$  to  $t$  after having fired a spike at  $\hat{t}$ . For *constant* input current, the survivor

function is related to the interval distribution by

$$S_h(t|\hat{t}) = 1 - \int_0^{t-\hat{t}} P(s) ds = \exp\left[-\int_{\hat{t}}^t \rho(s') ds'\right], \quad (4)$$

which yields  $P(s) = -\frac{d}{ds} S_h(\hat{t} + s|\hat{t})$ . In the general case of *time-dependent* input current, the time course of the survivor function depends on the time course of the input potential

$$h(t') = \int_0^{\infty} \kappa(s) I(t' - s) ds \quad (5)$$

that the neuron receives between  $\hat{t}$  and  $t$ . With  $u(t'|\hat{t}) = \eta(t' - \hat{t}) + h(t')$  we have  $S_h(t|\hat{t}) = \exp[-\int_{\hat{t}}^t f(u(t'|\hat{t})) ds']$ . The subscript  $h$  in the survivor function is intended to remind the reader that  $S_h(t|\hat{t})$  is a *functional* of the input potential  $h(t')$  with  $\hat{t} < t' \leq t$ . The survivor function will be used in the next section.

### III. PSTH AND EXACT RATE DYNAMICS

We consider a single neuron that receives a time-dependent stimulus  $I(t)$ . The very same stimulus is repeated several times and a PSTH is built up. The PSTH is an approximation to the exact time-dependent firing rate  $r(t)$  of the spiking neuron model in response to the stimulus  $I(t)$ .

For a spike response model with escape noise the exact time-dependent firing rate  $r(t)$  can be found from the normalization condition [3,19]

$$1 = \int_{-\infty}^t S_h(t|\hat{t}) r(\hat{t}) d\hat{t}, \quad (6)$$

which must hold at any time  $t$ . Taking the temporal derivative yields [19]

$$r(t) = \int_{-\infty}^t P_h(t|\hat{t}) r(\hat{t}) d\hat{t}, \quad (7)$$

where  $P_h(t|\hat{t}) = -\frac{d}{dt} S_h(t|\hat{t})$  is a generalized time-dependent interval distribution [3]. For constant input  $h = I_0$  we have  $P_h(t|\hat{t}) = P(t - \hat{t})$ . Equation (7) states that the current firing rate can be derived from the input-dependent interval distribution averaged over all firing rates in the past. In other words, a neuron that has fired at time  $\hat{t}$  in the past and receives the input potential  $h$  contributes with a weight  $P_h(t|\hat{t})$  to the firing rate at time  $t$ .

In Fig. 2 we perform a simulation that mimics a typical PSTH experiment. The input current, shown in the lower panel of Fig. 2, is the sum of a weak constant bias and a perturbation. The perturbation is composed of a positive or negative pulse of the form  $as \exp(-s)/\tau_s$  with  $s = (t - t_0)/\tau_s$ . Here  $\tau_s = 5$  ms is the standard value for the synaptic time constant [ $\tau_s = 1$  ms in Fig. 2(b)],  $t_0 = 60$  ms is the stimulus onset, and  $a$  is the perturbation strength. The time course of the stimulus roughly mimics an excitatory (or inhibitory) postsynaptic current. The stimulus is repeated for 5000 times and a PSTH with 0.1 ms resolution (0.1 ms is the time step of the simulation) is built up. The PSTH response is then

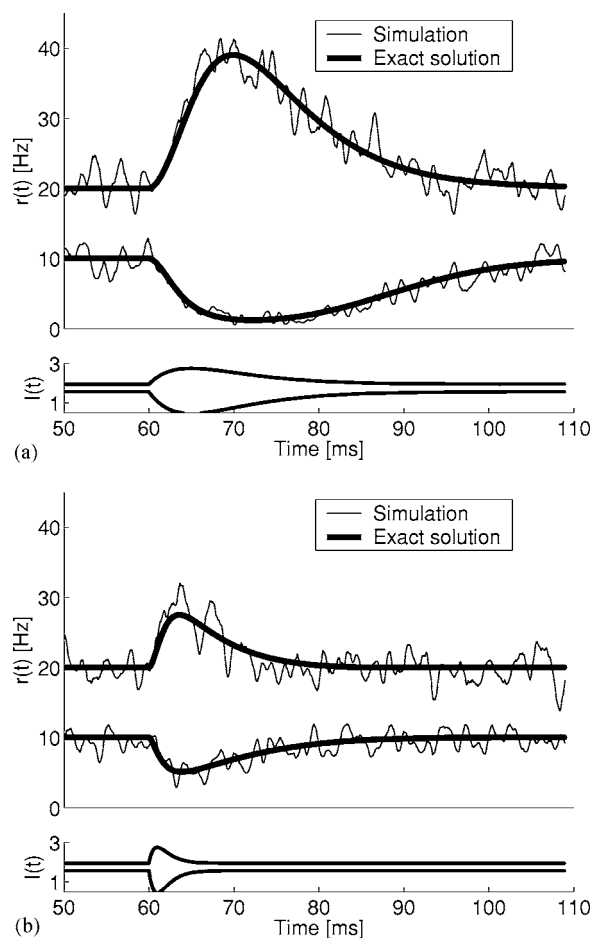


FIG. 2. The PSTH in response to a current pulse. Upper panel: the PSTH response  $r(t)$  of the simulated neuron (thin jagged line) and the result of the exact solution of the rate dynamics, Eq. (6) (smooth thick line), to the input  $I(t)$  plotted in the lower panel. Standard set of parameters, input with time constant  $\tau_s = 5$  ms in (a) and exceptionally  $\tau_s = 1$  ms in (b).

smoothed with a triangular filter of base of 1.5 ms and finally compared with the numerically integrated (time step 0.1 ms) result of Eq. (7) using the numerical method discussed in [21]. The agreement is excellent as it should be since Eq. (7) is an exact solution.

