Fractional Fuzzy Inference Systems (FFISs)-MATLAB HELP FILE

(FFIS_201210)

This document has been devoted to explain briefly how the fractional fuzzy inference systems (FFISs) can be implemented (or simulated) in MATLAB software environment through the use of the first version of the function FFIS.m, i.e. (FFIS_201210).

Detail information about FFISs may be found in the following reference paper:

Mehran Mazandarani, Li Xiu, *Fractional Fuzzy Inference System: The New Generation of Fuzzy Inference Systems*, IEEE Access, Vol. 8, pp. 126066-126082, 2020.
DOI: 10.1109/ACCESS.2020.3008064

Link: https://ieeexplore.ieee.org/document/9136681

The functions and fuzzy system in the related files have been built based on the above reference.

The package includes the following files:

1. The help file (Help_FFIS201210.pdf)
2. FFIS.m
3. The example file (example.m)
4. Fuzzy inference system (fis.fis)

The main file that should be used is FFIS.m. This function evaluates fractional fuzzy inference system. Its role is similar to the evalfis.m in the MATLAB software.

Syntax

output=FFIS(fis,Inputs,Fids)

Description

This function converts the typical fuzzy inference system fis to its corresponding fractional fuzzy inference system and evaluates it for the input values in Inputs and returns the resulting output values in output. The fractional indices of the consequent part membership functions take place in the cell array Fids.

How to use

In order to use the fractional fuzzy inference system through the function of FFIS, two general and yet simple steps should be taken: first, building a typical fuzzy inference system and save it; second, determining the fractional indices corresponding the membership functions of consequent part.

Building a typical fuzzy inference system

There are two ways for building a typical fuzzy inference system through the use of Fuzzy Logic Toolbox: Building Fuzzy Systems Using Fuzzy Logic Designer or Building Fuzzy Systems at the Command Line.
Figure 1 shows building the fuzzy system by the use of Fuzzy Logic Designer.

The following simple steps should be taken:

1. Determine the inputs and outputs membership functions. The function FFIS has been prepared such that it only accepts **triangular and/or trapezoidal membership functions (trimf and or trapmf) for the output**. See Fig. 2. However, there is no restriction on the type of membership functions of the inputs. In addition, the function FFIS works only for fuzzy systems with single-output.

