SENSORY INPUT ENCODING AND READOUT METHODS FOR IN VITRO LIVING NEURONAL NETWORKS

A Thesis
Presented to
The Academic Faculty

By

Robert L. Ortman

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Electrical and Computer Engineering

Georgia Institute of Technology

August 2012

SENSORY INPUT ENCODING AND READOUT METHODS FOR IN VITRO LIVING NEURONAL NETWORKS

Approved by:

Professor Christopher J. Rozell, Advisor
School or Electrical and Computer Engineering
Georgia Institute of Technology

Professor Steven M. Potter, Advisor
Wallace H. Coulter Department of Biomedical Engineering
Georgia Institute of Technology and Emory University

Professor Robert J. Butera
School or Electrical and Computer Engineering
Georgia Institute of Technology

Professor Ganesh Kumar Venayagamoorthy
Holcombe Department of Electrical and Computer Engineering
Clemson University

Date Approved: June 27, 2012

ACKNOWLEDGMENTS

This thesis would not have been possible without the encouragement, guidance, and support of my advisors Drs. Christopher Rozell and Steve Potter. I am grateful for Drs. Robert Butera and Kumar Venayagamoorthy for serving on my thesis committee and providing valuable advice. I further wish to thank my colleagues Riley Zeller-Townson and Jon Newman for their key contributions to the NeuroRighter system and other support. Additionally, I want to acknowledge my gratitude to the National Science Foundation for their support under the grant, "Neuroscience and Neural Networks for Engineering the Future Intelligent Electric Power Grid," as part of the EFRI program for COPN projects (NSF EFRI-COPN Project #0836017) and for their additional support via the Integrative Graduate Education and Research Traineeship (IGERT) "Hybrid Neural Microsystems" fellowship. Finally, I want to express my great appreciation to all my family and friends who provided support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BI	Burstiness Index
BIANN	Biologically Inspired Artificial Neural Network
CAT	Center of Activity Trajectory
COPN	Cognitive Optimization and Prediction
C-SVC	C (cost parameter)-Support Vector Classification
dAP	Directly evoked Action Potential
EFRI	Emerging Frontiers in Research and Innovation
I/O	Input / Output
IACUC	Institutional Animal Care and Use Committee
LNN	Living Neuronal Network
LS	Liquid State
LSM	Liquid State Machine
MEA	Microelectrode Array
MLP	Multilayer Perception
NI	National Instruments
NR	NeuroRighter
PCIe	Peripheral Component Interconnect Express
RBF	Radial Basis Functions
SALPA	Subtraction of Artifacts by Local Polynomial Approximation Algorithm
SNR	Signal to Noise Ratio
SVM	Support Vector Machine

SUMMARY

Establishing and maintaining successful communication stands as a critical prerequisite for achieving the goals of inducing and studying advanced computation in small-scale living neuronal networks. The following work establishes a novel and effective method for communicating arbitrary "sensory" input information to cultures of living neurons, living neuronal networks (LNNs), consisting of approximately 20 000 rat cortical neurons plated on microelectrode arrays (MEAs) containing 60 electrodes. The sensory coding algorithm determines a set of effective codes (symbols), comprised of different spatio-temporal patterns of electrical stimulation, to which the LNN consistently produces unique responses to each individual symbol. The algorithm evaluates random sequences of candidate electrical stimulation patterns for evoked-response separability and reliability via a support vector machine (SVM)-based method, and employing the separability results as a fitness metric, a genetic algorithm subsequently constructs subsets of highly separable symbols (input patterns). Sustainable input/output (I/O) bit rates of 16–20 bits per second with a 10% symbol error rate resulted for time periods of approximately ten minutes to over ten hours. To further evaluate the resulting code sets' performance, I used the system to encode approximately ten hours of sinusoidal input into stimulation patterns that the algorithm selected and was able to recover the original signal with a normalized root-mean-square error of 20–30% using only the recorded LNN responses and trained SVM classifiers. Response variations over the course of several hours observed in the results of the sine wave I/O experiment suggest that the LNNs may retain some short-term memory of the previous input sample and undergo neuroplastic

changes in the context of repeated stimulation with sensory coding patterns identified by the algorithm.

CHAPTER I

INTRODUCTION

1.1 Background

Biological neuronal systems possess vast computational power still unparalleled in state-of-the-art artificial neural networks (ANNs). They demonstrate an unmatched ability to solve pattern recognition and non-linear control problems. Elucidating the mechanisms underlying such abilities promises to not only answer key questions of neuroscience and computational intelligence but also to promote the development of new ANNs with vastly superior computational abilities.

Although *in vivo* neuronal systems undeniably learn to perform exceedingly complex computations, it is extremely difficult to achieve the level of fine control over development, learning, and monitoring possible with small-scale networks growing on microelectrode arrays (MEAs). MEA technology permits researchers to potentially employ neuronal cultures to perform arbitrary computations, limited only by the interface properties, hardware, and software. By plating living neuronal networks (LNNs) on MEAs, researchers can form a bidirectional interface between living neurons and computer systems, permitting extensive study of neuronal systems at the small network level (Taketani & Baudry, 2010). Via electrodes embedded in the MEA substrate, we can monitor and stimulate cultures for extended periods. MEA technology has already facilitated the study of basic learning and computation mechanisms in a closed-loop environment (Bakkum, Chao, & Potter, 2008) (Demarse & Dockendorf, 2005) (Demarse, Wagenaar, Blau, & Potter, 2001).

Before one can begin to fully examine many intriguing aspects of neuronal computation in these systems, however, it is essential to develop an effective method for communicating "sensory" inputs to the LNN and extracting state outputs from the LNN. Previous works examining computation in MEAs have presented various input/output (I/O) schemes, but they fall short of demonstrating a technique for communicating complex, high-bandwidth information to and from LNNs for extended periods (Bakkum et al., 2008) (Hafizovic et al., 2007) (Dockendorf, Park, He, Príncipe, & DeMarse, 2009) (Ruaro, Bonifazi, & Torre, 2005). The static goals of prior closed-loop studies, such as controlling a robot's movements among a few degrees of freedom, only require a very low I/O data rate (less than one bit per second) for communication with the LNN (Bakkum et al., 2008). However, highly desired computational goals such as time-series prediction and control of non-linear, non-stationary dynamical systems demand the development of new communication schemes capable of sustaining significantly greater data rates. In developing effective input coding algorithms, one must address not only the issue of finding stimuli capable of communicating effectively with LNNs but must also overcome the limitations inherent in using an I/O interface (the MEA) that significantly subsamples the spatial resolution of the LNN.

Previous researchers have applied liquid state machine (LSM) theory in order to better understand the necessary conditions for communicating and computing with LNNs (Hafizovic et al., 2007) (Dockendorf et al., 2009). Furthermore, LSMs have been effectively applied to solve a wide range of demanding prediction problems, including non-linear power system state control (Venayagamoorthy, 2007). I conducted this research as part of a collaboration aimed at improving biologically inspired artificial neural networks (BIANNs) being explored for power systems control (NSF EFRI-COPN Project #0836017). In the LSM context, the LNN serves as the "reservoir" (see Chapter 3.2 for explanation). However, unlike typical LSMs in which ANNs commonly function as the reservoir and one has the ability to transfer input data to each "neuron" precisely and independently, LNNs on MEAs may only receive inputs represented by sequences of electrode-specific stimulation. Each electrode transfers its corresponding input pulse through an electrolyte solution (the neurons' growth and support media) to tens to hundreds of neurons with inherently different weights for each receiving neuron. The value of the weights is unknown and beyond the user's control. The electrode interface also limits output, with each electrode receiving a weighted sum of the membrane voltages of nearby neurons (typically one to five cells) (Chao, Bakkum, & Potter, 2007). Furthermore, the neurons influenced by stimulation of a particular electrode do not necessarily match the neurons from which the electrode is receiving signals.

A useful sensory coding scheme must employ an algorithm capable of adapting to each specific culture and changing over time since every culture has a unique connectivity network, relationship to the MEA electrodes, and dynamics. As a consequence of neuroplasticity and other biological factors influencing LNN connectivity, a network's responses to a particular input vary over time. Consequently, sensory codes that are excellent for information transfer at one moment may be less effective later. When finding sets of effective sensory codes, the algorithm must therefore assess not only response separability but also reliability over time. For some types of experiments, if the LNN ceases to respond effectively to certain input patterns (begins producing irregular responses to a pattern or starts generating indistinguishable responses for certain patterns), a method could replace them with new, more effective patterns. However, it is important to establish if, although the LNN responses to a particular pattern set drift over time, the capacity of the LNN to produce a diverse repertoire of responses to the candidate patterns persists. If it does not, such a pattern replacement technique would not be as useful. Moving beyond the analysis of short-term reliability involved in initially forming effective sensory input pattern sets, I also present results characterizing pattern set reliability over time lengths much longer than the initial training period — up to ten hours.

Finally, in order to truly harness the computational potential of LNNs, future applications require advancing beyond only using the neurons as a reservoir and actually controlling the LNN's plasticity and memory to perform computations. As a result, ideal input patterns should evoke neither highly inconsistent, random responses nor completely repeatable responses: If the LNN state response is always identical to a given input pattern regardless of what preceded it, there would be no readable memory in the network. Although the induction of short-term memory is not a specific goal of this project, the data analysis of the sine wave I/O experiment assesses the influence of the prior input on the LNN's response to the present input.

1.2 Primary Research Aims and Accomplishments

The following work establishes a novel and effective method for communicating arbitrary "sensory" input information to LNNs consisting of approximately 20 000 rat cortical neurons plated on MEAs containing 60 electrodes. The sensory coding algorithm determines a set of effective codes (symbols), comprised of different spatio-temporal patterns of electrical stimulation, to which the LNN consistently produces unique responses to each individual symbol. The algorithm evaluates random sequences of candidate electrical stimulation patterns for evoked-response separability and reliability via a support vector machine (SVM)-based method, and employing the separability results as a fitness metric, a genetic algorithm subsequently constructs subsets of highly separable symbols (input patterns). Sustainable input/output (I/O) bit rates of 16–20 bits per second with a 10% symbol error rate resulted for time periods of approximately ten minutes to over ten hours. To further evaluate the resulting code sets' performance, I used the system to encode approximately ten hours of sinusoidal input into stimulation patterns the algorithm selected and was able to recover the original signal with a normalized root-mean-square error of 20–30% using only the recorded LNN responses and trained SVM classifiers. Response variations over the course of several hours observed in the results of the sine wave I/O experiment suggest that the LNNs may retain some short-term memory of the previous input sample and undergo neuroplastic changes in the context of repeated stimulation with sensory coding patterns identified by the algorithm.

