Published in final edited form as:
J Neural Eng. ; 16(6): 066022. doi:10.1088/1741-2552/ab3319.

Improving an open-source commercial system to reliably perform activity-dependent stimulation

Maxwell Murphy[1,2,*], **Stefano Buccelli**[3,4,*], **Yannick Bornat**[5], **David Bundy**[1], **Randolph Nudo**[1], **David Guggenmos**[†,1], **Michela Chiappalone**[†,3]

[1]Department of Rehabilitation Medicine, University of Kansas Medical Center, 3901 Rainbow Boulevard, Kansas City, KS, United States 66160

[2]Bioengineering Graduate Program, University of Kansas, Lawrence, Kansas, United States

[3]Rehab Technologies IIT-INAIL Lab, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

[4]Department of Neuroscience, Rehabilitation, Ophthalmology, Genetics and Maternal and 15 Child science (DINOGMI), University of Genova, L.go P. Daneo 3, 16132 Genova, Italy

[5]Laboratoire de l'Intégration du Matériau au Système (IMS), University of Bordeaux, Bordeaux INP, CNRS UMR 5218, 351 Cours de la Libération, 33405 Talence Cedex, France

Abstract

Objective.—Activity-dependent stimulation (ADS) is designed to strengthen the connections between neuronal circuits and therefore may be a promising tool for promoting neurophysiological reorganization following a brain injury. To successfully perform this technique, two criteria must be met: 1) spikes in the extracellular electrical field potential must be detected accurately at one site of interest, and 2) stimulation pulses generated at fixed (< 1ms jitter), low-latency (< 10ms) intervals relative to each detected spike must be delivered reliably to a second site of interest. Here, we aimed to improve noise rejection in a low-cost commercial system to reliably perform ADS in awake, behaving rats, while maintaining latency requirements.

Approach.—We implemented a spike detection state machine on a field-programmable gate array (FPGA). Because the accuracy of spike detection can be heavily reduced in awake and behaving animals due to biological artifacts such as movement and chewing, the state machine tracks candidate spike waveforms, checking them against multiple programmable thresholds and rejecting any spikes that fail to meet a programmed threshold criterion.

Main Results.—A series of offline analyses showed that our implementation was able to appropriately trigger stimulation during epochs of biological artifacts with an overall accuracy between 72% and 97%, fixed computational latency of 167μs, and an algorithmic latency of 300μs to 800μs.

dguggenmos@kumc.edu.

[*]first author equal contribution

[†]last author equal contribution

Significance.—Our improvements have been made open-source and are freely available to all scientists working on closed-loop neuroprosthetic devices. Importantly, the improvements are easily incorporated into existing workflows that utilize the Intan Stimulation and Recording Controller.

Keywords

FPGA; neuroprosthetics; spike detection

1. Introduction

Recent preclinical work has investigated the feasibility and efficacy of intracortical microstimulation (ICMS) coupled to neural activity to promote rehabilitation after brain injury (Guggenmos 2013, Azin 2011a, b). In brain-injured rats, constraining the timing of ICMS to within a few milliseconds of a detected extracellular action potential recorded in a second area improves motor skill beyond that achieved by randomly timed stimuli (Guggenmos 2013). This ICMS paradigm, known as activity-dependent stimulation (ADS), has also been used in healthy macaques to pair sites within motor cortex and alter evoked EMG output (Jackson 2006). The efficacy of these protocols relies both upon the accuracy of the spike detector and upon the reliability of subsequent low-latency (<10 ms) delivery of ICMS. Furthermore, because the invoked strengthening of connections between sites is thought to be generated by a Hebbian mechanism, low jitter in the delivery of stimuli (<1 ms) is critical; for example, the difference in timing between invoking maximal potentiation and maximal depression of synaptic efficacy in hippocampal cultures is <5 ms (Bi and Poo 1998). Depending upon the distance, type, and number of synapses that are putatively involved between the targets of ADS, it is also possible that the <10 ms latency constraint may be restricted to as low as <3–4 ms.

Historically, spike detection has been performed by applying a monopolar voltage threshold to the amplified and filtered neurophysiological signal, counting each rising edge of the resultant logical signal as the onset of a spike (Cheney and Fetz 1985). However, spike detection done in this way tends to conflate signals generated by movement and chewing with spikes from neural units when used in awake animal experiments, due to the similar frequency characteristics and larger amplitude of the former. For ADS, which relies upon the specific pairing of neurophysiological activity between two sites, non-specific stimulation due to biological noise sources would be obviously problematic.

Although many algorithms that are superior to monopolar voltage thresholds now exist and are easily implemented in various software packages for spike detection and sorting, the latency required in communicating with a host device can be prohibitive for ADS. Previously, ADS had been implemented in lightweight telemetric devices using an application-specific integrated circuit (Azin 2011a, b). However, for a long-term neurophysiological data acquisition solution, a more flexible architecture that can simultaneously acquire signals from hundreds of channels would be desirable. In addition, due to the timing constraints mentioned previously (<10 ms latency between detection and stimulation; <1 ms jitter in stimulus delivery), software solutions that involve a USB chain cannot be used. Therefore, the most tenable solutions need to be implemented algorithmically in hardware, such as through a field programmable gate array (FPGA), a PCIe card interfaced through an ethernet connection, or some other comparable digital signal processing unit.

Recently, the commercial availability of high-gain, high-resolution custom amplifier integrated circuits (Harrison 2007, Harrison and Charles 2003), which interface to a host device through a serial parallel interface (SPI) has made it possible to construct relatively inexpensive neurophysiological acquisition systems that scale to high numbers of recording channels. These systems, such as the acquisition system provided by Intan or the Open-Ephys acquisition board (Siegle 2017), use an FPGA to run the SPI that controls the amplifier chip while maintaining a buffer for USB communication with a host computer. Several proposed spike detection and spike sorting techniques take advantage of the FPGA, an integrated circuit that the end-user can reconfigure (Biffi 2010, Gibson 2013, Park 2017, Vallicelli 2017). Implementing the detection and sorting circuit on an FPGA allows the use of neurophysiological spiking as a reliable control signal in real-time, with low-latency; however, most implementations require custom integration with respect to the design of the full data acquisition circuit, which typically varies from laboratory to laboratory.

Here, we implemented a spike detection state machine designed to provide multiple threshold windows, reducing the likelihood of activity from sources other than spiking neural units on a single channel leading to the delivery of stimulation. The algorithm reduces the erroneous detection of spikes during biological noise in awake animals using an intuitive algorithm that requires minimal computational power. The implementation is conveniently designed to work as a modification to the existing open-source code provided by Intan for use in conjunction with their low-cost commercial platform for neurophysiological data acquisition and stimulus delivery. Importantly, the system allows the application of ADS with a fixed minimum latency <1 ms and has the potential to scale to a high number of channels in future design iterations.

2. Methods

2.1 Hardware architecture

The hardware architecture of the acquisition system and spike detector consists of three core components (Fig. 1):

1. Headstage: an amplifier circuit connected to a microelectrode array with an arbitrary number N of physical microelectrode leads placed near the neural substrate of interest;
2. FPGA: an interface that allows the amplifier circuit to multiplex both the incoming microelectrode signals and any outgoing stimulation commands to the appropriate microelectrodes; and,
3. Host: a general-purpose computer that provides an interface to the system, allowing the user to select the desired microelectrode channels and how a closed-loop stimulation scheme will be implemented.

