# Chapter 14
# Big Data Management in Neural Implants: The Neuromorphic Approach

**Arindam Basu, Chen Yi, and Yao Enyi**

## 14.1 Introduction: Brain as a Source of Big Data

In the age of the Internet of Things (IoT) with millions of interconnected sensors spewing out data, we are facing a data deluge—there is a need for solutions to store and process this data. A unique set of IoT applications relates to the human body—in particular wearables and implantables to collect data from the human brain for neuroscientific research, prostheses or medical interventions [1–6]. The study of the human brain is one of the most important frontiers in science research today—there is a lot of emphasis on this with several billion dollar efforts worldwide to understand more about the brain [7, 8]. To get an idea about the scale of data generated by the brain, we first note from anatomy that the average adult human cortex has approximately $10^{11}$ neurons, widely regarded as the fundamental computational unit of the brain, with $10^{14}$ synapses or interconnections [9]. Assuming average cortical firing rates (a neural firing or discharge refers to a digital like pulse also called a spike or action potential) of 1–10 Hz [10], the human brain is generating at least $10^{11}$ spikes or events per second and about $10^{14}$ synaptic operations per second. Assigning an unique address or identifier to each neuron would need $b_{\text{addr}} = \log_2(10^{11}) \approx 35$ bits—hence, the data rate generated by the brain is a whopping 3.5 Terabits/second. To put this in perspective, the exponential growth of data has put internet data in the exascale ( $10^{18}$ bits). One human brain can generate approximately the same amount of data in $10^6$ s or 50 days! Of course, this is an extreme case and we are not aiming to store all the neural firings of a human brain over his or her lifetime (at least not at this moment) and neither do we currently possess the technology to access this data (but we are constantly striving

A. Basu (✉) • C. Yi • Y. Enyi
School of EEE, Nanyang Technological University, Singapore, Singapore
e-mail: arindam.basu@ntu.edu.sg



A. Chattopadhyay et al. (eds.), *Emerging Technology and Architecture for Big-data Analytics*, DOI 10.1007/978-3-319-54840-1_14

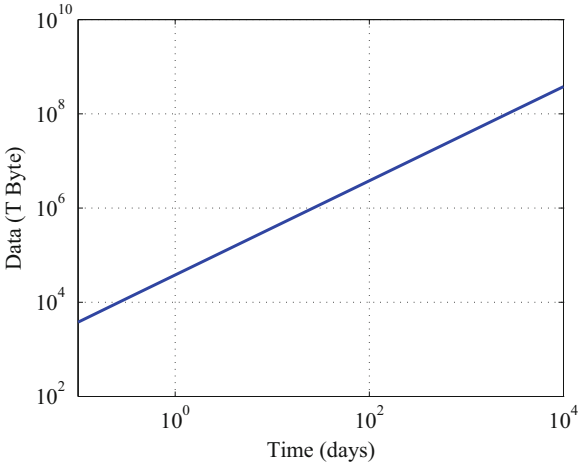


**Fig. 14.1** The brain as a source of big data: a single human brain generates data at a rate of 3.5 Terabits/second. The total data can reach exabyte scale within a year

to record data from more neurons and this is one of the prime goals of the Brain initiative)—but this helps to give an idea about the scale of the problem. Figure 14.1 shows the rapid scaling of data generated from a single human brain over time.

Just like any other application related to big-data, the problems of storage and manipulation exist in this data generated by the brain. However, an added problem stems in this case from the strict power dissipation requirement of electronics implanted within the brain to collect the data. Any electronics in contact with the cortical tissue cannot generate heat larger than 80 mW/cm$^2$ [11, 12] to avoid damaging the neural tissue (temperature rise less than 1 °C). Instead of implants, another option is to collect data non-invasively through EEG from the scalp—however, EEG provides a highly filtered (both spatially and temporally) picture of the brain activity and is not informative enough for activities with many degrees of freedom such as upper limb prostheses [13, 14]. Therefore, in the rest of this chapter, we only consider the case of neural recording from implanted electrodes that can provide enough information for dexterous motor control.

## 14.2 The Nature of Neural Data

The signals recorded by neural implants are obtained typically through microelectrode arrays such as the Utah or Michigan arrays [15–17]. The neural signals can be broadly divided into two categories—(1) Local Field Potentials (LFP) that are 1–10 mV in amplitude occupying a bandwidth of 1–100 Hz produced by combined activity of groups of neurons and (2) neural spikes or action potentials which are much smaller (10–100 μV in amplitude) but occupy a much larger bandwidth of $\approx$ 0.2–5 kHz. While both signals have useful information [18, 19], most of the studies on neural prosthetics that require fine motor manipulation typically use

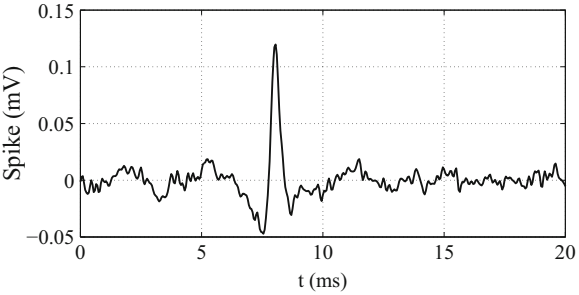


**Fig. 14.2** A neural spike recorded from the pre-frontal cortex of a rat. Neural spikes typically have a small amplitude $\approx$ 10–100 $\mu$V while occupying a large bandwidth of $\approx$ 0.2–5 kHz

neural spikes [20–23]. In this chapter, we will therefore focus on neural recording systems for sensing and transmitting neural spikes. Unlike LFP signals where the amplitude is informative, it is believed that spikes are like digital signals [24] where the amplitude is non-informative but the timing and firing rate of spikes are important. An example of a spike recorded from pre-frontal cortex of a rat is shown in Fig. 14.2.

## 14.3 System Architectures for Neural Spike Recording Systems: Neuromorphic Compression Schemes

The different blocks comprising a typical neural recording system are shown in Fig. 14.3a. In a typical system, the neural signal is amplified by a low-noise amplifier (LNA) [25–29], followed by an optional variable gain amplifier (VGA) and finally an analog-digital converter (ADC) [29–32] before being transmitted wirelessly. We can estimate the data rate for such a system under some mild assumptions. Denoting the number of recording channels as $N_{\text{chan}}$, ADC sampling rate and bit resolution as $f_{\text{ADC}}$ and $b_{\text{ADC}}$, respectively, the data rate $R_{\text{typ}}$ of a typical neural implant is given by:

$$R_{\text{typ}} = N_{\text{chan}} \times f_{\text{ADC}} \times b_{\text{ADC}} \tag{14.1}$$

As an example, for moderate values of $N_{\text{chan}} = 100, f_{\text{ADC}} = 20\,\text{kHz}$, and $b_{\text{ADC}} = 10$ bits, we get $R_{\text{typ}} = 20\,\text{Mbps}$—a huge data rate that will drain out an implant's battery in a matter of hours given typical power requirements of $\approx$ 50–1000 pJ/bit for wireless transmitters [33–36]. Hence, it is imperative to compress the data and reduce the concomitant power dissipation so that the neural recording system can be scaled in future to thousands or millions of channels. One possible way to do this is to take inspiration from the brain—in the absence of the implant, the brain would have processed the thousands of neural spikes recorded by the implant and given a refined command to the next region. Similarly, we can also use electronics to perform this signal processing on the implant, thus reducing the bandwidth of data to be transmitted. Figure 14.3 shows three different modes of compression based on