CHAPTER II

LABORATORY METHODS

2.1 Neuronal Cell Cultures

I enzymatically and mechanically dissociated cells from E18 (embryotic day 18) rat cortices to obtain a target density of approximately 2 500 cell/μL of medium (approximately 500 cells/mm^2) and then layered the neurons onto laminin-coated 60-electrode (59 recording/stimulation electrodes plus one ground) Multichannel Systems MEAs (30 μm diameter titanium nitride electrodes in a square grid with 200 μm spacing) (Bakkum et al., 2008) (Hales, Rolston, & Potter, 2010) (Potter & Demarse, 2001). I plated and grew cells in Jimbo's medium (containing 10% equine serum (Brewer, Torricelli, Evege, & Price, 1993), sodium pyruvate, insulin, and GlutaMAX™) (Jimbo & Kawana, 1992) (Potter, Wagenaar, & Demarse, 2005) (Wagenaar, Nadasdy, & Potter, 2006). When not in use, I stored the LNNs in an incubator at 35°C with 5% CO_2, 9% O_2, and 65% relative humidity in Teflon®-membrane sealed MEAs (Potter & Demarse, 2001). I performed experiments during three to six weeks *in vitro* on cultures of approximately 20 000 living neurons. All animals were euthanized in accordance with Georgia Institute of Technology's IACUC protocol.

2.2 Data Acquisition

The customized electrophysiology system, NeuroRighter (Rolston, n.d.), (Rolston, Gross, & Potter, 2009a) allows for versatile low-latency closed-loop experiments (Rolston, Gross, & Potter, 2009b). The hardware for stimulation and recording includes a Multichannel Systems MEA60 preamp to which the MEA is directly connected. The amplified MEA output, containing neural signals, passes through custom signal conditioning interface boards before terminating onto two National Instruments™ (NI) PCIe-6259 data acquisition cards (32 analog input channels each) installed in a PC. The stimulation output originates from the computer from a PCIe-6259 card via its four analog outputs and then passes through custom interface boards, multiplexer headstages, and into the MEA. The PCIe-6259 cards' digital outputs control the multiplexers. Independent recording and stimulation is possible from all 59 electrodes but cannot occur simultaneously.

2.3 Stimulus Artifact Suppression Techniques

Stimulus artifacts in the LNN recording system present a formidable obstacle to reliable data collection and analysis for these types of experiments. The system includes several techniques to ensure the recorded results accurately reflect neuronal activity. NeuroRighter incorporates band pass filtering and thresholding to detect spikes. In addition, NeuroRighter includes post-processing using the SALPA (subtraction of artifacts by local polynomial approximation) algorithm (Rolston, Gross, & Potter, 2009a), a variable time-constant polynomial curve fit used to subtract large voltage changes due to stimulation (Wagenaar & Potter, 2002). The real-time SALPA algorithm in NeuroRighter effectively suppresses a large amount of stimulation artifacts (Rolston, Gross, & Potter, 2009a) (Wagenaar & Potter, 2002), and further processing removes spikes with greater than 300 μV peak-peak amplitudes.

CHAPTER III

SENSORY INPUT ENCODING AND OPTIMIZATION METHODS

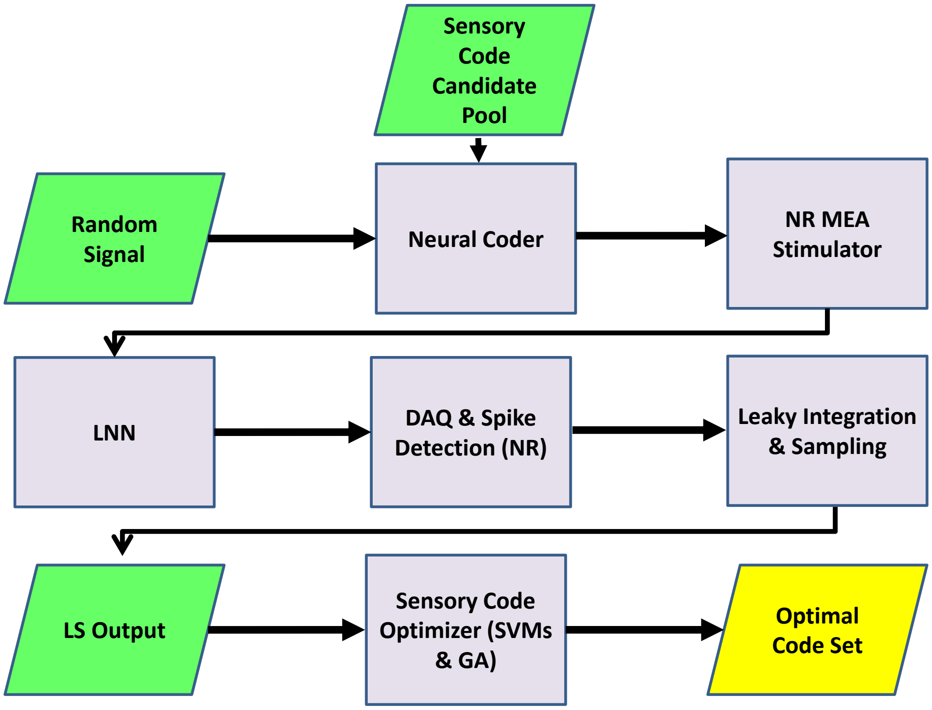
Sensory Code Candidate Pool
Random Signal
Neural Coder
NR MEA Stimulator
LNN
DAQ & Spike Detection (NR)
Leaky Integration & Sampling
LS Output
Sensory Code Optimizer (SVMs & GA)
Optimal Code Set

Figure 1: Sensory code optimization overview

3.1 Overview

Figure 1 presents a flowchart of the overall sensory code optimization system. The *Neural Coder* block takes as input both a digital signal, represented by an array of integer sample values, and a set of stimulation pattern definitions used to map each of the signal's sample values to specific LNN stimulation patterns consisting of a unique electrode sequence and frequency. For the pattern optimization stage, the mappings correspond to a pool of *n* candidate patterns chosen randomly from a range of biologically reasonable stimulation frequencies and electrode sequences of a preset length (typically four). The input signal is a random integer sequence drawn from a uniform distribution on the interval [0, n – 1]; each integer represents a different pattern in the candidate pool of n patterns.

In the subsequent stages, the NeuroRighter system stimulates the MEA on which the LNN is living based on the *Neural Coder* output; each individual electrode stimulus consists of a biphasic square pulse. NeuroRighter performs real-time spike detection in software. Offline analysis programs I implemented in MATLAB first perform leaky integration on the detected spikes and sample the results to produce data feature vectors, which attempt to capture the meaningful spatio-temporal information contained in the spike responses to each input pattern ("Leaky Integration & Sampling" box in Figure 1). A multi-class support vector machine (SVM) then builds a model based on the responses to a portion of the response data. Using the model, the SVM classifier attempts to classify the remaining data, and the mean classification accuracy serves as the metric for assessing response separability and reliability of candidate pattern sets (bottom row of Figure 1).

An SVM training algorithm builds a model used to separate the feature vectors (based on the LNN spike responses) into categories associated with the different input patterns. A hyperplane separates the data, represented by points in a high-dimensional space transformed by a kernel function, into two distinct regions. In order to achieve classification into more than two categories, a common approach (and used in this research as implemented by LIBSVM) is to train a set of binary classifiers (one for each category) such that each attempts to maximally separate the feature vectors belonging to its particular class from those belonging to all other classes. When applying the multi-class model after its construction, each binary classifier attempts to identify a given unknown feature vector, and the output is defined as the binary classifier with the largest graded response (winner-takes-all approach) (Chang & Lin, 2001). I chose to use SVM classifiers based on their flexibility, ability to perform non-linear classification, well-documented success in computational biology (Schölkopf, Tsuda, & Vert, 2004), and use by previous researchers studying input response separability in LNNs on MEAs (Hafizovic et al., 2007). Despite their superiority in many respects, a substantial drawback is the computational complexity, especially when applying them to multi-class problems and using non-linear kernels. However, I was able to mitigate the impact of computational complexity by using a CPU cluster (see Chapters 3.4–3.6 for more detail).

3.2 Liquid State Machine Background

The computational power of an LSM is derived from and dependent on the presence of a *reservoir* capable of computing a very large number of non-linear functions on the input signal. Given that input information is properly encoded for a particular reservoir, and the reservoir possesses sufficiently rich non-linear dynamics, complex non-linear systems can be modeled and predictions obtained using only linear combinations of the LSM state (Maass, Natschläger, & Markram, 2002). In order to effectively encode an input signal for the LSM, a representation must be determined such that input patterns representing different states in the system being modeled consistently evoke separable (distinguishable) responses in the reservoir (the *separation property* of LSMs) (Maass et al., 2002). I designed the algorithm discussed in the following sections to find highly separable subsets of candidate input patterns tested in random order.

3.3 Experimental Protocol for Testing Candidate Input Patterns and Reading the Liquid State

The experimental protocol for testing and evaluating the efficacy of input coding patterns consisted of initially stimulating the reservoir with random trains of symbols comprised of different spatio-temporal patterns. Each input pattern (symbol) is composed of a unique sequence of (typically four) electrodes stimulated at a specific frequency between 15 and 55 Hz. I structured the stimulation as interleaved symbols and spaces with symbol length varying from 55–200 ms and an inter-symbol delay of 100 ms, unless noted otherwise. For the results presented, the corresponding mean stimulation frequency across the MEA is approximately 13–26 Hz, which is fast enough to substantially reduce spontaneous bursting activity (widespread, synchronized neuronal firing) that could disrupt meaningful information transfer (Wagenaar, Madhavan, Pine, & Potter, 2005) (Madhavan, Chao, Wagenaar, Bakkum, & Potter, 2006). The stimulation waveform consists of 400 μs voltage-controlled biphasic square pulses with a peak-to-peak amplitude of 0.7 V (Wagenaar, Pine, & Potter, 2004).

The experimental input pattern training phase consists of thousands of stimulations with candidate input patterns (30–100 trials per unique symbol) chosen at random on a uniform distribution from a total set of 100–400 patterns (100 unless otherwise noted). Detected spikes from the response period following stimulation with each input symbol pass through a leaky integrator function, Equation 1, whose output is sampled at 16 evenly spaced time intervals five milliseconds apart. Such an approach is consistent with commonly used techniques for extracting responses from LSMs and LNNs (Hafizovic et al., 2007) (Dockendorf et al., 2009) (Maass et al., 2002) (Jaeger, Lukosevicius, Popovici, & Siewert, 2007).