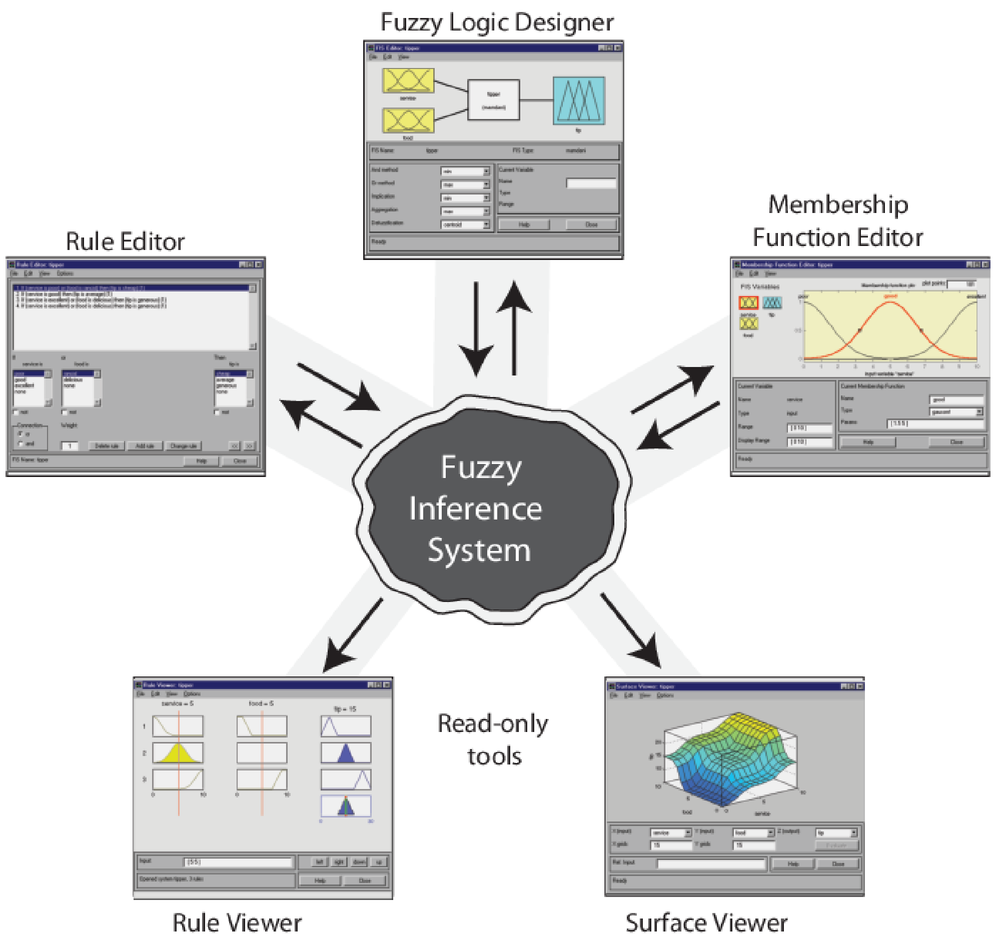
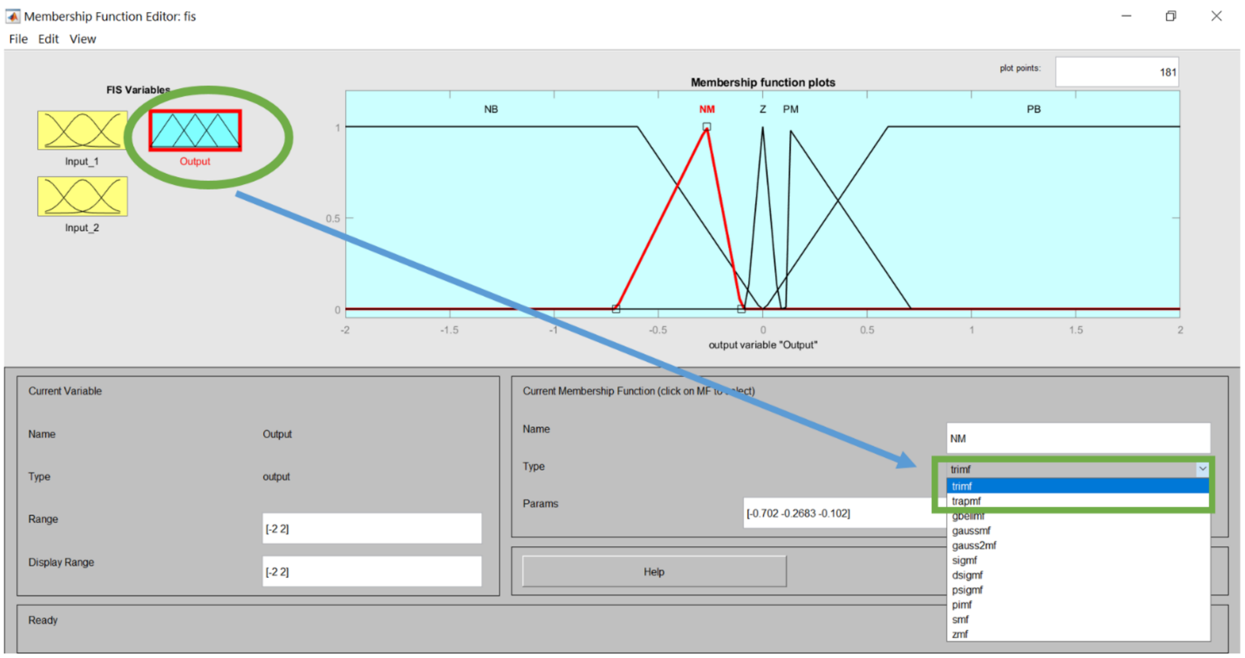


Figure 1. Fuzzy Logic Designer



plot points: 181

Current Variable

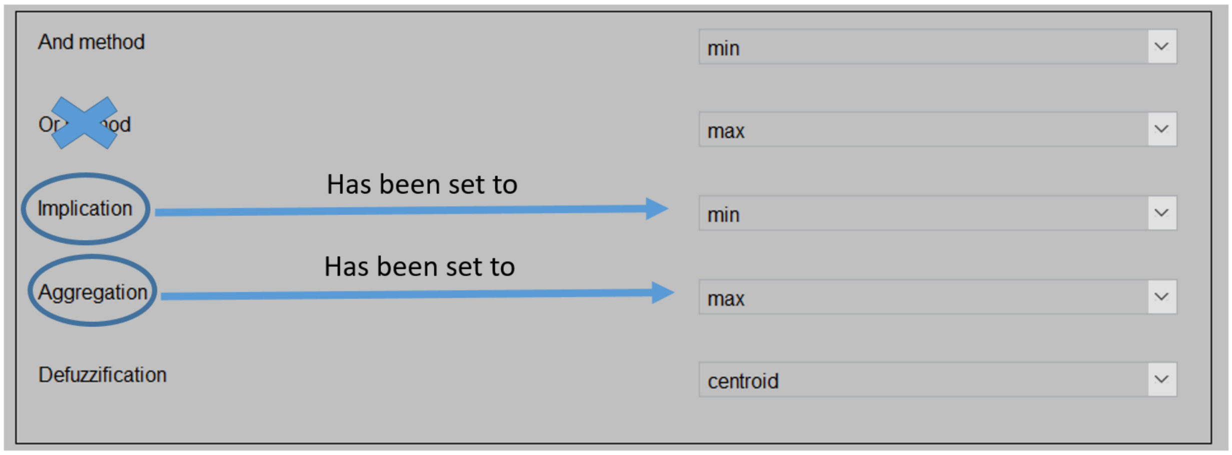
Name	Output
Type	output
Range	[-2 2]
Display Range	[-2 2]

