Low-power machine learners for implantable decoding

Shaikh, Shoeb Dawood

2021

Shaikh, S. D. (2021). Low-power machine learners for implantable decoding. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/146463>

<https://doi.org/10.32657/10356/146463>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 21:50:09 SGT

Low-power Machine Learners for Implantable Decoding

Shaikh Shoeb Dawood

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

27-01-21

.....

Date

.....

Shaikh Shoeb Dawood

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

27-01-21

.....

Date

.....

Prof. Arindam Basu

Authorship Attribution Statement

This thesis contains material from 7 publications published in the following peer-reviewed journals/from papers accepted at conferences in which I am listed as an author.

Chapter 1 contains parts of material published as [S. Shaikh and A. Basu](#), “Intelligent Intracortical Brain-Machine Interfaces (i^2BMI): Next generation of Scalable Neural Interfaces,” [Springer Handbook of Biochips: Integrated Circuits and Systems for Biology and Medicine](#), edited by Mohamad Sawan.

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I did literature review and prepared the manuscript drafts.

Chapter 2 contains parts of material published as [S. Shaikh, R. So, C. Libedinsky and A. Basu](#), “Experimental Comparison of Hardware-Amenable Spike Detection Algorithms for iBMIs,” [2019 9th International IEEE/EMBS Conference on Neural Engineering \(NER\)](#), San Francisco, CA, USA, 2019, pp. 754-757.

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Rosa So, Dr. Tafadzwa Sibindi and Asst Prof Camilo Libedinsky.
- I co-designed the study with A/Prof Arindam Basu and Dr. Rosa So, performed the experimental analysis and reported the findings.

Chapter 3 is published as [S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu](#), “Towards Intelligent Intracortical BMI (i^2BMI): Low-Power Neuromorphic Decoders That Outperform Kalman Filters,” in [IEEE Transactions on Biomedical Circuits and Systems](#), vol. 13, no. 6, pp. 1615-1624, Dec. 2019, doi: 10.1109/TB-CAS.2019.2944486.

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Rosa So, Dr. Tafadzwa Sibindi and Asst Prof Camilo Libedinsky.

- I co-designed the study with A/Prof Arindam Basu and Dr. Rosa So, performed the experimental analysis and reported the results.

Chapter 4 is published as [S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, “Real-time Closed Loop Neural Decoding on a Neuromorphic chip,” 2019 9th International IEEE/EMBS Conference on Neural Engineering \(NER\), San Francisco, CA, USA, 2019, pp. 670-673, doi: 10.1109/NER.2019.8717122.](#)

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Rosa So, Dr. Tafadzwa Sibindi and Asst Prof Camilo Libedinsky.
- I co-designed the study with A/Prof Arindam Basu and Dr. Rosa So.
- I conducted the real-time experiments and was guided by Asst Prof Camilo Libedinsky, Dr. Tafadzwa Sibindi and Dr. Rosa So.

Chapter 5 is published as [S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, “Sparse Ensemble Machine Learning to Improve Robustness of Long-Term Decoding in iBMIs,” in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 28, no. 2, pp. 380-389, Feb. 2020, doi: 10.1109/TNSRE.2019.2962708.](#)

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Rosa So, Dr. Tafadzwa Sibindi and Asst Prof Camilo Libedinsky.
- I co-designed the study with A/Prof Arindam Basu and Dr. Rosa So, performed the experimental analysis and reported the results.

Chapter 6 is submitted for publication as [S. Shaikh, C. Yi, A. Basu and R. So, “Lightweight Reinforcement Algorithms for Autonomous, Scalable, Neuromorphic intracortical Brain Machine Interfaces,” Neuromorphic Computing and Engineering, and a part of it is published as S. Shaikh, C. Yi, A. Basu and R. So, “Towards Autonomous Intracortical Brain Machine Interfaces: Applying Bandit Algorithms for Online Reinforcement Learning,” 2020 IEEE International Symposium on Circuits and Systems Conference \(ISCAS\), Seville, 2020.](#)

The contributions of the co-authors are as follows:

- A/Prof Arindam Basu provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Rosa So, Dr. Tafadzwa Sibindi and Asst Prof Camilo Libedinsky.
- I co-designed the study with A/Prof Arindam Basu and Dr. Rosa So, performed the experimental analysis and reported the results.

27-01-21

.....

Date

.....

Shaikh Shoeb Dawood

Acknowledgements

I consider myself extremely fortunate to have been inducted in Prof. Arindam Basu's BRAIN group. I would like to express my heartfelt gratitude to my supervisor, Prof. Basu for mentoring me throughout my PhD journey. He is a very kind mentor, who always stands by his students through thick and thin. His intellectual curiosity and infectious enthusiasm towards research have been a constant source of motivation for me. His ability to think through tough problems and communicate complex ideas in simple ways have always left me amazed and inspired. In the hyper-competitive world of academia, Prof. Basu's endless patience and temperament in guiding students is truly laudable.

This work would not have been possible without the helpful guidance of my co-supervisor, Dr. Rosa Qi Yue So. I would like to thank Dr. Rosa for introducing me to the exciting world of live animal research. Her consistent feedback and practical hands-on tips have been instrumental in shaping this work. I would like to thank BRAIN group's collaborator Prof. Camilo Libedinsky for his helpful suggestions, feedback and unrestricted use of experimental equipment. I greatly admire his cool demeanour and the desire to pursue profound scientific questions. I would like to thank my colleagues at NUS - Dr. Tafadzwa Sibindi, Abdur Rauf, Clement Lim who were of great help in setting up experiments. A Special thanks to Datou for painstakingly persevering through the research tasks. I would like to acknowledge the timely and helpful feedback given by my thesis advisory committee members, Prof. Cuntai Guan and Prof. Boon Chirn Chye. Thanks to Prof. Kai Keng Ang for sharing interesting research-related anecdotes and advice during the biweekly meetings at the Institute for Infocomm Research, A*STAR.

A special acknowledgement to my MSc thesis advisor, Dr. Mihai Sanduleanu for opening up the world of research to the younger me. His philosophy of 'divide-and-conquer' in solving research problems is something that I have carried with me through my PhD years. Closer home in Mumbai, I would like to thank Principal

Mrs. Jean Gomes and her team at Michael High School for their tireless and admirable educational work in the challenging environs of Bombay-70.

At BRAIN lab and its allied VIRTUS lab, I was lucky to be among some of the best technically inclined and fun peers. Thanks to Chen Yi, Aakash, Sumon, Roshan, Rahul, Nimesh, Jyotibdha, Vignesh, Prateek, Rishi, Pradeep, Ruchi, Govind, Vivek for all the technical and not so technical discussions. Staying abroad at times can be quite lonely and daunting. Luckily, such times were rare as I was in the good company of Moheed, Maruf, Sandesh, Tariq, Aashique, Khursheed, Saif, Nimesh, Shravan, Tauquir, Arif, Uvesh, EEE-GSC club members and musollah friends, among many others. Thank you all for your friendship.

Finally, I would like to thank my family - mom (Shahnaz), dad (Dawood), brother (Haris) and my late grandmother (Zaibun Nisa) for always being a pillar of strength. Thank you for giving me the freedom to pursue my passion. Lastly, I would like to thank my kind and lovely wife, Salma, for all the joy she has brought to my life.

Shoeb Shaikh, January 2021

“Sitāroñ se aage jahāñ aur bhī haiñ, abhī ishq ke imtihāñ aur bhī haiñ”

—Iqbal

Abstract

Approximately 6 million people in the US and roughly 1 in 50 people worldwide suffer from paralysis. This amounts to a significant number of people dependent on assistance from caregivers for daily living. The need to restore function in these people to perform activities for daily living has motivated development of a host of assistive technologies. One of the most promising technologies is the intracortical Brain Machine Interface (iBMI). This technology involves sensing intracortical signals from the surface of the motor cortex which are then passed through several signal processing and decoding algorithmic blocks to drive effectors such as a computer cursor for communication, robotic/natural arm for feeding, wheelchair for locomotion among others.

Despite these convincing demonstrations, significant problems persist in the adoption of these technologies for daily use. Firstly, the present systems involve the use of wires for data transmission impeding the mobility of the system as a whole and increasing the risk of infection. Secondly, the decoding algorithms are currently being implemented on a PC increasing the bulkiness of the overall system. Thirdly, the decoding models need to be calibrated daily, often involving lengthy procedures, thereby resulting in frustration for the end-user. Fourthly, researchers have arrived at the conclusion that it is imperative to simultaneously record from 10X or more neurons to improve decoding algorithms in BMI as well as gain fundamental insight into effects of long-term BMI usage. Lastly, cosmesis is also a major concern towards eventual adoption of these technologies. The form factor and visibility of the current systems run the risk of aggravating stigma among patients.

To solve the problems of wires, portability and cosmesis, fully implantable wireless iBMI systems have been touted as a reasonable solution. However, scaling the number of channels in such systems by another 10X is difficult due to power and bandwidth requirements of the wireless transmitter. One promising solution for that is to include more processing, up to the decoder, in the implant so that

transmission data-rate is reduced drastically. Earlier work on neuromorphic decoder chips only showed classification of discrete states. We present results for continuous state decoding using a low power neuromorphic decoder chip termed **S**pike-input **E**xtr_em_e **L**earning **M**achine (SELMA) that implements a nonlinear decoder without memory and its memory based version with time delayed bins, SELMA-bins. We have compared SELMA, SELMA-bins against state-of-the-art **S**teady **S**tate **K**alman **F**ilter (SSKF), a linear decoder with memory, across two different datasets involving a total of 4 non-human primates (NHPs). Results show at least a 10% (20%) or more increase in the fraction of variance accounted for (FVAF) by SELMA (SELMA-bins) over SSKF across the datasets. Estimated energy consumption comparison shows SELMA (SELMA-bins) consuming $\approx 9 \text{ nJ/update}$ (23 nJ/update) against SSKF's $\approx 7.4 \text{ nJ/update}$ for an iBMI with a 10 degree of freedom control. Thus, SELMA yields better performance against SSKF while consuming energy in the same range as SSKF whereas SELMA-bins performs the best with moderately increased energy consumption, albeit far less than energy required for raw data transmission. Furthermore, to test for closed loop performance, we present for the first time a real-time closed loop neuromorphic decoder chip-driven iBMI in a NHP based experimental setup. Decoded results show trial success rates and mean times to target comparable to those obtained by hand-controlled joystick. Neural control trial success rates of $\approx 96\%$ of those obtained by hand-controlled joystick have been demonstrated. Also, neural control has shown mean target reach speeds of $\approx 85\%$ of those obtained by hand-controlled joystick. These results pave the way for reducing transmission data rates in future scaled iBMI systems.

In order to resolve the problem of lengthy daily recalibration procedures, we present a novel sparse ensemble based machine learning approach to enhance robustness of iBMIs in the face of non-stationary distribution of input neural data across time. Each classifier in the ensemble is trained on a randomly sampled (with replacement) set of input channels. These sparse connections ensure that with a high chance, few of the base classifiers should be less affected by the variations in some of the recording channels. We have tested the generality of this technique on different base classifiers - linear discriminant analysis (LDA), support vector machine (SVM), extreme learning machine (ELM) and multilayer perceptron (MLP). Results show decoding accuracy improvements of up to $\approx 21\%$, 13% , 19% , 10% in experiment 3 and 7% , 9% , 7% , 9% in experiment 4 across test days while using the sparse ensemble approach over a single classifier model for LDA, SVM, ELM

and MLP algorithms respectively. Furthermore, improvements of up to $\approx 7(14)\%$, $8(15)\%$, $9(19)\%$, $7(15)\%$ in experiment 3 and $8(15)\%$, $12(20)\%$, $15(23)\%$, $12(19)\%$ in experiment 4 over Random Forest (Long-short Term Memory) have been obtained by sparse ensemble LDA, SVM, ELM, MLP respectively.

iBMIs are closed loop systems with real-time visual feedback available to the end user. In order to employ co-adaptation between the user and the iBMI system, we have explored the paradigm of online reinforcement learning. This paradigm is amenable towards increasing the nature of autonomy and decoding robustness of iBMIs against non-stationarity, compared to the prevalent supervised methods. In particular, using offline open-loop analysis on pre-recorded data, we show application of Banditron algorithm in discrete-state iBMIs and compare it against previously reported state-of-the-art RL algorithms – Hebbian RL, Attention gated RL, deep Q-learning. Post-hoc analysis has been performed on four pre-recorded experimental datasets involving multiple days of neural data procured from motor cortex of two non-human primates performing joystick-based movement-related tasks. Results indicate Banditron performing significantly better than state-of-the-art RL algorithms by at least $\approx 5\%$, 10% , 7% and 7% in experiments 5, 6, 7 and 8 respectively. In terms of hardware complexity, Banditron is found to yield at least an order of magnitude of reduction in the number of Multiply-and-Accumulate operations (MACs) compared to state-of-the-art RL algorithms. Furthermore, average improvements of at least 39% , 15% , 55% , 50% have been obtained in experiments 5, 6, 7 and 8 respectively over popularly used supervised classifier models - Linear Discriminant Analysis and Support Vector Machines. These results establish the competency of Banditron in yielding improvements over state-of-the-art algorithms presented in iBMI literature. Thus, we posit that with further work in the area of obtaining a reliable feedback signal, we are well poised to design autonomous iBMI systems.

Contents

Acknowledgements	xi
Abstract	xv
List of Figures	xxiii
List of Tables	xxxi
Acronyms	xxxiii
1 Introduction	1
1.1 Motivation and Background	1
1.2 Neurotechnology	2
1.3 Brain Machine Interface	3
1.3.1 Types of BMI	5
1.3.2 Why intracortical BMI?	6
1.3.3 Problems with state-of-the-art iBMIs	7
1.4 Objectives	12
1.5 Outline	13
2 Intracortical Brain Machine Interface	15
2.1 Action Potential - The Input Neural Signal	16
2.2 Recording Sensor - Microelectrode Array	18
2.3 Signal acquisition front-end	20
2.4 Feature Extraction	22
2.4.1 Spike Detection	23
2.4.1.1 Absolute value method	24
2.4.1.2 Nonlinear Energy Operator (NEO)	24
2.4.1.3 Wavelet Analysis	25
2.4.2 Spike Sorting	26
2.4.3 Local Field Potential	27
2.5 Decoder	28
2.5.1 Discrete state	29
2.5.2 Continuous state	30

2.5.3	Hybrid	31
2.6	Effector	32
2.7	Summary	32
3	Towards intelligent intracortical iBMIs (i^2BMI): An offline study	34
3.1	Introduction	34
3.2	Materials and Methods	36
3.2.1	Experiment 1	37
3.2.2	Experiment 2	37
3.3	Decoding methodology and algorithms	38
3.3.1	Algorithms	38
3.3.1.1	State-of-the-art - Kalman Filter	38
3.3.1.2	A Neuromorphic Alternative - Extreme Learning Machine	40
3.3.1.3	SELMA - A Neuromorphic Processor	42
3.3.1.4	ELM with memory	42
3.4	Measured Results	44
3.4.1	Analysis Methodology	44
3.4.2	Experimental Setup	45
3.4.3	Results	45
3.4.3.1	Sweeping bin-widths	45
3.4.3.2	Sweeping training times	46
3.4.3.3	Statistical tests	47
3.4.4	Comparison of Computational Complexity	49
3.5	Discussion	52
3.6	Conclusion	55
4	Towards intelligent intracortical iBMIs (i^2BMI): An online study	57
4.1	Introduction	57
4.2	Methods	58
4.2.1	Neural Signal Acquisition and Signal Processing	58
4.2.2	Behavioural Task	58
4.2.2.1	Native hand Control	58
4.2.2.2	Brain-Decoder Control	59
4.2.3	Real-time interface	60
4.2.3.1	SELMA - training mode	61
4.2.3.2	SELMA - decoding mode	62
4.3	Results	64
4.4	Discussion and Limitations	66
4.5	Conclusion	66
5	Sparse Ensemble Machine Learning to improve robustness of long-term decoding in iBMIs	67
5.1	Introduction	67

5.2	Materials and Methods	70
5.2.1	Signal Acquisition and Processing	71
5.2.1.1	Experiment 3	71
5.2.1.2	Experiment 4	73
5.2.2	Behavioural Tasks	73
5.2.2.1	Experiment 3	74
5.2.2.2	Experiment 4	74
5.3	Proposed Decoding Algorithm	74
5.3.1	Analysis Methodology	74
5.3.2	Sparse Ensemble Machine Learning	76
5.3.2.1	Method 1	78
5.3.2.2	Method 2	78
5.4	Results	78
5.4.1	Choice of Hyperparameters	79
5.4.2	Base Learners training	81
5.4.3	Decoding Results	83
5.5	Discussion	87
5.5.1	Comparison	87
5.5.2	Ensemble vs Sparse Ensemble	88
5.5.3	Combining Strategy: Sum vs Voting	89
5.5.4	Custom neural recording tools	89
5.6	Conclusion	90
6	Online Reinforcement-based Bandit Algorithms for autonomous Intracortical Brain Machine Interfaces	91
6.1	Introduction	91
6.2	Materials and Methods	95
6.2.1	Signal Acquisition and Processing	95
6.2.2	Behavioural Tasks	96
6.2.2.1	Experiment 5	96
6.2.2.2	Experiment 6	96
6.2.2.3	Experiment 7	97
6.2.2.4	Experiment 8	97
6.2.3	Pre-recorded dataset inputs and outputs	97
6.2.4	Datasets and train-test split	98
6.3	Reinforcement Learning Algorithms	99
6.3.1	State-of-the-art	99
6.3.1.1	AGREL	99
6.3.1.2	HRL	101
6.3.1.3	Deep Q-learning	102
6.3.2	Proposed Algorithm	103
6.3.2.1	Banditron	103
6.3.2.2	Banditron-RP	104

6.4	Results	104
6.4.1	Summary of models and Hyperparameter Tuning	105
6.4.2	Decoding Results - Ideal feedback	106
6.4.3	Decoding Results - Erroneous and sparse feedback	107
6.5	Discussion	110
6.5.1	Comparison of Computational Complexity	110
6.5.2	Comparison to state-of-the-art RL algorithms	113
6.5.3	Comparison to state-of-the-art supervised learning paradigm	114
6.5.4	Towards Autonomous iBMIs	114
6.6	Conclusion and Limitations	115
7	Contributions, Limitations and Future Work	116
7.1	Summary of Contributions	116
7.1.1	Towards portable, scalable and mobile iBMI systems	116
7.1.2	Mitigating non-stationarity	116
7.1.3	Lending independence to iBMI users	117
7.2	Limitations and Future Work	117
7.2.1	Real-time closed loop experiments	117
7.2.2	Implementation of a hardware implant	118
7.2.3	Biological Reward Signal	118
7.2.4	On-chip learning	118
7.2.5	Closed-loop stimulation	118
	List of Author's Publications	121
	Bibliography	123

List of Figures

1.1	Shows number of people worldwide suffering from neural disorders. It also shows a rough mapping between disorders and different regions of the brain	2
1.2	Shows examples of neural interfaces and their implantation locations [8] (CC BY 3.0 license)	3
1.3	Shows constituent blocks of a brain machine interface (BMI). The system takes neural activity as an input, which is then processed to obtain suitable features to drive effector such as an artificial limb, a computer cursor, a wheelchair etc. Typically users can visually see an effector's movement and this serves as a feedback mechanism. Other modes of feedback such as neural stimulation are also being explored to improve the overall efficiency of BMI systems.	4
1.4	Shows different techniques for acquiring signal from the brain. Different colours indicate approximate limits of depth and physical scale of activity captured in the respective techniques. Figure adapted from [14]	5
1.5	Illustration of an intracortical Brain Machine Interface (iBMI) [47] .	8
1.6	Duration of chronic human iBMIs reported across clinical studies [71]	10
1.7	Panel of action potentials recorded at input electrodes	11
1.8	Shows illustrations of different brain machine interface system options presented to the respondents in the survey [84]	12
1.9	Shows mapping between objectives and chapters in this thesis . . .	13
2.1	Typical block diagram of an iBMI system consisting of signal acquisition, feature extraction, decoder and effector blocks. Signal acquisition often takes place in the implant while the other blocks are implemented externally.	15
2.2	Simplified view of structure of neuron, adapted from [87].	16
2.3	Action potential leads to neurotransmitters being released across synaptic cleft thereby resulting in an electrical signal in the postsynaptic neuron [88].	17
2.4	Action potential results when the sum total of inhibition and excitation makes the membrane threshold surpass the threshold [88]. . .	17
2.5	(a) Shows a 10×10 grid of Utah array along with insulated gold wires that percutaneously connect to an external connector, (b) and (c) show side-view and backside of the Utah array, adapted from [59].	18

2.6	Shows two Utah arrays implanted in the motor and posterior cortex of a non-human primate. The implantation was done to be employed in a brain machine interface study, adapted from [92].	19
2.7	Example of neural signal captured by the Utah array, adapted from [93].	19
2.8	Shows popular architectures reported to realize neural front-ends. (a) This architecture shares one ADC across all input channels. (b) In this architecture, each input channel has its own ADC. (c) This architecture is a compromise between (a) and (b) where one ADC is shared between m channels, where m is a subset of total input channels.	21
2.9	Shows two popular methods to extract features from extracellular recordings - Local Field Potential (LFP) and Spikes. Figure is adapted from [99] (CC BY 4.0 license).	22
2.10	Shows the signal processing chain involved in detecting spikes. The analog front-end's raw neural signal is subjected to bandpass filtering, followed by an optional pre-emphasis step. Thereafter, thresholding is done to obtain unsorted spikes also known as Threshold crossings.	23
2.11	Shows pictorially how spikes emanating from two different neurons are sorted after being captured by the same electrode, adapted from [99].	26
2.12	Shows the signal processing chain involved in computing Local Field Potential (LFP). Low pass filtering with a cut off frequency of around 100 Hz of the analog front end signal is done to arrive at the LFP signal.	28
2.13	(a) Shows a motor imagery based experiment in which the participant is asked to imagine one of the four movements - index flexion, index extension, wrist flexion and wrist extension. (b) Shows a convolutional neural network (CNN) plus long-short term memory (LSTM) based decoder to infer the intended movement. Figure has been adapted from [78].	30
2.14	Shows a recurrent neural network used to predict movement of native hand of a non-human primate. Specifically movement parameters include two dimensional velocity ($v_x(t), v_y(t)$) and position ($p_x(t), p_y(t)$). Figure adapted from [49].	31
2.15	Shows a point and click type hybrid iBMI system employed by a subject to reply to an email on a commercial tablet, adapted from [83].	32
3.1	Block level representation of (a) conventional and (b) proposed wireless iBMI systems	35

3.2	NHPs A, B control a cursor on screen through a planar manipulator to reach targets flashed at random locations on a screen in a continuous manner one after another. A brief hold period is maintained between reaches. Image is obtained courtesy [145] (CC-BY license).	37
3.3	ELM neural network architecture with $\vec{X}$, $\vec{H}$ and $\vec{o}$ representing input, hidden and output layers respectively. $\mathbf{W}_{in}$, β represents first and second layer weights while $g(\cdot)$ stands for the activation function	40
3.4	(a) ELM chip-level architecture [143] is presented here. Firing rates are computed from spikes received as inputs by input window counters. Digital-Analog converters translate the firing rates into currents in each input channel. Current mirror array is used to realize first layer random projection. Current controlled oscillator is employed at the hidden nodes to yield a value proportional to the current at the respective hidden node. Second layer multiplication is implemented on a MSP430 microcontroller (MCU) in this work but can be integrated on-chip as well[165], (b) Picture of the test board in the form of a Portable External Unit (PEU) made up of machine learning co-processor (MLCP) and MCU is presented here [143]. MLCP implements the first layer of computation while MCU is employed for second layer computation.	43
3.5	Example segment of decoded 2D components of velocities in NHP B comparing SSKF, SELMA and SELMA-bins decoders. SELMA, SELMA-bins outperform SSKF in the above time snippet of ≈ 3 seconds. state-of-the-art decoded variable employed in iBMIs is the velocity of the effector [44, 48, 138, 149, 150] and hence we have used velocity to compare decoders.	46
3.6	Linear Regression(LR), KF, SSKF, SELMA and SELMA-bins (optimal number of bins = 3,3,2,3 for NHPs A,B,C,D respectively) models are trained on the first half period of respective datasets and the remaining half is used as testing set in (a)-(d). Plots show decoding results in the form of averaged R^2 across 2D components of movement velocity, as bin-widths are swept for computing firing rates. The proposed neuromorphic decoder - SELMA and its memory based version - SELMA-bins outperform LR, KF and SSKF decoders on all four datasets. Error bars denote standard deviation of results for randomly chosen input weights over 20 iterations for SELMA and SELMA-bins.	47
3.7	Plots capture Mean Squared Error (MSE) while varying binwidths for different classifiers for NHPs A, B, C and D in (a), (b), (c) and (d) respectively. The plotted results corroborate FVAF based findings (Figure 3.6) wherein SELMA and SELMA-bins outperform other classifiers.	48

3.8	Plots (a)-(d) show impact of amount of training data on decoding capability of different decoders. For each decoder, bin-width corresponding to highest R^2 in Figure 3.6 is used. The proposed neuro-morphic decoder - SELMA and its memory based version - SELMA-bins outperform LR, KF and SSKF decoders on all four datasets. Error bars denote standard deviation of results for randomly chosen input weights over 20 iterations for SELMA and SELMA-bins.	49
3.9	Plots capture Mean Squared Error (MSE) while varying training data for different classifiers for NHPs A, B, C and D in (a), (b), (c) and (d) respectively. The plotted results corroborate FVAF based findings (Figure 3.6) wherein SELMA and SELMA-bins outperform other classifiers.	50
3.10	Validation accuracy is plotted for varying values of number of hidden nodes for NHP A's training data with 64 input channels. R^2 saturates around 70 hidden layer neurons in this case.	51
3.11	Comparison of transmission data rates between systems transmitting a) raw data, b) firing rates and c) decoded outputs for a 10D iBMI operating at 50 Hz. It can be seen that the system with decoded outputs is more scalable since its output data rate is orders of magnitude lower than both other options.	53
4.1	Task - A wheelchair avatar starts from a central position on the computer screen in every trial with a target displayed pseudo-randomly in one of the three square-shaped locations.	59
4.2	Experimental setup for closed loop decoding with SELMA. Spikes are detected by the Ripple neural recording system and passed on to SELMA for subsequent decoding. Universal Asynchronous Receiver/Transmitter (UART) is used to establish communication between the microcontroller on SELMA board and the Ripple neural recording system. The final decoded output is passed to the stimulus PC via an ethernet cable. The stimulus PC runs a Psychophysics toolbox based virtual wheelchair program.	60
4.3	The TI-MSP430 microcontroller on board the SELMA-chip is interrupted by the digital trigger signal at the beginning of every trial. Thereafter, the hidden layer vectors are saved in microcontroller memory every 100 <i>ms</i> until the end of the trial. These hidden layer vectors are read back at the end of every trial to eventually make up the hidden layer matrix.	63
4.4	The learned weight matrix β is loaded in the microcontroller at the beginning of a decoding session. Decoded outputs from SELMA-chip are used to drive the wheelchair avatar on the stimulus PC at the frequency of 10 Hz.	64
4.5	Mean times to complete trials are comparable for closed-loop decoding by SELMA and hand controlled joystick experiments. Statistically insignificant difference is denoted by (*), $p > 0.01$ (unpaired two-sample t-test)	65

4.6	Percentage of successful trials for hand-controlled joystick and neural control. Statistically insignificant difference is denoted by (*), $p > 0.01$ (unpaired two-sample t-test)	65
5.1	Micro-motion of an electrode array leads to appearance/disappearance of a neuron's signals at a particular electrode as seen in recording conditions 1,2. Furthermore, in a more dire scenario as seen in recording condition 3, certain electrodes break down and stop yielding any meaningful recorded signal. Figure is not drawn to scale and is adapted from [49] (CC-BY license).	68
5.2	The principal angle is a parsimonious scalar metric to measure the similarity of neural activity across days [49, 192]. (a) and (b) present principal angles between the subspaces of neural activity recorded each day for NHPs E and F respectively. Smaller principal angles denote greater similarity and one can see recorded days being more similar to each other in NHP F than NHP E. (c) presents measured signal to noise ratio (SNR) values following the methodology given in [125] with mean (solid line) and standard deviation (shaded region) values across electrodes plotted for recorded days in NHPs E and F. As far as signal stability is concerned, we can see the average SNR values across days to be stable across days for both NHPs. NHP E shows greater variance in the SNR values among its electrodes compared to NHP F, showing that some electrodes capture better spikes than the others.	69
5.3	(a) Shows an NHP trained to control a joystick. Joystick movement was restricted to right, left and forward directions only. In the top portion, a multi-electrode array is shown to be implanted in the hand and arm regions of primary motor cortex [123], (b) Experiment 3 setup: NHP E was seated in a robotic wheelchair. The wheelchair translated in the forward, left or right directions in discrete time steps of 100 ms depending on the position of the joystick [123], (c) Experiment 4 setup: NHP F is trained to drive a virtual wheelchair using a joystick in a manner similar to Experiment 3. Figures (a), (b) are adapted from [123] (CC-BY license) and Figure (c) is adapted from [198] (CC-BY license)	72
5.4	Partitioning of data for training and testing to benchmark fixed models against daily retrained models	75
5.5	Schematic representation of sparse ensemble machine learning is presented here. Input to each of the M Base Learners (BLs) - $\vec{X}^j$ is obtained by sparsely sampling the entire input space - $\vec{X}$ with replacement. Outputs $\vec{o}^j$ from all M constituent BLs are combined to obtain the final decoded vector $\vec{o}$	77

5.6	k-fold cross-validation to obtain optimal value of number of inputs (U) to each base learner as per Equation (5.6) for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h). The results are obtained for different number of base learners (M) swept from 10 to 25 across different classifiers. The optimal value of U comes out to be 14 ($q = 2$) and 35 ($q = 5$) across classifiers for NHP E and NHP F respectively.	80
5.7	k-fold cross-validation to obtain optimal value of number of base learners (M) for NHP E and NHP F across different classifiers for the value of U obtained in Figure 5.6. The optimal value of M comes out to be around 13 for both NHPs.	81
5.8	<i>Unperturbed</i> decoder results across days for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h) have been presented. Classification models and their S pars E nsemble (SE) versions have been compared alongside RF, LSTM and daily retrained-LDA models in the above plots.	84
5.9	<i>Perturbed</i> decoder results across days for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h) while dropping the most informative electrode on the test day are presented. Fixed classification models and their S pars E nsemble (SE) versions have been compared alongside RF, LSTM and daily retrained-LDA models in the above plots.	85
5.10	Comparing Sparse Ensemble ELM (<i>SE_ELM</i>), Sparse Ensemble MLP (<i>SE_MLP</i>) against Ensemble ELM (<i>Ens_ELM</i>), Ensemble MLP (<i>Ens_MLP</i>) with summation as a method of combining outputs in unperturbed NHP E dataset. This figure highlights that introducing sparsity in the inputs to base learners lends robustness to ensemble based learning.	88
5.11	Average decoded results across test days for NHPs E and F in (a) and (b) respectively comparing summation and voting methods to arrive at the final ensemble output across different methods.	89
6.1	(a) Block level representation of Reinforcement Learning in context of iBMIs. The iBMI controller emits a discrete output action based on inputs - Neural states and rewards. For an Agent (iBMI Decoder) with N input electrodes and C target labels, (b) depicts a single layer neural network that is trained by Banditron algorithm, (c) represents a multi-layer neural network architecture that is employed by RL algorithms - AGREL, HRL and Deep Q-learning. Do note that the multi-layer architecture can also be extended to Banditron with fixed weights from first to the penultimate layer.	92

6.2	Shows (a) Batch and (b) Online Reinforcement Learning based learning schemes. Batch-trained models learn a model on an explicit set of training data. Typically, supervised learning schemes employed in iBMIs employ batch-based learning schemes. Online reinforcement learning-based models [219] do not employ training data for learning. Instead, they learn during the test phase by virtue of a scalar feedback ($\tilde{f}_i$) at every time-step, $t = i$	94
6.3	(a) This cartoon represents the general nature of the behavioural task performed in experiments 5, 6 and 7. In experiments 5 and 6, the nonhuman primate (NHP) is trained to maneuver a joystick-controlled wheelchair whose position updates every 100 ms. The wheelchair starts from a central position in every trial to reach the target over several time-steps, appearing in one of the three square-shaped locations [123]. However, in experiment 7, closed loop brain control was employed and decoded commands from a discrete classifier were used for controlling wheelchair motion. Wheelchair in both scenarios of the brain and joystick control was driven at a frequency of 10 Hz. (b) represents a center-out task corresponding to experiment 8, wherein the NHP was trained to move a joystick-controlled cursor starting from a central position on a computer screen to one of the eight different square-shaped target locations. Cursor position was updated every 100 ms and the NHP reached the target over several time-steps.	95
6.4	The first session's data of every recorded day was used for training <i>retrained</i> supervised models across all pre-recorded datasets. First session's data of the first day of recording was used to build <i>fixed</i> supervised models as well as tune hyper-parameters associated with Reinforcement Learning (RL) models. Sessions 2 and 3 of each day are used as test data across all techniques.	98
6.5	Decoding accuracy across four experiments has been reported for fixed and daily retrained supervised models - LDA, SVM alongside RL algorithms - AGREL, HRL, Q-learning, Banditron and Banditron-RP. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms. Banditron and Banditron-RP significantly outperform the state-of-the-art RL algorithms and <i>fixed</i> supervised models. . . .	106
6.6	Decoding accuracy across four experiments has been reported for RL algorithms - AGREL, HRL, Banditron, Banditron-RP and Q-learning for 10% error in feedback in (a), (b), (c), (d) and 20% error in feedback in (e), (f), (g), (h) respectively. Banditron and Banditron-RP are susceptible to erroneous feedback and cease to outperform state-of-the-art RL algorithms in all but experiment 6. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms.	108

-
- 6.7 Decoding accuracy across four experiments has been reported for RL algorithms - AGREL, HRL, Banditron, Banditron-RP and Q-learning withholding feedback signal across time-steps thereby introducing sparsity in feedback. (a), (b), (c), (d) depict results for 50% sparsity in feedback and (e), (f), (g), (h) depict results for 75% sparsity in feedback respectively. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms. Banditron and Banditron-RP significantly outperform the state-of-the-art RL algorithms. 109

List of Tables

1.1	Summary of neural acquisition methods	5
1.2	Summary of BMIs used for communication across different signal acquisition methods	7
3.1	Computational Complexity Comparison for a 2-D iBMI decoder . .	51
3.2	Adding signal processing in the implant	54
4.1	Summary of i^2BMI works	57
5.1	Hyperparameter tuning for different models	82
5.2	Training time and approximate number of parameters for unoptimized code of single classifier fixed models for NHP E	82
5.3	Improvement in decoding accuracy (%) for unperturbed (a) NHP E and (b) NHP F datasets afforded by Sparse Ensemble Models . . .	83
5.4	Improvement in decoding accuracy (%) for perturbed (a) NHP E and (b) NHP F datasets afforded by Sparse Ensemble Models . . .	86
6.1	Summary of training paradigms and algorithms	105
6.2	Hyperparameter tuning - parameter choices	105
6.3	Computational Complexity across different models	111

Acronyms

ADC	Analog to Digital Converter
AGREL	Attention-Gated Reinforcement Learning
ALS	Amyotrophic lateral sclerosis
AT	Absolute Threshold
BL	Base Learner
BMI	Brain Machine Interface
BRAIN	Brain Research through Advancing Innovative Neurotechnologies
CNN	Convolutional Neural Network
DBS	Deep Brain Stimulation
ECoG	Electrocorticography
EEG	Electroencephalogram
ELM	Extreme Learning Machine
Ens	Ensemble
FDA	Food and Drug Administration
fMRI	Field Magnetic Resonance Imaging
FVAF	Fraction of Variance Accounted For
GPU	Graphics Processing Unit
HRL	Hebbian Reinforcement Learning
i2BMI	Intelligent intracortical Brain Machine Interface
iBMI	intracortical Brain Machine Interface
IIR	Infinite Impulse Response
ITR	Information Transfer Rate
KF	Kalman Filter
LDA	Linear Discriminant Analysis
LFP	Local Field Potential
LNA	Low Noise Amplifier
LR	Linear Regression

LSTM	Long-short Term Memory
M1	Primary Motor
MAC	Multiply-and-Accumulate
MCU	Microcontroller Unit
MEA	Micro-electrode Array
MLCP	Machine Learning Co-processor
MLP	Multilayer perceptron
MSE	Mean Squared Error
NEF	Neural Engineering Framework
NEO	Nonlinear Energy Operator
NHP	Non-human Primate
PDA	Principal component analysis-based Domain Adaptation
PEU	Portable External Unit
PMd	Pre-motor Dorsal
PPC	Posterior Parietal Cortex
RF	Random Forest
RL	Reinforcement Learning
RNN	Recurrent Neural Network
S1	Primary Somatosensory
SCI	Spinal Cord Injury
SE	Sparse Ensemble
SELMA	Spike-input Extreme Learning Machine
SNR	Signal to Noise Ratio
SSKF	Steady-state Kalman Filter
SVM	Support Vector Machine
TCs	Threshold Crossings
UART	Universal Asynchronous Receiver Transmitter
WHO	World Health Organization

Chapter 1

Introduction

1.1 Motivation and Background

“As long as our brain is a mystery, the universe, the reflection of the structure of the brain, will also be.”, remarked the renowned Nobel winning neuroscientist Santiago Raman Cajal. It is in this spirit of inquiry that the Obama administration launched Brain Research through Advancing Innovative Neurotechnologies (BRAIN) initiative on April 2, 2013 [1]. Across the Atlantic, the European Union commissioned the Human Brain Project along similar lines on October 1, 2013 [2]. Soon Asian behemoths Japan, China followed suit in 2014 and 2016 respectively.

