Performance of real time separation of multineuron recordings with a DSP32C microprocessor

R. Gädicke *, K. Albus

Department of Neurobiology, MPI für Biophysikalische Chemie, Postfach 2841, 37 070 Göttingen, Germany

Abstract

The performance of a method for sorting of waveforms in multi-neuron data (Gädicke and Albus, 1995) is evaluated by using artificial spike patterns generated by the computer and by adding to these spikes noise or free running sine waves of varying frequency and amplitude to simulate EEG-waves. The DSP32C is capable of continuously processing spikes at 183.106 Hz. In addition to real-time sorting the DSP32C also performs a running average of the spikes sorted into each class and transfers data to the host computer. The ability of the system to analyse burst of activity is determined by the FIFO memory buffer (2048 samples, or 32.768 ms at 62.5 kHz sampling rate). Adding a 50 Hz sine wave discrimination worked correctly with sine wave amplitudes of up to 2.5 times that of the smallest spike. Combining spikes with noise revealed errors of inclusion and/or exclusion of less than 0.1% provided the models spikes were determined from noiseless spikes and the spike threshold was set above the noise peak level. When noisy spikes were used to define model spikes about 4% of the smallest amplitude spikes (signal to noise ratio 3.3) were incorrectly classified. For higher amplitude spikes (signal to noise ratio ≥ 5) the classification error was on average less than 1%. The artificial patterns used for performance testing are exactly defined and could be used to standardize the comparison between different sorting techniques. © 1997 Elsevier Science Ireland Ltd.

Keywords: Computer separation of waveforms; Complete spike form matching; Standard performance test; Multiunit recording

1. Introduction

We have recently described a method for real-time separation of multineuron recordings (Gädicke and Albus, 1995). The high processing speed of the DSP32C signal processor is utilized to implement a robust algorithm for waveform discrimination. The digitized (62.5 kHz) waveform itself is used as the model spike, and new spikes are assigned to a particular class when they match the respective model spike. Each electrode is served by its own processor and the algorithm allows for comparing and sorting complete waveforms in real time into eight different models per electrode.

In the present report the performance of our method is critically evaluated. Since human performance in sorting of waveforms in multi-neuron data is unreliable (Sarna et al., 1988), we decided to use artificial spike patterns generated by the computer and to add to these spikes noise or free running sine waves of varying frequency and amplitude. Such patterns are exactly defined and can be used to standardize the comparison between different sorting techniques.

2. Methods

The principles of operation of our method are described in detail in the original report (Gädicke and Albus, 1995). Here only a short summary will be given. After determination of the DC voltage offset and spike thresholds, model spikes are defined on the basis of 259

* Corresponding author. Tel:. + 49 551 2011652; fax: + 49 551 2011788; e-mail: RGADIC@GWDG.DE

segments of the recording. Each segment is 3 ms long and contains at least one complete spike. The procedure to define model spikes includes selecting a 2 ms period in each 3 ms segment, defining the starting time of the spike, and calculating parameters for each spike component. Values of a given parameter can be selected and the mean of the spikes from which these parameters were computed defined as a model spike. Real time sorting is performed by executing the actual match/mismatch test only after an incoming spike has been synchronized in time with model spikes.

The noiseless artificial spikes used for the performance testing were generated by 12 Bit D/A converters updated at a rate of 62.5 kHz. Due to a hardware restriction, the artificial spikes including interspike interval must be a multiple of 8192 samples, or 0.131072 s.

The artificial spikes used for most of the testing are displayed in Fig. 1. The initial component is positive in the first four spikes, and negative in the second four. The interspike interval (the time between the start of two successive spikes) was set to 256 sample points (4.096 ms). Each spike lasted for 128 sample points (2.048 ms). The duration of the test runs (between 44 and 47 s) was manually adjusted in order to collect approximately 1000 spikes in each of the eight spike classes.

Sine waves were generated by a conventional free running waveform generator. The output of a preamplifier was used as a noise source. The input of the preamplifier was connected to a 10 MOhm carbon film resistor.

