THE PERFORMANCE COMPARISON OF CHAOTIC TIME SERIES USING NONLINEAR AUTOREGRESSIVE NETWORK WITH EXOGENOUS INPUT TECHNIQUE

Peerapol Phutthawetmongkhong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Engineering Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2017

Thesis Title	The Performance Comparison of Chaotic Time Series using Nonlinear Autoregressive Network with Exogenous Input Technique
By	Peerapol Phutthawetmongkhong
Field of Study	Engineering Technology
Advisor	Asst. Prof. Dr. Wimol San-Um

The Graduate School of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirements for the Master's Degree

.... Dean of Graduate School
(Assoc. Prof. Dr. Pichit Sukchareonpong)
Month………. Date………, Year……….

Thesis Committees

.... Chairperson
(Asst. Prof. Dr. Surapong Pongyupinpanich)

.... Committee
(Asst. Prof. Dr. Wipawadee Wongsuwan)

.... Committee
(Asst. Prof. Dr. Warakorn Srichavengsup)

.... Advisor
(Asst. Prof. Dr. Wimol San-Um)

PEERAPOL PHUTTHAWETMONGKHONG. THE PERFORMANCE COMPARISON OF CHAOTIC TIME SERIES USING NONLINEAR AUTOREGRESSIVE NETWORK WITH EXOGENOUS INPUT TECHNIQUE. ADVISOR: ASST. PROF. DR. WIMOL SAN-UM, 67 PP.

This thesis proposes the performance comparison of chaotic time series using nonlinear autoregressive network with exogenous input technique (NARX). The NARX model is trained by Levenberg-Maquardt Algorithm. The used transfer functions in this thesis are Hyperbolic Tangent Sigmoid, Log-Sigmoid and Radial Basis. The chaotic time series data are generated by chaotic equations. The configuration delays are 2 and 4 to compare the results and find the optimal configuration value for create NARX model using the simulate data. Data were then analyzed as a data set using NARX model.

Graduate School Student's signature ……………............

Field of Study Engineering Technology Advisor's signature……………………...

Academic Year 2017

Acknowledgements

The author wishes to express his gratitude and respectfully dedicate his work to his parent and his family for endless encouragements and love. The author is most grateful to his supervisor, Asst. Prof. Dr. Wimol San-Um for his valuable supervision, encouragements, support, and chances throughout the study. Additionally, grateful acknowledgements are made to Dr. Surapong Pongyupinpanich, Asst. Prof. Dr. Wipawadee Wongsuwan, and Asst. Prof. Dr. Warakorn Srichavengsup members of thesis committee, for their valuable suggestions and comments. The author also acknowledges the Intelligent Electronic Systems Research Laboratory and Academic Services Division of Thai-Nichi Institute of Technology for technical and financial supports.

Peerapol Phutthawetmongkhong

Table of Contents

Table of Contents (Cont.)

List of Table

List of Table (Cont.)

TNI THAI - NICHI INSTITUTE OF TECHNOLOGY สถาบันเทคโนโลยีไทย-ญี่ปุ่น

List of Figures

List of Figures (Cont.)

Chapter 1
Introduction

1.1 Introduction

Chapter 1 discusses the causes and origins of the research process, comparison performance of Nonlinear Autoregressive Network with Exogenous Inputs Techniques (NARX), Inspiration for research, Importance of the problem research, objectives, expected result and technical terms description.

1.2 Backgrounds

The many problem in Thailand are related Time Series Data. The one of solution is predicting event or information for planning or avoid undesirable event.

This thesis proposes the comparison performance of time series data using NARX. The chaotic time series is nonlinear data such as weather, load energy or communication.

1.3 Motivations

In year 2016, the worst traffic in the world is in Bangkok, Thailand. People in Bangkok have extremely hasty lifestyle, especially in transportation. However, Bangkok traffic problem have not been resolved for a long time. People in Bangkok attempt to avoid traffic jam, whereas the traffic cannot be always accurately expected even with the use of current applications to view traffic data at that time. The application cannot accurately analyze traffic flow for user satisfaction include cannot predict traffic flow. The research initiates idea for increasing accurately traffic flow analytics system and predict traffic flow using NARX to improve the efficiency of analytics in order that people can predict traffic flow and transport plans or avoid the crowded route properly. The researcher starts from using chaotic time series instead of traffic flow to create NARX.

1.4 Statement of Problem and Hypothesis

The traffic flow analytics in Thailand is high tolerance. This makes commercial and personal applications unsatisfactory for some users. The researcher initiates from prediction process in this thesis for using to predict a traffic flow in the future.

1.5 Objectives

1.5.1 To study Chaotic Time Series

1.5.2 To study Nonlinear Autoregressive Network with Exogenous Inputs Techniques and Prediction Analysis

1.5.3 To study Hyperbolic Tangent Sigmoid Transfer Function, Log-Sigmoid Transfer Function and Radial Basis Transfer Function

1.6 Research Scopes

1.6.1 Prediction of Chaotic Time Series using NARX on MATLAB.

1.6.2 Comparison Hyperbolic Tangent Sigmoid Transfer Function, Log-Sigmoid Transfer Function and Radial Basis Transfer Function

1.7 Expected Outcomes

1.7.1 Gained knowledge on Chaotic Time Series.

1.7.2 Gained knowledge on NARX and Prediction Process.

1.8 Definitions

1.8.1 NARX is a nonlinear autoregressive network with exogenous input technique. This means the model relates the current value of a time series to both past values of the same series and current and exogenous series.

1.8.2 Chaotic Time Series are ubiquitous in nature such as the tornado, stock market, turbulence, and weather. Their functions are different in different situations. For example, in the case of tornado, the chaotic behavior is harmful to human beings and need to be avoided or controlled. But in the case of the activities in human brain, the chaotic behaviors are useful and necessary to sustain the normal functions of brain. Thus, it is an important task to understand chaos and let it serve human society better.

1.8.3 MATLAB is a computer program for computing mathematical operations from basic mathematics such as adding or subtracting to advanced mathematics such as image processing or artificial intelligence. MATLAB stands for Matrix Laboratory and is originally written for matrix computations. Each of mathematic operations in MATLAB is written in MATLAB a command which is simple mathematic signs, and advanced MATLAB commands are written in human language. Thus, MATLAB commands are easy to understand which makes MATLAB widely used in researches. Apart from mathematical computations, MATLAB can also be used as the computer programming editor using high-level programming language which is the MATLAB commands combined together as a script for the computer program. Unfortunately, MATLAB is not capable of generating the standalone computer program from the MATLAB codes without specified plugin, and the MATLAB codes are runnable only on MATLAB. Consequently, the MATLAB codes from MATLAB have to be written in another programming language again to create the standalone computer program.

1.8.4 Transfer Function is a representation in terms of spatial or temporal frequency, of the relation between the input and output of a linear time-invariant (LTI) system with zero initial conditions and zero-point equilibrium. For optical imaging devices, for example, the optical transfer function is the Fourier transform of the point spread function (hence a function of spatial frequency) i.e., the intensity distribution caused by a point object in the field of view. A number of sources however use "transfer function" to mean some input-output characteristic in direct physical measures rather than its transform to the s-plane.

Chapter 2

Related Theories and Literature Reviews

2.1 Introduction

This chapter describes information of related theories including Mackey Glass Equation, Chua's Circuit, Lorenz Equation, Chaotic Map, NARX Model and Levenberg-Marquardt Algorithm. The literature reviews of related research on NARX.

2.2 Related Theory

2.1.1 Mackey Glass Equation

Mackey Glass Equation [1] is nonlinear time delay differential equation. This Mackey Glass Equation displays a range of periodic and chaotic dynamics which compatible with simulate fluctuate data and nonlinear data. This equation use to simulate traffic flow data. Mackey Glass equation can be described mathematically as

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^{10}} - bx(t) \tag{2.1}$$

where a is real numbers, and $(t-\tau)$ is value at x time. Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics.

2.1.2 Chua's Circuit

Chua's Circuit [2] is simplest electronic circuit exhibiting classic chaos theory behavior. It produces oscillating waveform. The ease of construction of the circuit has made it a ubiquitous real-world example of a chaotic system.

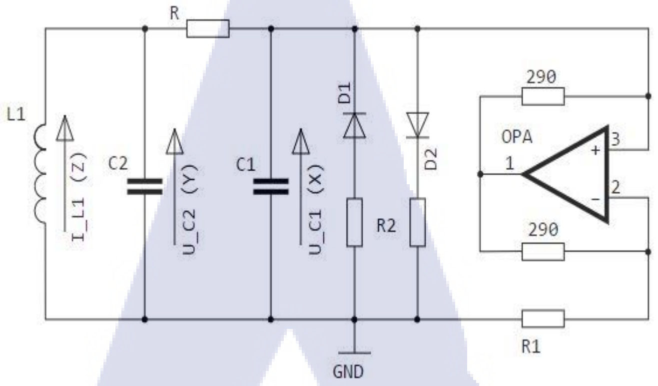


Figure 2.1 One version of Chua's Circuit

Analyzing the circuit using Kirchhoff's circuit laws, the dynamics of Chua's circuit can be accurately modeled by means of a system of three nonlinear ordinary differential equations in the variables $x(t)$, $y(t)$ and $z(t)$, which represent the voltages across the capacitors C_1 and C_2, and the electric current in the inductor L_1 , respectively. These equations are

$$\frac{dx}{dt} = [y - x - f(x)] \tag{2.2}$$

$$RC_2 \frac{dy}{dt} = x - y + Rz \tag{2.3}$$

$$\frac{dz}{dt} = -\beta y \tag{2.4}$$

The function f(x) describes the electrical response of the nonlinear resistor, and its shape depends on the particular configuration of its components. The parameters α and β are determined by the particular values of the circuit components.

2.1.3 Lorenz Equation

The Lorenz system [3] is a system of ordinary differential equations first studied by Edward Lorenz. It is notable for having chaotic solutions for certain parameter values and initial conditions. In particular, the Lorenz attractor is a set of chaotic solutions of the Lorenz system which, when plotted, resemble a butterfly or figure eight. The model is a system of three ordinary differential equations now known as the Lorenz equations can be described mathematically as

$$\dot{x} = \sigma(y - x) \tag{2.5}$$

$$\dot{y} = x(\rho - z) - y \tag{2.6}$$

$$\dot{z} = xy - \beta z \tag{2.7}$$

where σ, ρ and β are positive. Lorenz used the values $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$.

2.1.4 Chaotic Map

The system of the quadratic map [4] is chaotic because it has the following characteristics. It is nonlinear. The quadratic map can be synchronized through coupling.

2.1.5 NARX Model

Artificial Neural Network [5] is mathematical model apply to evaluate information with connectionist technique. This technique emulate from human brain. NARX [6] is a nonlinear autoregressive model with exogenous inputs for artificial neural network learning with loopback. This technique increases the accuracy for learning and prediction. The defining equation for the NARX model is

$$y(t) = f\left(y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u)\right) \tag{2.8}$$

where the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal. You can implement the NARX model by using a feed forward neural network to approximate the function f. A diagram of the resulting network is shown below, where a two-layer feed forward network is used for the approximation. This implementation also allows for a vector ARX model, where the input and output can be multidimensional.

There are many applications for the NARX network. It can be used as a predictor, to predict the next value of the input signal. It can also be used for nonlinear filtering, in which the target output is a noise-free version of the input signal. The use of the NARX network is shown in another important application, the modeling of nonlinear dynamic systems.

Before showing the training of the NARX network, an important configuration that is useful in training needs explanation. It should be considered that the outputs of the NARX network to be an estimate of the output of some nonlinear dynamic system that you are trying to model. The output is feedback to the input of the feed forward neural network as part of the standard NARX architecture, as shown in the left figure below. Because the true output is available during the training of the network, you could create a series-parallel architecture, in which the true output is used instead of feeding back the estimated output, as shown in the right figure below. This has two advantages. The first is that the input to the feed forward network is more accurate. The second is that the resulting network has a purely feed forward architecture, and static back propagation can be used for training.

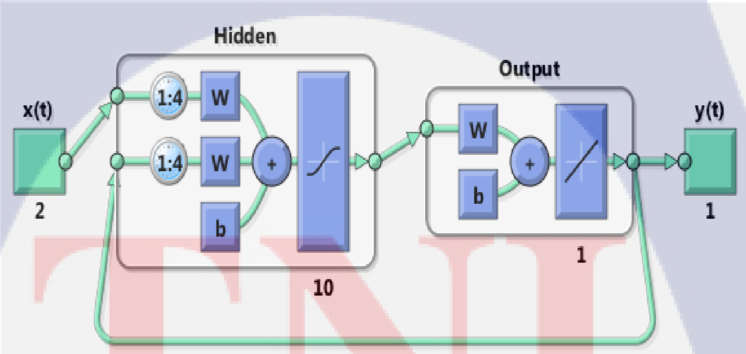


Figure 2.2 Structure of NARX Closed Loop

2.1.6 Levenberg-Marquardt Algorithm

In mathematics and computing, the Levenberg-Marquardt algorithm [7] (LMA or just LM), also known as the damped least-squares (DLS) method, is used to solve non-linear least squares problems. These minimization problems arise especially in least squares curve fitting.

The LMA is used in many software applications for solving generic curve-fitting problems. However, as with many fitting algorithms, the LMA finds only a local minimum, which is not necessarily the global minimum. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach.

