Accurate, Very Low Computational Complexity Spike Sorting Using Unsupervised Matched Subspace Learning

Majid Zamani, *Member, IEEE*, Jure Sokolić, Dai Jiang, *Member, IEEE*, Francesco Renna, *Member, IEEE*,
Miguel R.D. Rodrigues, *Senior Member, IEEE*, and Andreas Demosthenous, *Fellow, IEEE*

Abstract—**This paper presents an adaptable dictionary-based feature extraction approach for spike sorting offering high accuracy and low computational complexity for implantable applications. It extracts and learns identifiable features from evolving subspaces through matched unsupervised subspace filtering. To provide compatibility with the strict constraints in implantable devices such as the chip area and power budget, the dictionary contains arrays of $\{-1, 0 \text{ and } 1\}$ and the algorithm need only process addition and subtraction operations. Three types of such dictionary were considered. To quantify and compare the performance of the resulting three feature extractors with existing systems, a neural signal simulator based on several different libraries was developed. For noise levels σ_N between 0.05 and 0.3 and groups of 3 to 6 clusters, all three feature extractors provide robust high performance with average classification errors of less than 8% over five iterations, each consisting of 100 generated data segments. To our knowledge, the proposed adaptive feature extractors are the first able to classify reliably 6 clusters for implantable applications. An ASIC implementation of the best performing dictionary-based feature extractor was synthesized in a 65-nm CMOS process. It occupies an area of 0.09 mm^2 and dissipates up to about 10.48 μW from a 1 V supply voltage, when operating with 8-bit resolution at 30 kHz operating frequency.**

Index Terms—**Complexity optimization, digital ASIC, feature extraction, implantable devices, high performance classification, spike sorting, subspace tracking, unsupervised learning.**

I.INTRODUCTION

ADVANCES in microtechnology have enabled precise neural interaction monitoring using implantable microelectrode arrays [1]. The stimulation of neurons and information derived from neuron action potentials or spikes are important for the development of neural interfaces [2]. Such implantable devices can decipher neural signals and stimulate a particular pathway for biological functionality regularization or reverse disease progression by artificially modulating irregular or faulty electrical impulses for numerous applications [3]-[7]. Recorded neural signals using microelectrode arrays comprise activity from 5 to 10 neurons (multi-unit activity) [8]. Performance efficiency is improved when the activity of individual neurons (single-unit activity) can be distinguished. The process of identifying the activity of individual neurons, *spike sorting*, consists of four major steps: spike detection and alignment, feature extraction, dimensionality reduction and classification [9]. There are two groups of spike sorting algorithms. In the first group are: principle component analysis (PCA) [10], graph Laplacian features (GLF) [11], linear discriminant analysis (LDA) [12] and discrete wavelet transform (DWT) [13]. These raise several issues when implantable device restrictions, such as thermal dissipation [14], limited implant size and battery capacity (lifetime) are considered. Although learning can be embedded in these algorithms [10]-[13] to enhance sorting performance, they are complex, and embedding learning into them has significant additional implementation cost. The second group minimizes hardware complexity cost by eliminating the multiplications (or divisions). This group includes sub-band selective discrete derivatives (DDs) [9]-[15], zero crossing features (ZCFs) [16], first and second derivative (FDVSDV) spike features [17] and template matching (TM) [18]. The algorithms in this category mostly avoid embedding learning in order to reduce implementation cost but they experience performance drop when recording channel variations occur.

This paper introduces a new approach using dictionary-based feature extraction with learning. It combines high accuracy spike sorting with very low computational complexity. Feature extraction is improved in an energy efficient manner by identifying the most informative, yet low-dimensional structures, from the high-dimensional input neural data. Fig. 1(a) shows the main units of the dictionary-based feature extraction: i) a dictionary $\varphi(k)$ which stores the evolving subspace and only contains arrays of $\{-1, 0 \text{ and } 1\}$; and ii) customized matched subspace filtering for $\varphi(k)$. The dictionary $\varphi(k)$ is the basis of subspace tracking, and the proposed subspace learning embeds the optimal signatures of the informative subspace into $\varphi(k)$. The spike sorting adjusts to the varying characteristics of the input neural signals (e.g. noise variations, number of active spike waveforms, electrode drift, firing rate of the neurons) by unsupervised adaptive

Manuscript received November 15, 2019; revised December 27, 2019. This work was supported in part by the European Commision under H2020-EU.1.2.2. – FET Proactive (Agreement ID: 824071), and the Medical Research Council UK under Grant MR/R011648/1.

M. Zamani, D. Jiang, M.R.D. Rodrigues and A. Demosthenous are with the Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, UK. (e-mail: m.zamani@ucl.ac.uk, d.jiang@ucl.ac.uk, m.rodrigues@ucl.ac.uk, a.demosthenous@ucl.ac.uk).

J. Sokolić was with the Department of Electronic and Electrical Engineering, University College London. He is now with Metronik d.o.o., Slovenia. (e-mail: jure.sokolic@gmail.com).

F. Renna is with the Instituto de Telecomunicações, Faculdade de Ciências da Universidade do Porto, Rua do Campo Alegre 1021/1055, 4169-007 Porto, Portugal (e-mail: frarenna@dcc.fc.up.pt).

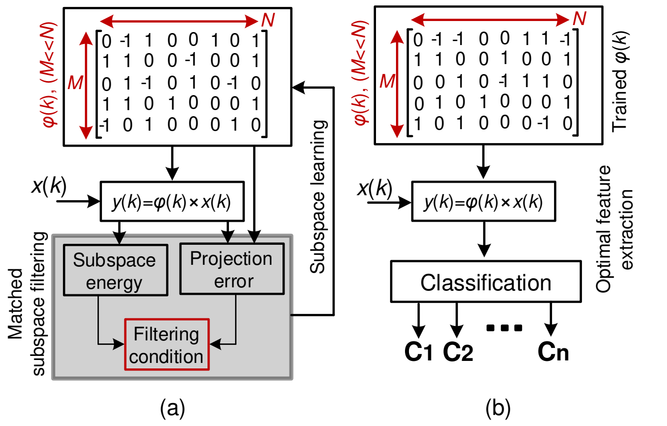

Fig. 1. Dictionary-based feature extraction in (a) learning phase and (b) optimal feature extraction. In the learning phase, the matched subspace filtering reconfigures $\varphi(k)$ to emulate ideal existing subspace. Once the optimal $\varphi(k)$ is constructed, the features of aligned spike waveforms $x(k)$ are obtained for classification $(\mathrm{C}1, \mathrm{C}2 \dots \mathrm{Cn})$. The constituent arrays of the dictionaries are $\{-1, 0 \text{ and } 1\}$ resulting in efficient hardware realization. N and M are the length of original and feature vectors respectively ($M << N$).

learning of subspaces providing reliable information for classification.

The rest of paper is organized as follows. Section II presents the novel dictionary-based feature extraction. It outlines the dictionaries considered, the subspace learning algorithm with embedded learning, the neural simulator developed for scrutinizing the operation of the feature extraction algorithms, and methods for comparison. Section III presents the spike sorting results. The overall complexity for all feature extractors is compared. Hardware implementation is also considered. Conclusions are drawn in Section IV.

II.Feature Extraction Core Design

A. Proposed Dictionary-Based Feature Extractor

1) Types of Dictionary $\varphi(k)$:

The three types of dictionary $\varphi(k)$ considered are formed by arrays of $\{-1, 0 \text{ and } 1\}$ as shown in Fig. 1. For a matrix that only contains $\{-1, 0 \text{ and } 1\}$, the Hadamard matrix [19] is the most popular. Its simple structure is an eligible candidate to store optimal weights for applying the subspace changes in the neural signal over time. The dictionary constructed based on the Hadamard matrix is referred to as $\varphi_{H_h(k)}$. The equiangular tight frame (ETF) is also a viable option to generate a dictionary containing $\{-1, 0 \text{ and } 1\}$. In this paper the construction procedures proposed by Fickus et al. [20] are utilized for generating the ETF dictionary ($\varphi_{ETF(k)}$). The third dictionary is generated based on the random Bernoulli matrix (RBM) [21]. As an example, construction of the RBM dictionary ($\varphi_{Bern(k)}$) has the following steps:

1) Preset a possibility p_{os}.
2) Initialize $\varphi_{Bern(k)}$ of $N \times 2N$ zero matrix.
3) Loop for $i = 1$ to N and $j = 1$ to $2N$, generate a random number between 0 and 1, if this number is larger than p_{os}, then assign 1 to $\varphi_{Bern(k)}$, otherwise remain 0.

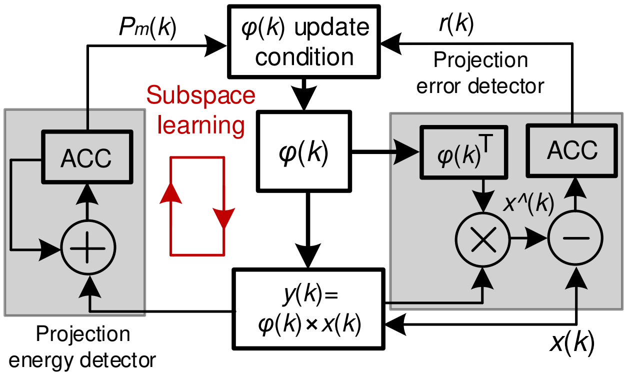

Fig. 2. Subspace filtering block diagram. The filtering is initiated by synthesizing the subspace information from the $P_m(k)$ and $r(k)$ detectors. The extracted information reveals the alignment of $\varphi(k)$ to the discriminative subspace. Subspace learning is performed by updating $\varphi(k)$ over multiple iterations.