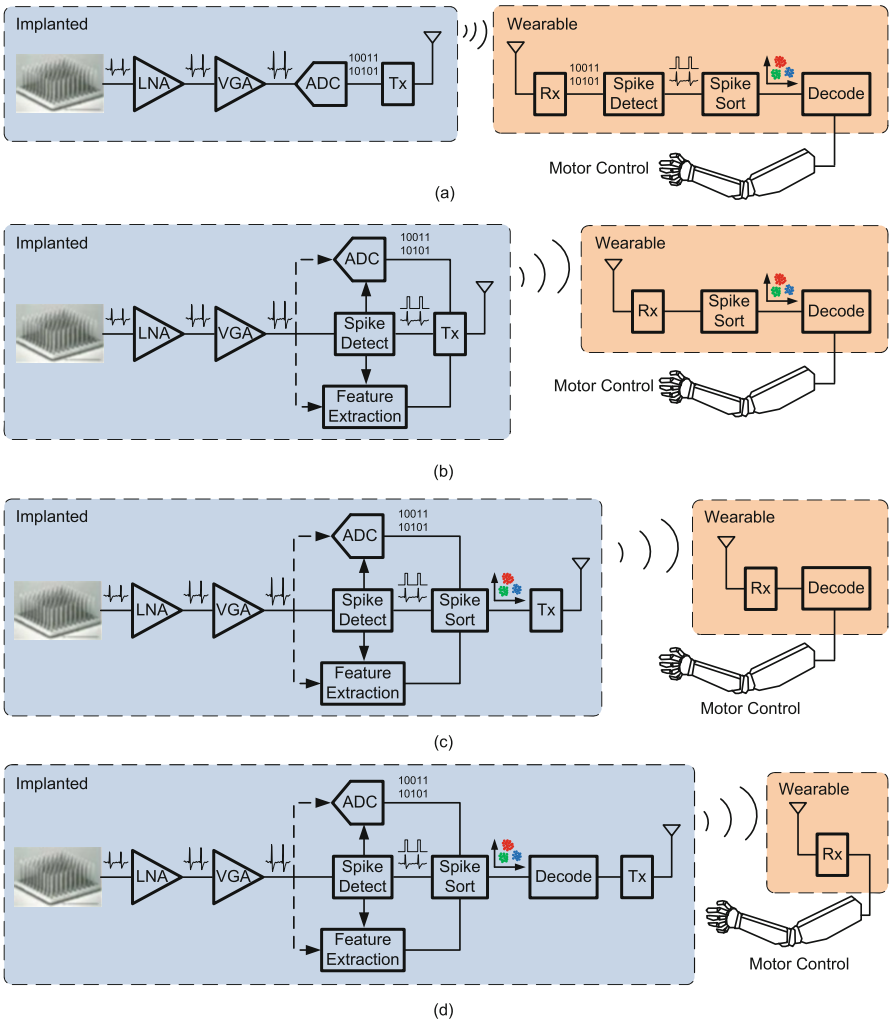


**Fig. 14.3** Block diagram of a typical neural recording system which senses, digitizes and wirelessly transmits the neural data. As an alternative to sending raw data, different neuromorphic schemes may be used as shown to achieve different rates of compression. (**a**) Typical. (**b**) Mode 1. (**c**) Mode 2. (**d**) Mode 3

the amount of signal processing kept on the implant. There is a trade-off in this case between amount of extra area and energy expended on signal processing in-implant versus the energy saved in reduced transmission. Clearly, it is not beneficial if the added circuits for signal processing burn as much energy as the energy saved in reduced data rate!

One way to perform the processing at very low energy/area overheads is to use neuro-inspired analog circuits, sometimes also called ‘neuromorphic’ circuits

following Carver Mead's seminal paper [37]. Mead and others [38] have shown that analog circuits require less energy and area than digital counterparts when processing signals at a low resolution, typically $\leq$ 8 bits. The brain also uses a similar principle by computing using analog quantities such as charge, currents and ionic concentrations and this is cited as one of the reasons for its power efficiency. This is hence well suited for processing noisy sensory signals where precision is limited by input signal to noise ratios. In the rest of the chapter, we will explore several such schemes to compress neural recording data by extracting information from it.

### *14.3.1 Compression Mode 1: Spike Detection*

The first scheme is inspired by a communication protocol used in neuromorphic chips. Several neuromorphic sensors and neural networks have been designed using brain-inspired analog processing principles [39–44] while noise robust digital pulses are used for communication [45, 46]. Since digital communication is much faster ($\sim$ 10 Gbps) than the average firing rate of a neuron ($\sim$ 10 Hz), the firing information of multiple neurons can be multiplexed on the same serial bus where the identity of the source neuron is encoded in a simultaneously transmitted digital address. This protocol is referred to as Address Event Representation (AER) and allows neuromorphic spiking chips to communicate data from $N$ neurons using only $\log_2(N)$ wires.

The AER scheme can be adopted for neural implants as well since in many cases, we are interested in only knowing the occurrence of spikes. In that case, circuits are needed to distinguish spikes from background noise—these are called spike detectors. Figure 14.3b denotes this scheme as Mode 1 with three possible variants. The earliest instance of such detectors is based on simple thresholding circuits[24] where it is assumed that the amplitude of the spike is larger than background noise by a certain amount. A feedback loop is used to track the baseline noise level and the spike detection threshold is set to a multiple of this value. However, this method was found to produce high false positives in noisy conditions and hence an improved detection method using a non-linear energy operator (NEO) has been proposed. The NEO operator is defined as:

$$\text{NEO}(V) = \left(\frac{dV}{dt}\right)^2 - \frac{d^2V}{dt^2} \cdot V \tag{14.2}$$

Several analog implementations of the NEO scheme have been reported [47–50] and an example of spike detection waveforms from the implementation in [47] is shown in Fig. 14.4. We refer to this method as Mode 1-A.

The spike detection method discards all information about the amplitude and shape of the neural spike—this information may, however, be useful at a later stage to decide the identity of the source neuron. Hence, two other variants of

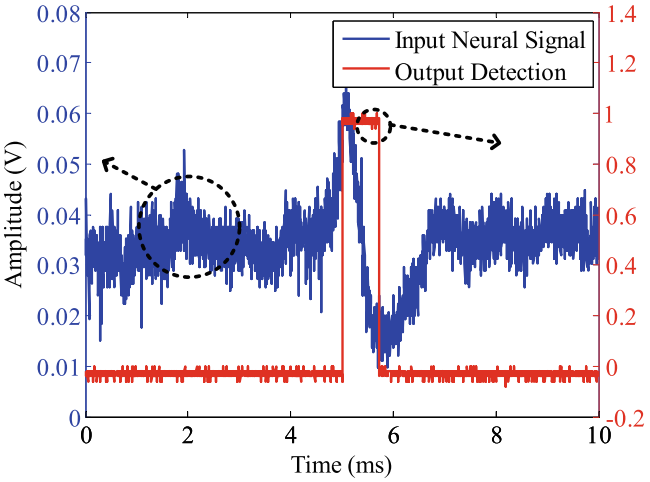


**Fig. 14.4** Input noisy neural signal and corresponding digital spike detection output from the implementation in [47]. Only the detection result can be transmitted thus eliminating background data

the previously mentioned detection scheme have been commonly used. In some cases [51, 52], the authors use a regular spike detector to trigger the capture of a pre-defined number of samples of the neural spike signal so that all the features of the wave shape are retained for future extraction. We refer to this method as Mode 1-B. The other prevalent approach is to extract the relevant features (such as maximum, minimum, temporal width, derivative extrema) from the neural spike waveform when triggered by the spike detector [36, 48, 53–56]. Only these features are now digitized and transmitted providing a good trade-off between data reduction and signal information retention. We refer to this as Mode 1-C.

