NOTE Communicated by Alain Destexhe

Employing the $\mathcal{Z}$-Transform to Optimize the Calculation of the Synaptic Conductance of NMDA and Other Synaptic Channels in Network Simulations

J. Köhn and F. Wörgötter
Department of Neurophysiology, Ruhr-Universität, 44780 Bochum, Germany

Calculation of the total conductance change induced by multiple synapses at a given membrane compartment remains one of the most time-consuming processes in biophysically realistic neural network simulations. Here we show that this calculation can be achieved in a highly efficient way even for multiply converging synapses with different delays by means of the $\mathcal{Z}$-transform. Using the example of an NMDA synapse, we show that every update of the total conductance is achieved by an iterative process requiring at most three multiplications, which together need only the history values from the two most recent iterations. A major advantage is that this small computational load is independent of the number of synapses simulated. A benchmark comparison to other techniques demonstrates superior performance of the $\mathcal{Z}$-transform. Nonvoltage-dependent synaptic channels can be treated similarly (Olshausen, 1990; Brettle & Niebur, 1994), and the technique can also be generalized to other synaptic channels.

1 Introduction

In most biophysically realistic network simulations, the conductance changes at every synapse have to be updated for every simulated time step. The divergence in the connection structure thereby determines how many synapses actually exist. Usually, however, this number exceeds the number of neurons by far. Thus, synaptic conductance update is the most time-consuming computational process in almost all biophysically realistic network simulations. This process has to be repeated for every membrane compartment, which could be a small dendritic cylinder or the total cell, depending on the network model's level of complexity. The basic problem is how to compute the total conductance g_{total} most efficiently for every postsynaptic membrane compartment that is given by the convolution of all input spike trains s_i with the characteristic synaptic functions g_i of the (possibly different) synapses at the current time t_x:

Neural Computation **10, 1639–1651** (1998)

In continuous time:

$$g_{total}(t_x) = \sum_{i=0}^{N} \omega_i \int_0^{t_x} s_i(\tau) g_i(t_x - \tau) d\tau, \tag{1.1}$$

where N is the total number of synapses and s_i denotes the input spike train consisting of delta pulses at synapse i. The synaptic weight is given by ω_i. In discrete time, the integral will be replaced by a second sum. The problem that hides beneath this equation is the fact that a rather long history of past incoming spikes has to be saved and used in order to calculate g_{total} at the current time t_x. These "history values" have to be stored in an extended memory queue in order to make them available to the next iteration step, and every history value enters in the calculation, leading to a rather high number of numerical operations.

Several algorithms have been proposed to accelerate this process based on different basic synapse models (Srinivasan & Chiel, 1993; Bernard, Ge, Stockley, Willis, & Wheal, 1994; Lytton, 1996). Of primary relevance here are the most commonly used models, based on so-called α functions. These functions are described by

$$g(t) = \hat{g} \frac{e}{\tau} t e^{-\frac{t}{\tau}}, \tag{1.2}$$

where $\hat{g}$ is peak conductance and τ is time constant (typically around 1 ms). They represent the characteristic synaptic function of a regular nonvoltage-dependent synapse, which can also be inhibitory. In 1990 Olshausen showed that the $\mathcal{Z}$-transform (Doetsch, 1967; Oppenheim & Schafer, 1975) can be employed to accelerate tremendously the calculation of the convolution between exponential functions and impulse inputs. Synaptic computations at regular nonvoltage-dependent synapses are the immediately obvious application of this technique. In a first step, all synapses modeled by the same α function are combined, and all individual inputs s_i are accumulated into one weighted input function s for every simulated membrane compartment. We call this function the accumulated spike train:

$$s(t) = \sum_{i=0}^{N} \omega_i s_i(t). \tag{1.3}$$

Then the convolution is solved in discrete time $t_n = nT$ using the $\mathcal{Z}$-transform, arriving at:

$$\begin{aligned} g_{\text{total}}(nT) = {} & T \cdot \hat{g}_A \cdot e^{-\frac{T}{\tau}} s((n-1)T) + 2e^{-\frac{T}{\tau}} g_{\text{total}}((n-1)T) \\ & - e^{-2\frac{T}{\tau}} g_{\text{total}}((n-2)T). \end{aligned} \tag{1.4}$$

