Neural Signal Processing and Closed-loop Control Algorithm Design for an Implanted Neural Recording and Stimulation System

Lei Hamilton, Marc McConley , Kai Angermueller, David Goldberg, Massimiliano Corba, Louis Kim, James Moran, Philip D. Parks and Sang ("Peter") Chin

Charles Stark Draper Laboratory, Cambridge,
Massachusetts
lhamilton@draper.com

Alik S. Widge[a,b], Darin D. Dougherty[a], Emad N. Eskandar[c]

[(a)] Department of Psychiatry, Massachusetts General Hospital and Harvard Medical School, Boston, Massachusetts
[(b)] Picower Institute for Learning & Memory, Massachusetts Institute of Technology, Cambridge, Massachusetts
[(c)] Department of Neurological Surgery, Massachusetts General Hospital and Harvard Medical School, Boston, Massachusetts

Abstract – A fully autonomous intracranial device is built to continually record neural activities in different parts of the brain, process these sampled signals, decode features that correlate to behaviors and neuropsychiatric states, and use these features to deliver brain stimulation in a closed-loop fashion. In this paper, we describe the sampling and stimulation aspects of such a device. We first describe the signal processing algorithms of two unsupervised spike sorting methods. Next, we describe the LFP time-frequency analysis and feature derivation from the two spike sorting methods. Spike sorting includes a novel approach to constructing a dictionary learning algorithm in a Compressed Sensing (CS) framework. We present a joint prediction scheme to determine the class of neural spikes in the dictionary learning framework; and, the second approach is a modified OSort algorithm which is implemented in a distributed system optimized for power efficiency. Furthermore, sorted spikes and time-frequency analysis of LFP signals can be used to generate derived features (including cross-frequency coupling, spike-field coupling). We then show how these derived features can be used in the design and development of novel decode and closed-loop control algorithms that are optimized to apply deep brain stimulation based on a patient's neuropsychiatric state. For the control algorithm, we define the state vector as representative of a patient's impulsivity, avoidance, inhibition, etc. Controller parameters are optimized to apply stimulation based on the state vector's current state as well as its historical values. The overall algorithm and software design for our implantable neural recording and stimulation system uses an innovative, adaptable, and reprogrammable architecture that enables advancement of the state-of-the-art in closed-loop neural control while also meeting the challenges of system power constraints and concurrent development with ongoing scientific research designed to define brain network connectivity and neural network dynamics that vary at the individual patient level and vary over time.

Keywords - Neural Stimulation, Neuropsychiatric Disorders, Closed-loop Control, Decode, Signal Processing

I. INTRODUCTION

A. Overview

Brain disorders, particularly PTSD, depression, and addiction, impair U.S. warfighters, veterans and civilians and are the leading cause of disability and lost productivity. Many of the currently available psychiatric treatments fail to treat the symptoms of the patient that are the most ill. Furthermore, existing brain stimulators modulate a single brain area in an open-loop fashion. If complex neuropsychiatric disorders reflect network-wide derangements in activity and connectivity, single deep brain stimulation targets cannot address them. And, a next-generation, ultra-flexible implantable stimulation and recording platform would seek to address this unmet need. The Defense Advanced Research Projects Agency (DARPA) Systems-Based Neurotechnology for Emerging Therapies (SUBNETS) program aims to develop next generation closed-loop control algorithms, software and requisite hardware to treat a broad spectrum of neuropsychiatric disorders. The implantable system is based on a distributed architecture that consists of high-density electrodes attached to multiple satellite *electronics modules* that are distributed around a centralized *hub electronics module* that provides command and control. Each satellite is connected to a high density electrode and a cable which connects to the central hub, both of which are hardwired connections. The cable from the satellite plugs into a connector header integrated within the main hub system. The connector header is design to allow for up to 5 discrete satellite systems to be connected to the central hub. Satellite modules are dynamically reconfigurable to provide on-demand neural recording and stimulation through hundreds of electrode channels, while the hub module controls adaptive neural stimulation in response to real-time neural

978-1-4244-9270-1/15/$31.00 ©2015 IEEE

activity. The satellite/hub system is implanted under the scalp and wirelessly communicates with an *external base station* for data streaming, reprogramming, and recharging. This next generation, ultra-flexible implantable stimulation and recording platform will generate the whole-brain data necessary for training models and testing algorithms.

