A tool for synthesizing spike trains with realistic interference

Leslie S. Smith [*], Nhamoinesu Mtetwa

Department of Computing Science and Mathematics, University of Stirling, UK

Abstract

Spike detection and spike sorting techniques are often difficult to assess because of the lack of ground truth data (i.e., spike timings for each neuron). This is particularly important for in vitro recordings where the signal to noise ratio is poor (as is the case for multi-electrode arrays at the bottom of a cell culture dish). We present an analysis of the transmission of intracellular signals from neurons to an extracellular electrode, and a set of MATLAB functions based on this analysis. These produce realistic signals from neighboring neurons as well as interference from more distant neurons, and Gaussian noise. They thus generate realistic but controllable synthetic signals (for which the ground truth is known) for assessing spike detection and spike sorting techniques. They can also be used to generate realistic (non-Gaussian) background noise. We use signals generated in this way to compare two automated spike-sorting techniques. The software is available freely on the web.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Spike detection; Spike sorting; Synthetic data

1. Introduction

The primary aim of this tool is to synthesize signals which simulate those that electrodes in extracellular multi-electrode arrays (MEAs) record in culture. The motivation for this tool is to provide "ground truth" data (that is, data for which the actual spiking situation is known, named "ground truth" because of the similar problem in remote sensing) for the testing of algorithms for spike detection and spike sorting. For data acquired using MEAs, this information could be provided through voltage sensitive dye-based imaging or concurrent intracellular recording of the neurons of interest. Providing the time resolution required using dye based imaging is difficult since it implies very high frame rates. Concurrent intracellular recording is difficult for MEAs, although it has been done for a single neuron and a tetrode electrode array by Harris et al. (2000). The difficulty we are attempting to overcome is that without ground truth data, it is not possible to compare different spike detection and spike sorting techniques (many techniques are reviewed in Lewicki (1998)). This is a difficult problem: Wood et al. (2004) report over 20% errors using semiautomated spike sorting. Many authors have generated ground truth data by taking recordings which contain spikes, and mixing them and adding noise (e.g., Atiya, 1992; Chandra and Optican, 1997; Quiroga et al., 2004; Zhang et al., 2004), or re-generating data with similar statistics (Wood et al., 2004). This presumes that suitable data already exist, and that the form that noise should take is known. A different approach has been taken by Menne et al. (2002); Mamlouk et al. (2005): they used GENESIS (Bower and Beeman, 1998) to create a multi compartment simulation of relevant neural circuitry (in hippocampus), and then summed the contribution from each compartment assuming homogenous resistivity and no capacitance.

We are interested in generating signals that are useful for testing spike detection and sorting techniques for in vitro recordings, where the underlying neural structure is unknown. In general, all that is known is the approximate density of the neurons. Further, inspection of extracellularly recorded spike shapes shows that these differ from intracellular spike shapes (see, for example, Harris et al. (2000)). We have therefore taken a different approach, and attempted to analyse the signal transfer between the intracellular spike throughout the spiking part of the neuron and the extracellular electrode in order to generate realistic test data for which the underlying spike patterns are known. We use this to synthesize both the signal from those neurons closest to the electrode, and interference from more distant neurons.

A related approach was taken by Nakatani et al. (2001) in the context of cuff electrodes. For in vitro multi-electrode array

---

[*] Corresponding author. Tel.: +44 1786 467435; fax: +44 1786 464551.
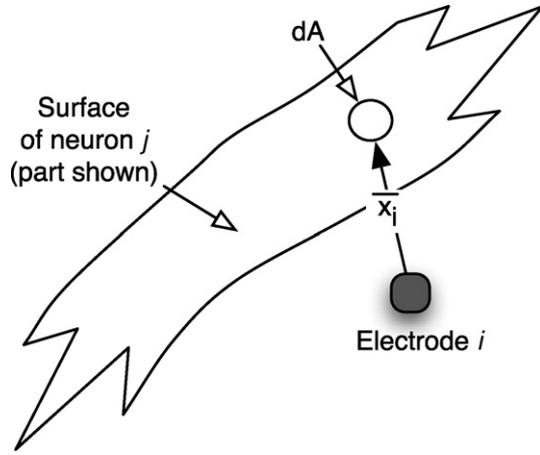 *E-mail address:* lss@cs.stir.ac.uk (L.S. Smith).

Fig. 1. Electrode *i* and patch of membrane.

(MEA) based recording, the culture is grown in dishes on which the electrodes are already present. Generally, the culture contains a mixture of neurons and glial cells, and there may be glial cells between the electrode and the neurons. Further, each electrode is usually near enough to a number of neurons to pick up signals from them, and the electrodes may not be moved to improve signal strength. Unlike the situation in Menne's work, the electrodes are generally too far apart to pick up signals from the same neurons. Thus, ICA is not a usable option. MEA recording necessitates automated techniques, simply because of the volume of data.

The signal received at an electrode from a neuron is transformed by both the neuron's geometry relative to the electrode and the effect of the path from spiking neuron to electrode. The details of the transformation are discussed in Section 2. We have developed a set of MATLAB routines which allow the user to emulate this transformation for a number of neurons. Our technique allows both precise noise control and precise control of the spike shapes collected at the electrode.

This paper is organised as follows: Section 2 contains a theoretical discussion of the nature of the signal picked up by an electrode. Section 3 discusses how to use this theory for generating signals which are like those from extracellular electrodes, and Section 4 assesses some spike sorting techniques using signals generated using this software. The software, and a user manual for it, are available (Smith and Mtetwa, 2006).

2. An analysis of extracellular cell recording

2.1. Effect of transmembrane current

We attempt to characterise the signal received at an extracellular electrode. Consider a current $I_{dA}(t)$ passing through a patch of membrane (from outside the neuron to inside the neuron), dA at location $\vec{x}_i = \vec{x}_i(d, A)$, where $\vec{x}_i$ is the vector from electrode i to the patch dA (see Fig. 1).