2.1.7 Hyperbolic Tangent Sigmoid Transfer Function

The Hyperbolic Tangent [8] defined in terms of exponential function as

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}} = \frac{e^{2x} + 1}{e^{2x} - 1} = \frac{1 - e^{-2x}}{1 - e^{-2x}}, x \neq 0 \tag{2.9}$$

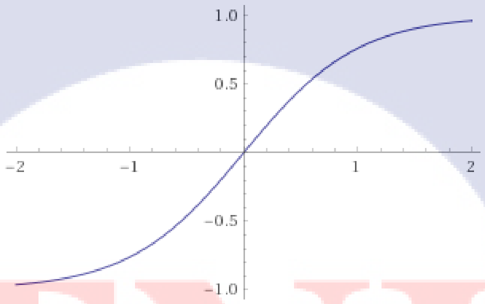


Figure 2.3 Hyperbolic Tangent Sigmoid Transfer Function Graph

2.1.8 Log-Sigmoid Transfer Function

A log-sigmoid function [9], also known as a logistic function, is given by the relationship as

$$\sigma(t) = \frac{1}{1 + e^{-\beta t}} \tag{2.10}$$

where β is a slope parameter. This is called the log-sigmoid because a sigmoid can also be constructed using the hyperbolic tangent function instead of this relation, in which case it would be called a tan-sigmoid. The sigmoid has the property of being similar to the step function, but with the addition of a region of uncertainty. Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons, although not exactly the same. Fig 2.4 is the graph of a sigmoid function.

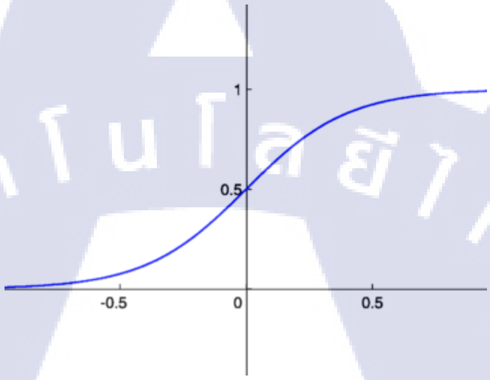


Figure 2.4 Log-Sigmoid Transfer Function Graph

Sigmoid functions are also prized because their derivatives are easy to calculate, which is helpful for calculating the weight updates in certain training algorithms. The derivative when $\beta = 1$ is given by

$$\frac{d\sigma(t)}{dt} = \sigma(t)[1 - \sigma(t)] \tag{2.11}$$

when $\beta \neq 1$, using $\sigma(\beta, t) = \frac{1}{1 + e^{-\beta t}}$, the derivative is given by

$$\frac{d\sigma(\beta, t)}{dt} = \beta[\sigma(\beta, t)[1 - \sigma(\beta, t)]] \tag{2.12}$$

TNI
THAI - NICHI INSTITUTE OF TECHNOLOGY

2.1.9 Radial Basis Transfer Function

Radial basis function (RBF) [10] networks typically have three layers. The three layers are an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The input can be modeled as a vector of real numbers. The output of the network given by

$$\varphi(x) = \sum_{i=1}^{N} a_i \rho(\| x - c_i \|) \tag{2.13}$$

where N is the number of neurons in the hidden layer, C_i is the center vector for neuron i, and a_i is the weight of neuron i in the linear output neuron. Functions that depend only on the distance from a center vector are radially symmetric about that vector, hence the name radial basis function. In the basic form all inputs are connected to each hidden neuron.

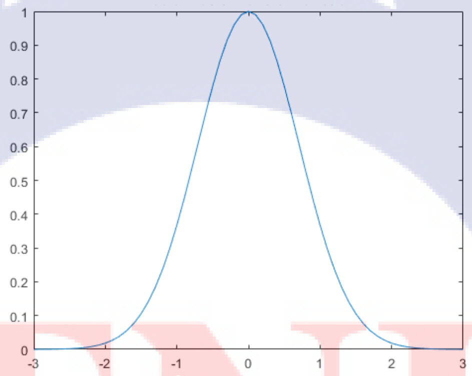


Figure 2.5 Radial Basis Transfer Function Graph

2.3 Literature Reviews

Table 2.1 Summary of related researches to the proposed approaches

Author	Year	Proposed Schemes
A. Thakur et al. [11]	2016	NARX Based Forecasting of Petrol Prices
L. Banjanović-Mehmedović et al. [12]	2016	Prediction of Cooperative Platooning Maneuvers Using NARX
S. Jaiswal et al. [13]	2016	Modeling the Measurement Error of Energy Meter Using NARX Model
L. Zhang et al. [14]	2016	NARX models for predicting power consumption of a horizontal axis wind turbine
F. Chang et al. [15]	2016	Estimating spatio-temporal dynamics of stream total phosphate concentration by soft computing techniques
L.K. Torres-Faurrieta et al. [16]	2016	Recruitment forecasting of yellow fin tuna in the eastern Pacific Ocean with artificial neuronal networks
A.G.R. Vaz et al. [17]	2016	An artificial neural network to assess the impact of neighboring photovoltaic systems in power forecasting in Utrecht, the Netherlands
Y. Chunshan and L.Xiaofeng [18]	2015	Study and Application of Data Mining and NARX Neural Networks in Load Forecasting
H. He et al. [19]	2015	Meridian ECG Information Transmission System Modeling Using NARX Neural Network
X. Zhe et al. [20]	2015	Water Distribution Network Modeling Based on NARX

A. Thakur et al. [11] presents the expectations of gasoline prices with dynamic neural imaging theory, with automated regression techniques. Accuracy in gasoline pricing is important in maintaining balance for consumer demand and oil producers.

L. Banjanović-Mehmedović et al. [12] offers an intelligent transport system that improves traffic agility. In this research, additional systems were developed to improve the predictive accuracy of the system by nonlinear autonomic neural network techniques with external input data.

S. Jaiswal et al. [13] presented a nonlinear regression model of the metrological error measurement with nonlinear automatic neural network techniques with four external inputs.

L. Zhang et al. [14] presents a linear energy prediction of horizontal axis wind turbine using nonlinear automata regression model with external input data. By dividing the model into two, the first uses two data inputs. The second uses three external input data. The result is the second model has the first high precision results.

F. Chang et al. [15] proposed an evaluation of the phosphate mineralization in streams by artificial intelligence techniques. The artificial intelligence technique used in this research is the nonlinear automatic regression neural network technique with external input data. Data used for river data in Taiwan with 10 years of water quality data collection. The results are based on nonlinear automatic linear regression techniques with external input data that accurately and efficiently detect the data.

L.K. Torres-Faurrieta et al. [16] proposed a prediction of yellowfin tuna in the eastern Pacific using nonlinear autoregressive neural network techniques with external input data. By dividing the models into two, the results show that the type of input fed to the neural network influences the prediction of fish and accuracy.

A.G.R. Vaz et al. [17] proposed the use of Neural Networks' neural network assessment system and its vicinity to anticipating the use of electricity and produce sufficiently for precise use. The result is a mean square error of 9 percent to 25 percent and a prediction of more than 15 minutes.

Y. Chunshan and L. Xiaofeng [18] present data mining relationships for predicting electricity demand. The use of data mining in the analysis of China's electricity use factors indicates that there are some key indicators and that the automatic neural network registers are not linear with external inputs. The results show a more accurate prediction model.

H. He et al. [19] presented an analysis of data on cardiopulmonary resections from 10 linear regression lines using nonlinear autonomic neural network technique with better external input data than network technique. Automatic linear is retracting artificial neural network with external input data. Signs of cardiac catheterization from the meridians are accurate to 0.98.

X. Zhe et al. [20] proposed the use of nonlinear automatic linear neural network techniques with external input data for predicting and controlling water distribution. Input data used is current and past data. The results from deploying to the real network are satisfying, easy to track and predict.

2.4 Conclusion

This chapter discusses the relevant theoretical information presented in this thesis consist of prediction and forecasting using NARX and training NARX. The literature reviews of 10 literatures related to the proposed NARX approaches were also included.

Chapter 3
Research Methodology

3.1 Introduction

This chapter describes research methodology of this thesis, involving research process, data collection, and research tools.

3.2 Research Process

3.2.1 Study the chaotic time series

3.2.2 Study the NARX

3.2.3 Simulate the time series data from equation

3.2.4 Collect the traffic volume

3.2.5 Training and prediction using NARX

3.3 Data Collection

The data in this thesis are simulating from chaotic time series equation and collect real traffic volume data from Department of Highways (DOH).

3.4 Research Tools

3.4.1 MATLAB R2011a

3.5 Conclusion

This chapter has presented research methodology of this thesis, including research process data collection, and research tools.

Chapter 4
Experiment Results

4.1 Introduction

This chapter proposes the prediction of simulate data set using NARX. The simulate data set generate by Mackey Glass Equation, Realistic Chua's Circuit, Lorenz Equation and Chaotic Map. MSE Result from each data set were analyzed and predicted with NARX on MATLAB.

4.2 Simulate Data Set

4.2.1 Mackey Glass Equation

Mackey Glass Equation using 4^{th}-Order Runge-Kutta method. This method decide step size is 1, τ is 17, simulate data set is 1000 and initial condition is 1.2.

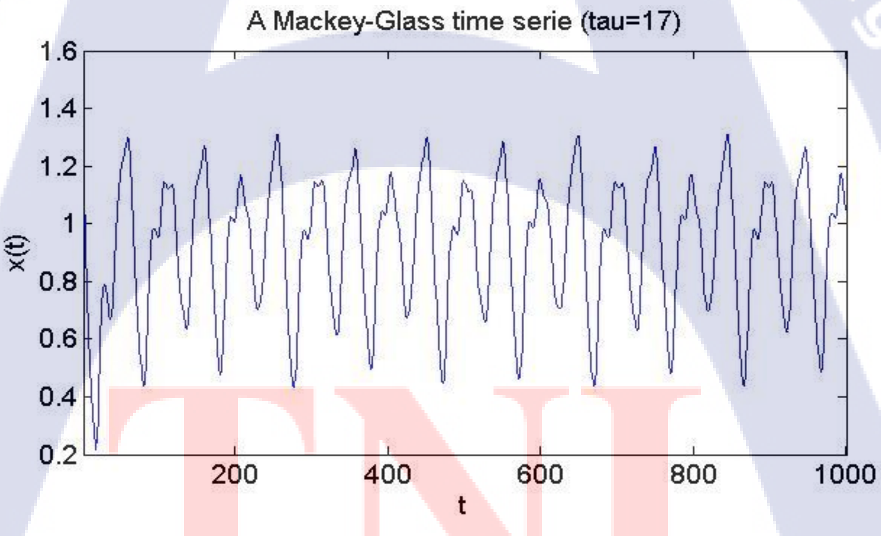


Figure 4.1 Simulated time series of Mackey Glass Equation

4.2.2 Chua's Circuit

Realistic Chua's Circuit is configured resistors are R_1=220 Ω, R_2=220 Ω, R_3=2200 Ω, R_4=22000 Ω, R_5=22000 Ω, R_6=3300 Ω, R_7=100 Ω, R_8=1000 Ω, R_9=1000 Ω, R_{10}=1800 Ω, R=1800 Ω, C=100nF, C_1=10nF and C_2=100nF

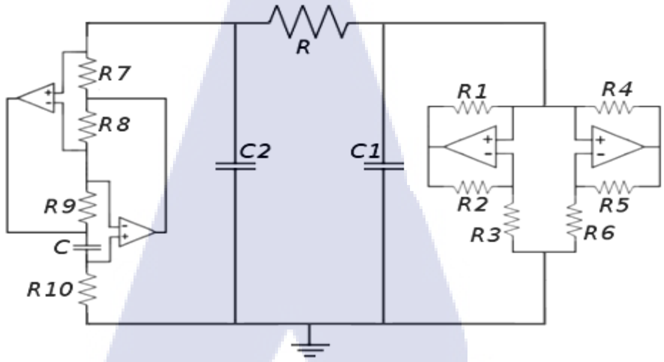


Figure 4.2 Schematics of the simulated circuit

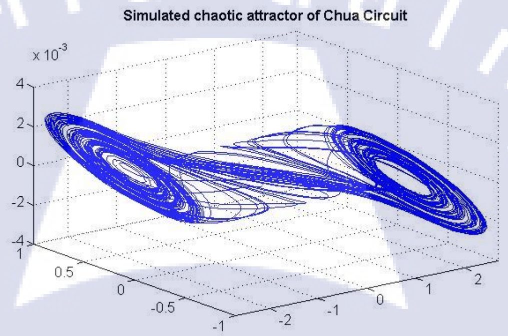


Figure 4.3 Simulated chaotic attractor of Chua Circuit

4.2.3 Lorenz Equation

Lorenz Equation using the values $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$.

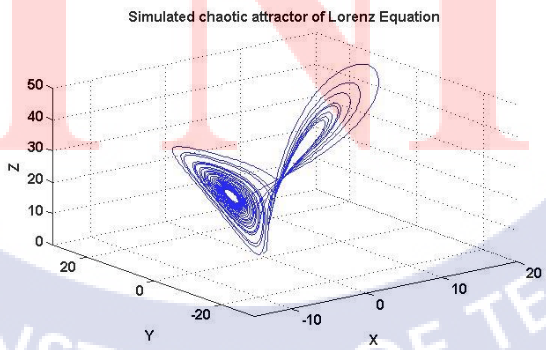


Figure 4.4 Simulated chaotic attractor of Lorenz Equation

4.2.4 Chaotic Map

Chaotic Map is generated with default value in MATLAB.

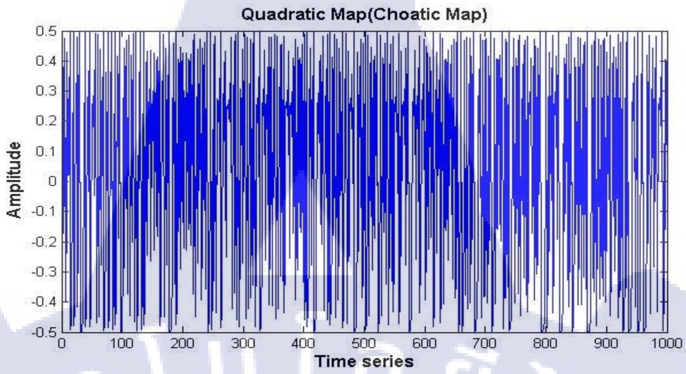


Figure 4.5 Simulated time series of Chaotic Map

4.3 Training NARX Model

Training NARX Model involve configure weight, number of hidden neurons, input and delays. The accuracy base on a configure value. This paper use simulate data sets input NARX by Network Time Series Tool in MATLAB. These simulate data sets split 3 set. Training set is 70%, validation set is 15% and testing set is 15%.