This implementation used a commercially available integrated circuit and pre-assembled headstage (RHS2116; Intan Technologies, Los Angeles, CA, USA) to connect to the microelectrodes. To interface with this circuit, we used the Intan Stimulation/Recording Controller, which consists of an FPGA evaluation board (XEM6010-LX45; Opal Kelly Inc., Portland, OR, USA) equipped with a Xilinx Spartan 6 FPGA (XC6SLX45–2; Xilinx Inc., San Jose, CA, USA), a 128-Mbyte SDRAM chip, a 100-MHz clock source, I/O connectors, and a USB 2.0 interface chip capable of streaming data to a host computer at rates exceeding 20 Mbyte/s. A desktop personal computer (Z230; Hewlett-Packard, Palo Alto, CA, USA) running Windows 7 (Microsoft, Redmond, WA, USA) was used to control the USB chain.

Intan provides a hardware design that embeds the open-source USB/FPGA interface developed by Opal Kelly. This design makes it possible to read and modify registers of the RHS2116 from a host computer. It consists of verilog Hardware Description Language (HDL) code written for the XEM6010-LX45 evaluation board. This code is synthesized using the free Xilinx ISE WebPack software. The resulting bitfile is locally stored on the board in a dedicated Flash memory and can be updated through the USB interface. It is loaded on the Spartan-6 FPGA at each power-up, allowing the FPGA to interpret commands and parameters issued by the user from the USB chain.

At its core, the USB/FPGA design provided by Intan is a state machine that controls SPI buses on up to eight peripheral RHS2116 amplifier circuits. The interface also contains a module that implements a short-latency threshold comparator on up to eight channels of digitized amplifier data streams routed to 16-bit digital-to-analog converters (DAC; AD5662; Analog Devices, Norwood, MA, USA) mounted on the evaluation board. The comparator logic state is routed to a TTL output wire that corresponds to the DAC channel number. The DAC module also implements a single-pole high-pass filter (HPF) on the selected amplifier data stream.

A second module, also included in the existing Intan USB/FPGA interface, contains a state machine that controls the delivery of ICMS to a selected amplifier channel. The module can be configured through the GUI to deliver stimuli on the rising or falling edge of a TTL input signal. Thus, by physically connecting pairs of TTL inputs and outputs, “closed-loop” stimulation based on the detection of threshold-crossing events (in this case, extracellular action potentials, or spikes) is already possible using the USB/FPGA interface as provided by the vendor.

The main contribution described herein is the addition of a state machine for spike detection that offers improved artifact rejection, while taking advantage of the short-latency comparator in the DAC module of the existing USB/FPGA interface. Importantly, we sought to make as few changes as possible to the existing toolkit provided and validated by the commercial vendor, in the hopes that any changes we introduced could be more easily integrated to existing workflows. Overall, the changes amount to an increase of 408 flip flops compared to the originally synthesized architecture, well within the bounds of the available resources on the XEM6010-LX45.

2.2 Software interface

Software was modified from the original open-source C/C++ code provided by Intan Technologies for use with the RHS2116 amplifier IC, retaining many similarities with the original. The software implements a GUI, which provides a front-end to the USB/FPGA interface. Modifications described in the present study were added using Qt (version 5.8). Applications were compiled for Windows 32- and 64-bit operating systems using compilers for Microsoft Visual Studio 2015. This modified GUI includes a tab that allows configuration of the DAC (Fig. 2A, left panel) and the popup window for visualizing spikes is altered to accommodate online specification of each of the four parameters for each DAC channel used in the state machine detector, as described in Fig. 2A.

2.3 Spike detection state machine

The core of the spike detection state machine is a simple logic cycle that runs in the main module of the USB/FPGA interface (Fig. 2A, right). The state machine allows up to 8 threshold levels (L_i, where *i* is an integer from 1 to 8) with the following user-defined parameters (Fig. 2A, left):

- **Threshold,** a_i, refers to the voltage value (μV) that the signal must pass through to count as a crossing. If the threshold is negative, then a crossing occurs when the signal value is less-than or equal-to the threshold value (Fig. 2B, multiplex logic). If the threshold is positive, then a crossing occurs when the signal is greater-than or equal-to the threshold value. This number is an unsigned 16-bit integer, which is limited between −5,000 μV and +5,000 μV, based on the dynamic range and scaling of the amplifier and DAC.

- **Start,** b_i, refers to the (inclusive) onset sample of the window L_i. If the state machine counter is less than this value, the threshold conditions for the specified window will not be considered in the state machine logic. The state machine switches from idle to active (as defined below) once the filtered amplifier data stream routed to DAC channel *i* meets the criteria for L_i, if $b_i = 0$.

- **Stop,** c_i, refers to the (exclusive) end sample of the window L_i. If the state machine counter is equal or higher than this value, the threshold conditions for the specified window will not be considered in the state machine logic. The maximum stop value, c_{max}, defines the total duration of the spike detection state machine.

- **Type,** d_i, refers to the amplitude bounding for window L_i. It depends upon the polarity of the threshold. A value of zero corresponds to an “include” type window, which means that the signal must be less than a negative threshold or greater than a positive threshold while the state machine counter is within the range defined by the start and stop samples (Fig. 2C). A value of one corresponds to an “exclude” type window, which enforces the opposite conditions (signal must be greater than a negative threshold or less than a positive threshold).

- **Enable,** e_i, refers to whether window L_i is involved in the decision circuit for the state machine. The state machine can run with as few as 1 and as many as 8 windows enabled.

In the specific example of Figure 2A, we have defined three levels (e.g. L_1, L_2, and L_3), where a_1 and a_2 are the blue 'inclusion' thresholds ($d_1 = d_2 = 0$) and a_3 is the red exclusion threshold ($d_3 = 1$). Therefore, the dark-grey spike, which crosses threshold a_3, is excluded, but the black spike is not. Likewise, the light-grey spike, which does not cross the a_2 blue 'inclusion' threshold is also excluded. In total, the state machine runs for c_{max} samples, starting whenever the state is 'idle' and the filtered signal is less than a_1.

The state machine increments a counter on the rising edge of the sample clock depending upon its current state, which is always in one of these three conditions:

1. idle, when one or more of the level criteria is not met or no DAC channel is enabled (Fig. 2A, grey);
2. active, when the criteria for each enabled DAC with a start value less than or equal to the current sample index and a stop value greater than the current sample index channel is true (Fig. 2A, black); or,
3. trigger, when the counter equals the largest enabled DAC window stop value (Fig. 2A, orange).

The counter increments only when the state machine is in the active state, and resets to zero any time it enters the idle state (Fig. 2A; right). If the state machine reaches the trigger state, it returns to the idle state on the ensuing sample clock cycle. Each state of the machine is reported by the high state on a unique pair of TTL output and input wires (see Supplementary section S4 for details).