Beyond philosophical musings, it is also of much practical interest to investigate the structure and functioning of the brain as hundreds of millions of people worldwide are currently suffering from brain related disorders. According to the World Health Organization (WHO), roughly 50 million people suffer from dementia, 50 million people suffer from epilepsy while 10 million people are plagued by Parkinson’s disease worldwide [3]. The estimated total cost resulting from these neural disorders amounts to approximately US\$800 billion in the United States alone. Figure 1.1 captures the different areas of the brain along with its associated pathologies. Thus, the underlying theme behind these multi-million dollar brain projects is to make a concerted effort to grasp the fundamental nature of its structure and functioning following the successful model of the Human Genome Project [4]. The enhanced understanding of the brain is expected to help researchers in treating, curing and even preventing brain disorders.

FIGURE 1.1: Shows number of people worldwide suffering from neural disorders. It also shows a rough mapping between disorders and different regions of the brain

1.2 Neurotechnology

The rush to study the brain has given rise to a spate of unprecedented technological developments, broadly termed as *Neurotechnology*. This emerging field has shown effective therapeutic benefits wherein conventional pharmaceutical drug-based treatments have failed or have been deemed too risky. For example, Deep Brain Stimulation (DBS) neurotechnology is employed as a surgical procedure to treat people from Parkinson's disease, when medications are unable to control motor symptoms [5]. DBS essentially involves implanting electrodes in targeted areas of the brain to deliver electrical impulses in order to change or block the abnormal neural activity that causes motor symptoms. Apart from Parkinson's, DBS has shown promising results in overcoming epilepsy, obsessive compulsive disorder, clinical depression, essential tremor among other things. It has been approved by the Food and Drug Administration (FDA) since 1997 and over 160,000 patients have benefited from it worldwide [6]. The market for DBS is slated to reach up to USD 1,676.8 million by the year 2026, exhibiting a compounded annual growth rate of 11.6 % [7]. Major players in this space include Medtronic, Abbott (ST. Jude Medical), Boston Scientific Corporation, Beijing PINS Medical Co. Ltd.

FIGURE 1.2: Shows examples of neural interfaces and their implantation locations [8] (CC BY 3.0 license)

Cochlear implant is another example of a fascinating neurotechnology which was FDA approved in the year 1985 for adults and in the year 1990 for children. It is a surgically implanted electronic device that restores partial hearing in patients who suffer from profound hearing loss resulting from inner-ear damage. The severity of inner-ear damage in these patients renders conventional sound amplification based hearing aids to be ineffective. Cochlear implants take a radically different approach to overcome hearing loss by electrically stimulating the auditory nerve fibres which interface with the brain. Thus, cochlear implants essentially try to reproduce the functioning of the damaged parts of the inner-ear. More than 400,000 people have been beneficiaries of a cochlear implant as reported in 2017 [9]. It is estimated that the global cochlear implant market size will reach up to USD 3.3 billion in the year 2027 [10]. Major companies dealing in cochlear implants include Medel, Cochlear Corporation and Advanced Bionics. Figure 1.2 captures some examples of neurotechnologies in use today.

1.3 Brain Machine Interface

Taking inspiration from the aforementioned neurotechnologies, a novel and more technically involved neurotechnology - the Brain Machine Interface (BMI) is beginning to emerge. To put it simply, a brain machine interface aims to translate

FIGURE 1.3: Shows constituent blocks of a brain machine interface (BMI). The system takes neural activity as an input, which is then processed to obtain suitable features to drive effector such as an artificial limb, a computer cursor, a wheelchair etc. Typically users can visually see an effector’s movement and this serves as a feedback mechanism. Other modes of feedback such as neural stimulation are also being explored to improve the overall efficiency of BMI systems.

thought into action. The input to these systems is neural signals which are subsequently processed to control external assistive devices/effectors. The effectors are meant to substitute/replace body parts and functions in BMI users. The initial target audience for this form of assistive technology is patients suffering from paralysis. It is estimated that as many as 1 in 50 people are suffering from paralysis in the US [11]. Thus, addressing the needs of this population is of immediate concern and forms the broad objective of this thesis. Besides paralysed patients, efforts are also ongoing to use non-invasive BMIs in healthy population for applications ranging from video-games to aids for improving working memory [12]. Silicon valley has been enamored by BMIs and tech moguls such as Elon Musk (Neuralink), Mark Zuckerberg (Facebook) are in the race to bring BMIs to the general public [13].

Figure 1.3 captures the constituent blocks of a typical brain machine interface. Neural activity emanating from the brain is subjected to the steps of signal acquisition, feature extraction and decoding to drive end-effectors. Some examples of effectors include a wheelchair used for locomotion, a computer cursor used for communication, a prosthetic hand used for feeding etc.

TABLE 1.1: Summary of neural acquisition methods

Methods	Neural Activity	Invasiveness	Temporal Resolution (s)	Spatial Resolution (mm)
fMRI	Metabolic	Non-invasive	~ 1	~ 1
NIRS	Metabolic	Non-invasive	~ 1	~ 5
EEG	Electrical	Non-invasive	~ 0.05	~ 10
ECoG	Electrical	Invasive	~ 0.003	~ 1
Intra- cortical	Electrical	Invasive	~ 0.003	~ 0.05

FIGURE 1.4: Shows different techniques for acquiring signal from the brain. Different colours indicate approximate limits of depth and physical scale of activity captured in the respective techniques. Figure adapted from [14]

1.3.1 Types of BMI

Capturing cerebral activity is a monumental technical task and scientists have been able to achieve this owing to the rapid advancements in electronics and instrumentation technology. This development has given birth to neural data acquisition technologies such as Functional Magnetic Resonance Imaging (fMRI), Electrocor-ticography (ECoG), intracortical microelectrode arrays, single cell recordings, Elec-troencephalography (EEG) among others. These methods can be categorized on the basis of their invasiveness as invasive or non-invasive. Table 1.1 summarizes the prevalent neural signal acquisition methods. Another way to classify these methods is on the basis of neural activity acquisition as metabolic and electrical.

To further elucidate, Figure 1.4 gives a pictorial representation of different signal acquisition methods. Different colours in the figure capture approximate limits of depth and the physical scale of activity recorded by the techniques. As we traverse the figure from left to right, the temporal resolution associated with recording modalities decreases. Two observations can be made at this point of time. Firstly, the electrical methods of neural activity acquisition clearly trump metabolic methods in terms of temporal resolution. Secondly, metabolic methods offer a better spatial resolution compared to electrical methods.

EEG and fNIRS are among the more popular non-invasive neural signal acquisition techniques [15]. These techniques offer the prospect of portability and low-cost. EEG records the integrated activity of cerebral cortex's pyramidal neurons. It measures the post-synaptic potentials linked to neural activations (e.g., event related potentials) or the magnitude of various oscillation frequencies (i.e., alpha, beta, delta, theta, mu and gamma) [16]. fNIRS on the other hand, relies on the principle of neurovascular coupling. It measures the changes in regional cerebral blood flow, oxygenated hemoglobin (Hb_{oxy}), and deoxygenated hemoglobin (Hb_{deoxy}) resulting from neuronal activation [17]. The measuring principle of fNIRS is similar to the measurement of blood-oxygenation-level-dependent responses in fMRI albeit with a relatively lower spatial resolution [18]. However, compared to fMRI, fNIRS yields a more affordable and easily implementable portable system.

1.3.2 Why intracortical BMI?

Non-invasive BMIs have shown a number of successful demonstrations. For example, using EEGs scientists have shown successful implementations of control of computer cursors [19–21], wheelchairs [22–28], robotic devices [29, 30], exoskeletons [31–33] among other things. Along similar lines, neural haemodynamic change capturing technologies such as fMRI and fNIRs have shown proof of concept demonstrations in establishing communication [34–37] and control [38, 39].

Table 1.2 shows a summary of reported BMI works to establish communication among healthy and motor-impaired patients across different signal acquisition methods. Clearly, invasive MEA based signal acquisition methods offer the best information transfer rates (ITR) among reported works by at least an order of magnitude.

TABLE 1.2: Summary of BMIs used for communication across different signal acquisition methods

Works	Year	Recording Modality	Impairment Cause	Average ITR (bits/sec)
[40]	2012	fMRI	None (Healthy)	0.083
[41]	2015	EEG	ALS	0.19
[42]	2016	ECoG	ALS	0.21
[43]	2018	EEG	None (Healthy)	0.338
[44]	2017	MEA	SCI	4.2

ITR stands for information transfer rate

The above reported works are compared among different subjects with different decoding techniques and experimental paradigms. Thus, ITR reported for comparison should be taken with a pinch of salt. However, despite experimental differences, it can be observed that electrical based methods outperform metabolic ones. Thus, reported works follow the temporal resolution column reported in Table 1.1, wherein faster BMIs correspond to technologies with finer temporal resolution values. Average information exchange rate in humans is 39.5 bits/second. Thus, based on the works so far, it is conclusive that MEA based technology has the potential to get us close to this figure compared to other modalities of signal acquisition.

The reason for MEA's superior performance is its close proximity to the generative source of neural activity, i.e. single neurons. Thus, MEA yields a very fine resolution signal capable of teasing out more task-relevant information and hence better quality of control. However, these techniques come with a safety concern. They require a neurosurgical procedure involving opening the scalp and penetration of the brain tissue. These procedures introduce infection, damage risks to the exposed tissue [45, 46]. Do note that these risks are associated with any form of surgery. On a brighter note, FDA approved implantable neurotechnologies - cochlear implants and DBS have demonstrated reasonable safety and are commonly used, thereby setting a good precedent for intracortical BMIs to follow.

1.3.3 Problems with state-of-the-art iBMIs

Despite impressive technological developments and proof-of-concept demonstrations, iBMIs have failed to transition from clinical trials to everyday use. As with

any other emerging technology, iBMIs face several unsolved problems. To illustrate the existing problems, we consider an example of a demonstration from one of the world's leading iBMI consortium - BrainGate. BrainGate is a joint venture involving Brown University, Case Western University, Massachusetts General Hospital and Stanford University. Figure 1.5 shows labeled physical constituents of an iBMI employed in clinical trials.

FIGURE 1.5: Illustration of an intracortical Brain Machine Interface (iBMI) [47]

The following are some of the most vexing challenges associated with state-of-the-art iBMIs:

- **Wires**

Wires are predominantly used to read out neural data emanating from MEAs in iBMIs [44, 46, 48, 49]. These wires leave an opening in the skull, thereby making it infection-prone. Furthermore, these percutaneous wires disturb the blood-brain barrier thereby invoking the brain's inflammatory response to foreign objects [50].

- **Portability**

Current data acquisition and processing is implemented on tethered bulky PCs. This increases the form factor of the overall system hindering portability. Portability is particularly important for iBMIs which deal with locomotion such as wheelchairs.

- **Bandwidth**

Solutions to the above two problems of eliminating wires and increasing portability, is to build an implantable wireless iBMI system [51]. However, current commercial neural data acquisition systems have data rates in the orders of tens of Mbits/sec. For example, a popular neural data acquisition vendor - Ripple exhibits data rates of approximately 48 Mbits/second [52] for a 100 channel input array. Furthermore, with the exponential rise in the number of simultaneously recorded neurons [53], the data rate is expected to grow exponentially. This increased data rate and subsequent communication bandwidth, makes the design of lossless transmission circuitry difficult, and consequently increases the power consumption of the transmitter drastically.

- **Power dissipation, heat and battery lifetime**

Proof of concept in-vivo implantable wireless iBMI systems supporting 4 to 100 channels have been reported [54–57]. A key challenge reported in these designs is to limit the power dissipation of the implant to $80\text{ mW}/\text{cm}^2$. This power dissipation limit is to prevent tissue damage from overheating [58–60]. Furthermore, apart from overheating another incentive to reduce power dissipation is to increase the battery lifetime. In case of [54], the authors have reported a battery lifetime of 7 hours, while using a rechargeable Lithium ion battery. With increase in data rates resulting from increase in input neurons, the problem of frequent recharging will only worsen for current architectures. The current iBMI systems are thus, far away in terms of battery lifetime from implantable pacemakers which typically last for up to 6 years. Furthermore, the physical process of recharging dissipates substantial heat to effect tissue damage and is another concern to be tackled.

- **Biocompatibility and longevity of interfaced electronics**

The iBMI system's biocompatibility and longevity are one of the most important factors in determining its clinical benefit. Studies have shown that the brain tissue reacts to the presence of chronic implanted MEAs [61–68]. This response is in terms of inflammatory reactions, glial scarring, fibrotic tissue encapsulation and neuronal migration and death. The aforementioned phenomena lead to degradation of the recorded signal quality over time. Furthermore, tissue reactions and integration before implantation issues have been implicated to be the leading causes resulting in failure of MEAs [69].

Despite all the above issues, a total of 9,254 published human implant days exist for Utah array. To date, [70] is the longest period of implantation with a total of 1,925 days. Thus far, a total of 48 human subjects have been enrolled in clinical trials across the world. Figure 1.6 gives a summary of reported human iBMIs [71].

FIGURE 1.6: Duration of chronic human iBMIs reported across clinical studies [71]

Solving the problem of biocompatibility is an active area of research. Novel biocompatible materials such as carbon-based nanotubes and polymers like poly(3,4-ethylene dioxythiophene) and poly(styrene sulfonate) are being considered as alternatives to metal electrodes [72]. Apart from interfaced MEAs, materials used for packaging the implant are also being studied for their biocompatibility [73–75].

• Nonstationarity

The current state-of-the-art systems require daily calibration procedures to set up the system for use. The reason for daily calibration is the changing nature of the input signal statistics, often referred to as non-stationarity. This results in frustration for the end-user. Approaches such as unsupervised

adaptation [76], transfer learning [77] and deep neural networks [49, 78] have been studied to overcome this problem.

- **Presence of Neuroengineer**

FIGURE 1.7: Panel of action potentials recorded at input electrodes

The main objective of building an iBMI system is to lend autonomy to patients afflicted by debilitating motor impairments. However, the current state-of-the-art systems require complex procedures to set up the system for use. For example, Figure 1.7 shows a grid plot of signals in the form of action potentials captured by electrodes in the MEA. The neuro-engineers look at such spike grids and manually select channels of interest, and/or resort to spike sorting to extract input features [79]. Efforts have been undertaken to solve these problems algorithmically without human supervision [49, 80–82]. Some initial clinical trials have been reported recently with the intention of making iBMI more user-friendly in home-based settings [82, 83].

- **Cosmesis**

Cosmesis is defined as the preservation and restoration of physical beauty. It is critical that the designed iBMI system addresses the cosmesis concerns among traumatized iBMI end users. A poorly designed system might perhaps end up stigmatizing the patient further and discourage adoption among the community. A systematic survey based assessment based on feedback from

potential end-users was presented in [84]. Figure 1.8 shows different BMI options presented to the respondents who participated in the survey [84]. The findings corroborated that the form factor of a BMI system is critical in driving adoption. Furthermore, the respondents were twice as likely to adopt surgically implanted wireless technologies as opposed to their wired counterparts.

FIGURE 1.8: Shows illustrations of different brain machine interface system options presented to the respondents in the survey [84]

1.4 Objectives

Based on the current state of iBMI systems reviewed so far, we observe that significant open areas of research exist. To sum up the problems from an electrical engineering standpoint, building a low power fully implantable iBMI system solves most of the problems listed in Section 1.3.3. Accordingly, following objectives were set for this thesis:

- Explore system level options for building fully implantable wireless iBMIs
- Verify decoding capability of first generation neuromorphic decoder chips against state-of-the-art algorithms
- Develop solutions to counter drift/non-stationarity in iBMI apparatus

- Explore strategies to enable co-adaptation between machine learner and the subject's brain

1.5 Outline

FIGURE 1.9: Shows mapping between objectives and chapters in this thesis

Chapter 2 presents different components of an intracortical BMI system. We walk through the various steps involved from capturing the neural signal to subsequent processing steps in order to drive an output effector.

Chapter 3 presents a system level exploration of building viable fully implantable wireless iBMI systems. Thereafter, it is argued that adding intelligence to the implant is a plausible option. Offline results associated with an intelligent iBMI decoder is presented.

Chapter 4 presents online results of an intelligent iBMI decoder presented in the previous chapter.

Chapter 5 presents the non-stationarity problem faced by iBMIs. A novel solution in the form of building sparse ensemble machine learners is presented to mitigate the impact of non-stationarity.

Chapter 6 introduces a co-adaptive form of iBMI based on the reinforcement paradigm of learning. A novel approach to facilitate online learning is presented and compared against state-of-the-art approaches.

Finally, we conclude in *Chapter 7* with future directions that this piece of work can lead to. Furthermore, we also present exciting avenues of work that can be undertaken in this dynamic field.

Chapter 2

Intracortical Brain Machine Interface

Intracortical Brain Machine Interfaces (iBMIs) have made it possible for patients suffering from paralysis to establish communication [44], locomotion [85] and even perform complex tasks such as feeding oneself [46, 86] solely through neural control. It is indeed heartwarming to see these chronic patients take a step towards leading normal lives as seen in demo videos presented in [46] and [86]. iBMI systems can be typically broken down into the following blocks – a) Signal acquisition and pre-processing, b) Feature extraction, c) Decoding and d) Effector (see Figure 2.1). We will focus on outlining these blocks in this chapter.

FIGURE 2.1: Typical block diagram of an iBMI system consisting of signal acquisition, feature extraction, decoder and effector blocks. Signal acquisition often takes place in the implant while the other blocks are implemented externally.

2.1 Action Potential - The Input Neural Signal

The neuron is considered to be the fundamental processing unit in the brain. Neurons are electrically excitable cells which transmit and process information via electrochemical signalling. They are responsible for sending motor commands to our muscles, for receiving sensory input and for conveying signals at all steps in between. Human brains have approximately a total of 86 billion neurons.

FIGURE 2.2: Simplified view of structure of neuron, adapted from [87].

A neuron is made up of three parts, a cell body known as soma, an axon and dendrites as seen in Figure 2.2. To draw an analogy with a tree, soma, axon and dendrites serve as trunk, roots and branches respectively. The axon is the output transmitting part of the neuron. Dendrites serve as receiving parts. A neuron receives information from other cells through its dendrites. Dendrites branch near the tips and have leaf-like structures known as spines. Soma houses the nucleus and DNA. It is in the soma that proteins are formed to be transported via axon and dendrites.

Neurons are basically electrical devices as seen in Figure 2.3. There exist many channels sitting in the cell membrane, which allow negative or positive ions to flow in/out of the cell. Typically, the inside part of the cell is more negative than the outside part. This potential difference between the inside part and outside part is known as the cell's membrane potential. The value of resting membrane potential is around -70 mV .

Membrane potential is not static. It is constantly increasing or decreasing depending on the inputs coming from axons of other neurons. Depending on the nature

FIGURE 2.3: Action potential leads to neurotransmitters being released across synaptic cleft thereby resulting in an electrical signal in the postsynaptic neuron [88].

FIGURE 2.4: Action potential results when the sum total of inhibition and excitation makes the membrane threshold surpass the threshold [88].

of inputs, the membrane potential becomes more positive or negative. Thus, these inputs are termed as excitatory and inhibitory for causing an increase and decrease in membrane potential respectively. When the combined total of excitatory and inhibitory inputs makes the membrane potential reach $\approx$ a threshold value of 50 mV , an action potential as shown in Figure 2.4 occurs. Action potential is the fundamental unit of communication between the neurons. Action potentials are also referred to as spikes in the neuroscience parlance. Action potentials last for up to 1 ms .

2.2 Recording Sensor - Microelectrode Array

The only FDA approved invasive form of neural signal acquisition is the Utah microelectrode array (MEA). It has been the workhorse of chronic iBMI experiments both in non-human primates and humans. Other non-FDA approved variants of microelectrode arrays include Michigan probes [89] and tetrodes [90].

FIGURE 2.5: (a) Shows a 10×10 grid of Utah array along with insulated gold wires that percutaneously connect to an external connector, (b) and (c) show side-view and backside of the Utah array, adapted from [59].

The Utah array was first reported in [91] more than two decades ago in 1996. It is built on a single silicon substrate of thickness 0.2 mm and side length of 4.2 mm . A 10 by 10 grid (Figure 2.5) of electrically isolated needles project out from this substrate. The length of each needle is 1.2 mm and the needles are etched chemically to produce a sharp tip. Each needle constitutes an independent electrode. All regions of electrodes except the tip are coated with polyimide to provide insulation. The tip of each electrode is metallized with platinum. The platinum coating essentially serves as an electrical interface between the electrode and the surrounding tissues. The length of the exposed surface is $\approx 50 \mu\text{m}$. Typical electrode impedance lies in the range $80 - 150 \text{ k}\Omega$. A hundred gold pads are deposited on the backside of the array. These pads have a hundred 1.25 mil insulated gold wires bonded to them, one wire for each pad. The other end of these wires connect to a percutaneous connector to establish a connection with external electronics.

Craniotomy is performed and the Utah array is pneumatically inserted in a single layer of up to 1.5 mm in cerebral cortex (Figure 2.6). The microelectrode array picks up changes in voltage in the extracellular environment of nearby neurons,

FIGURE 2.6: Shows two Utah arrays implanted in the motor and posterior cortex of a non-human primate. The implantation was done to be employed in a brain machine interface study, adapted from [92].

and translates it into a proportional current. This recorded current is passed downstream for further processing. Utah array has been implanted in cortical areas associated with movement such as primary motor cortex (M1), premotor dorsal cortex (PMd), posterior parietal cortex (PPC) to study brain machine interfaces. Furthermore, the Utah array has been implanted in the visual cortex to study ways to overcome visual impairments.

Besides neural recording, the Utah array is also used for stimulating neurons to provide a form of feedback to the brain. During stimulation, the electrodes translate input electronic currents into ionic currents through the extracellular media. Figure 2.7 shows a representative example of electrical activity captured by a Utah array electrode.

FIGURE 2.7: Example of neural signal captured by the Utah array, adapted from [93].

Despite significant advancements, Utah arrays cause issues in the form of tissue reactions and inflammations. Besides issues of biocompatibility, Utah arrays can

only be used to record from flat cortical surfaces, not the sulci. Furthermore, the recording depths are limited to 1.5 mm. To overcome these limitations, novel approaches such as shank based neuropixels [94] probe, and even more radical approaches such as distributed motes-based recording structures named as neural dust [95] are being pursued. These approaches are in their infancy and are yet to demonstrate real-time results similar to the Utah array.

2.3 Signal acquisition front-end

Raw signal captured from the microelectrode array is processed through front-end electronics composed of low noise amplifiers (LNAs), variable gain amplifiers, filters, analog to digital converters (ADCs). Broadly speaking, the front-ends can be classified in one of the three categories shown in Figure 2.8.

Figure 2.8(a) employs a single ADC being shared between all input channels. Each channel has its own LNA. An analog multiplexer is used to pass each channel's LNA's output to the ADC. This form of multiplexing is popularly known as time-division multiplexing. [96] reports a neural front end employing this architecture for 64 input electrodes. However, as the number of channels increase, the sampling frequency of ADC and multiplexer increase, leading to a concomitant rise in power consumption. Thus, this architecture is infeasible for the trend of increase in input electrodes.

An alternative architecture with one ADC in each channel is shown in Figure 2.8(b). Since, the bandwidth of neural signals is relatively low (≈ 300 Hz to 6000 Hz), the sampling frequency and power consumption of individual ADCs is low. Another difference in this architecture is the use of a digital multiplexer in lieu of an analog one in the previous architecture. Employing a digital multiplexer enables overcoming the issue of cross-talk and is relatively more robust to noise than an analog multiplexer. [97] is an example of this architecture for 16 input channels. Having an ADC in each channel is not plausible for the current channel count of around $\approx 10,000$ as it leads to unsustainable amount of area and power consumption.

A compromise between first and second architectures is to have one ADC shared between multiple but not all input channels as shown in Figure 2.8(c). There are

m rows and n columns in this architecture. An ADC is assigned to each of the m rows. This architecture gels well for an increased number of input channels for the correct choice of m and n . [98] is an example of this architecture for a 100-channel input front end.

FIGURE 2.8: Shows popular architectures reported to realize neural front-ends. (a) This architecture shares one ADC across all input channels. (b) In this architecture, each input channel has its own ADC. (c) This architecture is a compromise between (a) and (b) where one ADC is shared between m channels, where m is a subset of total input channels.

Building implantable neural signal acquisition front-ends poses extreme space and power design constraints. The following list captures design characteristics of a neural signal acquisition front-end:

- **Low power dissipation** – these systems are required to be low power to reduce load on batteries, wireless power coupling or any other form of power source.
- **Input referred noise** – Since the input signal is of the order of microvolts, low input referred noise systems ought to be designed.
- **DC offset rejection** – Offset voltage appears at the recording electrodes and needs to be rejected.
- **High Common Mode Rejection Ratio and Power Supply Rejection Ratio** – 60 Hz ambient noise, power supply noise and stimulation artifacts need to be effectively rejected from the recorded input signal.

2.4 Feature Extraction

FIGURE 2.9: Shows two popular methods to extract features from extracellular recordings - Local Field Potential (LFP) and Spikes. Figure is adapted from [99] (CC BY 4.0 license).

The next step after conditioning the raw neural signal is to extract meaningful features from it (see Figure 2.9). In machine learning parlance, this step is known

as feature extraction. Two popular techniques are frequently employed for feature extraction in iBMI systems - a) Spike detection and b) Local Field Potential (LFP).

2.4.1 Spike Detection

Detecting a spike begins with *filtering* as the first step. Bandpass filtering with cutoff frequencies $\approx 300 - 6000 \text{ Hz}$ is done to isolate the underlying spectrum of spikes from various sources of noise. Noise sources typically include power line noise, ambient recording environment noise, background noise resulting from summation of activity from distant neurons, front-end recording circuitry noise among other sources. This filtering step is typically implemented externally on a PC in Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States). Causal infinite impulse response (IIR) digital filters are popularly employed.

FIGURE 2.10: Shows the signal processing chain involved in detecting spikes. The analog front-end's raw neural signal is subjected to bandpass filtering, followed by an optional pre-emphasis step. Thereafter, thresholding is done to obtain unsorted spikes also known as Threshold crossings.

Post filtering, one can visually see spikes riding atop a noisy signal as seen in Figure 2.10. The filtered signal is then subjected to a *thresholding step*, wherein spikes are isolated from background noise. Choosing an appropriate threshold is a compromise between - a) missed spikes if the threshold is set too high, and b) false spikes as noise is classified as spikes if the threshold is set too low. Different algorithms have been reported to aid in performing effective spike detection. Popular among these include - absolute threshold, nonlinear energy operator (NEO) and wavelet based transform. To effectively isolate spikes, algorithms such as nonlinear energy operator and wavelet transform rely on a pre-emphasis step to accentuate spikes. Absolute value method does not include any pre-emphasis step and hence in Figure 2.10 pre-emphasis has been mentioned to be optional.

Authors in [100] have provided a comprehensive comparison of absolute threshold, nonlinear energy operator and wavelet transform based spike detection techniques. They have concluded that NEO and absolute threshold outperform wavelet based methods by a considerable margin. Accordingly, in this thesis, we have used absolute threshold and NEO based spike detection methods. Furthermore, they have shown that at higher signal-to-noise ratios, NEO and absolute value methods yield a roughly equivalent performance whereas at lower signal-to-noise ratios NEO outperforms absolute value method. Apart from these algorithms, some practitioners prefer to manually set the threshold to an arbitrary value based on visual inspection.

2.4.1.1 Absolute value method

This method computes the threshold as a multiple of median absolute deviation of the filtered neural signal. Computation based on the median mitigates the impact of spike firing rate on the calculated threshold. Previously reported threshold computation methods were based on estimation of standard deviation. However, the standard deviation methods are prone to higher error rates as the computed threshold varies as per the input firing rate. Ideally, the threshold should only be affected by the background noise and not the spike firing rate. Median absolute deviation tries to achieve this with the threshold computed as [80],

$$\begin{aligned}\sigma &= \text{median}(|X|)/0.6745 \\ Thr &= k \times \sigma\end{aligned}\tag{2.1}$$

where X is the filtered raw signal, and k is an integer typically between 2 to 5. Negative threshold crossings of the filtered raw signal are considered as spikes.

2.4.1.2 Nonlinear Energy Operator (NEO)

This method involves application of a pre-emphasis step before spike detection [101]. The idea behind this approach is to accentuate high frequency and high energy content, i.e. spikes, amidst background noise, thereby improving spike

detection performance. [100] has done a comprehensive analysis to show improvements afforded by NEO over absolute value method. NEO threshold calculation involves the following steps [101],

$$\begin{aligned}\psi[n] &= x^2[n] - x[n-1] \times x[n+1] \\ Thr &= l \times \frac{\sum_{n=1}^N \psi[n]}{N}\end{aligned}\tag{2.2}$$

where l is an integer swept across a range of values to realize an optimal performance and the optimal value varies according to the experiment. Threshold crossings of $\psi[n]$ over Thr are considered as spikes.