This will have an effect at electrode i, leading to a voltage $v_i(dA,t)$ being generated. Let us assume that the effect is linear in $I_{dA}(t)$. If we characterise this effect by a response function $r(\vec{x}_i)$ then the resultant potential at electrode i, will be:

$$v_i(dA, t) = -I_{dA}(t)r(\vec{x}_i(dA)) \tag{1}$$

where the negative sign arises because the current flows into the neuron. We are assuming that the extracellular fluid is ohmic (see Fig. 2). Using linearity, we can consider the effect of the currents from a whole neuron j, with surface N_j,

$$v_i(N_j, t) = -\int_{dA \in N_j} I_{dA}(t)r(\vec{x}_i(dA))dA \tag{2}$$

noting that $I_{dA}(t)$ will vary both in timing and size across N_j. The total effect for all contributing neurons (indexed by j) is then:

$$v_i(t) = \sum_j v_i(N_j, t) = -\sum_j \int_{dA \in N_j} I_{dA}(t)r(\vec{x}_i(dA))dA \tag{3}$$
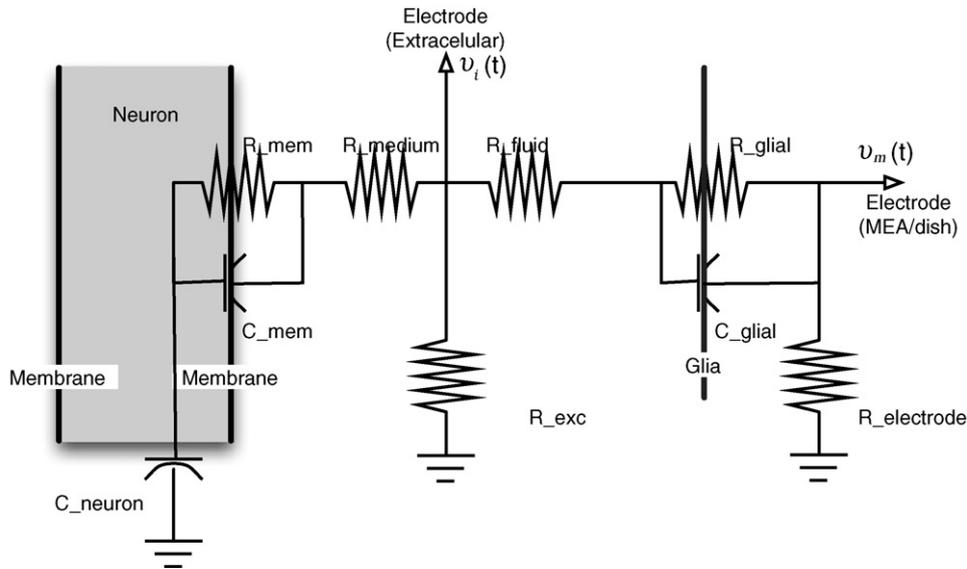

Fig. 2. Equivalent circuit description for transfer of charge from a point on a neuron to an electrode for an extracellular and a dish based electrode. The extracellular electrode is assumed to be near the neuron, but the dish electrode is assumed to have a layer of glia between it and the neural culture. There are a number of simplifications in this circuit: distributed resistances and capacitances have been lumped together, for example.

Note that $v_i(t)$ is the voltage arriving at the electrode i from the neurons: what actually gets recorded may differ. Some of the currents $I_{dA}(t)$ come from spikes (primarily those from the axon hillock and axons), and some come from non-spiking parts of the neuron surface.

Writing Ns_j for the spiking and Nns_j for the non-spiking part of neuron we have:

$$v_i^{\text{spike}}(N_j, t) = -\int_{dA \in Ns_j} I_{dA}(t) r(\vec{x}_i(dA)) dA \qquad (4)$$

and

$$v_i^{\text{non-spike}}(N_j, t) = -\int_{dA \in Nns_j} I_{dA}(t) r(\vec{x}_i(dA)) dA \qquad (5)$$

2.2. Action potentials

We can now attempt to characterise $v_i^{\text{spike}}(N_j, t)$ due to a single spike starting at time T in neuron N_j. Given that "at time T" means that T is the time of the initiation of the spike at the axon hillock (i.e., the start of the self-reinforcing Na^+ inflow), the spike will have no effect on $v_i^{\text{spike}}(N_j, t)$ for $t < T$. The spike will be transmitted from the soma down the branching axon: we assume that there is an upper bound, $T_{\max}$, on the duration of the effect of the spike. At each point y on the spiking membrane, the upswing of the spike is generated by runaway inflow of Na^+ ions, from the rapidly inactivating $I_{Na,t}$ channels, and the downswing by the outflow of K^+ ions from the I_K channels (Koch, 1999): these are the $I_{dA}(t)$ in Eqs. (1–5). These charge and discharge the intracellular (conducting) fluid of the neuron (which is largely insulated from the extracellular fluid by the insulating bilipid membrane). Thus, writing: $V_{\text{spike}}(y, t)$ for the intracellular spike at location y (with respect to an external ground), and ignoring ohmic conduction within the neuron

$$V_{\text{spike}}(y, t) = \int dt \int_{dA \in \text{nbhd}(y)} \frac{I_{dA}(t) dA}{C_y}$$
$$= \int dt \int_{dA \in \text{nbhd}(y)} \frac{(I_{dA}^{\text{Na}}(t) + I_{dA}^{\text{K}}(t)) dA}{C_y} \qquad (6)$$

where nbhd(y) is the membrane near y and C_y is the membrane capacitance of nbhd(y), so that:

$$V_{\text{spike}}'(y, t) = \frac{(I^{\text{Na}}(y, t) + I^{\text{K}}(y, t))}{C_y} \qquad (7)$$

where $I^{\text{Na}}(y, t)$ and $I^{\text{K}}(y, t)$ are the ionic currents integrated over nbhd(y).

These ionic currents alter the extracellular potential, as described in Eq. (4). Thus:

$$v_i^{\text{spike}}(N_j, t, \text{ion}) = -\int_{dA \in Ns_j} (I^{\text{Na}}(dA, t)$$
$$+ I^{\text{K}}(dA, t)) r(\vec{x}_i(dA)) dA \qquad (8)$$

Eqs. (7) and (8) show that the voltage at the electrode due to ionic currents from spikes in some small neighbourhood on the

spiking surface of a neuron is essentially minus the derivative of the intracellular spike weighted by the response function $r(\vec{x}_i)$. Clearly, the ionic currents will differ in differing parts of the cell. Further, for an extracellular electrode near the cell surface, $r(\vec{x}_i(dA))$ will tend to be large when $\vec{x}_i(dA)$ is small, leading to considerable variation depending on electrode location, as discussed in (Holt and Koch, 1999; Moffitt and McIntyre, 2005), where the extracellular signal is strongest nearest the spike initiation point.

