Journal of Neural Engineering

TOPICAL REVIEW • OPEN ACCESS

Technical survey of end-to-end signal processing in BCIs using invasive MEAs

To cite this article: Andreas Erbslöh *et al* 2024 *J. Neural Eng.* **21** 051003

Journal of Neural Engineering

CrossMark

OPEN ACCESS

RECEIVED
17 July 2023

REVISED
13 August 2024

ACCEPTED FOR PUBLICATION
26 September 2024

PUBLISHED
15 October 2024

TOPICAL REVIEW

Technical survey of end-to-end signal processing in BCIs using invasive MEAs

Andreas Erbslöh[1,5,*], Leo Buron[1,5,*], Zia Ur-Rehman[2,5,*], Simon Musall[3], Camilla Hrycak[1], Philipp Löhler[1], Christian Klaes[2], Karsten Seidl[1,4] and Gregor Schiele[1]

[1] University of Duisburg-Essen, Duisburg, Germany
[2] Ruhr University Bochum, Bochum, Germany
[3] Research Centre Jülich, Jülich, Germany
[4] Fraunhofer Institute for Microelectronic Circuits and Systems, Duisburg, Germany
[5] These authors contributed equally.
[*] Authors to whom any correspondence should be addressed.

E-mail: andreas.erbsloeh@uni-due.de, leo.buron@uni-due.de and zia.ur-rehman@ruhr-uni-bochum.de

Keywords: extracellular recording, low-power electronic, spike sorting, neural decoder, deep learning, neural signal processing, embedded systems

Abstract

Modern brain-computer interfaces and neural implants allow interaction between the tissue, the user and the environment, where people suffer from neurodegenerative diseases or injuries.This interaction can be achieved by using penetrating/invasive microelectrodes for extracellular recordings and stimulation, such as Utah or Michigan arrays. The application-specific signal processing of the extracellular recording enables the detection of interactions and enables user interaction. For example, it allows to read out movement intentions from recordings of brain signals for controlling a prosthesis or an exoskeleton. To enable this, computationally complex algorithms are used in research that cannot be executed on-chip or on embedded systems. Therefore, an optimization of the end-to-end processing pipeline, from the signal condition on the electrode array over the analog pre-processing to spike-sorting and finally the neural decoding process, is necessary for hardware inference in order to enable a local signal processing in real-time and to enable a compact system for achieving a high comfort level. This paper presents a survey of system architectures and algorithms for end-to-end signal processing pipelines of neural activity on the hardware of such neural devices, including (i) on-chip signal pre-processing, (ii) spike-sorting on-chip or on embedded hardware and (iii) neural decoding on workstations. A particular focus for the hardware implementation is on low-power electronic design and artifact-robust algorithms with low computational effort and very short latency. For this, current challenges and possible solutions with support of novel machine learning techniques are presented in brief. In addition, we describe our future vision for next-generation BCIs.

Used abbreviations

AC	Alternative Current
ADC	Analog-Digital Converter
AFD	Aligned first derivative
ASIC	Application-Specific Integrated Circuit
ASO	Amplitude Slope Operator
AT	Amplitude Thresholding
BCI	Brain-Computer-Interfaces
CAR	Common Average Reference
CAOM	Cluster Accept and Merge
CCA	Canonical Correlation Analysis
CIFF	Cascade of Integrators with Feed-Forward
CL	Competitive Learning
CMOS	Complementary Metal-Oxide-Semiconductor
CPU	Central Processing Unit
CNN	Convolutional Neural Networks
DAC	Digital-Analog-Converter
DBS	Deep Brain Stimulation
DC	Direct Current
DSL	DC Servo Loop
ECoG	Electrocorticography
EDO	Electrode Drift Offset

EEG	Electroencephalograph
EMG	Electromyography
ENOB	Effective Number of Bits
EF	Error Feedback
EOC	End of Conversion
ESN	Echo State Network
FE	Feature Extraction
FIR	Finite-Impulse-Response
fMRI	Functional Magnetic Resonance Imaging
fNIRS	Functional Near-Infrared Spectroscopy
FSDE	First and Second Derivative Extreme
FPGA	Field Programmable Gate Array
GANs	Generative Adversarial Network
IIR	Infinite-Impulse-Response
KF	Kalman Filter
KLDM	Kullback-Leibler Divergence Minimization
LSB	Least Significant Bit
LFADS	Latent Factor Analysis via Dynamical Systems
LSTM	Long Short-Term Memory
MA	Mean Absolute
MAD	Median Absolute Derivation
MEG	Magnetoencephalography
MCU	Microcontroller
MSB	Most Significant Bit
MEA	Microelectrode Array
NEO	Nonlinear Energy Operator
NS-ADC	Noise-Shaping ADC
NTF	Noise Transfer Function
LFP	Local Field Potentials
OSR	Oversampling Ratio
OTA	Operational Transconductance Amplifier
PCA	Principle Component Analysis
PDAC	Peak Detection with Area Computation
PVT	Process, Voltage and Temperature
rEFH	Recurrent Exponential-Family Harmonium
ReFIT-KF	Recalibrated Feedback Intention-Trained KF
RMS	Root-mean-square
RNN	Recurrent Neural Networks
SAR	Successive Approximation
SDA	Spike Detection Algorithm
SFS	Salient Features Selection
SNN	Spiking Neural Networks
SNR	Signal-to-noise ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TM	Template Matching
Q-RNN	Quasi Recurrent Neural Network
VCM	Common-Mode Voltage
VKF	Velocity Kalman Filter
WD	Window Discrimination
ZCA	Zero-Phase Component Analysis

1. Introduction

Neurodegenerative diseases and injuries of the nervous system result in a reduction in the quality of life of patients. For many of these diseases, there are currently no long-term cures or treatments available. Today, neural devices can relieve symptoms and substantially increase patients' quality of life.

- Patients with Parkinson suffer from uncontrollable tremors, leading to significant restrictions in everyday movements. One treatment option is the deep brain stimulation (DBS) of the midbrain, in which dopaminergic neurons in the substantia nigra are stimulated electrically to recover motor control and reduce tremors [97].
- Patients with Retinopathia pigmentosa go blind in the long term due to the death of the photosensitive cell layers. Retinal implants with recording and stimulation capabilities can be used to restore sight by translating a data stream from an external camera into neural signals of the retina via electrical or optical stimulation [45].
- Spinal cord injuries can often cause severe paralysis which leads to restricted freedom of movement. By recording the activity of motor neurons in the brain, it is possible to predict movement intentions and control an exoskeleton [50] or a prosthesis [33].
- Patients with severe paralysis or cognitive disorders can also suffer from speech impairments, resulting in social isolation and a strong reduction in their quality of life. Here, brain-computer interfaces (BCIs) in the motor cortex can be used to record neural population activity and directly decode intended speech or handwriting patterns [138].

In all of these cases, a reliable and real-time closed-loop signal processing of neural activity is necessary. In addition, a deeper understanding of neural information coding in different brain structures is required to further optimize decoder techniques. This would enable improved recognition of movement intentions or speech patterns to control an actuator or enhance targeted neurostimulation for haptic feedback or restoration of impaired sensory. To allow the seamless integration of such approaches into regular daily life advances in the implementation of end-to-end processing pipelines and AI-powered decoding techniques for on-implant and wearable neural devices and BCIs are needed. With increased number of electrodes it is necessary to move the processing to the brain, because sending all digitized raw data from all electrodes would damage the tissue due to the needed transmission power. Thus, the number of features should be reduced as much as possible. Therefore, the main challenge is to transfer the methods from a remote processor to resource-restricted hardware platforms, like an application-specific integrated circuit (ASIC) which have the highest power efficiency and the highest resource-optimization. There, the algorithm have to be optimized on their power consumption, computational resources (memory, area) and latency. Thus, the memory and computational effort have to be minimized to fit on small devices with low power techniques. In addition, the algorithms need to be adaptable to allow robust performance over long time scales.

To achieve high accuracy and long-term robustness during runtime, sophisticated signal processing algorithms must be used which can be supported by state-of-the-art machine/deep learning techniques, A neural decoder is then used to isolate the relevant information in the biological neural network. Depending on the neural structure, different decoder techniques are required. Also, neural decoders can be used for adaptive or closed-loop stimulation to adjust the stimulation parameters during runtime to induce specific neural response patterns [71].

This survey paper provides an overview of system architectures and techniques for achieving a signal processing in BCIs by using penetrating microelectrode arrays (MEA). Here the techniques on different hardware platforms (workstation, embedded, on-chip) of the end-to-end pipeline are discussed, from the analogue pre-processing stage with recording the neural input, spike sorting with spike detection, framing, feature extraction and clustering through to decoding movement intentions. The structure of this review is as follows: Chapter II explains the neural input, the pipeline and the corresponding requirements for the hardware implementation in more detail. Chapter III mentions methods for analogue processing with a focus on quantisation. Chapter IV covers neural signal pre-processing with spike sorting. Chapter V presents methods for neural decoding of motion intentions and chapter VI gives a brief outlook on future work.

2. Concept of an end-to-end BCI

This section introduces the basics of neurosignals and the steps of a neural signal pipeline using in invasive BCIs. Therefore, this section is divided into (a) characteristics of detectable neurosignals, (b) a high-level description of a possible distributed system architecture of a neural signal processing pipeline and (c) an overview of the corresponding challenges and design requirements. This knowledge is still necessary for the next sections. Also, we also discuss two operation modes, offline versus online processing and their use cases.

2.1. Characteristic of neural signals

Brain activities can be captured via invasive technologies like penetrating microelectrode array (MEA), e.g. in the motor cortex [125]. From recording extracellular neural activities, two important biosignal features are available on each electrode channel: Local field potentials (LFP) and action potentials (or spikes). The LFP is the recording signal of the constructive superposition of many neuronal activities inside the neural tissue. Typical characteristics

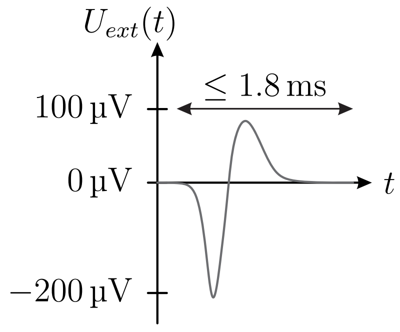

Figure 1. Timing of an extracellular spike waveform recorded from one channel of the invasive microelectrode array.

of this signal are peak-to-peak amplitudes $\hat{U}_{\mathrm{pp}}$ in the mV-range with low-frequency reactions in the time domain ($\hat{U}_{\mathrm{pp}} \leqslant 10\,\mathrm{mV}, 0.2\,\mathrm{Hz} \leqslant f \leqslant 200\,\mathrm{Hz}$). Spike activity serve as a stimulus transmission between individual neurons and are used for information transmission in neuronal networks ($\hat{U}_{\mathrm{pp}} \leqslant 500\ \mu\mathrm{V}, 100\,\mathrm{Hz} \leqslant f \leqslant 7\,\mathrm{kHz}$). Figure 1 shows an example of a spike waveform from extracellular recordings. Such types of spikes have typical values peak-to-peak amplitude up to 300 μV within a time window range between 1.2 ms and 1.8 ms.

In general, the spike shape depends on several factors of (i) electrode-tissue behaviour $\underline{H}_{\mathrm{tissue}}$ (healthy of the tissue, distance between electrode and neuron), (ii) impedance characteristic from the electrode, and (iii) characteristics of the analog preprocessing (noise, gain, filtering, input impedance). Formula (1) shows the input signal present at the preamplifier, which is the sum of (i) the noise voltage $\underline{U}_n$ through tissue/electrode and electronics and (ii) the voltage of the extracellular activities $\underline{U}_{\mathrm{ext}}$. The extracellular input is attenuated with

$$\underline{U}_{\mathrm{in,pre}}(t) = \frac{\underline{Z}_{\mathrm{pre}}}{\underline{Z}_{\mathrm{pre}} + \underline{Z}_{\mathrm{elec}}} \cdot \underline{U}_{\mathrm{ext}}(t) + \underline{U}_n(t)\,. \tag{1}$$

Therefore, the shape of the measured waveform from one neuron should be similar to the last few waveforms except for noise. However, the waveform may change over time due to electrode movement resulting in changed tissue impedance [7].

Typical values from recordings MEAs, like the Utah array by Blackrock Systems or Neuropixels by University College London, have an electrode impedance $\underline{Z}_{\mathrm{elec}}$ in the upper kΩ-range (e.g. Neuropixels with 149 kΩ at 1 kHz [22]). What these MEAs have in common is that the electrodes have a high impedance and a diameter of a few μm. This is necessary in order to be able to record neuronal activity well with high-density probes. For achieving a high signal quality, it is important that the input impedance of the pre-amplifier is 10-times larger than the electrode impedance.

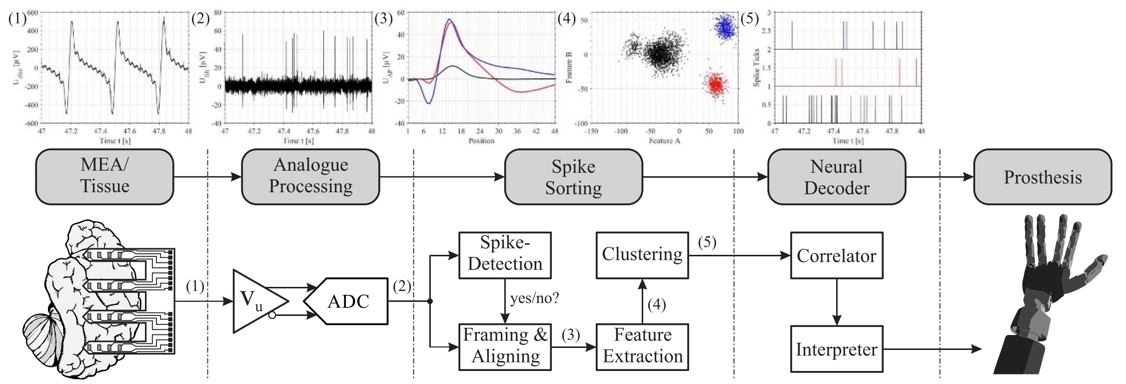

Figure 2. Illustration of an end-to-end neural signal processing pipeline from sensing the input from by multi-electrode array (MEA) over the analogue processing, the spike sorting as neural signal pre-processing and the neural decoding for readout the movement intentions from the motor cortex to control a prosthetic hand.

2.2. Signal processing pipeline

In processing the spike activity of the extracellular recordings, the activities from multiple neurons that are close to the same electrode are often measured together. Spike sorting techniques are therefore used to detect and isolate neural signals from individual cells. This is done by extracting various features from the measured input signal, such as the shape and magnitude of the spike waveform, and then clustering the spikes that originate from different neurons. When electrodes are placed nearby ($< 50\ \mu$m distance) the same spikes can also be recorded by multiple electrodes, strongly facilitating clustering performance by taking into account the spread of the spike waveform across recording sites [93]). The amount of recorded neurons strongly depends on the recorded brain region. For example, 26–47 neurons can theoretically be recorded within a radius of 50 μm around each electrode tip of the Utah array in the primary motor cortex in monkeys (neuron density varies from 50 000–90 000 neurons per mm^2 [144]) and even higher neuron density (300 000 neurons per mm^2) can be found in the rat hippocampus [40]. However, due to tissue perturbations upon electrode insertion and theoretical limitations in isolating low-magnitude spikes, the number of correctly identified neurons is usually limited to 8–10 neuron units per recording site [89]. Ideally, spike clusters map to individual neurons but in reality, the spikes of multiple neurons with weaker signals can be indiscernible and are therefore often combined in the same cluster [102]. The identified clusters are therefore usually described as multi- or single-unit activity, to indicate how likely they are to reflect the activity of a single neuron [93]. The clustering output results in a so-called spike train, a sequence of time points where spikes from a given cluster are detected.

Figure 2 presents a signal chain for processing spike activities within an end-to-end BCI or modern experimental tools, like Utah-Array [124], NeuroPixel probes [22] or NeuraLink's BCI-system [74]. In the following, the different steps of processing the bitstream of high-density MEA systems like NeuroPixels 2.0 (385 electrodes) are presented.

Analogue processing: All electrodes of the implanted MEA are connected to the recording front-end of the implant, in which all signals are first passed through the pre-amplifier, which removes some of the unwanted disturbances (e.g. movement artefacts, electronic noise) and the low-frequency LFP by band-pass filtering. Subsequently, the filtered spike activity can be digitized by an ADC with oversampling and noise-shaping to reduce the quantization noise of the ADC (signal in figure 2(2)). This raw data is sent telemetrically to a remote processor device [100]. For transmission of the raw data, high data rates in the upper MBit/s range are needed, which requires a high data transmission bandwidth and leads to high energy consumption. For example, the data rate per channel is in a range of 0.43 Mbit s^{-1} (NeuroPixels, 163.8 Mbit s^{-1} with 384 electrodes at 10-bit and 30 kHz) and 0.48 Mbit s^{-1} (Blackrock Cerebus combined with the Utah array, 184.8 Mbit s^{-1} with 96 electrodes at 16-bit and 30 kHz) by using external data acquisition system. With an implantable data acquisition system for the Utah array, the data rate is reduced to 0.16 Mbit s^{-1} [39] which provides long-term stable recording and strongly reduces energy consumption [16].

Spike sorting: The generated bitstream is pre-processed to reduce noise, artefacts, and cross-talk between recording channels. The resulting signal will be fed into the spike sorter pipeline. At this point, via a spike detection algorithm, a spike frame with the spike shape is captured from the bitstream (example in figure 2(3)) and processed in the next stage in order to determine the spike train. The spike

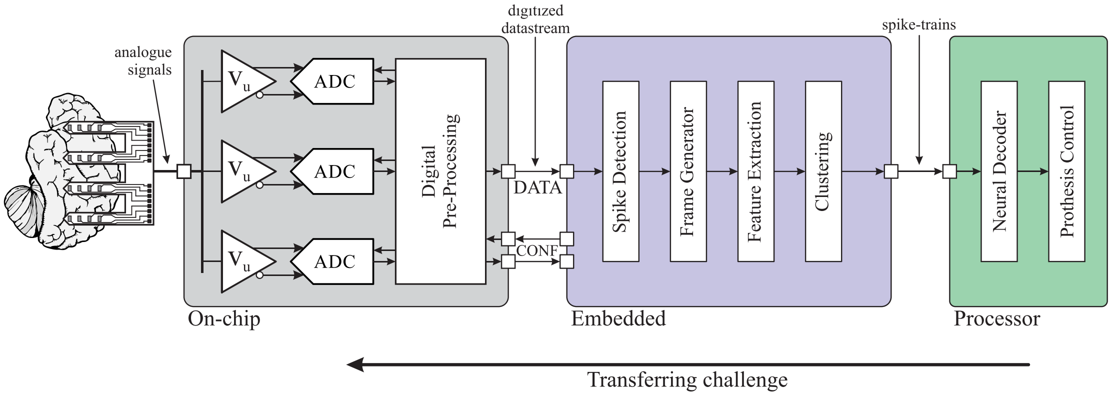

Figure 3. Illustration of the neural signal processing of the end-to-end BCI from system point-of-view in which the pre-amplifier has a voltage gain v_u and the analogue digital converter (ADC) for on-chip processing.

sorter is responsible for separating different neuronal responses from each other and from non-neural signals for each electrode usually implemented via feature extraction (FE) and clustering (example in figure 2(4)). In this process, specific characteristics (e.g. signal area, mean values, eigenvalues) are determined in a very computationally intensive way, and subsequently applied for clustering or classification to enable separation of spiking signals from individual neurons. To each detected time point of a spike frame, the corresponding cluster or classification number (spike-identifier) is determined to generate a spike tick (example in figure 2(5)). From the bitstream, a sequence of spike times from individual units is generated which serves as the input to a neural decoder.

