University Of Balamand

Faculty of Engineering

Fuzzy Logic Control

An Application of Fuzzy Clustering

Presented to: Dr. Issam Dagher

Presented by: Kifah DAHER

Date: 09-01-2013

Introduction:

This project is a survey of fuzzy set theory applied in cluster analysis. These fuzzy clustering algorithms have been widely studied and applied in a variety of substantive areas. it is one of the fuzzy clustering techniques based on objective function. Cluster analysis is one of the major techniques in pattern recognition. The importance of clustering is well appreciated in various areas such as engineering systems and image processing. The conventional clustering methods restrict that each point of the data set belongs to exactly one cluster. Fuzzy set theory gives an idea of uncertainty of belonging, which was described by a membership function. The use of fuzzy sets provides imprecise class membership information. The purpose of this project is briefly to survey the fuzzy set theory applied in cluster analysis.

Design:

The task is to identify a system using the fuzzy clustering approach. It is the problem of identifying the number of classes according to certain criterion and assigning the membership function of the system. Using Fuzzy C-means algorithm FCM, under MATLAB the function to be identified is sinusoidal y = sin (x) + noise. Based on the local least square technique the user enters the number of desired clusters and the desired threshold.

The system to be studied:

$$y = \sin(x)+0.1*\mathrm{rand}(1)$$

$$x= [0.:0.1*pi:2*pi]$$

The random number generated is considered the noise. We start by collecting the data and entering the data into FCM in order to get the Centers and the U matrix.

The approximation let us convert a non-linear system into linear regions using these regions the approximated output $\hat{y}$ is derived.

The code describing the work is the following:

```
clc
clear all
x=[0:0.1*pi:2*pi];
y=sin(x)+0.1*rand(1);
plot(x,y)
hold on
data=[x',y'];
l=length(data);
plot(data(:,1), data(:,2),'o','color','g');
xlabel('x','fontweight','bold')
ylabel('y=sin(x)+0.1*rand(1)','fontweight','bold')
disp('Enter the number of desired clusters');
n=input('');
[center,U]=fcm(data,n)
for b=1:n
    plot(center(b,1),center(b,2),'ko','LineWidth',2,'markersize',20);
end
t=input('Enter the desired threshold: ');
for i=1:n
    u(i,:)=U(i,:)>t;
    X(i,:)=u(i,:)'.*data(:,1);
    d=X(i,:)';
```

```
    d=d(d~=0);
    Y(i,:)=u(i,:)'.*data(:,2);
    e=Y(i,:)';
    e=e(e~=0);
    xl{i}=[d,ones(length(d),1)];
    theta{i}=((xl{i}'*xl{i})^-1)*xl{i}'*e;
    theta{i}
    hold on
    plot(d(d~=0), theta{i}(1)*d(d~=0)+theta{i}(2),'linewidth',2,'color','r');
end
figure
hold on
for a=1:n
    plot(x,U(a,:));
end
for i=1:1:length(x)
    for b=1:n
       B(b,1)= theta{b}(1)*x(i)+theta{b}(2);
    end
 y(1,i)=U(:,i)'*B;
 yh=y/sum(U(:,i));

end
figure
plot(x,yh);
```

We start by collecting the data, then asking the user for the number of desired clusters and for the desired threshold based on which the algorithm will continue.

$\hat{y} = a_i x + b_i$ Once the data exceeding the threshold which are close to the center given by the fcm function, they are fit into a line. Once building the equation we can derive **a** & **b** coefficient of the line equation.

Suppose $$y = \begin{bmatrix} y3 \\ y4 \\ y5 \end{bmatrix} = \begin{bmatrix} X3 & 1 \\ X4 & 1 \\ X5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

It is to derive the a & b so the line equation Y= X Θ where $\theta = \begin{bmatrix} a \\ b \end{bmatrix}$

Using linear algebra $X^T Y = X^T X \Theta \Rightarrow \Theta = (X^T X)^{-1} X^T Y$

This to represent the following rules:

$$\text{If x is A}_1 \text{ then } \hat{y} = a_1 x + b_1$$

$$\text{If x is A}_2 \text{ then } \hat{y} = a_2 x + b_2$$

$$\text{If x is A}_3 \text{ then } \hat{y} = a_3 x + b_3$$

Figure 1 below represent the signal plotted in blue and the data matching the sinewave plotted in green circles.

