Tutorial:
Chaotic System Control for Brain Stimulation & FPGA Hardware Implementation

Eur Ing Dr. Lei Zhang
Faculty of Engineering and Applied Science
University of Regina
Canada

The Second International Conference on Neuroscience and Cognitive Brain Information
BRAININFO 2017, July 22, 2017, Nice, France
Outlines

- Chaotic Systems
- Hénon Map Analysis and Control
- Artificial Neural Network Design for Hénon Map
- Artificial Neural Network Design for Lorenz System
- Fixed-point Implementation
- Model and VHDL-based FPGA Design
One Idea and Three Methods

One Idea:
- Chaotic system simulation, analysis and control for pattern recognition of brain activities and brain stimulation.

Three Methods:
- Chaotic systems analysis and control
- Artificial Neural Network (ANN) architecture design and optimization
- FPGA fixed-point hardware implementation
The Idea: Brain Research Program Overview

- **Brain Stimulation**
  - Parkinson’s Disease – tremor
  - Epilepsy – seizure

- **Chaotic Systems**
  - Dynamic Analysis and Control
  - Artificial Neural Network based Model

- **Machine Learning**
  - Feature Extraction of EEG Signals
  - Pattern Recognition and Classification
The Practical Goal: Brain Stimulation

- Electroencephalogram (EEG) uses electrodes attached to the scalp to capture brainwave signals;
- EEG signals captured from brain activities demonstrate chaotic behaviors (bifurcation etc.)
- Brain Stimulation
  - Deep brain stimulation
  - Non-invasive brain stimulation
    Eg. Direct current (tDCS), Electromagnetic, ultrasound
The Challenges and Remedies

- **Challenges**
  - EEG signals are individual dependent and the amount of available data is limited;
  - EEG signals are affected by noise
  - ANN training require big data

- **Remedies**
  - The outputs of chaotic systems are used to train ANN to simulate brain activities
  - FPGA hardware implementation for parallel processing and acceleration
Chaotic Systems

- A chaotic system is a bound system which obtains the existence of attractor.
- Outputs depends on initial values and system parameters;
- Predictability, probability and controllability;
- Examples:
  - 1D – Logistic map, Gaussian map
  - 2D – Hénon map
  - 3D – Lorenz system, Rösseler system
Hénon Map - Definition

Equations by definition:

\[ x_{n+1} = 1 + y_n - ax_n^2 \]
\[ y_{n+1} = bx_n \]

Reformed equations:

\[ x'_n = \frac{1}{\alpha} x_n, \quad y'_n = \frac{\beta}{\alpha} y_n \]
\[ x_{n+1} = \alpha + \beta y_n - x_n^2 \]
\[ y_{n+1} = x_n \]
Hénon Map Analysis

Jacobian Matrix:

\[
J(x_1, y_1) = \begin{pmatrix}
\frac{\partial P}{\partial x} & \frac{\partial P}{\partial y} \\
\frac{\partial Q}{\partial x} & \frac{\partial Q}{\partial y}
\end{pmatrix}
\]

Hénon I:

\[
J = \begin{pmatrix}
-2ax & 1 \\
\beta & 0
\end{pmatrix} = \begin{pmatrix}
-2.4x & 1 \\
0.4 & 0
\end{pmatrix}
\]

\[
Eig(J)_{(x_1=-1.1965)} = \left\{ \begin{array}{l}
\lambda_1 \approx 3.0047 \\
\lambda_2 \approx -0.1331
\end{array} \right.
\]

\[
Eig(J)_{(x_2=0.6965)} = \left\{ \begin{array}{l}
\lambda_1 \approx 0.2123 \\
\lambda_2 \approx -1.8839
\end{array} \right.
\]

Hénon II:

\[
J = \begin{pmatrix}
-2x & \beta \\
1 & 0
\end{pmatrix}
\]

\[
Eig(J)_{(x_1=0.8358)} = \left\{ \begin{array}{l}
\lambda_1 \approx 0.2123 \\
\lambda_2 \approx -1.8839
\end{array} \right.
\]

\[
Eig(J)_{(x_2=-1.4358)} = \left\{ \begin{array}{l}
\lambda_1 \approx 3.0047 \\
\lambda_2 \approx -0.1331
\end{array} \right.
\]

Critical points of period N orbit is stable as long as:

\[|\lambda_1| < 1 \text{ and } |\lambda_2| < 1\]
Hénon Map - Bifurcation

(a) & (c) The bifurcation points (h1 =0) are found at :
α= 0.27 (period one doubling)  
α= 0.85 (period two doubling)  
α = 0.99 (period four doubling)

(b) & (d) The bifurcation points (h1 =1) are found at :
β= 0.265 (period one doubling)  
β = 0.035 (period two doubling)  
β = 0.125 (period four doubling)
Hénon Map Bifurcation 3D

(a) $x$ over $\alpha$ bifurcation with various $\beta$

(b) $x$ over $\beta$ bifurcation with various $\alpha$
Hénon Map Lyapunov Exponents

\[ L(x_0) = \log \left( \text{Eig} \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} (J_i(x_0) \cdot J_i^T(x_0))^\frac{1}{2} \right) \]
Hénon Map Bifurcation Animation

\[ \text{a}=0.2 \sim 1.4, \ b=0.4 \]

\[ \text{a}=1.2, \ b=-0.6 \sim 0.4 \]
ANN Model Design for Chaotic Systems

- An feed forward ANN can be trained using the output values of a chaotic system.
- The training process is carried out on a computer and the weights and bias are generated for all neurons in an ANN architecture.
- The complexity of the ANN architecture defines the implementation cost and speed. Therefore it is beneficial to use less number of hidden neurons to achieve the target training performance.
A Simple Neuron Model

