

# Chapter 5

## Discussion

The research presented in this dissertation was motivated by the long term goal of building a fully autonomous Brain Machine Interface. Such a system would allow an amputated or paralyzed patient to use his brain to control a prosthetic limb, which would in turn return real-time feedback to the sensory cortex. Such devices need not be limited to prostheses of arms and legs; real-time feedback modulated control of wheelchairs, home appliances, and possibly even voice emulators may be of therapeutic value to millions of patients. The keys to advancing BMI technology will be threefold: exploiting the plasticity of the brain, increasing the basic understanding of how the brain processes information, and increasing the number of individual neurons that can be monitored simultaneously. The literature has shown that while it may be possible to build BMIs that monitor fewer than 100 neurons, the resulting prostheses may be limited in dexterity and accuracy.

One possible configuration for a fully implantable BMI would consist of a network of subdural integrated circuits, each monitoring its own bundle of cortical electrodes. The goal of each IC would be to minimize the amount of information that it must wirelessly transmit to an external receiver. For example, the IC may periodically send a packet containing the number of action potential events occurring on each channel.

This would require autonomous spike detection, which may be achieved in either the analog or digital domain. A more elaborate system would sort the AP spikes and transmit counts for each neuron, instead of for each channel; this approach would require a digital design.

The network of implanted ICs would continuously transmit spike times to a host processor which would in turn generate the control signals for the prosthesis. Such a processor could be embedded inside the prosthesis itself, obviating the need for any externally mounted batteries or electronic components. The processor might use a linear model or a neural network to generate the control signals, with model parameters being set during an initial training period.

A number of technical obstacles prevent such a system from becoming a reality. The work presented in this dissertation has attempted to engineer solutions to some of these problems while merely uncovering other ones. The hardware designs and software algorithms developed here have served as testbeds for the novel design approaches required to build a fully implantable, fully autonomous BMI.

## 5.1 Headstage

The headstage IC was designed to investigate the issues of noise performance, gain and filter matching, and buffering that will be required for the first stage of an implantable analog front end. The application of IC technology in this field is not new [27, 82, 45, 83, 84, 85]. However, integrated circuits have not yet been used to improve gain and filter cutoff matching in multichannel devices. It will be difficult to use differential amplification for rejecting common mode background signals without such circuits. The headstage IC designed in this work successfully used two matching feedback resistors to achieve tight control over gain and filter cutoff frequencies. Although precise resistor values are hard to attain in silicon, the centroid layout

technique was successfully used to achieve highly accurate resistor ratios. The gain of two was chosen as a good balance between the importance of applying gain in the first stage (to maximize SNR) and the requirement that the amplified electrode offset voltage not saturate the amplifiers.

### 5.1.1 Headstage Noise

Although the headstage IC was successfully used to record *in vivo* cortical signals, its input referred noise ( $10\mu V_{rms}$ ) is too large for small amplitude action potentials. The combination of electrode and biological noise sources in the typical neural recording produces a background noise floor of approximately  $20\mu V_{rms}$  [24]. The headstage circuit noise is then added to this background noise according to the equation:

$$\text{total noise} = \sqrt{\text{background noise}^2 + \text{circuit noise}^2} \quad (5.1)$$

Using this equation, the contribution from the Headstage circuit brings the total noise to  $\sqrt{20\mu V_{rms}^2 + 10\mu V_{rms}^2} = 22.4\mu V_{rms}$ . In general, increasing the noise from  $noise_1$  to  $noise_2$  reduces the SNR by a factor of:

$$\frac{\frac{\text{signal}}{\text{noise}_2} - \frac{\text{signal}}{\text{noise}_1}}{\frac{\text{signal}}{\text{noise}_1}} \quad (5.2)$$

$$= \frac{\text{noise}_1}{\text{noise}_2} - 1 \quad (5.3)$$

Therefore, by increasing the noise floor from  $20\mu V_{rms}$  to  $22.4\mu V_{rms}$ , the Headstage IC reduces the SNR by a factor of 10.7%. By comparison, the Plexon headstage that was compared against the Duke Headstage measured only  $3.5\mu V_{rms}$  of input referred noise. This increases the total noise to  $20.3\mu V_{rms}$  (Equation 5.1), and hence reduces the SNR by only 1.4%. Based on these and other observations [86] future headstage designs should not have an input referred noise exceeding  $5\mu V_{rms}$ , which corresponds

to an SNR reduction of  $\sim 3\%$ . Note that in the Headstage IC, the thermal noise contributions from the feedback resistors did not contribute significantly to the noise floor. Hence, to improve the noise performance in future designs, it will be necessary to rely on lower noise IC processes and op-amp architectures.