2.4.1.3 Wavelet Analysis

This method employs discrete wavelet transform as a means of pre-emphasis to enhance spike detection. Pre-emphasis based spike detection methods particularly work well in low signal to noise ratio (SNR) settings. [102–104] have reported improvement in performance using wavelet transforms over state-of-the-art methods. However, this superior performance comes at the cost of at least two orders of magnitude increase in computational complexity [100].

Application of this method involves the following steps [103]. Firstly wavelet transform is computed over 5 dyadic scales ($W(2^j, n)$, $j = 1, \dots, 5$). Thereafter, maximum of absolute values sum is obtained as,

$$j_{\max} = \operatorname{argmax}_{j \in \{3,4,5\}} \left(\sum_{n=1}^N |W(2^j, n)| \right)\tag{2.3}$$

Later pointwise product $P(n)$ is computed as,

$$P(n) = \prod_{j=j_{\max}-2}^{j_{\max}} |W(2^j, n)|\tag{2.4}$$

Finally, threshold is computed as,

$$Thr = m \frac{1}{N} \sum_{n=1}^N w(n) * P(n) \quad (2.5)$$

where $w(n)$ represents a Bartlett window and m is an integer multiplier. A Bartlett window based convolution is used to lessen the effects of noise, spurious peaks and signal peak mismatches over different scales [103]. Equations 2.3-2.5 are adapted from [103].

2.4.2 Spike Sorting

Electrodes typically pick up spiking activity of neighbouring 1-4 neurons. Accordingly, detected spikes can be of two kinds - Single unit and multi-unit. Multi-unit spikes are a result of overlapped activity of more than one neighbouring neuron. Single unit spikes on the other hand originate from individual neurons. Accordingly, there can be more than one single unit that can appear at a given electrode. The process of ascribing single unit spikes to putative individual neurons is known as sorting (see Figure 2.11). Due to the overlapped nature of multi-unit spikes, they cannot be reliably sorted.

FIGURE 2.11: Shows pictorially how spikes emanating from two different neurons are sorted after being captured by the same electrode, adapted from [99].

Sorting can be done manually wherein an experienced practitioner carefully sifts through recordings, defining templates for single unit spikes across all input electrodes. However, this approach takes several hours and results may differ among different practitioners [105]. Automated spike sorting algorithms involve steps of feature extraction followed by clustering. Popular feature extraction methods for spike feature extraction include principal component analysis [106], wavelets [107] among others. Clustering as the name suggests involves grouping points in the feature spaces into clusters, each of which corresponds to an individual neuron. Bayesian clustering [108], superparamagnetic clustering [80], hierarchical adaptive means clustering [109] are a few examples of clustering algorithms.

The impact of spike sorting in iBMIs is studied by various groups often with conflicting results. [110] have reported sorting to provide significantly enhanced results. The proponents of sorting argue that individual neurons respond differently to different actions. Thus, sorting subsequently helps in capturing this information and helps improve the decoding performance. [111, 112] on the other hand have reported similar or improved levels of performance without sorting. Unsorted spikes are simply referred to as *threshold crossings* (TCs). The theory of random projections is invoked as a possible reason for similar levels of performance [111]. The skeptics of sorting argue that, a) TCs yield more stable signals than sorted spikes, b) sorting imposes significant computational constraint on implanted hardware implementation and c) sorting algorithms are somewhat inexact due to unavailability of ground truth, waveform changes due to electrode drift.

2.4.3 Local Field Potential

Local field potential is the sum of excitatory and inhibitory dendritic potentials in a "listening sphere" around the electrode's tip [113]. It reflects the activity of a relatively large number of neurons approximately in the order of tens of thousands [114]. LFP is simply obtained by low pass filtering raw neural signal obtained from the analog front-end with cutoff frequency in the range of around 100 Hz (see Figure 2.12).

Studies have been reported comparing spike-based and LFP-based decoding [115, 116]. Spikes have emerged as the better of the two options so far. The reason for LFP's inferior performance is that larger signal and noise correlations exist

FIGURE 2.12: Shows the signal processing chain involved in computing Local Field Potential (LFP). Low pass filtering with a cut off frequency of around 100 Hz of the analog front end signal is done to arrive at the LFP signal.

among channels. However, LFPs in combination with spikes have been yielding improvements in performance than spikes alone [116]. Despite inferior decoding performance, LFPs have two points working in their favour. Firstly, they are relatively more stable and robust compared to action potentials. LFPs are less affected by micro-motion of electrodes or electrode degradation. In fact, [117] reports capturing meaningful LFPs from channels devoid of spikes. Secondly, LFPs are computationally friendly requiring only a low pass filtering operation with cutoff frequencies as low as 100 Hz. This significantly relaxes the sampling rate requirement of front-end ADCs and makes them hardware amenable.

2.5 Decoder

The iBMI decoding paradigm currently reported in literature involves learning a mapping between neuronal features and a suitable effector output signal to drive a neural prosthetic. Typically in iBMI systems, the decoding algorithms are implemented on a PC. Input features and effector outputs occur in discrete time-steps, t_s . These time-steps typically range from 20 ms to 100 ms. Reducing time-steps has been shown to improve neuroprosthetic control [118]. Inputs are typically binned in back-ward looking time-windows (T_w) of few tens to hundreds of ms.

In case of spike-based iBMIs, number of binned spikes are counted in every input channel at a given time, $t = i$ to form the input feature vector, $\vec{X}_i$. $\vec{X}_i$ represents an estimate of firing rate in spike-based iBMIs. Similarly, in case of LFP-based iBMIs, the average binned value in each channel yields the input feature vector, $\vec{X}_i$ at time, $t = i$. In general, input feature vector is given as,

$$\vec{X}_i = \begin{bmatrix} r_1(i) & r_2(i) & \cdots & r_{D-1}(i) & r_D(i) \end{bmatrix} \quad (2.6)$$

where D represents the number of input channels and $r_z(i)$ represents the input feature appearing at z^{th} electrode in a lookback time window of T_w seconds.

Corresponding to the input time-steps, the output variables are tracked to yield an output vector, $\vec{Y}_i$. Depending on the mode of output, iBMIs can be classified - a) Discrete state, b) Continuous state and c) Hybrid (mix of continuous state and discrete state).

2.5.1 Discrete state

In a discrete state iBMI, the output to the effector is one among many discrete choices. An example of this type of iBMI is key selection on a keyboard to enable communication [119]. The output vector in case of discrete state iBMIs is given as,

$$\vec{Y}_{Di} = \begin{bmatrix} o_1(i) & o_2(i) & \cdots & o_p(i) \end{bmatrix}; \vec{Y}_{Di} \in \mathbb{R}^p \quad (2.7)$$

where p represents the number of available choices. $\text{argmax}(\vec{Y}_{Di})$ yields the final decoded output. Other examples of discrete state iBMIs include choosing among grasp types in a hand-prosthetic [78, 120], finger movement classification [121, 122], wheelchair motion with discrete choices [123] etc.

Popular algorithms for discrete state iBMIs include linear discriminant analysis [123], support vector machines [124], Naive-Bayes classifier [119]. More recently, using large amounts of training data, deep learning based models have been reported. Specifically, a convolutional neural network (CNN) plus long-short term memory (LSTM) based model was reported in a four movement task involving wrist flexion, wrist extension, index flexion and index extension [78] (see Figure 2.13).

FIGURE 2.13: (a) Shows a motor imagery based experiment in which the participant is asked to imagine one of the four movements - index flexion, index extension, wrist flexion and wrist extension. (b) Shows a convolutional neural network (CNN) plus long-short term memory (LSTM) based decoder to infer the intended movement. Figure has been adapted from [78].

2.5.2 Continuous state

The continuous mode iBMI is relatively more popular with the outputs trying to predict a continuous trajectory in terms of variables such as position, velocity and acceleration. Example of this type of iBMI is a paralyzed person trying to feed oneself through a prosthetic guided by decoded velocity outputs [46]. The output vector can be multi-dimensional depending on the degrees of freedom of output control and is given as,

$$\vec{Y}_{Ci} = \begin{bmatrix} o_1(i) & o_2(i) & \cdots & o_p(i) \end{bmatrix}; \vec{Y}_{Ci} \in \mathbb{R}^p \quad (2.8)$$

where p represents the number of dimensions and $o_p(i)$ represents the continuous decoded output value for the p^{th} degree of freedom. Other prominent examples include continuous trajectory control of a cursor on a computer screen [125], robotic/natural arms [86, 126].

FIGURE 2.14: Shows a recurrent neural network used to predict movement of native hand of a non-human primate. Specifically movement parameters include two dimensional velocity ($v_x(t), v_y(t)$) and position ($p_x(t), p_y(t)$). Figure adapted from [49].

Popular algorithms include population vector algorithm [127], Wiener filter [128], Kalman filter [48]. Similar to the discrete state case, deep learning models such as Long-short term memory (LSTM) [129], recurrent neural networks (RNN) [49, 130] have been reported for continuous state decoding. Figure 2.14 shows application of a recurrent neural network to predict natural hand movement in a non-human primate [49].

2.5.3 Hybrid

A hybrid system combines both continuous and discrete modes of decoding. An example of this type of system is the 'point and click' cursor based system reported in [131].

In this system, the intention-driven cursor on screen traverses a continuous trajectory to reach the desired location. Once the desired location is reached, a click-function enables selection of the desired icon. This operation is akin to able-bodied individuals operating a mouse/track-pad. The cursor trajectory employs a continuous decoder whereas the click function employs a discrete decoder. Current state-of-the-art iBMI communication system reports using the hybrid decoding technique to achieve record information transfer rates of up to 4.2 bits/second [44]. Figure 2.15 shows an example of a hybrid iBMI system employed by a subject to interact with a tablet PC.

FIGURE 2.15: Shows a point and click type hybrid iBMI system employed by a subject to reply to an email on a commercial tablet, adapted from [83].

2.6 Effector

A host of effectors such as computer cursors, prosthetic arm, natural arm, wheelchair, keyboard have been employed. The choice of effector informs the design choices to be made in an iBMI system. For example, in a robotic arm based iBMI system [46], a continuous iBMI system is more amenable towards mimicking natural movements. Similarly, in an iBMI system with reasonable amount of finite choices such as robotic-finger movement [121], a discrete state iBMI is more feasible.

The choice of effector is driven by the temporal and spatial resolution of the recording neural acquisition technology. Invasive neural acquisition technologies such as intracortical electrode array and electrocorticography give a sub-millisecond temporal resolution. Thus, these techniques are widely used with effectors requiring higher temporal resolution such as a robotic arm and computer cursor [47]. In case of effectors such as wheelchair which can tolerate a relatively lesser temporal resolution, non-invasive techniques such as EEG and fNIRS have been reported [132].

2.7 Summary

The objective of this chapter was to walk the reader through the different blocks of a typical iBMI. We briefly reviewed the nature of input neural signals followed by the constituent blocks of - signal acquisition, feature extraction, decoder and effector. The idea behind doing so was to establish a certain level of familiarity

with state-of-the-art iBMI systems for the reader, such that subsequent chapters can be easily comprehended. In subsequent chapters, original works pertaining to this thesis are presented with a detailed literature review of the problems they are attempting to solve.

Chapter 3

Towards intelligent intracortical iBMIs (i^2BMI): An offline study

3.1 Introduction

In the previous chapter, the building blocks of an intracortical Brain Machine Interface system were introduced. This system essentially involves sensing intracortical signals from the surface of the motor cortex which are then passed through several signal processing and decoding algorithmic blocks to drive effectors such as a computer cursor for communication [44, 83], robotic/natural arm for feeding [46, 86], wheelchair for locomotion [123] among others.

Despite these convincing demonstrations, significant problems persist in the adoption of these technologies for daily use. Firstly, majority of the present iBMI systems involve the use of transcutaneous wires to connect the implanted electrodes to external recording equipment. This leaves an opening in the skull thereby increasing the risk of infection. Secondly, the larger form factor and subsequent visibility of the current iBMI systems in the form of a pedestal on a patient's head run the risk of raising cosmesis concerns among end users [133]. To counter both these problems, fully implantable wireless recording systems capable of transmitting raw neural data [54] have been reported as a viable solution. However, these systems require the use of large batteries making it challenging to implant [51]. Another major concern pertaining to these systems is the limited battery lifetime amounting

to a few hours. For e.g., [54] reports battery lifetime of up to 7 hours, necessitating frequent recharging with concomitant overheating issues. This problem only worsens further with increasing channel count [53, 94] hindering scalability due to overheating [134] as well as hitting bandwidth constraints [135].

FIGURE 3.1: Block level representation of (a) conventional and (b) proposed wireless iBMI systems

An alternative proposed architecture to address this issue [136] is to do signal processing up to decoding in the implant itself as indicated in Figure 3.1(b). The underlying idea is that this approach reduces the transmitted data rate by orders of magnitude [136] leading to an increase in battery lifetime. A key point to note is that no such in-vivo hardware system has been reported so far. The work we present here is a key step towards building such a fully implantable system.

In order to build such a system, signal acquisition, feature extraction and decoder blocks ought to be implemented in the fully implantable chip. Feature extraction block typically involves spike detection and low power implementations such as [137] consuming 40 nW/channel have been reported. While including a spike detector reduces bandwidth by 100X [136], the data rate still increases with increasing number of electrodes. The data rate can be made independent of the number of electrodes by integrating the intention decoder as well. However, the challenge in that case is to have sufficiently good decoding performance in limited power and area budget. [138] reports 6030 floating point operations per update for a Kalman

Filter decoder consuming $\approx 1.82 \text{ mW}$ on a $\times 86$ processor for a 96 electrode array iBMI system driving a simple 2D cursor. The number of floating point operations are bound to dramatically rise with the increase in electrode densities and more degrees of freedom of control, ruling out direct mapping of these operations intensive algorithms on a fully implantable chip [138]. Thus, alternative computationally inexpensive algorithmic approaches have been reported in [138–143]. The lowest reported power is obtained by a neuromorphic approach that uses mixed analog-digital processing for a carefully chosen algorithm that exploits transistor mismatch for random weights[143]. However, this work only showed usage of the decoder in discrete classification tasks.

Depending on the mode of output, iBMI systems can be split into continuous mode and discrete mode systems. A discrete mode system is the one where the decoder outputs a discrete variable among a finite number of choices, an example being key selection on a keyboard [119]. A continuous mode iBMI system on the other hand outputs a continuous variable such as cursor’s velocity on a computer screen [144]. The continuous mode system is the more popularly used one owing to its generality of use in applications ranging from cursor control [44, 48] to robotic arm control [46] and paralyzed limb control [86].

The contributions in this chapter can now be summarized as follows:

- Presenting measured results from the micropower decoder chip in [143] implementing a nonlinear decoder with and without memory for a continuous state decoding using publicly available datasets from 4 primates.
- Benchmarking the results against a Kalman Filter decoder (linear decoder with memory) to show consistently better performance.

Our results pave the way for widespread usage of randomized neural network based micropower decoders to scale future iBMI systems by the next factor of 10.

3.2 Materials and Methods

We have used two different datasets to present our analysis and we will briefly describe these datasets in this section. The basic idea is to train a decoder to predict movement velocities from binned neural spikes.

3.2.1 Experiment 1

This dataset is sourced from [145, 146]. It involves NHPs A, B performing a random target acquisition task through a planar manipulandum controlling a cursor being displayed on screen as shown in Figure 3.2. Neural data acquisition was carried through a 96-channel Utah electrode array implanted in primary motor cortex (M1) and premotor cortex (PMd) areas in NHP A and primary somatosensory cortex (S1) in NHP B. 164 and 52 neurons were isolated through a manual spike-sorting process in NHP A, B respectively. The spikes from these isolated neurons were used as inputs to iBMIs. Datasets for NHP A, B correspond to one session lasting for 21 and 51 minutes respectively.

FIGURE 3.2: NHPs A, B control a cursor on screen through a planar manipulandum to reach targets flashed at random locations on a screen in a continuous manner one after another. A brief hold period is maintained between reaches. Image is obtained courtesy [145] (CC-BY license).

3.2.2 Experiment 2

Data has been obtained courtesy [147]. NHPs C, D are trained to do self-paced reaching tasks without any time intervals. A virtual reality environment has been set up wherein the fingertip of NHP drives the cursor [148]. For more details of the experiment, one is encouraged to read [148]. A 96 channel Utah electrode array implanted in the M1 area has been used for signal acquisition while an electromagnetic sensor is employed for tracking fingertip position. Simple threshold crossings occurring at each electrode were marked as spikes which were used as inputs to iBMIs. Single recording session for NHPs C, D lasting for 10 and 19 minutes respectively were used in this analysis.

3.3 Decoding methodology and algorithms

As mentioned earlier, we consider continuous state decoding in this work. The input to an iBMI system at a given instance of time, $t = i$ consists of an input feature vector $\vec{X}_i$ given as,

$$\vec{X}_i = \begin{bmatrix} r_1(i) & r_2(i) & \cdots & r_{D-1}(i) & r_D(i) \end{bmatrix} \quad (3.1)$$

where $\vec{X}_i \in \mathbb{R}^D$ and $r_z(i)$ is computed as the number of spikes occurring at the z^{th} electrode looking back in a time bin - T_{bin} ms. For every $\vec{X}_i$, the continuous mode decoder outputs a continuous variable $\vec{Y}_i$, where $\vec{Y}_i \in \mathbb{R}^p$ and p represents the degrees of freedom of control. Different algorithms employ different techniques to learn the mapping from $\vec{X}_i$ to $\vec{Y}_i$. state-of-the-art variable $\vec{Y}_i$ employed in iBMIs today is the velocity of the effector [44, 48, 138, 149, 150]. We will first briefly describe the state-of-the-art technique - Kalman filter followed by a neuromorphic alternative.

3.3.1 Algorithms

3.3.1.1 State-of-the-art - Kalman Filter

Kalman filter [151, 152] and it's variants such as unscented Kalman filter [153, 154], steady state Kalman filter [155] have been the workhorse of decoder algorithms in iBMI systems. The proponents of this technique argue for its use over other techniques such as population vector [156], linear filtering [157] algorithms based on its sound probabilistic foundation, and the fact that it utilizes information over time in a recursive fashion [152]. Kalman filter models linear relationship between $\vec{Y}_i$ (kinematic state) and $\vec{X}_i$ (neural observations) at time $t = i$ as follows [151]:

$$\vec{Y}_i = \mathbf{A}\vec{Y}_{i-1} + \mathbf{w}_i \quad (3.2)$$

$$\vec{X}_i = \mathbf{C}\vec{Y}_i + \mathbf{q}_i \quad (3.3)$$

where $\mathbf{A} \in \mathbb{R}^{p \times p}$ represents the state matrix, $\mathbf{C} \in \mathbb{R}^{D \times p}$ represents the observation matrix, $\mathbf{w}_i \in \mathbb{R}^{p \times p}$ and $\mathbf{q}_i \in \mathbb{R}^{D \times D}$ are Gaussian noise sources and are defined as $\mathbf{w}_i \sim \mathcal{N}(0, \mathbf{W})$ and $\mathbf{q}_i \sim \mathcal{N}(0, \mathbf{Q})$. Decoder training involves learning the parameters $\mathbf{A}, \mathbf{C}, \mathbf{W}, \mathbf{Q}$ from training data.

At time step $t = i$, an a priori kinematic state estimate $\vec{Y}_i^-$ from previous time step $t = i - 1$ is computed following [152] as,

$$\vec{Y}_i^- = \mathbf{A}\vec{Y}_{i-1} \quad (3.4)$$

followed by computation of error covariance matrix $\mathbf{P}_i^-$, where $\mathbf{P}_i^- \in \mathbb{R}^{p \times p}$,

$$\mathbf{P}_i^- = \mathbf{A}\mathbf{P}_{i-1}\mathbf{A}^T + \mathbf{W} \quad (3.5)$$

Using the measured firing rate $\vec{X}_i$ and the estimated a priori kinematic state $\vec{Y}_i^-$, the final estimated kinematic state $\vec{Y}_i$ is obtained as,

$$\vec{Y}_i = \vec{Y}_i^- + \mathbf{K}_i \left(\vec{X}_i - \mathbf{C}\vec{Y}_i^- \right) \quad (3.6)$$

followed by computation of posterior error covariance matrix $\mathbf{P}_i \in \mathbb{R}^{p \times p}$,

$$\mathbf{P}_i = (\mathbf{I} - \mathbf{K}_i\mathbf{C}) \mathbf{P}_i^- \quad (3.7)$$

where $\mathbf{K}_i \in \mathbb{R}^{p \times D}$ is the Kalman gain matrix and is computed as,

$$\mathbf{K}_i = \mathbf{P}_i^- \mathbf{C}^T (\mathbf{C}\mathbf{P}_i^- \mathbf{C}^T + \mathbf{Q})^{-1} \quad (3.8)$$

A simpler lower complexity alternative - the steady state Kalman filter (SSKF) has been reported in [155]. In terms of performance, there is almost no deficit as the correlation coefficient between decoded velocities from steady state and original

Kalman filter has been reported to be as high as 0.99 [155]. Kalman gain is not updated at every time instance and instead held fixed as,

$$\mathbf{K} = (\mathbf{I} + \mathbf{W}\mathbf{C}\mathbf{Q}^{-1}\mathbf{C})^{-1} \mathbf{W}\mathbf{C}^T\mathbf{Q}^{-1} \quad (3.9)$$

The kinematic state $\vec{Y}_i$ is updated based on the below equation,

$$\vec{Y}_i = (\mathbf{I} - \mathbf{K}\mathbf{C})\mathbf{A}\vec{Y}_{i-1} + \mathbf{K}\vec{X}_i \quad (3.10)$$

3.3.1.2 A Neuromorphic Alternative - Extreme Learning Machine

FIGURE 3.3: ELM neural network architecture with $\vec{X}$, $\vec{H}$ and $\vec{O}$ representing input, hidden and output layers respectively. $\mathbf{W}_{in}$, β represents first and second layer weights while $g(\cdot)$ stands for the activation function

A neuromorphic electronic system is one in which elementary physical phenomena in the form of analog signals are used as computational primitives rather than discrete digital signals [158]. The benefit of this approach is that it can lead to orders of magnitude of reduction in power consumption for pattern recognition applications [158, 159]. As far as neuromorphic compatible learning algorithms are concerned, the neural engineering framework (NEF) [160, 161] is a popular choice. The NEF essentially encodes the inputs in a nonlinear fashion using random projections, which are then subsequently linearly decoded in order to model an arbitrary function. [138, 162] reports comparison of velocity Kalman filter against NEF in a center-out iBMI task involving NHPs. A related algorithm, developed

independently in the machine learning community, is the extreme learning machine which is akin to NEF without the recurrent connections. Briefly put, ELM is a single hidden layer feedforward neural network involving fixed, random nonlinear projections in the first layer followed by a single-shot learning of weights in the second layer [163, 164]. For an input feature vector $\vec{X}_i$ defined in Equation (6.1), the corresponding hidden layer vector is given as [164],

$$\begin{aligned} \vec{H}_i &= g(\mathbf{W}_{\text{in}} \cdot \vec{X}_i + \vec{b}) \\ \vec{b}, \vec{H}_i &\in \mathbb{R}^L; \mathbf{W}_{\text{in}} \in \mathbb{R}^{L \times D}; \vec{X}_i \in \mathbb{R}^D \end{aligned} \quad (3.11)$$

where $g(\cdot)$ stands for the activation function, $\mathbf{W}_{\text{in}}$ represents the input layer weights, L denotes the number of hidden neurons and $\vec{b}$ denotes the bias values associated with hidden layer neurons.

Output $\vec{Y}_i$ corresponding to X_i , H_i is given as,

$$\begin{aligned} \vec{Y}_i &= \boldsymbol{\beta} \cdot \vec{H}_i \\ \vec{Y}_i &\in \mathbb{R}^p; \boldsymbol{\beta} \in \mathbb{R}^{p \times L} \end{aligned} \quad (3.12)$$

where $\boldsymbol{\beta}$ represents the second layer weights and p is the number of degrees of freedom of output control.

Since the first layer weights are fixed, the optimal output weights $\boldsymbol{\beta}$ can be computed in a single step as shown below,

$$\begin{aligned} \boldsymbol{\beta} &= \mathbf{H}^\dagger \mathbf{Y} \\ \mathbf{Y} &\in \mathbb{R}^{p \times n}; \mathbf{H} \in \mathbb{R}^{L \times n} \end{aligned} \quad (3.13)$$

where $\mathbf{H}^\dagger$ represents the Moore-Penrose inverse of $\mathbf{H}$. Thus, single step calculation of output weight lends to extremely fast learning while skirting around problems of

local minima typically faced in gradient-based learning algorithms like backpropagation [164].

3.3.1.3 SELMA - A Neuromorphic Processor

SELMA [143] is an ultra-low power hardware implementation of ELM with 128 channel input capacity as shown in Figure 3.4. It takes inputs in the form of spikes and has the provision of configuring bin-width to compute firing rates. It exploits inherent device mismatch in a current mirror to provide fixed first layer random weights, thereby exploiting it as a single transistor multiplier. Second layer multiplication has been implemented on a DSP in the initial version [143] and subsequent versions [165, 166] have reported an integrated second stage on the same chip. For more details, interested readers are encouraged to refer to [143, 165, 166].

3.3.1.4 ELM with memory

In addition to ELM, we have probed if adding memory to the nonlinear ELM system helps in improving decoding capability. To this end, we have added information from previous $k - 1$ time bins to the current bin at time-step $t = i$ in the form of concatenating hidden layer vectors as follows,

$$\begin{aligned} \vec{H}_{bin-i} &= \begin{bmatrix} \vec{H}_i & \vec{H}_{i-1} & \cdots & H_{i-(k-1)} \end{bmatrix} \\ \vec{H}_{bin-i} &\in \mathbb{R}^{Lk} \end{aligned} \tag{3.14}$$

This method of introducing memory is inspired from Takens's delay embedding theorem [167]. The final output is obtained in the same manner as described in Equations. 5.8 and 3.13. The hardware results of ELM with delayed input bins have been henceforth referred to as *SELMA - bins*. The optimal number of bins - k for NHPs A, B, C, D come to 3,3,2,3 respectively.

FIGURE 3.4: (a) ELM chip-level architecture [143] is presented here. Firing rates are computed from spikes received as inputs by input window counters. Digital-Analog converters translate the firing rates into currents in each input channel. Current mirror array is used to realize first layer random projection. Current controlled oscillator is employed at the hidden nodes to yield a value proportional to the current at the respective hidden node. Second layer multiplication is implemented on a MSP430 microcontroller (MCU) in this work but can be integrated on-chip as well[165], (b) Picture of the test board in the form of a Portable External Unit (PEU) made up of machine learning co-processor (MLCP) and MCU is presented here [143]. MLCP implements the first layer of computation while MCU is employed for second layer computation.

3.4 Measured Results

3.4.1 Analysis Methodology

We present comparison of KF, SSKF, SELMA, SELMA-bins described in section 3.3.1 alongside simple linear regression (LR) across the experiments reported in section 3.2. The decoded 2D components of movement velocities of decoders operating at 10 Hz is being compared.

Bin-width for computation of firing rate is typically chosen as few hundreds of ms. Accordingly, we have compared the methods while sweeping bin-widths in steps of 100 ms, while training on the first half of each dataset and testing the remaining half. Do note that the aforementioned splits happen across time.

Decoder performance is also known to be subject to the amount of training data used to train a model. Thus, models were trained on the first $(1/8)^{th}$, $(1/4)^{th}$, $(3/8)^{th}$ and $(1/2)$ of each dataset in time. Thereafter, we tested these models on held out test data corresponding to the second half of each dataset. Thus, test data is consistent across models trained on different fractions of training data. In all types of analysis, electrodes with median firing rates less than 1 Hz were excluded.

The fraction of variance accounted for (FVAF) is used as a metric of comparison. FVAF is referred to as the R^2 statistic in literature. There is some scope of confusion here as the same term, R^2 , is also used to refer to the Coefficient of Determination (CoD, square of Pearson's Correlation Coefficient). Both forms of R^2 can be summarized as:

$$R^2 = 1 - \frac{\sum_i (y_i - (\alpha \hat{y}_i + \beta))^2}{\sum_i (y_i - \bar{y})^2} \quad (3.15)$$

where $\hat{y}_i$, y_i and $\bar{y}$ stand for predicted, true and mean values of the output variable.

In case of CoD, the values of α and β are selected to minimize the mean squared error between $\hat{y}_i$ and y_i . The range of CoD statistic is from $[0, 1]$. For FVAF, the parameters α and β are set to 1 and 0 respectively. The range of FVAF statistic is from $[-\infty, 1]$. Negative values correspond to the model injecting more variance in the process than there exists in the data [168].

CoD looks for a simple correlation between actual and predicted outputs, whereas FVAF looks for a perfect match between the two. In the field of iBMIs, we aspire to obtain an ideal match between actual and predicted outputs. Hence, we have used FVAF as a metric of comparison over CoD following the methodology reported in prior works [146, 168]. R^2 is computed for each of the 2D components of movement velocity and its average value is plotted in Figure 3.6. Do note that R^2 can assume negative values and it is a sign of model introducing more variance than actually exists in the data. Furthermore, to corroborate FVAF results, the mean squared error performance is reported in Figures 3.7 and 3.9.

3.4.2 Experimental Setup

The experimental setup in this work involved passing pre-recorded spikes through a custom Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) program written on a PC to the input channels of the chip and reading back hidden layers on the PC. During the training phase, read back hidden layers ($\mathbf{H}$), target matrix ($\mathbf{Y}$) were employed to obtain second layer output weight (β) following Equation. 3.13. A key difference to note is that SELMA was employed to build a regression model in this work as compared to a classification model reported in the earlier works [143, 169, 170]. During the testing phase, read back hidden layers ($\mathbf{H}$) were multiplied by β as per Equation. 5.8 to arrive at the decoded output $\hat{\mathbf{Y}}$. Although we are presenting offline results here, SELMA has been demonstrated to operate in a real-time closed loop setup involving a discrete-mode iBMI [170] and can easily be extended to continuous-mode iBMI as well.

3.4.3 Results

3.4.3.1 Sweeping bin-widths

Figures 3.6 and 3.7 show comparison of Linear Regression (LR), KF, SSKF, SELMA and SELMA-bins as the respective decoders are trained on the first half of NHP A, B, C and D datasets and tested on the remaining half. KF outperforms LR as reported in [152]. The performance of KF and SSKF is almost identical along the lines of previously reported results [155]. SELMA outperforms KF, SSKF by

FIGURE 3.5: Example segment of decoded 2D components of velocities in NHP B comparing SSKF, SELMA and SELMA-bins decoders. SELMA, SELMA-bins outperform SSKF in the above time snippet of ≈ 3 seconds. state-of-the-art decoded variable employed in iBMIs is the velocity of the effector [44, 48, 138, 149, 150] and hence we have used velocity to compare decoders.

accounting for $\approx 14\%$, 24% , 12% , 100% more variance than that obtained by KF, SSKF for NHPs A, B, C, D respectively. SELMA-bins goes even a step further in outperforming KF, SSKF by accounting for $\approx 23\%$, 39% , 26% , 146% more variance in NHPs A, B, C, D respectively. Furthermore, a visual comparison of decoding capability of different algorithms is presented in Figure 3.5.

3.4.3.2 Sweeping training times

Figures 3.8 and 3.9 show comparison of LR, KF, SSKF, SELMA and SELMA-bins as the respective decoders are trained on varying amounts of training data across datasets with the optimal bin-width for each decoder obtained in Figure 3.6. SELMA and SELMA-bins consistently outperform KF, SSKF across NHPs and varying amounts of training times in Figure 3.6. Maximum improvements of $\approx 15\%$, 35% , 16% , 200% for SELMA and 29% , 53% , 31% , 274% for SELMA-bins have been obtained over KF, SSKF across NHPs A, B, C and D respectively.

FIGURE 3.6: Linear Regression(LR), KF, SSKF, SELMA and SELMA-bins (optimal number of bins = 3,3,2,3 for NHPs A,B,C,D respectively) models are trained on the first half period of respective datasets and the remaining half is used as testing set in (a)-(d). Plots show decoding results in the form of averaged R^2 across 2D components of movement velocity, as bin-widths are swept for computing firing rates. The proposed neuromorphic decoder - SELMA and its memory based version - SELMA-bins outperform LR, KF and SSKF decoders on all four datasets. Error bars denote standard deviation of results for randomly chosen input weights over 20 iterations for SELMA and SELMA-bins.

3.4.3.3 Statistical tests

Statistical comparison has been performed over all the test results corresponding to the swept bin-widths and swept training times cases. Thus, the number of samples used for computing p - values correspond to 8, 8, 8 and 10 across NHPs A, B, C and D respectively. Results yield p -values of 0.008, 0.008, 0.008, 0.002 for SELMA against KF and p -values of 0.008, 0.008, 0.008, 0.002 for SELMA-bins against KF

FIGURE 3.7: Plots capture Mean Squared Error (MSE) while varying binwidths for different classifiers for NHPs A, B, C and D in (a), (b), (c) and (d) respectively. The plotted results corroborate FVAF based findings (Figure 3.6) wherein SELMA and SELMA-bins outperform other classifiers.

for NHPs A, B, C, D respectively. A two-tailed Wilcoxon signed-rank test was used to obtain p -values. Since, $p < 0.01$, the results are statistically significant.

