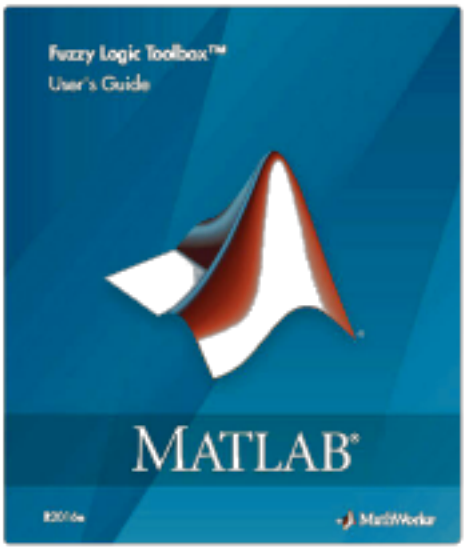
matlab1.ir

فهرست

۱- تاریخچه ۳
۱-۱- استفاده از متغیرهای زبان شناختی به جای اعداد ۶
۱-۲- کاربردها ۶
۲- جعبه ابزار منطق فازی متلب ۸
۲-۱- یک مثال: ۱۰
۲-۲- افزودن متغیر ورودی و خروجی جدید: ۱۰
۲-۳- تغییر نام متغیر ها: ۱۲
۲-۴- ذخیره سیستم فازی: ۱۳
۳- ویرایشگر توابع عضویت ۱۵
۳-۱- افزودن تابع عضویت جدید به یک متغیر ۱۸
۳-۲- توابع عضویت مورد استفاده در تولباکس متلب ۲۰
۴- پیاده سازی قوانین فازی ۲۳
۵- مشاهده Rule Surface ۲۷
۶- مشاهده قوانین ۲۸



@Matlabonline
کانال تخصصی آموزش Matlab
کلیه تولباکس ها و کلیه رشته ها
مقدماتی تا پیشرفته
کد رایگان همراه هزاران مثال حل شده
2016
دانشگاه تهران

۱- تاریخچه

دستیابی به دانش بدون ابهام، سالهای متمادی انسان را دچار چالش ساخته است. از هنگامی که ارسطو منطق دو ارزشی را معرفی کرده، تاکنون بشر توانسته با کمک و استفاده از آن به موفقیت های چشمگیری دست یابد؛ فن آوری رشد نموده و روز به روز کارآمد تر شده است.

در اوایل قرن بیستم، دانشمندان به این نتیجه رسیدند که ساختارهای سنتی علوم، پاسخگوی پدیده های کشف شده نیست. مشکلاتی که برای قوانین نیوتن در اندازه های مولکولی بوجود آمده بود، باعث شد نظر تمام دانشمندان و پژوهشگران به سمت پدیده های تصادفی جلب شود و همین امر منجر به رشد علوم آمار و احتمالات گردید. پدیده های احتمالات عباراتی بودند که بشدت در تمام شاخه های علوم بخصوص آنجا که سیستم ها پیچیده می شدند و یا تعداد مشاهدات افزایش می یافت، دیده می شد. اما آنچه احتمالات بدنبال آن بود، با ماهیت ابهامی که در سیستم ها وجود داشت، تفاوت های زیادی می کرد. با آنکه پدیده های تصادفی نمود یافته بودند، هنوز هم دانشمندان معتقد بودند که تنها راه افزایش کارآیی سیستم ها، افزایش دقت است.

منطق fuzzy گونه ای بسیار مهم از منطق است که توسط پروفسور دکتر لطفی زاده در سال ۱۹۶۵ مطرح شد و بطور جدی در مقابل منطق دودویی ارسطویی قرار گرفت و این منطق نه تنها در حوزه تئوری بلکه در صنعت نیز بکار رفته است و پژوهشگران زیادی را مشغول به تحقیق در این زمینه کرده است.

منطق fuzzy در ابتدا بعنوان روشی برای پردازش اطلاعات معرفی گردید که عضوهای یک مجموعه علاوه بر دو حالت قطعی عضو بودن و نبودن حالت بین این دو را نیز تعریف می کند. fuzzy به جای پرداختن به صفر و یک، از صفر تا یک را مورد بررسی و تحلیل قرار می دهد؛ به بیان دیگر مجموعه ای

که در منطق ارسطویی دارای دو عضو صفر ویک است در منطقfuzzy به مجموعه ای با بی نهایت عضو که دارای مقادیری از صفر تا یک هستند تبدیل می شود و بدین صورت منطق fuzzy به اعمال و طرز فکر آدمیان بیشتر نزدیک می شود.

تا دهه ۷۰ برخورد با این تئوری مجموعه ها برای کنترل سیستم ها بکار نرفت تا اینکه بعلت ناکافی بودن قابلیت های کامپیوترهای کوچکی که تا پیش از آن زمان بودند مورد توجه قرار گرفت و دکتر لطفی زاده استدلال کرد مردم نیازی به اطلاعات ورودی شمارشی بسیار دقیق ندارند و آنها هنوز قادر به کنترل تطبیقی هستند. اگر کنترل گرهای feed back (بازخورد) طوری برنامه ریزی شوند که ورودی های غیر دقیق را بپذیرند در این صورت آنها بسیار موثر تر و مفید تر خواهند بود چه بساکه ممکن است اجرای آنها بسیار آسانتر شود. اروپائیان و ژاپنی ها خیلی سریعتر از دیگر کشورها این تکنولوژی را پذیرفتند و محصولات واقعی همچون: دوربین های عکاسی، اجاقهای ماکروویو و ... در این زمینه ساختند و به جامعه بشری عرضه نمودند.

منطق fuzzy معتقد است که ابهام در ماهیت علم است. برخلاف دیگران که معتقدند که باید تقریب ها را دقیق تر کرد تا بهره وری و اثربخشی افزایش یابد، لطفی زاده معتقد است که باید به دنبال ساختن مدل هایی بود که ابهام را بعنوان بخشی از سیستم مدل نماید و سیستم هایی که اساس کار آن با دانش است جایگزین سیستم هایی که با داده ها تنظیم شده اند گردند و سیستم هایی با مرزهای قطعی و دست و پاگیر، برداشته شده و جای آنها را مرزهای خاکستری فرابگیرد.

منطق fuzzy ، حلال مسائل است و قابلیت این را دارد که هم در سیستم های میکروکنترلرهای کوچک و ساده پیاده شود و هم در کامپیوترهای چند کاناله، شبکه عظیم و یا در سیستم های کنترلی اجرا گردد. منطق fuzzy نیز در نرم افزار، سخت افزار و یا ترکیبی از آن دو می تواند کاربرد داشته باشد. منطق fuzzy روشی آسان برای رسیدن به نتایج معین بر پایه اطلاعات ورودی مبهم و غیر دقیق می

باشد. روش این منطق برای کنترل سیستم ها چگونگی تصمیم گیری یک انسان را تقلید می کند اما بسیار سریعتر و دقیق تر. مدل منطقfuzzy بر پایه و اساس تجربه بوده و بر تجربه کاربر تا فهمیدن تکنیکی سیستم تکیه دارد. بعنوان مثال فرض می شود فردی در اتاق خود مشغول مطالعه است و از آنجا که هوا گرم بوده، پنجره را کاملاً گشوده است. اگر بعد از نیم ساعت آن شخص اندکی احساس سرما نماید، چه خواهد کرد؟ در حالت طبیعی، "بلافاصله پنجره را کاملاً" خواهد بست یا "اندک اندک و به مرور زمان"آنرا خواهد بست و بعد از رسیدن به دمای مطلوب آنرا (درحالت نیمه باز و یا کاملاً بسته) رها خواهد کرد؛ فرض دوم محتمل تر است اما منطق دو ارزشی فقط یک پنجره را کاملاً باز می بیند یا کاملاً بسته.

*منطق **fuzzy** دارای خصوصیات منحصر به فردی برای کنترل بسیاری از سیستم ها می باشد؛ از جمله:*

۱-کار خود را بطور دائم ادامه می دهد؛ چرا که نیاز به ورودی های دقیق ندارد و می تواند طوری برنامه ریزی شود که اگر سنسور (حس گر) feed back قطع یا خراب شود بدون خطر و اشکال کارش را ادامه می دهد.

۲-چون روشهای کنترلر منطق fuzzy توسط کاربران تهیه می شود می تواند به راحتی اصلاح شود و تغییر کند تا عملیات سیستم را بهبود بخشد یا تغییر دهد.

۳-هر اطلاعاتی از سنسور که نشانه ای از عمل و عکس العمل های سیستم باشد برای منطق fuzzy کافی است و نیازی به تعداد کمی ورودی و یا چند خروجی کنترلی ندارد که بخواهد محاسبات را انجام دهد تا بتواند اجرا شود. این مسئله باعث می شود تا سنسورها ارزانتر باشند؛ پس هزینه و پیچیدگی سیستم کاهش می یابد.

۴-منطق fuzzy قادر است هر تعداد معقول ورودی پردازش نماید و خروجی های بی شماری را ایجاد نماید ولی چون تعیین سریع قواعد اصلی مشکل است، قواعدی که روابط متقابل بین ورودی ها و

خروجی ها را تعیین می کند هم باید مشخص شود پس بهتر است که سیستم کنترلی را به قطعات کوچک تقسیم کرد و از چندین کنترلر منطق fuzzy کوچکتر که هر کدام دارای مسئولیت محدودتری هستند، برای سیستم استفاده شود، چراکه یکی از قابلیت های منطق fuzzy همین است: "افزایش دقت با استفاده از کنترلرهای متعدد".

۵-منطق fuzzy می تواند سیستم های غیر خطی را کنترل کند که مدل کردن آنها با قواعد ریاضی بسیار سخت و یا غیر ممکن است.

