Learning Mappings in Brain Machine Interfaces with Echo State Networks

Yadunandana N. Rao, Sung-Phil Kim, Justin C. Sanchez, Deniz Erdogmus, Jose C. Principe, Jose M. Carmena[2,3]*, Mikhail A. Lebedev*[2,3]*, Miguel A. Nicolelis*[2,3,4]

Computational NeuroEngineering Lab, University of Florida, Gainesville, FL 32611
Dept. of Neurobiology[2], Center for Neuroengineering[3], Department of Biomedical Engineering[4]
Duke University, Durham, NC-27710

ABSTRACT

Brain Machine Interfaces (BMI) utilize linear or non-linear models to map the neural activity to the associated behavior which is typically the 2-D or 3-D hand position of a primate. Linear models are plagued by the massive disparity of the input and output dimensions thereby leading to poor generalization. A solution would be to use non-linear models like the Recurrent Multi-Layer Perceptron (RMLP) that provide parsimonious mapping functions with better generalization. However, this results in a drastic increase in the training complexity, which can be critical for practical use of a BMI. This paper bridges the gap between superior performance per trained weight and model learning complexity. Towards this end, we propose to use Echo State Networks (ESN) to transform the neuronal firing activity into a higher dimensional space and then derive an optimal sparse linear mapping in the transformed space to match the hand position. The sparse mapping is obtained using a weight constrained cost function whose optimal solution is determined using a stochastic gradient algorithm.

1. INTRODUCTION

In their widely acclaimed article, Wessberg *et al* showed that, it is possible to predict the hand position of a primate using cortical neuronal firing activity [1]. Since then, many BMI research groups have adopted diverse linear and nonlinear modeling frameworks [1-8] with the ultimate goal of translating brain activity into prediction of animal behavior. The inputs to these models are usually multidimensional neural recordings collected from selected regions of a monkey's brain. Although the linear models have simple training strategies, the limitation of linear mappings may hinder performance. Furthermore, these simple models have huge number of parameters and if the cost function is unable to properly select inputs, generalization suffers. Typical non-linear models that have been used in BMIs are non-linear Kalman filters [3], Time-Delay Neural Networks (TDNN), Recurrent Multi-Layer Perceptrons (RMLP) [6], particle filters [3] and nonlinear mixture models [7]. Remarkably, these models show only a slight improvement in performance for the experimental paradigms tested, but they can be designed with parsimonious architectures (fewer weights) [9]. However, the improvement in performance is gained at the expense of a significant leap in the computational costs of training these models. The problem is accentuated by the fact that these models may have to be re-trained occasionally to track the changing statistics and environment. This can severely restrict practical application of BMIs. In this paper, we propose a different architecture that has performance similar to that of RMLP, but the training has linear complexity as opposed to the RMLP training algorithm. Further, with respect to the linear model it has fewer weights. This new architecture provides a viable setup to simultaneously extract different output variables (e.g. hand position, velocity, griping force) from the same system state, creating a more flexible BMI system. At the core of the architecture is an Echo State Network (ESN) [10], which will be described briefly in the next section. The outputs of the ESN are then linearly combined by a *sparse matrix*. This will be discussed in section 3 of the paper followed by the experimental section in section 4, where we will design a BMI using this architecture.

2. ECHO STATE NETWORKS: OVERVIEW

Echo State Networks (ESN) first proposed by Jaeger [10] are recurrent networks with simplified learning mechanisms. ESNs exhibit some similarities to the "Liquid State Machines (LSM)" proposed by Maas *et al* [11], which possess universal approximation capabilities in myopic functional spaces. In this section, we will briefly summarize the basic principles of an ESN. ESN uses a "large reservoir" recurrent neural network (RNN) that can produce diversified representations of an input signal, which can then be instantaneously combined in an optimal manner to approximate a desired response. Fig. 1 shows the block diagram of an ESN. A set of input nodes denoted by the vector $\mathbf{u}_n \in \Re^{Mx1}$ is connected to a "reservoir" of N discrete-time recurrent networks by a connection matrix $\mathbf{W}_{in} \in \Re^{MxN}$. At any time instant n, the readout (state output) from the RNN reservoir is a column vector denoted by $\mathbf{x}_n \in \Re^{Nx1}$. In fig.1, we show two outputs and the associated feedback connection matrix, $\mathbf{W}_b \in \Re^{Nx2} = [\mathbf{w}_{b1}, \mathbf{w}_{b2}]$. The desired outputs form a column vector $\mathbf{d}_n = [d_{xn}; d_{yn}]$. The reservoir states are transformed by a static linear mapper that can receive contributions from the input. Each processing element (PE) in the reservoir can be implemented as a leaky integrator and the state output or the readout is given by the difference equation in (1).

