Soft Computing Methods and Applications

Lab Exercise and Assignment #6 (2021)

Fuzzy inference systems

1. Clone your Assignment 05 as a new Assignment 06.

2. If you didn't implement bool ShowMFCure (ShowSeries) flag and your fuzzy set always shows series points on the chart, implement the flag to let data storage of series and line display optional. By default, the series of a fuzzy set is not added and calculated nor displayed in the chart area of the universe. Therefore, the function that updates series points is by default not invoked. Add the flag (bool type) to let user update the property.

3. Extend the functionalities of the FuzzySet set class to support the development of fuzzy inference systems. Add required virtual and overridden functions and/or read-only properties: e.g.,

```
public virtual double MaxDegree{ get{…} }

public virtual double COACrispValue{ get{…} }

public virtual double BOACrispValue{ get{…} }
```

   In addition, to facilitate Tsukamoto inferencing, add another virtual function GetUniverseValueForADegree() to return the intersected value for a computed firing strength (degree). You need to design the computation procedures to cover different kinds of monotonic FS member functions.

```
public virtual double GetUniverseValueForADegree( double degree ) { … }
```

4. In the previous assignment, the list of user-specified fuzzy if-then rules is defined in your UI form class. Now it is the time to hand over the list to various fuzzy inference systems. Aggregate objects of the defined rules to define different fuzzy inference systems:

```
class MamdaniFuzzySystem

class SugenoFuzzySystem

class TsukamotoFuzzySystem
```

   Note that the IfThenFuzzyRule class is dedicated to Mamdani system; therefor it has function

   GenericFuzzySet FuzzyInFuzzyOutInferencing(List<GenericFuzzySet> condition) ,

   doing fuzzy-in-fuzzy-out inference.

5. In previous assignment, you have defined interface to let user construct a set of IfThenFuzzyRules and perform a single condition inferencing to get a single result, which is a union of multiple cut- or scaled- fuzzy output fuzzy sets. In this assignment, a complete crisp-in-crisp-out inferencing on all combinations of input universe values that yields a complete input-output map is required. Therefore, in addition to the function for FuzzyInFuzzyOut function, a CrispInCrispOut function is required for Mamdani system that aggregates the inferencing results from all rules:

```
Public class MamdaniFuzzySystem
{
       IfThenFuzzyRules[] allRules;
       // constructor …
        …
       public GenericFuzzySet FuzzyInFuzzyOutInferencing(
              FuzzySet[] conditions ) { … }
       public double CrispInCrispOutIferencing(
              double[] conditions, DefuzzificationType type )
              {
              // ask each rule do inferencing and aggregate their results (cut- or
       scaled- FSs.
              // return crisp value based on user’s selection
              }
}

public enum DefuzzificationType { COA, BOA, … }
```

To facilitate the crisp inferencing, you need to add an additional function to do crisp-in-fuzzy-out inferencing in your IfThenFuzzyRule class. For example,

```
       GenericFuzzySet CrispInFuzzyOutInferencing( doule[] conditions) {…}
```

6. Following the lab instructions to design UIs for inference system construction. The UIs will let the users define a set of fuzzy if-then rules. In addition, defuzzification mode can be selected to obtain crisp outputs.

7. Although user should be able to arbitrarily define output equations of a Sugeno rule, the UI design is complicated. Instead, we will hard-code the output equations for user to select in constructing their Sugeno rules. You are asked to define a SugenoIfThenFuzzyRule class to incorporate these hard-coded functions for selection. Moreover, proper UIs should be provided for the user to define output equations on output universe. Note that the conclusion part of a Sugeno rule is not a fuzzy set;

instead, it will be an index to one equation of our hard-coded equations. The output values of the hard-coded equations are returned from this class-scoped (static) function:

```
public static double GetOutputValue( double[] inputs, int equationID )
{
    switch(equationID )
    {
        case 0:  // Y=0.1X+6.4
            return 0.1 * inputs[0] + 6.4;
            break;
        case 1: // Y=0.5X+4
            return 0.5 * inputs[0] + 4;
            break;
        …
    }
}
```

