Design and Implementation of Neural Network Based Chaotic System Model for the Dynamical Control of Brain Stimulation

Lei Zhang

Faculty of Engineering and Applied Science
University of Regina
Regina, Canada S4S 0A2
Email: lei.zhang@uregina.ca

***Abstract*—Brain stimulation has been used in practice to treat neurological diseases, such as Parkinson's Disease and Epilepsy. However, the stimulation signals are generated based on trail and error; and the underpinning theory of this treatment is still unclear. Artificial neural network (ANN) resembles biological neural network in the brain and has been used for many artificial intelligence applications such as classification and pattern recognition. In order to generate accurate stimulation signals in brain stimulation treatment, it is beneficial to establish an ANN model to simulate the brain dynamics and study the effects of various stimulation signals. Previous research shows that brain activities captured by Electroencephalogram (EEG) demonstrate chaotic patterns. Chaotic systems, such as Hénon map can be represented by a set of mathematical equations, and therefore are predictable and controllable. The aim of this research is to implement an optimal ANN architecture model to generate the output pattern of a chaotic system, which can be used to simulate the brain dynamics under stimulation. This paper presented the preliminary work of an ANN architecture design and optimization for generating the outputs of Hénon map chaotic system, and the simulation results for controlling the chaotic system with periodic stimulation signals. The ANN design method and chaotic control method can be extended for other chaotic systems in general.**

Keywords–Brain stimulation; Chaotic systems; Artificial Neural networks; Dynamic control; Hénon map.

I. INTRODUCTION

The growing interest in brain stimulation as a form of neuromodulation has led to increased empirical data from clinical practice for further theoretical research. Targeted brain stimulation has been employed to treat neurological diseases, with reported success in controlling shaking in Parkinson's Disease and Epilepsy seizures. However, theoretical and analytical research is urgently in need to discover the impact, especially the potential long-term side effects of these focal perturbations. Therefore, it is critically important to develop theoretical models in order to design high performance dynamical control systems for brain stimulation.

Brain stimulation has been employed clinically to diagnose, monitor and treat neurological disorders, such as epilepsy seizures [1] [2] and Parkinson's disease [3]–[6]. Commonly used non-invasive stimulation methods include transcranial magnetic stimulation (TMS), transcranial direct current stimulation (tDCS) [7] [8] and transcranial focused ultrasound [9]. Various data recording methods and devices, such as positron emission computed tomography (PET) [6], electroencephalograph (EEG) [10], functional magnetic resonance imaging (fMRI) [11] [12], and recently some single devices [13]–[16] have been used to monitor the brain stimulation effects on brain activities. This provides valuable research data for further study to optimize stimulation protocols, which include identifying accurate target stimulation area and applying effective stimuli, in order to improve the performance of the treatment and meanwhile minimize or eliminate the potential side-effects.

Previous research reported that brain waves demonstrate chaotic behaviors [17] [18]. A chaotic system is a bound system which obtains the existence of an attractor. Chaotic time series are dynamic systems that are extremely sensitive to initial conditions and can exhibit complex external behavior. A chaotic system can be stable, periodic or chaotic depending on the system parameters as well as its initial conditions. A known chaotic system can be analyzed and controlled based on its system equations using conventional dynamic control methods. However, when the equations of a chaotic system are unknown, such as the EEG time series signals, the pattern recognition of such a system and the discovery of its system parameters become a challenging task, which is important for the dynamic control of the system.

Artificial Neural Network (ANN) is a leading machine learning method inspired by biological neural network structure. In recent years, ANN has been widely used for pattern recognition and classification based on a number of pre-defined features. An ANN model with a feedback loop can be designed to generate chaotic outputs by training the ANN using the output values of a chaotic system with selected parameters and initial conditions [19]. Since a chaotic system with specified initial values and system parameters can be represented by an ANN model, the system can be controlled by varying the weight and bias values of the ANN. The training process is carried out on a computer and the weights are generated for all neurons in an ANN architecture. These weights are then used in constructing an ANN model to generate the expected outputs for the target chaotic system. The implementation cost and speed of an ANN architecture is determined by its complexity, therefore it is beneficial to use less number of hidden neurons to achieve the target training performance. A simple ANN design generally has a 3-layer architecture, including one input, one hidden and one output layer. The MATLAB Neural Network Toolbox is used to train the ANN with three MATLAB training algorithms: Levenberg-Marquardt, Scaled Conjugate Gradient algorithm and Bayesian Regulation. The optimization of the ANN architecture is important for improving the performance of hardware implementation on a Field Programmable Gates Array (FPGA) device.

 ISBN: 978-1-61208-579-1

Network control theory has been used to study the effects of stimulation on brain networks [20]. Network control refers to the possibility of manipulating local interaction of dynamic components to steer the global system along a chosen trajectory. The neural network based chaotic model can aid the understanding of the underlying structural connectivity that modulates the system dynamics. A novel approach is presented to combine the ANN design and the dynamical control theory for the control of brain dynamics. In this approach, an ANN model is designed with optimized architecture based on Hénon map chaotic system. Hénon map [21] has its significance in studying chaotic systems with a simple 2-dimensional structure and is used initially as the study subject of the research project.

In previous related work, a model-based hardware implementation of the Hénon map is presented in [22] and the dynamic analysis of the system stability at critical points with varying system parameters is provided by [23]. The chaotic system can be controlled to change from chaotic mode to stable or periodic mode by adding a period stimulation pulse signals. The design approach and control method can be easily extended for other chaotic systems in general, such as 3-dimensional Lorenz attractor [24].

Section II describes the ANN design and training procedure; section III explains the chaos control of the Hénon map; section IV discusses the conclusion and future work.

