DCT Learning-Based Hardware Design for Neural Signal Acquisition Systems

Cosimo Aprile
LIONS and LSM, EPFL
Lausanne, Switzerland
cosimo.aprile@epfl.ch

Johannes Wüthrich
LSM, EPFL
Lausanne, Switzerland

Luca Baldassarre
LIONS and Gamaya
Lausanne, Switzerland

Yusuf Leblebici
LSM, EPFL
Lausanne, Switzerland

Volkan Cevher
LIONS, EPFL
Lausanne, Switzerland

ABSTRACT

This work presents an area and power efficient encoding system for wireless implantable devices capable of monitoring the electrical activity of the brain. Such devices are becoming an important tool for understanding, real-time monitoring, and potentially treating mental diseases such as epilepsy and depression. Recent advances on compressive sensing (CS) have shown a huge potential for sub-Nyquist sampling of neuronal signals. However, its implementation is still facing critical issues in delivering sufficient performance and in hardware complexity. In this work, we explore the trade-offs between area and power requirements applying a novel DCT Learning-Based Compressive Subsampling approach on a human iEEG dataset. The proposed method achieves compression rates up to $64\times$, increasing the reconstruction performance and reducing the wireless transmission costs with respect to recent state-of-art. This new fully digital architecture handles the data compression of each individual neural acquisition channel with an area of $490 \times 650 \mu m$ in 0.18 μm CMOS technology, and a power dissipation of only $2\mu W$.

Keywords

Compressive Sensing, neural signals, learning-based digital signal processing, area-efficient, low-power, signal recovery.

1. INTRODUCTION

Wireless implantable devices capable of monitoring the electrical activity of the brain are becoming an important tool to close the gap between current bulky medical solutions and wearable devices for the treatment of some widespread mental diseases. While such devices exist, it is still necessary to address several challenges to make them more practical in terms of area and power dissipation. Considering multiple site or high frequency oscillation (HFO) recordings, the
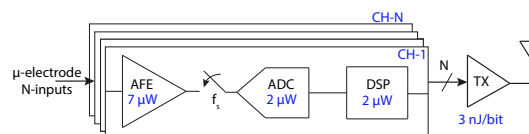
Figure 1: Block diagram of a multiple-channel neural recording system and power dissipation of its components.

payload of data telemetry from the implant to the external base station grows enormously.

The typical circuit blocks used in a multiple channel sensors for medical monitoring are shown in Figure 1. In a wireless monitoring system, the power consumed by the RF transmitter is usually one order of magnitude more than any other system on the chip [1]. Thus, data compression becomes crucial for reducing the power consumption of data telemetry without losing critical information. To this end, compressive sensing (CS) [2, 3] has been exploited in many recent approaches (e.g. [1, 4, 5] and references therein). In a nutshell, CS consists in taking fewer linear samples than dictated by the Shannon-Nyquist theorem, while still allowing robust off-line signal reconstruction. This is possible by exploiting the fact that the information content of a signal is often much lower than its raw data content.

In this work, we present a fully digital encoder for neuronal signals applying a DCT Learning-Based Compressive Subsampling approach (DCT-LBCS), which improves the reconstruction performance and reduces the data telemetry costs by 50% compared to [6], while trading-off the area and power DSP requirements.

The paper is organized as follows. We introduce the main concepts of Compressive Sensing and Learning Based Compressive Subsampling in Section 2. In Section 3, we describe the digital architecture tailored for LBCS. Numerical experiments are reported in Section 4, while in Section 5 we analyse and describe our circuit design. Conclusions are drawn in Section 6.

2. COMPRESSION ALGORITHMS

The main tenet of CS states that a signal $\mathbf{x} \in \mathbb{R}^N$ which has K non-zero coefficients can be robustly recovered from only $M = \mathcal{O}(K \log \frac{N}{K})$ samples $\mathbf{y} \in \mathbb{R}^M$,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} , \qquad (1)$$

where $\mathbf{y}$ offers a compressed version of $\mathbf{x}$, $\mathbf{A}$ is a linear op-

erator that either satisfies the Restricted Isometry Property (RIP) or is incoherent [7], and $\mathbf{w}$ accounts for measurement noise. If we are able to directly sample $\mathbf{y}$, we save both on storage and communication power. Recovering $\mathbf{x}$, though, usually requires to solve a non-linear optimization problem.