In order to quantify the discrimination, in addition to the time of occurance of the spikes and the spike class to which they were assigned the waveforms of the new spikes matched to a spike class or the outlier class were stored in the host computer. These data were then
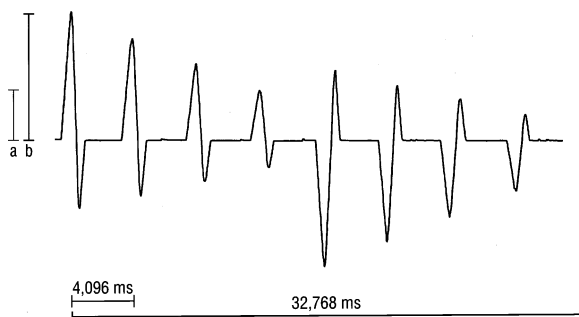

Fig. 2. Correct discrimination performance at 183.106 Hz.(a) Relative amplitude of biggest spike = 1.0. Eight spikes of the 32 spikes are absent.

compared with the order and the time of occurance of spikes fed into the system which were known to the experimenter.

When classifying errors made by our device in sorting waveforms to a particular class we followed in most parts the distinction made by Sarna et al. (1988) an exclusion error occurs when a given spike properly assigned to class j is incorrectly assigned to some other class i, including the outlier class. An inclusion error is when a given spike properly assigned to some other class i is incorrectly assigned to class j. Accordingly, as stated by Sarna and collaborators (1988), errors of inclusion will in part correspond to errors of exclusion from other classes. An exception occurs when artifacts are recognized as spikes (for example originating from noise; see below); if these are not assigned to the outlier class but to a spike class this error is classified as a pure error of inclusion. The eight spike classes were numbered 1–8 and the outlier class assigned the number 0. Part of the results has been published in abstract form (Gädicke and Albus, 1996).

3. Results

3.1. (a) Continuous firing

Given the D/A update rate of 16 μs, the technique for generating artificial spikes resulted naturally in waveforms 32.768 ms long. Each waveform consisted of eight different spikes, a total of 2048 samples; the interval between spikes was 4.096 ms as shown in Fig. 1. The order of the different spikes remained constant. The ratio of biggest spike to smallest spike is 1:0.4 or 2.5:1.

The DSP32C was not capable of processing spikes at the full rate of 32 spikes within 131.072 ms (244.141 Hz). The discrimination worked correctly when eight of the 32 spike intervals contained no spikes, leaving 24 spikes within 131.072 ms (183.106 Hz, see Fig. 2). This result was independent of whether the matching period

Fig. 1. Noiseless artificial spikes used for the performance testing. The amplitude ratio of the spikes (from left to right) is 1.0: 0.8: 0.6: 0.4: −1.0: −0.8: −0.6: −0.4. This input signal is measured prior to a 4th order high-pass filter. (a) Relative amplitude 0.4; (b) relative amplitude 1.0; spike amplitudes and time were measured on a digital oscilloscope. Small errors in the graph were caused by nonlinearities in the time axis of the plotter.
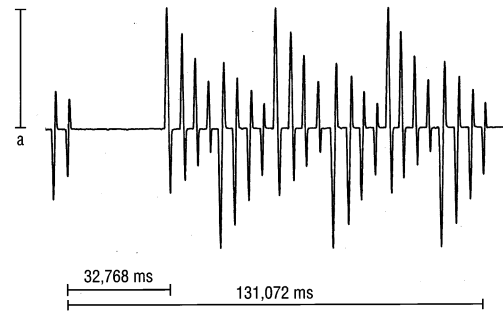
(see Gädicke and Albus, 1995) was set to 1 or 2 ms. The DSP32C performed not only real-time sorting, but also maintained a running average of the spikes sorted into each class and transferred data to the host computer.

3.2. (b) Bursts

Short spikes were generated at rates up to 2 kHz. It was found that at 2 kHz the spike was correctly assigned to the respective model spike when the burst duration was limited to 32 ms. As explained previously, the DSP is unable to process spikes at this high rate. Therefore the burst length was limited to the size of the FIFO input buffer associated with each DSP32C. The FIFO buffer is 2048 samples, or 32.768 ms at 62.5 kHz. The ability of the system to analyse burst of activity is thus determined by the FIFO memory buffer.

3.3. (c) Matching range and sorting performance

Usually the entire spike is used for the matching procedure. Alternatively only a short segment of a spike can be used for the matching procedure as well (Gädicke and Albus, 1995). For testing the limits of this procedure (shortening the matching period), two spikes having an amplitude ratio of 1:0.95 were generated (Fig. 3). The interspike interval was set to 4.096 ms and the spike duration to 2.048 ms. The two spikes alternated and the firing frequency was set to 183.106 Hz (see above).