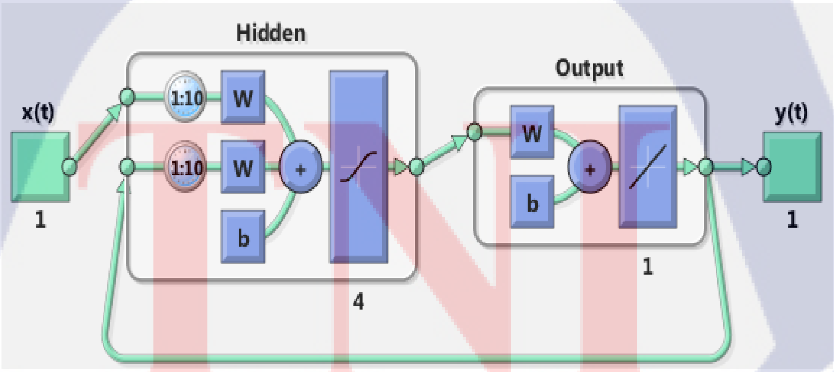


Figure 4.6 Structure of NARX Closed Loop Model

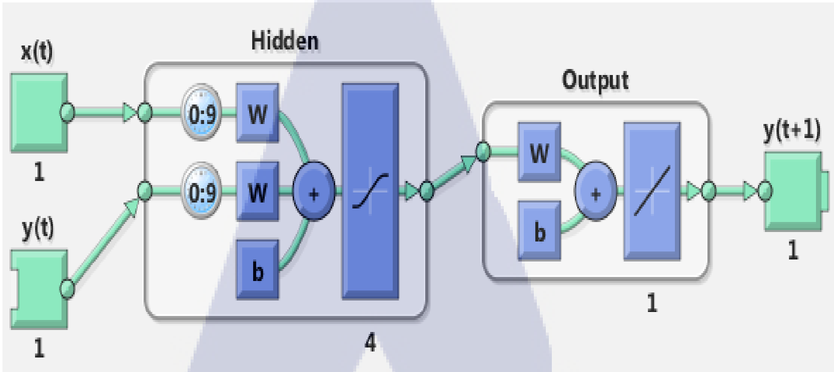


Figure 4.7 Structure of NARX to Predict One Step Ahead Model

4.4 Prediction Result

4.4.1 Hyperbolic Tangent Sigmoid Transfer Function

The results are divided into 5 groups by simulate data set. The summary of MSE results from processing through the NARX model with different number of delays and number of hidden neurons. The summary result shown in Table 4.1, 4.2, 4.3, 4.4, 4.5 and Fig.4.8, 4.9, 4.10, 4.11, 4.12, 4.13

Table 4.1 Prediction Result of simulated data from Mackey Glass Equation using Hyperbolic Tangent Sigmoid Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Mackey Glass Equation	Hyperbolic Tangent Sigmoid Transfer Function	2	1	3.71×10^{-5}
			2	3.74×10^{-5}
			3	3.98×10^{-5}
			4	3.14×10^{-5}
			5	2.47×10^{-5}
			6	2.26×10^{-5}
			7	3.04×10^{-5}
			8	2.65×10^{-5}
			9	2.81×10^{-5}
			10	2.48×10^{-5}
		4	1	7.81×10^{-6}
			2	6.93×10^{-6}
			3	6.82×10^{-6}
			4	1.12×10^{-5}
			5	4.06×10^{-6}
			6	5.28×10^{-6}
			7	4.69×10^{-6}
			8	5.37×10^{-6}
			9	4.81×10^{-6}
			10	5.04×10^{-7}

Table 4.2 Prediction Result of simulated data from Chua's Circuit using Hyperbolic Tangent Sigmoid Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Chua's Circuit	Hyperbolic Tangent Sigmoid Transfer Function	2	1	0.07
			2	1.43×10^{-4}
			3	1.41×10^{-4}
			4	1.27×10^{-4}
			5	1.14×10^{-4}
			6	1.17×10^{-4}
			7	1.02×10^{-4}
			8	1.05×10^{-4}
			9	9.82×10^{-5}
			10	7.67×10^{-5}
		4	1	0.07
			2	1.13×10^{-4}
			3	1.11×10^{-4}
			4	1.01×10^{-4}
			5	1.14×10^{-4}
			6	9.12×10^{-5}
			7	1.08×10^{-4}
			8	8.21×10^{-5}
			9	9.69×10^{-5}
			10	8.15×10^{-5}

Table 4.3 Prediction Result of simulated data from Lorenz Equation using Hyperbolic Tangent Sigmoid Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Lorenz Equation	Hyperbolic Tangent Sigmoid Transfer Function	2	1	24.68
			2	2.75
			3	2.10×10^{-4}
			4	1.88×10^{-4}
			5	1.62×10^{-4}
			6	1.60×10^{-4}
			7	1.55×10^{-4}
			8	1.59×10^{-4}
			9	1.46×10^{-4}
			10	1.46×10^{-4}
		4	1	24.57
			2	2.75
			3	1.92×10^{-4}
			4	1.79×10^{-4}
			5	1.72×10^{-4}
			6	1.54×10^{-4}
			7	1.59×10^{-4}
			8	1.57×10^{-4}
			9	1.54×10^{-4}
			10	1.42×10^{-4}

Table 4.4 Prediction Result of simulated data from Chaotic Map using Hyperbolic Tangent Sigmoid Function

<table>
<tr><th rowspan="2">Simulated Data Set</th><th rowspan="2">Transfer Function</th><th colspan="3">Experiment Results</th></tr>
<tr><th>Number of Delays</th><th>Number of Hidden Neurons</th><th>MSE</th></tr>
<tr><td rowspan="20">Chaotic Map</td><td rowspan="20">Hyperbolic Tangent Sigmoid Transfer Function</td><td rowspan="10">2</td><td>1</td><td>0.08</td></tr>
<tr><td>2</td><td>3.47×10^{-7}</td></tr>
<tr><td>3</td><td>2.09×10^{-7}</td></tr>
<tr><td>4</td><td>9.31×10^{-8}</td></tr>
<tr><td>5</td><td>2.69×10^{-9}</td></tr>
<tr><td>6</td><td>2.63×10^{-7}</td></tr>
<tr><td>7</td><td>3.19×10^{-8}</td></tr>
<tr><td>8</td><td>8.91×10^{-9}</td></tr>
<tr><td>9</td><td>1.21×10^{-8}</td></tr>
<tr><td>10</td><td>1.95×10^{-10}</td></tr>
<tr><td rowspan="10">4</td><td>1</td><td>0.12</td></tr>
<tr><td>2</td><td>0.05</td></tr>
<tr><td>3</td><td>3.45×10^{-7}</td></tr>
<tr><td>4</td><td>9.35×10^{-8}</td></tr>
<tr><td>5</td><td>3.31×10^{-7}</td></tr>
<tr><td>6</td><td>1.54×10^{-7}</td></tr>
<tr><td>7</td><td>1.44×10^{-7}</td></tr>
<tr><td>8</td><td>1.70×10^{-7}</td></tr>
<tr><td>9</td><td>4.81×10^{-8}</td></tr>
<tr><td>10</td><td>5.33×10^{-9}</td></tr>
</table>

Table 4.5 Prediction Result of real data from Collection using Hyperbolic Tangent Sigmoid Function

Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Department of Highways	Hyperbolic Tangent Sigmoid Transfer Function	2	1	1.79×10^{4}
			2	1.79×10^{4}
			3	2.05×10^{4}
			4	2.35×10^{4}
			5	1.87×10^{4}
			6	2.00×10^{4}
			7	1.88×10^{4}
			8	2.00×10^{4}
			9	2.61×10^{4}
			10	1.62×10^{4}
		4	1	1.86×10^{4}
			2	1.75×10^{4}
			3	1.40×10^{4}
			4	1.95×10^{6}
			5	2.10×10^{4}
			6	1.81×10^{4}
			7	1.91×10^{4}
			8	2.62×10^{4}
			9	1.82×10^{4}
			10	4.35×10^{5}

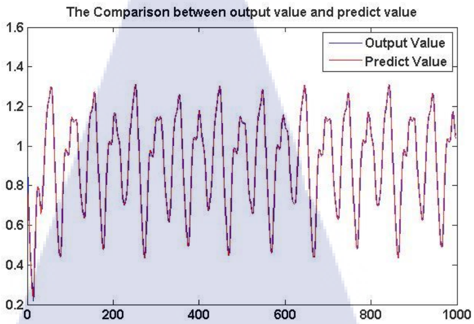


Figure 4.8 The Comparison between output values and predicted values from Mackey Glass Equation using Hyperbolic Tangent Sigmoid Transfer Function

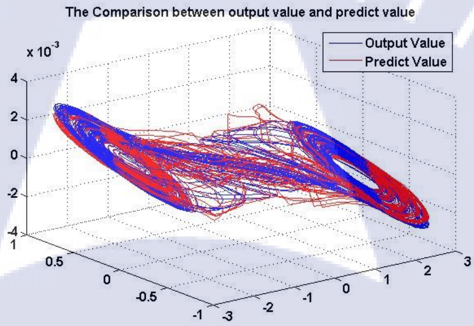


Figure 4.9 The Comparison between output values and predicted values from Chua's Circuit using Hyperbolic Tangent Sigmoid Transfer Function

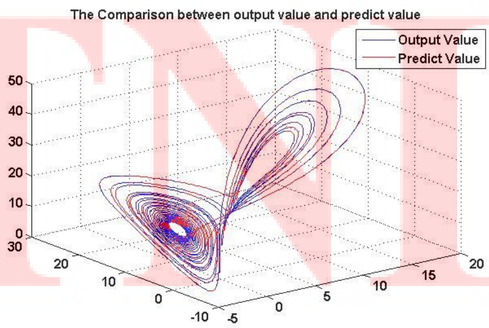


Figure 4.10 The Comparison between output values and predicted values from Lorenz Equation using Hyperbolic Tangent Sigmoid Transfer Function

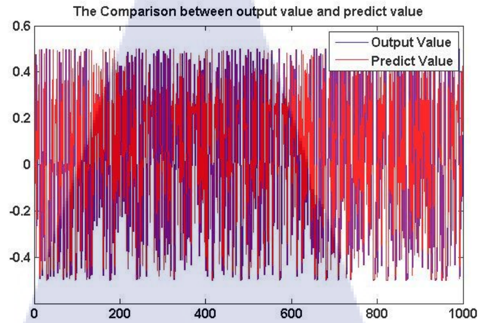


Figure 4.11 The Comparison between output values and predicted values from Chaotic Map using Hyperbolic Tangent Sigmoid Transfer Function

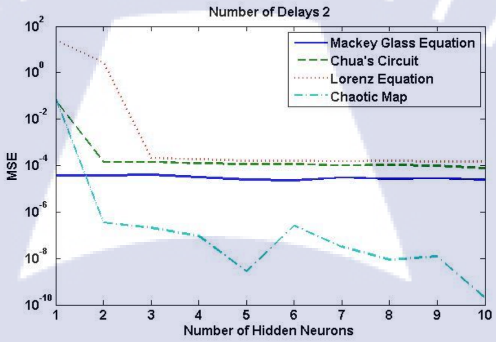


Figure 4.12 Result Experiment by Number of Delays 2 using Hyperbolic Tangent Sigmoid Transfer Function

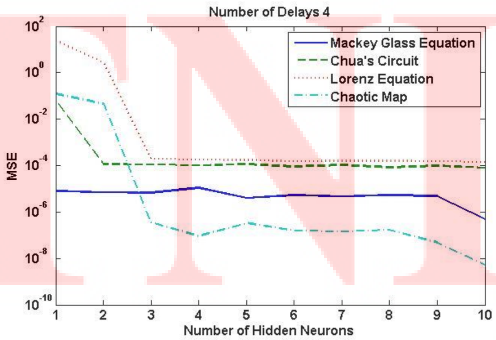


Figure 4.13 Result Experiment by Number of Delays 4 using Hyperbolic Tangent Sigmoid Transfer Function

4.4.2 Log-Sigmoid Transfer Function

The summary result shown in Table 4.6, 4.7, 4.8, 4.9, 4.10 and Fig.4.14, 4.15, 4.16, 4.17, 4.18, 4.19

Table 4.6 Prediction Result of simulated data from Mackey Glass using Log-Sigmoid Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Mackey Glass Equation	Log-Sigmoid Transfer Function	2	1	3.73×10^{-5}
			2	3.69×10^{-5}
			3	3.58×10^{-5}
			4	2.53×10^{-5}
			5	2.38×10^{-5}
			6	2.40×10^{-5}
			7	2.88×10^{-5}
			8	2.66×10^{-5}
			9	2.45×10^{-5}
			10	2.99×10^{-5}
		4	1	8.03×10^{-6}
			2	6.56×10^{-6}
			3	3.16×10^{-6}
			4	3.68×10^{-6}
			5	4.09×10^{-6}
			6	3.61×10^{-6}
			7	4.54×10^{-6}
			8	3.95×10^{-6}
			9	3.67×10^{-6}
			10	1.90×10^{-6}

