High Performance Flexible Protocol for Backscattered-based Neural Implants

Laura Arjona
Computer Science and Engineering
University of Washington
Seattle, USA
arjonal@cs.washington.edu

James Rosenthal, Joshua R. Smith
Electrical and Computer Engineering
University of Washington
Seattle, USA
{jamesdr, jrs}@uw.edu

Chet T. Moritz
Rehabilitation Medicine
Electrical and Computer Engineering
University of Washington
Seattle, USA
ctmoritz@uw.edu

***Abstract*—This work presents a custom high-performance protocol for bi-directional communication with neural implants, that will eventually enable closed-loop operation. This protocol presents a flexible configuration to communicate to neural implants with different characteristics. It can support different uplink data rates, a variable number of neural channels from 2 to 16, two types of digital signal modulation (Amplitude Shift-Keying, ASK, and Binary Shift-Keying, PSK), and different RF operation frequencies (915MHz being the default). The proposed protocol is implemented in C++ (preferred to Python because it enables fast signal processing algorithms), using GNU-Radio toolkit with custom communication blocks.**

***Index Terms*—neural implant, BCI, backscatter communication, wireless communication.**

I. Introduction

It is becoming increasingly clear that Brain Computer Interface (BCI) systems will eventually realize important new technologies that replace, restore, supplement, or improve function in people affected by neurological disorders. The term BCI describes devices that interface directly with the brain via recording and/or stimulation hardware [1].

Neural interface devices enable brain-controlled technology and provide tools for studying the brain and treating neural disorders. The next generation of such devices must be miniaturized and implantable to record neural signals and stimulate neurons [2]. Neural implant systems depend on software to acquire, synchronize, and process neural signals, and behavioral data in real time. The acquisition of those signals is typically accomplished using specialized hardware devices and requires the implementation of proprietary software interfaces to configure the device and acquire data in real time [3]. One main challenge in realizing the software for neural implants is the need to implement a wireless communication protocol with high data rate using low power, and to achieve bidirectional communication with low latency. Furthermore, the next generation of neural implants must operate in a closed-loop framework [1].

This work presents a custom protocol for bi-directional communication with backscattered-based neural implants, that will eventually enable a closed-loop operation. The proposed protocol can be implemented on a reader consisting of a commodity Software Defined Radio system and a PC. Thus, it will provide a widely available high performance system for researchers working with different neural implants. The protocol is implemented and initially validated using the system presented in Fig. 1. This system will be explained with detail in the next sections.

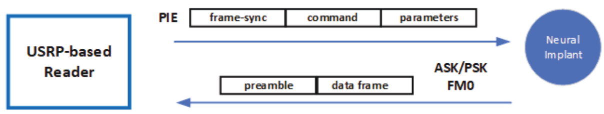

Fig. 1. A system description of the proposed communication protocol between the reader and the neural implant. The reader communicates with the implant using Pulse Interval Encoding (PIE)

II. Background on BCIs and Neural Implants

A brain-computer interface is a device that allows signals from the brain to control systems systems such as prosthetics, cursors, or robots with signals from the brain [4]. It also allow external signals to be delivered to the brain through neural stimulation. BCI systems can be traced back to early 1960s, and the very term BCI was coined in 1970 [3]. Despite their promising start, it was not until 1990 when this field when this field experienced considerable growth, due to the development of multi-electrode recordings and fast and inexpensive computers. Since then, interest and research efforts in BCIs have grown tremendously, with possibly hundreds of laboratories around the world studying this topic [5]–[8].

Neural implants are a widely studied type of BCIs, because they have the potential for significant impact in medicine, from restoring the use of the limbs after a spinal cord injury, to bioelectronic medicines. Backscatter communication is promising method for low power communication. In backscatter communication, an external reader (outside the body) generates radio waves. The energy constrained implant communicates by then selectively reflecting those radio waves. This approach requires much less power than the conventional method for sending data, in which the implant generates its own radio signals [7] [9]. Backscatter communication is particularly well-suited to communication data from implantable devices. Thus, this work focuses on backscatter-based neural implants.

