35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

A Flexible Algorithm Framework for Closed-Loop Neuromodulation Research Systems

Dave Carlson, Dave Linde, Ben Isaacson, Pedram Afshar, Duane Bourget, Scott Stanslaski, Paul Stypulkowski, and Tim Denison, *Member, IEEE*

***Abstract*— Modulation of neural activity through electrical stimulation of tissue is an effective therapy for neurological diseases such as Parkinson's disease and essential tremor. Researchers are exploring improving therapy through adjustment of stimulation parameters based upon sensed data. This requires classifiers to extract features and estimate patient state. It also requires algorithms to appropriately map the state estimation to stimulation parameters. The latter, known as the control policy algorithm, is the focus of this work. Because the optimal control policy algorithms for the nervous system are not fully characterized at this time, we have implemented a generic control policy framework to facilitate exploratory research and rapid prototyping of new neuromodulation strategies.**

I. Introduction

Neurostimulators are used to treat symptoms of neurological diseases such as Parkinson's disease and essential tremor; this therapy is called deep brain stimulation (DBS). A key challenge that must be overcome to provide effective DBS therapy is optimizing selection of stimulation parameters (e.g., amplitude, pulse width, and frequency).

Currently available neurostimulators require that the clinician be in the loop to program the stimulation parameters. This largely empirical procedure is often very time-consuming and programming sessions may be weeks or months apart. Patients receive a patient programmer that can be used to make limited adjustments in the interim.

Many bioengineering fields are exploring closed loop therapies, such as cardiac pacing [1], diabetes [2], and respiration [3]. In the case of neuromodulation, this is also a logical approach to explore. While manual adjustment is available today, automaticity might potentially provide improved response time and thereby react before symptoms manifest externally. In addition, turning off the device when not required is a pathway to increasing device longevity. Finally, from the patient's perspective, we wish to eliminate the burden of adjustment that requires device interaction.

Efforts are underway to provide automaticity in implantable neurostimulators [4]. Referencing Fig. 1, sensing is a required capability to allow the device to measure signals that reflect changing conditions in the environment. Sensors within the device obtain signals from the nervous system directly or indirectly. Local field potentials (LFPs) are examples of signals measured directly from the brain. Inertial signals are a more indirect surrogate for brain state [5]. Another necessary component is a classifier or classifiers to extract information from sensed signals to provide a state estimate [6]. Once a state estimate is determined, a control policy algorithm must be present to make appropriate decisions. Determining the optimal control policy requires understanding how transfer functions in the nervous system map to desired outcomes, and then applying that understanding into a dynamic method of parameter control. Given the outstanding questions about neuromodulation mechanisms of action, we see a need to implement a flexible research tool for rapid prototyping and hypothesis verification; a prototype and pilot results of such a system is the focus of this work.

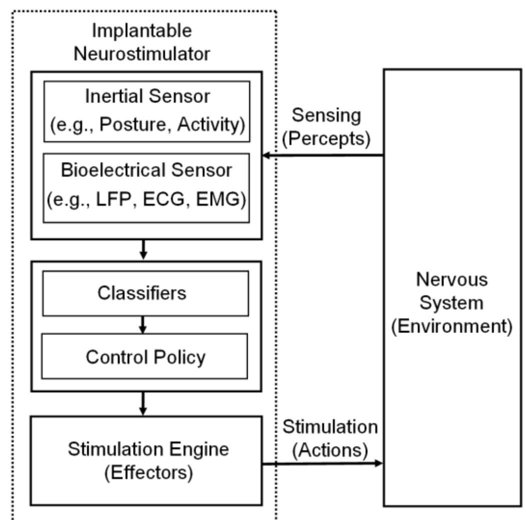

Figure 1. Model of a closed-loop neuromodulation system

II. Control Policy Considerations

There are many factors that need to be considered for the design of a control policy algorithm for a closed-loop neurostimulator. This section summarizes these factors.

A. Optimal Algorithms Largely Unknown

Development of control policy algorithms for closed-loop neuromodulation systems is in a very early stage. Although there are candidates for control policies [7, 8, 9], the desire remains to provide a flexible method to develop and test new control policy algorithms. This method provides the ability to explore a space that is largely unknown.