We can now derive the data rates $R_{1-A}$, $R_{1-B}$, and $R_{1-C}$ required by each of the compression schemes. Denoting the number of biological neurons recorded by the sensor as $N_{\text{neu}}$ (different from $N_{\text{chan}}$), firing rates of each neuron as $f_{\text{bio}}$ we can write the equations as:

$$R_{1-A} = N_{\text{neu}} \times f_{\text{bio}} \times \lceil \log_2(N_{\text{chan}}) \rceil \tag{14.3}$$

$$R_{1-B} = N_{\text{neu}} \times f_{\text{bio}} \times f_{\text{ADC}} \times b_{\text{ADC}} \times t_{\text{spk}} \tag{14.4}$$

$$R_{1-C} = N_{\text{neu}} \times f_{\text{bio}} \times N_f \times b_{\text{ADC}} \tag{14.5}$$

where $t_{\text{spk}}$ denotes the time span of the neural signal per spike transmitted in Mode 1-B, $N_f$ denotes the number of features extracted in Mode 1-C and other variables have same meaning as defined earlier. We can estimate the degree of compression by assuming some nominal values of the parameters: $N_{\text{neu}} = 200$, $f_{\text{bio}} = 10\,\text{Hz}$, $t_{\text{spk}} = 3\,\text{ms}$, $N_f = 4$, $N_{\text{chan}} = 100$, $f_{\text{ADC}} = 20\,\text{kHz}$ and $b_{\text{ADC}} = 10$ bits. Then the three data rates become $R_{1-A} = 14\,\text{kbps}$, $R_{1-B} = 120\,\text{kbps}$ and $R_{1-C} = 80\,\text{kbps}$. Compared to the typical data rate, these modes offer a compression between $\approx$ 100–1000×.

### 14.3.2 Compression Mode 2: Spike Sorting

The next possible scenario for compression is to use the features of the spike waveform to separate or classify each different wave shape into its own category representing a different source neuron. This method of assigning each distinct neural spike shape recorded on the same channel one unique identifier is called 'spike sorting' [57, 58]. Each category, in which spikes have similar shape, is believed to be generated by one neuron. The reasoning behind spike sorting is that the shape of spikes generated by neurons and recorded by an electrode is stereotypical, determined by the morphology of the dendritic trees of the neuron and the transmission pathway to the electrode. It is therefore believed that the shape of spikes from different neurons are distinct from each other and does not change over time, or at least over a significant amount of time. Though some work has demonstrated spike sorting may not be necessary for robust decoding performance [59, 60], the majority of work today still uses spike sorting to squeeze out as much information as possible from the neural recording implant.

Some authors have integrated a spike sorting classifier on the implant [61, 62]. While there are some implementations that have used supervised methods similar to template matching [63], most other approaches [64, 65] use unsupervised clustering techniques due to the advantage of not needing explicit training sessions. Figure 14.5 depicts the typical steps involved in spike sorting. After sorting, only the distinct identifier of the source neuron needs to be sent resulting in huge compression. We can estimate this data rate in Mode 2 as:

$$R_2 = N_{\text{neu}} \times f_{\text{bio}} \times \lceil \log_2(N_{\text{neu}}) \rceil \tag{14.6}$$

where the symbols have the same meaning defined earlier. Using the same values of the parameters used in the earlier Sect. 14.3.1, we can estimate the data rate for this mode to be $R_2 = 16$ kbps equivalent to a compression of $\approx 1000\times$ compared to a typical case.

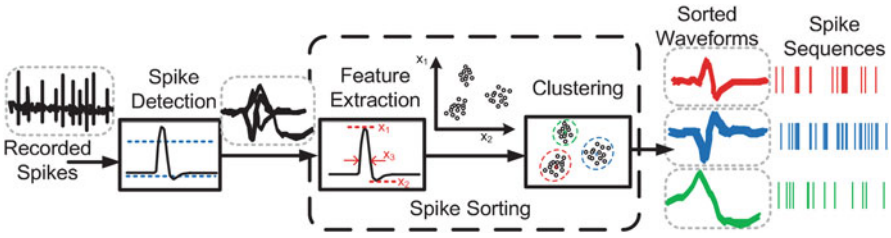


**Fig. 14.5** The steps involved in spike sorting include feature extraction followed by unsupervised clustering to separate the neural spikes into distinct categories according to their shape

### *14.3.3 Compression Mode 3: Intention Decoding*

The final and most advanced mode of compression is attained when the last stage of signal processing—decoding intentions from the recorded multi-channel spike train—is also integrated in the implant. This is shown in Fig. 14.3c as Mode 3. In this chapter, we focus on systems for motor prosthesis only—hence, in this case, intentions refer to 'motor' intentions or desire to move a limb. The fundamental of current decoding algorithms can be referred back to the work done by Georgopoulos and his colleagues [66, 67]. It is revealed in the experiment that the activity intensity of some neurons in the motor cortex is tuned to be a sinusoidal function of the movement direction of the arm with respect to a preferred direction where the activity reaches its maximum. They therefore proposed to represent each neuron by a vector indicating its preferred direction. The population vectors can be obtained by linear combination of all preferred vectors in the group weighted by the firing rate in the short time period of tens of millisecond, leading to a prediction on the velocity of upcoming arm movement [68].

Current state-of-the-art decoding algorithms for mapping population activity into motor intention can be categorized into two broad subgroups: inferential decoders [69–71] and classifiers [1, 20, 72]. However, most of these algorithms are run using bulky computers with wires connecting to the patient which impairs free movement and are a risk for infection. Recently, some approaches have been proposed for custom, low-power, compact hardware implementations of decoding algorithms [73–75] of which only one has shown measured results from a low-power integrated circuit [76] to decode motor intentions for dexterous finger movement as done in [20]. In the rest of the chapter, we elaborate on the details of this design, show the decoding performance and estimate achievable data compression using this scheme.

#### 14.3.3.1 Algorithm: Extreme Learning Machine

The machine learning algorithm used in this work is the Extreme Learning Machine (ELM) [77, 78]. It is a two-layer neural network (Fig. 14.6) where the first layer of weights from inputs to hidden neurons ($w_{ij}$ denotes weight from $i$-th input to $j$-th hidden neuron) are fixed and random. Only the weights in the second layer from the hidden neurons to output neurons need to be trained. Using $\beta_{ki}$ to denote the weight from the $i$-th hidden neuron to the $k$-th output neuron, we can express the $k$-th output $o_k$ as:

$$o_k = \sum_{i}^{L} \beta_{ki} g(\boldsymbol{w}_i, \boldsymbol{x}, b_i) = \sum_{i}^{L} \beta_{ki} h_i = \boldsymbol{h}^T \boldsymbol{\beta}_k$$
$$\boldsymbol{w}_i, \boldsymbol{x} \in \Re^D; \beta_{ki}, b_i \in \Re; \boldsymbol{h}, \boldsymbol{\beta}_k \in \Re^L \tag{14.7}$$