The size of the initially generated $\varphi_{Bern(k)}$ is set to 54 (N = spike length) × 108. This procedure also applies to $\varphi_{H_h(k)}$ and $\varphi_{ETF(k)}$ construction. These three dictionaries offer unique low-cost attributes in subspace learning which are utilized in unsupervised subspace learning.

2) Matched Unsupervised Subspace Learning

Investigation of a subspace offering highly discriminative features is predicated on the fact that the high-dimensional neural data lie in a low dimensional space. This key attribute allows efficient analysis of data and tracking the low-dimensional subspace evolving over time. The input neural signal $X(i)$ ($X = [\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{n}] \in \mathbb{R}^n$) is defined as $X(i) = S(i) + n(i)$ where $\mathrm{S}(i)$ contains the signal information that lies in the low-dimensional linear subspace which evolves over time and $n(i)$ accounts for noise. $X(i)$ is fed to the detection and alignment unit which identifies the spike waveforms and aligns them into a temporal reference (e.g. peak alignment). Thus, for the k^{th} data segment, it is assumed that there are W aligned spike waveforms which are passed to the feature extraction. The interest in feature extraction is to provide discriminative projection of the aligned spike waveforms to a M-dimensional (M << N) feature space. A customized and invariant matched subspace filtering and learning scheme is proposed which provides unsupervised evaluation of subspace usefulness in the stream of aligned spike waveforms $x(k)$.

a) Subspace Filtering Core

Subspace filtering uses two detectors: the accumulated energy of the feature vectors $P_m(k)$ and the projection error (or residual error $r(k)$) as a threshold for optimal alignment of the dictionary matrix $\varphi(k)$ arrays $\{-1, 0 \text{ and } 1\}$ to the existing subspace as shown in Fig. 2. M columns of the initially generated dictionary (e.g. 54 × 108 in RBM) are chosen and transposed to form the feature extraction dictionary $\varphi(k)$ where M depicts the desired number of features. $P_m(k)$ and $r(k)$ quantify the likelihood of correlation between $\varphi(k)$ arrays $\{-1, 0 \text{ and } 1\}$ and the original subspace, so $\varphi(k)$ can be tuned by iterative unsupervised learning. Computation of $P_m(k)$ is based on the mean squared value of the projected spike waveforms $y(k) = \varphi(k) \times x(k)$ expressed as:

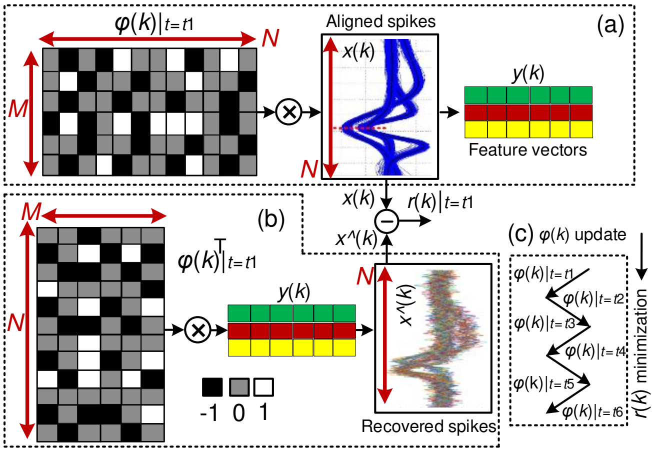

Fig. 3. (a) The projection path transforms aligned spikes to a lower dimension using $y(k) = \varphi(k) \times x(k)$. $\varphi(k)$ is the dictionary with $M \ll N$ in projection path and allows to evaluate subspace energy $P_m(k)$. (b) Data can be recovered in the reconstruction path $x^{\wedge}(k) = \varphi(k)^T y(i)$ which leads into projection error analysis $r(k) = x(k) - x^{\wedge}(k)$. The distorted and incomplete recovery is due to non-optimal representation of subspace by $\varphi(k)$ in reconstruction path ($N \ll M$). (c) Subspace learning iterations $\varphi(k)|_{t=t1}, \dots, \varphi(k)|_{t=t6}$ for embedding optimal signatures of the informative subspace into the initially generated dictionary $\varphi(k)$.

$$p_m(k) = \frac{1}{W} \sum_{i=(k-1)W+1}^{kW} (y_m(i))^2, m = 1, \dots, M \qquad (1)$$

where W is the number of feature vectors in the k^{th} data segment and M is the number of samples per feature vector. Since the generated dictionaries consist of $\{-1, 0 \text{ and } 1\}$ the squared and normalization factors can be neglected which avoids multiplications in energy calculations. Alternatively, $P_m(k)$ can be derived by accumulating the absolute value of features. The second detector used in original subspace estimation is $r(k)$ (see Fig. 2). The idea behind $r(k)$ stems from the fact that the projection of spike waveforms using a non-optimal $\varphi(k)$ always produces some error as shown in Fig. 3. The difference between the original vectors $x(k)$ and the recovered signal $x^{\wedge}(k)$ represents the residual error $r(k)$. The recovered signal $x^{\wedge}(k)$ is calculated by multiplying the transpose of the feature extraction matrix $\varphi(k)^T$ with the projected feature vectors $x^{\wedge}(k) = \varphi(k)^T y(i)$. The residue $r(k)$ for a window is derived by accumulating the absolute value of differences between original $x(i)$ and reconstructed $\varphi(k)^T y(i)$ waveforms:

$$r(k) = \sum_{i=(k-1)W+1}^{kW} \left|\left(x(i) - \varphi(k)^T y(i)\right)\right| \qquad (2)$$

Eq. (2) calculates the projection error of the k^{th} segment based on W feature vectors each containing M samples. It also should be noted that Eq. (2) does not require normalization $(1/W)$ due to removal of this factor in $p_m(k)$ (Eq. (1)). The transpose of the dictionary matrix $\varphi(k)^T$ also contains $\{-1, 0 \text{ and } 1\}$; hence, computing of the recovered signal $x^{\wedge}(k)$ is the sum of $\varphi(k)^T y(i)$ elements.

b) Subspace Learning

Algorithm 1. Dictionary-based feature extractor.

```
1. Initialize the dictionary $\varphi_{H_h(k)}, \varphi_{ETF(k)}$ or $\varphi_{Bern(k)} = [\varphi_1^T, \dots, \varphi_D^T]$.
2. Extract $M$ columns of vectors from dictionary $[\varphi_1^T, \dots, \varphi_D^T]$ and
   transpose them to form feature extraction matrix $\varphi(k)$ based on
   desired number of features ($M$).
3. For every segment, $y(k) = \varphi(k) \times x(k)$.
4. Start from first segment, after the computation of feature extraction,
   the feature extraction matrix $\varphi(k+1)$ for the next window is updated
   based on the following calculations:
   5. Absolute value accumulation of features:
      $p_m(k) = \sum_{i=(k-1)W+1}^{kW} |y_m(i)|, \quad m = 1, \dots, M.$
   6. Residue:
      $r(k) = \sum_{i=(k-1)W+1}^{kW} |(x(i) - \varphi(k)^T y(i))|.$
   7. Find the index of the "worst" feature vector:
      $m^* = \arg\min_m p_m(k).$
   8. If $r(k) < p_{m^*}(k)$ keep previous matrix $\varphi(k)$.
   9. If $r(k) \geq p_{m^*}(k)$ replace the $m^*$th row of $\varphi(k)$ by one of the
      column vectors in initially generated dictionary $[\varphi_1^T, \dots, \varphi_D^T]$, the
      calculation process is as follows:
         10. Extract and form a matrix $D'$ from the non-utilized column
             vectors of $[\varphi_1^T, \dots, \varphi_D^T]$ that contains $N - M'$ vectors.
         11. Compute and find the index $l = \arg\max_l |D'(k) \times x(k)|$.
         12. Update the $m^*$th row of $\varphi(k)$ by $D'_l$.
13. Back to step 3, continue the feature extraction process to the next
    window until the end of data.
```

The subspace learning utilizes an unsupervised and iterative process based on comparisons between $P_m(k)$ and $r(k)$. The iterations minimise the residual error $r(k)$ and embed the most discriminative subspace in the constructed dictionaries $\varphi(k)$ that are the basis of the initial subspace. In Eq. (2), the input vectors $x(k)$ are constantly changing over time so there is no control over this term, and it is not dependent on the feature extraction matrix $\varphi_{(k)}$. However, the second term in Eq. (2), $\varphi_{(k)}{}^T \underbrace{\left(\varphi_{(k)} x(k)\right)}_{y(k)}$ explicitly implies that to minimize the residual $r(k)$, the projection likelihood needs to be maximised [22]-[23]. The learning is initiated by projection of the first spike waveform by $y(k) = \varphi(k) \times x(k)$. It is assumed that the optimal features are not obtained after $\varphi(k)$ construction. Once the projection is carried out, the detectors in the filtering unit are activated to quantify the deviation of the initial $\varphi(k)$ to the ideal subspace. According to the $r(k)$ minimisation principle, two conditions for $\varphi(k)$ reconfiguration are derived: i) If $r(k) < P_m(k)$ is satisfied, the feature extraction process retains much of the energy in the original subspace which makes optimal separation of the spike waveforms possible. The residue of the projection is calculated and set as the threshold to realize the $\varphi(k)$ updating scheme. ii) $r(k) \geq P_m(k)$ indicates misalignment of $\varphi(k)$ arrays to the informative subspace. Since the main aim of feature extraction is to maximize the projection energy whilst minimizing the residual, the row in $\varphi(k)$ representing the least feature energy (m^*) after projection is replaced using a row closer to the residue direction. The replacement process begins by configuring the not-utilized columns matrix (D') in the initially generated dictionary (e.g. 54×108 in RBM) and

