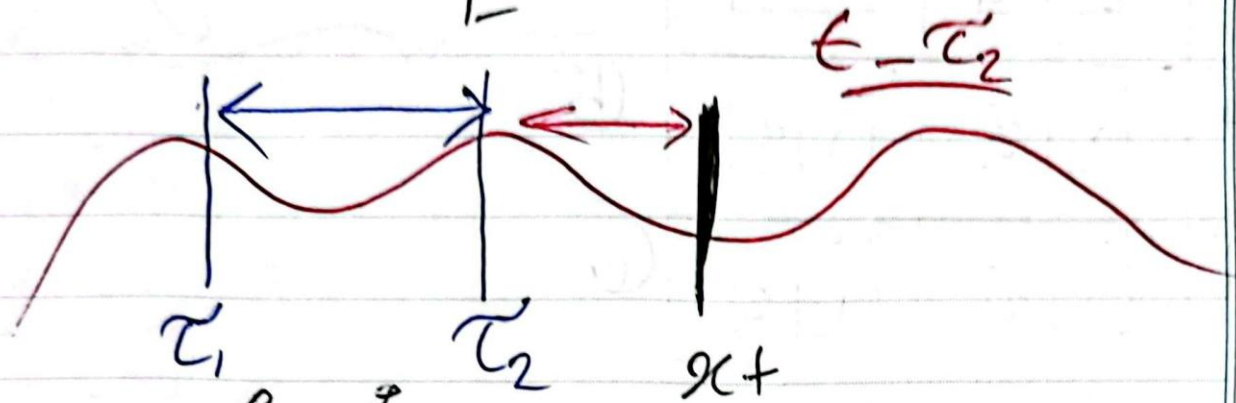


مثلاً ما یک سری از اعداد داریم



$$x_t = f(\{x_\tau \mid \tau_1 \leq \tau \leq x_t\})$$

می‌خواهیم پیش‌بینی x_t انجام دهیم
گاهی اوقات بازه τ_1 تا x_t غامض می‌دارد

به x_t اما گاهی اینفلوئنس. یعنی قبل از

آوردن آن به دست ما باید پیش‌بینی کنیم.

گاهی اوقات این محدوده صریح نیست

$$x_t = f(\{x_\tau \mid \tau \in A\})$$

KANDOO

در حالت کلی هدف ما این است که x_t را به صورت

یک سری از x_{t-8} زمان های گذشته متباین

$$x_t = f(x_{t-8_1}, x_{t-8_2}, \dots, x_{t-8_n})$$

↓
مقدارهای

تأخیرهای

این بیشترین سری ماه را در $\min 8_i$

مثال دیگر داریم

$$x_t = f(x_{t-2}, x_{t-5})$$

$t \geq 0$

داریم 4 این اعداد به هم به (t) می شود

که بیشترین این سری (5) است و سری (x)

	x_{t-5}	x_{t-2}	x_t	t
	x_2	x_3	x_1	5
Data	x_1	x_4	x_6	6
	x_0	x_5	x_7	7
	!	!	!	!

