What Is Sugeno-Type Fuzzy Inference?

The fuzzy inference process discussed so far is Mamdani's fuzzy inference method, the most common methodology. This section discusses the so-called Sugeno, or Takagi-Sugeno-Kang, method of fuzzy inference. Introduced in 1985 [9], it is similar to the Mamdani method in many respects. The first tw o parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference betw een Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.

A typical rule in a Sugeno fuzzy model has the form

If Input 1 = x and Input 2 = y, then Output is $z = ax + by + c$

For a zero-order Sugeno model, the output level z is a constant *($a=b=0$).*

The output level z_i of each rule is w eighted by the firing strength w_i of the rule. For example, for an AND rule w ith Input 1 = x and Input 2 = y, the firing strength is

$$w_i = AndMethod(F_1(x), F_2(y))$$

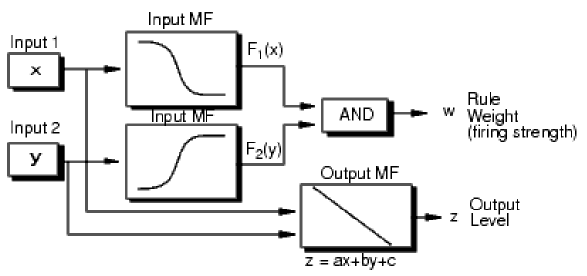
w here $F_{1,2}$ *(.)* are the membership functions for Inputs 1 and 2.

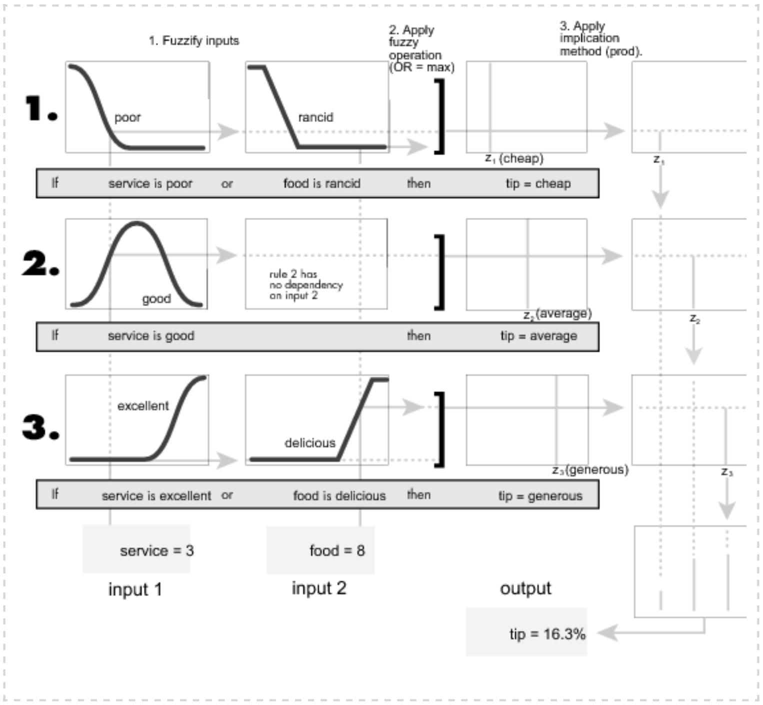
The final output of the system is the w eighted average of all rule outputs, computed as

$$\text{Final Output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i}$$

w here N is the number of rules.

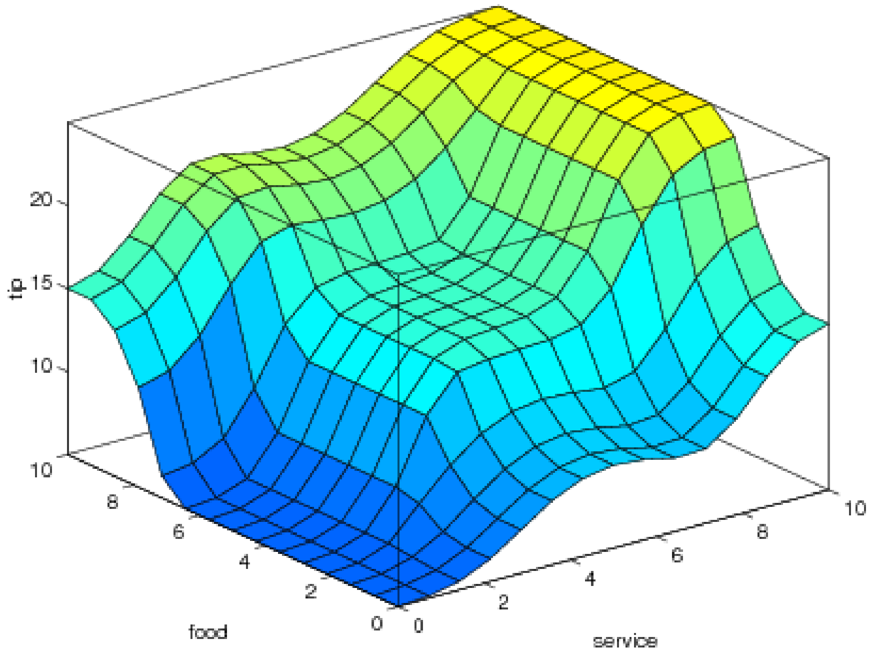
A Sugeno rule operates as show n in the follow ing diagram.