The matching period of one spike was held constant (54 sample points × 16 μs = 0.864 ms) while the matching period of the second spike was shortened. The two spikes were correctly discriminated down to a matching range of the second spike of nine sample points (0.144
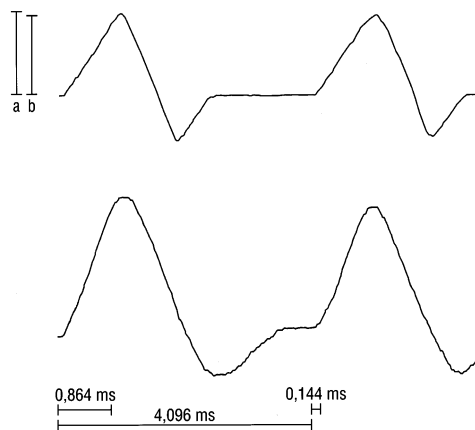

Fig. 3. Matching range and sorting performance. Noiseless spikes before (top) and after (bottom) high pass filtering. Relative amplitudes: (a) 1.0 for spike on the left; (b) 0.95 for spike on the right. Amplification setting for filtered spikes is different from that for unfiltered spikes.
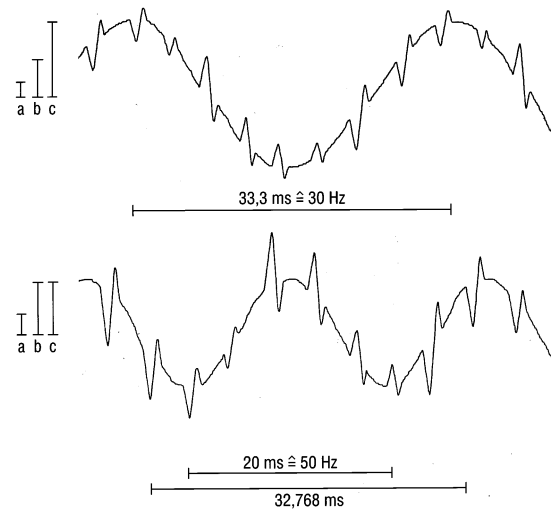

Fig. 4. Influence of slow waves parameters on spike discrimination. The eight different spikes presented at 183.106 Hz are added to sine waves of varying amplitude and frequency. Top: relative amplitude of biggest spike = 1.0 (b), of smallest spike = 0.4 (a) and of 30 Hz sine wave = 2.0 (c). Bottom: relative amplitude of biggest spike = 1.0 (b), of smallest spike = 0.4 (a) and of 50 Hz sine wave = 1.0 (c).

ms). If the matching range was further shortened (to six sample points = 0.096 ms) spike 2 was assigned to the model spike 1. This might be due to D/A and A/D errors, as well as rounding errors during the signal processing.

3.4. (d) The influence of slow wave (EEG) parameters on spike discrimination

The eight different spikes shown in Fig. 1 and presented at 183.106 Hz (Fig. 2) were added to sine waves of varying amplitude and frequency (Fig. 4). The timing of the spike train was independent of sine wave phase. The matching range set for all spikes was 54 sample points which is approximately 1 ms. The Bode plot of 4th order high pass filter used for spike filtering is given in Fig. 5. The result of varying sine wave amplitude and frequency on spike discrimination is shown in Table 1.

As can be seen from Table 1 (bold numbers), with the addition of a 50 Hz sine wave, discrimination was performed correctly with a sine wave amplitude of up to 2.5 times that of the smallest spike. With the addition of a 30 Hz sine wave, discrimination was performed correctly with a sine wave amplitude of up to five times that of the smallest spike.

3.5. (e) Influence of noise on spike discrimination performance

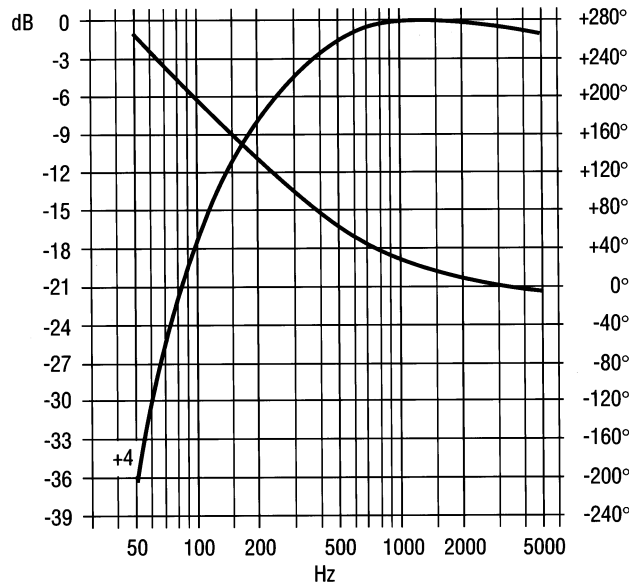Noise amplitude is determined in three different ways (Fig. 6):

Fig. 5. Bode plot of 4th order high pass filter used for spike filtering.