Theoretically, i.i.d. sub-Gaussian matrices are incoherent and also satisfy the RIP. Furthermore, they are universal, i.e., given an ortho-normal basis $\boldsymbol{\Phi}$ which allows for a sparser representation of a signal $\mathbf{x}$, the RIP or the incoherence of $\mathbf{A}\boldsymbol{\Phi}$ is the same as of the original $\mathbf{A}$ [7]. However, sub-Gaussian matrices are prohibitively expensive to use in practice, since they require $\mathcal{O}(MN)$ space and time.

Bernoulli (BERN) described in [1], Multi-Channel Sampling (MCS) [5] and Structured Hadamard Sampling (SHS) presented in [8] are randomized sampling approaches recently proposed for the compression of neural signals. These three architectures are very efficient on the sampling side, but require solving non-linear optimization problems to reconstruct the original signals.

Natural signals are often characterized by sparse and structured representations in time-frequency (or space-frequency) domains, such as provided by wavelets [9]. As described in [10] and references therein, a reduced number of samples required for stable recovery can be achieved considering additional structures in the signal $\mathbf{x}$, such as interdependencies between its non-zero coefficients or constraints on its support, during the recovery process.

As discussed in [8], the Hierarchical Group Lasso (HGL) approach gives the best performances over three different structured-sparsity recovery methods. Such approach has been used to compare the reconstructed iEEG signals sampled through BERN, MCS and SHS methods.

2.1 Learning-Based Compressive Subsampling

The LBCS method [11] consists on *linear encoding* and *linear decoding* with respect to a given orthonormal basis, resulting in a much simpler and faster solution compared to standard CS. LBCS can be summarized as follows. Given a signal $\mathbf{x} \in \mathbb{R}^N$, we consider the compression model

$$\mathbf{y} = \mathbf{P}_\Omega \boldsymbol{\Psi} \mathbf{x} , \qquad (2)$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$ is an orthonormal basis and $\mathbf{P}_\Omega \in \mathbb{R}^{M \times N}$ is a subsampling matrix whose rows are canonical basis vectors. The effect of applying $\mathbf{P}_\Omega$ to $\boldsymbol{\Psi}\mathbf{x}$ is to retain only the coefficients indexed by the set Ω, also known as the *subsampling map*. The vector $\mathbf{y} \in \mathbb{R}^M$ is the compressed version of $\mathbf{x}$, with a nominal compression rate (CR) of $\frac{N}{M}$. The signal $\mathbf{x}$ is then approximately recovered via the fast linear decoder

$$\hat{\mathbf{x}} = \boldsymbol{\Psi}^* \mathbf{P}_\Omega^T \mathbf{y} . \qquad (3)$$

Given a training set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ of m fully sampled signals of unit norm, the optimal subsampling map Ω is learnt by choosing the indices that capture most of the average energy in the transform domain:

$$\hat{\Omega} = \underset{\Omega, |\Omega|=M}{\arg\max} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \Omega} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2, \qquad (4)$$

where $\boldsymbol{\psi}_i$ is the i-th row of $\boldsymbol{\Psi}$. $\hat{\Omega}$ can be exactly found by selecting the M indices whose values of $\frac{1}{m}\sum_{j=1}^{m} |\langle \boldsymbol{\psi}_i, \mathbf{x}_j \rangle|^2$ are the largest [11]. The learnt sampling scheme is then used to directly sample only those transform coefficients indexed by $\hat{\Omega}$ for all signals $\mathbf{x}$.
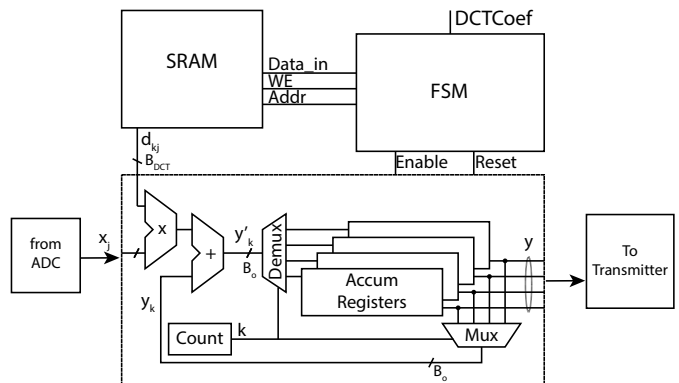

Figure 2: One channel block diagram showing the LBCS encoder and the matrix sequence generation logic.