The exact solution is nice to have and readily integrated numerically after transformation to a partial differential equation in refractory densities [3,21]. However, since the solution of Eq. (7) is given *implicitly* and depends on the firing rates in the past, it does not fully correspond to our intuitive notion of a typical rate model. In standard rate models, the firing rate is given by an explicit function or functional of the input. For example, in a cascade rate model as used in systems neuroscience [6,7], the rate

$$r(t) = g\left[\int L(s) I(t-s) ds\right] \quad (8)$$

is given by a convolution of the input with a filter  $L$ , followed by some nonlinear “squashing” function  $g$ —e.g., [7,9]. This type of rate model has been called a CM since it

involves a “cascade” of two processing steps—i.e., linear filtering followed by a nonlinear function  $g$ . The aim of the next two sections is to approximate the exact, but *implicit* rate equation (7) by an explicit expression. Before we turn to the CM (see Sec. V) we discuss now a systematic Volterra expansion.

#### IV. VOLTERRA EXPANSION

We decompose the stimulus into a constant contribution  $I_0$  and a time-dependent contribution  $\epsilon I_1(t)$ :

$$I(t) = I_0 + \epsilon I_1(t). \quad (9)$$

The parameter  $\epsilon$  allows us to scale the strength of the time-dependent contribution. For  $\epsilon=0$  the time-dependent contribution vanishes and the firing rate of the neuron takes a constant value  $r_0$ . For  $\epsilon \ll 1$  it can be considered as a weak perturbation; for  $\epsilon=O(1)$ , the perturbation is strong. A strong perturbation is characterized by the fact that the rate variations  $|r(t)-r_0|$  are of the same order as  $r_0$ .

In order to arrive at an explicit, but approximate, expression for the firing rate  $r(t)$ , we expand the exact rate equation (6) or (7) in the parameter  $\epsilon$ . It turns out that the expansion can be most easily done on the level of the normalization equation (6). Hence we write

$$r(t) = r_0 + \epsilon r_1(t) + \frac{\epsilon^2}{2} r_2(t) + \dots, \quad (10)$$

$$S_h(t|\hat{t}) = S_0(t|\hat{t}) + \epsilon S_1(t|\hat{t}) + \frac{\epsilon^2}{2} S_2(t|\hat{t}) + \dots, \quad (11)$$

where Eq. (11) denotes the Volterra expansion of the survivor function  $S_h(t|\hat{t})$ . Note that  $S_h(t|\hat{t})$  is a *functional* of the time course  $h(t')$  for  $\hat{t} < t' \leq t$ . Hence with  $h_1 = \int_0^\infty \kappa(s) I_1(t-s) ds$  we have

$$S_n(t|\hat{t}) = \int_0^{t-\hat{t}} dx_1 \cdots \int_0^{t-\hat{t}} dx_n \frac{\partial^n S(t|\hat{t})}{\partial h(t-x_1) \cdots \partial h(t-x_n)} \times h_1(t-x_1) \cdots h_1(t-x_n), \quad (12)$$

where the derivative is evaluated at  $h(t)=I_0$ .

It is convenient to introduce a filter  $L^{(n)}$  of  $n+1$  arguments:

$$L^{(n)}(y, x_1, \dots, x_n) = \frac{\partial^n S(y|0)}{\partial h_1(y-x_1) \cdots \partial h_1(y-x_n)} \times \Theta(y - \max_i(x_i)). \quad (13)$$

The filters are perturbation independent, but since the derivative in Eq. (13) is evaluated at  $h_0$ , they do depend on the bias  $h_0$ . In Fig. 3, we plot the integrated first-order filter  $\int L^{(1)}(y, x) dy$  for two different constant inputs, one corresponding to a 10 Hz firing rate and the other to 20 Hz. For higher constant input,  $L^{(1)}$  approaches a  $\delta$  function. This is explained by the fact that for higher rates the spike train becomes more regular, as seen in the lower panel of Fig. 1. The higher the constant input, the more deterministic the

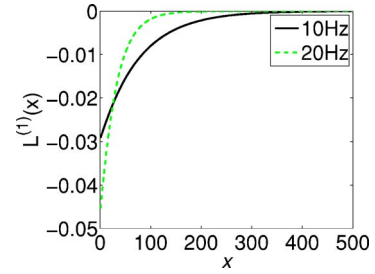


FIG. 3. (Color online) The integrated  $L^{(1)}$  filter,  $\overline{L^{(1)}}(x) \stackrel{\text{def}}{=} \int L^{(1)}(y, x) dy$ , plotted for two constant inputs corresponding to  $r_0=10$  Hz (solid line) and  $r_0=20$  Hz (dashed line).

neuron; hence, the transition of the survivor function  $S_h$  from 1 to 0 becomes sharper. Since the filters are derivatives of the survivor function, they approach a  $\delta$  function. This observation is studied in detail in [19].

In the Appendix we show that the definition of the filter  $L^{(n)}$  enables us to write the integral of  $S_n(t|\hat{t})$  with an arbitrary function  $z(\hat{t})$  as a  $(n+1)$ -fold convolution:

$$\int_{-\infty}^t S_n(t|\hat{t}) z(\hat{t}) d\hat{t} = \int_0^\infty dy \int_0^\infty dx_1 \cdots \int_0^\infty dx_n L^{(n)} \times (y, x_1, \dots, x_n) z(t-y) h_1(t-x_1) h_1(t-x_n). \quad (14)$$

After these preparations we now return to Eq. (6). Substituting the expansions (10) and (11) for  $r$  and  $S$  into (6) and sorting according to the order of  $\epsilon$  we get, for  $n \geq 1$ ,

$$0 = \frac{\epsilon^n}{n!} \int S_0(t|\hat{t}) r_n(\hat{t}) + \cdots + \binom{n}{i} S_i(t|\hat{t}) r_{n-i}(\hat{t}) + \cdots + S_n(t|\hat{t}) r_0 d\hat{t}. \quad (15)$$

Taking the temporal derivative and exploiting the fact that  $S_0(t|\hat{t})$  is just the standard survivor function (4) evaluated for constant input potential  $h=I_0$ , we get

$$r_n(t) = \int_0^\infty P_0(s) r_n(t-s) ds - \sum_{i=1}^n \binom{n}{i} \frac{d}{dt} \int_0^\infty S_i(t-s) r_{n-i}(t-s) ds. \quad (16)$$

With the identity (14) this result can be rewritten in condensed form as

$$r_n = P_0 * r_n - \sum_{i=1}^n \binom{n}{i} \frac{d}{dt} L^{(i)} * r_{n-i} * h_1 * \cdots * h_1, \quad (17)$$

where the asterisk denotes convolution.