Furthermore, one can observe decoding accuracy declining with increase in the amount of training data in some cases and is prominently observed in NHP B for KF, SSKF. This counter-intuitive result is due the non-stationary nature of neural data [171–175]. However, SELMA appears to be relatively robust against non-stationarity along the lines of similar results obtained previously in a discrete control paradigm in [169].

FIGURE 3.8: Plots (a)-(d) show impact of amount of training data on decoding capability of different decoders. For each decoder, bin-width corresponding to highest R^2 in Figure 3.6 is used. The proposed neuromorphic decoder - SELMA and its memory based version - SELMA-bins outperform LR, KF and SSKF decoders on all four datasets. Error bars denote standard deviation of results for randomly chosen input weights over 20 iterations for SELMA and SELMA-bins.

3.4.4 Comparison of Computational Complexity

In Figure 3.6, we observe that KF and SSKF yield almost identical performance across NHPs and varying values of bin-widths and training data. Despite being more computationally complex, KF does not proffer any improvement over SSKF in terms of decoding capability. Thus, in Table. 3.1, we present a computational complexity comparison of only SSKF against SELMA and SELMA-bins. The number of hidden neurons is a hyperparameter which is selected based on a validation set.

FIGURE 3.9: Plots capture Mean Squared Error (MSE) while varying training data for different classifiers for NHPs A, B, C and D in (a), (b), (c) and (d) respectively. The plotted results corroborate FVAF based findings (Figure 3.6) wherein SELMA and SELMA-bins outperform other classifiers.

As an example, Figure 3.10 shows this by way of k -fold ($k = 5$) cross-validation for NHP A.

In order to compare computational complexity, we consider the number of input channels as, $D = 64$ and the number of decoded outputs as, $p = 2$. For SELMA, we consider, number of hidden layer neurons as $L = 70$ based on the value obtained in Figure 3.10 for NHP A with 64 input channels. We consider the number of bins $k = 3$ for SELMA-bins.

For the decoder considered in Table. 3.1 and operated at 10 Hz, the power consumption comes to be around 13.2, 34.2 and 62.2 nW for SSKF, SELMA and

FIGURE 3.10: Validation accuracy is plotted for varying values of number of hidden nodes for NHP A's training data with 64 input channels. R^2 saturates around 70 hidden layer neurons in this case.

TABLE 3.1: Computational Complexity Comparison for a 2-D iBMI decoder

Algorithm/ Hardware	Analog MACs	Digital MACs	Energy/ update(nJ)	Power dissipation (nW)
SSKF	-	132	1.32	13.2
SELMA	4480	140	3.42	34.2
SELMA-bins	4480	420	6.22	62.2

MAC = multiply and accumulate, Energy/(Analog MAC) = 0.45 pJ [143], Energy/(Digital MAC) = 10 pJ [176], Decoder update frequency = 10 Hz, Number of input channels, $D = 64$, Number of hidden layer nodes in SELMA, $L = 70$, Number of time-bins, $k = 3$ and Number of outputs, $p = 2$.

SELMA-bins respectively. Furthermore, if we consider a relatively advanced case of a decoder with a 10 degree of freedom output to control an anthropomorphic arm [174] with the remaining decoder parameters same as Table. 3.1, the energy consumption comes to be ≈ 7.4 , 9 and 23 $nJ/update$ for SSKF, SELMA and SELMA-bins respectively. Thus, one can conclude that SELMA consumes energy in a comparable range to SSKF whereas SELMA-bins consumes relatively more energy than SSKF. The reason for energy consumption of SELMA to be in the same realm as SSKF, despite an overall greater number of operations, is the parsimonious analog implementation of the first layer. For example, if ELM were to be implemented using conventional digital hardware, energy consumption

for the aforementioned case would come to be significantly greater around ≈ 46.2 nJ/update .

3.5 Discussion

State-of-the-art iBMI decoders [44, 48, 138, 149, 150] employ velocity as the decoded effector variable as it has been shown to be superior to position based decoders [149]. Hence, we have based our analysis presented in Figures 3.6, 3.7, 3.8, 3.9 and 3.5 on decoding velocity.

Kalman filter models linear relationship between input and output variables and works best when the input noise is Gaussian. However, in practice often the assumptions of linearity and Gaussian nature of recording noise don't hold true in case of iBMIs [148]. ELM on the other hand, by virtue of random projection translates the input into a nonlinear higher dimensional space, thereby increasing the generalization ability of the model as stated in Cover's theorem [177]. Furthermore, we observe that this nonlinear decoder (ELM) is significantly helped by introduction of memory in the system by virtue of using delayed inputs along the lines of similar improvements reported in [143].

There have been recent initial offline studies which have shown even more complex nonlinear algorithms with memory such as Long Short Term Memory (LSTM) neural networks outperforming SSKF [150, 178]. However, we can observe three critical shortcomings in an LSTM-based approach. Firstly, it requires a larger amount of training data. For e.g., [178] reports using data spanning around 73 calendar days while using LSTM. Secondly, complex neural networks are known to overfit to noise and optimally tuning the hyper-parameters is often a daunting impediment to overcome [179]. Third and most importantly, these neural networks are much more computationally complex than both SSKF and SELMA. This results in higher training times and often dedicated hardware such as graphic processing units (GPUs) is required for training, making the system cumbersome to use [180]. Furthermore, the number of MACs involved for the LSTM network presented in [178] to control a 2D cursor comes to ≈ 22900 . This is around two orders of magnitude more computationally complex than SSKF, SELMA and possibly rules out the prospect of implementing the decoder in the implant itself. Note that we

have considered the same input and output conditions for LSTM as was used in Table. 3.1.

FIGURE 3.11: Comparison of transmission data rates between systems transmitting a) raw data, b) firing rates and c) decoded outputs for a 10D iBMI operating at 50 Hz. It can be seen that the system with decoded outputs is more scalable since its output data rate is orders of magnitude lower than both other options.

Electrophysiological signal acquisition techniques are advancing expeditiously with the latest probes [94, 181] capable of recording up to thousands of neuron channels. [53] reports a Moore-like law with the number of simultaneous neurons recorded doubling every ≈ 7 years over the last 5 decades. We consider a case of an iBMI controlling a 10D anthropomorphic arm at an operating frequency of $f_{op} = 50$ Hz. For such an application, we compare the resulting transmission data rates across systems transmitting a) raw data, b) firing rates and c) decoded outputs for varying number of input channels (N_{chan}) in Figure 3.11. Systems transmitting firing rates involve integration of a spike detector followed by a counter to yield firing rates at frequency - f_{op} for every input channel. In addition to the spike detector and counter blocks, systems transmitting decoded outputs involve integration of the decoder block as well in the implant as seen in Figure 3.1(b). Transmission data rates for the different cases can be computed as per the formulation below,

$$\begin{aligned}
Data_rate_{RawData} &= N_{\text{chan}} \times f_{\text{ADC}} \times b_{\text{ADC}} \\
Data_rate_{FiringRate} &= N_{\text{chan}} \times b_{\text{firing-rate}} \times f_{\text{op}} \\
Data_rate_{DecodedOutputs} &= p \times b_{\text{op}} \times f_{\text{op}}
\end{aligned} \tag{3.16}$$

where, calculation considerations in Figure 3.11 involve ADC Sampling rate (f_{ADC}) = 30 kHz, ADC bit resolution (b_{ADC}) = 10, Firing rate bit resolution ($b_{\text{firing-rate}}$) = 8, iBMI operation frequency (f_{op}) = 50 Hz, Number of outputs (p) = 10 and Decoded output bit resolution (b_{op}) = 8.

TABLE 3.2: Adding signal processing in the implant

Systems Output	Data rate (kbps)	Transmission power (W)	Added blocks power dissipation (μW)	Power dissipated normalized to Decoded Outputs (SELMA) system*
Raw Data	3×10^5	1.13	-	11728
Firing rate	400	$1.5m$	40	16
Decoded Outputs (SELMA-bins)	4	15.1μ	111.25	1.3
Decoded Outputs (SELMA)	4	15.1μ	81.25	1

*Power dissipation of all the blocks post signal acquisition in the implant is considered, Wireless transmitter power dissipation = 3.775 nJ/bit [54], $N_{\text{chan}} = 1000$, (Spike-detector Power consumption)/(input channel) = 40 nW. Thus, spike-detector's power consumption = 40 μW . SELMA is used as a decoder with $D = 1000$, $L = 1500$, $p = 10$. Accordingly, decoder power consumption equals 41.25 μW . For SELMA-bins we have considered time-bins, $k = 5$ with the same considerations as SELMA for D, L . Power consumption for SELMA-bins decoder comes to 71.25 μW .

Thus, one can observe that integrating signal processing blocks beyond conventional signal acquisition block in the implant leads to at least 3 or more orders of magnitude of transmission data rate reduction. The system transmitting decoded outputs is of particular significance since it makes the transmission data rate essentially agnostic to the number of input channels. However, one can be rightfully concerned about the increase in the power consumption due to - a) addition of firing rate computation block in system transmitting firing rates and b) addition of firing rate and decoder blocks in system transmitting decoded outputs. To gain an understanding about the amount of overall system level power saving, let us again consider a $10D$ iBMI operating at 50 Hz with $N_{\text{chan}} = 1000$. Data rate was computed using formulation in Equation. 3.16 and transmitter power dissipation was found by multiplying data rate to reported transmitter power efficiency of 3.775 nJ/bit [54]. In the firing rate computation block, only spike detection block was considered for simplicity with 40 nW/channel [137] power consumption. Finally, SELMA and SELMA-bins were employed as decoders considering a reasonably large number of hidden layer neurons, $L = 1500$. We consider the number of time-bins to be reasonably high at $k = 5$ for SELMA-bins. *Analog MACs* = 15×10^6 come out to be the same for SELMA and SELMA-bins whereas *Digital MACs* = 15×10^3 and *Digital MACs* = 75×10^3 for SELMA and SELMA-bins respectively. $\text{Energy}/(\text{Analog MAC}) = 0.45$ pJ [143], $\text{Energy}/(\text{Digital MAC}) = 10$ pJ [176]. The normalized power dissipation figure in the last column of Table. 3.2 shows a significant amount of power savings attained by virtue of incorporating both firing rate extraction and decoder blocks in the implant compared to the other two cases, thereby increasing battery lifetime.

A possible limitation of integrating intelligence in the implant is that it might run into debugging issues in case the implant fails to work. To circumvent this, we propose adding a *debug* mode functionality of transmitting either raw data or firing rate for a brief period of time to investigate the cause of failure.

3.6 Conclusion

We have presented offline hardware results of a neuromorphic chip - SELMA with and without memory across different experimental setups and NHPs. Consistent improvement over state-of-the-art technique - KF and SSKF has been achieved

over different operating parameters of bin-widths and training data corpus. Furthermore, we have also presented an energy consumption comparison for different use cases and found SELMA to consume power in the same range as simple linear SSKF. The reduction in power consumption while implementing ELM on SELMA is primarily attributed to the ingenious trick of using ultra low-power analog computation technique in the first layer. SELMA-bins on the other hand gives relatively improved results over SELMA with the rider of moderately increased power consumption as shown in Table. [3.2](#).

Chapter 4

Towards intelligent intracortical iBMIs (i^2BMI): An online study

4.1 Introduction

In the previous chapter, we presented a detailed case to embed intelligence in the implant. We christened this system as an intelligent intracortical BMI (i^2BMI). iBMI based studies typically begin with offline analysis followed by real-time on-line implementations. The idea is to get closer to an actual use-case scenario and subsequent clinical translation. However, as seen in Table 4.1, no real-time implementation has been reported yet.

TABLE 4.1: Summary of i^2BMI works

Works	Hardware	Bench- marking	Real-time analysis	Animal experiment
[139]	Yes	No	No	Rodent
[138]	No	Yes	No	NHP
[143]	Yes	No	No	NHP
[140]	Yes	No	No	Rat
[182]	Yes	Yes	No	NHP

NHP stands for Non-human primate

[138–140] involve systems based on a spiking neural network framework of learning. Among these works, [139, 140] have reported results only on offline pre-recorded

data while not comparing it against state-of-the-art techniques. [138] has addressed these shortcomings of benchmarking and showcasing real-time analysis. However, no hardware implementation was demonstrated in [138]. The objective of this chapter, thus, is to present a real-time fully brain controlled closed loop performance of the neuromorphic decoder - SELMA. In particular, the neuromorphic decoder's performance is compared against native hand-control following the paradigm reported in [48].

4.2 Methods

4.2.1 Neural Signal Acquisition and Signal Processing

Two floating microwire arrays (32 electrodes each) were implanted in an NHP (*Macaca fascicularis*) in the left primary motor cortex area. Ripple's grapevine neural recording system [52] was used for real-time recording of raw neural data sampled at 30 kHz. Spikes were detected using median absolute deviation method [80] with the threshold set as negative of five times the median. Inputs from all 64 input electrodes were considered in the analysis.

4.2.2 Behavioural Task

The behavioural task is performed in two modes of control -

1. Native hand control
2. Brain-decoder control

4.2.2.1 Native hand Control

An NHP is trained to maneuver a joystick using its left hand to drive a virtual wheelchair avatar towards a square target appearing on a computer screen in the hand-controlled joystick mode. In a given trial, a wheelchair avatar starts at the centre of the screen and the target is presented in a pseudo-random manner in one

FIGURE 4.1: Task - A wheelchair avatar starts from a central position on the computer screen in every trial with a target displayed pseudo-randomly in one of the three square-shaped locations.

of the three different directions – Forward, right and left in the form of a square of side 2 cm as seen in Figure 4.1. The wheelchair avatar is driven in a discrete control fashion at a frequency of 10 Hz with possible position updates being one of left, right, forward or stop depending on the position of the joystick. A trial is considered successful if the NHP reaches, and stays in the target area for 1.5 seconds under a total elapsed time of 13 seconds. The virtual wheelchair program is written using Psychophysics toolbox [183] in MATLAB on a PC. We refer to the PC running the virtual wheelchair program as the stimulus PC.

4.2.2.2 Brain-Decoder Control

Once the NHP was trained well enough to perform the trials consistently with greater than 80% success rates, we proceeded to the brain-decoder control mode. In this mode, the joystick is removed from the system. The wheelchair avatar movements are a result of the decoded outputs obtained from the neuromorphic decoder. Spike firing rates computed in backward-looking windows of time, $T_w = 500$ ms serve as input to the neuromorphic decoder. Firing rate vector, X_i at a given instant of time, $t = i$ is given as,

$$\vec{X}_i = \begin{bmatrix} r_1(i) & r_2(i) & \cdots & r_{D-1}(i) & r_D(i) \end{bmatrix} \quad (4.1)$$

where D represents the number of input channels ($D = 64$) and $r_z(i)$ represents the number of spikes appearing at z^{th} electrode in a lookback time window of T_w seconds.

FIGURE 4.2: Experimental setup for closed loop decoding with SELMA. Spikes are detected by the Ripple neural recording system and passed on to SELMA for subsequent decoding. Universal Asynchronous Receiver/Transmitter (UART) is used to establish communication between the microcontroller on SELMA board and the Ripple neural recording system. The final decoded output is passed to the stimulus PC via an ethernet cable. The stimulus PC runs a Psychophysics toolbox based virtual wheelchair program.

Correspondingly, the decoded output vector is given as,

$$\vec{Y}_{Di} = \begin{bmatrix} o_1(i) & o_2(i) & \cdots & o_p(i) \end{bmatrix}; \vec{Y}_{Di} \in \mathbb{R}^p \quad (4.2)$$

where $p = 4$ and represents one of the following choices - *left*, *right*, *forward* and *stop*. $\text{argmax}(\vec{Y}_{Di})$ gives the final decoded output.

4.2.3 Real-time interface

Real-time online decoding involves establishing synchrony between the following components depicted in Figure 4.2:

- PC interfaced with Ripple neural recording front-end via an ethernet cable (jointly referred to as the Ripple Neural Recording system)
- SELMA-chip
- Stimulus PC running the wheelchair avatar program

The Ripple Neural recording front-end yields spikes recorded in real-time from 64 implanted electrodes to an ethernet-linked PC. These spikes are read on the linked PC via a Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) based interface, Xippmex. Furthermore, a custom program is written in Matlab R19a to handle the real-time interaction between the Ripple front-end and SELMA-chip.

The spikes are passed as electrode channel addresses at which they occur to the SELMA chip. The SELMA chip uses a demultiplexer to translate the channel address to activate the appropriate input row with a spike. SELMA supports a maximum of 128 input channels. The input channels that receive spikes are automatically enabled and the ones that do not are left out of the picture.

SELMA supports a maximum of 128 hidden layer neurons. The hidden layer outputs are read by a Texas Instruments (TI) MSP430 low power microcontroller, which implements the second layer multiplication stage. The number of hidden neurons to be used can be configured via the TI microcontroller. In the current experiment, the second stage was configured to yield decoded outputs at 10 Hz. These outputs are passed over an ethernet cable to the stimulus PC's Matlab R19a program, which then drives the wheelchair avatar on screen. The end-to-end latency from spike detection to wheelchair avatar movement is $\approx 10\text{ ms}$. This latency is acceptable in our experimental scenario which requires an output only every 100 ms .

The experiment is split into 3 sessions carried out on a given day, with each session comprising 20 successful trials. A digital trigger signal is sent from the stimulus PC's Matlab R19a program at the beginning and end of every trial. These triggers are used to synchronize working between SELMA-chip and the stimulus PC. The real-time interface is set up to operate in two modes - *training* and *decoding* mode. Do note that the first layer weights are fixed during both training and decoding phase.

4.2.3.1 SELMA - training mode

Flowchart depicted in the Figure 4.3 captures the real-time interface operating in *training* mode. The objective in the training mode is to obtain the hidden layer

matrix, $\mathbf{H}$ such that the output weight beta, β is learned following the below equation [163].

$$\begin{aligned}\beta &= \mathbf{H}^\dagger \mathbf{Y}_D \\ \mathbf{Y}_D &\in \mathbb{R}^{p \times n}; \mathbf{H} \in \mathbb{R}^{L \times n}\end{aligned}\tag{4.3}$$

where $\mathbf{H}^\dagger$ represents the Moore-Penrose inverse of $\mathbf{H}$, $\mathbf{Y}_D$ represents the target output vector matrix, L represents the number of hidden layer neurons, p represents the number of outputs ($p = 4$) and n represents the number of time-steps. Thus, single step calculation of output weight lends to fast learning times of ≈ 80 ms.

The decoder was trained on a total of 3 sessions consisting of 20 trials each. The NHP was made to passively observe a series of successful trials for the first couple of sessions following the training paradigm in [48, 123]. This paradigm is popularly known as observation-based learning in iBMIs and is predicated on the existence of 'mirror' neurons in the motor areas [184]. 'Mirror' neurons fire during both observation and execution of goal directed movements. The target output vector, $\vec{Y}_{Di}$ corresponds to the position of the target at every instance of time. This is based on the assumption that the intention of the NHP at every instance of time, $t = i$ is to drive the cursor towards the target [48]. An intermediate model M_{int} was trained based on the first two sessions.

In the third session, M_{int} was employed in closed-loop with 50% assistance [123, 185] in the system. Thereafter, the final trained model M_f was obtained including data from all three sessions. In the third session, the target output vector is obtained in a manner similar to the first two sessions. Only successful trials were used for training the model.

4.2.3.2 SELMA - decoding mode

Flowchart depicted in Figure 4.4 capture the decoding mode. Here, the learned matrix β (see Equation 4.3) is loaded in the TI-MSP430 microcontroller at the beginning of a decoding session. Real-time decoded output vector at time, $t = i$ is obtained as [163],

FIGURE 4.3: The TI-MSP430 microcontroller on board the SELMA-chip is interrupted by the digital trigger signal at the beginning of every trial. Thereafter, the hidden layer vectors are saved in microcontroller memory every 100 ms until the end of the trial. These hidden layer vectors are read back at the end of every trial to eventually make up the hidden layer matrix.

$$\begin{aligned}
 \vec{Y}_{Di} &= \beta \cdot \vec{H}_i \\
 \vec{H}_i &\in \mathbb{R}^L; \vec{Y}_{Di} \in \mathbb{R}^p; \beta \in \mathbb{R}^{p \times L}
 \end{aligned} \tag{4.4}$$

where $\vec{H}_i$ is the hidden layer vector.

FIGURE 4.4: The learned weight matrix β is loaded in the microcontroller at the beginning of a decoding session. Decoded outputs from SELMA-chip are used to drive the wheelchair avatar on the stimulus PC at the frequency of 10 Hz.

4.3 Results

The trained model M_f was employed in closed-loop, while the NHP performed 20 successful trials in each session. Three such sessions were recorded until a total of 60 successful trials were recorded each day. Thus, a total of 180 brain-controlled successful trials were obtained.

Similarly, 180 native hand-control trials were obtained spread across three days, with 60 trials on each day. The native hand control and brain-controlled decoder

FIGURE 4.5: Mean times to complete trials are comparable for closed-loop decoding by SELMA and hand controlled joystick experiments. Statistically insignificant difference is denoted by (*), $p > 0.01$ (unpaired two-sample t-test)

results are compared in terms of speed and accuracy in Figures 4.5 and 4.6 respectively.

Mean successful trial completion speeds for forward, left and right targets by brain control have been shown to be $\approx 79\%$, 93% and 90% respectively of those obtained by hand-controlled joystick. The results are plotted in Figure 4.5.

Mean success rates for closed-loop brain control is $\approx 96\%$ of that obtained by hand-joystick control as shown in Figure 4.6.

FIGURE 4.6: Percentage of successful trials for hand-controlled joystick and neural control. Statistically insignificant difference is denoted by (*), $p > 0.01$ (unpaired two-sample t-test)

4.4 Discussion and Limitations

This experiment requires at least $\approx 70\%$ [123] of the decoded directions to match the target position for a trial to be successful. Chance level accuracy for the trained classifier model is 25%, so the chance level of success of a trial can be computed as $0.25^{(0.7 \times 130)} \times 100 \approx 0\%$.

ELM algorithm is known to take around 1.5 to 3 times the number of hidden neurons as fully tuned deep neural networks to show similar levels of performance [186]. This is due to the fact that the first layer is made up of fixed random weights which are not fine-tuned. However, do note that the analog implementation of random weights leads to 10X or more power savings making this approach attractive [143]. SELMA currently supports 128 input channels. This can be a limitation to adjust to the scaling of recorded input neurons [53]. Accordingly, updated versions of SELMA have to be implemented to support increasing the input channel counts and this remains an area of future work.

The gold standard for iBMI experiments before proceeding to human trials is to exhibit closed loop results in two NHPs. We have showcased closed loop results in only one NHP and this remains a limitation of this work.

4.5 Conclusion

From infection mitigation, scalability, portability and mobility standpoints, a fully implantable iBMI is the one plausible approach. Neuromorphically engineered iBMI systems have long been touted as a viable option, albeit without hardware-based demonstration in standard NHP based experimental setups. We have demonstrated for the first time fast and accurate neural control obtained via a neuromorphic chip decoder in a closed loop NHP based experimental setup. The decoding accuracies are similar to those obtained in hand-controlled joystick experiments paving the way for future implanted system development.

Chapter 5

Sparse Ensemble Machine Learning to improve robustness of long-term decoding in iBMIs

5.1 Introduction

Intracortical Brain Machine Interfaces (iBMIs) have made significant strides from the first neuronal population level studies reported in non-human primates (NHPs) [127, 187] to the recent impressive real-time demonstrations as seen in the previous chapter [44, 46, 86, 188]. Despite these compelling advances, barriers to clinical translation remain. One of the most pressing issues is the non-stationary nature of neural recordings (Figure 5.2). Recording condition changes such as micro-motion of electrodes (Figure 5.1), development of scar tissue, changes in electrode’s impedance are predominantly responsible for the exhibited nonstationarity across time [171–175]. Due to these changes, a decoder trained at a given point of time is often rendered unusable after tens of minutes to a few hours [76, 173, 189, 190].

To counter nonstationarity, the most commonly employed approach is to re-train the decoder every day using a set of supervised trials at the beginning of a recording session. The subject is made to perform a set of calibration trials and target labels are collected along with neural activity [172, 189–191]. This approach leads to an increased setup time and necessitates the presence of an expert to get the system up and running. Data collected from previous days/sessions is discarded and the

FIGURE 5.1: Micro-motion of an electrode array leads to appearance/disappearance of a neuron’s signals at a particular electrode as seen in recording conditions 1,2. Furthermore, in a more dire scenario as seen in recording condition 3, certain electrodes break down and stop yielding any meaningful recorded signal. Figure is not drawn to scale and is adapted from [49] (CC-BY license).

decoder utilizes only the most recently collected data to learn the mapping between neural activity and behaviour.

Despite the deleterious phenomenon of nonstationarity, studies have shown some commonality exhibited between neural recordings across time [173, 193, 194]. Accordingly, there are alternate approaches that aim to utilize this similarity to build robust iBMIs. For example, researchers in [77] have proposed a principal component analysis based domain adaptation (PDA) technique to reduce decoder calibration times by utilizing historical data adapted with a small sample set of current day’s data. However, consistent decoding results have only been reported for a maximum of three day period of separation between training and testing days and it remains to be seen if this technique still holds over a longer period of separation. Besides, it still resorts to explicit collection of supervised trials data in the beginning of a recording session necessitating suspension of decoder use even if that means for a relatively lesser time than standard calibration times for training a *de novo* decoder.

In order to avoid suspension of decoder use, an alternate strategy is to update the decoder in an unsupervised manner [76, 79, 195]. These techniques rely on tracking and updating changing neural statistics while the decoder is still in use. Authors in [76, 79] have presented a decoder being updated in a retrospective fashion after every block of trials in a human subject based point and click cursor experiment. Sustained performance has been reported employing this approach in a closed loop study involving 11 sessions spanning 29 days [79] and 6 sessions spanning 42 days [76] in two human subjects. However, for a decoder to be updated in a retrospective fashion, there is an implicit assumption about the latest calibrated decoder

FIGURE 5.2: The principal angle is a parsimonious scalar metric to measure the similarity of neural activity across days [49, 192]. (a) and (b) present principal angles between the subspaces of neural activity recorded each day for NHPs E and F respectively. Smaller principal angles denote greater similarity and one can see recorded days being more similar to each other in NHP F than NHP E. (c) presents measured signal to noise ratio (SNR) values following the methodology given in [125] with mean (solid line) and standard deviation (shaded region) values across electrodes plotted for recorded days in NHPs E and F. As far as signal stability is concerned, we can see the average SNR values across days to be stable across days for both NHPs. NHP E shows greater variance in the SNR values among its electrodes compared to NHP F, showing that some electrodes capture better spikes than the others.

performing with sufficient accuracy to complete the stated task. [195] reports a self-calibrating method that improves performance over a non-updated decoder in an offline setting in a discrete-control experiment. Results abide by the standard protocol of demonstrating improvements in two NHPs and sustained improvements

have been shown to last about 35 and 40 days in two NHPs. However, this approach requires a ‘burn in’ period [195] before the decoder updates to an optimum level and matches the current day/session’s distribution to yield better results.

Continuing along the lines of eliminating suspension of decoder use, there is another school of thought which focuses on harnessing a large magnitude of past data and builds a data driven robust solution to counter nonstationarity. This approach banks on computationally complex algorithms such as recurrent neural networks (RNNs) [49], long short term memory units (LSTMs), convolutional neural networks (CNNs) [78] to learn complex mappings between heterogeneous neural data across days and effector variables. [49] presents consistent offline performance over 55 and 59 days for two monkeys involved in a hand based target acquisition task employing a fixed recurrent neural network as a decoder. Similarly, [78] presents a deep neural network based decoder employing computationally complex LSTM neural networks to yield consistent offline results over 381 days in a human subject based study. Although both of these offline results are novel and impressive in terms of their sustained performance, the amount of data reported to optimally train these complex models is concerning. Furthermore, these models require hand-tuning of hyper-parameters [49, 78, 196] coupled with specialized hardware during the training phase. This makes it perhaps difficult to optimally apply these approaches in a clinical setting.

Taking note of the problems faced by the reviewed approaches, we present a general sparse ensemble decoding approach that performs consistently better than corresponding single classifier models of varying complexity in the face of incoming non-stationary data. Furthermore, we also present data driven hyper-parameter tuning guidelines in *Section 5.4.1* for different classification techniques to make this approach easier to use.

5.2 Materials and Methods

We have based our studies on two different sets of experiments each involving a different NHP as seen in Figure 5.3. Accordingly, our subsections are split into two parts corresponding to experiments 3 and 4. Two adult male macaques (*Macaca fascicularis*) have been used in the experiments. All conducted animal procedures

abide by the standards of the Agri-Food and Veterinary Authority of Singapore and the Singapore Health Services Institutional Animal Care and Use Committee (Singhealth IACUC 2012/SHS/757). Moreover, the procedures also comply with the recommendations prescribed in [197]. The animals were implanted with a titanium head post (Crist Instruments, MD, USA) prior to implantation of micro-electrode arrays (MicroProbes, MD, USA - NHP E; Blackrock, UT, USA - NHP F) in the hand/arm region of the primary motor cortex.

5.2.1 Signal Acquisition and Processing

5.2.1.1 Experiment 3

4 microelectrode arrays containing 16 electrodes each were implanted in the hand/arm region of the left primary motor cortex in NHP E. A 100-channel neural recording system [98, 199] was used for acquiring neural data sampled at 13 kHz from the implanted 64 electrodes. This raw data was then digitally band-pass filtered in Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) between 300 and 3000 Hz [123]. Typically, absolute threshold (AT) based method [80] is employed for the purpose of spike detection on commercial recording equipment. However, for this custom recording setup [98, 199], we found nonlinear energy operator (NEO) based spike detection method [101] to be more effective over AT, and hence we used NEO to isolate spikes. The probable reason for this is the increased input-referred noise in custom recording setups compared to the commercial ones [58, 200]. First 30 seconds of filtered raw data $x[n]$ on each day was used to set the spike detection threshold following the method given in [101] as,

$$\psi[n] = x^2[n] - x[n-1] \times x[n+1] \quad (5.1)$$

$$Thr_{NEO} = l \times \frac{\sum_{n=1}^N \psi[n]}{N} \quad (5.2)$$

where $\psi[n]$ is the NEO processed, pre-emphasized signal at time n . Thr_{NEO} is then computed as a multiple - l of average of $\psi[n]$ over N samples of filtered raw data. The optimal value of l is found to be 13. Threshold was computed at the beginning of each day and was held fixed for the entire day's analysis. NEO

FIGURE 5.3: (a) Shows an NHP trained to control a joystick. Joystick movement was restricted to right, left and forward directions only. In the top portion, a multi-electrode array is shown to be implanted in the hand and arm regions of primary motor cortex [123], (b) Experiment 3 setup: NHP E was seated in a robotic wheelchair. The wheelchair translated in the forward, left or right directions in discrete time steps of 100 ms depending on the position of the joystick [123], (c) Experiment 4 setup: NHP F is trained to drive a virtual wheelchair using a joystick in a manner similar to Experiment 3. Figures (a), (b) are adapted from [123] (CC-BY license) and Figure (c) is adapted from [198] (CC-BY license)

processed filtered raw data surpassing the threshold were classified as spikes and were used for subsequent analysis.

5.2.1.2 Experiment 4

1 microelectrode array containing 100 electrodes was implanted in the hand/arm region of the left primary motor cortex in NHP F. Ripple (UT, USA) grapevine neural interface processor was used for recording raw data sampled at 30 kHz from two front-ends capable of recording 32 channels each. Ripple front-end processor provided bandpass filtered data between 250 and 7500 Hz to yield $x[n]$ [52]. In a commercial recording setup with less recorded noise, the performance of NEO and AT are very close to each other [100]. Accordingly, in this setup, we observed AT was doing a reasonable job at spike detection. Hence, we used the relatively more popular technique, AT, for this setup. Similar results are expected with NEO as well. We used the first 10 seconds of each day's recording [80] as,

$$\sigma_N = \frac{\text{median}|x[n]|}{0.6745} \quad (5.3)$$

$$Thr_{AT} = k \times \sigma \quad (5.4)$$

where σ_N is the standard deviation estimate over N samples of filtered raw data $x[n]$, k is an integer typically between 2 and 5. For this dataset, we have employed $k = 3$. Spikes were computed as negative threshold crossings.

5.2.2 Behavioural Tasks

The experiments involved an NHP trained to maneuver a joystick to reach a visual target (see Figure 5.3). The joystick was spring-loaded to return to the centre position when released. Furthermore, it was restricted to move only in forward, right and left directions. The joystick position was tracked every 100 ms, and the instantaneous position of joystick was used to drive an effector in either of the three aforementioned directions or remain stationary. Appropriate joystick position thresholds were defined in each direction such that the effector moved once the position threshold was exceeded.

5.2.2.1 Experiment 3

This experiment has been described in detail in [123]. Briefly, the experiment consisted of four tasks - a) moving forward by 2 m, b) turning 90° right, c) turning 90° left and d) staying still for 5 seconds (*stop* task).