Current Membership Function (click on MF to select)

Name	NM
Type	trimf
Params	[-0.702 -0.2683 -0.102]

Figure 2. The admissible type of membership functions of the output in the function of FFIS.

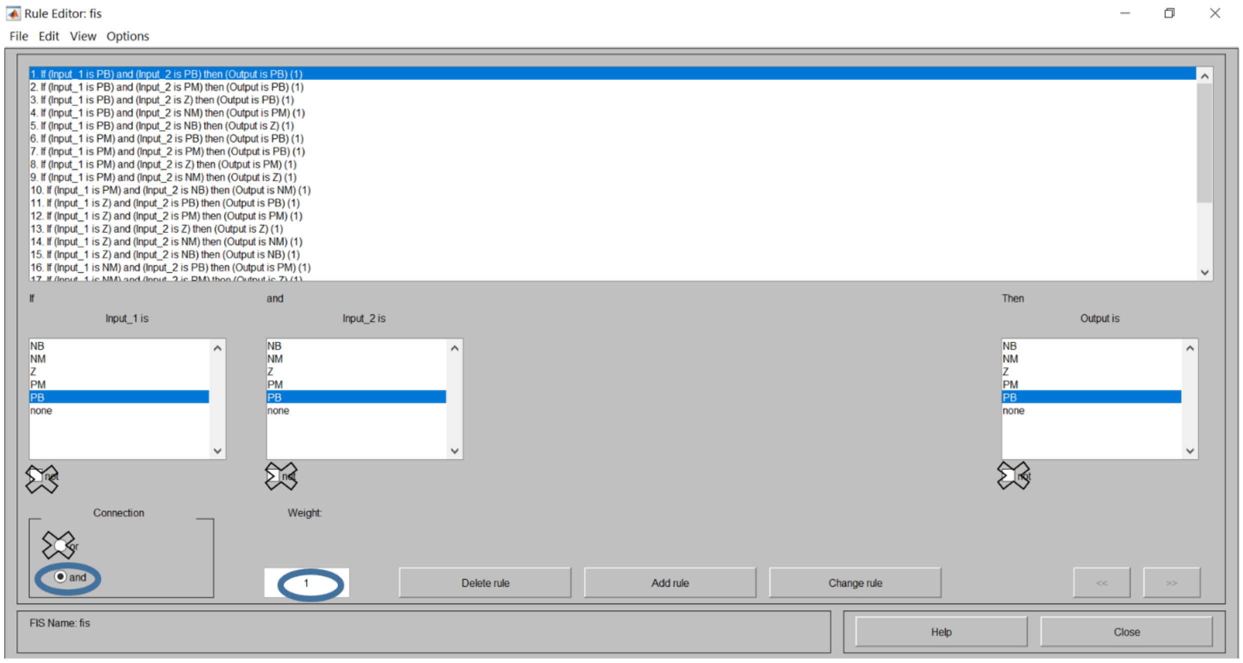
2. Determine the properties of the fuzzy system. In the function FFIS, the implication and aggregation have been set to “min” and “max”. But, the “And method” can be selected as “min” or “prod” operator. The “Defuzzification” operator is also optional. It should be noted that the FFIS function works for fuzzy system in which the variables in the antecedent part connect to each other only by the connector “And”. Thus, the antecedent part is not allowed to include “Or” connector. See Fig. 3.



And method	min
Or method	max
Implication	min
Aggregation	max
Defuzzification	centroid

Figure 3. The properties of fuzzy system.