B. Algorithm Key Innovations

To accommodate this ultra-flexible device, we designed the algorithm and software using an innovative, adaptable, and reprogrammable architecture that enables advancement in the state-of-the-art in closed-loop neural control while meeting the challenges of system power constraints. It is being concurrently developed with ongoing scientific research designed to define brain network connectivity and neural network dynamics that vary at the individual patient level and vary over time. For example, we have optimized the processing architecture to distribute and partition functionality across several parts of the system (satellite, hub, base station). We have also designed the software to provide flexibility to adapt the real-time processing via control algorithm parameter uploads to the hub. The system's architecture, functionality, and modes of operation allow decode and control parameters to be optimized in response to a patient or subject's response to therapeutic stimulation or other variables that contribute to neuropsychiatric state. Finally, the clinical and researcher user interface accommodates a wide variety of ways in which the system can be reconfigured at any given time, ranging from highly efficient low-power processing for autonomous operation to large-volume data collection for data streaming and feature identification. Algorithm innovations include real-time signal processing, autonomous spike sorting, neuropsychiatric state decoding, and closed-loop control algorithms optimized for low-power autonomous operation while providing the capability for adaptation and learning when data and computing resources are available.

C. Algorithm Architecture

Figure 1 depicts the high-level architecture of the processing algorithms. The three primary algorithm components are signal processing, decode, and control. Each algorithm consists of real-time components that run in the hub, additional lower-bandwidth adaptive components that run in the base station when present, and longer-term off-line components that run on the PC when present. High-bandwidth data streaming in one direction from the hub to the base station via the "neural link" is shown in the signal processing path by the bold arrow ("Neural link") in Figure 1. The details of these different levels of processing are provided in subsequent sections pertaining to the individual algorithm components.

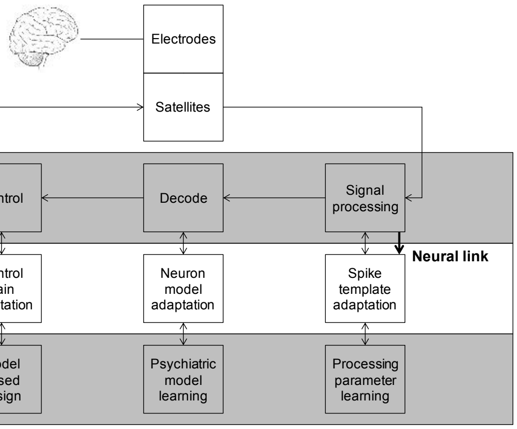


FIGURE 1: HIGH-LEVEL PROCESSING FLOW

The two types of signals that the system processes are neural spikes and local field potentials (LFPs). Each satellite can be configured to process spikes, LFPs, or a combination by setting the sampling rate and analog filter bandwidth as listed in Table 1.

TABLE 1: SATELLITE RECORDING CONFIGURATION PARAMETERS

Configuration	Sampling Rate	Bandwidth
LFP only	1 kHz	1-500 Hz
Spikes only	20 kHz	200-7,500 Hz
LFP and spikes	20 kHz	1-7,500 Hz

D. Configurations

We designed our system to have three different configurations in which the software will be used:

- *The autonomous configuration mode* is the primary system configuration in which the implanted device is not in communication with the base station. In this configuration, onboard processing is optimized to minimize power consumption and consists of recording on selected data channels; triggered data is stored to nonvolatile memory (for retrieval later when the implanted device is in communication with the base station); and, setting fixed-parameter signal processing, decode, and control algorithms.
- *The "base boost" configuration* mode allows adaptive closed-loop control processing using the base station processor. Two telemetry interfaces enable data sharing between the implanted central hub and the base station. A wireless low-bandwidth telemetry interface enables bidirectional communication. The "neural link" option adds a wired high-bandwidth interface for data streaming at much higher rates (20 Mbps) in one direction (from the implanted hub to the base station). This configuration mode enables additional real-time adaptation of some of the lower-level signal processing functions and also incorporates a graphical user interface that provides clinicians and researchers the ability to perform status and control functions. Our vision is that this configuration mode could be used in the hospital or clinical setting to allow extensive recording and calibration

of the decode algorithms with a patient in an untethered and freely behaving state.