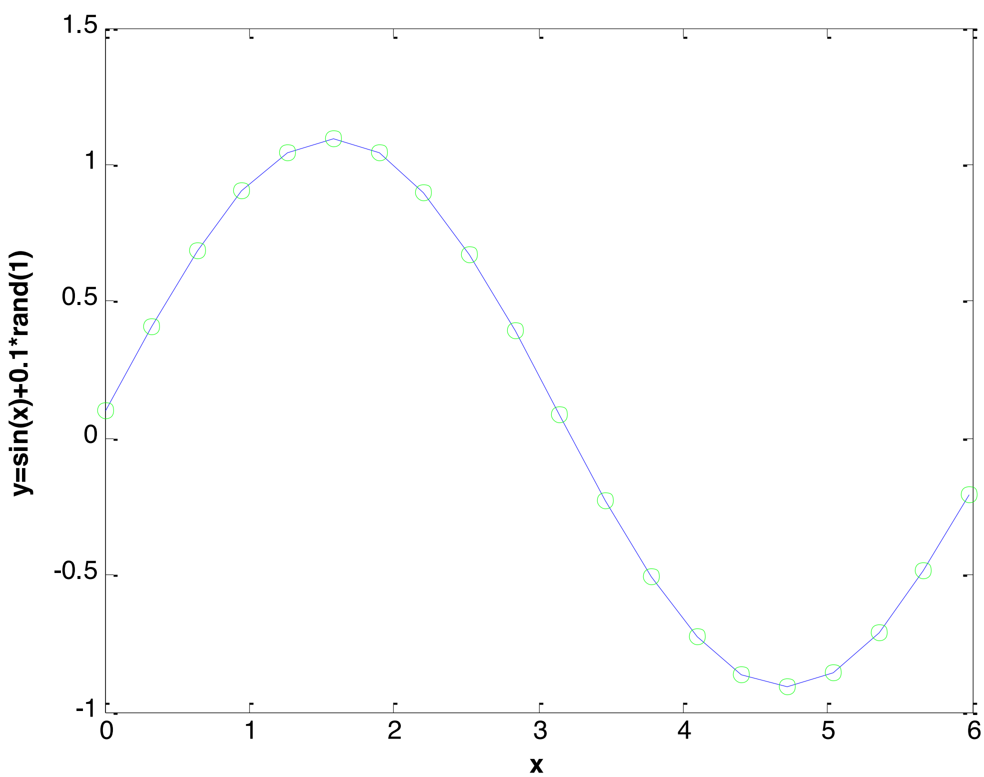


Figure 1: the signal and the data plot

For a first study we consider 3 clusters and 0.7 threshold, figure 2 shows the location of the three centers for each cluster,