Neural decoder: The resulting spike sequences from different neurons are then sent to the neuronal decoder, to allow the interpretation of movement intentions or responses to external stimulation from the neural activity. Another task of the decoder is that the detected spike frames can be assigned to the biological neuron type via a database to adapt the function of the existing neural structure. In addition, for long-term robust signal processing, several sensor inputs can be combined, e.g. EEG, ECoG, and LFP.

2.3. System design and requirements

The used modules of the end-to-end BCI pipeline from the analog processing to the neural decoder can be understood as a modular system. In each stage, different methods can be chosen or cascaded to perform spike sorting and interpret the resulting neural signals. Each module impacts the performance parameters like accuracy, latency, computational effort, and total power consumption. For example, integrating the spike sorter into a wearable platform or ASIC could impair the sorting accuracy but significantly reduce the data rate by up to 600-fold per channel by directly transmitting spike trains instead of raw data bitstream. Moreover, integrated spike sorting strongly decreases the latency, and power consumption of closed-loop applications. This survey paper gives an overview of these modular methods for performing spike sorting on different hardware systems. Figure 3 shows an example of a state-of-the-art end-to-end BCI which also describes the trend of transferring all necessary algorithms from the remote processor to the on-implant electronic. This can be done in three stages. Firstly, they are developed on workstations with high computational power and full data quality (Datatype: float, high sampling rates, ...). Secondly, the methods are optimized for a wearable device with low computational power and quantized input. Finally, more power-, memory- and latency optimizations take place in order to implement these methods into an ASIC for in-body neural devices. For a unique comparison, we distinguish where the calculations are performed because the position of the computing platform favours different hardware. Thus, we distinguish between (i) on-implant electronics, usually ASICs, (hereinafter referred to as on-implant), (ii) an on-body wearable device, usually wearable computing platforms like MCUs and FPGAs, (referred to below as wearable), and (iii) a remote processing workstation (hereafter called remote processor). Also, we differentiate between ii.a) data processing in real-time (online) or ii.b) data analysis after measurements (offline). In the future, to increase patients' life quality, the end-to-end BCI pipeline should run on implanted hardware or wearable hardware. Therefore, this paper has three main contributions.

- Classification of the used analogue processing for digitizing neural activity.
- Classification of spike detection, feature extraction and clustering algorithms for on-implant online spike sorting.
- Share our vision on the future of on-implant online spike sorting. Focus on the transition from remote to wearable to on-implant online spike sorting.

- Explanation of system architectures for neural signal processing in invasive BCIs.

The pipeline must be robust against environmental changes which include artefacts from muscle activity or electrical stimulation, modulation of signal shape due to bursting spike activity [18, 81] or electrode drift. The electrode drift changes in the signal shape due to physical movements of the electrode in the tissue. Some micrometers leads to a significant drop of the amplitude.

3. Digitization of neural input

The first step for invasive BCIs is to digitize the input signal to enable neural signal processing, like spike detection or sorting, in digital manner. Such a recording front-end consists of analogue circuit and the design is crucial for the whole signal processing pipeline in wearable systems. The design choices have a huge impact on the signal quality and integrity. Thus, we discuss each component of recording front-end in detail with the related requirements.

The analogue front-end of neural recording units can be divided into three modules: (i) Amplification and Filtering, (ii) Analogue-digital conversion and (iii) Compressed sensing. The last case includes other pre-processing techniques for artefact suppression and data-rate reduction which is discussed in the spike sorting chapter. At the end of this section, the topologies of recording front-ends for different MEAs is presented (low- vs. high-density). Figure 4 shows these modules including the relevant circuit topologies and methods, which is described in the following. In general, the research goal is to work on new system topologies in which there is an optimum between small chip area, low power consumption and low effective input noise with simultaneously high accuracy and artefact suppression for the following pipeline stages.

3.1. Analogue amplification and filtering

The module of amplification and filtering in figure 4 shows the related topologies. These pre-amplifiers have a band-pass filter characteristic in order to capture the neural input signal with the following requirements.

- **Input impedance:** To avoid the signal attenuation and damages at the electrode, the input impedance of the pre-amplifier should be 10-times larger than the electrode impedance to prevent charge transfers into the electrode which causes an accelerated electrode ageing ($\geqslant 10\,\mathrm{M}\Omega$).
- **Input noise:** To achieve high signal-to-noise ratios, the effective input noise of the pre-amplifier $\underline{U}_n$ should be less then 5 μV in the filter bandwidth in order to have the electrode noise as a primary source.
- **Input offset:** The pre-amplifier should be robust against the electrode drift offset (EDO) and stimulation artefacts which moves in the range up to 100 mV [12].

For this, a simple amplifier can be realised with a one-transistor inverter, but this topology is highly sensitive to process, voltage and temperature variations (PVT). Also, the output voltage is very sensitive against changes on the power supply and it requires a DC voltage at the input for setting the working point which is not recommended for use in neural recording of high-density MEAs.

The impact of these drawbacks can be reduced by using feedback circuits with operational transconductance amplifier (OTA) combined with differential signal processing. A often-used topology is the capacitive-coupled amplifier with the circuit diagram in figure 5(a). The midband gain is set over the capacity ratio $C_{\mathrm{in}}/C_{\mathrm{fb}}$ and the corner frequency of the low-pass f_{HP} can be adjusted via the transconductance of the OTA g_m and the capacity load at the output C_{L}. To achieve a high-pass corner frequency in the lower Hz-range, high ohmic resistors in the TΩ-range are realized with pseudo resistors but they are highly PVT-sensitive [34]. In [20], a tuneable version is presented which allows to modify the high-pass corner frequency to the desired value and is PVT-robust. In general, these amplifier topologies is often used for low-power applications due to low charging current of the high-impedance capacities. This results in reducing the bias current of the OTA to values in the nA-range in order to save power and chip area by using the g_m/I_{D} design methodology. The final OTA design depends on the desired noise characteristic which requires large transistor areas of the load and the differential stage in order to reduce the level of thermal and $1/f$ noise. The disadvantages of this amplifier is that (i) no DC input processing is possible and (ii) input impedance ($\underline{Z}_{\mathrm{pre}} = (2\pi\, f\, C_{\mathrm{in}})^{-1}$) in the lower M$\Omega$-range are available. This range is not sufficient to avoid a signal attenuation and a charge transfer into the tissue.

To handle DC voltages, chopper-stabilized amplifiers are more effective due to the modulation and the demodulation of the input signal. Chopping takes place via polarity-shifting switches that perform amplitude modulation with a square wave function via a digital clock. This causes a conversion from DC- to AC-signal and the other way around. Here, the carrier frequency is at the chopper frequency f_{ch}. In order to minimize the output offset, the duty cycle of the digital clock should be exactly 50%. At the output of the amplifier, the signal is a DC signal again and parasitic properties of the OTA (e.g. noise, offset, ...) are modulated up to f_{ch} which can be removed by a low-pass filter. This results in fewer design constraints of the OTA (smaller chip area and power consumption with the same noise characteristics)

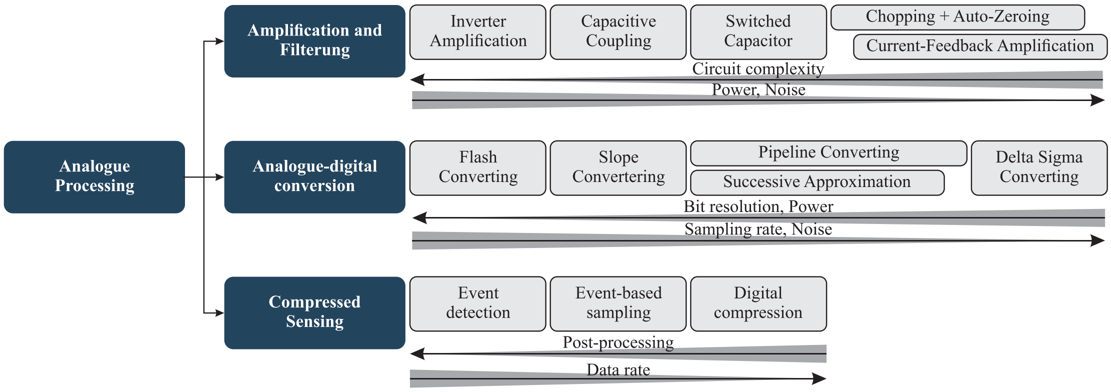

Figure 4. Overview of the module Analogue Processing for digitizing and pre-processing the neural input with comparing different methods of (i) amplification and filtering, (ii) analogue-digital conversion and (iii) compressed sensing on common metrics (High ⟶ Low).

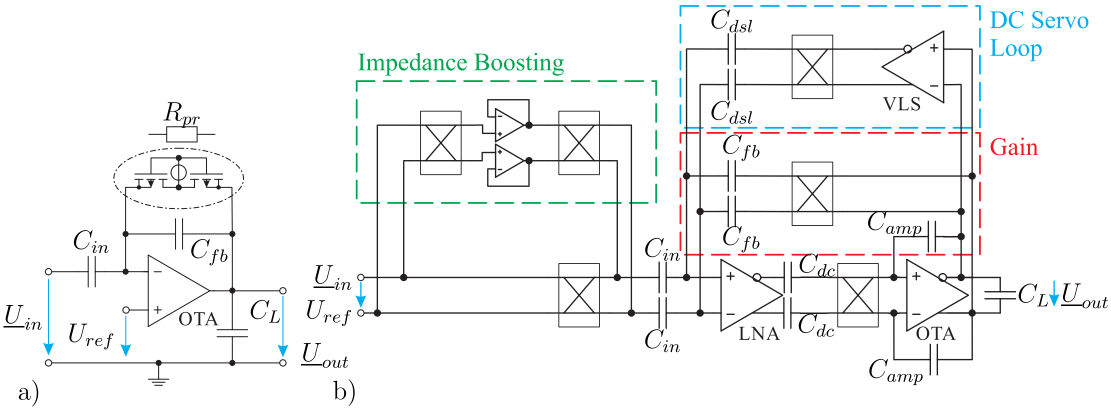

Figure 5. (a) CMOS circuit of a capacitive-coupled amplifier with pseudo resistance settling a low-frequency high-pass corner frequency (single-ended) - (b) CMOS circuit of capacitive-coupled chopper amplifier with input impedance boosting at the input and DC servo loop (DSL) in the feedback.

in order to compensate the increased circuit complexity. An auto-zero amplifier should be added in order to improve the noise properties at very low frequencies [90]. The disadvantages of choppers are that (i) the input impedance is even lower than with capacitive-coupled amplifiers ($f_{\mathrm{ch}} > f_{\mathrm{sig}}$) [60, 101] and (ii) due to the switching of the parasitic capacities from the switches, the charge current will generate voltage ripples on the output. The impact of (ii) can be reduced by using small transistors and reducing the chopping frequency f_{ch} [25].

Figure 5(b) shows the CMOS circuit diagram of the chopper-stabilized amplifier for neural application [79, 91]. Chopping takes place around the first low-noise OTA stage with a settable gain over the ratio $C_{\mathrm{in}}/C_{\mathrm{fb}}$. A DC servo loop (DSL) is used in the feedback for applying a high-pass filter characteristic in order to eliminate the electrode drift offset (EDO) in the range of $\pm 100\,\mathrm{mV}$. The desired corner frequency depends on the 0 dB-frequency $((R_{\mathrm{pr}}\ C_{\mathrm{int}})^{-1})$ of the integrator and the integrator gain $C_{\mathrm{dsl}}/C_{\mathrm{in}}$. The low-pass filter corner frequency is set by the OTA conductance g_m and the load capacity C_L at the output. The capacity C_{DC} adds an high-pass filter for blocking charge currents due to the active DSL in order to attenuate the ripples on the output voltage up to 60 dB [11]. A further method to reduce output ripples is to shift a time delay to the demodulator clock signal by the time constant of the ripple.

The module of impedance boosting tackles the problem of the low input impedance from previous topologies. A boosting can be achieved by adding (i) a positive feedback loop and (ii) an impedance buffer. In the following, the two methods are discussed briefly. The positive feedback loop is implemented easily by an additional path from the gain feedback to the input (left connection of the input capacity C_{in}) which decreases the effective input capacity by $(1 - C_{\mathrm{pfb}}/C_{\mathrm{fb}})$. The ratio $C_{\mathrm{pfb}}/C_{\mathrm{fb}}$ must be nearly zero in order to increase the input impedance and to avoid instability [109] which can not be avoided absolutely due to the PVT changes. To prevent this instability problem, the method of impedance boosting can be used. Here, an additional voltage buffer path is included at the input in order to charge the input capacity for certain time points of the chopping [100].

This requires a changing of the modulator clock signal in which a dead time between each clock edge is added. During this dead time, the input capacity are charged from the buffers and in the rest time from the electrode input. With this technique, the initial input impedance can be increased exponentially by the ratio of the chopping period T_{ch} to the dead time ΔT [109]. The bottleneck of this method is that using two different voltage buffers occurs to different offset voltages on both paths. This leads to ripple artefacts on the output due to the DSL. A method to compensate this is reported in [100] on which a fully-differential buffer with with switching properties is used. The benefits are a reduction of the offset voltage from mV-range to μV-range and the offset on both signal parts are identical.

A further reduction in chip area and noise properties of chopping amplifiers can achieved by changing the amplification over the capacity feedback to a current feedback [109]. Therefore, also the input capacity is lower then capacitive-coupled chopper amplifiers which leads to reduced output ripples due to less charge current at the input. In addition, a decoupling between gain and input impedance is available.

3.2. Analogue-digital converter

After amplification and filtering of the neural activities, these signals can be converted from the analogue domain into the digital domain for further processing within the end-to-end BCI. Figure 4 shows different analogue-digital conversion (ADC) techniques for neural application. In general, after the conversion the digital signal is presented over the ratio of the input signal to the voltage of the least significant bit (LSB) which depends on the applied voltage reference ΔU_{ref} $(= U_{\mathrm{ref}P} - U_{\mathrm{ref}N})$ and the ADC bit-resolution N. The residual voltage $U_{\mathrm{in}} - U_{\mathrm{ADC}}$ is a converting error or defined as quantization noise. For ideal converters, it moves in the range of $\{-1/2, 1/2\} \cdot U_{\mathrm{LSB}}$. This has an impact on the effective input noise of the whole recording pipeline via (2).

$$U_{n,\mathrm{eff}} = \sqrt{{U_{n,\mathrm{elec}}}^2 + \left(\frac{U_{n,\mathrm{amp}}}{v_u}\right)^2 + \left(\frac{U_{n,\mathrm{ADC}}}{v_u}\right)^2} \tag{2}$$

To achieve a total effective input noise of 20 $\mu\mathrm{V}_{\mathrm{eff}}$ with an electrode noise of 18 $\mu\mathrm{V}_{\mathrm{eff}}$ and an amplifier noise of 5 $\mu\mathrm{V}_{\mathrm{eff}}$, the input-related LSB voltage must be lower then 7.74 $\mu\mathrm{V}_{\mathrm{eff}}$. This can be achieved with an 15-bit ADC at a reference voltage ΔU_{ref} of 1.8 V and a gain v_u of 20 V/V.

Figure 4 shows different techniques for analogue-digital conversion. Regarding to these estimation with the required bit resolution at sampling rates up to 3 kHz with an high energy-efficiency, only the methods of successive approximation (SAR) and delta sigma converting ($\Delta\Sigma$) are suitable for the neural applications applications. In the following, these two topologies are presented shortly.

Figure 6(a) shows the setup of a first-order $\Delta\Sigma$ converter. It integrates the difference between the input signal and the feedback signal over time. The corresponding output voltage is compared with a reference voltage by using a clocked comparator to generate a digital 1-bit bitstream $D_{\Delta\Sigma}$. This digital signal updates the feedback voltage between the reference voltages $\{U_{\mathrm{ref}P}, U_{\mathrm{ref}N}\}$ of the ADC. This effects in a regulation of the difference voltage on the input to zero over the running time. In order to extract the analogue information from the pulse-density bitstream, a low-pass filtering in the digital domain is applied. This structure allows high bit resolution up to 24-bit with high accuracies, but this requires a very high oversampling rate (OSR) in combination with a decimation filtering and it needs a high-order modulator with multiple feedbacks to avoid stability problems. The big advantage of these structure is, that the integration of the residual voltage $\Delta\Sigma$ causes a noise transformation, where the quantization noise is shifted from the low frequency range to higher frequencies. Due to the low-pass filtering, the impact of the shaped noise is suppressed. However, such converters have a high static power consumption and are usable in application for low sampling rates.

Figure 6(b) shows the setup of a SAR-ADC, consisting of a comparator, a capacitive digital-to-analogue converter (C-DAC) to generate an internal reference voltage U_{sar} and the SAR logic. The advantages of the SAR ADC are that the power consumption is fully dynamic, the circuit complexity is lower and the design can be transferred quickly to smaller technology nodes. The disadvantage is that the resolution depends on the SAR logic and the bit resolution of the C-DAC. The SAR logic performs the binary search for driving the binary-weighted capacitances of the C-DAC to generate the digital output D_{ADC} in N conversion steps. The aim of the binary search is, that the difference between the input signal U_{in} and the generated voltage signal U_{sar} is closely to zero. This search is starting with the most significant bit (MSB) and it is updated from the result of the comparator from each conversion step. Control techniques of the binary search like the splitting the MSB-arrays and common-mode voltage (VCM)-based recovery [63] reduces the energy consumption per conversion with 99,53% and it leads to an area reduction up to 75% for the same bit-resolution.

