



Expert System

Fox Fuzzy Logics

شبیه سازی حرکت روباه و خرگوش

رشته: مهندسی نرم افزار
شماره دانشجویی: 38672114

آقای دکتر فیض درخشی
آقای بهزاد خسروی فر

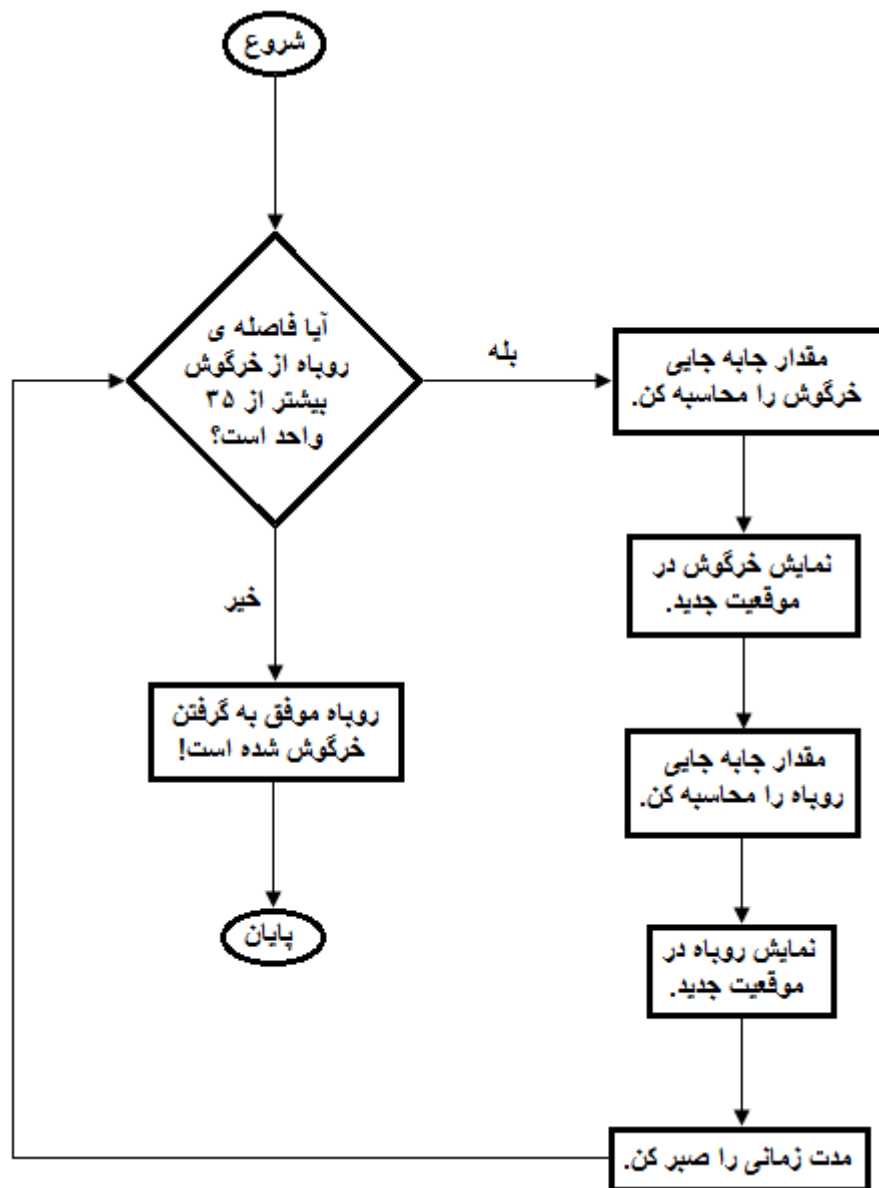
استاد مربوطه:
نویسنده:

1389/10/06

تعریف مسئله: محیطی را طراحی کنید که در آن بصورت گرافیکی حرکت دنبال کردن خرگوش توسط روباه شبیه سازی شود و در آن فقط سیستم کنترل روباه با استفاده از سیستم فازی پیاده سازی شود.

حل مسئله: برای شبیه سازی حرکات دنبال کردن خرگوش توسط روباه با استفاده از سیستم فازی از محیط برنامه نویسی C#.Net 3.5 استفاده کرده ایم. بطوری که هر دو موجودیت کاملاً گرافیکی می باشند.