For liquid state (LS) readout, I implemented leaky integration using the equation,

$$x_i(t) = \sum_{s_j} e^{-(t-s_j)/\tau}, \tag{1}$$

in which $x_i(t)$ are the values of the LS readout function over time for each electrode i. The s_j values in the summation are the spike occurrence times relative to the start of the response region for each spike, j, detected in the particular response period. The time constant, τ = 60 ms, limits the memory of the output to a physiologically relevant range (Hafizovic et al., 2007). Each stimulation trial corresponds to 59 LS outputs, one per electrode. Since 16 samples comprise each electrode's integrator output and there are 59 readout electrodes, every response produces a 944-dimensional vector.

3.4 Liquid State Readout and SVM Parameter Optimization

I determined leaky integration and SVM parameters based on the results of varying them over reasonable ranges and comparing separability performance results. I tested the leaky integration time constant, τ, over a range of 5–100 ms, and found 60 ms to be generally optimal. In addition, I varied the number of samples taken from the LS response period from one to 32, inclusive, in powers of two. Results improved substantially up to 16 samples, and no significant classification accuracy improvement occurred beyond 16 samples. Utilizing more samples (and therefore more SVM input features) significantly increases computation time and memory requirements; hence, I chose 16 samples. Figure 2 shows the relative performance for different combinations of the leaky integration time constant and the portion of the 100 ms response period used for LS output (starting at the beginning of the window). Pixel colors indicate performance, with the red end of the spectrum representing better results. More specifically, the colors correspond to the percentage of candidate pattern pairs with less than 10% decoding error out all combinations of the 100 candidate patterns (using the trained SVM classifier). The top portion of Figure 2 shows the mean results over approximately three hours of data collected in Experiments 1 and 2 using Culture A (see Table 1 for more information).

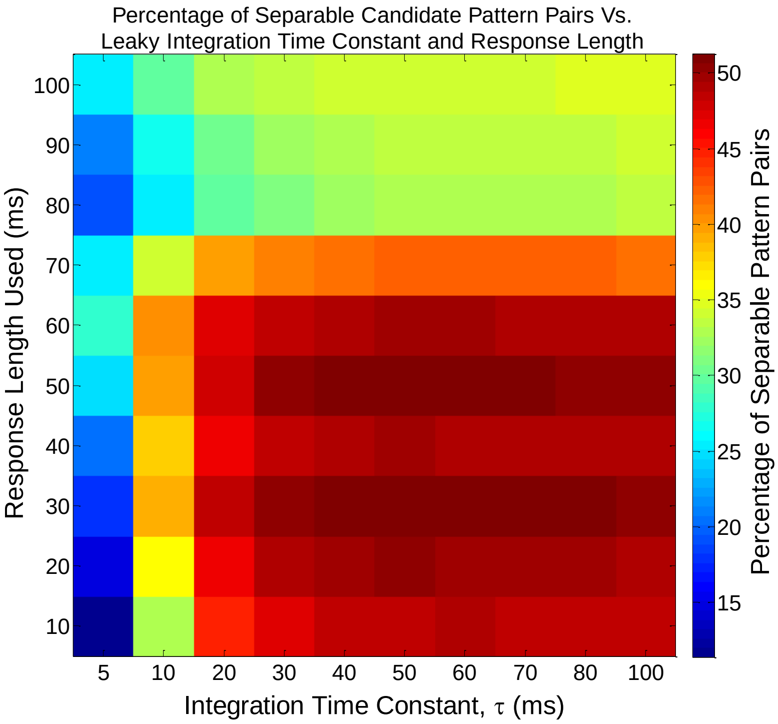

Figure 2: Leaky integration parameter optimization

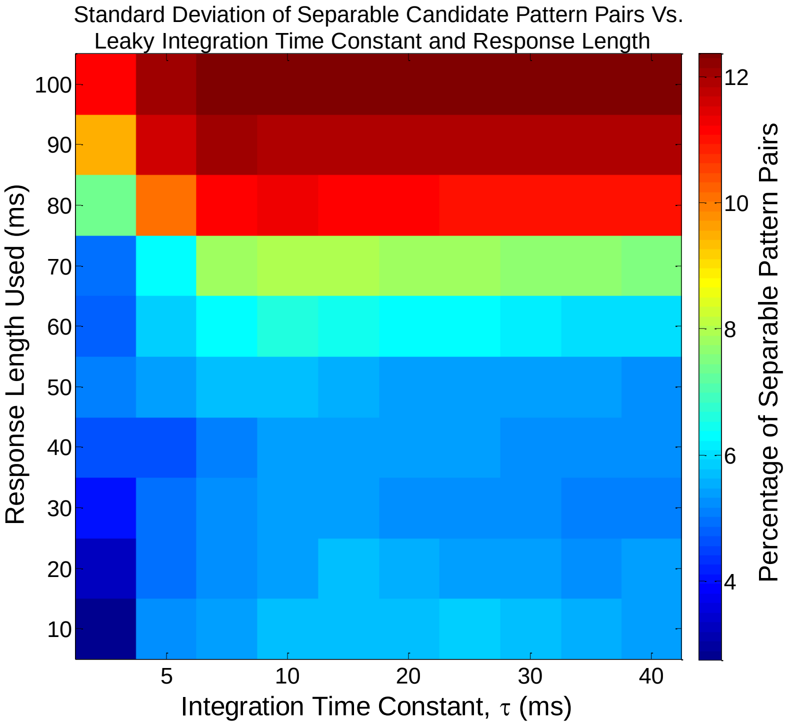

Figure 2: Leaky integration parameter optimization

I also selected the SVM kernel and corresponding parameters to achieve the most effective separation of evoked responses to different inputs. I tested the following kernels: linear, quadratic, third-order polynomial, forth-order polynomial, radial basis functions (RBF), and sigmoid (equivalent to a two-layer multilayer perceptron (MLP)). I chose to use the sigmoid kernel, Equation 2, since it produced the best results (highest classification accuracies). The RBF kernel also produced reasonably close results, but the others were largely ineffective at separating the responses. The sigmoid kernel is the hyperbolic tangent function,

$$\tanh(\gamma u' v + k_0), \tag{2}$$

in which the data vectors (LS responses) comprise u and v, and γ and k_0 represent the kernel parameters. Via a grid search approach, I found the optimal values to be as follows: C = 16 (cost parameter of C-SVC SVM), γ = (1 / number of features) =

$1/944 \approx 1.06 \cdot 10^{-3}$, and $k_0 = 0$. Although these parameters are not the best for every experiment, I fixed their values across all results in order to maintain consistency while sacrificing little performance.

3.5 Evaluation of Separability via an SVM Classifier

After conducting the experiments described in Chapter 4.1, I used offline analysis software to assess the separability and reliability of candidate input encoding patterns. I used the open source SVM package, LIBSVM 3.1, to solve the multiclass SVM training and classification problem employed to evaluate separability as follows (Chang & Lin, 2001). My analysis software randomly selects one-third of the response data to use for training and subsequently attempts classification on the remaining two-thirds. Repeated random sub-sampling cross-validation is used in order to eliminate the bias that might occur from only choosing one random training and classification set (Chang & Lin, 2001) (Geisser, 1993). Cross-validation reduces variance and protects against Type III statistical errors (Mosteller, 1948). My algorithm randomly reselects the training and testing groups 30 times (chosen based on the minimum number required to produce negligible variance) and then calculates the overall mean performance. I evaluated separability performance by calculating the mean classification accuracy for each particular set of patterns evaluated. Classification accuracy $(1 - \text{probability of symbol error})$ is defined as the ratio of the number of symbols correctly identified to the total number tested.

3.6 Pattern Set Optimization via a Genetic Algorithm

From each initial candidate set of patterns, I applied a genetic algorithm (Fraser, 1957) to find subsets of patterns which reliably evoke separable responses. The genetic algorithm determines size *n* subsets of the most separable patterns from the remaining patterns in the following manner: First, it assesses separability between all pairwise combinations ($n = 2$) of the candidate patterns. It subsequently selects the most separable subsets to serve as the parent sets for forming the next generation, pattern sets of size ($n + 1$). Mean classification accuracy serves as the fitness function. Next generation candidate sets of ($n + 1$) patterns are bred from the most separable of the size *n* subsets by appending a single additional pattern chosen from the set of candidate patterns. My software evaluates all combinations of new pattern sets subject to these fitness constraints (the next generation) and repeats the process to produce subsequent generations until the desired set size is attained, the separability performance drops below a certain threshold, or the computation time has been exhausted (depending on the application).

3.7 Parallel Computing

Due to the immense computational requirements of executing the sensory code optimization algorithm, I parallelized and executed the software via MATLAB Distributed Computing Server™ on a 64-CPU-core cluster at the Georgia Institute of Technology Laboratory for Neuroengineering. The large number of test pattern sets for each particular set size (generation) can be independently evaluated for separability; parallelizing this aspect of the algorithm significantly improves execution speed. Computational speed increases almost linearly with the number of CPU cores.

CHAPTER IV

SENSORY CODE OPTIMIZATION EXPERIMENTS AND RESULTS

4.1 Sensory Code Optimization Experiments

The following figures present long-term pattern separability data collected from four different cultures in seven experiments. For all experiments, I used the same 100-candidate pattern sets with 100 ms inter-pattern delays except for the experiment on Culture D, in which I used 400 candidate patterns with 50 ms delays. Each long-term experiment consists of numerous separate subsections of 10.7 minutes for all experiments except for the protocol employing 10 ms delays, in which subsections last 20.8 minutes, and the 400-candidate-pattern protocol, in which subsections last 50.3 minutes. The candidate pattern pool remained constant through each long-term experiment, but I recalculated the most separable pattern subsets for each experimental subsection, corresponding to each point in the figures of Chapter 4.2. Refer to Table 1 for more details on the parameters and conditions associated with each experiment presented in the following section. The mean spike rate is the culture-wide average detected spike rate during the experiment, and the “burstiness index” (BI) is defined as the following: It is a value normalized between zero and one such that zero indicates no bursts, and one indicates all spikes occurred within bursts. The BI algorithm determines the fraction of total spikes in an experiment that occurred in the 15% most active non-overlapping one-second windows (Wagenaar, Pine, & Potter, 2006).