In addition, there is also the electrical resistance and the capacitance of the membrane to take into account. The resistance of the membrane and the resistance between the extracellular fluid and the ground form a potential divider: in addition, the capacitance of the membrane and the resistance between the extracellular fluid and the ground act as a differentiator (see Fig. 2). Writing $v_i^{\text{spike}}(y, t, \text{mem})$ for the extracellular voltage from this source from a neighbourhood of y, we have:

$$v_i^{\text{spike}}(y, t, \text{mem})$$
$$= \frac{R_{\text{exc}}}{R_{\text{exc}} + R_{\text{medium}} + R_{\text{men}}} V_{\text{spike}}(y, t)$$
$$+ \frac{R_{\text{exc}}}{R_{\text{exc}} + R_{\text{medium}}} C_{\text{mem}} V_{\text{mem}}'(y, t)(R_{\text{exc}} + R_{\text{medium}}) \qquad (9)$$

where $V_{\text{mem}}'(y, t)$ is the derivative of the voltage across the membrane at y, R_{exc} is the resistance from extracellular electrode to ground, R_{medium} is the resistance of the medium between the extracellular electrode and y, and R_{mem} is the resistance across the membrane at y. Since $R_{\text{mem}} \gg R_{\text{exc}} + R_{\text{medium}}$, the contribution from the original spike will be small. Given that the electrode is near the membrane, the potential divider described by the fraction at the start of the second term will be near to unity. The overall effect at the extracellular electrode i from the spike on neuron N_j is therefore:

$$v_i^{\text{spike}}(N_j, t) = v_i^{\text{spike}}(N_j, t, \text{ion})$$
$$+ \int_{\text{nbhd}(y) \in Ns_j} v_i^{\text{spike}}(y, t, \text{mem}) \, dy \qquad (10)$$

and these two terms will have opposite signs.

For a patch clamp electrode, for a small patch of membrane only, R_{exc} is replaced by a capacitor, so that Eq. (8) no longer holds, but instead $v_i^{\text{spike}}(N_j, t, \text{ion})$ is proportional to (but of opposite polarity from) $V_{\text{spike}}(y, t)$ (and the rest of the neuron has no effect). In addition, V_{mem} and V_{mem}' are normally held at 0, and the currents involved in $R_{\text{exc}}/R_{\text{exc}} + R_{\text{mem}} V_{\text{spike}}(t)$ are relatively small, so that the ionic currents dominate.

For an electrode at the bottom of a culture dish, there is a further complication: glial cells are likely to form a (possibly incomplete) layer between the electrode and the neural culture. This layer may have a high resistance (R_{glial}), and will also act as a capacitor (C_{glial}) (see Fig. 2). However, in this case, we do not have ionic channels contributing to the potential at the electrode, so that only Eq. (9) is relevant. Writing $v_k^{\text{spike}}(N_j, t)$ for the voltage on electrode k at the bottom of the dish, the effect

of the voltage at electrode i is:

$$v_k^{\text{spike}}(N_j, t) = \frac{R_{\text{electrode}}}{R_{\text{electrode}} + R_{\text{fluid}} + R_{\text{glial}}} v_k^{\text{spike}}(N_j, t)$$
$$+ C_{\text{glial}} R_{\text{electrode}} v_k'^{\text{spike}}(N_j, t) \qquad (11)$$

Of course, we do not have both electrodes i and k. In reality, $v_k^{\text{spike}}(N_j, t)$ results from the integration of $v_k^{\text{spike}}(N_j, t)$ over a small volume of the intracellular fluid. The primary effect of this is a further small amount of low-pass filtering.

What we actually receive at electrode k (whether extracellular or MEA based) is $s_k(t)$ which consists of $v_k^{\text{spike}}(N_j, t)$ from many different neurons N_j plus an additional noise signal, $n(t)$ from the receiving apparatus itself. Focussing on the signal from one neuron, N_j we can write this as:

$$s_k(t) = v_k(t) + n(t) = v_k(N_j, t) + \sum_{p \neq j} v_k(N_p, t) + n(t)$$
$$= v_k^{\text{spike}}(N_j, t) + \sum_{p} v_i^{\text{non-spike}}(N_p, t)$$
$$+ \sum_{p \neq j} v_k^{\text{spike}}(N_p, t) + n(t) \qquad (12)$$

The first term is the "signal", the next two are interference (in the sense that they do not originate from the spike at neuron j), and the last is noise. Both the signal and the interference term in Eq. (12) depend on the nature of the connection between the neurons and the electrode k, as described in Eqs. (6–11). Part will be resistive, mediated by the ionic conduction of the medium part will be capacitive (and differentiating) due to the insulating membrane and the ion channels.

2.3. Modelling the effect of the extended spiking neural surface

What is the effect of the integration over the extent of the neuron? Because the time taken for spike movement from the spike initiation point on the soma through the axon is comparable to or larger than the spike duration, this integration will have a major effect on the shape of the voltage recorded at the electrode.

The spiking neural surface is a three-dimensional surface whose orientation and distance from the electrode varies. The signal contribution in Eq. (12), $v_k^{\text{spike}}(N_j, t)$ which comes from Eqs. (8) and (9), can be written:

order, and group the terms:

$$v_k^{\text{spike}}(N_j, t) = \frac{\text{d}}{\text{d}t} \int_{dA \in N_j} V_{\text{spike}}'(dA, t)(a_2 - r(\vec{x}_i))(dA)$$
$$+ \int_{dA \in N_j} a_1 V_{\text{spike}}(dA, t) dA \qquad (14)$$

We can approximate these integrals discretely. Further, we note that because of the mechanism of spike generation, $V_{\text{spike}}(dA, t)$ (and also therefore $V_{\text{spike}}'(dA, t)$) always has the same shape to a first approximation. We therefore write:

$$V_{\text{spike}}(dA, t) = V_{\text{spike}}(dA_0, t - \Delta(dA)) \qquad (15)$$

where dA_0 is the spike initiation point (axon hillock) and $\Delta(dA)$ is the time the spike takes to reach dA from the axon hillock. As a result, Eq. (14) becomes:

$$v_k^{\text{spike}}(N_j, t) = \frac{\text{d}}{\text{d}t} \sum_{dA} V_{\text{spike}}'(dA_0, t - \Delta(dA))(a_2 - r(\vec{x}_i))$$
$$+ \sum_{dA} V_{\text{spike}}(dA_0, t - \Delta(dA))a_1 \qquad (16)$$

In order to evaluate this, we can consider summing over time rather than area: each time interval δt may contain signals from a number of areas of the spiking surface

$$v_k^{\text{spike}}(N_j, n\delta t) = \sum_{i=1...k} V_{\text{spike}}'(t - i\delta t)b_2(i)$$
$$+ \sum_{i=1...k} V_{\text{spike}}(t - i\delta t)b_1(i) \qquad (17)$$

where b_1 and b_2 are appropriate sums of the a_1's, and a_2's and $r(\vec{x}_i)$ and $k = [\Gamma_{\text{max}}/\delta t]$. We note that the time intervals over which we sum need to be quite small, particularly since the spiking signal changes very rapidly. In fact, we should choose the time intervals to be smaller than half the duration of the most rapidly changing component of interest in the signal. This time interval is theoretically determined by the highest frequency component of the $V_{\text{spike}}'(t)$. Real neural signals contain energy content up to relatively high frequencies for example, the sharp knee in Fig. 1a of Naundorf et al. (2006). However, in general the fastest component of interest is the spike rise time, which is usually upwards of 200 μs (Naundorf et al., 2006; Zhang, 2004).

