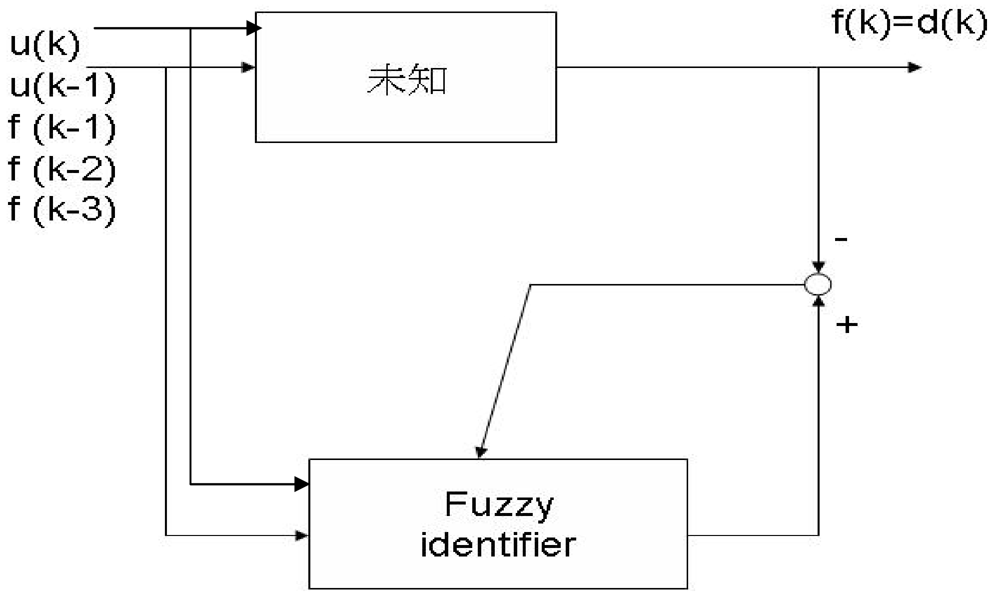
A discrete nonlinear system：



用ＦＵＺＺＹ去模擬那個未知的ＰＬＡＮＴ，Ｕ（k）是輸入訊號，Ｆ(k）是輸出訊號，Ｕ（k－１）是上一筆輸入訊號，Ｆ(k）=d（k）是輸出訊號，利用F（k－１）和F（k－２）和F（k－３）和u（k−1）和u（k）來比較模擬ＰＬＡＮＴ輸出。將未知輸出訊號與ＦＵＺＺＹ加減，再把值拉回ＦＵＺＺＹ做修正。

Algorithm：

$$d(k)=x(k)=\frac{x(k-1)\cdot x(k-2)\cdot x(k-3)+u(k)}{1+x^2(k-2)+x^2(k-3)}$$

$u(k)=\sin(0.01k\pi), k=1,2,...$

$d(k)=f(k)$

$x_1(k)=u(k)$

$x_2(k)=u(k-1)$

$x_3(k)=f(k-1)$

$x_4(k)=f(k-2)$

$x_5(k)=f(k-3)$

$$y(k)=\frac{\sum_{l=1}^{3}\{\bar{y}_l(k)\pi_{i=1}^{5}\exp[-(\frac{x_i(k)-\bar{x}_{li}(k)}{\sigma_{li}(k)})^2]\}}{\sum_{l=1}^{3}\{\pi_{i=1}^{5}\exp[-(\frac{x_i(k)-\bar{x}_{li}(k)}{\sigma_{li}(k)})^2\}}=\frac{\sum_{l=1}^{3}\bar{y}_l(k)r_l(k)}{\sum_{l=1}^{3}r_l(k)}=\frac{v(k)}{w(k)}$$

步驟：

1. using $x_1(k)=\sin(0.01k\pi)$ $x_2(k)=u(k-1)$ $d(k)=x(k)=\frac{x(k-1)\cdot x(k-2)\cdot x(k-3)+u(k}{1+x^2(k-2)+x^2(k-3)}$

   to compute plant output $d(k)=f(k)$

2. use $y(k)=\frac{\sum_{l=1}^{3}\{\bar{y}_l(k)\pi_{i=1}^{5}\exp[-(\frac{x_i(k)-\bar{x}_{li}(k)}{\sigma_{li}(k)})^2]\}}{\sum_{l=1}^{3}\{\pi_{i=1}^{5}\exp[-(\frac{x_i(k)-\bar{x}_{li}(k)}{\sigma_{li}(k)})^2\}}=\frac{\sum_{l=1}^{3}\bar{y}_l(k)r_l(k)}{\sum_{l=1}^{3}r_l(k)}=\frac{v(k)}{w(k)}$ to compute $r_e(k)$

