Chapter 4

Animal Studies

4.1 Introduction

A rigorous and automated method is required to select the stimuli applied by the arrays reviewed in Chapter 1 for SCI therapy. Such a method would allow application of these techniques by non-experts, or even autonomously, and additionally would allow for customization to individual patients and their unique, time-varying responses to the stimuli. This dissertation suggests that a variant of GP-BUCB or a similar GP-based active learning algorithm is suitable for this task.

To show that it would be feasible to use GP-BUCB as a learning system for SCI therapy, a closed-loop implementation of the algorithm in real animals was developed. These experiments represent a first step toward a more complex closed-loop implementation in human patients. A somewhat simplified problem was chosen for demonstrating feasibility. A variant of GP-BUCB was used to control an experiment in which a rat was stimulated using an epidural electrode array and the evoked potential in a muscle was measured via EMG. The goal of this experiment was to maximize the amplitude of the resulting evoked potential. While the evoked potential is not a complex motor behavior, this experiment does have many of the important characteristics of the full SCI therapy problem, in particular that the evoked potential varies with the pattern of active electrodes on the array and over the course of the animal's experimental lifetime, and that evoked potentials are critically dependent on the spinal interneuronal circuitry. Showing that the algorithm can successfully control this activity and additionally learn something about the structure of the spinal cord's responses (considered as a function over the space of active electrode configurations) demonstrates a major step toward a therapeutic implementation.

The simplifications inherent in using evoked potentials present a number of substantial advantages for demonstrating feasibility. First, because the evoked potentials represent a relatively low-level function of the spinal interneuron networks, it is reasonable to suggest that they may be less sensitive to the parameter choices than higher-level motor functions, making the search over stimuli inherently easier, appropriate for a feasibility experiment. Demonstrating that the regression models can indeed

capture the important features of an individual muscle response function while using relatively little data means that the responses of single muscles can be modeled effectively using Gaussian processes, plausibly leading to effective models of the high-level behavior based upon combining the predicted responses of multiple, individual muscles. Certainly, it is plausible to create a model which focuses on only high-level phenomena, e.g., user-reported quality of stimuli, but, particularly if physiological monitoring data will form an important component of the response monitoring (desirable in a fully-implanted system), this sort of prediction of high-level quality based upon low-level data is highly desirable. Conversely, if the activity of an individual muscle cannot be effectively captured by a GP, this argues that GPs are inappropriate for modeling the responses of individual muscles and that the system is likely too sensitive to model without exhaustive testing of all potential stimuli (an exponentially large set), suggesting that the full problem is nearly infeasible. The ability to successfully manage a problem like evoked potential optimization using a GP-BUCB-like algorithm is thus a necessary condition for success on the full clinical problem, making this an appropriate first step to applying active learning algorithms to SCI therapy. Second, the EMG recordings arising from a train of stimulus pulses can be temporally separated into the individual responses to each stimulus pulse, meaning that each separate pulse and response may be taken as an individual, independent observation; standing or stepping are much less easily separable into individual "observations." Further, in stepping or standing, it should be expected that successive observations (i.e., blocks of time within a bout, such as strides) would not be independent samples from the same distribution, but instead are highly dependent on their predecessors (e.g., a stumble on stride $n-1$ could reasonably be expected to affect stride n). Third, evoked potentials are naturally expressed as a scalar function of time for each muscle, and easily repeatable scalar measurements (i.e., peak-to-peak amplitude) have already been established for them. In contrast, stepping and standing are complex, high-level behaviors of many muscles, for which no single easily and automatically computed, ordinal measurement has yet been canonically established. There are, however, a variety of measurements which aim to quantify standing performance (Prieto et al., 1996; Santos et al., 2008) or to quantify stepping performance. The latter is generally by either human observation (Basso et al., 1995; Antri et al., 2002) or by automated post-hoc analysis (Fong et al., 2005; Cai et al., 2006). Particularly for measures of locomotor performance based on human-graded observations, it is not clear that the grading scale is ordinal, i.e., while the numerical grade may nominally correspond to quality, these might be more properly thought of as loosely ordered class labels. These label-based grading schemes are often designed to be easy for humans to implement, e.g., using visual features of the stride cycle which are easy to describe semantically, but difficult to describe mathematically, consequently making them very difficult to automate. Further, it is not clear what are "optimal" values of any of these measurements for SCI animals or humans (as opposed to normals), nor is it clear that attaining the nominally optimal value of an individual metric is therapeutically desirable

in either the long-term or short-term. It is critical for the practical success of a bandit algorithm that its reward function and the true utility function correspond closely with one another. If this is not the case and the algorithm converges to the maximizer of the reward function, the true therapeutic utility may not be maximized. The creation of a reward metric which is efficiently computable and faithfully matches optima of the therapeutic utility is a highly non-trivial problem in its own right, such that first demonstrating feasibility for an easier criterion is appropriate.

This chapter focuses on epidural electrostimulation (see Section 2.2.3.1), using flexible, parylene-based electrode arrays, along with simpler wired arrays, in active learning experiments in rats. Section 4.2 lays out the experimental procedures followed in this chapter, with description of the parylene arrays in Section 4.2.2 and the wired arrays in Section 4.2.3. The chosen reward metric, quantifying the evoked potential and thus measuring some aspects of the functional conductivity of the interneuronal network in the spinal cord, is discussed in detail in Section 4.3. Necessary modifications to the GP-BUCB algorithm for this experimental setting are discussed in Section 4.4 and particular choices for the covariance and mean functions are described in detail in Section 4.5. Section 4.6 presents the results of the experiments and a discussion of these results follows in Section 4.7. A few concluding remarks appear in Section 4.8.

4.2 Experimental Methods

Several aspects of the experimental preparations bear detailed discussion; in particular, the preparation of the animals themselves (Section 4.2.1), the parylene-based flexible electrode arrays (Section 4.2.2) and wired arrays (Section 4.2.3) used to deliver the stimulation to the animals' spinal cords, and the basic testing procedures (Section 4.2.4) will all be examined carefully.

4.2.1 Injury, Implantation, and Animal Care

The description below of the surgical and care procedures used in these experiments is largely derived from Gad et al. (2013). These procedures are similar to those developed by the same laboratory for cats and extensively detailed by Roy et al. (1992). All animals used in these studies are adult female Sprague Dawley rats, and approximately 300g in mass at time of implantation. The following procedures are performed on each animal, typically in a single surgery:

1. Partial laminectomy at the T8-T9 vertebral level and complete spinal transection at the T8 spinal segment, including the dura, via microscissors;

2. Placement of gel foam at the site of the transection as a coagulant and separator of the cut ends of the spinal cord;

3. Partial or full laminectomies of some vertebrae (T11, T12, L3, and L4 for animals receiving the parylene arrays, T12, T13, L1, and L2 for those receiving the wired arrays);

4. Implantation of the epidural electrostimulating array (see Section 4.2.3 and Section 4.2.2), inserted using the T11 and L4 laminectomies for the parylene array or the T12 and L2 laminectomies for the wired array, positioned such that the most rostral electrodes are placed in the middle of the T12 vertebral level, and sutured in place to the dura at both the rostral and caudal ends using 8-0 Ethilon sutures;

5. Implantation of one or two ground wires (each composed of 5 braided 0.003 cm gold wires, A-M Systems, Sequim, WA) in the parylene array animals, or one stainless steel, teflon coated wire in the wired array animals (0.304 mm, AS 632, Cooner Wire, Chatsworth, CA). These are placed in the mass of muscles dorsal to the spinal column;

6. Implantation of multi-stranded, Teflon-coated stainless steel EMG wires (AS 632, Cooner Wire) into the bellies of multiple leg muscles, typically left and right tibialis anterior (TA) and left and right soleus (Sol);

7. Attachment of one or two headplug connectors (Amphenol, Wallingford, CT), as required, screwed to the skull and additionally supported with dental cement.

All surgeries are performed under aseptic conditions and with general anesthesia (Isoflurane) delivered via face mask. Analgesia is also provided with buprenex (0.5-1.0 mg/kg, 3 times per day subcutaneously), begun before the end of surgery and continued at least 2 days post-operative. The animals were also treated with Baytril (an antibiotic), administered sub-cutaneously at the end of surgery and at 12 hour intervals thereafter for at least 3 days. The animals recover from anesthesia within incubators and are individually housed both preoperatively and postoperatively, with free access to food and water. The recovery period lasts for one week after surgery (day 7 post-operative, denoted P7), at which point experiments begin. Experiments continue as long as the animal and array both remain viable, or until 6 weeks post-operative (P42). Due to their spinal cord injuries, the animals' bladders must be manually expressed 2-3 times per day and their hind limbs must be manually moved through their range of motion once per day in order to retain joint mobility.

All procedures were in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and were approved by the Animal Research Committee at UCLA.

4.2.2 Parylene Arrays

The parylene-based microstimulating stimulating array is fabricated partially by MEMS techniques and partially using traditional microelectronics in the laboratory of Dr. Yu-Chong Tai of the California Institute of Technology (see Nandra et al., 2011; Gad et al., 2013). The MEMS portion of

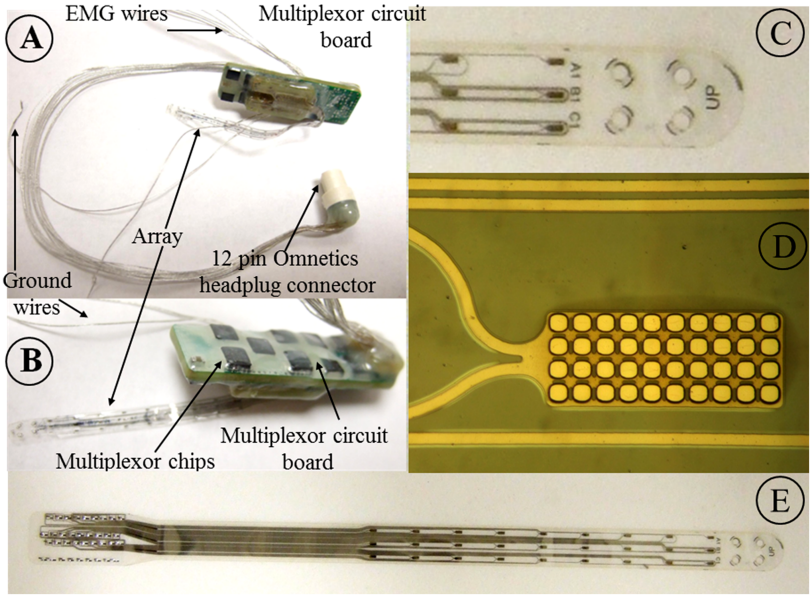


Figure 4.1: Parylene Array Device. (A) and (B): The complete implant, including the main circuit board, parylene electrode array, head plug, EMG wires, and ground wires. (C): Detail of the tip of the parylene array. (D): Detail of a single electrode. The bright, roughly square regions on the surface of the electrode are the open spaces allowing contact with the body; the darker regions defining these squares are the surface layer of parylene, constructed so as to prevent delamination. (E): View of the array, the entire parylene and platinum microfabricated portion of the implant. Figure reproduced under Open Access from Gad et al. (2013).

this device consists of a set of platinum electrodes and traces, embedded in a parylene C matrix. The array device is pictured in Figure 4.1. This construction is sized to the spinal cord of a rat (the parylene and platinum section is 59 mm x 3mm, while the circuit board, mounted dorsally on the spinal column, is 33.2 mm x 10.3 mm), is highly biocompatible, and allows the array to conform to the spinal cord as the animal moves. This last capability is important because, to deliver the same effective stimulus in any of a variety of body positions, the array must maintain roughly the same relative position to the spinal cord, i.e., conform to its movements. The device carries 27 electrodes, each 0.2 mm x 0.5 mm, partially covered by a criss-crossed pattern of parylene to prevent delamination. These electrodes are organized into three rostro-caudal columns, labeled "A" on the animal's left side, "B" on the midline, and "C" on the right. Each column is numbered from 1 to 9 moving caudally; the electrode in the 1 position is at the L2 spinal cord level (T12 vertebral level), the electrode in the 9 position is at the S2 spinal cord level (L2 vertebral level), and the remainder are equally spaced in between. The placement of the array is shown diagrammatically in Figure 4.2. The array is coupled to an implanted circuit board which controls the stimulus, records responses, and communicates with external circuitry through a headplug. A given stimulus is specified by the pairing of a single cathode and single anode from among 29 possible electrodes, the 27 on the array and 2 grounds located distally within the body. Considering only bipolar configurations (i.e.,

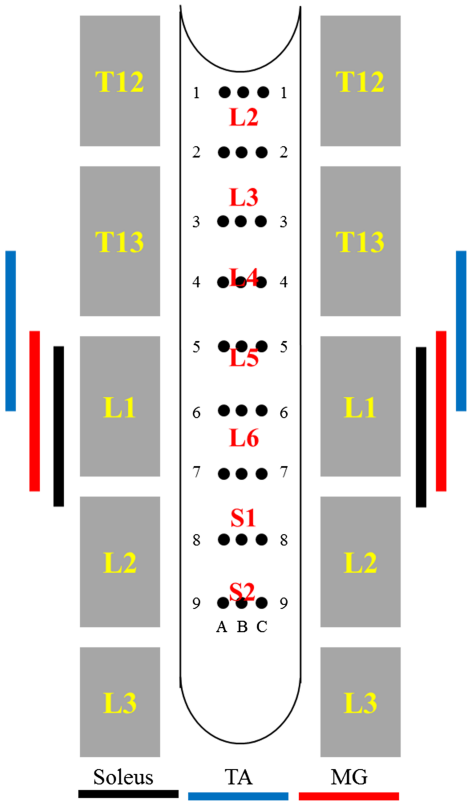


Figure 4.2: Placement of the array device relative to the spinal cord. The gray blocks represent the vertebrae, and are labeled by their conventional numbering scheme. The smaller red text laid over the array denotes the spinal segmental level lying underneath that portion of the array. The colored bars at the far left and right represent the spinal segmental locations of the motor pools of three distal leg muscles, the soleus, medial gastrocnemius, and the tibialis anterior. Figure reproduced under Open Access from Gad et al. (2013).

configurations in which both the cathode and the anode are on the epidural array), there are 702 possible stimulus pairs. Due to the design of the implanted circuit board, 36 of these pairs cannot be stimulated, but the remaining 666 can. The data from two animals tested with this type of array are presented in Section 4.6.2.

