Design Optimisation of Front-End Neural Interfaces for Spike Sorting Systems

Deren Y. Barsakcioglu*†, Amir Eftekhar*†, and Timothy G. Constandinou*†
*Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT, UK
†Centre for Bio-Inspired Technology, Institute of Biomedical Engineering, Imperial College London, SW7 2AZ, UK
Email: {deren.barsakcioglu10, a.eftekhar, t.constandinou}@imperial.ac.uk

***Abstract*—This work investigates the impact of the analogue front-end design (pre-amplifier, filter and converter) on spike sorting performance in neural interfaces. By examining key design parameters including the signal-to-noise ratio, bandwidth, filter type/order, data converter resolution and sampling rate, their sensitivity to spike sorting accuracy is assessed. This is applied to commonly used spike sorting methods such as template matching, 2^{nd} derivative-features, and principle component analysis. The results reveal a near optimum set of parameters to increase performance given the hardware-constraints. Finally, the relative costs of these design parameters on resource efficiency (silicon area and power requirements) are quantified through reviewing the state-of-the-art.**

I. Introduction

Understanding how the trillions of action potentials of the brain's billions of neurons produce our thoughts, perceptions, and actions is one of the greatest challenges of 21st century science. The ability to connect these neurons to electronics is presenting new opportunities for neural rehabilitation with prosthetic devices. For example, cochlear implants, are already impacting the quality of life of over 300,000 individuals with profound deafness.

Experimental recording of large numbers of neurons is thus an extremely important task, but one that requires overcoming several technical challenges. Recent years have seen the development of micro-fabricated neural probes such as the Utah and Michigan arrays, now commonplace in experimental labs, and likely soon in clinical applications such as brain-machine interfaces for paralysis [1], [2]. For any portable or implantable device, such probes require miniature electronics locally to amplify the weak neural signals, filter out noise and out-of-band interference and digitise for transmission. With recent advances in modern semiconductor technology, this is now possible and has sparked significant research activity in the community, particularly in this last decade [3]–[12].

The specifics of the electrode material, the electrode/tissue interface as well as the nature of the bio-potential signal itself pose challenges on the front-end microelectronics. The signals observed contain an electrode offset potential (due to the electrode-electrolyte interface) as well as both the extracellular action potentials (EAPs) and local field potentials (LFPs). The EAPs typically have amplitudes of $25\,\mu\text{V}-1\,\text{mV}$ with a signal band of $100\,\text{Hz}-3\,\text{kHz}$, whereas the LFPs have amplitudes up to $5\,\text{mV}$ with a signal band of $0.5-300\,\text{Hz}$. Additionally, the electrode-electrolyte interface introduces an offset that can be several 100's of mV, with the microelectrodes themselves contributing thermal noise due to their relatively high impedance. All these factors dictate the minimum requirements for the front-end electronics, that are additionally limited by resource constraints (power, size and bandwidth). In particular, the desire to make such systems implantable poses limits on size and thermal dissipation (i.e. to prevent tissue damage), as well as requiring wireless transmission (i.e. thus limiting communication channel capacity) [13], [14].

Spike sorting is a technique commonly used on EAP recordings to separate the signal into spike patterns of individual units (i.e. neurons). This is based on the fact that the dynamics of each neuron varies, in addition to the topological placement of the microelectrodes (i.e. in orientation and proximity). This results in each neuron having a slightly different spike profile when observed at the electrode and can be identified by means of feature extraction followed by clustering. There exists a multitude of different methods for achieving the feature extraction (eg. templates, peaks, derivatives, wavelets, principle component analysis), and clustering (eg. valley detection, k-means, super-paramagnetic) [15].

This work investigates trade-offs imposed by front-end circuits on spike sorting performance and hardware resources. The paper is organised as follows: Section II identifies the front-end requirements, Section III describes methods used in the optimisation, Section IV details the different parameter sensitivities on spike sorting accuracy, Section V discusses the resource efficiency, and Section VI draws conclusions.