All authors:Medtronic Neuromodulation, Minneapolis, MN 55432 USA (phone: 1-763-526-8789; e-mail: timothy.denison@medtronic.com)

B. *Firmware Verification and Validation*

The classifiers and control policy algorithms discussed are implemented in the firmware of an implantable neurostimulator. This firmware must pass a rigorous set of testing before it is allowed to be used in a clinical setting. If each control policy algorithm requires new firmware, each firmware version requires a full cycle of development, verification, test, and approval. Avoiding this burden is highly desirable.

C. *Enabling Broad Hypothesis Testing*

There are many opportunities to be explored in closed-loop neuromodulation. While it is not efficient to provide a different set of firmware for each study, we do wish to enable the rapid exploration of several hypotheses to enable identification of the most promising opportunities.

III. METHODS

This section presents a general-purpose framework based upon state machines to allow the user to build control policy algorithms in an implantable neurostimulator. These state parameters can be programmed quickly and with much lower risk than a code update. This flexible framework allows the control policy algorithm to be readily modified, such as updating the number of states and their transitions (within limits of the finite memory of the embedded system).

This strategy for development of control policy algorithms was chosen over linear methods, such as a PID (proportional-integral-derivative) controller, for a number of reasons. Although a PID controller is often considered to be the best controller in the absence of knowledge of the underlying process, in practice PID controllers can have shortcomings when systems are nonlinear or dynamic; this is obviously a consideration in the nervous system.

Please note our goal is to demonstrate a versatile tool for investigating nervous system models rather than to show that a particular control strategy is the best for a particular outcome. Our key point is that model-based control may be accommodating to a better understanding of underlying mechanisms of action, where the control policy must react to different modes of process behavior.

A. *Degrees of Freedom for Delivery of Stimulation*

Stimulation parameters that are adjustable in the neurostimulator are as follows:

- Contacts – anode and cathode (incl. the case)
- Amplitude (with programmable limits)
- Pulse width
- Frequency

Four different stimulation programs can be preprogrammed into the device (P1-P4). Each of these mutually exclusive programs contains a separate set of the above-listed parameters.

The options that are available to the control policy algorithm for adjusting stimulation are as follows:

- Program switch (select one of 4 programs P1-P4)
- Turn stimulation OFF
- Turn stimulation ON
- Increment stimulation amplitude (INC)
- Decrement stimulation amplitude (DEC)

B. *Sensing and Classifiers*

The neurostimulator has a bioelectrical sensor and an inertial sensor. Spectral features are extracted from the bioelectrical sensors [4] and piped into classifiers that can be trained using machine learning techniques [6]. Posture and activity features can be extracted from the inertial sensors. The outputs of these classifiers are the inputs to the control policy algorithm which controls the stimulation engine.

C. *State-Based Control Policy Algorithms*

To provide flexibility in designing, testing, and modifying control policy algorithms, we developed a system whereby each algorithm is a collection of states. Each state has *entry actions* that can affect stimulation and *exit conditions* that are selectively based upon classifier states and the state of a timer. Fig. 2 depicts a dialog box that enables the user to create one such state.

The exit conditions enable a transition from the current state to another state. The user can build a control policy algorithm by creating states and connecting them together to form a state machine.

The *Entry Actions* section of the dialog in Fig. 2 contains fields that enable the user to select actions to control delivery of stimulation.

The fields are as follows:

- Stimulation Program
  Options: P1, P2, P3, P4, No Change
- Stimulation Control
  Options: ON (with flags), OFF, No Change
- INC/DEC Flags
  Options: Set INC flag, Set DEC flag, No Change

Whenever the algorithm first enters a state, the list of entry actions is executed. The *Stimulation Program* field allows the stimulation program to be optionally switched to any of the four available programs (P1-P4) or left as it was.

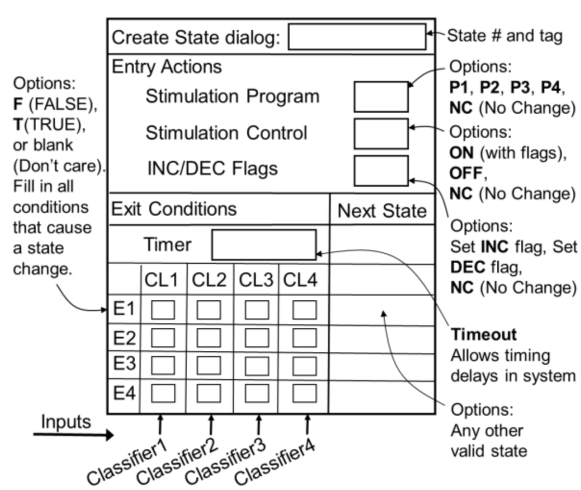

Figure 2. General-purpose control policy state framework dialog

The *Stimulation Control* field allows stimulation to be either turned ON, turned OFF, or remain unchanged. Whenever stimulation is turned ON with an unchanged stimulation program, the Increment/Decrement flags are checked and the amplitude is adjusted accordingly. The flags are then cleared.