This final form of the conductance update (Brettle & Niebur, 1994) shows that only two values of the history of g_{total} need to be taken into account in order to arrive at the exact solution, which is spelled out for regular exponential functions in Olshausen (1990).[1]

Here we will use the $\mathcal{Z}$-transform to derive the solution for g_{total} of the voltage-dependent NMDA channel, and we will show that the final result requires only two history values and a few multiplications. It should be noted that this article summarizes a known technique (Olshausen, 1990; Brettle & Niebur, 1994), extending it to NMDA channels and comparing it to other methods.

2 The Algorithm

The NMDA channel is characterized by a nonlinear current-voltage relationship in which the actual conductance g depends on the currently existing membrane potential V_m. It is modeled by:

$$g(t) = \hat{g}\frac{e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}}{1 + \eta[Mg^{2+}]e^{-(\gamma V_m)}}, \tag{2.1}$$

where $\hat{g}$ is peak conductance of NMDA channel, τ_1 and τ_2 are the first and second time constants of NMDA channel (τ_1 =40 ms, τ_2 =0.33 ms), Mg^{2+} is magnesium concentration (mM), η is 0.33 mM^{-1}, γ is 0.06/mV, and V_m is membrane potential in mV.

We will not discuss the validity of this equation which has been used by several authors (Jahr & Stevens, 1990a, 1990b; Mel, 1992a) to model the NMDA channel. Instead, we assume that it represents an NMDA channel with sufficient accuracy. This equation is split into two parts that can be treated independently, as shown by Bernard et al. (1994), because c is not dependent on the time but only on the membrane potential:

$$g(t) = c \cdot y(t) \tag{2.2}$$

$$y(t) = \hat{g} \cdot (e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}) \tag{2.3}$$

$$c = (1 + \eta[Mg^{2+}]e^{-\gamma V_m})^{-1}. \tag{2.4}$$

This method had also been used by Bernard et al. (1994), whose experimental results show that the rise time is independent of the Mg^{2+} concentration (Lester & Jahr, 1992; Stern, Edwards, & Sakmann, 1992).

[1] To our knowledge the derivation for general exponential functions is explicitly spelled out only in this technical report. This report may be hard to obtain, but we think that it should be easy to derive equation 1.4 in a similar way as the results shown here.

Equation 2.3 has to be $\mathcal{Z}$-transformed in discrete time t_n=nT:

$$Y(z) = \mathcal{Z}\{y(nT)\} = \mathcal{Z}\{\hat{g} \cdot (e^{-\frac{nT}{\tau_1}} - e^{-\frac{nT}{\tau_2}})\}, \tag{2.5}$$

with

$$\mathcal{Z}\{e^{-\frac{nT}{\tau_1}}\} = \frac{1}{1 - e^{-\frac{T}{\tau_1}} \cdot z^{-1}}; \quad \mathcal{Z}\{e^{-\frac{nT}{\tau_2}}\} = \frac{1}{1 - e^{-\frac{T}{\tau_2}} \cdot z^{-1}}. \tag{2.6}$$

These terms can be added in the $\mathcal{Z}$-domain just as in the time domain:

$$Y(z) = \hat{g}\frac{a_1 z^{-1} - a_2 z^{-1}}{(1 - a_1 z^{-1})(1 - a_2 z^{-1})}, \tag{2.7}$$

where we have set $a_1 = e^{-\frac{T}{\tau_1}}$ and $a_2 = e^{-\frac{T}{\tau_2}}$.