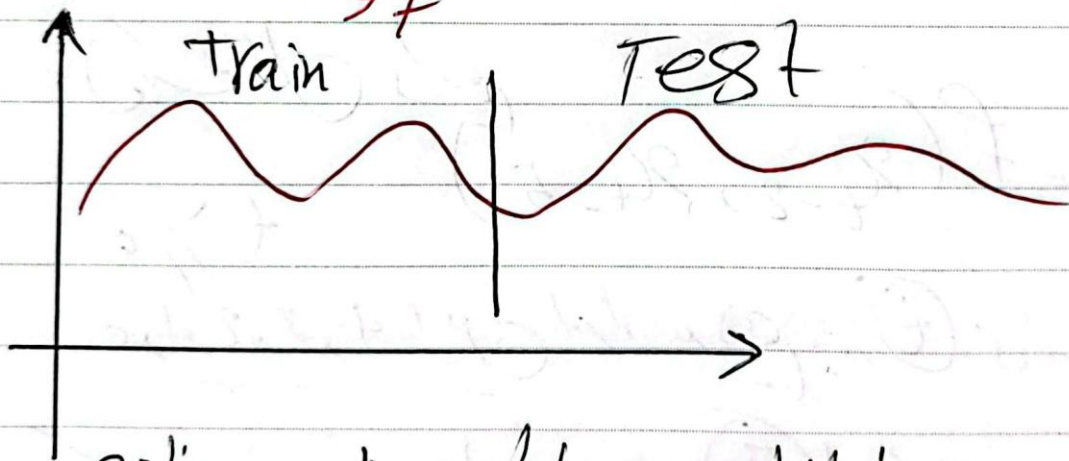
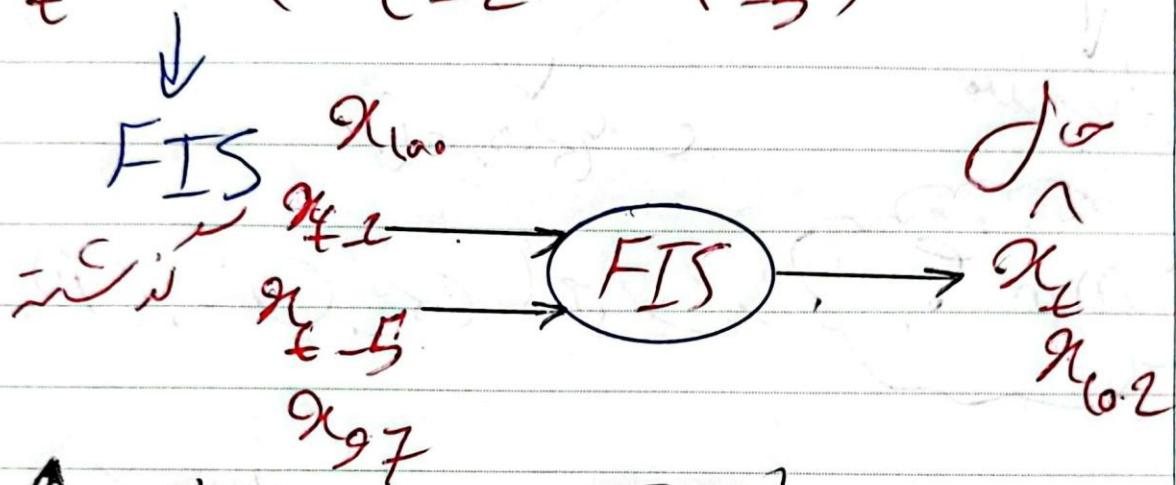
KANDOO

بہت

ایک سری اطلاعاتی راداریم جو وہی غرض
سری آن میں عزائم کے تابع بندہ۔ سیم باحد

بہت بصری

$$x_t = \hat{f}(x_{t-2}, x_{t-5})$$



حالیہ سری اطلاعاتی راداریم حل و عزائم
ایک سیم فانی راداریم کہیں ایک حل ہرگز معلوم
اللہ ربیع ہے ان آموزش ہم حال کہ سیم

KANDOO

از اطلاعات راننده راجی کل تصمیم می‌دهم.

ما از اینجا به بعد می‌خواهیم سری زمانی Mackey

glass را مدل سازی و پیش بینی کنیم که در فایل DDF

word اطلاعات کامل در آن است.

$$\dot{x}(t) = \beta \frac{x_c}{1 + x(t - \tau)^n}$$

← و راننده ناظر

این سیستم خیلی آشوب ناک نیست به نحوی که

حاصلیت خیلی زیاد در به شرایط اولیه خود دارد.