A trainer holding a piece of fruit reward served as a visual target for NHP E seated in a robotic wheelchair. A movement related trial was considered successful and rewarded with a piece of fruit, if the NHP reached the trainer within 15 seconds. For the stop trial, a task was considered successful and rewarded with a piece of fruit, if the monkey stayed still for a period of 5 seconds. The control of the robotic wheelchair was turned on at the beginning of each trial and was automatically turned off when it reached the target.

5.2.2.2 Experiment 4

In this experiment, a target is presented in a pseudo-random manner in one of the three different locations – Top, right and left relative to the centre of the screen in the form of a red square of side 2 cm on a black background. NHP F is seated in front of the screen in a primate chair. It is trained to drive the wheelchair avatar from the centre of the screen towards the target area. Once the wheelchair reaches the target area, the target’s colour changes from red to green and it serves as a cue for the NHP to stop moving. If the NHP manages to reach and stay in the target area under a total elapsed time of 13 seconds, the trial is considered successful and a juice reward is dispensed. The wheelchair avatar is driven at a frequency of 10 Hz in a manner similar to experiment 3, wherein instantaneous joystick signals are tracked to update the position of the wheelchair avatar.

5.3 Proposed Decoding Algorithm

5.3.1 Analysis Methodology

Incoming spike firing rates at time $t = i$ denoted as $r_z(i)$ are computed at each of the D electrodes as the number of spikes occurring at a particular electrode z looking back in a time window $T_w = 500$ ms to yield input feature vector $\vec{X}_i$ as,

FIGURE 5.4: Partitioning of data for training and testing to benchmark fixed models against daily retrained models

$$\vec{X}_i = \begin{bmatrix} r_1(i) & r_2(i) & \cdots & r_{D-1}(i) & r_D(i) \end{bmatrix} \quad (5.5)$$

$\vec{X}_i'$ s are computed every $T_s = 100$ ms for the entire duration of a trial. This means that there is a 400 ms overlap between consecutive $\vec{X}_i'$ s. Correspondingly, joystick signals are tracked every T_s to yield the target value represented as a one-hot encoded vector - $\vec{Y}_i$ at time $t = i$. Target value can correspond one of the four discrete values of *forward*, *right*, *left* and *stop* in our reported experiments. Accumulating $\vec{X}_i'$ s and $\vec{Y}_i'$ s over all successful trials across all training sessions yields training matrix $\mathbf{X}$ and training target $\mathbf{Y}$, where $\mathbf{X} \in \mathbb{R}^{D \times n}$, $\mathbf{Y} \in \mathbb{R}^{C \times n}$, n is the accumulated total number of instances across all training sessions and C is number of possible discrete target values. Only successful trials are used in this analysis. We used all testing data instances corresponding to successful trials in both NHPs. In NHP E, there was no problem of class imbalance, so we used all the training data instances corresponding to the successful trials. In NHP F, owing to the stop condition in each trial, there was a class imbalance problem. Hence, we employed an equal number of training data instances belonging to each class – *forward*, *right*, *left* and *stop* sampled randomly on each day. Training a decoder involves learning a mapping between $\mathbf{X}$ and $\mathbf{Y}$. Thus, a decoder outputs one of the four discrete outputs of *forward*, *right*, *left* and *stop* every $T_s = 100$ ms.

We have used a total of 35 sessions across 8 days recorded over a period of 66 days for NHP E. Typically, each day consists of 3 – 5 sessions and each session consists of 20 successful trials. Similarly, for NHP F, we have used a total of 30 sessions across 10 days recorded over a period of 77 days. For NHP F, we recorded 3 sessions every day with each session consisting of 60 successful trials. The number of input channels (D) correspond to 64 in both NHPs E and F.

In Figure 5.4, we present the methodology of partitioning of data to be used as training and testing sets. We train fixed models on days 1 – 3 for both datasets and compare them against daily retrained models on the remaining days. The training days span a period of 12 and 17 days for NHPs E and F respectively. In case of daily retraining, we have used 5 trials of each type in NHPs E and F. One must note that the test data for evaluating both types of models is the same.

5.3.2 Sparse Ensemble Machine Learning

Ensemble methods employ a divide and conquer approach, wherein a set of classifiers attempt to independently learn a hypothesis each, which when combined effectively translates into a more robust classifier [201, 202]. In our case, the ensemble is made up of M base learners as shown in Figure 5.5. The recording channels exhibit a certain degree of variability in firing rates across days. Micro-motion of electrodes as shown in Figure 5.1 are a probable reason for this variability as neural signals appear/disappear at the electrodes. Thus, each of the classifiers in the ensemble are trained on a different set of input recording channels - $\vec{X}^j$, sampled with replacement from the entire set $\vec{X}$, thereby reducing bias and contributing towards robustness [203]. The channels in each set are chosen randomly and the optimal number of channels (U) in each set are computed as,

$$U = q \times \text{int}(\log_2 D + 1) \quad (5.6)$$

based on the suggestion imparted by Leo Breiman in his seminal Random Forest paper [204]. D as stated previously represents the total number of input channels and q is an integer hyper-parameter. The number of channels chosen in each set

FIGURE 5.5: Schematic representation of sparse ensemble machine learning is presented here. Input to each of the M Base Learners (BLs) - $\vec{X}^j$ is obtained by sparsely sampling the entire input space - $\vec{X}$ with replacement. Outputs $\vec{o}^j$ from all M constituent BLs are combined to obtain the final decoded vector $\vec{o}$

dictates the strength of each classifier and the correlation between them. A certain degree of heterogeneity is desired among the different sets [201, 202, 204].

Mathematically, $\vec{X}^j$ can be arrived at as follows,

$$\begin{aligned}\vec{X}^j &= \mathbf{S}^j \cdot \vec{X} \\ \vec{X} &\in \mathbb{R}^D; \mathbf{S}^j \in \{0,1\}^{U \times D}\end{aligned}\tag{5.7}$$

where $\mathbf{S}^j$ is a sparse selection matrix. In this work, we restrict $\mathbf{S}^j$ to have exactly one 1 in every row and column in a random location. This amounts to selecting a random subset of electrodes as input $\vec{X}^j$ to j^{th} base learner. The number of such unique matrices is given by $\binom{U}{D}$. Thus, random channel set selection is done during training and hyperparameter optimization phase. The same set of random channels set are used during the testing phase.

Finally, the output $\vec{o}^j$ from each of the M base learners needs to be combined to yield a final output $\vec{o}$.

$$\begin{aligned}\vec{o} &= f(\vec{o}^1, \vec{o}^2, \dots, \vec{o}^M) \\ \vec{o}, \vec{o}^j &\in \mathbb{R}^C\end{aligned}\tag{5.8}$$

In this work, we have explored two intuitive ways to arrive at the final output as outlined below.

5.3.2.1 Method 1

We simply sum the outputs from all M base learners to obtain $\vec{o}$ as shown below,

$$\vec{o} = \sum_{j=1}^M \vec{o}^j \quad (5.9)$$

Final decoded value corresponds to the output class of $\vec{o}$ having the highest value among all C classes.

5.3.2.2 Method 2

An alternative method is to perform voting among the decision taken by each of the M base learners,

$$\vec{o} = \sum_{j=1}^M \mathbb{V}(\vec{o}^j) \quad (5.10)$$

where $\mathbb{V} : \mathbb{R}^C \rightarrow \{0, 1\}^C$ is a function whose output is 1 for the output class of $\vec{o}^j$ having the maximum value. Final decoded value in this case corresponds to the output class of $\vec{o}$ having maximum score among C classes.

5.4 Results

We present a decoding comparison of two variants of fixed models - a) a single classifier based model trained on the entire feature set and b) a sparse ensemble classification model - in order to highlight the effectiveness of our proposed approach. These fixed models are trained on the initial 3 days as shown in Figure 5.4. Linear discriminant analysis (LDA), support vector machine (SVM), extreme

learning machine (ELM) and multilayer perceptron (MLP) have been used to compare the two variants of fixed models. In addition, we have also added fixed versions of a widely used ensemble algorithm - random forest (RF) [204, 205], and state-of-the-art recurrent neural network - LSTM [78] to the comparison. Furthermore, we also contrast the decoding results of fixed models against a daily retrained model.

LDA and SVM have been quite popularly employed in both invasive and non-invasive forms of the brain machine interfaces [112, 124]. ELM has been shown to exhibit comparable results to the state-of-the-art algorithms such as SVM, along with the benefit of being trained quickly on a small amount of training data [143, 169]. MLPs and its variants such as LSTM have been enjoying popular support with the dawn of the deep learning revolution and have recently shown impressive results in iBMIs [49, 78, 146]. The aforementioned reasons drove us towards choosing these algorithms as a means of providing a comprehensive benchmark. For a daily retrained model, simple linear models have been shown to perform reasonably well with small amount of training data [123, 195] and hence we have used LDA to obtain a daily retrained model.

5.4.1 Choice of Hyperparameters

Sparse ensemble decoder training involves optimal tuning of number of inputs to each base learner (U) and the number of base learners (M). In general, we expect the decoding accuracy to increase as we increase M at the cost of increased computation. However, the relation with U is not so obvious and we explore that first. We employ the well known k -fold ($k = 5$) cross-validation method for this purpose [206].

U is set as per Equation 5.6 and the decoding accuracy is observed by sweeping q as shown in Figure 5.6 for NHP E and B's data respectively. This is done first while sweeping M from 10 to 25 in steps of 5. However, it is found that regardless of the value of M , decoding accuracy peaks for $q = 2$ for NHP E and $q = 5$ for NHP F across all algorithms. The corresponding optimal value of U comes out to be 14 ($q = 2$) and 35 ($q = 5$) for NHP E and NHP F respectively.

Similarly, the optimal number of base learners, M , is next determined by sweeping its value (in a much finer grid than in Figure 5.6) as shown in Figure 5.7, while

FIGURE 5.6: k-fold cross-validation to obtain optimal value of number of inputs (U) to each base learner as per Equation (5.6) for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h). The results are obtained for different number of base learners (M) swept from 10 to 25 across different classifiers. The optimal value of U comes out to be 14 ($q = 2$) and 35 ($q = 5$) across classifiers for NHP E and NHP F respectively.

FIGURE 5.7: k-fold cross-validation to obtain optimal value of number of base learners (M) for NHP E and NHP F across different classifiers for the value of U obtained in Figure 5.6. The optimal value of M comes out to be around 13 for both NHPs.

choosing optimal q found earlier. The optimal value is found to be around $M = 13$ across algorithms for both cases which is close to the value of $M = 15$ obtained in the coarse sweep of M in earlier experiments.

5.4.2 Base Learners training

LDA does not require any hyperparameter tuning. ELM requires only the tuning of number of hidden layer neurons and k -fold cross-validation was used to arrive at an optimal value. No activation function was employed at the hidden nodes in ELM. We used SVM with a Gaussian kernel. Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) function *bayesopt* was used over 30 iterations to arrive at the optimal hyperparameters for SVM and RF models. We have used a two hidden layer MLP and single hidden layer LSTM with bayesian optimization employed for hyperparameter tuning [207]. *Hyperopt* [196] library was used to tune the following hyperparameters listed in Table. 6.2 over 200 evaluations.

TABLE 5.1: Hyperparameter tuning for different models

Hyperparameter	Value Range	Optimal value				
		MLP	LSTM	RF	SVM	ELM
No. of hidden units (Layer 1)	{75, 100, ..., 200}	150(150)	100(175)	-	-	90(90)
No. of hidden units (Layer 2)	{75, 100, ..., 200}	150(175)	-	-	-	-
No. of epochs	{2, 3, ..., 10}	8(9)	4(9)	-	-	-
Batch size	{32, 64, 128, ..., 512}	256(256)	64(256)	-	-	-
Dropout rate	{0, 0.1, ..., 0.5}	0.2(0.1)	0.1(0)	-	-	-
Optimizer	adam, rmsprop, sgd	adam(adam)	sgd(rmsprop)	-	-	-
Activation function	sigmoid, tanh, relu	relu(tanh)	-	-	-	-
No. of time-steps	{1, 2, 3, 4}	-	2(3)	-	-	-
Learner type	Bag, AdaBoostM2, RUSBoost	-	-	Bag(Bag)	-	-
No. of learning cycles	[10, 500]	-	-	482(489)	-	-
Coding	onevsone, onevsall	-	-	-	onevsall(onevsone)	-
BoxConstraint	[0.001, 1000]	-	-	-	207(456)	-
KernelScale	[0.001, 1000]	-	-	-	104(125)	-

{ } represents a discrete set of choices used for hyperparameter tuning, whereas [] represents log-scaled range of values used for hyperparameter tuning.

TABLE 5.2: Training time and approximate number of parameters for unoptimized code of single classifier fixed models for NHP E

Algorithm	Training Time (in seconds)	No. of parameters (approximate)
LDA	0.15	256
SVM	2438.4	111930
ELM	0.08	360
MLP	2761.3	32700
RF	2284.69	10853194
LSTM	6480.33	92000

Table. 5.2 gives an estimate of the training time involved in training a single classifier model for the different algorithms on NHP E's dataset. *fitcdiscr*, *fitcecoc*, *fitcensemble* functions defined in Matlab R19a were used for training LDA, SVM and RF models respectively. ELM was implemented in Matlab R19a. MLP and LSTM were implemented in *Python3* on a Jupyter notebook using Keras library [208]. Simulations were run on an Intel Core i5 powered laptop to report the above timings. *timeit* library was used in python to compute training times. One can observe that for our dataset, complex algorithms - SVM, MLP, RF, LSTM take ≈ 4 orders of magnitude more time to train compared to single shot learning algorithms - LDA and ELM.

5.4.3 Decoding Results

With the hyperparameters fixed based on initial days' training data, we now evaluate the robustness of decoding over multiple days of neural recording in two versions of test datasets. The first version is the originally recorded test dataset referred to as the *unperturbed* dataset. In the second version, we have dropped the most informative electrode from the test day's data by setting its firing rate to zero as a way of increasing the severity of nonstationarity [49]. This test dataset is referred to as the *perturbed* dataset. Most informative electrode was computed as the electrode having the highest mutual information score between firing rate $\mathbf{X}$ and target $\mathbf{Y}$ [209].

The *unperturbed* and *perturbed* test results for our proposed sparse ensemble method against corresponding single classifier models, RF and LSTM can be seen in Figure 5.8 and Figure 5.9 respectively. Table. 5.3, 5.4 summarize the improvements offered by sparse ensemble methods.

TABLE 5.3: Improvement in decoding accuracy (%) for unperturbed (a) NHP E and (b) NHP F datasets afforded by Sparse Ensemble Models

Sparse Ensemble Models	Corresponding Single Classifier Models		Random Forest (RF)		Long Short Term Memory (LSTM)	
	Max	Average	Max	Average	Max	Average
SE_LDA	20.82	11.67 ± 7.77	7.26	3.35 ± 3.07	13.94	7.89 ± 3.86
SE_SVM	12.92	7.51 ± 3.37	8.48	4.34 ± 3.18	14.76	8.88 ± 3.93
SE_ELM	18.66	11.83 ± 6.74	8.78	5.19 ± 2.49	18.72	9.74 ± 6.04
SE_MLP	9.53	4.73 ± 4.16	7.47	2.46 ± 4.10	14.84	7.01 ± 7.84

(a)

Sparse Ensemble Models	Corresponding Single Classifier Models		Random Forest (RF)		Long Short Term Memory (LSTM)	
	Max	Average	Max	Average	Max	Average
SE_LDA	7.15	3.16 ± 2.66	7.77	2.06 ± 4.15	15.43	6.09 ± 6.03
SE_SVM	8.65	3.03 ± 2.79	12.06	3.84 ± 4.90	19.72	7.88 ± 6.38
SE_ELM	6.73	3.13 ± 1.73	15.08	4.84 ± 5.65	22.73	8.52 ± 7.45
SE_MLP	8.71	3.83 ± 3.86	11.57	4.22 ± 5.15	19.23	8.25 ± 7.16

(b)

FIGURE 5.8: *Unperturbed* decoder results across days for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h) have been presented. Classification models and their Sparse Ensemble (SE) versions have been compared alongside RF, LSTM and daily retrained-LDA models in the above plots.

FIGURE 5.9: *Perturbed* decoder results across days for NHP E - (a),(b),(c),(d) and NHP F - (e),(f),(g),(h) while dropping the most informative electrode on the test day are presented. Fixed classification models and their Sparse Ensemble (SE) versions have been compared alongside RF, LSTM and daily retrained-LDA models in the above plots.

TABLE 5.4: Improvement in decoding accuracy (%) for perturbed (a) NHP E and (b) NHP F datasets afforded by Sparse Ensemble Models

Sparse Ensemble Models	Corresponding Single Classifier Models		Random Forest (RF)		Long Short Term Memory (LSTM)	
	Max	Average	Max	Average	Max	Average
SE.LDA	24.16	12.33 ± 6.60	6.71	4.97 ± 1.57	13.29	7.66 ± 4.17
SE.SVM	11.44	7.11 ± 2.89	8.62	6.27 ± 2.49	14.28	8.96 ± 3.35
SE.ELM	21.66	9.21 ± 8.64	8.85	7.14 ± 2.64	15.42	9.82 ± 4.84
SE.MLP	8.82	3.68 ± 3.76	8.83	3.62 ± 4.89	14.30	6.30 ± 7.17

(a)

Sparse Ensemble Models	Corresponding Single Classifier Models		Random Forest (RF)		Long Short Term Memory (LSTM)	
	Max	Average	Max	Average	Max	Average
SE.LDA	13.76	8.27 ± 4.28	14.70	4.38 ± 7.64	47.71	28.8 ± 16.9
SE.SVM	18.96	13.00 ± 5.48	13.62	2.95 ± 8.32	46.62	27.4 ± 15.1
SE.ELM	7.33	3.73 ± 2.38	12.74	5.36 ± 4.86	45.75	29.8 ± 13.6
SE.MLP	28.42	14.08 ± 7.12	16.97	5.72 ± 6.14	44.86	30.1 ± 13.3

(b)

Statistical comparison of all the single classifier test day results consisting of LDA, SVM, ELM and MLP algorithms was performed against their sparse ensemble versions for a two-tailed Wilcoxon signed-rank test. Significance values of 2.93×10^{-4} , 1.43×10^{-4} , 7.79×10^{-4} and 3.79×10^{-6} are obtained for NHP E-unperturbed, NHP F-unperturbed, NHP E-perturbed and NHP F-perturbed cases respectively. Thus, statistically significant results ($p < 0.01$) are obtained when using sparse ensemble methods over a single classifier.

A baseline daily retrained LDA based classifier outperforms fixed sparse ensemble and single classifier methods in line with similar results presented in [195], across both NHPs in the unperturbed case in Figure 5.8. In Figure 5.9, we present results of a retrained LDA model on held out test trials (Figure 5.4) while dropping the most informative electrode in held out test trials only. As expected, a drastic drop is observed in the daily retrained model. In fact, sparse ensemble versions of LDA, SVM, ELM and MLP outperform a daily retrained model by an average 5.54%,

6.83%, 7.70%, 4.18% in NHP E and 14.21%, 13.81%, 15.20%, 15.55% in NHP F respectively.

5.5 Discussion

5.5.1 Comparison

Authors in [49] have reported steady improvements in offline decoding results over a daily retrained decoder in two NHPs spanning roughly two months. However, we believe approaches in general advocating for employment of complex deep neural networks [49, 78] to counter nonstationarity suffer from mainly three problems. Firstly, they seem to require large amounts of heterogeneous data spanning over a period of upwards of a year or more to perform at their best. Accordingly, we see *LSTM* performing sub-optimally in Figures 5.8 and 5.9. Secondly, the number of hyperparameters and their subsequent tuning for optimal performance is a serious impediment to be overcome in these complicated neural networks [196]. Thirdly, they risk running into latency issues owing to its computational complexity in terms of the number of operations that need to be performed to arrive at the decoded output [78, 210]. Furthermore, other problems such as longer training times, interpretability and ease of use prevail [180].

It has been shown with theoretical support that ensemble based systems tailored to alleviate the source of nonstationarity can lend to more stable results compared to single classifier based systems [211, 212]. Based on this reasoning, we chose to build an ensemble of classifiers, with each classifier being trained on a randomly chosen subset of features with replacement [203]. The intuition behind this approach is that multiple hypotheses corresponding to different subsets of electrodes would be learned by individual classifiers in the ensemble. Thus, with the introduction of appropriate diversity and method of aggregating the outputs, the ensemble would be robust against incoming nonstationarity. On the contrary, a single classifier based system trained on a given training data distribution tends to fail with a differing test data distribution [213].

5.5.2 Ensemble vs Sparse Ensemble

To validate if *sparse* sampling input electrodes to the individual base learners is indeed responsible for robustness against simply forming ensembles with all input features, we present comparison of Sparse Ensemble ELM (*SE_ELM*), Sparse Ensemble MLP (*SE_MLP*) against Ensemble ELM (*Ens_ELM*), Ensemble MLP (*Ens_MLP*) for NHP E in an unperturbed setting in Figure 5.10. We have used summation as a method of combining outputs in both methods with hyperparameter tuning carried out as per the methodology detailed in Section 5.4.1. From Figure 5.10, we can see that sparse ensemble versions of ELM, MLP outperform ensemble ELM, MLP by an average 11.78%, 4.33% highlighting the important role played by the sparse connection of input electrodes to each base learner. The reason why *sparse* works is that it ensures that different base learners learn a different hypothesis with different set of electrodes and hence the resultant sparse ensemble network is more robust compared to an ensemble with all input features, that lacks sufficient diversity. SVM and LDA are deterministic classifiers and hence their ensembles with all input features would yield the same output as that of a single classifier. Hence, SVM and LDA were not included in this analysis.

FIGURE 5.10: Comparing Sparse Ensemble ELM (*SE_ELM*), Sparse Ensemble MLP (*SE_MLP*) against Ensemble ELM (*Ens_ELM*), Ensemble MLP (*Ens_MLP*) with summation as a method of combining outputs in unperturbed NHP E dataset. This figure highlights that introducing sparsity in the inputs to base learners lends robustness to ensemble based learning.

5.5.3 Combining Strategy: Sum vs Voting

As shown in Section 5.3.2.2 voting as a method of combining outputs is also quite popular. However, we have presented detailed results using the summing approach since it consistently outperformed the voting approach as seen in Figure 5.11 across classifiers - LDA, SVM, ELM and MLP.

FIGURE 5.11: Average decoded results across test days for NHPs E and F in (a) and (b) respectively comparing summation and voting methods to arrive at the final ensemble output across different methods.

5.5.4 Custom neural recording tools

Design of custom miniaturized neural recording setups such as [98] involve trade-offs between noise, power and area [58]. The quest for lower power and area consumption in miniaturized chips often drives up the value of input referred noise. For example, the custom recording setup [98] reports an input referred noise of $4.4 \mu V$ *RMS*, whereas commercial front ends such as Ripple [52] report an input referred noise levels of less than $2.1 \mu V$ *RMS*. Thus, we hypothesize that this increased recorded noise in a custom setup for NHP E is the reason for (a) increased sensitivity in selecting an optimal M (Figure 5.7) and (b) need for pre-emphasis through NEO for effective spike detection. Despite this noise impediment in custom recording apparatus, sparse ensemble methods have the potential to yield consistent improvements as seen in the *Results* section.

5.6 Conclusion

In this study, we have presented a novel way to increase the robustness of decoding over a longer term in iBMIs. This is the first ever study reporting consistent improvement against state-of-the-art techniques over a period spanning roughly two months in two NHPs. Our novel and simple approach tackles the root of the nonstationarity problem, i.e. randomly changing firing rates and accounts for these changes by dividing a subset of the inputs randomly among each of the learners in an ensemble. Significant improvements can be seen in both actually recorded and manually perturbed data where the perturbation removes the most informative electrode (Figure 5.8 and Figure 5.9 respectively).

Chapter 6

Online Reinforcement-based Bandit Algorithms for autonomous Intracortical Brain Machine Interfaces

6.1 Introduction

The iBMI algorithmic approaches studied so far in this thesis are based on supervised learning approaches such as LDA, Kalman filter, Extreme Learning Machine, SVM, LSTM etc. The problem with supervised learning paradigm is that it requires a ground truth target label to be available at every instance of time to train an appropriate model. In case of patients who cannot move their limbs to control effector kinematics, getting this ground truth target label requires some workaround, while making some bold assumptions. It involves carefully designed calibration procedures that are designed with the assumption that the ground truth target label is always pointing towards the direction of the eventual goal in the designed trial [48, 78]. These requirements often necessitate the supervision of a neural engineer to ensure smooth operation of the iBMI system.

Furthermore, these calibration procedures take up several minutes of time. This greatly affects the ability of iBMI users, suffering from impairments such as amyotrophic lateral sclerosis (ALS), to remain alert during these prolonged calibration

rouines [214]. Imagine having to calibrate your mobile touch screen every time before using your handset.

FIGURE 6.1: (a) Block level representation of Reinforcement Learning in context of iBMIs. The iBMI controller emits a discrete output action based on inputs - Neural states and rewards. For an Agent (iBMI Decoder) with N input electrodes and C target labels, (b) depicts a single layer neural network that is trained by Banditron algorithm, (c) represents a multi-layer neural network architecture that is employed by RL algorithms - AGREL, HRL and Deep Q-learning. Do note that the multi-layer architecture can also be extended to Banditron with fixed weights from first to the penultimate layer.

To tackle the above problems, we propose applying online reinforcement learning based methods that:

- Learn the neural input to effector movement mapping on incoming samples on a sample-by-sample basis, thereby getting rid of explicit calibration procedures
- Do not require explicit measurement of effector kinematics and learn with a simple scalar reward that is obtained while interacting with the external environment

Briefly put, a reinforcement learning (RL) system involves an RL-agent outputting an *action* with inputs as - *states* and *rewards* at a given time-step - t_i (see Figure 6.1). *States* in case of iBMIs correspond to the input neural feature vector sensed from micro-electrode arrays at t_i , whereas *rewards* are scalar values obtained as a result of *action* taken at the previous time-step - t_{i-1} . Maximizing the score of total reward is the objective of the learning process in RL-based systems.

Promising RL based iBMI implementations have been reported in the past. This work aims to build up and extend on the reported techniques. Popular RL algorithm of the AlphaGo fame [215], Q-learning was reported in [216] wherein rats controlled a prosthetic arm to choose between two targets. The limitation of this approach is that the large neural state-action pair gives rise to the curse of dimensionality problem leading to generalization difficulty. To overcome this problem, [32] proposed applying attention-gated reinforcement learning and its variants [217, 218]. [32] reports improvement over Q-learning involving one NHP doing a center-out cursor task. An important point to bear in mind is that the reported approaches trained the RL-models in batches implying that these methods would still require an explicit block of training similar to the state-of-the-art supervised methods. Alternatively, [219] has presented an iBMI based on Hebbian Reinforcement learning (HRL) algorithm that is trained in an online fashion on a sample-by-sample basis at each time-step. This allows us to get rid of an explicit calibration block and results have been reported in two NHPs performing a two-target discrete task [219]. Furthermore, [219] has introduced a simplified version of a binary scalar reward to train the RL-model and preliminary results have reported biological sources to contain this reward information [220, 221]. However, all the reported approaches use multi-layer neural networks that are prone to the local optimum problem [218]. To overcome the generalization issue, we present application of Banditron [222] - an online prediction algorithm in an iBMI setting.

In summary, we are presenting an alternate online BMI paradigm along the lines of [219], wherein we do not explicitly collect training, validation data (see 6.2(a)). Rather, we propose to train our model sequentially with incoming data on a sample-by-sample basis as seen in Figure 6.2(b). This helps in getting rid of several minutes worth of fatigue-inducing calibration routines. In the experiments reported in this paper, the time required to conduct calibration routines ranges from approximately

FIGURE 6.2: Shows (a) Batch and (b) Online Reinforcement Learning based learning schemes. Batch-trained models learn a model on an explicit set of training data. Typically, supervised learning schemes employed in iBMIs employ batch-based learning schemes. Online reinforcement learning-based models [219] do not employ training data for learning. Instead, they learn during the test phase by virtue of a scalar feedback ($\vec{f}_i$) at every time-step, $t = i$.

5 to 14 minutes. The quantum of calibration time amounts to approximately one-third of the total experiment time in our reported experiments.

In keeping with this line of thinking, each incoming feature sample $\vec{X}_i$ at time-step $t = i$ is used to update the associated weights only once, based on a scalar feedback $\vec{f}_i$. This methodology is consistent across all the reported RL algorithms.

The main contributions of this work include:

- Comparing offline discrete-state decoding performance of Banditron against state-of-the-art RL algorithms - AGREL, HRL and Q-learning across four datasets spanning multiple days of recording
- Investigating the impact of erroneous and sparse nature of reward signal on the above reported RL algorithms
- Comparing the energy requirements of RL algorithms

- Reporting RL algorithms decoding performance against popular supervised classification methods - Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM). Do note that this is not a one-to-one comparison as RL and supervised methods employ altogether different paradigms of learning.

6.2 Materials and Methods

FIGURE 6.3: (a) This cartoon represents the general nature of the behavioural task performed in experiments 5, 6 and 7. In experiments 5 and 6, the nonhuman primate (NHP) is trained to maneuver a joystick-controlled wheelchair whose position updates every 100 ms. The wheelchair starts from a central position in every trial to reach the target over several time-steps, appearing in one of the three square-shaped locations [123]. However, in experiment 7, closed loop brain control was employed and decoded commands from a discrete classifier were used for controlling wheelchair motion. Wheelchair in both scenarios of the brain and joystick control was driven at a frequency of 10 Hz. (b) represents a center-out task corresponding to experiment 8, wherein the NHP was trained to move a joystick-controlled cursor starting from a central position on a computer screen to one of the eight different square-shaped target locations. Cursor position was updated every 100 ms and the NHP reached the target over several time-steps.

6.2.1 Signal Acquisition and Processing

We have used two adult male macaques (*Macaca fascicularis*) to conduct our experiments. We will refer to them as non-human primate (NHP) G and NHP H

respectively. A titanium head post (Crist Instruments, MD, USA) was affixed prior to implantation of microelectrode arrays in both NHPs. In NHP G, 4 microelectrode arrays containing 16 electrodes each [123], and in NHP H, 1 microelectrode array containing 100 electrodes were implanted in the hand/arm region of the left primary motor cortex respectively [182, 223]. Threshold crossings from the sensed neural signals were used as inputs [80, 123].

6.2.2 Behavioural Tasks

We have collected a total of 4 datasets, two from NHP G (experiments 5 and 7) and two from NHP H (experiments 6 and 8) respectively. Pictorial representation of the tasks can be seen in Figure 6.3. Brief description is provided below -

6.2.2.1 Experiment 5

This experiment involved a robotic wheelchair bound NHP G controlling its motion through a three-directional spring-loaded joystick [123]. The experiment consisted of four tasks - a) turning 90° right, b) moving forward by 2 m, c) turning 90° left and d) staying still for 5 seconds (*stop* task). A trainer holding a piece of reward served as a visual target and a trial was considered successful if NHP G managed to reach the trainer within an elapsed time of 15 seconds while the wheelchair was driven at a frequency of 10 Hz.

6.2.2.2 Experiment 6

In this experiment, NHP H was seated in a primate chair facing a computer screen and was trained to manipulate a three-directional joystick similar to experiment 5. This experiment consisted of three tasks, with a target being presented in a pseudo-random manner in one of the following locations – Top, right and left relative to the centre of the screen. The target was in the form of a red square of side 2 cm on a black background. Each trial began with a wheelchair avatar being displayed at the centre of the screen along with the target. A trial was considered successful if NHP H managed to reach and stay in the target area for under a total elapsed

time of 13 seconds. A juice reward was dispensed for every successful trial and the wheelchair avatar was driven at a frequency of 10 Hz.

6.2.2.3 Experiment 7

This experiment is similar to experiment 5 with the difference that the joystick was removed and closed-loop brain control was employed to drive the wheelchair driven at a frequency of 10 Hz. A Linear Discriminant Analysis classifier based decoder was calibrated to achieve closed loop brain control, details of which can be found in [123].

6.2.2.4 Experiment 8

In this experiment (see Figure 6.3 (b)), NHP H was trained to perform the classical centre-out task [48, 130, 224] through joystick control with targets appearing in one of the 8 different locations. A cursor appeared at the centre of the screen at the beginning of every trial and a trial was considered successful if the NHP managed to manipulate the cursor to reach and stay in the target area for 2 seconds under a total elapsed time of 10 seconds. The cursor position was updated every 100 ms via joystick control.

6.2.3 Pre-recorded dataset inputs and outputs

The number of spikes occurring in a backward looking window of time, $T_w = 500$ ms, at each of the input D electrodes constitutes the input feature vector, $\vec{X}_i$, corresponding to the time, $t = i$,

$$\vec{X}_i = \begin{bmatrix} r_1(i) & r_2(i) & \cdots & r_{D-1}(i) & r_D(i) \end{bmatrix} \quad (6.1)$$

where $\vec{X}_i \in \mathbb{R}^D$. $\vec{X}_i$'s are determined in a sliding window fashion with a step-size of $T_s = 100$ ms, for the entire duration of a trial across all the reported experiments. In case of supervised learning algorithms, a one-hot target vector $\vec{Y}_i$ ($\vec{Y}_i \in \mathbb{R}^C$) exists for corresponding $\vec{X}_i$. C stands for the number of discrete classes. In experiments

5, 6 and 7, $C = 4$ corresponding to joystick outputs - *left*, *right*, *forward* and *stop*. $C = 8$ corresponding to the eight different possible target locations (see Figure 6.3(b)) in experiment 8. RL algorithms do not employ $\vec{Y}_i$, instead a binary scalar feedback exists at every time-step, $t = i$. Feedback signal at $t = i$ is given as,

$$f_i = \begin{cases} +1 & c = c' \\ -1 & c \neq c' \end{cases} \quad (6.2)$$

where c and c' are predicted and true labels respective at $t = i - 1$.