The *INC/DEC Flags* entry action field enables the stimulation amplitude to be incremented (INC) or decremented (DEC). If stimulation is already ON, the adjustment will be made immediately. If not, a global flag is set (either INC or DEC) indicating that the next time stimulation is turned ON without a change of program in a future state, the stimulation amplitude will be adjusted accordingly (incremented or decremented). The new amplitude is relative to the previous "ON" value.

An expiring timer is one condition that causes a state transition. If the user enters a timer value in the *Timer* field, the timer is activated and will cause an exit from the state upon its expiration. The *Next State* field dictates to which state the algorithm will transition. The classifiers are updated and queried at a default rate of 200 ms, so this is used as the "tick rate" of the timer. In the GUI, times are represented in seconds. If no timer value is entered, the timer is not active and the state cannot be exited via the timer.

The classifiers (CL1-CL4) can be used to initiate a state transition. Up to 4 combinations of exit conditions (rows E1-E4) are checked in order to determine if the state should change. Each row contains options for each of the classifiers. These options are as follows:

- False
- True
- Don't Care (no transition)

Whenever the state of the classifiers matches a row in the table, the algorithm exits the current state and transitions to the corresponding *Next State*. All *Don't Care* conditions are ignored and do not contribute to a state transition.

D. Example: Building a Control Policy Algorithm

Fig. 3 shows an example of a control policy algorithm for modulating hippocampal network dynamics. This is based upon the embedded control policy described in [8].

Two classifiers are used as inputs to the control policy algorithm. One is an "after-discharge" detector (AD Det), which monitors for seizure-like activity and is operational even in the presence of stimulation [10]. Note that this detector can always cause a transition, even during events that are timed. This is designated as CL1 (Classifier 1) in Fig. 4 and Table 1. The second classifier is a suppression detector (abbreviated as "suppr" in Fig. 3) that monitors for a lower energy level in a selected frequency band and is designated CL2 (Classifier 2). Classifiers CL3 and CL4 are not used in this example. Stimulation is delivered (state 1) and the amplitude is adjusted according to the states of the two classifiers and the paths through the state machine. Fig. 4 shows how two of the states comprising the algorithm are entered into the system using the dialog box. The parameters entered for each state are shown in bold.

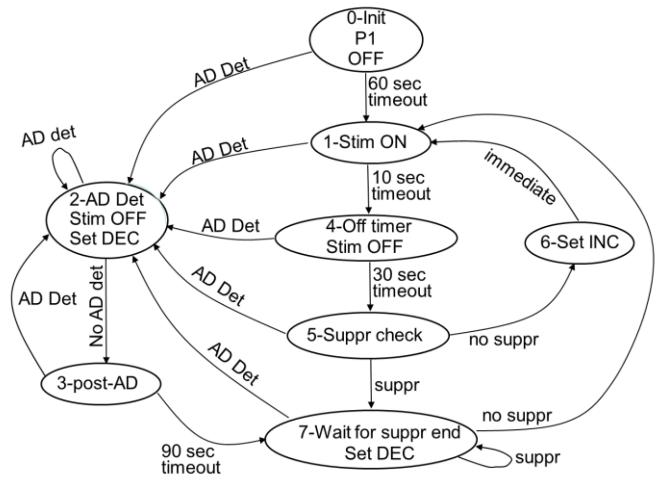

Figure 3. State diagram of example control policy algorithm

This algorithm initializes the system (state 0) by setting the stimulation program to P1 and turning the stimulator OFF. A 60 second timeout is set before the transition to state 1 (Stim ON).