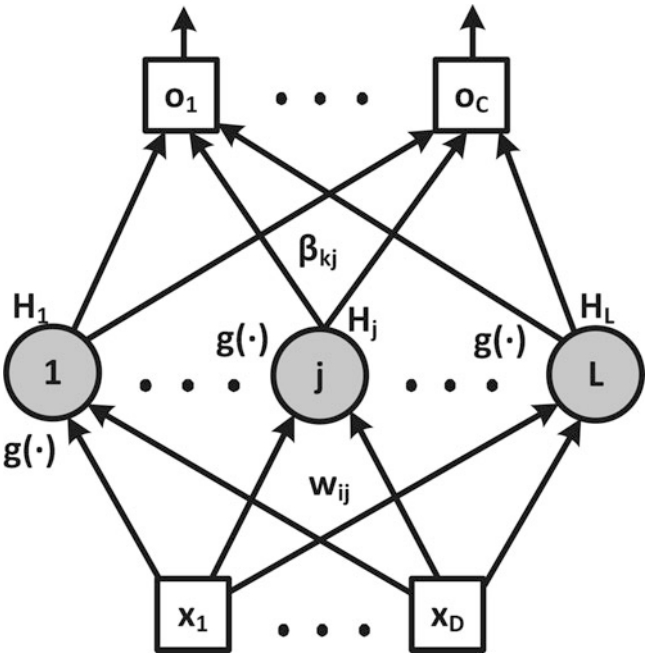


**Fig. 14.6** Extreme Learning Machine (ELM) is a two-layer neural network where the weights of the first layer are random and fixed. Only second layer weights are tuned according to the task

where **x** denotes the $D$-dimensional input vector, **h** is the $L$-dimensional output of the hidden layer, $g()$ is the non-linear activation function of the hidden layer and $b_i$ denotes the bias of the $i$-th hidden layer neuron. One of the commonly used activation functions is the additive node where $h_i = g(\boldsymbol{w}_i^T \boldsymbol{x} + b_i)$ and $g : \Re \rightarrow \Re$ is any non-linear function with finitely many discontinuities. While the outputs $o_k$ can be directly used for regression, for classification, we assign the input sample to the class belonging to the output neuron with the highest value.

The second layer weights can be obtained by a direct solution instead of typically used iterative methods such as back propagation for multi-layer neural networks—hence, the training time for ELM based systems is much smaller. The output weights for each of the $C$ classes can be optimized separately by using the same hidden layer values. Suppose there are $p$ samples and let $H$ denote the $p \times L$ hidden layer matrix where each row stores the output of the hidden neurons for one sample. Further, let $\boldsymbol{T}_k \in \Re^p$ denote the target or desired values for the $k$-th hidden neuron. Then, the ideal weights $\hat{\boldsymbol{\beta}}_k$ for the $k$-th hidden neuron is obtained as solution of the following optimization problem [78]:

$$\hat{\boldsymbol{\beta}}_k = \arg\min_{\beta_k} \|H\boldsymbol{\beta}_k - \boldsymbol{T}_k\|_2 + \gamma \|\boldsymbol{\beta}_k\|_2 \tag{14.8}$$

where the second term in the equation is needed for regularization and $\gamma$ is optimized on the validation set as a hyper-parameter. Closed form solutions to the value of $\boldsymbol{\beta}_k$ can be obtained in two different ways for the cases where the number of training samples is less or more than the number of hidden neurons [78].

To apply this neural network to neural decoding, the authors use an approach similar to [20] where the Artificial Neural Network is replaced by an ELM. The ELM decodes the onset time as well as the type of movement from the asynchronous neural spikes every $T_s = 20\,\text{ms}$. First, instantaneous firing rate $r_i(t_k)$ at time $t_k$ of each biological neuron is computed by counting the number of spikes in a time window $T_w = 100\,\text{ms}$. Then, the input feature vector to the ELM at time $t_k$ is defined by:

$$\boldsymbol{x}(t_k) = [r_1(t_k), r_2(t_k) \ldots r_D(t_k)] \tag{14.9}$$

The total number of output neurons $C$ in this case is equal to $M + 1$ where there are $M$ movement types and one extra neuron is used to classify the onset time of movement. For training, the last output for onset time is trained on the entire dataset while the others are trained only on neural data during movement. Also, the last neuron is trained to solve a regression problem where the target function is trapezoidal—it gradually rises from 0 to 1 to mimic the gradually increasing activity of biological neuron ensembles. To reduce false positives in detecting movement onset, further processing is done on this 'primary' output by voting across the decision for several consecutive time samples [76] to produce the post-processed output. Another special signal processing feature of the IC is ability to include time delayed versions of neuronal activity as additional inputs to the ELM, i.e the number of inputs $D$ to the ELM may be larger than the number of biological neurons $N$. This feature, referred to as Time-delay based dimension increase (TDBDI), is especially useful for chronic implants where the signal quality from many probes degrades with time due to scarring and fibrotic encapsulation.

The main reason for choosing the ELM algorithm is that most of the multiplications to be done in this architecture are the $D \times L$ random scalings in the first stage which can be done in very low energy and area using analog neuromorphic circuits. The mismatch induced errors [79] in analog circuits is not a problem in this case but can be part of the random coefficients. To get high accuracy, the trainable weights of the second stage can be implemented using digital circuits. However, this does not degrade system level energy efficiency as long as $D >> C$ which ensures that the number of multiplications in second stage are much less than that in the first stage. The circuit implementation of this algorithm is shown next.

#### 14.3.3.2 Chip Architecture

The system architecture for the neuromorphic ELM chip is shown in Fig. 14.7. Since biological firing rate are sparse, the AER protocol described in Sect. 14.3.1 is used to send the neural spikes to a desired channel based on the address or identity of the source neuron. Then, the input handling circuits (IHC) compute an average

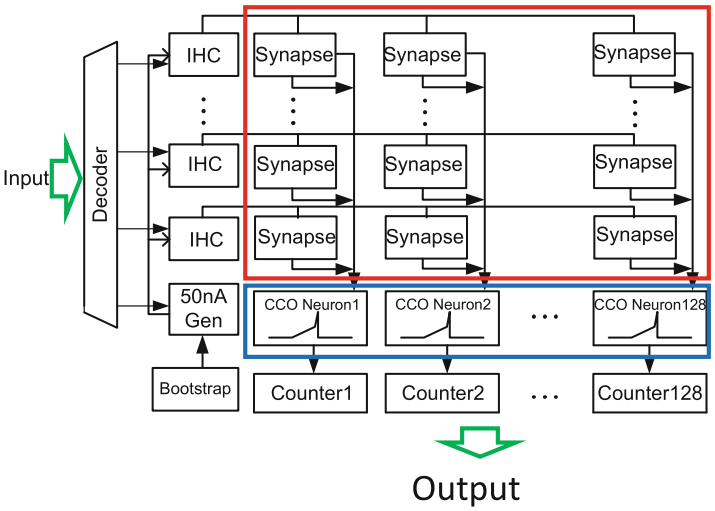


**Fig. 14.7** Overall architecture of the ELM based decoder IC has a decoder to pass input spikes to desired channel, input handling circuits (IHC) to calculate average firing rate of spikes as a feature, a synapse array to create the random weighting of inputs needed in stage 1 of ELM and an array of hidden neurons

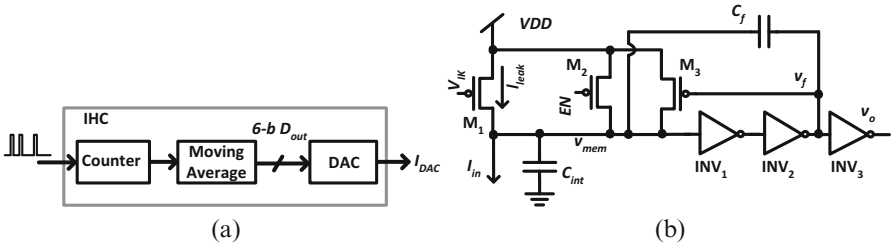


**Fig. 14.8** (**a**) The IHC block comprises a counter and a moving average circuit to compute average firing rate in digital domain. The DAC then converts the digital number to an analog current. (**b**) The neuron is made of a current controlled oscillator (CCO) that clocks a counter (not shown)