978-1-7281-0566-6/19/$31.00 ©2019 IEEE

III. Protocol for Backscatter-Based Neural Implants

It is becoming increasingly clear that BCI systems will eventually realize important new technologies that replace, restore, supplement, or improve function in people affected by neurological disorders. The extent to which these possibilities are realized, that is, the extent to which BCI technologies achieve substantial practical impact, is critically dependent on the ability to effectively and efficiently implement and test different BCI approaches. The purpose of BCI and neural implant software is to facilitate such implementation and testing across the whole range of research and development, that is, from basic research to clinical translation and, in some cases, even to commercialization For BCI firmware to fulfill this purpose, it needs to satisfy two important requirements [3]:

- Satisfy the technical demands:
  - Acquire signals from the brain and/or other psychological or behavioral sources.
  - Analyze these signals to produce output commands.
  - Produce the output and associated feedback.
- Rapidly facilitate the implementation, verification, and dissemination of any and all bci experiments planed in a particular laboratory.

A. Protocol Overview

The physical interface between a neural implant and a reader may be viewed as the signaling layer in a layered network communication system. The signaling interface defines frequencies, modulation, data coding, data rates, and other parameters required for RF communications. Regarding frequency range, neural implants shall receive power from and communicate with the reader within the frequencies rage from 860 to 920 MHz, inclusive. The choice of the frequency will be determined by local radio regulations and by the local radio-frequency environment.

B. Downlink Communication

This section presents the reader-to-implant communication.

1) Modulation and data encoding: The reader communicates with one implant at a time using Amplitude Shift Keying (ASK) modulation with Pulse Interval Encoding (PIE). PIE encoding is used so that there is ample radio frequency energy from the reader to power the implant. The reader uses a fixed Pulse Width (PW) duration of 0.5μs. A data-0 has a duration of $2PW$=1μs. A data-1 has a duration of two times that of a data-0, that is, $4PW$=2μs. Fig. 2 shows the PIE symbol specifications.

High values represent transmitted Continuous Wave (CW), and low values represent attenuated CW. Assuming equal probability of transmitting data-0 and data-1 symbols, the reader achieves a data rate of 1.5Mbps.

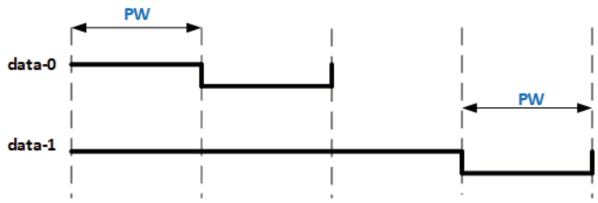

Fig. 2. PIE symbols.

C. Uplink Communication

This section presents the implant-to-reader communication system. The number of neural channel used in the communication will depend on the particular neural implant device. The proposed protocol supports communication with 2, 4, 8, or 16 channels, with 16 bits per channel. For example, the system presented in [5] uses the platform Intan technologies (RHS2116), which has 16 unipolar channels.

1) Modulation and data encoding: The implant shall use Amplitude Shift Keying (ASK) or Phase Shift Keying (PSK) modulation. For PSK, the protocol supports Binary-PSK (BPSK) and Differential Quadrature PSK (DQPSK). The data shall be encoded using FM0 baseband. FM0 inverts the baseband phase at every symbol boundary, and a data-0 has has a mid-symbol phase inversion. Every frame shall begin with a preamble of 6-bits-long: 101011.

Each data frame contains a 6-bit preamble, the data sensed from the neural channels, and the frame counter FC parameter of 8 bits-long. The FC parameter is a 8-bit frame counter permitting missing frames to be identified in the data stream in the event of bit errors that result in the loss of frame synchronization. The neural data shall be encoded using Hamming H(11,15). Then, the encoded frame will be interleaved. Details are provided in Section III.E.

D. Reader commands

The reader can transmit four different commands to communicate with the implant: $Start$, $Read$, $Cont$, and End. The reader commands begin with a frame synchronization (frame-sync) pulse. A frame-sync comprises a fixed-length start delimiter and a data-0 symbol. The delimiter has a duration of 3μs.

The $Start$ command (see Table I) initiates the communication process with a particular implant. The reader selects a particular implant by including the DID field in the command, consisting of 12 bits.

TABLE I
Reader $Start$ command

	Command	DID
# bits	2	12
Description	00	Device Identifier

The reader can receive data from 2, 4, 8, or 16 neural channels of the implant. The Read command (see Table II) enables the selection of particular channels with the $channels$

field. For example, if the reader wants to receive data from channels 1, 4, 8, 10, and 16, the $channels$ field of the Read command will be 1001000001000001. The field $FNCT$ specifies the number of frames received from the implant before transmitting a new command.