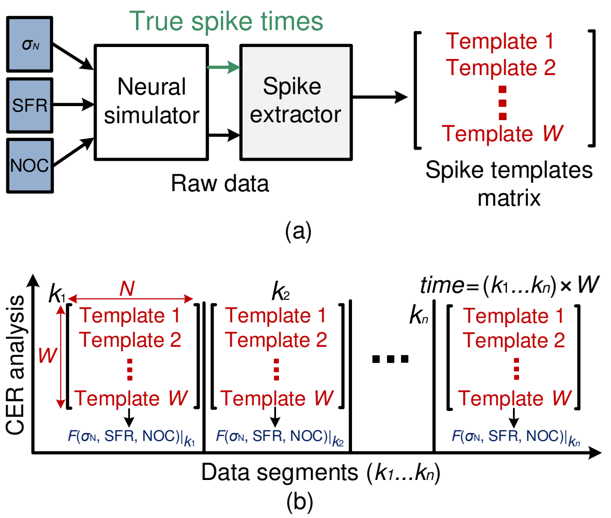

Fig. 4. (a) Signal-processing chain used to evaluate feature extraction algorithms classification performance. Spike extractor uses true spike times to identify the spike waveforms and it is considered as an ideal detector. The output of spike extractor is spike templates matrix with W templates. (b) The spike template matrix changes over the data segments ($K_1 \ldots K_n$). The matrix generation is the function of input factors to the simulator (e.g. $F(\sigma_N, \mathrm{SFR}, \mathrm{NOC})|_{k1}$) where SFR and NOC are spike firing rate a and number of clusters respectively.

exploring the column offering the highest projection energy ($l = \arg\max_{l} |D'\,(k) \times x(k)|$). The identified column D'_l is used to update $\varphi(k)$ and has good alignment to the optimal subspace. D'_l has unique $\{-1, 0 \text{ and } 1\}$ distribution (not a repetition of other columns). One row of $\varphi(k)$ is updated during each iteration, and $\varphi(k)$ converges to the informative subspace over learning steps $\{\text{e.g. } \varphi(k)|_{t=t_1}, \ldots, \varphi(k)|_{t=t_6}\}$. Algorithm 1 details the steps for adaptive dictionary-based feature extraction using unsupervised learning of discriminative subspace. The feature extraction algorithm reconfigures the $\varphi(k)$ intermittently to maintain optimal classification performance.

c) Optimal Feature Extraction

Having identified the optimal dictionary, the streaming phase in which the aligned spike waveforms are multiplied with $\varphi(k)$ is performed. Since the $\varphi(k)$ contains $\{-1, 0 \text{ and } 1\}$, the multiplications can be rewritten as a series of amplitude accumulation:

$$y(k) = \sum_{k=1}^{N} \varphi_{i,1} X_{1,j} + \cdots + \varphi_{i,k} X_{k,j},\ \varphi_{i\ldots k} = \{1,0,-1\}. \quad (3)$$

The spikes are constantly transformed to optimal feature vectors (FVs) and assigned to one of the clusters according to the identified cluster centroid by the classification unit.

B. Other Feature Extraction Methods for Comparison

1) Adaptive Discrete Derivatives (ADDs) [15]:

ADDs are computed by calculating the slope at each sample point over a number of different time scales:

$$\mathrm{ADDs} = amp[s(n) - s(n-\delta)|_{\delta=1\ldots7}] \quad (4)$$

where $amp = 1$ is the amplitude of the decomposition window, s is the spike waveform, n is the sample point, and δ is the scaling factor (time delay). Adjustment of the scaling factors ($\delta 1$, $\delta 2$, $\delta 3$) are based on three frequency sub-bands from $\delta = 1$ to $\delta = 7$ corresponding to the most deviated features (non-Gaussian features) for unsupervised clustering.

2) Updated Graph Laplacian Features (uGLF) [11]:

GLF is a linear feature extraction technique that simultaneously minimises the graph Laplacian and maximises variance [11]. The important attribute of GLF is that the points which are close to each other in high dimensional space remain close to each other after transformation to low dimensional space to ensure the clusters are compact and separable. In uGLF, a weighted graph representing the projection matrix is constructed [11] and updated according to a newly generated data batch in the neural simulator.

3) Updated PCA (uPCA) [10]:

In uPCA the projection matrix is updated according to the subspace changes. The projection matrix $\varphi_{(k)}$ is updated at every segment k using the standard PCA algorithm.

4) Rotated PCA (rPCA):

In rPCA, the feature extraction matrix is initialized using the standard PCA algorithm. The feature extraction matrix $\varphi(k)$ changes over time, adapting itself to the evolving subspace and projecting the signal $x(k)$ in a lower dimensional space while preserving the main characteristics of $x(k)$. The outlined subspace learning in Algorithm 1 is adopted for rPCA. rPCA has less computational complexity compared with the standard PCA by eliminating computations of singular value decomposition; however, it still requires multiplications and divisions for updating the feature extraction matrix.

E. Neural Data Simulator

A library-based neural simulator was developed to emulate extracellular recordings with realistic background noise and a known “ground truth” for evaluation of algorithms for spike waveform classification. To generate neural spikes, a database of synthetic spike waveforms containing 300 different average spike shapes was constructed. The spike shapes were extracted from the peripheral median nerve in pig (obtained with a multi-electrode cuff in vivo) [9], the neocortex and basal ganglia [24], from the right and left hippocampus and either from the right or left amygdala [25], [26]. Fig. 4 shows the procedure used for neural data generation and examination of the feature extraction algorithms. The neural spike waveforms are randomly selected and placed in a data stream from the spike library using a defined number of clusters (NOC) and spike firing rates (SFR). The data stream is then corrupted by additive noise with varying standard deviations (e.g. $\sigma_N = 0.01$) at random times. The extracted spike templates (Template $1 \ldots$Template W) form a matrix; it is the function of the simulator setting over each segment (e.g. $F(\sigma_N, \mathrm{SFR}, \mathrm{NOC})|_{k1}$) where k_1 shows the first data segment. The changes over each segment embed four elements (σ_N, SFR, NOC and $time = (k_1 \ldots k_n) \times W$) to emulate a real recording channel and changing subspace for in-depth analysis of the feature extraction methods at different conditions.

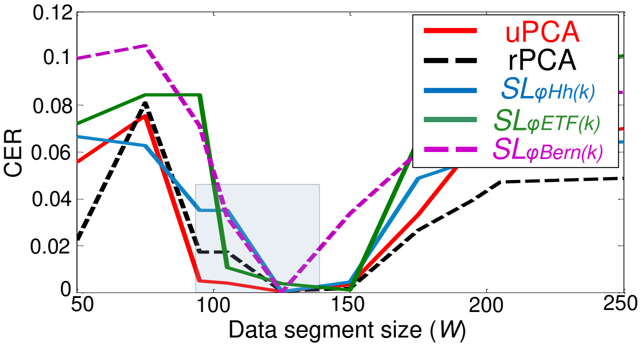

Fig. 5. Classification error (CER) versus data segment size W. The 'K-means' function in Matlab with the number of iterations set to 10 for near-optimum CER analysis was used. The parameters used in simulation were $\sigma_{\mathrm{N}} = 0.15$, NOC = 5 and M = 5.

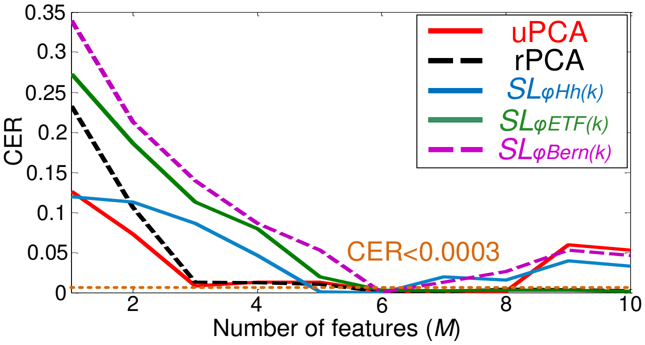

Fig. 6. Classification error (CER) versus number of features M. The parameters in simulation were: $\sigma_{\mathrm{N}} = 0.15$, NOC=5, $k = (1 \ldots 100)$, W = 125 and spike firing rate (SFR) = 40; the typical value for an active neuron [27]. M = 2 is chosen for uGLF based on [11].