$$\mathbf{x}_{n+1} = (1-\mu Ca)\mathbf{x}_n + \mu C(f(\mathbf{W}_{in}^T \mathbf{u}_{n+1} + \mathbf{W}\mathbf{x}_n + \mathbf{W}_b \mathbf{d}_n)) \qquad (1)$$

where, $0<\mu<1$, C is the time constant and a is the decay rate [10]. The point-wise non-linear function $f(.)$ is the standard sigmoid, $f(.) = tanh(.)$. From a signal processing point of view, the reservoir creates a set of bases functions to represent the input,

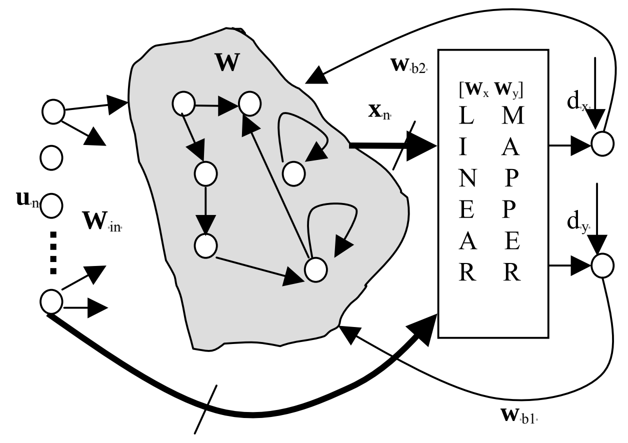

Figure 1. Block diagram of an Echo State Network

while the static mapper finds the optimal projection in this space. There are obvious similarities of this architecture to kernel machines, except that the kernels here are time functions (Hilbert spaces). We will now give the conditions under which an ESN can be "useful," which Jaeger aptly calls as the "Echo State Property." Loosely stated, the Echo State Property says that the current state is uniquely defined by the past values of the inputs and also the desired outputs if there is feedback. A weaker condition for the existence of echo states is to have the spectral radius of the matrix $\mu C\mathbf{W}$ +(1-μCa)**I** less than unity [10]. Further, the matrix **W** is sparse ensuring that the span of the representation space is sufficiently rich to construct the mapping to the desired response.

In most engineering applications of ESN, e.g. system identification and prediction, the optimal linear mapper (see fig.1) is obtained using standard recursive algorithms that minimize MSE. However, when the number of RNNs (N) in the reservoir is very high, the dimensionality of the input to the linear mapper increases. Additionally, if the desired signals are in a lower dimensional space as in our case with the 2-D hand trajectory, increasing N will result in over-fitting. This calls for explicit regularization in the estimation of the optimal linear mapper. Conventionally, adaptive weight decay using cross-validation is used for zeroing the weights that make little or no contribution to the cost. However, having a separate cross-validation set may not be feasible for real world applications that require repeated training. In the next section, we will present an *on-line scheme* that introduces additional constraints on the MSE criterion to result in an optimal and sparse linear mapper.

3. SPARSE-LMS ALGORITHM

Let the linear mapper (assume single output for now) be denoted by the weight vector $\mathbf{w} = \{w_i\}_{i=1\ldots L}$. Consider the cost function in equation (2).

$$J(\mathbf{w}) = E(e_k^2) + \lambda\left[\sum_{i=1}^{L}|w_i|^p - \alpha\right] \tag{2}$$

The first term is the regular Mean-Squared Error (MSE) and the second term is the constraint. The error e_k is the difference between the desired signal d_k and output y_k=$\mathbf{w}^T\mathbf{x}_k$. The constraint fixes the sum of the pth powers of the absolute values of the individual elements w_i of the weight vector **w** to a constant denoted by α. Note that, with p=1, we will be constraining the L_1 norm of the weight vector **w**. The penalty factor is denoted by λ. Instead of fixing the penalty term, we can include λ as an adaptive parameter in the cost function. In order to estimate λ, we modify the cost function in (2) as,

$$J(\mathbf{w},\lambda) = E(e_k^2) + \beta\lambda\left[\sum_{i=1}^{L}|w_i|^p - \alpha\right] - \beta\lambda^2 \tag{3}$$

where, β>0 that keeps the penalty factor λ bounded. The above cost function is sometimes referred to as the *augmented Lagrangian* [12]. The stochastic gradients of the cost are,

$$\begin{aligned}\frac{\partial J(\mathbf{w},\lambda)}{\partial w_i} &\approx -2e_k x_{ki} + \lambda\beta[pw_i^{p-1}sign(w_i^p)]\\ \frac{\partial J(\mathbf{w},\lambda)}{\partial \lambda} &= \beta\left(\sum_{i=1}^{L}|w_i|^p - \alpha\right) - 2\beta\lambda\end{aligned} \tag{4}$$