### 5.1.2 Headstage Packaging

Device packaging has emerged as one of the unforeseen obstacles to creating an implantable custom silicon circuit. Tightly packed recording electrodes will require high density analog processing integrated circuits. However, any physical reductions afforded by using custom ICs is negated by circuit packaging. Even the smallest IC package can increase the circuit area by a factor of two or three. One promising solution is to use flip-chip bonding to attach raw die directly to PC boards, eliminating the need for pins or even bonding wires. This technology is already being studied in the Wolf lab, but will likely increase the cost and complexity of mass producing neural recording hardware.

### 5.1.3 Electrode Offset Rejection

A major challenge when designing fully integrated analog ICs for neural processing is the need to measure microvolt level signals superimposed on large low frequency offset voltages. These offsets occur as a result of half cell potentials at the electrode-electrolyte interface and vary slowly in time with small movements of the electrode that disrupt the charge bilayer equilibrium. High pass filters can eliminate this offset, but large chip areas may be required for the necessary combinations of resistors and capacitors. Using discrete off-chip capacitors would require at least one capacitor I/O per channel [86], meaning that IC footprint sizes would be pad-determined as chips were scaled up in numbers of channels. Another solution is to combine each

electrode's capacitance with the buffer's input impedance to create an *ad hoc* high pass filter [82, 85]. However, this technique may require that the headstage amplifiers be matched to a particular type of electrode, or that trimmable devices be used to control input impedance [47]. Lowering the input impedance may also decrease the accuracy of the high-pass filter poles, thus reducing the CMRR of any differential amplification that follows [83]. Switching techniques such as switched capacitor filters and chopper modulated filters may add too much noise to the unamplified microvolt level neural signals. It may be possible to apply a modest gain to the unfiltered electrode signals and prevent saturation by increasing the power supply rails. However, this will increase power consumption which will be undesirable in an implanted system with limited power. The most practical solution will depend on the application, and will likely be a combination of these techniques. Designs should limit the number of off chip passive devices and should not be dependent on electrodes and circuits being specifically matched to one another.

## 5.2 Analog Front End

The analog front end was built as a prototype of an eventual fully integrated analog processing IC. However, the design will also be useful for research protocols requiring a low power multichannel circuit for processing and digitizing extracellular neural signals. The AFE requires no special in-house facilities to be duplicated. All components are available off the shelf and may be hand soldered. The circuit board design, which has been made available to the public, can be fabricated inexpensively at any number of board houses.

The small amplitude extracellular signals typically encountered in neural recordings necessitated a low-noise approach. Since noise performance is always limited by the first stage, the preamplifier was designed to be the quietest and to have the

highest gain. The preamplifier noise was limited by combining a low-noise, high power op-amp with a low-pass filter. The non-inverting architecture prevents loading of the the high-pass filter. While lower noise amplifiers are available, they generally require more power; the preamplifier design in Chapter 3 is a trade-off between power consumption and noise.

The differential stage was placed immediately after the preamplifier to (1) apply more gain in an early stage and (2) improve signal quality by allowing for the rejection of common-mode background signals. It was shown that reference selection can be instrumental in attenuating common mode signal artifacts such as those generated by motion or chewing. The active filter in the differential amplifier's feedback pathway improved the common mode range by adding a high-pass pole to attenuate low frequency offset voltages.

To minimize waveform distortion and preserve signal fidelity, the AFE was designed with Bessel filters. Since Bessel filters minimize phase distortion at the expense of sharp filter roll-offs, higher order filters are necessary. The system has a total of four high-pass and five low-pass poles. The measured phase response of the AFE was approximately linear ( $r^2 = 0.98$ ) making the group delay  $\left(-\frac{d\phi}{df}\right)$  nearly constant in the passband. The maximum group delay at any frequency is  $\sim 2msec$ .

The variable gain stage was placed after the three Sallen-Key filters in order to drive the capacitive input load of the 16:1 time-division multiplexer. The system's overall resolution varied from  $18.6nV/bit$  at maximum gain and resolution to  $4.9\mu V/bit$  at minimum gain and resolution.