- *The computer control configuration mode* allows "tethered" operation and can be used to provide real-time recording and closed-loop control with more sophisticated learning algorithms on the external computer. This configuration mode also allows high-bandwidth data streaming and storage on the largest number of channels (e.g., greater than 300). Data inputs from other external devices (e.g., psychophysiological measures, behavioral task performance) can also be combined with data from the implanted device to help determine algorithm parameters for closed-loop operation. For example, in a clinical or research setting, a patient could execute behavioral tasks that provide information about a specific neuropsychiatric state correlated with neural signals. The external computer also performs off-line parameter estimation. Off-line parameters can be transmitted to the hub to update the algorithms that are used in the autonomous and base boost configuration modes. The computer control configuration mode enables additional user interface options for data visualization and control by the clinician or researcher.

II. Signal Processing

A. *Signal Processing Overview*

Signal processing extracts features from raw neural signals that correlate with neuropsychiatric states of interest for decode and control functions. Multiple processing algorithms are applied on the hub, as depicted in Figure 2. Signal processing consists of high-rate processing of raw neural spike data (identified as "Spike detection" and "Spike sorting"), high-rate processing of local field potential (LFP) data (identified as "Time-frequency analysis"), and lower-rate feature derivation.

Many data channels and combinations of data channels for coupling calculations are available for processing. This leads to a combinatorial increase in the number of outputs from the feature derivation block in Figure 2 if all possible combinations of channels are used. However, we plan to use sparsity-based control where the number of features we use to decode will be one of parameters we will optimize over. Therefore, smaller subsets of channels and combinations of channels are likely sufficient for accurate decoding of a neuropsychiatric state and subsequent control at any given time. Therefore, decode and control performance requirements will be used to specify the selection of channels and combinations of channels to be used in real-time processing. The software will provide a flexible and adaptive interface to specify decode and control channels. The details corresponding to individual processing algorithms appear in the sections that follow.

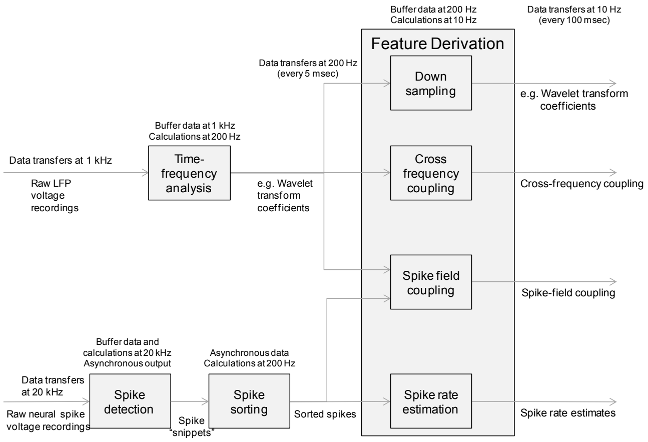


FIGURE 2: SIGNAL PROCESSING ALGORITHM COMPONENTS

B. *Neural Spikes*

There are several challenges associated with applying existing spike sorting algorithms to decode and control neuropsychiatric states. Many existing spike sorting algorithms [1][2][3][4] cannot be implemented for use in real-time on a small implantable device. For example, when using principal component analysis (PCA), the principal components are calculated using all detected spikes. The PCA clustering algorithm also requires large amounts of pre-existing data and often, numbers of clusters need to be preset. Thus, PCA is not feasible for a real-time closed-loop control system. Additionally, PCA clustering algorithms are designed with the assumption that the waveforms from the same neuron will not change over time (the experiment duration is limited so that signals do not drift). Because we expect our system to be implanted for an extended duration of time (e.g., months), we need to prepare for the possibility that these waveforms may change over time. Therefore, we selected a real-time algorithm based on spike template matching where such a change over time can be captured. Detailed descriptions of the neural spike processing steps (depicted in Figure 33) appear in the following sections.