We can now solve the rate equation (10) order by order. The solution to first order has been exploited in previous studies [8,19,21]. Fourier transform of  $r_1 = P_0 * r_1 - \frac{d}{dt} L^{(1)} * h_1$  yields

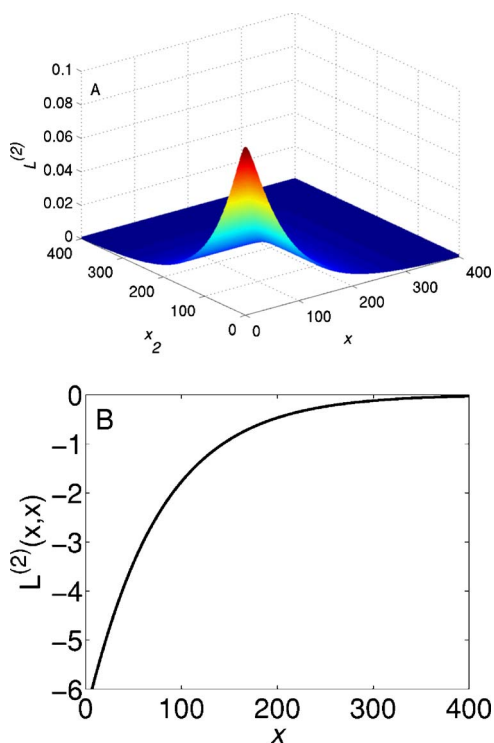


FIG. 4. (Color online) The integrated second-order filter  $\overline{L}^{(2)}(x_1, x_2) \stackrel{\text{def}}{=} \int dy L^{(2)}(y, x_1, x_2)$ . (a) Without the diagonal. (b) Only the diagonal. Notice the difference in scale and polarity on the y axis.

$$\hat{r}_1 = \hat{G}_1 \hat{h}_1 \stackrel{\text{def}}{=} \frac{i\omega r_0 \hat{L}^{(1)}}{1 - \hat{P}_0} \hat{h}_1, \quad (18)$$

and inverse Fourier transform yields the linear filter  $G_1$  from which we get  $r_1(t) = \int G_1(s) h_1(t-s) ds$ .

Similarly, we can derive  $G_2$ ,

$$\hat{r}_2 = \hat{G}_2 \hat{h}_1^2 \stackrel{\text{def}}{=} \frac{-i\omega(\hat{L}^{(1)}\hat{G}_1 + \hat{L}^{(2)}r_0)}{1 - \hat{P}_0} \hat{h}_1^2, \quad (19)$$

and again, inverse Fourier transform yields  $G_2$  from which we get  $r_2(t) = \int G_2(s, s') h_1(t-s) h_1(t-s') ds ds'$ .

In this section we focus on the contribution of the second-order term. With an escape noise model with escape function  $f$  introduced in Sec. II we find

$$\begin{aligned} L^{(2)}(y, x_1, x_2) \\ = S_0(y|0)[f'(y-x_1)f'(y-x_2) - \delta(x_1-x_2)f''(y-x_1)], \end{aligned} \quad (20)$$

where the primes denote the first and second derivatives—i.e.,  $f' = df/du$  and  $f'' = d^2f/du^2$ . In Fig. 4(b) we plot the integrated second-order filter  $\int dy L^{(2)}(y, x_1, x_2)$  as a function of  $x_1$  and  $x_2$ . We note the order-of-magnitude difference between the diagonal term and the rest.

In order to see how the different orders contribute, we return to the PSTH paradigm explored previously in Fig. 2(a). We compare the exact solution discussed earlier with

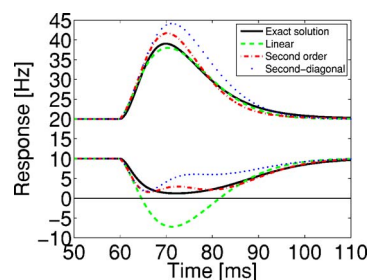


FIG. 5. (Color online) The exact solution (solid line) and the approximated response. Sum of linear filters up to first- (dashed line) and second- (dash-dotted line) order. In addition, the sum up to second order with only the diagonal term in Eq. (20) is plotted with a dotted line. The input perturbation is the same as in Fig. 2.

the first-order and second-order solutions derived in the present section. For *positive* perturbations, the first-order solution underestimates the effect of the perturbation, but this underestimation is overcorrected by the second-order contribution; cf. Fig. 5. While for small stimulation amplitudes the second order is better than the first-order approximation, as it should be, the first-order approximation is actually closer to the exact solution than the second-order one for the strong perturbation used in the figure. This suggests that for large positive perturbations the exact solution is mainly linear for positive perturbations. We will use this observation in Sec. V. For negative perturbation, the first-order solution predicts negative rates, which is physically impossible. The second-order term corrects this mistake, but still shows a significant difference to the exact solution; cf. Fig. 5.

From Fig. 4(b) we see that the diagonal term of Eq. (20) dominates the filter. We suggest, therefore, to take only the diagonal of the filter, thus reducing the number of dimensions to 1. Taking only the diagonal term  $x_1=x_2$  in Eq. (20), we have  $L_d^{(2)}(y, x_1, x_2) = \delta(x_1-x_2)S_0(y|0)[(f'(y-x_1))^2 - f''(y-x_1)]$  as the approximate second-order filter. In Fig. 5 this approximation is compared to the second-order solution. While inclusion of the diagonal term of the second-order offers an evident improvement over the first-order approximation for negative perturbations, it is not as good as the full second-order approximation. Note that the full second-order solution requires a double integration which is numerically expensive whereas the combination of the first-order term with the diagonal of the second-order term requires only two one-dimensional integrals.