الگوریتم کلی برنامه بصورت زیر است:



روند برنامه به زبان C#.Net :

```
//
// Keep the commands until the distance of the fox rabbit is
// more than 35
//
while (Distance(Fox.Location, Rabbit.Location)) > 35)
{
    //
    // Move Rabbit Code
    //
    moveRabbit(pRabbit.X, pRabbit.Y, pFox.X, pFox.Y,
               out DXR, out DYR);
    pRabbit.X += (int)DXR;
    pRabbit.Y += (int)DYR;
    show(Rabbit, pRabbit);
    //
    // Move Fox Code
    //
    moveFox(pRabbit.X, pRabbit.Y, pFox.X, pFox.Y,
            out DXF, out DYF);
    pFox.X += (int)DXF;
    pFox.Y += (int)DYF;
    show(Fox, pFox);
    //
    // this thread waited a few milliseconds.
    //
    Thread.CurrentThread.Join(50);
}
```

در کدهای بالا داریم :

DXR = مقدار جابه جایی خرگوش بر روی محور X

DYR = مقدار جابه جایی خرگوش بر روی محور Y

DXF = مقدار جابه جایی روباه بر روی محور X

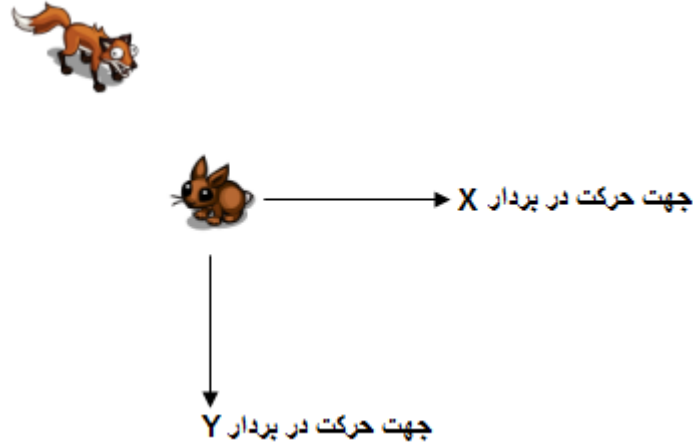
DYF = مقدار جابه جایی روباه بر روی محور Y

pFox = موقعیت روباه در صفحه

pRabbit = موقعیت خرگوش در صفحه

❖ حرکت خرگوش: `moveRabbit(...)`

حرکت خرگوش در این شبیه سازی بصورت تصادفی رخ می دهد ولی به سمتی حرکت می کند که به روباه نزدیک نشود. برای مثال یک نمونه حرکت خرگوش بصورت تصادفی و گریز از روباه شبیه سازی شده است:



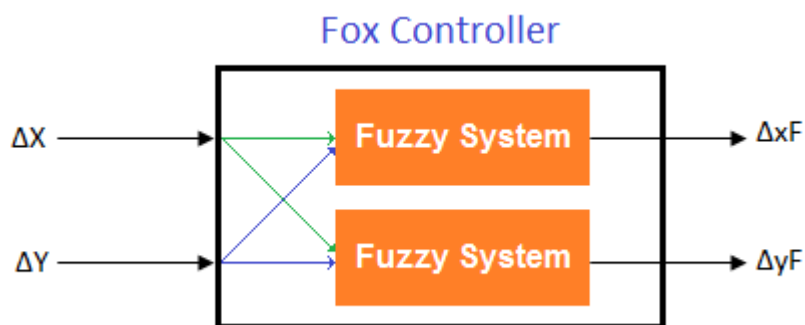
همان طور که مشاهده می شود هر دو حرکت محورهای مختصاتی موجب دور شدن خرگوش میشوند.

نکته: در برنامه ی پیاده سازی شده می توان بوسیله ی ماوس بر روی هر دو موجودیت کلیک کرده و تا زمان نگه داشتن کلید ماوس بصورت دستی و بدون وقفه عمل جابه جایی را انجام داد.

❖ حرکت روباه: `moveFox(...)`

بعد از حرکت خرگوش نوبت شبیه سازی حرکت روباه برای گرفتن خرگوش می باشد، که برای این عمل از سیستم فازی استفاده شده است.

برای ایجاد یک کنترلر که ورودی آن موقعیت روباه نسبت به خرگوش باشد و خروجی آن مقدار جابه جایی روباه برای موقعیت جدید باشد نیازمند دو سیستم فازی مشابه با ورودی های یکسان هستیم.



در کنترلر بالا مفاهیم زیر را داریم:

$\Delta X = x_{\text{Rabbit}} - x_{\text{Fox}}$ موقعیت نسبی روباه در راستای محور X

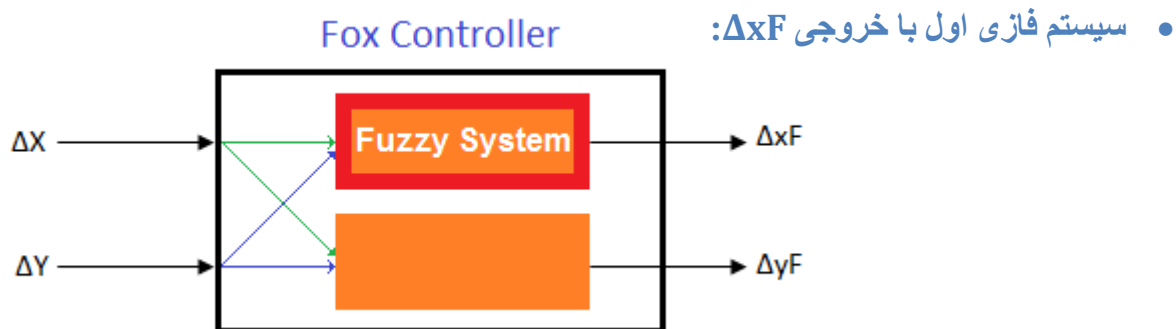
$\Delta Y = y_{\text{Rabbit}} - y_{\text{Fox}}$ موقعیت نسبی روباه در راستای محور Y