2.4 Surgical implant and recording for in vivo testing

All protocols for animal use were approved by the Kansas University Medical Center Institutional Animal Care and Use Committee in compliance with the Guide for the Care and Use of Laboratory Animals (Eighth Edition, The National Academies Press, 2011). Briefly adult male Long Evans rats were anesthetized using a combination of ketamine and xylazine as described previously (Nishibe 2010). A laminectomy was performed to minimize edema during the procedure. Five 00–80 stainless steel skull screws were fixed around the perimeter of the skull to improve attachment of the dental acrylic cap. Using stereotaxic coordinates, a craniectomy was made over sensorimotor cortex of the left hemisphere. Microwire arrays were positioned to span the rostral forelimb area (RFA), caudal forelimb area (CFA), and forelimb sensory cortex (S1), which was confirmed by a brief ICMS mapping procedure before insertion to a depth of approximately 1500 μm. An external silver wire on each array was tied to the same skull screw placed in the interparietal bone, which acted as a common ground. In the rat used for session *A* (recording sessions described below), the microwire array was a custom in-house design consisting of 32 channels of 33 μm diameter polyimide-coated tungsten wire (California Fine Wire Co., Grover Beach, CA), which were distributed throughout RFA, CFA, and S1 in a non-uniform grid pattern. The rat used for sessions *B* and *C* was implanted with a commercial microwire array (MicroProbes for Life Science, Gaithersburg, MD) consisting of 16 channels of nickel-chromium alloy 50 μm diameter wires arranged in a 4×4 grid with 250 μm site spacing implanted in S1. Qualitatively, spiking activity from both datasets was similar, but session *A* contained a few channels with large, stereotyped spikes, while spikes tended to be smaller in amplitude for sessions *B* and *C*. Prior to each recording, the rat was placed under anaesthesia via isofluorane induction, and subsequently one channel located within RFA was used for recording, while a single S1 channel was used in any stimulation sessions Electrode impedances ranged from 750 – 1,500 kΩ at recording sites. Recordings were made in 3- to 5-minute blocks during and after recovery from anaesthesia.

Recordings were made during three separate sessions. Recording sessions were assigned the codes '*A*,' '*B*,' and '*C*.' The main features and how these data were used within the current work are summarized in Table 1. Session *A* was taken from a first rat, three days after implantation, and contains a single epoch in which no stimulation was performed, which was used for subsequent offline characterizations due to the presence of large, stereotypical spike waveforms and low noise floor (RMS 18.6 μV, rectified median 11.3 μV). Sessions *B* and *C* were taken from a second rat approximately three months after the implantation. Session *B* tested the latency between spike detection using the state machine and onset of stimulation. Session *C* tested the online performance of the spike detection state machine using ad hoc parameters selected while the experiment was ongoing (e.g. to mimic a typical use case). Specific parameters for each recording session are reported in detail in Table S1; sub-indices indicate identical recording data that was re-run offline using a simulated test bench to characterize performance. To identify chewing periods (which bias performance toward false positive spike detection due to the presence of high-amplitude biological noise), a simultaneous video stream was synchronized with the neurophysiological data from session *C* through co-registration of a flashing LED that was tied to a digital input on the acquisition board.

2.5 Offline performance testing

Performance of the spike detection state machine was evaluated by comparing offline detection of spikes from the *in vivo* data from session *A*, either using a monopolar threshold detector or the state machine detector. To ensure that the analyses accurately captured online performance, we first validated the fidelity of the reconstructed recorded signals by ensuring that the DAC amplifier data stream and digital logic state streams recorded *in vivo* during session *C* matched those generated by the offline DAC filter and state machine simulation. Simulations were performed using test benches compiled in verilog, MATLAB (R2017a+), and Simulink (R2018b), as described in the supplementary methods section. The test benches are included in the online code repository along with the modified software and hardware code. Once we verified that there was no difference in the simulated digital logic state signals and the recorded ones, we used the DAC amplifier data stream recorded from session *A* to simulate the spikes detected using both a single-threshold detector (*A0*) as well as all events that entered the active and trigger states using the state machine detector (*A1*). For the monopolar threshold detector, spikes were only counted on the logical rising edge of the threshold crossing. Selection of a monopolar threshold was fixed at 40 μV, which was initially determined online by visual inspection of the spike scope to set a level that appeared qualitatively to reject noise while accepting most multi-unit spiking.

To characterize the ability of the spike detection state machine to reject artifact while still detecting viable spikes we calculated accuracy, defined as the ratio of the sum of correctly classified spikes (true positives; TP) and correctly classified artifacts (true negatives; TN) to the total number of spikes and artifacts detected. To determine whether spikes or artifacts detected during a simulation were correctly classified, a set of target classifications for spike and artifact waveforms were obtained offline using manual sorting to group similar waveforms. This consisted of a cluster cutting technique in which the spikes and artefactual waveforms were assigned iteratively through the manual selection of waveforms from the candidate set of waveforms detected as either spikes or artifacts by the detector, similar to the technique described in (Harris 2000). While this method of classifying multi-unit spike waveforms has limitations depending on the amplitude of units under consideration (Harris 2000), the purpose was to illustrate the ability of the spike detection state machine to reject artifactual waveforms, a situation for which an experienced operator is well-suited.

To verify our results on a dataset in which the ground truth spike times are already known, we synthesized an additional set of recordings (*C3*, in which a threshold detector was applied, and *C4,* in which the state machine detector was applied; parameters in table S1). In these simulations, known spike waveforms were added to a non-spiking recording channel at 1,500 uniformly sampled random samples throughout the duration of the sample record. It should be noted that in these simulations, identical recordings can yield slightly different numbers of total detected spike and artifact waveforms depending on which spike detection procedure was simulated even if the initial inclusion threshold is the same for the state machine detector and the threshold detector: because the state machine has a minimum duration that requires multiple samples in order to detect the spike, probabilistically there are more opportunities to identify candidate spike and artifact waveforms when using a single-threshold detector, potentially leading to a slightly higher number of total event classifications when using the monopolar threshold detector.

After either sorting the detected spike and artifact waveforms to obtain the target classifications or using the *a priori* known ground truth spike times as targets, performance was obtained using confusion matrices to compare the detected outputs (e.g. spikes or artifacts) against the target outputs (e.g. spike or artifact classifications of the detected outputs using offline sorting). Sensitivity (or true positive rate; TPR) was estimated as the ratio of correctly classified spikes to the sum of correctly classified spikes (true positives; TP) and outputs given as artifacts that were determined to be spikes by offline sorting (false negatives; FN). True negative rate (TNR) was estimated as the ratio of correctly classified artifacts (true negatives; TN) to the sum of correctly classified artifacts and outputs given as spikes that were determined to be artifacts by offline sorting (false positives; FP). Precision (positive predictive value) was estimated as the ratio of true positives to the sum of true positives and false positives. The false discovery rate (FDR) was estimated as the ratio of false positives to the sum of true positives and false positives. The false negative rate (FNR) was estimated as the ratio of false negatives to the sum of true positives and false negatives.

These last two metrics (FDR and FNR) were of special interest, as we aimed to reduce FDR while maintaining a low FNR.

