Generating Fuzzy Rules by Learning from Examples

CONTENT

1. INTRODUCTION

This paper provide the general method to generate fuzzy rule from numerical data. This method can be combine to both numerical data and linguistic information, which gives a fuzzy rule base.

This paper is divided into four sections. In Section 2, we propose a five step procedure for generating fuzzy rules from numerical data pairs. In Section 3, we prove that the resulting mapping is capable of approximating any nonlinear continuous function on a compact set to arbitrary accuracy. In Section IV, we are going to deal with the application of our new method.

Section -1)

This section deals with the advantage of combining both numerical data and linguistic information to generate fuzzy rule base. This fuzzy rule base can be applied to replace human controller as shown in below fig.
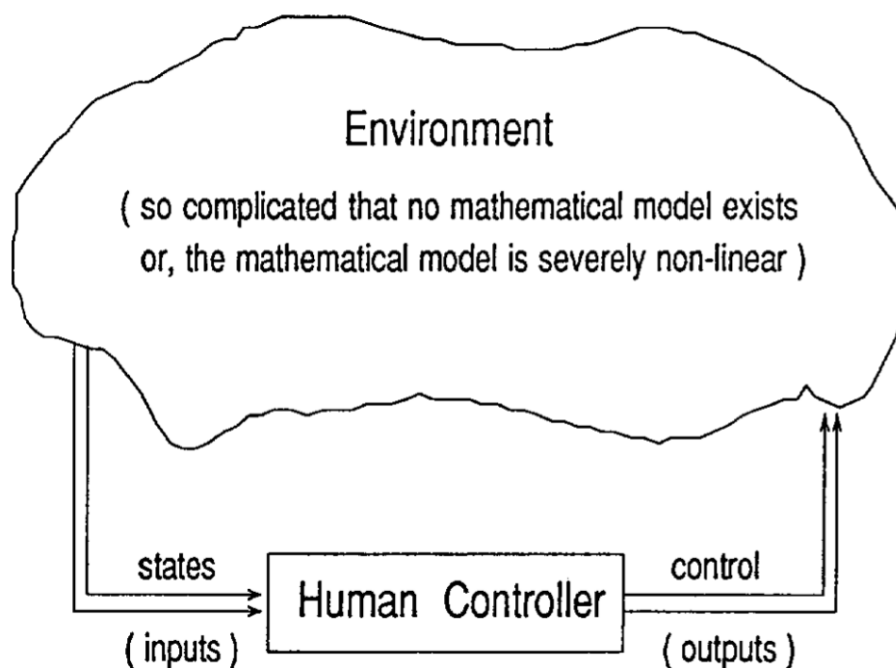
Section -2)

This section deals with generate fuzzy rule from numerical data.
- Division of input and output spaces of given numerical data into fuzzy.
- Generation of fuzzy rules from given data.
- Assignment of degree to each of the generated rule
- Create a combined fuzzy rule base based on both the generated rules and linguistic rules of human expert.
- Determination of mapping from input space to output space based on the combined fuzzy rule base using a defuzzifying procedure.

2. GENERATING FUZZY RULES FROM NUMERICAL DATA.

We are given a set of desired input-output data pairs

$(x1\,(1)\,,\,x2\,(1)\,,\,y\,(1))$, $(x1\,(2)\,,\,x2\,(2)\,,\,y\,(2))$, … … ….

The task is to generate a set of fuzzy rules from the desired input-output pairs and use these fuzzy rules to determine the mapping $f : (x1,\,x2) \rightarrow y$

Our approach consists of following five steps for generating fuzzy rule

a. Divide input and output space into fuzzy regions Suppose that domain interval of $x1$, $x2$ and y are

$x1 \rightarrow [x1\,-,\,x1\,+]$, $x2 \rightarrow [x2\,-,\,x2\,+]$ and $y \rightarrow [y-,\,y+]$

Where the domain interval of variable means that most probably variable will lie in this interval. Divide each domain interval into 2N+1 region and assign each region a fuzzy membership function. Then assign the symbolic representation for each membership function.

b. Generating fuzzy rule from given data pairs

First, we have to determine the degree of given inputs $x1\,(i)$, $x2\,(i)$ and output $y(i)$ in different regions. Second, assign a given inputs or outputs to the region with maximum degree.
In system identification and modeling problems there are lots of data pairs, and each data pair generates one rule, it is highly probable that there will be some conflicting rules, i.e., rules that have the same IF part but a different THEN part. One way to resolve this conflict is to assign a degree to each rule generated from data pairs, and accept only the rule from a conflict group that has maximum degree. In this way not only is the conflict problem resolved, but also the number of rules is alleviate. In this paper, the degree of rule is defined as product of degree of data pairs that generate the rule.