In Eq. (8), $r(\vec{x}_i(dA))$ (which contributes negatively to b_2) decreases exponentially with distance due to diffusion, and in Eq. (9), R_{medium} (which contributes inversely but positively to both b_1 and b_2) increases linearly with distance. Where the neuron is far from the electrode, it is likely that the b_2 and b_1 change

$$v_k^{\text{spike}}(N_j, t) = v_k^{\text{spike}}(N_j, t, \text{ion}) + v_k^{\text{spike}}(N_j, t, \text{mem}) = -\int_{dA \in N_j} I_{dA}(t) r(\vec{x}_i)(dA) + \int_{dA \in N_j} a_1 V_{\text{spike}}(dA, t) dA$$
$$+ \int_{dA \in N_j} a_2 V_{\text{spike}}'(dA, t) dA = \int_{dA \in N_j} V_{\text{spike}}'(dA, t)(a_2 - r(\vec{x}_i))(dA) + \int_{dA \in N_j} a_1 V_{\text{spike}}(dA, t) dA \qquad (13)$$

since $V_{\text{spike}}'(dA, t) = I_{dA}(t)$, and a_1 and a_2 (which are functions of dA) replace the fractions in Eq. (9). Noting that integration and differentiation are linear operations, we can reverse their

relatively slowly. The effect of the integration over the surface of the neuron will be to low-pass filter the signal (Struijk and Yoshida, 2004), because the distance of all the spiking parts of

the neuron will be nearly the same so that the transfer characteristics will vary little across the spiking surface of the neuron. However, where the neuron is close to the electrode, some of the spiking surface may be much closer to the electrode than other parts, so that there may be rapid variation in b_2 and b_1. Further, the precise nature of the variation depends on the relative geometry of the neuron and the electrode, and will depend strongly on the presence and density of active ion channels near the electrode. Thus, the shape of the signal received will depend on this geometry, and in particular, if part of the spiking surface is very close to the electrode this may have a major effect on the shape of the spike recorded. This agrees with the theory in (Holt and Koch, 1999; Moffitt and McIntyre, 2005), and the measurements in (Harris et al., 2000). One effect of this is that the detected spike shape is likely to differ for different neurons, particularly those closest to the electrode, even if the intracellular spike shapes are identical, thus assisting spike sorting.

2.4. Dish based MEA electrodes

For MEA type electrodes at the bottom of a culture dish, transmission of signals is likely to be complicated further by insulating glial cells between the neuron and the electrode causing a further mixture of resistive and capacitive (differentiating) coupling. Further, the potential will also depend on the nature of the transferrence of ionic current in the medium to electrical current in the electrode (not modelled here: (see Standen et al., 1987)).

The final result is that an MEA electrode detects a mixture of the intracellular spike, its first and second derivatives, arriving from the different parts of the spiking surface of the neuron. (We note that if there are other glial cells in the signal path, there may be further differentiation occurring as well.) If the neuron is relatively far away, the neuron spatial extent will smooth the signal: if it is nearer, then the precise shape of the signal will depend on the relative geometry of the neuron and the electrode. For situations in which the electrode itself may be maneuvered, the value of $\vec{x}_i$ (in Eq. (4)) can be reduced, improving the overall conductance between dA and the electrode (thus improving SNR), and causing the signal to come primarily from one section of the spiking area, thus reducing the degree of smoothing. One might also attempt to place the electrode where the capacitive effects of glial cells are minimised (for example by shaping points on the electrodes, or by lowering the electrodes on to the culture (Sandison et al., 2002)).

The interference terms in Eq. (12) identify two components. The first (and probably smaller of these) arises from non-spiking events from the neurons whose spikes we are attempting to detect. $I_{dA}(t)$ in these events is generally quite small, so that the contribution at the electrode is large only if $r(\vec{x}_i(dA))$ is large, which is likely to be the case only if dA is very close to the electrode (or if a patch is made to a non-spiking part of the electrode). Assuming that we are not using patch clamping, then this interference arises from synaptic events and non-axonic spikes (e.g., Ca^{2+} spikes). The second interference term is likely to be the dominant term: this arises from spikes in other neurons, relatively further away. In many situations, neurons are closely

packed, so that if it has not been possible to place the electrode very close to the neuron of interest, interference from other neurons will be dominant. The modes of transmission are as discussed earlier (and may differ in relative strength for different neurons). The final noise term arises from extraneous electromagnetic interference, and from the amplifiers used, and can be minimised by shielding and appropriate experimental design.

The distribution of the interference and noise terms is likely to be quite different for each of the three: for the first interference source, it is likely that these will be synaptic activity before and during spiking. For the second interference source, it is likely that some neighboring neurons receive similar input to the neuron of interest, so that they fire at similar times. Thus both of these interference sources will be correlated to the signal of interest. The spike times from other (and further away) neurons are likely to be relatively independent of the neurons of interest. Only the noise source is likely to be entirely uncorrelated.

3. Generating data for analysis

The primary aim of this work is to enable the comparison of a number of different spike detection and sorting techniques. We have produced a set of MATLAB functions to generate realistic data, using the analysis in Section 2 to guide data generation. The functions have a large number of variable parameters, enabling generation of a large variety of datasets. Synthetic data with realistic signals from the target neurons and which has a large quantity of realistic interference (that is, realistic data from other neurons) can be generated. The spiking of some of the interfering neurons can be correlated with target neurons, while other interfering neurons can be spiking independently. The parameters for each neuron can be set independently.

The data generation system has a number of phases: firstly, the spike times for the neurons of interest (target neurons) are generated, within some time interval. Spikes may be generated using either a Poisson or a Gaussian distribution. The mean inter-spike interval, and the degree of randomness of these spikes are adjustable parameters. Minimum inter-spike intervals can be enforced. Once these spike times have been determined, we can generate spike times for other neurons (correlated neurons) each of whose spike trains are correlated with one of the target neurons. The spike times for these correlated neurons are determined by allowing a (selectable) degree of jitter on the spike times of the (selected) target neuron. The jittered spike times may be distributed either normally around the original spike time (with selectable standard deviation), or uniformly, with a selectable maximum difference in time. The software allows generation of other sequences of spike times uncorrelated with the target spike times (independent neurons), again permitting Poisson or Gaussian distributions. We thus have one set of sequences of spike times, from the target neurons, plus two other sets of spike times, one correlated with the originals, and one not.