3. Determine the fuzzy if-then rule base. The weight has been set to “1” and only the “and” connector is admissible in this version of FFIS function. The connectors “not” and “or” do not work in this version of FFIS function. See Fig. 4.



```
1. If (Input_1 is PB) and (Input_2 is PB) then (Output is PB) (1)
2. If (Input_1 is PB) and (Input_2 is PM) then (Output is PB) (1)
3. If (Input_1 is PB) and (Input_2 is Z) then (Output is PB) (1)
4. If (Input_1 is PB) and (Input_2 is NM) then (Output is PM) (1)
5. If (Input_1 is PB) and (Input_2 is NB) then (Output is Z) (1)
6. If (Input_1 is PM) and (Input_2 is PB) then (Output is PB) (1)
7. If (Input_1 is PM) and (Input_2 is PM) then (Output is PB) (1)
8. If (Input_1 is PM) and (Input_2 is Z) then (Output is PM) (1)
9. If (Input_1 is PM) and (Input_2 is NM) then (Output is Z) (1)
10. If (Input_1 is PM) and (Input_2 is NB) then (Output is NM) (1)
11. If (Input_1 is Z) and (Input_2 is PB) then (Output is PB) (1)
12. If (Input_1 is Z) and (Input_2 is PM) then (Output is PM) (1)
13. If (Input_1 is Z) and (Input_2 is Z) then (Output is Z) (1)
14. If (Input_1 is Z) and (Input_2 is NM) then (Output is NM) (1)
15. If (Input_1 is Z) and (Input_2 is NB) then (Output is NB) (1)
16. If (Input_1 is NM) and (Input_2 is PB) then (Output is PM) (1)
17. If (Input_1 is NM) and (Input_2 is PM) then (Output is Z) (1)
```

Input_1 is: PB; Input_2 is: PB; Output is: PB

Connection: ( ) or [✓ and]

Weight: 1

FIS Name: fis

Figure 4. Determine the fuzzy if-then rule base.

4. Export the fuzzy system to the folder in which the function FFIS and your main program file exist. It can be done by: File -> Export -> To File and then **save it by the name of fis**. See figures 5 and 6.

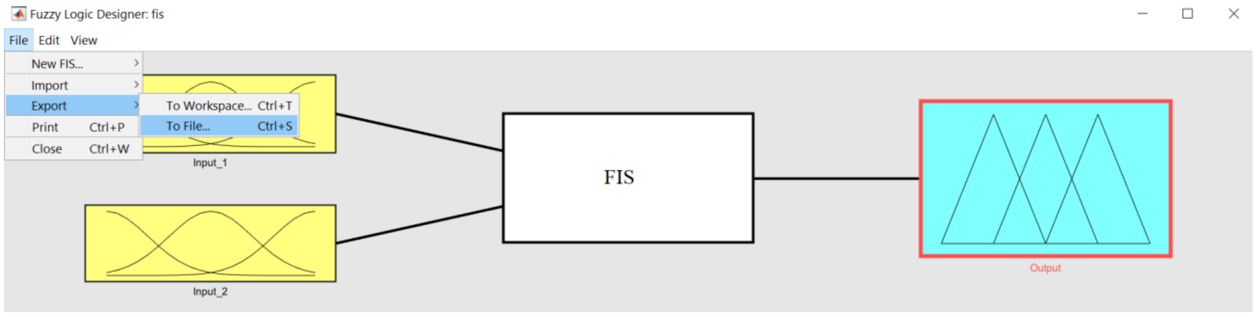


Figure 5. Export the fuzzy inference system to the main folder.