In practical situation, we often have some a priori information about the data pairs. For example, if we let an expert check given data pairs, the expert may suggest that some are very useful and crucial, but others are very unlikely and may be caused just by measurement errors.

c. Assign a Degree to Each Rule

   We can therefore assign a degree to each data pair that represents our belief of its usefulness. In this sense, the data pairs constitute a fuzzy set, i.e., the fuzzy set is defined as the useful measurements; a data pair belongs to this set to a degree assigned by a human expert. Hence based on the expert opinion for good data we assign higher degrees, and for bad data we assign lower degrees. In this way, human experience about the data is used in a common base as other information.
   For example-

   Rule 1 has degree, $D(Rule1) = mB1 (x1) mS1 (x2) mCE (y) = 0.8 * 0.7 * 0.9 = 0.504$

   Rule 2 has degree $D(Rule2) = mB1 (x1) mCE (x2) mB1 (y) = 0.6 * 1 * 0.7 = 0.42$

d. Combination of fuzzy rule base
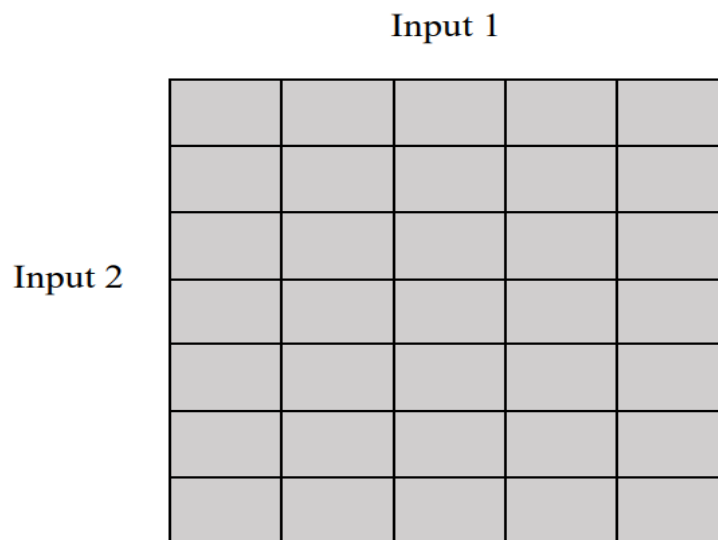   The form of combined fuzzy rule base is shown in below figure.

Fig. 2: Combined fuzzy rule base

   A combined fuzzy rule base is assigned rules from generated numerical data or linguistic rules. If there is more than one rule in one box of the fuzzy rule base, then use the rule that has maximum degree. In this way, both numerical and linguistic information are codified into a common framework-the combined fuzzy rule base.

e. Generation of mapping from inputs to output

To generate a mapping from inputs to output, defuzzification and aggregation are used. Defuzzification is used to convert fuzzy output into crisp output for each rule. The crisp output of each rule is then aggregated into a final crisp output with help of output degrees for each rule.

Defuzzification:

a. Centroid defuzzification method is used. The centroid of the output fuzzy subspace of a rule is defined as the minimum of the core of that fuzzy subspace. Therefore, the centroid is the minimum value of the output values in the output fuzzy subspace of the rule at which the membership value is 1.

b. Other methods of defuzzification may also be used according to the nature of problem that is being dealt with. The centroid can be taken as the minimum value corresponding to the maximum membership degree. For limited range practical data, this proves computationally efficient & more effective method.

Aggregation:

Defuzzification gives crisp output for each rule. To get overall crisp output, these defuzzified outputs are aggregated. For aggregation, output degree for each rule is calculated to represent the extent to which the rule contributes towards the final output.

3. Application to Truck backer-Upper Control

The control for Backing a truck to a loading dock is a nonlinear control problem for which no traditional control system design methods exist. As mentioned in the paper Nguyen and Widrow develop a neural network controller for the truck backer-upper problem and Kong and Kosko propose a fuzzy control strategy for the same problem.

The numerical/neural network controller only uses numerical data, and cannot utilize linguistic rules determined from expert drivers; on the other hand, the fuzzy controller only uses linguistic rules, and cannot utilize sampled data.