A convolution of the weighted input function $s(nT)$ with the numerator of the conductance function $y(nT)$ is equivalent to a multiplication of both functions in the $\mathcal{Z}$-domain:

$$h(nT) = y(nT) \star s(nT) \Leftrightarrow H(z) = Y(z) \cdot S(z). \tag{2.8}$$

After some conversions we arrive at:

$$H(z) = \hat{g}(a_1 - a_2)S(z)z^{-1} + (a_1 + a_2)H(z)z^{-1} - a_1 a_2 H(z)z^{-2}. \tag{2.9}$$

To get the inverse transform, we use the following feature of the $\mathcal{Z}^{-1}$-transform,

$$\mathcal{Z}^{-1}\{F(z) \cdot z^{-j}\} = f[(n - j)T], \tag{2.10}$$

and get as the final solution:

$$h[nT] = \hat{g}(a_1 - a_2)s[(n-1)T] + (a_1 + a_2)h[(n-1)T] - a_1 a_2 h[(n-2)T]. \tag{2.11}$$

Where $s[(n-1)T]$ is the summed total input at time step $(n-1)T$,

$$s[(n-1)T] = \sum_i \omega_i \, s_i[(n-1)T]. \tag{2.12}$$

It should be noted that $s[nT]$ does not contribute to $h[nT]$ because spikes arriving at the moment nT will affect the conductance only afterward.

$h[(n-1)T]$ and $h[(n-2)T]$ represent the last and the last-but-one iteration of the function h, which are now reutilized to compute the actual value

of $h[nT]$. Thus, very little computational effort is required to perform the ongoing iteration of h, and the final conductance is computed as:

$$g_{total}[nT] = c \cdot h[nT]. \tag{2.13}$$

This shows that the term c does not enter into the iterations. The validity of this has been demonstrated by Bernard et al. (1994). It should be noted that the final result is not an approximation; it is exact in the sense that the $\mathcal{Z}$-transform does not alter the accuracy of the computation of the convolution in discrete time.

Figures 1A and 1B show the behavior of such an NMDA synapse for a single input spike in comparison to a non-NMDA-synapse modeled with equations 1.2 and 1.4. In Figures 1C and 1D, the response to a 100 Hz spike train s for the NMDA synapse is depicted. The resulting curves are in accordance with those found in other simulation studies (Mel, 1992b; Tråvén et al., 1993). Curves determined by a conventional calculation of the convolution with single-precision floating-point variables are identical up to a numerical accuracy of 10^{-8} if we assume memory queue sizes of about 800 values in the conventional calculation (iteration time-step 0.05 ms). Note that in order to cover the complete response for high-frequency input spike trains, the required memory queue size is particularly large for the slow NMDA synapse. Thus, the increase in computational speed and the reduction of memory queue size in this case exceeds two orders of magnitude. Both estimates obviously depend on the accuracy (e.g., the size of the time-step) required for the conventional calculation.

3 Benchmark Comparison to Other Techniques

Several techniques have been described in order to speed up synaptic conductance calculation. Srinivasan and Chiel (1993) showed how multiple α functions could be consolidated by representing their summation in an iterated closed form. Lytton (1996) used a different type of synapse based on the work of Desthexe, Mainen, and Sejnowski (1994a, 1994b), which can be seen as a concave rising exponential $(1 - e^{-t})$ pieced together with one that is convex and falling (e^{-t}). The major advantage of the latter approach lies in the fact that actual transmitter application durations can be included in this model. Desthexe et al.'s set of equations is even more compact than that of Srinivasan and Chiel, and they can, with some additional effort, combine several synapses into one closed algorithmic form. Bernard et al. (1994) finally found a simplified description of the conductance changes occurring at an NMDA channel based on differential equations, but their algorithm is much more complicated than the one proposed here.

