FUZZY SYSTEM PROJECT REPORT

Habibollah Naeimi (SN.:9663124)

Instructor: Dr. Shahbazian

Petroleum University of Technology

Ahwaz Faculty of Petroleum

Table Look Up Algorithm

Fuzzy Systems Course

Instructor: Dr.Shahbazian

Designing Fuzzy System using Table Look Up Scheme Report

Habibollah Naeimi

Time Series Prediction.

Time series prediction is an important practical problem. Applications of time series prediction are in economic, and business planning, inventory and production control. Here we use the fuzzy system designed by the table look-up scheme to predict the Mackey-Glass chaotic time series generated by the following delay differential equation.

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$$

To write MATLAB code easier, we write this equation for τ = 30 as follows:

$$x(t+1) = \frac{0.2x(t-30)}{1+x^{10}(t-30)} + 0.9x(t)$$

The implementation of this equation is given in the " 2nd Part: Data\ Sampling" part of the code. This equation generates sampling Data and Training Data. Here is the code:

```
SAMPLES = rand(1,32);
for v=33:1033
    SAMPLES(v) = 0.2*SAMPLES(v-31)/
(1+(SAMPLES(v-31)^10))+0.9*SAMPLES(v-1);
```

```
end

SAMPLES = SAMPLES(34:end);
```

Rule Base.

Rule Base is the most complex and essential part of the program. In this method, a fuzzy rule base is created based on the available data from the system. Afterward, based on that rule base, a fuzzy system will be designed to model the system.

To form a rule base, we must first identify the member functions of data pairs. This code gives us the number and amount of the membership function using the slope, initial values, and the type of membership function.

```
Step = (UpBnd-LowBnd)/(Num-1);
        for i=1:Num
            Center = LowBnd+(i-1)*Step;
            Y(i) = trimf(x,[Center-Step,Center,
Center+Step]);
        end
```

This code is a created function named “ MFDetector.m ”.

After detecting membership function number and value, we use following code as a function to find the rules. The function is named “ RuleFinder.m ”.

```
e1 = numel(MFNum);
e2 = size(Pair,1);
```

```
MVal = zeros(1,e1);
MFun = zeros(1,e1);
Rule = zeros((size(Pair)));
RuleMemValue = zeros((size(Pair)));
    for i=1:e2
        for j=1:e1
            [MVal(j) MFun(j)] =
MFDetector(Pair(i,j),MFNum(j),LowBnd(j),UpBnd(j)
,MemFunTyp(j));
        end
        Rule(i,:)=MFun;
        RuleMemValue(i,:)=MVal;
    end
```

A brief explanation for this code is that our rules consist of membership numbers of functions derived from the previous function. And rule values as well.

The rules may have conflicts. So we use the “ConflictChecking.m” code to delete the conflicting rules.

```
k = 1;
    while ~isempty(Rule)
        First = repmat(Rule(1,1:end-1),
size(Rule,1),1);
        Rule_B = Rule(:,1:end-1);
        Index = Rule_B == First;
        Index = find(sum(Index,2) ==
numel(Rule_B(1,:)));
        [RuleDegree(k) Rule_Index] =
max(prod(Rule_MV(Index,:),2));
```

```
        Rule1(k,:) = Rule(Index(Rule_Index),:);
        Rule1MVlu(k,:) =
Rule_MV(Index(Rule_Index),:);
        k = k+1;
        m = 1;
        RuleBuffer = [];

        for n=1:size(Rule,1)
            if isempty(find(Index==n))
                RuleBuffer(m,:) = Rule(n,:);
                m = m+1;
            end
        end

        Rule = RuleBuffer;
    End
```

Here we calculate the degrees of the rules by membership function values, and by that, we can detect and delete conflicting rules.

Fuzzy System

Now we have the rules, and we form the rule base. And in the next step, we create a fuzzy system with the name of Time Series Prediction. A Mamdani fuzzy system with MIE.

Here is the code:

```
TSP = newfis('Time Series Prediction');
```

We should add Inputs and Outputs in a for loop with this code:

```
  TSP =
addvar(TSP,'input',Name,[LowBnd(i),UpBnd(i)]);
```

```
% For Triangular Membership Function.
            Step = (UpBnd(i)-LowBnd(i))/
(MemFuNu(i)-1);
            for j = 1:MemFuNu(i)
                Center = LowBnd(i)+(j-1)*Step;
                MemFName = [Name '_MF'
num2str(j)];
                TSP =
addmf(TSP,'input',i,MemFName,'trimf',[Center-
Step,Center,Center+Step]);
            end
```

And after all this, we can add our rule base to the fuzzy system using this code:

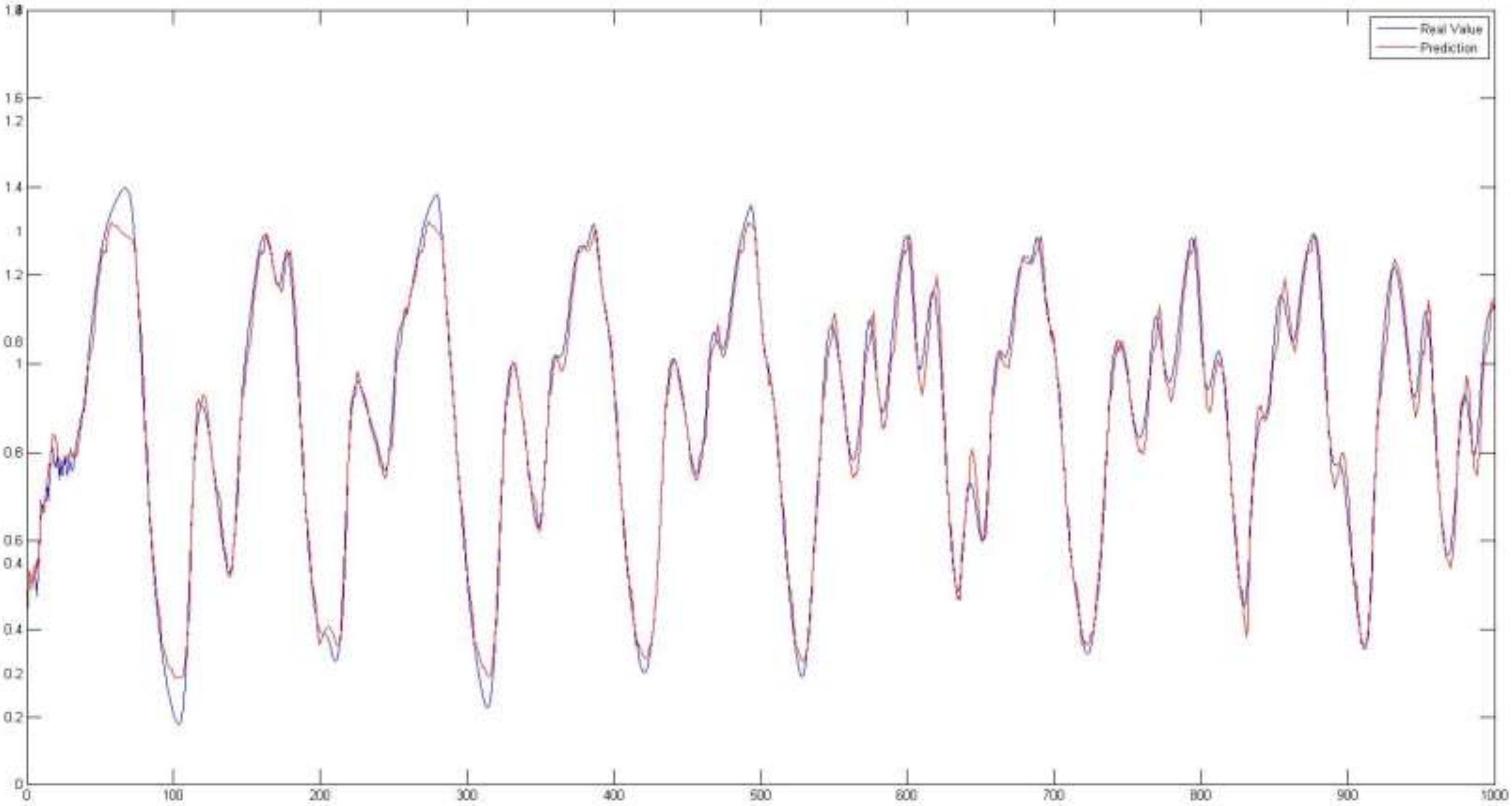
```
TSP = addrule(TSP,RulesList);
```

Results

This is the result of the program for these given parameters.

Fuzzy System detailes:

1. Name	Time Series Prediction
2. Type	Mamdani
3. Inputs/Outputs	[6 1]
4. NumInputMFs	[18 18 18 18 18 18]
5. NumOutputMFs	18
6. NumRules	287
7. AndMethod	min



8. OrMethod max

9. ImpMethod min

10. AggMethod max

11. DefuzzMethod centroid

Member ship Type: Gaussian

Number of All Rules: 500

Number of No Conflicting Rules: 287

Number of Conflicting Rules: 213

Error of Prediction = 0.26932%