۱-۱-استفاده از متغیرهای زبان شناختی به جای اعداد

پروفسور لطفی زاده در سال ۱۹۷۳ مفهوم متغیرهای fuzzy یا زبان شناختی را پیشنهاد کرد تصور کردن آنها بعنوان لغات یا موضوعات زبان شناختی بهتر از تصور کردن آنها بصورت اعداد است. ورودی های سنسور همچون : دما، جریان، فشار، سرعت و غیره هستند. در عین حال متغیرهای fuzzy خودشان صفاتی می باشند که متغیر را توصیف می کنند. بعنوان مثال: خطای (مثبت بزرگ)، خطای (مثبت کوچک)، خطای (صفر)، خطای (منفی کوچک)، خطای (منفی بزرگ). برای مینیمم کردن می توان متغیر های مثبت، صفر و منفی را برای هر یک از پارامترها در نظر گرفت. دامنه تغییرات اضافی از قبیل (خیلی بزرگ) و (خیلی کوچک) هم می توانند به محدوده پاسخگویی در شرایط استثنایی و یا بسیار غیر خطی اضافه شوند اما در سیستم اصلی نیازی به آن نیست.

۱-۲- کاربردها

منطق fuzzy تاکنون در شاخه های مختلف علوم بکار رفته است، اما شاید مهم ترین کاربردهای آنرا در سیستم های کنترلی بیابیم. از آنجایی که کنترل منطق fuzzy در ژاپن رشد فراوانی داشته است، شاید بتوان ژاپن را منشا کاربرد fuzzy در صنعت دانست. دکتر میشیوسوگنو تحقیقات فراوانی برای

کنترل کننده های fuzzy انجام داده است. او برای اولین بار کنترل کننده ی fuzzy را با حدود ۱۰۰قانون برای کنترل یک بالگرد درشرایط خطر ارائه داد. این مسئله قابل حل با روشهای کنترلی سابق نبوده و انسان هم برای کنترل بالگردها در این شرایط با مشکل مواجه بوده است. بنابراین، این مسئله یکی از مهم ترین دست آوردهای منطق fuzzy می باشد.

منطق fuzzy به عنوان روشی سودمند برای گروه بندی و کاربرد اطلاعات شناخته شده است و همین گونه ثابت گردیده که منطق fuzzy تا زمانی که از منطق کنترلی موجود بشری تقلید کند، گزینه ای عالی برای کاربرد در بسیاری از سیستم های کنترلی خواهد بود. منطق fuzzy می تواند در کامپیوترهای دستی کوچک تا سیستم های عظیم بکار رود. منطق fuzzy از یک برنامه غیر دقیق بسیار توصیفی استفاده می کند تا با اطلاعات ورودی بیشتر، شبیه یک کاربر انسان رفتار کند و و همچنان پس از خطای کاربرد به کار خود در پردازش اطلاعات ورودی و خروجی بپردازد و معمولاً در آغاز با اندک تنظیمی و یا حتی بدون نیاز به این امر شروع به کار می کند. منطق fuzzy نیازی به ورودی های دقیق ندارد و بطور ماندگار به کارش ادامه می دهد و می تواند هر تعداد معقولی از ورودی ها را پردازش کند اما پیچیدگی سیستم با ورودی ها و خروجی های بیشتر بسرعت افزایش می یابد و پردازشگرهای توزیع شده باعث آسان شدن عملیات می گردند.

امروزه در هر کجا نمی توان اثر منطق fuzzy را نادیده گرفت، از کنترل موشک و فضا پیماها گرفته تا کنترل ترافیک یک شهر بزرگ، حتی اثاثیه ها هم fuzzy شده اند؛ جارو برقی fuzzy، اجاق fuzzy، ماشین لباس شویی fuzzy و ... .

در آخر بیشترین مزیت منطق fuzzy که باعث بکار رفتن آن در رشد صنعت شده انعطاف آن در تحلیل داده ها و تصمیم گیریها است. در واقع منطق fuzzy روش دقیق فکر کردن در امور مبهم، غیر دقیق، تیره و تار و خاکستری است.

۲- جعبه ابزار منطق فازی متلب

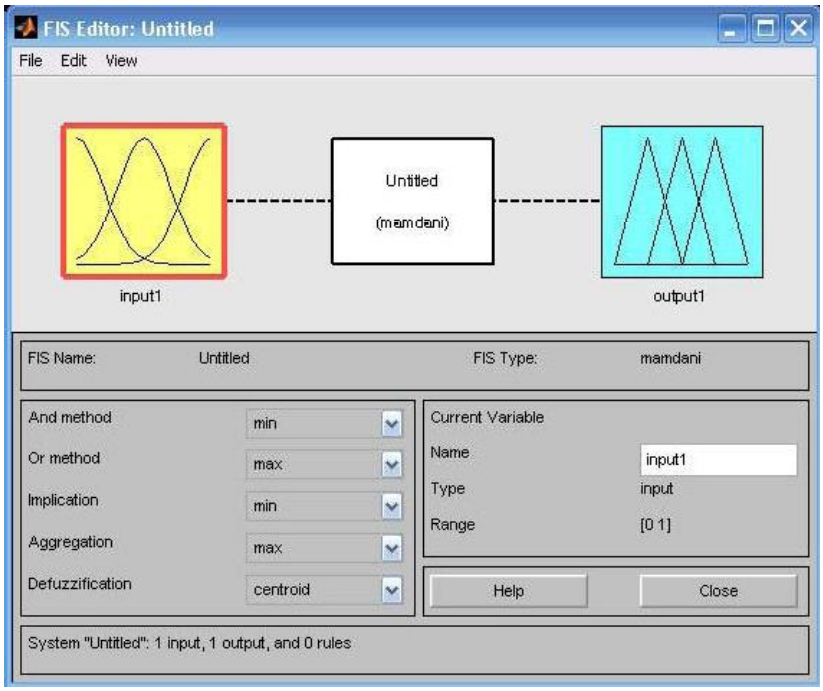
مثال انعام دادن را در نظر می گیریم. در این مثال قوانین زیر را در نظر می گیریم.

1. *If the service is poor or the food is rancid, then tip is cheap.*
2. *If the service is good, then tip is average.*
3. *If the service is excellent or the food is delicious, then tip is generous.*

برای شروع کار در خط فرمان MATLAB تایپ می کنیم.

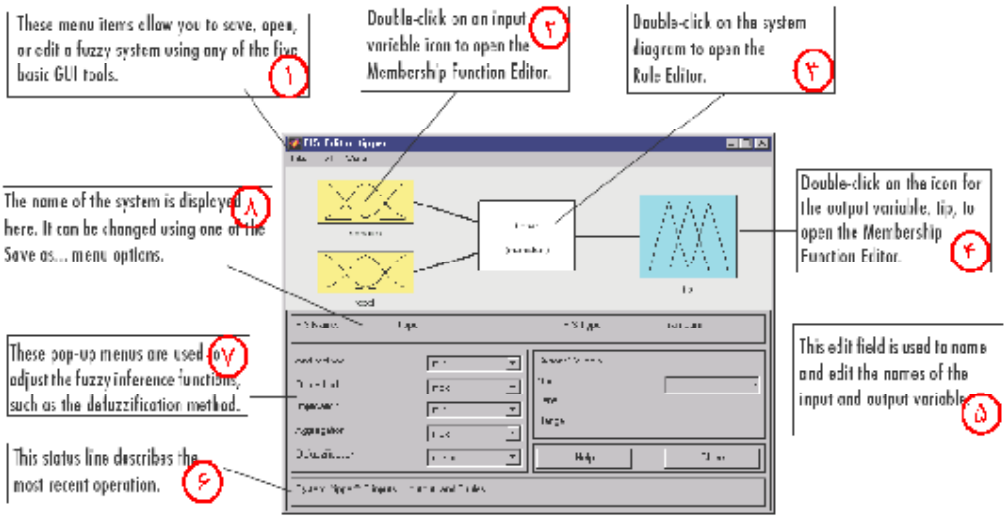
```
>> fuzzy
```

با این کار پنجره ای به صورت زیر باز می شود.



در این پنجره می توان هر نوع سیستم فازی را پیاده سازی کرده و حتی می توان ان را به صورت یک متغیر ساختاری وارد MATLAB Workspace کرده و تغییرات لازم را از خط فرمان متلب در ان ایجاد نمود.

شکل زیر اجزای این سیستم را نشان می دهد.



وظیفه هر یک از بخش ها:

بخش ۱: این منو به شما اجازه می دهد تا یک سیستم فازی را باز، ذخیره و یا ویرایش کنید.

بخش ۲: با کلیک روی یکی از متغیر های ورودی در این بخش می توانید توابع عضویت مربوط بـه ان را باز کرده و تغییرات لاز را در آن ایجاد نمایید.

بخش ۳: با کلیک بر روی سیستم در این بخش ویرایشگر قوانین باز می شود.

بخش ۴: با کلیک روی یکی از متغیر های خروجی در این بخش می توانید توابع عضویت مربوط به آن را باز کرده و تغییرات لاز را در آن ایجاد نمایید.

بخش ۵: این بخش برای نام گذاری و تغییر نام متغیرهای ورودی و خروجی استفاده می شود.

بخش ۶: این خط آخرین تغییرات را نشان می دهد.

بخش ۷: عملیات مربوط به منطق فازی را در این بخش می توانید تنظیم کنید.

بخش ۸: نام سیستم در اینجا دیده می شود.

۲-۱- یک مثال:

برای آشنایی با محیط فوق می توانید در خط فرمان متلب دستور زیر را اجرا کنید تا مثال دادن انعام (از مثالهای حل شده متلب) باز شود.

```
>> fuzzy tipper
```

با کلیک روی گزینه های مختلف، با نحوه عمل هر یک از منوها آشنا شوید.

در این نوشتار ما می خواهیم همان سیستم آماده شده در متلب را از نو بسازیم. بر ای این کار دستور زیر را اجرا می کنیم.

```
>> fuzzy
```

سیستم فازی ای که ما می خواهیم پیاده سازی کنیم، دارای ۲ ورودی و یک خروجی می باشد. ورودی اول کیفیت غذا و ورودی دوم کیفیت خدمات در رستوران می باشد. خروجی نیز میزان انعامی است که باید به پیش خدمت بدهیم. همانگونه که در بالا نیز آورده شد، طبیعی است که هرچه خدمات بهتر و غذا خوشمزه تر باشد، انعام بیشتری به پیش خدمت می دهیم. می توان سه قانون زیر را در نظر گرفت.