Therefore, we have compared our approach with that of Srinivasan and Chiel as well as with that of Lytton, but not with that of Bernard et al.. The three flow diagrams in Figure 2 show the algorithmic complexity of

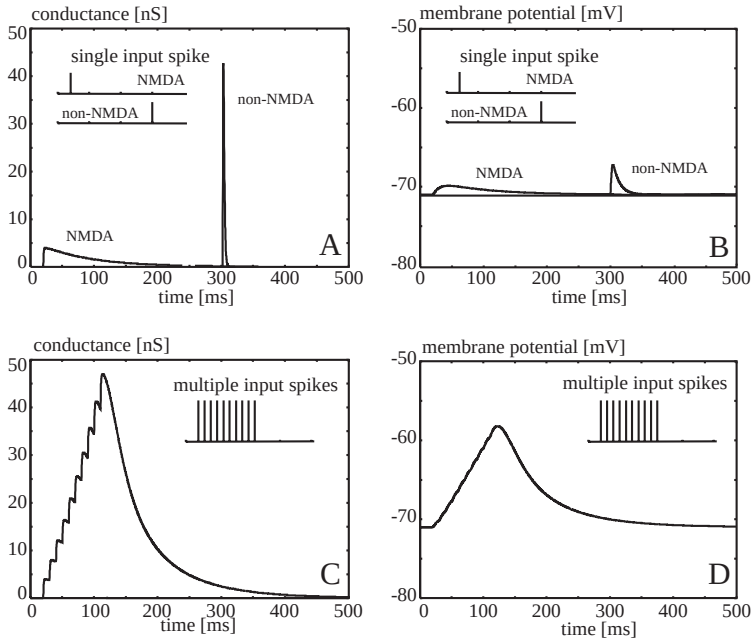

Figure 1: (A) Time course of the conductance of a simulated NMDA channel and a simulated non-NMDA channel for one incoming spike. (B) Corresponding membrane potentials. (C) Time course of the conductance of an NMDA channel during input of a 100 Hz spike train. (D) Corresponding membrane potential.

the approaches when modeling a single synapse. The $\mathcal{Z}$-transform has the simplest and Lytton's algorithm the most complex structure. Since spikes are rare events, the thin pathways are seldomly followed. The thick arrows, on the other hand, represent the passive decay case. Therefore, these command sequences occur much more often and are mainly responsible for the computational time needed. Table 1 compares how many components and operations are required to perform the calculations if a single synapse is modeled. All techniques are fairly similar, and the complexity of Lytton's algorithm pays off by reducing the number of operations to a minimum. The similarity of the three techniques is also reflected in the actual benchmark runs (see Figure 3 and, for the simulation parameters, Table 2).

We simulated a single synapse with different input spike frequencies between 5 and 500 Hz. A total of 50 million time steps was simulated, and each had a duration of 0.05 ms, leading to a total simulated time of 2500 seconds. All simulations were run on a SUN SPARC 5 client machine