نکته: در محیط C# جهت مثبت محور Y بطرف پایین می باشد بنابراین در برنامه $\Delta Y = y_{\text{Fox}} - y_{\text{Rabbit}}$ می باشد.

Δx_F مقدار جابه جایی روباه در راستای محور X

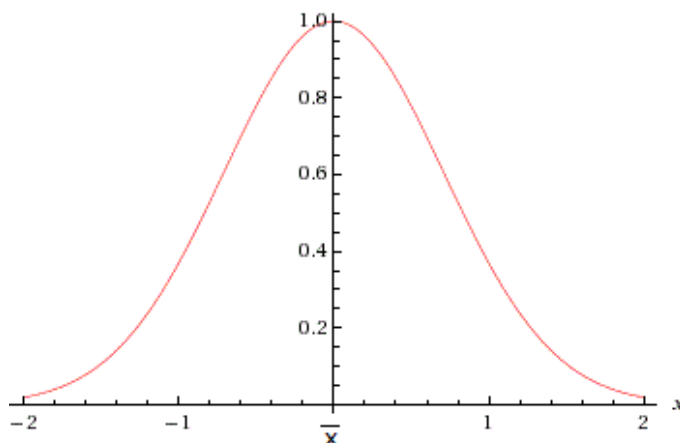
Δy_F مقدار جابه جایی روباه در راستای محور Y

ابتدا یک سیستم فازی از کنترلر روباه انتخاب می کنیم و اجزا آن را شرح می دهیم:



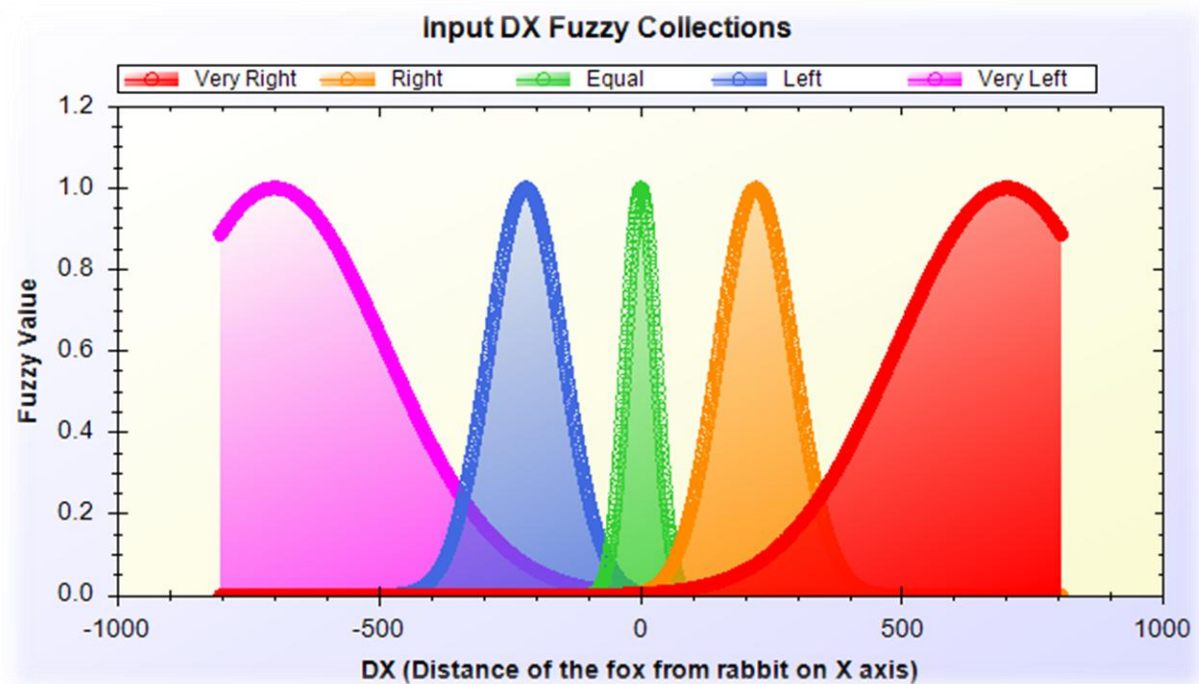
در این سیستم فازی دو ورودی و یک خروجی داریم بنابراین باید دو مجموعه فازی برای ورودیها و یک مجموعه فازی برای خروجی در نظر گرفت.