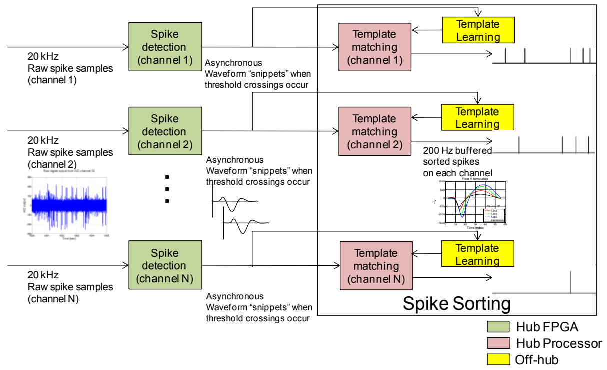

20 kHz
Raw spike samples
(channel 1)
Spike detection (channel 1)
Asynchronous
Waveform "snippets" when threshold crossings occur
Template matching (channel 1)
Template Learning
20 kHz
Raw spike samples
(channel 2)
Spike detection (channel 2)
Asynchronous
Waveform "snippets" when threshold crossings occur
Template matching (channel 2)
Template Learning
200 Hz buffered sorted spikes on each channel
20 kHz
Raw spike samples
(channel N)
Spike detection (channel N)
Asynchronous
Waveform "snippets" when threshold crossings occur
Template matching (channel N)
Template Learning
Spike Sorting
🟩 Hub FPGA
🟥 Hub Processor
🟨 Off-hub

FIGURE 3: SPIKE SORTING ALGORITHM COMPONENTS

1) *Spike detection on FPGA:*

The first step of neural spike processing is spike detection--the process of separating spikes from background noise. The continuous raw neural spike data at 20 kHz from each spike recording channel proceeds through spike detection on each channel and spikes are detected using an amplitude threshold. The outputs of spike detection are asynchronous spike snippets with associated time stamps. We have chosen to implement spike thresholding in the hub FPGA. The FPGA was chosen for the spike detection because it runs at faster rates than the software while providing flexibility to program various spike detection algorithms. We have two different implementations of automatic threshold algorithms [5]. The first is the absolute value (ABS) threshold and the second is Nonlinear Energy Operators (NEO). Both algorithms have shown good results in studies in published literature and choosing one or the other will depend on analysis of data acquired from the device. In addition, we also provide clinicians and researchers with the option of adjusting the threshold manually via the user interface.

2) Spike sorting:

a) Modified OSort approach

Snippets from spike detection are sorted by comparing with templates stored on the hub and by computing the Euclidean distances between spike snippets and templates.--This step is called *template matching*. Because spike waveforms will eventually drift when the base station is in communication with the hub, an adaptive templating algorithm will run to update the templates and compensate for drift. *Template learning* (Figure 3) occurs off the hub on the external computer in the computer control configuration mode. This is similar to the OSort algorithm [5][6]; however, it is but implemented in a distributed system (on the hub for template matching and off the hub for template learning) such that the algorithm operates in real-time and is power efficient. Buffered spike snippets are sent from the hub through high-bandwidth telemetry to the external computer, where a more computational expensive template learning algorithm uses buffered spike snippets to learn, update, and refine the templates. Additionally, the computer control configuration mode allows an experienced researcher or clinician to verify the templates visually (e.g., the user can choose to merge or separate different clusters) and to label the templates (e.g., the user can label different clusters into single units, multi-units, and noise) through a user interface on the external computer. After the templates are learned, they are uploaded to the hub for use in the real-time template matching algorithm.

b) Compressed Sensing approach

For spike signals, there are two approaches of designing a sparsifying dictionary *D* in the compressed sensing (CS) framework. The first approach is to use signal agnostic dictionaries such as the Wavelet or Gabor dictionary, as spikes can be sparsely represented in these time-frequency dictionaries. On the other hand, since each neuron registers unique and repetitive spike signals at the recording electrode, we can learn a signal dependent dictionary using prior information from the spikes. The spikes potentially can be more sparsely represented in the signal dependent dictionary. Therefore, the signal dependent dictionary may possibly outperform the signal agnostic dictionary in terms of compression rate, recovery quality and signal classification accuracy. Moreover, we have also found that using data directly as the sparsifying dictionary can also lead to similar performance with much lower computational cost. Because different neurons generate spikes that are unique and repetitive, our results demonstrate (see section IV of [9]) that the algorithm performance is improved if we represent the signal using dictionary atoms belonging to the same group (same neuron) rather than atoms from different groups (different neurons). To achieve this goal, we enforce group sparsity by dividing our dictionary into different groups and choosing the atoms from the same sub-dictionary.

