ELSEVIER

Journal of Neuroscience Methods 115 (2002) 29–43

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Signal-to-noise ratio improvement in multiple electrode recording

P.G. Musial [a,b], S.N. Baker [c], G.L. Gerstein [a,*], E.A. King [a], J.G. Keating [a]

[a] *Department of Neuroscience, University of Pennsylvania, A306 Richards Building, 3700 Hamilton Walk, Philadelphia, PA 19104-6085, USA*
[b] *Department of Neurophysiology, Nencki Institute of Experimental Biology, Warsaw, Poland*
[c] *Department of Anatomy, University of Cambridge, Cambridge CB2 3DY, UK*

Received 6 July 2001; received in revised form 26 November 2001; accepted 26 November 2001

Abstract

Recordings of spike trains made with microwires or silicon electrodes include more noise from various sources that contaminate the observed spike shapes compared with recordings using sharp microelectrodes. This is a particularly serious problem if spike shape sorting is required to separate the several trains that might be observed on a particular electrode. However, if recordings are made with an array of such electrodes, there are several mathematical methods to improve the effective signal (spikes) to noise ratio, thus considerably reducing inaccuracy in spike detection and shape sorting. We compare the theoretical basis of three such methods and evaluate their performance with simulated and real data.

Keywords: Signal/noise; Multi-electrode; Multi-neuron; Extracellular spikes; Waveform sorting

1. Introduction

Increasing numbers of laboratories are now making recordings of multiple neurons simultaneously, frequently in awake behaving animals. The appropriate technology has evolved from single sharp electrodes that allow clean observation of single unit activity to arrays of blunter electrodes, either small diameter insulated wires or silicon arrays. Each such electrode generally records action potentials from more than one neuron, thus forcing the use of spike shape sorting to separate the several activities for further analysis. Many computational techniques are available for shape sorting, and have recently been reviewed by Lewicki (1998). All such methods involve mapping some measure of individual spike shape to a point in multi-dimensional property space. The clustering of points in this space is then used to identify and differentiate among the several different spike trains present in the recording. Sometimes closely spaced arrays of electrodes are used—stereotrodes, triodes, tetrodes (McNaughton et al., 1983; Gray et al., 1995)—so that the action potential from a given neuron is recorded by more than one electrode. The signal seen by each such electrode is different because of the different relative neuron/electrode geometry. When the measured values to be used in the property space are obtained from different electrodes, waveform sorting can be more effective.

Whatever measures are used for the clustering process, it is ultimately necessary to define boundaries between the different clusters obtained. It is at this stage that the effects of signal to noise ratio become critically apparent. If the signal to noise ratio is favorable, the clusters will be well isolated with few outliers, and the separation process will be free of errors. If on the other hand the spike waveform does not rise much above the background noise, and particularly if this background noise has similar spectral content to the desired spikes, the clusters in the property space will overlap. It is then impossible to define clean boundaries, and hence there are inevitably errors of inclusion or exclusion in the assignment of an individual action potential to a particular neuron. Such poor sorting can render subsequent analysis at best inaccurate, and at worst invalid. Errors can be especially serious for cross-correlation and related analyses (Gerstein, 2000).

Obviously the best way to avoid problems in sorting is to have a good signal to noise ratio. Little can be done to increase the size of recorded spikes with a given

* Corresponding author. Tel.: +1-215-898-8752; fax: +1-215-573-5851.
E-mail address: george@mulab.physiol.upenn.edu (G.L. Gerstein).

0165-0270/02/$ - see front matter
PII: S0165-0270(01)00516-7

type and size of electrode; the relevant and usually fixed factors involve the impedance and shape of the electrode tip, and its placement relative to the nearby active neuron. The noise component of the recording however offers some possibilities for improvement, and that will be the topic of this paper.

We assume that the low frequency local field potentials have been filtered out of the recordings, so that our pass band is $\approx$ 300 Hz–8 kHz. By definition, noise in this residual recording is everything except the (relatively) large spikes. Such noise can come from a number of sources: (1) Johnson noise from individual electrode impedances, (2) Johnson or other noise from the reference electrode, (3) high frequency components of field potentials from distant active populations of neurons, (4) electromyogram (EMG) from muscles in the scalp, jaws, neck, and sometimes body, (5) electrical artifacts generated in the wiring harness by abrupt movements, and (6) other types of electrical artifact as the animal moves and touches various portions of the apparatus. Fortunately, the spatial gradient across an electrode array of the potentials from sources (2) to (6) differs from that of action potentials from a particular neuron, and makes possible the several approaches to be examined in this paper.

To make these differences clearer, consider a single electrode that is successively moved in 100 μm steps. At each location, the magnitude of the Johnson noise will be the same. The large spikes recorded will differ at each site. However, potentials from noise sources (2) to (5) will change little from site to site. Thus if a closely spaced electrode array is used, these potentials from distant sources will appear approximately the same on each electrode. This is amply illustrated in Fig. 3 of Bierer and Anderson (1999): for 50 μm electrode spacing, a particular spike train is essentially visible only on a single channel while the non-spike activity is almost identical across four channels and 150 μm. The same can be seen in our Fig. 4. It is the shared nature of the noise across multiple electrodes that will be used in this paper to produce an improvement in the signal to noise ratio. These ideas are only applicable to recordings made with an array of similar electrodes, and with inter-electrode spacing no more than several hundred micrometers.

One approach to this form of noise reduction is based on signal processing used in radar (Applebaum, 1976) and was tested on spike data by Bierer and Anderson (1999). Here we present two other approaches appropriate for arrays of both single electrodes and tetrodes. Using simulated data we carry out a parametric analysis of the performance of each method, and show that the three methods produce almost identical results. The theoretical basis for this is explored in Appendix A. Finally we demonstrate the improvement which can be obtained on real data.

2. Methods

2.1. Simulation of test data

In order to allow parametric study of the noise reduction methods explored here it is necessary to have independent knowledge of the data properties. This is impossible with real data, so that the main part of this paper uses simulated data with known parameters. We choose to mimic many of the parameters of our experimental recording system. Twelve 'recording' channels are simulated mimicking an electrode array with 100 μm spacing, sampled at 25 kHz and with a duration of 1 s. In total, 15 independent noise voltages are simulated as Gaussian random noise bandwidth limited to 300 Hz–5 kHz. The amplitudes of each are specified in advance, and can vary from 0 to 1. A different independent noise waveform is added to each one of the 12 recording channels, mimicking Johnson noise for that electrode. Another noise waveform is added equally to all recording channels, mimicking effects from an indifferent or reference electrode. Finally, two other noise waveforms are added to all channels, but with an amplitude gradient that varies linearly across the array from 1.0 at one side to 0.45 at the other. The side with maximum noise amplitude is different for each of these two common noise sources; they mimic high frequency components of field potentials generated by two distant neuronal populations. The choice of gradient parameter is reasonable for mimicking a 1.2 mm wide array, but in fact turns out to be irrelevant for the results; we have even tried random amplitude of these shared noise distributions across the array with identical final performance. For simplicity, we did not include simulation of the short noise events originating from scratching/chewing EMG or electrical artifact; such events are however apparent in the real data analyzed in Fig. 4.

Finally, we generate a spike train of specified rate and with a gamma interval distribution of specified order (order 1 would be a Poisson train, higher orders mimic interval distributions with refractory time; Baker and Gerstein, 2000). A single experimentally recorded spike waveform is then added to the 12 channels at each time location specified by the generated spike train. For this process we treat the 12 channels as if they were sequential rather than simultaneous, using sequential segments of the generated spike train. This is justified since we are using a steady state spike train without rate or pattern modulations by a stimulus event.

Thus the final output of the simulator is a 1 s 'recording' of 12 channels, with independent parameters setting a specified set of independent and shared noises, and with an independently timed spike train on each channel (Fig. 2A). (However, the waveform of the spike is the same on each channel, unlike real data; this again

is justified for our particular usage in this paper of the simulated data). Examination of individual spike waveforms in the data at this stage shows the clean original waveform badly distorted by the combined noise voltages; the distortions are of course different for each occurrence of the spike on any channel (Fig. 2C).

2.2. Algorithms for improvement of signal:noise ratio

2.2.1. General notation

Denote the data to be processed as matrix $\mathbf{D}(t,c)$, where t indexes the successive data samples in time (rows) and c indexes channels (columns). Let $\max(c) = K$ and $\max(t) = T$. For our particular simulated data sets $K = 12$ and $T = 25000$, representing 12 channels recorded for 1 s at 25000 samples per second.

2.2.2. The PCA cleaner

The general idea here is to use principal components analysis to predict as much of the noise contribution to each electrode as possible. For each electrode in turn, we find the two or three principal components that best represent the data from all the *other* electrodes. This representation mostly reflects the noise component of the data, and is only slightly dependent on the spike trains. An iterative procedure (Stage 2, described below) can subsequently be used to reduce the influence of the spike trains on the calculation. We project the data from the electrode being processed onto these principal components, finding thereby the coefficient that should multiply each principal component when we use them to predict the noise on this channel. This prediction is then subtracted from the data on the electrode being processed, yielding a first order estimate of the cleaned signal. The entire process is carried out successively for each electrode, i.e. 12 times for the 12 channels of data provided by the simulator.

In more detail, in order to clean the data from electrode i, we form a reduced data matrix $\mathbf{Dnoti} = \mathbf{D}(t,c_{\text{noti}})$ which contains the data from all channels *except* for i, the channel being processed.

