1 2 9 0

UNIVERSIDADE D COIMBRA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

PhD Program in Informatics – DEI FCTUC

Research Seminar on Adaptive Computing

Professor: António Dourado Pereira Correia

Student: Afonso Bezerra Lima Neto - 2019130856

Report Assignment I

1. To find a discrete transfer function:

Considering the given function G(s): $\frac{3(s+1)}{s^3+3s^2+4s+2}$, the followings code must be executed in Matlab:

```
num=[3 3]
den=[1 3 4 2]
[zz, pp, kk] = tf2zp(num, den) % to find zeros, poles and gains
maxPP = max(abs(real(pp)))
aux = (1/maxPP)*0.25
[numd, dend] = c2dm(num,den,aux,'zoh') % apply the discretization method
'zoh'
```

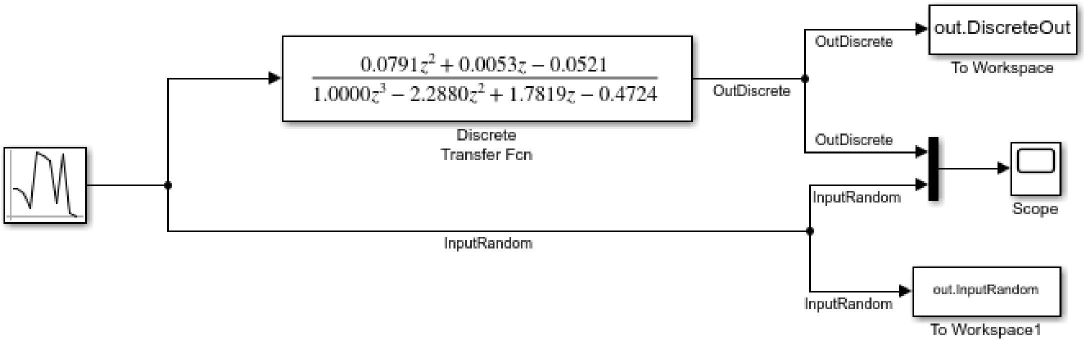
The results are:

```
pp =
  -1.0000 + 1.0000i
  -1.0000 - 1.0000i
  -1.0000 + 0.0000i

maxPP = 1.0000
aux = 0.2500
numd = [0 0.0791 0.0053 -0.0521]          % The discrete numerator coeficients
dend = [1.0000 -2.2880 1.7819 -0.4724]    % The discrete denominator coeficients
```

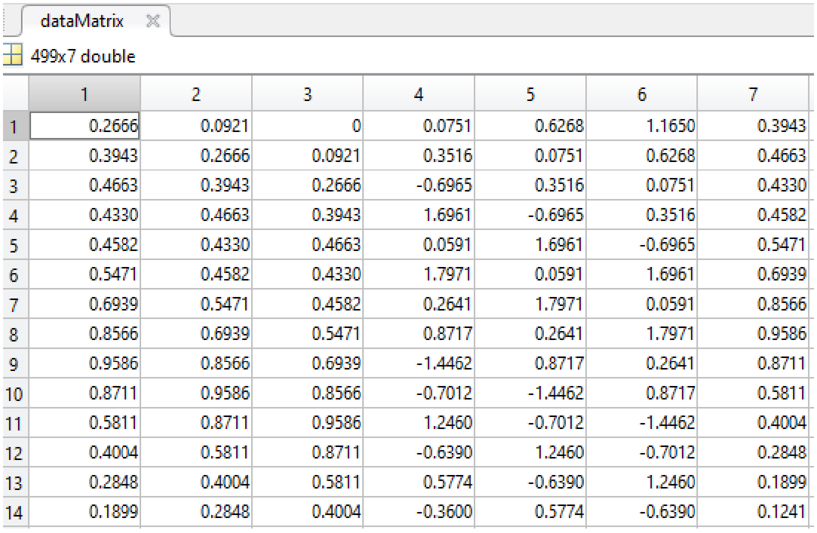
Thus, the following transfer function is obtained:

$$\frac{0.0791z^2+0.0053z-0.0521}{1.0000z^3-2.2880z^2+1.7819z-0.4724}$$

2. Using the above transfer function in Simulink, it’s possible to generate the dataset:



3. To read inputRandom and OutDiscrete data in Matlab and generate the data matrix according to the rules specified in the Assignment Document. The first 14 rows are presented below:


dataMatrix

499x7 double

	1	2	3	4	5	6	7
1	0.2666	0.0921	0	0.0751	0.6268	1.1650	0.3943
2	0.3943	0.2666	0.0921	0.3516	0.0751	0.6268	0.4663
3	0.4663	0.3943	0.2666	-0.6965	0.3516	0.0751	0.4330
4	0.4330	0.4663	0.3943	1.6961	-0.6965	0.3516	0.4582
5	0.4582	0.4330	0.4663	0.0591	1.6961	-0.6965	0.5471
6	0.5471	0.4582	0.4330	1.7971	0.0591	1.6961	0.6939
7	0.6939	0.5471	0.4582	0.2641	1.7971	0.0591	0.8566
8	0.8566	0.6939	0.5471	0.8717	0.2641	1.7971	0.9586
9	0.9586	0.8566	0.6939	-1.4462	0.8717	0.2641	0.8711
10	0.8711	0.9586	0.8566	-0.7012	-1.4462	0.8717	0.5811
11	0.5811	0.8711	0.9586	1.2460	-0.7012	-1.4462	0.4004
12	0.4004	0.5811	0.8711	-0.6390	1.2460	-0.7012	0.2848
13	0.2848	0.4004	0.5811	0.5774	-0.6390	1.2460	0.1899
14	0.1899	0.2848	0.4004	-0.3600	0.5774	-0.6390	0.1241

4. Now it’s time to generate the Fuzzy Inference System based on clusters of data. I used two methods: Subtractive and Fuzzy c-means, in order to find the center of the clusters and generate the respective “.fis” file:

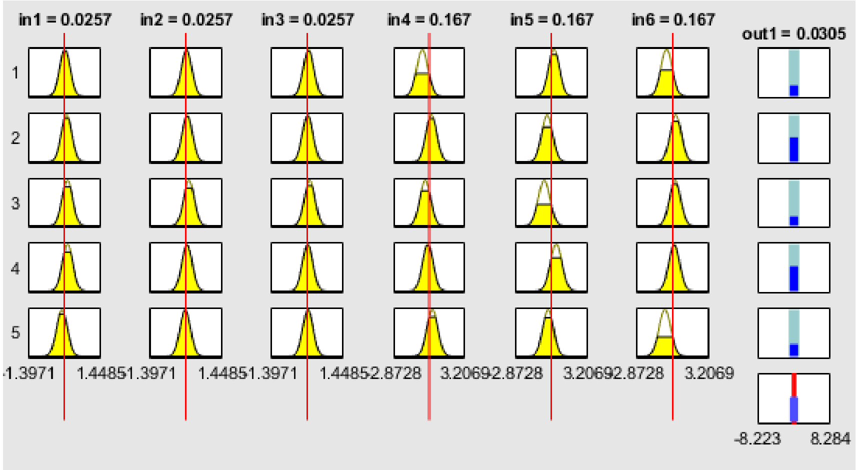
   a. The commands for Fuzzy c-means are:

```
inputData = dataMatrix(:,1:6);
outputData = dataMatrix(:,7);

optFCM = genfisOptions('FCMClustering','FISType','sugeno');
optFCM.NumClusters = 5;
optFCM.Verbose = 0;

myFisFCM = genfis(inputData,outputData,optFCM);
showrule(myFisFCM)

% run fuzzyLogicDesigner to save in a .fis file
fuzzyLogicDesigner(myFisFCM)
```

The FISFCM.fis contains the respective Fuzzy Inference System:

```
[System]
Name='FISFCM'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in1cluster1':'gaussmf',[0.203928250346964 0.0426073314668295]
MF2='in1cluster2':'gaussmf',[0.204577801218229 0.107421564573074]
MF3='in1cluster3':'gaussmf',[0.204911145392131 0.138976288684256]
MF4='in1cluster4':'gaussmf',[0.20503666270888 0.148299460896762]
MF5='in1cluster5':'gaussmf',[0.211014818245536 -0.0694404763532825]

[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in2cluster1':'gaussmf',[0.203719105007199 0.0556917793274846]
MF2='in2cluster2':'gaussmf',[0.204116578615613 0.0816358022642462]
MF3='in2cluster3':'gaussmf',[0.205265082851183 0.150582191054356]
MF4='in2cluster4':'gaussmf',[0.203519858501387 0.0746649276339012]
MF5='in2cluster5':'gaussmf',[0.204904018183953 0.00511965220042693]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in3cluster1':'gaussmf',[0.203383894798378 0.0803675216170815]
MF2='in3cluster2':'gaussmf',[0.204425104210593 0.048088759387525]
MF3='in3cluster3':'gaussmf',[0.203823125833397 0.128400482795589]
MF4='in3cluster4':'gaussmf',[0.203747988017466 0.0475188990998717]
MF5='in3cluster5':'gaussmf',[0.203523430515036 0.0599158101696461]

[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in4cluster1':'gaussmf',[0.481904920470063 -0.412302123395455]
MF2='in4cluster2':'gaussmf',[0.459356352624037 0.351945685273077]
MF3='in4cluster3':'gaussmf',[0.456514784030802 -0.165888698283529]
```

```
MF4='in4cluster4':'gaussmf',[0.451011085001758 0.0483482275127979]
MF5='in4cluster5':'gaussmf',[0.464656752777324 0.43449969433713]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in5cluster1':'gaussmf',[0.456607833705825 0.383572410498272]
MF2='in5cluster2':'gaussmf',[0.458426558616821 -0.18664634087017]
MF3='in5cluster3':'gaussmf',[0.486842443569865 -0.435797790362361]
MF4='in5cluster4':'gaussmf',[0.484611119792078 0.568196269045848]
MF5='in5cluster5':'gaussmf',[0.451501313525534 -0.106998835157245]

[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in6cluster1':'gaussmf',[0.473432930616692 -0.344511233373878]
MF2='in6cluster2':'gaussmf',[0.46390963019522 0.420628254557833]
MF3='in6cluster3':'gaussmf',[0.457131978768966 0.35953560128397]
MF4='in6cluster4':'gaussmf',[0.453263221224897 0.291253184978947]
MF5='in6cluster5':'gaussmf',[0.492680218359388 -0.483898961861499]

[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='out1cluster1':'linear',[2.28799999999988 -1.78189999999982
0.47239999999993 0.0791 0.00530000000000932 -0.0520999999999922 -
1.14761352364095e-16]
MF2='out1cluster2':'linear',[2.28799999999988 -1.78189999999982
0.47239999999993 0.0791 0.00530000000000932 -0.0520999999999922 -
1.14761352364095e-16]
MF3='out1cluster3':'linear',[2.28799999999988 -1.78189999999982
0.47239999999993 0.0791 0.00530000000000932 -0.0520999999999922 -
1.14761352364095e-16]
MF4='out1cluster4':'linear',[2.28799999999988 -1.78189999999982
0.47239999999993 0.0791 0.00530000000000932 -0.0520999999999922 -
1.14761352364095e-16]
MF5='out1cluster5':'linear',[2.28799999999988 -1.78189999999982
0.47239999999993 0.0791 0.00530000000000932 -0.0520999999999922 -
1.14761352364095e-16]

[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1
4 4 4 4 4 4, 4 (1) : 1
5 5 5 5 5 5, 5 (1) : 1
```



'1. If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) and (in6 is in6cluster1) then (out1 is out1cluster1) (1)'

'2. If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster2) then (out1 is out1cluster2) (1)'

'3. If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster3) then (out1 is out1cluster3) (1)'

'4. If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster4) and (in5 is in5cluster4) and (in6 is in6cluster4) then (out1 is out1cluster4) (1)'