Nowadays, the integration of the noise-shaping method into SAR ADCs is possible which results into a higher effective bit-resolution, reduced quantization noise (increased SNR) [48] and it enables an error-reduction method. This type of converters combine the benefits from $\Delta\Sigma$- (low noise) and SAR ADC (higher speed, high energy efficiency, low circuit complexity), which is a potential candidate for edge computing in Internet of Things- (IoT)

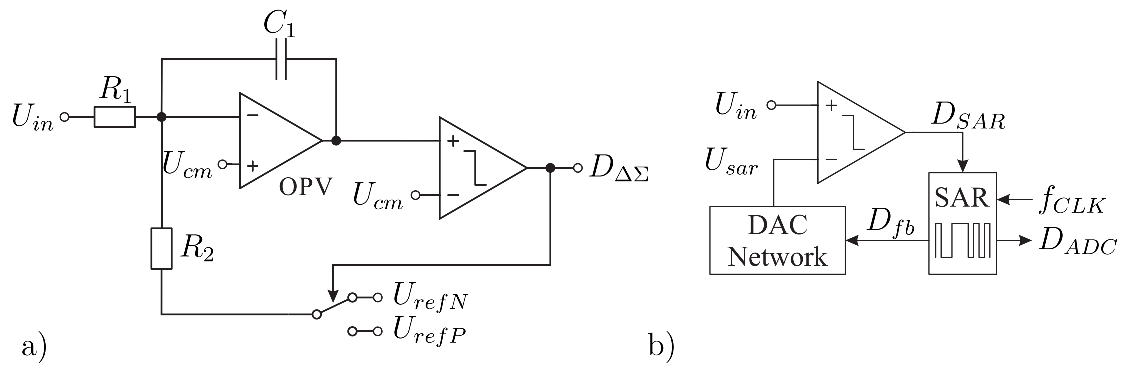

Figure 6. Electrical setup of (a) a delta-sigma analogue-digital converter (ADC) and (b) a successive approximation (SAR) ADC.

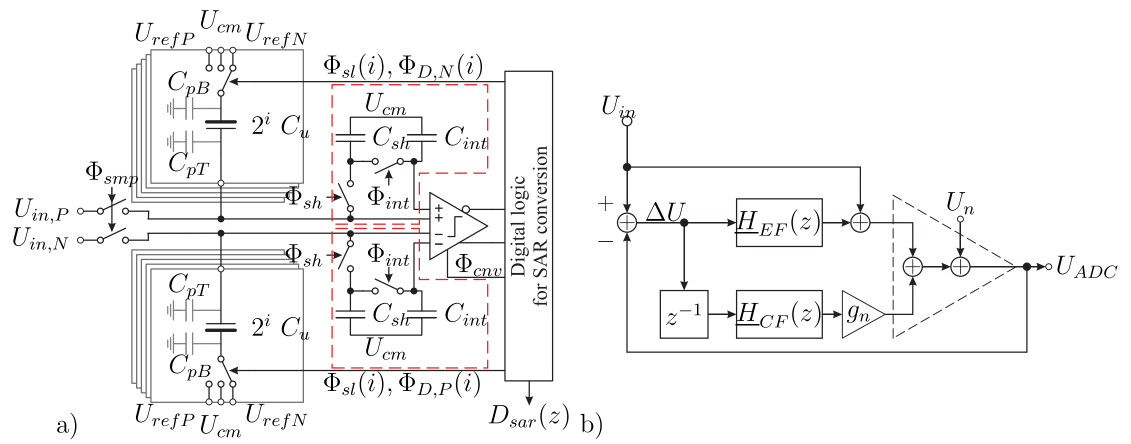

Figure 7. (a) CMOS circuit of the first order noise-shaping SAR ADC (extension in red lined box) with 4-input dynamic latch comparator - (b) Signal flow diagram for a EF and CIFF NS SAR-ADC with a 4-input comparator and a signal processing for the feedbacked residual voltage ΔU (without digital decimator and low-pass filter at the output).

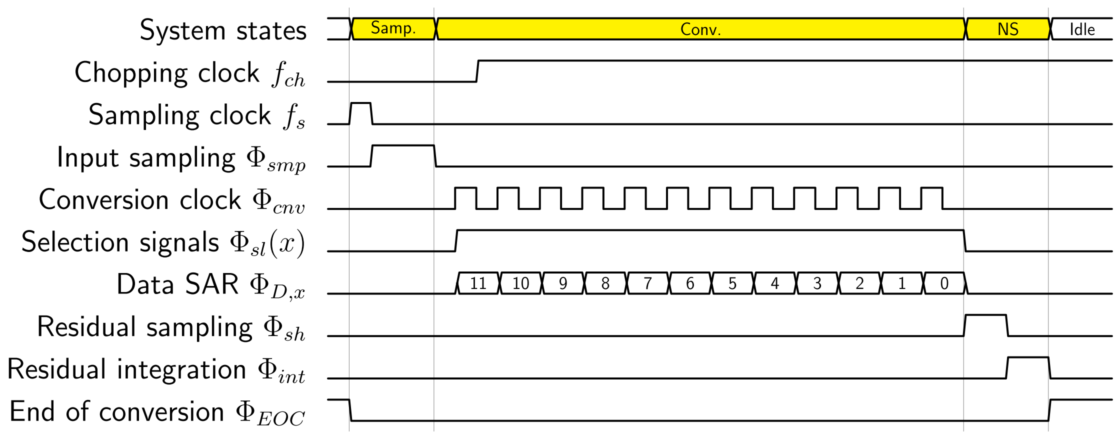

Figure 8. Timing diagram of the Successive Approximation Analog-Digital Converter (SAR ADC) control with noise shaping of the residual voltage ΔU after each conversion step.

and medical applications. In the following, the circuit implementation of a noise-shaping SAR-ADC (NS-ADC) is presented shortly. The CMOS circuit is shown in figure 7(a) with the corresponding timing diagram in figure 8 of the control signals. In general, the input signal can be applied to the top-plate and bottom-plate of the DAC capacities. The benefit of the top-plate charging is that a higher linearity can be achieved due to the lower parasitic capacities from the fabrication point-of-view, but for input sampling the input switches must be bootstrapped in order to reduce the parasitic discharging of the transistor. Important for the chopping pre-amplifier is that the impact of chopper artefacts/ripples on ADC output can be reduced by shifting the transition of the chopper to the conversion phase. In this time duration, the pre-amplifier is disconnected from the ADC input via the bootstrapped switch [99]. The execution steps of one full conversion including noise-shaping (additional circuits in the red box) are described below.

- **Sampling:** The conversion starts with the incoming flag of the sampling clock signal. During the sampling signal Φ_{smp}, the top-plates of the DAC-capacities are charged with the input signals $U_{\text{in},P}, U_{\text{in},N}$ on each side against the voltage U_{cm}.
- **Conversion:** During the conversion, the bineary search is performed for an N-bit ADC output in N conversion steps. The most significant bit (MSB) is decided directly in the first step over the sign of the voltage difference from the input signals. In the residual steps, the difference from the input signal and the reference voltage is generated in dependency of the results from the previous conversion steps via setting the capacities. The corresponding voltage shift depends data signal of each bit $\Phi_{D,x}(i)$ of the active selection signal $\Phi_{\text{sl}}(i)$.
- **Noise-shaping:** After the conversion, the residual voltage of the top-plate capacitances represents the error voltage of this ADC sampling event. This voltage will be stored on an additional capacitance C_{sh} which modifies the comparator outputs of the next conversion phase.
- **End of conversion (EOC):** After N conversion steps, the data word $D_{\text{sar}}(z)$ is determined and the signal Φ_{EOC} is active during the idle time before the next conversion will be triggered.

For performing noise shaping, different types of processing the residual voltage are available: residual integration with a cascade of integrators with feedforward (CIFF) and residue compensation with error feedback (EF). Both methods requires the residual voltage ΔU after a complete conversion. It builds up from the difference of the applied input voltage and the determined SAR output D_{sar} (see (3)).

$$\Delta U = U_{\text{in}}(z) - U_{\text{LSB}} \cdot D_{\text{sar}}(z). \tag{3}$$

In CIFF NS-ADCs, the residual voltage is sampled during the phase Φ_{sh} and integrated during the phase Φ_{int} Via a switched capacity circuit. This signal will be applied to the second comparator input for the next conversion step in which the comparator decision will be slightly adapted with the gain g_n in order to shift the comparator noise U_n to higher frequencies and to compensate DAC mismatches/errors during the runtime. Figure 7(a) shows the CMOS circuit diagram and figure 7(b) shows the signal flow diagram of such a CIFF NS-ADC.

$$D_{\text{out}}(z) = U_{\text{in}}(z) + \frac{U_n(z)}{1 + g_n\, z^{-1}\, \underline{H}_{\text{CF}}(z)} \tag{4}$$

$$D_{\text{out}}(z) = U_{\text{in}}(z) + \left(1 - \underline{H}_{\text{EF}}(z)\right) U_n(z) \tag{5}$$

$$\rightarrow D_{\text{out}}(z) = U_{\text{in}}(z) + \underbrace{\left(1 - z^{-1}\right)}_{=\text{NTF}} U_n(z). \tag{6}$$

The effectiveness of the noise shaping depends on the transfer function of the integrator $\underline{H}_n$ in the feedback. Formula (6) shows the digital output of the NS-ADC with the output result in which the noise transfer function (NTF) can be modified. With an ideal integrator ($\underline{H}_n = (1 - z^{-1})^{-1}$) the NTF is transformed into a first order high pass order. An improvement can be achieved by increasing the order number and by changing the transfer function in order to fit an optimum between in-band noise and out-of-band noise with FIR-IIR filtering [134]. Modern implementations are still using passive integration and summation in order to achieve minimum power consumption and to have a scaling-friendly technology [133].

$$\text{FoM}_w = \frac{P_{\text{lgc}} + P_{\text{dac}} + P_{\text{cmp}}}{2^{\text{ENOB}} \cdot \max(f_s)} \tag{7}$$

$$\text{ENOB} = \frac{\text{SNDR} - 1.72\text{dB}}{6.02\,\text{dB}} \tag{8}$$

$$\text{FoM}_s = \text{SNDR} + 10 \cdot \log\left(\frac{\text{GBW}}{P_{\text{tot}}}\right) \tag{9}$$

Important key metrics for the ADC characterization are the Walden figure of Merit FoM_w and Schreier FoM_s. The Walden FoM in (8) describes the power efficiency of ADCs in dependency of power consumption for one conversion cycle ΣP_x, the effective number of bits (ENOB) and the maximum sampling rate. With NS-ADC, a minimum FoM_w of 4.63 fJ/conv-step is achieved [118]. The FoM_s in (9) includes the harmonic distortion and SNR to the speed and power consumption. Here, the highest value of 183 dB are accessible. These values are achieved with passive NS SAR ADC (CIFF) by using capacitive stacking for summation and dynamic floating inverting buffers for sampling the residual voltage [133].

3.3. Compressing techniques

The last row of figure 4 shows some techniques for using compressed sensing are shown. In neural applications, an effective way to reduce the data rate is the introducing of the event detection which can also be used for event-based sampling. This can be done with the integration of an analogue spike detection, which is discussed in section 4.6.

Here, we want to mention the used compression technique in combination of an event detection in [73]. They present a method for robust readouts

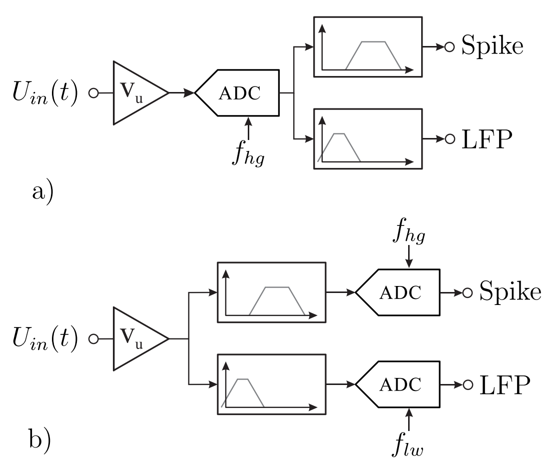

Figure 9. Different strategies in order to get spike activity and the low field potential from the neural input $U_{\mathrm{in}}(t)$: (a) Digital filtering - (b) Analogue filtering.

from massively parallel recordings with a data-rate reduction of 40x, in which each channel of the MEA has a single-slope ADC. The active readout of the neural event takes place if the value of the ADC output is outside of the estimated noise distribution. The reconstruction of the event takes place on a wearable device in order to reconstruct the spike waveform.

3.4. Topologies of recording front-ends

This section presents different often-used approaches for neural signal processing. Figure 9 shows two recording front-end topologies for different kind of applications, in which the separation of LFP and spike activity is done with filtering in digital domain (a) or in the analogue domain (b).

The benefit of a) is, that the implementation is very resource-efficient, in which the electronic of these probes have only one pre-amplifier and one ADC per channel or time-multiplexed-ADC for N number of channels. The disadvantage is, that the dynamic range of the LFP is dominant and higher bit resolution are required for achieving a high resolution of the spike activity. These structures are used in the Utah-Array on the external headstage or neural probes with electrode depth control via electrode time-multiplexing [113].

The major change of structure in figure 9(b) is, that the splitting of the LFP and spike activity is done in the analogue domain with a second amplifier stage. Each line has its own ADC with different sampling rates and bit resolution. These topologies are used in high-density MEA approaches like the NeuroPixel [22] or in neural systems for ECoG applications [100]. The benefit is that the focus is on achieving high signal quality for both biosignals. This is effected at the cost of a higher power and space requirement.

In future, there is research on novel approaches to integrate amplification and filtering directly into the ADC structure (direct digitisation), so that the necessary chip and power consumption for high-density applications can be further reduced [52, 119]. Also, inference effects and crosstalk between the different channels can be reduced. Furthermore, new neural probes must be developed in which the pre-amplifier or the hybrid-ADC is directly placed at the electrode directly for avoiding long wire cables.

When an electrical stimulation front-end is also implemented in order to provide an information flow into the tissue, then a methods for suppressing stimulation artefacts must be included. During the stimulation phase, an absolute change in the electrode voltage in the upper mV range is available. Without any action, the pre-amplifiers go into saturation and a recording is after a long settling time possible again [11]. This effect can be reduced by using the (i) blanking, (ii) pole-shifting and (iii) adaptive substraction method.

With the blanking method, the input of the pre-amplifier is switched from the active electrode to a reference during the stimulation period. This method is not effective because small voltage differences such as the electrode offset and residual charge on the electrode from the stimulation artefacts can lead to saturation. A better method is the pole shifting method, in which the high-pass corner frequency of the pre-amplifier is increased to high frequencies which results in a low total gain and the input of the pre-amplifier remains actively connected to the electrode [24]. The corresponding dead time is in the lower of μs range so that the responses can subsequently be recorded [98]. The adaptive substraction method tracks the stimulation-induced voltage change during the stimulation phase and adapts the pre-amplifier input in order to eliminate the artefact [104].

4. Spike sorting

After digitization of the neural input from extracellular recordings, the raw data must be processed in order to detect spike activity and to perform spike sorting for separating several neuron activities in order to get a spike-train for further neural decoding. This processing step is important because it compresses the number of features for the neural decoding drastically. Figure 10 shows an overview of different methods for pre-processing, spike detection, feature extraction and clustering in order to build a modular spike sorting pipeline. Each of the presented techniques is crucial for deploying a online signal processing pipeline. Therefore, we classify the techniques into their target application of the hardware location (remote, wearable, on-implant). Building on these classes, we provide detailed information about needed system architectures. First, we present and discuss the methods for each step of spike sorting starting with pre-processing in section 4.1, the spike

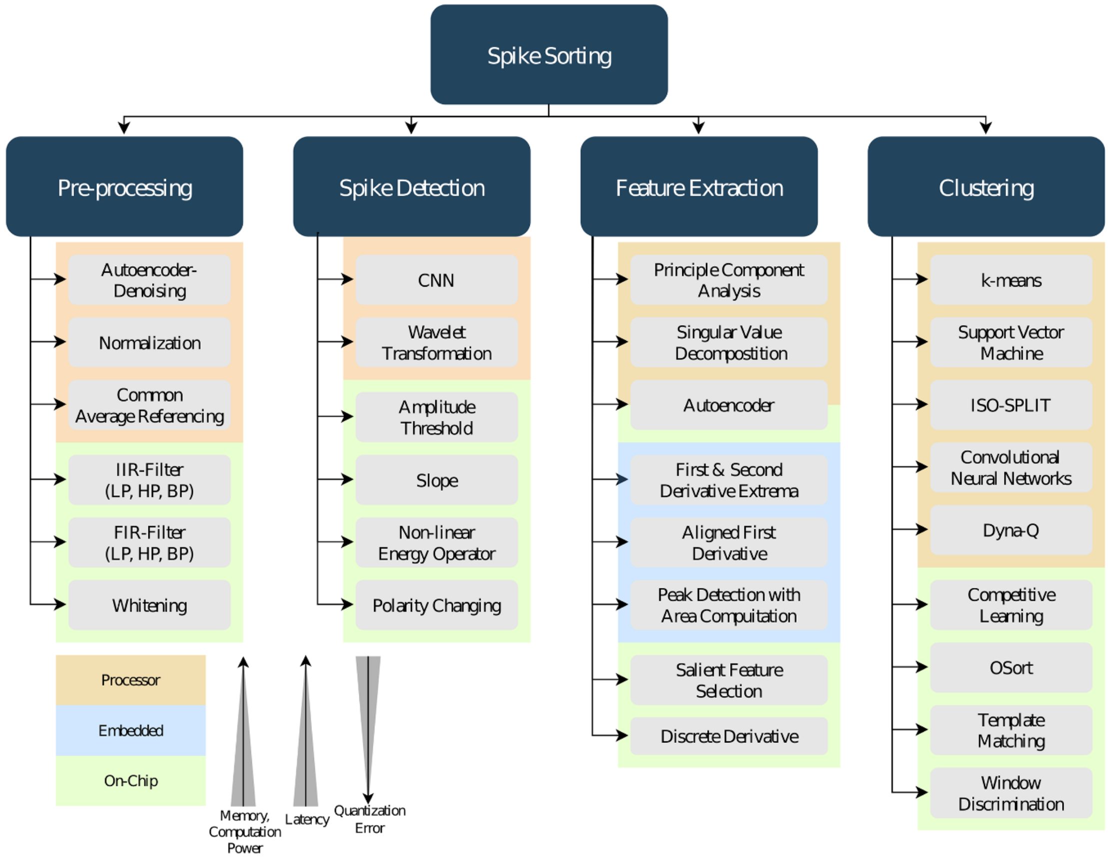

Figure 10. Overview of different modules of the Spike Sorting Pipeline which is used in the digital domain and is categorised into the suitable hardware platform (Processor, Embedded, On-Chip).

detection and frame generation in section 4.2, the feature extraction methods in section 4.3, and the clustering approaches in section 4.4. Also, an overview of different system architectures for spike sorters and their use cases are explained in section 4.5.

4.1. Pre-processing

In this section, we give an overview of the used pre-processing with a focus on filtering methods. They can be classified into three categories (i) frequency-specific, (ii) channel-specific, and (iii) channel-overreaching.

Frequency-specific filters are used to reduce noise and suppress artefacts and local field potentials. This is usually implemented with a band-pass filter (LFP: 0.1–100 Hz, Spike: 0.1–7 kHz). These filters can be implemented as an IIR filter and a FIR filter. While FIR can only be implemented digitally, IIR can be implemented digitally and analogue. The disadvantage of using FIR filters in neural applications is that the frequency selectivity increases with higher orders. But latency and resource consumption also increase due to the feed-forward structure. IIR filters are easy to implement due to the feedback structure and the latency is quite low in the range of the sampling period. For all of these reasons, the IIR implementation is recommended inside the analogue amplification stage and digital post-processing after quantization.