In the given example, the truck position is exactly determined by the three state variables, two input variable and one output variable
Whose inputs are $\phi \in [-90°, 2700]$ and $x \in [0,20]$, and whose output is $\theta \in [-40°, 40°]$, such that the final states will be $(X_f, \phi_f) = (10, 90°)$.
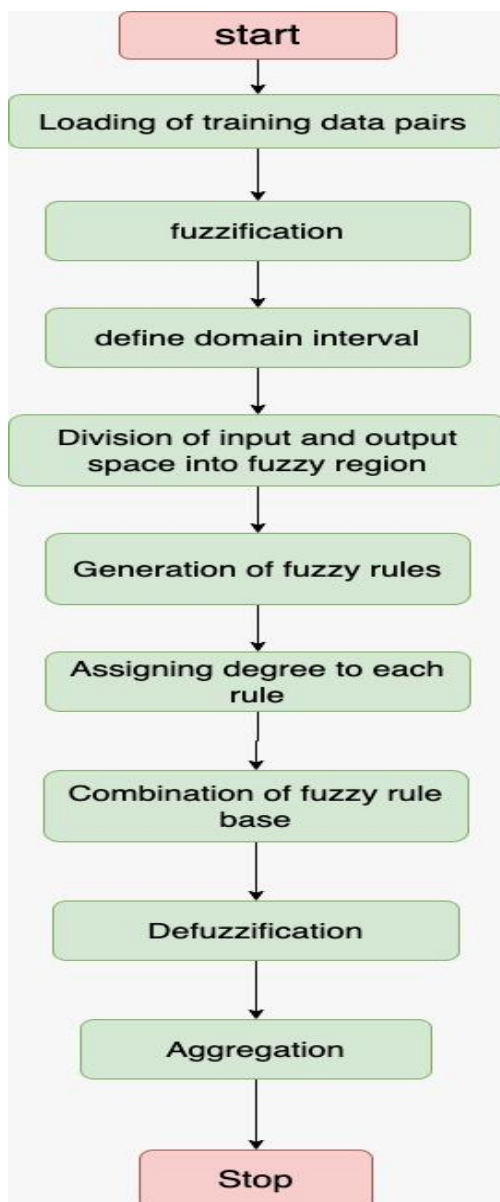
For finding out the next position of truck, following equations of dynamic state representation of truck are used and this is approximated kinematics:

$$x(t + 1) = x(t) + \cos[\emptyset(t) + \theta(t)] + sin[\ \theta(t)]sin[\emptyset(t)]$$

$$y(t + 1) = y(t) + \sin[\emptyset(t) + \theta(t)] - sin[\ \theta(t)]cos[\emptyset(t)]$$
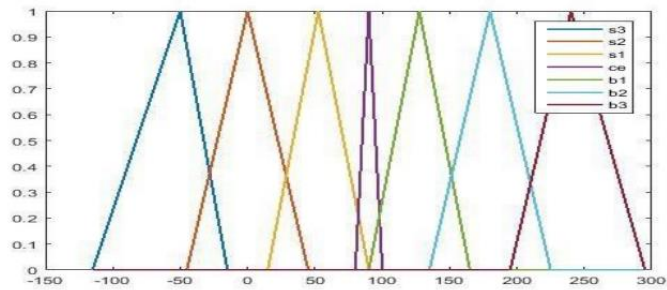
$$\emptyset(t + 1) = \emptyset(t) - \sin{-1}[\ 2\sin\{\theta(t)\}/b]$$

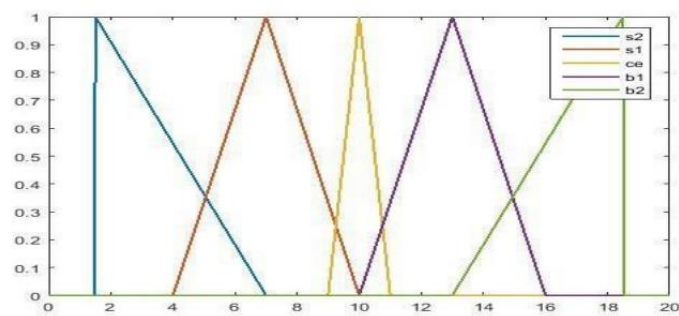The flowchart for Truck backer-Upper Control is shown below
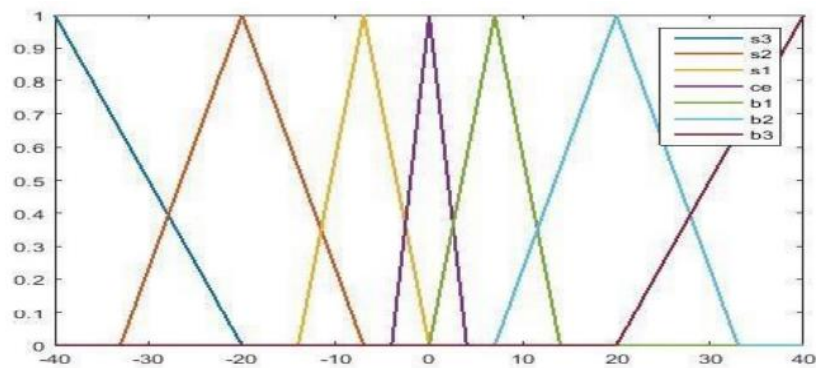

