

Clockless Continuous-Time Neural Spike Sorting: Method, Implementation and Evaluation

Yan Liu^{*†}, João L. Pereira^{†‡}, 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

[‡]Institute of Biophysics and Biomedical Engineering, University of Lisbon, 1749-016 Lisboa, Portugal

Email: {yan.liu06, t.constandinou}@imperial.ac.uk, joao.lovegrove@gmail.com

Abstract—In this paper, we present a new method for neural spike sorting based on Continuous Time (CT) signal processing. A set of CT based features are proposed and extracted from CT sampled pulses, and a complete event-driven spike sorting algorithm that performs classification based on these features is developed. Compared to conventional methods for spike sorting, the hardware implementation of the proposed method does not require any synchronisation clock for logic circuits, and thus its power consumption depend solely on the spike activity. This has been implemented using a variable quantisation step CT analogue to digital converter (ADC) with custom digital logic that is driven by level crossing events. Simulation results using synthetic neural data shows a comparable accuracy compared to template matching (TM) and Principle Components Analysis (PCA) based discrete sampled classification.

I. INTRODUCTION

There has been an increasing trend in developing large scale neural instrumentation requiring a compact physical size and minimal power consumption [1], [2]. Such devices provide new opportunities for experimental neuroscience, in studying fundamentals of the brain, in addition to various biomedical applications, such as neural prosthetics and Brain Machine Interfaces (BMI) [2], [3]. With modern microelectronic technology, it is now possible to integrate hundreds of neural recording channels in a single silicon chip with a total power consumption in the order of a few milliwatts [1], [3].

A typical neural recording architecture is shown in Fig. 1(a). This generally consists of an analogue front end including amplifier(s), filter(s), and an ADC for converting the analogue signal into digital sampled data (using a fixed sampling frequency). Whilst the static power consumption of the front-end amplifier is fundamentally limited by noise requirements, the dynamic power consumption is dependant on the neural information bandwidth and resolution requirements. For extracellular recordings, there are two main components to the neural signal: the Local Field Potential (LFP), a slow moving signal, and the Extracellular Action Potential (EAP) or spikes, and fast changing activity.

This spiking activity requires a low noise, (relatively) high bandwidth instrumentation with a large number of recording sites for observing local neural circuit, or network/multi-site activity. Scaling to larger channel counts linearly increases the static power consumption for mixed-signal circuits, as well as the dynamic power consumption in signal processing and interface circuits.

For applications that analyse or utilise single unit (neuron) activity, the extracellular recording contains no information

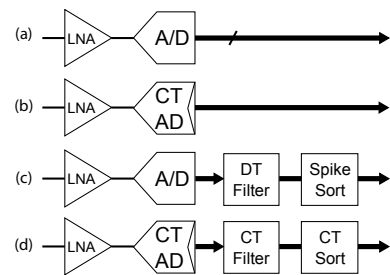


Fig. 1. Different architectures for neural recording interfaces: (a) discrete-time sampling with direct data output; (b) CT sampling with direct pulse stream; (c) discrete-time sampling with multi-unit, or spike sorted output; (d) CT sampling with CT-based spike processing.

other than the spike timing of each neuron. Here, spike sorting can achieve data reduction with no information loss, provided a good classification accuracy can be achieved [4]. Spike sorting is however typically performed off-line using computationally demanding methods [5], [6]. Recently, algorithms are emerging that are computationally-efficient [7], [8] towards efficient on-node implementations (see general architecture in Fig. 1(c)) for providing a significant reduction in data bandwidth. Conventional embedded hardware is however based on synchronous (clocked) logic that operates on uniformly sampled data and fixed resolution and is therefore suboptimal given the sparse nature of biological signals.

CT signal processing has recently emerged as a data-driven method that features a reduced quantization noise but also that power consumption is dependant on the input activity [9]. Recent work has adopted this technique in biomedical instrumentation for neural recording [10] (see general architecture in Fig. 1(b)). Other work has started to explore biosignal processing in the CT domain, for example, in QRS detection for ECG applications [11].

In this work, we have identified a set of features for representing neural spike features based on CT sampling, and developed a novel spike sorting algorithm that operates entirely in CT (see concept in Fig. 1(d)). A basic implementation is proposed using a variable-step CT ADC with 8-bit resolution, together with asynchronous combinational logic. This uses only linear operations and is entirely driven by event pulses. The remainder of this paper is organized as follows: Section II describes the CT sampling and feature extraction concept and also demonstrates the sorting performance based on these features, Section III describes the hardware implementation, Section IV evaluates the proposed system and compares to other methods, and finally Section V concludes this work.