II. ANN Design and Training

The ANN is a network of interconnected neurons arranged in multiple layers, including one input layer, one output layer and one or multiple hidden layers. A general mathematic representation of an individual neuron within an ANN architecture is shown by equation(1).

$$a_j^l = \sum_{i=1}^{N_{l-1}} w_{j,i}^l x_i + b_{j,0}^l \qquad j = 1, 2, ...N_l$$
$$y_j^l = f_l(a_j^l) \tag{1}$$

where N_l is the number of neurons at l-layer. Each hidden neuron j receives the output of each input neuron i from the input layer multiplied with a weight of $w_{j,i}^l$. The sum of all weighted inputs is used by an activation function f_l to produce the output of the hidden layer neuron and feed it forward to the output layer. A similar weighted sum is generated for each output neuron. $b_{j,0}^l$ is the bias of the jth neuron at the lth layer, which are added as noise to randomize the initial condition in order to get better chance to converge. The weight matrix connecting between the $(l-1)^{th}$ layer to the l^{th} layer is represented by equation (2).

$$\mathbf{W}^l = \begin{bmatrix} w_{1,1} & w_{1,2} & ... & w_{1,N_{l-1}} \\ w_{2,1} & w_{2,2} & ... & w_{2,N_{l-1}} \\ ... & ... & ... & ... \\ w_{N_l,1} & w_{N_l,2} & ... & w_{N_l,N_{l-1}} \end{bmatrix}_{N_l \times N_{l-1}} \tag{2}$$

Let $\mathbf{y}^l = (y_1, y_2, ..., y_{N_l})$ be the output vector from the l^{th} layer, the weighted sum vector to the $l+1^{th}$ layer is represented by (3), the outputs are calculated using (4), where f is the activation function.

$$\mathbf{a}_k^{l+1} = \sum_{j=1}^{N_l} \mathbf{y}_j * w_{j,k}, \quad (k = 1, 2, ..., N_{l+1})$$
$$\mathbf{a}^{l+1} = \mathbf{y}^l \mathbf{W}^l y_j^{l+1} = f(a_j^{l+1}), \quad j = 1, 2, ..., N_{l+1} \tag{3}$$

$$\mathbf{y}^{l+1} = f(\mathbf{a}^{l+1}) = \begin{bmatrix} y_1 * w_{1,1} & y_1 * w_{1,2} & ... & y_1 * w_{1,N_{l+1}} \\ y_2 * w_{2,1} & y_2 * w_{2,2} & ... & y_2 * w_{2,N_{l+1}} \\ ... & ... & ... & ... \\ y_{N_l} * w_{N_l,1} & ... & ... & y_{N_l} * w_{N_l,N_{l+1}} \end{bmatrix}_{N_l \times N_{l+1}} \tag{4}$$

A. ANN Architecture Design

The ANN model design process includes identifying the correct topology of the network. An optimal ANN architecture should contain minimal number of hidden layers and hidden neurons, and yet sufficient enough for representing the variability of the training data. A network with insufficient complexity will fail to learn the underlying function, while a network with more neurons and layers than required will cause overfitting of the model and fail to generalize. An ANN model design approach for chaotic systems based on model topology analysis [19] is used. After the construction of the ANN, its predictive ability needs to be measured. The accuracy is a quantification of the proximity between the outputs of the ANN and the target output values, measured by mean square error (MSE).

The implementation performance of the chaotic system depends on the ANN topology. For the Hénon map ANN design, both input and output layer has two neurons, one hidden layer is employed. In order to optimize the ANN architecture with a minimal number of the hidden neurons to improve implementation performance, 16 different ANN topologies with one to sixteen hidden neurons are trained using three training algorithms respectively to find the optimal ANN topology. Sigmoid function is used as the activation function for the neurons in the hidden layer. Ramp activation function is used for the output layer [25].

B. Training Data and Training Algorithms

The 6,000 training samples for Hénon map are generated in MATLAB using the ode23 method. During the training process, the first pair of samples (x_1, y_1) is provided as the 2 inputs of ANN, the second pair of samples (x_2, y_2) is provided as the target outputs. Then the second pair is provided as the inputs and the third pair as the target outputs, and so on. The training samples are divided into three subsets: training(70%), validation(15%) and testings(15%). The training set is used for computing the gradient and updating the network weight and bias values. The validation set is used to monitor the error during training process in order to avoid overfitting. The test set is used to test the training performance. The ANN training is carried out using three network training functions provided by the MATLAB Neuron Network Toolbox: $trainlm, trainbr$ and $trainscg$. $trainlm$ function updates weight and bias values based on Levenberg-Marquardt (LM) optimization [26]. $trainbr$ function also updates weights and biases based on LM optimization. It uses Bayesian Regulation (BR) process [27] to minimize and determine a combination of squared errors and weights, in order to produce a network

 ISBN: 978-1-61208-579-1

TABLE I. ANN TRAINING FUNCTION PARAMETERS

Training Fuctions	LM	BR	SCG
Learning rate	0.01	0.01	0.01
Momentum Constant	0.9	0.9	0.9
Maximum Epochs	1000	1000	1000
Maximum Training Time	inf	inf	inf
Performance Goal	0	0	0
Minimum Gradient	1.00E-07	1.00E-07	1.00E-06
Maximum Validation Checks	6	0	6
Mu	0.001	0.005	N/A
Mu Decrease Ratio	0.1	0.1	N/A
Mu Increase Ratio	10	10	N/A
Maximum mu	1.00E+10	1.00E+10	N/A
Sigma	N/A	N/A	5.00E-05
Lambda	N/A	N/A	5.00E-07

with good generalization. $trainscg$ function updates weight and bias values based on Scaled Conjugate Gradient (SCG) method [28]. The training parameters of these three MATLAB functions are listed in Table. I.

Each algorithm is used for 16 different ANN topologies, with the number of hidden neurons increasing from 1 to 16 in the architecture. Three training iterations are carried out per architecture per algorithm. Each training iteration runs for 1000 epochs. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. The learning rate is 0.01. It is a constant used by the training algorithm to update the weight and bias values at each step. The momentum is 0.9. This is another constant value for adjusting the learning rate by adding a proportion of the weight value in the previous step. As the training time is infinity (inf) and the training goal is 0, the training stops when the performance gradient falls below the Minimum Gradient, or the Maximum Epoch is reached, or the validation performance has increased more than the Maximum Validation Checks since the last time it decreases.

C. ANN Performance Evaluation

The ANN training result is measured by the error between the calculated ANN output y and the target training output $\hat{y}$. The training target is a threshold error value small enough for the output to be considered as correct. The performance of the ANN training process is evaluated by how fast and well the error converge to the target threshold. The most common method for measuring the output error is MSE, as illustrated respectively by (5).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{5}$$

where N is the number of outputs, which is 2 in the case of Hénon map: y_1 and y_2. For 16 architectures and three training algorithms: LM, BR and SCG, the ANN training is carried out three repeated iterations per architecture per algorithm. The training performance (MSE) are listed in Table. II.