1. If the service is poor or the food is rancid, then tip is cheap.

۱. اگر خدمات ضعیف و غذا بی مزه باشد، انعام کم است.

2. If the service is good, then tip is average.

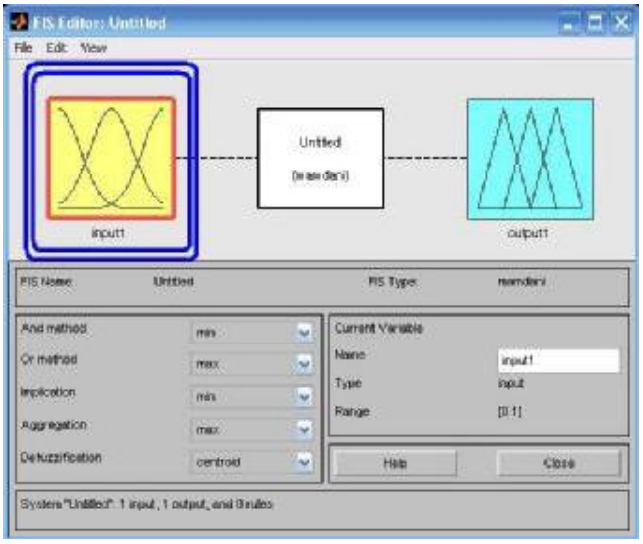
۲. اگر خدمات خوب باشد، انعام متوسط است

3. If the service is excellent or the food is delicious, then tip is generous.

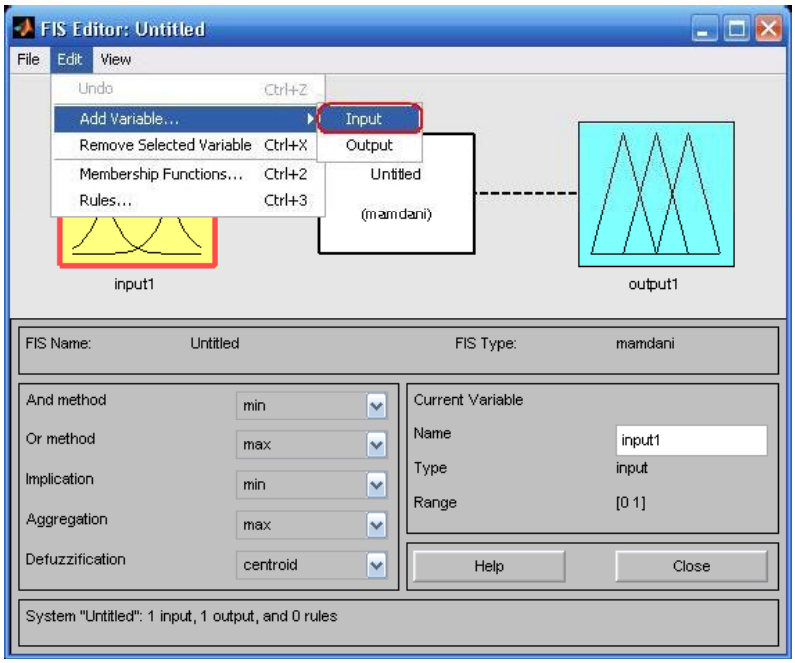
۱. اگر خدمات عالی و غذا خوشمزه باشد، انعام زیاد است.

۲-۲- افزودن متغیر ورودی و خروجی جدید:

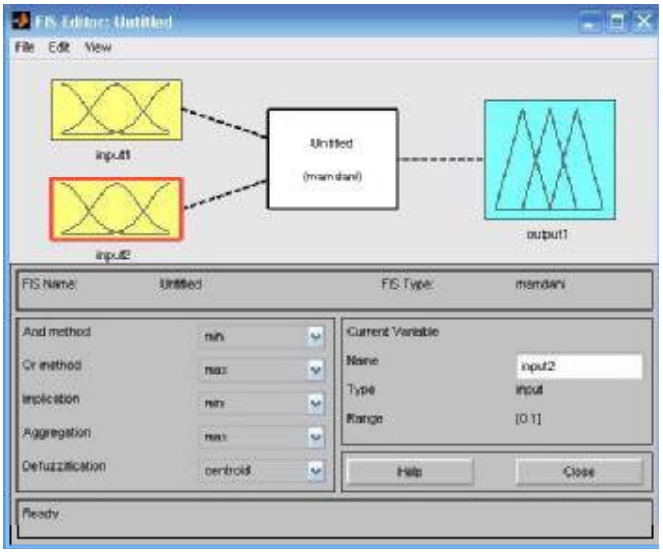
با اجرای دستور fuzzy پنجره زیر باز می شود.



همانگونه که مشاهده می شود، به صورت پیش فرض با باز شدن این پنجره، یک سیستم فازی با یک ورودی و یک خروجی نمایش داده می شود. سیستم فازی ما دارای دو ورودی و یک خروجی است. بنابراین باید یک ورودی به ان اضافه کنیم. برای انجام این کار شکل زیر را در نظر بگیرید.



با این کار متغیر جدیدی با نام input2 به سیستم فازی افزوده می شود. شکل زیر را ببینید.



۲-۳- تغییر نام متغیر ها:

برای انجام این کار گام های زیر را در نظر می گیریم.

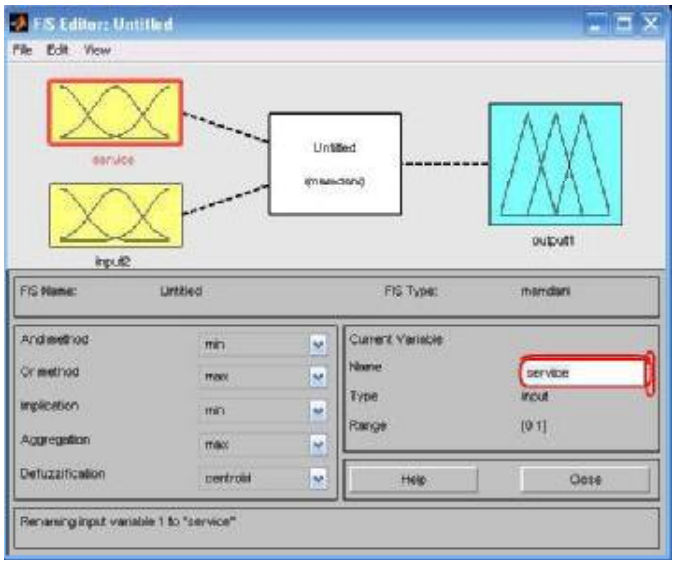
روی یک متغیر کلیلک کنید.

با این کار این متغیر با یک کادر قرمز روشن می شود.

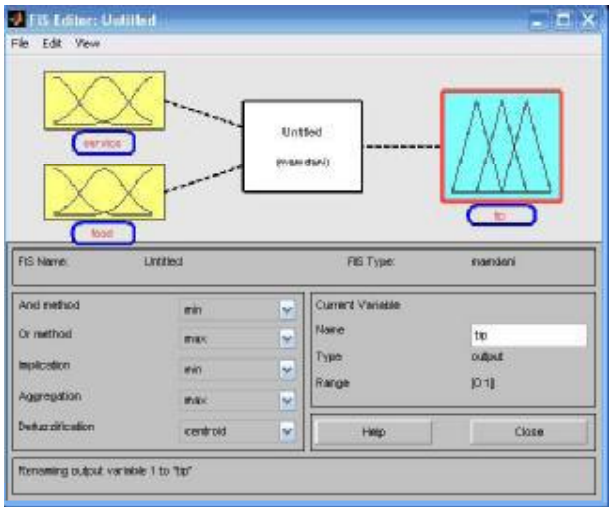
در کادر Name نام دلخواه خود را وارد کنید.

این کار را برای بقیه متغیر ها نیز تکرار کنید.

در شکل زیر چگونگی تغییر نام متغیر Input1 به service نشان داده شده است.

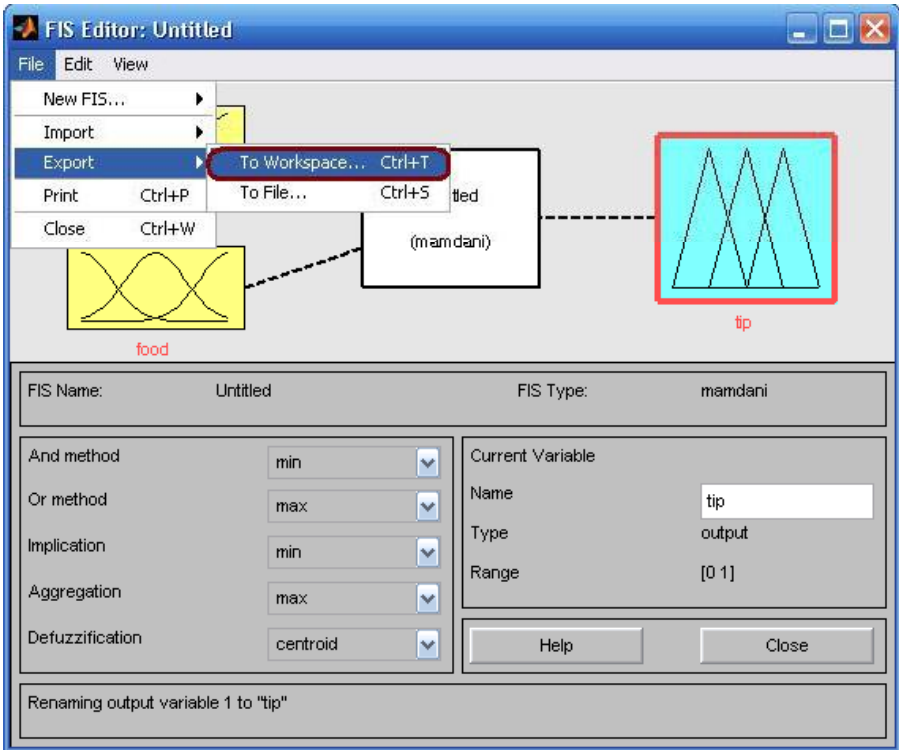


نام Input2 را به food و نام خروجی output1 را نیز به tip تغییر می دهیم. با این کار شکل فوق به صورت زیر خواهد بود.