Channel specific filters are used e.g. for denoising neural input. This can be achieved by a deep learning method, called autoencoder [5, 46, 111]. The autoencoder is divided into an encoder and a decoder. In the encoder, the incoming data are reduced to minimal representation. In the decoder, the features are used to reconstruct the original data. The neural network is usually trained by using the difference, called loss, between the original data and the reconstructed data to find the best-fitting minimal representation. Noisy spikes from the same neuron share a very similar minimal representation and are therefore reconstructed close to each other. This approach shows in ECG applications an increased SNR of 20 dB [68, 108] and in neural applications with increased SNR up to 13 dB and minimal error compared to conventional methods [51, 96].

Channel overreaching filters use the spatial information provided by high-density probes for reducing background noise and artefacts. Common Average Reference (CAR) method and spatial whitening are often used digitally in spike sorting pipelines [64, 81]. Figure 11 shows the electrode configuration for different recording strategies. (a) shows the normal

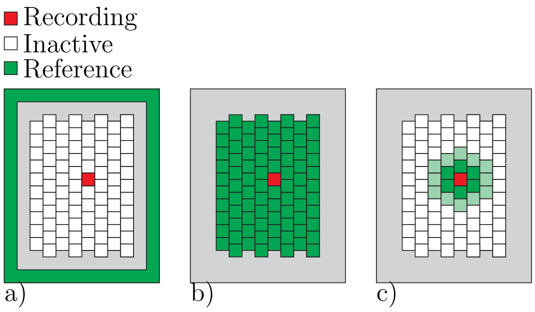

Figure 11. Methods of suppression artefacts in extracellular recordings with high-density electrode arrays: (a) normal configuration—(b) global common average reference—(c) local common average reference with distance-based weights.

configuration, in which neural input of the active electrode (red) is recorded against the global reference (green). This setup does not use spatial benefits and is therefore fragile to environmental influences. In (b), global CAR is used. CAR extracts the global average of each reference electrode and subtracts it from the recording input. The suppression factor here takes a maximum value of 1 if the interfering signal is present on all channels and the SNR increases by $\sqrt{N}$ with a number of reference electrodes considered. A)reduction of the suppression factor is reduced by channels that are defective (electrode, amplifier, ...) and thus do not allow the acquisition of signals. These channels can be detected by channel selection methods for extracting electrodes with non-neural activity [80] and neglected in the CAR and spike sorting pipeline. In (c), the local CAR algorithm is applied to allow for higher selectivity and further reductions in global and local artefacts like electrical stimulation. In addition, the Laplacian filter is used in EEG recordings for determining the reference from the neural input [127, 151]. The main difference between CAR and the Laplacian filter is that in Laplacian filter the input is weighted with the distance from the center electrode. Both methods achieves similar results.

Signal whitening is used to make signals of electrodes more independent from each other. This is useful for high-density spike sorters, that perform a single-channel spike sorting and merge the results afterwards. [18, 81] state that this drastically increased the performance of their algorithms. This is usually done by computing the covariance matrix on the electrode signals and then decorrelate the signals per channel by using matrix decomposition like the zero-phase component analysis (ZCA). ZCA whitening is often used because of its computational efficiency. On a data set X with n-channels and m-samples a covariance matrix C is calculated with (10).

$$C = m^{-1}\, X \cdot X^{T} \tag{10}$$

$$W_{\mathrm{ZCA}} = C^{-1/2} \tag{11}$$

$$Y = W_{\mathrm{ZCA}} \cdot X. \tag{12}$$

Afterwards, whitening matrix W_{ZCA} is determined via the inversion and squaring Λ with (11). Finally, W_{ZCA} is multiplied with its input X in order to the decorrelated matrix Y with (12). The computation of these channel-overreaching filters is done offline on a workstation and further research for hardware implementations is needed.

4.2. Spike detection and frame generation

After the pre-processing, the neural events inside the raw data of neural spike activity have to be detected. The used spike detection algorithm (SDA) extracts spike events and the following frame generator cuts a window/frame from the spike activity data stream at the time point of these events. These frames are passed to the spike sorter. In the following, the different methods for SDA are discussed.

In general, the SDA needs a threshold value for detecting spike events in the bitstream. The easiest method is to use amplitude thresholding (AT) on the neural raw data. Whenever the signal crosses the set threshold, a spike frame with a pre-defined window length is generated. Figure 12 (Left) shows an example of different SDA methods.

$$X_{\mathrm{th0}} = C \cdot \mathrm{median}\left(\frac{|x_{\mathrm{in}} - \overline{x_{\mathrm{in}}}|}{0.6745}\right) \tag{13}$$

$$X_{\mathrm{th1}}(z) = \frac{C}{\sqrt{N}} \sqrt{\sum_{x=1}^{x=N} \left(x_{\mathrm{in}}(z-x) - \overline{x_{\mathrm{in}}}\right)^{2}} \tag{14}$$

$$X_{\mathrm{th2}}(z) = 1.25\,\frac{C}{N} \cdot \sum_{x=1}^{N} |x_{\mathrm{in}}(z-x)| \tag{15}$$

$$X_{\mathrm{NEO}}(z) = x_{\mathrm{in}}(z)^{2} - x_{\mathrm{in}}(z-k)\; x_{\mathrm{in}}(z+k)\,. \tag{16}$$

For thresholding, different methods like the median absolute derivation (MAD) in (13), the window-determined root-mean-square (RMS) in (14) or the mean absolute (MA) in (15) are used [107]. In general, with all thresholding methods, the statistical deviation of the noise is determined with an additional scaling value C, which is determined via hyperparameter analysis on the used data set in order to reach a trade-off between missing spikes (false negative) and detecting noise/artefacts (false positive). This scaling value is usually in the absolute range of 4 and 6. In these calculations, the mean value calculation can be neglected, as the mean value of the bandpass-filtered raw data should be close to zero.

For the hardware implementation, MAD is not sufficient due to the high computational effort and is only used in offline processing. In hardware, noise

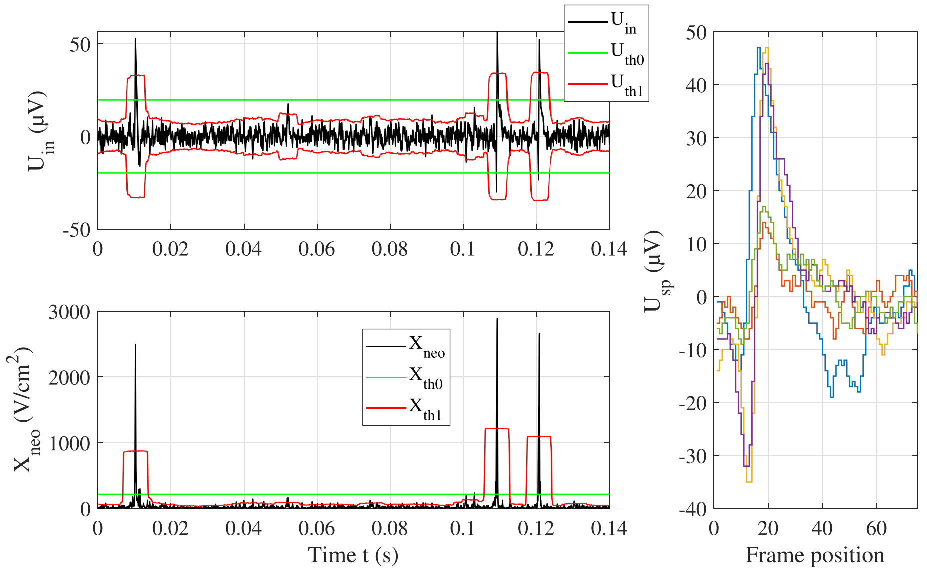

Figure 12. Performing and results from the spike detection algorithm (applied on dataset [93]) - Top: Determining the threshold directly on input signal—Bottom: Performing non-linear energy operator (NEO) on input signal with threshold calculation—Right: Detected spike frames (72 points).

distribution is determined more by window methods, like RMS or MA.

The disadvantage of using AT is that this method is prone to noise. Therefore, especially for input signals with low SNR, the nonlinear energy operator (NEO), also called Teaser Energy Operator, with (16) at $k = 1$ [72] is popular for processing neural inputs. This operator applies frequency-dependent amplification so that large signal changes are amplified and small changes like noise are damped. This gain effect comes from the window viewing method with multiplication, squaring and subtraction. The proposed SDA methods can be implemented in hardware easily [110, 130, 131].

The challenge in spike detection is, that the background activity is hard to detect due to a *SNR* below 0 dB. Here, some modifications of NEO-based SDA have been done in order to increase the sensitivity for detecting background activity. Therefore, methods like *k*NEO [72], MTEO [17], Wavelet-transformation-based spike detection [78], integer coefficient filter [21] and amplitude slope operator (ASO) [152] are available.

*k*NEO introduces the tuning parameter *k* in order to arrange the frequency-selective property of NEO in order to minimize the false-positive rate due to noise influences. MTEO is the superposition of several *k*NEO approaches, in which the maximum of all operators is used as output. The integer coefficient filter works like a short-window convolution for capturing spike-like windows.

$$\begin{aligned} X_{\mathrm{SDA}}(z) = 128\, x_{\mathrm{in}}(n) - 48\, x_{\mathrm{in}}(n-1) \\ - 156\, x_{\mathrm{in}}(n-2) - 36\, x_{\mathrm{in}}(n-3) \\ + 56\, x_{\mathrm{in}}(n-4) + 32\, x_{\mathrm{in}}(n-5)\,. \end{aligned} \quad (17)$$

Formula (17) shows the working principle without using any multiplier, but the parameters must be determined empirically with a hyperparameter optimization on the used data sets. This method achieves better accuracy results like NEO and AT with less computational effort [21].

The amplitude slope operator (ASO) [106, 152] reduces the computational effort by half compared to NEO by using only one multiplier, one subtraction and only two taps in the hardware. Formula (18) shows that a high amplification is achieved with a high slope and amplitude from the neural input x_{in}.

$$X_{\mathrm{ASO}}(z) = x_{\mathrm{in}}(z) \cdot [x_{\mathrm{in}}(z) - x_{\mathrm{in}}(z-k)] \quad (18)$$

$$X_{\mathrm{ADO}}(z) = \mathrm{abs}(x_{\mathrm{in}}(z) - x_{\mathrm{in}}(z-k))\,. \quad (19)$$

In addition, a higher accuracy has been shown in synthetic and real data [152]. The smoothing properties can be added by sweeping the tuning parameter *k*. An optimum is reported with $k = 4$ at a sampling rate of 30 kHz [106].

It is also reported, that the detection accuracy of the SDA can be increased by smoothing the SDA output with the Hamming or Bartlett window of length $4k + 1$ in order to suppress noise influences. The best result is achieved by using a tuning parameter *k* of 4 [107]. Also, the accuracy is sensitive against the firing rate of the input spikes, in which the accuracy decreases from 60% to 25% at a *SNR* of 0 dB when the firing rate increases from 10 Hz to 200 Hz [107]. This error appears from the used thresholding method during the runtime. With the normalization of the input as pre-processing or the noise estimation as post-processing, this effect can be minimised [107].

From hardware perspective, the logic consumption of NEO, MTEO and ASO are higher compared to the AT method, resulting from the necessary number of multiplication units and logic cells for calculating the SDA and the threshold value. This results in higher complexity on hardware and less number of SDA channels in a high-density recording unit for neural implants and it gets more critical if smoothing filters with an additional FIR filter is used. In order to achieve a trade-off between i) high accuracy, high robustness against noise and artefacts, ii) low logic utilization and low power consumption, the Absolute Difference Operator (ADO) is recommended [153]. Formula (19) shows that only the absolute different of two input values is used with a settable delay window k. It applies a high-pass filter on the neural input in order to remove LFP and other low-frequency artefacts. Its hardware implementation needs only 300 logic cells per unit with an detection accuracy of 96% in recordings with a runtime over 200 days [153, 154].

With the SDA trigger output of the available neural spike event, the corresponding spike frame is generated for further spike sorting. Figure 12(right) shows an example with a window size of 72 samples which are stored in a FIFO memory buffer. The window length of the spike frame depends on the ADC sampling rate f_s and the refraction time of the spikes τ_{sp} ($\approx 1,6\,\mathrm{ms}$). For example, the Data Acquisition System of the Utah Array from Blackrock Neurotech generates the spike frame within a window of 48 samples at a sampling rate of 30 kHz [128].

For some feature extraction methods, an alignment of spike frames is required in order to maximise cluster accuracy. The alignment of all spike frames is done at the window position at a delay time of 300 μs with the maximum peak, minimum peak, maximum absolute peak or maximum slope [29]. This delay requires a time delay filter between the SDA input and the frame generator input in order to extract neural information before the active SDA trigger output.

Current research also shows interest in deep learning approaches for spike detection by combining convolutional neural networks (CNN) and recurrent neural networks (RNN) with long short-term memory (LSTM) cells. These large CNN+LSTM networks are quite complex to implement on FPGA and are more often designed for offline processing on workstations. Also, the authors of [128, 129] used CNN architectures. One architecture is designed for the discard of unstable channels. Another one is built for a background activity rejection. Both are server-grade solutions.

To sum up, the accuracy of spike detection depends strongly on the threshold method in order to achieve high accuracy. In determining the scaling value C for hardware execution, a trade-off must be found between the detection accuracy, the noise sensitivity, the energy consumption of the logic and the computational effort of the whole spike sorter pipeline must be performed. Also, this value should be updated automatically during runtime.

4.3. Feature extraction

On the captured spike frames, the feature extraction (FE) for performing clustering in order to determine the spike trains for neural decoding has to be done. This step is necessary due to the scaling of the computational complexity of most clustering algorithms exponentially with the number of features. To reduce this overhead, feature extraction algorithms can be used. They compute a set of features that represent the spike frame without losing critical information. Clustering can then be performed on these features instead of the full spike frame signal. FE algorithms can be divided into:

i) matrix decomposition,
ii) geometric,
iii) important samples and
iv) deep learning based FE.

In the following, we describe these in more detail, including information on how they are implemented, which limitations they have, and what promising ideas are not covered by current research.

Matrix decomposition based FE uses the idea to represent the input spike frames as a matrix and then decompose this matrix to find a smaller representation of it. Two approaches that use matrix decomposition are principal component analysis (PCA) [43, 88] and singular value decomposition (SVD) [41]. PCA is used for spike sorting, e.g. by [18, 128]). It first computes a covariance matrix C out of m input spike frames X with zero mean with (20). Then, a matrix decomposition on the covariance matrix computes its eigenvalues Λ and eigenvectors V with (21). The highest ones can be used to select and weight the most characteristic features (or principal components) P with (22). In contrast, SVD represents the input spike frames as a matrix and decomposes this matrix directly into singular values [81] with a unitary matrix U and transposed eigenvectors V with (23). The principal components can also be computed when first computing SVD to calculate Σ. With (24) Λ can be computed. This is useful since computations on the covariance matrix can be ineffective for a high number of spike frames.

$$C = X^T X\,(m-1)^{-1} \tag{20}$$

$$C = V \Lambda V^T \tag{21}$$

$$P = X \cdot \Lambda \tag{22}$$

$$X = U \Sigma V^T \tag{23}$$

$$\Lambda = \Sigma^2\,(m-1)^{-1}. \tag{24}$$

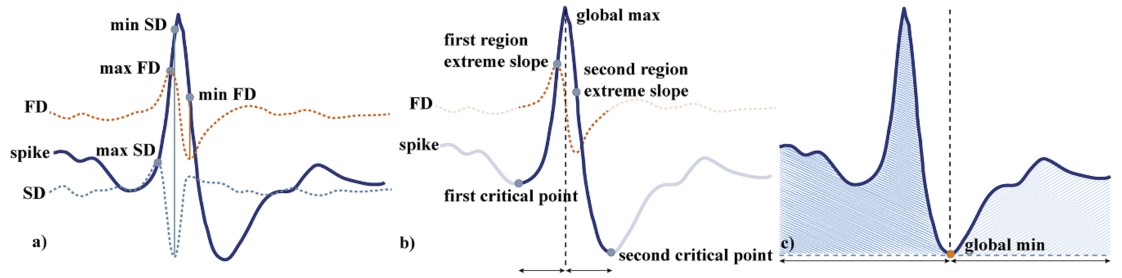

Figure 13. Visualization of performing a geometric feature extraction by using (a) aligned first derivative (AFD), (b) first and second derivative extreme (FSDE) and (c) peak detection with area computation (PDAC).

Both algorithms usually work offline on a workstation, since they need all spike frames as input. Online algorithms, like incremental PCA, are possible but have not been used widely for spike sorting, yet. Similarly, while these algorithms are computationally complex, hardware accelerators exist [8] and could be used for spike sorting. For on-implant, the computational power might be too high for hundreds of channels. Further research investigating the reduction of computational power is needed.

Geometric FE algorithms are using geometric calculable features. They can include the extreme point of the spike and its derivatives and areas under the curves. Aligned first derivative (AFD) [70] and first and second derivative extreme (FSDE) [84] are two examples. Usually, the FSDE algorithm should give the same features as AFD plus the features of the second derivative. Peak detection with area computation (PDAC) [13] calculates the areas under the curve and scales them with the difference between the minimum and maximum extreme. Figure 13 shows an example of the three mentioned methods. Alternatively, the authors propose to take the area above the curve divided by the maximum and scale them as well. All these geometric-based algorithms are unsupervised and are computed online, spike by spike. In addition, they are computationally inexpensive. Hence, it is convenient to implement them on FPGAs or ASICs, which makes them feasible for on-implant implementations. To our knowledge, no analogue implementation of one of those FE exists, but because of the low computational complexity. This could be feasible in combination with analogue spike detection heavily reducing the load of the ADCs.

Sample selection is another FE class. The idea is to reduce the features by only passing important samples to the clustering. To do so, an algorithm has to select the samples that distinguish the clusters most. Therefore the clusters need to be known in advance. This can be computationally expensive, but once selected the computational cost are almost nonexistent. To our knowledge, the only approach of this class is the salient features selection (SFS) proposed in [116]. They have a training and inference phase. In the training phase, a shadow spike sorter computes the incoming spikes and creates labels for a set of spikes. A mean waveform of each cluster is computed. For each cluster, an optimizer selects the samples of the mean waveforms that distinguish the cluster most. These samples are the configuration, that can be used for inference. The training can be computed on different hardware than the inference. This allows a FE inference to be implemented on-implant. The inference is not adapting to changes automatically. Those could be updated by the shadow spike sorter. Therefore an online spike sorter could compute the mean waveforms on a sparse number of frames per channel. A salient feature optimizer could then recalculate the salient samples and update the inference unit. The computation power of the inference unit scales with the number of channels, while the shadow spike sorter only needs a multiplexer to swap between the channels.