Table 1: Parameters for Long-Term Sensory Coding Experiments

Expt.#, Culture	Weeks *in vitro*	Candidates Tested	Inter-Pattern Delay (ms)	# of Trials Per Pattern Per Run	Time Per Run (min.)	Total Time (hours)	Mean Spike Rate (Hz); Burstiness Index (BI)	Plot Symbol
1, A	3	100	100	30	10.7	9.0	272; 0.17	●
2, A	3	100	100	30	10.7	9.5	196; 0.23	●
3, C	3	100	100	30	10.7	3.8	23.5; 0.33	+
4, B	4	100	100	30	10.7	9.5	17.6; 0.43	□
5, A	5	100	10	100	20.8	12.1	298; 0.10	○
6, D	5	400	50	100	50.3	12.5	156; 0.08	Δ, X
7, A	6	100	100	30	10.7	17.8	200; 0.25	●

4.2 Sensory Code Optimization Results

Figure 3 presents the sizes of the largest pattern sets whose mean classification accuracy is greater than or equal to 90%.

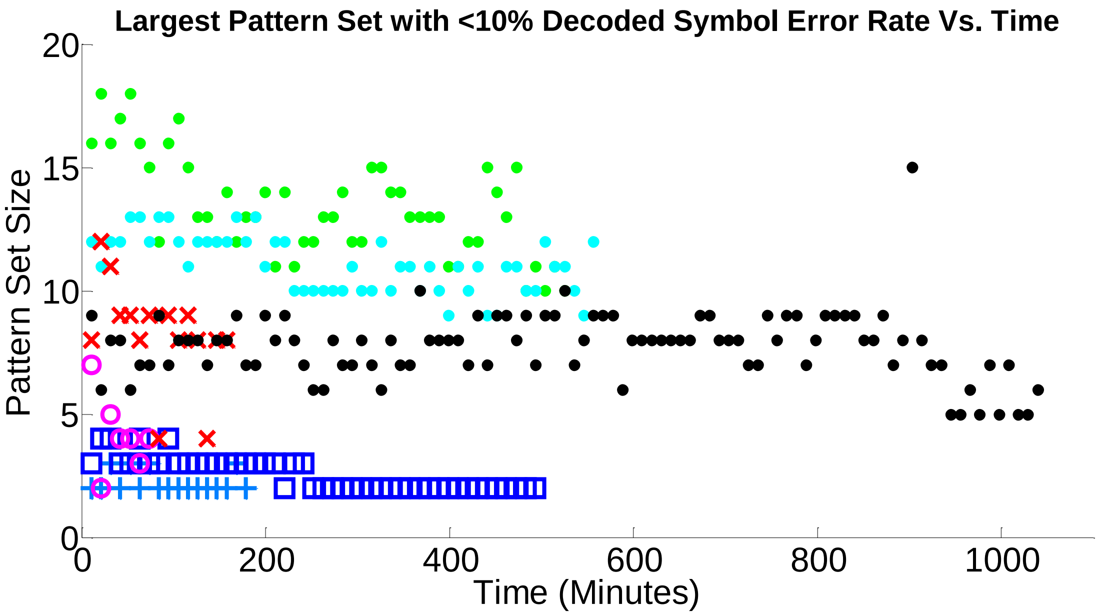

Figure 3: Long-Term input coding analysis with constrained error rate (<10%)

Figure 4 displays the results from the perspective of constraining the target set size instead of separability: The plot shows mean classification accuracies for the best 16-pattern sets found in each subsection (point).

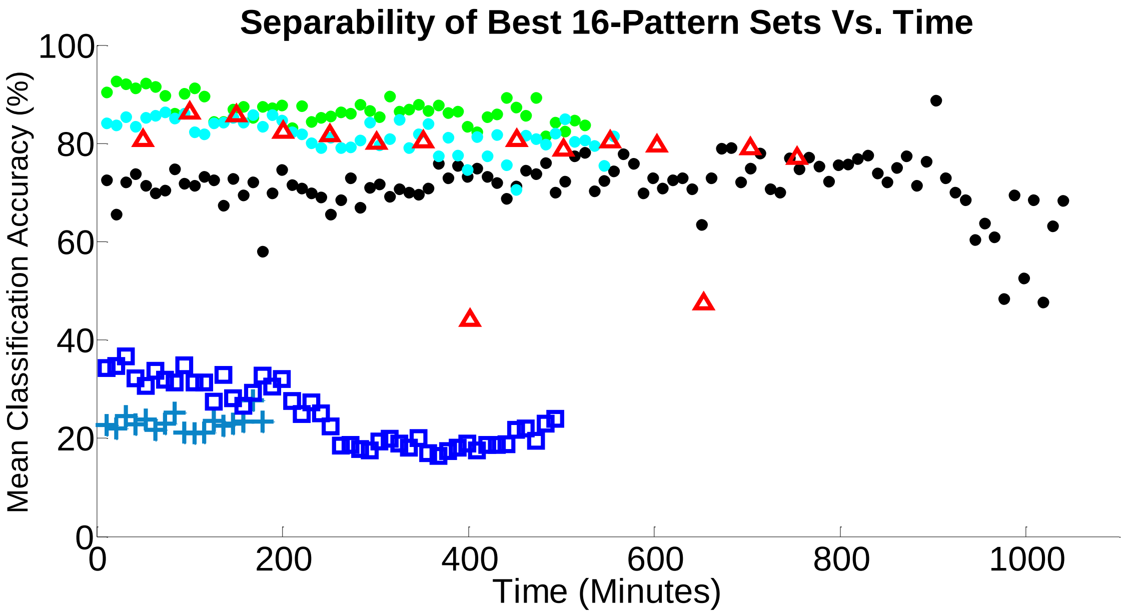

Figure 4: Long-Term input coding analysis with constrained pattern set size (16 patterns)

Figure 5 displays the bit rates corresponding to the input schemes determined by the sensory coding algorithm as it varies over time. I calculated the bit rate using the following equation:

$$bit\ rate = symbol\ rate \cdot \log_2(symbol\ set\ size). \qquad (3)$$

I computed the symbol rate used in Equation 3 by taking the reciprocal of the mean time required for the symbols' stimulation sequences (varies depending on the symbol) and the inter-symbol delay (fixed for a particular experiment).

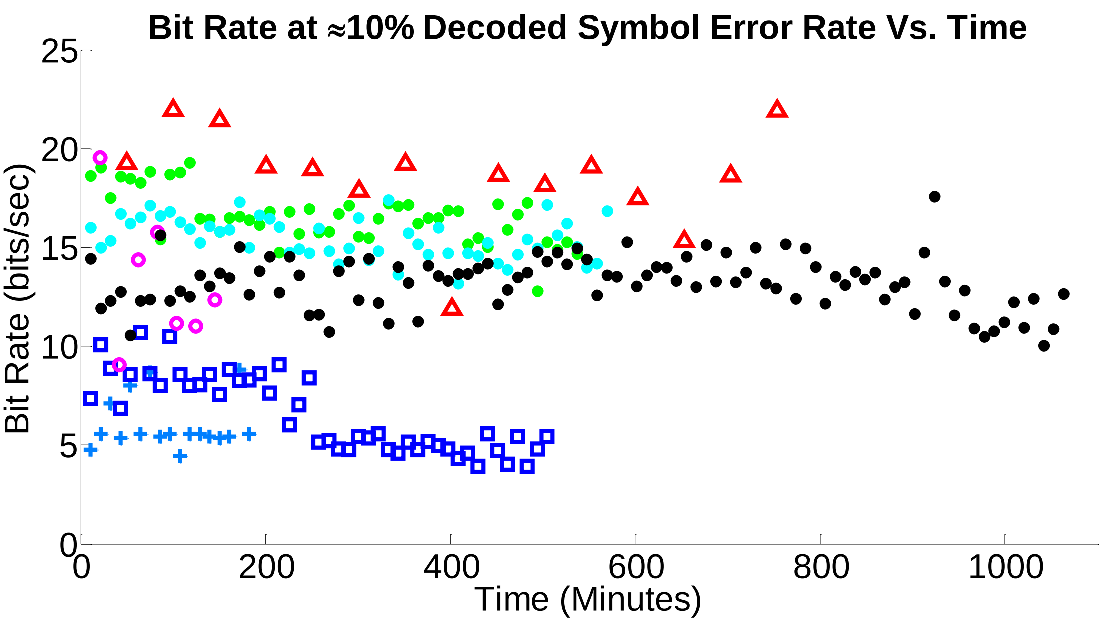

Figure 5: Long-Term input coding bit rate performance with error rate of ≈10%

Culture A consistently produced the best separability results (when correcting for the increased symbol rate used in the experiment on Culture D). The culture was highly active and living on an MEA with excellent quality electrodes (based on impedance measurements and SNR). Shortening the inter-symbol delay from 100 ms to 10 ms significantly reduced performance, especially when viewed from the vantage point of symbol error rate versus set size. However, when considering the overall bit rate, performance did not decline nearly as much due to the higher symbol rate, but the 10 ms delay case clearly resulted in a much smaller set of separable patterns given the same number of initial candidates. The results from Culture D are intriguing since the culture had a spontaneous average firing rate that was several times lower than that of Culture A and had poorer quality MEA electrodes but increased its firing rate and performed reasonably well using the 400 candidate pattern set with 50 ms inter-pattern delays. Figure 6 displays another facet of the seven experimental data sets discussed in this section. For each of the seven experiments shown in Table 1 (see Table 1 for legend), Figure 6 shows the average over the experiment subsections (time points plotted in Figures 3–5) of the mean classification accuracy values for the most separable subset of candidate patterns the genetic algorithm constructed for each symbol set size evaluated. The separability appears to decline more linearly with increasing symbol set size for Culture A (the best-performing, most active culture — represented by solid circle markers in Figure 6), whereas for the least active cultures (Cultures B and C), the decline more closely resembles exponential decay.