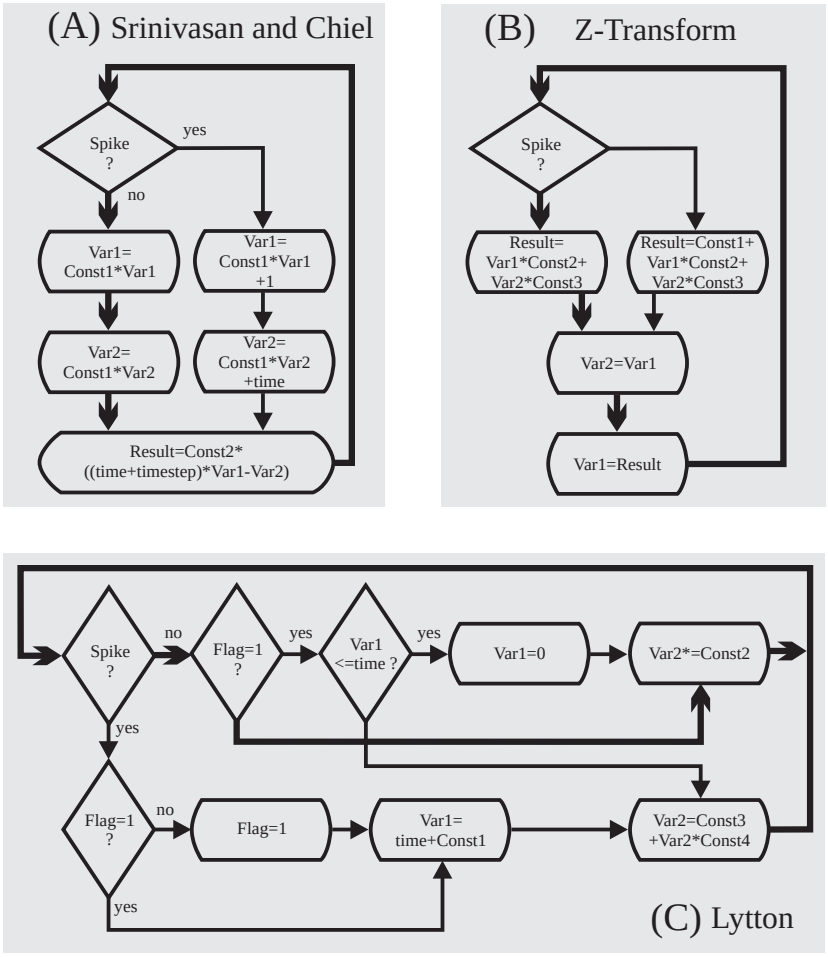

Figure 2: Flow diagrams of the different algorithms for an AMPA synapse (see equation 1.2). (A) The algorithm of Srinivasan and Chiel. (B) The Z-transform. (C) Lytton's algorithm.

in multiuser mode. No disk access was required during the simulations. We did not use any unusual computer operation mode (like "single user mode") in order to make sure that our benchmarks will also apply to an everyday laboratory situation. This can, however, lead to tiny differences when running the same simulation twice due to the differing state of activity

Table 1: Resources Needed for the Different Techniques in the Case of a Single Synapse.

	Srinivasan and Chiel	Lytton	$\mathcal{Z}$-Transform
Memory			
Flags	0	1	0
Variables	2	3	2
Constants	2	4	3
Operations			
Spike	$4*$ $4+$	$1*$ $2+$	$2*$ $2+$
No spike	$4*$ $2+$	$1*$	$2*$ $1+$

Note: $* =$ multiplication; $+ =$ addition, as shown in the flow diagrams in Figure 2.

Table 2: Parameters Used for the Benchmark Simulations.

	Srinivasan and Chiel	Lytton	$\mathcal{Z}$-Transform
Non-NMDA			
τ_1	1 ms	—	1 ms
α_0	—	600 s^{-1}	—
β	—	550 s^{-1}	—
C_{dur}	—	1 ms	—
NMDA			
τ_1	40 ms	—	0.33 ms
τ_2	40 ms	—	0.33 ms

Note: The difference between Lytton's and the other techniques does not make the parameters immediately comparable.

of the system's background demons. In the diagrams we plot the CPU time required for one simulated time step.

As expected from the flow diagrams, the computational effort remains constant for Srinivasan and Chiel and also for the $\mathcal{Z}$-transform. Lytton's technique shows a linear increase in computational time. It is the fastest up to an input frequency of about 210 Hz, which looks very promising because few neurons fire with such a high frequency for an extended period of time.

The problem of computational efficiency becomes truly relevant only in the context of simulating the convergence of multiple inputs onto a single cell (or compartment). This is the case with which Lytton and we are mainly concerned. Thus, the main part of Lytton's algorithm is the optimization of the calculation for many synapses with a shared time constant at a given neuron. He was able to simplify the update to two variables only, and he had to maintain only a single two-valued queue of spike arrival times and synaptic states (instead of N queues in the case of the regular