firing rate using digital circuits in two steps (Fig. 14.8a). First, a counter estimates instantaneous firing rates by counting the number of spikes in a time interval $T_s$. Then a moving average circuit finds average firing rate in a time window $T_w$. This digital number is then converted to an analog current $I_{DAC}$ using a digital to analog converter (DAC) so that following steps can be implemented in the analog domain. The major task of multiplication by a random number is performed by the synapse—a current mirror comprising identical minimum sized transistors. Ideally, without statistical variations, the current mirror would produce same output current as its input. However, due to mismatch and sub-threshold operation of the transistors, the output current from a mirror is given by:

$$I_{\text{out}} = e^{\Delta V_T / U_T} I_{\text{in}} \tag{14.10}$$

where $\Delta V_T$ denotes threshold voltage mismatch between the two mirror transistors and $U_T$ denotes thermal voltage. In this architecture, the diode connected transistor for every row is shared while the synapse just consists of a single mirror transistor. Hence, the weight of the synapse connecting $i$-th input to the $j$-th neuron is given by $w_{ij} = e^{\Delta V_{T,ij}/U_T}$. The sum of these currents are obtained by just wiring the drains of the mirror transistors together. Finally, this current is converted to the hidden layer output by passing it through a neuron circuit shown in Fig. 14.8b. The neuron is a current controlled oscillator (CCO) whose frequency of oscillation is given by:

$$f_{\text{CCO}} = \frac{I_{\text{in}} - I_{\text{leak}}}{C_f \times VDD} \tag{14.11}$$

This equation is valid as long as $I_{\text{in}} << I_{\text{rst}}$ where $I_{\text{rst}}$ denotes the reset current flowing through transistor M3 when turned fully on. The current $I_{\text{leak}}$ serves the function of the bias term $b_i$ in Eq. (14.7). Similar to the weights $w_{ij}$, these also follow a log-normal distribution. The digital pulses from the CCO are used to clock a counter which is enabled along with the neuron for $T_{\text{en}}$ seconds. Also, the counter can be stopped at a digitally programmable count value $h_{\max}$ which provides a saturating nonlinearity. Hence, the hidden layer output after the counter can be expressed as:

$$\begin{aligned} h &= f_{\text{CCO}} T_{\text{en}} \text{ if } f_{\text{CCO}} T_{\text{en}} < h_{\max} \\ &= h_{\max} \text{otherwise.} \end{aligned} \tag{14.12}$$

#### 14.3.3.3 Measurement Results

The chip described above was fabricated in 0.35 μm CMOS process. With 128 input channels and 128 hidden neurons, the die size of this chip was $4.95 \times 4.95 mm^2$. An example of the mismatch is shown in the variability in measured tuning curves of the hidden neurons (Fig. 14.9) when the input spike frequency of only one

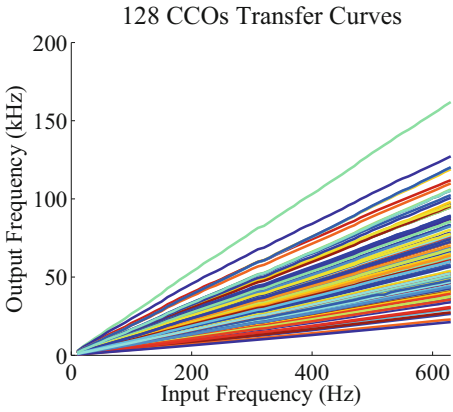


**Fig. 14.9** Measured transfer curves of the 128 hidden layer neurons on the chip obtained by sweeping the input spike frequency of one of the channels. The variation of the curves is due to statistical variations in the chip

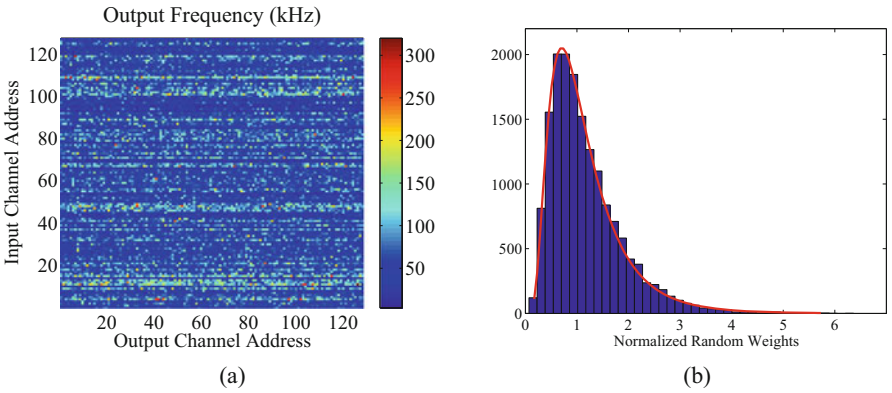


**Fig. 14.10** (**a**) A map of the threshold variation across the 128 × 128 synaptic current mirror transistors on one of the dies. (**b**) The weights due to mismatch fit a log-normal distribution as expected

channel is varied. A more detailed characterization of the mismatch across the entire synaptic array is shown in Fig. 14.10a. This figure is obtained by giving a fixed input frequency to each channel one by one and recording the hidden neuron firing frequency. These weights are fit to a log-normal distribution in Fig. 14.10b implying an underlying gaussian distribution of $\Delta V_T$. Across eight different dies, the mean of the gaussian distribution varies from −0.1 to 0.57 mV and the standard deviation varies from 16.2 to 17.6 mV.

The authors in [76] have applied the IC for decoding flexion and extension of fingers and wrist from neural activity recorded from the M1 region of a non-human primate. The experiment with the monkey is described in detail in [20]. In brief, monkeys are trained to move individual fingers and wrist based on visual input while simultaneously, a single-unit recording device implanted in the motor cortex is used to record the brain activity. This data contains information about the monkey's motor intention and is used for the decoding. The entire data set has experiments performed on three monkeys. This pre-recorded data was fed into the IC and the hardware performance has been benchmarked with software decoding results reported in [20].

Figure 14.11 shows an example of the decoding being performed—three different trials are shown. The bottom part of the figure shows neural spikes obtained after sorting from $N = 40$ M1 neurons. The middle panel shows the onset detection while the top panel shows predicted movement type. The authors reported that the decoding accuracy increases to $\approx 96\%$, at par with software results, for a hidden layer size of $L = 60$ neurons. It is also important to see how the decoding accuracy degrades when less number of biological M1 neurons are available for recording. This is shown in Fig. 14.12 for 8 different samples of the IC. It can be seen that using delayed samples to increase dimension (TDBDI) helps in boosting decoding accuracy for all samples. The result is specially significant when the number of M1 neurons is small. This clearly shows the benefit of TDBDI for chronic implants. For this IC, the authors report a power dissipation of 414 nW for the case of $D = 40$

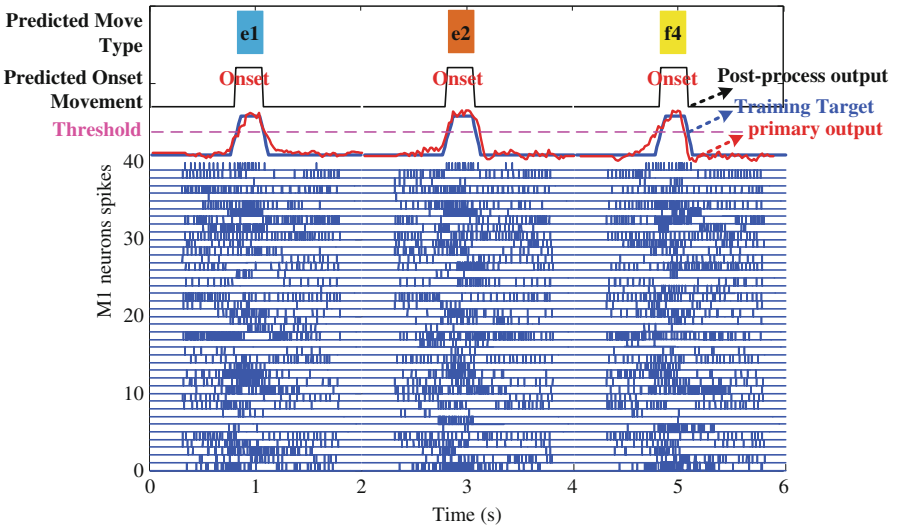


**Fig. 14.11** Example of a neural decoding trial where the chip uses $L = 60$ hidden layer neurons to decode the onset time and type of movement from $N = 40$ biological neurons recorded from the M1 region of a non-human primate. 12 types of movement are considered here—flexion and extension of five fingers and wrist

