A Closed-loop Compressive Sensing based Neural Recording System

Jie Zhang, Srinjoy Mitra, Yuanming Suo, Andrew Cheng, Tao Xiong, Frederic Michon, Marleen Welkenhuysen, Fabian Kloosterman, Peter S. Chin, Steven Hsiao, Trac D. Tran, Firat Yazicioglu, and Ralph Etienne-Cummings

Abstract

A Closed-loop Compressive Sensing (CS) based neural recording system is presented. Implemented using efficient digital circuit, this system is able to achieve >10 times data compression on the entire neural spike band (500 - 6KHz) while consuming only 0.83uW (@0.53VDD) additional digital power per electrode. When only the spikes are desired, the system is able to further compress the detected spikes by around 16 times. The entire system consists of an Application-Specific Integrated Circuit (ASIC) with 4 recording readout channels with CS circuits, a real time off-chip CS recovery block and a recovery quality evaluation block that provides a closed feedback to adaptively adjust compression rate. Since CS performance is strongly signal dependent, the ASIC has been tested *in vivo* and with standard public neural databases. Unlike other similar systems, the characteristic spikes and inter-spike data can both be recovered while guarantee >95% spike classification success rate. The complete signal processing circuit consumes <16uW/electrode.

1. Introduction

1.1 The need for efficient compression in neural recording systems

Neural recording microsystems are essential tools for neuroscientists to study the activity of the brain. These devices, consisting of one or more recording sites or electrodes, can be deployed within the cortex to collect neural action potentials (a.k.a 'spikes') generated by individual neurons. Studying these neural signals allows the neuroscientists to analyze the function and connectivity of brain circuits and their role in cognition and behavior [Mitra '13][Lopez '13]. Clinically, the neural recordings collected by the device can also be utilized to diagnose neuropsychological illnesses such as epilepsy, depression and traumatic brain injuries [Staba '02][Aziz '09].

The development of neural recording microsystems has continued to evolve in the past decades. The number of electrodes integrated into one device has increased from one [Hubel '59] to arrays that contains up to hundreds of electrodes [ShahrokhI '10]. However, given a cortical density of 100,000 neurons per mm^3 volume, the neural recording devices must be able to integrate even higher number of recording sites into a small volume to fully access the brain circuits [Braitenberg '91]. To prompt the next generation of neural recording device, the latest NeuroSeeker project funded by the European Commission aims to develop a neural probe with more than 10,000 electrodes [NeuroSeeker '13].

A major challenge that impedes massive electrode integration is the amount of data acquired by large number of electrodes in a device. Assuming each electrode is sampled at Nyquist rate of 20 kHz with at least 10 bits of resolution, the data collected by the system will exceed 200 Megabits-per-second (Mbps) as the number of integrated electrodes increases above 1000. This enormous amount of data poses significant challenges for the design of digital data readout interfaces. This is mostly due to the available power budget that can be dissipated close to the brain that does not result in a temperature rise that exceeds the safe limit of 1°C [Kim ‘06]. The challenge is even larger when neural probes with wireless transmission are considered. The limited weight of head-mounted devices that can be carried by small laboratory animals like mice, rats etc. (~10% of body weight) puts a severe restriction on the battery life of these devices.

The design challenges can be summarized below:

- *Readout Interface*: Most of the power in the recording circuits is needed to drive the output pads. Even with a wired connection, there are no standard cables that can carry this large amount of data and yet be lightweight and flexible for the animal to move in an unrestricted way.

- *Wireless transmission*: The same high data rate challenge occurs whenever wireless transmission of the data is desired. Furthermore, the current state-of-art wireless neural recording chips are limited in capacity, and consequently rarely allow more than a few Mbps to be transmitted [Abdelhalim ‘13].

Both of aforementioned issues are difficult to resolve without applying some kind of signal compression techniques prior to data readout or wireless transmission.

1.2 Prior works on neural signal compression

Many prior multi-electrode array designs rely on spike detection and windowing techniques to reduce transmission bandwidth [Mitra '13][Gosselin '09(a)][Chae '08][Gosselin '09(b)]. After a spike is identified through a threshold crossing detector, a small 1-2 ms long window around each spike is retained. This event based compression method achieves a decent Compression Rate (CR) for electrodes with sparse neuronal firing rates. When the aggregate firing rate of all detectable neurons is high (e.g. > 150 Hz), however, the CR is greatly reduced [Chen '13].

To further reduce the transmission bandwidth, wavelet transform based techniques have been proposed to provide compression for detected spikes [Oweiss '07][Kamboh '08]. While high CR can be achieved, the wavelet transform method requires significant amount of additional hardware. Its implementation consists of digital filters with additional memories that operate at a speed several times faster than the Nyquist rate of neural signals (~20 kHz). The complexity of the processing unit increases circuit area and on-chip power consumption, hence hinders the utilization of this technique to arrays of recording electrodes or silicon probes with large number of recording sites.

Compressive Sensing (CS) is a technique that gained popularity for compression of bio-signals due to its simple and power efficient mathematical operations using only additions and subtractions [Chen '13][Mamaghanian '11][Dixon '12][Charbiwala '13][Gangopadhyay '14]. Different from wavelet transform based techniques, data compression can be implemented using a few digital accumulators. Despite this advantage, previously implemented CS approaches only achieve limited CR before signal recovery quality degrades below an acceptable level for data analysis [Baluch '12]. The recovery quality of CS-based system heavily depends on the choice of sparsifying transforms (a.k.a. dictionary), through which the signal can be compactly represented. The limitation of previous systems is largely due to a less optimal choice of the dictionary.

Additionally, all previous compression systems are unable to measure the recovery quality of the signal since they have no knowledge of the original signal. Hence, the user has no understanding of how well the recovered signal resembles the real neural signal. Without such evaluation, these compression systems operate in open loop and provide no feedback to adjust the CR to balance the tradeoff between compression and recovery quality.

1.3 A Closed loop Compressive Sensing based compression system

In previous works, we have demonstrated that leveraging the unique shape of each neuron's spike, a signal dependent dictionary can be constructed and utilized to increase CR while maintaining high recovery quality in the CS framework [Zhang '14][Suo '13]. It is often the case that spike trains recorded on a single electrode contain spikes from several nearby neurons. Each neuron's spike has a characteristic shape and amplitude depending on its morphology and proximity to the recording electrode. Given that spike waveforms are generally stable over time, they can be used to learn a signal dependent dictionary to sparsely represent similar spikes recorded at the same electrode. We have also demonstrated that this method allows CS to achieve comparable CR and recovery quality as the wavelet transform based method, while using extremely efficient circuitry.

However, a signal dependent dictionary in the CS framework needs to be adaptable to accommodate changes in the neural signals that may occur during the recording. Without adaptation, the recovery quality would degrade over time because the learned dictionary can no longer represent spikes sparsely. To address this issue, we introduce a closed-loop Compressive Sensing neural recording system in this paper. The system includes: an application specific integrated circuits (ASIC) with 4 recording electrodes and compression circuits, an off-chip recovery algorithm that recovers the signal in real-time, and most importantly, a recovery quality evaluation method that provides adaptive closed-loop feedback to the ASIC for optimal tradeoff between CR and recovery quality.

In the main sections of the paper, we first introduce relevant background of Compressive Sensing and Dictionary Learning. We then describe the design of each component of the system and finally present a validation of the system using simulations and experimental data.

2. Background

We first introduce the basics of Compressive Sensing and the framework of Dictionary learning.

2.1. Compressive Sensing

Compressive Sensing originated as a theoretical framework regarding encoding and recovery of an S-sparse signal, x, of length N [Candes '06][Donoho '06]. A signal is S-sparse if it can be well approximated by its largest S coefficients in a certain transform domain (or a 'dictionary'), where $S \ll N$. The S-sparse signal, x can be encoded by a small measurement vector, y, of length M, such that:

$$y = Ax \qquad (1)$$

where $S < M \ll N$, and A is a sensing matrix of dimension $M \times N$. The CR achieved in this case is N/M. However, recovering x, given y and A, is not trivial because this system of linear equations contains more unknown variables than equations. Fortunately, considering matrix A satisfies the

Restricted Isometry Property (RIP) and x is S-sparse, this underdetermined problem can be solved and x can be recovered exactly with extremely high probability from y using optimization methods [Candes '06].

RIP is the key factor to determine the optimal choices of sensing matrices. RIP describes how well the distance of S-Sparse signal can be preserved after the projection using sensing matrix A. Many matrices, such as the random Gaussian, random Bernoulli, and Partial Fourier matrices all satisfies the RIP universally with a small number of M. Choices of sensing matrix can be determined based on specific applications and desired performance tradeoffs.

2.2. Dictionary Learning

The number of samples, M, required to successfully recover x is proportional to the sparsity, S, of the signal represented using a dictionary. Therefore, a desired dictionary should be able to represent x using as few coefficients as possible to improve CR. Various dictionary learning methods can be used for this purpose [Lewicki '00][Aharon '06][Engan 00']. Given L training signals $X = \left\{{x_l}_{l=1}^{L}\right\}$, the dictionary learning algorithms find a dictionary $\boldsymbol{D}$ that can represent the training signals using S-sparse signal $V = \left\{{v_l}_{l=1}^{L}\right\}$. In other words, it solves the optimization problem:

$$argmin \sum_{l=1}^{L} \|x_l - \boldsymbol{D} v_l\|_2^2 \quad such\ that\ \|v_l\|_0 \leq s, 1 \leq l \leq L \tag{2}$$

where S is the bound on the l_0-norm of S-sparse signal v_l. It sets the bound on the number of non-zero coefficients for every v_l.

3. Methods

Figure.1. presents the blocks of the entire system. We also describe the system design in detail in this section. First, we describe the ASIC which consists of analog preprocessing blocks, ADCs and the CS compression block. Next, we present our off-chip Dictionary Learning and Compressive Sensing recovery algorithms. Finally, we conclude the section describing our adaptive mechanism of the close-loop feedback between on-chip and off-chip blocks.

3.1. ASIC

As shown in Figure.1., the ASIC contains 4 neural recording channels with corresponding CS circuits. The signal is first conditioned by the analog front end and sampled by a shared 10 bit Successive Approximation Register (SAR) ADC. The ASIC can be configured to operate in either Dictionary Learning mode (DL) or Compression Mode (CM). During DL mode, the CS circuit is bypassed and the raw waveforms are transmitted to allow the off-chip dictionary learning algorithm to construct a dictionary. Then the chip is switched back to CM, where the raw data is condensed by the CS block. For a large arrays of electrodes, dictionary learning can be performed per group of electrodes to avoid large data transmission during a small period of time.

3.1.1. Analog Front End and ADC

The Analog Front End (AFE) consists of two gain stages. In the first stage, a capacitive coupled Instrumentation Amplifier (IA) is used. The output of the IA passes through a band-pass filter to extract neural spiking signals (500Hz– 6kHz). A Programmable Gain Amplifier (PGA) is used at the second stage to provide additional gain before the signal is sampled by the ADC. The AFE has an Integrated Noise of 3.1uVrms (500 – 6kHz band) and CMRR of 75dB, while providing a configurable gain of 230-6K. The Successive Approximation Register (SAR) ADC, operating at 80KHz, is used to sample the conditioned analog signals from all four recording electrodes. Operating at VDD of 1.8V, the AFE and ADC together consume 15µW per electrode.