We will simultaneously minimize and maximize (3) with respect to **w** and λ respectively using the stochastic updates given by,

$$w_i(k+1) = w_i(k) + \eta_w[2e_k x_{ki} - \lambda_k\beta pw_i^{p-1}sign(w_i^p)] \tag{5}$$

$$\lambda_{k+1} = \lambda_k + \eta_\lambda\beta\left(\sum_{i=1}^{L}|w_{ki}|^p - \alpha - 2\lambda_k\right) \tag{6}$$

where, η_w and η_λ are small positive step-sizes, w_{ki} denotes the ith element in the vector $\mathbf{w}_k$ and x_{ki} denotes the ith element in the vector $\mathbf{x}_k$. Allowing, $\lambda_{k+1} = \lambda_k$ as $k \to \infty$, it is easy to see that the sequence $\{\lambda_k\}$ converges to a value λ^* given by,

$$\lambda^* = \frac{1}{2}\left(\sum_{i=1}^{L}|w_i^*|^p - \alpha\right) \tag{7}$$

where, $\mathbf{w}^*$ is the asymptotic weight vector obtained from (5).

Convergence of λ: For stable asymptotic convergence of the penalty sequence to zero, a necessary condition is $0<2\beta\eta_\lambda<<1$.

Proof: Equation (6) can be rewritten as, $\lambda_{k+1} = \lambda_k(1-2\eta_\lambda\beta)$ $+\eta_\lambda\beta\left(\sum_{i=1}^{L}|w_{ki}|^p - \alpha_k\right)$. If, $0<2\beta\eta_\lambda<<1$ the equation becomes $\lambda_{k+1} = \lambda_k + \eta_\lambda\beta c_k$, where c_k is the expression for the constraint. By letting $\lambda_{k+1} = \lambda_k$, as $k\to\infty$, it is obvious that $c_k\to 0$ and consequently, the Lagrangian λ^* converges to zero. Proofs of convergence of **w** are fairly involved and are beyond the scope of this paper. It will suffice to say that the step-size parameter η_w plays a crucial role in the tradeoff between the speed of convergence and misadjustement in the final estimate. A robust update for **w** can be obtained by including a normalization term (similar to the NLMS algorithm) as shown in (8). Note that the normalization is just done on the gradient of the MSE cost.

$$w_i(k+1) = w_i(k) + \eta_w\left[\frac{2e_k x_{ki}}{\sigma + \mathbf{x}_k^T\mathbf{x}_k} - \lambda_k\beta pw_i^{p-1}sign(w_i^p)\right] \tag{8}$$

The constant σ is a small positive number used to avoid possible divisions by zero.

Selection of β, α and p: β is a positive stabilization constant that affects the convergence of the Lagrangian λ. It also acts as a balancing factor, weighing the constraint term against the MSE. A very high β will prioritize the constraint part of the cost function and the resulting residual MSE will be fairly high. On the other hand, a smaller β will tend to emphasize the MSE part and the constraint will not be strictly satisfied in the final estimate of **w**. In the experiments, we usually chose β in the interval (1-5). Constants p and α control the sparseness measure. Typically, both p and α are chosen to be unity. If α is made zero, there is a possibility of the weights converging to **0**. To prevent this, β should be scaled down appropriately. Also, the selection of α should be based on the input dimensionality. In our experiments, we chose α in the range 0.5-1, for data dimensions less than 200 and up to 1.5 if the dimensionality exceeds 200.