Deep learning FE are the most recently used class. The most used architecture uses autoencoder [5, 46, 111]. Autoencoders showed success as FE in different application cases [77, 95]. They are usually self-supervised and benefit from supervised training mechanisms. The encoder minimizes the number of features which are usually directly used for the clustering. The neural network is usually trained by using the difference, called loss, between the original data and the reconstructed data to find the best-fitting minimal representation. However, this will just lead to a feature reduction which allows a good reconstruction and does not take into account the cluster separability with the generated features. Seong *et al* [114] targets this issue and proposes a modification of the loss function that takes clustering accuracy into account. This way the authors improve the features for better clustering accuracy. Radmanesh *et al* [96] proposes a modification of the input layer which punishes noise sensitivity. While the inference is already implemented on FPGAs and ASICs, the training is usually done on workstations and needs further optimizations to be implemented on FPGAs or ASICs.

4.4. Clustering

Clustering is used to distinguish between the different neurons due to the neural response. The goal of the clustering algorithm is to assign all detected spike waveforms from the same neuron to its own cluster. The clustering algorithms have to deal with electrode drift, which causes a slight change in the features of the clusters over time. In addition, they have to delete clusters, when neurons die and create new ones when a new neuron moves into the measuring range of the electrode. This allows classification into three classes, (i) Cluster initialization given, (ii) Number of clusters given and (iii) Adaptation through runtime. In addition, we distinguish between analogue and digital implementations, and offline and online processing.

(i) Cluster initialization given: These are online clustering algorithms and can be configured for single- or multichannel activity. They can be divided into two sub-classes; Optimized for workstations, and optimized for on-implant. Both require a training phase, which is usually implemented offline. After the training, the configuration is given to the clustering algorithm for online inference.

A deep learning approach for clustering uses CNNs. [57, 94] present an approach for online spike sorting for multi-channel activity. The CNN is a classifier with a fixed number of clusters and needs no previous feature extraction. However, a feature extraction could be beneficial for the reduction of computational complexity. For the training, ground truth is required. Once trained the network can be used for online inference. The CNN is usually executed on a workstation. Like the other CNNs, they could also be implemented for on-implant. The adaptation during runtime would be possible if another system is training on a sparse subset and updates the weights of the inference system.

The next two algorithms, Template matching (TM) [81, 130] and window discrimination (WD) [29, 116] could be executed on-implant. TM is a classification algorithm used for online inference in the domain of spike sorting. The algorithm calculates the distance of the incoming spikes' features to all feature sets of each template. Either the algorithm matches the incoming spike to the closest template or additionally checks if the distance is below a threshold. If not the spike is discarded. The algorithm is computationally not intensive and can be implemented on-implant. The configuration of the templates can be done by another system. WD sets an upper and lower limit, called window, for each feature for each cluster. If each feature matches the window of a cluster, it is assigned to it otherwise the spike is discarded. This approach is even more efficient than template matching but also requires another system to generate the configuration for each window of each feature. [116] proposed this approach for on-implant spike sorting.

(ii) Number of clusters given: *k*-means [62] is the most common offline clustering algorithm. It is a partitional clustering algorithm executed on a remote processor with a fixed number of clusters. Except for the number of clusters, the algorithm is performed without any hyperparameter. Usually, the clustering is executed on previously extracted features. For each cluster, the mean is set to a random data point. Next, each data point is assigned to the cluster with the least increased variance. For calculating the variance the Euclidean distance is used. After the assignment, the cluster mean is recalculated. The assignment and cluster mean update is repeated till convergence. The algorithm is not robust due to bad cluster initialization. Since it is a very common clustering algorithm, various adaptations of *k*-means exist and one, *k*-means++ [3], improves the cluster initialization leading to a much more robust cluster algorithm.

Another well-known offline algorithm that is executed on a remote processor is the support vector machine (SVM). Typically it is used to distinguish between two classes and has also no hyperparameters. Therefore for each extracted feature, a variable is created, which defines a hyperplane. The hyperplane orients itself in between the clusters to divide and maximizes the area around it where no data point is. This way the clusters are parted away from each other the most. For each additional cluster, another hyperplane is needed to distinguish between the clusters. This approach is used by [96]. Both k-means and SVM can be used for single-channel activity.

Competitive learning (CL) can be used for the single-channel activity. The algorithm trains itself during inference. The cluster centroids are initialized with a random selection of the first incoming spikes. After initialization, incoming spikes are assigned to the nearest cluster. The cluster centroids are updated with a learning rate l and the difference between the assigned clusters centroid and the spikes features. This approach can adapt to compensate for the electrode drift [14]. One problem is that it can not adapt to the number of clusters during runtime. The algorithm can be executed on-implant.

(iii) Adaptation through runtime: The here presented algorithms adapt through the number of clusters through runtime, also called unsupervised clustering.

One solution to make clustering algorithms adapt the number of clusters after its runtime is the cluster accept or merge (CAOM) algorithm by [128]. This can be done offline after the feature extraction of all spikes and is usually executed digitally on a workstation. ur Rehman *et al* [128] use it with k-means and set a maximum number of clusters for k-means. Afterwards, they compute a similarity feature and depending on a threshold either merge or accept

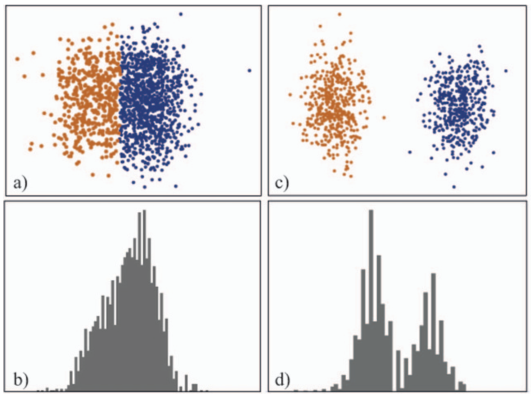

Figure 14. Visualization of ISO-Split: (a), (c) Feature Space of two clusters with different cluster center distances - (b), (d) Histogram of the 1D-projection.

a cluster. The proposed ISO-SPLIT [66] density-based clustering algorithm is executed on a remote processor. The algorithm was proposed in an offline version but recently an online-capable ISO-SPLIT version was proposed by [56]. The numbers of the cluster are adaptable. Each data point is a new cluster. Then it compares each cluster to each other. Figure 14 shows in (a) and (c) two clusters. Next, they are mapped on a 1D-projection of the two clusters, (b) and (d) shows the histogram of the 1D-projection. A statistical uni-modality test is performed on the 1D projection. If accepted, the clusters are merged, which is shown in figure 14(a). Otherwise, the data points of the two clusters are reassigned at the optimal split, like in figure 14(b). Since they are already split at the optimal point, the clusters are not rearranged. The comparison of the clusters is performed till convergence is reached. This algorithm is also implemented online where only changing clusters are compared to each other. An adaptation could also be to start with the closest cluster for comparison reducing the computational complexity. However, the algorithm's weakness is, that it stores each spike's features. Thus, it is only executed on a workstation. In general density-based clustering algorithms are not widely used in spike sorting. We would expect better performances from density-based than distance-based clustering algorithms because most feature extractors deliver non-circular features. Therefore, we recommend further research with density-based clustering algorithms.

Another clustering approach uses reinforcement learning. It is an online approach executed on a remote processor automatically adapting the number of clusters. The algorithm, called Dyna-Q [123], is based on the Q-learning algorithm. Reinforcement learning tries to find the optimal action for a certain task. First, an agent explores the almost random actions and the environment gives feedback, also called reward, to the actions. Over time the agent learns which feedback to expect for action and can now decide which actions to take for an optimal outcome. Dyna-Q learning adds a model phase after the action which stores the next state and the reward pairs to learn about the environment and train the agent on the self-created environment in a planning phase. Moghaddasi *et al* [70] use Dyna-Q reinforcement learning for clustering the features. The authors compute the reward for new clusters by the number of existing clusters and a number of features and scale that with a punishment coefficient. The reward for adding the spike to a cluster is computed on feature sets of 20 random spikes. They state that the stability of the algorithm is very sensitive to the punishment coefficient. We experienced the same. Thus, we recommend an investigation into a better approach for the reward for new cluster action. The algorithm is usually executed on a workstation. In our opinion, an implementation on-implant could possible with some adaptations. Therefore, we recommend using only a buffer of the last 20 spikes to compute the reward. Due to its computational intensity, we can not recommend the Dyna-Q algorithm.

The distance-based clustering algorithm, called OSort [105], can be executed online on-implant. The algorithm has two phases. In the assignment phase, incoming spike waveforms are matched to the nearest cluster centroid. If the distance is below the assignment threshold the spike is matched. If the closest cluster is far away a new cluster is created. In the cluster update phase, the clusters' centroid is updated. After the update, the distance between the new centroid and the other cluster's centroids is computed. If the new centroid is to close the clusters are merged. Usually, OSort is used without a feature extraction because of its lightweight implementation. If the clusters centroids would not be updated one could see it as a template for a spike cluster. Also, multiple adaptations are made to improve the algorithm's efficiency. OSort is used by [110, 131].

Concluding this, there are to our knowledge no online spike sorters adapting the number of clusters automatically and work without hyper-parameters, which is essential for on-implant spike sorting. The closest on-implant spike sorting approach that is also feasible for multi-channel approaches is OSort.

4.5. System architecture for spike sorting

The presented and explained modules from the last sections can be connected to one pipeline. In this section, we show some general approaches for the system architecture of state-of-the-art spike sorters in hardware and present a subset of concrete pipelines.

We can distinguish between different dimensions. One is the decision if the adaptation of the spike sorter takes place on the inference system or another computing platform. If it takes place on the inference system the algorithms are usually more complex. If not the algorithms of the inference systems need to be configurable. The adaptation can be performed on a sparse subset of the data. The second dimension is the placement of different phases. The inference could be split up into different computing platforms to reduce the data rate. Thus, the system can be split at any step after the spike detection.

In addition, we can distinguish between two application-dependent problems; (a) wide area coverage with sparse observation (e.g. Utah array) and (b) small area coverage with detailed observation (e.g. NeuroPixels). Depending on the neuron density of the implant brain area different system architectures are useful.

a) If the brain area has a low neuron density and typically not more than one neuron is expected, a system using only a spike detection without a feature extraction or sorter can be accurate enough. If a low number of neurons (one with maybe occasionally two or three) is expected, it might be useful to implement a simple spike sorter. This heavily depends on the application. However, for brain areas with more than one expected neuron, a spike sorter is necessary.
b) For small area coverage with detailed observation different system architectures can make sense. Not every channel of nowadays sensors-devices can be digitized in parallel. Therefore a selection of the most important channels is needed. For brain areas with low neuron density, a spike sorting should be performed for a neighbourhood. Then there are two options. Either the channels can be selected to which a neuron is closest so that one neuron per channel is mapped. Then a spike detection for those channels might be enough. However, this implies two problems with their own solutions. One is that spike detection might still be necessary because another neuron could be also very close and therefore its action potential could be high enough to trigger both channels' spike detections. Thus, these channels should be merged afterwards. The other disadvantage might be that the number of neurons exceeds the number of parallel digitized channels. Both mentioned problems can also be tackled with another system architecture. For those cases where neurons are relatively close together, an electrode can be selected where all spikes are easily detected. Then spike sorting can be performed for those electrodes. The increased overhead for a spike sorter might not be worth it, especially due to the overhead of signal routing during the runtime. Therefore, we suggest for future work to take a look at mixed system architectures.

With the different dimensions, we have seen four different system architectures in existing hardware spike sorters.

- **Type I:** spike-detection-based systems with pre-processing and spike detection
- **Type II:** feature-extraction-based systems with pre-processing, spike detection and feature extraction
- **Type III:** spike-sorting-based systems with pre-processing, spike detection, feature extraction and clustering
- **Type IV:** inference spike-sorting with pre-processing, spike detection, feature extraction and clustering and a training system to update the configuration

Table 1 shows an overview of different spike sorter pipelines which are already implemented in hardware.

4.6. Neuromorphic approaches

A new approach is using neuromorphic techniques as an on-implant neural processor for spike sorting and neural decoding. These structures try to emulate the biological neural network with neurons and synaptic input on technical systems. The benefits of these structures are that an event-based and in-memory computation is available which results in high energy efficiency, high data throughput, minimal data conversion, low memory requirements and low latency for several channel [117, 150]. The idea of performing spike sorting and neural decoding in neuromorphic structures should lead to high similarities between biological neural networks and artificial networks, which can significantly reduce computational effort and allow high plasticity. The artificial neurons are mostly working on the integrate-and-fire method and the classification is done on the winner-takes-it-all principle.

For achieving a long-term stable spike sorting, the actual learning rules are very sensitive to initial conditions, and quite often unstable without meta-plasticity [143]. Therefore, more research on new system architectures including pre-processing, algorithms for pattern recognition, online training phases and long-term stable and implantable memory devices, like memristor or resistive RAMs, are necessary. In the following, the concept of using spiking neural networks (SNN) and analogue computation via memristor are shortly discussed.

In memristor-based analogue computation, the neural input is applied on memristive crossbar arrays. Such memristors are non-volatile devices which shows resistive behaviour and it saves applied information until a reset voltage is achieved. In spike sorting, such devices are used in order to determine the resistive change when a spike waveform is applied and the changes can be clustered afterwards [35, 36, 117].

SNN uses the firing rate of each layer as a specific characteristic to the corresponding classification output using the spike-timing-dependent plasticity in the training phase [137]. In addition, SNN shows high

Table 1. Overview of different spike sorting systems running on hardware platforms—Comparing the used methods of spike detection algorithms (SDA), feature extraction (FE) and clustering with different properties and common metrics.

	Properties			Methods				Metrics				
Year	Type	Platform	Processing	SDA	FE	Clustering	Training	Acc. %	A_{tot}/ch. mm^2	P_{dis}/ch. μW	$P_{\text{dis}}/A_{\text{tot}}$ μW/mm^2	f_{clk} MHz
2009 [10]	II	ASIC	Digital	NEO	Min-Max	N/A	N/A	N/A	1.76	114	64.8	40
2011 [29]	III	ASIC	Digital	NEO	DD	kMeans	Offline	92	0.06	2.03	30	1.6
2013 [49]	III	ASIC	Digital	AT	PCA	kMeans	Offline	75	0.07	4.68	66.86	0.48
2013 [44]	III	FPGA	Digital	NEO	Hebbian	Fuzzy cMeans	Offline	96	N/M	N/M	N/M	N/M
2014 [65]	III	ASIC	Digital	NEO	DWT-PCA	kMeans	Offline	N/M	0.06	0.68	9.7	20
2015 [19]	III	ASIC	SNN	Pulse-coded	RNN		Online	N/M	N/M	N/M	N/M	N/M
2016 [47]	III	ASIC	Digital	AT	Min-Max	Bayesian	Offline	95.3	0.12	1.74	14.5	0.03
2016 [59]	III	FPGA	Digital	AT	N/A	OSort	Online	93	0.077	10.3	133.8	56
2016 [137]	III	Board	SNN	N/M	Memristive change		Online	86.6	N/M	8.1	N/M	N/M
2017 [141]	III	ASIC	Digital	NEO	Haar DWT	Decision Tree	Offline	76	0.023	0.75	32.61	0.16
2017 [14]	III	ASIC	Digital	NEO	PDAC	CL	Online	95.4	0.027	18.18	683.43	1
2017 [58]	III	FPGA	Digital	AT	Event polarity	Min-Max	Offline	96.4	N/M	N/M	N/M	N/M
2018 [147]	III	ASIC	Digital	*k*NEO	DD	C-Sort	Online	84.5	2.7	148	54.81	0.96
2019 [21]	III	ASIC	Digital	Integer	DD	kMeans	Online	86	0.03	0.175	58.33	0.025
2019 [131]	III	ASIC	Digital	NEO	N/A	OSort	Online	87	2.57	2.78	1.085	0.024
2019 [130]	III	ASIC	Digital	NEO	TM		Offline	90	0.03	0.064	2.13	0.024
2019 [35]	III	Board	Memristor	N/M	Resistive change		Offline	N/M	N/M	N/M	N/M	N/M
2019 [73]	I	ASIC	Digital	Dual-AT	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2020 [116]	IV	ASIC	Digital	N/M	Salient	WD	Online	88	6.55	905.9	138.24	N/M
2020 [148]	III	ASIC	Digital	N/M	Dictionary	Subspace	N/M	>92	0.09	10.48	116.44	N/M
2021 [132]	III	ASIC	Digital	NEO	Binarized NN		Offline	91	0.33	2.02	6.21	0.024
2021 [38]	III	ASIC	Analogue	AT	FSDE	kMeans	N/M	93.2	1.02	4.35	4.25	0
2021 [114]	III	ASIC	Digital	AT	CNN AE	kMeans	Offline	95.54	0.75	168.56	224.75	7.68
2022 [117]	III	Board	Memristor	N/M	Memristive TM		N/M	94.62	0.0008	2.15	2687.5	N/M
2022 [121]	III	ASIC	Digital	AT	SS	kMeans++	Online	92	0.023	0.33	143.48	0.025
2022 [149]	III	ASIC	Digital	NEO	AFD	CL	N/M	94.12	0.014	2.79	199.28	125

N/A: Not available, N/M: Not mentioned, ch: channel, Acc.: Accuracy,
A_{tot}: Total area for the pipeline for each channel, P_{dis} : Power dissipation of the pipeline, f_{clk}: Clock frequency.

stability against quantization errors where the accuracy loss is under 0,28% at a reduction from 32bit-float to 4bit-integer [120].

A first approach for SNN-based spike sorting in hardware has been done with [19] with an accuracy of 96% on synthetic datasets. It contains 256k neurons with 64k short-term (analogue, programmable) and 64k long-term plasticity (binary, trainable) arrays.

A new framework of SNN-based spike sorting is published with NeuSort [145], which achieves an accuracy of 78.69% with real datasets. It consists of a two-layer structure, in which the encoding layer (first layer) converts the neural input into spike sequences in order to detect if a spike is available and to extract specific features with the receptive field encoder. The perception layer (second layer) maps the spike sequences to the biological cluster with the integrate-and-fire approach. This system is suitable for online training using the Hebbian learning rules with the winner-take-all mechanism in order to generate the templates. With this learning technique, it is possible to detect changes in the spike waveform in order to update the corresponding cluster.

In order to improve the metrics of future spike sorters, SNN and memristors can be combined in order to reduce the circuit complexity for achieving high-density approaches. A first trial in [137] have tested it on a prototype which achieves an accuracy of 86.6% with a power consumption of 8.1 nW and a latency below 1 μs. For using the neuromorphic approaches in future implants, an advanced event-based processing of the neural input is necessary. Figure 15 shows an example of the used sampling technique applied to the neural input with (a) the classical Riemann method and with (b) the event-based Lebesque method. In the Lebesque sampling technique, a trigger signal is generated if the amplitude exceeds or falls below a delta [4, 103]. In the following, some approaches for memristor-based and SNN-based pre-processing are presented.