When the system makes the transition to state 1 (Stim ON), the corresponding entry actions are taken. The *Stimulation Program* field of the dialog box is NC, so no change is made to the stimulation program (which is P1). The *Stimulation Control* field is set to ON, so stimulation is turned ON. A 10 second timeout is set to time the burst of stimulation. If CL1 (AD Det) becomes true during the 10 seconds, a transition is made to state 2, otherwise the stimulation remains ON for 10 seconds until transitioning to state 4. Fig. 4(a) depicts how to add state 1 to the algorithm.

If CL1 (AD Det) becomes true, a transition to state 2 will occur. Upon entry to state 2, stimulation is turned OFF. The decrement (DEC) flag is set so the next burst of stimulation will have a lowered amplitude. The only way to exit state 2 is for the AD detection to end (CL1 becomes false).

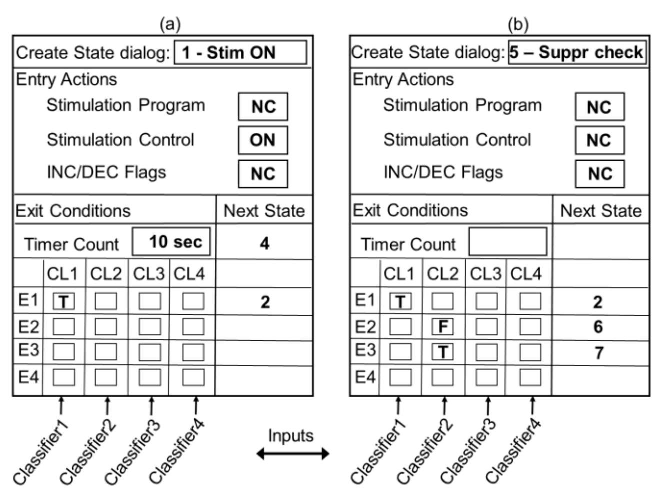

Figure 4. State dialog boxes for selected states of the example control policy algorithm. (a) Dialog box for building state 1. (b) Dialog box for building state 5.

State 3 is used to provide at least a 90 second delay before there is any chance of another burst of stimulation. State 4 turns stimulation OFF and provides a timeout. State 5 checks the output of the suppression detector (CL2) and acts accordingly. Fig. 4(b) depicts how to add state 5 to the control policy algorithm. Each of the exit conditions E1-E3 corresponds to a transition branch out of state 5 (Suppr check) in Fig. 3. Note that the timer is inactive, so all of the transitions are dependent upon the classifiers CL1 and CL2.

State 6 sets the INC flag. This will cause an increase in amplitude during the next state with an ON command. State 6 has an immediate transition to state 1.

State 7 is where this algorithm differs from that of [8]. A DEC flag has been inserted to allow the system to maintain equilibrium. It is possible for the amplitude to be lowered without detection of an after-discharge, which could be considered an undesirable side effect. State 7 waits for suppression to end before a transition to state 1, which will turn on stimulation.

Table 1 depicts how the example control policy algorithm is stored in internal memory. In state 6, "IT" stands for "immediate transition." The "suppr" loop depicted in Fig. 3 for state 7 is implicit because no entry is necessary for the system to remain in the current state.

This method provides a simple but powerful means to build and store a control policy algorithm. Once the table has been sent to the device, the algorithm can be enabled and the system will run in closed-loop mode.

Table 1. Internal storage of control policy algorithm as a table of states

State	Stim Prog	ON/OFF	INC/DEC	Timer	CL1	CL2	CL3	CL4	Next
0	P1	OFF	-	60	-	-	-	-	1
					T	-	-	-	2
1	NC	ON	-	10	-	-	-	-	4
					T	-	-	-	2
2	NC	OFF	DEC	-	F	-	-	-	3
3	NC	NC	-	90	-	-	-	-	7
					T	-	-	-	2
4	NC	OFF	-	30	-	-	-	-	5
					T	-	-	-	2
5	NC	NC	-	-	T	-	-	-	2
					-	F	-	-	6
					-	T	-	-	7
6	NC	NC	INC	IT	-	-	-	-	1
7	NC	NC	DEC	-	T	-	-	-	2
					-	F	-	-	1

E. Graphical User Interface

An interactive user interface tool has been created to facilitate the process of creating control policy algorithms. The tool contains a drag and drop state machine builder that allows states to be connected together. Entry actions and exit conditions can be specified via dialog boxes, and the algorithm is run through a rule checker to ensure consistency. The state machine table is automatically derived when the user wishes to send it to the device. Libraries of control policy algorithms can be developed, shared, and modified. This averts the need that users be state machine designers.