4.2.3 Wire-based Spinal Stimulating Arrays

A much simpler array design was also used to test automated stimulus selection. These arrays consist of seven teflon-coated, multi-stranded, stainless steel wires (five are 30 guage, A-M Systems, two are AS 632 wires, Cooner Wire) laid parallel to one another. These wires have openings cut into the insulation at the time of implantation to create exposed electrodes on the surface facing the dura. The wires are then sutured to the dura at the distal end to ensure consistent stimulus location and connected to the headplug on the proximal end. The EMG wires are similarly connected directly to the headplug. This preparation has the significant disadvantage that the number

of stimulating electrodes is reduced to seven, placed in locations corresponding to electrodes A1, A4, A9, B2, C1, C4, and C9 on the parylene array. With only seven electrodes, this device has substantially decreased flexibility of stimuli delivered; there are only 42 bipolar configurations which are possible, approximately 6% of the bipolar combinations possible on the parylene array. However, this technology is highly stable in the animals, providing good performance over a long experimental lifetime, something which is not as yet guaranteed with the experimental parylene arrays. The data from two animals tested with this type of array are presented in Section 4.6.1.

4.2.4 Animal Testing Procedures

For testing purposes, the animals are placed in a vest and harness device which supports their body weight. The device is positioned vertically such that the animal is in a bipedal standing position, with both of the hind feet in contact with a custom surface possessing good traction properties. In the closed-loop algorithm experiments, five bipolar pairs of electrodes, possibly with repetition, are selected by the algorithm for each batch. Each of these pairs of electrodes is used to administer a set of stimulus pulses, delivered at 1 Hz. The pulses are delivered in blocks of 10 (or 20, in some animals) individual pulses at each of several voltages. The EMG signals corresponding to evoked potential responses are recorded using an amplifier (A-M Systems) and custom recording software in LabVIEW (National Instruments, Austin, TX), with a sampling rate of 10 kHz on each channel. The data is then processed using custom software in MATLAB (The MathWorks, Inc., Natick, MA) which calculates peak-to-peak amplitudes of the evoked potential within particular delay windows, specified with reference to the onset of the stimulus pulse (see Table 4.2.4). One combination of stimulus voltage and delay window is pre-selected for use by the algorithm in making decisions, and these evoked potentials are recorded into the algorithm's memory. This cycle repeats, where at each opportunity for the algorithm to request experiments, the algorithm's accumulated memory is used to select the five actions constituting the next batch. On a typical testing day, five such batches are selected. A period of the algorithm's conserved memory of experimental observations is denoted a *run*; each run typically lasts several days, and several runs may be performed on the same animal, with the algorithm's memory wiped in between them.

The batch structure allows interleaving batches of algorithm-commanded experiments with batches of human-directed experiments; specifically, our sessions were structured such that the algorithm competed with a human experimenter in alternating batches, but the algorithm and human experimenter were blind to one another's actions and the responses generated. For decision-making purposes, data were not combined from both sources within runs; however, these two sources were combined to make decisions between runs or between animals, e.g., meta-analysis and tuning of hyperparameters.

Response Period	Start (ms)	End (ms)
Early Response	2.0	5.0
Middle Response	4.5	7.5
Late Response	8.5	11

Table 4.1: Post-stimulus latency windows roughly corresponding to zero, one, or several interneuronal delays within the spinal cord. In these experiments, the algorithm only observes the middle response. Note that using the above definitions, there is a small overlap between the early and middle responses.

4.3 Objective Function

In a UCB formalism (see Section 3.2), it is necessary to have a function which describes a notion of "reward" obtained for any particular action available, and which is also gradually learned from the data acquired; in this application, the reward function is chosen to describe the response of the spinal cord and muscles to the stimulus applied. Specifically, we chose to place the animal in a body-weight support harness and measure the peak-to-peak amplitude of the response of the left tibialis anterior muscle (LTA, a dorsiflexor of the foot) to each individual stimulus pulse, in a latency period termed the "middle response" (MR, 4.5 - 7.5 ms post-stimulus), corresponding to responses which likely involve a single interneuron in the spinal cord synapsing directly onto the motor neuron. These measurements were all conducted with a fixed stimulus amplitude of 5 V (7 V in animal 7), and at a stimulus frequency of 1 Hz, such that the response to each individual stimulus pulse was dissociated from its predecessor and successor in time. This response function was chosen because it provides a short-term, measurable surrogate for therapeutic effectiveness by measuring an indicator of interneuronal function. This is necessary because, while the end-goal is to improve a therapeutic outcome (e.g., standing or stepping), the credit assignment problem of associating this long-term therapeutic outcome with the therapy's constituent stimuli would require infeasibly large cohorts of animals. This sort of wholistic policy selection also does not provide feedback for improving the individual patient's therapy at the moment, nor does it provide individualized policies.

Having settled on an immediately measurable surrogate for the utility of therapeutic stimuli, this work uses a UCB-based algorithm to explore and exploit this response function, such that short- and long-term performance with respect to the reward function are appropriately traded off against one another. This is done via selecting actions which individually are expected to gain high reward (i.e., produce a large evoked potential response in the LTA during the MR latency window), yield a substantial amount of information regarding the performance function as a whole, or possibly both. Under a number of assumptions, including the assumption that the response function itself does not change with time, and with carefully chosen parameters, the GP-UCB, GP-BUCB, and GP-AUCB algorithms (discussed in Sections 3.2.4, 3.3, and 3.4) are all guaranteed to converge to a subset of actions which yield maximal reward, given a sufficient number of actions (See Theorem 1 in Section

3.3.2). In the case in question, if the response function were not changing with time, this result would guarantee convergence to the optimal stimulus with respect to the specified response amplitude, given sufficiently many stimuli. In actual fact, however, the response function is non-stationary, requiring an algorithm which can track these changes (i.e., alter its internal model appropriately as the responses alter over time). In order to solve this problem, the present work employs a modified version of the GP-BUCB algorithm. The problem specification, a careful discussion of the algorithmic challenges, including time variation, and specific modifications required to meet these challenges are discussed in Section 4.4.

In animals 2, 5, and 7, the algorithm was compared competitively with a human experimenter in terms of reward. This experimenter's search and exploitation strategy varied over the course of the several months of experimentation. Anecdotally, the procedure used in the later animals for a typical, five batch day, was as follows:

- In each of the first three batches, the experimenter selected the first three actions on the basis of knowledge of the anatomical location of the electrodes with respect to the distal leg motor pools, combined with observations from earlier batches in the day. These first three actions were typically chosen as "variations on a theme," e.g., polarity swaps or small perturbations to anode or cathode position.

- The fourth and fifth actions in each of the first three batches were selected on the basis of the first three actions in that batch; these were usually chosen to be small variations on whichever (if any) of the first three configurations were successful.

- In the fourth and fifth batches, the experimenter selected actions to explore portions of the cord and array which were deemed less likely to produce strong responses.

The human experimenter never selected an action (i.e., pair of electrodes, cathode and anode) more than once on a day; this presents a confounding factor as far as analyzing the competitive performance of the human versus the algorithm, since the two were acting under somewhat different rules, but this was judged to be necessary for scientific purposes. Further, the human was not restricted to use strictly 10 (or 20) pulses for each action, meaning that the human could administer enough pulses to thoroughly assess the results of any configuration, without requiring a repeated action. This may make a per-pulse examination of amplitudes more appropriate. The human experimenter does have the advantage, however, of choosing actions based on feedback obtained within the batch, i.e., does not operate on the strict feedback schedule used by the algorithm. This might be able to prevent the human from wasting time on costly blunders.

4.4 Modifications to the GP-BUCB algorithm

Formally, the GP-BUCB algorithm maintains a Gaussian process posterior over the decision set D, such that for any stimulus $\boldsymbol{x} \in D$, the reward f (i.e., peak-to-peak amplitude of the MR, 5 V stimulus, 1 Hz) for that stimulus is modeled as a normal random variable $f(\boldsymbol{x}) \sim \mathcal{N}(\mu_{\mathrm{fb}[t]}(\boldsymbol{x}), \sigma^2_{t-1}(\boldsymbol{x}))$ when making the tth decision. The algorithm calculates this posterior based on the series of actions $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t-1}\}$ selected in previous rounds and the observations $\{y_1, y_2, \ldots, y_{\mathrm{fb}[t]}\}$ so far observed. The algorithm then uses this posterior with Equation (3.7) to choose an action $\boldsymbol{x}_t$ for execution in round t. This algorithm is designed such that, with appropriately chosen values of the exploration-exploitation tradeoff parameter β_t, it will converge to the optimal stimulus under the chosen reward criterion for a static (i.e., time-invariant) response function.

Unfortunately, the problem setting does not conform to some of the requirements of the regret bounds currently available for the UCB family of algorithms. Most importantly, the spinal cord's responses are non-stationary. Further, these experiments were required to retain some neuroscientific value in terms of the data collected, rather than simply being an algorithmic exercise. Temporal non-stationarity is discussed in Section 4.4.1 and data value preservation constraints are discussed in Section 4.4.2.

4.4.1 Time Variation of the Reward Function

Of the two deviations from the theoretical setting mentioned above, time-variation is the more fundamental and critical challenge. Both within a session, due to fatigue, and across sessions, the stimulus-response mapping instantiated in the spinal cord is not static, making it essential to consider the variation of the spinal cord's responses in time. These across-session effects are particularly crucial, and may include degradation of the array, changes in its interaction with the surrounding tissue, and, most importantly, the combined effects of recovery and plastic adaptation of the spinal cord. It is the interaction of the algorithm's actions with the course of recovery and spinal plasticity which constitute the essential phenomenon of this therapeutic approach, but these also present a particularly difficult prediction problem for the algorithm's model of the spinal cord's responses. The requirement that the algorithm must model variation of the responses in time means that the algorithm must be able to use information from past observations to predict the current state of the spinal cord. This work chooses to model the response function as a GP which has another free dimension besides the stimulus parameters, the time of stimulation t (in days post-injury). The decision set available to the algorithm at any moment can be thought of as being perpendicular to the t axis in this space, such that the available actions $\boldsymbol{x} \in D$ are constant, but their location in t varies, producing a time-varying decision set D_t. The algorithm must now regress on a function of both stimulus applied $\boldsymbol{x}$ and time t, using data which correspond to some subset of the possible

stimuli and which were obtained at times $t' \leq t$. To do so, the covariance function must now be a map $k : (D \times t) \times (D \times t) \to \mathbb{R}$. Critically, the modeling problem becomes one of extrapolating in time, either forward on the order of days (from one session to another) or just a few minutes (from one batch to the next). This stands in sharp contrast to the nominal setting for the UCB family of algorithms, in which the observations of D become increasingly dense as the algorithm runs, such that the problem is more one of interpolation and the posterior uncertainty is non-increasing.

Because t is constantly increasing, the algorithm must cope with a degree of uncertainty which is increasing (for covariance functions which decay monotonically in time, following the intuitive notion that more recent observations are more useful); that is, the information the algorithm has available will be less and less relevant as time goes on, unless it acquires new observations. Since the posterior uncertainty cannot decrease beyond a finite level, dependent on the rate of change of the response function with respect to the arrival of observations, the algorithm will not ever truly "converge" in the sense of having vanishingly small uncertainty as to which action $\boldsymbol{x}$ in the decision set D is the optimal action at time t. This is similar to the well-known results for Kalman filters (Kalman, 1960) and less well-known results for Gaussian Markov processes (see Rasmussen and Williams, 2006, Appendix B), both of which describe non-zero steady-state limits for the uncertainty under a static observation model, given Gaussian disturbances. In the bandit setting, since the observations available for regression (and thus the reduction of uncertainty) and our actions (and thus the reward obtained) are coupled, this may prove additionally problematic.

However, the UCB formalism is quite robust, and the variation of the response function within the course of a day is not so substantial as to be insurmountable. Further, it is possible to draw substantial inference from observations obtained on previous days, since the day-to-day variation of each individual electrode configuration's corresponding response occurs on a slow timescale, often requiring several days for substantial changes. It is thus possible to be guided by previous days' measurements and to learn deviations from previous days during the course of an experimental session. Additionally, from a practical perspective, it is not necessarily important to find the absolute optimal action at any given time, but rather one which has sufficiently high performance to provide useful therapy. Indeed, some variety in the administered stimuli is appropriate; in robotic gait training, in which spinalized rodents are assisted by a robotic device, Cai et al. (2006) showed that an assistance paradigm which allowed some deviation from the nominal limb trajectory, but maintained appropriate inter-limb coordination, performed better than two competing policies which enforced the nominal trajectory or promoted particular individual limb trajectories without regard for inter-limb coordination. This same phenomenon may hold true for epidural electrostimulation, i.e., the application of a highly specific stimulus pattern with the exclusion of all other patterns may result in poorer therapeutic performance than a more diverse set of stimuli. Nevertheless, some notion of regret is a useful means of understanding what the algorithm is doing, and so we will

continue to discuss the algorithm in these terms.

4.4.2 Redundancy Control and Repeated Observations

The second substantial deviation from the GP bandit setting in these experiments has to do with repetitions of stimuli during an experimental day. It was desirable to extract information from these animals which also had broader scientific value, but the utility of the data set for other purposes primarily depends on sufficient diversity of measurements. A bandit algorithm, on the other hand, should display some form of convergence behavior, i.e., spend many queries on relatively few elements of the decision set D. The required compromise was that the algorithm was restricted from asking for more than two repetitions (generally) of a given electrode configuration on a given day. The number of allowed repetitions for each experimental run is presented in Table 4.5. Since a testing day for the algorithm typically consisted of 5 batches of 5 requests, on a typical day, a minimum 13 different pairs (i.e., distinct actions $\boldsymbol{x}$) had to be requested, forcing some diversity of exploration. This requirement was particularly stringent for the wired array animals, which had only 42 electrode pairs possible, i.e., $|D| = 42$, and thus the algorithm was required to cover some large portion of D each day. Further, as will be discussed in Section 4.7, the evoked potentials appear to be quite sensitive to the position of the anode; the requirement that at least 13 pairs must be used on each typical testing day meant that, for some anodes, every possible configuration using that anode was tested. This, in turn, meant that the algorithm could be forced out of high-reward portions of the decision set and into regions of much lower reward. This presented a number of challenges in analyzing the data, as, by design, convergence in the conventional sense of taking the same action repeatedly was not possible. This topic is discussed in the context of the wire-based animal results in Section 4.7.1.