8. Therefore, the designed SugenoIfThenFuzzyRule class should be

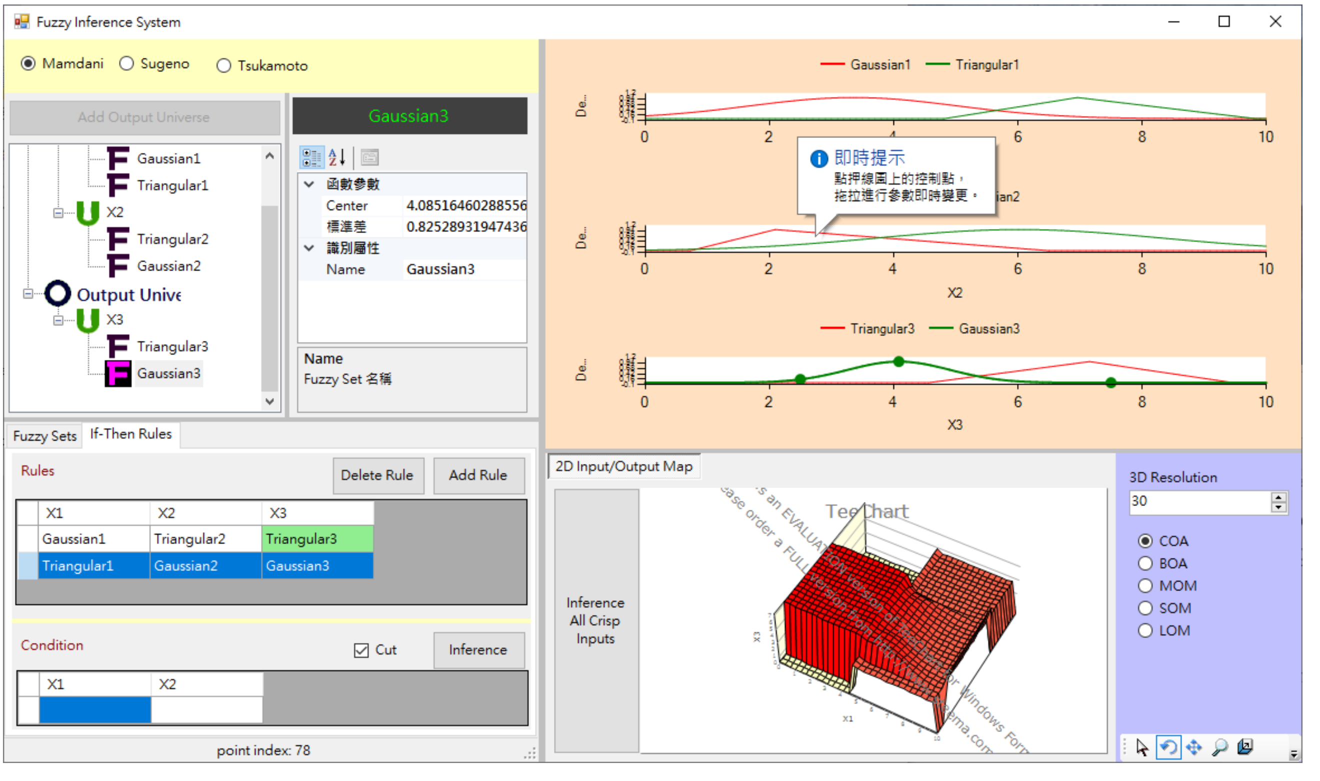
```
class SugenoIfThenFuzzyRule
{
    public static double GetOutputValue( double[] inputs, int equationID )
    { … }


    List<GenericFuzzySet> antecedents = new List<GenericFuzzySet>();
    int conclusionEquationID;

    public double CrispInCrispOutInferencing(
        double[] conditions, out double strength )
    {
        // determine firing strength
        // evaluate output value
        // return weighted (strengthed) output value
        …
```