II. Front-End Neural Interface Architecture

A typical architecture for a front-end neural interface (for applications using EAPs) is illustrated in Fig. 1. This consists of three main blocks: (1) the bio-potential amplifier, (2) bandpass filter, and (3) data converter. This section describes the requirements for each of these blocks:

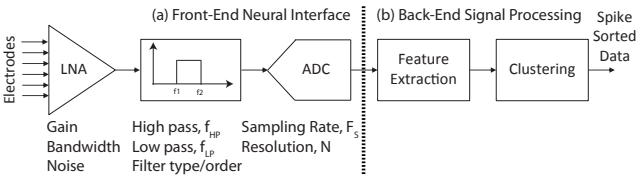

Fig. 1. General architecture of a brain machine interface (BMI) showing: (a) front-end neural interface, and (b) back-end spike sorting

1) Bio-potential Amplifier: A low noise preamplifier is required to boost the signal level from sub-mV to 10's of mV, such that the subsequent stages can have relaxed noise performance. This is often designed to be AC-coupled such as to remove any DC levels due to electrode offset, and typically provides a gain of 50−200, with a bandwidth of 3−10 kHz, and input-referred noise of $1-10\,\mu V_{rms}$. Key parameters for subsequent spike sorting are thus the Gain-Bandwidth Product (GBW) and input-referred noise (IRN).

2) Bandpass Filter: Following pre-amplification, a band-pass filter is required to: (1) reject the out-of-band LFPs (high pass), and (2) prevent aliasing (low pass). The high-pass cut-off frequency is typically set between 100−300 Hz, and low-pass between 3−10 kHz. Due to the close proximity between the high- and low-pass cut-off frequencies, a sharp response is required to avoid in-band attenuation and it is thus desirable to use a high order filter. A key challenge is however, to minimise the effect of phase distortion as this will impact subsequent signal processing. Key parameters for the filter design are therefore the filter type, order, and low-/high-pass cut-offs.

3) Data Converter: The main design specifications for the analogue-to-digital converter (ADC) are the resolution and sampling rate (typically 8-10 bit, and 16−32 kS/s). Although these set the numerical accuracy in subsequent spike sorting computation, this is fundamentally limited by the signal-to-noise ratio (SNR) and bandwidth of the signal. The resolution and sampling rate must therefore be optimised to minimise the raw data throughput.

III. Methodology

The optimisation is achieved by investigating the effect of each design parameter (i.e. those relevant to the front-end) on spike sorting performance within a simulation environment using Mathworks Matlab™ R2011b v7.13.

A. Spike Sorting

For the results reported herein, the spike sorting performance is quantified using the effective accuracy, calculated by: % total spikes classified × % spikes correctly classified (excluding spike detection). This has been benchmarked for 3 different, relatively computationally-efficient methods for spike sorting:

1) Template Matching (TM): Aligning the maximum peak of the signal with a template and using the Squared Euclidean Distance as a similarity/distance measure. The templates are created by taking the mean of the spikes (within each cluster), aligned to their individual maximum peaks.

2) Principle Component Analysis (PCA): Taking the first two principle components (for each spike) and using k-means for clustering (50 iterations). Uses the Matlab™ in-built functions (*princomp* and *kmeans*).

3) 2^{nd} Derivative Features (DER): Taking the minimum and maximum values of the 2^{nd} derivative (within each spike) and using k-means for clustering (50 iterations). Uses the Matlab™ in-built functions (*gradient* and *kmeans*) [16].

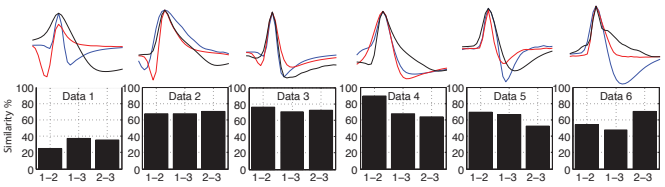

Fig. 2. Mean spike profiles for the 6 datasets with corresponding Bray-Curtis similarity measures applied between each neuron cluster within each dataset.

B. Test Datasets

The methods were applied to a total of 24 datasets based on simulated extracellular recordings [15], [17]. These contain 6 different groups (each using 3 single units). Each group is then used to generate 4 datasets at varying SNR levels (26, 20, 16.5, and 14 dB). The different SNRs have been obtained by superimposing attenuated spike waveforms such as to mimic the background activity observed at the electrode. The amplitudes and timing of both the spike and noise signals have been randomly distributed within preset ranges, such as to achieve the desired SNR and firing rate (with a target of 50 spikes/sec). Fig. 2 illustrates the mean ('spikes') of each cluster together with corresponding similarity measures.

IV. Design Optimisation

The front-end design parameters tested are: (1) SNR, (2) high-pass response, (3) low-pass response, (4) resolution, and (5) sampling rate. Each test is repeated on all 6 dataset groups with results showing the mean and spread. In all tests (with exception of high-pass response), the LFPs have been removed using a high order, zero-phase filter. In tests not involving noise variations, an SNR of 20 dB is used as we found it to be representative of the trends observed.