Another restriction arose for practical purposes; since the practical cost of setting up an individual experiment in terms of experimental time is essentially constant, regardless of the number of stimulus pulses applied, each experiment requested by the algorithm consisted of many successive stimulus pulses (usually 10 or 20), the number of which was known to the algorithm in advance. This required mechanisms for resolving how many observations had been obtained, versus how many had been requested, which further complicated the measurement of regret. Also, if the algorithm requested an action multiple times in the same batch, these observations were executed together (e.g., if 20 stimulus pulses would ordinarily be delivered for a single request of A1_C9, and this action was requested twice in the same batch, the experiment would be set up once, and 40 stimulus pulses would be delivered successively). On an anecdotal basis, this does not appear to have caused substantial fatigue during this sequence of stimuli, but it does present another mild complication in terms of how to analyze the experimental results.

4.5 Kernel and Mean Functions

A number of substantial difficulties are associated with the tuning of the hyperparameters and the associated model selection problem, particularly in the time dimension. Firstly, the data are collected by a biased observer, which selectively samples in regions of high reward, and only rarely samples in regions of low reward. Further, since GP-BUCB is not fully Bayesian, i.e., it chooses actions using a single set of hyperparameters θ and a single model class M (the kernel and mean functions), the set of actions $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}$ selected is itself highly dependent on θ and M. Particularly problematic are the kernel lengthscales, in a fashion analogous to sampling rate in digital signal analysis; if one does not sample sufficiently densely in the dimension one wishes to fit (analogous to sampling at too low a frequency, e.g., less than the Nyquist frequency), one cannot detect the presence of short lengthscales (analogous to high-frequency content), and since the model of the lengthscales present in the response function is precisely what determines what data the algorithm collects, this error could remain undiscovered. It is also possible that unstable behavior might occur in the case where the algorithm is allowed to also adaptively re-fit the hyperparameters, yet the data collection (via GP-BUCB) is not designed to take this procedure into account. Further, note that conditioning the posterior over model classes or hyperparameters on the algorithm used to generate the fitting set is not helpful either; since the interaction of the algorithm with the true system is only through the actions $\boldsymbol{x}$ and observations y (see Figure 4.3), precisely the same data one would use to compute the likelihood, the posterior over the set of hyperparameters Θ or set of model classes $\mathcal{M}$ is independent of the algorithm's internal GP model, given $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}$ and $\{y_1, y_2, \dots\}$. This essentially means knowing which model the algorithm used internally does not help if the acquired data are not informative. This leaves the possibility of using strong priors, but in particularly bad cases, this essentially devolves to hand-fitting the model.

Even in light of the above considerations, some attempts to fit the hyperparameters using the conjugate gradient method[1] on the likelihood or posterior were made. These attempts usually incorporated the human-generated data, which tended to be more diverse and did not have the same set of modeling biases, along with the algorithmically-generated data. Fitting was somewhat successful for the spatial lengthscales, but performed very poorly on the time lengthscales. This poor performance with respect to time lengthscales was most likely because the fit was dominated by the short-time changes (e.g., fatigue and queueing bias due to the redundancy control, discussed in Section 4.4.1) within the experimental sessions (on the order of 1-3 hours) rather than the underlying spinal variation taking place on timescales of days.

Guided by the fitting described above for the spatial lengthscales, careful examination of regression performance on data from earlier animals, and intuition, hyperparameters and kernel functions

[1] minimize.m in the GPML toolbox, Rasmussen and Nickisch (2010).

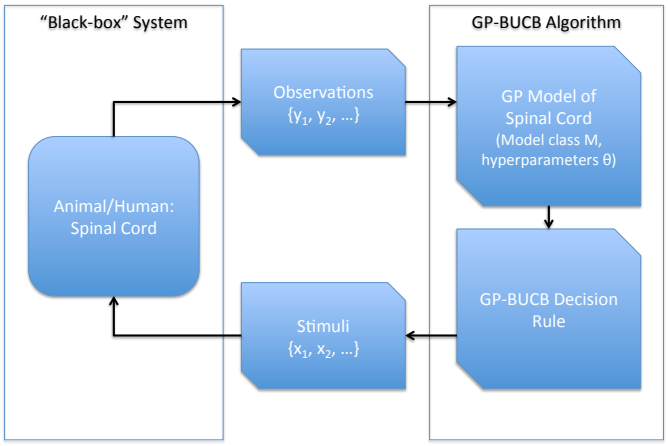


Figure 4.3: Simplified system diagram for the GP-BUCB algorithm interacting with the spinal cord. Given an internal model of the system, fully described by a particular model class M and a particular set of hyperparameters θ, the GP-BUCB algorithm is deterministic, up to tie-breaking. Note that the data likelihood $p(\{y_1, y_2, \dots\}|\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots\}, M^*, \theta^*)$ is independent of M and θ; thus, poor choices of θ and M could result in the collection of a set of data which is not discriminative between various model classes in $\mathcal{M}$, and knowledge of M and θ will not be of any help. This makes post-hoc (or periodic) model selection challenging.

were hand-selected. The kernel functions and hyperparameters used in the animal experiments are presented in Table 4.5. For the first two animals, a squared-exponential (or RBF) kernel, Equation (2.10), was used, with indepedent lengthscales for each dimension. Unfortunately, due to the strong smoothness assumptions implicit in this kernel (the squared exponential kernel implies the Gaussian process is infinitely mean-square differentiable; see Rasmussen and Williams, 2006, Section 4.1.1), problems resulted from intra-day variations in the responses of individual configurations, as well as from long gaps in testing, e.g., weekends. These effects can be seen in Figure 4.16; on day P34 in animal 2, the posterior was badly mis-specified, causing erratic sampling. For further discussion, see Section 4.7.5. In the third and fourth animals, a hybrid kernel $k_h(\boldsymbol{x}_i, \boldsymbol{x}_j)$ was employed, where

$$k_h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_1^2 k_m(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma_2^2 \delta(i, j), \tag{4.1}$$

k_m is a 3rd order Matérn kernel, Equation (2.15), and $\delta(i, j)$ is the Dirac delta function on the indices i, j of the stimuli, not the stimuli $\boldsymbol{x}_i, \boldsymbol{x}_j$ themselves. In essence, this changes the problem from one which is trying to infer the distribution of $f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in D$ to trying to infer $g(i) = f(\boldsymbol{x}_i) + \eta_i$, $\eta_i \sim \mathcal{N}(0, \sigma_2^2)$, a noisy function. Functionally, this means that the uncertainty over $f(\boldsymbol{x})$ never becomes less than a small, positive value, which is useful because it means that short-time variations in the function are subsumed into this noise term, leaving the overall shape to be captured by the

Animal	Run	Dates	Kernel Function	Kernel Hyperparameters	Mean Function	Mean Hyperparameters	Repetitions Allowed
3 (Parylene)	Run 1a	10/3, 5, & 6	covSEard	$l = [0.2740, 0.4659, 0.3297, 0.6300, 1.2018]$, $\sigma = 0.1244$, $\sigma_n = 0.0268$	constant	$c_0 = 0.0$ mV	2
	Run 1b	10/8	covSEard	$l = [0.2740, 0.4659, 0.3297, 0.6300, 1.2018]$, $\sigma = 0.1244$, $\sigma_n = 0.0268$	constant	$c_0 = 0.1$ mV	1
2 (Wired)	Run 1a	11/26 - 29	covSEard	$l = [0.5480, 0.9317, 0.6593, 1.2599, 2.4034]$, $\sigma = 0.1244$, $\sigma_n = 0.0268$	constant	$c_0 = 0.1$ mV	3
	Run 1b	12/3	covSEard	$l = [0.5480, 0.9317, 0.6593, 1.2599, 2.4034]$, $\sigma = 0.1244$, $\sigma_n = 0.0268$	constant	$c_0 = 0.9$ mV	2
	Run 1c	12/4	covSEard	$l = [0.1147, 46.9592, 0.1949, 5.0130, 0.9114]$, $\sigma = 0.5288$, $\sigma_n = 0.1708$	constant	$c_0 = 0.9$ mV	2
	Run 2	12/5-7 & 10	covSEard	$l = [2.9453, 6.3777, 3.4291, 2.3453, 2.4034]$, $\sigma = 0.7903$, $\sigma_n = 0.1364$	constant	$c_0 = 0.9$ mV	2
	Run 3	12/11-14 & 17	covSEard	$l = [2.9453, 6.3777, 3.4291, 2.3453, 2.4034]$, $\sigma = 0.7903$, $\sigma_n = 0.1364$	constant	$c_0 = 1.4$ mV	2
5 (Wired)	Run 1a	1/31 & 2/1	Hybrid	$l_1 = [1.0000, 2.7183, 1.0000, 2.7183, 2.7183]$, $\sigma_1 = 0.2865$, $\sigma_2 = 0.2231$, $\sigma_n = 0.1353$	Linear	$c_1 = 0.08$ mV/day, $c_0 = -0.2492$ mV	2
	Run 1b	2/6, 7, & 8	Hybrid	$l_1 = [1.0000, 2.7183, 1.0000, 2.7183, 2.7183]$, $\sigma_1 = 0.2865$, $\sigma_2 = 0.2231$, $\sigma_n = 0.1353$	Linear	$c_1 = 0.08$ mV/day, $c_0 = 1.7$ mV	2
	Run 2	2/11-15, 19-22, & 26	Hybrid	$l_1 = [1.0000, 2.7183, 1.0000, 2.7183, 2.7183]$, $\sigma_1 = 0.2865$, $\sigma_2 = 0.2231$, $\sigma_n = 0.1353$	Linear	$c_1 = 0.08$ mV/day, $c_0 = -0.1348$ mV	2
7 (Parylene)	Run 1	3/1, 4, & 5	Hybrid	$l_1 = [1.0000, 2.7183, 1.0000, 2.7183, 2.7183]$, $\sigma_1 = 0.2865$, $\sigma_2 = 0.2231$, $\sigma_n = 0.1353$	Linear	$c_1 = 0.08$ mV/day, $c_0 = 0.2513$ mV	2

Table 4.2: The modeling assumptions, e.g., kernel functions and hyperparameters, used by the algorithm to model the responses of animals in our experiments. Numerical designations of runs indicate memory resets of the algorithm, whereas letter suffixes indicate some change to the hyperparameters. For the linear mean function, the mean prediction at any time t in days post-injury is $m(t) = c_1 t + c_0$.

Matérn kernel. The moderately non-smooth nature of a GP with a Matérn kernel is also useful for preventing overshoots or sensitivity to intra-day variation in responses. Note that, while the choice of the 1st order Matérn kernel k_m implies the Gaussian process f is once mean-square differentiable (i.e., it has some degree of smoothness), the GP with the hybrid kernel, g, is not even mean-square continuous (i.e., samples from it can be extremely rough, though this is additive noise on a somewhat smooth trend; see Section 4.1.1, Rasmussen and Williams, 2006). The hybrid kernel proved to be quite successful and robust, as demonstrated in animal 5 by the continuation of runs over weekends, which had been problematic when using the Squared-Exponential kernel.

4.6 Results

A series of experiments in four rats prepared as in Section 4.2 was carried out at UCLA, including 37 sessions and approximately 1200 actions (670 actions selected by the algorithm). In three animals (3, 5, and 7) the experiment continued up until failure of the electrode array. In the remaining animal (number 2), the experiment terminated to allow analysis of the model selection problems described in Sections 4.5 and 4.7.5. The peak-to-peak amplitudes of the evoked potentials for all stimulus pulses recorded, for all four animals, are presented in Figure 4.4. These plots show substantial variation in the evoked potentials over the course of each experiment. In all four cases, they also show a distribution of evoked potential responses which has peak values similar to those generated by the human experimenter. The close matching between the responses evoked by the human and the algorithm indicates that the majority of the variation in responses was not directly a function of the action selection method (human or machine). Instead, the time variation in the evoked potential amplitudes resulting from each stimulus combination appears to be composed of both fluctuations

Animal	Run	Dates	Total Actions	Unique Stimuli Requested	$r_{algorithm}$	r_{human}
3 (Parylene)	Run 1	10/3, 5, 6, & 8	70	43	0.55993	N/A
2 (Wired)	Run 1	11/26 - 29, 12/3 & 4	80	23	-0.012436	-0.17223
	Run 2	12/5-7 &10	55	23	0.46169	0.021927
	Run 3	12/11-14 & 17	100	29	0.48137	0.28783
5 (Wired)	Run 1	1/31, 2/1, 6, 7, & 8	70	28	0.71533	0.030175
	Run 2	2/11-15, 19-22, & 26	240	33	0.92555	0.51492
7 (Parylene)	Run 1	3/1, 4, & 5	55	40	0.61046	-0.039921

Table 4.3: Actions allocated by the algorithm over the course of the run and the number of unique electrode pairs observed by the algorithm during the run. Also included are correlations between the observed mean reward for each configuration (a single shared value for both the algorithm and the human, calculated by averaging all peak-to-peak amplitudes of the evoked potentials observed by either during the entire course of the run) and the number of pulses each of the algorithm or human received for that configuration. Note that the human experimenter only began to deliberately exploit, rather than explore, the reward function after the second run of animal 2.

on a day-to-day basis and substantial, long-term trends. The latter of these may be due to spinal plasticity. Table 4.3 presents a list of experimental runs, the dates of experimental sessions, the total number of actions initiated by the algorithm in each run, and the number of unique stimuli employed.