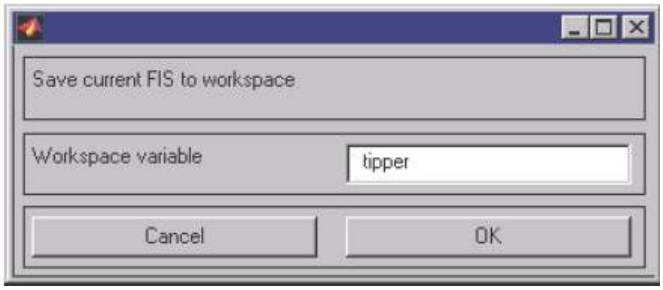


۲-۴- ذخیره سیستم فازی:

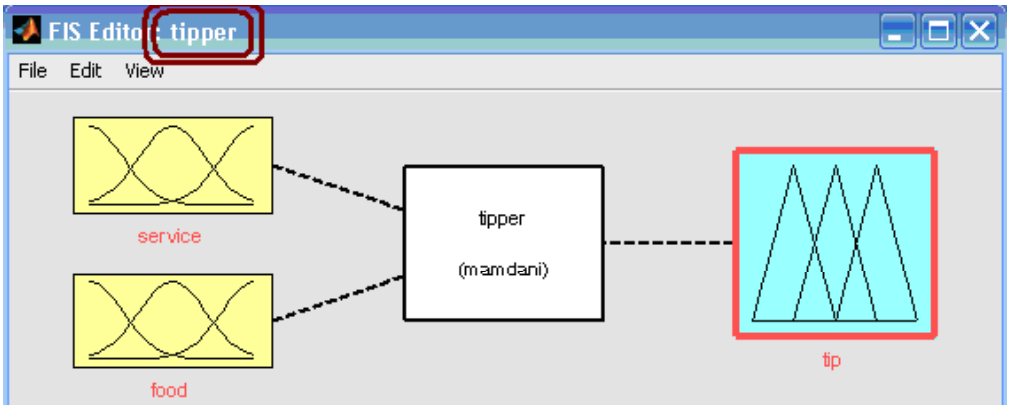
برای این کار مطابق شکل زیر عمل می کنیم.



با این کار پنجره زیر باز می شود. و در این پنجره نام دلخواه خود را وارد می کنیم. (tipper)

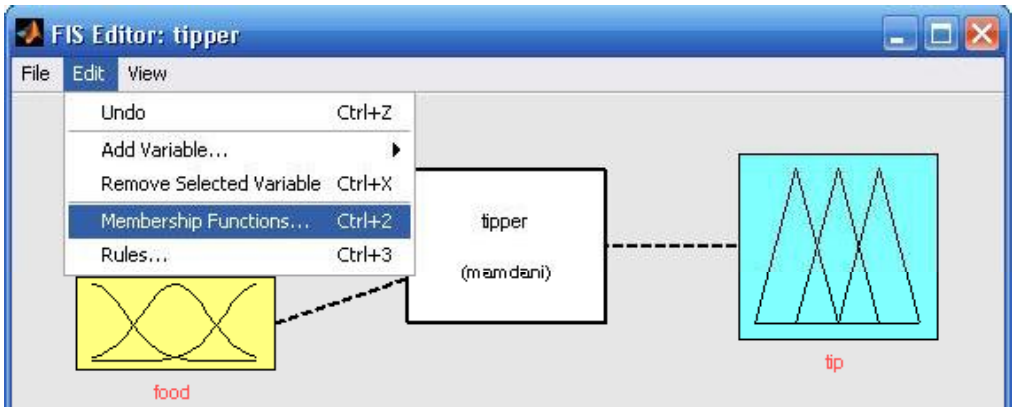


با این کار نام بالای پنجره به صورت فوق عوض می شود.

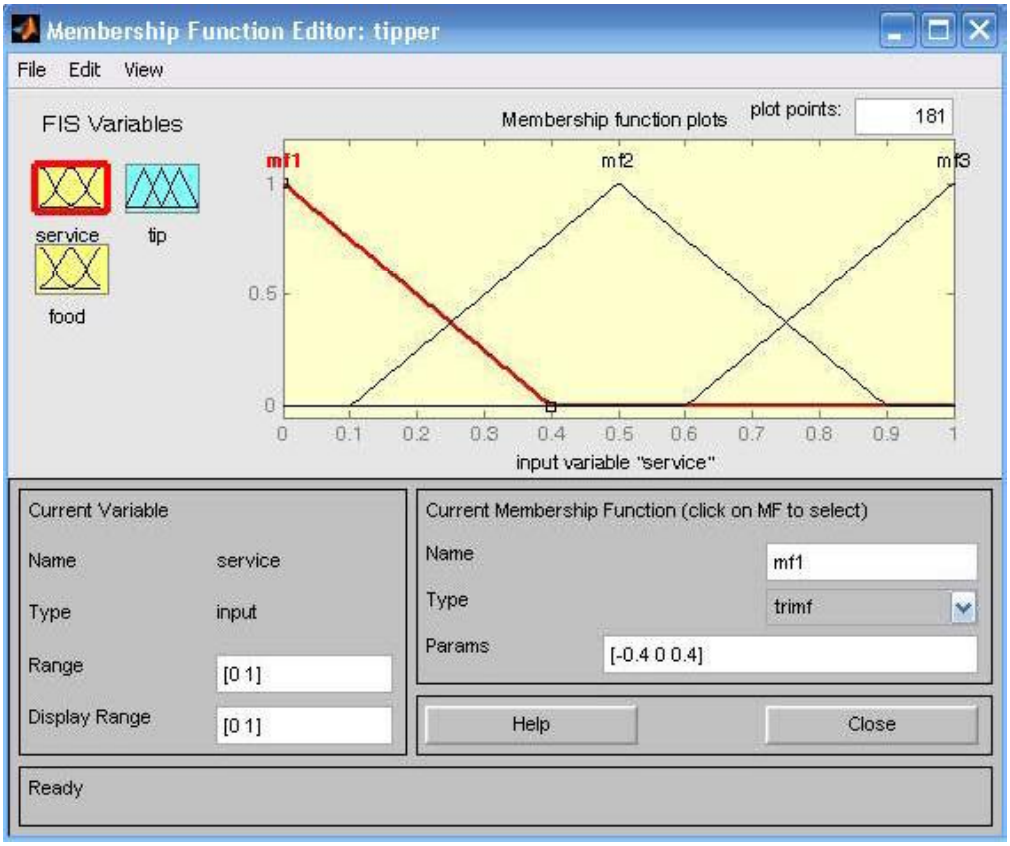


۳- ویرایشگر توابع عضویت

برای این کار روی ورودی دلخواه کلیک کرده و مطابق شکل زیر عمل می کنیم.



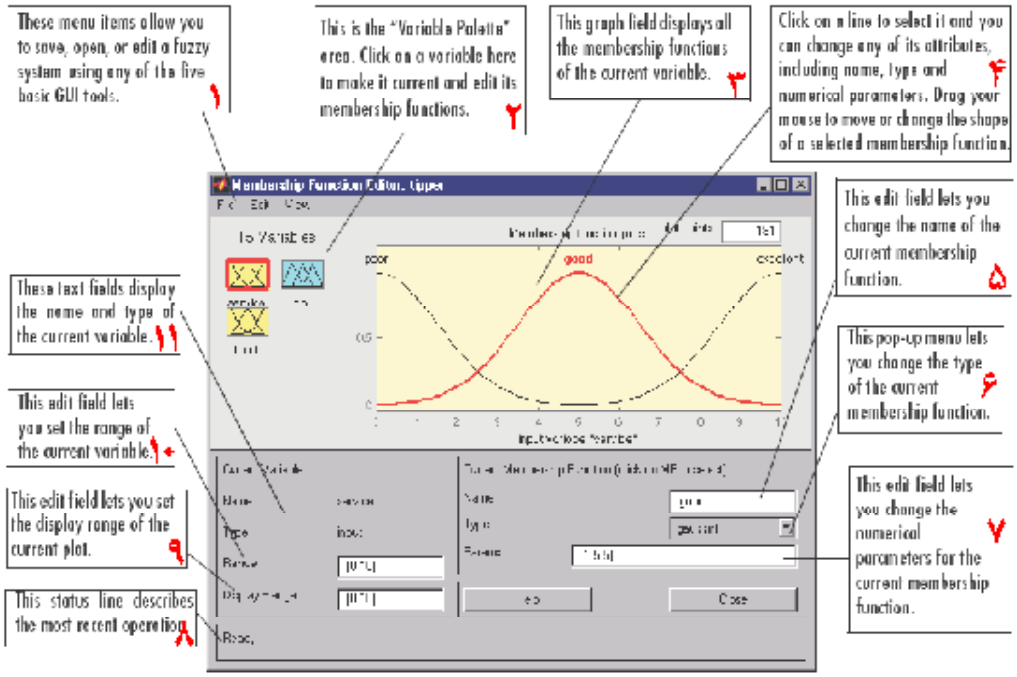
با این کار پنجره ای مطابق زیر باز می شود. در پنجره باز شده هر ورودی را به دلخواه انتخاب کـرده و تغییرات لازم را در آن می دهیم.



با کلیک کردن بر روی هر متغیر ورودی یا خروجی توابع عضویت مربوط به آن نمایش داده می شود.

در شکل بالا توابع عضویت مربوط به service نمایش داده می شود.

در شکل زیر تک تک اجزای این پنجره نمایش داده شده است.



بخش ۱: این منو به شما اجازه می دهد تا یک سیستم فازی را باز، ذخیره و یا ویرایش کنید.

بخش ۲: روی متغیر دلخواه خود کلیک کنید تا آن را انتخاب کرده و تغییرات لازم را در توابع عضویت آن بدهید.