3.5.1. U_{eff} (effective noise amplitude)

The effective amplitude U_{eff} of any AC signal is given by

$$U_{eff} = \sqrt{\frac{\sum_{n=1}^{N} (u(t) - U_0)^2}{N}} \tag{1}$$

The DC offset voltage U_0 is calculated using a 3 s recorded data segment (100 FIFOs full = $100 \times 2048 \times 16\ \mu s = 3.2768$ s). The mean value of these 204 800 data points is subtracted from each data point recorded during subsequent 3 s periods. The difference at each data point is then squared, the sum of the squares is divided by the number of data points and the square root of this result is taken as U_{eff} noise.

The standard deviation σ of any signal is given by Formula 2.

Table 1
Proportion of spikes classified as outliers as a function of slow wave parameters

5:1	3.7	0.2x	1.0	19.7	35.2	—	—	—
4:1	0.1	0.1	0.3	12.3	31.0	39.8	43.9x	—
3:1	0.1	0	0.1	2.7	25.1	36.6	39.0	—
2:1	0	0	**0**	0.2	8.5	30.3	35.4	36.8
1:1	0	0	0	0	**0**	5.5	20.5	29.4
Hz	10	20	30	40	50	60	70	80

Ordinate: ratio of sine wave amplitude (peak to zero voltage) to amplitude (peak to zero voltage) of biggest spike. The ratio of biggest spike to smallest spike is 1:0.4 or 2.5:1. Abscissa: sine wave frequency. x: The classification counted more spikes than were supplied in the test data. These spikes represent various combinations of sine wave fragments with spike components; they were incorrectly assigned to the model spikes representing the spikes with the smallest amplitude. The proportion of these waveforms was less than 0.1% at 20 Hz (see x at 5:1) and less than 5% at 70 Hz (see x at 4:1).
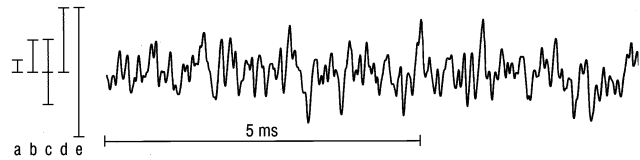

Fig. 6. Noise amplitudes a: U_{eff} noise (effective noise amplitude); b: Us90 noise (Uss90 noise/2); c: U_{ss}90 noise; d: U_s noise (U_{ss} noise/2); e: U_{ss} noise (peak to peak noise amplitude).

$$\sigma = \sqrt{\frac{\sum_{n=1}^{N} (u(n) - E)^2}{(N-1)}} \tag{2}$$

E is the mean value of the distribution. The standard deviation σ of the distribution is thus calculated in the same way as Ueff except that $N-1$ sample points are used instead of N for normalising the data. With large sample sets as in our case the values calculated using equations Eq. (1) and Eq. (2) are almost identical.

$u(t) = u(n) =$ voltage at t;
$U_0 = E =$ DC offset voltage;

3.5.2. U_{ss} = peak to peak noise amplitude

U_{ss} was calculated for five 3.2768 s periods which is approximately 200 000 sample points for each period (s. above). Given this amount of data of a gaussian distribution, more or less all sample points should be located within $\pm 4\ \sigma$ (Bronstein and Semendjajew, 1987). Since with such large sample sets the mean value is almost identical to the standard deviation (see above, Eq. (1)) and Eq. (2)) we expected U_{ss} to be 9.13 times U_{eff}. In fact, results ranged between 8.8 and 10.8 times U_{eff}. For practical reasons we set

$U_{ss} = 10 \times U_{eff}$;

The close agreement between the theoretical prediction and actual measurement indicates that the noise analyzed in our case has a gaussian distribution.