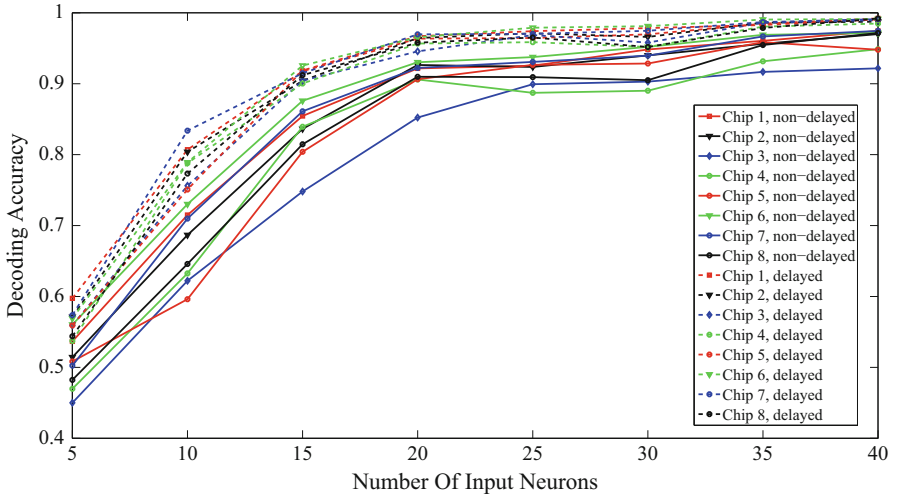


**Fig. 14.12** Using the time delayed samples for extra information helps in increasing decoding accuracy especially when the number of biological M1 neurons is small. The results are verified from 8 chips

and $L = 60$ resulting in an ultra-low energy per operation of 3.45 pJ/MAC where MAC refers to multiply and accumulate. This is much smaller than recently reported digital multiplier which requires 16–70 pJ/MAC [80–82].

We can now estimate the amount of data compression achievable in this mode of operation with an integrated neural decoder. In the beginning of a session, this system needs to transmit the raw data rate of $R_{\text{typ}}$ or $R_1$ or $R_2$. This data is used for training. Once trained however, the data rate $R_3$ to be transmitted is given by:

$$R_3 = f_{\text{deco}} \times \lceil \log_2(C) \rceil \tag{14.13}$$

where $C$ is the number of classes of movement and $f_{\text{deco}}$ is the rate of classification. As an example, for the case described earlier with $f_{\text{deco}} = 50\,\text{Hz}$ and $C = 13$, $R_3 = 200\,\text{bps}$ with a compression factor of $10^5$ over $R_{\text{typ}}$ showing the huge potential of compression obtainable this way.

## 14.4 Conclusion and Discussions

Implantable brain machine interfaces are an emerging area of research which can be used by patients with motor disabilities to interact naturally with prosthetics or devices such as wheelchairs. More broadly, neural implants can be used to treat other neural diseases such as Parkinson's, epilepsy or depression. In this chapter, we showed the issue of scaling neural implants to thousands of channels in the future stems from increasing wireless transmission rates of the order of 200 Mbps. It was also shown that it is possible to achieve variable rates of compression from 10–$10^5$ by incorporating more processing steps into the implanted chip as opposed to leaving it to the receiver module outside the body. To make this viable, the processing has to be done in ultra low power so that the power budget of the implant is not exceeded.

Neuromorphic or neuro-inspired analog circuits provide a viable alternative for reducing power dissipation beyond what is achievable from current digital circuits. In this chapter, we presented an extensive survey of the different levels of compression that are achievable when integrating spike detection, sorting or intention decoding within the neural implant. The most promising scheme for the future large scale implants—intention decoding—is described in great detail starting from the algorithm to chip architecture and details of sub-circuits. In the long term, we envision that as brain sensing technologies mature so that thousands of neurons can be simultaneously probed, integrated machine learners for intention decoding will become a common feature for managing the 'big data' originating from neural implants. However, to allow chronic or long-term recording using such devices, some challenges still need to be overcome. One of the major issues in long-term recordings is parameter drift such as change of probe impedance due to scarring or gliosis. Though the current solution has a feature of TDBDI to counter this, there is no automatic detection strategy of when to apply this and to which channels. This is a topic that deserves more attention in future. Also, the current method of training the machine learner used a trial structure where the time of movement was known—in real life operation, there will not be any such precise temporal markers and the training algorithm has to be modified to suit this. One promising possibility is reinforcement learning based training [83] but more work is needed in this direction.

Lastly, the current training paradigm used data from a monkey performing actual movements. To move to a prosthetic control using imagined movements only, there will be an aspect of visual feedback that will alter the neural data recorded by the chip—a phenomenon referred to as 'closed-loop' decoder training. In this case, we have to retrain the machine learner iteratively over several closed-loop experimental trials and convergence of such training for ELM based decoders is an open avenue for research.

**Acknowledgements** The authors acknowledge funding support from NTU and MOE, Mediatek for supporting chip design and Prof. Nitish Thakor for providing neural data from primate experiments.

## References

1. J. Wessberg, C. Stambaugh, J. Kralik, P. Beck, M. Laubach, J. Chapin, J. Kim, J. Biggs, M. Srinivasan, M. Nicolelis, Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. Nature **408**, 361–365 (2000)
2. L. Hochberg, M. Serruya, G. Friehs, J. Mukand, M. Saleh, A. Caplan, A. Branner, D. Chen, R. Penn, J. Donoghue, Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature **442**, 164–171 (2006)
3. L. Hochberg, D. Bacher, B. Jarosiewicz, N. Masse, J. Simeral, J. Vogel, S. Haddain, J. Liu, S. Cash, P. der Smagt, J. Donoghue, Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature **485**, 372–375 (2012)
4. M. Lebedev, M. Nicolelis, Toward a whole-body neuroprosthetic. Prog. Brain Res. **194**, 47–60 (2011)
5. M. Lebedev, M. Nicolelis, Brain-machine interfaces: past, present and future. Trends Neurosci. **29**(9), 536–546 (2006)
6. A. Kübler, B. Kotchoubey, J. Kaiser, J. Wolpaw, N. Birbaumer, Brain-computer communication: unlocking the locked in, American Psychological Association, Washington, DC, 2001
7. Human brain project official website https://www.humanbrainproject.eu/
8. The BRAIN initiative, NIH website http://www.nih.gov/science/brain/
9. K. Micheva, B. Busse, N. Weileer, N. O'Rourke, S. Simith, Single-synapse analysis of a diverse synapse population: proteomic imaging methods and markers. Neuron **68**, 639–653 (2010)
10. G. Buzsaki, K. Mizuseki, The log-dynamic brain: how skewed distributions affect network operations. Nat. Rev. Neurosci. **15**, 264–78 (2014)
11. T.M. Seese, H. Harasaki, G.M. Saidel, C. Davies, Characterization of tissue morphology, angiogenesis, and temperature in the adaptive response of muscle tissue in chronic heating. Lab. Invest. **78**, 1553–1562 (1998)
12. S. Kim, R. Normann, R. Harrison, F. Solzbacher, Preliminary study of the thermal impact of a microelectrode array implanted in the brain, in *Proceedings of IEEE Engineering in Medicine and Biology Conference* (2006), pp. 2986–2989
13. A. Usakli, Improvement of EEG signal acquisition: an electrical aspect for state of the art of front end. Comput. Intell. Neurosci. **2010**, 630649 (2010)
14. P. Konrad, T. Shanks, Implantable brain computer interface: challenges to neurotechnology translation. Neurobiol. Dis. **38**, 369–375 (2010)
15. A. Hoogerwerf, K. Wise, A three-dimentional microelectrode array for chronic neural recording. IEEE Trans. Biomed. Eng. **41**(12), 1136–1146 (1994)
16. C.T. Nordhausen, E.M. Maynard, R.A. Normann, Single unit recording capabilities of a 100 microelectrode array. Brain Res. **726**, 129–140 (1996)
17. A.L. Owens, T.J. Denison, H. Versnel, M. Rebbert, M. Peckerar, S.A. Shamma, Multi-electrode array for measuring evoked potentials from the surface of ferret primary auditory cortex. J. Neurosci. Methods **58**, 209–220 (1995)