Table 4.7 Prediction Result of simulated data from Chua's Circuit using Log-Sigmoid Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Chua's Circuit	Log-Sigmoid Transfer Function	2	1	0.07
			2	1.43×10^{-4}
			3	1.23×10^{-4}
			4	1.24×10^{-4}
			5	9.78×10^{-5}
			6	8.70×10^{-5}
			7	1.04×10^{-4}
			8	1.03×10^{-4}
			9	9.93×10^{-5}
			10	1.00×10^{-4}
		4	1	0.07
			2	1.13×10^{-4}
			3	1.04×10^{-4}
			4	1.10×10^{-4}
			5	1.09×10^{-4}
			6	1.10×10^{-4}
			7	8.44×10^{-5}
			8	9.70×10^{-5}
			9	7.31×10^{-5}
			10	8.53×10^{-5}

TNI
THAI-NICHI INSTITUTE OF TECHNOLOGY

Table 4.8 Prediction Result of simulated data from Lorenz Equation using Log-Sigmoid Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Lorenz Equation	Log-Sigmoid Transfer Function	2	1	24.63
			2	2.75
			3	2.06×10^{-4}
			4	1.81×10^{-4}
			5	1.67×10^{-4}
			6	1.60×10^{-4}
			7	1.53×10^{-4}
			8	1.57×10^{-4}
			9	1.50×10^{-4}
			10	1.50×10^{-4}
		4	1	24.57
			2	2.75
			3	2.75
			4	1.82×10^{-4}
			5	1.53×10^{-4}
			6	1.59×10^{-4}
			7	1.46×10^{-4}
			8	1.56×10^{-4}
			9	1.55×10^{-4}
			10	1.38×10^{-4}

Table 4.9 Prediction Result of simulated data from Chaotic Map using Log-Sigmoid Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Chaotic Map	Log-Sigmoid Transfer Function	2	1	0.08
			2	7.52×10^{-7}
			3	1.50×10^{-9}
			4	2.58×10^{-8}
			5	2.38×10^{-7}
			6	1.27×10^{-6}
			7	8.44×10^{-10}
			8	7.37×10^{-8}
			9	4.37×10^{-10}
			10	2.50×10^{-8}
		4	1	0.08
			2	0.11
			3	3.10×10^{-8}
			4	2.30×10^{-6}
			5	4.86×10^{-6}
			6	4.05×10^{-7}
			7	6.37×10^{-7}
			8	2.26×10^{-6}
			9	2.79×10^{-6}
			10	4.80×10^{-7}

Table 4.10 Prediction Result of real data from Collection using Log-Sigmoid Transfer Function

Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Department of Highways	Log-Sigmoid Transfer Function	2	1	1.80×10^{4}
			2	5.51×10^{4}
			3	1.85×10^{4}
			4	2.88×10^{4}
			5	1.82×10^{4}
			6	8.99×10^{4}
			7	2.06×10^{4}
			8	2.73×10^{4}
			9	3.04×10^{4}
			10	6.75×10^{4}
		4	1	1.71×10^{4}
			2	1.34×10^{4}
			3	2.22×10^{4}
			4	1.74×10^{4}
			5	1.79×10^{4}
			6	6.65×10^{4}
			7	5.10×10^{4}
			8	3.25×10^{5}
			9	2.81×10^{4}
			10	3.52×10^{5}

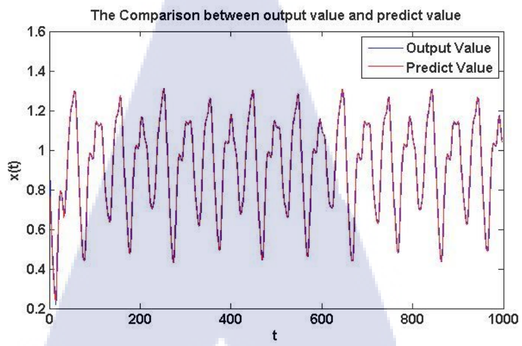


Figure 4.14 The Comparison between output values and predicted values from Mackey Glass Equation using Log-Sigmoid Transfer Function

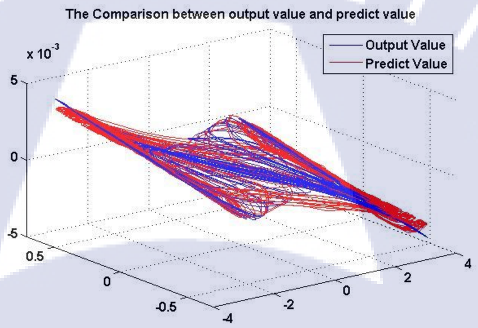


Figure 4.15 The Comparison between output values and predicted values from Chua’s Circuit using Log-Sigmoid Transfer Function

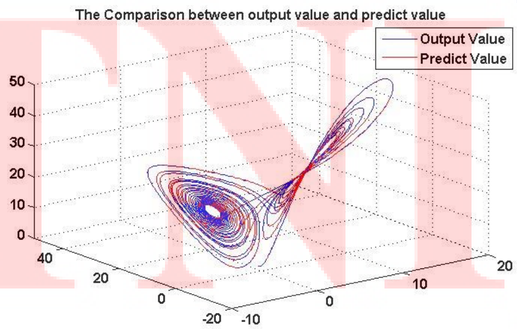


Figure 4.16 The Comparison between output values and predicted values from Lorenz Equation using Log-Sigmoid Transfer Function

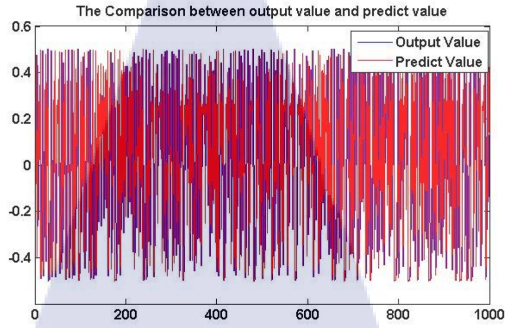


Figure 4.17 The Comparison between output values and predicted values from Chaotic Map using Log-Sigmoid Transfer Function

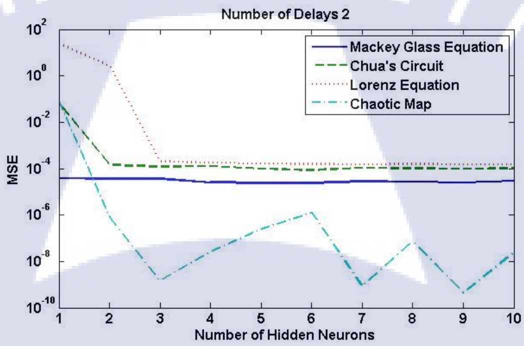


Figure 4.18 Result Experiment by Number of Delays 2 using Log-Sigmoid Transfer Function

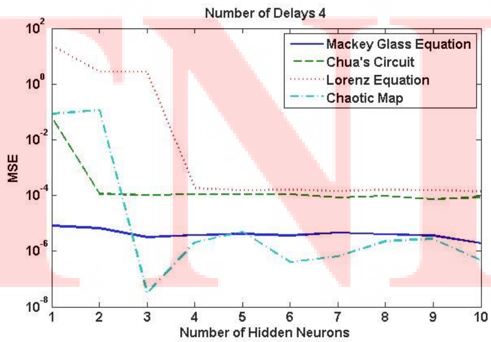


Figure 4.19 Result Experiment by Number of Delays 4 using Log-Sigmoid Transfer Function

4.4.3 Radial Basis Transfer Function

The summary result shown in Table 4.11, 4.12, 4.13, 4.14, 4.15 and Fig.4.20, 4.21, 4.22, 4.23, 4.24, 4.25

Table 4.11 Prediction Result of simulated data from Mackey Glass Equation using Radial Basis Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Mackey Glass Equation	Radial Basis Transfer Function	2	1	3.73×10^{-5}
			2	3.68×10^{-5}
			3	2.62×10^{-5}
			4	2.54×10^{-5}
			5	2.41×10^{-5}
			6	4.00×10^{-5}
			7	2.46×10^{-5}
			8	2.48×10^{-5}
			9	2.44×10^{-5}
			10	2.28×10^{-5}
		4	1	7.90×10^{-6}
			2	7.68×10^{-6}
			3	3.90×10^{-6}
			4	3.61×10^{-6}
			5	3.80×10^{-6}
			6	4.92×10^{-6}
			7	9.99×10^{-7}
			8	4.96×10^{-6}
			9	9.95×10^{-6}
			10	1.05×10^{-6}

Table 4.12 Prediction Result of simulated data from Chua's Circuit using Radial Basis Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Chua's Circuit	Radial Basis Transfer Function	2	1	0.07
			2	1.43×10^{-4}
			3	1.49×10^{-4}
			4	1.42×10^{-4}
			5	9.13×10^{-5}
			6	1.23×10^{-4}
			7	1.07×10^{-4}
			8	1.06×10^{-4}
			9	1.07×10^{-4}
			10	8.26×10^{-5}
		4	1	0.35
			2	1.13×10^{-4}
			3	1.06×10^{-4}
			4	9.50×10^{-5}
			5	8.64×10^{-5}
			6	1.07×10^{-4}
			7	8.96×10^{-5}
			8	1.02×10^{-4}
			9	9.30×10^{-5}
			10	8.58×10^{-5}

Table 4.13 Prediction Result of simulated data from Lorenz Equation using Radial Basis Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Lorenz Equation	Radial Basis Transfer Function	2	1	24.64
			2	7.39
			3	2.08×10^{-4}
			4	2.24×10^{-4}
			5	1.99×10^{-4}
			6	1.76×10^{-4}
			7	1.59×10^{-4}
			8	1.64×10^{-4}
			9	1.69×10^{-4}
			10	1.59×10^{-4}
		4	1	24.58
			2	2.75
			3	1.92×10^{-4}
			4	1.79×10^{-4}
			5	1.64×10^{-4}
			6	1.49×10^{-4}
			7	1.59×10^{-4}
			8	1.55×10^{-4}
			9	1.45×10^{-4}
			10	1.36×10^{-4}

Table 4.14 Prediction Result of simulated data from Chaotic Map using Radial Basis Transfer Function

Simulated Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Chaotic Map	Radial Basis Transfer Function	2	1	1.65×10^{-8}
			2	6.88×10^{-7}
			3	3.86×10^{-4}
			4	8.62×10^{-8}
			5	4.54×10^{-8}
			6	6.05×10^{-8}
			7	2.91×10^{-7}
			8	2.64×10^{-8}
			9	3.18×10^{-7}
			10	1.81×10^{-7}
		4	1	1.64×10^{-8}
			2	0.07
			3	0.04
			4	0.01
			5	1.74×10^{-9}
			6	1.75×10^{-7}
			7	0.05
			8	1.48×10^{-5}
			9	6.95×10^{-5}
			10	1.54×10^{-7}

Table 4.15 Prediction Result of real data from Collection using Radial Basis Transfer Function

Data Set	Transfer Function	Experiment Results		
		Number of Delays	Number of Hidden Neurons	MSE
Department of Highways	Radial Basis Transfer Function	2	1	3.26×10^{5}
			2	2.23×10^{4}
			3	1.73×10^{4}
			4	4.56×10^{4}
			5	6.01×10^{4}
			6	2.69×10^{4}
			7	6.72×10^{4}
			8	1.83×10^{4}
			9	4.51×10^{4}
			10	3.95×10^{4}
		4	1	3.27×10^{5}
			2	1.79×10^{4}
			3	1.85×10^{4}
			4	3.57×10^{4}
			5	4.05×10^{4}
			6	1.33×10^{4}
			7	2.43×10^{5}
			8	4.00×10^{4}
			9	3.23×10^{4}
			10	4.04×10^{4}

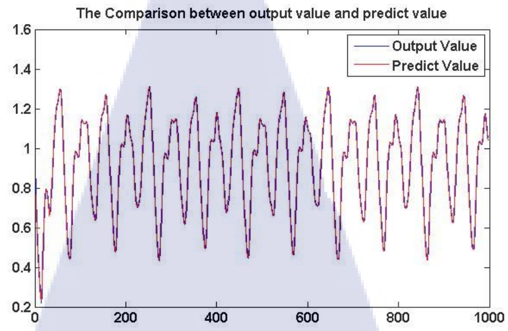


Figure 4.20 The Comparison between output values and predicted values from Mackey Glass Equation using Radial Basis Transfer Function

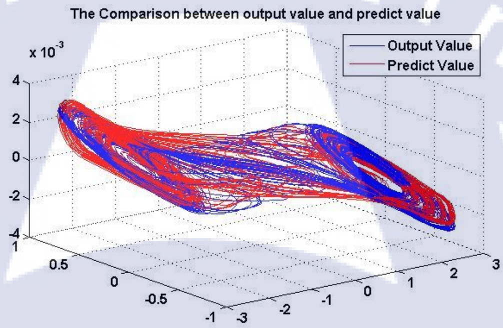


Figure 4.21 The Comparison between output values and predicted values from Chua's Circuit using Radial Basis Transfer Function

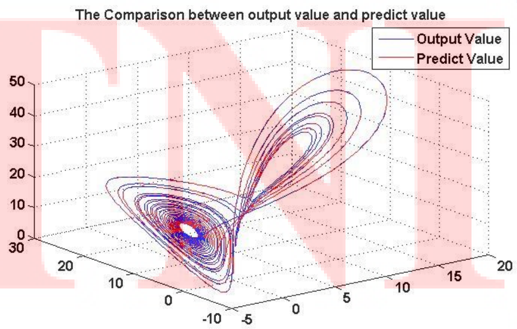


Figure 4.22 The Comparison between output values and predicted values from Lorenz Equation using Radial Basis Transfer Function