A. Signal-to-Noise Ratio

The overall effect of SNR on spike sorting accuracy is tested for all 6 dataset groups at the 4 SNR levels (as described previously). This represents the overall effective SNR including *noise* in both the front-end electronics and the signal itself (i.e. due to the background activity). The results (Fig. 3) show that by reducing the SNR (over the range 14−26 dB), the spike sorting accuracy decreases by up to 30% (depending on spike sorting method used). The results also show that the algorithm based on second derivative features has the highest noise immunity, with template matching being the most sensitive to noise.

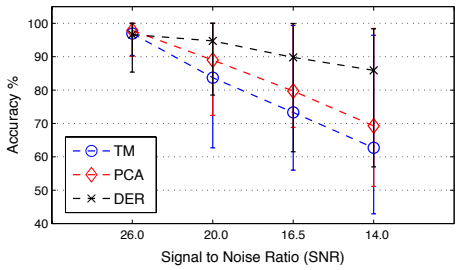

Fig. 3. Effect of SNR on spike sorting accuracy of the overall system (mean and spread shown over all data sets).

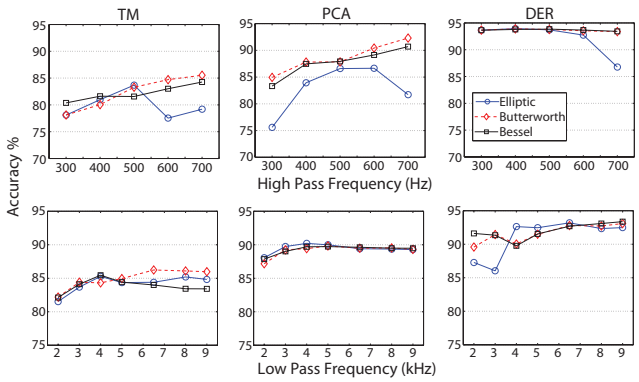

Fig. 4. Effect of high pass (top) and low pass (bottom) filtering on spike sorting accuracy (mean over all datasets for a noise level of 0.1, i.e. SNR = 20 dB). This is shown for 2^{nd} order Elliptic, Butterworth and Bessel filters.

B. High-pass Response

This test examines the effect of the high-pass filter response on spike sorting accuracy. Key parameters in the high-pass filter design are: filter type, order and cut-off frequency (i.e. f_{3dB}). To emulate the effect of LFPs we extracted (200 Hz low-pass filtered and normalised for a 5:1 LFP to EAP amplitude ratio) LFPs from a hippocampus data set from the CRCNS database. These were tested by pre-filtering the datasets using 2^{nd} order Elliptic, Butterworth and Bessel filter types at f_{3dB}={300,400,500,600,700}Hz. The results (Fig. 4) show that the high-pass frequency marginally increases the overall spike sorting accuracy across all datasets and filter types. This is mainly the result of LFP rejection coupled with phase distortion due to the analogue filtering.

C. Low-pass Response

This test examines the effect of the low-pass filter response on spike sorting accuracy. Key parameters in the low-pass filter design are as those previously identified for the high-pass. Again, these were tested by pre-filtering the datasets (using SNR=20 dB) using 2^{nd} order Elliptic, Butterworth and Bessel filter types at f_{3dB}={2,3,4,5,6.5,8,9,10,11.5}kHz. The results (Fig. 4) show that increasing the low pass cut-off improves the spike sorting accuracy as no phase distortion is introduced. However, it can be observed that beyond 4-6 kHz, there is moderate improvement in performance. Also the filter type appears to have negligible impact on performance (1.5% difference between types). In both low- and high-pass filtering, 2^{nd} vs. 4^{th} order filter implementations yielded negligible accuracy differences.

D. Resolution

This test examines the effect of data converter resolution (i.e. quantisation) on spike sorting accuracy (with scaling of the dataset to optimally fill the input range of the converter). The results are shown in Fig. 5(a). It can be observed that although the performance does increase with resolution, it saturates beyond 5−6 bit resolution. This maximum useful resolution is ultimately limited by the SNR. However, since the amplitude of the observed EAPs can vary, typically, by one order of magnitude, additional resolution is needed (i.e. 2−3 bit), given the fixed amplification gain.

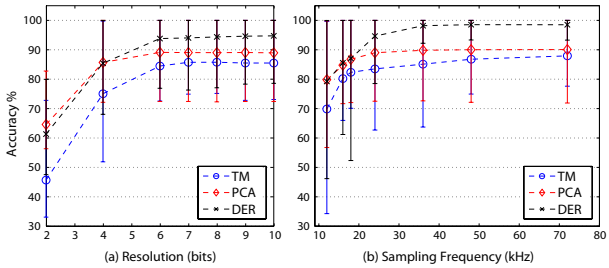

Fig. 5. Effect of data conversion (a) resolution, and (b) sampling rate on spike sorting accuracy (mean and spread shown over all datasets for a noise level of 0.1, i.e. SNR = 20 dB).