The training performance of the Levenberg-Marguardt algorithm is shown in Figure 2(a). It can be observed that the MSE values for all three iterations over the number of hidden neurons have non-monotonicity. Nevertheless, the MSE is decreasing in general. The MSE values for all three iterations decrease below 1.7E-07 $(= 1.7 \times 10^{-7})$ with only 2 hidden neurons. The MSE of iteration I increases when the number of hidden neurons increases from 2 to 3; and the MSE of iteration III has big increase from 2.3E-8 to 8.6E-6 (by 2 logarithmic scales) when the number of hidden neurons increase from 3 to 4. The best performance is achieved with 15 hidden neuron in iteration II, when MSE=1.1397E-10. Although the overall performance is improving with increasing number of hidden neurons, it is hard to predict accurately whether the performance can be improved by adding one more neuron for an individual iteration. The result is random.

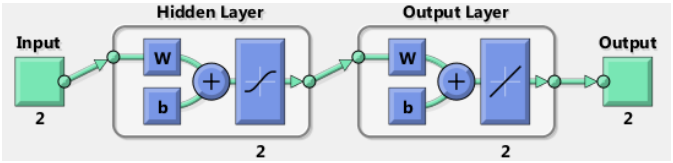


Figure 1. ANN Architecture for Hénon Map Chaotic System

The training performance of the Bayesian Regulation algorithm is shown in Figure 2(b). Similar to the LM algorithm, the MSE values for all three iterations decrease below 1E-07 with only 2 hidden neurons, but increase as the number increases from 2 to 3 for iterations I and III. The overall trend is for the MSE to gradually decrease while the number of hidden neurons increases, but the effect for adding or removing one neuron for each training iteration is unpredictable. The smallest MSE (3.1178E-12) is achieved by iteration III with 15 hidden neurons.

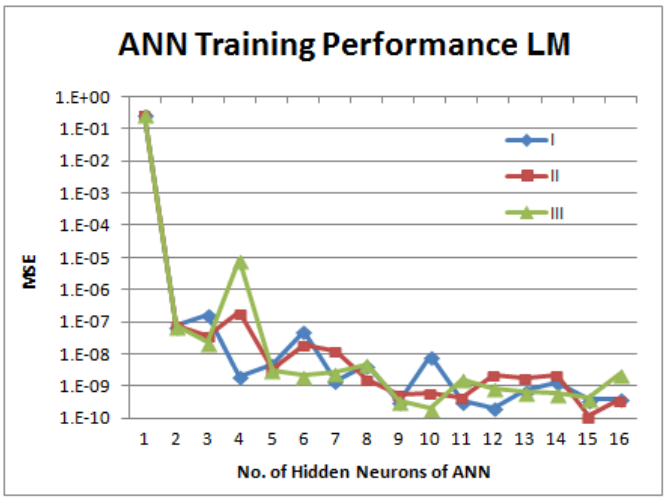
The training performance of the Scaled Conjugate Gradient algorithm is shown in Figure 2(c). The smallest performance (MSE=5.9783E-05) is achieved when n=4. The MSEs increases generally while n increases after n=4. The LM and BR algorithms have better training performance than the SCG algorithm.

The average MSE values of all three iterations for each algorithm is compared in Figure 2(d). The training performance can be only improved slightly once the number of hidden neurons is greater than 2. The increased number of hidden neurons will increase hardware resource utilization for the system implementation. The ANN model is generated using double precision floating point data format during the training process in MATLAB simulation environment. The fixed-point data format for the FPGA implementation does not require the target MSE to be smaller than the quantization error. For instance, the 32-bit fixed-point data format with 18 fractional bits can have $2^{-18}(\approx 3.8147e{-}06)$ resolution, which is bigger than the smallest MSE achieved by the ANN model with 2 hidden neurons using the LM and BR training algorithms. Therefore, the ANN model is designed using 2 hidden neurons. The ANN architecture is illustrated by Figure 1, including an input layer with two inputs, a hidden layer with two hidden neuron, and an output layer with two outputs. A Simulink Model is created using the ANN with delayed feed-back loop as shown in Figure 3; and the Simulink simulation output of the ANN-based Hénon map chaotic system is shown in Figure 4. The training performances and training states for the three training functions of the selected architecture with 2 hidden neurons are plotted in Figure 5. For LM and BR, the training stops at epoch 1000. For the SCG, the training stops at epoch 255 as the maximum number of validation fails reaches 6.