II. CT NEURAL SIGNAL PROCESSING

The basic CT sampling scheme based on a 4-bit level-crossing ADC is shown in Fig. 2. Here, the top plot shows the input signal (in blue) and sampled data (in red). The bottom plot shows the generated pulse train for positive level crossings, i.e. input is half a LSB larger than the previously sampled state. By accumulating the positive and negative pulses, the CT sampled data can be reconstructed by $S(x_i, t_i)$, if considering the signal is two dimensional, and for any i , $|x_i - x_{i-1}| = \text{LSB}$. It can be found (i.e. in a similar manner to the difference respect to time), the difference of the signal with respect to data is given by:

$$\frac{\Delta S(x, t)}{\Delta x} = t_i - t_{i-1} = dt_i \quad (1)$$

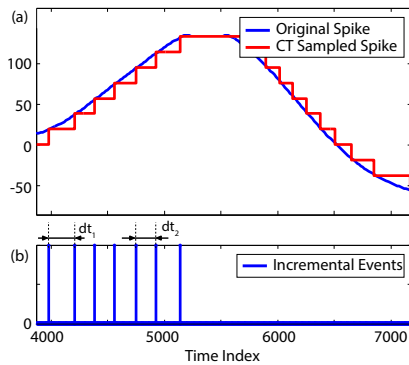


Fig. 2. CT-based sampling/data conversion. Shown are: (a) input and sampled data; (b) incremental pulse trains.

This provides an interesting indication about how the signal changes: with a smaller dt , e.g. dt_2 shown in Fig. 2, the signal increases faster in that value interval compared to the signal with time difference dt_1 . This is effectively a representation of derivative of the signal. Moreover, the *minimum* value of these derivatives reflects the *maximum* increase or decrease gradient of the signal. Since derivative extrema (with respect to time) can be used to classify spikes [12], these CT-based derivatives can be used as spike features in a similar manner. Furthermore the difference between these derivatives indicate the inflection points of the signal, which have negative values near the peaks.

These time derivative (TD) features in the CT domain along with the peak value of the spikes can be used to classify the different spike types. In Fig 3, two sets of neural spikes extracted from synthetic data with different SNRs are shown. The x-axis represents the peak value, y-axis the minimum value of TD in incremental trains, and z-axis the minimum value of TD in decremental trains (with y- and z-axes plotted on a log scale). Here it can be observed that the four different spike types can be easily distinguished for a high SNR, whereas for the lower SNR the clusters begin to overlap.

In [13], a variable-step CT ADC has been implemented to perform parallel comparisons with different step sizes at multiple level-crossing points. This scheme effectively reduces power consumption without compromising speed. This conversion scheme is illustrated in Fig. 4, with the top plot showing the input signal and sampled data, and bottom plot showing pulse trains with various pulse height (in discrete value of ± 1 , ± 2 , ± 4 and ± 8). From a derivative (i.e. feature) point of view, a

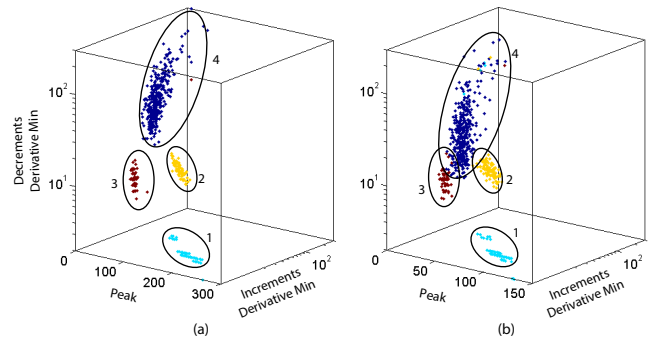


Fig. 3. Neural spikes plotted in 3-dimensional feature space. The features used are: peak value (x-axis); minimum incremental derivative (y-axis); minimum decremental derivative (z-axis). Plots shown for: (a) high SNR, and (b) low SNR.

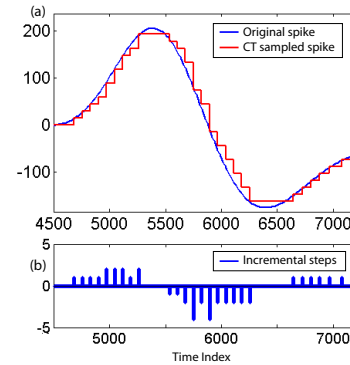


Fig. 4. CT-based sampling/data conversion with variable steps [13]. Shown are: (a) input and reconstructed sample; (b) variable step pulses. trains.

larger step means a faster changing input. It can be observed here that higher steps occur at the rising and falling edges of the spike peaks, and the more the large steps the sharper the signal. Thus the total number of different steps for both incremental and decremental crossings can be used as spike features in a similar manner as the time derivatives, but without requiring any extra timing.