III. Results and Discussion

In this section it is demonstrated that the accuracy of the feature extractors $SL\varphi_{H_h(k)}, SL\varphi_{ETF(k)}$ and $SL\varphi_{Bern(k)}$ is competitive with ADDs, uPCA, rPCA and uGLF but with significantly reduced computational complexity. An ASIC of $SL\varphi_{H_h(k)}$ is implemented for proof of concept.

A. Optimal Selection of Adaptive Parameters

The effect of data segment size (W) and number of features (M) on the feature extractors examined. W has the highest priority since the updating scheme of the adaptive methods is realized using segments of data. Different combinations of k and W are used to generate various data streams ($k \times W$). The variables are set to $k = 100$, $50 < W < 250$ and $M = 5$. Fig. 5 shows the classification error (CER) versus W where:

$$\mathrm{CER} = (1 - \mathrm{CA}_{\mathrm{CC}}) \tag{5}$$

and $\mathrm{CA}_{\mathrm{CC}}$ is the classification accuracy (i.e. number of truly assigned feature vectors over the total number of feature vectors). CER decreases in the range $95 < W < 125$ which establishes a robust margin for defining W. For $W > 125$ the CER begins to increase because the subspace update is not aligned with signal changes defined in the neural signal simulator. The worst case is when $\varphi_{(k)}$ is not updated according to the changes embedded in the incoming segments.

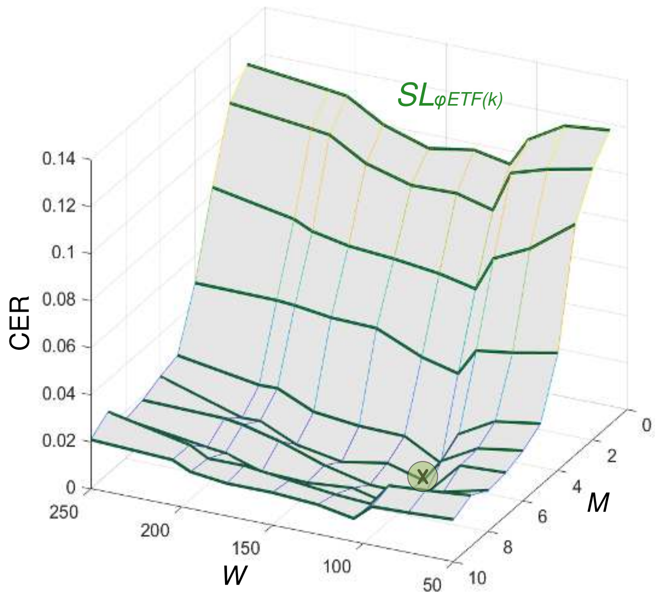

Fig. 7. 3D plot of hyperparameters M and W versus classification error (CER) for $SL\varphi_{ETF(k)}$. The optimal parameters are shown at cross (X) corresponding to M = 6, W = 105 and CER = 0.000144.

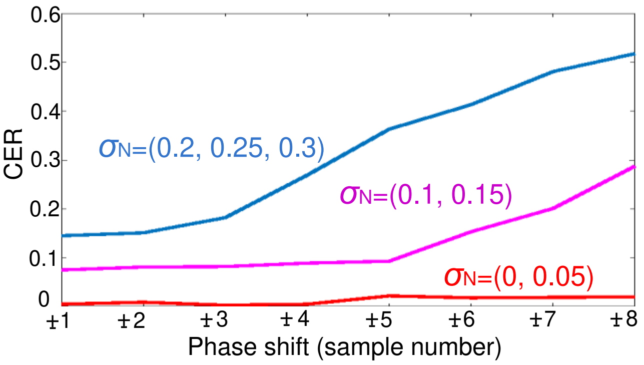

Fig. 8. Classification error (CER) versus alignment phase shift for different groups of noise level σ_{N}. CER is based on NOC = 5 and five iterations.

This is an interesting point since the subspace changes in the neural signals are too slow, which means the dictionary updating scheme is less sensitive to k and W. In Fig. 6 the CER of the feature extraction methods are compared as the number of features is varied ($1 < M < 10$) in neural data batches (see Fig. 4(b)). It can be seen that the reduction of M has an adverse effect on CER. This is because decreasing the value of M results in reducing the projection probability of the original features to a lower subspace. For example, the projection residual is determined in $\varphi_{(k)}$ by the number of dictionary rows M. For $M = 2$, $\varphi_{(k)}$ consists of two rows so that the probability of the residual being greater than projection is lower, resulting in the update of the feature extraction matrix. The optimal value of M for separating spike waveforms results in almost zero CER (<0.0003) when $M = 6$. A three-dimensional (3D) surface plot of M and W versus CER for $SL\varphi_{ETF(k)}$ is shown in Fig. 7. The optimum choice for M and W is at location X.

B) Classification Error of Feature Extractors

TABLE I
Classification Error Comparison of Feature Extraction Methods.

Dataset	Noise	Classification error (CER)								
		Complex methods			Proposed methods			High error methods		
		uPCA	rPCA	uGLF	$SL\varphi_{H_h(k)}$	$SL\varphi_{ETF(k)}$	$SL\varphi_{Bern(k)}$	ADDs	ZCF ***	FDVSDV
(NOC = 3)*	0.05	0.0005 **	0.0021	0.0020	0.0031	0.0011	0.0013	0.0169	0.0681	0.0932
	0.10	0.0091	0.0068	0.0115	0.0088	0.0149	0.0138	0.0702	0.1112	0.1671
	0.15	0.002	0.0094	0.0122	0.0194	0.0254	0.0204	0.1377	0.224	0.2647
	0.20	0.0603	0.0634	0.0712	0.0534	0.0741	0.0968	0.2307	0.3331	0.3931
(NOC = 4)	0.05	0.0010	0.0033	0.0026	0.0063	0.0024	0.0181	0.0246	0.1581	0.1915
	0.10	0.0019	0.0137	0.0126	0.0167	0.0218	0.0285	0.0465	0.2411	0.2973
	0.15	0.0217	0.0386	0.0262	0.0242	0.0434	0.0928	0.1906	0.3345	0.3947
	0.20	0.0619	0.0781	0.0651	0.0881	0.1001	0.1607	0.2413	0.4131	0.4831
(NOC = 5)	0.05	0.0016	0.0047	0.0056	0.0081	0.0144	0.0137	0.0242	0.2793	0.3146
	0.10	0.0074	0.0130	0.0304	0.0190	0.0414	0.0537	0.1444	0.3873	0.3923
	0.15	0.0358	0.0461	0.0401	0.0661	0.0722	0.1015	0.2271	0.4677	0.5541
	0.20	0.0818	0.0871	0.0832	0.0943	0.1471	0.1793	0.2777	0.5841	0.6687
(NOC = 6)	0.05	0.0036	0.0068	0.0241	0.0098	0.0236	0.0293	0.0819	0.3418	0.3723
	0.10	0.0113	0.0273	0.0365	0.0451	0.0452	0.0911	0.1472	0.4143	0.4861
	0.15	0.0449	0.0657	0.0712	0.1012	0.0982	0.1331	0.2215	0.5313	0.6541
	0.20	0.0954	0.1084	0.1214	0.1384	0.1644	0.1957	0.3078	0.6441	0.7187
Overall CER		0.0275	0.0359	0.0385	**0.0439**	**0.0556**	**0.0769**	0.1494	0.3458	0.4029

* The mean spike waveforms of $\mathrm{NOC} = 3{,}4, 5$ and 6 are randomly chosen from data library.

** The reported CER at a specific dataset and noise level is the average of five runs over 100 data segments using the K-means classifier.

*** ZCF [16] and FDVSDV [17] are not adaptive feature extractors.

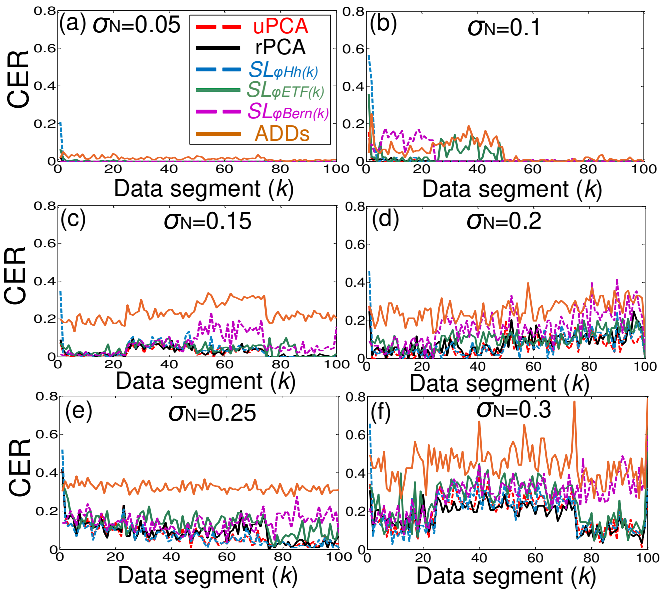

Fig. 9. Comparison of classification error (CER) of feature extractors versus noise level $\sigma_{\mathrm{N}} = (0.05, .., 0.3)$ for $\mathrm{NOC} = 4$. CER is the average of five runs across the data segments $k = (1 \dots 100)$. In each segment, the noise is generated and added to the randomly selected neural spike mean waveforms which cause CER variations across data segments.