In [6], LBCS is exploited using the Hadamard transformation matrix, which is particularly suited to a hardware implementation because each coefficient can be computed by performing only additions or subtractions. A DCT based transformation matrix yields better reconstruction performances considering the same frame size and compression rate. However, each DCT matrix entry needs more bit resolution which would require an increased hardware intensity. As will be discussed in Section 5, a trade-off can be found to reduce the hardware complexity over an increased reconstruction performance, motivating a DCT based encoding system (named DCT-LBCS).

2.2 Full DCT Compression vs DCT-LBCS

The optimal linear encoding would require a full transformation of signal window $\mathbf{x}$, followed by an *adaptive* compression that retains only the M largest coefficients of $\boldsymbol{\Psi}\mathbf{x}$ in absolute value. However, such adaptive encoding requires to first compute all the coefficients $\boldsymbol{\Psi}\mathbf{x}$, which results prohibitive with area and power consumptions. For such reason, a trade-off should be set to have a good signal quality after reconstruction, while limiting the power and area needs.

3. SYSTEM ARCHITECTURE

The one-channel sampling DCT-LBCS architecture proposed in this work is depicted in Figure 2. The embedded sampling and compression of the neural input signal follows the description presented in Section 2.1.

In the following, we fix $\boldsymbol{\Psi}$ equal to the DCT matrix. Let $\mathbf{D}_\Omega = \mathbf{P}_\Omega \boldsymbol{\Psi}$ be the matrix composed of the rows of $\boldsymbol{\Psi}$ indexed by Ω. We sequentially compute $\mathbf{y} = \mathbf{D}_\Omega \mathbf{x}$: looking at each component of $\mathbf{y}$, we have

$$y_k = \sum_{j=1}^{N} d_{kj} x_j, \ k \in \{1, \ldots, M\}, \qquad (5)$$

where d_{kj} is the (k, j)-entry of $\mathbf{D}_\Omega$.

The DCT transformation matrix $\mathbf{D}_\Omega$ contains real valued coefficients (positive and negative), which are stored into an SRAM, shown in Figure 2, with $N \times M$ cells of size B_{DCT}.

A finite state machine (FSM) drives the LBCS encoder sub-sampling procedure. The entries d_{kj} are stored into the chip memory in a sequential fashion through the *DCTCoef* input. The input signal x_j is the digital output of an A/D converter with a resolution of B_i bits. The sampling procedure starts once the memory is loaded and the operations are carried out by a single multiplier and an adder, which

are used in a time-multiplexed manner to accumulate the M output values into the registers.

At each time step j, x_j is multiplied to the DCT entry d_{kj}, and summed to the B_o-bit accumulator value y_k, updating each component following the rule $y'_k = y_k + d_{kj}x_j$, $k \in \{1, \ldots, M\}$. At the beginning of each window of length N, the registers are then reset ($\mathbf{y} = 0$). The enable signal is meant to drive the digital registers, so that each accumulator is updated before the next sample x_j arrives. This design choice avoids having one multiplier-adder per accumulator lane, but requires an internal digital clock frequency $f_{encoder} = M \times f_s$, where f_s is the signal sampling frequency.

The input data sampling frequency for the considered dataset is 5 kHz, and as further described in Section 4, choosing a window length $N = 256$ and a compression rate of $32\times$, the DCT-LBCS encoder frequency results to be $5 \text{ kHz} \times \frac{256}{32} = 40$ kHz, which is still in a relatively low frequency range. Indeed, if $M = \frac{N}{CR}$ is large, the internal clock frequency may become a limiting factor, requiring additional digital blocks for clock synchronization.

4. SIMULATIONS

The `iEEG.org` portal contains several datasets of EEG and iEEG data which are manually annotated by expert clinicians. We focus on the `I001-P034-D01` dataset, which consists of approximately 1 day, 8 hours and 10 minutes of recordings at 5kHz, or approximately $6 \cdot 10^8$ samples. In order to reduce the dataset size, we use samples only from the 12th and 13th seizure, and an equal number of samples before the seizure onset, for training and testing respectively. In order to better compare to the sampling strategy that combines samples across the channels (MCS), we consider only a sub-grid of 4×4 electrodes.

In the following we use such dataset to compare the numerical results obtained applying the DCT-LBCS encoder against the other approaches described in Section 2.