TABLE II
READER $Read$ COMMAND

	Command	**channels**	**FCNT**
# bits	2	16	8
Description	01	Active channels	# consecutive frames

The $Cont$ command (see Table III) is used to maintain he synchronization between the reader and the implant. The reader transmits a $Cont$ command after every $FCNT$ data frames received form the implant.

TABLE III
READER $Cont$ COMMAND

	Command
# bits	2
Description	10

Fig. 3 shows an example of a $Cont$ command transmitted by the reader.

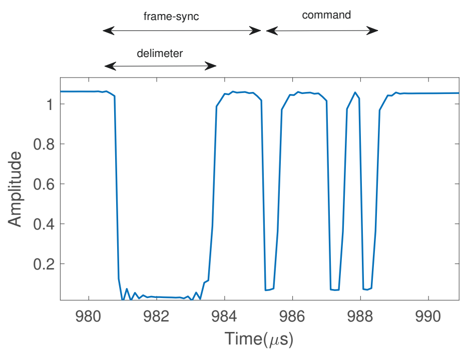

Fig. 3. Detail of reader $Cont$ command. Signal received at the USRP source.

Finally, the reader will terminate the communication with a particular implant with the End command (see Table IV).

TABLE IV
READER End COMMAND

	Command	**DID**
# bits	2	12
Description	11	Device Identifier

E. Neural implant data frames

The neural implant can transmit data from 2, 4, 8, or 16 neural channels to the reader, with 16 bits per channel. The recorded neural data, together with the DID and the FC, is packetized into a frame, and transmitted to the reader preceded by the 6-bits preamble.

1) Hamming encode: The neural implant shall encode the neural data using H(11,15), which provides single error detection and correction capability, providing robust data transmission in noisy environment. It means that for every 11 bits of data, a total of 15 bits will be transmitted. Each data frame consist on a variable number of bits, depending on the number of channels:

- 2 channels: 74-bits/frame, containing 32-bits of neural data and 42 bits of additional data (parity bits, DID and the FC)
- 4 channels: 126-bits/frame, containing 64-bits of neural data and 62 bits of additional data (parity bits, DID, FC, and control/application bits).
- 8 channels: 252-bits/frame, containing 128-bits of neural data and 124 bits of additional data (parity bits, DID, FC, and control/application bits).
- 16 channels: 504-bits/frame, containing 256-bits of neural data and 248 bits of additional data (parity bits, DID, FC, and control/application bits).

2) Interleave: Lastly, the implant shall perform a pattern interleave algorithm over the encoded frame. This technique is used to make forward error correction more robust with respect to burst errors. The interleaving algorithm shall use a permutation vector of length equal to the frame length. The permutation vector is not exchanged in the communication process, but it is known in advance by both the reader and the implant. Thus, if the frame is intercepted, the attacker will not know the permutation vector needed to de-interleave the frame. For example, in an scenario with 4 neural channels, the frame length is 126 bits. This means that there are 126! different possibilities for the permutation vector.

IV. READER IMPLEMENTATION

The reader that implements the proposed protocol consists of an USRP N210. The USRP is responsible for supplying the UHF communication carrier and receiving the backscatter subcarrier containing containing the uplink data from the implant. The USRP uses an SBX daughterboard, connected to a Linux PC. The transmit and receive ports of the daughterboard are connected with two circularly polarized patch antennas of 6dBi gain. The USRP is connected to the PC using Ethernet.

The protocol firmware is implemented in C++ and Python, using GNU Radio, an open source block-based software development toolkit. Fig. 4 shows the pipeline of the receiver design at the reader.

The gate block only passes samples to the next block after detects the end of transmitted reader signal is detected. Thus, the decoder block does not need to be processing samples continuously. The decoder block is responsible for the frame-synchronization, channel estimation, and detection of the implant data frames.