The feasibility of using these new feature sets is examined in Fig. 5(a). This uses the same data as previously in Fig. 3, here with the x-axis representing the peak value, y-axis the total number of -2 steps, and z-axis the total number of $+2$ steps. Four different clusters can be easily observed. However, the number of ± 2 steps varies over a range of $[10, 80]$ for cluster

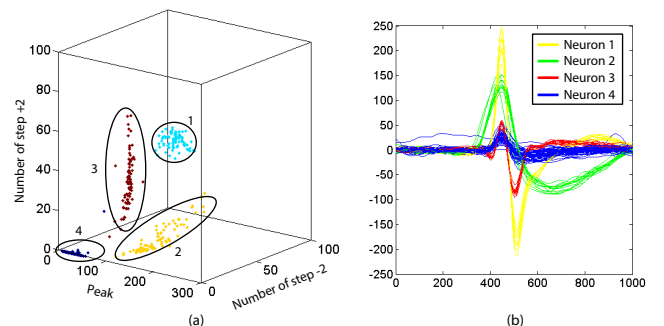


Fig. 5. Feature extraction using CT variable step features. Shown are: (a) neural spikes plotted in 3-dimensional feature space. The features used are: peak value (x-axis), number of $+2$ steps (y-axis), number of -2 steps (z-axis); (b) spikes separated using PCA and k -means clustering.

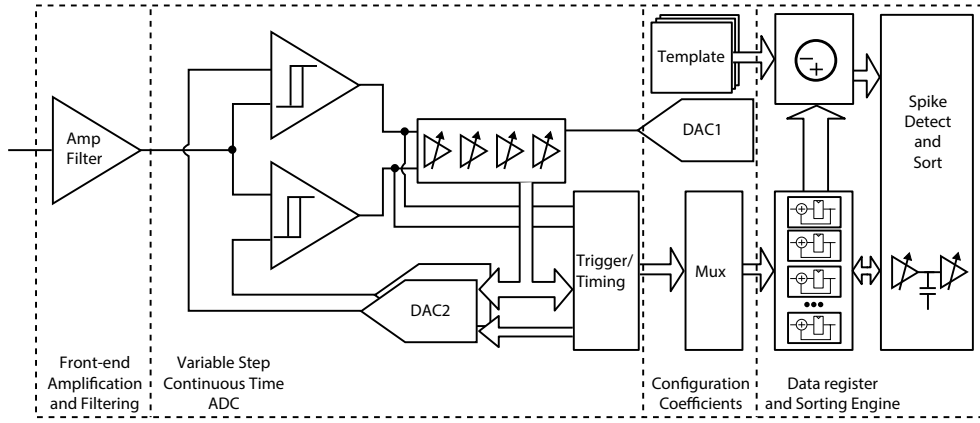


Fig. 6. Complete system architecture including analogue front end, CT-ADC (based on [13]), spike sorting and configuration memory.

2 and 3. This is because a step of ± 2 can be triggered by added noise, resulting in an increased feature spread. This can however be improved by utilising multiple summations or a longer delay in the ADC. The complete feature sets extracted from the same data of 20 seconds [14], including peak value, ± 2 , ± 4 , and ± 8 steps were then clustered using PCA + k -means. Results shown in Fig. 5(b) illustrate the different spike waveforms using different colours (according to the detected cluster). This clearly demonstrates that the proposed features can be used to perform spike sorting with good accuracy.

III. SYSTEM IMPLEMENTATION

This CT-based spike sorting scheme has been implemented in a typical $0.18 \mu\text{m}$ CMOS technology using the multi-step method described previously. This method avoids needing to use a timer to extract the dt feature and requires less subtraction.

A. Top-level Architecture

The top-level system diagram is illustrated in Fig. 6 showing the four main blocks. Front-end amplification and filtering uses similar topologies to those described in [14], however with lower tuned corner frequencies. The ADC design is based on [13], with LSB steps of ± 1 , ± 2 , ± 4 , and ± 8 . The delay of ADC conversion is tuned via DAC1, which is a 4-bit binary-weighted current mirror to feedback the various step sizes. The event triggers for the different step increments are fed into a set of counters via a MUX controlled by the configuration settings. The accumulated values are then stored and compared to template memory, which contain the feature coefficients (pre-determined through training). The spike detection and sorting engine have been implemented using standard digital cells. An extra timer providing a 2 ms delay is implemented using a three-stage current starved delay line. This is triggered by the positive threshold crossing and used to define the spike window thus also to avoid erroneous accumulation by spike aliasing. The logical flow of the detection and sorting algorithm is illustrated in Fig. 7.