TNI

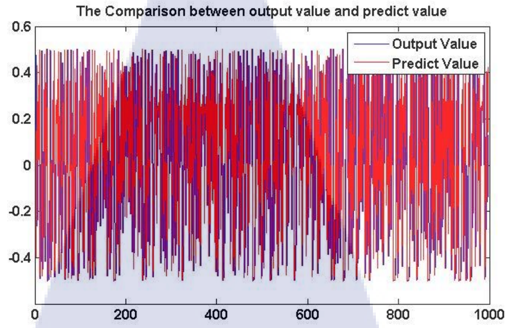


Figure 4.23 The Comparison between output values and predicted values from Chaotic Map using Radial Basis Transfer Function

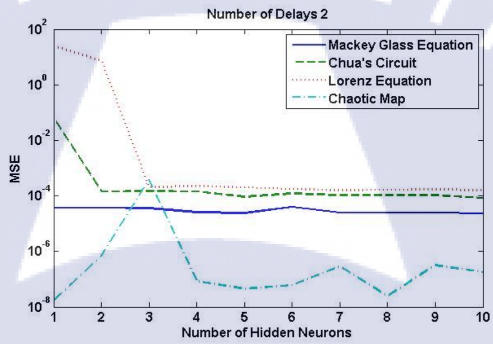


Figure 4.24 Result Experiment by Number of Delays 2 using Radial Basis Transfer Function

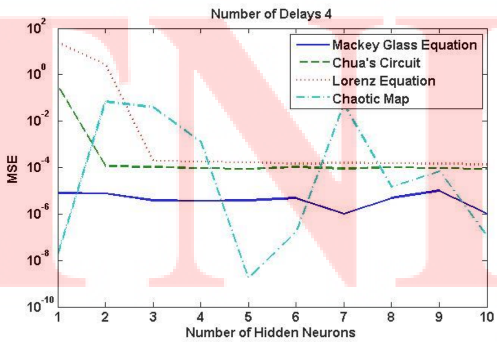


Figure 4.25 Result Experiment by Number of Delays 4 using Radial Basis Transfer Function

4.5 Conclusion

The simulate data sets are nonlinear data using to training neural network and compare to find optimal value for configuration NARX. The NARX technique is used to analyze and predict chaotic time series. It was found delays, hidden neurons, data volume and transfer function influenced the predictive accuracy of data. In the future, NARX Model can develop to improve accuracy for applies to communication system based on mobile application or web application.

Chapter 5
Conclusion

5.1 Introduction

This chapter summarizes the thesis research and suggestion for further researches and implementation. The first part of this chapter summarizes the objectives and proposed approaches in this thesis. The second part of this chapter discusses the results from the proposed implementations.

5.2 Summary

The objectives of this thesis as described in the first chapter were, to study Chaotic Time Series. This thesis has satisfied all objectives described in the first chapter. The Chaotic Time Series were generated on MATLAB, prediction using NARX with configuration and comparison performance in this thesis.

5.3 Suggestion

In the future, NARX Model Develop to improve accuracy for applying to communication system based on mobile application or web application.

5.4 Conclusion

The NARX technique is used to analyze and predict chaotic time series. It was found delays, hidden neurons, data volume and transfer function influenced the predictive accuracy of data.

References

สถาบันเทคโนโลยีไทย-ญี่ปุ่น
THAI - NICHI INSTITUTE OF TECHNOLOGY
TNI

References

[1] J. Zhao et al., "Levenberg-Marquardt algorithm for mackey-glass chaotic time series prediction," *Journal of Discrete Dynamics in Nature and Society*, vol. 2014, pp.1-6, November 2014.

[2] A. Lonescu and F. Munteanu, "A geometric and computational analysis of Chua's circuit system," *2016 International Conference on Applied and Theoretical Electricity*, ICATE, Faculty of Electrical Engineering University of Craiova, Craiova, Romania, October 6-8, 2016, pp.1-5.

[3] K. Cho and T. Miyano, "Chaotic cryptography using augmented lorenz equations aided by quantum key distribution," *Journal of IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp.478-487, December 2014.

[4] D. Lambić, "A new discrete chaotic map based on the composition of permutations," *Journal of Chaos, Solitons & Fractals*, vol. 78, pp.245-248, September 2015.

[5] S. Agatonovic-Kustin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *Journal of Pharmaceutical and Biomedical Analysis*, vol. 22, pp.717-727, June 2000.

[6] F. D. C. Lopes et al., "A control-oriented model of a PEM fuel cell stack based on NARX and NOE neural networks," *Journal of IEEE Transactions on Industrial Electronics*, vol. 62, pp.5155-5163, August 2015.

[7] K. Hammar et al., "Hierarchical levenberg-marquardt algorithm for identification of fractional nonlinear system," *2016 8th International Conference on Modelling, Identification and Control*, ICMIC, University of Medea, Algiers, Algeria, November 15-17, 2016, pp. 660-664.

[8] S. Zhang et al., "A new fault-location algorithm for extra-high-voltage mixed lines based on phase characteristics of the hyperbolic tangent function," *Journal of IEEE Transactions on Power Delivery*, vol. 31, pp.1203-1212, June 2016.

[9] L. Zhou and L. Zhang, “A Log-Sigmoid lagrangian neural network for solving nonlinear programming,” *2007 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, SNPD, Qingdao Technological University, Qingdao, China, July 30 - August 1, 2007, pp. 427-431.

[10] M. A. Selver and C. Guzelis, “Semiautomatic transfer function initialization for abdominal visualization using self-generating hierarchical radial basis function networks,” *Journal of IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp.395-409, May 2009.

[11] A. Thakur et al., “NARX based forecasting of petrol prices,” *5th International Conference on Reliability, Infocom Technologies and Optimization*, ICRITO, AIIT, Amity University Uttar Pradesh, Noida, India, September 7-9, 2016, pp. 610-614.

[12] L. Banjanović-Mehmedović et al., “Prediction of cooperative platooning maneuvers using NARX neural network,” *2016 International Conference on Smart Systems and Technologies*, SST, Faculty of Electrical Engineering Josip Juraj Strossmayer University of Osijek, Osijek, Croatia, October 12-14, 2016, pp. 287-292.

[13] S. Jaiswal et al., “Modeling the measurement error of energy meter using NARX model,” *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, Taipei International Convention Center, Taipei, Taiwan, May 23-26, 2016, pp. 1-6.

[14] L. Zhang et al., “NARX models for predicting power consumption of a horizontal axis wind turbine,” *2016 UKACC 11th International Conference on Control*, CONTROL, Belfast, Northern Ireland, UK, August 31 - September 2, 2016, pp.1-5.

[15] F. Chang et al., “Estimating spatio-temporal dynamics of stream total phosphate concentration by soft computing techniques,” *Journal of 26th European Symposium on Computer Aided Process Engineering*, vol. 38, 1st Edition, pp. 228-236, June 2016.

[16] L.K. Torres-Faurrieta et al., “Recruitment forecasting of yellowfin tuna in the eastern Pacific Ocean with artificial neuronal networks,” *Journal of Ecological Informatics*, vol. 36, pp. 106-113, November 2016.

[17] A.G.R. Vaz et al., “An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands,” *Journal of Renewable Energy*, vol. 85, pp.631-641, January 2016.

[18] Y. Chunshan and L. Xiaofeng, “Study and application of data mining and NARX neural networks in load forecasting,” *2015 4th International Conference on Computer Science and Network Technology*, ICCSNT, Halin Tianyue Hotel, Harbin, China, December 19-20, 2015, pp. 360-364.

[19] H. He et al., “Meridian ECG information transmission system modeling using NARX neural network,” *2016 IEEE/ACIS 15th International Conference on Computer and Information Science*, ICIS, Okayama Convention Center, Okayama, Japan, June 26-29, 2016, pp. 1-6.

[20] X. Zhe et al., “Water distribution network modeling based on NARX,” *Journal of IFAC-PapersOnLine*, vol. 48, no. 11, pp.72-77, January 2015.

TNI
THAI - NICHI INSTITUTE OF TECHNOLOGY

Appendices

Publication in: The TNI Academic Conference 2017

การวิเคราะห์และทำนายสภาพคล่องการจราจรด้วยเทคนิคโครงข่ายประสาทเทียมแบบถดถอยอัตโนมัติไม่เป็นเชิงเส้นร่วมกับข้อมูลอินพุตภายนอก

Traffic Flow Analytics and Prediction using Nonlinear Autoregressive Network with Exogenous Inputs Technique

พีรพล พุทธเวชมงคล[1] และ วิมล แสนอุ้ม

หลักสูตรปริญญาโทสาขาเทคโนโลยีวิศวกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีไทย-ญี่ปุ่น 1771/1 ถนนพัฒนาการ แขวงสวนหลวง เขตสวนหลวง กรุงเทพมหานคร 10250

ประเทศไทย หมายเลขโทรศัพท์ 02-763-2600 ต่อ 2905, 2910 หมายเลขโทรสาร 02-7632600 ต่อ 2900

[1]ผู้วิจัยหลัก E-mail: ph.peerapol_st@tni.ac.th

บทคัดย่อ

งานวิจัยนี้นำเสนอการวิเคราะห์และการทำนายสภาพคล่องการจราจรด้วยเทคนิคโครงข่ายประสาทเทียมแบบถดถอยอัตโนมัติไม่เป็นเชิงเส้นร่วมกับข้อมูลอินพุตภายนอก (NARX) โดยมีการแบ่งการกำหนดค่าในการเรียนรู้ของโครงข่ายประสาทเทียมเป็น 2 ส่วน ส่วนแรกการกำหนดค่าดีเลย์เท่ากับ 2 และส่วนที่สองค่าดีเลย์เท่ากับ 4 เพื่อเปรียบเทียบผลลัพธ์และหาค่าที่เหมาะสมที่สุดในการนำไปสร้างแบบจำลอง NARX โดยอาศัยการใช้ข้อมูลจำลองที่สร้างมาจากสมการแม็กกี้กลาสมีความผันผวนเหมือนสภาพคล่องการจราจร จำนวน 1000 ชุดข้อมูล และนำข้อมูลมาวิเคราะห์หาผลลัพธ์ด้วยแบบจำลอง NARX โดยสามารถนำไปพัฒนาต่อในการทำแอพพลิเคชั่นหรือเว็บแอพพลิเคชั่นที่ใช้ในกรุงเทพมหานครได้ต่อไปในอนาคต

คำสำคัญ: การวิเคราะห์, การทำนาย, สภาพคล่องการจราจร, สมการแม็กกี้กลาส, NARX

Abstract

This paper proposes the traffic flow analytics and prediction using nonlinear autoregressive network with exogenous inputs technique (NARX). The learning of neural network is divided into 2 configuration parts. The configuration delays are 2 and 4 to compare the results and find optimal configuration value for create NARX model using the simulate data. The chaotic simulate data set are generate by Mackey glass equation. Then analyze data set using NARX model. This able to developed to application or web application used in Bangkok.

Keywords: Analytics, Prediction, Traffic Flow, Mackey Glass Equation, NARX

1) ที่มาและความสำคัญ

เมื่อปี 2559 ที่ผ่านมาจากผลสำรวจสภาพการจราจรกรุงเทพมหานครเป็นอันดับที่ 1 ที่มีการจราจรติดขัดที่สุดในโลก[1] แต่วิถีชีวิตในกรุงเทพมหาครกลับมีความเร่งรีบในการเดินทางเป็นอย่างมาก อีกทั้งปัญหาสภาพการจราจรภายในกรุงเทพมหานครที่ยังไม่ได้รับการแก้ไขมาเป็นเวลานาน ทำให้ผู้คนเริ่มหันมาหลีกเลี่ยงเส้นทางที่มีการจราจรหนาแน่นแต่ทว่าก็ไม่สามารถคาดการได้อย่างถูกต้องเสมอไปแม้จะใช้แอพพลิเคชั่นที่มีในปัจจุบันเพื่อดูข้อมูลการจราจร ณ เวลานั้นๆ เนื่องแอพพลิเคชั่นยังไม่สามารถวิเคราะห์ผลลัพธ์ได้อย่างแม่นยำที่ตอบสนองความพึงพอใจของผู้ใช้ประกอบกับตัวแอพพลิเคชั่นยังไม่สามารถทำนายผลการจราจรล่วงหน้าได้ทำให้ผู้วิจัยมีแนวคิดที่จะริเริ่มในการทำระบบวิเคราะห์สภาพคล่องการจราจรที่มีความแม่นยำขึ้นและการทำนายผลสภาพคล่องการจราจรโดยการนำ NARX มาช่วยเพิ่มประสิทธิภาพของกระบวนการวิเคราะห์สภาพคล่องการจราจร เพื่อให้คนที่ใช้ชีวิตในเมืองหลวงสามารถรู้ล่วงหน้าถึงสภาพการจราจรและวางการเดินทาง หรือหลีกเลี่ยงเส้นทางที่แออัดได้อย่างถูกต้อง

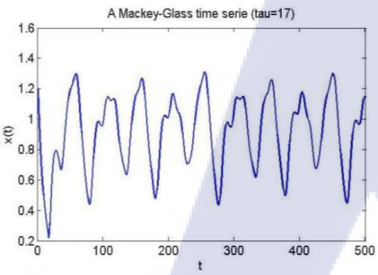
งานวิจัยนี้ได้ทำการเปรียบเทียบผลลัพธ์จากการทำนายสภาพคล่องการจราจรด้วยการกำหนดค่าในการเรียนรู้ของโครงข่ายประสาทเทียมที่ต่างกันออกไป เพื่อเปรียบเทียบและหาค่าที่เหมาะสมที่สุดในการสร้างแบบจำลอง NARX โดยจำลองข้อมูลสภาพการจราจรขึ้นมาจากสมการแม็กกี้กลาสที่มีความใกล้เคียงกัน เพื่อนำไปใช้ในการทำแอพพลิเคชั่นหรือเว็บแอพพลิเคชั่นต่อไปในอนาคต

2) ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1) สมการแม็กกี้กลาส

สมการแม็กกี้กลาสเป็นสมการดิฟเฟอเรนเชียลที่ไม่เป็นเชิงเส้นโดยมีค่าหน่วงเวลา สมการแม็กกี้กลาสจะแสดงช่วงของเวลาและสัญญาณอลวนซึ่งเหมาะแก่การนำมาใช้ในการจำลองข้อมูลที่มีความผันผวน ไม่เป็นเชิงเส้น ดังรูปที่ 1 โดยในงานวิจัยครั้งนี้จะใช้กับการจำลองข้อมูลสภาพการจราจร สมการแม็กกี้กลาสสามารถเขียนได้ดังสมการที่ 1

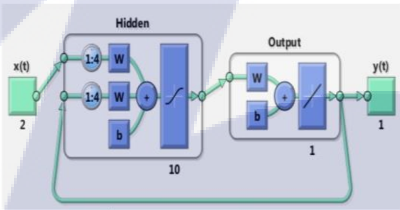
$$\frac{dx(t)}{dt} = \frac{ax(t-r)}{1+(x-r)^{10}} - bx(t) \qquad (1)$$



รูปที่ 1 ตัวอย่างข้อมูลที่สร้างจากสมการแม็กกี้กลาส

2.2) แบบจำลอง NARX

โครงข่ายประสาทเทียมเป็นโมเดลทางคณิตศาสตร์ที่ใช้ในการประมวลผลสารสนเทศด้วยการคำนวณด้วยเทคนิคคอนเนคชันนิส (Connectionist) ที่เลียนแบบมาจากสมองของมนุษย์ ส่วน NARX เป็นโครงข่ายประสาทเทียมชนิดหนึ่งที่ใช้ข้อมูลเชิงเวลาโดยอาศัยข้อมูลอินพุตภายนอกในการเรียนรู้ของโครงข่ายประสาทเทียมซึ่งสามารถป้อนกลับข้อมูลที่ได้เพิ่มความแม่นยำในการเรียนรู้ของโครงข่ายประสาทเทียม และสามารถใช้ทำนายข้อมูลในอนาคตได้อย่างแม่นยำ



รูปที่ 2 โครงสร้าง NARX แบบ Closed Loop

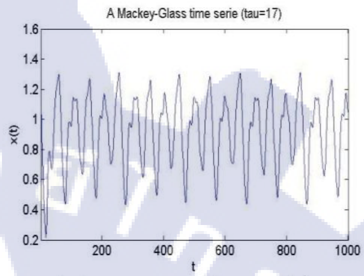
2.3) งานวิจัยที่เกี่ยวข้อง

NARX เป็นเทคนิคที่มีประโยชน์ในการคาดการณ์ข้อมูล จึงมีงานวิจัยที่นำ NARX ไปประยุกต์ใช้อย่างแพร่หลาย ทั้งด้านการคมนาคม ด้านพลังงาน ด้านสิ่งแวดล้อม รวมถึงด้านเศรษฐศาสตร์และด้านอื่นๆ งานวิจัยทางด้านคมนาคมของคุณ Lejla Banjanović-Mehmedović และคณะ [2] ได้นำ NARX ไปพัฒนาระบบการขนส่งอัจฉริยะที่ช่วยเพิ่มความคล่องตัวของการจราจร ผลจากการนำ NARX ไปใช้กับระบบทำให้ระบบมีความแม่นยำที่สูงขึ้น ด้านเศรษฐศาสตร์ คุณ Anita Thakur [3] และคณะ ได้นำ NARX ไปใช้กับการคาดการราคาน้ำมันเพื่อกำหนดการผลิตให้ตรงกับความต้องการของผู้บริโภคได้อย่างเหมาะสม ด้านพลังงานมีการนำไปประยุกต์ใช้ในหลายงานวิจัย เช่น คุณ Lidong Zhang และคณะ [4] ได้นำ NARX ในการทำนายการใช้พลังงานของกังหันลมแนวแกนนอนได้อย่างแม่นยำ หรือ คุณ A.G.R. Vaz และคณะ [5] ได้ใช้ NARX ในการประเมินผลของระบบพลังงานแสงอาทิตย์ของเมืองยูเทรกซ์และบริเวณใกล้เคียง เพื่อผลิตให้เพียงพอแก่การใช้งานและสามารถทำนายผลการใช้งานล่วงหน้าได้มากกว่า 15 นาที อีกทั้งงานวิจัยของคุณ Yang Chunshan และ Li Xiaofeng [6] ที่ศึกษาเกี่ยวกับความสัมพันธ์ของเหมืองข้อมูลกับการใช้ NARX ในการทำนายการใช้ไฟฟ้าของจีน ทำให้สามารถแบ่งแยกประเภทข้อมูลที่เป็นปัจจัยสำคัญต่อการทำนายอย่างชัดเจนและทำนายผลลัพธ์ได้อย่างแม่นยำยิ่งขึ้น แต่อย่างไรก็ตามการจะนำ NARX ไปใช้ต้องมีข้อมูลอินพุตที่ถูกต้องและการกำหนดค่าที่เหมาะสม

3) วิธีการดำเนินการวิจัย

3.1) สร้างข้อมูลจำลอง

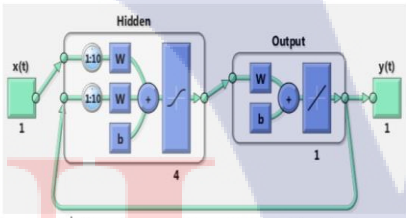
ใช้สมการแม็กกี้กลาสในการสร้างข้อมูลจำลอง เพื่อใช้ในการฝึกสอนโครงข่ายประสาทเทียม โดยใช้ตัวสมการ (1) ในสร้างข้อมูลจำลองด้วยวิธี 4th-Order Runge-Kutta ด้วยการกำหนดค่า Step Size เท่ากับ 1 และ τ เท่ากับ 17 ข้อมูลที่จำลองทั้งหมดจำนวน 1000 ชุด ค่าเริ่มต้นเท่ากับ 1.2 ผลลัพธ์ที่ได้การจำลองข้อมูลจากสมการแม็กกี้กลาส ดังรูปที่ 3



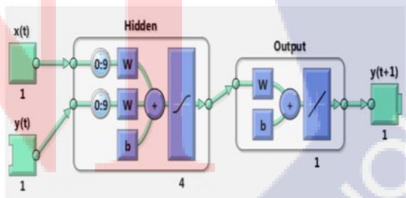
รูปที่ 3 ข้อมูลจำลอง 1000 ชุด ที่ได้จากสมการแม็กกี้กลาส

3.2) ฝึกสอนแบบจำลอง NARX

การฝึกสอนแบบจำลอง NARX จะเกี่ยวกับในการกำหนดค่าน้ำหนักที่เหมาะสม (Weight) รวมถึงโครงข่ายประสาทเทียมที่มีค่าชั้นแฝง (Number of Hidden Neurons) อินพุต (Input) และดีเลย์ (Number of Delays) เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด ในงานวิจัยนี้นำข้อมูลจำลองที่ได้ที่จากสมการแม็กกี้กลาสมาป้อนให้กับแบบจำลอง NARX ผ่านโปรแกรม MATLAB ด้วยชุดคำสั่งของ Neural Network Time Series Tool โดยแบ่งข้อมูลจำลองออกเป็น 3 ชุด ชุดแรกใช้ในการฝึกสอนโครงข่ายประสาทเทียม (Training) จำนวน 70 เปอร์เซ็นต์ ชุดที่สองใช้ในการตรวจสอบ (Validation) จำนวน 15 เปอร์เซ็นต์ ชุดที่สามใช้ในการทดสอบผล (Testing) จำนวน 15 เปอร์เซ็นต์ โครงสร้างแบบจำลอง NARX ที่ได้จะเป็นดังรูปที่ 4 และ 5



รูปที่ 4 โครงสร้างของแบบจำลอง NARX แบบ Close Loop



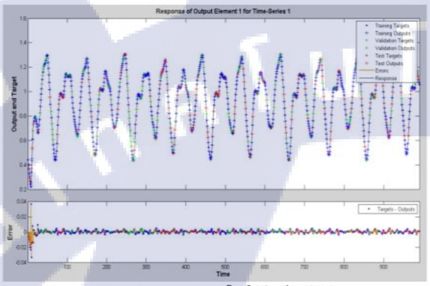
รูปที่ 5 โครงสร้างของแบบจำลอง NARX แบบ Predict 1 Step Ahead

ค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error : MSE) จะถูกใช้ในการวัดค่าความแม่นยำของ NARX ดังสมการที่ (2)

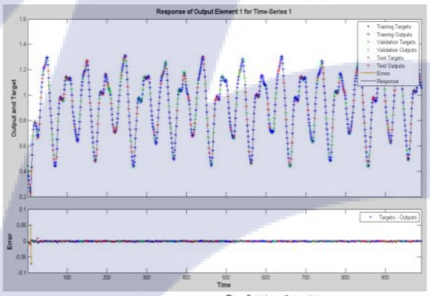
$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{y(i)-y_i}{y(i)}\right)^2 \quad (2)$$

4) ผลการวิจัย

ผลลัพธ์แบ่งออกเป็น 2 กลุ่ม ตามการกำหนดค่าดีเลย์ กลุ่มแรกใช้ค่าดีเลย์เท่ากับ 2 และกลุ่มที่สองใช้ค่าดีเลย์เท่ากับ 4 โดยรูปที่ 6 7 และ 8 จะแสดงความถูกต้องของการทำนายด้วยค่าดีเลย์ที่ต่างกัน แต่ค่าชั้นแฝงเท่ากับ 10

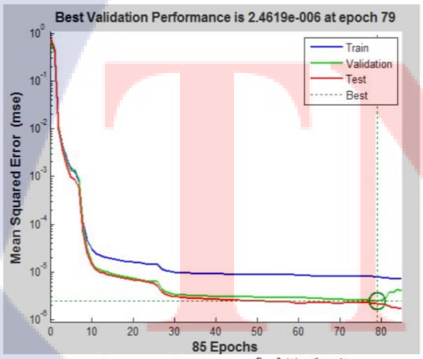


ก) กราฟแสดงการทำนายโดยใช้ดีเลย์เท่ากับ 2

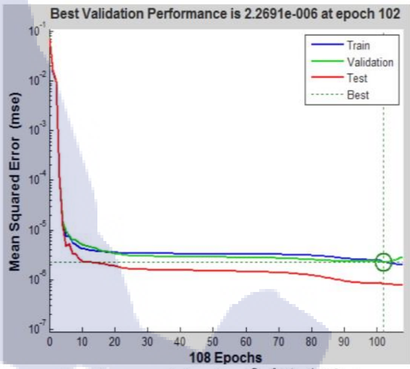


ข) กราฟแสดงการทำนายโดยใช้ดีเลย์เท่ากับ 4

รูปที่ 6 ผลลัพธ์เชิงเวลาข้อมูลของแบบจำลอง NARX

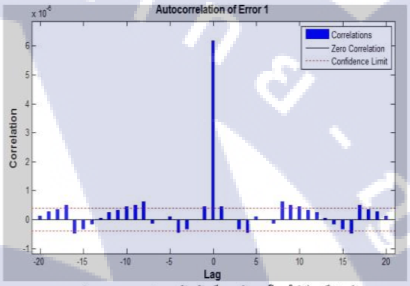


ก) กราฟแสดงความแม่นยำการทำนายโดยใช้ดีเลย์เท่ากับ 2

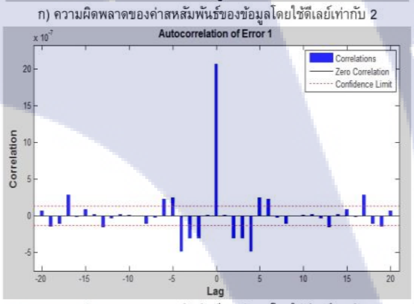


ข) กราฟแสดงความแม่นยำการทำนายโดยใช้ดีเลย์เท่ากับ 4

รูปที่ 7 ผลลัพธ์ความแม่นยำของแบบจำลอง NARX



ก) ความผิดพลาดของค่าสหสัมพันธ์ของข้อมูลโดยใช้ดีเลย์เท่ากับ 2



ก) ความผิดพลาดของค่าสหสัมพันธ์ของข้อมูลโดยใช้ดีเลย์เท่ากับ 4

รูปที่ 8 ความผิดพลาดของค่าสหสัมพันธ์ข้อมูลของแบบจำลอง NARX

ตารางที่ 1 สรุปผล MSE ที่ได้จากการประมวลผลผ่านแบบจำลอง NARX ด้วยค่าดีเลย์และค่าชั้นแฝงของโครงข่ายประสาทเทียมที่แตกต่างกัน

THAI-NICHI INSTITUTE OF TECHNOLOGY

ตารางที่ 1 สรุปผลลัพธ์ที่ได้จากการประมวลผลด้วย NARX

Number of Delays	Number of Hidden Neurons	MSE
2	1	0.00123
	2	3.81e-05
	3	8.64e-06
	4	6.60e-06
	5	7.29e-06
	6	7.80e-06
	7	4.70e-06
	8	8.57e-06
	9	1.88e-06
	10	6.11e-06
4	1	0.0012
	2	2.50e-05
	3	6.69e-06
	4	5.31e-06
	5	6.28e-07
	6	5.96e-07
	7	2.69e-06
	8	5.15e-06
	9	4.25e-06
	10	6.51e-06

5) สรุปผล

การนำโครงข่ายประสาทเทียมแบบ NARX มาใช้ในการวิเคราะห์และทำนายผลการจราจรจากข้อมูลจำลอง พบว่าค่าดีเลย์และค่าชั้นแฝงของโครงข่ายประสาทเทียมมีผลต่อความแม่นยำในทำนายข้อมูลล่วงหน้า