In the absence of a ground-truth value for the absolute maximum activation of the chosen muscle at any instant in time, the regret cannot be calculated, and so one may instead consider the reward. This quantity is expressed as $\tilde{w}_\tau = y_\tau$ if calculated for each individual stimulus pulse, where τ indexes in the order of individual stimulus pulses and y_τ is the corresponding observation of the response's peak-to-peak amplitude. If the reward is calculated on an action by action basis (i.e., in the fashion in which the algorithm or human makes decisions and requests actions, each composed of several stimulus pulses), the reward for action t is

$$w_t = \sum_{\tau=\tau_{min}(t)}^{\tau_{max}(t)} y_\tau,$$

where $\tau_{min}(t)$ and $\tau_{max}(t)$ are respectively the indices of the first and last pulses for action t. Results are presented in both by-pulse and by-action forms. Reward may also be examined in terms of the maximum value observed so far,

$$\tilde{W}_m(\tau') = \max_{\tau \leq \tau'}(\tilde{w}_\tau)$$

$$W_m(T) = \max_{t \leq T}(w_t),$$

where the maximum is found by comparison among the by-pulse stimulus responses observed ($\tilde{W}_m$)

or among the average responses observed for each individual action (W_m). Note that, at any given time, $\tilde{W}_m \geq W_m$, since the by-pulse maximum amplitude is more sensitive to random variation between pulses administered for a single action than is the average of these individual measurements. The maximum reward is analogous to the minimum regret (see Section 3.2). This quantity in some sense describes the thoroughness of the search over the stimulus space, since a high maximum reward value implies that high-performing stimuli have been found, and thus could conceivably be exploited. This quantity explicitly ignores the number of times high-performing stimuli are visited, so strategies like random search may perform well in terms of maximum reward, while not delivering effective therapy during the same period of time. The reward may also be examined in terms of the average reward so far observed,

$$\tilde{W}(\tau') = \frac{1}{\tau'} \sum_{\tau=1}^{\tau'} \tilde{w}_\tau$$
$$W(T) = \frac{1}{T} \sum_{t=1}^{T} w_t,$$

analogous to the average regret (see Section 3.2). This quantity measures how well the algorithm has traded off exploration and exploitation against one another; a high value for the average reward implies that the algorithm has spent most of its time choosing actions which perform well, yet also explored thoroughly enough to find these high-performing stimuli. In terms of these reward measures, superior performance of one algorithm or method relative to another at a given time index T or τ' is observable as a larger value of that method's maximum and/or average reward plot. Both the per-pulse and per-action results are used in the presentation of results below. The presentation of the results of the animal studies is divided into two; the results for the wired array animals appear in Section 4.6.1 and those of the parylene array animals in Section 4.6.2.

4.6.1 Wire-Based Array Animals: Results

Figures 4.4(a) and 4.4(b) show the responses for all stimuli administered to the wire-based array animals. Maximum and average regret for individual runs are presented in Figures 4.5, 4.6, and 4.7, for animal 2, and Figures 4.8 and 4.9 for animal 5. In both animals, testing included 15 experimental days, with 235 and 310 actions selected by the algorithm, respectively. Although the regions of the stimulus space which yielded strong responses were quite different in size between the two animals (discussed in Section 4.7.2), the algorithm learned the response function well in both. Apart from the final day of animal 2's experiment, tracking by the GP model (i.e., responsiveness to the time-variation in the response function) was sufficient to enable effective decision-making. Both of these animals also exhibited an increase in the level of responsiveness over the course of the experiments. The qualitative shape of this change in responsiveness may be a function of the recovery post-injury,

a function of the spinal plasticity, or both.

4.6.2 Parylene Microarray Animals: Results

Figures 4.4(c) and 4.10 (animal 3) and Figures 4.4(d) and 4.11 (animal 7) show the maximum and average reward results of these experiments. Though much briefer than the experiments in the wire-based array animals (4 days and 70 actions in animal 3 and 3 days and 55 actions in animal 7, rather than 15 days and over 200 actions in the wire-based array animals), the algorithm also found high-performing stimuli in the parylene array animal experiments and exploited them.

4.6.3 Computational Performance

The algorithm was implemented in the MATLAB programming language and run on one of two machines (a MacBook Pro, 2.2 GHz quad-core i7 processor, 8 GB RAM running Mac OS 10.6 and MATLAB R2012a; or an AMD Athlon 64 X2 Dual Core 3800+, 3 GB RAM machine running Ubuntu 12.04 and MATLAB R2011b). Recordings were processed and decisions were made by the algorithm approximately within the time required for the human experimenter to perform a batch of tests, i.e., in minutes, with few exceptions (generally errors in data processing). After having completed the computationally intensive analysis of the raw data, typical times to compose a new batch of actions using the GP-BUCB algorithm were on the order of seconds. Under the most severe conditions which occurred during the experiments discussed in this chapter, with $n = 4432$ individual evoked potential observations available at the end of animal 5's second experiment, computing a hypothetical new batch of five actions to begin the following day took 142.045 seconds of CPU time on the Ubuntu machine described above. The majority of this time was spent on the five executions of the Cholesky decomposition (more than 50 seconds) and five evaluations of the entire kernel matrix (also more than 50 seconds); both of these operations could be changed to execute only once per batch, taking advantage of structural characteristics of the Cholesky decomposition and kernel matrix to append rows and columns with the addition of each observation, rather than completely recalculating, thus saving substantial computational time. In terms of memory consumption, the largest objects in memory are the kernel matrices and the associated Cholesky decomposition results, which are $n \times n$ in size, where n is the number of observations.

Since n is expected to grow with time and both computational time required and memory consumption are critically dependent on n, one reasonable method for controlling computational load would be to "forget" (i.e., delete or not pass to the GP-BUCB algorithm) observations which were redundant by virtue of many more recent observations of the same configuration having been made. In the case mentioned, this could potentially reduce n significantly. It may be reasonable to impose a hardware-based cap on n instead or in parallel. Another reasonable measure would be to treat

average responses to actions as the observations to be fed into the algorithm, rather than individual evoked potential responses. This could potentially reduce n by a factor of $m = 10$ or 20, depending on how many pulses typically correspond to each action, potentially reducing computational time by a factor of as much as m^3, since computation of the Cholesky decomposition scales as $O(n^3)$.

4.7 Discussion

As there were relatively few animals involved in these experiments and the pieces of information which are to be extracted from them are relatively complex, careful dissection of the results from these experiments is necessary. The wire-based array experiments are addressed in Section 4.7.1, with a particular focus on inter-animal comparisons in Section 4.7.2. Section 4.7.3 deals with the experiments in the animals with parylene-based arrays. Finally, Section 4.7.5 focuses on the results obtained with respect to appropriate kernel functions and hyperparameters.

4.7.1 Wire-based Array Animals

Because of the long experimental lifetime of the preparation, the wired array experiments allow exploration of how the algorithm copes with time-variation in the response function. The time prediction problem is inherently one of extrapolation, rather than interpolation. The algorithm must maintain enough uncertainty over how the response function will evolve over time, such that the changes which occur are not unexpectedly large, while conversely limiting this uncertainty such that the model makes strong enough predictions to guide exploitative behavior. In general, the algorithm was successful in terms of future performance prediction; in animal 5, run 2, for example, the algorithm showed strong queueing behavior, i.e., it first selected what were both predicted and proven to be the best stimuli, before being forced by the repetition limit to apply different stimuli. This same run also demonstrated that the algorithm is capable of predicting forward in time over long intervals with no observations, e.g., weekends, as examined in detail in Figure 4.16; after three days with no testing, the algorithm showed the same strong queueing behavior and strong performance (Figure 4.12) as on P31. This problem is discussed more thoroughly in Section 4.7.5.

The major difficulty in analyzing the data from these animals is that the total number of actions the algorithm can request on a given day (typically 25, in 5 batches of 5 actions) is of the same order as 42, the size of the decision set D. Several individual criticisms follow from this fact:

- Assuming that the response function does not vary tremendously in character from day to day, exhaustive testing can be performed in two full days of experiments. Indeed, if a very simple model is considered in which k configurations are selected uniformly at random, without replacement, and there are two classes of stimuli (n of which produce a response, and the

remainder of which do not), probabilities of finding an effective stimulus in a given number of experiments may be calculated. Considering a situation in which two batches are used for searching ($k = 10$) and there is a non-negligible response if and only if a particular anode is picked ($n = 6$), the probability of successfully finding a responsive configuration is over 82%. For $k = 15$, this increases to over 94%, and for $k = 25$ (i.e., a full testing day), over 99%. This may imply that the problem could be solved without needing to make recourse to Gaussian processes as a model of the response function and that a classical bandit algorithm, which does not consider covariance between stimuli, would be sufficient, given modification for capturing time-variation. In particular, this suggests that simply finding a configuration which responds strongly is an insufficient measure of relative success in this setting.

- Because of the restriction on repetitions of any single stimulus, it might be difficult to differentiate between any of the many possible learning algorithms which have approximately the same performance, since those algorithms would most likely choose similar actions.

- It may be that this restriction on repetitions, combined with the small search space, means that the algorithm is forced to collect a sufficient set of data each day to track the time-evolution of the reward function, regardless of the kernel and hyperparameters chosen.

Each of these criticisms is addressed in turn. To the exhaustive testing argument, one may make two counter-arguments with support from the experimental results. First, the GP algorithm employed in this work appears to have effectively employed the structure of its model of the response function. In all five runs on wired array animals, the algorithm found high-performing stimuli in the first day. As discussed above, this is not particularly surprising given the relative size of the search space. In cases where the number of batches was fairly small (e.g., 2, as for both animals on their first day), which typically occurred while the animal initially became acclimated to the experiment, this result at least provides some weak evidence that the algorithm's initial search was well-structured for finding high-performing stimuli. Stronger support for the effectiveness of the algorithm's search may be derived from the fact that the algorithm avoided visiting many of the configurations in the stimulus space. A major difference between classical bandit algorithms and bandit algorithms on structured payoff functions is that, while classical bandit algorithms must eventually visit every action in the decision set, a structured model of the payoff function allows an algorithm to avoid doing so. As shown in Table 4.3, the number of unique stimuli chosen in the wired experiments over the length of any given run was substantially smaller than the size of the decision set; even in the second run on animal 5, in which testing lasted for 10 sessions and 240 actions, only 33 out of 42 configurations were ever selected by the algorithm. This strongly suggests that the structure of the GP model of the reward function enabled the algorithm to avoid exhaustive testing, while still effectively modeling the reward for untested configurations. Additionally, since

the algorithm is competitive with the human experimenter in terms of the maximum reward so far observed, while typically maintaing a better average reward, it follows that the algorithm thoroughly searches the space for the highest reward regions (as demonstrated by the maximum reward), while simultaneously exploiting the gathered data in a more effective fashion. These observations argue that the algorithm is making an effective exploration-exploitation tradeoff, founded on a model which captures the system's behavior.

In response to the second potential criticism, regarding limitations placed on the algorithm, it is true that there is a strong upper limit to the potential performance of the algorithm under these conditions, and that the number of repetitions allowed for any individual stimulus within a daily session strongly influences this constraint. However, the degree to which the algorithm was able to effectively allocate its actions, in spite of the constraints, can be examined. Table 4.3 shows that there was a strong correlation between the number of pulses observed for a given combination of electrodes (a measure of the number of actions allocated to that combination of electrodes for both the human and the algorithm) and the whole-run average response to that combination (retrospectively computed by averaging every peak-to-peak, MR response amplitude observed from both the human and machine experiments in that run). Note that this measure of response strength does not include any notion of normalization of each day's responses; thus, given that the responses generally increased in amplitude over the course of the experiment, it may be expected that poor-performing stimuli would be visited early on and then never revisited, contributing to a low amplitude mean response for such configurations. Conversely, high-performing configurations should be visited consistently and often. It may be, however, that this influence may be mitigated by the combination of data from both agents, along with the limit on repetition within a day, which enforced query diversity. In the case in which the reward function does not change with time, it should be expected that an algorithm which performs well should allocate more queries to actions which provide higher reward. While the relationship between reward and number of pulses observed should not be linear, especially not as the number of observations becomes very large and the function is well known, computing the correlation coefficient provides a gross measure of this discrimination. The correlation coefficient of greater than 0.9 in the case of animal 5, run 2 can be taken as particularly strong evidence that the algorithm is capable of selecting queries in an appropriate priority ordering. In this particular animal, the strong saw-tooth shape of the average reward in Figure 4.9(b) also shows queueing behavior; the best performing stimuli were consistently selected first every day (causing the upswing of the saw-tooth), followed by some collection of other stimuli, most of which were low-performing (causing the down-swing), but which received dramatically fewer queries allocated to them than the highest performing stimuli. One alternate explanation for this saw-tooth behavior is rapid fatigue; if the result of stimulation is a general decrease in response amplitude over the course of a testing day, a set of stimuli could be identified as

high-performing solely because they were selected early in the session, and others could be identified as low-performing due to being selected later and fatigue thus having set in by this time. However, this explanation is not consistent with the data obtained. First, the difference between weak and strong responses is more dramatic than might be expected from a gradual fatigue process, as seen in Figure 4.12(a). Second, the case where the same stimulus (i.e., pair of electrodes) is applied multiple times on a given day can give insight into the relative importance of generalized fatigue. Examining stimuli which resulted in substantially non-zero average response (an average of at least 0.2 mV over the day), the difference ΔV between the average response amplitudes at successive applications is essentially independent of the length of time between the two applications ($r = 0.0101$, $n = 142$ in data from animal 5). The differences in the sizes of the responses to successive applications of the same stimulus are small ($\Delta V = -0.0879 \pm 0.4878$ mV), even over long periods of time; for the 35 such intervals of one hour or more, the differences were -0.0537 ± 0.4295 mV. Were generalized fatigue a substantial factor, it would be reasonable to expect that long inter-application intervals would correspond with large, negative values of ΔV, since these long intervals would imply a substantial difference in time of application within the course of the experimental session, and thus the two instances would have occurred at substantially different points in the fatigue process.

To the third criticism, that the diverse set of data which the algorithm is forced to collect enables better tracking of time variation than would otherwise be possible, a nuanced answer must be given. Clearly, the constraint on repetitions will tend to produce more visits to poorly-performing stimuli than might otherwise occur. Occasional observations of these stimuli confirm that these stimuli remain poorly performing, thus contributing to tracking. Even without the constraint on repetitions, however, these stimuli would likely be revisited by the algorithm eventually due to the chosen kernels' description of temporally distant observations as relatively independent. This argues that the enforced diversity of the current paradigm may not cause a qualitative change in this respect. Additionally, because it is observed that most stimuli which perform poorly continue to perform poorly, re-visiting stimuli which were poor in the past is very unlikely to produce a surprisingly high reward; this argues that the practical value of good tracking on these configurations is relatively small, meaning that the experimental constraints may not cause too large of a performance gain in this respect. With regard to stimuli which are strongly responsive, but are not the absolute best on a given day, it may be that the diversity enforced by the constraint on repetitions may be a substantial aid to the algorithm's tracking; once the algorithm has "converged" to a subset of stimuli which produce strong responses, decisions among these could be strongly influenced by tracking. However, this better tracking may not actually confer a useful advantage with respect to reward because, due to the same constraint, many rankings of this subset of high-performing stimuli will result in the same actions being chosen (up to permutations in ordering).