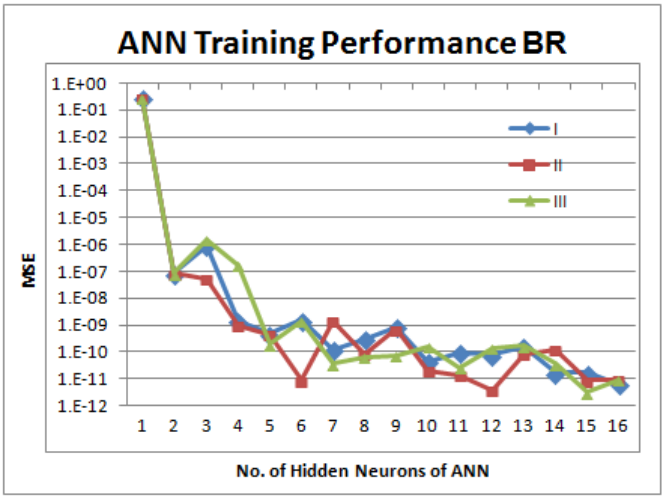
 ISBN: 978-1-61208-579-1

TABLE II. TRAINING PERFORMANCE (MSE) OF 3 TRAINING ALGORITHMS WITH 16 ARCHITECTURES – EPOCH 1000

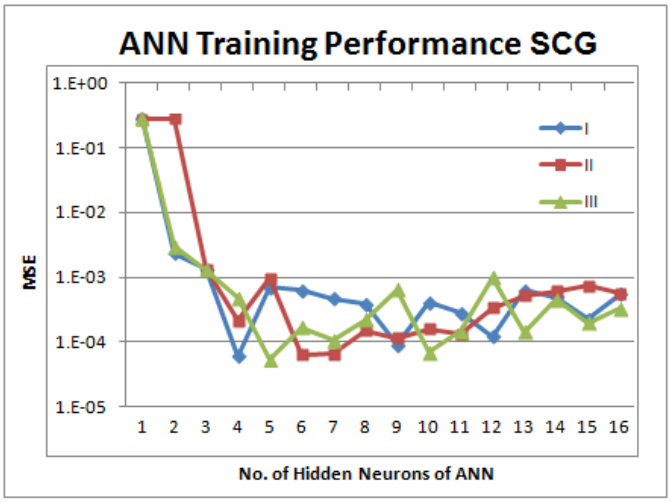
MSE	Levenberg-Marguardt (LM)			Bayesian Regulation (BR)			Scaled Conjugate Gradient (SCG)		
Neurons	I	II	III	I	II	III	I	II	III
1	0.2810	0.2810	0.2810	0.2810	0.2810	0.2810	0.2810	0.2810	0.2828
2	7.5920E-08	7.5411E-08	7.5841E-08	8.1986E-08	8.1497E-08	8.1474E-08	2.3649E-03	2.8101E-01	2.9964E-03
3	1.7042E-07	3.7541E-08	2.3484E-08	9.1948E-07	5.4859E-08	1.5598E-06	1.3356E-03	1.3299E-03	1.3042E-03
4	2.0187E-09	1.9369E-07	8.5939E-06	1.4455E-09	9.3581E-10	1.7478E-07	5.9783E-05	2.1042E-04	4.8217E-04
5	4.7927E-09	3.5006E-09	3.1200E-09	4.7033E-10	4.6482E-10	1.9332E-10	6.9701E-04	9.5137E-04	5.3533E-05
6	5.2199E-08	1.8775E-08	2.1202E-09	1.5030E-09	8.8459E-12	1.4452E-09	6.2341E-04	6.2949E-05	1.6708E-04
7	1.4853E-09	1.2016E-08	2.7629E-09	1.2308E-10	1.4387E-09	3.5900E-11	4.6338E-04	6.5173E-05	1.0628E-04
8	4.5645E-09	1.6143E-09	4.8532E-09	2.9488E-10	8.4649E-11	6.2597E-11	3.8860E-04	1.4984E-04	2.1843E-04
9	3.2232E-10	5.3065E-10	3.3891E-10	8.5851E-10	7.0971E-10	7.0933E-11	8.9483E-05	1.1520E-04	6.4021E-04
10	8.5592E-09	6.0050E-10	1.9888E-10	4.2823E-11	1.9704E-11	1.6047E-10	4.0709E-04	1.5928E-04	6.7822E-05
11	3.4364E-10	4.5139E-10	1.6145E-09	9.6083E-11	1.3954E-11	2.6842E-11	2.7726E-04	1.2903E-04	1.4429E-04
12	2.0994E-10	2.1447E-09	8.7262E-10	7.7306E-11	3.9343E-12	1.4594E-10	1.2205E-04	3.3368E-04	9.9476E-04
13	7.4649E-10	1.8049E-09	6.6516E-10	1.5392E-10	8.2397E-11	1.6851E-10	6.2968E-04	5.2315E-04	1.4134E-04
14	1.3254E-09	2.2304E-09	5.6996E-10	1.6843E-11	1.1866E-10	3.6662E-11	4.8872E-04	6.1157E-04	4.5022E-04
15	3.8671E-10	1.1397E-10	4.1528E-10	1.6680E-11	8.8259E-12	3.1178E-12	2.2542E-04	7.3086E-04	1.9543E-04
16	4.1175E-10	3.7105E-10	2.3944E-09	6.6127E-12	9.4176E-12	8.9851E-12	5.4813E-04	5.5410E-04	3.2486E-04