3.5.3. Estimation of noise amplitude by visual inspection (oscilloscope trace: analog mode, 10 ms per grid division)

To estimate the peak-to-peak noise, we drew a horizontal line through the middle of the positive-going noise spikes and another through the negative-going noise spikes. We estimated the noise amplitude as the difference between the two lines. By comparing the estimated noise value $U90_{ss}$ with U_{ss} noise and U_{eff} noise, respectively, we found we could approximate the relationship by:

U_{ss} noise $\cong 2.2 \times U_{ss}90$ noise (sample size = 200 000);
U_{eff} noise $\cong U_{ss}90$ noise/4.5;

The relationship between the three different quantities (including U_s noise = U_{ss} noise/2) is displayed in Fig. 6.

Table 2
Parameters for testing the influence of noise on spike discrimination performance

Parameters	Condition 1	Condition 2
Duration of test run	48.1 s	47.7 s
U_s biggest spikes	4.06	3.42
U_s smallest spikes	1.63	1.37
$U_{ss}90$ noise	1.0	1.0
U_s90 noise	0.5	0.5
U_{eff} noise	0.195	0.216
U_{ss} noise	2.05	2.26
U_s noise	1.02	1.13
Spike threshold	1.1	0.93
Confidence limit for all spikes (one side)	0.88	0.74
Signal/noise ratio for smallest spike = U_s smallest spikes/U_s90 noise	1.63:0.5 = 3.26	1.37:0.5 = 2.74

Parameter values were normalized on the basis of $U_{ss}90$ noise = 1; U_s spikes = amplitude of spikes/2; under condition 1 U_s values for spikes with intermediate amplitudes were 3.25 and 2.44, respectively; U_s90 noise = $U_{ss}90$ noise/2; U_s noise = U_{ss} noise/2; spike threshold and confidence limits, see legend to Fig. 7.

To test the influence of noise on spike discrimination we proceeded as follows. The model spikes were determined from the spikes as given in Fig. 1, that is, without noise and firing at a rate of approx. 183 Hz (see above). The matching range for all spikes was set to 0.864 ms (54 sample points). Noise was added to the spike train by means of a passive resistor network.
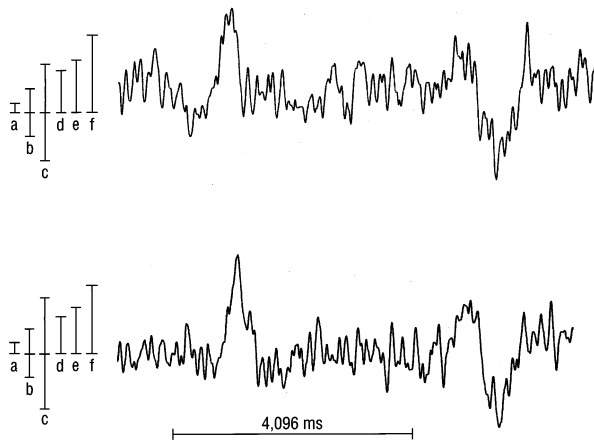

Fig. 7. Adding noise to the spike train. a: U_{eff} noise; b: $U_{ss}90$ noise; c: U_{ss} noise; d: confidence limits; e: positive spike threshold; f: spike amplitude without noise. The trace on top corresponds to condition 1, that at bottom to condition 2. Spike amplitude without noise/U_s90 noise: 3.26 (top); 2.74 (bottom). Confidence limits: confidence limits for matching of incoming spike; normally set to one standard deviation of the model spike (see Gädicke and Albus, 1995). Spike threshold: 'start spike sampling threshold' (see Gädicke and Albus, 1995).

Spike classification was tested at two noise amplitudes (condition 1, and 2, respectively; Table 2 and Fig. 7). In addition, the sorting performance was tested by using model spikes defined from noisy data (artificial spikes with noise; condition 3). In this condition, parameter settings were the same as in condition 1 with the exception of spike threshold and confidence limits which were systematically varied.

3.6. Result condition 1 (Fig. 7 top)

Spikes, 8808 (1101 for each of the eight classes) were generated and with the exception of 11 spikes (0.12%) all were assigned correctly to their respective classes (numbered 1–8). The 11 spikes were errors of exclusion; these spikes which would have been correctly assigned to one of the classes 1–8 were incorrectly assigned to class 0 (outliers, nine cases) or to a neighbour class (errors of both exclusion and inclusion, two cases).