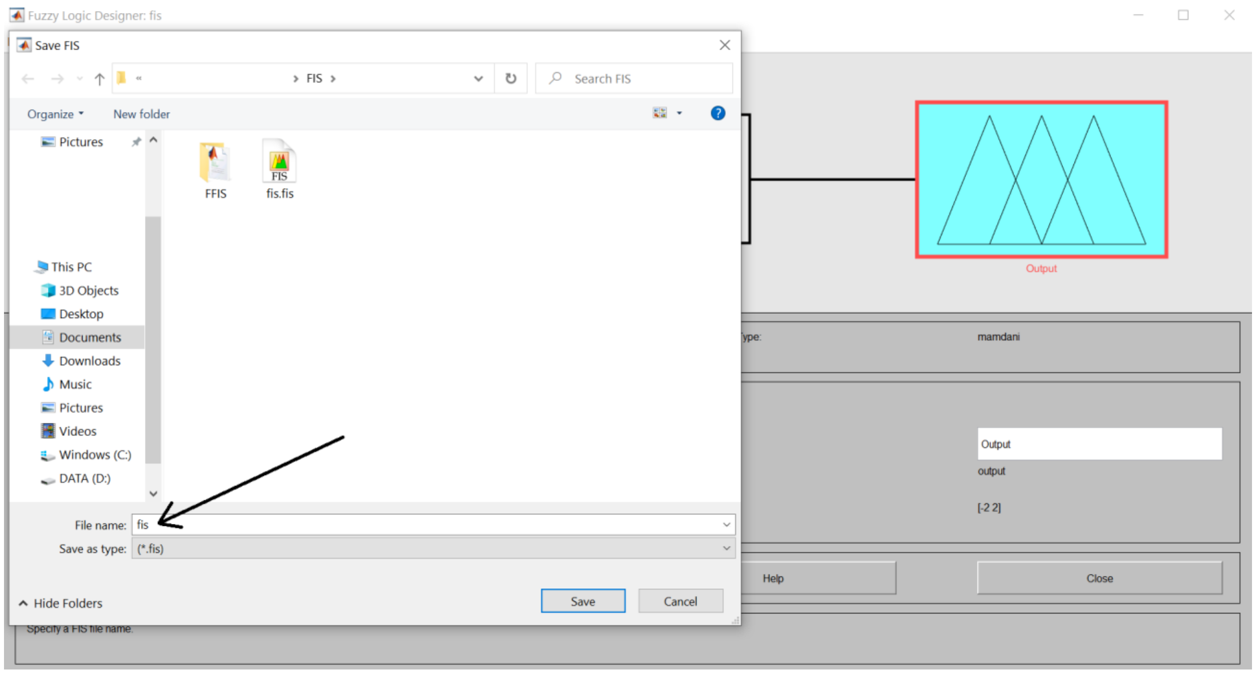


Figure 6. Save the fuzzy inference system with the name of fis.

In the main file (your desired script m-file) the output of this step is called by the command line

fis=readfis(fis);

The fis is used as the first argument in the function FFIS.m, see the syntax. The second argument is the inputs that is determined on the basis of the object of program. The third argument is the array of fractional indices whose structure is explained in the next.

Determining the fractional indices

The fractional indices structure is expressed as a cell array by

Fids={value_1,form_1, value_2,form_2,...,value_m,form_m}

In which value_i is a real number in the interval [0, 1] or a function of x.
value_i determins the fractional index of the ith fractional membership function in the consequent part membership functions plane.
There are two forms of fractional membership functions corresponding to two forms of fractional indices which are called α-form and β-form. For each fractional membership function must be assigned a value and a form.
form_i determines the form of the ith fractional membership function in the consequent part membership functions plane. For the α-form, form_i is set to 'a' or 0. For the β-form, form_i is set to 'b' or 1.

Example: Consider a fuzzy inference system with 5 membership functions in the consequent part membership functions plane which have been labeled as NB, NS, Z, PS, and PB. See Fig. 7.

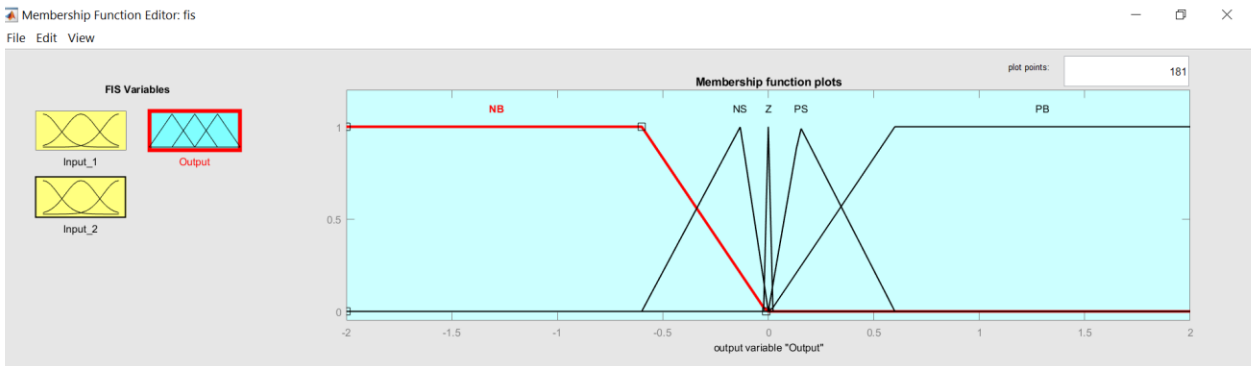


Figure 7. The consequent part membership functions plane.