3.1.2. Compresive Sensing Block

In CM mode, The CS block can be further configured into two sub-modes of operation: either the entire band-passed neural signal or only the spikes are compressed. When configured to compress the entire neural signal, the CS block preserves the fidelity of the spikes as well as the inter-spike signals. If only spikes are desired, the compressed output is only produced when a spike is detected by a threshold crossing detector applied to the absolute magnitude of the signal. 64 samples of the detected spikes are kept for compression.

The CS block implements the linear operation of equation (1): $\boldsymbol{y} = \boldsymbol{Ax}$. This equation can also be written

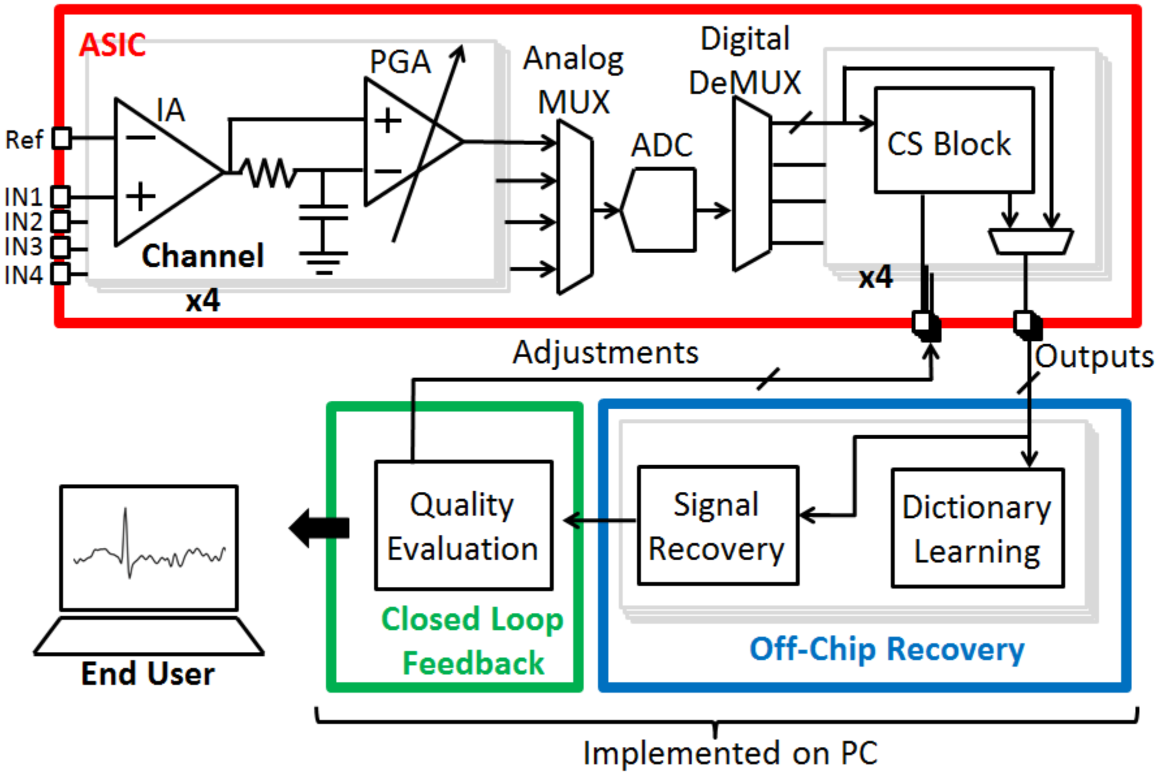


Figure 1. Overview of the Neural Recording ASIC, Off-chip recovery system and the Closed-Loop Feedback linking two sub-systems.

as a system of linear equations:

$$\begin{aligned} y_1 &= A_{1,1}x_1 + A_{1,2}x_2 + \cdots + A_{1,N}x_N \\ y_2 &= A_{2,1}x_1 + A_{2,2}x_2 + \cdots + A_{2,N}x_N \\ &\vdots \quad \vdots \quad \vdots \\ y_M &= A_{M,1}x_1 + A_{M,2}x_2 + \cdots + A_{M,N}x_N \end{aligned} \qquad (3)$$

where, $x_1 \dots x_N$ are the digitized neural signal from ADC at discrete time 1 to N, $y_1 \dots y_M$ are entries of compressed sample y of length M ($M \leq N$), and **A** is the sensing matrix of size $M \times N$. In our design, matrix **A** is a random Bernoulli matrix, which can be configured by the user. Among matrices that satisfies RIP requirement, random Bernoulli matrices are the most optimal for hardware implementation as their entries are either +1 or -1. Therefore the system of equation (3) can be implemented using M digital accumulators. Depending on the corresponding value of A, the accumulators either adds or subtracts digitized signal x_i from the value of the accumulator to generate y_i. Other matrices, such the random Gaussian and optimized sensing matrices all contains fractional entries [Elad '07][Sapiro '09]. Thus implementing equation (3) with these choices requires the use of digital or analog multipliers in addition to accumulators. These additional components consume a large amount of chip area. For example, an implementation of a Gaussian matrices using M-DAC occupies around 0.6 mm^2 [Gangopadhyay '14], whereas our digital implementation of a Bernoulli sensing operation only occupies 0.11 mm^2.

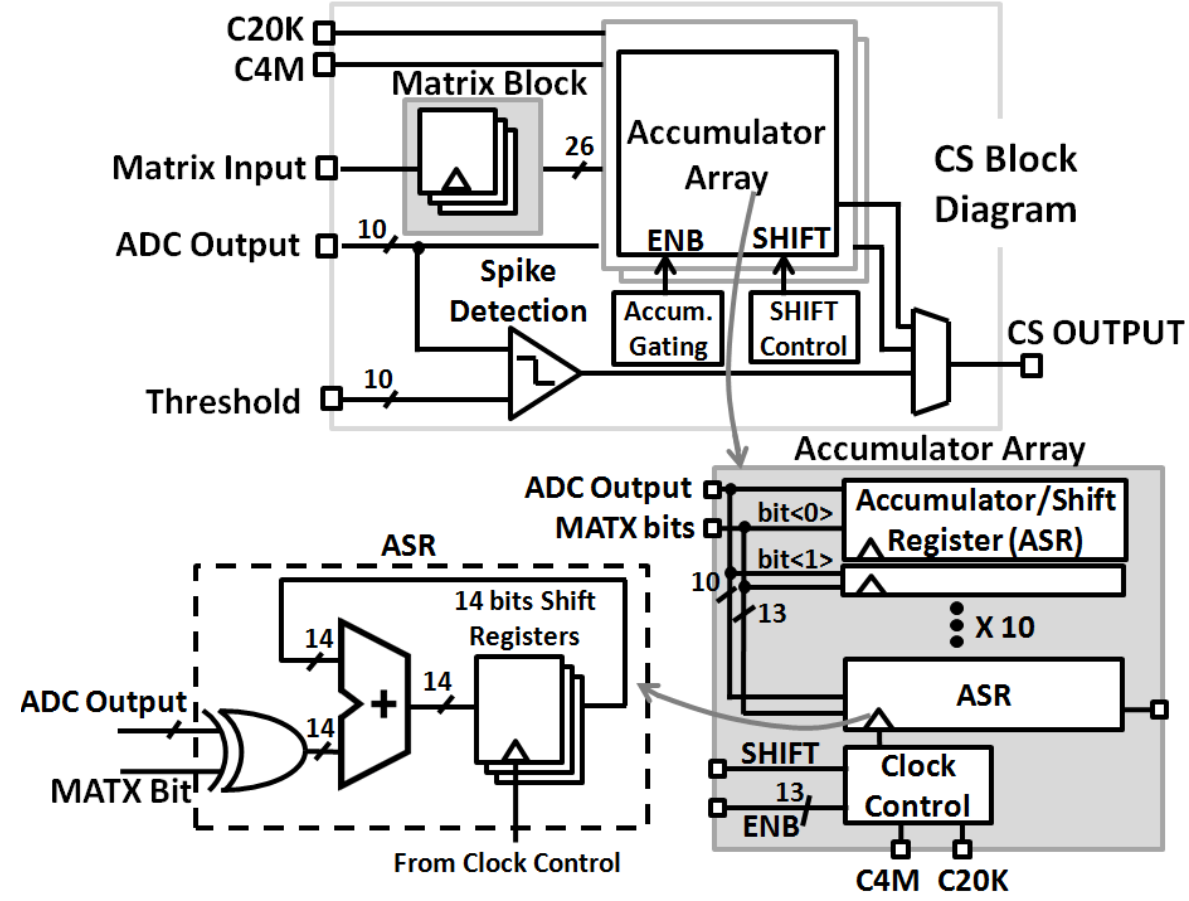


Figure 2. Architecture of the CS Block

The CS block uses arrays of accumulator shift-registers (ASRs) to implement the matrix multiplication of equation (1), shown in Figure 2. The accumulations are clocked at signal Nyquist rate of 20 kHz (C20K). The Matrix block, shared across all the channels, contains registers to hold one row of a random Bernoulli matrix. Their values are updated off-chip at every Nyquist period. Depending on the value of a particular matrix entry (either 1 or 0), the corresponding ASR either adds or subtracts the current digitized signal from the accumulated value. Each ASR can be disabled by applying clock gating to control the compression rate (CR=N/M). To avoid implementation of extra registers to buffer the data for transmission, a 4MHz (C4M) clock is used to shift the data in the ASRs to the output pin near the end of each accumulation cycle. Vector $\boldsymbol{y}$ is generated every N clock cycles. N can be configured to be either 128 or 64, depending on the operation mode, corresponding to signal length of 6.4ms and 3.2ms.

To conserve power, when the ASIC is configured to compress only the spikes, clock gating is applied to the ASRs and the matrix block so that they remain inactive until the Spike Detection block registers a threshold crossing event. A digital FIFO is used to buffer a variable length of pre-trigger samples (up to 15 samples) before a threshold crossing event. After the compressed vector y is sent off-chip, the CS block becomes inactive again until the next threshold crossing event. The spike detection information is also used when the ASIC is configured to compress the entire band-passed neural signal. It informs the off-chip recovery block how many spikes have occurred and their peak locations within the signal segment [Zhang '14]. A Micrograph of the ASIC is shown in Figure.3.

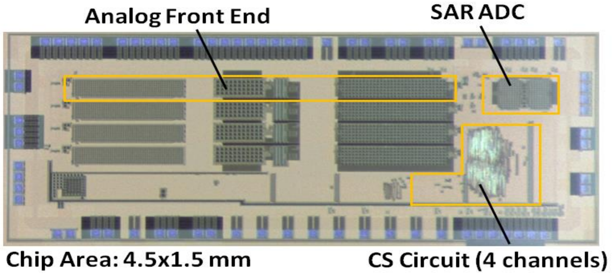


Figure 3. Micrograph of the ASIC

3.2. Off-Chip Dictionary Learning and Signal Recovery

The off-chip recovery algorithms consist of two blocks, the Dictionary Learning block and the signal recovery block. Both of these blocks are implemented using MATLAB.

3.2.1. Dictionary Learning

Operating in the Dictionary Learning Mode (DL), the ASIC bypasses the CS block and outputs the uncompressed neural signal. The raw signals form a training signal set that is used to learn a dictionary. The most straight forward method to construct a dictionary is to use detected neural signals to form bases in this dictionary [Suo '13]. Alternatively, dictionary learning methods such as K-SVD [Aharon '06] can be utilized to train a dictionary given a set of training spike waveforms [Zhang '14]. As described in section II, like other dictionary learning method K-SVD algorithm finds a dictionary, D, through iterations to minimize error between the training data and its corresponding sparse representations using D. Analysis has shown that dictionaries created with raw spikes result in slightly better reconstruction performance when tested using synthetic neural database (with various amount of additive noise) [Suo '13][Quiroga '04]. On the other hand, the K-SVD trained dictionary does well when evaluated using an in-vivo recording database [Henze '00][Suo '13][Suo '14].