بخاطر پیوستگی حرکات روباه و عدم شکسته بودن تغییرات حرکت، از منحنی های گوسین استفاده کرده ایم:



فرمول گوسین : $e^{-\left(\frac{x-\bar{x}}{\sigma}\right)^2}$

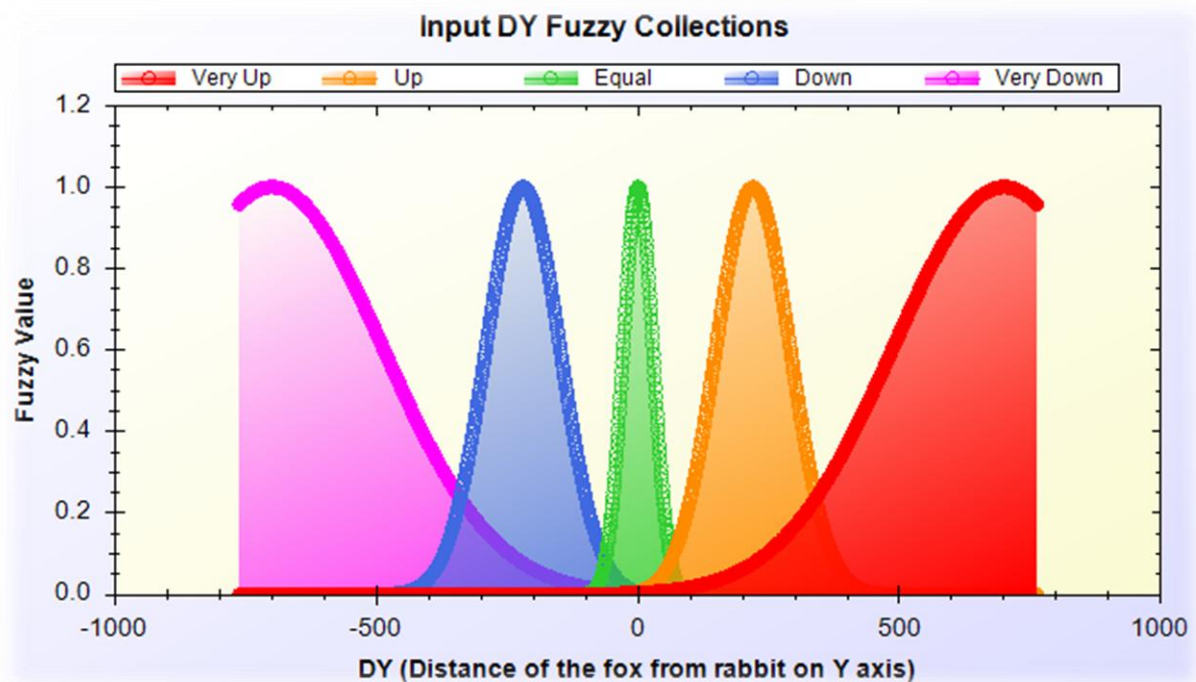
مجموعه فازی ورودی برای موقعیت نسبی روباه در راستای محور X :
همان طور که می دانیم در محور X نوع حرکت به سمت چپ و یا راست می باشد که آنگاه داریم:



نام منحنی گوسین	نام منحنی گوسین	واریانس σ	میانگین \bar{x}
VR	Very Right	300	700
R	Right	100	220
E	Equal	40	0
L	Left	100	-220
VL	Very Left	300	-700

هر نمودار در مجموعه دارای فرمول متفاوتی می باشد و این نمودارها برای صفحه نمایش هایی با ابعاد مختلف ، کارایی متفاوتی دارند. در برنامه حالت پیش فرض را 800×800 در نظر گرفته ایم. هرچند که با تغییر ابعاد صفحه نمایش ، برنامه خود را سازگار نگه می دارد.

مجموعه فازی ورودی برای موقعیت نسبی روباه در راستای محور y :
همان طور که می دانیم در محور y نوع حرکت به سمت بالا و یا پایین می باشد که آنگاه داریم:



نام منحنی گوسین	نام منحنی گوسین	واریانس σ	میانگین \bar{x}
Very Up	U	300	700
Up	E	100	220
Equal	D	40	0
Down	VD	100	-220
Very Down		300	-700

با در دست داشتن مجموعه های فازی ورودی سیستم می توان با در نظر داشتن خبره ی این سیستم (روباه) مجموعه قوانین را بدست آوریم و از روی این مجموعه قوانین که پایگاه دانش سیستم را تشکیل می دهند مجموعه فازی خروجی ΔxF را هم بدست آوریم.