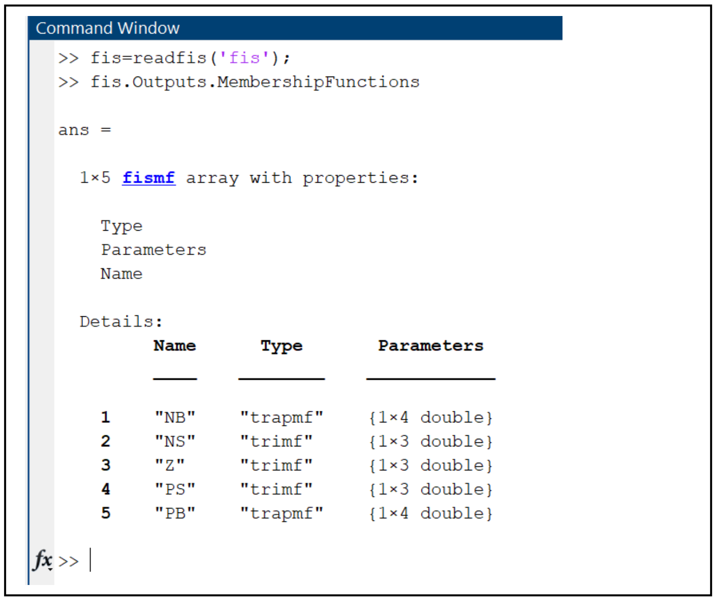
For the assignment of the fractional index value and the form to the ith fractional membership function of the output, we need to know the order or the number of membership functions. Thus, in the command window, first by

fis=readfis('fis');

the fis which had been saved in the previous step is loaded in the workspace. Then, by the command

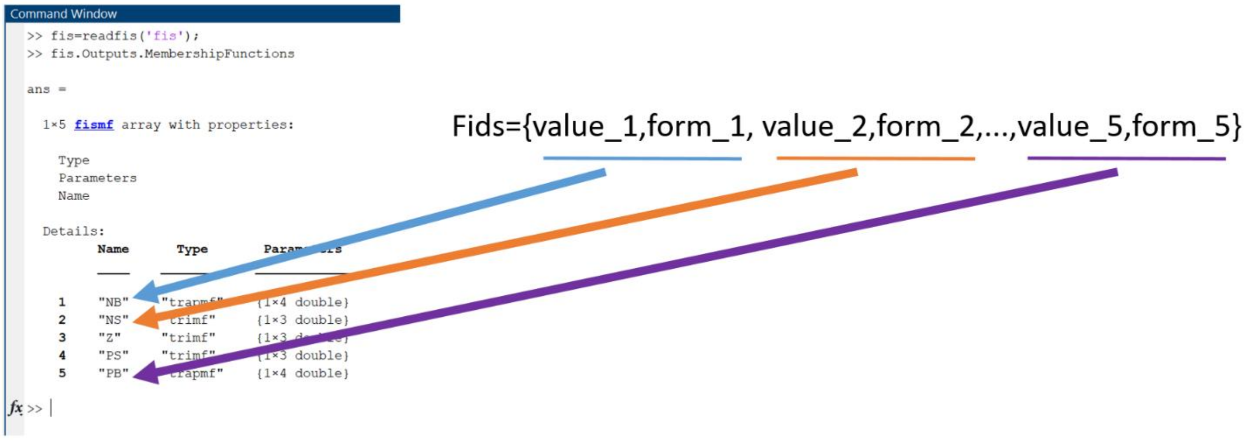
fis.Outputs.MembershipFunctions

The ith fractional membership function of the output is shown in the window. For the fis that has been considered in this example and shown in Fig. 7, we have



```
Command Window
>> fis=readfis('fis');
>> fis.Outputs.MembershipFunctions

ans =

  1×5 fismf array with properties:

    Type
    Parameters
    Name

  Details:
           Name      Type        Parameters
           ____    _________    ____________

    1      "NB"    "trapmf"     {1×4 double}
    2      "NS"    "trimf"      {1×3 double}
    3      "Z"     "trimf"      {1×3 double}
    4      "PS"    "trimf"      {1×3 double}
    5      "PB"    "trapmf"     {1×4 double}

fx >> |
```

Thus, in the cell array Fids, from the left to right, the value_i and form_i is assigned to the membership functions shown in the i-th row. See Fig. 8.



```
Command Window
>> fis=readfis('fis');
>> fis.Outputs.MembershipFunctions

ans =

  1×5 fismf array with properties:

    Type
    Parameters
    Name

  Details:
         Name      Type       Parameters
         ____    ________    ____________

    1    "NB"    "trapmf"    {1×4 double}
    2    "NS"    "trimf"     {1×3 double}
    3    "Z"     "trimf"     {1×3 double}
    4    "PS"    "trimf"     {1×3 double}
    5    "PB"    "trapmf"    {1×4 double}

fx >> |
```

Fids={value_1,form_1, value_2,form_2,...,value_5,form_5}

Figure 8. The order of assignment of fractional indices to the membership function.