Here we implement the K-SVD algorithm to learn the dictionary, due to its fast computation speed and superior performance over dictionaries created with raw spikes. For example, implemented using MATLAB on a PC with Intel Core i7 and 16 Gbyte of RAM, K-SVD algorithm takes approximately 0.01s to compute a dictionary of size 64 by 100, using around 300 observations of different spikes. The size of the dictionary and training data size could vary depending user's preferences.

3.2.2. Compressive Sensing Recovery

After a dictionary is trained, the ASIC switches back to the CM mode and outputs the compressed vector y. From the compressed measurement, y, the signal can be reconstructed by solving a L1-minimization problem:

$$argmin_x \|x\|_1 \; such \; that \; y = Ax \qquad (5)$$

We solve (5) using Matching pursuit methods due to their efficient computation time. A detailed discussion on the recovery method and signal model is provided here [Zhang '14]. The average computational time for recovery is around 1.3 ms if only spikes are reconstructed and 2 ms if the entire neural signals are reconstructed. This suggests that the system can recovery the around 700 spikes per second for real time applications. For large array of electrodes, multiple systems could be used to handle the recovery or FPGA implementation of the recovery algorithm could be developed to speed up the recovery.

We measure the recovery performance of a spike train using the Signal to Noise and Distortion Ratio ($SNDR$). In here, we add a subscript x to derive a notation $SNDR_x$ to represent $SNDR$ measured between the original signal and the recovered signal:

$$SNDR_x = \frac{1}{T}\sum_{i=1}^{T} 20log\frac{\|x_i\|_2}{\|x_i - \hat{x}_i\|_2} \qquad (6)$$

where x_i is ith spike belonging to a spike train having T spikes, and $\hat{x}_i$ is the reconstructed spikes from compressed measurements. $SNDR_x$ is a purely theoretical estimate. It has been used by other authors to verify the validity of their compression and recovery approaches using known signals [Chen '13].

3.3. Closed Loop Feedback

An obvious disadvantage of $SNDR_x$ is that it requires the knowledge of the original signal, which is not available when the ASIC is generating the compressed measurements. The failure to address this problem makes previously reported neural signal compressive sensing systems impractical for real recording application. Without a Quality Evaluation (QE) block, the users have no means to quantify the performance and adjust compression rate for optimal tradeoff between recovery quality and compression. Furthermore, the QE block is essential in our system where a learned dictionary is used. QE block can detect the case when the existing dictionary can no longer represent the neural spike trains and then switch the ASIC back to DL mode where a new dictionary will be learned. No data is lost while the system is in the DL mode, since raw data is transmitted in this mode. As we shall demonstrate in an *in-vivo* experiment, the system only need to switch to DL mode once for around 2 minutes during a two hour recording session.0020

As shown in Figure.1., QE block examines the quality of the recovered signal and provides feedbacks to the ASIC to adjust CR or to switch between DL mode and CM mode. QE block is also implemented using MATLAB.

Due to the inability to calculate $SNDR_x$, the QE block calculates Signal to Noise and Distortion Ratio measured in compressed domain ($SNDR_y$) as the metric for recovery performance. $SNDR_y$ is defined as:

$$SNDR_y = \frac{1}{T}\sum_{i=1}^{T} 20log_{10}\frac{\|y_i\|}{\|y_i - \hat{y}_i\|} \qquad (7)$$

$$\hat{y}_i = A\hat{x}_i$$

where y_i is the CS measurement of ith spike within a spike train containing total of T spikes, and $\hat{y}_i$ is the CS measurements estimated from the recovered spike $\hat{x}_i$. When a signal is not well reconstructed, the reconstruction error can also be reflected in the CS measurements after a linear mapping using the sensing matrix, A. In the Experiments section, we shall demonstrate the correlation between $SNDR_x$ and $SNDR_y$.

The QE block calculates the moving average and the standard deviation of the $SNDR_y$ across many time intervals. It can initiate feedback to the ASIC to increase the compression rate or to learn a new dictionary if the measured $SNDR_y$ decreases below a tolerable threshold set at a few standard deviation from the moving average.

4. Experiments and Results

In this section, we describe the experiments we conducted to validate each part of the system. First, we used a dataset from fifteen week long multi-electrode recording experiment to characterize the recovery performance. This data is essential to validate the efficacy of the system under different noise condition and over an extended period of time. Here we also show the correlation between performance metrics $SNDR_x$ and $SNDR_y$. Second, we used a dataset from a two hour tetrode experiment to evaluate the closed-loop feedback system and demonstrate the dynamic CR evaluation and dictionary updates. We further tested the ASIC's functionality by deploying it a recording experiment conducted on an awake Rhesus Macaque. Finally, we also characterized the system performance using a standard database in order to compare our system with previous published works.

4.1. Off-chip Recovery Performance Characterization

4.1.1 Experiment Setup

Data from a fifteen week long multi-electrode recording experiment is used to characterize the off-chip recovery performance. The close relationship between performance metrics $SNDR_x$ and $SNDR_y$ is also validated using this experiment. Data from this experiment is ideal for recovery performance evaluation since they contain recordings collected at many different electrodes with different Signal to Noise Ratio (SNR). Additionally, a few electrodes also detect multi-neuron activities. Both signal compression and recovery are carried out using offline algorithm implemented using a PC. Since the compression block on the ASIC is implemented using digital circuits, it does not introduce additional noise to the recording.

Therefore its performance can be exactly modeled by an offline algorithm. The performance of the ASIC is characterized through an *in-vivo* experiment discussed later in the paper.

In this fifteen week long recording experiment, a high density recording array containing 32 electrodes on a single shank (70 µm wide) is implanted in the thalamus of a rat's brain when it is under anesthesia. The scientific aim of the experiment is to examine the long-term recording SNR of an implanted silcon probe. We acquired one minute of raw data from the implanted electrodes every week for fifteen weeks. The recorded signals are digitized at 25KHz and filtered at 500-5KHz. In week 0, spikes are observed at 13 out of 32 electrodes. The relative position of the electrodes are shown in Figure 4 (b). Electrodes 1, 2, 7, 8, 9 and 16 have contact diameter of 10 µm, while electrodes 5 and 10 to 15 have contact diameter of 5 µm. During 15 weeks of recording, the electrode average SNR is found to be stable and does not suffer major loss over time.

For each recording electrode, we used 20% of the extracted spikes (around 50 - 120 spikes) to train a representation dictionary with K-SVD method, while using the remaining 80% as test signals to evaluate CS recovery performance. The raw recording first goes through an offline spike detection block which extract the spikes after their amplitude exceed a pre-set threshold at around 4 standard deviation above and below the average signal amplitude of the dictionary training data. For each detected spikes, 64 discrete samples are retained around a spike corresponding to 2.6ms temporal duration. In week 0's recording, electrode 8 records the spikes from two neurons, while the rest of the electrode only have one distinguishable spike cluster.

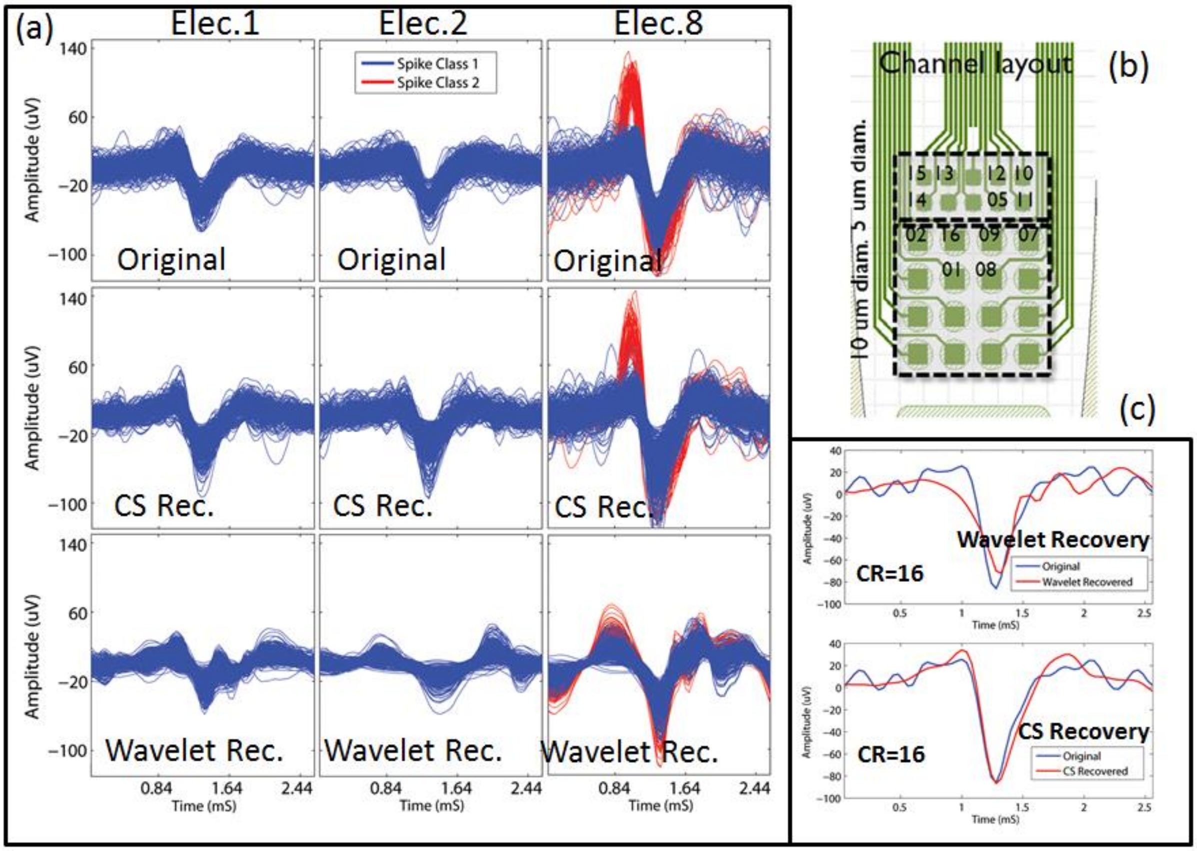


Figure 4. (a) Example of the original recorded neural waveforms and recovered waveforms using CS and Wavelet Transform method (week 0). (b) Layout of the electrode on the silicon probe. (c) Zoomed in views of CS and Wavelet recovery.

We compress each spike by multiplying it with a random Bernoulli sensing matrix of size M x 64, where M is the number of compressed samples. For comparison, we also present recovery results using wavelet transformed based recovery. In this method, the extracted spike first undergoes a wavelet transform. Wavelet components at M biggest locations determined using the training data are retained and used to reconstruct the spike. The wavelet used is Daubachies-8 wavelet, which is a standard wavelet choices for compression [Bulach '12]. Figure 4. illustrates the electrode layout on the silicon probe, the original signal and the signals recovered using CS and wavelet at three of the 13 electrodes at week 0.

As performance metrics, we first compute the $SNDR_x$ of the spike train at every electrode. In addition, for each recovered spike, we also compute the difference of its amplitude at its main trough compared to the original spike. To account for the variation of the CS recovery over the choice of sensing matrix, we compress and recover each spike train using 20 different randomly generated Bernoulli matrices. We then average $SNDR_x$ over the entire 20 trials to acquire a single measurement of $SNDR_x$ for the spike trains collected at that particular electrode.