4.1 Experimental Protocol and Performance Evaluation

The training portion of the dataset is used to learn the sampling pattern for both Had- and DCT-LBCS approaches and also to tune the variable density parameters for the SHS method. Then, the fixed sampling pattern is used by LBCS to compress all the signal windows in the test set. The reconstruction is then performed with the linear decoder (3). For the randomized methods, MCS, BERN and SHS, we draw 20 different sampling patterns from the relative distributions for each signal window in the test and reconstruct using the tree-based HGL norm, which was shown in [8] to yield the best results.

All the reconstructed windows for each channel j are concatenated together, forming the entire reconstructed signal $\hat{\mathbf{x}}_j$ for the test seizure. We then compute the SNR for each channel as $\text{SNR}_j = 20 \log_{10}\left(\frac{\|\mathbf{x}_j\|_2}{\|\mathbf{x}_j - \hat{\mathbf{x}}_j\|_2}\right)$, where $\mathbf{x}_j$ is the recorded signal for channel j, and average these SNRs to obtain our final measure of performance, $\text{SNR} = \frac{1}{\#ch}\sum_{i=1}^{\#ch}\text{SNR}_j$. For the randomized methods, we also average over the 20 draws.

4.2 Numerical results

The numerical experiments have been developed with all the methods described in this paper, varying the length of the signal window N, the ADC resolution B_i and the com-

Table 1: Performance (dB) N = 256, B_i = 10, B_{DCT} = 8

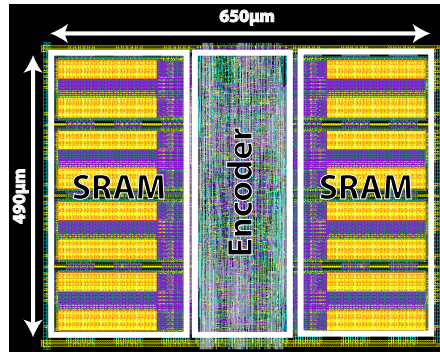| Method	Compression rate					
	2	4	8	16	32	64
DCT Adaptive	42.03	41.96	40.16	37.36	32.88	25.63
DCT LBCS	**41.65**	**40.66**	**38.59**	**35.55**	**31.00**	**23.97**
Had-Adaptive	41.60	39.86	36.38	31.40	25.42	19.43
Had-LBCS	40.79	37.64	33.27	28.48	23.27	18.06
SHS HGL	36.92	27.96	23.89	20.26	18.53	14.49
BERN HGL	37.48	26.69	20.49	16.87	13.53	11.15
MCS HGL	28.96	24.40	20.92	17.48	n.a.	n.a.

Figure 3: One-channel DCT-LBCS encoder layout for $N = 256$ and $CR = 32$.

pression rate CR.

The LBCS approach does not result to be very sensitive to the window length, and for the sake of consistency with the Had-LBCS approach proposed in [6], we present only results for $N = 256$ and $B_i = 10$ bits and resolution of DCT transformation matrix coefficient $B_{DCT} = 8$ bits.

Table 1 reports the reconstruction quality, in dB, obtained on the `I001-P034-D01` dataset. As expected, adaptive DCT compression sets the upper limit on the achievable performance. DCT-LBCS offers the best reconstruction quality at any compression rate, with an increase in the SNR of several dBs compared to the other methods. The Adaptive Hadamard yields the second best performance and sets the upper limit for the Hadamard-based approach. Interestingly, the DCT-LBCS method offers a comparable performance to the Adaptive Hadamard even at higher compression rate. In the SHS approach the variable density is adapted to the signals, but still fails at capturing as much structure as LBCS. The BERN and MCS methods offer a much inferior performance at high compression rates, because imposing structure only during reconstruction does not fully compensate the limitations of their structure-unaware sampling mechanisms.

As described in Section 2.2, given a fixed signal window length and compression rate, the best linear encoder is given by adaptively sampling the coefficients that capture most of the energy of each signal. The LBCS-based reconstruction performances are close to the ones obtained with the adaptive encoder, but at a fraction of its power and area cost.

The linear decoder (3) yields reconstructions with less computational cost of the other methods. Indeed, solving a single optimization problem with the HGL norm, using `DecOpt` [12], requires on average approximately 0.1 s, while the linear decoder requires only approximately 10^{-5} s for a signal with 256 samples.