با توجه به الگوهای خاص حرکتی روباه ، به مجموعه قوانین زیر رسیدیم:

1. if ΔX is VL and ΔY is VD ----> ΔXF is HL
2. if ΔX is L and ΔY is VD ----> ΔXF is LL
3. if ΔX is E and ΔY is VD ----> ΔXF is E
4. if ΔX is R and ΔY is VD ----> ΔXF is LR
5. if ΔX is VR and ΔY is VD ----> ΔXF is HR
6. if ΔX is VL and ΔY is D ----> ΔXF is HL
7. if ΔX is L and ΔY is D ----> ΔXF is LL
8. if ΔX is E and ΔY is D ----> ΔXF is E
9. if ΔX is R and ΔY is D ----> ΔXF is LR
10. if ΔX is VR and ΔY is D ----> ΔXF is HR

11. if ΔX is VL and ΔY is E $\rightarrow \Delta XF$ is HL
12. if ΔX is L and ΔY is E $\rightarrow \Delta XF$ is LL
13. if ΔX is E and ΔY is E $\rightarrow \Delta XF$ is E
14. if ΔX is R and ΔY is E $\rightarrow \Delta XF$ is LR
15. if ΔX is VR and ΔY is E $\rightarrow \Delta XF$ is HR
16. if ΔX is VL and ΔY is U $\rightarrow \Delta XF$ is HL
17. if ΔX is L and ΔY is U $\rightarrow \Delta XF$ is LL
18. if ΔX is E and ΔY is U $\rightarrow \Delta XF$ is E
19. if ΔX is R and ΔY is U $\rightarrow \Delta XF$ is LR
20. if ΔX is VR and ΔY is U $\rightarrow \Delta XF$ is HR
21. if ΔX is VL and ΔY is VU $\rightarrow \Delta XF$ is HL
22. if ΔX is L and ΔY is VU $\rightarrow \Delta XF$ is LL
23. if ΔX is E and ΔY is VU $\rightarrow \Delta XF$ is E
24. if ΔX is R and ΔY is VU $\rightarrow \Delta XF$ is LR
25. if ΔX is VR and ΔY is VU $\rightarrow \Delta XF$ is HR

با توجه به قوانین فوق می توان ماتریس قوانین 5×5 زیر را ساخت:

VR	L	E	R	VR	
HL	LL	E	LR	HR	VU
HL	LL	E	LR	HR	U
HL	LL	E	LR	HR	E
HL	LL	E	LR	HR	D
HL	LL	E	LR	HR	VD

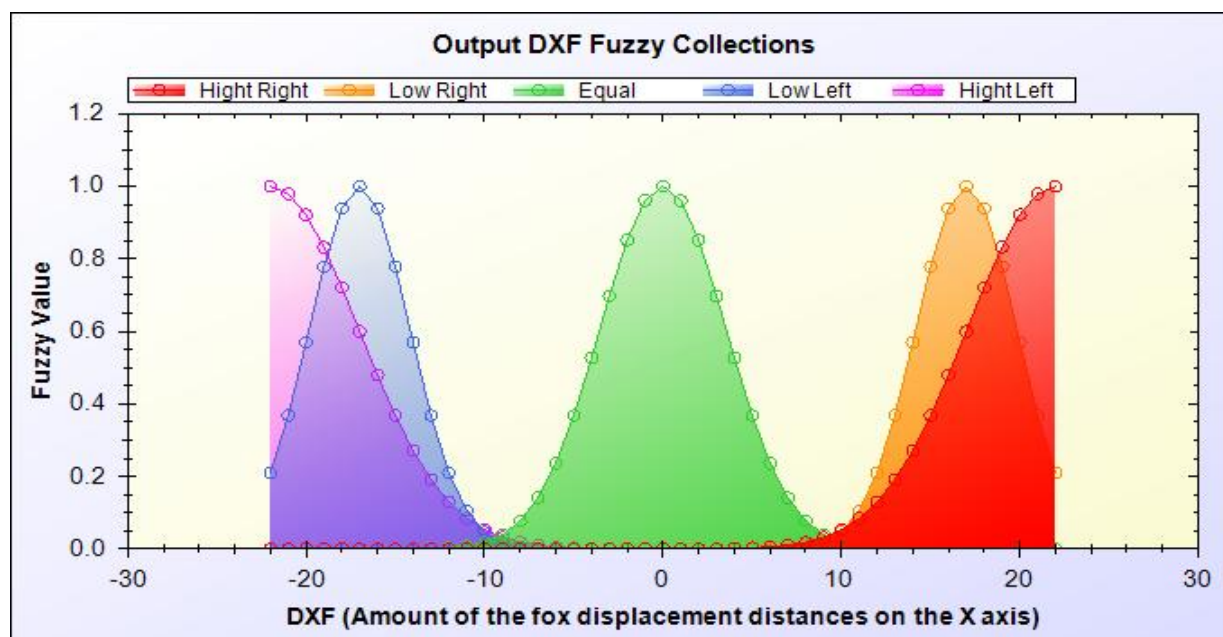
حال نوبت به ماتریس میانگین مراکز \bar{Y} رسیده است که مرکز (میانگین) منحنی های قوانین را به جای خود قوانین در ماتریس ذخیره می کند.