B. Offline Training

Before online spike sorting is possible, a training phase is required to determine the classification parameters. This is

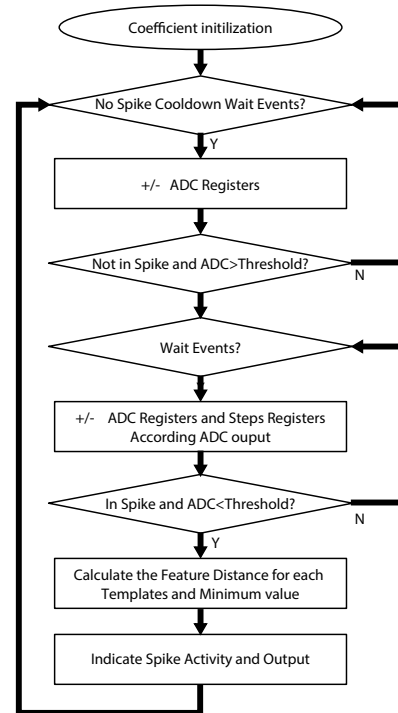


Fig. 7. Flow chart of the proposed algorithm for spike sorting

achieved offline using a conventional (computationally demanding) spike sorting algorithm to establish the clusters. The delay value used in the ADC can be configured during this process to optimise the feature spread, together with the features that provide the maximum separation between spike classes. The threshold values, selected features and classification boundaries are then uploaded into the in-channel memory.

C. Online Classification

After training, the system can achieve online classification using a basic asynchronous state machine. On spike detection, the event driven CT ADC generates the different pulse streams (for the selected increment/decrement values). These are then accumulated using the three counters until the delay element signifies the end of spike. This then triggers the feature

distance (FD) computation that provides a measure of how closely the counted feature sets match the training data. The FD is given by:

$$FD = \frac{\Delta N_1}{2Coef_1} + \frac{\Delta N_2}{2Coef_2} + \frac{\Delta N_3}{2Coef_3} + \frac{\Delta PEAK}{2Coef_4} \quad (2)$$

where $\Delta N_1, \Delta N_2, \Delta N_3$ are the differences between the counted steps (selected by the MUX) and the template memory, and $Coef_1, Coef_2, Coef_3, Coef_4$ are the corresponding coefficients. The spike is then classified according to the minimum FD value (i.e. comparing between the currently observed spike and the different classes in memory for that channel), and generating an address event output [15]. The unique feature of this signal path is the digital logic performing the spike processing is entirely event driven, requiring no clock whatsoever. The main source of static current consumption is therefore the self-biased three-stage comparator required for the CT ADC, which requires a $1.7 \mu A$ bias (described in [13]). The mismatch and linearity of DAC1 can be ignored. Excluding the comparator, a total current of 116 nA current consumption is measured through simulation.

IV. PERFORMANCE EVALUATION

To evaluate the proposed system, a behavioural model has been constructed in the Mathworks Simulink® environment with delay parameters and analogue filtering characteristics using the model described in [14]. Two data sets from [14] were used to test the system, and the spike sorted results were compared with Template Matching(TM) and PCA methods using the same data. It should be noted that for CT sorting, the coefficients were selected through training (for each individual data set), whereas for TM, the known template waveforms of the synthetic data were used. For PCA, 120 points of sampled data were used as input and k-mean iteration was set to 50. The sorting accuracy was defined as described in [14]:

$$P_d = \left(1 - \frac{N_e}{N_{su}}\right) \quad (3)$$

where N_e is the total number of missed spikes and false positives, and N_{su} is the number of spikes. The results are shown in Table I showing the classification accuracy of each method for each dataset. It can be observed that for the high SNR data, the proposed system achieved a sorting accuracy equivalent to the other methods, however, this becomes degraded when a reduced SNR. This is because any noise can incorrectly trigger a level crossing thus registering an increased or decreased number of adjacent (genuine) events. In the case of a low SNR, this erroneous triggering is increased, and thus can significantly distort the total amount of steps counted. Varying the internal delay of the ADC can in part suppress this effect, however this also reduces the response time of the trigger. An uneven step configuration could improve the performance with higher noise-immunity, and will be investigated in future work.