3. Results

3.1 Ability to detect waveforms of interest

An important feature of the spike detector state machine is the ability to identify relatively low-amplitude spikes during epochs that contain periods of relatively high-amplitude biological artifact. Biological noise, such as arises from mechanical vibration and EMG that occur during chewing and whisking, leads to large-amplitude, high-frequency (>300 Hz) deflections in the signals observed on electrodes embedded within cortex. To illustrate this, we isolated a short exemplar epoch from recording session *C* in which the presence of chewing was verified by synchronizing the electrophysiological data stream with video of the rat moving freely in the recording chamber. While these epochs of activity are likely generated by biological sources, they may still be undesirable during motor recordings designed to study neurophysiological spiking of units related to other motor behavior (i.e. forelimb movement during pellet retrievals). Unfortunately, the simple threshold detector produces many false-positive spike detections during such epochs (Fig. 3A, red highlighting). By contrast, the state machine detector is still able to correctly detect spikes (Fig. 3B, blue highlighting) during the noisy periods without mistakenly triggering from the same waveforms that are problematic for the threshold detector (Fig. 3B, green highlighting). Even within a single recording session and on a single recording amplifier channel, it was possible to distinguish between substantially different spike waveforms by customizing the parameters sent to the spike detection state machine online. Parameters that were selected online (recording *C0*, table S1) captured the smaller multi-unit activity (Fig. 3C), whereas offline adjustment of parameters led to the ability to isolate waveforms from the larger of the two units (Figs. 3D). Importantly, the ability to set the level parameters in real-time, thanks to the modified GUI (Fig. S2), improved ease-of-use compared to existing systems, in which a “training” recording must first be obtained and analysed offline before allowing parameters to be set (Guggenmos 2013, Azin 2011a, b).

3.2 Performance in awake ambulatory rats

To quantify the online performance of the state machine we performed manual offline sorting of spike and artifact waveforms (from session *C*). We considered the offline sorting as ground truth, which allowed us to compute confusion matrices comparing the online classification (e.g. spike or artifact) to the offline sorted classification for the same waveform for each monopolar threshold crossing (Fig 4). The number of spikes correctly detected by the online spike detector state machine was 2,163 out of 2,582 (meaning a sensitivity, or true positive rate, of 83.8%). The number of true negative (i.e. artifacts not detected as spikes) was 40,188 out of 40,835 (meaning a specificity, or true negative rate, of 98.4%). The number of artifacts incorrectly classified as spikes was 647, resulting in a 23% false discovery rate (FDR) for the online spike detector state machine. Artifacts that led to false-positives contained qualitative similarities with the spikes of interest, which may account for this value (Fig. 4B, FP-1 and FP-2). Overall, the online accuracy of the spike detector state machine was 97.5% (Fig. 4A; recording *C0*), which is inflated by a high number of true negative samples due to the relatively large number of artifacts passed by the monopolar threshold. In practice, this could be mitigated using a monopolar threshold set to a much higher value; however, while increasing the threshold could reduce the number of artifacts falsely detected as spikes, it would also reduce the number of true positive spikes and is therefore not a feasible solution. Indeed, even the synthetic insertion of large-amplitude (−150μV peak) spikes at known times to a non-spiking channel results in an FDR of 90.5% for a monopolar threshold of −100 μV, while the state machine detector yielded an FDR of 29.3% and overall accuracy of 72.2% (Fig. S3).

Using a channel selected for its low noise floor and large-amplitude spike waveforms recorded *in vivo* (session *A*), we computed the same performance measures used in the previous case (Fig. 5A). Performance overall was comparable (97.1% accuracy) due to the large number of correctly rejected waveforms. However, careful parameter selection also yielded an improved FDR (6.9%) and FNR (2.8%) for the state machine spike detector under these ideal conditions. We compared the best-case performance of our state machine detector to a monopolar threshold detector. Using identical recordings, there is a dramatic improvement in the FDR when using the state machine detector (189 artifacts characterized as spikes, of a total 2,075 spikes detected online, Fig. 5B) compared to the monopolar threshold detector (2,770 artifacts characterized as spikes, of a total 7,075 detected spikes, Fig. 5B). This improvement results from the rejection of artifactual waveforms, such as occur during epochs of biological noise (e.g. chewing, Figs. 3A, 3B).

3.3 Mean latency from spike peak to stimulus delivery

The total latency for an activity-dependent stimulus can be considered as the sum of the algorithmic latency (to reliably detect an event) and the computational latency (due to the system). Algorithmic latency, in this case, depends on the maximum number of samples needed to detect a spike. In this work, spikes were detected using state machines that varied between 300μs (session *B;* 9 samples at 30 kHz sample frequency) and 800μs (session *C*; 24 samples at 30 kHz sample frequency). Therefore, the exact algorithmic latency is specific to the parameterization of the user-defined threshold levels. Our work did not alter the computational latency between the event detection and the delivery of the stimulus. During session *B*, the Intan Stimulation/Recording Controller stimulation sequencer module delay was set to zero milliseconds, allowing us to estimate the computational latency as the minimum latency between the rising edge of the virtual TTL input corresponding to the trigger state of the spike detection state machine and the onset of stimulus artifact. The computational latency obtained in this way was 167μs (5 samples at 30 kHz sample frequency; Fig. S1). Therefore, the total latency of the system during spike detection was reliably less than 1 ms, mainly due to the algorithmic latency, and indicates that the detector is responsive on a timescale that is both fast and reliable enough to be used for performing ADS.

4. Discussion

We developed a modified version of an open-source commercial system to implement closed-loop stimulation with sub-millisecond latency. The main improvement is the implementation of a spike detection state machine with an interface that allows the application of eight reconfigurable thresholds to any combination of different or identical amplifier channels. The implemented state machine slightly reduces sensitivity (i.e. true positive rate, Fig. 4A), but drastically improves specificity (i.e. reduced FDR; Figs. 5A, 5B), which may be critical in designing closed-loop electrical stimulation paradigms in the central nervous system *in vivo*. This improvement in selectivity is particularly important when the stimulation paradigm must be implemented during ongoing natural behavior, such as chewing or whisking (Fig. 3).

Although the focus of this study was on applying the improved detector for use in ADS, we envision that this type of low-latency, highly-selective discriminator could be useful in a range of closed-loop applications. For example, feedback needs not be delivered in the form of stimulation pulses but could instead be incorporated as a part of the experimental design itself, such as the delivery of a reward contingent upon the discrimination of a unique spike waveform (Koralek 2012). However, the context of developing closed-loop neuroprostheses for applications such as neurorehabilitation provides important constraints. For example, a critical aspect of the ADS paradigm is the timing of stimuli based on the detection of a stereotyped waveform that represents a small group of cells near the recording microelectrode (Guggenmos 2013). Therefore, it is desirable to minimize the FDR while maintaining a detection algorithm that can be implemented with low latency and customizable sensitivity to maximize the chances of invoking Hebbian mechanisms of plasticity between cells at the detection site and those at the stimulation electrode (Bi and Poo 1998). It is possible that such a stimulation regime could be augmented by incorporating multiple stimulation sites at offset latencies from a single trigger source; this is also possible using the system presented in this study. Similarly, although not tested here, the modifications presented can apply simultaneous thresholds to several spatially distributed sites simultaneously. This provides a practical way to mitigate the large, non-neural sources of noise that result from a failure in the common-mode rejection, which are typically present on multiple channels simultaneously. Future versions of the discriminator presented here that scale to an arbitrarily large number of thresholds could then be useful in sorting using tetrodes or other high-density arrays.