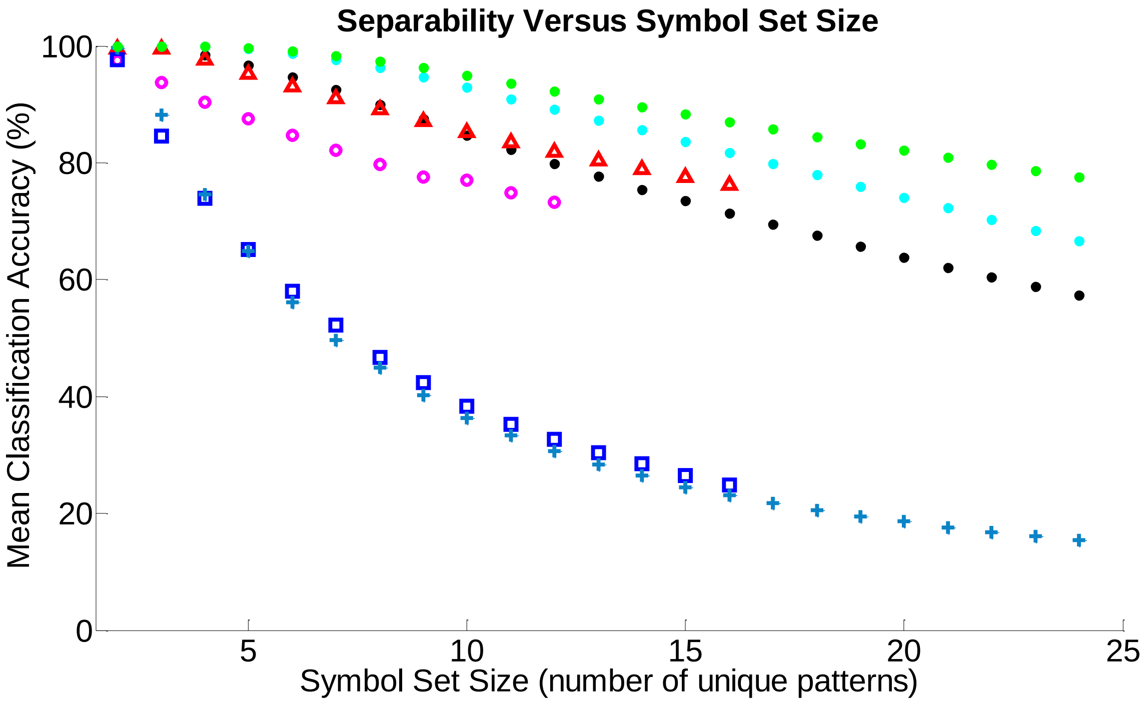

Figure 6: Separability versus symbol set size

In an effort to determine what factors might affect bit rate, I computed the spike rate and "burstiness" index (BI) for each subsection of each experiment. The spike rate is the culture-wide activity in spikes per second, and the BI is described in (Wagenaar, Pine, et al., 2006). I calculated the bit rates shown in Figure 7 in the same manner as those presented in Figure 5 — using Equation 3 and an acceptable symbol error rate of approximately 10%. In general, within a particular experiment and culture, the bit rate did not correlate strongly with spike rate or BI; however, across different cultures, higher bit rates correlated with higher spike rates and lower BI values. Figure 7 shows the lack of correlation between bit rate and spike rate along with BI within each of the seven experiments discussed (refer to Table 1). The bottom right plot of Figure 7A and Figure 7B shows that there is, however, an overall positive correlation between a culture's mean spike rate and its bit rate and a negative correlation between its BI and its bit rate. I computed the values in these plots (square markers) by averaging the bit rate, spike rate, and BI values over the time course of each of the seven experiments: Hence, each square marker represents the mean of the values plotted for each experiment in the other plots.

The results indicate that more active cultures and/or stimulation patterns that evoke more spiking activity are capable of sustaining a higher communication data rate using the methods explored in this research. In addition, the results show that greater bursting activity negatively impacts communication, which is in accordance with previous experiments showing that bursting undermines phenomena related to computation and learning in LNNs (Madhavan et al., 2006) (Wagenaar et al., 2005). By constructing my candidate stimulation patterns and random test signals to maintain a culture-wide stimulation frequency of 13–26 Hz (see Chapter 3.3), I most likely was able to substantially reduce bursting (Wagenaar, Pine, et al., 2006), but using the frequency constraints did not quiet bursting nearly as well in the less healthy LNNs, Cultures B and C.

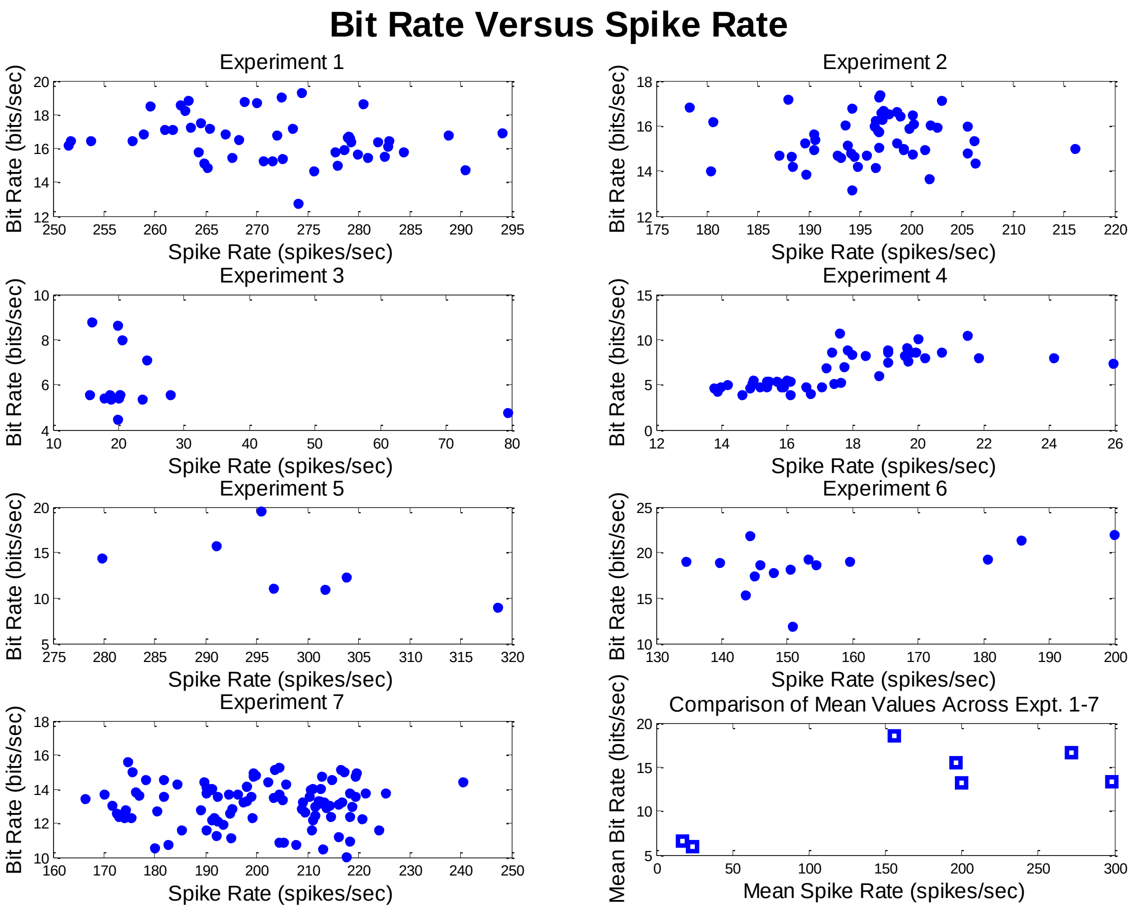

Figure 7A: Bit rate versus activity (spike rate)

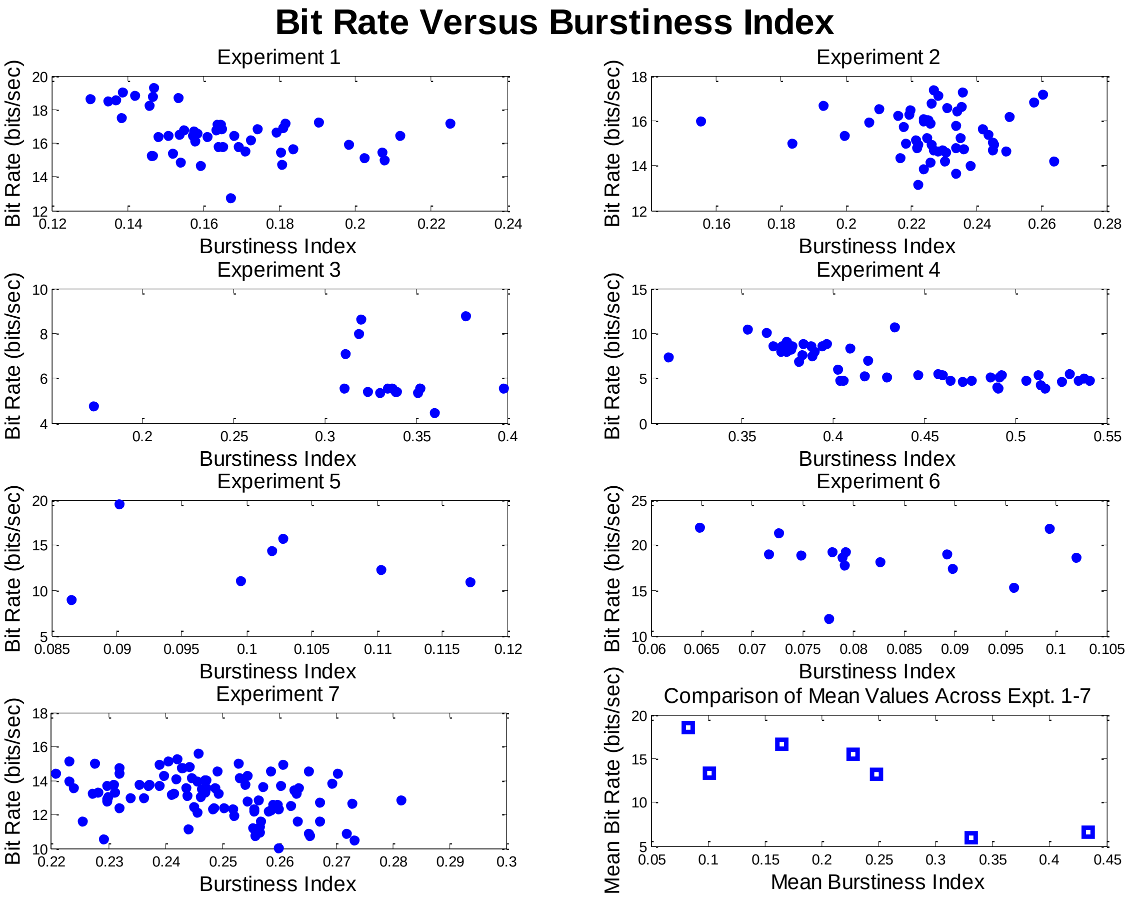

Figure 7B: Bit rate versus “burstiness”

CHAPTER V

LONG-TERM COMMUNICATION WITH LIVING NEURONAL NETWORKS

5.1 Input Encoding and Readout of a Sine Wave via an LNN

After establishing an effective method for determining effective sensory input pattern sets, I performed experiments to evaluate the ability to read a sensory input signal represented using the sensory coding technique described in Chapter 4. During the initial input pattern optimization, I stimulated the MEA for 10.7 minutes with 100 candidates (same method as described in Chapter 4.1). I executed the sensory coding algorithm on the results and found a set of 16 input patterns whose corresponding evoked responses were distinguishable by the trained SVM classifier 91% of the time. I then used these patterns to represent the sample values of a 0.16 Hz sine wave sampled at 4 Hz with 4 bits per sample. I mapped integer amplitude values $\in [0, 15]$ to input patterns such that patterns that evoked larger amounts of spiking activity (on average) represent larger input values.