### V. SINGLE- AND DOUBLE-CASCADE MODELS

A straightforward way of improving the approximation in Fig. 5 is to add higher orders. This would, however, be neither numerically efficient nor conceptually appealing. We therefore take another approach. We propose a CM in which the input goes through a cascade of processing steps: linear filters, nonlinear operations, and a final nonlinear function. This could either be a simple CM as in Eq. (8) or a more involved one as sketched in Fig. 6. The aim of the (final) nonlinear function is to approximate the cumulative effect of the missing higher orders.

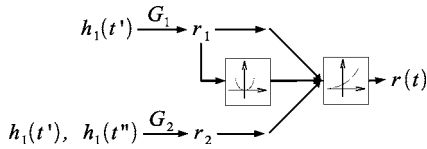


FIG. 6. A sketch of the information flow in the double CM. The input perturbation first goes through the  $G_i$  filters of increasing dimensions, Eqs. (18) and (19). In the second step, the result of the first-order filter,  $r_1$ , is squared, Eq. (25). Finally, a nonlinear squashing function, Eq. (21), is invoked. In the single CM only the uppermost path is taken.

The nonlinear function has to be wisely picked. Observing Fig. 5, we realize that the linear model approximation is reasonably good for positive perturbations, but completely wrong for strong negative perturbations since it predicts negative rates. Clearly, the response cannot be negative, and this hard bound introduces strong nonlinearities near zero. We therefore use as the nonlinear function a squashing function that is bounded from below by zero and has a linear behavior for positive input. These demands are met by the nonlinear function

$$c_1 \ln(1 + c_2 e^{c_3 z}), \quad (21)$$

where  $c_i$ ,  $i = 1, 2, 3$ , are constants to be determined later and  $z$  is a variable.

In order to obtain the first order CM, we write the approximated response up to first order in  $\epsilon$  as

$$\frac{r(t)}{r_0} = 1 + \epsilon \frac{r_1}{r_0} + O(\epsilon^2) \quad (22)$$

and we replace the right-hand side with the squashing function given in Eq. (21), using  $z = \epsilon r_1 / r_0$ . For this replacement to be correct at least to first order, we have to match the  $\epsilon$  coefficients of order  $\epsilon^0$  and  $\epsilon^1$ .

Expanding Eq. (21) to first order in  $\epsilon$  gives  $c_1 \ln(1 + c_2) + D \epsilon \frac{r_1}{r_0}$  with  $D = \frac{c_1 c_2 c_3}{1 + c_2}$ . Matching the zero order of this expansion and that of Eq. (22) yields  $c_1 = 1 / \ln(1 + c_2)$ . Matching also the first-order coefficients yields  $D = 1$  or  $c_3 = \alpha(c_2) \stackrel{\text{def}}{=} (1 + c_2) \ln(1 + c_2) / c_2$ . Hence  $c_1$  and  $c_3$  are both given as a function of  $c_2$  so that we are left with a single free parameter.

The single CM is then given by

$$r(t) = \frac{r_0}{\ln(1 + c_2)} \ln(1 + c_2 e^{\alpha(c_2) \epsilon r_1(t) / r_0}), \quad (23)$$

where  $r_1(t) = (G_1 * h_1)(t)$ ; see Eq. (18).

In general, the number of steps in the CM depends on the desired accuracy—i.e., the order of expansion. Additional steps are required for higher orders. The additional steps guarantee that the low orders, as given by the expansion, are not altered by the squashing function. In particular, we have to eliminate expressions like  $\epsilon^k r_l^m$  with  $k \neq l \neq m$ . In what follows, the procedure for the second order CM is given.

We introduce auxiliary variables  $y_1$  and  $y_2$ , and define the variable  $z$  in (21) as  $z = \epsilon y_1 + \frac{\epsilon^2}{2} y_2$ . We then expand Eq. (21) to second order in  $\epsilon$ , getting  $c_1 \ln(1 + c_2) + D \epsilon y_1 + D \frac{\epsilon^2}{2} (y_2 + \frac{c_3}{1 + c_2} y_1^2)$ , with  $D$  defined as above. At the same time we know that by definition

$$r(t) = r_0 \left( 1 + \epsilon \frac{r_1}{r_0} + \frac{\epsilon^2}{2} \frac{r_2}{r_0} \right) + O(\epsilon^3) \quad (24)$$

must hold. By comparing the  $\epsilon$  coefficients, we get the correct assignment for  $y_1$  and  $y_2$ :

$$y_1 \stackrel{\text{def}}{=} \frac{r_1}{r_0}, \quad y_2 \stackrel{\text{def}}{=} \frac{r_2}{r_0} - \frac{\ln(1 + c_2)}{c_2} \left( \frac{r_1}{r_0} \right)^2, \quad (25)$$

where  $r_1(t) = (G_1 * h_1)(t)$  and  $r_2(t) = (G_2 * h_1 * h_1)(t)$ ; see Eq. (19). This procedure defines the double CM:

$$r(t) = \frac{r_0}{\ln(1 + c_2)} \ln(1 + c_2 e^{\alpha(c_2) z}) \quad (26)$$

with  $z = \epsilon y_1 + \frac{\epsilon^2}{2} y_2$  and the substitutions defined in Eq. (25).

For an interpretation of the double CM, let us look at the flow diagram in Fig. 6. The input first goes through a set of two linear filters: in the next step, the filters' output is further manipulated according to (25) and finally passed through the squashing function.