To analyze the spike recovery quality under different noise levels, we calculate Signal to Noise Ratio (SNR) for each type of spikes. SNR is computed as:

$$SNR = \frac{Signal\ Amplitude}{Peak\ to\ Peak\ Amplitude\ of\ Noise\ Floor} \qquad (8)$$

Peak to peak amplitude of the noise floor is taken as six times the standard deviation of the recorded signal after spikes are removed, spanning ~99.7% of normally distributed noise data [Ludwig '09]. Spike clusters with a SNR of 1.1 or greater are considered to be discriminable units [Ludwig '09].

For electrode No.8, where distinguishable multi-unit activities are seen, we examined spike clusters' distance in the Principal Component subspace after they are reconstructed by CS and wavelet method. The larger the cluster distance, the easier it is to cluster the spikes. From the original spikes, we observed that there are two types of spikes at electrod No.8, class C1 and C2. They contain 608 and 86 spikes respectively. In this analysis, we first use the same training spikes to learn a dictionary for CS reconstruction. The spikes are then compressed and reconstructed using CS and Wavelet method for various CR. We perform the Principal Component Analysis (PCA) on the reconstructed spikes. Finally, we calculate the cluster mean of reconstructed C1 and C2 in the subspace spanned by their first two Prinical Components. Figure 6.b. shows the original spikes cluster C1, C2 and their means.

4.1.2. Results on CS recovery quality vs. Wavelet recovery quality

Figure 5.a. shows the SNR for spikes at each electrode. Figure 5.b. and 5.c. illustrate the recovery quality of CS and wavelet method for week 0's data. In this case, the length of compressed samples (M) is set to be 4, corresponding to CR of 16. In terms of both $SNDR_x$ and recovery error at the main trough of the spike, CS performs better or comparable to wavelet recovery method across all electrodes with different SNR.

Figure 6.a. presents the PCA cluster distance for C1 and C2 spike clusters when spikes are reconstructed across different CR by CS and wavelet method. The CS reconstructed clusters maintain very high separation even at CR of 32, when only 2 CS samples are retained to reconstruct the spike. This is because CS method only uses the sparse dictionary atom from either C1 or C2 to reconstruct a spike. As long as it can choose the correct atom during reconstruction, high cluster distance is guranteed. On the other hand, wavelet reconstructed cluster distance starts to decrease when CR increases above 9. This is because as more wavelet coefficients are removed, the clusters lose their discrimintive features. Figure 1.b. show the scatter cluster for the original spikes, while Figure 1.c. show the cluster of CS and wavelet reconstructed spikes at CR = 32.

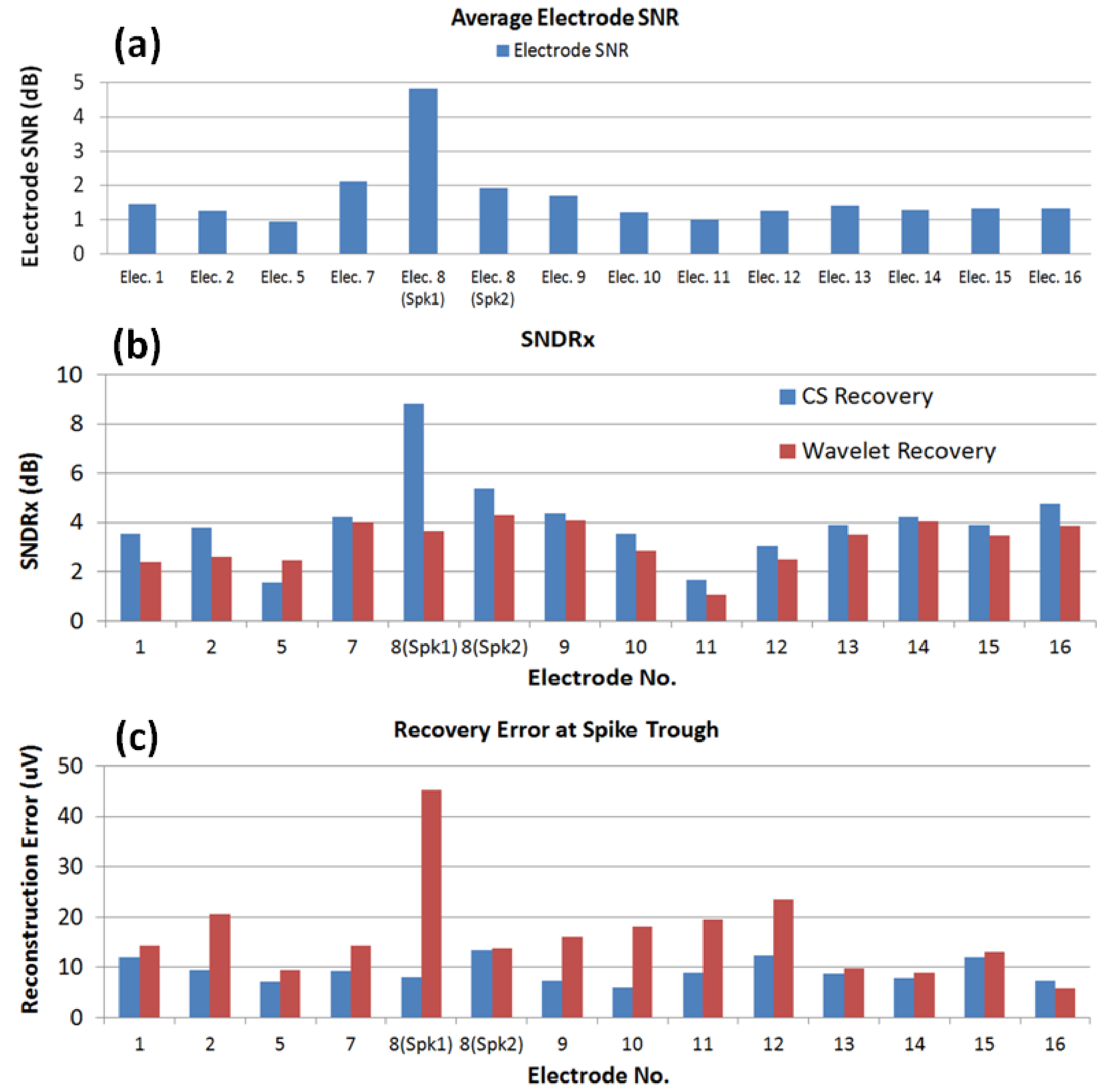


Figure 5. (a) SNR for every types of spike seen at electrodes at week 0. (b) $SNDR_x$ of the recovered spikes. (c) recovery error at main trough of the spikes.

The recovery results from this dataset suggest that CS performance is comparable to that of the wavelet transform based method. For clustering, the advantage of CS is more apparent at high CR (>9). The result from the clustering experiment is consistent with a similar spike classification experiment described in our previous publication [Zhang ‘14]. In terms of hardware power and area efficiency, a 6 Level wavelet transform would require around 32708 transistors [Oweiss '07]. On the other hand, to achieve CR=16, CS method only needs 4 digital accumulators of 13 bits, assuming each digitized data has 10 bits resolution. This corresponds to only 2496 transistors if the D-Flip Flop has 20 transistors and Full-Adder has 28 Transistors [Zhang '14]. Hence, the power consumption and the area required to implement the CS system is much more efficient than the wavelet method to achieve comparable performance. A quantitative comparison on hardware efficiency is elaborated in our previous work [Zhang '14].

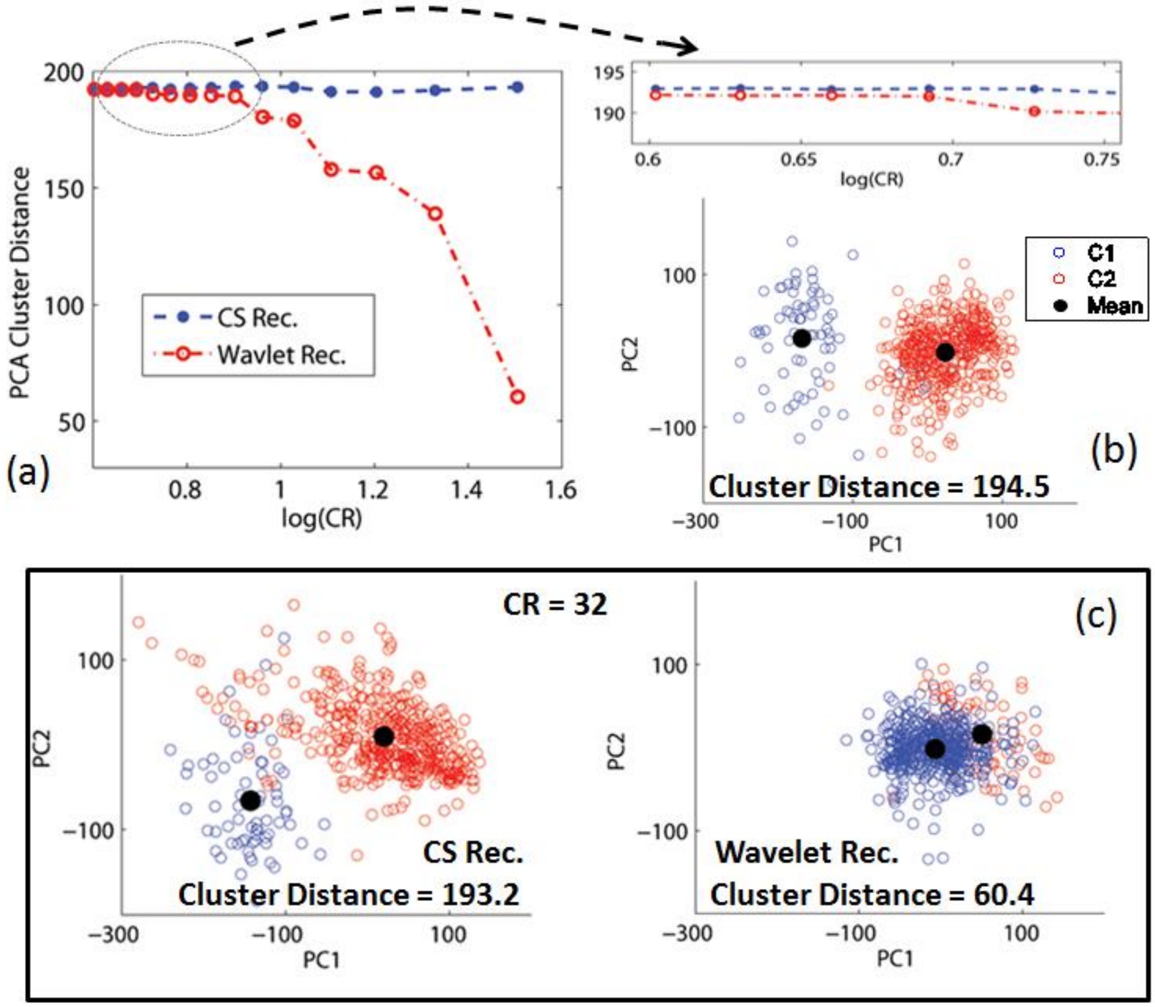


Figure 6. (a) PCA cluster distance with respect to compression rate. (b) Original spikes’ clusters, C1 and C2. (c) C1 and C2 cluster when the spikes are reconstruction using CS and wavelet method at CR = 32.