بخش ۳: این بخش تمام توابع عضویت مربوط به متغیر انتخاب شده را نشان می دهد.

بخش ۴: روی تابع عضویت دلخواه خود کلیک کنید تا ویژگیهای آن را تغییر دهید. همچنین با استفاده از ماوس مس توانید تغییرات لازم را در شکل تابع عضویت بدهید.

بخش ۵: تغییر نام تابع عضویت.

بخش ۶: نوع تابع عضویت را می توان از این منو تغییر داد.

بخش ۷: در این بخش شما می توانید پارامترهای مربوط به توابع عضویت را تغییر دهید.

بخش ۸: آخرین تغییرات انجام شده را نشان می دهد.

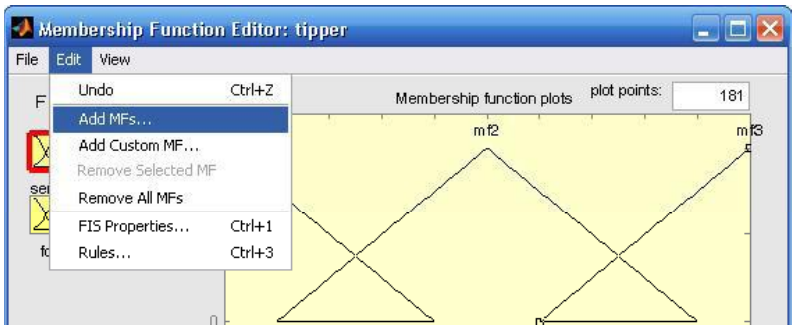
بخش ۹: بازه نمایش ترسیم جاری را نمایش می دهد.

بخش ۱۰: بازه مربوط به متغیر جاری را تنظیم می کند.

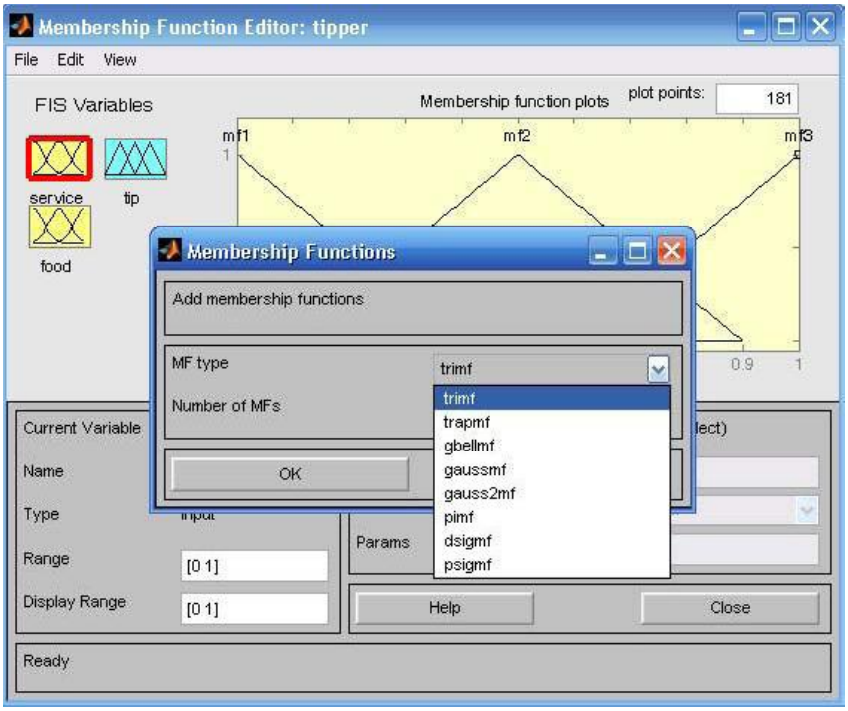
بخش ۱۱: نام و نوع متغیر را نشان می دهد.

۳-۱- *افزودن تابع عضویت جدید به یک متغیر*

برای این کار مطابق شکل زیر عمل می کنیم.

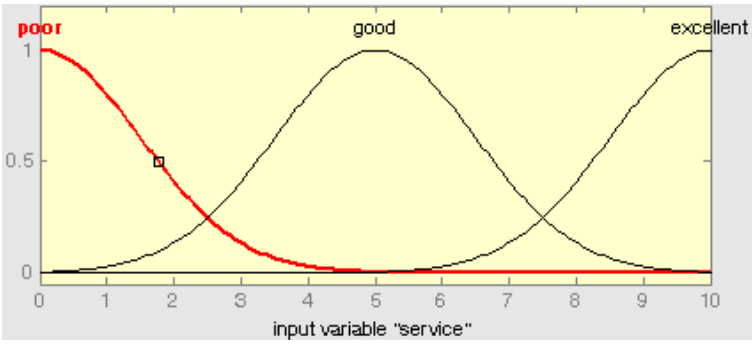


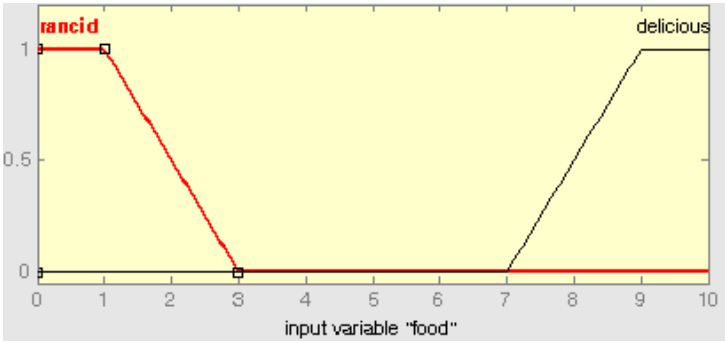
و پنجره زیر باز می شود.

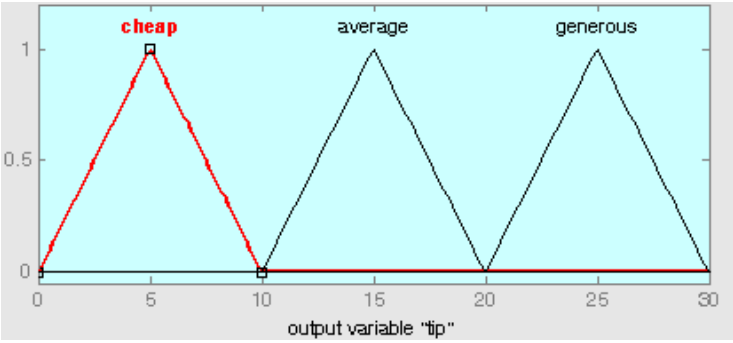


در این پنجره می توان نوع و تعداد توابع مورد نیاز برای افزودن را انتخاب کرد.

با انتخاب توابع عضویت مورد نیاز، برای سه متغیر توابع عضویت زیر را در نظر می گیریم.

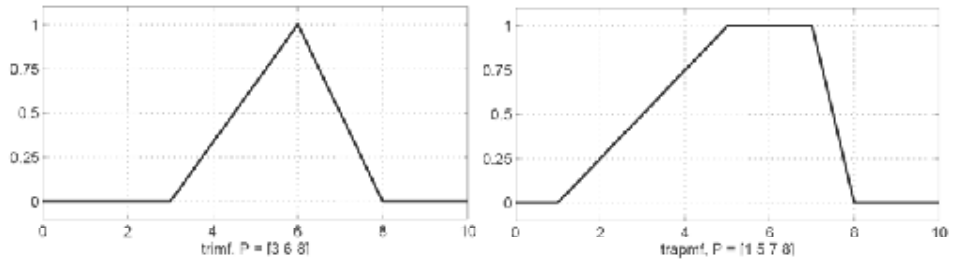




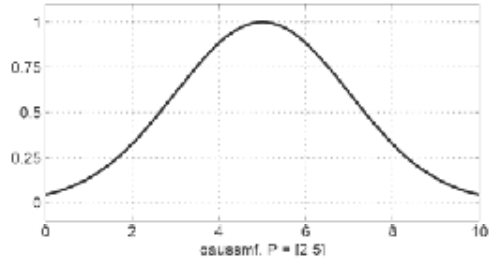


۳-۲- *توابع عضویت مورد استفاده در تولباکس متلب*

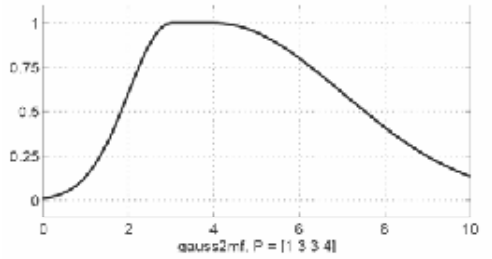
توابع عضویت مثلثی و ذوزنقه در زیر نشان داده شده اند. این توابع ساده ترین توابع مورد استفاده هستند.



تا بع عضویت گوسی ساده نیز که تنها با ۲ پارامتر میانگین و واریاس آن توصیف می شود در زیر نشان داده شده است.

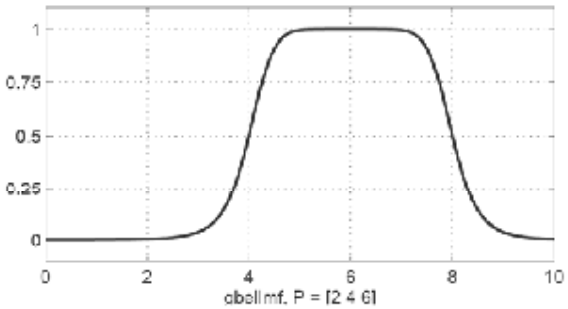


متلب از تابع عضویتی که ترکیب دو تابع عضویت گوسی است، پشتیبانی می‌کند. برای تعریف این تابع عضویت ۴ پارامتر به ترتیب برای واریانس و میانگین تابع اول و واریانس و میانگین تابع دوم نیاز است.