The covariance $\mathbf{Cnoti} = \text{cov}(\mathbf{Dnoti})$ has dimensions $(K-1)$ where $K = \max(c)$. We obtain the eigenvectors $\mathbf{V}_n(\text{noti})$ corresponding to the three largest eigenvalues using Matlab's 'eig' function. These three $(n = 1–3)$ eigenvectors also have dimension $(K-1)$. Each represents a set of weighting factors with which to sum the $(K-1)$ data channels (columns of **Dnoti**) in computing the corresponding principal component.

Thus the principal components are $\mathbf{pc}_n(\text{noti}) = \mathbf{Dnoti}\mathbf{V}_n(\text{noti})$, have a dimension of T where $T = \max(t)$ and give an optimal representation of the data from all but the ith channel.

We now project the data of the channel being processed, $\mathbf{D}(t,c_i)$, onto each of these principal components, obtaining a normalized coefficient $b_n(i)$. Letting the superscript T denote transpose of a vector or matrix, we have:

$$b_n(i) = \mathbf{D}(t,c_i)^{\mathrm{T}}\mathbf{pc}_n(\text{noti})/(\mathbf{pc}_n(\text{noti})^{\mathrm{T}}\mathbf{pc}_n(\text{noti})) \quad (1)$$

These coefficients are the amplitude fraction of the corresponding principal component that should be used for representing the noise on the ith channel.

Finally, to obtain $\mathbf{D}_{\text{clean}}(t,c_i)$, the estimate of the ith data channel without noise, we take the raw data values and subtract the contributions from each principal component specified by the $b_n(i)$.

$$\mathbf{D}_{\text{clean}}(t,c_i) = \mathbf{D}(t,c_i) - b_1(i)\mathbf{pc}_1(\text{noti}) - b_2(i)\mathbf{pc}_2(\text{noti}) - b_3(i)\mathbf{pc}_3(\text{noti}) \quad (2)$$

This entire process is carried out for each electrode in turn, producing the first stage cleaned data. Calculations were done in Matlab (MathWorks Inc.).

2.2.3. The regression cleaner

The general idea here is to use multiple linear regression to predict as much of the noise contribution to each electrode as possible. We find the weighted sum of the data on all other electrodes that best represents the data on the electrode being processed, and carry out this calculation on each electrode in turn. This representation mostly reflects the noise components of the data, and is only slightly dependent on the spike trains. An iterative procedure (Stage 2, described next) can subsequently be used to reduce the influence of the spike trains on the calculation. In a common use of regression analysis the regressors, i.e. the functions for which we seek optimal weights for the representation are a constant and a linear slope, and we seek two coefficients. In our case the regressors are the $K-1$ data vectors on the *other* electrodes, and we seek $K-1$ coefficients. The criterion for calculating these coefficients is a least squares fit between the weighted sum of regressors and the data vector on the electrode being processed. Once the coefficients are determined, we have an optimal representation of the noise, and can subtract it from the data vector on the electrode being processed. The result of this subtraction is called the residual, and is the first order cleaned signal. The entire process is carried out successively for each electrode.

In more detail, a weighted sum of the $K-1$ regressors $\mathbf{D}(t,c_{\text{noti}})$ will be used to approximate $\mathbf{D}(t,c_i)$:

$$\mathbf{D}(t,c_i) = \sum_{\substack{k=1 \\ k \neq i}}^{K} g_k \mathbf{D}(t,c_k) + \varepsilon_i(t) \quad (3)$$

where $\varepsilon_i(t)$ is a normal random variable with mean 0 and some variance σ^2.

The expectation of $\mathbf{D}(t,c_i)$ is then

$$E\{\mathbf{D}(t,c_i)\} = \sum_{\substack{k=1 \\ k\neq i}}^{K} g_k \mathbf{D}(t,c_k) \tag{4}$$

Thfle error in the approximation is $\varepsilon_i(t)$:

$$\varepsilon_i(t) = \mathbf{D}(t,c_i) - \sum_{\substack{k=1 \\ k\neq i}}^{K} g_k \mathbf{D}(t,c_k) \tag{5}$$

The g_k coefficients will be determined by minimizing the error in the least squares sense. Thus we seek

$$\min[\varepsilon_i^2(t)] = \min\left\{\mathbf{D}(t,c_i) - \sum_{\substack{k=1 \\ k\neq i}}^{K} g_k \mathbf{D}(t,c_k)\right\}^2 \tag{6}$$

The $K-1$ coefficients are calculated by taking and setting equal to zero the $K-1$ partial derivatives of Eq. (6) with respect to each of the g_k, and solving the resulting equations (Winer, 1971; Blum and Rosenblatt, 1972).

The optimal representation of the noise component of the data on electrode i is then given by Eq. (4), and the cleaned data on electrode i is the residual:

$$\mathbf{D}_{\text{clean}}(t,c_i) = \mathbf{D}(t,c_i) - \sum_{\substack{k=1 \\ k\neq i}}^{K} g_k \mathbf{D}(t,c_k) \tag{7}$$

The above procedure is carried out successively for each electrode in turn, producing the first order cleaned data. Calculations were done in Matlab (MathWorks Inc.) using the multiple regression function *regress*.

We note that successful application of the regression algorithm may not always be possible. Calculation of the $K-1$ coefficients a_k depends (Blum and Rosenblatt, 1972) on the existence of the inverse matrix:

$$\{\mathbf{D}(t,c_{\text{noti}})\mathbf{D}(t,c_{\text{noti}})^{\text{T}}\}^{-1} \tag{8}$$

where the superscript T indicates transpose. If the regressors (i.e. columns of $\mathbf{D}(t,c_{\text{noti}})$ representing the data on the *other* electrodes) are linearly independent the existence of the inverse matrix is assured. However, in our application the regressors have some degree of shared noise as well as individual noise (and spikes). If the individual noise components are sufficiently small relative to the shared components, linear independence of the regressors is violated, and the inverse matrix may not exist. However, we note that in all instances of real data analysis we have never encountered this problem.

2.2.4. The array cleaner

The general idea here is to find a linear combination of *all* channels that optimizes the ratio of signal to noise, where for our application the signal consists of spike waveforms and the noise is everything else in the recording. This approach is based on a processing method for an array of radar antennas (Applebaum, 1976) and has been implemented for multi-electrode recordings by Bierer and Anderson (1999). Note that the immediate objective here is different from the two algorithms described earlier. In those calculations we derived a representation of the noise components from the *other* channels and subtracted this from the channel being processed. The identity of the K channel recordings was preserved through both stages of the cleaning process. Here, in contrast, the channel identity is generally preserved only in the first order application of the algorithm; as before this allows better detection of whatever spikes may be in the data. The output of a second order application of the algorithm will, depending on parameters, consist of vectors that may still represent channels, or alternatively may represent a weighted sum of transformed data on the original channels. In the latter situation the output vectors are no longer necessarily associated with a particular channel but might be associated with a particular neuron. In spite of these apparently fundamental differences all three algorithms produce essentially the same improvement in the data (see Section 3) and are shown in Appendix A to be closely related.

The required basic linear transformation equation (derived in Applebaum, 1976) is:

$$\mathbf{Z}(t,u) = \mathbf{D}(t,c)\mathbf{C}_{\text{N}}^{-1}\mathbf{w}(c,u) \tag{9}$$

where $\mathbf{Z}(t,u)$ is the output of the algorithm, with u indexing the signal sources (individual neurons in the data for our application), and $\max(u) = U$. $\mathbf{D}(t,c)$ is the raw data which includes signal (spikes) and noise, $\mathbf{C}_{\text{N}}$ is the covariance matrix of *only* the noise in the data, $\mathbf{C}_{\text{N}}^{-1}$ is its inverse, and $\mathbf{w}$ is a weighting matrix described further. The dimensions of $\mathbf{Z}$ are T rows, U columns, $\mathbf{D}$ is T rows (sample time points) and K columns (channels), while $\mathbf{C}_{\text{N}}$, $\mathbf{C}_{\text{N}}^{-1}$ are $K \times K$, and $\mathbf{w}$ is K rows and U columns. Note that Eq. (9) can be interpreted as a projection of the data $\mathbf{D}$ onto a set of vectors $\mathbf{C}_{\text{N}}^{-1}\mathbf{w}(c,u_i)$, $i = 1,\ldots, U$. Each such vector is the optimal projection choice to maximize signal:noise for the ith neuron.

In the first order application of this algorithm we obviously cannot yet parse the raw data into signal and noise. Therefore, even though biased, we compute the covariance of the raw data. Thus initially we have $\mathbf{C}_{\text{N}} = \text{cov}(\mathbf{D}(t,c))$ and can usually obtain its inverse $\mathbf{C}_{\text{N}}^{-1}$.

In the original radar application there was only a single signal, and $\mathbf{w}$ was a column vector representing the distribution of signal amplitude across the K channels. In the present application we potentially have U signals (spikes from U different neurons), and we take $\mathbf{w}$ as a matrix where each of the U columns describes the amplitude distribution across the K channels of the spikes from each recorded neuron. Obviously, we do not have the information to determine such a $\mathbf{w}$ at this stage, and therefore choose it to be unitary, i.e. ones

on the diagonal, zeros elsewhere. This choice corresponds to the assumption that each channel of data contains spikes from only one neuron, i.e. that no two electrodes record the same neuron's spikes and that neuron identity corresponds to channel identity. Both the calculation of $\mathbf{C}_N$ and the choice of $\mathbf{w}$ can be improved in the subsequent second order application of the algorithm as indicated below.