The free parameter  $c_2$  is yet to be optimized. To that end, we compute the exact solution for inhibitory and excitatory synaptic current inputs of the form  $x e^x$  (cf. lower panel of Fig. 2). Both perturbations are given on top of a static bias input that corresponds to an output rate of  $r_0 = 10$  Hz. We then approximate the responses using the single CM as described in Eq. (23) for a range of  $c_2$  values. The responses are plotted in Fig. 7(a). The traces of the exact solution and those of the CM for both inputs display a unimodal response, with a clear extremum (i.e., maximum for positive perturbation and minimum for negative ones). We optimize  $c_2$  by minimizing the differences between the extrema of the exact solution and those of the CM. While this simple method may not be rigorous, it is simple enough to allow experimental application. In Fig. 7(b), the differences at the extrema are plotted as a function of  $c_2$ , revealing that the single CM agrees on a single optimal value for both types of perturbations,  $c_2 = 1.75$ . By construction, the performance of the CM in response to short current pulses is excellent, if the bias rate is  $r_0 = 10$  Hz—i.e., the one used during optimization of the parameter  $c_2$  [Fig. 8(a)].

Can we use the same value,  $c_2 = 1.75$ , also for other values of the reference rates  $r_0 \neq 10$  Hz? To answer this question, we give a series of current pulses of the form  $x e^x$  as in Fig. 2, but this time with different bias rates. In order to compare different bias rates, we always plot a normalized response  $[r(t) - r_0] / r_0$ ; cf. Fig. 8(a). In Fig. 8(b) the responses' extrema of the normalized exact solution and that of the single CM are plotted for constant inputs that correspond to  $r_0 = 5, 10$ , and 20 Hz. We emphasize that  $c_2$  is kept fixed for the various constant inputs. As expected, Fig. 8 shows good agreement for the  $r_0 = 10$  Hz case. This is due to the fact that optimiza-

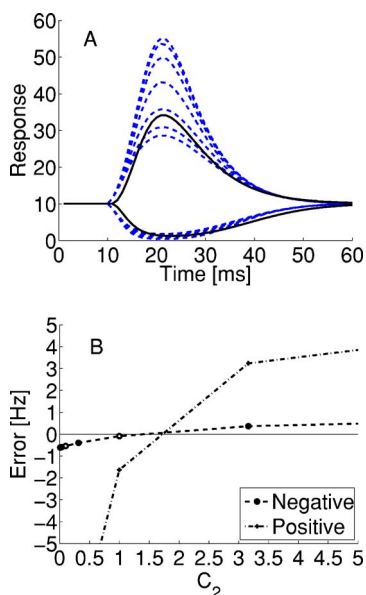


FIG. 7. (Color online) Optimizing  $c_2$  parameter. (a) Responses to positive and negative perturbations are computed for the exact solution (solid line). They are then computed for the single CM (dashed line) for a range of  $c_2$  values (from top to bottom:  $\log_{10}(c_2) = -2, -1.5, \dots, 1$ ). (b) The differences at the extrema of the response are plotted as a function of  $c_2$ . Negative perturbations are noted as a dashed line and positives perturbation as a dash-dotted line. The errors show a single optimal value at  $c_2 = 1.75$ .

tion of  $c_2$  was done for this particular constant input. It is also apparent that the approximation errors of the other rates are small and confined to positive perturbations. In general, the errors are only weakly dependent on the constant input level.

In order to appreciate the contribution of the CM's squashing function, we calculate the single- and double-CM responses to the same perturbations as in Fig. 2 and compare them again to the exact solution. Figure 9 should be compared to the approximation obtained by the simple linear filter model as plotted in Fig. 5. While no degradation in performance is observed in the positive perturbation, a significant improvement compared to Fig. 5 can be seen for the negative one. Also, note that a good fit is obtained already in the single CM and that little is gained by using the double CM.

To further assess our finding, we continue the comparison between the *single* CM and the exact solution, for a variety of input scenarios. The  $c_2$  value obtained above, 1.75, is optimal for the input mimicking a strong postsynaptic current, but not necessarily for other scenarios (see Fig. 10). In principle,  $c_2$  can be optimized for each input scenario,  $I(t)$ . In order to test the validity of the model, however, we use the same  $c_2$  value for all the other input scenarios.

A superposition of three incommensurable frequencies was given as a perturbation  $I_1(t) = 0.2[\sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t)]$ ; with  $f_1 = 1$  Hz,  $f_2 = 6.9$  Hz, and  $f_3 = 42.7$  Hz. A frequency sweep was also given with a perturbation  $I_1(t) = 0.4 \sin(2\pi \omega t)$ , with  $\omega$  going from 1 Hz to 20 Hz.

Pyramidal neurons *in vivo* are exposed to colored noise

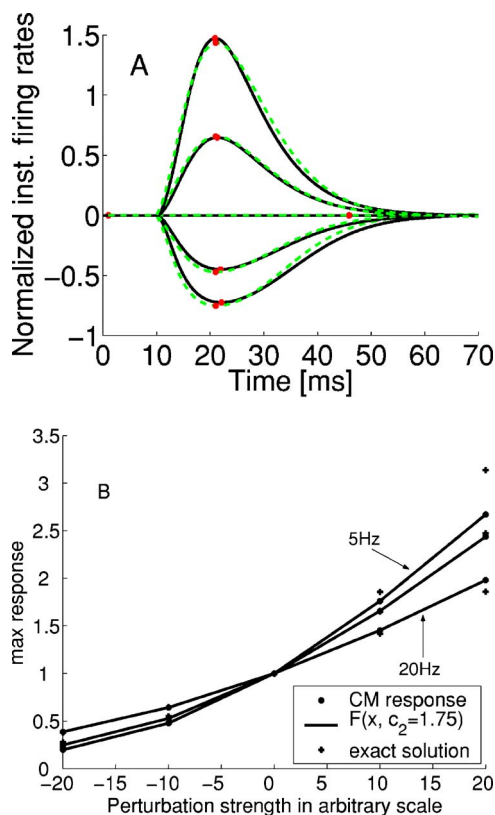


FIG. 8. (Color online) The effect of different constant input levels on the approximation error. (a) The normalized responses of the exact solution (solid line) and the CM (dashed line) for a series of pulses are computed. The difference at responses' extrema (marked by circles) is taken as measure of error. The output rate is set to 10 Hz. (b) The maximum normalized response of the exact solution (+) and of the CM (solid line and circles) as a function of perturbation amplitude for various mean output rates (5, 10, and 20 Hz).

that is characterized by a frequency cutoff at 200 Hz [24]. In Fig. 11, we compare the response of the exact solution to that of the single CM for filtered noise perturbation with cutoff frequency at 200 Hz.