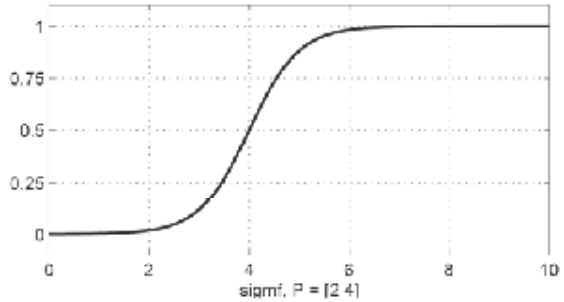


تابع زنگوله ای تعمیم یافته نیز توسط ۳ پارامتر مشخص می شود و با نام gbellmf شناخته می شود.

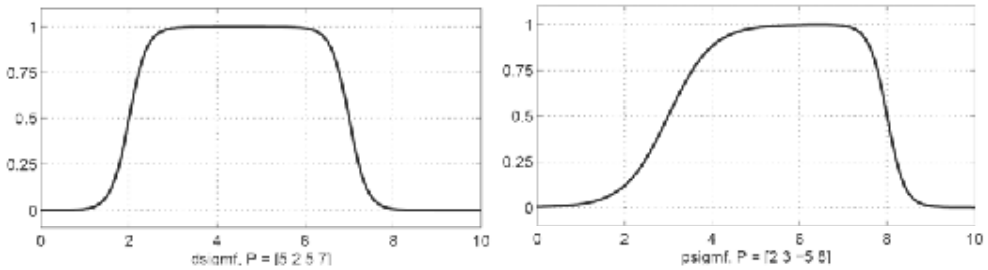
توابع عضویت گوسی و زنگوله ای مورد استفاده زیادی دارند. دلیل استفاده از آنها، نرمی آنها و نا صفر بودن در کل دامنه شان است.



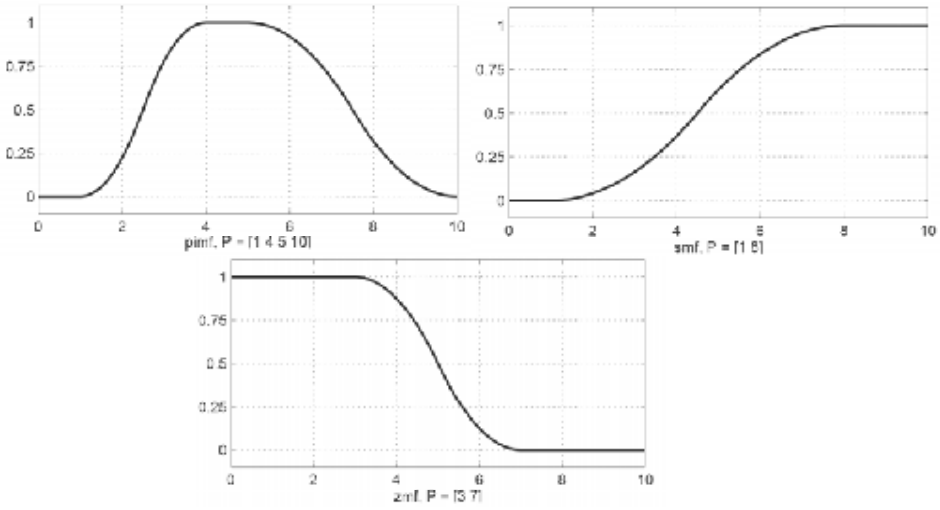
تابع عضویت سیگموئیدی نیز در متلب مورد استفاده قرار می گیرد. این تابع عضویت می تواند به سمت راست یا چپ باز باشد.



توابع عضویت dsigmf و psigmf به ترتیب حاصل تفاضل و ضرب دو تا سیگموئید هستند. و پارامتهای انها به ترتیب پارامترهای دو تا سیگموئیدی هستند که از هم تفاضل و یا کم می شوند.



توابع عضویت چند جمله ای نیز مورد استفاده زیادی دارند. Zmf یک تابع متقارن یه سمت چپ و smf یک تابع متقارن به سمت راست و pimf از دو طرف صفر و با یک پیک در وسط است.



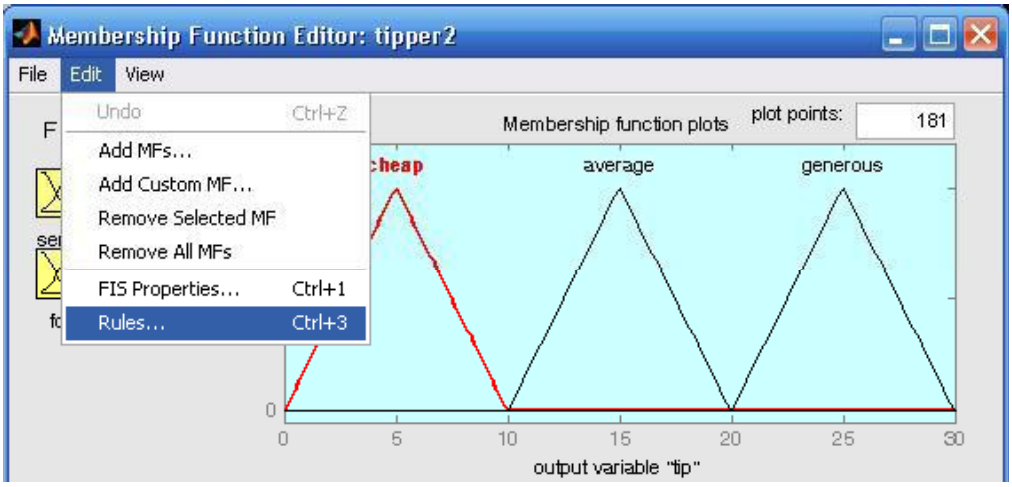
توابع عضویت در نظر گرفته شده بر حسب نیاز می توانند جایگزین همدیگر شده و معمولاً به علت پیچیدگی طراحی در سیستم های فازی، الویت خاصی را برای انها نمی توان در نظر گرفت. تنها نکته ای که در انتخاب توابع عضویت ضروری است، توجه به ویژگی های تنها است. مثلاً توابع مثلثی می توان fuzzy partitioning را ایجاد نمود (مجموع توابع عضویت در کل دامنه متغیر یک شود). در حالی که از توابع عضویتی مثل pimf بدین منظور نمی شود استفاده کرد. بعضی توابع هم دامنه محدودی دارند در حالی که بعضی در هیچ نقطه ای صفر نیستند.

حال با آماده کردن توابع عضویت و نامگذاری آنها، به پیاده سازی قوانین فازی می پردازیم.

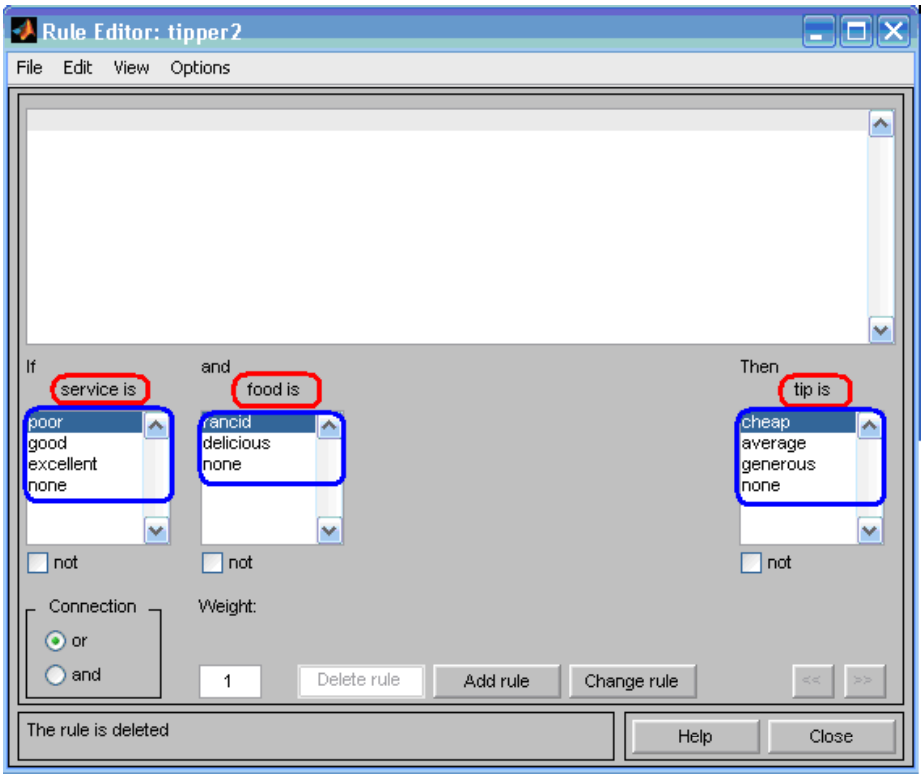
۴- پیاده سازی قوانین فازی

برای این کار مطابق شکل زیر عمل می کنیم. یا در خط فرمان متلب دستور زیر را اجرا می کنیم.

```
>> ruleedit
```