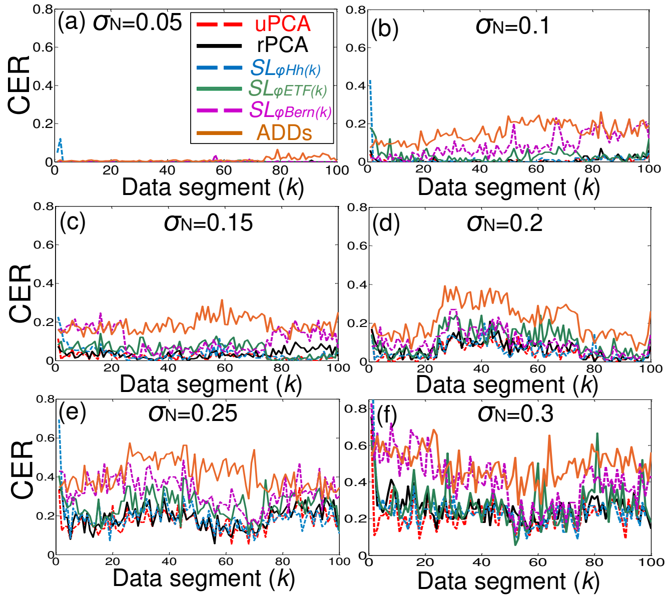

Fig. 10. Comparison of classification error (CER) of feature extractors versus noise level $\sigma_{\mathrm{N}} = (0.05, .., 0.3)$ for $\mathrm{NOC} = 6$. CER is the average of five runs across the data segments $k = (1 \dots 100)$. Six spike waveforms with different degrees of similarity are randomly chosen from the spike library in each segment and sent to the sorting chain.

The classification accuracy of the different feature extractors is compared as a function of noise level (σ_{N}), NOC and the similarity between the spike mean waveforms, using the K-means classifier [28] in Matlab and the neural data simulator in Section II-E. The spike waveforms were extracted and aligned at their peaks for feature extraction and classification evaluation. Typically, the classification performance can be compromised due to improper selection of the temporal reference in the alignment process. A test set-up investigated the feature extraction sensitivity to phase shift by adding offset in the alignment process.

The following limits can be identified over five runs at $\mathrm{NOC} = 5$ for $SL\varphi_{ETF(k)}$ as shown in Fig. 8:

1. For average CER on noise levels $\sigma_{\mathrm{N}} = (0,\ 0.05)$, there is at least ±8 samples phase shift robustness. The feature extractor shows extremely high robustness to the injected phase shift.
2. For average CER on noise levels $\sigma_N = (0.1,\ 0.15)$, there is a ±5 samples phase shift robustness. Feature extraction

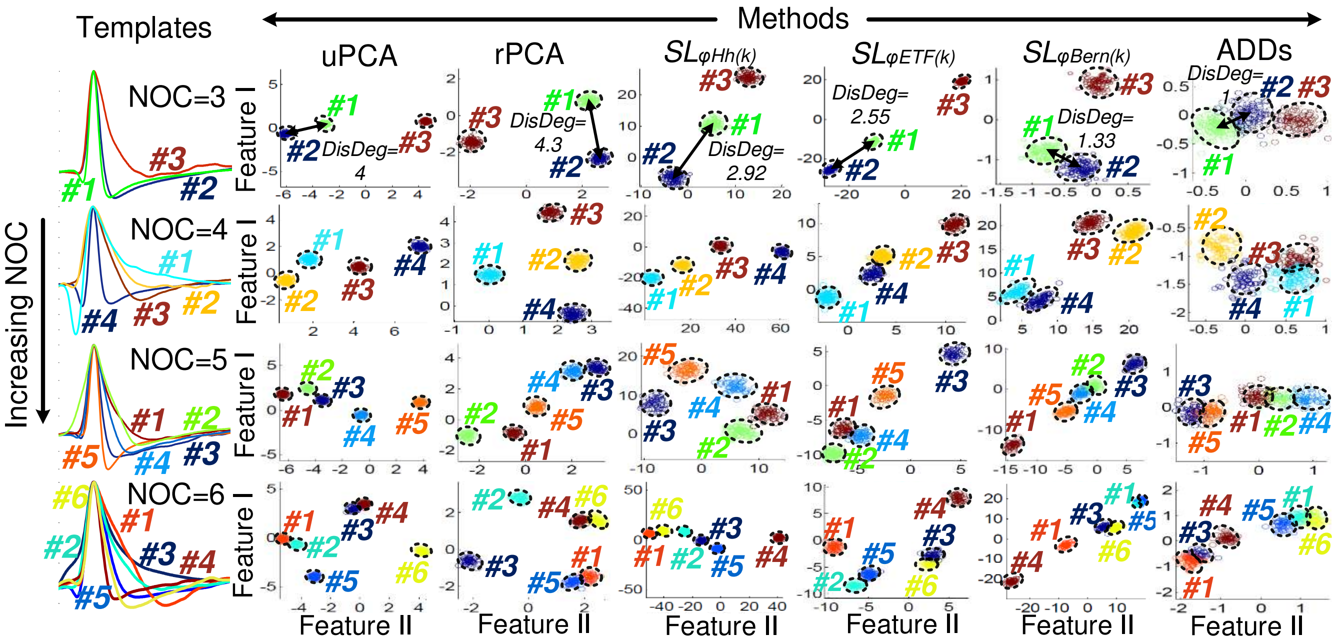

Fig. 11. 2D projection of clusters using two features with the most deviation from normality (Feature I and Feature II) for the selected spike mean waveforms shown in the first column. The test set-up in this figure includes three elements to emulate the real recording channel, $\mathrm{NOC} = (3, 4, 5 \text{ and } 6)$, similarity index between the mean waveforms and $\sigma_N = 0.15$. The other columns, from second to seventh, show 2D projection of the clusters using uPCA, rPCA, $SL\varphi_{H_h(k)}$, $SL\varphi_{ETF(k)}$, $SL\varphi_{Bern(k)}$ and ADDs. The projected clusters are color-coded and numbered according to the mean waveforms. To obtain optimal separation of feature vectors other parameters are set to: $N = 64$, $M = 6$ and spike firing rate (SFR = 40).

and consequently classification shows some sensitivity to the injected phase shift; however, it still performs well.

3. For average CER on noise levels $\sigma_N = (0.2, 0.25, 0.3)$, there is a ±2 samples phase shift robustness; the noisiest set-up which is still acceptable.

The alignment robustness can be explained by the fact that the proposed feature extraction is reconfigured according to the informative subspace. Thus, various alignment methods (e.g. peak alignment) can be used before dictionary-based feature extraction. The results in Table I show that dictionaries ($SL\varphi_{H_h(k)}$, $SL\varphi_{ETF(k)}$ and $SL\varphi_{Bern(k)}$) have an overall CER of less than 8% and produce almost the same total average CER compared with uPCA, rPCA and uGLF for different $\mathrm{NOC} = (3, 4, 5, 6)$, when noise is varied within the limits $\sigma_\mathrm{N} = (0.05, 0.1, 0.15 \text{ and } 0.2)$.

Fig. 9 and Fig. 10 show the CER variations of two different tests when the data stream specifications are set to $k = (1 \dots 100)$ and $W = 125$. Each segment (e.g. $k = 1$) considers random spike templates selection and noise distribution generation using the defined σ_N and NOC to emulate the recording channel variations over time $(time = (k_1 \dots k_n) \times W)$. In Fig. 9, $\mathrm{NOC} = 4$, $SL\varphi_{Bern(k)}$ and ADDs show higher noise and spike template similarity sensitivity compared with the other methods. For example, at $58 < k < 70$ for $\sigma_\mathrm{N} = 0.15$ in Fig. 9(c), $SL\varphi_{Bern(k)}$ is not capable of projecting the randomly selected spike waveforms into low dimensional feature space which results in higher CER. In Fig. 10, for NOC=6, the dictionary-based methods exhibit acceptable CER for the noise range $\sigma_\mathrm{N} = (0.05, 0.1, 0.15 \text{ and } 0.2)$. To our knowledge, the proposed dictionary-based feature extractors are the only low-cost methods that are capable of handling the wide range of variations generated in the data stream. The uGLF CER results are not plotted in Fig. 9 and Fig. 10 since it has almost similar CER behaviour to rPCA.

In addition, the results in Fig. 10 reveal that ADDs and $SL\varphi_{Bern(k)}$ demonstrate almost similar noise robustness when the noise is increased linearly over the range $\sigma_\mathrm{N} = (0.05, \dots, 0.3)$. In ADDs the decomposition scaling factors update scheme is performed at the beginning of the updated window (e.g. $W = (125, 225, 325, \dots)$). The CER plots of FDVSDV and ZCF versus $k = (1 \dots 100)$ (which are not plotted in Fig. 9 and Fig. 10) provide the most robust classification at $\mathrm{NOC} = (3,4)$ and $\sigma_\mathrm{N} = (0.05, 0.1)$. FDVSDV has higher noise sensitivity than ADDs because when the decomposition sub-bands are not selected in accordance with existing spike waveforms, the chosen distorted features are non-informative and their corresponding class cannot be simply identified.