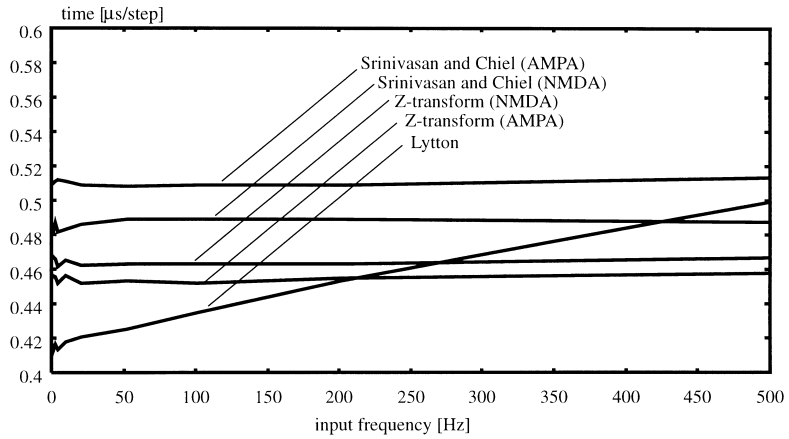

Figure 3: Benchmark comparison of the three techniques for a single AMPA and a single NMDA synapse. Lytton's technique (AMPA only) is the fastest for the single synapse case and reasonable input firing frequencies, but see Figure 4.

convolution). In addition, he required a few bookkeeping commands in his algorithm to keep track of the synaptic state changes. Thus, the algorithmic complexity increases in Lytton's algorithm. On the other hand, when using the $\mathcal{Z}$-transform, only a single one-values queue of the spike arrival times has to be maintained when more than one synapse is simulated. The linearity of the transform allows one to define an accumulated spike train (see equation 1.3), which does not affect the computational effort regardless of the number of synapses (the value of N in equation 1.3). And this also holds for synapses with different delays because a synaptic delay Δt can be implicitly included in the accumulated spike train by reassigning the time variable for this synapse to $t_{resultant} = t_{arrival} + t_{delay}$. Consequentially, although Lytton had found quite an ingenious way to do the bookkeeping and the update, it is rather obvious that the $\mathcal{Z}$-transform will be faster since it remained basically unchanged[2] (see Figure 4). It can be seen in Figure 4A that the $\mathcal{Z}$-transform is significantly faster than Lytton's algorithm when simulating 50 synapses. The difference between the curves, however, is largely independent of the number of simulated synapses (see Figure

[2] Obviously both algorithms require a queue for the inputs. Thus, the $\mathcal{Z}$-transform will be slightly slower. But apart from this, the $\mathcal{Z}$-transform does not need any other algorithmic addition, as opposed to Lytton's algorithm, which requires bookkeeping.

4B). As expected, it is due only to the additionally introduced bookkeeping in Lytton's algorithm and produces a constant computational overhead. This shifts the curves with respect to each other such that they do not cross anymore (compare Figure 3, where the bookkeeping was not implemented because only a single synapse was simulated). Note that in (probably) all real simulation problems, Lytton's algorithm can never be implemented without the bookkeeping because a convergence of more than one synapse onto any compartment must always be allowed. Thus, the performance advantage for the single synapse case (see Figure 3) may actually never be attainable in a real simulation problem where a situation similar to Figure 4 applies. The additional small linear increase in Lytton's curve results from the total increase in incoming spikes since all simulated synapses receive 100 Hz input (compare again Figure 3).

Srinivasan and Chiel did not optimize their algorithm for more than one synapse. Therefore, their algorithm is much slower than the two other ones (not shown).

4 Discussion

Based on the technical report of Olshausen (1990), Brettle and Niebur (1994) made use of the $\mathcal{Z}$-transform in order to calculate synaptic conductance more efficiently, but in their article, they did not give a full account of this technique. It may be for this reason that this technique is still not widely appreciated. Therefore, we have tried to give an account of how the $\mathcal{Z}$-transform could be used even in the context of the more complicated NMDA synapse in order to arrive at a rather simple iterative procedure that reduces the computational complexity and improves speed by at least two orders of magnitude as compared to a straightforward calculation of the convolutions. The comparison of the $\mathcal{Z}$-transform with other algorithms shows that its algorithmic complexity is minimal and that its performance exceeds that of the other techniques significantly as soon as multiple inputs are considered. Lytton's approach, on the other hand, has a more solid physiological foundation because actual transmitter application durations can also be implemented in his model.