4.1.3. Correlation between $SNDR_x$with $SNDR_y$

To verify that $SNDR_y$ can effectively be used as a metric to measure recovery quality, we must show that these two metrics are highly correlated. We calculate the $SNDR_x$ and $SNDR_y$ for the spike trains recording at every electrodes from week 0 to week 15, shown in Figure 7. On average, a one minute recording from each electrode, every week contains around 500 spikes (T=500). A regression analysis between $SNDR_x$ and $SNDR_y$ results in $F(1,206) = 2.59 \times 10^{31}$ and $P < 10^{-9}$, suggesting a strong linear relationship between $SNDR_x$ and $SNDR_y$. Therefore, we could use $SNDR_y$ as a alternative metrics to evaluate signal recovery quality.

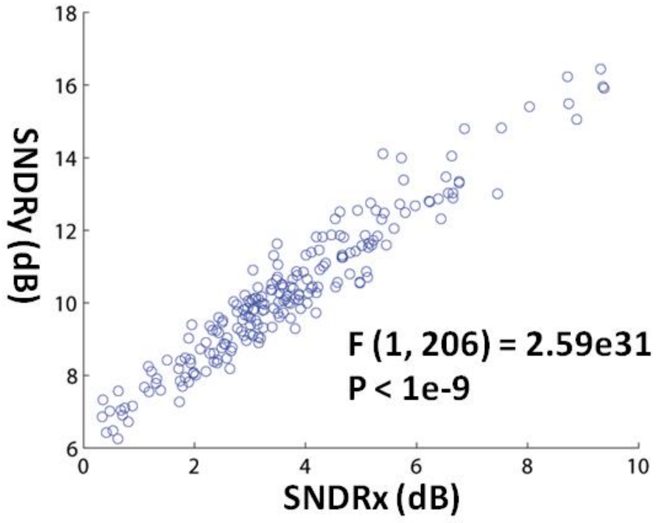


Figure 7. $SNDR_x$and $SNDR_y$ values for all the recording electrode over 15 weeks of recordings

4.2. Quality Evaluation Block and The Closed-loop Feedback Characterization

4.2.1. Experiment Setup

We use the data from a two hour tetrode recording experiment as well as a synthetically generated spike train to characterize performance of the closed loop feedback block. The tetrode recording was acquired with digital Lynx (from Neuralynx). In this experiment, a micro-drive array carrying tetrodes was chronically implanted on a rat. The recording is from the CA1 of the hippocampus. For the first hour of the experiment, the rat is sleeping inside of a box. Then it is placed on to a treadmill to perform running tasks. To detect spikes, a threshold is set at 100uV, and 32 samples of around the spikes are retained after a threshold crossing event. Similar to the previous experiment, the compression and recovery are all completed offline without using the ASIC, as the offline model is an exact replication of the ASIC functionality. In this continuous two hour experiment, we observed activities of different neurons at different time intervals. Hence we can evaluate the performance of the QE block when a new types of spikes are detected that were not included in the dictionary.

For each tetrode, the spikes collected during the first two minutes are used to train a dictionary. Then we compress the spikes using a random Bernoulli matrix of size 4×32, corresponding to CR of 8. Each spike is recovered using the same recovery method mentioned in previous section. $SNDR_y$ is calculated by the QE block at every minutes interval. We also computed $SNDR_x$, which is a truth recovery quality metric. The moving average and standard deviation of $SNDR_y$ is also calculated.

We computed two trials of compression and recovery: In the first trial, we compress and recover all the spikes collected on one of the tetrodes using the initially learned dictionary. In the second trial, QE blocks triggers the DL mode when $SNDR_y$ decreases by more than 4 standard deviation compare to its moving average. The recovery system then recovers the subsequent spikes in CM mode using the newly learned dictionary together with the initially trained dictionary.

In addition to the tetrode data, we have also created a synthetic spike train to further demonstrate improvement in recovery quality after closed loop feedback and dictionary retraining. These spikes, taken from the Leicester neural database [Quiroga '04], originates from three different neurons. Their shapes are shown in Figure 10 (b). 5000 spikes are drawn randomly from three spike clusters and placed 10 ms apart to form the synthetic spike train of 50 s in duration. In the first 10 s, only neuron 1 fires. When then followed by firing of neuron 2 between 10s to 20s. Then neuron 1 and 2 both fire between 20s to 30s before neuron 3 fires between 30s to 40s. Finally, all three neurons fire between 40s to 50s. The CR is set to 16, compressing 64 samples down to 4 samples. In the recovery experiment, we first recover the signal using dictionary learned from only spikes 1. We then repeat the experiment to allow dictionary re-training after observing a decrease of more than 4 Standard-Deviation in the $SNDR_y$.

4.2.2. Closed loop feedback experiment results

Figure 8.a. shows the $SNDR_x$ and $SNDR_y$ measured at every minute throughout the duration of the experiment. In Figure 8.b, the grey dots represent the first and second Principal Components of the all the spikes collected during the experiment, after a Principal Component Analysis (PCA). The red dots are the spikes used for dictionary learning. They are collected in the first 2 minutes of the experiments. The training data covers only a portion of the PCA space where the spikes occurred. The learned dictionary using the training data is shown in Figure 8.e. As we could expect, if a spikes falls into the PCA space overlapped by the training data, it can be well recovered. On the other hand, if spikes fall outside of this region then most likely they cannot be recovered with great accuracy since the their shapes are different than the dictionary.

In Figure 8.a., the recovery quality measured by $SNDR_x$ and $SNDR_y$ stays constant above 8 dB and 15 dB for the time intervals before 60 minutes. This is because the spikes detected during this time can be well represented by the learned dictionary. Examples of these spikes' PCA plot are shown in Figure 8.c., where the black dots represent the spikes detected between 50 and 51 minutes. Most of the black dots fall onto the PCA space covered by the training data, and therefore have similar shape than the dictionary. However, at around 61 to 62 minutes, when the rat is first placed on the treadmill, a lot more spikes are detected that do not fall into the PCA spaced covered by the training data (Figure 8.d). These spikes have different shapes compared to the training data and the learned dictionary, as shown in Figure 8.f. Therefore, they cannot be recovered accurately using the learned dictionary. There exists a significant amount of mismatch between the original and the recovered signal. Both $SNDR_x$ and $SNDR_y$ experience a decrease of 4.5 dB and 3.5 dB, more than 10 and 8 standard deviation from their corresponding running averages.

Figure 9. demonstrates the scenario when DL mode is trigged at 61 - 62 minutes interval when the measured $SNDR_y$ decreases by more than 4 standard deviation from its running average. Spikes from this time interval are used to learn a new dictionary. The new dictionary items are added to form a dictionary that is used to recover signal after 62 minutes, shown in Figure.9.c. When the system switch from DL mode back to CM mode, we see an increase of both $SNDR_x$ and $SNDR_y$ at 62 to 63 minutes compare to $SNDR_x$ and $SNDR_y$ measurement in Figure 8. Figure 9.d. shows the spikes appeared at 62 to 63 intervals can now be well recovered using the newly learned dictionary.

The results of the experiment using synthetic spike train is plotted in Figure.10.c. Without dictionary re-training, the $SNDR_y$ decreases around 15 to 25 dB whenever a new type of spikes appears. On the other hand, after re-training the dictionary at around 11s and 31s, the $SNDR_y$ remain constant around 28 dB when new types of spikes appear. The recovery dictionaries are shown in Figure.10.a.

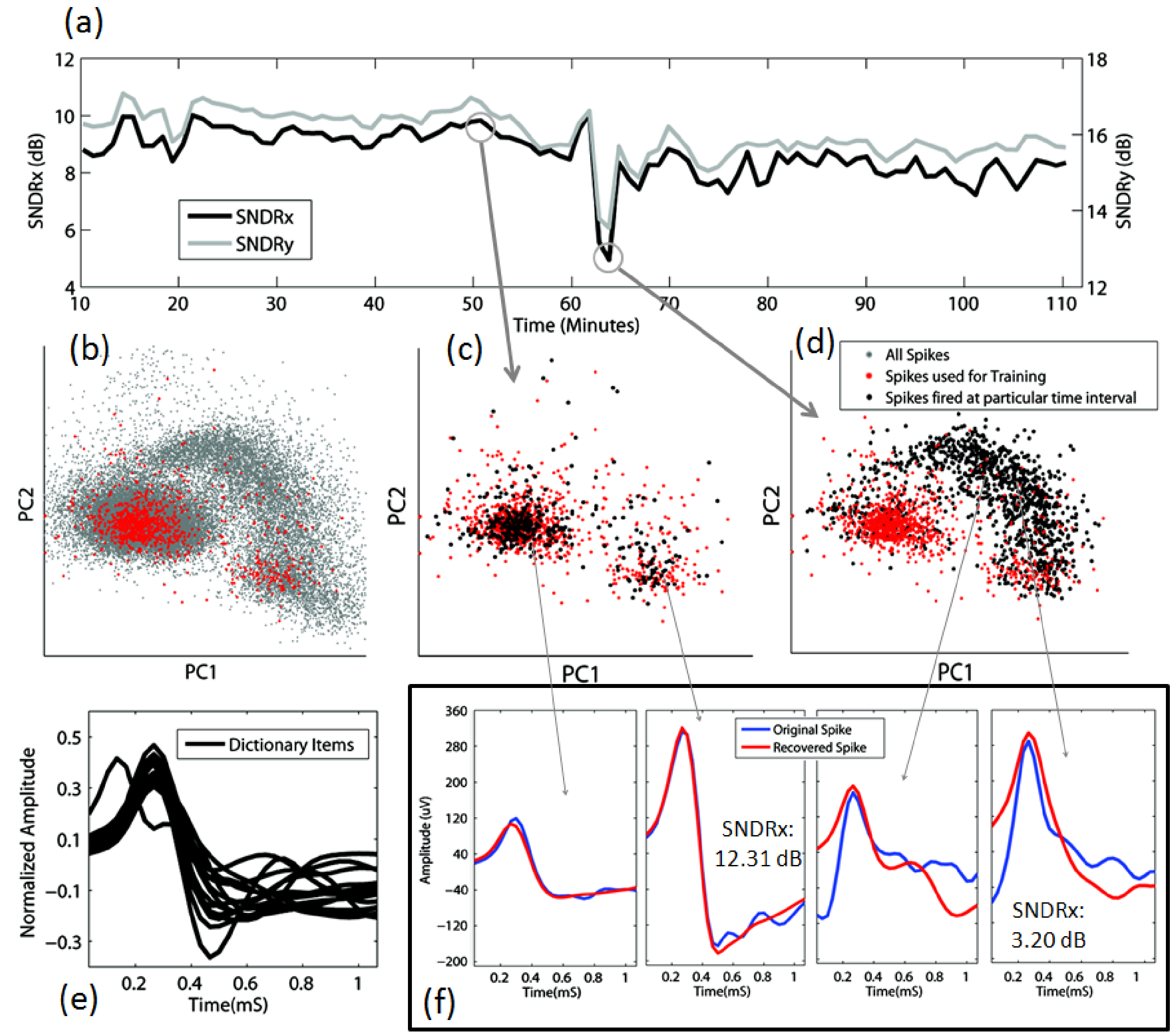


Figure 8. (a) $SNDR_x$and $SNDR_y$of the recovery over a 2 hour experiment. (b) Gray dots: PCA results of all the spikes collected in 2 hour experiment. Red dots: spikes used for dictionary learning. (c) Black Dots: spikes recorded around 50 – 51 minutes interval. They are well recovered since the their shapes do not vary too much from the dictionary. (d) Black Dots: spikes recorded around 62-63 minutes interval. A new type of spike starts to appear. As its shape vary significantly from the dictionary, the recovery results deteriorate, shown by a decrease of $SNDR_x$and $SNDR_y$. (e) The trained dictionary (f) Temporal view of original spikes and recovered spikes at different time intervals.