To visually assess the cluster separation quality obtained using different feature extraction methods, the scatter plots of the two-dimensional (2D) features extracted from the generated data are shown in Fig. 11. The first column of Fig. 11 on the left shows the selected cluster means for 2D projection test. The main aim of this test is to examine in-depth the sorting performance when NOC is increased and the similarity indices (SIs [29]) between them are varied. To achieve this, after defining the NOC in the neural simulator, a range can be defined for the SIs and the simulator selects the spike mean waveforms from the spike library according to the defined range (e.g. $0.5 < SI < 0.85$). As the NOC increases, for $\mathrm{NOC} = (3, 4, 5, 6)$ and $\sigma_\mathrm{N} = 0.15$ the clusters originating from different neurons tend to mix into each other. However,

TABLE II
Comparison of Feature Extraction Methods.

Method	Additions	Multiplications	[d]Overall complexity	DR	[e]DF $= M/N$	CER
ADDs	[a,b]$kW\ (3\mathrm{N} - (\delta 1 + \delta 2 + \delta 3)_{\delta\|_{1..,7}}$	-	30372	Extrema	6/162	0.1494
ZCF	$kW(N)$	-	10800	-	2/54	0.3458
FDVSDV	$kW(2N-3)$	-	21000	Extrema	4/108	0.4029
Spike shape	-	-	-	-	54/54	0.3888
[c]DWT [13]	$kW(4N)$	$kW(8N-10)$	88720	KS test	10/54	0.078
rPCA	$W(N^2+2N+1)+(k-1)WN$	$W(N^2+N)+(k-1)3WN+WM$	956820	-	6/54	0.0359
uPCA	$kW(N^2+2N+1)$	$kW(N^2+N)$	6545000	-	6/54	0.0275
uGLF	$kW(5N^2+2N+1)$	$kW(N^2+N)+(k+1)N+10WM$	8893740	-	2/54	0.0385
$SL\varphi_{(k)}$	$k(NM \times p + N^2M \times p + MW)$	-	97525	-	6/54	<0.08

W = number of spikes per window; *N* = sample number per spike; *M* = sample number per feature vector.
a: $\delta 1 + \delta 2 + \delta 3$ represent the scaling factors of three decomposition lines which each is adaptively set to 1…7 over time.
b: The estimated complexity of frequency synthesizer in ADDs is based on differentiation and accumulation of the spike waveforms $\beta(k-1)(W)(N)$ for extraction of localized differences [30] and its first derivative of the accumulated waveform with the same length as the pike waveform (*N* – 1). β defines the number of spikes used for learning informative sub-bands in ADDs.
c: DWT (four-level Haar wavelet), Kolmogorov-Smirnov (KS [31]) and superparamagnetic clustering (SPC) used in Waveclus. The algorithm is only tested using datasets containing three spike waveforms and noise level is swept over the range $\sigma_N = (0.05, 0.1, 0.15, 0.2)$.
d: Overall complexity for $k = 10$ and $W = 20$. $p = 0.65$ in $SL\varphi_{(k)}$ complexity calculation.
e: DF = dimensionality factor.

the borders of the clusters are still clear, and the separation strength is acceptable. The separation strength is quantified as 'discrimination degree' [9]; it is the ratio of intercluster distance to intracluster distance, defined as $DisDeg = (inter/intra)$. The $inter$ is the distance between two clusters and $intra$ is the radius of the cluster. The $DisDeg$ between green and blue clusters are shown in the first row of Fig. 11. The exception is the case of ADDs (the last column in Fig. 11) which introduces more overlap between the clusters, degrading the clustering performance. The $AvgDisDeg$ which is the average $DisDeg$ between all possible cluster combinations annotated in each plot proves that the 2D projections of uPCA and $SL\varphi_{H_h(k)}$ provide better discrimination.

C) Computational Complexity

Due to the resource constraints imposed by implantable devices, the computational complexity of feature extraction and clustering processes is as important as the accuracy. The computational complexity is defined in terms of the number of basic arithmetic operations needed to calculate each feature. It is expressed as [9]:

$$\mathrm{Comp} = \mathrm{N_{add/sub}} + 10 \times \mathrm{N_{mul/div}} \tag{6}$$

where $\mathrm{N_{add/sub}}$ is the number of additions (or subtractions), and $\mathrm{N_{mul/div}}$ is the number of multiplications (or divisions) required. For 10-bit resolution the complexity of multiplications (or divisions) are 10 times more costly than additions (or subtractions).

Table II shows the computational complexity, CER and dimensionality factor (DF) of each feature extraction method. The $\mathrm{N_{add/sub}}$ required to perform ADDs and FDVSDV are computed based on a delay factor required for subtracting the present and buffered samples. For example, consider the complexity of ADDs defined by the tuned scaling factors in decomposition lines (e.g. $\delta|_{3,4,7}$). In uPCA, the core of feature extraction utilizes standard PCA and $\varphi(k)$ is updated at each data segment. Thus, uPCA requires KW times more additions (N^2+2N+1) and multiplications (N^2+2N) than standard PCA. In the rPCA algorithm, the proposed subspace learning algorithm embeds the informative feature space partitions into the $\varphi(k)$. rPCA initialises the $\varphi(k)$ matrix using the standard PCA algorithm such that at the first data segment $(k=1)$ the number of $\mathrm{N_{add/sub}}$ and $\mathrm{N_{mul/div}}$ is the same as that for standard PCA. The rPCA algorithm requires N subtractions and $3MN+M$ multiplications for each input at subsequent segments (e.g. $k = 2 \ldots 100$). This means that for each segment consisting of W inputs, the algorithm requires W times $\mathrm{N_{add/sub}}$ and $\mathrm{N_{mul/div}}$.

Although $SL\varphi_{(k)}$ does not necessarily update the $\varphi(k)$ matrix at every segment (since the algorithm only updates $\varphi(k)$ if the norm of the residual is greater than the projection), for the purpose of analysis, the worst-case is considered, where the residual $r(k)$ is always higher than the projection so that the $\varphi(k)$ matrix is updated at every data segment (k). In the proposed approach, the dictionaries only contain arrays of $\{-1, 0 \text{ and } 1\}$ and so avoid multiplications and divisions and the second term in Eq. (6) is zero. To compute the number of $\mathrm{N_{add/sub}}$ for the three types of dictionary, a variable p is used to define the percentage of -1 and 1 in the dictionaries. In the expanded form of the projection output $y(k) = \sum_{k=1}^{N} \varphi_{i,1}X_{1,j} + \cdots + \varphi_{i,k}X_{k,j}$ there are $N \times p$ elements of -1 and 1, corresponding to $\mathrm{N_{add/sub}}$ in computing $y(k)$. It should be noted that projection and adaptive updating are executed k and $k-1$ times respectively. Considering $SL\varphi_{(k)}$, Eq. (6) derives $\mathrm{N_{add/sub}}$ in each step of the dictionary-based feature extraction as:

$$SL\varphi_{(k)}{}_{Comp} \tag{7}$$
$$= \begin{cases} M \times W \times (N \times p - 1) \times k, \ A^{'} \\ M \times (W-1) \times (k-1), \ \ B^{'} \\ \big(N + N \times (M \times p - 1) + N \times (W-1)\big) \times (k-1), \ C^{'} \\ (N - M) \times W(N \times p - 1) \times (k-1), \ D^{'} \end{cases}$$

where $A^{'}$, $B^{'}$, $C^{'}$ and $D^{'}$ calculate the complexity of $y(k) = \varphi(k) \times X(k)$, feature energy, amount of residue $r(k)$ and substituting vector in the $\varphi(k)$ matrix. The total number of

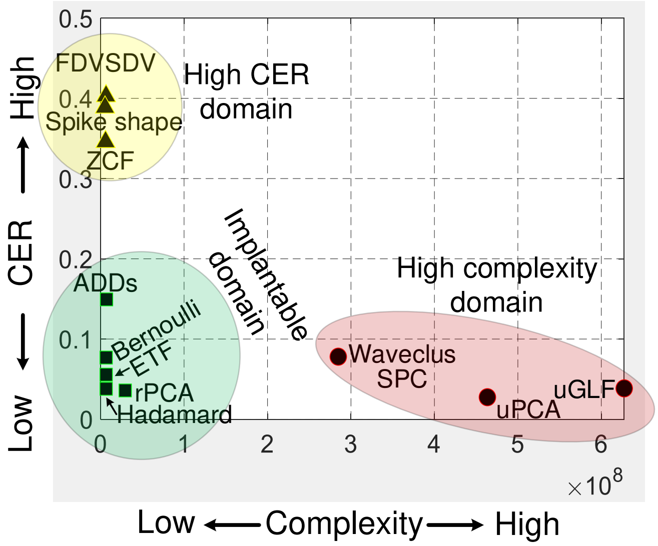

Fig. 12. Classification error (CER) versus computational complexity ($k = 100$, $W = 100$ and $c = 5$) for the different feature extraction methods listed in Table II. Three domains are highlighted in this figure: yellow which shows the high CER, the red domain covers the costly methods which are not suitable for implantable applications and the green domain represent the compatible methods for implantable spike sorting. In the green domain, the dictionaries offer better CER-complexity.