ในอนาคตพัฒนาแบบจำลอง NARX ให้มีความแม่นยำมากขึ้นและนำไปพัฒนาต่อเป็นระบบ ทำเป็นแอพพลิเคชั่นบนมือถือ เพื่อหลีกเลี่ยงปัญหาการจราจรที่หนาแน่นได้ต่อไปในอนาคต

เว็บไซต์

[1] INRIX. (23 มีนาคม 2560). INRIX 2016 Global Traffic Scorecard. [Online] แหล่งที่มา : http://inrix.com/resources/inrix-2016-global-traffic-scorecard/

วิทยานิพนธ์/สารนิพนธ์

[2] Lejla Banjanović-Mehmedović, Ivana Butigan, Mehmed Kantardžić and Suad Kasapović, "Prediction of Cooperative Platooning Maneuvers Using NARX Neural Network", 2016 International Conference on Smart Systems and Technologies (SST), pp. 287 – 292, 2016

[3] Anita Thakur, Aishwarya Tiwari, Saswat Kumar, Aditya Jain and Jagjot Singh, "NARX Based Forecasting of Petrol Prices", 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), pp. 610 – 614, 2016

[4] Lidong Zhang, Lijiao Ma, Kang Li, Shaojun Gan, Juan Yan, Xiandong Xu, Mingliang Zhu, Weiwei Li, "NARX models for predicting power consumption of a horizontal axis wind turbine" , 2016 UKACC 11th International Conference on Control (CONTROL), pp. 1 – 5, 2016

[5] A.G.R. Vaz, B. Elsinga, W.G.J.H.M. van Sark and M.C. Brito, "An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands", Journal, Volume 85, pp.631 - 641, 2016

[6] Yang Chunshan and Li Xiaofeng, "Study and Application of Data Mining and NARX Neural Networks in Load Forecasting", 2015 4th International Conference on Computer Science and Network Technology (ICCSNT 2015), pp. 360 – 364, 2015

Publication in: The 2017 International Conference on Embedded System and Intelligent Technology

The Prediction of Chaotic Time Series using Nonlinear Autoregressive Network with Exogenous Input Technique

Peerapol Phutthawetmongkhong
Centre of Excellence in Intelligent Integration System,
Thai-Nichi Institute of Technology
1771/1, Pattanakarn Rd., Suan Luang, Bangkok, 10250, THAILAND

Wimol San-Um
Centre of Excellence in Intelligent Integration System,
Thai-Nichi Institute of Technology
1771/1, Pattanakarn Rd., Suan Luang, Bangkok, 10250, THAILAND

Abstract—**This paper propose the prediction of time series using nonlinear autoregressive network with exogenous input technique (NARX). The learning of neural network is divided into 2 configuration parts. The configuration delays are 2 and 4 to compare the results and find the optimal configuration value for create NARX model using the simulate data. The chaotic simulate data set are generate by Chaotic Equation. Then analyze data set using NARX model. This able to developed to application or web application used in Bangkok.**

Keywords; Analytics; Prediction; Choatic Time Series; NARX;

I. Introduction

The predictive information is based on time series data. The chaotic time series is nonlinear data such as weather, load energy or communication. Recent researches the prediction were used to resolve the problem for communication, petrol price, energy [1, 2, 3]. There are various methods to predict information, including NARX for predicting time series data.

In year 2016, the worst traffic in the world is in Bangkok, Thailand [4]. People in Bangkok have extremely hasty lifestyle, especially in transportation. However, Bangkok traffic problem have not been resolved for a long time. People in Bangkok attempt to avoid traffic jam, whereas the traffic cannot be always accurately expected even with the use of current applications to view traffic data at that time. The application cannot accurately analyze traffic flow for user satisfaction include cannot predict traffic flow. The research initiates idea for increasing accurately traffic flow analytics system and predict traffic flow using NARX to improve the efficiency of analytics in order that people can predict traffic flow and transport plans or avoid the crowded route properly. The researcher starts from using chaotic time series instead of traffic flow to create NARX.

This paper compares the result of prediction with different time series data and different learning configuration of neural network to find the optimal value to create NARX model. The time series data simulate from chaotic equation. This result can develop to application or web application in future.

II. Related Theories and Literature Reviews

A. *Mackey Glass Equation*

Mackey Glass Equation is nonlinear time delay differential equation. This Mackey Glass equation displays a range of periodic and chaotic dynamics which compatible with simulate fluctuate data and nonlinear data. This equation use to simulate traffic flow data. Mackey Glass equation can be described mathematically as

$$\frac{dx(t)}{dt} = \frac{ax(t-r)}{1+x(t-r)^{10}} - bx(t) \qquad (1)$$

where a is real numbers, and $(t-r)$ is value at x time. Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics.

B. *Chua's Circuit*

Chua's Circuit is the simplest electronic circuit exhibiting chaos and stability. It produces oscillating waveform. The ease of construction of the circuit has made it a ubiquitous real-world example of a chaotic system.

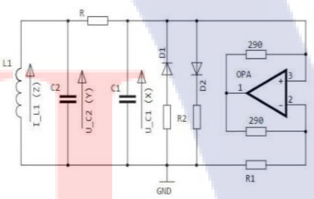


Figure 1. One version of Chua's Circuit

C. *Lorenz Equation*

Lorenz Equation is a system of three ordinary differential equations. The Lorenz Equation are given by

$$\dot{x} = \sigma(y-x), \dot{y} = rx - y - xz, \dot{z} = xy - bz \qquad (2)$$

where x, y and z make up the system state at time. σ, ρ and β are the system parameters.

D. Quadratic Map(Choatic Map)

The system of the quadratic map is chaotic because it has the following characteristics. It is nonlinear. The quadratic map can be synchronized through coupling.

E. NARX Model

Artificial Neural Network is mathematical model apply to evaluate information with connectionist technique. This technique emulate from human brain. NARX is a nonlinear autoregressive model with exogenous inputs for artificial neural network learning with loopback. This technique increases the accuracy for learning and prediction.

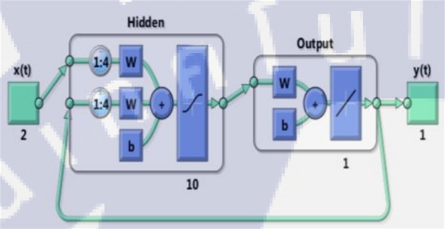


Figure 2. Structure of NARX Closed Loop

F. Literature Reviews

NARX is useful technique for predict information. NARX applied to communication, energy, environment, economic and etc. Lejla Banjanović-Mehmedoć's research [1] present an approach towards NARX improve Intelligent Transport systems. Anita Thakur's research [2] present an approach towards NARX for petrol price forecasting. Lidong Zhang's research [3] present an approach towards NARX for predict power consumption of a horizontal axis wind turbine. A.G.R Vaz's research [5] present an approach towards NARX predict power forecasting in Utrecht, the Netherland. Yang Chunshan and Li Xiaofeng's research [6] present an approach towards Data Mining with NARX for load forecasting. The above research mentioned NARX was used in many field by different factor. Implementing NARX to system, the result depend on variable type and configure NARX input.

III. REASEARCH METHODOLOGY

A. Simulate Data Set

- Mackey Glass Equation using 4th-Order Runge-Kutta method. This method decide step size is 1, τ is 17, simulate data set is 1000 and initial condition is 1.2.

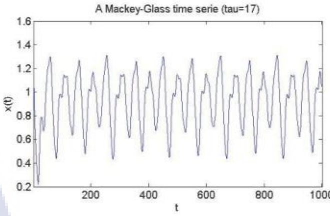


Figure 3. Simulate Data Set by Mackey Glass Equation

- Realistic Chua's Circuit is configured resistors are R1=220 Ohms, R2=220 Ohms, R3=2200 Ohms, R4=22000 Ohms, R5=22000 Ohms, R6=3300 Ohms, R7=100 Ohms, R8=1000 Ohms, R9=1000 Ohms, R10=1800 Ohms, R=1800 Ohms, C=100nF, C1=10nF and C2=100nF

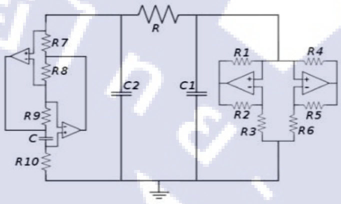


Figure 4. Schematic of simulated circuit

- Lorenz Equation using the values

$$\sigma = 10, \beta = \frac{8}{3}, \rho = 28 \quad (3)$$

- Quadratic Map(Chaotic Map) is generated with default value in MATLAB

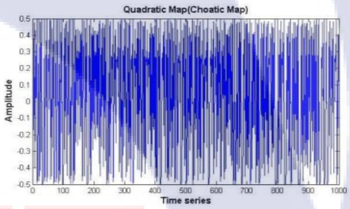


Figure 5. Simulate Data Set by Quadratic Map

B. Training NARX Model

Training NARX Model involve configure weight, number of hidden neurons, input and delays. The accuracy base on a configure value. This paper use simulate data sets input NARX by Network Time Series Tool in MATLAB. These simulate data sets split 3 set. Training set 70 percent. Validation set 15 percent. Testing set 15 percent. Fig.6 shows a structure NARX model.

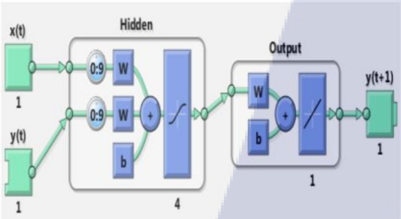


Figure 6. Structure NARX Predict 1 Step Ahead Model

Mean Square Error (MSE) evaluates the accuracy of NARX. Mean Square Error can be described mathematically as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y(i)-y_i}{y(i)}\right)^2 \qquad (4)$$

C. Pseudocode

NARX using all simulate data set to define input and target in Fig. 7

```
#   Define input
#   Define target
#   Define input delays
#   Define feedback delays
#   Define hidden layer
#   Create Neural Network
#   Prepare data for Network
#   Configure training ratio
#   Configure validation ratio
#   Configure testing ratio
#   Training Neural Network
#   Testing Neural Network
#   Create Close Loop by Neural Network
#   Prepare data for NARX
#   Training NARX
#   Remove Time Delays from NARX
#   Prepare data for Predict 1 Step Ahead
#   Predict 1 Step Ahead
```

Figure 7. Pseudocode for NARX

IV. Experiment Results

The results are divided into 4 groups by simulate data set. The summary of MSE results from processing through the NARX model with different number of delays and number of hidden neurons.

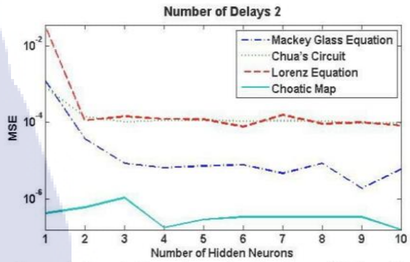


Figure 8. Result Experiment by Number of Delays 2

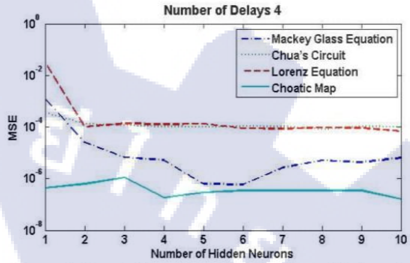


Figure 9. Result Experiment by Number of Delays 4

V. Conclusion

NARX technique is used to analyze and predict chaotic time series. It was found delays and hidden neurons influenced the predictive accuracy of data.

In the future, NARX Model Develop to improve accuracy for applies to communication system based on mobile application or web application.

References

[1] Lejla Banjanović-Mehmedović, Ivana Butigan, Mehmed Kantardžić and Suad Kasapović, "Prediction of Cooperative Platooning Maneuvers Using NARX Neural Network", 2016 International Conference on Smart Systems and Technologies(SST), pp. 287 – 292, 2016

[2] Anita Thakur, Aishwarya Tiwari, Saswat Kumar, Aditya Jain and Jagjot Singh, "NARX Based Forecasting of Petrol Prices", 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), pp. 610 – 614, 2016

[3] Lidong Zhang, Lijiao Ma, Kang Li, Shaojun Gan, Juan Yan, Xiandong Xu, Mingliang Zhu, Weiwei Li, "NARX models for predicting power consumption of a horizontal axis wind turbine" , 2016 UKACC 11th International Conference on Control (CONTROL), pp. 1 – 5, 2016

[4] INRIX. (March 23, 2017). INRIX 2016 Global Traffic Scorecard. [Online]

[5] A.G.R. Vaz, B. Elsinga, W.G.J.H.M. van Sark and M.C. Brito, "An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands", Journal, Volume 85, pp.631 - 641, 2016

[6] Yang Chunshan and Li Xiaofeng, "Study and Application of Data Mining and NARX Neural Networks in Load Forecasting", 2015 4th International Conference on Computer Science and Network Technology (ICCSNT 2015), pp. 360 – 364, 2015