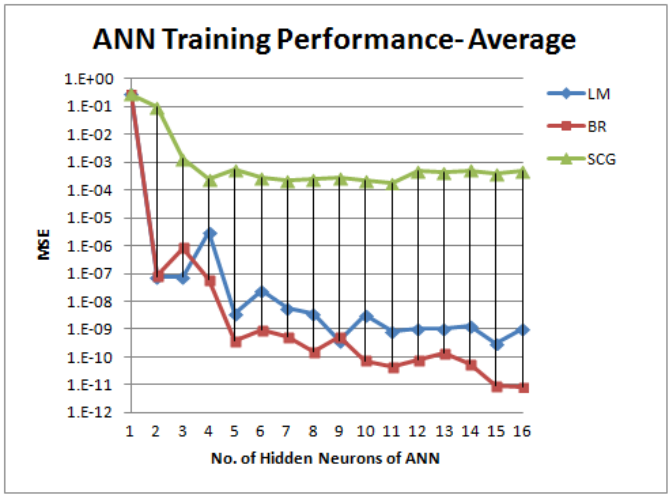
(a) Levenberg-Marquardt



(b) Bayesian Regulation



(c) Scaled Conjugate Gradient



(d) Average of 3 Iterations

Figure 2. ANN Training Performance for Hénon Map

 ISBN: 978-1-61208-579-1

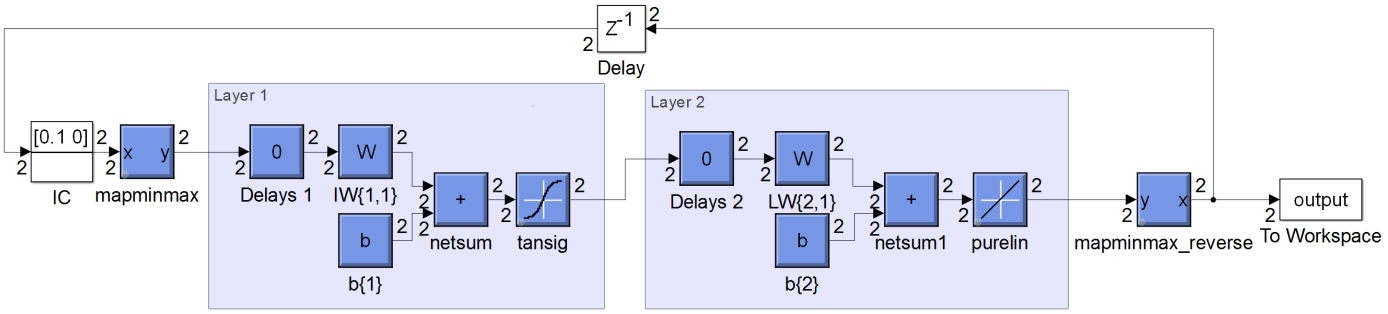


Figure 3. Simulink Model for ANN-based Hénon Map Chaotic System

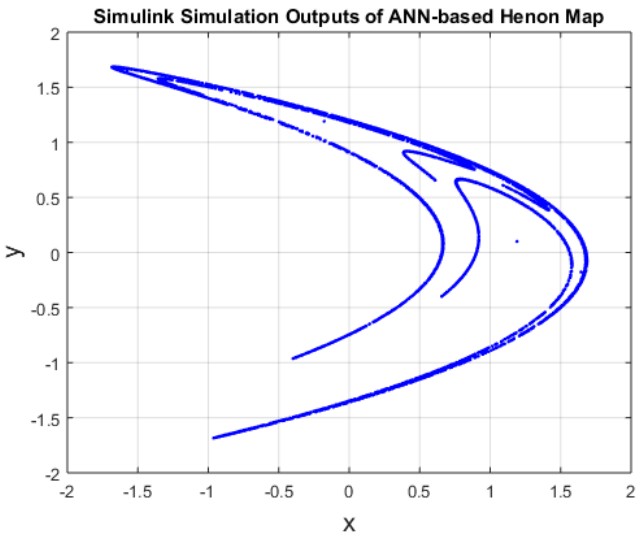


Figure 4. Simulink Simulation Outputs of ANN Model (6000 samples)

III. Chaos Control of the Hénon Map

In order to control chaotic systems for brain stimulation, periodic pulses can be generated as stimuli. The stability of a chaotic system at its critical points can be analyzed by calculating the eigenvalues of the Jacobian matrix of its system equations. Based on the analysis, stimulation signals can be generated to alter the state of the chaotic system.

A. Periodic Orbits and Critical Points

Given a 2-dimensional discrete system $X_{n+1} = F(X_n)$, which can be represented by (6).

$$x_{n+1} = P(x_n, y_n), \qquad y_{n+1} = Q(x_n, y_n) \tag{6}$$

where X is a vector in $\mathbf{R^2}$; F is a map of a domain D of $\mathbf{R^2}$ onto itself; P and Q are scalar valued functions. A critical point of a system is defined as a point at which $X_{n+1} = F(X_n) = X_n$ for all n [29]. The term 'critical point' is often referred to as 'fixed point' of dynamic system. The type of critical point is determined from the eigenvalues of the Jacobian matrix of the function $F(X)$ at the critical point. A critical point of period N is a point at which $X_{n+N} = F^N(X_n) = X_n$, for all n. Given that a discrete nonlinear system represented by (6) has a critical point at (x_s, y_s), the Jacobian matrix at the critical point can be represented by (7).

$$J(x_s, y_s) = \begin{pmatrix} \frac{\partial P}{\partial x} & \frac{\partial P}{\partial y} \\ \frac{\partial Q}{\partial x} & \frac{\partial Q}{\partial y} \end{pmatrix}\Bigg|_{(x_s, y_s)} \tag{7}$$

Suppose that the Jacobian matrix has eigenvalues λ_1 and λ_2. In the discrete case, the critical point is stable as long as $|\lambda_1| < 1$ and $|\lambda_2| < 1$, otherwise the critical point is unstable. The critical points of period one X_s satisfies $X_s = F(X_s)$.

The discrete Hénon map equations by definition are lised in (8).

$$x_{n+1} = 1 + y_n - ax_n^2, \qquad y_{n+1} = bx_n \tag{8}$$

where x_n and y_n are system variables, α and β are system parameters. A reformed equivalent set of equations are obtained by taking a transformation $x'_n = \frac{1}{\alpha}x_n, y'_n = \frac{\beta}{\alpha}y_n$, as listed in (9). It is easier to use the reformed equations for stability analysis and control [23].

$$x_{n+1} = \alpha + \beta y_n - x_n^2, \qquad y_{n+1} = x_n \tag{9}$$

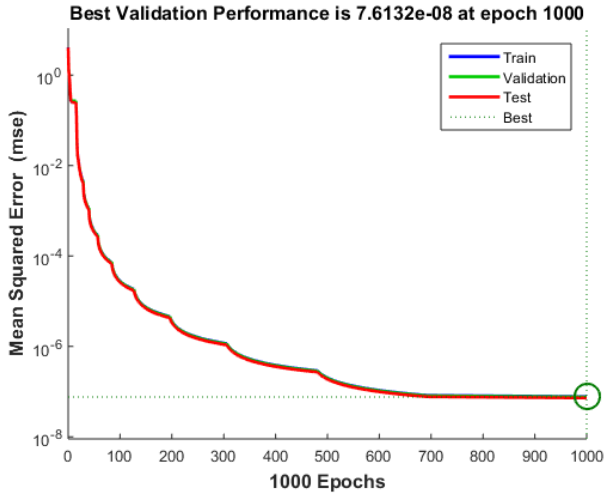
The critical points of period one and the Jacobian matrix for the original Hénon Map are listed in (10).

$$\begin{aligned} x_s =& \frac{(\beta - 1) \pm \sqrt{(1-\beta)^2 + 4\alpha}}{2\alpha} \\ y_s =& \beta \left(\frac{(\beta - 1) \pm \sqrt{(1-\beta)^2 + 4\alpha}}{2\alpha} \right) \\ J =& \begin{pmatrix} -2ax & 1 \\ \beta & 0 \end{pmatrix} \end{aligned} \tag{10}$$