VR	L	E	R	VR	
-22	-17	0	17	22	VU
-22	-17	0	17	22	U
-22	-17	0	17	22	E
-22	-17	0	17	22	D
-22	-17	0	17	22	VD

نکته: سرعت خرگوش 20 ثابت و حداکثر سرعت روباه در جهت محور x برابر با 22 در نظر گرفته شده است.



از روی ماتریس قوانین و میانگین مراکز \bar{Y} می توانیم مجموعه فازی خروجی ، برای سرعت روباه بر روی محور X را تعیین کنیم: (حداکثر سرعت روباه 22 در نظر گرفته شده است)



نام منحنی گوسین	نام منحنی گوسین	واریانس σ	میانگین \bar{x}
HR	High Right	7	22
LR	Low Right	4	17
E	Equal	5	0
LL	Low Left	4	-17
HL	High Left	7	-22

حال باید موتور استنتاج سیستم فازی مان را مشخص کنیم.
در این سیستم از موتور استنتاج ضرب و غیر فازی ساز میانگین مراکز با مرکز \bar{Y} و فازی ساز منفرد استفاده شده است. فرمول کلی این موتور استنتاج به شکل زیر می باشد:

$$f(x) = \frac{\sum_{l=1}^M \bar{y}^l (\prod_{i=1}^n \mu_{A_i^l}(x_i))}{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i))} = \frac{\sum (\bar{y} * Rulefire)}{\sum Rulefire}$$

در فرمول بالا ماتریس Rulefire از ضرب دو ماتریس که یکی $MDX_{1 \times 5}$ برای ذخیره مقادیر فازی ورودی Δx و دیگری $MDY_{5 \times 1}$ برای ذخیره مقادیر فازی ورودی Δy می باشد، بدست می آید.

```
// MDX:
//
//      | 0 | 1 | 2 | 3 | 4 |
//      |---|---|---|---|---|
//      | μ (DX) | μ (DX) | μ (DX) | μ (DX) | μ (DX) |
//      | VL | L | E | R | VR |
```

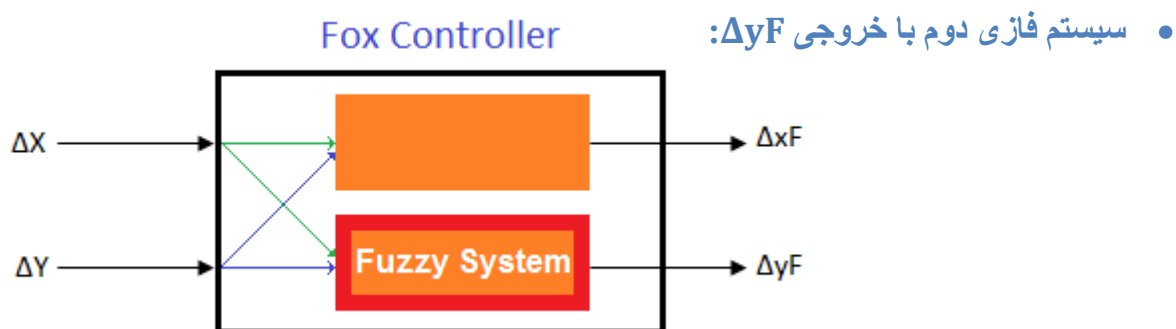
```

//
// MDY:
//      |   |
//      | 0 |  $\mu(DY)vu$ 
//      |   |
//      | 1 |  $\mu(DY)u$ 
//      |   |
//      | 2 |  $\mu(DY)e$ 
//      |   |
//      | 3 |  $\mu(DY)d$ 
//      |   |
//      | 4 |  $\mu(DY)vd$ 
//      |   |
//
//
// RuleFire  $_{5 \times 5} = MDY_{5 \times 1} \times MDX_{1 \times 5}$ 
//

```

نکته: در فرمول موتور استنتاج منظور از $\bar{y}.* Rulefire$ همان ضرب درایه به درایه (ضرب داخلی) دو ماتریس میانگین مراکز \bar{y} در ماتریس Rulefire می باشد.

در هر بار اجرای برنامه و در هر گام از حرکت روباه مقادیر ماتریس های MDY , MDX با توجه به Δx و Δy متفاوت جاری مقداردهی شده و ضرب ماتریس ها از سر گرفته می شود. در نتیجه جواب خروجی موتور استنتاج بر حسب ورودی تغییر خواهد یافت.



در این سیستم مجموعه های فازی ورودی هیچ تفاوتی با سیستم اول ندارد زیرا ورودی هر دو سیستم برابر می باشد.

بنابراین با در دست داشتن مجموعه های فازی ورودی سیستم می توان با در نظر داشتن خبره ی این سیستم (روباه) مجموعه قوانین را برای این سیستم نیز بدست آوریم و از روی این مجموعه قوانین که پایگاه دانش سیستم را تشکیل می دهند مجموعه فازی خروجی Δy_F را هم بدست آوریم.