'5. If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in5 is in5cluster5) and (in6 is in6cluster5) then (out1 is out1cluster5) (1)'

b. The commands for Subtractive are:

```
inputData = dataMatrix(:,1:6);
outputData = dataMatrix(:,7);

optSub = genfisOptions('SubtractiveClustering');
optSub.Verbose = 0;

myFisSUB = genfis(inputData,outputData,optSub);
showrule(myFisSUB)

% run fuzzyLogicDesigner to save the .fis file
fuzzyLogicDesigner(myFisSUB)
```

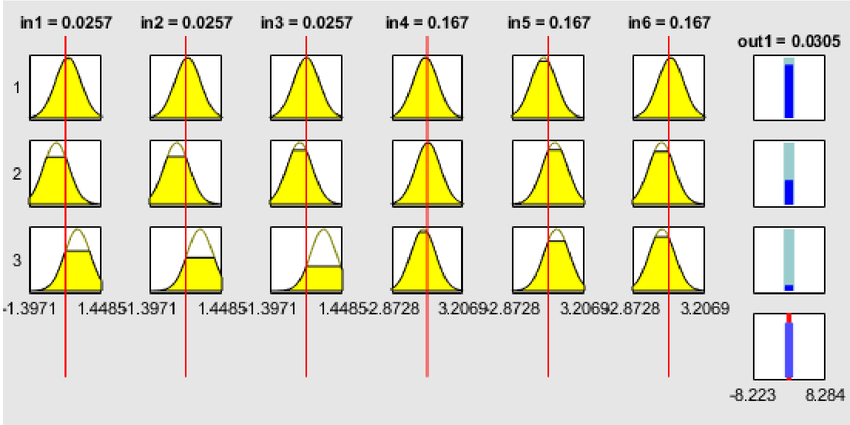
The FISSUB.fis contains the respective Fuzzy Inference System:

```
[System]
Name='FISSUB'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'
```

```
[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in1cluster1':'gaussmf',[0.503032173017372 0.131160501197739]
MF2='in1cluster2':'gaussmf',[0.503032173017372 -0.338425657499384]
MF3='in1cluster3':'gaussmf',[0.503032173017372 0.495950695023009]

[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in2cluster1':'gaussmf',[0.503032173017372 0.0984487706074979]
MF2='in2cluster2':'gaussmf',[0.503032173017372 -0.334730292805416]
MF3='in2cluster3':'gaussmf',[0.503032173017372 0.584690513313855]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in3cluster1':'gaussmf',[0.503032173017372 0.0369478595404813]
MF2='in3cluster2':'gaussmf',[0.503032173017372 -0.230561547893462]
MF3='in3cluster3':'gaussmf',[0.503032173017372 0.706076348201561]

[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in4cluster1':'gaussmf',[1.07474199355207 0.0166494938853639]
MF2='in4cluster2':'gaussmf',[1.07474199355207 0.227856907314768]
MF3='in4cluster3':'gaussmf',[1.07474199355207 -0.177618078276664]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in5cluster1':'gaussmf',[1.07474199355207 -0.22136078228451]
MF2='in5cluster2':'gaussmf',[1.07474199355207 0.706251990240337]
MF3='in5cluster3':'gaussmf',[1.07474199355207 0.864279576409738]

[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in6cluster1':'gaussmf',[1.07474199355207 0.336296999435117]
MF2='in6cluster2':'gaussmf',[1.07474199355207 -0.433373058325242]
MF3='in6cluster3':'gaussmf',[1.07474199355207 -0.397913854767017]

[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
```

MF1='out1cluster1':'linear',[2.28799999999767 -1.78189999999648 0.472399999998585 0.0791 0.00530000000018494 -0.0520999999999844 4.8260557500795e-16]
MF2='out1cluster2':'linear',[2.2879999999996 -1.7818999999994 0.472399999999758 0.0791000000000001 0.00530000000003143 -0.0520999999999735 -4.5186833771821e-16]
MF3='out1cluster3':'linear',[2.28800000000521 -1.78190000000786 0.472400000003159 0.0791000000000002 0.00529999999958741 -0.0521000000003483 1.17248158251235e-15]

[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1



'1. If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) and (in6 is in6cluster1) then (out1 is out1cluster1) (1)'

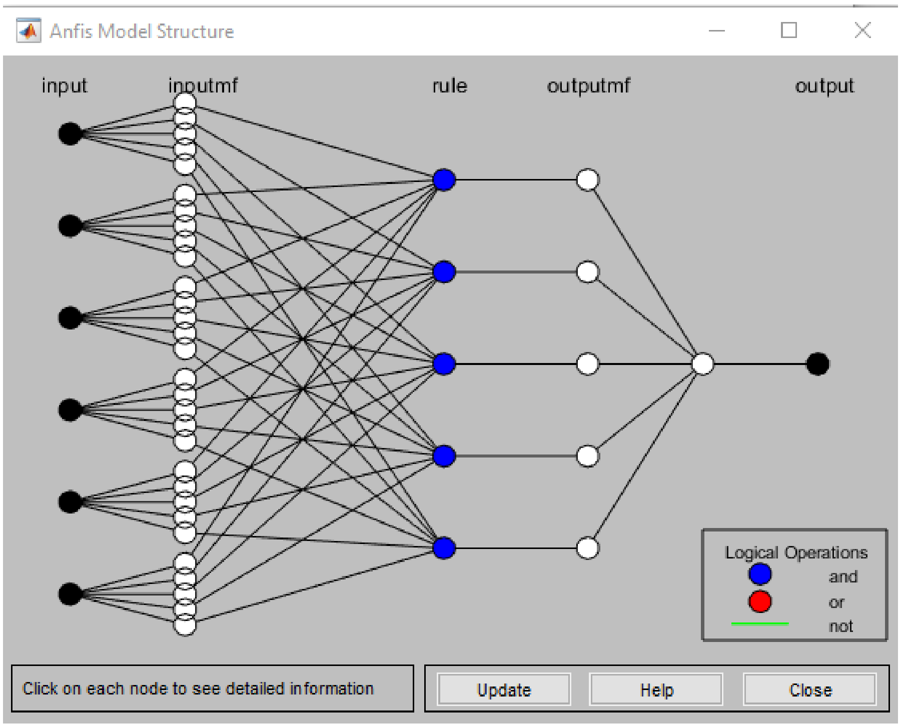
'2. If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) and (in6 is in6cluster2) then (out1 is out1cluster2) (1)'