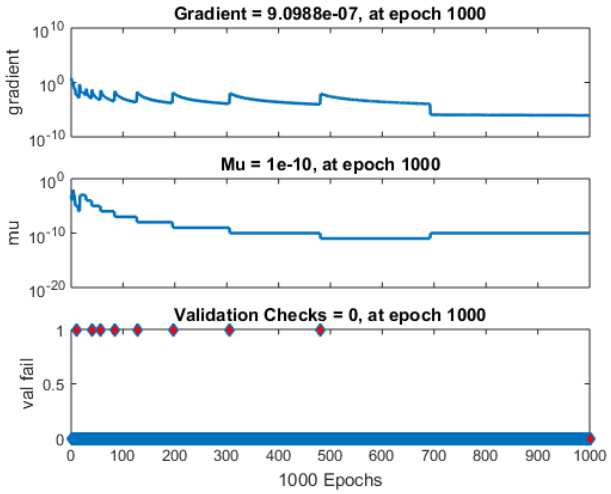
The critical points, Jacobian matrix and its two eigenvalues of the reformed Hénon map equations are listed in (11).

$$\begin{aligned} x_1 =& y_1 = \frac{\beta - 1 - \sqrt{(\beta-1)^2 + 4\alpha}}{2} \\ x_2 =& y_2 = \frac{\beta - 1 + \sqrt{(\beta-1)^2 + 4\alpha}}{2} \\ J =& \begin{pmatrix} -2x & \beta \\ 1 & 0 \end{pmatrix} \\ \lambda_{1,2} =& \begin{pmatrix} \sqrt{x^2 + \beta} - x & 0 \\ 0 & -\sqrt{x^2 + \beta} - x \end{pmatrix} \end{aligned} \tag{11}$$

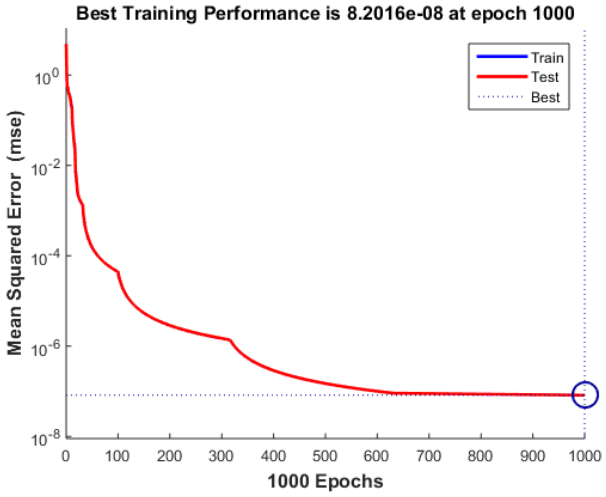
 ISBN: 978-1-61208-579-1



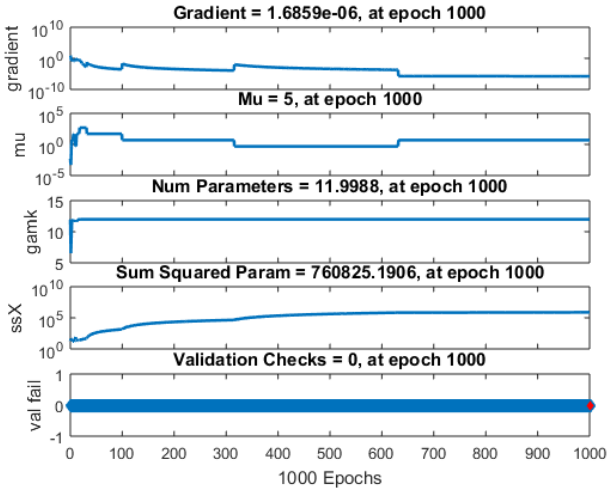
(a) Levenberg-Marquardt Training Performance



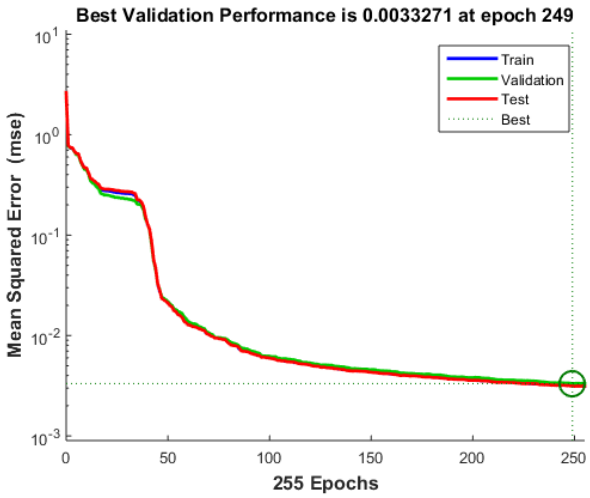
(b) Levenberg-Marquardt Training States



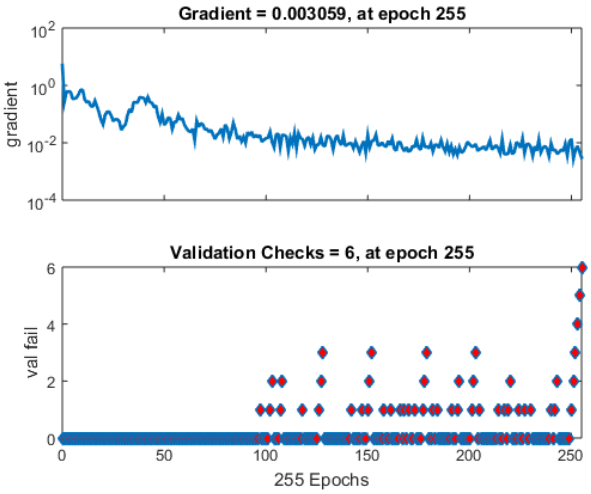
(c) Bayesian Regulation Training Performance