The implant encodes the information using FM0 line encoding. Level transitions occur on the bit boundaries. In addition, a transition occurs in the middle of bit ’0’. Thus, there is memory-based modulation, resulting in four different waveform per bit (see Fig. 5). Previous work [10] [11] have shown

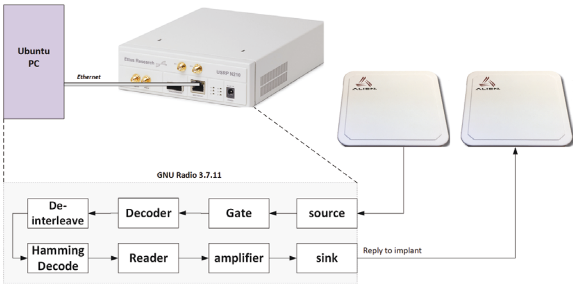

Fig. 4. A system description of the custom backscattered-based neural implant reader.

that after shifted examination of the transmitted waveform by $T/2$ before the beginning ot eh bit, where T is the bit (using one bit per symbol) rate, only two possible pulse shapes can be generated, referred to as S_0 and S_1.

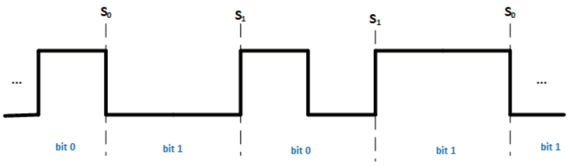

Fig. 5. FM0 signal

Synchronization for the data frame is performed by correlating the received signal with the known implant preamble of 6 bits. To deal with a possible variation in the implant nominal bit duration, the appropriate sampling instant is obtained such that the signal energy is maximized. The signal detection is performed based on the following Maximum Likehood (ML) rule:

$$\mathbb{R}\left(h'^*\left(y_1 - y_0\right)\right) \underset{S_0}{\overset{S_1}{\gtrless}} 0, \tag{1}$$

where $\mathbb{R}(z)$ denotes the real part of complex z, h' corresponds to the channel estimation, y_1 and y_1 refer to 2 consecutive samples of the received digitized signal.

At the end of the decoder blocks, the reader has the bit-stream transmitted by the implant. Next, it de-interleaves the frame applying the known permutation vector, and performs a hamming-decode algorithm.

The reader will further process and analyze the neural data, depending on the application. For example, the reader could perform a canonical correlation analysis (CCA) [5]. Finally, the reader block generates and transmits the next corresponding command based on the decoded and analyzed neural data.

A. Experimental evaluation

The proposed protocol has been implemented in the presented USRP-based reader using emulated neural implant data frames (see Fig. 4). The neural implant frames have been emulated by transmitting active frame signals with the USRP, and received at the USRP source. The USRP Digital-to-Analog conversion rate is set to 8Msamples/second. Considering that the implant is backscattering data at 2Mbits/second, the reader needs to process 4 samples per FM0 symbol from the implant. Also, one FM0 symbol (one bit) has a duration of 0.5μs.

Fig. 6 shows one emulated frame received at the USRP source, assuming an scenario with 4 neural channel data per frame.

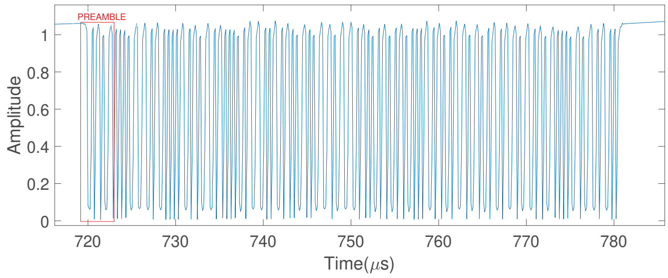

Fig. 6. One emulated neural implant data frame according to the proposed protocol, for 4 neural channels (a total of 126 bits per frame). Signal received in the USRP source.

Next, it is evaluated the time required by the reader to detect and decode one data frame. That is, to obtain the neural data from the received samples at the USRP source. A timer is set in the reader program to measure this parameter. The timer is started when the decoder block receives the first sample from the gate block. The timer is stopped when the neural data is recovered after performing the de-hamming algorithm. An average processing time of 71μs has been obtained experimentally. This means that the USRP-based reader, implementing the proposed protocol, is capable of decoding the neural data and be ready to transmit a new command in an average of a 71μs. According to [2], to enable users to naturally regulate their neural activity, a BCI system must compute and output feedback within a short period of time (≈100ms) or less. Thus, the proposed protocol implemented in SDR-based reader meets within with a wide margin to further analyze the neural data, and for the implant to perform the stimulation.