3.7. Result condition 2 (Fig. 7 bottom)

Spikes, 8744 (1093 for each of the eight classes) were generated and a total of 8964 spikes were discriminated. The 220 additional spikes represented low amplitude noise components. Most of them (206) were erroneously added to one of the two classes representing the spikes with the smallest amplitude and could therefore be classified as errors of inclusion. The remaining 14 spikes were classified as outliers and thus assigned to class 0. From the 8744 spikes, 8712 were correctly assigned to their respective classes. 16 spikes which would have been correctly added to one of the classes 1–8 were classified as outliers (errors of exclusion); another 16 spikes were incorrectly assigned to a neighbour class and thus represented both errors of exclusion and of inclusion. Taken together under condition 2 the total errors of inclusion accounted to [220 + 16 =] 236 spikes ([236/8964 =] 2.6%) and the total errors of exclusion to 32 spikes ([32/8964 =] 0.36%).

3.8. Result condition 3

With optimal settings of spike threshold and confidence limits the proportion of misclassified spikes was generally less than 2%. For example, in one of these test runs 8304 spikes were generated (1038 for each of the eight classes); with the exception of 116 spikes (1.4%) all were correctly assigned to their respective classes. From the 116 spikes 90 spikes were assigned to class 0 (exclusion errors) and 26 spikes to a neighbour class (errors of both exclusion and inclusion). The classification error was related to spike amplitude: it was on average 3.8% for the smallest amplitude spikes ($U_s = $ 1.63, see Table 2) and 1.1, 0.4 and 0.3% for the spikes

with U_s values of 2.44, 3.25 and 4.06, respectively (see Table 2). Thus, about 70% of the misclassified spikes belonged to the classes of the smallest amplitude spikes; almost all of the errors of inclusion were attributable to smallest amplitude spikes incorrectly assigned to their neighbour classes.

4. Discussion

The classification differs between condition 1 and condition 2 because the spike threshold in condition 1 was slightly above U_s90 noise whereas in condition 2 it was slightly below U_s90 noise. Thus in condition 2 some noise peaks crossing the spike threshold were erroneously assigned to the two spike classes representing the spikes with the smallest amplitudes. When noisy spikes were used to define sorting parameters the sorting performance became worse, in particular for the smallest amplitude spikes with a signal to noise ratio of approximately 3.3 (Table 2). In these classes the classification error was on average 4%; for the spikes with a signal to noise ratio of about 5 it was 1.1% and for spikes with signal to noise ratios of 6.6, or more than 6.6 it was less than 0.5%.

In order to avoid errors of inclusion as occured under condition 2, the following parameter settings should be performed:

spike threshold $> U_s90$ noise;

U_s spike without noise $> 1.6 \times U_{ss}90$ noise;

U_s spike with noise $> 2 \times U_{ss}90$ noise.

These settings should also ensure that errors of inclusion as observed under condition 3 are minimized.

Testing the performance of sorting devices on spike set defined by humans (Sarna et al., 1988) revealed that classification errors depended on spike amplitude. The largest discrepancies between devices for spike sorting and humans were in dealing with the smallest spikes.

The signal to noise ratio of the smallest spike (class A4) as estimated from Fig. 1b in Sarna and collaborators (1988) is approximately 2.5. In this case errors of both inclusion and exclusion were quite large (more than 20%) for both the principal component sorter and a software based sorter. Classification errors for spikes with signal to noise ratios of approximately 4–5 (classes A2 and A3 in Fig. 1b of Sarna et al., 1988) ranged between 10 and 20%.

The performance of our system could be improved by using a processor faster than the DSP32C. Significantly faster processors are already on the market. The performance could also be improved somewhat by reducing the sampling frequency. Through such improvements, high frequency bursts lasting longer than 32 ms as well as continuous firings up to approximately 500 Hz should be discriminated in real time.

References

Bronstein IN, Semendjajew KA. Taschenbuch der Mathematik. Frankfurt/Main: Hari Deutsch, 1987.

Gädicke R, Albus K. Real time separation of multineuron recordings with a DSP32C signal processor. J Neurosci Methods 1995;57:187–93.

Gädicke R, Albus K. Performance of real time separation of multineuron recordings with a DSP32C microprocessor. Proceedings 24th Göttingen Neurobiology Conference, Thieme, Stuttgart, vol. II, 1996:821.

Sarna MF, Gochin P, Kaltenbach J, Salganicoff M, Gerstein GL. Unsupervised waveform classification for multi-neuron recordings: a real-time, software based system. II. Performance comparison to other sorters. J Neurosci Methods 1988;25:189–96.