With the rising interest in applications of closed-loop technologies for stimulation of the central nervous system (Levi 2018), a number of methods for implementing closed-loop stimulation have been made openly available. These include software packages, such as Falcon (Ciliberti and Kloosterman 2017), the Open Ephys GUI (Siegle 2017), and NeuroRighter (Newman 2012); however, software implementations of online spike detection and triggered stimulation typically suffer from the latencies imposed when performing serial communication with the host computer. One exception is the Real-Time eXperiment Interface (RTXI, (Patel 2017)); however, because the system is designed to operate using a National Instruments Data Acquisition card (NI-DAQ), it may be difficult to scale to a very high channel architecture. On the other hand, the ADC of the RHS2116 is scalable, and because digitization occurs very close to the source (on the headstage), yields improved noise characteristics.

Hardware implementations, such as the synthesized bitfile that can be readily uploaded to effectively transform an FPGA into a commercial neurophysiological acquisition system, are not as widely distributed. Because hardware implementations typically have very specific design constraints and are optimized to meet those constraints, it is impractical to develop and distribute open-source hardware for closed-loop neuroprosthetics. Just as the RTXI system is not readily compatible with the Intan amplifier chips, hardware implementations (Ambroise 2017, Buccelli 2019) are designed to interface with *in vitro* microelectrode arrays that interface to an FPGA with different input and output pin configurations, making it difficult to provide a ubiquitous hardware bitfile for every experimental setup. Alternatively, moving from a hardware implementation in an FPGA to a custom application specific IC (and subsequent commercialization) becomes more practical for individual applications.

To minimize changes to the existing open-source software provided by Intan, the spike detection state machine was implemented in the DAC module. This imposes the limitation that only one spike detection state machine can run at a time. At a maximum, up to eight different amplifier channels could be polled for synchronous or near-synchronous events, or eight threshold criteria could be applied to the waveform of a single amplifier channel. Making substantial modifications to the existing FPGA might allow scaling of a spike detection state machine module to any arbitrary number of thresholds on different channels. A natural extension of this work would be to scale up the number of trigger sources for multi-stream ADS, particularly as FPGA evaluation boards with increased on-board resources become available. Generalizing the state machine to a higher number of independent channels by running it as a module that is independent of the DAC, automating the process of setting threshold levels (e.g. using spike “templates”), and integrating independent state machines to allow concurrent detection of events in multiple frequency ranges are currently being investigated to improve their application in closed-loop neuroprosthetic interfaces. Automating the process of setting threshold levels, especially as channel counts scale up, will be important, as the improved rejection may also reduce sensitivity, depending upon the ad hoc parameters set by the operator. Algorithmically, the state machine is fundamentally similar to the one implemented in (Azin 2011b); however, the state machine described here allows more flexibility in the parameterization of each threshold level by allowing the end-user to set each of the following four parameters while the application is running.

We have provided modifications to an existing interface for conducting electrophysiological experiments using closed-loop stimulation. These improvements allow spike detection to be performed with a higher selectivity at the expense of a reduced sensitivity. This trade-off is dependent upon the ad hoc selection of parameters, which can be adjusted by the experimenter in real-time. The architecture in which this improved spike detection state machine is implemented has the possibility to scale to a very high number of channels in the future, improving current and future functionality. Our contribution to the original design improves the accessibility of investigating of closed-loop stimulation paradigms, which may be necessary for effective, therapeutic neuroprosthetic systems.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgment

This study was made possible through the support of NIH grants R01NS030853 (RN), R03HD094608 (DG), T32HD057850 (MM, RN), and F32NS100339 (DB). Collaboration was made possible through the MAECI award (MC). The authors have no competing interests to disclose.

References

[1]. Guggenmos DJ, Azin M, Barbay S, Mahnken JD, Dunham C, Mohseni P, and Nudo RJ 2013 Restoration of function after brain damage using a neural prosthesis. P. Natl. Acad. Sci. USA, 110, 52, 21177–21182

[2]. Azin M, Guggenmos DJ, Barbay S, Nudo RJ, and Mohseni P 2011a A miniaturized system for spike-triggered intracortical microstimulation in an ambulatory rat. IEEE T. Bio-med. Eng, 58, 9, 2589–2597

[3]. Azin M, Guggenmos DJ, Barbay S, Nudo RJ, and Mohseni P 2011b A battery-powered activity-dependent intracortical microstimulation IC for brain-machine-brain interface. IEEE J. Solid-st. Circ, 46, 4, 731–745

[4]. Jackson A, Mavoori J, and Fetz EE 2006 Long-term motor cortex plasticity induced by an electronic neural implant. Nature, 444, 7115, 56–60 [PubMed: 17057705]

[5]. Bi G, and Poo M 1998 Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J. Neurosci, 18, 24, 10464–10472 [PubMed: 9852584]

[6]. Cheney PD, and Fetz EE 1985 Comparable patterns of muscle facilitation evoked by individual corticomotoneuronal (CM) cells and by single intracortical microstimuli in primates: evidence for functional groups of CM cells. J. Neurophysiol, 53, 3, 786–804 [PubMed: 2984354]

[7]. Harrison RR: 'A Versatile Integrated Circuit for the Acquisition of Biopotentials'. Proc. IEEE Custom Integrated Circuits Conference, San Jose, CA, 16–19 September 2007 2007 pp. 115–122

[8]. Harrison RR, and Charles C 2003 A low-power low-noise CMOS amplifier for neural recording applications. IEEE J. Solid-st. Circ, 38, 6, 958–965

[9]. Siegle JH, López AC, Patel YA, Abramov K, Ohayon S, and Voigts J 2017 Open Ephys: an open-source, plugin-based platform for multichannel electrophysiology. J. Neural Eng, 14, 4, 045003 [PubMed: 28169219]

[10]. Biffi E, Ghezzi D, Pedrocchi A, and Ferrigno G 2010 Development and Validation of a Spike Detection and Classification Algorithm Aimed at Implementation on Hardware Devices. Comput. Intel. Neurosc, 2010, 15

[11]. Gibson S, Judy JW, and Markovic D 2013 An FPGA-based platform for accelerated offline spike sorting. J. Neurosci. Meth, 215, 1, 1–11

[12]. Park J, Kim G, and Jung S 2017 A 128-Channel FPGA-Based Real-Time Spike-Sorting Bidirectional Closed-Loop Neural Interface System. IEEE T. Neur. Sys. Reh, 25, 12, 2227–2238

[13]. Vallicelli EA, De Matteis M, Baschirotto A, Rescati M, Reato M, Maschietto M, Vassanelli S, Guarrera D, Collazuol G, and Zeiter R: 'Neural spikes digital detector/sorting on FPGA'. Proc. IEEE BiomedicaCircuits and Systems Conference, Turin, Italy, 19–21 Oct. 2017 2017 pp. 1–4

[14]. Nishibe M, Barbay S, Guggenmos D, and Nudo RJ 2010 Reorganization of motor cortex after controlled cortical impact in rats and implications for functional recovery. J. Neurotraum, 27, 12, 2221–2232

[15]. Harris KD, Henze DA, Csicsvari J, Hirase H, and Buzsáki G 2000 Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J. Neurophysiol, 84, 1, 401–414 [PubMed: 10899214]

[16]. Koralek AC, Jin X, Long Ii JD, Costa RM, and Carmena JM 2012 Corticostriatal plasticity is necessary for learning intentional neuroprosthetic skills. Nature, 483, 7389, 331–335 [PubMed: 22388818]