- Inputs
- Weights
- Biases
- Summed Weights
- Activation Function
- Outputs
Artificial Neural Network

\[ a_j^l = \sum_{i=1}^{N_{l-1}} w_{j,i}^l x_i + b_{j,0}^l \quad j = 1, 2, \ldots N_l \]

\[ y_j^l = f_l(a_j^l) \]
ANN Training

- 3 Training Algorithms:
  - Levenberg-Marquardt (LM)
  - Bayesian Regularization (BR)
  - Scaled Conjugate Gradient (SCG)
- 16 Architectures (1 to 16 hidden neurons) for each algorithm
- 3 Training iterations for per architecture per algorithm
ANN Training Performance

- The ANN training result is measured by the error between the calculated output \( y \) and the target training output \( \hat{y} \).
- 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 Mean Squared Error – MSE

\[
MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2
\]
Hénon Map Training Results -LM

![Graph showing ANN training performance with MSE vs. number of hidden neurons for different models I, II, III.](image)
Hénon Map Training Results -BR

![ANN Training Performance BR](image)

The graph shows the ANN training performance for different numbers of hidden neurons of the ANN. The performance is measured by the Mean Squared Error (MSE). The graph indicates that the performance improves as the number of hidden neurons increases.
Hénon Map Training Results-SCG
Hénon Map Training Results

![ANN Training Performance-Average Graph](image-url)
Hénon Map ANN Architecture

Figure 1. ANN Architecture for Hénon Map Chaotic System

Figure 3. Simulink Model for ANN-based Hénon Map Chaotic System
Hénon Map Training Performance
2-hidden neurons LM

Best Validation Performance is 7.6132e-08 at epoch 1000
Hénon Map Training Performance
2-hidden neurons BR

Best Training Performance is $8.2016e-08$ at epoch 1000
Hénon Map Training Performance
2-hidden neurons SCG

Best Validation Performance is 0.0033271 at epoch 249

Gradient = 0.003059, at epoch 255

Validation Checks = 6, at epoch 255
Lorenz Chaotic System

\[
\begin{align*}
\frac{dx}{dt} &= \sigma(y - x) \\
\frac{dy}{dt} &= \rho x - y - xz \\
\frac{dz}{dt} &= -\beta z + xy
\end{align*}
\]
The Lorenz Butterfly (10,20,30)
Lorenz System ANN Model
3x8x3 ANN Architecture
Training Performance – LM
– 8 hidden neurons
Training Performance – BR
– 8 hidden neurons
Training Performance – SCG

– 8 hidden neurons

Best Validation Performance is 0.0040099 at epoch 1000
Best Training Performance- LM
Best Training Performance - BR
Best Training Performance - SCG
Averaged Training Results
Fixed-point Representation

• The range of the signed fixed-point is represented by

\[-(2^{N_i} - 2^{-N_f} + 1) \sim +(2^{N_i} - 2^{-N_f})\]

• where \(N_i\) be the number of integer bits, \(N_f\) be the number of fractional bits. The precision (step size) is \(2^{-N_f}\).
Hénon Map Fixed-point
Hénon Map Fixed-point Analysis

(a) Lyapunov Exponent - Floating Point
(b) Lyapunov Exponent - Fixed-point 8b
(c) Lyapunov Exponent - Fixed-point 6b

(d) Bifurcation - Floating Point
(e) Bifurcation - Fixed-point 8 bit
(f) Bifurcation - Fixed-point 6 bit
Hénon Map Chaotic Control: Periodic Proportional Pulses

(a) Period One

(b) Period Two
Periodic Proportional Pulses
Model-based Hénon Map Design
VHDL Vs Model-Based Designs

<table>
<thead>
<tr>
<th>Zynq 7020</th>
<th>VHDL Based Design I</th>
<th>VHDL Based Design II</th>
<th>Model Based Design$^a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data format</td>
<td>F32.29 F16.13 F16.13</td>
<td>F32.29 F16.13 F16.13</td>
<td>F32.18 F16.13</td>
</tr>
<tr>
<td>Sample period $T_s$</td>
<td>20 ns 20 ns 10 ns</td>
<td>20 ns 20 ns 10 ns</td>
<td>50 ns 20 ns</td>
</tr>
<tr>
<td>Worst Negative Slack</td>
<td>0.03 ns 7.593 ns -2.034 ns</td>
<td>7.45 ns 11.857 ns 2.452 ns</td>
<td>24.37 ns 1.32 ns</td>
</tr>
<tr>
<td>Max Frequency(MHz)</td>
<td>50.08 80.60 –</td>
<td>79.68 122.80 132.49</td>
<td>39.01 53.53</td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>172 16 16</td>
<td>123 16 16</td>
<td>366 150</td>
</tr>
<tr>
<td>No. of Registers</td>
<td>64 16 16</td>
<td>64 32 32</td>
<td>64 32</td>
</tr>
<tr>
<td>No. of Slices</td>
<td>44 4 5</td>
<td>40 10 10</td>
<td>126 56</td>
</tr>
<tr>
<td>No. of DSP</td>
<td>12 3 3</td>
<td>8 2 2</td>
<td>4 1</td>
</tr>
<tr>
<td>Total On-chip Power(W)</td>
<td>0.16 0.138 0.156</td>
<td>0.158 0.138 0.155</td>
<td>0.153 0.154</td>
</tr>
</tbody>
</table>

$$f_{max} = \frac{1}{T_s - WN S}$$

Design I : 3 multipliers; Design II: 2 multipliers; FPGA DSP: 18x18
Summary

One Idea

• Brain stimulation based on Chaotic systems simulation and Artificial Neural Network Design

Three Methods

• Chaotic systems analysis and control
• Artificial Neural Network (ANN) architecture design and optimization
• FPGA fixed-point hardware implementation
Q and A

Thank you!