Neural spikes generated by the same neuron are often recorded closely together in time by nearby channels in similar pattern and shape, which means there is a high correlation between measurements of different channels. To capture this correlation, we implement joint sparsity using row-l0 quasi-norm in our sparse coding stage via simultaneous orthogonal matching pursuit (SOMP) [7]. Sparse coefficients of all channels share the same support pattern by adding the joint sparsity constraint. Extending upon K-SVD [8], we use a multi-modal structured dictionary learning algorithm to incorporate both group sparsity and joint sparsity constraints. Multi-modal means that the dictionary is unique for each individual channel. Our algorithm iterates between a sparse coding stage and a codebook update stage [9].

C. Local Field Potential (LFP)processing

LFP processing consists primarily of time-frequency analysis and subsequent feature derivation from the frequency content in the signals. This is accomplished by applying a transform to the frequency domain on sliding windows of data. We have chosen a complex wavelet transform as the approach to baseline time-frequency analysis. The baseline design for the real-time processing is to produce amplitude and phase with 5-ms resolution based on sliding windows of LFP data on each channel. The following processing takes place for each LFP channel being processed at 200 Hz in the time-frequency analysis function:

1. Buffer the 5 most recent 1-kHz raw LFP samples.
2. Build a data window consisting of the 250 most recent 1-kHz raw LFP samples. (As a baseline, consider a circular buffer containing 250 points, in which the 5 oldest samples are replaced by the 5 most recent samples on each call)
3. For each specified frequency, calculate the real and imaginary parts of the discrete transform using the specified kernel parameters.
4. For each specified frequency, derive amplitude and phase from the real and imaginary parts of the discrete transform.

D. Feature Derivation

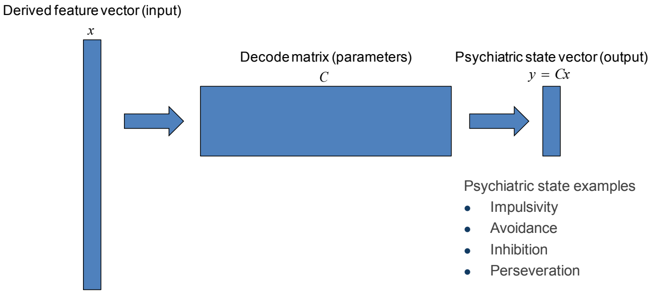
The results of the LFP wavelet transform and sorted spikes go through an additional feature derivation step to generate inputs for the decode algorithm. There are three main processing steps: cross-frequency coupling, spike-field coupling, and spike-rate estimation. Cross-frequency coupling is used to relate the phase at a lower frequency with the amplitude at a higher frequency. This analysis can be performed within a single channel or across multiple channels. Spike-field coupling determines the average LFP phase at different frequencies at which neuron spikes occurred. While sorted spikes will be provided at a high rate, spike rate estimates are also useful for the decode algorithm.

III. Decode

As depicted in Figure 1, the decode algorithm consists of: real-time processing that occurs on the hub processor; neuron model adaptation that can occur on the base station when present; and psychiatric model learning that can occur on the PC when present.

The decode algorithm operates on the derived feature vector (the outputs of signal processing from Figure 2) to calculate a neuropsychiatric state vector to be used as the feedback signal to control. In collaboration with clinical partners, we use a transdiagnostic approach to describe state vectors and addresses functional behavioral domains which overlap between categorical psychiatric diagnoses. The neuropsychiatric state consists of metrics on a variety of transdiagnostic domains, including impulsivity (in one's decision making), avoidance (related to one's ability to regulate fear) inhibition, perseveration (related to one's inability to change his mind), and others that correlate with neural recordings and that are intended to be influenced by control policies for deep brain stimulation.

Error! Reference source not found. shows the decode algorithm interface. The number of rows in the decode matrix is equal to the psychiatric state dimension, and the number of columns is equal to the derived feature dimension.