Although a precision differential amplifier was used with a CMRR of  $110dB$ , it was impossible in practice to measure CMRRs greater than  $\sim 42dB$ . This can be accounted for by the device tolerances in the preamplifier stage; using  $\pm 0.1\%$  resistors and  $\pm 5\%$  capacitors in the preamplifier, the expected worst case CMRR at  $1kHz$  is

39.2dB. To verify these calculations, the preamplifier was rebuilt using  $\pm 1\%$  resistors and  $\pm 10\%$  capacitors. In this case, the CMRR was measured to be 33.3dB at 1kHz, which is close to the predicted worst case CMRR of 25.4dB. The 8:1 multiplexers providing the reference signals for the differential amplifiers were found to have a negligible effect on the CMRR, even when a single multiplexer was driving all 16 reference lines.

In order to facilitate a portable, battery powered product, an emphasis was placed on minimizing power consumption. Low power op-amps with shutdown capability were used where possible. The predicted power consumption for each active channel is 7.3mW. The analog to digital converter (AD7495) and AC ground buffer (OP262), which are always on, consume a combined 10.2mW. The two low dropout voltage regulators together only use 0.38mW.

The predicted input referred noise for the preamplifier stage is  $0.7\mu V_{rms}$ , which includes the contributions of the opamp (Maxim 4253) and the passive feedback components. The prediction increases by less than 0.5% when the contribution of differential amplifier stage is added. The actual measured input referred noise ( $1.0\mu V_{rms}$ ) reflects a sum of the circuit noise and the ambient electromagnetic noise. This increase over the predicted noise value was measured in spite of the grounded metal shields. The noise sources reduce the overall system resolution; although the analog to digital converter produces 12 bit samples, only 9.4 bits are more significant than the noise floor at the minimum gain and 6.3 bits at the maximum gain.

### 5.3 Variable Gain

The variable gain was included on the AFE to improve dynamic range and to limit the number of ADC resolution bits. Limited numbers of bits means reduced bandwidth per channel, which in turn translates into more channels that may be passed over

the same wireless link. In practice, the variable gain was an indispensable tool for acquiring single unit signals, although the process of determining the gain for each channel was cumbersome and time consuming. As the number of channels in the BMI scales, it will become unreasonable to set each gain manually. A superior alternative would be to increase the ADC resolution by four bits (effectively a gain of  $2^4 = 16$ ) and then apply an automatic range selection algorithm, such as the one in the spike detector (see Section 3.4.1) to select the best eight bits. This will produce an autonomous system without increasing the amount of data that must be transmitted per channel.

## 5.4 Reference Channel Selection

The reference selection matrix was shown to be instrumental in eliminating common mode noise sources. Although this has proven to be a useful feature, it requires the operator to determine which electrode should be the reference. As with the variable gain, features requiring user input will not scale well as implantable BMI ICs move from dozens of channels to hundreds. For an autonomous real-time BMI, it will likely be more feasible to use unipolar recordings and to deal with the effects of motion artifacts at the software level. One alternative would be to group all electrodes into pairs and make bipolar recordings between each pair (or equivalently to use stereotrodes). In theory, this would reduce common mode noise and increase the number of electrodes per channel, although it would also increase the probability of overlapping action potentials, which in turn make spike sorting and spike detection less accurate. Such a hardware configuration has not been tested. Although reference selection may not be ideal for large BMI systems, it is clearly a useful tool that will be feasible for use in certain applications with low channel counts.

## 5.5 Digital Back End

The main purpose of the digital back end was to move digitized samples from the AFE into the wearable PC and out over the wireless Ethernet. The present design is versatile and may easily be modified in both hardware and software to adapt to changed design requirements and technology upgrades. Such flexibility ensures that not only will the Backpack be a useful tool at Duke, but also that it may be modified to meet the particular requirements of other research labs engaged in neural recording research. Software changes in the FPGA may be used to alter the sampling rate, the channel selection process, and the choice of which bits get transmitted for each sample. The latency may also be adjusted by altering the number of points stored in the FIFO. Hardware upgrades might include battery replacements to tailor to specific size/weight and lifespan requirements. Both the wearable PC and the wireless link may be replaced as newer, higher speed and lower power technology becomes available. For example, the wearable PC is currently being upgraded to a faster Linux-based PC. Instead of porting in data through the parallel port with programmed I/O, the upgraded system will use DMA-modulated USB data transfers. This will reduce the data bottleneck described in Section 4.1.1 and ultimately lead to improved data rates. The wireless card may be upgraded to the 802.11g standard, potentially increasing the wireless data rate by a factor of five. The Backpack's modular design makes such upgrades possible without redesigning the overall data flow.