E. Sampling Rate

This test examines the effect of data converter sampling rate on spike sorting accuracy. The results are shown in Fig. 5(b). It can be observed that increasing the sampling rate can significantly increase spike sorting accuracy, however saturating beyond 24−36 kHz. Increasing sampling rate is expected to improve performance, since this captures finer features which may help further differentiate the signals.

F. Template Similarity

In addition to these front-end design parameters, we also investigated the correlation of template similarity to accuracy at different noise levels for all three spike sorting methods. We hypothesise that the maximum similarity (using the Bray-Curtis similarity measure) between each of the three templates will account for the errors seen in sorting. For TM the similarity is measured between time-domain templates (as shown in Fig. 2). For PCA similarity was determined based on the extracted PCA components (first two) for each template and for derivatives the similarity is based on the 2^{nd} derivative of the templates (as per the spike sorting method). These results, illustrated in Fig. 6, show that there is a relation between the difficulty of template separation and spike sorting accuracy. Overall, the derivative approach shows the best tolerance to noise and dissimilarity.

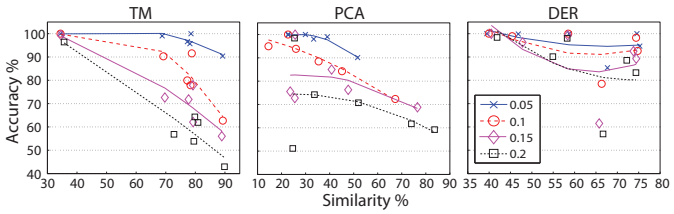

Fig. 6. The relation of accuracy and noise to template similarity (using the Bray-Curtis measure) for all three spike sorting methods.

V. Hardware Resource

Hardware resource requirements rely on key parameters that include power and area. Table I reviews recently reported state-of-the-art neural interfaces, listing the key design parameters

TABLE I
REVIEW OF STATE-OF-THE-ART NEURAL INTERFACES COMPARING KEY DESIGN PARAMETERS AND RESOURCE UTILISATION

		Amplifier				Filter			Data Converter			
Ref.	Tech. (μm)	Noise (μV_{rms})	Gain (dB)	Area (mm^2)	Power (μW)	Order, Type	Low-pass (kHz)	High-pass (Hz)	ENOB (bits)	Samp. Rate (kS/s)	Area (mm^2)	Power (μW)
[3]	0.5	5.1	60	0.16	42	1, G_mC	5	300, 800	10	15	0.15‡	1‡
[4]	0.13	1.95	38.3	-	12.5	-	-	-	8	10−100	0.27†	1
[5]	0.35	2.5	39.2	0.4†	0.03	1, G_mC	0.03−0.3	0.005−3.6	12	-	0.3†	0.2
[6]	0.35	4.9	40	0.07†	6.6	-	2−20	0.1−200	9	640	0.6†	-
[7]	0.13	14	40	0.3†	0.6	-	-	-	8	-	0.1†	-
[8]	0.35	6.1	54−73	-	12.8	-	5	0.5−50	8	111	-	2.8
[9]	0.18	5.4	49−66	0.03	16.7	G_mC	11.7	350*	8	125	0.02	1.9
[10]	0.13	2.2	-	-	-	2, G_mC + SC	10	280	10	31.3	0.3	1.1
[11]	0.35	2.3−2.9	40−75	0.9	371‡	4	0.2−6.2	2.6−572	-	-	-	-
[12]	0.13	3.8	47.5	0.06†	1.9	-	4.8−9.8	11.5-167	8	22.5−90	0.05†	0.5−1.8

Notes: †area estimated from microphotograph, ‡reported for multiple channels (value scaled per channel)

and achieved specifications for the pre-amplifier, filter and ADC (using reported data). In general for the pre-amplifier and data converter, the resource requirements are approximately directly proportional to the specifications. Specifically, the noise/gain for the pre-amplifier, and the resolution/sampling rate for the data converter are proportional to power consumption and area (depending upon technology used). The resource requirements for the filter are however more variable, due to the number of different applicable design methodologies. The filter order is typically the only parameter that is directly related to resource requirements.