Day post-injury	r	p	Number of configurations in common
15	0.7885	0.1130	5
20	0.4361	0.1564	12
21	0.6863	0.0197	11
22	0.5812	0.0144	17
23	0.6778	0.0055	15
24	0.2817	0.5405	7
28	0.5051	0.0100	25
29	0.6551	0.0032	18
30	0.4937	0.0270	20
31	0.5242	0.0802	12

Table 4.4: Days in both animals 2 and 5 in which some configurations were tested in common, combining human- and algorithm-commanded experiments for each animal. On many days, strong correlations were present between the responses for a given pair of electrodes across the two animals.

4.7.2 Cross-animal Comparisons

Due to the stability of the wired array implants used in animals 2 and 5, it was possible to collect a large amount of data from these two animals over a substantial period of time. These measurements allow comparison of the performance of individual pairs of electrodes across the two animals. One way to do this is to consider cases in which the same pair of electrodes (e.g., A1_A9) was tried in both animals on the same day post-injury. Some such comparisons are shown in Table 4.4 and Figure 4.15. While from only two animals, these data demonstrate that there is some fairly substantial repeatability between animals with respect to the strength of evoked potential elicited by a given stimulus on a given day post-injury. This appears to be particularly true for the highest-performing stimuli in animal 5, those with anode A9. It is interesting to note that animal 2, while also highly responsive to stimuli using A9 as an anode, had a larger set of effective stimuli than animal 5 (see Figure 4.15).

4.7.3 Parylene Array Animals

The analysis of data from the parylene array animals is primarily of interest as a means of assessing how well the algorithm can search the space D of electrode combinations for effective stimuli $\boldsymbol{x}$. Because this space is quite large (666 elements) relative to the number of combinations which can be tested on a given day (no more than 25), this is expected to be a challenging problem, requiring strong assumptions regarding the shape of the reward function $f(\boldsymbol{x})$ (i.e., the stimulus-response mapping) over the search space.

Both of the parylene array experiments were fairly brief; in both cases, the devices ceased to function at approximately two weeks post-injury. By comparison, the experimental lifetimes of both wired animals were roughly five weeks post-injury. Thus, it was particularly important for the algorithm to make intelligent choices with the few actions it had. It should be noted that the

amplitudes of the peak-to-peak responses observed were somewhat smaller in these animals than in animal 5, though they were approximately the same as in animal 2.

In spite of the brevity of these experiments, it appears that it is possible to make some assertions about the actions of the algorithms in these cases. In both animals, after finding a configuration (C8_A9 in animal 3, C6_A9 in animal 7) which produced strong responses, the algorithm moved sharply to exploit this configuration, allocating many double queries to its neighbors. Both of these configurations include an anode at the left, caudal corner of the array (the same region which worked well in both of the wired animals), which indicates that this could plausibly be the region of strongest response (and thus highest reward) in the space. Though not all such configurations were effective, the algorithm does appear to have been able to explore and exploit the structure of the response function appropriately. Of particular note is the fact that the algorithm was able to overcome the flat prior it had been given and find this consistent anatomical pattern, even with so few queries; the key observation which allowed this exploitative behavior occurred in only the 3rd batch for animal 3, and the 6th batch for animal 7. The fact that this search was so efficient and could be effectively exploited by selecting neighboring configurations argues that the structure of the GP model is providing a benefit over what would be possible with conventional bandit algorithms.

While no cross-comparison to a human experimenter was made in animal 3, in animal 7, interleaved batches were selected by the human and algorithm. Possibly due to anatomical prior information, the human experimenter was able to effectively find strongly responding stimuli on the first day of testing in animal 7, despite only having two batches. However, later in the experimental period, in both animals 3 and 7, the algorithm found multiple high-performing stimuli. In the final day of animal 7, for example, the responses were similar in magnitude to those produced from the human-commanded experiments, and there were more successful stimuli. Figure 4.14 shows this result.

4.7.4 Therapeutic Relevance

While it is very difficult to disentangle the long-term effects of the generally intensive training protocol, the specific stimuli chosen by the human, and those chosen by the algorithm, it is possible to examine the immediate responsiveness of the rat's spinal cord and muscles to those stimuli. This work has assumed that stimuli which elicit strong responses are therapeutically useful, whereas those which produce little to no muscle activity are not; using a threshold twitch strength to explicitly divide stimuli in such a fashion may provide insight into which agent is better at usefully allocating therapeutic actions. This all-or-nothing division of stimuli into effective or not stands in contrast to the notion of reward, used extensively in the preceding discussion. In particular, reward is sensitive to outliers, possibly to a degree which is not reflective of actual therapeutic performance, since fine differences in twitch strength between stimuli which produce strong responses can have effects on the

Animal and Run	Agent	Threshold (V)							
		0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
3 (Parylene)	Algorithm	0.229	0.157	0.100	0.043	0.000	0.000	0.000	0.000
	Human	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2 (Wired), Run 1	Algorithm	0.662	0.450	0.263	0.000	0.000	0.000	0.000	0.000
	Human	0.440	0.253	0.067	0.013	0.000	0.000	0.000	0.000
2 (Wired), Run 2	Algorithm	0.945	0.727	0.545	0.091	0.000	0.000	0.000	0.000
	Human	0.720	0.520	0.280	0.040	0.000	0.000	0.000	0.000
2 (Wired), Run 3	Algorithm	0.910	0.850	0.790	0.560	0.350	0.110	0.020	0.000
	Human	0.828	0.697	0.626	0.424	0.192	0.081	0.040	0.000
5 (Wired), Run 1	Algorithm	0.657	0.543	0.357	0.214	0.186	0.129	0.057	0.000
	Human	0.564	0.418	0.273	0.127	0.109	0.055	0.018	0.000
5 (Wired), Run 2	Algorithm	0.779	0.642	0.442	0.308	0.229	0.167	0.142	0.075
	Human	0.685	0.477	0.285	0.153	0.106	0.081	0.068	0.026
7 (Parylene)	Algorithm	0.255	0.091	0.036	0.000	0.000	0.000	0.000	0.000
	Human	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 4.5: Proportion of stimuli yielding satisfactory responses, at any of several different thresholds for therapeutic twitch strength. The algorithm produced a higher proportion of responses above each achieved threshold, for each animal tested competitively, with the exceptions of Animal 2, Run 1, 2.0V and Run 3, 3.5V.

average or maximum reward measures used above, but may not be of any functional relevance. A comparison based on thresholded average twitch amplitude elicited by actions is presented in Table 4.5. Note that the algorithm was able to allocate its actions such that a higher proportion of them received responses satisfying nearly all of the thresholds examined; if, as posited above, therapeutic effectiveness is a function of the proportion of stimuli which elicit a sufficiently strong response, then this result indicates that the algorithm is more effective in this regard than was the human expert's chosen strategy.

4.7.5 Kernels and Hyperparameters

For the Gaussian process model to capture the responses of the muscles faithfully, yet allow rapid learning, it is crucial that the kernel function, mean function, and their respective hyperparameters be well chosen, such that the Gaussian process prior matches fairly well with the structure of the data actually measured. If these modeling choices are made well, the algorithm will likely perform well, but poor choices can have pathological behavior, especially with respect to the time variation of the responses. Figure 4.16(a), showing the GP posterior at the beginning of day P34 of animal 2's experiment, provides a good example of what can happen under poor modeling assumptions; the GP model gave predictions which produced a query pattern unreflective of the structure of the response function. This performance is clearly dependent on the properties of the kernel function and mean function with regard to changes over time, but there exist kernels and mean functions for which this problem can be resolved, as demonstrated in Figure 4.16(b).

Since active learning experiments using GP models of the spinal cord's responses had not been

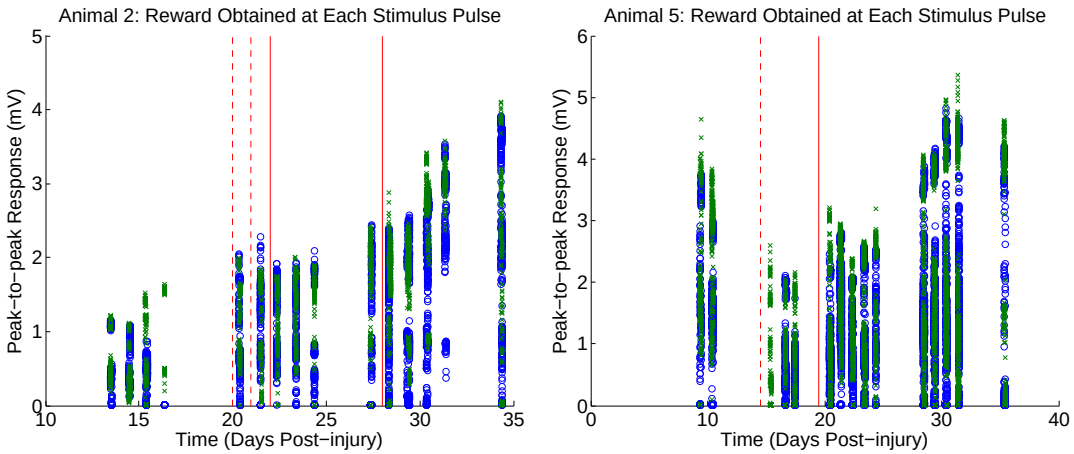
conducted before, it was critical to periodically re-examine the kernel, mean, and hyperparameters. After possible re-fitting during each animal's first run, subsequent hyperparameter re-fitting and/or kernel re-selection in that animal were accompanied by a wipe of the algorithm's memory; Table 4.5 and Figure 4.4 show these fitting events and memory wipes in more detail. It is important to note that the choice of when to re-fit or wipe the algorithm's memory was made by the human inspection, rather than by the algorithm; in this way the algorithm was in some sense protected from making prolonged, catastrophic errors. This does introduce some bias into the data, because the algorithm was not allowed to act nonsensically, but this was necessary to preserve the value of the experiments. Further, this did not confer a substantial advantage upon the algorithm over the human, because the human experimenters are constantly making just such checks for sensibility, e.g., using the responses elicited by the chosen stimuli to detect failures of the stimulating hardware. Many of these sorts of checks could and should be built into eventual implanted stimulators, but were beyond the scope of this work.

While the hybrid kernel, Equation (4.1), was successful for the experiments on animals 5 and 7, argument can be made that other kernels might be more appropriate. Two possiblities are fairly strongly suggested by the experience in these four animals. First, the noise term in the hybrid kernel could be replaced by a covariance term which gives covariance only between measurements of the same configuration on the same day, but otherwise treats observations as independent; this amounts to an assumption that there exists a hidden additive variable for each configuration on each day. Second, it may be that some linear kernel in time is reasonable, provided the algorithm has some "forgetting" of very old observations; this would somewhat account for the fact that, for anatomical reasons (which are thus the same from day to day), non-responsive configurations tend to remain non-responsive over the course of the animal's experimental lifetime.

4.8 Conclusions

From the four animals examined, it can be concluded that the algorithm is effective for selecting stimuli to maximize a simple experimental objective, consisting of the evoked potential amplitude during the middle response period (4.5-7.5 ms latency with respect to stimulus pulse onset), over the set of pairs of stimulating electrodes. Further, from the three animals in which comparisons with a human expert were considered, the algorithm achieves action-averaged reward which is superior to that achieved by the human experimenter, while matching the human experimenter's performance in terms of maximum reward. This indicates that the algorithm is able to allocate more actions to exploiting high-performing stimuli while still matching the human experimenter's effectiveness in finding the best stimuli. These results provide a strong indication that the GP model used by the algorithm effectively captures the variation of evoked potentials, that the algorithm's decision

rule effectively trades off exploration and exploitation, and that doing so enables the algorithm to provide effective stimulation in terms of maximizing evoked potentials. While this problem is substantially simpler than the related problem of maximizing standing or stepping performance, the capability demonstrated in these animal experiments, effectively modeling and exploiting the responses of individual muscles, is critical for fine-tuning these high-level behaviors.



(a) *Animal 2 (November)* (b) *Animal 5 (February)*

(c) *Animal 3 (October)* (d) *Animal 7 (March)*

Figure 4.4: Peak-to-peak amplitude (mV, reward) of all individual stimulus pulses for each animal, combined across runs. Blue circles: evoked potentials obtained by the human experimenter; Green 'x': those observed by the algorithm. Solid red lines denote a wipe of the algorithm's memory (thus dividing separate runs), with or without changes to the hyperparameters, while dashed lines denote changes to the hyperparameters without a memory wipe. The human experimenter and algorithm were not privy to each other's actions or the resulting rewards. They (typically) executed their actions as interleaved batches of experiments.