4. RESULTS OF TRUCK BACKER-UPPER PROBLEM

We are going to divide our inputs into fuzzy regions. The regions used are shown in below figure.

(a)

(b)

(c)

Fuzzy region (a) Input x (b) Input phi (c) Output theta
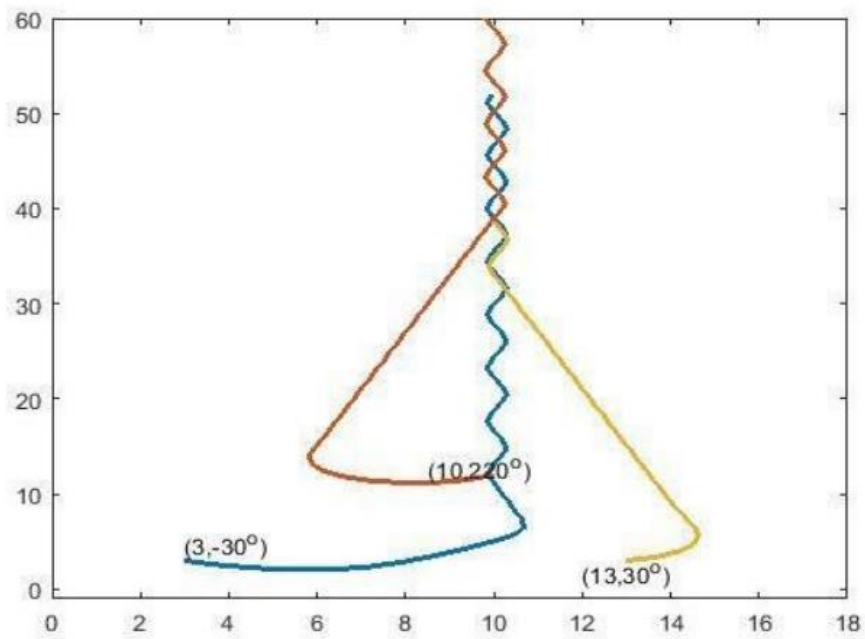
Numerical Control and simulation result:

For this simulation purpose, we used the 24 sets of the fuzzy rules as generated from the data after removing conflicting rule which are mentioned in the below Table

Input: x

Input: Phi

	S2	S1	CE	B1	B2
S3	S2	S3			
S2	S2	S3	S3	S3	
S1	B1	S1		S3	S2
CE	B2	B2		S2	S2
B1	B2	B3		B1	S1
B2		B3	B3	B3	B2
B3				B3	B2

Truck trajectory using numerical data only is shown below

Numerical and linguistic information control and simulation result:

For this simulation purpose, we used 27 sets of the fuzzy rules as generated from the data after removing conflicting rule, which are mentioned in the below table
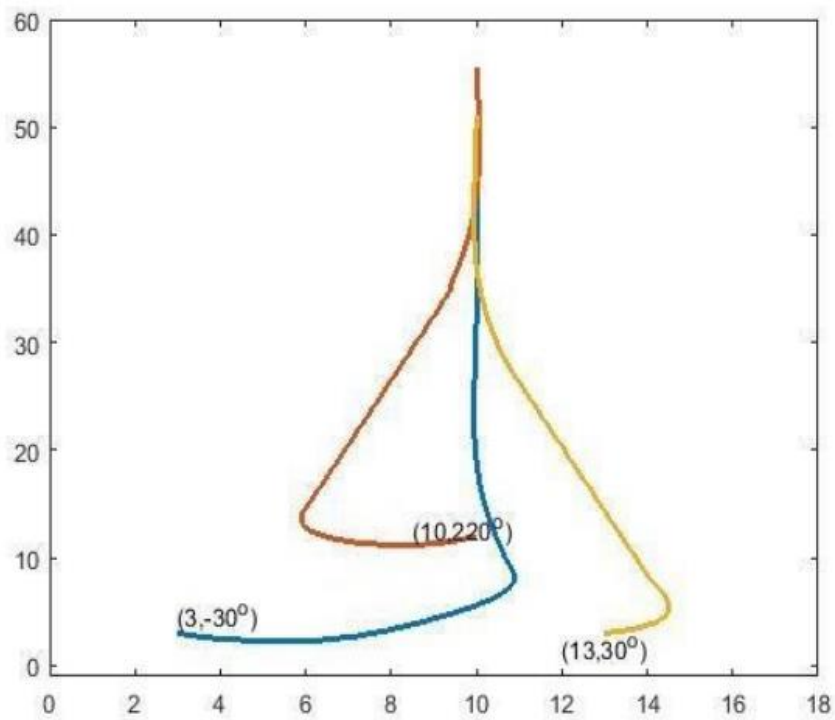
Rule base for Numerical and linguistic information control

Input: x

		S2	S1	CE	B1	B2
	S3	S2	S3			
	S2	S2	S3	S3	S3	
	S1	B1	S1	S2	S3	S2
Input: Phi	CE	B2	B2	CE	S2	S2
	B1	B2	B3	B2	B1	S1
	B2		B3	B3	B3	B2
	B3				B3	B2

Truck trajectory using both numerical data and linguistic information are shown in below Fig.
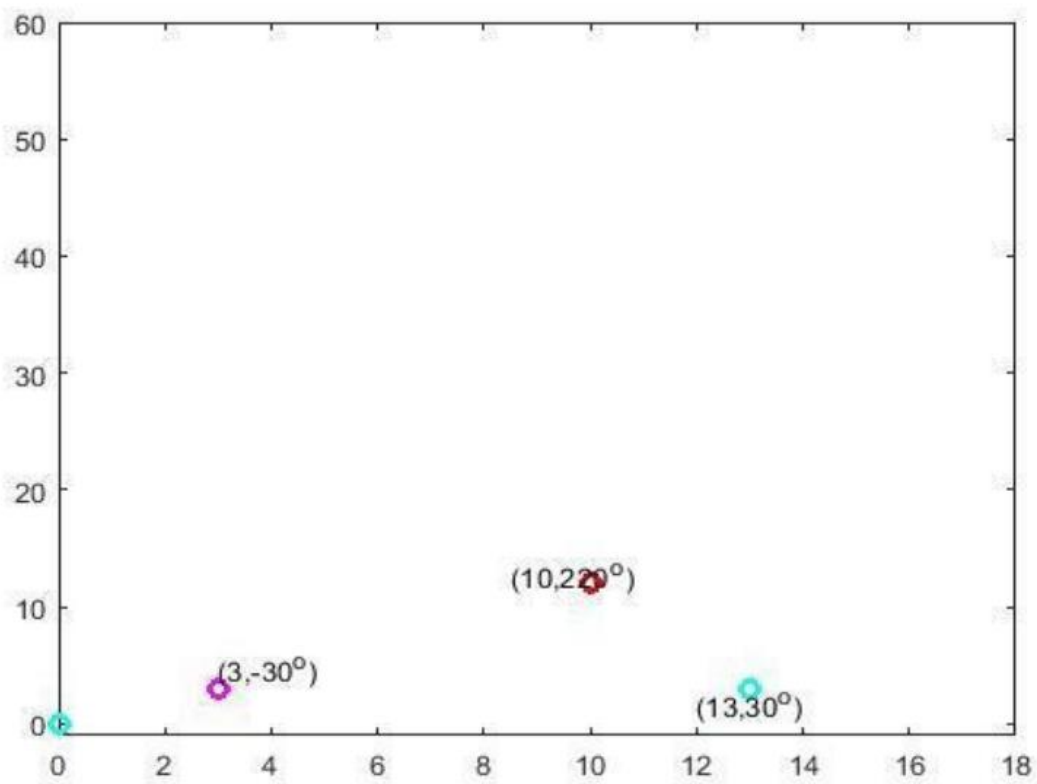
Selected linguistic rule and simulation result:

For this simulation purpose, we used the 3 sets of selected fuzzy rules as generated from the data which are shown in below table

<div align="center">

Input: x

</div>

		S2	S1	CE	B1	B2
Input: Phi	S3					
	S2					
	S1			S2		
	CE			CE		
	B1			B2		
	B2					
	B3					

Truck trajectory using selected linguistic information is shown in below Fig.