فب بزرگ مداسی Mackey glass در صلب

یک سری فایل های آماده در بر دارد.

$$B = 0.2$$

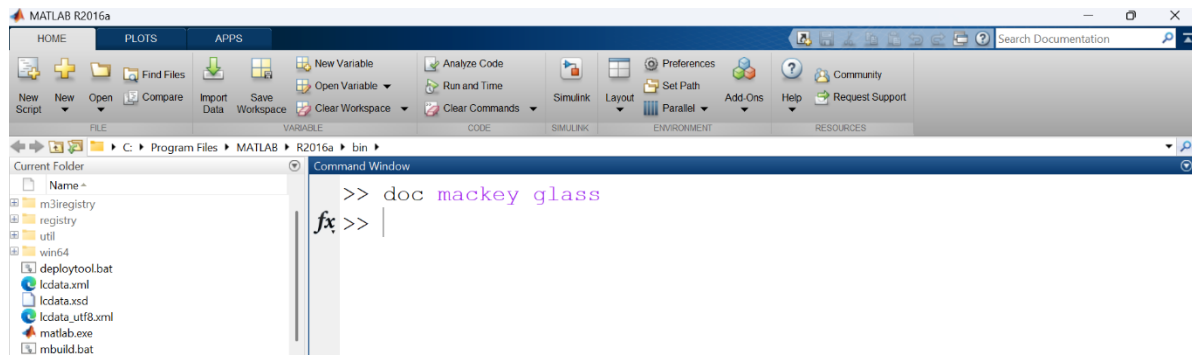
$$\tau = 17$$

$$n = 10$$

$$\gamma = 0.1$$

مقدار $\chi_0 = 0$ و
این مقدار را باید
در دسترس

برای مدلسازی سیستم فازی Mackey glass در متلب یک سری فایل‌های آماده برای این کار دارد که در ادامه برای فراخوانی و کار کردن با آن را برای شما می‌آورم :



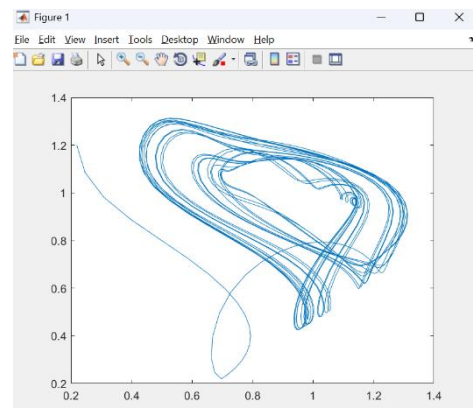
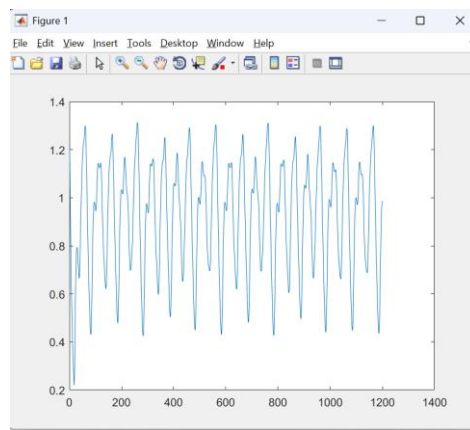
خب در ادامه کار به مطالعه بیشتر خود در رابطه با این سیستم آشوبناک می‌پردازیم .

خب برای این که با این فایل کار بخواهیم کنیم کافیه که در command window متلب بنویسیم load mgdata.dat که بنویسم یک سری اطلاعات در Workspace ما ایجاد میشود که در ادامه تصاویر مربوط به این بخش را برای شما می‌آورم .

خب در این تصویر به خوبی همه چیز نمایش داده شده است . در ادامه می‌خواهیم رفتار آشوب ناک سیستم را به صورت سری زمانی های مختلف ببینیم در اوجایی که متلب tu را عدد 17 گرفته می‌خواهیم یک دینامیک بسازیم که نمودار هایش را برای شما به نمایش می‌گذاریم :

خب در ادامه این کار ما می‌خواهیم رفتار آشوب ناک سیستم را هم برای شما با چندتا دستور ساده در متلب به نمایش بگذاریم که هست :

```
>> doc mackey glass
>> load mgdata.dat
>> x = mgdata(:,2) ;
>> plot(x)
>> x1 = x(18:end);
>> x2 = x(1:end-17);
>> plot(x1,x2)
fx>>
```



خب چیزی ئکه در بالا مشاهده شد فقط یک خروجی ساده از یک سیستم آشوب ناک که در متلب تابع آن نوشته شده است از قبل در حال حاضر ما میخواهیم که بیایم بر روی این دینامیک آشوبناک یک مدل فازی را بر روی آن سوار کنیم تا بتوانیم رفتار یک نقطه خاص را در آینده یا حال بتوانیم پیش بینی کنیم.