4. BRAIN MACHINE INTERFACE DESIGN

We will now utilize the ESN coupled with the sparse LMS algorithm to design a BMI. The neural recordings were collected in the Nicolelis primate laboratory at Duke University. Microwire electrode arrays [13] were chronically implanted in the dorsal premotor cortex, supplementary motor area, primary motor cortex and primary somatosensory cortex of an adult female monkey while performing a hand-reaching task. The task involved the presentation of a randomly placed target on a computer monitor in front of the monkey. The monkey used a hand-held joystick to move the computer cursor so that it intersects the target. While the monkey performed the motor task, the hand position was recorded in real time along with the corresponding neural activity from multiple channels [8]. In the modeling analysis presented here, 185 neurons were monitored; the neuronal spike events were then binned (added) in non-overlapping windows of 100ms and the behavioral datasets were downsampled and lowpass filtered to 10Hz. This data set was segmented into two exclusive parts: 5,400 samples for model training and 3,000 samples for model testing.

Design of the Echo State Network: The number of RNN units was chosen to be N=800. The input weight matrix $\mathbf{W}_{in}$(185x800) was fully connected with unity values. The recurrent connection matrix **W** had 1% randomly chosen non-zero weights. Moreover, we fixed all the non-zero weights to a value 0.5. Further, the parameters $\{a, C, \mu\}$ were set to $\{1,0.7,1\}$. Setting the spectral radius is crucial for this problem as it controls the dynamics and memory of the echo-states. Higher values are required for slow output dynamics and vice-versa. For the experiments in this paper, we used a spectral radius of 0.79. The output feedback as well as the direct connections between inputs and the linear mapper were turned off. The network state is set to zero initially. Training inputs were forced through the network and the states were updated using (1). The first 400 echo state outputs were discarded as transients. Remaining 5000 outputs were used to train the linear mapper.

Design of the Linear Mapper Using Sparse-LMS: The outputs from the RNN reservoir and the corresponding hand positions were used as training inputs and desired outputs respectively. The linear mapper is a matrix comprising of two column vectors $\mathbf{w}_x$, $\mathbf{w}_y$, each of length 800. The constraint parameters p, α were set to 1 and 1.5 respectively. The step-sizes for the weights (8) and the Lagrangian updates (6) were both chosen to be $1e$-3 and β was set to unity. The training was done for 20 epochs after which there was no significant reduction in the MSE.

Results and Discussions: Fig.2 shows the evolution of the Lagrangian term λ_k. Clearly, they converge to zero, which implies that the final weight vectors satisfy the imposed L_1 norm constraints. Fig.3 shows the learning curve for the weights. The residual MSE for the X direction is slightly more than that of the Y direction. A histogram of the weight vectors is plotted in fig. 4. Observe that most of the weights are zero, which demonstrates the sparseness of the weight matrix. Fig.5 shows the performance of the models on test data. Only the first 500 samples out of 3000 are shown here for clarity. The outputs of the linear mapper were then lowpass filtered by a 4th order Butterworth filter with normalized cutoff frequency of 0.2. The low pass filtering smoothens the output as the BMI interfaces with a robot arm.

The performance along Y is better and is evident from the overall correlation coefficients of {0.64, 0.78} for X and Y directions respectively. We further calculated the short-term correlation coefficients (fig. 6) over non-overlapping windows of 100 samples. It is clear that the short-term correlation coefficients are time varying for X and the model performs better only in short patches. This has been reported for other linear and non-linear models applied to this data [9]. A linear filter with ten tap delays per channel (totally 185x10x2 weights) gave correlation coefficients of {0.64, 0.75}. The RMLP with 185 inputs, 5 hidden nodes, and 2 outputs, gave correlation coefficients very similar to the proposed approach. By using weight decay, it is possible to reduce the number of parameters in the linear and RMLP models, but choosing a decay parameter is not trivial (requires a cross validation set for optimality). The proposed sparse LMS algorithm will automatically zero out the undesired parameters with an L_1 norm constraint.