5. Application to Time Series Prediction

Let z(k) where k = 1, 2,......n be time series. The problem of time series prediction can be formulated as: z(k− m+1), z(k− m+ 2).....z(k) , and determine z(k+1) where, m = number of inputs to determine output.

Mackey-Glass Chaotic series:

Chaotic time are generated from deterministic nonlinear system and are sufficiently that they appear to be random time series, however there are underlying deterministic maps that generate the series, chaotic time series are not random time series. The Mackey-Glass chaotic time series is generated from the following delay differential:

$$\frac{d(x)}{dt} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t) \quad \text{where } \tau > 17$$

Below Fig. shows the chaotic time series when tau = 17.
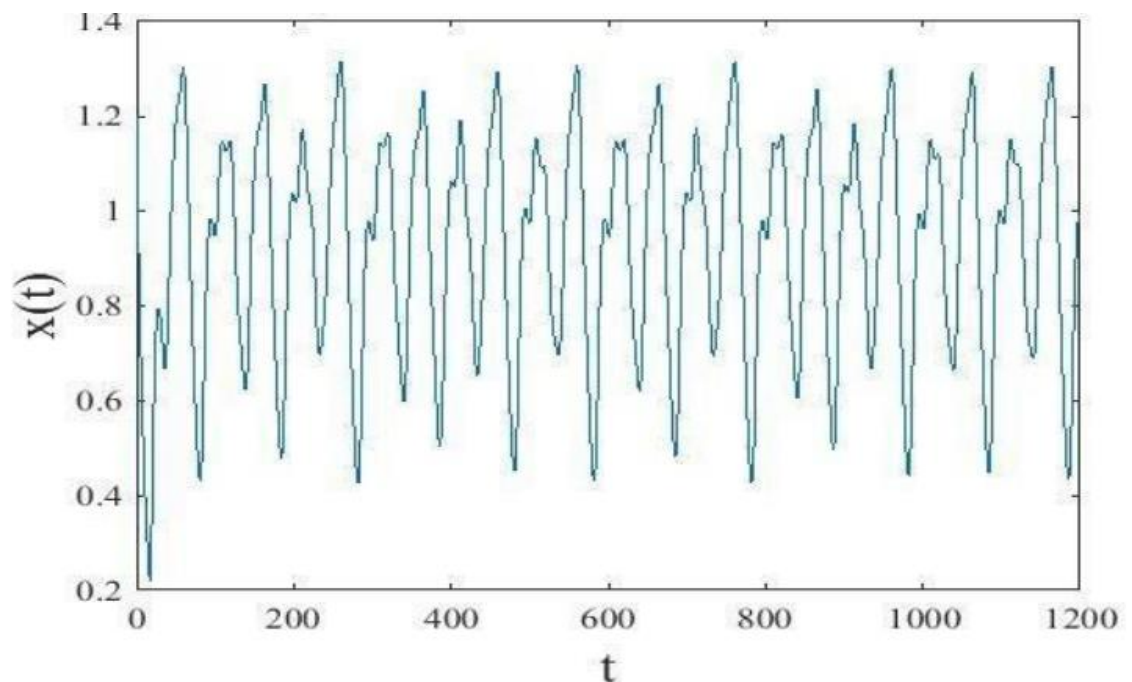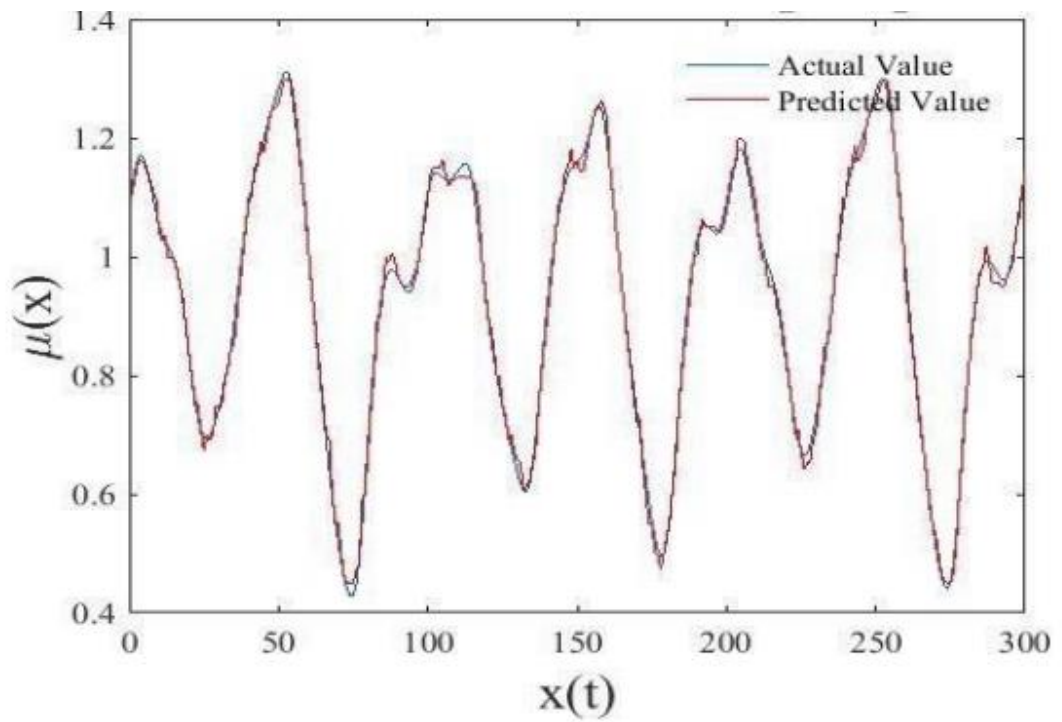
6. RESULTS OF TIME SERIES PREDICTION