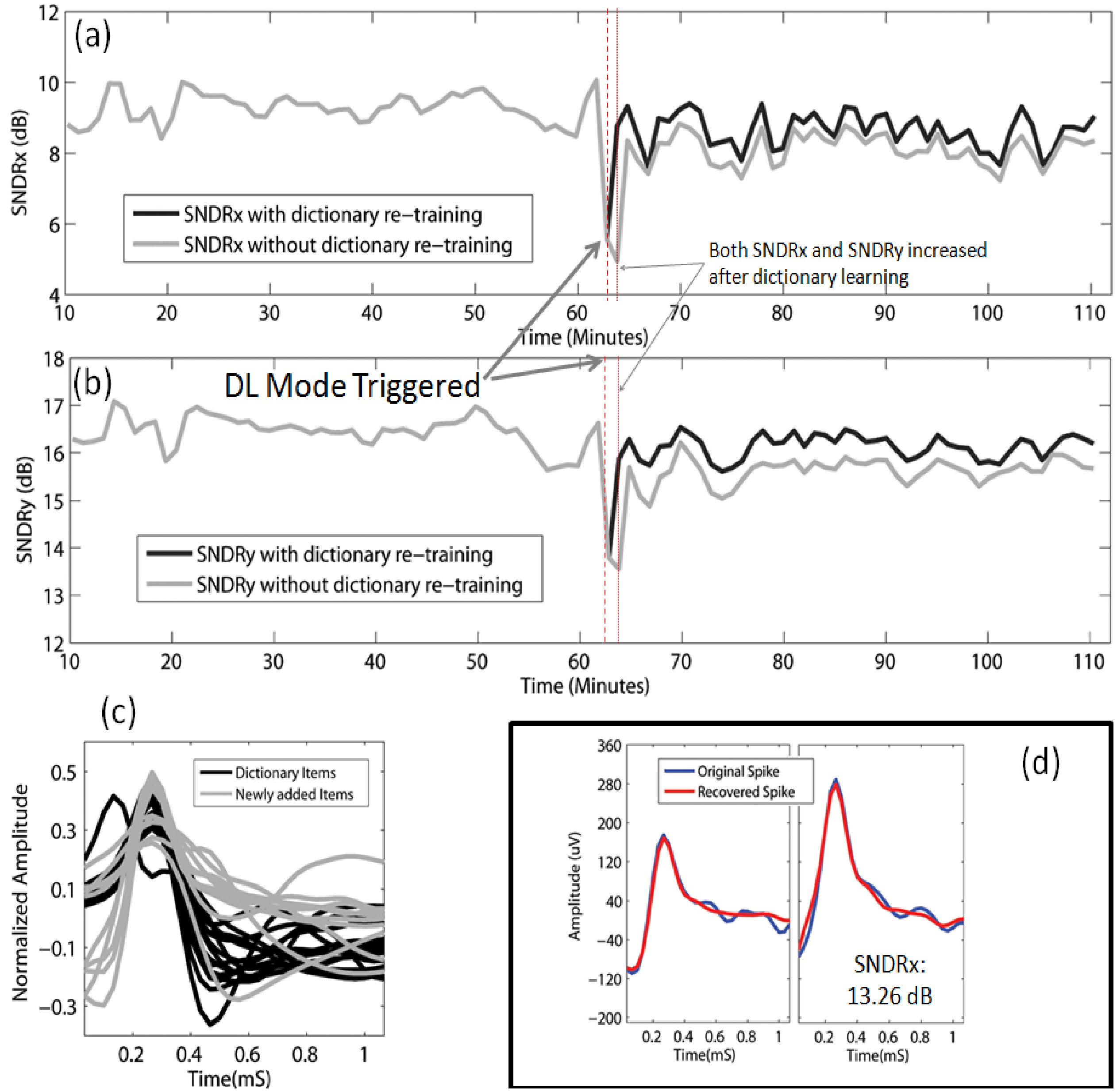


Figure 9. (a) $SNDR_x$of the a 2 hour experiment with and without dictionary retaining (b) $SNDR_y$ of the a 2 hour experiment with and without dictionary retaining (c) The dictionary after dictionary re-training. (d) Temporal view of original spikes and recovered spikes at 62-63 minutes interval.

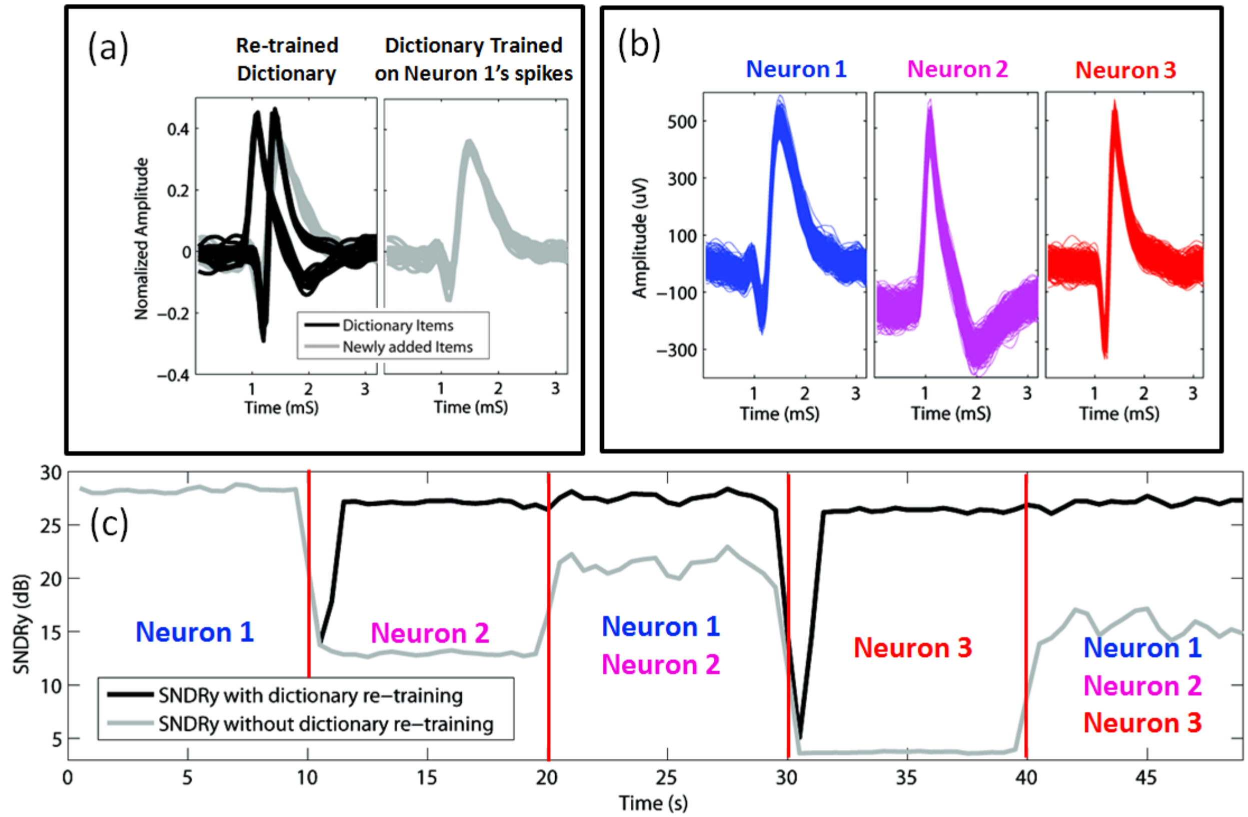


Figure 10. (a) Dictionary trained on Neuron 1's spikes and the re-trained dictionary after Neuron 3 fires. (b) Three types of spikes from different neurons. (d) $SNDR_y$ of the experiment with and without dictionary re-training.

4.3. In-vivo Experiment and System Characterization Using a standard Neural Database

4.3.1 Experiment Setup

We recorded neural data using the ASIC from a chronic microelectrode array positioned over premotor cortex of the right hemisphere of an awake Rhesus Macaque. The chronic array features 18 independently movable electrodes, with each electrode positioned by a screw mechanism at a resolution of 50 microns. The electrodes themselves were epoxylite-coated tungsten electrodes with impedances ranging from 4-6 MOhms (FHC Inc.). Contact can be made from each of the 18 electrode to any of the 4 electrode on the ASIC. Each electrode was driven into cortex until neural activity was found. From this point, electrodes would be maintained at this position for days or weeks. In the experiment, we acquire the digitized data directly after the ADC as well as the compressed data. A detailed experiment setup is shown in Figure 11. The monkey is not shown due to regulation and ethics reasons.

To compare the performance of the system with previous works, we also characterized the system using a standard neural database [Quiroga '04]. We utilized a 12-bit DAC followed by a 40dB attenuator to play back the recorded neural waveforms to be recorded by the ASIC. The ASIC then transmits the compressed samples to a PC where the signal recovery block and QE block are implemented.

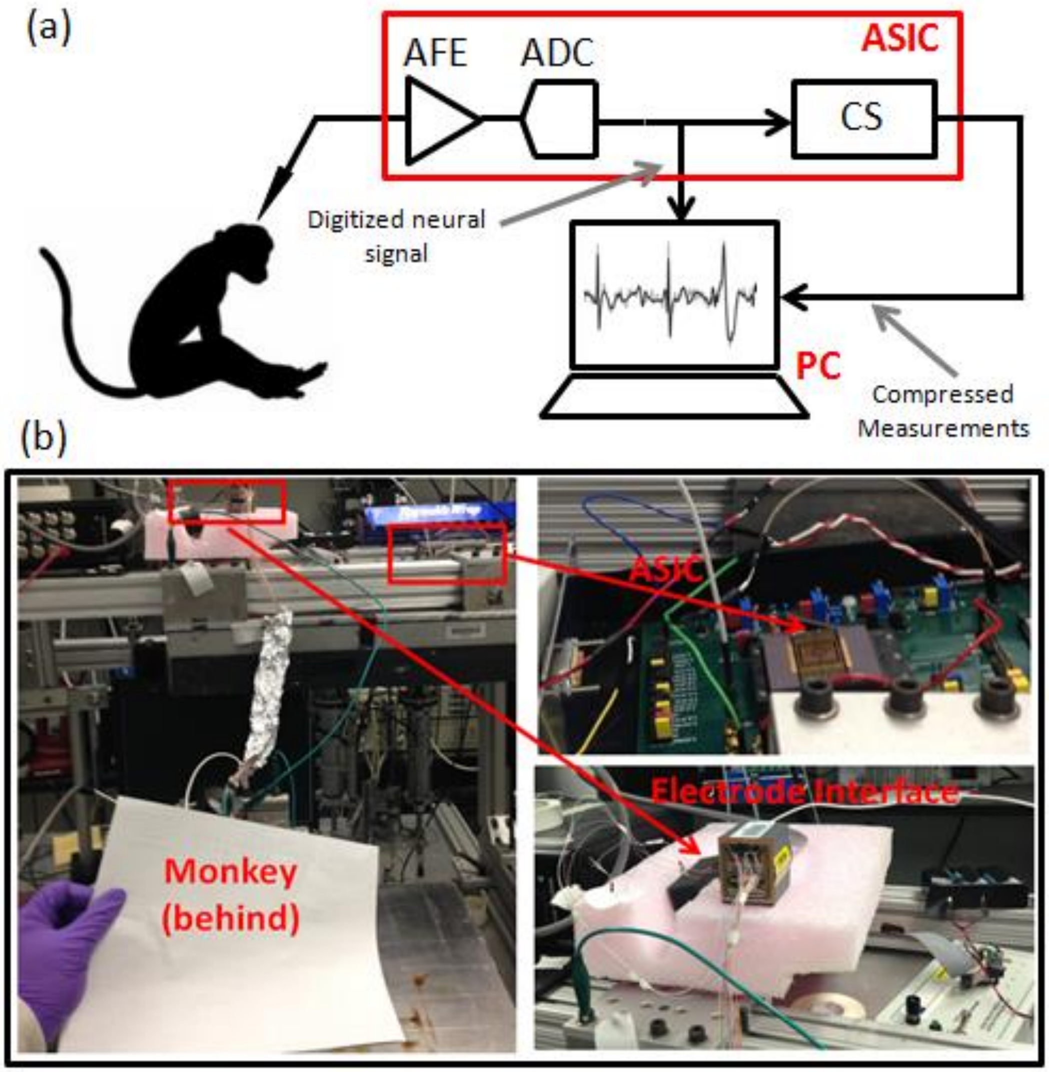


Figure 11. (a) In-vivo recording experiment using the ASIC. (b) The actually in vivo experiment.