'3. If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) and (in6 is in6cluster3) then (out1 is out1cluster3) (1)'

5. In order to optimize these Fuzzy Inference Systems, I used hybrid and backpropagation methods. However, split the data matrix in training (70% -> data from the beginning) and test (30% -> remaining data).
   a. Comands:

```
size=length(dataMatrix);
trainSize=round((size*0.7));
trainData=dataMatrixGenerated(1:trainSize,:);
testData=dataMatrixGenerated(trainSize+1:size,:);
```

6. The ANFIS model for FCM:



a. The FISFCM optimized by Hybrid method:

```
[System]
Name='FISFCMHyb'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'


[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in1cluster1':'gaussmf',[0.20319675173294 0.0412518558709949]
MF2='in1cluster2':'gaussmf',[0.211106194050416 0.108598936362734]
MF3='in1cluster3':'gaussmf',[0.197138953361984 0.139139065842394]
MF4='in1cluster4':'gaussmf',[0.20405667481472 0.148787241560201]
MF5='in1cluster5':'gaussmf',[0.211727167174901 -0.0699659850215811]

[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in2cluster1':'gaussmf',[0.204366163740471 0.0541907820898719]
```

```
MF2='in2cluster2':'gaussmf',[0.210983966433717 0.0823334519308659]
MF3='in2cluster3':'gaussmf',[0.200668501229767 0.150254163415181]
MF4='in2cluster4':'gaussmf',[0.20010784365137 0.0757757322208385]
MF5='in2cluster5':'gaussmf',[0.205094869148318 0.00503954671646644]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in3cluster1':'gaussmf',[0.206366366099761 0.0786193180349108]
MF2='in3cluster2':'gaussmf',[0.205599162802727 0.0476178910288905]
MF3='in3cluster3':'gaussmf',[0.205372377488191 0.128971448386236]
MF4='in3cluster4':'gaussmf',[0.198841021331171 0.0489111886474828]
MF5='in3cluster5':'gaussmf',[0.202926037626683 0.0602388872522134]

[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in4cluster1':'gaussmf',[0.481160533852856 -0.41236657923771]
MF2='in4cluster2':'gaussmf',[0.460786435418232 0.352987596776172]
MF3='in4cluster3':'gaussmf',[0.457687510566776 -0.167388140887955]
MF4='in4cluster4':'gaussmf',[0.448513564447033 0.0490049479987]
MF5='in4cluster5':'gaussmf',[0.462777315127246 0.434246013605882]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in5cluster1':'gaussmf',[0.456105301921469 0.383268308086589]
MF2='in5cluster2':'gaussmf',[0.460581487640219 -0.186553281983571]
MF3='in5cluster3':'gaussmf',[0.486296823589197 -0.43430249695606]
MF4='in5cluster4':'gaussmf',[0.485575362876188 0.567960138070189]
MF5='in5cluster5':'gaussmf',[0.45390358144621 -0.107762282416774]

[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in6cluster1':'gaussmf',[0.472574799786445 -0.344595434212221]
MF2='in6cluster2':'gaussmf',[0.464090187043888 0.42002325694286]
MF3='in6cluster3':'gaussmf',[0.457668322431054 0.36051406154974]
MF4='in6cluster4':'gaussmf',[0.451492212476371 0.290574577260341]
MF5='in6cluster5':'gaussmf',[0.494670274778399 -0.483768101090642]

[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='out1cluster1':'linear',[0.872888242114102 0.353588595294471 -0.385620005200573
0.0790964280037406 0.117233152674803 0.0425848533354474 1.37179750633667e-05]
MF2='out1cluster2':'linear',[0.862355774867929 0.369488226227544 -0.392002832014664
0.0790849201982997 0.118066617500216 0.0432974899468204 4.50904694503072e-05]
MF3='out1cluster3':'linear',[0.868755424372594 0.359803193957827 -0.388081431361103
0.079072862062165 0.117532218400336 0.0428689651690784 -5.69181083037712e-05]
MF4='out1cluster4':'linear',[0.863335918567111 0.368104728774326 -0.391525496959728
0.079067221388875 0.117993646288011 0.043199027136113 1.56473122400914e-05]
MF5='out1cluster5':'linear',[0.878044771457899 0.345839156247712 -0.382511317098839
0.0791104111111216 0.11682638982625 0.0422478464392243 4.47794931999734e-06]
```

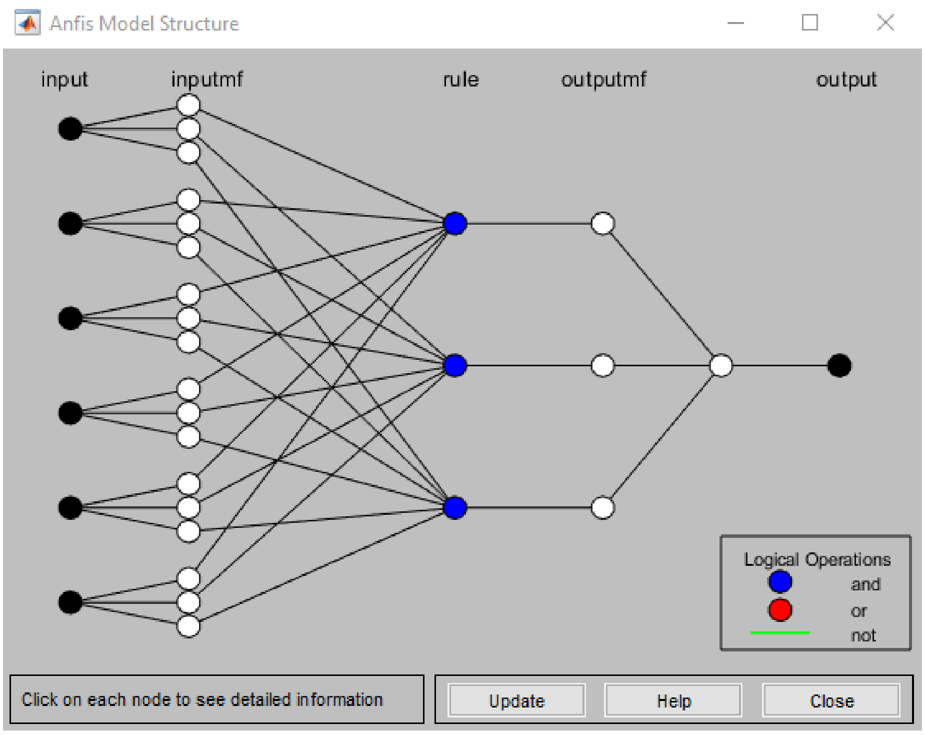
```
[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1
4 4 4 4 4 4, 4 (1) : 1
5 5 5 5 5 5, 5 (1) : 1
```