V. CONCLUSION

This paper has proposed a new CT-based spike sorting method using entirely event-driven processing. By using fixed, or a variable-step CT-ADC, it is shown how derivative features

TABLE I. SPIKE SORTING PERFORMANCE OF DIFFERENT SORTING METHODS [14]

Dataset	SNR	Average Sorting Accuracy %		
		CT	TM	PCA
D1	High	95.2	99.59	99.47
	Med.	98.7	99.54	99.43
	Low	72.0	98.49	97.77
D2	High	97.1	99.59	99.56
	Med.	89.8	99.05	99.15
	Low	73.2	97.07	97.48

can be easily obtained and combined with extrema, for online, on-node, efficient spike sorting. By using offline clustering to train the classification parameters it is described how a truly lightweight online classification can be achieved. A hardware implementation is a typical $0.18 \mu m$ CMOS technology is described. Simulation results based on behavioural model and synthetic neural data demonstrate the comparable sorting performance compared to other hardware implementable methods such as template matching.

REFERENCES

- [1] R. 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.
- [2] A.-T. Avestruz, W. Santa, D. Carlson, R. Jensen, S. Stanslaski, A. Helfenstine, and T. Denison, "A 5 w/channel spectral analysis ic for chronic bidirectional brain-machine interfaces," *IEEE JSSC*, vol. 43, no. 12, pp. 3006–3024, 2008.
- [3] D. Han, Y. Zheng, R. Rajkumar, G. S. Dawe, and M. Je, "A 0.45 v 100-channel neural-recording ic with sub-/channel consumption in 0.18 cmos," *IEEE Trans. BioCAS*, vol. 7, no. 6, pp. 735–746, 2013.
- [4] A. Eftekhar, S. E. Paraskevopoulou, and T. G. Constantinou, "Towards a next generation neural interface: Optimizing power, bandwidth and data quality," *Proc. IEEE BioCAS Conf.*, pp. 122–125, 2010.
- [5] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural comput.*, vol. 16, no. 8, pp. 1661–1687, 2004.
- [6] S. N. Kadir, D. F. Goodman, and K. D. Harris, "High-dimensional cluster analysis with the masked em algorithm," *Neural comput.*, 2014.
- [7] S. E. Paraskevopoulou, D. Wu, A. Eftekhar, and T. G. Constantinou, "Hierarchical adaptive means (ham) clustering for hardware-efficient, unsupervised and real-time spike sorting," *J. Neuroscience Methods*, vol. 235, pp. 145–156, 2014.
- [8] Y. Yang, C. S. Boling, and A. J. Mason, "Power-area efficient vlsi implementation of decision tree based spike classification for neural recording implants," *Proc. IEEE BioCAS Conf.*, pp. 380–383, 2014.
- [9] Y. Tsvividis, "Event-driven data acquisition and digital signal processing: a tutorial," *IEEE T-CAS II*, vol. 57, no. 8, pp. 577–581, 2010.
- [10] Y. Li, D. Zhao, W. Serdijn *et al.*, "A sub-microwatt asynchronous level-crossing adc for biomedical applications," *IEEE Trans. BioCAS*, vol. 7, no. 2, pp. 149–157, 2013.
- [11] X. Zhang and Y. Lian, "A 300-mv 220-nw event-driven adc with real-time qrs detection for wearable ecg sensors," *IEEE Trans. BioCAS*, vol. 8, no. 6, pp. 834–843, 2014.
- [12] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi, A. Eftekhar, and T. G. Constantinou, "Feature extraction using first and second derivative extrema (fsde) for real-time and hardware-efficient spike sorting," *J. of Neuroscience Methods*, vol. 215, no. 1, pp. 29–37, 2013.
- [13] C. Weltin-Wu and Y. Tsvividis, "An event-driven clockless level-crossing adc with signal-dependent adaptive resolution," *IEEE JSSC*, vol. 48, no. 9, pp. 2180–2190, 2013.
- [14] D. Y. Barsakcioglu, Y. Liu, P. Bhunjun, J. Navajas, A. Eftekhar, A. Jackson, R. Quian Quiroga, and T. G. Constantinou, "An analogue front-end model for developing neural spike sorting systems," *IEEE Trans. BioCAS*, vol. 8, no. 2, pp. 216–227, 2014.
- [15] R. Douglas, M. Mahowald, and C. Mead, "Neuromorphic analogue vlsi," *Ann. Rev. Neuroscience*, vol. 18, pp. 255–281, 1995.