## 5.6 Digital Design

The decision to work with digitized data was motivated by a number of factors. The processing algorithms that must eventually be incorporated into an autonomous BMI (i.e. spike detection and spike sorting) are simpler to implement with programmable

logic devices and DSPs rather than with analog logic components. Furthermore, a digital design is more flexible, allowing for different software-selectable modes. Depending on the user's preference, a digital system can transmit raw data, spike waveforms, or sorted spike counts of any bin size. An analog system would be confined to only one mode, and reconfiguring that mode would require major hardware level changes.

Digital telemetry is desirable because it provides immunity to channel noise; error correction and packet retransmission can guarantee that the signal received at the host PC matches the transmitted information byte for byte. Digital telemetry may also simplify the procedure of increasing the number of transmitted channels. Adding new channels is equivalent to simply increasing the data rate, which is transparent to the rest of the system as long as the revised data rate does not exceed the telemetry system's capacity. By comparison, adding new channels to an analog telemetry system equates to either time or frequency multiplexing the data, or to modulating at new carrier frequencies. These changes must be accounted for with appropriate hardware level changes at the receiver. Digital telemetry hardware may be replaced with standalone modules such as Bluetooth or Ultrawideband. Using standalone modules would eliminate the need for the wearable PC, thus reducing the size, weight, and power consumption of the entire Backpack.

## 5.7 Latency

BMI data acquisition hardware must be built to minimize data latency, since delays between cognitive events and the prosthesis's response can adversely affect performance [3]. The majority of the Backpack latency is in the buffer times in the FIFO and the wearable PC. These latencies varied depending on the number of active channels, the ADC resolution, and the location of a particular sample within its re-

spective packet. The delay can be reduced in software by reducing the number of bytes stored in the FIFO before transmission, but the effects of such a reduction on the system throughput have not been measured. At present, latencies do not represent a significant problem for real time BMIs, since most control algorithms currently depend on spike counts that are tallied in bins as large as  $100msec$  [19]. However, since different populations of neurons communicate with different degrees of temporal resolution [16], future BMIs may require reduced bin sizes, hence increasing data latencies. Thus, the data latency issue may emerge in the future as a factor requiring improvement.

## 5.8 Information Transmission

The Backpack's practicality is enhanced by its ability to transfer either the raw data signal or the detected spike waveforms. Having access to the raw signal allows for the faithful transmission of subthreshold depolarizations and hyperpolarizations that may be indicative of important synaptic behavior [87]. Observing the raw signal also helps determine signal qualities such as noise characteristics, signal to noise ratio, and the presence of spikes. However, switching to a spike detection mode reduces the amount of data being transmitted, which may translate into more channels able to be passed over the wireless link or lower power consumption. Further data reductions could be achieved by transmitting only bin counts of either sorted or detected spikes, although this has not been tested.

## 5.9 Multipath / UDP

Multipath echoes can adversely affect the signal quality of any wireless transmission network [88]. In the Backpack, multipath manifests itself by occasionally causing UDP packets to be lost during transmission. Since the UDP network protocol does

not allow for retransmission of missed or garbled data, this information is permanently lost. In contrast, TCP packets are guaranteed to arrive, but the overhead necessary to ensure such performance reduces the wireless throughput by 50 – 70%, depending on packet size [89]. The Backpack was tested in a standard laboratory setting and was able to transmit 12 channels at a range of nine meters. However, during the spike detection tests, which required opening two UDP sockets to the same host (one apiece for the master and slave), UDP packets were more frequently lost or garbled. In general, UDP was found to be a functional but unreliable transmission protocol. Future work should consider other data protocols for improving transmission reliability; TCP may be worth reconsidering with the reduced bandwidth spike detection data.

## 5.10 Portability

This work has not yet been tested with a freely moving subject; all data in this research came from anesthetized or restrained subjects. One hurdle that remains to untethering the Backpack is the development of a custom jacket that lets the animal carry the electronics while protecting them from damage. A preliminary jacket design has been tested (see Appendix C for pictures). The jacket must protect not only the Backpack, but also the headstage and the cable that connects them together. The jacket currently under development is hooded and zips in the back from the lumbar region up to the head. The single rear zipper makes it difficult for the subject to open the jacket. Preliminary tests suggest that it will be necessary to habituate the subject to the jacket over the course of weeks or months.