6.2.4 Datasets and train-test split

FIGURE 6.4: The first session’s data of every recorded day was used for training *retrained* supervised models across all pre-recorded datasets. First session’s data of the first day of recording was used to build *fixed* supervised models as well as tune hyper-parameters associated with Reinforcement Learning (RL) models. Sessions 2 and 3 of each day are used as test data across all techniques.

We have analyzed four datasets acquired from two NHPs in an offline manner. In experiments 5 and 6, we have used a total of 8 days of recorded data with 3 sessions on each day. In experiments 7 and 8, we have used data recorded on 4 separate days with 3 sessions on each day.

Figure 6.4 shows the manner of train-test data split. Test data is consistent across all algorithms and corresponds to the sessions 2 and 3 on recorded days, across all experiments. Training data corresponds to session 1 of each day for supervised daily *retrained* models, and session 1 of day 1 for supervised *fixed* models. RL algorithms do not employ explicit training data. They are randomly initialized at the beginning of a recorded day and are updated in a purely online manner on test data.

6.3 Reinforcement Learning Algorithms

6.3.1 State-of-the-art

6.3.1.1 AGREL

Authors in [225] have introduced attention-gated reinforcement learning (AGREL) as a three-layer neural network which takes neural state as an input and maps it to corresponding action space. j^{th} hidden layer node's output is given as [32, 225],

$$Y_j = \frac{1}{1 + \exp\left(-\sum_{i=0}^N v_{ij}r_i\right)} \quad (6.3)$$

where N stands for input electrodes, r_i is the spike firing rate appearing at the i^{th} electrode, v_{ij} represents the input layer weights and $r_0 = 1$ to factor in input weight bias.

AGREL uses stochastic softmax rule to choose a winning neuron among C output neurons. Here C stands for the number of possible different actions. k^{th} neuron's output - Z_k and the probability of choosing one among C actions is given as,

$$Z_k = \sum_{j=0}^M w_{jk} Y_j$$

$$\Pr(Z_k = 1) = \frac{\exp\left(\sum_{j=0}^M w_{jk} Y_j\right)}{\sum_{k'=1}^C \exp\left(\sum_{j=0}^M w_{jk'} Y_j\right)} \quad (6.4)$$

where w_{jk} represents the output layer weights, M is the number of hidden nodes.

Instantaneous reward is defined as,

$$\delta = \begin{cases} 1 - Z_c & c = c' \\ -1 & c \neq c' \end{cases} \quad (6.5)$$

where c and c' are predicted and true actions respectively.

In order to expedite learning, AGREL defines an expansive function based on instantaneous reward as follows,

$$f(\delta) = \begin{cases} \frac{\delta}{1-\delta}, & \delta \geq 0 \\ \delta, & \delta < 0 \end{cases} \quad (6.6)$$

Weights are initialized randomly drawn from a standard uniform distribution in the interval $(-1,1)$ [225]. At every iteration of prediction following the forward pass, weights are updated in the backward pass as follows,

$$\Delta w_{jk} = \beta Y_j Z_c f(\delta) \quad (6.7)$$

$$\Delta v_{ij} = \alpha x_i Y_j f(\delta) (1 - Y_j) \sum_{k=1}^C Z_k w'_{kj} \quad (6.8)$$

where α and β are learning rates.

6.3.1.2 HRL

HRL uses a three-layer neural network akin to AGREL albeit with a difference in the way it goes about learning. Output of each hidden node ($OutH_i$) is given as [219],

$$OutH_i = \tanh \left(\left[\begin{array}{c} \vec{X} \quad b_i \end{array} \right] * \mathbf{WH}_i \right)$$

$$\vec{X} \in \mathbb{R}^N; \quad b_i \in \mathbb{R}^1; \quad \mathbf{WH}_i \in \mathbb{R}^{N+1} \quad (6.9)$$

where $\vec{X}$ is the input feature vector, b_i is a bias term and $\mathbf{WH}_i$ stands for input weights.

Action value at the output nodes is given as,

$$AV_j = \tanh \left(\left[S(\overrightarrow{OutH}) \quad b_h \right] * \mathbf{WO}_j \right)$$

$$\overrightarrow{OutH} \in \mathbb{R}^M; \quad b_h \in \mathbb{R}^1; \quad \mathbf{WO}_j \in \mathbb{R}^{M+1} \quad (6.10)$$

where $S(\cdot)$ stands for the sign function, b_h is a bias term and $\mathbf{WO}_j$ is the output weights. The output node with the highest action value among a total of C nodes is chosen to be the final output in this scheme. Feedback f is defined as,

$$f = \begin{cases} +1 & c = c' \\ -1 & c \neq c' \end{cases} \quad (6.11)$$

where c and c' are predicted and true actions respectively.

Initial weights are initialized randomly from a standard normal distribution [219] and are updated at every iteration with learning rates μ_O , μ_H in the following manner with action value output vector, $\overrightarrow{AV} \in \mathbb{R}^C$,

$$\begin{aligned} \Delta \mathbf{WH} &= \mu_H * f \left(\begin{bmatrix} \vec{X} \\ b_i \end{bmatrix} (S(\overrightarrow{OutH}) - \overrightarrow{OutH}) \right) \\ &+ \mu_H(1 - f) \left(\begin{bmatrix} \vec{X} \\ b_i \end{bmatrix} (1 - S(\overrightarrow{OutH}) - \overrightarrow{OutH}) \right) \end{aligned} \quad (6.12)$$

$$\begin{aligned} \Delta \mathbf{WO} &= \mu_O * f * \left(\begin{bmatrix} \overrightarrow{OutH} \\ b_o \end{bmatrix} (S(\overrightarrow{AV}) - \overrightarrow{AV}) \right) \\ &+ \mu_O(1 - f) \left(\begin{bmatrix} \overrightarrow{OutH} \\ b_o \end{bmatrix} (1 - S(\overrightarrow{AV}) - \overrightarrow{AV}) \right) \end{aligned} \quad (6.13)$$

6.3.1.3 Deep Q-learning

It involves a multi-layer neural network that takes the current state, s as an input variable and outputs Q values, $Q(s, c)$ value for each possible action in the action space set $[C] = 1, \dots, c$ [226, 227]. We have used a two hidden layer neural network implementation in the this study. Final output, y , is selected following the ϵ -greedy policy as [226],

$$y = \begin{cases} \arg \max_{c \in C} Q(s, c), & p(1 - \epsilon) \\ \text{random action from } C, & p(\epsilon) \end{cases} \quad (6.14)$$

where ϵ determines the exploration-exploitation tradeoff.

Q-learning involves the following update rule,

$$Q(s, c) \leftarrow Q(s, c) + \alpha \left[r + \gamma \max_{c'} Q(s', c') - Q(s, c) \right] \quad (6.15)$$

where α is the learning rate, r stands for the reward obtained when taking action c in state, s , γ is the discount factor and $\max_{c'} Q(s', c')$ is the maximum possible Q -value in the next state, s' .

Furthermore, the mean squared loss function to be minimized during the update phase is defined as,

$$\text{MSE} = \left(r + \gamma \max_{c'} Q(s', c') - Q(s, c) \right)^2 \quad (6.16)$$

where $r + \gamma \max_{c'} Q(s', c')$ represents the target and $Q(s, c)$ is the prediction.

6.3.2 Proposed Algorithm

6.3.2.1 Banditron

Initially at time $t = 1$, Banditron weight matrix is initialized as $\mathbf{W}^1 = \mathbf{0} \in \mathbb{R}^{C \times D}$, where C represents the number of output classes and D represents the number of input channels [222]. At subsequent time-steps, $t = i$, as the iBMI system receives inputs, $\vec{X}_i \in \mathbb{R}^D$, Banditron utilizes current weight matrix $\mathbf{W}^i$ to arrive at an intermediate output, $\hat{y}_i$,

$$\hat{y}_i = \underset{r \in [C]}{\text{argmax}} \left(\mathbf{W}^i \vec{X}_i \right)_r, \forall r \in [C] \quad (6.17)$$

where $[C] = 1, \dots, c$.

Thereafter, for $\forall r \in [C]$, Banditron defines a probability, $P(r)$ over classes as,

$$P(r) = (1 - \epsilon) \mathbf{1}[r = \hat{y}_i] + \frac{\epsilon}{C} \quad (6.18)$$

The final predicted value, $\tilde{y}_i$, is randomly sampled according to probability $P(r)$. This enables the algorithm to *exploit* current weight matrix ($\mathbf{W}^i$) to choose $\hat{y}_i$ with probability $1 - \epsilon + \frac{\epsilon}{C}$, and *explore* with probability $\frac{\epsilon}{C}$ to randomly choose among all the C output labels. Thus, ϵ determines the exploration-exploitation tradeoff in Banditron. Suggested range for choosing ϵ is given as $\epsilon \in (0, 0.5)$.

Based on the feedback (1 if actual output, y_i equals final predicted value, $\tilde{y}_i$ and 0 otherwise), the weight matrix is updated as,

$$\mathbf{W}^{i+1} = \mathbf{W}^i + \tilde{\mathbf{U}}^i \quad (6.19)$$

where $\tilde{\mathbf{U}}^i \in \mathbb{R}^{C \times D}$ is the update matrix given as,

$$\tilde{\mathbf{U}}_{r,j}^i = \overrightarrow{X_{i,j}} \left(\frac{\mathbf{1}[y_i = \tilde{y}_i] \mathbf{1}[\tilde{y}_i = r]}{P(r)} - \mathbf{1}[\hat{y}_i = r] \right) \quad (6.20)$$

Do note that the true label (y_i) is never revealed to the algorithm, but we indirectly obtain full information when, $[\tilde{y}_i = y_i]$. Equations 6.17-6.20 are adapted from [222].

6.3.2.2 Banditron-RP

Banditron is inspired by simple linear perceptron and hence it follows that it is limited in its capability to learn linearly separable patterns [222]. However, superior performance has been reported by non-linear methods over linear ones in iBMIs [49, 182]. Thus, we have introduced non-linearity in the form of obtaining feature vector - $\vec{f}_i \in \mathbb{R}^M$ as a non-linear random projection (RP) of $\vec{X}_i$ given as,

$$\vec{f}_i = g(\mathbf{W}_{rand} \vec{X}_i) \quad (6.21)$$

where $\mathbf{W}_{rand} \in \mathbb{R}^{D \times M}$ is the fixed random projection weights chosen from a standard uniform distribution in the interval (0,1), $g(\cdot)$ corresponds to the activation function which in our case is sigmoid. Learning proceeds in the same manner as Banditron with $\vec{f}_i$ serving as input instead of $\vec{X}_i$. We henceforth refer to this variant of Banditron with input $\vec{f}_i$ as *Banditron - RP*.

6.4 Results

6.4.1 Summary of models and Hyperparameter Tuning

TABLE 6.1: Summary of training paradigms and algorithms

	Supervised Learning	Reinforcement Learning
Batch training, weight frozen	LDA _{fixed} , LDA _{retrain} , SVM _{fixed} , SVM _{retrain}	—
Only online, update	—	Banditron, AGREL, HRL, Q-learning, Banditron-RP

TABLE 6.2: Hyperparameter tuning - parameter choices

Hyperparameter	Associated Algorithm	Swept Range of Values
Kernel	SVM	Linear, Gaussian, Polynomial
Number of hidden units	AGREL, HRL, Q-learning, Banditron-RP	75,100,150,...,200
Explore-exploit trade-off (ϵ)	Banditron, Banditron-RP Q-learning, AGREL	0.0001,0.001,...,0.1
Discount factor (γ)	Q-learning	0.1,0.2,...,0.9
Learning rates ($\alpha, \beta, \mu_O, \mu_H$)	AGREL, Q-learning, HRL	0.001,0.01,0.1

Table 6.1 gives a summary of algorithms and their respective training paradigms analysed in this paper, whereas Table 6.2 captures the range of values and choices considered for arriving at optimal hyper-parameters for different algorithms. For SVM, we have used bayesian hyper-parameter optimization method *bayesopt* provided in Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) to tune hyper-parameters over 30 iterations in a five-fold cross-validation manner based on training set. LDA does not have any associated hyper-parameters. Hyper-parameters corresponding to the different RL algorithms are tuned on first recorded day's session 1 across all datasets.

6.4.2 Decoding Results - Ideal feedback

We have reported classification accuracy as a measure of performance to compare decoding prowess of different techniques. Experiments 5, 6 and 7 represent a 4 class classification problem - *right*, *forward*, *stop* and *left*. Experiment 8 on the other hand has 8 distinguishable classes corresponding to the 8 different directions shown in Figure 6.3(b).

Test Results - Ideal feedback

FIGURE 6.5: Decoding accuracy across four experiments has been reported for fixed and daily retrained supervised models - LDA, SVM alongside RL algorithms - AGREL, HRL, Q-learning, Banditron and Banditron-RP. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms. Banditron and Banditron-RP significantly outperform the state-of-the-art RL algorithms and *fixed* supervised models.

Results in Figure 6.5 show Banditron's average improvements over AGREL, HRL and Q-learning of $6.01 \pm 2.54\%$, $14.5 \pm 1.83\%$, $4.83 \pm 1.83\%$ in experiment 5, $20.98 \pm 14.95\%$, $14.97 \pm 1.95\%$, $10.07 \pm 11\%$ in experiment 6, $6.7 \pm 6.66\%$, $16.62 \pm 8.23\%$, $9.12 \pm 9.76\%$ in experiment 7 and $7.44 \pm 2.72\%$, $25.04 \pm 4.35\%$, $16.85 \pm 2.53\%$ in experiment 8 respectively. Statistical comparison of Banditron against AGREL, HRL and Q-learning yield p-values of 0.0078, 0.0078, 0.0078 in experiment 5 and 0.0391, 0.0078, 0.0391 in experiment 6 respectively for a Wilcoxon signed-rank test.

This shows that the improvements afforded by Banditron are statistically significant ($p < 0.05$) over AGREL, HRL and Q-learning. We have not reported p-values for experiments 7 and 8 as the number of data points are too small [228]. Furthermore, nonlinear version of Banditron – Banditron-RP yields average improvement of $\approx 6\%$ and 2% over Banditron in experiments 7 and 8 while performing as good as Banditron in experiments 5 and 6 respectively.

Meanwhile, Banditron yields substantial average improvements over fixed models LDA_{fixed} , SVM_{fixed} of $39.4 \pm 18.39\%$, $41.06 \pm 14.09\%$ in experiment 5, $14.89 \pm 11.66\%$, $18.19 \pm 10.9\%$, in experiment 6, $61.02 \pm 10.62\%$, $55.91 \pm 20.31\%$, in experiment 7 and $51.51 \pm 7.35\%$, $50.05 \pm 13.86\%$ experiment 8 respectively. Changes in recording conditions resulting from electrode impedance changes, micro-motion of electrodes etc. lead to non-stationary distribution of input neural data across days [171, 223]. This in turn leads to a drop in performance of fixed models - LDA_{fixed} , SVM_{fixed} due to the underlying shifts in data distribution [49, 223].

In case of comparison to daily retrained models, Banditron yields superior performance with average improvements over $LDA_{retrain}$, $SVM_{retrain}$ of 5.21 ± 5.69 , 8.19 ± 3.6 in experiment 5, 24.36 ± 7.17 , 21.48 ± 4.89 in experiment 7 and 30 ± 19.65 , 27.87 ± 18.16 in experiment 8 respectively. In experiment 6, Banditron performs almost as good as daily retrained classifiers with $\approx 94\%$ level of performance.

6.4.3 Decoding Results - Erroneous and sparse feedback

In Figure 6.5, we have assumed the feedback at every time-step to be ideal corresponding to $+1$ for correct action and -1 otherwise along the lines of paradigm presented in [219, 229–231]. In order to study the impact of quality of feedback signals, we have added additional controls of, a) introducing error in feedback signals and b) making feedback sparse.

Errors are introduced in the feedback signal across 10% and 20% of the total time-steps on each day across all datasets. The erroneous time-steps on each day are chosen via a Matlab R19a (The MathWorks, Inc., Natick, Massachusetts, United States) function *randi* which chooses the fraction of erroneous time-steps from total time-steps following a uniform discrete distribution. Introducing error involves

Test Results - 10% Error in feedback

Test Results - 20% Error in feedback

FIGURE 6.6: Decoding accuracy across four experiments has been reported for RL algorithms - AGREL, HRL, Banditron, Banditron-RP and Q-learning for 10% error in feedback in (a), (b), (c), (d) and 20% error in feedback in (e), (f), (g), (h) respectively. Banditron and Banditron-RP are susceptible to erroneous feedback and cease to outperform state-of-the-art RL algorithms in all but experiment 6. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms.

Test Results - 50% Sparse Feedback

Test Results - 75% Sparse Feedback

FIGURE 6.7: Decoding accuracy across four experiments has been reported for RL algorithms - AGREL, HRL, Banditron, Banditron-RP and Q-learning with holding feedback signal across time-steps thereby introducing sparsity in feedback. (a), (b), (c), (d) depict results for 50% sparsity in feedback and (e), (f), (g), (h) depict results for 75% sparsity in feedback respectively. Shaded regions represent standard deviation of results across 20 iterations of random instantiations of probabilistic algorithms. Banditron and Banditron-RP significantly outperform the state-of-the-art RL algorithms.

changing the feedback signal at a given time-step with value $+1$ to -1 and vice-versa. Furthermore, to study the impact of frequency of availability of feedback signal, we have reduced its occurrence to every second time-step (50% sparsity) and every fourth time-step (75% sparsity). Sparse feedback entails that the RL-decoder model is updated only when feedback is available. Figures 6.6 and 6.7 capture the results associated with erroneous and sparse feedback respectively.

When deliberately introducing error in feedback signal, Banditron ceases to yield the best performance among RL algorithms in all but one experiment, experiment 6. In experiment 5, Q-learning performs the best with $6.96 \pm 3.93\%$, $12.97 \pm 3.48\%$ average improvement over Banditron in 10%, 20% error cases. In experiments 7 and 8, AGREL performs the best with average improvements of $6.88 \pm 10.67\%$, $9.4 \pm 13.03\%$ over Banditron in 10%, 20% error cases respectively in experiment 7 and average improvements of $13.52 \pm 7.48\%$, $19.55 \pm 8.55\%$ in 10%, 20% error cases respectively in experiment 8. However, in experiment 6, Banditron still yields the best response over AGREL, HRL and Q-learning by $12.57 \pm 15.69\%$, $12.12 \pm 1.52\%$, $6.07 \pm 12.13\%$ in 10% error case and $8.87 \pm 14.5\%$, $11.72 \pm 1.1\%$, $6.84 \pm 11.62\%$ in 20% error case respectively. This goes to show that correct feedback is essential to Banditron yielding the best performance.

In case of introducing sparsity, Banditron still yields the overall best performance over other RL algorithms as seen in Figure 6.7. In case of 50%(75%) sparse feedback scenarios, Banditron achieves percentage improvements of at least $\approx 9\%(11\%)$, $11\%(14\%)$, $11\%(13\%)$, $6\%(3\%)$ in experiments 5, 6, 7 and 8 respectively over AGREL, HRL and Q-learning.

6.5 Discussion

6.5.1 Comparison of Computational Complexity

We consider an iBMI system with N inputs and C option discrete control for computation. The number of multiply-and-accumulate operations (MACs) required during prediction for single layer classifiers such as LDA, Banditron can be given as,

TABLE 6.3: Computational Complexity across different models

Algorithms	MACs update	MACs prediction	Memory (kB)
LDA, SVM	-	256	$N \times C \times W_{res} = 0.5$
AGREL	5604	5524	$[M(N+1) + C(M+1)]W_{res}$ $= 10.8$
HRL	11513	5524	$[M(N+1) + C(M+1)]W_{res}$ $= 10.8$
Q-learning	28481	12004	$[M_1(N+1) + M_2(M_1+1)$ $+ C(M_2+1)]W_{res} = 23.3$
Banditron	64	256	$N \times C \times W_{res} = 0.5$
Banditron-RP	160	5440	$[M(N+C)]W_{res} = 10.6$

MAC = multiply and accumulate, Number of input channels, $N = 64$, Number of hidden layer nodes, $M, M_1, M_2 = 80$, Number of outputs, $C = 4$, Weight resolution(W_{res}) = 16 bits.

$$MAC_{Predict_linear} = N \times C \quad (6.22)$$

For single hidden layer neural network based approaches such as HRL, AGREL with M hidden nodes and a bias term in both input-hidden ($\mathbf{WH}_i$) and hidden-output weight ($\mathbf{WO}$) matrices, the number of MACs required during prediction is given as,

$$MAC_{Predict_NN} = M(N+1) + C(M+1) \quad (6.23)$$

For Banditron-RP, we do not consider any bias term and accordingly, the number of MACs during prediction is given as,

$$MAC_{Predict_Banditron-RP} = M \times N + N \times C \quad (6.24)$$

In Q-learning we have used two-hidden layers with M_1 and M_2 nodes respectively and corresponding bias terms. Accordingly, the number of MACs required during prediction step is given as,

$$\begin{aligned} MAC_{Predict_Q} &= M_1(N + 1) + M_2(M_1 + 1) \\ &+ C(M_2 + 1) \end{aligned} \quad (6.25)$$

We have ignored the number of MACs required to implement the activation function for the sake of simplicity.

RL algorithms are recursively updated and MACs expended by single-layer Banditron and multi-layer neural network systems - AGREL, HRL, Q-learning and Banditron-RP can be formulated as below,

$$MAC_{Update_AGREL} = 4 + M(N + 6) \quad (6.26)$$

$$\begin{aligned} MAC_{Update_HRL} &= 5 + 2(M + C) \\ &+ 2(M + 1)(N + C + 2) \end{aligned} \quad (6.27)$$

$$\begin{aligned} MAC_{Update_Q} &= 2M_1(1 + N + M_2) + 1 \\ &+ M_2(2 + N) \end{aligned} \quad (6.28)$$

$$MAC_{Update_Banditron-RP} = 2 \times M \quad (6.29)$$

For a single layer Banditron based network, number of MACs required per update is given as,

$$MAC_{Update_Banditron} = 2 \times N \quad (6.30)$$

Do note that state-of-the-art supervised schemes such as LDA, SVM are trained at the beginning of the day/session and are held fixed without being updated iteratively.

Consider a case of a $N = 64$ input electrode iBMI system driving a 4-option ($C = 4$) discrete control similar to reported experiments. For neural networks based approaches HRL and AGREL, let us assume the number of hidden nodes to be $M = 80$. With these assumptions, we can arrive at the Table. 6.3 comparing the number of MACs and memory requirements across algorithms. Note that we have considered a linear version of SVM in the below reported table and we consider the weights to be 16-bits across all algorithms.

The values populated in Table. 6.3 show that Banditron is at least an order of magnitude less computationally complex in terms of number of MACs and memory required than neural network based RL algorithms - AGREL, HRL and Q-learning. This low complexity feature fits well with the prospect of implementing low power iBMI decoders [140, 159, 182, 232]. Furthermore, the number of MACs required by Banditron are in the same order of magnitude as LDA, SVM.

6.5.2 Comparison to state-of-the-art RL algorithms

It seems rather implausible that a single layer shallow network architecture - Banditron performs better than multi-layer neural network architectures - AGREL, HRL, Q-learning. However, we posit that the reason for Banditron's superior performance is its ability to learn in an online setting [222], inspired from perceptron. Multi-layer neural network based approaches like HRL, AGREL, Q-learning find it hard to converge in an online setting. They typically resort to training in batches over multiple epochs [32]. An interesting avenue of future work would be to train weights of a deep network using an initial batch of data using techniques like Q-learning followed by online update of the last weight layer from there on using

Banditron. This could potentially merge the benefits of both approaches and improve performance over Banditron only training in scenarios with feedback error.

6.5.3 Comparison to state-of-the-art supervised learning paradigm

Supervised learning corresponds to the state-of-the-art decoding methods employed in iBMIs. Thus, we have employed popular techniques - Linear Discriminant Analysis and Support Vector Machines to establish a baseline for the reinforcement learning based results. One must note two stark differences between supervised learning and RL algorithms, a) Supervised algorithms employ an explicit training phase, whereas RL algorithms do not have one and b) Supervised algorithm based models employed in iBMIs typically remain fixed once trained, whereas RL algorithms are constantly updated as they interact with the environment. Thus, this comparison should be seen as a way to establish context in relation to state-of-the-art techniques and not as a one-to-one comparison.

6.5.4 Towards Autonomous iBMIs

The online continuous adaptive nature of the reported paradigm of iBMI operation coupled with possible biological sources of reward signal lead us to envision an autonomous iBMI system wherein no explicit calibration trials are required and the system simply learns from the feedback signal. Thus, this paradigm has the potential to drastically reduce the involvement of the neural engineer in enabling day to day usage of iBMI by patients.

However, this requires the presence of a valid biological reward/error signal. In this study, we are using binary evaluative feedback signal at every time-step along the lines of work [219, 229]. The reason we have opted for this approach is that recent works have shown this kind of signal to be present in regions of the brain such as nucleus accumbens [220], primary motor cortex [221, 230, 231, 233–236]. Apart from these regions, other candidate sources of obtaining this signal include EEG [237], ECoG [238] among others. We certainly feel that more work is needed in this area to fetch a reliable and stable feedback signal.

An alternative to sourcing the reward signal from a biological source is to derive it from the effector’s trajectory as the agent attempts to move it towards a target [218]. In this approach [218], action chosen by the agent is rewarded if it moves towards the target and punished otherwise. This approach, while not useful for day to day autonomous usage by a patient, will still be useful in conducting iBMI experiments where the ground truth target is known from experimental design. We have reported open-loop results and it will be interesting to compare supervised approaches such as ReFIT [48] against RL-based approaches [218] in a closed-loop brain control setting.

6.6 Conclusion and Limitations

We have presented decoding results across different RL algorithms in a binary evaluative feedback scenario. Bandit algorithms – Banditron, Banditron-RP are seen to be the best performing candidate among compared algorithms in an ideal feedback setting. Furthermore, they are seen to be relatively more robust to sparse feedback (see Figure 6.7), i.e. when a feedback signal is not available at all time steps as is likely going to be the case in real BMI scenarios. However, one must point out that Banditron’s superior performance is subject to accuracy of feedback. As seen in Figure 6.6, it is observed that Banditron’s performance is not the best compared to other algorithms when feedback signals are erroneous. This implies a better strategy would be to estimate confidence in the feedback signal first before updating weights using Banditron—this is an aspect for future in-depth study.

As far as the computational complexity is concerned, Banditron has the potential to offer an order of magnitude of reduction in computational complexity compared to state-of-the-art RL algorithms. This has the potential to prolong the battery life of the iBMI system. Finally, results presented in Section 6.4 show that the nature of adaptive learning of this paradigm is naturally robust against non-stationarities.

Chapter 7

Contributions, Limitations and Future Work

7.1 Summary of Contributions

7.1.1 Towards portable, scalable and mobile iBMI systems

To solve the problems of portability, scalability and mobility, we have presented the case to embed intelligence in a fully implantable iBMI. To that end, hardware results of the Spike-input Extreme Learning (SELMA) chip have been presented for offline and online analysis in Chapters 3 and 4 respectively.

The offline study presented in Chapter 3 reports improvements over state-of-the-art Kalman filter in four datasets corresponding to different non-human primates (NHPs). Furthermore, the cost to implement decoding in the implant is studied. It is found that an intelligent iBMI system gives power savings of at least an order of magnitude. In Chapter 4, a closed loop online study is presented to gauge SELMA's performance in relation to native hand movements.

7.1.2 Mitigating non-stationarity

Non-stationarity is a vexing problem facing state-of-the-art iBMI systems. To mitigate this problem, we have presented a novel sparse ensemble machine learning

approach in Chapter 5. This approach has been shown to yield statistically significant improvements over state-of-the-art decoding methods in offline analysis on pre-recorded data sourced from two NHPs. Furthermore, the approach has been shown to be agnostic to the choice of base learning classification algorithm. It has been found to be effective across a swathe of popularly used classification algorithms such as Linear Discriminant Analysis, Support Vector Machines, Extreme Learning Machine and Multi-layer perceptron.

7.1.3 Lending independence to iBMI users

Calibration trials often require the presence of neuro-engineers and caregivers to help in setting up iBMI systems for use. This ends up undermining the purpose of an iBMI system, which is essentially being developed to lend autonomy to the end-users. Reinforcement learning coupled with the prospect of an intrinsically available biological signal, presents an alternate paradigm of minimizing the need for external intervention. To advance research along this paradigm, chapter 6 presents a novel application of an online reward-based algorithm - Banditron. Banditron has been shown to significantly outperform state-of-the-art RL methods in offline analysis on four pre-recorded datasets, sourced from two NHPs. The hardware amenability of Banditron is also studied in context to state-of-the-art algorithms. It has been found that Banditron is at least an order of magnitude less computationally complex than state-of-the-art algorithms.

7.2 Limitations and Future Work

7.2.1 Real-time closed loop experiments

With the exception of Chapter 4, the results reported in this thesis are based on open-loop offline analysis on pre-recorded datasets. The hypothesis put forward in the reported works needs to be validated in a real-time closed loop setting with at least two NHPs, as per the standards followed in the iBMI research community. In particular, iBMI experiments need to be performed in continuous, discrete and hybrid state settings for a comprehensive analysis.

7.2.2 Implementation of a hardware implant

The intelligent iBMI (i^2BMI) concept has been tested with a printed circuit board housing SELMA chip, placed externally in the reported works. The next step is to integrate the SELMA-chip with the analog-front end on a single die, yielding a fully implantable i^2BMI . Furthermore, packaging and biocompatibility requirements should also be addressed before proceeding with implantation of i^2BMI .

7.2.3 Biological Reward Signal

An ideal feedback signal is assumed while performing the RL-based analysis in Chapter 6. There have been preliminary reports of a biological representation of this feedback signal in brain regions such as Nucleus Accumbens and motor cortex. However, obtaining this signal while the iBMI system is in use remains an area of future work.

Furthermore, the reward signal is assumed to be present at every instance of time. However an interesting question for future studies is – Can the iBMI system still perform well if the system gets delayed reward only at the end of a trial (attempted movement)?

7.2.4 On-chip learning

The presented i^2BMI concept has mainly focused on gains afforded by doing on-chip inference. Training is done off-chip typically on dedicated hardware such as PCs. However, to get completely rid of bulky hardware, it is imperative to look at options to do on-chip training. Thus, future versions i^2BMIs need to be designed with the option of on-chip learning.

7.2.5 Closed-loop stimulation

Most of the reported iBMIs employ only visual feedback. However, lately studies [239] have reported improvements in iBMI dexterity by incorporating somatosensory stimulation. The argument is that sense of touch derived from stimulation

enhances skillful control. Thus, i^2BMs need to adapt to a) incorporate circuitry to deliver stimulation pulses and b) extract features reliably in the presence of stimulation artifacts.

List of Author's Publications

Journal Articles

S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, "Towards Intelligent Intracortical BMI (i^2 BMI): Low-Power Neuromorphic Decoders That Outperform Kalman Filters," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1615-1624, Dec. 2019.

S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, "Sparse Ensemble Machine Learning to Improve Robustness of Long-Term Decoding in iBMIs," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 380-389, Feb. 2020.

S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, "Lightweight Reinforcement Algorithms for Autonomous, Scalable, Neuromorphic intracortical Brain Machine Interfaces," in *Neuromorphic Computing and Engineering (in submission)*.

Book Chapter

S. Shaikh and A. Basu, "Intelligent Intracortical Brain-Machine Interfaces (i^2 BMI): Next generation of Scalable Neural Interfaces," Springer Handbook of Biochips: Integrated Circuits and Systems for Biology and Medicine, edited by Mohamad Sawan.

Conference Proceedings

S. Shaikh, C. Yi, A. Basu and R. So, “Cortical motor intention decoding on an analog co-processor with fast training for non-stationary data,” *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Turin, 2017, pp. 1-4.

S. Shaikh, R. So, T. Sibindi, C. Libedinsky and A. Basu, “Real-time Closed Loop Neural Decoding on a Neuromorphic chip,” *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, San Francisco, CA, USA, 2019, pp. 670-673.

S. Shaikh, R. So, C. Libedinsky and A. Basu, “Experimental Comparison of Hardware-Amenable Spike Detection Algorithms for iBMIs,” *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, San Francisco, CA, USA, 2019, pp. 754-757.

S. Shaikh, R. So, C. Libedinsky and A. Basu, “Towards Autonomous Intracortical Brain Machine Interfaces: Applying Bandit Algorithms for Online Reinforcement Learning,” *2020 IEEE International Symposium on Circuits and Systems Conference (ISCAS)*, Sevilla, Spain 2020.