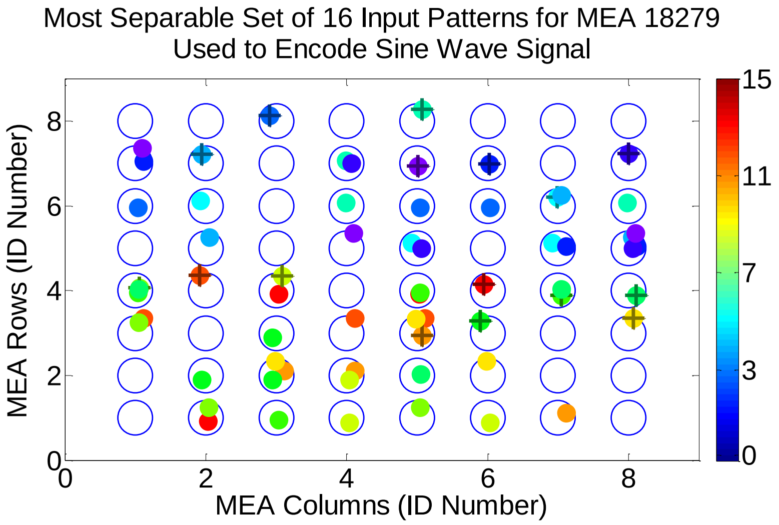

Figure 8: Pattern set used to encode sine wave input

Figure 8 shows the MEA electrode locations of the 16 most separable patterns (91% mean classification accuracy) identified from a test set of 100 candidate patterns based on the results of a 10.7-minute pattern optimization experiment on Culture A. Sequences of four electrodes marked with the same color (Figure 8) comprise each pattern I used to represent a particular input value. The final electrode in each sequence has a "+" marker superimposed to set it apart since it contributes greater influence on response separability because the response window contains directly evoked action potentials (dAPs) elicited by stimulating that electrode. The color bar on the right of the plot indicates the input signal values represented by each particular pattern. Larger valued colors (as specified on the color bar) correspond to patterns that evoked greater amounts of spiking activity in the LNN (during the initial sensory code optimization phase). I then used the resulting sensory-encoded sine wave to stimulate the LNN continuously for 10.3 hours via NeuroRighter.

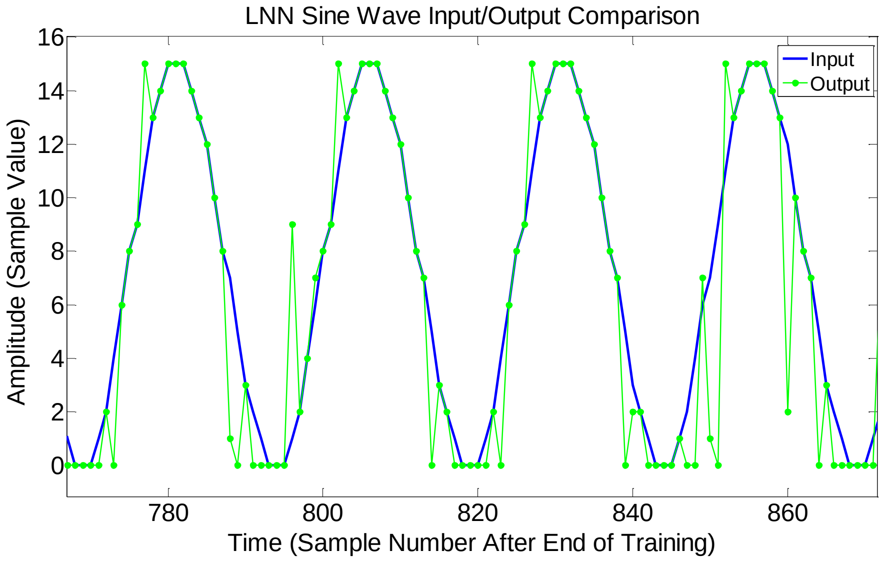

Figure 9: Sine wave I/O in LNN

A decoder must translate neural spiking responses into meaningful output data. Previous closed loop learning experiments in LNNs used the center of activity trajectory (CAT) to represent network output (Chao et al., 2007). The CAT approach translates the LNN electrical activity into a two-dimensional vector reflecting the spatially weighted average position of electrode activity. The CAT approach reduces high-dimensional LNN activity to two dimensional output samples, losing fine spatio-temporal structure. Consequently, I employed a new output decoding method using trained multi-class SVM classifiers.

The following experimental results support the effectiveness of the SVM-based decoding method. I trained the decoder on the first 25 minutes of LS response data and then used it to decode the sine wave input from the LNN responses to the encoded sensory input (sine wave). My trained decoder achieved readout with relatively low error. Figure 9 presents a small segment of the decoded data (green waveform) compared against the original sinus input (blue waveform). Although the example in Figure 9 may appear to indicate that certain decoding errors are more common, such trends typically persist only over a few seconds to a few minutes of data and are therefore not representative of the overall symbol confusion statistics. In Figure 10, the normalized mean error rate (blue), and symbol error rate (red) indicate degradation in decoding accuracy over time as the LNN state drifts. However, retraining the detector periodically (approximately every two hours) maintained much more accurate decoding over the course of approximately ten hours without requiring any changes in stimulation patterns. In fact, the error decreased, indicating that the input patterns representing the sine wave evoked more separable and/or reliable responses over time.

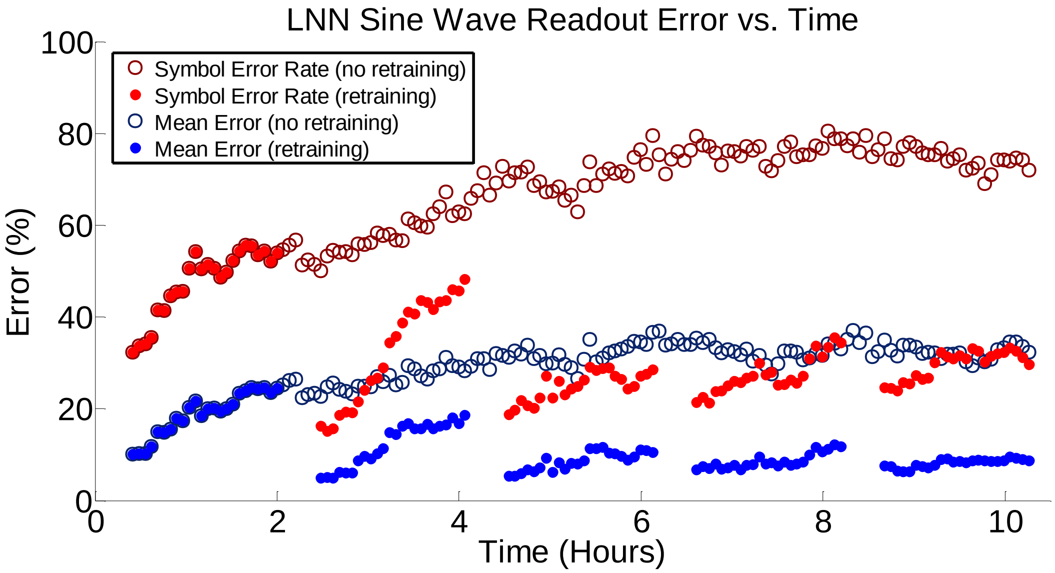

Figure 10: Sine wave readout error over the course of 10.3 hours

The sensory coding algorithm clearly demonstrated its effectiveness for encoding basic sensory inputs such as the sine wave, but as network activity and responses drift over time, one must retrain the detector to maintain accurate decoding. Such LNN state drift would most likely produce changes in the "meaning" to the network of each input pattern over long time periods, disrupting potential computations.

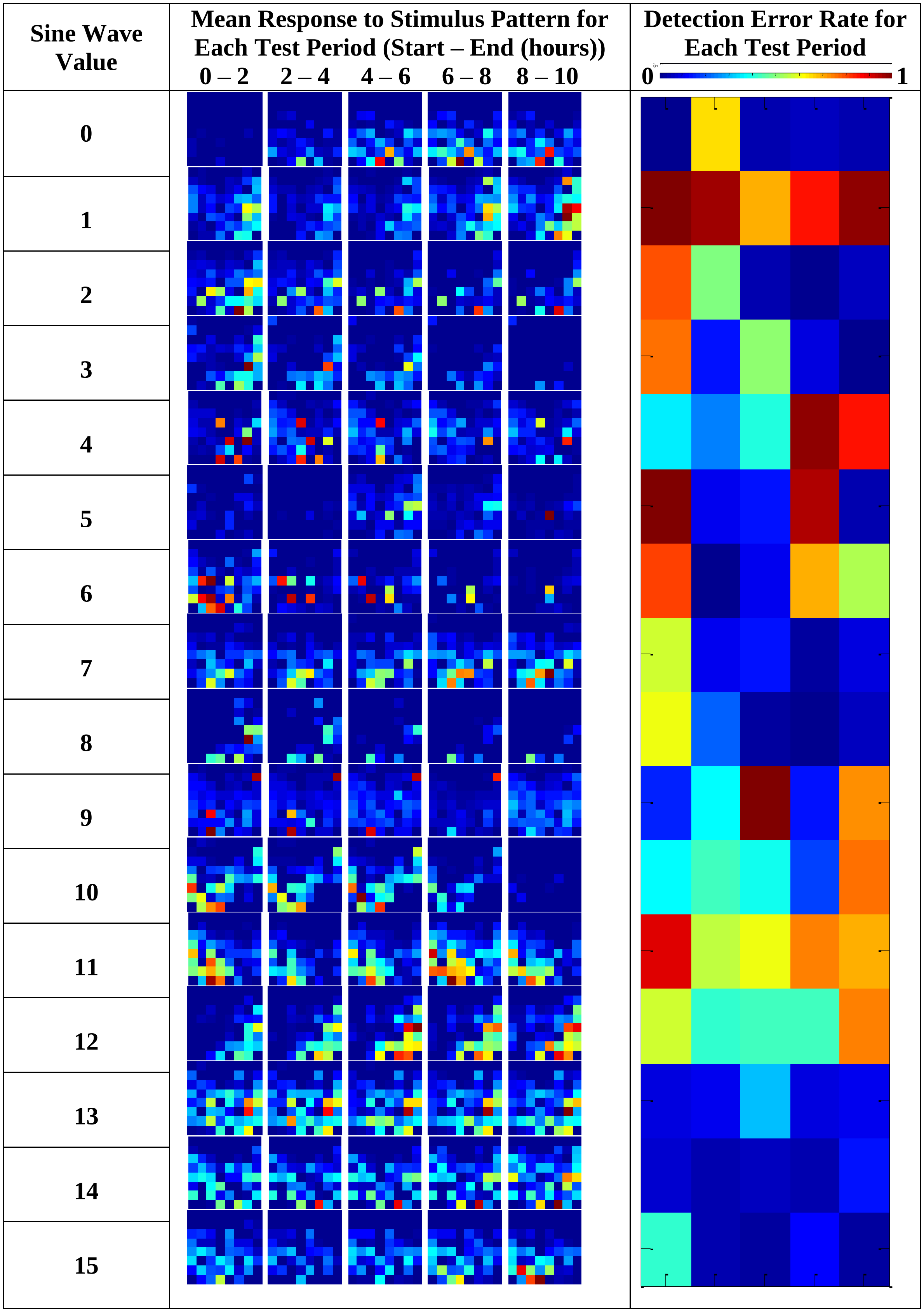

Figure 11: Changes over time in mean response to patterns representing sine wave values