5. CIRCUIT DESIGN AND VALIDATION

The circuit implementation has been defined following the experimental results discussed in Section 4 and considering

Table 2: Comparison With Published Work

Parameter	[1]	[5]	[6]	This Work
Compression Method	BERN	MCS	Had LBCS	DCT LBCS
Technology [μm CMOS]	0.09	0.18	0.09	0.18
Compression Rate	20	16	16	32
Compression Power [μW]	1.9	17.83*	1.0	2.0
Compression Area [mm^2]	0.090	0.090	0.044	0.3
Recovered Signal [dB]	15.76	17.48	28.48	31.00
TX-Power @ f_s [μW]	1.5	0.94	1.7	0.85

* Compression power cost over 16 channels.

the trade-off between area and power requirements. The target signal reconstruction quality is set to 30 dB. Considering a sampling window length of 256 samples and assuming an ADC resolution of $B_i = 10$ bits, the Had-LBCS method reaches 30 dB performance with a compression ratio $CR = 16$. As reported in Table 1, with the DCT-LBCS approach a compression ratio $CR = 32$ still allows to have a performance higher than 30 dB (and improved with respect to the Had-LBCS design). Thus, we are allowed to relax the number of bits to transmit, which is directly related to the RF data transmission cost. The internal encoder core clock frequency is $f_{encoder} = M \times f_s = 40$ kHz with the accumulator resolution set as $B_o = B_i + \log_2(N) + 1$ to avoid overflow. This leads to define an *effective compression ratio* defined as $CR_{eff} = CR \times \frac{B_i}{B_o}$, which takes into account the actual number of bits per accumulator, after the compression.

Table 2 reports the performance of the system and presents a comparison with recent published work. In this table is summarized the compression power and area requirements for each methods discussed in this paper. It also reports the simulated recovered signal and transmitter performances, highlighting how the DCT-LBCS approach reduces the RF data telemetry cost while improving by almost 3 dB the performances with respect to the best approach presented in [6]. On the other hand, the area requirement is higher because of an increased bit resolution per DCT matrix entry and because of a different CMOS technology node. However, considering a multiple channel application, the memory content is shared among all the channels, reducing the impact of the storage area over the overall chip.

The architecture shown in Figure 2 has been implemented in a 1P6M 0.18 μm CMOS technology. The layout of the fully digital one-channel encoder is shown in Figure 3. To verify the functionality of the digital encoder, the digitized neuronal data is directly given as input to the DCT-LBCS block. A post place-and-route simulation has verified that the M outputs given by the encoder are equal to the expected values computed through MATLAB software. The simulation has been run considering a worst case scenario with slow-slow process corner operating at 1.8 V, which results in an estimated power consumption of the DCT-LBCS encoder around 2 μW. The silicon area of the encoder block is 490×650 μm.

6. CONCLUSIONS

This work presents an on-the-fly data compression system applying a novel DCT-LBCS approach, which improves the reconstruction performance and reduces the data telemetry costs by 50%, while trading-off the area and power requirements compared to [6]. The memory that stores the subsampled DCT matrix entries occupies a relative large area. However, in a multichannel implementation, the memory

content is shared among all the channels, reducing the impact of the storage area over the overall chip area. The faster DCT-LBCS off-line recovery and higher reconstruction quality than standard CS makes it suitable to any sparse data acquisition system for which fully sampled signals are available for training (e.g., image processing and remote sensing).

Acknowledgment

7. REFERENCES

[1] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[2] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[3] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," in *IEEE International Symposium on Circuits and Systems*, 2007, pp. 1959–1962.

[5] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 6, pp. 857–870, December 2014.

[6] C. Aprile, L. Baldassarre, V. Gupta, J. Yoo, M. Shoaran, Y. Leblebici, and V. Cevher, "Learning-based near-optimal area-power trade-offs in hardware design for neural signal acquisition," in *Proceedings of the 26th edition of Great Lakes Symposium on VLSI*. ACM, 2016, pp. 433–438.

[7] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. Springer, 2013.

[8] L. Baldassarre, C. Aprile, M. Shoaran, Y. Leblebici, and V. Cevher, "Structured sampling and recovery of ieeg signals," in *6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015.

[9] S. Mallat, *A wavelet tour of signal processing*. Academic press, 1999.

[10] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher, "Structured sparsity: Discrete and convex approaches," in *Compressed Sensing and its Applications*. Springer, 2015, pp. 341–387.

[11] L. Baldassarre, Y.-H. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher, "Learning-based compressive subsampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 809–822, 2016.

[12] Q. Tran-Dinh and V. Cevher, "A primal-dual algorithmic framework for constrained convex minimization," *arXiv preprint arXiv:1406.5403*, 2014.