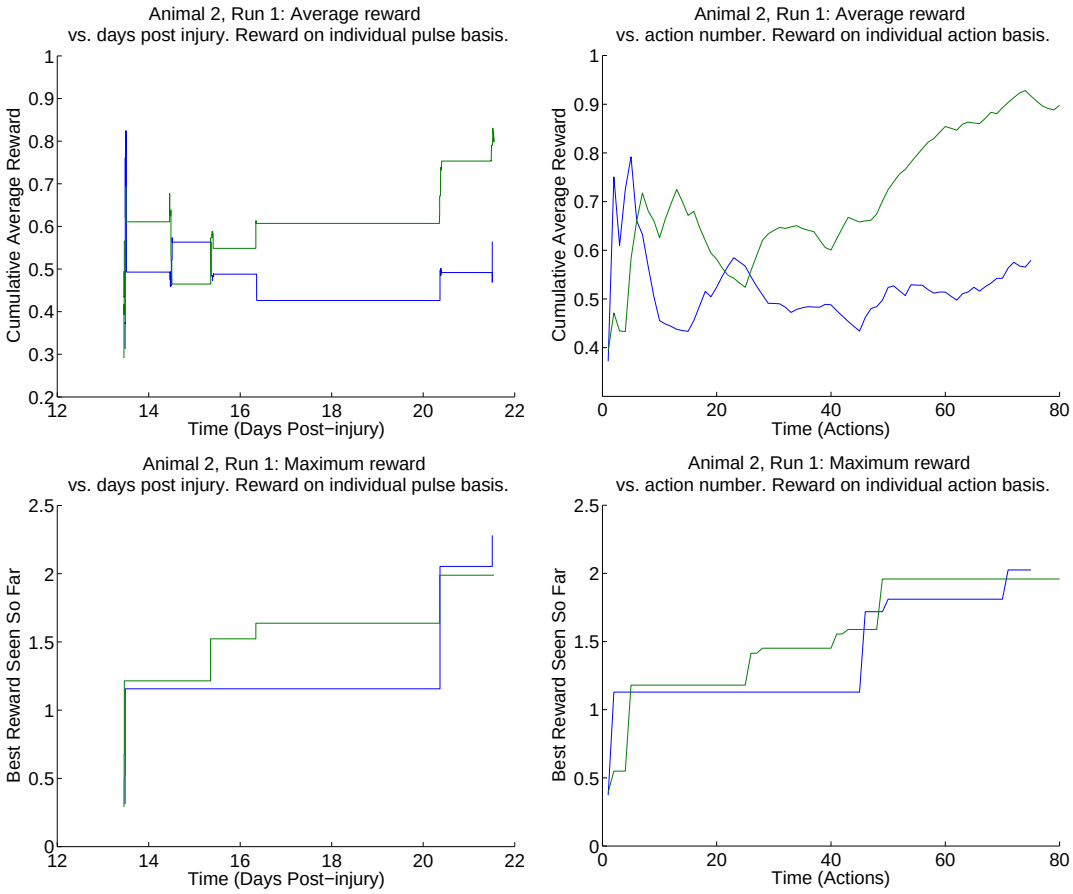


Figure 4.5: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 5V stimulus at 1Hz, in mV) in Run 1 of animal 2, the first wired array animal and second animal tested. This was the first competitive experiment between the algorithm and human. Average reward measures the algorithm's tendency to consider the reward generated by *every* action, rather than just the best; the behavior of the best action vs. time is captured by the maximum reward. Note that the algorithm's average reward is typically superior to that of the human experimenter, while the algorithm also maintains superior or competitive maximum reward, as of similar time or action index.

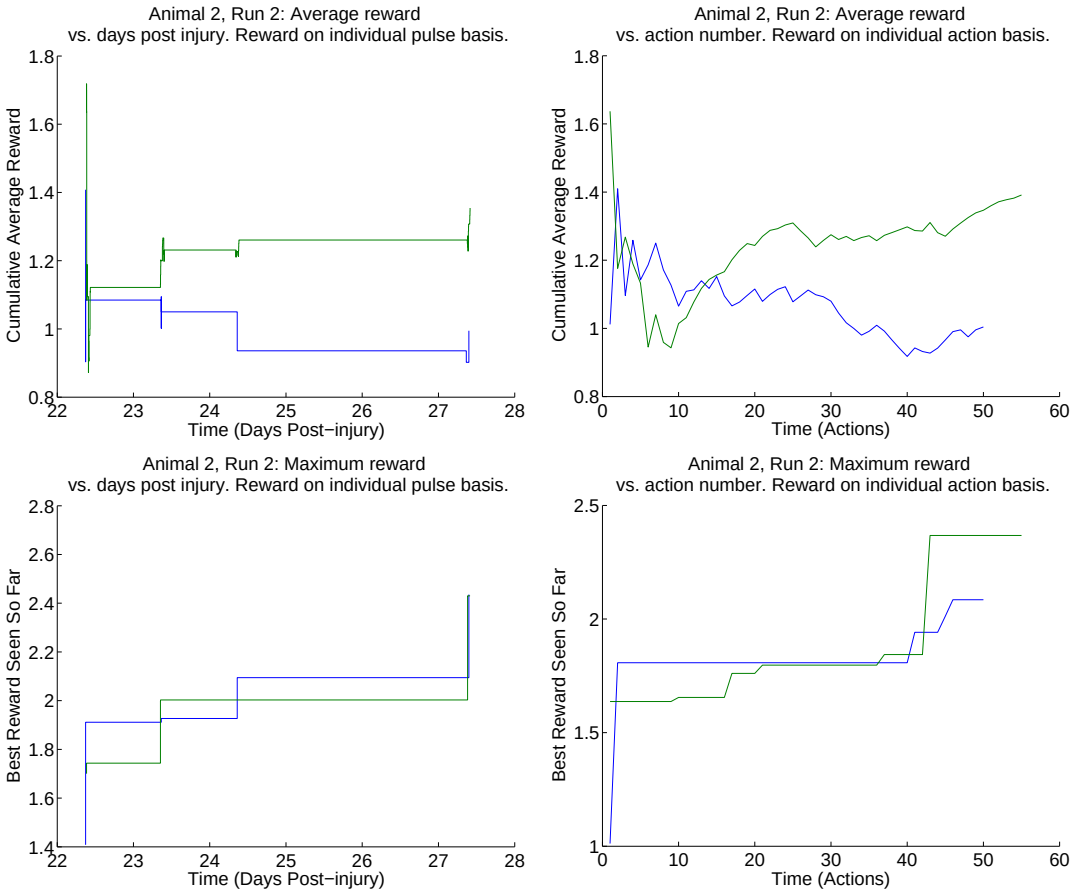


Figure 4.6: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 5V stimulus at 1Hz, in mV) in Run 2 of animal 2. The algorithm's average reward is again typically superior to that of the human experimenter, while maintaining competitive maximum reward, as of similar time or action index.

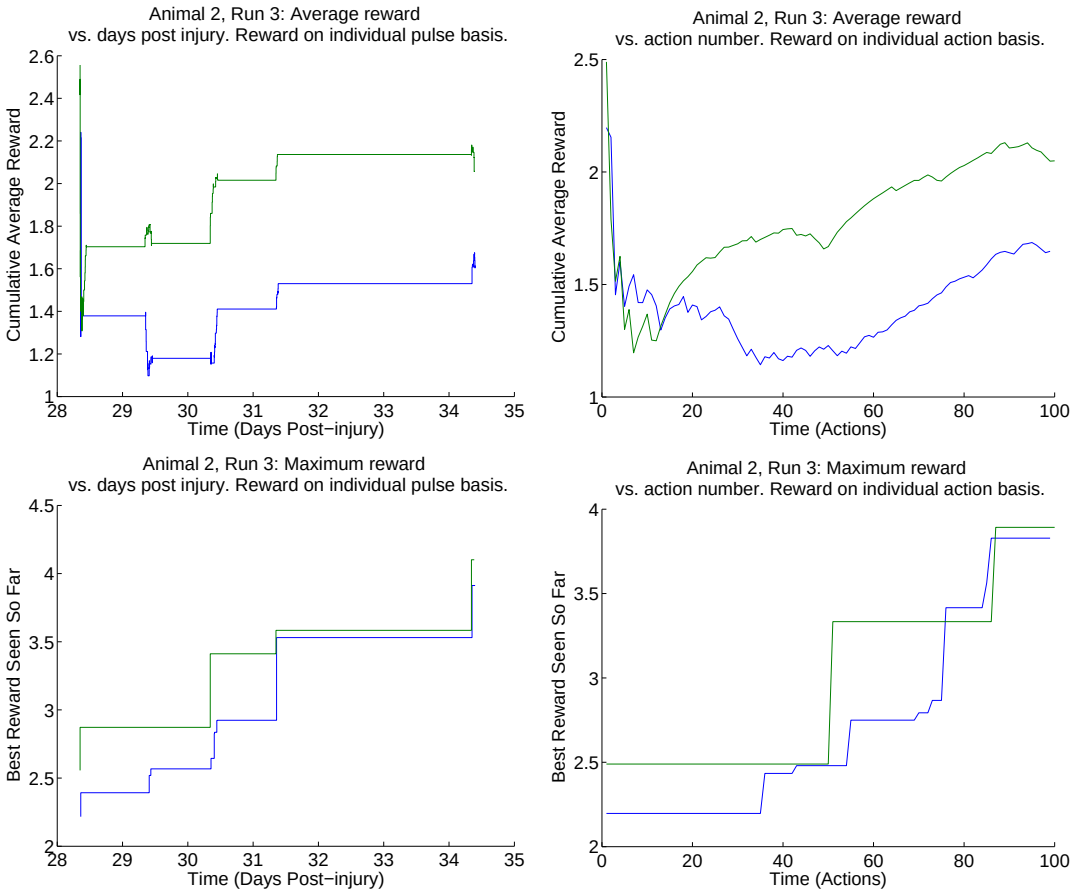


Figure 4.7: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 5V stimulus at 1Hz, in mV) in Run 3 of animal 2. Once again, the algorithm's average reward is typically superior to that of the human, and the algorithm's maximum reward is competitive or better.

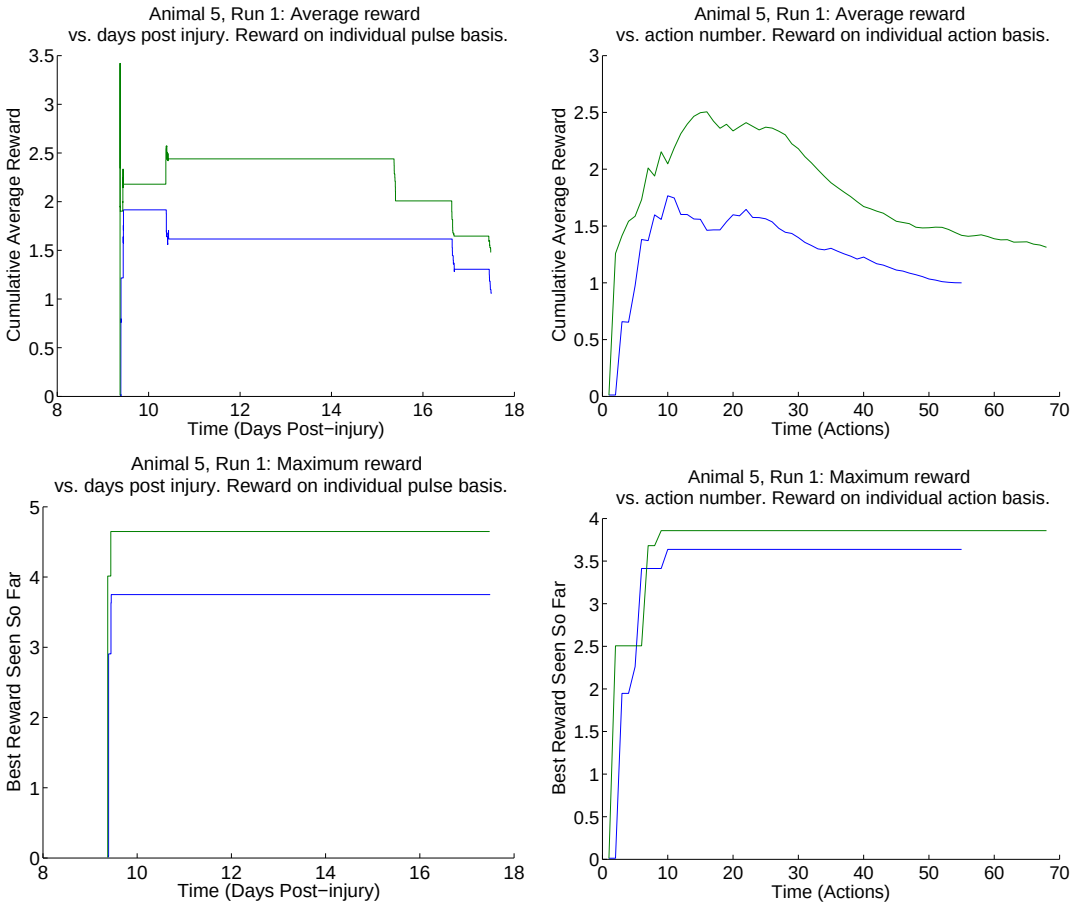


Figure 4.8: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 5V stimulus at 1Hz, in mV) in Run 1 of animal 5. For animal 5, 20 pulses were delivered per action. The algorithm shows substantially stronger performance than the human experimenter in both average and maximum reward. Note that the human experimenter did not execute any experiments on the 15th day post-injury (P15), whereas the algorithm was able to execute three batches, or 15 stimuli. Alternate versions of the action-based reward plots are presented in Appendix C. P15-17 all showed substantially reduced evoked potential amplitudes relative to days P9-10, as visible in Figure 4.4(b), producing the decline in average reward apparent in (a) and (b).

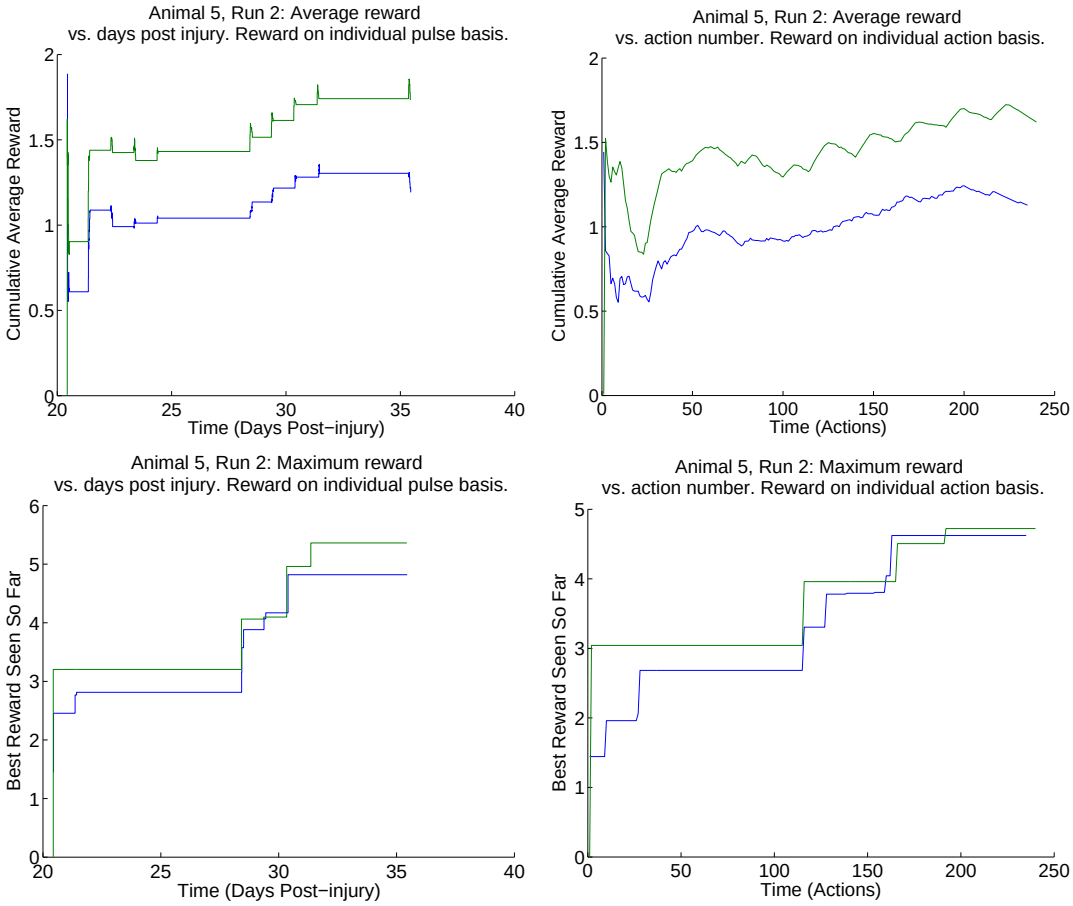


Figure 4.9: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 5V stimulus at 1Hz, in mV) in Run 2 of animal 5.