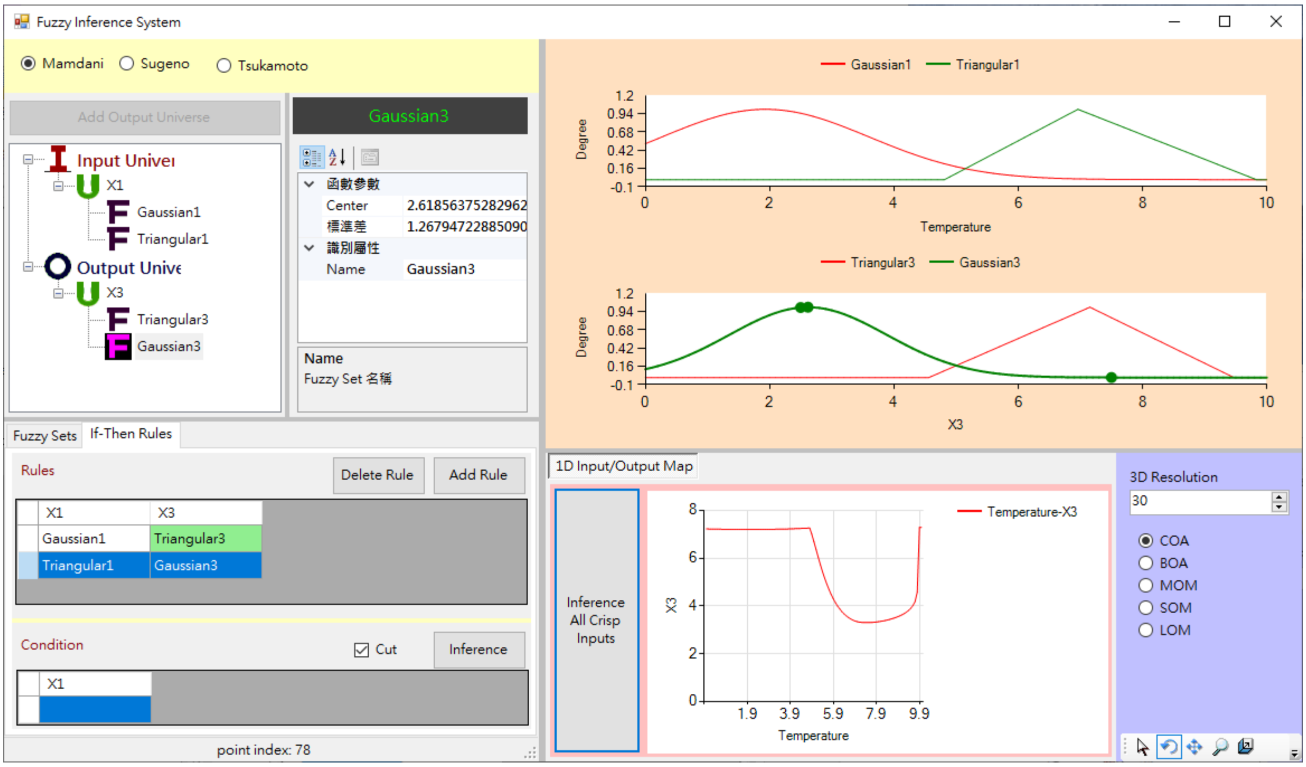
Note that no fuzzy-in referencing in a Sugeno model, neither in a Tsukamoto model. Moreover, since the Sugeno or Tsukamoto system may select averaged output as the inferencing result, we need to return both the firing strength and the weighted output value. The “out” modifier on argument “double strength” states that strength is an output argument.

9. Similarly, you should define a TsukamotoIfThenRule class for modeling a Tsukamoto rule. The only difference might be the conclusion part, which should be a monotonic FS. You need to define constructors for both classes to ask for antecedent and conclusion data. The crisp-in-crisp-out

inferencing in Tsukamoto rule is similar to the one in Sugeno model where both firing strength and weighted output value should be returned to Tsukamoto system.

10. Provide UIs for the users to execute automatic crisp-in-crisp-out inferencing and obtain the output response lines or surfaces (on 3D chart). MS Chart does not support 3-D surface charting. You can download a trial version of TeeChart component from Steema Inc. Alternatively, you can check our bulletin board for a binary library of TeeChart. Refer to the lab instructions of implementing a 3D surface plot. For a crisp value inferencing, the condition turns out to be an array of real numbers of double type. Notice that, in this crisp-input case, no intersection of two fuzzy sets (a singleton fuzzy set and antecedent fuzzy set) and the maximal degree of it are required for inferencing. We simply use the given crisp input value to evaluate its membership degree, which turns out to be the firing strength of the antecedent term for the given crisp value and the rule.