Note that with the stated conditions, the first order $\mathbf{Z}(t,u)$ is not necessarily equivalent to the first order $\mathbf{D}_{clean}(t,c)$ obtained from the PCA and regression algorithms. One difference is that each column of $\mathbf{Z}$ generally has a *different* scaling factor, a problem that does not arise in $\mathbf{D}_{clean}$. It is convenient to normalize each column of $\mathbf{Z}$ so that $\mathbf{Z}(t,c_i)$ is divided by $\mathbf{C}_N^{-1}(i,i)$.

The algorithm expressed in Eq. (9) is somewhat unfamiliar in neuroscience, so that it is appropriate to examine its origin. Here we present a derivation of the array processing algorithm using the method of discriminant analysis. The data to be processed are contained as before in matrix $\mathbf{D}(t,c)$. There are K channels and each has T time points. Now consider a new K-dimensional vector space in which individual vectors will represent the data values at consecutive time points across all K channels, i.e. the rows of $\mathbf{D}$. At each time point we assume there is a linear sum of the signal and noise. For the present purpose we will assume that $U=1$, i.e. that the signal consists of spikes *only from one particular neuron* (but observed with different amplitude across the K channels). The noise comes from shared and individual sources, so that it may be variously correlated across the K channels. If we examine the T noise and signal vectors separately, they will form two clusters in our vector space.

Let $s(t)$ be the time course of the signal (spikes) and let its amplitude be distributed across the K channels with weights described by column vector $\mathbf{w}_i$ ($i=1,\ldots,K$). Thus, *at each time point* t ($t=1,\ldots,T$) the signal across all channels can be described as $\mathbf{S}(t)=s(t)\mathbf{w}^T$; both $\mathbf{S}(t)$ and $\mathbf{w}^T$ are row vectors of dimension K. Analogously, the noise is described *at each time point* t by $\mathbf{N}(t)$, also a K-dimensional row vector. For purposes of illustration, let us take $K=2$, i.e. two channels only. As seen on Fig. 1, all signal points (i.e. $\mathbf{S}(t)$ for each of the T values of t) lie on the direction defined by $\mathbf{w}^T$. To make Fig. 1 clearer, we assume the signal has a DC component larger than the noise variance so that the signal and noise clusters are clearly separated. The noise component of the data (i.e. $\mathbf{N}(t)$ for each of the T values of t) forms an ellipsoid-like cloud whose shape depends on the pattern of noise correlations between channels (Fig. 1, lower panel). In the case of zero correlation between channels, the noise cloud is spherical (Fig. 1, upper panel). However, the noise cluster is always centered at the origin because we assume it has no DC component.

The problem of finding the channel mix that will maximize the signal to noise ratio is equivalent to a search for the one-dimensional projection from our K-space that yields maximal separation of the projected signal and noise clusters, i.e. a discriminant analysis. If the noise is not correlated across channels, the optimal direction for the projection is parallel to the signal vectors (see projection marked by **o** in Fig. 1, upper panel). If the noise is correlated, the optimal projection is a trade off between projections onto the signal direction **s** and the direction **n** that has minimal variance of the noise (see Fig. 1, lower panel).

Let us define the separation (Δ) of signal and noise clusters *after* projection as

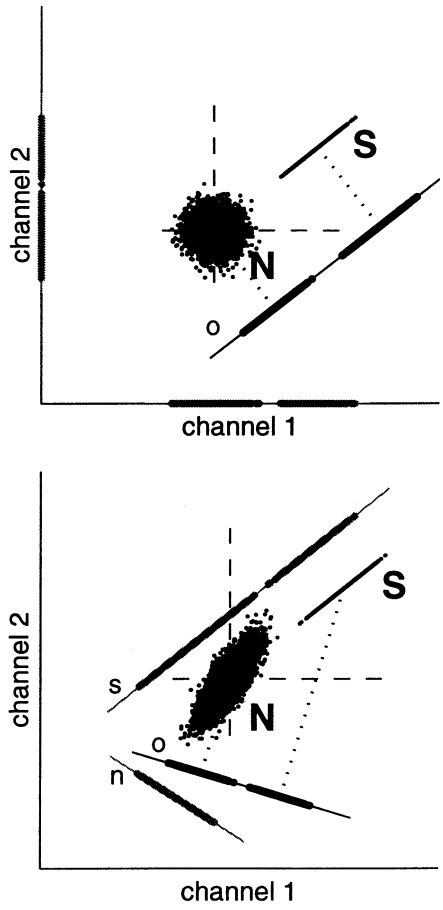

Fig. 1. Illustration of the optimal projection for maximal signal to noise ratio in the array algorithm. This is a two-dimensional caricature of the K-dimensional data space (i.e. simulation of two channel recording, with each axis representing one channel). The signal and noise clusters (**S**, **N**, respectively) are well separated for illustrative purposes by assuming a DC component for the signal. Upper figure: uncorrelated noise. The variance of the noise cluster is the same in all possible projections. Thus the optimal projection (marked by **o**) yielding maximal separation (Δ; see text) is the one leading to maximal signal amplitude, i.e. projection onto a vector parallel to the signal vectors. Lower figure: noise correlated across channels. The optimal projection (**o**) is determined by $\mathbf{C}_N^{-1}\mathbf{S}$ and is a trade-off between the signal direction **s** and the minimal noise variance direction **n**.

$$\Delta = \frac{d^2}{V_S + V_N} \tag{10}$$

where d is the difference between mean values of the projected signal and the projected noise; V_S, V_N are variances of the projected noise and signal, respectively.

Let the direction of the optimal one-dimensional projection be defined by the vector **a**. For any time point the vector $\mathbf{S}(t)$ projected onto **a** is $\mathbf{a}^T\mathbf{S}(t)$ (analogously, for the noise vectors: $\mathbf{a}^T\mathbf{N}(t)$). These are both inner products. Thus, given the signal (**S**) and noise (**N**) clusters the separation Δ can be expressed as a function of projection vector **a**:

$$\Delta = \frac{(\mathbf{a}^T\bar{\mathbf{S}} - \mathbf{a}^T\bar{\mathbf{N}})^2}{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{a}^T\mathbf{S}(t) - \mathbf{a}^T\bar{\mathbf{S}})^2 + \frac{1}{T}\sum_{t=1}^{T}(\mathbf{a}^T\mathbf{N}(t) - \mathbf{a}^T\bar{\mathbf{N}})^2} \tag{11}$$

where $\bar{\mathbf{N}}$, $\bar{\mathbf{S}}$ are the mean values of **S**, **N**, respectively, over the time of analysis.

But

$$\frac{1}{T}\sum_{t=1}^{T}(\mathbf{a}^T\mathbf{S}(t) - \mathbf{a}^T\bar{\mathbf{S}})^2 = \mathbf{a}^T\left(\frac{1}{T}\sum_{t=1}^{T}(\mathbf{S}(t) - \bar{\mathbf{S}})(\mathbf{S}(t) - \bar{\mathbf{S}})^T\right)\mathbf{a} = \mathbf{a}^T\mathbf{C}_S\mathbf{a} \tag{12}$$

where $\mathbf{C}_S$ is covariance matrix of the signal. After applying the same operation to the noise cluster **N** and denoting the noise covariance matrix as $\mathbf{C}_N$ we can rewrite Eq. (11) in the following way:

$$\Delta = \frac{(\mathbf{a}^T\bar{\mathbf{S}} - \mathbf{a}^T\bar{\mathbf{N}})^2}{\mathbf{a}^T(\mathbf{C}_S + \mathbf{C}_N)\mathbf{a}} \tag{13}$$

Projection of the clusters on vector **a** is intended to maximize separation. Vector **a** can be found by setting to zero the partial derivative of Eq. (13) with respect to **a**. This gives (Rao, 1973, p. 60):

$$\mathbf{a} = (\mathbf{C}_S + \mathbf{C}_N)^{-1}(\bar{\mathbf{S}} - \bar{\mathbf{N}}) \tag{14}$$

Since we assume that there is no DC component of noise on any channel we can neglect the $\bar{\mathbf{N}}$ term in Eq. (13). Also, since $\mathbf{S}(t) = s(t)\mathbf{w}$, we get $\bar{\mathbf{S}} = \bar{s}\mathbf{w}$, where $\bar{s}$ is the mean value of $s(t)$ over time:

$$\bar{s} = \frac{1}{T}\sum_{t=1}^{T}s(t)$$

We are not interested in the norm of **a**, but only in its direction, so we can write

$$\mathbf{a} = (\mathbf{C}_S + \mathbf{C}_N)^{-1}\mathbf{w} \tag{15}$$

This can be further simplified. Multiply Eq. (15) by $(\mathbf{C}_S + \mathbf{C}_N)$ from the left giving:

$$\mathbf{C}_S\mathbf{a} + \mathbf{C}_N\mathbf{a} = \mathbf{w} \tag{16}$$

But

$$\mathbf{C}_S = \frac{1}{T}\sum_{t=1}^{T}(\mathbf{S}(t) - \bar{\mathbf{S}})(\mathbf{S}(t) - \bar{\mathbf{S}})^T = \frac{1}{T}\sum_{t=1}^{T}(s(t) - \bar{s})\mathbf{w}(s(t) - \bar{s})\mathbf{w}^T = v(s)\mathbf{w}\mathbf{w}^T \tag{17}$$

where $v(s)$ is the variance of $s(t)$:

$$v(s) = \frac{1}{T}\sum_{t=1}^{T}(s(t) - \bar{s})^2$$

Multiplying Eq. (17) by **a** from the right and rearranging terms:

$$\mathbf{C}_S\mathbf{a} = v(s)\mathbf{w}(\mathbf{w}^T\mathbf{a}) = v(s)(\mathbf{w}^T\mathbf{a})\mathbf{w} = \beta\mathbf{w} \tag{18}$$