b. The FISFCM optimized by Backpropagation method:

```
[System]
Name='FISFCMBack'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=5
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in1cluster1':'gaussmf',[0.204435872557712 0.0424329175370261]
MF2='in1cluster2':'gaussmf',[0.205219178244866 0.107378994199849]
MF3='in1cluster3':'gaussmf',[0.204703952538069 0.138862293811268]
MF4='in1cluster4':'gaussmf',[0.205456553060156 0.148289244686225]
MF5='in1cluster5':'gaussmf',[0.21000331403767 -0.0694142875282039]

[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in2cluster1':'gaussmf',[0.204178908075546 0.0556142260446388]
MF2='in2cluster2':'gaussmf',[0.204842423566733 0.0816614964322076]
MF3='in2cluster3':'gaussmf',[0.205155245281537 0.150538381161005]
MF4='in2cluster4':'gaussmf',[0.204052192710663 0.0746611150835347]
MF5='in2cluster5':'gaussmf',[0.203358521820876 0.00510572595591061]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='in3cluster1':'gaussmf',[0.203841616623255 0.080375315545213]
MF2='in3cluster2':'gaussmf',[0.205259163385756 0.0481427285004169]
MF3='in3cluster3':'gaussmf',[0.203802844329066 0.128407174245282]
MF4='in3cluster4':'gaussmf',[0.204331153716206 0.0475169155719622]
MF5='in3cluster5':'gaussmf',[0.201625308101308 0.0598674304931894]

[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in4cluster1':'gaussmf',[0.482187839507588 -0.412211632824067]
```

```
MF2='in4cluster2':'gaussmf',[0.459858363673668 0.352061789253941]
MF3='in4cluster3':'gaussmf',[0.456395702968096 -0.165846610514082]
MF4='in4cluster4':'gaussmf',[0.451322774085692 0.0483872667018797]
MF5='in4cluster5':'gaussmf',[0.46402939302226 0.434353012845388]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in5cluster1':'gaussmf',[0.456821168898357 0.383565503896172]
MF2='in5cluster2':'gaussmf',[0.458778969625092 -0.186722749765705]
MF3='in5cluster3':'gaussmf',[0.48682269707974 -0.435800490306977]
MF4='in5cluster4':'gaussmf',[0.484690485450258 0.568210895105257]
MF5='in5cluster5':'gaussmf',[0.450605547119699 -0.107060466059292]
[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=5
MF1='in6cluster1':'gaussmf',[0.473901083218043 -0.344650779338459]
MF2='in6cluster2':'gaussmf',[0.464236868487773 0.42057514313386]
MF3='in6cluster3':'gaussmf',[0.457171543012923 0.359408379881816]
MF4='in6cluster4':'gaussmf',[0.453450595925912 0.291215842721634]
MF5='in6cluster5':'gaussmf',[0.492194784003123 -0.483786659832295]

[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=5
MF1='out1cluster1':'linear',[2.28809250847018 -1.78177176758463 0.472559533674615
0.0779060281036141 0.00606360950964869 -0.0530668847064562 0.00120694580996838]
MF2='out1cluster2':'linear',[2.28815754043078 -1.78180883582886 0.47240052009133
0.0797960941369835 0.00500850172438012 -0.0514522841007416 0.000822717815848342]
MF3='out1cluster3':'linear',[2.2888258715405 -1.78100241743163 0.473159026268339
0.078386819535019 0.00351208108474546 -0.0510091584572587 0.0019702523879757]
MF4='out1cluster4':'linear',[2.28825790599364 -1.78179587533055 0.472400375006354
0.0791675662548462 0.0062561133395093 -0.0517210281042744 0.000768918539120406]
MF5='out1cluster5':'linear',[2.28698772858877 -1.78262391859207 0.471880209979654
0.0816586823767592 0.00478185106451871 -0.0549994857508084 0.00283930919757474]

[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1
4 4 4 4 4 4, 4 (1) : 1
5 5 5 5 5 5, 5 (1) : 1
```