مجموعه قوانین در این سیستم عبارتند از:

1. if ΔX is VL and ΔY is VD ----> ΔXF is HD
2. if ΔX is L and ΔY is VD ----> ΔXF is HD
3. if ΔX is E and ΔY is VD ----> ΔXF is HD
4. if ΔX is R and ΔY is VD ----> ΔXF is HD
5. if ΔX is VR and ΔY is VD ----> ΔXF is HD
6. if ΔX is VL and ΔY is D ----> ΔXF is LD
7. if ΔX is L and ΔY is D ----> ΔXF is LD
8. if ΔX is E and ΔY is D ----> ΔXF is LD
9. if ΔX is R and ΔY is D ----> ΔXF is LD
10. if ΔX is VR and ΔY is D ----> ΔXF is LD
11. if ΔX is VL and ΔY is E ----> ΔXF is E
12. if ΔX is L and ΔY is E ----> ΔXF is E
13. if ΔX is E and ΔY is E ----> ΔXF is E
14. if ΔX is R and ΔY is E ----> ΔXF is E
15. if ΔX is VR and ΔY is E ----> ΔXF is E
16. if ΔX is VL and ΔY is U ----> ΔXF is LU
17. if ΔX is L and ΔY is U ----> ΔXF is LU
18. if ΔX is E and ΔY is U ----> ΔXF is LU
19. if ΔX is R and ΔY is U ----> ΔXF is LU
20. if ΔX is VR and ΔY is U ----> ΔXF is LU
21. if ΔX is VL and ΔY is VU ----> ΔXF is HU
22. if ΔX is L and ΔY is VU ----> ΔXF is HU
23. if ΔX is E and ΔY is VU ----> ΔXF is HU
24. if ΔX is R and ΔY is VU ----> ΔXF is HU
25. if ΔX is VR and ΔY is VU ----> ΔXF is HU

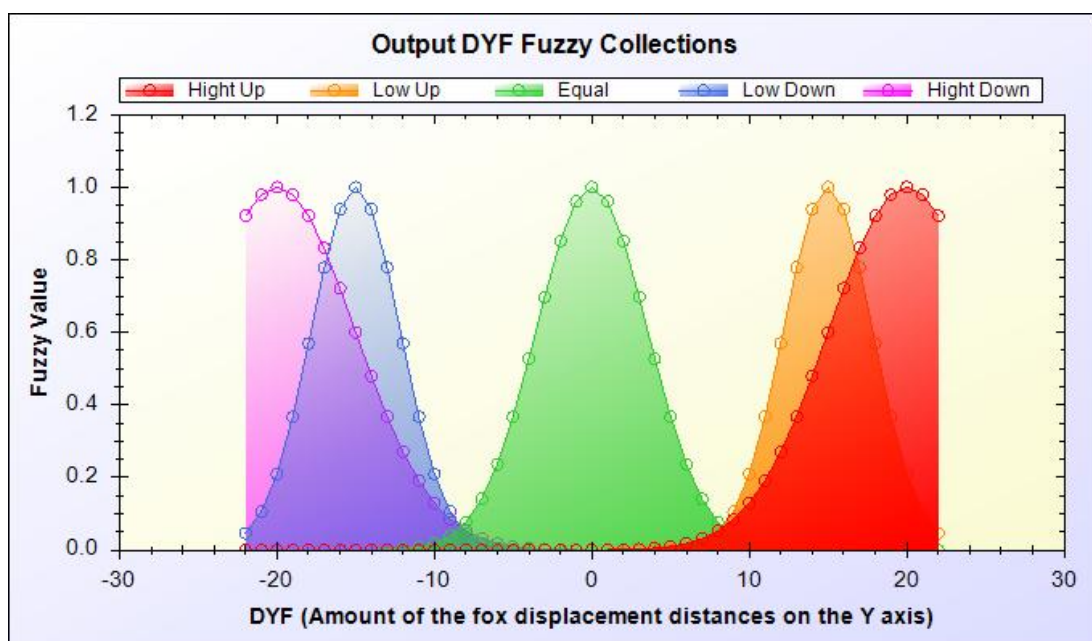
با توجه به قوانین فوق می توان ماتریس قوانین 5×5 زیر را ساخت:

VR	L	E	R	VR	
HU	HU	HU	HU	HU	VU
LU	LU	LU	LU	LU	U
E	E	E	E	E	E
LD	LD	LD	LD	LD	D
HD	HD	HD	HD	HD	VD

حال نوبت به ماتریس میانگین مراکز \bar{Y} رسیده است که مرکز (میانگین) منحنی های قوانین را به جای خود قوانین در ماتریس ذخیره می کند. (در C# جهت مثبت محور y بطرف پایین است)

VR	L	E	R	VR	
-20	-20	-20	-20	-20	VU
-15	-15	-15	-15	-15	U
0	0	0	0	0	E
15	15	15	15	15	D
20	20	20	20	20	VD

از روی ماتریس قوانین و میانگین مراکز \bar{Y} می توانیم مجموعه فازی خروجی ، برای سرعت روباه بر روی محور Y را تعیین کنیم: (حداکثر سرعت روباه 22 در نظر گرفته شده است)