[17]. Levi T, Bonifazi P, Massobrio P, and Chiappalone M 2018 Editorial: Closed-Loop Systems for Next-Generation Neuroprostheses. Front. Neurosci, 12, 26 [PubMed: 29483859]

[18]. Ciliberti D, and Kloosterman F 2017 Falcon: a highly flexible open-source software for closed-loop neuroscience. J. Neural Eng, 14, 4, 045004 [PubMed: 28548044]

[19]. Newman JP, Zeller-Townson R, Fong M-F, Arcot Desai S, Gross RE, and Potter SM 2012 Closed-Loop, Multichannel Experimentation Using the Open-Source NeuroRighter Electrophysiology Platform. Front. Neural Circuits, 6, 98 [PubMed: 23346047]

[20]. Patel YA, George A, Dorval AD, White JA, Christini DJ, and Butera RJ 2017 Hard real-time closed-loop electrophysiology with the Real-Time eXperiment Interface (RTXI). Plos. Comput. Biol, 13, 5, e1005430 [PubMed: 28557998]

[21]. Ambroise M, Buccelli S, Grassia F, Pirog A, Bornat Y, Chiappalone M, and Levi T 2017 Biomimetic neural network for modifying biological dynamics during hybrid experiments. Artificial Life and Robotics, 22, 3, 398–403

[22]. Buccelli S, Bornat Y, Colombi I, Ambroise M, Martines L, Pasquale V, Bisio M, Tessadori J, Nowak P, Grassia F, Averna A, Tedesco M, Bonifazi P, Difato F, Massobrio P, Levi T, and Chiappalone M 2019 A neuroprosthetic system to restore neuronal communication in modular networks. bioRxiv, 514836

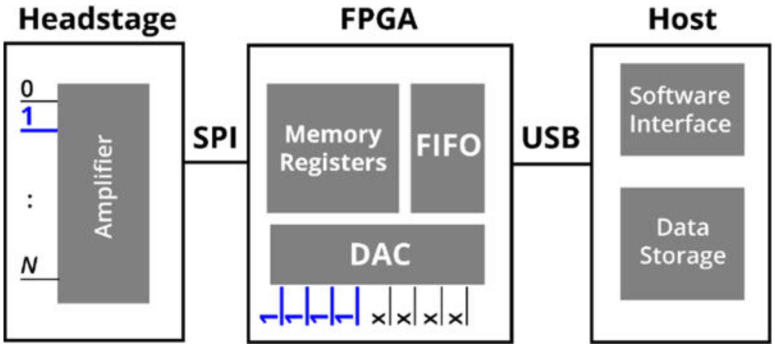

Figure 1: Overview of system architecture and implementation.

An amplifier chip is interfaced to the field-programmable gate array (FPGA), via a serial-parallel interface (SPI). *N* electrode channels are routed to a high-gain amplifier. On board the FPGA, amplifier data from the FIFO buffer are piped to the host device via a USB interface. The digitized signals from any selected combination of amplifier channels (blue) can also be routed to up to 8 digital-to-analog converter (DAC) channels, where threshold comparator logic can be applied with sub-millisecond latency. In this example, 4 threshold windows are applied to the filtered data stream from amplifier channel 1.

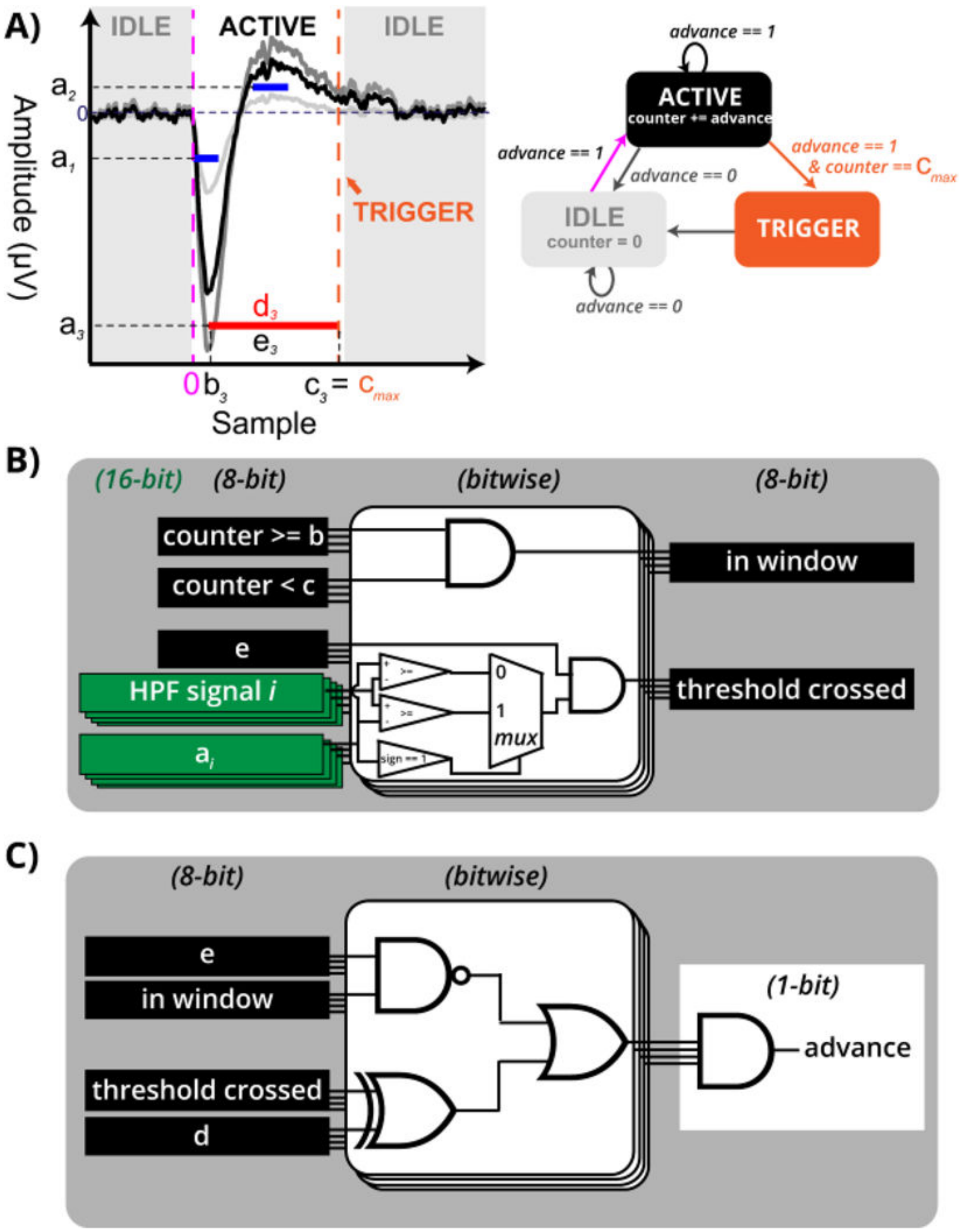

Figure 2: Spike detection state machine implementation.