(d) LBayesian Regulatio Training States



(e) Scaled Conjugate Gradient Training Performance

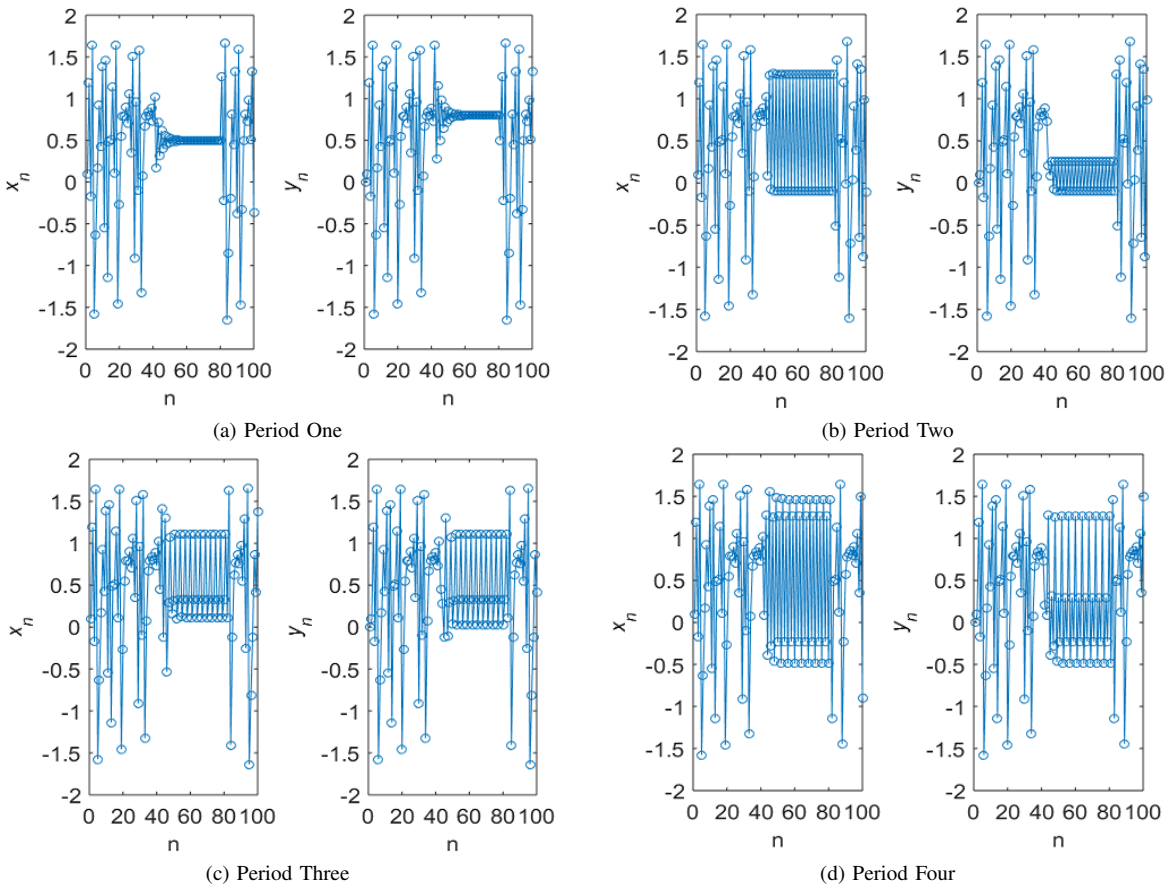


(f) Scaled Conjugate Gradient Training States

Figure 5. ANN Training Performance for ANN with 2 Hidden Neurons

The Hénon map has two real critical points of period one if and only if $(1-\beta)^2 + 4\alpha > 0$. Therefore $\alpha > 0, |\beta| < 1$. The determinate of the Jacobian matrix is $|\beta|$, which is less than 1.

 ISBN: 978-1-61208-579-1



(a) Period One (b) Period Two

(c) Period Three (d) Period Four

Figure 6. MATLAB Simulation for Hénon Map Chaos Control. $\alpha = 1.2, \beta = 0.4, k = 0.2$; Control period: $40 < n < 80$; Initial values: $x_0 = 0.1, y_0 = 0$

B. *Chaos Control Using Periodic Proportional Pulses Method*

Control and synchronization of chaotic system using the *periodic proportional pulses* method is evaluated in [29] [30] and can be adapted for the control of brain stimulation. Instantaneous pulses can be applied to the system variables X_n every N iterations. Define the composite function $C = KF^N$. K is the diagonal matrix with diagonal elements k_1 and k_2, which can be derived for a given period N and a given critical point X_s. A critical point $X_s = (x_s, y_s)$ of the function F satisfies $KF^N(X_s) = (X_s)$. The Jacobian(J) of C has two eigenvalues. The critical point is locally stable if both the modulus of eigenvalues are less than one. The absolute of the determinate of $J : |det(J)| = |\beta| < 1$ indicates that the Hénon chaotic system is a attractor.

Practically, this method can be easily applied when dealing with chaos control with periodic orbits of low periods. Figure 6 shows time series signals with period-one, period two, period three and period-four stimulation respectively.

IV. Conclusion and Future Works

The paper presented an ANN-based Hénon map chaotic system model design and the optimization of the ANN architecture with the consideration for FPGA hardware implementation. The ANN-based chaotic system model is designed for simulating chaotic brain activities and the dynamic control of the chaotic system by varying the weight and bias values of the ANN. The ANN has a 3-layer architecture and is designed using 16 different topologies with 1 to 16 hidden neurons respectively. It is shown that the simple topology of 2 hidden neurons can be implemented to generate the desired chaotic outputs, which is highly beneficial to improve performance and reduce resource utilization for hardware implementation. The ANN model was trained by 3 MATLAB training functions: $trainlm$, $trainbr$ and $trainscg$, which can be used to generate Simulink model for simulation, and for the further FPGA hardware model development using the Xilinx System Generator in the MATLAB software environment. The ANN design approach with topology optimization can be extended to other chaotic systems and will be used for the design and development of a dynamic control system for non-invasive brain stimulation in order to treat neurological disease. The chaotic system analysis and control is also discussed based on Hénon map, which can be extended to other chaotic systems. The control simulation results are presented as preliminary research work and will be applied to the ANN-based chaotic system control for brain stimulation. Future work includes the hardware implementation of the designed ANN on FPGA device. The ANN design and optimization approach will be used with other state-of-the-art training algorithms to compare the training speed and training performance in order to develop an optimal online training algorithm for hardware implementation.

 ISBN: 978-1-61208-579-1

REFERENCES

[1] V. C. Terra et al., "Vagus nerve stimulator in patients with epilepsy: indications and recommendations for use," Arquivos de Neuro-Psiquiatria, vol. 71, no. 11, nov 2013, pp. 902–906.