Suppose that Fids={0.3, 'a', 0.5, 'b', 1,'a', 0.9, 'b', 0,'b'}. Then, based on the structure explained above, the fractional membership function labeled as NB is assigned a fractional index with the value of 0.3, and NB is in the α-form. The fractional membership function labeled as PS is assigned a fractional index with the value of 0.9, and NB is in the β-form. See the following

$$\text{Fids}=\{\underbrace{0.3,\ \text{'a'}}_{\overrightarrow{NB}},\ \underbrace{0.5,\ \text{'b'}}_{\overleftarrow{NS}},\ \underbrace{1,\text{'a'}}_{Z},\ \underbrace{0.9,\ \text{'b'}}_{\overleftarrow{PS}},\ \underbrace{0,\text{'b'}}_{\overleftarrow{PB}}\}$$

For the arrows in the top of labels indicate that the membership function is a fractional membership function. Moreover, the direction of the arrows indicate the form of fractional membership function. The left arrow corresponds to the α-form and the right arrow corresponds to the β-form. The set of α-form and β-form determines the mode of fractional fuzzy inference systems among which the aggressive mode is particular mode. For more details refer to the reference mentioned in the first page of this document.

For the assignment of functional fractional index to a membership function, a function of x needs to be written in the place of value_i. The function needs to range over [0, 1] or a subset of [0, 1]. The following cell array presents an example in which the fractional membership functions labeled as NS and PS have been assigned a functional fractional index as x^2.

$$\text{Fids}=\{\underbrace{0.1,\ \text{'a'}}_{\overrightarrow{NB}},\ \underbrace{\text{'x^2'},\ \text{‘a'}}_{\overrightarrow{NS}},\ \underbrace{1,\text{'a'}}_{Z},\ \underbrace{\text{'x^2'},\ \text{'b'}}_{\overleftarrow{PS}},\ \underbrace{0.1,\text{'b'}}_{\overleftarrow{PB}}\}$$

It should be noted that considering a functional fractional index as x^2 means that the fractional index is a quadratic function of the truth degree coming from the antecedent part. Indeed, the variable x plays the role of μ which is the truth degree coming from the antecedent part and sometimes is called the fire strength of the "if" part. Some more functional fractional indices are 'x'; '0.1*x+0.9*(x^2)'; '1-x'; '0.5*(x^2)'; '0.9*x', 'sqrt(x)'.

It should be noted that by setting all the values of fractional indices to 1, the fractional fuzzy inference system reduces to the well-known Mamdani's FIS. In other words, by

Fids={1, 'a', 1, 'b', 1,'a', 1, 'b', 1,'b'}
The output of fractional fuzzy inference system is the same as that of Mamdani's FIS, and in fact, "FFIS.m" function becomes equivalent to "evalfis.m".
For more details about functional fractional indices and fractional fuzzy inference systems refer to the reference in the first page of this document.

By determining the fractional indices, the fractional fuzzy inference system is ready to work through the use of FFIS.m function.

Example. Control of Inverted Pendulum System.

In this example the model-based control of an inverted pendulum system is presented. The model of the inverted pendulum system, adopted from the reference paper mentioned in the first page, is considered as the following nonlinear differential equations:

$$\ddot{\theta}(t) = \frac{g\sin(\theta(t)) - 0.5aml\dot{\theta}^2(t)\sin(2\theta(t)) - a\cos(\theta(t))\bar{u}(t)}{\frac{4l}{3} - aml\cos^2(\theta(t))}$$

For more details about the above model refer to the reference. The control structure has been considered as