   $w(k)$ $v(k)$ $y(k)$

3. using $\bar{y}_l(k+1)=\bar{y}_l(k)+\alpha\frac{(y(k)-d(k))r_l(k)}{w(k)}$

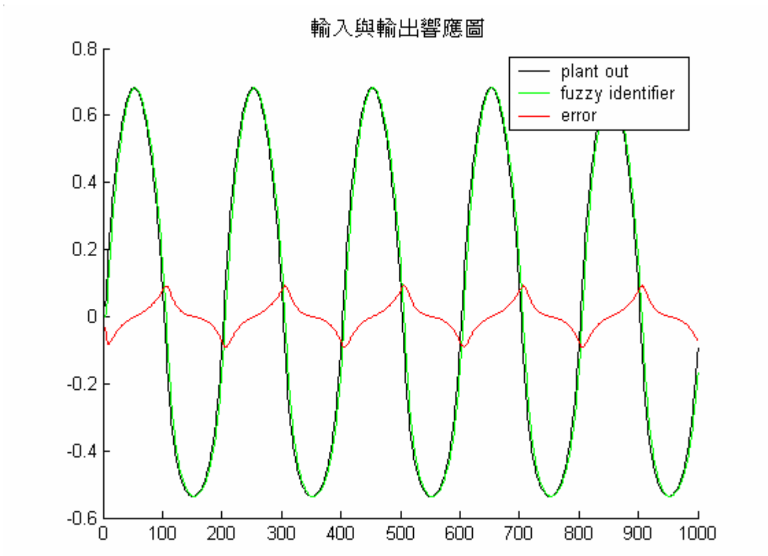
   $$\bar{x}_{li}(k+1)=\bar{x}_{li}(k)+\alpha\frac{2(y(k)-d(k))(\bar{y}_l(k)-y(k))r_l(k)(x_i(k)-\bar{x}_{li}(k))}{w(k)\delta_{li}^2(k)}$$

   $\delta_{li}(k+1)=\delta_{li}(k)+\alpha\frac{2(y(k)-d(k))(\bar{y}_l(k)-y(k))r_l(k)(x_i(k)-\bar{x}^{2}_{li}(k))}{w(k)\delta_{li}^3(k)}$ to train $\bar{y}_l(k+1)$

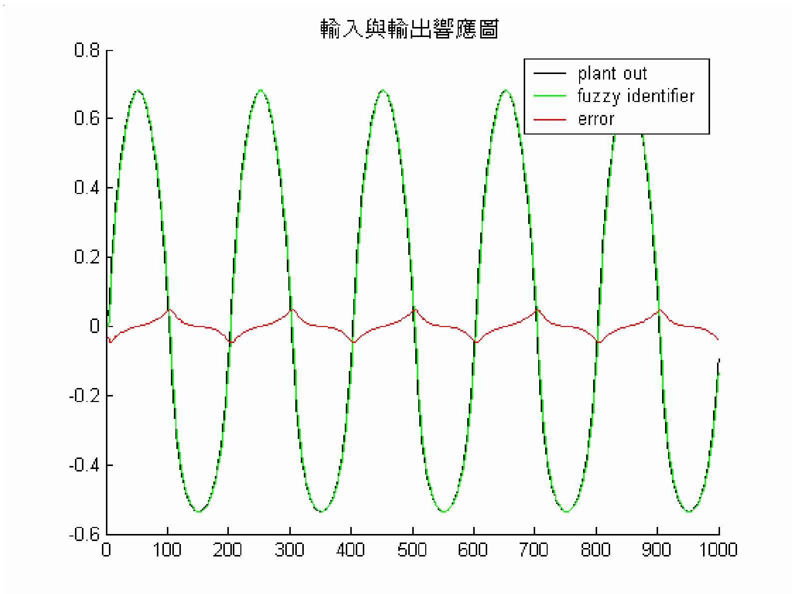
   $\bar{x}_{li}(k+1)$ $\delta_{li}(k+1)$ let $k=k+1$ goto step 1