18. V. Aggarwal, M. Mollazadeh, A.G. Davidson, M.H. Schieber, N.V. Thakor, State-based decoding of hand and finger kinematics using neuronal ensemble and LFP activity during dexterous reach-to-grasp movements. J. Neurophysiol. **109**(12), 3067–3081 (2013)
19. K. Rupp, M. Schieber, N.V. Thakor, Local field potentials mitigate decline in motor decoding performance caused by loss of spiking units, in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Chicago, Aug 2014, pp. 1298–1301
20. V. Aggarwal, S. Acharya, F. Tenore, H. Shin, R. Etienne-Cummings, M. Schieber, N. Thakor, Asynchronous decoding of dexterous finger movements using M1 neurons. IEEE Trans. Neural Syst. Rehabil. Eng. **16**, 3–14 (2008)
21. S. Acharya, F. Tenore, V. Aggarwal, R. Etienne-Cummings, M. Schieber, and N. Thakor, Decoding individuated finger movements using volume-constrained neuronal ensembles in the M1 hand area. IEEE Trans. Neural Syst. Rehabil. Eng. **16**, 15–23 (2008)
22. M. Velliste, S. Perel, M. Spalding, A. Whitford, A. Schwartz, Cortical control of a prosthetic arm for self-feeding. Nature **453**, 1098–1101 (2008)
23. V. Gilja, P. Nuyujukian, C.A. Chestek, J.P. Cunningham, B.M. Yu, J.M. Fan, M.M. Churchland, M.T. Kaufman, J.C. Kao, S.I. Ryu, K.V. Shenoy, A high-performance neural prosthesis enabled by control algorithm design. Nat. Neurosci. **15**, 1752–1757 (2012)
24. R. Harrison, The design of integrated circuits to observe brain activity. Proc. IEEE **96**(7), 1203–1216 (2008)
25. Y. Chen, A. Basu, L. Liu, X. Zou, R. Rajkumar, G.S. Dawe, M. Je, A digitally assisted, signal folding neural recording amplifier. IEEE Trans. Biomed. Circuits Syst. **8**(4), 528–542 (2014)
26. R. Harrison, A low-power integrated circuit for adaptive detection of action potentials in noisy signals, in *Proceeding of the 25th Annual International Conference of the IEEE EMBS* (2003)
27. W. Wattanapanitch, M. Fee, R. Sarpeshkar, An energy-efficient micropower neural recording amplifier. IEEE Trans. Biomed. Circuits Syst. **1**(2), 136–147 (2007)
28. R. Ginosar, Y. Perelman, Analog frontend for multichannel neuronal recording system with spike and LFP separation. J. Neurosci. Methods **153**, 21–26 (2006)
29. F. Shahrokhi, K. Abdelhalim, D. Serletis, P.L. Carlen, R. Genov, The 128-channel fully differential digital integrated neural recording and stimulation interface. IEEE Trans. Biomed. Circuits Syst. **4**(3), 149–161 (2010)
30. M. Mollazadeh, K. Murari, G. Cauwenberghs, N. Thakor, Micropower CMOS integrated low-noise amplification, filtering, and digitization of multimodal neuropotentials. IEEE Trans. Biomed. Circuits Syst. **3**, 1–10 (2009)
31. W. Wattanapanitch, R. Sarpeshkar, A low-power 32-channel digitally programmable neural recording integrated circuit. IEEE Trans. Biomed. Circuits Syst. **5**, 592–602 (2011)
32. R.R. Harrison, P.T. Watkins, R.J. Kier, R.O. Lovejoy, D.J. Black, B. Greger, F. Solzbacher, A low-power integrated circuits for a wireless 100-electrode neural recording system. IEEE J. Solid State Circuits **42**(1), 123–133 (2007)
33. M. Yin, D.A. Borton, J. Aceros, W.R. Patterson, A.V. Nurmikko, A 100-channel hermetically sealed implantable device for chronic wireless neurosensing applications. IEEE Trans. Biomed. Circuits Syst. **7**(2), 115–128 (2013)
34. J. Tan, W.S. Liu, C.H. Heng, Y. Lian, A 2.4 GHz ULP reconfigurable asymmetric transceiver for single-chip wireless neural recording IC. IEEE Trans. Biomed. Circuits Syst. **8**(4), 497–509 (2014)
35. S.X. Diao, Y.J. Zheng, Y. Gao, S.J. Cheng, X.J. Yuan, M.Y. Je, A 50-Mb/s CMOS QPSK/O-QPSK transmitter employing injection locking for direct modulation. IEEE Trans. Microwave Theory Tech. **60**(1), 120–130 (2012)
36. M. Chae, Z. Yang, M. Yuce, L. Hoang, W. Liu, A 128-channel 6 mW wireless neural recording IC with spike feature extraction and UWB transmitter. IEEE Trans. Neural Syst. Rehabil. Eng. **17**(4), 312–321 (2009)
37. C. Mead, Neuromorphic electronic systems. IEEE Proc. **78**(10), 1629–1636 (1990)
38. R. Sarpeshkar, Efficient precise computation with noisy components: extrapolating from an electronic cochlea to the brain. PhD thesis, California Institute of Technology, Pasadena, CA (1997)