برای این منظور در دامه کار روال همان قبل را داریم که یک فایل m را باز میکنیم .
خب ما در ابتدای کار فایل را که ساختیم `mgdata.det` را فراخوانی کرده و تمامی سطر های آن و ستون دوم آن نیاز داریم .

خب برای پیش بینی میخواهیم ببینیم تاخیر ها را پیش بینی کنیم با سیستم فازی که این تاخیر ها شامل (6و12و18و24) است که اولین جایی که میشخه همه این تاخیر ها را تعریف کرد 25 ام عدد هست چون مابقی 0 یا منفی میشوند.

حال اولین تاخیر را مینویسیم در سیستم که میشود 25 تا آخرین سطر .

حال برای تاخیر 6 میشود $25 - 6 = 19$ خب اولین تاخیر برای عدد 6 میشود 19 .

برای 12 میشود $25 - 12 = 13$ اولین تاخیر برای عدد 12 است .

همین طور برای اعداد دیگر را هم انجام میدهیم که در اسکرین شات زیر نحوه اعمال کردن آن در متلب را هم به نمایش میگذاریم .

خب این الان در حال حاضر تاخیر
های سیستم خود را دادیم و الان
میخواهیم دیتاهای خود را درست بکنیم
که درون یک ماتریس ریخته میشود .

```
x = mgdata(:,2);
x0 = x(25:end);
x6 = x(19:end-6);
x12 = x(13:end-12);
x18 = x(7:end-18);
x24 = x(1:end-24);
```

خب با این کار خو ماتریس ورودی و خروجی خو را ساختیم .

```
data = [x24 x18 x12 x6 x0];
```

خب اگر خروجی اینو میخواهید ببینید در workspace یک ماتریس 5 ستونی و n هم ردیف دارد.

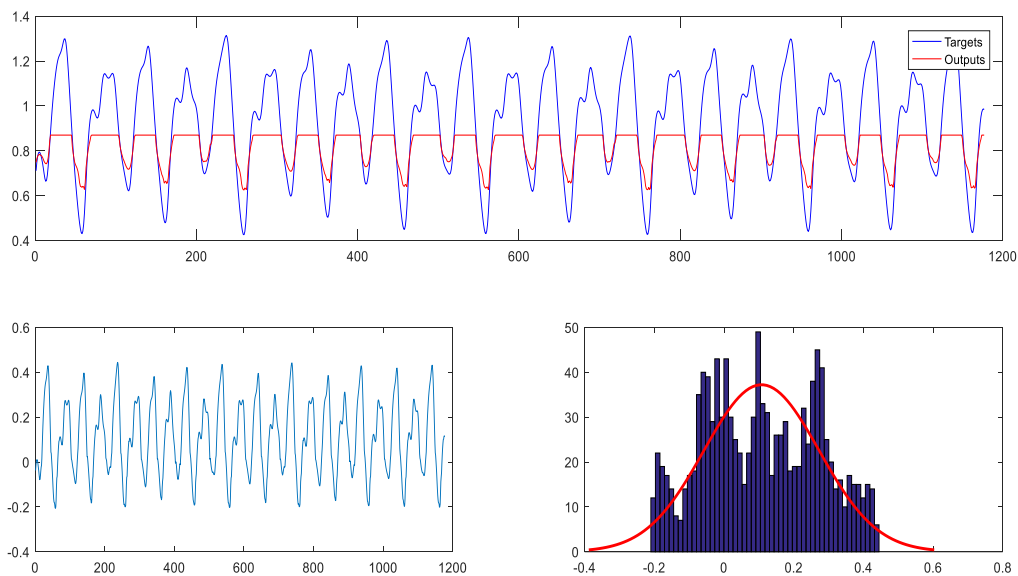
اینم از خروجی دیتنا .