To conclude, we tested our model on a wide range of input scenarios (Figs. 9–11). For all these input scenarios the exact solution is hardly distinguishable from the single CM. The double CM is not shown since its solution is indistinguishable from the exact solution. Already the single CM obtains an excellent estimate of the exact solution. These results, together with the simple optimization procedure offered, make the single CM an attractive candidate for an effective rate model of biological neurons.

## VI. DISCUSSION

In this paper we have developed an effective rate model for a spiking neuron of the integrate-and-fire type—i.e., a spike response model with escape noise. In previous work [19] the full rate dynamics was given in form of an integral equation (see also [18,20]) and a linearized version in form of a linear filter. Here, we have continued beyond first order

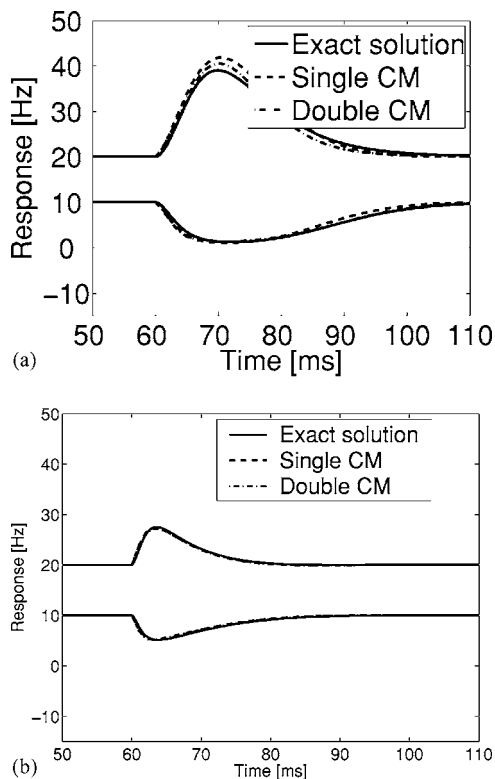


FIG. 9. The exact solution (solid line) compared to the single- (dashed line) and double- (dash-dotted line) CM response. The  $c_2$  constant is set to 1.75. The input perturbation is the same as in Fig. 2 and characterized by a time constant of 5 ms for (a) and of 1 ms for (b).

and expanded the exact solution, Eq. (7), up to arbitrary order. The expansion, Eq. (10), gives an explicit expression of the rate as a function of its input. We emphasize that the given rate  $r(t)$  is the instantaneous rate, averaged over realizations of the stochastic process, not over time. Hence, the dynamics of the rate, up to desired level of accuracy, can be derived in explicit form.

Our explicit formulas for the firing rate do *not* imply an increase in numerical efficiency compared to the rapid integration scheme of the exact solution (7) based on partial differential equations for the refractory density as discussed in [3,21]. In particular, the numerical solution of the full equation is faster than the second-order Volterra model or the double CM. The main advantage of the explicit expressions of the rate in a single equation is to simplify communication with experimental neuroscientists. In fact, our single CM is equivalent to the LNP model widely used in systems of neuroscience, in particular vision [6,7]. We see the identification of a mapping between spiking neuron models and a CM of the LNP type as the main achievement of our paper. As an aside we suggest that the second-order Volterra model could potentially be useful as a conceptual framework for experimental systems in neuroscience.

In general, any functional can be expanded into a Volterra series [25], hence the general structure of the solution, Eq. (17), is not surprising. The specific form of the filters, however, is determined by the neuron model, Eq. (1), and the noise model, Eq. (2). In particular, refractoriness influences

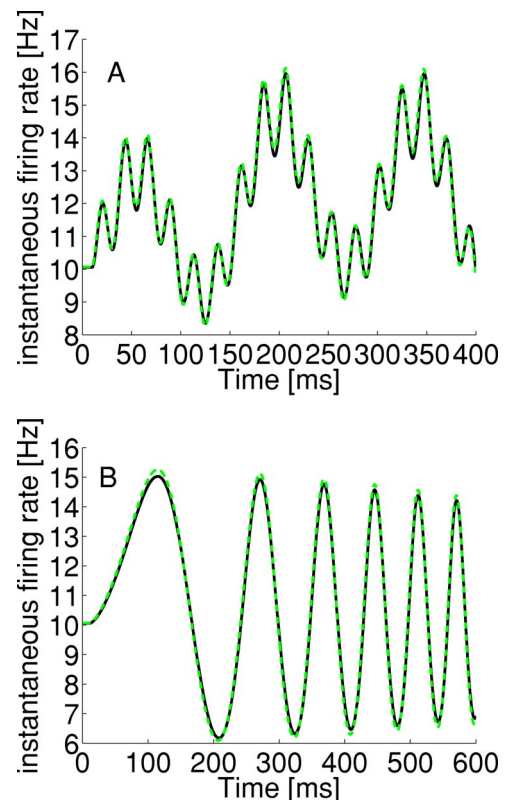


FIG. 10. (Color online) Comparison of the exact solution (solid line) and the single CM (dashed line). (a) Superposition of three frequencies: 1, 6.9, and 42.7 Hz. (b) A frequency sweep going from 1 Hz to 20 Hz.

the shape of the filters. We emphasize that the filters depend only on the mean input (which determines the reference point for the Volterra expansion) but are otherwise input independent. Hence, once the filters are calculated, computing the approximated response simply amounts to filtering the input. This can be efficiently computed by using fast Fourier transform as in Eqs. (18) and (19).

For the Gaussian hazard function used here, Eq. (2), we found that the diagonal dominates the second-order filter. Hence we proposed a simplification in which the filter includes only the diagonal. This method gives an approximation that is not as good as the full second-order filter, but is much better than using only the first-order filter, as can be seen in Fig. 5. The advantage of this simplification is the reduction in filter dimensionality: instead of a two-dimensional filter, we are left with a one-dimensional filter.