We use realistic intracellular spike shapes, (currently taken from (Naundorf et al., 2006) or from the Hodgkin Huxley simulator HHSim (Touretzky et al., 2004), but able to be taken from elsewhere) to generate the intracellular potential for each of these

spike trains. The effect of the spatial extent of the neuron is then recreated using Eq. (17): weighted delayed spikes are summed, with the actual weights set by the user. The delay default is set to 30 μs (it is user-selectable and should be a multiple of the sample period), allowing contributions up to about 16 kHz. Logically, we should compute the signal for the whole time interval at the axon hillock, then perform this delay and sum: however, computing the signal spike by spike is much more efficient (particularly for long intervals with relatively few spikes). Where the inter-spike interval is less than the length of the (delayed-and-summed) spike shape, we join them smoothly, and in such a way that both reach their maximum potential.

We then generate first and second differentials for all of these signals, and set all these signals and differentials linearly to a normalised amplitude. For each neural signal source (target neurons, correlated and uncorrelated neurons), we form a signal by linearly mixing the weighted, summed intracellular signals and first and second differentials: the mixing parameters are set by the user (and may be different for each neuron). These linearly mixed signals (characteristic of each spiking neuron) can then themselves be linearly mixed. This allows any required target signal to correlated or uncorrelated interference ratio. Gaussian random noise (of selectable size) can be added at the end. The final signal can then be scaled to an appropriate range. In this way, we can generate a noisy signal similar to that which would be picked up by a real electrode, but for which the actual timing of the spikes is known, and the precise form and extent of the noise can be adjusted.

The software can also be used to generate "spiky" background noise. To do this, the number of target neurons and correlated neurons are set to zero. Such noise could be used to test spike detection and sorting algorithms on a mixture of real data and appropriate noise. For example, using 15 uncorrelated neurons, with a mixture of Gaussian and Poisson distributions, where each neuron has a mean inter-spike interval of between 25 and 40 ms, and using a completely flat set of weightings, we can generate spiky data which is made up of many overlapping spikes, and has a kurtosis of 4.08. This can be tailored to the user's needs, for example by using fewer or more neurons, or altering the inter-spike interval, or altering the delay weightings.

Further details of the noisy spike generation system, including a detailed discussion of the parameters (and MATLAB code) may be found in (Smith and Mtetwa, 2006).

4. Assessing two spike sorting techniques

We have generated sets of data both without and with interference to compare the capabilities of two existing spike sorting techniques. We used Wave_clus (Quiroga et al. (2004)) to compute the spike times: this uses a simple threshold technique, setting the threshold based on the median value of the signal. This is then used to extract a 64-element sample vector (with 20 samples before and 44 samples after the threshold event). At the sampling rate used (24 Ksamples/s) this represents a window of 2.67 ms. We used Wave_clus's capability to generate both the first three PCA component coefficients, and a 10-element wavelet based representation of these 64-element vectors.

The PCA and wavelet based vectors were used as input to both KlustaKwik Version 1.7 (Harris, 2003) (which uses the CEM algorithm (Celeux and Govaert, 1992)) and Wave_clus's spike sorting facility (which uses a supraparamagnetic clustering technique, described in Quiroga et al. (2004)) to cluster and then sort the spike trains. The way we used these reflects our interest in automated spike sorting. Supraparamagnetic clustering (SPC) is a stochastic technique, so that different runs can produce different results. We simply used the first result, since an experimenter may not be able to automate assessment of what is an appropriate result (that is, they may not know the correct number of spike trains). (We do note, however, that, in general, there is little difference between the first and subsequent SPC runs.) We did not take advantage of Wave-clus's capability for forcing reclassification.

Using this software, we can create waveforms for target neurons which differ in a controlled way, and we can add specific amounts of correlated and un-correlated interference to produce realistic extracellular signals. We used two target neurons. One (T1) has a Poisson distribution, and the other (T2) has a Gaussian distribution. Both have identical intracellular spikes (from Naundorf et al. (2006)). The spike shapes at the recording electrode differ because different transfer characteristics and/or spatial extents have been modeled. Two experiments are described below. In the first, the spike shapes received at the electrode from the two target neurons are quite different (both in terms of transfer characteristic and spatial extent), and a number of levels of interference are added. In the second, the spike shapes are different only in transfer characteristic, and one of the spike shapes is gradually made more like the other so that the difference in spike shapes decreases between measurements.

4.1. Using dissimilar spike shapes

The software was used to generate two target spike shapes (which are quite different from each other: see Fig. 3), with varying amounts of additional correlated and uncorrelated interference. The aim was to determine how well the different spike
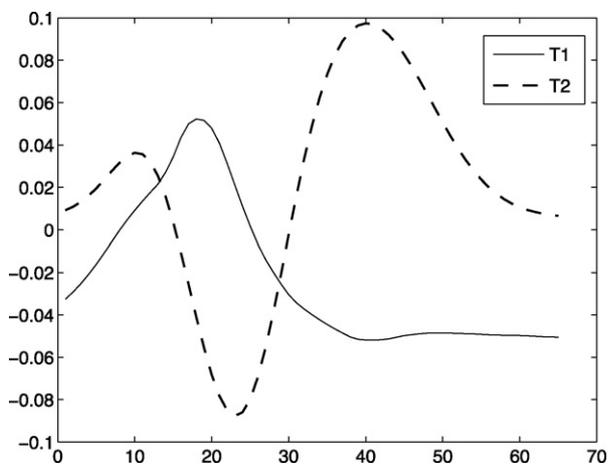

Fig. 3. Shapes of the two dissimilar target neuron spikes (T1 and T2) as received at the putative electrode. *X* axis is in samples at 24 Ksamples/s. *Y* axis is arbitrary voltage-like units.

Table 1
Spike sorting using wavelets and PCA (first three components) for both superpara-magnetic clustering (SPC) and KlustaKwik spike sorting techniques for signals with no interference

Wavelets

Target	SPC				KlustaKwik				
	0	1	2	FoM	0	1	2	3	FoM
T1	50	0	432	0.92	482	0	0	0	0.74
T2	35	573	0		51	99	360	98	

PCA

Target	SPC						KlustaKwik					
	0	1	2	3	4	FoM	0	1	2	3	4	FoM
T1	66	0	253	86	77	0.68	212	0	220	0	50	0.54
T2	106	502	0	0	0		161	373	0	74	0	

The table rows show the number of spikes from each target class (0–4) found in each cluster, and the figure of merit (FoM): see text for description. Two spikes were inserted (due to collisions between spike trains).

sorting techniques could cope with different levels of realistic interference. The sections of electrode signal generated were 30 s long, and contained 482 spikes from T1, and 608 spikes from T2. The results from spike sorting the signal with no added noise are shown in Table 1. In this experiment and the following ones, spikes are marked as missing if no spike was detected within 1 ms of a target spike, and spikes are marked as inserted if no target spike was present within 1 ms of the time of spike detection. We note that missing and inserted spikes are the result of the spike detection, rather than the spike sorting stage. Class 0 is a catch-all (i.e., not clustered) for both SPC and KlustaKwik (KlustaKwik class numbers have been renumbered: class 0 is not used, and class numbers have all been reduced by 1).