Figure 11 presents detail on the dynamics of the LNN responses to the input patterns encoding the sine wave. The numbers in the left column label the sine wave value represented by a particular input. The middle column presents the mean MEA activity in the response period following stimulation by each pattern. Each square plot in the five columns within the middle column shows the mean responses to stimulation with a particular input pattern (indicated by the values in the first column) over a two-hour time interval in the experiment (approximate start and end times of each interval are indicated at the top of the middle column). I computed the mean activity levels by summing the sampled liquid state vectors for each response to a particular input pattern for each individual electrode over time and then averaging these values for all the responses to that input present in the specified time interval. In order to allow comparison of the changes in magnitude of the response over the time intervals for the same input pattern, I have presented the mean responses for each pattern with values normalized between zero and one (the color bar at the top of the right column shows the color mapping). Zero indicates the least activity, and one indicates the greatest activity within each row of the figure. The right column shows the symbol error rate associated with each input pattern when using the SVM-based decoder retrained during the first 20 minutes of each interval (training region data excluded from decoding results). The error rates shown correspond to the same two-hour intervals plotted in the middle column and use the same color mapping with zero corresponding to 0% symbol error rate, and one corresponding to a 100% symbol error rate.

Overall, the LNN responses to particular inputs do not change significantly for the strong majority of the input patterns, which is beneficial for maintaining a stable encoding scheme. Although the changes in responses that did occur caused the input pattern set to evoke more separable responses (based on improved decoding performance after retraining the classifier) over the course of the experiment (see Figure 10), further studies are necessary to establish whether a neuroplastic tendency, characteristic of learning, is driving this behavior or if it is the result of random response variations. In either case, this algorithm, combined with the type of analysis presented in Figures 10–11, could be used to provide extremely useful feedback to a closed-loop input pattern set optimization algorithm in which individual patterns that do not perform well would be removed and replaced by new patterns whose mean evoked responses are expected to differ substantially from those of the current set (producing greater separability).

The third column of Figure 11 clearly shows that certain patterns were much more likely to be incorrectly decoded: In future studies one could introduce an additional constraint on the sensory mapping algorithm to encode similarly valued samples to patterns that are less separable in order to reduce the impact of decoding errors. Since such a constraint competes with the goal of mapping input signal samples to patterns in order of evoked response magnitude, one must address the resultant optimization problem. It is also interesting to note that later in the experiment, pattern separability and therefore decoding accuracy (when retrained) increased substantially. This may indicate some degree of adaptation to the input stimuli, but further experiments are necessary to extensively evaluate neuroplasticity under these circumstances.

5.2 Short-Term Memory

Figure 12 shows the results of attempting to determine the previous sine wave sample value given only the LNN response to the present sample. I obtained the results by training separate binary SVM classifiers for each sample using the two possible preceding samples for each present sample in the data set as the SVM classes. The leaky integrated spiking responses in the 100 ms following the present samples comprise the training data. I used the mean classification accuracy results presented for five two-hour data sets of sine wave-based stimulation collected successively from Culture A (the experimental data presented in Chapter 5.1). The code set remained constant. I performed SVM training and classification on one-third and two-thirds of the data, respectively, separately for each experiment and then averaged the results. The control (black bars) represents the chance case against which to compare the classification accuracy: The control varies depending on the sample due to the varying distributions of preceding samples for a particular sample value. The same value always precedes some of the samples (values 3–6 and 10–12), preventing the previously described analysis so I removed them from the results presented in Figure 12; all remaining sine wave values have only two possible values preceding them, making a binary SVM classifier sufficient. Nearly half of the tested patterns (sine wave values) were associated with significant classification accuracy for their preceding sample value (150–300 ms in the past) given only the responses to the present input, providing some evidence for short-term memory. I observed greater memory persistence for samples associated with input stimulation patterns evoking large amounts of activity in the network, but there is not enough data to draw robust conclusions about this relationship.

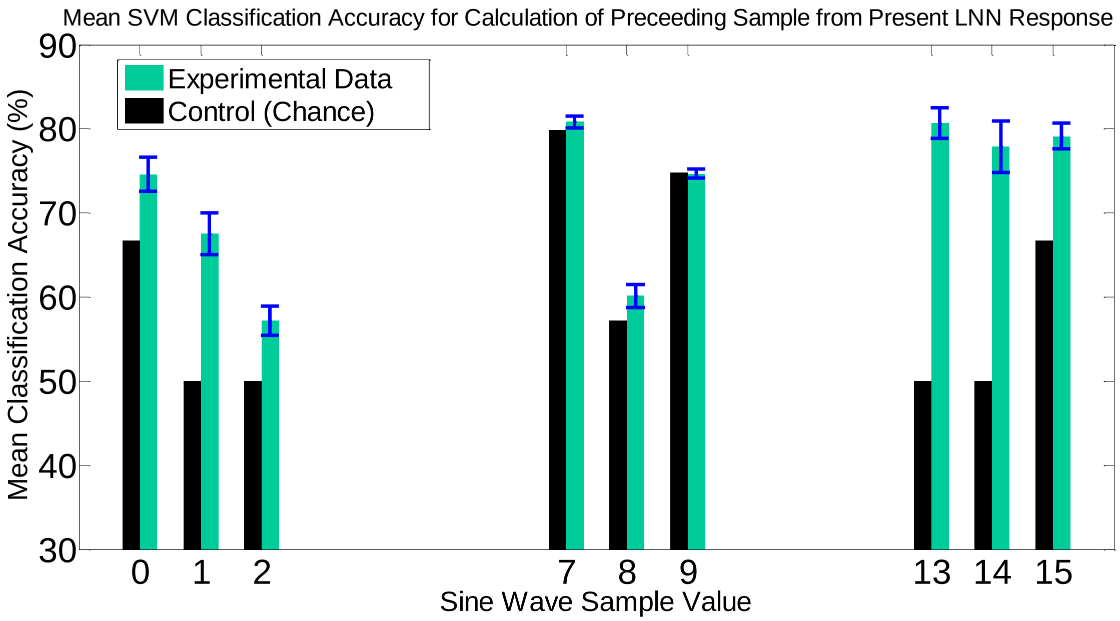

Figure 12: Short-term memory of previous sample value during sine wave I/O experiment. Error bars (blue) indicate ± one standard deviation.

CHAPTER VI

APPLICATIONS AND FUTURE WORK

6.1 Future Work

Researchers conducting future studies could implement a closed-loop pattern optimization system to enhance the functionality of the sensory coding algorithm and provide a mechanism for quickly adapting to changing LNN responses. One could further explore characteristics of pattern separability through new experiments and use the results to enhance such a closed-loop system. Further investigation of neuroplastic changes occurring during long-term stimulation as well as short-term memory could help answer long-standing questions about learning in living neuronal systems and lead to further improvement of sensory coding schemes.

Additionally, input preprocessing may enhance neural information processing and learning. For example, certain types of sensory inputs may benefit from being presented to the LNN as stimuli driven by wavelet coefficients of the signals (instead of time samples), and input preprocessing based on time-delay embeddings such as the Takens' delay embedding theorem (Takens, 1981) might also be useful when one intends to use the LNN to perform prediction on the input. Sensory pathways in living brains are thought to employ multistage processing of sensory input before it reaches regions of high-level pattern recognition and processing (Olshausen & Field, 1997): Certain types of preprocessing of input data prior to its translation into spatio-temporal electrode stimulation sequences could potentially emulate parts of the preprocessing of sensory inputs thought to occur in living brains.

6.2 Power Systems Control Application

The development of high-bandwidth input/output schemes for LNNs serves as a critical prerequisite for the application of LNN-based computation to power system control problems, which are beyond the reach of present computational systems but expected to become more tractable with improved BIANNs (NSF EFRI-COPN Project #0836017). The future goal of power systems control serves as an excellent application with which to evaluate the performance of LNN-based computation on non-linear, non-stationary dynamical systems. Traditional control systems techniques have been used extensively, and ANNs have been explored more recently to control power systems (Venayagamoorthy & Harley, 2002) (Ray & Venayagamoorthy, 2008). However, consumption demands are significantly increasing and variations in sources and loads are becoming more rapid and substantial. Unlike the smaller range of fluctuation produced by traditional sources of power, emerging sources like wind and solar farms produce much wider output variation over time (Venayagamoorthy, 2009) (Saber & Venayagamoorthy, 2009).

Recent studies have successfully applied ANN-based intelligent control algorithms to certain power system control problems (Park, Harley, & Venayagamoorthy, 2003) (Shamsollahi & Malik, 1999) (Flynn et al., 1997). However, the best ANNs have not been able to achieve the degree of optimal control and significant scalability found in biological networks. The development of effective methods for communicating inputs into LNNs and decoding their liquid state information may help guide successful approaches for communicating with artificial spiking neural networks that our colleagues are exploring as a potentially superior replacement for traditional adaptive control techniques.