5.3.2. Raw and recorded waveforms from the *In-vivo* experiment

We recorded neural signals in from 3 of the 18 electrodes placed within the premotor cortex of a Rhesus Macaque, where the electrodes SNR > 1.1. To calculate SNR, the method described in Section 4.1 is used. Around one minute of data is collect prior to compression to train a dictionary using method outlined in Section 3.1c. Around one minute of the data are collected for each electrode. During recording the ASIC is configured to compress the entire neural signal instead of just the spikes. Figure 12.a. shows the recovery results from an electrode with SNR=6.21. For this electrode, the CR is set to 12.8. Two types of spikes are seen at this electrode. Off-chip recovery block could recover both with great precision, achieving an $SNDR_x$ of 9.52 dB. Figure 12.b. shows the recovery quality of an electrode with lower SNR=2.14, where only one spike cluster is seen. CR is set to 4.3. Off-chip recovery block could recover spikes and the inter-spike signal at great accuracy, achieving spike $SNDR_x$ of 4.14 dB. This result validates the functionality of the complete system and demonstrates that its performance agrees with the offline analysis presented in previous sections.

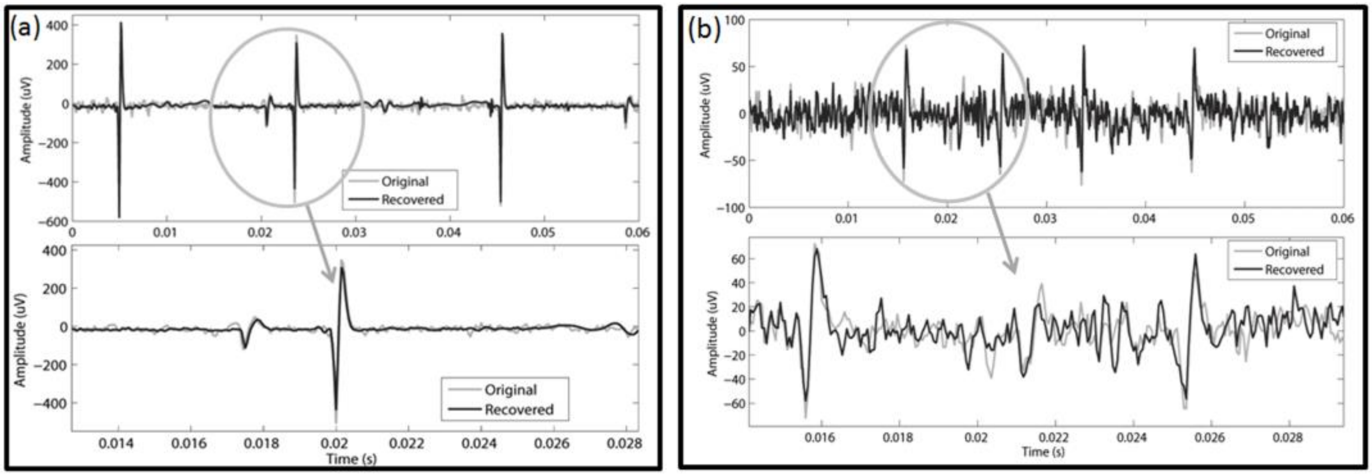


Figure 12. (a) In vivo test data acquired by the system in CES mode. The electrode SNR=6.21. (a) In vivo test data acquired by the system in CES mode. The electrode SNR=2.14

5.3.3. System characterization using a standard neural database

In order to compare our system's performance with previous published works, we characterized the system performance using a standard neural data base [Quiroga '04]. The dataset used is named “Easy1” in the database. All the spikes in this dataset have nomalized amplitude of 1, with various amount of noise added as zero mean Gaussian noise with standard deviation (s.t.d.) from 0.05 to 4.0. Figure 13.a. shows a temporal view of the recovery at CR=5 and 16 for Easy1, with noise s.t.d. = 0.05. Figure 13.b demonstrates the recovery quality and power consumption with respect to CR. The classification accuracy decreases as we increase CR. But as CR increase, more accumulators are turned off, therefore the power consumption of the ASIC decreases. For this particular dataset, name, spike classification accuracy reaches >95% when CR decreases below 21.3.

Figure.14 shows comparison of our system to prior works. All these systems intend to perform on-chip compression of neural signals to achieve reduction in data bandwidth for wireless (or wireline) communication. Like previous work, we have evaluated our approach on Easy1 dataset from [Quiroga '04] across all the noise standard deviations (from 0.05 to 4.0). The lowest CR needed to achieve >95% spike classification accuracy for all the dataset is used here as a performance metric (CR@95%).The CS circuit in this work can function with VDD of 0.53V without performance degradation and hence consumes only 0.83uW (per electrode) for the compression architecture. The CS block itself uses only 0.11mm^2 area per electrode. Even with the lowest power consumption and comparable area, this implementation achieves more than 5 times better compression rate (CR=10.6) than the state-of-the-art. The total power consumption per electrode (<15.83µW), including AFE and ADC, is comparable to published state of the art systems.

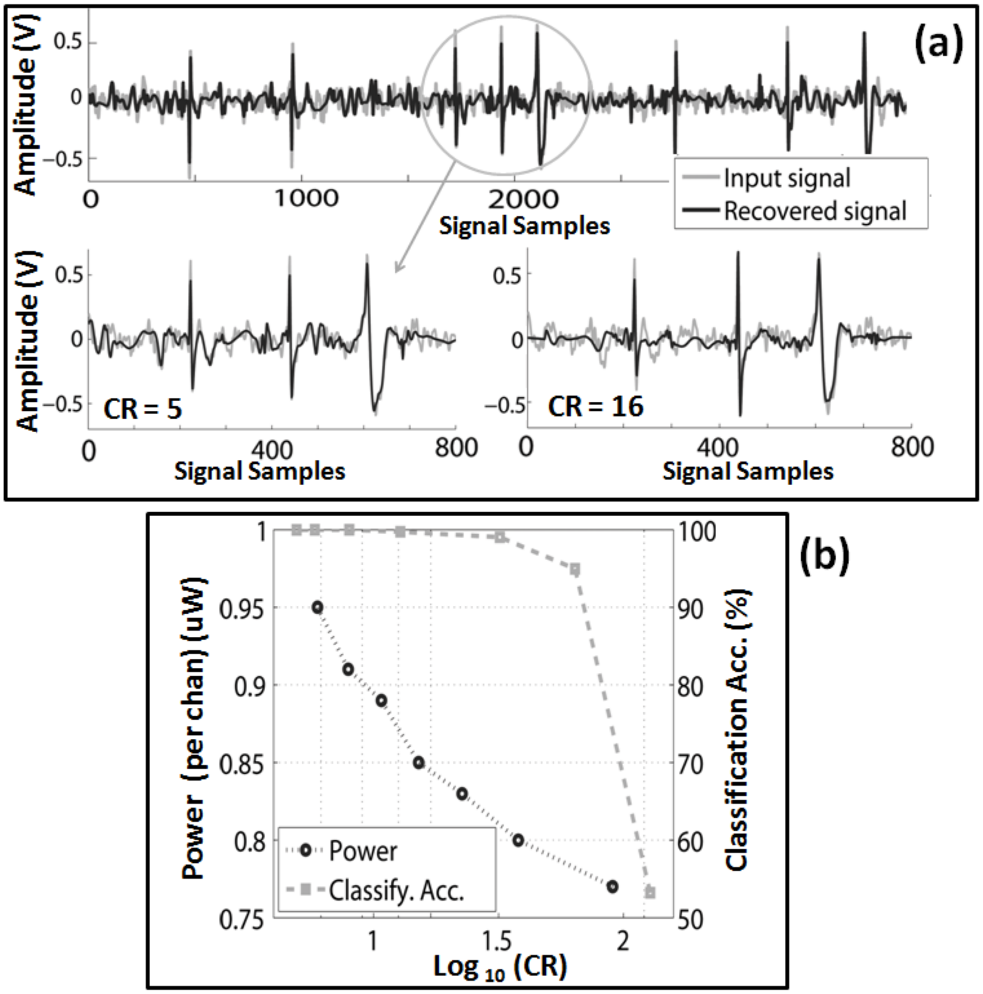


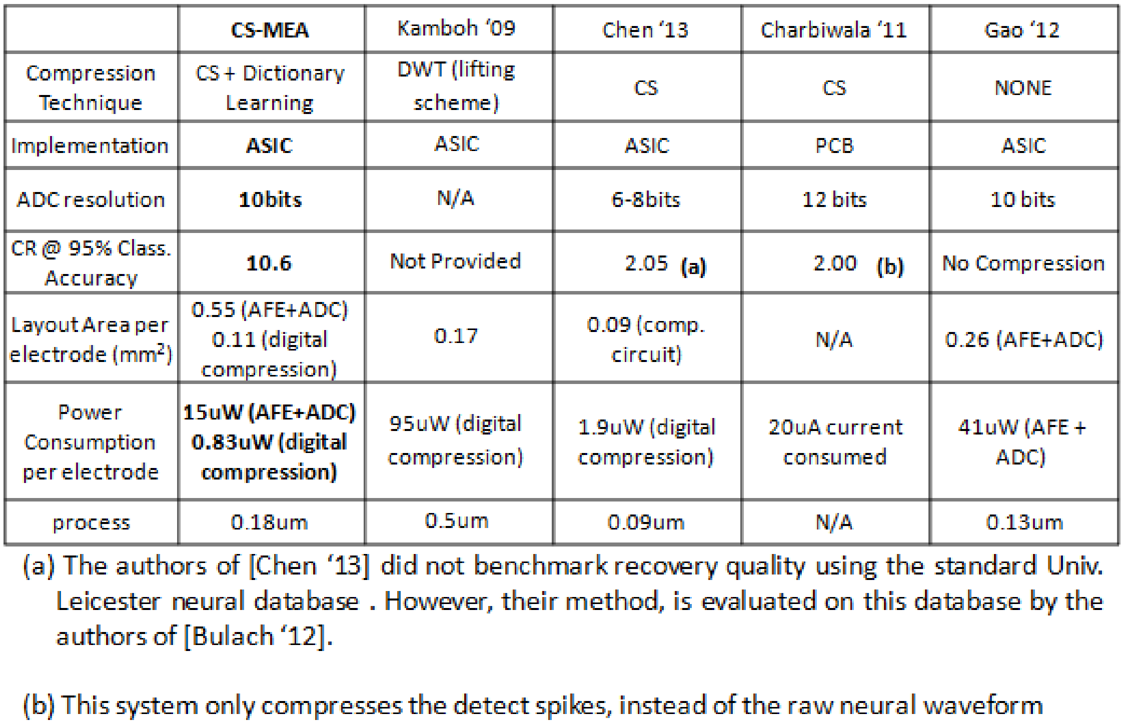
Figure 13. (a) Recovery quality at CR=5 and 16. (b) ASIC power consumption and spike classification accuracy with respect to CR. Classification is evaluated using the Easy1 database with noise s.t.d of 0.05 from [Quiroga '04] using a wavelet classification method similar to [Quiroga '04]

6. Conclusion

We have demonstrated a compressive sensing neural recording system. Using a learned dictionary, this system is capable of achieving high rate of compression for both raw neural signals (CR > 10.6) as well as detected spikes (CR > 16). This system is extremely low power (<0.83uW per electrode), and consumes very small area (<0.11mm^2). Thus this system can be scaled and integrated into large recording arrays containg thousands of electrodes.