Bibliography

- [1] BRAIN Initiative, 2020. URL <https://braininitiative.nih.gov/>. Accessed on March 7th, 2020. [1](#)
- [2] Human Brain Project, 2020. URL <https://www.humanbrainproject.eu/en/>. Accessed on March 7th, 2020. [1](#)
- [3] World Health Organization, 2020. URL <https://www.who.int/>. Accessed on March 7th, 2020. [1](#)
- [4] Human Genome Project, 2020. URL <https://www.genome.gov/>. Accessed on March 7th, 2020. [1](#)
- [5] Deep Brain Stimulation for Parkinson's Disease, 2020. URL <https://www.singhealth.com.sg/news/medical-news/parkinson-disease-deep-brain-stimulation>. Accessed on March 7th, 2020. [2](#)
- [6] Andres M Lozano and Robert E Gross. Introduction to Deep Brain Stimulation. *Neurosurgical focus*, 42(videosuppl2):Intro, 2017. [2](#)
- [7] Deep Brain Stimulation Devices Market Size, Share Trends Analysis Report, By Product, By Application (Parkinson's Disease, Essential Tremor), By End Use, By Region, And Segment Forecasts, 2019 - 2026, 2020. URL <https://www.fortunebusinessinsights.com/industry-reports/deep-brain-stimulation-dbs-devices-market-100559>. Accessed on March 7th, 2020. [2](#)
- [8] Johannes B Erhardt, Erwin Fuhrer, Oliver G Gruschke, Jochen Leupold, Matthias C Wapler, Jürgen Hennig, Thomas Stieglitz, and Jan G Korvink. Should patients with brain implants undergo MRI? *Journal of Neural Engineering*, 15(4):041002, 2018. [xxiii](#), [3](#)
- [9] E Overstreet and M Hoen. Cochlear implants: Considerations regarding the relationship between cognitive load management and outcome. *The Hearing Review*, 25(7):33–35, 2018. [3](#)
- [10] Cochlear Implant Market Size, Share Trends Analysis Report by Type Of Fitting (Unilateral Implant, Bilateral Implant), by End Use (Adult, Pediatric), by Region, and Segment Forecasts, 2020 - 2027,

2020. URL https://www.researchandmarkets.com/reports/4209733/cochlear-implant-market-size-share-and-trends?utm_source=dynamic&utm_medium=BW&utm_code=dtqp9k&utm_campaign=1330815+-+Global+Cochlear+Implant+Market+Size%2c+Share+%26+Trends+Analysis+Report+2019-2026&utm_exec=chdo54bwd. Accessed on March 7th, 2020. 3
- [11] Paralysis statistics - Reeve Foundation. URL <https://www.christopherreeve.org/living-with-paralysis/stats-about-paralysis>. 4
- [12] Royal Society. *iHuman: Blurring lines between mind and machine*. Royal Society, 2019. 4
- [13] Eric Smalley. The business of brain-computer interfaces. *Nature biotechnology*, 37(9):978, 2019. 4
- [14] How to measure brain activity in people, 2020. URL <https://qbi.uq.edu.au/brain/brain-functions/how-measure-brain-activity-people>. Accessed on April 8th, 2020. xxiii, 5
- [15] Alisa Berger, Fabian Horst, Sophia Müller, Fabian Steinberg, and Michael Doppelmayr. Current State and Future Prospects of EEG and fNIRS in Robot-Assisted Gait Rehabilitation: A Brief Review. *Frontiers in human neuroscience*, 13:172, 2019. 6
- [16] Kristel Knaepen, Andreas Mierau, Eva Swinnen, Helio Fernandez Tellez, Marc Michielsen, Eric Kerckhofs, Dirk Lefeber, and Romain Meeusen. Human-Robot Interaction: Does Robotic Guidance Force Affect Gait-Related Brain Dynamics during Robot-Assisted Treadmill Walking? *PloS one*, 10(10):e0140626, 2015. 6
- [17] Marco Ferrari, Leonardo Mottola, and Valentina Quaresima. Principles, techniques, and limitations of near infrared spectroscopy. *Canadian journal of applied physiology*, 29(4):463–487, 2004. 6
- [18] Stefan P Koch, Sophie Koendgen, Riad Bourayou, Jens Steinbrink, and Hellmuth Obrig. Individual alpha-frequency correlates with amplitude of visual evoked potential and hemodynamic response. *Neuroimage*, 41(2):233–242, 2008. 6
- [19] Jonathan R. Wolpaw, Dennis J. McFarland, Gregory W. Neat, and Catherine A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78(3):252–259, March 1991. ISSN 0013-4694. doi: 10.1016/0013-4694(91)90040-B. URL <https://www.sciencedirect.com/science/article/pii/001346949190040B>. 6

- [20] Jonathan R Wolpaw and Dennis J McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–54, December 2004. ISSN 0027-8424. doi: 10.1073/pnas.0403504101. URL <http://www.ncbi.nlm.nih.gov/pubmed/15585584><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC535103>.
- [21] Ioulietta Lazarou, Spiros Nikolopoulos, Panagiotis C Petrantonakis, Ioannis Kompatsiaris, and Magda Tsolaki. EEG-Based Brain–Computer Interfaces for Communication and Rehabilitation of People with Motor Impairment: A Novel Approach of the 21st Century. *Frontiers in human neuroscience*, 12: 14, 2018. 6
- [22] Rifai Chai, Sai Ho Ling, Gregory P. Hunter, Yvonne Tran, and Hung T. Nguyen. Brain–Computer Interface Classifier for Wheelchair Commands Using Neural Network With Fuzzy Particle Swarm Optimization. *IEEE Journal of Biomedical and Health Informatics*, 18(5):1614–1624, September 2014. ISSN 2168-2194. doi: 10.1109/JBHI.2013.2295006. URL <http://ieeexplore.ieee.org/document/6683068/>. 6
- [23] Jinyi Long, Yuanqing Li, Hongtao Wang, Tianyou Yu, Jiahui Pan, and Feng Li. A Hybrid Brain Computer Interface to Control the Direction and Speed of a Simulated or Real Wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(5):720–729, September 2012. ISSN 1534-4320. doi: 10.1109/TNSRE.2012.2197221. URL <http://ieeexplore.ieee.org/document/6213127/>.
- [24] M. M. Moore. Real-world applications for brain-computer interface technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):162–165, 2003. doi: 10.1109/TNSRE.2003.814433.
- [25] Chun Sing Louis Tsui, John Q. Gan, and Huosheng Hu. A Self-Paced Motor Imagery Based Brain-Computer Interface for Robotic Wheelchair Control. *Clinical EEG and Neuroscience*, 42(4):225–229, October 2011. ISSN 1550-0594. doi: 10.1177/155005941104200407. URL <http://www.ncbi.nlm.nih.gov/pubmed/22208119><http://journals.sagepub.com/doi/10.1177/155005941104200407>.
- [26] Hongtao Wang, Yuanqing Li, Jinyi Long, Tianyou Yu, and Zhenghui Gu. An asynchronous wheelchair control by hybrid EEG–EOG brain–computer interface. *Cognitive Neurodynamics*, 8(5):399–409, October 2014. ISSN 1871-4080. doi: 10.1007/s11571-014-9296-y. URL <http://www.ncbi.nlm.nih.gov/pubmed/25206933><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4155067><http://link.springer.com/10.1007/s11571-014-9296-y>.

- [27] Tobias Kaufmann, Andreas Herweg, and Andrea Kübler. Toward brain-computer interface based wheelchair control utilizing tactually-evoked event-related potentials. *Journal of NeuroEngineering and Rehabilitation*, 11(1):7, January 2014. ISSN 1743-0003. doi: 10.1186/1743-0003-11-7. URL <http://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-11-7>.
- [28] Yuanqing Li, Qiyun Huang, Zhijun Zhang, Tianyou Yu, and Shenghong He. An EEG-/EOG-Based Hybrid Brain-Computer Interface: Application on Controlling an Integrated Wheelchair Robotic Arm System. *Frontiers in Neuroscience*, 13:1243, 2019. 6
- [29] BJ Edelman, J Meng, D Suma, C Zurn, E Nagarajan, BS Baxter, CC Cline, and B He. Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. *Science robotics*, 4(31):eaaw6844, 2019. 6
- [30] Christopher Wirth, Jake Toth, and Mahnaz Arvaneh. “You Have Reached Your Destination”: A Single Trial EEG Classification Study. *Frontiers in Neuroscience*, 14:66, 2020. 6
- [31] Jose L. Contreras-Vidal and Robert G. Grossman. NeuroRex: A clinical neural interface roadmap for EEG-based brain machine interfaces to a lower body robotic exoskeleton. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, volume 2013, pages 1579–1582. IEEE, July 2013. ISBN 978-1-4577-0216-7. doi: 10.1109/EMBC.2013.6609816. URL <http://www.ncbi.nlm.nih.gov/pubmed/24110003><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3801427><http://ieeexplore.ieee.org/document/6609816/>. 6
- [32] Yiwen Wang, Fang Wang, Kai Xu, Qiaosheng Zhang, Shaomin Zhang, and Xiaoxiang Zheng. Neural Control of a Tracking Task via Attention-Gated Reinforcement Learning for Brain-Machine Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(3):458–467, May 2015. ISSN 1534-4320. doi: 10.1109/TNSRE.2014.2341275. 93, 99, 113
- [33] Madiha Tariq, Pavel M Trivailo, and Milan Simic. EEG-based BCI control schemes for lower-limb assistive-robots. *Frontiers in human neuroscience*, 12: 312, 2018. 6
- [34] Seung-Schik Yoo, Ty Fairney, Nan-Kuei Chen, Seh-Eun Choo, Lawrence P Panych, HyunWook Park, Soo-Young Lee, and Ferenc A Jolesz. Brain-computer interface using fMRI: spatial navigation by thoughts. *Neuroreport*, 15(10):1591–1595, 2004. 6
- [35] Bettina Sorger, Tabea Kamp, Nikolaus Weiskopf, Judith Caroline Peters, and Rainer Goebel. When the Brain Takes ‘BOLD’ Steps: Real-Time fMRI Neurofeedback Can Further Enhance the Ability to Gradually Self-regulate Regional Brain Activation. *Neuroscience*, 378:71–88, 2018.

- [36] Amanda Kaas, Rainer Goebel, Giancarlo Valente, and Bettina Sorger. Topographic Somatosensory Imagery for Real-Time fMRI Brain-Computer Interfacing. *Frontiers in Human Neuroscience*, 13, 2019.
- [37] Androu Abdalmalak, Daniel Milej, Lawrence Yip, Ali R Khan, Mamadou Diop, Adrian M Owen, and Keith St Lawrence. Assessing Time-Resolved fNIRS for Brain-Computer Interface Applications of Mental Communication. *Frontiers in Neuroscience*, 14:105, 2020. 6
- [38] Jong-Hwan Lee, Jeongwon Ryu, Ferenc A Jolesz, Zang-Hee Cho, and Seung-Schik Yoo. Brain-machine interface via real-time fMRI: Preliminary study on thought-controlled robotic arm. *Neuroscience letters*, 450(1):1–6, 2009. 6
- [39] Qi Zhao, Chunguang Li, Jiacheng Xu, and Hedian Jin. FNIRS based brain-computer interface to determine whether motion task to achieve the ultimate goal. In *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 136–140. IEEE, 2019. 6
- [40] Bettina Sorger, Joel Reithler, Brigitte Dahmen, and Rainer Goebel. A Real-Time fMRI-Based Spelling Device Immediately Enabling Robust Motor-Independent Communication. *Current Biology*, 22(14):1333–1338, 2012. 7
- [41] Lynn M McCane, Susan M Heckman, Dennis J McFarland, George Townsend, Joseph N Mak, Eric W Sellers, Debra Zeitlin, Laura M Tenteromano, Jonathan R Wolpaw, and Theresa M Vaughan. P300-based Brain-Computer Interface (BCI) Event-Related Potentials (ERPs): People with Amyotrophic Lateral Sclerosis (ALS) vs. Age-Matched Controls. *Clinical Neurophysiology*, 126(11):2124–2131, 2015. 7
- [42] Mariska J Vansteensel, Elmar GM Pels, Martin G Bleichner, Mariana P Branco, Timothy Denison, Zachary V Freudenburg, Peter Gosselaar, Sacha Leinders, Thomas H Ottens, Max A Van Den Boom, et al. Fully Implanted Brain-Computer Interface in a Locked-In Patient with ALS. *New England Journal of Medicine*, 375(21):2060–2066, 2016. 7
- [43] Zhimin Lin, Chi Zhang, Ying Zeng, Li Tong, and Bin Yan. A novel P300 BCI speller based on the Triple RSVP paradigm. *Scientific reports*, 8(1):1–9, 2018. 7
- [44] Chethan Pandarinath, Paul Nuyujukian, Christine H Blabe, Brittany L Sorice, Jad Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife*, 6:e18554, 2017 2017. ISSN 2050-084X. doi: 10.7554/eLife.18554. xxv, 7, 8, 15, 31, 34, 36, 38, 46, 52, 67
- [45] Magdo Bortole, Marco Controzzi, Iolanda Pisotta, and Andrés Úbeda. BMIs for Motor Rehabilitation: Key Concepts and Challenges. pages 235–247. Springer, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-642-38556-8_12. URL http://link.springer.com/10.1007/978-3-642-38556-8_12. 7

- [46] Jennifer L Collinger, Brian Wodlinger, John E Downey, Wei Wang, Elizabeth C Tyler-Kabara, Douglas J Weber, Angus J C McMorland, Meel Veliste, Michael L Boninger, and Andrew B Schwartz. High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet*, 381(9866): 557–64, 2013 2013. ISSN 1474-547X. doi: 10.1016/S0140-6736(12)61816-9. [7](#), [8](#), [15](#), [30](#), [32](#), [34](#), [36](#), [67](#)
- [47] BrainGate. URL <https://www.braingate.org/>. [xxiii](#), [8](#), [32](#)
- [48] Vikash Gilja, Paul Nuyujukian, Cindy A. Chestek, John P. Cunningham, Byron M. Yu, Joline M. Fan, Mark M. Churchland, Matthew T. Kaufman, Jonathan C. Kao, Stephen I. Ryu, and Krishna V. Shenoy. A high-performance neural prosthesis enabled by control algorithm design. *Nature Neuroscience*, 15(12):1752–1757, 2012. ISSN 10976256. doi: 10.1038/nn.3265. [xxv](#), [8](#), [31](#), [36](#), [38](#), [46](#), [52](#), [58](#), [62](#), [91](#), [97](#), [115](#)
- [49] David Sussillo, Sergey D Stavisky, Jonathan C Kao, Stephen I Ryu, and Krishna V. Shenoy. Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 7:1–12, 2016. ISSN 2041-1723. doi: 10.1038/ncomms13749. [xxiv](#), [xxvii](#), [8](#), [11](#), [31](#), [68](#), [69](#), [70](#), [79](#), [83](#), [87](#), [104](#), [107](#)
- [50] Stuart F Cogan, TD Plante, and J Ehrlich. Sputtered iridium oxide films (SIROFs) for low-impedance neural stimulation and recording electrodes. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4153–4156. IEEE, 2004. [8](#)
- [51] Autumn J. Bullard, Samuel R. Nason, Zachary T. Irwin, Chrono S. Nu, Brian Smith, Alex Campean, P. Hunter Peckham, Kevin L. Kilgore, Matthew S. Willsey, Parag G. Patil, and Cynthia A. Chestek. Design and testing of a 96-channel neural interface module for the Networked Neuroprosthesis system. *Bioelectronic Medicine*, 5(1):3, Feb 2019. ISSN 2332-8886. doi: 10.1186/s42234-019-0019-x. [9](#), [34](#)
- [52] Ripple Custom Neuroscience Research Tools, 2020. URL <http://rippleneck.com/>. Accessed on March 7th, 2020. [9](#), [58](#), [73](#), [89](#)
- [53] Ian H Stevenson and Konrad P Kording. How advances in neural recording affect data analysis. *Nature neuroscience*, 14(2):139–42, 2011 2011. ISSN 1546-1726. doi: 10.1038/nn.2731. [9](#), [35](#), [53](#), [66](#)
- [54] David A. Borton, Ming Yin, Juan Aceros, and Arto Nurmikko. An implantable wireless neural interface for recording cortical circuit dynamics in moving primates. *Journal of Neural Engineering*, 10(2), 2013. ISSN 17412560. doi: 10.1088/1741-2560/10/2/026010. [9](#), [34](#), [35](#), [54](#), [55](#)
- [55] Ming Yin, D. A. Borton, J. Aceros, W. R. Patterson, and A. V. Nurmikko. A 100-Channel Hermetically Sealed Implantable Device for Chronic Wireless Neurosensing Applications. *IEEE Transactions on Biomedical Circuits and Systems*, (2):115–128, 2013 . ISSN 1932-4545. doi: 10.1109/TBCAS.2013.2255874.

- [56] Michael Rizk, Chad A. Bossetti, Thomas A. Jochum, Stephen H. Callender, Miguel A L Nicolelis, Dennis A. Turner, and Patrick D. Wolf. A fully implantable 96-channel neural data acquisition system. *Journal of Neural Engineering*, 6(2):026002, 2009. ISSN 17412552. doi: 10.1088/1741-2560/6/2/026002.
- [57] Xilin Liu, Milin Zhang, Basheer Subei, Andrew G Richardson, Timothy H Lucas, and Jan Van der Spiegel. The PennBMBI: A general purpose wireless Brain-Machine-Brain Interface system for unrestrained animals. *IEEE Transactions on Biomedical Circuits and Systems*, 9(2):248–258, 2015. 9
- [58] Reid R Harrison. The Design of Integrated Circuits to Observe Brain Activity. *Proceedings of the IEEE*, 96(7):1203–1216, 2008. ISSN 0018-9219. doi: 10.1109/jproc.2008.922581. 9, 71, 89
- [59] Sohee Kim, Richard A. Normann, R. R. Harrison, and Florianade Solzbacher. Preliminary Study of the Thermal Impact of a Microelectrode Array Implanted in the Brain. *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2986–2989, 2006. xxiii, 18
- [60] Sohee Kim, Prashant Tathireddy, Richard A. Normann, and Florian Solzbacher. In vitro and in vivo study of temperature increases in the brain due to a neural implant. *2007 3rd International IEEE/EMBS Conference on Neural Engineering*, pages 163–166, 2007. 9
- [61] James C Barrese, Juan Aceros, and John P Donoghue. Scanning electron microscopy of chronically implanted intracortical microelectrode arrays in non-human primates. *Journal of Neural Engineering*, 13(2):026003, 2016. 9
- [62] Roy Biran, David C Martin, and Patrick A Tresco. Neuronal cell loss accompanies the brain tissue response to chronically implanted silicon microelectrode arrays. *Experimental neurology*, 195(1):115–126, 2005.
- [63] Bryan J Black, Aswini Kanneganti, Alexandra Joshi-Imre, Rashed Rihani, Bitan Chakraborty, Justin Abbott, Joseph J Pancrazio, and Stuart F Cogan. Chronic recording and electrochemical performance of Utah microelectrode arrays implanted in rat motor cortex. *Journal of neurophysiology*, 120(4):2083–2090, 2018.
- [64] Nicholas F Nolte, Michael B Christensen, Paul D Crane, John L Skousen, and Patrick A Tresco. BBB leakage, astrogliosis, and tissue loss correlate with silicon microelectrode array recording performance. *Biomaterials*, 53:753–762, 2015.
- [65] Vadim S Polikov, Patrick A Tresco, and William M Reichert. Response of brain tissue to chronically implanted neural electrodes. *Journal of neuroscience methods*, 148(1):1–18, 2005.

- [66] Allison M Stiller, Joshua Usoro, Christopher L Frewin, Vindhya R Danda, Melanie Ecker, Alexandra Joshi-Imre, Kate C Musselman, Walter Voit, Romil Modi, Joseph J Pancrazio, et al. Chronic Intracortical Recording and Electrochemical Stability of Thiolene/Acrylate Shape Memory Polymer Electrode Arrays. *Micromachines*, 9(10):500, 2018.
- [67] DH Szarowski, MD Andersen, S Retterer, AJ Spence, M Isaacson, HG Craighead, JN Turner, and W Shain. Brain responses to micro-machined silicon devices. *Brain research*, 983(1-2):23–35, 2003.
- [68] JN Turner, W Shain, DH Szarowski, M Andersen, S Martins, M Isaacson, and Harold Craighead. Cerebral astrocyte response to micromachined silicon implants. *Experimental neurology*, 156(1):33–49, 1999. [9](#)
- [69] Bin Liu, Eric Kim, Anika Meggo, Sachin Gandhi, Hao Luo, Srinivas Kallakuri, Yong Xu, and Jinsheng Zhang. Enhanced biocompatibility of neural probes by integrating microstructures and delivering anti-inflammatory agents via microfluidic channels. *Journal of Neural Engineering*, 14(2):026008, 2017. [9](#)
- [70] Nicolas Y Masse, Beata Jarosiewicz, John D Simeral, Daniel Bacher, Sergey D Stavisky, Sydney S Cash, Erin M Oakley, Etsub Berhanu, Emad Eskandar, Gerhard Friehs, et al. Non-causal spike filtering improves decoding of movement intention for intracortical BCIs. *Journal of neuroscience methods*, 236:58–67, 2014. [10](#)
- [71] Autumn J Bullard, Brianna C Hutchison, Jiseon Lee, Cynthia A Chestek, and Parag G Patil. Estimating risk for future intracranial, fully implanted, modular neuroprosthetic systems: A systematic review of hardware complications in clinical deep brain stimulation and experimental human intracortical arrays. *Neuromodulation: Technology at the Neural Interface*, 2019. [xxiii](#), [10](#)
- [72] Maurizio Gulino, Kim Donghoon, Salvador Pané, Sofia Duque Santos, and Ana Paula Pêgo. Tissue Response to Neural Implants: The Use of Model Systems Toward New Design Solutions of Implantable Microelectrodes. *Frontiers in neuroscience*, 13:689, 2019. [10](#)
- [73] Gürkan Yılmaz and Catherine Dehollain. Packaging of the Implant. In *Wireless Power Transfer and Data Communication for Neural Implants*, pages 81–87. Springer, 2017. [10](#)
- [74] Adam Khalifa, Yasha Karimi, Milutin Stanaćević, and Ralph Etienne-Cummings. Novel integration and packaging concepts of highly miniaturized inductively powered neural implants. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 234–237. IEEE, 2017.
- [75] Caroline K Bjune, Thomas F Marinis, Tirunelveli S Sriram, Jeanne M Brady, James Moran, Philip D Parks, Alik S Widge, Darin D Dougherty, and

- Emad N Eskandar. Packaging Architecture for an Implanted System that Monitors Brain Activity and Applies Therapeutic Stimulation. *Journal of Microelectronics and Electronic Packaging*, 13(2):64–70, 2016. [10](#)
- [76] Beata Jarosiewicz, Anish A. Sarma, Daniel Bacher, Nicolas Masse, J.D. Simeral, Brittany Sorice, Erin M. Oakley, Christine Blabe, Chethan Pandarinath, Vikash Gilja, Sydney S. Cash, Emad Eskandar, Gerhard Friehs, Jaimie Henderson, Krishna Shenoy, John Donoghue, and Leigh Hochberg. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science Translational Medicine*, 7:313ra179–313ra179, 11 2015. [11](#), [67](#), [68](#)
- [77] Peng Zhang, Xuan Ma, Luyao Chen, Jin Zhou, Changyong Wang, Wei Li, and Jiping He. Decoder calibration with ultra small current sample set for intracortical brain-machine interface. *Journal of Neural Engineering*, 15(2), 2018. ISSN 17412552. doi: 10.1088/1741-2552/aaa8a4. [11](#), [68](#)
- [78] Michael A Schwemmer, Nicholas D Skomrock, Per B Sederberg, Jordyn E Ting, Gaurav Sharma, Marcia A Bockbrader, and David A Friedenberg. Meeting brain–computer interface user performance expectations using a deep neural network decoding framework. *Nature medicine*, 24(11):1669, 2018. [xxiv](#), [11](#), [29](#), [30](#), [70](#), [79](#), [87](#), [91](#)
- [79] Beata Jarosiewicz, Anish A Sarma, Jad Saab, Brian Franco, Sydney S Cash, Emad N Eskandar, and Leigh R Hochberg. Retrospectively supervised click decoder calibration for self-calibrating point-and-click brain–computer interfaces. *Journal of Physiology-Paris*, 110(4):382–391, 2016. [11](#), [68](#)
- [80] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. *Neural Comput.*, 16(8):1661–1687, 2004 2004. ISSN 0899-7667. [11](#), [24](#), [27](#), [58](#), [71](#), [73](#), [96](#)
- [81] Yuning Yang and Andrew J. Mason. Hardware Efficient Automatic Thresholding for NEO-Based Neural Spike Detection. *IEEE Transactions on Biomedical Engineering*, 64(4):826–833, 2017. ISSN 15582531. doi: 10.1109/TBME.2016.2580319.
- [82] John D Simeral, Thomas Hosman, Jad Saab, Sharlene N Flesher, Marco Vilela, Brian Franco, Jessica Kelemen, David M Brandman, John G Cianciello, Paymon G Rezaii, et al. Home Use of a Wireless Intracortical Brain-Computer Interface by Individuals With Tetraplegia. *medRxiv*, 2019. [11](#)
- [83] Paul Nuyujukian, Jose Albites Sanabria, Jad Saab, Chethan Pandarinath, Beata Jarosiewicz, Christine H. Blabe, Brian Franco, Stephen T. Mernoff, Emad N. Eskandar, John D. Simeral, Leigh R. Hochberg, Krishna V. Shenoy, and Jaimie M. Henderson. Cortical control of a tablet computer by people with paralysis. In *PloS one*, 2018. [xxiv](#), [11](#), [32](#), [34](#)

- [84] Christine H Blabe, Vikash Gilja, Cindy A Chestek, Krishna V Shenoy, Kim D Anderson, and Jaimie M Henderson. Assessment of brain-machine interfaces from the perspective of people with paralysis. *Journal of Neural Engineering*, 12(4):043002, 2015. [xxiii](#), [12](#)
- [85] Sankaranarayani Rajangam, Po-He Tseng, Allen Yin, Gary Lehew, David Schwarz, Mikhail A Lebedev, and Miguel AL Nicolelis. Wireless Cortical Brain-Machine Interface for Whole-Body Navigation in Primates. *Scientific reports*, 6(1):1–13, 2016. [15](#)
- [86] A. Bolu Ajiboye, Francis R. Willett, Daniel R. Young, William D. Memberg, Brian A. Murphy, Jonathan P. Miller, Benjamin L. Walter, Jennifer A. Sweet, Harry A. Hoyen, Michael W. Keith, P. Hunter Peckham, John D. Simeral, John P. Donoghue, Leigh R. Hochberg, and Robert F. Kirsch. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet*, 389(10081):1821–1830, 2017 2017. ISSN 1474547X. doi: 10.1016/S0140-6736(17)30601-3. [15](#), [30](#), [34](#), [36](#), [67](#)
- [87] Neuron Hand-tuned.svg, Quasar Jarosz CC BY SA 3.0, via Wikimedia Commons, 2020. URL https://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg. Accessed on March 20th, 2020. [xxiii](#), [16](#)
- [88] The Brain, Queensland Brain Institute, 2020. URL <https://qbi.uq.edu.au/brain-basics/brain-physiology>. Accessed on March 20th, 2020. [xxiii](#), [17](#)
- [89] James D Weiland and David J Anderson. Chronic neural stimulation with thin-film, iridium oxide electrodes. *IEEE Transactions on Biomedical Engineering*, 47(7):911–918, 2000. [18](#)
- [90] M Recce. The tetrode: a new technique for multi-unit extracellular recording. In *Soc. Neurosci. Abstr.*, volume 15, page 1250, 1989. [18](#)
- [91] Craig T. Nordhausen, Edwin M. Maynard, and Richard A. Normann. Single unit recording capabilities of a 100 microelectrode array. *Brain Research*, 726(1-2):129–140, July 1996. ISSN 0006-8993. doi: 10.1016/0006-8993(96)00321-6. URL <https://www.sciencedirect.com/science/article/pii/0006899396003216?via%3Dihub>. [18](#)
- [92] Jong-ryul Choi, Seong-Min Kim, Rae-Hyung Ryu, Sung-Phil Kim, and Jeong-woo Sohn. Implantable neural probes for brain-machine interfaces—current developments and future prospects. *Experimental neurobiology*, 27(6):453, 2018. [xxiv](#), [19](#)
- [93] Rikky Muller. *Low Power, Scalable Platforms for Implantable Neural Interfaces*. PhD thesis, UC Berkeley, 2013. [xxiv](#), [19](#)

- [94] James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017. [20](#), [35](#), [53](#)
- [95] Dongjin Seo, Ryan M Neely, Konlin Shen, Utkarsh Singhal, Elad Alon, Jan M Rabaey, Jose M Carmena, and Michel M Maharbiz. Wireless Recording in the Peripheral Nervous System with Ultrasonic Neural Dust. *Neuron*, 91(3):529–539, 2016. [20](#)
- [96] Amir M Sodagar, Kensall D Wise, and Khalil Najafi. A Wireless Implantable Microsystem for Multichannel Neural Recording. *IEEE Transactions on Microwave Theory and Techniques*, 57(10):2565–2573, 2009. [20](#)
- [97] Benoit Gosselin, Amer E Ayoub, Jean-François Roy, Mohamad Sawan, Franco Lepore, Avi Chaudhuri, and Daniel Guitton. A Mixed-Signal Multi-chip Neural Recording Interface With Bandwidth Reduction. *IEEE Transactions on Biomedical Circuits and Systems*, 3(3):129–141, 2009. [20](#)
- [98] Xiaodan Zou, Lei Liu, Jia Hao Cheong, Lei Yao, Peng Li, Ming-Yuan Cheng, Wang Ling Goh, Ramamoorthy Rajkumar, Gavin Stewart Dawe, Kuang-Wei Cheng, et al. A 100-Channel 1-mW Implantable Neural Recording IC. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(10):2584–2596, 2013. [21](#), [71](#), [89](#)
- [99] Hernan Gonzalo Rey, Carlos Pedreira, and Rodrigo Quian Quiroga. Past, present and future of spike sorting techniques. *Brain Research Bulletin*, pages 106–117. ISSN 18732747. doi: 10.1016/j.brainresbull.2015.04.007. [xxiv](#), [22](#), [26](#)
- [100] Sarah Gibson, Jack W Judy, and Dejan Marković. Technology-Aware Algorithm Design for Neural Spike Detection, Feature Extraction, and Dimensionality Reduction. *IEEE transactions on neural systems and rehabilitation engineering*, 18(5):469–478, 2010. ISSN 1558-0210. doi: 10.1109/TNSRE.2010.2051683. [24](#), [25](#), [73](#)
- [101] Sudipta Mukhopadhyay and G. C. Ray. A new interpretation of nonlinear energy operator and its efficacy in spike detection. *IEEE Transactions on Biomedical Engineering*, 45(2):180–187, 1998. ISSN 00189294. doi: 10.1109/10.661266. [24](#), [25](#), [71](#)
- [102] Vahid Shalchyan, Winnie Jensen, and Dario Farina. Spike Detection and Clustering With Unsupervised Wavelet Optimization in Extracellular Neural Recordings. *IEEE Transactions on Biomedical Engineering*, 59(9):2576–2585, 2012. ISSN 00189294. doi: 10.1109/TBME.2012.2204991. [25](#)
- [103] Kyung Hwan Kim and Sung June Kim. A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio. *IEEE Transactions on Biomedical Engineering*, 50(8):999–1011, 2003. doi: 10.1109/TBME.2003.814523. [25](#), [26](#)

- [104] G. C. F. Lee, C. Libedinsky, C. Guan, and R. So. Use of wavelet transform coefficients for spike detection for a Robust Intracortical Brain Machine Interface. In *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 540–543, 2017. doi: 10.1109/NER.2017.8008408. 25
- [105] F. Wood, M.J. Black, C. Vargas-Irwin, M. Fellows, and J.P. Donoghue. On the Variability of Manual Spike Sorting. *IEEE Transactions on Biomedical Engineering*, 51(6):912–918, June 2004. ISSN 0018-9294. doi: 10.1109/TBME.2004.826677. URL <http://www.ncbi.nlm.nih.gov/pubmed/15188858><http://ieeexplore.ieee.org/document/1300782/>. 27
- [106] Kenneth D. Harris, Darrell A. Henze, Jozsef Csicsvari, Hajime Hirase, and György Buzsáki. Accuracy of Tetrode Spike Separation as Determined by Simultaneous Intracellular and Extracellular Measurements. *Journal of Neurophysiology*, 84(1):401–414, July 2000. ISSN 0022-3077. doi: 10.1152/jn.2000.84.1.401. URL <http://www.ncbi.nlm.nih.gov/pubmed/10899214><http://www.physiology.org/doi/10.1152/jn.2000.84.1.401>. 27
- [107] Takashi Takekawa, Yoshikazu Isomura, and Tomoki Fukai. Accurate spike sorting for multi-unit recordings. *European Journal of Neuroscience*, 31(2):263–272, January 2010. ISSN 0953816X. doi: 10.1111/j.1460-9568.2009.07068.x. URL <http://www.ncbi.nlm.nih.gov/pubmed/20074217><http://doi.wiley.com/10.1111/j.1460-9568.2009.07068.x>. 27
- [108] Michael S. Lewicki. Bayesian Modeling and Classification of Neural Signals. In *Maximum Entropy and Bayesian Methods*, pages 235–253. Springer Netherlands, Dordrecht, 1996. doi: 10.1007/978-94-015-8729-7_19. URL http://link.springer.com/10.1007/978-94-015-8729-7_19. 27
- [109] Sivylla E. Paraskevopoulou, Di Wu, Amir Eftekhari, and Timothy G. Constantinou. Hierarchical Adaptive Means (HAM) clustering for hardware-efficient, unsupervised and real-time spike sorting. *Journal of Neuroscience Methods*, 235:145–156, September 2014. ISSN 0165-0270. doi: 10.1016/J.JNEUMETH.2014.07.004. URL <https://www.sciencedirect.com/science/article/pii/S0165027014002489>. 27
- [110] Sonia Todorova, Patrick Sadtler, Aaron Batista, Steven Chase, and Valérie Ventura. To sort or not to sort: The impact of spike-sorting on neural decoding performance. *Journal of Neural Engineering*, 11:056005, 08 2014. doi: 10.1088/1741-2560/11/5/056005. 27
- [111] Eric M Trautmann, Sergey D Stavisky, Subhaneil Lahiri, Katherine C Ames, Matthew T Kaufman, Daniel J O’Shea, Saurabh Vyas, Xulu Sun, Stephen I Ryu, Surya Ganguli, et al. Accurate Estimation of Neural Population Dynamics without Spike Sorting. *Neuron*, 103(2):292–308, 2019. 27
- [112] Jun Dai, Peng Zhang, Hongji Sun, Xin Qiao, Yuwei Zhao, Jinxu Ma, Shaohua Li, Jin Zhou, and Changyong Wang. Reliability of motor and sensory neural decoding by threshold crossings for intracortical brain-machine interface. *Journal of Neural Engineering*, 16(3):036011, 2019. 27, 79