Variables - data												
data												
1177x5 double												
	1	2	3	4	5	6	7	8	9	10	11	12
1	1.2000	0.6586	0.3614	0.2465	0.7107							
2	1.0858	0.5959	0.3270	0.3078	0.7500							
3	0.9825	0.5392	0.2959	0.3972	0.7756							
4	0.8890	0.4879	0.2678	0.4947	0.7896							
5	0.8044	0.4415	0.2423	0.5833	0.7940							
6	0.7278	0.3994	0.2192	0.6556	0.7903							
7	0.6586	0.3614	0.2465	0.7107	0.7802							
8	0.5959	0.3270	0.3078	0.7500	0.7648							
9	0.5392	0.2959	0.3972	0.7756	0.7453							
10	0.4879	0.2678	0.4947	0.7896	0.7226							
11	0.4415	0.2423	0.5833	0.7940	0.6975							
12	0.3994	0.2192	0.6556	0.7903	0.6752							

از اینجا به بعد باید سیستم فاری خود را بسازیم که همانند قبل از تابع که ساختیم استفاده میکنیم. کنه استم اون تابع این است که در ریپازیتوری گیت هاب خود هم آن را آپلود کرده ام .

CreateFisUsingLookupTable فقط دوتا آرگومان اول را میدهیم یعنی دیتا و تقسیم بندی دیتا های ما و مابقی اطلاعات را جوری میگذاریم تنظیم شود که تابع در درس قبلی به صورت پیش فرض گوسی فرض نموده ایم . بقیه روند کد نویسی ما همان تابعی است

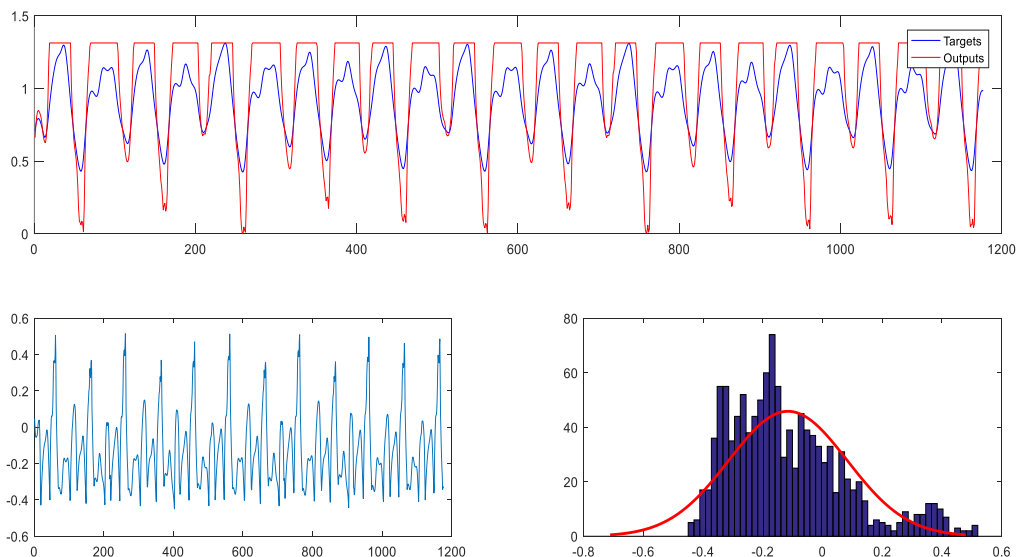
که در جلسه قبلی نوشته شده است و تیکه آخر کد نویسی ما فقط رسم نمودار ها و پلات ها است و کار خاصی ننموده ایم و خروجی کار را به نمایش میگذاریم :



خب در این تصویر دیده میشود دیتای ما قرمر رنگ است زیاد فیت نموده روی نمودار آشوبناک وجود دارد برای همین منظور ما میاییم این ها را نرمال سازی میکنیم با اضافه کردن چند خط کد به فایل خود .

```
outputs = evalfis(input,fis);  
outputs = ((outputs -min(outputs))/(max(outputs)-min(outputs)));  
outputs = min(target)+max(target)-min(target)*outputs;  
Error = target - outputs;
```

یعنی دیتا هی خود را نرمال سازی نموده ایم که تا جایی که ممکن است روی نمودار آشوب بیفتد :



بعد اضافه کردن چند خط کد خود و نرماتل سازی دیده میشود رفتار دیتا های ما نزدیک تر به نمودار دینامیک آشوب شده .