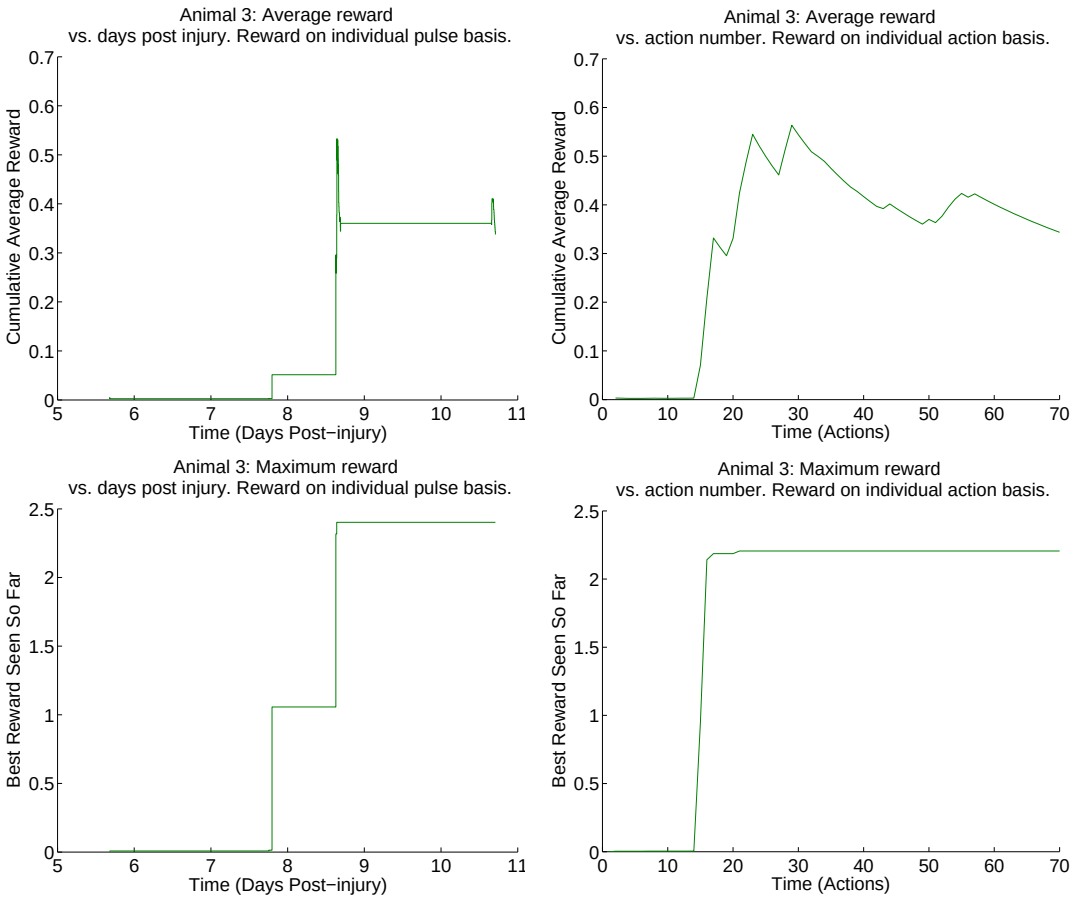


Figure 4.10: Algorithm's reward (Peak-to-Peak amplitude of MR evoked by a 6V stimulus at 1Hz, in mV) in animal 3, the first parylene array animal and the first fully-closed-loop experiment conducted in this work. In this experiment, the first conducted, only one batch of actions (5 individual stimulus combinations) was conducted on day P5, producing the apparently poor performance until part way through the second session, on the evening of P7.

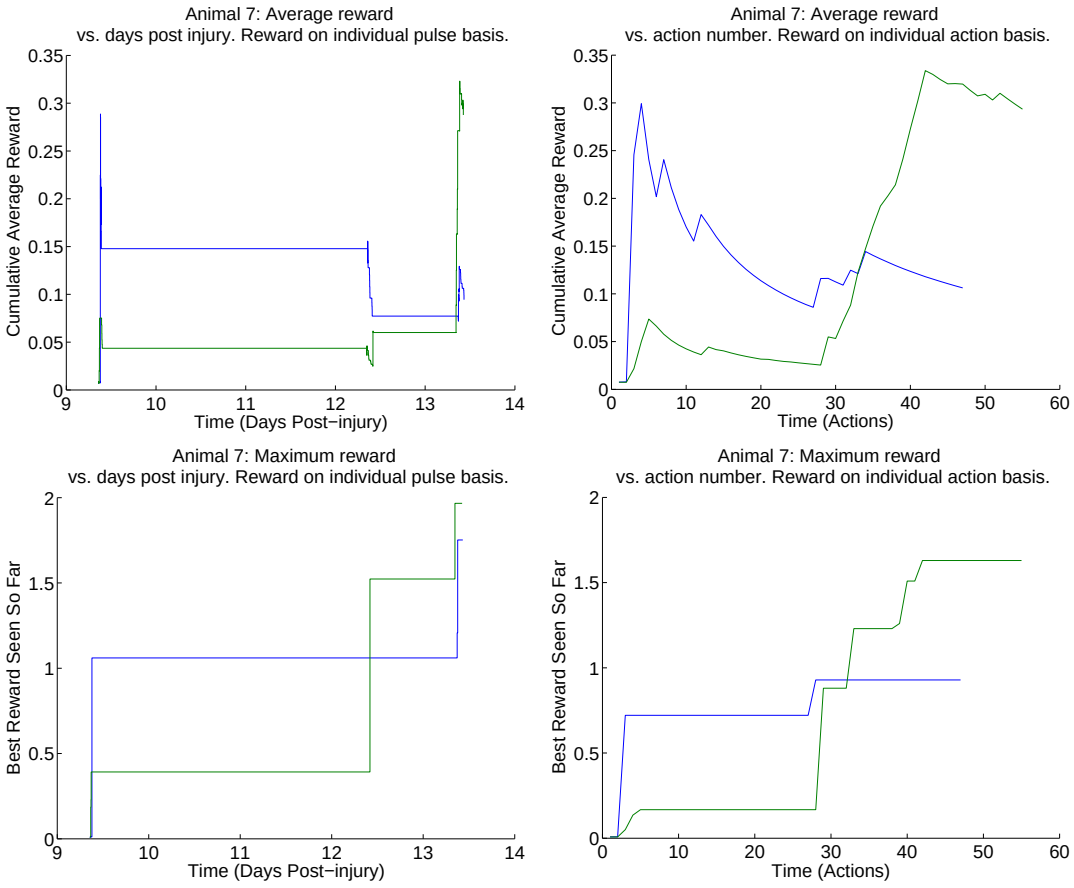


Figure 4.11: Human experimenter's (Blue) and Algorithm's (Green) reward (Peak-to-Peak amplitude of MR evoked by a 7V stimulus at 1Hz, in mV) in animal 7, the second parylene array animal. For animal 7, 20 pulses were delivered per action. Note that on day P9, both the algorithm and human were limited to two batches, due to the animal's unfamiliarity with the training paradigm. This meant that the algorithm only received feedback from the first batch for the purpose of making decisions on P9. Similarly, on P12, testing ended before the fourth batch by the human experimenter. Alternate versions of the action-based plots, which compensate for these missed actions, are presented in Appendix C. While the human experimenter found three configurations on the first day which produced relatively strong responses, the algorithm did not find any such responses. This not surprising, given the flat prior of the algorithm and the size of the search space (666 pairs) as compared to the 10 stimuli administered. The algorithm first found a relatively strong response (from configuration C6_A9) within the last batch of day P12. On P13, the algorithm made choices to heavily exploit neighbors of this configuration throughout batches 1, 2, and 3. During batch 5, and to a certain extent, during batch 4, the algorithm resumed exploration of configurations which bore little resemblance to C6_A9.

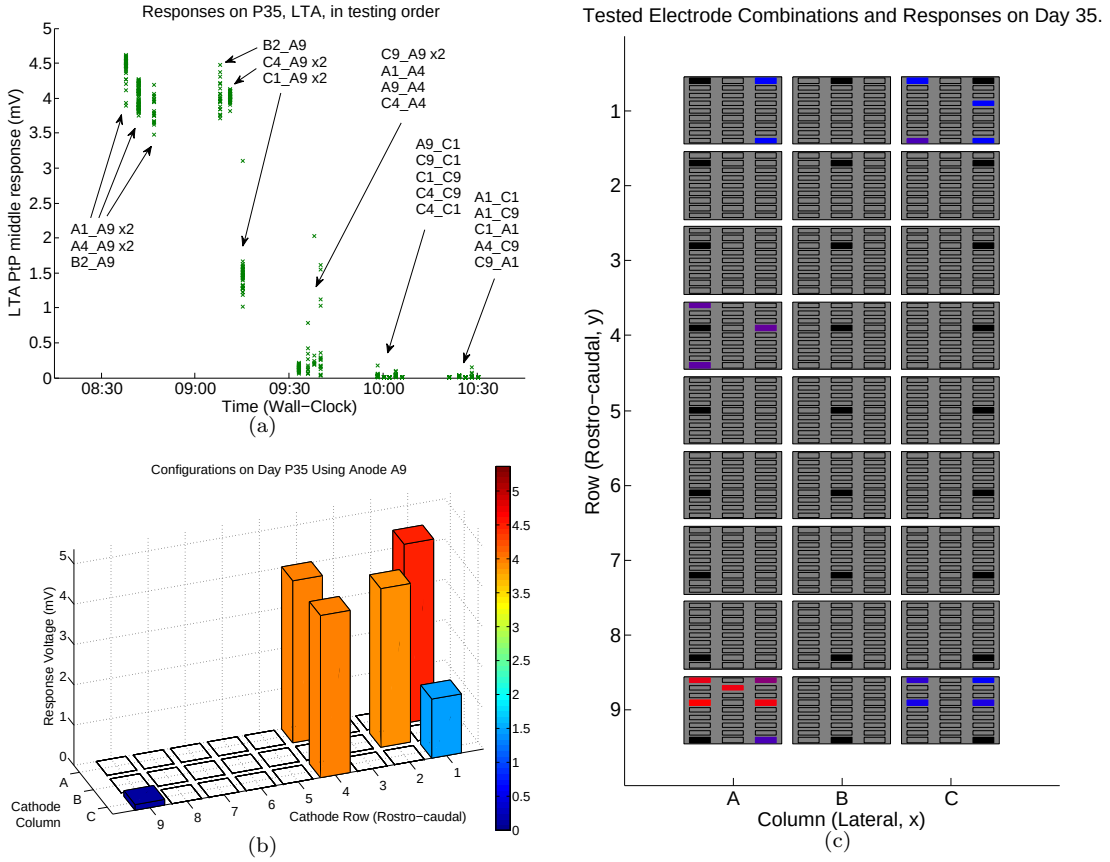


Figure 4.12: Left TA middle responses (mV, peak-to-peak amplitude) to algorithmically requested stimuli, animal 5, run 2, day P35. (a): Responses shown according to the time of commencement. Each point corresponds to a single evoked potential, i.e., a stimulus pulse and response. Note that the algorithm requests stimulus configurations in batches of 5, possibly including double repeats of some actions, and that the algorithm chose stimuli in the approximate order of their efficacy. When all six options (A1, A4, B2, C1, C4, and C9) to pair as cathodes with anode A9 are exhausted, the algorithm is then forced to use other anodes. The sensitivity of the responses to changing the anode suggests that in future experimental preparations, greater density of potential anodes near A9 is required. (b): Responses to configurations with A9 as the anode, averaged over all individual pulses for each configuration. Both color and bar height designate amplitude of response, in mV. The rostral cathodes generally paired effectively with A9 as an anode, while the combination C9_A9 was ineffective. This may suggest that the broad rostro-caudal region of stimulation is important, or that a small, but crucial target for stimulation is rostral to the 9 row and caudal to the 4 row. (c): Relative strength of pulse-averaged peak-to-peak responses, shown with respect to spatial location of the stimulus on the cord. The large boxes show anode location and the smaller boxes within the anode boxes correspond to cathode location; the extreme lower left box corresponds to (b). Red corresponds to the strongest responses seen on P35, blue to the weakest (nearly 0 mV), and purple to intermediate response strength. This pattern shows a relatively diverse search, combined with exploitation of configurations with A9 as the anode. Note that configurations using C9 (on the right side of the spinal cord, and extreme caudal end of the array) and a rostral cathode failed to elicit strong responses; the middle response in the TA appears to be quite sensitive to the lateral location of the anode as well.