For simulation of this paper, the value of m = 9 and l =1 have been chosen. m is the number inputs used to predict the value of next time instance. Above fig. shows 1200 points of chaotic time-series. The test has been performed for various combinations of training data and test data.
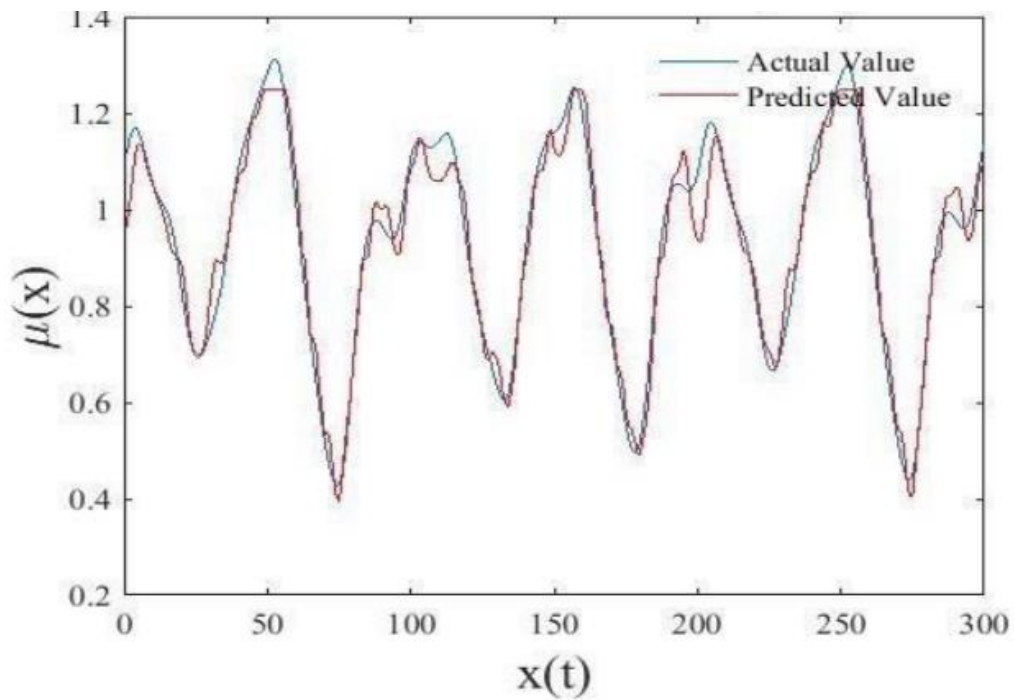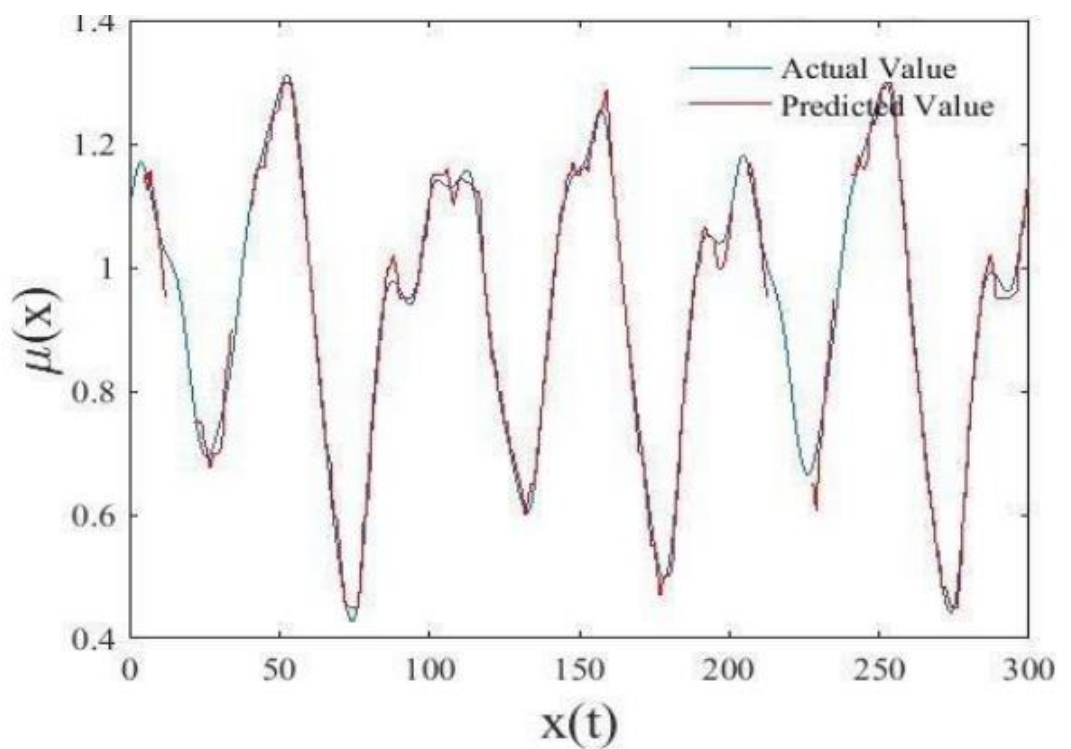