For memristor-based pre-processing, an analogue spike detection like an amplitude-window discriminator with dual thresholds [58] or nonlinear energy operator (NEO) [53, 54, 142] can be used in order to control the memristor for further processing. Analogue NEO has a high circuit complexity by using a high-pass filter, a squarer/multiplier and a subtractor. An improved method with enhanced energy derivation has reduced the circuit complexity and achieves higher robustness against artefacts due to the LFP [24]. It only needs a second-order high-pass

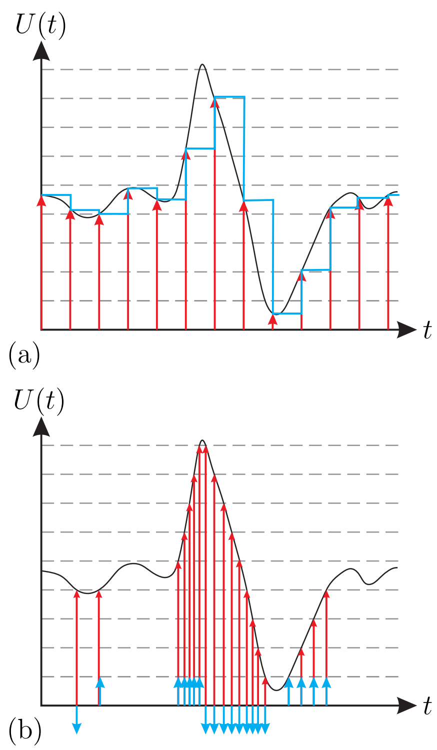

Figure 15. Example of sampling techniques: (a) Rieman (classical with ADCs) - (b) Lebesque (event-based).

filter and a squarer. For thresholding, (i) a constant voltage [24, 54, 58], (ii) lossy peak detection [53], (iii) low-pass filtered input [23, 54] or (iv) automatic noise estimation [142] can be used, but they have only be tested on simulated data sets and until now not on neural inputs directly. For the further spike sorting process, a time delay of 500 μs via a high-order all-pass filter [38] is required in order to capture the spike information prior to the active trigger output. But it needs a large chip area.

For SNN-based pre-processing, analogue spike detection can also be used to identify the time point of a spike, but the classification needs pattern recognition to the corresponding neuron type. For this, a pulse-density bitstream with polarity detection, like the example in figure 15(b), is used for the SNN via a one-bit delta-modulator ADC [19, 37], a threshold adjustable 1-bit comparator [58] or a frequency-modulation with voltage controlled oscillators and spike-encoding [87]. The output of this bitstream has a ternary weighting in order to distinguish between the states (i) no activity, (ii) positive or (iii) negative event.

This technique allows a reconstruction of the input signal by counting the ternary bitstream over time in order to apply classical machine learning techniques for benchmarking.

$$E_{\mathrm{evt}} = \int_{t_0}^{t_0+\Delta T} P(t)\ \mathrm{d}t \qquad (25)$$

For benchmarking different kind of system architectures, we propose to use the metric of the energy consumption for processing one event with (25). This metric allows a comparison between neuromorphic and digital strategies which considers the power consumption of the architecture $P(t)$ and the computational duration ΔT.

5. Neural decoder

Neural decoding involves turning the complex activity patterns in our brains into understandable signals, essentially interpreting what's going on inside the biological neural network. What we can learn from brain activity varies greatly based on the specific brain area being observed. This could range from understanding sensory processing, thought and memory, to the brain's involvement in planning and carrying out movements. In all these scenarios, the primary aim is to capture the essential information embedded within these neural patterns [146].

In the realm of BCIs that utilize advanced MEAs that penetrate the brain, these interpreted neural signals are put to work. The neural decoder is used to predict motor behavior, perceptions, or cognitive states and the interpreted data can be used in various ways, for example, to control a prosthetic device in real-time or enable direct brain-to-text communication. The extent of optimization of the neural signal processing pipeline varies depending on the intended use. For instance, fine-tuning the decoding process in motor decoding BCIscan significantly enhance the control over a prosthetic hand, whereas focusing too much on details of spike-sorting might not markedly improve accuracy [27].

This chapter is divided into a short overview of (i) different types of neural inputs and (ii) decoding techniques, including data pre-processing.

5.1. Types of neural signals

To fully understand how various decoder architectures work, it is crucial to understand what technologies are used to record neural signals. This recording can be done in humans, as well as in non-human primates and other research animals, using a range of techniques. These include invasive methods, like microelectrode arrays (MEAs) or electrocorticography (ECoG) and non-invasive methods, such as electroencephalography (EEG), functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG) and functional near-infrared spectroscopy (fNIRS).

The degree of invasiveness, along with the temporal and spatial resolution, varies significantly across these recording modalities. Invasive techniques provide more immediate access to specific brain regions, thus offering superior temporal and spatial resolution. For instance, Neuropixels probes can track the activity of hundreds of individual neurons with the precision of detecting single neuron activity. In contrast, EEG measures the collective electrical fields produced by brain activity over broader areas, yielding much lower signal strength and resolution [9]. fMRI, while capable of monitoring neural activity across the entire brain, relies on the slower process of blood oxygenation to deduce neural activity, with a spatial resolution of approximately one cubic millimeter. These differences highlight the inherent trade-offs between achieving high spatial and temporal resolution versus the level of invasiveness. Electrophysiological methods, such as MEAs, are particularly noteworthy. Their high resolution and the possibility for integration into fully implanted closed-loop systems make them especially promising for applications designed for patients that rely on assistive technology daily.

This paper focuses on neural signal processing using spikes recorded by MEA. Methods for decoding spike trains—sequences of neural action potentials—to identify movement intentions are discussed in the next section.

5.2. Overview of methods for decoding movement intentions

Neural decoders for neuroprostheses control have been in development for about 40 years. Research in the field of decoding movement intentions is currently very active. Therefore, we will provide a detailed overview only of the most current trends here.

Early approaches to decoding neural signals, such as the population vector method [28], relied heavily on intricate feature engineering and lacked robustness. However, the emergence of advanced machine learning technologies, especially deep learning, has marked a significant paradigm shift within the field of neuroscience. Deep learning has proven to be a potent tool, enhancing both the accuracy and adaptability of neural decoding for various tasks [61]. This advancement is supported by numerous studies which have shown that modern machine learning techniques can surpass traditional decoding strategies [31], including Kalman filters and logistic regression, in performance. The subsequent section will showcase a selection of studies focused on the neural decoding of movement intentions (see table 2), utilizing supervised machine learning methodologies. While not comprehensive, this overview aims to highlight significant and recent contributions to the domain, illustrating the transformative impact of machine learning on enhancing our understanding and interpretation of neural signals. Understanding the connection between neural activity and its corresponding state, such as the position of a hand, necessitates a statistical model capable of accurately identifying and filtering out noise and artifacts from the training data. Over time, this model learns the intricate relationship between the patterns of neural activity and the associated behavioral task, effectively mapping brain signals to specific physical actions or states.

Linear models have long been a staple for mapping neural signals to motor activity. Techniques like Linear Filtering, utilizing linear equations, have shown to be effective for real-time control of 2D cursor movement based on neural activity [83, 115, 136]. However, this approach operates on the premise of a linear relationship between the firing rates of neurons and motor actions. The level of confidence or uncertainty of the models is also not taken into account, which could limit their applicability for predicting complex, temporal motor patterns. To overcome these limitations, the Kalman Filter was introduced. This technique is adept at accounting for noise within the data and operates over much shorter intervals (approximately 70 ms), making it more suited for real-time applications. The Kalman Filter employs a recursive algorithm to update estimates of the target state continually, enhancing the prediction performances.

When controlling a cursor using neural activity in a position-based closed-loop system, the movements tend to be longer and more curved compared to the direct, straight-line paths typically executed by people without individuals. Additionally, users often find it challenging to bring the cursor to a halt precisely at a target or to keep it steady in a fixed location. To address these challenges, researchers have explored the use of velocity-based closed-loop control systems incorporating a Kalman Filter, specifically designed for velocity (referred to as Velocity Kalman Filter or VKF), to achieve more accurate and stable control of the cursor through neural signals [55]. This approach was tested in a study involving two tetraplegic patients, using a task that requires moving a cursor to a target and then returning it to the center, to evaluate the effectiveness of neural-based cursor control. While employing velocity-based systems has led to improvements in controlling the cursor through neural activity, the overall performance still lags significantly behind that of natural arm movements, posing a substantial barrier to their practical application in clinical settings. One of the shortcomings of the VKF is its slower trajectory completion times and reduced accuracy in comparison to natural arm movements. It tends to require more time for target acquisition, and the paths it creates are longer. To overcome the limitations of the Velocity Kalman Filter (VKF) in neural-based cursor control, Gilja *et al* [30]. introduced the recalibrated feedback intention-trained Kalman filter (ReFIT-Kalman Filter). This approach enhances VKF by implementing a two-stage optimization to better match the neural prosthesis with intended velocity estimates and by incorporating both cursor position and velocity into the decoding process, thereby improving control accuracy. Experiments with two monkeys

Table 2. Overview of different implementations of neural decoding for the movement intention using spikes.

Year	Decoding Objective	Primate	Architecture	Methods Comparing
2003 [140]	Cursor Control	NHP	KF	Linear Filter
2008 [55]	Cursor Control	Human	VKF	KF
2012 [30]	Cursor Control	NHP	ReFIT-KF	VKF
2012 [122]	Cursor Control	NHP	ESN (RNN variant)	VKF
2018 [135]	Hind limb Kinematics	NHP	LSTM	WF, PLDS+WF, XGBoost, RNN
2018 [67]	Reach Kinematics	NHP	rEFH (RBM variant)	WF, KF, UKF
2018 [26]	Wrist EMG	NHP	GAN(ADAN)-LSTM	LSTM, CCA-LSTM, KLDM-LSTM
2019 [76]	Wrist EMG	NHP	LSTM	WF, WC
2019 [85]	Reach Kinematics	NHP	sd-LSTM	VKF, Velocity-LSTM
2019 [126]	Reach and Hind limb Movement	NHP	Multilayer LSTM	WF, KF, UKF, LSTM
2020 [31]	Reach Kinematics	NHP	LSTM	WF, WC, KF, NB, SVR, XGB, FNN, RNN, GRU, Ensemble
2020 [75]	Hand Kinematics, Cursor Control	NHP	SBP+ReFIT-KF	KF, SVM
2021 [2]	Reach Kinematics	NHP	Quasi-RNN	WF, WC, KF, UKF, SRNN, GRU, LSTM
2022 [139]	Hand Kinematics	NHP	ReFIT-NN	ReFIT-KF

showed that ReFIT-KF significantly outperforms VKF in cursor control tasks. Cursor movements with ReFIT-KF were straighter, more similar to natural arm movements, and completed more quickly. The efficiency in target acquisition with ReFIT-KF was markedly higher, achieving 75%–85% of the performance of natural arm control and doubling the efficiency of VKF.

The successful application of ReFIT-KF in controlling neural prostheses has been adapted in other modern decoding settings. Specifically, the integration of spiking-band power (SBP), a neural feature for motor prediction defined as an average of absolute 300–1000 Hz band-pass-filtered signal, with ReFIT-KF has been employed to decode the one-dimensional movements of individual fingers (index, middle, ring, and pinky). In research conducted by Nason *et al* [75], this combination of SBP and ReFIT-KF was evaluated against support vector machines (SVM) in classifying two-dimensional finger movements without restrictions. Furthermore, SBP combined with ReFIT-KF was also benchmarked against the standard Kalman Filter in predicting two-dimensional arm movements in a center-out task.

While linear approximation methods have achieved success in controlling neural prostheses, their effectiveness in executing more complex tasks-such as the precise and realistic movement of individual fingers-is somewhat constrained. Neural signals are inherently dynamic and non-stationary, and they are frequently influenced by non-stationary noise. The complexity of neural signals poses significant challenges to the reliability of linear approximations in capturing the full spectrum of nuanced movements required for detailed tasks.

For enhanced and more reliable control of prosthetics in complex tasks, incorporating non-linearity into existing models is crucial. This is because non-linear models are capable of approximating more complex mathematical functions. A prime example of such a function approximator is a neural network. It learns the desired relationship between inputs and outputs by minimizing the discrepancy between estimated and actual outputs through backpropagation. Willsey *et al* [139] introduced the state-of-the-art recalibrated feedback intention-trained neural network (ReFIT-NN), an advancement over ReFIT-KF. ReFIT-NN is specifically designed to accurately decode brain activity related to finger movements, enabling the replication of natural finger motions at high velocity. This shallow feedforward neural network undergoes a two-stage training process: initially, it learns optimized parameters (weights) akin to those in a classic Kalman Filter (KF), and subsequently, it fine-tunes these parameters to correct for any misalignments when the prosthetic limb is not moving towards the target. The efficacy of the ReFIT-NN decoder was tested using data from two Rhesus macaques, each implanted with Utah arrays in M1. The findings revealed that ReFIT-NN surpasses the performance of its linear predecessor, achieving a 36% improvement in throughput during two-dimensional finger movement tasks.

Deep learning techniques such as Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Autoencoders have set new benchmarks, outperforming both linear models and statistical non-linear methods like the VKF in decoding neural signals for kinematic control tasks [2, 26, 31, 76, 82, 85, 122, 126]. RNNs, known for their ability to model temporal dependencies due to their recursive nature [112], along with their variant, the Echo State Network (ESN) [122], have shown superior performance in tasks like reach decoding in monkeys. For instance, ESN surpassed VKF across several metrics including success rate, distance ratio, and average error angle during a center-out reach task. However, it did not match the performance of ReFIT-KF, which benefits from its two-stage online decoding process. Continuous training of RNN-based ESNs has been suggested to enhance performance robustness over time. Nevertheless, a limitation of standard RNNs is their susceptibility to the vanishing gradient problem, leading to difficulties in learning long sequences, which can diminish decoding performance.

LSTMs solve the forgetfulness issue inherent in traditional RNNs, excelling over both RNNs and standard filtering techniques such as the Kalman Filter in decoding locomotion-related hind limb kinematics [42]. This was demonstrated in studies involving non-human primates, with single-unit activity recorded from the leg region of M1 [135]. The effectiveness of LSTMs was further highlighted in processing data from a large neuron population (134–402 neurons) across several brain areas for complex tasks like arm reaching and walking [126]. By incorporating multiple layers, LSTMs adeptly managed high-dimensional data, outperforming traditional filters and capturing essential physiological features of brain activity, such as directional tuning and dynamics.

The robustness and generalization across different brain areas is a key trait which is desired in a neural decoder. In a comprehensive comparison [31] reported that LSTM based neural decoder outperformed most standard decoders, for example the Kalman Filter, SVM, a standard RNN and Naive Bayes. The cursor control task was performed by monkeys when recorded from the motor cortex and somatosensory cortex area of the brain.

In arm movements, the motor cortex encodes directional information more robustly than speed, suggesting that estimating speed and direction separately rather than combined velocity could enhance decoding accuracy [32]. In [85] a specialized dual-mode LSTM (sd-LSTM) was developed to predict speed and direction independently, leveraging the non-linear relationship between neural signals and arm kinematics. Tested on data from non-human primates performing a center-out task with recordings from a 96-channel microelectrode array in M1, utilizing 158 neurons, sd-LSTM demonstrated superior performance over both the VKF and a velocity-predicting LSTM.

Previous studies have developed task-specific neural decoders, limiting their use in diverse behavioral settings. Recognizing the need for decoders that perform well across a broad spectrum of activities, research by Naufel *et al* [76] tested an LSTM-based decoder's ability to map neural signals from the primary motor cortex (M1) to muscle activity across three different motor behaviors: unloaded movements, spring-loaded movements, and isometric contractions. The decoder was trained on these tasks simultaneously, using a weighted cost function inversely proportional to muscle activity variance to mitigate bias toward tasks with higher variability. This method allowed the LSTM to surpass the performance of traditional Wiener Filter decoder, demonstrating its versatility and effectiveness in a dynamic task environment.

Ahmadi *et al* [2] enhanced neural decoding by introducing Entire Spiking Activity (ESA) as an input, used with a Quasi Recurrent Neural Network (Q-RNN) to decode hand movements from M1 neural signals in non-human primates. This ESA-Q-RNN approach outperformed conventional methods (Wiener Filter, Kalman Filter, RNN and LSTM) in accuracy for specific tasks. Q-RNN combines a Convolutional Neural Network (CNN) for parallel data processing and a pooling module for managing temporal dependencies, enabling efficient learning of both short- and long-term neural patterns with reduced computational resources.

The non-stationary nature of neural data means that shifts in implanted electrodes can lead to recordings from different neurons across sessions, causing rapid changes in the relationship between neural activity and behavior. In this context, domain adaptation algorithms are crucial. Farshchian *et al* [26] developed a method using Generative Adversarial Networks (GANs) to transform high-dimensional neural data into a stable, generalized low-dimensional latent space, which is then mapped to decision space using a LSTM. They evaluated the effectiveness of this approach against other domain adaptation techniques, including Kullback-Leibler divergence minimization (KLDM) and canonical correlation analysis (CCA).

Pandarinath *et al* [82] introduced a machine learning method called latent factor analysis via dynamical systems (LFADS), employing non-linear recurrent neural networks (RNNs) to model neural spiking activity as if it were produced by a dynamical system. LFADS, building on variational auto-encoder principles, captures trial-to-trial variability and generates underlying firing rates from observed spiking activity, outperforming other methods like Gaussian process factor analysis in tests.

Makin *et al* [67] aimed to improve motor control through BCIs with a new filter, the recurrent exponential-family harmonium (rEFH), which models spike counts with Poisson distribution and incorporates non-linear dynamics. This method offers a novel approach by not just viewing neural activity as mere observations of kinematic states, but by considering the latent dynamics that could underlie these spike counts.

In conclusion, the application of deep learning methods for decoding movement intentions holds significant promise. Currently, these models primarily operate on CPUs within workstations or laptops. However, efforts are underway to adapt model inference for embedded systems, a transition that requires further research. A notable example is Chen *et al* [15], who have implemented a neural decoder on a field-programmable gate array (FPGA), enabling real-time decoding of neural activity from calcium imaging data. This advancement indicates a move towards more versatile and efficient deployment of neural decoding technologies.

6. Future vision

Recent advancements in analog processing, embedded spike sorting, and neural decoding techniques have significantly improved. A key challenge in migrating algorithms from a remote processor to the on-implant level lies in determining the required signal resolution for quantization. This is essential to minimize computational demands and memory usage.

Currently, the application of deep learning architectures in neuroscience is having a significant impact. However, there remain unresolved issues that necessitate innovative solutions. These are briefly discussed in the subsequent sections.

6.1. Future of analogue processing

Analogue processing research must optimize two main system topologies to support deep learning techniques in future devices. This involves transitioning offline processing methods to hardware, while evaluating the accuracy and noise robustness of the entire pipeline to minimize hardware resources and computational effort. Figure 16 illustrates two strategies for (a) event-based sampling and processing of spike frames and (b) detecting all neural spike events using low-power denoising methods. Further research into NS-ADC topologies is necessary, which includes amplification in the feedback path and allows a time-multiplexing of the input to reduce the power and area consumption. This topology is popular for neural applications due to its low complexity and high-energy efficiency.

To use neuromorphic techniques in future devices, extensive research into long-term stable and compact memristor devices, as well as advanced signal processing and learning techniques is required, to ensure robust and sustainable processing of neural signals. Implementing this topology could lead to enhanced energy efficiency while reducing computational demands.