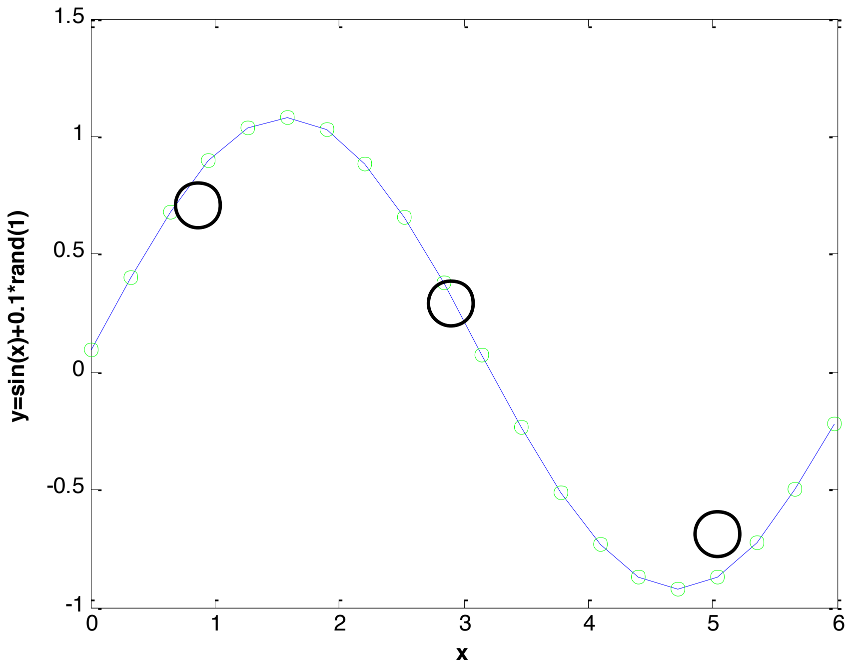


Figure 2: Centers location for 3 clusters

The resulting data from fcm function are the centers and the U matrix here below in figure 3:

```
center =

    5.0395   -0.6908
    2.8946    0.2884
    0.8652    0.7102


U =

  Columns 1 through 8

    0.0366    0.0155    0.0025    0.0021    0.0141    0.0326    0.0491    0.0514
    0.1135    0.0548    0.0099    0.0099    0.0755    0.2091    0.4079    0.6550
    0.8498    0.9297    0.9876    0.9880    0.9105    0.7584    0.5430    0.2936

  Columns 9 through 16

    0.0298    0.0019    0.0263    0.1729    0.4365    0.6984    0.8716    0.9609
    0.8820    0.9952    0.9544    0.7666    0.4930    0.2516    0.1034    0.0307
    0.0882    0.0029    0.0193    0.0605    0.0706    0.0500    0.0250    0.0084

  Columns 17 through 20

    0.9931    0.9820    0.9355    0.8666
    0.0053    0.0136    0.0482    0.0979
    0.0016    0.0043    0.0163    0.0354
```

Figure 3: Centers and U matrix

Then we the lines parameters $\theta = \begin{bmatrix} a \\ b \end{bmatrix}$ and since we have 3 centers 3 lines are expected:

```
ans =

a1    0.4255
b1   -2.8946

ans =

a2   -0.9562
b2    3.0804

ans =

a3    0.6435
b3    0.1831
```

Then the lines are plotted in each region as shown in figure 4 below:

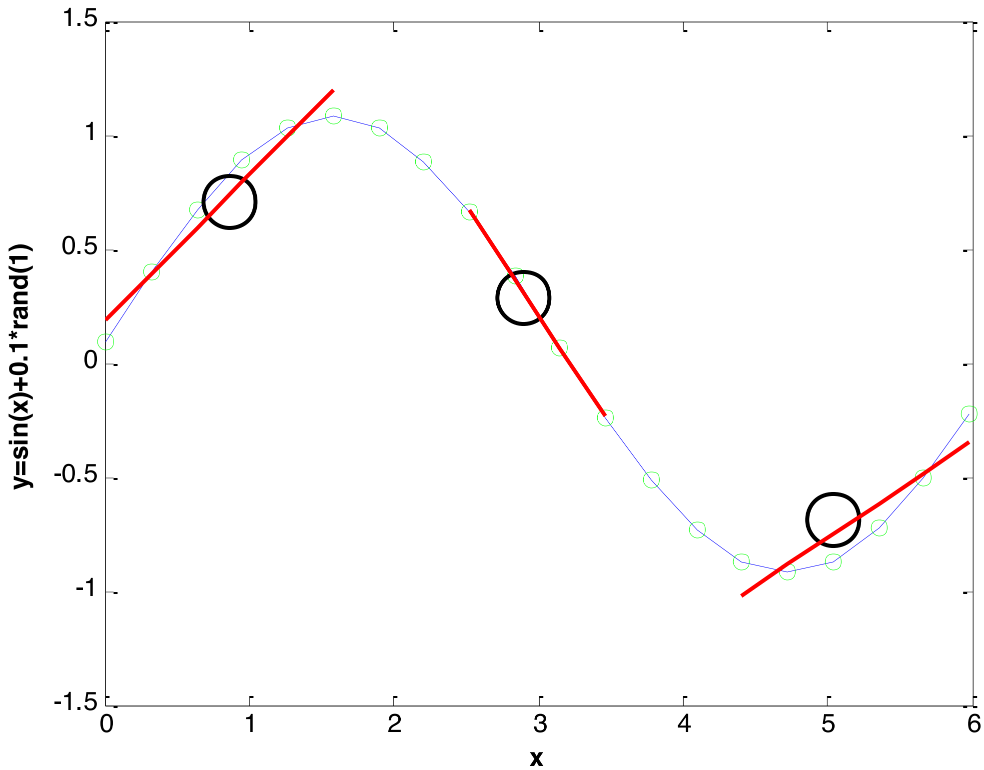


Figure 4: the lines fitting the equation in each region

The membership functions are shown in figure 5, they are discrete and not continuous as we can see.

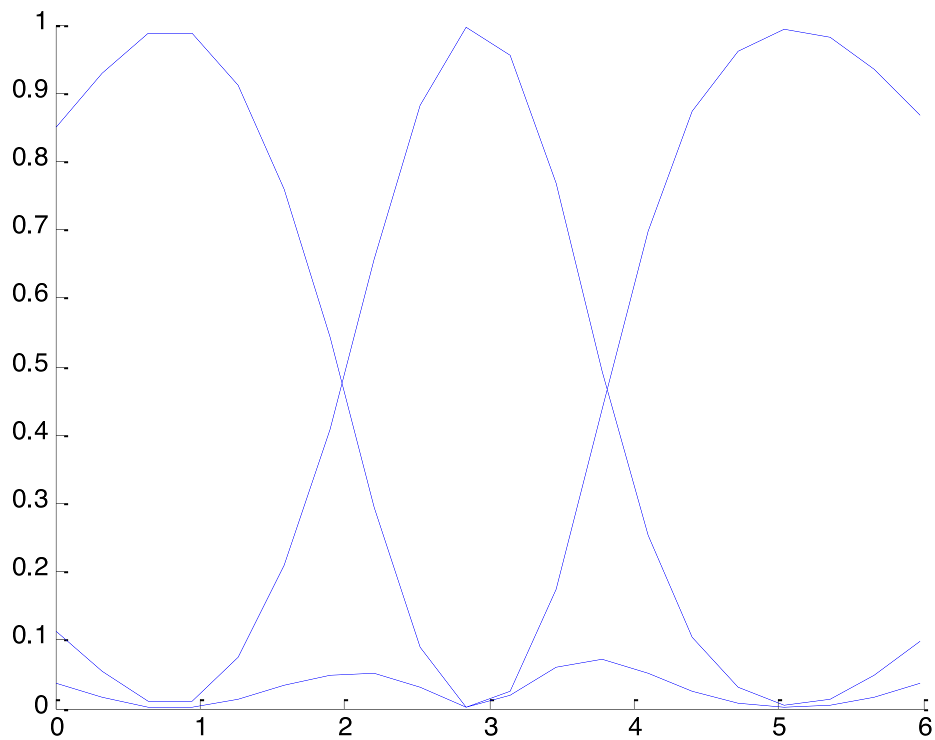


Figure 5: Membership Functions

Finally to test the model we derive $\hat{y}$ of the overall system; it is show in figure 6

$$\hat{y} = \frac{\mu A_1(x)(a_1 x + b_1) + \mu A_2(x)(a_2 x + b_2) + \mu A_3(x)(a_3 x + b_3)}{\mu A_1(x) + \mu A_2(x) + \mu A_3(x)}$$