Regarding the application of the sensory coding algorithm explored in this research to potential future experiments involving attempts to train LNNs to perform prediction and control of various functions and systems, including those associated with power system dynamics, the sensory coding algorithm demonstrates a substantial ability to effectively encode such data for an LNN plated on an MEA. Although the bit rates achieved are less than those typical for real-time prediction/control using ANNs, such a result is expected and acceptable: The purpose of such research is to demonstrate that LNNs are capable of learning to perform prediction and control even if it we cannot use real-time data. The ultimate goal is obviously not to use LNNs on MEAs to perform real-time control of complex power systems but instead to understand and duplicate the superior learning methods of the LNNs in ANNs, which supercomputers could potentially simulate at much faster speeds, achieving real-time prediction and control. If successful, this would combine the learning advantages of living neuronal systems with the speed advantages of electronic computing systems, producing a new generation of intelligent systems with widespread applications.

CHAPTER VII

CONCLUSIONS

My work has established and applied a method for successfully finding spatio-temporal electrical stimulation patterns capable of reliably transferring sensory input information to and from LNNs growing on MEAs. Furthermore, I have developed an effective LNN state readout approach and assessed potential plasticity and memory effects present in the LNNs in the context of the I/O scheme. These findings will enhance the ability of researchers to communicate with small living neuronal cultures, bringing us closer to achieving and studying complex computation in LNNs. Furthermore, the understanding gained from these techniques may directly enhance the usability of biologically inspired artificial spiking neural networks.

REFERENCES

Bakkum, D. J., Chao, Z. C., & Potter, S. M. (2008). Spatio-temporal electrical stimuli shape behavior of an embodied cortical network in a goal-directed learning task. *Journal of Neural Engineering*, *5*(3), 310-323. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/18714127

Brewer, G. J., Torricelli, J. R., Evege, E. K., & Price, P. J. (1993). Optimized survival of hippocampal neurons in B27-supplemented Neurobasal, a new serum-free medium combination. *Journal of Neuroscience Research*, *35*(5), 567-76. Wiley Online Library. doi:10.1002/jnr.490350513

Chang, C.-C., & Lin, C.-J. (2001). LIBSVM: A Library for Support Vector Machines. Retrieved April 30, 2011, from http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Chao, Z. C., Bakkum, D. J., & Potter, S. M. (2007). Region-specific network plasticity in simulated and living cortical networks: comparison of the center of activity trajectory (CAT) with other statistics. *Journal of Neural Engineering*, *4*(3), 294-308. NIH Public Access. doi:10.1088/1741-2560/4/3/015

Demarse, T. B., & Dockendorf, K. P. (2005). Adaptive flight control with living neuronal networks on microelectrode arrays. *Proceedings 2005 IEEE International Joint Conference on Neural Networks 2005*, *3*, 1548-1551. IEEE. doi:10.1109/IJCNN.2005.1556108

Demarse, T. B., Wagenaar, D. A., Blau, A. W., & Potter, S. M. (2001). The Neurally Controlled Animat: Biological Brains Acting with Simulated Bodies. *Autonomous Robots*, *11*(3), 305-310. Springer. doi:10.1023/A:1012407611130

Dockendorf, K. P., Park, I., He, P., Príncipe, J. C., & DeMarse, T. B. (2009). Liquid state machines and cultured cortical networks: the separation property. *Bio Systems*, *95*(2), 90-7. doi:10.1016/j.biosystems.2008.08.001

Flynn, D., McLoone, S., Irwin, G. W., Brown, M. D., Swidenbank, E., & Hogg, B. W. (1997). Neural control of turbogenerator systems. *Automatica, 33*(11), 1961-1973. doi:10.1016/S0005-1098(97)00142-8

Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Sciences, 10*, 484-491.

Geisser, S. (1993). *Predictive Inference* (p. 240). New York, NY: Chapman and Hall/CRC; 1 edition. Retrieved from http://www.amazon.com/Predictive-Inference-

Monographs-Statistics-Probability/dp/0412034719/ref=sr_1_1?ie=UTF8&qid=1322266489&sr=8-1

Hafizovic, S., Heer, F., Ugniwenko, T., Frey, U., Blau, A. W., Ziegler, C., & Hierlemann, A. (2007). A CMOS-based microelectrode array for interaction with neuronal cultures. *Journal of neuroscience methods*, *164*(1), 93-106. doi:10.1016/j.jneumeth.2007.04.006

Hales, C. M., Rolston, J. D., & Potter, S. M. (2010). How to culture, record and stimulate neuronal networks on micro-electrode arrays (MEAs). *Journal of visualized experiments JoVE*. JoVE. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/20517199

Jaeger, H., Lukosevicius, M., Popovici, D., & Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, *20*(3), 335-352. Elsevier. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17517495

Jimbo, Y., & Kawana, A. (1992). Electrical stimulation and recording from cultured neurons using a planar electrode array. *Bioelectrochemistry and Bioenergetics*, *29*(2), 193-204. doi:10.1016/0302-4598(92)80067-Q

Maass, W., Natschläger, T., & Markram, H. (2002). Real-Time Computing Without Stable States. *Neural Computation*, *14*, 2531-2560.

Madhavan, R., Chao, Z. C., Wagenaar, D. A., Bakkum, D. J., & Potter, S. M. (2006). Multi-site stimulation quiets network-wide spontaneous bursts and enhances functional plasticity in cultured cortical networks. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, *1*, 1593-1596. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17946052

Mosteller, F. (1948). A k-Sample Slippage Test for an Extreme Population. *The Annals of Mathematical Statistics*, *19*(1), 58-65. Institute of Mathematical Statistics. Retrieved from http://www.jstor.org/stable/2236056

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*(23), 3311-3325. Elsevier. doi:10.1016/S0042-6989(97)00169-7

Park, J.-W., Harley, R. G., & Venayagamoorthy, G. K. (2003). Adaptive-critic-based optimal neurocontrol for synchronous generators in a power system using MLP/RBF neural networks. *IEEE Transactions on Industry Applications*, *39*(5), 1529-1540. doi:10.1109/TIA.2003.816493

Potter, S. M., & Demarse, T. B. (2001). A new approach to neural cell culture for long-term studies. *Journal of Neuroscience Methods*, *110*(1-2), 17-24. Elsevier. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/11564520

Potter, S. M., Wagenaar, D. A., & Demarse, T. B. (2005). Closing the loop : stimulation feedback systems for embodied mea cultures. (M. Taketani & M. Baudry, Eds.)*Data Processing*, 215–242. Springer. doi:10.1007/0-387-25858-2_9

Ray, S., & Venayagamoorthy, G. K. (2008). Real-time implementation of a measurement-based adaptive wide-area control system considering communication delays. *Engineering and Technology*, *2*(1), 62- 70. doi:10.1049/iet-gtd

Rolston, J. D. (n.d.). Creating NeuroRighter. Retrieved November 25, 2011, from http://groups.google.com/group/neurorighter-users?pli=1

Rolston, J. D., Gross, R. E., & Potter, S. M. (2009a). A Low-Cost Multielectrode System for Data Acquisition Enabling Real-Time Closed-Loop Processing with Rapid Recovery from Stimulation Artifacts. *Frontiers in neuroengineering*, *2*(July), 17. Frontiers Research Foundation. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2722905&tool=pmcentrez&rendertype=abstract

Rolston, J. D., Gross, R. E., & Potter, S. M. (2009b). NeuroRighter: closed-loop multielectrode stimulation and recording for freely moving animals and cell cultures. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, *2009*, 6489-6492. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/19964440

Ruaro, M. E., Bonifazi, P., & Torre, V. (2005). Toward the neurocomputer: image processing and pattern recognition with neuronal cultures. *IEEE Transactions on Biomedical Engineering*, *52*(3), 371-383. IEEE. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15759567

Saber, A. Y., & Venayagamoorthy, G. K. (2009). One million plug-in electric vehicles on the road by 2015. *2009 12th International IEEE Conference on Intelligent Transportation Systems*, (april), 1-7. IEEE. doi:10.1109/ITSC.2009.5309691

Schölkopf, B., Tsuda, K., & Vert, J.-P. (2004). *Kernel Methods in Computational Biology* (p. 400). MIT Press. Retrieved from http://books.google.com/books?id=SwAooknaMXgC&pgis=1

Shamsollahi, P., & Malik, O. P. (1999). Real-time implementation and experimental studies of a neural adaptive power system stabilizer. *IEEE Transactions on Energy Conversion*, *14*(3), 737-742. doi:10.1109/60.790944

Takens, F. (1981). Detecting strange attractors in turbulence. (D. A. Rand & L. S. Young, Eds.)*Dynamical Systems and Turbulence*, *898*(1), 366-381. Springer. doi:10.1007/BFb0091903

Taketani, M., & Baudry, M. (2010). *Advances in Network Electrophysiology: Using Multi-Electrode Arrays* (Vol. 2010, p. 496). Springer. Retrieved from http://books.google.com/books?id=4nOkcQAACAAJ&pgis=1

Venayagamoorthy, G. K. (2007). Online design of an echo state network based wide area monitor for a multimachine power system. *Neural Networks*, *20*(3), 404-413. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17513088

Venayagamoorthy, G. K. (2009). Potentials and promises of computational intelligence for smart grids. *2009 IEEE Power Energy Society General Meeting*, 1-6. IEEE. doi:10.1109/PES.2009.5275179

Venayagamoorthy, G. K., & Harley, R. G. (2002). A continually online trained neurocontroller for excitation and turbine control of a turbogenerator. *IEEE Transactions on Energy Conversion*, *16*(3), 1263-1267. IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1003445

Wagenaar, D. A., Madhavan, R., Pine, J., & Potter, S. M. (2005). Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation. *Journal of Neuroscience*, *25*(3), 680-688. Soc Neuroscience. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15659605

Wagenaar, D. A., Nadasdy, Z., & Potter, S. M. (2006). Persistent dynamic attractors in activity patterns of cultured neuronal networks. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, *73*(5 Pt 1), 051907. APS. Retrieved from http://link.aps.org/doi/10.1103/PhysRevE.73.051907

Wagenaar, D. A., Pine, J., & Potter, S. M. (2004). Effective parameters for stimulation of dissociated cultures using multi-electrode arrays. *Journal of Neuroscience Methods*, *138*(1-2), 27-37. Elsevier. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/15325108

Wagenaar, D. A., Pine, J., & Potter, S. M. (2006). An extremely rich repertoire of bursting patterns during the development of cortical cultures. *BMC neuroscience*, *7*(1), 11. doi:10.1186/1471-2202-7-11

Wagenaar, D. A., & Potter, S. M. (2002). Real-time multi-channel stimulus artifact suppression by local curve fitting. *Journal of Neuroscience Methods*, *120*(2), 113-120. Elsevier. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12385761