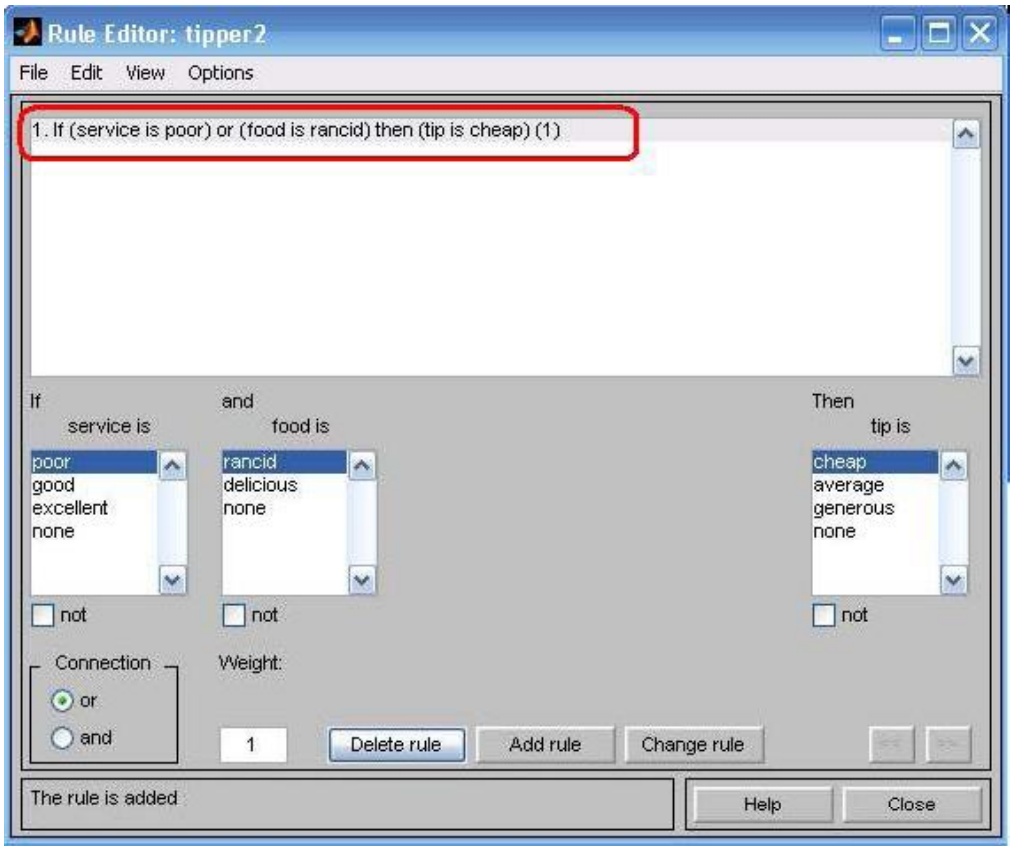
با این کار پنجره زیر باز می شود.



فرض کنید می خواهیم قانون اول یعنی قانون زیر را پیاده سازی کنیم.

1. If the service is poor or the food is rancid, then tip is cheap.

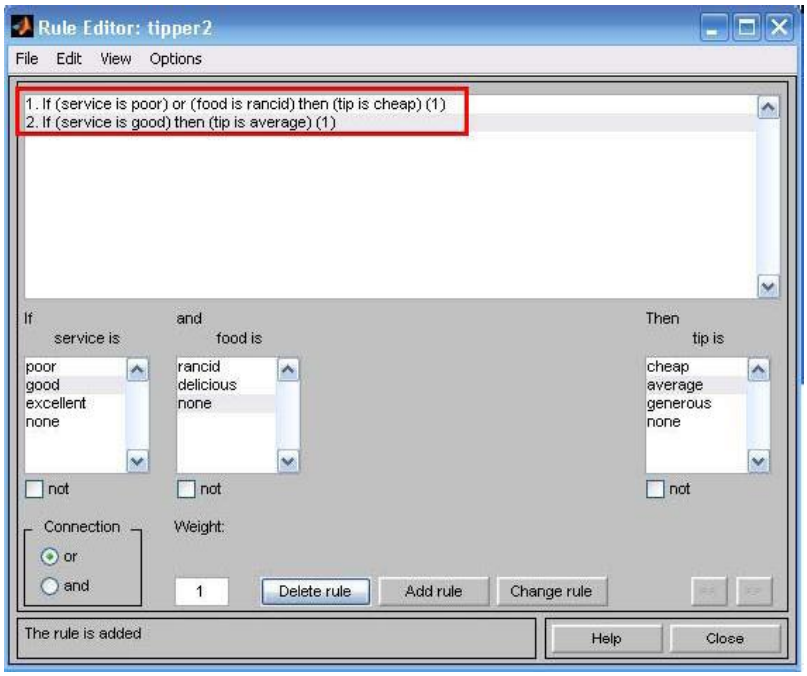
برای این کار با تنظیمات به صورت فوق دکمه Add Rule را کلیک می کنیم. با این کـار در بخـش بالای پنجره این قانون ظاهر خواهد شد.



برای قانون دوم نیز خواهیم داشت.

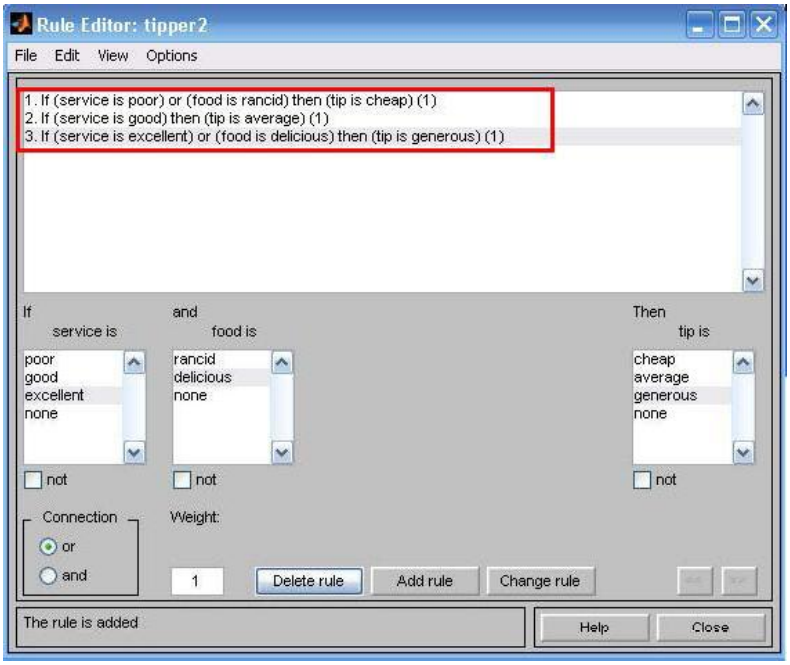
2. If the service is good, then tip is average.

برای این کار با انتخاب گزینه ها به صورت زیر دکمیه Add rule را کلیک می کنیم. با این کار قانون مربوط به ان نیز ظاهر می شود.



در نهایت با پیاده سازی قانون سوم نیز خواهیم داشت.

3. If the service is excellent or the food is delicious, then tip is generous.

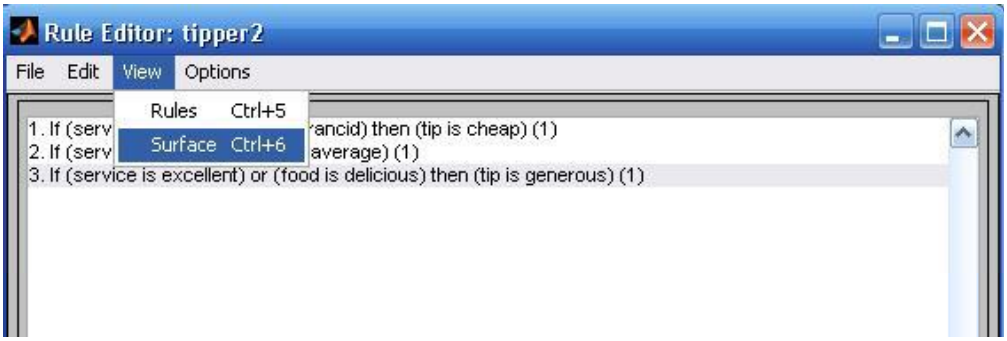


بدین ترتیب قوانین سیستم نیز پیاده سازی شده و سیستم بطور کامل اماده است.

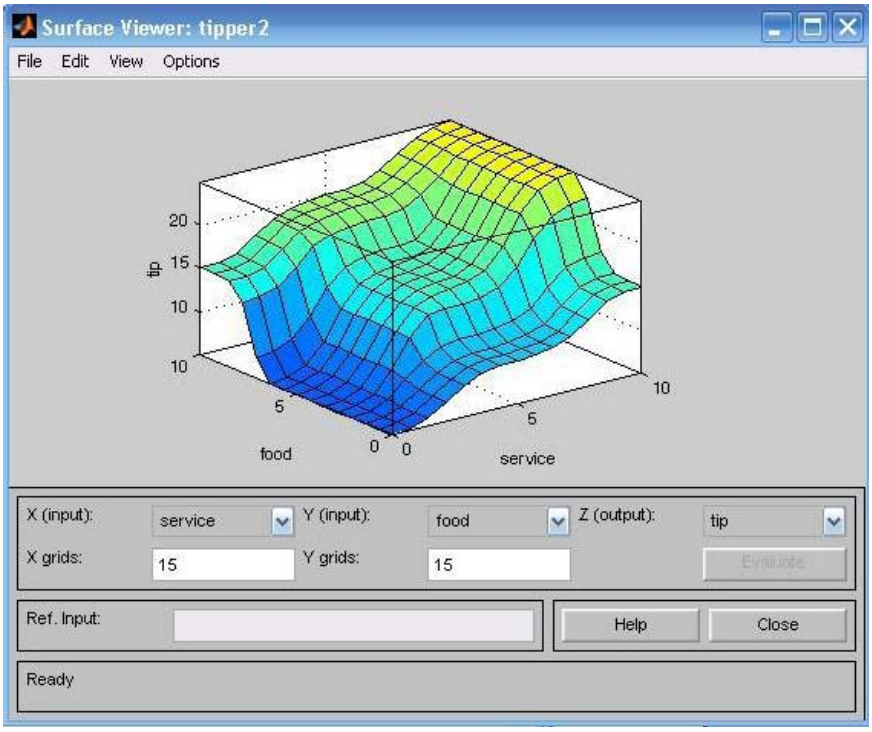
حال به آزمایش و تست سیستم می پردازیم. یکی از راههای ارزیابی سیستم فـازی مـشاهده Rule surface آن است.

۵– مشاهده Rule Surface

برای این کار مطالبق شکل زیر عمل می کنیم.



با این کار پنجره ای مطابق شکل زیر باز می شود.