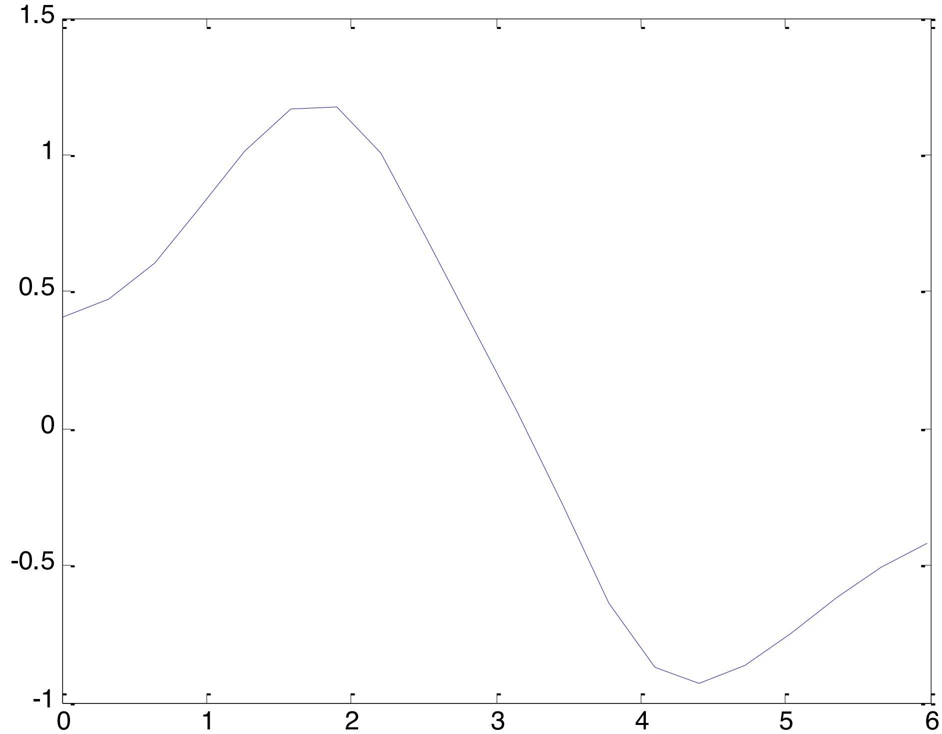


Figure 6: the model output

As we can see the result of the model is not a perfect sinusoidal but for 3 clusters the result is acceptable. For more analysis we redo the model for 6 clusters:

For 6 clusters we have the following results: 6 centers

```
center =

    2.4715    0.6291
    5.6834   -0.5083
    0.2874    0.3053
    3.4913   -0.2960
    4.5895   -0.9079
    1.3651    0.9628
```