VI. CONCLUSION

With next generation neural interfaces targeting 100s to 1000s of channels, the resource-budget on the front-end electronics are becoming increasingly stringent. To inform front-end optimisation, this work has quantified the effect of varying key design parameters on the performance of downstream spike sorting. By running our tests on multiple datasets that have been generated based on realistic parameters, the *ground truth* (i.e. absolute accuracy) is reported. By testing three different methods (template matching, principle component analysis and 2^{nd} derivative feature) for spike sorting, the trends to parameter sensitivities are shown to be broadly applicable.

The results show that for systems operating in a high-noise environment, the 2^{nd} derivative feature method is most robust (compared to TM and PCA). Furthermore, reducing the SNR, degrades the accuracy of spike sorting accuracy by approximately 1% per dB (for DER), 2% per dB (PCA), and 3% per dB (TM). With filtering, in particular the phase distortion introduced by the high pass filter significantly degrades the spike sorting accuracy, and thus a zero-phase or linear-phase response filter here is highly desirable. The optimum cut-off frequencies for the low- and high-pass filters have been determined to be 500 Hz and 8 kHz, with a 2^{nd} order response being sufficient. Finally, the data converter resolution is fundamentally limited by the neural SNR, and thus the input-referred system noise performance should be comparable. For SNR=20 dB, the optimum resolution for the converter is 8 bit (allowing 2−3 bit for variation in spike amplitudes) with an optimum sampling rate of between 24−32 kHz.

ACKNOWLEDGMENT

This work was funded by the EPSRC grant EP/I000569/1. We would like to thank Dr. Andrew Jackson and Prof. Quian Quiroga for their valuable input and continued collaboration.

REFERENCES

[1] M. Velliste, et al., "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008.

[2] J. Simeral, et al., "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array," *J. Neural Eng.*, vol. 8, p. 025027, 2011.

[3] R. Harrison, et al., "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE JSSC*, vol. 42, no. 1, pp. 123–133, 2007.

[4] S. Rai, et al., "A 500μw neural tag with 2μvrms afe & freq.-multiplying mics/ism fsk transmitter," *Proc. IEEE ISSCC*, pp. 212–213, 2009.

[5] X. Zou, et al., "A 1-v 450-nw fully integrated programmable biomedical sensor interface chip," *IEEE JSSC*, vol. 44, no. 4, pp. 1067–1077, 2009.

[6] M. Chae, et al., "A 128-channel 6 mw wireless neural recording ic with spike feature extraction and uwb transmitter," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 17, no. 4, pp. 312–321, 2009.

[7] Z. Xiao, et al., "A 20μw neural recording tag with supply-current-modulated afe in 0.13μm cmos," *Proc. IEEE ISSCC*, pp. 122–123, 2010.

[8] F. Shahrokhi, et al., "The 128-channel fully differential digital integrated neural recording and stimulation interface," *IEEE Trans. BioCAS*, vol. 4, no. 3, pp. 149–161, 2010.

[9] W. Wattanapanitch and R. Sarpeshkar, "A low-power 32-channel digitally programmable neural recording integrated circuit," *IEEE Trans. BioCAS*, vol. 5, no. 6, pp. 592–602, 2011.

[10] H. Gao, et al., "Hermese: 96-channel full datarate direct neural interface in 0.13μm cmos," *IEEE JSSC*, vol. 47, no. 4, pp. 1043–1055, 2012.

[11] C. Lopez, et al., "A multichannel integrated circuit for electrical recording of neural activity, with independent channel programmability," *IEEE Trans. BioCAS*, vol. 6, no. 2, pp. 101–110, 2012.

[12] A. Rodriguez-Perez, et al., "A low-power programmable neural spike detection channel with embedded calibration and data compression," *IEEE Trans. BioCAS*, vol. 6, no. 2, pp. 87–100, 2012.

[13] Z. Zumsteg, et al, "Power feasibility of implantable digital spike sorting circuits for neural prosthetic systems," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 13, no. 3, pp. 272–279, 2005.

[14] A. Eftekhar, S. Paraskevopoulou, and T. Constandinou, "Towards a next generation neural interface: Optimizing power, bandwidth and data quality," *Proc. IEEE BioCAS*, pp. 122–125, 2010.

[15] R. Quian Quiroga, et al., "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," 2004.

[16] S. Paraskevopoulou, D. Barsakcioglu, M. Saberi, A. Eftekhar, and T. Constandinou, "Feature extraction using first and second derivative extrema (fsde) for real-time and hardware-efficient spike sorting," *J. Neuroscience Methods.*, 2013.

[17] F. Thorbergsson, et al., "Minimizing data transfer with sustained performance in wireless brain-machine interfaces," *J. Neural Eng.*, vol. 9, no. 3, 2012.