While the linear-nonlinear-Poisson model framework is widely used in the neuroscience literature, its justification has so far been mainly heuristic. In this contribution, we have chosen a simplified noisy spiking neuron model in order to gain a deeper understanding of the relation of LNP models to spiking neuron models. In our approach, the linear filter can be calculated exactly and we have shown that the accuracy of the approximation obtained with linear filters can be significantly improved by transforming the filters' output through a nonlinear function. The nonlinear function replaces the effect of the missing high orders. We found that the nonlinear function (i) should be positive in order to pre-



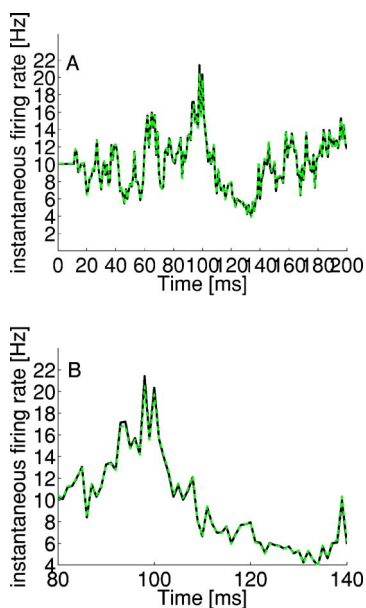


FIG. 11. (Color online) Comparison of the exact solution (solid line) and the single CM (dashed line) for low-pass-filtered noise input. (a) The response to filtered noise input perturbation with a cutoff frequency at 200 Hz. (b) Zoom into (a).

vent negative firing rates, (b) should behave close to linearly for large positive perturbations, and (c) should be a smooth function. Our function, Eq. (21), meets these criteria.

Our CM constructed along those lines boosts the performance of the first-order approximation with only one single free parameter, called  $c_2$  in this paper. In addition, the optimal value of this parameter only weakly depends on the constant input level  $I_0$  as Fig. 8 shows. While this optimization procedure is not rigorous, its simplicity should allow its application to experimental data. In a first step, the best linear filter should be extracted using standard methods. In a second step, the single free parameter of the CM has to be determined. Our results suggest that this can be done for a simple perturbation, such as a simulated synaptic input current. Once fitting is done, the CM should be able to describe the response to *any* perturbation with reasonable accuracy.

Finally, our approach has allowed us to develop higher-order CM's—e.g., the double CM as sketched in Fig. 6. For the perturbations we tried, the double CM's response is indistinguishable from the exact solution. However, since the double CM has an increased structural complexity compared to the single-CM one and since it offers no numerical advantages compared to an efficient implementation of the exact solution, we advocate the use of either the exact solution or the single CM rather than the double CM.

In this paper we have used the escape-rate noise model with a fairly simple Gaussian hazard function, Eq. (2). This hazard function matches the behavior of the diffusive noise model for a subthreshold input. In order to match the diffusive noise model in other scenarios, a more elaborate hazard function has to be taken [21]. It remains to be investigated how the approximated response with a more elaborate hazard function compares with the firing rate of an integrate-and-fire model with diffusive noise (stochastic arrival of background

spikes). An alternative approach to calculating the response of an integrate-and-fire model with diffusive noise is by direct numerical solution of the membrane potential density equations [15,17,26–28].

An approach similar to ours was taken by Shriki *et al.* [29]. In their model, the input is first passed through a second-order bandpass filter that models the effect of the synaptic filtering, then through a threshold-linear function. The threshold linear function is justified by experimental results that show a linear  $f-I$  curve which is independent of the leak conductance. The threshold  $\theta$ , however, is leak conductance dependent. It is interesting to note that their phenomenological model and our analytically derived model are both converging into a similar structure; that is, a CM of first- or second-order filter and then a threshold-linear function as the nonlinear function. Note, however, that our model is designed to capture fast perturbations whereas theirs focuses on slow variations of the firing rate (see also [30]).

Recently several studies have established a link from electrophysiological data or detailed neuron models to simplified stochastically spiking neuron models [10,31–35]. We hope that our present work contributes towards establishing a further link from spiking neurons to rate models of the CM family.

In this paper we have connected the firing rate of the PSTH with the dynamics of a single-neuron model, characterized by parameters such as the threshold, noise, refractoriness, membrane time constant, and synaptic time constant. Changes in any of these parameters will influence the time course of the PSTH in a manner predictable by our theory [see, for example, Figs. 9(a) and 9(b) for the influence of the synaptic time constant  $\tau_s$ ]. Analogously, we expect that the same methods can be applied to a pair of two neurons in order to analyze the joint PSTH, including correlations. Ideally, such an analysis could reveal the fraction of shared input as well as functional connectivity between the two neurons.

### APPENDIX

In this appendix we show that the  $n$ th term  $S_n(t|\hat{t})$  can be written as an  $(n+1)$ -dimensional filter operating on the input perturbation,  $h_1(t)$ .

Expanding  $S_h(t|\hat{t})$  in a Volterra series [25] around  $h_0$  yields the  $n$ th term

$$S_n(t|\hat{t}) = \int_{-\infty}^t \dots \int_{-\infty}^t \frac{\partial^n S(t|\hat{t})}{\partial h_1(s_1) \dots \partial h_1(s_n)} \times \Theta(s_1 - \hat{t}) \dots \Theta(s_n - \hat{t}) h_1(s_1) ds_1 \dots h_1(s_n) ds_n, \tag{A1}$$

with the derivative taken at the unperturbed input  $h(t)=h_0$ . The  $\Theta(s_i - \hat{t})$  factors are required since  $S_h(t|\hat{t})$  is not defined for  $t < \hat{t}$ .

The Volterra series'  $n$ th-order term is similar to the Taylor expansion's  $n$ th-order term, only an integration is performed, since  $S(t|\hat{t})$  is not a function but a functional.