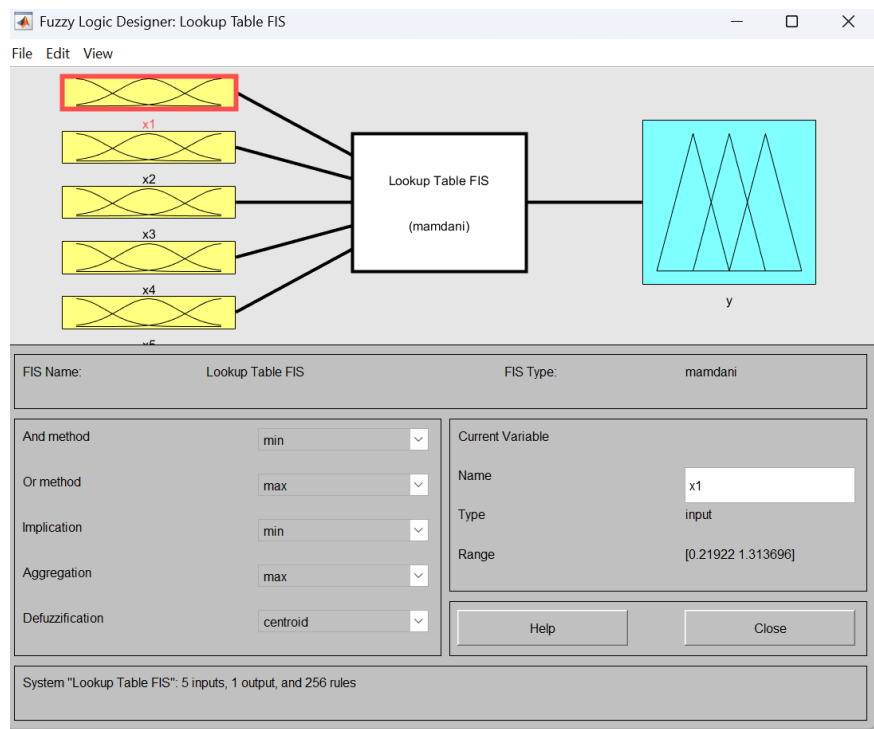
برای نرمال سازی یک دیتا به فرم زیر عمل میکنیم :

$$\frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

این نرمال سازی دیتا میباشد .

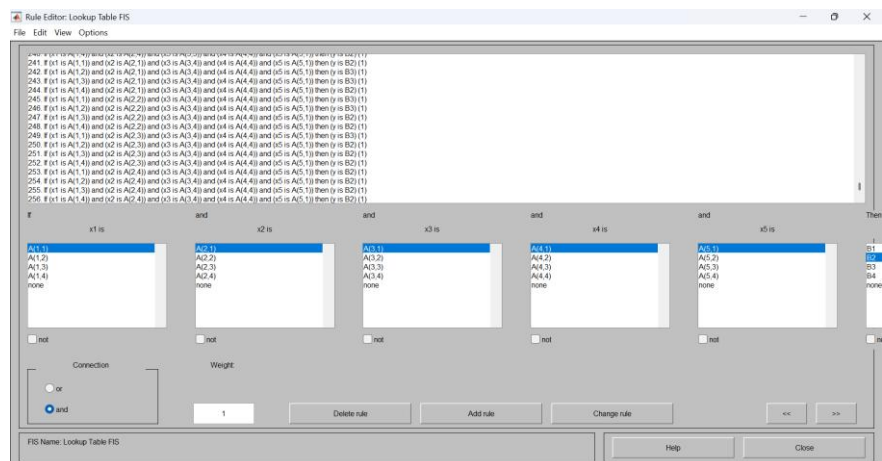
حال این نموداری که خروجی گرفتیم کاری است که با پشت پرده سیستم فازی که در متلب وجود دارد انجام می شود . برای این که بخواهیم ببینیم که در سیستم فازی چه رخ میدهد کافیه که دستور زیر را بزنید اگر با کدهای من استفاده میکنید که در ریپازیتوری من است :