It should be pointed out that—as opposed to many other approaches—the $\mathcal{Z}$-transform can be applied to more than one type of synapse. The solutions for a regular non-NMDA synapse and the NMDA synapse were given here, but it is easy to see that, for example, the equation that describes the calcium-activated current through the respective synapse in the mollusc Tritonia and other species can also be reduced by means of the $\mathcal{Z}$-transform. The original equation is given by Getting (1989):

$$g_c(t) = \frac{\tau_d}{\tau_d - \tau_o}(e^{-t/\tau_d} - e^{-t/\tau_o}), \tag{4.1}$$

and the $\mathcal{Z}$-transform follows the same steps as above.

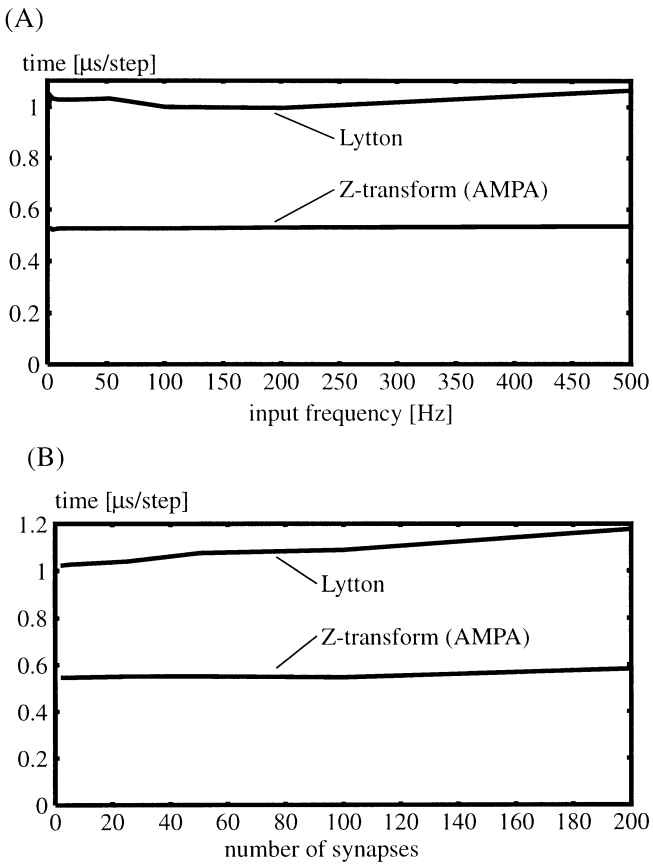

Figure 4: Benchmark for more than one synapse comparing the algorithm of Lytton with the $\mathcal{Z}$-transform. (A) Number of synapses fixed at 50 and variable average input frequency. (B) Input frequency fixed at 100 Hz and variable number of synapses.

It is immediately obvious that the $\mathcal{Z}$-transform can be applied to all synaptic functions that consist of multiplicative or additive combinations of exponential terms.

The largest gain in computational efficiency of the $\mathcal{Z}$-transform is achieved when many synapses converge onto a single compartment. But even in a cell model with a high degree of compartmentation, the $\mathcal{Z}$-transform may be preferential because any applicable version of Lytton's algorithm will al-

ways require the bookkeeping procedure, which reduces its computational efficiency to below that of the $\mathcal{Z}$-transform in almost all situations. Performance differences in a model with a high degree of compartmentation may be marginal, though.

The actual choice of an algorithm for modeling a synapse should therefore be guided by several factors:

1. *Do I need highly realistic synapses plus multiple voltage-dependent channels?* Then probably none of the discussed algorithms can be used and the differential equations need to be implemented explicitly.
2. *Can I live with less realism and use a fast pooling approach (little compartmentation) but still with an explicitly modeled transmitter release?* Then use Lytton's.
3. *Do I need maximal speed and a high degree of pooling?* Then use the $\mathcal{Z}$-transform.