7. The ANFIS model for Subtractive:



a. The FISSUB optimized by Hybrid method:

```
[System]
Name='FISSUBHyb'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in1cluster1':'gaussmf',[0.533105528703421 0.188951656176213]
MF2='in1cluster2':'gaussmf',[0.47967651903559 -0.344769773630531]
MF3='in1cluster3':'gaussmf',[0.502442126801145 0.479251496761449]

[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in2cluster1':'gaussmf',[0.540853045610932 0.159936853090036]
MF2='in2cluster2':'gaussmf',[0.482919424215853 -0.343326666944355]
MF3='in2cluster3':'gaussmf',[0.501354896208444 0.573148886041063]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in3cluster1':'gaussmf',[0.543228574246685 0.113682623801685]
MF2='in3cluster2':'gaussmf',[0.478583153699333 -0.259470467383687]
MF3='in3cluster3':'gaussmf',[0.496816665448716 0.705941550688051]
```

```
[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in4cluster1':'gaussmf',[1.11544803731138 0.0461881426042432]
MF2='in4cluster2':'gaussmf',[1.08697122427327 0.207318269966213]
MF3='in4cluster3':'gaussmf',[1.01701401405906 -0.19277410693924]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in5cluster1':'gaussmf',[1.13297300266809 -0.194008052142635]
MF2='in5cluster2':'gaussmf',[1.06658151115823 0.668321550173955]
MF3='in5cluster3':'gaussmf',[1.05391197593049 0.884928310505447]

[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in6cluster1':'gaussmf',[1.10617122214737 0.341248576380815]
MF2='in6cluster2':'gaussmf',[1.09101428285013 -0.423486793214924]
MF3='in6cluster3':'gaussmf',[1.00839448722751 -0.420842456457768]

[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='out1cluster1':'linear',[0.897742149233273 0.316145864661859 -0.370573231848332
0.0790992824876198 0.115259933367491 0.0409255390072035 -6.3176995076392e-06]
MF2='out1cluster2':'linear',[0.885774422235934 0.334228683698691 -0.377839880052881
0.0791021560287103 0.116214929230702 0.041739347344067 5.21113486602618e-05]
MF3='out1cluster3':'linear',[0.870353092062564 0.357443632352897 -0.387158553353995
0.0790826381396501 0.117455075554637 0.0427463931624358 -3.5623671329401e-05]

[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1
```

b. The FISSUB optimized by Backpropagation method:

```
[System]
Name='FISSUBBack'
Type='sugeno'
Version=2.0
NumInputs=6
NumOutputs=1
NumRules=3
AndMethod='prod'
OrMethod='probor'
ImpMethod='prod'
AggMethod='sum'
DefuzzMethod='wtaver'

[Input1]
Name='in1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in1cluster1':'gaussmf',[0.502823985655774 0.131094990093116]
MF2='in1cluster2':'gaussmf',[0.503160495446989 -0.338311267530688]
MF3='in1cluster3':'gaussmf',[0.50307751216927 0.495945848181888]
```

```
[Input2]
Name='in2'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in2cluster1':'gaussmf',[0.502814884447492 0.0983683792066203]
MF2='in2cluster2':'gaussmf',[0.50313855065494 -0.334629995242577]
MF3='in2cluster3':'gaussmf',[0.503074414444226 0.584658312345976]

[Input3]
Name='in3'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='in3cluster1':'gaussmf',[0.502790539926078 0.0368674898309134]
MF2='in3cluster2':'gaussmf',[0.503118891309211 -0.230519381629158]
MF3='in3cluster3':'gaussmf',[0.503130134865945 0.705976633885798]

[Input4]
Name='in4'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in4cluster1':'gaussmf',[1.07458829486018 0.0166038875542835]
MF2='in4cluster2':'gaussmf',[1.07480735807963 0.227855184164863]
MF3='in4cluster3':'gaussmf',[1.07484126140023 -0.177572774978066]

[Input5]
Name='in5'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in5cluster1':'gaussmf',[1.07461120051086 -0.221254573474991]
MF2='in5cluster2':'gaussmf',[1.07496048406238 0.706132662070521]
MF3='in5cluster3':'gaussmf',[1.07500504493012 0.864133864189069]

[Input6]
Name='in6'
Range=[-2.87275126966852 3.20690754366444]
NumMFs=3
MF1='in6cluster1':'gaussmf',[1.07460713100409 0.336194609485278]
MF2='in6cluster2':'gaussmf',[1.07490599436012 -0.433269265623343]
MF3='in6cluster3':'gaussmf',[1.07493865030659 -0.397796356525113]


[Output1]
Name='out1'
Range=[-1.39709931547322 1.44848037009149]
NumMFs=3
MF1='out1cluster1':'linear',[2.28726108290726 -1.78257610044353 0.471949465714607
0.0783429736653273 0.00798255272259209 -0.0551483471251275 -0.00425454738544047]
MF2='out1cluster2':'linear',[2.28854890226791 -1.78126688525381 0.473020640419253
0.0782764127895448 0.00496864463094985 -0.0521593717461456 -0.00173452575353558]
MF3='out1cluster3':'linear',[2.28681337006798 -1.78309946062204 0.471274429691294
0.0791996994891441 0.00497429642033952 -0.0525967672733436 -0.00226728912854374]

[Rules]
1 1 1 1 1 1, 1 (1) : 1
2 2 2 2 2 2, 2 (1) : 1
3 3 3 3 3 3, 3 (1) : 1
```

8. Now it's time to compute the performance of the optimized Fuzzy Inference Systems.
    a. First of all, I got a range of data that wasn't used neither in training nor in test.
    b. Then I called evalfis method in Matlab for each FIS file. Below are the commands:

```
evalfismyFisFCM = evalfis(FISFCMBack, inputChk)
evalfismyFisFCMBP = evalfis(FISFCMBack, inputChk)
evalfismyFisFCMHibrid = evalfis(FISFCMHyb, inputChk)
evalfismyFisSUB = evalfis(FISSUBBack, inputChk)
evalfismyFisSUBBP = evalfis(FISSUBBack, inputChk)
evalfismyFisSUBHibrid = evalfis(FISSUBHyb, inputChk)
```

    c. Next, I created a table concatenating the arrays:

```
ds = table(outputChk, evalfismyFisFCM, evalfismyFisFCMBP,
evalfismyFisFCMHibrid, evalfismyFisSUB, evalfismyFisSUBBP,
evalfismyFisSUBHibrid);
ds.Properties.VariableNames = {'Observed', 'FCM', 'FCM_BP', 'FCM_Hibr',
'Sub', 'Sub_BP', 'Sub_Hibr'}
```

    d. Next, I calculated the errors:

```
ds.Err_FCM = (ds.Observed - ds.FCM)
ds.Err_FCM_BP = (ds.Observed - ds.FCM_BP)
ds.Err_FCM_Hibr = (ds.Observed - ds.FCM_Hibr)
ds.Err_Sub = (ds.Observed - ds.Sub)
ds.Err_Sub_BP = (ds.Observed - ds.Sub_BP)
ds.Err_Sub_Hibr = (ds.Observed - ds.Sub_Hibr)
```

    e. Next, I calculated the Squared Errors:

```
ds.SqrErr_FCM = ds.Err_FCM.^2
ds.SqrErr_FCM_BP = ds.Err_FCM_BP.^2
ds.SqrErr_FCM_Hibr = ds.Err_FCM_Hibr.^2
ds.SqrErr_Sub = ds.Err_Sub.^2
ds.SqrErr_Sub_BP = ds.Err_Sub_BP.^2
ds.SqrErr_Sub_Hibr = ds.Err_Sub_Hibr.^2
```

    f. Finally, the Mean Squared Errors where calculated:

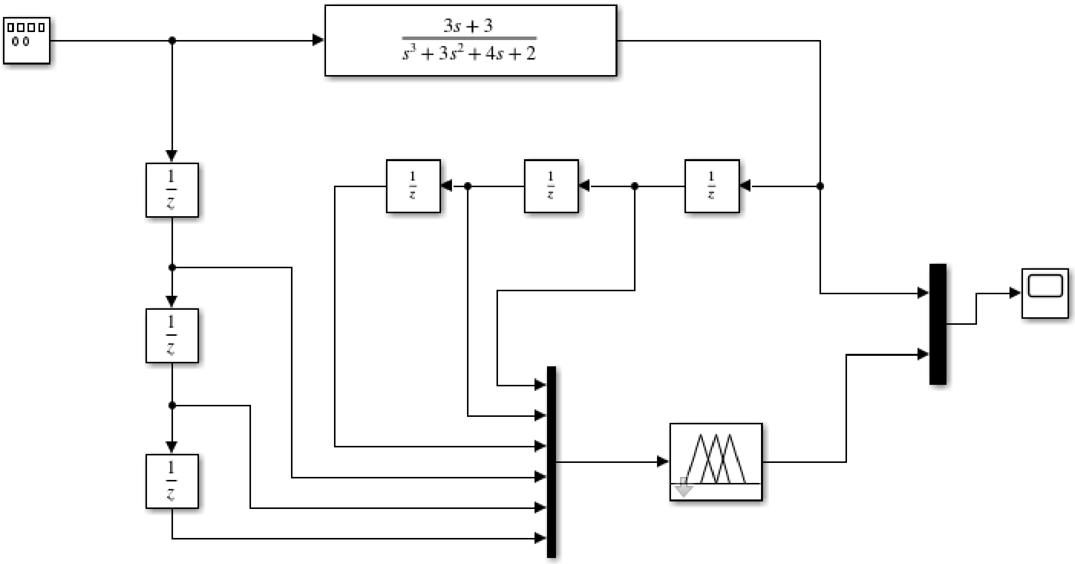
```
mseFCM = num2str(sum(ds.SqrErr_FCM)/height(ds),'%.10f')
mseFCM_BP = num2str(sum(ds.SqrErr_FCM_BP)/height(ds),'%.10f')
mseFCM_Hibr = num2str(sum(ds.SqrErr_FCM_Hibr)/height(ds),'%.10f')
mseSub = num2str(sum(ds.SqrErr_Sub)/height(ds),'%.10f')
mseSub_BP = num2str(sum(ds.SqrErr_Sub_BP)/height(ds),'%.10f')
mseSub_Hibr = num2str(sum(ds.SqrErr_Sub_Hibr)/height(ds),'%.10f')
```

    The table below contains the results:

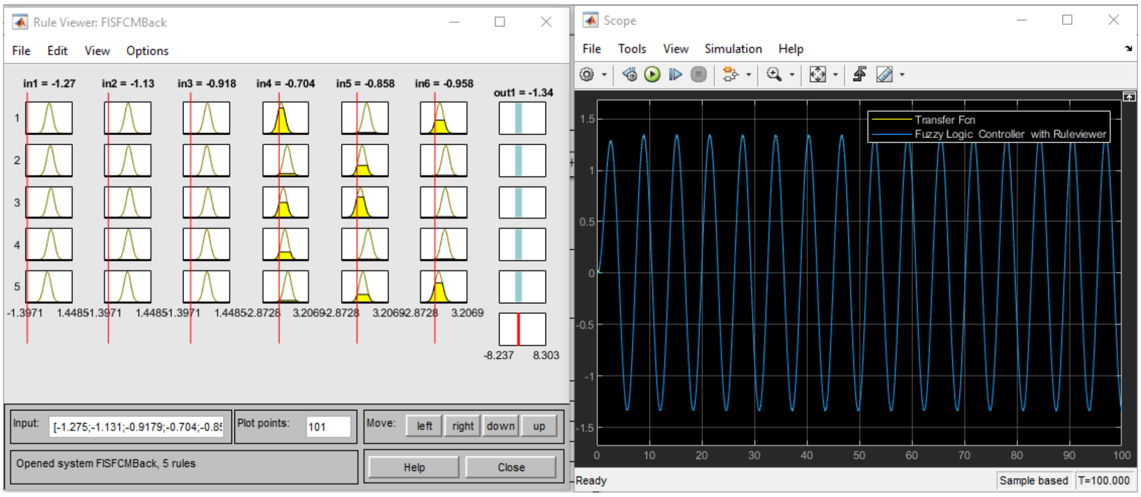
mseFCM	mseFCM_BP	mseFCM_Hibr	mseSub	mseSub_BP	mseSub_Hibr
0.0000270530	0.0000270530	0.0000000025	0.0000262986	0.0000262986	0.0000000023

    Thus, considering the least MSE is the best, the Subtractive Clustering with Hybrid optimization was the best, followed by Fuzzy c-means Clustering with Hybrid optimization. Backpropagation wasn't so good.