In order to make spike-sorting work on noiseless signals in Wave_clus, spike detection had to be set to use the signal mean (rather than median) because so much of the signal had the value exactly 0. This is, of course, not characteristic of real signals. For the wavelet results for KlustaKwik, only the first five coefficients were used as input: using all 10 results in KlustaKwik almost always failing to separate out two clusters (everything is placed in one cluster). We permitted any number of clusters to be generated internally during processing. However, doing this for PCA gave poor results (38 clusters detected!). We therefore set the maximum possible number of clusters to 5 in this case. In later work, we found that this did not have any effect except to avoid multiple spurious clusters. The only exception to this was processing the zero noise wavelet data.

Only the SPC technique applied to the wavelet pre-processed signals detects exactly two clusters. The others all produce more than two.

Table 2
Spike sorting with uncorrelated interference only

SNR	C	N	Wavelet					PCA					M	I
			0	1	2	3	4	0	1	2	3	4		
4.6	S	T1	48	0	336	0	93	44	0	433	0	0	5	7
		T2	41	400	0	167	0	36	572	0	0	0	0	
	K	T1	25	333	101	18	0	36	0	441	0	0	5	
		T2	217	0	0	0	391	33	575	0	0	0	0	
3.2	S	T1	83	0	290	97	0	36	0	434	0	0	12	1
		T2	74	533	0	0	0	53	551	3	0	0	1	
	K	T1	27	57	386	0	0	15	438	2	15	0	12	
		T2	43	0	0	564	0	12	2	560	33	0	1	
2.3	S	T1	22	0	233	1	0	16	0	234	6	0	226	1
		T2	30	512	0	60	0	43	498	0	61	0	6	
	K	T1	15	188	1	51	1	12	8	1	235	0	226	
		T2	14	0	68	0	520	14	73	515	0	0	6	
1.6	S	T1	10	2	31	0	0	3	2	38	0	0	439	9
		T2	13	257	41	0	0	5	257	49	0	0	297	
	K	T1	4	1	1	26	11	3	38	2	0	0	439	
		T2	4	258	39	0	10	7	49	255	0	0	297	

Table rows show clusters found for each target class (0–4). Four different SNRs were used (see text for how the SNR was measured). The column labeled C is the type of clustering (K, KlustaKwik, S, SPC), and the column labeled N is the target neuron. The column labeled M is the number missing (i.e., spikes form this target neuron not found), and the column labeled I is the number inserted (i.e., spikes detected which do not correspond to spikes in the original dataset).

Keeping the same spike times, different types and amounts of interference were added. Wave_clus was set to use the median in spike detection (which is the default). The results from using only uncorrelated interference are shown in Table 2. Again the maximum number of clusters used by KlustaKwik was limited to 5. (Class 5 was never assigned, but decreasing the maximum number of possible clusters to four results in much poorer classification: not specifying the maximum number of clusters results in large numbers of spurious classes). Using all 10 wavelet coefficients, KlustaKwik again failed to differentiate between clusters. The results shown use only the first five wavelet coefficients.

We have calculated SNR using the ratio of the peak levels of the signal to the peak level of the interference, rather than the ratio of signal to interference power. We have done this since neither signal nor interference is Gaussian, and because the target neuron signals are nearly constant (and thus have almost zero energy) most of the time, whereas the interfering signal (being the sum of 50 spike trains) has power much more of the time. As a result, even when the target neurons spikes are much larger than the noise, the ratio of signal to interference power would be less than 1. Even worse, the ratio of signal to interference power would depend on the target neuron-spiking rate. Using peak level ratios for SNR calculation overcomes both these problems. The interference level was increased by $\sqrt{2}$, equivalent to an increase of about 3 dB between tests. We used this in preference to target peak: interference standard deviation (used in Quiroga et al. (2004)) primarily because the latter will also depend on the number of correlated and uncorrelated spike trains, rather than on their spike magnitude.

As the SNR decreases, the number of spikes missed increases. The T2 signal has a larger peak, and so withstands more interference, so that the number of T1 spikes missed increases first.

We note that a relatively modest increase in noise level makes a large difference to the number of spikes missed: this is also visible in Table 3.

Most of the time more than two clusters are detected, and spikes from the same target class are often assigned to different numbered clusters. This, and the default class 0 make it difficult to use Table 2 directly to compare the techniques. To overcome this, we introduce a figure of merit for spike sorting where the ground truth is known. The formula we use is

$$M = \frac{1}{n}\sum_{j=1}^{n}\left((\max_{i} T_j(i) - \sum_{k=1 k\neq j}^{n} T_k(i))/N_j\right) \quad (18)$$

where N_j is the number of spikes from target neuron j, n is the number of target neurons (here, 2), and $T_j(i)$ is the number of spikes from target neuron j found in cluster i. M is measure of the degree to which the different clusters detected follow the different target neurons: it reaches its maximum value of 1 when each target neuron's spikes are present in exactly one cluster, but is insensitive to the ordering of the clusters. It is intended to be a measure which reflects what a user might wish to do with the classification: that is, concentrate on those clusters which are maximally different for different spiking target neurons. Although M is sensitive to non-classification (i.e., assignment to cluster 0) and to misclassifications, we will normally quote the number of non-classifications and misclassifications as well. The top half of Table 3 shows the figure of merit for the clusters in Table 2. The bottom half of Table 2 shows spike sorting with both correlated and uncorrelated noise: note that the interference level for the two different types of interference was set to be the same, so that there is actually more noise in this case, even although the peak-level based SNR recorded does not show this.