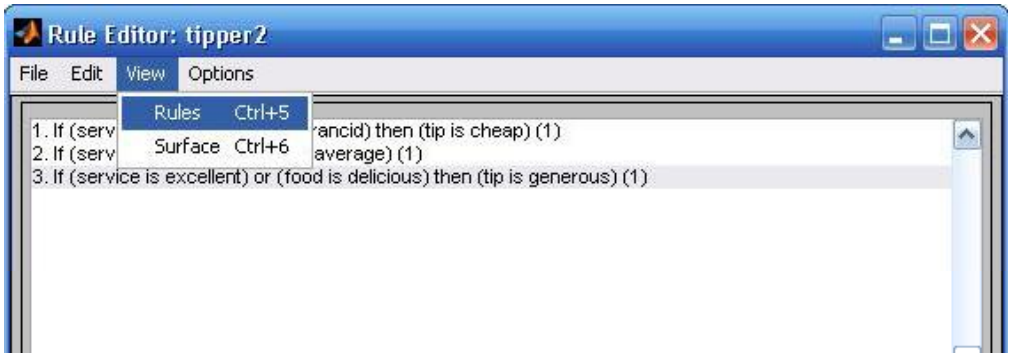


در این پنجره ارتباط فازی ایجاد شده میان خروجی و ورودی های نشان داده میشود. با تغییر بعضی از گزینه های موجود می توان شیوه نمایش را تغییر داد.

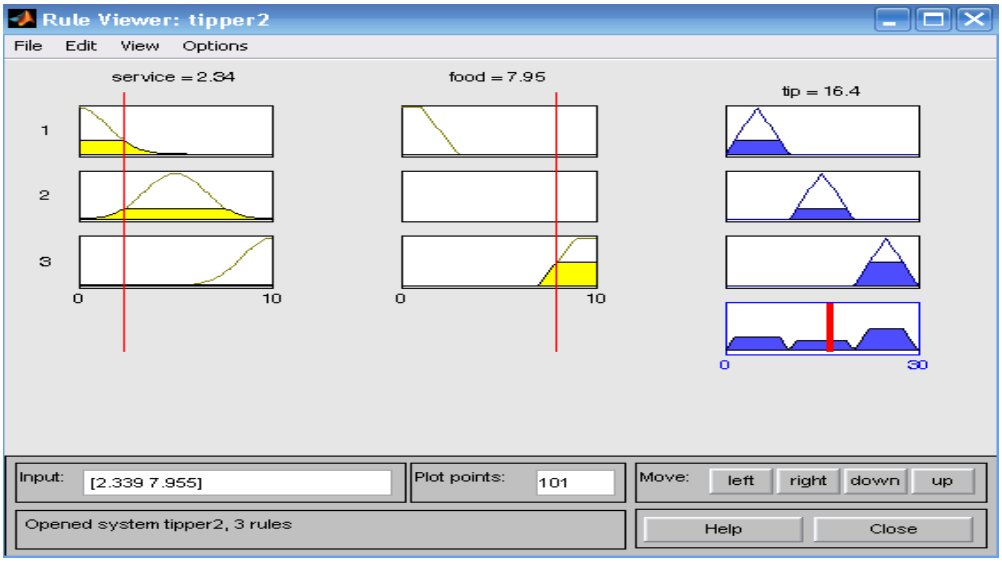
یکی از راه های دیگر ارزیابی و تست سیستم فازی در تولباکی متلب استفاده از نشانن دهنده قوانین است.

۶- مشاهده قوانین

برای این کار مطالبق شکل زیر عمل می کنیم.

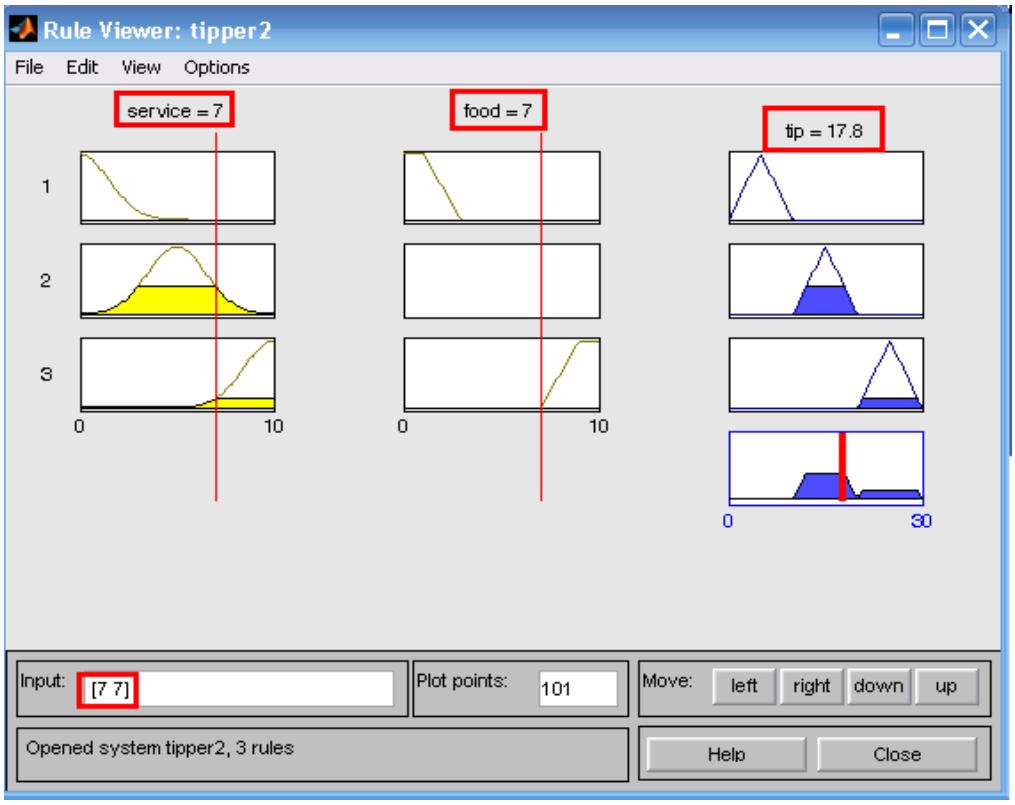


با این کار پنجره ای مطابق شکل زیر باز می شود.



در این پنجره با تغییر میله های عمودی میزان هر یک از ورودی های فازی را می توان تعیین کرده و خروجی را مشاهده نمود. به عنوان مثال اگر بخواهیم میزان انعام را برای خدمات و غذای زیر بیابیم خواهیم داشت:

Service = 7 , food = 7 => tip = ?



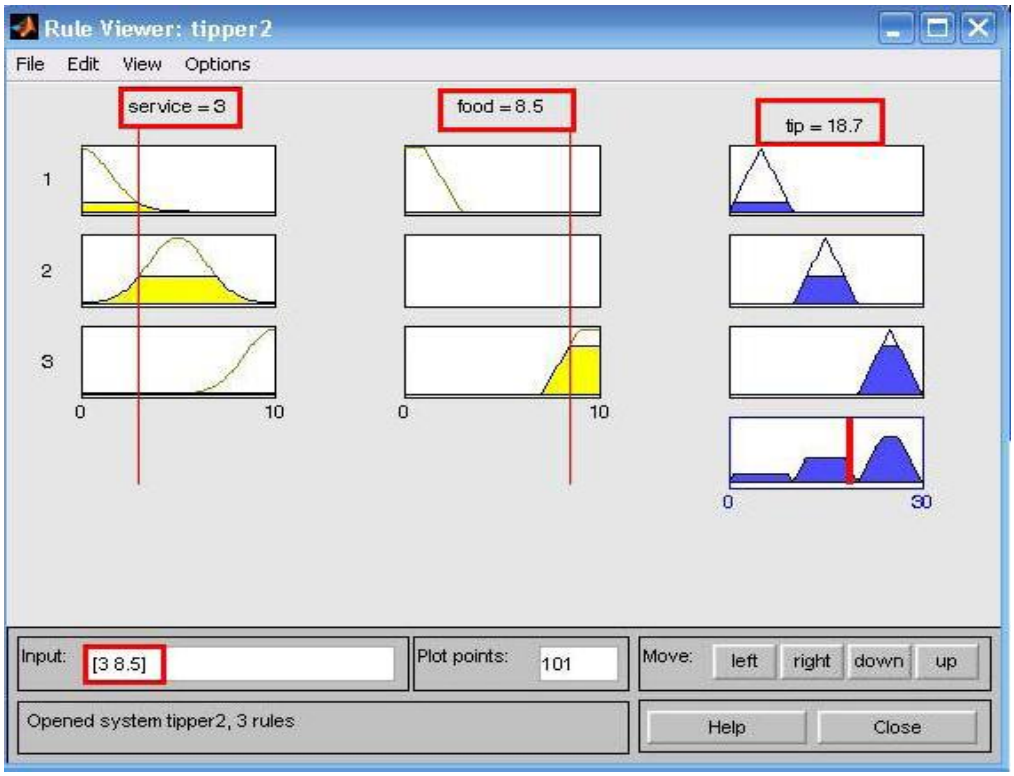
پس خواهیم داشت:

Tip = 17.8

به عنوان یک مثال دیگر اگر داشته باشیم:

Service = 3 , food = 8.5

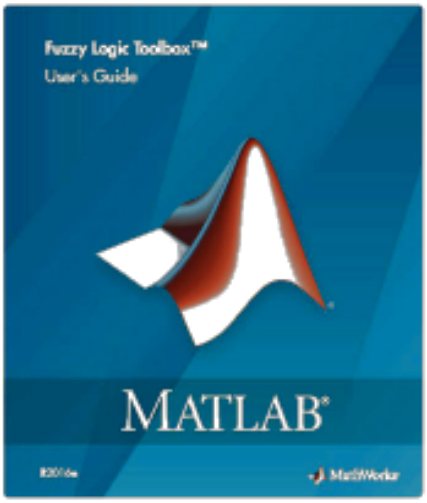
در این صورت tip چقدر خواهد بود؟ خواهیم داشت.



بنابراین tip برابر خواهد بود با

Tip = 18.7

پایان



@Matlabonline
کانال تخصصی آموزش Matlab
کلیه تولباکس ها و کلیه رشته ها
مقدماتی تا پیشرفته
کد رایگان همراه هزاران مثال حل شده
2016
دانشگاه تهران