IV. RESULTS: CHRONIC PROTOTYPE VERIFICATION IN-VIVO

We verified our system building on an established ovine protocol [11], from which we have over two years of transfer function data. The implanted neurostimulator was programmed with the algorithm of the previous section and run in closed-loop mode. The hypothesis is that bursts of stimulation can create a lowering of the network excitability (suppression) and that proper modulation of stimulation can maintain suppression while avoiding after-discharges. Network excitability is measured as a 10 second average of the beta channel, which measures the relative energy in a frequency band centered at 20 Hz.

Fig. 5 shows a segment of data collected during this experiment. A difference between this algorithm and the embedded closed-loop algorithm in [8] is the addition of the decrement (DEC) flag in state 7. Another change was to start with low stimulation amplitude with the intention of avoiding the AD Det branch of the algorithm (state 2). This initial stimulation amplitude was also lower than that which could be expected to cause network suppression, so a number of increments were needed before suppression occurred.

The paths through the state machine of Fig. 3 during the experiment can be tracked in Fig. 5 through the use of the state labels. In the case of state 5 and state 6, the time durations are very short (200 ms) compared to the scale of the figure, so these are depicted with a line next to the state label. Corresponding stimulation amplitudes are also presented.

At 60 seconds into the figure, the first burst of stimulation is delivered with an amplitude of 0.1V. The control policy algorithm steps through states 4, 5, and 6 before returning to state 1, where it delivered the second burst of stimulation, this time with an amplitude of 0.2V. This sequence of states repeated until the amplitude reached 0.6V, whereupon the suppression detector (CL2) was first triggered. This caused a trip through state 7, where it waited until the end of suppression detection.

The algorithm was able to zero in on a region of stimulation amplitudes and maintain the desired network suppression without causing AD's, thereby demonstrating in a prototype the desired automaticity for parameter selection and the utility of the flexible control policy framework.