Table 3
Spike sorting with correlated and both uncorrelated and correlated interference, showing figure of merit

Data	SNR	P	SPC			KK			*M*	*I*
			FoM	MC	UC	FoM	MC	UC		
UC	4.6	PCA	0.92	0	80	0.93	0	69	5	7
UC	4.6	WAV	0.67	0	358	0.11	477	0	5	7
UC	3.2	PCA	0.90	5	85	0.91	4	75	13	1
UC	3.2	WAV	0.74	0	250	0.86	0	127	13	1
UC	2.3	PCA	0.65	1	125	0.67	1	107	232	1
UC	2.3	WAV	0.67	1	107	0.62	1	149	232	1
UC	1.6	PCA	0.21	51	8	0.21	51	10	737	9
UC	1.6	WAV	0.21	46	21	0.24	1	69	737	9
C,UC	4.6	PCA	0.91	2	87	0.91	3	79	10	46
C,UC	4.6	WAV	0.70	1	294	0.68	0	353	10	46
C,UC	3.2	PCA	0.83	41	56	0.86	8	94	39	38
C,UC	3.2	WAV	0.86	1	111	0.83	0	138	39	38
C,UC	2.3	PCA	0.52	91	20	0.56	5	163	284	27
C,UC	2.3	WAV	0.58	5	136	0.43	3	317	284	27
C,UC	1.6	PCA	0.19	59	7	0.19	59	3	741	35
C,UC	1.6	WAV	0.19	51	16	0.21	12	83	741	35

Table rows show the figure of merit, the number misclassified, and the number not classified. P is pre-processing type, FoM is figure of merit, MC is misclassified (i.e., target 1 spikes classed as target 2 or vice versa), UC is unclassified spikes (either in the catch-all group, or in some other cluster apart from the one selected in the figure of merit), *M* is spikes missed, and *I* is spikes inserted.

In general, the wavelet based pre-processing followed by SPC seems to result in larger numbers of unclassified spikes. This tails off at poor SNR, but by then very large numbers of spikes are simply missed. From the experiment reported here, and from others, we find that at better signal to noise ratios (better than 3:1), wavelet pre-processing tends to leave more spikes unclassified. At poorer SNRs, the results from SPC with either PCA or wavelet pre-processing are very similar, and PCA based techniques seem to result in more misclassification, but fewer unclassified spikes. We also note that PCA followed by KlustaKwik sometimes fails, as is the case in Table 3 line two. There, much better results were obtained when a maximum number of six clusters was allowed (figure of merit = 0.67, with no misclassification and 361 unclassified): in other experiments we find that such failures occur about 10% of the time. Using a maximum of six clusters for the rest of the table results in general in rather poorer figures of merit, and rather more unclassified spikes. In general, permitting a larger maximal number of clusters does overcome this problem, but at the cost of a poorer figure of merit (which arises because a what should be a single cluster is split across a number of clusters).

Comparing these results with those of Quiroga et al. (2004), we find that the difference between the different techniques is less marked that their Table 2 suggests. As noted above, we computed noise differently: for the spike rates used, the equivalent noise levels, using target peak and interference standard deviations are 0.045, 0.64, 0.09 and 0.13 for the uncorrelated noise only, and 0.05, 0.07, 0.01 and 0.14 for correlated plus uncorrelated interference. In particular, we find that SPC applied to the PCA data performs as well as SPC applied to wavelet data. Part of the reason for this may be that we are using SPC in an automatic way, accepting the initial classification as noted in Section 4.

4.2. Varying the difference between the spike shapes

In this test, two spike shapes were generated, differing only in their transfer characteristics, and then sets of data in which one of the spike shapes was made more and more like the other were generated. The aim was to assess how well the different spike sorting techniques could differentiate between similar spike shapes. In performing this experiment, care had to be taken that the spike shapes remained appropriate: for example, the spike shapes should not have two positive peaks with a period of zero value in the middle. In addition, we arranged that the peak-peak size of all the spike shapes used was the same so that spike detection would be equally likely for all spike shapes. The initial spike shapes chosen (see Fig. 4) were relatively similar. The other datasets (datasets 2–10, in Table 4) were generated by making the T1 spike shape more and more like the T2 spike shape in nine steps using simple linear in-terpolation. In dataset 10 the T1 and T2 spike shapes are identical. Even for dataset 1, we found that neither supraparamagnetic clustering nor KlustaKwik applied to PCA data differentiated between them. We therefore report only the wavelet-based techniques: see Table 4. Again, KlustaKwik uses only the first five wavelet coefficients. A fixed level of interference was used: the signal:interference ratio was 2.5:1:1 (Signal:correlated interference:uncorrelated inter-
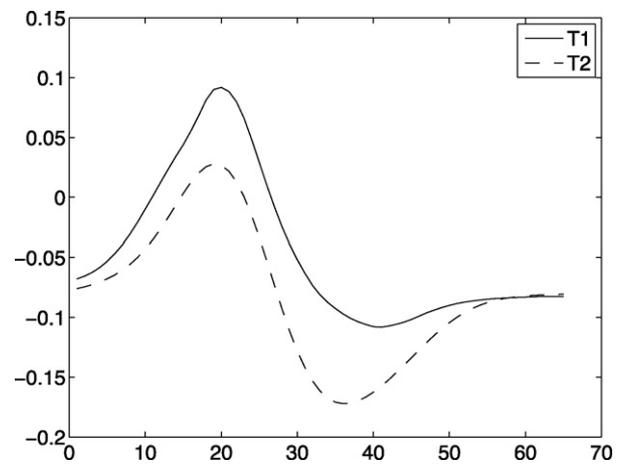

Fig. 4. Shapes of the two similar target neuron spikes (T1 and T2) as received at the putative electrode. *X* axis is in samples at 24 Ksamples/s.

ference), equivalent to a noise level of 0.1 in Quiroga et al. (2004).

In this case, it is clear that the wavelet based pre-processing is critically important. However, the KlustaKwik algorithm outperforms supraparamagnetic clustering: it maintains a high figure of merit with a gradually increasing number of unclassified spikes as the spike shapes become more and more similar. SPC has more unclassified spikes from the start, and fails entirely after dataset 4. KlustaKwik misclassifies hardly any spikes, right up to dataset 9, whereas SPC makes a larger number of misclassifications even in dataset 3. The performance of the KlustaKwik algorithm is surprisingly good, since by dataset 9 the spike shapes look virtually identical. In other similar experiments, similar results were obtained: if the original spikes are a little more different, then PCA does separate them, but the figure of merit is below that produced by wavelet based pre-processing. In general SPC misclassifies more, and generally has a larger number of unclassified spikes as well, although the difference is not always as marked as in Table 4. However, our experience is that KwikKlusters manages to separate the different spikes even when they are very similar, and always outperforms SPC in this respect.