Derived feature vector (input)
x
Decode matrix (parameters)
C
Psychiatric state vector (output)
$y = Cx$
Psychiatric state examples
Impulsivity
Avoidance
Inhibition
Perseveration

FIGURE 4: DECODE ALGORITHM INTERFACE

IV. Control

A. Control Overview

Coupled with the decode module, our controller operates in neuropsychiatric states and network-graph spaces where the neuropsychiatric state vector is informative of a patient's impulsivity, avoidance, inhibition, cognitive flexibility, etc. The controller is optimized to apply stimulation based on the neuropsychiatric state vector's current state, its history, and other derivatives to determine the optimal stimulation commands to apply (e.g., in terms of power consumption, robustness to noise). We implement a network model-based control algorithm by processing recorded signals (spike and LFP) from multiple brain regions to form a graph.Then, we exploit the structures of the graphs to identify the control parameters that can bring the brain to the most desired neuropsychiatric state as measured by observed behavior and neural signals. Figure 5 shows the control algorithm interface. The stimulation chips accept high-level commands such as amplitude, frequency, and duration of stimulation and direct these into currents to be applied at the stimulating electrodes. We depict the example of a proportional-integral-derivative (PID) controller. The real-time software will be configured as a fixed-gain multivariable feedback law with parameters to specify gain matrices to implement controllers of either this form or having the form of an observer-based controller.

Finally, the control algorithm consists of real-time processing that occurs on the hub processor, control gain adaptation that can occur on the base station when present, and model-based design that can occur on the PC when present. We are also exploring the possibilities of our network model-based control design paradigm.

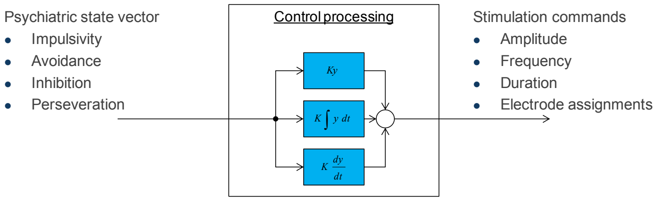


FIGURE 5: CONTROL ALGORITHM INTERFACE

V. Summary

We designed algorithms (include signal processing, decode and control) and software for an implantable neurological recording and stimulation system with closed-loop control. The software design uses an innovative and reprogrammable architecture that enables advancement of the state-of-the-art in closed-loop neural control while meeting the challenges of implantable system power constraints. Algorithm innovations include real-time signal processing, autonomous spike sorting, neuropsychiatric state decoding, and closed-loop control algorithms optimized for low-power autonomous operation while providing the capability for adaptation and learning when data and computing resources are available.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA), Biological Technologies Office (BTO), under contract number W911NF-14-2-0045. The opinions presented are those of the authors alone and not of DARPA, Draper Laboratory, or the Massachusetts General Hospital.

References

[1] Lewicki, M.S., “A review of methods for spike sorting: the detection and classification of neural action potentials,” Network: Computation in Neural Systems, Vol. 9, No. 4, 1998, pp. R53-R78.

[2] Gibson, S., Judy, J.W., Markovic, D., Spike Sorting. IEEE Signal Processing Magazine, Vol. 29, No. 1, p. 124, 2012.

[3] Gibson, S., Judy, J. W., Markovic, D., “Comparison of spike-sorting algorithms for future hardware implementation,” in Engineering in Medicine and Biology Society, 2008. EMBS 2008, 30th Annual International Conference of the IEEE, August 2008, pp. 5015-5020.

[4] Quiroga, R., Nadasdy, Z., Ben-Shaul, Y., “Unsupervised spike detection an, No. 8, 2004, pp. 1661-1687.

[5] Rutishauser, U., Schuman, E.M., Mamelak, A.N., “Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo,” Journal of Neuroscience Methods, Vol. 154, No. 1, 2006, pp. 204-224.

[6] Karkare, V., Gibson, S., Markovic, D.,. “A 75-μW, 16-Channel Neural Spike-Sorting Processor with Unsupervised Clustering,” IEEE Journal of Solid-State Circuits, Vol. 48, No. 9, 2013, pp. 2230-2238.

[7] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, “Simultaneous sparse approximation via greedy pursuit,” In Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP’05). IEEE International Conference on, vol. 5, pp. v–721, 2005.

[8] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” Signal Processing, IEEE Transactions on, vol. 54, no. 11, pp. 4311–4322, 2006.

[9] Tao Xiong, Siwei Liu, Yuanming Suo, Jie Zhang, Ralph Etienne-Cummings, Sang (Peter) Chin, Trac D. Tran, “Compressed Sensing of Multi-Channel Neural Recordings”, Proceedings of BioCAS, 2014.