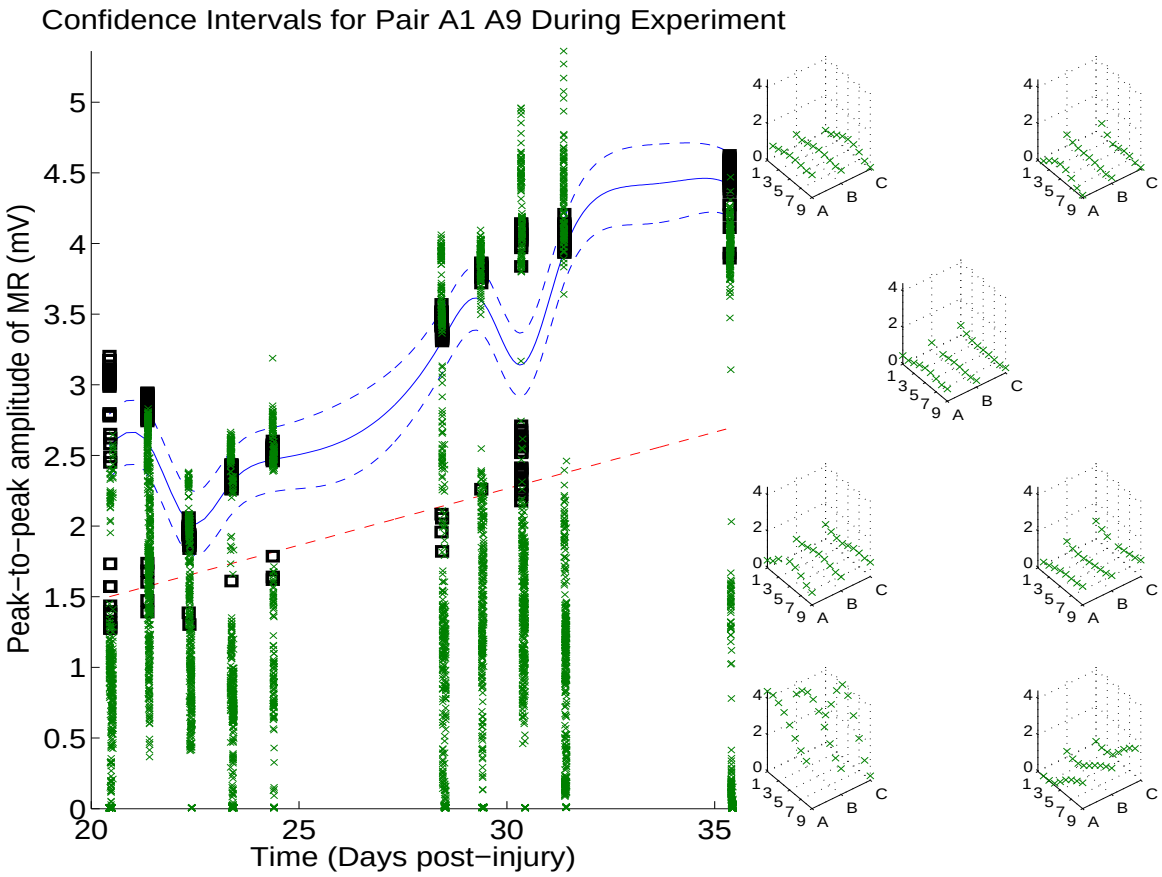


Figure 4.13: Retrospective over the entirety of run 2 for animal 5, focusing on algorithmic predictions for the stimulus pair A1_A9 during the experiment. These retrospective plots were used as part of the process for hand-fitting the covariance functions. The left panel shows the mean (solid) and ± 1 standard deviation confidence intervals (dashed) for the Gaussian process posterior over the response function $f(\text{A1,A9},t)$, with respect to the time t post-injury. A black square denotes an observation of a response evoked by the pair of interest, A1_A9, and a green 'x' denotes an observation of a pulse for any other configuration. These other configurations may be more or less "distant" from A1_A9 in terms of their covariance under the specified kernel function. The prior mean of the entire GP with respect to time (which is invariant to the stimulus configuration) is shown as a red, dashed line. The right panel shows the predicted spatial mean function 5-6 minutes after the last observation, roughly when another batch could have begun. Each subplot corresponds to a possible anode. These anodes are A1, A4, and A9 descending the left column, B2 in the center column, and C1, C4, and C9 in the right column. Within each subplot, the isometric views show variation of the predicted mean peak-to-peak response (vertical axis, mV) over the cathode location (rostro-caudal on the left side, lateral on the right). All cathodes (i.e., A1-A9, B1-B9, C1-C9) are shown for ease of visual assessment, even though this animal has implanted with a wired array and thus only those locations listed above for the anodes were available.

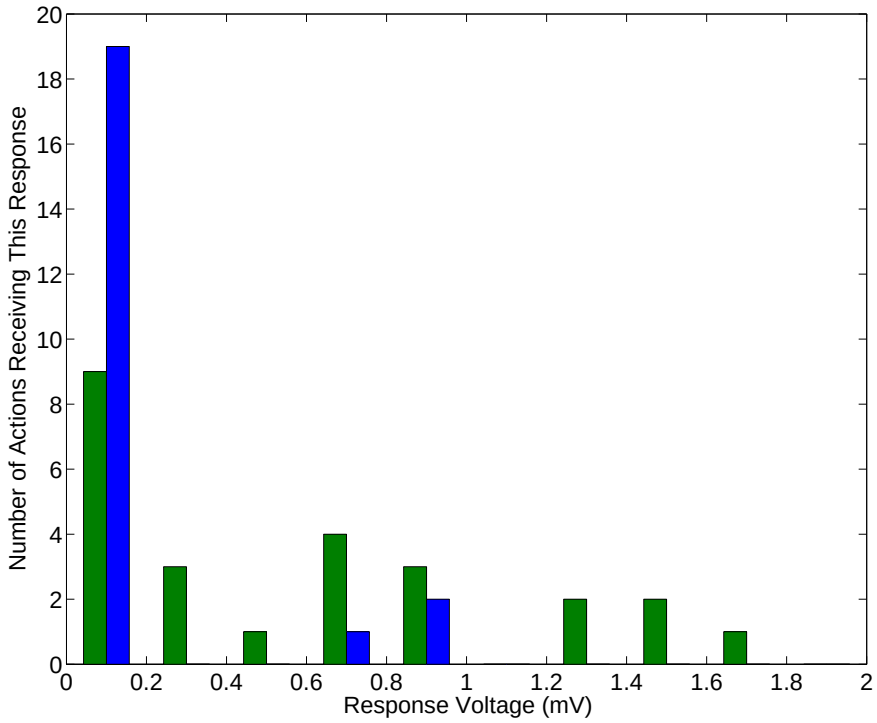
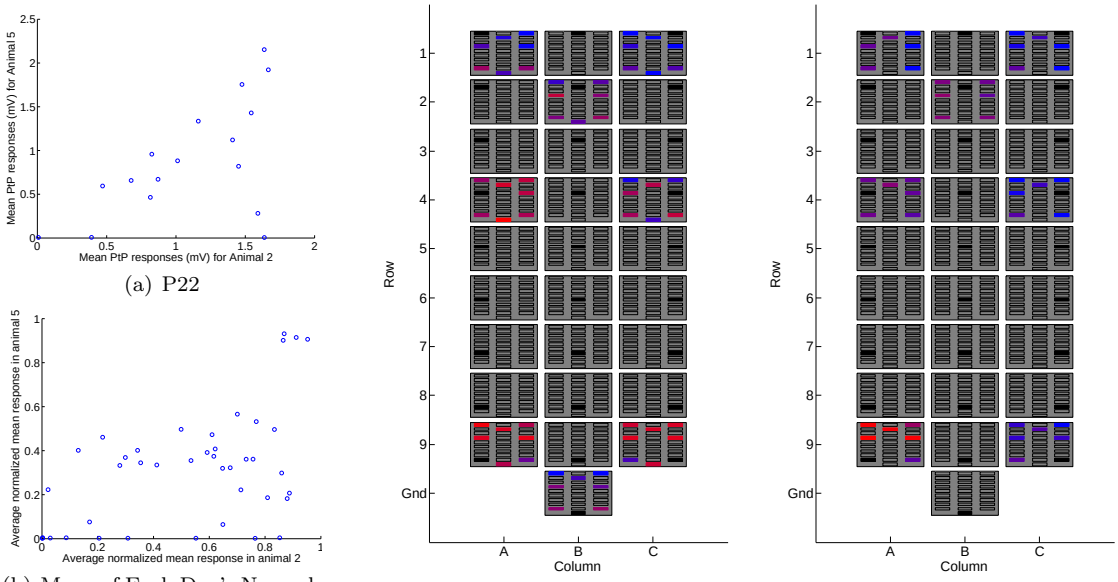


Figure 4.14: Rewards (peak-to-peak evoked potential amplitudes during the MR period) obtained by the algorithm and the human experimenter on day P13 of animal 7's experiment. The algorithm initiated 25 actions (17 unique pairs of electrodes) and the human initiated 23 actions (all unique). Note that the algorithm devoted substantially more actions to exploiting strongly-responding stimuli, and found several such stimuli.



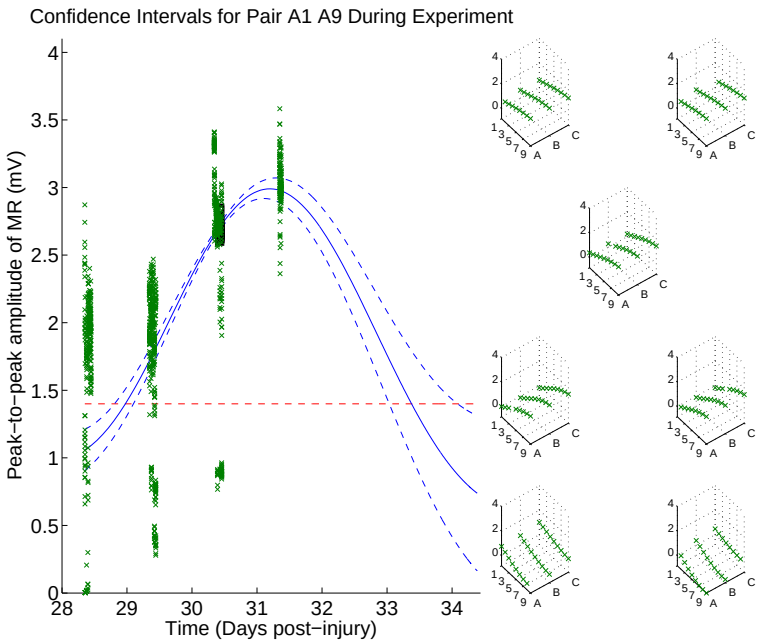
(a) P22

(b) Mean of Each Day's Normalized Mean

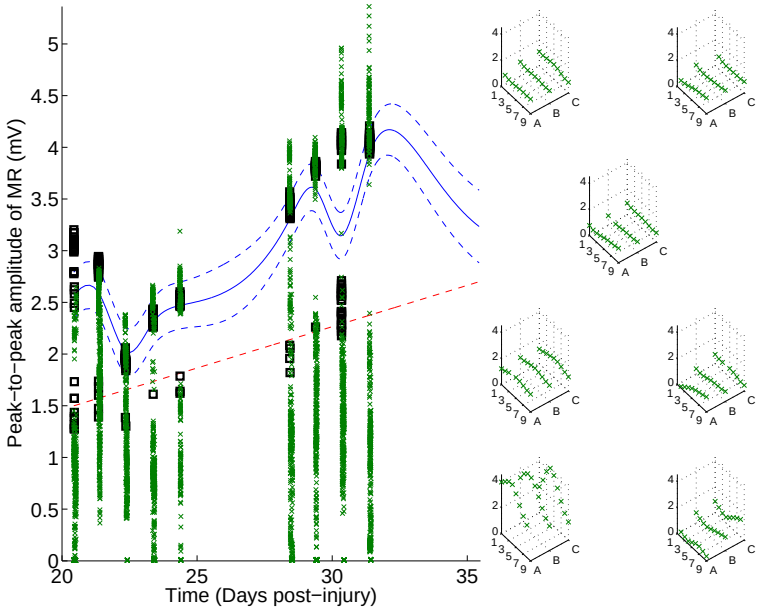
(c) Animal 2

(d) Animal 5

Figure 4.15: Comparison of stimuli and responses observed in the two wired array animals, combining human- and algorithm-commanded data for each animal. (a): On many days, response amplitudes showed similar patterns across animals 2 and 5. The absolute amplitudes of responses in animal 2 were somewhat smaller than those in animal 5 (see Figure 4.4). For most stimuli, there was qualitative agreement in response strength, excepting those for which animal 2 had much stronger responses than in animal 5. (b): For any day on which at least 15 distinct stimuli were present from each of animal 2 and animal 5, normalized mean responses were computed for every pair of electrodes tested that day. For each configuration which was tested on any such day, the mean across such days of the normalized mean response was computed; all bipolar stimuli fell within this set. The mean normalized mean responses are shown. The correlation coefficient for these distributions is $r = 0.5151$; with the removal of the cluster of outliers in the upper right (those with A9 as the anode), this decreases to $r = 0.3510$. (c) & (d): Mean normalized mean response amplitudes for each configuration, from both animal 2 (c) and from animal 5 (d), shown with respect to location on the array; color corresponds to mean normalized mean response strength, where red is close to 1 and blue is close to 0, major box represents anode location, and minor box represents cathode location. Configurations present on any eligible day are shown, regardless of whether or not they were tested on that day in both animals. The subset of highly excitable configurations for animal 2 is strikingly broader than that for animal 5. Note that configurations including the ground wire (i.e., monopolar stimuli) were tested by the human experimenter in animal 2, and are shown using the bottom-most box in each representation of the array (large representation for anode or small representation for cathode).



(a) Animal 2, run 3



(b) Animal 5, run 2

Figure 4.16: The consequences of undesirable kernel characteristics; posterior predictions. The left panel for each subfigure focuses on A1_A9. See Figure 4.13 for more details. (a): The posterior predictions over the stimulus space as of the beginning of day P34 in animal 2, run 3 (based on data acquired on P31 and earlier) demonstrate a severe undershoot in the posterior predictions relative to actual performance. This is a consequence of the smoothness characteristics of the squared-exponential kernel, the time-lengthscale used, and the low noise assumed. The poor spatial predictions (particularly evident in the lower left subplot of the right panel, representing the posterior mean over configurations using A9 as an anode) caused erratic sampling, visible as broadly distributed rewards at the right of Figure 4.4(a). (b): Using the hybrid kernel described in Section 4.5, the posterior predictions at the start of day P35 in animal 5, run 2 (based on data from P31 and earlier) do not display this same pathology, and thus strong queueing behavior was present, as shown in Figure 4.13.