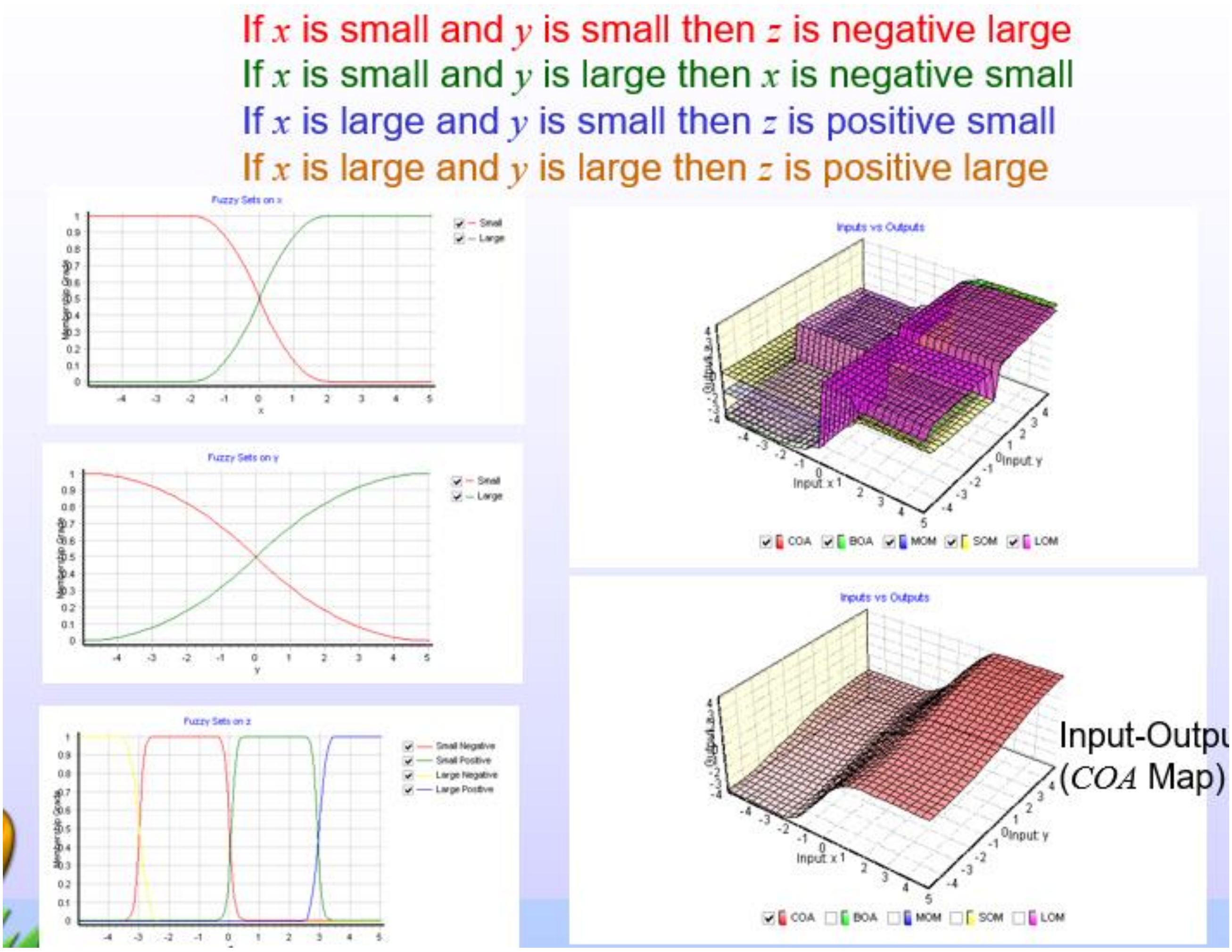


11. Add model save and read capability, as possible as you could. This will save the tedious construction steps. Run your application to construct fuzzy inference systems and inference the crisp input-output response as shown in the following examples.