9. Below are the tests in Simulink for each FIS:



a. FCM with Backpropagation



Input: [-1.275;-1.131;-0.9179;-0.704;-0.8

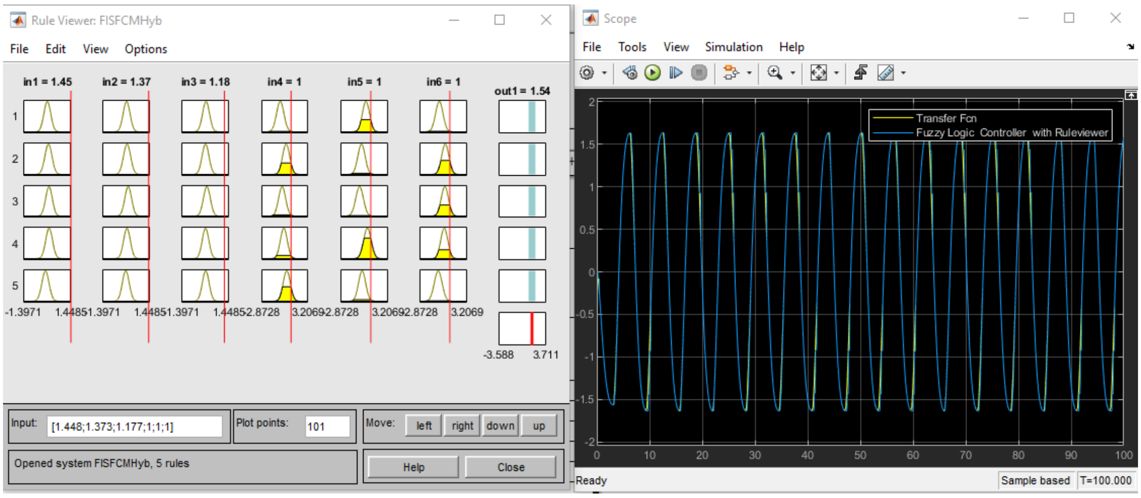
Plot points: 101

Move: left right down up

Opened system FISFCMBack, 5 rules

Help Close

b. FCM with Hybrid



Input: [1.448;1.373;1.177;1;1;1]

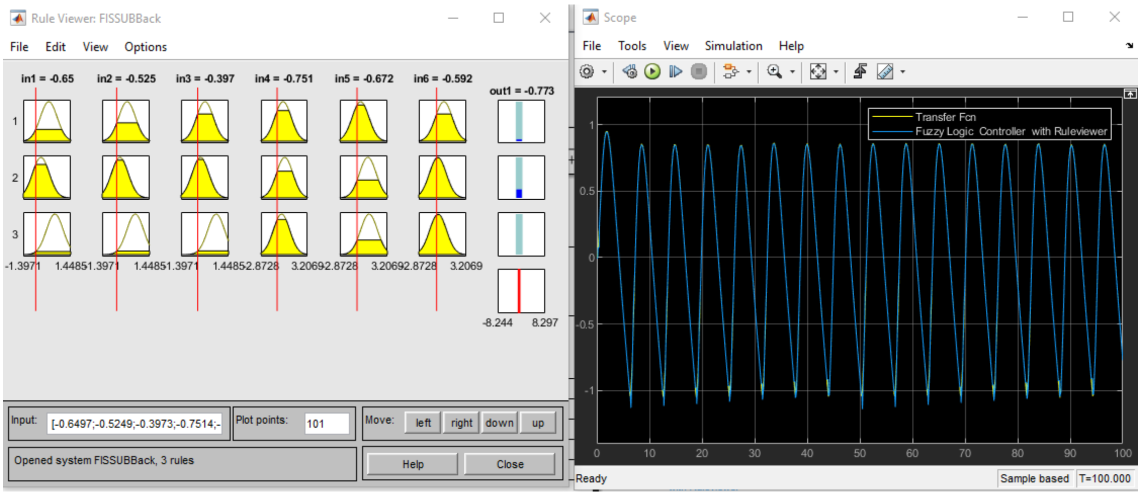
Plot points: 101

Move: left right down up

Opened system FISFCMHyb, 5 rules

Help Close

c. Subtractive with Backpropagation



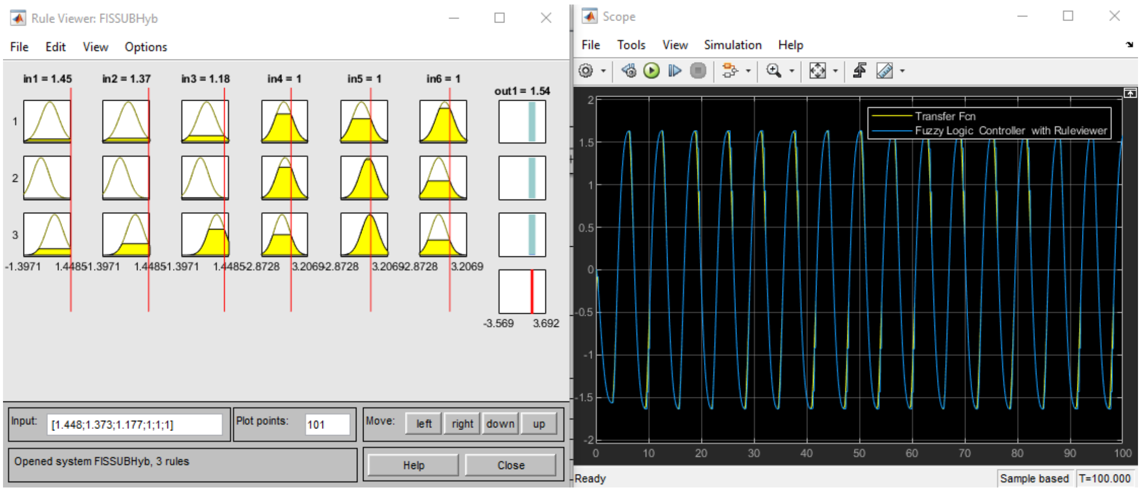
Rule Viewer: FISSUBBack

Input: [-0.6497;-0.5249;-0.3973;-0.7514;-

Plot points: 101

Opened system FISSUBBack, 3 rules

d. Subtractive with Hybrid



Rule Viewer: FISSUBHyb

Input: [1.448;1.373;1.177;1;1;1]

Plot points: 101

Opened system FISSUBHyb, 3 rules