## 5.11 Spike Detection Simulations

Although spike detection has been studied extensively by others, its implementation in a BMI presents new problems. BMI spike detectors must work in real-time using limited computational resources that must be divided among dozens of channels. BMIs must also be fully autonomous; as the number of channels in a system scales, it will become unreasonable for a user to manually determine threshold levels or create spike templates. The simulations in Chapter 2 investigated how best to emphasize neural spikes against background noise, and did not address the issue of how to determine the optimal threshold value for a particular combination of preprocessor and signal SNR.

Although none of the preprocessors that were examined stood out clearly as the best, the absolute value operator had a slight edge over the other algorithms, but only for computationally limited detectors. When computational complexity was disregarded as a factor in comparing spike detectors, the matched filter preprocessor scored the highest. Performance was not affected by whether the matched filter template was specific (i.e. based solely on the spikes in the signal under detection) or non-specific (i.e. an average of spikes from a number of different signals). This is significant because it means that autonomous matched filtering might be used to detect spikes without requiring the user to manually create a template from a test set of data. In all cases, the results indicated that the best way to improve spike detection performance is to enhance signal to noise ratio. The findings also demonstrate that SNR gain is not necessarily a good predictor of a preprocessor's spike detection performance.

The cost function succeeded in combining the probability of detection and the rate of false alarms into a single metric that could be plotted versus threshold value. This simplified comparisons between the different preprocessors and led to a more

intuitive interpretation of neural spike detection than would have been possible with ROC curves alone. The constant terms in Equation 2.1 may be modified to reflect the design constraints of different BMI architectures. For example, if a particular BMI will not include spike sorting in the receiver, then false positive detections will directly contribute to the prosthesis control, negatively affecting its performance. In such a case, the weights  $w_1$  and  $w_2$  may be made equal to increase the importance of minimizing false positives relative to that of maximizing correctly detected spikes.

## 5.12 Threshold Selection

Thresholds for the Backpack spike detector were determined by multiplying the mean of the signal absolute value by a constant (*alpha*). This work did not attempt to compare autonomous threshold selection algorithms for use with the spike detector. The research in Chapter 2 has merely established methods for increasing the range of acceptable thresholds in which optimal spike detection may be achieved. Widening this range increases the acceptable margin of error for an automated threshold selection algorithm. An autonomous BMI must be able to determine its own threshold levels without user input. A study comparing the effects of different threshold selection routines on cost function scores would be a useful contribution to the literature. A common threshold setting technique is to set the threshold as a constant multiple of the standard deviation of the signal [50]. Two novel techniques were devised but were not formally evaluated. The first is to sweep a threshold across a test set of preprocessed data and to count the number of positive threshold crossings at each threshold. As the threshold is increased from the minimum range value, the count will taper off suddenly when the threshold exceeds the noise floor. The location of this corner may potentially be of use in setting the experimental threshold. The second technique is to use a test set of data to set the threshold such that a constant