The rearrangement in Eq. (18) is allowed since $\mathbf{w}^T\mathbf{a}$ is an inner product of vectors **w** and **a**, and hence a scalar. The new constant β is the product of that scalar with $v(s)$. Eq. (18) states that $\mathbf{C}_S\mathbf{a}$ is parallel to **w**. This is not surprising because, as seen in Fig. 1, all points in the signal cluster are aligned with vector **w**. This means that $\mathbf{C}_S$, the covariance matrix of the signal, has only one eigenvector which is parallel to **w**. Now, using Eq. (18) we can transform Eq. (16) into $\mathbf{C}_N\mathbf{a} = \alpha\mathbf{w}$, where $\alpha = (1 - \beta)$, a scalar. Thus the direction on which to project the data for optimal signal:noise separation is defined by:

$$\mathbf{a} = \mathbf{C}_N^{-1}\mathbf{w} \tag{19}$$

which is the same as shown in Eq. (9).

2.2.5. Iteration of the algorithms

After the first application of any of the earlier three cleaning algorithms there is a considerable reduction of all common mode noise on each channel, leaving largely the individual Johnson like noise component, and of course all the large spikes. However, for each channel, the noise representation produced by the PCA and regression cleaning algorithms was based on original data from the $K-1$ *other* channels, including their spikes. All these spikes will appear as small events in the noise representation with amplitude of about $1/(K-1)$ of the original spike amplitudes. When the subtraction of noise representation from original data on the channel being processed is carried out, the residual, besides showing a reduction of noise, will in addition show these small spike waveforms (inverted). A similar situation occurs with the array algorithm. The initial calculation of the 'noise' covariance is necessarily made using the raw data, and is therefore biased by whatever spikes are present. It is of course desirable to eliminate the artifactual 'mixing' across channels. The following procedure is used.

In the raw data with its large amount of noise, detection of spike events would generally not be reliable. However, after the first stage of cleaning by any of the algorithms there is considerable noise reduction,

and we can readily and accurately detect the spike occurrences on each channel. Such spike detection can be done either by an amplitude threshold or a time derivative (slope) criterion. We now go back to the original data and remove a small section around each of the now well defined spike occurrences. We could substitute zeros (or the mean) for each of these edited sections, but there is a better choice. Consider the matrix formed by the original raw data minus the first order cleaned:

$$\mathbf{F}(t,c) = \mathbf{D}(t,c) - \mathbf{D}_{\text{clean}}(t,c) \tag{20}$$

In this difference matrix, spikes are removed, while noise components remain. We use this matrix as a source for substitution into the edited chunks of the original data. By this procedure we have edited the original data so that spikes are removed and substituted by a close approximation of the residual noise at each time point where there had been a spike. Call this edited data $\mathbf{D}_{\text{mod}}(t,c)$.

Now we repeat the entire cleaning algorithm procedure. For the PCA and regression algorithms we process each channel with the noise representation calculated from the *other* $K-1$ channels, but always using as input the appropriate $\mathbf{D}_{\text{mod}}(t,c_{\text{noti}})$ from the composite spike free modification of the original data. After this second stage cleaning, common mode noise is much reduced and spike shapes therefore much improved, all as in the first stage cleaning. But now there is no artifactual channel mixing with the small inverted spike waveforms from all other channels. For the array algorithm the 'noise' covariance is now calculated from the spike free $\mathbf{D}_{\text{mod}}(t,c)$, which eliminates the mixing. For K channels this cross-contamination has relative amplitudes of $1/(K-1)$; it is more noticeable therefore when the number of electrodes is smaller than in the present 12 channel example, and the second stage cleaning procedure as described earlier is more necessary. Thus the first stage of cleaning is used only for accurate spike detection and substitution. The spike free data are then used in a second stage of cleaning.

2.2.6. Tetrode recordings, 'unitary events' and near-coincidences

When dealing with recordings made with multiple stereotrodes or tetrodes, the same spikes are observed at *exactly* the same time on more than one channel within a tetrode because spacing is under 50 μm. This multiplicity gives additional possibilities for spike sorting, but at first glance causes problems with the cleaning algorithms as described earlier. The PCA and regression algorithms, besides reducing noise on the processed channel, would then have an amplitude reducing effect on its spikes since they are shared across some channels. In stage one of the calculation, this reduction could reach a relative factor of $3/(K-1)$ for a recording made with $K/4$ tetrodes (the total number of channels is K). This would likely make the identification of spikes more difficult. Even if spike identification is successful this amplitude reduction would contaminate the composite spike free modification of the original data needed for Stage 2, and consequently distort the final spike waveform. The simple solution for PCA and regression algorithms, although it reduces the effective K, is to exclude *all* channels of the stereotrode or tetrode being processed from the computations both in the initial and second stage cleaning procedure (not shown in this paper). For the array algorithm it is necessary to make an appropriate choice of the weighting matrix $\mathbf{w}$ for the second stage procedure, as described by Bierer and Anderson (1999).

A related problem occurs if the objective is to study 'unitary spike events' (Grün et al., 1999), i.e. excess near coincidences among different neurons on different channels. If the time coincidence is *exact*, even though different waveforms are involved, the cleaning algorithms will indeed reduce amplitude and distort waveforms (see next paragraph). However, usually a tolerance of 1–5 ms is allowed to search for and define 'unitary events'. If the several spikes comprising such an event have as little as 300 μs of time offset or jitter among them, the cleaning algorithms have almost no effect on final spike amplitude or waveform (not shown). The computations to remove shared signal are on a time point by time point basis, and apparently with even a minimal offset the interference between near simultaneous waveforms on different channels is low. Thus the use of the cleaning algorithms should have little effect on detection of 'unitary spike events'.

The interference applies to coincidences between spike trains (within 300 μs) and will cause the algorithms to reduce amplitude and distort waveforms on such events. For a pair coincidence, the distortion of a given spike waveform will essentially be by subtraction of $1/(K-1)$ of the near coincident spike waveform on some other channel. In the case of equal amplitude waveforms on the two channels, and with 12 channels as in our data, this is less than a 10% distortion, and in detail depends on the exact time offset between the near coincident spikes. In our simulations the individual channel noise (which is not affected by the algorithms) is at least the same fraction of spike amplitude, and is frequently larger in real data. Thus, the waveform distortions contribute under half of the residual spike waveform variability, and in practice can be neglected. The rate of near coincidences of course depends on the individual spike rates, but since the interference effects are small, the rate dependence can also be neglected. None of these effects have any noticeable effect on individual spike detection.

2.3. Performance assessment

There are a number of ways to demonstrate the noise reduction performance of these several cleaning algorithms. We could, before and after the cleaning process, show a superposition of the spike waveform over all channels (a single spike waveform is used for all channels in the simulated data). The noise-induced distortions of the waveform will be smaller after cleaning, as can easily be shown by a variance calculation. Another method appropriate for simulated data would be to project each cleaned waveform onto the known elemental waveform (an inner product of two vectors). This is a one-dimensional goodness of fit measure, and we can examine its mean and variance.

However, it is preferable to use a measure that is demonstrable in terms of clustering in a shape property space. For this we again use principal components. We have available a library of waveforms recorded in similar experiments, and have calculated the principal components that best describe this set of waveforms. Recall that *all* spikes from our simulated data set consist of the same waveform plus noise. Each such waveform is projected onto the first and second of the library PCA components, thus defining a point that is plotted on a PC1–PC2 plane. This process is carried out over all spikes in the raw and cleaned data, and directly demonstrates in the relative cluster sizes the extent of improvement by the cleaning process. Since the clusters are roughly circular for the simulated data, we can draw a bounding circle that takes in 95% of the points by examining the distribution of distances from each cluster mean. The ratio of circle diameters for the raw and cleaned data gives a good single number measure of the performance of the cleaning algorithm. The same process carried out on real data rarely results in circular clusters because the occurrences of each underlying waveform have some degree of amplitude or shape variability. Furthermore there will be different clusters, and possibly more than one cluster from the spikes recorded on each electrode. For real data we simply examined the raw and cleaned clusters on each electrode, using longer data sets to obtain reasonable statistics.

For simulated data we used the assessment method above to study the cleaning results from the different algorithms as a function of noise properties and spike amplitudes. For real data we simply compared the results of the different algorithms.

3. Results

3.1. A typical generated data set

The simple simulator described above allowed full control of the relative amplitudes of spikes, individual channel noise and three independent sources of shared noise. In addition, the spike rate and inter-spike interval distribution (order of the gamma distribution), and the waveform used for the spikes, could be varied. All parameters were constant within a given simulation, so that for example there were no spike rate modulations that would mimic stimulus presentation. However, it should be noted that the algorithms we use for shared noise reduction are very insensitive to spike rate modulation (not shown but described in the section on tetrode recordings and near coincidences). Fig. 2A shows a typical one second data set. The spike amplitude was 0.4 (for the negative peak) and individual channel noise amplitude 0.1. The three shared noise sources each had an amplitude 0.4. The upper panel of Fig. 2C shows a superposition of many noise-contaminated waveforms from the raw data set of Fig. 2A.