- [113] György Buzsáki, Costas A Anastassiou, and Christof Koch. The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes. *Nature reviews neuroscience*, 13(6):407–420, 2012. [27](#)
- [114] Michael Denker, Sébastien Roux, Henrik Lindén, Markus Diesmann, Alexa Riehle, and Sonja Grün. The Local Field Potential Reflects Surplus Spike Synchrony. *Cerebral Cortex*, 21(12):2681–2695, 2011. [27](#)
- [115] Arjun K Bansal, Wilson Truccolo, Carlos E Vargas-Irwin, and John P Donoghue. Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials. *Journal of neurophysiology*, 107(5):1337–1355, 2012. [27](#)
- [116] Eun Jung Hwang and Richard A Andersen. The utility of multichannel local field potentials for brain–machine interfaces. *Journal of Neural Engineering*, 10(4):046005, 2013. [27](#), [28](#)
- [117] Robert D Flint, Eric W Lindberg, Luke R Jordan, Lee E Miller, and Marc W Slutzky. Accurate decoding of reaching movements from field potentials in the absence of spikes. *Journal of Neural Engineering*, 9(4):046006, 2012. [28](#)
- [118] Maryam M Shانهchi, Amy L Orsborn, Helene G Moorman, Suraj Gowda, Siddharth Dangi, and Jose M Carmena. Rapid control and feedback rates enhance neuroprosthetic control. *Nature communications*, 8(1):1–10, 2017. [28](#)
- [119] Gopal Santhanam, Stephen I. Ryu, Byron M. Yu, Afsheen Afshar, and Krishna V. Shenoy. A high-performance brain-computer interface. *Nature*, 442(7099):195–198, 2006. ISSN 14764687. doi: 10.1038/nature04968. [29](#), [36](#)
- [120] Cynthia A Chestek, Vikash Gilja, Christine H Blabe, Brett L Foster, Krishna V Shenoy, Josef Parvizi, and Jaimie M Henderson. Hand posture classification using electrocorticography signals in the gamma band over human sensorimotor brain areas. *Journal of Neural Engineering*, 10(2):026002, 2013. [29](#)
- [121] V. Aggarwal, S. Acharya, F. Tenore, H. Shin, R. Etienne-Cummings, M. H. Schieber, and N. V. Thakor. Asynchronous Decoding of Dexterous Finger Movements Using M1 Neurons. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(1):3–14, 2008. doi: 10.1109/TNSRE.2007.916289. [29](#), [32](#)
- [122] Ahmed Jorge, Dylan A Royston, Elizabeth C Tyler-Kabara, Michael L Boninger, and Jennifer L Collinger. Classification of Individual Finger Movements Using Intracortical Recordings in Human Motor Cortex. *Neurosurgery*, 2020. [29](#)
- [123] Camilo Libedinsky, Rosa So, Zhiming Xu, Toe K Kyar, Duncun Ho, Clement Lim, Louiza Chan, Yuanwei Chua, Lei Yao, Jia Hao Cheong, et al. Independent Mobility Achieved through a Wireless Brain-Machine Interface. *PloS*

- one, 11(11):e0165773, 2016. [xxvii](#), [xxix](#), [29](#), [34](#), [62](#), [66](#), [71](#), [72](#), [74](#), [79](#), [95](#), [96](#), [97](#)
- [124] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15 3:031005, 2018. [29](#), [79](#)
- [125] Leigh R Hochberg, Mijail D Serruya, Gerhard M Friehs, Jon A Mukand, Maryam Saleh, Abraham H Caplan, Almut Branner, David Chen, Richard D Penn, and John P Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164, 2006. [xxvii](#), [30](#), [69](#)
- [126] John E Downey, Jeffrey M Weiss, Katharina Muelling, Arun Venkatraman, Jean-Sebastien Valois, Martial Hebert, J Andrew Bagnell, Andrew B Schwartz, and Jennifer L Collinger. Blending of brain-machine interface and vision-guided autonomous robotics improves neuroprosthetic arm performance during grasping. *Journal of NeuroEngineering and Rehabilitation*, 13(1):28, 2016. [30](#)
- [127] Apostolos P Georgopoulos, Ronald E Kettner, and Andrew B Schwartz. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, 8(8):2928–2937, 1988. [31](#), [67](#)
- [128] Joseph E O’Doherty, Mikhail Lebedev, Timothy L Hanson, Nathan Fitzsimmons, and Miguel AL Nicolelis. A Brain-Machine Interface Instructed by Direct Intracortical Microstimulation. *Frontiers in Integrative Neuroscience*, 3:20, 2009. [31](#)
- [129] N. Ahmadi, T. G. Constandinou, and C. Bouganis. Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 415–419, 2019. [31](#)
- [130] Benyamin Allahgholizadeh Haghi, Spencer Kellis, Sahil Shah, Maitreyi Ashok, Luke Bashford, Daniel Kramer, Brian Lee, Charles Liu, Richard Andersen, and Azita Emami. Deep Multi-State Dynamic Recurrent Neural Networks Operating on Wavelet Based Neural Features for Robust Brain Machine Interfaces. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14487–14498. Curran Associates, Inc., 2019. [31](#), [97](#)
- [131] Beata Jarosiewicz, Anish A. Sarma, Daniel Bacher, Nicolas Y. Masse, John D. Simeral, Brittany L Sorice, Erin M. Oakley, Christine H. Blabe, Chethan Pandarinath, Vikash Gilja, Sydney S. Cash, Emad N. Eskandar, Gerhard Friehs, Jaimie M. Henderson, Krishna V. Shenoy, John P. Donoghue, and Leigh R. Hochberg. Virtual typing by people with tetraplegia using a

- self-calibrating intracortical brain-computer interface. *Science Translational Medicine*, 7:313ra179–313ra179, 2015. [31](#)
- [132] Dennis J Mcfarland and Jonathan R Wolpaw. Brain–Computer Interfaces for the Operation of Robotic and Prosthetic Devices. In *Advances in computers*, volume 79, pages 169–187. Elsevier, 2010. [32](#)
- [133] Christine H. Blabe, Vikash Gilja, Cindy A. Chestek, Krishna V. Shenoy, Keith D. Anderson, and Jaimie M. Henderson. Assessment of brain-machine interfaces from the perspective of people with paralysis. *Journal of Neural Engineering*, 12 4:043002, 2015. [34](#)
- [134] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher. Thermal Impact of an Active 3-D Microelectrode Array Implanted in the Brain. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4):493–501, 2007. doi: 10.1109/TNSRE.2007.908429. [35](#)
- [135] N. K. Jha S. Mohammed and N. Verma. Enabling Advanced Inference on Sensor Nodes Through Direct Use of Compressively-sensed Signals. *Proceedings of the Design Automation and Test in Europe (DATE)*, 2012. [35](#)
- [136] Arindam Basu, Chen Yi, and Yao Enyi. Big data management in neural implants: The neuromorphic approach. In *Emerging Technology and Architecture for Big-Data Analytics*, pages 293–311. Springer, 2017. [35](#)
- [137] E. Yao, Y. Chen, and A. Basu. A 0.7 V, 40 nW Compact, Current-Mode Neural Spike Detector in 65 nm CMOS. *IEEE Transactions on Biomedical Circuits and Systems*, 10(2):309–318, 2016. doi: 10.1109/TBCAS.2015.2432834. [35](#), [55](#)
- [138] Julie Dethier and Paul Nuyujukian. Design and validation of a real-time spiking-neural-network decoder for brain machine interfaces. *Journal of Neural Engineering*, 10(3):036008, 2013 . [xxv](#), [35](#), [36](#), [38](#), [40](#), [46](#), [52](#), [57](#), [58](#)
- [139] Benjamin I Rapoport, Lorenzo Turicchia, Woradorn Wattanapanitch, Thomas J Davidson, and Rahul Sarpeshkar. Efficient Universal Computing Architectures for Decoding Neural Activity. *PLoS One*, 7(9):e42492, 2012. [57](#)
- [140] Fabio Boi, Timoleon Moraitis, Vito De Feo, Francesco Diotalevi, Chiara Bartolozzi, Giacomo Indiveri, and Alessandro Vato. A Bidirectional Brain-Machine Interface Featuring a Neuromorphic Hardware Decoder. *Frontiers in Neuroscience*, 10(December):1–15, 2016 2016. ISSN 1662-453X. doi: 10.3389/fnins.2016.00563. [57](#), [113](#)
- [141] Zhewei Jiang, Chisung Bae, Joonseong Kang, Sang Joon Kim, and Mingoo Seok. Microwatt end-to-End digital neural signal processing systems for motor intention decoding. *Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017*, pages 1008–1013, 2017. doi: 10.23919/DATE.2017.7927138.

- [142] Leo Yi Chen, Enyi Yao, and Arindam Basu. A 128 channel 290 GMACs/W machine learning based co-processor for intention decoding in brain machine interfaces. *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3004–3007, 2015.
- [143] Yi Chen, Enyi Yao, and Arindam Basu. A 128-Channel Extreme Learning Machine-Based Neural Decoder for Brain Machine Interfaces. *IEEE Transactions on Biomedical Circuits and Systems*, 10(3):679–692, 2016. ISSN 19324545. doi: 10.1109/TBCAS.2015.2483618. [xxv](#), [36](#), [42](#), [43](#), [45](#), [51](#), [52](#), [55](#), [57](#), [66](#), [79](#)
- [144] Sung-Phil Kim, John D. Simeral, Leigh R. Hochberg, John P. Donoghue, Gerhard M. Friehs, and Michael J. Black. Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19: 193–203, 2011. [36](#)
- [145] Joshua I. Glaser, Matthew G. Perich, Pavan Ramkumar, Lee E. Miller, and Konrad Paul Kording. Population coding of conditional probability distributions in dorsal premotor cortex. In *Nature Communications*, 2018. [xxv](#), [37](#)
- [146] Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020. [37](#), [45](#), [79](#)
- [147] Joseph E. O’Doherty, Mariana M. B. Cardoso, Joseph G. Makin, and Philip N. Sabes. Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology [Dataset], 2017. URL <https://zenodo.org/record/583331>. [37](#)
- [148] Joseph G. Makin, Joseph E. O’Doherty, Mariana Cardoso, and Philip N. Sabes. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *Journal of Neural Engineering*, 15 2: 026010, 2017. [37](#), [52](#)
- [149] Sung-Phil Kim, John D. Simeral, Leigh R. Hochberg, John P. Donoghue, and Michael J Black. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *Journal of Neural Engineering*, 5 4:455–76, 2008. [xxv](#), [38](#), [46](#), [52](#)
- [150] Nur Ahmadi, Timothy G. Constandinou, and Christos-Savvas Bouganis. Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network. *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 415–419, 2019. [xxv](#), [38](#), [46](#), [52](#)
- [151] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1):35–45, 1960. ISSN 1528901X. [38](#)

- [152] W Wu, M. J. Black, Y. Gao, M. Serruya, A. Shaikhouni, J. P. Donoghue, and Elie Bienenstock. Neural Decoding of Cursor Motion Using a Kalman Filter. In *Advances in Neural Information Processing Systems 15*. 38, 39, 45
- [153] Zheng Sheng Li, Joseph E. O'Doherty, Timothy L. Hanson, Mikhail A. Lebedev, C. S. Henriquez, and Miguel A. L. Nicolelis. Unscented Kalman Filter for Brain-Machine Interfaces. In *PloS one*, 2009. 38
- [154] Simin Li, Jie Li, and Zhensheng Li. An Improved Unscented Kalman Filter Based Decoder for Cortical Brain-Machine Interfaces. In *Frontiers in neuroscience*, 2016. 38
- [155] Wasim Q. Malik, Wilson Truccolo, Emery N. Brown, and Leigh R. Hochberg. Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(1):25–34, 2011. ISSN 15344320. doi: 10.1109/TNSRE.2010.2092443. 38, 39, 40, 45
- [156] David Maurice Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. Direct Cortical Control of 3D Neuroprosthetic Devices. *Science*, 296 5574: 1829–32, 2002. 38
- [157] Mijail D. Serruya, Nicholas G. Hatsopoulos, Liam Paninski, Matthew R Fellows, and John P Donoghue. Instant neural control of a movement signal. *Nature*, (6877):141–2. ISSN 0028-0836. doi: 10.1038/416141a. 38
- [158] C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10): 1629–1636, Oct 1990. ISSN 0018-9219. doi: 10.1109/5.58356. 40
- [159] Arindam Basu, Jyotibdha Acharya, T. Karnik, and et. al. Low-Power, Adaptive Neuromorphic Systems: Recent Progress and Future Directions. *IEEE Journal on Emerging Topics in Circuits and Systems*, 8(1):6–27, 2018. 40, 113
- [160] Chris Eliasmith and Charles Hal Anderson. Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems. *IEEE Transactions on Neural Networks*, 15:528–529, 2002. 40
- [161] Chris Eliasmith, Terrence C. Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A Large-Scale Model of the Functioning Brain. *Science*, 338 6111:1202–5, 2012. 40
- [162] Julie Dethier, Paul Nuyujukian, Chris Eliasmith, Terrence Stewart, Shauki Elasaad, Krishna V Shenoy, and Kwabena A Boahen. A Brain-Machine Interface Operating with a Real-Time Spiking Neural Network Control Algorithm. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL <https://proceedings.neurips.cc/paper/2011/file/25df35de87aa441b88f22a6c2a830a17-Paper.pdf>. 40

- [163] Guang Bin Huang, Qin Yu Zhu, and Chee Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006. ISSN 09252312. doi: 10.1016/j.neucom.2005.12.126. [41](#), [62](#)
- [164] G. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012. doi: 10.1109/TSMCB.2011.2168604. [41](#), [42](#)
- [165] Yi Chen, Zheng Wang, Aakash Patil, and Arindam Basu. A 2.86-TOPS/W Current Mirror Cross-Bar-Based Machine-Learning and Physical Unclonable Function Engine For Internet-of-Things Applications. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66:2240–2252, 2019. [xxv](#), [42](#), [43](#)
- [166] Zheng Wang, Yi Chen, Aakash Patil, Jayasanker Jayabalan, Xueyong Zhang, Chip Hong Chang, and Arindam Basu. Current Mirror Array: A Novel Circuit Topology for Combining Physical Unclonable Function and Machine Learning. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65: 1314–1326, 2018. [42](#)
- [167] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981. [42](#)
- [168] A Fagg and G. W. Ojakangas and Leonard E. Miller and Nicholas G. Hatsopoulos. Kinetic Trajectory Decoding Using Motor Cortical Ensembles. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17: 487–496, 2009. [44](#), [45](#)
- [169] Shoeb Shaikh, Chen Yi, Arindam Basu, and Rosa So. Cortical motor intention decoding on an analog co-processor with fast training for non-stationary data. In *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, October . ISBN 978-1-5090-5803-7. doi: 10.1109/BIOCAS.2017.8325073. [45](#), [48](#), [79](#)
- [170] Shoeb Shaikh, Rosa Q. So, Tafadzwa Sibindi, Camilo Libedinsky, and Arindam Basu. Real-time Closed Loop Neural Decoding on a Neuromorphic chip. *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 670–673, 2018. [45](#)
- [171] J D Simeral, S-P Kim, M J Black, J P Donoghue, and L R Hochberg. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *Journal of Neural Engineering*, 8(2):025027. [48](#), [67](#), [107](#)
- [172] János Perge, Mark Homer, Wasim Malik, Sydney Cash, Emad Eskandar, Gerhard Friehs, John Donoghue, and Leigh Hochberg. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *Journal of Neural Engineering*, 10:036004, 04 2013. [67](#)

- [173] János Perge, Shaomin Zhang, Wasim Malik, Mark Homer, Sydney Cash, Gerhard Friehs, Emad Eskandar, John Donoghue, and Leigh Hochberg. Reliability of directional information in unsorted spikes and local field potentials recorded in human motor cortex. *Journal of Neural Engineering*, 11:046007, 06 2014. [67](#), [68](#)
- [174] B Wodlinger, J E Downey, E C Tyler-Kabara, A B Schwartz, M L Boninger, and J L Collinger. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *Journal of Neural Engineering*, 12(1):016011, 2015. [51](#)
- [175] Tyson Aflalo, Spencer Kellis, Christian Klaes, Brian Lee, Ying Shi, Kelsie Pejsa, Kathleen Shanfield, Stephanie Hayes-Jackson, Mindy Aisen, Christi Heck, et al. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, 348(6237):906–910, 2015. [48](#), [67](#)
- [176] Manuel De la Guia Solaz and Richard Conway. Razor Based Programmable Truncated Multiply and Accumulate, Energy-Reduction for Efficient Digital Signal Processing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(1):189–193, 2014. [51](#), [55](#)
- [177] Thomas M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, (3):326. ISSN 0367-7508. doi: 10.1109/PGEC.1965.264137. [52](#)
- [178] Tommy Hosman, Marco Vilela, Daniel Milstein, Jessica N. Kelemen, David M. Brandman, Leigh R. Hochberg, and John D. Simeral. BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation. *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 1066–1071, 2018. [52](#)
- [179] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011. [52](#)
- [180] David M. Brandman, Michael C. Burkhardt, Jessica Kelemen, Brian Franco, Matthew T. Harrison, and Leigh R. Hochberg. Robust Closed-Loop Control of a Cursor in a Person with Tetraplegia using Gaussian Process Regression. *Neural Computation*, 30(11):2986–3008, 2018. doi: 10.1162/neco_a_01129. [52](#), [87](#)
- [181] Elon Musk et al. An integrated brain-machine interface platform with thousands of channels. *Journal of medical Internet research*, 21(10):e16194, 2019. [53](#)
- [182] S. Shaikh, R. So, T. Sibindi, C. D. Libedinsky, and A. Basu. Towards Intelligent Intra-cortical BMI (i²BMI): Low-power Neuromorphic Decoders that outperform Kalman Filters. *IEEE Transactions on Biomedical Circuits and*

- Systems*, pages 1–1, 2019. doi: 10.1109/TBCAS.2019.2944486. 57, 96, 104, 113
- [183] David H Brainard. The psychophysics toolbox. *Spatial vision*, 10(4):433–436, 1997. 59
- [184] Dennis Tkach, Jake Reimer, and Nicholas G Hatsopoulos. Observation-based learning for brain–machine interfaces. *Current Opinion in Neurobiology*, 18(6):589–594, 2008. 62
- [185] Patrick Sadtler. *Investigating the neural basis of learning using brain-computer interfaces*. PhD thesis, University of Pittsburgh, 2015. 62
- [186] Ali Rahimi and Benjamin Recht. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009. URL <https://proceedings.neurips.cc/paper/2008/file/0efe32849d230d7f53049ddc4a4b0c60-Paper.pdf>. 66
- [187] Apostolos P Georgopoulos, John F Kalaska, Roberto Caminiti, and Joe T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2(11):1527–1537, 1982. 67
- [188] S. Shaikh, R. So, T. Sibindi, C. Libedinsky, and A. Basu. Real-time closed loop neural decoding on a neuromorphic chip. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 670–673, March 2019. doi: 10.1109/NER.2019.8717122. 67
- [189] Daniel Bacher, Beata Jarosiewicz, Nicolas Y. Masse, Sergey D. Stavisky, John D. Simeral, Katherine Newell, Erin M. Oakley, Sydney S. Cash, Gerhard Friehs, and Leigh R. Hochberg. Neural Point-and-Click Communication by a Person With Incomplete Locked-In Syndrome. *Neurorehabilitation and Neural Repair*, 29(5):462–471, 2015. doi: 10.1177/1545968314554624. PMID: 25385765. 67
- [190] Beata Jarosiewicz, Nicolas Y. Masse, Daniel Bacher, Sydney S. Cash, Emad N. Eskandar, Gerhard Friehs, John P. Donoghue, and Leigh R. Hochberg. Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. *Journal of Neural Engineering*, 10 4:046012, 2013. 67
- [191] Mark L. Homer, Arto V. Nurmikko, John P. Donoghue, and Leigh R. Hochberg. Sensors and Decoding for Intracortical Brain Computer Interfaces. *Annual review of biomedical engineering*, 15:383–405, 2013. 67
- [192] Andrew V Knyazev and Merico E Argentati. Principal angles between subspaces in an A-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040, 2002. xxvii, 69

- [193] Karunesh Ganguly and Jose M. Carmena. Emergence of a stable cortical map for neuroprosthetic control. *PLOS Biology*, 7:1–13, 07 2009. doi: 10.1371/journal.pbio.1000153. URL <https://doi.org/10.1371/journal.pbio.1000153>. 68
- [194] Paul Nuyujukian, Jonathan C. Kao, Joline M. Fan, Sergey D. Stavisky, Stephen I. Ryu, and Krishna V. Shenoy. Performance sustaining intracortical neural prostheses. *Journal of Neural Engineering*, 11 6:066003, 2014. 68
- [195] William Bishop, Cynthia C. Chestek, Vikash Gilja, Paul Nuyujukian, Justin D. Foster, Stephen I. Ryu, Krishna V. Shenoy, and Byron M. Yu. Self-recalibrating classifiers for intracortical brain-computer interfaces. *Journal of Neural Engineering*, 11(2), 2014. ISSN 17412552. doi: 10.1088/1741-2560/11/2/026001. 68, 69, 70, 79, 86
- [196] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015. ISSN 1749-4699. doi: 10.1088/1749-4699/8/1/014008. 70, 81, 87
- [197] *Guidelines for the care and use of mammals in neuroscience and behavioral research*. Washington, D.C. : National Academies Press, c2003. ISBN 0309089034. 71
- [198] Juan Alvaro Gallego, Matthew G. Perich, Stephanie N. Naufel, C. R. Ethier, Sara A. Solla, and Lee E. Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. In *Nature Communications*, volume 9, page 4233, 2018. xxvii, 72
- [199] Kuang-Wei Cheng, Xiaodan Zou, Jia Hao Cheong, Rui-Feng Xue, Zhiming Chen, Lei Yao, Hyouk-Kyu Cha, San Jeow Cheng, Peng Li, Lei Liu, Luis Andia, Chee Keong Ho, Ming-Yuan Cheng, Zhu Duan, Ramamoorthy Rajkumar, Yuanjin Zheng, Wang Ling Goh, Yongxin Guo, Gavin Stewart Dawe, Woo-Tae Park, and Minkyu Je. 100-Channel wireless neural recording system with 54-Mb/s data link and 40%-efficiency power link. *2012 IEEE Asian Solid State Circuits Conference (A-SSCC)*, pages 185–188, 2012. 71
- [200] Cynthia A Chestek, Vikash Gilja, Paul Nuyujukian, Ryan J Kier, Florian Solzbacher, Stephen I Ryu, Reid R Harrison, and Krishna V Shenoy. HermesC: low-power wireless neural recording system for freely moving primates. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(4): 330–338, 2009. 71
- [201] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6:21–45, 2006. 76, 77
- [202] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014. 76, 77

- [203] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8): 832–844, 1998. ISSN 01628828. doi: 10.1109/34.709601. URL <http://ieeexplore.ieee.org/document/709601/>. 76, 87
- [204] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 76, 77, 79
- [205] Manuel Fernández Delgado, Eva Cernadas, Senén Barro, and Dinani Gomes Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014. 79
- [206] Nathalie Japkowicz and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, USA, 2011. ISBN 0521196000, 9780521196000. 79
- [207] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL <https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf>. 81
- [208] François Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015. 82
- [209] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 1991. ISBN 9780471062592. 83
- [210] Nicholas D. Skomrock, Michael A. Schwemmer, Jordyn E. Ting, Hemang R. Trivedi, Gaurav Sharma, Marcia A. Bockbrader, and David A. Friedenberg. A Characterization of Brain-Computer Interface Performance Trade-Offs Using Support Vector Machines and Deep Neural Networks to Decode Movement Intent. *Frontiers in Neuroscience*, 12:763, 2018. 87
- [211] Gregory Ditzler, Gail L. Rosen, and Robi Polikar. Discounted expert weighting for concept drift. *2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*, pages 61–67, 2013. 87
- [212] Gregory Ditzler, Gail Rosen, and Robi Polikar. Domain adaptation bounds for multiple expert systems under concept drift. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 595–601. IEEE, 2014. 87
- [213] Joaquin Quiñero-Candela. *Dataset shift in machine learning*. [electronic resource]. Neural information processing series. Cambridge, Mass. : MIT Press, c2009. ISBN 9780262255103. 87

- [214] David M Brandman, Tommy Hosman, Jad Saab, Michael C Burkhart, Benjamin E Shanahan, John G Cincibello, Anish A Sarma, Daniel J Milstein, Carlos E Vargas-Irwin, Brian Franco, Jessica Kelemen, Christine Blabe, Brian A Murphy, Daniel R Young, Francis R Willett, Chethan Pandarinath, Sergey D Stavisky, Robert F Kirsch, Benjamin L Walter, A Bolu Ajiboye, Sydney S Cash, Emad N Eskandar, Jonathan P Miller, Jennifer A Sweet, Krishna V Shenoy, Jaimie M Henderson, Beata Jarosiewicz, Matthew T Harrison, John D Simeral, and Leigh R Hochberg. Rapid calibration of an intracortical brain–computer interface for people with tetraplegia. *Journal of Neural Engineering*, 15(2):026007, jan 2018. doi: 10.1088/1741-2552/aa9ee7. 92
- [215] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL <http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html>. 93
- [216] J. DiGiovanna *, B. Mahmoudi, J. Fortes, J. C. Principe, and J. C. Sanchez. Coadaptive Brain–Machine Interface via Reinforcement Learning. *IEEE Transactions on Biomedical Engineering*, 56(1):54–64, 2009. doi: 10.1109/TBME.2008.926699. 93
- [217] F. Wang, Y. Wang, K. Xu, H. Li, Y. Liao, Q. Zhang, S. Zhang, X. Zheng, and J. C. Principe. Quantized Attention-Gated Kernel Reinforcement Learning for Brain–Machine Interface Decoding. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):873–886, 2017. doi: 10.1109/TNNLS.2015.2493079. 93
- [218] Xiang Zhang, Student Member, Camilo Libedinsky, Rosa So, Jose C Principe, Yiwen Wang, and Senior Member. Clustering Neural Patterns in Kernel Reinforcement Learning Assists Fast Brain Control in Brain-Machine Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 4320(c):1–10, 2019. doi: 10.1109/TNSRE.2019.2934176. 93, 115
- [219] Eric A Pohlmeyer, Babak Mahmoudi, Shijia Geng, Noeline W Prins, and Justin C Sanchez. Using Reinforcement Learning to Provide Stable Brain-Machine Interface Control Despite Neural Input Reorganization. *PloS one*, 9(1):e87253, 2014. xxix, 93, 94, 101, 107, 114
- [220] Noeline W. Prins, Justin C. Sanchez, and Abhishek Prasad. Feedback for reinforcement learning based brain-machine interfaces using confidence metrics. *Journal of Neural Engineering*, 14(3), 2017. ISSN 17412552. doi: 10.1088/1741-2552/aa6317. 93, 114

- [221] Miri Benyamini, Samuel R. Nason, Cynthia A. Chestek, and Miriam Zacksenhouse. Neural Correlates of error processing during grasping with invasive brain-machine interfaces. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 215–218. IEEE, March . ISBN 978-1-5386-7921-0. doi: 10.1109/NER.2019.8717020. [93](#), [114](#)
- [222] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient Bandit Algorithms for Online Multiclass Prediction. *Proceedings of the 25th International Conference on Machine Learning*, pages 440–447, 2008. [93](#), [103](#), [104](#), [113](#)
- [223] S. Shaikh, R. So, T. Sibindi, C. Libedinsky, and A. Basu. Sparse Ensemble Machine Learning to Improve Robustness of Long-Term Decoding in iBMIs. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28 (2):380–389, Feb 2020. ISSN 1558-0210. doi: 10.1109/TNSRE.2019.2962708. [96](#), [107](#)
- [224] S. Shah, B. Haghi, S. Kellis, L. Bashford, D. Kramer, B. Lee, C. Liu, R. Andersen, and A. Emami. Decoding Kinematics from Human Parietal Cortex using Neural Networks. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 1138–1141, March 2019. doi: 10.1109/NER.2019.8717137. [97](#)
- [225] Pieter R. Roelfsema and Arjen van Ooyen. Attention-Gated Reinforcement Learning of Internal Representations for Classification. *Neural Computation*, 17(10):2176–2214, 2005. doi: 10.1162/0899766054615699. [99](#), [100](#)
- [226] Richard S Sutton and Andrew G Barto. *Reinforcement learning - second edition*. 2018. ISBN 0262193981. doi: 10.1016/S0140-6736(51)92942-X. [102](#)
- [227] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. [102](#)
- [228] Robert V Hogg, Elliot A Tanis, and Dale L Zimmerman. *Probability and Statistical Inference*, volume 993. Macmillan New York, 1977. [107](#)
- [229] Babak Mahmoudi, Eric A Pohlmeier, Noeline W Prins, Shijia Geng, and Justin C Sanchez. Towards autonomous neuroprosthetic control using Hebbian reinforcement learning. *Journal of Neural Engineering*, 10(6):066005, December 2013. ISSN 1741-2560. doi: 10.1088/1741-2560/10/6/066005. [107](#), [114](#)
- [230] A. Tarigoppula, N. Rotella, and J. T. Francis. Properties of a temporal difference reinforcement learning brain machine interface driven by a simulated motor cortex. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3284–3287, Aug 2012. doi: 10.1109/EMBC.2012.6346666. [114](#)

- [231] B. T. Marsh, V. S. A. Tarigoppula, C. Chen, and J. T. Francis. Toward an Autonomous Brain Machine Interface: Integrating Sensorimotor Reward Modulation and Reinforcement Learning. *Journal of Neuroscience*, 35(19): 7374–7387, 2015. ISSN 0270-6474. doi: 10.1523/jneurosci.1802-14.2015. [107](#), [114](#)
- [232] Rahul Sarpeshkar. *Ultra Low Power Bioelectronics*. 2010. ISBN 9780521857277. doi: 10.1017/CBO9780511841446. [113](#)
- [233] Matthew R Roesch and Carl R Olson. Neuronal activity related to anticipated reward in frontal cortex: does it represent value or reflect motivation? *Annals of the New York Academy of Sciences*, 1121(1):431–446, 2007. [114](#)
- [234] Arjun Ramakrishnan, Yoon Woo Byun, Kyle Rand, Christian E Pedersen, Mikhail A Lebedev, and Miguel AL Nicolelis. Cortical neurons multiplex reward-related signals along with sensory and motor information. *Proceedings of the National Academy of Sciences*, 114(24):E4841–E4850, 2017.
- [235] Pavan Ramkumar, Brian Dekleva, Sam Cooler, Lee Miller, and Konrad Kording. Premotor and Motor Cortices Encode Reward. *PloS one*, 11(8):e0160851, 2016.
- [236] David B McNiel, John S Choi, John P Hessburg, and Joseph T Francis. Reward value is encoded in primary somatosensory cortex and can be decoded from neural activity during performance of a psychophysical task. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3064–3067. IEEE, 2016. [114](#)
- [237] Alex Kreilinger, Christa Neuper, and Gernot R Müller-Putz. Error potential detection during continuous movement of an artificial arm controlled by brain–computer interface. *Medical & biological engineering & computing*, 50(3):223–230, 2012. [114](#)
- [238] Tomislav Milekovic, Tonio Ball, Andreas Schulze-Bonhage, Ad Aertsen, and Carsten Mehring. Error-related electrocorticographic activity in humans during continuous movements. *Journal of Neural Engineering*, 9(2):026007, 2012. [114](#)
- [239] Jennifer L Collinger, Robert A Gaunt, and Andrew B Schwartz. Progress towards restoring upper limb movement and sensation through intracortical brain-computer interfaces. *Current Opinion in Biomedical Engineering*, 8: 84–92, 2018. [118](#)