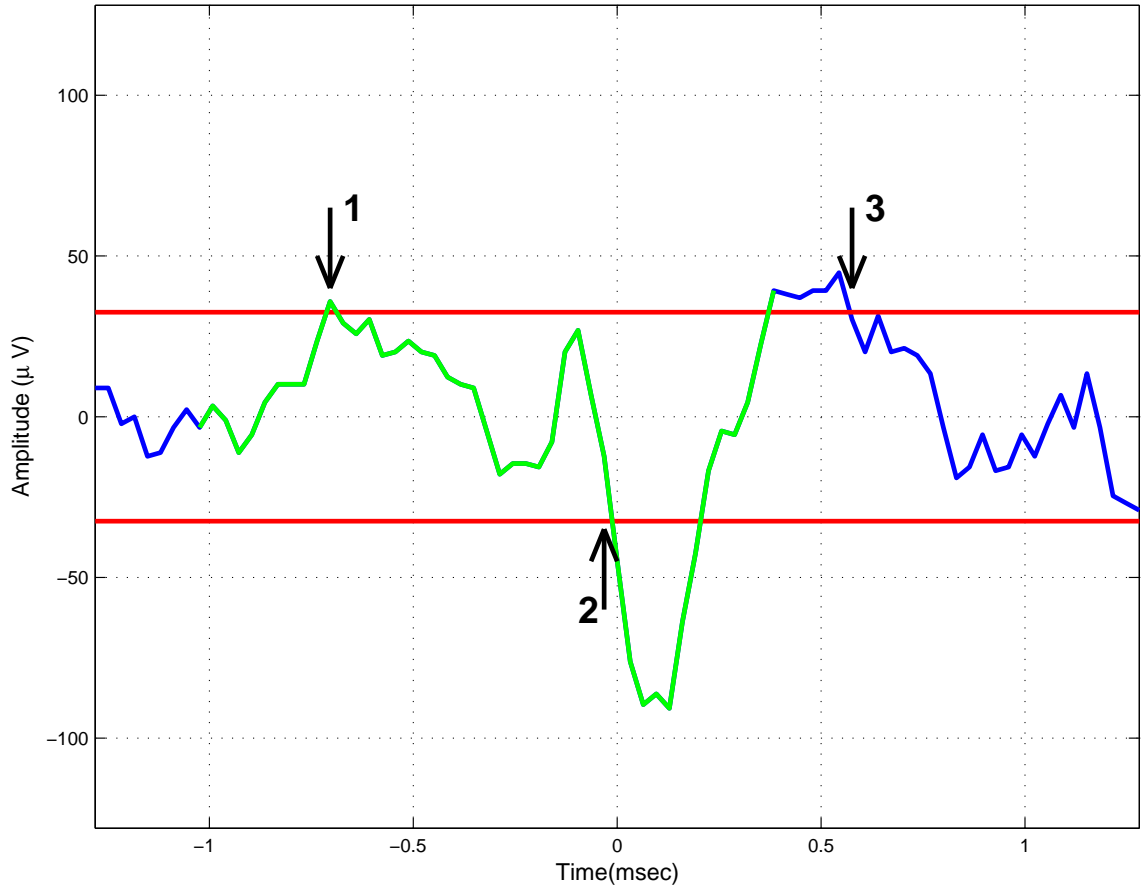
number of detections are guaranteed to be transmitted per second, even if many of them are false positives that must be identified and discarded by the host PC. This algorithm ensures that the Backpack is always operating at or near its maximum telemetry bandwidth.

### 5.13 *In Vivo* Spike Detection

The *in vivo* evaluation of the Backpack spike detection module successfully demonstrated the absolute value preprocessor at work in a real time hardware implementation. Over 92,000 detectable events on 24 channels in two subjects (one rat and one owl monkey) were passed through the Backpack. In most cases, when the threshold level was set optimally, the Backpack successfully detected greater than 90% of the spikes, especially if the SNR was greater than 2.1:1.

The cost function trends observed in Chapter 2 were consistent with the *in vivo* spike detection data collected by the Backpack. Figure 4.11 shows that, in general, the cost function score increases from  $\alpha = 2$  to  $\alpha = 3$ , and then decreases as the threshold is further increased. At low thresholds, there are many false positive detections, which lower the cost function score (see Equation 2.1). Once the threshold exceeds the noise floor (typically at  $\alpha = 3$ ) the incidence rate of false positives decreases and so the cost function score increases. Once the threshold increases above the peaks of the spikes, both correct detections and false alarms go to zero and the cost function reduces to the third term of Equation 2.1.

In several cases, the cost function scores exhibited a local minimum around  $\alpha = 5$ . These are caused by a “masking” phenomenon wherein a spike is not detected because it occurs during a refractory period created by an earlier false detection. An example is shown in Figure 5.1. The spike’s true arrival time (marker 2) was determined by the Offline Spike Sorter. The signal first crosses the threshold at marker 1, initiating



**Figure 5.1:** Spike detection “masking”. A pair of thresholds applied by the *Abs* operator (red) are applied to an action potential (green). The first threshold crossing (marker 1) triggers a 40 sample refractory period which causes the true spike arrival time (marker 2) to be missed. The refractory period ends at marker 3. This common situation forces one false alarm (at marker 1) and one missed detection (at marker 2).

a 40 sample refractory period that ends at marker 3. The detection at marker 1 is counted as a false positive, since it is more than 20 samples away from marker 2. The threshold crossing at marker 2 does not qualify as a detectable event since it falls within the refractory period. The cost function is therefore penalized with one false detection and one missed detection. When the threshold is increased, the false detection at marker 1 no longer occurs. This allows the threshold crossing at marker 2 to be correctly identified, thus increasing the cost function score.