6.2. Future of urting

The ideal spike sorting algorithm for future devices would operate online, learns autonomously during runtime, and be implemented directly on the implant. Employing deep learning techniques shows promise in enhancing spike sorters to meet these criteria, although currently only a few models achieve this. Therefore, we anticipate key developments in the following spike sorting modules i) spike detection, ii) feature extraction, and iii) clustering.

i) In spike detection, the threshold method is critical, where deep learning topologies can be employed to predict threshold values. These methods can be transferred into the analogue circuits to enable event-based sampling. Also, new deep learning architectures that enable unsupervised and automatic spike detection, including thresholding, are essential, as noise sensitivity significantly impacts both the accuracy and computational efficiency of the entire process.

ii) For feature extraction, we anticipate a significant shift toward the use of autoencoders, given their noise insensitivity and capability for automatic training. However, the computational power required for on-implant spike sorting does not scale efficiently with the increase in electrode numbers. Consequently, we foresee the adoption of simpler feature extractors when dealing with a large array of electrodes

iii) For clustering, the online approaches developed so far are not feasible. While OSort and competitive learning are already available for on-implant computing, their scaling for multichannel spike sorting is not enough and the potential for optimization is limited. We expect to see more density-based clustering methods for online spike sorting.

Concerning architecture, one practical approach could involve the use of a 'shadow' spike sorter. This would provide a pre-trained deep neural network, templates for template matching (TM), or windows for waveform discrimination (WD) to the inference system. Consequently, we anticipate that a subset of incoming spikes will be continuously sent to the shadow spike sorter. While this approach still reduces the data rate, it enables optimized inference. This is particularly important as the scarcity of data presents a significant challenge in deep learning. Overall, we expect an increase in the use of shadow spike sorter systems for adaptive purposes and the implementation of simpler spike sorters for on-implant applications.

We anticipate high-density probes to become increasingly prevalent and more biocompatible in the future. A study on non-human primates demonstrated that Utah arrays led to a 63% reduction in neuron density around the probes [86], a phenomenon that likely also occurs in the human brain. However, while high-density probes significantly increase data rates and cover only a small brain area, BCIs ideally require a broader area to be accessible to the neural decoder. This would necessitate multiple implants in the brain. The advantage of high-density probes is that they simplify the task of

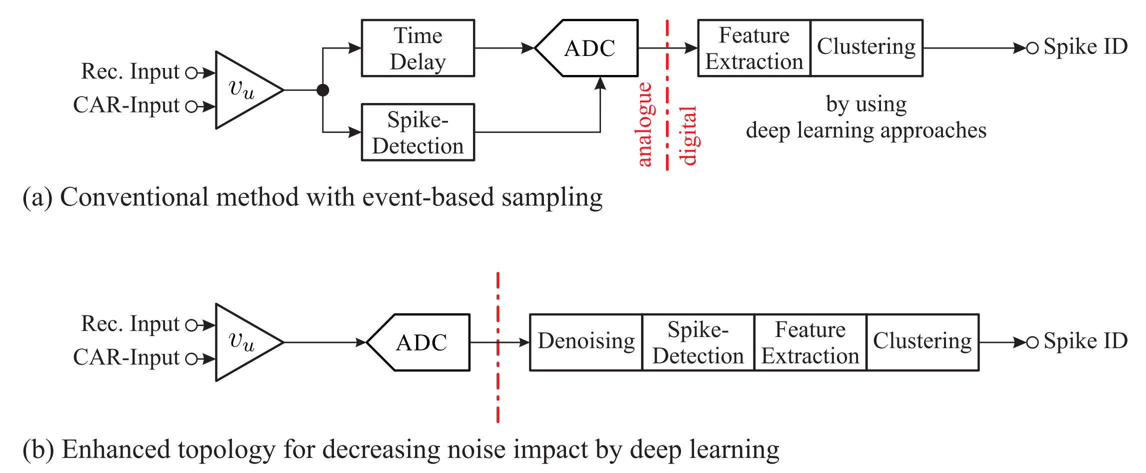

Figure 16. Design topologies for implementing a spike sorting pipeline into hardware for one channel: (a) State-of-the-Art - (b) proposed structure to tackle the crucial threshold impact on the accuracy and computational effort of the pipeline.

spike sorting. As the signal is likely detected by multiple electrodes, noise becomes less of an issue, and noise filtering is enhanced by the spatial information. Consequently, the spike sorting algorithms required could be simpler and more accurate compared to single-channel spike sorters.

In the field of spike sorting, selecting neural channels of interest is a recognized practice. Yet, the automation of this process-choosing electrode channels with high neural activity during recordings-is not widespread. Implementing online selection of channels could significantly enhance the performance of neural decoders by focusing on useful channels.

6.3. Future of neural decoding

Deep learning based neural decoders (among others ESN, rEFH, GAN, LSTM) have outperformed classical methods (among others VKF, WF, KF, WC) in various behavioral tasks such as cursor movement prediction, reach kinematics and wrist EMG predictions [26, 67, 85, 122]. However, the extensive parameter space of these deep learning algorithms demands more computational resources for both training and inference compared to statistical function approximators

The computational demands of deep learning pose significant challenges for effectively integrating these architectures into embedded hardware systems implanted in the body. For real-time or clinical applications, these algorithms must not only be highly accurate but also require reduced computational costs to function efficiently within embedded systems. Future advancements must focus on optimizing these methods for hardware execution, taking into account resource consumption, computational effort, performance, and latency. Currently, various deep learning techniques such as fully-connected neural networks [1], convolutional neural networks [6], and LSTM cells [69, 92] are implementable in such devices.

7. Conclusion

This survey paper provides a detailed overview of the current state of the art in end-to-end signal processing of brain-computer interfaces with invasive electrode arrays for movement intention detection.

Great progress has been made in all of these areas in recent years. We described and classified different SOTA methods and techniques for analogue processing, spike sorting and neural decoding. Also, we highlighted the advantages and drawbacks of those methods for possible on-implant integrations. In addition, we could show that different system architectures for spike sorting-based systems can be useful under certain circumstances. Concluding our observations, we briefly presented our future vision for next-generation BCI pipelines. We expect the use of deep learning methods in neuroscience will make further advances. Thus, we are optimistic that future implementations in hardware will be developed so that patients can receive more benefits.

Data availability statement

The data cannot be made publicly available upon publication because no suitable repository exists for hosting data in this field of study. The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

Andreas Erbslöh https://orcid.org/0000-0001-6702-892X
Leo Buron https://orcid.org/0009-0001-8939-4784
Zia Ur-Rehman https://orcid.org/0009-0003-1281-4829
Simon Musall https://orcid.org/0000-0002-9461-1042

Philipp Löhler https://orcid.org/0000-0003-1473-9853
Christian Klaes https://orcid.org/0000-0003-4767-9631
Karsten Seidl https://orcid.org/0000-0001-6197-5037
Gregor Schiele https://orcid.org/0000-0003-4266-4828

References

[1] Abdelsalam A M, Boulet F, Demers G, Pierre Langlois J M and Cheriet F 2018 An efficient fpga-based overlay inference architecture for fully connected dnns *Int. Conf. on ReConFigurable Computing and FPGAs (ReConFig)*
[2] Ahmadi N, Constandinou T G and Bouganis C-S 2021 Robust and accurate decoding of hand kinematics from entire spiking activity using deep learning *J. Neural Eng.* **18** 026011
[3] Arthur D and Vassilvitskii S 2007 K-means++: The advantages of careful seeding *Proc. 18th Annual ACM-SIAM Symp. on Discrete Algorithms* (Society for Industrial and Applied Mathematics) pp 1027–35
[4] Astrom K and Bernhardsson B 2002 Comparison of riemann and lebesgue sampling for first order stochastic systems *Proc. 41st IEEE Conf. on Decision and Control*
[5] Bengio Y 2009 Learning deep architectures for AI *Found. Trends® Mach. Learn.* **2** 1–127
[6] Bouguezzi S, Fredj H B, Belabed T, Valderrama C, Faiedh H and Souani C 2021 An efficient FPGA-based convolutional neural network for classification: ad-mobilenet *Electronics* **10** 2272
[7] Buccino A P, Garcia S and Yger P 2022 Spike sorting: new trends and challenges of the era of high-density probes *Prog. Biomed. Eng.* **4** 022005
[8] Burger A, Urban P, Boubin J and Schiele G 2020 An architecture for solving the eigenvalue problem on embedded FPGAS *Architecture of Computing Systems* (Springer International Publishing) pp 32–43
[9] Burle B, Spieser L, Roger C, Casini L, Hasbroucq T and Vidal F 2015 Spatial and temporal resolutions of EEG: Is it really black and white? a scalp current density view *Int. J. Psychophysiol.* **97** 210–20
[10] Chae M S, Yang Z, Yuce M R, Hoang L and Liu W 2009 A 128-channel 6 mw wireless neural recording IC with spike feature extraction and UWB transmitter *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 312–21
[11] Chandrakumar H and Marković D 2015 A simple area-efficient ripple-rejection technique for chopped biosignal amplifiers *IEEE Trans. Circuits Syst.* II **62** 189–93
[12] Chandrakumar H and Marković D 2016 A 2 μw 40 mvpp linear-input-range chopper- stabilized bio-signal amplifier with boosted input impedance of 300 mΩ and electrode-offset filtering *IEEE Int. Solid-State Circuits Conf. (ISSCC)*
[13] Chang Y-J, Hwang W-J and Chen C-C 2016 A low cost VLSI architecture for spike sorting based on feature extraction with peak search *Sensors* **16** 2084
[14] Chen H-Y, Chen C-C and Hwang W-J 2017 An efficient hardware circuit for spike sorting based on competitive learning networks *Sensors* **17** 2232
[15] Chen Z, Blair G J, Guo C, Zhou J, Romero-Sosa J-L, Izquierdo A, Golshani P, Cong J, Aharoni D and Blair H T 2023 A hardware system for real-time decoding of *in vivo* calcium imaging data *eLife* **12** e78344
[16] Chestek C A, Gilja V, Nuyujukian P, Kier R J, Solzbacher F, Ryu S I, Harrison R R and Shenoy K V 2009 Hermesc: low-power wireless neural recording system for freely moving primates *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 330–8
[17] Choi J H, Jung H K and Kim T 2006 A new action potential detector using the mteo and its effects on spike sorting systems at low signal-to-noise ratios *IEEE Trans. Biomed. Eng.* **53** 738–46
[18] Chung J E, Magland J F, Barnett A H, Tolosa V M, Tooker A C, Lee K Y, Shah K G, Felix S H, Frank L M and Greengard L F 2017 A fully automated approach to spike sorting *Neuron* **95** 1381–94
[19] Corradi F and Indiveri G 2015 A neuromorphic event-based neural recording system for smart brain-machine-interfaces *IEEE Trans. Biomed. Circuits Syst.* **9** 699–709
[20] Djekic D, Fantner G, Lips K, Ortmanns M and Anders J 2018 A 0.1% THD, 1-MΩ to 1-GΩ tunable, temperature-compensated transimpedance amplifier using a multi-element pseudo-resistor *IEEE J. Solid-State Circuits* **53** 1913–23
[21] Do A T, Zeinolabedin S M A, Jeon D, Sylvester D and Kim T T-H 2019 An area-efficient 128-channel spike sorting processor for real-time neural recording with 0.175μW/channel in 65-nm CMOS *IEEE Trans. Very Large Scale Integr. Syst.* **27** 126–37
[22] Dutta B *et al* 2019 The neuropixels probe: a CMOS based integrated microsystems platform for neuroscience and brain-computer interfaces *IEEE Int. Electron Devices Meeting* pp 10.1.1–4
[23] Dwivedi S and Gogoi A K 2018 A novel adaptive real-time detection algorithm for an area-efficient CMOS spike detector circuit *AEU - Int. J. Electr. Commun.* **88** 87–97
[24] Erbslöh A, Viga R, Seidl K and Kokozinski R 2021 Artefact-suppressing analog spike detection circuit for firing-rate measurements in closed-loop neurostimulators *IEEE Sens. J.* **22** 11328–35
[25] Fan Q, Sebastiano F, Huijsing J H and Makinwa K A A 2011 A 1.8 μW 60 nV/$\sqrt{}$ Hz capacitively-coupled chopper instrumentation amplifier in 65 nm CMOS for wireless sensor nodes *IEEE J. Solid-State Circuits* **46** 1534–43
[26] Farshchian A, Gallego J A, Miller L E, Solla S A, Cohen J P and Bengio Y 2018 Adversarial domain adaptation for stable brain-machine interfaces *7th Int. Conf. on Learning Representations, ICLR 2019*
[27] Fraser G W, Chase S M, Whitford A and Schwartz A B 2009 Control of a brain-computer interface without spike sorting *J. Neural Eng.* **6** 055004
[28] Georgopoulos A P, Lurito J T, Petrides M, Schwartz A B and Massey J T 1989 Mental rotation of the neuronal population vector *Science* **243** 234–6
[29] Gibson S, Judy J and Marković D 2012 Spike sorting: The first step in decoding the brain: the first step in decoding the brain *IEEE Signal Process. Mag.* **29** 124–43
[30] Gilja V *et al* 2012 A high-performance neural prosthesis enabled by control algorithm design *Nat. Neurosci.* **15** 1752–7
[31] Glaser J I, Benjamin A S, Chowdhury R H, Perich M G, Miller L E and Kording K P 2020 Machine learning for neural decoding *Eneuro* **7** 1–16
[32] Golub M D, Yu B M, Schwartz A B and Chase S M 2014 Motor cortical control of movement speed with implications for brain-machine interface control *J. Neurophysiol.* **112** 411–29
[33] Grahn P, Mallory G, Berry M, Hachmann J, Lobel D and Lujan L 2014 Restoration of motor function following spinal cord injury via optimal control of intraspinal microstimulation: toward a next generation closed-loop neural prosthesis *Front. Neurosci.* **8** 296
[34] Guglielmi E, Toso F, Zanetto F, Sciortino G, Mesri A, Sampietro M and Ferrari G 2020 High-value tunable pseudo-resistors design *IEEE J. Solid-State Circuits* **55** 2094–105
[35] Gupta I, Serb A, Khiat A, Trapatseli M and Prodromakis T 2019 Spike sorting using non-volatile metal-oxide memristors *Faraday Discuss.* **213** 511–20

[36] Gupta I, Serb A, Khiat A, Zeitler R, Vassanelli S and Prodromakis T 2016 Real-time encoding and compression of neuronal spikes by metal-oxide memristors *Nat. Commun.* **7** 12805
[37] Haessig G, Lesta D G, Lenz G, Benosman R and Dudek P 2020 A mixed-signal spatio-temporal signal classifier for on-sensor spike sorting *IEEE Int. Symp. on Circuits and Systems (ISCAS)*
[38] Hao H, Chen J, Richardson A G, der Spiegel J V and Aflatouni F 2021 A 10.8 μW neural signal recorder and processor with unsupervised analog classifier for spike sorting *IEEE Trans. Biomed. Circuits Syst.* **15** 351–64
[39] Harrison R R, Watkins P T, Kier R J, Lovejoy R O, Black D J, Greger B and Solzbacher F 2007 A low-power integrated circuit for a wireless 100-electrode neural recording system *IEEE J. Solid-State Circuits* **42** 123–33
[40] Henze D A, Borhegyi Z, Csicsvari J, Mamiya A, Harris K D and Buzsáki G (2000). Intracellular features predicted by extracellular recordings in the hippocampus *in vivo J. Neurophysiol.* **84** 390–400
[41] Hestenes M R 1958 Inversion of matrices by biorthogonalization and related results *J. Soc. Indus. Appl. Math.* **6** 51–90
[42] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
[43] Hotelling H 1933 Analysis of a complex of statistical variables into principal components *J. Educ. Psychol.* **24** 417–41
[44] Hwang W-J, Lee W-H, Lin S-J and Lai S-Y 2013 Efficient architecture for spike sorting in reconfigurable hardware *Sensors* **13** 14860–87
[45] Im M and Kim S 2020 Neurophysiological and medical considerations for better performing microelectronic retinal prothesis *J. Neural Eng.* **17** 1–9
[46] Japkowicz N, Hanson S J and Gluck M A 2000 Nonlinear autoassociation is not equivalent to PCA *Neural Comput.* **12** 531–45
[47] Jiang Z, Cerqueira J P, Kim S, Wang Q and Seok M 2016 1.74-μw/ch, 95.3%-accurate spike-sorting hardware based on bayesian decision *IEEE Symp. on VLSI Circuits (VLSI-Circuits)*
[48] Jie L, Tang X, Liu J, Shen L, Li S, Sun N and Flynn M P 2021 An overview of noise-shaping sar adc: From fundamentals to the frontier *IEEE Open J. Solid-State Circuits Soc.* **1** 149–61
[49] Karkare V, Gibson S and Marković D 2013 A 75-μW, 16-channel neural spike-sorting processor with unsupervised clustering *IEEE J. Solid-State Circuits* **48** 2230–8
[50] Kathe C *et al* 2022 The neurons that restore walking after paralysis *Nature* **611** 540–7
[51] Kechris C, Delitzas A, Matsoukas V and Petrantonakis P C 2021 Removing noise from extracellular neural recordings using fully convolutional denoising autoencoders *43rd Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)*
[52] Kim C, Joshi S, Courellis H, Wang J, Miller C and Cauwenberghs G 2018 Sub- μ V_{rms}-noise sub- μW/channel ADC-direct neural recording with 200-mV/ms transient recovery through predictive digital autoranging *IEEE J. Solid-State Circuits* **53** 3101–10
[53] Kim J and Ko H 2019 Self-biased ultralow power current-reused neural amplifier with on-chip analog spike detections *IEEE Access* **7** 109
[54] Kim J P, Lee H and Ko H 2018 0.6 V, 116 nW neural spike acquisition IC with self-biased instrumentation amplifier and analog spike extraction *Sensors* **8** 1–13
[55] Kim S-P, Simeral J D, Hochberg L R, Donoghue J P and Black M J 2008 Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia *J. Neural Eng.* **5** 455–76
[56] kin Tam W and Nolan M F 2022 pyNeurode: a real-time neural signal processing framework (available at: https://ieeexplore.ieee.org/document/9937512)
[57] Li Z, Wang Y, Zhang N and Li X 2020 An accurate and robust method for spike sorting based on convolutional neural networks *Brain Sci.* **10** 835
[58] Liu X, Zhang M, Richardson A G, Lucas T H and der Spiegel J V 2017 Design of a closed-loop, bidirectional brain machine interface system with energy efficient neural feature extraction and pid control *IEEE Trans. Biomed. Circuit Syst.* **11** 729–42
[59] Liu Y, Sheng J and Herbordt M C 2016 A hardware design for in-brain neural spike sorting *IEEE High Performance Extreme Computing Conf. (HPEC)*
[60] Liu Y, Zhou Z, Zhou Y, Li W and Wang Z 2020 A low-noise chopper amplifier with offset and low-frequency noise compensation dc servo loop *Electronics* **9** 1–12
[61] Livezey J A and Glaser J I 2020 Deep learning approaches for neural decoding across architectures and recording modalities *Brief. Bioinf.* **22** 1577–91
[62] Lloyd S 1982 Least squares quantization in PCM *IEEE Trans. Inf. Theory* **28** 129–37
[63] Lu Y, Zhou T, Huang J, Wang L, Chen M and Li Y 2021 Msb-split VCM-based charge recovery symmetrical switching with set-and-down asymmetrical switching method for dual-capacitive arrays SAR ADC *Analog Integr. Circuit Signal Process.* **106** 669–81
[64] Ludwig K A, Miriani R M, Langhals N B, Joseph M D, Anderson D J and Kipke D R 2009 Using a common average reference to improve cortical neuron recordings from microelectrode arrays *J. Neurophysiol.* **101** 1679–89
[65] Ma T-C, Chen T-C and Chen L-G 2014 Design and implementation of a low power spike detection processor for 128-channel spike sorting microsystem *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*
[66] Magland J F and Barnett A H 2015 Unimodal clustering using isotonic regression: Iso-split (arXiv:1508.04841)
[67] Makin J G, O'Doherty J E, Cardoso M M and Sabes P N 2018 Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm *J. Neural Eng.* **15** 026010
[68] Marwan B, Samann F and Schanze T 2022 Denoising of ECG with single and multiple hidden layer autoencoders *Curr. Directions Biomed. Eng.* **8** 652–5
[69] Mazumder A N, Rashid H-A and Mohsenin T 2020 An energy-efficient low power lstm processor for human activity monitoring *IEEE 33rd Int. System-on-Chip Conf. (SOCC)*
[70] Moghaddasi M, Shoorehdeli M A, Fatahi Z and Haghparast A 2020 Unsupervised automatic online spike sorting using reward-based online clustering *Biomed. Signal Process. Control* **56** 101701
[71] Montes V, Gehlen J, Lück S, Mokwa W, Müller F, Walter P and Offenhäusser A 2019 Towards a bidirectional communication between retinal cells and a prosthetic device - a proof of concept *Front. Neurosci.* **13** 1–19
[72] Mukhopadhyay S and Ray G 1998 A new interpretation of nonlinear energy operator and its efficacy in spike detection *IEEE Trans. Biomed. Eng.* **45** 180–7
[73] Muratore D G, Tandon P, Wootters M, Chichilnisky E J, Mitra S and Murmann B 2019 A data-compressive wired-or readout for massively parallel neural recording *IEEE Trans. Biomed. Circuits Syst.* **13** 1128–40
[74] Musk E and Neuralink 2019 An integrated brain-machine interface platform with thousands of channels *J. Med. Internet Res.* **21** e16194
[75] Nason S R *et al* 2020 A low-power band of neuronal spiking activity dominated by local single units improves the performance of brain-machine interfaces *Nat. Biomed. Eng.* **4** 973–83
[76] Naufel S, Glaser J I, Kording K P, Perreault E J and Miller L E 2019 A muscle-activity-dependent gain between motor cortex and EMG *J. Neurophysiol.* **121** 61–73