Changing variables  $x_i \stackrel{\text{def}}{=} t - s_i$  and  $y = t - \hat{t}$ , we get

$$S_n(t|t-y) = \int_0^\infty \cdots \int_0^\infty \frac{\partial^n S(t|t-y)}{\partial h_1(t-x_1) \cdots \partial h_1(t-x_n)} \times \Theta(y - \max_i(x_i)) \cdot h_1(t-x_1) \times dx_1 \cdots h_1(t-x_n) dx_n. \quad (\text{A2})$$

Now, since the derivative is taken at the unperturbed input, it is translational invariant with respect to time. Hence we can arbitrarily set the zero in a convenient point in time, such as the last spike time. Setting  $\hat{t}=0$ , we have  $t=y$ .

Defining the  $n+1$  dimension filter

$$L^{(n)}(y, x_1, \dots, x_n) \stackrel{\text{def}}{=} \frac{\partial^n S(y|0)}{\partial h_1(y-x_1) \cdots \partial h_1(y-x_n)} \times \Theta(y - \max_i(x_i)),$$

we can write the  $n$ th term as

$$S_n(t|y) = \int_0^\infty dx_1 \cdots \int_0^\infty dx_n L^{(n)}(y, x_1, \dots, x_n) \times h_1(t-x_1) \cdots h_1(t-x_n)$$

or, in a convolution notation,  $S_n(t|y) = [L^{(n)}(y, x_1, \dots, x_n) * h_1 * \cdots * h_1](t)$ . The above change of variables can also be done for the expression  $\int_{-\infty}^t S_n(t|\hat{t}) r_m(\hat{t}) d\hat{t}$ .

Again, setting  $\hat{t}=0$  yields a convolution with  $r_m$  as well:

$$\int_{-\infty}^t S_n(t|\hat{t}) r_m(\hat{t}) d\hat{t} = [L^{(n)}(y, x_1, \dots, x_n) * r_m * h_1 * \cdots * h_1](t),$$

which gives Eq. (14).

- 
- [1] A. L. Hodgkin and A. F. Huxley, *J. Physiol. (London)* **117**, 500 (1952).
- [2] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA, 1991).
- [3] W. Gerstner and W. K. Kistler, *Spiking Neuron Models* (Cambridge University Press, Cambridge, UK, 2002).
- [4] L. Lapique, *J. Physiol. Pathol. Gen.* **9**, 620 (1907), cited in H. C. Tuckwell, *Introduction to Theoretic Neurobiology* (Cambridge University Press, Cambridge, UK, 1988).
- [5] R. B. Stein, *Biophys. J.* **7**, 37 (1967).
- [6] E. J. Chichilnisky, *Network* **12**, 199 (2001).
- [7] O. Schwartz and E. P. Simoncelli, *Nat. Neurosci.* **4**, 819 (2001).
- [8] W. Gerstner, *Neural Networks* **14**, 599 (2001).
- [9] B. Aguera y Arcas and A. Fairhall, *Neural Comput.* **15**, 1789 (2003).
- [10] L. Paninski, J. Pillow, and E. Simoncelli, *Neural Comput.* **16**, 2533 (2004).
- [11] F. Theunissen and J. Miller, *J. Comput. Neurosci.* **2**, 149 (1995).
- [12] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes—Exploring the Neural Code* (MIT Press, Cambridge, MA, 1996).
- [13] M. Berry and M. Meister, *J. Neurosci.* **18**, 2200 (1998).
- [14] M. Oram, M. Wiener, R. Lestienne, and B. Richmond, *J. Neurophysiol.* **81**, 3021 (1999).
- [15] N. Brunel and V. Hakim, *Neural Comput.* **11**, 1621 (1999).
- [16] S. Fusi and M. Mattia, *Neural Comput.* **11**, 633 (1999).
- [17] N. Brunel, F. S. Chance, N. Fourcaud, and L. F. Abbott, *Phys. Rev. Lett.* **86**, 2186 (2001).
- [18] H. R. Wilson and J. D. Cowan, *Biophys. J.* **12**, 1 (1972).
- [19] W. Gerstner, *Neural Comput.* **12**, 43 (2000).
- [20] B. W. Knight, *J. Gen. Physiol.* **59**, 734 (1972).
- [21] A. Herrmann and W. Gerstner, *J. Comput. Neurosci.* **11**, 135 (2001).
- [22] H. E. Plesser and W. Gerstner, *Neural Comput.* **12**, 367 (2000).
- [23] D. R. Cox, *Renewal Theory* (Methuen, London, 1962).
- [24] A. Destexhe, M. Rudolph, and D. Pare, *Nat. Rev. Neurosci.* **4**, 739 (2003).
- [25] V. Volterra, *Theory of Functionals and of Integral and Integro-differential Equations* (Dover, New York, 1959).
- [26] B. W. Knight, *Neural Comput.* **12**, 473 (2000).
- [27] D. Nykamp and D. Tranchina, *J. Comput. Neurosci.* **8**, 19 (2000).
- [28] A. Omurtag, B. Knight, and L. Sirovich, *J. Comput. Neurosci.* **8**, 51 (2000).
- [29] O. Shriki, D. Hansel, and H. Sompolinsky, *Neural Comput.* **15**, 1809 (2003).
- [30] A. Treves, *Network* **4**, 259 (1993).
- [31] B. Aguera y Arcas, A. Fairhall, and W. Bialek, *Neural Comput.* **15**, 1715 (2003).
- [32] W. M. Kistler, W. Gerstner, and J. L. van Hemmen, *Neural Comput.* **9**, 1015 (1997).
- [33] R. Jolivet, T. Lewis, and W. Gerstner, *J. Neurophysiol.* **92**, 959 (2004).
- [34] J. Pillow, L. Paninski, and E. Simoncelli, in *Advances in Neural Information Processing Systems*, edited by S. Thrun, L. Saul, and B. Schölkopf (2004), Vol. 16, pp. 1311–1318.
- [35] J. Keat, P. Reinagel, R. Reid, and M. Meister, *Neuron* **30**, 803 (2001).