In particular when modeling a large or very large network, the performance gain factor of about 2 between the $\mathcal{Z}$-transform and Lytton's algorithm will certainly be beneficial, because individual simulations often take more than 10 hours. In some special cases, even more abstract connections can be defined—for example, by following the approach of Amit and Tsodyks (1991), which models the membrane potential in a nonstandard way and is basically confined to rates and currents. Due to their intrinsic assumptions, however, this approach cannot be directly compared with the other algorithms studied here.

Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft and the Human Frontier Science Program RG0035/1997-B. C-functions for all three techniques for AMPA and NMDA synapses can be obtained via ftp from ftp://www.neurop.ruhr-uni-bochum.de/pub/z-trans/.

References

Amit, D. J., & Tsodyks, M. V. (1991). Quantitative study of attractor neural networks retrieving at low spike rates: I. Substrate-spikes, rates and neuronal gain. *Network, 2*, 259–273.

Bernard, C., Ge, Y. C., Stockley, E., Willis, J. B., & Wheal, H. V. (1994). Synaptic integration of NMDA and non-NMDA receptors in large neuronal network models solved by means of differential equations. *Biol. Cybernet., 70*, 267–273.

Brettle, D., & Niebur, E. (1994). Detailed parallel simulation of a biological neuronal network. *IEEE Comp. Sci. Eng. 1*, 31–43.

Desthexe, A., Mainen, Z. F. & Sejnowski, T. J. (1994a). An efficient method for

computing synaptic conductances based on a kinetic model of receptor binding. *Neural Comp., 6*, 14–16.

Destexhe, A., Mainen, Z. F. & Sejnowski, T. J. (1994b). Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic mechanism. *J. Comp. Neurosci., 1*, 195–230.

Doetsch, G. (1967). *Anleitung zum praktischen Gebrauch der Laplace-Transformation und der Z-Transformation*. Munich: R. Oldenbourg Verlag.

Getting, P. (1989). Reconstruction of small neural networks. In C. Koch & I. Segev (Eds.), *Methods in neuronal Modeling*, (pp. 171–195). Cambridge, MA: MIT Press.

Jahr, C. E., & Stevens, C. F. (1990a). A quantitative description of NMDA receptor-channel kinetic behavior. *J. Neurosci., 10*, 1830–1837.

Jahr, C. E., & Stevens, C. F. (1990b). Voltage dependence of NMDA-activated macroscopic conductances predicted by single-channel kinetics. *J. Neurosci., 10*, 3176–3182.

Lester, R. A., & Jahr, C. E. (1992). NMDA behavior depends on agonist affinity. *J. Neurosci., 12*, 635–643.

Lytton, W. W. (1996). Optimizing synaptic conductance calculation for network simulations. *Neural Comp., 8*, 501–509.

Mel, B. (1992a). NMDA-based pattern discrimination in a modeled cortical neuron. *Neural Comp., 4*, 502–517.

Mel, B. (1992b). *Information processing in an excitable dendritic tree* (CNS Memo No. 17). Pasadena, CA: California Institute of Technology.

Olshausen, B. (1990). *Discrete-time difference equations for simulating convolutions* (Tech. Memo). Pasadena: California Institute of Technology.

Oppenheim, A. V. & Schafer, R. W. (1975). *Digital-signal processing*. London: Prentice Hall International.

Srinivasan, R., & Chiel, H. J. (1993). Fast calculation of synaptic conductances. *Neural Comp., 5*, 200–204.

Stern, P., Edwards, F. A., & Sakmann, B. (1992). Fast and slow components of unitary EPSCs on stellate cells elicited by focal stimulation in slices of rat visual cortex. *J. Physiol., Lond., 428*, 707–722.

Tråvén, H. G. C., Brodin, L., Lansner, A., Ekeberg, Ö., Wallén, P. & Grillner, S. (1993). Computer simulations of NMDA and non-NMDA receptor-mediated synaptic drive: Sensory and supraspinal modulation of neurons and small networks. *J. Neurophysiol., 70*, 695–709.

Received June 6, 1997; accepted March 2, 1998.