3.2. The cleaning of simulated data

We applied all three algorithms to a wide variety of data from our simulator, examining a range of individual and shared noise levels relative to a constant spike amplitude. Fig. 2B and the lower panel of Fig. 2C give an example of the cleaning process using the PCA algorithm, for the data shown in Fig. 2A. There is an impressive reduction in noise, such that individual spikes are much more clearly evident and have much smaller variance of shape.

The noise reduction is demonstrated more quantitatively in Fig. 2D, which shows projections of the spike waveforms before (circles) and after (crosses) the cleaning procedure onto a PC1 and PC2 calculated from a library of spike waveforms. A 95% bounding circle has been drawn about each cluster, and for these data the ratio of diameters is 0.19 (cleaned/raw). For data with the same levels of individual noise but zero shared noise both circles have the same diameter as the inner circle in Fig. 2D (not shown). This means that the algorithm removes essentially all of whatever shared noise is present.

Full parametric studies for the three algorithms are shown in Fig. 3, plotting the bounding circle ratio (cleaned/raw) as determined from results like Fig. 2D over a range of individual (0–0.3) and shared (0–0.4) noise amplitudes for a fixed spike amplitude (1.0). We show the full performance surface for the PCA algorithm in Fig. 3A. Fig. 3B and C shows the *difference* between that surface and those produced by the regression and array algorithms, with vertical gain multiplied by 10. The differences in the performance of the three algorithms are small and apparently randomly distributed over the investigated parameter space; all three algorithms give equally good results. This 'experimental' result is corroborated in the Appendix where we

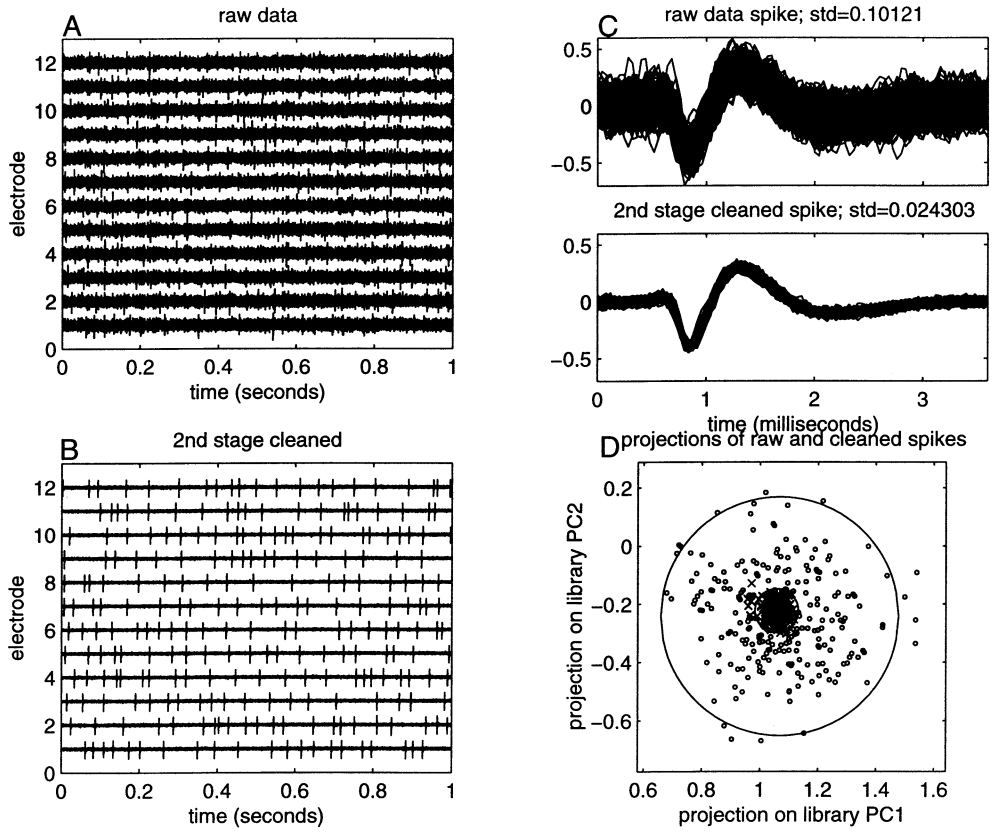

Fig. 2. (A) Simulated data mimicking 1 s of 12-channel recording sampled with 25 kHz frequency. There is a different independent noise on each channel with amplitude 0.1 and bandwidth 300 Hz–8 kHz; one source of shared noise is distributed equally across all channels, amplitude 0.4 and bandwidth 300 Hz–8 kHz there are two additional sources of shared noise distributed across all channels, each with a different spatial pattern, maximum amplitude 0.4 and bandwidth 300 Hz–5 kHz. An experimentally recorded spike waveform (scaled to negative amplitude equal to 0.4) is added to noise on all channels. Interspike intervals follow a fourth order gamma distribution; spike rate 30 per second. See text for further details. (B) Data from (A) after second stage of the cleaning process using the PCA algorithm. Note the extent of noise reduction. The cleaning process removes noise shared across channels and leaves spike waveforms with individual (i.e. channel specific) noise. (C) Upper panel: superposition of noise-contaminated spike waveforms from all channels of the raw data shown on (A). Lower panel: the same after second stage cleaning with the PCA algorithm. Standard deviations are given above each panel. (D) Projections of spike waveforms onto the first two principal components calculated from a library of experimentally recorded spike waveforms. '○': raw data, '×': after cleaning. Circles bounding 95% of the points in each cluster quantify the reduction of noise caused variance in the spike waveforms.

demonstrate the underlying mathematical similarity of the three algorithms. For each algorithm the performance surface given by the set of cluster diameter ratios is somewhat curved as a function of individual and shared noise amplitudes. Smaller values of the ratio indicate better cleaning, and occur for smaller values of individual noise but also for larger values of the shared noise amplitude. The latter result is deceptive: the algorithms are able to remove most of the shared noise no matter how big it is. Therefore the improvement seems larger when this shared noise is larger, since it forms a greater fraction of the total noise.

3.3. The cleaning of real data

Application of the regression cleaner to a noisy real data set is shown in Fig. 4, together with the corresponding projections of spike waveforms from one electrode onto a library PC1–PC2 plane. Arrows in Fig. 4A indicate shared spike like events that are removed by the cleaning process. Note also that much of the other 'noise' is also shared, since it gets removed.

With real data there may be different spike waveforms in each of the several electrodes, so that it is inappropriate to lump waveforms across channels as we did for the simulated data. Instead we process a longer stretch of data, the PCA calculation being applied over each consecutive second. Then waveforms from any particular channel can be collected and projected on the library principal components. The bounding circle is no longer appropriate. Accordingly Fig. 4C and D shows waveform projection planes before and after cleaning

for a particular electrode. In this case there is apparently only a single waveform that forms a diffuse cluster with a widespread background. As in the simulated data, the cluster size after cleaning is considerably smaller and denser than in the raw data. However, a considerable background scattering of points remains, so that although there is improvement, isolation of these identified spike events by a boundary on the plane remains imperfect. Clearly data that initially had a better spike/noise ratio would fare better; the cleaning methods described here are useful but can only do so much. For these data the other two cleaning algorithms produce very similar results to those in Fig. 4 (not shown).

PCA cleaning of another recording (made with only three electrodes) is shown in Fig. 5 as waveform projection planes. Panel A shows the waveform clusters for the raw data when spikes are detected by a threshold amplitude criterion in the raw data. There are many outlier waveforms, many of which turn out to be shared and removable; the total number of points $n = 2267$. Panel B shows the waveform clusters for the raw data, but now the spikes are detected by threshold *after* cleaning. Most of the outliers are removed compared to Panel A since spike detection is now much more reliable so that $n = 1699$. Finally Panel C shows the waveform clusters in the cleaned data with spikes detected in the cleaned data (as for Panel B). The difference between Panels B and C demonstrates the removal of shared noise from the spike waveforms since they show the same spike events before and after the cleaning process. Panel C has denser clusters with better separation. Note that there are three outliers in the lower part of Panel C which were not evident in Panel B. In the raw data (Panel B) these were lumped into the main clusters. After removal of the shared noise, however, these three events were sufficiently atypical that they became outliers. Below each panel there is a histogram