[2] C. M. DeGiorgio and S. E. Krahl, "Neurostimulation for drug-resistant epilepsy," CONTINUUM: Lifelong Learning in Neurology, vol. 19, jun 2013, pp. 743–755.

[3] T. D.-B. S. for Parkinson's Disease Study Group, "Deep-brain stimulation of the subthalamic nucleus or the pars interna of the globus pallidus in parkinson's disease," New England Journal of Medicine, vol. 345, no. 13, sep 2001, pp. 956–963.

[4] M. C. Rodriguez-Oroz, "Bilateral deep brain stimulation in parkinson's disease: a multicentre study with 4 years follow-up," Brain, vol. 128, no. 10, jul 2005, pp. 2240–2249.

[5] G. Deuschl et al., "A randomized trial of deep-brain stimulation for parkinson's disease," New England Journal of Medicine, vol. 355, no. 9, aug 2006, pp. 896–908.

[6] S. T. Grafton et al., "Normalizing motor-related brain activity: Subthalamic nucleus stimulation in parkinson disease," Neurology, vol. 66, no. 8, apr 2006, pp. 1192–1199.

[7] S. R. Soekadar, M. Witkowski, E. G. Cossio, N. Birbaumer, S. E. Robinson, and L. G. Cohen, "In vivo assessment of human brain oscillations during application of transcranial electric currents," Nature Communications, vol. 4, jun 2013.

[8] L. D. J. Fiederer et al., "Electrical stimulation of the human cerebral cortex by extracranial muscle activity: Effect quantification with intracranial EEG and FEM simulations," IEEE Transactions on Biomedical Engineering, vol. 63, no. 12, dec 2016, pp. 2552–2563.

[9] K. Yu, A. Sohrabpour, and B. He, "Electrophysiological source imaging of brain networks perturbed by low-intensity transcranial focused ultrasound," IEEE Transactions on Biomedical Engineering, vol. 63, no. 9, sep 2016, pp. 1787–1794.

[10] F. Ferreri and P. M. Rossini, "TMS and TMS-EEG techniques in the study of the excitability, connectivity, and plasticity of the human motor cortex," Reviews in the Neurosciences, vol. 24, no. 4, jan 2013.

[11] A. Pascual-Leone et al., "Characterizing brain cortical plasticity and network dynamics across the age-span in health and disease with TMS-EEG and TMS-fMRI," Brain Topography, vol. 24, no. 3-4, aug 2011, pp. 302–315.

[12] J. Leitao, A. Thielscher, S. Werner, R. Pohmann, and U. Noppeney, "Effects of parietal TMS on visual and auditory processing at the primary cortical level - a concurrent TMS-fMRI study," Cerebral Cortex, vol. 23, no. 4, apr 2012, pp. 873–884.

[13] H. Kim, P. Yves, and K. Lee, "Development of chip-less and wireless neural probe functioning stimulation and reading in a single device," Microelectronic Engineering, vol. 158, jun 2016, pp. 118–125.

[14] "Device and circuitry for controlling delivery of stimulation signals," US Patent 9 289 608, 2016.

[15] M. Parastarfeizabadi, A. Z. Kouzani, I. Gibson, and S. J. Tye, "A miniature closed-loop deep brain stimulation device," in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Institute of Electrical and Electronics Engineers (IEEE), aug 2016.

[16] Y. Su, S. Routhu, K. Moon, S. Lee, W. Youm, and Y. Ozturk, "A wireless 32-channel implantable bidirectional brain machine interface," Sensors, vol. 16, no. 10, sep 2016, p. 1582.

[17] E. Pereda, R. Q. Quiroga, and J. Bhattacharya, "Nonlinear multivariate analysis of neurophysiological signals," Progress in Neurobiology, vol. 77, no. 1-2, sep 2005, pp. 1–37.

[18] R. Falahian, M. Mehdizadeh Dastjerdi, M. Molaie, S. Jafari, and S. Gharibzadeh, "Artificial neural network-based modeling of brain response to flicker light," Nonlinear Dynamics, vol. 81, no. 4, 2015, pp. 1951–1967.

[19] L. Zhang, "Artificial Neural Network Model Design and Topology Analysis for FPGA Implementation of Lorenz Chaotic Generator," in 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Apr. 2017, pp. 216–219.

[20] S. Gu et al., "Controllability of structural brain networks," Nature Communications, vol. 6, oct 2015, p. 8414.

[21] M. Hénon, "A two-dimensional mapping with a strange attractor," Communications in Mathematical Physics, vol. 50, no. 1, Feb 1976, pp. 69–77. [Online]. Available: http://dx.doi.org/10.1007/BF01608556

[22] L. Zhang, "Fixed-point FPGA Model-Based Design and Optimization for Hénon Map Chaotic Generator," in Latin American Symposium of Circuits and Systems (LASCAS), Feb. 2017, pp. 105–108.

[23] L. Zhang, "Hénon Map Chaotic System Analysis and VHDL-based Fixed-point FPGA Implementation for Brain Stimulation," in 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Apr. 2017, pp. 808–811.

[24] L. Zhang, "System Generator Model-based FPGA Design Optimization and Hardware Co-simulation for Lorenz Chaotic Generator," in 2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS 2017), Wuhan, Jun. 2017, pp. 170–174.

[25] L. Zhang, "FPGA Implementation of Fixed-point Neuron Models with Threshold, Ramp and Sigmoid Activation Functions," in 2017 4th International Conference on Mechanics and Mechatronics Research (ICMMR 2017), Xian, Jun. 2017.

[26] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," IEEE Transactions on Neural Networks, vol. 5, no. 6, Nov 1994, pp. 989–993.

[27] F. D. Foresee and M. T. Hagan, "Gauss-newton approximation to bayesian learning," in Neural Networks,1997., International Conference on, vol. 3, Jun 1997, pp. 1930–1935 vol.3.

[28] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," NEURAL NETWORKS, vol. 6, no. 4, 1993, pp. 525–533.

[29] S. Lynch, Dynamical Systems with Applications using MATLAB. Springer International Publishing, 2014.

[30] N. Chau, "Controlling chaos by periodic proportional pulses," Phys. Lett., vol. A 234, 1997, p. 193197.

 ISBN: 978-1-61208-579-1