1. Fuzzify inputs
2. Apply fuzzy operation (OR = max)
3. Apply implication method (prod).
1.
poor
rancid
z₁ (cheap)
z₁
If service is poor or food is rancid then tip = cheap
2.
good
rule 2 has no dependency on input 2
z₂ (average)
z₂
If service is good then tip = average
3.
excellent
delicious
z₃ (generous)
z₃
If service is excellent or food is delicious then tip = generous
service = 3
food = 8
input 1
input 2
output
tip = 16.3%

The preceding figure show s the fuzzy tipping model developed in previous sections of this manual adapted for use as a Sugeno system. Fortunately, it is frequently the case that singleton output functions are completely sufficient for the needs of a given problem. As an example, the system `tippersg.fis` is the Sugeno-type representation of the now -familiar tipping model. If you load the system and plot its output surface, you w ill see that it is almost the same as the Mamdani system you have previously seen.

```
a = readfis('tippersg');
gensurf(a)
```



The easiest w ay to visualize first-order Sugeno systems is to think of each rule as defining the location of a moving singleton. That is, the singleton output spikes can move around in a linear fashion in the output space, depending on w hat the input is. This also tends to make the system notation very compact and efficient. Higher-order Sugeno fuzzy models are possible, but they introduce significant complexity w ith little obvious merit. Sugeno fuzzy models w hose output membership functions are greater than first order are not supported by Fuzzy Logic Toolbox softw are.

Because of the linear dependence of each rule on the input variables, the Sugeno method is ideal for acting as an interpolating supervisor of multiple linear controllers that are to be applied, respectively, to different operating conditions of a dynamic nonlinear system. For example, the performance of an aircraft may change dramatically w ith altitude and Mach number. Linear controllers, though easy to compute and w ell suited to any given flight condition, must be updated regularly and smoothly to keep up w ith the changing state of the flight vehicle. A Sugeno fuzzy inference system is extremely w ell suited to the task of smoothly interpolating the linear gains that w ould be applied across the input space; it is a natural and efficient gain scheduler. Similarly, a Sugeno system is suited for modeling nonlinear systems by interpolating betw een multiple linear models.

To see a specific example of a system w ith linear output membership functions, consider the one input one output system stored in `sugeno1.fis`.

```
fismat = readfis('sugeno1');
getfis(fismat,'output',1)
```

This syntax returns:

```
Name = output
          NumMFs = 2
          MFLabels =
                  line1
                  line2
          Range = [0 1]
```

The output variable has tw o membership functions.

```
getfis(fismat,'output',1,'mf',1)
```

This syntax returns:

```
Name = line1
          Type = linear
          Params =
                  -1    -1
```

```
getfis(fismat,'output',1,'mf',2)
```

This syntax returns:

```
        Name = line2
          Type = linear
          Params =
                                        1      -1
```

Further, these membership functions are linear functions of the input variable. The membership function `line1` is defined by the equation

$$output = (-1) \times input + (-1)$$

and the membership function `line2` is defined by the equation

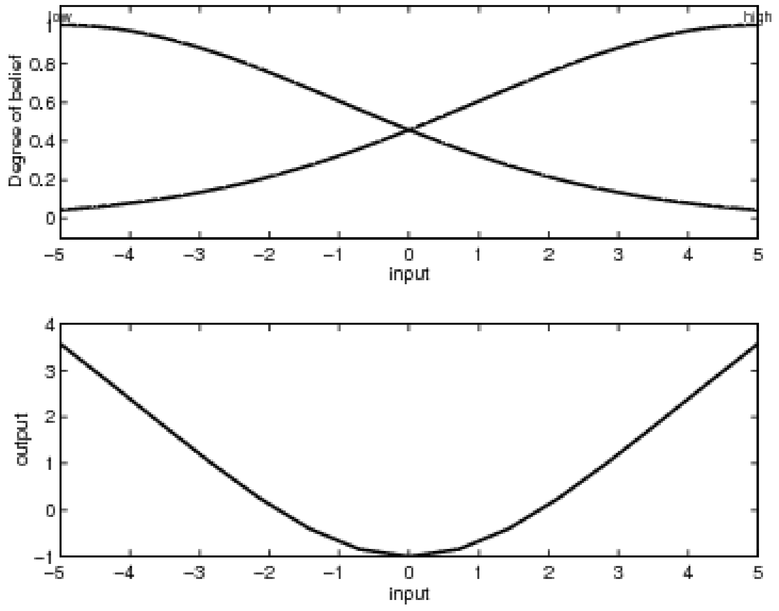
$$output = (1) \times input + (-1)$$

The input membership functions and rules define which of these output functions are expressed and when:

```
showrule(fismat)
ans =
1. If (input is low) then (output is line1) (1)
2. If (input is high) then (output is line2) (1)
```

The function `plotmf` shows us that the membership function `low` generally refers to input values less than zero, while `high` refers to values greater than zero. The function `gensurf` shows how the overall fuzzy system output switches smoothly from the line called `line1` to the line called `line2`.

```
subplot(2,1,1), plotmf(fismat,'input',1)
subplot(2,1,2),gensurf(fismat)
```



As this example shows, Sugeno-type system gives you the freedom to incorporate linear systems into your fuzzy systems. By extension, you could build a fuzzy system that switches between several optimal linear controllers as a highly nonlinear system moves around in its operating space.

More About