Because the demonstrated compressed sensing technique rely on reconstructing the spike using dictionary learned using small duration of raw recording, an open-loop recording cannot adapt to changes in spike shape. By introducing close-looped recording with reconstruction performance evaluation, the system can detect and adapt to this changed, thus making the system more practical for long term recording.

While showing superior performance, the proposed system also has a few limitations: The experiments demonstrate that the proposed CS method is able to achieve extremely high compression rate for recording channels with high SNR. The compression rate degrades as electrode SNR decreaes, as noise affects the performance of the dictionary learning algorithm. Furthermore, without a proper dictionary, the system might not be able to reconstruct the waveform of a sparsely firing neuron. We will address these limitations in our future work.



	CS-MEA	Kamboh '09	Chen '13	Charbiwala '11	Gao '12
Compression Technique	CS + Dictionary Learning	DWT (lifting scheme)	CS	CS	NONE
Implementation	**ASIC**	ASIC	ASIC	PCB	ASIC
ADC resolution	**10bits**	N/A	6-8bits	12 bits	10 bits
CR @ 95% Class. Accuracy	**10.6**	Not Provided	2.05 **(a)**	2.00 **(b)**	No Compression
Layout Area per electrode (mm^2)	0.55 (AFE+ADC) 0.11 (digital compression)	0.17	0.09 (comp. circuit)	N/A	0.26 (AFE+ADC)
Power Consumption per electrode	**15uW (AFE+ADC)** **0.83uW (digital compression)**	95uW (digital compression)	1.9uW (digital compression)	20uA current consumed	41uW (AFE + ADC)
process	0.18um	0.5um	0.09um	N/A	0.13um

(a) The authors of [Chen '13] did not benchmark recovery quality using the standard Univ. Leicester neural database . However, their method, is evaluated on this database by the authors of [Bulach '12].

(b) This system only compresses the detect spikes, instead of the raw neural waveform

Figure 14. Comparison of our work with other state-of-the-art approaches.

Reference

[Braitenberg '91] Braitenberg, V., & Schüz, A. (1991). *Anatomy of the cortex: Statistics and geometry*. Springer-Verlag Publishing, 1991.

[Staba '02] Staba, R. J., Wilson, C. L., Bragin, A., Fried, I., & Engel, J. (2002). Sleep states differentiate single neuron activity recorded from human epileptic hippocampus, entorhinal cortex, and subiculum. *The Journal of neuroscience*,*22*(13), 5694-5704

[Hubel '59] Hubel, David H., and Torsten N. Wiesel. "Receptive fields of single neurones in the cat's striate cortex." *The Journal of physiology* 148.3 (1959): 574-591.

[ShahrokhI '10] Shahrokhi, Farzaneh, et al. "The 128-channel fully differential digital integrated neural recording and stimulation interface." *Biomedical Circuits and Systems, IEEE Transactions on* 4.3 (2010): 149-161.

[NeuroSeeker '13] http://www.neuroseeker.eu/

[Kim ‘06] Kim, S., Normann, R. A., Harrison, R., & Solzbacher, F. (2006, August). Preliminary study of the thermal impact of a microelectrode array implanted in the brain. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE* (pp. 2986-2989). IEEE.

[Aziz '09] Aziz, J. N., Abdelhalim, K., Shulyzki, R., Genov, R., Bardakjian, B. L., Derchansky, M., Derchansky., D., Serletis., and Carlen, P. L. (2009). 256-channel neural recording and delta compression microsystem with 3D electrodes. *Solid-State Circuits, IEEE Journal of*, *44*(3), 995-1005.

[Lopez '13] Lopez, C. M., Andrei, A., Mitra, S., Welkenhuysen, M., Eberle, W., Bartic, C., Puers, P., Yazicioglu, R.F., and Gielen, G. "An implantable 455-active-electrode 52-channel CMOS neural probe" In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International* (pp. 288-289). IEEE.

[Abdelhalim '13] Abdelhalim, K.; Jafari, H.M.; Kokarovtseva, L.; Perez Velazquez, J.L.; Genov, R., "64-Channel UWB Wireless Neural Vector Analyzer SOC With a Closed-Loop Phase Synchrony-Triggered Neurostimulator," *Solid-State Circuits, IEEE Journal of* , vol.48, no.10, pp.2494,2510, Oct. 2013

[Mitra '13] Mitra, S., et.al. "24-channel dual-band wireless neural recorder with activity-dependent power consumption," ISSCC Feb. 2013.

[Gosselin '09(a)] B. Gosselin and M. Sawan. An ultra low-power cmos automatic action potential detector. Neural Systems and Rehabilitation Engineering, IEEE Transactions on, pages 346–353, Aug. 2009

[Chae '08] M. Chae, W. Liu, Z. Yang, T. Chen, J. Kim, M. Sivaprakasam, and M. Yuce. A 128-channel 6mw wireless neural recording ic with on-thefly spike sorting and uwb tansmitter. Solid-State Circuits Conference, pages pp. 146–603, 2008.

[Gosselin '09(b)] B. Gosselin, A. E. Ayoub, J. F. Roy, M. Sawan, F. Lepore, A. Chaudhuri, and D. Guitton. A mixed-signal multichip neural recording interface with bandwidth reduction. Biomedical Circuits and Systems, IEEE Transactions on, pages 129–141, June. 2009.

[Oweiss '07] K.G. Oweiss, A. Mason, Y. Suhail, A.M. Kamboh, and K.E. Thomson.A scalable wavelet transform vlsi architecture for real-time signal processing in high-density intra-cortical implants. Circuits and Systems I. IEEE Transactions on, pages 1266–1278, June. 2007.

[Kamboh '08] A. M. Kamboh, A. Mason, and K. G. Oweiss. Analysis of lifting and b-spline dwt implementations for implantable neuroprosthetics. Journal of Signal Processing Systems, pages 249–261, Sept. 2008.

[Mamaghanian '11] H Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes. Biomedical Engineering, IEEE Transactions on, 58(9):2456–2466, Sept. 2011.

[Gangopadhyay '14] Gangopadhyay, D., Allstot, E. G., Dixon, A. M., Natarajan, K., Gupta, S., & Allstot, D. J. (2014). Compressed Sensing Analog Front-End for Bio-Sensor Applications. *Solid-State Circuits, IEEE Journal of*, *49*(2), 426-438.

[Dixon '12] A.M.R. Dixon, E.G. Allstot, D. Gangopadhyay, and D.J. Allstot. Compressed sensing system considerations for ecg and emg wireless biosensors. Biomedical Circuits and Systems, IEEE Transactions on, 6:156–166, April. 2012.

[Charbiwala '11] Z. Charbiwala, V. Karkare, S. Gibson, D. Markovic, and M. B. Srivastava. Compressive sensing of neural action potentials using a learned union of supports. Body Sensor Networks (BSN), 2011 International Conference on, pages 53 – 58, May. 2011.

[Zhang '14] Zhang, J., et.al, "An Efficient and Compact Compressed Sensing Microsystem for Implantable Neural Recordings," IEEE Trans. BioCAS, Aug. 2013.

[Suo '13] Suo, Y., Zhang, J., Etienne-Cummings, R., Tran, T. D., & Chin, S. (2013, October). Energy-efficient two-stage Compressed Sensing method for implantable neural recordings. In *Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE* (pp. 150-153). IEEE.

[Suo '14] Suo, Y., et.al, "Structured Dictionary Learning for Classification," arxiv.org/pdf/1406.1943

[Candes '06] E. Candes, Romberg J, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on, pages 489 – 509, Feb. 2006.

[Donoho '06] D. L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[Aharon '06] 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.

[Lewiciki '00] Lewicki, M., and Sejnowski, T. (2000). Learning overcomplete representations.*Neural computation*, *12*(2), 337-365.

[Engan '00] Engan, K., Aase, S. O., and Husøy, J. H. (2000). Multi-frame compression: Theory and design. *Signal Processing*, *80*(10), 2121-2140.

[Henze '00] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzs´aki, “Intracellular features predicted by extracellular recordings in the hippocampus in vivo,” *Journal of neurophysiology*, vol. 84, no. 1, pp. 390–400, 2000.

[Elad '07] M Elad. Optimized projections for compressed sensing. Signal Processing, IEEE Transactions on, pages 5695–5702, Dec. 2007.

[Sapiro '09] J. M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. Image Processing, IEEE Transactions on, pages 1395–1408, July. 2009.

[Tropp '07] J.A. Tropp. Signal recovery from random measurements via orthogonal matching pursuit. Information Theory. IEEE Transactions on, pages 4655 – 4666, Dec. 2007.

[Needell '09] Needell, D., & Tropp, J. A. (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, *26*(3), 301-321.

[Baraniuk '10] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. Information Theory. IEEE Transactions on, pages 1982–2001, April. 2010.

[Ludwig '09] Ludwig, K. A., Miriani, R. M., Langhals, N. B., Joseph, M. D., Anderson, D. J., & Kipke, D. R. (2009). Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *Journal of neurophysiology*,*101*(3), 1679.

[Kamboh '09] Kamboh, A. M., Oweiss, K. G., & Mason, A. J. (2009, May). Resource constrained VLSI architecture for implantable neural data compression systems. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on* (pp. 1481-1484). IEEE.

[Quiroga '04] Quiroga, R. Quian, Zoltan Nadasdy, and Yoram Ben-Shaul. "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering."*Neural computation* 16.8 (2004): 1661-1687.

[Charbiwala '12] Charbiwala, Z., et.al. CapMux: A scalable analog front end for low power compressed sensing. IEEE International Green Computing Conference (IGCC), Jun. 2012

[Gao '12] Gao, H., et.al. "HermesE: A 96-Channel Full Data Rate Direct Neural Interface in 0.13um CMOS," IEEE JSSC, 47(4), 1043-1055. Apr. 2012.

[Bulach '12] Bulach, C., et.al. Evaluation study of compressed sensing for neural spike recordings. IEEE EMBC, 2012 (pp. 3507-3510), Aug. 2012.