1. NARX.m

```
% Solve an Autoregression Problem with External
% Input with a NARX Neural Network
% Script generated by NTSTOOL

% This script assumes the variables on the right of
% these equalities are defined:

inputSeries = mackeyGlassInput;
targetSeries = mackeyGlassOutput;

% inputSeries = tonndata(realChuaInput,false,false);
% targetSeries = tonndata(realChuaOutput,false,false);

% inputSeries = tonndata(lorenzInput,false,false);
% targetSeries = tonndata(lorenzOutput,false,false);

% inputSeries = chaoticMapInput;
% targetSeries = chaoticMapOutput;

% inputSeries = TRAFFIC_CELL;
% targetSeries = TRAFFIC_CELL;

% Create a Nonlinear Autoregressive Network with External Input
delay = 1:2;
hiddenLayerSize = 10;
inputDelays = delay;
feedbackDelays = delay;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);

% Prepare the Data for Training and Simulation
% The function PREPARETS prepares time series data
```

```
% for a particular network, shifting time by the minimum
% amount to fill input states and layer states.
% Using PREPARETS allows you to keep your original
% time series data unchanged, while easily customizing it
% for networks with differing numbers of delays, with
% open loop or closed loop feedback modes.
[inputs,inputStates,layerStates,targets] = ...
        preparets(net,inputSeries,{},targetSeries);

% net.layers{1}.transferFcn = 'logsig';
% net.layers{1}.transferFcn = 'radbas';

% Set up Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Test the Network
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs);

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
```

```
% figure, plotregression(targets,outputs)
% figure, plotresponse(targets,outputs)
% figure, ploterrcorr(errors)
% figure, plotinerrcorr(inputs,errors)
% Closed Loop Network
% Use this network to do multi-step prediction.
% The function CLOSELOOP replaces the feedback input with a direct
% connection from the output layer.
netc = closeloop(net);
netc.name = [net.name ' - Closed Loop'];
view(netc)
[xc,xic,aic,tc] = preparets(netc,inputSeries,{},targetSeries);
yc = netc(xc,xic,aic);
closedLoopPerformance = perform(netc,tc,yc);
% Early Prediction Network
% For some applications it helps to get the prediction a
% timestep early.
% The original network returns predicted y(t+1) at the same
% time it is given y(t+1).
% For some applications such as decision making, it would
% help to have predicted y(t+1) once y(t) is available, but
% before the actual y(t+1) occurs.
% The network can be made to return its output a timestep early
% by removing one delay so that its minimal tap delay is now
% 0 instead of 1.  The new network returns the same outputs as
% the original network, but outputs are shifted left one timestep.
nets = removedelay(net);
nets.name = [net.name ' - Predict One Step Ahead'];
view(nets)
[xs,xis,ais,ts] = preparets(nets,inputSeries,{},targetSeries);
ys = nets(xs,xis,ais);
earlyPredictPerformance = perform(nets,ts,ys);
```

2. mackeyglass.m

```
% This script generates a Mackey-Glass time series using the 4th order
% Runge-Kutta method.
% The code is a straighforward translation in Matlab of C source code provided by
Roger Jang,
% which is available <http://neural.cs.nthu.edu.tw/jang/dataset/mg/mg.c here>

%% The theory
% Mackey-Glass time series refers to the following, delayed differential
% equation:

% $$\frac{dx(t)}{dt}=\frac{ax(t-\tau)}{1+x(t-\tau)^{10}}-bx(t)
% \hspace{1cm} (1)$$

% It can be numerically solved using, for example, the 4th order
% Runge-Kutta method, at discrete, equally spaced time steps:

% $$x(t+\Delta t) = mackeyglass\_rk4(x(t), x(t-\tau), \Delta t, a, b)$$

 % where the function <mackeyglass_rk4.html mackeyglass_rk4> numerically solves
the
% Mackey-Glass delayed differential equation using the 4-th order Runge
% Kutta. This is the RK4 method:

% $$k_1=\Delta t \cdot mackeyglass\_eq(x(t), x(t-\tau), a, b)$$
% $$k_2=\Delta t \cdot mackeyglass\_eq(x(t+\frac{1}{2}k_1), x(t-\tau), a, b)$$
% $$k_3=\Delta t \cdot mackeyglass\_eq(x(t+\frac{1}{2}k_2), x(t-\tau), a, b)$$
% $$k_4=\Delta t \cdot mackeyglass\_eq(x(t+k_3), x(t-\tau), a, b)$$
% $$x(t+\Delta t) = x(t) + \frac{k_1}{6}+ \frac{k_2}{3} + \frac{k_3}{6} +
% \frac{k_4}{6}$$
```

```
% where <mackeyglass_eq.html mackeyglass_eq> is the function which return
% the value of the Mackey-Glass delayed differential equation in (1)
% once its inputs and its parameters (a,b) are provided.

%% Input parameters
a        = 0.2;     % value for a in eq (1)
b        = 0.1;     % value for b in eq (1)
tau      = 17;      % delay constant in eq (1)
x0       = 1.2;     % initial condition: x(t=0)=x0
deltat   = 1;       % time step size (which coincides with the integration step)
sample_n = 999; % total no. of samples, excluding the given initial condition
interval = 1;       % output is printed at every 'interval' time steps

%% Main algorithm
% * x_t            : x at instant t         , i.e. x(t)        (current value of x)
% * x_t_minus_tau  : x at instant (t-tau)   , i.e. x(t-tau)
% * x_t_plus_deltat : x at instant (t+deltat), i.e. x(t+deltat) (next value of x)
% * X              : the (sample_n+1)-dimensional vector containing x0 plus all other
computed values of x
% * T              : the (sample_n+1)-dimensional vector containing time samples
% * x_history      : a circular vector storing all computed samples within x(t-tau) and
x(t)

time = 0;
index = 1;
history_length = floor(tau/deltat);
x_history = zeros(history_length, 1); % here we assume x(t)=0 for -tau <= t < 0
x_t = x0;

X = zeros(sample_n+1, 1); % vector of all generated x samples
T = zeros(sample_n+1, 1); % vector of time samples
```

```
for i = 1:sample_n+1,
        X(i) = x_t;
        if (mod(i-1, interval) == 0),
        disp(sprintf('%4d %f', (i-1)/interval, x_t));
        end
        if tau == 0,
        x_t_minus_tau = 0.0;
        else
        x_t_minus_tau = x_history(index);
        end
        x_t_plus_deltat = mackeyglass_rk4(x_t, x_t_minus_tau, deltat, a, b);
        if (tau ~= 0),
        x_history(index) = x_t_plus_deltat;
        index = mod(index, history_length)+1;
        end
        time = time + deltat;
        T(i) = time;
        x_t = x_t_plus_deltat;
end
figure
plot(T, X);
set(gca,'xlim',[0, T(end)]);
xlabel('t');
ylabel('x(t)');
title(sprintf('A Mackey-Glass time serie (tau=%d)', tau));
```

3. mackeyglass_eq.m

```
% This function returns dx/dt of Mackey-Glass delayed differential equation
% $$\frac{dx(t)}{dt}=\frac{ax(t-\tau)}{1+x(t-\tau)^{10}}-bx(t)$$
% *Matlab code:*
function x_dot = mackeyglass_eq(x_t, x_t_minus_tau, a, b)
        x_dot = -b*x_t + a*x_t_minus_tau/(1 + x_t_minus_tau^10.0);
end
% <mackeyglass.html _back to main_>
```

4. mackeyglass_rk4.m

```
% This function computes the numerical solution of the Mackey-Glass
% delayed differential equation using the 4-th order Runge-Kutta method

% $$k_1=\Delta t \cdot mackeyglass\_eq(x(t), x(t-\tau), a, b)$$
% $$k_2=\Delta t \cdot mackeyglass\_eq(x(t+\frac{1}{2}k_1), x(t-\tau), a, b)$$
% $$k_3=\Delta t \cdot mackeyglass\_eq(x(t+\frac{1}{2}k_2), x(t-\tau), a, b)$$
% $$k_4=\Delta t \cdot mackeyglass\_eq(x(t+k_3), x(t-\tau), a, b)$$
% $$x(t+\Delta t) = x(t) + \frac{k_1}{6}+ \frac{k_2}{3} + \frac{k_3}{6} +
\frac{k_4}{6}$$

% Here is the code for <mackeyglass_eq.html mackeyglass_eq>,
% the Mackey-Glass delayed differential equation

% *Matlab code:*
function x_t_plus_deltat = mackeyglass_rk4(x_t, x_t_minus_tau, deltat, a, b)
        k1 = deltat*mackeyglass_eq(x_t,          x_t_minus_tau, a, b);
        k2 = deltat*mackeyglass_eq(x_t+0.5*k1,   x_t_minus_tau, a, b);
        k3 = deltat*mackeyglass_eq(x_t+0.5*k2,   x_t_minus_tau, a, b);
        k4 = deltat*mackeyglass_eq(x_t+k3,       x_t_minus_tau, a, b);
        x_t_plus_deltat = (x_t + k1/6 + k2/3 + k3/3 + k4/6);
end
% <mackeyglass.html _back to main_>
```

5. StartRealChua.m

```
[t,y] = ode45(@RealChua,[0 0.05],[-0.5 -0.2 0]);
plot3(y(:,1),y(:,2),y(:,3))
grid
```

6. RealChua.m

```
function out = RealChua(t,in)

x = in(1); %v_1
y = in(2); %v_2
z = in(3); %i_L

C1  = 10*10^(-9);   %10nF
C2  = 100*10^(-9);  %100nF
R = 1800;           %1.8k Ohms
G = 1/R;

%Chua Diode
R1 = 220; R2 = 220; R3 = 2200; R4 = 22000; R5 = 22000; R6 = 3300;

Esat = 9; %9V batteries
E1 = R3/(R2+R3)*Esat;
E2 = R6/(R5+R6)*Esat;

m12 = -1/R6;
m02 = 1/R4;
m01 = 1/R1;
m11 = -1/R3;

m1 = m12+m11;
```

```
if(E1>E2)
            m0 = m11 + m02;
else
            m0 = m12 + m01;
end

mm1 = m01 + m02;
Emax = max([E1 E2]);
Emin = min([E1 E2]);

if abs(x) < Emin
            g = x*m1;
elseif abs(x) < Emax
            g = x*m0;
            if x > 0
            g = g + Emin*(m1-m0);
            else
            g = g + Emin*(m0-m1);
            end
elseif abs(x) >= Emax
          g = x*mm1;
          if x > 0
          g = g + Emax*(m0-mm1) + Emin*(m1-m0);
          else
          g = g + Emax*(mm1-m0) +  Emin*(m0-m1);
          end
end

%end Chua Diode
%Gyrator
R7  = 100;  %100 Ohms
R8  = 1000; %1k Ohms
```

```
R9  = 1000; %1k Ohms
R10 = 1800;
C   = 100*10^(-9); %100nF
L = R7*R9*C*R10/R8; %18mH

%end Gyrator

% Chua's Circuit Equations
xdot = (1/C1)*(G*(y-x)-g);
ydot = (1/C2)*(G*(x-y)+z);
zdot  = -(1/L)*y;

out = [xdot ydot zdot]';
```

7. lorenz.m

```
function [x,y,z] = lorenz(rho, sigma, beta, initV, T, eps)
% LORENZ Function generates the lorenz attractor of the prescribed values
% of parameters rho, sigma, beta
%
%   [X,Y,Z] = LORENZ(RHO,SIGMA,BETA,INITV,T,EPS)
%       X, Y, Z - output vectors of the strange attactor trajectories
%       RHO     - Rayleigh number
%       SIGMA   - Prandtl number
%       BETA    - parameter
%       INITV   - initial point
%       T       - time interval
%       EPS     - ode solver precision
%
% Example.
%        [X Y Z] = lorenz(28, 10, 8/3);
%        plot3(X,Y,Z);

        if nargin<3
        error('MATLAB:lorenz:NotEnoughInputs','Not enough input arguments.');
        end

        if nargin<4
        eps = 0.000001;
        T = [0 25];
        initV = [0 1 1.05];
        end

        options = odeset('RelTol',eps,'AbsTol',[eps eps eps/10]);
        [T,X] = ode45(@(T,X) F(T, X, sigma, rho, beta), T, initV, options);

        plot3(X(:,1),X(:,2),X(:,3));
```

TNI THAI - NICHI INSTITUTE OF TECHNOLOGY สถาบันเทคโนโลยีไทย-ญี่ปุ่น

```
        axis equal;
        grid;
        title('Lorenz attractor');
        xlabel('X');
        ylabel('Y');
        zlabel('Z');
        x = X(:,1); y = X(:,2); z = X(:,3);
        return
end

function dx = F(T, X, sigma, rho, beta)
% Evaluates the right hand side of the Lorenz system
% x' = sigma*(y-x)
% y' = x*(rho - z) - y
% z' = x*y - beta*z
% typical values: rho = 28; sigma = 10; beta = 8/3;

      dx = zeros(3,1);
      dx(1) = sigma*(X(2) - X(1));
      dx(2) = X(1)*(rho - X(3)) - X(2);
      dx(3) = X(1)*X(2) - beta*X(3);
      return
end
```

8. chaoticmap.m

```
%THIS PROGRAM IS WRITTEN FOR DEMONSTRATION OF QUADRATIC MAP (CHAOTIC MAP)
%AND ITS TRAJECTORY,2D MAPPING AND AUTOCORRELATION.
clc; close all; clear all;

A = 4;  B = .5;  phin = 0.15;  phi(1) = B - A*(phin^2);
for ib = 2:1:1000
        phi(ib) = B - (A*(phi(ib-1).^2));
end

AST = xcorr(phi,phi);
for tt = 1:1:length(phi)-2
        XX(tt) =  phi(tt+2);
        YY(tt) =  phi(tt+1);
        ZZ(tt) =  phi(tt);
end


figure(1); plot(phi); title('\bf QUADRATIC map');
xlabel('\bf Time series'); ylabel('\bf Amplitude');
figure(2); plot3(ZZ,YY,XX,'r.'); title('\bf Pseudo phase space trajectories'); grid on;
figure(3); plot(ZZ,YY,'k.'); title('\bf Mapping'); xlabel('\bf X(n)'); ylabel('\bf X(n+1)');
figure(4); plot(AST); title('\bf Auto correlation'); xlabel('\bf Time');
ylabel('\bf correlation value');
% vary B(bifurcation parameter) = .25 P1,.32 P2,.35 &.37 HIGHER periods,.38
% CHAOTIC OSCILLATIONS
```