39. P. Lichtsteiner, C. Posch, T. Delbruck, A 128 × 128 120dB 15us latency asynchronous temporal contrast vision sensor. IEEE J. Solid State Circuits **43**(2), 566–576 (2008)
40. V. Chan, S.-C. Liu, A. van Schaik, AER EAR: a matched silicon cochlea pair with address event representation interface. IEEE Trans. Circuits Syst. I **54**(1), 48–59 (2007)
41. G. Indiveri, E. Chicca, R. Douglas, A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Trans. Neural Netw. **17**(1), 211–221 (2006)
42. G. Indiveri, E. Chicca, R.J. Douglas, Artificial cognitive systems: from VLSI networks of spiking neurons to neuromorphic cognition. Cogn. Comput. **1**, 119–127 (2009)
43. S. Brink, S. Nease, P. Hasler, S. Ramakrishnan, R. Wunderlich, A. Basu, B. Degnan, A learning-enabled neuron array IC based upon transistor channel models of biological phenomenon. IEEE Trans. Biomed. Circuits Syst. **7**(1), 71–81 (2013)
44. B.V Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J.V. Arthur, P.A. Merolla, K. Boahen, Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. Proc. IEEE **102**(5), 699–716 (2014)
45. K. Boahen, Point-to-point connectivity between neuromorphic chips using address events. IEEE Trans. Circuits Syst. II **47**(5), 416–434 (2000)
46. S. Furber, F. Galluppi, S. Temple, L. Plana, The SpiNNaker project. Proc. IEEE **102**(5), 652–665 (2014)
47. Y. Enyi, C. Yi, A. Basu, A 0.7 V, 40 nW compact, current-mode neural spike detector in 65 nm CMOS. IEEE Trans. Biomed. Circuits Syst. **10**(2), 309–318 (2016)
48. J. Holleman, A. Mishra, C. Diorio, B. Otis, A micro-power neural spike detector and feature extractor in .13 $\mu$m CMOS, in *Proceedings of the IEEE Custom Integrated Circuits Conference*, Sept 2008, pp. 333–336
49. E. Koutsos, S.E. Paraskevopoulou, T.G. Constandinou, A 1.5 uW NEO-based spike detector with adaptive-threshold for calibration-free multichannel neural interfaces, in *Proceedings of the International Symposium on Circuits and Systems*, May 2013, pp. 1922–1925
50. Y.-G. Li, Q. Ma, M.R. Haider, Y. Massoud, Ultra-low-power high sensitivity spike detectors based on modified nonlinear energy operator, in *Proceedings of the International Symposium on Circuits and Systems*, May 2013, pp. 137–140
51. L. Liu, L. Yao, X. Zou, W.L. Goh, M. Je, Neural recording front-end IC using action potential detection and analog buffer with digital delay for data compression, in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, July 2013, pp. 747–750
52. Y. Perelman, R. Ginosar, An integrated system for multichannel neuronal recording with spike/LFP separation, integrated A/D conversion and threshold detection. IEEE Trans. Biomed. Eng. **54**(1), 130–137 (2007)
53. R.H. Olsson, K.D. Wise, A three-dimensional neural recording microsystem with implantable data compression circuitry. IEEE J. Solid State Circuits **40**(12), 2796–2804 (2016)
54. T. Horiuchi, T. Swindell, D. Sander, P. Abshire, A low-power CMOS neural amplifier with amplitude measurement for spike sorting, in *Proceedings of the 2004 International Symposium on Circuits and Systems*, vol. 4 (2004), pp. 23–26
55. T. Horiuchi, D. Tucker, K. Boyle, P. Abshire, Spike discrimination using amplitude measurements with a low-power CMOS neural amplifier, in *IEEE International Symposium on Circuits and Systems (ISCAS)* (2007)
56. A. Bhaduri, E. Yao, A. Basu, Pulse-based feature extraction for hardware-efficient neural recording systems, in *International Symposium on Circuits and Systems (ISCAS)*, Montreal, May 2016
57. R.Q. Quiroga, Spike sorting. Scholarpedia **2**(12), 3583 (2007)
58. R.Q. Quiroga, Z. Nadasdy, Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput. **16**(8), 1661–1687 (2004)
59. E. Stark, M. Abeles, Predicting movement from multiunit activity. J. Neurosci. **27**, 8387–8394 (2007)
60. V. Ventura, Spike train decoding without spike sorting. Neural Comput. **20**, 923–963 (2008)

61. S. Gibson, J. Judy, D. Markovic, Spike sorting: the first step in decoding the brain. IEEE Signal Process. Mag. **29**, 124–143 (2012)
62. V. Karkare, S. Gibson, C. Yang, H. Chen, D. Markovic, A 75 uW, 16-channel neural spike-sorting processor with unsupervised clustering, in *IEEE Symposium on VLSI Circuits Digest of Technical Papers* (2011)
63. A. Patil, S. Shen, E. Yao, A. Basu, Random projection for spike sorting: decoding neural signals the neural network way, in *Biomedical Circuits and Systems (BioCAS)*, Atlanta, Oct (2015)
64. V. Karkare, S. Gibson, D. Marković, A 130-W, 64-channel neural spike-sorting DSP chip. IEEE J. Solid State Circuits **46**(5), 1214–1222 (2011)
65. V. Karkare, S. Gibson, D. Markovic, A 75-$\mu$W, 16-channel neural spike-sorting processor with unsupervised clustering. IEEE J. Solid State Circuits **48**(9), 2230–2238 (2013)
66. A. Georgopoulos, J. Kalaska, R. Caminiti, J. Massey, On the relations between the direction of two-dimensional arm movements and cell discharge in primate motorcortex. J. Neurosci. **2**, 1527–1537 (1982)
67. A. Georgopoulos, J. Kalaska, R. Caminiti, J. Massey, Spatial coding of movement: a hypothesis concerning the coding of movement direction by motor cortical populations. Exp. Brain Res. Suppl. **7**, 327–336 (1983)
68. A. Georgopoulos, A. Schwartz, R. Kettner, Neuronal population coding of movement direction. Science **233**, 1357–1440 (1986)
69. W. Wu, M. Black, D. Mumford, Y. Gao, E. Bienenstock, J. Donoghue, Modeling and decoding motor cortical activity using a switching Kalman filter. IEEE Trans. Biomed. Eng. **51**, 933–942 (2004)
70. W. Wu, Y. Gao, E. Bienenstock, J. Donoghue, M. Black, Bayesian population decoding of motor cortical activity using a Kalman filter. Neural Comput. **18**, 80–118 (2006)
71. A. Brockwell, A. Rojas, R. Kass, Recursive bayesian decoding of motor cortical signals by particle filtering. J. Neurophysiol. **91**, 1899–1907 (2004)
72. S. Lin, J. Si, A. Schwartz, Self-organization of firing activities in monkey's motor cortex: trajectory computation from spike signals. Neural Comput. **9**, 607–621 (1997)
73. B. Rapoport, W. Wattanapanitch, H. Penagos, S. Musallam, R. Andersen, R. Sarpeshkar, A biomimetic adaptive algorithm and low-power architecture for implantable neural decoders, in *31st Annual International Conference of the IEEE EMBS* (2009)
74. B. Rapoport, L. Turicchian, W. Wattanapanitch, T. Davidson, R. Sarpeshkar, Efficient universal computing architectures for decoding neural activity. PLoS ONE **7**, e42492 (2012)
75. J. Dethier, V. Gilja, P. Nuyujukian, S.A. Elassaad, K.V. Shenoy, K. Boahen, Spiking neural network decoder for brain-machine interfaces, in *5th International IEEE/EMBS Conference on Neural Engineering (NER), 2011* (2011)
76. C. Yi, Y. Enyi, A. Basu, A 128 channel extreme learning machine based neural decoder for brain machine interfaces. IEEE Trans. Biomed. Circuits Syst. **10**(3), 679–692 (2016)
77. G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machines: theory and applications. Neurocomputing **70**, 489–501 (2006)
78. G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(2), 513–529 (2012)
79. P.R. Kinget, Device mismatch and tradeoffs in the design of analog circuits. IEEE J. Solid State Circuits **40**(6), 1212–1224 (2005)
80. Y. He, C.H. Chang, A new redundant binary booth encoding for fast 2n-bit multiplier design. IEEE Trans. Circuits Syst. I **56**(6), 1192–1201 (2009)
81. K.S. Chong, B.H. Gwee, J.S. Chang, A micropower low-voltage multiplier with reduced spurious switching. IEEE Trans. VLSI **13**(2), 255–265 (2005)
82. M. La Guia de Solaz, R. Conway, Razor based programmable truncated multiply and accumulate, energy-reduction for efficient digital signal processing. IEEE Trans. VLSI **23**(1), 189–193 (2015)
83. J. DiGiovanna, B. Mahmoudi, J. Fortes, J. Principe, J. Sanchez Co-adaptive brain machine interface via reinforcement learning. IEEE Trans. Biomed. Eng. **54**(64), 56–61 (2009)