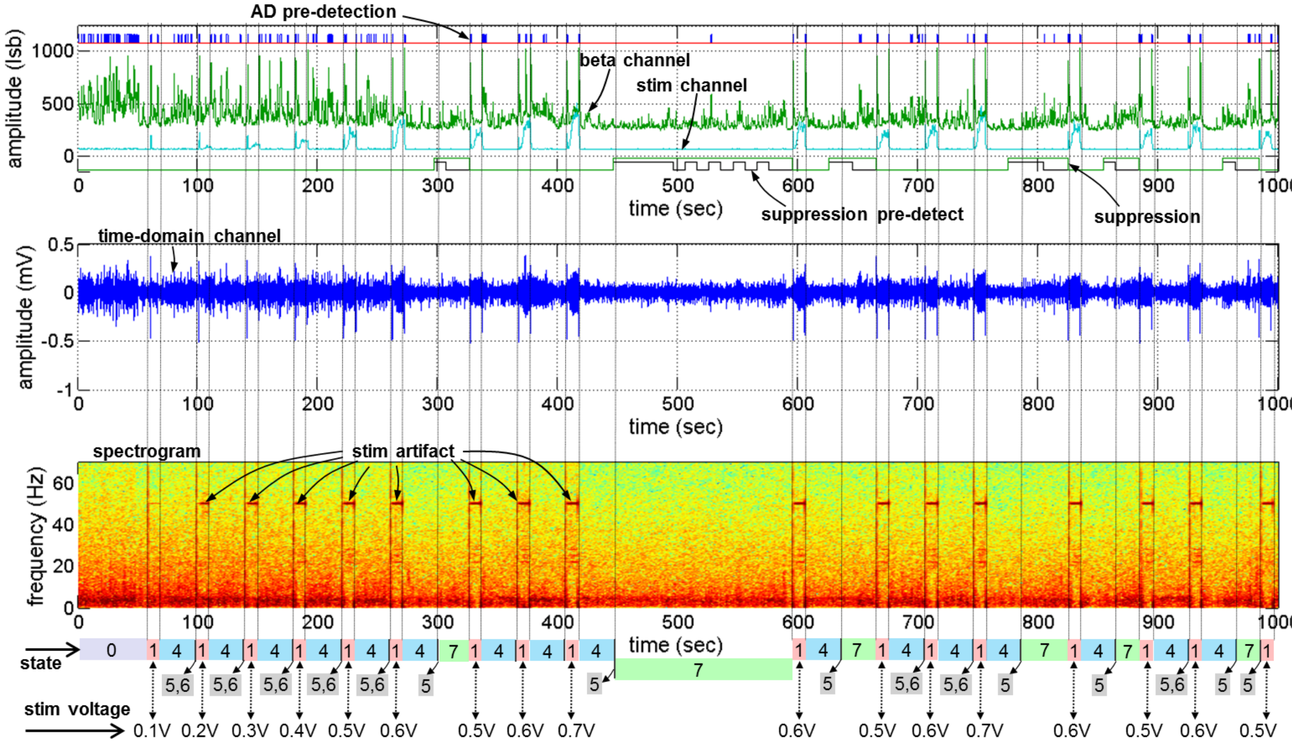

Figure 5. Segment of data collected during execution of a closed-loop control policy algorithm showing servo control in an ovine model

V. Discussion

Attempting to enhance therapies with closed loop control is a logical step in the evolution of DBS. Given the need for enabling investigators to explore this space from first principles, and with the remaining questions to be resolved around mechanisms of action, we believe providing a flexible framework for building control policy algorithms is a key requirement of achieving this goal.

The generic control policy framework demonstrated in this prototype provides a flexible method of building, exploring, and refining control policy algorithms that are compatible with the architecture of the implantable neurostimulator. A library of such algorithms can be built by researchers and/or the engineering team. The flexibility of exploring new paradigms of neuromodulation can be achieved without the burden of a firmware update for each algorithm, an important constraint for clinical translation.

The next step for this research system is to translate the methodologies to preclinical models of disease, with the goal to develop and verify algorithms for optimal stimulator control in chronic studies that represent the intended use cases for neuromodulation.

Conflict of Interest Statement

The authors are employees of Medtronic, Inc. and hold intellectual property in this field of practice. All work in this paper is investigational and not approved for use in the U.S.

References

[1] P. Palmisano *et al.,* "Closed-loop cardiac pacing vs. conventional dual-chamber pacing with specialized sensing and pacing algorithms for syncope prevention in patients with refractory vasovagal syncope: results of a long-term follow-up," *Europace* (2012) 14 (7):1038-1043.

[2] S. D. Patek *et al.,*"Modular closed-loop control of diabetes," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 11, November 2012, pp.2986-2999.

[3] N. Claure, MSc PhD *et al*, "Automated closed loop control of inspired oxygen concentration," *Respiratory Care*, January 1, 2013 vol. 58 no. 1, 151-161.

[4] S. Stanslaski *et al.,* "An implantable bi-directional brain-machine interface system for chronic neuroprosthesis research," *IEEE EMBC* 2009, pp. 5494-5497. DOI: 10.1109/IEMBS.2009.5334562

[5] I. Tien *et al.*, "Characterization of gait abnormalities in Parkinson's disease using a wireless inertial sensor system," *IEEE EMBC*, vol. 2010, pp. 3353–3356. DOI: 10.1109/IEMBS.2010.5627904

[6] A. Shoeb *et al.*, "A micropower support vector machine based seizure detection architecture for embedded medical devices," *IEEE EMBC*, vol. 2009, pp. 4202–4205. DOI: 10.1109/IEMBS.2009.5333790

[7] T. Yamamoto *et al.,* "On-demand control system for deep brain stimulation for treatment of intention tremor," *Neuromodulation* 2012. DOI: 10.1111/j.1525-1403.2012.00521.x

[8] P. Afshar *et al.,* "A translational platform for prototyping closed-loop neuromodulation systems," *Frontiers in Neural Circuits*, 6:117. DOI:10.3389/fncir.2012.00117

[9] F. Sun *et al.*, "Responsive cortical stimulation for the treatment of epilepsy," *Neurotherapeutics : the journal of the American Society for Experimental NeuroTherapeutics,* 2008;5(1):68-74.

[10] S. Stanslaski *et al.*, "Design and validation of a fully implantable, chronic, closed-loop neuromodulation device with concurrent sensing and stimulation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 4, July 2012, pp. 410-421. DOI: 10.1109/TNSRE.2012.2183617

[11] P. Stypulkowski *et al.,* "Chronic evaluation of a clinical system for deep brain stimulation and recording of neural network activity," *Stereotactic Functional Neurosurgery,* 2013;91:220–232 DOI: 10.1159/000345493.