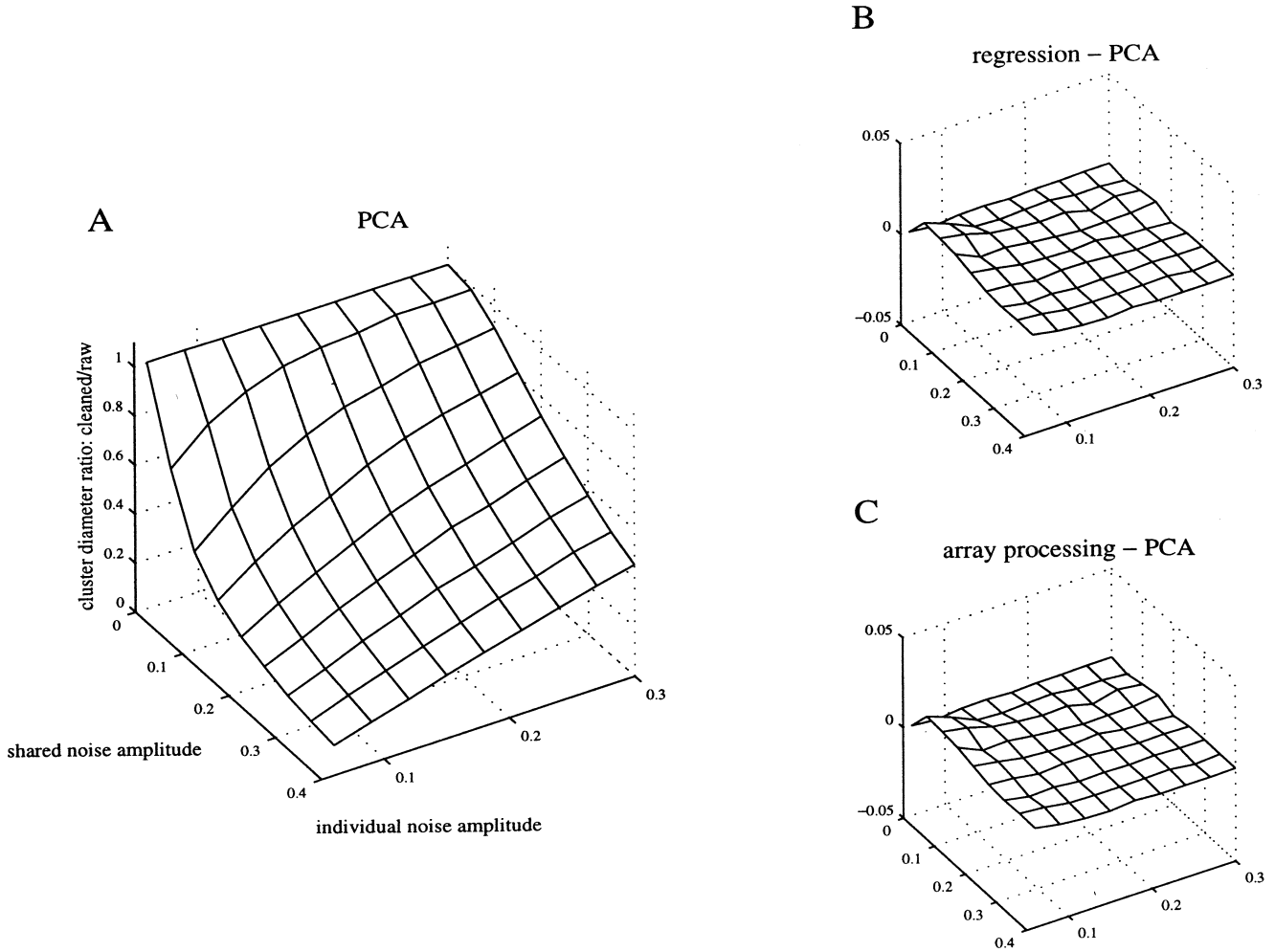

Fig. 3. Parametric performance surfaces for the three algorithms. Bounding circle diameter ratios (cleaned/raw) are determined as in Fig. 2D over a range of individual and shared noise amplitudes for a fixed spike amplitude of 1. The lower the ratio, the more effective the cleaning. (A) Surface for the PCA cleaner; (B) and (C) *differences* between the PCA surface and the regression and array surfaces. Note that the vertical gain in (B) and (C) is ten times that in (A). All three algorithms gave the same results within the investigated parameter range.

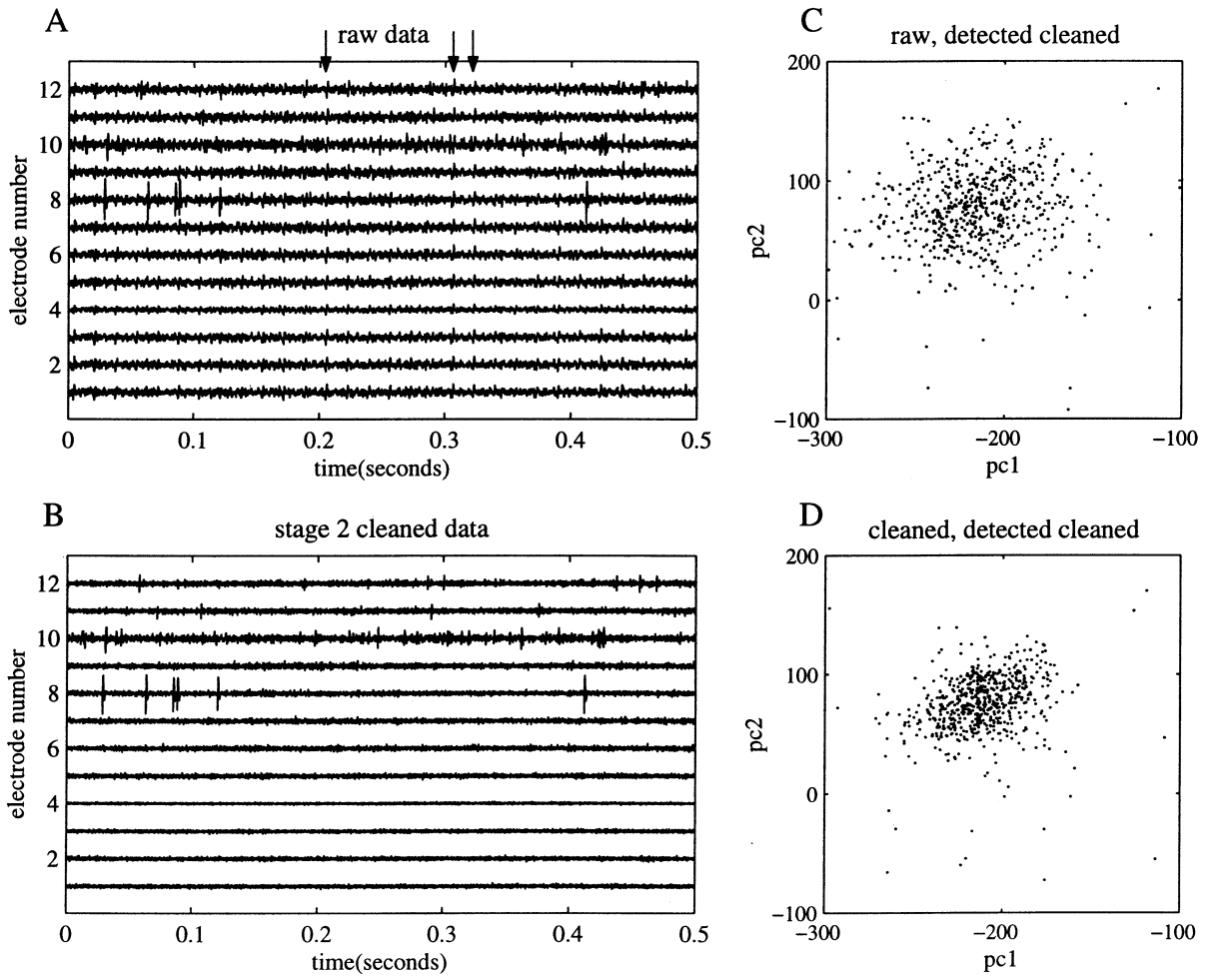

Fig. 4. Application of the regression cleaner to a real noisy data set obtained from multielectrode recordings in the rat auditory cortex. (A) Raw data. Arrows indicate shared noise events present on all channels. These artifacts (removed after cleaning, see the lower panel) would probably have been detected as neuronal spikes. (B) Same data after cleaning. Shared noise events and general background have been removed. Note that spikes on channels 11 and 12, hardly visible in the raw data, are easy to detect after cleaning. (C) Projections of spike waveforms from one electrode (channel 8 in (A)) onto the plane of first two principal components from the library. (C) Waveforms in the raw data but detected in the cleaned data; (D) waveforms in the cleaned data detected in the cleaned data. Note the cluster shrinkage although there are also some points left in a widespread background.

of the counts between PC2 = ± 100, demonstrating more quantitatively the successive reduction of outliers and improvement in waveform cluster density and separation.

4. Discussion

We propose two new algorithms for removing correlated noise from multichannel recordings made with arrays of micro-wires or other electrodes. We have compared these methods to the array processing algorithm (Applebaum, 1976) previously implemented for spike detection and sorting in multichannel data (Bierer and Anderson, 1999; Rebrik et al., 1999; only for detection). The first new method is based on principal component analysis. In this approach we subtract from the data in a given channel that part of its noise content that is correlated with the principal component representation of noise on the remaining channels. It is necessary to edit these remaining channels to remove spikes before computing the principal components. The second method is based on multiple linear regression. First we find a linear combination of the data on the other channels that best expresses (in terms of least squares) the noise in a given channel. Then, using these calculated regression coefficients, we subtract that linear combination of data on other channels from the one being processed. Although requiring different procedures, both new algorithms appear to be as effective as the array processing algorithm in simulation studies over a broad range of shared and individual noise parameters. As we show in the Appendix A, all three shared-noise cleaning methods, despite being based on

different concepts, are in fact mathematically equivalent. Each method however has some specific advantages that we mention in more detail later in this section. We have also given an alternate derivation of the array processing formula. This makes use of linear discriminant analysis to separate maximally signal and noise in multichannel data, and leads to the same recipe for optimal mixing of the data. Although this derivation does not lead to a new method we believe it helps to understand the topic.