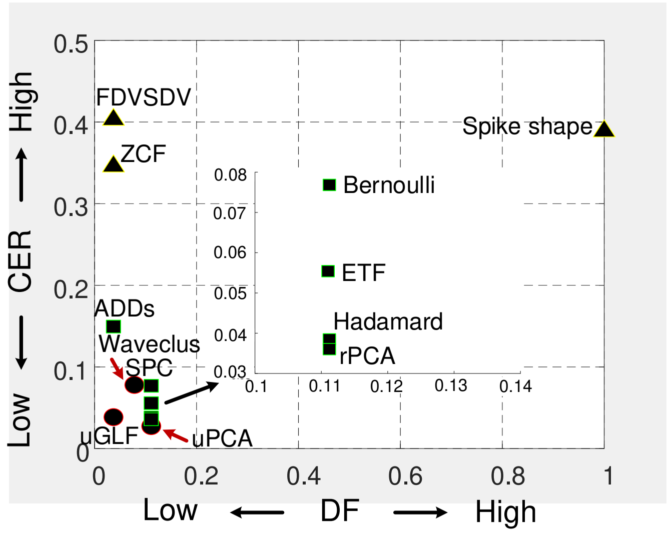

Fig. 13. Classification error (CER) versus dimensionality factor (DF) for the different feature extraction methods.

$N_{add/sub}$ in dictionary-based schemes is the sum of the terms in Eq. (7) $(A' + B' + C' + D')$. Since k is large, $k - 1 \cong k$, the overall complexity is $k(NM \times p + N^2M \times p + MW)$.

As shown in Table I, $SL\varphi_{ETF(k)}$ achieves an average CER of less than 0.06 (6%) over five runs consists of 100 data segments $k = (1 \dots 100)$ and it offers 327 times lower complexity in comparison with uPCA. The overall CER difference between $SL\varphi_{ETF(k)}$ and uPCA is 2.81%. rPCA has 18 times higher complexity compared with $SL\varphi_{ETF(k)}$ and has 1.97% lower overall classification error than $SL\varphi_{ETF(k)}$ using the K-means classifier. Compared to ADDs, $SL\varphi_{ETF(k)}$ has a 9.4% lower CER with 0.8 times the complexity of ADDs for $p = 0.65$.

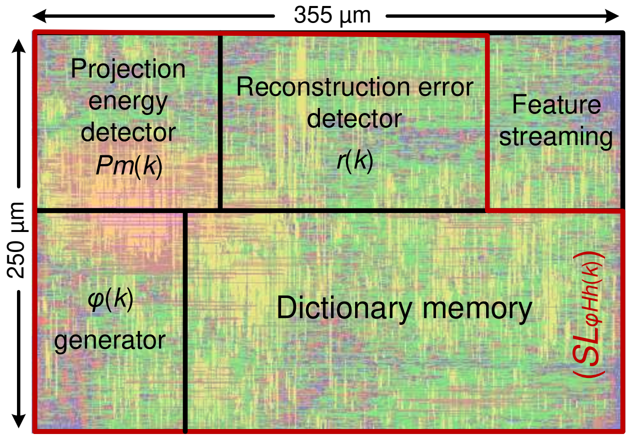

Fig. 14. Layout of the ASIC feature extractor in a 65-nm CMOS process.

TABLE III. ASIC Feature Extractor Summary.

Parameter	Value
Technology	65-nm (TSMC)
Core size	0.09 mm^2
Supply voltage	1 V
Feature streaming power	181 nW
Subspace learning power	3.5 – 10.48 µW
Power density	116.44 µW/mm^2
Operating frequency	30 kHz
Feature extractor type	Adaptive
Memory size	1 kB
Training model	Subspace learning

The K-means operation consists of two different phases: training and data streaming. The distance of each incoming feature vector to the previously finalized centroids is calculated and assigned to the nearest one in the streaming phase.The overall classification complexity of K-means is defined based on Euclidean distance calculations and is estimated to have $kWMc$ additions and $20kWMc$ multiplications per feature vector where M is the number of features per feature vector and c is the number of clusters.

Fig. 12 shows the CER versus computational complexity of feature extraction and sorting for $c = 5$. The dictionary-based feature extractors achieve a better trade-off between CER and complexity. Three different regions are shown: high CER (yellow), high complexity (red) and the green region suitable for implantable applications. Since the projection error is not sensitive to W due to the nature of neural signals, k can be minimized in practical applications allowing lower computational complexity. Fig. 13 shows the average CER versus DF for the methods in Table II. DF is the ratio of feature space dimensions to the number of samples per spike (M/N). In summary, the feature extraction methods $SL\varphi_{H_h(k)}$, $SL\varphi_{ETF(k)}$ and $SL\varphi_{Bern(k)}$ provide better trade-off between clustering error and computational complexity and introduce a new class of feature extractors (green region in Fig. 12) with improved sorting performance within the implantable devices constraints. The feature extractors are not at present capable of resolving overlapping spikes which would degrade performance. An overlap detector and a more powerful subspace matched filtering scheme will be included in the next generation of dictionary-based feature extractors.

D) Hardware Implementation Results

A hardware version of the feature extractor with the Hadamard dictionary ($SL\varphi_{H_h(k)}$) was tested using an FPGA (Artix-7-XC7A200T) with a MATLAB/Simulink interface to a PC [32]. The CER was evaluated using approximately 1000 sets of simulated neural data using the neural simulator in Section II-E, with varying numbers of spike waveforms $\mathrm{NOC} = (3, 4, 5, 6)$ and noise levels $\sigma_N = (0.05, 0.1, 0.15 \text{ and } 0.2)$. The system achieved an average CER of about 6 % over five runs.

A preliminary ASIC design of the $SL\varphi_{H_h(k)}$ feature extractor was implemented using a 65-nm CMOS process for a 1 V supply voltage. The ASIC was synthesized using Synopsys Design Compiler and place-and-route was done using Cadence SoC Encounter. The ASIC, shown in Fig. 14, occupies 0.09 mm^2. Simulation suggests a power dissipation of 181 nW when running at 8-bit resolution and 30 kHz operating frequency in the feature streaming phase. The power consumption for subspace learning varies between 3.5 µW and 10.48 µW. It is a function of recording channel characteristics including the number of active neurons, noise level and the number of leaning iterations when capturing the informative subspace. The power density of the ASIC, in the worst-case scenario during the learning phase, is 116.44 µW/mm^2. It is 6.87 times lower than the power density (800 µW/mm^2) reported to damage brain cells [33]. Table III summarizes the performance of the ASIC.

IV. Conclusion

A new methodology for realization of robust adaptable dictionary-based methods for learning the informative subspace parameters in neural activity has been examined. Three types of dictionary which contain only $\{-1, 0 \text{ and } 1\ \}$ (Hadamard, ETF and random Bernoulli) for implementing adaptive subspace updating schemes have been compared with existing systems. The subspace changes are recorded by projection of the original vectors into a low dimensional space. This offers segmentation and retention of useful information with low fidelity loss. Due to the use of $\{-1, 0 \text{ and } 1\}$ the adaptive updating schemes avoid the high computational cost associated with singular value decomposition calculations. Using simulated noisy neural signals and the K-means algorithm to separate the clusters it has been shown that these dictionary methods have accuracies similar to uPCA and rPCA. For example, ETF has an overall classification error of 5.56%, which is 2.81% higher than uPCA, but it has 327 times lower computational complexity and therefore power requirements. They have also been shown to be significantly more robust than existing computationally efficient methods such as ADDs, FDVSDV and ZCF.

The $SL\varphi_{H_h(k)}$ algorithm has been tested with an FPGA interfaced to MATLAB/Simulink and shows minor CER deviations (1.61%) from the simulation models. It has also been synthesized as an ASIC using a 65-nm CMOS process. In an area 0.09 mm^2 it consumes up to about 10.48 µW, or 116.44 µW/mm^2. The power density in the ASIC is 6.87 times lower than the power density reported to damage brain cells. The dictionary-based feature extractors are, therefore, potentially highly suitable for implantable devices. Future work will include detection of overlapping spikes and a more powerful subspace matched filtering scheme.

References

[1] R.J. Vetter, J.C. Williams, J.F. Hetke, E.A. Nunamaker, and D.R. Kipke, "Chronic neural recording using silicon-substrate microelectrode arrays implanted in cerebral cortex," *IEEE Trans. Biomed. Eng.*, vol. 51, no 6, pp. 896–904, 2004.

[2] N. Bullard, et al., "Design and testing of a 96-channel neural interface module for the networked neuroprosthesis system," *Bioelectronic Medicine*, vol. 5, no. 1, pp. 1-14, Feb 2019.

[3] A. Mohammed, M. Zamani, R. Bayford, and A. Demosthenous, "Toward on-demand deep brain stimulation using Online Parkinson's disease prediction driven by dynamic detection," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 25, no. 12, pp. 2441–2452, Dec. 2017.

[4] N. Tran, S. Bai, J. Yang, H. Chun, O. Kavehei, Y. Yang, V. Muktamath, D. Ng, H. Meffin, M. Halpern, and E. Skafidas, "A complete 256-electrode retinal prosthesis chip," *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 751–765, Mar. 2014.

[5] M. Azin, D. Guggenmos, S. Barbay, R. Nudo, and P. Mohseni, "A battery-powered activity-dependent intracortical microstimulation ic for brain-machine-brain interface," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 731–745, Apr. 2011.