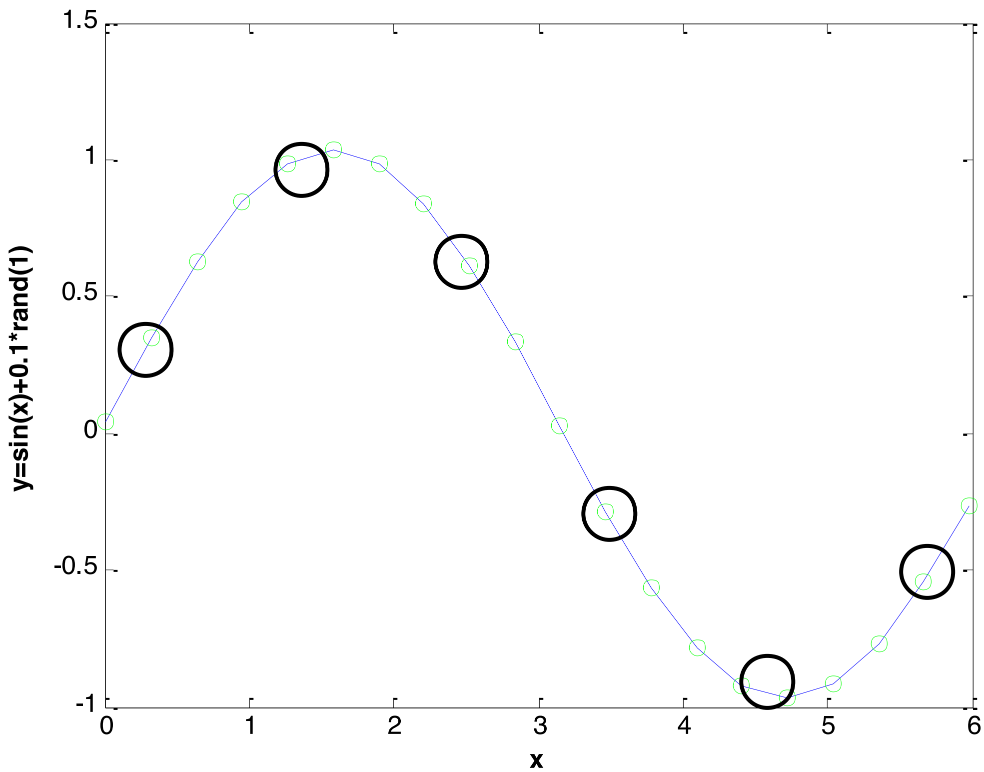


Figure 7: Centers location (6 clusters)

The line equations parameters are the following:

ans =

a1= -0.7112

b1= 2.4080

ans =

a2= 0.8015

b2= -5.0674

ans =

a3= 0.9820

b3= 0.0340

ans =

a4= -0.8828

b4= 2.7748

ans =

a5= -0.1459

b5= -0.2769

ans =

a6= 0.2946

b6= 0.5836

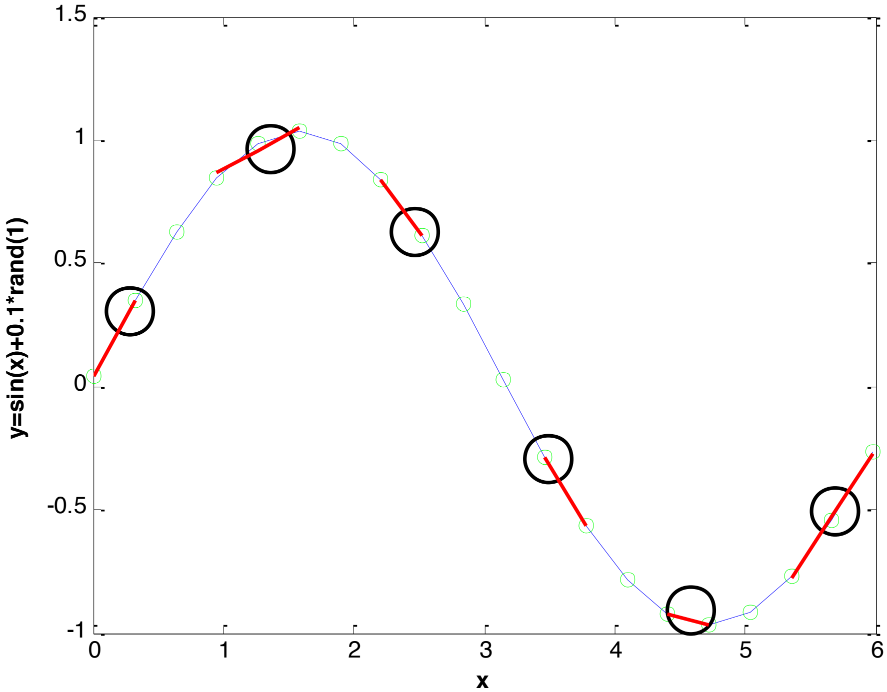
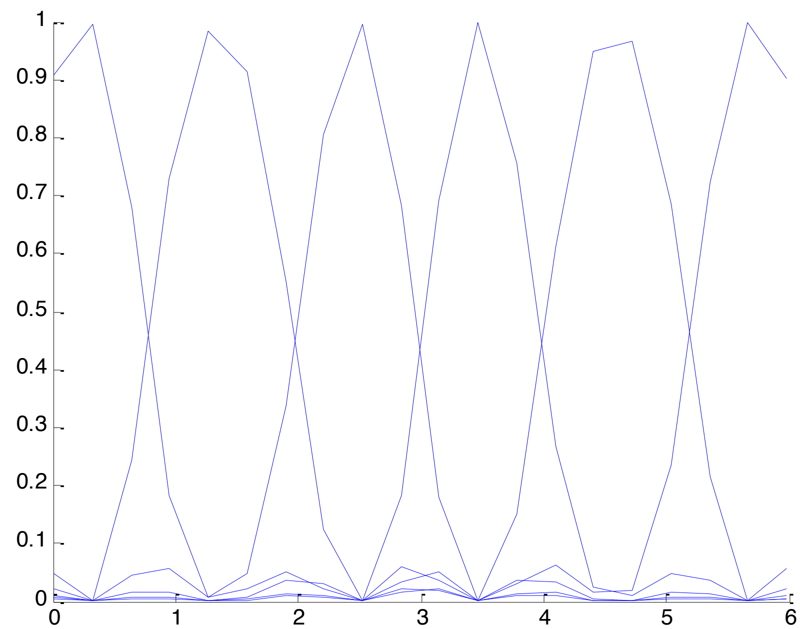