5. CONCLUSIONS

This paper presents the use of Echo State Networks combined with static optimal and sparse linear mappers as a decoding algorithm for a BMI. ESNs provide a representational recurrent infrastructure that brings the information from the past of the input into the present sample *without* adapting parameters. The short-term memory of the ESNs is designed *apriori* thereby requiring only an adaptive linear or nonlinear regressor (static mapper) to implement functional mappings. As demonstrated here, they perform at the same level as other linear and nonlinear methods. However, the training can be done on-line in O(N) time, unlike the Wiener or the RMLP that have O(N^2) training algorithms. The number of trainable weights is also lower than each of these networks, and more importantly, ESNs will scale up much better. In fact, we believe that the recurrent reservoir also contains information for velocity and force mappings, which means that different regressors can capture these mappings in parallel. All these features (performance, scalability, data requirements, and algorithmic complexity) are very important for a practical BMI. We carefully designed the read out portion of the ESN to guarantee maximal generalization. We implemented an on-line procedure that adaptively creates a sparse interconnection matrix based on a L_1 norm penalty. This is novel, because weight decay uses an L_2 norm and moreover, one has to optimally set the forgetting factor on a cross validation data set. ESNs hold the promise of better performance and flexibility to design practical BMIs. However, the claim has to be quantified on different data sets and the overall generalization ability has to be observed. Lastly, we would like to mention the appeal of the ESN as a biologically plausible computational model. If one thinks of the echo states as neuronal states, we can see how a distributed, recurrent topology is capable of representing information about past inputs into a diffuse set of states (neurons). For the most part, the interconnectivity and the value of the weights (synapses) are immaterial for representation. They become however critical for readout (approximation). In this respect, there are very close ties between ESN and liquid state machines, as observed by Maas. These ideas may become

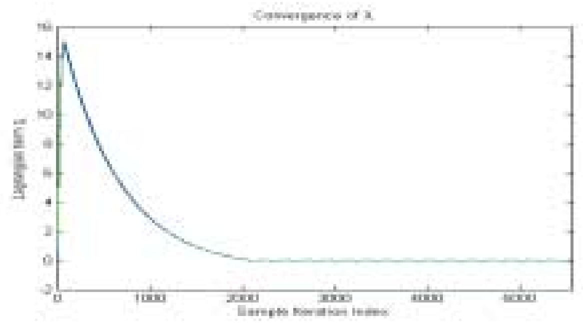

Figure 2. Convergence of the Lagrangian terms λ

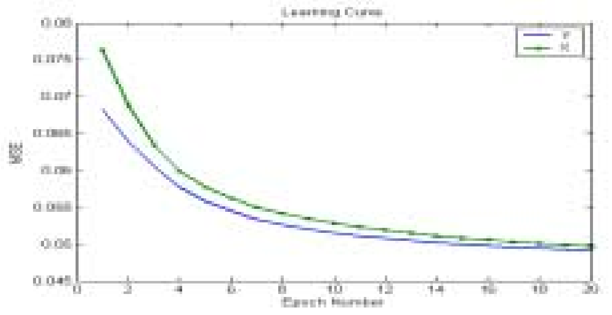

Figure 3. Learning curve

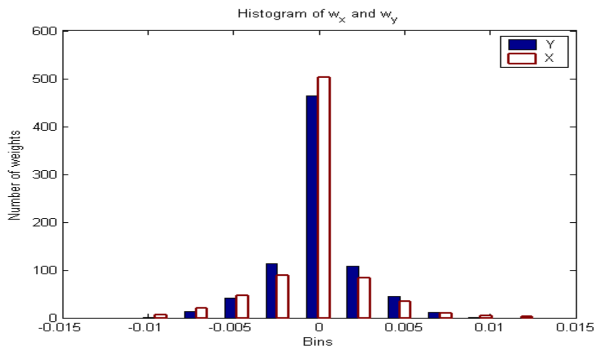

Figure 4. Histogram of the weights

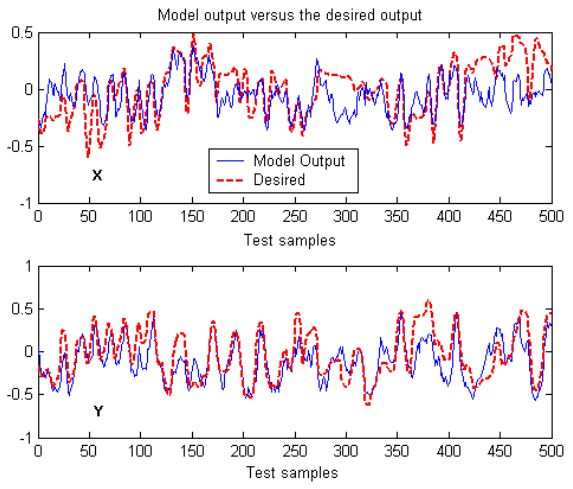

Figure 5. Model output and the desired output

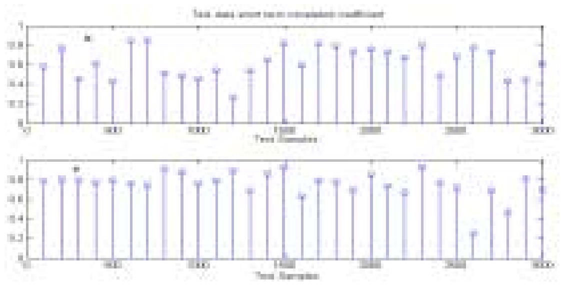

Figure 6. Windowed correlation coefficients

useful in developing new distributed paradigms for plasticity and characterization of neuronal response in motor cortex.