[77] Nejedly P *et al* 2023 Utilization of temporal autoencoder for semi-supervised intracranial EEG clustering and classification *Sci. Rep.* **13** 744
[78] Nenadic Z and Burdick J 2005 Spike detection using the continuous wavelet transform *IEEE Trans. Biomed. Eng.* **52** 74–87
[79] Noshahr H, Nabavi F M and Sawan M 2020 Multi-channel neural recording implants: a review *Sensors* **20** 1–29
[80] Okreghe C O, Zamani M and Demosthenous A 2023 A deep neural network-based spike sorting with improved channel selection and artefact removal *IEEE Access* **11** 15131–43
[81] Pachitariu M, Steinmetz N, Kadir S and Carandini M 2016 Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels *bioRxiv Preprint* https://doi.org/10.1101/061481 (posted online 30 June 2016, accessed 12 February 2023)
[82] Pandarinath C *et al* 2018 Inferring single-trial neural population dynamics using sequential auto-encoders *Nat. Methods* **15** 805–15
[83] Paninski L, Fellows M R, Hatsopoulos N G and Donoghue J P 2004 Spatiotemporal tuning of motor cortical neurons for hand position and velocity *J. Neurophysiol.* **91** 515–32
[84] Paraskevopoulou S E, Barsakcioglu D Y, Saberi M R, Eftekhar A and Constandinou T G 2013 Feature extraction using first and second derivative extrema (FSDE) for real-time and hardware-efficient spike sorting *J. Neurosci. Methods* **215** 29–37
[85] Park J and Kim S P 2019 Estimation of speed and direction of arm movements from m1 activity using a nonlinear neural decoder *7th Int. Winter Conf. on Brain-Computer Interface, BCI 2019* p 77
[86] Patel P, Welle E, Letner J, Shen H, Bullard A, Caldwell C, Vega-Medina A, Richie J, Thayer H and Patil P 2023 Utah array characterization and histological analysis of a multi-year implant in non-human primate motor and sensory cortices *J. Neural Eng.* **20** 014001
[87] Pathak R, Dash S, Mukhopadhyay A K, Basu A and Sharad M 2017 Low power implantable spike sorting scheme based on neuromorphic classifier with supervised training engine *IEEE Computer Society Annual Symp. on VLSI (ISVLSI)*
[88] Pearson K 1901 LIII. Ion lines and planes of closest fit to systems of points in space/i *London, Edinburgh Dublin Phil. Mag. J. Sci.* **2** 559–72
[89] Pedreira C, Martinez J, Ison M and Quian Quiroga R 2012 How many neurons can we see with current spike sorting algorithms? *J. Neurosci. Methods* **211** 58–65
[90] Pham X T, Vu T K, Nguyen T D and Pham-Nguyen L 2022 A 1.2 μw 41 db ripple attenuation chopper amplifier using auto-zero offset cancelation loop for area-efficient biopotential sensing *Electronics* **11** 1149
[91] Pérez-Prieto N and Delgado-Restituto M 2021 Recording strategies for high channel count, densely spaced microelectrode arrays *Fronties* **15** 1–21
[92] Qian C, Ling T and Schiele G 2023 Energy efficient lstm accelerators for embedded fpgas through parameterised architecture design *Architecture of Computing Systems* pp 3–17
[93] Quiroga R Q, Nadasdy Z and Ben-Shaul Y 2004 Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering *Neural Comput.* **16** 1661–87
[94] Rácz M, Liber C, Németh E, Fiáth R, Rokai J, Harmati I, Ulbert I and Márton G 2020 Spike detection and sorting with deep learning *J. Neural Eng.* **17** 016038
[95] Radhakrishnan A, Friedman S F, Khurshid S, Ng K, Batra P, Lubitz S A, Philippakis A A and Uhler C 2023 Cross-modal autoencoder framework learns holistic representations of cardiovascular state *Nat. Commun.* **14** 2436
[96] Radmanesh M, Rezaei A A, Jalili M, Hashemi A and Goudarzi M M 2022 Online spike sorting via deep contractive autoencoder *Neural Netw.* **155** 39–49
[97] Ramasubbu R, Lang S and Kiss Z H 2018 Dosing of electrical parameters in deep brain stimulation (DBS) for intractable depression: a review of clinical studies *Front. Psychiatry* **9** 302
[98] Reich S, Fritschi D, Sporer M and Ortmanns M 2023 In vitro study of artifact-recovery using a 32-channel neuromodulator platform *IEEE Trans. Circuits Syst.* I **70** 1–10
[99] Reich S, Kunze G, Sporer M and Ortmanns M 2022 Analysis of chopper ripple reduction by delayed sampling *17th Conf. on Ph.D Research in Microelectronics and Electronics (PRIME)*
[100] Reich S, Sporer M, Haas M, Becker J, Schüttler M and Ortmanns M 2021 A high-voltage compliance, 32-channel digitally interfaced neuromodulation system on chip *IEEE J. Solid-State Circuits* **56** 2476–87
[101] Reich S, Sporer M and Ortmanns M 2021 A chopped neural front-end featuring input impedance boosting with suppressed offset-induced charge transfer *IEEE Trans. Biomed. Circuits Syst.* **15** 402–11
[102] Rey H G, Pedreira C and Quian Quiroga R 2015 Past, present and future of spike sorting techniques *Brain Res. Bull.* **119** 106–17
[103] Reyes c, Jaramillo F, Zhang B, Kulkarni C and Orchard M 2018 Just-in-time point prediction using a computationally-efficient lebesgue-sampling-based prognostic method: application to battery end-of-discharge prediction *PHM_CONF* 10
[104] Rozgić D, Hokhikyan V, Jiang W, Akita I, Basir-Kazeruni S, Chandrakumar H and Marković D 2019 A 0.338 cm^3, artifact-free, 64-contact neuromodulation platform for simultaneous stimulation and sensing *IEEE Trans. Biomed. Circuits Syst.* **13** 38–55
[105] Rutishauser U, Schuman E M and Mamelak A N (2006). Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo J. Neurosci. Methods* **154** 204–24
[106] Saggese G and Strollo A G M 2021 A low power 1024-channels spike detector using latch-based ram for real-time brain silicon interfaces *Electronics* **10** 3068
[107] Saggese G, Tambaro M, Vallicelli E A, Strollo A G M, Vassanelli S, Baschirotto A and Matteis M D 2021 Comparison of sneo-based neural spike detection algorithms for implantable multi-transistor array biosensors *Electronics* **10** 410
[108] Samann F and Schanze T 2022 Multiple parallel hidden layers autoencoder for denoising ecg signal *Curr. Dir. Biomed. Eng.* **8** 161–4
[109] Samiei A and Hashemi H 2019 A chopper stabilized, current feedback, neural recording amplifier *IEEE Solid-State Circuits Lett.* **2** 17–20
[110] Schaffer L, Nagy Z, Kincses Z, Fiath R and Ulbert I 2021 Spatial information based OSort for real-time spike sorting using FPGA *IEEE Trans. Biomed. Eng.* **68** 99–108
[111] Schmidhuber J 2015 Deep learning in neural networks: an overview *Neural Netw.* **61** 85–117
[112] Schmidt R M 2019 Recurrent neural networks (RNNS): a gentle introduction and overview (arXiv:1912.05911)
[113] Seidl K, Schwaerzle M, Ulbert I, Neves H P, Paul O and Ruther P 2012 Cmos-based high-density silicon microprobe arrays for electronic depth control in intracortical neural recording-characterization and application *J. Microelectromech. Syst.* **21** 1426–35
[114] Seong C, Lee W and Jeon D 2021 A multi-channel spike sorting processor with accurate clustering algorithm using convolutional autoencoder *IEEE Trans. Biomed. Circuits Syst.* **15** 1441–53
[115] Serruya M D, Hatsopoulos N G, Paninski L, Fellows M R and Donoghue J P 2002 Instant neural control of a movement signal *Nature* **416** 141–2
[116] Shaeri M and Sodagar A M 2020 A framework for on-implant spike sorting based on salient feature selection *Nat. Commun.* **11** 1–9

[117] Shi Y, Ananthakrishnan A, Oh S, Liu X, Hota G, Cauwenberghs G and Kuzum D 2022 A neuromorphic brain interface based on rram crossbar arrays for high throughput real-time spike sorting *IEEE Trans. Electron Devices* **69** 2137–44

[118] Shu Y-S, Kuo L-T and Lo T-Y 2016 An oversampling SAR ADC with DAC mismatch error shaping achieving 105 db SFDR and 101 db sndr over 1 khz bw in 55 nm cmos *IEEE J. Solid-State Circuits* **51** 2928–40

[119] Sporer M, Reich S, Kauffman J G and Ortmanns M 2022 A direct digitizing chopped neural recorder using a body-induced offset based dc servo loop *IEEE Trans. Biomed. Circuits Syst.* **16** 409–18

[120] Sulaiman M B G, Juang K-C and Lu C-C 2020 Weight quantization in spiking neural network for hardware implementation *IEEE Int. Conf. on Consumer Electronics - Taiwan (ICCE-Taiwan)*

[121] Sun J, Li T, Guo T, Li Y, Fu C and Liu Y 2022 Toward ultra-large scale neural spike sorting with distributed sorting channels and unsupervised training *IEEE Int. Symp. on Circuits and Systems (ISCAS)*

[122] Sussillo D, Nuyujukian P, Fan J M, Kao J C, Stavisky S D, Ryu S and Shenoy K 2012 A recurrent neural network for closed-loop intracortical brain-machine interface decoders *J. Neural Eng.* **9** 026027

[123] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* (MIT press)

[124] Szostak K, Grand L and Constandinou T 2017 Neural interfaces for intracortical recording: Requirements, fabrication methods and characteristics *Front. Neurosci.* **11** 665

[125] Tam W, Wu T, Zhao Q, Keefer E and Yang Z 2019 Human motor decoding from neural signals: a review *BMC Biomed. Eng.* **1** 22

[126] Tseng P-H, Urpi N A, Lebedev M and Nicolelis M 2019 Decoding movements from cortical ensemble activity using a long short-term memory recurrent network *Neural Comput.* **31** 1085–113

[127] Tsuchimoto S, Shibusawa S, Iwama S, Hayashi M, Okuyama K, Mizuguchi N, Kato K and Ushiba J 2021 Use of common average reference and large-laplacian spatial-filters enhances eeg signal-to-noise ratios in intrinsic sensorimotor activity *J. Neurosci. Methods* **353** 109089

[128] ur Rehman M S *et al* 2020 SpikeDeep-classifier: a deep-learning based fully automatic offline spike sorting algorithm *J. Neural Eng.* **18** 1–22

[129] ur Rehman M S *et al* 2019 Spikedeeptector: a deep-learning based method for detection of neural spiking activity *J. Neural Eng.* **16** 056003

[130] Valencia D and Alimohammad A 2019 An efficient hardware architecture for template matching-based spike sorting *IEEE Trans. Biomed. Circuits Syst.* **13** 481–92

[131] Valencia D and Alimohammad A 2019 A real-time spike sorting system using parallel OSort clustering *IEEE Trans. Biomed. Circuits Syst.* **13** 1700–13

[132] Valencia D and Alimohammad A 2021 Neural spike sorting using binarized neural networks *IEEE Trans. Neural Syst. Rehabil. Eng.* **29** 206–14

[133] Wang T-H, Wu R, Gupta V and Li S 2021 A 13.8-ENOB 0.4 PF-CIN 3rd-order noise-shaping sar in a single-amplifier EF-CIFF structure with fully dynamic hardware-reusing KT/C noise cancelation *IEEE Int. Solid- State Circuits Conf. (ISSCC)*

[134] Wang X, Wang H and Pun K-P 2021 A capacitor-reused 2b/cycle active-passive second-order noise-shaping SAR ADC *Solid State Electron. Lett.* **3** 27–31

[135] Wang Y, Truccolo W and Borton D A 2018 Decoding hindlimb kinematics from primate motor cortex using long short-term memory recurrent neural networks *Proc. Annual Int. Conf. IEEE Engineering in Medicine and Biology Society, EMBS 2018-July* pp 1944–7

[136] Warland D K, Reinagel P and Meister M 1997 Decoding visual information from a population of retinal ganglion cells *J. Neurophysiol.* **78** 2336–50

[137] Werner T, Vianello E, Bichler O, Garbin D, Cattaert D, Yvert B, De Salvo B and Perniola L 2016 Spiking neural networks based on oxram synapses for real-time unsupervised spike sorting *Front. Neurosci.* **10** 474

[138] Willett F, Avansino D T, Hochberg L R, Henderson J M and Shenoy K V 2021 High-performance brain-to-text communication via handwriting *Nature* **593** 249–54

[139] Willsey M S, Nason-Tomaszewski S R, Ensel S R, Temmar H, Mender M J, Costello J T, Patil P G and Chestek C A 2022 Real-time brain-machine interface in non-human primates achieves high-velocity prosthetic finger movements using a shallow feedforward neural network decoder *Nat. Commun.* **13** 6899

[140] Wu W, Black M, Gao Y, Bienenstock E, Serruya M, Shaikhouni A and Donoghue J 2003 Neural decoding of cursor motion using a kalman filter *Advances in Neural Information Processing Systems* vol 15

[141] Yang Y, Boling S and Mason A J 2017 A hardware-efficient scalable spike sorting neural signal processor module for implantable high-channel-count brain machine interfaces *IEEE Trans. Biomed. Circuits Syst.* **11** 743–54

[142] Yao E, Chen Y and Basu A 2016 A 0.7 v, 40 nw compact, current-mode neural spike detector in 65 nm cmos *IEEE Trans. Biomed. Circuits Syst.* **10** 309–18

[143] Yger P and Gilson M 2015 Models of metaplasticity: a review of concepts *Front. Comput. Neurosci.* **9** 138

[144] Young N, Collins C and Kaas J 2013 Cell and neuron densities in the primary motor cortex of primates *Front. Neural Circuits* **7** 30

[145] Yu H, Qi Y and Pan G 2023 Neusort: an automatic adaptive spike sorting approach with neuromorphic models *J. Neural Eng.* **20** 056006

[146] Zacksenhouse M, Lebedev M A, Carmena J M, O'Doherty J E, Henriquez C and Nicolelis M A 2007 Cortical modulations increase in early sessions with brain-machine interface *PLoS One* **2** 1–10

[147] Zamani M, Jiang D and Demosthenous A 2018 An adaptive neural spike processor with embedded active learning for improved unsupervised sorting accuracy *IEEE Trans. Biomed. Circuits Syst.* **12** 665–76

[148] Zamani M, Sokolić J, Jiang D, Renna F, Rodrigues M R D and Demosthenous A 2020 Accurate, very low computational complexity spike sorting using unsupervised matched subspace learning *IEEE Trans. Biomed. Circuits Syst.* **14** 221–31

[149] Zeinolabedin S M A *et al* 2022 A 16-channel fully configurable neural soc with 1.52 μw/ch signal acquisition, 2.79 μw/ch real-time spike classifier and 1.79 tops/w deep neural network accelerator in 22 nm FDSOI *IEEE Trans. Biomed. Circuits Syst.* **16** 94–107

[150] Zhang T, Azghadi M R, Lammie C, Amirsoleimani A and Genov R 2023 Spike sorting algorithms and their efficient hardware implementation: a comprehensive survey *J. Neural Eng.* **20** 021001

[151] Zhang Y, Valsecchi M, Gegenfurtner K R and Chen J 2023 Laplacian reference is optimal for steady-state visual-evoked potentials *J. Neurophysiol.* **130** 557–68

[152] Zhang Z and Constandinou T G 2021 Adaptive spike detection and hardware optimization towards autonomous, high-channel-count bmis *J. Neurosci. Methods* **354** 109103

[153] Zhang Z, Feng P, Oprea A and Constandinou T G 2023 Calibration-free and hardware-efficient neural spike detection for brain machine interfaces *IEEE Trans. Biomed. Circuits Syst.* **17** 725–40

[154] Zhang Z, Savolainen O W and Constandinou T G 2022 Algorithm and hardware considerations for real-time neural signal on-implant processing *J. Neural Eng.* **19** 016029