A) *Left:* Example of a spike that would be included (black) and waveforms that would be rejected (grey) by the three state machine levels depicted (L_1, L_2, and L_3, denoted by corresponding thresholds a_1, a_2, and a_3). The dark-grey waveform exceeds the red exclusion threshold (a_3), while the light-grey waveform does not meet the second blue inclusion threshold (a_2). The black spike is included because the absolute value of its negative component does not exceed the absolute value set by a_3, while the absolute value of its positive component exceeds the level set by the second blue inclusion level a_2. The parameters (a-e) are defined by the user during acquisition and are illustrated for the red exclusion level shown. *Right:* state flow diagram for the spike detection state machine. By default, the detector is in the idle state (grey), but transitions to active (black) as soon as the data stream fulfils the parameters for the earliest window (magenta). If the waveform meets all criteria specified by the defined levels, the state switches to trigger (orange), then automatically reverts to idle. **B)** Threshold logic in the DAC module. For each of the 8 DAC channels, the corresponding parameters determine if the machine is within the start and stop points of the window, relative to when the counter started, as well as whether it crossed the threshold (depending on threshold polarity). **C)** Active and idle counter incrementing logic. If the data stream meets criteria of each enabled level that applies to the current counter value, the counter is advanced by 1.

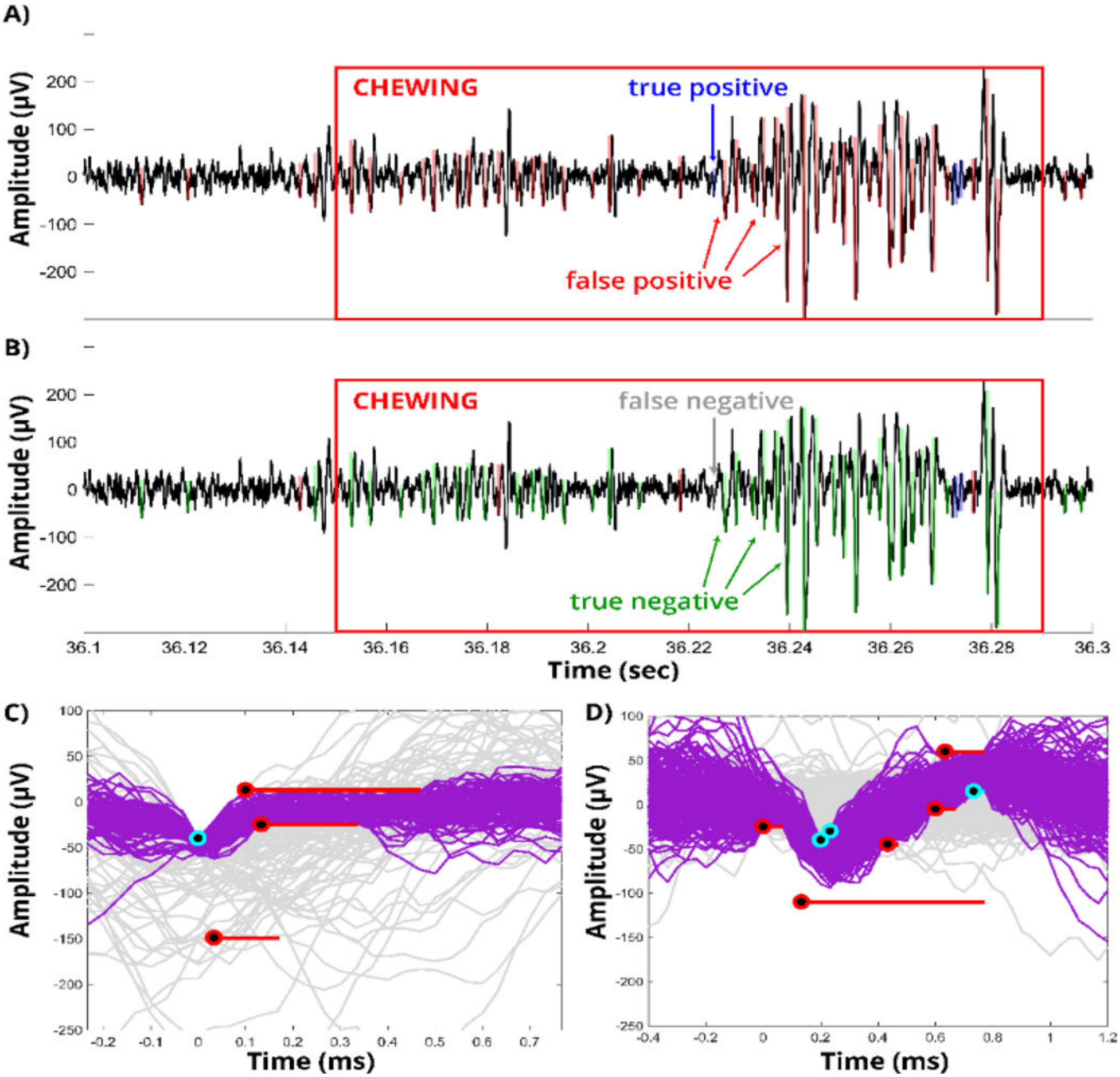

Figure 3: Qualitative performance of the implemented spike detection.

A) 200ms of high-pass filtered data from session *C* during single-threshold (−40μV; simulation *C2*) spike detection. Red box represents a 130ms epoch of chewing. Red highlighting (false positive) indicates spikes that were wrongly detected. Blue highlighting (true positive) shows spikes that were correctly identified. **B)** Same data as in panel A with superimposed detections from the state machine spike detector. Green highlighting (true negative) indicates artifacts that were correctly rejected by the state machine. Grey highlighting (false negative) indicates a case of true spike not detected by the state machine. **C)** Random sub-sampling of 250 detected (magenta) and 250 rejected (grey) waveforms using the spike detection state machine in real-time (recording *C0*), using the digital outputs from the online state machine. Flat lines represent threshold levels. Black spots represent inclusive samples that must meet the threshold criteria, while ends of lines are open to represent the non-inclusive threshold criterion. Cyan thresholds must be exceeded, whereas red thresholds must not be exceeded. **D)** An offline reconstruction (recording *C1*) was used to simulate the state machine using different window parameters. This random sub-sampling of 250 detected and 250 rejected waveforms indicates how the parameters could be set differently to isolate spikes from a different unit. Note that increasing the duration of the state machine also increases the total time to detection.