[6] M. Capogrosso et al., "A brain-spine interface alleviating gait deficits after spinal cord injury in primates," *Nature*, vol. 539, pp. 284–288, Nov. 2016.

[7] A. Mendez, A. Belghith, M. Sawan, "A DSP for sensing the bladder volume through afferent neural pathways," *IEEE Trans. Biomed. Eng.*, vol. 8, pp. 552–564, Jun. 2014.

[8] G. Buzsáki, A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, no. 5679, pp. 1926–1929, Jun. 2004.

[9] M. Zamani and A. Demosthenous, "Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper limb neural prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 716–726, Jul. 2014.

[10] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, pp. 433–459, 2010.

[11] Y. Ghanbari, P. E. Papamichalis, and L. Spence, "Graph-Laplacian Features for Neural Waveform Classification," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 5, pp. 1365–1372, May 2011.

[12] M. R. Keshtkaran, and Z. Yang, "Noise-robust unsupervised spike sorting based on discriminative subspace learning with outlier handling," *J. Neural Eng.*, p.036003, March. 2017.

[13] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *J. Neural Comp.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[14] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher, "Thermal impact of an active 3-D microelectrode array implanted in the brain," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 4, pp. 493–501, Dec. 2007.

[15] M. Zamani, D. Jiang, and A. Demosthenous, "An adaptive neural spike processor with embedded active learning for improved unsupervised sorting accuracy," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 3, pp. 665–676, Jun. 2018.

[16] A. Kamboh and A. Mason, "Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 1, pp. 1–9, Jan. 2013.

[17] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi MR, A. Eftekhar A, and T. G. Constandinou, "Feature extraction using first and second derivative extrema (FSDE) for real-time and hardware-efficient spike sorting," *J. Neurosci. Methods*, vol. 215, no. 1–2, pp. 29–37, Jan. 2013.

[18] V. Karkare, S. Gibson, and D. Markovic´, "A 75-µW, 16-channel neural spike-sorting processor with unsupervised clustering," *IEEE J. Solid State Circuits*, vol. 48, no. 9, pp. 2230–2238, Sep 2013.

[19] J. Seberry, B. JWysocki, and T. AWysocki, "On some applications of Hadamard matrices," *Metrika*, vol. 62, nos. 2–3, pp. 221–239, 2005.

[20] M. Fickus, D. G. Mixon, J. D. Peterson and J. Jasper, "Steiner equiangular tight frames redux,", *Proc. 2015 Int. Conf. Sampling Theory and Applications (SampTA)*, Washington, DC, 2015, pp. 347-351.

[21] V. Rojkova and M. Kantardzic, "Feature extraction using random matrix theory approach," *Proc. Sixth Int. Conf. Machine Learning and Applications (ICMLA 2007)*, Ohio, Dec 2007, pp. 410–416.

[22] R. Vidal, Y. Ma, and S. Sasry. "Generalized principal component analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945-1959, Dec. 2005.

[23] L. Scharf and B. Friedlander, "Matched subspace detectors," *Signal Process.*, vol. 42, no. 8, pp. 2146–2157, 1994.

[24] Available online: https://www2.le.ac.uk/centres/csn/research-2/spike-sorting

[25] U. Rutishauser, E. M. Schuman, and A. N. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo," *J. Neurosci. Methods*, vol. 154, no. 1–2, pp. 204–224, Jun. 2006.

[26] P. Mitra and H. Bokil, *Observed Brain Dynamics.* Oxford University Press, 2007.

[27] R. R. Harrison, P. T. Watkins, R. J. Kier, R. J. Lovejoy, B. Greer, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 123–133, Jan. 2007.

[28] D. J. Bora and A. K. Gupta, "Effect of different distance measures on the performance of K-Means algorithm: An experimental study in Matlab," *Int. J. Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2501-2506, 2014.

[29] J. Lian, G. Garner, D. Muessig, and V. Lang, "A simple method to quantify the morphological similarity between signals," *J. Signal Process.*, vol. 90, no. 2, pp. 684–688, Feb. 2010.

[30] M. Zamani, and A. Demosthenous, "Dimensionality reduction using asynchronous sampling of first derivative features for real-time and computationally efficient neural spike sorting," *Proc. ICECS,* Abu Dhabi, United Arab Emirates, Dec. 2013, pp. 237–240.

[31] H. W. Lilliefors, "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *J. Amer. Statist. Assoc.*, vol. 62, no. 318, pp. 399–402, June 1967.

[32] M. Zamani, "Computationally Efficient Adaptive Spike Processor With Real-Time Decoding of Neural Signals for Implantable Applications," Ph.D. Thesis, UCL, London, UK, 2017. Available online: https://discovery.ucl.ac.uk/id/eprint/1556329/.

[33] T. M. Seese, H. Harasaki, G. M. Saidel, and C. R. Davies, "Characterization of tissue morphology, angiogenesis, and temperature in the adaptive response of muscle tissue to chronic heating," *Lab. Invest.*, vol. 78, no. 12, pp. 1553–1562, Dec. 1998.

Majid Zamani (S'13–M'17) was born in Tehran, Iran, in 1984. He received the M.Sc. degree in microelectronics from the Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2011, and the Ph.D. degree from University College London (UCL), London, U.K., in 2017. He is currently a Research Associate with the Analog and Biomedical Electronics Group, UCL. His research interests include design and fabrication of advanced and energy-efficient computational systems utilizing pattern recognition, machine learning and computer vision algorithms, especially for wearable and implantable biomedical applications. He was recipient of the Overseas Research Scholarship and a UCL Graduate Research Scholarship to pursue his Ph.D. degree. He was also the recipient of the Best Researcher M.Sc. Student Award.

Jure Sokolić received the Diploma degree in electrical engineering from University of Ljubljana, Slovenia, in 2013 and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London, U.K., in 2017. He was a Vest Scholar at Duke University in 2016–2017, a postdoctoral research associate at the Department of Biomedical Engineering, King's College London, U.K., in 2017-2018; and a data scientist at Tesco PLC in 2018-2019. Since 2020 he has been a program manager at Metronik d.o.o., Slovenia.

Francesco Renna (SM'19) received the Laurea Specialistica degree in telecommunication engineering and the Ph.D. degree in information engineering, both from the University of Padova, Padova, Italy, in 2006 and 2011, respectively. Between 2007 and 2019, he held Visiting Researcher and Postdoctoral appointments with Infineon Technology AG, Princeton University, Georgia Institute of Technology (Lorraine Campus), Supelec, University of Porto, Duke University, University College London, and University of Cambridge. Since 2019, he has been an Assistant Researcher with the Instituto de Telecomunicações and University of Porto, Porto, Portugal. His research interests include high-dimensional information processing and biomedical signal and image processing.
Dr. Renna was the recipient of a Marie Skłodowska-Curie Individual Fellowship and research fellowships from the Portuguese Foundation for Science and Technology.

Miguel R. D. Rodrigues (SM'15) received the Licen-ciatura degree in electrical and computer engineering from the University of Porto, Porto, Portugal, and the Ph.D. degree in electronic and electrical engineering from the University College London (UCL), London, U.K. He is currently a Professor of Information Theory and Processing at UCL and a Turing Fellow with the Alan Turing Institute - the UK National Institute of Data Science and Artificial Intelligence. His research lies in the general areas of information theory, information processing, and machine learning. His work has led to over 200 articles in leading journals and conferences in the field, a book on "Information-Theoretic Methods in Data Science" published by Cambridge University Press, and the IEEE Communications and Information Theory Societies Joint Paper Award 2011.

Dr. Rodrigues is an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY, and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. He was an Associate Editor of the IEEE COMMUNICATIONS LETTERS, and Lead Guest Editor of the Special Issue on "Information-Theoretic Methods in Data Acquisition, Analysis, and Processing" of the IEEE JOURNAL ON SELECTED TOPICAS IN SIGNAL PROCESSING. He served as Co-Chair of the Technical Programme Committee of the IEEE Information Theory Workshop 2016, Cambridge, UK. He is a member of the IEEE Signal Processing Society Technical Committee on "Signal Processing Theory and Methods", and the EURASIP SAT on Signal and Data Analytics for Machine Learning (SiG-DML).

Andreas Demosthenous (S'94–M'99–SM'05–F'18) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 1992, 1994, and 1998, respectively. He is currently a Professor with the Department of Electronic and Electrical Engineering, UCL, and leads the Analog and Biomedical Electronics Group. He has made outstanding contributions to improving safety and performance in integrated circuit design for active medical devices, such as spinal cord and brain stimulators. He has numerous collaborations for cross-disciplinary research, both within the U.K. and internationally. He has authored over 300 articles in journals and international conference proceedings, several book chapters, and holds several patents. His research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal processing applications.

Dr. Demosthenous is a fellow of the Institution of Engineering and Technology and a Chartered Engineer. He was a co-recipient of a number of Best Paper Awards and has graduated many Ph.D. students. He was an Associate Editor from 2006 to 2007 and the Deputy Editor-in-Chief from 2014 to 2015 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS, and an Associate Editor from 2008 to 2009 and the Editor-in-Chief from 2016 to 2019 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS. He is an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS and serves on the International Advisory Board of Physiological Measurement. He has served on the technical committees for a number of international conferences, including the European Solid-State Circuits Conference (ESSCIRC) and the International Symposium on Circuits and Systems (ISCAS).