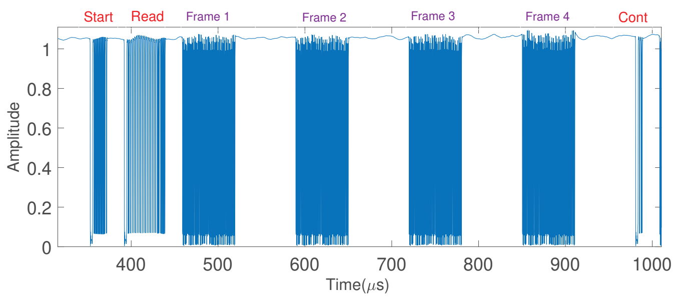

Fig. 7. Example of communication process between the USRP-based reader, and emulated neural implant. Signal received at the USRP source. The frame contains data from 4 neural channels (a total of 126 bits per frame), and $FCNT$=4.

V. Future work

Ongoing work includes a real time validation of the full protocol with two existing neural implants: the NeuralCLIP [5] and the NeuroDisc [6]. The previously presented reader, consisting on a USRP N210 will be employed. The NeuralCLIP uses ASK modulation, while the NeuroDisc uses DQPSK. Thus, they will provide two different validation scenarios.

Acknowledgment

This work has been supported by the UW Institute for Neuroengineering and the Washington Research Foundation Funds for Innovation in Neuroengineering. Funded in part by the Center for Neurotechnology, a National Science Foundation-Engineering

Research Center under Grant EEC-1028725

References

[1] C. T. Moritz, "Now is the Critical Time for Engineered Neuroplasticity," Neurotherapeutics, 2018.

[2] M. A. L. Nicolelis, "Actions from thoughts," Nature, vol. 409, no. 6818, pp. 403–407, 2001.

[3] 2. C. S. Nam, A. Nijholt, and F. Lotte. "Brain–Computer Interfaces Handbook: Technological and Theoretical Advances"," CRC Press, January 2018, 1st Edition.

[4] Rajesh PN Rao, "Towards neural co-processors for the brain: combining decoding and encoding in brain–computer interfaces," Current Opinion in Neurobiology, Volume 55, 2019.

[5] V. Ranganathan et al., "NeuralCLIP: A Modular FPGA-Based Neural Interface for Closed-Loop Operation," 9th International IEEE/EMBS Conference on Neural Engineering (NER), San Francisco, CA, USA, 2019, pp. 791-794.

[6] J. Rosenthal E. Kampianakis, A. Sharma, and M, S. Reynolds, "A 6.25 Mbps, 12.4 pJ/bit DQPSK Backscatter Wireless Uplink for the NeuroDisc Brain-Computer Interface," IEEE Biomedical Circuits and Systems Conference (BioCAS), 2018.

[7] S.J. Thomas, R.R. Harrison, A. Leonardo and M.S. Reynolds, "A Battery-Free Multichannel Digital Neural/EMG Telemetry System for Flying Insects," IEEE Transactions on Biomedical Circuits and Systems, vol. 6, no. 5, pp. 424-436, Oct. 2012.

[8] Y. Lin et al., "A Battery-Less, Implantable Neuro-Electronic Interface for Studying the Mechanisms of Deep Brain Stimulation in Rat Models,' in IEEE Transactions on Biomedical Circuits and Systems, vol. 10, no. 1, pp. 98-112, Feb. 2016.

[9] V. Talla, V. Ranganathan, B. Mahoney, J.R. Smith, "Dual Band Wireless Power and Bi-Directional Data Link for Implanted Devices in 65 nm CMOS," IEEE Symposium on Circuits & Systems (ISCAS), Montreal, Canada, 2016.

[10] A. Bletsas, J. Kimionis, A. G. Dimitriou, and G. N. Karystinos, "Single-antenna coherent detection of collided FM0 RFID signals," IEEE Trans. Commun. , vol. 60, no. 3, pp. 756–766, Mar. 2012.

[11] N. Kargas, F. Mavromatis and A. Bletsas, "Fully-Coherent Reader With Commodity SDR for Gen2 FM0 and Computational RFID," IEEE Wireless Communications Letters, vol. 4, no. 6, pp. 617-620, Dec. 2015.