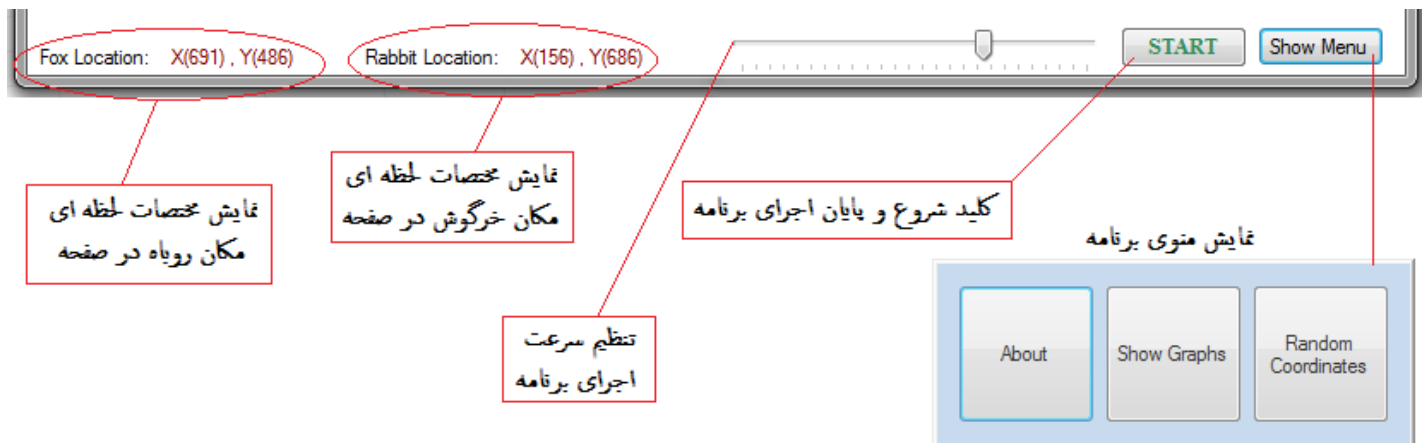
میانگین \bar{x}	واریانس σ	نام منحنی گوسین	نام منحنی گوسین
20	7	High Up	HU
15	4	Low Up	LU
0	5	Equal	E
-15	4	Low Down	LD
-20	7	High Down	HD

موتور استنتاج در این سیستم نیز همانند سیستم اول می باشد با این تفاوت که ماتریس Rulefire به ماتریس میانگین مراکز \bar{Y} بدست آمده از این سیستم ضرب داخلی می شود و نتیجه را می سازد.

❖ شرح نمای کاربری برنامه شبیه ساز:

مهمترین فاکتور در این برنامه اندازه صفحه برنامه می باشد که می توانید بر حسب نیاز آن را تغییر دهید ولی باید در نظر داشته باشد که باید از گزینه show graphs تنظیمات مربوطه را انجام دهید تا در هر لحظه خروجی قابل قبولی را دریافت کنید.

در زیر منوی کاربری برنامه شرح داده شده است:



❖ پیش نیاز نرم افزاری سیستم:

این برنامه در محیط Visual Studio 2008 Service Pack 2 تحت .Net Framework 3.5 نوشته شده است. بنابراین برای اجرای برنامه در کامپیوترهای مختلف نیاز است تا بعضی از نرم افزارها از قبل در سیستم نصب شده باشند.

برای دیدن سورس کدهای برنامه و اجرای آنها می توان از Visual Studio 2008 یا ورژنهای بالاتر استفاده نمود. برای اجرای تنها فایل اجرایی برنامه، می بایست فایل کتابخانه ای ZedGraph.dll در همان مسیر فایل FoxRabbit.exe قرار گیرد. (در شاخه Bin این عمل انجام گرفته است)

مسیر فایل اجرایی برای سیستمهای 32 بیتی در داخل پوشه زیر می باشد:

FoxRabbit \ FoxRabbit \ bin \ x86 \ Debug \ FoxRabbit.exe

مسیر فایل اجرایی برای سیستمهای 64 بیتی در داخل پوشه زیر می باشد:

FoxRabbit \ FoxRabbit \ bin \ Debug \ FoxRabbit.exe

نرم افزارهای پیش نیاز برای ویندوز XP و یا پایین تر :

.Net Framework 3.5

Windows Installer 3.1

نرم افزارهای پیش نیاز برای ویندوز 7 , Vista و یا بالاتر:

.Net Framework 3.5

برای راحتی کار با برنامه و اجرای آن یک فایل نصب در شاخه:

FoxRabbit \ publish \ setup.exe

قرار گرفته است که خود تمام برنامه های پیش نیاز نرم افزاری را در صورت عدم وجود نصب می کند و یک میانبر برنامه در دسکتاپ کامپیوتر قرار می دهد.