12. Video tape the execution of each model to show the modeling and inferencing capability of your App. You don't need to video tape detailed construction steps. Instead, video tape the constructed model and inferencing results, by browsing the tree view, property grid, and interactively rotate the 3D response surface to demonstrate your app did generate the model and inference the results.

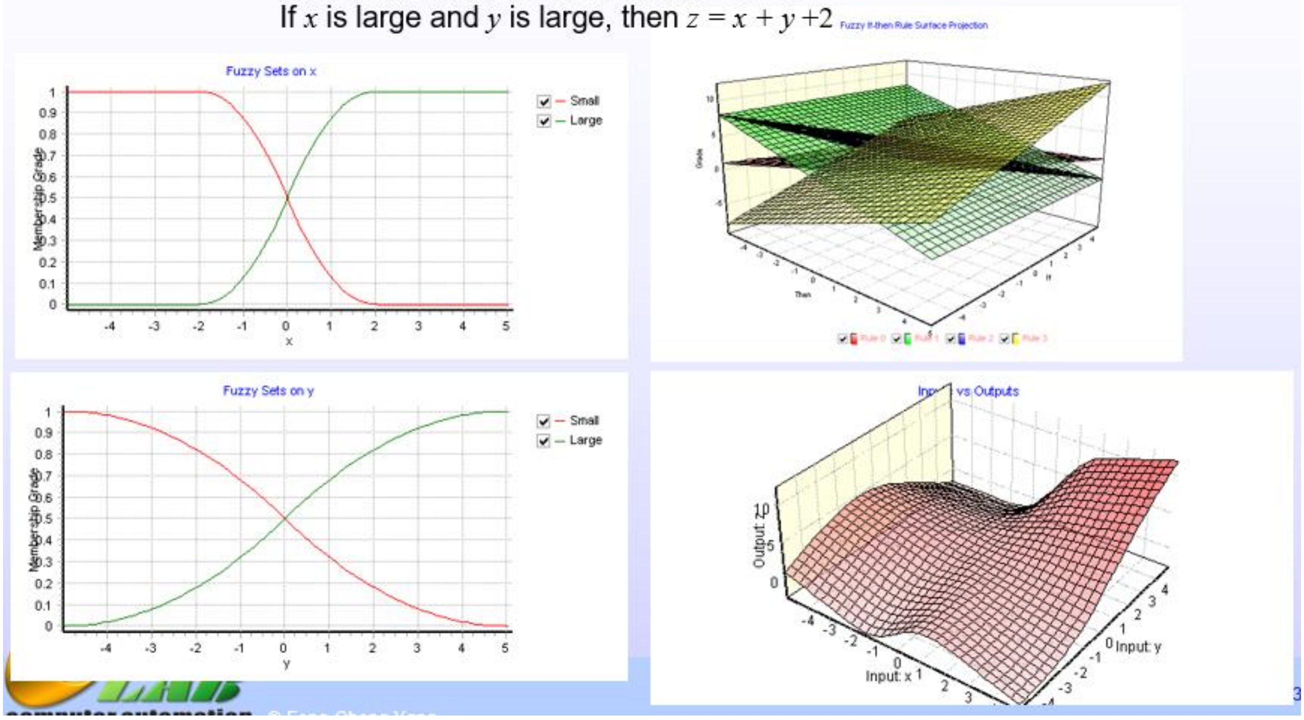
13. If your submission exceeds the upload limit to COOL, move your video file to cloud and submit a text file indicating the download link. Submit the link information as a text file and send it with your source code as well.

Mamdani fuzzy model shown in Example 4.2 (Fig. 4.6)