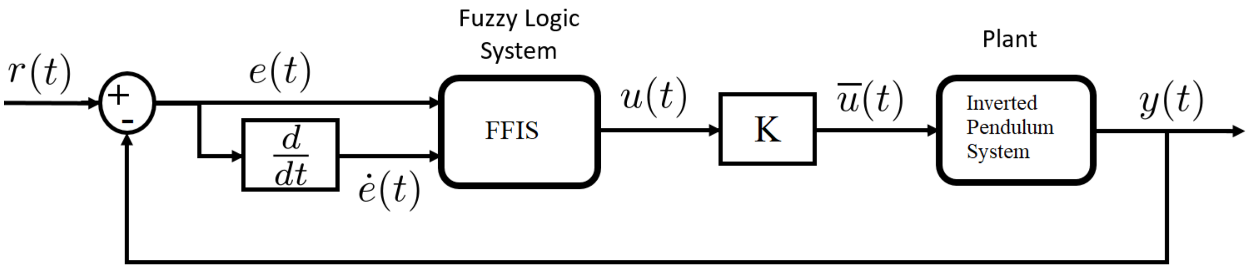
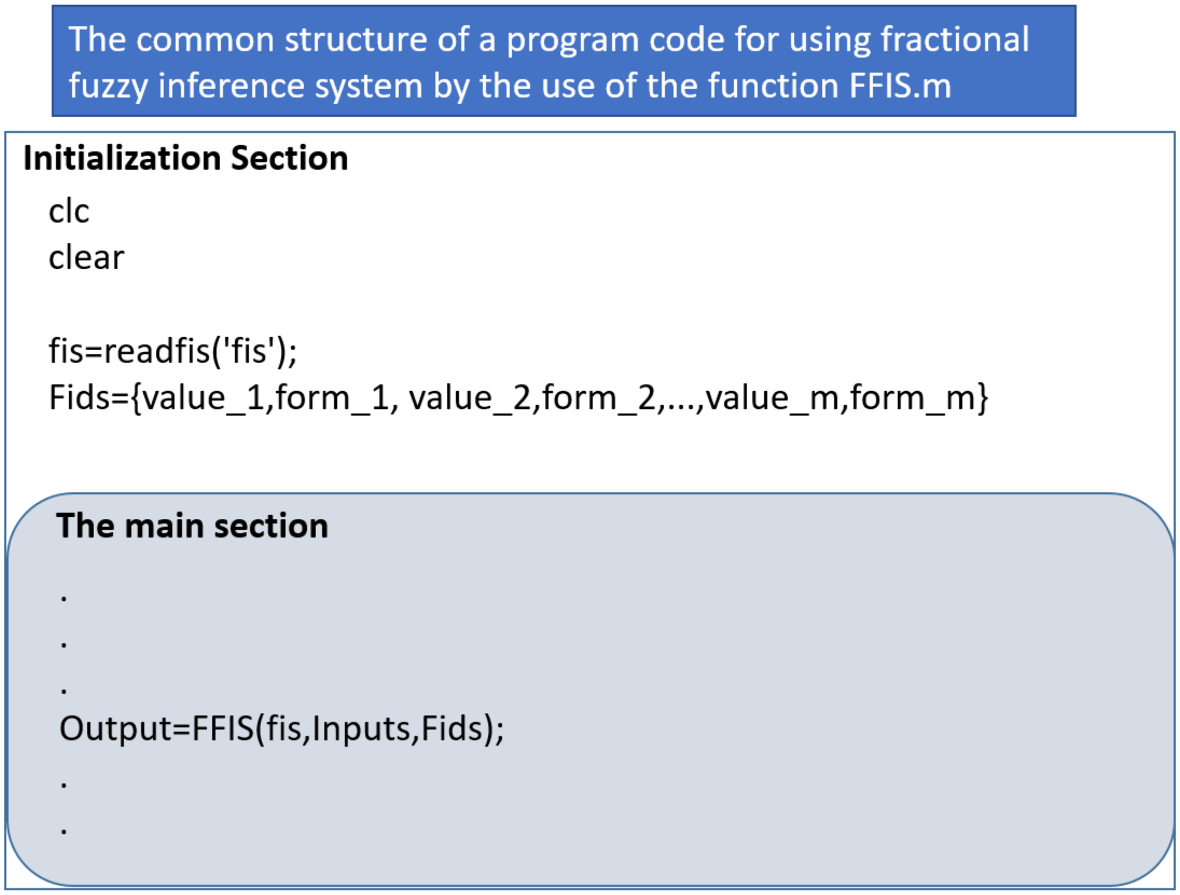


Figure 9. The closed loop control structure.

The membership functions of the inputs and the output; and the fuzzy if-then rules have been built through the use of fuzzy logic designer and the fuzzy inference system has been saved by the name of "fis.fis". This file exists in the folder. The inputs $e, \dot{e}$ (the error and the derivative of the error) have been considered as the θ, $\dot{\theta}$ respectively. In the main program (example.m), θ, $\dot{\theta}$ (or equivalently $e, \dot{e}$) have been considered as x1(cnt) and x2(cnt), respectively. “cnt” has the role of discretized time, or the counter. For more details about the membership functions and fuzzy systems properties, simply open the fis.fis using the following commands:

```
fis=readfis('fis');
fuzzy(fis)
```

Fig. 10 illustrates the common MATLAB programming code structure for applying fractional fuzzy inference system by the use of first version of the function FFIS.m.



The common structure of a program code for using fractional fuzzy inference system by the use of the function FFIS.m

Initialization Section

```
clc
clear

fis=readfis('fis');
Fids={value_1,form_1, value_2,form_2,...,value_m,form_m}
```

The main section

```
.
.
.
Output=FFIS(fis,Inputs,Fids);
.
.
```

Figure 10. The common structure of a MATLAB program code for applying FFIS.m.

If there is any question contact the authors of the reference paper.