All three algorithms have some common features. All of them require two steps. Initially the raw data spikes are often masked and distorted by the noise. Therefore, in a first stage, we apply the particular method to the raw data set in order to detect spikes better. At this stage the noise removal process is biased because its parameters were based on data that included spikes in the other channels (for PCA and regression) or all channels (for array). In addition the biased first stage process introduces small cross-channel artifacts. In a second stage, we then again process each channel, using edited data in which we substitute noise for the spikes detected in the first stage on the other channels (for PCA and regression) or all channels (for array). The noise used in these short spike-centered substitution windows is obtained by subtracting the first stage cleaned data from the original raw data. (In principle the entire process could be iterated, using successively improved estimates of the spike waveforms to estimate the noise substitution for the next stage. We found little improvement beyond the second stage as used throughout this paper.)

In experimental recordings, it is highly likely that the shared noise characteristics and distribution across channels can change in time. This suggests that noise processing should be done dynamically in a short and moving time window, typically about 1 s in duration. There is a trade off for all three methods: for short time windows, even in case of stationary noise, the results are more sensitive to individual channel noise and thus less reliable (although taking less computation time).

If the number of shared noise sources is small, the PCA method offers some reduction of computation time and memory usage (due to the more economic noise representation; see Appendix A) while maintaining high accuracy. There will be some relation between the number of independent shared noise sources and the number of PCs that must be used in the cleaning process. For our simulated data with three shared noise sources and for all the real data we have examined, the amplitude of PC3 is about an order of magnitude smaller than PC1 and PC2, and therefore has little influence on the cleaning process. It would however be a good rule-of-thumb to examine the relative magnitudes of PCs for the data being processed, and to carry the computation out to as many terms as appropriate.

With the recent development of independent component analysis (ICA; Lee, 1998; Brown et al., 2001 for a tutorial review) we attempted to use this method to denoise multichannel data but our attempts were not successful. One of the limitations of the ICA method at the current stage of development is that it requires the availability of at least as many recorded channels as the number of sources (Lee, 1998, p. 63). This is easily obtained in optical recording, but is rarely possible with

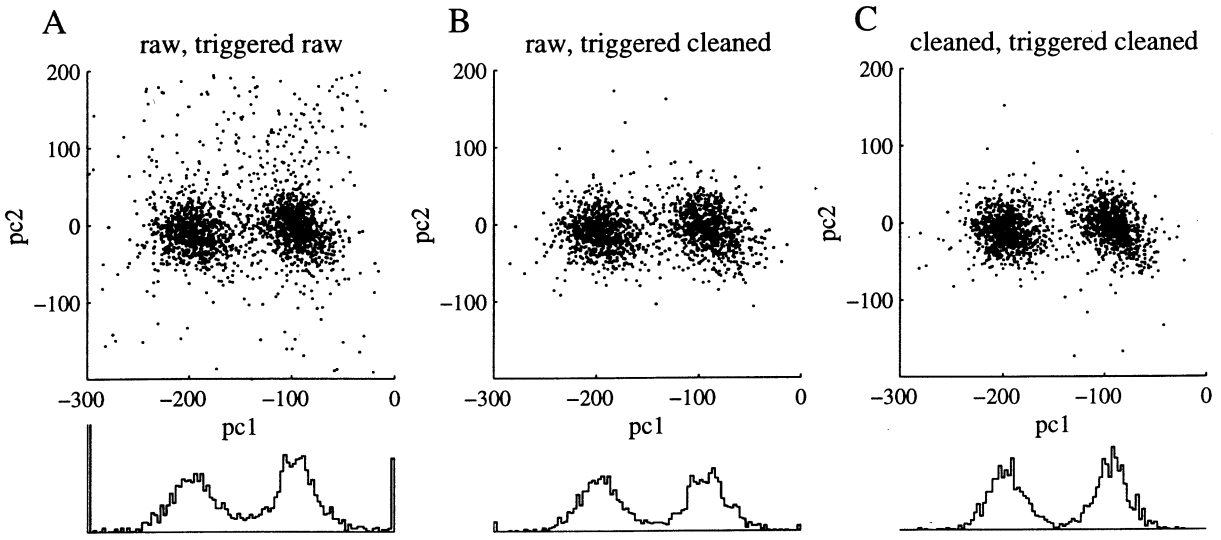

Fig. 5. Projections of spike waveforms from one electrode in a 3-channel recording. (A) Waveforms in the raw data detected in the raw data by amplitude threshold; (B) waveforms in the raw data but detected in the cleaned data; (C) waveforms in the cleaned data detected in the cleaned data. Each panel also has a histogram of the point counts between PC2 = ± 100. Note the narrowing of clusters in (C) after cleaning; this leads to better separation and therefore improvement of spike sorting.

electrodes. For N electrodes (channels) in a typical experiment we might expect $N/2$ (or more) spike trains, N individual noise sources, and several shared noise sources. The number of sources clearly outnumbers the N available channels and separation does not work.

Whereas array processing and PCA methods are both linear computations, the regression method can easily be generalized. Instead of a linear combination in the regression equation, it would be possible to incorporate nonlinear dependencies between channels. A possible candidate for such nonlinear effects would be capacitance between electrodes. A further enhancement to the procedure would be to include time delays into the regression equation. This would allow the solution of deconvolution problems, where there are shared sources of noise which are differently filtered by electrodes that have slightly varied characteristics. That variant of the regression method could also be helpful in the case of data multiplexed during the sampling process.

We have not explored non-linear methods like wavelets for either spike detection or noise reduction. Such application may be difficult when the spectral characteristics of noise and spikes are similar, as in the real and simulated data considered in this paper.

If there are fewer shared noise sources than the number of channels and if individual channel noise amplitude is negligible, there is a problem with redundant representation of noise. In this situation the covariance matrix becomes singular which means that array processing and the regression method are no longer valid. The PCA approach can still work assuming we use only as many principal components as the number of shared noise sources. However, this problem occurred only in simulated data when individual noise amplitudes were made considerably smaller than the shared noise amplitudes. We never faced the problem of covariance matrix singularity in experimental data because the individual channel noise was high enough.

Recall that both PCA and regression algorithms require a time-point by time-point subtraction. Accuracy in calculating the waveforms for this process requires a high sampling rate. A rule-of-thumb for satisfactory performance with our data is about 2.5 times the Nyquist limit at the highest frequency of the noise bandpass. Thus for an upper noise bandpass of 5 kHz we use a 25 kHz sampling rate (i.e. 40 μs between samples).

It should be noted that the cleaning algorithms described in this paper do not alter the underlying spike waveforms in any way as would for example a filter. The only effect of the cleaning procedure is to reduce whatever shared noise contribution there is to the spike shape variance, and thereby to improve performance of any subsequent shape sorting process. It should also be noted that since the second stage algorithms eliminate the influence of spikes on the shared noise estimate, and then operate on each time point individually, the cleaning performance is not affected by rate modulations of the spike trains or by amplitude non-stationarities of the shared noise. Individual channel noise, stationary or not, is of course unaffected by these algorithms.

The real and simulated examples given in this paper have all assumed recording arrangements as for typical spike train studies, with a bandpass of 300 Hz–5 kHz. However, such limitation is not fundamental or necessary. If real recordings include *shared* low frequency signal or noise, as might happen in a study of evoked or field potentials, the algorithms will be effective in eliminating them. Low frequency signal or noise that are individual to each channel will not be affected, and would cause serious problems if these same data are also to be used in subsequent spike shape sorting efforts. Adequate sorting requires appropriate preliminary high pass filtering.

In real spike data we do often observe high individual channel noise in comparison to shared noise. None of the three cleaning methods will then offer much improvement in the quality of spike waveforms. However, they will still get rid of unwanted shared events that may even mimic spikes; an example is abrupt movement artifacts present simultaneously on more than one channel. Routine application of cleaning algorithms to recorded data can only improve quality, costs little in computation time, and would seem highly worthwhile.

Acknowledgements

Supported by NIH Grants MH 46428 and DC 01249 to G.L.G. and a Wellcome Trust Fellowship to S.N.B. A program implementing the PCA cleaning algorithm in MATLAB 5.3 is available by sending an email request to software@mulab.physiol.upenn.edu.