خب در تصویر هم به خوبی پیدا هست که از سیستم فازی که استفاده کرده ایم از 4 ورودی تشکیل شده یک خروجی منطق فازی mamdani وجود دارد و یک سری شرط ها و قوانین هم است که ما در تابع خود از قبل برای این تعریف نموده



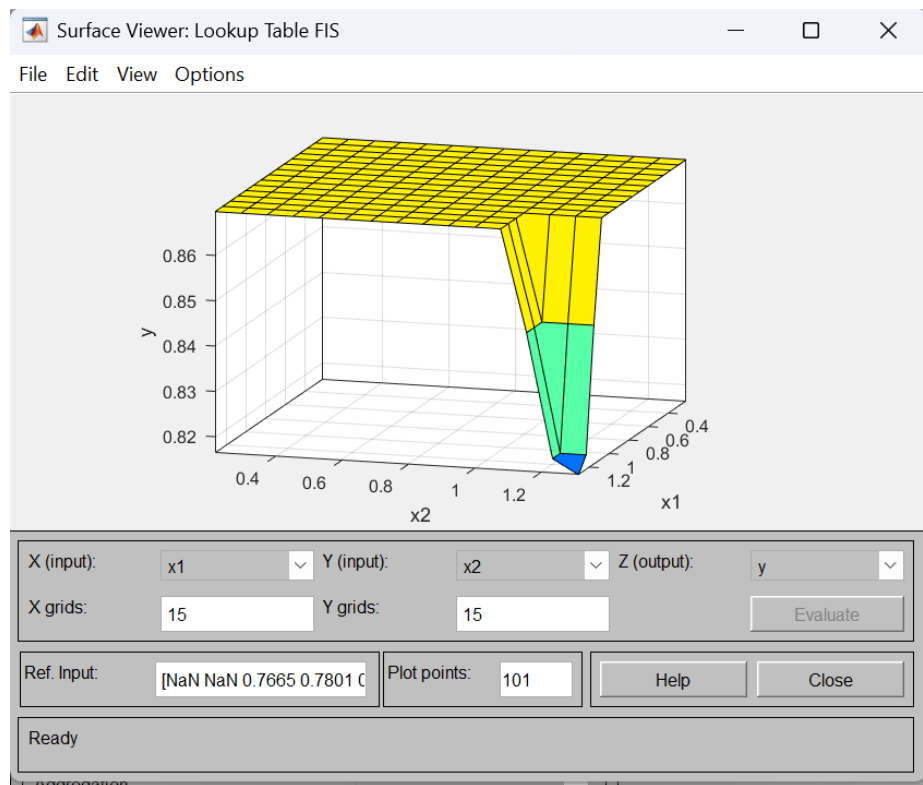
ایم که به صورت اتوماتیک کارو پیش میبرد. برای این کار 256 تا قاعده برای خود در نظر گرفته است که هم شکل قوائد و تصویر یا سطحی که به ما میدهد را به شما نمایش میدهم :

این تصویر قاعده هیابکار برده شده در سیستم فازی هست که میتونید خودتان چک کنید .



و نمودار سطحی آن هم به فرم زیر است :

اینم تصویر سطحی
از خروجی کار در
سیستم فازی که
استفاده‌ش کرده ایم
برای این دیتا‌های
خود بر حسب
دینامیک آشوب
Maeky-
glasses



برای این که دیتا‌های ما چقدر به یکدیگر نزدیک هستند در app4 یک برنامه نوشته شده است که خروجی آن با رگرسیون گیری اطلاعات ما است و نشان میدهد که اطلاعات ما چقدر به هم نزدیک است یا دور و.. و یک شکل هم از خروجی آن آورده شده است :