Figure 8: the lines fitting the values (6 clusters)



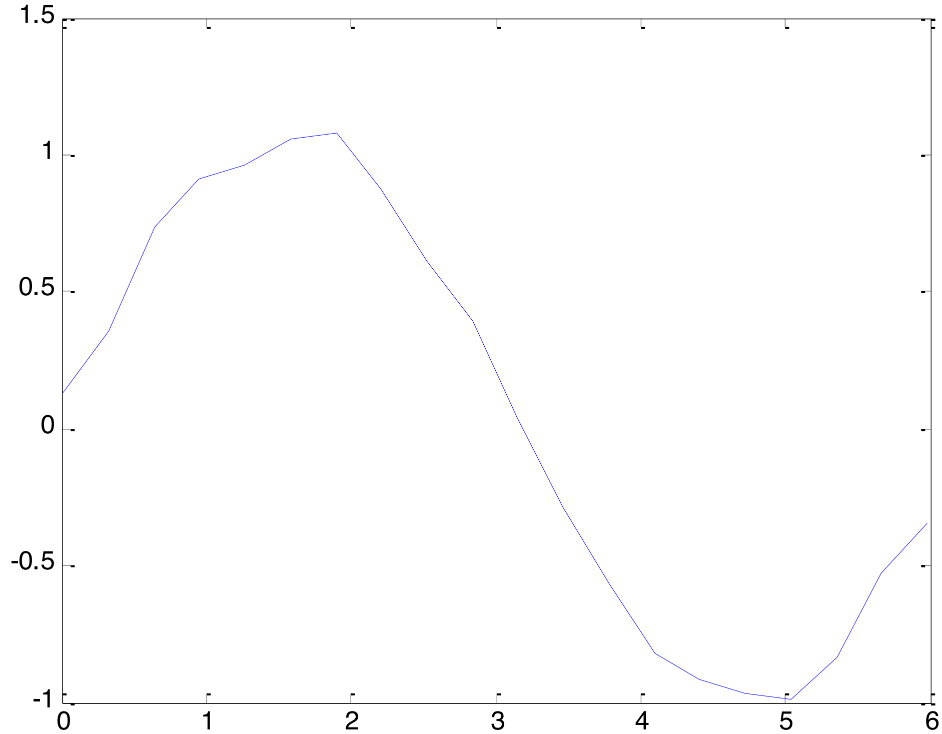


Figure 9: the model output (6 clusters)

As we can see the model output has a better shape and better in 6 clusters than 3 clusters which is logical.

Conclusion:

An effectively formal way to reconstruct a non linear model is by dividing it into linear region then reconstructs the output. We conclude that the fuzzy clustering algorithms have obtained great success in a variety of substantive areas. The presented may give a good extensive view that with the increase of the cluster numbers the output signal is more precise.