The action potentials that comprise the *in vivo* Backpack spike detection data range in SNR from 1.9 : 1 to 9.1 : 1. This is representative of the full range of typical cortical microelectrode recordings in both rats and primates [24, 25]. Since the absolute value preprocessor only depends on the relative amplitudes of the spike versus the noise floor, it is reasonable to extrapolate that the results of Section 4.2.2 will apply to other cortical microwire recordings, both in subjects of the same and of different species. This assumes only that the background neural noise must be stationary once the threshold value is computed. At present, the Backpack computes a new threshold for each channel every 60 seconds, which is sufficient since that individual SNRs are typically stable over the course of several days [24, 25]. Nevertheless, the 60 second window is a software setting that may be adjusted up or down as necessary to match changing noise amplitudes.

This work has not attempted to address the question of how spike detection might change in response to varying parameters such as electrode length, electrode cross sectional area, distance of the electrode from a neuron, type of neuron being recorded from, and placement of the neutral reference electrode. Changes in these parameters may alter neural signal waveforms, which in turn may effect the choice of optimal preprocessor. More work is required in this area to validate the results of the work presented here.

A limitation on the spike detection results is the lack of a gold standard against which to compare the Backpack’s performance. True spike times were not known *a priori*, but were instead approximated using the Offline Spike Sorter to analyze the raw streamed data from the slave module. The results of the Offline Spike Sorter analysis inherently reflect the bias of the person who sorted the data (the author, in this case). This bias may be eliminated by testing the spike detector in a complex neural model that keeps track of both intracellular and extracellular action potentials [80]. Alternatively, it may be possible to simultaneously record both intracellular and extracellular potentials *in vivo* [36]. In both cases, knowledge of the transmembrane potential yields actual spike times, making possible an unbiased assessment of the spike detector’s performance. It may be possible to address the bias problem by having multiple trained observers each spike sort the data, and then averaging the resulting cost function scores. However, since spike detection results can vary by as much as 30% between trained observers, this will not necessarily reduce the element of human bias or the rates of spike detection errors [90, 91].

The spike detection module eliminated approximately 97% of the neural signal (i.e. data compressed to 3%). While this figure represents a substantial decrease in the data, it depends heavily on the mean firing rate (MFR), which was 9.3 spikes per second per channel. This MFR value is low when compared with data recorded from the motor cortex of a macaque actively participating in a motor task; under these conditions, a recent study reported  $\text{MFR} = 8.9$  spikes per second *per neuron* [92]. Assuming an average of two neurons per channel [24, 25], a typical sustained MFR would therefore be  $2 \times 8.9 \approx 18$  spikes per second per channel. Hence, assuming a linear relationship between MFR and data compression, the expected compression under these more typical conditions would be  $3\% \times \frac{18 \text{ sp/sec/ch}}{9.3 \text{ sp/sec/ch}} = 5.8\%$ . However, data compression may not necessarily scale linearly with MFR, since it is unclear whether

false positive rates will increase with firing rate. More *in vivo* studies must be done in the future to make a better approximation of how data compression varies with spike firing rates.

It is important to note that data compression does not automatically translate into increased channel counts. Bossetti et al recently recommended that a wireless spike detection system's available bandwidth should exceed the actual data rate by as much as 5:1 in order to reduce data latencies that arise due to neuronal bursting [92]. Therefore, spike detection algorithms can only be used for increasing channel counts when they reduce the data by factors of greater than 5:1. However, spike detection will still pay dividends, as reducing the amount of transmitted data reduces power consumption and thus increases the battery life in a portable system. Spike detection is also an essential first step for any automated spike detection or binning algorithm.

Channel count estimates for future revisions of the Backpack are difficult to make, since they depend on a number of as-yet untested factors. These include (1) performance enhancements from switching to new wireless technologies (such as IEEE 802.11g), (2) upgrading the wearable PC to a faster model with DMA-modulated USB data acquisition, and (3) the relationship between MFR and the percentage of data reduction in the spike detector. However, based on conservative estimates, a system with as many as 300 channels might be possible. And so, by using off the shelf technology and simple but robust processing algorithms, it will be possible to expand this work to build a wireless neural data acquisition system for use with sophisticated future BMIs. These will help to restore autonomy to amputated and paralyzed patients, and contribute to unlocking the deepest mysteries of the human brain.