In above fig.: Time-Series Prediction from x(701) to x(1000) using numerical fuzzy predictor when 700 training data (from x(1) to x(700) are used) and number of fuzzy region is 9
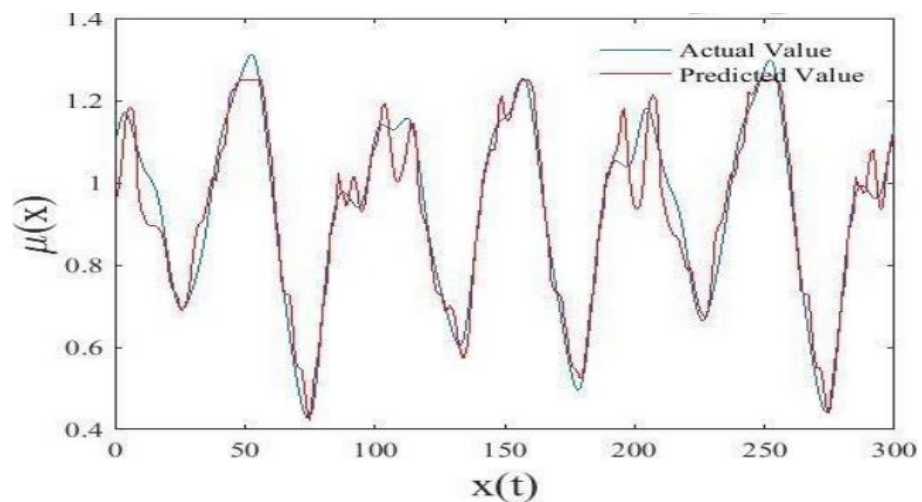
In above Fig. : Time-Series Prediction from x(701) to x(1000) using numerical fuzzy predictor when 200 training data (from x(501) to x(700) are used) and number of fuzzy region is 9

In above fig. 10: Time-Series Prediction from x(701) to x(1000) using numerical fuzzy predictor when 700 training data (from x(1) to x(700) are used) and number of fuzzy region is 29

In above fig. 11: Time-Series Prediction from x(701) to x(1000) using numerical fuzzy predictor when 200 training data (from x(501) to x(700) are used) and number of fuzzy region is 29