輸出結果：

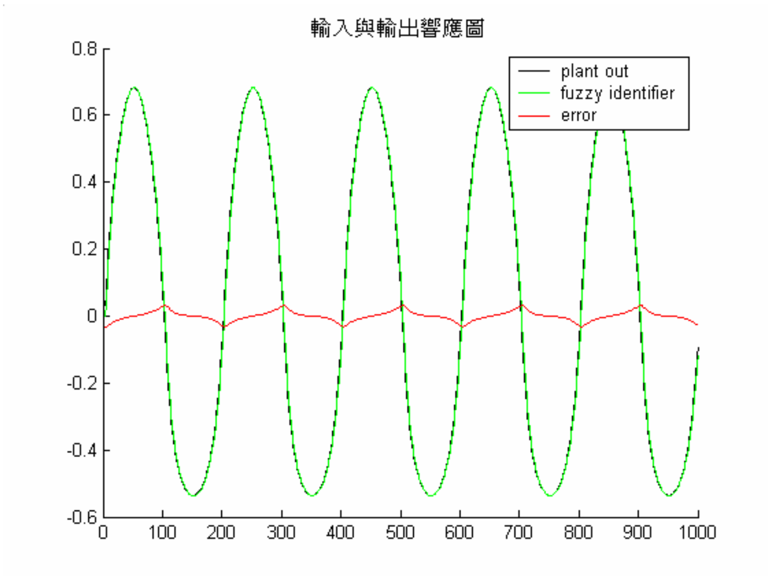
當$\alpha=1$，對應輸出波形



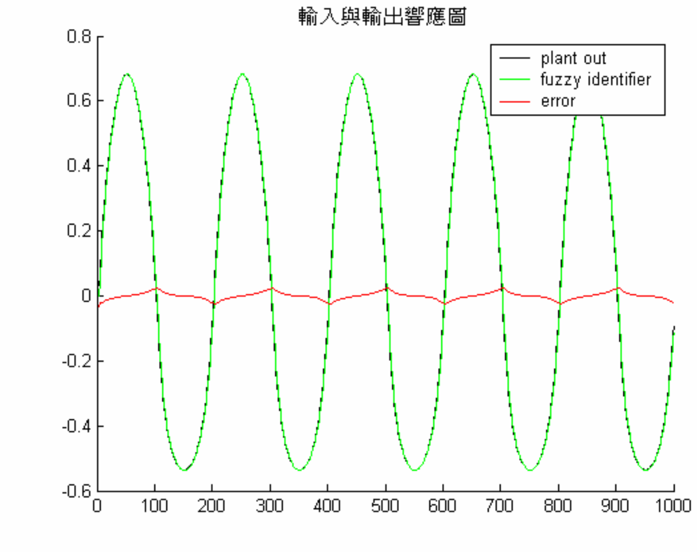
當$\alpha=2$，對應輸出波形



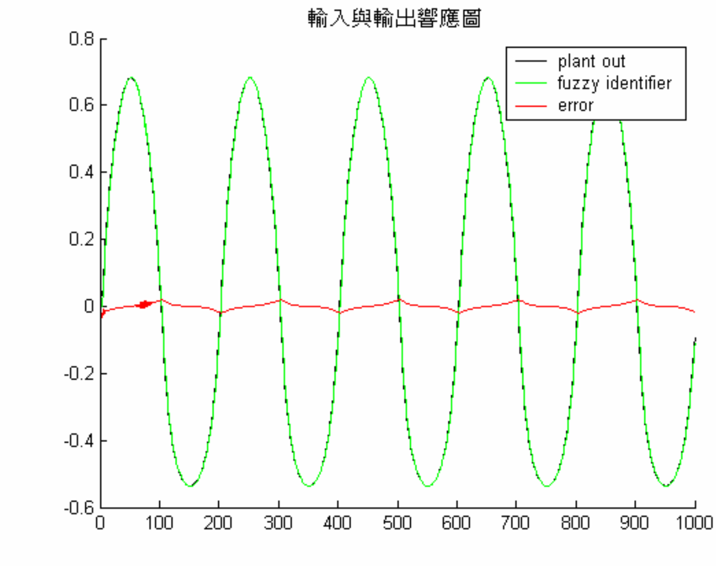
當$\alpha=3$，對應輸出波形



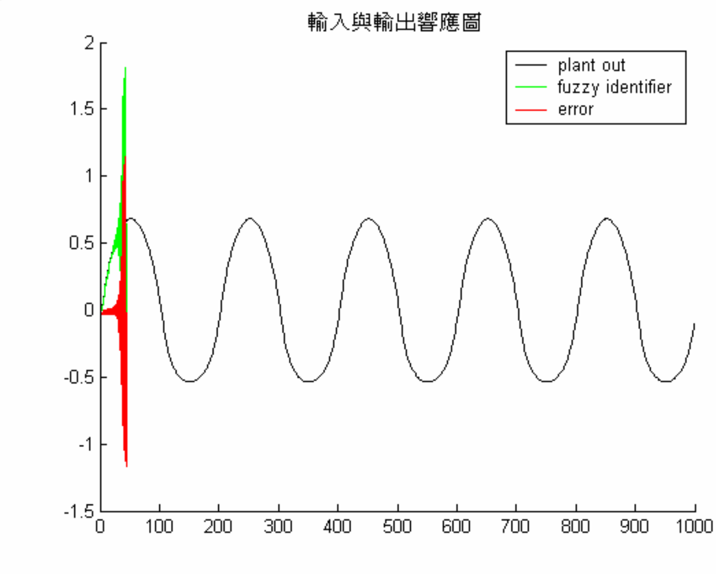
當 $\alpha = 4$ ，對應輸出波形



當 $\alpha = 5$ ，對應輸出波形



當 $\alpha = 6$ ，對應輸出波形



心得：

在撰寫程式的過程中，$\overline{y}_l(k)$　$\overline{x}_{li}(k)$　$\delta_{li}(k)$ 在經過計算過程中，會慢慢收斂，所以一開始的假設，則可隨意設置（盡量不要設定為0），而α的值，則是自己給定的，從上面的輸出圖形和誤差值可以得，α的選擇很重要，當選擇太低的值的話，誤差訊號會比較大，在收斂的速度會比較慢，而選擇太高的值的話，輸出訊號再一開始就發散而無法收斂（就像最後一個圖型一樣），所以在選擇α的值的時候，要特別注意這點。

程式：

```
clear
clc
%設定暫存器
x(5,1000)=0;acentric=[2 2 2 2 2;2 2 2 2 2;2 2 2 2 2];
ambulate=[0 0 0 0 0;0 0 0 0 0;0 0 0 0 0];k=4;total1=0;
total2=0;conter=[0 0 0];rate=4;error(1000)=0;d(1000)=0;
%計算輸入訊號
for i=1:1000
    x(1,k)=sin(0.01*i*pi);
    x(2,k)=x(1,k-1);
    k=k+1;
end
%計算 plant 輸出訊號
k=4;
for i=1:1000
    d(k)=(x(1,k-1)*d(k-1)*d(k-2)*d(k-3)+x(1,k))/(1+d(k-2)^2+d(k-3)^2);
    x(3,k)=d(k-1);
    x(4,k)=d(k-2);
    x(5,k)=d(k-3);
    k=k+1;
end
%計算 Fuzzy identifier 的輸出訊號
k=4;