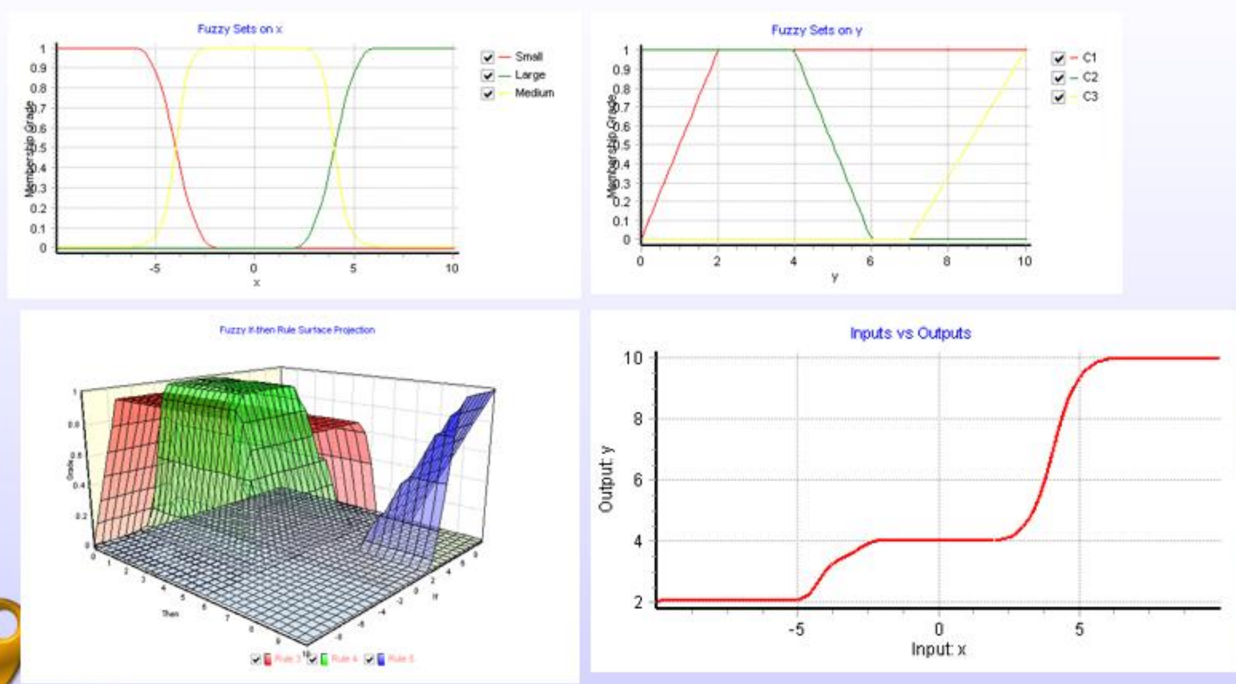
Sugeno fuzzy model shown in Example 4.4 (Fig. 4.10)

If x is small and y is small, then $z=-x+y+1$
If x is small and y is large, then $z = -y + 3$
If x is large and y is small, then $z = -x + 3$
If x is large and y is large, then $z = x + y + 2$

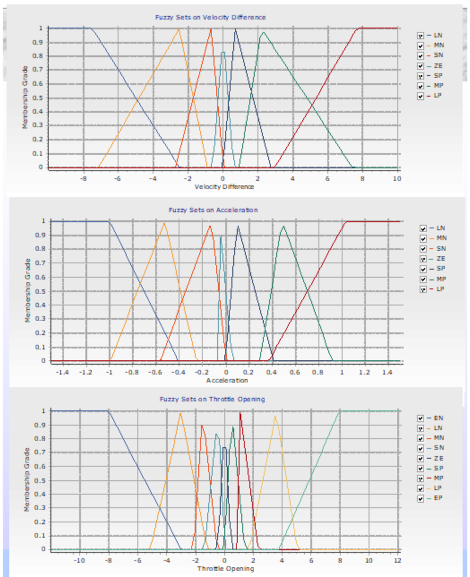


Tsukamoto fuzzy model shown in Example 4.5 (Fig. 4.12)

If x is small then y is c_1
If x is medium then y is c_2
If x is large then y is c_3



Platoons of cars (Mamdani)(optional)



Velocity Difference

Throttle / Acceleration	LN	MN	SN	ZE	SP	MP	LP
LN	ZE	MP	MP	MP	LP	LP	EP
MN	MN	ZE	ZE	MP	MP	MP	LP
SN	MN	SN	SN	SP	SP	MP	LP
ZE	LN	MN	SN	ZE	SP	MP	LP
SP	LN	MN	SN	SN	SP	SP	MP
MP	EN	LN	MN	MN	SN	SP	MP
LP	EN	LN	LN	MN	MN	SN	ZE