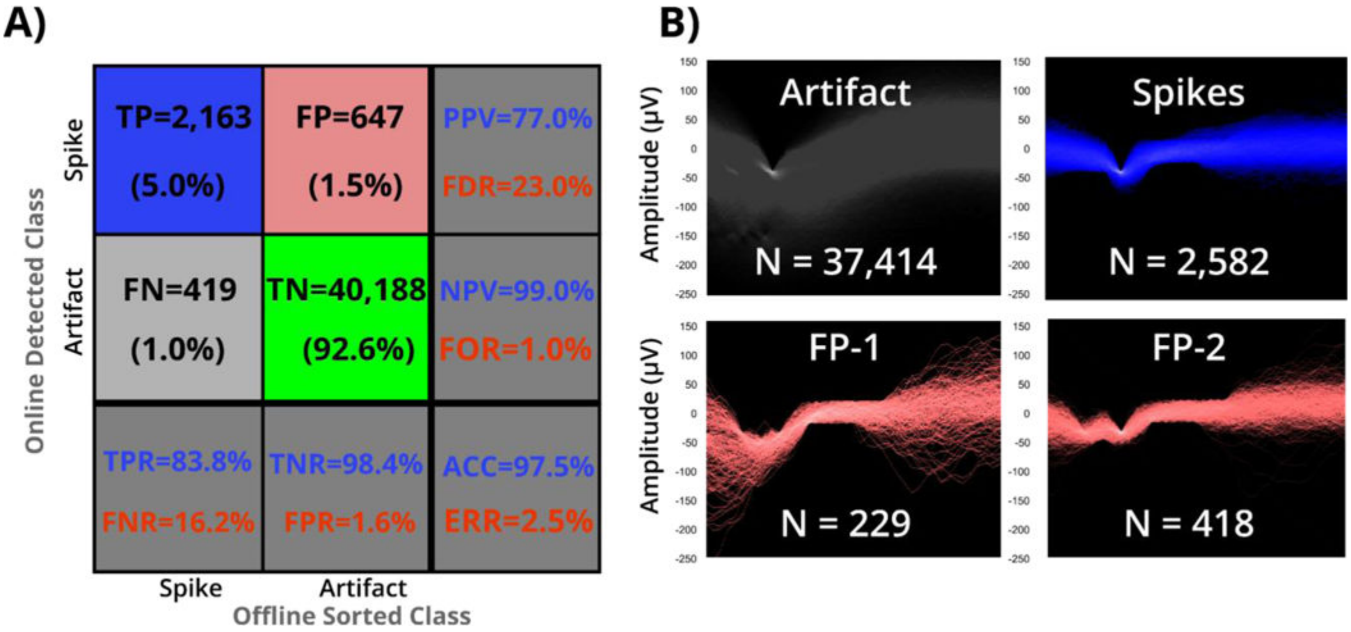

Figure 4: Typical performance compared to offline sorted spikes.

A) Confusion matrix for comparison of online spike detection state machine performance after manual offline sorting of spike and artifact waveforms (recording *C0*). The blue box contains the number of true positive spikes and the percentage of the overall detected events that fit this category. The salmon box contains the number of false positive spikes detected by the algorithm, as determined by manual sorting. The grey box indicates the number of rejected spikes (which entered but did not complete the state machine) that were scored offline as spikes. The green box represents waveforms that were rejected by the state machine and were also classified manually offline as artifact. The top row on the far-right column show the positive predictive value (PPR) in blue and false discovery rate (FDR) in red. The second row on the far-right column shows the negative predictive value (NPV) in blue, and the false omission rate (FOR) in red. The first column on the bottom of the matrix show the sensitivity (or true positive rate, TPR) in blue and the false negative rate (FNR) in red. The second column on the bottom of the matrix shows the true negative rate (TNR) in blue and the false positive rate (FPR) in red. The box in the bottom right of the plot shows the overall accuracy (ACC) in blue and its complement (the error percentage, ERR) in red. **B)** Offline sorting used for comparison. Lighter regions indicate a higher density of waveforms passing through those voltage values. Spikes were manually sorted using cluster cutting to separate units into characteristic waveforms. Magenta outline indicates spike profile used for offline sorting in panel A. Bottom two panels (FP-1 and FP-2) are characteristic waveform types that sometimes passed the state machine conditions, contributing to the number of false positives.

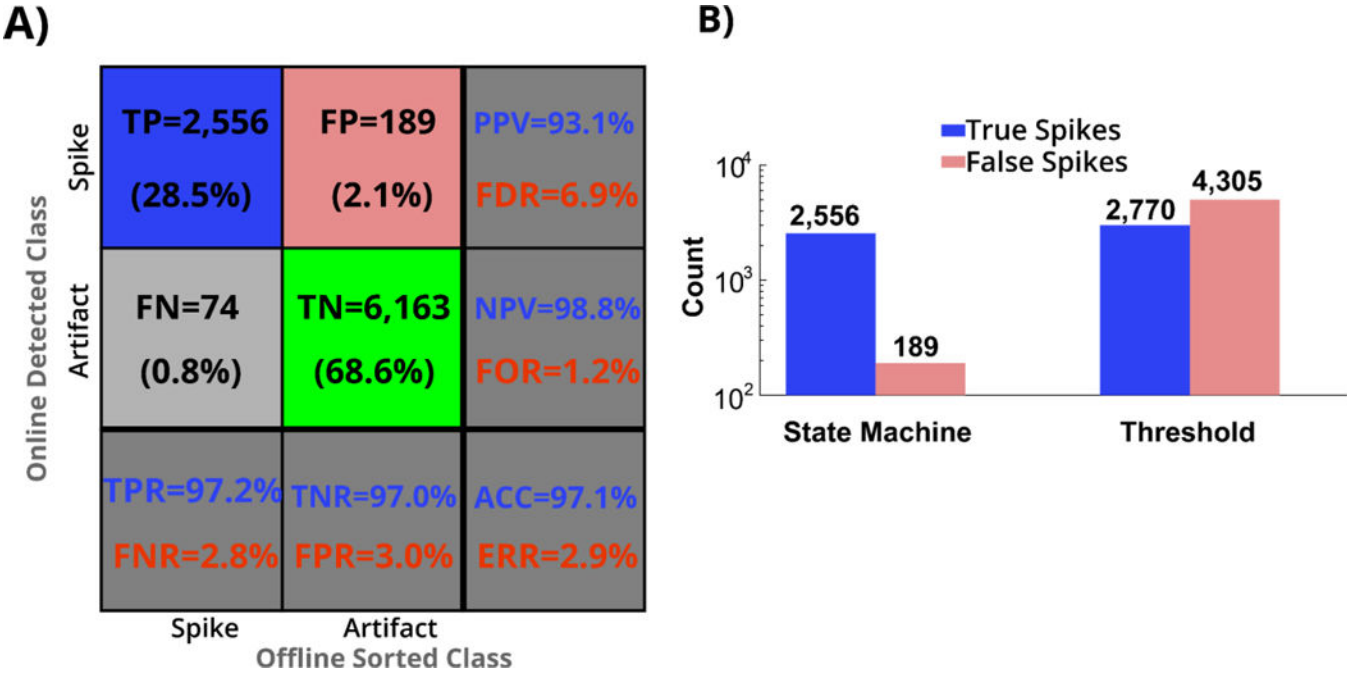

Figure 5: Ideal performance compared to offline sorted spikes and monopolar threshold detection.

A) Simulated performance using an ideal *in vivo* recording with large spikes (recording *A1*). Although the simulated performance is applied to a channel with high-amplitude spike waveforms, the overall accuracy effectively remains consistent. This is due to the relatively large proportion of waveforms that are correctly rejected (middle box). **B)** Manual offline sorting performed for recording *A1* (presented in panel A), as well as a comparison to performance of true (blue) to false (salmon) discoveries for the state machine detector and a simple threshold detector for the same dataset (recording *A0*).

Table 1:
Summary of recording data sets taken from rats.

Recordings were taken from awake, ambulatory rats implanted in RFA and S1. Columns describe whether stimulation was used, the main feature that distinguishes that recording dataset from the others, and the reason the recording was used in this study.

Name	Stim?	Feature	Use
A	No	Large stereotypical spikes; low noise	Offline performance
B	Yes	Stimulus artifacts	Test latency of stimulation
C	No	Typical use case; synchronized video	Online performance