Appendix A

We show in the following section that for the case of multichannel single wire recordings (spikes of any particular neuron present in one channel only) the array processing algorithm is equivalent to the two methods we propose in this paper (principal component analysis and multiple linear regression). For simplicity let us assume that the signal spikes come from only one neuron, and are present only on channel 1 (this may require renumbering of the K channels). Thus, in Eq. (9) we have $U = 1$, and $\mathbf{w}$ is a K-dimensional column vector. Its first component is one, and all remaining components are zeros ($\mathbf{w}(c_1) = 1$ and $\mathbf{w}(c_i) = 0$ for $i = 2,\ldots,K$). Eq. (9) then simplifies to:

$$\mathbf{Z}(t) = \mathbf{D}(t,c)\mathbf{C}_{\mathrm{N}}^{-1}\mathbf{w}(c) \tag{A.1}$$

At this point let us simplify the notation. Set $Z = \mathbf{Z}(t)$, $D_i = \mathbf{D}(t,c_i)$, $\mathbf{C}^{-1} = \mathbf{C}_{\mathrm{N}}^{-1}$, and $w_i = \mathbf{w}(c_i)$. Then

$$\begin{aligned} Z &= \sum_{i,j=1}^{K} (\mathbf{C}^{-1})_{ij} D_i w_j = \sum_{i=1}^{K} (\mathbf{C}^{-1})_{i1} D_i \\ &= (\mathrm{C}^{-1})_{11} D_1 + \sum_{i=2}^{K} (\mathrm{C}^{-1})_{i1} D_i \end{aligned} \tag{A.2}$$

But

$$(\mathbf{C}^{-1})_{ij} = (-1)^{i+j} \frac{\det \mathbf{M}_{ji}}{\det \mathbf{C}} \tag{A.3}$$

where $\mathbf{M}_{ji}$ is a submatrix of $\mathbf{C}$ obtained after deleting the jth row and ith column (Bretscher, 1997, p. 259).

$$\begin{aligned} Z &= D_1 \frac{\det \mathbf{M}_{11}}{\det \mathbf{C}} + \sum_{i=2}^{K} D_i (-1)^{i+1} \frac{\det \mathbf{M}_{1i}}{\det \mathbf{C}} \\ &= \frac{\det \mathbf{M}_{11}}{\det \mathbf{C}} \left(D_1 - \sum_{i=2}^{K} D_i (-1)^{i} \frac{\det \mathbf{M}_{1i}}{\det \mathbf{M}_{11}} \right) \end{aligned} \tag{A.4}$$

Let us use the Laplace expansion of $\det \mathbf{M}_{1i}$ along the first column of $\mathbf{M}_{1i}$ (Bretscher, 1997, p. 285):

$$\det \mathbf{M}_{1i} = \sum_{k=2}^{K} c_{k1} (-1)^{k} \det \mathbf{N}_{ki} \tag{A.5}$$

where c_{k1} are elements of the covariance matrix $\mathbf{C}$ and $\mathbf{N}_{ki}$ is a submatrix of $\mathbf{M}_{1i}$ obtained after deleting the $(k-1)$th row and the first column (note that the rows of $\mathbf{M}_{1i}$ are numbered from 2 to K).

Denote the normalization factor $(\det \mathbf{M}_{11}/\det \mathbf{C})$ by η. Thus

$$\begin{aligned} Z &= \eta \left(D_1 - \sum_{i=2}^{K} D_i \frac{(-1)^{i}}{\det \mathbf{M}_{11}} \sum_{k=2}^{K} c_{k1} (-1)^{k} \det \mathbf{N}_{ki} \right) \\ &= \eta \left(D_1 - \sum_{i,k=2}^{K} c_{k1} (-1)^{i+k} \frac{\det \mathbf{N}_{ki}}{\det \mathbf{M}_{11}} D_i \right) \\ &= \eta \left(D_1 - \sum_{i,k=2}^{K} c_{k1} (\mathbf{M}_{11}^{-1})_{ik} D_i \right) \end{aligned} \tag{A.6}$$

Using matrix notation and choosing to express the D_i as a row vector and the c_{k1} as a column vector (the reverse choice is also possible but less convenient):

$$Z = \eta \left(D_1 - [D_2 \cdots D_K] \mathbf{M}_{11}^{-1} \begin{bmatrix} c_{21} \\ \vdots \\ c_{K1} \end{bmatrix} \right) \tag{A.7}$$

Since $\mathbf{M}_{11}$ is in fact a covariance matrix of noise in channels 2,…,K, the expression subtracted in Eq. (A.7) from D_1 is a multiple linear regression formula for approximation of noise in channel 1 by noises in channels 2,…,K (Winer, 1971). Note that while D_1 consists of spikes and noise, there are no spikes present in D_2,…,D_K so after the subtraction channel 1 is left with the spikes and individual noise. In other words, within a normalization factor η, and for the case that spikes of a given neuron are only present on channel 1, we have transformed Eq. (9) for the array algorithm into Eq. (A.7) which is the regression algorithm proposed in this paper.

Let us now define a new matrix $\mathbf{B}$ fulfilling the condition that $\mathbf{B}\mathbf{B} = (\mathbf{M}_{11})^{-1}$.

Eq. (A.7) can be rewritten as:

$$Z = \eta \left(D_1 - [D_2 \cdots D_K] \mathbf{B}\mathbf{B} \begin{bmatrix} c_{21} \\ \vdots \\ c_{K1} \end{bmatrix} \right) \tag{A.8}$$

The term $[D_2 \cdots D_K]\mathbf{B}$ represents data in channels 2,…,K after decorrelating by $\mathbf{B}$. Let us denote this transformed data by $[\tilde{D}_2 \cdots \tilde{D}_K]$. Note that the covariance matrix of $[\tilde{D}_2 \cdots \tilde{D}_K]$ is unity (Applebaum, 1976) which means that transformed vectors are orthogonal to each other and each of unitary variance.

Since there is no DC component in noise the covariance coefficient c_{k1} is

$$c_{k1} = \frac{1}{T} \sum_{t=1}^{T} D_k(t) n_1(t)$$

where $n_1(t)$ is the noise on channel 1. Thus, the term

$$\mathbf{B} \begin{bmatrix} c_{21} \\ \vdots \\ c_{K1} \end{bmatrix}$$

can be transformed in the following way:

$$\begin{aligned} \mathbf{B} \frac{1}{T} \sum_{t=1}^{T} \begin{bmatrix} D_2(t) n_1(t) \\ \vdots \\ D_K(t) n_1(t) \end{bmatrix} &= \frac{1}{T} \sum_{t=1}^{T} \left(\mathbf{B} \begin{bmatrix} D_2(t) \\ \vdots \\ D_K(t) \end{bmatrix} \right) n_1(t) \\ &= \frac{1}{T} \sum_{t=1}^{T} \begin{bmatrix} \tilde{D}_2(t) \\ \vdots \\ \tilde{D}_K(t) \end{bmatrix} n_1(t) \\ &= \begin{bmatrix} \tilde{c}_2(t) \\ \vdots \\ \tilde{c}_K(t) \end{bmatrix} \end{aligned} \tag{A.9}$$

where $\tilde{c}_k$, $k = 2,\ldots,K$, is a set of covariance coefficients of noise on channel 1 with noises on the other channels after being transformed by $\mathbf{B}$. Now we can rewrite Eq. (A.8) as

$$Z = \eta \left(D_1 - \sum_{k=2}^{K} \tilde{D}_k \tilde{c}_k \right) \tag{A.10}$$

Since vectors $\tilde{D}_k$, $k = 2,\ldots,K$, have unitary variances (and zero mean values) Eq. (A.10) can be interpreted as subtraction from D_1 of that part of its noise content that is correlated with vectors constituting an orthogo-

nal representation of noise in the remaining channels. However, this procedure can not be done unambiguously based just on raw data in channels 2,…,K because there may be correlation of the raw data across channels. The outcome would depend on the order in which the channels were used in processing. The procedure would favor channels taken first while reducing contributions of the later ones. The orthogonal representation of noise determined by **B** as in Eq. (A.8) is only one of many possibilities. The representation based on the principal components analysis proposed in this paper is different but also orthogonal. Thus the PCA and regression algorithms are conceptually related insofar as both subtract a (different) orthogonal representation of the noise on the other channels from the channel being processed. The PCA approach is particularly advantageous (see also Section 4) since it offers reduction in dimensionality of the noise representation. This usually means economy in terms of memory usage and computational time.

References

Applebaum SP. Adaptive arrays. IEEE Trans Antennas Propag 1976;AP-24:585–98.

Baker SN, Gerstein GL. Improvements to the sensitivity of gravitational clustering for multiple neuron recordings. Neural Computat 2000;12:2597–620.

Bierer SM, Anderson DJ. Multi-channel spike detection and sorting using an array processing technique. Neurocomputing 1999;26–27:947–56.

Blum JR, Rosenblatt JI. Probability and statistics. Philadelphia, PA: W.B. Saunders, 1972.

Bretscher O. Linear algebra with applications. Upper Saddle River, NJ: Prentice-Hall, 1997.

Brown GD, Yamada S, Sejnowski TJ. Independent component analysis at the neural cocktail party. Trends Neurosci 2001;24:54–63.

Gerstein GL. Cross-correlation measures of unresolved multi-neuron recordings. J Neurosci Methods 2000;100:41–51.

Grün S, Diesmann M, Grammont F, Riehle A, Aertsen A. Detecting unitary events without discretization of time. J Neurosci Methods 1999;94:67–79.

Gray CM, Maldonado PE, Wilson M, McNaughton B. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. J Neurosci Methods 1995;63(1–2):43–54.

Lee T-W. Independent component analysis. Theory and applications. Dordrecht: Kluwer Academic, 1998.

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

McNaughton BL, O'Keefe JO, Barnes CA. The stereotrode: a new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records. J Neurosci Methods 1983;8:391–7.

Rao CR. Linear statistical inference and its applications. New York: Wiley, 1973.

Rebrik SP, Wright BD, Emondi AA, Miller KD. Cross-channel correlations in tetrode recordings: implications for spike-sorting. Neurocomputing 1999;26–27:1033–8.

Winer BJ. Statistical principles in experimental design. New York: McGraw-Hill, 1971:69–71.