Acknowledgements: This work was supported by DARPA sponsored grant # ONR-450595112.

6. REFERENCES

[1] J. Wessberg, C.R. Stambaugh, J.D. Kralik, P.D. Beck, M. Laubach, J.K. Chapin, J. Kim, S.J. Biggs, M.A. Srinivasan, M.A.L. Nicolelis, “Real-time Prediction of Hand Trajectory by Ensembles of Cortical Neurons in Primates,” *Nature*, vol. 408, pp. 361-365, 2000.

[2] J.K. Chapin, K.A. Moxon, R.S. Markowitz, M.A.L. Nicolelis, “Real-time Control of a Robot Arm Using Simultaneously Recorded Neurons in the Motor Cortex,” *Nature Neuroscience*, vol. 2, pp. 664-670, 1999.

[3] Y. Gao, M.J. Black, E. Bienenstock, W. Wu, J.P. Donoghue, “A Quantitative Comparison of Linear and Non-linear Models of Motor Cortical Activity for the Encoding and Decoding of Arm Motions,” *IEEE EMBS CNE*, Capri, Italy, 2003.

[4] A.B. Schwartz, D.M. Taylor, S.I.H. Tillery, “Extraction Algorithms for Cortical Control of Arm Prosthetics,” *Current Opinion in Neurobiology*, vol. 11, pp. 701-708, 2001.

[5] M.D. Serruya, N.G. Hatsopoulos, L. Paninski, M.R. Fellows, J.P. Donoghue, “Brain-Machine Interface: Instant Neural Control of a Movement Signal,” *Nature*, vol. 416, pp. 141-142, 2002.

[6] J.C. Sanchez, S.P. Kim, D. Erdogmus, Y.N. Rao, J.C. Principe, J. Wessberg, M.A.L. Nicolelis, “Input-Output Mapping Performance of Linear and Nonlinear Models for Estimating Hand Trajectories from Cortical Neuronal Firing Patterns,” *Proc. Neural Networks for Signal Processing*, pp. 139-148, Martigny, Switzerland, 2002.

[7] S.P. Kim, J.C. Sanchez, D. Erdogmus, Y.N. Rao, J.C. Principe, M.A.L. Nicolelis, “Divide-and-Conquer Approach for Brain Machine Interfaces: Nonlinear Mixture of Competitive Linear Models,” *Neural Networks*, vol. 16, no. 5-6, pp. 865-871, Jun 2003.

[8] J.M. Carmena, M.A. Lebedev, R.E. Crist, J.E. O'Doherty, D.M. Santucci, D.F. Dimitrov, P.G. Patil, C.S. Henriquez, M.A.L. Nicolelis, “Learning to Control a Brain-Machine Interface for Reaching and Grasping by Primates,” PLoS Biology, vol. 1, pp. 193-208, 2003.

[9] J.C. Sanchez, D. Erdogmus, J.C. Principe, J. Wessberg, M.A.L. Nicolelis, “A Comparison Between Nonlinear Mappings and Linear State Estimation to Model the Relation from Motor Cortical Neuronal Firing to Hand Movements,” SAB’02, pp. 59-65, Scotland, Aug 2002.

[10] H. Jaeger, “The “Echo State” Approach to Analyzing and Training Recurrent Neural Networks,” GMD Report 148, *GMD-German National Research Institute for Computer Science*, 2001.

[11] W. Maas, T. Natschläger, H. Markram, “Real-time Computing without Stable States: A New Framework for Neural Computation Based on Perturbations,” *Neural Computation*, 14(11):2531-2560, 2002.

[12] D.G. Luenberger, *Linear and Non-linear Programming, Addison Wesley*, 1989.

[13] M.A.L. Nicolelis, D. Dimitrov, J.M. Carmena, R. Crist, G. Lehew, J.D. Kralik, S.P. Wise, “Chronic, Multi-Site, Multi-Electrode Recordings in Macaque Monkeys,” *Proc. National Academy of Sciences of the U.S.A.*, vol. 100, no. 19, pp. 11041-11046, Sep 2003.