Table 4
The 10 datasets have spike type 1 linearly morphed from the original to spike type 2 shape: in dataset 10 they are identical

Dataset	SPC			KK			*M*	*I*
	FoM	MC	UC	FoM	MC	UC		
1	0.58	1	401	0.79	3	174	61	72
2	0.64	1	354	0.62	0	357	49	53
3	0.55	16	410	0.70	0	275	64	62
4	0.47	14	500	0.72	1	251	57	55
5	0.20	43	822	0.68	0	297	55	47
6	0.05	454	88	0.69	1	282	63	34
7	0.05	444	109	0.71	0	259	58	28
8	0.036	445	122	0.69	0	272	70	45
9	0.06	445	93	0.69	0	284	64	48
10	0.08	447	63	0.056	414	154	66	34

Columns are the same as for Table 3.

5. Conclusions and further work

We have presented a biophysical model for the transfer of electrical signals from neural spikes to an extracellular electrode. In this model, we have considered the electrode to be a voltage sensor: that is a very high input impedance device. From the analysis in Section 2, we have produced a piece of software in MATLAB which can generate realistic signals and interference taking into account both the transfer characteristics between neuron and electrode, and the spatial configuration of the neuron and the electrode. We have shown that this software can generate synthetic signals which can be of use in assessing the effectiveness of algorithms for spike detection and sorting: the many parameters allow for a very considerable range of configurations. The software may be used directly to generate test signals, or as a mechanism for generating realistic non-Gaussian background noise. Further, because both the signals and the noise are synthesized, precise control can be maintained over both: this is not possible using techniques which generate ground truth data by taking recordings which contain spikes, and mixing them and adding noise. This capability enables the types of comparisons described in Section 4.

To illustrate the usefulness of this software, we have tested two different spike-sorting techniques with two different forms of preprocessed synthetic signal data. From these experiments (which were carried out with fixed parameter sets in the spike sorting software, as would be the case for automatic spike sorting), the results seem to depend partly at least on signal: noise ratio. At high SNR, PCA based pre-processing seems to work better, and SPC and KwikKlusters both perform at the same level. At lower SNR, both wavelet and PCA based pre-processing seem to perform at about the same level. SPC spike sorting tends to fail to assign more spikes to either of the two targets, and KwikKlusters tends misclassify more spikes. Where the spike shapes are very similar, the wavelet based pre-processing works much better than PCA based pre-processing. KwikKlusters seems to be more capable of detecting differences where these differences are very small. There are parameter sensitivities as well: KwikKlusters tends not to work well at all when applied to all 10 wavelet coefficients. As we noted in Section 4.1, varying certain parameters in KwikKlusters changes the performance. Further, if the user is allowed to interact with the system while generating clusters (as is the case with Wave_clus), better results can be obtained, particularly when the user has some pre-knowledge of the likely spike shapes, and of the number of different neurons being recorded. However, this is beyond the scope of this paper, since we are interested in automatic classification of spikes.

The software could be extended to generate more than one electrode signal. Where the electrodes are so far apart that they are independent (which is normally the case in current MEAs), no extension is necessary (the software can simply be run more than once). However, if, for example, the electrodes were closely spaced tetrodes, we could use their precise positioning, and determine the precise parameters for the neurons for each electrode, and thus produce a set of synthetic spike trains, one for each electrode. Another possible extension would be to allow the modelling of bursting neurons: given the intracellular characteristics, we can produce the summed weighted delayed signal.

Acknowledgements

References

Atiya A. Recognition of multiunit neural signals. IEEE Trans Biomed Eng 1992;39:723–9.

Bower J, Beeman D. The book of GENESIS. 2nd ed. Springer-Verlag; 1998.

Celeux G, Govaert G. A classification EM algorithm for clustering and two stochastic versions. Comput Statistics Data Anal 1992;14(3):315–22.

Chandra R, Optican L. Detection, classification and superposition resolution of action potentials in multiunit single channel recordings by an on-line real-time neural network. IEEE Trans Biomed Eng 1997;44:403–12.

Harris, K, 2003. Klustakwik spike sorting, at http://klustakwik.sourceforge.net/.

Harris K, Henze D, Csicsvari J, Hirase H, Buszaki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J Neurophysiol 2000;84:401–4.

Holt G, Koch C. Electrical interactions via the extracellular potential near cell bodies. J Comput Neurosci 1999;6:169–84.

Koch C. Biophysics of computation. Oxford; 1999.

Lewicki M. A review of methods for spike sorting: the detection and classification of neural potentials. Network: Comput Neural Syst 1998;9:R53–78.

Mamlouk A, Sharp H, Menne K, Hoffmann U, Martinetz T. Unsupervised spike sorting with ICA and its evaluation using GENESIS algorithms. Neurocomputing 2005;65-66:275–82.

Menne K, Folkers A, Malina T, Maex R, Hoffmann U. Test of spike sorting algorithms on the basis of simulated network data. Neurocomputing 2002;44–46:1119–26.

Moffitt M, McIntyre C. Model based analysis of cortical recording with silicon microelectrodes. Clin Neurophysiol 2005;116:2240–50.

Nakatani H, Watanabe T, Hoshimiya N. Detection of nerve action potentials under low signal-to-noise ratio condition. IEEE Trans Biomed Eng 2001;48:845–9.

Naundorf B, Wolf F, Volgushev M. Unique features of action potential initiation in cortical neurons. Nature 2006;440(7087):1060–3.

Quiroga R, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 2004;16:1661–87.

Sandison M, Curtis A, Wilkinson C. Effective extra-cellular recording from vertebrate neurons in culture using a new type of micro-electrode array. J Neurosci Methods 2002;114:63–71.

Smith L, Mtetwa N, 2006. Manual for the noisy spike generator matlab software, at http://www.cs.stir.ac.uk/~lss/noisyspikes/.

Standen NB, Gray PTA, Whitaker MJ. Microelectrode techniques: the plymouth workshop handbook. Cambridge: The Company of Biologists; 1987.

Struijk J J, Yoshida K, 2004. Volume conduction, at www.smi.hst.aau.dk/npig/seminar/2004/0312JJSVolumeConduction.pdf.

Touretzky D, Albert M, Daw N, Ladsariya A, 2004. HHsim: Graphical hodgkin-huxley simulator, at http://www.cs.cmu.edu/~dst/HHsim/.

Wood F, Black M, Vargas-Irwin C, Fellows M, Donoghue J. On the variability of manual spike sorting. IEEE Trans Biomed Eng 2004;51: 912–8.

Zhang P-M, Wu J-Y, Zhou Y, Liang P-L, Yuan J-Q. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. J Neurosci Methods 2004;135:55–65.

Zhang Z-w. Maturation of layer V pyramidal neurons in the rat prefrontal cortex: Intrinsic properties and synaptic function. J Neurophysiol 2004;91(3):1171–82.