for z=1:1000
    sum=1;
    total1=0;
    total2=0;
    for l=1:3
        for i=1:5
            sum=sum*exp(-((x(i,k)-ambulate(l,i))/acentric(l,i))^2);
        end
```

```
        total1=total1+sum*conter(l);
        total2=total2+sum;
    end
    output(k)=total1/total2;
    sum=1;
%計算 y_l(k+1)  x_li(k+1)  δ_li(k+1) 的值
   for l=1:3
        for i=1:5
            sum=sum*exp(-((x(i,k)-ambulate(l,i))/acentric(l,i))^2);
            temp1=ambulate(l,i);
            temp2=acentric(l,i);

ambulate(l,i)=temp1-(rate*2*(output(k)-d(k))*(conter(l)-output(k))*sum*(x(i,k)-temp1))/(total2*temp2^2);

acentric(l,i)=temp2-(rate*2*(output(k)-d(k))*(conter(l)-output(k))*sum*(x(i,k)-temp1)^2)/(total2*temp2^3);
        end
        conter(l)=conter(l)-(rate*(output(k)-d(k))*sum)/total2;
    end
    error(k)=output(k)-d(k);
    k=k+1;
end
%畫出輸入、輸出、誤差訊號響應圖
i=4:1000
plot(i,error(i));
title('誤差訊號圖');
figure
hold on
i=4:1000
plot(i,d(i),'k');
plot(i,output(i),'g');
plot(i,error(i),'r');
title('輸入與輸出響應圖');
legend('plant out','fuzzy identifier','error');
```